



UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y COMUNICACIONES

TEMA:

**“SISTEMA DE CONTROL DIFUSO Y MONITOREO PARA GRANJAS
AVICOLAS UTILIZANDO SUELO RADIANTE CON ENERGÍA SOLAR”**

Proyecto de Trabajo de Graduación. Modalidad: Proyecto de investigación, presentado previo a la obtención del título de Ingeniero en Electrónica y Comunicaciones.

LINEA DE INVESTIGACIÓN: Tecnologías de la Información y Comunicación.

AUTOR: Cristhian Fernando Monta Toapanta.

TUTOR: Ing. Paulina Ayala, Mg.

Ambato – Ecuador

Enero 2020

APROBACIÓN DEL TUTOR

En mi calidad de tutor del Trabajo de Investigación sobre el tema: SISTEMA DE CONTROL DIFUSO Y MONITOREO PARA GRANJAS AVICOLAS UTILIZANDO SUELO RADIANTE CON ENERGÍA SOLAR, del señor MONTA TOAPANTA CRISTHIAN FERNANDO, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato, Enero del 2020

EL TUTOR



.....

Ing. Paulina Ayala, Mg

AUTORÍA

El presente Proyecto de Investigación titulado: **SISTEMA DE CONTROL DIFUSO Y MONITOREO PARA GRANJAS AVICOLAS UTILIZANDO SUELO RADIANTE CON ENERGÍA SOLAR**, es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, Enero del 2020.

A handwritten signature in blue ink, enclosed within a hand-drawn oval. The signature is stylized and appears to read 'Cristhian Monta Toapanta'. Below the signature is a horizontal dashed line.

Cristhian Fernando Monta Toapanta

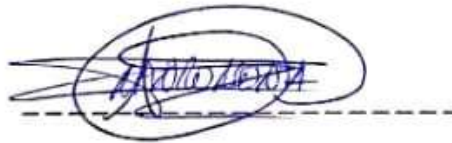
C.I.: 220058800-8

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato, Enero del 2020.

A handwritten signature in blue ink, enclosed within a hand-drawn oval. The signature is written over a horizontal dashed line.

Cristhian Fernando Monta Toapanta

C.I.: 220058800-8

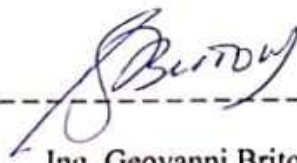
APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. Geovanni Brito e Ing. Santiago Altamirano, revisó y aprobó el Informe Final del trabajo de graduación titulado "SISTEMA DE CONTROL DIFUSO Y MONITOREO PARA GRANJAS AVÍCOLAS UTILIZANDO SUELO RADIANTE CON ENERGÍA SOLAR", presentado por el señor Cristhian Fernando Monta Toapanta de acuerdo al numeral 9.1 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.



Ing. Elsa Pilar Urrutia Mg.

PRESIDENTA DEL TRIBUNAL



Ing. Geovanni Brito

DOCENTE CALIFICADOR



Ing. Santiago Altamirano

DOCENTE CALIFICADOR

DEDICATORIA

A mis padres Sabina y Fernando que han sabido como guiarme por el camino del bien brindándome incondicionalmente su apoyo, cariño y amor.

A mis hermanos Romina y Josué que siempre hemos estado juntos en las buenas y en las malas.

A toda mi familia en especial a mi abuelita Electra que me ha sabido brindar sus palabras de aliento y muestras de cariño hacia mi persona. Infinitas gracias.

A María Elizabeth que ha compartido conmigo durante toda esta aventura universitaria.

Cristhian Fernando Monta Toapanta

AGRADECIMIENTO

A mis padres que han hecho posible todo esto gracias a sus esfuerzos y sacrificios y que cada día sienta más ganas de ser como ellos.

A mi Alma Mater la Universidad Técnica de Ambato, a la Facultad de Ingeniería en Sistemas, Electrónica e Industrial y a la Carrera de Ingeniería en Electrónica y Comunicaciones por haberme forjado profesionalmente.

A los docentes de mi querida Facultad que nos han sabido brindar sus conocimientos sin restricciones en especial a la Ingeniera Paulina Ayala quien con cariño ha sabido guiarme durante todo este proceso.

A mis amigos de aulas y de vida, gracias por permitirme llegar lejos.

Cristhian Fernando Monta Toapanta

ÍNDICE GENERAL DE CONTENIDOS

PORTADA.....	i
A.PÁGINAS PRELIMINARES	ii
APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DE LA COMISIÓN CALIFICADORA	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
RESUMEN EJECUTIVO	xv
ABSTRACT.....	xvi
B.CONTENIDOS	xvi
CAPITULO I.....	1
1.1.Antecedentes Investigativos.....	1
1.1.1.Contextualización del Problema	1
1.2.Fundamentación Teórica.....	2
1.2.1.La Industria de la Avicultura en el Ecuador.....	2
1.2.2.Controlador de Lógica Difusa.....	4
1.2.3.Monitoreo de procesos	8
1.2.4.Servidores.....	11
1.2.5.Energías Renovables	15
1.2.6.Suelo Radiante	21
1.2.7.Energía Solar Térmica	23
1.2.8.Sensores de Temperatura y Humedad.....	24
1.2.9.Monitoreo Industrial.....	26
1.2.10.Sistema de Tuberías	28
1.3.Objetivos	30

1.3.1.Objetivo General	30
1.3.2.Objetivos Específicos.....	30
CAPÍTULO II	31
2.1. Materiales.....	31
2.1.1. Raspberry Pi 3 Model B+.....	37
2.1.2. Arduino IDE.....	39
2.1.3. Wemos D1 mini	40
2.1.4. Bomba de agua 800l/h.....	41
2.1.5. Sensor de temperatura DS18B20	41
2.1.6. Sensor de humedad y temperatura DHT11	43
2.1.7. Driver doble para control de velocidad de motor TB6612FNG	43
2.1.8. Servidor LAMP.....	44
2.1.9. Punto de Acceso Inalámbrico N 450Mbps TL-WA901ND.....	45
2.1.10.Ventilación	45
2.1.11.Calentador de agua solar	46
2.1.12.Módulo RTC DS3231	47
2.2.Métodos.....	48
2.2.1.Modalidad de la Investigación:	48
2.2.2.Población y Muestra:.....	49
2.2.3.Recolección de Información:	49
2.2.4.Procesamiento y Análisis de Datos:	49
2.3.Desarrollo del Proyecto.....	51
2.3.1.Análisis de la situación actual de las naves avícolas.....	51
2.3.2.Estudio de temperatura, humedad y ventilación en las naves avícolas.....	52
2.3.3.Análisis y comparación de los Métodos de Inferencia Difuso.....	54
2.3.4.Desarrollo del Sistema de Control con Método de inferencia de Mamdani	56
2.3.5.Determinación del tipo de tecnología para el monitoreo de naves avícolas.	61

2.3.6.Desarrollo de la base de datos utilizando los servidores LAMP.....	62
2.3.7.Desarrollo de interfaz Web.	70
2.3.8.Programación del sistema de control difuso usando hardware y software libre	76
2.3.9.Selección del sistema de piso radiante óptimo para naves avícolas.....	79
2.3.10.Implementación del prototipo	81
2.3.11.Pruebas del prototipo	84
CAPÍTULO III	90
3.1.Análisis y discusión de los resultados	90
CAPÍTULO IV	96
4.1.Conclusiones	96
4.2.Recomendaciones.....	97
C. MATERIALES DE REFERENCIA.....	98
Bibliografía	98
Anexos	105
Anexo A: Códigos de programación en Raspberry Pi y Arduino.....	105
Anexo B. Fotografías de la instalación.	126

ÍNDICE DE FIGURAS

Figura 1. Encuesta de Superficie y Producción Agropecuaria ESPAC	3
Figura 2. Controlador difuso.....	4
Figura 3. Ejemplo de subconjuntos borrosos para el conjunto temperatura.	5
Figura 4. Ejemplo de fusificación de una variable.....	5
Figura 5. Representación gráfica de los distintos métodos de defusificación	8
Figura 6. Estructura de un sistema de monitoreo.....	10
Figura 7. Estructura de un servidor web	13
Figura 8. Clasificación de las energías renovables.	18
Figura 9. Esquema de funcionamiento de una instalación de energía solar activa	19
Figura 10. Esquema de funcionamiento ACS.....	20
Figura 11. Distribución en serpentín.....	29
Figura 12. Distribución en doble serpentín	29
Figura 13. Distribución en espiral.....	29
Figura 14. Diseño del prototipo del sistema de control y monitoreo.	31
Figura 15. Diagrama de control de bloques del proceso	32
Figura 16. Raspberry Pi 3 B+.....	39
Figura 17. Arduino IDE Software.....	40
Figura 18. Wemos D1 Mini	40
Figura 19. Bomba de agua 800l/h	41
Figura 20. Sensor DS18B20.....	42
Figura 21. Sensor DHT11	43
Figura 22. TB6612FNG	44
Figura 23. Servidor LAMP	45
Figura 24. Ventilador	46
Figura 25. Componentes de calentador solar	47
Figura 26. RTC 3231 Arduino	48
Figura 27. Estructura de funciones.....	51
Figura 28. Interfaz gráfica, método de inferencia tipo Mamdani.	56
Figura 29. Funciones de membresía para la variable temperatura.....	57
Figura 30. Funciones de membresía para la variable tiempoMotor.....	58
Figura 31. Funciones de membresía para la variable humedad.	58

Figura 32. Funciones de membresía para la variable ventilador.....	59
Figura 33. Editor de reglas difusas.....	60
Figura 34. Ventana visualizador de reglas.	60
Figura 35. Ventana principal PuTTY.....	63
Figura 36. Comando sudo su.....	63
Figura 37. Comando upgrade.....	64
Figura 38. Ruta archivo dhcpcd.conf.....	64
Figura 39. Edición de archivo dhcpcd.conf.....	64
Figura 40. Ingreso con ip estática.....	65
Figura 41. Comando apt-get install mariadb-server-10.0.....	65
Figura 42. Comando apt-get install python-mysqldb.....	65
Figura 43. Comando dpkg-reconfigure phpmyadmin.....	66
Figura 44. Ventana principal phpMyAdmin.....	66
Figura 45. Base de datos tesis.....	67
Figura 46. Tablas contenidas en base de datos tesis.....	67
Figura 47. Variables en tabla datos.....	68
Figura 48. Variables en tabla datos_fuzzy.....	69
Figura 49. Variables en tabla USUARIOS_PASS.....	69
Figura 50. Auto incremento.....	70
Figura 51. Comando apt-get install vsftpd.....	70
Figura 52. Ruta del fichero vsftpd.conf.....	70
Figura 53. Configuración del fichero vsftpd.conf.....	70
Figura 54. Ventana principal FileZila.....	71
Figura 55. Circuito impreso en baquelita.....	82
Figura 56. Sistema de control difuso y monitoreo.....	82
Figura 57. Impresión a láser del circuito.....	83
Figura 58. Placa en baquelita.....	83
Figura 59. Placa final.....	84
Figura 60. Prueba de conexión entre el equipo terminal y el microcontrolador.....	85
Figura 61. Prueba de conexión entre el equipo terminal y Raspberry PI 3.....	85
Figura 62. Prueba de conexión entre la Raspberry PI3 y el microcontrolador.....	86
Figura 63. Prueba de envío de datos hacia servidor.....	86
Figura 64. Inicio de sesión.....	87

Figura 65. Ventana principal inicio de sesión.....	88
Figura 66. Resultado día 1, SP Vs. PV Temperatura.....	90
Figura 67. Resultado día 1, SP Vs. PV Humedad.....	90
Figura 68. Resultado día 2, SP Vs. PV Temperatura.....	91
Figura 69. Resultado día 2, SP Vs. PV Humedad.....	91
Figura 70. Resultado día 3, SP Vs. PV Temperatura.....	92
Figura 71. Resultado día 3, SP Vs. PV Humedad.....	92
Figura 72. Resultado día 4, SP Vs. PV Temperatura.....	93
Figura 73. Resultado día 4, SP Vs. PV Humedad.....	93
Figura 74. Resultado día 5, SP Vs. PV Temperatura.....	94
Figura 75. Resultado día 5, SP Vs. PV Humedad.....	94
Figura 76. Resultado día 6, SP Vs. PV Temperatura.....	95
Figura 77. Resultado día 6, SP Vs. PV Humedad.....	95

ÍNDICE DE TABLAS

Tabla 1. Definición de los cuatro métodos de inferencia más populares.....	6
Tabla 2. Comparación entre Apache e IIS.	13
Tabla 3. Comparación entre sistemas de suelos radiantes.	34
Tabla 4. Comparación de sensores de temperatura.	35
Tabla 5. Comparación de sensores de humedad.	36
Tabla 6. Comparación de microcontroladores	36
Tabla 7. Comparacion Raspberry's	37
Tabla 8. Características Raspberry Pi 3 B+	38
Tabla 9. Características Wemos D1 Mini	40
Tabla 10. Características Bomba de agua 800l/h	41
Tabla 11. Datos de producción de los diferentes tipos de naves.....	52
Tabla 12. Temperatura en la Crianza en naves avícolas.	53
Tabla 13. Humedad relativa en el proceso de crianza de aves en naves avícolas.....	53
Tabla 14. Ventilación por depresión en naves avícolas.	54
Tabla 15. Valores máximos y mínimos de variables.	56
Tabla 16. Estructura de los principales estándares de la familia IEEE 802.11	62
Tabla 17. Direccionamiento IP	84
Tabla 18. Presupuesto	89

RESUMEN EJECUTIVO

La evolución de las tecnologías avanzan a pasos agigantados, es por eso que el presente proyecto de investigación describe la implementación de un prototipo de monitoreo y control difuso de granjas avícolas con suelo radiante y energías solar, debido a que al utilizar energías renovables conjuntamente con sistemas de control moderno y aprovechando las tecnologías brindadas al utilizar hardware y software libre se obtuvo ventajas tanto ambientales como monetarias, evidenciando así una mejora en los procesos de producción avícola.

El sistema de monitoreo está basado y diseñado a partir de servidores de distribución gratuita, creando interfaces de acceso y monitoreo acordes al uso que se le quiera dar. Como principales beneficios presentados por este tipo de sistemas es que existe abundante información en la red, además de ello no es necesario el pago de ningún valor por su uso, por lo contrario posee una desventaja la cual se presenta al momento de diseñar el sistema, presentando inconvenientes al momento de codificar las interfaces.

El apartado del control difuso simula el comportamiento de un operador que está a cargo de la planta, teniendo así mediante funciones de membresía que no son más que valores provistos por alguien que sabe de cómo funciona exactamente el proceso. Para concatenar todos estos datos se hace uso de reglas difusas Si-Entonces que al final obtienen un solo valor numérico que fue procesado de acuerdo a la variable a controlar.

Las bondades que brinda el sol y la ubicación geográfica del Ecuador, hace de este recurso una fuente importante de energía renovable que fue utilizado para el proceso de calefacción de ambientes, generando así un entorno idóneo para el proceso de crianza de las aves.

Palabras Clave: lógica difusa, energía solar, monitoreo de procesos.

ABSTRACT

The evolution of technologies is advancing by leaps and bounds, which is why this research project describes the implementation of a prototype monitoring and diffuse control of poultry farms with underfloor heating and solar energy, due to the fact that when using renewable energy with energy systems modern control and taking advantage of the technologies provided when using free hardware and software, both environmental and monetary advantages are obtained, thus demonstrating an improvement in the processes of poultry production.

The monitoring system is based and designed from free distribution servers, creating access and monitoring interfaces according to the use that you want to give it. As main benefits presented by this type of systems is that there is abundant information on the network, in addition to this it is not necessary to pay any value for its use, on the contrary it has a disadvantage which is presented at the time of designing the system, presenting inconveniences when coding interfaces.

The diffuse control section simulates the behavior of an operator who is in charge of the plant, thus having through membership functions that are nothing more than values provided by someone who knows how exactly the process works. To concatenate all this data, we use diffuse rules Yes-So that in the end they obtain a single numerical value that will be processed according to the variable to be controlled.

The benefits offered by the sun and the geographical location of Ecuador, make this resource an important source of renewable energy that will be used for the process of heating environments, thus generating an ideal environment for the process of raising birds.

Keywords: diffuse logic, solar energy, process monitoring.

CAPITULO I

MARCO TEÓRICO

1.1. Antecedentes Investigativos

1.1.1. Contextualización del Problema

Para el presente trabajo fueron consideradas investigaciones relacionadas a sistemas de control difuso, en las diferentes plataformas de investigación, repositorios digitales de Universidades, por lo tanto, se evidencia lo siguiente:

En Argentina, en el año 2018, Sebastián Alfredo Ili Flores, Hisashi Joaquín Konno, Alejandro M. Massafra y Luciano Schiaffino desarrollaron un “Control Simultáneo de Lógica Difusa de Humedad y Temperatura en Incubadoras Neonatales” teniendo como objetivo principal de este trabajo el diseñar una estrategia de control de lógica difusa para una regulación adecuada de la temperatura y la humedad activa para una incubadora neonatal. Para el sistema de control de humedad y temperatura, se establecieron 90 reglas de decisión de lógica difusa en un sistema tipo Mamdani que se implementó en un microcontrolador ARM Cortex M4 de 32 bits. [1]

El tema de investigación “Diseño y Simulación de un Sistema de Control de Temperatura e Iluminación para la Crianza de Pollos en Haciendas Usando una Red de Sensores” desarrollada por Andrés Cabrera en el año 2017, realizó un sistema de control que a través de una red inalámbrica conformada por módulos Xbee en conjunto con sensores de temperatura obtener valores reales para llevar un correcto control de la crianza de pollos en haciendas, además de poder controlar remotamente el encendido y apagado de la iluminación que requieren las aves en su etapa de desarrollo. [2]

En el año 2017, Henry Villacís realizó un “Diseño de los Sistemas de Automatización para la Ampliación de una Granja Avícola” que permitió la automatización de los principales procesos que intervienen en la crianza de las aves, presentando una propuesta que, cumpliendo con las normas establecidas por los organismos de control

y vigilancia, genere una importante reducción de costos, además presenta sistemas automatizados de ventilación, iluminación exterior e interior y apertura y cierre de cortinas realizado en el software de control LabVIEW, el mismo que utiliza programación gráfica que ofrece al usuario un entorno amigable para su uso. [3]

El tema de investigación “Controlador de Lógica Difusa de Temperatura y Humedad Dentro de un Invernadero Agrícola” desarrollado por Rim BEN ALI, Emna ARIDHI, Mehdi ABBES y Abdelkader MAMI en el año 2016 en Túnez, permitió desarrollar un modelo dinámico térmico para describir el ambiente del invernadero y para predecir la temperatura interior y la humedad relativa. El invernadero agrícola se modeló utilizando el entorno Matlab-Simulink. Los resultados de la simulación mostraron la efectividad de controlador para lograr un clima interior favorable. [4]

En el año 2015, Jean Claude Mugisha, Bernard Munyazikwiye y Hamid Reza Karimi desarrollaron un “Diseño de Sistema de Control de Temperatura Utilizando PID Convencional y Controlador Inteligente de Lógica Difusa” en Taiwán, en el que presentaron el diseño de control de temperatura para hornos industriales de tratamiento térmico mediante el uso de controladores PID y lógica inteligente difusa. Encontrando que la respuesta de salida de Fuzzy Logic es muy precisa en términos de sobreimpulso y error de estado estable en comparación con la de PID. Además, ambos controladores de temperatura se modelaron y simularon utilizando el software MATLAB. [5]

1.2. Fundamentación Teórica

1.2.1. La Industria de la Avicultura en el Ecuador

La avicultura abarca un campo fundamental para el sustento y desarrollo socio-económico del país, tal como la agricultura, compuesta por una gran cadena de producción brindando abasto para el alimento de las aves, siendo el maíz, la soya, y los balanceados en general; además abarca el sector del transporte que brinda su servicio para el traslado del producto hacia distintos lugares. La mano de obra que se utiliza en este campo también está involucrada directa e indirectamente, ya que se mejora la calidad y estilo de vida de los habitantes aledaños al sector de producción. [6]

El consumo de productos derivados de naves avícolas procedente de criaderos, es uno de los principales alimentos que componen la canasta básica de la población

ecuatoriana, ya que es accesible para todos los niveles socio-económicos del país, en específico a los sectores con menos poder adquisitivo. La producción avícola se extiende a todas las regiones del país, registrándose naves productoras en todas las provincias del Ecuador, motivo por el cual la producción no se ve interrumpida en todas las épocas del año. Por lo general el periodo de inicio a fin de crianza es de alrededor de 42 días con un peso final promedio de 2.2 Kg. 12 [6]

Según la Industria Avícola Digital, para el Ecuador en el año 2012 el consumo por cada persona fue de 27 Kg de pollo, ocupando así el lugar número siete en América Latina; tomando datos de la Corporación Nacional de Avicultores, el 14% del Producto Interno Bruto Agropecuario está representado por aproximadamente 1600 avicultores que generan 222 millones de aves, representando un continuo crecimiento desde los años 90's. [7]

En el Ecuador desde el año 2015 al 2019 según la Encuesta de Superficie y Producción Agropecuaria ESPAC se presenta diversidad de indicadores que se relacionan con la producción de la avicultura en el país, evidenciando así técnicamente el progreso y el incremento de la demanda de este tipo de productos a nivel del país, causa principal para el incremento de este sector económicamente. [8]

En el siguiente gráfico se muestra la evolución en la producción nacional de aves y huevos dentro de los años 2015-2019

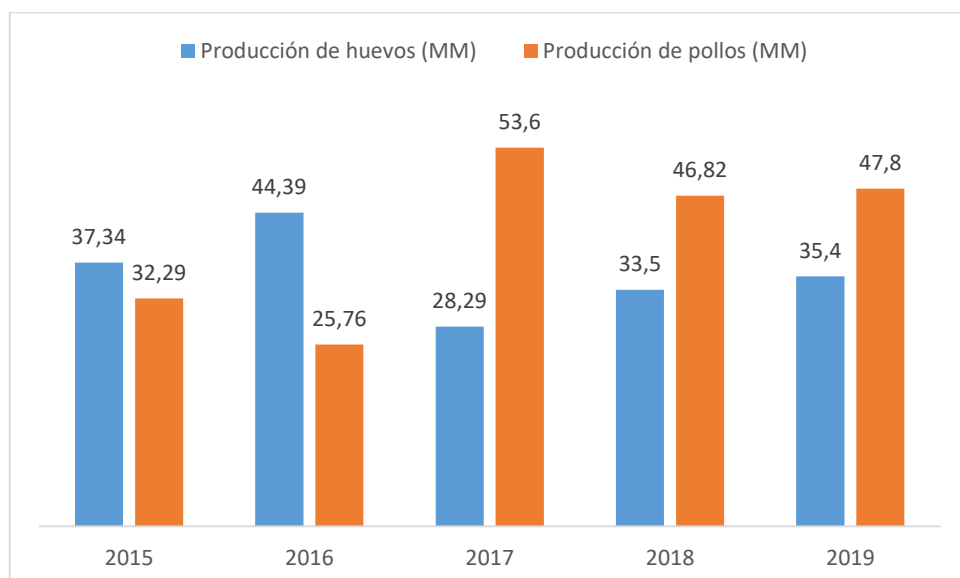


Figura 1. Encuesta de Superficie y Producción Agropecuaria ESPAC [8]

1.2.2. Controlador de Lógica Difusa

El proceso de un sistema difuso tiene tres pasos. Fusificación, Evaluación de Reglas y Defusificación. En la fusificación, los valores nítidos de entrada se transforman en grados de pertenencia en los conjuntos difusos. El grado de pertenencia de cada valor nítido en cada conjunto difuso se determina conectando el valor en la función de pertenencia asociada con el conjunto difuso. En el paso de evaluación de la regla, a cada regla difusa se le asigna un valor de fuerza. La fuerza está determinada por los grados de pertenencia de los valores de entrada nítidos en los conjuntos difusos de la parte antecedente de la regla difusa. La etapa de defusificación transpone las salidas difusas en valores nítidos. En la siguiente ilustración se observa claramente el proceso de un sistema de control difuso: [5]

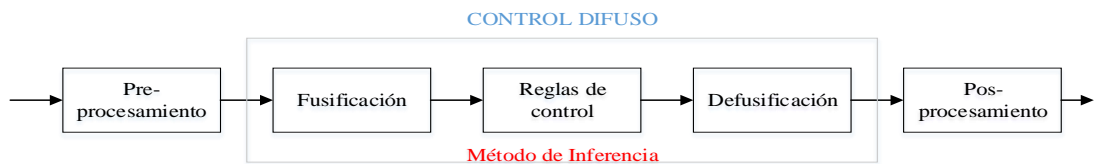


Figura 2. Controlador difuso. [5]

Conjuntos borrosos

Los conjuntos borrosos son una extensión de los conjuntos clásicos, donde se añade una función de pertenencia, definida esta como un número real entre 0 y 1, así se introduce el concepto de conjunto o subconjunto borroso y se lo asocia a un determinado valor lingüístico, definido por una palabra o etiqueta lingüística, donde esta es el nombre del conjunto o subconjunto. Por cada conjunto se define una función de pertenencia o membresía denominada $\mu_{A(x)}$, indica el grado en que la variable x está incluida en el concepto representado por la etiqueta A ($0 \leq \mu_{A(x)} \leq 1$), si esta función toma el valor 0 significa que tal valor de x está absolutamente incluido en A . En la ilustración 2 se puede apreciar un ejemplo donde el conjunto temperatura (con variable x) está subdividido en 3 subconjuntos $\{Baja, Media, Alta\}$, con sus respectivas funciones de membresía $\{\mu_{Baja(x)}, \mu_{Media(x)}, \mu_{Alta(x)}\}$. [9]

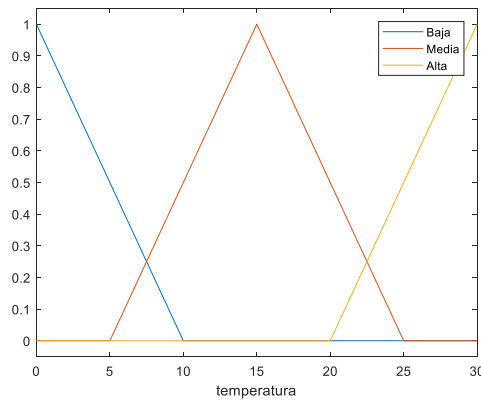


Figura 3. Ejemplo de subconjuntos borrosos para el conjunto temperatura.

Elaborado por: el investigador.

Sea X una colección de objetos, expresados en forma genérica por x . Entonces, un conjunto difuso A en X , se define como un conjunto de pares ordenados.

$$A = \{(x, \mu_{A(x)}) / x \in X\}$$

Donde $\mu_{A(x)}$ es una función de pertenencia cuya etiqueta es A y su dominio es x . [9]

Fusificación

El control difuso siempre involucra este proceso, esta operación es realizada en todo instante de tiempo, es la puerta de entrada al sistema de inferencia difusa. Es un procedimiento matemático en el que se convierte un elemento del universo de discurso (variable medida del proceso) en un valor en cada función de membresía a las cuales pertenece. [9]

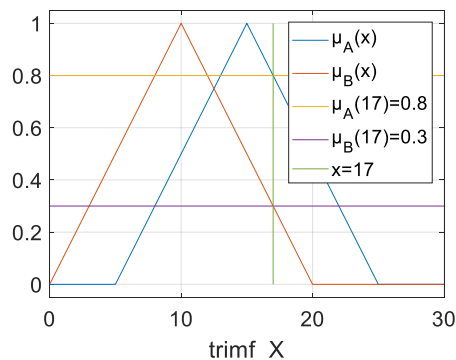


Figura 4. Ejemplo de fusificación de una variable.

Elaborado por: el investigador.

Para comprender mejor se analiza la ilustración 3 que proporciona los siguientes resultados: $\mu_A(17) = 0.8, \mu_B(17) = 0.3$.

El valor de temperatura igual a 17 pertenece a dos conjuntos con distintos grados en cada uno.

Inferencia borrosa

Las reglas difusas representan el conocimiento y la estrategia de control, pero cuando se asigna información específica a las variables de entrada en el antecedente, la inferencia difusa es necesaria para calcular el resultado de las variables de salida del consecuente, este resultado es en términos difusos, es decir que se obtiene un conjunto difuso de salida de cada regla, que posteriormente junto con las demás salidas de reglas se obtuvo la salida del sistema. [9]

Existe una gran cantidad de métodos de inferencia difusa, pero hay dos que generan mejores resultados en el campo del control mostrados a continuación en la tabla 4, estos son inferencia de Mamdani por mínimos (Mamdani minimum inference), R_M y la inferencia Sugeno Takagi, R_{BP} . [9]

Tabla 1. Definición de los cuatro métodos de inferencia más populares. [9]

Método de inferencia	Definición
Mamdani minimum inference, R_M	$\min(\mu, \mu_W(z)), \forall z$
Sugeno Takagi inference, R_{BP}	$\max(\mu + \mu_W(z) - 1, 0)$

Donde μ_W es la función de pertenencia del conjunto de salida w.

Defusificación

La defusificación es un proceso matemático usado para convertir un conjunto difuso en un número real. El sistema de inferencia difusa obtiene una conclusión a partir de la información de la entrada, pero es en términos difusos. Esta conclusión o salida difusa es obtenida por la etapa de inferencia borrosa, esta genera un conjunto borroso pero el dato de salida del sistema debe ser un número real y debe ser representativo de todo el conjunto obtenido en la etapa de agregado, es por eso que existen distintos

métodos que arrojan resultados distintos, los mismos que se muestran a continuación:
[9]

- **Defusificación por centro de gravedad:** también conocido como centroide, está dado por la siguiente representación matemática:

$$y_d = \frac{\int_S y \mu_Y(y) dy}{\int_S \mu_Y(y) dy}$$

Donde μ_Y es la función de pertenencia del conjunto de salida Y, cuya variable de salida es y. S es el dominio o rango de integración.

- **Defusificación por dentro de área (COA):** también llamado de altura, el centro de gravedad es aproximado por el centro de gravedad de un arreglo de masas puntuales, las cuales son el centro de gravedad de cada conjunto de salida correspondiente a cada regla, con masa igual al grado de pertenencia en ese punto de su centro de gravedad. Si se llama δ_l al centro de gravedad del conjunto difuso de salida B_l de la l-ésima regla, el centro de gravedad queda determinado por:

$$y_d = \frac{\sum_{l=1}^R \delta_l \mu_{B_l}(\delta_l)}{\sum_{l=1}^R \mu_{B_l}(\delta_l)}$$

Donde R es el número de reglas.

- **Criterio de máximo (MC):** la salida es aquella para la cual la función de membresía alcanza su máximo valor.
- **El método de centro de área (COA) o de centro de gravedad (COG):** la salida responde al método uno y método dos.
- **El método de la media de máximo:** (MOM, middle of maximum). La salida es el mínimo valor de todos aquellos que generan el valor más alto de la función de membresía.
- **El método del máximo más chico:** (SOM, smallest of maximum). La salida es el mínimo valor de todos aquellos que generan el valor más alto de la función de membresía.
- **El método del máximo más grande:** (LOM, largest of maximum). La salida es el máximo valor de todos aquellos que generan el valor más alto de la función de membresía.

- **Bisector de área:** la salida es el valor que separa el área bajo la curva en dos sub-áreas iguales.

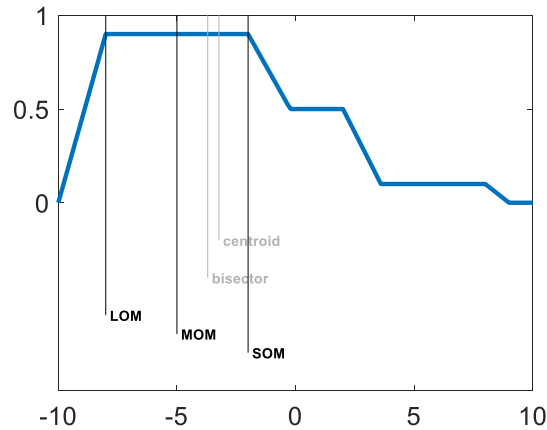


Figura 5. Representación gráfica de los distintos métodos de defusificación

Elaborado por: el investigador.

1.2.3. Monitoreo de procesos

Es el hardware y software a través del cual un operario interactúa dinámicamente con un controlador. También llamado Interface Hombre-Máquina (HMI por sus siglas en inglés). Un HMI puede mostrar un panel de control físico con botones y luces indicadoras hasta una computadora industrial con gráficos a colores corriendo para mostrar procesos y controlarlos a través de estas. [10]

Los sistemas HMI se utilizan objetivamente como una herramienta de supervisión y control. Entre los objetivos más importantes se puede encontrar [11]:

- **Economía:** brinda la facilidad de ver qué está ocurriendo en la planta desde la oficina que enviar a un operador a hacer dicha tarea, permitiendo ahorrar así recursos económicos.
- **Accesibilidad:** por ejemplo en una estación eólica completamente la tenemos visualizada desde una mesa de trabajo, permitiendo modificar parámetros y variables de operación de los generadores, pudiendo así dejar fuera de funcionamiento cualquiera de ellos.
- **Mantenimiento:** con el proceso de obtención de datos, se pudo visualizar anomalías presentes en la planta, si n que necesariamente haya presencia de

operarios en el sitio, pudiendo así llegar a controlar directamente la falla en el lugar luego de ser detectada.

- **Ergonomía:** es la técnica que permite que la relación entre la máquina y el usuario sea lo más amigable y entendible posible.
- **Gestión:** permite sacar el mayor rendimiento posible del sistema al recopilar datos para posteriormente ser evaluados de diversas maneras acorde a las exigencias de la planta.
- **Flexibilidad:** provee un ahorro en tiempo y recursos ya que no fue necesario como por ejemplo un cableado externo para cambiar algún parámetro de visualización en la operación.
- **Conectividad:** la gran ventaja de los sistemas HMI es que pueden ser puestos en operación mediante diversidad de protocolos acordes al requerimiento de cada proceso, lo cual permite la interconexión de diferentes proveedores.

Sistemas de monitoreo remoto

Los sistemas de monitoreo remoto son aquellos que envían y reciben datos o indicaciones a distancia por medio de un sistema de telecomunicaciones. La palabra remoto es un término que no implica sólo el concepto de actuación sino que, también, le añade la idea de monitorización o presentación de datos en el centro de control del sistema, lo que implica la adquisición de datos del campo por las estaciones remotas.

[12]

Características de un sistema de monitoreo

Las características principales de los sistemas de monitoreo son las siguientes: [13]

- Robustez de los elementos a instalarse.
- Seguridad de los sistemas de comunicaciones y procesos de funcionamiento del sistema.
- Anti vandalismo.
- Autonomía.
- Mantenimiento sencillo y económico.
- Escalabilidad.
- Uso de estándares comerciales.

Estructura básica de un sistema de monitoreo

La estructura básica de un sistema de monitoreo puede estar gobernada por un conjunto de centros de control jerarquizado o en paralelo que comparten información o se traspasa entre ellos. La estructura básica puede describirse en términos de centro de monitoreo, unidad de transmisión remota y los medios de comunicación como el esquema mostrado en la siguiente figura: [12]

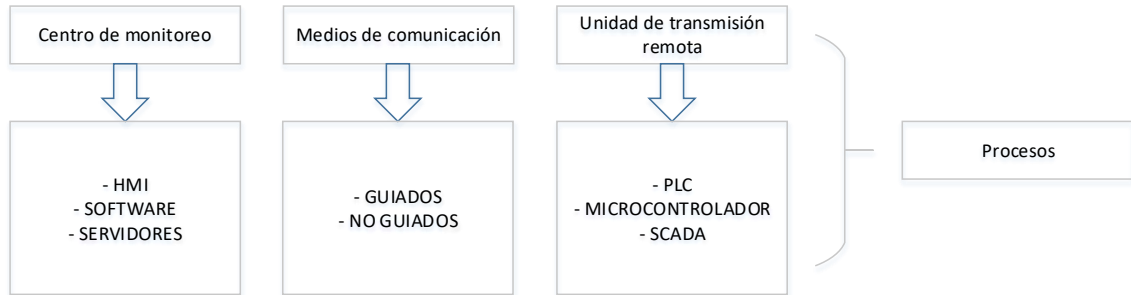


Figura 6. Estructura de un sistema de monitoreo. [12]

Centro de monitoreo

El Centro de Control es el conjunto de dispositivos, módulos funcionales e interfaces con los canales de comunicación que permitan llevar a cabo correctamente las funciones de supervisión y monitorización del estado actual de la red, que puede obtenerse mediante la programación de microcontroladores, PLC'S, servidores, ordenadores entre otros. [12]

Unidad de Transmisión Remota

Es un dispositivo electrónico que permite controlar un determinado número de entradas/salidas y enviarlas a un sistema de control superior, generalmente un PLC, o bien directamente a un SCADA. Las funciones de una RTU pueden ir entre la recolección de información o controlar el estatus de las aplicaciones conectadas al sistema en base a los datos de entrada. Las RTU's generalmente están localizadas cerca de los procesos monitoreados y transfieren información a los centros de control; están diseñadas para operar en forma segura en ambientes hostiles, protegidas de la erosión, humedad, polvo y de otros contaminantes atmosféricos. [14]

Sistema de Comunicación

Un sistema de Comunicación permite el intercambio de información entre un punto llamado emisor y otro llamado receptor. La calidad y velocidad de la transmisión depende de varios factores entre los cuales se encuentran el número de estaciones remotas, la cantidad de datos que procesa u obtiene, la localización geográfica de cada estación, los servicios y técnicas de comunicación que se manejen. Para poder asegurar la transmisión de información es necesario determinar el tipo de hardware y software que se va a utilizar en el sistema, con el fin de evitar el ruido que suele aparecer en los canales de comunicación. [12]

1.2.4. Servidores

Un servidor es un equipo informático robusto que es utilizado en la red para proporcionar servicios a computadores llamados clientes quienes hacen uso de protocolos de comunicación para que se dé la transacción de información. Deben tener gran capacidad de soportar peticiones de diferentes clientes y emitir una respuesta en un tiempo aceptable. Existen diversos servidores dependiendo su funcionalidad entre ellos se encuentran: [15]

- **Servidores locales:** Los servidores locales son aquellos que ofrecen un servicio dentro de una red local.
- **Servidores globales:** Los servidores globales son aquellos que prestan servicios a través de Internet.
- **Servidor de aplicaciones:** Los servidores de aplicaciones comparte programas a los usuarios de esa red o entre grupo de usuarios.
- **Servidor de impresión:** Los servidores de impresión sirven para compartir la impresora como recurso.
- **Servidor de ficheros:** Los servidores de ficheros como su nombre lo indica comparte ficheros entre grupos o todos los usuarios, cada uno puede tener su espacio de almacenamiento en el servidor.
- **Servidor de correo:** El servidor de correo es aquel que envía y recibe mensajes.
- **Servidor Web:** El servidor web es aquel que almacena un programa que implementa el protocolo HTTP Hyper Text Transfer Protocol. Trabaja en función

a las peticiones que le llegan desde un ordenador a través de un navegador de Internet.

Características de los principales componentes físicos de un servidor

Las características de los principales componentes físicos de un servidor son las siguientes: [15]

- **Memoria RAM:** La memoria RAM (Random Access Memory) es la encargada de almacenar las instrucciones que pone en ejecución la unidad de procesamiento. Como el servidor es el encargado de escuchar las peticiones de varios usuarios, es importante que la memoria RAM, sea lo suficientemente rápida para mejorar el tiempo de respuesta del servidor.
- **Unidad Central de Proceso:** La unidad central de proceso (CPU) es el cerebro del computador, el CPU de un servidor no tiene recargo extraordinario, debido a que no se realizan grandes operaciones de cálculo u operaciones que requieran gran cantidad de procesamiento, por lo tanto el CPU debe ser de características normales para un ordenador.
- **Placa base:** La placa base es la parte del ordenador que comunica a todos sus componentes, por lo que es importante su calidad para el correcto funcionamiento del servidor.
- **Placa de comunicaciones:** La placa de comunicaciones es el punto de conexión entre el computador e Internet, determina la cantidad de información que puede procesar el servidor. Es recomendable usar una placa de 10 o 100 Mbps a un conmutador de 100Mb y si es posible utilizar medios de transmisión sin pérdidas de energía como la fibra óptica para mejorar la rapidez de la red de comunicación.

Servidor Web

Un servidor web se utiliza para dar acceso a clientes a información contenida en base de datos, aplicaciones web, etc., sin embargo también se puede utilizar para probar aplicaciones web sin la necesidad de subirlas al internet. Utiliza el protocolo HTTP para la comunicación. Su diagrama esquemático se observa en la siguiente figura: [16]

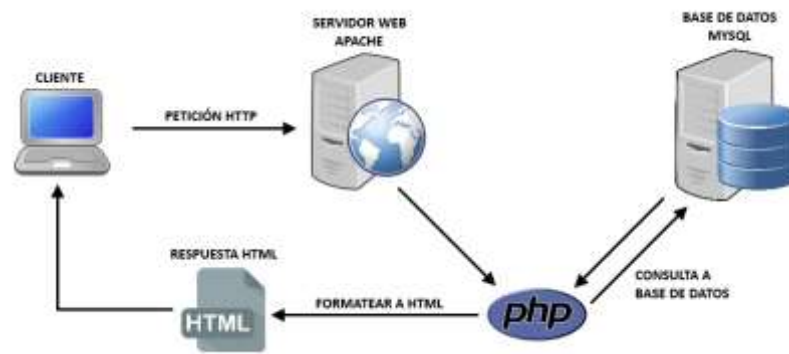


Figura 7. Estructura de un servidor web. [16]

Existen varios servidores web entre los más destacados se encuentran: El servidor Apache que es de código abierto y es uno de los más utilizados, y el servidor Internet Information Services (IIS) de Microsoft que es un servidor desarrollado exclusivamente para Windows. [17]

En la siguiente tabla se puede observar una comparación entre un Servidor Apache y el servidor IIS de software propietario.

Tabla 2. Comparación entre Apache e IIS. [17]

APACHE	IIS
Software libre	Software propietario (licencia)
Modular	Modular
Utilizado en servidores raíz de internet	Usado en ambientes corporativos

Servidor APACHE

El servidor APACHE es un servidor Web de software libre, que tiene las siguientes características: [16]

- Estable: Es estable debido a la profunda robustez que impide caídas o cambios en el servicio.
- Flexible: Es flexible debido a que trabaja con estándares HTTP y con varias extensiones web como PHP, SSL, etc.
- Extensible: Es extensible porque dispone de módulos que amplían sus servicios.
- Multiplataforma: Es multiplataforma porque está disponible para varios sistemas operativos.

Protocolo de comunicación Hyper Text Transfer Protocol

El protocolo Hyper Text Transfer Protocol (HTTP) es un protocolo de comunicación utilizado en las transacciones entre los elementos de la arquitectura Web, es decir, clientes, servidores, proxies y demás. La manera de comunicarse se basa en peticiones por parte del cliente a un navegador, los llamados recursos son la información requerida e identificada mediante una URL. [18]

El puerto que se utiliza generalmente es el 80 sobre el protocolo TCP. El cliente envía la información de la siguiente manera: [18]

- a. Envía el método de petición GET o POST
- b. El documento que desea recibir
- c. La versión de HTTP que se va a utilizar en la comunicación
- d. El tipo de datos que se recibirán.

Métodos de petición más utilizados

- **GET:** El método GET devuelve un fichero, es decir, los campos de un formulario son incorporados a la dirección. No es recomendable si se envía datos delicados como contraseñas o cuentas privadas.
- **POST:** El método POST regresa los datos a un servidor. Es el método apropiado para enviar datos a un formulario con diversos campos. [16]

Protocolo TCP/IP

El protocolo TCP/IP es el protocolo más básico de Internet, suministra todos los servicios necesarios para el transporte de datos. Los protocolos son un tipo de software que actúan en la red para poder realizar la comunicación de datos. [19]

Protocolo IP

El protocolo IP es un protocolo no orientado a la conexión, se utiliza en el emisor y el receptor de la red, se realiza a través de una red de paquetes. El protocolo IP procesa los paquetes de datos de manera independiente al definir su representación, ruta y envío. [19]

Servidores de aplicaciones

El servidor de aplicaciones es el encargado de dar acceso a los usuarios a recursos necesario para obtener información de una base de datos, seguridad, mantenimiento y demás. Se basa en un sistema de tres capas. [15]

- **Capa 1:** La capa 1 es la que interactúa con el usuario, y funciona gracias a navegadores gráficos.
- **Capa 2:** La capa 2 es el servidor de aplicaciones de la red local
- **Capa 3:** La capa 3 es el servidor de base de datos. El servidor de aplicaciones interactúa con el servidor Web para ofrecer la respuesta que desea obtener el usuario y se lo llama Servidor de aplicaciones Web

Servidor de base de datos

El servidor base de datos es una aplicación instalada en un ordenador que sirve para almacenar y compartir archivos con otros usuarios de la red. Autoriza el acceso remoto a todos los archivos almacenados en él, que pueden ser documentos de texto, hojas de cálculo, imágenes, videos, etc. [19]

Pueden compartir información con muchos clientes, que pueden ser usuarios de un navegador web hasta servidores de aplicaciones, provee gran seguridad y servicios independientemente de la plataforma que se utilice. [19]

1.2.5. Energías Renovables

Se denomina energía renovable a la que se obtiene de fuentes naturales virtualmente inagotables, unas por la inmensa cantidad de energía que contienen, y otras porque son capaces de regenerarse por medios naturales. Durante los últimos años, debido al incremento del coste de los combustibles fósiles y los problemas medioambientales derivados de su explotación, estamos asistiendo a un renacer de las energías renovables. [20]

Tipos de energías renovables

Existen diferentes métodos para aprovechar esta energía, dependiendo del recurso natural que aprovechen: [21]

Eólica

Transforman la energía del viento en energía útil. Los aparatos que se emplean para transformar la energía del viento en energía útil se denominan aerogeneradores. Estos aerogeneradores constan, principalmente de un rotor; encargado de transformar la energía del viento en energía mecánica. Las palas, que se encargan de captar el viento incidente. El multiplicador; que adapta las bajas velocidades de giro del rotor a las necesariamente más altas del generador. El generador; que transforma la energía mecánica procedente del rotor en energía eléctrica. [21]

Cuando hay vientos más fuertes, se hace necesario gastar parte del exceso del viento para evitar daños en el aerogenerador. Por tanto, todos los aerogeneradores están diseñados con algún tipo de control de potencia. La potencia producida por los aerogeneradores depende del área de rotación del rotor y de la velocidad del viento. [21]

Hidráulica

Aprovechan la energía potencial de un curso de agua, mediante una central hidroeléctrica que la transforma en energía eléctrica. [21]

Como se sabe, el aprovechamiento de la energía hidráulica se lleva a cabo en las centrales hidroeléctricas. Una central hidroeléctrica consta de: Embalse; sirve para acumular el agua del río. Presa; se encarga de retener el agua del embalse. Desde la presa, mediante una tubería forzada, se enlaza el embalse con la sala de máquina. Central hidroeléctrica; en esta instalación se encuentran las turbinas (transforman la energía cinética del agua en energía de rotación), generador-alternador (dispositivo unido a la turbina que convierte la energía de rotación en energía eléctrica) y el transformador (transforma la energía que produce el generador en corriente para poder ser transportada). [21]

Solar

Se aprovecha la radiación solar constante durante todo el año, pero intermitente por el día y la noche. Se necesita un sistema de captación y almacenamiento. [21]

La utilización de la energía solar puede ir encaminada hacia diversos objetivos: [21]

- Energía solar pasiva; aprovecha la energía solar sin elementos mecánicos.
- Energía solar térmica; concentran el efecto térmico de la radiación solar sobre un fluido, para producir calor.
- Energía solar fotovoltaica; se basa en el efecto fotovoltaico que se produce al incidir la luz sobre materiales semiconductores, para producir energía eléctrica.

Geotérmica

Es la energía que se almacena en el interior de la tierra en forma de calor. Tiene su origen en una serie de reacciones químicas naturales. El uso que se hace de la energía geotérmica se puede dividir en tres categorías, según la temperatura de los recursos disponibles: [21]

- Alta temperatura ($>150^{\circ}\text{C}$): Se utiliza principalmente para la producción de energía eléctrica en centrales.
- Media temperatura ($50\text{-}150^{\circ}\text{C}$): Es ideal para una utilización directa del calor almacenado.
- Baja temperatura ($<50^{\circ}\text{C}$): Se utiliza conjuntamente con bombas de calor geotérmicas.

Biomasa

Se considera como biomasa, el aprovechamiento térmico o eléctrico del conjunto de materia orgánica de origen vegetal o animal (incluyendo los materiales procedentes de su transformación natural o artificial). [21]

La energía de la biomasa corresponde entonces a toda aquella energía que puede obtenerse de ella, bien sea a través de su quema directa o su procesamiento para conseguir otro tipo de combustible. [21]

Los usos de la biomasa en aplicaciones energéticas son principalmente la producción de gas, energía calórica (térmica) y energía eléctrica. A continuación se expone un esquema donde podemos ver la clasificación de las energías renovables:

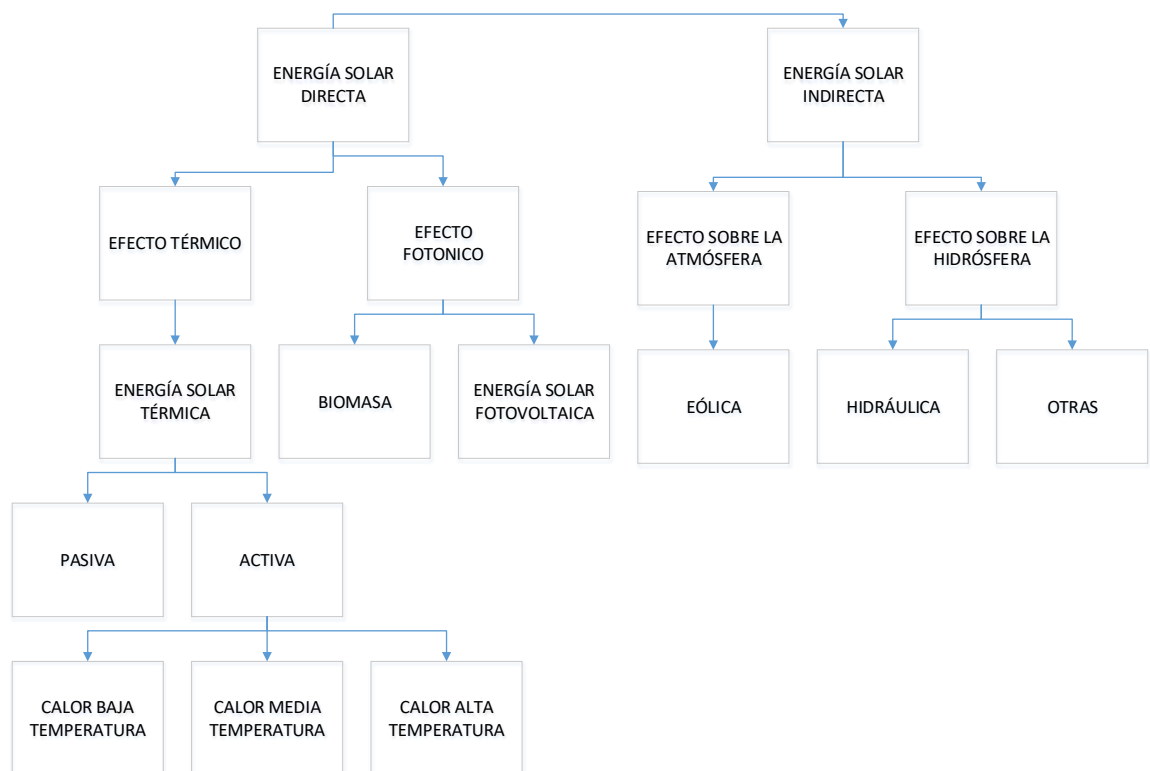


Figura 8. Clasificación de las energías renovables. [22]

Descripción de la energía solar

Durante un año, la energía que desprende el Sol sobre la Tierra es cuatro mil veces mayor que la energía que se consume por todos los habitantes de la Tierra. [20]

Está claro que la energía que recibimos del Sol es inagotable, por lo que es importante que la sepamos aprovechar de cara al futuro, aunque hay que tener en cuenta que esta energía está sometida a fluctuaciones más o menos bruscas, siendo en los periodos donde más se necesita de esta energía (invierno), cuando disponemos de ella en menor medida. Es importante continuar con el desarrollo de la tecnología de captación, acumulación y distribución de la energía solar, para conseguir las condiciones que la hagan competitiva. [20]

La energía que se recibe del Sol se puede aprovechar de forma activa o pasiva. El aprovechamiento de forma pasiva es aquella que capta la energía solar, la almacena y la distribuye de forma natural, sin necesidad de ningún elemento mecánico, utilizando procedimientos de ventilación natural. [20]

Su funcionamiento está basado en las características y la disposición de los materiales utilizados en la construcción del edificio. Cuando se construye un edificio y se diseña un sistema de aprovechamiento de la energía solar de forma pasiva, se tiene en cuenta la orientación del edificio, colocar elementos de protección (aislamientos, barbacanas, persianas, etc) o la dirección de los vientos dominantes, entre otros factores. [20]

El aprovechamiento de forma activa consiste en utilizar la energía de la radiación para calentar un fluido. Su funcionamiento se basa en la captación de la energía solar mediante captadores, el almacenamiento de esta energía y su posterior distribución como se muestra en la figura a continuación.

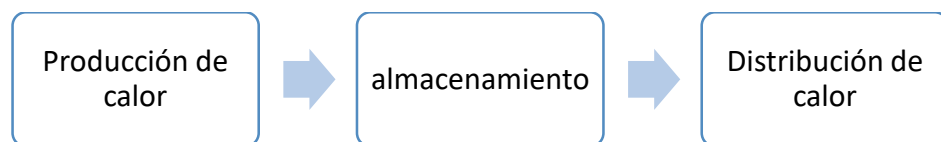


Figura 9. Esquema de funcionamiento de una instalación de energía solar activa

Elaborado por: el investigador

La energía solar activa se puede clasificar en: energía solar activa de baja temperatura; alcanza temperaturas de unos 80°C, la energía solar activa de media temperatura; que puede alcanzar temperaturas hasta 120°C, para ello se emplean captadores solares de vacío o de concentración, y la energía solar activa de alta temperatura; basada en la producción de energía eléctrica, utiliza unos sistemas de concentración de la energía en un solo punto para conseguir temperaturas más altas (2000°C). [20]

La aplicación de la energía solar se centra en la producción de calor o electricidad. El calor se consigue con captadores o colectores térmicos, y la electricidad gracias a los módulos fotovoltaicos. Los dos procesos son muy diferentes entre sí, y no tienen nada que ver ni en lo técnico ni en su aplicación. [20]

Hablamos primero de los sistemas de aprovechamiento térmico. El calor recogido en los colectores puede destinarse a satisfacer numerosas necesidades. Por ejemplo, se puede obtener agua caliente para consumo doméstico o industrial, o bien para dar calefacción a nuestros hogares, hoteles, colegios, fábricas, etc. Incluso podemos climatizar las piscinas y permitir el baño durante gran parte del año. [20]

A continuación, nos centramos en las principales aplicaciones de este sistema, que son la producción del agua caliente sanitaria (ACS) y calefacción.

Agua Caliente Sanitaria.

La producción de agua caliente sanitaria, ACS, es la aplicación más popular de la energía solar térmica, dado su buen rendimiento. El tratarse de un consumo continuado a lo largo del año y a una temperatura media (38°C-40°C), la producción de ACS es perfectamente compatible con el aprovechamiento de la radiación solar. [23]

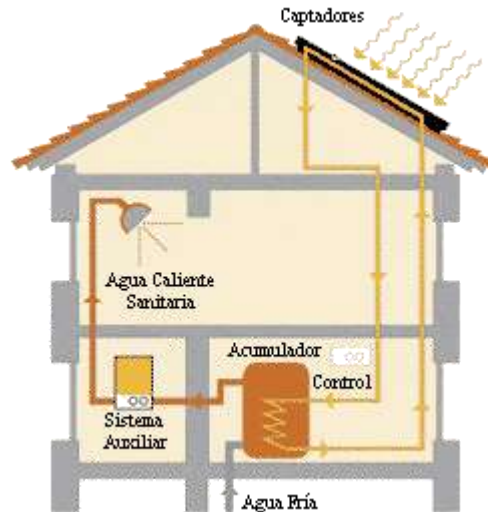


Figura 10. Esquema de funcionamiento ACS. [23]

La tecnología más utilizada para este uso es el colector solar plano o captador de placa plana, formado por un absorbedor de tubos, dentro de una caja de cerrada con una cubierta de cristal que produce el efecto invernadero en su interior. [23]

Estos captadores pueden llegar a temperaturas de 70 o 80°C con facilidad, trabajando a temperaturas mayores que las ambientales. La energía conseguida por los captadores, se puede guardar en acumuladores de agua que luego son suministrados al consumo. Un buen aislamiento del acumulador, así como de los conductos permite conservar el calor durante la noche, pudiendo utilizar el agua caliente aun sin Sol. [23]

Actualmente, se instalan captadores solares en multitud de aplicaciones: casas unifamiliares, bloques comunitarios, centros deportivos, edificios sanitarios, etc. [23]

La producción de agua caliente para calefacción se está convirtiendo también en una forma usual de aprovechar la energía del Sol. Pero en este caso, hay que tener en cuenta un factor muy importante: el sistema utilizado para transmitir el calor al ambiente.

A continuación se explica, de menos a más eficientes, sistemas de calefacción. [23]

Radiadores

Los radiadores son sistemas de calor por radiación que trabajan a altas temperaturas (80°C). Contrario a la energía solar que trabaja a bajas temperaturas (45°C). Por este motivo la calefacción por radiadores es un sistema absolutamente inapropiado para la energía solar térmica. [23]

En caso que se quiera aplicar la energía solar en una instalación ya hecha de radiadores, puede conseguirse pero sería una instalación poco viable y compleja. [23]

Fancoils

Un fancoil es un elemento emisor con un intercambiador de calor al que se le lleva agua caliente o fría y al hacer circular aire mediante un ventilador es capaz de devolver al ambiente el aire calentado o enfriado, tiene la ventaja de poder trabajar con bajas temperaturas en calefacción y poder utilizarlo, tanto para calentar como para enfriar, dependiendo de la temperatura de entrada del agua. [23]

El mayor inconveniente es que la climatización se realiza con movimiento impuesto del aire, lo que influye negativamente en la sensación térmica y reseca el ambiente. La mayor ventaja que tiene es la gran velocidad con la que acondiciona una estancia. Al igual que el suelo radiante, el hecho de trabajar a baja temperatura permite utilizar cualquier fuente de energía para su funcionamiento: gas natural, gasoil, bomba de calor y especialmente energías renovables como la solar, donde su aprovechamiento es óptimo. [23]

1.2.6. Suelo Radiante

La calefacción por suelo radiante es el sistema que proporciona un mayor nivel de confort dentro de una vivienda o local. Consiste en una red de tuberías enterradas en el suelo por las que circula agua, normalmente entre 35 y 45 °C, dependiendo del tipo de suelo. Entre las ventajas de este sistema destacan: [22]

- La distribución de temperaturas que se consigue en la sala, manteniendo una mayor temperatura a nivel del suelo y descendiendo ésta con la altura. De esta manera no se desaprovecha energía calentando innecesariamente la parte alta de la estancia. Esto significa que el suelo radiante también es un sistema muy adecuado para el calentamiento de locales altos.

- El hecho de trabajar a baja temperatura permite utilizar cualquier fuente de energía para su funcionamiento: gas natural, gasoil, bomba de calor y especialmente energías renovables como la solar, donde su aprovechamiento es óptimo.

Cuando se trabaja con energía solar para obtener calefacción, el suelo radiante es óptimo para este fin. La principal dificultad que presenta el suelo radiante es que sólo es aplicable en viviendas o locales en construcción o donde se pretenda realizar una reforma importante, ya que las tuberías necesarias para su funcionamiento abarcan todo el suelo de las estancias a calefactar. [22]

Funcionamiento

Este tipo de calefacción por agua caliente a baja temperatura está basado en el impulso a través de tuberías empotradas en el suelo permitiendo que el calor que se emite por estas sea irradiado directamente hacia el piso y posteriormente es emitida al ambiente en forma de ‘energía radiante’, tratándose así de un sistema con inercia térmica, debido a que los circuitos emisores salen desde los colectores de emisión y retorno. [24]

Propiedades

Desde la perspectiva del confort, se percibe en el tema de calefacción una diferencia pequeña tal que la temperatura entre los pies y la cabeza sea mínima y este es el principio básico de funcionamiento del suelo radiante. Debido a este proceso el calor se distribuye uniformemente, debido a que el piso es el que distribuye el calor por el ambiente. [24]

La velocidad de movimiento del aire caliente es muy escasa, por el motivo de que la diferencia de temperatura del aire es relativamente pequeña. Las escasas presencias de corrientes de aire son otro factor que ahorra mayor energía, debido a que estas corrientes no llevan el calor a zonas no necesarias de calefacción. El agua caliente se puede obtener desde diversas fuentes de energía, especialmente de colectores solares térmicos. [24]

Componentes

En el Suelo Radiante son importantes algunos componentes adicionales para que éste funcione y se pueda controlar de manera adecuada. Los componentes más importantes

del Suelo Radiante son: la bomba, sistema de distribución, válvula reguladora y los purgadores. [25]

- **Bomba:** Es la responsable de hacer circular el agua a través de las tuberías que se encuentran instaladas bajo el suelo. Esta bomba no consume mucha potencia pero debe de seleccionarse una que sea capaz funcionar con agua caliente, ya que hay que recordar que la temperatura de entrada al sistema está alrededor de 40°C.
- **Sistema de Distribución:** Es el encargado de distribuir el agua caliente que viene de la fuente hacia los distintos sistemas instalados en la edificación y a la vez recoger el agua fría que sale de los distintos sistemas.
- **Válvula reguladora:** Esta válvula es importante para regular el aporte calorífico a la habitación donde ha sido instalado el Suelo Radiante, en caso el ambiente esté muy caliente ésta debe de cerrarse un poco; en cambio si el ambiente está un poco frío debe de abrirse. Esta parte puede ser automatizada, de modo que, una termopila pueda medir la temperatura de la habitación y para que luego un controlador ordene abrir o cerrar un poco la válvula, según sea el caso.
- **Purgadores:** Es un dispositivo de seguridad, el cual se encarga de evacuar el aire del sistema.

1.2.7. Energía Solar Térmica

La energía solar es una fuente de energía inagotable permitiendo obtener diversidad de maneras de aprovecharla, teniendo así una de ellas la energía solar térmica, permitiendo mediante la radiación solar obtener calor utilizando captadores solares térmicos. [26]

Los colectores solares son dispositivos con una capacidad de obtener la energía proveniente mediante la radiación solar, aprovechándola para calentar ciertos fluidos, que generalmente es agua. Dependiendo de esta funcionalidad, estos pueden ser de baja, media o alta temperatura, siendo los primeros que han desarrollado un mayor crecimiento comercial. [26]

Dentro de las aplicaciones más comunes de la energía solar térmica está el complementar el abastecimiento de agua caliente sanitaria, pudiendo así suministrar de hasta un 70% de la necesidad durante el año. [26]

Sin embargo, una aplicación menos extendida de la energía solar térmica es la que se utiliza como calefacción por suelo radiante o como calentamiento de piscinas cubiertas o exteriores, debido a que estas operan a temperaturas más bajas que los sistemas de radiadores convencionales. [26]

1.2.8. Sensores de Temperatura y Humedad

Dentro de las variables físicas a controlar se encuentran la temperatura y la humedad, por ende en el siguiente análisis se muestra los diferentes tipos de sensores que sirven para dicho propósito:

Sensores de Temperatura

Un sensor de temperatura es un elemento que transforma la variación de temperatura (variable física) en señales eléctricas, para posteriormente ser procesadas por dispositivos eléctricos o electrónicos. [27]

Sensor de temperatura termopar

Este tipo de sensores está compuesto por dos alambres de metales distintos que están unidos en cualquiera de sus extremos. Para medir los cambios de temperatura en esta unión termopar se genera un voltaje por cada variación de temperatura, es decir, si la temperatura aumenta, el voltaje subirá. [27]

Estos sensores son económicos de fabricar y muy fiables pero con la desventaja principal que no son precisos. [27]

Sensor de temperatura por resistencia

Están compuestos por materiales que varían su resistencia eléctrica al variar la temperatura. Dentro de este tipo de sensores podemos encontrar dos tipos de dispositivos con estas características mencionados a continuación: [27]

RTD (Resistance Temperature Detector), la variación de la resistencia del metal varía de forma más o menos lineal en dependencia de la temperatura, por otro lado se encuentran los termistores que están basados en semiconductores cerámicos, estos funcionan de manera similar pero con la variación de la resistencia en forma no lineal en dependencia de la variación de la temperatura. [27]

Sensor de temperatura bimetalico

Son también llamados termostatos, su funcionamiento está basado en interruptores que generan una salida de tipo on/off y conmutan de un estado a otro cuando se alcanza un determinado valor de temperatura. [27]

Estos sensores constan de un bimetálico, es decir, una pieza que está formada por dos metales con diferente coeficiente de dilatación térmica. Cuando estos se calientan, uno de los lados tiende a dilatarse más que el otro, resultando una curva traducida en un movimiento mecánico. Los sensores bimetalicos no necesitan de una fuente de alimentación y no poseen una sensibilidad adecuada como las RTD o los termopares. [27]

Sensor de temperatura monolíticos o de silicio

Son dispositivos desarrollados a base de las propiedades térmicas resultantes entre la unión de semiconductores (uniones PN), generalmente la dependencia entre la tensión base – emisor de los transistores bipolares con la temperatura cuando la corriente del colector es constante. [27]

Generalmente estos dispositivos poseen sus propios circuitos de análisis y procesamiento de señales, así también diversidad de funciones de interfaces con el mundo exterior. [27]

Sensores de Humedad

Los sensores de humedad miden la cantidad de líquido o la humedad relativa en un área determinada. Las magnitudes obtenidas por estos sensores se convierten en señales eléctricas. [28]

Existen una gran variedad de sensores de humedad, mencionando a continuación conforme se relacionen al principio físico para realizar la correspondiente medición: [28]

Sensores de humedad mecánicos

Son los que en presencia de la humedad, ciertos materiales cambian de dimensiones teniendo entre estos como por ejemplo: las fibras sintéticas u orgánicas, o el tipo de cabello. [28]

Sensores de humedad conductivos

Teniendo en cuenta las propiedades del agua para conducir corriente, mediante unas rejillas de reducido tamaño de oro, a través de ellas circule una corriente eléctrica y en dependencia de esta fue posible deducir el valor de humedad. [28]

Sensores de humedad capacitivos

En presencia de la humedad, la variación de elementos capacitivos es mayor o menor, obteniendo así un valor aproximado de humedad. [28]

Sensores de humedad infrarrojos

Estos tipos de sensores están diseñados para absorber parte de la radiación que está inmersa en el vapor de agua. [28]

Sensores de humedad resistivos

Estos son empleados en su gran mayoría para fines agrónomos, ya que miden la conductividad de la tierra, es decir, a mayor cantidad de agua en la tierra, mayor conductividad posee esta. [28]

1.2.9. Monitoreo Industrial

Este tipo de monitoreo está orientado al proceso de optimización en plantas industriales y en fábricas, debido a que utiliza herramientas y tecnologías con dispositivos inteligentes que recopilan datos de una manera óptima, esencial para la toma de decisiones y así poder elevar eficientemente las operaciones y la productividad. [29]

El internet de las cosas en los procesos de monitoreo industrial

Hoy en día la toma de decisiones acertadas para resolver problemas y de manera eficaz y oportuna se ha convertido en un reto en las plantas industriales, por lo que existe la necesidad de sistemas tecnológicos capaces de supervisar, controlar y monitorear remotamente todos estos procesos. [29]

La aplicación de los sistemas denominados “Internet de las cosas en sistemas de monitoreo en plantas industriales”, permiten gestionar y optimizar los procesos en tiempo real de toda la planta, haciendo uso de herramientas tecnológicas como:

tabletas, smartphones, computadoras, etc., que en dependencia de su uso, permiten monitorear y supervisar los procesos de las plantas industriales desde cualquier punto donde se encuentre el operario y en todo momento. [29]

En conclusión y mostrando sus ventajas más importantes, los sistemas IoT en procesos industriales permiten al operario contar en el momento y en tiempo real con información detallada y precisa de las operaciones de los sistemas que lo conforman, además de ello, aplicando sistemas inalámbricos que faciliten la tarea de enviar la información a servidores dedicados para este tipo de procesos, pudiendo así generar informes en el momento, entre otros beneficios. [29]

Comunicaciones inalámbricas:

La comunicación inalámbrica no es más que un concepto que se emplea para hablar del uso de la tecnología sin cables, razón por la que se puede establecer una conexión entre varios dispositivos y sin un gran volumen de elementos físicos extras.

Tecnologías inalámbricas:

En la actualidad, existe diversidad de estándares de tecnologías inalámbricas, entre las cuales se tiene como referencia las siguientes:

Estándar IEEE 802.11:

El éxito de IEEE 802.11 o Wi-Fi radica en que, al ser el primer estándar implementado en el mercado, se convirtió en el más utilizado para la creación de WLANs. Además, ha demostrado su capacidad para ofrecer acceso de banda ancha en múltiples entornos públicos a precios asequibles. [30]

Ventajas:

- **Movilidad:** Las redes inalámbricas pueden proveer a los usuarios de una LAN acceso a la información en tiempo real en cualquier lugar dentro de la organización. Esta movilidad incluye oportunidades de productividad y servicio que no es posible con una red alámbrica.

- **Fácil manejo e instalación:** La instalación de una red inalámbrica puede ser tan rápida y fácil y además que puede eliminar la posibilidad de tirar cable a través de paredes y techos.
- **Flexibilidad en la instalación:** La tecnología inalámbrica permite a la red ir donde la alámbrica (red cableada) no puede ir.
- **Costo de propiedad reducido:** Los beneficios y costos a largo plazo son superiores en ambientes dinámicos que requieren acciones y movimientos frecuentes.
- **Escalabilidad:** Los sistemas de WLAN's pueden ser configurados en una variedad de topologías para satisfacer las necesidades de las instalaciones y aplicaciones específicas. Las configuraciones son muy fáciles de cambiar y además es muy fácil la incorporación de nuevos usuarios a la red.

Desventajas: [31]

- **Interferencia:** Debido al rango de frecuencia en el que trabajan (2.4Ghz) están muy propensas a las interferencias de señales muy comunes como lo son los teléfonos inalámbricos.
- **Seguridad:** Debido a la forma en que fueron creadas son muy vulnerables y accesibles con dispositivos móviles.

1.2.10. Sistema de Tuberías

Está conformado por una red de tuberías distribuidas para poder transferir de manera adecuada el calor, existen tres formas de hacer las instalaciones las cuales son las siguientes: [25]

- **Distribución en serpentín:** Es la forma más sencilla de hacer la distribución, pero la desventaja de este sistema radica en la gran diferencia de temperaturas que existe entre la salida y la entrada del agua de circulación; esto puede hacer que el usuario sienta esa diferencia en el ambiente en el que se encuentra.



Figura 11. Distribución en serpentin. [25]

- **Distribución en doble serpentin:** A diferencia que la distribución en serpentin, ésta no genera una gran diferencia de temperaturas entre la entrada y salida, ya que siempre va una tubería más fría con una más caliente, compensando la diferencia de temperaturas que se tenía en la distribución anterior. La desventaja de este sistema se genera al momento de dar la forma a la distribución ya que a veces se tiene que formar radios muy pequeños que dificultan la instalación.



Figura 12. Distribución en doble serpentin

Fuente: Diseño energético para suelo radiante [25]

- **Distribución en espiral:** Al igual que la distribución en doble serpentin, esta no genera una notoria diferencia de temperaturas en la habitación, además no se generan radios pequeños en el momento de la instalación, por ello, este tipo de instalaciones es la recomendada para los suelos radiantes.

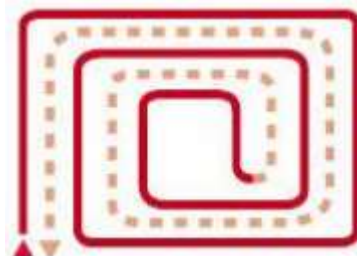


Figura 13. Distribución en espiral. [25]

1.3.Objetivos

1.3.1. Objetivo General

Implementar un prototipo de sistema de monitoreo y control difuso de la humedad y temperatura en naves avícolas con suelo radiante solar y ventilación forzada.

1.3.2. Objetivos Específicos

- Evaluar los métodos de inferencia difusa aplicados al control de humedad y temperatura en naves avícolas.
- Analizar las tecnologías existentes de monitoreo y su esquema de comunicaciones en naves avícolas.
- Diseñar un prototipo de sistema de monitoreo y control difuso de la humedad y temperatura en naves avícolas con suelo radiante solar y ventilación forzada.

CAPÍTULO II

METODOLOGÍA

2.1. Materiales

A continuación se realizó el diseño y la implementación de la propuesta del trabajo de investigación mediante el uso de la plataforma de entorno de desarrollo de software libre Arduino y el sistema operativo Raspbian de la distribución Linux, además del uso de sensores y actuadores para el control de proceso y el aprovechamiento del sol como energía renovable para impulsar calor a través del sistema como se muestra en la siguiente figura:

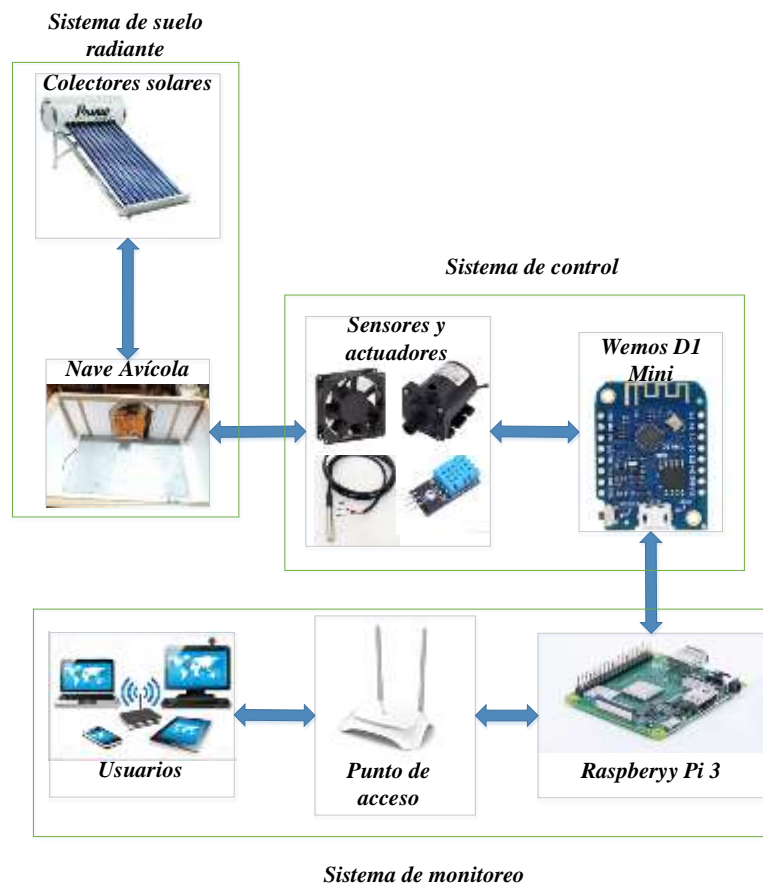


Figura 14. Diseño del prototipo del sistema de control y monitoreo.

Elaborado por: el investigador

Mediante la integración de todos los dispositivos que intervienen en cada uno de las tres etapas se logró, en primer lugar, la obtención de datos reales de temperatura y humedad mediante el uso de los sensores, una vez obtenidos estos datos son enviados al microcontrolador el cual realiza dos procesos: primero, el proceso de control que consta en utilizar lógica difusa mediante el uso del modelo de Mamdani adaptar las señales de temperatura y humedad idóneas para el correcto crecimiento y desarrollo de las aves en las naves, y segundo el proceso de almacenamiento de los datos en la base de datos montada en servidores LAMP mediante la utilización de la microcomputadora Raspberry Pi3, como última etapa, una vez obtenidos los datos, estos fueron mostrados mediante conexión inalámbrica hacia los usuarios registrados en el sistema.

A continuación se muestra un diagrama de bloques que permite visualizar las etapas que se siguen en todo el proceso:

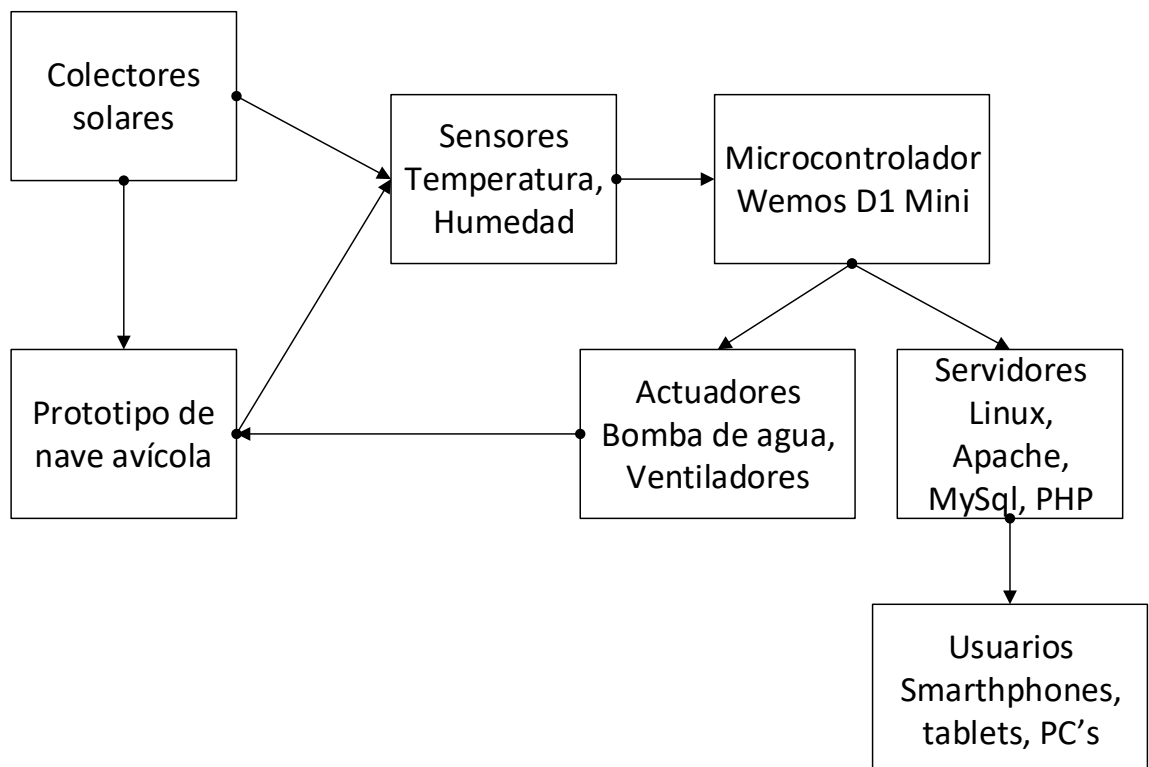


Figura 15. Diagrama de control de bloques del proceso

Elaborado por: el investigador

Tablas comparativas de entre los materiales que componen cada etapa

Sistema de Suelo Sadiante

Hoy en día prevalecen dos sistemas de suelo radiante. El primero utiliza la energía eléctrica como fuente emisora de calor y el segundo utiliza fundamentalmente la recirculación de agua por tuberías.

- Suelo radiante eléctrico

Este tipo de suelo radiante únicamente se alimenta de energía eléctrica, instalándose bajo un revestimiento en el suelo, su funcionamiento está basado en una estructura en forma de bobinas que al calentarse traspasan el calor al suelo exterior.

Al ser propulsada por energía eléctrica este tipo de instalaciones deben ser debidamente aisladas, por lo que generalmente se utiliza una lámina impermeable aislante de poliestireno extruido.

- Suelo Radiante por Agua

Este tipo de suelo radiante es mucho más eficiente que el anterior, debido a que este proporciona un mayor ahorro de energía, debido a que su eficiencia y durabilidad es mayor. [32]

Con solo tener una fuente que provea de agua caliente que bordee los 46°C, el sistema fácilmente consigue una temperatura ambiente de entre los 18°C y los 22°C. [32]

A continuación se muestra una tabla comparativa con los principales parámetros que intervienen en estos tipos de sistemas: [32]

Tabla 3. Comparación entre sistemas de suelos radiantes.

Suelo radiante eléctrico	Suelo radiante por agua
Su funcionamiento se basa en la instalación de una resistencia calefactora colocada sobre un material aislante, esparciendo por el suelo uniformemente el calor que esta genera.	Su funcionamiento está basado en la instalación de tuberías en el piso por la cual circula agua caliente, haciendo de la placa que los recubre una superficie homogénea de calor.
Sistema de mucho confort y funcionamiento a baja temperatura que oscilan entre los 25°C a 29°C.	Este sistema opera con temperaturas más altas, estas oscilan entre los 35°C y los 40°C.
Sistema de calefacción inmediata, por lo tanto es más fácil de controlar y brinda más confort.	La fuente emisora de calor tiene que calentar el agua, por lo tanto tarda algo más en climatizar el ambiente.
Facilita su control mediante cronotermostatos, en caso de ser necesario para cada una de las divisiones.	La temperatura es controlada a través del encendido del sistema de propulsión del agua.
No requiere de un calentador.	Requiere de una fuente emisora de calor
Este sistema generalmente está individualizado por secciones.	Posee un circuito continuo, manejado por válvulas por zona.
En el ámbito de seguridad no necesitan mantenimientos durante largos periodo de tiempo.	Requiere mantenimientos mínimos debido a que en algún momento durante el transcurso del tiempo se necesite purgar el sistema

Elaborado por: el investigador

Sistema de Control

Dentro de esta etapa se procede a analizar los tipos de sensores que intervienen en la toma de datos dentro del prototipo de nave avícola, estos son: sensores de humedad y sensores de temperatura, además para procesar estos datos se analizó diferentes tipos de microcontroladores como se muestra a continuación:

- **Sensores de Temperatura**

En la siguiente tabla se muestra la comparación técnica entre distintos sensores de temperatura más populares eficientes para el desarrollo de prototipos y proyectos con microcontroladores:

Tabla 4. Comparación de sensores de temperatura.

Modelo	Sistema	Calibración	Consumo	Lectura	Precisión	Rango °C	Voltaje
LM32DZ	Analógico	°C	Bajo	10mV/°C	0,5 °C	-55 a 150	4 – 30 V
TMP36	Analógico	°C	Bajo	10mV/°C	±2 °C	-40 a 125	2,7 a 5,5 V
LM75	Digital	°F - °C	Bajo	ADC 11 bits	±2 °C	-25 a 100	2,8 a 5,5 V
TMP100	Digital	°F - °C	Bajo	ADC 9-12 bits	±2 °C	-25 a 85	2,7 a 5,5 V
DS18B20	Digital	°F - °C	Bajo	ADC 9-12 bits	±0,5 °C	-10 a 85	3 a 5,5 V
BMP180	Digital	°C	Ultra bajo	0,1 °C	1 °C	-20 a 100	3,3 a 5 V

Elaborado por: el investigador

- **Sensores de Humedad**

En la siguiente tabla se muestra la comparación técnica entre distintos sensores de humedad más populares eficientes para el desarrollo de prototipos y proyectos con microcontroladores:

Tabla 5. Comparación de sensores de humedad.

Modelo	Sistema	Rango de humedad	Velocidad de muestreo	Voltaje	Corriente máxima
DHT11	Digital	20 – 80% / \pm 5%	1Hz	3-5V	2.5mA
DHT22	Digital	0 – 100% / \pm 2-5%	0.5Hz	3-5V	2.5mA
SHT7X	Digital	0 – 100% / \pm 3%	1Hz	2.4 – 5.5 V	3.5mA
FC-28	Analógico	0-100%	-	5 V	–

Elaborado por: el investigador

- Microcontroladores

La tecnología avanza día a día a pasos agigantados, recientemente han salido al mercado varios chips que pueden ser implementados en diferentes tipos de controladores, es por ello que se ha visto la necesidad de comparar cada uno de ellos mostrados en la siguiente tabla: [33]

Tabla 6. Comparación de microcontroladores

Especificaciones	ESP8266	ESP32
MCU	Single-Core 32 bits	Single-Core 32 bits
Wi-Fi 802.11 b/g/n	Si, ancho de banda 20 MHz	Si, ancho de banda 40 MHz
Bluetooth	No	Bluetooth 4.2
Frecuencia típica de operación	80 MHz	160 MHz
GPIO	17	36
SPI/I2C/I2S/UART	2/1/2/2	4/2/2/2
ADC	10-bits	12-bits
Sensor de temperatura	No	Si
Temperatura de Trabajo	-40 °C – 125 °C	-40 °C – 125 °C

Elaborado por: el investigador

Sistema de Monitoreo

Para desarrollar el sistema de monitoreo en este proyecto, se analizó distintos tipos de modelos de Raspberry existentes en el mercado, a continuación en la tabla se detalla sus características y comparaciones:

Tabla 7. Comparacion Raspberry's

Características	Raspberry Zero	Raspberry 4 B	Raspberry Pi 3
CPU	1GHz ARM1176JZF-s	1.5GHz QUAD ARM Cortex-A72	1.2GHz QUAD ARM Cortex-A53
RAM	512 MB	1 – 4 GB	1GB
USB	1 Micro	2(2.0), 2(3.0)	4
Boot	uSD	uSD	uSD
Red	No	ETH 1000 Dual Band, Wifi, Bluetooth.	ETH 10/100, Wifi, Bluetooth
Alimentación	160mA 0.8w / 5v	2.5A 12.5w / 5V	2.5A 12.5W / 5V
Tamaño	65 x 30 mm	85 x 56 mm	85 x 56 mm

Elaborado por: el investigador

2.1.1. Raspberry Pi 3 Model B+

La Raspberry Pi 3 B+ es el último producto dentro del rango de las Raspberry Pi 3, posee un procesador de cuatro núcleos de 64-bits corriendo a 1.4 Ghz, además se conecta inalámbricamente a la LAN utilizando las frecuencia de 2.4 Ghz y 5 Ghz, a continuación, se observan las especificaciones más a detalle: [34]

Tabla 8. Características Raspberry Pi 3 B+

Procesador	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4 GHz
Memoria	1GB LPDDR2 SDRAM
Conectividad	2.4 GHz y 5 GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE Gigabit Ethernet sobre USB 2.0 (máximo rendimiento 300 Mbps) 4 × USB 2.0 puertos
Acceso	Extensión de GPIO de 40 pines
Video y Sonido	1 × tamaño completo HDMI Puerto de visualización DSI MIPI Puerto de cámara CSI MIPI Salida estéreo de 4 polos y puerto de video compuesto
Multimedia	H.264, decodificación MPEG-4 (1080p30); Codificación H.264 (1080p30); OpenGL ES 1.1, gráficos 2.0
Soporte de tarjeta de memoria SD	Formato micro SD para cargar el sistema operativo y el almacenamiento de datos.
Potencia de entrada	5 V / 2.5 A DC a través de conector micro USB 5 V DC a través de cabecera GPIO Power over Ethernet (PoE): habilitado (requiere PoE HAT por separado)
Entorno	Temperatura de funcionamiento, 0–50 °C.

Elaborado por: el investigador



Figura 16. Raspberry Pi 3 B+ [34]

2.1.2. Arduino IDE

Arduino Nació en el año 2005 el Instituto de Diseño Interactivo de Ivrea (Italia). Arduino apareció por la necesidad de contar con un dispositivo para utilizar en aulas que fuera de bajo coste. La idea original fue, fabricar una placa para uso interno de la escuela. [35]

Los principales responsables de la idea y diseño de Arduino fueron Massimo Banzi, David Cuartielles, David Mellis, Tom Igoe y Gianluca Martino. [35]

Arduino es libre y extensible: esto quiere decir que cualquiera que desee ampliar y mejorar el diseño hardware de las placas como el entorno de desarrollo, puede hacerlo sin problemas. Esto permite que exista un rico ecosistema de placas electrónicas no oficiales para distintos propósitos y de librerías de software de tercero, que pueden adaptarse mejor a nuestras necesidades.

Características: [35]

- Arduino tiene una gran comunidad.
- Su entorno de programación es multiplataforma.
- Lenguaje de programación de fácil comprensión.
- Bajo costo.
- Re-usabilidad y versatilidad.



Figura 17. Arduino IDE Software [35]

2.1.3. Wemos D1 mini

La tarjeta Wemos D1 mini posee 11 pines digitales de entrada/salida, todos los pines soportan interrupciones/pwm/I2C/one-wire excepto el pin D0, además posee 1 una entrada análoga (3.2V entrada máxima). Mediante su puerto Micro USB fácilmente se conecta con el IDE, siendo compatible con MicroPython, Arduino, nodemcu. En la siguiente tabla podemos observar más a detalle las especificaciones técnicas: [36]

Tabla 9. Características Wemos D1 Mini

Microcontrolador	ESP-8266EX
Voltaje de operación	3.3V
Pines digitales E/S	11
Pin analógico de entrada	1 (Máximo 3.2V)
Velocidad de reloj	80MHz/160MHz
Flash	4 Mega bytes
Longitud	34.2mm
Ancho	25.6mm
Peso	3g

Elaborado por: el investigador



Figura 18. Wemos D1 Mini [36]

2.1.4. Bomba de agua 800l/h

Esta bomba de corriente continua utiliza componentes electrónicos avanzados y un eje resistente al desgaste de alta calidad, genera un funcionamiento suave, de alta eficiencia, buen rendimiento, larga vida útil; puede brindar un trabajo continuo de larga duración, bajo ruido, seguridad y protección del medio ambiente, son ampliamente utilizadas en la industria, la investigación científica y la industria aeroespacial. En la siguiente tabla se muestra las especificaciones a más detalle: [37]

Tabla 10. Características Bomba de agua 800l/h

Nombre	Bomba de agua
Material	Plástico
Color	Negro
Entrada/Salida	Cabezal macho de ½"
Voltaje	12 V CC/24 V CC
Corriente nominal máxima	1000mA
Poder	22 W
Flujo máximo	800 L/H
Cabezal de agua máximo	5 m
Temperatura máxima del agua circulante	100 °C
Tamaño	6,5 cm (D) x 8 cm (H)

Elaborado por: el investigador



Figura 19. Bomba de agua 800l/h [37]

2.1.5. Sensor de temperatura DS18B20

Es un dispositivo que realiza su comunicación digitalmente, integrando en sus tres terminales los pines Vcc, GND y Data. Este dispositivo nos brinda la opción de

comunicación a través del protocolo OneWire, permitiendo conectar múltiples dispositivos a través de un solo cable, diferenciando de la gran mayoría de protocolos que funcionan utilizando dos cables. [38]

Características del cable: [38]

- Recubrimiento de acero inoxidable con un diámetro de 6mm por un largo de 30mm
- Largo total: 91cm
- Diámetro del cable unifilar: 4mm
- Incluye un sensor de temperatura DS18B20

Conexión: [38]

En dependencia del fabricante, los cables del sensor varía siguiendo las especificaciones siguientes:

- Sensor con 4 cables: cable de datos es blanco, GND es negro, Vcc es rojo, el recubrimiento de malla es GND.
- Sensor con 3 cables: cable de datos es amarillo o blanco, GND es negro y Vcc es rojo.

Características: [38]

- Intervalo de medición de temperatura: -55 a 125 °C
- Resolución para adquisición de datos: configuración de 9 a 12 bits
- Protocolo de comunicación: interfaz One_Wire
- Identificador interno único de 64 bits
- Pueden conectarse múltiples sensores al mismo pin
- Rango de precisión: $\pm 0,5^{\circ}\text{C}$ (de -10°C a 85°C)
- Lectura de datos menor a 750ms
- Opera desde los 3.0 V a 5.5 V



Figura 20. Sensor DS18B20 [38]

2.1.6. Sensor de humedad y temperatura DHT11

Dispositivo que permite medir humedad y temperatura simultáneamente, brindando salidas netamente digitales, además integra un microcontrolador internamente para el tratamiento de la señal. [39]

Características: [39]

- Voltaje de operación: $3Vdc \leq Vcc \leq 5Vdc$
- Señal de salida: Digital
- Rango de medida de temperatura: De 0 a 50 °C
- Resolución de temperatura: 0.1 °C
- Rango de medida de humedad: De 20% a 90% RH
- Precisión de humedad: 4% RH
- Resolución de humedad: 1%RH
- Tiempo de respuesta: 1 segundo



Figura 21. Sensor DHT11 [28]

2.1.7. Driver doble para control de velocidad de motor TB6612FNG

Esta pequeña placa es una forma fácil de usar el controlador de motor dual TB6612FNG de Toshiba, que puede controlar de forma independiente dos motores de CC bidireccionales o un motor paso a paso bipolar. Una tensión de motor recomendada de 4.5 V a 13.5 V y una salida de corriente pico de 3 A por canal (1 A continuo) hacen de este un excelente controlador de motor para motores de baja potencia. [40]

Especificaciones: [40]

- Controlador de motor de puente doble H: puede manejar dos motores de CC o un motor paso a paso bipolar
- Voltaje de motor recomendado (VMOT): 4.5 V a 13.5 V (puede funcionar hasta 2.5 V con rendimiento reducido)
- Voltaje lógico (VCC): 2.7 V a 5.5 V
- Máximo de corriente de salida: 3 A por canal
- Corriente de salida continua: 1 A por canal (se puede conectar en paralelo para entregar 2 A continuo)
- Frecuencia máxima de PWM: 100 kHz
- Circuito de apagado térmico incorporado
- Filtros de condensadores en ambas líneas de alimentación.
- Protección de potencia inversa en la alimentación del motor.



Figura 22. TB6612FNG [40]

2.1.8. Servidor LAMP

Una de las combinaciones de aplicaciones más populares en LINUX es la conocida como LAMP, donde LAMP es el acrónimo de Linux Apache MySQL PHP. Hay que tener en cuenta que la mayoría de los sitios de Internet, hoy en día, utiliza estos servidores LAMP o su versión para Windows WAMP. [41]

Funciones de cada elemento: [41]

- Linux es el sistema operativo de código abierto. En este caso, RaspbianOS
- Apache2 es el servidor web.
- MySQL es el gestor de bases de datos de código abierto.
- PHP es un lenguaje de programación web



Figura 23. Servidor LAMP [41]

2.1.9. Punto de Acceso Inalámbrico N 450Mbps TL-WA901ND

Mediante el uso de este punto de acceso se logra la conexión y el envío y recepción de datos inalámbricamente a través de las tarjetas y los servidores, como principales características se tienen las siguientes: [42]

Características: [42]

- Velocidad de transmisión inalámbrica de 450 Mbps, provee una experiencia inalámbrica N fluida
- Es compatible con múltiples modos de operación: Punto de acceso, Cliente, Universal / Repetidor WDS, Puente Inalámbrico
- Fácil configuración de una conexión segura encriptada WPA al presionar el botón QSS
- De hasta 30 metros de alimentación a través de Ethernet para un despliegue flexible

2.1.10. Ventilación

Para mantener un área ventilada y con suficiente flujo de aire, para ello se utilizó el siguiente ventilador detallado a continuación: [43]

Especificaciones: [43]

- Voltaje de referencia: 12 V
- Rango de operación de voltaje: 7 – 13.8 V
- Corriente de referencia: 0.2 A
- Flujo de aire: 0,91 m³/min
- Ruido: 21dB



Figura 24. Ventilador [43]

2.1.11. Calentador de agua solar

Los calentadores de agua solares de agua, son aptos para cualquier tipo de ambiente, ya sea doméstico o industrial, en cualquier entorno donde se necesite eficientemente calentar el agua, evitando el uso de combustibles contaminantes. Estos calentadores solares poseen tres partes principales, detalladas a continuación: [44]

- Tubos colectores al vacío, que son los encargados de absorber la energía proveniente del sol y transferirla al agua.
- Tanque térmicamente aislado
- Estructura metálica que soporta a los equipos.

Estos calentadores solares están operativos desde el momento mismo de instalación, generalmente ubicados en las terrazas o azoteas de los hogares, o a su vez en un lugar abierto directamente orientado hacia el sol, debiendo obtener la mayor parte de la radiación solar durante el día. [44]

El proceso de calentado del agua está basado en el fenómeno denominado termosifón, es decir, sin necesidad de utilizar un sistema de recirculación artificial, el agua caliente es más ligera que el agua fría, por ende tiende a subir, estableciendo así una circulación natural. Con la finalidad de mantener el agua el mayor tiempo posible caliente se hace uso de un tanque térmicamente sellado para evitar que se pierda el calor generado. [44]

Gracias a la privilegiada ubicación de la geografía ecuatoriana y la sofisticada tecnología de los calentadores, fácilmente el líquido puede llegar a obtener una temperatura máxima que bordea los 90 °C. Para el uso cotidiano en condiciones normales la temperatura del agua no representa riesgos para el ser humano, teniendo en cuenta que para el uso de agua caliente no sobrepasa los 42 °C. [44]

En la siguiente figura se puede visualizar cómo está compuesto un calentador solar de agua, es decir, las partes que lo conforman para realizar todo el proceso de calentamiento del líquido: [44]



Figura 25. Componentes de calentador solar. [44]

2.1.12. Módulo RTC DS3231

Para controlar las fechas exactas que se necesitan a partir del inicio del proceso de crianza de las aves, es necesario contar con la presencia de un reloj, el mismo que nos

ayuda a comparar el día del proceso y el día actual, en la siguiente imagen se muestra el dispositivo a emplear: [45]

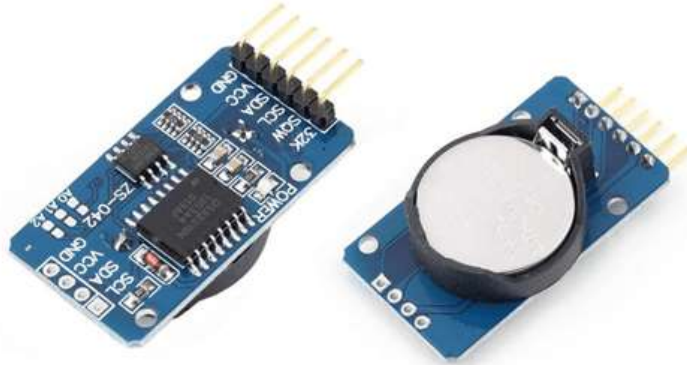


Figura 26. RTC 3231 Arduino [45].

Características: [45]

- Alimentación de voltaje: 3.0 a 5.0 Voltios.
- RTC de alta precisión, prestando facilidades para el mantenimiento de fecha/hora.
- Con un rango de operación de temperaturas de 0 °C a 40 °C de $\pm 2\text{PPM}$
- Memoria EEPROM I2C con módulo de reloj DS3231
- Posee batería de respaldo
- Registra años, meses, días, horas, minutos y segundos, además posee compensación de días en años bisiestos.
- Salida de señal cuadrada programable.

2.2. Métodos

2.2.1. Modalidad de la Investigación:

Investigación Aplicada

El presente proyecto fue definido como proyecto de Investigación Aplicada, porque se realizó un prototipo tangible funcional como producto de la investigación realizada en el proyecto.

Investigación bibliográfica-documental

La investigación bibliográfica-documental se obtuvo de libros, revistas, artículos y base de datos científicas de la Universidad Técnica de Ambato que proporcionaron el conocimiento de las indagaciones existentes como teorías, resultados, experimentaciones, instrumentos y técnicas usadas para llevar a cabo el proyecto.

Investigación de campo

Se realizó la investigación de campo para el análisis y estudio del sistema de control electrónico, para la obtención de información de la situación actual de las granjas avícolas y el análisis de su entorno.

2.2.2. Población y Muestra:

Por la característica de la investigación no se precisó población y muestra.

2.2.3. Recolección de Información:

Para el desarrollo del proyecto, se utilizaron documentos, tales como libros, paper's investigativos, proyectos de investigación previos tomados de los repositorios pertenecientes a la Universidad Técnica de Ambato, así como de repositorios de las diferentes Universidades del país, además de dispositivos electrónicos para la adquisición de datos.

2.2.4. Procesamiento y Análisis de Datos:

El procesamiento y análisis de datos contemplan diferentes planes para cada etapa, que se menciona a continuación:

Plan de procesamiento de la información:

- Aseguramiento de los datos suministrados de la investigación bibliográfica son limpios, correctos y útiles.
- Ordenamiento de elementos secuencialmente de conjuntos relacionados a sistemas de control difuso, sistemas de monitoreo, granjas avícolas, suelo radiante y energía solar.
- Reducción de los detalles de los datos a sus principales puntos de interés específicos relacionados al sistema a desarrollar.
- Combinación de múltiples piezas de datos para obtener un producto semejante al propuesto.
- Colección, organización, análisis, interpretación y presentación de dispositivos electrónicos óptimos para el procesamiento de datos.
- Lista detallada o resumen de los datos de información computarizada acorde a la investigación propuesta utilizando herramientas tecnológicas, comparando los distintos métodos de defusificación.

Plan de análisis de resultados:

- Validación del sistema de control difuso verificando su correcto funcionamiento al momento de su operación.
- Clasificación de los sistemas de monitoreo dando prioridades jerárquicas a los usuarios finales.
- Recapitulación de información de granjas avícolas.
- Agregación de sistema piso radiante para temperar la nave avícola adecuadamente reemplazando el GLP.
- Análisis de sistemas de calefacción mediante colectores de temperatura a base de energía solar.
- La información obtenida englobada en el sistema como tal fue almacenada en una base de datos y mostrada al usuario final.

2.3. Desarrollo del Proyecto

Teniendo en cuenta la lista de exigencias se puede empezar a elaborar la estructura general del sistema como se muestra en la siguiente figura:

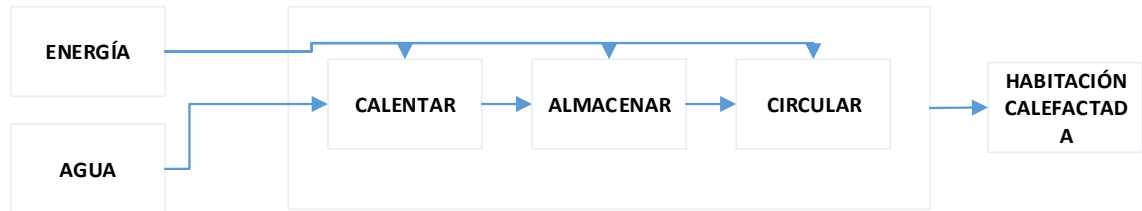


Figura 27. Estructura de funciones

Elaborado por: el investigador

2.3.1. Análisis de la situación actual de las naves avícolas.

Las naves avícolas necesitan continuamente mejorar y revisar el tema relacionado a la eficiencia energética para ahorrar costos, teniendo en cuenta que el tamaño promedio alberga alrededor de 20000 a 25000 pollos.

Con respecto al tamaño de cada nave avícola se tomó en cuenta que a menor tamaño fueron menos productivas, menos eficientes y menos automatizadas, caso contrario con las naves de mayor tamaño que fueron más productivas, más eficientes y con una mejor automatización.

Existen grupos de granjas en dependencia de su antigüedad, teniendo así como las más longevas denominadas como TIPO I, las elaboradas entre los últimos 15 a 20 años son conocidas como TIPO II, las instalaciones modernas de tipo llave en mano como TIPO III, y las naves tipo IV que resultan de la construcción paralela de dos naves TIPO III.

En la siguiente tabla se muestran parámetros fundamentales comparativos entre cada una de los cuatro tipos de naves avícolas existentes actualmente: [46]

Tabla 11. Datos de producción de los diferentes tipos de naves.

Parámetros	TIPO I	TIPO II	TIPO III	TIPO IV
m²	1200	1800	2025	4050
Densidad por m²	10	13	15	15
Pollos Entrada	12000	23400	30375	60750
Edad (días)	49,1	45,9	43,1	43,1
Bajas	7%	6%	5%	5%
Pollos Salientes	11160	21996	28856	57713
Kilos Carne	31248	61589	80797	161596

Elaborado por: el investigador

2.3.2. Estudio de temperatura, humedad y ventilación en las naves avícolas.

Temperatura

La fisionomía de los pollos de engorde es diferente a la de los humanos; al momento de encubar el ave, la temperatura debe ser de 37.6°C, teniendo como característica principal que al momento de nacer no pueden controlar por sí mismo su temperatura, a este efecto se lo denomina “poiquilothermia”, por tal motivo resulta una situación crítica el mantener la temperatura del entorno en el cual se desarrollan por lo menos las dos primeras semanas desde su nacimiento, pasado este tiempo estos ya han desarrollado la habilidad de regular su temperatura, llamando a este proceso “homeothermia”. [6]

Teniendo muy en cuenta estas observaciones se tiene la tabla 1 de resumen de temperatura óptima durante el proceso de crianza de los pollos: [47]

Tabla 12. Temperatura en la Crianza en naves avícolas.

EDAD	TEMPERATURA °C
1er. – 2do. Día	30 – 32
3er. – 7mo. Días	29 – 30
2da. Semana	27 – 29
3era. Semana	25 – 27
4ta. Semana	23 – 25
5ta. Semana en adelante.	21 – 23

Elaborado por: el investigador

Humedad

El término humedad hace referencia a la saturación del agua en porcentaje con respecto al aire en cualquier temperatura dada. Los rangos de humedad afectan directamente al ave en la capacidad para enfriarse mediante el jadeo, influyendo directamente en la producción de amoniaco dentro del criadero. Teniendo en cuenta que, a menor temperatura, la humedad fue mayor y viceversa, es por ello que se tiene la siguiente tabla de resumen de los valores adecuados de humedad dentro del galpón criadero de aves: [47]

Tabla 13. Humedad relativa en el proceso de crianza de aves en naves avícolas.

EDAD	HUMEDAD RELATIVA (%)
1er. – 2do. Día	35 – 40
3er. – 7mo. Días	35 – 45
2da. Semana	40 – 45
3era. Semana	45 – 50
4ta. Semana	50 – 55
5ta. Semana en adelante.	55 – 60

Elaborado por: el investigador

Ventilación

Hoy en día la mayoría de ventilaciones en naves avícolas se basan en sistemas que operan mediante depresión, llamada así porque dentro de la granja avícola se tiene una presión inferior a la del exterior, tratando de lograr que la granja avícola presente un vacío. Para llegar a este punto es necesario controlar con precisión la extracción como

la entrada de aire o ventanas, consiguiendo así el ajuste entre la entrada de aire y la salida de aire. [48]

La presión negativa o también conocida como depresión es medida en Pascales (Pa):

$$1Pa = 0.1 Kg/m^2$$

Hablando de aire neto, las aves necesitan cierta cantidad de caudal o aire neto (m^3/h) como se muestra en la siguiente tabla 3: [48]

Tabla 14. Ventilación por depresión en naves avícolas.

Edad (días)	Peso (Kg)	Caudal de Aire ($m^3/h/pollo$)
7	0.189	0.22
42	2.2	9.8

Elaborado por: el investigador

2.3.3. Análisis y comparación de los Métodos de Inferencia Difuso.

Los sistemas de inferencia son desarrollados para diversidad de tipos de controles, existen dos tipos de inferencia: inferencia difusa de Sugeno e inferencia difusa de Mamdani, a continuación se muestran las principales características entre cada uno de ellos:

Inferencia difusa de Mamdani: [49] [50]

- La función de membresía de salida está presente.
- El resultado es nítido, obtenido a través de la defusificación de las reglas.
- La superficie de salida es no continua
- Está avalado para sistemas MISO y MIMO
- Las reglas son interpretables, expresivas y consecuentes.
- Menos flexibilidad en el diseño del sistema.

Inferencia difusa de Sugeno: [49] [50]

- La función de membresía de salida no está presente.
- No posee defuzzificación, por lo tanto se obtiene un resultado nítido utilizando el promedio ponderado de las reglas consecuentes.
- La superficie de salida es continua.
- Está avalado únicamente para sistemas MISO
- Su interpretación no es muy clara.
- Brinda mayor flexibilidad en el diseño del sistema.

Una vez analizadas las ventajas y desventajas de los dos tipos de inferencia difusas, se procede a la elección del sistema de inferencia tipo Mamdani debido a que este posee y está avalado para trabajar con dos o más salidas al mismo tiempo. Este tipo de inferencia difusa fue aplicado al control de temperatura y humedad y además se utilizó el software MATLAB, aprovechando su amplia gama de herramientas de simulación desarrolladas, en este caso en concreto se hizo uso de la herramienta “Fuzzy Logic Tool Box”. [51]

El FLT Fuzzy Logic Toolbox por sus siglas en inglés, haciendo uso de un ambiente de desarrollo MATLAB es una herramienta para desarrollar sistemas difusos, utilizando la interfaz gráfica de usuario (GUI). [51]

Para poder diferenciar claramente cada uno de los dos métodos de inferencia difusos a analizar, se hizo uso de cinco herramientas gráficas para el modelado, el apartado de la edición y por último la observación: [51]

- FIS (Fuzzy Inference System), sistema de inferencia difuso.
- MF(Membership Function), funciones de membresía.
- Editor de reglas difusas.
- Observador de reglas difusas.
- Observador de superficie difusa.

2.3.4. Desarrollo del Sistema de Control con Método de inferencia de Mamdani

La ventana principal del Fuzzy Inference System presenta detalladamente la información principal para desarrollar un sistema difuso, tales como las variables de entrada, las variables de salida, además del método utilizado para la inferencia difusa como se muestra en la siguiente imagen:

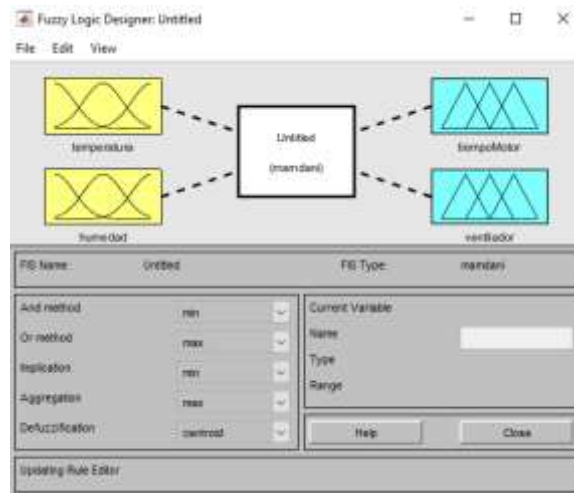


Figura 28. Interfaz gráfica, método de inferencia tipo Mamdani.

Elaborado por: el investigador

Para obtener los valores que fueron contenidos en las funciones de membresía, se analizaron los valores mínimos y máximos tanto de las variables de entrada como de las variables de salida, a continuación se muestra una tabla con dichos valores:

Tabla 15. Valores máximos y mínimos de variables.

Variable	Valor máximo	Valor mínimo	Valor medio	Valor referencia máximo	Valor referencia mínimo	Valor referencia medio
Temperatura	32.69 °C	20.5 °C	26.595 °C	6.095 °C	-6.095 °C	0 °C
Humedad	60%	35%	47.5%	12.5%	-12.5%	0%
Tiempo Motor	25 s	0 s	12.5 s	12.5 s	-12.5 s	0 s
Bits Ventilador	1023 bits	500 bits	761.5 bits	261.5 bits	-261.5 bits	0 bits

Elaborado por: el investigador

Funciones de membresía

En este apartado se redactó cinco funciones de membresía triangulares para cada variable de entrada y de salida, por lo tanto, en total se obtuvo veinte funciones de membresía detalladas a continuación:

- Variable temperatura:

$$M(\text{EFG}) = \text{TRAPMF}(-6,095 -6,095 -4,065 -2,035).$$

$$M(\text{EFP}) = \text{TRIMF}(-4,065 -2,035 0).$$

$$M(\text{EC}) = \text{TRIMF}(-2,035 0 2,025).$$

$$M(\text{ECP}) = \text{TRIMF}(0 2,025 4,055).$$

$$M(\text{ECG}) = \text{TRAPMF}(2,025 4,055 6,095 6,095).$$

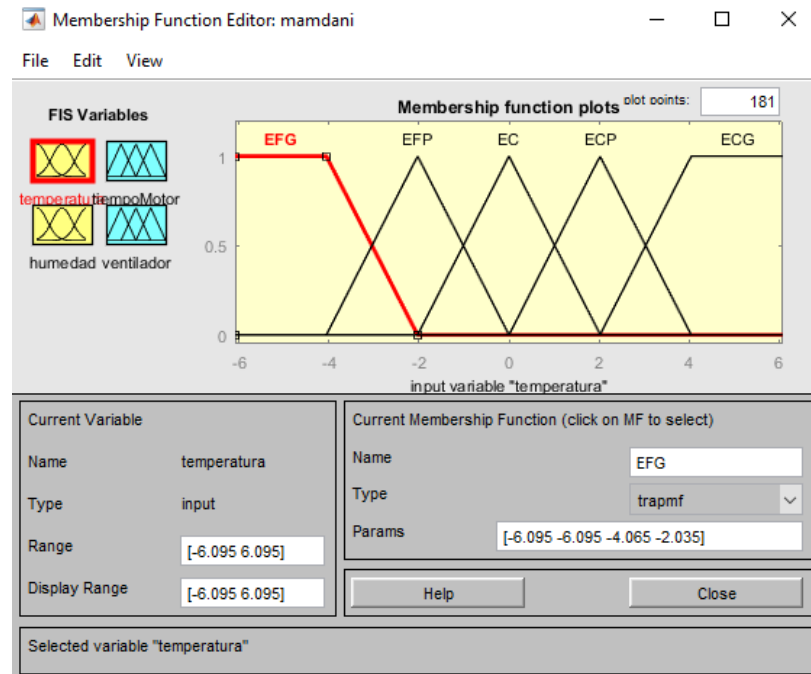


Figura 29. Funciones de membresía para la variable temperatura

Elaborado por: el investigador

- Variable tiempoMotor:

$$M(\text{TFG}) = \text{TRAPMF}(-12,5 -12,5 -8,34 -4,17).$$

$$M(\text{TFP}) = \text{TRIMF}(-8,34 -4,17 0).$$

$$M(\text{TC}) = \text{TRIMF}(-4,17 0 4,16).$$

$$M(\text{TCP}) = \text{TRIMF}(0 4,16 8,33).$$

$$M(\text{TCG}) = \text{TRAPMF}(4,16 8,33 12,5 12,5).$$

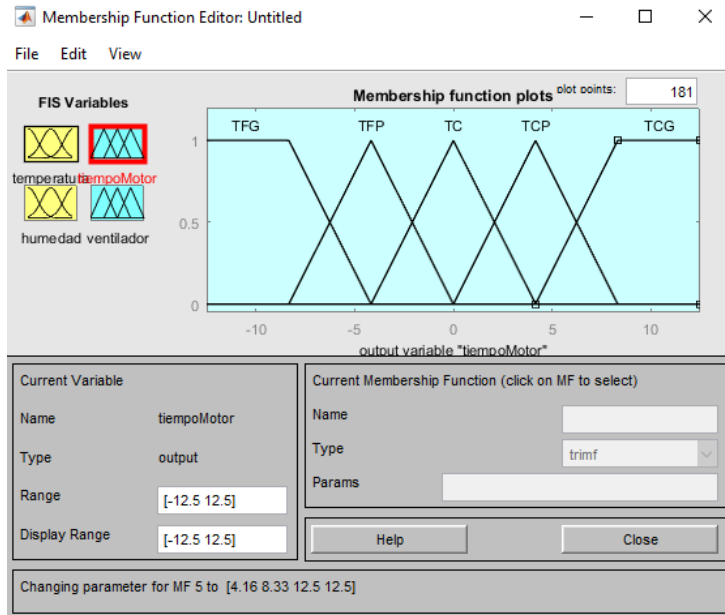


Figura 30. Funciones de membresía para la variable tiempoMotor.

Elaborado por: el investigador

- **Variable humedad:**

$$M(\text{HNG}) = \text{TRAPMF}(-12,5 \ -12,5 \ -8,34 \ -4,17).$$

$$M(\text{HNP}) = \text{TRIMF}(-8,34 \ -4,17 \ 0).$$

$$M(\text{HC}) = \text{TRIMF}(-4,17 \ 0 \ 4,16).$$

$$M(\text{HPP}) = \text{TRIMF}(0 \ 4,16 \ 8,33).$$

$$M(\text{HPG}) = \text{TRAPMF}(4,16 \ 8,33 \ 12,5 \ 12,5).$$

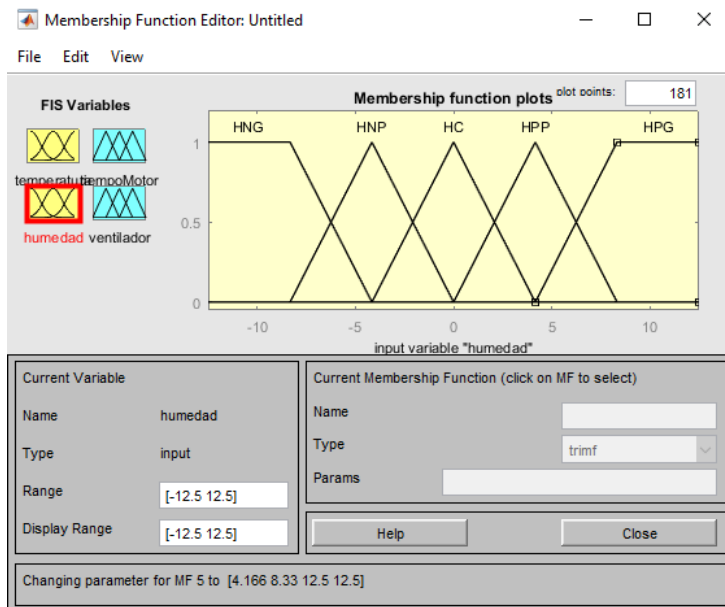


Figura 31. Funciones de membresía para la variable humedad.

Elaborado por: el investigador

- **Variable ventilador:**

$$M(\text{BNG}) = \text{TRAPMF}(-261,5 \ -261,5 \ -174,3333333 \ -87,16666667).$$

$$M(\text{BNP}) = \text{TRIMF}(-174,3333333 \ -87,16666667 \ 0).$$

$$M(\text{BC}) = \text{TRIMF}(-87,16666667 \ 0 \ 87,16666667).$$

$$M(\text{BPP}) = \text{TRIMF}(0 \ 87,16666667 \ 174,3333333).$$

$$M(\text{BPG}) = \text{TRAPMF}(87,16666667 \ 174,3333333 \ 261,5 \ 261,5).$$

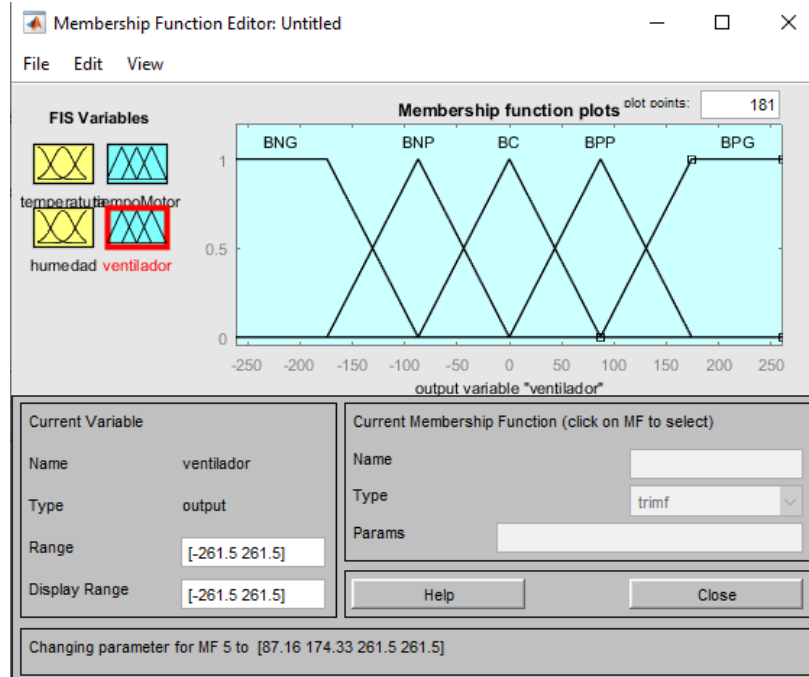


Figura 32. Funciones de membresía para la variable ventilador.

Elaborado por: el investigador

Editor de reglas difusas

Después de haber desarrollado la base fundamental del controlador difuso que son la edición de las funciones de membresía que contiene la información necesaria obtenida desde un operador conocedor del proceso a desarrollar, se pasó a transcribir todas estas al editor de reglas. En este apartado se encuentran los nexos lógicos como NOT, OR o AND, los cuales fueron utilizados para las reglas si-entonces mostrado a continuación:



Figura 33. Editor de reglas difusas.

Elaborado por: el investigador

Visualizador de reglas difusas

Después de haber desarrollado el algoritmo de fusificación, la ventana de visualizador de reglas muestra un esquema general de cómo se comporta el sistema de control difuso a variables de set points de entrada con respecto a sus señales de salida como se observa a continuación:

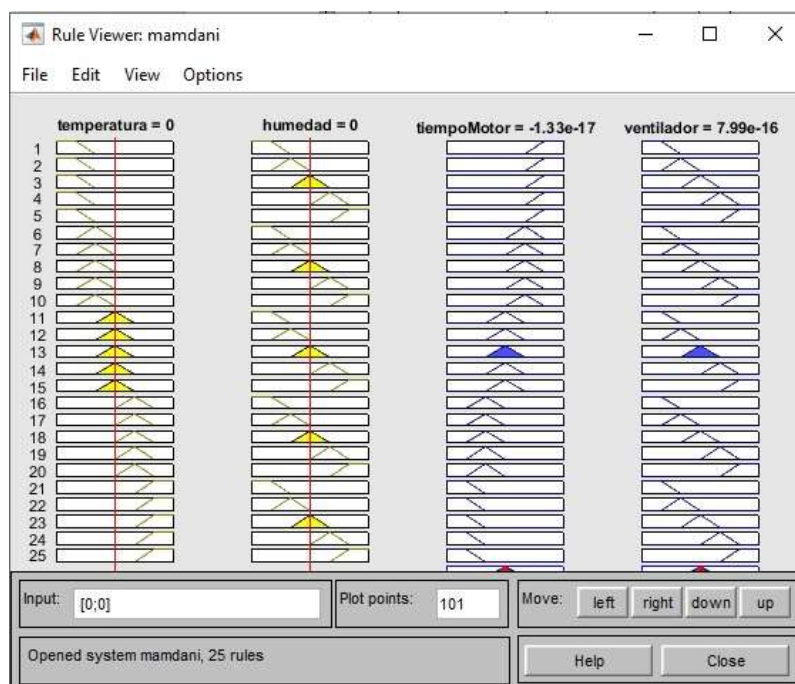


Figura 34. Ventana visualizador de reglas.

Elaborado por: el investigador

Analizando más a detalle, se puede observar que existen cuatro columnas las mismas que representan las variables del proceso, es decir temperatura, humedad, tiempoMotor y ventilador respectivamente, además de ello se observa que se generaron 25 filas que representan a cada una de las reglas difusas SI-ENTONCES añadidas en el programa.

Como ejemplo se puede observar que:

Regla 1: Si temperatura = 0 y humedad = 0, Entonces tiempoMotor = 0 y ventilador = 0.

Como nota importante a tomar en cuenta es que los valores de temperatura y humedad están relacionados a la diferencia que existe entre el set point y la variable de proceso, es decir que si por ejemplo se tiene seteado el set point en un valor de 28 °C y la humedad en un 55%, en el caso de que la variable de proceso tenga exactamente estos mismos valor el error tiende a cero por ende toma los valores descritos en la figura anterior.

2.3.5. Determinación del tipo de tecnología para el monitoreo de naves avícolas.

Para el envío y recepción de datos desde el servidor hacia el microcontrolador se hizo uso del estándar IEEE 802.11, el cual permite operar inalámbricamente. Debido al bajo consumo de ancho de banda usado por el microcontrolador, para subir datos hacia los servidores se seleccionó la versión 'b', ya que además de prestar un mayor alcance, permite enviar hasta un máximo de 11 Mbps que fueron suficientes para establecer la comunicación con el Raspberry PI3. En la siguiente tabla se muestra un cuadro comparativo de la estructura de los principales estándares de la familia IEEE 802.11: [30]

Tabla 16. Estructura de los principales estándares de la familia IEEE 802.11

	802.11 legacy	802.11b	802.11a	802.11g	802.11n
Banda	2.4Ghz / 850-950nm	2.4GHz	5Ghz	2.4GHz	2.4GHz / 5GHz
Capa Física	FHSS / DSSS / IR	DSSS	OFDM	DSSS / OFDM	MIMO / OFDM
Tasa Máxima	2 Mbps	11Mbps	54 Mbps	54 Mbps	600 Mbps
Alcance interior	20m	40m	35m	40m	70m
Alcance exterior	100m	150m	120m	150m	300m
Rendimiento	Buen rendimiento	Rendimiento medio	Mejor rendimiento	Rendimiento medio	Máximo rendimiento
Consumo	Bajo consumo	Bajo consumo	Mayor consumo	Bajo consumo	Bajo consumo
Canales sin solapamiento	No	3 canales simultáneos	12 canales simultáneos		
Compatibilidad		Incompatible con 802.11a	Incompatible con 802.11b	Compatible con 802.11b	Compatible con todos

Elaborado por: el investigador

2.3.6. Desarrollo de la base de datos utilizando los servidores LAMP.

Antes de empezar en la elaboración de la aplicación del proyecto web, se debió tomar en cuenta los siguientes parámetros: [52]

- Tener previamente instalado el Sistema Operativo RASPBIAN NOOBS en la Raspberry PI 3.
- Una vez instalado el Sistema Operativo, habilitar el acceso remoto, ya sea vía consola serial, telnet o SSH.

Pasos para instalar los servidores LAMP

- Acceder mediante SSH previamente consultada la dirección IP de la Raspberry Pi, ejecutando el comando “ip address” en la ventana de consola

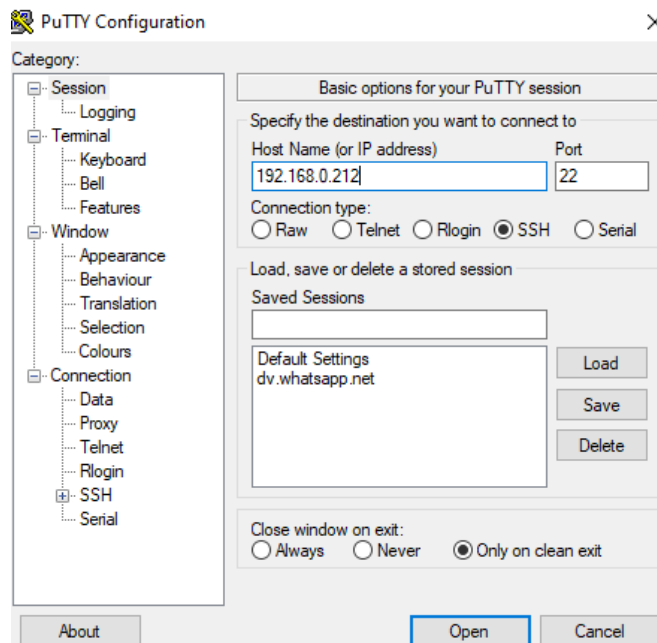


Figura 35. Ventana principal PuTTY

Elaborado por: el investigador

- Ingresar los datos de usuario y contraseña para poder acceder a la interfaz SSH de Raspberry.
- sudo su
- apt-get update

```
pi@raspberrypi: ~  
pi@raspberrypi:~ $ sudo su  
root@raspberrypi:/home/pi#  
root@raspberrypi:/home/pi# apt-get update
```

Figura 36. Comando sudo su

Elaborado por: el investigador

- apt-get upgrade

```
pi@raspberrypi: ~
root@raspberrypi:/home/pi# apt-get upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
```

Figura 37. Comando upgrade

Elaborado por: el investigador

Para fijar una IP estática seguir los siguientes pasos:

- Editar el fichero /etc/dhcpd.conf

```
pi@raspberrypi: ~
root@raspberrypi:/home/pi# nano /etc/dhcpd.conf
```

Figura 38. Ruta archivo dhcpd.conf

Elaborado por: el investigador

- Agregar al inicio del fichero los parámetros como se muestran en la imagen:

```
pi@raspberrypi: ~
GNU nano 3.2 /etc/dhcpd.conf
interface eth0
static ip_address=192.168.0.110/24
static routers=192.168.0.1
static domain_name_servers=8.8.8.8 8.8.4.4
```

Figura 39. Edición de archivo dhcpd.conf

Elaborado por: el investigador

Guardar los cambios ejecutados.

- Reiniciar con el comando init 6.
- Iniciar sesión en la terminal Putty con la nueva IP fija asignada.

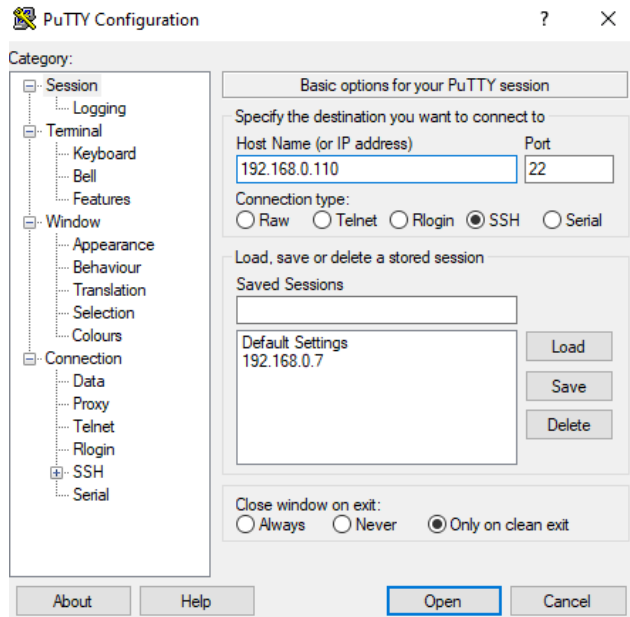


Figura 40. Ingreso con ip estática

Elaborado por: el investigador

- Sudo su
- apt-get install mariadb-server-10.0

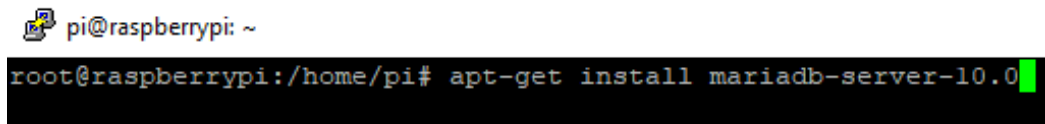


Figura 41. Comando apt-get install mariadb-server-10.0

Elaborado por: el investigador

- apt-get install python-mysqldb

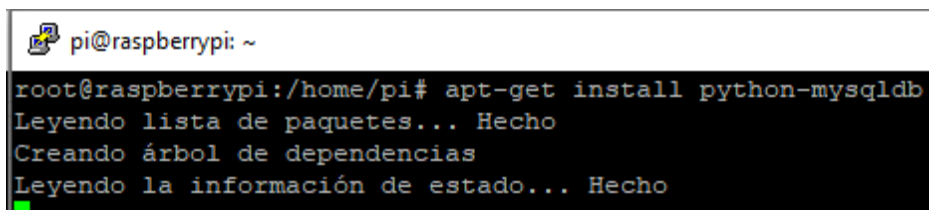


Figura 42. Comando apt-get install python-mysqldb

Elaborado por: el investigador

Para configurar el servidor phpmyadmin y acceder mediante usuario root con todos los privilegios seguir los siguientes pasos:

- dpkg-reconfigure phpmyadmin

```
pi@raspberrypi: ~  
root@raspberrypi:/home/pi# dpkg-reconfigure phpmyadmin  
Determining localhost credentials from /etc/mysql/debian.cnf: succeeded.
```

Figura 43. Comando `dpkg-reconfigure phpmyadmin`

Elaborado por: el investigador

- Método de conexión de MySQL de la base de datos por phpmyadmin: socket de unix
- Nombre de la base de datos administrativa de usuario: root
- La contraseña de la base de datos del usuario administrativo: 1234
- MySQL nombre de usuario de phpmyadmin: root
- MySQL nombre de base de datos por phpmyadmin: phpmyadmin
- Servidor Web para reconfigurar automáticamente: apache2
- Ignorar ERROR 1045
- Dpkg-reconfigure mysql-server 5.5
- Nueva contraseña para MySQL, el usuario "root": 1234
- Repita la contraseña para MySQL, el usuario "root": 1234



Figura 44. Ventana principal phpMyAdmin

Elaborado por: el investigador

En caso de existir errores al momento de ingresar al servidor phpMyAdmin, editar el archivo `‘/usr/share/phpmyadmin/libraries/sql.lib.php’` y reemplazar lo siguiente:

- Buscar: (count(\$analyzed_sql_results['select_expr']== 1)
- Reemplazar: (count(\$analyzed_sql_results['select_expr'])== 1

Creación de bases de datos:

Para poder almacenar los datos obtenidos mediante la sensorización del prototipo se creó una base de datos llamada 'tesis', como se muestra a continuación:

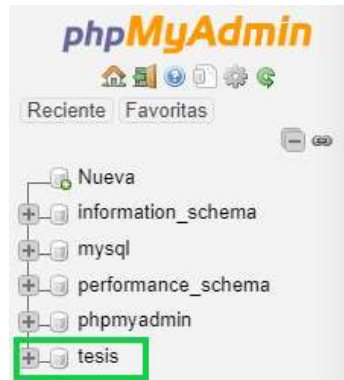


Figura 45. Base de datos tesis

Elaborado por: el investigador

Dentro de la base de datos 'tesis', se crearon tres tablas como se muestran a continuación:

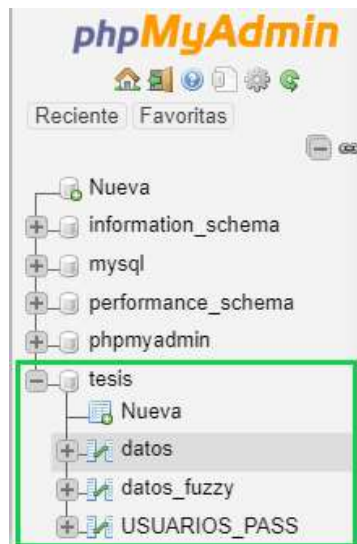
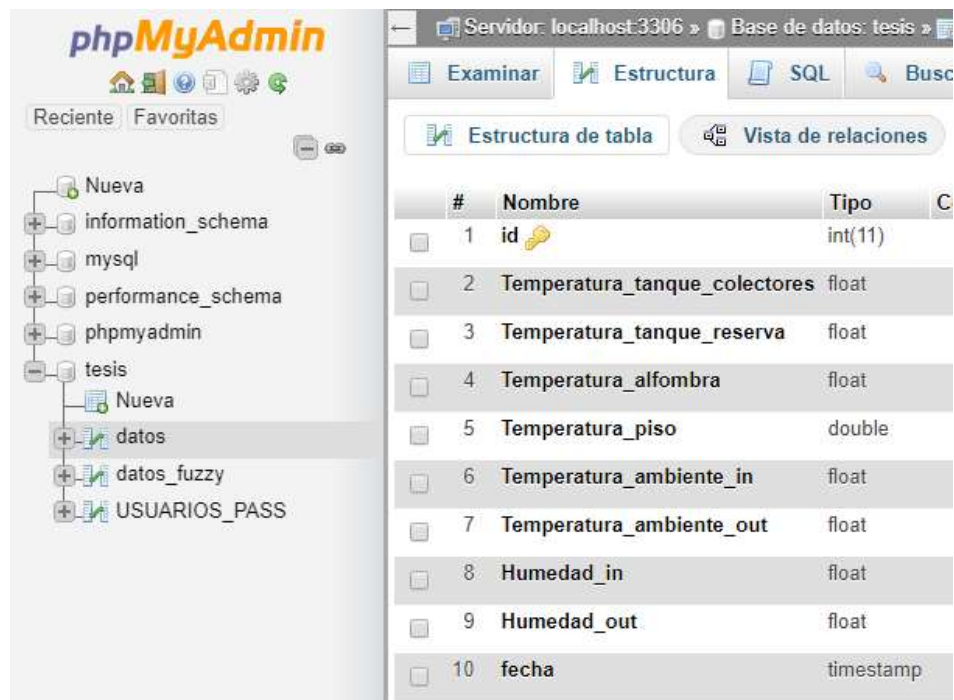


Figura 46. Tablas contenidas en base de datos tesis

Elaborado por: el investigador

La tabla ‘datos’ es la encargada de almacenar los datos recolectados por todos los sensores que intervienen el proceso de control de la temperatura, la humedad, la ventilación y el encendido o apagado del motor.

A continuación se llenó la tabla ‘datos’ con los siguientes parámetros mostrados en la figura:



#	Nombre	Tipo	C
1	id	int(11)	
2	Temperatura_tanque_colectores	float	
3	Temperatura_tanque_reserva	float	
4	Temperatura_alfombra	float	
5	Temperatura_piso	double	
6	Temperatura_ambiente_in	float	
7	Temperatura_ambiente_out	float	
8	Humedad_in	float	
9	Humedad_out	float	
10	fecha	timestamp	

Figura 47. Variables en tabla datos

Elaborado por: el investigador

La tabla ‘datos_fuzzy’ fue la encargada de almacenar los valores obtenidos a través de la sensorización de la variable de proceso y el valor de referencia o set point requerido en ese momento para el respectivo control. Posteriormente esta tabla fue utilizada para realizar una gráfica en tiempo real comparativa de la señal de referencia con la señal requerida.

La siguiente figura muestra las variables a almacenarse en la tabla ‘datos_fuzzy’:



Figura 48. Variables en tabla datos_fuzzy

Elaborado por: el investigador

Los datos necesarios para el registro dentro del sistema de monitoreo fueron almacenados dentro de la tabla 'USUARIOS_PASS', esta tabla posee la información de usuario y passwords de todos los usuarios permitidos, mostrados en la siguiente figura:

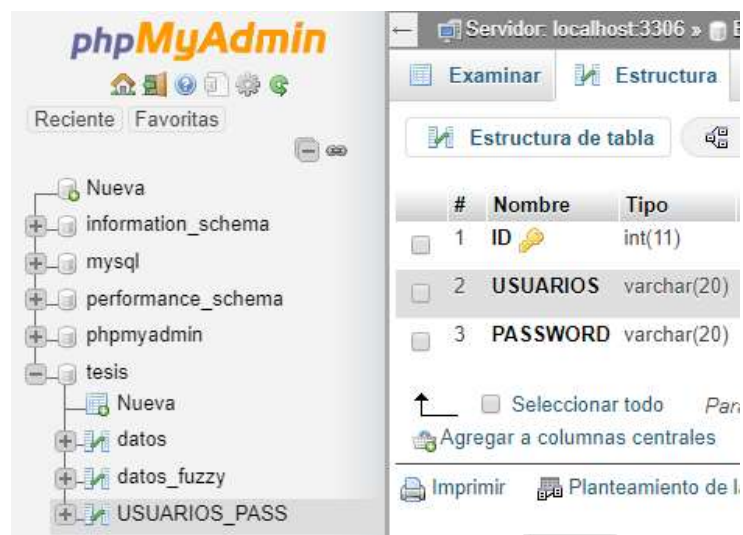


Figura 49. Variables en tabla USUARIOS_PASS

Elaborado por: el investigador

Nota: cabe mencionar que todas las variables 'id' fueron seteadas dentro de cada tabla marcadas la casilla 'AI' (Auto Increment), como se muestra en la siguiente figura:



Figura 50. Auto incremento

Elaborado por: el investigador

2.3.7. Desarrollo de interfaz Web.

Para facilitar la edición de archivos '.php' y '.html' necesarios para la ejecución de la interfaz, se habilito el protocolo FTP en la Raspberry para la transferencia de archivos editados en el terminal que corre el servicio de SSH, para ello debimos realizar los siguientes pasos detallados a continuación:

- Una vez iniciada sesión en el terminal SSH y logueados como super usuarios, se procedió a insertar el comando 'apt-get install vsftpd'.

```
pi@raspberrypi: ~  
root@raspberrypi:~# apt-get install vsftpd  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias... 50%
```

Figura 51. Comando apt-get install vsftpd

Elaborado por: el investigador

- Ingresamos al fichero vsftpd.conf

```
pi@raspberrypi: ~  
root@raspberrypi:~# nano /etc/vsftpd.conf
```

Figura 52. Ruta del fichero vsftpd.conf

Elaborado por: el investigador

- Descomentamos la línea 'write_enable=YES'

```
# Uncomment this to enable any form of FTP write command.  
write_enable=YES  
#
```

Figura 53. Configuración del fichero vsftpd.conf

Elaborado por: el investigador

- Reiniciamos el servicio con el comando 'service vsftpd restart'

A continuación se procedió a abrir el programa ‘FileZilla’ en la máquina que se está ejecutando el servicio SSH e ingresamos con los datos de usuario como se muestra a continuación:

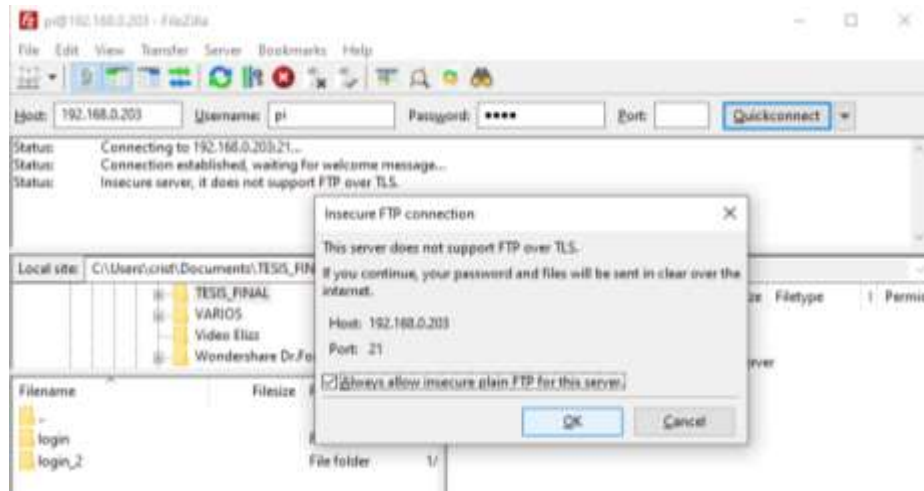


Figura 54. Ventana principal FileZilla

Elaborado por: el investigador

Si en caso de tener problemas al enviar archivos hacia el servidor en la Raspberry, simplemente cambiamos la clave por defecto del usuario PI e ingresamos nuevamente.

Se procede a conceder permisos de lectura, escritura y ejecución a la carpeta raíz contenedora de los directorios a ejecutarse mediante el servidor apache con la línea de comando ‘`chmod 777 /var/www/html/`’.

Una vez llegado a este punto el servidor está listo para mostrar cuantas páginas se desee crear y ejecutar, por lo tanto se creó una carpeta nombrada ‘tesis’ en la que contiene los archivos necesarios para la ejecución del sistema de monitoreo.

Programación para el apartado de inicio de sesión

Para iniciar sesión dentro de los servidores montados en la Raspberry pi3 se creó el archivo ‘login_w3.php’, ‘validar_w3.php’, ‘informacion_contrasena.php’, ‘cierre.php’ y la imagen ‘pollito.png’ dentro de la carpeta ‘tesis’, en la sección Anexos se encuentran completamente las líneas de código de todos los scripts.

A continuación se describe las líneas más importantes de cada archivo:

- login_w3.php

```
<button onclick="document.getElementById('id01').style.display='block'" style="width:100%;">Ingresar</button>
```

Al hacer clic sobre el botón Ingresar, inmediatamente se buscará en el documento la etiqueta 'id01' y se procede con su ejecución.

```
<div id="id01" class="modal">
<form class="modal-content animate" method="POST" action="validar_w3.php">
```

Dentro de la etiqueta 'id01' de tipo 'div' contenida en la clase 'modal' que permite abrir una ventana emergente para el ingreso de datos de autenticación que fueron enviados mediante el método 'POST' hacia el documento 'validar_w3.php' inmediatamente después de haber presionado el botón 'Login' que es de tipo 'submit'.

```
<button type="button"
onclick="document.getElementById('id01').style.display='none'"
class="cancelbtn">Cancelar</button>
```

Cuando se hace clic sobre el botón 'Cancelar', inmediatamente se oculta la clase 'modal' y regresa a la ventana principal.

```
<span class="psw">¿ Olvidaste la <a
href="informacion_contrasena.php">contraseña?</a></span>
```

Cuando el usuario no recuerda sus credenciales de ingreso, al clicar sobre el hipervínculo 'contraseña', al instante se redirige al script de información 'informacion_contrasena.php'.

- validar_w3.php

En este script se compararon los datos recibidos desde el script 'login_w3'.php que contienen los datos de usuario y contraseña. Para este procedimiento fue necesario abrir una conexión con la base de datos que contienen las listas de usuario y passwords habilitados para el acceso.

```
$base=new PDO("mysql:host=localhost; dbname=tesis","root","1234");
```

Esta sentencia habilito el acceso a la base de datos 'tesis' ubicada dentro del local host.

```
$sql="SELECT * FROM USUARIOS_PASS WHERE USUARIOS= :login AND
PASSWORD= :password";
```


Una vez dentro de la base de datos, se busco la tabla ‘USUARIOS_PASS’ y se guardaron los ‘USUARIOS’ en la variable ‘login’ y los ‘PASSWORDS’ en la variable ‘password’ que fue necesario para posteriormente compararlos

```
$login=htmlentities(addslashes($_POST["nombre"]));  
$password=htmlentities(addslashes($_POST["password"]));
```

Las variables ‘nombre’ y ‘password’ obtenidas desde el script ‘login_w3.php’ fueron almacenadas en las variables ‘\$login’ y ‘\$password’ respectivamente.

```
$resultado->bindValue(":login", $login);  
$resultado->bindValue(":password", $password);  
$resultado->execute();  
$numero_registro=$resultado->rowCount();
```

Estas líneas de código compararon fila por fila dentro de la base de datos si coincidieron los datos ingresados con algún registro de UDUARIO y PASSWORD almacenándolos en la base de datos.

```
if($numero_registro!=0){  
    session_start();  
    $_SESSION["usuario"]=$_POST["nombre"];  
    header("location:datos_folder/todo_datos.php");  
    }else{  
        header("location:login_w3.php");  
    }
```

Si en alguna fila coincidieron tanto USUARIOS como PASSWORDS comparados con los datos enviados desde el script de autenticación, se inicia sesión y se almacena el nombre del Usuario en una variable POST para posteriormente ser utilizada a conveniencia dentro del script ‘datos_folder/todo_datos.php’, el mismo que se abre si o si cumple esta condición.

En caso de no cumplir la condición, se vuelve a abrir el script ‘location:login_w3.php’ para reingresar nuevamente usuarios y contraseñas válidos.

- **informacion_contrasena.php**

```
<h2>Para poder recuperar la contraseña, por favor comunícate con el  
administrador de la página.<br>Gracias</h2>  
<h2>Fernando Monta</h2>  
<h2>Tel: 0969317448</h2>  
<a href="login_w3.php" class="button">Regresar</a>
```

Este script muestra información necesaria para contactarse con el administrador de la página, el cuál proveerá de ayuda al usuario para poder ingresar al sistema.

Para salir de la página y regresar a la página de ingreso, es necesario clicar sobre el botón Regresar.

- **cierre.php**

```
<?php
session_start();
session_destroy();
header("location:login_w3.php");
?>
```

Para cerrar sesión y que nuestras datos de ingreso no se queden guardados, se procede a destruir la sesión, este script se ejecuta únicamente cuando el usuario presiona sobre el botón ‘Cerrar sesión’ que se mencionará en los siguientes apartados.

Programación para el apartado de subida de datos al servidor

Para subir los datos obtenido mediante sensores en el prototipo dentro de los servidores montados en la Raspberry pi3 se crearon los archivos ‘abrir_conexion.php’, ‘cerrar_conexion.php’, ‘fuzzy_upload.php’ y ‘temperatura.php’ dentro de la carpeta ‘uploads’ contenida en la carpeta ‘tesis’, en la sección Anexos se encuentran completamente las líneas de código de todos los scripts.

A continuación se describe las líneas más importantes de cada archivo:

- **abrir_conexion.php**

```
$host = "localhost";
$user = "root";
$pass = "1234";
$db = "tesis";
$conexion = mysqli_connect($host,$user,$pass,$db);
```

En este script se declara en variables los datos de autenticación que permitieron ingresar hacia la base de datos en el servidor phpmyadmin para posteriormente generar la conexión de ingreso hacia los mismos.

- **cerrar_conexion.php**

```
mysqli_close($conexion);
```

Este script únicamente cierra la conexión abierta en la base de datos.

- **fuzzy_upload.php**

```
require("abrir_conexion.php");
$temp_mysql = mysqli_real_escape_string($conexion, $_GET['set_point']);
$temp_mysql2 = mysqli_real_escape_string($conexion, $_GET['process_value']);
$temp_in = "INSERT INTO datos_fuzzy (set_point, process_value)
VALUES('".$temp_mysql."','".$temp_mysql2."')";
mysqli_query($conexion, $temp_in);
require("cerrar_conexion.php");
```

En este script, primeramente se abrió la conexión hacia la base de datos llamando al archivo 'abrir_conexion.php', luego se guardaron las variables obtenidas con la función 'GET' desde el microcontrolador para posteriormente enviarlas a la tabla 'datos_fuzzy' y finalmente se cierre la conexión invocando al script 'cerrar_conexion.php'.

- **temperatura.php**

```
$temp_in = "INSERT INTO datos (Temperatura_tanque_colectores,
Temperatura_tanque_reserva, Temperatura_alfombra, Temperatura_piso,
Temperatura_ambiente_in, Temperatura_ambiente_out, Humedad_in,
Humedad_out)
VALUES('".$temp_mysql."','".$temp_mysql2."','".$temp_mysql3."','".$temp_mysql4."
'."','".$temp_mysql5."','".$temp_mysql6."','".$temp_mysql7."','".$temp_mysql8."')";
```

Al igual que en el script anterior, este envía datos hacia la tabla 'datos'.

Programación para el apartado de muestra de datos desde el servidor

Para mostrar los datos obtenidos mediante sensores en el prototipo dentro de los servidores montados en la Raspberry pi3 se crearon los archivos 'DatoSensores.php', 'DatosFuzzy.php', 'todo_datos.php' y el archivo 'jquery-1.9.0.min.js' contenido dentro de la carpeta 'jquery', todos estos scripts están dentro de la carpeta 'datos_folder' contenida en la carpeta 'tesis', en la sección Anexos se encuentran completamente las líneas de código de todos los scripts.

A continuación se describe las líneas más importantes de cada archivo:

- **DatoSensores.php**

```
$pdo=new PDO("mysql:dbname=tesis;host=localhost","root","1234");
```

Esta línea de código nos permitió acceder desde una aplicación diferente a PHP a los datos que se almacenen dentro del servidor mysql, en este caso la aplicación que acceda a ella fue de tipo JSON.

- **DatosFuzzy.php**

Al igual que el script anterior, en este se hizo uso de una aplicación tipo JSON para posteriormente acceder a los datos almacenados en la base de datos y mostrarlos desde cualquier servidor.

- **todo_datos.php**

```
session_start();  
if(!isset($_SESSION["usuario"])){  
    header("location:/tesis/login_w3.php");  
}
```

Este script fue el principal, ya que muestra la información obtenida desde la base de datos. Para hacer esto posible, primeramente hace un inicio de sesión, cabe recalcar que a este archivo es accesible siempre y cuando el usuario haya iniciado sesión en la misma.

2.3.8. Programación del sistema de control difuso usando hardware y software libre

Para el desarrollo de un correcto control difuso se utilizó el método de Mamdani, el cual fue programado con 25 reglas difusas, las cuales fueron obtenidas a través de la experiencia de un operario conocedor del funcionamiento de la planta.

Las reglas difusas con relación a las funciones de membresía detalladas anteriormente fueron las siguientes:

1. siErrorNegativoGrandeANDHumedadNegativoGrande,
entoncesVoltajeNegativoGrandeANDBitsNegativoGrande
2. siErrorNegativoGrandeANDHumedadNegativoPequeno,
entoncesVoltajeNegativoGrandeANDBitsNegativoPequeno
3. siErrorNegativoGrandeANDHumedadCero,
entoncesVoltajeNegativoGrandeANDBitsCero
4. siErrorNegativoGrandeANDHumedadPositivoPequeno,
entoncesVoltajeNegativoGrandeANDBitsPositivoPequeno
5. siErrorNegativoGrandeANDHumedadPositivoGrande,
entoncesVoltajeNegativoGrandeANDBitsPositivoGrande

6. siErrorNegativoPequenoANDHumedadNegativoGrande, entoncesVoltajeNegativoPequenoANDBitsNegativoGrande
7. siErrorNegativoPequenoANDHumedadNegativoPequeno, entoncesVoltajeNegativoPequenoANDBitsNegativoPequeno
8. siErrorNegativoPequenoANDHumedadCero, entoncesVoltajeNegativoPequenoANDBitsCero
9. siErrorNegativoPequenoANDHumedadPositivoPequeno, entoncesVoltajeNegativoPequenoANDBitsPositivoPequeno
10. siErrorNegativoPequenoANDHumedadPositivoGrande, entoncesVoltajeNegativoPequenoANDBitsPositivoGrande
11. siErrorCeroANDHumedadNegativoGrande, entoncesVoltajeCeroANDBitsNegativoGrande
12. siErrorCeroANDHumedadNegativoPequeno, entoncesVoltajeCeroANDBitsNegativoPequeno
13. siErrorCeroANDHumedadCero, entoncesVoltajeCeroANDBitsCero
14. siErrorCeroANDHumedadPositivoPequeno, entoncesVoltajeCeroANDBitsPositivoPequeno
15. siErrorCeroANDHumedadPositivoGrande, entoncesVoltajeCeroANDBitsPositivoGrande
16. siErrorPositivoPequenoANDHumedadNegativoGrande, entoncesVoltajePositivoPequenoANDBitsNegativoGrande
17. siErrorPositivoPequenoANDHumedadNegativoPequeno, entoncesVoltajePositivoPequenoANDBitsNegativoPequeno
18. siErrorPositivoPequenoANDHumedadCero, entoncesVoltajePositivoPequenoANDBitsCero
19. siErrorPositivoPequenoANDHumedadPositivoPequeno, entoncesVoltajePositivoPequenoANDBitsPositivoPequeno
20. siErrorPositivoPequenoANDHumedadPositivoGrande, entoncesVoltajePositivoPequenoANDBitsPositivoGrande
21. siErrorPositivoGrandeANDHumedadNegativoGrande, entoncesVoltajePositivoGrandeANDBitsNegativoGrande
22. siErrorPositivoGrandeANDHumedadNegativoPequeno, entoncesVoltajePositivoGrandeANDBitsNegativoPequeno
23. siErrorPositivoGrandeANDHumedadCero, entoncesVoltajePositivoGrandeANDBitsCero
24. siErrorPositivoGrandeANDHumedadPositivoPequeno, entoncesVoltajePositivoGrandeANDBitsPositivoPequeno
25. siErrorPositivoGrandeANDHumedadPositivoGrande, entoncesVoltajePositivoGrandeANDBitsPositivoGrande

Para llevar a cabo la toma de decisiones en dependencia de un valor de referencia y un valor de proceso, se utilizó el microcontrolador WemosD1mini implementando el código en el IDE de Arduino. En la sección de anexos se adjunta toda la programación del sistema de control.

- Programación de sensores en Arduino

Para llevar a cabo este proceso, primeramente fue necesario instalarse las librerías necesarias para el correcto funcionamiento de los sensores, luego se declaran los pines de entrada y de salida y se inicializan las variables a utilizar.

```
float* temperatura_ds18b20(){
  float* array_temp = new float[4];
  sensors.requestTemperatures(); //envía el comando para obtener las temperaturas
  array_temp[0] = sensors.getTempC(address1);//Se obtiene la temperatura en °C del
  sensor 1 Temperatura Tanque Colectores
  array_temp[1] = sensors.getTempC(address2);//Se obtiene la temperatura en °C del
  sensor 2 Temperatura Tanque Reserva
  array_temp[2] = sensors.getTempC(address3);//Se obtiene la temperatura en °C del
  sensor 3 Temperatura Alfombra
  array_temp[3] = sensors.getTempC(address4);//Se obtiene la temperatura en °C del
  sensor 4 Temperatura Piso
  return array_temp;

  float* dht11(){
  float* array_dht11 = new float[4];
  array_dht11[0] = dht.readHumidity(); ///EN PORCENTAJE Humedad In
  array_dht11[1] = dht.readTemperature(); ///EN °C Temperatura Ambiente In
  array_dht11[2] = dht2.readHumidity(); ///EN PORCENTAJE Humedad out
  array_dht11[3] = dht2.readTemperature(); ///EN °C Temperatura Ambiente out
  return array_dht11;
  }
}
```

- Programación de envío de datos sensados hacia el servidor

Este fue el método encargado de enviar datos desde los sensores hacia el servidor, los mismos son enviados utilizando el método GET, comunicándose a través del puerto 80.

```
void connectToServer2(String variable1, String variable2, double valor_1, double
valor_2) {
  if (client.connect(host, 80)) {
  //Serial.println("making HTTP request...");
  //Lo interesante, apuntar a la página por donde mando los datos
  //En este caso apuntar a temperatura.php OJO
  client.print("GET /tesis/uploads/fuzzy_upload.php?" + variable1 + "=");
  client.print(valor_1);
  client.print("&" + variable2 + "=");
  client.print(valor_2);
  client.println(" HTTP/1.1");
  client.println("HOST: " + (String)host);
  client.println();
}
```

```

}
// note the time of this connect attempt:
lastAttemptTime = millis();
client.stop();
client.flush();
}

```

- Programación de reglas difusas

Las siguientes líneas de código son las encargadas de ejecutar las funciones de control difuso por cada regla difusa insertada, es decir, 25 reglas difusas, cada una con su antecedentes y consecuentes como se muestra a continuación.

```

FuzzyRuleAntecedent* siErrorNegativoGrandeANDHumedadNegativoGrande =
new FuzzyRuleAntecedent();
siErrorNegativoGrandeANDHumedadNegativoGrande -> joinWithAND(ENG,
HNG); FuzzyRuleConsequent*
entoncesVoltajeNegativoGrandeANDBitsNegativoGrande = new
FuzzyRuleConsequent();
entoncesVoltajeNegativoGrandeANDBitsNegativoGrande -> addOutput(VNG);
entoncesVoltajeNegativoGrandeANDBitsNegativoGrande -> addOutput(BNG);
FuzzyRule* reglaDifusa1 = new FuzzyRule(1,
siErrorNegativoGrandeANDHumedadNegativoGrande,
entoncesVoltajeNegativoGrandeANDBitsNegativoGrande);
fuzzy -> addFuzzyRule(reglaDifusa1);

```

2.3.9. Selección del sistema de piso radiante óptimo para naves avícolas.

El sistema de calefacción electo debido al gran índice de radiación solar emitido en el Ecuador es el tipo de suelo radiante por agua, ya que se trata de un sistema que impulsa agua caliente a través de tuberías empotradas en el piso, generando así calor, el mismo que es absorbido por el piso y luego emitido hacia el ambiente en forma de energía radiante, siendo así un sistema con una buena inercia térmica.

Dentro del sistema solar de suelo radiante, se obtuvo el siguiente análisis matemático energético, detallado a continuación:

El índice de calor emitido por el sol se obtiene midiendo el área de absorción del colector térmico y la radiación solar disponible en un tiempo.

$$Constante\ Solar = \frac{potencia\ producida\ por\ el\ sol}{superficie\ esférica}$$

$$\text{Constante Solar} = \frac{3.78 \times 10^{26} \text{ W}}{4 * \pi * r^2}$$

$$\text{Constante Solar} = \frac{3.78 \times 10^{26} \text{ W}}{4 * \pi * (149.000.000.000 \text{ m})^2}$$

$$\text{Constante Solar} = \frac{3.78 \times 10^{26} \text{ W}}{4 * \pi * (149 \times 10^9 \text{ m})^2}$$

$$\text{Constante Solar} = \frac{1350 \text{ W}}{\text{m}^2}$$

$$\text{calor disponible} = \text{radiación solar} * \text{area}$$

$$\text{calor disponible} = \frac{1350 \text{ W}}{\text{m}^2} * (2 * \pi * r * h + 2 * \pi * r * h)$$

$$\text{calor disponible} = 2040 \text{ W}$$

Para saber cuál es el calor que obtiene el suelo radiante desde el colector solar se utiliza la siguiente ecuación:

$$Q_{ag} = m_{ag} * Cp_{ag} * \Delta T$$

Donde Q_{ag} es el calor del agua, m_{ag} es el flujo másico del agua, Cp_{ag} es el calor específico del agua y ΔT es la variación de la temperatura del agua. Para calcular el flujo másico del agua se utiliza la siguiente expresión matemática:

$$m_{ag} = \rho * \frac{V}{t}$$

Donde ρ simboliza la densidad del agua, V es el volumen del agua y t es el tiempo.

Por lo tanto la ecuación para encontrar el calor del agua viene dada por la siguiente expresión:

$$Q_{ag} = \rho * \frac{V}{t} * Cp_{ag} * \Delta T$$

$$Q_{ag} = \frac{997 \text{ Kg}}{\text{m}^3} * \frac{0.8 \text{ m}^3}{3600 \text{ s}} * \frac{4.186 \text{ KJ}}{\text{Kg}^\circ\text{C}} * 10^\circ\text{C}$$

$$Q_{ag} = \frac{9.27 \text{ KJ}}{\text{s}}$$

Ahora, para calcular el calor que se obtuvo en el piso radiante se utilizó la siguiente ecuación:

$$Q_{air} = m_{air} * Cp_{air} * \Delta T$$

Donde Q_{air} es el calor del aire que se calienta en el suelo radiante, m_{air} es el flujo másico del aire que genera al circular a través del piso radiante, Cp_{air} es el calor específico del aire y ΔT es la variación de la temperatura del ambiente al no circular agua y al circular agua. Para calcular el flujo másico del aire se utilizó la siguiente expresión matemática:

$$m_{air} = \rho * Vel * A$$

Donde ρ simboliza la densidad del aire, V es la velocidad del agua a través de la tubería del piso radiante y A es el área a enfriarse mediante el piso radiante.

Por lo tanto la ecuación para encontrar el calor del aire viene dada por la siguiente expresión

$$Q_{air} = \rho * Vel * A * Cp_{air} * \Delta T$$

$$Q_{air} = \rho * Vel * A * Cp_{air} * \Delta T$$

$$Q_{ag} = \frac{1.225Kg}{m^3} * \frac{1.5m}{s} * 0.96m^2 * \frac{1.009KJ}{Kg^{\circ}C} * 5^{\circ}C$$

$$Q_{ag} = \frac{8.8993KJ}{s}$$

Finalmente la eficiencia térmica del sistema de calefacción por piso radiante utilizando agua con energía solar se evaluó en la siguiente ecuación:

$$Eff = \frac{Q_{air}}{Q_{ag}}$$

$$Eff = \frac{\frac{8.8993KJ}{s}}{\frac{9.27KJ}{s}}$$

$$Eff = 0.96$$

$$Eff = 0.96 * 100$$

$$Eff = 96\%$$

2.3.10. Implementación del prototipo

Con la ayuda del software Proteus, se realizó el diseño de la placa que contiene los dispositivos electrónicos, tales como entradas y salidas de sensores y actuadores,

dispositivo controlador de velocidad de ventilador y el microcontrolador como tal. En la siguiente figura se aprecia de manera clara y precisa cómo está distribuido el circuito para el presente proyecto:

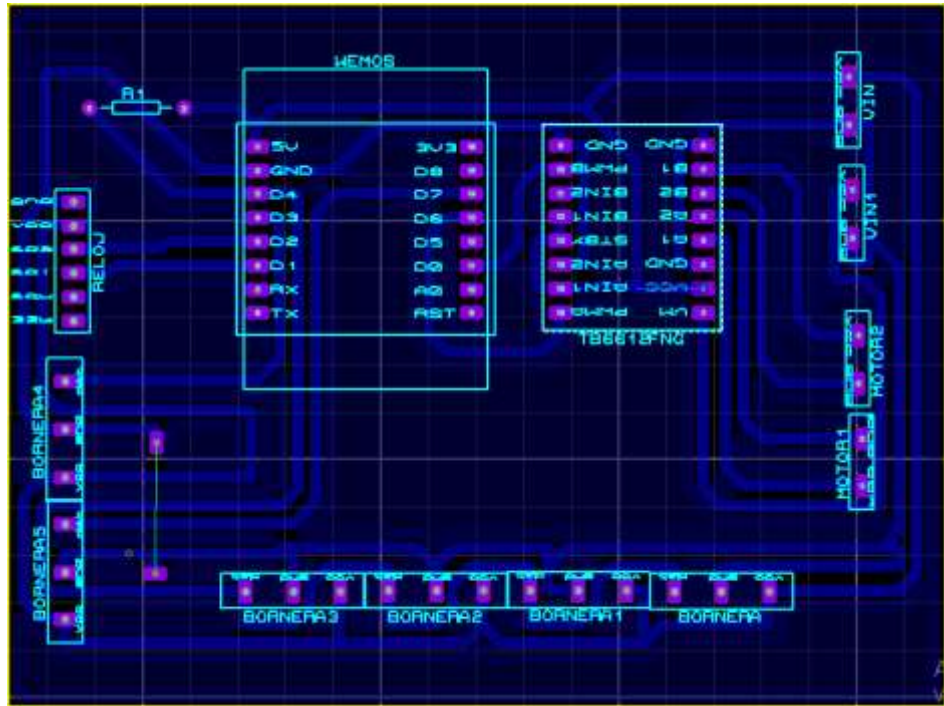


Figura 55. Circuito impreso en baquelita.

Elaborado por: el investigador

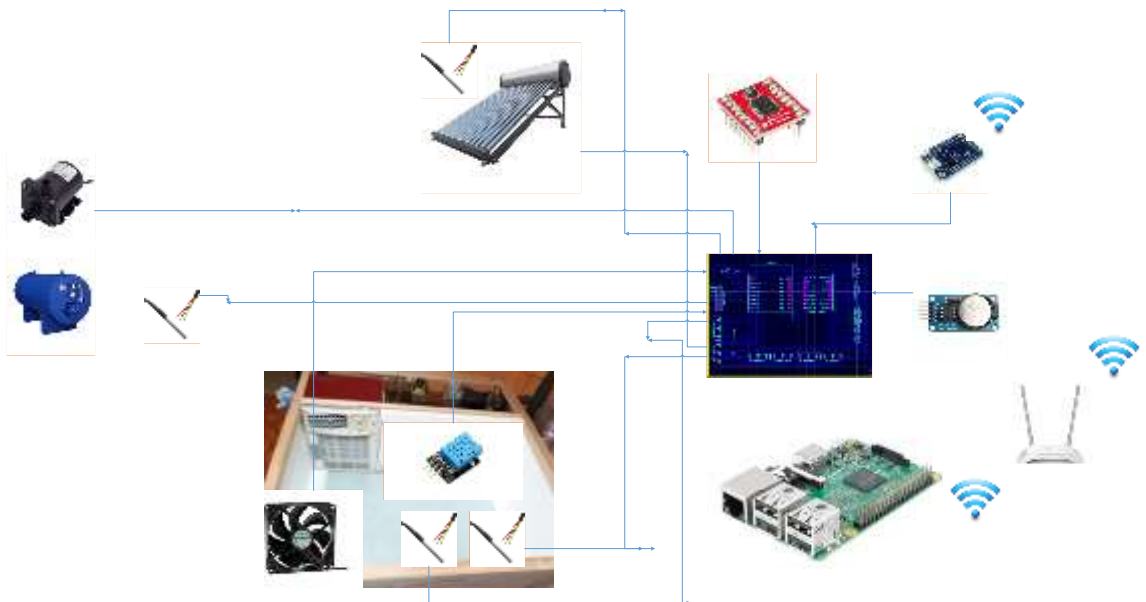


Figura 56. Sistema de control difuso y monitoreo.

Elaborado por: el investigador

Para la implementación de la placa en baquelita, primero se imprimió el circuito en papel foto, para luego aplicarlo sobre la placa de cobre y poder obtener las pistas, mediante una plancha., como se puede observar en la figura:

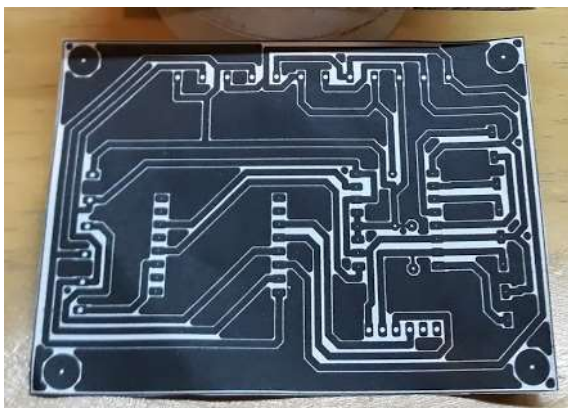


Figura 57. Impresión a láser del circuito.

Elaborado por: el investigador

Luego de realizar el planchado del circuito por un tiempo prolongado se procedió a retirar el papel de la baquelita, y a limpiar la placa, a continuación se colocó el ácido en un recipiente plástico, donde se sumergió a la placa, moviéndola continuamente de un lado hacia otro, para que se produzca la reacción química y obtener las pistas del circuito.

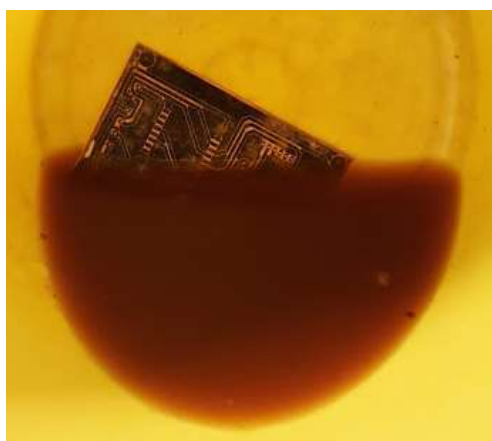


Figura 58. Placa en baquelita.

Elaborado por: el investigador

Finalmente se realizó las perforaciones con un taladro, para ubicar los componentes, y luego soldarlos, teniendo así una placa final como la que se muestra a continuación:

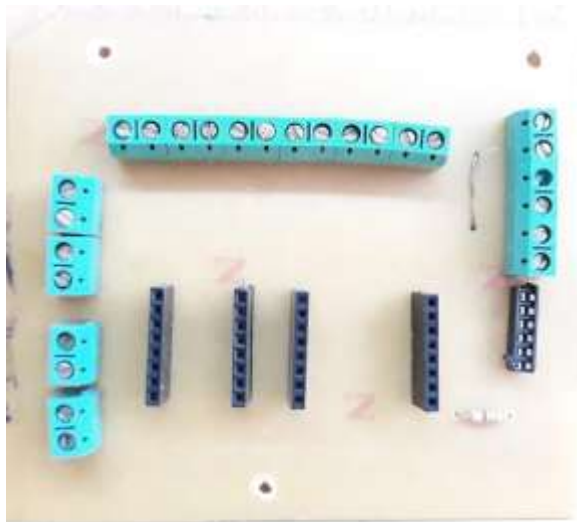


Figura 59. Placa final.

Elaborado por: el investigador

2.3.11. Pruebas del prototipo

Prueba de comunicación entre equipos con la red interna LAN.

A continuación se procedió a mostrar mediante comando ping la comunicación entre la tarjeta WemosD1Mini y los servidores montados en la Raspberry PI3.

La siguiente tabla muestra las direcciones IP asignadas estáticamente a los dispositivos:

Tabla 17. Direccionamiento IP

Dispositivo	Dirección IP	Máscara de subred
Wemos D1 Mini	192.168.0.111	255.255.255.0
Raspberry PI 3	192.168.0.110	255.255.255.0
Computador SSH	192.168.0.208	255.255.255.0

Elaborado por: el investigador

En la siguiente figura se muestra la conectividad que existe entre el equipo terminal y la Wemos D1 mini:

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.18362.535]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\HP>ping 192.168.0.111

Haciendo ping a 192.168.0.111 con 32 bytes de datos:
Respuesta desde 192.168.0.111: bytes=32 tiempo=59ms TTL=255
Respuesta desde 192.168.0.111: bytes=32 tiempo=282ms TTL=255
Respuesta desde 192.168.0.111: bytes=32 tiempo=197ms TTL=255
Respuesta desde 192.168.0.111: bytes=32 tiempo=7ms TTL=255

Estadísticas de ping para 192.168.0.111:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 7ms, Máximo = 282ms, Media = 136ms

C:\Users\HP>
```

Figura 60. Prueba de conexión entre el equipo terminal y el microcontrolador

Elaborado por: el investigador

La siguiente imagen, muestra la conexión existente entre el equipo terminal y la RaspberryPI3:

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.18362.535]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\HP>ping 192.168.0.110

Haciendo ping a 192.168.0.110 con 32 bytes de datos:
Respuesta desde 192.168.0.110: bytes=32 tiempo=3ms TTL=64
Respuesta desde 192.168.0.110: bytes=32 tiempo=4ms TTL=64
Respuesta desde 192.168.0.110: bytes=32 tiempo=16ms TTL=64
Respuesta desde 192.168.0.110: bytes=32 tiempo=3ms TTL=64

Estadísticas de ping para 192.168.0.110:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 3ms, Máximo = 16ms, Media = 6ms

C:\Users\HP>
```

Figura 61. Prueba de conexión entre el equipo terminal y Raspberry PI 3

Elaborado por: el investigador

La imagen siguiente muestra la conexión mediante terminal SSH entre el equipo servidor Raspberry PI3 y la tarjeta microcontroladora Wemos D1 mini:

```
pi@raspberrypi: ~  
pi@raspberrypi:~$ ping 192.168.0.111  
PING 192.168.0.111 (192.168.0.111) 56(84) bytes of data.  
64 bytes from 192.168.0.111: icmp_seq=1 ttl=255 time=28.9 ms  
64 bytes from 192.168.0.111: icmp_seq=2 ttl=255 time=40.6 ms  
64 bytes from 192.168.0.111: icmp_seq=3 ttl=255 time=25.8 ms  
64 bytes from 192.168.0.111: icmp_seq=4 ttl=255 time=45.3 ms  
64 bytes from 192.168.0.111: icmp_seq=5 ttl=255 time=56.6 ms  
64 bytes from 192.168.0.111: icmp_seq=6 ttl=255 time=68.3 ms  
^C  
--- 192.168.0.111 ping statistics ---  
6 packets transmitted, 6 received, 0% packet loss, time 13ms  
rtt min/avg/max/mdev = 25.787/44.250/68.318/14.851 ms  
pi@raspberrypi:~$
```

Figura 62. Prueba de conexión entre la Raspberry PI3 y el microcontrolador.

Elaborado por: el investigador

Prueba de envío de resultados a la base de datos

Para realizar esta prueba se procedió a simular el link que genera el código programado en la tarjeta microcontroladora enviando este dato mediante puerto 80. En la siguiente figura se muestra el valor enviado y el valor recibido en la base de datos:

192.168.0.110/tesis/uploads/temperatura.php?Temperatura_tanque_colectores=2&Temperatura_tanque_reserva=3&Temperatura_alfombra=4&Temperatura_piso=5&Temperatura_ambiente_in=6&Temperatura_ambiente_out=7&Humedad_in=8&Humedad_out=9

ID	Temperatura_tanque_colectores	Temperatura_tanque_reserva	Temperatura_alfombra	Temperatura_piso	Temperatura_ambiente_in	Temperatura_ambiente_out	Humedad_in	Humedad_out
10270	2	3	4	5	6	7	8	9

Figura 63. Prueba de envío de datos hacia servidor

Elaborado por: el investigador

Prueba de inicio de sesión

Para poder ingresar al sistema, se debió introducir el link http://192.168.0.110/tesis/login_w3.php que redigirá a la interface que se muestra en la siguiente figura:



The image shows a login form with the following elements:

- Header: **Bienvenidos**
- Instruction: **Para poder ingresar por favor da clic en el botón de abajo e ingresa tu**
- Form fields:
 - Usuario:** Input field with placeholder text "Ingresar Usuario".
 - Contraseña:** Input field with placeholder text "Ingresar Contraseña".
- Buttons:
 - A green button labeled **Login**.
 - A red button labeled **Cancelar**.
- Additional features:
 - A checkbox labeled **Recordarme**.
 - A link labeled **¿Olvidaste la contraseña?**.

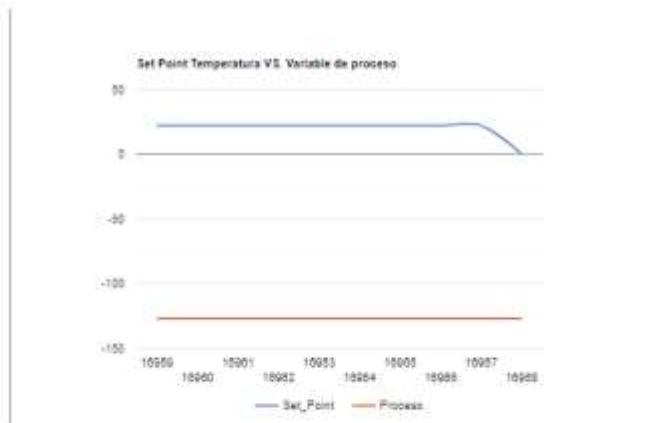
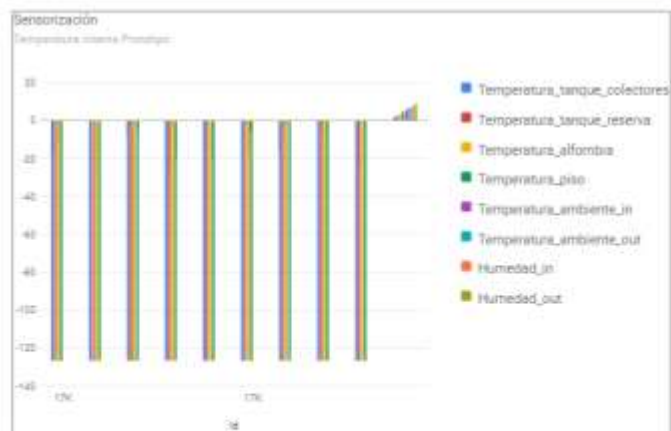
Figura 64. Inicio de sesión

Elaborado por: el investigador

Si el usuario ingresó correctamente las credenciales de inicio de sesión, pudo ingresar y visualizar los datos sensados en tiempo real, caso contrario se quedará visualizando la pantalla de inicio de sesión. En la siguiente figura se muestra el script ejecutado una vez iniciado correctamente sesión:

Bienvenido Nando

Últimos datos sensados	
Temperatura_tanque_colectores =	2
Temperatura_tanque_reserva =	3
Temperatura_alfombra =	4
Temperatura_piso =	5
Temperatura_ambiente_in =	6
Temperatura_ambiente_out =	7
Humedad_in =	8
Humedad_out =	9
Fecha =	2020-01-15 03:58:49
Contador =	16979



Cerrar sesión

Figura 65. Ventana principal inicio de sesión

Elaborado por: el investigador

2.3.12. Presupuesto

Tabla 18. Presupuesto

PRESUPUESTO		
MATERIALES	UNIDAD	COSTO (DÓLARES)
Raspberry PI3	1	\$65.00
Wemos D1 mini	1	\$10.00
Bomba de agua 800l/h	1	\$20.00
Sensor DS18B20	4	\$5.00
Sensor DHT11	2	\$3.00
Driver TB6612FNG	1	\$5.00
Router TL-WA901ND	1	\$20
Ventilador 12V	1	\$6
RTC DS323	1	\$5
Calentador de Agua Solar	1	\$300
Tubería pvc ½ “	10m	\$30
Conectores para tubería	50	\$30
Serpentín suelo radiante	80m	\$40
Baquelita	1	\$3
Ácido Férrico	1	\$1
Espadines	4	\$2
Borneras de 2 entradas	20	\$6
Prototipo a escala de madera	1	\$100
Aislación térmica de prototipo	1	\$10
Cortapico	1	\$6
Fuente 12V	2	\$3
TOTAL		\$670

Fuente: Tablaa diseñada por el investigador

CAPÍTULO III

RESULTADOS Y DISCUSIÓN

3.1. Análisis y discusión de los resultados

De un total de seis días de ejecución, cada uno representando una valor de temperatura y humedad diferente dentro de los valores establecidos para la crianza de pollos, comparados con la variable de proceso se obtuvieron los siguientes resultados:

- Día 1

Variable de set point de Temperatura = 31 °C

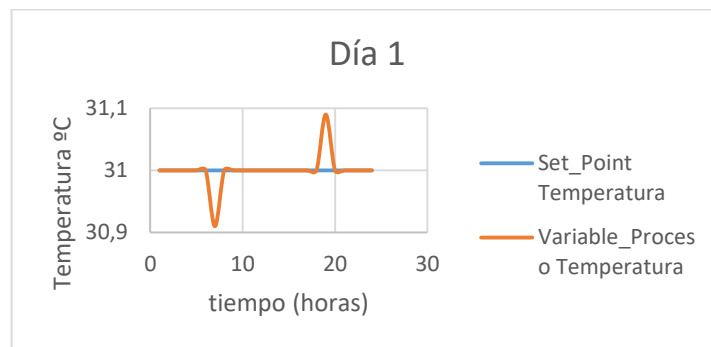


Figura 66. Resultado día 1, SP Vs. PV Temperatura

Elaborado por: el investigador

Variable de set point de Humedad = 37.5%



Figura 67. Resultado día 1, SP Vs. PV Humedad

Elaborado por: el investigador

- **Día 2**

Variable de set point de Temperatura = 29.5 °C

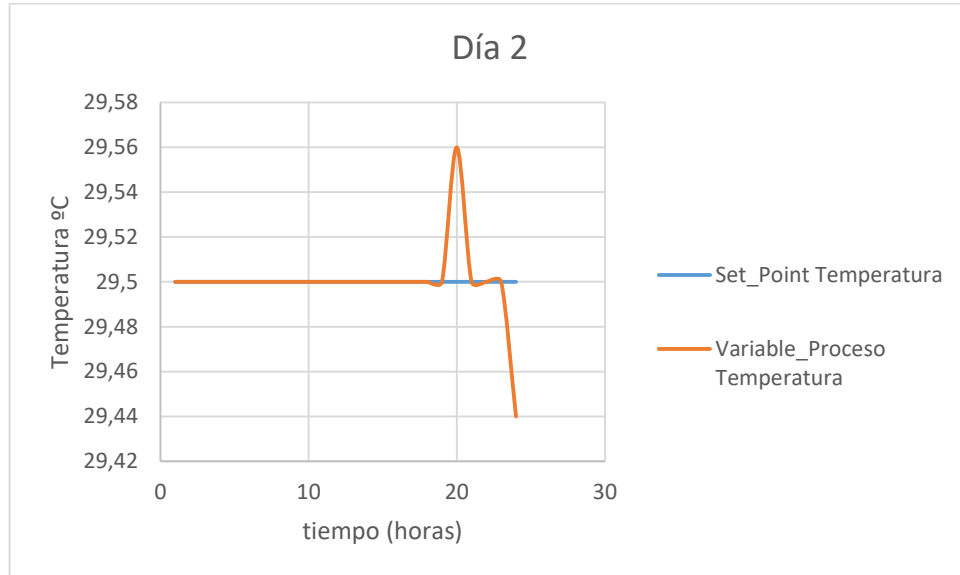


Figura 68. Resultado día 2, SP Vs. PV Temperatura

Elaborado por: el investigador

Variable de set point de Humedad = 40%

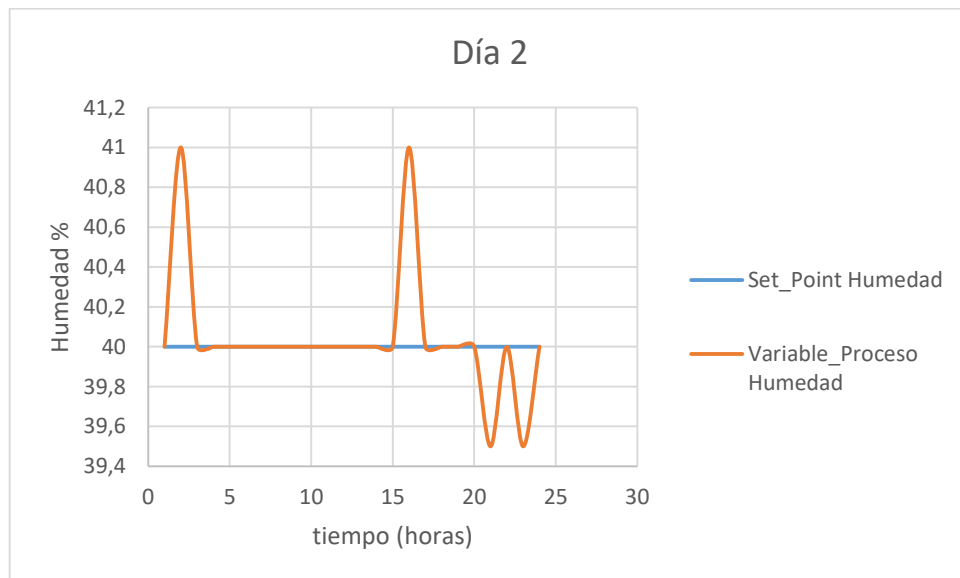


Figura 69. Resultado día 2, SP Vs. PV Humedad

Elaborado por: el investigador

- **Día 3**

Variable de set point de Temperatura = 28 °C

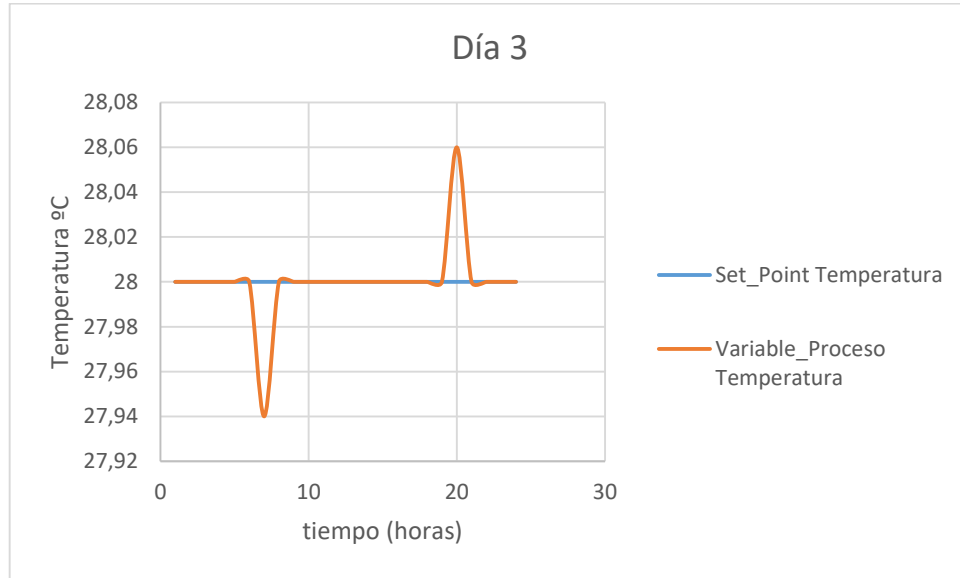


Figura 70. Resultado día 3, SP Vs. PV Temperatura

Elaborado por: el investigador

Variable de set point de Humedad = 42.5%

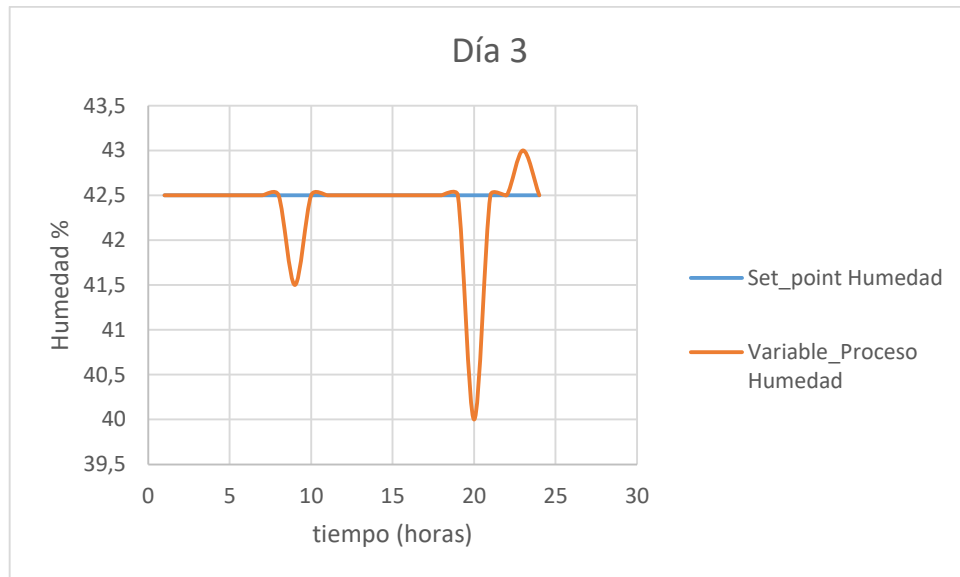


Figura 71. Resultado día 3, SP Vs. PV Humedad

Elaborado por: el investigador

- **Día 4**

Variable de set point de Temperatura = 26 °C

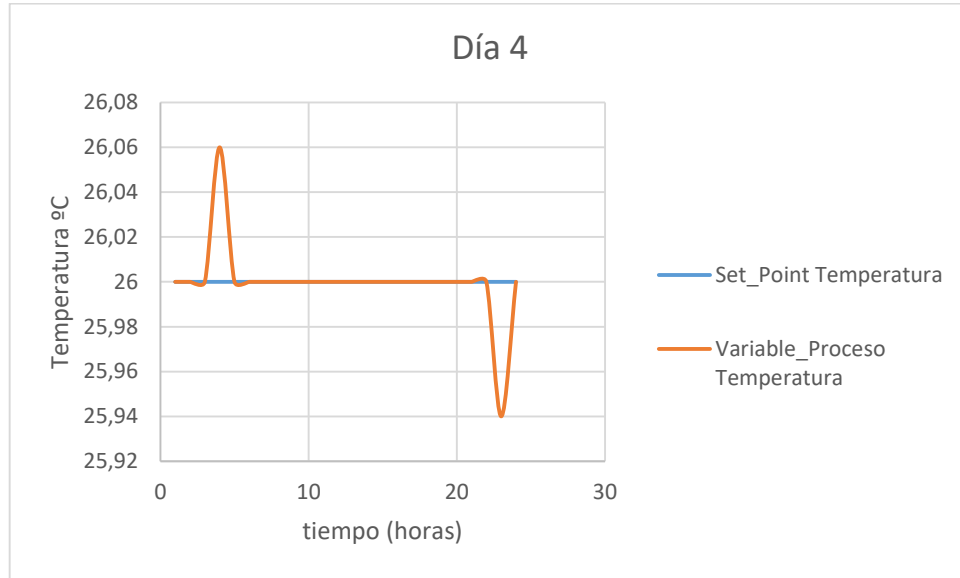


Figura 72. Resultado día 4, SP Vs. PV Temperatura

Elaborado por: el investigador

Variable de set point de Humedad = 47.5%

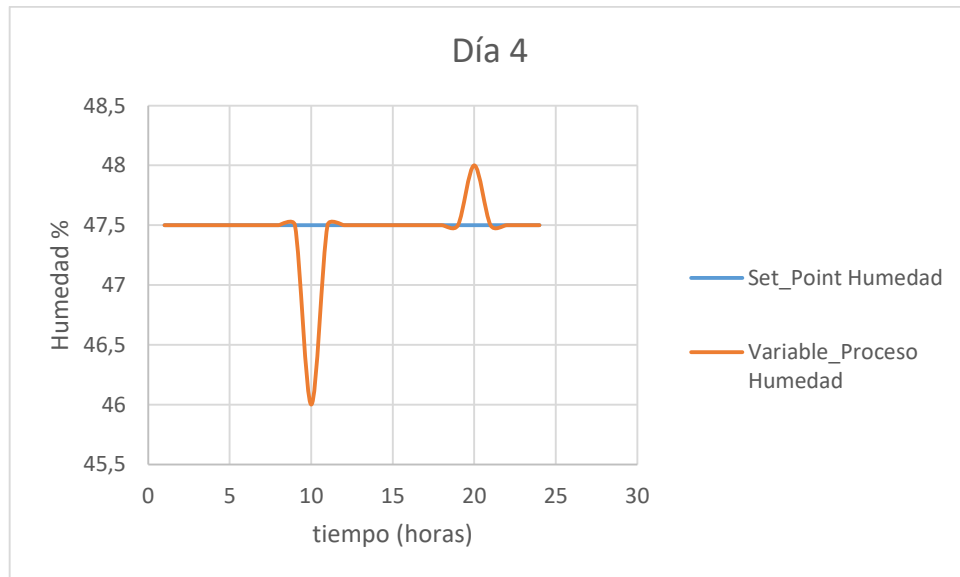


Figura 73. Resultado día 4, SP Vs. PV Humedad

Elaborado por: el investigador

- **Día 5**

Variable de set point de Temperatura = 24 °C

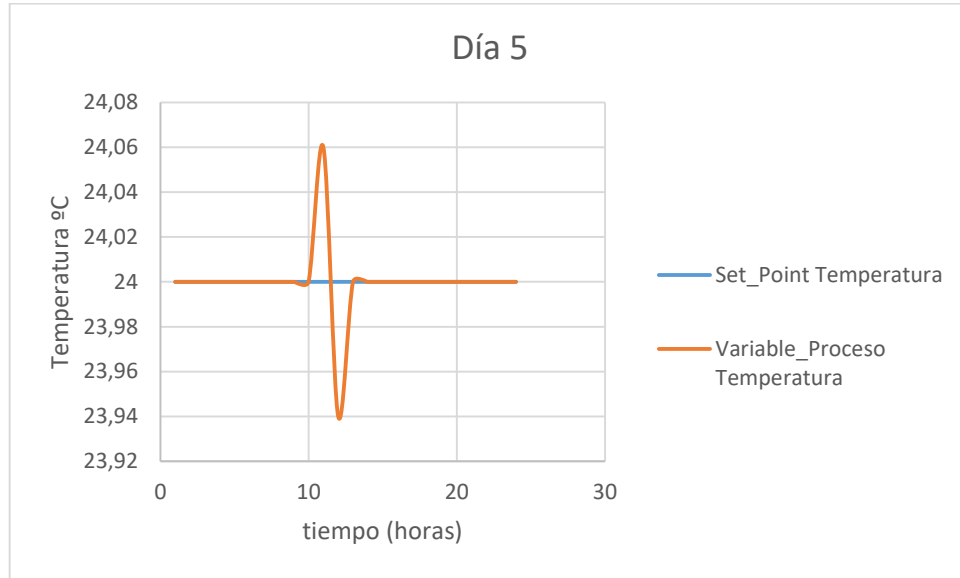


Figura 74. Resultado día 5, SP Vs. PV Temperatura

Elaborado por: el investigador

Variable de set point de Humedad = 52.5%



Figura 75. Resultado día 5, SP Vs. PV Humedad

Elaborado por: el investigador

- **Día 6**

Variable de set point de Temperatura = 22 °C

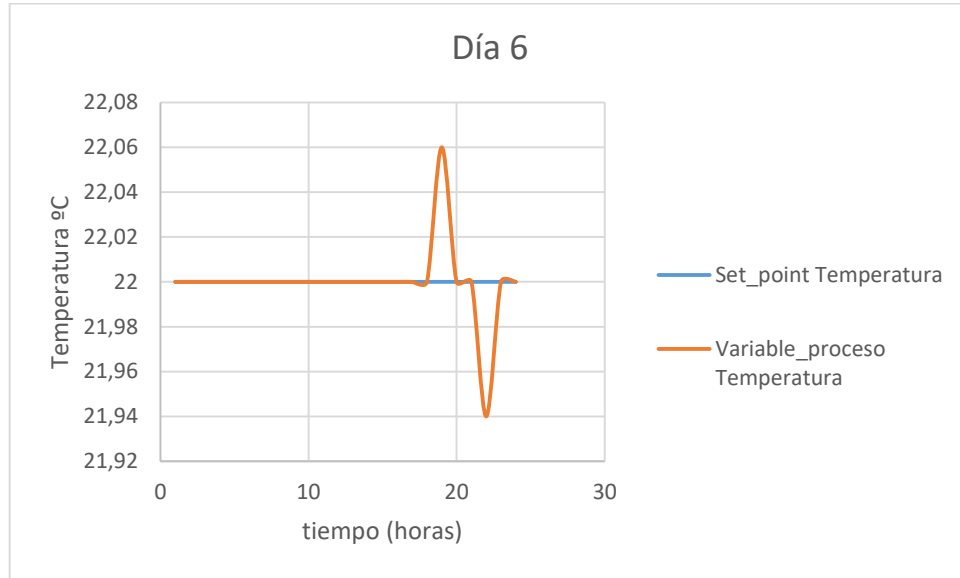


Figura 76. Resultado día 6, SP Vs. PV Temperatura

Elaborado por: el investigador

Variable de set point de Humedad = 57.5%

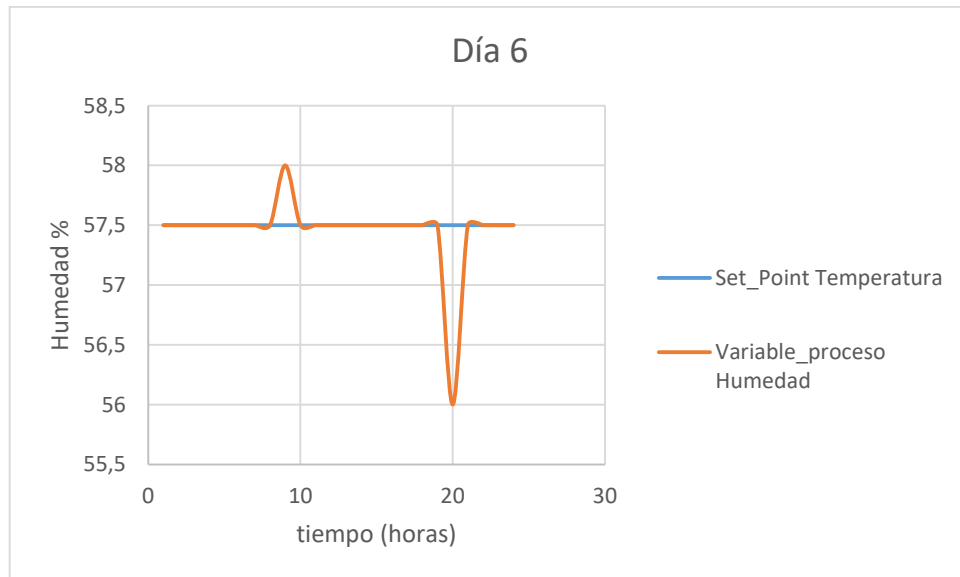


Figura 77. Resultado día 6, SP Vs. PV Humedad

Elaborado por: el investigador

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- Los métodos de inferencia referidos al control difuso, son eficientes, versátiles y sencillos, resultando así mediante el uso del método de la centroide la obtención de un solo valor obtenido de un conjunto de reglas difusas encapsuladas en áreas solapadas entre sí, llegando así a determinar que el método de Mamdani es el que más se asemejó al comportamiento de un humano dentro de la toma de decisiones para llevar a cabo un proceso de control. El principal inconveniente que presenta este método fue que aumenta el número de reglas exponencialmente al aumentar el número de variables a controlar, por lo tanto, mientras más reglas hay que editar, más complejo se vuelve saber si son o no adecuadas.
- Utilizando hardware y software libre para el proceso de monitoreo, se realizó una interfaz gráfica, la cual permite monitorear en tiempo real los procesos que se están realizando en la planta, mediante la utilización de servidores LAMP (Linux, Apache, MySQL, PHP) y la inmersión de los datos en la red, desde cualquier lugar dentro de la red de área local fue posible navegar dentro de los datos de proceso, siempre y cuando el usuario esté debidamente registrado en la base de datos que contiene a los mismos, haciendo posible utilizar dispositivos inteligentes como smartphones, tablets, computadores, etc. La principal ventaja es que además de ser segura la navegación por los servidores, su mantenimiento fue reducido, además de ello al ser una distribución gratuita, no fue necesario la adquisición de licencias para operar, evitando así costos extras.
- La implementación de un sistema de control difuso y monitoreo para granjas avícolas utilizando suelo radiante con energía solar utilizando el método de inferencia de Mamdani, obtuvo buenos resultados, ya que mediante la aplicación de este método fue posible controlar dos variables de entrada y dos variables de salida, además de ello, gracias a las ventajas que presenta este tipo de controlador,

no fue necesario la obtención de un modelo matemático de la planta, pero si fuere necesario obtener información de cómo opera, es decir, se necesitó de funciones de membresías y reglas de control difusas. Como contraparte, al utilizar energías renovables se evita en gran medida la emisión de gases contaminantes al ambiente, mejorando así la calidad de vida de los habitantes y de las aves que se crían dentro de este proceso.

4.2. Recomendaciones

- Al trabajar con sistemas de colectores solares, se recomienda mensualmente realizar un mantenimiento preventivo a todos los elementos que conforman el sistema de calentamiento solar, ya que el polvo y demás partículas podrían bajar el rendimiento de estos. Al momento de hacerlo, se debe tener cuidado ya que estos equipos generalmente son de vidrio y resultan ser demasiado frágiles.
- Para evitar sobrecalentamientos en los dispositivos electrónicos y que operen de manera adecuada y óptima, se recomienda ubicarlos en un área ventilada y libre de polvo, o en su defecto colocar carcasas protectoras a los equipos, para evitar así que el polvo pueda generar inconvenientes en un futuro y generar algún tipo de daño a los equipos.
- Al trabajar con agua caliente, es posible sufrir quemaduras, estas no fueron de consideración pero es preferible utilizar algún tipo de protección y estar debidamente concentrados mientras se opera al momento de realizar las conexiones para el piso radiante. Además de ello una vez instalado todos los equipos y materiales en la planta, es necesario purgar las tuberías y recipientes, debido a que el aire dentro de estos puede ocasionar graves daños al sistema de circulación del agua caliente.

C. MATERIALES DE REFERENCIA

Bibliografía

- [1] S. A. Ili Flores, H. J. Konno, A. Massafra y L. Schiaffino, «Simultaneous Humidity and Temperature Fuzzy Logic Control in Neonatal Incubators,» 17 12 2018. [En línea]. Available: <https://ieeexplore.ieee.org/document/8577290/>. [Último acceso: 20 12 2019].
- [2] A. R. Cabrera Castro, «DISEÑO Y SIMULACIÓN DE UN SISTEMA DE CONTROL DE TEMPERATURA E ILUMINACIÓN PARA LA CRIANZA DE POLLOS EN HACIENDAS USANDO UNA RED DE SENSORES,» 2017. [En línea]. Available: <http://repositorio.ug.edu.ec/bitstream/redug/23721/1/B-CINT-PTG-N.202.Cabrera%20Castro%20Andr%C3%A9s%20Ram%C3%B3n.pdf>. [Último acceso: 20 12 2019].
- [3] H. F. Villacís López, «DISEÑO DE LOS SISTEMAS DE AUTOMATIZACION PARA LA AMPLIACION DE UNA GRANJA AVICOLA,» 12 2017. [En línea]. Available: <https://bibdigital.epn.edu.ec/bitstream/15000/19018/1/CD-8415.pdf>. [Último acceso: 20 12 2019].
- [4] R. Ben Ali, E. Aridhi, M. Abbes y A. Mami, «Fuzzy logic controller of temperature and humidity inside an agricultural greenhouse,» 26 05 2016. [En línea]. Available: <https://ieeexplore.ieee.org/document/7478929>. [Último acceso: 20 12 2019].
- [5] J. C. Mugisha, B. Munyazikwiye y H. R. Karími, «Design of temperature control system using conventional PID and Intelligent Fuzzy Logic controller,» 28 01 2016. [En línea]. Available: <https://ieeexplore.ieee.org/document/7391893/>. [Último acceso: 20 12 2019].
- [6] P. Barros, «EVALUACIÓN DE UN SUBPRODUCTO DE DESTILERÍA DE ALCOHOL (VINAZA) COMO ADITIVO EN LA ALIMENTACIÓN DE POLLOS DE ENGORDE,» 2019. [En línea]. Available:

- <http://dspace.esPOCH.edu.ec/bitstream/123456789/63/1/17T0921.pdf>. [Último acceso: 16 09 2019].
- [7] V. Duplaá, «Plan de negocios para la creación de una granja avícola de pollos Broiler en la ciudad de Milagro,» 2016. [En línea]. Available: http://dspace.utpl.edu.ec/bitstream/123456789/15767/1/Duplaa_Yepez_Veronica_Marjorie.pdf. [Último acceso: 16 09 2019].
- [8] S. Rosales Tapia, «Estudio de Mercado Avícola enfocado a la Comercialización del Pollo en Pie,» [En línea]. Available: <https://www.scpm.gob.ec/sitio/wp-content/uploads/2019/03/ESTUDIO-AVCOLA-VERSION-PUBLICA.pdf>. [Último acceso: 08 01 2020].
- [9] Universidad de Piura, «LÓGICA DIFUSA Y SISTEMAS DE CONTROL,» [En línea]. Available: http://www.biblioteca.udep.edu.pe/bibvirudep/tesis/pdf/1_185_184_133_1746.pdf. [Último acceso: 11 09 2019].
- [10] J. Falco, K. Stouffer, A. Wavering y F. Proctor, «IT Security for Industrial Control Systems,» [En línea]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=821684. [Último acceso: 07 01 2020].
- [11] A. Rodríguez Penin, Sistemas SCADA 2a, Marcombo, 2007.
- [12] J. I. Escudero Fombuena y R. J. Luque, Telecontrol de Redes Eléctricas, Sevilla, 1994.
- [13] A. Ruiz Canales y J. M. Molina Martínez, Automatización y telecontrol de sistemas de riego, Barcelona: Marcombo, 2010.
- [14] C. Silva, «Unidades de Transmisión Remota: Control y supervisión remota en tiempo real,» [En línea]. Available: <http://www.emb.cl/electroindustria/articulo.mvc?xid=1303&edi=68&xit=unid>. [Último acceso: 10 01 2020].

- [15] M. Colobran Huguet, J. M. Arqués Soldevila y E. M. Galindo, Administración de sistemas operativos en red, Editorial UOC, 2008.
- [16] E. Mifsuf Talón, «Apache,» [En línea]. Available: <http://descargas.pntic.mec.es/mentor/visitas/Apache.pdf>. [Último acceso: 10 01 2020].
- [17] J. Niño, Servidores de aplicaciones web (Aplicaciones web), Editex, 2011.
- [18] D. Rodríguez, «Servidores Web: Apache e IIS,» [En línea]. Available: <http://www.cc.uah.es/drg/docencia/Servidores/ServidoresWeb4x1.pdf>. [Último acceso: 10 01 2020].
- [19] A. B. García Mariscal, Modelo de programación web y bases de datos, Editorial Elearning, 2015.
- [20] P. Rufes Martínez, Energía solar térmica: técnicas para su aprovechamiento, Marcombo, 2010.
- [21] A. Madrid Vicente, Guía completa de las energías renovables y fósiles, AMV Ediciones, 2012.
- [22] L. Carribero Pérez, «Diseño de una instalación solar térmica para la producción de agua caliente sanitaria y calefacción por suelo radiante en una vivienda,» 06 2010. [En línea]. Available: <https://rodin.uca.es/xmlui/bitstream/handle/10498/9763/34755597.pdf?sequence=1>. [Último acceso: 10 01 2020].
- [23] J. De Andrés y Rodríguez-Pomatta, Calefacción y agua caliente sanitaria: (serie de instalaciones térmicas en la edificación), A. Madrid Vicente, 1991.
- [24] Á. Miranda Barreras y L. Jutglar, Técnicas de Calefacción, Marcombo, 2010, 2010.
- [25] A. Olivera Oliva, «Diseño energético de un suelo radiante para una sala de 10 m² ubicada a 4000 msnm en Langui-Cuzco,» 03 2011. [En línea]. Available: <http://tesis.pucp.edu.pe/repositorio/bitstream/handle/20.500.12404/805/OLIV>

ERA_OLIVA_DAVY_DISEÑO_ENERGETICO_CUZCO.pdf?sequence=1&isAllowed=y. [Último acceso: 10 01 2020].

- [26] A. Aranda y I. Zabalza, *Energía solar térmica (Serie Energías renovables)*, Prensas de la Universidad de Zaragoza, 2010.
- [27] «Sensores de Temperatura,» [En línea]. Available: <https://proyectoarduino.com/sensor-de-temperatura/>. [Último acceso: 08 01 2020].
- [28] «Sensor de humedad,» [En línea]. Available: <https://www.mecatronicalatam.com/tutorial/es/sensores/sensor-de-humedad>. [Último acceso: 09 01 2020].
- [29] V. D. I. I. d. I. Cosas, «Internet de las Cosas en Monitoreo Industrial,» [En línea]. Available: <https://www.vesat.cl/servicios/monitoreo-industrial/>. [Último acceso: 10 01 2020].
- [30] W. L. T. College, «Familia IEEE 802.11,» [En línea]. Available: <http://bibing.us.es/proyectos/abreproy/11579/fichero/f.+Cap%C3%ADtulo+2+-+Familia+IEEE+802.11.pdf>. [Último acceso: 17 12 2018].
- [31] C. X. Ferigra Orellana y C. D. Ojeda Flores, «Diseño de una red wlan para cobertura total en el campus de la Universidad Católica de Santiago de Guayaquil empleando tecnología Cisco,» Enero 2013. [En línea]. Available: <http://repositorio.ucsg.edu.ec/bitstream/3317/8576/1/T-UCSG-PRE-TEC-ITEL-239.pdf>. [Último acceso: 10 Enero 2019].
- [32] E. d. redacción, «Diferencias entre suelo radiante eléctrico y suelo radiante por agua,» [En línea]. Available: <https://www.caloryfrio.com/aire-acondicionado/aire-instalaciones-componentes/suelo-radiante-electrico-suelo-radiante-agua.html>. [Último acceso: 13 01 2020].
- [33] L. Miller, «Wemos D1 Mini NodeMCU,» 05 08 2018. [En línea]. Available: <https://www.learnrobotics.org/blog/which-controller-should-i-use-wemos-d1-mini-or-nodemcu/>. [Último acceso: 15 01 2020].

- [34] Raspberry Pi Foundation, «Raspberrypi.org,» [En línea]. Available: <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf>. [Último acceso: 18 03 2019].
- [35] MCI electronics, «Arduino,» [En línea]. Available: <http://arduino.cl/que-es-arduino/>. [Último acceso: 19 03 2019].
- [36] Wemos, «Wemos Electronics,» 28 12 2018. [En línea]. Available: https://wiki.wemos.cc/products:d1:d1_mini. [Último acceso: 19 03 2019].
- [37] STRATASYS solution, «© 2019 GrabCAD,» 28 10 2018. [En línea]. Available: <https://grabcad.com/library/water-pump-800l-h-dc5-12v-g1-2-1>. [Último acceso: 19 03 2019].
- [38] D. SEMICONDUCTOR, «DS18B20 Programmable Resolution 1-Wire® Digital Thermometer,» [En línea]. Available: <https://cdn.sparkfun.com/datasheets/Sensors/Temp/DS18B20.pdf>. [Último acceso: 01 07 2020].
- [39] Omniblug, «Sensor de temperatura y humedad DHT11,» [En línea]. Available: <http://www.omniblug.com/sensor-temperatura-humedad-DHT11-DHT22.html>. [Último acceso: 2020 01 07].
- [40] Pololu, «Robotics & Electronics,» [En línea]. Available: <https://www.pololu.com/product/713>. [Último acceso: 19 03 2019].
- [41] openSUSE Contributors & others, «© 2001–2019 SUSE LLC,» 10 11 2018. [En línea]. Available: https://es.opensuse.org/SDB:Instalar_un_servidor_LAMP. [Último acceso: 19 03 2019].
- [42] TP-Link Technologies Co., Ltd. [En línea]. Available: https://www.tp-link.com/ar/products/details/cat-12_TL-WA901ND.html. [Último acceso: 19 03 2019].

- [43] «DC FAN General specifications,» [En línea]. Available: <https://www.farnell.com/datasheets/1878245.pdf>. [Último acceso: 09 01 2020].
- [44] B. P. Carvajal Olmedo, «PLAN DE COMERCIO EXTERIOR Y NEGOCIOS INTERNACIONALES DE LA IMPORTACION DE CALENTADORES DE AGUA SOLARES,» 2011. [En línea]. Available: <https://repositorio.espe.edu.ec/bitstream/21000/4124/1/T-ESPE-057059.pdf>. [Último acceso: 10 01 2020].
- [45] I. Maxim Integrated Products, «Extremely Accurate I2C-Integrated RTC/TCXO/Crystal,» 03 2015. [En línea]. Available: <https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>. [Último acceso: 11 01 2020].
- [46] G. Arellano Peche, «EVOLUCIÓN Y SITUACIÓN ACTUAL DE LOS COSTES DE PRODUCCIÓN EN LAS GRANJAS DE BROILERS,» [En línea]. Available: <https://avicultura.info/wp-content/uploads/2018/10/1018-ESPECIAL-NAVES-1.pdf>. [Último acceso: 08 01 2020].
- [47] D. Erazo y V. Salgado, «DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE TEMPERATURA Y HUMEDAD PARA UN GALPÓN DE POLLOS DE LA AVÍCOLA “LA ESPERANZA”,» 06 2014. [En línea]. Available: <http://repositorio.utn.edu.ec/bitstream/123456789/3766/5/04%20MEC%20059%20Tesis.pdf>. [Último acceso: 06 09 2019].
- [48] I. Viladomiu, «Ventilación por depresión en naves avícolas,» 27 02 2016. [En línea]. Available: <https://avicultura.info/ventilacion-por-depresion-en-naves-avicolas/>. [Último acceso: 16 09 2019].
- [49] A. Kaur y A. Kaur, «Comparison of Mamdani-Type and Sugeno-Type Fuzzy Inference Systems for Air Conditioning System,» 05 2012. [En línea]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.486.1238&rep=rep1&type=pdf>. [Último acceso: 09 01 2020].

- [50] A. H. a. N. D. Georganas, «A comparison of Mamdani and Sugeno fuzzy inference systems for evaluating the quality of experience of Hapto-Audio-Visual applications,» 2008. [En línea]. Available: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4685304&isnumber=4685283>. [Último acceso: 09 01 2020].
- [51] I. The MathWorks, «Fuzzy Logic Toolbox, Design and simulate fuzzy logic systems,» [En línea]. Available: <https://www.mathworks.com/products/fuzzy-logic.html>. [Último acceso: 09 01 2020].
- [52] R. P. FOUNDATION, «NOOBS DOWNLOAD,» [En línea]. Available: <https://www.raspberrypi.org/downloads/noobs/>. [Último acceso: 11 01 2020].

Anexos

Anexo A: Códigos de programación en Raspberry Pi y Arduino.

A. Códigos de programación en Raspberry.

Anexo 1: localhost/login_w3.php

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
h2{text-align:center;
}
body {font-family: Arial, Helvetica, sans-serif;}
/* Full-width input fields */
input[type=text], input[type=password] {
width: 100%;
padding: 12px 20px;
margin: 8px 0;
display: inline-block;
border: 1px solid #ccc;
box-sizing: border-box;
}
/* Set a style for all buttons */
button {
background-color: #4CAF50;
color: white;
padding: 14px 20px;
margin: 8px 0;
border: none;
cursor: pointer;
width: 100%;
}
button:hover {
opacity: 0.8;
```

```

}
/* Extra styles for the cancel button */
.cancelbtn {
width: auto;
padding: 10px 18px;
background-color: #f44336;
}
/* Center the image and position the close button */
.imgcontainer {
text-align: center;
margin: 24px 0 12px 0;
position: relative;
}
img.avatar {
width: 10%;
border-radius: 20%;
}
.container {
padding: 16px;
}
span.psw {
float: right;
padding-top: 16px;
}
/* The Modal (background) */
.modal {
display: none; /* Hidden by default */
position: fixed; /* Stay in place */
z-index: 1; /* Sit on top */
left: 0;
top: 0;
width: 100%; /* Full width */
height: 100%; /* Full height */
overflow: auto; /* Enable scroll if needed */
background-color: rgb(0,0,0); /* Fallback color */
background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
padding-top: 60px;
```

```

}
/* Modal Content/Box */
.modal-content {
    background-color: #fefefe;

    margin: 5% auto 15% auto; /* 5% from the top,
15% from the bottom and centered */

    border: 1px solid #888;

    width: 80%; /* Could be more or less, depending
on screen size */
}
/* The Close Button (x) */
.close {
    position: absolute;

    right: 25px;

    top: 0;

    color: #000;

    font-size: 35px;

    font-weight: bold;
}
.close:hover,
.close:focus {
    color: red;

    cursor: pointer;
}
/* Add Zoom Animation */
.animate {
    -webkit-animation: animatezoom 0.6s;

    animation: animatezoom 0.6s
}
@-webkit-keyframes animatezoom {
    from {-webkit-transform: scale(0)}
    to {-webkit-transform: scale(1)}
}
@keyframes animatezoom {
    from {transform: scale(0)}
    to {transform: scale(1)}
}
/* Change styles for span and cancel button on extra
small screens */

```

```

@media screen and (max-width: 300px) {
    span.psw {
        display: block;

        float: none;
    }

    .cancelbtn {
        width: 100%;
    }
}
</style>
</head>
<body>
<h2>Bienvenidos</h2>
<h2>Para poder ingresar por favor da clic en el botón
de abajo e ingresa tu usuario y contraseña.</h2>
<button
onclick="document.getElementById('id01').style.di
splay='block'"
style="width:100%;">Ingresar</button>
<div id="id01" class="modal">
    <form class="modal-content animate"
method="POST" action="validar_w3.php">
        <div class="imgcontainer">
            <span
onclick="document.getElementById('id01').style.di
splay='none'" class="close" title="Close
Modal">&times;</span>
            
        </div>
        <div class="container">
            <label for="uname"><b>Usuario</b></label>
            <input type="text" placeholder="Ingresar
Usuario" name="nombre" required>
            <label for="psw"><b>Contraseña</b></label>
            <input type="password" placeholder="Ingresar
Contraseña" name="password" required>
            <button type="submit">Login</button>
            <label>
                <input type="checkbox" checked="checked"
name="remember"> Recordarme
            </label>
        </div>
    </form>
</div>

```

```

<div class="container" style="background-color:#f1f1f1">

    <button type="button"
onclick="document.getElementById('id01').style.display='none'" class="cancelbtn">Cancelar</button>

    <span class="psw">¿Olvidaste la <a href="informacion_contrasena.php">contraseña?</a></span>

</div>

</form>s

</div>

<script>
// Get the modal
var modal = document.getElementById('id01');

// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
    if (event.target == modal) {
        modal.style.display = "none";
    }
}
</script>
</body>
</html>

```

Anexo 2: localhost/validar_w3.php

```

<!doctype html>

<html>

<head>

<meta charset="utf-8">

<title>Documento sin título</title>

</head>

<body>

<?php

try{

    $base=new PDO("mysql:host=localhost;dbname=tesis","root","1234");

    $base->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $sql="SELECT * FROM USUARIOS_PASS WHERE USUARIOS= :login AND PASSWORD= :password";

    $resultado=$base->prepare($sql);

```

```

    $login=htmlentities(addslashes($_POST["nombre"]));

    $password=htmlentities(addslashes($_POST["password"]));

    $resultado->bindValue(":login", $login);

    $resultado->bindValue(":password", $password);

    $resultado->execute();

    $numero_registro=$resultado->rowCount();

    if($numero_registro!=0){

        //echo"<h2>adelante</h2>";

        session_start();

        $_SESSION["usuario"]=$_POST["nombre"];

        header("location:datos_folder/todo_datos.php");

    }else{

        header("location:login_w3.php");

    }

} catch(Exception $e){

    die("Error: " . $e->getMessage());

}

?>

</body>

</html>

Anexo 3: localhost/informacion_contrasena.php

<!DOCTYPE html>

<html>

<head>

<style>

.button {

    background-color: #4CAF50;

    border: none;

    color: white;

    padding: 15px 32px;

```

```

text-align: center;

text-decoration: none;

display: inline-block;

font-size: 16px;

margin: 4px 2px;

cursor: pointer;
}

</style>

</head>

<body>

<h2>Para poder recuperar la contraseña, por favor
comúnicate con el administrador de la
página.<br>Gracias</h2>

<h2>Fernando Monta</h2>

<h2>Tel: 0969317448</h2>

<a href="login_w3.php"
class="button">Regresar</a>

</body>

</html>

```

Anexo 4: localhost/cierre.php

```

<!doctype html>

<html>

<head>

<meta charset="utf-8">

<title>Documento sin título</title>

</head>

<body>

<?php
session_start();

session_destroy();

header("location:login_w3.php");

?>

</body>

```

```
</html>
```

Anexo 5: localhost/uploads/abrir_conexion.php

```

<?php
$host = "localhost";
$user = "root";
$password = "1234";
$db = "tesis";

$conexion =
mysqli_connect($host,$user,$password,$db);

$tabla_db1 = "datos"; // tabla de usuarios

?>

```

Anexo 6: localhost/uploads/cerrar_conexion.php

```

<?php
mysqli_close($conexion);

?>

```

Anexo 7: localhost/uploads/fuzzy_upload.php

```

<?php
require("abrir_conexion.php");

$temp_mysql =
mysqli_real_escape_string($conexion,
$_GET['set_point']);

$temp_mysql2 =
mysqli_real_escape_string($conexion,
$_GET['process_value']);

$temp_in = "INSERT INTO datos_fuzzy (set_point,
process_value)
VALUES('".$temp_mysql."','".$temp_mysql2."')";

mysqli_query($conexion, $temp_in);

require("cerrar_conexion.php");

?>

```

Anexo 8: localhost/uploads/temperatura.php

```

<?php
require("abrir_conexion.php");

$temp_mysql =
mysqli_real_escape_string($conexion,
$_GET['Temperatura_tanque_colectores']);

$temp_mysql2 =
mysqli_real_escape_string($conexion,
$_GET['Temperatura_tanque_reserva']);

$temp_mysql3 =
mysqli_real_escape_string($conexion,
$_GET['Temperatura_alfombra']);

```

```

$temp_mysql4 =
mysqli_real_escape_string($conexion,
$_GET['Temperatura_piso']);

$temp_mysql5 =
mysqli_real_escape_string($conexion,
$_GET['Temperatura_ambiente_in']);

$temp_mysql6 =
mysqli_real_escape_string($conexion,
$_GET['Temperatura_ambiente_out']);

$temp_mysql7 =
mysqli_real_escape_string($conexion,
$_GET['Humedad_in']);

$temp_mysql8 =
mysqli_real_escape_string($conexion,
$_GET['Humedad_out']);

$temp_in = "INSERT INTO datos
(Temperatura_tanque_colectores,
Temperatura_tanque_reserva,
Temperatura_alfombra, Temperatura_piso,
Temperatura_ambiente_in,
Temperatura_ambiente_out, Humedad_in,
Humedad_out)
VALUES('".$temp_mysql1."','".$temp_mysql2."','".$temp_mysql3."','".$temp_mysql4."','".$temp_mysql5."','".$temp_mysql6."','".$temp_mysql7."','".$temp_mysql8."')";

mysqli_query($conexion, $temp_in);
require("cerrar_conexion.php");

?>

Anexo 9: localhost/datos_folder/DatoSensores.php

<?php
header('Content-Type: application/json');

$pdo=new
PDO("mysql:dbname=tesis;host=localhost","root","1234");

switch($_GET['q']){
    // Buscar Último Dato
    case 1:

        $statement=$pdo->prepare("SELECT id,
        Temperatura_tanque_colectores,
        Temperatura_tanque_reserva,
        Temperatura_alfombra, Temperatura_piso,
        Temperatura_ambiente_in,
        Temperatura_ambiente_out, Humedad_in,
        Humedad_out, fecha FROM datos ORDER BY id
        DESC LIMIT 0,10");

        $statement->execute();

        $results=$statement->fetchAll(PDO::FET
        CH_ASSOC);

        $json=json_encode($results);

        echo $json;

        break;

    case 2:

        $statement=$pdo->prepare("SELECT * FROM
        datos where id= (SELECT max(id) FROM datos);

        $statement->execute();

        $results=$statement->fetchAll(PDO::FET
        CH_ASSOC);

        $json=json_encode($results);

        echo $json;

        break;

        // Buscar Todos los datos
        default:

        $statement=$pdo->prepare("SELECT
        Temperatura,Humedad FROM datos ORDER BY
        id ASC");

        $statement->execute();

        $results=$statement->fetchAll(PDO::FET
        CH_ASSOC);

        $json=json_encode($results);

        echo $json;

        break;

    }

?>

Anexo 10:
localhost/datos_folder/DatosFuzzy.php

<?php
header('Content-Type: application/json');

$pdo=new
PDO("mysql:dbname=tesis;host=localhost","root","1234");

```

```

switch($_GET['q']){
    // Buscar Último Dato
    case 1:

        $statement=$pdo->prepare("SELECT id, set_point,
        process_value FROM datos_fuzzy ORDER BY id
        DESC LIMIT 0,10");

        $statement->execute();

        $results=$statement->fetchAll(PDO::FET
        CH_ASSOC);

        $json=json_encode($results);

        echo $json;

        break;
    }
?>

```

Anexo 11: localhost/datos_folder/todo_datos.php

```

<!DOCTYPE html>
<?php
session_start();
if(!isset($_SESSION["usuario"])){
    header("location:/tesis/login_w3.php");
}
?>
<html>
<style type="text/css">
.tg {border-collapse:collapse;border-
spacing:0;border-color:#999;}
.tg td{font-family:Arial, sans-serif;font-
size:14px;padding:10px 5px;border-
style:solid;border-
width:1px;overflow:hidden;word-
break:normal;border-
color:#999;color:#444;background-
color:#F7FDFA;}
.tg th{font-family:Arial, sans-serif;font-
size:14px;font-weight:normal;padding:10px
5px;border-style:solid;border-
width:1px;overflow:hidden;word-
break:normal;border-
color:#999;color:#fff;background-
color:#26ADE4;}

```

```

.tg .tg-l182{font-weight:bold;font-style:italic;text-
decoration:underline;font-size:18px;font-
family:Arial, Helvetica, sans-
serif !important;;border-color:#000000;text-
align:center}

```

```

.tg .tg-xygg{font-weight:bold;font-size:18px;font-
family:Arial, Helvetica, sans-
serif !important;;border-color:#000000;text-
align:right}

```

```

.tg .tg-dtjc{background-color:#D2E4FC;font-
size:18px;font-family:Arial, Helvetica, sans-
serif !important;;border-color:#000000;text-
align:left}

```

```

#tablita{
    background-color: #fefefe;
    margin: 5% auto 15% auto; /* 5% from the top,
15% from the bottom and centered */
    border: 1px solid #888;
    width: 80%; /* Could be more or less, depending
on screen size */
}

```

```

.button {
    background-color: #4CAF50;
    border: none;
    color: white;
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    margin: 4px 2px;
    cursor: pointer;
}

```

```
</style>
```

```
<head>
```

```
<title>Tabla Últimos Datos</title>
```

```
</head>
```

```
<h1 style="color:blue; text-align: center;"><?php
echo "Bienvenido
".$_SESSION["usuario"];?></h1>
```

```
<script type="text/javascript" src="jquery/jquery-
1.9.0.min.js"></script>
```

```
<script type="text/javascript">
```

```
$(document).ready( function(){
```

```

refresh();
});
function refresh()
{
    setInterval(function() {
var JSON=$.ajax({
    url:"DatoSensores.php?q=2",
    dataType: 'json',
    async: false}).responseText;
    var Respuesta=jQuery.parseJSON(JSON);

document.getElementById("Temperatura_tanque_coletores").innerHTML =
Respuesta[0].Temperatura_tanque_coletores;

document.getElementById("Temperatura_tanque_reservA").innerHTML =
Respuesta[0].Temperatura_tanque_reserva;

document.getElementById("Temperatura_alfombraA").innerHTML =
Respuesta[0].Temperatura_alfombra;

document.getElementById("Temperatura_pisO").innerHTML = Respuesta[0].Temperatura_piso;

document.getElementById("Temperatura_ambiente_iN").innerHTML =
Respuesta[0].Temperatura_ambiente_in;

document.getElementById("Temperatura_ambiente_ouT").innerHTML =
Respuesta[0].Temperatura_ambiente_out;

document.getElementById("Humedad_iN").innerHTML = Respuesta[0].Humedad_in;

document.getElementById("Humedad_ouT").innerHTML = Respuesta[0].Humedad_out;

document.getElementById("fecha").innerHTML = Respuesta[0].fecha;

document.getElementById("contador").innerHTML = Respuesta[0].id;
    }, 900);
    }
</script>

```

```

<table class="tg" id="tablita">
<tr>
<th class="tg-1182" colspan="2">Últimos datos
sensados</th>
</tr>
<tr>
<td class="tg-xygg">Temperatura_tanque_coletores =</td>
<td class="tg-dtjc"
id="Temperatura_tanque_coletoresS"></td>
</tr>
<tr>
<td class="tg-xygg">Temperatura_tanque_reserva =</td>
<td class="tg-dtjc"
id="Temperatura_tanque_reservA"></td>
</tr>
<tr>
<td class="tg-xygg">Temperatura_alfombra
=</td>
<td class="tg-dtjc"
id="Temperatura_alfombraA"></td>
</tr>
<tr>
<td class="tg-xygg">Temperatura_piso =</td>
<td class="tg-dtjc"
id="Temperatura_pisO"></td>
</tr>
<tr>
<td class="tg-xygg">Temperatura_ambiente_in
=</td>
<td class="tg-dtjc"
id="Temperatura_ambiente_iN"></td>
</tr>
<tr>
<td class="tg-xygg">Temperatura_ambiente_out
=</td>
<td class="tg-dtjc"
id="Temperatura_ambiente_ouT"></td>
</tr>
<tr>
<td class="tg-xygg">Humedad_in =</td>
<td class="tg-dtjc" id="Humedad_iN"></td>

```

```

</tr>
<tr>
  <td class="tg-xygg">Humedad_out =</td>
  <td class="tg-dtjc" id="Humedad_ouT"></td>
</tr>
<tr>
  <td class="tg-xygg">Fecha =</td>
  <td class="tg-dtjc" id="fechA"></td>
</tr>
<tr>
  <td class="tg-xygg">Contador =</td>
  <td class="tg-dtjc" id="contadoR"></td>
</tr>
</table>
<style>
#columnchart_material{
  background-color: #fefefe;

  margin: 5% auto 15% auto; /* 5% from the top,
15% from the bottom and centered */

  border: 1px solid #888;

  width: 80%; /* Could be more or less, depending
on screen size */

  height: 500px;
}
#curve_chart{
  background-color: #fefefe;

  margin: 5% auto 15% auto; /* 5% from the top,
15% from the bottom and centered */

  border: 1px solid #888;

  width: 80%; /* Could be more or less, depending
on screen size */

  height: 500px;
}
</style>
<script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></s
cript>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.
2.0/jquery.min.js"></script>
<script type="text/javascript">
    google.charts.load('current',
    {'packages':['bar']});

    google.charts.setOnLoadCallback(drawChart);

    function drawChart() {

      var data =
      google.visualization.arrayToDataTable([

        ['Id', 'Temperatura_tanque_colectores',
        'Temperatura_tanque_reserva',
        'Temperatura_alfombra', 'Temperatura_piso',
        'Temperatura_ambiente_in',
        'Temperatura_ambiente_out', 'Humedad_in',
        'Humedad_out'],

        [2014, 2015, 1000, 400, 1000, 400, 400,
        1000, 400],

        [2014, 2015, 1000, 400, 1000, 400, 400,
        1000, 400],

        [2014, 2015, 1000, 400, 1000, 400, 400,
        1000, 400],

        [2014, 2015, 1000, 400, 1000, 400, 400,
        1000, 400],

        [2014, 2015, 1000, 400, 1000, 400, 400,
        1000, 400],

        [2014, 2015, 1000, 400, 1000, 400, 400,
        1000, 400],

        [2014, 2015, 1000, 400, 1000, 400, 400,
        1000, 400]

      ]);

      var options = {

        chart: {

          title: 'Sensorización',

          subtitle: 'Temperatura interna Prototipo',

        }

      };

      var chart = new
      google.charts.Bar(document.getElementById('colu
      mnchart_material'));

      chart.draw(data,
      google.charts.Bar.convertOptions(options));

      setInterval(function() {

        var JSON=$.ajax({

```



```

url:"DatoSensores.php?q=1",
dataType: 'json',
async: false}).responseText;

var Respuesta=jQuery.parseJSON(JSON);
data.setValue(0, 0,Respuesta[0].id);

data.setValue(0,
1,Respuesta[0].Temperatura_tanque_colectores);

data.setValue(0,
2,Respuesta[0].Temperatura_tanque_reserva);

data.setValue(0,
3,Respuesta[0].Temperatura_alfombra);

data.setValue(0,
4,Respuesta[0].Temperatura_piso);

data.setValue(0,
5,Respuesta[0].Temperatura_ambiente_in);

data.setValue(0,
6,Respuesta[0].Temperatura_ambiente_out);

data.setValue(0,
7,Respuesta[0].Humedad_in);

data.setValue(0,
8,Respuesta[0].Humedad_out);

data.setValue(1,
0,Respuesta[1].id);

data.setValue(1,
1,Respuesta[1].Temperatura_tanque_colectores);

data.setValue(1,
2,Respuesta[1].Temperatura_tanque_reserva);

data.setValue(1,
3,Respuesta[1].Temperatura_alfombra);

data.setValue(1,
4,Respuesta[1].Temperatura_piso);

data.setValue(1,
5,Respuesta[1].Temperatura_ambiente_in);

data.setValue(1,
6,Respuesta[1].Temperatura_ambiente_out);

data.setValue(1,
7,Respuesta[1].Humedad_in);

data.setValue(1,
8,Respuesta[1].Humedad_out);

data.setValue(2,
0,Respuesta[2].id);

data.setValue(2,
1,Respuesta[2].Temperatura_tanque_colectores);

data.setValue(2,
2,Respuesta[2].Temperatura_tanque_reserva);

```

```

data.setValue(2,
3,Respuesta[2].Temperatura_alfombra);

data.setValue(2,
4,Respuesta[2].Temperatura_piso);

data.setValue(2,
5,Respuesta[2].Temperatura_ambiente_in);

data.setValue(2,
6,Respuesta[2].Temperatura_ambiente_out);

data.setValue(2,
7,Respuesta[2].Humedad_in);

data.setValue(2,
8,Respuesta[2].Humedad_out);

data.setValue(3,
0,Respuesta[3].id);

data.setValue(3,
1,Respuesta[3].Temperatura_tanque_colectores);

data.setValue(3,
2,Respuesta[3].Temperatura_tanque_reserva);

data.setValue(3,
3,Respuesta[3].Temperatura_alfombra);

data.setValue(3,
4,Respuesta[3].Temperatura_piso);

data.setValue(3,
5,Respuesta[3].Temperatura_ambiente_in);

data.setValue(3,
6,Respuesta[3].Temperatura_ambiente_out);

data.setValue(3,
7,Respuesta[3].Humedad_in);

data.setValue(3,
8,Respuesta[3].Humedad_out);

data.setValue(4,
0,Respuesta[4].id);

data.setValue(4,
1,Respuesta[4].Temperatura_tanque_colectores);

data.setValue(4,
2,Respuesta[4].Temperatura_tanque_reserva);

data.setValue(4,
3,Respuesta[4].Temperatura_alfombra);

data.setValue(4,
4,Respuesta[4].Temperatura_piso);

data.setValue(4,
5,Respuesta[4].Temperatura_ambiente_in);

data.setValue(4,
6,Respuesta[4].Temperatura_ambiente_out);

data.setValue(4,
7,Respuesta[4].Humedad_in);

data.setValue(4,
8,Respuesta[4].Humedad_out);

```

```

        data.setValue(5,
0,Respuesta[5].id);

        data.setValue(5,
1,Respuesta[5].Temperatura_tanque_colectores);

        data.setValue(5,
2,Respuesta[5].Temperatura_tanque_reserva);

        data.setValue(5,
3,Respuesta[5].Temperatura_alfombra);

        data.setValue(5,
4,Respuesta[5].Temperatura_piso);

        data.setValue(5,
5,Respuesta[5].Temperatura_ambiente_in);

        data.setValue(5,
6,Respuesta[5].Temperatura_ambiente_out);

        data.setValue(5,
7,Respuesta[5].Humedad_in);

        data.setValue(5,
8,Respuesta[5].Humedad_out);

        data.setValue(6,
0,Respuesta[6].id);

        data.setValue(6,
1,Respuesta[6].Temperatura_tanque_colectores);

        data.setValue(6,
2,Respuesta[6].Temperatura_tanque_reserva);

        data.setValue(6,
3,Respuesta[6].Temperatura_alfombra);

        data.setValue(6,
4,Respuesta[6].Temperatura_piso);

        data.setValue(6,
5,Respuesta[6].Temperatura_ambiente_in);

        data.setValue(6,
6,Respuesta[6].Temperatura_ambiente_out);

        data.setValue(6,
7,Respuesta[6].Humedad_in);

        data.setValue(6,
8,Respuesta[6].Humedad_out);

        data.setValue(7,
0,Respuesta[7].id);

        data.setValue(7,
1,Respuesta[7].Temperatura_tanque_colectores);

        data.setValue(7,
2,Respuesta[7].Temperatura_tanque_reserva);

        data.setValue(7,
3,Respuesta[7].Temperatura_alfombra);

        data.setValue(7,
4,Respuesta[7].Temperatura_piso);

        data.setValue(7,
5,Respuesta[7].Temperatura_ambiente_in);

        data.setValue(7,
6,Respuesta[7].Temperatura_ambiente_out);

        data.setValue(7,
7,Respuesta[7].Humedad_in);

        data.setValue(7,
8,Respuesta[7].Humedad_out);

        data.setValue(9,
0,Respuesta[9].id);

        data.setValue(9,
1,Respuesta[9].Temperatura_tanque_colectores);

        data.setValue(9,
2,Respuesta[9].Temperatura_tanque_reserva);

        data.setValue(9,
3,Respuesta[9].Temperatura_alfombra);

        data.setValue(9,
4,Respuesta[9].Temperatura_piso);

        data.setValue(9,
5,Respuesta[9].Temperatura_ambiente_in);

        data.setValue(9,
6,Respuesta[9].Temperatura_ambiente_out);

        data.setValue(9,
7,Respuesta[9].Humedad_in);

        data.setValue(9,
8,Respuesta[9].Humedad_out);

        chart.draw(data, options);
    }, 1000);
}

```

```

</script>
<script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></s
cript>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.
2.0/jquery.min.js"></script>
<script type="text/javascript">
    google.charts.load('current',
    {'packages':['corechart']});
    google.charts.setOnLoadCallback(drawChart);
    function drawChart() {
        var data =
        google.visualization.arrayToDataTable([
            ['Id', 'Set_Point', 'Proceso'],
            ['2004', 1, 10],
            ['2005', 2, 9],
            ['2006', 3, 8],
            ['2007', 4, 7],
            ['2008', 5, 6],
            ['2009', 6, 5],
            ['2010', 7, 4],
            ['2015', 8, 3],
            ['2055', 9, 2],
            ['20111', 10, 111]
        ]);
        var options = {
            title: 'Set Point Temperatura VS. Variable de
            proceso',
            curveType: 'function',
            legend: { position: 'bottom' }
        };
        var chart = new
        google.visualization.LineChart(document.getEleme
        ntById('curve_chart'));
        chart.draw(data, options);
        setInterval(function() {
            var JSON=$.ajax({
                url:"DatosFuzzy.php?q=1",
                dataType: 'json',
                async: false}).responseText;
            var Respuesta=jQuery.parseJSON(JSON);
            data.setValue(9, 0,Respuesta[0].id);
            data.setValue(9, 1,Respuesta[0].set_point);
            data.setValue(9,
            2,Respuesta[0].process_value);
            data.setValue(8,
            0,Respuesta[1].id);
            data.setValue(8, 1,Respuesta[1].set_point);
            data.setValue(8,
            2,Respuesta[1].process_value);
            data.setValue(7,
            0,Respuesta[2].id);
            data.setValue(7, 1,Respuesta[2].set_point);
            data.setValue(7,
            2,Respuesta[2].process_value);
            data.setValue(6,
            0,Respuesta[3].id);
            data.setValue(6, 1,Respuesta[3].set_point);
            data.setValue(6,
            2,Respuesta[3].process_value);
            data.setValue(5,
            0,Respuesta[4].id);
            data.setValue(5, 1,Respuesta[4].set_point);
            data.setValue(5,
            2,Respuesta[4].process_value);
            data.setValue(4,
            0,Respuesta[5].id);
            data.setValue(4, 1,Respuesta[5].set_point);
            data.setValue(4,
            2,Respuesta[5].process_value);
            data.setValue(3,
            0,Respuesta[6].id);
            data.setValue(3, 1,Respuesta[6].set_point);
            data.setValue(3,
            2,Respuesta[6].process_value);
            data.setValue(2,
            0,Respuesta[7].id);
            data.setValue(2, 1,Respuesta[7].set_point);
            data.setValue(2,
            2,Respuesta[7].process_value);
            data.setValue(1,
            0,Respuesta[8].id);
            data.setValue(1, 1,Respuesta[8].set_point);
            data.setValue(1,
            2,Respuesta[8].process_value);

```

```

        data.setValue(0,
0,Respuesta[9].id);
        data.setValue(0, 1,Respuesta[9].set_point);
        data.setValue(0,
2,Respuesta[9].process_value);
        chart.draw(data, options);
    }, 1000);
}
</script>
<body>
<div id="columnchart_material"></div>
<div id="curve_chart"></div>
</body>
<a href="/tesis/cierre.php" class="button">Cerrar
sesión</a>
</html>

```

B. Código de programación en Arduino:

Anexo 12: MIMO_Fuzzy_CONTROL.ino

```

////RELOJ////
#include <Wire.h>
#include "RTClib.h"
////RELOJ////
////wemos_d1_mini////
#include <ESP8266WiFi.h>
WiFiClient client;
const char* ssid = "RED SABINA";
const char* password = "ciscocisco";
const char* host = "192.168.0.110";
unsigned long lastAttemptTime = 0;
int contador=0;
IPAddress ip(192,168,0,111);
IPAddress gateway(192,168,0,1);
IPAddress subnet(255,255,255,0);
////wemos_d1_mini////
////ds18b20////
#include <OneWire.h>
#include <DallasTemperature.h>

```

```

OneWire ourWire(2); //Se establece el pin 2 como
bus OneWire
DallasTemperature sensors(&ourWire); //Se declara
una variable u objeto para nuestro sensor
DeviceAddress address1 = {0x28, 0xA8, 0x48,
0x79, 0xA2, 0x16, 0x3, 0x12}; //sensor temperatura
tanque colectores
DeviceAddress address2 = {0x28, 0x3A, 0x27,
0x79, 0xA2, 0x16, 0x3, 0xB3}; //sensor temperatura
tanque reserva
DeviceAddress address3 = {0x28, 0xC4, 0x2B,
0x79, 0xA2, 0x16, 0x3, 0xF3}; //sensor
Temperatura Alfombra
DeviceAddress address4 = {0x28, 0xFF, 0x20, 0xF,
0xA1, 0x17, 0x4, 0x94}; //sensor Temperatura Piso
////ds18b20////
////DHT-11////
#include "DHT.h"
#define DHTPIN 13 // Digital pin connected to the
DHT sensor IN
#define DHTTYPE DHT11 // DHT 11
#define DHTPIN2 0 // Digital pin connected to the
DHT sensor OUT
#define DHTTYPE2 DHT11 // DHT 11
DHT dht(DHTPIN, DHTTYPE);
DHT dht2(DHTPIN2, DHTTYPE2);
////DHT-11////
////////FUZZY LOGIC////////
#include <Fuzzy.h>
#include <FuzzyComposition.h>
#include <FuzzyInput.h>
#include <FuzzyIO.h>
#include <FuzzyOutput.h>
#include <FuzzyRule.h>
#include <FuzzyRuleAntecedent.h>
#include <FuzzyRuleConsequent.h>
#include <FuzzySet.h>
Fuzzy *fuzzy = new Fuzzy(); //objeto fuzzy de la
clase Fuzzy y luego se inicializa
float sensor = 0;
float sensor_hum = 0;
float referencia = 0;
float referencia_hum = 0;

```

```

////////FUZZY LOGIC////////
////////PIN_PWM////////

int const salidaControl = 12; //sale el voltaje de
control hacia el motor

int const salidaVentilador = 16; //sale el voltaje de
control hacia el motor

////////PIN_PWM////////

////RELOJ/////

int dia_inicio=13;

int mes_inicio=1;

int dias_total=0;

int meses[13] = {0, 31, 29, 31, 30, 31, 30, 31, 31,
30, 31, 30, 31};

RTC_DS3231 rtc;

char daysOfTheWeek[7][12] = {"Domingo",
"Lunes", "Martes", "Miércoles", "Jueves",
"Viernes", "Sábado"};

////RELOJ/////

void setup()
{
    ////RELOJ/////

    Serial.begin(9600);

    if (! rtc.begin()) {
        Serial.println("Couldn't find RTC");
        while (1);
    }

    if (rtc.lostPower()) {
        Serial.println("RTC lost power, lets set the
time!");
        rtc.adjust(DateTime(F(__DATE__),
F(__TIME__)));
    }

    ////RELOJ/////

    //////////PWM////////

    pinMode(salidaControl,OUTPUT);
    pinMode(salidaVentilador,OUTPUT);

    //////////PWM////////

    ////wemos_d1_mini/////

    // We start by connecting to a WiFi network
    Serial.println();

    Serial.println();

    Serial.print("Connecting to ");

    Serial.println(ssid);

    WiFi.mode(WIFI_STA);

    WiFi.config(ip, gateway, subnet);

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("");

    Serial.println("WiFi connected");

    Serial.println("IP address: ");

    Serial.println(WiFi.localIP());

    ////wemos_d1_mini/////

    ////DHT-11////

    dht.begin();

    dht2.begin();

    ////DHT-11////

    ////ds18b20////////

    sensors.begin(); //Se inicia el sensor

    ////ds18b20////////

    //////////FUZZY LOGIC////////

    ////ENTRADA sensor Temperatura Piso////

    FuzzyInput* error = new FuzzyInput(1);

    FuzzySet* ENG = new FuzzySet(-6.095, -6.095, -
4.065, -2.035); //función de membresía trapezoidal
error -> addFuzzySet(ENG);

    FuzzySet* ENP = new FuzzySet(-4.065, -2.035, -
2.035, 0); //función de membresía triangular
error -> addFuzzySet(ENP);

    FuzzySet* EC = new FuzzySet(-2.035, 0, 0,
2.025); //función de membresía triangular
error -> addFuzzySet(EC);

    FuzzySet* EPP = new FuzzySet(0, 2.025, 2.025,
4.055); //función de membresía triangular
error -> addFuzzySet(EPP);

    FuzzySet* EPG = new FuzzySet(2.025, 4.055,
6.095, 6.095); //función de membresía trapezoidal

```

```

error -> addFuzzySet(EPG);

fuzzy -> addFuzzyInput(error);

/////ENTRADA sensor HUMEDAD IN/////

FuzzyInput* error_humedad = new FuzzyInput(2);

FuzzySet* HNG = new FuzzySet(-12.5, -12.5, -
8.34, -4.17); //función de membresía trapezoidal

error_humedad -> addFuzzySet(HNG);

FuzzySet* HNP = new FuzzySet(-8.34, -4.17, -
4.17, 0); //función de membresía triangular

error_humedad -> addFuzzySet(HNP);

FuzzySet* HC = new FuzzySet(-4.17, 0, 0, 4.16);
//función de membresía triangular

error_humedad -> addFuzzySet(HC);

FuzzySet* HPP = new FuzzySet(0, 4.16, 4.16,
8.33); //función de membresía triangular

error_humedad -> addFuzzySet(HPP);

FuzzySet* HPG = new FuzzySet(4.16, 8.33, 12.5,
12.5); //función de membresía trapezoidal

error_humedad -> addFuzzySet(HPG);

fuzzy -> addFuzzyInput(error_humedad);

/////SALIDA tiempo_bomba/////

FuzzyOutput* voltaje = new FuzzyOutput(1);

FuzzySet* VNG = new FuzzySet(-12.5, -12.5, -
8.34, -4.17); //función de membresía trapezoidal

voltaje -> addFuzzySet(VNG);

FuzzySet* VNP = new FuzzySet(-8.34, -4.17, -
4.17, 0); //función de membresía triangular

voltaje -> addFuzzySet(VNP);

FuzzySet* VC = new FuzzySet(-4.17, 0, 0, 4.16);
//función de membresía triangular

voltaje -> addFuzzySet(VC);

FuzzySet* VPP = new FuzzySet(0, 4.16, 4.16,
8.33); //función de membresía triangular

voltaje -> addFuzzySet(VPP);

FuzzySet* VPG = new FuzzySet(4.16, 8.33, 12.5,
12.5); //función de membresía trapezoidal

voltaje -> addFuzzySet(VPG);

fuzzy -> addFuzzyOutput(voltaje);

/////SALIDA ventilador/////

FuzzyOutput* ventilador = new FuzzyOutput(2);

FuzzySet* BNG = new FuzzySet(-261.5, -261.5, -
174.3333333, -87.16666667); //función de
membresía trapezoidal

```

```

ventilador -> addFuzzySet(BNG);

FuzzySet* BNP = new FuzzySet(-174.3333333, -
87.16666667, -87.16666667, 0); //función de
membresía triangular

ventilador -> addFuzzySet(BNP);

FuzzySet* BC = new FuzzySet(-87.16666667, 0,
0, 87.16666667); //función de membresía triangular

ventilador -> addFuzzySet(BC);

FuzzySet* BPP = new FuzzySet(0, 87.16666667,
87.16666667, 174.3333333); //función de
membresía triangular

ventilador -> addFuzzySet(BPP);

FuzzySet* BPG = new FuzzySet(87.16666667,
174.3333333, 261.5, 261.5); //función de
membresía trapezoidal

ventilador -> addFuzzySet(BPG);

fuzzy -> addFuzzyOutput(ventilador);

////////////////REGLAS DUFUSAS////////////////

//////////////// 1 //////////////////

FuzzyRuleAntecedent*
siErrorNegativoGrandeANDHumedadNegativoGra
nde = new FuzzyRuleAntecedent();

siErrorNegativoGrandeANDHumedadNegativoGra
nde -> joinWithAND(ENG, HNG);

FuzzyRuleConsequent*
entoncesVoltajeNegativoGrandeANDBitsNegativo
Grande = new FuzzyRuleConsequent();

entoncesVoltajeNegativoGrandeANDBitsNegativo
Grande -> addOutput(VNG);

entoncesVoltajeNegativoGrandeANDBitsNegativo
Grande -> addOutput(BNG);

FuzzyRule* reglaDifusa1 = new FuzzyRule(1,
siErrorNegativoGrandeANDHumedadNegativoGra
nde,
entoncesVoltajeNegativoGrandeANDBitsNegativo
Grande);

fuzzy -> addFuzzyRule(reglaDifusa1);

//////////////// 2 //////////////////

FuzzyRuleAntecedent*
siErrorNegativoGrandeANDHumedadNegativoPeq
ueno = new FuzzyRuleAntecedent();

siErrorNegativoGrandeANDHumedadNegativoPeq
ueno -> joinWithAND(ENG, HNP);

```

```
FuzzyRuleConsequent*
entoncesVoltajeNegativoGrandeANDBitsNegativo
Pequeno = new FuzzyRuleConsequent();
```

```
entoncesVoltajeNegativoGrandeANDBitsNegativo
Pequeno -> addOutput(VNG);
```

```
entoncesVoltajeNegativoGrandeANDBitsNegativo
Pequeno -> addOutput(BNP);
```

```
FuzzyRule* reglaDifusa2 = new FuzzyRule(2,
siErrorNegativoGrandeANDHumedadNegativoPequeno,
entoncesVoltajeNegativoGrandeANDBitsNegativo
Pequeno);
```

```
fuzzy -> addFuzzyRule(reglaDifusa2);
```

```
//////////////////// 3 //////////////////
```

```
FuzzyRuleAntecedent*
siErrorNegativoGrandeANDHumedadCero = new
FuzzyRuleAntecedent();
```

```
siErrorNegativoGrandeANDHumedadCero ->
joinWithAND(ENG, HC);
```

```
FuzzyRuleConsequent*
entoncesVoltajeNegativoGrandeANDBitsCero =
new FuzzyRuleConsequent();
```

```
entoncesVoltajeNegativoGrandeANDBitsCero ->
addOutput(VNG);
```

```
entoncesVoltajeNegativoGrandeANDBitsCero ->
addOutput(BC);
```

```
FuzzyRule* reglaDifusa3 = new FuzzyRule(3,
siErrorNegativoGrandeANDHumedadCero,
entoncesVoltajeNegativoGrandeANDBitsCero);
```

```
fuzzy -> addFuzzyRule(reglaDifusa3);
```

```
//////////////////// 4 //////////////////
```

```
FuzzyRuleAntecedent*
siErrorNegativoGrandeANDHumedadPositivoPequeno = new FuzzyRuleAntecedent();
```

```
siErrorNegativoGrandeANDHumedadPositivoPequeno -> joinWithAND(ENG, HPP);
```

```
FuzzyRuleConsequent*
entoncesVoltajeNegativoGrandeANDBitsPositivoPequeno = new FuzzyRuleConsequent();
```

```
entoncesVoltajeNegativoGrandeANDBitsPositivoPequeno -> addOutput(VNG);
```

```
entoncesVoltajeNegativoGrandeANDBitsPositivoPequeno -> addOutput(BPP);
```

```
FuzzyRule* reglaDifusa4 = new FuzzyRule(4,
siErrorNegativoGrandeANDHumedadPositivoPequeno,
```

```
entoncesVoltajeNegativoGrandeANDBitsPositivoPequeno);
```

```
fuzzy -> addFuzzyRule(reglaDifusa4);
```

```
//////////////////// 5 //////////////////
```

```
FuzzyRuleAntecedent*
siErrorNegativoGrandeANDHumedadPositivoGrande = new FuzzyRuleAntecedent();
```

```
siErrorNegativoGrandeANDHumedadPositivoGrande -> joinWithAND(ENG, HPG);
```

```
FuzzyRuleConsequent*
entoncesVoltajeNegativoGrandeANDBitsPositivoGrande = new FuzzyRuleConsequent();
```

```
entoncesVoltajeNegativoGrandeANDBitsPositivoGrande -> addOutput(VNG);
```

```
entoncesVoltajeNegativoGrandeANDBitsPositivoGrande -> addOutput(BPG);
```

```
FuzzyRule* reglaDifusa5 = new FuzzyRule(5,
siErrorNegativoGrandeANDHumedadPositivoGrande,
entoncesVoltajeNegativoGrandeANDBitsPositivoGrande);
```

```
fuzzy -> addFuzzyRule(reglaDifusa5);
```

```
//////////////////// 6 //////////////////
```

```
FuzzyRuleAntecedent*
siErrorNegativoPequenoANDHumedadNegativoGrande = new FuzzyRuleAntecedent();
```

```
siErrorNegativoPequenoANDHumedadNegativoGrande -> joinWithAND(ENP, HNG);
```

```
FuzzyRuleConsequent*
entoncesVoltajeNegativoPequenoANDBitsNegativoGrande = new FuzzyRuleConsequent();
```

```
entoncesVoltajeNegativoPequenoANDBitsNegativoGrande -> addOutput(VNP);
```

```
entoncesVoltajeNegativoPequenoANDBitsNegativoGrande -> addOutput(BNG);
```

```
FuzzyRule* reglaDifusa6 = new FuzzyRule(6,
siErrorNegativoPequenoANDHumedadNegativoGrande,
entoncesVoltajeNegativoPequenoANDBitsNegativoGrande);
```

```
fuzzy -> addFuzzyRule(reglaDifusa6);
```

```
//////////////////// 7 //////////////////
```

```
FuzzyRuleAntecedent*
siErrorNegativoPequenoANDHumedadNegativoPequeno = new FuzzyRuleAntecedent();
```

```
siErrorNegativoPequenoANDHumedadNegativoPequeno -> joinWithAND(ENP, HNP);
```

```
FuzzyRuleConsequent*  
entoncesVoltajeNegativoPequenoANDBitsNegativoPequeno = new FuzzyRuleConsequent();
```

```
entoncesVoltajeNegativoPequenoANDBitsNegativoPequeno -> addOutput(VNP);
```

```
entoncesVoltajeNegativoPequenoANDBitsNegativoPequeno -> addOutput(BNP);
```

```
FuzzyRule* reglaDifusa7 = new FuzzyRule(7,  
siErrorNegativoPequenoANDHumedadNegativoPequeno,  
entoncesVoltajeNegativoPequenoANDBitsNegativoPequeno);
```

```
fuzzy -> addFuzzyRule(reglaDifusa7);
```

```
//////////////////// 8 //////////////////
```

```
FuzzyRuleAntecedent*  
siErrorNegativoPequenoANDHumedadCero = new FuzzyRuleAntecedent();
```

```
siErrorNegativoPequenoANDHumedadCero ->  
joinWithAND(ENP, HC);
```

```
FuzzyRuleConsequent*  
entoncesVoltajeNegativoPequenoANDBitsCero =  
new FuzzyRuleConsequent();
```

```
entoncesVoltajeNegativoPequenoANDBitsCero ->  
addOutput(VNP);
```

```
entoncesVoltajeNegativoPequenoANDBitsCero ->  
addOutput(BC);
```

```
FuzzyRule* reglaDifusa8 = new FuzzyRule(8,  
siErrorNegativoPequenoANDHumedadCero,  
entoncesVoltajeNegativoPequenoANDBitsCero);
```

```
fuzzy -> addFuzzyRule(reglaDifusa8);
```

```
//////////////////// 9 //////////////////
```

```
FuzzyRuleAntecedent*  
siErrorNegativoPequenoANDHumedadPositivoPequeno = new FuzzyRuleAntecedent();
```

```
siErrorNegativoPequenoANDHumedadPositivoPequeno -> joinWithAND(ENG, HPP);
```

```
FuzzyRuleConsequent*  
entoncesVoltajeNegativoPequenoANDBitsPositivoPequeno = new FuzzyRuleConsequent();
```

```
entoncesVoltajeNegativoPequenoANDBitsPositivoPequeno -> addOutput(VNP);
```

```
entoncesVoltajeNegativoPequenoANDBitsPositivoPequeno -> addOutput(BPP);
```

```
FuzzyRule* reglaDifusa9 = new FuzzyRule(9,  
siErrorNegativoPequenoANDHumedadPositivoPequeno,  
entoncesVoltajeNegativoPequenoANDBitsPositivoPequeno);
```

```
fuzzy -> addFuzzyRule(reglaDifusa9);
```

```
//////////////////// 10 //////////////////
```

```
FuzzyRuleAntecedent*  
siErrorNegativoPequenoANDHumedadPositivoGrande = new FuzzyRuleAntecedent();
```

```
siErrorNegativoPequenoANDHumedadPositivoGrande -> joinWithAND(ENP, HPG);
```

```
FuzzyRuleConsequent*  
entoncesVoltajeNegativoPequenoANDBitsPositivoGrande = new FuzzyRuleConsequent();
```

```
entoncesVoltajeNegativoPequenoANDBitsPositivoGrande -> addOutput(VNP);
```

```
entoncesVoltajeNegativoPequenoANDBitsPositivoGrande -> addOutput(BPG);
```

```
FuzzyRule* reglaDifusa10 = new FuzzyRule(10,  
siErrorNegativoPequenoANDHumedadPositivoGrande,  
entoncesVoltajeNegativoPequenoANDBitsPositivoGrande);
```

```
fuzzy -> addFuzzyRule(reglaDifusa10);
```

```
//////////////////// 11 //////////////////
```

```
FuzzyRuleAntecedent*  
siErrorCeroANDHumedadNegativoGrande = new FuzzyRuleAntecedent();
```

```
siErrorCeroANDHumedadNegativoGrande ->  
joinWithAND(EC, HNG);
```

```
FuzzyRuleConsequent*  
entoncesVoltajeCeroANDBitsNegativoGrande =  
new FuzzyRuleConsequent();
```

```
entoncesVoltajeCeroANDBitsNegativoGrande ->  
addOutput(VC);
```

```
entoncesVoltajeCeroANDBitsNegativoGrande ->  
addOutput(BNG);
```

```
FuzzyRule* reglaDifusa11 = new FuzzyRule(11,  
siErrorCeroANDHumedadNegativoGrande,  
entoncesVoltajeCeroANDBitsNegativoGrande);
```

```
fuzzy -> addFuzzyRule(reglaDifusa11);
```

```
//////////////////// 12 //////////////////
```

```
FuzzyRuleAntecedent*  
siErrorCeroANDHumedadNegativoPequeno = new FuzzyRuleAntecedent();
```



```

siErrorCeroANDHumedadNegativoPequeno ->
joinWithAND(EC, HNP);

FuzzyRuleConsequent*
entoncesVoltajeCeroANDBitsNegativoPequeno =
new FuzzyRuleConsequent();

entoncesVoltajeCeroANDBitsNegativoPequeno ->
addOutput(VC);

entoncesVoltajeCeroANDBitsNegativoPequeno ->
addOutput(BNP);

FuzzyRule* reglaDifusa12 = new FuzzyRule(12,
siErrorCeroANDHumedadNegativoPequeno,
entoncesVoltajeCeroANDBitsNegativoPequeno);

fuzzy -> addFuzzyRule(reglaDifusa12);

//////////////////// 13 //////////////////////

FuzzyRuleAntecedent*
siErrorCeroANDHumedadCero = new
FuzzyRuleAntecedent();

siErrorCeroANDHumedadCero ->
joinWithAND(EC, HC);

FuzzyRuleConsequent*
entoncesVoltajeCeroANDBitsCero = new
FuzzyRuleConsequent();

entoncesVoltajeCeroANDBitsCero ->
addOutput(VC);

entoncesVoltajeCeroANDBitsCero ->
addOutput(BC);

FuzzyRule* reglaDifusa13 = new FuzzyRule(13,
siErrorCeroANDHumedadCero,
entoncesVoltajeCeroANDBitsCero);

fuzzy -> addFuzzyRule(reglaDifusa13);

//////////////////// 14 //////////////////////

FuzzyRuleAntecedent*
siErrorCeroANDHumedadPositivoPequeno = new
FuzzyRuleAntecedent();

siErrorCeroANDHumedadPositivoPequeno ->
joinWithAND(EC, HPP);

FuzzyRuleConsequent*
entoncesVoltajeCeroANDBitsPositivoPequeno =
new FuzzyRuleConsequent();

entoncesVoltajeCeroANDBitsPositivoPequeno ->
addOutput(VC);

entoncesVoltajeCeroANDBitsPositivoPequeno ->
addOutput(BPP);

FuzzyRule* reglaDifusa14 = new FuzzyRule(14,
siErrorCeroANDHumedadPositivoPequeno,
entoncesVoltajeCeroANDBitsPositivoPequeno);

fuzzy -> addFuzzyRule(reglaDifusa14);

//////////////////// 15 //////////////////////

```

```

FuzzyRuleAntecedent*
siErrorCeroANDHumedadPositivoGrande = new
FuzzyRuleAntecedent();

siErrorCeroANDHumedadPositivoGrande ->
joinWithAND(EC, HPG);

FuzzyRuleConsequent*
entoncesVoltajeCeroANDBitsPositivoGrande =
new FuzzyRuleConsequent();

entoncesVoltajeCeroANDBitsPositivoGrande ->
addOutput(VC);

entoncesVoltajeCeroANDBitsPositivoGrande ->
addOutput(BPG);

FuzzyRule* reglaDifusa15 = new FuzzyRule(15,
siErrorCeroANDHumedadPositivoGrande,
entoncesVoltajeCeroANDBitsPositivoGrande);

fuzzy -> addFuzzyRule(reglaDifusa15);

//////////////////// 16 //////////////////////

FuzzyRuleAntecedent*
siErrorPositivoPequenoANDHumedadNegativoGra
nde = new FuzzyRuleAntecedent();

siErrorPositivoPequenoANDHumedadNegativoGra
nde -> joinWithAND(EPP, HNG);

FuzzyRuleConsequent*
entoncesVoltajePositivoPequenoANDBitsNegativo
Grande = new FuzzyRuleConsequent();

entoncesVoltajePositivoPequenoANDBitsNegativo
Grande -> addOutput(VPP);

entoncesVoltajePositivoPequenoANDBitsNegativo
Grande -> addOutput(BNG);

FuzzyRule* reglaDifusa16 = new FuzzyRule(16,
siErrorPositivoPequenoANDHumedadNegativoGra
nde,
entoncesVoltajePositivoPequenoANDBitsNegativo
Grande);

fuzzy -> addFuzzyRule(reglaDifusa16);

//////////////////// 17 //////////////////////

FuzzyRuleAntecedent*
siErrorPositivoPequenoANDHumedadNegativoPeq
ueno = new FuzzyRuleAntecedent();

siErrorPositivoPequenoANDHumedadNegativoPeq
ueno -> joinWithAND(EPP, HNP);

FuzzyRuleConsequent*
entoncesVoltajePositivoPequenoANDBitsNegativo
Pequeno = new FuzzyRuleConsequent();

entoncesVoltajePositivoPequenoANDBitsNegativo
Pequeno -> addOutput(VPP);

```

entonces VoltajePositivoPequenoANDBitsNegativoPequeno -> addOutput(BNP);

FuzzyRule* reglaDifusa17 = new FuzzyRule(17, siErrorPositivoPequenoANDHumedadNegativoPequeno, entonces VoltajePositivoPequenoANDBitsNegativoPequeno);

fuzzy -> addFuzzyRule(reglaDifusa17);

//////////////////// 18 //////////////////////

FuzzyRuleAntecedent* siErrorPositivoPequenoANDHumedadCero = new FuzzyRuleAntecedent();

siErrorPositivoPequenoANDHumedadCero -> joinWithAND(EPP, HC);

FuzzyRuleConsequent* entonces VoltajePositivoPequenoANDBitsCero = new FuzzyRuleConsequent();

entonces VoltajePositivoPequenoANDBitsCero -> addOutput(VPP);

entonces VoltajePositivoPequenoANDBitsCero -> addOutput(BC);

FuzzyRule* reglaDifusa18 = new FuzzyRule(18, siErrorPositivoPequenoANDHumedadCero, entonces VoltajePositivoPequenoANDBitsCero);

fuzzy -> addFuzzyRule(reglaDifusa18);

//////////////////// 19 //////////////////////

FuzzyRuleAntecedent* siErrorPositivoPequenoANDHumedadPositivoPequeno = new FuzzyRuleAntecedent();

siErrorPositivoPequenoANDHumedadPositivoPequeno -> joinWithAND(EPP, HPP);

FuzzyRuleConsequent* entonces VoltajePositivoPequenoANDBitsPositivoPequeno = new FuzzyRuleConsequent();

entonces VoltajePositivoPequenoANDBitsPositivoPequeno -> addOutput(VPP);

entonces VoltajePositivoPequenoANDBitsPositivoPequeno -> addOutput(BPP);

FuzzyRule* reglaDifusa19 = new FuzzyRule(19, siErrorPositivoPequenoANDHumedadPositivoPequeno, entonces VoltajePositivoPequenoANDBitsPositivoPequeno);

fuzzy -> addFuzzyRule(reglaDifusa19);

//////////////////// 20 //////////////////////

FuzzyRuleAntecedent* siErrorPositivoPequenoANDHumedadPositivoGrande = new FuzzyRuleAntecedent();

siErrorPositivoPequenoANDHumedadPositivoGrande -> joinWithAND(EPP, HPG);

FuzzyRuleConsequent* entonces VoltajePositivoPequenoANDBitsPositivoGrande = new FuzzyRuleConsequent();

entonces VoltajePositivoPequenoANDBitsPositivoGrande -> addOutput(VPP);

entonces VoltajePositivoPequenoANDBitsPositivoGrande -> addOutput(BPG);

FuzzyRule* reglaDifusa20 = new FuzzyRule(20, siErrorPositivoPequenoANDHumedadPositivoGrande, entonces VoltajePositivoPequenoANDBitsPositivoGrande);

fuzzy -> addFuzzyRule(reglaDifusa20);

//////////////////// 21 //////////////////////

FuzzyRuleAntecedent* siErrorPositivoGrandeANDHumedadNegativoGrande = new FuzzyRuleAntecedent();

siErrorPositivoGrandeANDHumedadNegativoGrande -> joinWithAND(EPG, HNG);

FuzzyRuleConsequent* entonces VoltajePositivoGrandeANDBitsNegativoGrande = new FuzzyRuleConsequent();

entonces VoltajePositivoGrandeANDBitsNegativoGrande -> addOutput(VPG);

entonces VoltajePositivoGrandeANDBitsNegativoGrande -> addOutput(BNG);

FuzzyRule* reglaDifusa21 = new FuzzyRule(21, siErrorPositivoGrandeANDHumedadNegativoGrande, entonces VoltajePositivoGrandeANDBitsNegativoGrande);

fuzzy -> addFuzzyRule(reglaDifusa21);

//////////////////// 22 //////////////////////

FuzzyRuleAntecedent* siErrorPositivoGrandeANDHumedadNegativoPequeno = new FuzzyRuleAntecedent();

siErrorPositivoGrandeANDHumedadNegativoPequeno -> joinWithAND(EPG, HNP);

```

FuzzyRuleConsequent*
entoncesVoltajePositivoGrandeANDBitsNegativoPequeno = new FuzzyRuleConsequent();

entoncesVoltajePositivoGrandeANDBitsNegativoPequeno -> addOutput(VPG);

entoncesVoltajePositivoGrandeANDBitsNegativoPequeno -> addOutput(BNP);

FuzzyRule* reglaDifusa22 = new FuzzyRule(22,
siErrorPositivoGrandeANDHumedadNegativoPequeno,
entoncesVoltajePositivoGrandeANDBitsNegativoPequeno);

fuzzy -> addFuzzyRule(reglaDifusa22);

//////////////////// 23 //////////////////////

FuzzyRuleAntecedent*
siErrorPositivoGrandeANDHumedadCero = new
FuzzyRuleAntecedent();

siErrorPositivoGrandeANDHumedadCero ->
joinWithAND(EPG, HC);

FuzzyRuleConsequent*
entoncesVoltajePositivoGrandeANDBitsCero =
new FuzzyRuleConsequent();

entoncesVoltajePositivoGrandeANDBitsCero ->
addOutput(VPG);

entoncesVoltajePositivoGrandeANDBitsCero ->
addOutput(BC);

FuzzyRule* reglaDifusa23 = new FuzzyRule(23,
siErrorPositivoGrandeANDHumedadCero,
entoncesVoltajePositivoGrandeANDBitsCero );

fuzzy -> addFuzzyRule(reglaDifusa23);

//////////////////// 24 //////////////////////

FuzzyRuleAntecedent*
siErrorPositivoGrandeANDHumedadPositivoPequeno = new FuzzyRuleAntecedent();

siErrorPositivoGrandeANDHumedadPositivoPequeno -> joinWithAND(EPG, HPP);

FuzzyRuleConsequent*
entoncesVoltajePositivoGrandeANDBitsPositivoPequeno = new FuzzyRuleConsequent();

entoncesVoltajePositivoGrandeANDBitsPositivoPequeno -> addOutput(VPG);

entoncesVoltajePositivoGrandeANDBitsPositivoPequeno -> addOutput(BPP);

FuzzyRule* reglaDifusa24 = new FuzzyRule(24,
siErrorPositivoGrandeANDHumedadPositivoPequeno,

```

```

entoncesVoltajePositivoGrandeANDBitsPositivoPequeno);

fuzzy -> addFuzzyRule(reglaDifusa24);

//////////////////// 25 //////////////////////

FuzzyRuleAntecedent*
siErrorPositivoGrandeANDHumedadPositivoGrande = new FuzzyRuleAntecedent();

siErrorPositivoGrandeANDHumedadPositivoGrande -> joinWithAND(EPG, HPG);

FuzzyRuleConsequent*
entoncesVoltajePositivoGrandeANDBitsPositivoGrande = new FuzzyRuleConsequent();

entoncesVoltajePositivoGrandeANDBitsPositivoGrande -> addOutput(VPG);

entoncesVoltajePositivoGrandeANDBitsPositivoGrande -> addOutput(BPG);

FuzzyRule* reglaDifusa25 = new FuzzyRule(25,
siErrorPositivoGrandeANDHumedadPositivoGrande,
entoncesVoltajePositivoGrandeANDBitsPositivoGrande);

fuzzy -> addFuzzyRule(reglaDifusa25);

////////////////FUZZY LOGIC////////////////////

}

void loop()
{
    //////////////////RELOJ////////

    DateTime now = rtc.now();

    if(mes_inicio!=now.month()){

        dias_total=meses[mes_inicio]-
        dia_inicio+now.day();

        for(int i=now.month()-1; i>mes_inicio; i--){

            dias_total = dias_total+meses[i];

        }

    }else{

        dias_total=now.day()-dia_inicio;

    }

    //////////////////RELOJ////////

    //////////////////REFERENCIAS////////

    if(dias_total>=1&&dias_total<=2){

        referencia = 31;

```

```

referencia_hum = 37.5;
}
if(dias_total>=3&&dias_total<=7){
referencia=29.5;
referencia_hum = 40;
}
if(dias_total>=8&&dias_total<=14){
referencia=28;
referencia_hum = 42.5;
}
if(dias_total>=15&&dias_total<=21){
referencia=26;
referencia_hum = 47.5;
}
if(dias_total>=22&&dias_total<=28){
referencia=24;
referencia_hum = 52.5;
}
if(dias_total>=29){
referencia=22;
referencia_hum = 57.5;
}
}
//////////REFERENCIAS//////////
float* tempe_ds18b20 = temperatura_ds18b20();
float* datos_dht11 = dht11();
contador++;
if(contador%5==0){
connectToServer("Temperatura_tanque_colectores"
, "Temperatura_tanque_reserva",
"Temperatura_alfombra", "Temperatura_piso",
"Temperatura_ambiente_in",
"Temperatura_ambiente_out", "Humedad_in",
"Humedad_out",
tempe_ds18b20[0], tempe_ds18b20[1],
tempe_ds18b20[2], tempe_ds18b20[3],
datos_dht11[1], datos_dht11[3], datos_dht11[0],
datos_dht11[2]);
connectToServer2("set_point", "process_value",
referencia, tempe_ds18b20[2]);
contador=0;
}
}
//////////FUZZY LOGIC//////////
sensor = tempe_ds18b20[2];
float e = referencia - sensor;
fuzzy ->setInput(1, e);
fuzzy -> fuzzify();
float v = fuzzy -> defuzzify(1);
int delay_t = v*1000;
if(v<0){
if(sensor==referencia){
digitalWrite(salidaVentilador, LOW);
digitalWrite(salidaControl, LOW);
Serial.print((String)"referencia = "+referencia+"
sensor = "+sensor+" tiempo = "+v+" error = "+e+"
tiempo_ms = "+delay_t);
Serial.println();
}
}
else{
digitalWrite(salidaVentilador, HIGH);
digitalWrite(salidaControl, LOW);
Serial.print((String)"referencia = "+referencia+"
sensor = "+sensor+" tiempo = "+v+" error = "+e+"
tiempo_ms = "+delay_t);
Serial.println();
}
}
}
else{
digitalWrite(salidaVentilador, LOW);
digitalWrite(salidaControl, HIGH);
delay(delay_t);
digitalWrite(salidaControl, LOW);
for(int i=0; i<20; i++){
//////////php/////
tempe_ds18b20 = temperatura_ds18b20();
datos_dht11 = dht11();
contador++;
if(contador%5==0){
connectToServer("Temperatura_tanque_colectores"
, "Temperatura_tanque_reserva",
"Temperatura_alfombra", "Temperatura_piso",
"Temperatura_ambiente_in",
"Temperatura_ambiente_out", "Humedad_in",
"Humedad_out",
}
}
}
}

```

```

    tempe_ds18b20[0], tempe_ds18b20[1],
    tempe_ds18b20[2], tempe_ds18b20[3],
    datos_dht11[1], datos_dht11[3], datos_dht11[0],
    datos_dht11[2]);

    connectToServer2("set_point", "process_value",
referencia, tempe_ds18b20[2]);

    contador=0;
}

if(tempe_ds18b20[2]==referencia){

break;

}

//////////php//////////

delay(delay_t);

Serial.print((String)referencia = "+referencia+"
sensor = "+tempe_ds18b20[2]+" tiempo = "+v+"
error = "+e+" tiempo_ms = "+delay_t+" FORi =
"+i);

Serial.println();
}

}

sensor_hum = datos_dht11[0];

float e_hum = referencia_hum - sensor_hum;

fuzzy ->setInput(2, e_hum);

fuzzy -> fuzzify();

float v_hum = fuzzy -> defuzzify(2);

analogWrite(salidaVentilador, v_hum);

//////////FUZZY LOGIC//////////

}

void connectToServer(String variable1, String
variable2, String variable3, String variable4, String
variable5, String variable6, String variable7, String
variable8, double valor_1, double valor_2, double
valor_3, double valor_4, double valor_5, double
valor_6, double valor_7, double valor_8) {

if (client.connect(host, 80)) {

client.print("GET
/tesis/uploads/temperatura.php?"+variable1+"=");

client.print(valor_1);

client.print("&"+variable2+"=");

client.print(valor_2);

client.print("&"+variable3+"=");

client.print(valor_3);

client.print("&"+variable4+"=");

client.print(valor_4);

client.print("&"+variable5+"=");

client.print(valor_5);

client.print("&"+variable6+"=");

client.print(valor_6);

client.print("&"+variable7+"=");

client.print(valor_7);

client.print("&"+variable8+"=");

client.print(valor_8);

client.println(" HTTP/1.1");

client.println("HOST: "+(String)host);

client.println();

}

//Serial.println((String)host+"/tesis/uploads/tempera
tura.php?"+variable1+"="+valor_1+"&"+variable2
+"="+valor_2+"&"+variable3+"="+valor_3+"&"+v
ariable4+"="+valor_4+variable5+"="+valor_5+"&
"+variable6+"="+valor_6+"&"+variable7+"="+valor
_7+"&"+variable8+"="+valor_8);

}

// note the time of this connect attempt:

lastAttemptTime = millis();

client.stop();

client.flush();

}

void connectToServer2(String variable1, String
variable2, double valor_1, double valor_2) {

if (client.connect(host, 80)) {

//Serial.println("making HTTP request...");

//Lo interesante, apuntar a la página por donde
mando los datos

//En este caso apuntar a temperatura.php OJO

client.print("GET
/tesis/uploads/fuzzy_upload.php?"+variable1+"=");

client.print(valor_1);

client.print("&"+variable2+"=");

client.print(valor_2);

client.println(" HTTP/1.1");

client.println("HOST: "+(String)host);

client.println();

}

```

```

// note the time of this connect attempt:
lastAttemptTime = millis();

client.stop();

client.flush();

}

float* temperatura_ds18b20(){

float* array_temp = new float[4];

sensors.requestTemperatures(); //envía el comando
para obtener las temperaturas

array_temp[0] = sensors.getTempC(address1);//Se
obtiene la temperatura en °C del sensor 1
Temperatura Tanque Colectores

array_temp[1] = sensors.getTempC(address2);//Se
obtiene la temperatura en °C del sensor 2
Temperatura Tanque Reserva

array_temp[2] = sensors.getTempC(address3);//Se
obtiene la temperatura en °C del sensor 3
Temperatura Alfombra

array_temp[3] = sensors.getTempC(address4);//Se
obtiene la temperatura en °C del sensor 4
Temperatura Piso

return array_temp;

}

float* dht11(){

float* array_dht11 = new float[4];

array_dht11[0] = dht.readHumidity(); //EN
PORCENTAJE Humedad In

array_dht11[1] = dht.readTemperature(); //EN °C
Temperatura Ambiente In

array_dht11[2] = dht2.readHumidity(); //EN
PORCENTAJE Humedad out

array_dht11[3] = dht2.readTemperature(); //EN °C
Temperatura Ambiente out

return array_dht11;

}

```

Anexo B. Fotografías de la instalación.



