



UNIVERSIDAD NACIONAL DE LOJA
Área de la Energía, las Industrias y los Recursos
Naturales No Renovables

INGENIERÍA ELECTROMECÁNICA

TEMA:

M

ODELO DE PREDICCIÓN DE MICROCLIMA DE
INVERNADEROS, CASO DE ESTUDIO:
INVERNADERO DE ORQUIDEAS DEL JARDÍN
BOTÁNICO "REINALDO ESPINOSA".

**TESIS DE GRADO PREVIO A OPTAR
POR EL TÍTULO DE INGENIERO
ELECTROMECÁNICO**

AUTORES:

- Victor Fernando Herrera Sarango
- José Alfredo Paucar Jumbo

Loja – Ecuador

CERTIFICACIÓN

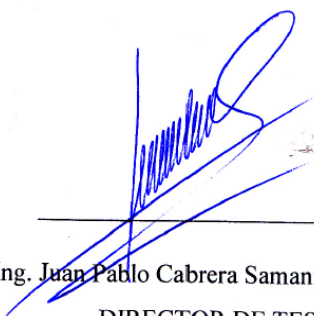
Ing. Juan Pablo Cabrera Samaniego

DIRECTOR DE TESIS

CERTIFICA:

Haber dirigido, asesorado, revisado y corregido el presente trabajo de tesis de grado, en su proceso de investigación cuyo tema versa en: **Modelo de predicción de microclima de invernaderos, caso de estudio: Invernadero de Orquídeas del Jardín Botánico “Reinaldo Espinoza”**., previa a la obtención del título de Ingeniero Electromecánico, realizado por los señores egresados: **Víctor Fernando Herrera Sarango y José Alfredo Paucar Jumbo**, la misma que cumple con la reglamentación y políticas de investigación, por lo que autorizo su presentación y posterior sustentación y defensa.

Loja, 03 de Octubre de 2013



Ing. Juan Pablo Cabrera Samaniego Mgs. Sc
DIRECTOR DE TESIS

AUTORÍA

Nosotros, Víctor Fernando Herrera Sarango y José Alfredo Paucar Jumbo declaramos ser autores del presente trabajo de tesis y eximimos expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente aceptamos y autorizamos a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio Institucional-Biblioteca Virtual.

Autor 1: Víctor Fernando Herrera Sarango

Firma:



Cédula: 110443677-7

Fecha: Noviembre 07 del 2013

Autor 2: José Alfredo Paucar Jumbo.

Firma:



Cédula: 1104613797

Fecha: Noviembre 07 del 2013

**CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DE LOS AUTORES,
PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL, Y
PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.**

Nosotros, Víctor Fernando Herrera Sarango y José Alfredo Paucar Jumbo declaramos ser autores de la tesis titulada **Modelo de predicción de microclima de invernaderos, caso de estudio: Invernadero de Orquídeas del Jardín Botánico “Reinaldo Espinosa”**, como requisito para optar al grado de Ingenieros en Electromecánica; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

Para constancia de esta autorización, en la ciudad de Loja, a los siete días del mes de Noviembre del dos mil trece.

Firma:



Autor 1: Víctor Fernando Herrera Sarango

Cédula: 110443677-7

Dirección: Loja, San Sebastián

Teléfono: 2562431; Cel.: 0993021808

Firma:



Autor 2: José Alfredo Paucar Jumbo

Cédula: 110461379-7

Dirección: Catamayo, Nueva Esperanza

Teléfono: 2555257; Cel.: 0980843316

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Juan Pablo Cabrera Samaniego

Tribunal de Grado: Ing. Jorge Patricio Muñoz Vizhñay, Mg.Sc.

Ing. Jorge Enrique Carrión González, Mg.Sc

Ing. Manuel Ignacio Ayala Chauvin Mg.Sc.

PENSAMIENTOS

“La perseverancia es muy importante para el éxito. Y que si uno no se cansa de llamar a la puerta con el vigor y la paciencia necesarios, alguien le abrirá al final”

Longfellow

"Nunca consideres el estudio como una obligación sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber”

Albert Einstein

“Si una persona es perseverante, aunque sea dura de entendimiento, se hará inteligente; y aunque sea débil se transformará en fuerte”.

Leonardo Da Vinci

DEDICATORIA

Este proyecto dedico primeramente a Dios por ser quien guía mis pasos; a mis Padres por darme el apoyo necesario para poder seguir adelante y a su vez la confianza que depositaron en mí; a mis hermanas y hermanos que siempre han sido mi inspiración y fortaleza para no decaer en los obstáculos que se presentan en el camino.

José Alfredo Paucar Jumbo.

El presente proyecto dedico al Todopoderoso por ser quien guía mi camino; a mis Padres y Familiares por todo el esfuerzo y apoyo en mi vida Estudiantil y Personal, gracias por todo, a mis hermanos y amigos por su apoyo para con mi persona ustedes han sido un pilar fundamental en mi vida. Infinitas Gracias.

Victor Fernando Herrera Sarango.

AGRADECIMIENTO

Expresamos en estas líneas el más sincero agradecimiento a todas aquellas personas que de una u otra forma han contribuido a la realización de esta tesis.

En primer lugar quisiéramos dirigirnos a nuestros padres para expresarles nuestro más profundo sentir de agradecimiento por todos aquellos sacrificios que nuestra formación ha supuesto para ellos. Asimismo agradecerles a nuestros Familiares por todo el cariño y apoyo moral que siempre nos han demostrado.

A nuestro director de tesis, Ing. Juan Pablo Cabrera, agradecerle la ayuda incondicional que siempre ha prestado y el haber sabido guiar nuestro trabajo en los momentos más delicados. Su gran cultura científica, su disponibilidad y su simplicidad son algunos de los tantos elementos que han favorecido el desarrollo de la presente tesis.

Al Ing. Marcelo Gutiérrez encargado del Jardín Botánico “Reinaldo Espinosa”, por las facilidades prestadas para con nosotros en el desarrollo del presente trabajo de Tesis.

A la Universidad Nacional de Loja por donde hemos pasado una etapa muy importante de la vida y hemos recibido una educación ejemplar.

A los docentes de la carrera de Ingeniería en Electromecánica por sus conocimientos científico, moral y humanista brindados en el proceso de enseñanza.

A nuestros apreciados amigos y amigas por haber compartido esta etapa de nuestras vidas, a todos quienes colaboraron con su granito de arena para la realización de este trabajo y han acompañado de una u otra forma durante este proceso.

TABLA DE CONTENIDOS

a. TÍTULO.....	1
b. RESUMEN.....	2
c. INTRODUCCIÓN.....	4
d. REVISIÓN DE LITERATURA.....	5
d.1 Invernadero.....	5
d.1.1 ¿Para qué sirve un invernadero?.....	5
d.1.2 Estructura.....	5
d.1.3 Cobertura.....	5
d.1.4 Orientación.....	6
d.2. Orquídeas.....	7
d.2.1 ¿Que son?.....	7
d.2.2 Requerimientos de riego de agua.....	8
d.2.2.1 Riego excesivo.....	9
d.2.2.2 Riego deficiente.....	9
d.2.2.3 Calidad del agua de riego.....	10
d.2.3 Requerimientos de Humedad.....	11
d.2.3.1 Algunos consejos para aumentar la Humedad Ambiental.....	11
d.2.4 Requerimientos de luz.....	12
d.2.5. Requerimientos de Temperatura.....	14
d.2.6 Aireación.....	15
d.2.6.1 Aspectos a considerar para evitar la mala ventilación.....	15
d.3. Redes Neuronales.....	16
d.3.1 Introducción.....	16
d.3.2 El Modelo Biológico.....	18
d.3.2.1 El axón.....	18
d.3.2.2 Las dendritas.....	18
d.3.2.3 Entradas.....	19
d.3.2.4 Salidas.....	20
d.3.2.4 Pesos.....	20
d.3.3 El Modelo Artificial.....	20

d.3.3.1 Características de las Redes Neuronales Artificiales.....	21
d.4 La Neurona Artificial.....	23
d.4.1 Estructura básica.....	23
d.4.2 Clasificación de las Redes Neuronales Artificiales.....	26
d.5 Redes de alimentación hacia adelante o Feedforward.....	27
d.5.1 Redes simple capa.....	27
d.5.2 Redes multicapa.....	28
d.5.3 Redes recurrentes.....	30
d.5.4 Redes con estructura enrejada.....	31
d.6 Entrenamiento de las Redes Neuronales Artificiales.....	32
d.6.1 Entrenamiento supervisado.....	33
d.6.2. Entrenamiento no supervisado.....	34
e. MATERIALES Y MÉTODOS.....	35
e.1 Materiales.....	35
e.1.1 Equipo de medición Fluke 975 Air Meter.....	35
e.1.1.1 Características.....	35
e.1.2. Matlab.....	36
e.1.2.1 Herramientas.....	37
e.1.2.2 Toolbox de Matlab NTSTOOL.....	38
e.1.3 Software Fluke View Forms.....	40
e.1.3.1 Características de FlukeView.....	40
e.1.3.2 Fluke View Forms Basic.....	40
e.1.4 Recursos informáticos.....	41
e.2 Métodos.....	42
e.2.1 Técnicas de modelado y recolección de datos.....	42
e.2.1.1 Procedimiento de recolección de datos.....	42
e.2.2 Procedimiento para autoajustar datos.....	47
e.2.3 Procedimiento para la crear la Red Neuronal mediante el Toolbox NTSTOOL.....	48
e.2.3.1 Selección de la Red.....	48
e.2.3.2 Toma de datos.....	49
e.2.3.3 Selección para su validación.....	49

e.2.3.4 Selección de la Arquitectura de la Red.....	51
e.2.3.5 Entrenamiento de la Red.....	52
e.2.4. Procedimiento para predicción y validación de datos.....	52
f.- RESULTADOS.....	53
f.1 Procesamiento de datos.....	53
f.1.1 Importar datos a Matlab.....	54
f.2. Resultados de los modelos generados al entrenar la Red Neuronal y predicción.....	55
f.2.1 Resultados de la Red Neuronal con datos de Temperatura.....	56
f.2.1.1 Resultados de predicción con valores de Temperatura.....	58
f.2.2. Resultados de la Red Neuronal con datos de Humedad Relativa.....	60
f.2.2.1 Resultados de predicción con valores de Humedad Relativa.....	62
f.2.3 Resultados de la Red Neuronal con datos de CO2.....	64
f.2.3.1 Resultados de predicción con valores de CO2.....	66
f.2.4 Resumen de los resultados de los modelos utilizando las RNA.....	68
g.- DISCUSIÓN.....	69
h.- CONCLUSIONES.....	71
i.- RECOMENDACIONES.....	72
j.- BIBLIOGRAFÍA.....	73
k. ANEXOS.....	75

SIMBOLOGÍA

MATLAB	(MATrixLABoratory) lenguaje de alto nivel y un ambiente interactivo.
RNA	Red neuronal auto-regresiva.
RN	Red neuronal.
MSE	Error Cuadrático Medio.
R	Regresión.
Delays	Retrasos de la red.
k	Unidad fundamental de procesamiento de cualquier modelo de RNA.
W	Peso de la neurona artificial.
W_{kp}	Conexión entre la neurona k y la neurona p de la capa previa.
X	Señales de entrada de la red.
Σ	Sumador del total de la red.
F	Función de activación.
Θ_k	Entrada negativa que disminuye la entrada neta de la función de activación.
A_k	Salida de la función de activación
V_k	Potencial de activación de la red
n	Número de la iteración.
$W_{kj}(n)$	Peso sináptico entre las neuronas k de la capa actual y j de la capa previa.
$\Delta W_{kj}(n)$	Ajuste aplicado a los pesos en el paso n.

Bias	Entrada adicional positiva.
HT	Datos de Temperatura.
DTtemp	Datos autoajustados de Temperatura.
HR	Datos de Humedad Relativa.
DThr	Datos autoajustados de Humedad Relativa.
CO2	Datos de Dióxido de Carbono.
DTco2	Datos autoajustados de Dióxido de Carbono.
net	Red Neuronal a ser evaluada.
A1	Entradas de valoración para la Red.
A2	Estados iniciales de la serie temporal.
Val	Datos de validación para la Red.
DA1	Entradas de valoración para la Red autoajustados.
DA2	Estados iniciales de la serie temporal autoajustados.
DVal	Datos de validación para la Red autoajustados.
outputs	Resultados de predicción de la serie temporal.
e	Errores de predicción.

a. TÍTULO

**Modelo de predicción de microclima de invernaderos, caso de estudio:
Invernadero de Orquídeas del Jardín Botánico “Reinaldo Espinosa”**

b. RESUMEN

En la producción de cultivos bajo condiciones de invernadero es importante optimizar y controlar el manejo del ambiente, usando modelos dinámicos; los modelos no lineales auto-regresivos, basados en mediciones de entradas, presenta un escenario de trabajo que permite estudiar sistemas complejos y no-lineales capaces de predecir series de tiempo, haciendo uso de la técnica de inteligencia artificial en particular Redes Neuronales (RN).

En el presente trabajo se desarrolló modelos neuronales auto-regresivos para predecir el comportamiento de la Temperatura, la Humedad Relativa y el CO₂ en el interior del invernadero.

Como resultado de la creación de la Red Neuronal se tiene que con las variables de Temperatura, Humedad Relativa y CO₂ están en rendimientos de 99,578 %, 96,61 % y 98,29 % respectivamente. Y en datos de error de predicción con un valor 0,45 %, todos estos resultados se los obtuvieron con tiempos de cálculo razonables.

Palabras Clave: *invernadero, ntstool, redes neuronales, datos.*

SUMMARY

In the cultivation production under hothouse conditions it is important to optimize and to control the handling of the ambience, using dynamic models; the auto-regressive not linear models, based on earnings measurements, it presents a stage of work that allows to study complex and not linear systems capable of predicting series of time, making use of the skill(technology) of artificial intelligence in particular Networks Neuronales (RN).

In the present work one developed models neuronales auto-regressive to predict the behavior(manner) of the Temperature, the Relative Moisture and the CO₂ inside the hothouse.

As result of the creation of the Network Neuronal has that with the variables of Temperature, Relative Moisture and CO₂ they are in yields of 99,578 %, 96,61 % and 98,29 % respectively. And in information of error of prediction with a value 0,45 %, all these results obtained them with reasonable calculation times.

Keywords: greenhouse, ntstool, neural networks, data.

c. INTRODUCCIÓN

Los invernaderos son construcciones agrícolas con cubierta transparente, cuyo objetivo es atrapar la energía solar y crear artificialmente las condiciones ambientales que requieren las plantas, en este caso nuestra investigación se la realizó en el invernadero de orquídeas del jardín botánico “Reinaldo Espinosa” de la Universidad Nacional de Loja, mediante el cual se elaboraremos modelos con redes neuronales auto-regresivas para predecir las variables de Temperatura, Humedad Relativa y CO₂ en el interior de dicho invernadero.

El invernadero de orquídeas de nuestra universidad es de tipo capilla y tiene las siguientes dimensiones:

- ✓ 10 m (largo)
- ✓ 4 m (ancho) y,
- ✓ 3,50 m (altura).

El invernadero fue monitoreado, cada 10 min, obteniendo un total de 144 datos diarios durante el intervalo de tiempo de Octubre a Noviembre del 2012, dándonos como resultado 8 640 datos totales de las variables a predecir.

La adquisición de datos se la realizó mediante el equipo denominado Fluke 975 Air Meter; en cual mide, registra y visualiza de forma simultánea los valores de temperatura, humedad, CO₂ y CO en su pantalla de retroiluminación. Por lo que se garantiza la medición más real de los datos obtenidos. Se hizo uso de los software de uso específico:

- ✓ Fluke ViewForms Basic y,
- ✓ Matlab

El primero, que facilitó la descarga de los datos del equipo de censado en archivos .xlms (archivos de Microsoft Excel), y el segundo mediante el Toolbox NTSTOOL, que sirvió para el procesamiento, entrenamiento y validación de los modelos generados.

d. REVISIÓN DE LITERATURA

d.1 Invernadero

Cada vez más los invernaderos forman parte de procesos productivos que involucran a especies vegetales; tanto en grandes empresas como en el hogar para producción de hortalizas para el autoconsumo. Hasta hace un tiempo, los invernaderos eran una práctica costosa, que solo se justificaba para cultivos muy valiosos. Hoy, gracias a la existencia en el mercado de nuevos materiales, los invernaderos constituyen una herramienta útil y económica con la cual es posible prolongar los periodos de crecimiento de las plantas en general.

d.1.1 ¿Para qué sirve un invernadero?

Un invernadero es una construcción especial que sirve para crear y mantener las condiciones ambientales apropiadas para el cultivo de especies vegetales; sean verduras, plantas ornamentales o plantas para forestación.

d.1.2 Estructura.

Puede estar constituida por diversos materiales, los más comunes son el metal y la madera. Actualmente, el costo entre estos dos materiales mantiene una relación de 3:1, es decir que una estructura de metal cuesta tres veces más que una de madera. Con respecto a la vida útil de estas estructuras, la de metal está estimada en 25 años, con un pequeño mantenimiento cada tres años; mientras que en madera podemos esperar una duración de 5, con mantenimiento cada 2 años.

d.1.3 Cobertura.

Es el elemento que ejerce la verdadera protección del cultivo, porque permite el paso de la luz y el calor y constituye una barrera para el frío, el viento, y cualquier otra condición climática que no favorezca el buen desarrollo de las plantas. La cobertura debe cumplir los siguientes requisitos fundamentales:

- Resistencia física.
- Duración suficiente para que su utilización sea rentable.
- Máxima transparencia a radiación de onda corta, que es la luz solar que se recibe durante el día.

Esta última característica es la que conducen a que la temperatura del interior sea superior a la del exterior. Los materiales más comunes utilizados como cobertura son el plástico y el vidrio, cada uno con sus ventajas y sus limitaciones.

d.1.4 Orientación

Uno de los factores que más incide en la producción de cualquier especie vegetal es la luz, por lo que debemos procurar que esta llegue lo mejor posible al invernadero. La orientación del mismo hará que los rayos solares penetren en mayor o menor grado. La orientación más conveniente es Este – Oeste, teniendo como restricción técnica que el lado más largo del invernadero este orientado hacia el Norte.



Figura 1. Vista de un invernadero común.

Fuente: zamora.infoinfo.com.mx/ficha/constructora_de_invernaderosS.A.

d.2 Orquídeas



Figura 2. Orquídeas que se cultivan en el invernadero de la UNL.

Fuente: Elaboración los autores.

d.2.1 ¿Qué son? :

Las Orquídeas son plantas herbáceas de la Familia Orchidaceae, clase Liliopsida (monocotiledóneas), muy abundantes, con más de 600 géneros y 17 000 especies en el mundo. Aunque son más abundantes en los trópicos, también existen especies en ambientes templados, desde el nivel del mar a grandes altitudes.

Se caracterizan por poseer flores muy vistosas, hermafroditas (ambos sexos en la misma flor), zigomorfas (con 1 solo plano de simetría), trímeras (3 sépalos y 3 pétalos) y una columna central que sustenta las estructuras reproductivas masculinas (anteras) y femeninas (pistilo) llamada ginostemo.

El pétalo inferior se llama labelo y su morfología define a los distintos géneros de orquídeas. Las flores pueden ser aisladas o en inflorescencia y son polinizadas por insectos. El polen se encuentra aglomerado, formando una masa llamada polinio el que tiene un extremo con un ensanchamiento glandular, pegajoso, que sirve para que el polinio se adhiera al cuerpo del insecto polinizador.

El fruto es una cápsula seca con muchas semillas pequeñas, sin endosperma y con embrión no diferenciado.

Tienen dos tipos básicos de crecimiento:

Simpodial, en las que el nuevo crecimiento se produce en sentido horizontal, a partir de un tallo subterráneo o rizoma, generando una sub-unidad capaz de producir una flor o inflorescencia y de ser eventualmente separada de la planta (ej. Cattleya).

Monopodial, en las que el nuevo crecimiento se produce en sentido vertical, con lo cual la planta crece constantemente en altura.

d.2.2 Requerimientos de Riego / agua



Figura 3. Riego de orquídeas en el invernadero.

Fuente: vyrsa.com/catalogo/agricultura/métodos/agro-stand.aspx

El riego es otro aspecto muy importante en el cultivo de las orquídeas epifíticas. Estas plantas crecen sobre árboles u otras plantas en sus hábitats naturales y obtienen la humedad desde el aire y del agua de lluvia que escurre por la superficie de las ramas. Esto significa que las raíces de las orquídeas epifíticas no están nunca sumergidas en agua en sus hábitat natural.

La forma de regar a las orquídeas es, por lo tanto, un poco distinta a la del resto de las plantas terrestres. Cuánta agua debe aplicar y con qué frecuencia depende de varias cosas:

1. Cuan seco es el ambiente donde usted cultiva las orquídeas.
2. El tamaño de la maceta.
3. El tamaño de la planta.
4. El tipo de sustrato en el que crece la planta.
5. El tipo de orquídea.
6. Cuan cálido es el clima donde este.
7. Cuan activamente está creciendo la planta.
8. A cuanto viento está expuesta la planta.

Por lo tanto, no es posible dar una regla general de riego que sea posible aplicar a todas las orquídeas. En general, observe las hojas y raíces de sus plantas; ellas le indicarán si la frecuencia y cantidad de agua de riego que está usando es la adecuada o no.

d.2.2.1 Riego excesivo

La mayoría de las personas están acostumbradas a cultivar plantas que crecen en tierra y no en corteza, musgo u otro medio en el cual se crecen las plantas de orquídea del tipo epífitas. Debido a que el sustrato de crecimiento es muy distinto a la tierra, los principiantes en el cultivo de las orquídeas deben aprender a regar adecuadamente sus plantas. Los principiantes normalmente riegan su primera planta de orquídea inmediatamente llegando a casa y, como el sustrato de crecimiento especial para orquídeas se verá seco unas pocas horas más tarde, ellos vuelven a aplicar un poco más de agua. Este ciclo continúa y el resultado es un riego excesivo, con un consecuente ahogamiento de la planta.

d.2.2.2 Riego deficiente

Muchas veces aquellos principiantes en el cultivo de orquídeas epífitas que han escuchado que estas plantas no deben ser regadas en forma excesiva, tienden a regar en forma deficiente. Ellos principalmente fumigan agua a las hojas de la planta y aplican muy poca agua en forma de riego al sustrato de crecimiento, el que no alcanza a humedecerse completamente.

Si su orquídea es regada en forma deficiente, entonces los síntomas visibles son: Lamentablemente, los síntomas visibles de falta de riego en la parte aérea de la planta son casi los mismos que los descritos anteriormente por exceso de riego. Sin embargo, si usted remueve suavemente a su planta de orquídea de la maceta para poder observar las raíces, entonces podrá tener una mejor evaluación del problema. Si las raíces están firmes, entonces es probable que el problema sea escasez de riego. Si las raíces están negras y como puré, entonces la planta está sufriendo exceso de riego.

1. La planta se ve marchita, pero tiene sus raíces sanas.
2. Las hojas más delgadas de algunas orquídeas, como las Miltonias, se plisan como acordeón y no se desarrollan. En este caso, aunque usted aumente el riego, las hojas afectadas no mejorarán; la situación es irreversible para las hojas afectadas, pero las hojas nuevas serán normales.
3. La planta se ve débil y las hojas se sueltan fácil.

d.2.2.3 Calidad del agua de riego

Las orquídeas tienen requerimientos claros sobre la calidad del agua de riego. Ellas no toleran aguas duras (con muchos minerales), con cloro ni otro tipo de contaminantes orgánicos. Por ello, lo ideal es regarlas con agua destilada o desmineralizada, aunque esto pueda ser más caro. Su planta le agradecerá este esfuerzo, creciendo mejor y floreciendo más.

Utilidad de ablandar el agua potable con ablandadores comerciales, antes de usarla para regar las plantas de orquídea:

Existen distintos tipos de ablandadores de agua en el mercado y el agua purificada por ellos puede ser o no adecuada para regar sus plantas de orquídea. Si el proceso de ablandamiento consiste en métodos de intercambio de iones, entonces el agua puede ser utilizada para regar plantas de orquídea. Sin embargo, si el agua es ablandada utilizando sales de sodio (sal), entonces no use esta agua para regar sus orquídeas. En este caso, el calcio y magnesio del agua son reemplazados por sodio, el que es tóxico para las orquídeas en las cantidades liberadas por este tipo de ablandadores.

d.2.3. Requerimientos de Humedad

Tabla 1. Requerimiento de humedad normal y natural de un invernadero.

Humedad Relativa Natural	65 % y 85 %.
Humedad Relativa Normal	60 % y 90 %

Fuente: Elaboración los autores.

En la tabla 1 se puede visualizar donde es el mejor ambiente de humedad relativa para las orquídeas epifíticas; junto con ello el rendimiento de humedad que normalmente tiene un invernadero. Una casa calefaccionada, con muchas alfombras y cortinas podría llegar a tener una humedad tan baja como del 5 %, lo que es inadecuado para cultivar orquídeas.

Si usted puede cultivar bien otras plantas al interior de su casa, entonces podrá cultivar bien plantas de orquídea. Pero si la humedad del aire es un problema para cultivar otro tipo de plantas, entonces sin duda también será un problema para cultivar orquídeas.

No se requiere un equipamiento sofisticado para mejorar la humedad del aire al interior de su casa. Bandejas de tamaño adecuado, llenas con agua y piedrecillas serán suficientes para mejorar la humedad del aire en torno a sus plantas de orquídea.

El agua que se evapore de la bandeja será suficiente para mejorar el ambiente seco de su casa. Sin embargo, asegúrese de que la base de la maceta o las hojas de la planta no estén en contacto directo con el agua de la bandeja.

d.2.3.1 Algunos consejos para aumentar la humedad ambiental:

1. En un invernadero, hay que humedecer las paredes y el piso.
2. En el interior de su casa, llenar una bandeja poco profunda con agua. Colocar piedrecillas dentro de la bandeja, junto con el agua, y asegurarse de poner la maceta con la planta de orquídea por sobre la bandeja; evitar que la base de la maceta quede en contacto directo con el agua y/o piedrecillas contenidas en la bandeja.
3. Poner sus plantas de orquídeas agrupadas o junto con otras plantas, pero no demasiado para que el aire pueda circular entre ellas. Cuando las plantas se

agrupan, ellas generan un micro clima benéfico a su alrededor, debido a su propia evaporación de agua por las hojas.

4. Utilizar humidificadores ambientales. Existen de muchos tipos y precios. Sin embargo, antes de invertir en uno, invierta primero en la compra de un higrómetro (medidor de humedad relativa del aire), de forma de verificar si realmente la humedad en torno a sus plantas de orquídea es muy baja, intentar primero los métodos indicados anteriormente.
5. En las épocas del año en que el aire es más seco, trasladar sus plantas a la cocina, baño u otro lugar donde se use agua caliente. La evaporación del agua caliente ayuda a humedecer el aire.

d.2.4. Requerimientos de Luz



Figura 4. El Sol y su influencia en el desarrollo de las Orquídeas.

Fuente: vyrsa.com/catalogo/agricultura/métodos/agro-stand.aspx

Las orquídeas requieren la mayor cantidad posible sin que esa cantidad de luz les pueda producir quemaduras. Una suficiente cantidad de luz determina el buen crecimiento y desarrollo.

Cada género tiene sus diferentes requerimientos o necesidades de luz, algunos géneros llegan a tolerar inclusive la luz directa del sol como en el caso de la schomburgkia.

Tabla 2. Requerimientos de luz de las orquídeas en sus diferentes etapas de crecimiento

2 500 lux	Plantas pequeñas
4 000 lux	Plantas jóvenes
8 000 lux	Plantas adultas
12 000 a 15 000 lux	Plantas próximas a floración

Fuente: Elaboración los autores.

Se puede determinar si la cantidad de luz que está recibiendo una planta es demasiada o muy poca, si las hojas adquieren un color verde oscuro, el desarrollo es deficiente y no hay floración la planta está recibiendo menos luz de la que necesita; si las hojas se tornan amarillentas tenemos un claro mensaje de que la planta está expuesta a cantidades excesivas de luz. Cuando la planta recibe las cantidades adecuadas de luz sus hojas tienen un color verde claro, usualmente con una superficie brillante en el nuevo crecimiento y florece regularmente.

Cuando hay mucha humedad y poco viento es mejor reducir la cantidad de luz. Cuando hay altas temperaturas es conveniente disminuir a la mitad o en una tercera parte la luz. Debido a lo anterior lo más recomendable sombrear algunas orquídeas en invernadero para asegurar que haya disponible la cantidad de luz apropiada para un buen crecimiento.

Otro de los objetivos de sombrear algunas orquídeas es evitar las altas temperaturas. En nuestro clima tropical no existe problema de fotoperiodos y no es necesario escoger las flores por estación. En otros países se requiere de un control estricto de temperatura y luz para obtener floraciones en determinadas épocas del año.

Tabla 3. Nombres de las orquídeas del Jardín Botánico “Reinaldo Espinosa”

Nombre Científico	Nombre Común
Cattleya	Cattleya, Catleya, Lirio de mayo, Lirio de San Juan.
Phalaenopsis	Orquídea alevilla, Orquídea mariposa.
Cypripediaceae	Zapatilla verde.
Góngora scaphephorus	Orquídea Góngora
Masdevallia ayabacana	Masdevallia
Oncidium	Dama danzante
Cymbidium	Cimbidio, Híbridos de Cymbidium
Catasetum	Cebolleta

Fuente: Elaboración los autores.

d.2.5.Requerimientos de Temperatura



Figura 5. La Temperatura un factor muy importante dentro del invernadero.

Fuente: vyrsa.com/catalogo/agricultura/métodos/agro-stand.aspx.

Este factor se relaciona con el lugar de origen de cada especie. En algunos casos es un factor muy determinante del cual puede depender la vida de las plantas. En nuestro país no se dan temperaturas extremas lo cual facilita el cultivo de las orquídeas durante todo el año.

Están los géneros que se desarrollan mejor en temperaturas cálidas las cuales se dan desde los 0 hasta los 1 000 metros sobre el nivel del mar. La contraparte son los géneros que se desarrollan en temperaturas más frescas las cuales se ubican de los 800 hasta los 1 500 metros sobre el nivel del mar.

Tabla 4. Clasificación de las orquídeas según el tipo de Clima.

Tipo de Clima	Orquídea	Temperatura
Clima Frio	Cymbidiums	10 °C
Clima Intermedio	Cattleyas, Oncidiums, Miltonias.	18 °C a 24 °C en el día. 13 °C a 16 °C en la noche
Clima Caliente	Vandas, Phalaenopsis.	21 °C a 30 °C en el día. 18 °C a 21 °C en la noche.

Fuente: Elaboración los autores.

d.2.6 Aireación



Figura 6. El Aire y su importancia en el cuidado de los cultivos.

Fuente: arqhys.com/fotos/ventilacion-invernaderos.html.

En la naturaleza una brisa constante rodea a las orquídeas, lo que les ayuda a sobrellevar las altas temperaturas, es así como la adecuada ventilación proporciona a las plantas una defensa contra las infecciones por bacterias y hongos, las cuales tienen su caldo de cultivo en la alta humedad y muy poco movimiento de aire; eso sí, se debe proteger a las orquídeas de fuertes ráfagas de viento y de viento muy seco y frío así como de viento muy caliente.

d.2.6.1 Aspectos a considerar para evitar la mala ventilación:

- Colocar las plantas con suficiente espacio entre ellas (que no se toquen) en un lugar donde haya libre circulación del aire, evitar espacios estrechos y sofocantes. De esta

manera las plantas se secan rápidamente y no den oportunidad a que los agentes causantes de enfermedades puedan prosperar.

- Los recipientes o tutores en que se cultivan las orquídeas deben permitir la entrada y salida del aire, ejemplos: canastas, vasijas especiales para orquídeas (con muchos huecos laterales) y troncos.

d.3. Redes Neuronales

d.3.1. Introducción.

Las computadoras digitales convencionales se rigen por una memoria de almacenamiento central, con un único y complicado procesador que se encarga de todo el trabajo, el cual hacen en forma secuencial siguiendo un algoritmo de control determinado, que es diseñado y programado por una persona o grupo de ellas. En este algoritmo pueden existir saltos a subrutinas, ya sea por software o por hardware, pero siempre hay un regreso al punto donde se produjo la llamada, para continuar con el trabajo secuencial. Son utilizadas para resolver tareas tales como:

- Problemas matemáticos y científicos.
- Creación, manipulación y mantenimiento de bases de datos.
- Comunicaciones electrónicas.
- Procesamiento de textos.
- Gráficos y autoedición.
- Funciones de control sencillas que dan inteligencia a los electrodomésticos, etc.

Sin embargo, esto sólo cubre una reducida área de las necesidades de los usuarios, pues la mayoría de los problemas prácticos son de carácter simbólico y no numérico. El hombre ha continuado entonces en la búsqueda de la máquina inteligente y de soluciones cada vez mejores. Todos sus esfuerzos se han agrupado en la rama de las Ciencias Computacionales denominada Inteligencia Artificial, la cual ha atravesado por diferentes etapas. Son conocidos entre otros, los algoritmos de búsqueda, los algoritmos genéticos, la lógica difusa, los sistemas expertos, las redes neuronales, etc.

En el enfoque de las RNA lo que se pretende es emular las funciones del cerebro humano simulando el comportamiento de la unidad básica de este: las neuronas. Una neurona humana es de 5 a 6 veces más lenta que una compuerta de silicio, parte más elemental de una microcomputadora.

Los eventos en un “chip” ocurren en el rango de los nanosegundos (10^{-9} s) mientras que los neurales tardan milisegundos (10^{-3} s). Sin embargo, la eficiencia energética del cerebro es de, aproximadamente, 10^{-16} J por operación por segundo, mientras que en las mejores computadoras de hoy día es del orden de 10^{-6} J por operación por segundo. Esta gran diferencia se debe al gran número de neuronas o células nerviosas en la corteza cerebral y a las masivas interconexiones que existen entre ellas. En 1990 se estimaban en alrededor de 100 billones las neuronas en la corteza cerebral y en alrededor de 60 trillones las sinapsis o conexiones entre ellas teniendo en cuenta que la salida de cada neurona está interconectada con otras 1 000 a 10 000 neuronas.

Al nacer, el cerebro tiene una gran estructura y la capacidad de construir sus propias reglas a través de lo que se conoce usualmente como experiencia. Aunque constantemente se aprende, el desarrollo más dramático del cerebro ocurre en los dos primeros años de vida. En esta primera etapa de desarrollo se forman alrededor de un millón de sinapsis por segundo. No obstante, en un cerebro adulto también se pueden formar nuevas sinapsis entre neuronas o modificarse las ya existentes.

d.3.2. El modelo biológico.

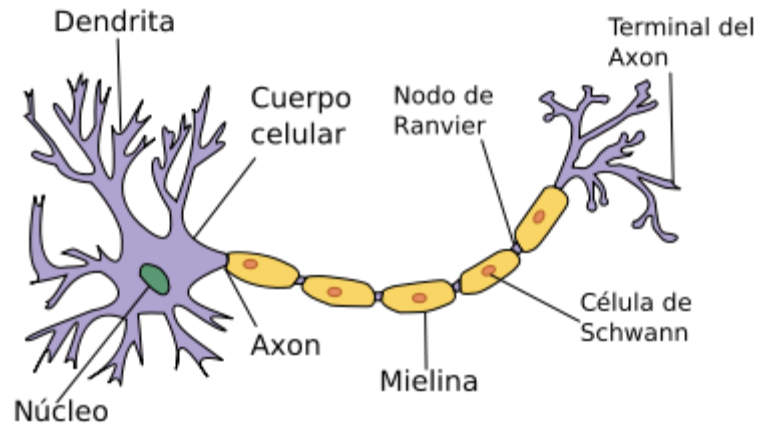


Figura 7. Diagrama esquemático de una neurona biológica.

Fuente: monografias.com/Articulo_Tecnico/Redes_Neuronales.

Una neurona está formada por el cuerpo celular y diferentes prolongaciones como se muestra en la figura 7.

d.3.2.1. El axón

Por aquí transitan los impulsos nerviosos o potenciales de acción desde el cuerpo celular hacia la siguiente célula. Los axones pueden agruparse y formar lo que comúnmente llamamos fibra nerviosa. La terminación axonal tiene forma abultada y se llama botón presináptico, el cual contiene las vesículas sinápticas incluyendo en su interior a los neurotransmisores, que son sustancias químicas responsables de transmitir los mensajes a la neurona que le sucede.

d.3.2.2 Las dendritas

Con número y estructura variable según el tipo de neurona, transmiten los potenciales de acción desde las neuronas adyacentes hacia el cuerpo celular o soma.

Se unen entre ellas a través de una unión discontinua llamada sinapsis. Si dicha sinapsis ocurre por contacto físico se trata de una sinapsis eléctrica, y si es a través de una hendidura se llama sinapsis química.

Son células excitables especializadas para la recepción de estímulos y la conducción del impulso nervioso.

Las neuronas se hallan en el encéfalo, la médula espinal y los ganglios nerviosos y están en contacto con todo el cuerpo. A diferencia de la mayoría de las otras células del organismo, las neuronas normales en el individuo maduro no se dividen ni se reproducen (como una excepción las células olfativas sí se regeneran). Los nervios mielinados del sistema nervioso periférico también tienen la posibilidad de regenerarse a través de la utilización del neurolema, una capa formada de los núcleos de las células de Schwann.

Es importante reconocer que en un principio las redes neuronales fueron desde su origen propuestas para modelar la actividad del cerebro humano, pero este posee una mayor complejidad de la que el modelo puede capturar. Entonces, los modelos neuronales computacionales no son fieles representaciones de sistemas biológicos reales. A pesar de la intensa investigación en la neurobiología y la psicología, muchas importantes interrogantes de cómo el cerebro funciona siguen sin encontrar una respuesta (TURBAM Efraim, 2001).

Tabla 5. – Relaciones entre redes neuronales biológicas y artificiales

Red Neuronal Biológica	Red Neuronal Artificial
Cuerpo Celular	Nodo
Dendritas	Entrada
Axón	Salida
Sinápsis	Peso
Baja Velocidad	Alta Velocidad
Muchas neuronas (10 a la 9)	Pocas neuronas (desde una docena a cientos de miles)

Fuente: monografias.com/Articulo_Tecnico/Redes_Neuronales.

d.3.2.3. Entradas

Cada entrada corresponde a un simple atributo. Por ejemplo, si el problema consiste en decidir si aprobar o no un préstamo, algunos atributos podrían ser el nivel de ingreso del que se le aplique, su edad y su tipo de vivienda. El valor numérico, o la representación de un atributo es la entrada de la red. Muchos tipos de datos, como texto, imágenes o la voz, pueden ser usados como entradas, pero

un pre- procesamiento puede ser necesitado para convertir la información en entradas significativas desde información simbólica, o para escalar la información. La computación neuronal sólo puede procesar valores numéricos, si un problema involucra atributos cualitativos, debe ser pre-procesada.

d.3.2.4. Salidas

Las salidas de una red contienen la solución a un problema. Por ejemplo, en el caso de la aprobación del préstamo puede ser sí o no. La red neuronal asigna valores numéricos, como 1 para SI y 0 para NO. El propósito de la red neuronal es computar los valores de la salida. A menudo, se requiere un post proceso de las salidas.

d.3.2.5. Pesos

Los pesos son elementos claves en una red neuronal, expresan la fuerza relativa (o valor matemático) de la información de entrada o de las muchas conexiones que transfieren información de una capa a otra. En otras palabras, los pesos expresan la relativa importancia de cada entrada a un elemento del proceso y en última instancia las salidas. Los pesos son cruciales porque mantienen patrones almacenados de información, y es mediante repetidos ajustes de los mismos que las redes aprenden.

d.3.3. El Modelo Artificial

Las RNA, también conocidas como Neuro-computadoras, procesadores paralelos o sistemas conexionistas, consisten en un conjunto de células de cálculo simples conocidas como unidades de procesamiento, neuronas artificiales o simplemente neuronas, que están interconectadas entre sí y que tienden a simular el sistema nervioso del cerebro humano.

Son máquinas que se diseñan para modelar la forma en que el cerebro realiza una tarea particular, por ello se dice que están biológicamente inspiradas y a ello deben su nombre. Son realizadas usando componentes electrónicas aunque también pueden ser simuladas por un software en una microcomputadora digital. Como se verá más adelante, las RNA procesan la información que les llega directamente por sus entradas y la almacenan en una memoria local que poseen, denominada peso sináptico.

Se asemejan al cerebro en dos aspectos fundamentales:

1. Adquieren el conocimiento a través de un proceso de aprendizaje para lo cual se utilizan los llamados algoritmos de aprendizaje, cuya función es modificar los pesos sinápticos de la red en forma ordenada, de manera que se logre el objetivo deseado.
2. Las conexiones interneurónicas, conocidas como pesos sinápticos o simplemente pesos, son usadas para almacenar el conocimiento.

d.3.3.1 Características de las Redes Neuronales Artificiales.

Las RNA han sido desarrolladas en una amplia variedad de configuraciones, sin embargo, a pesar de esta aparente diversidad tienen en común las siguientes características fundamentales:

- **Analogía neurobiológica.** Como se dijo antes están compuestas por infinidad de elementos, mucho más simples, llamados neuronas artificiales, interconectados en una red similar a la que forman las neuronas en el cerebro humano. Luego, se pretende que realicen la mayoría de las funciones elementales de las neuronas biológicas, en forma análoga a éstas. Son una prueba de que el procesamiento paralelo es posible físicamente, además de ser muy potente y rápido.
- **Son capaces de aprender.** Se dice que aprenden de la experiencia. El procedimiento usado para realizar el proceso de aprendizaje se llama algoritmo de aprendizaje y su función es modificar los pesos sinápticos de la red en forma ordenada, de manera que se cumpla el objetivo deseado. Son capaces de aprender por ejemplos, por lo que esta tarea resulta relativamente sencilla.
- **Pueden modificar su propia topología.** Si trabajan “on-line” las redes neuronales pueden modificar su comportamiento en respuesta a sus alrededores a similitud de lo que ocurre en el cerebro, esto está motivado por el hecho de que unas neuronas pueden “morir” y nuevas conexiones sinápticas desarrollarse. Para un conjunto de entradas y sus correspondientes salidas ellas se autoajustan para producir una respuesta adecuada.

- **Abstracción.** Es la habilidad de extraer un ideal a partir de entradas imperfectas. Por ejemplo, una red puede ser entrenada en una secuencia de versiones distorsionadas de la letra A y, después de un adecuado entrenamiento, al aplicar cualquiera de esas secuencias distorsionadas la red producirá una letra A perfectamente formada, es decir, la red produce algo que nunca antes ha tenido.
- **Son capaces de generalizar ejemplos previos a otros nuevos** comparando todo lo nuevo que se les presenta con experiencias anteriores, es decir, una vez entrenada la red ella puede producir respuestas razonables ante entradas no utilizadas durante el entrenamiento, por ejemplo, si como entrada se da una muestra similar a una previamente aprendida por la red o bien se presenta de forma distorsionada, el sistema sigue siendo capaz de reconocerla.
- **No linealidad.** Una neurona es básicamente un elemento no lineal, luego una red es por sí misma no lineal, pero con una no linealidad especial pues está distribuida por toda la red. Esta es una característica altamente importante por la gran potencialidad que da a la red para el trabajo con sistemas físicos no lineales.
- **Adaptabilidad.** Son capaces de adaptar su salida a los cambios en el medio que las rodea. Si son entrenadas para trabajar en un medio específico pueden ser fácilmente entrenadas para adaptarse a los cambios en el medio circundante. Esta capacidad las hace ideales para el trabajo en control adaptativo, procesamiento de señales, reconocimiento y clasificación de patrones, etc.
- **Son robustas,** pues siempre responden con alguna salida incluso en el caso de que se le presenten entradas que no ha visto nunca antes.
- **Alta tolerancia a los fallos.** Si una neurona de la red o sus conexiones se dañan, debido a la naturaleza distribuida de la información la red continúa trabajando. En dependencia del daño, la red presenta una natural degradación en su comportamiento pero no un fallo catastrófico.

d.4 La Neurona Artificial

d.4.1 Estructura básica.

La neurona artificial es la unidad fundamental de procesamiento de la información de cualquier modelo de RNA y a partir de ahora la identificaremos con la letra k . Como se dijo antes, fue diseñada para imitar las características de primer orden de la neurona humana. Su modelo se muestra en la figura 2 y en ella se identifican tres elementos básicos:

1. **Un conjunto de sinapsis o conexiones provenientes de otras neuronas o del exterior.** Cada conexión está caracterizada por un peso o fortaleza propio, cuya función es análoga a la de la unión sináptica de la neurona biológica, y que se identificará, a partir de este momento, con la letra W con dos subíndices, el primero se refiere a la neurona analizada y el segundo a la entrada de la sinapsis a la cual se refiere el peso. El peso W_{kp} , por ejemplo, corresponde a la conexión entre la neurona k y la neurona p de la capa previa, y es positivo si la sinapsis asociada es excitadora, negativo, si es inhibidora y 0 si la conexión no existe.
2. **Un sumador (Σ)** para sumar las señales de entrada, pesadas por sus respectivas conexiones. Específicamente, una señal X_p que se encuentra a la entrada de la sinapsis p que la conecta a la neurona k , es multiplicada por el peso sináptico W_{kp} . Estas entradas ponderadas son sumadas constituyendo esta operación una combinación lineal, que se identifica en la figura como NET_k .
3. **Una función de activación (F)** para limitar la amplitud de la salida (A_k) de la neurona k . Esta función limita el rango de amplitud permisible de la señal de salida a valores finitos, motivo por el cual también se conoce como función ‘squashing’. Típicamente el rango de amplitud normalizado de salida de una neurona es el intervalo cerrado $[0,1]$ o el intervalo $[-1,1]$.

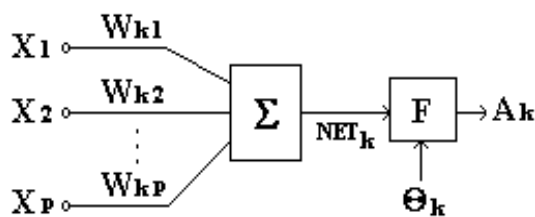


Figura 8. Modelo no lineal de una neurona artificial.

Fuente: monografias.com/Articulo_Tecnico/Redes_Neuronales.

Las señales de entrada (X) son normalmente continuas aunque también pueden ser discretas.

El modelo de la neurona puede además incluir una entrada adicional aplicada externamente a la función de activación. Si es negativa es conocida como entrada de umbral y denotada por Θ_k , como se muestra en la figura 8, y tiene el efecto de disminuir la entrada neta a la función de activación F de la neurona. Es un parámetro externo a la neurona k . En términos matemáticos, la salida (A_k) en este caso será igual a:

$$A_k = F(\text{NET}_k - \Theta_k) \quad (1).$$

Dónde:

F es la función de activación.

Si la entrada adicional es positiva es más conocida como bias, y hay autores que acostumbran a denotarlo por B_k para diferenciarlo del umbral ya que realiza el efecto contrario, es decir, aumenta la entrada neta a la función de activación. Matemáticamente la salida correspondiente es:

$$A_k = F(\text{NET}_k + B_k) \quad (2).$$

El trabajo de la neurona artificial puede resumirse entonces de forma muy simple:

Una neurona k recibe un conjunto de entradas X_1, X_2, \dots, X_p (cada una de las cuales representa la salida de otra neurona de una capa previa o, simplemente, proviene del exterior), y calcula un valor de salida A_k (que envía a otras neuronas).

Para ello, cada entrada es multiplicada por un peso (análogo a la fortaleza sináptica de la conexión que une la neurona k con la señal X), que es la potencia con que dicha entrada afecta a la neurona, y todas las "entradas pesadas" son entonces sumadas para determinar el nivel de activación o combinación lineal de la neurona (NET_k), siendo:

$$NET_k = \sum_{j=1}^p W_{kj} X_j \quad (3).$$

donde $W_{k1}, W_{k2}, \dots, W_{kp}$, son los pesos sinápticos de las conexiones que llegan a la neurona k .

Finalmente, una función de activación determina el valor de salida, utilizando la expresión vista anteriormente para la salida A_k .

De manera equivalente las ecuaciones anteriores se pueden reformular adicionando una nueva sinapsis de entrada $X_0 = +1$ ó -1 y peso igual a B_k o Θ_k , respectivamente, como:

$$V_k = \sum_{j=0}^p W_{kj} X_j \quad y \quad A_k = F(V_k) \quad (4) \text{ y } (5).$$

Y el modelo de la neurona k queda como en la figura 9, equivalente matemáticamente al anterior.

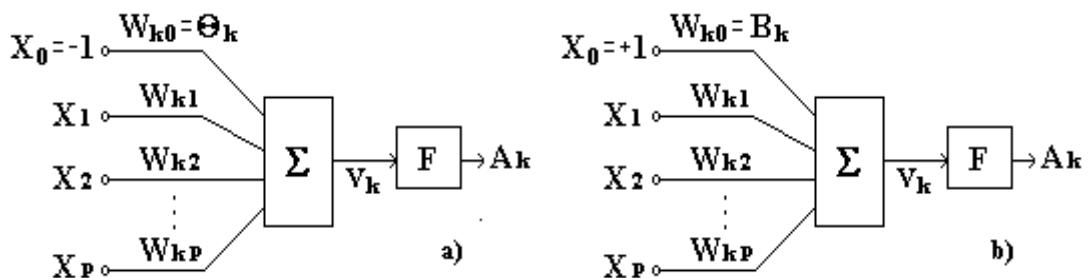


Figura 9. Modelo no lineal de una neurona artificial.

Fuente: monografias.com/Articulo_Tecnico/Redes_Neuronales.

a) Con umbral adicional.

b) Con bias adicional.

Nota: El uso del umbral tiene el efecto de aplicar una transformación afín a la salida de la combinación lineal (NET_k) en el modelo, como se muestra por:

$$V_k = NET_k - \Theta_k \quad (6).$$

$$V_k = NET_k + B \quad (7).$$

En particular, dependiendo de si se trata de umbral ($\Theta_k < 0$) o de bias ($\Theta_k > 0 = B_k$), la relación entre el nivel de activación interna efectiva (V_k), también llamado potencial de activación, y la combinación lineal de la salida NET_k , es modificada como se muestra en la figura.

d.4.2. Clasificación de las Redes Neuronales Artificiales

Aunque una simple neurona puede realizar ciertas tareas, sobre todo de reconocimiento de patrones simples, la potencia está realmente cuando las neuronas se conectan formando redes. La manera en la cual esto se hace está íntimamente relacionada con el algoritmo de aprendizaje que será utilizado para entrenar la red. En cualquier caso se pueden encontrar tres tipos de neuronas:

- Aquellas que reciben estímulos externos y que toman la información de la entrada. Están relacionadas con el aparato sensorial en el sistema biológico, mientras que en el artificial pueden ser señales provenientes de sensores o de otros sectores del sistema. Se llaman ***nodos de entrada*** y no realizan cálculos como los explicados para las neuronas.
- Las neuronas de entrada transmiten la información a ciertos elementos internos que se ocupan de su procesamiento. Es en las sinapsis y neuronas correspondientes a este segundo nivel donde se genera cualquier tipo de representación interna de la información. Puesto que no tienen contacto con el exterior, estos elementos se denominan ***neuronas ocultas***.
- Una vez finalizado el período de procesamiento, la información llega a las unidades de salida, cuya misión es dar la respuesta del sistema, estas señales pueden controlar directamente potencias u otros sistemas. Las neuronas son conocidas como ***neuronas de salida***.

En general se pueden identificar diferentes arquitecturas neuronales dependiendo de la forma en que se conectan las neuronas. Ellas son:

- Redes de alimentación hacia adelante o feedforward.
- Redes recurrentes.
- Redes con estructuras enrejadas.

Aunque ya se ha dicho, debe señalarse que alrededor del 90 % de las RNA que se utilizan en ingeniería hoy día, en nuestras aplicaciones, tienen arquitectura feedforward.

d.5. Redes de alimentación hacia adelante o Feedforward.

Estas redes, como su nombre lo dice, no tienen conexiones de realimentación, es decir, el flujo de señales siempre va hacia adelante, desde la entrada hacia la salida. Pueden ser simple capa o multicapa. Se conoce como **capa o nivel** a un conjunto de neuronas cuyas entradas provienen de la misma fuente (que puede ser otra capa de neuronas o un conjunto de elementos sensores) y cuyas salidas se dirigen al mismo destino (que puede ser otra capa de neuronas o la salida de la red). Se asume que los pesos de las conexiones están asociados con las neuronas de la capa siguiente, luego una capa consiste de un conjunto de pesos y de las neuronas subsiguientes. Veamos a continuación las características de cada una de estas RNA.

d.5.1. Redes simple capa.

Es la red más sencilla y agrupa varias neuronas colocadas en una capa, como se muestra en la figura 9. Los pequeños nodos cuadrados de la izquierda sirven sólo para distribuir las entradas, ellos no realizan cálculos por lo que no constituyen neuronas artificiales como las explicadas hasta ahora, también se agrupan formando una capa a la que muchos autores se refieren como capa 0.

Un patrón de entradas X tiene cada uno de sus elementos conectados a cada neurona artificial a través de pesos separados. La salida de cada neurona simple es la suma pesada de las entradas, en las cuales se incluye el bias o umbral, afectada por la función de activación.

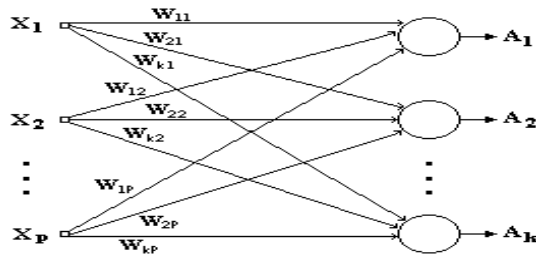


Figura 10. Diagrama de flujo de una red neuronal simple capa.

Fuente: monografias.com/Articulo_Tecnico/Redes_Neuronales.

La programación se realiza en forma matricial, por lo que se puede plantear que:

$$A = F(W * X, B), \quad (8).$$

Dónde:

F es la función de activación.

W: es la matriz de los pesos sinápticos, de orden (k , p) es decir, k (número de neuronas de la capa) filas y p (número de entradas) columnas, de forma tal que W_{21} por ejemplo, corresponde al peso de la conexión de la primera entrada con la segunda neurona de la capa.

X: es la matriz de las entradas, de orden (p , z) es decir, p (número de entradas) filas y z (número de patrones de entrada) columnas, de forma tal que X_{53} por ejemplo, corresponde a la entrada número 5 del tercer patrón de entrenamiento.

B: es un vector columna donde cada fila corresponde al bias de la neurona correspondiente de la capa.

A: es la matriz de las salidas de la red, de orden (k , z) es decir, k (número de neuronas de la capa) filas y z (número de patrones de entrada) columnas, de forma tal que A_{32} por ejemplo, corresponde a la salida de la neurona número 3 de la capa al aplicar a la red el segundo patrón de entrenamiento.

d.5.2. Redes multicapa.

Se distinguen por la presencia de una o más capas ocultas. Consisten de un conjunto alternado de neuronas y pesos. Están formadas por la conexión en cascada de un grupo de capas simples, siendo la salida de una capa la entrada de la capa siguiente, de manera

que los nodos fuente en la capa de entrada suministran los respectivos elementos del patrón de activación (vector de entrada) el cual constituye las señales aplicadas a las neuronas de la primera capa oculta, las señales de salida de estas son usadas como entradas de la siguiente capa y así ocurre en toda la red. El conjunto de señales de salida de las neuronas de la capa de salida constituye la respuesta de la red al patrón de activación suministrado. Los nodos de cálculo son entonces correspondientemente llamados neuronas o unidades ocultas o neuronas de salida.

$$p - h_1 - h_2 - \dots - q \quad (9).$$

Dónde:

p es el número de nodos fuente, h_i es el número de neuronas ocultas de la capa i , y

q es el número de neuronas de la capa de salida.

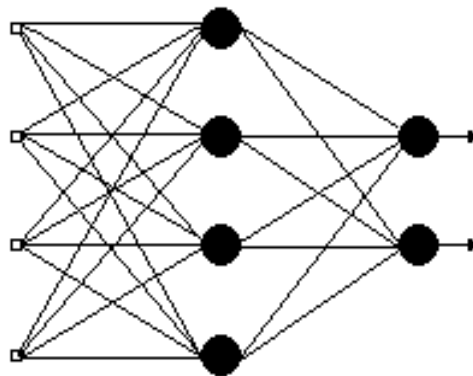


Figura 11. Gráfico arquitectónico de una red neuronal de dos capas.

Fuente: Monografias.com/Articulo_Tecnico/Redes_Neuronales.

Se dice que una red multicapa está totalmente conectada si cada neurona de una capa de la red está conectada a cada una de las neuronas de la capa siguiente, en otro caso están parcialmente conectadas. Un caso particular de esta última es la red localmente conectada, un ejemplo de la cual se muestra en la figura 11 en la que existe una simple capa oculta cuyas neuronas se conectan a un conjunto local de nodos fuente que se encuentran situados en sus alrededores inmediatos y una capa de salida en que cada neurona está conectada a un conjunto local de neuronas ocultas. Como se observa, la red feedforward multicapa localmente conectada tiene una estructura especializada en la cual cada neurona oculta responde a las variaciones locales de la señal fuente.

En una red multicapa las capas de neuronas ocultas son particularmente importantes cuando el tamaño de la capa de entrada es grande.

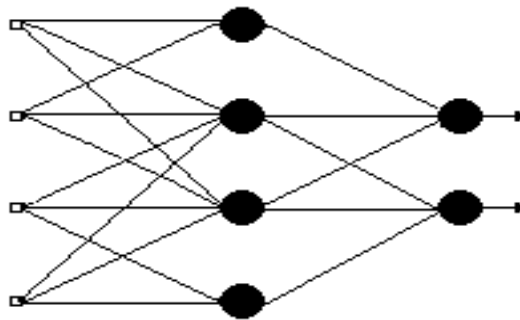


Figura 12. Gráfico arquitectónico de una red feedforward localmente conectada.

Fuente: Monografias.com/Articulo_Tecnico/Redes_Neuronales.

Si se utilizan funciones de activación lineales con ganancia $a=1$ entonces se pueden multiplicar los pesos de las distintas conexiones en cascada dando lugar a una red equivalente de una sola capa. Estas redes tienen grandes limitaciones por lo que se utilizan frecuentemente las funciones de activación no lineales sigmoidales explicadas anteriormente.

Para el trabajo matemático o de programación de una red multicapa en cada capa de neuronas se aplican las mismas matrices explicadas para la red monocapa.

d.5.3. Redes recurrentes.

Las redes explicadas anteriormente no tienen conexiones de realimentación de ningún tipo y son llamadas no recurrentes o redes de alimentación hacia adelante (feedforward). Son de considerable interés y ampliamente utilizadas fundamentalmente en aplicaciones de ingeniería. En estas redes las salidas son solamente determinadas por las entradas y los valores de los pesos. Las redes que contienen al menos una conexión de realimentación son llamadas recurrentes. En estos casos la salida es determinada por las entradas corrientes y por las salidas previas, es decir, el estado actual de la salida depende en parte de las salidas previas por lo que se dice que tienen memoria.

Se ha demostrado que la presencia de los lazos de realimentación tiene un profundo impacto en la capacidad de aprendizaje de la red así como en su comportamiento posterior, sin embargo incluyen el uso de ramas de retardo (denotadas por z^{-1} en las

figuras) que influyen fuertemente en el comportamiento no lineal de las neuronas y de la red en su conjunto.

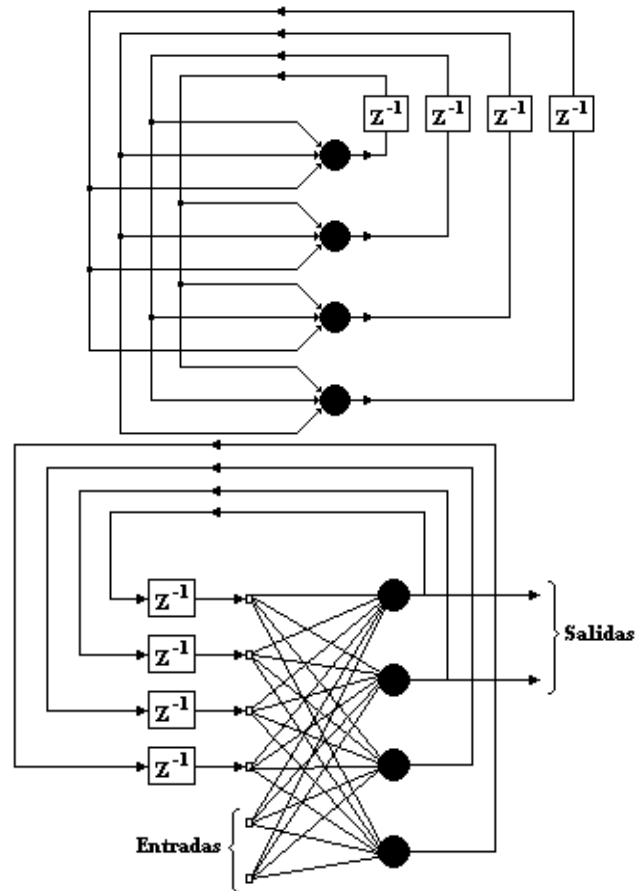


Figura 13. Ejemplo de red recurrente sin auto realimentación.

Fuente: Monografias.com/Articulo_Tecnico/Redes_Neuronales.

d.5.4 Redes con estructura enrejada.

Son arreglos de neuronas con un correspondiente conjunto de nodos fuente que suministran las señales de entrada. Son redes feedforward con las neuronas “arregladas” en filas y columnas (ver figura 14).

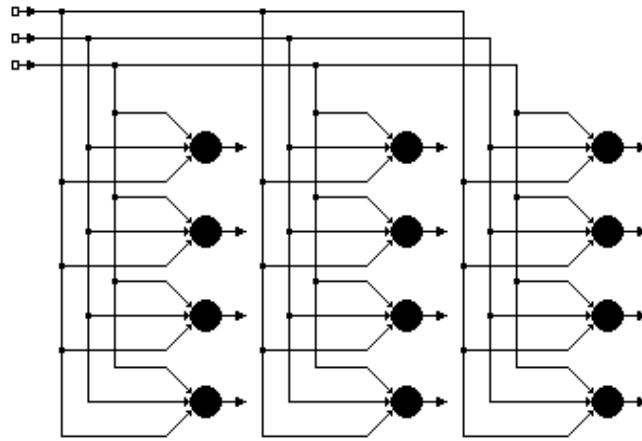


Figura 14. RNA con estructura enrejada de 4 dimensiones y una capa de entrada con 3 nodos fuente.

Fuente: Monografias.com/Articulo_Tecnico/Redes_Neuronales.

d.6. Entrenamiento de las Redes Neuronales Artificiales

El funcionamiento de una red consta de dos etapas: la de entrenamiento o aprendizaje y la de reconocimiento. Es, precisamente, la capacidad de aprender, una de las propiedades más interesantes y de mayor significado práctico de las redes neuronales. Una red aprende de sus alrededores a través de un proceso iterativo en el cual se producen cambios en los valores de los pesos sinápticos y de los umbrales o biases de sus neuronas y que se llama algoritmo de aprendizaje. El tipo de aprendizaje está determinado por la manera en la cual tiene lugar el cambio de esos parámetros.

Inicialmente no se conoce el conjunto de pesos sinápticos que permite obtener la salida deseada para los diferentes estímulos aplicados, y se supone uno de forma aleatoria. La etapa de entrenamiento se inicia cuando se presentan a la red de forma secuencial los vectores o patrones de entrada con el objetivo de ajustar los pesos, de manera que la aplicación de un conjunto de entradas produzca el conjunto de salidas deseado con un error menor que el permisible. Durante el entrenamiento, en cada paso del proceso iterativo, se cambian los pesos de la red que gradualmente convergen a valores tales que permitan que cada vector de entrada produzca el vector de salida deseado. El aprendizaje incluye un proceso estadístico por lo que no existe una única salida para los pesos y los umbrales o biases de la red.

Matemáticamente durante el aprendizaje:

$$W_{kj}(n+1) = W_{kj}(n) + \Delta W_{kj}(n) \quad (10).$$

Dónde:

n es el número de la iteración.

$W_{kj}(n)$ es el peso sináptico entre las neuronas k de la capa actual y j de la capa previa.

$\Delta W_{kj}(n)$ es el ajuste aplicado a los pesos en el paso n .

Los algoritmos de entrenamiento se clasifican en supervisados y no supervisados.

Luego de la etapa de aprendizaje la red es capaz ya de entrar a la etapa de reconocimiento en la cual puede presentarse a la red un vector nuevo que no formó parte del entrenamiento, y entonces la red deberá decidir a cuál de los vectores de entrada conocidos se parece más y dar un vector de salida en consecuencia.

d.6.1. Entrenamiento supervisado.

Incluye la modificación de los pesos sinápticos de la RNA aplicando un conjunto de patrones de entrenamiento o ejemplos de tareas, cada ejemplo consiste de una señal de entrada única y su correspondiente salida deseada. Durante el entrenamiento los pesos se modifican de forma que se minimice la diferencia entre la respuesta deseada y la salida actual producida por la señal de entrada, de acuerdo con un criterio estadístico apropiado. El entrenamiento es repetido para muchos ejemplos del conjunto hasta que la red permanezca en estado estable donde no hay un cambio significativo de los pesos. Los ejemplos de entrenamiento son reaplicados durante el entrenamiento pero en orden diferente cada vez.

Requiere que cada vector de entrada tenga su vector objetivo representando la salida deseada; juntos son llamados “par de entrenamiento”. Por eso se dice que es supervisado siendo la salida deseada “el maestro”.

Cuando es procesado el último vector de entrada se pregunta si el error es menor que el permisible y en caso negativo se repite nuevamente el proceso para todos los patrones de entrada. En caso positivo termina el proceso de entrenamiento y queda fija la matriz de pesos de interconexión.

A pesar de tener éxito en muchas aplicaciones, entre ellas las de ingeniería, el algoritmo supervisado ha sido criticado por ser poco adecuado biológicamente; es difícil concebir un mecanismo de entrenamiento en el cerebro que compare las salidas actuales y la deseada.

d.6.2. Entrenamiento no supervisado.

En él no se requiere vector objetivo para la salida, y no se realiza, por consiguiente, una comparación con un ideal predeterminado. El conjunto de entrenamiento consiste sólo de un vector de entrada. El algoritmo de entrenamiento modifica los pesos de la red para producir un vector de salida. La aplicación de un vector de entrenamiento o de uno similar a él producirán el mismo vector de salida. El proceso de entrenamiento en consecuencia extrae las propiedades estadísticas de un conjunto de entrenamiento y grupos de vectores similares en clases. Aplicando un vector de una clase dada a la entrada se producirá un vector de salida específico, pero no hay forma de determinar antes del entrenamiento cual patrón de salida específico será producido por una clase de vector de entrada dado. De aquí se deduce que la salida de tales redes tienen generalmente que ser transformadas en una forma comprensible posteriormente al proceso de entrenamiento. Esto no representa un problema serio ya que es usualmente simple identificar la relación entrada - salida establecida por la red.

e. MATERIALES Y MÉTODOS

e.1 Materiales

Los materiales utilizados para el presente proyecto investigativo se detallan a continuación:

e.1.1 Equipo de medición Fluke 975 Air Meter

Es la herramienta que utilizamos para la obtención de datos, cuyas características se muestra a continuación:



Figura 15. Fluke 975 Air Meter

Fuente: fluke.com/fluke/Air-Testers/Fluke975

e.1.1.1 Características:

- Mide, registra y visualiza de forma simultánea los valores de temperatura, humedad, CO₂ y CO en su nítida pantalla LCD con retroiluminación.
- Caudal y velocidad del aire con un solo botón y sonda disponible
- Temperatura de bulbo húmedo y de punto de rocío
- Cálculo del porcentaje de aire exterior
- Función de calibración en campo de CO₂ y CO
- Función de autocomprobación al encenderse
- Retroiluminación automática
- Compensación automática ante cambios de presión barométrica

- Valores mínimo, máximo y promedio de todos los parámetros medidos y calculados.
- Interfaz de usuario en varios idiomas.
- Unidades métricas y estándar
- Alarmas visuales y acústicas para umbrales.
- Gran capacidad de registro de datos, tanto de forma continua como discontinua, que pueden descargarse a un PC gracias a la interfaz USB
- Bloqueo del teclado por seguridad
- Software Fluke View Forms®.

e.1.2 Matlab

MATLAB (MATrixLABoratory) es un lenguaje de alto nivel y un ambiente interactivo creado por MathWorks. Está diseñado para realizar rápida y eficientemente tareas matemáticas computacionalmente intensivas.

MATLAB integra la computación matemática con funciones de visualización y un lenguaje de gran eficacia a fin de ofrecer un entorno flexible para la realización de computación técnica. La arquitectura abierta facilita el uso de MATLAB y de los productos acompañantes para explorar datos, crear algoritmos y herramientas a medida que permitan obtener de manera rápida la información y confieran ventajas competitivas.

Profesionales técnicos de todo el mundo confían en MATLAB para acelerar su investigación, reducir el tiempo invertido en el análisis y desarrollo y reducir los costos del proyecto, además de generar soluciones efectivas. El entorno MATLAB estimula la creatividad y permite comprobar y comparar rápidamente múltiples alternativas, permitiendo así la obtención de mejores soluciones.

e.1.2.1 Herramientas

MATLAB dispone de herramientas para:

- Adquisición de datos
- Exploración y análisis de datos
- Visualización y procesamiento de imágenes
- Prototipos y desarrollo de algoritmos
- Modelado y simulación
- Programación y desarrollo de aplicaciones

Una de las herramientas más poderosas de MATLAB es Simulink, un entorno gráfico para modelar, simular y analizar sistemas dinámicos tanto en tiempo continuo como discreto.

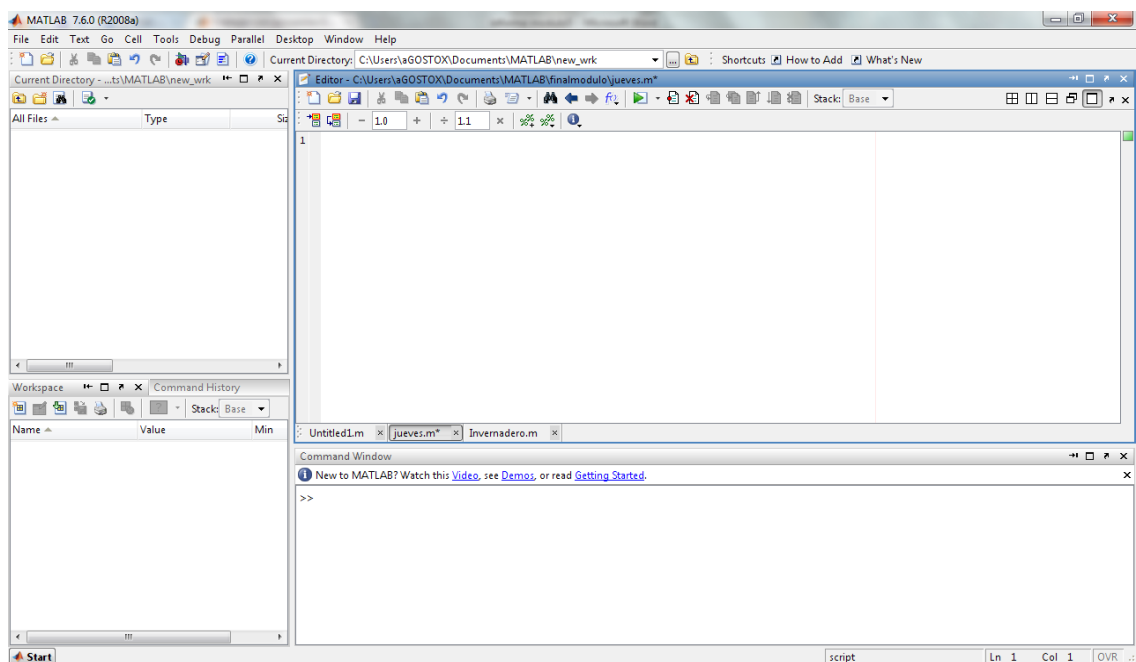


Figura 16. Entorno de MATLAB, a la izquierda las ventanas de directorio y workspace, a la derecha la ventana de comandos.

Fuente: mathworks.es/programs/trials/trial_request.html

e.1.2.2 Toolbox de Matlab NTSTOOL

La predicción es una especie de filtro dinámico, en el que se utilizan los valores pasados de una o más series de tiempo para predecir valores futuros. Redes neuronales dinámicas, que incluyen líneas de retardo que se utilizan para el filtrado no lineal y de predicción.

Hay muchas aplicaciones para la predicción. Por ejemplo, un analista financiero podría querer predecir el valor futuro de una acción, bono u otro instrumento financiero. Un ingeniero puede ser que desee para predecir el fracaso inminente de un motor a reacción.

Los modelos de predicción también se utilizan para la identificación del sistema, en la que se construye modelos dinámicos de sistemas físicos. Estos modelos dinámicos son importantes para el análisis, la simulación, la vigilancia y el control de una variedad de sistemas, incluyendo sistemas de fabricación, procesos químicos, la robótica y los sistemas aeroespaciales.

Herramientas de series de tiempo con Redes Neuronales (NTSTOOL).

La función `ntstool` abre una interfaz gráfica de usuario, que permite resolver problemas de series de tiempo con tres tipos de redes neuronales figura 17:

- redes neuronales `NarX` (autorregresiva con entrada externa),
- redes `NAR` (neuronal autorregresiva), y
- tiempo de retardo con redes neuronales. Que sigue un formato similar a la función neuronal (`nftool`), la agrupación (`nctool`), y el patrón de herramientas de reconocimiento (`nprtool`).

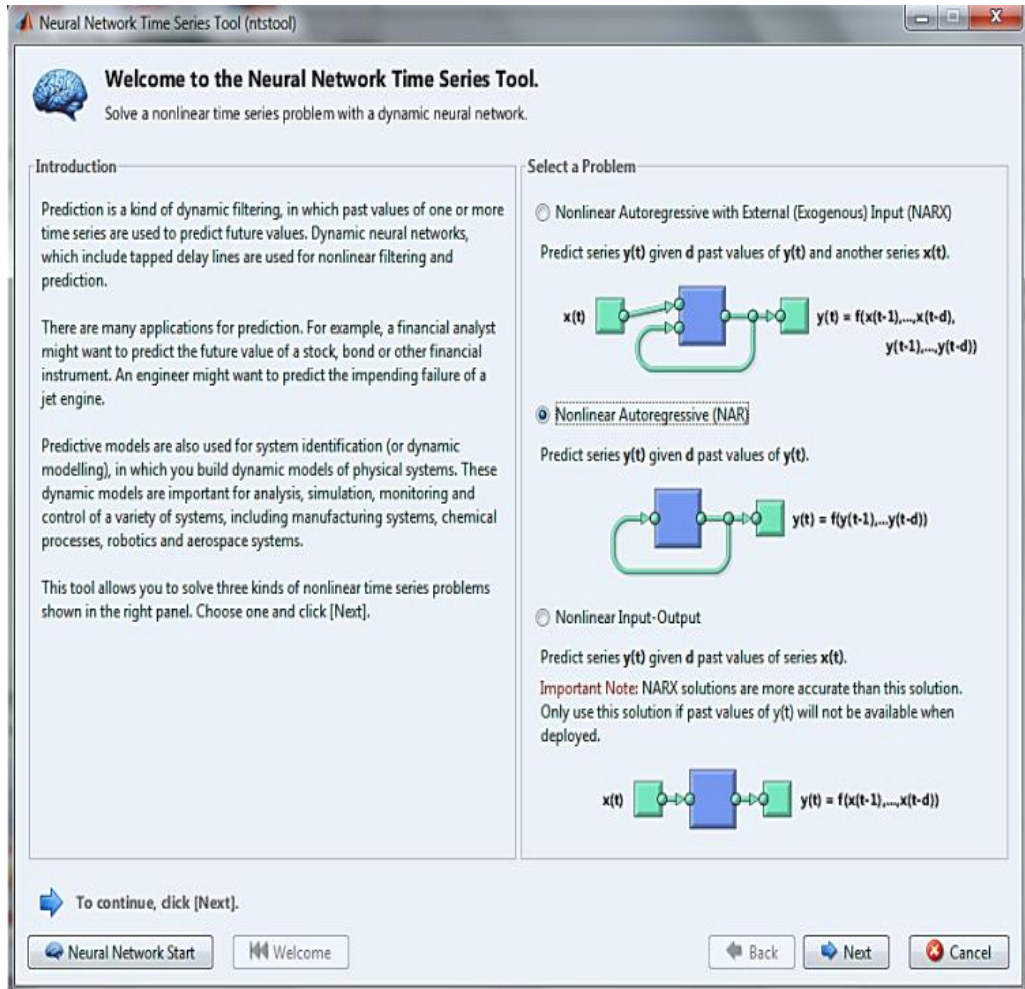


Figura 17. Entorno de MATLAB, a la izquierda las ventanas de directorio y workspace, a la derecha la ventana de comandos.

Fuente: mathworks.es/programs/trials/trial_request.html

e.1.3 Software Fluke View Forms

e.1.3.1 Características de Fluke View

Fluke View adquiere y documenta lecturas de los medidores digitales respaldados a través de puertos de comunicación IR (infrarrojo) o estándar en serie.

- Presenta los resultados adquiridos en gráficos y tablas.
- Permite guardar e imprimir los resultados.
- Permite recuperar y observar resultados guardados previamente.
- Permite el registro en tiempo real y directo a una PC conectado con medidores que respaldan el registro interactivo.
- Permite transferir las lecturas adquiridas a otras aplicaciones.
- Incluye plantillas de formularios estándar para presentar los resultados de forma profesional.
- Incluye el programa Fluke View Forms Designer que permite la creación de plantillas de formularios (no incluido en Fluke View Forms Basic) adaptadas a las necesidades del usuario.

e.1.3.2. Fluke View Forms Basic

Fluke View Forms Basic es una versión menos personalizada de Fluke View Forms, diseñada para proporcionar sólo características de comunicación básica de datos y de visualización en pantalla. Contiene un subconjunto de funciones proporcionadas por el producto completo de Fluke View Forms, y puede utilizarse sólo con un número limitado de plantillas de formulario estándar.



Figura 18. Entorno de Fluke View Forms Basic.

Fuente: fluke.com/fluke/Air-Testers/Fluke975

e.1.4 Recursos informáticos

Computadora portátil, necesaria para receptar los datos que se adquieren en el equipo de medición, y a su vez procesar para la creación de la red y la predicción de las variables. Se muestra en la figura 19.



Figura 19. Computadora para adquisición y procesamiento de datos.

Fuente: loja.com/computador-portátil hp.

e.2 Métodos

e.2.1 Técnicas de modelado y recolección de datos

- Se empleará la técnica de la observación y la toma de datos utilizando un instrumento de medición denominado Fluke 975 Air Meter.
- El Software Fluke View Forms® que le permite documentar, almacenar y analizar lecturas individuales o conjuntos de medidas y, a continuación, convertirlas en documentos con apariencia profesional para procesar los datos.
- El software Matlab para el procesamiento, entrenamiento y validación de los modelos generados.

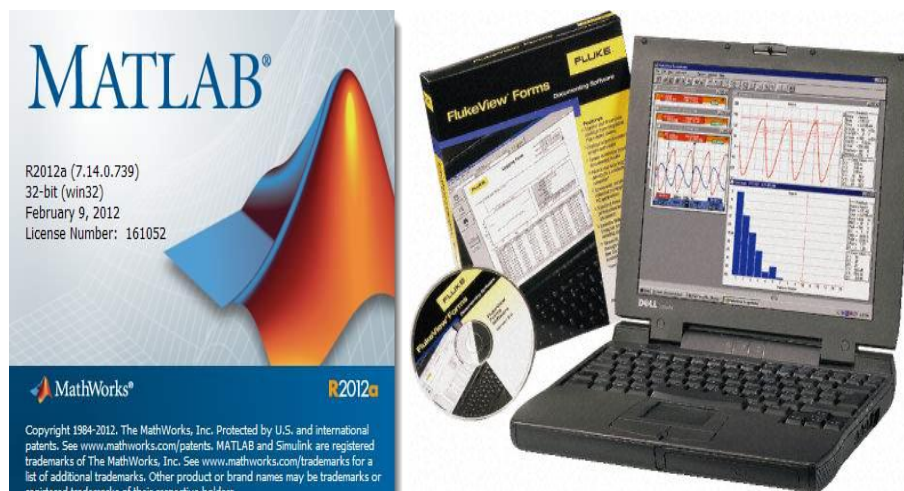


Figura 20. Entorno de los Software Utilizados.

Fuente: fluke.com/fluke/Air-Testers/Fluke975-fluke-view-forms

e.2.1.1 Procedimiento de recolección de datos

La adquisición de la base de datos se la realizó en un intervalo de 10 min, en un período de 2 meses desde el 01 de Octubre al 30 de Noviembre del 2012, dando como resultado 8 640 datos para la creación de la red tabla 6; posteriormente para la validación de la misma.

Tabla 6. – Datos totales de Temperatura, Humedad relativa y CO2

FLUKE 975			
Temperatura	% HR	CO2	Sello de tiempo
16,5	0,859	653	01/10/2012 0:09
16,3	0,861	671	01/10/2012 0:19
16,2	0,861	661	01/10/2012 0:29
16,2	0,862	676	01/10/2012 0:39
16,3	0,860	669	01/10/201 0:49
16,4	0,860	687	01/10/2012 0:59
16,4	0,858	678	01/10/2012 1:09
16,4	0,856	682	01/10/2012 1:19
16,5	0,858	676	01/10/2012 1:29
16,5	0,858	686	01/10/2012 1:39
16,5	0,859	674	01/10/2012 1:49
16,4	0,858	665	01/10/2012 1:59
16,4	0,856	682	01/10/2012 2:09
16,4	0,859	681	01/10/2012 2:19
16,5	0,857	692	01/10/2012 2:29
16,4	0,859	679	01/10/2012 2:39
16,5	0,859	681	01/10/2012 2:49

16,4	0,857	702	01/10/2012 2:59
16,4	0,858	688	01/10/2012 3:09
16,4	0,8584	695	01/10/2012 3:19
16,4	0,858	717	01/10/2012 3:29
16,4	0,8599	702	01/10/2012 3:39
16,4	0,859	688	01/10/2012 3:49
16,4	0,860	687	01/10/2012 3:59
16,4	0,858	694	01/10/2012 4:09
16,3	0,858	698	01/10/2012 4:19
16,3	0,857	699	01/10/2012 4:29
16,3	0,857	697	01/10/2012 4:39
16,2	0,857	701	01/10/2012 4:49
16,2	0,857	692	01/10/2012 4:59
16,1	0,858	688	01/10/2012 5:09
16,1	0,855	705	01/10/2012 5:19
16	0,854	701	01/10/2012 5:29
16	0,856	697	01/10/2012 5:39
16	0,855	683	01/10/2012 5:49
16	0,855	688	01/10/2012 5:59
16	0,854	676	01/10/2012 6:09
16	0,854	670	01/10/2012 6:19

16,1	0,855	691	01/10/2012 6:29
16,2	0,853	661	01/10/2012 6:39
16,2	0,854	668	01/10/2012 6:49
16,3	0,853	653	01/10/2012 6:59
16,3	0,855	639	01/10/2012 7:09
16,5	0,851	610	01/10/2012 7:19
16,7	0,853	621	01/10/2012 7:29
16,8	0,849	603	01/10/2012 7:39
17	0,850	598	01/10/2012 7:49
17	0,854	622	01/10/2012 7:59
17,1	0,855	608	01/10/2012 8:09
17,2	0,851	621	01/10/2012 8:19
17,6	0,841	581	01/10/2012 8:29
17,8	0,839	566	01/10/2012 8:39

Fuente: Elaboración los autores.

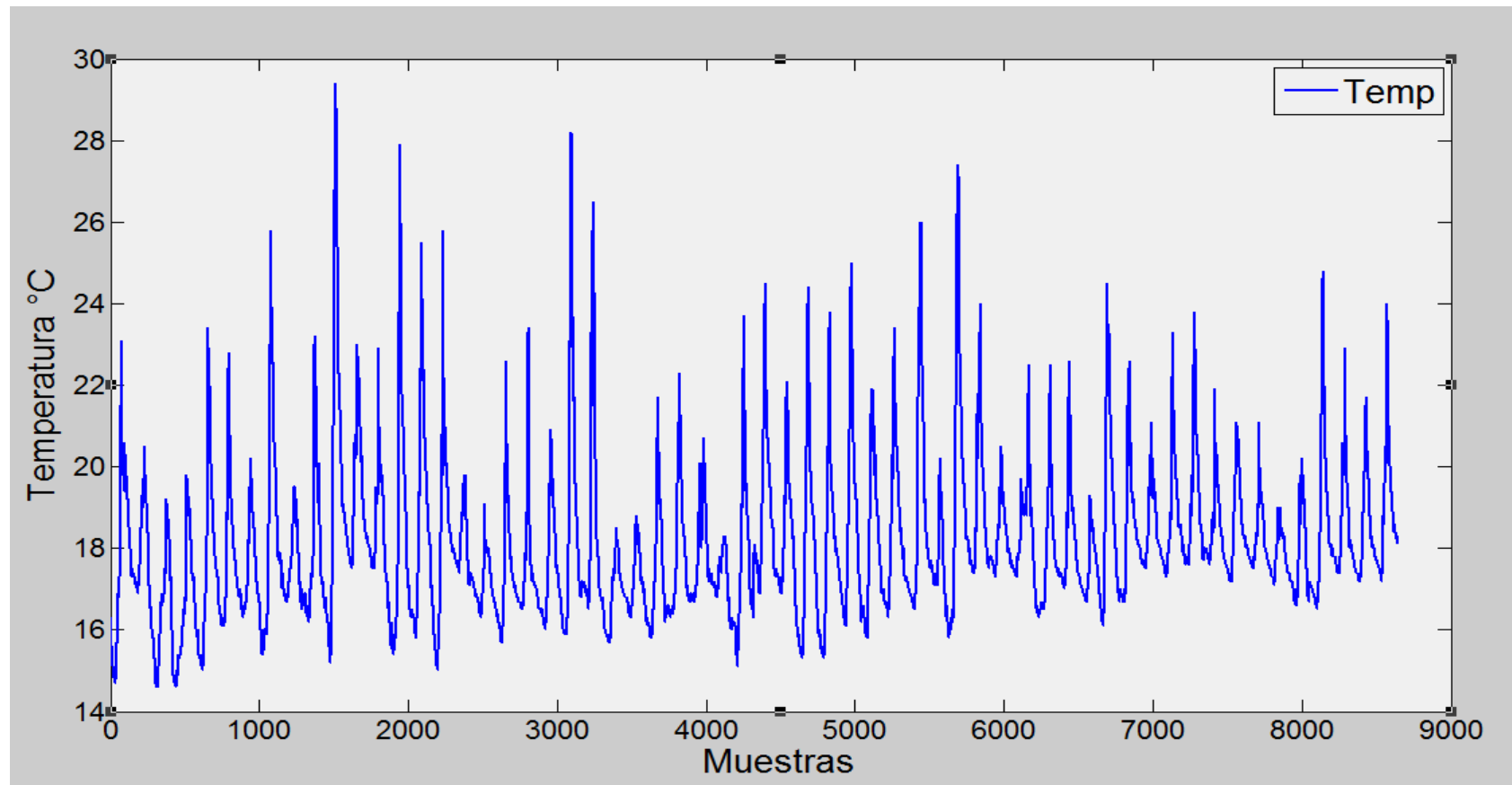


Figura 21. Temperatura vs Muestras (8 640 datos totales), Dónde: 1 muestra = 1dato = 10 minutos

Fuente: Elaboración los autores.

e.2.2 Procedimiento para autoajustar datos

Para alcanzar un alto rendimiento de la red (RNA) procedemos a autoajustar los datos que hacen que la curva se desestabilice figura 22; para ello utilizamos la técnica de la depuración denominada Ajuste Data (Anexo 2). De esta manera alcanzamos un alto grado de firmeza para la creación de la RED NEURONAL.

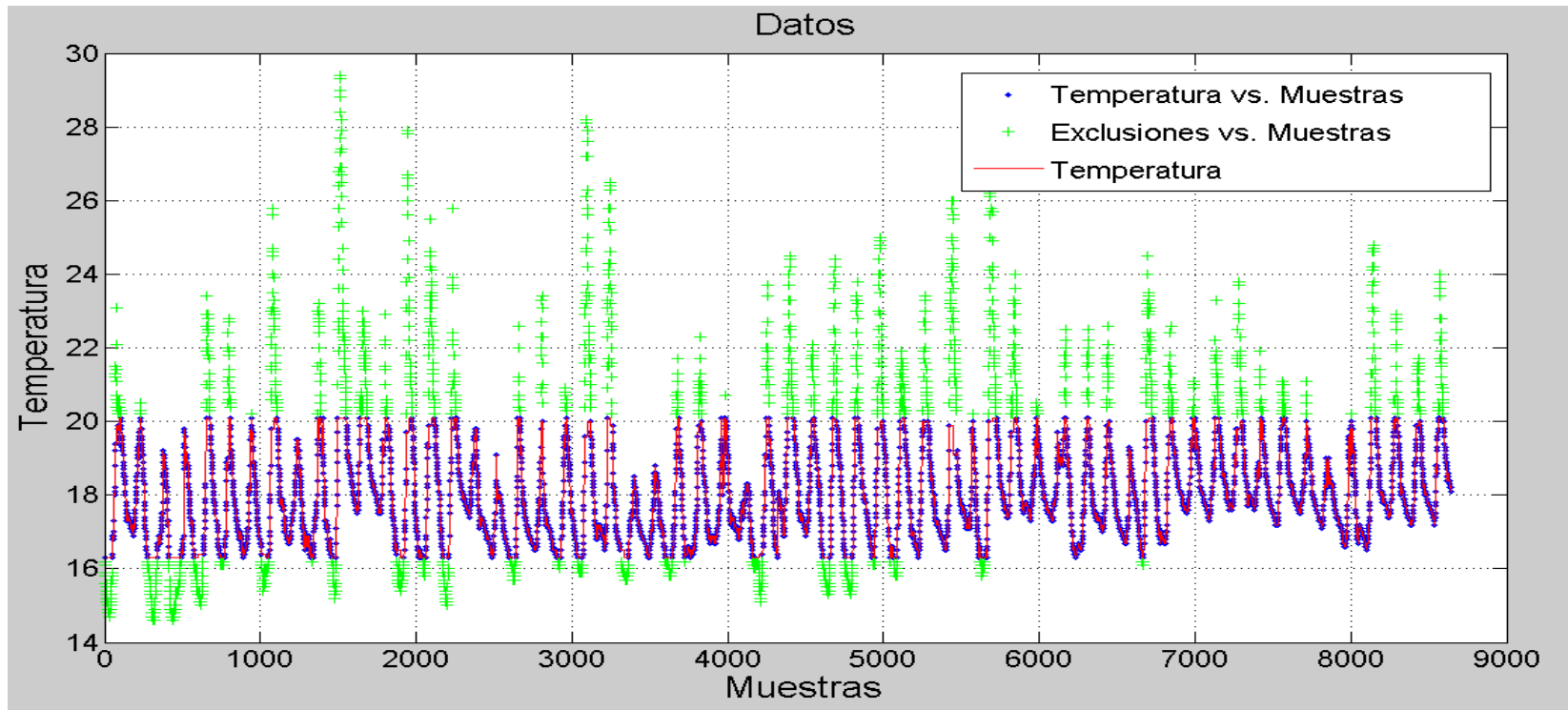


Figura 22. Temperatura vs Muestras (8 640 datos ajustados)

Fuente: Elaboración los autores.

e.2.3 Procedimiento para la crear la Red Neuronal mediante el Toolbox NTSTOOL.

Mediante el Toolbox NTSTOOL se procede a la creación de la red neuronal, para lo cual se presentan algunas ventanas; cuyo proceso se lo presenta a continuación;

e.2.3.1 Selección de la Red.

Dentro de este parámetro vamos resolver un problema de series de tiempo no lineal con una red neuronal dinámica, en el que se utilizan los valores pasados de una o más series de tiempo para predecir valores futuros.

Esta herramienta le permite resolver tres tipos de problemas no lineales de series de tiempo; en este caso vamos a Predecir series $y(t)$ dado valores pasados d de $y(t)$, como lo muestra la figura 23.

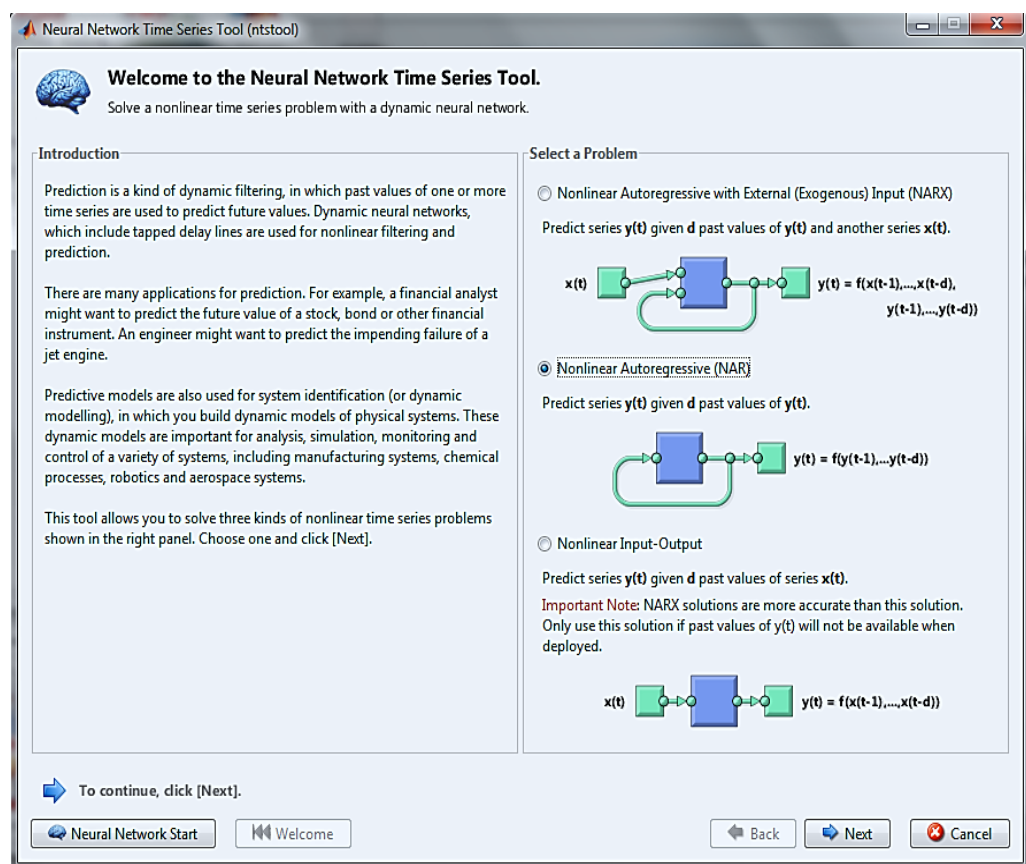


Figura 23. Selección de la Red no lineal Autorregresiva (RNA).

Fuente: Elaboración los autores.

e.2.3.2 Toma de datos.

En este proceso se ingresa los datos ajustados para la creación de la red y se lo realiza en forma de una matriz en fila, figura 24.

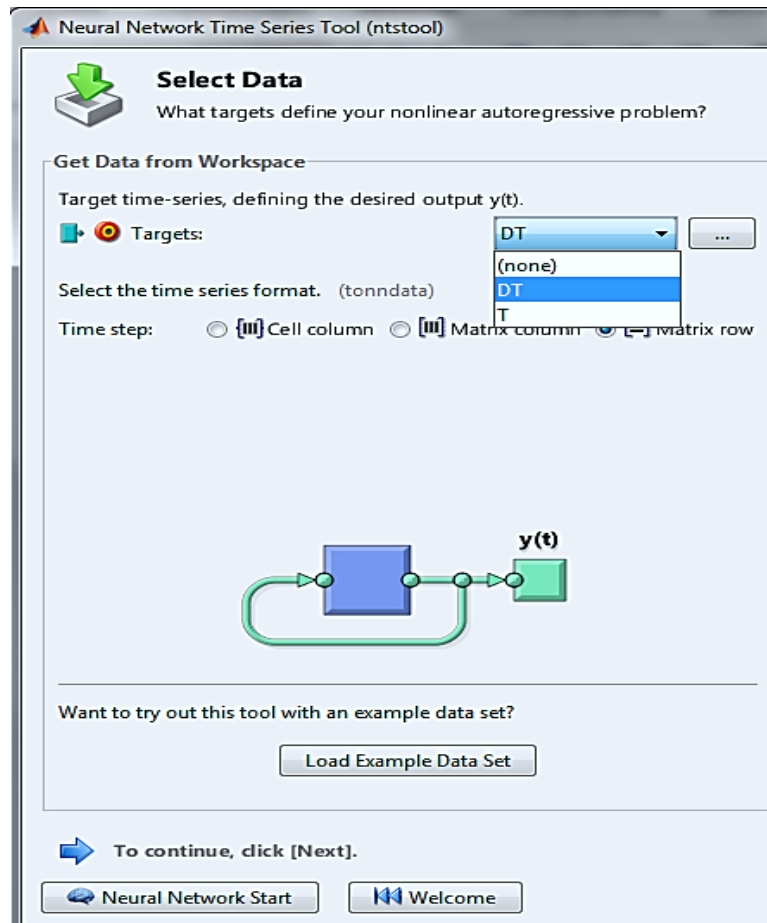


Figura 24. Ingreso de los datos Ajustados.

Fuente: Elaboración los autores.

e.2.3.3 Selección para su validación.

Otro paso importante para obtener la RNA es seleccionar los parámetros de Entrenamiento, Validación y Prueba; figura 25.

El primero se presenta a la red durante el entrenamiento, y la red se ajusta de acuerdo a su error asignándole un 70 % de los datos totales.

El segundo se utiliza para medir la generalización de la red, y para detener la formación, cuando se detiene la mejora de generalización; este tiene un 15 %. Y,

El tercero que aunque no tienen ningún efecto en la formación pero igual proporciona una medida independiente del rendimiento de la red durante y después de la formación; cuenta con un 15 %.

Todos los porcentajes antes mencionados son valores predeterminados y los más recomendables al utilizar este tipo de redes.

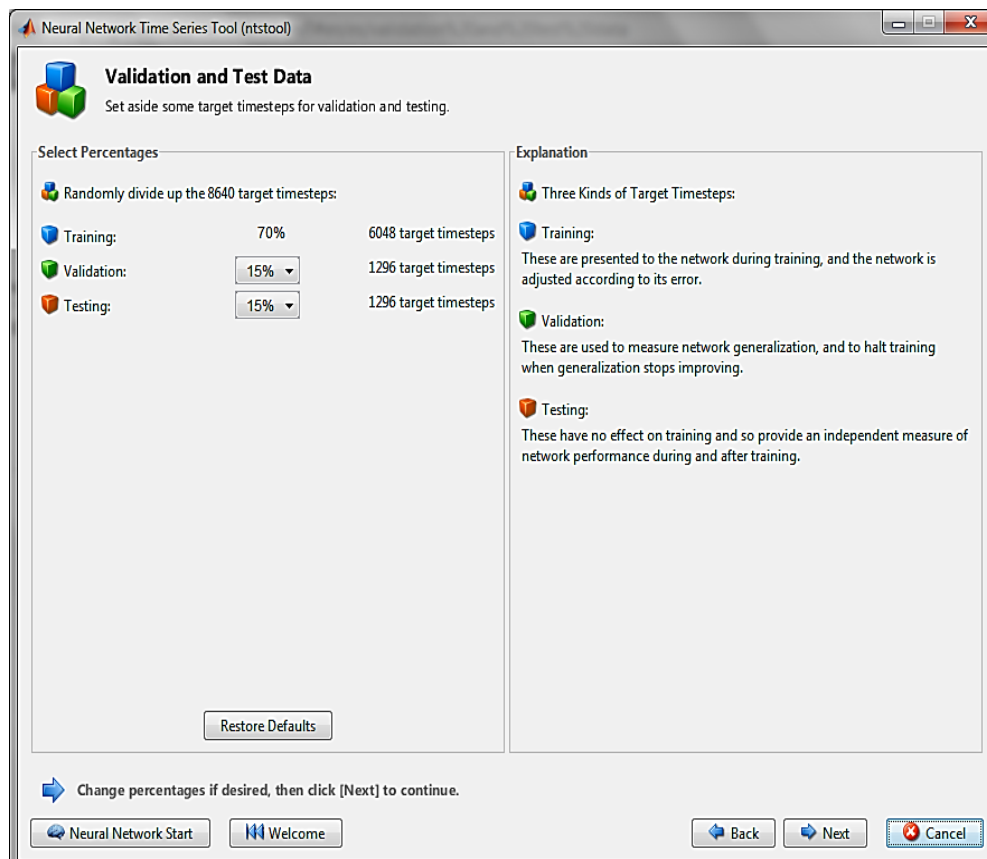


Figura 25. Selección de los datos de validación y prueba.

Fuente: Elaboración los autores.

e.2.3.4 Selección de la Arquitectura de la Red.

De esta manera se puede decir que tenemos en su mayoría la RNA, puesto que es la parte más relevante, porque aquí se le designa a la RNA el número de neuronas y el número de **retrasos** (delays); por consiguiente se debe aplicar muchas veces el cambiar estos dos parámetros hasta encontrar la RNA adecuada para nuestro interés.

Y así como muestra la figura 26 y luego de realizar muchas pruebas se pudo determinar que para nuestra RNA se necesita un número de neuronas y delays que va entre 13 y 23 respectivamente.

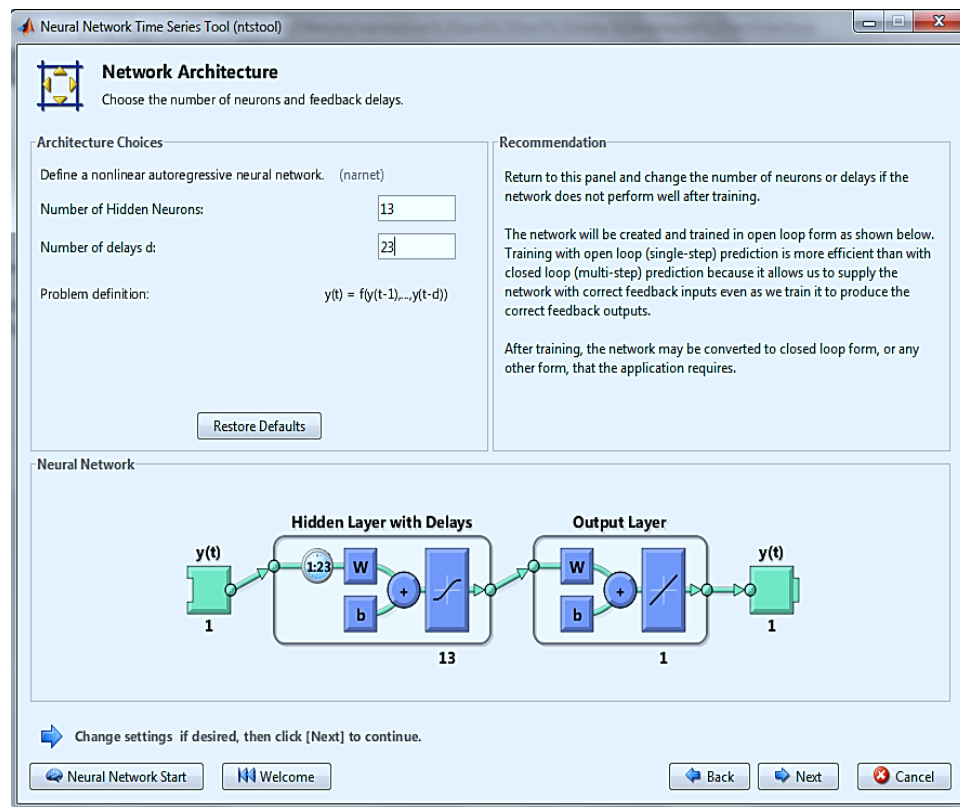


Figura 26. Arquitectura de la red.

Fuente: Elaboración los autores.

e.2.3.5 Entrenamiento De La Red

Por medio del entrenamiento se concluye con la creación de la RED NEURONAL figura 27, la cual hay que generar diferentes muestras para determinar la que cuente con su mayor rendimiento.

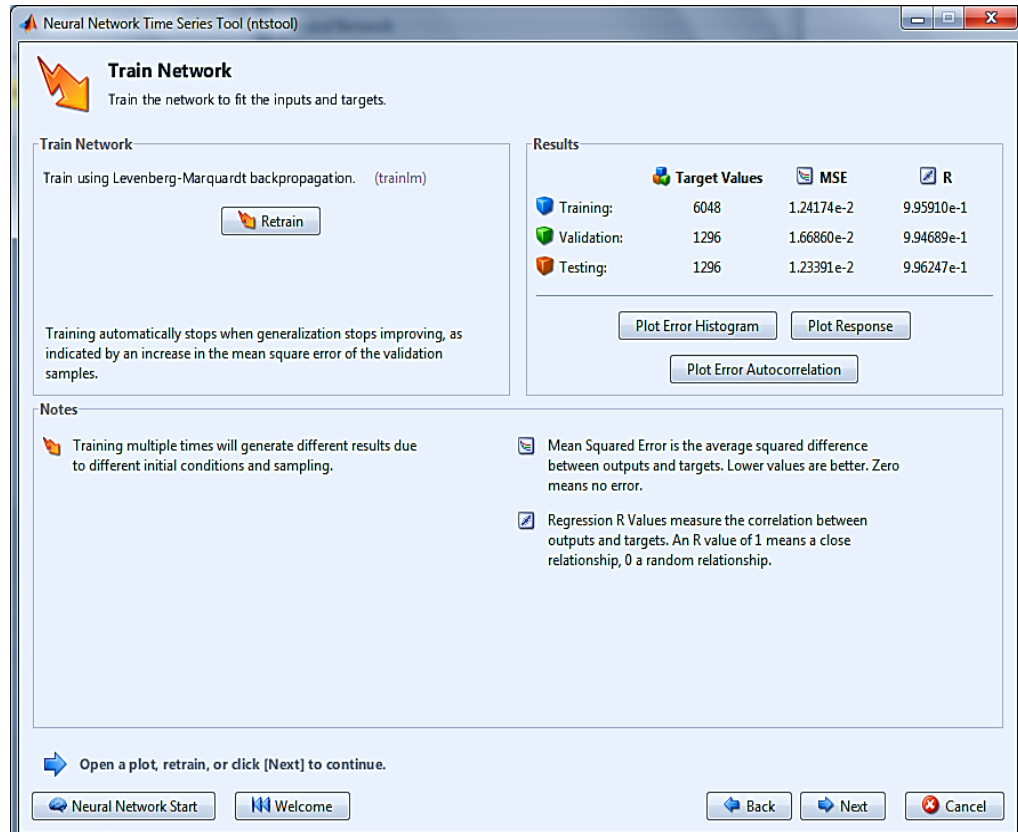


Figura 27. Entrenar la red.

Fuente: Elaboración los autores.

e.2.4. Procedimiento para predicción y validación de datos.

Para la predicción escogemos 2 días seguidos, repartidos de la siguiente manera:

Del primer día.

- Tomamos los últimos 121 datos de ese periodo (Predicción)

Del segundo día.

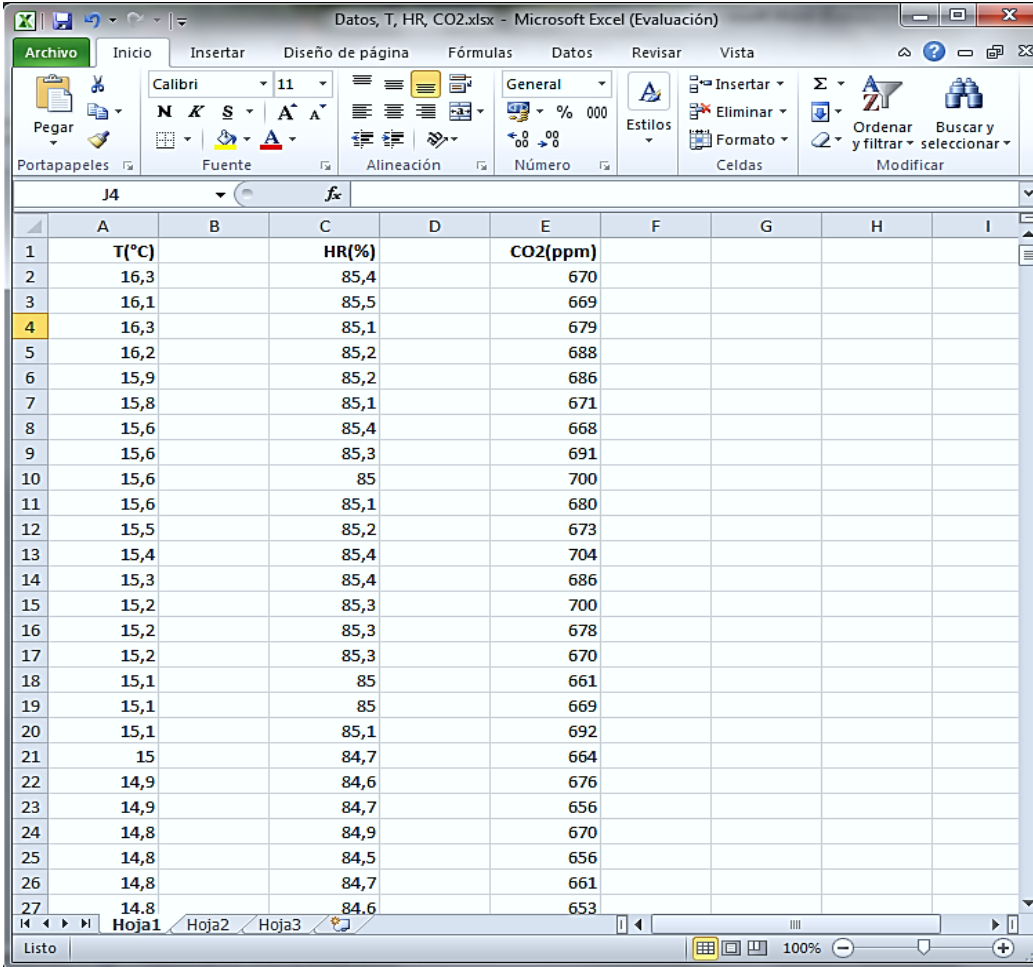
- Tomamos los primeros 23 datos de ese periodo (Predicción) y,
- Tomamos los últimos 121 datos de ese periodo (Validación).

Los datos son tomados mediante el número de delays. Anexo 6, 7 y 8.

f. RESULTADOS

f.1 Procesamiento de datos

Luego de haber realizado el respectivo trabajo de recolección de datos con el Fluke 975 y mediante el Software Fluke View Forms® tenemos como resultado un archivo en formato .XLS, óptimo para trabajar en el programa MATLAB para la creación de la red neuronal.



	A	B	C	D	E	F	G	H	I
1	T(°C)		HR(%)		CO2(ppm)				
2	16,3		85,4		670				
3	16,1		85,5		669				
4	16,3		85,1		679				
5	16,2		85,2		688				
6	15,9		85,2		686				
7	15,8		85,1		671				
8	15,6		85,4		668				
9	15,6		85,3		691				
10	15,6		85		700				
11	15,6		85,1		680				
12	15,5		85,2		673				
13	15,4		85,4		704				
14	15,3		85,4		686				
15	15,2		85,3		700				
16	15,2		85,3		678				
17	15,2		85,3		670				
18	15,1		85		661				
19	15,1		85		669				
20	15,1		85,1		692				
21	15		84,7		664				
22	14,9		84,6		676				
23	14,9		84,7		656				
24	14,8		84,9		670				
25	14,8		84,5		656				
26	14,8		84,7		661				
27	14,8		84,6		653				

Figura 28. Variables almacenadas en formato .XLS.

Fuente: Elaboración los autores.

f.1.1 Importar datos a Matlab

Ya con los datos listos para trabajar como archivo .XLS, se procede a cargar en el programa MATLAB, buscando la ubicación de la carpeta que los contiene tal como se muestra en la figura 29.

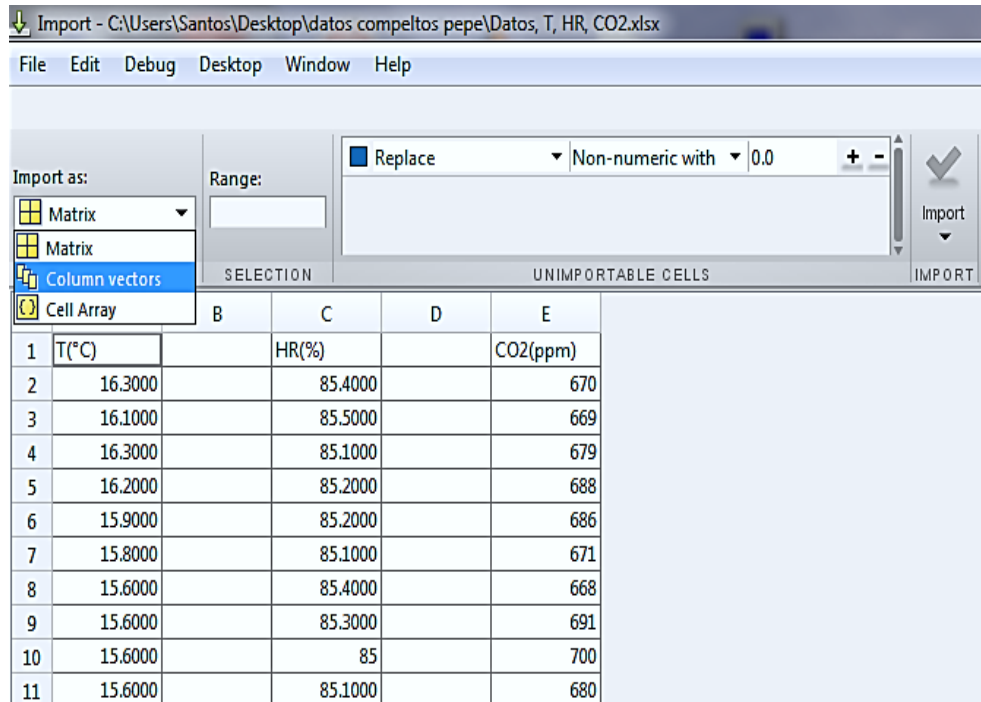


Figura 29. Datos cargados al programa MATLAB

Fuente: Elaboración los autores.

Luego se procede a la importación de los mismos en forma de columna de vectores; empezamos con la variable de Temperatura la que se cargara en el workspace para la creación de la primera red neuronal figura 30.

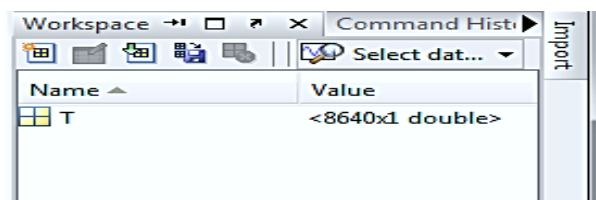


Figura 30. Datos cargados al Workspace

Fuente: Elaboración los autores.

f.2. Resultados de los modelos generados al entrenar la red neuronal y predicción.

La forma estructural de las variables de Temperatura, Humedad Relativa y CO2 son las mismas para el proceso de entrenamiento y predicción figura 31.

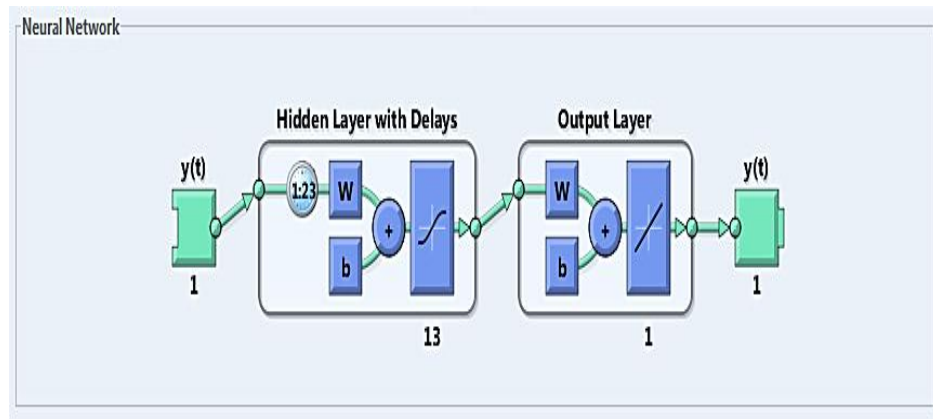


Figura 31. Forma Estructural de la Red.

Fuente: Elaboración los autores.

Así mismo todos los modelos generados con la GUI ntstool tienen las siguientes características:

Tipo de Red Neuronal: No linear Autorregresiva (RNA)

Numero de neuronas ocultas: 13

Numero de delays: 23

Función de Entrenamiento: TRAINLM

Función de División de Datos: DIVIDERAND

Función de Rendimiento: MSE

Función de Derivación: DEFAULTDERIV

f.2.1 Resultados de la Red Neuronal con datos de Temperatura

De acuerdo con los respectivos entrenamientos de la RNA con datos de Temperatura que se ha realizado tenemos:

MSE: 0,011

R: 0,996

Dónde: **MSE** es el Error Cuadrático Medio, y **R** es la Regresión.

A continuación las gráficas de Validación, Regresión y Error.

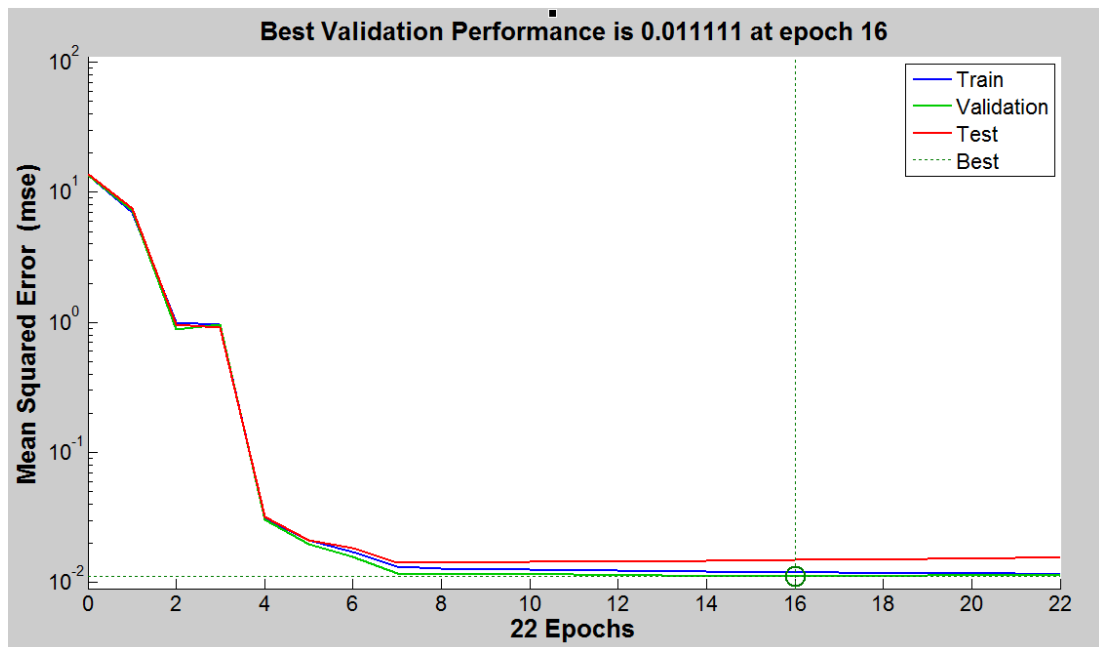


Figura 32. Rendimiento de validación de la red.

Fuente: Elaboración los autores.

Como se puede observar en la figura 32, el Rendimiento de Validación de la Red se encuentra ubicado en 0,011 en la etapa 16; siendo este donde mejor funcionamiento tiene la Red Neuronal.

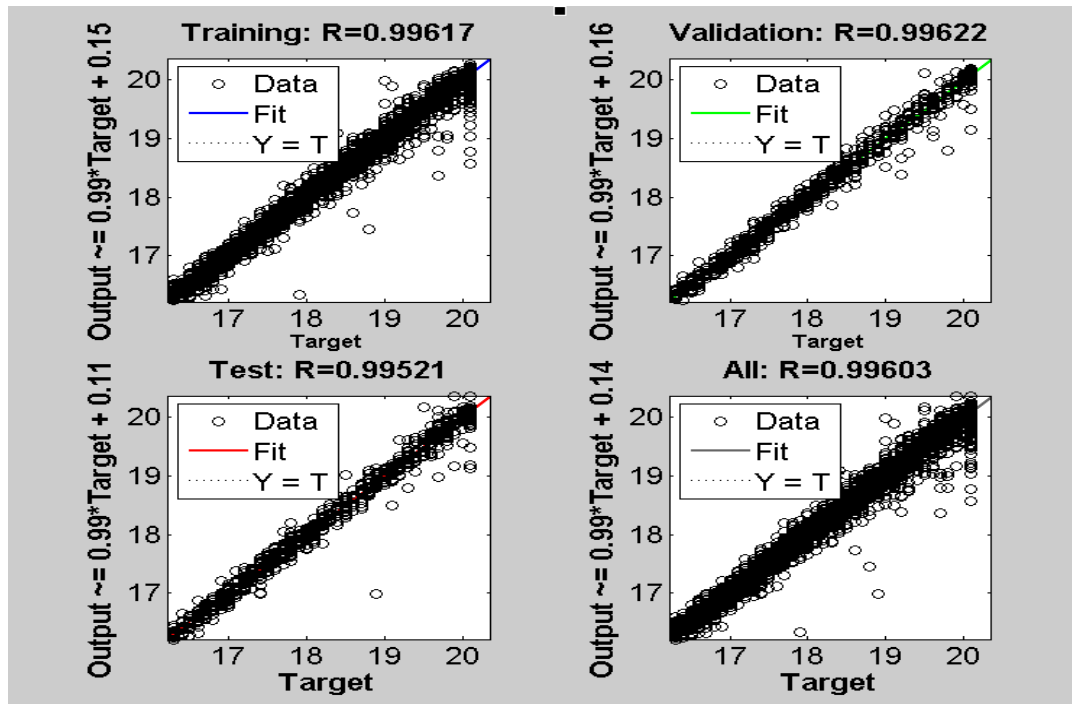


Figura 33. Diagrama de Regresión.

Fuente: Elaboración los autores.

Como podemos apreciar en la figura 33, del Diagrama de Regresión, el mejor resultado que beneficia a la Red Neuronal en la variable de Temperatura es el valor de rendimiento de 99,603 % ya que con esto garantizamos la óptima predicción de la RNA.

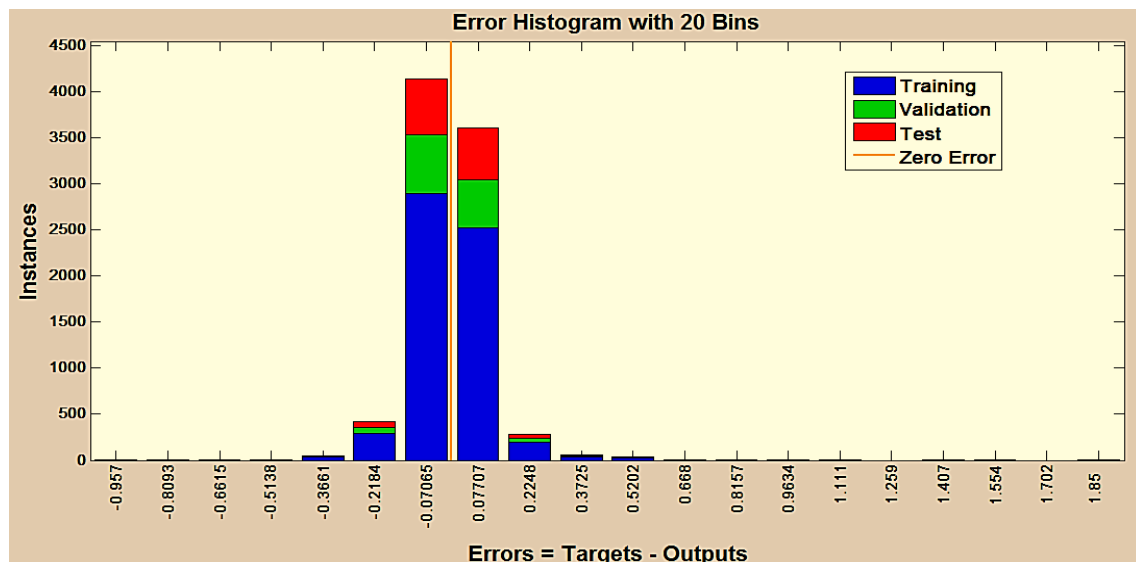


Figura 34. Error del histograma de la Red.

Fuente: Elaboración los autores.

El histograma de error de la Red de la figura 34, muestra que la media del error alrededor de cero es de más de 4 000 instancias, con valores secundarios más representativos de menos de 500 instancias.

f.2.1.1 Resultados de predicción con valores de Temperatura.

Según el proceso de predicción tenemos que los valores de las salidas (outputs) corresponden a los valores de validación (Dval) con un error mínimo, figura 35.

Y con ello el histograma de Error, figura 36.

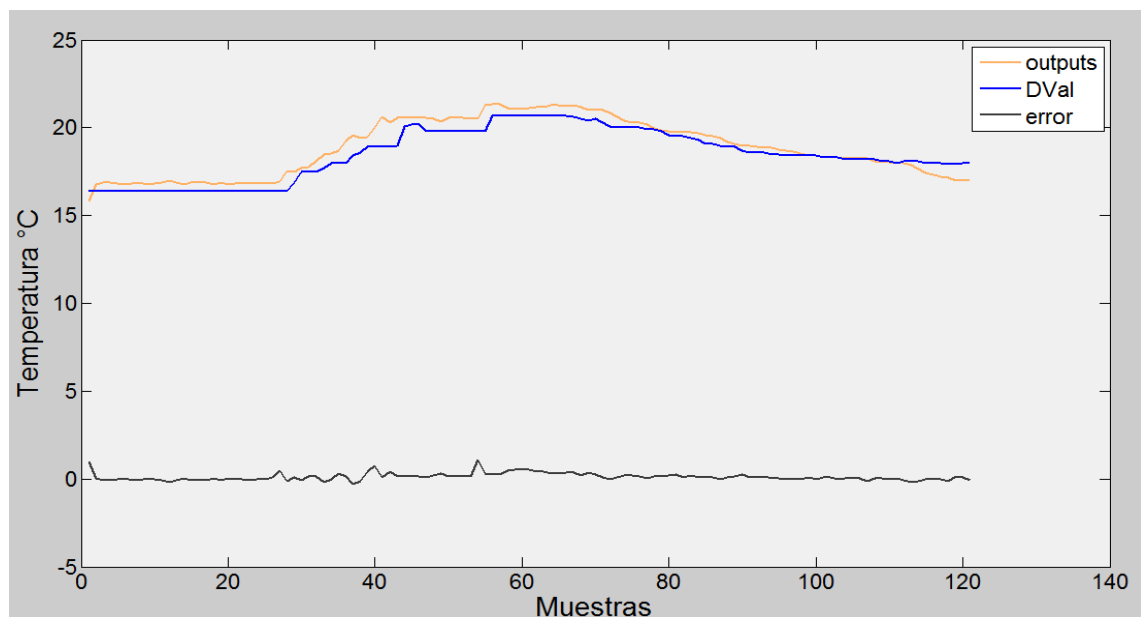


Figura 35. Curva de Validación, Predicción y Error

Fuente: Elaboración los autores.

Como se puede estimar en la figura 35, la curva de Validación se asemeja mucho a la curva de Predicción dando un margen de error casi despreciable, por lo cual se puede decir que este proceso cumple con los requerimientos planteados.

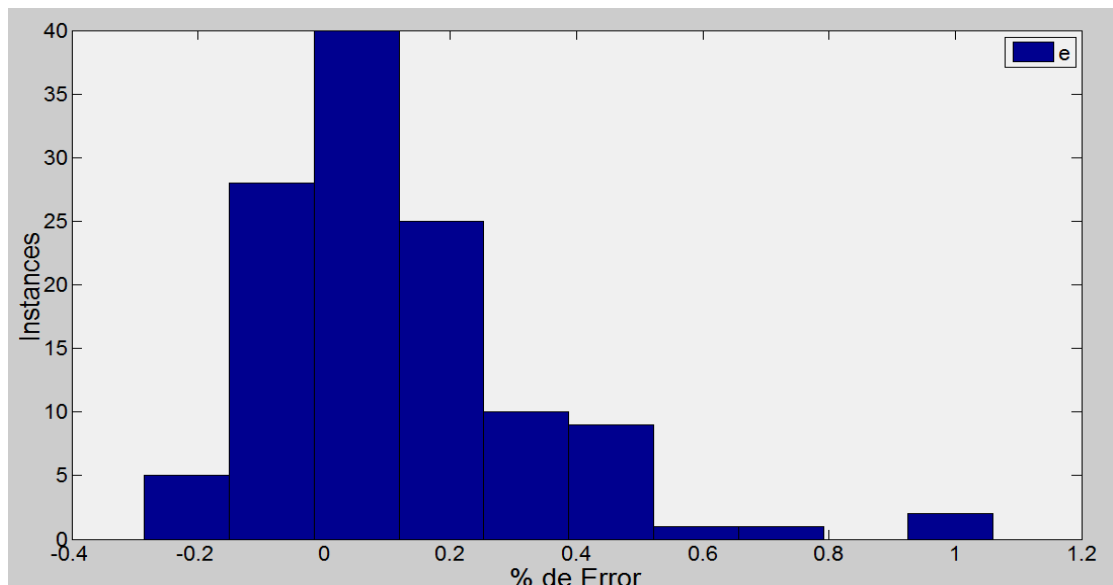


Figura 36. Histograma de Error de Predicción.

Fuente: Elaboración los autores.

El histograma de error de predicción de la figura 36. Muestra que la media del error alrededor de cero es de más de 40 instancias, con valores secundarios más representativos de menos de 25 instancias.

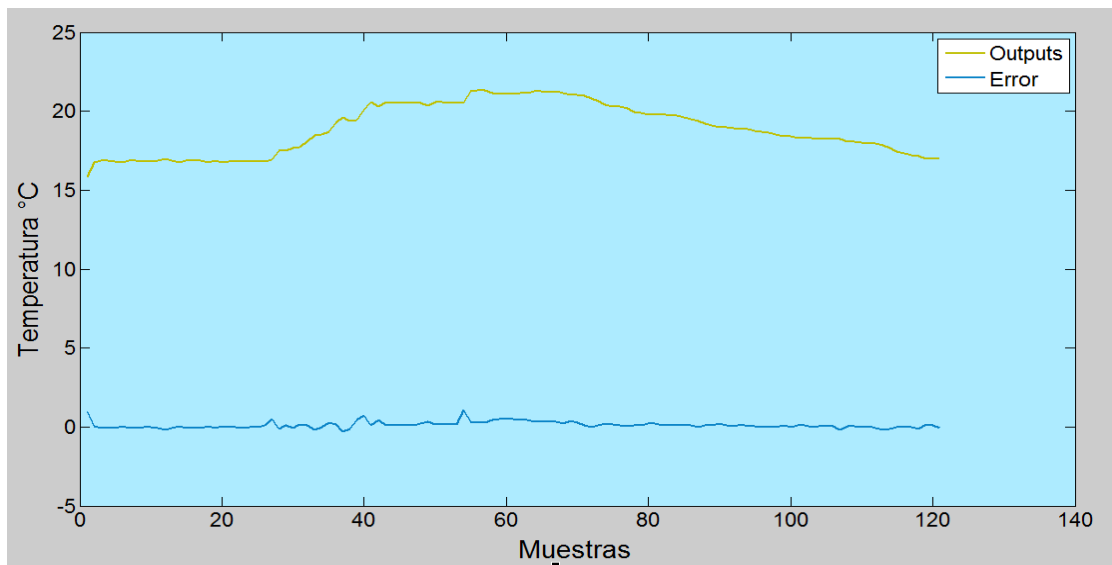


Figura 37. Curva de Predicción.

Fuente: Elaboración los autores.

Con el respectivo proceso y tal como muestra la figura 37. Sobre la Curva de Predicción, tenemos el resultado de un día de valores de Temperatura.

Los valores en este día corresponden a una temperatura que está dentro de los 15 °C a los 23 °C, los cuales se tomaron los datos del día 23 de Octubre del 2 012 para la evaluación correspondiente que a su vez nos dio como predicción los datos del día 24 de Octubre del 2 012.

f.2.2 Resultados de la red con datos de Humedad Relativa.

Con los respectivos entrenamientos de la NAR con datos de Humedad Relativa que se ha realizado tenemos:

MSE: 0,196

R: 0,967

Dónde: **MSE** es el Error Cuadrático Medio, y **R** es la Regresión.

A continuación las gráficas de Validación, Regresión y Error.

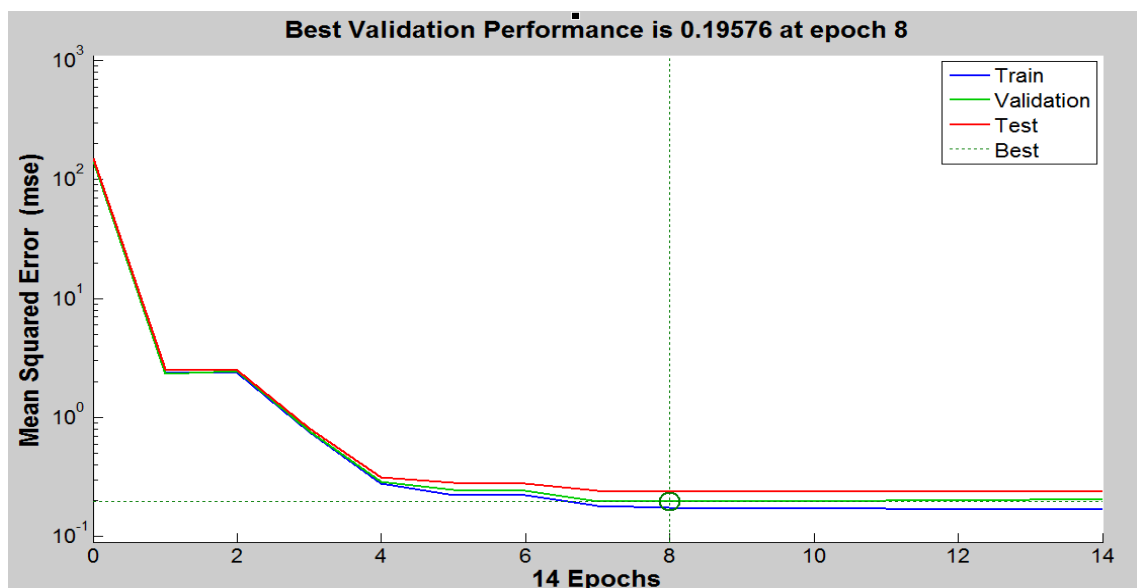


Figura 38. Rendimiento de validación de la red.

Fuente: Elaboración los autores.

Como se puede observar en la figura 38, el Rendimiento de Validación de la Red se encuentra ubicado en 0,196 en la etapa 8; siendo este donde mejor funcionamiento tiene la Red Neuronal.

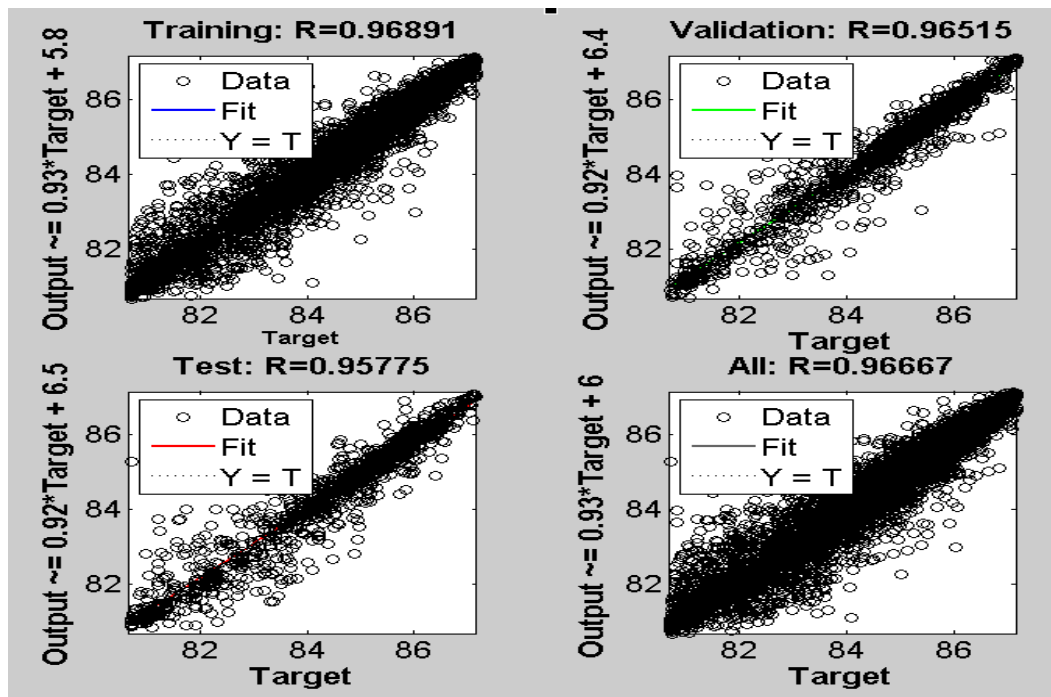


Figura 39. Diagrama de Regresión.

Fuente: Elaboración los autores.

Como podemos apreciar en la figura 39, del Diagrama de Regresión, el mejor resultado que favorece a la Red Neuronal en la variable de Humedad Relativa es el valor de rendimiento de 96,667 %, puesto que así se garantiza una adecuada predicción de la RNA; y que además está dentro de nuestros valores de rendimiento a aplicar el cual está por encima del 90 %.

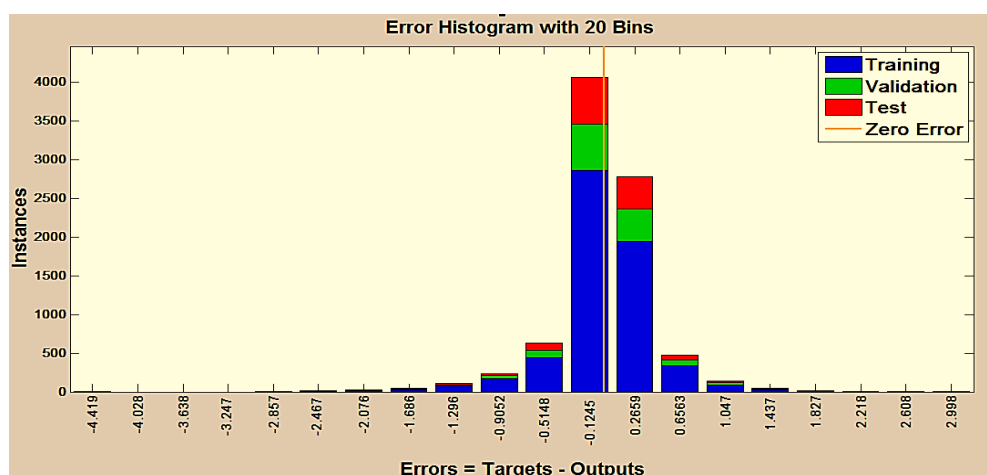


Figura 40. Error del histograma de la Red.

Fuente: Elaboración los autores.

El histograma de error de la Red de la figura 40. Muestra que la media del error alrededor de cero es de más de 4 000 instancias, con valores secundarios más representativos de menos de 800 instancias.

f.2.2.1 Resultados de predicción con valores de Humedad Relativa.

Según el proceso de predicción tenemos que los valores de las salidas (outputs) corresponden a los valores de validación (Dval) con un error mínimo, figura 41.

Y con ello el histograma de Error, figura 42.

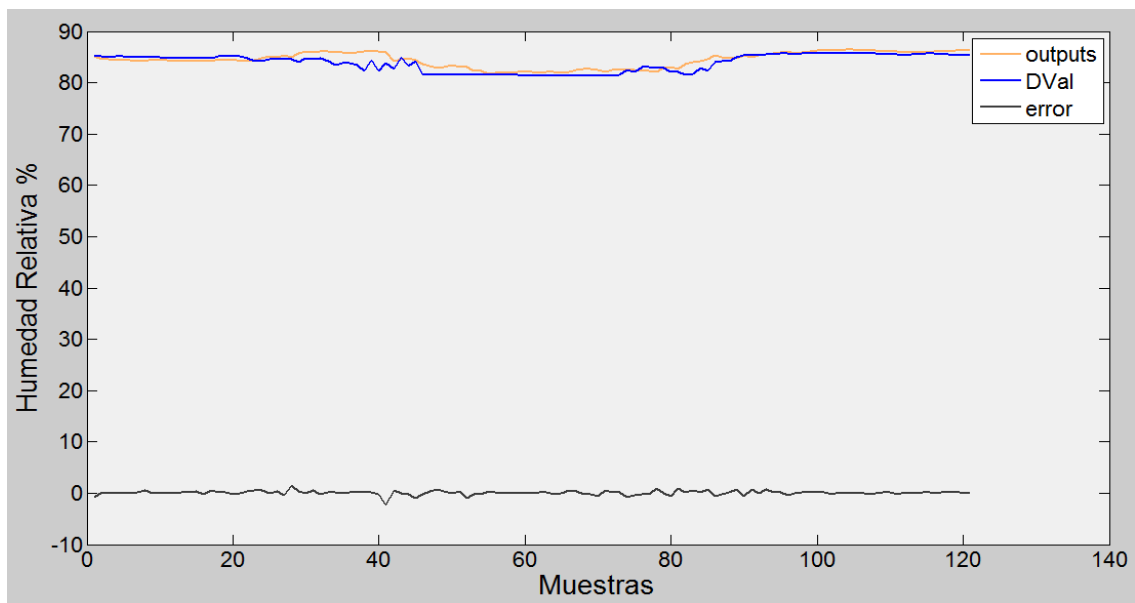


Figura 41. Curva de Validación, Predicción y Error

Fuente: Elaboración los autores.

Como se puede apreciar en la figura 41 la curva de Validación tiene bastante similitud a la curva de Predicción dando un margen pequeño de error, es por eso que se puede decir que este proceso está dentro de los requerimientos trazados.

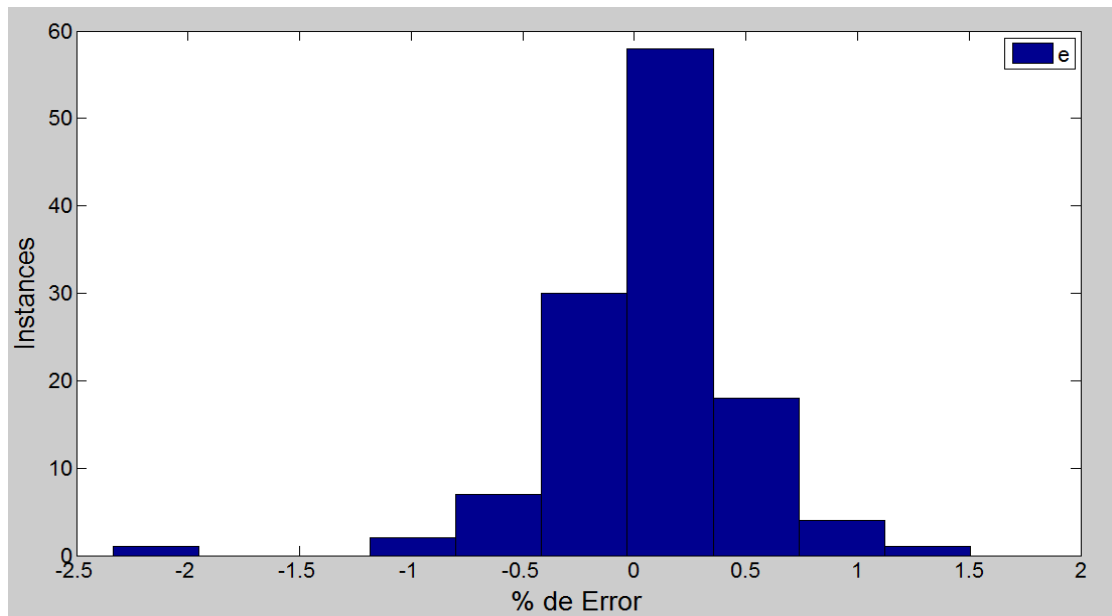


Figura 42. Histograma de Error de Predicción.

Fuente: Elaboración los autores.

El histograma de error de predicción de la figura 42, nos da a conocer que la media del error alrededor de cero es de más de 50 instancias, con valores secundarios más representativos de menos de 30 instancias.

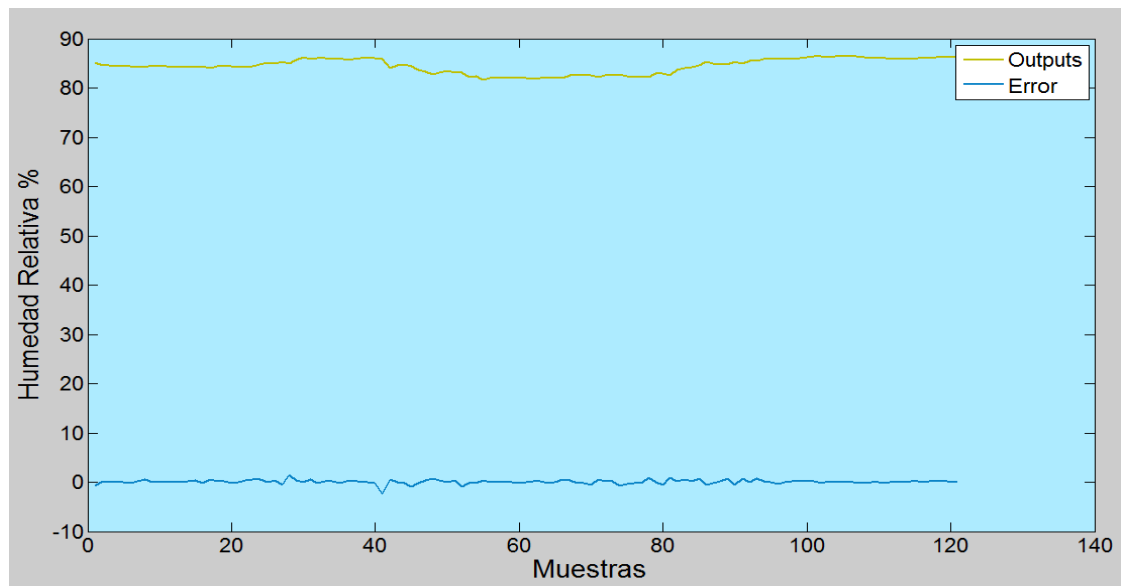


Figura 43. Curva de Predicción.

Fuente: Elaboración los autores.

Al realizar el proceso correspondiente y tal como se observa en la figura 43, sobre la curva de predicción, tenemos el resultado de una muestra de un día de valores de Humedad Relativa. Que corresponden a la toma de datos del día 23 de Octubre del 2 012 para la evaluación correspondiente que a su vez nos dio como predicción los datos del día 24 de Octubre del 2 012.

f.2.3 Resultados de la red con datos de CO2

Con los entrenamientos de la NAR con datos de CO2 que se ha realizado tenemos:

MSE: 146,77

R: 0,983

Dónde: **MSE** es el Error Cuadrático Medio, y **R** es la Regresión.

A continuación las gráficas de Validación, Regresión y Error.

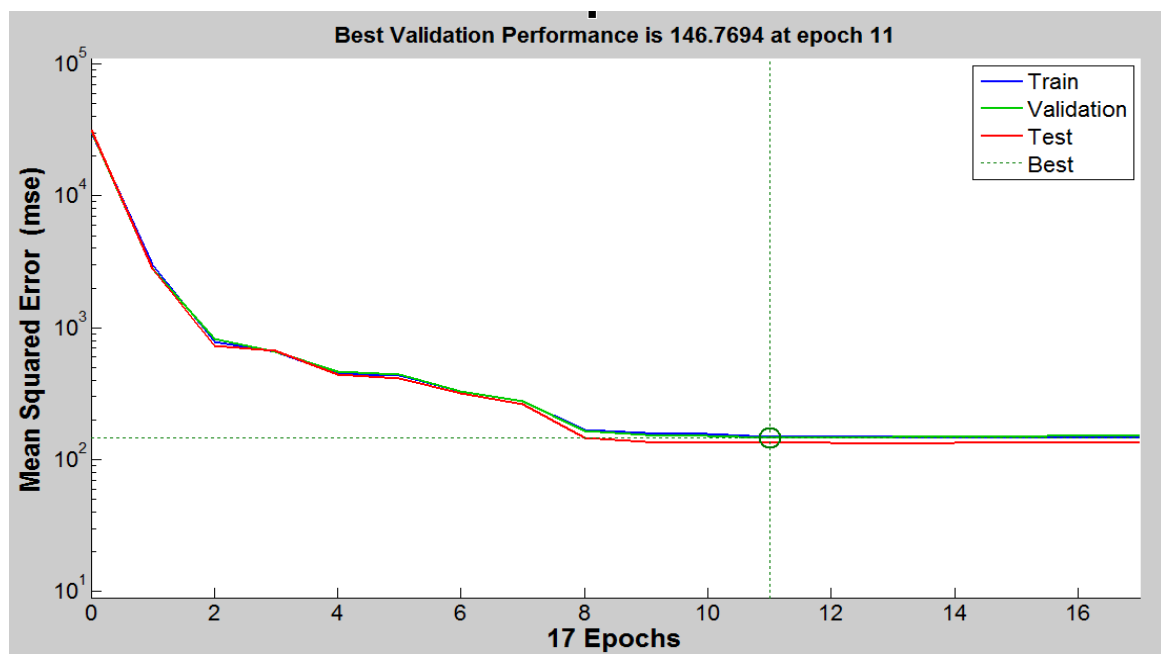


Figura 44. Rendimiento de validación de la red.

Fuente: Elaboración los autores.

Como se puede observar en la figura 44, el Rendimiento de Validación de la Red se encuentra ubicado en 146,769 en la etapa 11; siendo este donde mejor funcionamiento tiene la Red Neuronal.

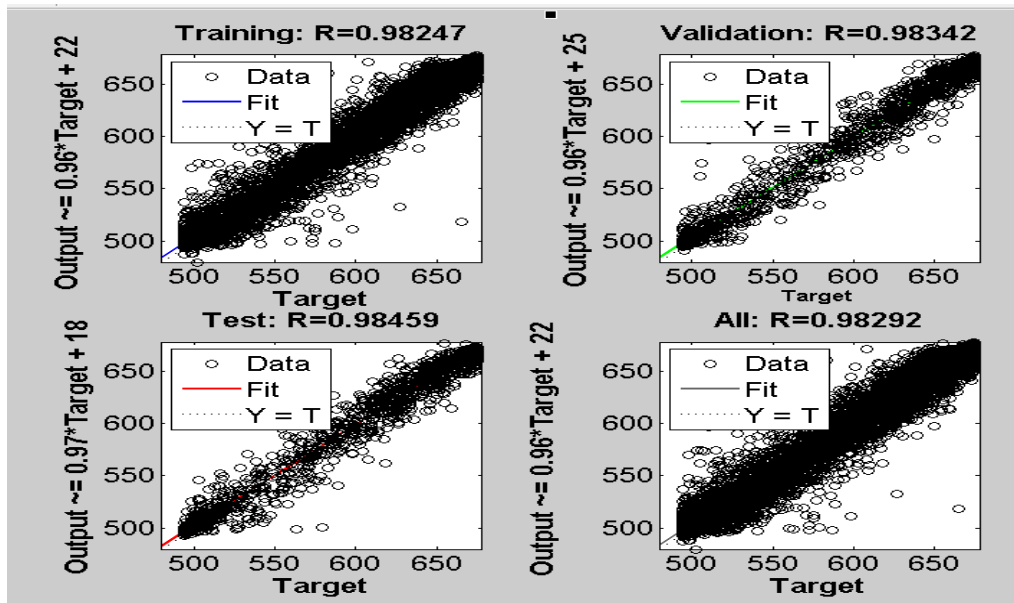


Figura 45. Diagrama de Regresión.

Fuente: Elaboración los autores.

Como podemos apreciar en la figura 45, del Diagrama de Regresión, el mejor resultado que favorece a la Red Neuronal en la variable de CO2 es el valor de rendimiento del 98,292 % con esto garantizamos la óptima predicción de la RNA; y al igual que las variables de Temperatura y Humedad Relativa están dentro de los requerimientos los cuales son que este superior al valor de un 90 %.

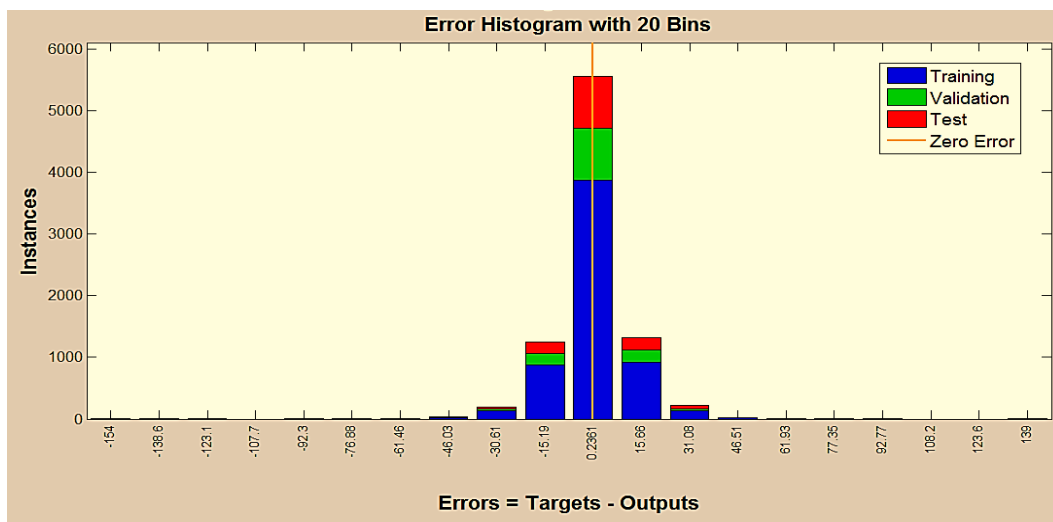


Figura 46. Error del histograma de la Red.

Fuente: Elaboración los autores.

El histograma de error de la Red de la figura 46. Muestra que la media del error alrededor de cero es de más de 5 500 instancias, con valores secundarios más representativos de menos de 1 500 instancias.

f.2.3.1 Resultados de predicción con valores de CO2.

Según el proceso de predicción tenemos que los valores de las salidas (outputs) corresponden a los valores de validación (Dval) con un error mínimo, figura 47.

Y con ello el histograma de Error, figura 48.

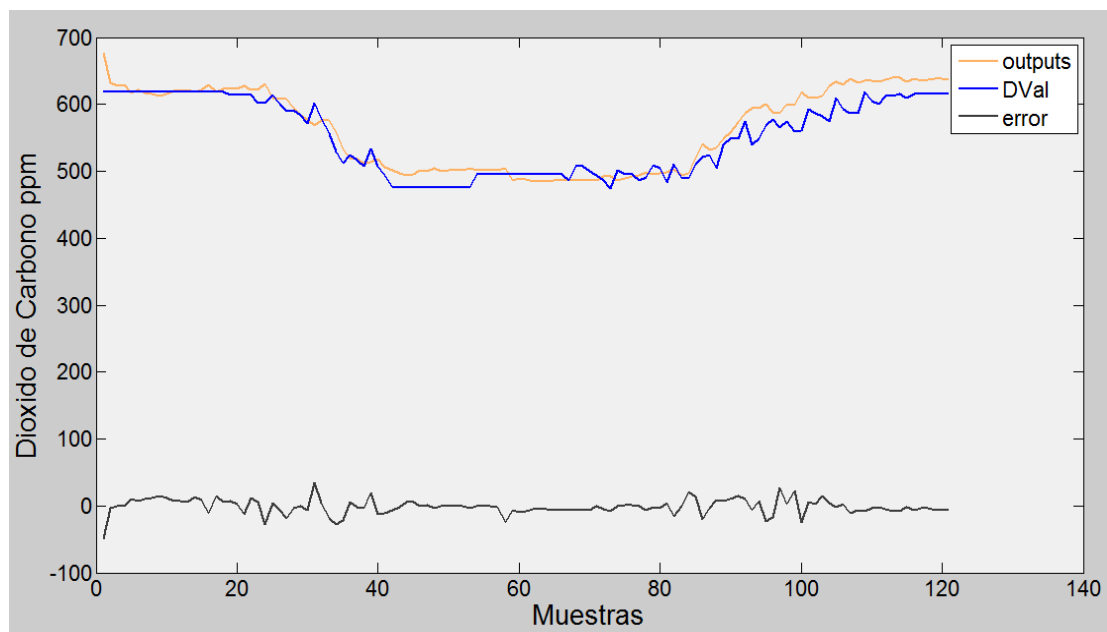


Figura 47. Curva de Validación, Predicción y Error

Fuente: Elaboración los autores.

Como se puede apreciar en la figura 47 la curva de Validación es bastante similar a la curva de Predicción dando un margen pequeño de error, es por eso que se puede decir que este proceso está dentro de los requerimientos planteados.

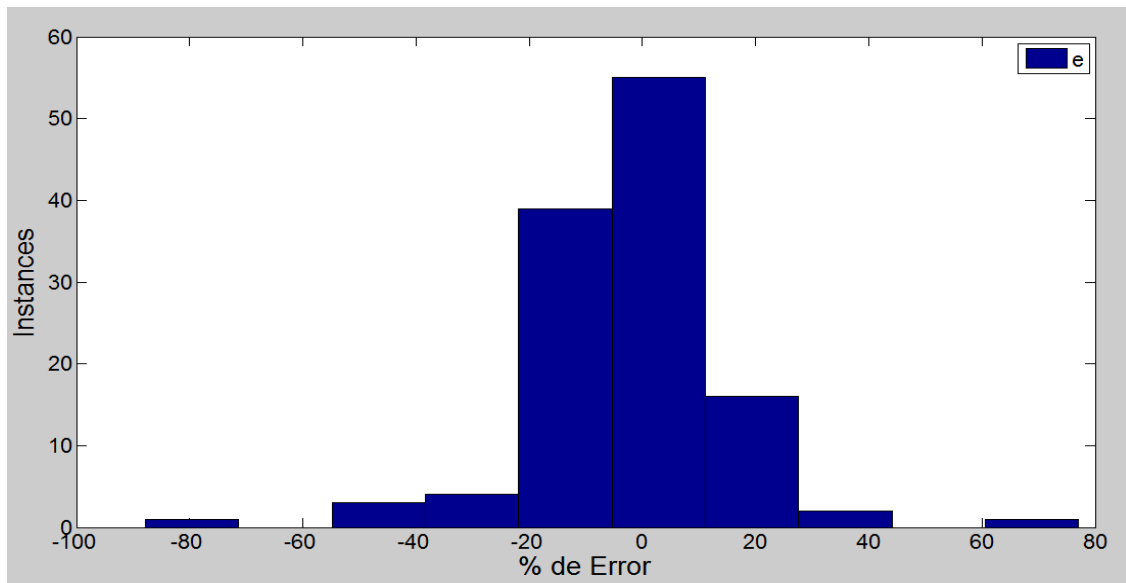


Figura 48. Histograma de Error de Predicción.

Fuente: Elaboración los autores.

El histograma de error de predicción de la figura 48, nos da a conocer que la media del error alrededor de cero es de más de 55 instancias, con valores secundarios más representativos de menos de 40 instancias.

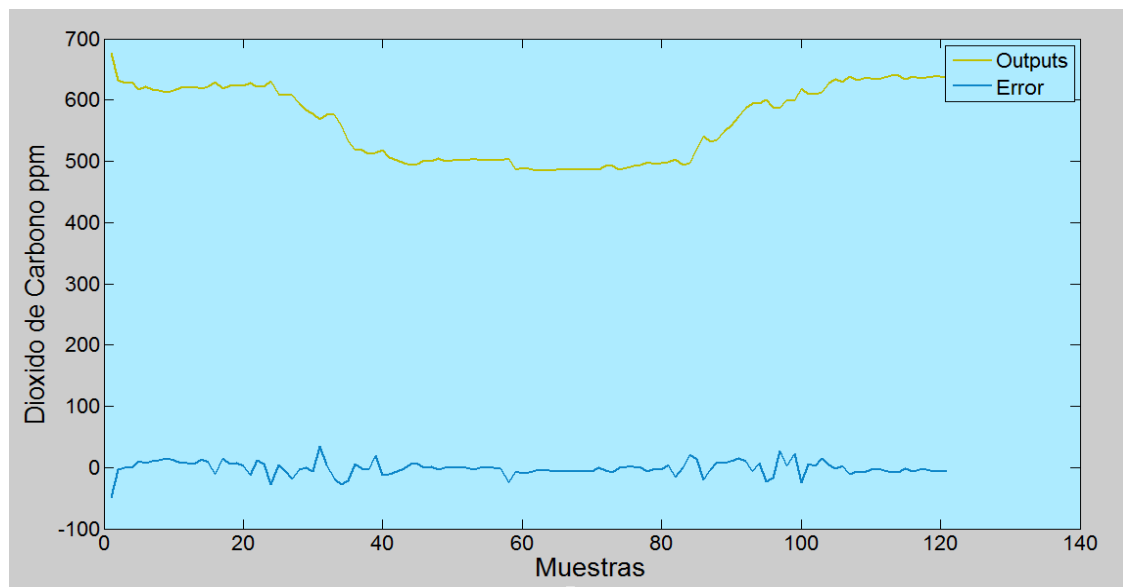


Figura 49. Curva de Predicción.

Fuente: Elaboración los autores.

Luego de realizar el proceso correspondiente y tal como se observa en la figura 49 sobre la Curva de Predicción, tenemos el resultado de una muestra de un día de valores de CO₂, que corresponden a la toma de datos del día 23 de Octubre del 2012 para la evaluación correspondiente que a su vez nos dio como predicción los datos del día 24 de Octubre del 2012.

f.2.4 Resumen de los resultados de los modelos utilizando las RNA.

Tabla 7. Modelos con Redes Neuronales Autorregresiva. (RNA)

VARIABLES	# DELAYS	# NEURONAS	MSE	(R)
TEMPERATURA	10	30	0.013	0.996
TEMPERATURA 1	20	10	0.012	0.996
TEMPERATURA 2	22	15	0.012	0.995
TEMPERATURA 3	13	23	0.011	0.996
HUMEDAD RELATIVA	10	30	0.171	0.965
HUMEDAD RELATIVA 1	20	10	0.173	0.962
HUMEDAD RELATIVA 2	22	15	0.161	0.963
HUMEDAD RELATIVA 3	13	23	0.196	0.967
CO ₂	10	30	136	0.982
CO ₂ 1	20	10	138	0.981
CO ₂ 2	22	15	134	0.980
CO₂ 3	13	23	146	0.982

Fuente: Elaboración los autores.

g.- DISCUSIÓN

En resumen, dentro de este trabajo investigativo se realiza un estudio para la predicción de las siguientes variables: Temperatura Ambiente, Humedad Relativa y CO₂, en el invernadero de Orquídeas de la universidad Nacional de Loja, aportando así al estudio de microclimas en nuestra ciudad.

El proyecto de investigación se lo realizó en la ciudad de Loja bajo sus condiciones ambientales, la adquisición de datos necesarios para generar los modelos finales se realizó con el equipo Fluke 975 Air Meter, obteniendo las variables antes mencionadas.

Para realizar las diferentes pruebas se recopiló un conjunto de datos tomados en los meses de Octubre y Noviembre del 2012, con un intervalo de tiempo de muestreo de diez minutos, dándonos como resultado 144 datos diarios y 8 640 datos en total.

La validación y predicción se la realizó con los mismos datos tomados de los meses de Octubre y Noviembre.

Según los datos obtenidos por el Fluke 975 el invernadero de nuestra Universidad cumple con los requerimientos de hábitat de la Orquídeas que allí se encuentran, lo mostramos en la siguiente tabla.

Tabla 8. Valores óptimos de las variables.

VARIABLES	HABITAT	INVERNADERO
Temperatura	16 °C a 24 °C	18,18 °C
Humedad Relativa	65 % y 85 %	83,94 %
CO₂	400 ppm y 800 ppm	605,1 ppm

Fuente: Elaboración los autores.

Cabe señalar que se realiza el riego al invernadero de Orquídeas todos los días Viernes en la mañana, según el monitoreo del Fluke 975 las variables en estudio varían sus valores debido al riego pero esto no influye significativamente en la estructura de la Red.

Tabla 9. Mejores resultados utilizando las RNA

VARIABLES	# DELAYS	# NEURONAS	MSE	R
TEMPERATURA	13	23	0.011	0.996
HUMEDAD	13	23	0.196	0.967
CO 2	13	23	146	0.982

Fuente: Elaboración los autores.

En la tabla 9 se muestra los mejores resultados después de haber realizado las diferentes pruebas de entrenamiento con la NAR de las variables de: Temperatura, Humedad Relativa y CO2.

Los mismos que tienen 13 Delays y 23 Neuronas, las demás pruebas realizadas nos daban como resultado valores del MSE y REGRESION no acordes a nuestros objetivos, por lo que despreciamos esos valores.

Finalmente, podemos señalar que de acuerdo a los diferentes análisis y pruebas que realizamos al conjunto de datos recopilados, encontramos la mejor estructura de la RNA que se acopla a nuestro caso de estudio de microclimas, teniendo como resultado una óptima validación y predicción de nuestra Red Neuronal.

h.- CONCLUSIONES

- Con las técnicas de Inteligencia Artificial se modeló y simuló las variables de Temperatura, Humedad Relativa y CO₂ con una precisión por encima del 96 %.
- Con la base de datos recopilada, se crearon las entradas de la Red Neuronal, y se obtuvieron las variables internas de Temperatura, Humedad Relativa y CO₂.
- Se establece que la estructura de la Red Neuronal presentó un excelente desempeño con la variable de Temperatura, la misma que presenta un valor de regresión del 99,6 % en la creación de la red, y un error de predicción que está entre el -0,28 % al 1,06 %.
- Se constituye que con la variable de Humedad Relativa al crear la estructura de la Red Neuronal, presentó un valor óptimo de regresión del 96,7 % y un error de predicción que está entre el -2,33 % al 1,50 %.
- Con la base de datos de la variable de CO₂, se obtuvo un valor de regresión del 98,2 %, y un error de predicción que está entre el -4,87 % al 3,38 % demostrando que las Redes Neuronales son realmente precisas a la hora de calcular predicción de microclimas en un invernadero.

i.- RECOMENDACIONES

- Continuar investigando en la carrera, estas aplicaciones con inteligencia artificial, en la actualidad la mayoría de procesos mediana y gran complejidad se los resuelve con el uso de la inteligencia artificial.
- Utilizar el código fuente de esta investigación cuando se realicen proyectos similares, por cuanto es flexible a modificaciones en función de los requerimientos del usuario, se puede extender la base de datos a un intervalo de tiempo mayor, con lo que se generara que los resultados tengan mayor precisión en la predicción de las variables.
- Instalar sensores de monitoreo en el invernadero con fines didácticos, con lo que se generaran bases de datos de intervalos de tiempo más prolongados, y estos datos puedan ser utilizados para futuras investigaciones en la carrera.
- Realizar diferentes pruebas para el entrenamiento de la Red Neuronal, obteniendo así una óptima estructura de la misma, siendo esta la base fundamental para la predicción de las diferentes variables.

j.- BIBLIOGRAFÍA

Libros

- Castilla, N. 2005. Invernaderos de plástico: tecnología y manejo. Editorial Mundi-Prensa. España. 37-146 pp.
- Cortés D., G. J. (2002, mayo). *Uso de Redes Neuronales para Analizar Modelos Mecánicos de Tejidos Biológicos*. Bogotá, Colombia.
- Costa Branco, P. (1998). learning from exmples using Fuzzy Logic in modelling
- J.-S. R. Jang. (1993). *ANFIS: adaptative-network-based fuzzy inference system*. (Vol. 23).
- I. Vinuela, P. Galván, M. Inés, “*Redes de Neuronas Artificiales: Un Enfoque Práctico*”, Prentice Hall. Madrid 2004.

Revistas

- Cabrera Samaniego, J. (2012).Script Ajuste Data. *Proceso de Depuración de datas*. Ecuador-Loja, *Revista*.
- L., R. P. (s.f.). Identificacion de motor de induccion con bomba centrifuga usando lógica difusa. *revista energetica*.

Tesis

- CHAHUARA Quispe, José Carlos, Torre. 2005. Tesis. Universidad Nacional Mayor de San Marcos. Facultad de Ingeniería Electrónica. EAP de Ingeniería Electrónica. Lima, Perú.
- COSTA Branco, P. J. 1998. Learning from examples using Fuzzy Logic in modelling and control of an electro-hydraulic actuator. Tesis Doctoral. Instituto Superior Técnico de Lisboa.
- A. Munoz, “*Aplicación de Técnicas de Redes Neuronales Artificiales al Diagnostico de Procesos Industriales*”, Tesis Doctoral, Universidad Pontificia Comillas de Madrid. 1996.
- D. Rivero, “*Desarrollo y Simplificación de Redes de Neuronas Artificiales mediante el uso de Técnicas de Computación Evolutiva*”, Tesis Doctoral, Universidad de Coruña. 2007.

Páginas web

- *Mathworks.com.* (s.f.). Obtenido de • <http://www.mathworks.com>
- *Mathworks.com.* (s.f.). Obtenido de • http://www.mathworks.es/programs/trials/trial_request.html?eventid=661447004&s_cid=vid_ui_trials
- *www.monografias.com.* (s.f.). Obtenido de <http://www.monografias.com/trabajos22/orquideas/orquideas.shtml>

k. Anexos

Anexo #1

- Adecuación para dejar el Fluke 975 en el invernadero de Orquídeas

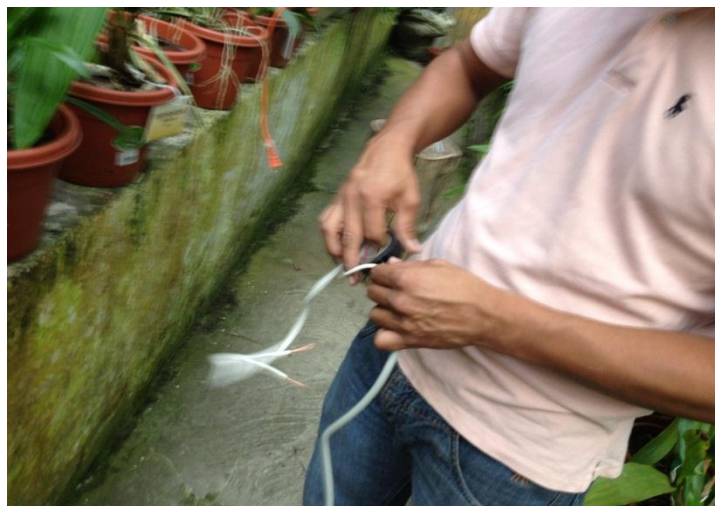


Figura 50. Conexiones eléctricas para adecuar el Fluke 975 dentro del Invernadero.

Fuente: Elaboración los autores.

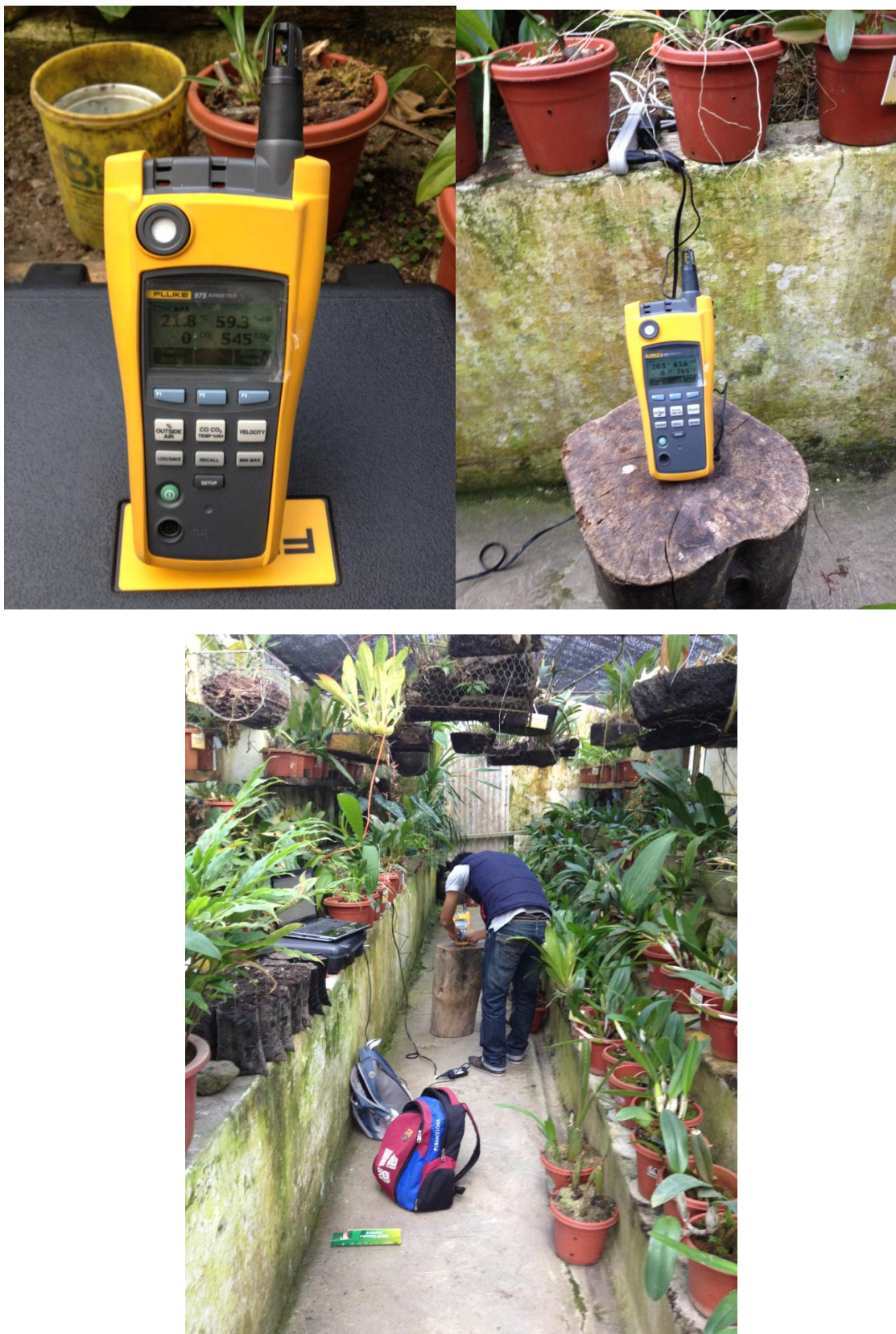


Figura 51. Primeras pruebas del funcionamiento del equipo de censado.

Fuente: Elaboración los autores.

Anexo #2

- **Script 1. Proceso de Autoajuste de datos.**

Nota: El presente proceso de autoajuste de datos es el mismo; tanto para Temperatura Ambiente, Humedad Relativa y CO2. Así como también para los días de predicción y validación de nuestra RNA, simplemente hay que ir cambiando las variables a ser ajustadas.

```
function [DT] = AjusteData(T)

%*****GUIA DE ELEMENTOS DE LA FUNCIÓN*****
%*****

% fitresult ==> Modelo resultante del ajuste de la Data.

%*****

%% PÁMETROS BÁSICOS DE LA FUNCIÓN

    [T,~] = size(T);
    muestras = 1:T;
    muestras = muestras';
    mu3 = mean(T); % Promedio de la Data
    sigma3 = std(T);

%% PRE-PROCESAMIENTO DE LOS DATOS
    % Preparando datos de las curvas

    [xData, yData] = prepareCurveData( muestras, T);

    % Seteo de las opciones de ajuste.

    ft = 'nearestinterp'; % Configurando el modelo de ajuste
    opts = fitoptions(ft);
    opts.Normalize = 'on';

    % Determinación de las exclusiones en los datos

    ex = excludedata( xData, yData, 'range',...
        [(-1*sigma3)+mu3 (1*sigma3)+mu3] );
```

```

opts.Exclude = ex;

%% DETERMINACIÓN DEL MODELO DE AJUSTE
    % Determinación modelo de ajuste (CuveFitingToolBox)

[fitresult] = fit( xData, yData, ft, opts );

    % Gráfico de data ajustada vs original y valores excluidos

figure( 'Name', 'Análisis estadístico de los datos' )
h = plot( fitresult, xData, yData, ex );
legend( h, 'Temperatura vs. Tiempo', 'Exclusiones vs. Tiempo',...
'Temperatura', 'Location', 'SouthEast' );
title('Datos'),xlabel( 'Muestras' ),ylabel( 'Temperatura' );
grid on

    % Presentando datos de salida

Modelo = fitresult;
DataLista = feval(fitresult,muestras);

```

Anexo# 3

- **Script 2. Proceso de creación de la Red Neuronal con datos de Temperatura.**

```
%%*****CREACIÓN DE LA RED NEURONAL CON VALORES DE TEMPERATURA*****
%%*****
%%*****GUIA DE ELEMENTOS*****
%%*****
% Temp ==> Datos de Temperatura.
% DTtemp ==> Datos autoajustados de Temperatura.
% net ==> Red Neuronal a ser evaluada.
%%*****
%%*****
%% Importar datos de Temperatura a Matlab
% Importar datos totales adquiridos de Temperatura.
load('Temp.mat');

% Grafica de los datos totales adquiridos.
% Create figure
figure1 = figure;
% Create axes
axes1 = axes('Parent',figure1,...
    'Color',[0.941176474094391 0.941176474094391 0.941176474094391]);
box(axes1,'on');
hold(axes1,'all');
% Create plot
plot(Temp,'Parent',axes1,'LineWidth',2,'DisplayName','Temp');
% Create xlabel
xlabel({'Muestras'},'FontSize',12);
% Create ylabel
ylabel({'Temperatura'},'FontSize',12);
% Create legend
legend1 = legend(axes1,'show');
set(legend1,'FontSize',12);

% Autoajuste de los datos totales de Temperatura
[DTtemp] = AjusteData1(Temp);

%% Red Neuronal mediante "ntstool"
ntstool
targetSeries = tonndata(DTtemp,false,false);

% Crear una Red Autorregresiva no lineal
feedbackDelays = 1:23;
hiddenLayerSize = 13;
net = narnet(feedbackDelays,hiddenLayerSize);

% Preparar los datos para la Formación y Simulación
% Uso PREPARETS le permite mantener sus datos originales de series de
% tiempo sin cambios, mientras que personaliza para redes con
diferentes
% números de los retrasos
[inputs,inputStates,layerStates,targets] =
preparets(net,{}, {},targetSeries);

% División de configuración de datos de entrenamiento , validación y
% prueba.
```

```

net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Entrenamiento de la red.
[net,tr] = train(net,inputs,targets,inputStates,layerStates);

% Prueba de la red.
outputs = net(inputs,inputStates,layerStates);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs);

% Ver la red
view(net);

% Cerrar el circuito de la red
% Utilice esta red para hacer la predicción.
% El CLOSELOOP sustituye a la entrada, es decir a los datos
autoajustados
% de Temperatura, con una directa conexión.
netc = closeloop(net);
[xc,xic,aic,tc] = preparets(netc,{}, {}, targetSeries);
yc = netc(xc,xic,aic);
perfc = perform(net,tc,yc);

% Red de predicción temprana
% La red se puede hacer para volver su salida a un paso de tiempo
temprano
% por la eliminación de un retraso, de manera que su retraso mínimo
sera
% ahora 0 en lugar de 1 . La nueva red devuelve los mismos resultados
que
% la red original , pero las salidas se desplazan dejando un paso de
tiempo.
nets = removedelay(net);
[xs,xis,ais,ts] = preparets(nets,{}, {}, targetSeries);
ys = nets(xs,xis,ais);
closedLoopPerformance = perform(net,tc,yc);

%% Graficas de los resultados de la red
% Grafica del rendimiento de validación.
% Create figure
figure1 = figure('Tag','TRAINING_PLOTPERFORM','NumberTitle','off',...
    'Name','Neural Network Training Performance (plotperform), Epoch
22, Validation stop.');
```

```

% Create multiple lines using matrix input to semilogy
semilogy1 = semilogy(X1,YMatrix1,'Parent',axes1,'LineWidth',2);
set(semilogy1(1),'Color',[0 0 1],'DisplayName','Train');
set(semilogy1(2),'Color',[0 0.8 0],'DisplayName','Validation');
set(semilogy1(3),'Color',[1 0 0],'DisplayName','Test');
```

```

% Create ylabel
ylabel('Mean Squared Error (mse)','FontWeight','bold','FontSize',12);
```

```

% Create xlabel
xlabel('22 Epochs','FontWeight','bold','FontSize',12);
% Create legend
legend(axes1,'show');
% Grafica del error del histograma.
% Create figure
figure1 = figure('Tag','TRAINING_PLOTERRHIST','NumberTitle','off',...
    'Name','Neural Network Training Error Histogram (ploterrhist),
Epoch 22, Validation stop. ');
%% Uncomment the following line to preserve the X-limits of the axes
% xlim(axes1,[-0.973004021036363 1.91611304767748]);
%% Uncomment the following line to preserve the Y-limits of the axes
% ylim(axes1,[0 6369]);
box(axes1,'on');

% Create ylabel
ylabel('Instances','FontWeight','bold','FontSize',12);

% Create text
text('Parent',axes1,'Units','normalized',...
    'String','Errors = Targets - Outputs',...
    'Position',[0.5 -0.22 0],...
    'HorizontalAlignment','center',...
    'FontWeight','bold',...
    'FontSize',12);

% Create legend
legend1 = legend(axes1,'show');
set(legend1,'FontSize',12,'Color',[1 1 1]);

% Grafica de la regresión.
% Create figure
figure1 =
figure('Tag','TRAINING_PLOTREGRESSION','NumberTitle','off',...
    'Name','Neural Network Training Regression (plotregression), Epoch
22, Validation stop. ');

% Create axes
axes1 = axes('Parent',figure1,...
    'Position',[0.13 0.593876272242609 0.304276096520464
0.331123727757391],...
    'PlotBoxAspectRatio',[1 1 1]);
%% Uncomment the following line to preserve the X-limits of the axes
% xlim(axes1,[16.21869101456 20.3613094720354]);
%% Uncomment the following line to preserve the Y-limits of the axes
% ylim(axes1,[16.21869101456 20.3613094720354]);
box(axes1,'on');

% Create xlabel
xlabel('Target','FontWeight','bold','FontSize',12);
% Create ylabel
ylabel('Output ~= 0.99*Target +
0.15','FontWeight','bold','FontSize',12);
% Create legend
legend1 = legend(axes1,'show');
set(legend1,'Location','NorthWest');

```

Anexo # 4

• Script 3. Proceso de creación de la Red Neuronal con datos de Humedad Relativa.

```
%%*CREACIÓN DE LA RED NEURONAL CON VALORES DE HUMEDAD RELATIVA*****
%*****
%*****GUIA DE ELEMENTOS*****
%*****
% HR ==> Datos de Humedad Relativa.
% DThr ==> Datos autoajustados de Humedad Relativa.
% net ==> Red Neuronal a ser evaluada.
%*****
%*****
%% Importar datos de Humedad Relativa a Matlab
% Importar datos totales adquiridos de Temperatura.
load('HR.mat');

% Grafica de los datos totales adquiridos.
% Create figure
figure1 = figure;
% Create axes
axes1 = axes('Parent',figure1,...
    'Color',[0.941176474094391 0.941176474094391
    0.941176474094391]);
box(axes1,'on');
hold(axes1,'all');

% Create plot
plot(HR,'LineWidth',2,'DisplayName','HR');

% Create xlabel
xlabel({'Muestras'});

% Create ylabel
ylabel({'Humedad Relativa %'});

% Autoajuste de los datos totales de Humedad Relativa
[DThr] = AjusteData2(HR);
%% Red Neuronal mediante "ntstool"
ntstool
targetSeries = tonndata(DThr,false,false);

% Crear una Red Autorregresiva no lineal
feedbackDelays = 1:23;
hiddenLayerSize = 13;
net = narnet(feedbackDelays,hiddenLayerSize);

% Preparar los datos para la Formación y Simulación
% Uso PREPARETS le permite mantener sus datos originales de series
de
% tiempo sin cambios, mientras que personaliza para redes con
diferentes
% números de los retrasos.
[inputs,inputStates,layerStates,targets] =
preparets(net,{}, {},targetSeries);
```

```

% División de configuración de datos de entrenamiento , validación
y
% prueba.
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Entrenamiento de la red.
[net,tr] = train(net,inputs,targets,inputStates,layerStates);
% Prueba de la red.
outputs = net(inputs,inputStates,layerStates);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs);
% Ver la red
view(net)

% Cerrar el circuito de la red
% Utilice esta red para hacer la predicción.
% El CLOSELOOP sustituye a la entrada, es decir a los datos
autoajustados
% de Humedad Relativa, con un directa conexión.
netc = closeloop(net);
[xc,xic,aic,tc] = preparets(netc,{}, {}, targetSeries);
yc = netc(xc,xic,aic);
perfc = perform(netc,tc,yc);

% Red de predicción temprana
% La red se puede hacer para volver su salida a un paso de tiempo
temprano
% por la eliminación de un retraso, de manera que su retraso mínimo
sera
% ahora 0 en lugar de 1 . La nueva red devuelve los mismos
resultados que
% la red original , pero las salidas se desplazan dejando un paso
de tiempo.
nets = removedelay(net);
[xs,xis,ais,ts] = preparets(nets,{}, {}, targetSeries);
ys = nets(xs,xis,ais);
closedLoopPerformance = perform(netc,tc,yc);

%% Graficas de los resultados de la red
% Grafica del rendimiento de validación.
% Create figure
figure1 =
figure('Tag','TRAINING_PLOTPERFORM','NumberTitle','off',...
'Name','Neural Network Training Performance (plotperform),
Epoch 22, Validation stop.');
```

```

% Create multiple lines using matrix input to semilogy
semilogy1 = semilogy(X1,YMatrix1,'Parent',axes1,'LineWidth',2);
set(semilogy1(1),'Color',[0 0 1],'DisplayName','Train');
set(semilogy1(2),'Color',[0 0.8 0],'DisplayName','Validation');
set(semilogy1(3),'Color',[1 0 0],'DisplayName','Test');
% Create ylabel
ylabel('Mean Squared Error
(mse)','FontWeight','bold','FontSize',12);
% Create xlabel
xlabel('22 Epochs','FontWeight','bold','FontSize',12);
```

```

% Create legend
legend(axes1,'show');
% Grafica del error del histograma.
% Create figure
figure1 =
figure('Tag','TRAINING_PLOTERRHIST','NumberTitle','off',...
      'Name','Neural Network Training Error Histogram (ploterrhist),
Epoch 22, Validation stop.');
```



```

%% Uncomment the following line to preserve the X-limits of the
axes
% xlim(axes1,[-0.973004021036363 1.91611304767748]);
%% Uncomment the following line to preserve the Y-limits of the
axes
% ylim(axes1,[0 6369]);
box(axes1,'on');
% Create ylabel
ylabel('Instances','FontWeight','bold','FontSize',12);
% Create text
text('Parent',axes1,'Units','normalized',...
     'String','Errors = Targets - Outputs',...
     'Position',[0.5 -0.22 0],...
     'HorizontalAlignment','center',...
     'FontWeight','bold',...
     'FontSize',12);
% Create legend
legend1 = legend(axes1,'show');
set(legend1,'FontSize',12,'Color',[1 1 1]);
```



```

% Grafica de la regresión.
% Create figure
figure1 =
figure('Tag','TRAINING_PLOTREGRESSION','NumberTitle','off',...
      'Name','Neural Network Training Regression (plotregression),
Epoch 22, Validation stop.');
```



```

% Create axes
axes1 = axes('Parent',figure1,...
     'Position',[0.13 0.593876272242609 0.304276096520464
0.331123727757391],...
     'PlotBoxAspectRatio',[1 1 1]);
%% Uncomment the following line to preserve the X-limits of the
axes
% xlim(axes1,[16.21869101456 20.3613094720354]);
%% Uncomment the following line to preserve the Y-limits of the
axes
% ylim(axes1,[16.21869101456 20.3613094720354]);
box(axes1,'on');
```



```

% Create xlabel
xlabel('Target','FontWeight','bold','FontSize',12);
% Create ylabel
ylabel('Output ~= 0.99*Target +
0.15','FontWeight','bold','FontSize',12);
% Create legend
legend1 = legend(axes1,'show');
set(legend1,'Location','NorthWest')
```


Anexo # 5

• Script 4. Proceso de creación de la Red Neuronal con datos de CO2.

```
%%**CREACIÓN DE LA RED NEURONAL CON VALORES DE DIOXIDO DE CARBONO**
%*****
%*****GUIA DE ELEMENTOS*****
%*****
% CO2 ==> Datos de Dióxido de Carbono.
% DTco2 ==> Datos autoajustados de Dióxido de Carbono.
% net => Red Neuronal a ser evaluada.
%*****
%*****
%% Importar datos de Dióxido de Carbono a Matlab
% Importar datos totales adquiridos de Temperatura.
load('CO2.mat');
% Grafica de los datos totales adquiridos.
% Create figure
figure1 = figure;

% Create axes
axes1 = axes('Parent',figure1,...
    'Color',[0.941176474094391 0.941176474094391
0.941176474094391]);
box(axes1,'on');
hold(axes1,'all');

% Create plot
plot(CO2,'LineWidth',2,'DisplayName','CO2');

% Create xlabel
xlabel({'Muestras'});

% Create ylabel
ylabel({'Dioxido de Carbono ppm'});
% Autoajuste de los datos totales de Dioxido de Carbono
[DTco2] = AjusteData3(CO2);
%% Red Neuronal mediante "ntstool"
ntstool
targetSeries = tonndata(DTco2,false,false);

% Crear una Red Autorregresiva no lineal
feedbackDelays = 1:23;
hiddenLayerSize = 13;
net = narnet(feedbackDelays,hiddenLayerSize);

% Preparar los datos para la Formación y Simulación
% Uso PREPARETS le permite mantener sus datos originales de series
de tiempo sin cambios, mientras que personaliza para redes con
diferentes
% números de los retrasos.
[inputs,inputStates,layerStates,targets] =
preparets(net,{}, {},targetSeries);
% División de configuración de datos de entrenamiento , validación
y prueba.
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
```

```

% Entrenamiento de la red.
[net,tr] = train(net,inputs,targets,inputStates,layerStates);

% Prueba de la red.
outputs = net(inputs,inputStates,layerStates);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs);

% Ver la red
view(net)
% Cerrar el circuito de la red
% Utilice esta red para hacer la predicción.
% El CLOSELOOP sustituye a la entrada, es decir a los datos
autoajustados
% de Humedad Relativa, con un directa conexión.
netc = closeloop(net);
[xc,xic,aic,tc] = preparets(netc,{}, {}, targetSeries);
yc = netc(xc,xic,aic);
perfc = perform(net,tc,yc);

% Red de predicción temprana
% La red se puede hacer para volver su salida a un paso de tiempo
temprano
% por la eliminación de un retraso, de manera que su retraso mínimo
sera
% ahora 0 en lugar de 1 . La nueva red devuelve los mismos
resultados que
% la red original , pero las salidas se desplazan dejando un paso
de tiempo.
nets = removedelay(net);
[xs,xis,ais,ts] = preparets(nets,{}, {}, targetSeries);
ys = nets(xs,xis,ais);
closedLoopPerformance = perform(net,tc,yc);
%% Graficas de los resultados de la red
% Grafica del rendimiento de validación.
% Create figure
figure1 =
figure('Tag','TRAINING_PLOTPERFORM','NumberTitle','off',...
'Name','Neural Network Training Performance (plotperform),
Epoch 22, Validation stop. ');
% Create multiple lines using matrix input to semilogy
semilogy1 = semilogy(X1,YMatrix1,'Parent',axes1,'LineWidth',2);
set(semilogy1(1),'Color',[0 0 1],'DisplayName','Train');
set(semilogy1(2),'Color',[0 0.8 0],'DisplayName','Validation');
set(semilogy1(3),'Color',[1 0 0],'DisplayName','Test');
% Create ylabel
ylabel('Mean Squared Error
(mse)','FontWeight','bold','FontSize',12);
% Create xlabel
xlabel('22 Epochs','FontWeight','bold','FontSize',12);
% Create legend
legend(axes1,'show');
% Grafica del error del histograma.
% Create figure
figure1 =
figure('Tag','TRAINING_PLOTERRHIST','NumberTitle','off',...
'Name','Neural Network Training Error Histogram (ploterrhist),
Epoch 22, Validation stop. ');

```

```

%% Uncomment the following line to preserve the X-limits of the
axes
% xlim(axes1,[-0.973004021036363 1.91611304767748]);
%% Uncomment the following line to preserve the Y-limits of the
axes
% ylim(axes1,[0 6369]);
box(axes1,'on');

% Create ylabel
ylabel('Instances','FontWeight','bold','FontSize',12);
% Create text
text('Parent',axes1,'Units','normalized',...
     'String','Errors = Targets - Outputs',...
     'Position',[0.5 -0.22 0],...
     'HorizontalAlignment','center',...
     'FontWeight','bold',...
     'FontSize',12);
% Create legend
legend1 = legend(axes1,'show');
set(legend1,'FontSize',12,'Color',[1 1 1]);
% Grafica de la regresión.
% Create figure
figure1 =
figure('Tag','TRAINING_PLOTREGRESSION','NumberTitle','off',...
     'Name','Neural Network Training Regression (plotregression),
Epoch 22, Validation stop.');
```



```

% Create axes
axes1 = axes('Parent',figure1,...
     'Position',[0.13 0.593876272242609 0.304276096520464
0.331123727757391],...
     'PlotBoxAspectRatio',[1 1 1]);
%% Uncomment the following line to preserve the X-limits of the
axes
% xlim(axes1,[16.21869101456 20.3613094720354]);
%% Uncomment the following line to preserve the Y-limits of the
axes
% ylim(axes1,[16.21869101456 20.3613094720354]);
box(axes1,'on');
% Create xlabel
xlabel('Target','FontWeight','bold','FontSize',12);

% Create ylabel
ylabel('Output ~= 0.99*Target +
0.15','FontWeight','bold','FontSize',12);

% Create legend
legend1 = legend(axes1,'show');
set(legend1,'Location','NorthWes')

```

Anexo # 6

Script 5. Proceso predicción y validación con datos de Temperatura.

```
%%*****PREDICCIÓN Y VALIDACIÓN DE DATOS DE TEMPERATURA*****
%%*****
%%*****GUÍA DE ELEMENTOS*****
%%*****
% A1 => Entradas de valoración para la Red.
% A2=> Estados iniciales de la serie temporal.
% Val => Datos de validación para la Red.
% DA1 => Entradas de valoración para la Red autoajustados.
% DA2=> Estados iniciales de la serie temporal autoajustados.
% DVal => Datos de validación para la Red autoajustados.
% outputs => Resultados de predicción de la serie temporal.
% e => Errores de predicción.
% net => Red Neuronal a ser evaluada.
%%*****
%% IMPORTAR RED NEURONAL Y DATOS DE TEMPERATURA
% Importar entradas de valoración
load('A1.mat')
% Autoajustar entradas de valoración
[DA1] = AjusteData1a(A1);
% Importar estados iniciales
load('A2.mat')
% Autoajustar estados iniciales
[DA2] = AjusteData1a(A2);

% Importar datos de validación
load('Val.mat')
% Autoajustar datos de validación
[DVal] = AjusteData1a(Val);
% Importar Red Neuronal de Temperatura
load('net.mat')
%% PRE-POCESAMIENTO DE LOS DATOS
% Establecimiento de vector de entrada completo
conct = vertcat(DA2,DA1);
% Estableciendo serie-objetivo para ser ingresada a la Red
targetSeries = tonndata(conct,false,false);
% Determinando datos necesarios para evaluar la Red
[inputs,inputStates,layerStates] = preparets(net,{}, {},targetSeries);

%% EVALUACIÓN DE LA RED NEURONAL
% Evaluando la Red
outputs = net(inputs,inputStates,layerStates);
outputs = cell2mat(outputs);

%% DETERMINACIÓN DE ERRORES DE PREDICCIÓN
% Calculo de los errores
e = gsubtract(DA1',outputs);
% Calculo del valor de regresión
[R,~,~] = regression(DVal',outputs);
% Calculo del error cuadrático medio
M = mse(e);
%% GRAFICAS DE COMPORTAMIENTO DE PREDICION DE DATOS
% Grafica de datos de predicción, validación y error.
% Create figure
grafico = figure;
```

```

% Create axes
axes1 = axes('Parent',grafico,...
    'Color',[0.941176474094391 0.941176474094391 0.941176474094391]);
box(axes1,'on');
hold(axes1,'all');

% Create multiple lines using matrix input to plot

plot(outputs,'LineWidth',2,'Color',[1 0.694117665290833
0.39215686917305],...
    'DisplayName','outputs');
hold on
plot(DVal,'LineWidth',2,'DisplayName','DVal');

hold on
plot(e,'LineWidth',2,...
    'Color',[0.235294118523598 0.235294118523598
0.235294118523598],...
    'DisplayName','error');
% Create xlabel
xlabel({'Muestras'},'FontSize',12);

% Create ylabel
ylabel({'Temperatura'},'FontSize',12);

% Create legend
legend1 = legend(axes1,'show');
set(legend1,'Color',[1 1 1]);

%Grafica del error del histograma.
hist(e)
%Grafica predicción y error.
% Create figure
grafico = figure;
% Create axes
axes1 = axes('Parent',grafico,...
    'Color',[0.678431391716003 0.921568632125854 1]);
box(axes1,'on');
hold(axes1,'all');
% Create multiple lines using matrix input to plot
plot(outputs,'Parent',axes1,'LineWidth',2,...
    'Color',[0.749019622802734 0.749019622802734 0],...
    'DisplayName','Outputs');
hold on
plot(e,'LineWidth',2,...
    'Color',[0.0431372560560703 0.517647087574005
0.780392169952393],...
    'DisplayName','Error');
% Create xlabel
xlabel({'Muestras'},'FontSize',12);

% Create ylabel
ylabel({'Temperatura'},'FontSize',12);

% Create legend
legend(axes1,'show');

```

Anexo # 7

Script 6. Proceso predicción y validación con datos de Humedad Relativa.

```
%%*****PREDICCIÓN Y VALIDACIÓN DE DATOS DE HUMEDAD RELATIVA*****
%%*****
%%*****GUÍA DE ELEMENTOS*****
%%*****
% A1 => Entradas de valoración para la Red.
% A2=> Estados iniciales de la serie temporal.
% Val => Datos de validación para la Red.
% DA1 => Entradas de valoración para la Red autoajustados.
% DA2=> Estados iniciales de la serie temporal autoajustados.
% DVal => Datos de validación para la Red autoajustados.
% outputs => Resultados de predicción de la serie temporal.
% e => Errores de predicción.
% net => Red Neuronal a ser evaluada.
%%*****
%%*****
%% IMPORTAR RED NEURONAL Y DATOS DE TEMPERATURA
% Importar entradas de valoración
load('A1.mat')
% Autoajustar entradas de valoración
[DA1] = AjusteData2a(A1);

% Importar estados iniciales
load('A2.mat')
% Autoajustar estados iniciales
[DA2] = AjusteData2a(A2);

% Importar datos de validación
load('Val.mat')
% Autoajustar datos de validación
[DVal] = AjusteData2a(Val);

% Importar Red Neuronal de Temperatura
load('net.mat')

%% PRE-POCESAMIENTO DE LOS DATOS
% Establecimiento de vector de entrada completo
conct = vertcat(DA2,DA1);
% Estableciendo serie-objetivo para ser ingresada a la Red
targetSeries = tonndata(conct,false,false);
% Determinando datos necesarios para evaluar la Red
[inputs,inputStates,layerStates] = preparets(net,{}, {},targetSeries);

%% EVALUACIÓN DE LA RED NEURONAL
% Evaluando la Red
outputs = net(inputs,inputStates,layerStates);
outputs = cell2mat(outputs);

%% DETERMINACIÓN DE ERRORES DE PREDICCIÓN
% Calculo de los errores
e = gsubtract(DA1',outputs);
% Calculo del valor de regresión
[R,~,~] = regression(DVal',outputs);
% Calculo del error cuadrático medio
```

```

M = mse(e);
% GRAFICAS DE COMPORTAMIENTO DE PREDICION DE DATOS
% Grafica de datos de predicción, validación y error.
% Create figure
grafico = figure;
% Create axes
axes1 = axes('Parent',grafico,...
    'Color',[0.941176474094391 0.941176474094391 0.941176474094391]);
box(axes1,'on');
hold(axes1,'all');

% Create multiple lines using matrix input to plot
plot(outputs,'LineWidth',2,'Color',[1 0.694117665290833
0.39215686917305],...
    'DisplayName','outputs');
hold on
plot(DVal,'LineWidth',2,'DisplayName','DVal');

hold on
plot(e,'LineWidth',2,...
    'Color',[0.235294118523598 0.235294118523598
0.235294118523598],...
    'DisplayName','error');
% Create xlabel
xlabel({'Muestras'},'FontSize',12);
% Create ylabel
ylabel({'Humedad Relativa'},'FontSize',12);
% Create legend
legend1 = legend(axes1,'show');
set(legend1,'Color',[1 1 1]);

%Grafica del error del histograma.
hist(e)
%Grafica predicción y error.
% Create figure
grafico = figure;
% Create axes
axes1 = axes('Parent',grafico,...
    'Color',[0.678431391716003 0.921568632125854 1]);
box(axes1,'on');
hold(axes1,'all');

% Create multiple lines using matrix input to plot
plot(outputs,'Parent',axes1,'LineWidth',2,...
    'Color',[0.749019622802734 0.749019622802734 0],...
    'DisplayName','Outputs');
hold on
plot(e,'LineWidth',2,...
    'Color',[0.0431372560560703 0.517647087574005
0.780392169952393],...
    'DisplayName','Error');
% Create xlabel
xlabel({'Muestras'},'FontSize',12);
% Create ylabel
ylabel({'Humedad Relativa'},'FontSize',12);
% Create legend
legend(axes1,'show');

```

Anexo # 8

Script 7. Proceso predicción y validación con datos de Dióxido de Carbono.

```
%%*****PREDICCIÓN Y VALIDACIÓN DE DATOS DE DIOXIDO DE CARBONO*****
%%*****
%%*****GUÍA DE ELEMENTOS*****
%%*****
% A1 => Entradas de valoración para la Red.
% A2=> Estados iniciales de la serie temporal.
% Val => Datos de validación para la Red.
% DA1 => Entradas de valoración para la Red autoajustados.
% DA2=> Estados iniciales de la serie temporal autoajustados.
% DVal => Datos de validación para la Red autoajustados.
% outputs => Resultados de predicción de la serie temporal.
% e => Errores de predicción.
% net => Red Neuronal a ser evaluada.
%%*****
%%*****
%% IMPORTAR RED NEURONAL Y DATOS DE DIOXIDO DE CARBONO
% Importar entradas de valoración
load('A1.mat')
% Autoajustar entradas de valoración
[DA1] = AjusteData3a(A1);

% Importar estados iniciales
load('A2.mat')
% Autoajustar estados iniciales
[DA2] = AjusteData3a(A2);

% Importar datos de validación
load('Val.mat')
% Autoajustar datos de validación
[DVal] = AjusteData3a(Val);

% Importar Red Neuronal de Dioxido de Carbono
load('net.mat')
%% PRE-POCESAMIENTO DE LOS DATOS
% Establecimiento de vector de entrada completo
conct = vertcat(DA2,DA1);
% Estableciendo serie-objetivo para ser ingresada a la Red
targetSeries = tonndata(conct,false,false);
% Determinando datos necesarios para evaluar la Red
[inputs,inputStates,layerStates] = preparets(net,{}, {},targetSeries);

%% EVALUACIÓN DE LA RED NEURONAL
% Evaluando la Red
outputs = net(inputs,inputStates,layerStates);
outputs = cell2mat(outputs);

%% DETERMINACIÓN DE ERRORES DE PREDICCIÓN
% Calculo de los errores
e = gsubtract(DA1',outputs);
% Calculo del valor de regresión
[R,~,~] = regression(DVal',outputs);
% Calculo del error cuadrático medio
M = mse(e);
```



```

% GRAFICAS DE COMPORTAMIENTO DE PREDICION DE DATOS
% Grafica de datos de predicción, validación y error.
% Create figure
grafico = figure;

% Create axes
axes1 = axes('Parent',grafico,...
    'Color',[0.941176474094391 0.941176474094391 0.941176474094391]);
box(axes1,'on');
hold(axes1,'all');

% Create multiple lines using matrix input to plot

plot(outputs,'LineWidth',2,'Color',[1 0.694117665290833
0.39215686917305],...
    'DisplayName','outputs');
hold on
plot(DVal,'LineWidth',2,'DisplayName','DVal');
hold on
plot(e,'LineWidth',2,...
    'Color',[0.235294118523598 0.235294118523598
0.235294118523598],...
    'DisplayName','error');
% Create xlabel
xlabel({'Muestras'},'FontSize',12);
% Create ylabel
ylabel({'Dioxido de Carbono'},'FontSize',12);
% Create legend
legend1 = legend(axes1,'show');
set(legend1,'Color',[1 1 1]);
%Grafica del error del histograma.
hist(e)
%Grafica predicción y error.
% Create figure
grafico = figure;
% Create axes
axes1 = axes('Parent',grafico,...
    'Color',[0.678431391716003 0.921568632125854 1]);
box(axes1,'on');
hold(axes1,'all');
% Create multiple lines using matrix input to plot
plot(outputs,'Parent',axes1,'LineWidth',2,...
    'Color',[0.749019622802734 0.749019622802734 0],...
    'DisplayName','Outputs');
hold on
plot(e,'LineWidth',2,...
    'Color',[0.0431372560560703 0.517647087574005
0.780392169952393],...
    'DisplayName','Error');
% Create xlabel
xlabel({'Muestras'},'FontSize',12);
% Create ylabel
ylabel({'Dioxido de Carbono'},'FontSize',12);
% Create legend
legend(axes1,'show');

```

Anexo # 9. Gráfica de autoajuste de datos totales de Temperatura

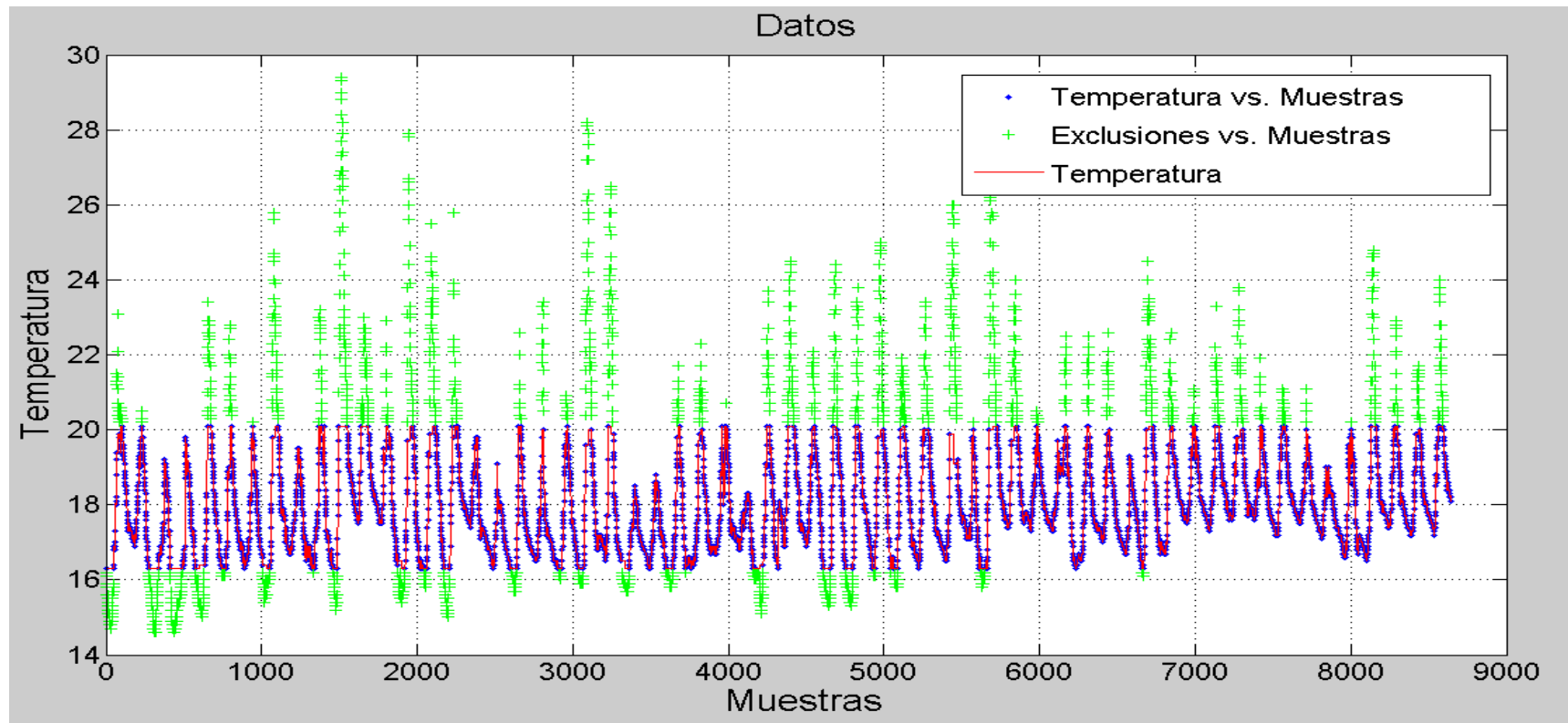


Figura 52. Temperatura vs Muestras (8640 datos ajustados)

Fuente: Elaboración los autores.

Anexo # 10. Gráfica de autoajuste de datos totales de Humedad Relativa.

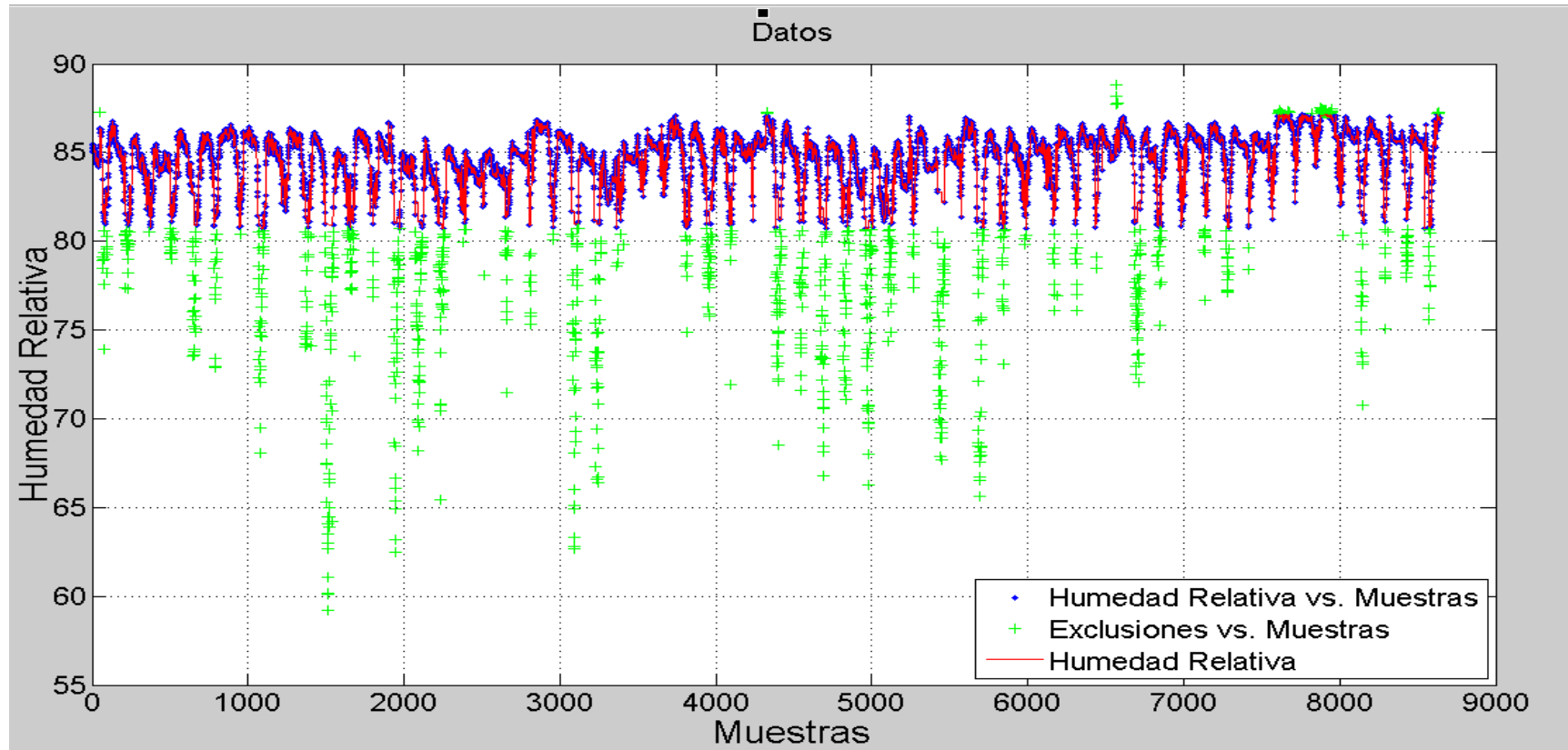


Figura 53. Humedad Relativa vs Muestras (8640 datos ajustados)

Fuente: Elaboración los autores.

Anexo # 11. Gráfica de autoajuste de datos totales de Dióxido de Carbono.

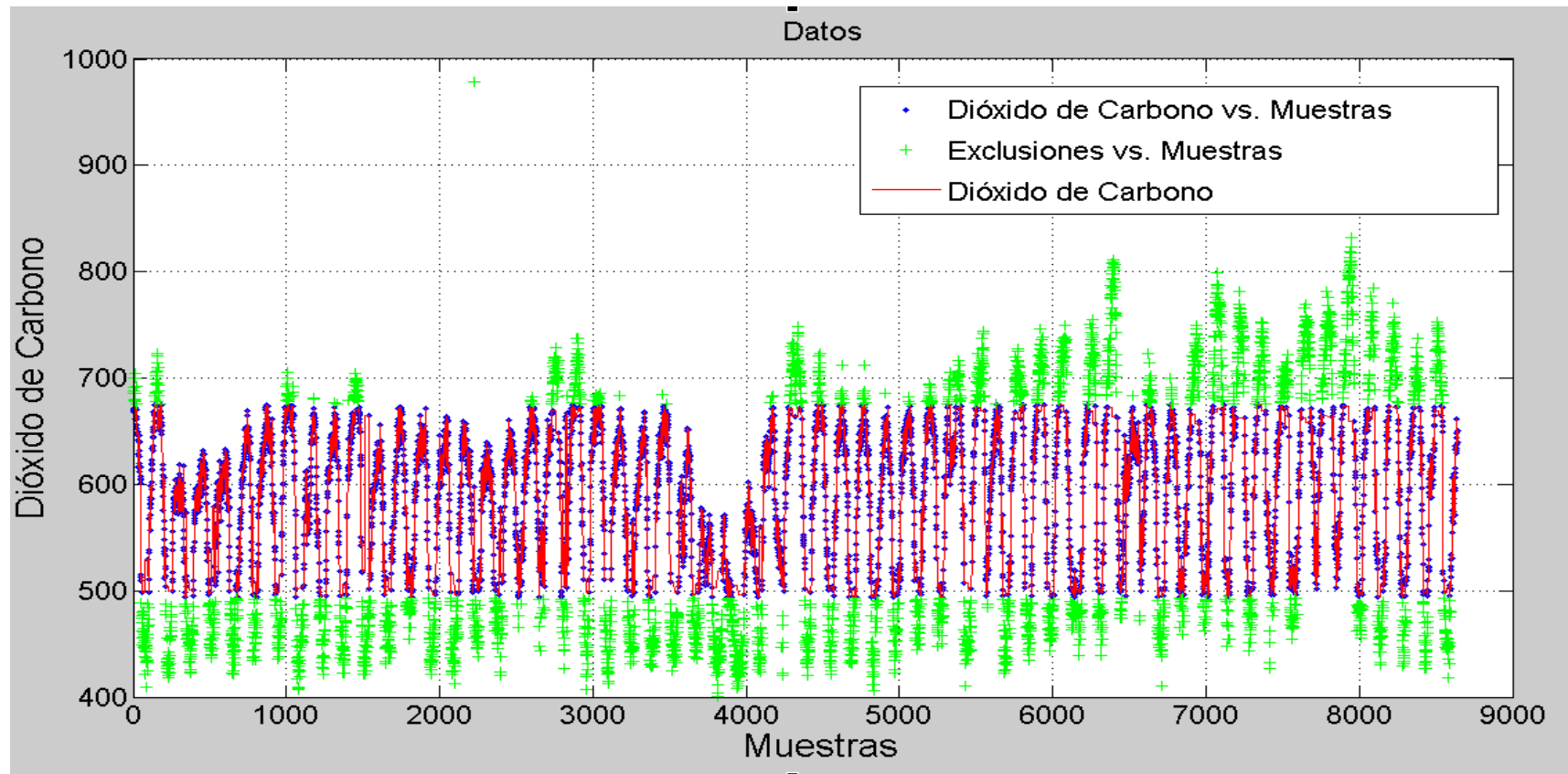


Figura 54. CO2 vs Muestras (8640 datos ajustados)

Fuente: Elaboración los autores.