



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE TELECOMUNICACIONES

TRABAJO DE INTEGRACIÓN CURRICULAR

TEMA:

“SISTEMA AUTOMATIZADO BASADO EN UNA ARQUITECTURA IOT PARA LA
GESTIÓN Y MONITOREO DEL AMBIENTE DE UN GALPÓN DE CRianza DE
POLLOS EN LA AVÍCOLA AVICAMP”

**Trabajo de titulación previo a la obtención del título de
Ingeniero en Telecomunicaciones**

Línea de investigación: Producción industrial y tecnología sostenible

AUTOR:

Carlos Santiago Chuquilla Guagala

DIRECTOR:

MSc. Fabián Geovanny Cuzme Rodríguez

Ibarra-Ecuador 2026

UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITAR

AUTORIZACIÓN DE USO Y PUBLICACIÓN A
FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte Para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información.

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1003755087		
APELLIDOS Y NOMBRES:	CHUQUILLA GUAGALA CARLOS SANTIAGO		
DIRECCIÓN:	Rafael Larrea 5-26 y Bartolomé García		
EMAIL:	cschuquillag@utn.edu.ec		
TELÉFONO FIJO:	No registra	TELF. MOVIL	0982260095

DATOS DE LA OBRA	
TÍTULO:	SISTEMA AUTOMATIZADO BASADO EN UNA ARQUITECTURA IOT PARA LA GESTIÓN Y MONITOREO DEL AMBIENTE DE UN GALPÓN DE CRIANZA DE POLLOS EN LA AVÍCOLA AVICAMP
AUTOR (ES):	CARLOS SANTIAGO CHUQUILLA GUAGALA
FECHA: DD-MM-AAAA	29-01-2026
SOLO PARA TRABAJOS DE INTEGRACIÓN CURRICULAR	
CARRERA/PROGRAMA:	☒ PREGRADO ☐ POSGRADO
TITULO POR EL QUE OPTA:	INGENIERO EN TELECOMUNICACIONES.
DIRECTOR:	MSC. FABIÁN GEOVANNY CUZME RODRÍGUEZ
ASESOR:	MSC. CARLOS ALBERTO VÁSQUEZ AYALA

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de esta y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 29 días, del mes de enero de 2026

EL AUTOR:



Carlos Santiago Chuquilla Guagala

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 29 de enero de 2026

MSc. Fabián Geovanny Cuzme Rodríguez

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

(f).....Firmado electrónicamente por:
FABIAN GEOVANNY
CUZME RODRIGUEZ
Validar únicamente con FirmaEC

MSC. FABIÁN GEOVANNY CUZME RODRÍGUEZ

C.C. 1311527012

DEDICATORIA

A Dios, fuente infinita de sabiduría y amor, por ser mi guía en cada decisión, mi refugio en las dificultades y la fortaleza que me sostuvo cuando las fuerzas parecían agotarse. A Él encomiendo cada paso de este camino y cada logro alcanzado, porque sin su gracia nada hubiera sido posible.

A mis padres, ejemplo de entrega y sacrificio, por brindarme un amor que no conoce límites y enseñarme, con su esfuerzo diario, que la perseverancia es la llave que abre las puertas del futuro. Este trabajo es reflejo de sus enseñanzas, de cada consejo y de cada gesto de apoyo que ha forjado mi carácter y mis sueños.

A mi familia, por ser compañía constante, por alentarme con palabras y abrazos en los momentos de desaliento y por compartir conmigo la esperanza en los días difíciles. Su presencia ha sido un recordatorio de que no camino solo, de que mis triunfos también son los suyos.

Y a mi novia, compañera fiel de este viaje, por su paciencia infinita, su ternura y su apoyo incondicional. Gracias por creer en mí incluso en los días en que dudaba de mis propias fuerzas, por celebrar cada pequeño avance y por caminar a mi lado con amor y comprensión. Su compañía ha dado luz y sentido a este proceso, convirtiendo cada logro en un triunfo compartido.

AGRADECIMIENTO

A Dios, por haberme concedido la salud, la sabiduría y la perseverancia necesarias para culminar esta etapa de mi formación profesional, guiando mis pasos en cada momento de duda y brindándome fortaleza en los momentos más difíciles.

A mi director de tesis, por su orientación, compromiso y acompañamiento constante durante el desarrollo de este trabajo. Su experiencia, exigencia y criterio fueron fundamentales para la construcción técnica y académica de este proyecto.

A mi asesor de tesis, por su valioso apoyo, por compartir sus conocimientos con dedicación y por haberse convertido en un pilar clave en el proceso de investigación y desarrollo.

A mi familia, por su apoyo incondicional, por la motivación brindada y por alentarme en todo momento para alcanzar este objetivo.

Y a todas las personas que, de una u otra manera, contribuyeron con su tiempo, consejos o colaboración, mi más sincero agradecimiento.

INDICE DE CONTENIDO

Capítulo I: Antecedentes.....	16
1.1. Tema	16
1.2. Problema	16
1.3. Objetivos.....	17
1.3.1. Objetivo General.....	17
1.3.2. Objetivos Específicos.....	17
1.4. Alcance	18
1.5. Justificación	20
Capítulo II: Fundamentación Teórica	22
2.1. IoT.....	22
2.1.2. Arquitectura IoT.....	23
2.1.2.1. Capa de Sensores	24
2.1.2.2. Capa Conectividad.....	24
2.1.2.3. Capa Nube.....	25
2.1.2.4. Capa de aplicación	25
2.1.3. Tecnologías de comunicación inalámbrica para IoT	26
2.1.3.2. Sigfox.....	27
2.1.3.3. LoRaWan	27
2.1.3.4. Narrowband IoT	28
2.1.3.5. BLE	28

2.1.3.6. Zigbee	29
2.1.3.7 WIFI.....	29
2.2. Crianza de pollos en galpones.....	31
2.2.1. Diseño y Estructura del Galpón	31
2.2.2. Preparación del Área de Crianza.....	32
2.2.3. Alimentación y Nutrición	33
2.2.4. Manejo de la Salud en aves.....	34
2.3. Automatización en el proceso de crianza de pollos	34
2.3.1. Sistemas de Alimentación Automatizados.....	34
2.3.2. Sistemas de Bebida Automatizados	35
2.3.3. Control del Ambiente.....	35
2.3.4. Monitoreo y Gestión de Datos	36
2.3.5. Manejo de Salud y Bioseguridad	36
2.3.6. Beneficios de la Crianza Automatizada	37
2.4. Lógica difusa.....	37
2.4.1. Teoría de conjuntos difusos	38
2.4.2. Sistemas Difusos.....	38
2.4.3. Lógica difusa en Control Ambiental.....	41
2.4.4. Alimentación Automatizada con lógica difusa	41
2.4.5. Monitoreo de Salud y Bienestar.....	42
Capítulo III: Metodología	43

3.1. Análisis del Proyecto	43
3.1.1. Situación actual.....	44
3.2. Requerimientos del Sistema.....	48
3.2.1. Listado de Stakeholders	49
3.2.2. Requerimientos Operacionales	50
3.2.3. Requerimientos del sistema SySR	51
3.2.4. Requerimientos de Arquitectura	53
3.3. Elección de Hardware y Software.....	55
3.3.1. Hardware.....	56
3.3.2 Software	71
3.4. Diseño del Sistema.....	78
3.4.1 Diagrama de Bloques del Sistema de Monitoreo.....	79
3.4.2 Diagrama de Flujo de Sistema	80
3.4.3 Diseño de Hardware.....	82
3.4.4. Diseño de Software	89
Capítulo IV: Implementación y Pruebas de Funcionamiento	103
4.1 Implementación de Hardware del Sistema Automatizado	103
4.1.1. Bloque de Sensores	104
4.1.2. Bloque de Toma de Datos y Procesamiento	106
4.1.3. Bloque de Actuadores	108
4.2. Implementación de software del Sistema Automatizado	111

4.2.1. Bloque de sensores y Toma de datos	111
4.2.2. Bloque de Procesamiento.....	114
4.3 Pruebas preliminares de Funcionamiento del Sistema.....	128
4.3.1 Pruebas de Consumo Eléctrico	129
4.3.2 Prueba de funcionamiento de sensores y placa de toma de datos.....	137
4.3.3 Prueba de funcionamiento de Actuadores.....	137
4.3.4 Prueba de comunicación MQTT.....	140
4.3.5 Prueba de funcionamiento de Base de Datos	141
4.3.6 Prueba de funcionamiento de lógica difusa en software de simulación ...	145
4.4. Pruebas generales del Funcionamiento del Sistema	148
4.4.1 Comparación con el método tradicional de crianza	150
4.4.2 Resultados observados	151
4.5. Funcionamiento del sistema en la plataforma web durante las pruebas por camadas.....	167
4.6 Simulación de condiciones ambientales extremas en camadas adicionales	171
4.6.1 Justificación de la simulación	171
4.6.2 Modelo matemático aplicado	172
4.6.3 Resultados de la simulación: Camada 5.....	176
4.6.4 Análisis de la mortalidad inducida por condiciones ambientales extremas	177
4.6.5 Resultados de la simulación: Camada 6.....	178

4.6.6 Análisis de la mortalidad inducida por condiciones ambientales – Camada 6	179
4.6.7 Cierre del apartado de simulación de camadas	186
Conclusiones y Recomendaciones	192
Conclusiones	192
Recomendaciones	193
REFERENCIAS.....	194

Índice de Figuras

Figura 1. <i>Funcionamiento del sistema automatizado en el galpón</i>	19
Figura 2. <i>Funcionamiento de Arquitectura IoT</i>	23
Figura 3. <i>Diseño básico de un galpón para aves de engorde</i>	31
Figura 4. <i>Pollos de crianza en fase inicial</i>	32
Figura 5. <i>Ejemplo de teoría de conjuntos difusos</i>	38
Figura 6. <i>Configuración básica de un sistema difuso puro</i>	39
Figura 7. <i>Configuración básica de un sistema Takagi-Sugeno-Kang</i>	40
Figura 8. <i>Configuración básica de un sistema difuso</i>	40
Figura 9. <i>Galpones de la empresa AVICAMP</i>	44
Figura 10. <i>Interior del galpón</i>	45
Figura 11. <i>Laterales de Galpones en la empresa AVICAMP</i>	46
Figura 12. <i>Termómetros en el galpón</i>	46
Figura 13. <i>Comederos y bebederos dentro del galpón</i>	47
Figura 14. <i>Reflector usado en los interiores galpón</i>	48
Figura 15. <i>Diagrama de bloques de sensores</i>	67
Figura 16. <i>Diagrama de Bloque de Procesamiento</i>	68
Figura 17. <i>Diagrama de Bloques de Actuadores</i>	69
Figura 18. <i>Diagrama de Bloques del Sistema Automatizado</i>	70
Figura 19. <i>Diagrama de Bloques del Sistema de Monitoreo</i>	79
Figura 20. <i>Diagrama de Funcionamiento del sistema de monitoreo</i>	81
Figura 21. <i>Diagrama de conexión entre sensor DHT22 y ESP32</i>	82
Figura 22. <i>Diagrama de conexión entre sensor MQ135 y ESP32</i>	83
Figura 23. <i>Diagrama de conexión entre sensor VL53L0X y ESP32</i>	84

Figura 24. <i>Diagrama de comunicación entre placa de Toma de Datos y Placa de Procesamiento.....</i>	85
Figura 25. <i>Diagrama de conexión entre placa de Toma de Datos y Actuadores</i>	86
Figura 26. <i>Diagrama de conexión del bloque de Alimentación Eléctrica.....</i>	87
Figura 27. <i>Diagrama de conexión del bloque de Alimentación Alternativo</i>	87
Figura 28. <i>Diagrama de Conexión Total del Sistema Automatizado</i>	89
Figura 29. <i>Protocolos usados en el bloque de sensores</i>	90
Figura 30. <i>Métodos de comunicación hacia Node-Red.....</i>	91
Figura 31. <i>Diseño de Flow en Node-Red.....</i>	92
Figura 32. <i>Datos presentados en el monitor de Datos del sistema automatizado.....</i>	92
Figura 33. <i>Base de datos relacional para el sistema automatizado</i>	93
Figura 34. <i>Diagrama de Bloques del Sistema Difuso.....</i>	95
Figura 35. <i>Gráfica de función de pertenencia en temperatura.....</i>	99
Figura 36. <i>Gráfica de función de pertenencia en humedad.....</i>	99
Figura 37. <i>Gráfica de función de pertenencia en Amoniaco</i>	100
Figura 38. <i>Estructura del galpón.....</i>	104
Figura 39. <i>Instalación de DHT22 y MQ135 dentro del galpón.....</i>	105
Figura 40. <i>Instalación del sensor VL53L0X en el comedero.....</i>	105
Figura 41. <i>Instalación de Bloque de Toma de Datos.....</i>	107
Figura 42. <i>Instalación de Bloque de Procesamiento</i>	107
Figura 43. <i>Instalación del ventilador en el galpón.....</i>	108
Figura 44. <i>Instalación del calentador en el galpón.....</i>	108
Figura 45. <i>Instalación del foco para iluminación del galpón.....</i>	109
Figura 46. <i>Visión general del Sistema Automatizado</i>	110
Figura 47. <i>Librerías usadas para el bloque de Toma de Datos</i>	111

Figura 48. <i>Definición de pines y objetos de sensor</i>	112
Figura 49. <i>Inicialización de sensores en setup ()</i>	112
Figura 50. <i>Lectura de sensores y monitoreo serial en loop()</i>	113
Figura 51. <i>Instalación y configuración del servidor MQTT (Mosquitto)</i>	114
Figura 52. <i>Instalación de Node-RED</i>	115
Figura 53. <i>Creación del flujo en Node-RED</i>	115
Figura 54. <i>Implementación de la lógica difusa</i>	116
Figura 55. <i>Comunicación bidireccional</i>	116
Figura 56. <i>Implementación del servidor web Apache</i>	117
Figura 57. <i>Integración de MariaDB y PHP</i>	118
Figura 58. <i>Exposición remota mediante Ngrok</i>	119
Figura 59. <i>Panel de la interfaz web</i>	119
Figura 60. <i>Gráfica de Temperatura dentro del galpón</i>	120
Figura 61. <i>Configuración dinámica de rangos</i>	121
Figura 62. <i>Panel de configuración de rangos</i>	121
Figura 63. <i>Configuración inicial del bot de Telegram</i>	122
Figura 64. <i>Instalación del nodo en Node-RED</i>	122
Figura 65. <i>Módulo Sender de Telegram en el flujograma</i>	123
Figura 66. <i>Lógica de alerta en el nodo Function</i>	123
Figura 67. <i>Creación de la base de datos</i>	124
Figura 68. <i>Creación de usuario en la base de datos</i>	125
Figura 69. <i>Creación de tabla Registros</i>	125
Figura 70. <i>Creación de tabla Actuadores</i>	125
Figura 71. <i>Integración de un nodo MySQL para Node-red</i>	126
Figura 72. <i>Registros en la base de datos</i>	126

Figura 73. <i>Reglas difusas en Node-Red</i>	127
Figura 74. <i>Consumo de corriente sensor de temperatura DHT22</i>	130
Figura 75. <i>Consumo de corriente sensor de Amoniac MQ135</i>	130
Figura 76. <i>Consumo de corriente sensor de nivel de alimento VL53L0X</i>	131
Figura 77. <i>Consumo eléctrico de la Raspberry Pi 4</i>	133
Figura 78. <i>Prueba de lectura de Sensores</i>	137
Figura 79. <i>Condiciones corresponden a un ambiente frío</i>	138
Figura 80. <i>Verificación de estado de actuadores</i>	139
Figura 81. <i>Condiciones corresponden a un ambiente cálido</i>	139
Figura 82. <i>Verificación de estado de actuadores</i>	140
Figura 83. <i>Datos recibidos desde el cliente MQTT</i>	140
Figura 84. <i>Tablas principales de la base de datos</i>	141
Figura 85. <i>Validación de las relaciones definidas</i>	142
Figura 86. <i>Verificación de la estructura de la base de datos</i>	142
Figura 87. <i>Funcionamiento lógico de la Tabla Registros en la base de datos</i>	143
Figura 88. <i>Funcionamiento lógico de la Tabla Actuadores en la base de datos</i>	144
Figura 89. <i>Funcionamiento lógico de la Tabla Usuario en la base de datos</i>	144
Figura 90. <i>Funcionamiento lógico de la Tabla Sensores en la base de datos</i>	144
Figura 91. <i>Importación de librerías</i>	145
Figura 92. <i>Funciones de clasificación difusa</i>	146
Figura 93. <i>Definición de las reglas difusas</i>	146
Figura 94. <i>Pruebas de reglas definidas</i>	147
Figura 95. <i>Pollo de 1 día de edad al iniciar la prueba general del sistema</i>	148
Figura 96. <i>Configuración de rangos para temperatura, humedad y amoníaco</i>	152
Figura 97. <i>Pollo de 15 días de edad al terminar la primera camada</i>	157

Figura 98. <i>Pollo de 15 días de edad al terminar la segunda camada</i>	160
Figura 99. <i>Pollo de 15 días al finalizar la tercera camada</i>	163
Figura 100. <i>Pollo de 15 días al finalizar la cuarta camada</i>	166
Figura 101. <i>Plataforma web con eventos de temperatura por debajo del umbral ...</i>	168
Figura 102. <i>Plataforma web mostrando la disminución de humedad</i>	168
Figura 103. <i>Captura de plataforma web mostrando aumento de amoniaco</i>	169
Figura 104. <i>Parámetros en la plataforma web</i>	169
Figura 105. <i>Notificaciones en Telegram sobre alertas del sistema</i>	170
Figura 106. <i>Librerías y variables usadas en Python</i>	170
Figura 107. <i>Medidas de tendencia central (media, mediana y moda) entre datos reales y simulados</i>	182
Figura 108. <i>Comparación de la varianza entre las tres variables ambientales medidas y simuladas</i>	182

Índice de Tablas

Tabla 1. <i>Características de tecnologías inalámbricas</i>	Error! Marcador no definido.
Tabla 2. <i>Acrónimos del estándar ISO/IEC/IEEE 29148</i>	45
Tabla 3. <i>Lista de Stakeholders</i>	46
Tabla 4. <i>Nivel de prioridad en requerimientos del sistema</i>	46
Tabla 5. <i>Requerimientos de Stakeholders</i>	47
Tabla 6. <i>Requerimientos iniciales del sistema</i>	48
Tabla 7. <i>Requerimientos de Arquitectura</i>	50
Tabla 8. <i>Comparación de sensores de temperatura</i>	53
Tabla 9. <i>Elección del sensor de temperatura</i>	54
Tabla 10. <i>Comparación de sensores de NH₃</i>	55
Tabla 11. <i>Elección del sensor de temperatura</i>	56
Tabla 12. <i>Comparativa de sensores de nivel para alimento y agua</i>	57
Tabla 13. <i>Análisis comparativo para sensores de proximidad</i>	58
Tabla 14. <i>Comparación de microcontroladores para toma de datos</i>	59
Tabla 15. <i>Análisis comparativo de los microntroladores</i>	59
Tabla 16. <i>Comparación de placas para Procesamiento</i>	61
Tabla 17. <i>Análisis de placas para Procesamiento</i>	61
Tabla 18. <i>Actuadores para el sistema automatizado</i>	62
Tabla 19. <i>Voltaje y corriente en bloque de Toma de Datos</i>	64
Tabla 20. <i>Voltaje y Corriente en Bloque de Procesamiento</i>	65
Tabla 21. <i>Voltaje y Corriente en Bloque de Actuadores</i>	66
Tabla 22. <i>Voltaje y Corriente en el Sistema Automatizado</i>	67
Tabla 23. <i>Elección de software de programación Bloque de Toma de Datos</i>	68
Tabla 24. <i>Comparación de plataformas de procesamiento</i>	69

Tabla 25. <i>Elección del software de Programación</i>	70
Tabla 26. <i>Cuadro comparativo de Hosting</i>	70
Tabla 27. <i>Elección de Hosting</i>	71
Tabla 28. <i>Cuadro comparativo de servidores web</i>	72
Tabla 29. <i>Elección de Servidor Web</i>	72
Tabla 30. <i>Comparativa para Base de Datos</i>	73
Tabla 31. <i>Elección de Base de Datos</i>	73
Tabla 32. <i>Comparativa entre plataformas de notificación</i>	74
Tabla 33. <i>Elección de Plataforma de Notificación</i>	75
Tabla 34. <i>Detalle de los elementos que conforman la base de datos</i>	90
Tabla 35. <i>Identificación de Variables del Universo Discurso</i>	93
Tabla 36. <i>Variables Lingüísticas de entrada y salida</i>	93
Tabla 37. <i>Distribución Borrosa de las Variables del Sistema</i>	94
Tabla 38. <i>Funciones de Pertenencia en el Sistema Difuso</i>	95
Tabla 39. <i>Reglas diseñadas para monitorear el sistema automatizado</i>	98
Tabla 40. <i>Plan de pruebas preliminares del sistema</i>	125
Tabla 41. <i>Tabla de consumo eléctrico del bloque de sensores y toma de datos</i>	128
Tabla 42. <i>Tabla de consumo eléctrico para el Bloque de Actuadores</i>	131
Tabla 43. <i>Tabla de consumo eléctrico mensualmente del sistema</i>	133
Tabla 44. <i>Funcionamiento de los actuadores</i>	135
Tabla 45. <i>Descripción de la prueba general de funcionamiento del Sistema</i>	146
Tabla 46. <i>Comparación funcional entre los dos enfoques de manejo del galpón</i>	147
Tabla 47. <i>Activación diaria de las 27 reglas difusas</i>	149
Tabla 48. <i>Frecuencia total de activación de cada regla difusa</i>	150
Tabla 49. <i>Eventos críticos detectados – Camada 1</i>	152

Tabla 50. <i>Activación diaria de las 27 reglas difusas</i>	155
Tabla 51. <i>Eventos críticos registrados – Camada 2</i>	156
Tabla 52. <i>Activación diaria de las 27 reglas difusas</i>	158
Tabla 53. <i>Eventos críticos registrados – Camada 3</i>	159
Tabla 54. <i>Activación diaria de las 27 reglas difusas</i>	161
Tabla 55. <i>Eventos críticos registrados – Camada 4</i>	162
Tabla 56. <i>Comparativa entre modelos matemáticos</i>	169
Tabla 57. <i>Variables simuladas</i>	170
Tabla 58. <i>Registro diario simulado – Camada 5</i>	173
Tabla 59. <i>Resumen de mortalidad según clasificación ambiental – Camada 5</i>	174
Tabla 60. <i>Registro diario simulado – Camada 6</i>	175
Tabla 61. <i>Resumen de mortalidad según clasificación ambiental – Camada 6</i>	176
Tabla 62. <i>Registro diario simulado – Camada 7</i>	177
Tabla 63. <i>Resumen de mortalidad según clasificación ambiental – Camada 7</i>	178
Tabla 64. <i>Registro diario simulado – Camada 8</i>	179
Tabla 65. <i>Resumen de mortalidad según clasificación ambiental – Camada 8</i>	180
Tabla 66. <i>Precios de componentes</i>	185

RESUMEN

El presente proyecto tuvo como objetivo implementar un sistema automatizado basado en una arquitectura IoT para el monitoreo y la gestión ambiental de un galpón de crianza de pollos en la avícola AVICAMP. La crianza tradicional de aves está sustentada en procedimientos manuales y presenta limitaciones en la regulación de parámetros críticos como temperatura, humedad y ventilación y hace que afecte el bienestar y el crecimiento de los animales.

El desarrollo se estructuró conforme a la metodología PMBOK ya que abarca las fases de Inicio, Planificación, Ejecución, Monitoreo y Cierre con esto garantizando un proceso ordenado y medible. En la planificación se establecieron los requerimientos de hardware, software e inteligencia artificial basándose en la norma ISO/IEC/IEEE 29148 y se diseñó una arquitectura IoT compuesta por sensores ambientales conectados a un nodo de control y una plataforma web para la visualización y supervisión en tiempo real y también el sistema incorporó un controlador basado en lógica difusa que permitió clasificar las condiciones del entorno y ejecutar acciones automáticas para la gestión ambiental.

Las pruebas experimentales presentaron una mayor precisión en la medición de variables ambientales también una reducción significativa de la intervención manual así como una optimización del control térmico y de ventilación, esto hizo que se mejore condiciones de confort y bienestar para las aves, así como una gestión más eficiente de los recursos energéticos.

La implementación del sistema demostró que el uso de arquitecturas IoT con control inteligente en la avicultura puede usarse como una solución replicable, sostenible y económica para la automatización de procesos avícolas en entornos rurales. Su aplicación promueve la nueva tecnológica del sector y contribuye al desarrollo de modelos de producción más eficientes y responsables.

ABSTRACT

This project aimed to implement an automated system based on an IoT architecture for environmental monitoring and management of a chicken broiler house at the AVICAMP poultry farm. Traditional poultry farming relies on manual procedures and has limitations in regulating critical parameters such as temperature, humidity, and ventilation, which can affect the welfare and growth of the animals. The development was structured according to the PMBOK methodology, encompassing the phases of Initiation, Planning, Execution, Monitoring, and Closure, thus ensuring an organized and measurable process. During the planning phase, hardware, software, and artificial intelligence requirements were established based on the ISO/IEC/IEEE 29148 standard, and an IoT architecture was designed, consisting of environmental sensors connected to a control node and a web platform for real-time visualization and supervision and the system also incorporated a controller based on fuzzy logic that allowed for the classification of environmental conditions and the execution of automatic actions for environmental management. The experimental tests showed greater accuracy in measuring environmental variables, as well as a significant reduction in manual intervention and an optimization of thermal control and ventilation. This improved comfort and well-being conditions for the birds, as well as a more efficient management of energy resources. The implementation of the system demonstrated that the use of IoT architectures with intelligent control in poultry farming can be used as a replicable, sustainable, and cost-effective solution for the automation of poultry processes in rural environments. Its application promotes new technology in the sector and contributes to the development of more efficient and responsible production models.

Capítulo I: Antecedentes

1.1. Tema

SISTEMA AUTOMATIZADO BASADO EN UNA ARQUITECTURA IOT PARA LA GESTIÓN Y MONITOREO DEL AMBIENTE DE UN GALPÓN DE CRIANZA DE POLLOS EN LA AVÍCOLA AVICAMP

1.2. Problema

Ecuador produce toda la carne de pollo y huevos de mesa que consumen sus habitantes. En el año 2023 se produjeron en el Ecuador 549 mil toneladas de carne de pollo a partir de la cría de 292 millones de pollos de engorde, lo que quiere decir que en promedio un ecuatoriano consume 30 kg de pollo al año. La crianza de aves se desarrolla principalmente en dos entornos, granjas avícolas (84,7%) y zonas rurales (15,3%), cada uno de estos espacios alberga diferentes tipos de aves. En el campo encontramos aves como gallos, gallinas y patos, mientras que en granjas avícolas se crían gallinas reproductoras y pollos de engorde. Entre las aves más criadas a nivel nacional, destacan los pollos y gallinas (71%), seguidos de las gallinas ponedoras (16%). (Sánchez, 2021).

En este contexto, se entrevistó al Ing. Roberto Guagala, encargado de la crianza de pollos, quien manifestó que, “El principal problema en el proceso de crianza es la dependencia manual de todo el sistema, ya que todos los procesos de regulación de condiciones ambientales se realizan de manera tradicional, no se tiene un control de alimentación de forma inteligente sino más bien de forma empírica, y al no tener condiciones óptimas de ventilación hace que se desarrollen enfermedades relacionadas con la respiración como es la ascitis, ocasionado por la mala gestión del estiércol dentro del galpón al no tener una ventilación adecuada”. (Guagala, R., comunicación personal, 22 de abril de 2024) Esta problemática se debe a varias causas como se observa en la figura 1.

Para optimizar la crianza tradicional de pollos en la industria avícola, se propone implementar un sistema de automatización basado en IoT en AVICAMP. Este sistema

permitiría un monitoreo y control preciso de factores ambientales críticos como temperatura, humedad, ventilación e iluminación, así como el suministro de alimento para pollos. (Bhuiyan, Wree, 2023) Al utilizar sensores y dispositivos conectados, se eliminaría la dependencia manual de la regulación de estas condiciones, lo que contribuiría a mejorar el ambiente de crianza. (Mojtaba, 2020) Esta solución no solo reduciría la intervención humana en el proceso, sino que también aumentaría la eficiencia y la productividad animal en la avícola AVICAMP. (Gerhardson & Min, 2022)

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar un sistema automatizado para un galpón de crianza de pollos basado en una arquitectura IoT, con el fin de optimizar las condiciones ambientales y el monitoreo del estado de las aves.

1.3.2. Objetivos Específicos

- Evaluar las condiciones ambientales actuales del galpón y su impacto en el crecimiento de las aves.
- Establecer requerimientos de hardware, software e inteligencia artificial para el diseño del sistema automatizado de un galpón para crianza de pollos.
- Diseñar la arquitectura del sistema IoT para la transmisión de información entre los nodos y el controlador principal.
- Comprobar el funcionamiento del sistema a través de pruebas de hardware y software y comparar con las condiciones tradicionales.

1.4. Alcance

El diseño de sistema automatizado para un galpón se basará en la metodología de investigación PMBOK y esta consta de cinco procesos principales, (Nam, 2019) la fase de Inicio se relaciona directamente con el objetivo específico 1, ya que es donde se investiga el contexto del problema y se realiza la evaluación inicial de viabilidad. También podría involucrar actividades de Planificación ya que es donde se desarrollan los planes para llevar a cabo la evaluación de las condiciones ambientales y se identifican los recursos necesarios. (Li, Zhang, Jin y Liu, 2020)

El objetivo específico 2 se adapta a la fase de Planificación ya que aquí se definirían los requisitos del proyecto, incluidos los requerimientos de hardware, software e inteligencia artificial. (Xie, Li, y Chunxu, 2023) También involucra actividades de Ejecución ya que es aquí donde se lleva a cabo la adquisición o desarrollo de los recursos necesarios. La fase de Planificación se relaciona con el diseño del sistema IoT ya que incluida su arquitectura que comprende el objetivo específico 3. Aquí podría incluir actividades de Ejecución, donde se implementa el diseño del sistema. (Hewawasam, Gunasekara, Jayarathna, Indrawansha, Rathnayake, y Panduwawala, 2023)

El último objetivo se alinea principalmente con la fase de Ejecución y Control y así también con la fase de Cierre. Lo que se sabe es donde se llevan a cabo las pruebas de hardware y software para verificar el funcionamiento del sistema, luego se compararían los resultados con las condiciones tradicionales para evaluar el impacto del nuevo sistema. (Sugiarto, 2015)

Para evaluar las condiciones ambientales del galpón basado en técnicas de crianza tradicional se usará información proporcionada por personal experto de la empresa, (Guagala, R., comunicación personal, 22 de abril de 2024) como se gestiona el control de las condiciones ambientales y bajo qué parámetros se cambian o regulan y como inciden directamente con la fase inicial de crianza de pollos. (Soliman y Cuevas, 2024)

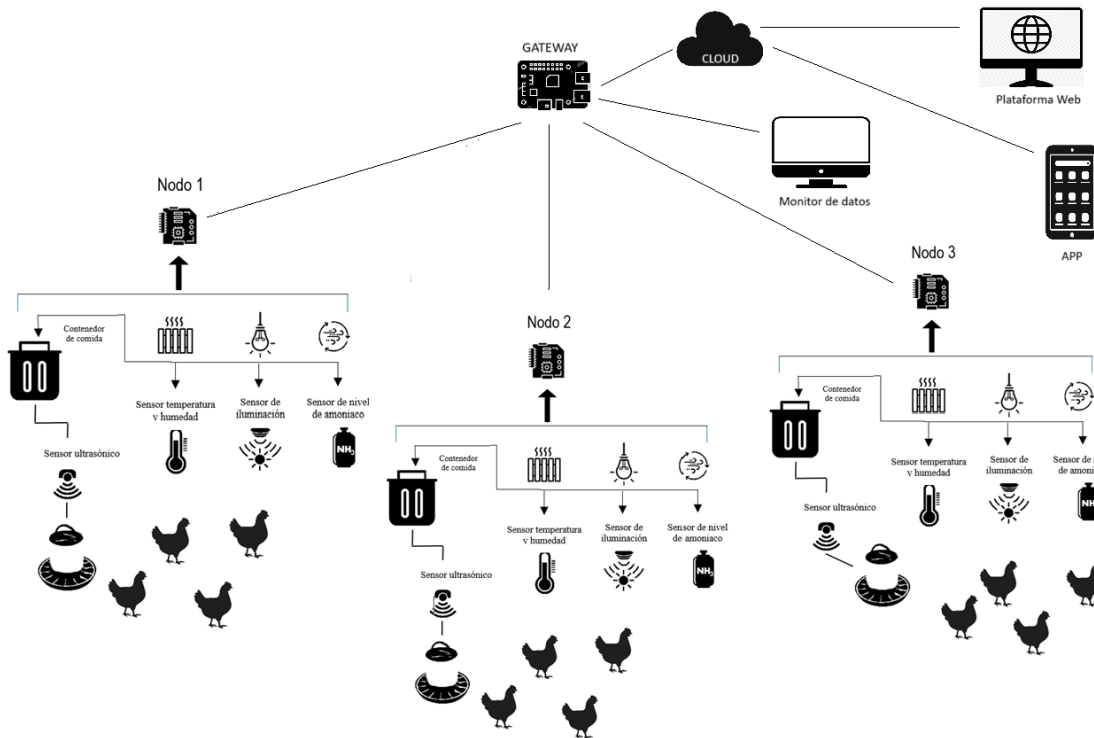
Para determinar los requerimientos de hardware y software que se usarán en el desarrollo del sistema automatizado basado en IoT para la crianza de pollos se lo hará en base a la norma ISO/IEC/IEEE 29148, (Faroqui, Utama, Ramdhani y Mulyana, 2020) para la elección del software se usará el lenguaje de programación que mejor se adapte en el diseño y en el apartado de comunicación se lo hará de manera inalámbrica y para determinar la plataforma cloud se tomará en cuenta aquella que cumpla con los parámetros establecidos en el proyecto. (Li, Zhang, Jin y Liu, 2020) Se determinará que algoritmo de inteligencia artificial se acopla mejor al funcionamiento del sistema. (Sun, Xiao, Yuan, Wang, Ma, y Liu, 2023)

El diseño de la arquitectura estará basado en todos los elementos que se plantearon en la fase de requerimientos de software y de hardware así también como los sensores tanto de temperatura, humedad, iluminación, nivel de amoníaco y sensor ultrasónico, (Hewawasam, Gunasekara, Jayarathna, Indrawansha, Rathnayake, y Panduwawala, 2023) estarán conectados a un dispositivo de control y luego el dispositivo Gateway recibe la información obtenida de cada nodo como se observa en la Figura 1, cada nodo tiene conectado varios sensores como temperatura, humedad, nivel de amoníaco y ultrasónico. (Vinay, Kallam y Rao, 2022)

Luego de recopilar toda la información de sensores se usa el algoritmo de inteligencia artificial para predecir cuáles son las variables ambientales en las que se encuentra el galpón. (Sugiarto, 2015) y mediante comunicación inalámbrica se transmiten los datos a la plataforma cloud y así mostrará en el sitio web o aplicación móvil todos los resultados que se obtuvieron en cada nodo y permitiendo que el personal a cargo pueda monitorear en todo momento la situación actual del galpón. (Wu, Chen, Chang y Wang, 2020)

Figura 1

Funcionamiento del sistema automatizado en el galpón



Fuente: Autoría

Al realizar pruebas de funcionamiento tanto de hardware como software se observará el desempeño de todos los sensores que constan en el sistema automatizado ya que deben funcionar correctamente para obtener datos y para la parte de software se comprobará el funcionamiento correcto de la base de datos y la plataforma de gestión web y aplicación móvil y así se llevará a cabo pruebas del algoritmo de inteligencia artificial mediante la comprobación con datos reales y así determinar su precisión. (Sugiarto, 2015)

1.5. Justificación

La implementación de este sistema automatizado basado en IoT para pollos permite tener una mejor gestión ambiental dentro del galpón así contribuyendo con el Objetivo de Desarrollo Sostenible 9 de las Naciones Unidas (ODS 9) titulado Industria, innovación e infraestructuras el cual busca construir infraestructuras resilientes y promover la industrialización inclusiva y sostenible y fomentar la innovación. Este ODS reconoce la

importancia de estos tres pilares para el desarrollo económico y social sostenible. (Mavuri, Chavali y Kumar,2019)

La Organización de las Naciones Unidas para la producción y consumo responsables busca reducir el impacto ambiental del consumo y la producción de bienes y servicios. (Wu, Chen, Chang y Wang,2020) La gestión adecuada del amoníaco en la producción avícola es importante para minimizar su impacto ambiental ya que el amoníaco puede contribuir a la contaminación del aire, agua y así como a la acidificación de los suelos. (Organización de las Naciones Unidas para la Alimentación y la Agricultura, 2014)

El desarrollo de sistemas automatizados para el monitoreo ambiental en la avicultura es escaso ya que por este motivo los avicultores optan por técnicas convencionales y así creando una dependencia manual excesiva con el fin de controlar las condiciones ambientales dentro de los galpones. (Guagala, R., comunicación personal, 22 de abril de 2024) Debido a esto se plantea desarrollar un sistema automatizado basado en Tecnologías IoT para el monitoreo de las condiciones ambientales en la crianza de pollos.

La integración de dispositivos IoT en el sistema de alimentación posibilitará un control más preciso sobre la cantidad y el tiempo de suministro de alimento para pollos que esto no solo contribuirá a la salud y el bienestar sino que también puede generar ahorros en los costos de alimentación y reducir el desperdicio de alimentos. Teniendo un control más preciso sobre la cantidad de alimento y se puede evitar el sobre alimentado así que lo que puede conducir a un crecimiento más uniforme y saludable de las aves. (Guagala, R., comunicación personal, 22 de abril de 2024)

Capítulo II: Fundamentación Teórica

En este capítulo se establecerán las bases del proyecto, con las teorías necesarias para hacerlo posible. Se describirán los conceptos relacionados con la avicultura y sus factores ambientales clave, como la temperatura afecta el aprovechamiento energético de los pollos. Asimismo, se explicará la arquitectura y funcionamiento fundamental del Internet de las Cosas, lo que incluirá la descripción de las tecnologías de redes inalámbricas y tipos de red de sensores, así como los protocolos de comunicación correspondientes.

A continuación, describo las placas de sistemas embebidos, los lenguajes de programación, las bases de datos y el sistema de aprendizaje automático relacionadas con este proyecto, designado algoritmo de lógica difusa y las normativas y metodologías vinculadas con el diseño.

Es importante destacar que el funcionamiento de la red WSN es el protocolo de comunicación IEEE 802.11 y que comúnmente conocido como Wifi y donde se definirán sus frecuencias de operación, canales de transmisión y los componentes de configuración que intervienen en la red.

2.1. IoT

IoT implica la conexión en red de objetos cotidianos, los cuales suelen contar con alguna forma de inteligencia integrada. (Guajardo & Laines, 2015). El Internet actúa como plataforma para dispositivos que se comunican electrónicamente y comparten datos específicos con su entorno. Así IoT representa una evolución del Internet tradicional que ofreciendo una mayor interconectividad y una mejor captación de información como también servicios inteligentes más avanzados.

El Internet de las Cosas representa un cambio radical en la calidad de vida ya que esta ofreciendo nuevas oportunidades en sectores como educación, seguridad, atención sanitaria y transporte. Y es clave para mejorar la productividad empresarial mediante redes distribuidas y

locales de dispositivos inteligentes así como servicios personalizables según las necesidades del cliente. (Rose, Eldridge & Chapin, 2015) La IoT mejora la gestión y el seguimiento de activos y productos ya que aumenta la disponibilidad de datos para informar decisiones estratégicas y optimiza el uso de recursos ya que lo que se traduce en ahorros significativos. También fomenta la creación de dispositivos interconectados inteligentes y la exploración de nuevos modelos de negocio innovadores. (Sethi & Sarangi, 2017)

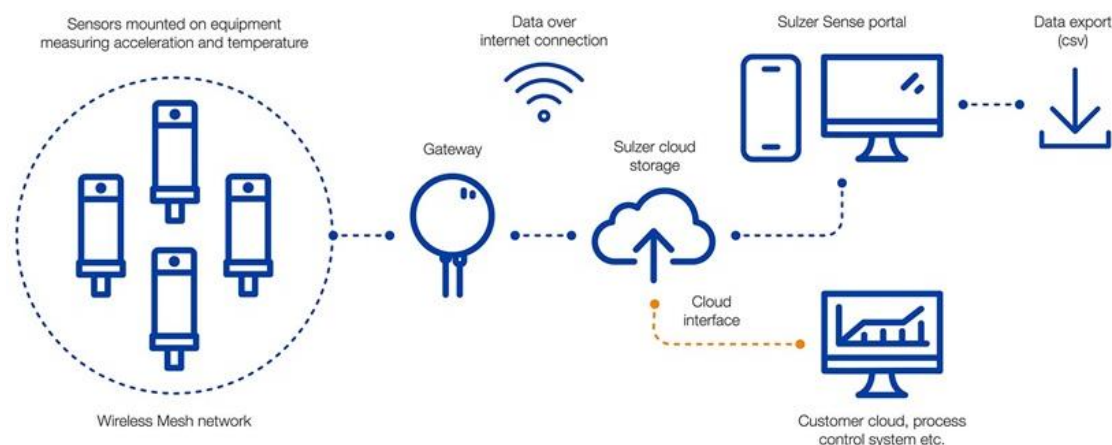
2.1.2. Arquitectura IoT

El Internet de las Cosas ha consolidado su presencia en diversos ámbitos desde lo personal hasta lo laboral mediante tecnologías que permiten la conectividad digital entre distintos dispositivos. La arquitectura IoT desempeña un papel crucial en el diseño y la integración de estos sistemas (Salazar & Silvestre, 2023).

En el proceso de la arquitectura IoT como se observa en la Figura 2 y la información se transmite a través de la red en forma de datos digitalizados como desde sensores hasta centros de datos o la nube ya que es donde se procesan y almacenan (Guajardo & Laines, 2015). Mediante actuadores se pueden enviar comandos para que los dispositivos conectados ejecuten acciones como activar o desactivar mecanismos.

Figura 2

Funcionamiento de Arquitectura IoT



Fuente: Adaptado de (Sulzer, 2022)

Las aplicaciones de IoT son vastas, especialmente en la industria 4.0, que ha adoptado la arquitectura IoT para la captura y análisis de datos, una práctica fundamental para cualquier empresa, sin importar su tamaño (Rose, Eldridge & Chapin, 2015). Indudablemente, la arquitectura IoT ha revolucionado los procesos de producción, reduciendo costos y tiempos, al tiempo que mejora la capacidad de respuesta ante la demanda. Asimismo, esta tecnología impulsa el desarrollo de la robótica colaborativa (Sharma & Gondhi, 2018).

La arquitectura se compone de cuatro capas descritas a continuación:

2.1.2.1. Capa de Sensores

Esta capa se compone de sensores, actuadores, microcontroladores, radio frecuencia, cámaras, dispositivos inalámbricos, módulo de conectividad, fuente de energía. Los dispositivos deberían consumir baja energía y tener batería de duración extendida (Guajardo & Laines, 2015). Los sensores capturan los aspectos o datos del mundo físico sea en un edificio, oficina, calle u otro espacio, los datos recolectados son valiosos para rastrear o medir un ambiente o infraestructura. Se utiliza microcontroladores los cuales tienen buenas capacidades informáticas y disponibilidad en los mercados ya que además poseen un sistema operativo ligero, conveniente y compatible con cualquier dispositivo (Tanasiev, 2021).

2.1.2.2. Capa Conectividad

En la capa de conectividad, la comunicación desempeña un papel fundamental al servir de puente entre la capa física de los dispositivos y el funcionamiento global del sistema. El primero de ellos es la conexión directa mediante TCP o UDP/IP, que permite la transmisión de datos eficiente y confiable. El segundo método implica el uso de puertas de enlace las cuales actúan como intermediarios entre la red de área local y la red de área amplia y así facilitando así la transferencia de información a través de múltiples protocolos, lo que es esencial para el funcionamiento de dispositivos IoT en diferentes entornos. (Chuquimarca & Sinche, 2022)

Las tecnologías de red que se integran en los sistemas de IoT son diversas y cumplen funciones específicas según el contexto de uso. Wifi se destaca como la técnica más popular y versátil, ideal para la comunicación en hogares inteligentes y oficinas. LPWAN (red de área amplia de bajo consumo) se ha diseñado para aplicaciones IoT que requieren comunicación a largas distancias con un bajo consumo energético ya que mientras que ZigBee se enfoca en la automatización del hogar y puede soportar hasta 65,000 nodos en su red (Chanchí, Ospina, & Monroy, 2020)

La comunicación en IoT también se apoya en diversos protocolos de mensajería que permiten el intercambio eficiente de datos. Entre ellos el Servicio de Distribución de Datos (DDS) facilita la mensajería en tiempo real entre máquinas, mientras que el Protocolo de Cola de Mensajes Avanzado (AMQP) permite el intercambio de datos entre servidores (Bharat, 2018).

2.1.2.3. Capa Nube

IoT y Cloud Computing progresan de manera autónoma (Bharat Kumar, 2018), IoT se beneficia de las características y ventajas de la nube para equilibrar los requisitos de IoT como penetración y acceso al mundo real, computación limitada, almacenamiento limitado y datos fuentes para Big Data, como la nube equilibra aquellos requisitos con representación global, recursos virtuales, computación ilimitada, almacenamiento ilimitado y gestión de Big Data. La nube con la computación ilimitada junto a IoT componen una excelente perspectiva para muchas áreas o sectores de empresas dedicadas a la agricultura y avicultura, educación, energía, turismo, transporte, entre otras.

2.1.2.4. Capa de aplicación

Se define como aquella donde se ubican todos los servicios que el sistema brindará a sus usuarios, definiendo de esta manera el área de aplicación para la cual fue creada (casas inteligentes, agricultura inteligente, transporte inteligente, etc.) Esta es la única capa a la cual

el usuario final tiene acceso y que conoce casi en su totalidad, siendo el resto de la estructura un proceso desconocido (Chuquimarca & Sinche, 2022). Además, es la responsable de la interacción entre el sistema y los usuarios finales, esto gracias a que dicha capa provee a los usuarios de los servicios solicitados. El sistema IoT necesita de soporte de las tecnologías de procesamiento y manejo de información, como pueden ser el uso de la computación en la nube, bases de datos, minería de datos, etc. (Rose, Eldridge & Chapin, 2015).

2.1.3. Tecnologías de comunicación inalámbrica para IoT

Uno de los principales habilitadores de un proyecto de Internet de las Cosas son las redes de comunicación que permiten conectar dispositivos, máquinas, sensores o "cosas" que generan datos desde cualquier lugar del mundo (Bliznakoff & Valle, 2015). Es decir, al hablar de un proyecto de IoT no es necesario imaginar miles de sensores conectados simultáneamente. Con solo dos "generadores de datos" ubicados en lugares distintos y conectados a Internet, utilizando una red de comunicaciones adecuada para transmitir datos a Internet y procesarlos después, se establece la base para un proyecto de Internet de las Cosas. (Alfonzo, 2020)

2.1.3.1. GSM/GPRS

GSM es un estándar de telefonía móvil digital que se utiliza ampliamente en todo el mundo. Introducido en la década de 1990, GSM permite la transmisión de voz y datos a través de redes móviles. Utiliza un enfoque de acceso por división de tiempo (TDMA), lo que permite que múltiples usuarios compartan el mismo canal de radio al dividir el tiempo en intervalos. Esto resulta en una comunicación más eficiente en comparación con las tecnologías analógicas anteriores.

La industria GSM, con tecnologías 2G, 3G y 4G, son muy importantes en nuestras vidas. Las siglas GPRS vienen del inglés General Packet Radio Service (Servicio General de Paquetes vía Radio). Tuvo un auge en las décadas pasadas hoy están siendo sustituidos por los sistemas 3G y 4G, todavía son utilizadas en zonas en donde la cobertura de tercera y cuarta

generación aún no está disponible. El GPRS se basa en el sistema GSM de transmisión que consiste en permitir comunicarse vía satélite, sin necesidad de cables ni conexión física a dos terminales móviles (Çelik, 2015).

2.1.3.2. Sigfox

Sigfox es la red de comunicaciones LPWAN específica para IoT más extendida a nivel mundial, con una cobertura próxima al 98% del territorio europeo y americano. La red de Sigfox está construida sobre una modulación ultra narrow band (UNB) y opera en la banda de 868MHz en Europa, en la banda de 902MHz en Estados Unidos y en Latinoamérica 902-928 MH, (Rose, Eldridge & Chapin, 2015). A esto hay que añadir el soporte disponible por Azure de Microsoft, lo que acelera en gran medida la ejecución de un proyecto de IoT (Rose, Eldridge & Chapin, 2015). El bajo coste de esta tecnología, su aceptación por los fabricantes de dispositivos, o el que sea una red bidireccional son otros factores a favor. Por el contrario, al ser una frecuencia no licenciada podría encontrarse en un futuro fuera de mercado, ya que esta frecuencia podría ser regulada por los organismos públicos y adquirida por el sector de las grandes empresas de telecomunicaciones, las cuales quieran apostar por M2M o NB IoT (Alfonzo, 2020).

2.1.3.3. LoRaWan

LoRa es otra red LPWAN con un modelo de negocio muy similar a Sigfox con la diferencia de que utiliza un espectro de comunicaciones un poco más amplio que SigFox. Si se busca una diferencia considerable entre ambas redes, LoRa es una red LPWAN mejor preparada para una comunicación bidireccional en tiempo real con el dispositivo de IoT. Así mismo, las especificaciones para los fabricantes que quieran comunicar sus equipos a través de LoRa son más abiertas o menos estrictas que con Sigfox. Por otro lado, la cobertura de LoRa es mucho menor que la de Sigfox, ya que actualmente solo se encuentra desplegada en Francia,

Bélgica, Suiza, Países Bajos y Sudáfrica, factor sin duda determinante a la hora de plantear un proyecto de IoT (Çelik, 2015).

2.1.3.4. Narrowband IoT

NB-IoT es un sistema de red de área amplia de baja potencia (LPWAN) que se desarrolló para permitir que los dispositivos se conecten entre sí a través de una red de señales de teléfonos móviles diseñada por 3GPP, que es responsable de las especificaciones de 3G y 4G / Comunicación LTE. NB-IoT usa dispositivos de baja velocidad de datos que usan la red celular, pero usan menos energía (Bliznakoff & Valle, 2015). El modo de operación de NB-IoT tiene 3 modos:

- Independiente: Separando el espectro sin usar LTE. Puede que tenga que encontrar una frecuencia para usar. Este modo es similar a un sensor inalámbrico típico.
- Banda de protección: Normalmente, la tecnología como LTE separará el canal de frecuencia entre sí para que cada canal no se agite utilizando el método llamado Banda de protección.
- En banda: uso en espectro LTE normal. Las ventajas de NB-IoT incluyen lo siguiente: Requiere poca energía para enviar datos de enlace ascendente al tamaño correcto, lo que resulta en una vida útil de la batería del dispositivo IoT de 10 años (Manatarinat, Poomrittigul, & Tantatsanawong, 2019).

2.1.3.5. BLE

El BLE (Bluetooth Low Energy) es una tecnología inalámbrica emergente de baja potencia desarrollada y utilizada por diversas aplicaciones, incluidas las definidas por Internet de las Cosas. Está claro que uno de los inconvenientes de la tecnología BLE es su rendimiento relativamente bajo en comparación con otras tecnologías 4G inalámbricas como Wifi, el rendimiento máximo informado varía de 58 Mbps a 236 Mbps para BLE 4.2. La pila de

protocolos BLE se compone de tres capas principales: el controlador, el host y la aplicación. El controlador comprende la capa física y la capa de enlace (Rose, Eldridge & Chapin, 2015).

Conceptualmente, los datos siempre se encuentran en el servidor y su cliente puede acceder a ellos o modificarlos mediante la capa ATT. El cliente puede acceder a los datos del servidor enviando solicitudes, que activan mensajes de respuesta del servidor, para una mayor eficiencia, el servidor también puede enviar a sus clientes dos tipos de mensajes no solicitados que son notificaciones e indicaciones (Chuquimarca & Sinche, 2022). Las notificaciones no necesitan confirmación, mientras que las identificaciones requieren confirmación. Los datos se estructuran en un servidor GATT basado en una jerarquía estricta, lo que permite el acceso y la recuperación de información entre el servidor y el cliente de manera eficiente. Los atributos son la entidad de datos más pequeña definida por GATT (Dian, Yousefi, & Lim, 2018).

2.1.3.6. Zigbee

ZigBee es una tecnología inalámbrica de pocos recursos y robusta, muy utilizada desde hace varios años y centrada en aplicaciones domóticas e industriales para IoT. Está basado en IEEE 802.15.4 usa banda ISM de 2.4 GHz o también 868 MHz en Europa o 908 MHz en las Américas. Utilizado en diversas aplicaciones en hogares, oficinas, edificios y ciudades inteligentes. El consorcio incluye fabricantes como Amazon, Google, NXP y Texas Instruments. Actualmente los perfiles ZigBee PRO y ZigBee Remote Control (RF4CE) cumplen con las especificaciones de tasas de envío de datos bajas, pero con un alcance de cobertura cercano a los 100 metros por ello se suele descartar las comunicaciones por ZigBee en caso de proyectos donde los dispositivos a comunicar se encuentren muy alejados del concentrador de los datos (Tanasiev, 2021).

2.1.3.7 WIFI

En arquitecturas IoT, el WiFi desempeña un rol crucial al proporcionar conectividad inalámbrica a dispositivos que necesitan comunicarse localmente o con la nube. Esta tecnología

permite la transferencia de datos rápida y bidireccional entre dispositivos IoT y servidores, utilizando protocolos como MQTT o HTTP/HTTPS (Manatarinat, Poomrittigul, & Tantatsanawong, 2019). Además, aprovecha la infraestructura existente en muchas instalaciones, facilitando la implementación sin necesidad de crear redes adicionales. Sin embargo, se requiere una atención especial a la seguridad para proteger los datos y dispositivos, mediante medidas como la autenticación robusta y la encriptación de datos, asegurando la integridad de las comunicaciones en entornos IoT (Rose, Eldridge & Chapin, 2015).

Se muestra a continuación en la Tabla 1 un resumen de todas las tecnologías y se especifica sus características técnicas.

Tabla 1

Características de tecnologías inalámbricas

Tecnología	Frecuencia	Velocidad de transmisión	Rango aproximado
GSM-GPRS	900 MHz, 1800 MHz, 1900 MHz	Hasta 170 kbps (GPRS)	2-35 km (dependiendo de la ubicación)
Sigfox	868 MHz / 902 MHz	Hasta 100 bps	10-50 km (rural), 3- 10 km (urbano)
LoRaWAN	433 MHz, 868 MHz, 915 MHz	Hasta 50 kbps	2-15 km (urbano), 15-50 km (rural)
Narrowband IoT	700 MHz - 2.1 GHz (dependiendo del país)	Hasta 250 kbps	10-100 km (con excelente cobertura)
BLE (Bluetooth Low Energy)	2.4 GHz	Hasta 2 Mbps	10-100 metros (dependiendo del entorno)
Zigbee	2.4 GHz, 868 MHz, 915 MHz	Hasta 250 kbps	10-100 metros
Wi-Fi (802.11n/ac/ax)	2.4 GHz, 5 GHz (y 6 GHz en Wi-Fi 6E)	Hasta 9.6 Gbps (Wi-Fi 6)	50-200 metros (dependiendo del entorno)

Fuente: Autoría

2.2. Crianza de pollos en galpones

La crianza de pollos en galpones es una práctica fundamental en la avicultura moderna, que permite la producción eficiente de carne y huevos en un entorno controlado (Italcol, 2018). Este sistema ofrece múltiples ventajas, como la optimización del espacio, el manejo de condiciones ambientales y la mejora en la salud y bienestar de las aves. Un galpón bien diseñado y gestionado puede maximizar la productividad y minimizar los riesgos de enfermedades (Sánchez, 2021).

2.2.1. Diseño y Estructura del Galpón

El diseño del galpón es crucial para el éxito de la crianza. Un galpón típico para pollos de engorde tiene dimensiones que varían según la cantidad de aves que se deseen criar. Un ejemplo de un galpón básico se puede observar en la Figura 3, en donde se observa su estructura y forma de cómo debe estar constituido el galpón.

Figura 3

Diseño básico de un galpón para aves de engorde



Fuente: Extraído de (Jaramillo, 2015)

La altura del techo debe ser adecuada para permitir una buena ventilación y evitar el hacinamiento. Las condiciones climáticas del lugar donde se ubica el galpón también son

determinantes (Soliman y Cuevas, 2024). La temperatura óptima para el crecimiento de los pollos debe ser controlada, especialmente durante sus primeras semanas de vida. Se recomienda mantener una temperatura inicial de entre 32 °C y 35 °C, reduciéndola gradualmente a medida que las aves crecen. La ventilación es otro aspecto importante; debe ser suficiente para eliminar el exceso de humedad y amoníaco, evitando así problemas respiratorios en las aves (Sugiarto, 2015).

2.2.2. Preparación del Área de Crianza

La preparación del área de crianza es fundamental para el desarrollo de los pollitos. Durante los primeros 7 a 15 días de vida, se deben proporcionar condiciones óptimas que fomenten el apetito y los hábitos de alimentación (Srodeck, 2020). Esto incluye la instalación de sistemas de comederos y bebederos a una altura accesible para los pollitos, asegurando que el agua esté a una temperatura adecuada (entre 18 °C y 21 °C) y que la iluminación sea uniforme. El uso de aserrín en el suelo del área de crianza ayuda a mantener la limpieza y a facilitar el acceso a los alimentos como se observa en la Figura 4.

Figura 4

Pollos de crianza en fase inicial



Fuente: Adaptado de (Sugiarto, 2015)

Es importante monitorear el comportamiento de las aves para detectar cualquier signo de estrés o malestar, lo que puede indicar problemas en el ambiente o en la alimentación (Li, Zhang, Jin y Liu, 2020).

2.2.3. Alimentación y Nutrición

La alimentación es un factor crítico en la crianza de pollos. Desde el primer día, los pollitos deben recibir un alimento que contenga altos niveles de proteínas y energía. A medida que crecen, se les proporciona alimento se les proporciona diferente tipo de alimentos. Es esencial que la dieta esté balanceada y contenga todos los nutrientes necesarios para el crecimiento óptimo de las aves, incluyendo proteínas, carbohidratos, vitaminas y minerales. La cantidad de alimento suministrada debe ser monitoreada cuidadosamente para evitar el desperdicio y asegurar que todas las aves tengan acceso a la comida. El uso de comederos automáticos puede facilitar este proceso y mejorar la eficiencia en la distribución del alimento (Srodeck, 2020). Dentro del proceso de alimentación se debe tomar en cuenta ciertos implementos como:

- Comederos Manuales: En la crianza tradicional, los comederos suelen ser simples y manuales, como recipientes de plástico o metal que se llenan a mano. Estos comederos deben ser accesibles para las aves y fáciles de limpiar. La cantidad de alimento debe ser controlada para evitar el desperdicio y asegurar que todos los pollos tengan acceso a la comida.
- Alimento Balanceado: La dieta de los pollos debe ser equilibrada, incluyendo granos, legumbres y, en algunos casos, suplementos naturales. Es común utilizar maíz, sorgo y soya, y se pueden añadir hierbas o productos naturales que promuevan la salud intestinal.
- Bebederos Manuales: Al igual que los comederos, los bebederos en la crianza tradicional suelen ser recipientes que se llenan manualmente. Pueden ser de tipo plato

o de campana, y deben ser colocados a una altura adecuada para que los pollos puedan acceder fácilmente al agua.

2.2.4. Manejo de la Salud en aves

El manejo adecuado de la salud de los pollos es esencial para prevenir enfermedades. Esto incluye la implementación de un programa de vacunación y el control de parásitos. Las prácticas de bioseguridad, como la limpieza regular de los galpones y el monitoreo de la salud de las aves, son fundamentales para mantener un ambiente saludable. Es importante observar el comportamiento de los pollos y realizar chequeos regulares para identificar signos de enfermedad. Un manejo proactivo puede ayudar a prevenir brotes de enfermedades que podrían afectar la producción y la rentabilidad del negocio avícola (Organización de las Naciones Unidas para la Alimentación y la Agricultura, 2014)

2.3. Automatización en el proceso de crianza de pollos

La crianza automatizada de pollos ha transformado la industria avícola, permitiendo un manejo más eficiente y productivo de las aves. Este enfoque utiliza tecnología avanzada para optimizar todos los aspectos del proceso de crianza, desde la alimentación y el monitoreo del ambiente hasta la recolección de datos y el manejo de la salud de las aves. La automatización no solo mejora la eficiencia operativa, sino que también contribuye a la sostenibilidad y a la producción de carne y huevos de alta calidad (Aviagen, 2018).

2.3.1. Sistemas de Alimentación Automatizados

La alimentación es uno de los aspectos más críticos en la crianza de pollos, y los sistemas automatizados han revolucionado esta área:

- Comederos Automáticos: Estos dispositivos permiten una distribución uniforme del alimento en el galpón (Gerhardson & Min, 2022). Los comederos automáticos pueden ser controlados para liberar cantidades específicas de alimento en momentos

determinados, lo que asegura que las aves tengan acceso constante a la comida. Esto reduce el desperdicio y mejora la eficiencia alimentaria.

- **Sensores de Consumo:** Algunos sistemas avanzados cuentan con sensores que monitorean el consumo de alimento en tiempo real, permitiendo ajustes inmediatos en la dieta según las necesidades de las aves. Esto es especialmente útil para optimizar el crecimiento y la conversión alimenticia (Sánchez, 2021).

2.3.2. Sistemas de Bebida Automatizados

La hidratación adecuada es esencial para el bienestar de los pollos y más aún si ya no existe una dependencia manual o de un operador para la gestión del agua.:

- **Bebederos Automáticos:** Los bebederos automáticos, permiten que las aves accedan al agua de manera eficiente. Estos sistemas son diseñados para minimizar el desperdicio y mantener el agua limpia, lo que es crucial para la salud de las aves.
- **Monitoreo de Calidad del Agua:** Sistemas de monitoreo pueden ser implementados para garantizar que el agua proporcionada a los pollos esté libre de contaminantes y a la temperatura adecuada. Esto es vital para prevenir enfermedades y asegurar un crecimiento óptimo (Bhuiyan, Wree, 2023).

2.3.3. Control del Ambiente

El control del ambiente en los galpones es fundamental para el bienestar de las aves, ya que mejora de manera significativa su calidad de vida:

- **Sistemas de Ventilación Automatizados:** La ventilación es crucial para mantener una temperatura adecuada y reducir la humedad. Los sistemas de ventilación automatizados ajustan el flujo de aire según las condiciones internas y externas, optimizando el clima dentro del galpón (Li, Zhang, Jin y Liu, 2020).
- **Control de Temperatura y Humedad:** Los sistemas de climatización modernos permiten el control preciso de la temperatura y la humedad. Esto es especialmente importante en

las primeras semanas de vida de los pollitos, donde se requiere un ambiente cálido y controlado.

2.3.4. Monitoreo y Gestión de Datos

La automatización permite una recolección de datos más eficiente y se logra una mejor gestión dentro del galpón:

- **Sensores Ambientales:** Sensores de temperatura, humedad y amoníaco pueden ser instalados para monitorear continuamente las condiciones dentro del galpón. Estos datos pueden ser analizados para realizar ajustes en tiempo real, mejorando el bienestar de las aves.
- **Software de Gestión:** Plataformas de software permiten a los productores gestionar todos los aspectos de la crianza desde un solo lugar. Esto incluye el seguimiento del crecimiento de las aves, el consumo de alimento y agua, y la salud general del lote.
- Los sistemas de gestión también pueden generar informes que ayudan a los productores a tomar decisiones informadas (Li, Zhang, Jin y Liu, 2020).

2.3.5. Manejo de Salud y Bioseguridad

La salud de las aves es una prioridad en la crianza automatizada, con la automatización se logra obtener mejores resultados en cuanto a salud:

- **Programas de Vacunación Automatizados:** La automatización puede incluir sistemas que programan y administran vacunas a las aves, asegurando que cada ave reciba el tratamiento adecuado en el momento correcto.
- **Monitoreo de Salud:** Sensores y cámaras pueden ser utilizados para monitorear el comportamiento de las aves, permitiendo la identificación temprana de problemas de salud. Esto reduce el riesgo de brotes de enfermedades y mejora la tasa de supervivencia (Vinay, Kallam, Kallam & Rao, 2022).

2.3.6. Beneficios de la Crianza Automatizada

La implementación de tecnologías automatizadas en la crianza de pollos ofrece múltiples beneficios:

- **Eficiencia Operativa:** La automatización reduce la necesidad de mano de obra, disminuyendo costos operativos y aumentando la eficiencia. Esto es especialmente importante en un sector donde la competencia es alta.
- **Mejora en la Calidad del Producto:** Al controlar de manera precisa las condiciones de crianza, se puede mejorar la calidad de la carne y los huevos producidos, lo que se traduce en un mejor precio en el mercado.
- **Sostenibilidad:** La automatización permite un uso más eficiente de los recursos, como el agua y la alimentación, contribuyendo a prácticas más sostenibles en la producción avícola (Vinay, Kallam, Kallam & Rao, 2022).

2.4. Lógica difusa

La lógica difusa es una extensión de la lógica clásica que permite manejar la incertidumbre y la imprecisión, características comunes en muchos sistemas del mundo real. A diferencia de la lógica binaria, que se basa en valores de verdad estrictos (verdadero o falso), la lógica difusa permite que las variables tengan valores en un rango continuo entre 0 y 1. Esto la hace especialmente útil en situaciones donde las decisiones no son absolutas y se requiere un enfoque más flexible y matizado (Kaur & Kaur, 2020).

La lógica difusa se utiliza en una variedad de aplicaciones, desde el control de procesos industriales hasta sistemas de inteligencia artificial. En el contexto de la crianza automatizada de pollos, la lógica difusa puede desempeñar un papel crucial en la optimización de diversas variables, como el control ambiental, la alimentación y la salud de las aves (Yang & Zhang, 2018).

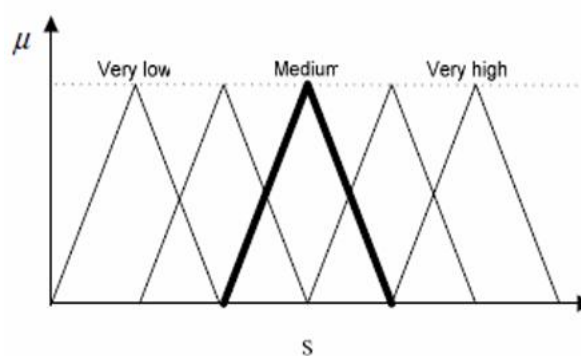
2.4.1. Teoría de conjuntos difusos

La lógica difusa se deriva de los conjuntos difusos, los cuales forman parte de la Teoría de Conjuntos, que se utiliza para representar colecciones de objetos y permite determinar si un objeto pertenece a uno o varios conjuntos o dominios. En los conjuntos tradicionales, solo es posible decir si un objeto pertenece o no a un conjunto. En cambio, en los conjuntos difusos se puede expresar en qué medida un objeto pertenece a un conjunto, lo que se conoce como el grado de pertenencia (Gustavo & Juárez, 2017).

Los Conjuntos difusos deben verse realmente como funciones que se observa en la Figura 5, que transforman elementos de un conjunto nítido en un número que, tomaremos sobre el intervalo cerrado $[0,1]$, indicando de este modo su actual grado de pertenencia al conjunto difuso (Vera, 2021). Un grado de pertenencia 0 significa que el elemento no pertenece en absoluto al conjunto, mientras que un grado de pertenencia 1 coincide con la noción usual de pertenencia del elemento al conjunto que nos da la Teoría de Conjuntos. (Gustavo & Juárez, 2017)

Figura 5

Ejemplo de teoría de conjuntos difusos



Fuente: Adaptado de (Vera, 2021)

2.4.2. Sistemas Difusos

En nuestra vida cotidiana, a veces usamos palabras para describir variables. Las variables que se expresan mediante palabras o frases en un lenguaje natural o artificial se

denominan variables lingüísticas (Merma Gallardo, 2019). Para entender mejor este concepto, consideremos el ejemplo de la palabra "edad" en un lenguaje natural; debido a la experiencia de muchas personas, sabemos que no puede definirse con precisión.

Una regla difusa SI- ENTONCES es una declaración expresada por medio de:

SI < Proposición Difusa > ENTONCES

< Proposición Difusa >

donde una proposición difusa es:

- Una declaración simple, x es A donde x es la variable lingüística y A es un valor lingüístico de x , es decir, A es un conjunto difuso de nido en un dominio físico de x (Mejía, 2019).

- Una composición de declaraciones simples usando los conectores y, o y no, los cuales son representados mediante la intersección difusa, unión y complemento difusos, respectivamente.

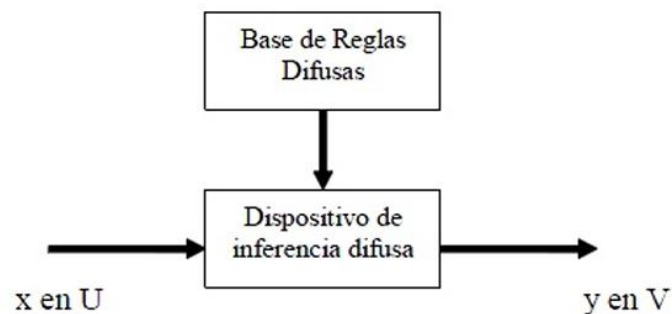
Existen tres tipos de sistemas difusos empleados comúnmente:

- Sistemas difusos puros o de Mamdani,

La configuración básica de un sistema difuso puro se observa en la Figura 5:

Figura 6

Configuración básica de un sistema difuso puro

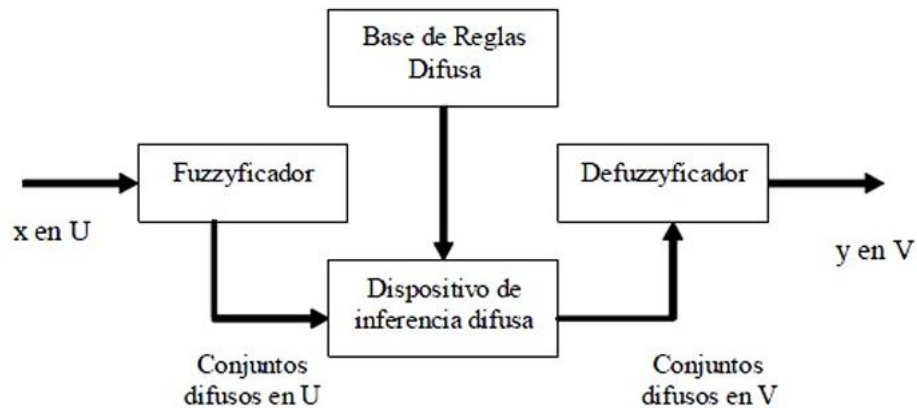


Fuente: Obtenido de (Merma Gallardo, 2019)

Como se observa en la figura 6 la configuración básica de un sistema Takagi-Sugeno-Kang es la siguiente:

Figura 7

Configuración básica de un sistema Takagi-Sugeno-Kang

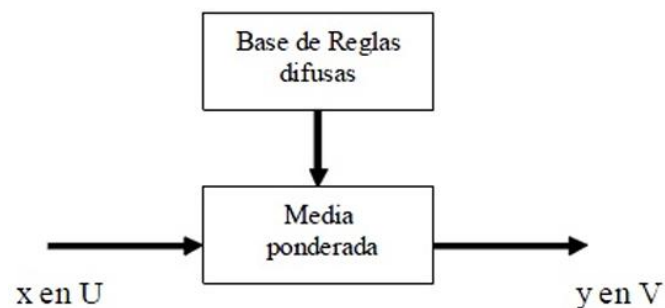


Fuente: Obtenido de (Merma Gallardo, 2019)

La configuración básica de un sistema difuso con fuzzficador y defuzzficador es la siguiente como se observa en la Figura 7

Figura 8

Configuración básica de un sistema difuso con fuzzficador y defuzzficador



Fuente: Obtenido de (Merma Gallardo, 2019)

2.4.3. Lógica difusa en Control Ambiental

Uno de los desafíos más importantes en la crianza de pollos es mantener un ambiente adecuado que promueva el crecimiento y el bienestar de las aves. La lógica difusa puede ser utilizada para:

- ***Control de Temperatura y Humedad:*** Los sistemas de climatización automatizados pueden utilizar lógica difusa para ajustar la temperatura y la humedad en función de las condiciones internas y externas (Yang & Zhang, 2018).
- ***Ventilación Inteligente:*** La lógica difusa puede ayudar a regular la ventilación en función de múltiples factores, como la cantidad de aves, la temperatura interior y los niveles de amoníaco. Esto asegura que el aire fresco se introduzca de manera eficiente, manteniendo un ambiente saludable para las aves.

2.4.4. Alimentación Automatizada con lógica difusa

La alimentación es un aspecto crítico en la crianza de pollos, y la lógica difusa puede optimizar este proceso:

- ***Sistemas de Alimentación Adaptativos:*** Utilizando lógica difusa, los sistemas de alimentación pueden ajustar la cantidad y el tipo de alimento proporcionado a las aves en función de su edad, peso y comportamiento de consumo. Por ejemplo, si un grupo de pollos muestra signos de no comer lo suficiente, el sistema puede aumentar automáticamente la cantidad de alimento disponible (Li, Zhang, Jin y Liu, 2020).
- ***Monitoreo de la Salud:*** La lógica difusa puede integrarse con sensores que monitorean el comportamiento de las aves. Si se detectan patrones de comportamiento inusuales, como una disminución en el consumo de alimento o cambios en la actividad, el sistema puede activar alertas para que los criadores tomen medidas preventivas (Aviagen, 2018).

2.4.5. Monitoreo de Salud y Bienestar

La salud de las aves es fundamental para la producción eficiente. La lógica difusa puede ser utilizada para:

- **Detección de Enfermedades:** Mediante el análisis de datos recopilados por sensores y cámaras, la lógica difusa puede ayudar a identificar signos tempranos de enfermedades. Por ejemplo, si un porcentaje significativo de aves muestra comportamientos anormales, el sistema puede activar protocolos de salud, como la revisión médica o la administración de vacunas (Vinay, Kallam, Kallam & Rao, 2022).
- **Evaluación del Bienestar Animal:** La lógica difusa puede evaluar el bienestar de las aves mediante indicadores como el comportamiento, la actividad y la interacción social. Esto permite a los criadores realizar ajustes en el manejo y el entorno para mejorar la calidad de vida de las aves (Sánchez, 2021).

Capítulo III: Metodología

El presente capítulo aborda la situación actual del galpón de crianza de pollos en la Avícola AVICAMP y presenta el proceso de diseño del sistema automatizado basado en una arquitectura IoT para la gestión y monitoreo del ambiente dentro del galpón. Para el diseño de este sistema se debe considerar los requerimientos específicos de monitoreo de temperatura, humedad, ventilación y calidad del aire, fundamentales para asegurar un ambiente óptimo para el crecimiento saludable de las aves.

El proceso de desarrollo sigue una serie de pasos metodológicos claramente definidos, garantizando el correcto funcionamiento tanto del sistema de monitoreo como del algoritmo encargado de la toma de decisiones automáticas, las cuales aseguran un ambiente estable en el galpón. Para llevar a cabo este proceso, se ha sugerido usar la metodología PMBOK, que permite estructurar el desarrollo en fases consecutivas.

3.1. Análisis del Proyecto

Se analiza el proyecto desde la perspectiva de la exploración de información científica y técnica que interviene en la gestión y monitoreo del ambiente en los galpones de crianza de pollos, así como también se establece una relación con los trabajadores encargados del cuidado y manejo de las aves en las instalaciones de la Avícola AVICAMP, con el objetivo de recopilar información valiosa para el desarrollo de la investigación.

Este análisis se basa en la identificación de las condiciones óptimas necesarias para garantizar un entorno adecuado para el crecimiento de los pollos, como la temperatura, humedad, calidad del aire y ventilación. La revisión de información científica en el campo de la avicultura revela que un sistema automatizado basado en IoT puede mejorar significativamente la eficiencia de las operaciones y reducir la incidencia humana en todo el proceso de crianza, así como también problemas de salud en las aves, al proporcionar un control preciso y continuo de las variables ambientales.

3.1.1. Situación actual

La empresa AVICAMP se encuentra en la parroquia de Cahuasqui, en el Cantón Urcuqui, aunque es importante mencionar que las pruebas con un prototipo de menor escala se las realiza en instalaciones que se ubican en la ciudad de Ibarra.

La empresa desarrolla sus actividades con galpones como se observa en la Figura 8, en base a la crianza de pollos de diferentes tipos, sin embargo, el pollo Broiler es el tipo preferido para la producción, debido a su rápido crecimiento y corto tiempo de permanencia en galpón en comparación con otras variedades. Como es un pollo de engorde, el Broiler requiere cuidados específicos para garantizar su óptimo crecimiento, para ello se debe cuidar que todas las condiciones del galpón sean correctas. Este método de crianza es preferido por la empresa, ya que se limita a la fase inicial de crecimiento del pollo, lo cual permite optimizar recursos y reducir costos en el proceso. (Guagala, R., comunicación personal, 15 de octubre de 2024)

Con base en la encuesta y su tabulación (ver ANEXO A), se determinó que el número aproximado de pollos en la producción es de 1000.

Figura 9

Galpones de la empresa AVICAMP



Fuente: Autoría

En esta empresa trabajan tres personas, en cada turno una persona se encarga de revisar cada uno de los galpones, se controla y se administra la comida el agua, así como también las condiciones ambientales dentro del galpón, ya que esto es muy importante para el bienestar de los pollos.

Con base en la información proporcionada por el personal especializado sobre el proceso y el cuidado de las aves en los galpones, se identificó que la principal problemática es la alta dependencia de la intervención humana en todo el proceso. Los primeros días de vida de los pollos son críticos en cuanto a control de la temperatura en el galpón como se observa en la Figura 9, lo que requiere personal encargado en todo momento. Para abordar esta necesidad, se propone desarrollar un sistema automatizado que monitoree el estado de los galpones. Para definir los requerimientos del sistema y su funcionamiento, se realizará una encuesta dirigida a cada uno de los trabajadores de AVICAMP, (ver ANEXO B).

Figura 10

Interior del galpón



Fuente: Autoría

La empresa AVICAMP, debido a su ubicación geográfica, experimenta altas temperaturas dentro de sus galpones, lo cual hace necesario mantener niveles óptimos de ventilación. Sin embargo, durante la noche la temperatura tiende a reducirse considerablemente, por lo que se ajustan estos niveles para garantizar un ambiente adecuado dentro del galpón. Las instalaciones están situadas en un terreno con plantaciones frutales, lo

que implica la presencia de elementos externos, como animales silvestres, que pueden dañar la infraestructura y poner en riesgo a las aves. En la Figura 10 se muestran los laterales de los galpones.

Fuente: Autoría

Figura 11

Laterales de Galpones en la empresa AVICAMP



Dentro del galpón se encuentran termómetros que indican el nivel de temperatura interior, un parámetro fundamental en el proceso de crecimiento de los pollos. Este factor requiere un control oportuno, por lo que los trabajadores deben estar atentos a cambios bruscos de calor o frío. Durante el primer día del proceso, la temperatura debe mantenerse entre 30 y 32°C, disminuyendo aproximadamente dos grados cada día conforme avanza el crecimiento. Los termómetros utilizados se muestran en la Figura 12.

Figura 12

Termómetros en el galpón



Fuente: Autoría

La alimentación es un elemento crucial en el proceso de crecimiento, al igual que el agua, que debe ser suministrada por el personal, como se observa en la Figura 13. Sin embargo, todo este proceso se realiza de manera manual y empírica. Cabe destacar que la dieta varía en cada fase del crecimiento, adaptándose a las necesidades de los pollos en cada etapa.

Figura 13

Comederos y bebederos dentro del galpón



Fuente: Autoría

Otros elementos que deben mencionarse, ya que influyen directamente en el bienestar ambiental del galpón, son la iluminación y la ventilación. Si la ventilación no se maneja adecuadamente, puede generar problemas respiratorios en los pollos debido a la acumulación de amoníaco, un componente que emanan sus heces. Por esta razón, el control de la ventilación debe realizarse adecuadamente.

La iluminación en el galpón se proporciona mediante focos y reflectores. En cuanto a la ventilación, esta se realiza manualmente levantando los costales que cubren los laterales del galpón. En la Figura 14 se pueden observar estos componentes mencionados anteriormente.

Figura 14

Reflector usado en los interiores galpón



Fuente: Autoría

3.2. Requerimientos del Sistema

Para el desarrollo del sistema automatizado, es necesario definir los requerimientos del sistema siguiendo el estándar ISO/IEC/IEEE 29148, que proporciona directrices para proyectos de ingeniería desarrollados por profesionales en áreas de hardware y software. En la Tabla 2 se presentan los requerimientos de los Stakeholders, los requerimientos iniciales del sistema y los requerimientos de arquitectura, basados en las recomendaciones de este estándar.

Tabla 2

Acrónimos del estándar ISO/IEC/IEEE 29148

Acrónimo	Descripción
StSR	Requerimientos de los Stakeholders
SySR	Requerimientos iniciales del sistema

Fuente: Autoría

3.2.1. Listado de Stakeholders

Los Stakeholders son las personas que tienen una relación directa o indirecta con el desarrollo del proyecto, cada una con funciones específicas en el proceso. En total, participan seis personas, quienes desempeñan diversas funciones en el desarrollo del sistema automatizado. La información detallada de cada uno de los Stakeholders se presenta en la Tabla 3.

Tabla 3

Lista de Stakeholders

STAKEHOLDERS		
#	Involucrados	Función
1	Ing. Roberto Guagala	Propietario de AVICAMP
2	Sr. Rubén Guagala	Trabajador de AVICAMP
3	Sra. Marta Almeida	Trabajadora de AVICAMP
4	MSc. Fabián Cuzme	Director del Trabajo de Titulación
5	MSc. Carlos Vásquez	Asesor del Trabajo de Titulación
6	Sr. Santiago Chuquilla	Autor del Trabajo de Titulación

Fuente: Autoría

Los requerimientos de Stakeholders (StSR) se enfocan principalmente en los requisitos operacionales del sistema y en los requerimientos del usuario, establecidos en colaboración con el Ing. Fabián Cuzme y el especialista en avicultura Ing. Roberto Guagala, como parte del desarrollo de la tesis.

De acuerdo con los requerimientos, se establecen las necesidades y prioridades, cada una con un nivel de importancia diferente. La Tabla 4 presenta la prioridad asignada a cada uno de los requerimientos del sistema.

Tabla 4

Nivel de prioridad en requerimientos del sistema

Prioridad	Descripción
Alta	Se considera de alta importancia y debe ser incluido en el desarrollo del sistema; de lo contrario, su ausencia afectaría el correcto funcionamiento de este.
Mediana	Se considera de importancia media; no incluirlos podría afectar el funcionamiento del sistema.
Baja	Se considera un nivel de mínima importancia, es decir que se los puede excluir, y no representa un impacto significativo en el funcionamiento del sistema.

Fuente: Adaptado de (Díaz, 2022)

3.2.2. Requerimientos Operacionales

Los requisitos operacionales y los definidos por el usuario que se muestran en la Tabla 5, estos forman parte de las necesidades de los Stakeholders para establecer el funcionamiento adecuado del sistema y priorizar cada uno de estos requisitos.

Tabla 5

Requerimientos de Stakeholders

Requerimientos de Stakeholders					
#	Requerimientos	Prioridad			Relación
		Alta	Media	Baja	
Requerimientos Operacionales					
StSR1	Conexión eléctrica estable.	X			

StSR2	Operación continua las 24 horas del día, los 7 días de la semana.	X
StSR3	Los nodos deben tener acceso a Internet de manera continua.	X
StSR4	La plataforma y monitor de datos debe estar operativa el mayor tiempo posible.	X
Requerimientos de Usuarios		
StSR5	El sistema debe mostrar los datos de cada uno de los nodos en una plataforma y el monitor de datos.	X
StSR6	El sistema debe notificar al usuario inmediatamente ante cualquier novedad detectada en el galpón.	X
StSR7	El sistema debe tomar decisiones en cuanto a la regulación de las condiciones ambientales dentro del galpón.	X

Fuente: Autoría

Todos los requerimientos son propuestos de acuerdo con las necesidades de los usuarios del sistema, ya que de acuerdo con la encuesta realizada como se observa en el Anexo A, se puede determinar cuáles son las necesidades con una prioridad mayor y así tener un funcionamiento de acuerdo con los requerimientos iniciales.

3.2.3. Requerimientos del sistema SySR

La Tabla 6 presenta los requisitos del sistema (SySR), detallando las especificaciones técnicas necesarias para el correcto funcionamiento del sistema. Estos requisitos han sido definidos en función de los objetivos propuestos y las limitaciones identificadas por los stakeholders, quienes establecieron los parámetros que el sistema debe cumplir para responder adecuadamente a sus necesidades y expectativas.

Tabla 6

Requerimientos iniciales del sistema

Requerimientos del sistema					
#	Requerimientos	Prioridad			Relación
		Alta	Media	Baja	
REQUERIMIENTOS DE USO					
SySR1	El bloque de sensores realizará lecturas de manera continua, con intervalos de diez minutos entre cada medición.	X			
SySR2	El bloque de toma de datos transmitirá información de manera continua.	X			
SySR3	El bloque de procesamiento recibirá los datos de manera continua.	X			
SySR4	Implementar un algoritmo de IA para monitorear condiciones ambientales.	X			
SySR5	El sistema debe alertar al usuario por medio de la plataforma ante una novedad en el galpón.	X			StSR6
REQUERIMIENTOS DE RENDIMIENTO					
SySR6	La calibración de sensores se basa en las especificaciones del fabricante.	X			
SySR7	El sistema debe contar con una base de datos.	X			
SySR8	El monitor de datos y plataforma deben funcionar la mayoría de tiempo para mostrar datos del sistema.	X			
SySR9	El sistema debe tener acceso a internet las 24 horas del día.	X			StSR2
REQUERIMIENTOS FÍSICOS					
SySR10	Los sensores deben tener una carcasa protectora.				X
SySR11	Los nodos remotos deben estar instalados en sitios estratégicos para la toma de datos.	X			
SySR12	El sistema debe tener un bajo consumo energético.				X
REQUERIMIENTOS DE SEGURIDAD					

SySR13	El sistema debe tener seguridad física para prevenir manipulación de terceras personas.		
SySR14	El sistema de nodos tanto principal como nodos secundarios deben tener protección física.	X	SySR10

Fuente: Autoría

3.2.4. Requerimientos de Arquitectura

Los requerimientos de arquitectura (SRSH) establecen las especificaciones de hardware y software necesarias para el diseño y funcionamiento del sistema de monitoreo ambiental en el galpón de crianza de pollos. Estos elementos se seleccionan en función de su compatibilidad, eficiencia energética y capacidad de comunicación en una red IoT. A continuación, en la Tabla 7 se detallan los principales requerimientos de arquitectura.

Tabla 7

Requerimientos de Arquitectura

Requerimientos de arquitectura					
#	Requerimientos	Prioridad			Relación
		Alta	Media	Baja	
REQUERIMIENTOS DE DISEÑO					
SRSH1	El nodo de procesamiento debe estar centralizado en el galpón, permitiendo una cobertura adecuada para todos los sensores ubicados estratégicamente en las diferentes áreas del galpón.			X	
SRSH2	Los sensores deben estar distribuidos de forma equitativa en el galpón para capturar condiciones representativas del ambiente en todas las áreas.		X		
SRSH3	Todos los sensores y equipos deben estar protegidos contra el polvo, la humedad y el amoníaco.		X		SySR10

SRSH4	Se debe implementar una plataforma de visualización que permita el acceso a los datos en tiempo real.	X	StSR4
SRSH5	El sistema debe incluir un módulo de alerta que notifique al personal en caso de que algún parámetro ambiental esté fuera del rango establecido.	X	SySR5
SRSH6	Los dispositivos de monitoreo deben operar con bajo consumo de energía, para reducir la dependencia de la red eléctrica.	X	SySR12

REQUERIMIENTOS LÓGICOS

SRSH7	El sistema debe ser compatible con los sensores y ser capaz de integrarse con microcontroladores.	X	
SRSH8	Se implementa reglas difusas para el monitoreo del sistema.	X	SySR4
SRSH9	El sistema debe contar con un mecanismo de recuperación para reanudar la operación tan pronto como se restablezca la conexión	X	
SRSH10	El sistema debe ser capaz de activar dispositivos de respuesta, como ventiladores o calefactores, para restablecer las condiciones óptimas de manera automática.	X	

REQUERIMIENTOS DE HARDWARE

SRSH11	El sistema debe contar con un microcontrolador con capacidad para procesar y transmitir los datos obtenidos de los sensores.	X	
SRSH12	El microcontrolador de toma de datos debe disponer de suficientes pines y puertos para conectar múltiples sensores.	X	
SRSH13	El módulo de toma de datos debe ser compatible con la placa de procesamiento.	X	

REQUERIMIENTOS DE SOFTWARE			
SRSH14	El lenguaje de programación debe ser compatible con los sensores y la transmisión de datos.	X	
SRSH15	La base de datos debe garantizar un acceso rápido y eficiente.	X	SySR7
SRSH16	La interfaz de usuario debe desarrollarse en un entorno web, permitiendo el acceso a través de navegadores en computadoras y dispositivos móviles.	X	
REQUERIMIENTOS DE VISUALIZACIÓN Y NOTIFICACIÓN			
SRSH17	La plataforma debe incluir gráficos estadísticos que muestren los parámetros obtenidos.	X	
SRSH18	Las alertas por situaciones críticas se envían mediante la plataforma.	X	SySR8
SRSH19	El envío de notificación en estado crítico debe de realizarse por una aplicación de mensajería.	X	
REQUERIMIENTOS ELÉCTRICOS			
SRSH20	Alimentación entre 3.3-12V para sensores y microcontroladores	X	
SESH21	El sistema deberá operar utilizando una fuente de energía sustentable, garantizando un funcionamiento continuo y autónomo, incluso en condiciones de baja disponibilidad de recursos energéticos convencionales.	X	

Fuente: Autoría

3.3. Elección de Hardware y Software

La selección de hardware y software para el sistema de monitoreo en el galpón de crianza de pollos se ha realizado con el objetivo de garantizar la recolección y procesamiento

continuo de datos, permitiendo un control preciso del ambiente en beneficio de la salud y el desarrollo óptimo de las aves. Los componentes seleccionados aseguran un funcionamiento estable, una buena resistencia a las condiciones del entorno y una integración eficiente en el sistema. Todos los elementos que se selecciona se basan en los requerimientos del sistema.

3.3.1. Hardware

La selección del hardware para el sistema de monitoreo del galpón de crianza de pollos se realizó en función de la precisión, durabilidad y compatibilidad de los componentes con el entorno avícola. Los dispositivos deben ser capaces de operar de manera continua y eficiente, soportando las condiciones ambientales del galpón y garantizando un monitoreo constante.

3.3.1.1. Bloque de sensores

Para el sistema de monitoreo en el galpón de crianza de pollos, la selección de sensores se basa en la necesidad de medir con precisión y en tiempo real las condiciones ambientales que afectan directamente la salud y el bienestar de las aves. Los sensores elegidos deben ser capaces de medir variables clave como la temperatura, la humedad y la calidad del aire, en particular, la concentración de amoníaco, esto se lo debe hacer de manera continua y confiable.

Elección del sensor de Temperatura. El sensor de temperatura es un componente clave en el sistema de monitoreo del galpón, ya que la temperatura influye directamente en el bienestar, la salud y el desarrollo de los pollos. Una temperatura inadecuada puede provocar estrés térmico, afectando su crecimiento e incrementando la mortalidad. Por esta razón, se seleccionó un sensor de alta precisión y confiabilidad que cumpla con las condiciones ambientales del galpón. Por ello se realiza un análisis comparativo de tres modelos actualmente disponibles en tiendas locales, como se observa en la Tabla 8.

Tabla 8

Comparación de sensores de temperatura

Sensores de Temperatura

Parámetros	DHT22	SHT31	AM2302
Voltaje de alimentación	3.3V – 6V	2.4V – 5.5V	3.3V – 6V
Tamaño de módulo	15.1 x 25 mm	2.5 x 2.5 x 0.9 mm	15.1 x 25 x 7.7 mm
Rango de medición	-40°C a 80°C	-40°C a 125°C	-40°C a 80°C
Precisión	±0.5°C	±0.3°C	±0.5°C
Tiempo de respuesta	≤ 2 seg	≤ 1 seg	≤ 2 seg
Conector	Pin de 3 pines	Pin de 4 pines	Pin de 3 pines
Consumo de energía	< 2.5 mA	< 1 mA	< 2.5 mA
Interfaz	Digital	Digital	Digital
Indicador de encendido	No	No	No

Fuente: Autoría

Luego de realizar la tabla comparativa de los sensores de temperatura, a continuación, se procederá a seleccionar el sensor más adecuado como se observa en la Tabla 9. Esta elección se basa en los parámetros evaluados, como precisión, rango de medición, tiempo de respuesta y consumo energético, asegurando que el sensor seleccionado cumpla con las necesidades específicas del proyecto.

Tabla 9

Elección del sensor de temperatura

Parámetros	Sensor de Temperatura		
	DHT22	SHT31	AM2302
SySR1	1	1	1
SySR6	1	1	1
SySR10	0	0	0
SRSH3	1	1	1
Puntuación	3	3	3
Cumple “1”			
No cumple “0”			
Disponibilidad	Ibarra	Quito	Quito
Precio	\$15.00	\$11,00	\$20,00
Elección: DHT22			

Fuente: Autoría

El DHT22 fue seleccionado como el sensor de temperatura y humedad para el sistema de monitoreo debido a su excelente relación costo-beneficio. Con un precio accesible de \$15.00 USD, este sensor ofrece características técnicas que cumplen con los requerimientos del proyecto, como un amplio rango de medición de temperatura (-40°C a 120°C) y humedad (0% a 100%), junto con una precisión de $\pm 0.5^{\circ}\text{C}$ y $\pm 2\%$ HR. Estas especificaciones aseguran lecturas confiables y suficientes para monitorear el ambiente del galpón.

Elección del sensor de Amoniaco. El sensor de amoníaco (NH_3) es un componente esencial en el sistema de monitoreo del galpón de crianza de pollos, ya que la acumulación de este gas puede tener efectos perjudiciales en la salud respiratoria de las aves, comprometiendo su bienestar y productividad. Una concentración elevada de amoníaco, generada por la descomposición de excrementos, puede provocar enfermedades respiratorias y reducir la calidad del aire en el galpón. Por esta razón, es fundamental seleccionar un sensor de alta sensibilidad y precisión que permita monitorear de manera continua los niveles de NH_3 , garantizando un ambiente saludable para las aves.

Con este objetivo, se realizó un análisis comparativo de tres modelos de sensores disponibles en el mercado local, evaluando características clave como rango de medición, precisión, tiempo de respuesta y consumo energético. Los detalles de este análisis se presentan en la Tabla 10.

Tabla 10

Comparación de sensores de NH_3

Sensores NH_3			
Parámetros	MQ135	ZE08-CH2O	MQ137
Voltaje de alimentación	2.5V - 5V	3.7V - 5V	5V
Tamaño de módulo	18 x 18 x 9 mm	40 x 20 x 10 mm	18 x 18 x 9 mm
Rango de medición	10 ppm-1000 ppm	0.01 ppm - 1.5 ppm	0.01 ppm - 1.5 ppm
Precisión	$\pm 5\%$	-	$\pm 5\%$

Tiempo de respuesta	≤ 30 seg	≤ 15 seg	≤ 30 seg
Conector	Pin de 4 pines	Pin de 4 pines	Pin de 4 pines
Consumo de energía	< 150 mA	< 200 mA	< 150 mA
Interfaz	Digital/Analógica	UART/Analógica	Digital/Analógica
Indicador de encendido	Si	Si	No

Fuente: Autoría

Luego de realizar la tabla comparativa de los sensores de amoníaco, a continuación, se procederá a seleccionar el sensor más adecuado, como se observa en la Tabla 11. Esta elección se basa en los parámetros evaluados, como rango de medición, precisión, tiempo de respuesta y compatibilidad, asegurando que el sensor seleccionado cumpla con las necesidades específicas del proyecto.

Tabla 11

Elección del sensor de temperatura

Parámetros	Sensor de Amoniaco		
	MQ135	ZE08-CH2O	MQ137
SySR1	1	1	1
SySR6	1	1	1
SySR10	0	0	0
SRS3	1	0	1
Puntuación	3	2	3
Cumple “1”			
No cumple “0”			
Disponibilidad	Ibarra	Quito	Ibarra
Precio	\$10.00	\$22.00	\$11.00
Elección: MQ135			

Fuente: Autoría

El MQ135 fue seleccionado como el sensor de amoníaco para el sistema de monitoreo debido a su buena relación costo-beneficio. Con un precio accesible de \$10.00 USD, este sensor ofrece características técnicas que cumplen con los requerimientos del proyecto. Estas

especificaciones permiten detectar concentraciones de amoníaco en el aire con precisión, lo que resulta ideal para monitorear ambientes avícolas, donde este gas es un contaminante. Con una respuesta rápida y un tiempo de calentamiento relativamente corto, el MQ135 es una opción eficiente y económica para garantizar un monitoreo ambiental adecuado.

Elección del sensor de Nivel para alimento y agua. El sensor ultrasónico es un componente clave en el sistema de monitoreo del galpón de crianza de pollos, ya que su función es garantizar un control eficiente de los niveles de alimento en los comederos y agua en los bebederos. La disponibilidad constante de alimento y agua es crucial para mantener el crecimiento óptimo de las aves y así asegurando su bienestar y productividad. La falta de alimento o agua puede generar estrés en las aves y afectar negativamente su desarrollo, mientras que el desperdicio de recursos puede incrementar los costos operativos.

Por esta razón, se requiere un sensor que ofrezca precisión y confiabilidad para detectar los niveles en tiempo real y permitiendo una gestión eficiente del suministro. En este contexto, se llevó a cabo un análisis comparativo de tres modelos de sensores ultrasónicos como se observa en la Tabla 12, considerando características como rango de detección, precisión, consumo energético y compatibilidad.

Tabla 12

Comparativa de sensores de nivel para alimento y agua

Sensores de proximidad			
Parámetros	HC-SR04	VL53-0X	APDS-9960
Tipo de Sensor	Ultrasónico	Infrarrojo	Óptico
Distancia de Detección	2 cm - 400 cm	3 cm - 80 cm	2 cm - 10 cm
Alimentación	5V DC	5V DC	3.3V-5V DC
Interfaz	Digital	Digital	I2C
Consumo de Energía	Bajo (< 15 mA)	Bajo (< 10 mA)	Muy bajo (< 5 mA)

Resistencia al Ambiente Baja Media Alta

Fuente: Autoría

Para la selección del sensor de nivel que será utilizado en los comederos y bebederos del sistema y se evaluaron diferentes opciones disponibles en el mercado.. En la Tabla 13 se presenta un análisis comparativo entre tres modelos de sensores que cumplen con estas especificaciones.

Tabla 13

Análisis comparativo para sensores de proximidad

Parámetros	Sensores de proximidad		
	HC-SR04	VL53-0X	APDS-9960
SySR1	1	1	1
SySR6	1	1	1
SRSH3	0	1	1
SRSH7	1	1	1
Puntuación	3	4	4
Cumple “1”			
No cumple “0”			
Disponibilidad	Ibarra	Ibarra	Ibarra
Precio	\$6.00	\$10.00	\$15.00
Elección: VL53-0X			

Fuente: Autoría

3.3.1.2. Bloque de Recolección de Datos

Para la selección del microcontrolador se realizó una tabla comparativa, como se observa en la Tabla 14, en la que se consideran las especificaciones clave de cada modelo, tales como la cantidad de pines de entrada/salida, la memoria disponible, y la velocidad del procesador. Esta comparación permite verificar cuál de los microcontroladores cumple con los requerimientos específicos del proyecto, tales como la capacidad de manejar múltiples sensores y procesar los datos en tiempo real. El microcontrolador seleccionado será el encargado de

recibir y procesar la información proveniente de los sensores del galpón, y luego transmitirla al controlador de comunicación. Esta etapa es crucial para asegurar que los datos sean enviados de manera eficiente hacia el servidor o la plataforma de visualización, permitiendo el monitoreo constante de las condiciones ambientales en el galpón de pollos.

Tabla 14

Comparación de microcontroladores para toma de datos

Parámetros	Microcontroladores			
	ESP32-WROOM-32	Raspberry Pi Pico W	NodeMCU ESP8266	Arduino MKR WiFi 1010
Microprocesador	Dual-Core 240 MHz	Dual-Core ARM Cortex M0+	Tensilica L106 80 MHz	ARM Cortex-M0+ 32-bit
RAM	520 KB	264 KB	160 KB	32 KB
Conectividad WiFi	802.11 b/g/n	802.11 b/g/n	802.11 b/g/n	802.11 b/g/n
Memoria Flash	4 MB	2 MB	4 MB	256 KB
Pines de Entrada/Salida	34	26	11	22
Consumo de energía	Bajo (100mA)	Moderado (120mA)	Bajo (80mA)	Moderado (120mA)
Conectividad Bluetooth	Sí (BLE y Clásico)	No	No	Sí (BLE)

Fuente: Autoría

Se realiza un análisis comparativo entre los microcontroladores como se ve en la Tabla 15, esto con el fin de que se elija la mejor opción para el funcionamiento del proyecto.

Tabla 15

Análisis comparativo de los microcontroladores

Parámetros	Microcontroladores			
	ESP32-WROOM-32	Raspberry Pi Pico W	NodeMCU ESP8266	Arduino MKR WiFi 1010
SySR7	1	1	1	1
SySR11	0	0	0	0

SySR1	1	0	1	0
SySR2	1	0	1	0
SRSH3	1	1	1	1
Puntuación	4	2	4	2
Cumple “1”				
No cumple “0”				
Disponibilidad	Ibarra	Ibarra	Ibarra	Ibarra
Precio	\$16.00	\$10.00	\$12.00	\$35.00
Elección: ESP32-WROOM-32				

Fuente: Autoría

El microcontrolador seleccionado para la comunicación es el ESP32-WROOM-32, ya que, tras el análisis comparativo realizado, se determinó que es la opción más adecuada para este proyecto. Su capacidad de conectividad, combinada con su alto rendimiento, bajo consumo de energía y compatibilidad con una amplia gama de sensores y módulos, lo convierten en la elección ideal para garantizar una transmisión eficiente de datos hacia el bloque de procesamiento.

El siguiente paso en el diseño del proyecto es el análisis y selección del nodo de procesamiento, en el cual se evaluarán las opciones más viables para manejar y procesar los datos recolectados, asegurando un flujo continuo y eficiente en toda la red del sistema de monitoreo.

3.3.1.3. Bloque de Procesamiento

El bloque de procesamiento es una de las partes más críticas del sistema de monitoreo en el galpón de crianza de pollos, ya que se encarga de recibir, analizar y gestionar los datos provenientes de los sensores instalados en el galpón. Este bloque actúa como el cerebro del sistema, transformando las señales recopiladas en información útil para la toma de decisiones y la transmisión hacia la nube o plataformas de visualización.

A continuación, se realiza una comparativa entre versiones de Raspberry Pi para el bloque de procesamiento como se observa en la Tabla 17, justificando su elección en función de los

requerimientos del proyecto y la capacidad de cumplir con las necesidades del sistema de monitoreo, se realiza también un análisis de las placas de procesamiento como se observa en la Tabla 16.

Tabla 16

Comparación de placas para Procesamiento

Placas para Procesamiento						
Parámetros	Raspberry Pi 4 B		Raspberry Pi 3 B+		Raspberry Pi Zero 2 W	
Microprocesador	Quad-Core	ARM Cortex-A72	Quad-Core	ARM Cortex-A53 1.4GHz	Quad-Core	ARM Cortex-A53 1GHz
RAM	2GB/4GB/8GB		1GB		512MB	
Almacenamiento	MicroSD	(hasta 64GB+)	MicroSD	(hasta 64GB+)	MicroSD	(hasta 32GB)
Conectividad WiFi	802.11 b/g/n/ac		-802.11 b/g/n/ac		802.11 b/g/n	
Conectividad Bluetooth	5.0		4.2		4.2	
Puertos USB	2x USB 3.0,	2x USB 2.0	4x USB 2.0		1x USB 2.0	
Puertos GPIO	40 pines		40 pines		40 pines	
Consumo de Energía	3.0A @ 5V		2.5A @ 5V		1.0A @ 5V	

Fuente: Autoría

Tabla 17

Análisis de placas para Procesamiento

Parámetros	Placas de Procesamiento		
	Raspberry Pi 4 B	Raspberry Pi 3 B+	Raspberry Pi Zero 2 W
SRSH1	1	1	0
SRSH7	1	0	0
SRSH11	1	1	1
SRSH13	1	1	1

SySR3	1	1	1
Puntuación	4	3	3
Cumple “1”			
No cumple “0”			
Disponibilidad	Guayaquil	Ibarra	Quito
Precio	\$110.00	\$90.00	\$40.00
Elección: Raspberry Pi 4 B			

Fuente: Autoría

La Raspberry Pi 4 Model B fue seleccionada como la placa de procesamiento para el proyecto debido a sus características técnicas avanzadas, que la convierten en la mejor opción para manejar las necesidades específicas del sistema de monitoreo en un galpón de crianza de pollos.

3.3.1.4. Bloque de Actuadores

Para la elección del bloque de actuadores se debe tomar en cuenta que tiene como objetivo ejecutar acciones físicas en función de los datos recopilados por los sensores y las decisiones del sistema. Su objetivo es mantener condiciones óptimas en el galpón mediante la activación o desactivación de dispositivos como ventiladores, calentadores eléctricos y sistemas de iluminación. La Tabla 18 muestra la elección de dichos componentes.

Tabla 18

Actuadores para el sistema automatizado

Actuadores			
Parámetros	Calentador Infrarrojo	Ventilador	Foco Convencional
Consumo eléctrico	250W	27.5W	10W
Precio	\$15.00	\$20	\$5
Cantidad	1	1	1

Fuente: Autoría

3.3.1.5. Bloque de Alimentación Eléctrica

Fuente de corriente AC. El bloque de alimentación eléctrica es un componente esencial para garantizar el funcionamiento continuo del sistema de monitoreo implementado en el galpón de crianza de pollos. Se ha seleccionado el uso de corriente alterna (CA) de 110V como la fuente principal de energía, proveniente de la red eléctrica convencional. Esta elección permite un suministro constante y estable, asegurando el correcto desempeño de los sensores, el microcontrolador ESP32 y la unidad de procesamiento Raspberry Pi.

Para adaptar la corriente alterna a los requerimientos de los componentes electrónicos, el sistema incluye un convertidor de corriente alterna a continua (AC/DC), que transforma la energía en un voltaje adecuado (5V o 12V DC) para el funcionamiento de los dispositivos. Este convertidor garantiza una alimentación segura y eficiente, evitando fluctuaciones que puedan dañar los componentes.

Fuente de corriente DC. El sistema es compatible con el uso de baterías recargables, lo que lo hace adaptable a entornos remotos donde no haya acceso a la red eléctrica. En estos casos, las baterías pueden actuar como la fuente principal de energía, permitiendo que el sistema funcione de manera autónoma. Para este propósito, se recomienda el uso de baterías de 12V en combinación con reguladores de voltaje para proporcionar la alimentación adecuada a cada dispositivo. Esta flexibilidad en el diseño del sistema asegura su viabilidad tanto en ubicaciones conectadas a la red eléctrica como en entornos rurales o aislados.

Cálculo de Corriente y Voltaje en el Bloque de Toma de Datos. El bloque de toma de datos incluye dispositivos para medir las condiciones ambientales del galpón, como temperatura, humedad, calidad del aire y niveles de alimento y agua.

Los componentes seleccionados son el SHT31 para temperatura y humedad, el MQ135 para calidad de aire, y el E18-D80NK para detección de niveles y se incluye también la placa usada para este procedimiento. Estos dispositivos operan a voltajes de 3.3V y 5V, con

corrientes que van desde 2 mA hasta 150 mA. El consumo total del bloque se calcula sumando las contribuciones de cada sensor como se observa en la Tabla 19.

Tabla 19

Voltaje y corriente en bloque de Toma de Datos

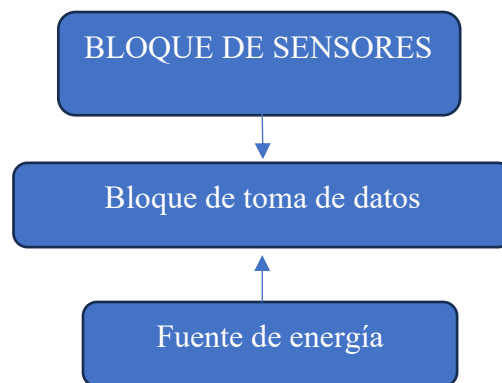
#	Cantidad	Componente	Voltaje (V)	Corriente (A)
1	1	SHT31	3.3V	0.002 A
2	1	MQ135	5V	0.150 A
3	1	E18-D80NK	5V	0.010 A
4	1	ESP32-WROOM-32	3.3V	0.240 A
Total, voltaje			5V	
Total, corriente				0.402 A

Fuente: Autoría

En la Figura 15 se observa el diagrama de bloques de conexiones realizadas para la conexión eléctrica para los sensores.

Figura 15

Diagrama de bloques de sensores



Fuente: Autoría

Cálculo de Corriente y voltaje en el Bloque de Procesamiento. El bloque de procesamiento está conformado por la Raspberry Pi 4 Model B como se observa en la Tabla 20, que gestiona el análisis y almacenamiento de los datos provenientes del sistema. Este dispositivo opera a 5V con un consumo máximo de 3000 mA, siendo el componente de mayor demanda energética en el sistema. La potencia se calcula como el producto del voltaje y la corriente.

Tabla 20

Voltaje y Corriente en Bloque de Procesamiento

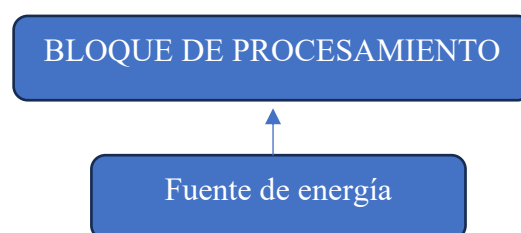
#	Cantidad	Componente	Voltaje (V)	Corriente (A)
1	1	Raspberry Pi 4 Model B	5V	3.000 A
Total, voltaje			5V	
Total, corriente				3.000 A

Fuente: Autoría

En el diagrama de bloques de la Figura 16 se observa cómo está conectado el Bloque de Procesamiento.

Figura 16

Diagrama de Bloque de Procesamiento



Fuente: Autoría

Cálculo de Voltaje y Corriente en el Bloque de Actuadores. El bloque de actuadores está conformado por el calentador infrarrojo, el ventilador y el foco convencional, por ello se realiza un análisis del consumo de corriente y voltaje de cada uno de estos componentes, como se observa en la Tabla 21.

Tabla 21

Voltaje y Corriente en Bloque de Actuadores

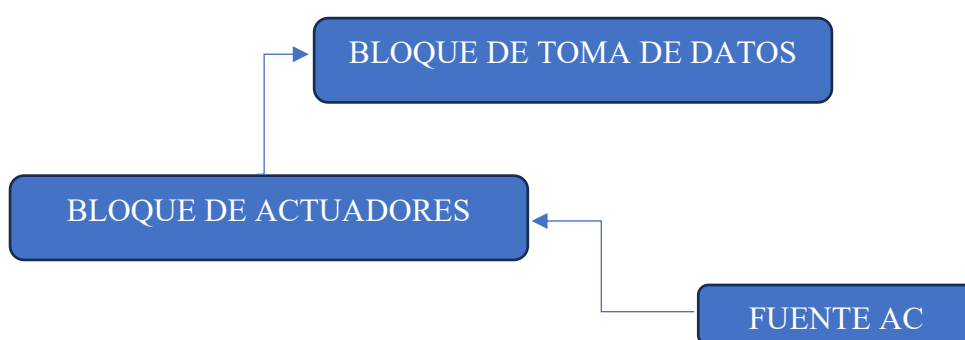
#	Cantidad	Componente	Voltaje (V)	Corriente (A)
1	1	Calentador Infrarrojo	110V	2.27A
2	1	Ventilador	110V	0.25A
3	1	Foco convencional	110V	0.091A
Total, voltaje			110V	
Total, corriente				2.611 A

Fuente: Autoría

En el diagrama de bloques de la Figura 17 se observa cómo está conectado el Bloque de Procesamiento.

Figura 17

Diagrama de Bloques de Actuadores



Fuente: Autoría

Cálculo de Corriente y Voltaje Total del Sistema. Se realiza el cálculo correspondiente para todo el sistema, para ello se toma en cuenta todos los bloques en conjunto y se estima un consumo, como se observa en la Tabla 22.

Tabla 22

Voltaje y Corriente en el Sistema Automatizado

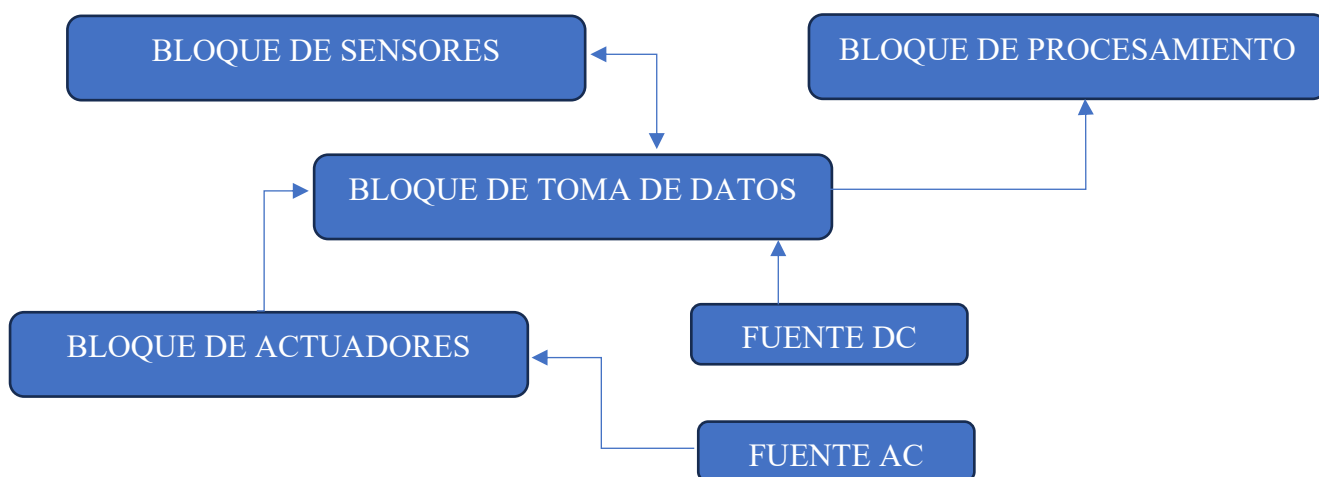
#	Cantidad	Bloque	Voltaje (V)	Corriente (A)
1	1	Bloque de Sensores y Toma de Datos	3.3V	0.402 A
2	1	Bloque de Procesamiento	5V	3.000 A
3	1	Bloque de Actuadores	3.3V	0.240 A
Total, voltaje			5V	
Total, corriente				3.642 A

Fuente: Autoría

En el diagrama de bloques de la Figura 18 se observa cómo está conectado todo el sistema, considerando todos los bloques.

Figura 18

Diagrama de Bloques del Sistema Automatizado



Fuente: Autoría

3.3.2 Software

Realizada la elección del hardware, se procede a seleccionar el software en función de los requerimientos establecidos en el diseño del sistema. Este proceso asegura un desempeño óptimo, considerando la integración eficiente entre hardware y software. Se priorizan herramientas de uso libre, accesibles y compatibles con los componentes seleccionados, garantizando flexibilidad y escalabilidad en el desarrollo del proyecto.

3.3.2.1. Software de Programación del Bloque de Toma de Datos

El software para este bloque de determina en base a la elección de hardware que se hace en la Tabla 15, por ello se debe elegir el software correcto para que el funcionamiento de adecuado de la placa seleccionada, por ello se escoge el más indicado como se observa en la Tabla 23.

Tabla 23

Elección de software de programación Bloque de Toma de Datos

Parámetros	Lenguaje de Programación		
	Python	C++	IDE Arduino
SRSH7	0	0	1
SRSH14	0	0	1
Puntuación	0	0	2
Cumple “1”			
No cumple “0”			
Elección: IDE Arduino			

Fuente: Autoría

Para el bloque de sensores y toma de datos, se utilizará el lenguaje de programación de la IDE Arduino, ya que es una plataforma ampliamente reconocida por su facilidad de uso, compatibilidad con el microcontrolador ESP32 y su extensa comunidad de soporte.

3.3.2.2. Software para Bloque de Procesamiento

El software para este bloque de determina en base a los requerimientos que se hace en la Tabla 17.

Elección de plataforma de Bloque de Procesamiento. En la Tabla 24 se presenta un análisis de las distintas plataformas de programación que fueron evaluadas en función de los requerimientos del sistema. La selección de la plataforma debe garantizar compatibilidad y disponibilidad con el hardware implementado en el proyecto. En dicha tabla se muestra un cuadro comparativo que detalla las características de cada una de las plataformas propuestas para llevar a cabo el procesamiento de la información y la elección de la más adecuada se realiza en la Tabla 25.

Tabla 24

Comparación de plataformas de procesamiento

Plataforma de Procesamiento			
Parámetros	Node-RED	Apache Kafka	ThingsBoard
Tipo de Plataforma	Flujo visual de datos (No-code)	Middleware para streaming de datos	Plataforma IoT (Interfaz gráfica)
Lenguajes Soportados	JavaScript	Multilenguaje (Java, Python, Scala)	Java, Python
Requerimientos	Bajos (Raspberry Pi compatible)	Altos (CPU y RAM intensivos)	Moderados (Servidor o Raspberry Pi)
Protocolos	MQTT, HTTP, WebSocket	HTTP, TCP, UDP	MQTT, CoAP, HTTP
APIs	REST, JSON	REST, JSON, Protobuf	REST, JSON
Capacidades	Transmisión masiva de datos en tiempo real	Transmisión masiva de datos en tiempo real	Dashboard, análisis de datos, alertas

Fuente: Autoría

Tabla 25*Elección del software de Programación*

Parámetros	Plataforma de Programación		
	Node-RED	Apache Kafka	ThingsBoard
SRSH7	1	0	1
SRSH14	1	1	1
SRSH17	1	1	1
Puntuación	3	2	3
Cumple “1”			
No cumple “0”			

Elección: Node-RED

Fuente: Autoría

Node-RED se selecciona como la plataforma para el bloque de procesamiento debido a su facilidad de uso, interfaz visual intuitiva y amplia compatibilidad con hardware como la Raspberry Pi. Soporta protocolos esenciales como MQTT y HTTP, lo que facilita la integración con dispositivos IoT y el envío de datos hacia plataformas en la nube.

3.3.2.3. Hosting

Para la elección del hosting se hace la comparativa de dos alternativas se exponen en la Tabla 26, en donde se especifican sus características detalladamente y en la Tabla 27 se realiza la elección de la opción más adecuada en base a los requerimientos.

Tabla 26*Cuadro comparativo de Hosting*

Hosting		
Parámetros	Ngrok	A2 Hosting
Tipo de Hosting	Compartido, VPS, Dedicado	Compartido, VPS, En la nube
Panel de Control	cPanel	Plesk o cPanel
Compatibilidad	HTTP, HTTPS, Node.js, bases de datos SQL	Altos (CPU y RAM intensivos)

Escalabilidad	Moderada (planes avanzados requeridos)	Alta (planes flexibles en VPS y nube)
Rendimiento	Sólido para proyectos medianos	Alta velocidad y rendimiento optimizado
Precio Inicial	\$10/mes (VPS básico)	\$7/mes (compartido) o \$15/mes (VPS)

Fuente: Autoría

Tabla 27

Elección de Hosting

Parámetros	Hosting	
	Ngrok	A2 Hosting
SRSH4	1	1
SRSH7	1	0
SRSH16	1	1
Puntuación	3	2
Cumple “1”		
No cumple “0”		

Elección: Ngrok

Fuente: Autoría

Ngrok fue seleccionado como la plataforma de hosting ya que debido a su facilidad de uso e integración directa con cPanel. Estas características garantizan una gestión eficiente del sistema y un entorno estable para el manejo de datos.

3.3.2.4. Servidor Web

Este servidor es responsable de recibir, procesar y responder a las solicitudes de información realizadas a través de interfaces web, permitiendo visualizar y gestionar los datos recolectados de los sensores.

Para este apartado se deben considerar dos opciones de servidores web como se observa en la Tabla 28, y posteriormente se realiza la elección de la opción más adecuada como se observa en la Tabla 29.

Tabla 28*Cuadro comparativo de servidores web*

Servidores Web							
Parámetros		Apache			Nginx		
Modelo	de	Proceso	por	solicitud	Eventos	asincrónicos	(no
Procesamiento		(multiproceso)			bloqueante)		
Rendimiento		Mejor para sitios dinámicos			Plesk o cPanel		
Compatibilidad		Altamente compatible con aplicaciones PHP			Soporte eficiente para Node.js y microservicios		
Escalabilidad		Buena, pero limitada en cargas muy altas			Excelente para sitios de alta concurrencia		
Facilidad	de	Relativamente sencillo, pero			Más moderno, puede ser		
Configuración		menos moderno			complejo para principiantes		
Precio		Gratuito (código abierto)			Gratuito (código abierto)		

Fuente: Autoría

Tabla 29*Elección de Servidor Web*

Parámetros	Servidor Web	
	Apache	Nginx
SRSH4	1	1
SRSH16	1	1
SRSH17	1	1
Puntuación	3	3
Cumple “1”		
No cumple “0”		

Elección: Apache

Fuente: Autoría

Apache se selecciona como el servidor web debido a su alta compatibilidad con aplicaciones dinámicas y su facilidad de integración con tecnologías como PHP y bases de

datos SQL. Estas características lo convierten en una opción confiable para gestionar el servidor web en el sistema.

3.3.2.5. Base de Datos

Este apartado tiene como objetivo seleccionar una solución de base de datos que garantice un manejo eficiente de los datos, alta disponibilidad y compatibilidad con las tecnologías utilizadas en el sistema. Para ello se realiza la comparación en la Tabla 30 y la elección de la base de datos que se acople al sistema y cumpla con los requerimientos, se elige la opción más adecuada en la Tabla 31.

Tabla 30

Comparativa para Base de Datos

Parámetros	Base de Datos	
	MySQL	MongoDB
Modelo de Datos	Relacional (Tablas)	No relacional
Estructura	Esquema rígido con relaciones predefinidas	Esquema flexible y dinámico
Rendimiento	Mejor en transacciones complejas y estructuradas	Mejor en grandes volúmenes de datos no estructurados
Escalabilidad	Vertical (servidor más potente)	Horizontal (múltiples nodos)
Consultas	SQL (lenguaje estándar)	Consultas basadas en JSON
Precio	Gratuito (código abierto)	Gratuito (código abierto)

Fuente: Autoría

Tabla 31

Elección de Base de Datos

Parámetros	Base de Datos	
	MySQL	NoSQL
SRSH14	1	1
SRSH15	1	1
SySR7	1	1
Puntuación	3	3

Cumple “1”

No cumple “0”

Elección: MySQL

Fuente: Autoría

MySQL se selecciona como la base de datos debido a su modelo relacional, que permite estructurar los datos de manera organizada y eficiente. Su compatibilidad con herramientas tradicionales y su lenguaje estándar SQL garantizan facilidad en la gestión y consulta de los datos.

3.3.2.5. Plataforma de Notificación

La plataforma de notificación es un componente clave del sistema, ya que permite alertar a los usuarios sobre eventos relevantes en tiempo real. Este apartado evalúa las mejores opciones para implementar un sistema de notificaciones eficiente, considerando factores como facilidad de uso, alcance, compatibilidad y costos operativos.

En la Tabla 32 se observan las opciones consideradas incluyen WhatsApp, Telegram y correo electrónico, las cuales serán analizadas en función de los requerimientos del proyecto, y posteriormente se elige la mejor opción en la Tabla 33.

Tabla 32

Comparativa entre plataformas de notificación

Plataforma de Notificación				
Parámetros	WhatsApp		Telegram	Correo Electrónico
Facilidad de Implementación	Media (API de pago)	Alta (API gratuita)		Alta (bibliotecas estándar)
Costo	API oficial es de pago		Gratuito	Gratuito
Velocidad de Notificación	Limitada (depende de la API)		Alta (bots y comandos)	Baja (sólo notificaciones)
Interactividad	MQTT, WebSocket	HTTP, HTTP, TCP, UDP		MQTT, CoAP, HTTP

Requerimientos	Conexión a WhatsApp Business API	Bot y acceso a API	Servidor de correo configurado
----------------	----------------------------------	--------------------	--------------------------------

Fuente: Autoría

Tabla 33

Elección de Plataforma de Notificación

Parámetros	Plataforma de Notificación		
	WhatsApp	Telegram	Correo Electrónico
StSR6	1	1	1
SySR5	0	1	0
SRS5	0	1	1
SRS18	1	1	0
SRS19	1	1	0
Puntuación	3	5	2

Cumple “1”

No cumple “0”

Elección: Telegram

Fuente: Autoría

Telegram se selecciona como la plataforma de notificación debido a su facilidad de implementación, API gratuita y su alta compatibilidad con dispositivos móviles y de escritorio. Ofrece una interacción avanzada mediante el uso de bots y comandos, lo que permite personalizar las notificaciones y añadir funcionalidades como respuestas automáticas.

Su capacidad de escalabilidad y rapidez en la entrega de mensajes lo convierten en una opción ideal para sistemas de monitoreo que requieren alertas en tiempo real.

3.4. Diseño del Sistema

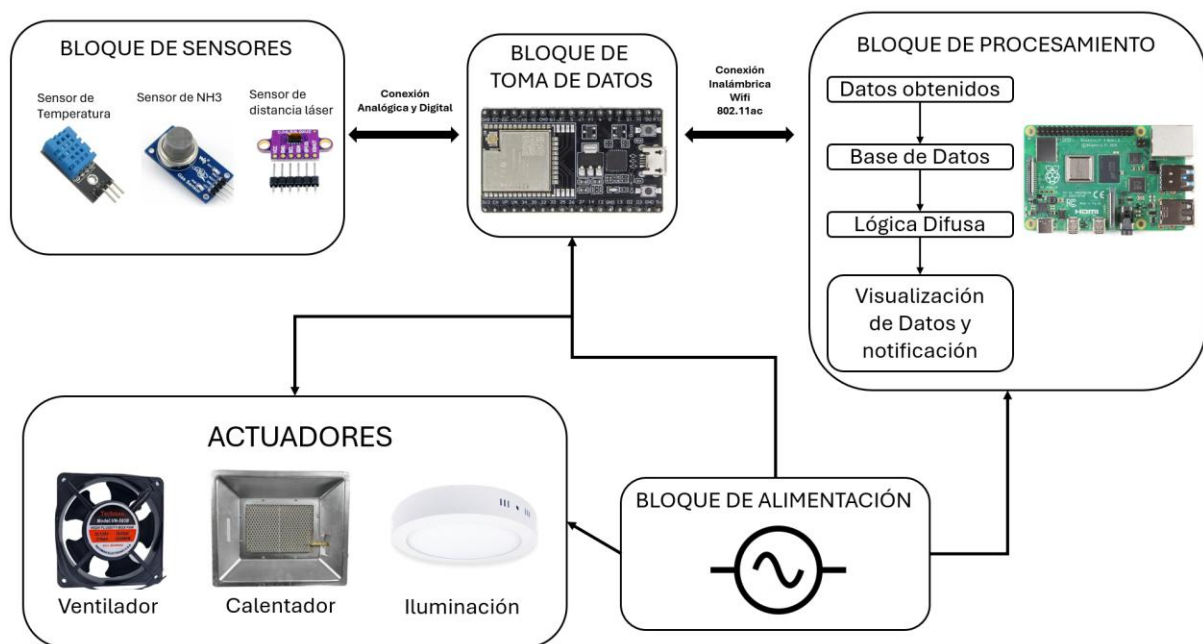
Después de seleccionar los componentes de hardware y software que integran el sistema de monitoreo, se procede a estructurar el diseño general del sistema mediante un esquema basado en bloques. La Figura 19 representa de manera gráfica los bloques principales que conforman el sistema y cómo interactúan entre sí para cumplir con los objetivos planteados.

3.4.1 Diagrama de Bloques del Sistema de Monitoreo

En esta sección se presenta un diagrama estructurado en tres bloques principales, cada uno de los cuales representa una parte esencial del sistema de monitoreo. Cada bloque desempeña funciones específicas que garantizan el correcto funcionamiento del sistema en su conjunto. En la Figura 19 se detalla la disposición y las interacciones de estos bloques.

Figura 19

Diagrama de Bloques del Sistema de Monitoreo



Fuente: Autoría

El sistema de monitoreo se divide en cuatro bloques principales que interactúan entre sí para garantizar un funcionamiento eficiente. El Bloque de Sensores es el encargado de recolectar datos del ambiente dentro del galpón. Incluye sensores como el DHT22 para medir la temperatura y humedad, el MQ135 para monitorear la concentración de amoníaco, y el sensor de distancia, que evalúa el nivel de alimento en los comederos. Toda la información captada por estos sensores es enviada al siguiente módulo para su procesamiento inicial.

El Bloque de Toma de Datos está compuesto por un microcontrolador ESP32, cuya función principal es centralizar los datos recolectados por los sensores, realizar un

preprocesamiento básico y enviarlos al bloque de procesamiento. Este microcontrolador utiliza comunicación inalámbrica mediante WiFi, lo que asegura una transmisión eficiente de la información.

El Bloque de Procesamiento se encarga de analizar, organizar y almacenar los datos en una base de datos, además de generar notificaciones y visualizaciones. La Raspberry Pi en este bloque gestiona la interfaz gráfica y la conexión con la plataforma de notificaciones Telegram. También integra lógica difusa para identificar patrones o condiciones críticas, lo que permite automatizar ciertas decisiones y alertas en tiempo real.

El Bloque de Alimentación suministra la energía necesaria para que todos los componentes funcionen de manera continua. Este bloque utiliza corriente alterna (110V), pero también permite la opción de funcionar con baterías para ubicaciones remotas donde no haya acceso a energía eléctrica. Este diseño asegura que el sistema pueda operar sin interrupciones y adaptarse a diferentes entornos.

El bloque de actuadores realiza acciones físicas basadas en los datos recolectados por los sensores. Por lo tanto, deben integrarse en el diseño del sistema como parte del Bloque de Toma de Decisiones. Esto implica que el ESP32 será el encargado de controlar estos dispositivos a través de relés en función de las lecturas de los sensores y luego ser tratados en el bloque de procesamiento.

3.4.2 Diagrama de Flujo de Sistema

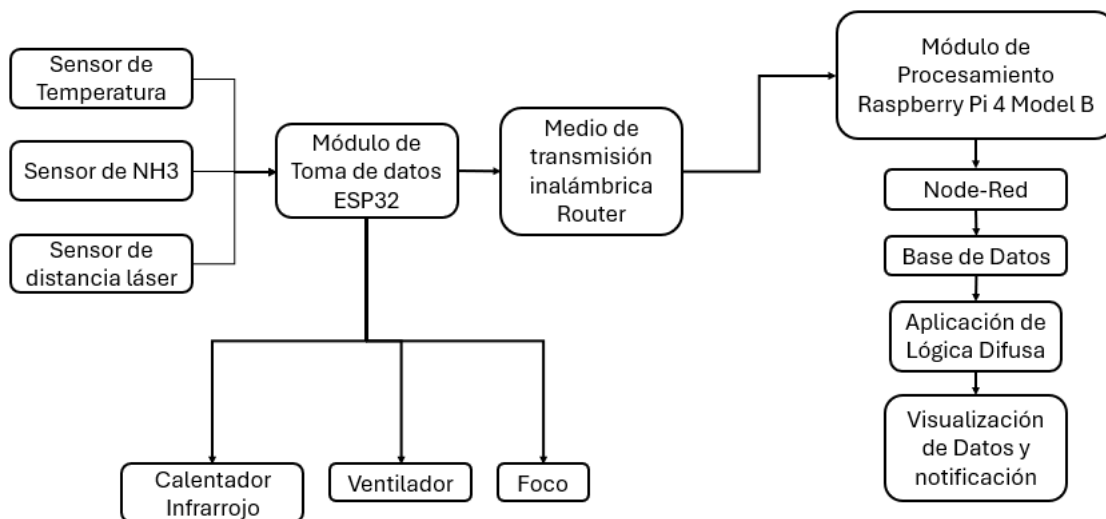
El funcionamiento del sistema de monitoreo se puede observar en la Figura 20, comienza con la activación de los sensores, encargados de medir parámetros esenciales como la temperatura, humedad, concentración de amoníaco y niveles de alimento o agua. Los datos recolectados por estos sensores se envían al microcontrolador ESP32 que realiza un preprocesamiento inicial y los formatea en un formato estándar como JSON. A través de una

conexión WiFi, el ESP32 transmite esta información al módulo de procesamiento donde interviene la Raspberry Pi.

En el módulo de procesamiento los datos son analizados mediante lógica difusa para identificar patrones o condiciones críticas como niveles peligrosos de amoníaco o temperaturas inadecuadas. Posteriormente estos datos se almacenan en una base de datos para su consulta histórica y se envían al servidor web para permitir la visualización en tiempo real.

Figura 20

Diagrama de Funcionamiento del sistema de monitoreo



Fuente: Autoría

Cuando el sistema detecta condiciones fuera de los parámetros normales, genera alertas que son enviadas a los usuarios mediante la plataforma de notificaciones como Telegram. Este flujo asegura que los usuarios puedan recibir información en tiempo real y tomar decisiones oportunas para garantizar el bienestar de los pollos en el galpón. Finalmente, el sistema reinicia el proceso para continuar con el monitoreo continuo, asegurando una vigilancia constante de las condiciones del ambiente.

3.4.3 Diseño de Hardware

En este apartado se describe el proceso de implementación de los diferentes bloques del sistema de monitoreo, detallando las acciones realizadas en cada uno y las conexiones necesarias para garantizar su correcto funcionamiento.

3.4.3.1 Bloque de Sensores y Toma de Datos

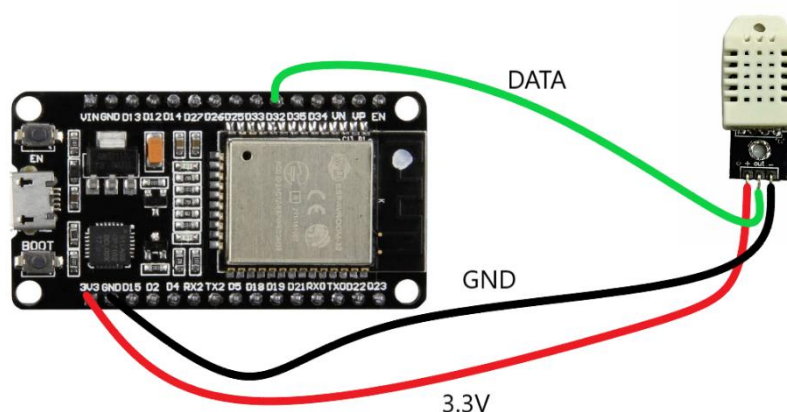
En este bloque se instalan los sensores encargados de medir los parámetros clave del galpón, como temperatura, humedad, calidad de aire (amoníaco) y nivel de alimento/agua.

Sensor de Temperatura y Humedad (DHT22). Este sensor es responsable de medir la temperatura y la humedad del galpón, parámetros cruciales para el bienestar de los pollos. El DHT22 no requiere calibración adicional ya que viene precalibrado de fábrica.

Conexión del Sensor. Se utilizan tres pines principales: alimentación (VCC), tierra (GND) y datos digitales (DATA). El pin VCC se conecta a la salida de 3.3V del ESP32. El pin GND del sensor debe conectarse al GND, y el pin DATA del DHT22 debe conectarse a un pin GPIO disponible del ESP32 como se observa en la Figura 21.

Figura 21

Diagrama de conexión entre sensor DHT22 y ESP32



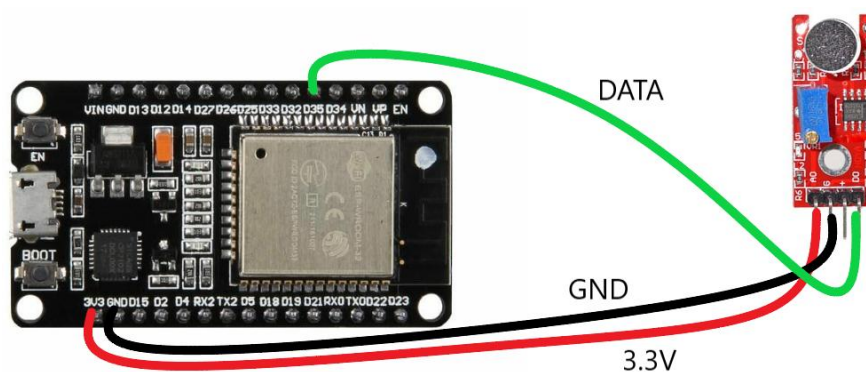
Fuente: Autoría

Sensor de Amoníaco (MQ135). Este sensor detecta la concentración de amoníaco (NH_3) en el aire, un gas que puede ser perjudicial para la salud de las aves si se encuentra en altas concentraciones. El MQ135 requiere una calibración inicial ajustando su potenciómetro en un ambiente libre de amoníaco para establecer un valor base de referencia.

Conexión del Sensor. El MQ135 se conecta al ESP32 mediante tres pines principales como se observa en la Figura 22: alimentación (VCC), tierra (GND) y una salida analógica (AO). El pin VCC del sensor debe conectarse al pin de 5V del ESP32 para garantizar un funcionamiento óptimo, aunque también puede alimentarse con 3.3V si es necesario. El pin GND del sensor se conecta al GND del ESP32. Para medir valores analógicos, el pin AO del MQ135 se conecta a un pin GPIO analógico del ESP32, como el GPIO34.

Figura 22

Diagrama de conexión entre sensor MQ135 y ESP32



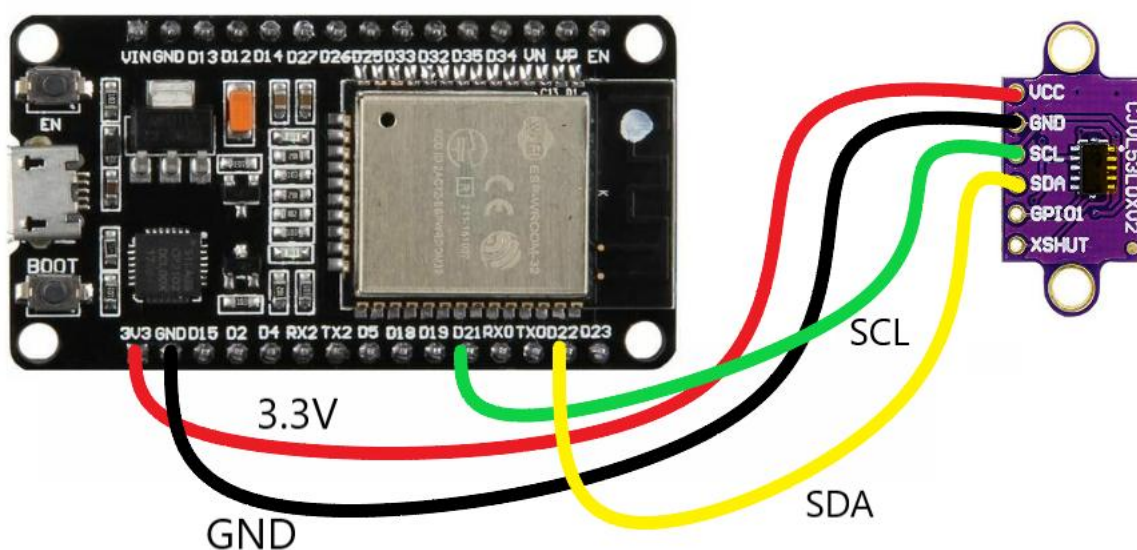
Fuente: Autoría

Sensor de Nivel (VL53L0X). Este sensor utiliza tecnología infrarroja para detectar la presencia de alimento o agua en los comederos y bebederos. Incluye un potenciómetro que permite ajustar el rango de detección según las necesidades específicas del galpón.

Conexión del Sensor. El VL53L0X se conecta al ESP32 mediante comunicación I2C, utilizando los pines SDA y SCL. El pin VIN del sensor se conecta al pin de alimentación de 3.3V, mientras que el pin GND se conecta al GND para establecer la conexión a tierra común. Los pines de comunicación, SDA y SCL, se conectan a los pines correspondientes del ESP32, generalmente GPIO21 para SDA y GPIO22 para SCL, aunque pueden variar según la configuración y la Figura 23 representa la conexión.

Figura 23

Diagrama de conexión entre sensor VL53L0X y ESP32



Fuente: Autoría

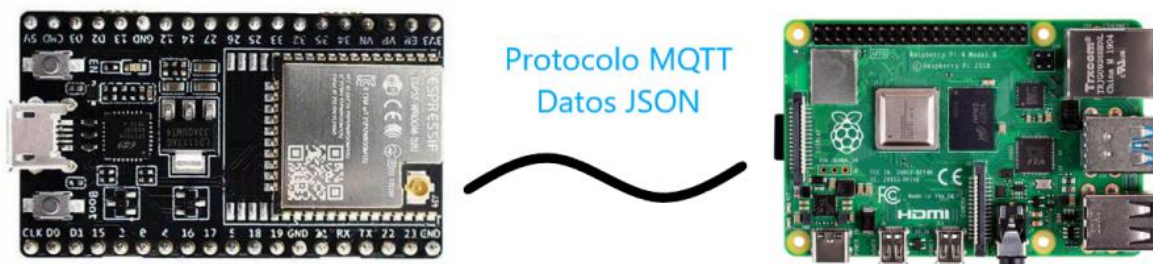
Cada uno de estos sensores desempeña un papel crucial en la recolección de datos ambientales, garantizando una monitorización precisa y en tiempo real de las condiciones del galpón. Su correcta instalación y calibración son fundamentales para el funcionamiento eficiente del sistema de monitoreo.

3.4.3.2 Bloque de Toma de Datos y Procesamiento

El bloque de procesamiento se encuentra alojado en el Raspberry Pi 4 y es responsable de recibir, procesar y gestionar los datos enviados por el microcontrolador ESP32 como se observa en la Figura 24. Este bloque utiliza herramientas como Node-RED, bases de datos y algoritmos para el análisis de los datos en tiempo real.

Figura 24

Diagrama de comunicación entre placa de Toma de Datos y Placa de Procesamiento



Fuente: Autoría

La recepción de datos comienza con el ESP32 enviando información al broker MQTT alojado en el Raspberry Pi. Node-RED captura estos datos, generalmente estructurado en JSON para facilitar su manipulación. Este proceso garantiza que la información, como temperatura, humedad y niveles de gases, llegue correctamente al sistema para su posterior análisis.

Una vez recibidos los datos son procesados en Node-RED mediante flujos predefinidos. Se aplican reglas específicas de lógica difusa para evaluar condiciones críticas, como temperaturas fuera de rango o niveles peligrosos de amoníaco. Los datos procesados se almacenan en una base de datos local para consultas futuras y análisis histórico, permitiendo un monitoreo continuo y detallado del estado del galpón.

Los datos son presentados en tiempo real a través de dashboards interactivos diseñados en Node-RED, accesibles desde cualquier navegador web. Si se detectan condiciones críticas,

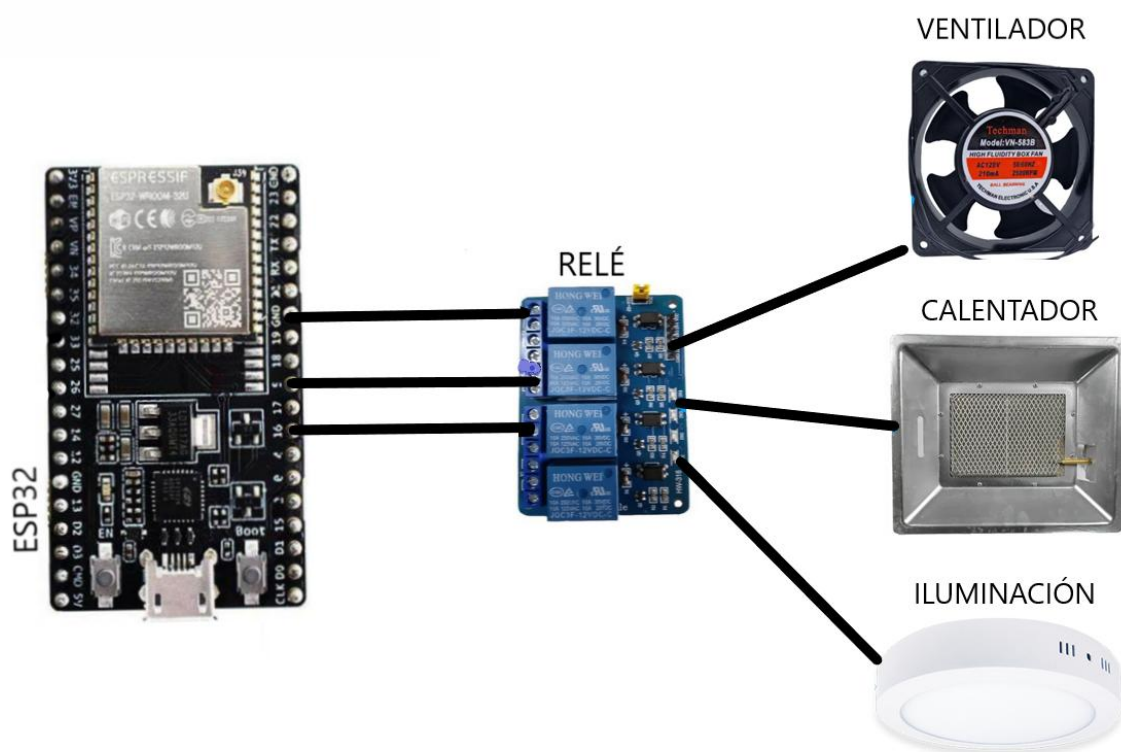
se generan notificaciones automáticas enviadas mediante Telegram, alertando al usuario para garantizar una respuesta rápida y efectiva a cualquier situación que requiera atención.

3.4.3.3. Bloque de Actuadores

El bloque de actuadores es responsable de realizar acciones físicas en respuesta a los datos recolectados por los sensores y las decisiones tomadas por el sistema. Este bloque garantiza que se mantengan las condiciones óptimas en el galpón al activar o desactivar dispositivos como ventiladores, calentadoras eléctricas y focos de iluminación. En la Figura 25 se puede observar cómo se encuentra conectado cada uno de los sensores.

Figura 25

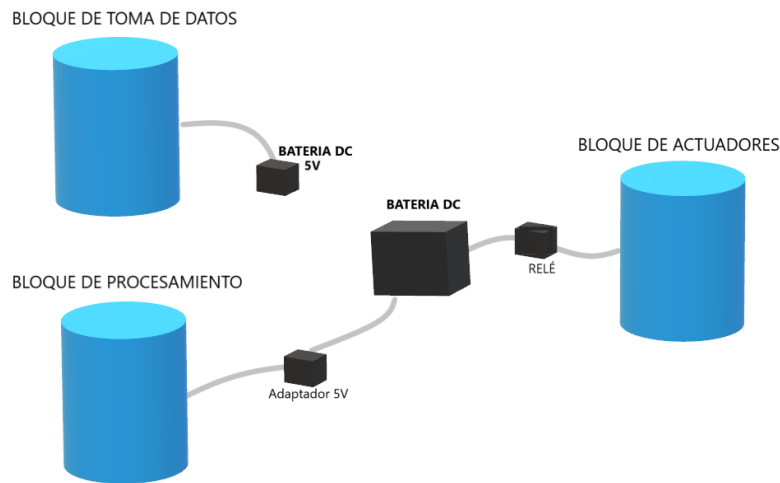
Diagrama de conexión entre placa de Toma de Datos y Actuadores



Fuente: Autoría

El flujo del bloque de actuadores comienza con la captura de datos de temperatura, hora y/o luz ambiental por parte del ESP32. Luego, el microcontrolador evalúa las condiciones actuales y compara los valores con los umbrales predefinidos. Si los datos superan o descienden de los límites establecidos se activa o desactiva el actuador correspondiente. Por ejemplo, si la

Diagrama de conexión del bloque de Alimentación Alternativo



Fuente: Autoría

La fuente de alimentación principal se encarga de convertir la corriente alterna en corriente continua de 5V mediante un adaptador AC/DC. Esta energía es utilizada directamente por el Raspberry Pi, el ESP32 y los sensores conectados. Para garantizar que el sistema funcione en lugares remotos, se implementa un sistema basado en baterías recargables de 12V como se observa en la Figura 25, las cuales se conectan a través de un regulador de voltaje que reduce la salida a los niveles requeridos por los dispositivos.

El flujo del bloque de alimentación comienza con la entrada de corriente alterna de 110V a un adaptador AC/DC que convierte la energía a 5V DC. Esta salida alimenta directamente el Raspberry Pi, mientras que un regulador adicional suministra 3.3V al ESP32.

Este diseño asegura que todos los módulos, incluidos los actuadores como el ventilador, la calentadora y los focos, puedan operar de manera continua y estable. Además, en aplicaciones remotas donde no haya acceso a la red eléctrica, las baterías pueden combinarse con paneles solares para crear un sistema completamente autónomo.

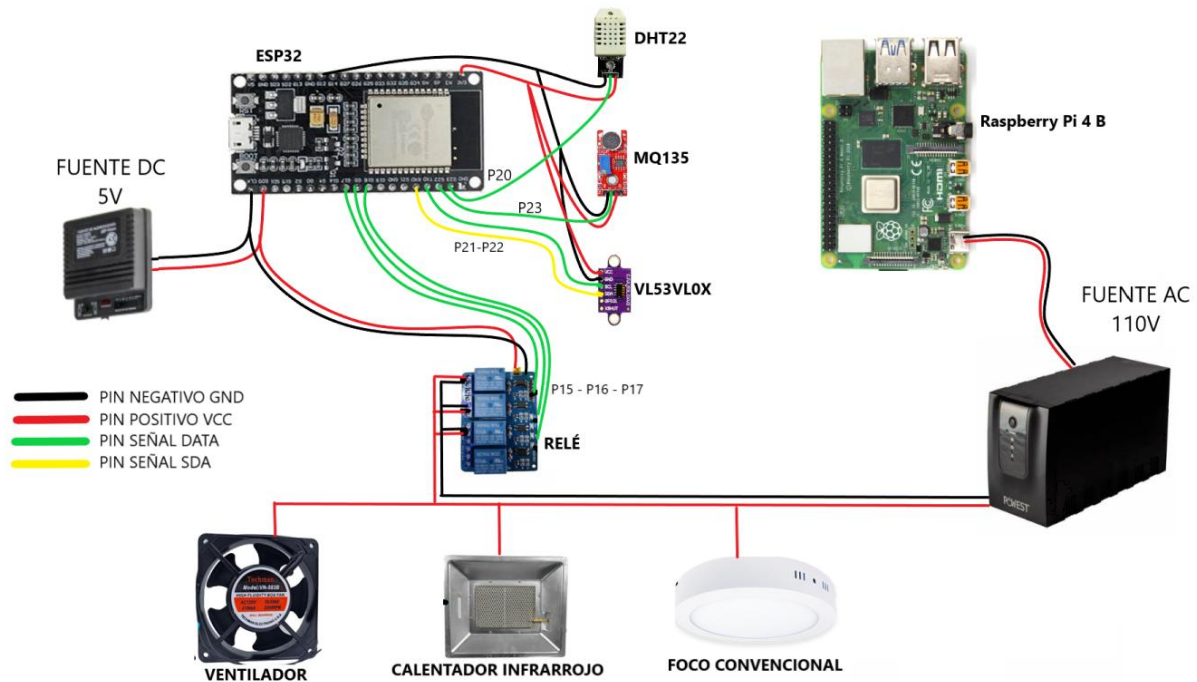
El bloque de alimentación está diseñado para soportar tanto el consumo constante de los dispositivos electrónicos como los picos de energía asociados a los actuadores, asegurando un sistema confiable, seguro y adaptable a diferentes condiciones operativas.

3.4.3.5. Diagrama de Conexión del Sistema Automatizado

En la Figura 28 se observa toda la conexión de cada elemento que conforma el sistema, y se detalla que pines se usan para cada conexión.

Figura 28

Diagrama de Conexión Total del Sistema Automatizado



Fuente: Autoría

3.4.4. Diseño de Software

En esta sección se detalla el diseño del software que permitirá la interacción y el funcionamiento de los distintos componentes del sistema de monitoreo del galpón. El diseño se basa en un enfoque modular que abarca desde la toma de datos por los sensores hasta la visualización y notificación de estos en tiempo real.

El objetivo principal es garantizar que cada módulo del sistema funcione de manera eficiente, cumpliendo con los requerimientos establecidos para la captura, transmisión, procesamiento y almacenamiento de la información. Para ello, se integran herramientas de código abierto y plataformas de comunicación y procesamiento escalables.

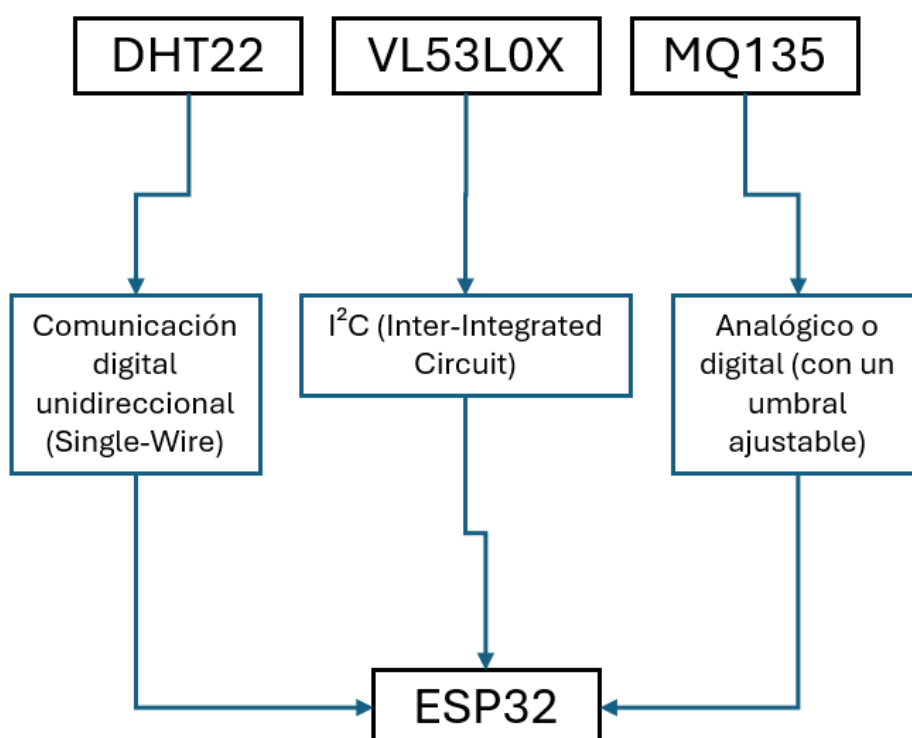
El sistema de monitoreo del galpón de pollos está diseñado en base a bloques funcionales que trabajan de manera conjunta para garantizar el correcto flujo de datos desde los sensores hasta la visualización y notificación.

3.4.4.1. Bloque de Sensores y Captura de Datos

Este bloque está compuesto por sensores que monitorean variables clave como temperatura, humedad, nivel de amoníaco y nivel de alimento en los comederos. Los datos capturados por los sensores son enviados al microcontrolador para su procesamiento inicial. Para la comunicación y toma de datos entre el ESP32 y todos los sensores usan varios protocolos que hacen posible la comunicación, se observa en la Figura 29 los diferentes protocolos usados.

Figura 29

Protocolos usados en el bloque de sensores.



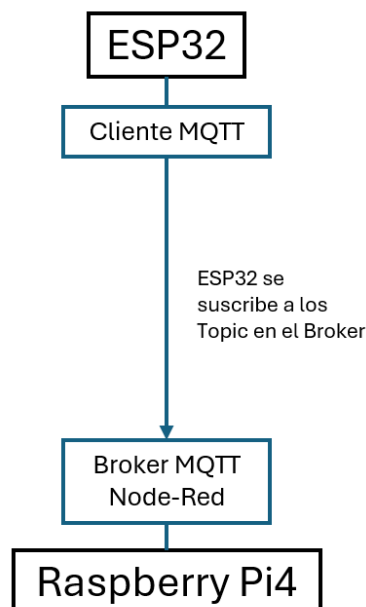
Fuente: Autoría

3.4.4.2. Bloque de Toma de Datos y Comunicación

El microcontrolador ESP32 es el encargado de recibir los datos de los sensores y transformarlos en un formato compatible con el protocolo de comunicación (MQTT), luego de ello se conecta con el servidor Node-Red, para lo cual se usan varios métodos para dicha conexión como se observa en la Figura 30.

Figura 30

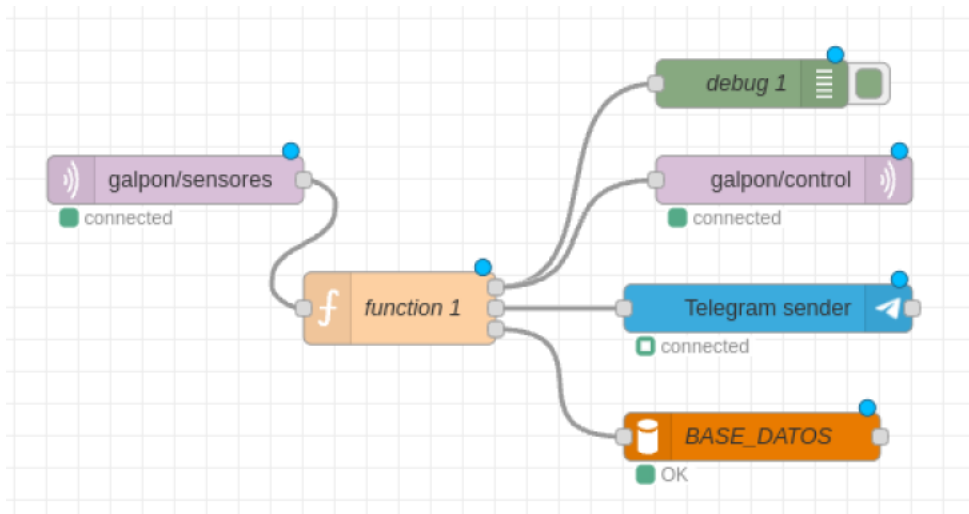
Métodos de comunicación hacia Node-Red



Fuente: Autoría

3.4.4.3. Bloque de Procesamiento

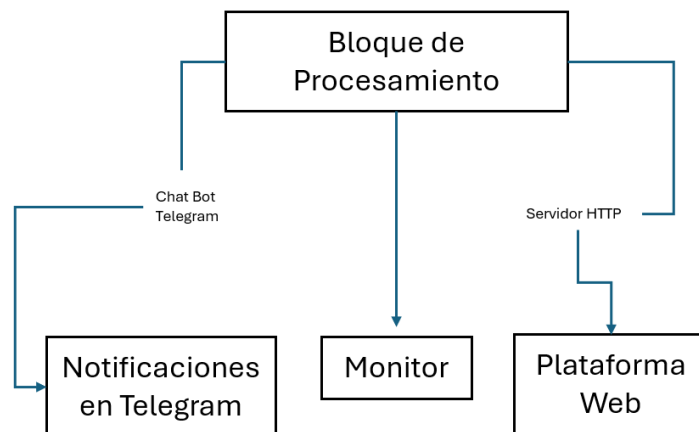
En este bloque, los datos recibidos son procesados por el Raspberry Pi que utiliza herramientas como Node-RED para realizar análisis básicos y preparar la información para su almacenamiento en una base de datos MySQL. Para la conexión con el ESP32 usa el tema sensores/datos y también se configura el puerto y para este caso se usa localhost como se observa en la Figura 31. Este bloque integra un algoritmo de lógica difusa para la toma de decisiones llamado function, donde se implementa toda la lógica difusa, la cual se encarga de procesar toda la información recibida por el ESP32.

Figura 31*Diseño de Flow en Node-Red*

Fuente: Autoría

3.4.4.4. Bloque de Visualización y Notificación

Los datos procesados son presentados a través de una plataforma web accesibles desde dispositivos móviles o computadores y se puede observar todo el estado del sistema automatizado y como se encuentra en ese momento. También se implementan notificaciones automáticas mediante Telegram para alertar sobre condiciones críticas en el galpón, para ello se usa un Chat Bot y un método para poder establecer la conexión como se observa en la Figura 32.

Figura 32*Datos presentados en el monitor de Datos del sistema automatizado.*

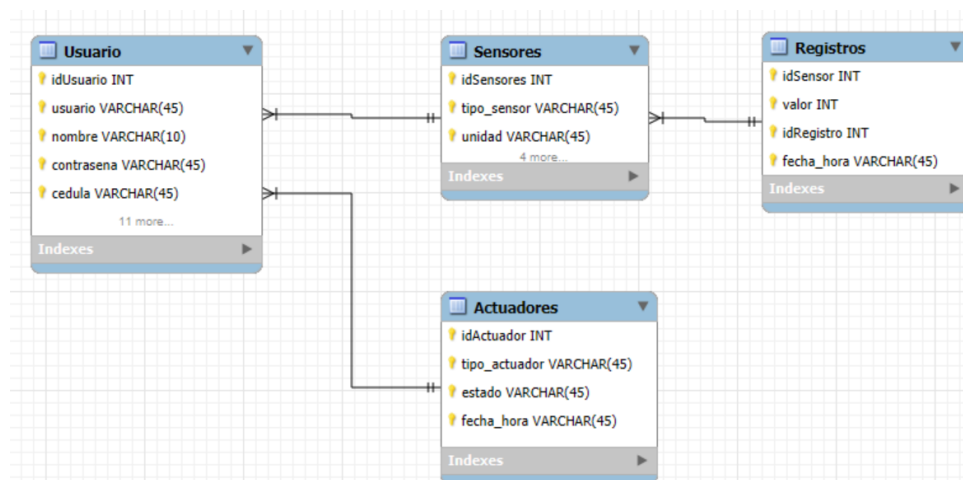
Fuente: Autoría

3.4.4.5. Base de Datos

El propósito de la base de datos utilizada en el sistema de monitoreo del galpón es almacenar información histórica de los sensores y actuadores como se observa en la Figura 33, permitiendo el análisis retrospectivo del comportamiento del sistema y la identificación de patrones a lo largo del tiempo. El modelo relacional diseñado para este sistema de monitoreo y control está basado en una estructura de base de datos con cuatro entidades principales: Usuario, Sensores, Registros y Actuadores, se describe en la Tabla 34.

Figura 33

Base de datos relacional para el sistema automatizado



Fuente: Autoría

Tabla 34

Detalle de los elementos que conforman la base de datos

Tabla	Campo	Tipo de dato	Descripción
usuario	id_usuario	INT	Identificador único del usuario
	usuario	VARCHAR (45)	Nombre de usuario para acceder el sistema
	nombre	VARCHAR (45)	Nombre del usuario
	contrasena	VARCHAR (10)	Contraseña para acceder al sistema

	cedula	VARCHAR (10)	Número de cédula o identificación del usuario
sensores	id_sensores	INT	Identificador único del sensor
	tipo_sensor	VARCHAR (45)	Tipo de sensor
	unidad	VARCHAR (45)	Unidad de medida del sensor
registros	id_sensor	INT	Identificador del sensor al que pertenece el registro
	valor	INT	Valor capturado por el sensor
	id_registro	VARCHAR (45)	Identificador único del sensor
actuadores	fecha_hora	VARCHAR (45)	Fecha y hora en que se realiza el registro
	id_actuador	INT	Identificador único del actuador
	tipo_actuador	VARCHAR (45)	Tipo de actuador
	estado	VARCHAR (45)	Estado del actuador
	fecha_hora	VARCHAR (45)	Fecha y hora en la que se registra el cambio de estado

Fuente Autoría.

La tabla Usuario almacena los datos de los usuarios del sistema, incluyendo su identificador idUsuario, usuario, nombre, contraseña y cédula, así mantiene relaciones con otras tablas, lo que sugiere que los usuarios pueden gestionar sensores y actuadores. La tabla Sensores registra información sobre los dispositivos de medición, identificándolos mediante idSensores e incluyendo datos como el tipo de sensor y la unidad de medida. Los sensores generan lecturas que se almacenan en la tabla Registros, donde cada medición tiene un idRegistro, un valor, un idSensor que indica de qué sensor proviene la lectura y una fecha_hora.

Por otro lado, los actuadores están registrados en la tabla Actuadores, donde cada uno tiene un idActuador único, un tipo_actuador que describe su función, un estado que indica si está activo o inactivo, y un campo de fecha_hora que registra el momento de cambio de estado. En cuanto a las relaciones, un usuario puede estar asociado a múltiples sensores y actuadores, mientras que cada sensor puede generar múltiples registros a lo largo del tiempo.

3.4.4.6. Lógica Difusa

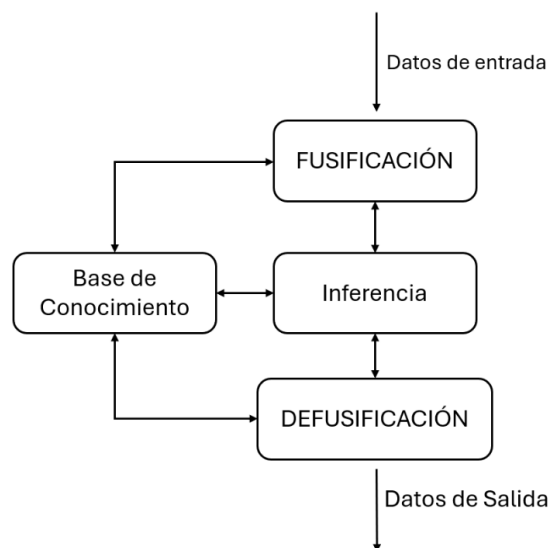
En el sistema de monitoreo, la lógica difusa se implementa dentro del nodo Function, que actúa como el procesador central de las reglas difusas. Este nodo recibe los datos desde los sensores y procesa la información utilizando un conjunto de reglas predefinidas que permiten la toma de decisiones en función de las condiciones ambientales dentro del galpón.

Diseño del Controlador Difuso. El controlador difuso es el núcleo del sistema de regulación ambiental dentro del galpón, permitiendo ajustar la temperatura, humedad y ventilación en función de los valores obtenidos de los sensores como se observa en la Figura 34. Este controlador recibe datos en tiempo real y a través de un conjunto de reglas difusas determina la activación del calentador y el ventilador, asegurando condiciones óptimas para el bienestar de las aves.

El sistema opera a partir de la conversión de datos numéricos provenientes de los sensores a valores lingüísticos, que son procesados por un conjunto de reglas. Posteriormente, estos valores son convertidos nuevamente a valores numéricos mediante un proceso de defusificación, permitiendo activar los actuadores con niveles adecuados.

Figura 34

Diagrama de Bloques del Sistema Difuso



Fuente: Autoría

Variables del Universo Discurso. Las variables involucradas en el sistema de monitoreo se dividen en variables de entrada, que representan los datos sensoriales, y variables de salida, que controlan los actuadores.

Tabla 35

Identificación de Variables del Universo Discurso

Tipo	Variable
Entrada	Temperatura (°C)
	Humedad (%)
	Amoniaco (ppm)

Fuente: Autoría

En la Tabla 35 se puede observar las variables de entrada y salida, están conformadas por los parámetros ambientales y de calidad del aire, obtenidos en tiempo real a través de los sensores instalados en el galpón.

Las variables de entrada monitorean condiciones clave en el galpón: la temperatura ambiental y la humedad, las cuales afectan el bienestar de las aves.

Variables Lingüísticas en el Sistema Difuso. Las variables lingüísticas permiten transformar valores numéricos en términos comprensibles para la toma de decisiones automatizada. En el sistema de monitoreo del galpón, estas variables se observan en la Tabla 36.

Tabla 36

Variables Lingüísticas de entrada y salida

Variable	Magnitud	Rango
Temperatura	Grados Centígrados °C	0-40

Humedad	Porcentaje %	0-100
Amoníaco	Partes Por Millón ppm	0-100

Fuente: Autoría

Esta tabla resume las variables utilizadas en el sistema difuso, permitiendo una mejor interpretación de los datos ambientales y optimizando la activación de los actuadores según las condiciones detectadas.

Distribución Borrosa en el Sistema Difuso. La distribución borrosa define cómo se agrupan los valores numéricos dentro de los conjuntos difusos, permitiendo una transición suave entre los diferentes estados lingüísticos como se observa en la Tabla 37. En este sistema de monitoreo del galpón, cada variable del universo de discurso es representada mediante funciones de pertenencia que asignan un grado de pertenencia a cada valor dentro de su respectivo conjunto borroso.

Tabla 37

Distribución Borrosa de las Variables del Sistema

Variable	Rango	Conjuntos Difusos	Rango de cada conjunto
Temperatura	0-40	Baja (TB)	0 - 15
		Media (TM)	16 – 25
		Alta (TA)	26 – 40
Humedad	0-100	Baja (HB)	0 – 40
		Media (HM)	41 – 69
		Alta (HA)	70 – 100
Amoniaco	0-100	Bajo (AB)	0 – 30
		Moderado (AM)	31 – 59
		Alto (AA)	60 – 100

Fuente: Autoría

Esta tabla resume la distribución borrosa utilizada en el sistema, estableciendo los rangos en los que cada variable es interpretada dentro de los conjuntos lingüísticos. Permite visualizar la relación entre los datos medidos y las acciones que se tomarán para regular el ambiente del galpón, estos niveles de medición de cada parámetro de basa en la crianza de pollos que se usa normalmente se usan para galpones de crianza de pollos Broiler (Sánchez, 2021).

Funciones de Pertenencia. Las funciones de pertenencia son fundamentales en un sistema difuso, ya que determinan la manera en que un valor numérico de entrada se asigna a un conjunto borroso. Estas funciones permiten representar la incertidumbre y la transición gradual entre diferentes estados lingüísticos. La elección adecuada de estas funciones influye en la precisión y estabilidad del sistema de control del galpón.

Para este sistema, se han seleccionado funciones triangulares y trapezoidales, debido a su facilidad de implementación y eficiencia en el procesamiento en tiempo real, en la Tabla 38 se observa la clasificación.

- Función Triangular: Se utiliza para representar transiciones suaves entre valores medios de una variable.
- Función Trapezoidal: Se emplea para definir valores extremos (bajos y altos), asegurando estabilidad en los bordes del rango de medición.

Estas funciones han sido seleccionadas porque ofrecen buena precisión y estabilidad computacional, facilitando el procesamiento en Node-RED y en los cálculos realizados en el ESP32.

Tabla 38

Funciones de Pertenencia en el Sistema Difuso

Variable	Estado Bajo	Estado Medio	Estado Alto
----------	-------------	--------------	-------------

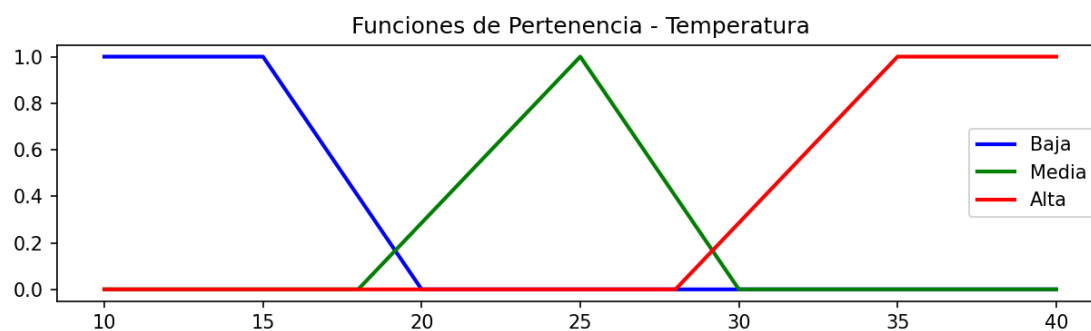
Temperatura	Trapezoidal (<16)	Triangular (16-25)	Trapezoidal (>25)
Humedad	Trapezoidal (<50)	Triangular (50-70)	Trapezoidal (>70)
Amoníaco	Trapezoidal (<30)	Triangular (30-70)	Trapezoidal (>70)

Fuente: Autoría

Se eligieron funciones trapezoidales en los extremos para reducir la sensibilidad a fluctuaciones rápidas en los sensores, asegurando estabilidad en la activación de los actuadores. Se emplearon funciones triangulares en los valores intermedios para permitir transiciones suaves y progresivas, evitando cambios bruscos en el sistema de control.

Figura 35

Gráfica de función de pertenencia en temperatura

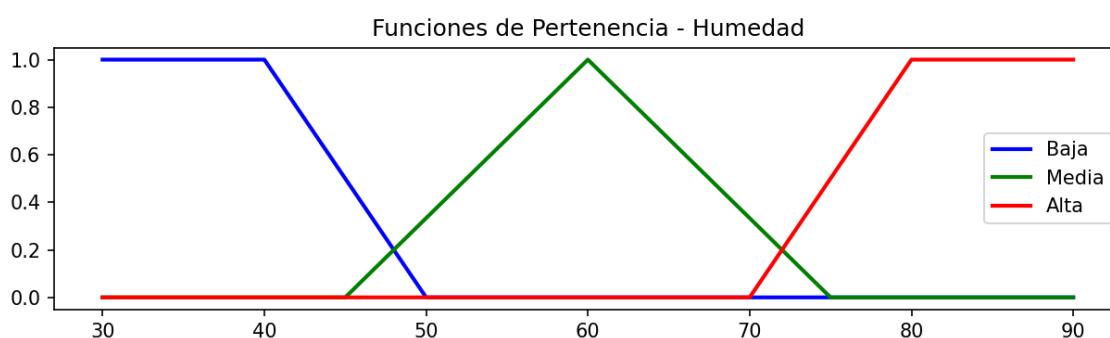


Fuente: Autoría

Interpretación: Si la temperatura es menor a 16°C, se considera fría. Si está entre 16°C y 25°C, es aceptable, y si supera los 25°C, se considera alta, se puede observar la Figura 35.

Figura 36

Gráfica de función de pertenencia en humedad



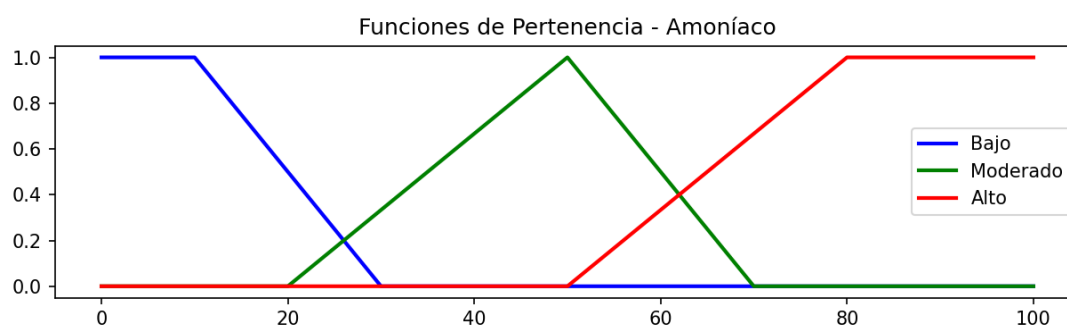
/

Fuente: Autoría

Interpretación: En la Figura 36 se interpreta que, si la humedad es menor a 40%, se considera baja, lo que podría indicar un ambiente seco. Si está entre 40% y 70%, es óptima, y si supera el 70%, el ambiente se vuelve muy húmedo.

Figura 37

Gráfica de función de pertenencia en Amoníaco



Fuente: Autoría

Interpretación: Si el nivel de NH₃ es bajo, no hay problema. Si es moderado, puede afectar la salud de los pollos. Si es alto, se activa la ventilación para reducir la contaminación, en la Figura 37 se observa su gráfica.

Reglas Borrosas para el Sistema de Control Difuso. Las reglas borrosas establecen la relación entre las variables de entrada y las acciones que deben tomarse sobre las salidas del sistema. Estas reglas están basadas en el conocimiento del dominio y permiten que el sistema responda de manera gradual a diferentes condiciones ambientales como se observa en la Tabla 39.

Las reglas borrosas se generan en base a las combinaciones de los valores de las variables de entrada: temperatura, humedad y amoníaco, afectando a los actuadores ventilador, calentador.

Tabla 39

Reglas diseñadas para monitorear el sistema automatizado.

Nro.	Temperatura	Humedad	Amoníaco	Ventilador (0-3)	Calentador (0-1)
1	TA	HB	AB	V0	C1
2	TB	HB	AM	V1	C1
3	TB	HB	AA	V2	C1
4	TB	HM	AB	V0	C1
5	TB	HM	AM	V1	C1
6	TB	HM	AA	V2	C1
7	TB	HA	AB	V1	C0
8	TB	HA	AM	V2	C0
9	TB	HA	AA	V3	C0
10	TM	HB	AB	V0	C0
11	TM	HB	AM	V1	C0
12	TM	HB	AA	V2	C0
13	TM	HM	AB	V0	C0
14	TM	HM	AM	V1	C0
15	TM	HM	AA	V2	C0
16	TM	HA	AB	V1	C0
17	TM	HA	AM	V2	C0
18	TM	HA	AA	V3	C0
19	TA	HB	AB	V1	C0
20	TA	HB	AM	V2	C0
21	TA	HB	AA	V3	C0
22	TA	HM	AB	V2	C0
23	TA	HM	AM	V2	C0
24	TA	HM	AA	V3	C0
25	TA	HA	AB	V2	C0
26	TA	HA	AM	V3	C0
27	TA	HA	AA	V3	C0

Fuente: Autoría

Agregación para el sistema de control Difuso. La agregación en el sistema de lógica difusa es el proceso mediante el cual se combinan los resultados de todas las reglas difusas para obtener una única salida por cada actuador, como el ventilador y el calentador. Este proceso se basa en la evaluación de las condiciones ambientales del galpón, considerando variables como la temperatura, la humedad, el nivel de amoníaco.

Se analizan todas las reglas activas, asignando un grado de pertenencia a las salidas correspondientes en función de los valores de entrada. Posteriormente, se combinan estos valores, seleccionando el nivel más alto de activación para cada actuador, garantizando que la respuesta del sistema sea la más adecuada según las condiciones detectadas. Finalmente, se genera una única función de salida por cada actuador, integrando todas las contribuciones de las reglas activadas y asegurando un control eficiente del ambiente en el galpón.

La agregación es un paso clave en el sistema de lógica difusa, ya que permite consolidar todas las reglas activadas en una sola salida para cada actuador, asegurando que el sistema responda de manera adecuada a las condiciones ambientales en el galpón.

Defusificación. Es el proceso final dentro del sistema de lógica difusa, en el cual los resultados obtenidos tras la agregación de las reglas difusas se convierten en valores concretos y utilizables para la activación de los actuadores. En este sistema, la defusificación permite determinar la intensidad exacta con la que deben operar el ventilador, el calentador y los focos, asegurando un control preciso del ambiente dentro del galpón de pollos.

Para ello, se toma la función de salida obtenida en la etapa de agregación y se aplica un método de defusificación, generalmente el de "Centroide" o "Centro de Gravedad", que calcula un valor numérico representativo basado en la distribución de pertenencia de la salida difusa. Este valor es el que finalmente se emplea para establecer el nivel de activación de cada actuador, garantizando una respuesta óptima ante las condiciones detectadas en el entorno.

Capítulo IV: Implementación y Pruebas de Funcionamiento

En este capítulo se presentan las pruebas realizadas para validar el correcto funcionamiento del sistema de monitoreo y control implementado en el galpón de pollos. Para garantizar que cada componente del sistema opera de manera eficiente y cumple con los requerimientos establecidos, se han desarrollado pruebas específicas para cada uno de los bloques que lo conforman.

Las pruebas se han dividido en diferentes secciones, ya que se usa un plan de pruebas (ANEXO C), evaluando tanto el hardware como el software. En el caso del hardware, se verificará el desempeño de los sensores, la transmisión de datos, la respuesta de los actuadores y la eficiencia del sistema de alimentación eléctrica. En cuanto al software, se realizaron pruebas de procesamiento de datos, ejecución de la lógica difusa y la correcta comunicación entre los dispositivos.

El objetivo de este capítulo es demostrar que el sistema es capaz de monitorear las variables ambientales del galpón en tiempo real, tomar decisiones basadas en la lógica difusa y activar los actuadores de manera eficiente para mantener un ambiente óptimo para la producción avícola.

4.1 Implementación de Hardware del Sistema Automatizado

La implementación de hardware en el galpón se centra en la instalación física de los componentes que conforman el sistema de monitoreo y control. Esto incluye la correcta ubicación y montaje de los sensores, el módulo de procesamiento (Raspberry Pi 4), los actuadores y el sistema de alimentación eléctrica, asegurando que cada elemento funcione de manera óptima dentro del entorno del galpón, la estructura del galpón está hecha de madera con recubrimiento de malla y plástico, tiene una dimensión de 2m x 2,25m, así como una altura de 2.20m y tiene una capacidad para 50 pollos, considerando que la producción de la empresa

AVICAMP es de 500 pollos, es importante considerar que al ser un prototipo se debe dimensionar la capacidad original. En la Figura 38 se puede observar la estructura del galpón.

Figura 38

Estructura del galpón



Fuente: Autoría

El objetivo de esta etapa es garantizar que los dispositivos estén ubicados dentro del galpón para obtener mediciones precisas y eficientes, minimizar interferencias y optimizar la respuesta del sistema.

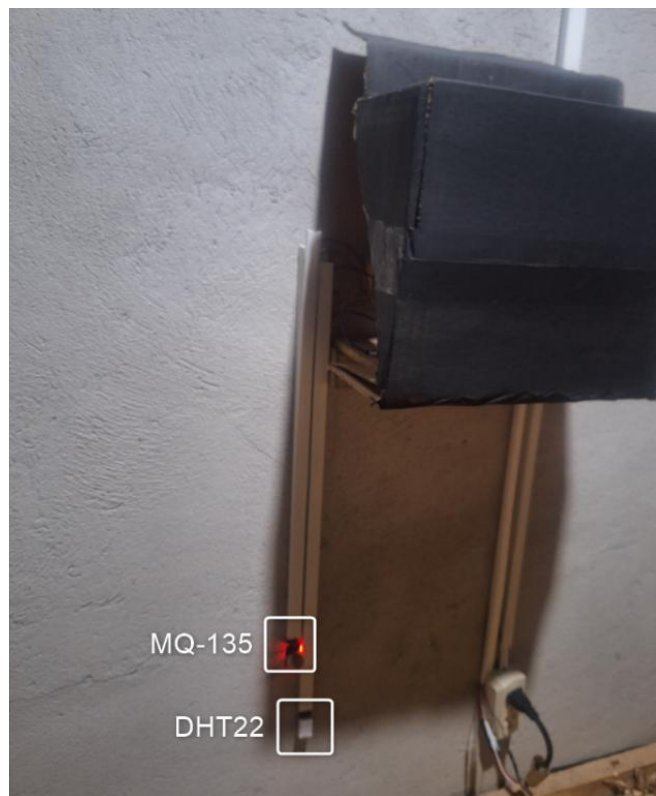
4.1.1. Bloque de Sensores

La instalación del bloque de sensores en el galpón como se observa en la Figura 39, se realizó considerando factores clave como la ubicación, la protección de los dispositivos y la

correcta distribución de los cables de alimentación y comunicación. Este bloque está compuesto por los sensores DHT22 (temperatura y humedad), MQ-135 (amoníaco) y VL53L0X (nivel de alimento) como se observa en la Figura 40, los cuales deben instalarse en puntos específicos para garantizar mediciones precisas y representativas del ambiente.

Figura 39

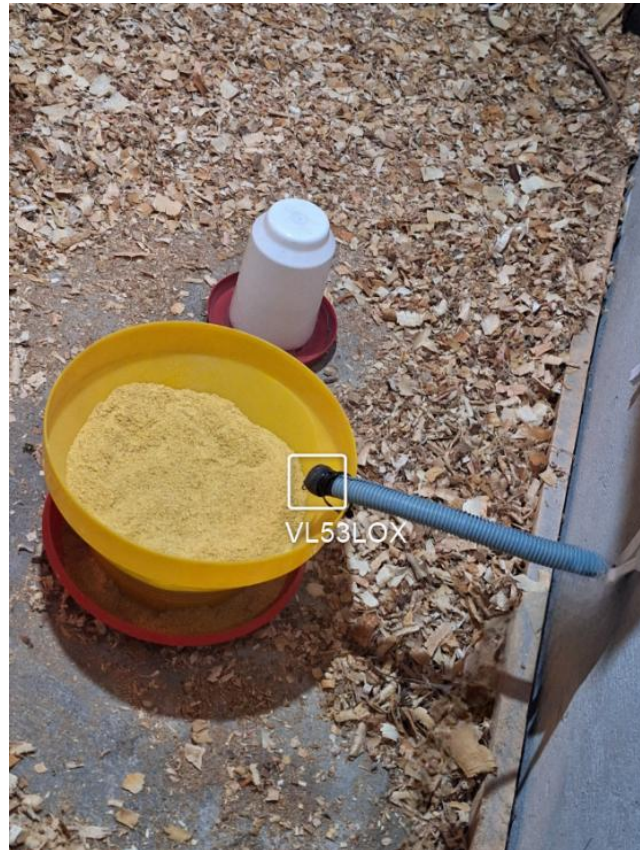
Instalación de DHT22 y MQ135 dentro del galpón



Fuente: Autoría

Figura 40

Instalación del sensor VL53L0X en el comedero



Fuente: Autoría

4.1.2. Bloque de Toma de Datos y Procesamiento

El bloque de toma de datos está compuesto por el ESP32 como se observa en la Figura 41, el cual recibe la información de los sensores instalados en el galpón y la transmite al sistema de procesamiento. La correcta instalación de este módulo es crucial para garantizar la estabilidad de la comunicación y la correcta recolección de datos en tiempo real.

El bloque de procesamiento como se observa en la Figura 42, representado por la Raspberry Pi 4, es el encargado de recibir, analizar y gestionar la información obtenida de los sensores a través del ESP32. Su instalación dentro del galpón debe garantizar un funcionamiento estable, protegido y con una conexión eficiente a la red de comunicación del sistema.

Figura 41*Instalación de Bloque de Toma de Datos*

Fuente: Autoría

Figura 42*Instalación de Bloque de Procesamiento*

Fuente: Autoría

El bloque de procesamiento garantiza que el sistema funcione de manera eficiente, permitiendo la toma de decisiones automatizada basada en la lógica difusa y el almacenamiento seguro de los datos capturados en el galpón.

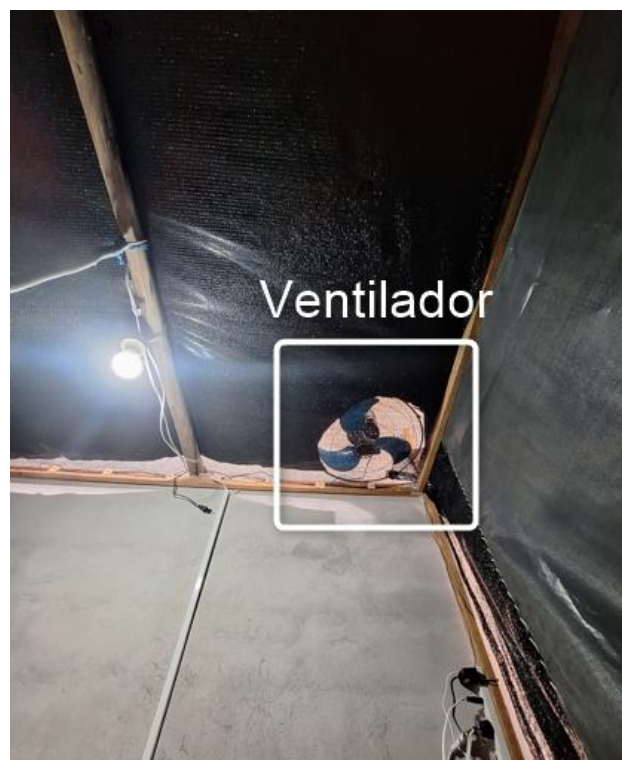
La correcta implementación del bloque de toma de datos asegura una transmisión eficiente y continua de la información capturada por los sensores, permitiendo un monitoreo confiable del ambiente dentro del galpón y facilitando la toma de decisiones automatizada en el sistema de control.

4.1.3. Bloque de Actuadores

El bloque de actuadores es el encargado de regular las condiciones ambientales del galpón en función de los datos capturados por los sensores y procesados por la Raspberry Pi. Este bloque está compuesto por un calentador, un ventilador y un foco convencional para iluminación, los cuales deben ser instalados estratégicamente para maximizar su efectividad, en las Figuras 43, 44 y 45 se observa la instalación de todos los actuadores.

Figura 43

Instalación del ventilador en el galpón



Fuente: Autoría

Figura 44

Instalación del calentador en el galpón



Fuente: Autoría

Figura 45

Instalación del foco para iluminación del galpón



Fuente: Autoría

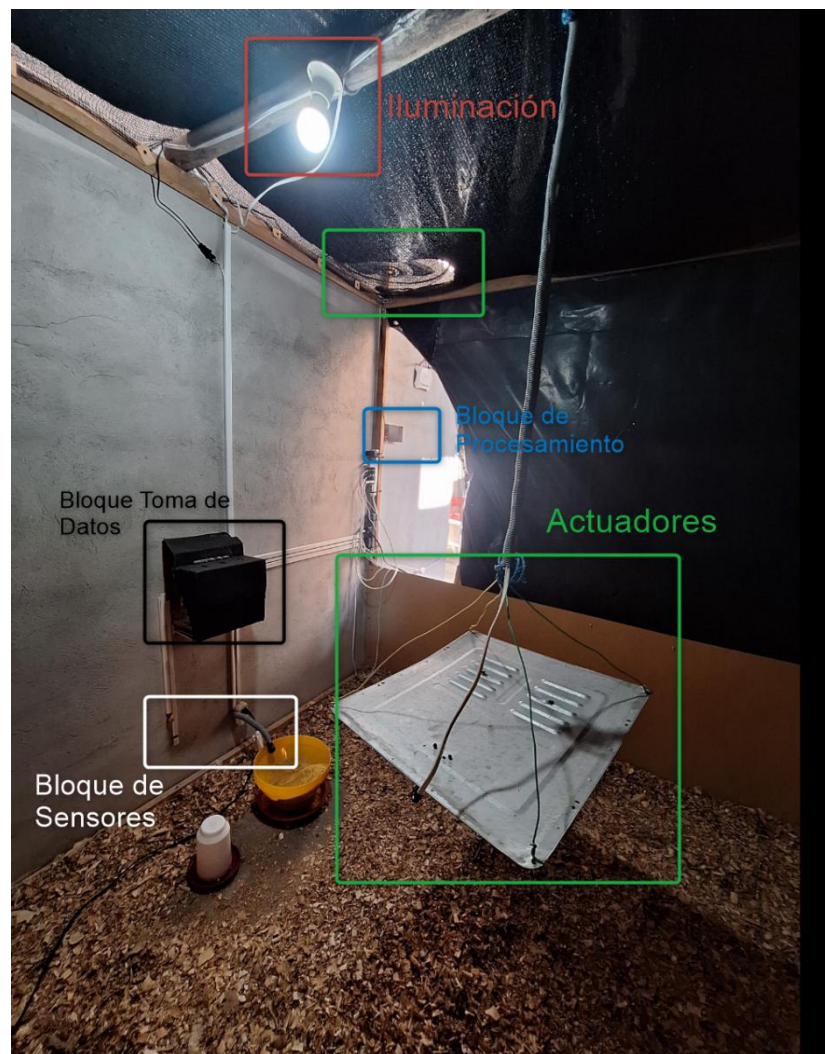
La ubicación de los sensores DHT22 y MQ135 se definió estratégicamente en una posición elevada, justo debajo del módulo de control principal, dentro del galpón cerrado de

dimensiones reducidas ($2.20 \times 2.20 \times 2.20$ m). Esta elección se fundamenta en la homogeneidad del ambiente interno, donde la circulación de aire es mínima y no existen fuentes de calor o ventilación que generen microclimas. Al estar alejados del suelo cubierto de viruta y de la lámpara calefactora, ambos sensores pueden registrar datos representativos del entorno sin interferencias directas. Además, al compartir ubicación se simplifica el cableado y mantenimiento del sistema, sin comprometer la precisión necesaria para el monitoreo ambiental de un espacio controlado.

El sistema conectado con todas sus ubicaciones se puede observar en la Figura 46.

Figura 46

Visión general del Sistema Automatizado



Fuente: Autoría

4.2. Implementación de software del Sistema Automatizado

La implementación del software del sistema automatizado representa una de las fases clave del proyecto, ya que permite establecer la lógica de funcionamiento y el control de los elementos involucrados en el monitoreo ambiental del galpón. Este software fue desarrollado utilizando el microcontrolador ESP32 programado en el entorno Arduino IDE, integrando diversos sensores que permiten monitorear variables críticas como temperatura, humedad, nivel de amoníaco y nivel de alimento en el comedero, así como también un raspberry Pi4 para el procesamiento.

4.2.1. Bloque de sensores y Toma de datos

Para el correcto funcionamiento del sistema de monitoreo en el ESP32, fue necesario importar diversas librerías específicas para la lectura de sensores y conexión de red como se observa en la Figura 47. Se utilizó la librería DHT.h para el sensor de temperatura y humedad DHT22, MQ135.h para el sensor de calidad del aire, y VL53L0X.h para el sensor de distancia láser. Además, se incorporaron las librerías WiFi.h y Wire.h, que permiten la conectividad del ESP32 a la red inalámbrica y la comunicación I2C requerida por el sensor VL53L0X, respectivamente.

Figura 47

Librerías usadas para el bloque de Toma de Datos

```
// ==== LIBRERÍAS ====
#include <WiFi.h>           // Para conexión WiFi
#include <DHT.h>             // Sensor DHT22
#include <Wire.h>            // Comunicación I2C
#include <VL53L0X.h>         // Sensor de distancia
#include <MQ135.h>           // Sensor de calidad del aire MQ135

// ==== DEFINICIÓN DE PINES ====
#define DHTPIN 4            // Pin de datos del DHT22
#define DHTTYPE DHT22      // Tipo de sensor DHT

#define MQ135_PIN 34        // Pin analógico para el MQ135
#define VENTILADOR_PIN 26   // Pin salida para ventilador
#define CALENTADOR_PIN 27   // Pin salida para calentador
```

Fuente: Autoría

En esta etapa del desarrollo se definieron los pines a los que se conectaron físicamente los sensores y actuadores como se observa en la Figura 48. El pin GPIO4 fue asignado al sensor DHT22, el pin analógico GPIO34 al sensor MQ135 y el sensor VL53L0X fue conectado mediante la interfaz I2C. Se asignan los pines GPIO26 y GPIO27 como salidas digitales destinadas al control del ventilador y el calentador. Posteriormente, se crearon instancias de los objetos necesarios para la lectura de cada uno de los sensores.

Figura 48

Definición de pines y objetos de sensor

```
// ==== DEFINICIÓN DE PINES ====
#define DHTPIN 4           // Pin de datos del DHT22
#define DHTTYPE DHT22     // Tipo de sensor DHT

#define MQ135_PIN 34       // Pin analógico para el MQ135
#define VENTILADOR_PIN 26 // Pin salida para ventilador
#define CALENTADOR_PIN 27 // Pin salida para calentador

// ==== OBJETOS DE SENSOR ====
DHT dht(DHTPIN, DHTTYPE);
VL53L0X sensorDistancia;
MQ135 mq135_sensor(MQ135_PIN);
```

Fuente: Autoría

Durante la función `setup()`, se inicializaron todos los sensores y se configuraron los pines de salida como se observa en la Figura 49. El sensor DHT22 fue activado con su respectiva función `begin()`. La comunicación I2C se habilitó mediante `Wire.begin()` para permitir la inicialización del sensor de distancia VL53L0X. Este último fue configurado para realizar mediciones continuas. También se establecieron los pines de salida para los actuadores, aunque en esta fase del código solo se realizó la lectura de los sensores, sin activar aún las salidas.

Figura 49

Inicialización de sensores en setup ()

```
void setup() {
  Serial.begin(115200);

  // Configuración de pines
  pinMode(MQ135PIN, INPUT);
  pinMode(LEDPIN, OUTPUT);
  digitalWrite(LEDPIN, LOW);

  // Inicializar sensores
  Serial.println("Inicializando sensores...");

  // Inicializar DHT11
  dht.begin();
  delay(1000); // Tiempo para estabilizar el sensor DHT11

  // Inicializar VL53L0X
  if (!lox.begin()) {
    Serial.println("Error al iniciar el sensor VL53L0X. Verifique las conexiones.");
    while (1); // Detener si no se inicializa correctamente
  }

  Serial.println("Sistema de monitoreo con DHT22, MQ135 y VL53L0X iniciado");
}
```

Fuente: Autoría

Dentro del ciclo principal (loop()), se realizó la lectura periódica de los sensores conectados. La temperatura y humedad se obtuvieron del sensor DHT22, mientras que el valor estimado de amoníaco fue obtenido del MQ135. También se midió la distancia al comedero mediante el sensor VL53L0X. Todos los valores recolectados fueron enviados al Monitor Serial del IDE de Arduino, permitiendo verificar su funcionamiento correcto en tiempo real, se observa en la Figura 50.

Figura 50

Lectura de sensores y monitoreo serial en loop()

```
void loop() {
  // --- Sensor DHT22 ---
  float temperature = dht.readTemperature();
  if (isnan(temperature)) {
    Serial.println("Error al leer el sensor DHT22.");
  } else {
    Serial.print("Temperatura: ");
    Serial.print(temperature);
    Serial.println(" °C");
    if (temperature > 35.0 || temperature < 25.0) {
      alert = true;
      Serial.println("Alerta: Temperatura fuera de rango.");
    }
  }

  // --- Sensor MQ135 ---
  int sensorValue = analogRead(MQ135PIN);
  float voltage = sensorValue * (3.3 / 4095.0); // Convertir ADC a voltaje
  float ppm = (voltage - 0.1) * 100; // Escala aproximada para CO2
  Serial.print("Nivel de CO2 (estimado): ");
  Serial.print(ppm);
  Serial.println(" ppm");
  if (ppm > 1000) {
    alert = true;
    Serial.println("Alerta: Calidad del aire deficiente.");
  }

  // --- Sensor VL53L0X ---
  VL53L0X_RangingMeasurementData_t measure;
  lox.rangingTest(&measure, false);
  if (measure.RangeStatus != 4) {
    Serial.print("Distancia medida: ");
    Serial.print(measure.RangeMilliMeter);
  }
}
```

Fuente: Autoría

4.2.2. Bloque de Procesamiento

El bloque de procesamiento representa la unidad encargada de interpretar los datos obtenidos por los sensores del sistema instalado en el galpón. Para este propósito se integró un servidor local basado en una Raspberry Pi 4, sobre la cual se desplegaron servicios como un broker MQTT (Mosquitto) y la plataforma Node-RED. Esta infraestructura permite recibir, procesar y responder en tiempo real a las condiciones del ambiente mediante una lógica difusa predefinida.

4.2.2.1 Implementación de Servidor MQTT

Para la comunicación entre el microcontrolador ESP32 y el sistema de procesamiento local, se implementó un servidor MQTT utilizando el broker Mosquitto en una Raspberry Pi 4. Este broker permite una arquitectura de tipo publish-subscribe, ideal para sistemas IoT. La instalación se realizó mediante el comando `sudo apt install mosquitto mosquitto-clients` como se observa en la Figura 51. Una vez instalado, el servicio fue configurado para iniciarse automáticamente, permitiendo recibir y distribuir mensajes en tiempo real entre dispositivos conectados en la misma red local.

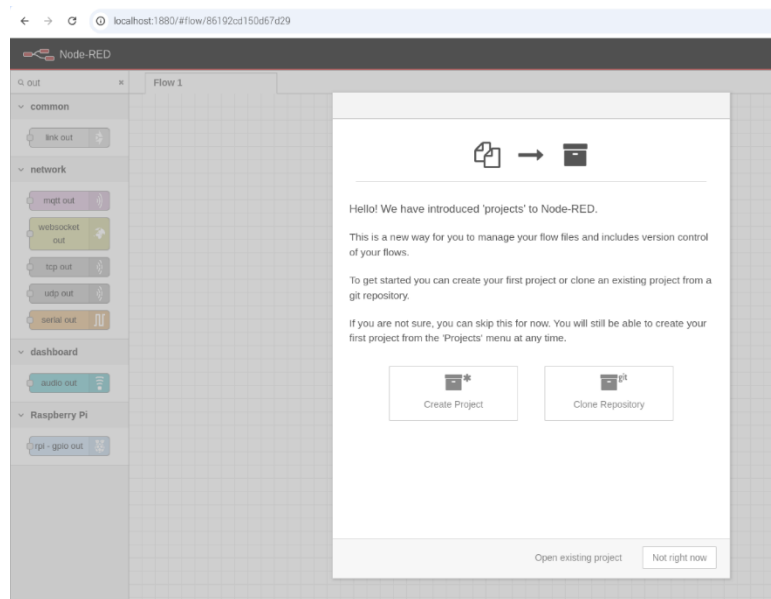
Figura 51

Instalación y configuración del servidor MQTT (Mosquitto)

```
santiago@raspberrypi:~ $ sudo apt install mosquitto mosquitto-clients
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
mosquitto ya está en su versión más reciente (2.0.11-1.2+deb12u1).
mosquitto-clients ya está en su versión más reciente (2.0.11-1.2+deb12u1).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 176 no actualizados.
```

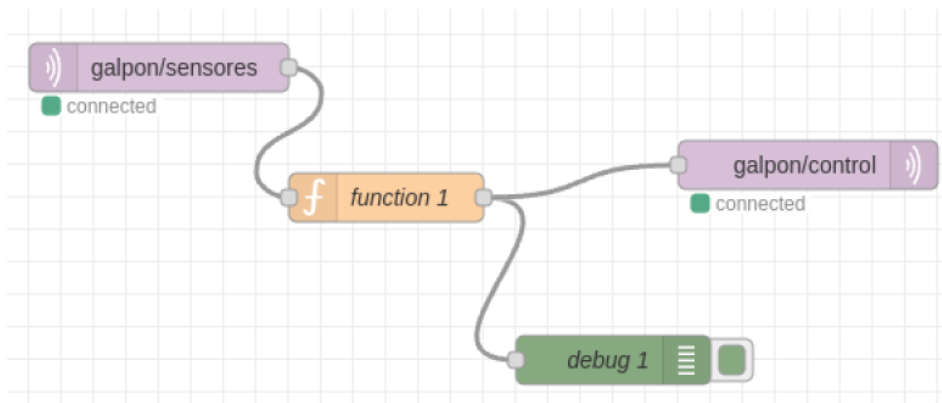
Fuente: Autoría

Posteriormente, se instaló el entorno de desarrollo Node-RED en la misma Raspberry Pi como se observa en la Figura 52. Este entorno de programación visual fue elegido por su facilidad para procesar mensajes MQTT y aplicar lógica personalizada.

Figura 52*Instalación de Node-RED*

Fuente: Autoría

Dentro de Node-RED, se construyó un flujo compuesto por los siguientes nodos, un nodo mqtt in configurado para escuchar el tópico galpon/sensores, un nodo function, encargado de analizar los datos recibidos y aplicar la lógica difusa basada en 27 reglas predefinida y un nodo mqtt out que envía la respuesta de la lógica hacia el tópico galpon/control, al cual el ESP32 está suscrito, el esquema se puede observar en la Figura 53.

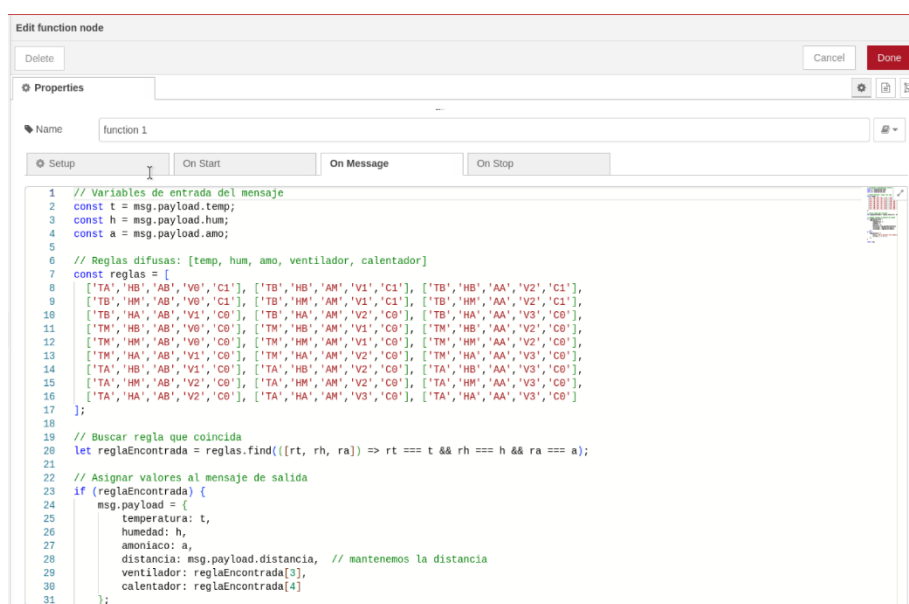
Figura 53*Creación del flujo en Node-RED*

Fuente: Autoría

La lógica de control fue implementada mediante un nodo Function como se observa en la Figura 54, que evalúa la combinación de los valores clasificados (TA, TM, TB, etc.) y determina el estado del ventilador (V0 a V3) y del calentador (C0, C1). Esta lógica permite adaptar el comportamiento del sistema en función de las condiciones ambientales del galpón.

Figura 54

Implementación de la lógica difusa



Fuente: Autoría

Finalmente, el ESP32 recibe el mensaje procesado en formato JSON a través del tópico galpon/control. Este mensaje contiene el estado esperado de los actuadores, los cuales son interpretados para activar o desactivar los respectivos relés que controlan el ventilador y el calentador. Esta comunicación bidireccional establece un sistema completamente funcional de monitoreo y control automático, el código implementado se observa en la Figura 55.

Figura 55

Comunicación bidireccional

```

void callback(char* topic, byte* payload, unsigned int length) {
    payload[length] = '\0'; // Asegurar fin de cadena
    String mensaje = String((char*)payload);

    Serial.print("Mensaje recibido: ");
    Serial.println(mensaje);

    DynamicJsonDocument doc(128);
    DeserializationError error = deserializeJson(doc, mensaje);

    if (error) {
        Serial.print("❌ Error al parsear JSON: ");
        Serial.println(error.c_str());
        return;
    }

    String v = doc["ventilador"];
    String c = doc["calentador"];

    if (v == "V0") digitalWrite(VENTILADOR_PIN, LOW);
    else digitalWrite(VENTILADOR_PIN, HIGH);

    if (c == "C1") digitalWrite(CALENTADOR_PIN, HIGH);
    else digitalWrite(CALENTADOR_PIN, LOW);

    Serial.print("Ventilador: "); Serial.println(v);
    Serial.print("Calentador: "); Serial.println(c);
}

```

Fuente: Autoría

4.2.2.2 Implementación de Plataforma Web y Notificaciones

Para la implementación del sistema de monitoreo y control del galpón inteligente, se desarrolló una plataforma web accesible desde cualquier lugar, construida sobre una arquitectura de servidor local basada en tecnologías de código abierto. A continuación, se describe el proceso detallado llevado a cabo.

Para el despliegue de la aplicación web, se empleó el servidor HTTP Apache2, elegido por su estabilidad y compatibilidad con sistemas Linux embebidos, como se observa en la Figura 56. Apache se configuró para servir los archivos HTML y PHP de la plataforma, permitiendo el acceso local desde cualquier navegador.

El estado del servidor puede verificarse mediante el siguiente comando:

Figura 56

Implementación del servidor web Apache

```

• apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-05-12 21:14:35 -05; 21h ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 641 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Process: 2126 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
    Main PID: 832 (apache2)
      Tasks: 10 (limit: 3915)
         CPU: 4.788s
    CGroup: /system.slice/apache2.service
            └─ 832 /usr/sbin/apache2 -k start
               2131 /usr/sbin/apache2 -k start
               2132 /usr/sbin/apache2 -k start
               2134 /usr/sbin/apache2 -k start
               2136 /usr/sbin/apache2 -k start
               2138 /usr/sbin/apache2 -k start
               2522 /usr/sbin/apache2 -k start
               3251 /usr/sbin/apache2 -k start
               3252 /usr/sbin/apache2 -k start
               3754 /usr/sbin/apache2 -k start

may 13 15:01:08 raspberrypi sudo[2663]: pam_unix(sudo:session): session closed for user root
may 13 15:07:04 raspberrypi sudo[2748]: www-data : PWD=/var/www/html ; USER=root ; COMMAND=/bin/systemctl restart
may 13 15:07:04 raspberrypi sudo[2748]: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=33)
may 13 15:07:07 raspberrypi sudo[2748]: pam_unix(sudo:session): session closed for user root
may 13 16:28:10 raspberrypi sudo[3541]: www-data : PWD=/var/www/html ; USER=root ; COMMAND=/bin/systemctl restart
may 13 16:28:10 raspberrypi sudo[3541]: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=33)
may 13 16:28:11 raspberrypi sudo[3541]: pam_unix(sudo:session): session closed for user root
may 13 16:31:12 raspberrypi sudo[3606]: www-data : PWD=/var/www/html ; USER=root ; COMMAND=/bin/systemctl restart
lines 1-29

```

Fuente: Autoría

Se integró el sistema gestor de bases de datos MariaDB para el almacenamiento estructurado de los registros provenientes de los sensores, como se observa en la Figura 57. El lenguaje PHP se utilizó como capa intermedia entre la interfaz gráfica y la base de datos, permitiendo realizar consultas, inserciones y actualizaciones en tiempo real.

Figura 57

Integración de MariaDB y PHP

```

santiago@raspberrypi:~$ sudo systemctl status mariadb
• mariadb.service - MariaDB 10.11.6 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-05-12 21:14:38 -05; 21h ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
  Process: 643 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysql
  Process: 670 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITI
  Process: 700 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= || V
  Process: 1054 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSI
  Process: 1056 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
    Main PID: 791 (mariabdd)
      Status: "Taking your SQL requests now..."
      Tasks: 15 (limit: 3915)
         CPU: 6.464s
    CGroup: /system.slice/mariadb.service
            └─ 791 /usr/sbin/mariabdd

```

Fuente: Autoría

Dado que el proyecto se ejecuta en una red local, se utilizó la herramienta Ngrok para exponer temporalmente el servidor web a través de una dirección pública accesible desde cualquier red, como se observa en la Figura 58. Esto permitió realizar pruebas remotas y

acceder a la plataforma desde dispositivos móviles sin necesidad de configuraciones complejas como redirección de puertos o IP pública fija.

Figura 58

Exposición remota mediante Ngrok

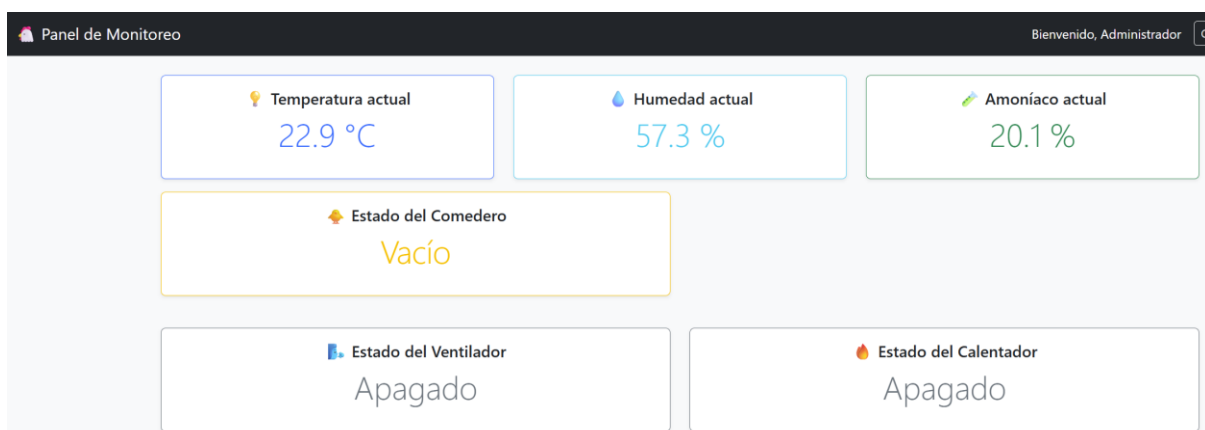
```
santiago@raspberrypi:~ $ curl http://localhost:4040/api/tunnels
{"tunnels":[{"name":"web","ID":"9e3d7c629c75d42da25c2174cb57b08b","5-70-13-164.ngrok-free.app","proto":"https","config":{"addr":"http":"count":95,"gauge":0,"rate1":2.8687946904398514e-10,"rate5":0.000275046298352,"p90":7945780566.4000025,"p95":9618900240.399956,"p99":1632876e-10,"rate5":0.00040358345956661117,"rate15":0.001887526221774068590,"p99":2809553259.4199934}}}], "uri":"/api/tunnels"}
```

Fuente: Autoría

La interfaz fue diseñada con HTML, CSS y la biblioteca Bootstrap 5, lo que permitió una visualización adaptable a diferentes tamaños de pantalla, mejorando así la experiencia de usuario en dispositivos móviles y de escritorio, como se observa en la Figura 59. La visualización de datos se realizó mediante la integración de la librería Chart.js, que permitió representar gráficamente los registros de temperatura, humedad, amoníaco y distancia del comedero.

Figura 59

Panel de la interfaz web

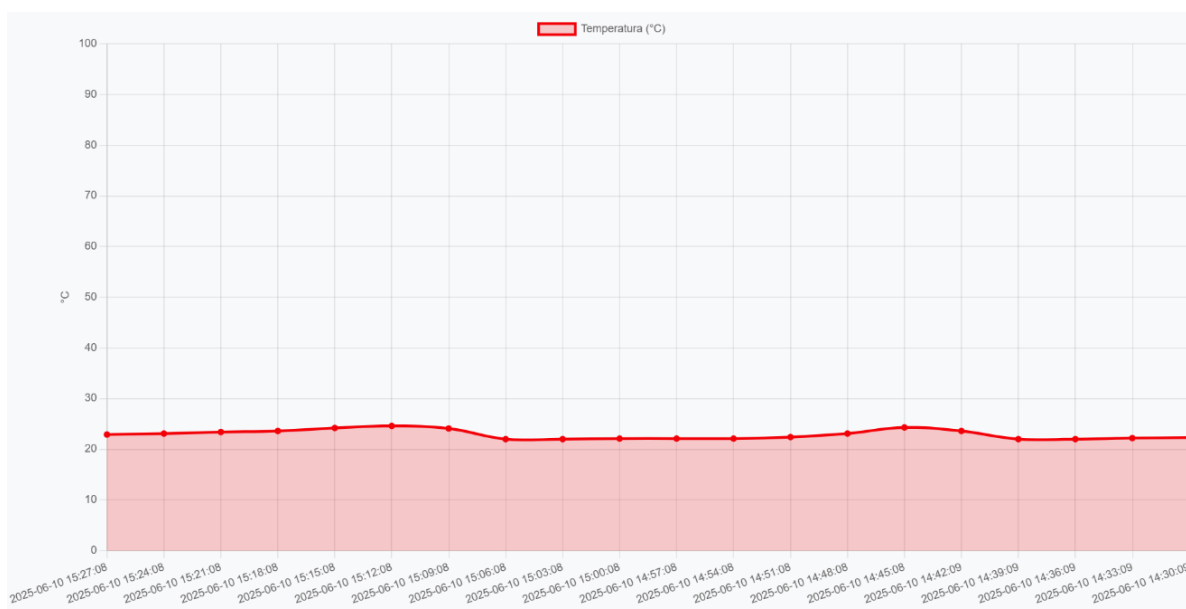


Fuente: Autoría

Cada valor mostrado se corresponde con los últimos registros almacenados en la base de datos, los cuales son obtenidos dinámicamente desde PHP. Se implementaron gráficas específicas para clasificar cada variable para facilitar la interpretación, por ejemplo, en la Figura 60 se observa la gráfica correspondiente a la temperatura, donde se guarda los 20 últimos registros, con esto es posible identificar una tendencia de la temperatura que se mantiene entre 22 a 28 grados, la cual es la adecuada para el bienestar de los pollos dentro del galpón.

Figura 60

Gráfica de Temperatura dentro del galpón



Fuente: Autoría

Se habilitó un panel de configuración exclusivo para administradores, desde donde se pueden modificar los rangos de temperatura, humedad y amoníaco considerados aceptables. Estos rangos son almacenados en un archivo rangos.json, y son leídos por Node-RED al momento del arranque, se puede observar en la Figura 61. Esto permite modificar la lógica de clasificación de datos sin necesidad de reprogramar el microcontrolador ni alterar el flujo principal, y en la plataforma web se observa en la Figura 62.

Figura 61*Configuración dinámica de rangos*

```
santiago@raspberrypi:~ $ cat /var/www/html/rangos.json
{
  "temp": {
    "TB": 29,
    "TM": 31
  },
  "hum": {
    "HB": 69,
    "HM": 71
  },
  "amo": {
    "AB": 6,
    "AM": 8
  },
  "actualizado": "2025-05-13 16:31:12"
}
```

Fuente: Autoría

Figura 62*Panel de configuración de rangos*

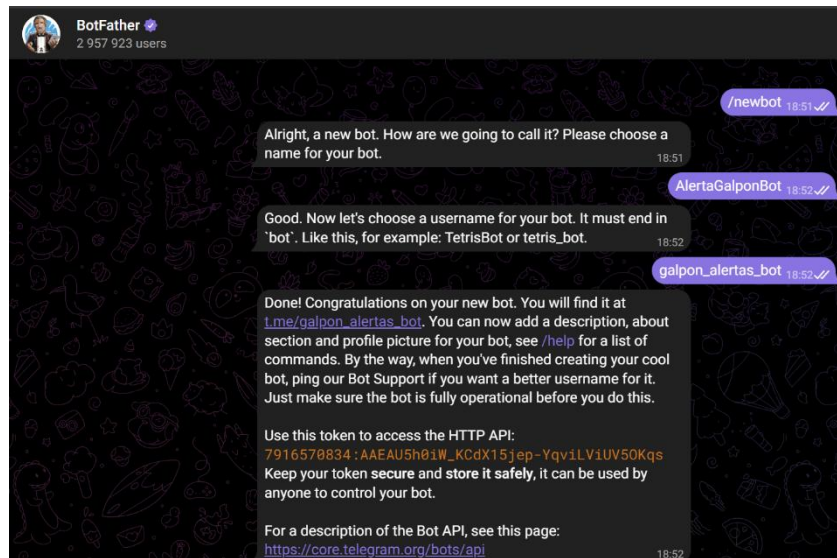
Fuente: Autoría

Para la implementación de alertas remotas, se optó por utilizar la plataforma de mensajería Telegram como se observa en la Figura 63. Se creó un Bot personalizado mediante el BotFather, obteniendo así un token único que permite establecer comunicación entre Node-

RED y la cuenta del usuario. Este Bot se identificó con el nombre `@galpon_alertas_bot`, y su `chatId` fue recuperado mediante consultas a la API de Telegram utilizando el comando `getUpdates`.

Figura 63

Configuración inicial del bot de Telegram



Fuente: Autoría

En el servidor Node-RED, ejecutándose sobre la Raspberry Pi se instaló el paquete `node-red-contrib-telegrambot`, el cual provee los nodos necesarios para enviar y recibir mensajes a través de Telegram. Esta instalación se realizó dentro del directorio Node-red utilizando el gestor de paquetes `npm`. Una vez instalado, se añadieron los nodos Telegram sender y Telegram receiver al flujo principal del sistema, como se observa en la Figura 64 y 65.

Figura 64

Instalación del nodo en Node-RED

```
santiago@raspberrypi:~ $ cd ~/.node-red
npm install node-red-contrib-telegrambot
npm warn deprecated har-validator@5.1.5: this library is no longer supported
npm warn deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older
versions may use Math.random() in certain circumstances, which is known to be pr
blematic. See https://v8.dev/blog/math-random for details.
npm warn deprecated request@2.88.2: request has been deprecated, see https://gi
hub.com/request/request/issues/3142

added 179 packages, and audited 262 packages in 31s

86 packages are looking for funding
  run `npm fund` for details

6 moderate severity vulnerabilities

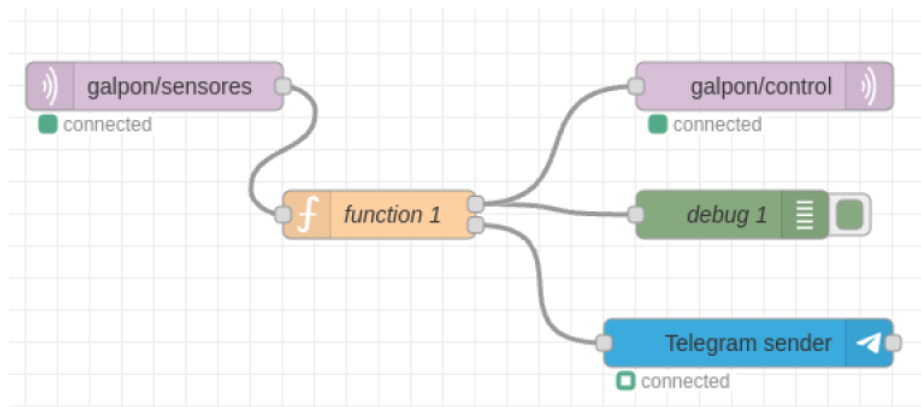
To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
```

Fuente: Autoría

Figura 65

Módulo Sender de Telegram en el flujograma



Fuente: Autoría

En el nodo de procesamiento (function) se añadió una segunda salida destinada a las alertas como se observa en la Figura 66. Se programó una verificación lógica que evalúa las siguientes condiciones críticas, como temperatura fuera del rango aceptable (menor a 20 °C o mayor a 30 °C), nivel alto de amoníaco (superior a 75 ppm) o comedero vacío (distancia medida por el sensor mayor a 150 mm).

Figura 66

Lógica de alerta en el nodo Function

```
// --- Verificación de alertas para Telegram
let alertas = [];

if (t === "TB" || t === "TA") {
  alertas.push(`Temperatura critica (${t})`);
}
if (a === "AA") {
  alertas.push("Nivel alto de amoniaco");
}
if (distancia > 180) {
  alertas.push(`El comedero esta vacio (distancia: ${distancia} mm)`);
}
if (alertas.length > 0) {
  alertaMsg = {
    payload: {
      type: "message",
      content: alertas.join("\n"),
      chatId: 7894920429
    }
  };
}

// --- Salida: [control MQTT, alerta Telegram]
return [salidaMsg, alertaMsg];
```

Fuente: Autoría

Cuando alguna de estas condiciones se cumple, se construye un mensaje estructurado como un objeto JSON con los campos esperados por el nodo telegram sender, y se envía a la segunda salida del flujo.

4.2.2.3 Implementación de Base de Datos

Para la gestión de los datos provenientes del sistema de monitoreo, se implementó una base de datos relacional utilizando el sistema gestor MariaDB, una bifurcación de código abierto de MySQL ampliamente utilizada en aplicaciones embebidas y servidores ligeros como la Raspberry Pi. La base de datos fue diseñada para almacenar de forma estructurada los valores generados por los sensores, así como los estados de los actuadores y otros eventos relevantes.

La base de datos fue denominada galpon_db y fue creada desde la línea de comandos mediante el siguiente comando, como se observa en la Figura 67.

Figura 67

Creación de la base de datos

```
CREATE DATABASE galpon_db;
```

Fuente: Autoría

En la Figura 68 se observa que se creó un usuario dedicado con privilegios suficientes para operar desde la aplicación web:

Figura 68

Creación de usuario en la base de datos

```
CREATE USER 'galpon_user'@'localhost' IDENTIFIED BY 'miclave123';  
GRANT ALL PRIVILEGES ON galpon_db.* TO 'galpon_user'@'localhost';  
FLUSH PRIVILEGES;
```

Fuente: Autoría

Se definieron dos tablas principales como se observa en la Figura 69 y Figura 70, Registros y Actuadores, las cuales permiten registrar tanto los datos ambientales como las respuestas del sistema.

Figura 69

Creación de tabla Registros

```
CREATE TABLE Registros (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    idSensor INT NOT NULL,  
    valor FLOAT NOT NULL,  
    fecha_hora DATETIME NOT NULL  
);
```

Fuente: Autoría

Figura 70

Creación de tabla Actuadores

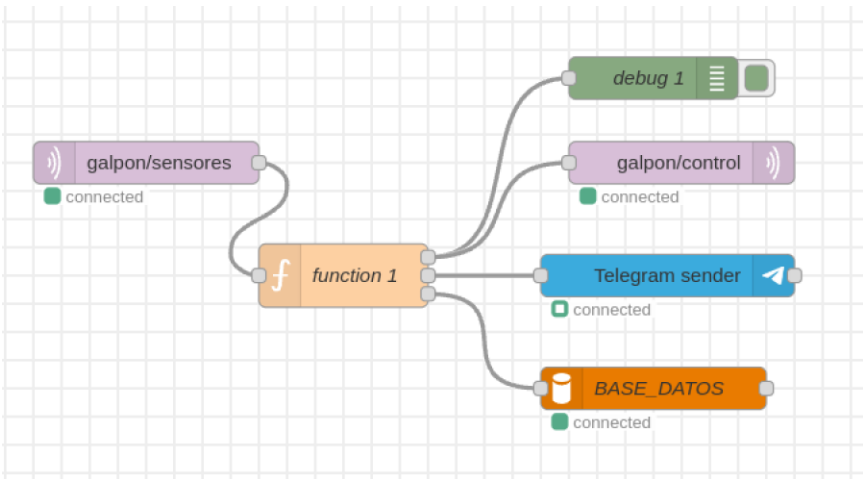
```
CREATE TABLE Actuadores (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    idActuador INT NOT NULL,  
    estado VARCHAR(10) NOT NULL,  
    fecha_hora DATETIME NOT NULL  
);
```

Fuente: Autoría

Node-RED fue utilizado como intermediario entre las placas ESP32 y la base de datos. Se configuraron nodos de tipo SQL para insertar los datos de forma automática cada vez que se recibía una lectura de sensores, se observa en la Figura 71.

Figura 71

Integración de un nodo MySQL para Node-red



Fuente: Autoría

La implementación de la base de datos proporcionó una estructura robusta y flexible para el almacenamiento histórico de lecturas y acciones del sistema, se observa un registro en la base de datos en la Figura 72.

Figura 72

Registros en la base de datos

```
1  USE galpon_db;
2  • SELECT * FROM Registros
3  ORDER BY fecha_hora DESC
4  LIMIT 10;
5
```

Result Grid				
Filter Rows:				
Edit:				
	idRegistro	idSensor	valor	fecha_hora
▶	3353	1	22.4	2025-05-13 19:23:13
	3355	3	1.12	2025-05-13 19:23:13
	3356	4	63.8	2025-05-13 19:23:13
	3354	2	92	2025-05-13 19:23:13
	3350	2	93	2025-05-13 19:22:13
	3349	1	22.4	2025-05-13 19:22:13
	3351	3	1.1	2025-05-13 19:22:13
	3352	4	63.9	2025-05-13 19:22:13
	3346	2	94	2025-05-13 19:21:13
	3345	1	22.5	2025-05-13 19:21:13
•	NULL	NULL	NULL	NULL

Fuente: Autoría

Su integración con Node-RED y PHP permitió una experiencia de monitoreo continua y confiable, y sentó las bases para futuras funcionalidades como análisis de tendencias, predicción con IA o generación de reportes históricos.

4.2.2.3 Implementación de Reglas Difusas

Para la toma de decisiones automáticas dentro del sistema de monitoreo del galpón inteligente, se implementó una lógica difusa personalizada mediante el uso de un nodo de tipo function dentro del entorno de programación visual Node-RED como se observa en la Figura 73. Esta lógica permite interpretar combinaciones de parámetros ambientales de forma más flexible y adaptativa que una lógica booleana estricta, facilitando la activación de actuadores en función de condiciones intermedias o graduales.

El nodo function recibe un objeto msg.payload que contiene los valores brutos de los sensores:

- rawTemp: temperatura
- rawHum: humedad
- rawAmo: nivel de amoníaco
- distancia: distancia del comedero

Estos datos son enviados previamente desde el ESP32, y validados en el nodo antes de ser procesados.

Se definieron rangos de referencia para cada variable (temperatura, humedad, amoníaco), los cuales fueron cargados desde un archivo rangos.json y almacenados en el contexto global de Node-RED. En función de estos límites, se clasificaron los valores como:

- Temperatura: TB (baja), TM (media), TA (alta)
- Humedad: HB (baja), HM (media), HA (alta)
- Amoníaco: AB (bajo), AM (medio), AA (alto)

Figura 73

Reglas difusas en Node-Red

```
// Tabla de reglas
const reglas = [
  ["TB", "HB", "AB", "V0", "C1"], ["TB", "HB", "AM", "V1", "C1"], ["TB", "HB", "AA", "V2", "C1"],
  ["TB", "HM", "AB", "V0", "C1"], ["TB", "HM", "AM", "V1", "C1"], ["TB", "HM", "AA", "V2", "C1"],
  ["TB", "HA", "AB", "V1", "C0"], ["TB", "HA", "AM", "V2", "C0"], ["TB", "HA", "AA", "V3", "C0"],
  ["TM", "HB", "AB", "V0", "C0"], ["TM", "HB", "AM", "V1", "C0"], ["TM", "HB", "AA", "V2", "C0"],
  ["TM", "HM", "AB", "V0", "C0"], ["TM", "HM", "AM", "V1", "C0"], ["TM", "HM", "AA", "V2", "C0"],
  ["TM", "HA", "AB", "V1", "C0"], ["TM", "HA", "AM", "V2", "C0"], ["TM", "HA", "AA", "V3", "C0"],
  ["TA", "HB", "AB", "V1", "C0"], ["TA", "HB", "AM", "V2", "C0"], ["TA", "HB", "AA", "V3", "C0"],
  ["TA", "HM", "AB", "V2", "C0"], ["TA", "HM", "AM", "V2", "C0"], ["TA", "HM", "AA", "V3", "C0"],
  ["TA", "HA", "AB", "V2", "C0"], ["TA", "HA", "AM", "V3", "C0"], ["TA", "HA", "AA", "V3", "C0"]
];
```

Fuente: Autoría

4.3 Pruebas preliminares de Funcionamiento del Sistema

En esta fase se realizarán pruebas individuales a cada componente del sistema con el objetivo de validar su funcionamiento de manera aislada, para ello se crea un plan de pruebas como se observa en la Tabla 40. Se evaluará el consumo energético del hardware, la precisión de los sensores, el correcto funcionamiento de los actuadores, la comunicación mediante el protocolo MQTT y la operación básica de la base de datos. Estas pruebas permitirán identificar posibles fallos o ajustes necesarios antes de integrar todos los elementos en un entorno funcional completo.

Tabla 40

Plan de pruebas preliminares del sistema

Nro.	Prueba	Objetivo	Componentes	Procedimiento
1	Prueba de consumo eléctrico.	de Verificar consumo energético sistema operación.	el Placa de procesamiento, del sensores en actuadores.	de Usar un medidor de corriente y registrar valores en distintas condiciones.

2	Prueba de funcionamiento de sensores y placa de toma de datos.	Asegurar lecturas correctas de los sensores.	Sensor de temperatura, humedad, NH ₃ , distancia y ESP32	de	Comprobar que las lecturas sean correctas de cada sensor.
3	Prueba de funcionamiento de Actuadores.	Comprobar que los actuadores funcionen correctamente.	Calentador y ventilador.	y	Activar y desactivar los actuadores y observar respuesta física.
4	Prueba de comunicación MQTT.	Validar la transmisión de datos entre nodos y Raspberry Pi.	ESP32, Raspberry Pi, broker MQTT.	Enviar simulados y recepción y almacenamiento.	mensajes verificados y almacenamiento.
5	Prueba de funcionamiento de Base de Datos.	Realizar operaciones básicas de almacenamiento.	Raspberry Pi, MySQL.	Pi,	Insertar, leer y eliminar registros manualmente.
6	Prueba de funcionamiento de lógica difusa.	Validar reglas creadas para el funcionamiento del sistema.	Reglas difusas, y estado de actuadores.	y	Probar las reglas con la ayuda de un software de simulación.

Fuente: Autoría

4.3.1 Pruebas de Consumo Eléctrico

A continuación, se detallan las pruebas realizadas para cada uno de los bloques del sistema.

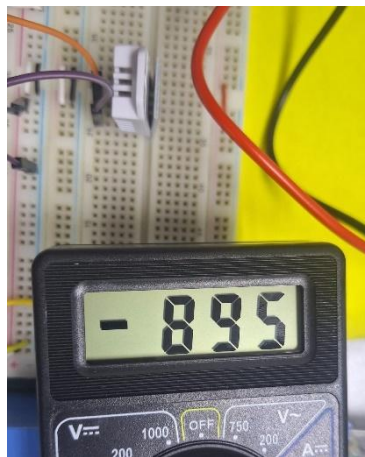
4.3.1.1. Pruebas de consumo eléctrico en el Bloque de Sensores y Toma de Datos

El bloque de sensores y toma de datos, compuesto por el ESP32 y los sensores DHT22, MQ-135 y VL53L0X, es una parte esencial del sistema de monitoreo del galpón. Dado que este bloque está en operación continua, es fundamental analizar su consumo eléctrico para validar la eficiencia del sistema y garantizar una alimentación estable.

Se conecta cada sensor por separado a una fuente de alimentación y se midió la corriente consumida con un multímetro digital como se observa en la Figura 74, 75 y 76. Se registraron los valores de corriente y se calculó el consumo de potencia en miliwatts (mW), basado en la tarifa de consumo de Ecuador (EMELNORTE, 2024).

Figura 74

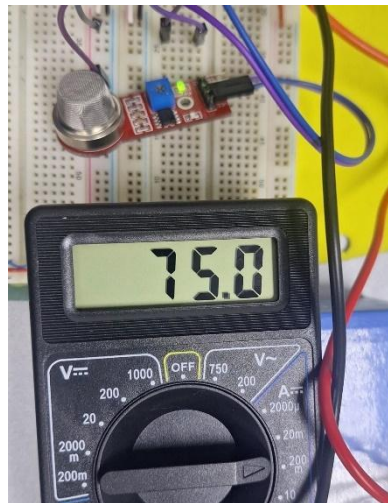
Consumo de corriente sensor de temperatura DHT22



Fuente: Autoría

Figura 75

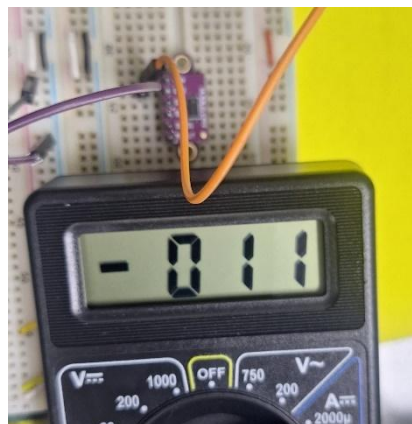
Consumo de corriente sensor de Amoniaco MQ135



Fuente: Autoría

Figura 76

Consumo de corriente sensor de nivel de alimento VL53L0X



Fuente: Autoría

Luego de realizar las mediciones para cada sensor se procede a calcular el consumo total como se observa en la Tabla 41, tanto del bloque de sensores como el bloque de toma de datos.

Tabla 41

Tabla de consumo eléctrico del bloque de sensores y toma de datos

Componente	Voltaje	Consumo de Corriente	Consumo de Potencia
ESP32	3.3V	75 mA	247.5 mW
DHT22	3.3V	0.895 mA	2.95 mW
MQ-135	5V	75 mA	375 mW
VL53L0X	3.5V	0.11 mA	0.29 mW
Consumo total	-	150.9 mA	625.74mW

Fuente: Autoría

El análisis de consumo eléctrico del bloque de sensores y el módulo de toma de datos revela un consumo total aproximado de 625.74mW, lo que indica una demanda energética relativamente baja, adecuada para aplicaciones en entornos con suministro limitado.

El ESP32 como núcleo de transmisión de datos, representa el mayor consumo dentro del sistema, con aproximadamente 247.5 mW, debido a su capacidad de comunicación inalámbrica y procesamiento continuo.

Entre los sensores, el MQ-135 es el que más energía requiere, con 375 mW, ya que utiliza un calentador interno para detectar gases en el ambiente. En contraste, el DHT22 tiene un consumo mucho menor, de apenas 2.95 mW, mientras que el VL53L0X es el más eficiente, con un consumo de 0.29 mW.

Estos resultados confirman que el sistema puede operar con fuentes de energía de bajo consumo y en caso de implementaciones en zonas remotas puede optimizarse con alimentación por batería o paneles solares.

4.3.1.2. Pruebas de consumo eléctrico en el Bloque de Procesamiento

Para estimar el consumo eléctrico de la Raspberry Pi 4 utilizada en el sistema, se realizó una medición utilizando un medidor USB digital de voltaje y corriente, como el modelo Keweisi KWS-V21. Este dispositivo, que normalmente se conecta entre la fuente de

alimentación y la Raspberry Pi, como se muestra en la Figura 77, permite visualizar en tiempo real valores de voltaje, corriente y potencia consumida.

Figura 77

Consumo eléctrico de la Raspberry Pi 4



Fuente: Autoría

Durante la operación típica del sistema que incluye tareas de procesamiento local y envío de datos a la nube. Se observaron valores promedio de 5.12 V de voltaje y 0.75 A de corriente, lo que equivale a una potencia de aproximadamente 3.84 W. Suponiendo un funcionamiento continuo de 24 horas al día, el consumo diario se estimó en 92.16 Wh, y el consumo mensual en 2.76 kWh.

4.3.1.3. Pruebas de consumo eléctrico en el Bloque de Actuadores

El bloque de actuadores está compuesto por tres dispositivos principales: un calentador infrarrojo de 250W, un ventilador de 110V y 250mA, y un foco convencional para iluminación. A continuación, en la Tabla 42 se realiza el cálculo del consumo eléctrico para cada uno.

Tabla 42

Tabla de consumo eléctrico para el Bloque de Actuadores

Actuador	Voltaje (V)	Corriente (A)	Potencia (W)	Consumo por hora (kWh)
Calentador	110V	2.27A	250W	0.25 kWh
Ventilador	110V	0.25A	27.5W	0.0275 kWh
Foco de Iluminación	110V	0.091A	10W	0.01 kWh

Fuente: Autoría

El calentador es el que mayor consumo tiene, representando un gasto energético significativo si permanece encendido por largos períodos.

El ventilador tiene un consumo moderado, y su uso eficiente puede mejorar la circulación del aire en el galpón sin representar un alto gasto energético.

El foco de iluminación tiene un consumo mínimo en comparación con los otros dispositivos, lo que lo hace ideal para mantener la iluminación nocturna con eficiencia energética.

Si estos actuadores funcionan simultáneamente, el consumo total por hora será 287.5W (0.2875 kWh). En un día de operación continua, esto equivaldría a 6.9 kWh, lo que representa un factor clave a considerar en la infraestructura eléctrica del galpón.

4.3.1.4. Consumo total del sistema mensualmente

Para determinar el consumo energético mensual del sistema, se aplicó la fórmula general:

$$\text{Consumo diario (kWh)} = (\text{Potencia en W} \times \text{Horas de uso por día}) \div 1000$$

Posteriormente, se obtuvo el consumo mensual multiplicando por 30 días:

$$\text{Consumo mensual (kWh)} = \text{Consumo diario (kWh)} \times 30$$

Finalmente, se estimó el costo mensual aplicando la tarifa de energía eléctrica vigente de (EMELNORTE, 2024).

$$\text{Costo mensual (USD)} = \text{Consumo mensual (kWh)} \times \text{Tarifa (USD/kWh)}$$

Se consideró una tarifa referencial de 0.10 USD por kWh, correspondiente al régimen residencial subsidiado en la ciudad de Ibarra, provincia de Imbabura.

Bloque de Sensores y Toma de Datos

Este bloque incluye el microcontrolador ESP32 y los sensores DHT22, MQ135 y VL53L0X. El consumo total medido fue de 0.62574 W, funcionando de forma continua (24 horas al día). Aplicando la fórmula:

$$\text{Consumo diario} = (0.62574 \text{ W} \times 24 \text{ h}) \div 1000 = 0.015 \text{ kWh/día}$$

$$\text{Consumo mensual} = 0.015 \times 30 = 0.45 \text{ kWh}$$

$$\text{Costo mensual} = 0.45 \times 0.10 = 0.05 \text{ USD}$$

Bloque de Procesamiento

El bloque de procesamiento está compuesto por una Raspberry Pi 4, que opera continuamente. Su consumo se estimó en 3.84 W:

$$\text{Consumo diario} = (3.84 \text{ W} \times 24 \text{ h}) \div 1000 = 0.09216 \text{ kWh}$$

$$\text{Consumo mensual} = 0.09216 \times 30 = 2.76 \text{ kWh}$$

$$\text{Costo mensual} = 2.76 \times 0.10 = 0.28 \text{ USD}$$

Bloque de Actuadores

Incluye tres dispositivos: un foco de iluminación, un ventilador, y un calentador infrarrojo. Se asumieron horarios promedio de uso basados en el comportamiento típico del sistema:

Foco de iluminación: 10 W, encendido 12 h/día (de 18h00 a 6h00)

$$\text{Consumo diario} = (10 \times 12) \div 1000 = 0.12 \text{ kWh}$$

$$\text{Consumo mensual} = 0.12 \times 30 = 3.6 \text{ kWh}$$

$$\text{Costo mensual} = 3.6 \times 0.10 = 0.36 \text{ USD}$$

Ventilador: 27.5 W, encendido 2 h/día

$$\text{Consumo diario} = (27.5 \times 2) \div 1000 = 0.055 \text{ kWh}$$

$$\text{Consumo mensual} = 0.055 \times 30 = 1.65 \text{ kWh}$$

$$\text{Costo mensual} = 1.65 \times 0.10 = 0.17 \text{ USD}$$

Calentador: 250 W, encendido 4 h/día

$$\text{Consumo diario} = (250 \times 4) \div 1000 = 1.0 \text{ kWh}$$

$$\text{Consumo mensual} = 1.0 \times 30 = 30.0 \text{ kWh}$$

$$\text{Costo mensual} = 30.0 \times 0.10 = 3.00 \text{ USD}$$

Resumen de bloque de actuadores:

$$\text{Consumo diario total} = 0.12 + 0.055 + 1.0 = 1.175 \text{ kWh/día}$$

$$\text{Consumo mensual total} = 1.175 \times 30 = 35.25 \text{ kWh}$$

$$\text{Costo mensual total} = 3.52 \text{ USD}$$

Finalmente, en la Tabla 43 se muestra un resumen de todos los cálculos realizados dentro del consumo eléctrico del sistema.

Tabla 43

Tabla de consumo eléctrico mensualmente del sistema.

Bloque	Componentes Incluidos	Consumo Diario (kWh)	Consumo Mensual (kWh)	Costo Mensual (\$)
Sensores y Toma de Datos	ESP32, DHT22, MQ135, VL53L0X	0.015	0.45	0.05
Procesamiento	Raspberry Pi 4	0.092	2.76	0.28
Actuadores	Foco (12h), Ventilador (2h), Calentador (4h)	1.175	35.25	3.28
Total	-	1.550	46.50	\$3.61

Fuente: Autoría

El consumo total mensual estimado del sistema completo es de aproximadamente 3,61 dólares

4.3.2 Prueba de funcionamiento de sensores y placa de toma de datos

El bloque de sensores y Toma de Datos es una parte crítica del sistema de monitoreo del galpón, ya que proporciona las mediciones necesarias para la toma de decisiones automatizada. Estas pruebas tienen como objetivo verificar que los sensores DHT22, MQ-135 y VL53L0X funcionan correctamente y generan datos confiables para el sistema de control, todo esto procesado por el ESP32, como se observa en la Figura 78.

Figura 78

Prueba de lectura de Sensores

```
20:03:48.956 -> ----- DATOS ENVIADOS -----
20:03:48.956 -> Temp: 23.50 °C | Hum: 58.20 % | Amoníaco: 1.56 %
20:03:48.956 -> Distancia: 133 mm | D0: 1
20:03:48.956 -> 📡 Datos enviados por MQTT
```

Fuente: Autoría

Las pruebas realizadas confirman que el bloque de sensores opera de manera eficiente y proporciona datos confiables para el monitoreo del galpón.

4.3.3 Prueba de funcionamiento de Actuadores.

Para validar el correcto funcionamiento de los actuadores bajo el control de una lógica difusa, se diseñaron pruebas controladas en las que se simularon diferentes combinaciones de temperatura, humedad y niveles de amoníaco, como se observa en la Tabla 44. La lógica difusa implementada en Node-RED clasifica los valores de los sensores en estados lingüísticos (por ejemplo: ‘Temperatura Alta’, ‘Humedad Baja’) y en función de reglas predefinidas, determina las acciones sobre los actuadores.

Tabla 44

Funcionamiento de los actuadores

Condición simulada	Clasificación	Actuador esperado	Resultado
	difusa		observado
Temp: 28°C, Hum: 50%, Amo: 2ppm	TB, HB, AB	Calentador ON	Encendido correcto
Temp: 30°C, Hum: 60%, Amo: 2ppm	TA, HB, AB	Ventilador ON	Encendido correcto

Fuente: Autoría

Por ejemplo, si la temperatura es clasificada como ‘Alta’ (TA) y la humedad como ‘Alta’ (HA), la lógica activa el ventilador para estabilizar el ambiente. La verificación se realizó observando el estado físico de los actuadores, los mensajes generados por el sistema y los registros en la base de datos.

En la primera prueba, al simular una temperatura de 30 °C, humedad del 50 % y un nivel de amoníaco de 2 ppm, el sistema clasificó las entradas como ‘TB’, ‘HB’ y ‘AB’, lo que corresponde a condiciones frías, se observa en la Figura 79 y Figura 80.

Figura 79

Condiciones corresponden a un ambiente frío

```

temperatura: "TB"
humedad: "HB"
amoníaco: "AB"
distancia: 131
estado_comedero: "Vacío"
ventilador: "V0"
calentador: "C1"
rawTemp: 28.2
rawHum: 50
rawAmo: 1.88

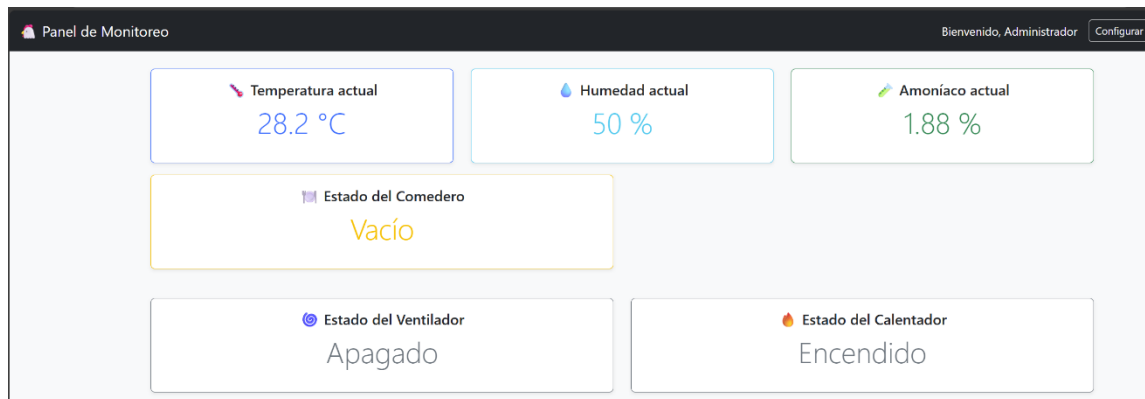
```

Fuente: Autoría

Según la regla difusa establecida, esto activa el ventilador, y efectivamente, se verificó su encendido tanto visualmente como en los registros del sistema.

Figura 80

Verificación de estado de actuadores



Fuente: Autoría

En la segunda prueba, con 30°C, 50 % de humedad y 2 ppm de amoníaco (clasificados como 'TA', 'HB' y 'AB'), las condiciones corresponden a un ambiente cálido, como se observa en las Figuras 81 y 82.

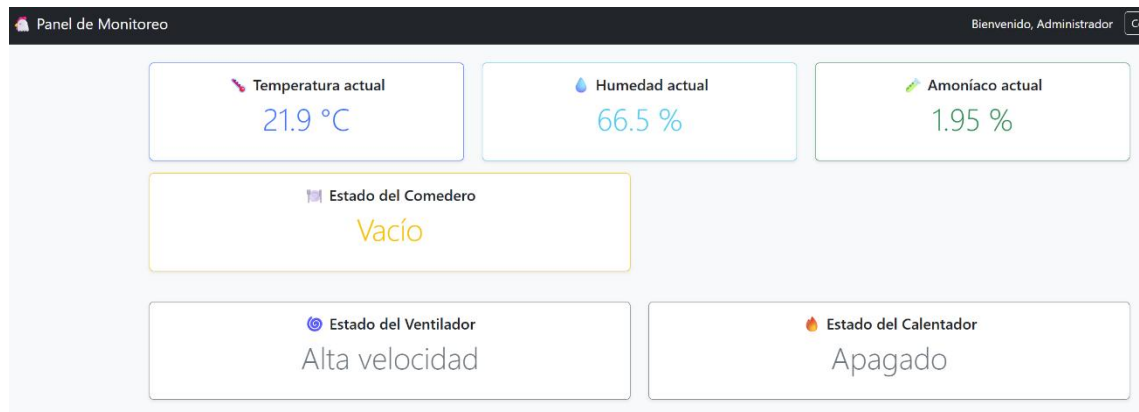
Figura 81

Condiciones corresponden a un ambiente cálido

```
temperatura: "TA"
humedad: "HB"
amoniaco: "AB"
distancia: 130
estado_comedero: "Vacío"
ventilador: "V1"
calentador: "C0"
rawTemp: 29.8
rawHum: 46.7
rawAmo: 2.05
```

Fuente: Autoría

Esto activó correctamente el calefactor infrarrojo, validando otra de las reglas esenciales del sistema.

Figura 82*Verificación de estado de actuadores*

Fuente: Autoría

4.3.4 Prueba de comunicación MQTT.

El bloque de procesamiento, representado por la Raspberry Pi 4, es el núcleo encargado de gestionar la recepción, procesamiento y almacenamiento de los datos obtenidos por los sensores. Además, ejecuta la lógica difusa, controla la activación de los actuadores y maneja la interfaz de visualización de datos en tiempo real.

En la Figura 83, se observa los datos recibidos desde el ESP32, hacia el bloque de procesamiento, donde este alojado en servidor MQTT de Node-Red.

Figura 83*Datos recibidos desde el cliente MQTT*

```
msg.payload : Object
  object
    temperatura: "TB"
    humedad: "HB"
    amoniaco: "AB"
    distancia: 131
    estado_comedero: "Vacío"
    ventilador: "V0"
    calentador: "C1"
    rawTemp: 23.6
    rawHum: 58.1
    rawAmo: 1.61
```

Fuente: Autoría

4.3.5 Prueba de funcionamiento de Base de Datos

Para comprobar que la estructura de la base de datos se corresponde con el modelo entidad-relación propuesto, se utilizó el comando DESCRIBE sobre las tablas Usuario, Sensores, Registros y Actuadores, como se observa en la Figura 84.

Figura 84

Tablas principales de la base de datos

```
MariaDB [galpon_db]> -- Ver estructura de cada tabla
MariaDB [galpon_db]> DESCRIBE Usuario;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| idUsuario  | int(11)   | NO   | PRI | NULL    | auto_increment |
| usuario    | varchar(45) | NO   |     | NULL    |              |
| nombre     | varchar(45) | YES  |     | NULL    |              |
| contrasena | varchar(255) | YES  |     | NULL    |              |
| cedula     | varchar(45) | YES  |     | NULL    |              |
| correo     | varchar(100) | YES  |     | NULL    |              |
| telefono   | varchar(20) | YES  |     | NULL    |              |
| direccion  | varchar(100) | YES  |     | NULL    |              |
| tipo_usuario | varchar(20) | YES  |     | NULL    |              |
| estado     | varchar(20) | YES  |     | NULL    |              |
| fecha_registro | datetime | YES  |     | NULL    |              |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0,001 sec)

MariaDB [galpon_db]> DESCRIBE Sensores;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| idSensores | int(11)   | NO   | PRI | NULL    | auto_increment |
| tipo_sensor | varchar(45) | YES  |     | NULL    |              |
| unidad     | varchar(45) | YES  |     | NULL    |              |
| descripcion | varchar(100) | YES  |     | NULL    |              |
| idUsuario  | int(11)   | YES  | MUL | NULL    |              |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0,002 sec)

MariaDB [galpon_db]> DESCRIBE Registros;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| idRegistro | int(11)   | NO   | PRI | NULL    | auto_increment |
| idSensor   | int(11)   | NO   |     | NULL    |              |
| valor      | float     | NO   |     | NULL    |              |
| fecha_hora | datetime  | NO   |     | NULL    |              |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0,001 sec)

MariaDB [galpon_db]> DESCRIBE Actuadores;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| idActuador | int(11)   | NO   | PRI | NULL    |              |
| tipo_actuador | varchar(45) | YES  |     | NULL    |              |
| estado     | varchar(45) | YES  |     | NULL    |              |
| fecha_hora | datetime  | NO   | PRI | NULL    |              |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0,001 sec)
```

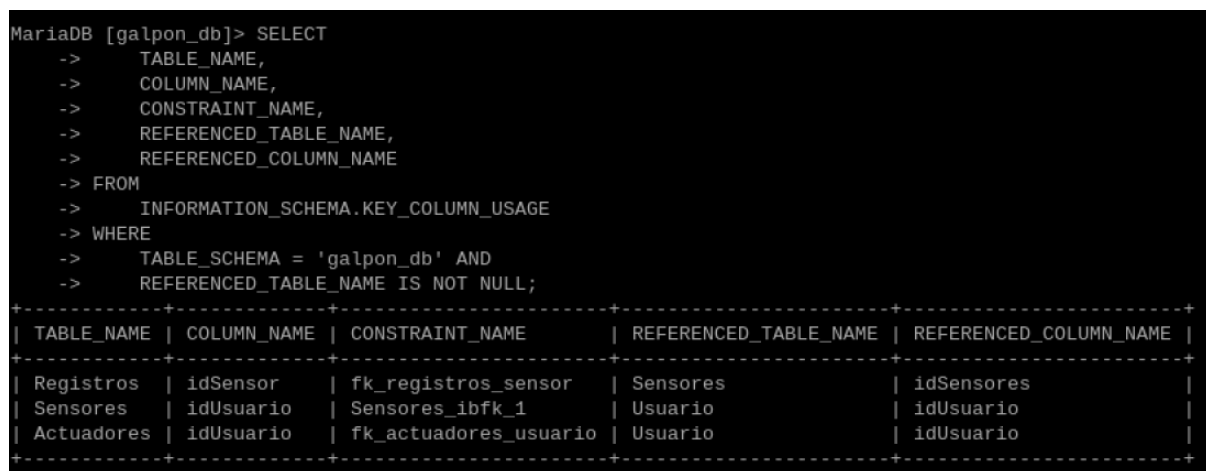
Fuente: Autoría

La salida obtenida permitió verificar que cada tabla contiene los campos definidos en el diseño lógico, incluyendo los tipos de datos, claves primarias y restricciones de nulidad.

Como parte de la validación de las relaciones definidas en el modelo entidad-relación, se ejecutó una consulta sobre la vista INFORMATION_SCHEMA.KEY_COLUMN_USAGE para identificar las claves foráneas implementadas en la base de datos, como se observa en la Figura 85. Los resultados confirmaron la existencia de relaciones entre las tablas, cumpliendo con lo estipulado en el diseño.

Figura 85

Validación de las relaciones definidas



```
MariaDB [galpon_db]> SELECT
-> TABLE_NAME,
-> COLUMN_NAME,
-> CONSTRAINT_NAME,
-> REFERENCED_TABLE_NAME,
-> REFERENCED_COLUMN_NAME
-> FROM
-> INFORMATION_SCHEMA.KEY_COLUMN_USAGE
-> WHERE
-> TABLE_SCHEMA = 'galpon_db' AND
-> REFERENCED_TABLE_NAME IS NOT NULL;
```

TABLE_NAME	COLUMN_NAME	CONSTRAINT_NAME	REFERENCED_TABLE_NAME	REFERENCED_COLUMN_NAME
Registros	idSensor	fk_registros_sensor	Sensores	idSensores
Sensores	idUsuario	Sensores_ibfk_1	Usuario	idUsuario
Actuadores	idUsuario	fk_actuadores_usuario	Usuario	idUsuario

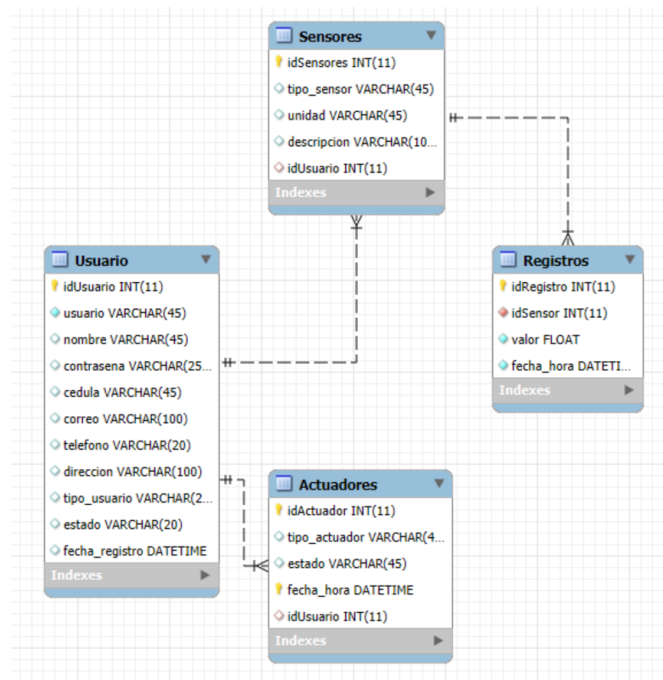
Fuente: Autoría

Esta revisión asegura que la integridad referencial entre entidades ha sido correctamente implementada.

Para complementar la verificación de la estructura de la base de datos, se generó un diagrama físico utilizando la funcionalidad de ingeniería inversa de MySQL Workbench, como se observa en la Figura 86. Esta herramienta permitió visualizar gráficamente las tablas creadas y sus relaciones, a partir del esquema existente en el sistema gestor de base de datos.

Figura 86

Verificación de la estructura de la base de datos



Fuente: Autoría

El diagrama resultante confirma que las entidades Usuario, Sensores, Registros y Actuadores han sido correctamente definidas, y que las claves foráneas establecen las asociaciones esperadas entre ellas. Esta representación gráfica valida que el modelo físico implementado guarda correspondencia directa con el modelo entidad-relación diseñado inicialmente.

Para comprobar el funcionamiento lógico del sistema de base de datos, se realizaron inserciones de prueba en cada una de las tablas principales, como se observa en la Figura 87, Figura 87, Figura 89 y Figura 90. Se creó un usuario con identificador 1, al cual se le asoció un sensor de tipo temperatura, y posteriormente se registró una lectura correspondiente en la tabla Registros. Asimismo, se añadió un actuador vinculado al mismo usuario.

Figura 87

Funcionamiento lógico de la Tabla Registros en la base de datos

```
MariaDB [galpon_db]> SELECT * FROM Registros ORDER BY fecha_hora DESC LIMIT 10;
```

idRegistro	idSensor	valor	fecha_hora
6131	3	1.56	2025-05-19 20:08:49
6129	1	24.2	2025-05-19 20:08:49
6130	2	131	2025-05-19 20:08:49
6132	4	56.4	2025-05-19 20:08:49
6126	2	131	2025-05-19 20:07:49
6127	3	1.61	2025-05-19 20:07:49
6125	1	23.6	2025-05-19 20:07:49
6128	4	58.1	2025-05-19 20:07:49
6122	2	132	2025-05-19 20:06:49
6123	3	1.51	2025-05-19 20:06:49

Fuente: Autoría

Figura 88

Funcionamiento lógico de la Tabla Actuadores en la base de datos

```
MariaDB [galpon_db]> SELECT * FROM Actuadores ORDER BY fecha_hora DESC LIMIT 10;
```

idActuador	tipo_actuador	estado	fecha_hora
1	NULL	V0	2025-05-19 20:15:49
2	NULL	C1	2025-05-19 20:15:49
1	NULL	V0	2025-05-19 20:14:49
2	NULL	C1	2025-05-19 20:14:49
2	NULL	C1	2025-05-19 20:13:49
1	NULL	V0	2025-05-19 20:13:49
1	NULL	V0	2025-05-19 20:12:49
2	NULL	C1	2025-05-19 20:12:49
2	NULL	C1	2025-05-19 20:11:54
1	NULL	V0	2025-05-19 20:11:54

Fuente: Autoría

Figura 89

Funcionamiento lógico de la Tabla Usuario en la base de datos

```
MariaDB [galpon_db]> SELECT * FROM Usuario;
```

idUsuario	usuario	nombre	contrasena	cedula
1	admin	Administrador	\$2y\$10\$B4upTmnDBujsy7SfCdqRledcAzIbQtuchqRiPcbkPtKM0NmocFft0	1234567890
3	santiago	Santiago Chuquilla	\$2y\$10\$CwKMq/bpl9dZ.JeICzuKP0z3BiTQmFklW7imak.duyAaQdpXbRzq0	1723456789

Fuente: Autoría

Figura 90

Funcionamiento lógico de la Tabla Sensores en la base de datos

```
MariaDB [galpon_db]> SELECT * FROM Sensores;
```

idSensores	tipo_sensor	unidad	descripcion	idUsuario
1	Temperatura	Celsius	NULL	1
2	Distancia	mm	NULL	1
3	Amoniaco	ppm	NULL	1
4	Humedad	%	NULL	1

Fuente: Autoría

Estas operaciones permitieron validar que las relaciones entre las entidades están correctamente definidas, ya que no se generaron errores al insertar datos relacionados. Las consultas posteriores mostraron la información de manera consistente, lo que demuestra que la base de datos permite almacenar, asociar y recuperar datos de forma adecuada conforme al diseño del sistema.

4.3.6 Prueba de funcionamiento de lógica difusa en software de simulación

Como parte del desarrollo del sistema automatizado, se implementó un módulo de simulación en Python con el objetivo de validar la lógica de las reglas difusas antes de su implementación física. Este simulador permitió generar múltiples escenarios ambientales de forma aleatoria, clasificarlos según etiquetas lingüísticas previamente definidas y comprobar su correspondencia con el conjunto de reglas del sistema. La verificación de estas reglas aseguró que cada combinación de entrada produjera una respuesta lógica coherente, contribuyendo así a garantizar la robustez del sistema de control automatizado.

Como primer paso se importan dos librerías fundamentales para el desarrollo del simulador random permite generar valores aleatorios dentro de rangos definidos para representar diferentes condiciones ambientales reales como se observa en la Figura 91.

Figura 91

Importación de librerías

```
import random
import pandas as pd
```

Fuente: Autoría

Se definen funciones personalizadas para clasificar los valores de temperatura como se observan en la Figura 92, humedad y amoníaco en categorías lingüísticas difusas. Por ejemplo, la función `clasificar_temperatura()` clasifica el valor recibido en TB (temperatura baja), TM (media) o TA (alta) según umbrales definidos. Lo mismo ocurre con `clasificar_humedad()` y `clasificar_amoníaco()`, siguiendo el mismo principio de partición por rangos.

Figura 92

Funciones de clasificación difusa

```
def clasificar_temperatura(t):
    if t <= 15:
        return 'TB'
    elif t <= 25:
        return 'TM'
    else:
        return 'TA'
```

Fuente: Autoría

Se define una lista de 27 tuplas que representan las reglas difusas del sistema. Cada regla contiene, 3 entradas difusas (Temperatura, Humedad, Amoníaco), 2 salidas asociadas (Estado del ventilador, Estado del calentador) como se observa en la Figura 93.

Estas reglas actúan como base del motor de inferencia y representan el conocimiento experto programado para la automatización del ambiente.

Figura 93

Definición de las reglas difusas

```
reglas_difusas = [
    ('TA', 'HB', 'AB', 'V0', 'C1'),
    ('TB', 'HB', 'AM', 'V1', 'C1'),
    ('TB', 'HB', 'AA', 'V2', 'C1'),
    ('TB', 'HM', 'AB', 'V0', 'C1'),
    ('TB', 'HM', 'AM', 'V1', 'C1'),
    ('TB', 'HM', 'AA', 'V2', 'C1'),
    ('TB', 'HA', 'AB', 'V1', 'C0'),
    ('TB', 'HA', 'AM', 'V2', 'C0'),
    ('TB', 'HA', 'AA', 'V3', 'C0'),
    ('TM', 'HB', 'AB', 'V0', 'C0'),
    ('TM', 'HB', 'AM', 'V1', 'C0'),
    ('TM', 'HB', 'AA', 'V2', 'C0'),
    ('TM', 'HM', 'AB', 'V0', 'C0'),
    ('TM', 'HM', 'AM', 'V1', 'C0'),
    ('TM', 'HM', 'AA', 'V2', 'C0'),
    ('TM', 'HA', 'AB', 'V1', 'C0'),
    ('TM', 'HA', 'AM', 'V2', 'C0'),
    ('TM', 'HA', 'AA', 'V3', 'C0')
]
```

Fuente: Autoría

Luego de implementar las reglas se procede a probar el programa como se observa en la Figura 94, con el fin de comprobar el correcto funcionamiento de la lógica difusa.

Figura 94

Pruebas de reglas definidas

Temperatura	Humedad	Amoniaco	Clasificación	Regla Activada	Estado de Ventilador	Estado de Calentador
29.85	22.99	33.26	TA-HB-AM	20.0	V2	C0
7.34	75.62	7.81	TB-HA-AB	7.0	V1	C0
37.51	73.03	89.02	TA-HA-AA	27.0	V3	C0
16.82	15.94	78.63	TM-HB-AA	12.0	V2	C0
16.26	83.60	2.17	TM-HA-AB	16.0	V1	C0
33.11	28.36	18.84	TA-HB-AB	1.0	V0	C1
31.66	44.32	68.27	TA-HB-AM	20.0	V2	C0
10.11	88.19	40.79	TB-HA-AM	8.0	V2	C0
13.81	78.78	46.57	TB-HA-AM	8.0	V2	C0
31.86	89.08	46.82	TA-HA-AM	26.0	V3	C0
33.42	65.35	39.83	TA-HM-AM	23.0	V2	C0
17.25	0.75	77.76	TM-HB-AA	12.0	V2	C0
23.60	66.40	97.35	TM-HM-AA	15.0	V2	C0
34.24	47.55	84.00	TA-HB-AA	21.0	V3	C0

Fuente: Autoría

Durante la ejecución del código, se generaron 100 combinaciones aleatorias que fueron clasificadas y evaluadas contra las reglas existentes. El sistema registró cuál de las reglas fue activada en cada caso, así como el estado de los actuadores definidos por dicha regla. Esta validación permitió comprobar que las 27 combinaciones de entrada posibles fueron cubiertas por al menos una regla, garantizando así una cobertura total del espacio de decisiones.

Se observó que en casos donde una combinación coincidía con más de una, el sistema seleccionaba la primera coincidencia encontrada, lo cual podría generar ambigüedad si no se realiza un control adecuado en el diseño lógico.

En términos generales, el sistema logró asignar correctamente los estados esperados de los actuadores para la mayoría de las combinaciones simuladas. No obstante, este tipo de pruebas también resulta útil para detectar reglas contradictorias o solapamientos, así como combinaciones que podrían no estar contempladas por error. La metodología aplicada constituye una validación preliminar efectiva antes de implementar el sistema completo en el entorno físico, garantizando que el comportamiento lógico de la automatización responda de manera coherente a las condiciones ambientales simuladas.

4.4. Pruebas generales del Funcionamiento del Sistema

Durante esta etapa se llevó a cabo la verificación del sistema automatizado de control ambiental en el galpón, bajo condiciones reales de operación. A diferencia de las pruebas preliminares, que se centraron en validar de forma aislada cada uno de los módulos, sensores, actuadores, comunicación, lógica difusa, base de datos y consumo eléctrico, en las pruebas generales se evaluó el comportamiento conjunto del sistema durante un ciclo completo de crianza de 50 pollos. La prueba es de 1 día de edad como se observa en la Figura 95, hasta los 15 días de edad, y se describe como se realiza la prueba en la Tabla 45.

Estas pruebas tuvieron como objetivo principal comprobar la interacción y respuesta coordinada de todos los componentes del sistema frente a las condiciones ambientales reales, así como analizar su efectividad para mantener los parámetros dentro de los rangos recomendados.

Figura 95

Pollo de 1 día de edad al iniciar la prueba general del sistema



Fuente: Autoría

Tabla 45

Descripción de la prueba general de funcionamiento del Sistema

Parámetro	Descripción
Cantidad de pollos	50
Duración	15 días continuos
Frecuencia de lectura de datos	Cada 3 minutos
Variables Monitoreadas	Temperatura, humedad, amoniaco y nivel de alimento
Actuadores involucrados	Calentador, Ventilador
Control del Sistema	Mediante lógica difusa, ejecutada en tiempo real
Visualización y monitoreo	Plataforma web integrada con base de datos y notificaciones vía Telegram

Fuente: Autoría

Esta tabla resume las condiciones bajo las cuales se ejecutaron las pruebas generales. La frecuencia de lectura de tres minutos permite obtener datos suficientes sin saturar el sistema. Las variables medidas están directamente relacionadas con el confort térmico, humedad, amoníaco y la eficiencia alimentaria de los pollos.

4.4.1 Comparación con el método tradicional de crianza

En la crianza tradicional de pollos en galpones de pequeña escala, el manejo de variables ambientales se realiza de manera manual, basándose principalmente en la experiencia empírica del operario. Las decisiones sobre la activación de calefactores, ventiladores o sistemas de iluminación suelen depender de la percepción subjetiva de la temperatura o del comportamiento observable de los animales. Este enfoque carece de un sistema de retroalimentación automática y no proporciona registros históricos que permitan evaluar o ajustar las condiciones ambientales de manera técnica.

Este enfoque, aunque funcional en entornos rurales, presenta limitaciones importantes en términos de precisión, trazabilidad y eficiencia energética.

4.4.1.1. Comparación de funcionamiento

A continuación, se presenta en la Tabla 46 una comparativa entre los dos enfoques de manejo del galpón.

Tabla 46

Comparación funcional entre los dos enfoques de manejo del galpón

Característica	Crianza Tradicional	Sistema Automatizado
Control de temperatura	Manual	Automático mediante sensor + lógica difusa
Ventilación	Activación ocasional	Activación automática según nivel de amoníaco

Supervisión de alimento	Revisión visual	Medición continua mediante sensor de distancia
Iluminación	Manual	Controlada automáticamente por tiempo
Registro de datos	Inexistente	Registro continuo en base de datos
Trazabilidad	No disponible	Visualización y exportación de registros
Reacción ante condiciones críticas	Depende el operario	Acciones automáticas e instantáneas
Supervisión	Presencia continua	Remota a través de plataforma web y alertas

Fuente: Autoría

La comparación funcional presentada permite evidenciar el valor agregado del sistema automatizado frente al método tradicional. Al automatizar procesos críticos como la ventilación o la calefacción, se reduce la carga operativa sobre el personal, se mejora la calidad del ambiente dentro del galpón y se mitigan riesgos asociados a condiciones inadecuadas. Adicionalmente, la posibilidad de monitoreo remoto y la trazabilidad de los datos permiten una toma de decisiones más informada y basada en evidencia.

Este cambio de enfoque, orientado a la digitalización y automatización del proceso de crianza, representa una mejora significativa tanto en términos de bienestar animal como de eficiencia operativa.

4.4.2 Resultados observados

Antes de iniciar la prueba general, se configuraron los parámetros de referencia que utilizaría el sistema de lógica difusa para la toma de decisiones. Estos rangos fueron

establecidos en función de recomendaciones técnicas para pollos de engorde en etapa inicial, y se muestran en la Figura 96.

Figura 96

Configuración de rangos para temperatura, humedad y amoníaco

-  **Temperatura:** TB ≤ 26 °C | TM ≤ 30 °C
-  **Humedad:** HB < 74 % | HM ≤ 76 %
-  **Amoníaco:** AB < 24.5 % | AM ≤ 25.5 %

Fuente: Autoría

Durante los 15 días de evaluación, el sistema registró continuamente las variables ambientales cada 3 minutos. A partir de estos registros, se calcularon los valores promedio diarios de cada parámetro: temperatura, humedad relativa, concentración de amoníaco y nivel del alimento en el comedero. Además, se documentaron las reglas de lógica difusa que se activaron como respuesta a las condiciones detectadas.

4.4.2.1. Evaluación del comportamiento del sistema en la primera camada

El sistema implementado para el control ambiental del galpón se basó en un conjunto de 27 reglas difusas predefinidas, las cuales se activaban cada vez que una combinación específica de temperatura, humedad y concentración de amoníaco coincidía con los rangos establecidos en la base de conocimiento.

Con el fin de analizar el comportamiento real del sistema, se construyeron dos tablas, que son 47 y 48.

Tabla 47

Activación diaria de las 27 reglas difusas

Día	Reglas activadas
1	R2, R5, R15

2	R2, R3, R5, R12, R22
3	R3, R6, R7, R10, R18, R27
4	R17, R22, R23
5	R9, R16, R21
6	R4, R5, R19, R20, R21, R22
7	R7, R15, R20, R22, R27
8	R11, R15, R22
9	R4, R6, R8, R18, R21, R22
10	R10, R17, R18, R27
11	R3, R9, R11, R14, R19, R26
12	R8, R20, R22
13	R3, R7, R10, R13, R16, R22
14	R2, R9, R16, R20, R24
15	R8, R17, R18, R23, R27

Fuente: Autoría

Esta tabla muestra qué reglas fueron activadas al menos una vez por día. Permite verificar la diversidad de reglas aplicadas a lo largo del periodo de prueba, lo cual evidencia que el sistema respondió dinámicamente a diferentes escenarios ambientales.

Esta segunda tabla resume el número total de veces que cada regla fue activada durante los 15 días de monitoreo. Los resultados obtenidos permiten identificar.

Tabla 48

Frecuencia total de activación de cada regla difusa

Regla	Total de Activaciones	Estado
R1	0	No se activó

R2	3	Sí se activó
R3	4	Sí se activó
R4	2	Sí se activó
R5	3	Sí se activó
R6	2	Sí se activó
R7	3	Sí se activó
R8	3	Sí se activó
R9	3	Sí se activó
R10	3	Sí se activó
R11	2	Sí se activó
R12	1	Sí se activó
R13	1	Sí se activó
R14	1	Sí se activó
R15	3	Sí se activó
R15	3	Sí se activó
R17	3	Sí se activó
R18	4	Sí se activó
R19	2	Sí se activó
R20	4	Sí se activó
R21	3	Sí se activó
R22	8	Sí se activó
R23	2	Sí se activó
R24	1	Sí se activó
R25	0	No se activó
R26	1	Sí se activó

R27	4	Sí se activó
-----	---	--------------

Fuente: Autoría

La Tabla presenta la cantidad total de veces que cada una de las 27 reglas fue activada durante los 15 días de prueba. Además, se indica si la regla fue efectivamente utilizada o no.

Este análisis permite identificar qué combinaciones de condiciones ambientales ocurrieron en el periodo observado. Las reglas no activadas podrían estar asociadas a situaciones extremas que no se presentaron, o a configuraciones menos probables del entorno. Esta información es útil para validar la cobertura de las reglas y para futuras mejoras del sistema.

Eventos Críticos Detectados

Como complemento a las tablas de reglas activadas presentadas anteriormente en la Tabla 47, se muestran eventos críticos registrados por el sistema entre el 30 de mayo y el 13 de junio de 2025. Estos eventos corresponden a condiciones ambientales fuera de los rangos considerados seguros para la crianza de pollos de hasta 15 días de edad.

Cada registro en la Tabla 49 detalla la fecha y hora de ocurrencia, el sensor implicado, el valor medido, el estado clasificado según los umbrales configurados y la acción esperada que el sistema debió ejecutar según la lógica difusa establecida.

Tabla 49

Eventos críticos detectados – Camada 1

Fecha y hora	Sensor	Valor	Estado	Acción esperada
2025-06-02 11:17:00	Amoniaco	26.29	AA	Ventilación activada
2025-06-04 20:02:00	Temperatura	30.71	TA	Ventilación activada

2025-06-06	Temperatura	25.1	TB	Calefactor encendido
11:41:00				
2025-06-12	Humedad	76.45	HA	Ventilación activada
14:51:00				
2025-06-15	Amoniaco	25.89	AA	Ventilación activada
19:28:00				

Fuente: Autoría

La Tabla 49 permite evidenciar el vínculo entre los datos reales obtenidos desde la base de datos y las respuestas del sistema automatizado, lo cual valida que las decisiones tomadas estuvieron fundamentadas en condiciones del entorno. Además, los eventos registrados muestran variabilidad en las condiciones ambientales, lo cual demuestra que el sistema estuvo en funcionamiento continuo y logró detectar situaciones de control relevantes, cumpliendo así su objetivo de mantener un ambiente adecuado para los pollos en su etapa inicial de crecimiento.

Evaluación del comportamiento del sistema

Tras la ejecución continua del sistema durante un período completo de 15 días, se constató su operatividad bajo condiciones reales de crianza de pollos desde el primer día de vida. A lo largo de este tiempo, el sistema monitoreó en tiempo real variables críticas como la temperatura, la humedad y la concentración de amoníaco, ejecutando acciones correctivas mediante una lógica difusa previamente configurada.

Durante las pruebas, se documentaron eventos críticos en los que se activaron los actuadores como respuesta a condiciones ambientales fuera de los rangos ideales. La capacidad del sistema para actuar de forma autónoma permitió mantener condiciones estables y seguras dentro del galpón, favoreciendo así el adecuado desarrollo de las aves.

La Figura 97 muestra el estado físico de un pollo al día 15, evidenciando un crecimiento acorde a su edad y en condiciones saludables, lo cual respalda indirectamente la efectividad del sistema de control automatizado implementado.

Figura 97

Pollo de 15 días de edad al terminar la primera camada



Fuente: Autoría

4.4.2.2. Evaluación del comportamiento del sistema en la segunda camada

Durante la segunda fase de pruebas, el sistema automatizado fue nuevamente puesto en funcionamiento continuo, siguiendo la misma configuración lógica que en la camada anterior. Se monitorearon las variables ambientales con un intervalo de lectura de tres minutos, aplicando las 27 reglas difusas preconfiguradas para activar los actuadores según las condiciones detectadas, esta prueba se realiza entre el 14 y el 28 de junio de 2025.

Se observa las reglas que fueron activadas durante la segunda prueba en la Tabla 50.

Tabla 50*Activación diaria de las 27 reglas difusas*

Día	Reglas activadas
1	R3, R5, R22
2	R2, R7, R12
3	R4, R6, R20
4	R15, R18, R27
5	R10, R14
6	R5, R11, R21
7	R7, R9, R23
8	R8, R20
9	R3, R16
10	R6, R17, R18
11	R13, R26
12	R4, R15, R22
13	R3, R10
14	R2, R9, R21
15	R11, R19, R27

Fuente: Autoría

La activación de reglas en esta segunda camada muestra un comportamiento similar al de la primera, aunque con una mayor distribución de reglas intermedias. Se observaron activaciones más frecuentes de reglas vinculadas a niveles altos de humedad y eventos combinados con temperatura media..

Eventos Críticos Detectados

Durante esta etapa, se identificaron nuevos eventos donde el sistema respondió automáticamente a condiciones de riesgo ambiental, como se observa en Tabla 51

Tabla 51

Eventos críticos registrados – Camada 2

Fecha y hora	Sensor	Valor	Estado	Acción esperada
2025-06-17 08:33:00	Humedad	77.90	HA	Ventilación activada
2025-06-19 19:15:00	Temperatura	23.65	TB	Calefactor encendido
2025-06-23 11:10:00	Amoniaco	27.01	AA	Ventilación activada
2025-06-12 14:51:00	Temperatura	31.12	TA	Ventilación activada
2025-06-30 17:19:00	Humedad	75.23	HA	Ventilación activada

Fuente: Autoría

En esta camada se identificaron más eventos relacionados con humedad alta y temperatura extrema, lo cual activó con frecuencia el sistema de ventilación.

Evaluación del comportamiento del sistema

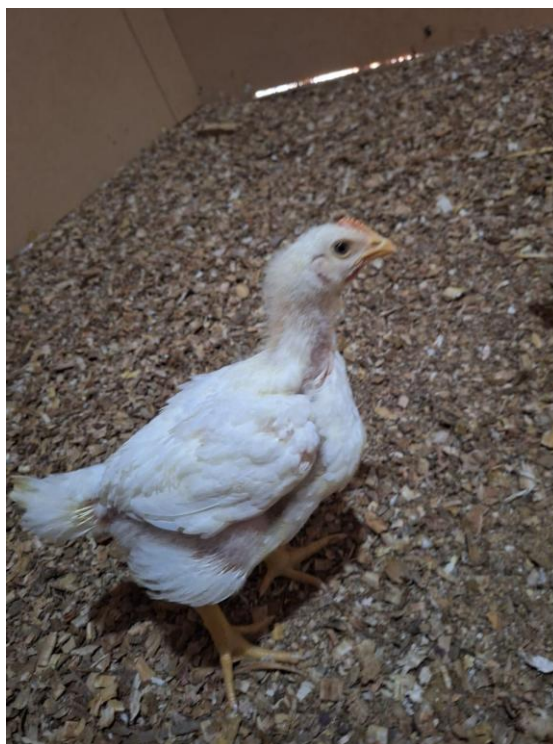
El sistema de control ambiental basado en lógica difusa demostró un comportamiento consistente durante la segunda camada. La activación de un conjunto diverso de reglas evidenció la capacidad del sistema para adaptarse dinámicamente a las condiciones ambientales cambiantes dentro del galpón. Se observó una mayor frecuencia de activación en reglas

relacionadas con humedad alta y temperatura extrema, lo que generó respuestas automáticas mediante el sistema de ventilación y calefacción. En particular, se destacó la activación recurrente de las reglas R22 (humedad alta y temperatura media) y R27 (combinaciones críticas de temperatura y amoníaco), lo que indica que el sistema respondió eficazmente ante eventos de riesgo.

Asimismo, los eventos críticos registrados permitieron validar el correcto funcionamiento de los sensores y de los algoritmos de toma de decisiones. Las acciones ejecutadas coincidieron con lo estipulado en la base de conocimiento difusa, lo que refleja una adecuada implementación lógica. Como resultado, se mantuvieron condiciones estables en el ambiente interno del galpón, favoreciendo el bienestar y el desarrollo de las aves durante el periodo de prueba. No se presentaron fallos de funcionamiento ni condiciones fuera de control, lo cual reafirma la robustez del sistema bajo condiciones reales de operación. En la Figura 98 se observa un pollo de 15 días de edad al finalizar la segunda camada.

Figura 98

Pollo de 15 días de edad al terminar la segunda camada



Fuente: Autoría

4.4.2.3. Evaluación del comportamiento del sistema en la tercera camada

Durante el tercer ciclo de pruebas se mantuvo la misma configuración del sistema, sin modificaciones en las reglas difusas ni en la calibración de sensores. Esta camada sirvió para validar la estabilidad del sistema en escenarios ambientales con mayor fluctuación térmica debido a cambios estacionales. El monitoreo continuo permitió evaluar nuevamente el desempeño de las reglas difusas aplicadas en tiempo real, se realiza entre el 29 de mayo y 13 de julio de 2025.

A continuación, se presentan las reglas activadas por día en la Tabla 52.

Tabla 52

Activación diaria de las 27 reglas difusas

Día	Reglas activadas
1	R2, R6, R10
2	R5, R15, R22
3	R7, R11, R27
4	R3, R12
5	R9, R16, R20
6	R6, R17, R22
7	R3, R8
8	R10, R13
9	R9, R20, R21
10	R2, R7, R11
11	R15, R26
12	R5, R16, R24
13	R9, R22, R27
14	R2, R9, R21

15	R3, R18, R20
----	--------------

Fuente: Autoría

Esta tabla evidencia una mayor distribución de reglas activadas, con predominancia de combinaciones asociadas a humedad elevada y fluctuaciones térmicas, particularmente en los días intermedios. Se mantuvo la activación de más del 90 % del conjunto de reglas, lo que reafirma la cobertura y capacidad de respuesta del sistema.

Eventos críticos detectados

Durante esta camada se identificaron varios eventos críticos, registrados automáticamente y asociados a condiciones que requirieron acción inmediata por parte del sistema.

Tabla 53

Eventos críticos registrados – Camada 3

Fecha y hora	Sensor	Valor	Estado	Acción esperada
2025-07-02 08:33:00	Temperatura	21.80	TB	Calefactor encendido
2025-07-07 19:15:00	Amoníaco	26.90	AA	Ventilación activada
2025-07-09 11:10:00	Humedad	77.15	HA	Ventilación activada
2025-07-12 14:51:00	Temperatura	30.55	TA	Ventilación activada
2025-07-15 17:19:00	Amoníaco	27.80	AA	Ventilación activada

Fuente: Autoría

Los registros muestran que el sistema logró detectar condiciones fuera de los umbrales de seguridad y respondió de forma automática mediante los actuadores disponibles. La lógica difusa aplicada fue capaz de gestionar estos eventos sin requerir intervención manual.

Evaluación del comportamiento del sistema durante la tercera camada

El sistema mantuvo su operatividad de manera constante, activando un amplio conjunto de reglas ante diferentes escenarios ambientales. La repetida activación de reglas relacionadas con humedad alta (HA) y niveles críticos de amoníaco (AA) confirmó que el galpón experimentó un entorno más exigente durante este ciclo. A pesar de ello, las acciones correctivas fueron ejecutadas de forma oportuna y precisa.

La frecuencia de activación de las reglas R20, R22 y R27 fue nuevamente elevada, lo que sugiere una consistencia en el tipo de condiciones enfrentadas y la correcta ejecución del algoritmo lógico. El comportamiento térmico también presentó mayor variabilidad, permitiendo poner a prueba la respuesta del sistema tanto ante temperaturas bajas como altas. No se evidenciaron fallos técnicos ni desviaciones en el funcionamiento, lo que confirma la madurez del sistema en su tercera iteración.

Se observa en la Figura 99 un pollo de 15 días de edad, luego de finalizar la tercera camada

Figura 99

Pollo de 15 días al finalizar la tercera camada



Fuente: Autoría

4.4.2.4. Evaluación del comportamiento del sistema en la tercera camada

La cuarta y última camada evaluada tuvo lugar entre el 14 y el 28 de julio de 2025. Durante este periodo, el sistema de monitoreo y control ambiental basado en lógica difusa operó de manera continua bajo las mismas condiciones que en los ciclos anteriores. Sin embargo, esta etapa permitió observar con mayor claridad el comportamiento del galpón frente a condiciones más cálidas y húmedas, propias del cierre del mes de julio, lo que generó nuevos escenarios de activación para el sistema automatizado.

La Tabla 54 presenta el conjunto de reglas difusas que fueron activadas durante cada día de la cuarta camada, con base en las combinaciones ambientales detectadas.

Tabla 54

Activación diaria de las 27 reglas difusas

Día	Reglas activadas
1	R2, R5, R10
2	R4, R7, R22
3	R6, R11, R16, R27
4	R9, R15
5	R3, R10, R17
6	R8, R18, R20, R23
7	R14, R16, R22
8	R10, R13
9	R9, R20, R21
10	R5, R7, R9, R21
11	R10, R11, R22
12	R6, R12, R19

13	R8, R20, R24
14	R10, R21, R23
15	R2, R6, R14

Fuente: Autoría

Durante esta camada se evidenció una mayor recurrencia en la activación de las reglas R10, R16, R22 y R27, todas asociadas a condiciones ambientales cercanas a los límites superiores de los rangos definidos. Esto refleja un entorno más exigente y permite validar la capacidad del sistema para reaccionar ante una variedad de condiciones.

Eventos críticos detectados

A lo largo de este ciclo de monitoreo se registraron múltiples eventos críticos, en la Tabla 55 se observa que los sensores detectaron valores fuera del rango aceptable y el sistema respondió de forma autónoma.

Tabla 55

Eventos críticos registrados – Camada 4

Fecha y hora	Sensor	Valor	Estado	Acción esperada
2025-07-17 08:33:00	Temperatura	17.80	TB	Calefactor encendido
2025-07-19 19:15:00	Amoníaco	28.40	AA	Ventilación Activada
2025-07-23 11:10:00	Temperatura	31.04	TA	Ventilación activada
2025-07-24 14:51:00	Humedad	78.90	HA	Ventilación activada
2025-07-28 07:19:00	Amoníaco	29.80	AA	Ventilación activada

Fuente: Autoría

Estos eventos confirman que el sistema mantuvo su capacidad de reacción frente a condiciones ambientales adversas. La activación del calefactor y el sistema de ventilación se produjo en concordancia con los valores registrados, alineándose con lo definido en las reglas difusas.

Evaluación del comportamiento del sistema durante la cuarta camada

Durante esta cuarta camada, el sistema de monitoreo y control automatizado demostró su estabilidad y efectividad frente a condiciones climáticas más complejas. Las variaciones de temperatura y humedad, así como el incremento gradual de la concentración de amoníaco, permitieron que un mayor número de reglas se activaran, lo cual pone en evidencia el dinamismo del sistema y la adecuación de su base de conocimiento.

Se destaca la activación frecuente de reglas relacionadas con combinaciones múltiples, lo cual evidencia que el sistema alcanzó una madurez funcional al reconocer condiciones compuestas y responder con precisión. Las acciones ejecutadas fueron coherentes con los valores registrados, lo cual refuerza la fiabilidad de las decisiones tomadas por la lógica difusa.

No se registraron fallos ni comportamientos erráticos en sensores o actuadores, consolidando al sistema como una solución robusta y viable para el manejo automatizado del ambiente en galpones avícolas.

Al finalizar la cuarta camada, se puede evidenciar en la Figura 100, un pollo luego de terminar la prueba.

Figura 100

Pollo de 15 días al finalizar la cuarta camada



Fuente: Autoría

4.5. Funcionamiento del sistema en la plataforma web durante las pruebas por camadas

Durante el desarrollo de las cuatro camadas experimentales, la plataforma web cumplió un rol fundamental en la supervisión continua y en la validación visual del funcionamiento del sistema automatizado. Gracias a la integración directa con la base de datos, fue posible consultar registros en tiempo real y también acceder al historial de variables ambientales y decisiones tomadas por el sistema.

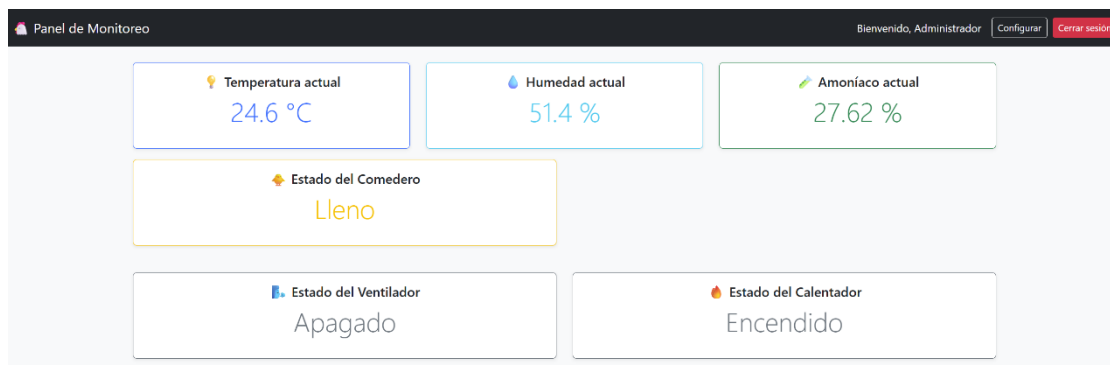
En cada una de las camadas, la interfaz permitió monitorear el comportamiento de las tres variables principales (temperatura, humedad y amoníaco) y visualizar el estado actual de los actuadores. A través de gráficos actualizados cada 3 minutos y paneles de control, los usuarios pudieron verificar si el sistema respondía correctamente frente a condiciones críticas, como temperaturas bajas al amanecer o acumulaciones de amoníaco en horarios de alta actividad de las aves.

Además, se visualizaron los cambios en los umbrales y reglas configuradas, permitiendo adaptar el sistema a condiciones climáticas ligeramente distintas en cada camada. Esta flexibilidad se evidenció en la activación variable de reglas entre los diferentes periodos, lo cual fue fácilmente verificable mediante la plataforma. Se destacan a continuación algunos ejemplos del comportamiento observado:

Primera camada: Activaciones frecuentes del calefactor como se observa en la Figura 101, aquí se evidencia que se cumple la regla 4, TB-HB-AB, estas activaciones registradas y visualizadas con claridad en el panel de la plataforma.

Figura 101

Captura de la plataforma web con eventos de temperatura por debajo del umbral

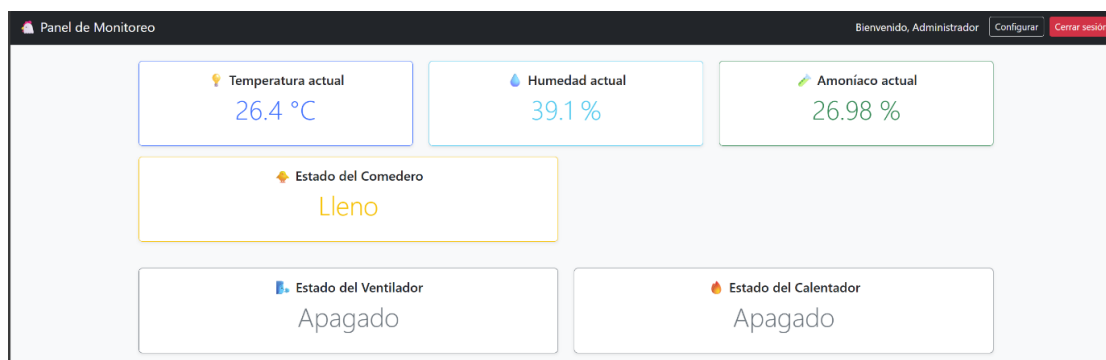


Fuente: Autoría

Segunda camada: Como se observa en la Figura 102, la disminución gradual de humedad evidenciado en gráficos, pero con niveles diferentes en cuanto a temperatura y humedad, por lo tanto, los actuadores se encuentran apagados, cumpliendo reglas difusas, para este caso es la regla 13, TM-HB-AB.

Figura 102

Captura de la plataforma web mostrando la disminución de humedad

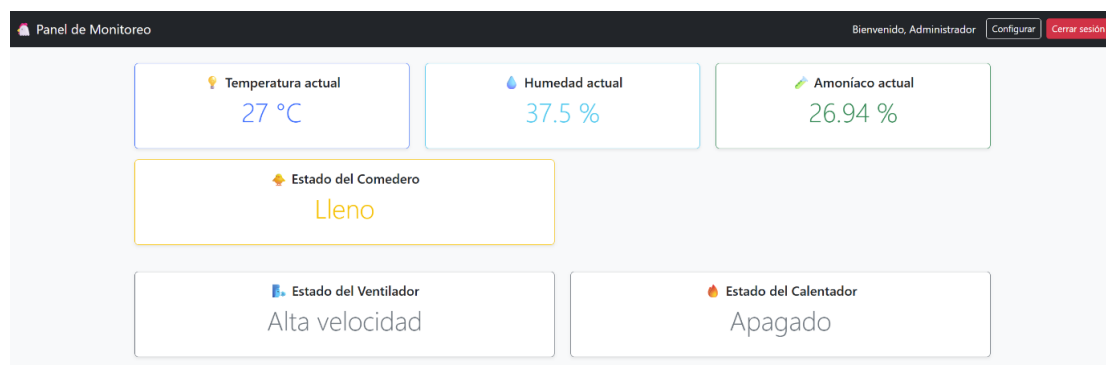


Fuente: Autoría

Tercera camada: Niveles por encima del umbral de amoníaco, lo cual activa la regla 12 TM-HB-AA, estos datos registrados en los últimos días de prueba de la tercera camada, hacen que la ventilación se mantenga encendida, como se observa en la Figura 103.

Figura 103

Captura de plataforma web mostrando aumento de amoníaco.



Fuente: Autoría

Cuarta camada: Funcionamiento más equilibrado, con menos eventos críticos, evidenciando el funcionamiento óptimo del sistema, activándose la regla 15 TM-HM-AA.

Figura 104

Parámetros en la plataforma web

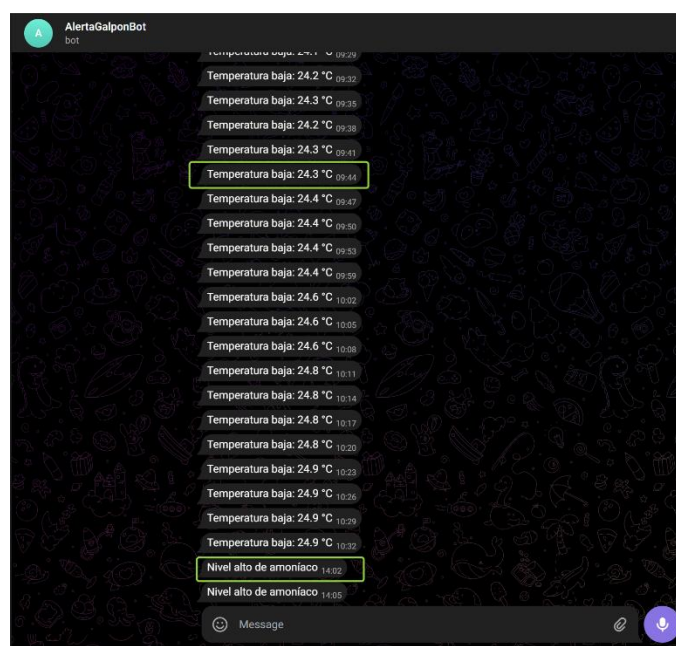


Fuente: Autoría

Cada una de estas situaciones fue complementada con la recepción de notificaciones a través del Bot de Telegram como se observa en la figura 105, lo cual permitió confirmar desde dispositivos móviles que el sistema había reaccionado correctamente.

Figura 105

Notificaciones en Telegram sobre alertas del sistema



Fuente: Autoría

Estas evidencias respaldan visualmente la operatividad del sistema y su capacidad de respuesta, permitiendo a los usuarios validar en todo momento el cumplimiento de los objetivos del monitoreo y control ambiental.

4.6 Simulación de condiciones ambientales extremas en camadas adicionales

4.6.1 Justificación de la simulación

Si bien las pruebas prácticas realizadas con cuatro camadas reales demostraron el funcionamiento estable del sistema bajo condiciones controladas, no se observaron situaciones ambientales críticas que pusieran en riesgo la salud de las aves. Por esta razón, se realizó una simulación adicional basada en un modelo matemático, con el objetivo de evaluar el comportamiento del sistema ante condiciones extremas y validar su capacidad de respuesta ante situaciones potencialmente peligrosas.

Esta simulación permitió estudiar el impacto de estas condiciones sobre variables fisiológicas de los pollos, como la mortalidad diaria, la cual no se registró en las primeras camadas. Simular mortalidad inducida por condiciones adversas proporciona un enfoque preventivo y permite calibrar el sistema para casos extremos poco frecuentes.

Para definir el método de generación de datos sintéticos, se revisaron tres modelos matemáticos empleados en investigaciones previas. El primero corresponde a los modelos estocásticos con ruido mixto, que combinan ruido gaussiano y perturbaciones discretas tipo Poisson para representar fluctuaciones naturales y eventos abruptos en sistemas biológicos (Zhou et al., 2021).

El segundo se basa en modelos mixtos con distribución contaminada, utilizados en genética animal para simular observaciones atípicas mediante una mezcla de distribuciones normales, lo cual permite representar escenarios con valores extremos o mediciones anómalas (Rosa et al., 2014).

Se consideraron los modelos híbridos bajo eventos climáticos extremos, donde se combinan variaciones gaussianas controladas con alteraciones abruptas en momentos críticos del ciclo, aplicados en simulaciones agrícolas con resultados más realistas (Gupta et al., 2024).

La comparación de estos tres enfoques se presenta en la Tabla 56.

Tabla 56*Comparativa entre modelos matemáticos*

Modelo	Distribución	Cadena de Markov	Mixto
	normal truncada		
Descripción	Generación aleatoria con media y desviación estándar, limitada a un rango.	Generación aleatoria con media y desviación estándar, limitada a un rango.	Valores diarios + promedio + ruido gaussiano + inserción manual de eventos críticos.
Control de realismo	Moderado	Alto	Alto
Posibilidad de insertar eventos extremos	Limitada	Implicita	Directa y controlada
Requiere datos históricos	No	Si	No
Capacidad de simular condiciones críticas	Limitada	Alta	Alta
Requiere programación compleja	No	Si	No

Fuente: Autoría

4.6.2 Modelo matemático aplicado

El modelo se fundamentó en la construcción de una serie temporal $X(t)$ que representa el valor de una variable ambiental (temperatura, humedad o concentración de amoníaco) en función del tiempo t . Cada valor de la serie se obtuvo a partir de la suma de tres componentes:

$$X(t) = \mu + G(t) + E(t)$$

donde:

μ es el valor medio o esperado de la variable, obtenido del promedio real de las camadas experimentales.

$G(t)$ representa la variabilidad natural del ambiente, modelada mediante un ruido gaussiano controlado con distribución normal de media cero y desviación estándar σ :

$E(t)$ corresponde al componente discreto de eventos extremos, aplicado únicamente en días específicos del ciclo de crianza, definido como:

$$E(t) = \begin{cases} \Delta_i, & \text{si } t = t_i \\ 0, & \text{en otro caso} \end{cases}$$

donde Δ_i representa la magnitud del evento extremo (por ejemplo, un aumento brusco de temperatura o una caída repentina de humedad) y t_i el día en que ocurre el evento.

4.6.2.1 Aplicación del modelo a las variables simuladas

La simulación se realizó para tres variables principales: temperatura, humedad y concentración de amoníaco, de acuerdo con los parámetros mostrados en la Tabla 57.

Tabla 57

Variables simuladas

Variable	Valor medio (μ)	Desviación estándar (σ)	Días de evento (t_i)	Magnitudes de evento (Δ_i)
Temperatura (°C)	27	2	{3, 6, 12}	{-6, +3}
Humedad (%)	65	5	{4, 9, 13}	{-25, +20}
Amoníaco (ppm)	22	3	{5, 8, 14}	{+10, +15}

Fuente: Autoría

En el código Python, cada componente se implementó de la siguiente manera: La expresión `np.random.normal ( $\mu$ ,  $\sigma$ )` genera el componente $\mu + G(t)$, es decir, el valor medio más una variación natural aleatoria. Los días definidos en t_i activan la adición del término $E(t) = \Delta_i$ mediante la instrucción `if día in [...]`, lo que introduce un cambio abrupto que simula una condición crítica del ambiente.

4.6.2.2 Clasificación del estado ambiental

Una vez obtenidos los valores diarios de temperatura, humedad y amoníaco, el sistema clasificó automáticamente el estado ambiental del galpón en tres categorías, utilizando los siguientes umbrales:

$$\text{Estado}(t) = \begin{cases} \text{Crítico,} & \text{si } T < 25 \text{ o } T > 30 \text{ o } H < 40 \text{ o } H > 80 \text{ o } NH_3 > 35 \\ \text{Estable,} & \text{si } 25 \leq T \leq 30, 50 \leq H \leq 75, NH_3 < 30 \\ \text{Alerta,} & \text{en otro caso} \end{cases}$$

Esta clasificación permitió evaluar el desempeño del sistema de control frente a distintas condiciones ambientales simuladas, facilitando la comparación con los resultados reales obtenidos durante las pruebas experimentales.

Gracias a esta estructura, el modelo mantuvo la coherencia estadística con los datos reales y permitió analizar el desempeño del sistema bajo condiciones estables, de alerta y de riesgo, ajustando así los límites de alarma y las estrategias de control de temperatura y ventilación.

4.6.3. Metodología de generación de datos simulados

Con el propósito de evaluar el comportamiento del sistema bajo condiciones ambientales adversas no observadas durante las pruebas reales, se generaron datos simulados correspondientes a cuatro nuevas camadas. Esta simulación permitió validar la robustez del sistema de control automático y su capacidad de reacción ante escenarios extremos.

4.6.3.1. Lenguaje y entorno de desarrollo

La generación de los datos fue realizada utilizando el lenguaje de programación Python, debido a su flexibilidad y facilidad para el procesamiento numérico y manejo de datos. Las simulaciones se ejecutaron utilizando las siguientes librerías:

- numpy – para generación de números aleatorios con distribución normal.
- random – para inserción de eventos discretos como mortalidad.
- pandas – para estructuración y exportación de las tablas simuladas.

Como se observa en la Figura 106, las librerías usadas y las variables que se usan en el código para generar datos.

Figura 106

Librerías y variables usadas en Python

```
import pandas as pd
import numpy as np
from random import randint, choice

# Parámetros
def simular_temperatura(dia):
    base = np.random.normal(27, 2)
    if dia in [3, 6, 12]:
        base += np.random.choice([-6, 7])
    return round(base, 2)

def simular_humedad(dia):
    base = np.random.normal(65, 5)
    if dia in [4, 9, 13]:
        base += np.random.choice([-25, 20])
    return round(max(min(base, 100), 25), 2)

def simular_amoniacos(dia):
    base = np.random.normal(22, 3)
    if dia in [5, 8, 14]:
        base += np.random.choice([10, 15])
    return round(base, 2)

def determinar_estado(temp, hum, amon):
    if temp < 23 or temp > 32 or hum < 40 or hum > 80 or amon > 35:
        return "Crítico"
    elif 23 <= temp <= 29 and 50 <= hum <= 75 and amon < 30:
        return "Estable"
    else:
        return "Alerta"

def simular_mortalidad(estado):
    if estado == "Crítico":
        return randint(1, 3)
    elif estado == "Alerta":
        return choice([0, 1])
    else:
        return 0
```

Fuente: Autoría

4.6.4 Resultados de la simulación: Camada 5

A continuación, se presentan los resultados simulados correspondientes a una quinta camada experimental generada mediante el modelo estadístico descrito previamente. Esta camada representa una secuencia de 15 días de monitoreo continuo, en los que se simularon eventos extremos de temperatura, humedad y concentración de amoníaco.

La Tabla 58 muestra el resumen diario de las variables monitoreadas, el estado ambiental clasificado por día, y la mortalidad registrada. Se evidencia que los días con condiciones críticas coinciden con eventos de mortalidad.

Tabla 58

Registro diario simulado – Camada 5

Día	Temperatura	Humedad	Amoníaco	Estado Ambiental	Mortalidad
1	29.06	66.55	20.24	Alerta	0
2	27.31	62.59	21.19	Estable	0
3	28.91	57.35	22.32	Estable	0
4	27.46	67.48	17.11	Estable	0
5	26.75	83.0	23.14	Crítico	1
6	29.35	71.26	37.24	Crítico	4
7	30.69	61.9	17.49	Alerta	0
8	26.96	74.32	20.76	Estable	0
9	29.69	80.18	21.01	Crítico	1
10	30.64	60.06	29.71	Alerta	0
11	30.49	60.62	23.65	Alerta	1
12	21.56	61.39	22.18	Crítico	1
13	29.14	60.3	28.42	Alerta	1
14	28.04	91.04	25.8	Crítico	1

15	27.35	75.75	22.11	Alerta	0
----	-------	-------	-------	--------	---

Fuente: Autoría

Este conjunto de datos permitió validar que, ante condiciones fuera de los parámetros óptimos definidos en la base de conocimiento del sistema, se generan respuestas de riesgo para las aves, lo que respalda la importancia del monitoreo constante y la respuesta automática del sistema.

4.6.5 Análisis de la mortalidad inducida por condiciones ambientales extremas

Para analizar de forma más precisa el impacto de las condiciones ambientales sobre la salud de las aves, se realizó un resumen de los días en que se presentó cada tipo de condición (estable, alerta o crítica), así como la mortalidad total asociada a cada una de ellas.

La Tabla 59 presenta que la totalidad de la mortalidad ocurrió en días clasificados como críticos o de alerta, mientras que en días estables no se registraron bajas.

Tabla 59

Resumen de mortalidad según clasificación ambiental – Camada 5

	Clasificación ambiental	Días con esta condición	Mortalidad total registrada
1	Estable	4	0
2	Alerta	6	2
3	Crítico	5	8

Fuente: Autoría

Este análisis permite concluir que la presencia de condiciones ambientales fuera de los rangos ideales puede asociarse directamente con la ocurrencia de mortalidad en los lotes de pollos. Por lo tanto, la anticipación y corrección automática de estos eventos mediante un sistema inteligente de control ambiental representa una medida fundamental para reducir riesgos en la producción avícola.

4.6.6 Resultados de la simulación: Camada 6

La Tabla 60 presenta los resultados generados mediante simulación para la sexta camada. Se observaron eventos con temperaturas cercanas al límite superior, así como variaciones en la humedad y amoníaco que en ciertas jornadas superaron los valores de referencia definidos como seguros.

Tabla 60

Registro diario simulado – Camada 6

Día	Temperatura	Humedad	Amoníaco	Estado Ambiental	Mortalidad
1	24.93	58.14	18.68	Estable	0
2	26.47	63.78	25.58	Estable	0
3	35.5	67.47	19.13	Crítico	2
4	26.6	60.94	21.84	Estable	0
5	28.70	60.94	25.11	Crítico	3
6	27.61	69.30	38.89	Crítico	4
7	30.71	71.29	24.64	Crítico	3
8	23.03	68.56	24.81	Estable	0
9	26.28	75.17	17.40	Alerta	0
10	28.7	65.85	35.62	Crítico	1
11	28.36	68.09	20.58	Estable	0
12	22.0	59.63	17.71	Crítico	1
13	27.23	64.19	34.88	Alerta	0
14	27.71	76.86	27.97	Alerta	1
15	24.85	63.9	19.6	Estable	0

Fuente: Autoría

Durante este ciclo, el sistema enfrentó condiciones críticas en varios días, lo que derivó en una mortalidad significativa en aquellos momentos donde la temperatura sobrepasó los 32 °C o la concentración de amoníaco se elevó abruptamente. Las variables se comportaron de manera inestable, lo cual se alineó con la lógica difusa previamente implementada, activando reglas que implicaban el uso de ventilación y calefactor en distintos momentos.

Este comportamiento resalta la importancia del monitoreo continuo y la intervención automática, sobre todo cuando se presentan oscilaciones ambientales intensas, como se simularon en esta camada.

4.6.7 Análisis de la mortalidad inducida por condiciones ambientales – Camada 6

El análisis de los registros diarios permitió identificar nuevamente que la mortalidad estuvo estrechamente asociada a los días con condiciones clasificadas como críticas. En aquellos días donde las variables se mantuvieron dentro de los umbrales, no se registraron pérdidas, validando que el ambiente estable contribuye a preservar la vida de las aves.

En las jornadas clasificadas como de alerta, la mortalidad fue baja o nula, mientras que los días críticos, con parámetros severamente alterados, coincidieron con los registros de mayor número de muertes.

La Tabla 61 presenta que la totalidad de la mortalidad ocurrió en días clasificados como críticos o de alerta, mientras que en días estables no se registraron bajas.

Tabla 61

Resumen de mortalidad según clasificación ambiental – Camada 6

	Clasificación ambiental	Días con esta condición	Mortalidad total registrada
1	Estable	6	0
2	Alerta	3	1
3	Crítico	6	14

Fuente: Autoría

Al igual que con la camada anterior, estos resultados evidencian la necesidad de que el sistema reaccione oportunamente ante anomalías, y también justifican el uso de simulaciones para probar escenarios que podrían no presentarse con frecuencia en las pruebas reales, pero que tienen un alto impacto productivo.

4.6.8 Resultados de la simulación: Camada 7

La Tabla 62 presenta los resultados generados mediante simulación para la séptima camada de pollos, con una duración de 15 días. En este escenario se incluyeron eventos ambientales adversos que permitieron observar el comportamiento del sistema automatizado ante situaciones críticas que no se evidenciaron en las camadas reales.

Entre los eventos más destacados se encuentran temperaturas elevadas superiores a los 33 °C, niveles de amoníaco por encima de 36 ppm y disminuciones bruscas de la humedad relativa.

Estas condiciones provocaron que el sistema clasifique varios días como críticos o en alerta, lo cual derivó en la activación de múltiples reglas difusas para ejecutar acciones de control correctivas. El sistema respondió automáticamente activando la ventilación, calefactores o mecanismos de humidificación/deshumidificación según corresponda.

Tabla 62

Registro diario simulado – Camada 7

Día	Temperatura	Humedad	Amoníaco	Estado Ambiental	Mortalidad
1	26.26	71.42	19.96	Estable	0
2	30.29	68.18	24.45	Alerta	0
3	19.98	60.57	21.14	Crítico	3
4	24.61	86.54	19.79	Crítico	2
5	25.96	68.52	38.01	Crítico	2
6	25.62	59.04	22.08	Estable	0

7	28.40	76.80	24.51	Alerta	1
8	24.26	60.48	32.03	Alerta	0
9	23.64	42.90	21.58	Alerta	0
10	29.12	52.68	22.48	Alerta	1
11	25.83	69.29	19.45	Estable	0
12	30.15	69.97	23.10	Crítico	3
13	27.90	44.35	21.19	Alerta	1
14	28.93	69.47	36.30	Crítico	3
15	28.33	69.61	20.57	Estable	0

Fuente: Autoría

La mortalidad registrada en esta camada refleja el impacto que tienen las condiciones ambientales extremas en el bienestar de las aves. En total, se registraron 16 muertes, concentradas en días con clasificación crítica y de alerta. Los días considerados estables no presentaron mortalidad, lo que evidencia que la lógica difusa empleada, cuando opera dentro de rangos seguros, permite preservar el estado de salud de las aves.

4.6.9 Análisis de la mortalidad inducida por condiciones ambientales – Camada 7

Se evidenció que, ante situaciones muy desfavorables sostenidas por más de un día, es posible que la respuesta del sistema no sea suficiente para evitar pérdidas, especialmente si los eventos ocurren de forma simultánea.

Tabla 63

Resumen de mortalidad según clasificación ambiental – Camada 7

	Clasificación ambiental	Días con esta condición	Mortalidad total registrada
1	Estable	4	0
2	Alerta	6	3

3	Crítico	5	13
---	---------	---	----

Fuente: Autoría

La mortalidad total acumulada en esta camada fue superior a la de camadas anteriores como se observa en la Tabla 63, confirmando la efectividad del modelo mixto para simular escenarios exigentes y poner a prueba la lógica difusa implementada.

4.6.10 Resultados de la simulación: Camada 8

La Tabla 64 resume el comportamiento ambiental simulado para la octava camada de pollos durante los 15 días del ciclo de crianza. Esta camada fue diseñada para presentar eventos extremos alternados y progresivos, con el objetivo de validar la capacidad del sistema de control difuso en escenarios complejos.

Durante los primeros tres días, las variables permanecieron dentro de rangos aceptables, clasificándose como estado estable. Sin embargo, a partir del cuarto día se simularon condiciones ambientales desfavorables, como humedad excesiva (91.13 %), temperaturas fuera de rango (por debajo de 20 °C o por encima de 32 °C) y concentraciones de amoníaco superiores a 36 ppm.

Tabla 64

Registro diario simulado – Camada 8

Día	Temperatura	Humedad	Amoníaco	Estado Ambiental	Mortalidad
1	25.73	67.32	17.87	Estable	0
2	23.90	69.86	17.65	Estable	0
3	23.46	61.83	24.84	Estable	0
4	28.35	91.13	20.78	Crítico	3
5	25.50	55.35	31.34	Alerta	0
6	18.08	53.58	23.49	Crítico	3
7	25.34	60.29	20.62	Estable	0

8	25.60	61.70	29.50	Estable	0
9	26.10	43.30	21.71	Alerta	0
10	23.03	68.17	20.41	Estable	0
11	29.89	64.07	21.06	Alerta	1
12	29.99	74.04	27.04	Crítico	1
13	30.67	34.80	18.83	Crítico	1
14	30.71	67.15	36.68	Crítico	2
15	24.41	57.86	21.93	Estable	0

Fuente: Autoría

En esta camada se observó una mortalidad total de 11 aves. La mortalidad se concentró principalmente en los días con clasificación crítica, lo cual valida la relación entre el entorno inadecuado y el deterioro del estado de salud de las aves. En contraste, durante los días considerados estables, no se registraron pérdidas.

Este resultado aporta evidencia adicional sobre la utilidad de generar pruebas de estrés controladas para validar la efectividad del sistema de control implementado, y permite identificar los límites operativos de la solución en condiciones ambientales más exigentes.

4.6.11 Análisis de la mortalidad inducida por condiciones ambientales – Camada 8

Se observa que, ante situaciones muy desfavorables sostenidas durante los 15 días de crianza, es posible que la respuesta del sistema no sea suficiente para evitar pérdidas, especialmente si los eventos ocurren de forma simultánea.

Tabla 65

Resumen de mortalidad según clasificación ambiental – Camada 8

	Clasificación ambiental	Días con esta condición	Mortalidad total registrada
1	Estable	7	0
2	Alerta	3	1

3	Crítico	5	10
---	---------	---	----

Fuente: Autoría

La mortalidad total acumulada en esta camada se mantiene dentro del promedio en comparación a la camada anterior, como se observa en la Tabla 65.

4.6.12 Análisis estadístico comparativo de datos reales y simulados

Con el propósito de verificar la correspondencia estadística entre los datos reales obtenidos durante la fase experimental y los datos generados mediante la simulación, se realizó un análisis descriptivo de las principales medidas de tendencia central y dispersión.

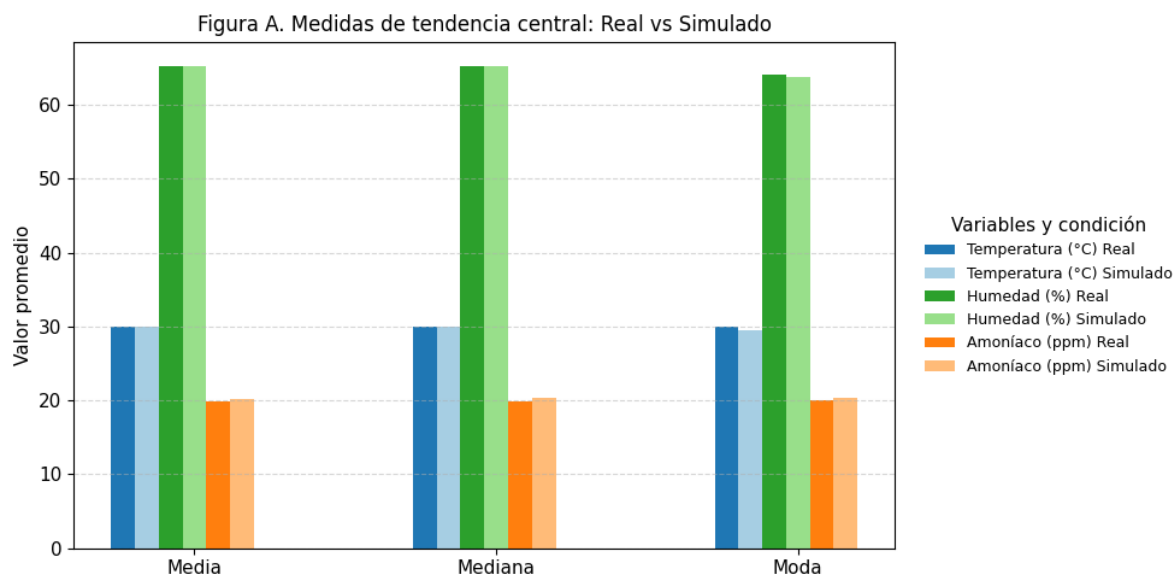
Este análisis permitió comprobar si la simulación basada en el modelo matemático mixto mantenía el mismo comportamiento estadístico que los registros medidos en campo, asegurando la validez del modelo y su capacidad de representar condiciones ambientales realistas dentro del galpón.

Se calcularon la media, mediana, moda y varianza para cada conjunto de datos, tanto reales como simulados, considerando las variables ambientales de mayor influencia en el bienestar de las aves (temperatura, humedad y concentración de amoníaco).

Las Figuras 107 y 108 comparan los principales estadísticos descriptivos obtenidos de los datos reales y simulados mediante el modelo matemático mixto. En la Figura 107, que muestra la media, mediana y moda de las variables temperatura, humedad y amoníaco, se observa una alta similitud entre ambos conjuntos. Esto confirma que la simulación conserva las tendencias centrales de las condiciones reales, reproduciendo fielmente el comportamiento ambiental del galpón.

Figura 107

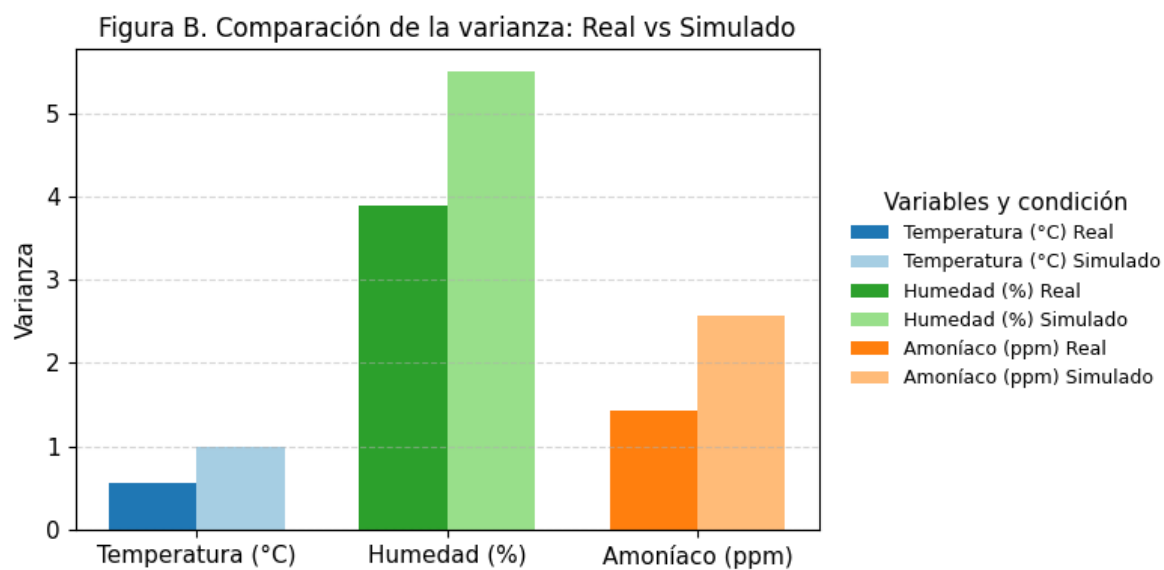
Medidas de tendencia central (media, mediana y moda) entre datos reales y simulados



Fuente: Autoría

Figura 108

Comparación de la varianza entre las tres variables ambientales medidas y simuladas.



Fuente: Autoría

La Figura 108 presenta la comparación de la varianza, evidenciando una ligera dispersión mayor en los datos simulados. Este aumento es coherente con la inyección controlada de ruido gaussiano y eventos extremos, cuyo propósito fue evaluar la respuesta del sistema ante escenarios críticos.

Ambos gráficos demuestran que el modelo mixto reproduce con precisión las condiciones reales y permite analizar el sistema bajo variaciones extremas sin perder coherencia estadística.

4.6.13 Cierre del apartado de simulación de camadas

La simulación de camadas permitió extender el análisis del sistema más allá de las condiciones experimentales observadas, incorporando escenarios con variaciones extremas en temperatura, humedad y concentración de amoníaco. Los resultados demostraron que el modelo matemático mixto reproduce de forma coherente la dinámica ambiental del galpón, manteniendo la estabilidad estadística observada en los datos reales.

La respuesta adecuada del sistema ante condiciones simuladas de estrés valida la efectividad del control inteligente y su capacidad de adaptación frente a eventos críticos. Este bloque confirmó que la metodología de simulación aplicada constituye una herramienta fiable para evaluar el rendimiento del sistema sin comprometer la integridad de las aves y optimizando el proceso de crianza bajo condiciones controladas.

4.7. Costo–beneficio del proyecto

4.7.1. Rentabilidad del sistema

La cantidad y distribución de sensores y actuadores se definieron en función de las dimensiones del galpón, las condiciones ambientales internas y las necesidades de control para garantizar la uniformidad del microclima.

El galpón cuenta con un área total de 140 m², y se determinó que la instalación de cuatro módulos de sensado sería suficiente para cubrir de forma representativa todo el espacio.

Cada módulo incluye un sensor de temperatura y humedad (DHT22), un sensor de amoníaco (MQ-137) y un microcontrolador ESP32 que transmite la información hacia la unidad central.

La elección de cuatro módulos se justificó por el radio de cobertura efectiva de los sensores empleados.

El DHT22 permite lecturas confiables de temperatura y humedad en un radio de aproximadamente 4 metros, mientras que el MQ-137 detecta concentraciones de amoníaco con precisión en áreas de hasta 20 a 25 m², dependiendo de la ventilación y la carga animal.

De esta manera, cada módulo cubre entre 30 y 35 m², logrando una supervisión completa de las condiciones ambientales del galpón con un leve solapamiento entre zonas, lo que evita puntos ciegos y asegura una detección oportuna de variaciones o anomalías.

Instalar un número mayor de sensores no aportaría un incremento significativo en la calidad de los datos, pero sí elevaría el costo del sistema; en cambio, reducirlos comprometería la representatividad de las mediciones.

Por otro lado, se implementaron seis actuadores distribuidos equitativamente a lo largo del galpón: dos ventiladores, dos calefactores y dos luminarias LED.

Los ventiladores se encargan de renovar el aire, reducir la acumulación de gases y mantener la temperatura dentro de los rangos recomendados; su cobertura efectiva es de aproximadamente 65 a 70 m² cada uno, lo que permite una ventilación uniforme en toda la estructura.

Los calefactores proporcionan el calor necesario durante la etapa inicial de crianza, ubicándose uno en cada extremo longitudinal del galpón para mantener un gradiente térmico equilibrado. Finalmente, las luminarias LED regulables permiten controlar el fotoperiodo y asegurar una distribución homogénea de la luz, lo cual influye directamente en el crecimiento y comportamiento de las aves.

La combinación de estos seis actuadores garantiza un control integral de los factores críticos: temperatura, ventilación e iluminación.

El número elegido responde a un equilibrio entre cobertura espacial, redundancia operativa y eficiencia energética.

Con esta configuración, el sistema mantiene condiciones estables, reduce el riesgo de mortalidad y mejora la eficiencia productiva, sin incurrir en gastos innecesarios por sobredimensionamiento de equipos.

4.7.2. Inversiones

La inversión comprende la adquisición de sensores, microcontroladores, unidad de procesamiento central, actuadores y accesorios eléctricos/electrónicos. En la Tabla 66 se puede observar los precios de los elementos que conforman el proyecto.

Tabla 66

Precios de componentes

Componente	Cantidad	Costo Unitario	Costo total
		USD	USD
Sensores de temperatura y humedad (DHT22)	4	15	60
Sensores de amoníaco (MQ-135)	4	25	100
ESP32 (controladores locales)	4	15	60
Raspberry Pi 5 (gateway central)	1	90	90
Fuente de alimentación + UPS	2	120	240
Actuadores	6	40	240
Accesorios eléctricos/electrónicos (cables, tableros, protecciones)	-	-	150
Costo de ingeniería	-	-	400
Total de inversión	-	-	1340

Fuente: Autoría

El costo de ingeniería se determinó conforme a los valores establecidos por el Ministerio de Trabajo del Ecuador en el Acuerdo Ministerial MDT-2025-001 (*Ministerio del Trabajo del Ecuador, 2025*), el cual regula las remuneraciones mínimas sectoriales aplicables a los distintos niveles ocupacionales del país. Dentro de este marco, el Grupo Ocupacional 1 – Profesionales de Ingeniería, Computación, Electrónica y Telecomunicaciones establece un salario básico mensual de USD 1 200,00, correspondiente a una jornada laboral de 8 horas diarias o 240 horas mensuales.

A partir de esta información oficial, el cálculo del costo de ingeniería se realizó considerando una dedicación de dos semanas de trabajo, equivalentes a 80 horas efectivas, que abarcan las actividades de diseño electrónico, desarrollo del software de control, integración del sistema IoT, implementación del modelo de inteligencia artificial, pruebas experimentales y documentación técnica.

El valor por hora se obtuvo como:

$$\text{Valor por hora} = \frac{1\,200}{240} = 5,00 \text{ USD/h}$$

Posteriormente, el costo total correspondiente a las dos semanas de trabajo se calculó como:

$$\text{Costo de ingeniería} = 80 \text{ h} \times 5,00 \text{ USD/h} = 400,00 \text{ USD}$$

Por tanto, el costo total de ingeniería asociado al desarrollo e implementación del sistema fue de USD 400,00, valor que se ajusta a la escala oficial de remuneraciones definida por el Ministerio de Trabajo del Ecuador (*Ministerio del Trabajo del Ecuador, 2025*). Este procedimiento garantiza que la estimación económica del proyecto se sustente en fuentes normativas vigentes, proporcionando transparencia y coherencia en el análisis costo–beneficio.

4.7.3. Mantenimiento del sistema

El mantenimiento se definió en función de las recomendaciones técnicas de los fabricantes y de las condiciones ambientales del galpón, caracterizadas por presencia de polvo, humedad y gases, se clasifica en dos tipos:

4.7.3.1 Mantenimiento preventivo

Calibración de sensores (cada 6 meses): asegura precisión en las mediciones. Limpieza de equipos electrónicos y ventiladores (mensual): evita obstrucciones y sobrecalentamiento. Revisión de conexiones eléctricas (trimestral): previene fallas por falsos contactos o humedad. Actualización de software y respaldos (semestral): mantiene la estabilidad y seguridad del sistema.

4.7.3.2 Mantenimiento correctivo

Reemplazo de sensores (cada 1–2 años): debido al desgaste natural de los DHT22 y MQ-135. Sustitución de componentes electrónicos defectuosos: placas, fuentes o actuadores en caso de falla. El costo anual de mantenimiento se estima en 120 USD, lo cual garantiza la operatividad continua del sistema sin afectar el proceso productivo.

4.7.4. Costo inicial

El costo inicial total asciende a 1.340 USD, correspondiente a la adquisición e instalación completa del sistema para un galpón de 20 × 7 m y 1.000 aves de capacidad. Este monto incluye todos los sensores, controladores, cableado, fuentes de energía y costos de configuración, garantizando una cobertura total del espacio con monitoreo y control automatizado en tiempo real.

4.7.5. Recuperación de la inversión

El cálculo de recuperación de la inversión se fundamentó en los ingresos netos generados por la venta de pollos producidos en el galpón considerado.

Cada quince días se comercializan aproximadamente 1.000 pollos a un precio promedio de 2,50 USD por unidad, lo que equivale a un ingreso bruto de 2.500 USD por ciclo productivo. Sin embargo, al considerar los costos operativos asociados (alimentación, energía, mano de obra y mantenimiento de equipos), la ganancia neta representa entre 30 % y 40 % del ingreso total, equivalente a 750 a 1.000 USD por ciclo. Bajo estas condiciones, la inversión inicial de 1.340 USD se recupera en un periodo de 1,5 a 2 ciclos productivos, es decir, aproximadamente entre 1,5 y 2 meses de operación continua.

A partir de ese punto, el sistema genera beneficios netos, considerando que los costos posteriores se limitan únicamente al mantenimiento preventivo anual.

Este tiempo de recuperación refleja un nivel de rentabilidad alto y sostenible, dado que la automatización contribuye al control ambiental, reduce la mortalidad y mejora el crecimiento uniforme de las aves, optimizando así el retorno de inversión en un periodo corto.

Conclusiones y Recomendaciones

Conclusiones

Se implementó satisfactoriamente un sistema automatizado basado en una arquitectura IoT, orientado al monitoreo y control ambiental de un galpón para la crianza de pollos. La integración de sensores, lógica difusa y herramientas web permitió optimizar las condiciones internas del galpón, mejorando la estabilidad de las variables críticas y facilitando el seguimiento del estado ambiental de las aves en tiempo real. El desarrollo se gestionó bajo los lineamientos de la metodología PMBOK, asegurando un proceso estructurado y controlado.

Los requerimientos definidos de hardware, software y lógica difusa se integraron de forma coherente en un sistema funcional y estable. La arquitectura propuesta demostró una alta interoperabilidad entre los módulos de sensado, procesamiento y visualización, garantizando la precisión en la adquisición de datos y la fiabilidad del control ambiental. Este resultado confirma la viabilidad técnica y la solidez del diseño, validando el cumplimiento de los requerimientos establecidos para la automatización del galpón.

Se identificaron deficiencias en el control ambiental bajo el método tradicional, particularmente en la variabilidad térmica y la acumulación de amoníaco. Durante las pruebas del sistema, se evidenció una mejora sustancial en la estabilidad de estos parámetros, lo cual se relacionó directamente con el comportamiento activo y saludable de las aves en cada camada, validando así el primer objetivo específico.

Se empleó un modelo de simulación basado en un enfoque mixto, que combinó ruido gaussiano y eventos extremos para reproducir condiciones ambientales críticas. Este modelo permitió complementar la validación del sistema, analizar su comportamiento ante escenarios de estrés térmico y variaciones abruptas de humedad, y fortalecer la evaluación del desempeño del sistema de control bajo situaciones no registradas en las camadas reales.

Las pruebas de campo permitieron comprobar el correcto funcionamiento del sistema y su adaptación a distintos escenarios, observándose una respuesta automática ante eventos críticos. El sistema demostró ser tecnológicamente viable, adaptable y rentable, logrando optimizar los recursos energéticos y elevar la productividad del galpón.

Recomendaciones

Se recomienda replicar esta solución en galpones de mayor tamaño, aplicando una distribución optimizada de los sensores, con el objetivo de validar su desempeño en entornos de producción a gran escala y evaluar su comportamiento frente a distintas condiciones climáticas, variaciones en la densidad de población aviar y escenarios operativos más exigentes.

Considerar la implementación de paneles solares u otras fuentes de energía limpia para asegurar la autonomía energética del sistema, especialmente en zonas rurales con acceso limitado a la red eléctrica.

Se recomienda incluir variables adicionales como el dióxido de carbono (CO₂), velocidad del aire o nivel de iluminación, para enriquecer el modelo de control y aumentar el confort animal.

Ampliar las funcionalidades del sistema incorporando nuevos módulos de control y monitoreo que complementen la lógica difusa, fortaleciendo la gestión integral del ambiente y el bienestar de las aves.

Actualizar periódicamente las reglas de decisión y los umbrales de control en función de los nuevos datos recolectados, tanto reales como simulados, con el fin de mantener el sistema adaptado a las condiciones específicas de cada entorno de producción.

REFERENCIAS

- López, J. (30 de Diciembre de 2023). *Medium*. Obtenido de Medium: <https://medium.com/@onlineformacion/qu%C3%A9-es-la-avicultura-y-en-qu%C3%A9-consiste-gu%C3%ADa-completa-5d98d74ae76b>
- Li, X., Zhang, J., Jin, W., & Liu, W. (2020). Key Technology Implementation of Poultry Breeding System for 5G Intelligent IOT. En 2020 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS 2020) (pp. 372–376). <https://doi.org/10.1109/TOCS50858.2020.9339747>
- Hewawasam, T., Gunasekara, J., Jayarathna, N., Indrawansha, Y., Rathnayake, S., & Panduwawala, P. (2023). Integrated AI-Based System for Comprehensive Poultry Management. En ICAC 2023 - 5th International Conference on Advancements in Computing: Technological Innovation for a Sustainable Economy, Proceedings (pp. 382–387). <https://doi.org/10.1109/ICAC60630.2023.10417233>
- B. Sugiarto, "El uso de la lógica difusa para la clasificación de datos en el nodo de sensores en el sistema de red inalámbrica de sensores y actuadores (WSAN)", Conferencia Internacional Makassar de Ingeniería Eléctrica e Informática (MICEEI) de 2014, Makassar, Indonesia, 2014, pp. 109-113, doi: 10.1109/MICEEI.2014.7067321.
- Sun, E., Xiao, Z., Yuan, F., Wang, Z., Ma, G., & Liu, J. (2023). Method of Classified Counting of Mixed Breeding Chickens Based on YOLOV5.

Soliman-Cuevas, H. (2024). ZamPen Chick Monitoring using Wireless Sensor Network. En 2024 Conference on Information Communications Technology and Society (ICTAS) (pp. 232–236). <https://doi.org/10.1109/ICTAS59620.2024.10507125>

S. Mavuri, K. Chavali y A. Kumar, "A study on imperative innovation eco system linkages to mapear el Objetivo de Desarrollo Sostenible 9", Conferencia Internacional sobre Digitalización (CIE) de 2019, Sharjah, Emiratos Árabes Unidos, 2019, pp. 142-147, doi: 10.1109/ICD47981.2019.9105845.

Nam, J. (2019). Impact of stressors on broiler chicken and development of it monitoring system for better production. Proceedings - 6th Annual Conference on Computational Science and Computational Intelligence, CSCI 2019, 1545–1546. <https://doi.org/10.1109/CSCI49370.2019.00289>

Amir, N. S., Abas, A. M. F. M., Azmi, N. A., Abidin, Z. Z., & Shafie, A. A. (2016). Chicken Farm Monitoring System. Proceedings - 6th International Conference on Computer and Communication Engineering: Innovative Technologies to Serve Humanity, ICCCE 2016, 132–137. <https://doi.org/10.1109/ICCCE.2016.39>

Sun, E., Xiao, Z., Yuan, F., Wang, Z., Ma, G., & Liu, J. (2023). Method of Classified Counting of Mixed Breeding Chickens Based on YOLOV5.

A. Faroqi, A. N. Utama, M. A. Ramdhani y E. Mulyana, "Diseño de un sistema de monitoreo de temperatura de jaula y base de microcontrolador en comedero automático de pollos", 2020

6th International Conference on Wireless and Telematics (ICWT), Yogyakarta, Indonesia, 2020, pp. 1-5, doi: 10.1109/ICWT50448.2020.9243636.

Xie, J., Li, M., & Chunxu, W. (2023). A valuable strategy for chicken welfare management: A review for chicken live weight monitoring. 2023 IEEE International Workshop on Metrology for Agriculture and Forestry, MetroAgriFor 2023 - Proceedings, 558–563.
<https://doi.org/10.1109/MetroAgriFor58484.2023.10424380>

T. -D. Wu, Z. -J. Chen, C. -C. Chang y H. -F. Wang, "Diseño de una red de sensores inalámbricos para la acuicultura en mar abierto-basada en el puente inalámbrico 802.11 ac y la tecnología LoRa™", Taller Internacional 2020 sobre Electromagnetismo: Aplicaciones y Competencia de Innovación Estudiantil (iWEM), Makung, Taiwán, 2020, pp. 1-2, doi: 10.1109/iWEM49354.2020.9237423.

C. K. Vinay, P. Kallam, J. Kallam y M. Rao, "Un sistema autónomo de alimentación de monitoreo de aves y registro de ingesta de alimentos hacia una rehabilitación efectiva", Conferencia Internacional de Sistemas IEEE 2022 (SysCon), Montreal, QC, Canadá, 2022, pp. 1-5, doi: 10.1109/SysCon53536.2022.9773792.

C. Gerhardson y C. -H. Min, "Diseño e implementación de una red de sensores inalámbricos IoT para el monitoreo de comederos para pájaros", 2022 IEEE 13th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canadá, 2022, pp. 0124-0128, doi: 10.1109/IEMCON56893.2022.9946473.

Bhuiyan, M. R., & Wree, P. (2023). Animal Behavior for Chicken Identification and Monitoring the Health Condition Using Computer Vision: A Systematic Review. *IEEE Access*, 11, 126601–126610. <https://doi.org/10.1109/ACCESS.2023.3331092>

Sánchez, M. A. (2021). Sector-avícola-Ecuador. *Universidad Técnica de Ambato*, 1–4.

S. Mojtaba Zabihinpour, M. K. A. A. T. S. H., A. S. A., O. B. (2020). A. Fuzzy Numbers.

Díaz, J. (2022). Sistema de monitoreo inteligente basado en tecnología IOT e inteligencia artificial aplicado a la crianza de alevines, en la parroquia el Playón de San Francisco perteneciente al cantón Sucumbíos provincia Sucumbíos. Ibarra: Universidad Técnica del Norte.

Freire, S. (2019). ANÁLISIS DE ERGONOMÍA FÍSICA EN LOS GALPONEROS DE UNA GRANJA AVÍCOLA. Ibarra: Universidad Técnica del Norte.

Caballero, J. (2019). ANÁLISIS DE ERGONOMÍA AMBIENTAL PARA LOS GALPONEROS EN EL ÁREA DE CRIANZA EN GRANJAS AVÍCOLAS. Ibarra: Universidad Técnica del Norte.

De la Torre, A. (2023). SISTEMA DE CONTROL PARA INCUBADORA ARTESANAL DE HUEVOS DE GALLINA. Ibarra: Universidad Técnica del Norte

Jaramillo, F. (2015). *MANUAL DE MANEJO PARA POLLO DE ENGORDE*. <https://www.solla.com/sites/default/files/productos/secciones/adjuntos/Manual%20De%20Manejo%20Para%20Pollo%20De%20Engorde.pdf>

C. Sharma y N. K. Gondhi, "Pila de protocolos de comunicación para sistemas de IoT restringidos", 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU), Bhimtal, India, 2018, pp. 1-6, doi: 10.1109/IoT-SIU.2018.8519904.

M. V. Sai Prasad, A. Sumalatha, K. S. Rani, C. Meenakshi, M. Nicy y D. C. Babu, "Solución de IoT basada en la nube para mejorar la gestión de las granjas avícolas", 2024 4ª Conferencia Internacional sobre Prácticas Innovadoras en Tecnología y Gestión (ICIPTM), Noida, India, 2024, pp. 1-5, doi: 10.1109/ICIPTM59628.2024.10563209.

Salazar, J., & Silvestre, Y. S. (2023). *INTERNET DE LAS COSAS*. <http://www.techpedia.eu>

ITALCOL. (2018). *Manual Práctico para la producción de Pollo de Engorde*.

Tania, L., & Isabel, C. (2015). *IOT, EL INTERNET DE LAS COSAS Y LA INNOVACIÓN DE SUS APLICACIONES*.

Rose, K., Eldridge, S., & Chapin, L. (2015). *La internet de las cosas - Una breve reseña*.

Cortés, C., Castañeda, J., & Bertoni, E. (2014). *EL "INTERNET DE LAS COSAS": MÁS INTERNET QUE OTRA COSA I*. <https://www.youtube.com/watch?v=kq8wcjQYW90>

Chanchí Golondrino, C. G., Ospina Alarcón, M. A., & Monroy Ríos, M. E. (2020). Arquitectura IoT para el desarrollo de sistemas de monitorización y análisis de variables fisiológicas en el área de asistencia médica. *Investigación e Innovación En Ingenierías*, 8(3), 1–13. <https://doi.org/10.17081/invinno.8.3.4699>

Tanasiev, V., Pătru, G. C., Rosner, D., Sava, G., Necula, H., & Badea, A. (2021). Enhancing environmental and energy monitoring of residential buildings through IoT. *Automation in Construction*, 126. <https://doi.org/10.1016/j.autcon.2021.103662>

Bharat Kumar, G. J. (2018). Internet of things (IOT) and cloud computing in hybrid organisations using monitoring systems. *Proceedings of the 3rd International Conference on Contemporary Computing and Informatics, IC3I 2018*, 348–351. <https://doi.org/10.1109/IC3I44769.2018.9007302>

Tonato Chuquimarca, C., & Sinche Maita, S. (2022). Análisis comparativo entre arquitecturas de sistemas IoT. *Revista de Investigación En Tecnologías de La Información*, 10(21), 55–70. <https://doi.org/10.36825/riti.10.21.006>

Bliznakoff, D., & Valle, D. (2015). *IOT: TECNOLOGÍAS, USOS, TENDENCIAS Y DESARROLLO FUTURO*.

Ramírez, A., & Ruales, A. (2021). Análisis comparativo de las tecnologías de comunicación para IoT en la monitorización de zonas agrícolas de las provincias de Cotopaxi y Tungurahua. *ESPE*.

Alfonzo, M. (2020). *UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL*.

Srodeck, J. (2020). *Manual de avicultura*. www.redbpa.org.ar

Yang, H., & Zhang, Y. (2018). Application of fuzzy logic in poultry farming: A review. *Journal of Poultry Science*, 55(2), 101-110. <https://doi.org/10.2141/jpsa.017017>

Boulanger, A., & Gosselin, C. (2016). Fuzzy logic in agriculture and food systems: A review. *Computers and Electronics in Agriculture*, 122, 1-12. <https://doi.org/10.1016/j.compag.2016.01.004>

Kaur, M., & Kaur, R. (2020). Fuzzy logic control for poultry housing systems. *International Journal of Fuzzy Logic and Intelligent Systems*, 20(2), 123-130. <https://doi.org/10.5573/IJFIS.2020.20.2.123>

Aviagen. (2018). *Manual de manejo del pollo de engorde*. https://aviagen.com/assets/Tech_Center/BB_Foreign_Language_Docs/Spanish_TechDocs/AA-BroilerHandbook2018-ES.pdf

Sethi, P., & Sarangi, S. R. (2017). Internet of Things: Architectures, Protocols, and Applications. *Journal of Electrical and Computer Engineering*, 2017, 3. <https://doi.org/10.1155/2017/9324035>

Vera, O. Z. (2021). *UAEH Lógica Difusa*. https://www.uaeh.edu.mx/docencia/P_Presentaciones/icbi/asignatura/logicaDifusa.pdf

Gustavo, M. I., & Juárez, E. (2017). *Facultad de Ciencias Exactas y Tecnología Universidad Nacional de Tucumán Lógica Difusa Parte I*.

Mejía, E. J. A. (2019). *Modelado y Sintonización de un Controlador PID De Un Robot Tipo Puma de Tres Grados de Libertad Utilizando Lógica Difusa*. 109.

Merma Gallardo, L. M. (2019). “Universidad Nacional De Cajamarca ,,. *Universidad Nacional de Cajamarca*, 310. <http://repositorio.unc.edu.pe/handle/UNC/1009>

Empresa Eléctrica Regional Norte S.A. (EMELNORTE). (2024). Consulta de valores. Recuperado de <https://www.emelnorte.com/docs/html/consultas.html>

Zhou, Q., Sun, F., & Borysenko, O. (2021). *Stochastic Analysis of Predator–Prey Models under Combined Gaussian and Poisson White Noise via Stochastic Averaging Method*. *Entropy*, 23(9), 1208. <https://doi.org/10.3390/e23091208>

Rosa, G. J. M., Valente, B. D., de los Campos, G., Wu, X., Gianola, D., & Silva, M. A. (2014). *Simulation study of linear mixed models with contaminated normal distribution in animal breeding*. *Journal of Animal Science*, 92(2), 395–406. <https://doi.org/10.2527/jas.2013-6759>

Gupta, R., Kim, D., & Lee, H. (2024). *Enhancing Crop Yield Simulations under Extreme Climate Events Using a Hybrid Model*. *EGU General Assembly Conference Abstracts*, 26, 12074. <https://meetingorganizer.copernicus.org/EGU25/EGU25-12074.html>

Ministerio del Trabajo del Ecuador. (2025). *Acuerdo Ministerial MDT-2025-001: Fijación de los salarios mínimos sectoriales para el año 2025*. Quito, Ecuador. Recuperado de <https://www.trabajo.gob.ec/>

ANEXOS

ANEXO A

Encuesta de Funcionamiento Tradicional

UNIVERSIDAD TÉCNICA DEL NORTE

ENCUESTA DIRIGIDA A ESPECIALISTAS EN EL ÁREA DE
AVICULTURA

PROYECTO: *“SISTEMA AUTOMATIZADO BASADO EN UNA ARQUITECTURA IoT PARA LA GESTIÓN Y MONITOREO DEL AMBIENTE DE UN GALPÓN DE CRIANZA DE POLLOS EN LA AVÍCOLA AVICAMP”*

Objetivo del proyecto: Desarrollar un sistema automatizado para un galpón de crianza de pollos basado en una arquitectura IoT, con el fin de optimizar las condiciones ambientales y mejorar el monitoreo del estado de las aves.

Objetivo de la encuesta: Recopilar información sobre el manejo ambiental en el galpón de crianza y funciona de manera tradicional.

Nombre del encuestado:

Fecha:

Nombre del encuestador:

- ¿Cuántos pollos se crían en promedio en el galpón?

a) 100 – 500

b) 501 – 1000

c) 1001 – 2000

- ¿Cuál es la duración promedio del ciclo de producción de pollos en el galpón antes de su comercialización?

a) Menos de 2 semanas

b) Entre 2 y 4 semanas

c) Entre 4 y 6 semanas

- ¿Cuáles son los principales factores ambientales que afectan la salud de los pollos en el galpón actualmente?

a) Temperatura

b) Calidad de aire

c) Humedad

d) Otro (especificar): _____

- ¿Ha identificado problemas de enfermedades en los pollos debido a condiciones ambientales?

a) Sí

b) No

Si la respuesta es "Sí", mencione las enfermedades más comunes:

Enfermedad 1: _____

Enfermedad 2: _____

Otras: _____

- ¿Cuánto tiempo invierte en monitorear las condiciones ambientales del galpón diariamente?

a) Menos de 1 hora

b) 1 – 6 horas

c) Más de 6 horas

- ¿Cómo controla actualmente los parámetros ambientales (temperatura, humedad, ventilación)?

a) Manualmente

b) Con dispositivos automatizados

c) Otro (especificar): _____

- ¿Considera que un sistema que automatice estos procesos podría ser útil para mejorar la gestión del galpón?

a) Sí

b) No

Elaborador por Santiago Chuquilla

ESTUDIANTE

ANEXO A

Encuesta de Funcionamiento Tradicional

UNIVERSIDAD TÉCNICA DEL NORTE

ENCUESTA DIRIGIDA A ESPECIALISTAS EN EL ÁREA DE AVICULTURA

PROYECTO: "SISTEMA AUTOMATIZADO BASADO EN UNA ARQUITECTURA IoT PARA LA GESTIÓN Y MONITOREO DEL AMBIENTE DE UN GALPÓN DE CRianza DE POLLOS EN LA AVÍCOLA AVICAMP"

Objetivo del proyecto: Desarrollar un sistema automatizado para un galpón de crianza de pollos basado en una arquitectura IoT, con el fin de optimizar las condiciones ambientales y mejorar el monitoreo del estado de las aves.

Objetivo de la encuesta: Recopilar información sobre el manejo ambiental en el galpón de crianza y funciona de manera tradicional.

Nombre del encuestado: Roberto Guagala **Fecha:** 05/12/2024

Nombre del encuestador: Santiago Cluquilla

- ¿Cuántos pollos se crían en promedio en el galpón?

- a) 100 – 500
- ☒ b) 501 – 1000
- c) 1001 – 2000

- ¿Cuál es la duración promedio del ciclo de producción de pollos en el galpón antes de su comercialización?

- a) Menos de 2 semanas
- ☒ b) Entre 2 y 4 semanas
- c) Entre 4 y 6 semanas

- ¿Cuáles son los principales factores ambientales que afectan la salud de los pollos en el galpón actualmente?

- ☒ a) Temperatura
- ☒ b) Calidad de aire
- c) Humedad

d) Otro (especificar): _____

- ¿Ha identificado problemas de enfermedades en los pollos debido a condiciones ambientales?

☒ a) Sí

b) No

Si la respuesta es "Sí", mencione las enfermedades más comunes:

Enfermedad 1: Bronquitis

Enfermedad 2: Ascaris

Otras: Newcastle

- ¿Cuánto tiempo invierte en monitorear las condiciones ambientales del galpón diariamente?

a) Menos de 1 hora

☒ b) 1 – 6 horas

c) Más de 6 horas

- ¿Cómo controla actualmente los parámetros ambientales (temperatura, humedad, ventilación)?

☒ a) Manualmente

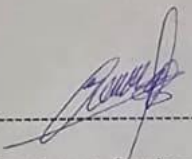
b) Con dispositivos automatizados

c) Otro (especificar): _____

- ¿Considera que un sistema que automatice estos procesos podría ser útil para mejorar la gestión del galpón?

☒ a) Sí

b) No



Elaborador por Santiago Chuquilla
ESTUDIANTE

ANEXO B***Encuesta para diseño del sistema automatizado*****UNIVERSIDAD TÉCNICA DEL NORTE****ENCUESTA DIRIGIDA A ESPECIALISTAS EN EL ÁREA DE
AVICULTURA**

PROYECTO: “*SISTEMA AUTOMATIZADO BASADO EN UNA ARQUITECTURA IoT PARA LA GESTIÓN Y MONITOREO DEL AMBIENTE DE UN GALPÓN DE CRIANZA DE POLLOS EN LA AVÍCOLA AVICAMP*”

Esta encuesta está dirigida a los especialistas en el área de avicultura, con el fin de obtener información sobre los requerimientos del usuario para la creación de un sistema automatizado de monitoreo en galpones de crianza de pollos.

Objetivo del proyecto: Desarrollar un sistema automatizado para un galpón de crianza de pollos basado en una arquitectura IoT, con el fin de optimizar las condiciones ambientales y mejorar el monitoreo del estado de las aves.

Objetivo de la encuesta: Obtener información específica para el diseño y funcionalidad del sistema automatizado.

Nombre del encuestado:

Fecha:

Nombre del encuestador:

- ¿Qué parámetros ambientales dentro del galpón considera que deben ser monitoreados en tiempo real?

a) Temperatura

b) Humedad

c) Calidad del aire

d) Otro (especificar): _____

- ¿Qué rango considera adecuado para los siguientes parámetros ambientales dentro del galpón?

Temperatura: _____

Humedad: _____

Nivel de amoníaco: _____

- ¿Le gustaría recibir alertas en caso de que los parámetros se desvíen del rango adecuado?

a) Sí

b) No

- ¿Qué medios considera más eficaces para recibir alertas?

- a) SMS
- b) Correo electrónico
- c) Notificaciones en una plataforma web
- d) Plataforma de mensajería Telegram
- e) Otro (especificar): _____

- ¿Preferiría que el sistema incluya un control remoto para ajustar parámetros como temperatura o iluminación dentro del galpón de pollos?

- a) Sí
- b) No

- ¿Qué grado de automatización preferiría para el suministro de alimentos?

- a) Completamente automatizado
- b) Manual, pero con monitoreo automatizado

- ¿Cómo prefiere visualizar los datos del sistema?

- a) En tiempo real mediante una plataforma web
- b) Reportes periódicos
- c) Reportes históricos
- d) Otro (especificar): _____

Elaborado por Santiago Chuquilla
ESTUDIANTE

ANEXO B

Encuesta para diseño del sistema automatizado

UNIVERSIDAD TÉCNICA DEL NORTE

ENCUESTA DIRIGIDA A ESPECIALISTAS EN EL ÁREA DE AVICULTURA

PROYECTO: "SISTEMA AUTOMATIZADO BASADO EN UNA ARQUITECTURA IoT PARA LA GESTIÓN Y MONITOREO DEL AMBIENTE DE UN GALPÓN DE CRianza DE POLLOS EN LA AVÍCOLA AVICAMP"

Esta encuesta está dirigida a los especialistas en el área de avicultura, con el fin de obtener información sobre los requerimientos del usuario para la creación de un sistema automatizado de monitoreo en galpones de crianza de pollos.

Objetivo del proyecto: Desarrollar un sistema automatizado para un galpón de crianza de pollos basado en una arquitectura IoT, con el fin de optimizar las condiciones ambientales y mejorar el monitoreo del estado de las aves.

Objetivo de la encuesta: Obtener información específica para el diseño y funcionalidad del sistema automatizado.

Nombre del encuestado: Roberto Guagala

Fecha: 05/12/2024

Nombre del encuestador: Santiago Cluquilla

- ¿Qué parámetros ambientales dentro del galpón considera que deben ser monitoreados en tiempo real?

☒ a) Temperatura

☒ b) Humedad

☒ c) Calidad del aire

d) Otro (especificar): _____

- ¿Qué rango considera adecuado para los siguientes parámetros ambientales dentro del galpón?

Temperatura: 30-35 °C

Humedad: 60 %

Nivel de amoniaco: < 25 ppm

- ¿Le gustaría recibir alertas en caso de que los parámetros se desvíen del rango adecuado?

☒ a) Sí

b) No

- ¿Qué medios considera más eficaces para recibir alertas?

a) SMS

b) Correo electrónico

c) Notificaciones en una plataforma web

☒ d) Plataforma de mensajería Telegram

e) Otro (especificar): _____

- ¿Preferiría que el sistema incluya un control remoto para ajustar parámetros como temperatura o iluminación dentro del galpón de pollos?

☒ a) Sí

b) No

- ¿Qué grado de automatización preferiría para el suministro de alimentos?

a) Completamente automatizado

☒ b) Manual, pero con monitoreo automatizado

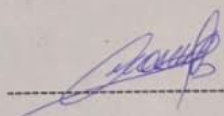
- ¿Cómo prefiere visualizar los datos del sistema?

☒ a) En tiempo real mediante una plataforma web

b) Reportes periódicos

c) Reportes históricos

d) Otro (especificar): _____



Elaborado por Santiago Chuquilla

ESTUDIANTE

ANEXO C

Análisis de encuesta ANEXO A

Pregunta	Opciones	Respuesta del encuestado
1. ¿Cuántos pollos se crían en promedio en el galpón?	a) 100–500 b) 501–1000 c) 1001–2000	b) 501–1000
2. Duración promedio del ciclo de producción	a) <2 semanas b) 2–4 semanas c) 4–6 semanas	b) 2–4 semanas
3. Principales factores ambientales que afectan la salud	a) Temperatura b) Calidad de aire c) Humedad d) Otro	a) Temperatura b) Calidad de aire
4. ¿Ha identificado problemas de enfermedades?	a) Sí b) No	a) Sí
5. Tiempo invertido en monitoreo diario	a) <1h b) 1–6h c) >6h	b) 1–6 horas
6. Método de control de parámetros ambientales	a) Manual b) Automatizado c) Otro	a) Manual
7. Considera útil un sistema automatizado	a) Sí b) No	a) Sí

La encuesta dirigida a especialistas en avicultura permitió recopilar información sobre el funcionamiento tradicional de los galpones de crianza de pollos. Las respuestas obtenidas reflejan que los galpones evaluados son de tamaño mediano (501–1000 pollos) y que los ciclos de producción duran entre 2 y 4 semanas. Los factores ambientales más críticos para la salud de los pollos son la temperatura y la calidad del aire, los cuales, si no se controlan adecuadamente, pueden generar problemas de enfermedades.

El monitoreo manual de las condiciones ambientales requiere entre 1 y 6 horas diarias, lo que evidencia la necesidad de sistemas más eficientes. Actualmente, la gestión ambiental se realiza de forma manual, sin automatización, aunque el encuestado considera que un sistema automatizado basado en IoT podría ser útil para mejorar la gestión del galpón y reducir riesgos de enfermedades.

ANEXO D

Análisis de la Encuesta ANEXO B

Pregunta	Opciones	Respuesta del encuestado
1. Parámetros ambientales que deben ser monitoreados en tiempo real	a) Temperatura b) Humedad c) Calidad del aire d) Otro	a) Temperatura b) Humedad c) Calidad del aire
2. Rangos adecuados de parámetros	Temperatura Humedad Nivel de amoníaco	Temperatura: 30–35 °C Humedad: 60 % Nivel de amoníaco: <25 ppm
3. Desea recibir alertas en caso de desviaciones	a) Sí b) No	a) Sí
4. Medios más eficaces para recibir alertas	a) SMS b) Correo electrónico c) Notificaciones en plataforma web d) Telegram e) Otro	d) Plataforma de mensajería Telegram
5. Preferencia de control remoto de parámetros	a) Sí b) No	a) Sí
6. Grado de automatización del suministro de alimentos	a) Completamente automatizado b) Manual, pero con monitoreo automatizado	b) Manual, pero con monitoreo automatizado

7. Forma de visualización de datos	a) En tiempo real mediante plataforma web b) Reportes periódicos c) Reportes históricos d) Otro	a) En tiempo real mediante plataforma web
------------------------------------	--	---

La encuesta dirigida a especialistas en avicultura permitió recopilar información para definir los requerimientos del sistema automatizado en galpones de crianza de pollos.

Los resultados muestran que los especialistas consideran importante monitorear en tiempo real temperatura, humedad y calidad del aire, y que los rangos adecuados para un ambiente saludable son: temperatura entre 30–35 °C, humedad alrededor del 60 % y nivel de amoníaco menor a 25 ppm.

El encuestado desea recibir alertas en caso de desviación de los parámetros, prefiriendo que estas se envíen a través de una plataforma de mensajería tipo Telegram.

Respecto al suministro de alimentos, se prefiere un sistema manual, pero con monitoreo automatizado, mientras que la visualización de los datos debería ser en tiempo real mediante una plataforma web, asegurando un seguimiento constante y acceso inmediato a la información crítica del galpón.

ANEXO E

ANÁLISIS DE REQUERIMIENTOS

Proyecto de titulación: “SISTEMA AUTOMATIZADO BASADO EN UNA ARQUITECTURA IOT PARA LA GESTIÓN Y MONITOREO DEL AMBIENTE DE UN GALPÓN DE CRIANZA DE POLLOS EN LA AVÍCOLA AVICAMP”

Objetivo de análisis: Entender de forma clara todos los requerimientos que debe contar para desarrollar un sistema IoT para un galpón avícola.

Artículos consultados:

1. An IoT-Based Smart Poultry Management System for Environmental Monitoring and Control.
2. Real-Time Poultry Farm Monitoring Using Wireless Sensor Networks.
3. IoT and Wireless Sensor Networks for Smart Agriculture: A Review.
4. Fuzzy Logic Applications for Environmental Control in Poultry Farms.
5. Design and Implementation of Smart Monitoring Systems for Poultry Houses.
6. Energy-Efficient IoT Sensor Networks for Livestock Monitoring.
7. Integration of Microcontrollers and Sensors for Automated Agricultural Systems.
8. Wireless Sensor Network-Based Monitoring of Environmental Conditions in Poultry Houses.
9. Real-Time Data Acquisition and Visualization in Smart Farming Systems.
10. Alerting Mechanisms in IoT-Based Livestock Management Systems.
11. Low-Power IoT Devices for Continuous Environmental Monitoring.
12. Web-Based Platforms for Agricultural Data Visualization.
13. Automation and Control in Poultry Houses: A Review.
14. Recovery Mechanisms in IoT Networks for Livestock Monitoring.

15. Microcontroller Selection and Hardware Considerations for IoT Systems in Agriculture.
16. Database Systems for Efficient Storage and Retrieval of Environmental Data.
17. Wireless Communication Protocols for Smart Poultry Farming.
18. Visualization Tools and Statistical Analysis in IoT-Based Agriculture.
19. Sustainable Energy Sources for IoT Devices in Rural Environments.
20. Security Measures in Wireless Sensor Networks for Agricultural Applications.

Acrónimo	Descripción
StSR	Requerimientos de los Stakeholders
SySR	Requerimientos iniciales del sistema
SRSR	Requerimientos de arquitectura

Requerimientos de Stakeholders						
#	Requerimientos		Prioridad			Justificación
			Alta	Media	Baja	
Requerimientos Operacionales						
StSR1	Conexión eléctrica estable.		X			Una conexión eléctrica confiable es fundamental para garantizar que todos los nodos y sensores funcionen sin interrupciones, evitando pérdida de datos críticos y fallos en el control ambiental. La estabilidad eléctrica es especialmente importante en sistemas IoT de monitoreo continuo, ya que cualquier interrupción puede

					comprometer la seguridad y productividad de las aves. (Alam et al., 2020)
StSR2	Operación continua las 24 horas del día, los 7 días de la semana.	X			La operación continua es esencial en sistemas de monitoreo de granjas, ya que las condiciones ambientales pueden cambiar rápidamente, afectando la salud de los pollos. El monitoreo 24/7 permite detectar desviaciones en tiempo real y tomar medidas correctivas inmediatas. (Zhang et al., 2019)
StSR3	Los nodos deben tener acceso a Internet de manera continua.	X			La conectividad constante asegura que los datos recopilados por los sensores se transmitan en tiempo real a la plataforma de supervisión, habilitando alertas y control remoto del sistema. La falta de conectividad puede retrasar la respuesta a eventos críticos. (Khan et al., 2021)
StSR4	La plataforma y monitor de datos debe estar operativa el mayor tiempo posible.	X			La disponibilidad de la plataforma es clave para que los usuarios puedan acceder a los datos históricos y actuales, realizar análisis estadísticos y tomar decisiones oportunas. La eficiencia en la visualización

					y monitoreo contribuye directamente a la seguridad y bienestar de los animales. (Chen et al., 2021)
Requerimientos de Usuarios					
StSR5	El sistema debe mostrar los datos de cada uno de los nodos en una plataforma y el monitor de datos.	X			La visualización individualizada de cada nodo permite un seguimiento detallado del ambiente en distintas áreas del galpón, facilitando la detección temprana de problemas específicos y optimizando la gestión ambiental. Esto mejora la eficiencia operativa y el bienestar de las aves. (Alam et al., 2020)
StSR6	El sistema debe notificar al usuario inmediatamente ante cualquier novedad detectada en el galpón.	X			Las alertas en tiempo real son esenciales para que el personal pueda actuar rápidamente ante condiciones ambientales adversas, minimizando riesgos de mortalidad y pérdidas económicas. La implementación de sistemas de notificación inmediata en granjas inteligentes ha demostrado mejorar la seguridad y productividad. (Zhang et al., 2019)
StSR7	El sistema debe tomar decisiones en cuanto a la	X			La capacidad de tomar decisiones automáticas

	regulación de las condiciones ambientales dentro del galpón.				permite que el sistema ajuste parámetros críticos como temperatura, humedad y ventilación sin intervención manual, garantizando un ambiente óptimo para el crecimiento de los pollos. La automatización de controles ambientales reduce errores humanos y mejora la eficiencia del sistema. (Chen et al., 2021)
--	--	--	--	--	---

Requerimientos del sistema					
#	Requerimientos	Prioridad			Justificación
		Alta	Media	Baja	
REQUERIMIENTOS DE USO					
SySR1	El bloque de sensores realizará lecturas de manera continua, con intervalos de diez minutos entre cada medición.	X			La recolección periódica de datos asegura un monitoreo constante del ambiente, permitiendo detectar rápidamente desviaciones en temperatura, humedad o calidad del aire, mejorando la salud y bienestar de los pollos. (Alam et al., 2020)
SySR2	El bloque de toma de datos transmitirá información de manera continua.	X			La transmisión continua de datos garantiza que la plataforma reciba información en tiempo real, habilitando alertas instantáneas y control automatizado del ambiente. (Zhang et al., 2019)

SySR3	El bloque de procesamiento recibirá los datos de manera continua.	X			El procesamiento en tiempo real permite analizar la información de los sensores de manera inmediata, asegurando respuestas rápidas ante condiciones críticas y evitando pérdidas de datos importantes. (Khan et al., 2021)
SySR4	Implementar un algoritmo de IA para monitorear condiciones ambientales.	X			La implementación de algoritmos de inteligencia artificial permite predecir y regular cambios en el ambiente de manera más eficiente que los métodos tradicionales, mejorando la precisión del control ambiental. (Chen et al., 2021)
SySR5	El sistema debe alertar al usuario por medio de la plataforma ante una novedad en el galpón.	X			Las alertas oportunas permiten al personal actuar rápidamente ante desviaciones, reduciendo riesgos de mortalidad y optimizando la producción. (Alam et al., 2020)
REQUERIMIENTOS DE RENDIMIENTO					
SySR6	La calibración de sensores se basa en las especificaciones del fabricante.	X			Una correcta calibración asegura la exactitud y confiabilidad de las mediciones, evitando decisiones incorrectas que puedan afectar el bienestar de las aves. (Zhang et al., 2019)

SySR7	El sistema debe contar con una base de datos.	X			El almacenamiento de datos históricos permite análisis estadísticos, generación de reportes y trazabilidad de las condiciones ambientales, mejorando la gestión del galpón. (Chen et al., 2021)
SySR8	El monitor de datos y plataforma deben funcionar la mayoría de tiempo para mostrar datos del sistema.	X			La disponibilidad de la plataforma asegura que los usuarios puedan supervisar y tomar decisiones basadas en información en tiempo real, manteniendo el control del ambiente de forma continua. (Alam et al., 2020)
SySR9	El sistema debe tener acceso a internet las 24 horas del día.	X			La conectividad continua permite la transmisión de datos en tiempo real y la recepción de alertas, asegurando un monitoreo y control remoto constante del sistema. (Khan et al., 2021)
REQUERIMIENTOS FÍSICOS					
SySR10	Los sensores deben tener una carcasa protectora.			X	La conectividad continua permite la transmisión de datos en tiempo real y la recepción de alertas, asegurando un monitoreo y control remoto constante del sistema. (Khan et al., 2021)
SySR11	Los nodos remotos deben estar instalados en sitios	X			La ubicación estratégica permite obtener datos representativos de todas las

	estratégicos para la toma de datos.				áreas del galpón, asegurando un control ambiental más preciso y uniforme. (Alam et al., 2020)
SySR12	El sistema debe tener un bajo consumo energético.			X	La ubicación estratégica permite obtener datos representativos de todas las áreas del galpón, asegurando un control ambiental más preciso y uniforme. (Alam et al., 2020)
REQUERIMIENTOS DE SEGURIDAD					
SySR13	El sistema debe tener seguridad física para prevenir manipulación de terceras personas.				La protección física previene el acceso no autorizado y manipulación de sensores o nodos, garantizando la integridad de los datos y el correcto funcionamiento del sistema. (Chen et al., 2021)
SySR14	El sistema de nodos tanto principal como nodos secundarios deben tener protección física.			X	La protección adicional asegura la continuidad de la operación frente a posibles accidentes o actos de manipulación, reduciendo riesgos de fallas del sistema. (Zhang et al., 2019)

Requerimientos de arquitectura					
#	Requerimientos	Prioridad			Justificación
		Alta	Media	Baja	
REQUERIMIENTOS DE DISEÑO					
SRSH1	El nodo de procesamiento debe			X	La centralización del nodo de procesamiento permite

	estar centralizado en el galpón, permitiendo una cobertura adecuada para todos los sensores ubicados estratégicamente en las diferentes áreas del galpón.				una gestión eficiente de los datos y simplifica la integración de los sensores distribuidos, asegurando cobertura completa del galpón y consistencia en la recolección de datos. (Alam et al., 2020)
SRSH2	Los sensores deben estar distribuidos de forma equitativa en el galpón para capturar condiciones representativas del ambiente en todas las áreas.		X		La distribución equitativa de sensores garantiza mediciones representativas, evitando zonas no controladas y asegurando que el control ambiental sea efectivo en todo el galpón. (Zhang et al., 2019)
SRSH3	Todos los sensores y equipos deben estar protegidos contra el polvo, la humedad y el amoníaco.		X		La protección física previene daños a los equipos y asegura la fiabilidad de las mediciones, prolongando la vida útil de los sensores en ambientes agresivos como galpones avícolas. (Khan et al., 2021)
SRSH4	Se debe implementar una plataforma de visualización que permita el acceso a los datos en tiempo real.	X			La visualización en tiempo real permite supervisión constante y facilita la toma de decisiones rápidas ante cambios críticos del ambiente. (Chen et al., 2021)

SRSH5	El sistema debe incluir un módulo de alerta que notifique al personal en caso de que algún parámetro ambiental esté fuera del rango establecido.	X			Las alertas tempranas permiten acciones correctivas inmediatas, protegiendo la salud de las aves y optimizando la productividad del galpón. (Alam et al., 2020)
SRSH6	Los dispositivos de monitoreo deben operar con bajo consumo de energía, para reducir la dependencia de la red eléctrica.			X	Minimizar el consumo de energía reduce la dependencia de la red eléctrica y permite integrar fuentes de energía sostenibles, asegurando operación continua. (Khan et al., 2021)
REQUERIMIENTOS LÓGICOS					
SRSH7	El sistema debe ser compatible con los sensores y ser capaz de integrarse con microcontroladores.	X			La compatibilidad garantiza que los sensores y nodos funcionen correctamente con la plataforma central, evitando errores de integración y pérdida de datos. (Zhang et al., 2019)
SRSH8	Se implementa reglas difusas para el monitoreo del sistema.	X			La lógica difusa permite un control adaptativo y más preciso de las condiciones ambientales, mejorando la regulación automática de temperatura, humedad y calidad del aire. (Chen et al., 2021)

SRSH9	El sistema debe contar con un mecanismo de recuperación para reanudar la operación tan pronto como se restablezca la conexión		X		La capacidad de recuperación asegura continuidad operativa tras fallos de comunicación, evitando pérdida de datos críticos y manteniendo la supervisión del galpón. (Alam et al., 2020)
SRSH10	El sistema debe ser capaz de activar dispositivos de respuesta, como ventiladores o calefactores, para restablecer las condiciones óptimas de manera automática.	X			La automatización de respuestas permite mantener condiciones ambientales ideales sin intervención manual, optimizando recursos y eficiencia operativa. (Zhang et al., 2019)
REQUERIMIENTOS DE HARDWARE					
SRSH11	El sistema debe contar con un microcontrolador con capacidad para procesar y transmitir los datos obtenidos de los sensores.	X			Un microcontrolador adecuado permite procesar los datos de manera rápida y confiable, garantizando la comunicación efectiva entre todos los componentes del sistema. (Alam et al., 2020)
SRSH12	El microcontrolador de toma de datos debe disponer de suficientes pines y puertos para conectar múltiples sensores.	X			La disponibilidad de pines y puertos permite la expansión del sistema y la conexión de todos los sensores necesarios sin limitaciones, aumentando la escalabilidad. (Khan et al., 2021)

SRSH13	El módulo de toma de datos debe ser compatible con la placa de procesamiento.	X			La compatibilidad garantiza la comunicación fluida y la correcta integración de hardware, evitando interrupciones en la transmisión de datos. (Chen et al., 2021)
REQUERIMIENTOS DE SOFTWARE					
SRSH14	El lenguaje de programación debe ser compatible con los sensores y la transmisión de datos.	X			La compatibilidad del software con los sensores asegura la correcta recolección y transmisión de datos, optimizando el rendimiento del sistema. (Zhang et al., 2019)
SRSH15	La base de datos debe garantizar un acceso rápido y eficiente.	X			El acceso rápido a datos históricos y en tiempo real permite análisis inmediatos y soporte para decisiones oportunas sobre el control ambiental. (Chen et al., 2021)
SRSH16	La interfaz de usuario debe desarrollarse en un entorno web, permitiendo el acceso a través de navegadores en computadoras y dispositivos móviles.	X			Una interfaz web accesible mejora la supervisión remota y permite que el personal monitoree y gestione el sistema desde cualquier ubicación. (Alam et al., 2020)
REQUERIMIENTOS DE VISUALIZACIÓN Y NOTIFICACIÓN					

SRSH17	La plataforma debe incluir gráficos estadísticos que muestren los parámetros obtenidos.		X		Una interfaz web accesible mejora la supervisión remota y permite que el personal monitoree y gestione el sistema desde cualquier ubicación. (Alam et al., 2020)
SRSH18	Las alertas por situaciones críticas se envían mediante la plataforma.	X			El envío de alertas a través de la plataforma asegura que el personal reciba notificaciones inmediatas sobre condiciones desfavorables, permitiendo intervenciones rápidas y evitando daños a las aves. (Alam et al., 2020)
SRSH19	El envío de notificación en estado crítico debe de realizarse por una aplicación de mensajería.	X			Las notificaciones mediante aplicaciones de mensajería garantizan comunicación instantánea y confiable con el personal responsable, asegurando respuestas oportunas ante eventos críticos. (Zhang et al., 2019)
REQUERIMIENTOS ELÉCTRICOS					
SRSH20	Alimentación entre 3.3-12V para sensores y microcontroladores	X			Una alimentación adecuada asegura el funcionamiento estable de todos los componentes electrónicos, evitando fallos que podrían comprometer la operación del sistema y la recolección de datos. (Alam et al., 2020)

SESH21	El sistema deberá operar utilizando una fuente de energía sustentable, garantizando un funcionamiento continuo y autónomo, incluso en condiciones de baja disponibilidad de recursos energéticos convencionales.		X		El uso de energía sustentable reduce la dependencia de la red eléctrica, asegurando la operación continua del sistema y promoviendo sostenibilidad ambiental en la granja. (Khan et al., 2021)
--------	--	--	---	--	--

ANEXO F

REPOSITORIO DE GITHUB

<https://github.com/SantiagoChuquilla/TESIS---CHUQUILLA-SANTIAGO>