



DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

**“DESARROLLO DE UNA NARIZ ELECTRÓNICA PARA UN ROBOT  
DETECTOR DE GASES TÓXICOS”**

POR

**Daniel Alberto García Salomón**

**TESIS**

**PRESENTADA COMO REQUISITO PARCIAL PARA OBTENER EL  
GRADO DE MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA**

**DIRECTOR DE TESIS**

**Dr. José Luis Meza Medina**

**ISSN: 0188-9060**

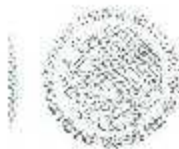


**RIITEC: (16)-TMCIE-2014**

Torreón, Coahuila, México,  
Diciembre 2014

SEP

SECRETARÍA DE  
EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO  
Instituto Tecnológico de La Laguna

"2014, Año de Octavio Paz

Torreón, Coah., 03/Diciembre/2014

**DR. JOSE LUIS MEZA MEDINA**  
**JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**

Por medio de la presente, hacemos de su conocimiento que después de haber sometido a revisión el trabajo de tesis titulado:

**"Desarrollo de una nariz electrónica  
para un robot detector de gases tóxicos"**

Desarrollado por el **C. Daniel Alberto García Salomón**, con número de control **M1213018** y habiendo cumplido con todas las correcciones que se le indicaron, estamos de acuerdo que se le conceda la autorización de la fecha de examen de grado para que proceda a la impresión de la misma.

**ATENTAMENTE**  
EDUCACIÓN TECNOLÓGICA FUENTE DE INNOVACIÓN

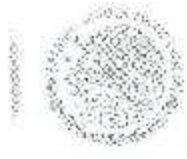
  
**Dr. José L. Meza Medina**  
**Asesor/Director de Tesis**

**Dr. Francisco Valdés Perezgasga**  
**Comité Tutorial**

  
**Dr. Juan Sitientes Mijares**  
**Comité Tutorial**

  
**M.C. Martín Vázquez Rueda**  
**Comité Tutorial**





2014, Año de Octavio Paz

Torreón, Coah., **03/Diciembre/2014**  
Dependencia: DEPI/CPCIE  
Oficio: DEPI/CPCIE/150/2014  
Asunto: Autorización de  
impresión de tesis.

**C. DANIEL ALBERTO GARCÍA SALOMÓN**  
**CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA.**  
**PRESENTE**

Después de haber sometido a revisión su trabajo de tesis titulado:

**“Desarrollo de una nariz electrónica  
para un robot detector de gases tóxicos”**

Habiendo cumplido con todas las indicaciones que el jurado revisor de tesis hizo, se le comunica que se le concede la autorización con número de registro **RIITEC: (16)-TMCIE-2014**, para que proceda a la impresión del mismo.

**ATENTAMENTE**  
EDUCACIÓN TECNOLÓGICA FUENTE DE INNOVACIÓN

**DR. JOSÉ LUIS MEZA MEDINA**  
Jefe de la División de Estudios de Posgrado e Investigación  
del Instituto Tecnológico de la Laguna

SECRETARÍA DE  
EDUCACIÓN PÚBLICA  
INSTITUTO TECNOLÓGICO  
de la Laguna  
División de Estudios de Posgrado e Investigación

JLM/RHSR



División de Estudios de Posgrado  
Instituto Tecnológico de La Laguna  
Apartado Postal 681  
Boulevard Revolución y Calzada Cuauhtémoc  
Torreón, Coahuila, México, C.P. 27000.

Division of Postgraduate Studies  
La Laguna Institute of Technology  
P.O. Box 681  
Boulevard Revolución and Calzada Cuauhtémoc  
Torreón, Coahuila, Mexico, C.P. 27000.

## Dedicatorias

*A mis padres*  
*José Pilar García Altamirano*  
*Florinda Salomón García*

*A mis hermanos*  
*José Ricardo García Salomón*  
*Flor Elisa García Salomón*

*con todo mi cariño*

*Amor y verdad son las dos cosas de Dios,  
verdad es el fin, y el amor es el camino*  
*(Mahatma Gaudhi)*

## Agradecimientos

La presente Tesis es un esfuerzo en el que directa o indirectamente han participado varias personas, opinando, corrigiendo, dando ánimo, aconsejando, acompañando en los momentos difíciles y en los momentos de felicidad.

Agradezco a Dios por todas las cosas que me ha dado en la vida y permitirme alcanzar este momento.

Quiero expresar mi agradecimiento por la confianza, interés y dirección de este trabajo al Dr. José Luis Meza Medina, por sus valiosos consejos y ayuda a la Ing. Sandra Casillas Bolaños, por la atenta lectura, comentarios y atinadas correcciones al Dr. Francisco Valdés Pérezgasga, al Dr. Juan Sifuentes Mijares y al M.C. Martín Vázquez Rueda.

Gracias también a mis Padres y hermana que me han acompañado en esta importante etapa de mi vida, por su apoyo incondicional, por entender mis ausencias y malos momentos.

Agradezco también al Consejo Nacional de Ciencia y Tecnología (CONACYT) por su apoyo para la realización de este trabajo de tesis.

Gracias a todos.

# Desarrollo de una Nariz Electrónica para un robot detector de gases tóxicos

Daniel Alberto García Salomón

## Resumen

Para este trabajo se desarrolla un prototipo de nariz electrónica con el objetivo de ser implementado en una plataforma robótica para la detección de gases tóxicos en ambientes de riesgo para el ser humano.

Para llevar a cabo este trabajo se estudia la teoría de redes neuronales para el reconocimiento de patrones así como de la teoría de los fundamentos de las narices electrónicas.

Se utiliza un arreglo de sensores comúnmente llamado matriz de sensores el cual crea una huella digital de gas que se quiere clasificar, haciendo posible identificarlo de otros, esto se logra debido a que los sensores reaccionan de acuerdo a los compuestos que se le presentan y dependiendo de su sensibilidad podemos obtener una señal eléctrica que se utiliza para entrenar la Red Neuronal.

Una Red Neuronal Backpropagation es creada mediante el "Toolbox" de Redes Neuronales de Matlab para el reconocimiento de patrones de los gases muestreados y se crea una Interfaz gráfica con la herramienta GUIDE de Matlab para la comunicación con el usuario.

**Palabras clave:** Neuroná Biológica, Nariz Electrónica, Redes Neuronales, Reconocimiento de patrones, Red Backpropagation, Detector de gas.



# Development of an Electronic Nose for Toxic gas detector robot

Daniel Alberto García Saizomón

## Abstract

For this work a prototype electronic nose was developed in order to be implemented on a robotic platform for the detection of toxic gases in hazardous environments for humans.

To carry out this work the theory of neural networks for pattern recognition and theory of the foundations of electronic noses is studied.

An array of sensors commonly called sensor matrix, is used to create a digital fingerprint of the gas that is to be classified, making it possible to identify other one, this is possible because the sensors react according to the compounds that are presented and depending its sensitivity we can obtain an electrical signal that is used to train the neural network.

A Backpropagation neural network is created using the "Toolbox" of Matlab Neural Networks for Pattern Recognition sampled gases and a graphical interface is created with the Matlab GUIDE tool for communication with the user.

**Key words:** Biological Neuron, Electronic Nose, Neural Networks, Pattern Recognition, Backpropagation Network, Gas detector.

---

# ÍNDICE GENERAL

<b>1. Introducción</b>	<b>1</b>
1.1. Planteamiento del problema	3
1.2. Objetivo de la tesis	3
1.2.1. Objetivos específicos	3
<b>2. Fundamentos de las narices electrónicas</b>	<b>4</b>
2.1. Olfato humano	4
2.1.1. El sistema olfativo	4
2.1.2. Proceso del sistema olfativo	5
2.2. Nariz electrónica	6
2.2.1. Orígenes de la nariz electrónica	7
2.2.2. Comparación de la nariz electrónica y el olfato humano	8
2.2.3. Aplicaciones	9
2.2.4. Componentes de la nariz electrónica	11
<b>3. Algoritmos de reconocimiento de patrones con redes neuronales</b>	<b>18</b>
3.1. Antecedentes de las redes neuronales	18
3.2. Neurona biológica	22
3.3. Neurona artificial	24
3.3.1. Ventajas de las redes neuronales	25
3.3.2. Niveles o capas de una red neuronal	27
3.4. Tipos de entrenamiento de las Redes Neuronales	29
3.4.1. Entrenamiento supervisado	29
3.4.2. Entrenamiento no supervisado	31

3.3. Perceptrón	33
3.3.1. Separación lineal	34
3.6. Red backpropagation	35
3.6.1. Arquitectura de la red Backpropagation	37
3.6.2. Propagación hacia adelante	38
3.6.3. Propagación hacia atrás	38
3.6.4. Algoritmo de Levenberg - Marquardt	40
3.6.5. Ventajas e inconvenientes	43
<b>4. Desarrollo del prototipo de nariz electrónica</b>	<b>44</b>
4.1. Sensores MQ	44
4.1.1. Respuesta de los sensores MQ	46
4.1.2. Circuito básico	47
4.2. Captura de los datos	48
4.2.1. Arduino Mega 2560	48
4.2.2. Pines de la tarjeta arduino	49
4.2.3. Enlace Arduino-Matlab	50
4.2.4. Módulo Xbee	51
4.3. Interfaz gráfica	55
4.3.1. Realización de la interfaz	55
4.3.2. Programación	56
<b>5. Resultados experimentales</b>	<b>62</b>
5.1. Pruebas experimentales	62
5.2. Entrenamiento de la red neuronal	66
5.3. Resultados	68
<b>6. Conclusiones</b>	<b>72</b>
6.1. Trabajos futuros	73
<b>Apéndices</b>	<b>74</b>
<b>A. Programa de la interfaz gráfica</b>	<b>74</b>
<b>B. Datos capturados de las pruebas experimentales</b>	<b>85</b>
B.1. Alcohol: Matriz "A"	85
B.2. Amoníaco: Matriz "A"	89
B.3. Metano: Matriz "M"	92



B.4. Gas LP: Matriz "LP"	96
<b>C. Resultados de entrenamiento de la red neuronal</b>	<b>100</b>
C.1. Resultado del entrenamiento de la Red Neuronal	100
<b>Bibliografía y Referencias</b>	<b>103</b>



---

## ÍNDICE DE FIGURAS

2.1. Olfato Humano [8]	6
2.2. Comparación del olfato humano y nariz electrónica [4]	8
2.3. Aplicación de una nariz electrónica en alimentos [4]	9
2.4. Aplicación de una nariz electrónica en la medicina [15]	10
2.5. Aplicación de una nariz electrónica en la medición de la calidad del aire [16]	11
2.6. Sistema de la Nariz Electrónica [4]	12
3.1. Neurona Biológica [51]	22
3.2. Comunicación entre neuronas y la liberación de neurotransmisores [50]	23
3.3. Neurona Artificial [30]	25
3.4. Ejemplos de una red totalmente conectada [30]	27
3.5. Tipos de Redes Neuronales [30]	28
3.6. Influencia de la salida de la neurona $N_j$ en la entrada de la neurona $N_i$ [51]	30
3.7. Esquema básico del Perceptrón [51]	33
3.8. Red perceptrón de dos capas	34
3.9. Ejemplo de separabilidad [51]	35
3.10. Estructura Backpropagation de tres capas [51]	37
3.11. Cálculo de los valores $\delta_j$ de la capa de salida [51]	39
3.12. Trayectoria del algoritmo Levenberg-Marquardt [50]	43
4.1. Sensores serie MQ [53]	45
4.2. Partes del sensor MQ [53]	45
4.3. Característica de sensibilidad el sensor MQ-8 [53]	46
4.4. Circuito básico [53]	47

4.5. Circuito para MQ-7 y MQ-9 [53]	48
4.6. Tarjeta Arduino Mega 2560 [54]	49
4.7. Módulo Xbee [55]	51
4.8. Circuito básico Xbee [55]	52
4.9. Aplicación XCTU	53
4.10. Comunicación punto a punto de Módulos Xbee [55]	54
4.11. Arduino Xbee Shield [52]	54
4.12. Explorador Xbee Sparkfun [55]	55
4.13. Pantalla de la Interfaz Gráfica	56
4.14. Funciones de Activación	60
4.15. Estructura de la Red en Matlab	60
5.1. Resistencia de calentamiento del sensor MQ [53]	63
5.2. Prototipo utilizado para el experimento	64
5.3. Matriz de sensores	61
5.4. Comunicación inalámbrica de la tarjeta Arduino	65
5.5. Esquema del experimento realizado	65
5.6. Matriz de sensores	66
5.7. Resultados del entrenamiento	69
5.8. Gráfica del entrenamiento	70
5.9. Identificación del Amoníaco	70
5.10. Pantalla de la identificación del Amoníaco	71



---

## ÍNDICE DE TABLAS

2.1. Técnicas de Análisis de Patrones Multivariables aplicadas a los datos de la nariz electrónica . . . . .	17
3.1. Funciones de Activación . . . . .	25
4.1. Sensores utilizados . . . . .	44
4.2. Partes del sensor . . . . .	46
4.3. Condiciones de trabajo para el sensor serie MQ . . . . .	47
4.4. Datos de la Arduino Mega 2560 . . . . .	49
4.5. Parámetros para la comunicación punto-punto . . . . .	54
4.6. Parámetros utilizados en la Red Neuronal . . . . .	61
5.1. Componentes del sensor MQ . . . . .	63

---

---

# CAPÍTULO 1

---

## INTRODUCCIÓN

---

El sistema olfativo humano, con toda su complejidad, a menudo no nos es suficiente. Es bien sabido que el hombre recurre a narices más sensibles como la de los perros cuando se trata de rastrear, detectar sustancias peligrosas, explosivas o ilegales. Antiguamente en las minas de carbón se bajaba con un canario que indicaba que la atmósfera era respirable mientras éste cantara (el monóxido de carbono y el metano mataban al pajarito antes que a los mineros). En ese caso no se pretendía aprovechar un sistema olfativo más sensible, sino utilizar al canario como alarma sonora y visual ante la imposibilidad de evaluar la calidad del aire por parte del hombre.

De la idea de emular el sistema olfativo biológico con los sensores de gas nace el concepto de nariz electrónica, que se define como un instrumento que consta de una colección de sensores químicos electrónicos y de un sistema de reconocimiento de patrones adecuado para reconocer olores (compuestos) simples o complejos y trata de caracterizar distintas mezclas de gases [1]. O sea, que la nariz electrónica va más allá del simple sensor e intenta, al igual que la nariz biológica, reconocer patrones de olores mediante complejos algoritmos numéricos.

La misión de este instrumento no es analizar cuantitativamente aromas complejos (determinar en qué cantidades se encuentran los compuestos químicos que los integran y en qué cantidades se encuentran), sino obtener información cualitativa del conjunto. En resumen, obtener, analizar y reconocer huellas olfativas de aromas complejos, valorando en su conjunto los componentes de la muestra aromática a analizar o clasificar, imitando de esta forma el funcionamiento del sistema olfativo humano [2].

Un olor es la respuesta exclusiva de cualquier sistema sensorial olfativo a un determinado compuesto químico disuelto en el aire, permitiendo su reconocimiento. A las moléculas de este compuesto en su fase gaseosa se las denomina volátiles. Mimetizando el anterior proceso, el funcionamiento de la nariz electrónica también dará una respuesta exclusiva cuando determinados volátiles se encuentren en el ambiente, lo que permitirá reconocer los compuestos de interés [3].





Las moléculas de olor son expuestas a la nariz electrónica, los patrones químicos presentes en la muestra del aroma son detectados por los sensores (que es el sistema de detección más utilizado), los cuales transforman esta entrada química en una señal eléctrica produciendo para cada aroma un único patrón de respuesta, designado como huella digital olfativa (la cual contiene idealmente la complejidad del aroma) [4].

Las respuestas de las señales generadas por los sensores se analizan utilizando diferentes métodos analíticos [5]. Los más reconocidos son: el análisis gráfico (diagrama de barras y diagramas polares), el análisis estadístico multivariado, análisis de componentes principales (PCA), el análisis de función de discriminantes (DFA) y el análisis de clusters (CA), redes neuronales artificiales (ANN) y funciones de base radial (RBF). La elección del método depende de los datos disponibles y del tipo de resultado requerido [6].

Las redes neuronales artificiales (ANN) son estructuras de procesamiento de información paralela y distribuida, que intentan emular las funciones computacionales elementales de la red nerviosa del cerebro humano, con base en la interconexión de multitud de elementos de procesamiento completamente local. Los modelos de ANN combinan modelos matemáticos de las células nerviosas y modelos de arquitecturas que describen las interconexiones existentes entre estas células [7].

El proceso de reconocimiento de patrones mediante redes neuronales se divide en dos etapas: una de entrenamiento de la red y otra de simulación de esta. En la primera, el sistema se adapta a las diferentes clases de aromas creando una matriz de pesos que utilizará más tarde para simular. Para este proceso es necesario que la estadística de las muestras del proceso cubra la estadística del espacio de aromas. Una vez entrenada la red introducimos el vector del que queremos saber la salida y esperamos resultados.

Desde la construcción de los primeros prototipos de narices electrónicas en los años 80, sus campos de desarrollo y sus aplicaciones en los campos de la alimentación, control de calidad, cura medioambiental, diagnóstico médico, control de procesos industriales, desarrollo de fragancias y cosméticos, seguridad y toxicología, etc. no han parado de crecer.

Todos estos avances han dado lugar a la fabricación comercial de narices electrónicas. La mayoría de las narices electrónicas diseñadas hasta la fecha son para uso comercial, algunas otras son utilizadas por los gobiernos para seguridad, y otras son usadas en equipos hospitalarios [4].

## 1.1. Planteamiento del problema

La exposición a gases tóxicos es una situación que se vive diariamente en numerosas industrias, en estos lugares se deben tomar medidas de prevención constantes para evitar situaciones que incluso pueden resultar en la muerte de alguna persona.

Ya sea en la industria minera, construcción, química o de servicios sanitarios, entre otras, siempre está presente el riesgo de inhalar gases tóxicos provenientes de diversos procesos productivos.

Las Narices Electrónicas son una excelente opción para la detección de gases tóxicos, ya que pueden determinar una gran cantidad gases, además montadas en una plataforma robótica pueden ser utilizadas en la exploración de minas o en procesos industriales para detectar fugas de algún gas tóxico.

Debido a la capacidad de las Narices Electrónicas para reconocer y comparar olores individuales o complejos nos da la posibilidad de analizar el olor o gas que la Nariz Electrónica está reconociendo gracias a las respuestas de los diferentes sensores que componen la matriz de sensores. Ya que el objetivo de una Nariz Electrónica es relacionar el olor o gas que está percibiendo con una respuesta almacenada en la memoria, entonces podemos generar una base de datos tan compleja como requiera la aplicación que le daremos a la nariz y así tener la posibilidad de analizar lugares donde exista un riesgo por exposición a gases mediante exploraciones.

El presente trabajo se pretende desarrollar un prototipo de Nariz Electrónica que reconozca diferentes gases tóxicos en lugares donde existen o se manejan una gran cantidad de gases o sustancias tóxicas que ponen en riesgo la salud del ser humano.

## 1.2. Objetivo de la tesis

El objetivo de este trabajo de tesis es desarrollar un prototipo de nariz electrónica para un robot que detecte gases tóxicos en un ambiente de riesgo para el ser humano mediante la teoría de reconocimiento de patrones con Redes Neuronales.

### 1.2.1. Objetivos específicos

- Elaborar una matriz de sensores para obtener la "huella digital" de los gases que se quieren reconocer.
- Crear y entrenar una Red Neuronal utilizando el Neural Network Toolbox de Matlab, que sea capaz de reconocer distintos gases.
- Crear una interfaz gráfica para el usuario de la Nariz Electrónica.
- Desarrollar un prototipo de Nariz Electrónica con comunicación inalámbrica al ordenador.



---

---

## CAPÍTULO 2

---

# FUNDAMENTOS DE LAS NARICES ELECTRÓNICAS

En esta sección se tratan los temas referentes a los conceptos relacionados con las narices electrónicas los cuales nos ayudarán a entender cómo se desarrollará el prototipo de nariz electrónica y las bases de su funcionamiento.

### 2.1. Olfato humano

Probablemente, el olfato es más antiguo y el menos comprendido de nuestros cinco sentidos. A través de la evolución se ha mantenido conectado con las partes del cerebro que se convirtieron en el archivo de la clasificación de nuestras respuestas emocionales, ligando íntimamente los olores de las cosas con nuestras emociones. Es importante porque a través de él, los animales encuentran su alimento a distinguir las sustancias comestibles de las que no son, encuentran pareja, detectan al depredador, ubican su hábitat, etc.

Sus funciones más importantes son las de sistema de alarma ponernos en guardia frente al peligro y como recolector de información nos proporciona valiosos datos sobre el mundo exterior. Los estímulos provenientes del medio ambiente transmiten información a los circuitos procesadores, que son el encéfalo y la médula espinal, en donde se interpretan y se envían las señales hacia los componentes periféricos que mueven el cuerpo.

#### 2.1.1. El sistema olfativo

El olfato es el sentido encargado de percibir y procesar los olores. Es el sistema sensorial más primitivo y el único directamente conectado con el sistema límbico el cual es un sistema formado por varias estructuras cerebrales que gestionan respuestas fisiológicas ante estímulos emocionales. Está relacionado con la memoria, atención, instintos sexuales, emociones (por ejemplo placer, miedo, agresividad), personalidad y la conducta.

La Nariz forma parte del sentido del olfato, del Sistema respiratorio y vocal. Se puede dividir en región externa, el apéndice nasal, y una región interna constituida por dos cavidades principales (fosas nasales) que están separadas entre sí por el tabique vertical. Los bordes de los orificios nasales están recubiertos de pelos fuertes que atraviesan las aberturas y sirven para impedir el paso de sustancias o partículas extrañas.

Las cavidades nasales son altas y profundas, y constituyen la parte interna de la nariz. Se abren en la parte frontal por los orificios nasales y, en el fondo, terminan en una abertura en cada lado de la parte superior de la faringe. La región olfativa de la nariz es la responsable del sentido del olfato, la membrana mucosa es muy gruesa y adopta una coloración amarilla.

El sentido del olfato humano es 10 mil veces más sensible que cualquier otro de nuestros sentidos, los receptores de la sensación olfativa son células nerviosas derivadas del propio sistema nervioso central y se estima que contamos con 100 millones de tales receptores, el olfato es el único lugar donde el sistema nervioso central está directamente expuesto al ambiente. La nariz humana distingue entre más de 10 000 aromas diferentes [8].

Se han identificado tres vías olfativas. La primera es conocida como sistema olfativo arcaico, que se encarga de los reflejos olfativos básicos; luego, un sistema llamado antiguo, que proporciona un control automático para el aprendizaje parcial de la ingestión de alimentos, así como el rechazo de alimentos tóxicos o poco saludables; finalmente existe una tercera vía, un sistema recientemente identificado que se encarga de la percepción consciente del olfato. Desde el punto de vista fisiológico, el sentido del olfato y el gusto están relacionados entre sí y son parte de nuestro sistema sensorial químico [8].

Se puede decir que hay siete tipos de receptores existentes en las células de la mucosa olfatoria. Las investigaciones sobre el olfato señalan que las sustancias con olores similares tienen moléculas del mismo tipo. Estos tipos son: almizcle, almizcle, flores, menta, éter (líquido para limpieza en seco), acre (avinagrado) y podrido, olores que corresponden a los siete tipos de receptores existentes en las células de la mucosa olfatoria. No obstante, los datos reportados en los últimos años sugieren que hay por lo menos 100 sensaciones primarias de olor [8].

### 2.1.2. Proceso del sistema olfativo

El proceso del olfato sigue los siguientes pasos.

- Las moléculas del olor en forma de vapor (compuestos químicos) que están flotando en el aire llegan a las fosas nasales y se disuelven en las mucosidades que se ubican en la parte superior de cada una de ellas [8].
- Debajo de las mucosidades se encuentran las células receptoras especializadas, también llamadas neuronas receptoras del olfato, las cuales detectan los olores desde los cilios denominados folículos o vesículas olfatorias [8].
- Las neuronas receptoras del olfato transmiten la información a los bulbos olfatorios que se encuentran en la parte de atrás de la nariz [8].



- Los bulbos olfatorios tienen receptores sensoriales que en realidad son parte del cerebro y que envían mensajes directamente a los centros más primitivos del cerebro, donde se estimulan las emociones y memorias (estructuras del sistema límbico), así como a los centros "avanzados", donde se modifican los pensamientos conscientes (neocórtex) [8].
- Estos centros cerebrales perciben los olores y tienen acceso a recuerdos que nos traen a la memoria personas, lugares o situaciones relacionadas con esas sensaciones olfativas [8].
- Finalmente, el epitelio olfativo tiene unas glándulas encargadas de segregar una solución enzimática cuya misión es eliminar las moléculas olorosas que han excitado las neuronas correspondientes, limpiando en cierto modo la mucosa olfativa de las sustancias presentes en ella ya detectadas [8].

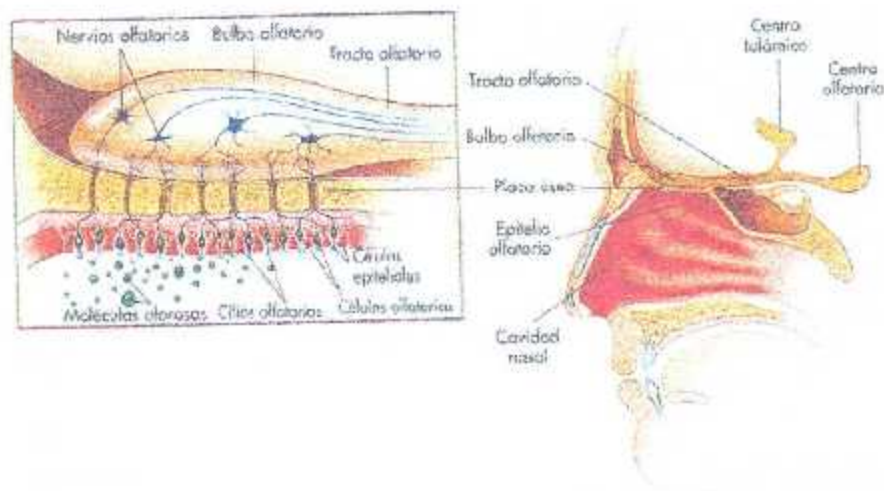


Figura 2.1: Olfato Humano [8]

## 2.2. Nariz electrónica

Con el afán de buscar condiciones de vida más seguras y confortables, se han construido robots para que reemplacen al hombre en ciertos tipos de tareas. A estos robots se les ha provisto de dispositivos que emulan algunos de los sentidos humanos, garantizando así que algunas tareas se realicen de manera más segura, confiable y menos costosa. Entre estos sentidos se destacan la visión, la audición, el olfato y el gusto [4].

Una Nariz Electrónica es un instrumento que puede decir, es decir, es capaz de realizar análisis cualitativos y/o cuantitativos de una mezcla de gases, vapores y olores. La nariz electrónica es un instrumento de olfato artificial que permite distinguir y reconocer aromas utilizando sensores de gas [9].

Una Nariz Electrónica tiene al menos 4 partes con diferentes funciones: la primera realiza la adecuación de la mezcla gaseosa y el muestreo, el conjunto de sensores de gas hace la detección, la electrónica de control se dedica a la gestión del conjunto de sensores y adecuación de la señal, y finalmente, la computadora, con adecuados algoritmos de clasificación de patrones, extrae los rasgos característicos o "huellas" de cada aroma y presenta los resultados en la interfaz con el usuario [10].

Una definición generalmente aceptada de un sistema de olfato electrónico puede ser: "instrumento que comprende una agrupación de sensores químicos con sensibilidades parcialmente solapadas junto a un sistema de reconocimiento de patrones, capaz de analizar y reconocer aromas simples o complejos" [1].

### 2.2.1. Orígenes de la nariz electrónica

Los orígenes de la nariz electrónica se remontan a los años 60, cuando la compañía Bacharach Inc., construyó un dispositivo conocido con el nombre de Sniffer, el cual constaba de un solo sensor de gas y por lo tanto, no se consideró una nariz electrónica. En la década de los 80, surgen dos grupos de investigadores, en la Universidad de Warwick en Gran Bretaña y en el Argonne National Laboratory (ANL) en Estados Unidos [4].

Posteriormente en Japón se comenzó a investigar la frescura de los pescados utilizando matrices de sensores MOX (sensores semiconductores de óxido metálico). Por lo tanto, en los tres continentes se inició el desarrollo de la tecnología de matrices de sensores olfativos en los años 80 [11].

Es precisamente en esta época cuando el concepto de nariz electrónica como sistema inteligente aparece realmente. Una de sus primeras y más populares definiciones es la de Gardner y Barlett: "Instrumento que comprende una agrupación de sensores químicos con sensibilidades parcialmente solapadas junto a un sistema de reconocimiento de patrones, capaz de analizar y reconocer aromas simples o complejos" [12].

Las narices electrónicas fueron desarrolladas en su comienzo para reemplazar a los paneles de expertos en la clasificación de aromas, debido a que estos se consideran costosos, difíciles de transportar, y algunas veces subjetivos, debido a que sus valoraciones se pueden ver afectadas por el cansancio, estado anímico, el clima y otros factores externos. Algunas veces estos paneles de expertos se han apoyado en cromatógrafos de gases y espectrómetros de masas, no sólo clasificando olores sino también cuantificándolos, todo esto con un elevado costo en tiempo y en dinero [4].

En la actualidad las narices electrónicas no son solamente usadas para clasificaciones de aroma, sino también para la detección de olores. De ahí que su importancia y utilización se haya extendido a aplicaciones en la agroindustria, medio ambiente, la seguridad, y la medicina. A continuación se recogen algunas investigaciones que se han realizado según el campo de aplicación de los sistemas olfativos artificiales [4].



### 2.2.2. Comparación de la nariz electrónica y el olfato humano

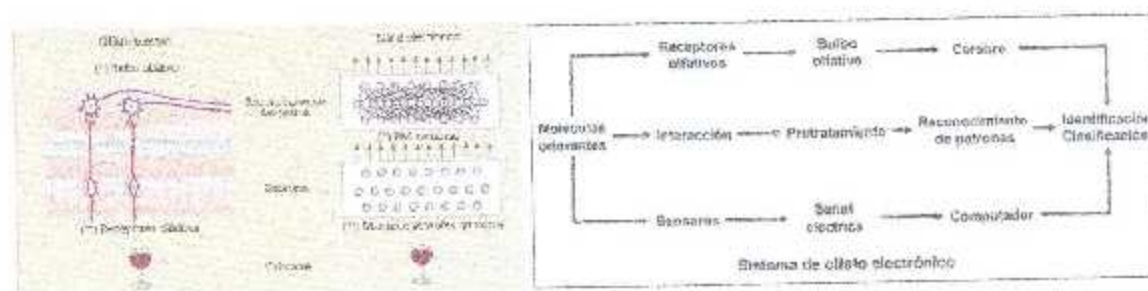
El sentido del olfato inicia su funcionamiento cuando es estimulado por las moléculas que hay en el aire que respiramos o por sustancias volátiles que contienen los alimentos que se llevan a la boca [4].

El sentido del olfato empieza con la nariz e incluye otras partes de la cabeza y el cerebro. En el proceso del olfato se ven implicadas las fosas nasales, las neuronas receptoras del olfato, el bulbo olfativo y el cerebro, que es donde se realiza el proceso de identificación de un aroma. Es importante mencionar que a lo largo de estos últimos años se han ido conociendo mejor los mecanismos del sistema del olfato humano, aunque todavía quedan detalles por comprender [13].

Cada olor que hay en la naturaleza es una mezcla de sustancias químicas y tiene sus propias características, que lo diferencian de los demás olores [4].

A pesar de que los seres humanos tienen una sensibilidad limitada (en comparación con algunos animales) son capaces de percibir, identificar y clasificar olores [4].

Las moléculas de olor son expuestas a la nariz electrónica, los patrones químicos presentes en la muestra del aroma son detectados por los sensores (que es el sistema de detección más utilizado), los cuales transforman esta entrada química en una señal eléctrica produciendo para cada aroma un único patrón de respuesta, designado como huella digital olfativa (la cual contiene idealmente la complejidad del aroma). Finalmente a esta respuesta se le aplican técnicas de reconocimiento de patrones para discriminar, clasificar y/o predecir el tipo de aroma que se está analizando [4].



a) Olfato humano

b) Nariz Electrónica

Figura 2.2: Comparación del olfato humano y nariz electrónica [4].

### 2.2.3. Aplicaciones

#### 2.2.3.1. Alimentación

En particular las narices electrónicas son de gran aplicación en el control de calidad de los productos alimenticios. Actualmente ya hay narices electrónicas que son capaces de clasificar vinos en función de sus aromas. Estudios similares se han realizado pero con jaranes como objeto del estudio.

También en el mundo de la alimentación se utilizan las narices electrónicas para controlar los aromas de frutas y pescados, con el fin de evitar el consumo de alimento vencido y para el control de cocción justa.

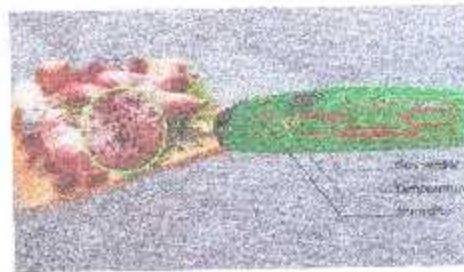


Figura 2.3: Aplicación de una nariz electrónica en alimentos [14].

#### 2.2.3.2. Diagnóstico Médico

Las narices electrónicas son capaces de analizar el aire exhalado (así como distintos olores del cuerpo humano) para diagnosticar diferentes patologías como la diabetes, insuficiencias hepáticas, infecciones respiratorias, etc.

El olor de la orina puede indicar o detectar problemas en el hígado o en la vejiga, o bien detectar diabetes. Y también tienen aplicación en la detección de enfermedades de la piel o infecciones bacterianas, tales como las que acompañan las heridas más comunes o en quemaduras.

Actualmente se está trabajando en la detección de cáncer a partir del análisis molecular del aire exhalado. Este nuevo método de diagnóstico permite la detección del cáncer en fases iniciales y es no-invasivo; además de aportar nueva información con respecto a los métodos tradicionales del análisis de sangre y orina. Estudios recientes demuestran que la detección de cáncer de pulmón y esófago en fases muy iniciales son posibles con el análisis del aire exhalado. En la figura 2.4 Se observa el análisis molecular del aire exhalado mediante una nariz electrónica.





Figura 2.4: Aplicación de una nariz electrónica en la medicina [15]

### 2.2.3.3. Medio Ambiente

Es un área de gran importancia, y la portabilidad de las narices electrónicas y su capacidad para identificar contaminantes, hacen que estos dispositivos tengan una gran aplicabilidad para el cuidado medioambiental. Se pueden utilizar para detectar mezclas de combustibles, pérdidas de aceite, olores de efluentes industriales y urbanos, identificación de residuos tóxicos, control de la calidad del aire en ambientes urbanos, emisiones gaseosas de industrias, etc.

La nariz electrónica permite localizar la fuga de una especie gaseosa e identificar el compuesto o la mezcla potencialmente peligrosos. Con esa información los equipos de emergencia pueden seleccionar la estrategia de contención y de protección más eficaz para cada caso específico.

Durante las sequías del verano, la tierra seca emite etileno que contribuye a la inflamabilidad de zonas de vegetación densa. En esas situaciones de riesgo de incendio, las narices electrónicas podrían ayudar a la prevención de incendios. En zonas industriales, la emisión de gases inflamables puede producir fuegos. En ausencia de oxígeno, se crean fuegos sin llamas que son difíciles de detectar. Sensores de gas, ej. de CO o de NO<sub>2</sub>, pueden dar alarma y ayudar a prevenir dichos fuegos.

Es conocido, que en domicilios la emisión de gases puede causar explosiones. Sensores de gas pueden avisar a la gente cuando hay peligro. A los vertederos se destinan restos orgánicos, plásticos, papel, vidrio, metales, pinturas, pañales, baterías, y muchas cosas más. Durante la descomposición se forman lixiviados que arrastran los productos tóxicos presentes que contaminan las aguas subterráneas y se liberan al aire importantes cantidades de gases como metano, CO<sub>2</sub> o gases tóxicos. Durante incendios, se pueden liberar a la atmósfera gases tan tóxicos como las dioxinas. Las narices electrónicas se pueden y en unos lugares se están utilizando para la detección preventiva de gases inflamables para prevenir incendios. También se están utilizando robots para la identificación de tierra contaminada para poder cambiarla efectivamente.



Figura 2.5: Aplicación de una nariz electrónica en la medición de la calidad del aire [16].

#### 2.2.3.4. Aplicaciones para el futuro

El objetivo de las narices electrónicas es mejorar las condiciones de vida, y con ese afán se han desarrollado aplicaciones para los campos de la alimentación, diagnóstico médica y el medio ambiente. Pero las narices electrónicas todavía tienen camino por recorrer en esos campos.

Hoy en día, en el mundo agro-alimentario, es de vital importancia conocer el origen y los procesos seguidos para la obtención de los alimentos, puesto que esos factores determinan su calidad. Se puede idear un sistema olfativo que determine la usidad y el origen de los alimentos en función de sus aromas, llegando a averiguar, por ejemplo, si un alimento está o no libre de transgénicos.

Los efectos a la exposición a sustancias tóxicas no sólo dependen de la concentración de dicha sustancia, sino que también son función del tiempo de exposición. Así pues, las narices electrónicas son capaces de monitorizar el perfil de concentraciones a las que se ha estado expuesto y evaluar los riesgos.

#### 2.2.4. Componentes de la nariz electrónica

Lo más importante para una nariz electrónica es tomar una muestra de algún tipo de aroma o muestra gaseosa y poder identificarla, clasificarla y/o cuantificarla. Esto significa que una nariz electrónica está constituida por varias partes: el sistema de muestreo, la matriz de sensores (sistema de detección más utilizado), el sistema electrónico que extrae los parámetros fundamentales de la señal proveniente del sensor, y el ordenador donde se llevará a cabo la identificación del olor, usando técnicas de reconocimiento de patrones [4].

En el sistema de la nariz electrónica existen tres módulos, a saber: químico, electrónico y el software. La parte química es la que se refiere a acondicionar la muestra que se tenga y a la medición que realizan los sensores; la parte electrónica es la que se refiere al acondicionamiento de la señal eléctrica que se obtiene a la salida de los sensores, y a la extracción de rasgos o características eléctricas que nos brindan cada uno de los sensores de la matriz; y la parte del software es la que se refiere al reconocimiento de la señal y a su respectiva visualización en el sistema [17].



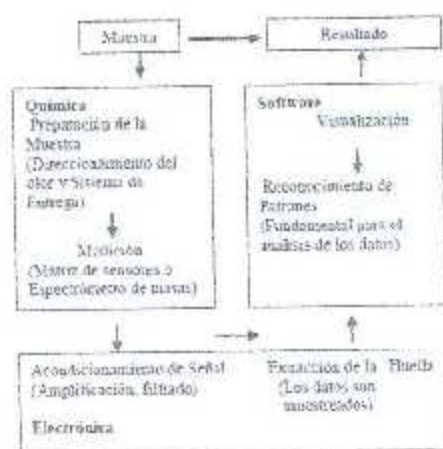


Figura 2.6: Sistema de la Nariz Electrónica [4].

### 2.2.4.1. Sensores químicos

En la nariz electrónica, los encargados de crear una señal eléctrica para que posteriormente el procesado matemático y el correspondiente patrón reconozcan el aroma o gas son los sensores. Hay distintas tecnologías de sensores, cada una con sus puntos fuertes de aplicación en función del gas a detectar y las concentraciones esperadas para la medida, las condiciones ambientales, el tiempo de vida requerido para el sensor, etc.

En la pasada década se desarrollaron diferentes clases de sensores químicos para la determinación cuantitativa y cualitativa de los compuestos volátiles y para encontrar la relación con la calidad sensorial del producto[18].

Algunos tipos de sensores químicos son:

#### 1.- Sensores Electroquímicos

Están formados por dos electrodos sumergidos en un medio electrolítico común. El electrolito es aislado de las influencias externas mediante una barrera, que puede ser una membrana permeable al gas, un medio de difusión o un capilar.

Durante el funcionamiento, un voltaje polarizado es aplicado a los electrodos y cuando el gas penetra en el sensor una reacción redox genera una corriente eléctrica proporcional a la concentración del gas. Se pueden utilizar sensores electroquímicos específicos de gas para detectar la mayoría de los gases tóxicos comunes, incluidos CO, H<sub>2</sub>S, Cl<sub>2</sub>, SO<sub>2</sub> etc. en una amplia variedad de aplicaciones de seguridad. Los sensores electroquímicos son compactos, requieren muy poca energía, muestran una gran linealidad y repetitividad, y generalmente tienen una larga vida útil, normalmente de uno a tres años.

#### 2.- Sensores de Conductividad Térmica

Consisten en la disposición de al menos dos filamentos con propiedades conductoras y térmicas ("termistores") formando parte de un "puente de Wheastone". Cada filamento se ubica



en una célula independiente y el conjunto está a una temperatura definida. En la célula de referencia se encierra una cantidad determinada de un gas estándar (por ejemplo aire) En la célula de medida penetra el gas a detectar. Su conductividad térmica, diferente de la del gas de referencia, hace que la temperatura del filamento se altere y, en consecuencia, se desequilibre el circuito "puente de Wheastone".

### 3.- Sensores Catalíticos

Consiste en un pequeño elemento denominado "pellistor", "perla" que está formado por un filamento de Platino calentado eléctricamente. Este filamento está recubierto primariamente con una base cerámica y posteriormente por una dispersión catalítica de Paladio o Rodio en un sustrato de Torio.

Cuando una mezcla de aire y gas inflamable se pone en contacto con la superficie caliente del catalizador, se produce una combustión que aumenta la temperatura de la "perla" lo cual altera la resistencia del filamento de platino que a su vez es medida en un circuito tipo "puente de Wheastone". El cambio de resistencia está directamente relacionado con la concentración de gas presente.

### 4.- Sensores Infrarrojos

El sensor infrarrojo de punto se basa en el hecho de que muchos gases combustibles tienen bandas de absorción en el espectro infrarrojo.

Esta técnica funciona bajo el principio de absorción de infrarrojos de doble longitud de onda, según el cual la luz atraviesa la mezcla en dos longitudes de onda, una de las cuales se ajusta al pico de absorción del gas que se pretende detectar, mientras que la otra no. Las dos fuentes de luz se pulsan alternativamente y se guían a lo largo de un camino óptico común para que salgan a través de una "ventana" con protección anti deflagración y, a continuación, a través del gas de muestra. Posteriormente, un retro reflector refleja otra vez los haces, regresando una vez más a través del gas para volver a la unidad. Aquí un detector compara las fuerzas de las señales de los haces de referencia y muestra y, por medio de una resta, se proporciona una medida de la concentración de gas.

### 5.- Sensor Semicondutor

El sensor es fabricado con materiales semiconductores opera por la propiedad de adsorción de gas en la superficie de un óxido calentado depositada en una base de sílice. En la mayoría de casos el proceso es muy similar al utilizado en la manufactura de "chips".

La absorción de la muestra de gas en la superficie del óxido, seguida de una oxidación catalítica, termina en un cambio de la resistencia eléctrica del material oxidado que puede relacionarse con la concentración del gas.

Todos los tipos de sensores muestran interacciones físicas y químicas con los compuestos volátiles que se quieren medir. Los sensores integrados a una nariz electrónica deben cumplir los siguientes criterios: alta sensibilidad para los compuestos químicos, similar a la de la nariz humana; baja sensibilidad a la humedad y la temperatura; alta reproducibilidad estabilidad; corto tiempo de reacción y fácil calibración [19].

Las tecnologías utilizadas actualmente en narices electrónicas son: semiconductores de óxi-



dos metálicos (MOS), transistores de campo eléctrico con semiconductores de óxido metálico (MOSFET), conductores de polímero orgánico (CP) y cristales piezoeléctricos (BAW) [10]. Los sensores MOS tienen selectividad y reproducibilidad pobres, pero son poco dependientes de la humedad, lo que les da una vida útil de entre 3 y 5 años; los sensores MOSFET y CP tienen una aceptable sensibilidad, pero sus respuestas dependen de la temperatura de operación y, en gran medida, de la humedad, características que reduce su vida útil a tiempos que oscilan entre 1 y 3 años [20].

Los sensores, generalmente, dan una respuesta típica que representa la medida del cambio de algún parámetro físico, por ejemplo, conductividad o corriente. Los tiempos de respuesta van de segundos a minutos, y su valor cambia de acuerdo con el estímulo recibido que proviene de los compuestos aromáticos de la muestra.

La nariz electrónica está diseñada con una serie de sensores, cada uno de ellos con cierta especificidad en un grupo de compuestos. Cada sensor, entonces, dará un valor de conductividad, obteniéndose al final una matriz de datos de  $m$  columnas por  $n$  filas, donde las columnas serán el número de sensores que posea la nariz y  $n$  el número de muestras [21].

#### 2.2.4.2. Procesamiento de la señal

En la etapa del pre-procesado de la señal, la cual se refiere a la aplicación de una serie de posibles operaciones a la señal proveniente de las etapas electrónicas para explotar la información que estos datos representasen, se extrae información de la respuesta de los sensores y se preparan los datos para el análisis de patrones, convirtiéndose en la primera etapa, donde los datos son muestreados y almacenados en la memoria del ordenador. Por lo tanto, la selección de esta señal puede tener una influencia significativa en el funcionamiento de las etapas siguientes [4].

La manipulación de la respuesta del sensor con respecto a su línea base es la primera etapa del pre procesado de la señal, que consiste en la manipulación de la respuesta del sensor con respecto a su línea base, es decir, a su respuesta a una muestra de referencia; con el propósito de compensar sus derivas entre otras cosas [4].

La técnica más utilizada consiste en dividir la respuesta del sensor ante una muestra entre la respuesta del sensor a la muestra de referencia, proporcionando una respuesta sin dimensiones ( $R_s/R_0$  donde  $R_s$  es la respuesta del sensor ante un gas y  $R_0$  es la respuesta del sensor ante una referencia usualmente aire).

La segunda etapa del pre procesamiento es comprimir la respuesta de la matriz de sensores para formar un vector de características o la huella digital olfativa. En principio, con la compresión se pretende transportar la misma información, pero empleando la menor cantidad de espacio. De acuerdo al procedimiento empleado para generar la huella digital olfativa dinámica [4].

Los métodos globales son procedimientos típicamente utilizados para asegurar que las magnitudes del sensor sean comparables, evitando que los procedimientos siguientes de reconocimiento de patrones sean saturados por sensores con valores arbitrariamente grandes [4].



### 2.2.4.3. Análisis de Patrones

Los datos obtenidos de la etapa anterior, son finalmente analizados por técnicas de reconocimiento de patrones, las cuales actúan como un sistema de procesamiento de la señal.

El conjunto de sensores genera una matriz, donde las columnas representarán un vector respuesta asociado con un olor en particular, las filas son la respuesta de un sensor individual a las diferentes medidas.

$$R = \begin{pmatrix} X_{11} & X_{12} & X_{1m} \\ X_{21} & X_{22} & X_{2m} \\ \dots & X_{i2} & \dots \\ X_{n1} & X_{n2} & X_{nm} \end{pmatrix}$$

Donde  $X_{ij}$  es la respuesta del sensor  $i$  a la muestra de olor  $j$

Los sensores de olor al igual que su contra parte humana no son completamente específicos para un determinado olor, sino que poseen especificidades solapadas, por lo tanto, un sensor de olor responderá a una variedad de olores pero con una sensibilidad variante (por ejemplo, rapidez e intensidad de la respuesta). Como resultado, los términos diferentes de la diagonal de la matriz de respuesta son comúnmente diferentes de cero, y precisamente bajo estas condiciones, se requieren los patrones de reconocimiento, para procesar los datos y solucionar el problema de la predicción de clase [4].

Las técnicas de Reconocimiento de Patrones aplicadas a los de la matriz electrónica según Pearse T. pueden ser [22]:

**Paramétricas:** presuponen una distribución teórica de probabilidad para la distribución de los datos. Se puede estimar cualquier parámetro como la media, varianza, proporción, etc... de los datos, es decir, que pueden ser descritas por una función de densidad probabilística.

**No Paramétricas:** Son técnicas estadísticas que no suponen ningún modelo probabilístico teórico. Son menos potentes que las técnicas paramétricas, aunque tienen la ventaja que se pueden aplicar más fácilmente. Este tipo de técnica ha dado lugar a las redes neuronales y a los sistemas expertos y son inspiradas biológicamente.

Y estas a su vez pueden ser:

1.- **Técnicas Supervisadas:** que son aquellas que utilizan el aprendizaje previo, de tal manera que un vector de olor desconocido es analizado usando las relaciones encontradas previamente de un grupo de vectores de olores conocidos usados en la calibración inicial [23].

2.- **Técnicas No supervisadas:** que son aquellas que no utilizan el aprendizaje previo, discriminando entre los vectores de olores desconocidos sin la presentación del descriptor correspondiente. También llamados algoritmos de descubrimiento del conocimiento, se usan generalmente para la extracción de información útil a partir de grandes volúmenes de datos



[22].

Recientemente se han propuesto modelos de clasificación para realizar predicciones basadas en técnicas de descubrimiento del conocimiento tales como el análisis de asociación.

Algunas de las técnicas de reconocimiento de patrones más utilizadas son:

1.- Análisis Discriminante Lineal (LDA): El Análisis Discriminante equivale a un análisis de regresión donde la variable dependiente es categórica y tiene como categorías la etiqueta de cada uno de los grupos, las variables independientes son continuas y, determina a qué grupos pertenecen los objetos. Se trata de encontrar relaciones lineales entre las variables continuas que mejor discriminen en los grupos dados a los objetos. Además, se trata de definir una regla de decisión que asigne un objeto nuevo, que no sabemos clasificar previamente, a uno de los grupos predichos. Existen muchas maneras de realizar un Análisis de Función Discriminante, pero el método clásico es el Análisis Discriminante Lineal, en donde un conjunto de datos es linealmente separable siempre que exista un hiperplano que separe efectivamente los datos pertenecientes a cada una de las diferentes clases. Tiene un procedimiento de clasificación en el cual se considera que las clases tienen una distribución normal y una dispersión igual. El procedimiento LDA maximiza la variación entre las categorías y reduce al mínimo la variación dentro de las categorías para optimizar la resolución entre las clases. Este método siempre proporcionará una función que separe a las clases por hiperplanos, cuantificando al mismo tiempo la calidad de la función de discriminación a través del valor propio, la proporción de varianza explicada, el coeficiente de correlación canónica y los cálculos de los datos que hayan sido clasificados correctamente o incorrectamente [24].

2.- Análisis de Componentes Principales (PCA): es una técnica estadística utilizada para reducir la dimensionalidad (número de variables) de un conjunto de datos. El PCA busca la proyección según la cual los datos queden mejor representados en términos de mínimos cuadrados. Este método construye una transformación lineal que escoge un nuevo sistema de coordenadas para el conjunto original de datos en el cual la varianza de mayor tamaño del conjunto de datos es capturada en el primer eje (llamado el Primer Componente Principal), la segunda varianza más grande es el segundo eje, y así sucesivamente. Para construir esta transformación lineal debe construirse primero la matriz de coeficientes de correlación. Debido a la simetría de esta matriz existe una base completa de vectores propios de la misma. La transformación que lleva de las antiguas coordenadas a las coordenadas de la nueva base es precisamente la transformación lineal necesaria para reducir la dimensionalidad de datos. Una de las ventajas de PCA para reducir la dimensionalidad de un grupo de datos, es que retiene aquellas características del conjunto de datos que contribuyen más a su varianza manteniendo un orden de bajo nivel de los componentes principales e ignorando los de alto nivel. El objetivo es que esos componentes de bajo orden a veces contienen el aspecto más importante de esa información. Además, es un método lineal sin supervisión que ha sido ampliamente usado en varias investigaciones de matrices electrónicas, tanto en obras simples como complejas [4].

La naturaleza de los datos de la matriz electrónica puede ser tan compleja que a menudo es necesario utilizar un método de reconocimiento de patrones más poderoso, que sea capaz de tratar con datos no lineales, y que tenga ventajas futuristas sobre los métodos convencionales, tales como capacidad de aprendizaje, auto organización, generalización y tolerancia al



mido, para tal fin, se han desarrollado métodos inspirados biológicamente tales como MLP (Perceptrón multicapa), ANN (Redes Neuronales Artificiales), FIS (Sistema de inferencia borrosa), GA (Algoritmos genéticos), entre otros [4].

Las técnicas de reconocimiento de patrones basadas en las redes neuronales artificiales, han sido ampliamente usadas para aplicaciones de sensores de gas.

En la tabla 2.1 se muestran técnicas de análisis multivariantes aplicadas a los datos de la Nariz Electrónica [22].

Tipo de análisis	Método	Entrenamiento	Técnica de análisis
Quimiometría y Estadística	Cuantitativo		MLR
			PLS
	Análisis de patrones	No supervisados	NFS
			Wavelets
		Supervisados	PCR
			DEA LDA
Inspirados Biológicamente	ANN: Redes Neuronales Artificiales	No Supervisados	SOM
			MLP
		Supervisados	MLP
			PNN
			RBF
			LVQ
			ART
			Fuzzy Artmap
	Métodos Borrosos	Supervisados por sí mismos	FIS
			FNN
		Supervisados	FCM
			GA
	Otros	Supervisados por sí mismos	NFS
			Wavelets

Tabla 2.1: Técnicas de Análisis de Patrones Multivariantes aplicadas a los datos de la nariz electrónica

- |   |  |
|---|--|
| MLR.- Regresión lineal Múltiple           | PLS.- Mínimos cuadrados parciales              |
| PCA.- Análisis de componentes principales | CA.- Análisis Cluster                          |
| NN.- El vecino más próximo                | Ward.- Agrupación jerárquica                   |
| PCR.- Regresión del Componente Principal  | DEA.- Análisis de la Función discriminante     |
| LDA.- Análisis discriminante lineal       | SOM.- Mapas Autoorganizados                    |
| MLP.- Perceptrón Multicapa                | PNN.- Redes Neuronales Probabilísticas         |
| RBF.- Redes de Funciones de Base Radial   | LVQ.- Cuantificación del vector de aprendizaje |
| ART.- Teoría de la Resonancia Adaptativa  | Fuzzy Artmap.- Lógica Borrosa con mapas Art    |
| FIS.- Sistema de Inferencia Difusa        | FNN.- Redes Neuronales Borrosas                |
| FCM.- Agrupaciones Borrosas               | GA.- Algoritmos Genéticos                      |
| NFS.- Sistemas Neuroborrosos              | Wavelets.- Análisis de ondas                   |





---

---

## CAPÍTULO 3

---

# ALGORITMOS DE RECONOCIMIENTO DE PATRONES CON REDES NEURONALES

Las redes Neuronales Artificiales o Redes Neuronales, constituyen un paradigma computacional alternativo al tradicional basado en la programación de secuencia de instrucciones; el Enfoque conexionista está inspirado en conocimientos provenientes de la Neurociencia pero sus métodos provienen de diversas disciplinas como la Física Estadística, Psicología, Teoría del Conocimiento, Teorías de Sistemas, entre otras. Sus aplicaciones potenciales se orientan principalmente al campo de las Ciencias de la Computación y de la Ingeniería, entre las que se pueden mencionar: la extracción de características desde un conjunto de datos complejos, aplicaciones orientadas al reconocimiento óptico de caracteres (OCR) y procesamiento de imágenes; implementación directa y paralela de algoritmos de búsqueda y correspondencia; problemas que deben manejar datos contradictorios, difusos, probabilísticos e incompletos. Cabe destacar que las redes neuronales están particularmente bien situadas con respecto a las aplicaciones para reconocimiento de patrones [25].

### 3.1. Antecedentes de las redes neuronales

Cuando se narra la corta pero intensa historia de las redes neuronales también conocidas como modelos conexionistas se suele fijar el origen en los trabajos de McCulloch y Pitts [26], sin embargo, existen trabajos anteriores que abrieron el camino a estos investigadores. Entre estos trabajos podemos destacar el realizado por Kari Lashley [27] en los años 20. En su trabajo de 1950 se resume su investigación de 30 años; en su trabajo destaca que el proceso de aprendizaje es un proceso distribuido y no local a una determinada área del cerebro. Un estudiante de Lashley, D. Hebb [28] recoge el testigo de su maestro y determina una de las reglas de aprendizaje más usadas en la regla del conexionismo y que, lógicamente, se conoce con el nombre de aprendizaje hebbiano. Las contribuciones de este investigador aparecieron publicadas en su libro *The Organization of the Behavior*. Además propone que la conectividad

del cerebro cambia continuamente conforme un organismo aprende cosas nuevas, creándose asociaciones neuronales con estos cambios. En su postulado de aprendizaje, Hebb sigue lo sugerido por Ramón y Cajal [29]. Al afirmar que la efectividad de una sinapsis variable entre dos neuronas se incrementa por una repetida activación de una neurona sobre otra a través de esta sinapsis. Desde un punto de vista neurofisiológico la regla planteada por Hebb sería una regla variante-temporal, con un alto mecanismo interactivo que incrementa la eficacia sináptica como una función de la actividad pre y post sináptica. Desde un punto de vista conexionista la regla de Hebb es un tipo de aprendizaje no supervisado (no se necesita ningún "maestro") en el que las conexiones entre dos neuronas se incrementan si ambas se activan al mismo tiempo [30].

La siguiente gran contribución a considerar es el trabajo de McCulloch y Pitts [26]. En este trabajo, se fijan las características de trabajo de lo que, posteriormente, se va a conocer como neurona de McCulloch-Pitts. Este tipo de neurona es un dispositivo binario (salida 0 ó 1), tiene un umbral de funcionamiento por debajo del cual está inactiva y puede recibir entradas excitatorias o inhibitorias cuya acción es absoluta: si existe alguna de estas entradas la neurona permanece inactiva. El modo de trabajo es simple, si no existe ninguna entrada inhibitoria se determina la resultante de las entradas excitatorias y si ésta es mayor que el umbral, la salida es 1 y si no, la salida es 0. Se puede observar que, con un elemento tan simple como el que se acaba de definir, se pueden implementar un gran número de funciones lógicas mediante su combinación con elementos similares. Además, dado el estado de la neurofisiología en 1943, el modelo de McCulloch-Pitts se acercaba a lo conocido por esa época acerca de la actividad sináptica neuronal. Esta capacidad de modelizar funciones lógicas desató la euforia por estos elementos individuales [30].

En 1956, Rochester, Holland, Habt y Duda [31] presentan un trabajo en el que, por primera vez, se verifica mediante simulaciones una teoría neuronal basada en el postulado de Hebb [28]. Para realizar este trabajo eminentemente práctico, se tuvieron que hacer varias suposiciones que, inicialmente, no estaban en el trabajo de Hebb [28]. Por ejemplo se acotó el valor de las sinapsis que, en principio, podía crecer sin límite [30].

Otro gran genio matemático, John Von Neumann [32] se planteó ideas conexionistas: en una recopilación de sus trabajos posterior a su muerte sugiere como posible camino para mejorar los ordenadores, de los cuales se puede considerar como uno de los padres, el estudio del sistema nervioso central. En 1958 se producen las aportaciones de Selfridge y Rosenblatt [33]. Estas contribuciones plantean implementaciones físicas de sistemas conexionistas. En su trabajo Selfridge plantea el sistema conocido como *Fandemonium*. Este sistema consta de una serie de capas compuestas por lo que se conoce como "demonios". Cada una de las diferentes capas de este sistema se reparte las diferentes tareas a realizar [30].

Por su parte, Rosenblatt [34], quince años después del estudio de McCulloch-Pitts, presenta una nueva aproximación al problema de reconocimiento de patrones mediante la introducción del perceptrón. Rosenblatt, planteó un dispositivo que realizara tareas que le interesaran a los psicólogos. El hecho que fuera una máquina capaz de aprender la hacía irresistiblemente atractiva para los ingenieros [30].

En 1960 Widrow y Hoff [35] presentan su ADELPH. Estas siglas tienen una historia curiosa: cuando las redes neuronales estaban en su máximo apogeo eran el acrónimo de Adaptive



Linear Neuron"; cuando las cosas empezaron a ir mal para las redes neuronales pero este sistema se seguía usando por los buenos resultados obtenidos con él se cambió a "Adaptive Linear Element". El sistema planteado por Widrow estaba regido por un algoritmo de aprendizaje muy sencillo denominado "LMS" (Least Mean Square). Con este trabajo se propone un sistema adaptativo que puede aprender de forma más precisa y rápida que los perceptrones existentes. El trabajo de Widrow permitió el desarrollo de un área del procesado digital de señales (control de sistemas) que se conoce con el nombre de procesado (control) adaptativo [30].

Block [36], presenta en 1962 un trabajo que estudia los perceptrones más concretamente, presenta resultados sobre el perceptrón "MARK I" con 400 dispositivos receptores foto sensitivos dispuestos en una matriz 20 por 20 con un conjunto de 8 unidades de salida [30].

El trabajo de Minsky y Papert [37] titulado *Perceptrons* que paralizó durante 10 años el avance de este campo de la inteligencia artificial. Este trabajo, que fue escrito y expuesto brillantemente, puso de manifiesto las limitaciones de los perceptrones. Estas limitaciones hacían referencia a la clase de problemas que se podían resolver usando estos elementos. Minsky y Papert demostraron que un perceptrón sólo podía resolver problemas linealmente separables que, para desgracia de los conexionistas, son los menos. Además los autores expusieron, y por esto se les ha criticado, sus opiniones sobre las extensiones de los perceptrones (a sistemas multicapa); ellos plantearon su absoluta inutilidad práctica. También hay que tener en cuenta que, en el momento de la publicación de su trabajo, Minsky y Papert trabajaban en otro campo de la inteligencia artificial. Sin embargo, como se demostró más tarde, se equivocaron en sus conjeturas [30].

El trabajo de Minsky y Papert [34] supuso una paralización de los trabajos sobre temas conexionistas, sin embargo algunos investigadores continuaron trabajando [30].

Kohonen [35] y Anderson [39] proponen el mismo modelo de memoria asociativa de forma simultánea. A modo de demostración de los diferentes campos de conocimiento que engloban los sistemas conexionistas estos autores tienen una formación diferente (Kohonen es ingeniero eléctrico y Anderson es neurofisiólogo). En el modelo artificial planteado la neurona es un sistema lineal que usa como regla de aprendizaje la regla de Hebb modificada [30].

En 1980, Stephen Grossberg [40], uno de los autores más prolíficos en el campo de las redes neuronales, establece un nuevo principio de auto-organización desarrollando las redes neuronales conocidas como "ART" (Adaptive Resonance Theory). Grossberg ha planteado diferentes modelos neuronales que han prescurado una gran utilidad práctica (principalmente en el campo del reconocimiento de patrones) [30].

En 1982 J. Hopfield [41] publica un trabajo clave para el resurgimiento de las redes neuronales. Gran parte del impacto de este trabajo se debió a la fama de Hopfield como distinguido físico teórico. En él, desarrolla la idea del uso de una función de energía para comprender la dinámica de una red neuronal recurrente con uniones sinápticas simétricas. En este primer trabajo, Hopfield sólo permite salidas bipolares (0 ó 1). En un trabajo posterior amplía la función energía plurcada para estos sistemas permitiendo la salida continua de las neuronas. El principal uso de estas redes ha sido como memorias y como instrumento para resolver problemas de optimización como el problema del viajante [20].



En el mismo año de 1982 Kohonen [42], publica un importante artículo sobre mapas autoorganizativos que se ordenan de acuerdo a unas simples reglas. El aprendizaje que se da en el modelo planteado no necesita de un "maestro"; estamos ante un aprendizaje de tipo no supervisado. Al año siguiente, en el número especial sobre modelos neuronales de la revista especializada *IEEE Transactions on Systems, Man and Cybernetics*, aparecen dos trabajos de gran importancia en el desarrollo de las redes neuronales. Fukushima, Miyake e Ito [43] presentan una red neuronal, el *Neocognatron*, de tal forma que combinando ideas del campo de la fisiología, ingeniería y de la teoría neuronal crean un dispositivo que es capaz de ser aplicado con éxito en problemas de reconocimiento de patrones. Este trabajo, y de ahí lo de *Neo*, supone un perfeccionamiento de un modelo anterior presentado por los mismos autores y conocido como *Cognatron*. Este sistema fue probado con la tarea de identificar números escritos a mano [30].

El segundo trabajo, presentado por Barto, Sutton y Anderson [44] estudia el aprendizaje reforzado y su aplicación en control. En este trabajo se plantea este nuevo tipo de aprendizaje en el que, a diferencia de trabajos anteriores sobre modelos supervisados, no es necesario un conocimiento total del error cometido por la red; lo único que se necesita es conocer el signo del error [30].

En 1986 aparece un trabajo que, junto al de Hopfield [45], resucitará el interés por las redes neuronales. En este trabajo Rumelhart, Hinton y Williams [46], desarrollan el algoritmo de aprendizaje de retro propagación (*backpropagation*) para redes neuronales multicapa dando una serie de ejemplos en los que se muestra la potencia del método desarrollado. A partir de ese año, el número de trabajos sobre redes neuronales ha aumentado exponencialmente apareciendo un gran número de aportaciones tanto a los métodos de aprendizaje como a las arquitecturas y aplicaciones de las redes neuronales. Se podría destacar de entre todas estas aportaciones el trabajo de Broomhead y Lowe [47] y el de Poggio y Girosi [48] sobre el diseño de redes neuronales en capas usando *RBF* (*Radial Basis Functions*), el trabajo intensivo desarrollado sobre las máquinas de vectores soporte, el desarrollo de la unión entre elementos neuronales y difusos y, por último, los trabajos sobre neuronas de pulsos (*spike neurons*). Finalmente hay que hacer mención a uno de los "motores" en el desarrollo de las redes neuronales: la predicción en series temporales. Una generalización de las redes *TDNN* (orientadas especialmente para ser usadas con series temporales) la realizó Eric Wan [49]. En su trabajo los pesos sinápticos, conexiones sinápticas [30].



## 3.2. Neurona biológica

El cerebro consta de un gran número de elementos altamente interconectados llamados neuronas (aproximadamente  $10^{11}$  neuronas y  $10^4$  conexiones por neurona. Estas neuronas tienen tres componentes principales: las dendritas, el cuerpo de la célula o soma, y el axón. Las dendritas, son el árbol receptor de la red, son como fibras nerviosas que cargan de señales eléctricas el cuerpo de la célula. El cuerpo de la célula, realiza la suma de esas señales de entrada. El axón es una fibra larga que lleva la señal desde el cuerpo de la célula hacia otras neuronas. El punto de contacto entre un axón de una célula y una dendrita de otra célula es llamado sinapsis, la longitud de la sinapsis es determinada por la complejidad del proceso químico que estabiliza la función de la red neuronal [50].

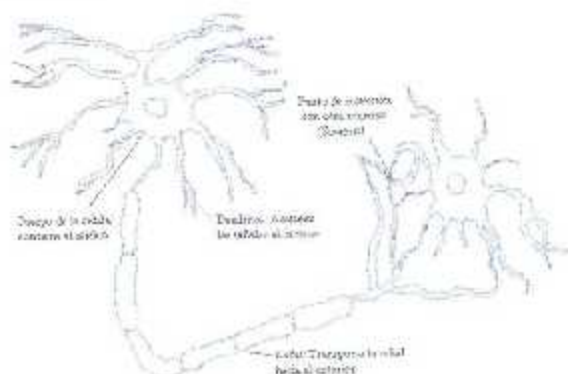


Figura 3.1: Neurona Biológica [51].

Algunas de las estructuras neuronales son determinadas en el nacimiento, otra parte es desarrollada a través del aprendizaje, proceso en que nuevas conexiones neuronales son realizadas y otras se pierden por completo. El desarrollo neurológico se hace crítico durante los primeros años de vida, por ejemplo está demostrado que si a un cachorro de gato, se le impide usar uno de sus ojos durante un periodo corto de tiempo, el nunca desarrollara una visión normal en ese ojo. Las estructuras neuronales continúan cambiando durante toda la vida, estos cambios consisten en el refuerzo o debilitamiento de las uniones sinápticas [50].

Como consecuencia de los primeros estudios sobre la base neural de los sistemas mnémicos (relacionados con la memoria), se creía que el almacenamiento de la memoria asociativa, tanto implícita como explícita, requerían de un circuito neuronal muy complejo. Entre quienes comenzaron a oponerse a este enfoque se hallaba Donald O. Hebb, quien sugirió que el aprendizaje asociativo podría ser producido por un mecanismo celular sencillo y propuso que las asociaciones podrían formarse por una actividad neuronal coincidente: "Cuando un axón de la célula A excita la célula B y participa en su activación, se produce algún proceso de desarrollo o cambio metabólico en una o en ambas células, de suerte que la eficacia de A, como célula excitadora de B, se intensifica". Según la regla Hebbiana de aprendizaje, el que coincida la actividad de las neuronas pre sinápticas (suministran el impulso de entrada) con la de las pos sinápticas (reciben el impulso) es muy importante para que se refuerce la conexión entre ellas, este mecanismo es llamado pre-postasociativo [50].

Todas las neuronas conducen la información de forma similar, esta viaja a lo largo de axones



en breves impulsos eléctricos, denominados potenciales de acción; los potenciales de acción que alcanzan una amplitud máxima de unos 100 mV y duran 1 ms, son resultado del desplazamiento a través de la membrana celular de iones de sodio dotados de carga positiva, que pasan desde el fluido extracelular hasta el citoplasma intracelular; la concentración extracelular de sodio supera enormemente la concentración intracelular [50].

La membrana en reposo mantiene un gradiente de potencial eléctrico de -70mv, el signo negativo se debe a que el citoplasma intracelular está cargado negativamente con respecto al exterior; los iones de sodio no atraviesan con facilidad la membrana en reposo, los estímulos físicos o químicos que reducen el gradiente de potencial, o que despojaricen la membrana, aumentan su permeabilidad al sodio y el flujo de este ion hacia el exterior acentúa la despolarización de la membrana, con lo que la permeabilidad al sodio se incrementa más aún [50].

Alcanzado un potencial crítico denominado "umbral", la realimentación positiva produce un efecto regenerativo que obliga al potencial de membrana a cambiar de signo. Es decir, el interior de la célula se torna positivo con respecto al exterior, al cabo de 1 ms, la permeabilidad del sodio decae y el potencial de membrana retorna a -70mv, su valor de reposo. Tras cada explosión de actividad iónica, el mecanismo de permeabilidad del sodio se mantiene refractario durante algunos milisegundos; la tasa de generación de potenciales de acción queda así limitada a unos 200 impulsos por segundo, o menos [50].

Los potenciales de acción, son señales de baja frecuencia conducidas en forma muy lenta, estos no pueden saltar de una célula a otra, la comunicación entre neuronas viene siempre mediada por transmisores químicos que son liberados en las sinapsis [50].

En la figura 3.2 se observa la comunicación entre neuronas y el proceso químico de la liberación de neurotransmisores.

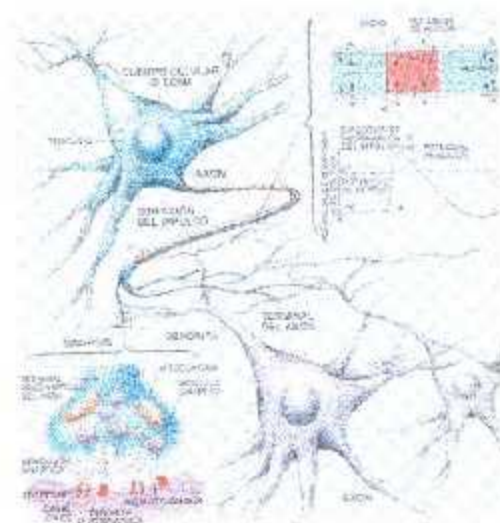


Figura 3.2: Comunicación entre neuronas y la liberación de neurotransmisores [50].

Cuando un potencial de acción llega al terminal de un axón son liberados transmi-

res alojados en diminutas vesículas, que después son vertidos en una hendidura de unos 20 nanómetros de anchura que separa la membrana pre sináptica de la pos sináptica; durante el apogeo del potencial de acción, penetran iones de calcio en el terminal nervioso, su movimiento constituye la señal determinante de la exocitosis sincronizada, esto es la liberación coordinada de moléculas neurotransmisoras. En cuanto son liberados, los neurotransmisores se enlazan con receptores pos sinápticos, causando el cambio de la permeabilidad de la membrana [50].

Cuando el desplazamiento de carga hace que la membrana se aproxime al umbral de generación de potenciales de acción, se produce un efecto excitador y cuando la membrana resulta estabilizada en la vecindad el valor de reposo se produce un efecto inhibitorio. Cada sinapsis produce sólo un pequeño efecto, para determinar la intensidad (frecuencia de los potenciales de acción) de la respuesta cada neurona ha de integrar continuamente hasta unas 1000 señales sinápticas, que se suman en el soma o cuerpo de la célula [50].

En algunas neuronas los impulsos se inician en la unión entre el axón y el soma, y luego se transmiten a lo largo del axón a otras células nerviosas. Cuando el axón está cerca de sus células destino, se divide en muchas ramificaciones que forman sinapsis con el soma o axones de otras células. Las sinapsis pueden ser excitatorias o inhibitorias según el neurotransmisor que se libere; cada neurona recibe de 10 000 a 100 000 sinapsis y su axón realiza una cantidad similar de sinapsis [50].

Los fenómenos que ocurren en la sinapsis son de naturaleza química, pero tienen efectos eléctricos laterales que se pueden medir [50].

### 3.3. Neurona artificial

La neurona artificial fue diseñada para emular las características del funcionamiento básico de la neurona biológica. En esencia, se aplica un conjunto de entradas a la neurona, cada una de las cuales representa una salida de otra neurona. Cada entrada se multiplica por su peso o ponderación correspondiente análoga al grado de conexión de la sinapsis. Todas las entradas ponderadas se suman y se determina el nivel de excitación o activación de la neurona. Una representación vectorial del funcionamiento básico de una neurona artificial se indica según la siguiente expresión de la ecuación [51].

$$NET = X * W \quad (3.1)$$

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

Normalmente la señal de salida NET suele ser procesada por una función de activación F para producir la señal de salida de la neurona. La función F puede ser una función lineal, o una función umbral o una función no lineal que simula con mayor exactitud las características de transferencia no lineales de las neuronas biológicas [51].



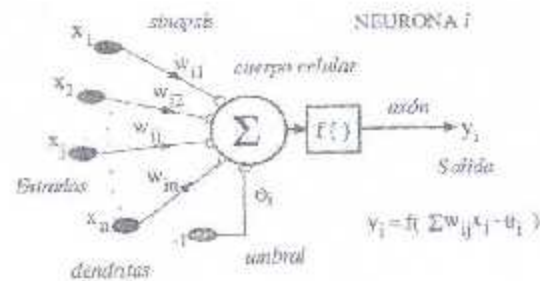


Figura 3.3: Neurona Artificial [30]

Las funciones  $F$  más utilizadas son la función Sigmoid y Tangente hiperbólica expresadas en la siguiente Tabla.

Este tipo de modelo de neurona artificial ignora muchas de las características de las neuro-

Sigmoid	$Salida = \frac{1}{1 + e^{-NET}}$
Tangente hiperbólica	<b>tanh</b> (NET)

Tabla 3.1: Funciones de Activación

nas biológicas. Entre ellas destaca la omisión de retardos y de sincronismo en la generación de la salida. No obstante, a pesar de estas limitaciones las redes construidas con este tipo de neurona artificial presentan cualidades y atributos con cierta similitud a la de los sistemas biológicos [51].

*Red Neuronal.* Una red neuronal es una estructura compuesta de un número de unidades interconectadas (neuronas artificiales). Cada unidad posee una característica de entrada/salida e implementa una computación local o función. La salida de cualquier unidad está determinada por su característica de entrada/salida, su interconexión con otras unidades y posiblemente de sus entradas externas. Sin embargo es posible un "trabajo a mano", la red desarrolla usualmente una funcionalidad general a través de una o más formas de entrenamiento [52].

### 3.3.1. Ventajas de las redes neuronales

Las Redes Neuronales tienen propiedades que las hacen especialmente atractivas para ser usadas en una gran cantidad de problemas prácticos:

#### 3.3.1.1. Son sistemas distribuidos no lineales

Una neurona es un elemento no lineal por lo que una interconexión de ellas (red neuronal) también será un dispositivo no lineal. Esta propiedad permitirá la simulación de sistemas no lineales y caóticos, simulación que con los sistemas clásicos lineales, no se puede realizar [30].





### 3.3.1.2. Aprendizaje Adaptativo

La capacidad de aprendizaje adaptativo es una de las características más atractivas de redes neuronales. Esto es, aprenden a llevar a cabo ciertas tareas mediante un entrenamiento con ejemplos ilustrativos.

### 3.3.1.3. Son sistemas tolerantes a fallos

Una red neuronal, al ser un sistema distribuido, permite el fallo de algunos elementos individuales (neuronas) sin alterar significativamente la respuesta total del sistema. Este hecho las hace especialmente atractivas frente a los computadores actuales que, por lo general, son sistemas secuenciales de tal forma que un fallo en uno de sus componentes conlleva que el sistema total no funcione [30].

### 3.3.1.4. Auto organización

Una RNA crea su propia representación de la información en su interior, quitándole esta tarea al usuario.

### 3.3.1.5. Adaptabilidad

Una red neuronal tiene la capacidad de modificar los parámetros de los que depende su funcionamiento de acuerdo con los cambios que se produzcan en su entorno de trabajo (cambios en las entradas, presencia de ruido, etc.). Con respecto a la capacidad de adaptación hay que tener en cuenta que ésta no puede ser tampoco excesivamente grande ya que conduciría a tener un sistema inestable respondiendo a pequeñas perturbaciones. Este es el problema conocido como el dilema plasticidad-estabilidad [30].

### 3.3.1.6. Flexibilidad

Una RNA puede manejar cambios no importantes en la información de entrada, como señales con ruido u otros cambios en la entrada (por ejemplo si la información de entrada es la imagen de un objeto, la respuesta correspondiente no sufre cambios si la imagen cambia un poco su brillo o el objeto cambia ligeramente).

### 3.3.1.7. Establecen relaciones no lineales entre datos

Las redes neuronales son capaces de relacionar dos conjuntos de datos mediante relaciones complejas [30].

### 3.3.1.8. Posibilidad de implementación

Esta posibilidad permite que estos sistemas puedan ser aplicados en sistemas de tiempo real, simulando sistemas biológicos mediante elementos de silicio [30].



Todas estas ventajas hacen el uso de las redes neuronales especialmente atractivo en un gran número de aplicaciones.

### 3.3.2. Niveles o capas de una red neuronal

La distribución de neuronas dentro de la red se realiza formando niveles o capas, con un número determinado de dichas neuronas en cada una de ellas. A partir de su situación dentro de la red, se pueden distinguir tres tipos de capas:

*De entrada:* es la capa que recibe directamente la información proveniente de las fuentes externas de la red [52].

*Ocultas:* son internas a la red y no tienen contacto directo con el entorno exterior. El número de niveles ocultos puede estar entre cero y un número elevado. Las neuronas de las capas ocultas pueden estar interconectadas de distintas maneras, lo que determina, junto con su número, las distintas topologías de redes neuronales [52].

*De salidas:* transfieren información de la red hacia el exterior [52].

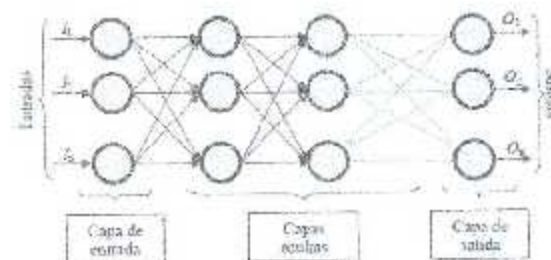


Figura 3.4: Ejemplos de una red totalmente conectada [30]

En la figura 3.4 se muestra la estructura de una red multicapa, en la que cada nodo o neurona únicamente está conectada con neuronas de un nivel superior. Notar que hay más conexiones que neuronas en  $o_j$ ; en este sentido, se dice que una red es totalmente conectada si todas las salidas desde un nivel llegan a todos y cada uno de los nodos del nivel siguiente.

#### 3.3.2.1. Topología de las redes neuronales

La topología o arquitectura de una red neuronal consiste en la organización y disposición de las neuronas en la misma, formando capas o agrupaciones de neuronas más o menos alejadas de la entrada y salida de dicha red. En este sentido, los parámetros fundamentales de la red son: el número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de conexiones entre neuronas [52].

*Redes monocapa:* En las redes monocapa, se establecen conexiones entre las neuronas que pertenecen a la única capa que constituye la red. Las redes monocapas se utilizan generalmente en tareas relacionadas con lo que se conoce como autoasociación (regenerar información de



entrada que se presenta a la red de forma incompleta o distorsionada) [52].

*Redes multicapas:* Las redes multicapas son aquellas que disponen de un conjunto de neuronas agrupadas en varios (2, 3, etc.) niveles o capas. En estos casos, una forma para distinguir la capa a la que pertenece una neurona, consistiría en fijarse en el origen de las señales que recibe a la entrada y el destino de la señal de salida. Normalmente, todas las neuronas de una capa reciben señales de entrada desde otra capa anterior (la cual está más cerca a la entrada de la red), y envían señales de salida a una capa posterior (que está más cerca a la salida de la red). A estas conexiones se las denomina conexiones hacia adelante o "feedforward" [52].

Sin embargo, en un gran número de estas redes también existe la posibilidad de conectar la salida de las neuronas de capas posteriores a la entrada de capas anteriores; a estas conexiones se las denomina conexiones hacia atrás o "feedback" [52].

Estas dos posibilidades permiten distinguir entre dos tipos de redes con múltiples capas: las redes con conexiones hacia adelante o redes "feedforward", y las redes que disponen de conexiones tanto hacia adelante como hacia atrás o redes "feedforward/feedback" [52].

*Red Recurrente:* este tipo de red se diferencia de las anteriores en la existencia de lazos de realimentación en la red. Estos lazos pueden ser entre neuronas de diferentes capas, neuronas de la misma capa o, entre una misma neurona. Esta estructura la hace especialmente adecuada para estudiar la dinámica de los sistemas no lineales [52].

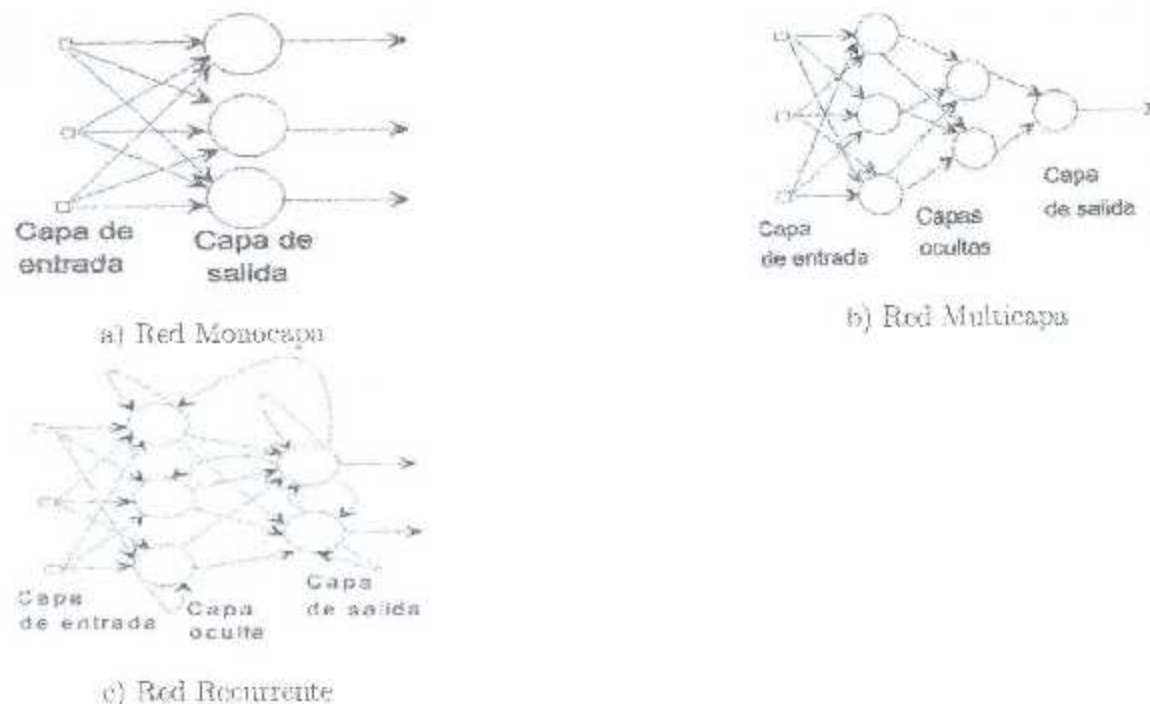


Figura 3.5: Tipos de Redes Neuronales [30].

*Conexión entre neuronas:* La conectividad entre los nodos de una red neuronal está relacionada con la forma en que las salidas de las neuronas están canalizadas para convertirse en entradas de otras neuronas. La señal de salida de un nodo puede ser una entrada de otro elemento de proceso, o incluso ser una entrada de sí mismo (conexión autorecurrente). Cuando ninguna salida de las neuronas es entrada de neuronas del mismo nivel o de niveles precedentes, la red se describe como de conexión hacia delante. Cuando las salidas pueden ser conectadas como entradas de neuronas de niveles previos o del mismo nivel, incluyéndose ellas mismas, la red es de conexión hacia atrás. Las redes de propagación hacia atrás que tienen lazos cerrados son llamadas: *sistemas recurrentes* [52].

### 3.4. Tipos de entrenamiento de las Redes Neuronales

Una de las principales características de las Redes Neuronales es su capacidad de aprendizaje. El entrenamiento de las Redes Neuronales muestra algunos paralelismos con el desarrollo intelectual de los seres humanos. No obstante aun cuando parece que se ha conseguido entender el proceso de aprendizaje conviene ser moderado porque el aprendizaje de las Redes Neuronales está limitado [51].

El objetivo del entrenamiento de una Redes Neuronales es conseguir que una aplicación determinada, para un conjunto de entradas produzca el conjunto de salidas deseadas o mínimamente consistentes. El proceso de entrenamiento consiste en la aplicación secuencial de diferentes conjuntos o vectores de entrada para que se ajusten los pesos de las interconexiones según un procedimiento predeterminado. Durante la sesión de entrenamiento los pesos convergen gradualmente hacia los valores que hacen que cada entrada produzca el vector de salida deseado [51].

Los algoritmos de entrenamiento o los procedimientos de ajuste de los valores de las conexiones de las Redes Neuronales se pueden clasificar en dos grupos: Supervisado y No Supervisado [51].

#### 3.4.1. Entrenamiento supervisado

Estos algoritmos requieren el 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 reajustar la red y cambiar los pesos de acuerdo con un algoritmo que tiende a maximizar el error [51].

Las parejas de vectores del conjunto de entrenamiento se aplican secuencialmente y de forma cíclica. Se calcula el error y el ajuste de los pesos por cada pareja hasta que el error para el conjunto de entrenamiento cetero sea un valor pequeño y aceptable [51].

En este tipo de aprendizaje se suelen considerar, a su vez, tres formas de llevarlo a cabo, que dan lugar a los siguientes aprendizajes supervisados:

- 1) Aprendizaje por corrección de error:



BIBLIOTECA — CENTRO DE  
GRADUADOS E INVESTIGACION  
I. T. L.

- 2) Aprendizaje por refuerzo.
- 3) Aprendizaje estocástico.

### 3.4.1.1. Aprendizaje por corrección de error

Consiste en ajustar los pesos de las conexiones de la red en función de la diferencia entre los valores deseados y los obtenidos a la salida de la red, es decir, en función del error cometido en la salida [52].

Un ejemplo de este tipo de algoritmos lo constituye la *regla de aprendizaje del Perceptron*, utilizada en el entrenamiento de la red del mismo nombre que desarrolló Rosenblatt en 1958 [52]. Esta es una regla muy simple, para cada neurona en la capa de salida se le calcula la desviación a la salida objetivo como el error,  $\delta$ . El cual luego se utiliza para cambiar los pesos sobre la conexión de la neurona precedente. El cambio de los pesos por medio de la regla de aprendizaje del Perceptron se realiza según la siguiente regla:

$$\Delta W_{ij} = \sigma^{\text{out}_j}(a_{qi} - \text{out}_i); \quad (3.2)$$

donde:  $a_{qi}$  es la salida deseada/objetivo de la neurona de salida  $N_i$ ,  $\delta_i = (a_{qi} - \text{out}_i)$  la desviación objetivo de la neurona  $N_i$  y  $\sigma$  el aprendizaje.

La salida de la neurona  $N_j(\text{out}_j)$  se utiliza, porque este valor influye en la entrada global y, por ende, en la activación y luego en la salida de la neurona  $N_i$ . Esto es semejante a un "efecto en cadena".

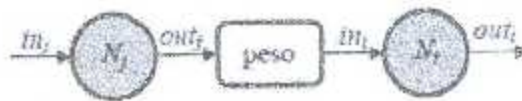


Figura 3.6: Influencia de la salida de la neurona  $N_j$  en la entrada de la neurona  $N_i$  [51].

Otro algoritmo muy conocido y que pertenece a esta clasificación es la *regla de aprendizaje Delta* o regla del mínimo error cuadrado (LMS Error Least Mean Squared Error), que también utiliza la desviación a la salida objetivo, pero toma en consideración a todas las neuronas predecessoras que tiene la neurona de salida. Esto permite cuantificar el error global cometido en cualquier momento durante el proceso de entrenamiento de la red, lo cual es importante, ya que cuanto más información se tenga sobre el error cometido, más rápido se puede aprender. Luego el error calculado ( $\delta$ ) es igualmente repartido entre las conexiones de las neuronas predecessoras [52].

Por último se debe mencionar la *regla de aprendizaje de propagación hacia atrás* o de *back-propagation*, también conocido como regla LMS multicapa, la cual es una generalización de la regla de aprendizaje Delta. Esta es la primer regla de aprendizaje que permitió realizar cambios sobre los pesos en las conexiones de la capa oculta [52].



### 3.4.1.2. Aprendizaje por esfuerzo

Se trata de un aprendizaje supervisado, más lento que el anterior, que se basa en la idea de no disponer de un ejemplo completo del comportamiento deseado, es decir, de no indicar durante el entrenamiento exactamente la salida que se desea que proporcione la red ante una determinada entrada [52].

En el aprendizaje por refuerzo la función del supervisor se reduce a indicar mediante una señal de refuerzo si la salida obtenida en la red se ajusta a la deseada (éxito = +1 o fracaso = -1), y en función de ello se ajustan los pesos basándose en un mecanismo de probabilidades. Se podría decir que en este tipo de aprendizaje la función del supervisor se asemeja más a la de un crítico (que opina sobre la respuesta de la red) que a la de un maestro (que indica a la red la respuesta concreta que debe generar), como ocurriría en el caso de supervisión por corrección del error [52].

### 3.4.1.3. Aprendizaje estocástico

Consiste básicamente en realizar cambios aleatorios en los valores de los pesos de las conexiones de la red y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad.

En el aprendizaje estocástico se suele hacer una analogía en términos termodinámicos, asociando a la red neuronal con un sólido físico que tiene cierto estado energético. En el caso de la red, la energía de la misma representaría el grado de estabilidad de la red, de tal forma que el estado de mínima energía correspondería a una situación en la que los pesos de las conexiones consiguen que su funcionamiento sea el que más se ajusta al objetivo deseado [52].

Según lo anterior, el aprendizaje consistiría en realizar un cambio aleatorio de los valores de los pesos y determinar la energía de la red (habitualmente la función energía es una función de Lyapunov). Si la energía es menor después del cambio, es decir, si el comportamiento de la red se acerca al deseado, se acepta el cambio; si, por el contrario, la energía no es menor, se aceptaría el cambio en función de una determinada y preestablecida distribución de probabilidades [52].

## 3.4.2. Entrenamiento no supervisado

Los sistemas neuronales con entrenamiento supervisado han tenido éxito en muchas aplicaciones y sin embargo tienen muchas críticas debido a que desde el punto de vista biológico no son muy lógicos. Resulta difícil creer que existe un mecanismo en el cerebro que compare las salidas deseadas con las salidas reales [51].

Los sistemas no supervisados son modelos de aprendizaje más lógicos en los sistemas biológicos. Desarrollados por Kohonen (1984) y otros investigadores, estos sistemas de aprendizaje no supervisado no requieren de un vector de salidas deseadas y por tanto no se realizan comparaciones entre las salidas reales y salidas esperadas. El conjunto de vectores de entrenamiento consiste únicamente en vectores de entrada. El algoritmo de entrenamiento modifica los pesos de la red de 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 [51].

Estas redes deben encontrar las características, regularidades, correlaciones o categorías que se puedan establecer entre los datos que se presentan en su entrada. Existen varias posibilidades en cuanto a la interpretación de la salida de estas redes, que dependan de su estructura y del algoritmo de aprendizaje empleado [52].

En algunos casos, la salida representa el grado de familiaridad o similitud entre la información que se le está presentando en la entrada y las informaciones que se le han mostrado hasta entonces (en el pasado). En otro caso, podría realizar una clusterización (clustering) o establecimiento de categorías, indicando la red a la salida a qué categoría pertenece la información presentada a la entrada, siendo la propia red quien debe encontrar las categorías apropiadas a partir de las correlaciones entre las informaciones presentadas [52].

En cuanto a los algoritmos de aprendizaje no supervisado, en general se suelen considerar dos tipos, que dan lugar a los siguientes aprendizajes:

- 1).- Aprendizaje Hebbiano
- 2).- Aprendizaje comparativo y competitivo

#### 3.4.2.1. Aprendizaje Hebbiano

Esta regla de aprendizaje es la base de muchas otras, la cual pretende medir la familiaridad o extraer características de los datos de entrada. El fundamento es una suposición bastante simple: si dos neuronas  $N_i$  y  $N_j$  toman el mismo estado simultáneamente (ambas activas o ambas inactivas), el peso de la conexión entre ambas se incrementa. Las entradas y salidas permitidas a la neurona son: -1, 1 o 0, 1 (neuronas binarias). Esto puede explicarse porque la regla de aprendizaje de Hebb se originó a partir de la neurona biológica clásica, que solamente puede tener dos estados: activa o inactiva [52].

#### 3.4.2.2. Aprendizaje competitivo y comparativo

Se orienta a la clusterización o clasificación de los datos de entrada. Como característica principal del aprendizaje competitivo se puede decir que, si un patrón nuevo se determina que pertenece a una clase reconocida previamente, entonces la inclusión de este nuevo patrón a esta clase matizará la representación de la misma. Si el patrón de entrada se determinó que no pertenece a ninguna de las clases reconocidas anteriormente, entonces la estructura y los pesos de la red neuronal serán ajustados para reconocer la nueva clase [52].





### 3.5. Perceptrón

La arquitectura del Perceptrón, llamada mapeo de patrones (pattern-mapping), aprende a clasificar modelos mediante un aprendizaje supervisado. Los modelos que clasifica suelen ser generalmente vectores con valores binarios (0,1) y las categorías de la clasificación se expresan mediante vectores binarios [51].

El Perceptrón presenta dos capas de unidades procesadoras (PE) y sólo una de ellas presenta la capacidad de adaptar o modificar los pesos de las conexiones. La arquitectura del Perceptrón admite capas adicionales pero éstas no disponen la capacidad de modificar sus propias conexiones [51].

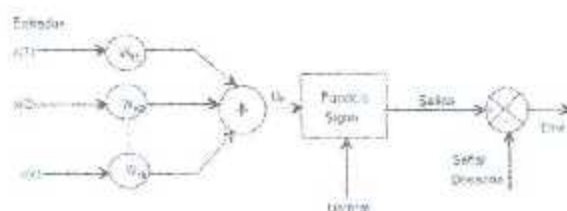


Figura 3.7: Esquema básico del Perceptrón [51]

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

$$y = \sum_{k=1}^n W_k * X_k$$

Un aspecto común en muchas de las redes neuronales es la entrada especial llamada "bias" representada en la parte superior izquierda de la figura (entrada  $x(1)$ ). Esta entrada siempre presenta un valor fijo, +1 y funciona como una masa en un circuito eléctrico donde no varía de valor (se puede utilizar como un valor constante de referencia) [51].

El Perceptrón comprueba si la suma de las entradas ponderadas es mayor o menor que un cierto valor umbral y genera la salida "U<sub>n</sub>" según la ecuación siguiente [51].

$$\text{si } y > 0 \text{ entonces } U_n = 1 \tag{3.3}$$

$$\text{si } y < 0 \text{ entonces } U_n = 0 \tag{3.4}$$

La salida  $U_n$  es transmitida a lo largo de la línea de salida y constituye uno de los componentes del vector de salida de la red [51].

Las redes Perceptrón de dos capas, tienen una capa de entrada y una capa de unidades procesadoras que constituyen la capa de salida [51].



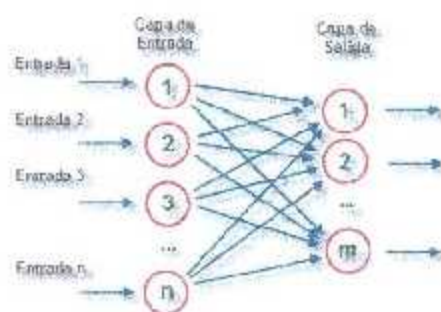


Figura 3.8: Red perceptrón de dos capas.

A lo largo de los años 50 y 60 se desarrollaron muchos tipos de topologías de redes basadas en la arquitectura del Perceptrón. Las topologías con tres o más capas se caracterizan porque la regla de aprendizaje del perceptrón sólo adapta los pesos o valores de una capa [51].

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 [51].

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 [51].

La adaptación de los pesos se puede realizar mediante diferentes reglas. Una de las reglas más simples de aprendizaje del Perceptrón se indica en la siguiente ecuación [51].

$$W_{nk(nuevo)} = W_{nk(antigo)} + C(t_j - y_j) X_k \quad (3.5)$$

Siendo  $t_j$  el valor de la salida deseada,  $y_j$  el valor de salida producida por la unidad procesadora,  $X_k$  el valor de la entrada y  $C$  el coeficiente de aprendizaje [51].

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 convergido. 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 [51].

### 3.5.1. Separación lineal

El mayor inconveniente del Perceptrón, a pesar del éxito que ha tenido en muchas aplicaciones de clasificación de patrones es la imposibilidad de adaptar los pesos de todas las capas. En los años en los que se realizó el Perceptrón, los investigadores no fueron capaces de diseñar un algoritmo que propagara las correcciones de los pesos a través de redes multicapa [51].



La principal limitación funcional del Perceptrón es que una unidad de salida sólo puede clasificar patrones linealmente separables, es decir, las clases de patrones que pueden separarse en dos clases mediante una línea. Este concepto se puede extender a tres o más dimensiones simplemente separando dos clases mediante planos e hiperplanos [51].

En este punto Minsky y Papert centraron las críticas al Perceptrón en su publicación *Perceptrons* (1969). El libro incluía opiniones negativas sobre la posibilidad de extender el Perceptrón en una herramienta útil en la computación neuronal; por ejemplo para pequeños problemas de clasificación de patrones como el OR exclusivo, el Perceptrón es incapaz de resolverlo con éxito [51].

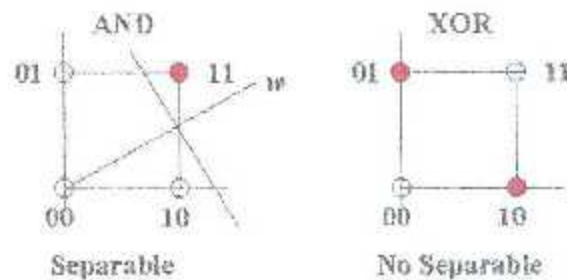


Figura 3.9: Ejemplo de separabilidad [51]

Afortunadamente para la computación neuronal surgieron nuevas reglas de aprendizaje para redes multicapa y nuevas arquitecturas, entre ellas la más popular *Backpropagation*, que resolvieron entre otros los problemas de clasificación de patrones no separables linealmente [51].

### 3.6. Red backpropagation

La invención del algoritmo *Backpropagation* ha desempeñado un papel vital en el resurgimiento del interés de las redes neuronales artificiales. *Backpropagation* es un método de entrenamiento de redes multicapa. Su potencia reside en su capacidad de entrenar capas ocultas y de este modo supera las posibilidades restringidas de las redes de una única capa [51].

El concepto básico de *Backpropagation* fue presentado en 1974 por Paul Werbos e independientemente reinventado por David Parker en 1982, y también presentado en 1986 por Rumelhart, Hinton y Williams. La duplicidad de esfuerzos y trabajos es frecuente en cualquier disciplina, y más en el mundo de las redes neuronales debido a su naturaleza interdisciplinaria [51].

La regla de aprendizaje del Perceptrón de Rosenblatt y el algoritmo LMS de Widrow y Hoff fueron diseñados para entrenar redes de una sola capa. Estas redes tienen la desventaja que solo pueden resolver problemas linealmente separables, fue esto lo que llevó al surgimiento de las redes multicapa para sobrepasar esta dificultad en las redes hasta entonces conocidas.



Uno de los grandes avances logrados con la Backpropagation es que esta red aprovecha la naturaleza paralela de las redes neuronales para reducir el tiempo requerido por un procesador secuencial para determinar la correspondencia entre unos patrones dados. Además el tiempo de desarrollo de cualquier sistema que se este tratando de analizar se puede reducir como consecuencia de que la red puede aprender el algoritmo correcto sin que alguien tenga que deducir por anticipado el algoritmo en cuestión [50].

La mayoría de los sistemas actuales de compute se han diseñado para llevar a cabo funciones matemáticas y lógicas a una velocidad que resulta asombrosamente alta para el ser humano. Sin embargo la destreza matemática no es lo que se necesita para solucionar problemas de reconocimiento de patrones en entornos ruidosos, característica que incluso dentro de un espacio de entrada relativamente pequeño, puede llegar a consumir mucho tiempo. El problema es la naturaleza secuencial del propio computador; el ciclo tomar – ejecutar de la naturaleza Von Neumann solo permite que la máquina realice una operación a la vez. En la mayoría de los casos, el tiempo que necesita la máquina para llevar a cabo cada instrucción es tan breve (típicamente una milonésimas de segundo) que el tiempo necesario para un programa, así sea muy grande, es insignificante para los usuarios. Sin embargo, para aquellas aplicaciones que deban explorar un gran espacio de entrada o que intentan correlacionar todas las permutaciones posibles de un conjunto de patrones muy complejo, el tiempo de computación necesario se hace bastante grande [50].

Lo que se necesita es un nuevo sistema de procesamiento que sea capaz de examinar todos los patrones en paralelo. Idealmente ese sistema no tendría que ser programado explícitamente, lo que haría es adaptarse a sí mismo para aprender la relación entre un conjunto de patrones dado como ejemplo y ser capaz de aplicar la misma relación a nuevos patrones de entrada. Este sistema debe estar en capacidad de concentrarse en las características de una entrada arbitraria que se asemeje a otros patrones vistos previamente, sin que ninguna señal de ruido lo afecte. Este sistema fue el gran aporte de la red de propagación inversa, Backpropagation [50].

La Backpropagation es un tipo de red de aprendizaje supervisado, que emplea un ciclo propagación – adaptación de dos fases. Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, esto se propaga desde la primera capa a través de las capas superiores de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas [50].

Las salidas de error se propagan hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida. Sin embargo las neuronas de la capa oculta sólo reciben una fracción de la señal total del error, basándose aproximadamente en la contribución relativa que haya aportado cada neurona a la salida original. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total. Basándose en la señal de error percibida, se actualizan los pesos de conexión de cada neurona, para hacer que la red converja hacia un estado que permita clasificar correctamente todos los patrones de entrenamiento [50].

La importancia de este proceso consiste en que, a medida que se entrena la red, las neuronas de las capas intermedias se organizan a sí mismas de tal modo que las distintas neuronas



aprenden a reconocer distintas características del espacio total de entrada. Después del entrenamiento, cuando se les presente un patrón arbitrario de entrada que contenga ruido o que esté incompleto, las neuronas de la capa oculta de la red responderán con una salida activa si la nueva entrada contiene un patrón que se asemeje a aquella característica que las neuronas individuales hayan aprendido a reconocer durante su entrenamiento. Y a la inversa, las unidades de las capas ocultas tienen una tendencia a inhibir su salida si el patrón de entrada no contiene la característica para reconocer, para la cual han sido entrenadas [50].

Varias investigaciones han demostrado que, durante el proceso de entrenamiento, la red Backpropagation tiende a desarrollar relaciones internas entre neuronas con el fin de organizar los datos de entrenamiento en clases. Esta tendencia se puede extrapolar, para llegar a la hipótesis consistente en que todas las unidades de la capa oculta de una Backpropagation son asociadas de alguna manera a características específicas del patrón de entrada como consecuencia del entrenamiento. Lo que sea o no exactamente la asociación puede no resultar evidente para el observador humano, lo importante es que la red ha encontrado una representación interna que le permite generar las salidas deseadas cuando se le dan las entradas, en el proceso de entrenamiento. Esta misma representación interna se puede aplicar a entradas que la red no haya visto antes, y la red clasificará estas entradas según las características que compartan con los ejemplos de entrenamiento [50].

### 3.3.1. Arquitectura de la red Backpropagation

La unidad procesadora se caracteriza por realizar una suma ponderada de las entradas llamada  $U_{in}$ , presentar una salida  $y_n$  y tener un valor  $\delta_j$  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  $W_{ij}$ , y es modificado durante el proceso de aprendizaje [51].

Normalmente, la Backpropagation utiliza tres o más capas de unidades procesadoras. La capa inferior es la capa de entrada, y se caracteriza por ser la única capa cuyas unidades procesadoras reciben entradas desde el exterior. Sirven como puntos distribuidores, no realizan ninguna operación de cálculo. Las unidades procesadoras de las demás capas procesan las señales. La siguiente capa superior es la capa oculta, y todas sus 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 de la red [51].

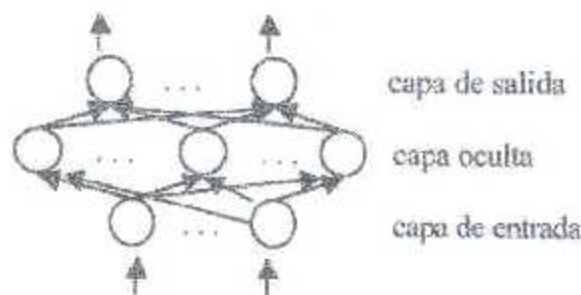


Figura 3.10: Estructura Backpropagation de tres capas [51].

Las redes Backpropagation tienen un método de entrenamiento supervisado. A la red se

le presenta parejas de patrones, un patrón de entrada emparejado con un patrón de salida deseada. Por cada presentación los pesos son ajustados de forma que disminuya el error entre la salida deseada y la respuesta de la red [51].

El algoritmo de aprendizaje backpropagation conlleva una fase de propagación hacia adelante y otra fase de propagación hacia atrás. Ambas fases se realizan por cada patrón presentado en la sesión de entrenamiento [51].

### 3.6.2. Propagación hacia adelante

Esta fase de propagación hacia adelante se inicia cuando se presenta un patrón en la capa de entrada de la red. Cada unidad de la entrada se corresponde con un elemento del vector patrón de entrada. Las 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. A continuación las demás capas realizarán la fase de propagación hacia adelante que determina el nivel de activación de las otras capas [51].

La unidad procesadora  $j$  obtiene la cantidad  $U_n$  [51].

$$U_n = \sum_i X_i \cdot W_{ij} \quad (3.6)$$

y genera la salida o nivel de activación.

$$y = f(U_n) \quad (3.7)$$

La función  $f$  es una función umbral genérica, entre las que cabe destacar la función Sigmoid y la función Hiperbólica [51].

El valor de la salida de la unidad  $j$  es enviado o transmitido a lo largo de todas las conexiones de salida de dicha unidad [51].

Conviene indicar que las unidades procesadoras de la capa de entrada no realizan ninguna operación de cálculo con sus entradas, ni operaciones con funciones umbrales, sólo asumen su salida como el valor del correspondiente elemento del vector de entrada [51].

Por otro lado, algunas redes backpropagation utilizan unidades llamadas bias como parte de cualquiera de las capas ocultas y de la capa de salida. Estas unidades presentan constantemente un nivel de activación de valor 1. Además esta unidad está conectada a todas las unidades de la capa inmediatamente superior y los pesos asociados a dichas conexiones son ajustables en el proceso de entrenamiento. La utilización de esta unidad tiene un doble objetivo, mejorar las propiedades de convergencia de la red y ofrecer un nuevo efecto umbral sobre la unidad que opera [51].

### 3.6.3. Propagación hacia atrás

Una vez se ha completado la fase de propagación hacia adelante se inicia la fase de corrección o fase de propagación hacia atrás. Los cálculos de las modificaciones de todos los pesos de las conexiones empiezan por la capa de salida y continúa hacia atrás a través de



BIBLIOTECA - CENTRO DE  
GRADUADOS E INVESTIGACION  
A. T. L.

todas las capas de la red hasta la capa de entrada. Dentro de los tipos de ajuste de pesos se puede clasificar dos grupos, ajuste de unidades procesadoras de la capa de salida y ajuste de unidades procesadoras de las capas ocultas [51].

*Ajuste de Pesos de la Capa de Salida:* el ajuste de estos pesos es relativamente sencillo debido a 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 el valor deseado especificado en el patrón del conjunto de entrenamiento [51].

A partir del resultado de la comparación se calcula un valor de error  $\delta_j$ , para cada unidad de la capa de salida [51].

$$\delta_j = (t_j - X_j) f'(U_n) \quad (3.8)$$

siendo  $t_j$  el valor de salida deseado y  $f$  la derivada de la función umbral  $f$  [51].

La Figura siguiente muestra el cálculo de los valores  $\delta_j$  de las unidades de la capa de salida.

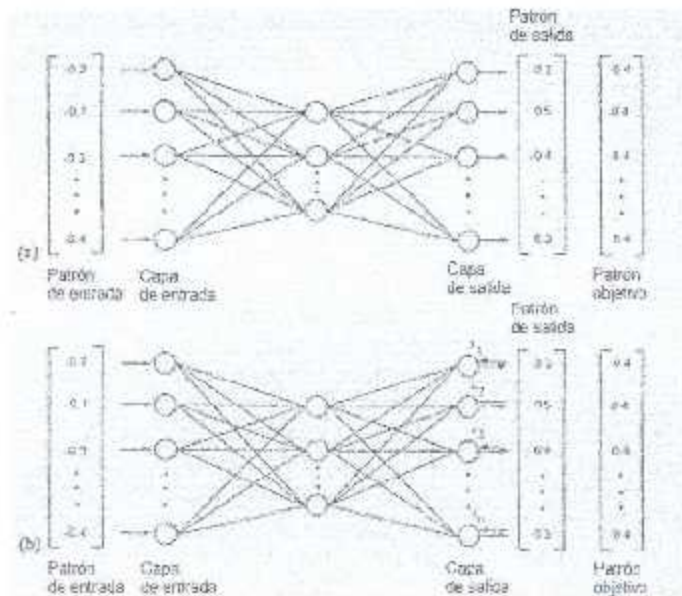


Figura 3.11: Cálculo de los valores  $\delta_j$  de la capa de salida [51]

*Ajuste de Pesos de las Capas Ocultas:* estas capas no tienen un vector de salidas deseadas y por tanto no se puede seguir el método de propagación de error mencionado en el caso de unidades procesadoras de la capa de salida [51].

$$\delta_j = [\sum_k \delta_k W_{kj}] f'(U_n) \quad (3.9)$$

El ajuste de los pesos asociados a las conexiones se realiza a partir del valor  $\delta_j$  de cada unidad de proceso. Cada peso es ajustado según la regla generalizada (Rumelhart y McClelland 1986) [51].

$$\Delta W_{ij} = \eta \delta_j X_{ij} \quad (3.10)$$





La variable  $\eta$  es el coeficiente de aprendizaje. Este coeficiente, normalmente entre 0.25 y 0.75 refleja el grado de aprendizaje de la red. Algunos veces este coeficiente es modificado de un valor inicial alto a valores progresivamente menores durante la sesión del entrenamiento con el objetivo de lograr un mejor aprendizaje [51].

*Convergencia:* en el proceso de entrenamiento o aprendizaje de la Backpropagation es frecuente medir cuantitativamente el aprendizaje mediante el valor RMS (*Root Mean Square*) del error de la red. Esta medida refleja el modo en el que la red está logrando respuestas correctas; a medida que la red aprende, su valor RMS decrece [51].

Debido a que los valores de salida de la red y los valores de salidas deseadas son valores reales, es necesario definir un parámetro de corte o un valor umbral del valor RMS del error de la red que permita decir que la red se aproxima a la salida deseada y considerar que la respuesta es correcta [51].

La convergencia es un proceso en el que el valor RMS del error de la red tiende cada vez más al valor 0. La convergencia no siempre es fácil de conseguirla porque a veces el proceso puede requerir un tiempo excesivo o bien porque la red alcanza un mínimo local y deja de aprender [51].

### 3.6.4. Algoritmo de Levenberg - Marquardt

Este algoritmo es una modificación del método de Newton, el que fue diseñado para minimizar funciones que sean la suma de los cuadrados de otras funciones no lineales: es por ello que el algoritmo de Levenberg - Marquardt, tiene un excelente desempeño en el entrenamiento de redes neuronales donde el rendimiento de la red está determinado por el error medio cuadrático [50].

El método de Newton para optimizar el rendimiento  $c(x)$  es:

$$X_{k+1} = X_k - A_k^{-1}g_k \quad (3.11)$$

$$A_k = \nabla^2 c(x)|_{x=x_k} \quad (3.12)$$

$$g_k = \nabla c(x)|_{x=x_k} \quad (3.13)$$

Si asumimos que  $c(x)$  es una suma de funciones cuadráticas

$$c(x) = \sum_{i=1}^n v_i^2(x) = v^T(x)v(x) \quad (3.14)$$

El gradiente puede ser escrito entonces en forma matricial:

$$\nabla c(x) = 2J^T(x)v(x) \quad (3.15)$$

Donde  $J(x)$  es la matriz jacobiana.

Ajustando el método de Newton, obtenemos el algoritmo de Levenberg Marquardt [50].

$$x_{k+1} = x_k - [J^T(x_k)J(x_k) + \mu_k I]^{-1} J^T(x_k)v(x_k) \quad (3.16)$$



o determinando directamente el incremento.

$$\Delta x_k = - [J^T(x_k)J(x_k) + \mu_k I]^{-1} J^T(x_k)e(x_k) \quad (3.17)$$

La nueva constante  $\mu$  determina la tendencia del algoritmo, cuando  $\mu_k$  se incrementa, este algoritmo se aproxima al algoritmo de pasos descendientes para tasas de aprendizaje muy pequeñas; cuando  $\mu_k$  se decrecienta este algoritmo se convierte en el método de Gauss - Newton [50].

El algoritmo comienza con un valor pequeño para  $\mu_k$ , por lo general 0.01, si en ese paso no se alcanza el valor para  $e(x)$  entonces el paso es repetido con un multiplicado por un factor  $\theta > 1$ . Si se ha escogido un valor pequeño de paso en la dirección de paso descendiente,  $e(x)$  debería disminuir. Si un paso produce un pequeño valor para  $e(x)$ , entonces el algoritmo tiende al método de Gauss - Newton, el que se supone garantiza una rápida convergencia. Este algoritmo genera un compromiso entre la velocidad del método de Gauss-Newton y la garantía de convergencia del método de paso descendiente [50].

Los elementos de la matriz Jacobiana necesarios en el algoritmo de Levenberg-Marquardt son de este estilo:

$$[J]_{h,i} = \frac{\partial e_{k,g}}{\partial x_i} \quad (3.18)$$

Donde  $x$  es el vector de parámetros de la red, que tiene la siguiente forma:

$$x^T = [x_1, x_2, \dots, x_n] = [W_{1,1}^1, W_{1,2}^1, \dots, W_{S^1,R}^1, b_1^1, \dots, b_{S^1}^1] \quad (3.19)$$

Para utilizar este algoritmo en las aplicaciones para redes multicapa, se redefinirá el término sensibilidad de forma que sea más simple hallarlo en cada iteración [50].

$$S_{i,h}^m = \frac{\partial e_{k,g}}{\partial n_{i,h}^m} \quad (3.20)$$

Donde

$$h = (q - 1)S^M + k \quad (3.21)$$

Para utilizar este algoritmo en las aplicaciones para redes multicapa, se redefinirá el término sensibilidad de forma que sea más simple hallarlo en cada iteración [50].

$$[J]_{h,i} = \frac{\partial e_{k,g}}{\partial w_{i,j}^m} = \frac{\partial e_{k,g}}{\partial n_{i,g}^m} * \frac{\partial n_{i,g}^m}{\partial w_{i,j}^m} = S_{i,h}^m * \frac{\partial n_{i,g}^m}{\partial w_{i,j}^m} = S_{i,h}^m * a_{j,i}^{m-1} \quad (3.22)$$

o para las ganancias

$$[J]_{h,i} = \frac{\partial e_{k,g}}{\partial b_i^m} = \frac{\partial e_{k,g}}{\partial n_{i,g}^m} * \frac{\partial n_{i,g}^m}{\partial b_i^m} = S_{i,h}^m * \frac{\partial n_{i,g}^m}{\partial b_i^m} = S_{i,h}^m \quad (3.23)$$

De esta forma, cuando la entrada  $pQ$  ha sido aplicada a la red y su correspondiente salida  $a^M Q$  ha sido computada, el algoritmo Backpropagation de Levenberg-Marquardt es inicializado con [50]:

$$S_q^{iM} = -f^M(n_i^M) \quad (3.24)$$



Cada columna de la matriz  $S_Q^M$  debe ser propagada inversamente a través de la red para producir una fila de la matriz Jacobiana. Las columnas pueden también ser propagadas conjuntamente de la siguiente manera [50]:

$$S_q^m = f^n(n_q^m)(W^{m+1})^T S_q^{m+1} \quad (3.25)$$

Las matrices sensibilidad total para cada capa en el algoritmo de Levenberg-Marquardt son formadas por la extensión de las matrices computadas para cada entrada [50]:

$$S^m = [S_1^m][S_2^m] \dots [S_Q^m] \quad (3.26)$$

Para cada nueva entrada que es presentada a la red, los vectores de sensibilidad son propagados hacia atrás, esto se debe a que se ha calculado cada error en forma individual, en lugar de derivar la suma al cuadrado de los errores. Para cada entrada aplicada a la red habrá  $S^M$  errores, uno por cada elemento de salida de la red y por cada error se generara una fila de la matriz Jacobiana [50].

Este algoritmo puede resumirse de la siguiente manera:

- Se presentan todas las entradas a la red, se calculan las correspondientes salidas y cada uno de los errores según [50].

$$e_q = t_q - n_q^M \quad (3.27)$$

se calcula después, la suma de los errores cuadrados para cada entrada  $e(x)$ .

- Se calculan las sensibilidades individuales y la matriz sensibilidad total y con estas, se calculan los elementos de la matriz Jacobiana [50].

- Se obtiene

$$\Delta x_k \quad (3.28)$$

- Se recalcula la suma de los errores cuadrados usando  $x_k + \Delta x_k$ . Si esta nueva suma es más pequeña que el valor calculado en el paso 1 entonces se divide  $\mu$  por  $\vartheta$ , se calcula  $x_{k+1} = x_k + \Delta x_k$  y se regresa al paso 1. Si la suma no se reduce entonces se multiplica  $\mu$  por  $\vartheta$  y se regresa al paso 3 [50].

El algoritmo debe alcanzar convergencia cuando la norma del gradiente de [50].

$$\nabla e(x) = 2J^T(x)n(x) \quad (3.29)$$

sea menor que algún valor predeterminado, o cuando la suma de los errores cuadrados ha sido reducida a un error que se haya trazado como meta [50].

El comportamiento de este algoritmo se visualiza en la figura 3.12, la cual muestra la trayectoria de convergencia con  $\mu = 0.01$  y  $\vartheta = 5$  [50].



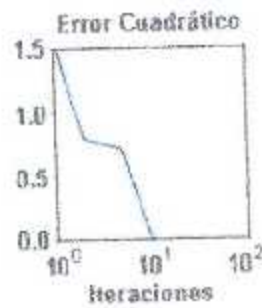


Figura 3.12: Trayectoria del algoritmo Levenberg-Marquardt [50].

Este algoritmo converge en menos iteraciones que cualquier otro método, por supuesto requiere mucho más cálculo iterativo, debido a que implica el cálculo de matrices inversas. A pesar de su gran esfuerzo computacional sigue siendo el algoritmo de entrenamiento más rápido para redes neuronales cuando se trabaja con un moderado número de parámetros en la red, si el número de parámetros es muy grande utilizarlo resulta poco práctico [50].

### 3.6.5. Ventajas e inconvenientes

La principal ventaja de la Red Backpropagation es su capacidad genérica de mapeo de patrones. La red es capaz de aprender una gran variedad de relaciones de mapeo de patrones. No requiere un conocimiento matemático de la función que relaciona los patrones de la entrada y los patrones de salida. La Backpropagation sólo necesita ejemplos de mapeo para aprender. La flexibilidad de esta red es aumentada con la posibilidad de elegir número de capas, interconexiones, unidades procesadoras, constante de aprendizaje y representación de datos. Como resultado de estas características la red Backpropagation es capaz de participar con éxito en una amplia gama de aplicaciones [50].

El mayor inconveniente es el tiempo de convergencia. Las aplicaciones reales pueden llegar a tener miles de ejemplos en el conjunto de entrenamiento y ello requiere días de tiempo de cálculo. Además la backpropagation es susceptible de fallar en el entrenamiento, es decir, la red puede que nunca llegue a converger [51].

Existen una variedad de técnicas desarrolladas para disminuir el tiempo de convergencia y evitar los mínimos locales. El término de "momentum" se utiliza para aumentar la velocidad del proceso de convergencia. Otra forma de mejorar la convergencia se basa en la variación del parámetro de aprendizaje  $\eta$  comenzando con valores altos y adquiriendo progresivamente valores más pequeños. Entre las técnicas utilizadas para evitar los mínimos locales destacan cambiar la red, cambiar el conjunto de entrenamiento y añadir ruido aleatorio a los pesos [51].



---

---

## CAPÍTULO 4

---

### DESARROLLO DEL PROTOTIPO DE NARIZ ELECTRÓNICA

En este capítulo se presenta el desarrollo del prototipo de Nariz Electrónica. Se explica cada una de sus partes, empezando por los sensores, el sistema de adquisición o captura de datos, la interfaz gráfica y programación.

#### 4.1. Sensores MQ

Los sensores utilizados en el prototipo son los siguientes [53]:

Modelo	Gas Objetivo
MQ-2	Gas combustible general
MQ-3	Alcohol
MQ-4	Gas natural, Metano
MQ-6	LPG, Propano
MQ-7	Monóxido de carbono
MQ-8	Hidrógeno
MQ-9	CO, Gas combustible

Tabla 4.1: Sensores utilizados

Los sensores utilizados en el prototipo son los sensores semiconductores de Oxido Metálico de la serie MQ Hanwei, estos sensores tienen una respuesta a varios gases sin embargo tienen una mayor sensibilidad al gas objetivo mostrado en la tabla 4.1



Figura 4.1: Sensores serie MQ [53].

En la figura 4.2 se presenta como está conformado el sensor de la serie MQ [53].

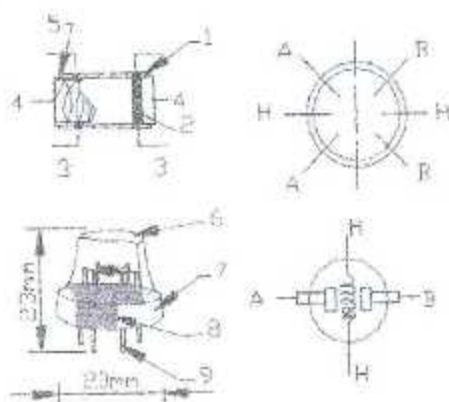


Figura 4.2: Partes del sensor MQ [53]

En la tabla 4.2 se describen las partes que integran el sensor.

	Partes	Materiales
1	Capa sensible al gas	$SnO_2$
2	Electrodo	Al
3	Línea de electrocos	Pt
4	Bobina de calefacción	Ni-Cr alloy



5	Cerámicas tubular	$AL_2O_3$
6	Red anti explosión	Gasas de acero inoxidable
7	Anillo de sujeción	Cobrizado Ni
8	Base de resina	Bakelite
9	Pin del tubo	Cobrizado Ni

Tabla 4.2: Partes del sensor

#### 4.1.1. Respuesta de los sensores MQ

Los sensores semiconductores de óxido metálico varían su resistencia ante la presencia del gas al cual son sensibles, de acuerdo a esto se puede obtener la respuesta de los sensores y determinar su sensibilidad [53].

La respuesta de los sensores a los diferentes gases se muestra en una gráfica de la hoja de datos como en la mostrada en la figura 4.3

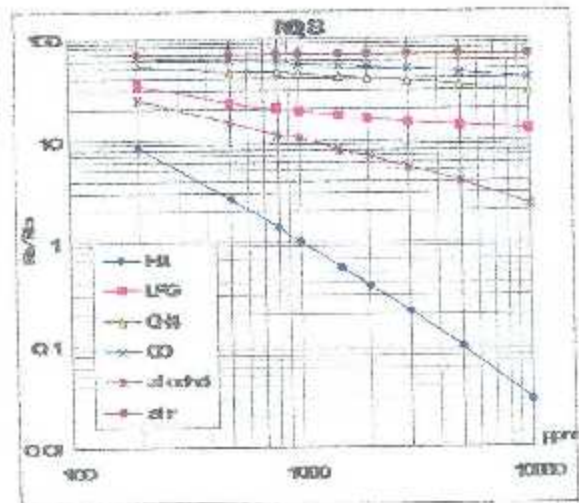


Figura 4.3: Característica de sensibilidad el sensor MQ-8 [53]

En la gráfica mostrada en la figura 4.3 podemos observar que la respuesta con mayor sensibilidad mostrada por el sensor es ante la presencia del Hidrógeno ( $H_2$ ), sin embargo también tiene respuesta ante la presencia de otros gases como LPG y Alcohol cuya respuesta de sensibilidad es mejor.

La sensibilidad se determina de acuerdo a la presencia del gas en concentración de partes



por millón (ppm) y la respuesta del sensor mediante la fórmula:

$$R_s/R_0 \quad (4.1)$$

Donde  $R_0$  es la resistencia del sensor a 1000 ppm de aire limpio y  $R_s$  es la resistencia del sensor ante la concentración de algún tipo de gas, como se puede observar en la gráfica se utiliza la resistencia del sensor ante aire limpio como referencia para determinar la sensibilidad [53].

#### 4.1.2. Circuito básico

En la figura 4.4 se muestra el circuito básico para la implementación de los sensores serie MQ.

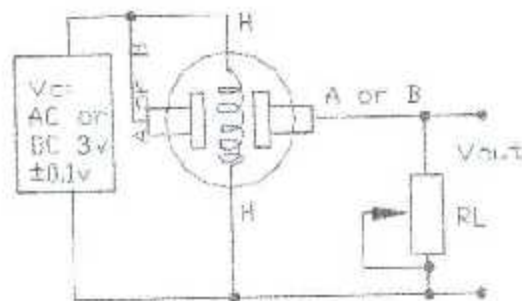


Figura 4.4: Circuito básico [53].

En la tabla se presentan las condiciones de trabajo utilizadas.

Símbolo	Nombre del parámetro	Condición técnica	Etiqueta
$V_C$	Voltaje del circuito	$3v \pm 0.1$	AC o DC
$V_H$	Voltaje de calefacción	$3v \pm 0.1$	AC o DC
$P_L$	Resistencia de carga	$2.2k\Omega$	
$R_H$	Resistencia de calentamiento	$31 \pm 5\%$	
$P_H$	Consumo de calor	menos de 800 mW	

Tabla 4.3: Condiciones de trabajo para el sensor serie MQ



Como ya se explicó previamente los sensores de la serie MQ varían su resistencia ante la presencia del gas al cual son sensibles, esta sensibilidad varía de acuerdo al gas presentado y por lo tanto la variación de resistencia es diferente a cada gas, para obtener la respuesta del sensor es necesario conectarlo a un circuito el cual es mostrado en la figura 4.4.

Este circuito funciona como un divisor de tensión entre la resistencia del sensor y  $R_L$ , obteniendo un voltaje a la salida "Vout", este voltaje es el que utilizaremos para observar la respuesta del sensor, durante la realización de las pruebas a los sensores de determino que el voltaje para alimentar el sensor fuera de 3 vdc en lugar de los 5 vdc recomendado, debido a que con 5 vdc había un calentamiento excesivo de los sensores.

En el circuito mostrado es de gran importancia la resistencia  $R_L$  debido a que esta determina en gran medida el voltaje que observaremos a la salida del circuito, en la práctica se eligió una  $R_L$  de 2.2 K $\Omega$  debido a que la variación de voltaje a la salida fue mayor por lo que esta respuesta nos ayuda bastante al capturar los datos de los gases.

En el caso de los sensores MQ-7 y MQ-9 se tiene que alimentar  $V_H$  con un voltaje de 1.5 vdc, el circuito es mostrado en la figura 4.5.

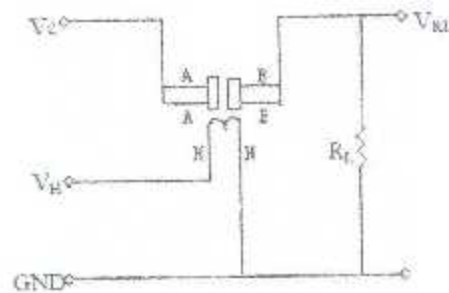


Figura 4.5: Circuito para MQ-7 y MQ 9 [53]

Los siete sensores fueron montados en una tablilla electrónica con dos reguladores de voltaje para obtener los 3 vdc y 1.5 vdc que se necesitan para alimentarlos.

## 4.2. Captura de los datos

### 4.2.1. Arduino Mega 2560

Para capturar los datos obtenidos por los sensores, introducirlos a la computadora y posteriormente procesarlos es necesario una tarjeta adquirenta para esto se utiliza la tarjeta Arduino Mega 2560 mostrada en la figura 4.6







Figura 4.6: Tarjeta Arduino Mega 2560 [54]

Microprocesador	voltaje de entrada	voltaje del sistema	PWM	E/A Entradas	Entradas Analógicas	Memoria
ATmega2560	7-12v	5v	14	54	16	256Kb

Tabla 4.4: Datos de la Arduino Mega 2560

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar [54].

Arduino puede tomar información del entorno a través de sus pines de entrada y puede afectar a aquello que le rodea controlando luces, motores y otros actuadores con los pines de salida. El micro controlador en la placa Arduino se programa mediante el lenguaje de programación Arduino y el entorno de desarrollo Arduino. Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar el micro a un ordenador, pero tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software [54].

## 4.2.2. Pines de la tarjeta arduino

### 4.2.2.1. Pines Digitales

Cada uno de los 54 pines digitales pueden utilizarse como entradas o salidas usando las funciones *pinMode()*, *digitalWrite()*, y *digitalRead()*. Operan a 5 voltios. Cada pin puede proporcionar e recibir una intensidad máxima de 40mA y tiene una resistencia interna (desconectada por defecto) de 20-50kOhms [54].

### 4.2.2.2. Pines puerto serie

Los pines del conexionado serie son: Serie 0: 0 (RX) y 1 (TX); Serie 1: 19 (RX) y 18 (TX); Serie 2: 17 (RX) y 16 (TX); Serie 3: 15 (RX) y 14 (TX). Usado para recibir (RX) y transmitir (TX) datos a través de puerto serie TTL. Los pines Serie 0 (RX) y 1 (TX) están conectados a los pines correspondientes del chip FTDI USB-a-TTL, es decir, que están conectados a la conexión a través de USB con el ordenador [54].



### 4.2.2.3. Pines Analógicos

El Arduino Mega tiene 16 entradas analógicas, y cada una de ellas proporciona una resolución de 10bits (1024 valores) utiliza las funciones *analogRead()*, *analogWrite()*. Por defecto se mide de tierra a 5 voltios, aunque es posible cambiar la cota superior de este rango usando el pin AREF y la función *analogReference()*. El nombre de los pines es alfanumérico, todos van precedidos de 'A' [54].

Estos pines proporcionan una salida PWM (*Pulse Wave Modulation*, modulación de onda por pulsos) de 8 bits de resolución (valores de 0 a 255) a través de la función *analogWrite()* [54].

Para el caso de nuestro prototipo se utilizan las entradas Analógicas (7 entradas) para recibir los datos provenientes de los sensores, como ya se menciono tienen una resolución de 10 bits (1024 valores) para un valor de entrada de 0 a 5 vdc.

### 4.2.3. Enlace Arduino-Matlab

Para utilizar la Interfaz gráfica GUI de Matlab y poder programar la tarjeta Arduino desde la propia programación de la interfaz se utilizo la librería "arduino" para Matlab y se descarga el archivo "adiosrv.pde" a la tarjeta mediante el software Arduino, esto nos permite la comunicación directa entre Matlab y la tarjeta Arduino, creando un objeto para la tarjeta y utilizándolo para la comunicación con el puerto serie.

Los pasos para realizar el Enlace Arduino-Matlab son los siguientes:

- 1.- Tener instalado el Matlab (versión 2010a como mínimo), y descargar los archivos mencionados en las preguntas frecuentes.
- 2.- Abrir la carpeta "arduino-1.0.4-windows" y luego el "ARDUINO" (software de programación para la tarjeta).
- 3.- Conectar la placa ARDUINO a través del cable USB.
- 4.- Verificar por cual "COM" (puerto) quedo conectado.  
Nos dirigimos a Inicio>Equipo "Click derecho" >Administrar>Administrador de dispositivos>Puertos>Mega 2560, al lado derecho de "Arduino Mega 2560" sale el COMX, siendo X un nuevo entre 0 y 6 usualmente.
- 5.- Dentro del ARDUINO ir a Herramientas>Tarjeta y elegir la "placa ARDUINO MEGA 2560"
- 6.- De nuevo ir a Herramientas>Puerto Serial y seleccionar nuestro puerto COMX, que ubicamos en el paso 4.
- 7.- Vamos a introducir un sketch dentro de nuestra placa ARDUINO. Esta sketch nos permitira hacer la conexión entre el ARDUINO Y MATLAB, se le instala este sketch a la placa para que pueda entender las ordenes enviadas desde el MATLAB.  
Dentro del Arduino: Archivo>Abrir, Buscamos dentro de los archivos descargados en la siguiente dirección: Arduino\O>pde>adiosrv>adiosrv.pde
- 8.- Teniendo el adiosrv abierto le damos a "cargar" y esperamos hasta que diga "Carga terminada."
- 9.- Cerramos el "ARDUINO" para dejar de tener el puerto ocupado, así poder establecer



ahora la conexión a través del MATLAB.

10.- Se abre como administrador el MATLAB, MATLAB > Click derecho > Ejecutar como administrador.

11.- El directorio "Current Folder" lo ubicamos dentro de la carpeta Arduino\0.

12.- Procedemos a realizar la instalación adecuada insertando los comandos.

- Escribimos `install_arduino`

- Ahora escribimos `h=arduino('COMX')`. Donde X es el número hallado en el paso 4. Deben aparecer muchas instrucciones. Dichas instrucciones encendiéndolas orientaran al usuario en la programación.

Ahora se puede programar la tarjeta arduino desde Matlab utilizando los comandos de programación de arduino solo que anteponiendo el nombre del objeto arduino y un punto antes del comando esto indica precisamente que es un comando de programación de arduino.

Por ejemplo:

$$a.pinMode(13,'output'); \quad (4.2)$$

$$a.digitalWrite(13,1); \quad (4.3)$$

$$a.digitalWrite(13,0); \quad (4.4)$$

La ecuación 4.2 configura el Pin 13 digital como salida, la ecuación 4.3 pone en alto la salida 13 y la ecuación 4.4 pone en bajo la salida 13.

#### 4.2.4. Modulo Xbee

Para realizar la comunicación inalámbrica entre la tarjeta Arduino donde se reciben los datos de los sensores y la computadora donde se tiene el software para la interfaz gráfica y la programación con la que se realizara el reconocimiento de patrones, se utilizan dos módulos Xbee de 1mW con antena Digi WRL-08665 series 1 [55].



Figura 4.7: Modulo Xbee [55].

##### 4.2.4.1. Especificaciones

- Estándar IEEE 802.15.4/ ZigBee
- Frecuencia inalámbrica: 2.4GHz-2.4835GHz



- Alcance: Interior 30m y Exterior: 100m
- Tasa de datos por radiofrecuencia (RF data rate): 250 kbps
- Sensibilidad del receptor -92dBm
- Series 1

ZigBee es un protocolo de comunicación inalámbrico basado en el estándar de comunicaciones para redes inalámbricas IEEE 802.15.4. Creado por ZigBee Alliance, una organización teóricamente sin ánimo de lucro, de más de 200 empresas (destacan Mitsubishi, Honeywell, Philips, entre otras), muchas de ellas fabricantes de semiconductores. ZigBee permite que dispositivos electrónicos de bajo consumo puedan realizar sus comunicaciones inalámbricas. Especialmente redes de sensores en entornos industriales, médicos, etc [55].

Las comunicaciones Zigbee se realizan en la banda libre de 2.4GHz. A diferencia de bluetooth, este protocolo no utiliza FHSS (Frequency hopping), sino que realiza las comunicaciones a través de una única frecuencia, es decir, de un canal. Normalmente puede escogerse un canal de entre 16 posibles. El alcance depende de la potencia de transmisión del dispositivo así como también del tipo de antenas utilizadas (cerámicas, dipolos, etc). El alcance normal con antena dipolo (Digi) en línea de vista es de aproximadamente (tomando como ejemplo el caso de la versión de 1mW de potencia) de 100m y en interiores de unos 30m. La velocidad de transmisión de datos de una red Zigbee es de hasta 256kbps. Una red Zigbee la pueden formar, teóricamente, hasta 65535 equipos, es decir, el protocolo está preparado para poder controlar en la misma red esta cantidad enorme de dispositivos [55].

#### 4.2.4.2. Circuito básico

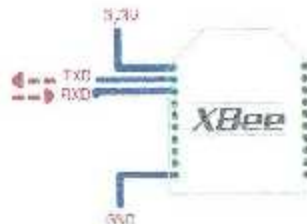


Figura 4.3: Circuito básico Xbee [55].

El módulo requiere una alimentación desde 2.8 a 3.4 V, la conexión a tierra y las líneas de transmisión de datos por medio del UART (TXD y RXD) para comunicarse con un micro controlador, o directamente a un puerto serial utilizando algún conversor adecuado para los niveles de voltaje [55].

#### 4.2.4.3. Configuración del módulo Xbee

X-CTU es una aplicación basada en Windows que proporciona Digi Internacional (fabricante de los módulos Xbee). Este programa está diseñado para interactuar con los archivos de firmware que se encuentran en los productos Digi RF y proporcionar una fácil manera de usar la interfaz gráfica de usuario [55].

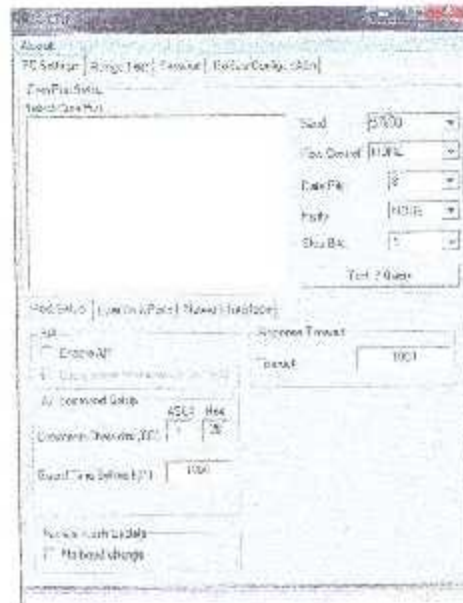


Figura 4.9: Aplicación XCTU

Para la configuración de los módulos Xbee se selecciona en la pestaña "PC Settings" del X-CTU el puerto donde está conectado el módulo Xbee, la velocidad de comunicación Baud y se presiona el botón "Test/Query", para confirmar la comunicación con el módulo.

Posteriormente en la pestaña "Modem Configuration" se presiona el botón "Read" para leer y detectar el módulo Xbee, una vez detectado aparece la ventana con todos los parámetros del módulo con los cuales podemos realizar las modificaciones.

Punto a punto es la conexión ideal para reemplazar comunicación aerial por un cable. Sólo se debe configurar la dirección. Para ello se utilizan los comandos MY y el DL. La idea, es que se defina arbitrariamente una dirección para un módulo, usando el comando MY, el cual se va a comunicar con otro que tiene la dirección DL, también definida arbitrariamente. Con esto cada módulo define su dirección con MY, y escribe la dirección del módulo al cual se desea conectar usando DL [55].

Para realizar una comunicación punto a punto se utilizan los siguientes parámetros:

- **PAN ID.**- Representa el código de la red que pretendemos mostrar
- **DH y DL.**- forman la dirección de destino (DH parte alta y DL parte baja) del módulo (01 para el emisor y 00 para el receptor).
- **MY.**- Representa la dirección con la que se comunica el módulo (el emisor se comunica





con el elemento 0 que es el receptor y el receptor se comunica con el elemento 1 que es el emisor).

	Xbee-Envía	Xbee-Rocebe
PAN ID	9032	9032
DH	0	0
DL	1	0
MY	0	1

Tabla 4.5: Parámetros para la comunicación punto-punto

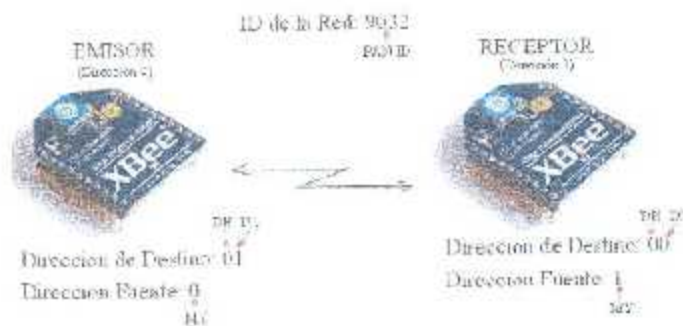


Figura 4.10: Comunicación punto a punto de Módulos Xbee [55]

#### 4.2.4.4. Dispositivos para la conexión de los módulos Xbee

Para montar el módulo Xbee emisor en la tarjeta Arduino se utiliza la Arduino Xbee Shield, donde solo se necesita configurarla con el "jumper" en la posición Xbee para que el ordenador se comunique mediante el módulo Xbee, mientras que si el "jumper" se pone en la posición USB el ordenador se comunicara con la Arduino mediante la comunicación USB.



Figura 4.11: Arduino Xbee Shield [52].

Se utiliza el Explorador Xbee Sparkfun para conectar el módulo Xbee receptor a la computadora mediante una USB y poder recibir los datos del módulo Xbee emisor.



Figura 4.12: Explorador Xbee Sparkfun [55]

### 4.3. Interfaz gráfica

Se utiliza el software Matlab para realizar la interfaz gráfica para y la programación de reconocimiento de patrones mediante redes neuronales, debido a las herramientas que ofrece con el "Neural Networks toolbox", su entorno de programación GUIDE o GUI como comúnmente se le conoce.

Matlab es un lenguaje de alto desempeño diseñado para realizar cálculos técnicos, integra el cálculo, la visualización y la programación en un ambiente fácil de utilizar donde los problemas y las soluciones se expresan en una notación matemática, es un sistema interactivo cuyo elemento básico de datos es el arreglo que no requiere de dimensionamiento previo. Esto permite resolver muchos problemas computacionales, específicamente aquellos que involucren vectores y matrices, en un tiempo mucho menor al requerido para escribir un programa en un lenguaje escalar no interactivo tal como C o Fortran [56].

Una de las cosas más importantes de Matlab es que los "toolbox" le permiten aprender y aplicar la teoría. Los "toolbox" son grupos de comandos de Matlab (archivos M) que extienden el ambiente de Matlab para resolver problemas de áreas específicas de la ciencia e ingeniería. Por ejemplo, existen "toolbox" para las áreas de Procesamiento Digital de Señales, Sistemas de Control, Redes Neuronales, Lógica Difusa, Wavelets, etc. [56].

Matlab Guide es un entorno de programación visual que ofrece Matlab para poder realizar y ejecutar programas de Simulación a medida de forma simple que necesiten ingreso continuo de datos, tiene las características básicas de todos los programas visuales como Visual Basic o Visual C++.

#### 4.3.1. Realización de la interfaz

Se realiza una interfaz gráfica donde se puede visualizar la entrada de voltajes o señales a la tarjeta adquisitoria Arduino provenientes de los sensores mediante gráficas y datos numéricos en tiempo real, además estos datos son capturados utilizando una opción llamada "Captura" la cual capturará los datos de los sensores en un tiempo determinado por el usuario.



estos datos son guardados en archivos de Excel en forma de vectores para utilizarlos posteriormente.

Una vez que se tienen los datos se tiene la opción de "Entrenamiento de la Red Neuronal", donde se tienen las opciones de seleccionar los datos de las pruebas realizadas con los gases, mandándolos llamar de los archivos Excel donde se guardaron, y entrenar la red con estos datos para que puedan ser identificados o diferenciados.

Se tiene otra opción llamada "Reconocimiento" donde al presionar el botón "Identificar" el programa vuelve a leer los datos que los sensores están recibiendo en ese momento y son muestreados por la red neuronal la cual realiza una simulación y determina cual gas es el presentado a los sensores o si no lo identifica con los datos entrenados.

Se tiene una opción identificada con el nombre de "Muestras" donde se elige cada cuantas muestras leídas se realiza la simulación para identificar el gas que se está presentando en la matriz de sensores.

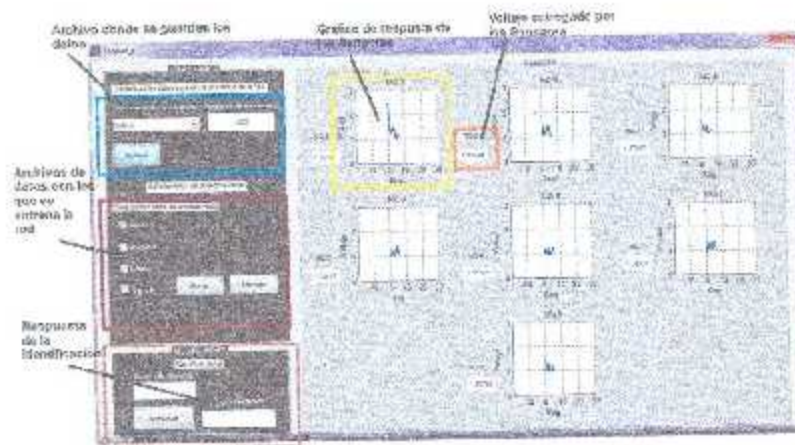


Figura 4.13: Pantalla de la Interfaz Gráfica

### 4.3.2. Programación

A continuación se realizara una breve explicación de la programación de las funciones de la Interfaz.

Primera mente se crea el objeto *a=arduino* en "opening function" para poder enlazar la tarjeta Arduino con Matlab mediante la librería previamente instalada indicando el puerto que se va a leer, también se utiliza el comando "handles" para poder almacenar datos en una variable.

```
function Capture_OpeningFcn(hObject, eventdata, handles, varargin)

handles.a=arduino('COM4');
handles.output = hObject;

% Update handles structure
```



```
guidata(handles, handles);
```

### 4.3.2.1. Captura

En la opción de "Captura" se grafican los datos recibidos y se presentan en la pantalla numéricamente para esto se crean los "Axes" los cuales son una opción de matlab para poner imágenes y gráficas.

Para crear la gráfica en el "Axes" se utilizan las siguientes líneas de código donde se le dan las características a la gráfica y la variable con la que se va a graficar "I1" y "I2".

```
%Preparacion de la figura %%%%%%%%%%%%%%%
axes(handles.axes1)
cla;
title('MQ-3');
xlabel('Seg. ');
ylabel('Voltaje');
l1 = line(nan,nan,'Color','b','LineWidth',2);
grid on;
hold on;

axes(handles.axes2)
cla;
title('MQ-2');
xlabel('Seg. ');
ylabel('Voltaje');
l2 = line(nan,nan,'Color','b','LineWidth',2);
grid on;
hold on;
```

Después se leen los datos de la tarjeta Arduino con el comando "handles.a.analogRead()" se presentan numéricamente en la pantalla con "set()" en el "Edit text" correspondiente y se dibujan en la gráfica los datos nuevamente utilizando "set" para la variable con que se va a graficar ("I1" o "I2" en estos ejemplos).

```
t=toc;
% lee del puerto serie
y(i)=(handles.a.analogRead(15)*5)/1024;
f(i)=(handles.a.analogRead(14)*5)/1024;

set(handles.edit4,'String',(handles.a.analogRead(15)*5)/1024);
set(handles.edit5,'String',(handles.a.analogRead(14)*5)/1024);

% dibujar en la figura
x = linspace(0,t/rate,i);
axes(handles.axes1)
ylim([0 3])
xlim([t-25 t+25])
set(l1,'YData',y(1:i),'XData',x);

axes(handles.axes2)
ylim([0 3])
xlim([t-25 t+25])
```



```
set(12, 'YData', f(1:i), 'XData', x);
```

Finalmente se leen los datos de las entradas analógicas de la tarjeta Arduino, se guardan en las variables ("v11" y "v22" en estos ejemplos) para posteriormente guardarlos en forma de vectores en una sola variable ("V") después se guardan en un archivo Excel utilizando el comando "xlswrite".

```
v11(i,1)=(handles.a.analogRead(13)*5)/1024;
v22(i,1)=(handles.a.analogRead(14)*5)/1024;

V=[v11 v22 v33 v44 v55 v66 v77];
global captura;
if captura==1
    xlswrite('Alcohol',V);
elseif captura==2
    xlswrite('Amoniac',V);
```

#### 4.3.2.2. Entrenar

Para la opción de "Entrenamiento de la Red Neuronal" se utiliza el botón "Entrenar" en el cual se toman los datos de los archivos Excel y se guardan en una variable para utilizarlos en el entrenamiento, posteriormente se le asigna un valor deseado de salida con los que identificará la red neuronal los datos del gas con el cual se está entrenando, finalmente con estos datos se entrena la red neuronal.

```
M=xlswread('Metano');
A=xlswread('Amoniac');
medida=size(M);
u=medida(1,1);
i=1;
G=1;
t=zeros;
while i<=u
    t(G,i)=1;
    i=i+1;
end
medida2=size(A);
n2=medida2(1,1);
i=1;
G=1;
t1=zeros;
while i<=n2
    t1(G,i)=20;
    i=i+1;
end
Entradas=[M;A];
T=[t t1];
net = newff([minmax(Entradas')],[20,10,1],{'logsig','logsig','purelin'},'traiblm');

net.trainparam.show=50;
net.trainparam.lr=0.05;
net.trainparam.max_fail=5;
net.trainparam.mem_reduc=1;
```



```
net.trainparam.mfx_grad=1e-10;
net.trainparam.mu=0.001;
net.trainparam.mu_bic=10;
net.trainparam.time=inf;
net.trainparam.epochs=300;
net.trainparam.goal=1e-5;

[net, tr]=train(net, Extraiss',T);
```

#### 4.3.2.3. Identificar

En la opción de Reconocimiento, utilizando el botón "Identificar" se vuelve a utilizar la programación para graficar los datos que llegan a las entradas analógicas de la tarjeta Arduino y mostrar los valores numéricamente en la pantalla.

Dependiendo de con cuales datos de gases se haya entrenado la red neuronal serán los gases que podrá identificar y que podrá mostrar en la pantalla.

Se utiliza el comando "sim()" para simular la red neuronal con la variable "Ident" donde están los datos que se están muestreando en ese momento e identificar que gas es el presentado a los sensores

```
Codigo de identificación
global chec1 chec2 chec3 chec4 chec5;
if (mod(i, Muestras)==0)
    Ident=[y(i,1);f(i,1);z(i,1);r(i,1);g(i,1);h(i,1);a(i,1)];
    if (chec1==1)
        red=sim(handles.net, Ident);
        A=round(red);

        if A==10
            set(handles.edit2, 'String', 'Alcohol');
        elseif A==20
            set(handles.edit2, 'String', 'Amoníaco');
        elseif A==30
            set(handles.edit2, 'String', 'Metano');
        elseif A==40
            set(handles.edit2, 'String', 'Butano');
        else
            set(handles.edit2, 'String', 'Desconocido');
        end
    end
end
```

El código del programa completo está en el Apéndice A.

#### 4.3.2.4. Red Neuronal

Se utiliza el "Neural Networks toolbox" de Matlab para crear la red neuronal con la cual se identifican los patrones de las muestras de gas en las pruebas.

Se crea una red neuronal "feed-forward backpropagation" con el comando "net=newff()" asignando las entradas con los datos de los gases muestreados y salidas deseadas de la red



neuronal, se utilizan veinte neuronas de entrada, diez neuronas en la capa oculta y una en la capa de salida.

Se utilizan las funciones de transferencia (también llamadas funciones de activación) "logsig" para la capa de entrada y la capa oculta, mientras que para la capa de salida se utiliza la función "pureline".

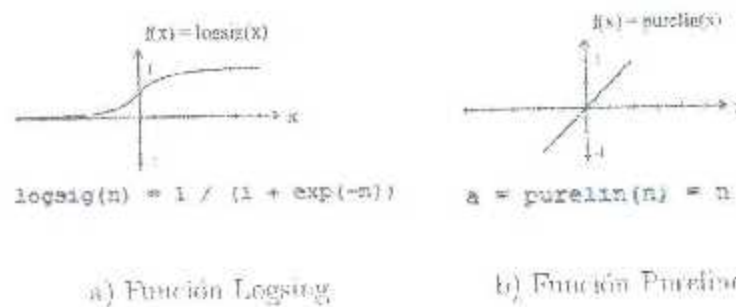


Figura 4.4: Funciones de Activación

El entrenamiento utilizado en la red neuronal es "Levenberg-Marquardt" mediante el comando "trainlm".

El algoritmo de Levenberg-Marquardt (LM) es un algoritmo iterativo de optimización en el que el método de iteración presenta una ligera modificación sobre el método tradicional de Newton. Las ecuaciones normales  $N\Delta = J^T J\Delta = J^T \xi$  ( $J$  representa el jacobiano de la función,  $\Delta$  los incrementos de los parámetros y  $\xi$  el vector de errores residuales del ajuste), son reemplazadas por las ecuaciones normales aumentadas  $N'\Delta = J^T \xi$ , donde  $N'_i = (1 + \lambda_i)N_i$  y  $N'_i = N_i$  para  $i \neq j$ . El valor de  $\lambda$  es inicialmente puesto a algún valor, normalmente  $\lambda = 10^{-3}$ . Si el valor de  $\Delta$  obtenido resolviendo las ecuaciones aumentadas conduce a una reducción del error, entonces el incremento es aceptado y  $\lambda$  es dividido por 10 para la siguiente iteración. Por otro lado si el valor de  $\Delta$  conduce a un aumento del error, entonces  $\lambda$  es multiplicado por 10 y se resuelven de nuevo las ecuaciones normales aumentadas, este proceso continúa hasta que el valor de  $\Delta$  encontrado da lugar aun decremento del error. Este proceso de resolver repetidamente las ecuaciones normales aumentadas para diferentes valores de  $\lambda$  hasta encontrar un valor aceptable de  $\Delta$  es lo que constituye una iteración del algoritmo de LM.

A continuación en la figura 4.15 se presenta la estructura de la red neuronal en Matlab.



Figura 4.15. Estructura de la Red en Matlab

Los parámetros aplicados a la red neuronal son mostrados en la tabla 4.6.

Parámetros	Significado
net.trainparam.max_fail	Errores máximos de validación
net.trainparam.min_grad	Gradiente mínimo
net.trainparam.mu	Valor inicial para incremento mu
net.trainparam.mu_inc	Factor de incremento
net.trainparam.time	Tiempo máximo de entrenamiento en segundos
net.trainParam.epochs	Máximo número de épocas a entrenar
net.trainParam.goal	Objetivo del entrenamiento

Tabla 4.6: Parámetros utilizados en la Red Neuronal

El código de la red neuronal utilizada en Matlab es el siguiente.

```
net = newff(minmax(Entradas'), [20,10,1], {'logsig','logsig','purelin'}, 'trainlm');

net.trainparam.show=50;
net.trainparam.lr=0.05;
net.trainparam.max_fail=5;
net.trainparam.mem_reduc=1;
net.trainparam.min_grad=1e-10;
net.trainparam.mu=0.001;
net.trainparam.mu_inc=10;
net.trainparam.time=inf;
net.trainparam.epochs=300;
net.trainparam.goal=1e-5;

[net, tr]=train(net, Entradas', 1);
```



---

---

## CAPÍTULO 5

---

### RESULTADOS EXPERIMENTALES

Para las pruebas experimentales y probar la red neuronal se capturaron los datos de varios gases, los cuales fueron: alcohol, amoníaco, Gas LP y metano.

A continuación se presentan los resultados obtenidos y la realización de las pruebas experimentales.

#### 5.1. Pruebas experimentales

- Para las pruebas experimentales con el alcohol y amoníaco se utiliza un matraz kitasato, en donde se genera vacío a 20 inHg (pulgadas de mercurio) mediante una bomba de vacío. Se introduce al matraz la muestra con la que se realizaran las pruebas con una jeringa ya que en el caso del alcohol y amoníaco es líquido (10 mililitros de muestra en cada caso), el vacío generado en el matraz kitasato es para que el gas liberado por el alcohol y el amoníaco viaje más rápido hacia donde se encuentra la matriz de sensores.
- Para el metano se produce una reacción con aserrín y ácido sulfúrico dentro del matraz kitasato (30 gramos de aserrín por 10 mililitros de ácido sulfúrico), posteriormente la muestra es presentada a la matriz de sensores.
- En el caso del gas LP se utiliza un soplete desechable y se introduce a la matriz de sensores utilizando la presión que genera el soplete.

En todos los casos la matriz de sensores se encuentra encapsulada para no tener entrada de algún otro compuesto que pueda meter ruido a la captura de los datos.

Las pruebas se realizaron bajo condiciones de temperatura ambiente ya que aunque la sensibilidad de los sensores es afectada por la temperatura, estos cuentan con una resistencia de calentamiento la cual es alimentada con un voltaje constante (igual al voltaje de alimentación

del sensor 3 vdc) para mantener la temperatura del sensor constante (50 C° aproximadamente) y que no se vea afectado por las variaciones de temperatura del medio ambiente, el circuito básico es mostrado en la figura 5.1, este circuito básico pudiera ser modificado para variar el voltaje de la resistencia de calefacción y modificar así la sensibilidad del sensor.



Figura 5.1: Resistencia de calentamiento del sensor MQ [53].

	Parte	Material
1	Capa de detección de gas	$SrO_2$
2	Electrodo	Au
3	Linea del Electrodo	Pt
4	Resistencia de calentamiento	Ni - Cr
5	Cerámica tubular	$SrO_2$
6	Red anti explosión	malla SU 8316100
7	Anillo de abrazadera	Ni
8	Basa de resina	baquelite
9	Pin del tubo	Ni

Tabla 5.1: Componentes de sensor MQ

La matriz de sensores está conectada a la tarjeta Arduino y esta envía los datos que se está recibiendo a la computadora donde mediante la interfaz gráfica se capturan los datos en una matriz de datos que posteriormente utilizaremos para entrenar la red neuronal.

Las pruebas se repitieron en varias ocasiones, primeramente se realizaron las pruebas para capturar los datos los gases y poderlos trabajar con redes neuronales, para esto se realiza una captura de datos por día comenzado con el alcohol, después el amoníaco, gas  $LP$  y finalmente con el metano.

Una vez que se logra tener un entrenamiento correcto de la red neuronal, se realizan pruebas de identificación repitiendo los experimentos nuevamente (uno por día comenzando con el alcohol) comprobando la identificación correcta del gas.

Finalmente se vuelve a realizar la captura de los datos, el entrenamiento e identificación de cada uno de los gases para corroborar los resultados experimentales.

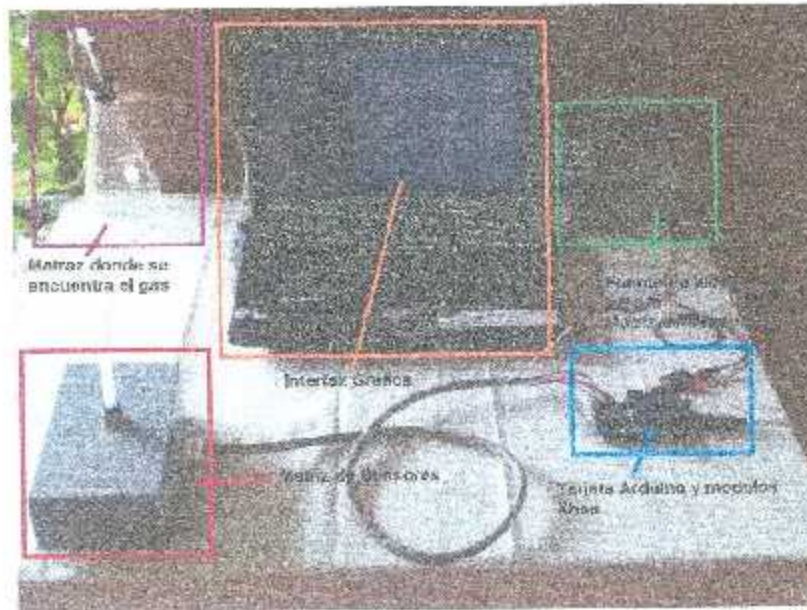


Figura 5.2: Prototipo utilizado para el experimento

En la figura 5.2 se muestra el prototipo que se utilizó para la realización de los experimentos donde se tiene la matriz de sensores, el sistema de comunicación inalámbrica (Tarjeta Arduino y módulos XBee), una fuente utilizada para alimentar la matriz de sensores con 12 vdc, la interfaz gráfica y el matraz donde se encuentra el gas con el que se realizará el experimento.



Figura 5.3: Matriz de sensores

En la figura 5.3 se observa la matriz de sensores utilizada para el prototipo.

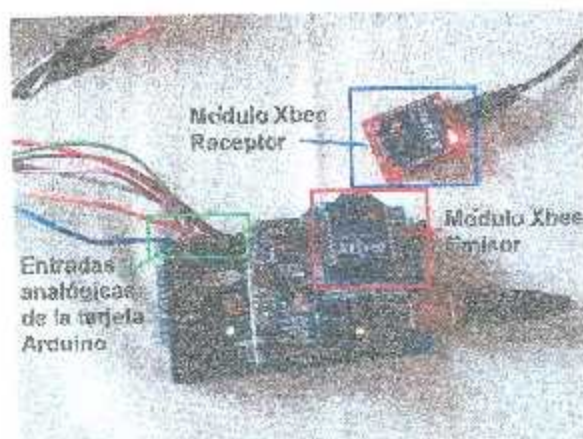


Figura 5.3: Comunicación inalámbrica de la tarjeta Arduino

La figura 5.4 muestra la comunicación inalámbrica de la tarjeta Arduino con el ordenador mediante los módulos Xbee.

El tiempo que dura cada captura de datos es de 1000 segundos (más de 16 minutos aproximadamente), se comienza a tomar lectura de los sensores al instante en que se libera el gas hacia ellos y a los 850 segundos aproximadamente se retira el gas de los sensores por lo que tienen 150 segundos para estabilizarse y llegar prácticamente a su estado inicial.



Figura 5.5: Esquema del experimento realizado.

En la figura 5.5 observamos el esquema de como se realizaron las pruebas, donde primero se preparó la muestra en el matraz kitasato, después el gas de la muestra llega a los sensores que envían los datos a la tarjeta adquisidora Arduino y se registran en la computadora.

Para la captura de los datos de los gases se utiliza una matriz de sensores, los cuales fueron descritos en el capítulo cuatro, esta matriz entregará una "huella digital" la cual es la combinación de los datos entregados por los sensores.

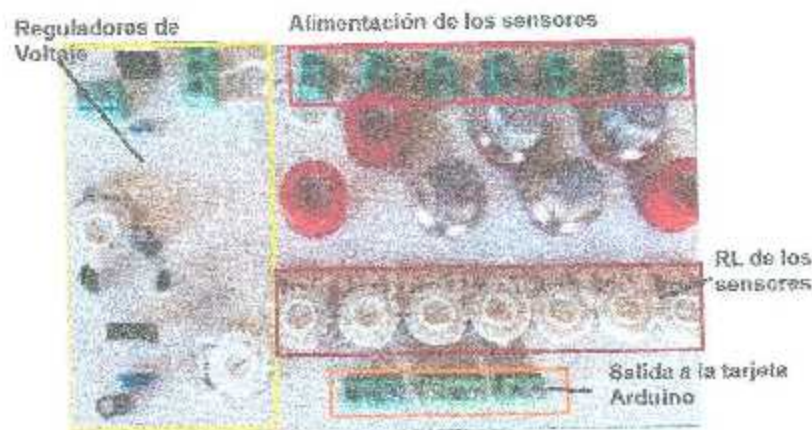


Figura 5.6: Matriz de sensores

En la figura 5.6 podemos observar la tarjeta donde se encuentran montados los sensores que se utilizaron para realizar las pruebas experimentales, y que conforman la matriz de sensores, se tiene también dos reguladores de voltaje que entregan 3 y 1.5 volts que son la alimentación requerida por los sensores, la "RL" necesaria para realizar el divisor de tensión y generar el voltaje que llegara a la tarjeta Arduino.

## 5.2. Entrenamiento de la red neuronal

Para el entrenamiento de la Red Neuronal se utilizan las matrices que se capturaron con las pruebas experimentales, una vez que se tienen todas las matrices se forma una sola matriz llamada "Entradas" con todos los datos, como ya se menciono cada matriz corresponde a una "luzella digital" y a cada una se le asigna una salida deseada, para saber a qué gas corresponde cada entrada de datos a la Red y así entrenarla para que pueda reconocer una muestra de gas de otro.

```
M=xlsread('Metano');
A=xlsread('Amoníaco');
AL=xlsread('Alcohol');
LP=xlsread('LP');
medida=size(M);
u=medida(1,1);
i=1;
G=1;
t=zeros;
while i<=u
    t(G,i)=1;
    i=i+1;
end
medida=size(A);
u2=medida2(1,1);
i=1;
G=1;
t1=zeros;
while i<=u2
    t1(G,i)=20;
```

```

        i=i-1;
    end
    medida=size(AI);
    u=medida(1,1);
    i=1;
    G=1;
    t2=zeros;
    while i<=u
        t2(G,i)=30;
        i=i+1;
    end
    medida=size(IP);
    u=medida(1,1);
    i=1;
    G=1;
    t3=zeros;
    while i<=u
        t3(G,i)=10;
        i=i+1;
    end
    Entradas=[M;A;AI;IP];
    T=[t t1 t2 t3];
    net = newff(minmax(Entradas'),[20,10,1],{'logsig','logsig','purelin'},'trainlm');
    net.trainparam.show=50;
    net.trainparam.tr=0.05;
    net.trainparam.max_fail=5;
    net.trainparam.mem_reduc=1;
    net.trainparam.min_grad=1e-10;
    net.trainparam.nm=0.001;
    net.trainparam.mu_inc=10;
    net.trainparam.time=inf;
    net.trainparam.epochs=300;
    net.trainparam.goal=1e-5;

    [net, tr]=train(net,Entradas',T);

```

En el listado del programa anterior se observa como se cargan los datos de cada matriz de las muestras guardadas en Excel para asignarles las salidas deseadas utilizando el comando "while" y crear la matriz "Entradas" con la que se entrena la red.

Para observar el comportamiento de la red con los datos de entrenamiento se simula la red con la matriz con la que fue entrenada para observar las salidas que arroja, si las salidas que entrega la red son las mismas que las salidas deseadas con las que fue entrenada entonces el resultado del entrenamiento es exitoso.

```
prueba=sim(net,Entradas');
```

Una vez entrenada la Red Neuronal y que los datos han sido exitosos, se prueba nuevamente la red entrenada utilizando el mismo procedimiento antes mencionado para generar los datos de entrenamiento, es decir se vuelven a preparar las muestras para que nuevamente sean presentados a los sensores y así simular en tiempo real la red con los datos que esta capturando en ese momento la interfaz gráfica.

Los datos de entrenamiento de cada gas con los que se realizaron las pruebas se encuen-



tran en el Apéndice B.

### 5.3. Resultados

Los Resultados obtenidos en el entrenamiento de la Red Neuronal fueron satisfactorios, ya que al meter nuevamente los datos con los que fue entrenada la Red en la simulación, ésta arroja las salidas deseadas para cada "huella digital" de los gases muestreados por lo que podemos decir que el entrenamiento fue exitoso.

```
>> sim(net ,M(130 ,:))

ans =

    1.0001

>> sim(net ,A(300 ,:))

ans =

    20.0001

>> sim(net ,Al(270 ,:))

ans =

    29.9999

>> sim(net ,IP(80 ,:))

ans =

    40.0000
```

En el anterior listado del programa se realizó la prueba con los datos de entrenamiento introduciendo una muestra de cada "huella digital" que se toma, donde el comando `sim(net,"",:)` simula la red "net" con la matriz con la que se quiere simular, en nuestro caso solo se simula una muestra, es por eso que entre paréntesis se encuentra un número que es la muestra ( renglón de la matriz) y los dos puntos son todos los datos de la muestras (todas las columnas).

```
>> M(130 ,:)

ans =

    1.7090    1.8164    1.8066    1.7578    2.0068    0.5371    0.6680

>> M(130 ,:)

ans =

    1.7090
    1.8164
    1.8066
    1.7578
```



2.6668
0.5371
0.6689

Aquí se ejemplifica lo mencionado en el párrafo anterior donde cada columna es un dato entregado por uno de los siete sensores (voltaje) en la muestra 130 de Metano, para poder meter la muestra a la Red Neuronal y simularla tiene que ser en forma de vector como se puede observar.

El entrenamiento de la red arroja los siguientes Resultados:

- Iteraciones = 97
- Tiempo de entrenamiento = 54 segundos
- Error =  $3.66 \times 10^{-6}$

Estos resultados mostrados en la figura 5.7 quieren decir que se necesitaron 54 segundos para realizar 97 iteraciones y lograr entrenar la Red Neuronal con un error mínimo del  $1 \times 10^{-5}$ , sin embargo se logró un error de  $3.66 \times 10^{-6}$  lo cual nos indica que se logró el objetivo del entrenamiento.



Figura 5.7: Resultados del entrenamiento

En la figura 5.8 observamos la gráfica donde se tiene el entrenamiento ("Train"), el objetivo del error deseado ("Goal" =  $1 \times 10^{-5}$ ) y el resultado logrado ("Best") en las 97 iteraciones.





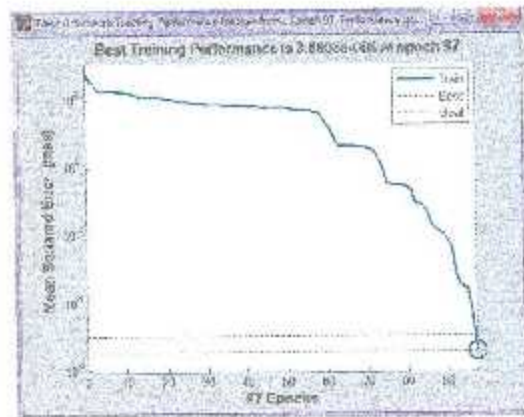


Figura 5.8: Gráfica del entrenamiento

La prueba del entrenamiento de la Red Neuronal con los datos de las muestras de los gases fue satisfactoria ya que la Red puede identificar cada uno de los gases que fueron muestreados.

Para la prueba de identificación de patrones en tiempo real con la interfaz gráfica se reproducen las condiciones con las cuales fueron tomados los datos de entrenamiento para la Red Neuronal, se entrena la Red con los datos de entrenamiento y se procede a leer lo que está recibiendo la matriz de sensores en ese momento.

En esta prueba el resultado fue bueno ya que aunque en los primeros segundos la interfaz gráfica indica "Desconocido", a los 85 segundos aproximadamente para el Amoniaco, 130 segundos para el alcohol, 30 segundos para el gas LP, lograron ser reconocidos por la Red Neuronal y la interfaz gráfica los mostró en la pantalla.

La rapidez con la que la Red Neuronal logró reconocer las muestras de los gases se debió a varios factores, como la velocidad con la que viaja el gas hasta los sensores, el tiempo en que tarda el gas en inundar el recipiente donde se encuentra encapsulado la matriz de sensores, etc.

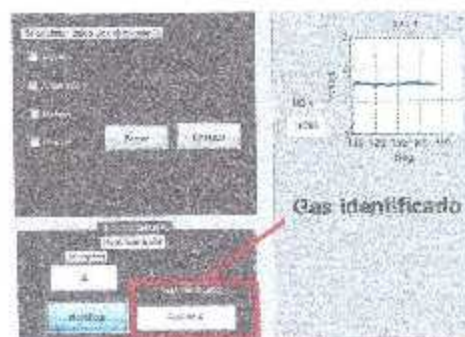


Figura 5.9: Identificación del Amoniaco

En la figura 5.9 se observa la pantalla donde se está identificando el Amoniaco.

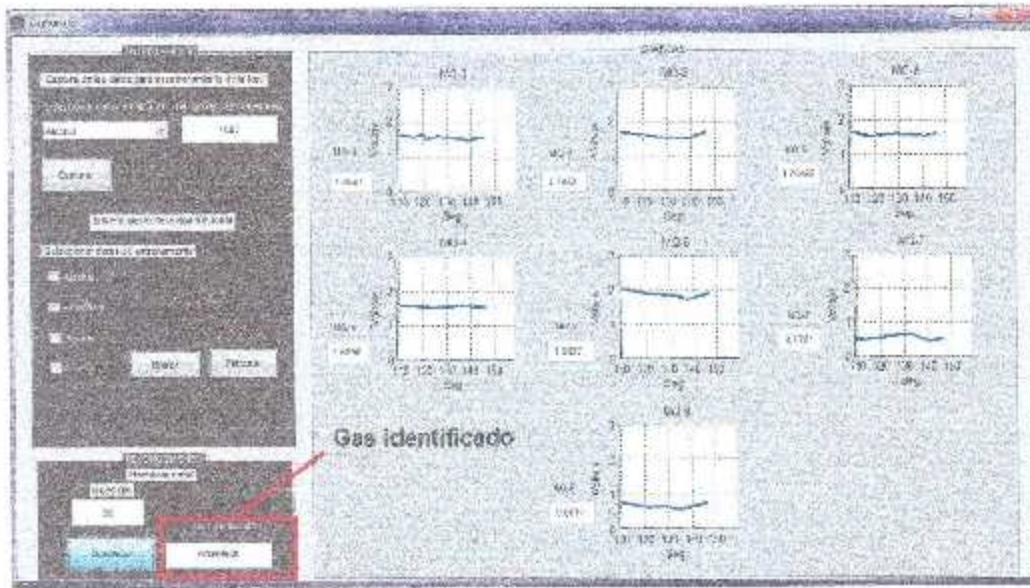


Figura 5.10: Pantalla de la identificación del Amoníaco

En la figura 5.10 se observa la pantalla con las gráficas recibidas por los sensores para la identificación del amoníaco.

---

---

## CAPÍTULO 6

---

### CONCLUSIONES

- La Nariz Electrónica es capaz de emular el olfato humano y su desarrollo ha crecido en los últimos años debido a la cantidad de aplicaciones en las que puede ser utilizada como: la medicina (ayudar en el diagnóstico de algunas enfermedades), Medio Ambiente (en la regulación de algunos contaminantes) o en la seguridad (para la detección de componentes químicos dañinos para la salud), cuya principal variable son los sensores utilizados los cuales son elegidos de acuerdo a la aplicación.
- A pesar de que la nariz electrónica puede emular el funcionamiento de la nariz humana y es capaz de identificar un aroma o componente de otro, aun falta mucho en el desarrollo de estos prototipos para poder igualar al olfato biológico, ya que aun no existen sensores con la sensibilidad y lo suficientemente selectivos para compararse con el sistema olfativo de los seres vivos.
- La Nariz Electrónica desarrollada en este trabajo de tesis es capaz de identificar cada una de las diferentes muestras de gases presentados, utilizando la teoría de Redes Neuronales para la identificación de patrones.
- Las Redes Neuronales son la herramienta más importante en el desarrollo de este prototipo ya que las ventajas que ofrece como el Aprendizaje Adaptativo, Tolerancia a fallos, Auto organización, Adaptabilidad y Flexibilidad hacen de las redes neuronales ideales para la identificación de patrones sobre otros métodos.
- La red Neuronal utilizada para la identificación de las diferentes muestras de gases es la red "feed-forward backpropagation" debido a su capacidad de generalización, en donde el algoritmo de entrenamiento "Levenberg-Marquardt" fue con el que mejores resultados se obtuvieron.
- Durante la realización de las pruebas experimentales se comprobó que la respuesta de cada uno de los sensores ante la presencia de un gas varía de acuerdo a la sensibilidad que tiene respecto a dicho gas, en algunas pruebas hubo sensores que tuvieron una baja respuesta y una respuesta más alta ante otro, esto es debido a la selectividad de los sensores, la cual es la capacidad que tienen para reaccionar solo ante la presencia

del componente para el que son fabricados ya que en teoría solo deberían reaccionar ante ese componente.

- La matriz de sensores se debe elegir de acuerdo a las necesidades de cada aplicación ya que dependiendo de esta elección obtendremos una "huella digital" que es la combinación de los datos de la matriz de sensores, que como ya se mencionó anteriormente es necesario por su selectividad (pueden reaccionar a varios gases, no solo para el que fueron creados, aunque para este sean más sensibles) además de que en el caso de querer identificar un compuesto o aroma se requieren los sensores que reaccionen para cada parte de dicho compuesto.
- Con los resultados experimentales obtenidos en este trabajo de tesis podemos concluir que es posible utilizar una nariz electrónica para identificar una gran cantidad de gases dañinos para la salud en un ambiente de riesgo para el ser humano mediante una adecuada selección de sensores.

## 6.1. Trabajos futuros

- Implementar una plataforma robótica para montar la Nariz Electrónica y tener acceso a lugares donde exista riesgo para el ser humano.
- Desarrollar un sistema para tomar muestras del medio ambiente (succiones) que lleguen a la Nariz Electrónica cada cierto tiempo para muestrear las condiciones del lugar donde se encuentre la Nariz.
- Desarrollar un control para la plataforma robótica que la dirija hacia donde la nariz detecta un gas para buscar fugas de gases automáticamente.



---

---

## APÉNDICE A

---

### PROGRAMA DE LA INTERFAZ GRÁFICA

```
function varargout = Captura_6(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @Captura_6_OpeningFcn, ...
                  'gui_OutputFcn',  @Captura_6_OutputFcn, ...
                  'gui_LayerFcn',   [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function Captura_6_OpeningFcn(hObject, eventdata, handles, varargin)
handles.a=arduino('COM');
handles.output = hObject;
guidata(hObject, handles);

function varargout = Captura_6_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function togglebutton1_Callback(hObject, eventdata, handles)
die=get(hObject, 'value');
handles.die=die;
handles.a.pinMode(52, 'output');
handles.a.digitalWrite(52,1);
rate=1.75;
Muestras=handles.edit13;
Ah=0;
Am=0;
Me=0;
Bu=0;
```

```

AhAm=0;
y = zeros;
f = zeros;
x = zeros;
r = zeros;
g = zeros;
h = zeros;
u = zeros;
i=1;
axes(handles.axes1)
cla;
title('MQ-3');
xlabel('Seg. ');
ylabel('Voltaje');
l1 = line(nan,nan,'Color','b','LineWidth',2);
grid on;
hold on;
axes(handles.axes2)
cla;
title('MQ-2');
xlabel('Seg. ');
ylabel('Voltaje');
l2 = line(nan,nan,'Color','b','LineWidth',2);
grid on;
hold on;
axes(handles.axes3)
cla;
title('MQ-8');
xlabel('Seg. ');
ylabel('Voltaje');
l3 = line(nan,nan,'Color','b','LineWidth',2);
grid on;
hold on;
axes(handles.axes4)
cla;
title('MQ-4');
xlabel('Seg. ');
ylabel('Voltaje');
l4 = line(nan,nan,'Color','b','LineWidth',2);
grid on;
hold on;
axes(handles.axes5)
cla;
title('MQ-6');
xlabel('Seg. ');
ylabel('Voltaje');
l5 = line(nan,nan,'Color','b','LineWidth',2);
grid on;
hold on;
axes(handles.axes6)
cla;
title('MQ-7');
xlabel('Seg. ');
ylabel('Voltaje');
l6 = line(nan,nan,'Color','b','LineWidth',2);
grid on;
hold on;

```



```

axes(handles.axes7)
cla;
title('AQ-6');
xlabel('Seg. ');
ylabel('Voltaje');
l7 = line(nan,nan,'Color','b','LineWidth',2);
grid on;
hold on;
tic
while (handles.dic==1)
    t=toc;
    % leer del puerto serie
    y(i,1)=(handles.a.analogRead(15)*5)/1024;
    f(i,1)=(handles.a.analogRead(14)*5)/1024;
    z(i,1)=(handles.a.analogRead(13)*5)/1024;
    r(i,1)=(handles.a.analogRead(12)*5)/1024;
    g(i,1)=(handles.a.analogRead(11)*5)/1024;
    b(i,1)=(handles.a.analogRead(10)*5)/1024;
    n(i,1)=(handles.a.analogRead(9)*5)/1024;
    set(handles.edit4,'String',(handles.a.analogRead(15)*5)/1024);
    set(handles.edit5,'String',(handles.a.analogRead(14)*5)/1024);
    set(handles.edit7,'String',(handles.a.analogRead(13)*5)/1024);
    set(handles.edit8,'String',(handles.a.analogRead(12)*5)/1024);
    set(handles.edit9,'String',(handles.a.analogRead(11)*5)/1024);
    set(handles.edit10,'String',(handles.a.analogRead(10)*5)/1024);
    set(handles.edit11,'String',(handles.a.analogRead(9)*5)/1024);
    %Notiyo de identificacion
    global chec1 chec2 chec3 chec4;
    if (mod(i,Muestras)==0)
        Ident=[y(i,1);f(i,1);z(i,1);r(i,1);g(i,1);b(i,1);n(i,1)];
        if(chec1==1)
            red=sim(handles.net,Ident);
            Aa=round(red);
        end
        if(chec2==1)
            red=sim(handles.net,Ident);
            Am=round(red);
        end
        if(chec3==1)
            red=sim(handles.net,Ident);
            Ma=round(red);
        end
        if(chec4==1)
            red=sim(handles.net,Ident);
            Ba=round(red);
        end
        if Aa==1
            set(handles.edit2,'String','Alcohol');
        elseif Am==20
            set(handles.edit2,'String','Amonisco');
        elseif Ma==30
            set(handles.edit2,'String','Mezcla');
        elseif Ba==40
            set(handles.edit2,'String','Gas LP');
        else
            set(handles.edit2,'String','Desconocido');
        end
    end
end

```



```

end

x = linspace(0,1/rate,i);
axes(handles.axes1)
ylim([0 3])
xlim([-25 t+25])
set(11,'YData',y(1:i),'XData',x);
axes(handles.axes2)
ylim([0 3])
xlim([-25 t+25])
set(12,'YData',f(1:i),'XData',x);
axes(handles.axes3)
ylim([0 3])
xlim([-25 t+25])
set(13,'YData',z(1:i),'XData',x);
axes(handles.axes4)
ylim([0 3])
xlim([-25 t+25])
set(14,'YData',r(1:i),'XData',x);
axes(handles.axes5)
ylim([0 3])
xlim([-25 t+25])
set(15,'YData',g(1:i),'XData',x);
axes(handles.axes6)
ylim([0 3])
xlim([-25 t+25])
set(16,'YData',h(1:i),'XData',x);
axes(handles.axes7)
ylim([0 3])
xlim([-25 t+25])
set(17,'YData',u(1:i),'XData',x);
drawnow
i=i+1;
end
if (handles.dio==0)
clear 11;
clear 12;
clear 13;
clear 14;
clear 15;
clear 16;
axes(handles.axes1)
cla;
axes(handles.axes2)
cla;
axes(handles.axes3)
cla;
axes(handles.axes4)
cla;
axes(handles.axes5)
cla;
axes(handles.axes6)
cla;
axes(handles.axes7)
cla;
end

```





```

function pushbutton1_Callback(hObject, eventdata, handles)
handles.a.piuMoce(52, 'output');
handles.a.digitalWrite(52,1);
total=handles.edi11;
rate=1.8;
i = 1;
y = zeros;
f = zeros;
z = zeros;
r = zeros;
g = zeros;
h = zeros;
u = zeros;
v11 = zeros;
v22 = zeros;
v33 = zeros;
v44 = zeros;
v55 = zeros;
v66 = zeros;
v77 = zeros;
t=1;
i=1;
%Preparacion de la figura %%%%%%%%%%%
axes(handles.axes1)
cla;
title('MQ-3');
xlabel('Seg. ');
ylabel('Voltaje');
l1 = line(nan,nan, 'Color', 'b', 'LineWidth', 2);
grid on;
hold on;
axes(handles.axes2)
cla;
title('MQ-2');
xlabel('Seg. ');
ylabel('Voltaje');
l2 = line(nan,nan, 'Color', 'b', 'LineWidth', 2);
grid on;
hold on;
axes(handles.axes3)
cla;
title('MQ-8');
xlabel('Seg. ');
ylabel('Voltaje');
l3 = line(nan,nan, 'Color', 'b', 'LineWidth', 3);
grid on;
hold on;
axes(handles.axes4)
cla;
title('MQ-4');
xlabel('Seg. ');
ylabel('Voltaje');
l4 = line(nan,nan, 'Color', 'b', 'LineWidth', 2);
grid on;
hold on;
axes(handles.axes5)
cla;

```



```

title('MQ-6');
xlabel('Seg. ');
ylabel('Voltaje');
l5 = line(nan,nan,'Color','b','LineWidth',2);
grid on;
hold on;
axes(handles.axes6)
cla;
title('MQ-7');
xlabel('Seg. ');
ylabel('Voltaje');
l6 = line(nan,nan,'Color','b','LineWidth',2);
grid on;
hold on;
axes(handles.axes7)
cla;
title('MQ-8');
xlabel('Seg. ');
ylabel('Voltaje');
l7 = line(nan,nan,'Color','b','LineWidth',2);
grid on;
hold on;
tic
while t<ttotal
    t=toc;
    % leer del puerto serie
    y(i)=(handles.a.analogRead(15)*5)/1024;
    f(i)=(handles.a.analogRead(14)*5)/1024;
    z(i)=(handles.a.analogRead(13)*5)/1024;
    r(i)=(handles.a.analogRead(12)*5)/1024;
    g(i)=(handles.a.analogRead(11)*5)/1024;
    h(i)=(handles.a.analogRead(10)*5)/1024;
    n(i)=(handles.a.analogRead(9)*5)/1024;
    v11(i,1)=(handles.a.analogRead(15)*5)/1024;
    v22(i,1)=(handles.a.analogRead(11)*5)/1024;
    v33(i,1)=(handles.a.analogRead(13)*5)/1024;
    v44(i,1)=(handles.a.analogRead(12)*5)/1024;
    v55(i,1)=(handles.a.analogRead(11)*5)/1024;
    v66(i,1)=(handles.a.analogRead(10)*5)/1024;
    v77(i,1)=(handles.a.analogRead(9)*5)/1024;
    set(handles.edit4,'String',(handles.a.analogRead(15)*5)/1024);
    set(handles.edit5,'String',(handles.a.analogRead(14)*5)/1024);
    set(handles.edit7,'String',(handles.a.analogRead(13)*5)/1024);
    set(handles.edit8,'String',(handles.a.analogRead(12)*5)/1024);
    set(handles.edit9,'String',(handles.a.analogRead(11)*5)/1024);
    set(handles.edit10,'String',(handles.a.analogRead(10)*5)/1024);
    set(handles.edit11,'String',(handles.a.analogRead(9)*5)/1024);
    x = linspace(0,rate,i);
    axes(handles.axes1)
    ylim([0 3])
    xlim([t-25 t+25])
    set(11,'YData',y(1:i),'XData',x);
    axes(handles.axes2)
    ylim([0 3])
    xlim([t-25 t+25])
    set(12,'YData',f(1:i),'XData',x);
    axes(handles.axes3)

```



```

ylim([0 3])
xlim([t-25 t+25])
set(13, 'YData', z(1:i), 'XData', x);
axes(handles.axes4)
ylim([0 3])
xlim([t-25 t+25])
set(14, 'YData', r(1:i), 'XData', x);
axes(handles.axes5)
ylim([0 3])
xlim([t-25 t+25])
set(15, 'YData', g(1:i), 'XData', x);
axes(handles.axes6)
ylim([0 3])
xlim([t-25 t+25])
set(16, 'YData', h(1:i), 'XData', x);
axes(handles.axes7)
ylim([0 3])
xlim([t-25 t+25])
set(17, 'YData', u(1:i), 'XData', x);
drawnow
i=i+1;
end
handles.a.digitalWrite(53,0);
V=[v11 v22 v33 v44 v55 v66 v77];
global captura;
if captura==1
    xlswrite('Alcohol',V);
elseif captura==2
    xlswrite('Amoníaco',V);
elseif captura==3
    xlswrite('Metano',V);
elseif captura==4
    xlswrite('LP',V);
end
axes(handles.axes1)
cla;
axes(handles.axes2)
cla;
axes(handles.axes3)
cla;
axes(handles.axes4)
cla;
axes(handles.axes5)
cla;
axes(handles.axes6)
cla;
axes(handles.axes7)
cla;
cla;

function edit1_Callback(hObject, eventdata, handles)
val=get(hObject, 'string'); %almacena valor ingresado
Newval=str2double(val); %transformar a formato double
handles.edit1=Newval; %almacena en identificador
guidata(hObject, handles);

function edit1_CreateFcn(hObject, eventdata, handles)

```



```

if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, '
    defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function pushbutton2_Callback(hObject, eventdata, handles)
clear handles.net
clear handles.net2
clear handles.net3
clear handles.net4
clear handles.net5

function pushbutton3_Callback(hObject, eventdata, handles)
global chec1 chec2 chec3 chec4;
Entradas=zeros;
T=zeros;
if chec1==1
AL=xlsread('Alcohol');
medida=size(AL);
u=medida(1,1);
p=V(1,:);
i=1;
j=1;
G=1;
k=100;
t=zeros;
T=zeros;
while i<=u
    t(G,i)=1;
    i=i+1;
end
Entradas=[Entradas;AL];
T=[T t];
end
if chec2==1
A=xlsread('Amoniaco');
medida=size(A);
u=medida(1,1);
p=V(1,:);
i=1;
j=1;
G=1;
k=100;
t=zeros;
T=zeros;
while i<=u
    t1(G,i)=20;
    i=i+1;
end
Entradas=[Entradas;A];
T=[T t1];
end
if chec3==1
M=xlsread('Metano');
medida=size(M);
u=medida(1,1);
p=V(1,:);

```



```

i=1;
j=1;
G=1;
k=100;
t=zeros;
T=zeros;
while i<=n
    t2(G,i)=30;
    i=i+1;
end
Entradas=[Entradas;t];
T=[T t2];
end
if check==1
IP=xlswread('IP');
medida=size(IP);
v=medida(1,1);
p=V(1,:);
i=1;
j=1;
G=1;
k=100;
t=zeros;
T=zeros;
while i<=n
    t3(G,i)=40;
    i=i+1;
end
Entradas=[Entradas;IP];
T=[T t3];
net = newff(minmax(Entradas'),[20,10,1],{'logsig','logsig','purelin'},'trainlm');

net.trainparam.show=50;
net.trainparam.lr=0.05;
net.trainparam.max_fail=5;
net.trainparam.mem_reduc=1;
net.trainparam.min_grad=1e-10;
net.trainparam.ms=0.001;
net.trainparam.num_bac=10;
net.trainparam.time=inf;
net.trainparam.epochs=300;
net.trainparam.goal=1e-5;
[net,tr]=train(net,Entradas',T);
end
guidata(hObject,handles);

function edit2_Callback(hObject, eventdata, handles)
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)
function edit4_CreateFcn(hObject, eventdata, handles)

```



```

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)
function edit5_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function popupmenu1_Callback(hObject, eventdata, handles)
global captera;
captera=get(hObject,'value') %almacena valor ingresado
function popupmenu1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function checkbox1_Callback(hObject, eventdata, handles)

```



```

global chec1;
chec1=get(hObject,'Value')

function checkbox2_Callback(hObject, eventdata, handles)
global chec2;
chec2=get(hObject,'Value')

function checkbox3_Callback(hObject, eventdata, handles)
global chec3;
chec3=get(hObject,'Value')

function checkbox4_Callback(hObject, eventdata, handles)
global chec4;
chec4=get(hObject,'Value');

function edit13_Callback(hObject, eventdata, handles)
val=get(hObject,'string'); %almacena valor ingresado
Newval=str2double(val); %transformar a formato double
handles.edit3=Newval; %almacena en identificador
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```



## APÉNDICE B

### DATOS CAPTURADOS DE LAS PRUEBAS EXPERIMENTALES

#### B.1. Alcohol: Matriz AL

2.1484375	2.3681640625	0.2318359375	2.470703125	2.5244140625	0.56640625	1.728515625
2.1327890625	2.3681640625	0.1818359375	2.470703125	2.5222065625	0.5615244375	1.7333984375
2.1155546875	2.3681640625	0.13671875	2.485515625	2.51953125	0.56640625	1.73828125
2.158203125	2.40234375	0.13671875	2.51953125	2.5752136875	0.56640625	1.71875
2.1484375	2.3678953125	0.13671875	2.4951171875	2.548828125	0.5712899625	1.74846875
2.1387890625	2.3681640625	0.1318359375	2.4951171875	2.51353125	0.5615244375	1.748515625
2.1149578125	2.3583081875	0.1318359375	2.4009375	2.5160481875	0.5615244375	1.71875
2.1112976125	2.36828125	0.1416015625	2.470703125	2.5043828125	0.5615244375	1.723328125
2.129909375	2.36028125	0.1918359375	2.468208125	2.51953125	0.5615244375	1.728115625
2.1357890625	2.370046875	0.1918359375	2.190703125	2.5244140625	0.5615244375	1.7333984375
2.138671875	2.3828125	0.13671875	2.485515625	2.5399625	0.56640625	1.7331640625
2.1484375	2.393578125	2.1318359375	2.4651171875	2.548828125	0.5712899625	1.743460625
2.1117890625	2.375040875	0.1518359375	2.470703125	2.5222065625	0.5615244375	1.7333984375
2.1435546875	2.3779296875	0.13671875	2.485515625	2.525266875	0.56640625	1.73828125
2.1357890625	2.374166875	0.1918359375	2.190703125	2.5244140625	0.5615244375	1.7333984375
2.138671875	2.4828125	0.1918359375	2.190703125	2.529296875	0.56640625	1.73828125
2.138671875	2.3828125	0.13671875	2.485515625	2.5430458125	0.56640625	1.73828125
2.1484375	2.3579953125	0.1318359375	2.48046875	2.529296875	0.56640625	1.7333984375
2.1435546875	2.3538125	0.13671875	2.485515625	2.529296875	0.56640625	1.73828125
2.138671875	2.3848125	0.1518359375	2.48046875	2.5211796875	0.56640625	1.73828125
2.1484375	2.3878953125	0.1518359375	2.460294875	2.5350625	0.56640625	1.74846875
2.1240234375	2.36298125	0.1918359375	2.1655203125	2.516184375	0.5615244375	1.728515625
2.138671875	2.370046875	0.129653125	2.470703125	2.5244140625	0.5615244375	1.7333984375
2.1435546875	2.3631340625	0.1818359375	2.48046875	2.520296875	0.56640625	1.73828125
2.1435546875	2.3525125	0.13671875	2.485515625	2.50189296875	0.56640625	1.7431640625
2.1333203125	2.392878125	0.1518359375	2.48046875	2.543955125	0.5712899625	1.74846875
2.1484375	2.3878953125	0.1318359375	2.490531875	2.548828125	0.56640625	1.7431640625
2.1435546875	2.3779296875	0.1918359375	2.48046875	2.523306875	0.56640625	1.73828125
2.1435546875	2.3828125	0.1918359375	2.485515625	2.5411796875	0.56640625	1.7431640625
2.1679875	2.407265625	0.1518359375	2.514845375	2.5732121875	0.5615244375	1.757109375
2.1533203125	2.392578125	0.1818359375	2.48046875	2.543955125	0.5712899625	1.74846875
2.1533203125	2.392578125	0.13671875	2.4951171875	2.543955125	0.56640625	1.74846875



Apéndice B. Datos capturados de las pruebas experimentales

2.14820125	2.40234375	0.13671875	2.5	2.5537106375	0.5712890625	1.7678125
2.1726315625	2.14209375	0.13671875	3.5	2.5834783625	0.570171875	1.7578125
2.158203125	2.3674609375	0.1318359375	2.4853515625	2.5341798875	0.56640625	1.73828125
2.153209125	2.367068125	0.126953125	2.490344375	2.5341798875	0.56640625	1.748046875
2.1435340875	2.3828125	0.1318359375	2.4853515625	2.529296875	0.56640625	1.7431640625
2.138403125	2.3974609375	0.13671875	2.5	2.545825125	0.56640625	1.7529206875
2.1435340875	2.3779296875	0.13671875	2.4756830375	2.51953125	0.56640625	1.7330284375
2.143671875	2.3779296875	0.13671875	2.48046875	2.5341740625	0.56640625	1.7431640625
2.14716875	2.4072365625	0.13671875	2.5	2.5300025	0.56640625	1.7431640625
2.153209125	2.3974609375	0.13671875	2.49234375	2.5439453125	0.5712890625	1.7529206875
2.138403125	2.3981040625	0.13671875	2.4609375	2.5146484375	0.5615234375	1.728515625
2.138671875	2.373946875	0.13671875	2.470703125	2.5344140625	0.56640625	1.73828125
2.158203125	2.40234375	0.13671875	2.5	2.545825125	0.570171875	1.7578125
2.1484375	2.3870359125	0.126953125	2.490236875	2.5439453125	0.5712890625	1.748046875
2.16796875	2.40234375	0.13671875	2.4853515625	2.53696875	0.56640625	1.7431640625
2.12890625	2.36328125	0.1318359375	2.4699375	2.509769375	0.5615234375	1.7333884375
2.1317896625	2.374016875	0.1318359375	2.470703125	2.5146484375	0.56640625	1.7333884375
2.138671875	2.3674609375	0.1318359375	2.470703125	2.51363125	0.56640625	1.73328125
2.1240234375	2.4173296875	0.1318359375	2.4755860375	2.5241440625	0.56640625	1.73828125
2.1435340875	2.3779296875	0.1387875	2.4853515625	2.5241440625	0.56640625	1.7431640625
2.138671875	2.373946875	0.1416015625	2.470703125	2.51953125	0.570171875	1.73828125
2.1337800625	2.3779296875	0.1318359375	2.4735850625	2.5241440625	0.56640625	1.73828125
2.138671875	2.3779296875	0.1318359375	2.4735850625	2.5241440625	0.56640625	1.7431640625
2.1435340875	2.3779296875	0.1318359375	2.48046875	2.5241140625	0.56640625	1.7431640625
2.1484375	2.3828125	0.1318359375	2.48046875	2.5241140625	0.56640625	1.7431640625
2.1484375	2.3928125	0.1267875	2.4853515625	2.5341798875	0.5712890625	1.748046875
2.1484375	2.38795125	0.1318359375	2.480236875	2.5341798875	0.5712890625	1.748046875
2.1633039125	2.3870359125	0.1318359375	2.480236875	2.53996375	0.5610016875	1.7589296875
2.158203125	2.40234375	0.1318359375	2.5	2.545825125	0.5610016875	1.7620853125
2.1484375	2.3870359125	0.13671875	2.490236875	2.5711796875	0.5712890625	1.7620853125
2.1337800625	2.363040875	0.13671875	2.4632963125	2.51963125	0.56946875	1.737578125
2.1484375	2.3828125	0.13671875	2.48046875	2.529296875	0.56640625	1.748046875
2.153209125	2.40234375	0.13671875	2.490171875	2.5439453125	0.56640625	1.7724039375
2.1484375	2.3828125	0.13671875	2.48046875	2.5244140625	0.56640625	1.7431640625
2.1435340875	2.3828125	0.13671875	2.4756830375	2.5237100375	0.56640625	1.748046875
2.1435340875	2.3828125	0.1318359375	2.4853515625	2.529296875	0.576171875	1.748046875
2.1484375	2.3879539125	0.13671875	2.4853515625	2.5341798875	0.56640625	1.748046875
2.138671875	2.3828125	0.13671875	2.48046875	2.5244140625	0.5712890625	1.7431640625
2.1435340875	2.3828125	0.1318359375	2.48046875	2.5244140625	0.5712890625	1.748046875
2.1435340875	2.3828125	0.1318359375	2.48046875	2.5244140625	0.5712890625	1.7431640625
2.1484375	2.3828125	0.13671875	2.48046875	2.5244140625	0.5712890625	1.748046875
2.1484375	2.3828125	0.1318359375	2.48046875	2.5244140625	0.5712890625	1.748046875
2.1435340875	2.3828125	0.1318359375	2.48046875	2.5244140625	0.5712890625	1.7431640625
2.1435340875	2.3828125	0.1318359375	2.48046875	2.5244140625	0.5712890625	1.7431640625
2.1435340875	2.3828125	0.1318359375	2.48046875	2.5244140625	0.5712890625	1.7431640625
2.1435340875	2.3828125	0.1318359375	2.48046875	2.5244140625	0.5712890625	1.7431640625
2.1435340875	2.3828125	0.1318359375	2.48046875	2.5244140625	0.5712890625	1.7431640625
2.1435340875	2.3828125	0.1318359375	2.48046875	2.5244140625	0.5712890625	1.7431640625
2.1435340875	2.3828125	0.1318359375	2.48046875	2.5244140625	0.5712890625	1.7431640625
2.1435340875	2.3828125	0.1318359375	2.48046875	2.5244140625	0.5712890625	1.7431640625
2.1435340875	2.3828125	0.1318359375	2.48046875	2.5244140625	0.5712890625	1.7431640625
2.1435340875	2.3828125	0.1318359375	2.48046875	2.5244140625	0.5712890625	1.7431640625
2.1435340875	2.3828125	0.1318359375	2.48046875	2.5244140625	0.5712890625	1.7431640625
2.1435340875	2.3828125	0.1318359375	2.48046875	2.5244140625	0.5712890625	1.7431640625
2.1435340875	2.3828125	0.1318359375	2.48046875	2.5244140625	0.5712890625	1.7431640625



Apéndice B. Datos capturados de las pruebas experimentales

2.1184475	2.30678125	0.1318359375	2.490271375	2.57051125	0.571200625	1.7678125	
2.1158206125	2.302675125	0.1318359375	2.496231375	2.5690025	0.571289025	1.762665125	
2.118938125	2.307490375	0.13671875	2.4951171875	2.5690025	0.576171875	1.7626358125	
2.1158203125	2.307490375	0.13671875	2.5	2.5	2.5439451125	1.7626958125	
2.10796875	2.40254375	0.15671875	2.5048828125	2.548928125	0.5810496875	1.774009575	
2.10796875	2.4072265625	0.15671875	2.5048828125	2.5491625	0.576171875	1.7626958125	
2.1530850375	2.4024375	0.13671875	2.5	2.5	2.5537100375	0.576171875	1.77784375
2.1728515625	2.433109375	0.13671875	2.505705625	2.55850375	0.5810546875	1.77784375	
2.138671875	2.373048875	0.1318359375	2.470709125	2.51953125	0.5722890625	1.748066875	
2.1337800625	2.373048875	0.13671875	2.4705829375	2.5146484375	0.5712890625	1.748066875	
2.138671875	2.367693125	0.13671875	2.4755830625	2.51958125	0.5715890625	1.7626958125	
2.1533208125	2.367693125	0.1416016625	2.4651171875	2.5190025	0.576171875	1.7774638975	
2.1435540875	2.377293875	0.13671875	2.46946875	2.5244180625	0.5817578125	1.7549296875	
2.1543203125	2.367693125	0.13671875	2.4873515825	2.5341756875	0.576171875	1.7678125	
2.12899725	2.3681640625	0.13671875	2.4412500625	2.5048828125	0.5690625	1.7481640625	
2.1247254675	2.3583984375	0.1318359375	2.4560046875	2.5	0.56940625	1.79828125	
2.1240204375	2.3583984375	0.1318359375	2.4560046875	2.500766025	0.56940625	1.7401640625	
2.1337890625	2.36728125	0.1318359375	2.4560046875	2.5098828125	0.56940625	1.7401640625	
2.12899725	2.36338125	0.12671875	2.4693675	2.5048828125	0.5712890625	1.75828125	
2.12506625	2.35328125	0.1218959375	2.4693675	2.5048828125	0.56940625	1.7431640625	
2.1337890625	2.3581640625	0.1218959375	2.4658203125	2.509755625	0.5715291375	1.7431640625	
2.11980625	2.3981640625	0.1318359375	2.4658203125	2.509755625	0.5712890625	1.743099475	
2.1337890625	2.373048875	0.1318359375	2.470709125	2.5098828125	0.5712890625	1.748066875	
2.1307890625	2.373048875	0.136059125	2.4560046875	2.509755625	0.5712890625	1.748066875	
2.1337890625	2.373048875	0.120859125	2.470709125	2.51953125	0.576171875	1.7549296875	
2.1435540875	2.367693125	0.13671875	2.48048875	2.5244180625	0.576171875	1.7678125	
2.1758015625	2.4024375	0.13671875	2.5146484375	2.5631705625	0.5810546875	1.787103875	
2.1728515625	2.412109375	0.13671875	2.509755625	2.55850375	0.5810546875	1.782235625	
2.1826171875	2.421875	0.1318359375	2.51953125	2.568253375	0.5866875	1.7816581875	
2.1875	2.431546875	0.13671875	2.5390625	2.607421675	0.595766125	1.808610625	
2.1728515625	2.431875	0.13671875	2.5146484375	2.53559375	0.576171875	1.782235625	
2.1875	2.431875	0.13671875	2.51953125	2.5694786625	0.581546875	1.8096403125	
2.16790875	2.421875	0.1313971875	2.51953125	2.5694786625	0.581546875	1.787103875	
2.1826171875	2.421875	0.13671875	2.51953125	2.5694786625	0.581546875	1.787103875	
2.1826171875	2.421875	0.13671875	2.5048828125	2.5694786625	0.5810546875	1.787103875	
2.177734375	2.4024375	0.1318359375	2.5059125	2.5694786625	0.5810546875	1.787103875	
2.1826171875	2.421875	0.13671875	2.5244180625	2.568529375	0.5839375	1.7919931875	
2.1826171875	2.4267578125	0.1569375	2.5244180625	2.578432875	0.5879375	1.7919931875	
2.1875	2.4267578125	0.1416016625	2.51953125	2.572421875	0.5850375	1.7818931875	
2.1923928125	2.421875	0.1316015625	2.55850375	2.668339375	0.5859375	1.787103875	
2.1826171875	2.431875	0.1316015625	2.51953125	2.668339375	0.5810546875	1.796875	
2.177734375	2.4169931875	0.13671875	2.5146484375	2.669170325	0.5859375	1.77784375	
2.177734375	2.4267578125	0.15671875	2.51953125	2.63851375	0.5810546875	1.787103875	
2.177734375	2.4169931875	0.14684375	2.5146484375	2.65853575	0.5810546875	1.787103875	
2.1533203125	2.397609475	0.14084375	2.490234375	2.6341796875	0.576171875	1.7678125	
2.1484375	2.387609475	0.13571875	2.49684375	2.629206875	0.576171875	1.762695125	
2.1494875	2.387609475	0.13671875	2.4853515625	2.629206875	0.5810546875	1.762695125	
2.1484375	2.387609475	0.13671875	2.4853515625	2.629206875	0.576171875	1.762695125	
2.1333203125	2.387609475	0.13671875	2.48916875	2.6341796875	0.576171875	1.7678125	
2.1484375	2.387609475	0.13671875	2.48916875	2.629206875	0.576171875	1.762695125	
2.1484375	2.387609475	0.1171875	2.4853515625	2.629206875	0.576171875	1.762695125	
2.143664875	2.3828125	0.121165375	2.48046875	2.51953125	0.5712890625	1.762695125	
2.1377890625	2.3681640625	0.13671875	2.48046875	2.4651171875	0.5712890625	1.7529296875	
2.1337890625	2.3681640625	0.13671875	2.4658203125	2.500766025	0.5712890625	1.7529296875	
2.1337890625	2.373606875	0.1400136625	2.470709125	2.509755625	0.5712890625	1.7529296875	
2.1337890625	2.373606875	0.13671875	2.4658203125	2.5146484375	0.576171875	1.7626958125	
2.1337890625	2.373606875	0.1318359375	2.470709125	2.51953125	0.576171875	1.7626958125	
2.1337890625	2.377896875	0.13671875	2.4858203125	2.51953125	0.5712890625	1.7529296875	
2.1337890625	2.373606875	0.1318359375	2.470709125	2.603735625	0.5712890625	1.7529296875	
2.138671875	2.373606875	0.13671875	2.4858203125	2.5146484375	0.576171875	1.7529296875	
2.1337890625	2.373606875	0.1318359375	2.4658203125	2.5244180625	0.5712890625	1.7578125	





### B.2. Amoniaco: Matriz "A"

1.0658390635	1.7529296875	1.6796875	1.6162109375	1.9677734375	0.56640625	0.7421875
1.8552734375	1.748016875	1.6943359375	1.611328125	1.9671734375	0.566430625	0.732621875
1.6856390625	1.738046875	1.73428125	1.6092188375	1.9677734375	0.56649625	0.72265625
1.6658390625	1.7326396875	1.7296528125	1.6103188375	1.962390625	0.5712890625	0.712890625
1.699921875	1.7600296875	1.738671875	1.6162109375	1.992390625	0.5712890625	0.703125
1.6836390625	1.7520296875	1.72851509375	1.6162109375	1.932390625	0.56640625	0.693330875
1.6679290625	1.7529296875	1.7041010625	1.616109375	1.9577734375	0.56640625	0.7171734375
1.6650390625	1.7529296875	1.7481840625	1.611328125	1.9677734375	0.56640625	0.705953125
1.6858390625	1.7634984375	1.7333984375	1.6162109375	1.9677734375	0.5712890625	0.673828125
1.6650390625	1.7584306875	1.60021875	1.611328125	1.9677734375	0.576171875	0.6787109375
1.6658390625	1.748040875	1.629650125	1.611328125	1.977590625	0.56640625	0.717010125
1.6650390625	1.748040875	1.7333984375	1.6162109375	1.9677734375	0.56640625	0.70171875
1.6802125	1.748046875	1.71875	1.611328125	1.9677734375	0.56640625	0.6787109375
1.6650390625	1.732896875	1.71875	1.611328125	1.9677734375	0.56640625	0.68333375
1.6650390625	1.7526396875	1.7330328125	1.611328125	1.962496625	0.56640625	0.6787109375
1.6650390625	1.7529296875	1.738428125	1.611328125	1.962390625	0.56640625	0.712890625
1.6650390625	1.7529296875	1.6748046875	1.611328125	1.97296025	0.56640625	0.703125
1.6650390625	1.7629296875	1.704384375	1.6162109375	1.9677734375	0.56640625	0.65333375
1.6650390625	1.7529296875	1.689450125	1.6162109375	1.962390625	0.56640625	0.713125
1.6650390625	1.748046875	1.5845703125	1.611328125	1.9677734375	0.56640625	0.68333375
1.6650390625	1.7529296875	1.71875	1.611328125	1.9677734375	0.56640625	0.689453125
1.6650390625	1.7529296875	1.694453125	1.6162109375	1.9677734375	0.56640625	0.68333375
1.6650390625	1.7529296875	1.74326430625	1.611328125	1.9677734375	0.56640625	0.701553125
1.6650390625	1.7529296875	1.69821875	1.611328125	1.9677734375	0.5712890625	0.6787109375
1.6650390625	1.7578125	1.609921875	1.611328125	1.9677734375	0.5617878125	0.708125
1.6650390625	1.7529296875	1.69651875	1.6162109375	1.9677734375	0.5712890625	0.669468125
1.66015625	1.7529296875	1.71875	1.611328125	1.9677734375	0.56640625	0.7080078125
1.589921875	1.7529296875	1.748046875	1.611328125	1.9677734375	0.570171875	0.69333375
1.6650390625	1.748046875	1.6796875	1.611328125	1.962890625	0.56640625	0.7171734375
1.6650390625	1.7529296875	1.6845703125	1.611328125	1.962890625	0.5712890625	0.72265625
1.66015625	1.7529296875	1.69921875	1.6162109375	1.9677734375	0.5712890625	0.72265625
1.6650390625	1.7529296875	1.72851509375	1.611328125	1.9677734375	0.56640625	0.72265625
1.66015625	1.7529296875	1.72851509375	1.611328125	1.9677734375	0.5712890625	0.63333375
1.6650390625	1.7529296875	1.680453125	1.611328125	1.9677734375	0.5712890625	0.7275390625
1.6650390625	1.7529296875	1.7431010625	1.611328125	1.9677734375	0.56640625	0.723125
1.6650390625	1.7529296875	1.708984375	1.611328125	1.9677734375	0.5712890625	0.6787109375
1.6650390625	1.7529296875	1.71875	1.611328125	1.962890625	0.56640625	0.72265625
1.6650390625	1.7529296875	1.708984375	1.611328125	1.9677734375	0.5712890625	0.747078125
1.6650390625	1.7529296875	1.69921875	1.611328125	1.9677734375	0.5712890625	0.7373046875
1.6650390625	1.7529296875	1.689453125	1.611328125	1.97296025	0.5712890625	0.692451875
1.6650390625	1.7529296875	1.71875	1.6162109375	1.9677734375	0.5712890625	0.69333375
1.6650390625	1.7529296875	1.70821875	1.611328125	1.9677734375	0.60040625	0.679828125
1.6650390625	1.7578125	1.708984375	1.6054453125	1.9677734375	0.5712890625	0.7080078125
1.6650390625	1.7529296875	1.689453125	1.611328125	1.9677734375	0.5712890625	0.732421875
1.6650390625	1.7529296875	1.7333984375	1.611328125	1.9677734375	0.5712890625	0.751934125
1.609921875	1.7529296875	1.708984375	1.611328125	1.9677734375	0.5713890625	0.7373046875
1.6650390625	1.7529296875	1.7341010625	1.611328125	1.962890625	0.5712890625	0.688170625
1.6650390625	1.7529296875	1.6943359375	1.611328125	1.9677734375	0.5712890625	0.7373046875
1.6650390625	1.7529296875	1.6650390625	1.611328125	1.9677734375	0.5712890625	0.72265625
1.6650390625	1.7529296875	1.7041010625	1.611328125	1.9677734375	0.5712890625	0.7275390625
1.6650390625	1.7529296875	1.7138671875	1.611328125	1.9677734375	0.5712890625	0.712890625
1.6650390625	1.7529296875	1.630453125	1.611328125	1.9677734375	0.56640625	0.709378125
1.6650390625	1.7529296875	1.7451010625	1.611328125	1.9077734375	0.5712890625	0.6680453125
1.6650390625	1.7529296875	1.72851509375	1.611328125	1.9677734375	0.56640625	0.6680453125
1.6650390625	1.7529296875	1.69921875	1.611328125	1.9677734375	0.5713890625	0.747078125
1.6650390625	1.7529296875	1.68921875	1.611328125	1.9677734375	0.56640625	0.67824125
1.6650390625	1.7529296875	1.7138671875	1.604453125	1.9677734375	0.5712890625	0.7275390625
1.6650390625	1.7529296875	1.7343359375	1.6094453125	1.9677734375	0.5712890625	0.7171734375
1.66015625	1.7529296875	1.71875	1.6054453125	1.9677734375	0.5712890625	0.6787109375







Apéndice B. Datos capturados de las pruebas experimentales

1.6748046875	1.7578125	1.7290328125	1.6015625	1.9775390625	0.576171875	0.703125
1.6748046875	1.7578125	1.69021875	1.6015625	1.9775390625	0.576171875	0.684765625
1.6650390625	1.7578125	1.7138671875	1.6015625	1.9775390625	0.576171875	0.6884765625
1.660991875	1.752635625	1.7138671875	1.6015625	1.9775390625	0.576171875	0.7275390625
1.660991875	1.7578125	1.73828125	1.6015625	1.9775390625	0.576171875	0.6884765625
1.6748046875	1.7578125	1.7333084375	1.6015625	1.9775390625	0.576171875	0.7170703125
1.6748046875	1.7578195	1.7343984375	1.6015625	1.9775390625	0.576171875	0.6982421875
1.6748046875	1.7578125	1.708084375	1.6015625	1.9775390625	0.576171875	0.7565390625
1.6748046875	1.7578125	1.720328125	1.6015625	1.9775390625	0.576171875	0.693369375
1.6748046875	1.7578125	1.689453125	1.6015625	1.9775390625	0.576171875	0.7508459375
1.6748046875	1.7020953125	1.689453125	1.600790625	1.97265625	0.576171875	0.701125
1.6748046875	1.7578125	1.719040625	1.600790625	1.9775390625	0.576171875	0.693931875
1.602921875	1.7578125	1.723828125	1.5015625	1.9775390625	0.576171875	0.6889453125
1.6748046875	1.7578125	1.6845703125	1.5015625	1.9775390625	0.576171875	0.68350625
1.669321875	1.7578125	1.6845703125	1.5015625	1.9775390625	0.576171875	0.7086078125
1.6748046875	1.702993125	1.704103625	1.6018625	1.982421875	0.576171875	0.7470703125
1.689021875	1.7578125	1.669071875	1.6018625	1.9775390625	0.576171875	0.693369375
1.6748046875	1.7578125	1.69921875	1.5966790625	1.9775390625	0.576171875	0.6889453125
1.6748046875	1.7578125	1.7198671875	1.6018625	1.9775390625	0.576171875	0.717734375
1.6748046875	1.7578125	1.711640625	1.600445125	1.9775390625	0.576171875	0.6889453125
1.6790875	1.7028053125	1.6788875	1.5000790625	1.97205625	0.5810540625	0.6889453125
1.669921875	1.7578125	1.723828125	1.6015625	1.9775390625	0.576171875	0.7573046875
1.6748046875	1.7578125	1.71875	1.6015625	1.9775390625	0.576171875	0.717734375
1.601921875	1.7578125	1.728418625	1.6015625	1.9775390625	0.576171875	0.712809375
1.6790875	1.7578125	1.70551175	1.5066790625	1.9775390625	0.576171875	0.6889453125
1.6790875	1.7578125	1.69921875	1.5000790625	1.9775390625	0.576171875	0.693369375
1.6748046875	1.693265625	1.725615625	1.5036790625	1.9775390625	0.5810540625	0.7565390625

B.3. Metano: Matriz "M"

1.7138671875	1.826171875	1.7578125	1.7724609375	2.01171875	0.537109375	0.654206875
1.7138671875	1.8310546875	1.801734375	1.7724609375	2.016603625	0.537109375	0.629828125
1.7138671875	1.8310546875	1.83835625	1.7794609375	2.0171875	0.5322265625	0.7117734375
1.7138671875	1.826171875	1.811324375	1.7724609375	2.016603625	0.5322265625	0.782921875
1.7138671875	1.8312690625	1.7794609375	1.7724609375	2.016603625	0.537109375	1.68350625
1.7138671875	1.8310546875	1.7578125	1.7724609375	2.01171875	0.5322265625	0.7273890625
1.7138671875	1.8310546875	1.79828125	1.7724609375	2.016603625	0.5322265625	0.698459375
1.7138671875	1.826171875	1.7724609375	1.7724609375	2.01171875	0.537109375	0.693369375
1.7138671875	1.826171875	1.7823265625	1.7724609375	2.01171875	0.537109375	0.693369375
1.7138671875	1.8310546875	1.7578125	1.7724609375	2.016603625	0.5322265625	0.678759375
1.7138671875	1.8310546875	1.79828125	1.7724609375	2.016603625	0.537109375	0.684765625
1.7138671875	1.8310546875	1.79828125	1.7724609375	2.016603625	0.537109375	0.7273890625
1.7138671875	1.826171875	1.811324375	1.767734375	2.016603625	0.537109375	0.698459375
1.7138671875	1.826171875	1.7734375	1.7724609375	2.01171875	0.5322265625	0.8101706875
1.7138671875	1.826171875	1.7734375	1.7724609375	2.01171875	0.537109375	0.717734375
1.7138671875	1.8310546875	1.7916921875	1.7724609375	2.01171875	0.537109375	0.703125
1.7236328125	1.826171875	1.709953125	1.7724609375	2.01171875	0.537109375	0.693369375
1.7138671875	1.826171875	1.7631640625	1.7724609375	2.01171875	0.5322265625	0.654206875
1.7138671875	1.826171875	1.787109375	1.767734375	2.016603625	0.537109375	0.70109375
1.7138671875	1.826171875	1.801734375	1.7724609375	2.016603625	0.537109375	0.703125
1.7138671875	1.8310546875	1.7578125	1.7724609375	2.01171875	0.5420991875	0.72355625
1.7138671875	1.826171875	1.798875	1.767734375	2.01171875	0.537109375	0.625
1.7138671875	1.826171875	1.7631640625	1.767734375	2.01171875	0.5322265625	0.712809375
1.7138671875	1.8369375	1.7578125	1.767734375	2.016603625	0.537109375	0.693369375
1.7138671875	1.8310546875	1.723828125	1.767734375	2.01171875	0.537109375	0.678759375
1.7138671875	1.826171875	1.8043359375	1.767734375	2.01171875	0.537109375	0.6889453125
1.728684375	1.826171875	1.7236328125	1.767734375	2.01171875	0.537109375	0.787109375
1.7138671875	1.826171875	1.6650390625	1.767734375	2.01171875	0.5322265625	0.68350625







Apéndice B. Datos capturados de las pruebas experimentales

1.713864375	1.8212890025	1.806640025	1.7026938125	2.01271875	0.541992875	0.654266875
1.713864375	1.8212890025	1.7626938125	1.7026938125	2.01171875	0.541992875	0.6398451375
1.713864375	1.8212890025	1.81840025	1.698553125	2.0068350375	0.541992875	0.6945729375
1.713864375	1.8212890025	1.795875	1.7026938125	2.01171875	0.541992875	0.625
1.70884375	1.8212890025	1.806640025	1.7578125	2.0108350375	0.537109375	0.6940825
1.70884375	1.8212890025	1.806640025	1.7026938125	2.01171875	0.537109375	0.68359375
1.70884375	1.8212890025	1.806640025	1.7578125	2.01171875	0.541992875	0.699421875
1.70884375	1.8212890025	1.806640025	1.7578125	2.0068350375	0.537109375	0.7421875
1.70884375	1.8212890025	1.792226625	1.763963825	2.008359375	0.537109375	0.712890625
1.70884375	1.8212890025	1.767578125	1.7578125	2.0068350375	0.537109375	0.693309375
1.713864375	1.8212890025	1.7626938125	1.7578125	2.01171875	0.537109375	0.706078125
1.70884375	1.8212890025	1.7431640625	1.7578125	2.01171875	0.537109375	0.6201790625
1.713864375	1.8212890025	1.8212890025	1.7026938125	2.0068350375	0.537109375	0.654299875
1.70884375	1.8212890025	1.836171875	1.7578125	2.0068350375	0.541992875	0.625
1.70884375	1.8212890025	1.787109375	1.7578125	2.0068350375	0.537109375	0.72306025
1.70884375	1.8212890025	1.782226625	1.7578125	2.01171875	0.537109375	0.6940825
1.70884375	1.8212890025	1.767578125	1.7578125	2.015015025	0.541992875	0.727089025
1.70884375	1.8212890025	1.782226625	1.7578125	2.01171875	0.541992875	0.654299875
1.713864375	1.8212890025	1.806640025	1.7578125	2.01171875	0.537109375	0.65359375
1.70884375	1.8212890025	1.8017578125	1.7578125	2.0068350375	0.537109375	0.688165625
1.70884375	1.8212890025	1.60060625	1.7026938125	2.01171875	0.537109375	0.628828125
1.7041015625	1.8212890025	1.8017578125	1.7578125	2.01171875	0.541992875	0.704125
1.70884375	1.8212890025	1.7774375	1.7578125	2.01171875	0.537109375	0.704125
1.70884375	1.8212890025	1.798515625	1.7578125	2.01171875	0.537109375	0.654299875
1.70884375	1.8212890025	1.733984375	1.7578125	2.009031375	0.541992875	0.7121875
1.70884375	1.8212890025	1.7578125	1.7578125	2.01171875	0.541992875	0.68359375
1.70884375	1.8212890025	1.7626938125	1.7578125	2.01171875	0.541992875	0.68359375
1.70884375	1.8212890025	1.792226625	1.7578125	2.0068350375	0.537109375	0.77774875
1.70884375	1.8212890025	1.748406875	1.7578125	2.01171875	0.537109375	0.6689453125
1.70884375	1.8212890025	1.796875	1.7529296875	2.01171875	0.541992875	0.708125
1.70884375	1.8212890025	1.8017578125	1.7529296875	2.010015025	0.541992875	0.65359375
1.70884375	1.8212890025	1.772226625	1.7578125	2.0068350375	0.541992875	0.6940825
1.70884375	1.8212890025	1.752929625	1.7578125	2.0068350375	0.541992875	0.704125
1.70884375	1.8212890025	1.73328125	1.7578125	2.01171875	0.541992875	0.727089025
1.709984375	1.8212890025	1.8311578125	1.7529296875	2.01171875	0.537109375	0.688470625
1.70884375	1.8212890025	1.80060625	1.7578125	2.0068350375	0.537109375	0.6689453125
1.70884375	1.8212890025	1.78828125	1.7578125	2.0068350375	0.537109375	0.68359375
1.70884375	1.8212890025	1.7774375	1.7529296875	2.01171875	0.537109375	0.688470625
1.70884375	1.8212890025	1.733984375	1.7529296875	2.0068350375	0.541992875	0.673828125
1.70884375	1.8212890025	1.70884375	1.7578125	2.001853125	0.537109375	0.73212875
1.70884375	1.8212890025	1.748406875	1.7426296875	2.0068350375	0.541992875	0.6306484375
1.70884375	1.8212890025	1.80060625	1.7578125	2.0068350375	0.537109375	0.737520025
1.7041015625	1.8212890025	1.733984375	1.7529296875	2.0068350375	0.537109375	0.712875
1.70884375	1.8212890025	1.76644875	1.7578125	2.01171875	0.541992875	0.6691706875
1.70884375	1.8212890025	1.8212890025	1.7529296875	2.01171875	0.546875	0.6854764025
1.70884375	1.8212890025	1.748406875	1.7529296875	2.01171875	0.537109375	0.704125
1.7041015625	1.8212890025	1.7724000375	1.7578125	2.01171875	0.537109375	0.6787100375
1.70884375	1.8212890025	1.748406875	1.7529296875	2.01171875	0.541992875	0.727089025
1.70884375	1.8212890025	1.7774375	1.7529296875	2.01171875	0.537109375	0.654266875
1.98084375	1.8212890025	1.7578125	1.7529296875	2.0058559375	0.537109375	0.698399375
1.70884375	1.8212890025	1.7578125	1.7529296875	2.010015025	0.537109375	0.7177734875
1.7041015625	1.8212890025	1.713984575	1.7529296875	2.0068350375	0.541992875	0.665458125
1.70884375	1.8212890025	1.791069375	1.7529296875	2.0068350375	0.537109375	0.6643625
1.7041015625	1.8212890025	1.73828125	1.7529296875	2.0068350375	0.537109375	0.7177734875
1.70884375	1.8212890025	1.796875	1.7529296875	2.0068350375	0.541992875	0.64433125
1.7041015625	1.8212890025	1.748406875	1.7529296875	2.0068350375	0.537109375	0.654266875
1.7041015625	1.8212890025	1.787109375	1.7529296875	2.0068350375	0.541992875	0.6787100375
1.70884375	1.8212890025	1.733984375	1.7529296875	2.0068350375	0.537109375	0.678928125
1.70884375	1.8212890025	1.7626938125	1.7529296875	2.0068350375	0.541992875	0.6689453125
1.70884375	1.8212890025	1.792226625	1.7529296875	2.0068350375	0.541992875	0.688470625
1.70884375	1.8212890025	1.733984375	1.7529296875	2.0068350375	0.541992875	0.654266875
1.70884375	1.8212890025	1.7529296875	1.7529296875	2.0068350375	0.541992875	0.688470625
1.70884375	1.8212890025	1.748406875	1.7529296875	2.0068350375	0.541992875	0.694140625



Apéndice B. Datos capturados de las pruebas experimentales

1.7041015625	1.8217890625	1.7246128125	1.7529296875	2.0668359375	0.54110921875	0.6693529375
1.708984375	1.8222890625	1.71734375	1.7529296875	2.01171875	0.54110921875	0.7476798125
1.708984375	1.8115234375	1.8212890625	1.7529296875	2.0668359375	0.54110921875	0.6884785625
1.708984375	1.81640025	1.7431640625	1.7529296875	2.0668359375	0.54110921875	0.709125
1.708984375	1.81640025	1.7910921875	1.7529296875	2.01171875	0.537109375	0.693350375
1.708984375	1.81640025	1.7236235125	1.7529296875	2.0368359375	0.54110921875	0.693
1.708984375	1.81640025	1.7724806375	1.7529296875	2.0068359375	0.54110921875	0.6040025
1.708984375	1.81640025	1.7534006375	1.748046875	2.01171875	0.54110921875	0.614140025
1.708984375	1.81640025	1.733898375	1.748046875	2.0068359375	0.540875	0.717754875
1.708984375	1.81640025	1.767578125	1.748046875	2.01171875	0.54110921875	0.6787109375
1.708984375	1.81640025	1.7578125	1.7529296875	1.9548359375	0.54110921875	0.6531796875
1.7041015625	1.81640025	1.796875	1.748046875	2.0668359375	0.54110921875	0.6983766025
1.708984375	1.81640025	1.753858125	1.7529296875	2.0668359375	0.54110921875	0.682423875
1.708984375	1.8115234375	1.7610921875	1.7529296875	2.09103125	0.54110921875	0.713890625
1.7041015625	1.81640025	1.753898375	1.7529296875	2.0668359375	0.537109375	0.5982421875
1.7041015625	1.81640025	1.7724806375	1.748046875	2.0668359375	0.54110921875	0.693350375
1.708984375	1.81640025	1.7178125	1.7529296875	2.01171875	0.54110921875	0.66369375
1.7041015625	1.8115234375	1.77734375	1.7529296875	2.01171875	0.54110921875	0.68353375
1.7041015625	1.81640025	1.7629296875	1.748046875	2.0068359375	0.537109375	0.6391796875
1.708984375	1.81640025	1.765084375	1.748046875	2.0068359375	0.54110921875	0.635
1.7041015625	1.81640025	1.762095625	1.7529296875	2.0068359375	0.54110921875	0.7175300025
1.7041015625	1.81640025	1.8017078125	1.748046875	2.0368359375	0.537109375	0.752421875
1.708984375	1.8115234375	1.78858125	1.748046875	2.068359375	0.54110921875	0.72260025
1.708984375	1.8115234375	1.7734006375	1.748046875	2.068359375	0.54110921875	0.693350375
1.7041015625	1.81640025	1.7529296875	1.748046875	2.068359375	0.54110921875	0.693350375
1.7041015625	1.8115234375	1.762095625	1.748046875	2.0078159375	0.54110921875	0.6404140025
1.7041015625	1.8115234375	1.7529296875	1.748046875	2.0078159375	0.54110921875	0.7080078125
1.7041015625	1.81640025	1.7529296875	1.748046875	2.0078159375	0.54110921875	0.673828125
1.708984375	1.81640025	1.74386125	1.748046875	2.0088359375	0.537109375	0.6298481375
1.708984375	1.81640025	1.743040625	1.748046875	2.0088359375	0.537109375	0.708078125
1.7041015625	1.81640025	1.7721660375	1.748046875	2.068359375	0.54110921875	0.709125
1.7041015625	1.81640025	1.7529296875	1.748046875	2.068359375	0.54110921875	0.6591796875
1.68498125	1.81640025	1.7822890625	1.748046875	2.068359375	0.54110921875	0.7080078125
1.7041015625	1.8115234375	1.7822890625	1.748046875	2.068359375	0.54110921875	0.693350375
1.7011015625	1.8115234375	1.7822890625	1.748046875	2.068359375	0.54110921875	0.6982421875
1.7011015625	1.8115234375	1.7822890625	1.748046875	2.068359375	0.54110921875	0.715421875
1.7041015625	1.81640025	1.787109375	1.7529296875	2.0088359375	0.54110921875	0.6319925
1.7041015625	1.81640025	1.787109375	1.748046875	2.0088359375	0.54110921875	0.693350375
1.7041015625	1.8115234375	1.7724806375	1.748046875	2.068359375	0.54110921875	0.693350375
1.7041015625	1.81640025	1.748046875	1.748046875	2.068359375	0.54110921875	0.722890625
1.7011015625	1.8115234375	1.8017078125	1.748046875	2.0088359375	0.54110921875	0.6787109375
1.7011015625	1.8115234375	1.767578125	1.748046875	2.068359375	0.537109375	0.673425125
1.7041015625	1.81640025	1.7822890625	1.748046875	2.068359375	0.54110921875	0.717754875
1.7041015625	1.8115234375	1.7578125	1.748046875	2.0088359375	0.540675	0.6601796875
1.708984375	1.8115234375	1.7528125	1.7435640625	2.0078159375	0.54110921875	0.6884765625
1.7041015625	1.81640025	1.7678125	1.748046875	2.068359375	0.54110921875	0.6690625
1.7041015625	1.81640025	1.787109375	1.748046875	2.068359375	0.54110921875	0.693350375
1.7041015625	1.8115234375	1.7822890625	1.748046875	2.068359375	0.54110921875	0.6983421875
1.7041015625	1.8115234375	1.7822890625	1.748046875	2.068359375	0.54110921875	0.717754875
1.7041015625	1.8115234375	1.7822890625	1.748046875	2.068359375	0.54110921875	0.717754875
1.7041015625	1.8115234375	1.7822890625	1.748046875	2.068359375	0.54110921875	0.6983421875
1.7041015625	1.81640025	1.7910921875	1.748046875	2.068359375	0.54110921875	0.6591796875
1.7041015625	1.81640025	1.7236235125	1.7431040025	2.068359375	0.54110921875	0.6591796875
1.7041015625	1.8115234375	1.748046875	1.7431040025	2.00953125	0.54110921875	0.6717754875
1.7041015625	1.81640025	1.7041015625	1.7431040025	2.00953125	0.54110921875	0.6717754875
1.7041015625	1.8115234375	1.787109375	1.7431040025	2.00953125	0.54110921875	0.6884765625
1.7041015625	1.8115234375	1.707878125	1.748046875	2.01171875	0.54110921875	0.6640025
1.7041015625	1.8115234375	1.796875	1.7431040025	2.008359375	0.54110921875	0.6640025
1.7041015625	1.81640025	1.748046875	1.7431040025	2.008359375	0.54110921875	0.725320625
1.69921875	1.8115234375	1.5017678125	1.7431040025	2.008359375	0.537109375	0.68239375
1.7041015625	1.8215234375	1.7078125	1.7431040025	2.06193125	0.5406875	0.6640025



B.4. Gas LP: Matriz "LP"

1.2800625	1.4453125	1.2039438125	1.6015625	1.31705625	1.5673828125	1.824125
1.5504120625	1.337500625	1.6015625	1.30559375	.5625	1.2690625	1.1969521875
1.250703025	1.4697265625	1.281790625	1.484375	1.239763625	1.4889578125	1.2158206125
1.29046875	1.484375	1.2109375	1.35871875	1.1767578125	1.4008010625	1.2060546875
1.1765675	1.364016875	1.46194075	1.1825384375	1.1662296875	1.171875	1.2232421875
1.16625009375	1.411328125	1.280708175	1.38671875	1.1368921875	1.220703125	1.18110625
1.552733375	1.2258839375	1.270290675	1.19140625	1.1741140625	1.147408875	1.26951125
1.2159263125	1.3144763675	1.24000375	1.303719375	1.150991875	1.1428125	1.1053854375
1.1613671875	1.2462117875	1.4060349375	1.2189375	1.408009575	1.201171875	1.4444396875
1.5383859375	1.215171875	1.4306640625	1.13879125	1.1111825125	1.1865234375	1.42578125
1.201171875	1.5178510625	1.2255669375	1.30559375	1.1668021875	1.1295421875	1.1689921875
1.906750875	1.4615671875	1.2368515625	1.4507058125	1.201172075	1.416915925	1.159757125
1.3283208125	1.201171875	1.4453125	1.1767578125	1.4111956125	1.1574009375	1.416905625
1.181510625	1.4309640625	1.18117825	1.194140625	1.177875	1.3037100375	1.1376838125
1.1865234375	1.328125	1.1174309375	1.4208394375	1.1279290875	1.4306640625	1.2369596875
1.484375	1.2265859375	1.42578125	1.1562826625	1.328125	1.1276953125	1.3016010625
1.070051125	1.1865234375	1.181640625	1.1572256625	1.201171875	1.129040875	1.225
1.40484375	1.1573206625	1.5294875	1.1375058125	1.533706125	1.191515025	1.420808375
1.1562890625	1.4507058125	1.2253839375	1.4160150625	1.62109675	1.4213671875	1.181640625
1.1185546875	1.39140625	1.474009375	1.1865234375	1.3815350625	1.1379656625	1.40025
1.40025	1.220703125	1.2421171875	1.19140625	1.1781640625	1.1276206875	1.36859375
1.5128558675	1.169021875	1.401678125	1.142578125	1.4893879125	1.171875	1.2660546875
1.5771484375	1.1767578125	1.5155566875	1.1688021875	1.1634783625	1.1579959125	1.3720703125
1.240703125	1.4306640625	1.20010875	1.350484875	1.1913834375	1.3134763625	1.1474609375
1.451076125	1.25	1.4453125	1.3060540675	1.1925734375	1.201171875	1.2109375
1.1962890625	1.201171875	1.2154206125	1.1828125	1.3255890625	1.144578125	1.3783625
1.4892878125	1.220703125	1.4941409375	1.201171875	1.3222421875	1.1474609375	1.2000546875
1.1474609375	1.444290875	1.1085984375	1.387421875	1.062890625	1.171875	1.171875
1.201171875	1.463548675	1.171875	1.355078125	1.162109375	1.4001290875	1.1865234375
1.4306640625	1.1902500625	1.42578125	1.261171875	1.208394375	1.40184375	1.15234075
1.1279290875	1.40025	1.1572265625	1.171875	1.2841796875	1.1338125	1.1474609375
1.1572265625	1.19010625	1.127875	1.2353859375	1.208428125	1.162109375	1.240171875
1.2744140625	1.2744140625	1.176903125	1.15294375	1.184610625	1.162109375	1.2640484375
1.171875	1.2016015625	1.353656375	1.318369375	1.1962890625	1.33948675	1.128346875
1.456078125	1.10140625	1.3403640625	1.220703125	1.474009375	1.1865234375	1.396381175
1.1865234375	1.3791783125	1.131171875	1.425546875	1.1865234375	1.416505625	1.1767578125
1.1999390675	1.261171875	1.4539608675	1.19140625	1.318350375	1.142578125	1.39671875
1.1665601875	1.404375	1.1662890625	1.1383944375	1.11338125	1.1672906625	1.112578125
1.4589609375	1.15140625	1.455546875	1.1895254375	1.1714740625	1.1474609375	1.196990625
1.1962890625	1.4268984375	1.1572265625	1.3525390625	1.14554075	1.298828125	1.1865234375
1.1767578125	1.20140625	1.162109375	1.3134763625	1.1572265625	1.2195375	1.1767578125
1.2646484375	1.474009375	1.09375	1.5155566875	1.2886671875	1.494140925	1.12340375
1.474009375	1.201171875	1.38125125	1.18140625	1.46010625	1.142578125	1.3184309375
1.1865234375	1.376953125	1.1572265625	1.306484375	1.12294375	1.310284875	1.1366921875
1.307890625	1.171875	1.15234675	1.1181940625	1.1672265625	1.109515625	1.2306540675
1.4897465625	1.2253859375	1.4501953125	1.3066546875	1.38671875	1.1767578125	1.376953125
1.50140625	1.19010625	1.1521640625	1.1683984375	1.1378653125	1.083984375	1.28033125
1.399484375	1.201171875	1.40025	1.2158203125	1.2253859375	1.1767578125	1.07421875
1.15234375	1.34763625	1.199921875	1.10025	1.15284375	1.3666918675	1.1572265625
1.1474609375	1.171875	1.220703125	1.17689875	1.2353515625	1.9791015625	1.2744140625
1.0988326125	1.2109375	1.296728125	1.20140625	1.2109375	1.1572265625	1.162109375
1.181640625	1.3330078125	1.1767578125	1.9720709125	1.1672265625	1.1697265625	1.1767578125
1.290953125	1.1767578125	1.176953125	1.1865234375	1.14078125	1.4506840625	1.103616625
1.3109375	1.1194765625	1.3063640675	1.3871875	1.162109375	1.430540875	1.1689921875
1.1767578125	1.1841790875	1.201171875	1.4013671875	1.171875	1.1400640625	1.1767578125
1.19140625	1.513671875	1.38098125	1.489678125	1.181640625	1.176953125	1.1474609375
1.4111338125	1.2253859375	1.1404296875	1.201171875	1.1141640625	1.1572265625	1.1009921875
1.4607265625	1.171875	1.2283206125	1.3865234375	1.4453125	1.103021875	1.40025
1.484375	1.2253859375	1.482578125	1.181640625	1.4258984375	1.15134375	1.1404296875
1.1767578125	1.0889671875	1.313594375	1.1328125	1.1474609375	1.3646484375	1.1328125



Apéndice B. Datos capturados de las pruebas experimentales

1.33209125	1.3816915025	1.3960516875	1.4013071875	1.171875	1.396484375	1.1870953125
1.3109375	1.3720763125	1.38171875	1.38859375	1.1835234375	1.133046875	1.2890625
1.291171875	1.435781375	1.382109375	1.3730708125	1.187510625	1.376053125	1.1767876125
1.181640625	1.250705375	1.169921875	1.3720708125	1.166092875	1.378953125	1.129046875
1.284796875	1.1474008375	1.393181875	1.1609921875	1.19110625	1.1870653125	1.1883234375
1.291171875	1.181630625	1.1757578125	1.199346875	1.2263830375	1.1829125	1.406375
1.171875	1.386481875	1.1757578125	1.3333078125	1.1111328125	1.09875	2.51567875
1.178350375	1.245171875	1.169921875	1.1707578125	1.083854375	1.1379453125	1.1328125
1.2109375	1.4453125	1.109375	1.174609375	1.171875	1.176053125	1.12578125
1.4892578125	1.2109375	1.1693904125	1.19110625	1.4392066375	1.182106675	1.3330875125
1.1767876125	1.410018925	1.171875	1.174609375	1.15281875	1.435546875	1.219375
1.4111028125	1.291171875	1.184375	1.2003516875	1.2953915625	1.155046875	1.247144375
1.2572265625	1.166694875	1.335705625	1.181643625	1.34789125	1.18294375	1.123893125
1.15944375	1.120594875	1.1062800625	1.25	1.129046875	1.2318924125	1.15214375
1.1767876125	1.259785625	1.23046875	1.201171875	1.3158263125	1.189084375	1.290708125
1.1171099375	1.4268984375	1.1866244375	1.1897265625	1.1503781875	1.46484375	1.184324375
1.2109375	1.1828125	1.328125	1.171875	1.254703625	1.15254375	1.2300946875
1.178350375	1.3336325	1.169921875	1.2361171875	1.1781640625	1.1306610625	1.15234375
1.19140625	1.2158205125	1.1767876125	1.1279266875	1.1066921875	1.1000230625	1.25
1.1093234375	1.3359516625	1.09110625	1.23046875	1.453378125	1.1002800625	1.337890625
1.2109375	1.142578125	1.117409375	1.245171875	1.15294375	1.17140625	1.142078125
1.357421875	1.199280625	1.17586625	1.18109875	1.278296875	1.117409375	1.416615625
1.38982125	1.196289625	1.3330278125	1.181643625	1.4013671875	1.1572265625	1.4111338125
1.201171875	1.186524375	1.18140625	1.1328125	1.162109375	1.1093550375	1.123016875
1.1069921875	1.4111098125	1.109375	1.49410625	1.181640625	1.155678125	1.168993875
1.2109375	1.106375	1.22970125	1.176053125	1.1883234375	1.18031125	1.168994875
1.181640625	1.3523390625	1.181640625	1.191603625	1.181640625	1.229703125	1.1828125
1.2004940375	1.1079206625	1.250705325	1.109816625	1.2421764875	1.123168875	1.396484375
1.3818339375	1.186824375	1.34753625	1.17931875	1.181640625	1.15261575	1.083984375
1.171875	1.15334375	1.171875	1.1086428125	1.1369234375	1.245171875	1.131610625
1.1806643625	1.15234375	1.39688375	1.1434699375	1.1316473625	1.1379843125	1.39688375
1.171875	1.1427754375	1.42578125	1.26953125	1.1181640625	1.157121875	1.1609921875
1.1427754375	1.0888071875	1.357521875	1.383084375	1.1307139875	1.106443125	1.39819375
1.102109375	1.628125	1.22070125	1.1572265625	1.464296875	1.103515625	1.176015625
1.13009875	1.2841796875	1.2548826125	1.3939491875	1.2109375	1.2841796875	1.320703125
1.171609375	1.259785625	1.1474800375	1.2518828125	1.176993125	1.2158208125	1.2101171875
1.1962896625	1.109375	1.171875	1.1083084375	1.1001984375	1.109375	1.1767876125
1.29046875	1.4390609375	1.181640625	1.42578125	1.1579459375	1.3781798125	1.225889375
1.494140625	1.3903516875	1.4892578125	1.4060546875	1.411528125	1.162109375	49578125
1.142578125	1.18230491875	1.171875	1.1279266875	1.182999875	1.16625	1.3818339375
1.1865234375	1.246234375	1.1965234375	1.393949125	1.147409375	1.2016016625	1.1572265625
1.1990234375	1.25	1.337996625	1.1407899375	1.18339375	1.17834375	1.34796625
1.166921875	1.201171875	1.1376053125	1.2348828125	1.1328125	1.147956625	1.132109375
1.2109375	1.3320078125	1.1314765625	1.3037109375	1.261171875	1.1758208125	1.1328125
1.4892578125	1.2158203125	1.489909875	1.1562800625	1.439078125	1.291171875	1.403071875
1.181640625	1.474609375	1.1699921875	1.2537516625	1.083867575	1.36836875	1.1572265625
1.15334375	1.48540875	1.240234375	1.3823046875	1.19140625	1.3791105625	1.4999234375
1.171875	1.162109375	1.15234375	1.20963125	1.2548838125	1.30830375	1.142578125
1.1962896625	1.1328125	1.240234375	1.1181640625	1.1672366625	1.2253850375	1.229703125
1.2354316625	1.2158203125	1.49404375	1.206468875	1.197093125	1.1707878125	1.0888671875
1.30859375	1.1962896625	1.4913050625	1.17978125	1.2548838125	1.1062800625	1.201171875
1.171875	1.1934453125	1.1865234375	1.2848828125	1.1572265625	1.4111338125	1.1767876125
1.4892578125	1.326793125	1.1561953125	1.109375	1.337596625	1.184324375	1.2353516625
1.103815625	1.171875	1.0595798125	1.08516625	1.023986625	0.95769125	1.142578125
1.6502734375	1.1669921875	1.181640625	1.137093125	1.095926125	1.1572265625	1.09375
1.15334375	1.23046875	1.1069921875	1.169315625	1.1695234375	1.10830375	1.1181640625
1.4111338125	1.2109375	1.3790783125	1.181640625	1.188309375	1.181640625	1.17926875
1.2902896625	1.439078125	1.2158203125	1.48484375	1.109375	1.4111338125	1.186234375
1.4892578125	1.206840875	1.4999098625	1.1767876125	1.416023625	1.26171875	1.193515625
1.15140625	1.474609375	1.169375	1.1181640625	1.141378125	1.58421875	1.291171875
1.2548838125	1.337890625	1.2833516625	1.185678125	1.340234375	1.328125	1.10140825
1.4892578125	1.2538203125	1.4111338125	1.19140625	1.3427754375	1.117409375	1.4111338125



Apéndice B. Datos capturados de las pruebas experimentales

1.12346875	1.1200040025	1.1474603075	1.066484375	1.1328125	1.0074375	1.123046875
1.2766546875	1.3525490625	1.281646225	1.20157875	1.308125	1.137635125	1.1181640025
1.1307265625	1.3548828125	1.377942875	1.25	1.3525300625	1.240291875	1.244234375
1.2158703125	1.137635125	1.304740625	1.174675	1.318560375	1.1572267625	1.0693579375
1.1767578125	1.3730703125	1.4637265625	1.09375	1.490234375	1.103515625	1.54206875
1.10140625	1.4308784375	1.181640625	1.378953125	1.13140625	1.411392125	1.258203125
1.4097205625	1.239765625	1.142678125	1.4892578125	1.133615625	1.370963125	1.220706125
1.2000346875	1.5087890625	1.20174875	1.4607265625	1.2060543675	1.4111328125	1.786524475
1.408125	1.381640325	1.4306641625	1.25234375	1.4208984375	1.182169375	1.2741146625
1.186523375	1.2329421875	1.3726736125	1.0086328125	1.4208984375	1.1528125	1.5243303125
1.1062800625	1.201001375	1.220703125	1.4433125	1.40484375	1.186640625	1.513571875
1.3279731875	1.2266341875	1.533203125	1.1363234375	1.463125	1.174800375	1.453078125
1.306484375	1.1865234375	1.4607265625	1.1707578125	1.40578125	1.147460375	1.443125
1.2063546875	1.4015671875	1.386323125	1.3191705625	1.142575125	1.147460375	1.87004625
1.15140625	1.04765625	1.1982953625	1.3628046875	1.373265625	1.3232921875	1.181640625
1.4509433125	1.201171875	1.345117575	1.1092803625	1.25	1.171575	1.206484375
1.376951125	1.142578125	1.243174875	1.2158303125	1.330078125	1.171575	1.2548625125
1.0962800625	1.173609375	1.2744140625	1.1083584375	1.40434375	1.103515625	1.30484375
1.4111578125	1.2265890625	1.328125	1.2660646875	1.2003546575	1.1572265625	1.053984675
1.235255625	1.1962890625	1.173875	1.2109375	1.1181640625	1.2841790625	1.367093125
1.1062800625	1.411328125	1.0940625	1.450106125	1.1572265625	1.343160375	1.000921875
1.1865234375	1.4208984375	1.1066921875	1.40625	1.362109375	1.357422875	1.0109375
1.484375	1.2109375	1.1794921875	1.10145625	1.58071875	1.150692875	1.461387575
1.10140625	1.340254375	1.137695125	1.4589040625	1.1181640625	1.4404290625	1.1767578125
1.2558203125	1.3530078125	1.1572265625	1.315359375	1.15231975	1.26033125	1.1707578125
1.226703125	1.20171875	1.3406375	1.342774475	1.171875	1.2109375	1.142578125
1.105515025	1.147460375	1.396484375	1.171875	1.3858950375	1.1038125	1.174000375
1.2744140625	1.64206875	1.324125	1.5478515625	1.3427734375	1.562734375	1.0083075
1.245171875	1.2714140625	1.4111328125	1.270206375	1.303816625	1.30850375	1.3726703125
1.279296875	1.2151171875	1.2744140625	1.25	1.38671875	1.10140625	1.30830675
1.396484375	1.2744140625	1.25646875	1.1953890625	1.25	1.1572265625	1.245171875
1.201171875	1.4506640625	1.386234375	1.4530640625	1.23703125	1.4010015025	1.3060540875
1.376951125	1.123016875	1.2841790625	1.1865234375	1.1053984375	1.2099316875	1.2109375
1.1892578125	1.19140625	1.425546875	1.1902890625	1.414615625	1.171875	1.425446875
1.142734475	1.2000346875	1.23056875	1.1767578125	1.0003950375	1.147460375	1.2060540875
1.1767578125	1.25	1.171875	1.2916012625	1.143578125	1.416015625	1.1402934375
1.153078125	1.226703125	1.336484375	1.181620625	1.2841790625	1.1279296875	1.38671875
1.1373265625	1.435078125	1.15140625	1.1901050125	1.1787078125	1.406640625	1.609921875
1.4435125	1.30171875	1.404906875	1.31640625	1.3144785625	1.15234375	2.09443125
1.171875	1.43578125	1.142578125	1.279296875	1.11328125	1.285878125	1.309921875
1.10140625	1.2800625	1.2060540875	1.2257859375	1.1572265625	1.1181640625	1.234705625
1.105515025	1.201171875	1.1083584375	1.25	1.3109375	1.129646875	1.540390675
1.1572265625	1.484375	1.2109375	1.296826125	1.148678125	1.298826125	1.143148875
1.557421875	1.044521875	1.533203125	1.07421875	1.48410625	1.172875	1.3136364625
1.479421875	1.200540875	1.4892578125	1.1572265625	1.111328125	1.3865234375	1.103875
1.19140625	1.435546875	1.1865234375	1.10387875	1.15234375	1.3427734375	1.2138203125
1.4842578125	1.1062890625	1.50830625	1.860231875	1.1208154375	1.169409375	1.2578125
1.357078125	1.220703125	1.3013671875	1.20171875	1.3607409375	1.176638125	1.2548828125
1.171875	1.1453125	1.10140625	1.43578125	1.181640625	1.375073125	1.162109375
1.2000346875	1.083084375	1.2158203125	1.1572265625	1.127090875	1.19140625	1.3023046875
1.2000346875	1.1609231875	1.4111328125	1.1767578125	1.4208984375	1.142578125	1.5030625
1.250765625	1.2255850375	1.171875	1.1572265625	1.2548828125	1.474800375	1.2411790625
1.1376951125	1.1134765625	1.352109375	1.356134375	1.147460375	1.40625	1.1909921875
1.2158303125	1.1328125	1.180934375	1.083084375	1.324125	1.226646875	1.3023090625
1.10856825	1.5350860375	1.0086328125	1.366484375	1.1639921875	1.435546875	1.2768703125
1.5030625	1.2060540875	1.43578125	1.38640625	1.1238684375	1.1572265625	1.42578125
1.470016625	1.226703125	1.307890625	1.201171875	1.226703125	1.1376951125	1.3916013625
1.171875	1.240234375	1.1767578125	1.169293875	1.1940625	1.09375	1.3894625
1.25	1.1609231875	1.3037109575	1.181640625	1.171875	1.240234375	1.1279296875
1.5185546875	1.220703125	1.416015625	1.171875	1.3916013625	1.1572265625	1.411328125
1.171875	1.17878125	1.42678125	1.226703125	1.0366028125	1.2039453125	1.1528125
1.9109375	1.0086328125	1.328125	1.3376033125	1.3671875	1.129646875	1.1902890625



Apéndice B. Datos capturados de las pruebas experimentales

1.430649025	1.2109275	1.30859375	1.81640025	1.147400375	1.17234375	1.803884875
1.182100375	1.3530078125	1.2893244375	1.4101290625	1.1572265625	1.3025390625	1.39140025
1.2100375	1.2880625	1.23048875	1.337800625	1.20171875	1.43375125	1.1007521875
1.1865294375	1.18671875	1.1632621875	1.396484375	1.338125	1.25	1.1572265625
1.2148203125	1.416205625	1.2235850375	1.2744140625	1.1767078125	1.15294375	1.201171875
1.191140625	1.181640625	1.375350375	1.1474600375	1.14825	1.1609921875	1.416015625
1.50380625	1.11294375	1.4990234375	1.1565234375	1.336484375	1.1669921875	1.2548828125
1.1767578125	1.3089375	1.1572265625	1.181640025	1.133515625	1.2500625	1.1474600375
1.220703125	1.40484375	1.240394375	1.34565625	1.150183125	1.084981375	1.4579030625
1.3332421875	1.25	1.3134785625	1.19140625	1.00375	1.16384375	1.0731615625
1.1376958125	1.2363515625	1.435740875	1.39140625	1.4018071875	1.15234375	1.43375125
1.1869921875	1.103515325	1.5478515625	1.1181640625	1.311328125	1.181640025	1.491140625
1.3330678125	1.2109375	1.3841930875	1.161640625	1.3134765625	1.1279296875	1.2071875
1.34725025	1.102109375	1.484375	1.1685214375	1.40484375	1.1572265625	1.102109375
1.2109375	1.2548828125	1.1767078125	1.0883671875	1.1279296875	1.0791615625	1.1083884375
1.318509875	1.2109375	1.4590234375	1.3962800625	1.4897265625	1.1767078125	1.0818051375
1.576553125	1.235889375	1.387890625	1.201171875	1.1767078125	1.171875	1.1565234375
1.1962890625	1.3327390625	1.2709375	1.2841785625	1.19140625	1.3916015625	1.1669921875
1.479600375	1.231171875	1.474009375	1.201171875	1.40615	1.19140625	1.4101290625
1.1767578125	1.4458125	1.1865234375	1.458078125	1.171875	1.4501953125	1.1767078125
1.4404290875	1.3069540625	1.350353125	1.171875	1.201171875	1.1474603375	1.3328125
1.201171875	1.3623990625	1.1767078125	1.2744140625	1.1767078125	1.336484375	1.19140625
1.8189903125	1.171875	1.1767078125	1.19140625	1.0791615625	1.2703906875	1.1579296875
1.562890625	1.24795625	1.181640625	1.4697265625	1.162103375	1.376951125	1.102109375
1.23646875	1.1962890625	1.1767078125	1.2451171875	1.176963125	1.09375	1.4501953125
1.25	1.2363515625	1.396484375	1.220703125	1.3720783125	1.171875	1.4458125
1.171875	1.4458125	1.2009540875	1.474009375	1.171875	1.29935406875	1.201171875



# APÉNDICE C

## RESULTADOS DE ENTRENAMIENTO DE LA RED NEURONAL

### C.1. Resultado del entrenamiento de la Red Neuronal

Para realizar la simulación se tomaron las primeras 150 muestras de cada matriz de datos, es decir los primeros 150 renglones de las matrices "AL" (Alcohol), "A" (Amoníaco), "M" (Metano) y "LP" (Gas LP). El resultado mostrado en la columna simulación de las primeras 150 muestras corresponden al Alcohol, las siguientes al Amoníaco, Metano y las últimas 150 al Gas LP.

Muestra	Simulación	Muestra	Simulación	Muestra	Simulación	Muestra	Simulación
1	1.0321462243538	151	50.016420277561	301	33.284690351376	451	80.002687553672
2	0.09644889992567	152	10.964774254424	302	23.058729827782	452	45.002887568388
3	1.026128206650	153	50.018857018823	303	24.966567892843	453	35.36752741161
4	1.0002396728296	154	52.987000986131	304	29.9641518700763	454	12.907499361231
5	0.99689256974221	155	20.002864039385	305	17.07231241539	455	69.992587185399
6	1.0004442571068	156	50.011179487543	306	17.075374916388	456	40.003836437521
7	0.99930192810875	157	19.9968555593291	307	22.9983681540814	457	69.99048002828
8	0.9995353736451	158	13.987664493465	308	21.998610824678	458	82.990517301988
9	1.000101607887	159	50.006927593989	309	60.999132399878	459	50.997527611968
10	1.0002151882353	160	19.999290307779	310	20.9986862482771	460	23.001377228154
11	0.99977192799254	161	30.01251743382	311	58.998988800589	461	40.004711092388
12	1.0000225357947	162	20.006994352608	312	27.998207644422	462	39.999780804025
13	1.00121277032194	163	15.988890912975	313	30.999246576977	463	30.999056763283
14	1.0001093104958	164	20.01003019348	314	27.998229676156	464	40.000523874079
15	0.9997415877808	165	20.011110070376	315	26.99868119122	465	60.997517730798
16	1.00010970624585	166	30.017443256975	316	20.999740147656	466	30.999200907543
17	0.9998063227333	167	20.011989272324	317	20.998712497782	467	40.00005178259
18	0.99985257198218	168	30.016821129257	318	20.998437148168	468	39.999567129873
19	0.99957210077003	169	19.997070903957	319	20.999157902315	469	40.999304978947
20	0.99990419284668	170	19.998286828667	320	20.998742129532	470	50.999758878097
21	1.000196031782	171	30.0123278697656	321	29.9985744802827	471	40.9997339241983
22	1.00067219083944	172	50.017683680172	322	21.998343127608	472	51.999346524075
23	1.00042140078025	173	50.009048925115	323	29.998250362162	473	38.99920460738
24	0.99929671636627	174	19.99847307592	324	29.998758987379	474	40.999917730073
25	0.99987109003616	175	30.0087576286602	325	20.9990137785948	475	40.999322899385
26	1.00012659266951	176	30.0093957193803	326	29.998670192635	476	30.999616248714
27	0.99991756510839	177	19.999011514182	327	29.998812820637	477	39.999511389820
28	1.0001784987921	178	20.0001587870186	328	20.998931820329	478	11.999826404625
29	0.99953738378858	179	20.01431242266	329	20.998793734254	479	10.999473769092
30	1.00038887097971	180	20.003728032581	330	29.998626142811	480	39.999408524020
31	1.00006687208219	181	30.015020938499	331	18.998468025570	481	10.9999775998177
32	0.99941062167376	182	20.0046745069697	332	29.998278560695	482	20.999715607796
33	0.99978885748922	183	10.99742962268	333	28.998728377503	483	21.9993667108563



Apéndice C. Resultados de entrecruzamiento de la red neuronal

33	1.00055501739986	214	20.0007430243757	394	20.9987105072675	574	20.9994301617298
35	0.99983240500705	215	20.0001194758831	395	20.998643798295	575	20.0003213036010
36	0.99908126551712	216	20.01055121097	396	20.9981812830455	576	20.0003013921847
37	0.9995613312577	217	20.001066043313	397	20.998282235171	577	20.9997730254553
38	0.99963755500726	218	20.0001325936637	398	20.9985571179254	578	20.0004405085508
39	0.999022273192708	219	20.0109954631512	399	20.9984861505057	579	20.0005662233079
40	0.999902200033133	220	20.0076303203704	400	20.9980206688153	580	20.9998408101102
41	0.998540988339596	221	20.0087248089017	401	20.9992343481665	581	20.9997861317393
42	0.999300128051683	222	20.0017450123081	402	20.9983301579061	582	20.0004115807145
43	0.999634847009753	223	20.0008109165101	403	20.9987800016402	583	20.9993907800818
44	1.00010667713344	224	20.0116804887051	404	20.9985915745545	584	20.0002966755049
45	0.999961993958765	225	20.0016820812485	405	20.9990187715601	585	20.0008697300838
46	0.999108472217188	226	20.008009936136	406	20.998995070814	586	20.9992727356183
47	1.00028990504309	227	20.0066739237581	407	20.9987591479248	587	20.0005901729033
48	0.99972434582936	228	20.005247706562	408	20.9983278312653	588	20.0006424329609
49	1.000125117882798	229	20.0107727549319	409	20.9989593972155	589	20.9999578476762
50	1.00035028741324	230	20.0083437000048	410	20.9990179007134	590	20.9990688794852
51	1.00031176744851	231	20.0122816391008	411	20.9990397177018	591	20.0003171987049
52	0.999396339080021	232	20.008607729762	412	20.998852240046	592	20.0008022709068
53	0.999011227234279	233	20.009060164289	413	20.9993524709859	593	20.999324682511
54	0.999112229665385	234	20.0110849275131	414	20.9990736267134	594	20.9999514304441
55	1.00045945833175	235	20.0091896405176	415	20.9988754730400	595	20.999455778543
56	0.999021397431437	236	20.0113452812750	416	20.9990154304004	596	20.999630338408
57	1.0001530772453	237	20.0094639308873	417	20.9991353779383	597	20.0007148658085
58	0.999741034706171	238	20.0116816987651	418	20.9989701754939	598	20.000427580643
59	1.00006432743108	239	20.0019754662310	419	20.9992042680848	599	20.0003023997715
60	0.999056005386811	240	20.0027167777991	420	20.9992879917667	600	20.0009018361504
61	0.999475706808775	241	20.0053778270401	421	20.9991620226033	601	20.000150379369
62	1.0000880862298	242	20.0097079907321	422	20.9991376983432	602	20.99924784323107
63	0.999430917946212	243	20.0105255325458	423	20.999359073492	603	20.000303420499
64	0.999951247115085	244	20.0099901794967	424	20.9997848683527	604	20.999784722212
65	1.00026829875301	245	20.0096420758961	425	20.9996882879053	605	20.000064779497
66	1.00008988697582	246	20.001043088385	426	20.9990110038737	606	20.9995311006086
67	0.999274433857819	247	20.010460900065	427	20.9985241128902	607	20.9998830662525
68	0.999828296803035	248	20.0007849030648	428	20.9983830557203	608	20.0001047080158
69	0.999524313748072	249	20.0104782498593	429	20.9981579824178	609	20.99919173700270
70	0.99961036135036	250	20.008295138413	430	20.998013174022	610	20.0003771044219
71	0.99950214753794	251	20.0086369490997	431	20.9993966302696	611	20.9997761273382
72	0.999501390632635	252	20.002761495159	432	20.9991632645221	612	20.0004049285055
73	1.00061207825137	253	20.009344094912	433	20.999301208710	613	20.999965006212
74	0.9999478081083051	254	20.0027183096745	434	20.9978576262349	614	20.9997131181038
75	0.999568263483265	255	20.9997904906332	435	20.9993084066223	615	20.00002408273
76	0.999863025037548	256	20.0059566528093	436	20.9992159318349	616	20.0006761433857
77	1.00007515662125	257	20.012069550104	437	20.9994470533427	617	20.99996031512157
78	0.999777692751455	258	20.0088230689538	438	20.9992991022158	618	20.000351224452
79	0.99956255617801	259	20.9998711422243	439	20.9990754383105	619	20.9997269542851
80	0.99983985458026	260	20.0009353368932	440	20.9994076966036	620	20.000452735045
81	0.99948860253131	261	20.0081912679390	441	20.9991923158924	621	20.0001506623300
82	0.99966850199638	262	20.002333780675	442	20.999074623981	622	20.0002505667396
83	0.999399096408973	263	20.0008106483153	443	20.9990298182036	623	20.0013562817076
84	1.00089526340012	264	20.0018240489385	444	20.9994778267036	624	20.0003135234886
85	1.00067953854629	265	20.0099128151138	445	20.9995174141685	625	20.999978514648
86	0.99983381161028	266	20.9729368592715	446	20.9994157485619	626	20.0095010073941
87	0.999676369060297	267	20.0022823810212	447	20.9992376853068	627	20.0008054234244
88	1.00019080195784	268	20.0021969858816	448	20.9991377294926	628	20.0000081080388
89	0.999406721354351	269	20.005739186350	449	20.9990397279803	629	20.0002045193517
90	0.99978142212305	270	20.002808240823	450	20.9989675826646	630	20.0007420500099
91	0.999395850158069	271	20.0003318986311	451	20.9994001966719	631	20.9996478298241
92	0.999874812449338	272	20.0041283807838	452	20.9990608518475	632	20.9996306649677
93	0.99941438101126	273	20.0002294383077	453	20.9992896920091	633	20.9998609150100
94	0.99918134114617	274	20.0010243952018	454	20.9993184606979	634	20.9998191316095



Apéndice C. Resultados de entrenamiento de la red neuronal

95	1.60055967434057	275	21.0033087602636	453	29.0995068225375	635	39.9996073182751
96	0.999845648959673	276	19.9958166229657	456	29.9995137507599	636	39.9998251391884
97	1.00090409359923	277	21.0033618831175	457	29.0991347060375	637	39.9998875329531
98	1.00059263903131	278	19.9987553189954	458	29.9994248732311	638	39.9997241336552
99	0.999839653307071	279	21.0034514523129	459	29.9995963412388	639	40.001550088133
100	1.00013116573337	280	20.0031790817617	460	29.99975941071214	640	39.999894286729
101	1.00005579535093	281	21.0030823801906	461	29.9993163136313	641	39.9997659233066
102	1.00001772349973	282	20.0034731775909	462	29.99931802434	642	40.000926805886
103	1.0001567248003	283	19.9993809094371	463	29.9992283503293	643	39.9998275686876
104	1.00018205913184	284	19.9999042113989	464	29.9996039990515	644	39.999510813367
105	0.999512761926545	285	20.0066023788514	465	29.9988460126274	645	39.999795501091
106	0.999681761338598	286	20.0082492298907	466	29.998363387158	646	39.99959494501
107	0.999079738442704	287	19.9997533131040	467	30.0006379985085	647	-0.0003036073845
108	1.00032171088509	288	20.0012328878036	468	29.9987801032868	648	39.00099926170
109	1.0017462807172	289	19.9989591492546	469	29.9993582676918	649	39.9992014712705
110	0.999734443496019	290	19.9988978246535	470	29.999706975929	650	40.000302938789
111	0.999423350599172	291	20.0000931197127	471	29.9998167598691	651	39.999013497829
112	1.00000399309916	292	19.99951343218922	472	29.9998013391859	652	39.9997066736671
113	0.999773017833988	293	19.9990633806006	473	29.9994794466528	653	40.0001940378239
114	1.00017031149053	294	20.0011045854823	474	29.9998209214635	654	39.9999187154503
115	0.999916486653916	295	20.0025036488109	475	29.9996997224585	655	39.9993736588328
116	1.00108036811345	296	19.99782414435792	476	29.9992180653381	656	40.0006087835746
117	1.00248355124611	297	20.001123244550	477	29.99937393710069	657	40.001493866875
118	0.99904487186417	298	19.9961222236033	478	29.9990061719263	658	40.0015061817968
119	0.999461612323561	299	19.9980993778571	479	29.9992468880114	659	39.9992468310272
120	1.00186932212618	300	20.0010910711119	480	29.9995707310195	660	39.9990451352687
121	0.999682021161529	301	20.0000937943854	481	29.9996475989915	661	40.000393028541
122	0.999852344526286	302	19.9990751000657	482	29.9994773184783	662	39.999553793854
123	0.99914593591314	303	19.9993828129669	483	29.9993156931723	663	39.9995829041802
124	0.999680130843144	304	20.00039826973392	484	29.9993579998111	664	40.000329022952
125	0.999162903168336	305	19.9997180561458	485	29.99887616228409	665	40.00043714348
126	0.99999831868997	306	19.9995150043495	486	29.998339293027	666	39.9999543178869
127	1.00018635356912	307	20.0013981402361	487	29.9995147381197	667	39.999998433889
128	1.00078001281001	308	19.9983227442537	488	29.9995907476403	668	40.0013063613509
129	0.999724745819687	309	19.9972359319861	489	29.9991901107289	669	40.0005234135145
130	1.00028973334315	310	20.005824309509	490	29.9993011051589	670	39.9990344151407
131	0.99957782307805	311	19.9967135439872	491	29.9990286619244	671	39.999168738468
132	0.99983711153212	312	20.0001553965731	492	29.9995305112537	672	39.9995596310105
133	0.999539140896183	313	20.0035563003215	493	29.9991757483331	673	40.0003290033975
134	1.00184891951316	314	20.0002365496319	494	29.999827114073	674	40.000476811073
135	0.999372900401163	315	20.000198008601	495	29.9998783373075	675	39.9994674977725
136	1.000042035723813	316	19.9997437745894	496	29.999469886006	676	40.001904306891
137	1.00047832868571	317	19.998848643	497	29.999180017798	677	40.0006926838055
138	0.999667113917186	318	19.9987522433244	498	29.9991027829884	678	40.000224258211
139	1.00039643729021	319	19.99903992381543	499	29.9993951218995	679	40.000264979333
140	0.999201483728418	320	20.0012078911161	500	29.9997127199985	680	39.9997786681923
141	0.999340785296432	321	19.999702907057	501	29.998606060000	681	39.999331364571
142	1.00030287123213	322	19.9996591898977	502	29.999690411887	682	39.9996100073165
143	0.999256830752337	323	20.000308658803	503	29.999151449129	683	39.999568979e072
144	0.999463036523651	324	19.9981225177818	504	39.9993680261788	684	39.9998701303976
145	0.999128515779187	325	19.9980708199934	505	39.99907533349129	685	39.999593559656
146	0.999577107323775	326	19.9985044841134	506	39.9990671006373	686	40.000138808884
147	0.999521518515314	327	19.9991611262559	507	39.9991079989072	687	39.999703511530
148	1.00010818028853	328	19.998480030327	508	39.9992549389974	688	40.0007926970516
149	1.00026360282914	329	19.9980144671743	509	39.9992683083858	689	39.9998447833281
150	0.999287994309792	330	19.998535382967	510	29.99936336036042	690	39.9997474095222
151	0.99997015618091	331	19.999596075129	511	39.9999930032829	691	40.0001551788816
152	0.99936143048534	332	19.9989171210717	512	29.9998035490684	692	39.99982074053179
153	0.999474336745619	333	19.997598010684	513	29.9987432537671	693	40.0003822400502
154	0.9999661173219863	334	19.9987900689738	514	39.9995954887379	694	39.9994281151067
155	0.999937238934998	335	19.9987300263629	515	29.9988031800107	695	39.9998197885483
156	0.999947731864407	336	19.9985138539037	516	39.9995688010329	696	39.9995721912488



Apéndice C. Resultados de entrenamiento de la red neuronal

157	1.00012007252052	337	13.9972233571932	517	29.9994030654235	697	39.9999761810265
158	1.000235941533030	338	13.9958862557662	518	29.9997712017656	698	39.9995114527319
159	1.0006808048591	339	13.9951160777309	519	29.9993069863192	699	40.0002611789451
160	1.0001412300232	340	13.9963085848327	520	30.0002100889165	700	39.9996105192189
161	0.999829335930273	341	13.9961422100186	521	30.999752871736	701	39.9996413469816
162	0.999358298379305	342	13.9971995899017	522	29.9996319309197	702	39.9999176925838
163	0.999779102152534	343	13.9972232311469	523	29.9999036768536	703	39.9998855202035
164	0.999203097557365	344	13.9988171397584	524	29.9998099863132	704	40.0002956613301
165	1.00081358096108	345	13.9972267932064	525	29.9995811068787	705	39.9999377416156
166	0.99912750498303	346	13.9966769823989	526	29.9991429138777	706	39.9999545821464
167	0.999817147840125	347	13.99831198249835	527	29.9988820633538	707	39.9991539887707
168	0.999733078060321	348	13.9960970589646	528	29.9997079742011	708	39.9997580593075
169	0.99993908292762	349	13.9993460244064	529	29.9993249469817	709	40.0002993894806
170	1.00057513188943	350	13.9957530590618	530	29.9989489339636	710	40.0008028772351
171	0.999770844754064	351	13.9972190893627	531	30.0002914731747	711	39.9997896035362
172	0.999301193762599	352	13.9966531438969	532	29.9999471901157	712	40.000022207598
173	0.999537881887173	353	13.99852110904796	533	30.0001170683965	713	40.0001851134671
174	0.99982624091092	354	13.9989340726997	534	29.9997244518401	714	40.0000608757985
175	0.99997147054034	355	13.9950611965013	535	29.999918532862	715	39.9997772336976
176	0.999913994828553	356	13.9969181522314	536	29.9998346912831	716	39.9997737930017
177	1.00048749120337	357	13.9968042193192	537	29.9993346912831	717	40.0005830639872
178	0.999519225009356	358	13.9983085653315	538	29.9987600280640	718	39.9995894378357
179	1.00049847425813	359	13.9984335870883	539	29.9983182786225	719	39.9996109824346
180	0.99975073777619	360	13.9990698881005	540	29.9998933732906	720	39.9992287496456



---

## BIBLIOGRAFÍA Y REFERENCIAS

- [1] J.W. Gardner y P.N. Bartlett, *Electronic Noses: Principles and Applications*, Oxford University Press (1999).
- [2] J.Lazano, J.P.Santos, M.Alexandre, I.Sayago, M.García, A.Martínez, M.Cabellos\*, J.Girón \*\*, M.C.Herrillo, *Sistema de nariz electrónica para el análisis para la evolución del vino controlada por PC*, Dept. Arquitectura de Computadores y Automática, Fac. Físicas, Universidad Complutense de Madrid, (Enero 2006).
- [3] Mari Carmen Herrillo Cúcmes y Daniel Matatagui, *Narices electrónicas para la detección de agentes de guerra química*, Sociedad Española de Ciencias Sensoriales, (Septiembre 2012).
- [4] Iveth Moreno, Rony Cabellero, Ramón Galán, Fernando Matía, Agustín Jiménez, *Revista Iberoamericana de Automática e Informática Industrial: La Nariz Electrónica: Estado del Arte*, ISSN: 1697-7912, Vol. 6, Núm. 3, (Julio 2009).
- [5] Marsil R. (editor), *Techniques for analyzing food aroma*, New York, United States: Marcel Dekker Inc, Hodings D., *Electronic nose: sensor array based instruments that the human nose*; p. 331-321 (1997).
- [6] Schaller E, Bosser J, Escher F, *Electronic Noses and their application to food*, LWT, 21:305-316 (1998 Jun 20).
- [7] Martha C. Quicazan S., Amanda C. Díaz M., Carlos M. Zuluanga D., *La nariz Electrónica, una novedosa herramienta para el control de procesos y calidad en la industria agroalimentaria* Vitae, vol. 18, núm. 2, pp: 209-217, Universidad de Antioquia, Colombia (2011).
- [8] Octavio Maldonado Saavedra, Karina Gatiérrez Frago, Carlos A. Lobato Tapia, Marisol Herrera Rívero y Enrique Méndez Bolaina, *El sistema olfatorio: el sentido de los olores*, Revista de divulgación científica y tecnológica de la Universidad Veracruzana, Volumen XXV, Número 2, (Mayo-Agosto de 2012), <http://www.uv.mx/cienciahombre/revistae/vol25num2/articulos/sistema/>

- [9] Martínez Ortigoza Gloria Elizabeth, *Proceso inalmbrico de señales en redes inteligentes de sensores*, Universidad Veracruzana Facultad De Ingeniería Mecánica Eléctrica, Xalapa Veracruz (2009).
- [10] J. Javier Peláez Moreno, *Implementación del algoritmo de selección de variables conocido con el nombre de "Simulated Annealing" (Reconocido Simulado)*, Escola Técnica Superior Enginyeria, Universitat Rovira i Virgili, (Enero 2005).
- [11] Stetter J. R. and Peurose W. R. "The Electrochemical Nose". *Electrochemistry Encyclopedia*. Departmento of Biological, Chemical, and Physical Sciences Illinois Institute of Technology Chicago (2001).
- [12] Gardner J. W. and Barlett P.N., *Electronic Noses: Principles and Applications*. Oxford University Press (1999).
- [13] Rodríguez -Gil G. *El Poderoso Sentido del Olfato*. En Resources, Volumen 11, N.2. (2004).
- [14] Pamela Padilla, *Nariz electrónica detecta comida rancia*. Telemundo 47.(7-04-2014).<http://www.telemundo47.com/noticias/Nariz-electronica-para-evitar-enfermedades-254202301.html>
- [15] Marcis Leja,Hu Liu,Hossain Haick, *Breath testing: the future for digestive cancer detection*,Expert Reviews 10.1586/17474124.2013.811033 UK Ltd ISSN 1747-4124 (2013)
- [16] Roberto Mujica, *Contaminación en áreas urbanas*, Organization Distribution and Sales: Geneva.(2011)
- [17] Durán C. M. A., Director de Tesis Jesús Brezmes, *Tesis Doctoral, Diseño y optimización de los subsistema de un sistema de olfato electrónico para aplicaciones agroalimentarias e industriales*. Universitat Rovira i Virgili, España (2005).
- [18] López-Ferría S, Cárdenas S, García-Mesa J, Valcárcel M. *Simple and rapid instrumental characterization of sensory attributes of virgin olive oil based on the direct coupling headspace-mass spectrometry*, J Chromatogr A. 1188(2): 308 - 313 (2008 Apr 25).
- [19] Schaller E, Bossert J, Escher F. *Electronic Noses and their application to food*. LWT. 21: 305-316 (1998 Jan 20).
- [20] Haugen J, Kvaal K. *Electronic Nose and Artificial Neural Network*. Meat Sci. 49(1): 8273-886 (1998).
- [21] Lozano J, Santos J, Horillo M. *Classification of white wine aromas with an electronic nose*. Talanta 67(3): 610-616 (2005 Sep 15).
- [22] Pearce T. C., Schiffman S. S., Nagle H. T. and Gardner J. W. *Handbook of Machine Olfaction. Electronic Nose Technology*. Wiley-Vch (2003).
- [23] Cortijo F. *Técnicas Superiores, Aproximación no Paramétrica, Reconocimiento de patrones*. Universidad de Granada. España (2003).



- [24] Moreno M. N. y López V. F. *Uso de Técnicas no Supervisadas en la Construcción de Modelos de Clasificación en Ingeniería del Software. Tendencias de la Minería de Datos en España*. R. Gradles, J. C. Riquelme, J. S. Aguilar- Ruiz (eds.), 143-153 (2004).
- [25] B. Ripley. *Pattern Recognition and Neural Network*. Cambridge University Press, (1996).
- [26] McCulloch, W S y Pitts, W H. *A Logical Calculus of the Ideas Immanent in Nervous Activity*. Bulletin of Mathematical Biophysics, 5:115-133 (1943).
- [27] Karl Laszley, *Brain mechanisms and intelligence*, MA: MIT Press (1929).
- [28] Hebb, D O. *The Organization of Behavior: a Neuropsychological Theory*. New York: Wiley (1949).
- [29] Ramón y Cajal, S. *Histologie du Système Nerveux*. Paris: Maloine (1911).
- [30] Emilio Soria, Antonio Blanco. *Redes Neuronales Artificiales*. Autores Científico-Técnicos y Académicos, ACTA, (Marzo 2002).
- [31] Rochester, N, Holland, J H, Haibt, L H y Fuda, W L. *Test on a cell assembly theory of the action of the brain, using a large digital computer*. IRE Transactions on Information Theory. IT-2: 80-93 (1956).
- [32] Von Neumann, J *First draft of a report on the EDVAC*, republicado en *The Origins of Digital Computers: Selected Papers*. Randall, B (ed) Berlin:Springer,1982, (1945).
- [33] Rosenblatt, F. *The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain*. Psychological Review, 65, No.6: 386-408 (1958).
- [34] Rosenblatt, F. *Principles of Neurodynamics*. Washington: Spartan Books (1962).
- [35] Widrow, B y Hoff, M E. *Adaptive switching circuits*. 1960 IRE WESCON Convention Record, New York: IRE, 96-104 (1960).
- [36] *Block Mark II*. Washington: Spartan Books (1962).
- [37] Minsky, M y Pappert, S J. *Perceptrons*. Cambridge, MA: MIT Press (1969).
- [38] Kohonen, T. *Correlation matrix memories*. IEEE Trans. on Computers, C-21: 353-358 (1972).
- [39] Anderson, J A. *A simple neural network generating an interactive memory*. Mathematical Biosciences 14: 197-220 (1972).
- [40] Grossberg, S (ed). *The Adaptive Brain (Vols I y II)*. New York: North Holland (1980).
- [41] Hopfield, J J. *Neural networks and physical systems with emergent collective computational abilities*. Proc. Natl Acad. Sci. 79: 2554-2558 (1982).
- [42] Kohonen, T. *Self-organized formation of topologically correct feature maps*. Biological Cybernetics 43: 50-69 (1982).



- [43] Fukushima, Miyake e Ito, *Transactions on Neocognitron Systems*, Man and Cybernetics IEEE, (1983).
- [44] Richard S. Sutton, Andrew G. Barto, *Learning and Computational Neuroscience: Foundations of Adaptive Networks*, M. Gabriel and J. Moore, Eds., pp. 497-537. MIT Press, (1990).
- [45] Hopfield, J J y Tank, D W *Neural Computation of Decisions in Optimization Problems*. Biol. Cybernetics, 52:141-152 (1985).
- [46] Rumelhart, D E, Hinton, G E y Williams, R J, *Learning internal representations by error propagation in Rumelhart D E y McClelland J L (eds) Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1. Cambridge, MA: MIT Press, pp 318-362 (1986).
- [47] D. S. Hrcmhead, David Lowe, *Multivariable Functional Interpolation and Adaptive Networks*, Speech Research Unit, Royal Signals and Radar Establishment, St. Andrews Road, Great Malvern, Worcs. WR14 3PS, United Kingdom, 321- 355 (1988).
- [48] Tomaso Poggio, Federico Girosi, *Networks for approximation and learning*, proceedings of the IEEE, vol 78, No 9, (September 1990).
- [49] Eric A. Wan y Rudolph van der Merwe, *The Unscented Kalman Filter for Nonlinear Estimation*, Oregon Graduate Institute of Science y Technology 20000 NW Walker Rd, Beaverton, Oregon 97006, (2004).
- [50] María Isabel Acosta Buitrago, Camilo Alfonso Zuluaga Muñoz, *Tutorial sobre redes neuronales aplicadas en ingeniería eléctrica y su implementación en un sitio web*, Universidad Tecnológica de Pereira, Facultad de Ingeniería Eléctrica, (2000).
- [51] Xabier Basogain Olabe, *Redes neuronales y sus aplicaciones*, Escuela Superior de Ingeniería de Bilbao, EHU, (Junio 2003).
- [52] Carlos Alberto Ruiz, Marta Susana Basualdo, *Redes Neuronales: Conceptos Básicos y Aplicaciones*. Universidad Tecnológica Nacional - Facultad Regional Rosario Departamento de Ingeniería Química, (Marzo 2001).
- [53] Hanwei, *MQ series Semiconductor Gas Sensor*, Data sheet, <http://www.hwsensor.com/jwwebeditor/uploadfile/201101241131043217.pdf>
- [54] Microelectronicos, *Arduino Mega 2560*, Data sheet, <http://www.microelectronicos.com/datasheets/>
- [55] Andrés Oyaree, *Guía del Usuario XBEE Series 1*, Ingeniería MCI LTDA, Revisión (Julio 2010).
- [56] M. C. José Jaime Esqueda Elizondo, *MATLAB e Interfaces gráficas*, Universidad Autónoma de Baja California, Unidad Tijuana, CONATEC (2002).
- [57] Hines E. L., Llobet E., Gardner J. W. *Electronic noses: a review of signal processing techniques*. IEE Proc.-Circuits Devices Syst., Volume 146, 297-310 (1999).



