



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES
DE COMUNICACIÓN

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERA EN ELECTRÓNICA Y REDES DE
COMUNICACIÓN

TEMA:

“SISTEMA DE ALERTA Y MONITOREO DE PRESENCIA DE
MATERIAL PARTICULADO EN EL AIRE, BASADO EN EL MODELO DE
GESTIÓN DE RED SNMP EN UNA RED WSN CON TECNOLOGÍA IEEE
802.11 PARA INVERNADEROS FLORÍCOLAS”

AUTOR: BENAVIDES CALIZ WILMER ENRIQUE

DIRECTOR: MSC. VÁSQUEZ AYALA CARLOS ALBERTO

IBARRA - ECUADOR

2022



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

BIBLIOTECA UNIVERSITARIA

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

IDENTIFICACIÓN DE LA OBRA

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

| DATOS DEL CONTACTO | | | |
|--------------------------------|---|------------------------------------|------------|
| CÉDULA DE IDENTIDAD: | 0401861828 | | |
| APELLIDOS Y NOMBRES: | Benavides Caliz Wilmer Enrique | | |
| DIRECCIÓN: | Via Otavalo- Selva alegre, Cdla. Los lagos | | |
| EMAIL: | wilmerbenavides@hotmail.es | | |
| TELÉFONO FIJO: | | TELÉFONO MÓVIL | 0968810854 |
| DATOS DE LA OBRA | | | |
| TÍTULO: | Sistema de alerta y monitoreo de presencia de material particulado en el aire, basado en el modelo de gestión de red SNMP en una red WSN con tecnología IEEE 802.11 para invernaderos florícolas. | | |
| AUTOR: | Benavides Caliz Wilmer Enrique | | |
| FECHA: | 25 de noviembre de 2022 | | |
| PROGRAMA: | PREGRADO <input checked="" type="checkbox"/> | POSTGRADO <input type="checkbox"/> | |
| TÍTULO POR EL QUE OPTA: | Ingeniería en Electrónica y Redes de Comunicación | | |
| ASESOR/DIRECTOR: | Ing. Carlos Alberto Vásquez Ayala, MSc | | |

CONSTANCIAS

El autor manifiesta que la obra de la presente autorización es original y se la realizo sin violar derechos de derechos de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que se 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 25 días del mes de noviembre de 2022.

EL AUTOR.

A handwritten signature in blue ink, appearing to read "Wilmer Benavides", is written over a horizontal dashed line.

Benavides Caliz Wilmer Enrique

C.I.:0401861828



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN

MAGISTER VÁSQUEZ AYALA CARLOS ALBERTO, MSc., DIRECTOR
DEL PRESENTE TRABAJO DE TITULACIÓN CERTIFICA:

Que, el presente trabajo de Titulación “SISTEMA DE ALERTA Y MONITOREO
DE PRESENCIA DE MATERIAL PARTICULADO EN EL AIRE, BASADO EN EL
MODELO DE GESTIÓN DE RED SNMP EN UNA RED WSN CON TECNOLOGÍA
IEEE 802.11 PARA INVERNADEROS FLORÍCOLAS” ha sido realizada en su totalidad
por: Sr. Wilmer Enrique Benavides Caliza, bajo mi supervisión:

Es todo en cuanto puedo certificar en honor de la verdad.

.....
Ing. Carlos Alberto Vásquez Ayala, MSc

Director de Tesis



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

DEDICATORIA

El presente trabajo de titulación lo dedico a todas las personas que formaron parte de esta etapa de formación académica.

Wilmer Benavides.

AGRADECIMIENTO

Agradezco en primer lugar a mi madre quien, con su fuerte convicción de proporcionarnos un mejor futuro, me supo apoyar en este proceso de formación, a mis hermanas y hermano.

A mi director MSc. Calos Vásquez quien con sus consejos y conocimiento me supo guiar con el desarrollo y culminación de este trabajo de titulación.

A mis colegas Chandi, Fredy, Romel, Max, Guano, Axel, Macas, Adonis, Vicente con quienes vivimos, día tras día este proceso de formación, con quienes he compartido buenos y malos momentos, y lo más importante nunca nos dimos por vencidos.

Y a Mayra y su familia quien me brindo su apoyo, de la cual siempre estaré agradecido por ese gesto hacia mi persona.

Wilmer Benavides

ÍNDICE DE CONTENIDO

| | |
|---|------|
| IDENTIFICACIÓN DE LA OBRA | i |
| CONSTANCIAS | ii |
| CERTIFICACIÓN..... | iii |
| DEDICATORIA..... | iv |
| AGRADECIMIENTO..... | v |
| ÍNDICE DE CONTENIDO..... | vi |
| ÍNDICE DE TABLAS..... | xi |
| ÍNDICE DE TABLAS..... | xvii |
| ÍNDICE DE ECUACIONES..... | xx |
| RESUMEN..... | xxi |
| ABSTRACT..... | xxii |
| CAPITULO I. Antecedentes..... | 1 |
| 1.1. Problemática..... | 1 |
| 1.2. Justificación..... | 2 |
| 1.3. Alcance..... | 3 |
| 1.4. Objetivos..... | 6 |
| 1.4.1. Objetivo General..... | 6 |
| 1.4.2. Objetivos Específicos..... | 6 |
| CAPITULO II. Justificación Teórica..... | 7 |
| 2.1. Redes Inalámbricas De Sensores..... | 7 |
| 2.1.1. Elementos De Una Red De Sensores Inalámbrico..... | 7 |
| 2.1.1.1. Nodo Sensor..... | 7 |
| 2.1.1.2. Gateway..... | 10 |
| 2.1.1.3. Estación Base..... | 11 |
| 2.1.1.4. Red Inalámbrica..... | 11 |
| 2.1.1.5. Topología Física De Una Red De Sensores Inalámbrica..... | 11 |
| 2.1.2. Factores Que Influyen En El Diseño De Las WSN..... | 12 |
| 2.1.2.1. Tolerancia A Fallos..... | 12 |
| 2.1.2.2. Coste De Producción..... | 13 |
| 2.1.2.3. Restricciones Del Hardware..... | 13 |
| 2.1.2.4. Consumo De Energía..... | 13 |
| 2.1.2.5. Costo..... | 13 |
| 2.1.2.6. Ambiente..... | 13 |
| 2.1.2.7. Conectividad y Topología..... | 14 |
| 2.1.3. Modelo Referencia OSI para WSN..... | 14 |
| 2.2. Estándar De Comunicación..... | 15 |

| | | |
|---|---|----|
| 2.2.1. | Estándar IEEE 802.11..... | 15 |
| 2.2.2. | Pila De Protocolos | 16 |
| 2.2.3. | Control De Acceso Al Medio (MAC) IEEE 802.11g..... | 17 |
| 2.2.3.1. | Protocolo CSMA/CA..... | 18 |
| 2.2.4. | Estructura De Trama 802.11g..... | 22 |
| 2.2.4.1. | MPDU Frame..... | 23 |
| 2.2.5. | Capa Física 802.11g | 27 |
| 2.2.5.1. | Estructura de la trama PLCP 802.11g..... | 30 |
| 2.2.5.2. | Subcapa ERP-OFDM PMD..... | 31 |
| 2.3. | Gestión De Red | 33 |
| 2.3.1. | Definición De Gestión De Red..... | 33 |
| 2.3.2. | Objetivo De La Gestión De Redes | 33 |
| 2.3.3. | Funciones De Gestión De Red..... | 34 |
| 2.3.4. | Modelos De Gestión De Redes..... | 35 |
| 2.3.4.1. | Modelo De Gestión internet..... | 36 |
| 2.3.4.2. | Arquitectura De Administración SNMPv2..... | 36 |
| 2.4. | Aprendizaje automático..... | 41 |
| 2.4.1. | Clasificación supervisada | 42 |
| 2.4.1.1. | Algoritmos De Clasificación a Base De Distancia | 42 |
| 2.4.1.2. | Rendimiento De Modelo De Clasificación | 43 |
| 2.5. | Contaminación Ambiental..... | 45 |
| 2.5.1. | Contaminación Del Aire..... | 46 |
| 2.5.1.1. | Plaguicidas | 47 |
| 2.5.1.2. | Efectos de plaguicidas en la salud humana..... | 47 |
| 2.5.2. | Indicadores De Calidad Del Aire..... | 49 |
| CAPITULO III. Desarrollo Experimental | | 51 |
| 3.1. | Metodología | 51 |
| 3.1.1. | Estándar ISO/IEC/IEEE 15288 | 51 |
| 3.2. | Análisis de la Situación Actual | 52 |
| 3.3. | Técnica De Investigación | 53 |
| 3.3.1. | Observación Directa En Invernadero Florícola..... | 53 |
| 3.3.2. | Entrevista..... | 54 |
| 3.4. | Definición de requerimientos | 55 |
| 3.4.1. | Descripción General Del Sistema..... | 55 |
| 3.4.2. | Definición De Requerimientos Del Sistema..... | 56 |
| 3.4.2.1. | Construcción De Atributos De Requerimientos | 56 |
| 3.4.2.2. | Nomenclatura De Requerimientos..... | 57 |

| | | |
|------------|---|----|
| 3.4.2.3. | Requerimientos De Stakeholders..... | 57 |
| 3.4.2.4. | Requerimientos Del Sistema..... | 59 |
| 3.4.2.5. | Requerimiento De Arquitectura..... | 60 |
| 3.5. | Elección de hardware y software | 63 |
| 3.5.1. | Elección De Hardware..... | 63 |
| 3.5.1.1. | Sistema Embebido Para Nodo sensor | 63 |
| 3.5.1.2. | Sistema Embebido Para Gateway | 65 |
| 3.5.1.3. | Sensores De Gas | 66 |
| 3.5.2. | Elección De Software | 71 |
| 3.5.2.1. | Lenguaje De Programación | 71 |
| 3.5.2.2. | Plataforma de desarrollo de código | 72 |
| 3.5.2.3. | Software Para El Monitoreo De Redes | 74 |
| 3.6. | Definición De Arquitectura..... | 75 |
| 3.6.1. | Bloque de adquisición de datos | 78 |
| 3.6.2. | Bloque de alimentación | 78 |
| 3.6.3. | Bloque de procesamiento y comunicación | 79 |
| 3.6.4. | Bloque de recepción y retransmisión | 79 |
| 3.6.5. | Bloque de gestión | 79 |
| 3.6.6. | Bloque de visualización..... | 80 |
| 3.7. | Definición De Diseño..... | 80 |
| 3.7.1. | Descripción General Del Funcionamiento Del Sistema Electrónico..... | 80 |
| 3.7.2. | Diagrama De Flujo Del Proceso Del Sistema | 81 |
| 3.7.3. | Diseño De Hardware | 83 |
| 3.7.3.1. | Bloque De Adquisición de datos..... | 83 |
| 3.7.3.1.1. | Diagrama de conexión del sensor MQ-2..... | 83 |
| 3.7.3.1.2. | Diagrama De Conexión Del Sensor MQ-7..... | 83 |
| 3.7.3.1.3. | Diagrama De Conexión Del Sensor MQ-135..... | 84 |
| 3.7.3.1.4. | Diagrama de conexión del sensor DTH11..... | 85 |
| 3.7.3.2. | Bloque De Alimentación | 85 |
| 3.7.3.3. | Bloque de procesamiento y comunicación | 86 |
| 3.7.3.3.1. | Diseño De La Placa Del Circuito Impreso (PCB) Para Nodo Sensor | |
| | 87 | |
| 3.7.3.4. | Bloque De Recepción Y Retransmisión | 88 |
| 3.7.3.4.1. | Diseño De La Placa Del Circuito Impreso (PCB) Para Gateway | 88 |
| 3.7.4. | Diseño De Software..... | 89 |
| 3.7.4.1. | Bloque De Adquisición De Datos..... | 89 |
| 3.7.4.1.1. | Diagrama De Flujo Del Bloque De Adquisición De Datos..... | 89 |

| | | |
|---|--|-----|
| 3.7.4.1.2. | Calibración De Sensores..... | 90 |
| 3.7.4.2. | Bloque De Procesamiento y Comunicación | 95 |
| 3.7.4.2.1. | Diagrama De Flujo De Bloque De Procesamiento Y Comunicación..... | 96 |
| 3.7.4.3. | Bloque De Recepción y Retransmisión | 98 |
| 3.7.4.3.1. | Diagrama De Flujo De Funcionamiento..... | 98 |
| 3.7.4.4. | Bloque De Gestión..... | 100 |
| 3.7.4.4.1. | Descripción General Del Modelo De Gestión De Red..... | 100 |
| 3.7.4.4.1. | Diagrama de flujo del funcionamiento | 100 |
| 3.7.4.4.2. | Base de datos de información..... | 102 |
| 3.7.4.4.3. | Agente SNMP..... | 104 |
| 3.7.4.4.4. | Gestor SNMP..... | 105 |
| 3.7.4.5. | Bloque De Visualización | 107 |
| 3.7.4.5.1. | Interfaz De Visualización y Gestión De Red WSN..... | 109 |
| 3.7.4.5.2. | Diagrama de flujo del bloque de visualización | 111 |
| 3.7.4.5.3. | Base De Datos | 112 |
| 3.7.5. | Diseño De La Red WSN..... | 113 |
| 3.7.5.1. | Direccionamiento..... | 116 |
| 3.8. | Implementación..... | 117 |
| 3.8.1. | Implementación De Hardware..... | 117 |
| 3.8.1.1. | Bloque De Alimentación | 117 |
| 3.8.1.2. | Bloque De Adquisición De Datos..... | 118 |
| 3.8.1.3. | Bloque De Procesamiento Y Comunicación..... | 118 |
| 3.8.1.4. | Bloque De Encaminamiento De Datos | 119 |
| 3.8.2. | Implementación De Software | 120 |
| 3.8.2.1. | Bloque De Adquisición De Datos..... | 120 |
| 3.8.2.2. | Bloque De Procesamiento Y Comunicación..... | 122 |
| 3.8.2.2.1. | Selección De Datos..... | 122 |
| 3.8.2.2.2. | Limpieza de datos | 123 |
| 3.8.2.2.3. | Minería de datos | 125 |
| 3.8.2.2.4. | Envío De Datos..... | 129 |
| 3.8.2.2.5. | Agente SNMP..... | 130 |
| 3.8.2.3. | Bloque De Recepción Y Retransmisión | 131 |
| 3.8.2.4. | Bloque De Gestión..... | 133 |
| 3.8.2.5. | Bloque de visualización | 135 |
| 3.8.3. | Implementación en el case al Gateway y nodos sensores | 136 |
| 3.9. | Integración Del Sistema | 138 |
| CAPITULO IV. Verificación Y Resultados..... | | 140 |

| | |
|---|-----|
| 4.1. Verificación y Validación Del Funcionamiento Del Sistema, Mediante Recomendaciones Especificadas En El Estándar ISO/IEC 30141:2018 | 140 |
| 4.1.1. Bloque De Alimentación | 140 |
| 4.1.1.1. Test eléctrico | 141 |
| 4.1.2. Bloque De Adquisición De Datos | 144 |
| 4.1.2.1. Test de lectura de datos | 144 |
| 4.1.3. Bloque De procesamiento y comunicación | 148 |
| 4.1.3.1. Test de rendimiento | 149 |
| 4.1.3.2. Test de comunicación | 153 |
| 4.1.4. Bloque De Recepción Y Retransmisión | 167 |
| 4.1.4.1. Test de comunicación | 168 |
| 4.1.5. Bloque De Gestión | 170 |
| 4.1.5.1. Test de gestión | 171 |
| 4.1.6. Bloque De Visualización | 173 |
| 4.1.6.1. Test de visualización de datos | 174 |
| 4.2. Resultados | 183 |
| CAPITULO V. Conclusiones Y Recomendaciones | 183 |
| 5.1. Conclusiones | 183 |
| 5.2. Recomendaciones | 193 |
| BIBLIOGRAFÍA | 194 |
| ANEXOS | 202 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1 <i>Elementos de una red de sensores inalámbrico</i> | 7 |
| Figura 2 <i>Componentes de hardware de un nodo sensor</i> | 8 |
| Figura 3 <i>Componentes de software de un nodo sensor</i> | 9 |
| Figura 4 Topología tipo estrella | 12 |
| Figura 5 <i>Estándar IEEE 802.11 y la pila de protocolos del modelo de referencia OSI</i> 16 | |
| Figura 6 Arquitectura 802.11 MAC | 17 |
| Figura 7 Emisión de una trama con CSMA/CA..... | 19 |
| Figura 8 Ejemplo del incremento exponencial de CW..... | 21 |
| Figura 9 <i>Modo de procedimiento de las tramas RTS y CTS</i> | 22 |
| Figura 10 Encapsulación de unidad de datos según la capa..... | 23 |
| Figura 11 <i>Formato de la trama de datos de 802.11</i> | 24 |
| Figura 12 Capas físicas de IEEE 802.11g | 28 |
| Figura 13 Throughput 802.11g..... | 29 |
| Figura 14 ERP-OFDM PPDU (802.11g) | 30 |
| Figura 15 Subcampo Rate de PLCP Header..... | 31 |
| Figura 16 Mapa de frecuencia de canal a 2,4 GHz..... | 32 |
| Figura 17 Modelo de gestión de red SNMP | 33 |
| Figura 18 <i>Modelos de gestión de redes</i> | 35 |
| Figura 19 <i>Arquitectura de administración SNMPv2</i> | 37 |
| Figura 20 <i>Estructura de identificadores de objetos</i> | 39 |
| Figura 21 PDU para SNMPv2..... | 40 |
| Figura 22 <i>Diagrama de dispersión del algoritmo K-NN</i> | 43 |
| Figura 23 <i>Matriz de errores</i> | 44 |
| Figura 24 Proceso del ciclo de vida del sistema..... | 52 |

| | |
|--|-----|
| Figura 25 Invernadero florícola mientras se realiza la fumigación..... | 54 |
| Figura 26 Entorno de Arduino IDE..... | 74 |
| Figura 27 Arquitectura general del sistema..... | 77 |
| Figura 28 Diagrama de flujo del proceso del sistema | 82 |
| Figura 29 Distribución y conexión de pines del sensor MQ-2..... | 83 |
| Figura 30 Distribución y conexión del sensor MQ-7 | 84 |
| Figura 31 Distribución y conexión del sensor MQ-135 | 84 |
| Figura 32 Distribución y conexión del sensor DTH11..... | 85 |
| Figura 33 Diagrama de distribución y conexión para la alimentación del circuito..... | 86 |
| Figura 34 Diseño Circuital del nodo sensor | 87 |
| Figura 35 Diseño de placa electrónica..... | 87 |
| Figura 36 Diagrama Circuital de Gateway | 88 |
| Figura 37 Diseño de placa electrónica para Gateway | 89 |
| Figura 38 Diagrama de flujo de bloque de sensores..... | 90 |
| Figura 39 Características de sensibilidad del sensor MQ-2 | 91 |
| Figura 40 Características de sensibilidad del sensor MQ-7 | 94 |
| Figura 41 Características de sensibilidad del sensor MQ-135 | 95 |
| Figura 42 Diagrama de flujo de bloque de procesamiento y comunicación | 97 |
| Figura 43 Diagrama de flujo para el bloque de recepción y retransmisión de datos..... | 99 |
| Figura 44 Diagrama de flujo del funcionamiento del gestor SNMP | 101 |
| Figura 45 Estructura MIB privada para el modelo de gestión de red WSN..... | 102 |
| Figura 46 Componentes de agente SNMP para los nodos SNMP..... | 104 |
| Figura 47 Componentes de gestor SNMP | 106 |
| Figura 48 Indicador de objetos | 108 |
| Figura 49 Valores de objetos tipo String..... | 108 |

| | |
|--|-----|
| Figura 50 Login de la herramienta de gestión | 109 |
| Figura 51 Ventana principal de la herramienta PRTG NETWORK MONITOR | 110 |
| Figura 52 Indicador tipo velocímetro de tiempo de respuesta de ping | 111 |
| Figura 53 Valores de objetos establecidos para cada NodoSensor | 111 |
| Figura 54 Diagrama de flujo del ciclo de funcionamiento del bloque de visualización | 112 |
| Figura 55 Creación de tablas para la base de datos | 113 |
| Figura 56 Vista del invernadero por medio de Google maps | 114 |
| Figura 57 Invernadero modelado en 3D | 114 |
| Figura 58 Ubicación del Gateway en el invernadero | 115 |
| Figura 59 Distribución de nodos en el invernadero..... | 116 |
| Figura 60 Distribución y conexión de bloque de alimentación | 117 |
| Figura 61 Conexión de líneas de sensor con la PCB | 118 |
| Figura 62 Anclaje del Node MCU esp32 con la placa | 119 |
| Figura 63 Anclaje del Node MCU esp8266 con la placa de circuito impreso | 119 |
| Figura 64 Selección de dos puntos en la tabla de sensibilidad del sensor MQ7 | 120 |
| Figura 65 Declaración de variables para calibración de sensores. | 121 |
| Figura 66 Métodos creados para la calibración, suavizado de la señal y calculo ppm del gas..... | 122 |
| Figura 67 Datos adquiridos por los sensores en diferentes escenarios..... | 123 |
| Figura 68 Resultados del filtrado CNN | 124 |
| Figura 69 Matriz resultante del proceso de limpieza..... | 125 |
| Figura 70 Resultado de aplicar algoritmo KNN con número de vecinos K=3..... | 126 |
| Figura 71 Asignación de variables para algoritmo KNN | 127 |
| Figura 72 Clasificación en base a distancia entre dos puntos. | 127 |
| Figura 73 Algoritmo para ordenar los elementos anteriormente clasificados | 128 |

| | |
|--|-----|
| Figura 74 Proceso de asignación de etiquetas | 128 |
| Figura 75 Asignación de objetos para cada uno de los sensores | 129 |
| Figura 76 Envío de datos a servidor SQL. | 130 |
| Figura 77 Creación de objetos de acuerdo con el RFC 1213 | 131 |
| Figura 78 Configuración de Gateway | 132 |
| Figura 79 Creación de objetos de acuerdo con el RFC 1213 para Agente SNMP Gateway | 133 |
| Figura 80 Configuración de alertas | 133 |
| Figura 81 Traps enviados desde el NODO 3..... | 134 |
| Figura 82 Alerta de fallo a través de un mensaje trap | 134 |
| Figura 83 Distribución de elementos dentro de PRTG NETWORK MONITOR..... | 135 |
| Figura 84 Valores guardados en base de datos..... | 136 |
| Figura 85 Nodo sensor adaptado en el case..... | 137 |
| Figura 86 Implementación del Gateway en el case | 137 |
| Figura 87 Nodos ubicados dentro del invernadero florícola. | 138 |
| Figura 88 Integración de Gateway en florícola | 139 |
| Figura 89 Línea temporal de voltaje..... | 143 |
| Figura 90 Comprobación de voltaje | 143 |
| Figura 91 Comunicación serial por puerto COM de nodo sensor y ordenador..... | 146 |
| Figura 92 Auto calibración de sensores..... | 146 |
| Figura 93 Impresión por serial plotter de valores obtenidos por sensores | 147 |
| Figura 94 Texto de confirmación de conexión entre nodo y el servidor MySQL..... | 147 |
| Figura 95 Verificación de datos guardados en base de datos MySQL..... | 148 |
| Figura 96 Impresión correcta de la etiqueta correspondiente al resultado del algoritmo KNN | 151 |

| | |
|--|-----|
| Figura 97 Tramas de autenticación y asociación..... | 155 |
| Figura 98 Relación entre tamaño de paquete [bytes] y latencia[ms]..... | 155 |
| Figura 99 Protocolo ICMP para verificación de latencia y bit rate..... | 156 |
| Figura 100 Capturas de paquetes en estación de gestión | 157 |
| Figura 101 Estructura de un mensaje get-requets encapsulado en un datagrama UDP. | 158 |
| Figura 102 Mensaje SNMP get-response | 159 |
| Figura 103 Cabeceras UDP | 160 |
| Figura 104 Cabecera IPv4 | 161 |
| Figura 105 Cabecera LLC | 163 |
| Figura 106 Cabecera MAC..... | 163 |
| Figura 107 Cabecera PLCP | 165 |
| Figura 108 Duración de la trama en el aire | 167 |
| Figura 109 Captura de tráfico entre nodo, Gateway y estación base | 169 |
| Figura 110 Scanning activo entre Nodo y Gateway | 170 |
| Figura 111 Captura de paquetes | 170 |
| Figura 112 Trafico SNMP generado entre nodos y estación de gestión | 173 |
| Figura 113 Verificación cambio de color de acuerdo con los límites establecidos..... | 173 |
| Figura 114 Red WSN activa..... | 176 |
| Figura 115 Valores de sensores desde APP | 176 |
| Figura 116 Recepción y visualización de mensajes TRAP..... | 177 |
| Figura 117 Notificación de SENSOR NO READ..... | 177 |
| Figura 118 Notificación SENSOR DOWN..... | 178 |
| Figura 119 Resúmenes de registros tipo histograma..... | 179 |
| Figura 120 Visualización de la base de datos MySQL | 179 |

| | |
|--|-----|
| Figura 121 Notificación por correo | 180 |
| Figura 122 Notificación de eventos del software de monitoreo..... | 181 |
| Figura 123 Visualización de notificaciones en dispositivo móvil..... | 182 |
| Figura 124 Red WSN activa monitorizada desde aplicación móvil y la interfaz de visualización para ordenadores..... | 184 |
| Figura 125 Fumigación del bloque con sistema de monitoreo activo | 185 |
| Figura 126 Niveles de concentración de material particulado normal. | 186 |
| Figura 127 Valores de NoX, PROPANO y CO en incremento durante la fumigación. | 186 |
| Figura 128 Valor de ICA..... | 187 |
| Figura 129 Incremento en la concentración y cambio de color en cada objeto. | 187 |
| Figura 130 Línea temporal de la variación de PROPANO durante la fumigación. | 188 |
| Figura 131 Línea temporal de la variación de CO durante la fumigación. | 189 |
| Figura 132 Línea temporal de la variación de NOx durante la fumigación. | 190 |

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 1 <i>Funcionalidades del protocolo de comunicación</i> | 14 |
| Tabla 2 Funcionalidades del protocolo de gestión | 15 |
| Tabla 3 Tipos de tramas de acuerdo con la combinación del campo tipo y subtipo | 24 |
| Tabla 4 <i>Interpretaciones de los campos Address 1, 2, 3 y 4 en función de To DS y From DS</i> | 26 |
| Tabla 5 Tipo y descripción de cada PDU snmpv2 | 40 |
| Tabla 6 <i>Efectos agudos producidos por los plaguicidas en el cuerpo humano</i> | 48 |
| Tabla 7 <i>Efectos crónicos producidos por los plaguicidas en el cuerpo humano</i> | 49 |
| Tabla 8 Límites numéricos de cada categoría de índices de calidad de aire en (ppm).. | 50 |
| Tabla 9 Abreviatura de requerimientos | 57 |
| Tabla 10 Lista de involucrados directos e indirectos..... | 57 |
| Tabla 11 Requerimientos de Stakeholders | 58 |
| Tabla 12 Requerimientos del sistema | 59 |
| Tabla 13 Requerimientos de arquitectura | 61 |
| Tabla 14 Valor referencial para requerimientos | 63 |
| Tabla 15 Elección de Hardware para Nodo Sensor | 64 |
| Tabla 16 Características técnicas del microcontrolador NODEMCU ESP32 | 64 |
| Tabla 17 Elección de Sistema embebido para Gateway | 65 |
| Tabla 18 Características técnicas NodeMCU esp8266..... | 66 |
| Tabla 19 Elección de sensores de gases | 67 |
| Tabla 20 Características técnicas del sensor MQ-2..... | 68 |
| Tabla 21 Características técnicas del sensor MQ-7 | 69 |
| Tabla 22 Características técnicas del sensor MQ-135..... | 69 |
| Tabla 23 Características técnicas del sensor DTH11 | 70 |

| | |
|--|-----|
| Tabla 24 Elección de lenguaje de programación | 71 |
| Tabla 25 Elección de la plataforma de desarrollo | 72 |
| Tabla 26 | 74 |
| Tabla 27 Cálculo de consumo corriente | 79 |
| Tabla 28 Objetos de subárbol de NodoWSN..... | 103 |
| Tabla 29 Traps | 103 |
| Tabla 30 Direccionamiento | 116 |
| Tabla 31 Valores límites de cada contaminante común en la atmosfera en partes por millón (PPM)..... | 124 |
| Tabla 32 Pruebas de numero óptimo de vecinos para algoritmo KNN | 125 |
| Tabla 33 Tabla de verificación de requerimientos de sistema y arquitectura para el bloque de alimenta..... | 141 |
| Tabla 34 Test Eléctrico para bloque de alimentación..... | 141 |
| Tabla 35 Tabla de verificación de requerimientos de sistema y arquitectura para el bloque de adquisición de datos | 144 |
| Tabla 36 Test de adquisición de datos | 145 |
| Tabla 37 Tabla de verificación de cumplimiento de requerimientos para bloque de procesamiento y comunicación | 149 |
| Tabla 38 Test de rendimiento | 149 |
| Tabla 39 Matriz de confusión | 151 |
| Tabla 40 Accuracy | 153 |
| Tabla 41 Test de comunicación | 153 |
| Tabla 42 Cálculo del tamaño total de datos..... | 159 |
| Tabla 43 Calculo cabecera UDP..... | 161 |
| Tabla 44 Calculo cabecera ipv4..... | 162 |

| | |
|--|-----|
| Tabla 45 Cálculo del tamaño total MPDU | 164 |
| Tabla 46 Calculo tamaño PPDU..... | 165 |
| Tabla 47 Tabla de verificación de cumplimiento de requerimientos para bloque de adquisición y retransmisión | 167 |
| Tabla 48 Test de comunicación | 168 |
| Tabla 49 Tabla de verificación de cumplimiento de requerimientos para bloque de gestión..... | 171 |
| Tabla 50 Test para el bloque de gestión | 171 |
| Tabla 51 Tabla de verificación de cumplimiento de requerimientos para bloque de visualización | 174 |
| Tabla 52 Test para el bloque de gestión | 174 |

ÍNDICE DE ECUACIONES

| | |
|---|----|
| Ecuación 1 Formula para el cálculo de distancia entre nodos..... | 12 |
| Ecuación 2 Valor DIF en los intervalos interframe | 20 |
| Ecuación 3 Valor Random backoff time en los intervalos interframe..... | 20 |
| Ecuación 4 Formula de distancia para determinación de vecinos cercanos en KNN.... | 43 |
| Ecuación 5 Ecuación para determinar la exactitud de un algoritmo | 44 |
| Ecuación 6 Ecuación para determinar la tasa de error de un algoritmo | 45 |
| Ecuación 7 Ecuación para determinar la precisión de un algoritmo | 45 |
| Ecuación 8 Ecuación para determinar el consumo de corriente de dispositivos | 78 |
| Ecuación 9 Ecuación para el cálculo de la resistencia superficial del sensor..... | 92 |
| Ecuación 10 Ecuación para obtener los parámetros de sensibilidad del gas en ppm | 93 |

RESUMEN

El presente proyecto describe el diseño e implementación de un sistema electrónico de alerta y monitoreo de material particulado producido por el uso de plaguicidas y fungicidas utilizados para el cultivo de rosas, el monitoreo se realiza mediante una red de sensores inalámbricos WSN bajo el protocolo de comunicación IEEE 802.11g. Además, posee un modelo de gestión de red denominado internet el cual permite el monitoreo del estado de los recursos de los nodos, gateway y estación base.

El sistema cuenta con tres sensores de la familia MQ que ayudan a medir la concentraciones de CO, propano y NOx y un sensor DHT que permite medir la temperatura y humedad, los datos recolectados por estos sensores son analizados y procesados mediante un algoritmo de clasificación supervisada KNN, el cual proporciona un índice de contaminación (NORMAL, ALERTA y EMERGENCIA) que ayudaran alertar el estado de contaminación que existe en el invernadero, estos datos son adaptados en mensaje GET RESPONSE del protocolo SNMP los mismo que son encapsulado en un datagrama UDP utilizando el puerto UDP 161 y enviados vía inalámbrica a la estación base. Cada sensor, Gateway y estación base cuenta con un agente SNMP que se encarga de proporcionar la información del dispositivo mediante mensajes GET RESPONSE y notificar sobre eventos inesperados que ocurren en los dispositivos mediante TRAPs, todos estos datos son almacenados en la base de datos del software gestión, los cuales se los puede visualizar en forma de histogramas para una mejor comprensión por el usuario.

El sistema cuenta con una interfaz gráfica y una aplicación móvil donde el usuario puede observar en tiempo real las concentraciones de material particulado en el invernadero y los estados de los dispositivos de la red, logrando así prevenir el ingreso temprano de los trabajadores, logrando reducir el contacto directo de estas sustancias con los encargados del cultivo de rosas.

ABSTRACT

This project describes the design and implementation of an electronic system for warning and monitoring of particulate matter produced by the use of pesticides and fungicides used for the cultivation of roses, monitoring is done through a wireless sensor network WSN under the IEEE 802.11g communication protocol. In addition, it has a network management model called internet which allows monitoring the status of the resources of the nodes, gateway and base station.

The system has four sensors of the MQ family that help to measure the concentrations of CO, propane and NO_x and a DHT sensor that allows measuring temperature and humidity, the data collected by these sensors are analyzed and processed by a supervised classification algorithm KNN, which provides a contamination index (NORMAL, ALERT and EMERGENCY) that will help to alert the state of contamination that exists in the greenhouse, these data are adapted in GET RESPONSE message of the SNMP protocol which are encapsulated in a UDP datagram using the UDP port 161 and sent via wireless to the base station. Each sensor, gateway and base station has an SNMP agent that is responsible for providing the device information through GET RESPONSE messages and notify about unexpected events that occur in the devices through TRAPs, all these data are stored in the management software database, which can be displayed in the form of histograms for better understanding by the user.

The system has a graphic interface and a mobile application where the user can observe in real time the concentrations of particulate matter in the greenhouse and the status of the network devices, thus preventing the early entry of workers and reducing the direct contact of these substances with those responsible for the cultivation of roses.

CAPITULO I. Antecedentes

1.1.Problemática

Según (Marlene, 2016), los trabajadores en el área de la floricultura están expuestos a riegos físicos, químicos y biológicos que provocan enfermedades laborales, donde la mayoría de las intoxicaciones son por plaguicidas del tipo carbamatos, organofosforados y organoclorado. Según el Instituto de Estadísticas del Ecuador el 52% de los floricultores presentan Hipertensión Arterial, 14% Anemia tóxica, 12% Leucopenia, 26% Hepatitis, 25% inestabilidad genética, se presenta signos clínicos de moderados – severos de toxicidad, debido a la constante exposición y la carencia de equipo de bioseguridad en el área de trabajo.

Los trabajadores no son los únicos afectados por las concentraciones de material particulado en el aire, ya que las diferentes especies de flores necesitan de un ambiente idóneo para su crecimiento y producción. También las redes de sensores WSN se ven afectadas por falta de un modelo de gestión que ayude a monitorizar los parámetros operativos de los dispositivos que ayuden a identificar incidencias en hardware y software en cada uno de los nodos desplegados.

En base a estos problemas se plantea realizar un sistema de alerta y monitoreo, dirigida a los interiores de los invernaderos florícolas, se plantea trabajar mediante una red de sensores inalámbricos WSN, los cuales permitirán la captura de las métricas para el cálculo del índice de calidad del aire, dichas métricas serán analizadas mediante un algoritmo de clasificación supervisada que permitirá generar la alerta si los índices de calidad del aire superan los valores permitidos. Además de la implementación de un modelo de gestión de red SNMP que permitirá mantener monitorizado cada nodo desplegado y así lograr identificar cada estado de la red de sensores como se lo hace tradicionalmente en una red de computadoras.

Tener un sistema de monitoreo de calidad del aire, ayuda a identificar la presencia de material particulado presente en el aire que a simple vista no se puede observar, además con la integración del modelo de gestión de red internet se podrá tener una gestión adecuada para garantizar la vida útil del sistema y a su vez presentar un sistema optimo que ayude a proporcionar alertas y permita al usuario tomar las medidas necesarias tanto para prevenir la exposición de los trabajadores, como para gestionar la adecuación del ambiente para las flores.

1.2.Justificación

La uno de los objetivos de organización mundial de la salud OMS (Organizacion Mundial de la Salud, 2017) con respecto a productos agroquímicos, es proteger la salud pública mediante el establecimiento de límites máximos de exposición y límites máximos de residuos de los plaguicidas en los alimentos y el agua, según una encuesta realizada por (Venegas, Cortés, Baltazar, & Serrano, 2016) el 23 % (nunca) y el 28% (a veces) de trabajadores no tienen el conocimiento de la categoría de toxicidad de una agroindustria florícola. Por lo cual le implementación de sistemas que ayuden a monitorizar el grado de contaminación y concentración de material particulado en el aire, es fundamental para precautelar la salud de las personas y a su vez reducir el tiempo de exposición directa a este tipo de contaminación.

Según (Córdoba & Buitrago, 2013) en los últimos años las aplicaciones con redes de sensores ha crecido exponencialmente, por lo cual el monitoreo de los diversos recursos de los sensores es indispensable para tener sistemas con alta disponibilidad, dicho esto la integración de modelos de gestión en redes de sensores es indispensable por la necesidad de tener sistemas monitorizados para garantizar su correcto funcionamiento, por lo cual se plante implantar el modelo de gestión de red internet o SNMP debido a que actualmente es un modelo de gestión utilizado universalmente y se está ampliando a todo

tipo de redes (no sólo TCP/IP) incluido OSI, por lo cual tener una red de sensores monitorizada permite no solo la adquisición de los datos del ambiente de aplicación, también permite tener datos del estado de los recursos de los equipos que componen la red de señores inalámbricos, para garantizar su correcto funcionamiento y una vida útil más prolongada.

Según (Novoa & Suarez, 2019) El estándar ISO/IEC 30141 es la primera arquitectura de referencia que tiene como objetivo armonizar el mundo de IoT; su aplicación permite que IoT sea más efectivo, interrumpible, seguro y resistente, de tal forma que se maximicen los beneficios y se reduzcan los riesgos. Dicho esto, la implementación de protocolos de monitoreo de red, proporcionar la generación de un sistema seguro reduciendo los riesgos de averías en hardware y errores de funcionamiento en software debido a los chequeos constantes implementados en un servidor de gestión.

Con el presente proyecto se plantea integrar el nuevo concepto de sociedad 5.0 según (Ogawa, 2018), en este nuevo concepto se trata de poner a la persona en el centro de las transformaciones tecnológicas, creando sistemas que no solo se enfoquen en la revolución industrial, al contrario, crear ambientes de armonía entre la revolución industrial y el ser humano, tratando de enfocarse en precautelar la salud de las personas.

1.3.Alcance

La realización del presente proyecto se basa en el desarrollo de un sistema de monitoreo de calidad del aire mediante el modelo de gestión de red SNMP y una red de sensores inalámbricos WSN, que permitirá recolectar información que ayude a generar un índice de niveles de concentración de diferentes gases y material particulado presente en el interior de invernaderos, para lograr prevenir y reducir la exposición de los

trabajadores a un ambiente peligroso y a su vez ayudar a identificar un ambiente idóneo para la correcta producción florícola.

Para el desarrollo del sistema se plantea aplicar el estándar ISO/IEC/IEEE 15288, el cual proporciona un marco común de descripciones de procesos para el ciclo de vida del sistema en fase de diseño, también proporciona procesos que admiten la definición, control y mejora de los procesos del ciclo de vida.

Para la etapa definición de requerimientos se debe realizar un estudio bibliográfico para identificar los diferentes parámetros físicos y químicos que permitirá identificar los niveles de concentración de material particulado en el aire, una vez establecidas los parámetros se procede a realizar una comparativa de los dispositivos electrónicos, que permitirán la recolección, procesamiento, envío y recepción de los datos.

Para la recolección de los parámetros de calidad del aire y estados de los recursos del sistema (hardware) y servicios (software), se realizará la integración de un modelo de gestión de red internet o SNMP. Basado en su arquitectura se tiene una estación de gestión la cual posee una interfaz de usuario con la capacidad de convertir las solicitudes del usuario a peticiones de monitorización y control de elementos remotos, creando una base de datos con información de las MIBs de los elementos de red gestionados, en las MIBs se define las variables usadas por el protocolo SNMP para supervisar y controlar los componentes de una red y están compuestas por una serie de objetos que representan cada recurso gestionado, cada objeto manejado en un MIB posee un identificador de objeto único (OID) que sirve para nombrarlo, es aquí donde se implementaran MIBs adecuadas para lograr la adquisición de datos de objetos (sensores) y de los recursos del sistema, para enlazar la estación de gestión y los dispositivos gestionados(nodos) se utiliza el

protocolo simple de monitoreo de red snmp, este protocolo es un protocolo de capa aplicación de tipo datagrama (UDP) permite a la estación de gestión puede solicitar el valor de OID (GET), ejecutar una acción o configurar un valor (SET) y se puede establecer notificaciones espontaneas de incidente(TRAP).

Para la identificación del índice de calidad de aire, se plantea usar un algoritmo de clasificación supervisada, que a partir de los datos recolectados permite realizar la comparación de los datos obtenidos con datos conocidos establecidos en un matriz de entrenamiento y de esta manera generar una categorización de acuerdo con el grado de contaminación y peligrosidad del aire.

Para el envío de los datos desde los nodos al Gateway, se utilizará tecnología de radio frecuencia IEEE 802.11 con una frecuencia de 2.4 GHz, celda de cobertura de 75 metros y como protocolo de control de acceso al medio se utilizará CSMA/CA.

El servidor de gestión se encargará de recibir los datos y mostrar al usuario mediante una interfaz de página web, donde el usuario tendrá acceso mediante credenciales de autenticación para acceder a los servicios del servidor de gestión, donde podrá agregar host, mibs, servicios, contactos y revisar los estados de los recursos de los nodos y los parámetros e índice de calidad del aire. Los datos recibidos se almacenarán en una base de datos que servirán como apoyo para identificar un ambiente idóneo para la producción florícola. Cuando el índice de calidad del aire supere los límites permitidos se generar alertas mediante notificaciones se usará un servidor de mensajes instantáneo.

Para garantizar la seguridad, confianza y una base tecnológica con medidas y sistemas robustos se plantea como solución el uso del estándar ISO/IEC 30141:2018, ya que dicho estándar proporciona un marco común para el desarrollo de aplicaciones IoT, que permite desarrollar sistemas fiables, seguros, protegidos, respetuosos con la

privacidad y capaces de afrontar interrupciones debidas a catástrofes naturales o ciberataques.

1.4.Objetivos

1.4.1. Objetivo General

Diseñar un sistema embebido de alerta y monitoreo mediante una red de sensores inalámbricos WSN basado en el modelo de gestión de red SNMP, para la recolección, clasificación, análisis y reporte de la información que permita la identificación de los índices de contaminación en el aire.

1.4.2. Objetivos Específicos

Realizar un estudio bibliográfico relacionado a redes WSN, comunicación inalámbrica y el modelo de gestión de red SNMP que permita establecer los requerimientos necesarios para el desarrollo de sistema.

Diseñar el sistema de monitoreo, mediante tecnologías inalámbricas y el modelo de gestión de red internet, bajo la metodología propuesta por el estándar ISO/IEC/IEEE 15288.

Procesar la información recolectada por el sistema de monitoreo para encontrar patrones que permitan generar el índice de calidad del aire, mediante algoritmos de clasificación supervisada.

Evaluar el sistema de monitoreo mediante pruebas de verificación y validación especificadas en el estándar ISO/IEC 30141:2018

CAPITULO II. Marco Teórica

2.1.Redes Inalámbricas De Sensores

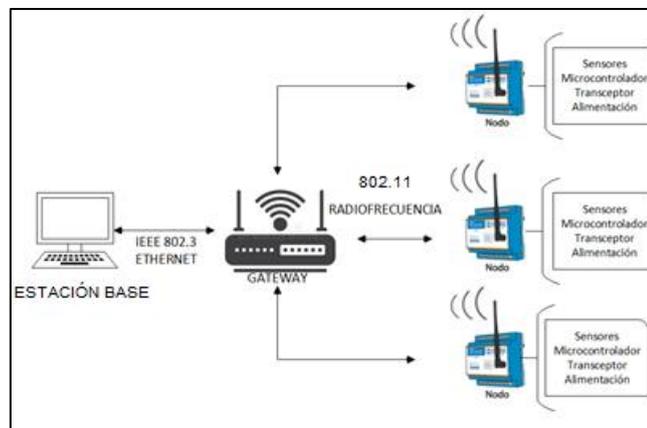
Según (Aakvaag & Frey, 2016), un sistema WSN (Wireless Sensor Network) de sensores inalámbricos es una red con numerosos dispositivos distribuidos espacialmente, que utilizan sensores para controlar diversas condiciones en distintos puntos, entre ellas la temperatura, el sonido, la vibración, la presión y movimiento o los contaminantes.

2.1.1. Elementos De Una Red De Sensores Inalámbrico

Una red de sensores inalámbricos está formada de diferentes que elementos que trabajan conjuntamente para la captura, procesamiento y envío de datos hacia la estación base. En la Figura 1 se muestra los componentes básicos que forman una red de sensores WSN.

Figura 1

Elementos de una red de sensores inalámbrico



Fuente: Elaborada por el autor

2.1.1.1.Nodo Sensor

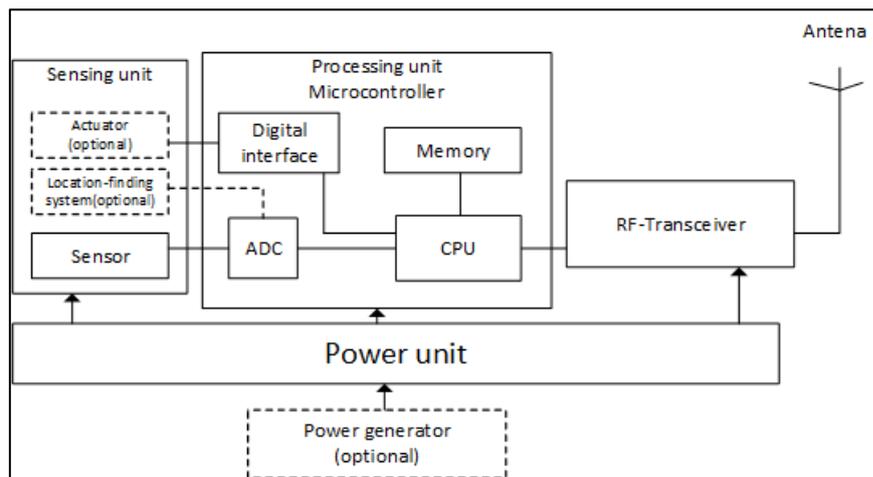
El nodo sensor está compuesto de componentes de hardware y software que se describen a continuación:

Hardware

Un nodo sensor básicamente está constituido de cuatro unidades indispensables para su funcionamiento, en la Figura 2 se puede observar la arquitectura de todos los componentes de hardware de un nodo sensor de una WSN.

Figura 2

Componentes de hardware de un nodo sensor



Fuente: adaptado de Garbarino, J. (2011), Protocolos para redes inalámbricas de sensores.

Sensor: es un dispositivo electrónico conocidos como transductores que en su estructura hardware está diseñado para transformar de una condición física o química a señales eléctricas medibles, es decir los señores cuentan con la capacidad de detectar un cambio en su entorno y traducirlo a una señal analógica o digital (Aakvaag & Frey, 2016). Algunos de los parámetros que se puede obtener mediante el uso de sensores son: temperatura, movimiento y presión, aceleración, luminosidad, humedad, ultrasonido, caudal, proximidad, CO2

Unidad de procesamiento (Microcontrolador): un microcontrolador es básicamente el cerebro del nodo, es un ordenador en un chip que posee la capacidad necesaria para controlar y procesar el flujo de datos obtenidos por los sensores,

autosuficiente y con una unidad de almacenamiento donde se aloja el sistema operativo y el código que indica las acciones a realizar (Aakvaag & Frey, 2016).

Unidad de transmisión y recepción (transceiver): es el encargado de la recepción y transmisión de datos en el nodo, utilizando señales de radiofrecuencia por un canal inalámbrico (Aakvaag & Frey, 2016).

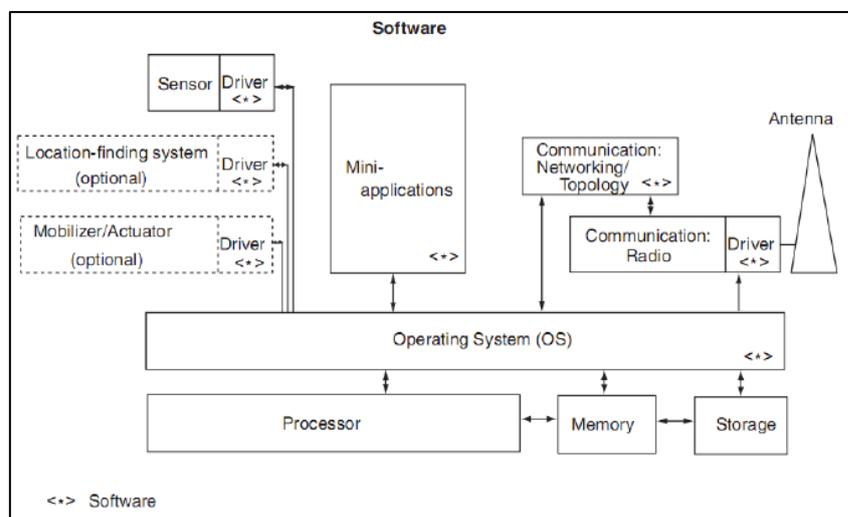
Unidad de energía: es el sistema encargado de proporcionar la energía necesaria para el funcionamiento correcto del nodo sensor.

Software

La Figura 3 muestra la arquitectura de componentes de software de un nodo sensor necesario para el correcto funcionamiento.

Figura 3

Componentes de software de un nodo sensor



Fuente: adaptado de Garbarino, J. (2011). Protocolos para redes inalámbricas de sensores.

Los nodos sensores básicos generalmente trabajan con cinco subsistemas de software, a continuación, se menciona cada uno de ellos:

Sistema operativo: es el microcódigo que usualmente se le denomina como middleware, es microcódigo común de microcontroladores utilizado por todos los programas de más alto nivel (Garbarino, 2017).

Controladores de sensores: módulos de software que administra las funciones de las unidades de sensores u otros componentes que forman parte del nodo sensor.

Procesadores de comunicación: según (Garbarino, 2017), los procesadores de comunicación son módulos que administran funciones de comunicación, encaminamiento, almacenamiento y reenvío de paquetes, mantenimiento de la topología de red, control de acceso al medio, cifrado, corrección de errores, etc.

Controladores de comunicación: son módulos que gestionan las tareas de utilización de los enlaces de comunicación por radio, la modulación, sincronización (Garbarino, 2017).

Mini-aplicaciones de proceso de datos: son las aplicaciones desarrolladas para el procesamiento de los datos, por ejemplo, algoritmos de clasificación supervisada (Garbarino, 2017).

2.1.1.2. Gateway

La pasarela o puerta de enlace es un dispositivo que sirve como pasarela o puente de comunicación de datos entre los nodos sensores y la estación base, la puerta de enlace recibe la información de los nodos que usan una arquitectura y un protocolo de comunicación inicial en este caso red inalámbrica 802.11, los interconecta y traduce a la arquitectura y protocolo de comunicación de la red destino en este caso red cableada 802.3 (Aakvaag & Frey, 2016).

2.1.1.3. Estación Base

La estación base es la encargada de la recolección de información generada por la red de sensores y encaminada por la puerta de enlace hasta el, normalmente es basado en un ordenador donde el administrador de la red con la información recolectada puede almacenarla, procesarla y la muestra al usuario final (Aakvaag & Frey, 2016).

2.1.1.4. Red Inalámbrica

La red inalámbrica es el medio de comunicación que utiliza la red de sensores para enviar y recibir los datos, utilizando ondas de radio en un rango de frecuencias especificadas por las bandas de radiofrecuencias industriales, científicas y médicas (ISM) estipuladas por la ITU (Unión Internacional de Telecomunicaciones) (Martínez & Meré, 2014).

2.1.1.5. Topología Física De Una Red De Sensores Inalámbrica

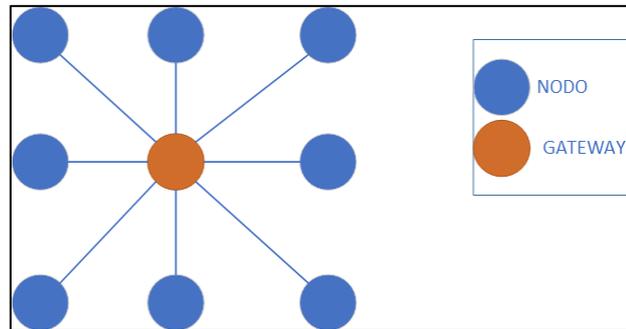
Según (Martínez & Meré, 2014), una topología es una descripción esquemática de la distribución de los componentes de la red, incluyendo los nodos, Gateway y líneas de conexión, a continuación, se muestra la topología física para una WSN:

Topología Tipo Estrella

En topología en estrella todos los nodos que conforman la red se conectan al Gateway, donde el Gateway es el servidor y los nodos son considerados los clientes, en la Figura 4 se puede observar como todos los nodos se conectan al Gateway formando una topología tipo estrella.

Figura 4

Topología tipo estrella



Fuente: Elaborada por el autor

2.1.2. Factores Que Influyen En El Diseño De Las WSN

Para el correcto diseño de una red de sensores inalámbrica es necesario tomar en cuenta los diferentes factores físicos y teóricos que pueden presentar un inconveniente al momento del despliegue de la red WSN. A continuación, se muestra cuáles son los factores que influyen en el diseño de una WSN (Ramírez, 2016).

2.1.2.1. Tolerancia A Fallos

La escalabilidad en una red de sensores inalámbricos depende del número de nodos sensores que se podría llegar a desplegar en un área para monitorizar los eventos que se generan. Dependiendo de la aplicación el número de nodos sensores pueden llegar a ser cientos o miles de nodos sensores, en la *Ecuación 1* Se muestra la fórmula utilizada para calcular la densidad o la distancia entre nodos sensores (Ramírez, 2016).

$$\mu(R) = (N\pi R^2)/A$$

Ecuación 1 Formula para el cálculo de distancia entre nodos.

Adaptado de: Ramírez, C. L. (2012). Diseño de una arquitectura para redes de sensores

Donde N es el número de nodos sensores desplegados en la región A, y R el rango de transmisión.

2.1.2.2.Coste De Producción

Depende estrictamente del costo de cada de construcción de cada nodo sensor, ya que una red WSN está compuesta de un número elevado de nodos sensores, por lo que le costo de cada nodo sensor debe ser bajo para justificar el despliegue de una red WSN (Ramírez, 2016)

2.1.2.3.Restricciones Del Hardware

Un nodo sensor este compuesto por los siguientes componentes para su correcto funcionamiento: sensor, microcontrolador, transceptor y administrador de energía, no obstante, dependiendo de la aplicación estos componentes necesitan de componentes adicionales como, por ejemplo, componentes de localización, servomotores, paneles solares donde estos componentes adicionales pueden llegar a saturar la unidad de microcontrolador. Debido a esto se debe diseñar correctamente el sistema embebido teniendo el claro la finalidad de su aplicación (Ramírez, 2016).

2.1.2.4.Consumo De Energía

La vida útil de un nodo sensor esta conjuntamente relacionado con el tiempo de duración de la batería, en casos que no sea recargable. Debido a este factor el diseño de los sistemas embebidos y algoritmos tomando en cuenta la energía (Garbarino, 2017).

2.1.2.5.Costo

Tomando en cuenta que una red de sensores inalámbricos puede estar formada de cientos o miles de sensores, el costo por cada nodo sensor afectara al costo de toda la red.

2.1.2.6.Ambiente

Las redes de sensores inalámbricas básicamente están diseñadas para cubrir espacios geográficamente remotos o espacios con diversos estados climáticos por lo cual es un punto para tomar en el diseño de redes WSN (Garbarino, 2017).

2.1.2.7. Conectividad y Topología

Se necesitan protocolos ad hoc diseñados para soportar cambios de topología debido a la movilidad, disponibilidad de recursos, pérdidas de información, apagones, mal funcionamiento, interferencia, etc. (Garbarino, 2017)

2.1.3. Modelo Referencia OSI para WSN

Tanto la puerta de enlace, nodo sensor y la estación base maneja una pila de protocolos que combina de manera óptima el uso de energía y el encaminamiento de los datos, logrando la integración de los datos a los protocolos de red, comunicación entre la red WSN de manera eficiente a través del medio inalámbrico (Ramírez, 2016).

Tabla 1

Funcionalidades del protocolo de comunicación

| Capas | Descripción |
|-------------------------|--|
| Capas superiores | Aplicaciones residentes en la red, procesamiento, agregación, procesamiento de consultas externas, y base de datos externa. |
| Capa 4 | Transporte , incluyendo diseminación y acumulación de datos, cache y almacenamiento. |
| Capa 3 | Red , administración dinámica de la topología y proporciona el encaminamiento de datos. |
| Capa 2 | Enlace , administración del canal compartido, competencia por el canal, y acceso al medio, sincronización y localización. |
| Capa 1 | Física , canal de comunicación, modulación, transmisión y procesamiento de señales. |

Fuente: adaptado de Garbarino, J. (2017). Protocolos para redes inalámbricas de sensores

Tabla 2

Funcionalidades del protocolo de gestión

| Planos | Descripción |
|---------------------------------------|---|
| Plano de gestión de tareas | Los planos de gestión de tareas controlan la energía, el movimiento y la distribución de tareas entre los nodos sensores. Estos planos ayudan a los nodos sensores a coordinar las tareas de monitorización y, sobre todo, el bajo consumo de energía. |
| Plano de gestión de movimiento | El plano de gestión de movilidad detecta y registra el movimiento de los nodos sensores de manera que la ruta de regreso hacia el administrador siempre se conserve, y los nodos sensores puedan mantener el rastro de quienes son sus nodos sensores vecinos. |
| Plano de gestión de energía | El plano de gestión de energía administra cómo los nodos sensores usan la energía. |

Fuente: adaptado de Garbarino, J. (2017). Protocolos para redes inalámbricas de sensores

2.2. Estándar De Comunicación

2.2.1. Estándar IEEE 802.11

El estándar IEEE 802.11 es un conjunto de estándares que especifica el modelo de comunicación para las redes inalámbricas de área local (WLAN), lanzada en 1997 con velocidades de 1 o 2 Mbps utilizando técnica de modulación salto de frecuencia o espectro de dispersión de secuencia directa, utilizando la banda ISM de 2.4 GHz. Con el paso de los años esta tecnología ha evolucionado y se han lanzado varias versiones que mejoran las características (Cabrera, 2020).

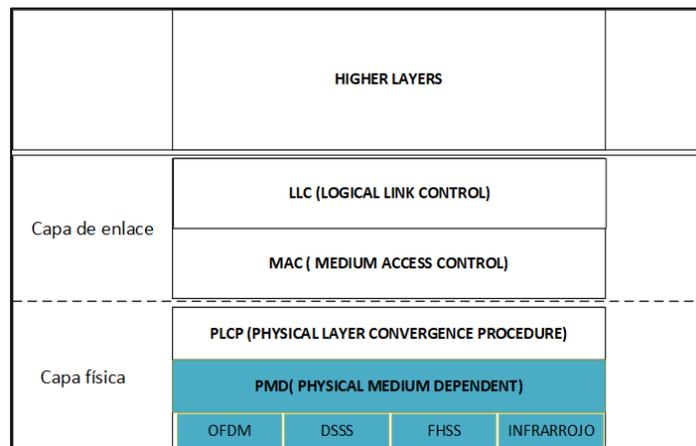
El estándar IEEE 802.11 trabaja a nivel de capa física y capa enlace de datos del modelo de referencia OSI.

2.2.2. Pila De Protocolos

La arquitectura lógica del estándar IEEE 802.11 está definida por el nivel MAC PHY del modelo de referencia OSI. En la Figura 6 se especifica las diferentes capas que interviene y opera el estándar IEEE 802.11 en relación del modelo de referencia OSI.

Figura 5

Estándar IEEE 802.11 y la pila de protocolos del modelo de referencia OSI



Fuente: adaptado de Fiallos, S. F. (2017). Estudio y análisis para la actualización de red wlan

La capa de enlace se divide en dos subcapas:

La subcapa **LLC (Logical Link Control)**, que ofrece un servicio de transporte único para todas las tecnologías, gestionando los enlaces lógicos de capa de enlace y proporciona una interfaz común para el nivel de red, esta subcapa es común a todos los estándares IEEE 802.11 (Cabrera, 2020).

La subcapa **MAC (Media Access Control)**, donde se especifica el protocolo de acceso al medio, especificando cada una de las funcionalidades o procedimientos que ocurren en las redes inalámbricas al momento de la transmisión entre el receptor y emisor,

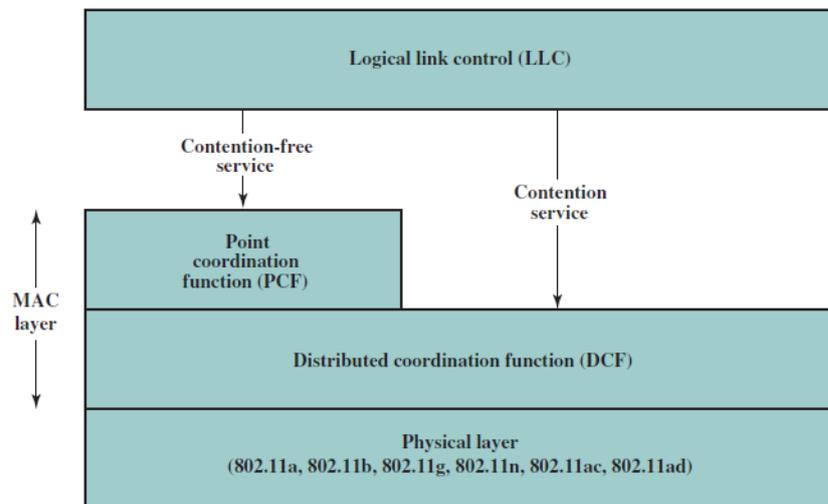
también la posibilidad de realizar fragmentación de las tramas y los mecanismos de encriptación para dar confidencialidad a los datos transmitidos (Cabrera, 2020).

2.2.3. Control De Acceso Al Medio (MAC) IEEE 802.11g

La subcapa de control de acceso al medio MAC proporciona un mecanismo de control de acceso distribuido con un control centralizado opcional. En la Figura 6 se muestra la arquitectura donde la subcapa inferior de la capa MAC es la coordinación distribuida función (DCF). DCF utiliza un algoritmo de contención para brindar acceso a todo el tráfico. El tráfico asíncrono ordinario utiliza directamente DCF (Stallings, Data , 2014).

Figura 6

Arquitectura 802.11 MAC



Fuente: adaptado de Stallings(2014) Data and Computer Communications

Los métodos de acceso al medio permiten evitar las colisiones de los paquetes evitando la transmisión simultánea de las estaciones que forman parte de la misma red inalámbrica, por lo que es necesario la regulación del acceso al canal, debido a que todas las estaciones utilizan el mismo medio físico para el intercambio de información (Stallings, Data , 2014).

El nivel MAC se encarga de la entrega de las unidades informativas de nivel superior (LLC) entre estaciones origen y destino, garantizando el éxito de la transferencia y ocupándose de la gestión de la denominada lógica de coordinación para acceso al medio. (Cabrera, 2020)

Distributed coordination function (DCF) o en español función de coordinación distribuida: usado en redes Ad-hoc y en redes de infraestructura, al no existir un control centralizado, las estaciones y puntos de acceso son iguales, por lo que puede existir colisiones o varias estaciones transmitiendo al mismo tiempo, para evitarlo se utiliza un protocolo de acceso al medio CSMA/CA (Carrier Sense Multiple Access/Colision Avoidance) (Cabrera, 2020).

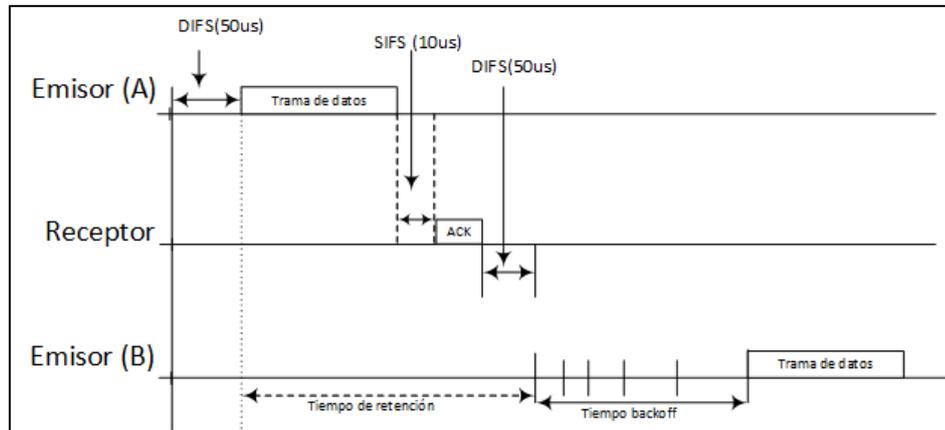
2.2.3.1. Protocolo CSMA/CA

El protocolo de acceso múltiple por detección de portadora y prevención de colisiones pertenece a la clase de protocolos de acceso (CSMA), que realizan un escaneo (sensing) del canal antes de realizar una transmisión, cuando la trama esta lista la estación transmisora comprueba si el canal esta inactivo u ocupado, solo si el canal esta libre la estación podrá transmitir, logrando así reducir las colisiones (Gutiérrez, 2020).

En la Figura 7 se muestra el mecanismo de operación de emisor y transmisor para evitar colisiones, cuando la A desea transmitir espera un tiempo DIFS y envía la trama de datos, el receptor espera un tiempo SIFS y envía una acuse de recibo ACK, en este caso el emisor B sabe que A ya acabo de transmitir pero no envía de inmediato la trama, espera un tiempo DIFS y un valor de backoff time que permite reducir colisiones, si en ese tiempo de backoff el canal esta libre transmite la trama, caso contrario se recalcula el tiempo de backoff (Gutiérrez, 2020).

Figura 7

Emisión de una trama con CSMA/CA



Fuente: adaptado de ANSI/IEEE Std 802.11, Wireless LAN Medium Access Control (MAC) and physical Layer (PHY) Specifications

Los intervalos interframe del nivel MAC 802.11g

Las tramas del nivel MAC están espaciadas en el tiempo a intervalos llamados IFS (InterFrame Space).

ASlotTime: El intervalo de tiempo es de 20 μ s, excepto cuando el BSS consta solo de ERP, en este caso se puede usar un intervalo de tiempo opcional de 9 μ s. si las estaciones son capaces de admitir esta opción. El intervalo de tiempo opcional de 9 μ s, no se utilizará si la red tiene una o más estaciones que no consten asociados con ERP. (ANSI/IEEE sta 802.11, 2003)

SIFS (Short IFS): es el más breve intervalo de tiempo definido. Viene utilizado para transmitir sobre el medio, con ERP utilizará un SIFS de 10 μ s. (Marchán, 2018).

DIFS (DCF IFS): viene utilizado por las estaciones que operan en modalidad DCF para transmitir tramas de datos (MPDU) o tramas de gestión (MMPDU, MAC Management Protocol Data Unit) (Marchán, 2018). Se determina utilizando la Ecuación

2 donde el resultando es dado por una unidad temporal definida a su vez por el tipo de nivel físico utilizado aSlotTime (ANSI/IEEE sta 802.11, 2003).

$$DIFS = aSIFSTime + 2 \times aSlotTime$$

$$DIFS = 10 \mu s + 2 \times 20 \mu s = 50 \mu s$$

***Ecuación 2** Valor DIF en los intervalos interframe*

Fuente: adaptado de ANSI/IEEE Std 802.11, Wireless LAN Medium Access Control (MAC) and physical Layer (PHY) Specifications

Random backoff time: este valor se da cuando una estación a esperado un tiempo DIFS y el canal aún está ocupado, la estación generara un random backoff para un tiempo de aplazamiento adicional antes de transmitir, este tiempo de backoff viene dado por la Ecuación 3.

$$\text{Backoff Time} = \text{Random}() \times aSlotTime$$

***Ecuación 3** Valor Random backoff time en los intervalos interframe*

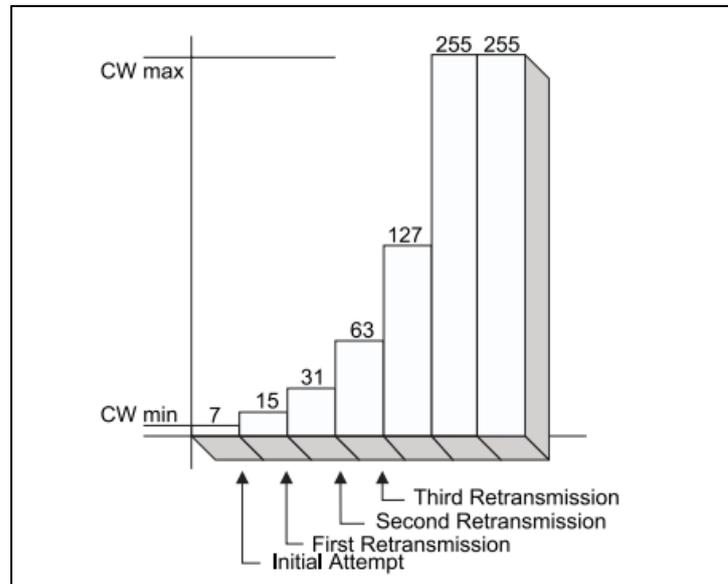
Fuente: adaptado de ANSI/IEEE Std 802.11, Wireless LAN Medium Access Control (MAC) and physical Layer (PHY) Specifications

Donde:

Random() = valor entero pseudoaleatorio extraído de una distribución uniforme sobre el intervalo [0,CW], donde CW es un número entero dentro del rango de valores de las características PHY aCWmin y aCWmax, $aCWmin \leq CW \leq aCWmax$. (aCWmin=15; aCWMax=1023). En la Figura 8 se muestra el crecimiento exponencial de las ventanas de contención CW.

Figura 8

Ejemplo del incremento exponencial de CW



Fuente: adaptado de ANSI/IEEE Std 802.11, Wireless LAN Medium Access Control (MAC) and physical Layer (PHY) Specifications

Tramas RTS (Request to send) Y CTS (clear to send)

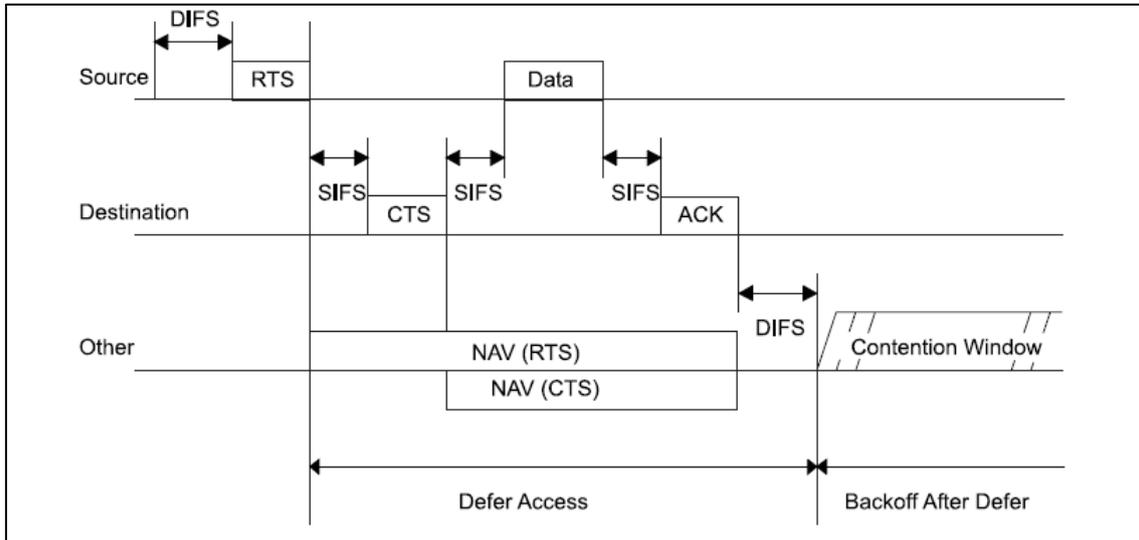
El protocolo IEEE 802.11 plantea el uso de una breve trama de control petición de envío (RTS, Request To Send) y una breve trama de vía libre para el envío (CTS, Clear To Send) para reservar el acceso al canal.

El mecanismo prevé que la estación, antes de transmitir la trama de datos, envíe al receptor la trama de control RTS, especificando en el campo *Duration* el tiempo total necesario para la transmisión de la trama de datos y del relativo frame ACK. El destinatario responde después de un SIFS con otra trama de control CTS, en el cual copia el campo de duración. Tal información consiente a todos los nodos que ven el intercambio de los dos paquetes de actualizar el NAV. Los nodos que ven solo el CTS (eventuales nodos escondidos), actualizan el NAV en un momento sucesivo respecto a aquellos que

ven el RTS (Marchán, 2018). En la Figura 9 se muestra las modalidades de acceso al canal según las tramas CTS, RTS y los intervalos de tiempo interframe.

Figura 9

Modo de procedimiento de las tramas RTS y CTS



Fuente: adaptado de ANSI/IEEE Std 802.11, Wireless LAN Medium Access Control (MAC) and physical Layer (PHY) Specifications

2.2.4. Estructura De Trama 802.11g

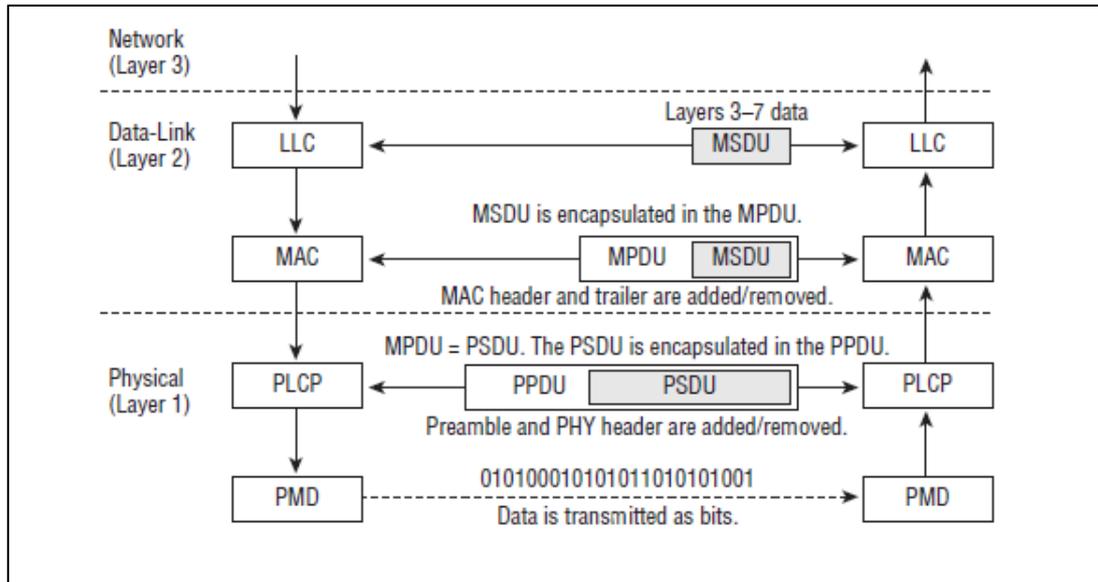
MSDU (MAC service data unit) Cuando la capa de red (capa 3) envía datos a la capa de enlace de datos, esos datos se transfieren a la LLC y se conocen como Unidad de datos de servicio MAC (MSDU), la La longitud de la MSDU debe ser menor o igual a 2304 octetos.

MPDU (MAC protocol data unit) Cuando la subcapa LLC envía la MSDU a la subcapa MAC, la subcapa MAC le agrega una cabecera de 30 bytes y un FCS de 4 bytes. Y la encapsulada en una unidad de datos de protocolo MAC (MPDU), la longitud de un MPDU de 3895 octetos.

PPDU (PLCP protocol data unit) cuando la subcapa PLCP recibe la MPDU le agrega un preámbulo y un encabezado PHY (40 bits) y crea la Unidad de datos de protocolo PLCP (PPDU).

Figura 10

Encapsulación de unidad de datos según la capa



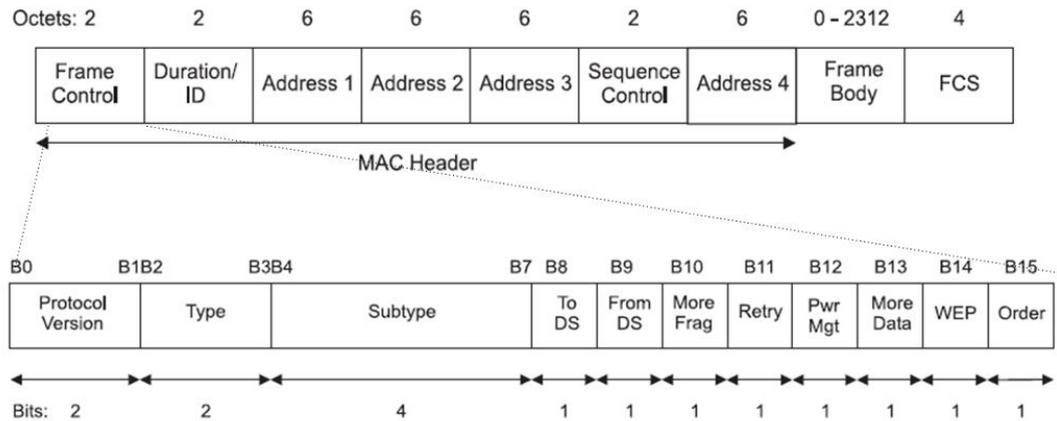
Fuente: adaptado de ANSI/IEEE Std 802.11, Wireless LAN Medium Access Control (MAC) and physical Layer (PHY) Specifications

2.2.4.1.MPDU Frame

En la Figura 11 se muestra el formato de la trama IEEE 802.11, también conocido como protocolo MAC unidad de datos (MPDU). Este formato general se utiliza para todos los marcos de datos y control, pero no todos los campos se utilizan en todos los contextos (Stallings, 2014). Los campos son los siguientes:

Figura 11

Formato de la trama de datos de 802.11



Fuente: adaptado de Tanenbaum, A. S., & Wetherall, D. J. (2017). Redes de computadoras.

Frame control: es el campo inicial que está constituido por 11 subcampos con un tamaño de 2 bits, contiene la siguiente información:

Protocol Version: es un subcampo de dos bits que permite la coexistencia de varias versiones del protocolo.

Type: es un subcampo de dos bits que indica si se trata de una trama de datos, control o gestión.

Subtype: es un subcampo de cuatro bits que indica si es una trama de CTS O RTS. En la Tabla 3. Se muestra el tipo de trama dependiendo la combinación de tipo y subtipo.

Tabla 3

Tipos de tramas de acuerdo con la combinación del campo tipo y subtipo

| Tipo | Subtipo | Trama |
|--------------------------|---------|----------------------------|
| Tramas de gestión | | |
| 00 | 0000 | Petición de asociación |
| 00 | 0001 | Respuesta de asociación |
| 00 | 0010 | Petición de re-asociación |
| 00 | 0011 | Respuesta de re-asociación |
| 00 | 0100 | Petición probe |
| 00 | 0101 | Respuesta probe |

| | | |
|--------------------------|------|------------------------|
| 00 | 1000 | Beacon |
| 00 | 1001 | ATIM |
| 00 | 1010 | Des-Asociación |
| 00 | 1011 | Autenticación |
| 00 | 1100 | Des- Autenticación |
| Tramas de datos | | |
| 10 | 0000 | Datos |
| 10 | 0001 | Datos + CF-ACK |
| 10 | 0010 | Datos + CF-Poll |
| 10 | 0011 | Datos + CF-ACK+CF-Poll |
| 10 | 0100 | Null |
| 10 | 0101 | CF-ACK |
| 10 | 0110 | CF-Poll |
| 10 | 0111 | Datos + CF-ACK+CF-Poll |
| Tramas de control | | |
| 01 | 1010 | Power Save (PS) |
| 01 | 1011 | RTS |
| 01 | 1100 | CTS |
| 01 | 1101 | ACK |
| 01 | 1110 | CF-End |
| 01 | 1111 | CF-end + CF-ACK |

Fuente: adaptado de ANSI/IEEE Std 802.11, Wireless LAN Medium Access Control (MAC) and physical Layer (PHY) Specifications

To DS: es un subcampo de un solo bit que indica si la trama va dirigida al punto de acceso, que coordina las comunicaciones en el sistema inalámbrico centralizados.

From DS: un subcampo que indica si la trama proviene del punto de acceso.

More fragm: un subcampo de un solo bit que, si se establece en 1, indica que siguen más fragmentos.

Retry: un subcampo de un bit que indica que la trama es un reintento.

Pwr Mgt: es un subcampo de un solo bit que indica si el emisor está adoptando el modo de ahorro de energía.

More Data: un subcampo de un bit que advierte que el emisor tiene más datos para enviar.

WEP: un subcampo de un bit que indica si la trama esta encriptada con WEP(Wireless Equivalent Privacy).

Order: es el último subcampo de un bit que indica si las tramas con este campo en 1, se van a procesar la trama en orden.

Duration/ID: es un campo de dos bytes que indica cuanto tiempo va a estar ocupando el canal la trama.

Address 1, 2, 3 y 4: son cuatro campos de direcciones de seis bytes que contienen una dirección en el formato de la trama MAC y se utilizan para indicar el Basic Service Set Identifier (BSSID), el Destination Address (DA), el Source Address (SA), el Receiver Address (RA) y el Transmitter Address (TA) (Fiallos, 2017).

En la Tabla 4 se puede observar cómo es la distribución de los campos To DS=0 y From DS=0 para cada address.

Tabla 4

Interpretaciones de los campos Address 1, 2, 3 y 4 en función de To DS y From DS.

| To DS | From DS | Address 1 | Address 2 | Address 3 | Address 4 |
|-------|---------|-----------|-----------|-----------|-----------|
| 0 | 0 | DA | SA | BSSID | N/A |
| 0 | 1 | DA | BSSID | SA | N/A |
| 1 | 0 | BSSID | SA | DA | N/A |
| 1 | 1 | RA | TA | DA | SA |

Fuente: adaptado de ANSI/IEEE Std 802.11, Wireless LAN Medium Access Control (MAC) and physical Layer (PHY) Specifications

Sequence Control: Es un campo de 2 bytes que almacena los números de la trama. Detecta fragmentos duplicados y determina el orden de fragmento para capas superiores. Entre los 16 bits, los primeros 4 bits proporcionan identificación al fragmento y los 12 bits resto contienen el número de secuencia que aumenta con cada transmisión (Fiallos, 2017).

Frame Body: este es un campo de tamaño variable que transporta la carga útil desde las capas superiores con un tamaño máximo del campo de datos de 2312 bytes (Fiallos, 2017).

FCS: este es un campo de 32 bits que contiene el código CRC (código de redundancia cíclica) a 32 bit.

2.2.5. Capa Física 802.11g

La capa física se divide en dos subcapas:

La subcapa superior, **PLCP (Physical Layer Convergence Procedure)**, se encarga de adaptar las diversas especificaciones de la subcapa PMD a la subcapa MAC (Cabrera, 2020).

IEEE 802.11g utiliza DSSS, OFDM o ambos para proporcionar transmisión de velocidades de datos de hasta 54 Mbps en la banda ISM de 2,4 GHz. IEEE 802.11g combina las características de IEEE 802.11, IEEE 802.11a e IEEE 802.11b. Como resultado, combina la aplicación de ambos DSSS y OFDM obtenido cuatro capas físicas diferentes donde remitente y el receptor pueden elegir y usar una de las cuatro capas durante un intercambio de fotogramas en la Figura 12 se muestra los parámetros de las diferentes capas físicas de 802.11g (Khanduri, Rattan, & Uniyal, 2013).

Figura 12

Capas físicas de IEEE 802.11g

| Physical layer | Data rates (Mbps) | PLCP preamble +header delay | | PLCP preamble + header length | |
|------------------------|----------------------------|-----------------------------|--------|-------------------------------|----------|
| | | Short | Long | Short | Long |
| ERP- DSSS (Essential) | 1,2, 5.5,11 | 96µs | 192 µs | 120 bits | 192 bits |
| ERP-OFDM (Essential) | 6, 9,12, 18,24, 36, 48, 54 | 20µs | | 40 bits | |
| ERP- PBCC (Arbitrary) | 1,2,5.5, 11, 22, 33 | 96µs | 192 µs | 120 bits | 192 bits |
| DSSS- OFDM (Arbitrary) | 6, 9,12, 18,24, 36, 48, 54 | 96µs | 192 µs | 120 bits | 192 bits |

Fuente: adaptado de Khanduri, R., Rattan, S. S., & Uniyal, A. (2013). Understanding the Features of IEEE 802.11g in High Data Rate Wireless LAN

ERP-DSSS/CCK: Las especificaciones 802.11b mejoraron los datos velocidad y compatibilidad mantenida con el 802.11 original estándar DSSS. Utiliza un esquema de codificación más eficiente conocido como codificación de código complementario (CCK) para proporcionar una tasa de datos de 11 Mbps (Khanduri, Rattan, & Uniyal, 2013).

ERP-OFDM: IEEE 802.11g definió esta capa física. IEEE 802.11a utiliza esquemas de modulación multiportadora que es la multiplexación por división de frecuencia ortogonal (OFDM) por utilizar el espectro de radio de 2,4 GHz para proporcionar velocidades de datos de 6 Mbps a 54 Mbps (Khanduri, Rattan, & Uniyal, 2013).

ERP-DSSS/PBCC: esta capa física ofrece los mismos datos velocidades como la capa física DSSS/CCK. DSSS/PBCC fue introducido por IEEE 802.11b. Se utiliza para proporcionar mayor rendimiento en el rango a las tasas de 5.5 y 11 Mbps por utilizando

tecnología DSSS con código de convolución binaria de paquetes (PBCC) esquema (Khanduri, Rattan, & Uniyal, 2013).

DSSS-OFDM: una combinación híbrida de DSSS y OFDM es utilizado por esta nueva capa física. la transmisión del encabezado físico del paquete lo realiza DSSS, mientras que el OFDM realiza la transmisión de la carga útil del paquete. Este uso de la capa física para cubrir aspectos de interoperabilidad (Khanduri, Rattan, & Uniyal, 2013).

Throughput 802.11g

En la Tabla 13 se muestra el rendimiento dependiendo de la distancia y de mecanismos para reducción de colisiones.

Figura 13

Throughput 802.11g

| Distancia (metros) | 802.11g (Mbps) | 802.11g Mixed Enviroment With CTS-toSelf (Mb/s) | 802.11g Mixed Environment with RTS/CTS (Mb/s) |
|---------------------------|-----------------------|--|--|
| 3 | 24.7 | 14.7 | 11.8 |
| 15 | 24.7 | 14.7 | 11.8 |
| 30 | 19.8 | 12.7 | 10.6 |
| 45 | 12.4 | 9.1 | 8 |
| 60 | 4.9 | 4.2 | 4.1 |
| 76 | 1.6 | 1.6 | 1.6 |
| 91 | 0.9 | 0.9 | 0.9 |

Fuente: adaptado de Khanduri, R., Rattan, S. S., & Uniyal, A. (2013). Understanding the Features of IEEE 802.11g in High Data Rate Wireless LAN

El rendimiento no es el mismo que la tasa de datos para sistema de red, el rendimiento de la estación más rápida se reduce por la velocidad de la estación más lenta debido a un mayor tiempo tomado por una estación para transmitir una trama de datos a una tasa de bits menor, como un resultado el tiempo de canal utilizado para la transmisión de las estaciones serán disminuciones. En un entorno WLAN uniforme, el IEEE 802.11g

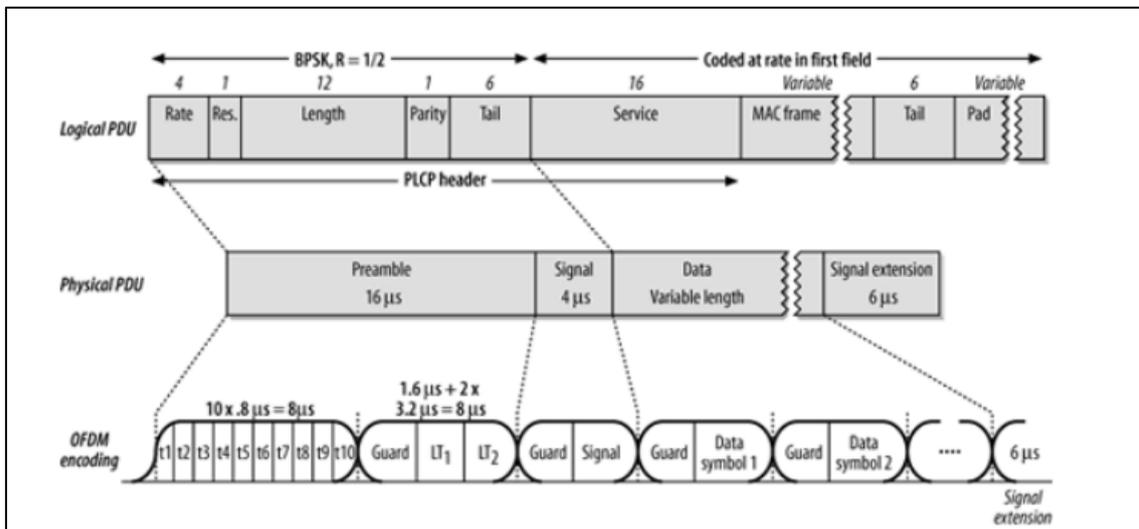
proporciona el mejor rendimiento cuando el punto de acceso (AP) 802.11g se comunica solo con clientes 802.11g,

2.2.5.1. Estructura de la trama PLCP 802.11g

ERP-OFDM es, con diferencia, la PPDU implementada con mayor frecuencia en el estándar 802.11g, y admite velocidades de datos de 6, 9, 12, 18, 24, 36, 48 y 54 Mbps. El ERP-OFDM PPDU tiene tres partes: preámbulo, encabezado y campo de datos. En la Figura 14 se observa la estructura del PPDU ERP-OFDM (CWAP, 2018)

Figura 14

ERP-OFDM PPDU (802.11g)



Fuente: adaptado de CWAP (2018) PHYSICAL LAYER 802.11

PLCP Preamble: permite al receptor adquirir una señal OFDM entrante (detección de señal) y sincronizar su demodulador.

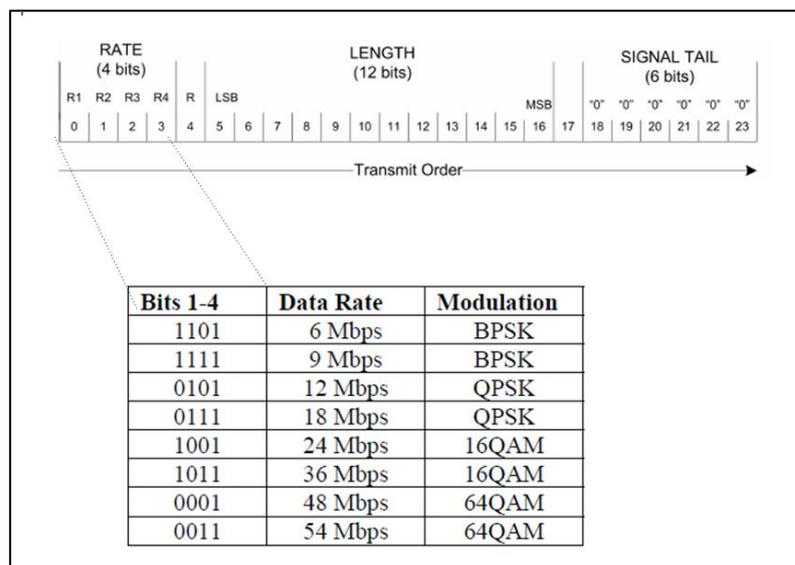
PLCP Header: El subcampo rate consta de 4 bits, indica la tasa de modulación y codificación del resto de la PPDU, el subcampo Reservado (1 bit) se establece en 0 ya que actualmente no se utiliza. El subcampo de longitud (12 bits) indica el número de octetos en la PSDU que el nivel MAC está solicitando actualmente al nivel PHY para

transmitir. El subcampo de paridad es un bit de paridad positivo (par) de un bit, basado en los primeros 17 bits (0-16).

El subcampo Rate consta de 4 bits, como se describe en la Figura 15, e indica la tasa de modulación y codificación del resto de la PDU, comenzando inmediatamente después del campo Señal.

Figura 15

Subcampo Rate de PLCP Header



Fuente: adaptado de ANSI/IEEE Std 802.11, Wireless LAN Medium Access Control (MAC) and physical Layer (PHY) Specifications

Data: El subcampo Servicio consta de 16 bits, con los primeros 7 bits como ceros para sincronizar el desaleatorizador en el receptor. La PSDU es la unidad de datos que se envía desde la capa MAC para transmisión en el medio inalámbrico y tiene un máximo longitud de 4095 octetos (CWAP, 2018).

2.2.5.2.Subcapa ERP-OFDM PMD

La subcapa inferior, **PMD (Physical Media Dependent)**, que corresponde al conjunto de especificaciones y características de cada uno de los sistemas de transmisión

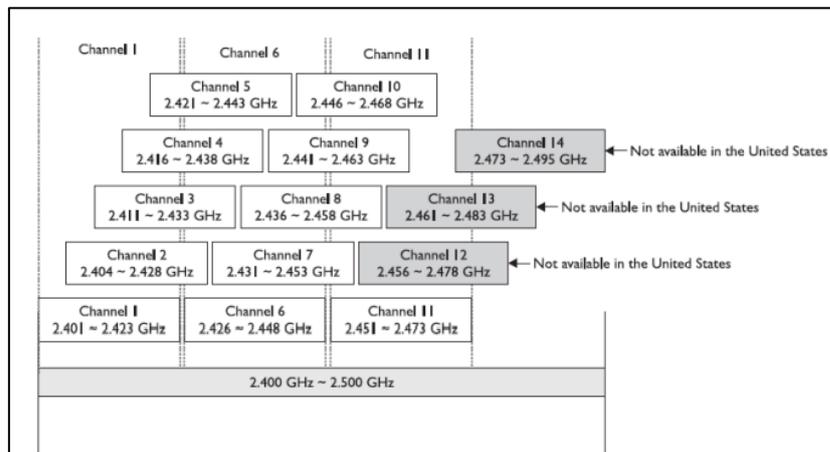
a nivel físico. El estándar define cuatro: Infrarrojos, FHSS, DSSS o OFDM (Cabrera, 2020).

El ERP-OFDM PMD realiza la transmisión y recepción real de PPDU bajo la dirección del PLCP. Para brindar este servicio, el PMD interactúa directamente con el medio inalámbrico y proporciona la modulación OFDM y demodulación de las transmisiones (CWAP, 2018).

Frecuencias y Asignación Canales

Los dispositivos 802.11 operan en el espectro de 2,4 GHz y dentro de las bandas Industrial, Científica y Médica (ISM). El espectro de 2,4 GHz se dividió en catorce canales de 22 MHz de ancho, con frecuencia por canal de 5 MHz (Fiallos, 2017). En la Figura 16. Se muestra el mapa de frecuencias asignadas para IEEE 802.11 y una distribución de canales.

Figura 16 Mapa de frecuencia de canal a 2,4 GHz.



Fuente: adaptado de Fiallos, S. F. (2017). Estudio y análisis para la actualización de red

wlan

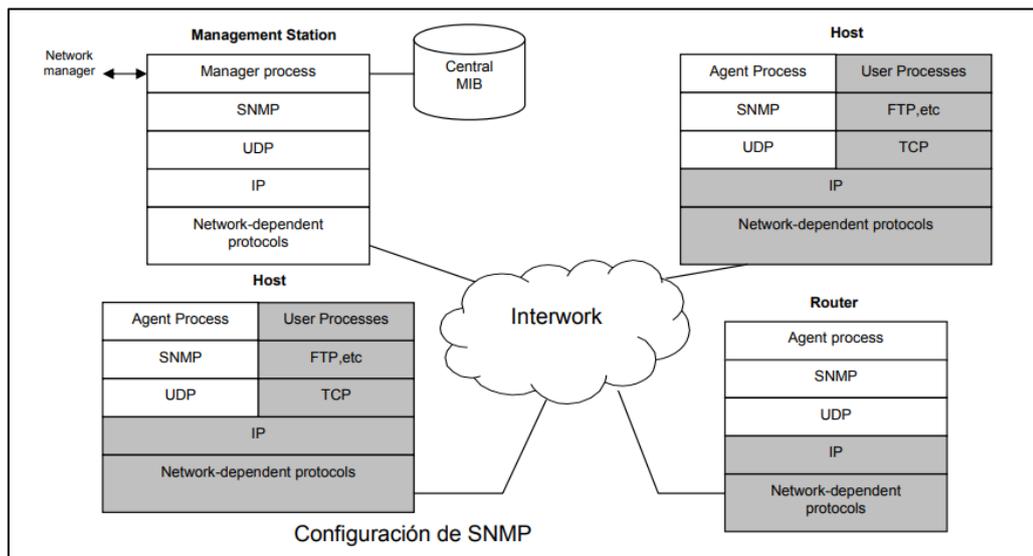
2.3. Gestión De Red

2.3.1. Definición De Gestión De Red

La gestión de red se enfoca en diversos aspectos como la planificación, organización, monitoreo, supervisión y control de los equipos y dispositivos que conforman la red, con el objetivo de garantizar el nivel de servicio, utilizando diferentes recursos de forma óptima y eficaz que juntamente con el administrador de red logran tener la red totalmente funcional (Perez, 2015) (Tituaña, 2016). En Figura 17 se muestra la arquitectura que se establece para modelos de gestión de redes.

Figura 17

Modelo de gestión de red SNMP



Fuente: adaptado de Pérez, S. A. (2015). *Estudio comparativo de los sistemas de gestión y monitoreo basados en los requerimientos generales de la red de un campus universitario.*

2.3.2. Objetivo De La Gestión De Redes

El objetivo principal de la gestión de redes es proporcionar una solución que logre garantizar el correcto funcionamiento de la red, logrando prevenir o lograr dar una breve solución a los inconvenientes que pueden causar fallos en los equipos de red tanto a nivel de software como hardware (Perez, 2015).

Existen varios objetivos de gestión red que se establecen como requerimientos para garantizar la calidad de servicio en la red gestionada, a continuación, se menciona cada una de ellas:

Alta disponibilidad de red: al hablar de disponibilidad, se habla del tiempo que se mantiene operativa, el tiempo de respuesta al solicitar una petición a los servicios de la red y el porcentaje de tolerancia a la inoperatividad (Perez, 2015).

Reducción de costos de operación de la red: la administración heterogénea y multiprotocolo es fundamental para la operatividad de la red (Perez, 2015).

Reducción de cuellos de botella en la red: la monitorización de los equipos de CORE de la red que manejan grandes cantidades de información, logrando así el balanceo de carga, mejorando la disponibilidad de la red (Perez, 2015).

Incrementar la flexibilidad de operación e integración: con la constante actualización de hardware y software, los sistemas de gestión de red deben poseer la capacidad de ser adaptables, logrando migrar a las redes de nueva generación sin causar un costo elevado de implementación y lograr la coexistencia entre diferentes tecnologías (Perez, 2015).

2.3.3. Funciones De Gestión De Red

Las funciones principales de la gestión de red son: controlar y monitorizar a los elementos de red.

Monitorización: dentro del monitoreo se tiene dos enfoques fundamentales en la monitorización de elementos de red:

Monitoreo activo: El gestor constantemente requiere información de estado del elemento monitorizado, con la finalidad de verificar el correcto funcionamiento y

tiempo de respuesta del elemento de red gestionados, en este caso los elementos de red gestionados notifican solo cuando el gestor lo solicita (Bayas, 2015).

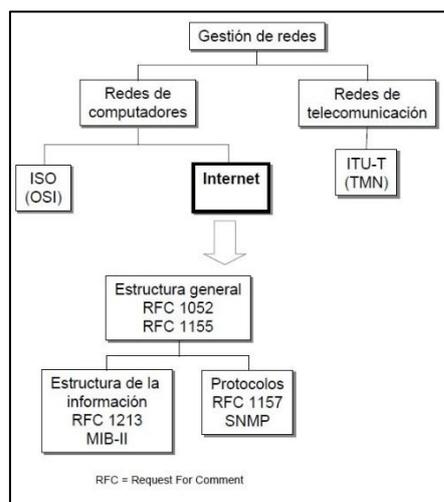
Monitoreo pasivo: el agente notifica espontáneamente al gestor cuando ocurre un evento inusual en algún componente en el elemento de red. Esto lo realiza sin necesidad que exista una petición del gestor (Bayas, 2015).

2.3.4. Modelos De Gestión De Redes

El crecimiento constante de las redes de datos y el requerimiento de disponibilidad de la información ha creado la necesidad de la implementación de sistemas de gestión de redes con la capacidad de testear, configurar, controlar monitorizar los elementos de red, implementando tareas de diseño, coordinación hardware, software y los profesionales de la administración de red para lograr proporcionar la calidad de servicio, también debe prever posibles puntos de falla y evitarlos. En este sentido, varios organismos de estandarización como ISO, ITU-T e IETF han trabajado en normas de administración de redes (Bayas, 2015). En la Figura 18 se muestra el diagrama de bloques muestra la distribución de modelos de gestión de red dependiendo de las áreas de aplicación.

Figura 18

Modelos de gestión de redes



Fuente: adaptado de Romero, M. (2017) Gestión de Redes. Sistemas Avanzados de Comunicaciones.

2.3.4.1. Modelo De Gestión internet

El constante crecimiento de las redes de computadoras era imposible realizar la gestión de la red de manera autónoma por la cual se empieza a utilizar protocolos que ayuden a verificar que existe camino de retorno para los datos haciendo pruebas de conectividad, el protocolo usado fue ICMP (Internet Control Message Protocol, así como el PING(Packet internet Grouper), a fines de los 80 era totalmente ineficiente, por lo que se empieza a buscar nuevos modelos para la gestión de las redes (Tituaña, 2016).

Debido a esta problemática se dio el surgimiento de nuevos grupos de trabajo que propusieron nuevos protocolos de gestión: SGMP (protocolo simple de monitorización de pasarelas), HEMS (sistema de gestión de entidades de alto nivel) y CMOT(CMIP sobre TCP), obteniendo mejores resultados con el protocolo SGMP que en el futuro evolucionaría con el nombre de SNMP(protocolo simple de gestión de red) después de varias modificaciones y mejoras sería estandarizado en los 90, SNMP se ha considerado como uno de los protocolos de gestión de redes con mayor relevancia e importancia de la actualidad, debido a la proliferación de redes basadas en la arquitectura TCP/IP (protocolo de control de transmisión/protocolo de internet) y está basado en el estándar de gestión de redes IP especificado en el [RFC 1212] (Tituaña, 2016).

2.3.4.2. Arquitectura De Administración SNMPv2

Para existir la coexistencia entre el protocolo SNMPV1 y SNMPV2, por lo que en el [RFC 1908] se especifican los esquemas de aplicación para la coexistencia, esta versión ya permite el soporte de comunicación gestor-gestor (PDU Protocol Data Unit informrequest). La lectura de tablas completas en una sola operación (PDU getbulkrequest) (Sosa, 2016) . En la Figura 19 se muestra la arquitectura de administración usado por SNMPv2 para la gestión de redes TCP/IP está formada por los elementos.

notificar a las peticiones (acciones, información) impuestas por la estación de gestión(gestor) (Tituaña, 2016).

Base de información de administración (MIB Management Information Base): se puede definir a la MIB como la colección de objetos que pueden proporcionar los recursos de cada elemento gestionado, los recursos puede ser representados como objetos, cada objeto es un dato variable que representa un aspecto del agente administrador (Perez, 2015).

Según (Tituaña, 2016), hay objetos genéricos que son definidos por el IETF y pueden ser administrados por cualquier sistema de administración de red compatible con SNMP, la norma MIB-II especificada en el RFC 1213, define las variables para algunas características de los dispositivos por ejemplo los estadísticos funcionales de un interfaz de un host, bytes enviados o recibidos, etc.

Protocolo de gestión de redes: para que la estación de gestión y el agente administrados puedan intercambiar información, deben ligarse a un protocolo que pueda especificar el comportamiento de ambas partes, el protocolo usado se denomina SNMP (protocolo simple de administración de red) el cual es usado para la administración de redes basadas en la arquitectura TCP/IP (Perez, 2015).

Estructura o Sintaxis De Gestión De Información (SMI): Especificado por el [RFC 1155] donde se establece la manera de identificar los objetos gestionados y su comportamiento, el SMI especifica la sintaxis de identificación que tendrá cada uno de los objetos gestionados de la red. Por ejemplo, la interfaz FastEthernet de un router.

Según (Perez, 2015) La SMI (Structure of Management Information), define la estructura bajo la cual podrá ser definida y construida una MIB. Señala aspectos tales como el tipo de datos que podrá albergar la MIB, cómo pueden representarse y

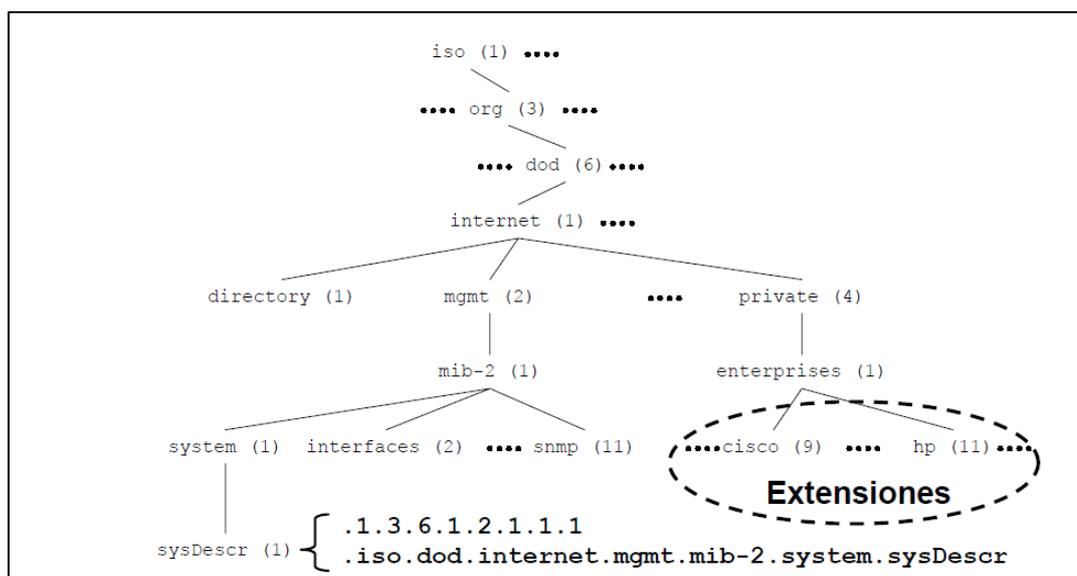
denominarse los recursos en la MIB, donde la sintaxis de un tipo de objeto se define utilizando la notación ASN.1 que se enfoca específicamente en el tipo de objeto y no con la instancia del objeto.

Identificador del objeto (OID): es el nombre único que posee un objeto gestionado, según (ISO/IEC-8824-1, 2015) los objetos gestionados se organizan en una jerarquía tipo árbol, y se representa por una cadena de caracteres que se componen de una serie de números enteros o nombres separados por puntos, estos caracteres ordenados jerárquicamente representan un objeto gestionado dentro del agente (Perez, 2015).

En la Figura 20 se muestra el árbol de jerarquía de objetos el cual cuenta con los niveles de nodos y sub-nodos, donde el nodo raíz principal se derivan más sub-nodos dependiendo del objeto a identificar este tomará un valor único, por lo cual el OID que comprenderá cualquier objeto de internet tendrá como prefijo inicial *1.3.6.1* o también se puede escribir como *iso.org.dod.internet*, este valor comprende el nombre de identificación asignado jerárquicamente para los objetos de internet (Perez, 2015).

Figura 20

Estructura de identificadores de objetos



Fuente: adaptado de Romero, M. (2017) Gestión de Redes. Sistemas Avanzados de Comunicaciones.

Formato común para PDU SNMPv2

En la Figura 21 se muestra la estructura común de las unidades de datos del protocolo SNMPv2.

Figura 21

PDU para SNMPv2

| | | | | | | | | |
|----------|-----------|--------------|-------------|----------------|-----------------|-----|----------------|-----------------|
| PDU Type | RequestID | Error Status | Error Index | VarBind 1 name | VarBind 1 value | ... | VarBind n name | VarBind n value |
|----------|-----------|--------------|-------------|----------------|-----------------|-----|----------------|-----------------|

Fuente: adaptado de Subramanian, Gonsalves, & Rani (2015) Network management: principles and practice.

Type PDU: es un valor tipo integer que indica el tipo de PDU, en la Tabla 5 se muestra los tipos de PDU para SNMPv2.

Tabla 5

Tipo y descripción de cada PDU snmpv2

| FIELD | TYPE | VALUE | Description |
|-------|------|--------------------|--|
| PDU | 0 | GetRequest-PDU | Se genera y transmite una GetRequest-PDU a petición de valores específicos de la MIB. |
| | 1 | GetNextRequest-PDU | Proporciona un medio para moverse por la MIB. Petición del objeto siguiente a uno dado de la MIB. |
| | 2 | Response-PDU | Devuelve los valores solicitados por las operaciones anteriores gestor a gestor o agente a gestor. |
| | 3 | SetRequest-PDU | Permite asignar un valor a una variable. |
| | 4 | obsolete | No está en uso (utilizado para TRAP-PDU en SNMPv1) |
| | 5 | GetBulkRequest-PDU | Petición de múltiples valores. |
| | 6 | InformRequest-PDU | Transmite información no solicitada (gestor a gestor). |
| | 7 | SNMPv2-Trap-PDU | Permite a los agentes informar de sucesos inusuales |

Fuente: adaptado de Subramanian, Gonsalves, & Rani (2015) Network management: principles and practice.

Request ID: especifica el ID único de cada solicitud. Se utiliza para hacer coincidir una solicitud y la respuesta correspondiente.

Non repeaters/Max repetitions: una operación GetBulk es igual a las operaciones GetNext consecutivas. Los dos parámetros se utilizan para establecer el número de operaciones GetNext que se incluirán en una operación GetBulk.

Error status: especifica el estado de un error que se produce cuando se procesa una solicitud.

Error index: especifica el índice de un error. Si se produce una excepción, se proporciona información sobre la variable (en enlaces de variables) que causa la excepción.

Variable bindings: especifica la lista de nombres de variables y los valores correspondientes (Subramanian, Gonsalves, & Rani, 2015).

2.4. Aprendizaje automático

Según (Murphy, 2015), se define al aprendizaje automático como un conjunto de métodos que pueden detectar automáticamente patrones en los datos, y luego usar los patrones descubiertos para predecir datos futuros, o para realizar otros tipos de toma de decisiones bajo incertidumbre.

El aprendizaje automático puede contener varias técnicas para la construcción de modelos que permiten la detección automática de patrones, es decir existe un algoritmo que tiene la habilidad de clasificar los datos recolectados y presentar al usuario un resultado, por ejemplo el aprendizaje supervisado es una técnica de construcción de modelos predictivos mediante la deducción de funciones a partir datos de entrenamiento continuo de los sistemas, estos datos de entrenamiento son datos previamente clasificados, analizados y etiquetados dependiendo de los atributos que poseen, estos

datos de entrenamiento sirven como punto de comparación con los nuevos datos adquiridos (Emiliano, 2018).

2.4.1. Clasificación supervisada

La clasificación supervisada tiene su origen desde un conjunto de datos existentes que se la denomina datos de entrenamiento y cada dato está asociado a una etiqueta, se construye un modelo en la fase de entrenamiento utilizando dichas etiquetas, que nos dicen si un dato está clasificado correcta o incorrectamente por el modelo. Una vez construido el modelo se lo utiliza para clasificar nuevos datos que, en esta fase, ya no necesitan etiqueta para su clasificación, aunque sí la necesitan para evaluar el porcentaje de objetos bien clasificados (Alhojely, 2016).

2.4.1.1. Algoritmos De Clasificación a Base De Distancia

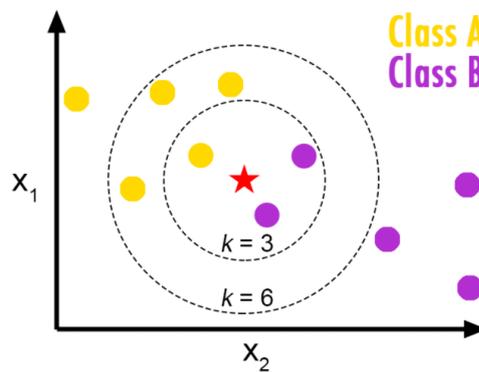
Este tipo de algoritmos se basan en la distancia que existen entre los nuevos datos que son recolectados con los datos existentes en los datos de entrenamiento, utilizando una métrica que permite la comparación de los nuevos objetos, las distancias son calculadas dependiendo la función de las diferentes clases, entre más corta sea la distancia entre el nuevo dato con el dato de entrenamiento, este dato pertenecerá a la misma clase, un algoritmo que su modelo es basado en la distancia es el algoritmo k vecinos más cercano o “k-Nearest Neighbours” (k-NN), Este algoritmo requiere el cálculo de la distancia más corta entre los datos de entrenamiento y el dato a clasificar, para así identificar a que clase pertenece (Yigit, 2013).

En la Figura 22 se muestra el diagrama de dispersión, donde se tiene dos tipos de clases, el objetivo es que cuando se ingresa un nuevo dato este se pueda clasificar y saber a qué clase puede pertenecer. Cuando se ingresa un nuevo dato (estrella), K-NN procede a clasificar y clasificar y asignarlo a una clase, el algoritmo K-NN posee una variable K

la cual representa el número de vecinos que el algoritmo debe buscar en el diagrama, cuando $K=3$ se puede observar en el diagrama que se crea un tipo de circunferencia la cual abarca a tres puntos próximos a la estrella, como se puede observar 2 de ellas son de clase B, entonces el dato(estrella) pertenecerá a la clase B. Este algoritmo clasifica al nuevo dato de acuerdo con el número de vecinos cercanos que este tenga (Salcedo, 2018).

Figura 22

Diagrama de dispersión del algoritmo K-NN.



Fuente: adaptado de Salcedo, L. (2018). Introducción al Machine Learning.

En la *Ecuación 4* se muestra como es la obtención de los K vecinos más cercanos donde se debe encontrar la distancia entre los datos a clasificar $P = (p_1, \dots, p_m)$ los datos guardados $Q = (q_1, \dots, q_m)$, para el cálculo de la distancia se utiliza la ecuación euclidiana (Salcedo, 2018).

$$d_E(P, Q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

Ecuación 4 *Formula de distancia para determinación de vecinos cercanos en KNN*

Fuente: adaptado de Salcedo, L. (2018). Introducción al Machine Learning.

2.4.1.2. Rendimiento De Modelo De Clasificación

Dado el algoritmo de clasificación es fundamental comprobar el correcto funcionamiento de este, para esto se lleva una serie de pruebas para la verificación y aceptación del modelo de clasificación.

Matriz de confusión: también conocida como matriz de errores, es una herramienta empleada para visualizar el desempeño de un algoritmo de clasificación supervisada (Jose, 2018).

En la Figura 23 se explicada cada uno de los campos: VP es la cantidad de positivos que fueron clasificados correctamente como positivos por el modelo. VN es la cantidad de negativos que fueron clasificados correctamente como negativos por el modelo. FN es la cantidad de positivos que fueron clasificados incorrectamente como negativos. FP es la cantidad de negativos que fueron clasificados incorrectamente como positivos (Jose, 2018)

Figura 23

Matriz de errores

| | | Predicción | |
|-------------|-----------|---------------------------|---------------------------|
| | | Positivos | Negativos |
| Observación | Positivos | Verdaderos Positivos (VP) | Falsos Negativos (FN) |
| | Negativos | Falsos Positivos (FP) | Verdaderos Negativos (VN) |

Fuente: adaptado de Jose, P. (2018). Data Science and Machine Learning Bootcamp with R.

Exactitud (Accuracy): la *Ecuación 5* es utilizada para determinar qué porcentaje de la data clasifica correctamente, obtenida de la división entre los verdaderos positivos, negativos y la suma de falsos y verdaderos positivos y negativos (Jose, 2018).

$$Exactitud = \frac{VP + VN}{VP + VN + FP + FN}$$

Ecuación 5 Ecuación para determinar la exactitud de un algoritmo

Fuente: adaptado de Jose, P. (2018). Data Science and Machine Learning Bootcamp with R.

Tasa de error (Misclassification Rate): la *Ecuación 6* es utilizada para determinar el porcentaje de la data clasifica incorrectamente, se obtiene de la división entre los falsos positivos, negativos y la suma de falsos, verdaderos positivos y negativos (Jose, 2018).

$$Tasa\ de\ error = \frac{FP + FN}{VP + VN + FP + FN}$$

Ecuación 6 Ecuación para determinar la tasa de error de un algoritmo

Fuente: adaptado de Jose, P. (2018). Data Science and Machine Learning Bootcamp with R.

Precisión: cuando la predicción son valores positivos, da la probabilidad de que, dada una predicción positiva, la realidad sea positiva también. Se la obtiene a partir de la ecuación *Ecuación 7* que es la división entre los verdaderos positivos y la suma de verdaderos y falsos positivos (Jose, 2018).

$$Precisión = \frac{VP}{VP + FP}$$

Ecuación 7 Ecuación para determinar la precisión de un algoritmo

Fuente: adaptado de Jose, P. (2018). Data Science and Machine Learning Bootcamp with R.

2.5. Contaminación Ambiental

La contaminación ambiental se determina por la cantidad de material particulado presente en el aire, originada por agentes (físicos, químicos o biológicos) que pueden ser nocivos para la salud, la vida vegetal o animal, debido a la gran demanda de productos los agricultores se han visto en la necesidad de usar productos químicos para mejorar la producción y control de plagas. Alterando el estado natural de un medio en el cual se desarrolla la actividad agrícola (Organización Mundial de la Salud (OMS), 2021).

- **Invernaderos**

Los invernaderos florícolas en Ecuador son muy usados para el control de temperatura y cantidad lumínica que necesita las plantas para su correcta producción y reducir pérdidas por cambios climáticos o excesos de lluvias. No obstante, así como posee ventajas también existen desventajas ya que, al ser un ambiente cerrado y el uso progresivo de pesticidas e insecticidas utilizados para control de plagas y enfermedades de la planta, dichos productos químicos producen contaminación en el aire donde los trabajadores tienen contacto directo con el material particulado en el aire y al no existir flujo de aire el material particulado se mantiene más tiempo en el aire (Villavicenci & Carrión, 2021).

2.5.1. Contaminación Del Aire

La contaminación del aire consiste en una elevada concentración de gases y partículas que flotan en el ambiente reduciendo la cantidad calidad del aire. La contaminación del aire proviene de diversas fuentes; no obstante, el sector transporte es uno de los grandes emisores de contaminantes, sobre todo en cuanto a las partículas pequeñas, y los compuestos orgánicos volátiles (COV) y los óxidos de nitrógeno (NO_x), los cuales son precursores de la formación de ozono. (Molina, 2014)

- **Químicos usados en el área florícola**

Los plaguicidas son sustancias peligrosas, que puede disponer una empresa muy fácilmente, para utilizarlas en la eliminación de plagas, pero el uso frecuente y a medida que se encuentre en contacto con las personas por un largo tiempo, provocan daños preocupantes en la salud de los trabajadores. La (Organización Mundial de la Salud, 2021) considera que existe casos de intoxicaciones por plaguicidas de los cuales el 10% corresponden a los casos letales, donde la mayoría de los plaguicidas del tipo carbamatos,

organofosforados y organoclorados corresponde a los países de tercer mundo (Marlene, 2016).

2.5.1.1.Plaguicidas

Los plaguicidas también conocidos como pesticidas son productos químicos utilizados en la agricultura, ya que son diseñados para combatir las plagas, maleza y mejorar la producción, estos productos debido a su composición química es toxica para la salud humana y el ecosistema (Organizacion Mundial de la Salud (OMS), 2019). Existen tres tipos de plaguicidas:

Insecticidas: se enfoca principalmente en evitar las plagas que atacan el follaje de las plantas, algunos ejemplos de insecticidas según su composición: organofosforados, carbamatos, organoclorados, piretrinas y piretroides (Organizacion Mundial de la Salud (OMS), 2019).

Herbicidas: se enfoca principalmente en combatir el crecimiento de plantas no deseadas consideradas como maleza, algunos ejemplos de herbicidas según su composición: compuestos clorofenilicos, pentaclorofenol, Diqua, Paraquat (Organizacion Mundial de la Salud (OMS), 2019).

Fungicidas: se enfoca principalmente en combatir o impedir el crecimiento de hongos y mohos perjudiciales para las plantas o animales. algunos ejemplos de fungicidas según su composición: Carbamatos, Compuestos de azufre, Tiocarbamatos, Etileno bisditiocarbamato, Tioftalimidas, Compuestos organometálico, Hidrocarburos halogenados, Compuestos metálicos (Organizacion Mundial de la Salud (OMS), 2019).

2.5.1.2.Efectos de plaguicidas en la salud humana

La existir exposición directa y constante con este tipo de agroquímicos causan intoxicación, enfermedades y pueden dañar los órganos del ser humano, a continuación,

se realiza una explicación de los efectos nocivos en la salud dependiendo del tipo de plaguicidas:

Efectos agudos: estos aquellos efectos que se producen de manera inmediata después de mantener una exposición directa a pequeñas dosis de plaguicidas en lapsos menores a 24 horas (Fundación PLAGBOL, 2018). En esta Tabla 6 se hace relación a los efectos agudos producidos por productos químicos utilizados en la agricultura

Tabla 6

Efectos agudos producidos por los plaguicidas en el cuerpo humano.

| MANIFESTACIONES (Signos o Síntomas) | PLAGUICIDAS QUE PRODUCEN |
|---|---|
| Arritmias cardíacas | Fluoroacetato sódico, Fumigantes halocarbonos, Nicotina, Fluoruro sódico, Óxido de etileno, Clorato sódico, Talio. |
| Taquicardia | Nitrofenoles, Pentaclorofenol, Cianamida |
| Dolor de cabeza | Organofosforados, Insecticidas carbámicos, Nicotina, Arsenicales, Inorgánicos, Mercurio orgánico, Compuestos de cadmio. |
| Depresión, estupor, coma, fallo respiratorio, muchas veces sin convulsiones | Organofosforados, Insecticidas carbámicos, Fluoruro de sodio Borato, Diquat |
| Ataques/convulsiones | Organoclorados, Estricnina, Crimidina, Fluoroacetato sódico, Nicotina, Cianuro, Acrilonitrilo, Metaldehido, Talio, DEET,. |
| Espasmos musculares | Organofosforados, Insecticidas carbámicos, Nicotina, Fluoruro de sulfurilo. |

Fuente: adaptado de Fundación PLAGBOL. (2018). Manual de Diagnóstico Tratamiento y Prevención de Intoxicaciones Agudas por plaguicidas

Efectos crónicos: estos efectos se producen después de un tiempo prolongado de mantener contacto directo a pequeñas dosis de plaguicidas por lapsos mayores a semanas,

meses o años (Fundación PLAGBOL, 2018). En esta Tabla 7 se hace relación a los efectos crónicos producidos por productos químicos utilizados en la agricultura.

Tabla 7

Efectos crónicos producidos por los plaguicidas en el cuerpo humano.

| EFFECTOS CRÓNICOS | PLAGUICIDAS |
|---------------------------------------|---|
| Lesiones del Sistema Nervioso Central | Insecticidas órganoclorados y organofosforados; fungicidas mercuriales. |
| Dermatitis de contacto | Paraquat; captafol; 2,4-D y mancozeb. |
| Formación de cataratas | Diquat. |
| Cistitis hemorrágica | Clordimeform. |
| Neumonitis y fibrosis pulmonar | Paraquat |
| Disminución del índice de fertilidad | Captán (en animales y posiblemente en hombres) y el Agente Naranja (2,4-D + 2,4,5-T). |

Fuente: adaptado de Fundación PLAGBOL. (2018). Manual de Diagnóstico Tratamiento y Prevención de Intoxicaciones Agudas por plaguicidas

2.5.2. Indicadores De Calidad Del Aire

Los indicadores de calidad del aire son sustancias contaminantes atmosféricos que se emiten constantemente y son los causantes de afectar a la salud de los seres vivos, a continuación, se mencionara algunos de los más importantes indicadores de calidad del aire y sus respectivos valores establecidos por (Organización Mundial de la Salud (OMS), 2021). En la Tabla 8 se presenta los valores límites, para cada contaminante común de la atmósfera, junto con el código de colores según su toxicidad.

Monóxido De Carbono: El monóxido de carbono (CO) es un tipo de gas toxico para los seres vivos, es un gas incoloro e inoloro, todas las personas están expuestas a este gas, pero la inhalación en grandes cantidades puede llegar a ser mortal. La principal fuente

es los motores de combustión fósil o biomasa. (Morera & Araya, 2017). Los límites fijados para el monóxido de carbono son: 35 ppm por 8 horas.

Óxidos De Nitrógeno (NOx): Los motores a combustión son los principales emisores de NOx, Los NOx representan a una familia de siete compuestos. En realidad, la EPA regula sólo el dióxido de nitrógeno (NO2) como un suplente para esta familia de compuestos porque es la forma más predominante de NOx en la atmósfera que es generada por actividades antropogénicas. El NO2 no es sólo un contaminante importante del aire por sí sólo, sino que también reacciona en la atmósfera para formar ozono troposférico (O3) y lluvia ácida. (ZAMBRANO, 2016) Los límites fijados para el dióxido de nitrógeno son:

- 40 ug/m3 de media anual.
- 200 ug/m3 de media en 1h.

Tabla 8

Límites numéricos de cada categoría de índices de calidad de aire en (ppm).

| Categoría | CO | NOx | C3H8 | PM2.5 | PM10 |
|---------------------------------|-------------|------------|-------------|--------------|-------------|
| Nivel deseable u óptimo. | 1 – 2.4 | 0 - 5 | 0 - 100 | 0 - 25 | 0 - 50 |
| Nivel aceptable o bueno. | 2.5 – 4.4 | 5 – 10 | 101- 500 | 26 – 50 | 51 – 100 |
| Nivel de precaución. | 4.5 – 6.4 | 10 – 30 | 500 – 1000 | 51 – 150 | 101 – 250 |
| Nivel de alerta. | 8.5 – 15.4 | 30 -50 | 1001 – 2000 | 151 -250 | 251 – 400 |
| Nivel de alarma. | 15.5 – 30.4 | 50 – 110 | 2001 – 3000 | 251 – 350 | 401 – 500 |
| Nivel de emergencia. | >30.5 | >140 | >3000 | >350 | >500 |

Fuente: adaptado de Municipio del Distrito Metropolitano de Quito. (2017). Informe anual 2017 de la calidad del aire en Quito.

CAPITULO III. Requerimientos y diseño

En el presente capítulo, se presenta un análisis de la situación actual donde se realiza una descripción general de todo el proceso de diseño y elaboración del sistema de alerta y monitoreo, tanto en funcionamiento como en estructura de hardware y software para el cual se utilizará la metodología especificada por estándar ISO/IEC/IEEE 15288 que describe los procesos del ciclo de vida del sistema.

3.1. Metodología

3.1.1. Estándar ISO/IEC/IEEE 15288

ISO / IEC / IEEE 15288 es el estándar internacional para los procesos del ciclo de vida del sistema. Aunque fue desarrollado bajo el Comité Técnico Conjunto de tecnología de la información, este sistema de ciclo de vida framework es extensible a muchos dominios del sistema. Porque cubre todo el ciclo de vida y es aplicable a todos los niveles de la jerarquía arquitectónica del sistema, este estándar ha demostrado ser útil a todos los miembros del equipo involucrados en el desarrollo y mantenimiento de un sistema, no solo sistemas ingenieros (ISO / IEC / IEEE 15288, 2015).

ISO / IEC / IEEE 15288: 2015 también proporciona procesos que respaldan la definición, el control y la mejora de los procesos del ciclo de vida del sistema utilizados dentro de una organización o un proyecto. Las organizaciones y proyectos pueden utilizar estos procesos al adquirir y suministrar sistemas (ISO / IEC / IEEE 15288, 2015).

ISO / IEC / IEEE 15288: 2015 se refiere a aquellos sistemas que son hechos por el hombre y pueden configurarse con uno o más de los siguientes elementos del sistema: hardware, software, datos, humanos, procesos (por ejemplo, procesos para proporcionar servicio a los usuarios), procedimientos (por ejemplo, instrucciones del operador),

instalaciones, materiales y entidades naturales (ISO / IEC / IEEE 15288, 2015). En la Figura 24 se muestra cada nivel o etapa del estándar ISO/IEC/IEEE 15288 y los procesos de ciclo de vida del sistema.

Figura 24

Proceso del ciclo de vida del sistema

| ISO/IEC 15288:2008 System Life Cycle Process | ISO/IEC 15288:2008 Proceso del ciclo de vida del sistema |
|---|--|
| <i>6.4 Technical Processes</i> | <i>Procesos Técnicos</i> |
| 6.4.1 Business or Mission Analysis | Análisis de Negocios o Misiones |
| 6.4.2 Stakeholder Needs and Requirements Definition | Necesidades de las partes interesadas y Definición de requisitos |
| 6.4.3 Systems Requirements Definition | Definición de requisitos de sistemas |
| 6.4.4 Architectural Definition | Definición arquitectónica |
| 6.4.5 Design Definition | Definición de diseño |
| 6.4.6 System Analysis | Análisis de sistemas |
| 6.4.7 Implementation | Implementación |
| 6.4.8 Integration | Integración |
| 6.4.9 Verification | Verificación |
| 6.4.10 Transition | Transición |
| 6.4.11 Validation | Validación |
| 6.4.12 Operation | Operación |
| 6.4.13 Maintenance | Mantenimiento |
| 6.4.14 Disposal | Disposición |

Fuente: adaptado de ISO / IEC / IEEE 15288: 2015. (2015). System and software engineering - system life cycle processes

3.2. Análisis de la Situación Actual

De acuerdo con uno de los objetivos de organización mundial de la salud OMS (Organización Mundial de la Salud, 2017) con respecto a productos agroquímicos, es proteger la salud pública mediante el establecimiento de límites máximos de exposición y límites máximos de residuos de los plaguicidas en los alimentos y el agua, además, según una encuesta realizada por (Venegas, Cortés, Baltazar, & Serrano, 2016) el 23 %

(nunca) y el 28% (a veces) de trabajadores no tienen el conocimiento de la toxicidad de los agroquímicos utilizados en el cultivo de flores y los efectos que causan a largo y corto plazo en la salud. Ya que los trabajadores del área de cultivo se exponen a lapsos prolongados a un ambiente con residuos de estos compuestos químicos debido a las contantes fumigaciones. Y debido a que es un sector el cual proporciona fuente de trabajo contando con alrededor de 422 empresas florícolas a nivel nacional, es necesario la implementación de un sistema que permita determinar la calidad del aire dentro de los invernaderos para lograr así precautelar la salud de los trabajadores.

Además, que en la actualidad el uso las redes de sensores inalámbricos ha tenido gran acogida por su amplia gama de aplicaciones, es necesario la implementación de modelos de gestión de redes, por lo cual este sistema se centra la implantación de un sistema que sea gestionable por el administrador de red.

3.3.Técnica De Investigación

Debido a que no existe registros de los niveles de concentración de gases y material particulado dentro de los invernaderos de producción florícola, es importante la recolección de información por lo cual se usa la investigación de campo, realizando la observación directa en el invernadero florícola. Logrando así establecer los criterios necesarios para el desarrollo del sistema de alerta y monitoreo.

3.3.1. Observación Directa En Invernadero Florícola.

A través de la observación directa se puede determinar de manera precisa como es el ambiente en el cual trabajan los encargados de cultivo y fumigación de las rosas, logrando observar que al momento que se realiza la fumigación es cuando más concentración de productos químicos en el aire, además de ser visible se puede percibir

mediante el olfato. En la Figura 25 se puede observar cómo es visible los niveles de concentración de producto químicos en el aire.

Figura 25

Invernadero florícola mientras se realiza la fumigación.



Fuente: elaborado por el autor

3.3.2. Entrevista

De acuerdo con la entrevista realizada con el Ingeniero Agrónomo De la Cruz Pedro jefe de la finca florícola, se obtuvieron los resultados que se presentan en el ANEXO A, en base a esta entrevista se realizó el análisis para lograr establecer los requerimientos que debe cumplir el sistema.

El sistema debe poseer un diseño que permita la optimización de espacio y recursos para una fácil implementación y gestión.

El sistema debe permanecer activo durante un tiempo superior a 1 hora al día, debido a que el periodo de fumigación es alrededor de 1 hora, por lo cual es sistema debe

estar activo un tiempo antes y después de la fumigación, hasta que las concentraciones de material particulado se encuentren en niveles permitidos.

El sistema debe poseer un número de nodos sensores que permita determinar la concentración de material particulado en todo el invernadero, debido a que estas particuladas se extienden a lo largo de todo el invernadero.

El sistema debe poseer sensores para detectar los componentes activos de los fungicidas e insecticidas usados para el control de plagas y hongos que afectan a las plantas.

El sistema debe poseer una interfaz que permita visualizar las concentraciones de material particulado en el invernadero. Además de proporcionar alerta cuando los niveles de concentración excedan los permitidos.

3.4. Definición de requerimientos

3.4.1. Descripción General Del Sistema.

El sistema de alerta y monitoreo consta de varios nodos, un Gateway y una estación base, cada nodo consta de una unidad de recolección de datos mediante sensores, unidad de procesamiento y unidad de comunicación inalámbrica bajo el protocolo IEEE 802.11, cada nodo tiene como función la recolección de datos, censado constante del ambiente y procesamiento de los datos recolectados mediante algoritmos de clasificación supervisado que permitirán la generación de un índice de calidad del aire.

El Gateway o puerta de enlace es la encargada del encaminamiento de los datos bajo la arquitectura TCP/IP enviados por los nodos sensores hacia la estación base, todos los nodos se conectan de forma inalámbrica a él Gateway formando una topología tipo estrella.

La estación base posee una unidad de comunicación inalámbrica la cual se conecta de manera inalámbrica al Gateway, de esta manera mantiene comunicación entre los nodos sensores, la estación base tiene como objetivo la adquisición y visualización de los datos enviados por cada nodo sensor.

Cada nodo sensor posee en su estructura una agente SNMP el cual es encargado de la generación de la MIB donde se recolecta cada uno de los estados de los recursos del nodo que se desee monitorizar, la estación base posee al gestor el cual realizara las peticiones de las MIBs a cada uno de los nodos cuando sea necesario. El envío y recepción de mensajes SNMP serán enviadas de forma inalámbrica bajo el estándar de comunicación IEEE 802.11 y la arquitectura TCP/IP utilizando el puerto UDP 161 y 162.

3.4.2. Definición De Requerimientos Del Sistema

Mediante el análisis de la información recolectada de las diferentes tecnologías y modelos de información relacionadas en la implementación de este proyecto, se puede determinar los elementos y requerimientos necesarios para cumplir con los objetivos planteados, esto permite poner a evaluación los requerimientos de usuario, requerimientos de sistema y requerimientos de arquitectura.

3.4.2.1. Construcción De Atributos De Requerimientos

La construcción de atributos se elabora en base a tres requerimientos: Stakeholders, sistema y arquitectura, cada requerimiento debe ser definido de tal manera que posea una estrecha relación con los requerimientos de cada uno de los aspectos a cumplir del proyecto, además de que cada requerimiento debe ser verificable, medible y limitable.

3.4.2.2.Nomenclatura De Requerimientos

Para cada uno de los requerimientos necesarios para el desarrollo del proyecto, es necesario asignarles una abreviatura que permita un mejor manejo de los datos. En la Tabla 9 se muestra las abreviaturas a utilizar.

Tabla 9

Abreviatura de requerimientos

| Requerimiento | Abreviatura |
|----------------------|--------------------|
| Stakeholders | STSR |
| Sistema | SYSR |
| Arquitectura | SRSR |

Fuente: elaborado por el autor

3.4.2.3.Requerimientos De Stakeholders

Los Stakeholders son las partes involucradas directas e indirectamente con el desarrollo del este proyecto. Donde los involucrados proporcionan las características necesarias para el desarrollo y diseño del sistema. En la Tabla 10 se presentan los involucrados de la presenta investigación

Tabla 10

Lista de involucrados directos e indirectos

| Numero de orden | Descripción | Nombre |
|------------------------|-------------------------|-----------------------|
| 1 | Agricultores florícolas | |
| 2 | Jefe finca florícola | Msc. Pedro De La Cruz |
| 3 | Tutor | Msc. Vásquez Carlos |
| 4 | Asesor | Msc. Michilena Jaime |
| 6 | Desarrollador | Sr. Benavides Wilmer |

Fuente: elaborado por el autor

El propósito del proceso de definición de Stakeholders, es definir los requisitos de un sistema que puede proporcionar las capacidades que necesitan los usuarios en un

entorno definido, siendo el entorno el invernadero florícola y como usuarios a los trabajadores y propietario. En la Figura 11 se evalúan los Stakeholders operacionales y de usuario.

Tabla 11

Requerimientos de Stakeholders

| STSR | | | | |
|---------------------------------------|---|-------------------------------------|-------------------------------------|--------------------------|
| Requerimientos De Stakeholders | | | | |
| Número | Requerimientos | Prioridad | | |
| | | Alta | Media | Baja |
| Requerimientos operacionales | | | | |
| STSR1 | Adecuada ubicación de los nodos en el invernadero florícola. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| STSR2 | Sistema debe poseer un rango de cobertura para todo el invernadero. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| STSR3 | Los nodos sensores deben detectar la concentración de material particulado de plaguicidas, herbicidas e insecticidas. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| STSR4 | El sistema debe generar una alerta cuando los niveles de concentración de gas superen los niveles permitidos. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| STSR5 | El sistema debe poseer un diseño optimizado para la fácil implementación. | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| STSR6 | El sistema debe poseer de bajo costo de implementación. | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| STSR7 | El sistema debe permanecer activo por un tiempo mayor a 1 hora diaria. | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Requerimientos del usuario | | | | |
| STSR8 | El usuario no debe manipular ningún componente de hardware y software de los nodos y Gateway. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| STSR9 | El usuario no debe modificar ningún parámetro del servidor de gestión. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

| | | | | |
|---------------|---|-------------------------------------|--------------------------|--------------------------|
| STSR10 | Interfaz gráfica para visualización los índices de concentración de material particulado de fácil interpretación para el usuario. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| STSR11 | El sistema debe mostrar al usuario el estado de los recursos de cada nodo. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| STSR12 | El sistema debe notificar al usuario en caso de ocurrir un evento inesperado en el estado del nodo. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Fuente: elaborado por el autor

3.4.2.4.Requerimientos Del Sistema

Según (ISO / IEC / IEEE 15288, 2015), la definición de requerimientos del sistema crea un conjunto de requisitos mensurables del sistema que especifican, desde la perspectiva del proveedor, qué características, atributos y requisitos funcionales y de desempeño debe poseer el sistema, en para satisfacer los requisitos de las partes interesadas. Por tanto, los requerimientos que se va a analizar son los requerimientos de uso, performace, interfaces, modos, estados y físicos que tengan relación con los requerimientos de Stakeholders. En la Tabla 12 se establecen los requerimientos del sistema necesarios para el correcto funcionamiento y diseño del sistema.

Tabla 12

Requerimientos del sistema

| SYSR | | | | |
|-----------------------------------|--|-------------------------------------|--------------------------|--------------------------|
| Requerimientos del sistema | | | | |
| Numero | Requerimientos | Prioridad | | |
| | | Alta | Media | Baja |
| Requerimientos de uso | | | | |
| SYSR1 | Sensores sensibles a diferentes concentraciones de material particulado. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SYSR2 | Fuente de alimentación portable (baterías) a 5 Voltios para el encendido del sistema embebido. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SYSR3 | Los nodos sensores deben ubicarse estratégicamente en el invernadero. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

| Requerimientos de performace | | | | |
|--|--|-------------------------------------|-------------------------------------|--------------------------|
| SYSR4 | Procesamiento de los datos obtenidos por los diferentes sensores. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SYSR5 | Visualización grafica del índice de calidad del aire. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SYSR6 | Agente SNMP adaptado para cada nodo sensor. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SYSR7 | Visualización grafica del estado de los recursos de los nodos sensores. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SYSR8 | Facilidad de administración de los recursos de la red WSN. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Requerimientos de interfaces | | | | |
| SYSR9 | Cada nodo debe tener una cantidad de pines análogos mayor a 5. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SYSR10 | Un switch de encendido y apagado. | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| SYSR11 | Módulo de comunicación inalámbrico compatible con el protocolo IEEE 802.11g. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Requerimientos de modos/estados | | | | |
| SYSR12 | El sistema proporcionara una alerta cuando los índices de calidad del aire sean mayores a los índices permitidos. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SYSR13 | El sistema proporcionara una alerta cuando existan eventos imprevistos en los nodos sensores y Gateway. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Requerimientos físicos | | | | |
| SYSR14 | Los nodos sensores deben montarse en un case que permita su fácil movilidad, instalación y permita proteger de daños en los componentes. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Fuente: elaborado por el autor

3.4.2.5.Requerimiento De Arquitectura

Los requerimientos de arquitectura hacen referencia a los componentes hardware, software y elementos eléctricos del sistema con las características en base al funcionamiento que tendrá el sistema. Para lograr cubrir los requisitos de la arquitectura

se tomará los requerimientos lógicos, de diseño, de hardware, software y eléctricos. En la Tabla 13 se describe los requerimientos de arquitectura.

Tabla 13

Requerimientos de arquitectura

| SRS | | | | |
|---------------------------------------|---|------------------|--------------|-------------|
| Requerimientos de arquitectura | | | | |
| Numero | Requerimientos | Prioridad | | |
| | | Alta | Media | Baja |
| Requerimientos lógicos | | | | |
| SRS1 | Comunicación inalámbrica entre nodos y el Gateway bajo el protocolo IEEE 802.11g | ■ | □ | □ |
| Requerimientos de diseño | | | | |
| SRS2 | Diseño de placa donde se acople los sensores, microcontrolador y sistema de alimentación eléctrica | ■ | □ | □ |
| SRS3 | Los sensores deben ubicarse de manera que no interfieran con el microcontrolador. | ■ | □ | □ |
| SRS4 | Envío y recepción de datos vía inalámbrica. | ■ | □ | □ |
| SRS5 | La topología de la WSN debe ser tipo estrella. | ■ | □ | □ |
| SRS6 | Case que contenga al nodo sensor y no interfiera en la toma de datos. | ■ | □ | □ |
| SRS7 | Sistema de bajo costo energético. | □ | ■ | □ |
| SRS8 | Base de datos para almacenamiento de valores enviados por los nodos sensores | ■ | □ | □ |
| Requerimientos de software | | | | |
| SRS9 | Lenguaje de programación de código abierto. | ■ | □ | □ |
| SRS10 | Soporte con lenguaje de programación orientada a objetos. | ■ | □ | □ |
| SRS11 | Plataforma de programación con editor de código, compilador, depurador y constructor de interfaz gráfica (GUI). | ■ | □ | □ |
| SRS12 | Soporte de Librerías de sensores y módulos de comunicación inalámbricos. | ■ | □ | □ |

| | | | | |
|---------------|---|-------------------------------------|-------------------------------------|--------------------------|
| SRSH13 | Compatibilidad con Protocolo de Wi-Fi IEEE 802.11g | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SRSH14 | Compatibilidad con Protocolos de red como IPv4/TCP/UDP/HTTP/ICMP. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SRSH15 | Compatibilidad con modelo de gestión de redes SNMP. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SRSH16 | Soporte para monitorizar vía protocolos SNMP. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SRSH17 | Flexibilidad para la notificación de eventos. | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| SRSH18 | Compatibilidad con base de datos MySQL en base Linux. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SRSH19 | Software de monitorización de redes con interfaz web. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Requerimientos de hardware

| | | | | |
|---------------|--|-------------------------------------|-------------------------------------|--------------------------|
| SRSH20 | Sistema embebido con más de 5 puertos ADC. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SRSH21 | Sistema embebido con alta capacidad de procesamiento. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SRSH22 | Sistema embebido con módulo de comunicación Wi-Fi compatible con IEEE 802.11g. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SRSH23 | Sensor debe detectar Monóxido de carbono (CO) | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SRSH24 | Sensor debe detectar Dióxido de carbono (CO ₂) | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| SRSH25 | Sensor debe detectar amonio (NH ₄) | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SRSH26 | Sensor debe detectar Óxidos de nitrógeno (NO _x) | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SRSH27 | Sensor debe detectar Alcohol, Sulfuros, Benceno (C ₆ H ₆) | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| SRSH28 | Sensor debe detectar propano, hidrógeno y metano (CH ₄). | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SRSH29 | Sensor debe detectar humedad relativa, temperatura. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| SYSH30 | Sistema embebido que trabaje a 5V. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Requerimientos físicos

| | | | | |
|---------------|--|-------------------------------------|--------------------------|--------------------------|
| SYSH31 | Los nodos sensores deben montarse en un case que permita su fácil movilidad, instalación y permita proteger de daños en los componentes. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
|---------------|--|-------------------------------------|--------------------------|--------------------------|

Requerimientos Eléctricos

| | | | | |
|---------------|--|---|---|---|
| SYSH32 | Fuente de alimentación recargable de 5V. |  |  |  |
| SYSH33 | Fuente de alimentación externa para recargar la batería. |  |  |  |
| SYSH34 | Conexión GND y VCC para los sensores |  |  |  |

Fuente: elaborado por el autor

3.5. Elección de hardware y software

3.5.1. Elección De Hardware

Una vez establecidos los requerimientos del sistema es posible la elección de hardware como software que requiere el sistema, esta elección se la realiza mediante una tabla comparativa de las especificaciones de cada dispositivo tecnológico a utilizar cumpliendo con los requerimientos de arquitectura, estas especificaciones se les dará un valor referencial para detallar si cumple o no cumple el requerimiento. En la Tabla 14 se muestra cómo se establece el valor de “1” para si cumple y “0” para no cumple.

Tabla 14

Valor referencial para requerimientos

| Descripción | Valor |
|--------------------|--------------|
| Cumple | 1 |
| No cumple | 0 |

Fuente: elaborado por el autor

3.5.1.1. Sistema Embebido Para Nodo sensor

Para la elección el microcontrolador que estará encargado del procesamiento de los datos, la gestión de los sensores para la recolección de datos y la compatibilidad con el modelo de gestión SNMP para el monitoreo, este debe cumplir con los requerimientos del sistema anteriormente especificados, en la Tabla 15 se muestra un listado de los microcontroladores que podrían adaptarse a los requerimientos solicitados.

Tabla 15

Elección de Hardware para Nodo Sensor

| Hardware | Requerimiento | | | | | | | Valoración total |
|-----------------|---------------|--------|--------|--------|--------|--------|------|------------------|
| | SRSH11 | SRSH12 | SRSH18 | SRSH19 | SRSH20 | SRSH28 | YSR9 | |
| Arduino Due | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 5 |
| Arduino Mega | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 5 |
| NodeMCU Esp8266 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 5 |
| NodeMCU Esp32 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 7 |

Elección: se eligió el microcontrolador NodeMCU esp32 cumple con todos los requerimientos de funcionamiento, además del bajo costo y tamaño reducido incorpora una antena Wi-Fi y Bluetooth por lo cual es el ideal para los Nodos sensores de la red WSN.

Fuente: elaborado por el autor

En la Tabla 16 se muestra las principales características técnicas del módulo NodeMCU esp32.

Tabla 16

Características técnicas del microcontrolador NODEMCU ESP32

| NodeMCU esp32 | |
|-------------------------------|-------------------------------|
| Interfaz | Ethernet, I2C, I2S, SPI, UART |
| Frecuencia | 2,4 GHz ~ 2,5 GHz |
| Temperatura de funcionamiento | -40 °C ~ 85 °C |
| Voltaje de alimentación | 5 V |



| | |
|-------------------------|--|
| Corriente de suministro | 500 mA |
| Número de canales ADC | 16 |
| Número de GPIO | 32 |
| ROM | 448KB, para arranque y funciones básicas |
| Procesador principal | Tensilica Xtensa LX6 de 32 bits |
| Protocolos Wi-Fi | IEEE 802.11 b/g/n (802.11n up to 150 Mbps) |

Fuente: adaptado de Espressif Systems. (2021). ESP32 Series.

3.5.1.2.Sistema Embebido Para Gateway

Para la selección del microcontrolador encargado de la interconectar de todos los elementos de la red de sensores y basados en los requerimientos de diseño establecidos anteriormente, en la Tabla 17 se realiza una comparativa de los posibles candidatos para el Gateway.

Tabla 17

Elección de Sistema embebido para Gateway

| Hardware | Requerimiento | | | | | | | Valoración total |
|-----------------|---------------|-------|--------|--------|--------|--------|--------|------------------|
| | SYSR11 | SRSH1 | SRSH11 | SRSH12 | SRSH13 | SRSH19 | SRSH20 | |
| Arduino Due | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 5 |
| Arduino Mega | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 5 |
| NodeMCU Esp8266 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 7 |
| Raspberry Pi. | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 5 |

Elección: se eligió el microcontrolador NodeMCU esp8266 cumple con todos los requerimientos de funcionamiento, bajo costo y tamaño reducido e incorpora una antena Wi-Fi.

Fuente: elaborado por el autor

En la Tabla 18 se muestra las principales características técnicas del módulo NodeMCU esp8266.

Tabla 18*Características técnicas NodeMCU esp8266***NodeMCU esp8266**

| | |
|---------------------------------|-----------------------------------|
| Velocidad de datos | 54 Mbps |
| Rango de frecuencia | 2.4GHz-2.5GHz (2400MHz-2483.5MHz) |
| Rango de temperatura | - 40°~125° |
| Voltaje de trabajo | 3.0~3.6V |
| Corriente de suministro nominal | 500 mA |
| WiFi mode | station/softAP/SoftAP+station |
| Seguridad Wi-Fi | WPA/WPA2 |
| Network Protocols | IPv4, TCP/UDP/HTTP/FTP |
| Protocolos Wi-Fi | IEEE 802.11 b / g / n |

Fuente: adaptado de Espressif Systems IOT Team. (2015). ESP8266EX Datasheet

3.5.1.3.Sensores De Gas

Dentro de las familias de agroquímicos más usados en el área florícola, existen compuestos químicos que al estar en constante contacto son tóxicos para la salud A entre ellos se tiene Amoníaco (NH₃), Óxidos de nitrógeno (NO_x), Alcohol, Sulfuros, Benceno, (C₆H₆), Metano(CH₄) , Monóxido de carbono (CO), dióxido de carbono (CH₂), humo, por lo cual se ha tomado en cuenta los sensores de la familia MQ los cuales son los ideales para la detección y además cumplen con los requerimientos establecidos anteriormente,

en la Tabla 19 se hace una comparativa de los sensores que pueden cumplir los requerimientos.

Tabla 19

Elección de sensores de gases

| Hardware | Requerimiento | | | | | | | Valoración total |
|----------|---------------|--------|--------|--------|--------|--------|--------|------------------|
| | SYSR21 | SRSH22 | SRSH23 | SRSH24 | SRSH25 | SRSH26 | SRSH27 | |
| MQ- 2 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 2 |
| MQ- 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| MQ-8 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| MQ-7 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 2 |
| MQ-9 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| DTH11 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 |
| MQ-135 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 4 |

Elección: se eligió los sensores MQ-2, MQ-7, MQ-135, DTH11 ya que cumplen con los requerimientos establecidos.

Fuente: elaborado por el autor

Sensor MQ-2

El MQ-2 Sensor de Gas tiene sensibilidad especial para medir concentraciones de gas en el aire, es sumamente utilizado para medir LPG, propano, hidrógeno, metano y otros combustibles. En la Tabla 20 se especifica las principales características.

Tabla 20

Características técnicas del sensor MQ-2

| | |
|-------------------------------|--|
| MQ-2 |  |
| Voltaje de Operación adecuado | 5V DC |
| Rango de detección | 300 a 10000 ppm |
| Resistencia de censado | 1K Ω 50ppm |
| Tiempo de Respuesta | 10s |
| Tiempo de recuperación | 30s |
| Temperatura de trabajo | -20 °C ~ +55 °C |
| Humedad | \leq 95% RH |
| Corriente | 150mA a 5V |
| Concentración de oxígeno | 21% |

Fuente: adaptado de Hanwei Eletronics co.,ltd (2015) Technical Data MQ-2 Sensor

Sensor MQ-7

El sensor MQ-7 permite medir gas de Monóxido de Carbono (CO), ideal para detectar concentraciones dañinas de CO en el aire y así evitar sus daños en la salud. El sensor MQ-7 puede detectar concentraciones en el rango de 20 a 2000ppm. En la Tabla 21 se especifican las características técnicas del sensor.

Tabla 21

Características técnicas del sensor MQ-7

| | |
|-------------------------------|--|
| MQ-7 |  |
| Voltaje de Operación adecuado | 5V DC |
| Rango de detección | 20ppm- 2000ppm |
| Temperatura de trabajo | -20 °C ~ +50°C |
| Humedad | ≤ 95% RH |
| Corriente | 150mA a 5V |
| Concentración de oxígeno | 21% |

Fuente: adaptado de Hanwei Eletronics co.,ltd (2015) Technical Data MQ-7 Sensor

Sensor MQ-135

Sensor de calidad de aire con alta sensibilidad al Amoníaco (NH₃), óxidos de nitrógeno (NO_x), alcohol, sulfuros, benceno (C₆H₆), CO₂, humo y otros gases nocivos. El sensor MQ-135 es sensible a otros gases en el aire, por lo que no es posible obtener buenos valores de CO₂. En la Tabla 22 se muestra las características técnicas del sensor MQ-135.

Tabla 22

Características técnicas del sensor MQ-135

| | |
|-------------------------------|--|
| MQ-135 |  |
| Voltaje de Operación adecuado | 5V DC |
| Rango de detección | 10ppm - 300ppm NH ₃ |

| | |
|-------------------------------|-------------------------|
| | 10ppm - 1000ppm Benceno |
| | 10ppm - 300ppm Alcohol |
| Temperatura de trabajo | -20 °C ~ +50°C |
| Humedad | ≤ 95% RH |
| Temperatura de uso | -10°C – 45°C |
| Temperatura de almacenamiento | -20°C – 70°C |
| Corriente | 150mA a 5V |
| Concentración de oxígeno | 21% |

Fuente: adaptado de Hanwei Eletronics co.,ltd (2015) Technical Data MQ-135 Sensor

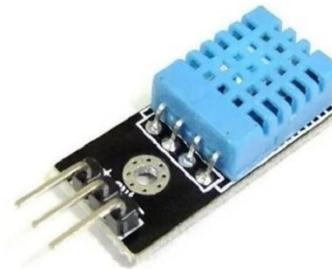
Sensor DTH11

El DHT11 es un sensor de temperatura y humedad con salida digital calibrada. Su tecnología garantiza la alta fiabilidad y una excelente estabilidad a largo plazo. En la Tabla 23 se muestra las características técnicas del sensor.

Tabla 23

Características técnicas del sensor DTH11

DTH11



| | |
|----------------------------|----------------------------------|
| Rango medición TEMPERATURA | 0°C a 60°C. |
| Error medición TEMPERATURA | ±2°C. |
| Rango medición HUMEDAD | 20% a 90% RH (Humedad Relativa). |
| Voltaje alimentación | 3,3v a 5v |

Fuente: adaptado de Hanwei Eletronics co.,ltd (2015) Technical Data DTH11 Sensor

3.5.2. Elección De Software

Para la elección del software que se va a utilizar para el desarrollo del proyecto se debe tomar en cuenta los diversos requerimientos de software, arquitectura y compatibilidad con los requerimientos de hardware que se especificaron anteriormente.

3.5.2.1. Lenguaje De Programación

Para la elección del lenguaje de programación se debe tomar en cuenta que se va a trabajar con datos que requieren un lenguaje orientado a objetos y con compatibilidad de librerías, además se permitir ejecutarse antes de implementarse para evitar problemas en el sistema embebido. En la Tabla 24 se realiza la comparativa de los diversos lenguajes de programación.

Tabla 24

Elección de lenguaje de programación

| Software | Requerimiento | | | | | | Valoración total |
|-----------|---------------|------|-------|-------|-------|-------|------------------|
| | SYSH8 | SRS9 | SRS11 | SRS12 | SRS13 | SRS14 | |
| C ++ | 1 | 1 | 1 | 1 | 1 | 1 | 6 |
| Javascríp | 0 | 1 | 0 | 1 | 1 | 1 | 4 |
| Python | 1 | 0 | 1 | 1 | 1 | 1 | 5 |

Elección: se eligió el lenguaje de programación C ++ ya que cumple con todos los requerimientos necesarios además que es necesario la compilación del código antes de su ejecución.

Fuente: elaborado por el autor

C ++

C++ es un lenguaje de programación orientada a objetos que proviene de la extensión del lenguaje C para que pudiese manipular objetos.

Algunas de las características más importantes que posee el lenguaje C++ son:

- a) **Compatibilidad con bibliotecas:** A través de bibliotecas hay muchas funciones que están disponible y que ayudan a escribir código rápidamente.
- b) **Orientado a Objetos:** El foco de la programación está en los objetos y la manipulación y configuración de sus distintos parámetros o propiedades.
- c) **Rapidez:** La compilación y ejecución de un programa en C++ es mucho más rápida que en la mayoría de los lenguajes de programación.
- d) **Compilación:** En C++ es necesario compilar el código de bajo nivel antes de ejecutarse, algo que no ocurre en otros lenguajes.
- e) **Didáctico:** Aprendiendo programación en C++ luego es mucho más fácil aprender lenguajes como Java, C#, PHP, Javascript, etc. (Carballeira, 2015)

3.5.2.2. Plataforma de desarrollo de código

Para la elección de la plataforma para el desarrollo del código es necesario que cumpla con los requerimientos de software, y compatibilidad con los sistemas embebidos. además de compatibilidad con lenguaje de programación C ++, permita añadir librerías y que posea un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). En la Tabla 25 se realiza la comparativa de las diversas plataformas de desarrollo.

Tabla 25

Elección de la plataforma de desarrollo

| Software | Requerimiento | | | | Valoración total |
|-------------|---------------|-------|--------|--------|------------------|
| | SYSH8 | SRSH9 | SRSH10 | SRSH11 | |
| Arduino IDE | 1 | 1 | 1 | 1 | 4 |
| MiniBloq | 0 | 1 | 0 | 1 | 2 |

| | | | | | |
|-------------|---|---|---|---|---|
| ArduBlock | 1 | 1 | 0 | 0 | 2 |
| MicroPython | 1 | 0 | 1 | 1 | 3 |

Elección: se eligió la plataforma de desarrollo Arduino IDE ya que cumplen con todos los requerimientos y compatibilidad requerida.

Fuente: elaborado por el autor

ARDUINO IDE

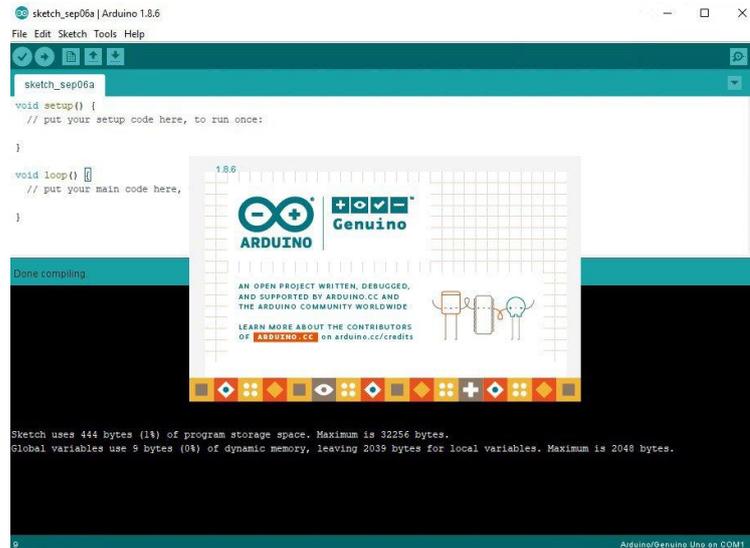
El entorno de desarrollo integrado Arduino, o software Arduino (IDE), contiene un editor de texto para escribir código, un área de mensajes, una consola de texto, una barra de herramientas con botones para funciones comunes y una serie de menús. Se conecta al hardware Arduino para cargar programas y comunicarse con ellos. (ARDUINO, 2022) Algunas de las características se mencionan a continuación:

- a) Soporte multiplataforma de arduino
- b) Detección automática de la placa conectada
- c) Muestra memoria Flash y SRAM ocupada por un sketch o proyecto
- d) Autoguardado al compilar y cargar sketch
- e) Carga de sketch vía red (wifi o ethernet).

En la Figura 26 se muestra el entorno de trabajo de la plataforma de desarrollo de código abierto Arduino IDE.

Figura 26

Entorno de Arduino IDE



Fuente: adaptado de ARDUINO (2022) What is Arduino?

3.5.2.3. Software Para El Monitoreo De Redes

Para la elección del software para el monitoreo de redes debe cumplir con los requerimientos de software establecidos anteriormente, que permitirán observar los recursos de los dispositivos de la red utilizando modelo de gestión de red internet SNMP, proporcionando alertas flexibles sobre los eventos que se generen en la red WSN y una visualización en una interfaz gráfica web. En la Tabla 26 se muestra una comparativa de los diferentes softwares para el monitoreo de red.

Tabla 26

Elección del software para el monitoreo de red.

| Software | Requerimiento | | | | Valoración total |
|----------|---------------|-------|-------|-------|------------------|
| | SYSH14 | SRS15 | SRS16 | SRS17 | |
| Nagios | 1 | 0 | 0 | 1 | 2 |
| Zabbix | 0 | 1 | 0 | 1 | 3 |
| PRTG | 1 | 1 | 1 | 1 | 4 |

Elección: se eligió el software PRTG ya que cumple con los requerimientos establecidos y además de ser software libre.

Fuente: elaborado por el autor

PRTG Network Monitor

Es una herramienta de monitoreo de redes fundamental para asegurar el funcionamiento de los sistemas informáticos y para evitar fallos en la red. Además de poseer un soporte para todas las versiones de SNMP y otros protocolos de monitoreo de red. Es una herramienta fácil de configurar y con una amplia gama de dispositivos soportados, además de poseer una interfaz web muy interactiva (Paessler AG, 2022). A continuación, unas de las principales características de PRTG:

- a) Administración remota a través de navegador web, dispositivos móviles, y aplicación de Windows
- b) Notificaciones sobre fallos a través de correo electrónico, mensajería instantánea, SMS, etc.
- c) Amplia selección de sensores
- d) Soporte de todos los métodos comunes de adquisición de datos de utilización (SNMP, sniffer de paquetes, NetFlow, sFlow, jFlow, IPFIX).

Para el diseño de software se ha tomado como referencia cada uno de los bloques de la arquitectura planteada, ya que para cada uno de los bloques existe un diseño de software involucrado que permitirá el correcto funcionamiento del sistema.

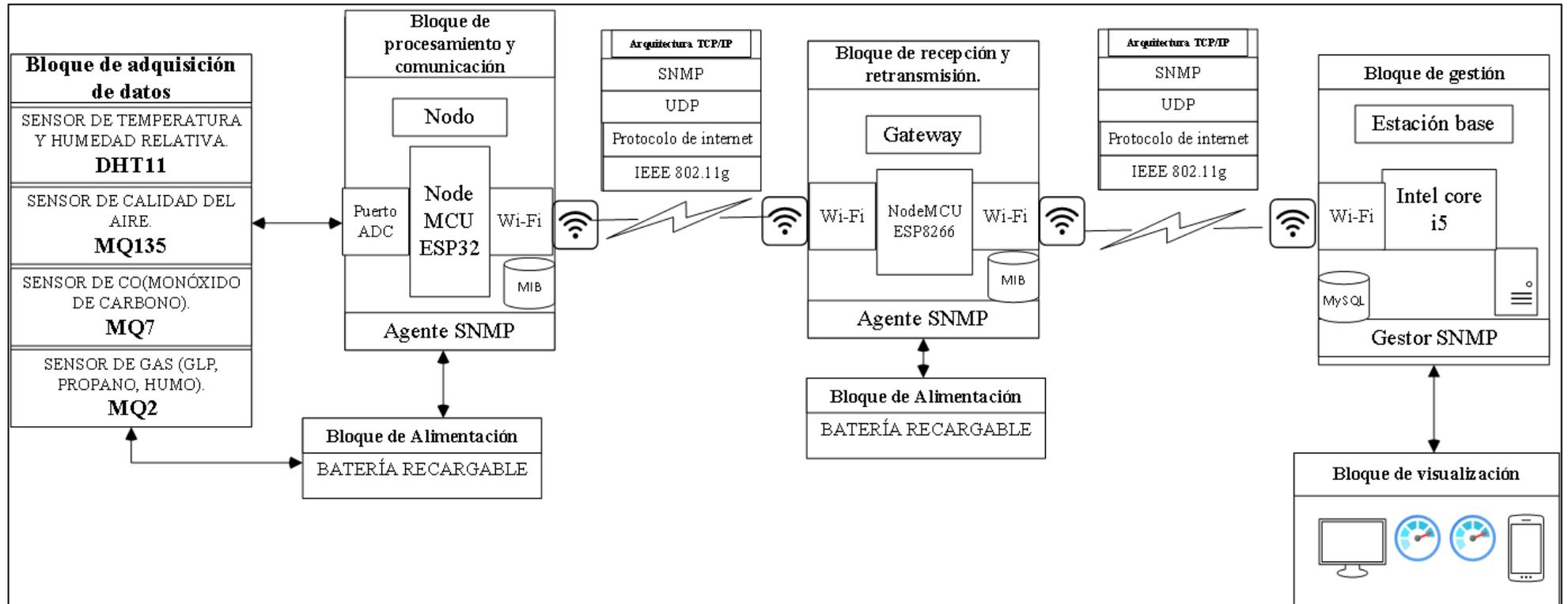
3.6. Definición De Arquitectura

De acuerdo con el estándar ISO /IEC/ IEEE 15288 se debe establecer y definir la arquitectura que dé a conocer los elementos son parte fundamental para el sistema electrónico de tal forma que sea entendible y muestre la distribución de todos los

elementos del sistema para cumplir con los requerimientos del sistema. En la Figura 27 se muestra la arquitectura establecida para el sistema de alerta y monitoreo. El cual se lo ha establecido en bloques para obtener una mejor distribución de cada uno de los procesos que se encargara cada elemento que compone la arquitectura.

Figura 27

Arquitectura general del sistema



Fuente: elaborado por el autor

La arquitectura se ha dividido en 6 bloques fundamentales para el funcionamiento del sistema, a continuación, se describe cada uno de los bloques:

3.6.1. Bloque de adquisición de datos

En este bloque se encuentra todos los sensores cuya función principal es la detección de gases tóxicos resultado del uso de agroquímicos para control de producción y de plagas en las flores, el uso de estos productos genera un residuo de gases tóxicos como son el amoniaco, alcohol, benceno, humo y dióxido de carbono (CO₂), monóxido de carbono (CO), óxido de nitrógeno (NO_x), propano, hidrógeno, metano, temperatura y humedad relativa ambiente.

3.6.2. Bloque de alimentación

La alimentación para los dispositivos como el Nodo Sensor y el Gateway, esta batería debe ser recargable, el cual debe ser adaptado a cada uno de los elementos para su funcionamiento. Para determinar el consumo de corriente que tendrá el dispositivo se debe realizar utilizando la *Ecuación 8* que se muestra a continuación (Espressif Systems, 2021):

$$I_T = \sum_{i=1}^n I_i + I_{i+1} \dots \dots \dots + I_n$$

Ecuación 8 Ecuación para determinar el consumo de corriente de dispositivos

Fuente: obtenido de Pablo A. (2016), *Electrónica aplicada* 2.^a edición

En la Tabla 27 se muestra cada elemento y su consumo de corriente, logrando determinar así el consumo total del dispositivo.

Tabla 27*Cálculo de consumo corriente*

| Elemento | Voltaje | Corriente |
|-----------------|----------------|------------------|
| Sensor QM-135 | 5 V | 150 mA |
| Sensor QM-2 | 5 V | 150 mA |
| Sensor QM-7 | 5 V | 150 mA |
| Sensor DTH11 | 5 V | 150 mA |
| NodeMCU-ESP32 | 5 V | 180 mA |
| TOTAL | 5 V | 780 mA |

Fuente: elaborado por el autor

Una vez calculado el consumo energético que tendrá el dispositivo se determina que se debe trabajar con una batería de 5 V y capacidad > 780 mA.

3.6.3. Bloque de procesamiento y comunicación

Este es el bloque encargado de analizar los datos obtenidos por los sensores, donde se procesa los datos digitales obtenidos a valores medibles ya sea ug/m³ o ppm, además en este bloque se analiza los datos mediante un algoritmo de clasificación supervisado K-NN el cual permite generar el índice de calidad de aire. En el nodeMCU se le ha agregado un agente SNMP el cual enviara al gestor SNMP el estado de los recursos del NodeMCU y sensores, todos estos datos se envían de manera inalámbrica formando una topología tipo estrella bajo el protocolo IEEE 802.11g.

3.6.4. Bloque de recepción y retransmisión

Este bloque se interconecta a todos los nodos sensores de la red WSN con la estación base, de tal manera que receptan las tramas enviadas por los nodos sensores y las encamina hacia la estación base o de las ordenes enviadas desde la estación base hacia los nodos sensores.

3.6.5. Bloque de gestión

En este bloque el administrador de la red puede realizar o enviar las órdenes a los nodos sensor y visualizar los datos obtenidos, además en este bloque contiene el gestor

SNMP el cumple la función de realizar las peticiones de los recursos a los agentes SNMP o a su vez recibir los TRAPS enviados por los agentes SNMP cuando ocurre un evento imprevisto.

3.6.6. Bloque de visualización

En este bloque se muestra al usuario los valores obtenidos mediante los nodos sensores de las concentraciones de gases en el aire mediante un indicie representado por un semáforo, además de ver individualmente los valores de cada sensor en una base de datos y poder observar el estado de los recursos de los nodos y Gateway.

3.7. Definición De Diseño

Una vez establecido la arquitectura se procede a realizar el diseño del prototipo donde se proporciona la suficiente información y datos detallados sobre el sistema y cada uno de los elementos para permitir la implementación de acuerdo con la arquitectura y los requerimientos establecidos.

3.7.1. Descripción General Del Funcionamiento Del Sistema Electrónico

El presente sistema de alerta y monitoreo consiste en el desarrollo de un sistema electrónico de detección de concentraciones de material particulado de los diferentes compuestos químicos utilizados en la floricultura, mediante el uso de sensores MQ que están compuestos por un sensor electroquímico que varía su resistencia al estar en contacto con las sustancias.

Inicialmente los sensores MQ y DTH realizan un escaneo del medio para la detección de la presencia de sustancias tóxicas en el aire, los datos obtenidos por los sensores se envían como valores digitales al microcontrolador por los puertos ADC, estos valores son procesados internamente para obtener valores en ppm, el microcontrolador NodeMCU esp32 tendrá alojado el algoritmo de clasificación supervisada K-NN y un

Agente SNMP, además el microcontrolador está equipado con un módulo de comunicación inalámbrica Wi-Fi bajo el protocolo 802.11g que es utilizado para enviar los datos, adicionalmente se le ha adaptado una batería recargable de litio de 3.7 voltios con una capacidad de 4000 mAh que ayudara a proporcionar la alimentación al sistema electrónico. El Gateway está formado por un NodeMCU esp8266, tiene alojado un Agente SNMP, también incorporado una batería recargable, la estación base está formada por un ordenador el cual servirá como estación base que tiene alojado al Gestor SNMP.

3.7.2. Diagrama De Flujo Del Proceso Del Sistema

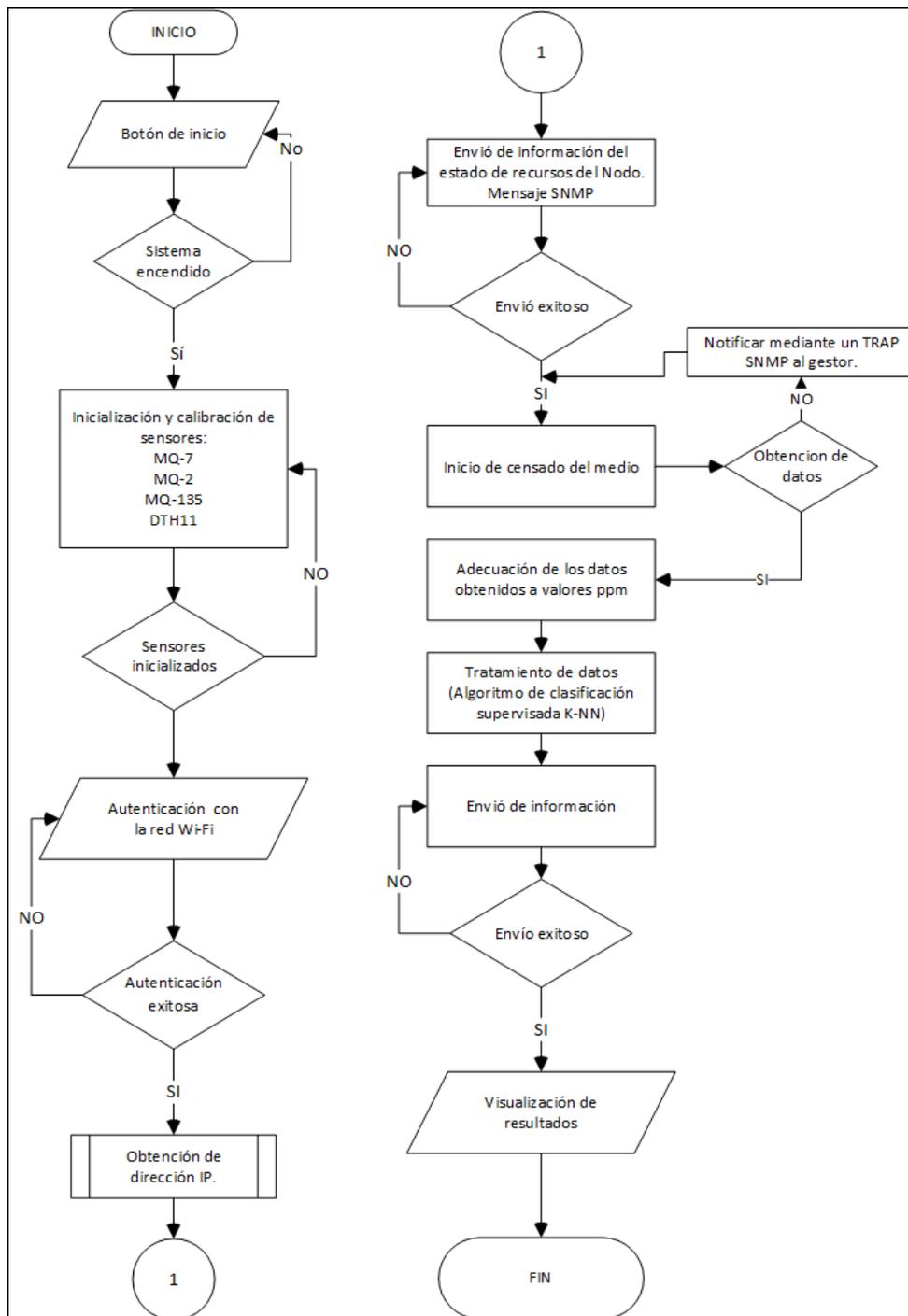
La Figura 28 es la representación gráfica del funcionamiento electrónico y acciones de cada elemento del sistema en determinados momentos mediante un diagrama de flujo. Se inicia con el encendido del sistema embebido mediante el uso de un interruptor entre la batería y el Microcontrolador que al activarse suplirá de energía al NodeMCU esp32 y a los sensores MQ-7, MQ-2, MQ-135, TDH11. Una vez encendidos los sensores se calibran y se inicializa las librerías de cada uno de los sensores y módulo de comunicación inalámbrico, una vez calibrados empieza el censado del ambiente y envío de datos obtenidos al microcontrolador donde se realiza la clasificación para lograr identificar el índice de contaminación.

Para la comunicación inalámbrica el dispositivo debe autenticarse y formar parte de la celda del Gateway, para poder enviar los datos. En el microcontrolador se encuentra aloja un Agente SNMP el cual recolecta la base de información MIB y envía estos datos que permiten al administrador de la red el estado de los recursos de cada sensor, si un sensor presenta valor de censado en cero, nivel de batería muy baja, tiempo operativo en cero se envía un TRAP a la estación base indicando el evento. El Gateway recibe la trama y encamina la trama hacia la estación base, el Gateway también aloja un Agente SNMP que recolecta la información del estado de los recursos del dispositivo y envía a la estación

base o Gestor. La estación base envía ordenes SNMP y recibe la información tanto de las concentraciones de sustancias toxicas como del estado de los recursos de cada dispositivo de la red WSN. Es aquí donde se puede observar los índices en una interfaz gráfica.

Figura 28

Diagrama de flujo del proceso del sistema



Fuente: elaborado por el autor

3.7.3. Diseño De Hardware

3.7.3.1. Bloque De Adquisición de datos

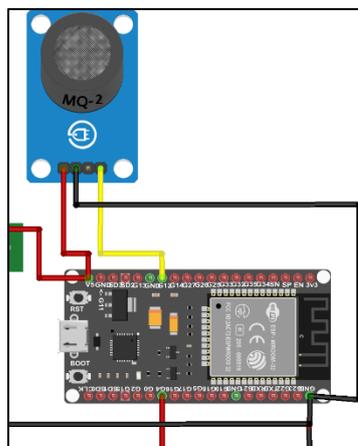
El bloque de adquisición de datos está conformado por cuatro sensores tres de los cuales son de la familia MQ y un sensor de humedad y temperatura. Los cuales permiten la detección de los diferentes gases.

3.7.3.1.1. Diagrama de conexión del sensor MQ-2

Este sensor se alimenta mediante una batería recargable por el puerto VCC a 5V del sensor MQ-2 y del Node MCU esp32, el pin GND al pin GND del Node MCU esp32 y el pin analógico se conecta al pin ADC 12 de Node MCU esp32. En la Figura 29 se muestra la distribución y conexión.

Figura 29

Distribución y conexión de pines del sensor MQ-2



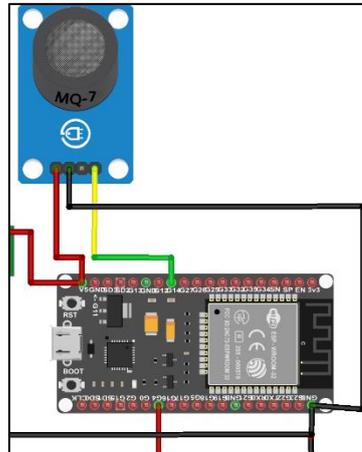
Fuente: elaborado por el autor

3.7.3.1.2. Diagrama De Conexión Del Sensor MQ-7

Este sensor se alimenta mediante una batería recargable por el puerto VCC a 5V del sensor MQ-7 y del Node MCU esp32, el pin GND al pin GND del Node MCU esp32 y el pin analógico se conecta al pin ADC 14 de Node MCU esp32. En la Figura 30 se muestra la distribución y conexión.

Figura 30

Distribución y conexión del sensor MQ-7



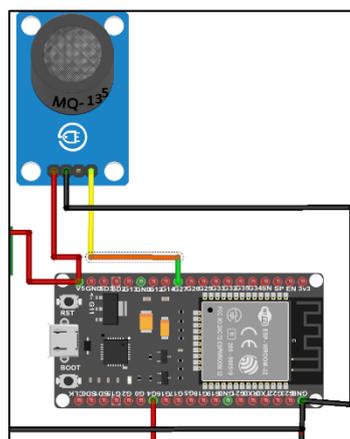
Fuente: elaborado por el autor

3.7.3.1.3. Diagrama De Conexión Del Sensor MQ-135

Este sensor se alimenta mediante una batería recargable por el puerto VCC a 5V del sensor MQ-135 y del Node MCU esp32, el pin GND al pin GND del Node MCU esp32 y el pin analógico se conecta al pin ADC 27 de Node MCU esp32. En la Figura 31 se muestra la distribución y conexión.

Figura 31

Distribución y conexión del sensor MQ-135



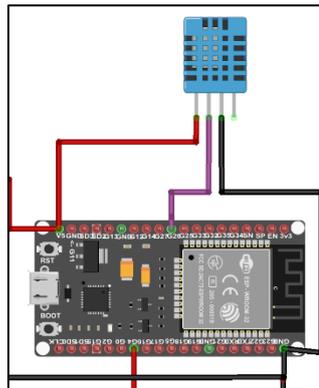
Fuente: elaborado por el autor

3.7.3.1.4. Diagrama de conexión del sensor DTH11

Este sensor se alimenta mediante una batería recargable por el puerto VCC a 5V del sensor DTH11 y del Node MCU esp32, el pin GND al pin GND del Node MCU esp32 y el pin analógico se conecta al pin ADC 28 de Node MCU esp32. En la Figura 32 se muestra la distribución y conexión.

Figura 32

Distribución y conexión del sensor DTH11



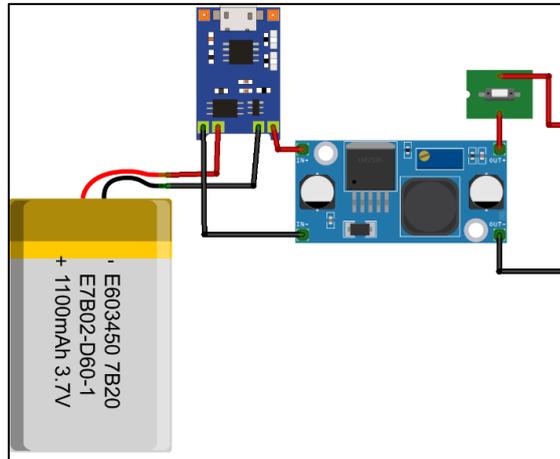
Fuente: elaborado por el autor

3.7.3.2. Bloque De Alimentación

Para proporcionar la alimentación a todo el sistema electrónico se utiliza una batería recargable, un módulo proporcione carga a la batería y un módulo que convierta el voltaje de 3.7v a 5v, ya que todos los sensores trabajan a 5v, por lo cual se utiliza una batería de iones de litio 18650 de 4000 mAh a 3.7V, un módulo cargador TP4056 y el módulo convertidor elevador DC-DC MT3608. La batería se conecta al módulo de carga y al convertidor elevador de voltaje el cual va a entregar 5 voltios, el convertidor elevador de voltaje se conecta a la entrada Vin del Node MCU ESP32 y a la línea que alimentara a todos los sensores, en la Figura 33 se muestra la distribución y conexión de los módulos y las líneas de alimentación.

Figura 33

Diagrama de distribución y conexión para la alimentación del circuito



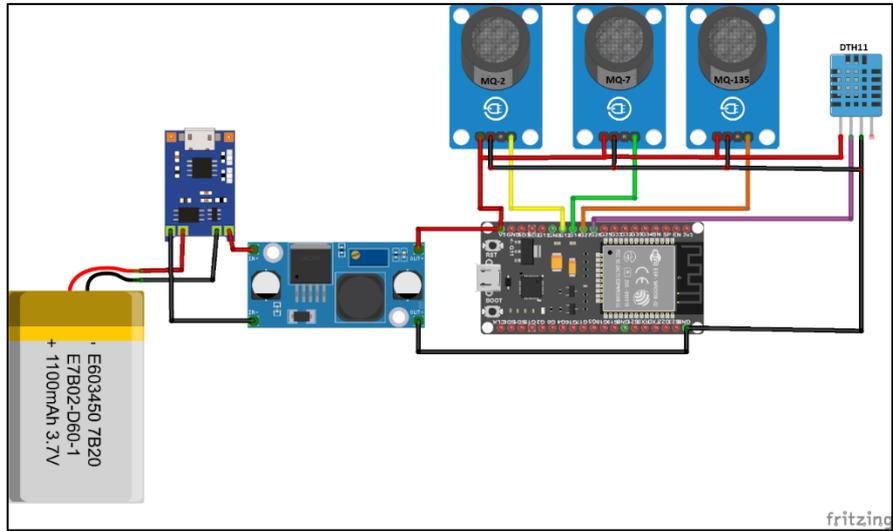
Fuente: elaborado por el autor

3.7.3.3. Bloque de procesamiento y comunicación

El bloque de procesamiento y comunicación está constituido por un NodeMCU esp32 el cual es una placa de desarrollo que cuenta con un procesador Tensilica Xtensa LX6 de 32 bits el cual permite la implementación de sistemas electrónicos que requieran adquirir datos y procesarlos internamente antes de ser enviados, además cuenta con un módulo de comunicación inalámbrica bajo el protocolo IEEE 802.11 para el envío de datos, esta alimentado por una batería recargable de 3.7 V y una capacidad de 4000 mAh. El NodeMCU se conectan los sensores MQ-2, MQ-7. MQ-135 Y DTH11 a los pines ADC para la toma de datos y a los pines VCC 5V Y GND para la alimentación como se muestra en la Figura 34.

Figura 34

Diseño Circuital del nodo sensor



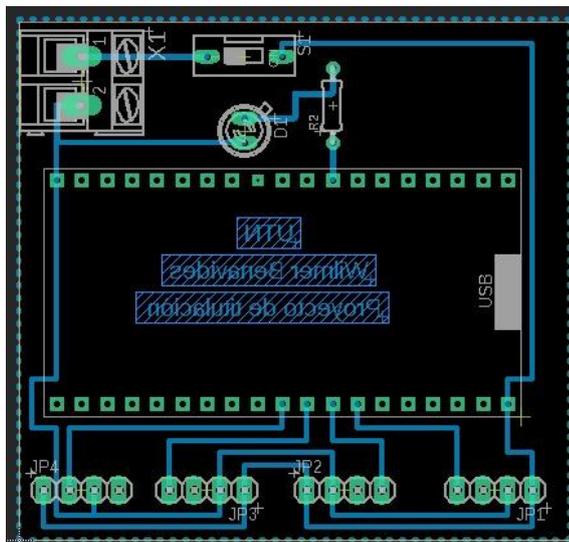
Fuente: elaborado por el autor

3.7.3.3.1. *Diseño De La Placa Del Circuito Impreso (PCB) Para Nodo Sensor*

Para el diseño de la placa electrónica se utilizó el software Eagle, donde se puede diseñar placas PCB y cuenta con librerías de cada uno de los elementos que se va a utilizar con sus dimensiones y distribuciones de pines, en la Figura 35 se muestra el diseño implementado.

Figura 35

Diseño de placa electrónica



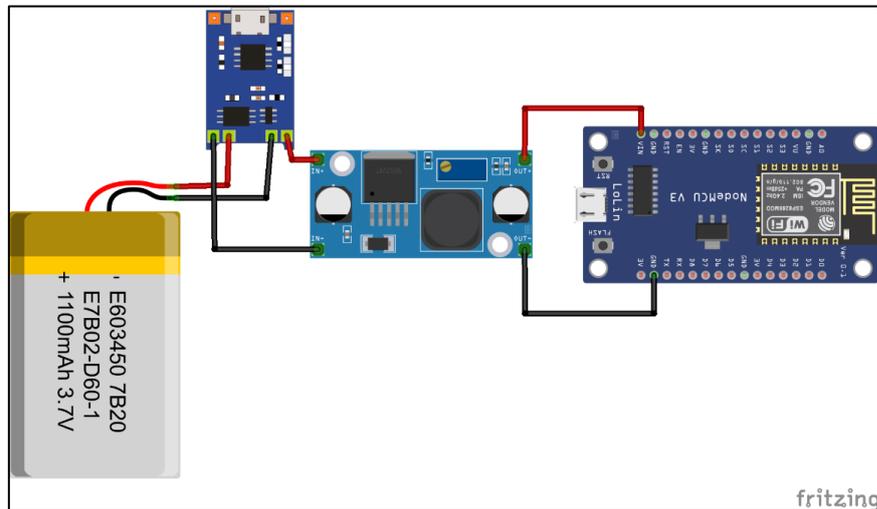
Fuente: elaborado por el autor

3.7.3.4. Bloque De Recepción Y Retransmisión

Para el diseño del Gateway se ha establecido el uso de un NodeMCU esp8266 ya que cumplía los requerimientos establecidos, posee un módulo de comunicación bajo el protocolo IEEE 802.11 para la comunicación, que será el medio por el cual se comunicaran la estación base con los nodos sensores, es alimentada por el bloque de alimentación que proporciona los 5 voltios que se necesita, en la Figura 36 se muestra la distribución y conexión.

Figura 36

Diagrama Circuitual de Gateway



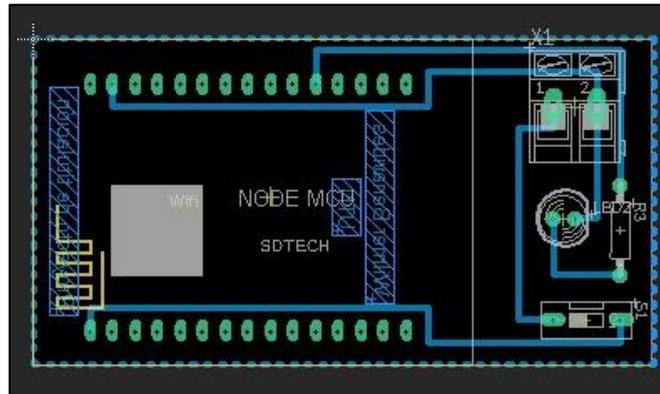
Fuente: elaborado por el autor

3.7.3.4.1. Diseño De La Placa Del Circuito Impreso (PCB) Para Gateway

Para el diseño de la placa electrónica se utilizó el software Eagle, donde se puede diseñar placas PCB y cuenta con librerías de cada uno de los elementos que se va a utilizar con sus dimensiones y distribuciones de pines, en la Figura 37 se muestra el diseño implementado.

Figura 37

Diseño de placa electrónica para Gateway



Fuente: elaborado por el autor

3.7.4. Diseño De Software

3.7.4.1. Bloque De Adquisición De Datos

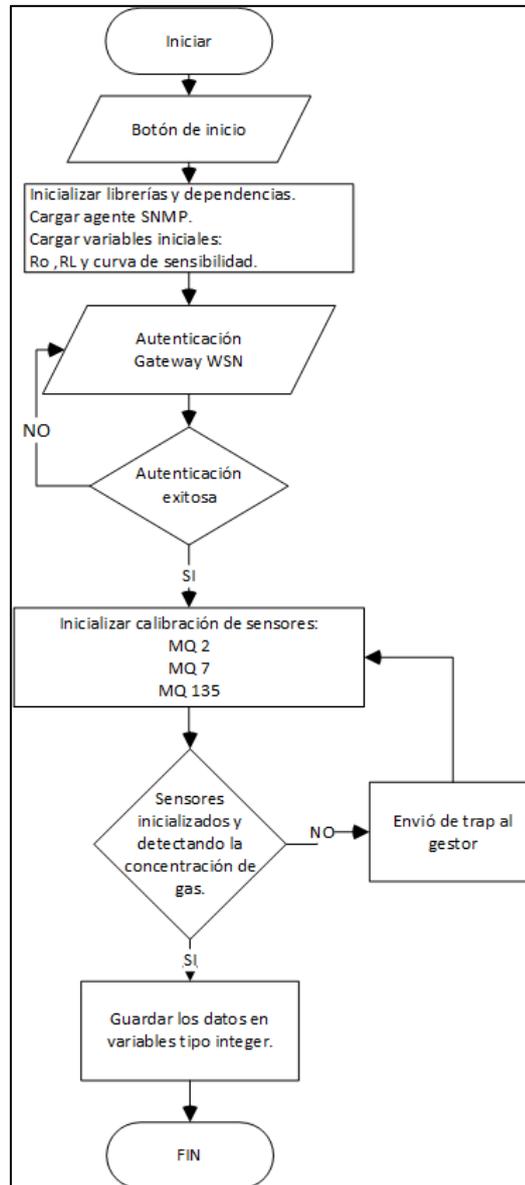
Para el bloque de adquisición de datos que consta de cuatro sensores entre ellos MQ-2, MQ-7, MQ-135 los cuales requieren una calibración necesaria para la correcta detección de las sustancias toxicas en el aire, debido a esto la calibración se la realiza asignando la orden mediante programación y se ejecuta en el NodeMCU esp32, ya que al encender el dispositivo la primera acción es calibración de los sensores para posteriormente censar el medio.

3.7.4.1.1. Diagrama De Flujo Del Bloque De Adquisición De Datos

El diagrama de flujo plasma el ciclo de funcionamiento que se debe seguir para un correcto funcionamiento. Donde el primer paso es verificar que los sensores estén encendidos, iniciar las dependencias, cargar el agente SNMP y realizar la calibración e impresión de datos, en la Figura 38 se muestra el diagrama de flujo para el bloque de adquisición de datos.

Figura 38

Diagrama de flujo de bloque de sensores



Fuente: elaborado por el autor

3.7.4.1.2. Calibración De Sensores

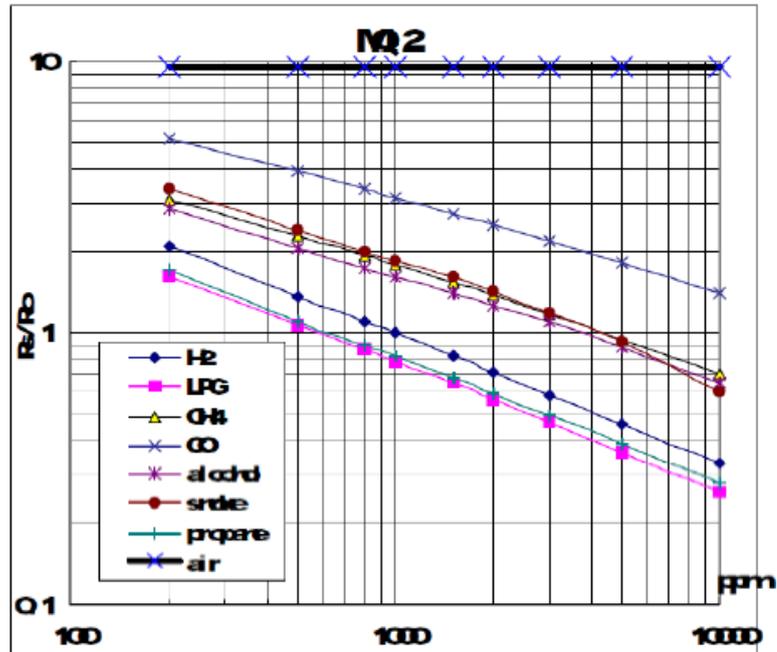
Sensor MQ-2

El sensor MQ-2 está compuesto por un material semiconductor de dióxido de estaño (SnO₂) y dependiendo de la concentración de gas aumenta su conductividad, este sensor es utilizado para la detección de GLP, i-butano, propano, metano, alcohol,

hidrógeno y humo. En la Figura 39 se muestra la curva de sensibilidad a los diferentes gases del sensor MQ-2.

Figura 39

Características de sensibilidad del sensor MQ-2



Fuente: adaptado de Hanwei Eletronics co.,ltd (2015) Technical Data MQ-2 Sensor

A continuación, se muestra las típicas características de sensibilidad del MQ-2 durante varios gases en su:

- Temperatura: 20 °C,
- Humedad: 65%,
- Concentración de O₂ 21%
- $R_L = 5k\Omega$
- R_0 : resistencia del sensor a 1000 ppm de H₂ en el aire limpio.
- R_s : resistencia del sensor en varias concentraciones de gases.

R_s es la resistencia que varía según las diferentes cantidades de gases detectados por lo cual es la variable para tomar en cuenta, Para el cálculo de la resistencia superficial R_s se utiliza la *Ecuación 9*.

$$R_s = \left(\frac{V_c * R_L}{V_{out}} \right) - R_L$$

Ecuación 9 Ecuación para el cálculo de la resistencia superficial del sensor

Fuente: adaptado de Hanwei Eletronics co.,ltd (2015) Technical Data MQ-2 Sensor

Donde, V_c el voltaje de entrada 5V, V_{out} el voltaje de salida y R_L es la resistencia Variable que se especifica en el datasheet dependiendo del sensor.

Una vez obtenido los valores R_s y R_o se procede a generar una ecuación que permita interpretar la (Figura 39) al no ser una función lineal ya que posee deformaciones y más separaciones en algunos ejes, por lo que sigue una tendencia a ser una función logarítmica. Por lo cual para llegar a dicha función se parte suponiendo que es una función lineal en la *Ecuación 13* se establece la ecuación utilizada para ecuaciones lineales.

$$y = mx + b$$

Donde, x = eje de las abscisas, y = eje de las ordenadas y m = pendiente de la curva.

Al ser una función logarítmica y para no afectar a la ecuacion se plica logaritmo a ambas partes.

$$\log(y) = m * \log(x) + b$$

Se procede a calcular la pendiente partiendo de dos puntos(x_1,y_1) y (x_2,y_2) conocidos de la gráfica.

$$m = \frac{\log(y)}{\log(x)}$$

$$m = \frac{\log(y_2) - \log(y_1)}{\log(x_2) - \log(x_1)}$$

$$m = \frac{\log(y_2/y_1)}{\log(x_2/x_1)}$$

$$x = \log(ppm), y = \log\left(\frac{RS}{RO}\right), m = \frac{\log(y_2/y_1)}{\log(x_2/x_1)}$$

Con estos valores ya se puede reemplazar y llegar a los valores en PPM del gas a encontrar.

$$\log\left(\frac{RS}{RO}\right) = m * \log(PPM) + b$$

$$PPM = 10^{\left\{\frac{\log\left(\frac{RS}{RO}\right) - b}{m}\right\}}$$

Ecuación 10 Ecuación para obtener los parámetros de sensibilidad del gas en ppm

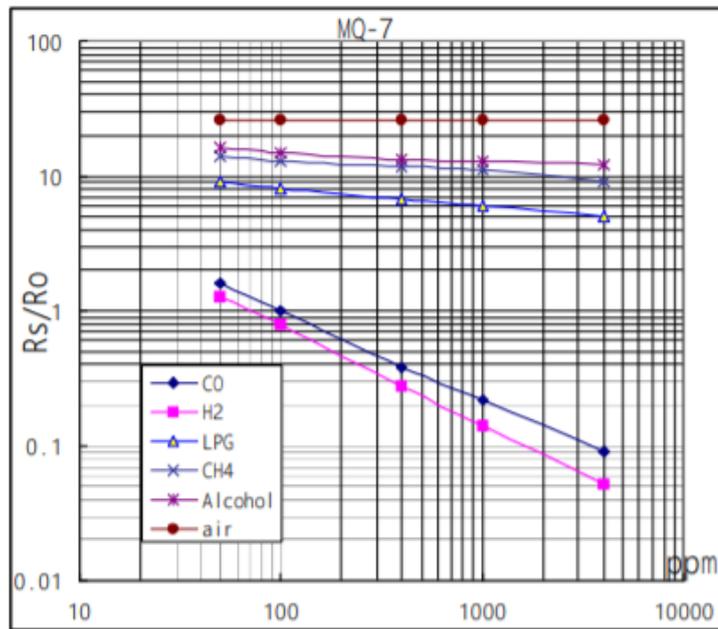
Fuente: adaptado de Lopez, S. R. (2018). Aplicación de sistema de control y mantenimiento del nivel de CO2 en espacios cerrados.

Donde (x1,y1) un punto sobre la curva y (x2,y2) otro más distantes, estos valores se obtienen de la curva de sensibilidad del sensor especificados en la Figura 43. Y dependiendo la curva de cada sensor se utiliza la misma ecuación.

Sensor MQ-7

El sensor MQ-7 es usado para la detección de monóxido de carbono (CO), como se muestra en la Figura 40 la curva de sensibilidad dependiendo del tipo de gas.

Figura 40 Características de sensibilidad del sensor MQ-7



Fuente: adaptado de Hanwei Eletronics co.,ltd (2015) Technical Data MQ-7 Sensor

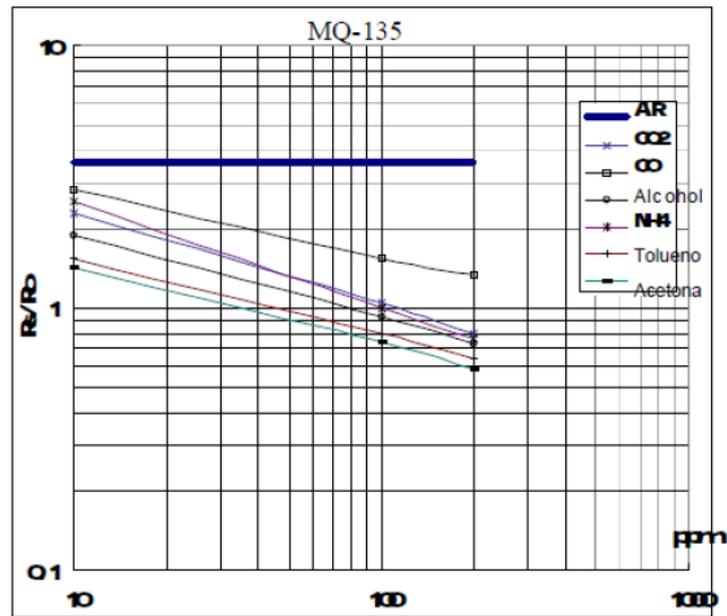
A continuación, se muestra las típicas características de sensibilidad del MQ-7 durante varios gases en su:

- T_a : 20°C; Humedad: 65%
- Concentración de O₂: 21%
- $R_L = 10K\Omega$
- R_0 = Resistencia del sensor a 100 ppm de CO en el aire limpio
- R_s = Resistencia del sensor a varias concentraciones de gases

Sensor MQ-135

El sensor MQ-135 ampliamente usado en dispositivos de control de calidad del aire y son adecuados para detectar NH₃, NO_x, alcohol, benceno, humo, CO₂. En la Figura 41 se muestran la curva de sensibilidad para cada gas.

Figura 41 Características de sensibilidad del sensor MQ-135



Fuente: adaptado de Hanwei Eletronics co.,ltd (2015) Technical Data MQ-135 Sensor

A continuación, se muestra las típicas características de sensibilidad del MQ-135 durante varios gases en su:

- Temperatura: 20°C
- Humedad: 65% $RL:20k\Omega$
- R_0 : Resistencia del sensor a 100ppm de NH_3 en el aire libre.
- R_s : Resistencia del sensor a varias concentraciones de gases

Empleando la *Ecuación 9* antes mencionada se procede a generar el código de programación que al momento de encender el dispositivo y antes de empezar a censar el medio el sensor MQ-2 este calibrado para una correcta toma de datos.

3.7.4.2. Bloque De Procesamiento y Comunicación

Para el procesamiento de los datos obtenidos por el bloque de adquisición de datos se seguirá el proceso de KDD (Knowledge Discovery in Databases), en el cual indica los pasos a seguir en la adquisición, limpieza, minería e interpretación de grandes volúmenes

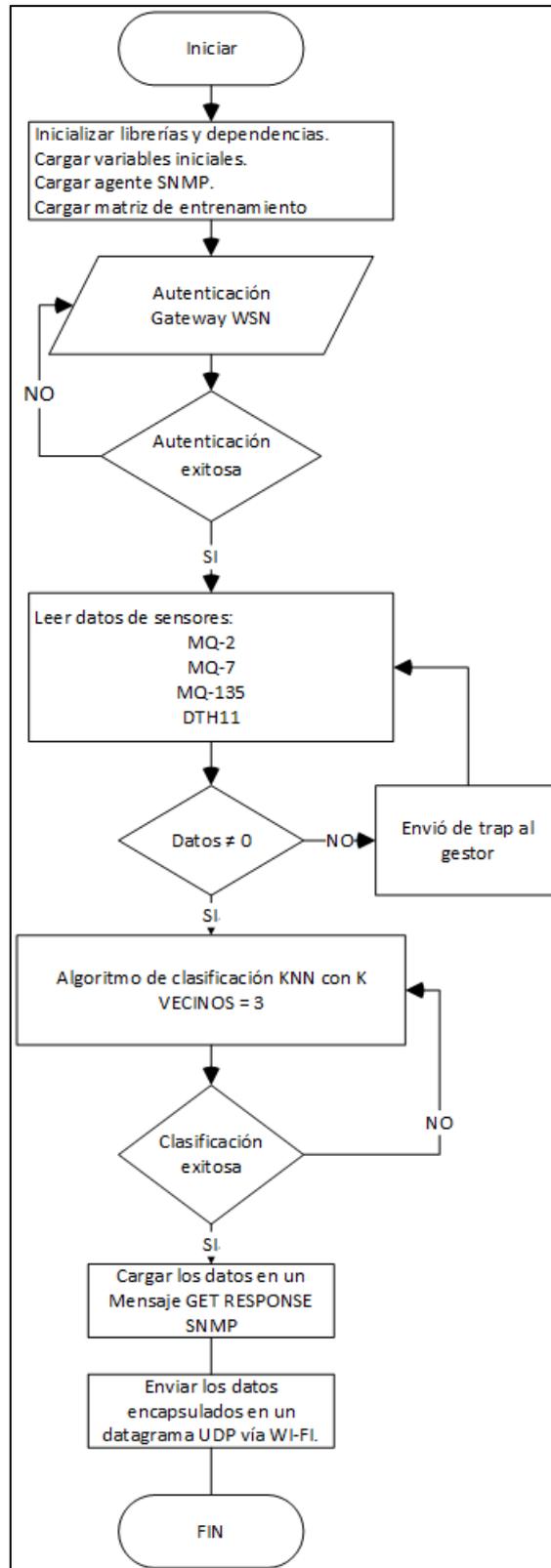
de datos. Mediante un algoritmo que se programara e implementara en cada nodo sensor, donde estos valores se ingresaran en un mensaje SNMP, para después ser enviada por el medio inalámbrica utilizando el protocolo IEEE 802.11g.

3.7.4.2.1. Diagrama De Flujo De Bloque De Procesamiento Y Comunicación

El diagrama de flujo propuesto para el bloque de procesamiento y comunicación muestra el ciclo de funcionamiento para realizar el procesamiento de los datos mediante el algoritmo de clasificación supervisada KNN y posteriormente se realiza se envían mediante el uso de módulo de comunicación inalámbrica utilizando el protocolo SNMP para el encapsulamiento de datos en los datagramas UDP como se muestra en la Figura 42.

Figura 42

Diagrama de flujo de bloque de procesamiento y comunicación



Fuente: elaborado por el autor

3.7.4.3. Bloque De Recepción y Retransmisión

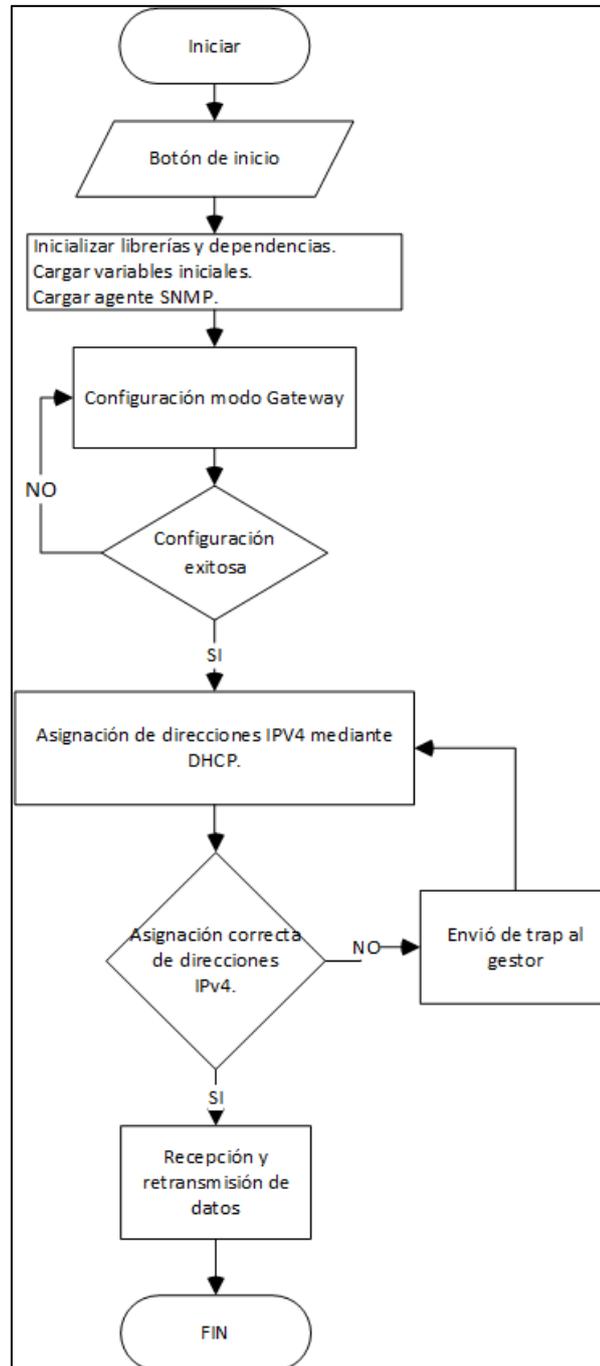
Este bloque se encarga de proporcionar un medio para que todos los dispositivos de la red WSN se comuniquen entre sí, se ha establecido el NodeMCU esp8266 para proporcionar la función de Gateway ya que cumple con los requerimientos de hardware requeridos, en este Gateway se ha realizado la configuración tal que logre proveer la conexión entre la estación base y los nodos de la red, en las configuraciones se ha establecido como “SSID= GATEWAR_WSN” y una contraseña para la autenticación de los dispositivos que conformaran la red WSN, además se le ha añadido un agente SNMP para el monitoreo.

3.7.4.3.1. Diagrama De Flujo De Funcionamiento

El diagrama de flujo propuesto para el bloque de recepción y retransmisión de datos, como puede observar en la Figura 43 se muestra el ciclo de funcionamiento para realizar el encaminamiento de los datos que son enviados vía inalámbrica por los nodos sensores hacia la estación base, iniciando con cargar las librerías necesarias para la asignación de direcciones ipv4 y para el encaminamiento de segmentos tanto para TCP, UDP e ICMP, también se carga el agente SNMP para el monitoreo del dispositivo, posteriormente se realiza la asignación de dirección ip usando DHCP a cada nodo sensor que se formen parte de la celda y se realiza la recepción y retransmisión de los datos que son enviados desde nodos a la estación base y de la estación base a los nodos sensores.

Figura 43

Diagrama de flujo para el bloque de recepción y retransmisión de datos



Fuente: elaborado por el autor

3.7.4.4. Bloque De Gestión

3.7.4.4.1. Descripción General Del Modelo De Gestión De Red

El bloque de gestión es el encargado de proporcionar un modelo de gestión de red capaz de implementarse en dispositivos que forman una WSN, por lo cual se ha establecido un modelo de gestión que trabaje con los tres componentes de la red WSN Nodo sensor, Gateway y estación base, bajo el paradigma de agente-gestor. Debido a esto se ha implementado el modelo de gestión de red internet SNMP, que cumple con todos los requerimientos establecidos como menor uso de recursos de procesamiento y al ser uno de los protocolos más implementados en la actualidad posee un amplio campo de implementación y soporte.

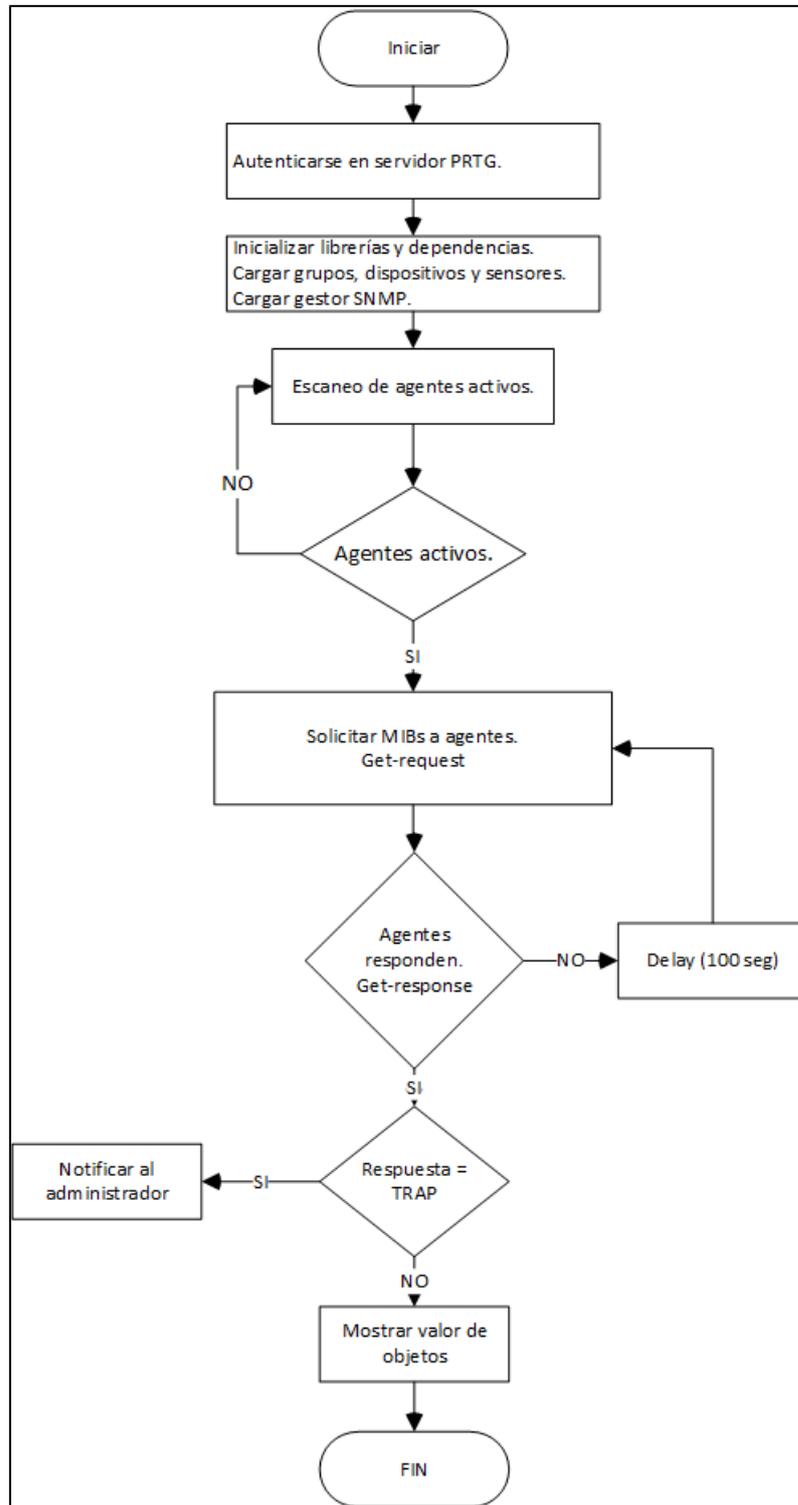
En cuanto a la versión de protocolo SNMP se ha optado por la versión SNMPv2c por sus funcionalidades que permiten la implementación en casi cualquier dispositivo, además por poseer SNMP v2 TRAP lo cual es una tarea muy importante para el monitoreo de una red ya que proporciona una notificación instantánea al ocurrir un evento sin necesidad de una petición del gestor.

3.7.4.4.1. Diagrama de flujo del funcionamiento

El diagrama de flujo muestra el ciclo de funcionamiento del gestor SNMP, como se muestra en la Figura 44 se inicia con la autenticación con el servidor PRTG NETWORK MONITOR para así acceder a todas las herramientas del software, posteriormente el gestor empieza a solicitar a los agentes que se pronuncien mediante un escaneo para verificar que agentes están activos en caso de no responder espera un tiempo y vuelve a solicitar, con lo agentes activos el gestor realiza una petición de la MIB mediante un mensaje GET-REQUEST, al cual los agentes responderán, si la respuesta es un TRAP notifica al administrador, caso contrario muestra los valores de cada objeto (cada sensor) en la ventana de visualización.

Figura 44

Diagrama de flujo del funcionamiento del gestor SNMP



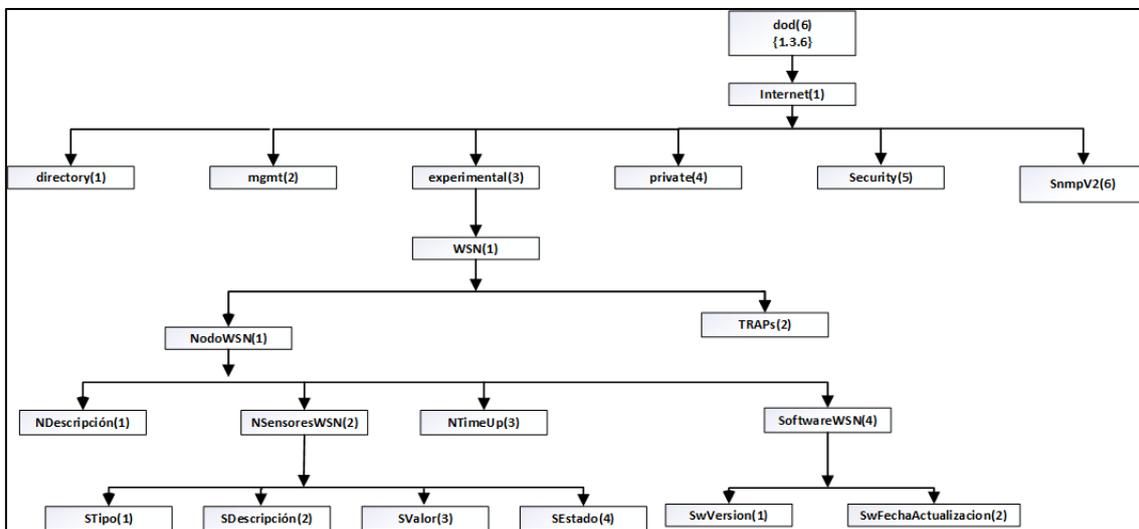
Fuente: elaborado por el autor

3.7.4.4.2. Base de datos de información

El modelo de gestión de red Internet especifica una estructura de información, en el cual se utiliza OID y las bases de información de gestión MIBs. Por lo cual el modelo de gestión de red para la WSN debe ser capaz de identificar las MIBs de tipo estándar (MIB-II definida por el RFC 1213), no obstante, no todas las MIBs son aplicables con una red WSN, por lo cual es necesario la incorporación de una MIB de tipo privada, ya que la red WSN maneja sensores y microcontroladores, objetos que no se encuentran disponibles en la MIB-II, por lo cual la MIB privada se encargara de medir en tiempo operativo del Nodo, Nombre de sensor, versión y valor tomado por cada sensor. Con esta iniciativa se establece la estructura de base de la información MIBs para una WSN como se muestra en la Figura 45.

Figura 45

Estructura MIB privada para el modelo de gestión de red WSN



Fuente: elaborado por el autor

En la Tabla 28 se muestra la descripción de los objetos del subárbol de sensores WSN el cual está basado en el modelo de MIB-II.

Tabla 28

Objetos de subárbol de NodoWSN

| OBJETO | DESCRIPCIÓN | SINTAXIS | PERMISO | OID |
|---------------------|---|---------------------|----------------|-------------------|
| NDescripción | Detalles característicos del Nodo sensor | DisplayString | Read-only | 1.3.6.1.3.1.1.1 |
| NSensoresWSN | Lista de sensores presentes en el Nodo | SEQUENCE of NodoWSN | Not-accessible | 1.3.6.1.3.1.1.2 |
| NTimeUP | Tiempo encendido | TIMETICKS | Read-only | 1.3.6.1.3.1.1.3 |
| SoftwareWSN | Detalles de Versión del software instalado | SEQUENCE of NodoWSN | Not-accessible | 1.3.6.1.3.1.1.4 |
| STipo | Tipo de sensor | DisplayString | Read-only | 1.3.6.1.3.1.1.2.1 |
| SDescripcion | Detalles característicos del sensor | DisplayString | Read-only | 1.3.6.1.3.1.1.2.2 |
| SValor | Valor en ppm de sustancia captada por el sensor | INTEGER | Read-only | 1.3.6.1.3.1.1.2.3 |
| SEstado | Sensor encendido o apagado | STRING { UP, DOWN } | Read-only | 1.3.6.1.3.1.1.2.4 |
| SwVersion | Versión del software instalada | DisplayString | Read-only | 1.3.6.1.3.1.1.4.1 |
| SwFechaAct | Fecha de la última actualización realizada | DisplayString | Read-only | 1.3.6.1.3.1.1.4.2 |

Fuente: elaborado por el autor

Establecido los objetos que se utilizaran para la base de información de gestión se procede a especificarlos mediante el uso de código de programación, donde se especifica cada uno de los objetos que se va a monitorizar del nodo sensor.

De igual manera en la Tabla 29 se muestra la descripción para el objeto TRAPs donde se especifica cada una de las notificaciones posibles.

Tabla 29

Traps

| OBJETO | DESCRIPCIÓN | OID |
|-----------------------|---|-----------------|
| NODO REBOOTING | El Nodo se está reiniciando constantemente. | 1.3.6.1.3.1.2.1 |

| | | |
|-----------------------|--|-----------------|
| SENSOR NO READ | El sensor no está detectando ningún valor. | 1.3.6.1.3.1.2.2 |
| SENSOR DOWN | El sensor se encuentra apagado. | 1.3.6.1.3.1.2.3 |

Fuente: elaborado por el autor

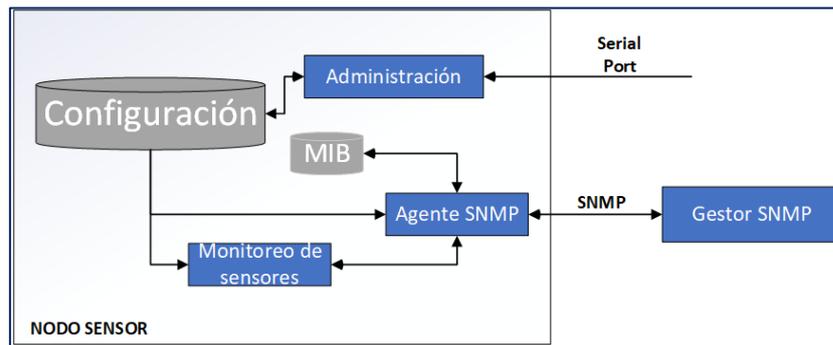
Establecido los objetos que se utilizaran para los traps se procede a especificarlos mediante el uso de código de programación, donde se especifica cada uno de los OIDs correspondientes a los traps.

3.7.4.4.3. *Agente SNMP*

Tomando en cuenta los objetivos de la gestión de red siendo este caso en dispositivos IoT, se ha establecido los siguientes módulos que se han desarrollado para el agente de los dispositivos, en la Figura 46 se muestra la distribución de los módulos que conforman el agente SNMP.

Figura 46

Componentes de agente SNMP para los nodos SNMP



Fuente: elaborado por el autor

El agente SNMP implementa las funcionalidades del protocolo SNMP correspondiente a la versión 2c, a continuación, se describe cada uno de los módulos del agente:

Agente SNMP

Recepción de mensajes SNMP enviados por el gestor: estos mensajes corresponden a la lectura (GetRequest, GetNextRequest) y escritura (SetRequest) de valores de los objetos de información.

Envío de mensajes SNMP al gestor: estos mensajes corresponden a la respuesta (GetResponse) de las peticiones del gestor, y las notificaciones o mensajes espontáneos TRAPs enviados cuando ocurre un evento inesperado en el dispositivo.

Lectura y escritura de los MIBs: correspondiente a cada uno de los objetos especificados para el NodoSensor.

Monitoreo de sensores

Este módulo de encarga de monitorizar cada uno de los sensores implementados en el dispositivo, es decir, si están activos o desactivados mediante una lectura periódica ya que los sensores deben estar constantemente censando el medio y obteniendo datos, si un sensor no presenta valores censados se activa un TRAP que notificara al gestor a través del agente SNMP.

Administración

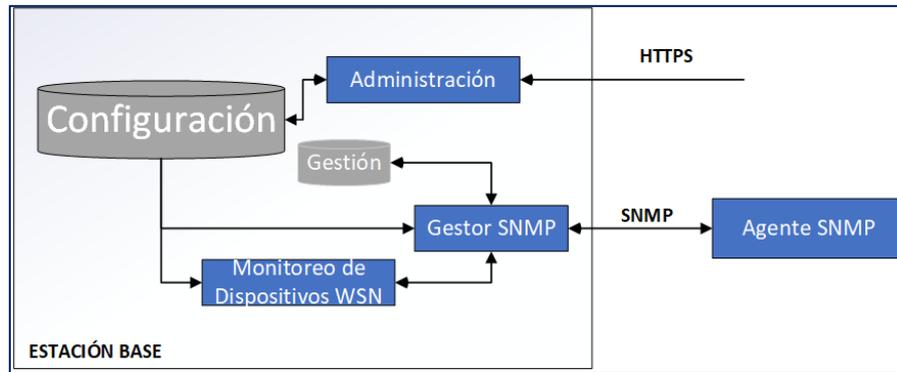
Mediante este módulo se realiza las configuraciones vía puerto serial, donde se puede acceder a al código fuente que posee el dispositivo y realizar las diferentes correcciones cuando presente algún error de software el dispositivo.

3.7.4.4. Gestor SNMP

Tomando en cuenta las funcionalidades de un gestor SNMP, se ha establecido los siguientes módulos que proporcionan las funcionalidades para la gestión de la red WSN en la Figura 47 se puede observar la distribución de dichos módulos.

Figura 47

Componentes de gestor SNMP



Fuente: elaborado por el autor

A continuación, se describe las funcionalidades de cada uno de los módulos que implementa las funcionalidades del protocolo SNMP correspondiente a la versión 2c:

Gestor SNMP

En gestor SNMP es el encargado de enviar al agente las diferentes peticiones para consultar la información de gestión, el estado de los recursos de los recursos del dispositivo, estos mensajes corresponden a:

- GetRequest: petición de valores específicos de la MIB.
- GetNextRequest: petición del objeto siguiente a uno dado de la MIB.
- GetBulkRequest: petición de múltiples valores.

Recepción de los mensajes enviados por el agente SNMP como respuesta a las peticiones del agente, Traps que se envían sin solicitud del gestor.

Almacenamiento de la información obtenido de los agentes SNMP en una base de datos de gestión.

Monitoreo de dispositivos WSN

Este módulo se encarga de monitoreo del estado de cada uno de los dispositivos de la red WSN, verificado el estado es decir si se encuentra activo, si responde a las ordenes enviadas por parte del gestor.

Administración

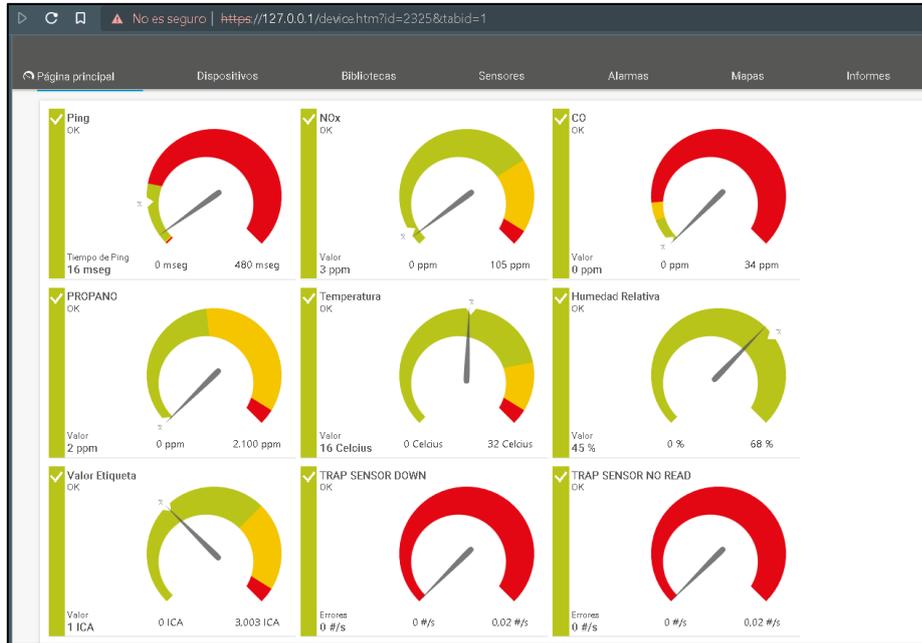
Mediante este módulo se realiza la configuración para realizar modificaciones en los diferentes aspectos que corresponder a la gestión de la red como ingreso de nuevos dispositivos mediante la dirección ip, agregar nuevos OIDs, revisar los estados de cada uno de los dispositivos.

3.7.4.5. Bloque De Visualización

Para la visualización de los objetos gestionados por el gestor, se ha establecido la implementación el software de monitoreo de red PRTG NETWORK MONITOR el cual permite ver en tiempo real la información recolectada por el agente SNMP que se a integrado en cada NODO y una gestión completa que incluye: ingreso de OID para cadenas de texto, valores enteros y time, recepción de traps y notificación vía correo o sms. En la Figura 48 se puede observar como se muestra los valores obtenidos mediante unos indicadores tipo velocímetro para los valores de tipo entero (Integer).

Figura 48

Indicador de objetos



Fuente: elaborado por el autor

Y para los objetos de tipo cadena de texto (String) se muestra en una tabla como se puede ver en la Figura 49.

Figura 49

Valores de objetos tipo String.

| | | | | | | | |
|------|--------------------------------|----|-------------------|----------------|------------|-------|---|
| +1. | ✓ Ping | OK | OK | Tiempo de Pir | 17 mseg | ★★★★★ | □ |
| +2. | ✓ Tiempo de actividad | OK | OK | Disponibilidad | 10 m 33s | ★★★★☆ | □ |
| +3. | ✓ SysContact | OK | WILMER BENAVIDES | Tiempo de res | 188 mseg | ★★★★☆ | □ |
| +4. | ✓ SysDescr | OK | SNMP Agent NODO 1 | Tiempo de res | 153 mseg | ★★★★☆ | □ |
| +5. | ✓ SysName | OK | NodeMCUEsp82 | Tiempo de res | 229 mseg | ★★★★☆ | □ |
| +6. | ✓ SysLocation | OK | Invernadero # 1 | Tiempo de res | 864 mseg | ★★★★☆ | □ |
| +7. | ✓ NOx | OK | OK | Valor | 4 ppm | ★★★★★ | □ |
| +8. | ✓ CO | OK | OK | Valor | 0 ppm | ★★★★★ | □ |
| +9. | ✓ PROPANO | OK | OK | Valor | 5 ppm | ★★★★★ | □ |
| +10. | ✓ Temperatura | OK | OK | Valor | 16 Celcius | ★★★★★ | □ |
| +11. | ✓ Humedad Relativa | OK | OK | Valor | 40% | ★★★★★ | □ |
| +12. | ✓ Valor Etiqueta | OK | OK | Valor | 1 ICA | ★★★★★ | □ |
| +13. | ✓ TRAP SENSOR DOWN | OK | OK | Errores | 0 #/s | ★★★★★ | □ |
| +14. | ✓ TRAP SENSOR NO READ | OK | OK | Errores | 0 #/s | ★★★★★ | □ |
| +15. | ✓ TRAP NODO REBOOTING | OK | OK | Errores | 0 #/s | ★★★★★ | □ |
| +16. | ✓ StatusSensorMQ135 | OK | OK | Tiempo de res | 195 mseg | ★★★★☆ | □ |
| +17. | ✓ StatusSensorMQ7 | OK | OK | Tiempo de res | 251 mseg | ★★★★☆ | □ |
| +18. | ✓ StatusSensorMQ2 | OK | OK | Tiempo de res | 185 mseg | ★★★★☆ | □ |
| +19. | ✓ StatusSensorDTH11Temperatura | OK | OK | Tiempo de res | 243 mseg | ★★★★☆ | □ |

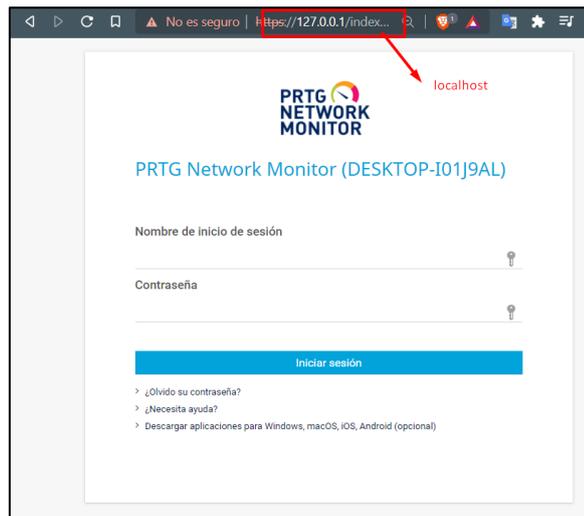
Fuente: elaborado por el autor

3.7.4.5.1. Interfaz De Visualización y Gestión De Red WSN.

Después de proceder a la instalación como se indica en el ANEXO B, se procede a ingresar al servidor mediante la dirección localhost (127.0.0.1) usando el cualquier navegador, al ingresar se debe autenticar como se muestra en la Figura 50.

Figura 50

Login de la herramienta de gestión

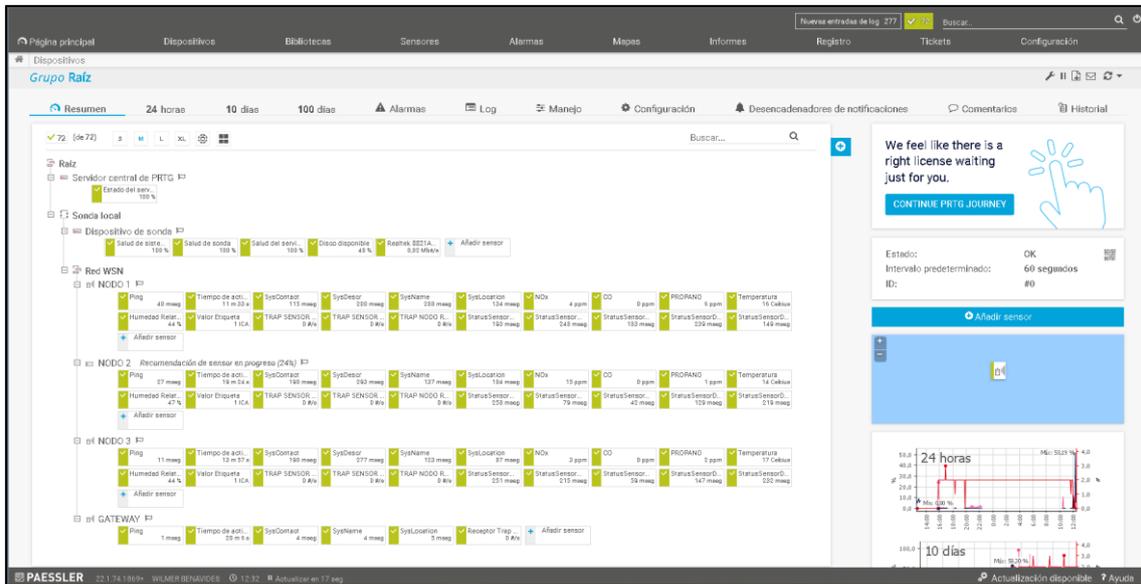


Fuente: elaborado por el autor

En la ventana principal se puede observar todos los grupos que posee la herramienta de gestión, en este caso se ha creado el grupo RED WSN como se muestra en la Figura 51 donde se muestra cada dispositivo denominados NODO y cada nodo posee sensores que son los objetos para gestionar.

Figura 51

Ventana principal de la herramienta PRTG NETWORK MONITOR

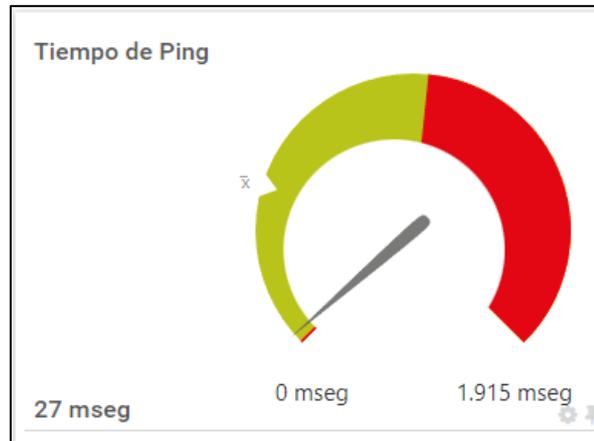


Fuente: elaborado por el autor

A cada nodo se le ha agregado los objetos que permiten tener monitorizado el dispositivo de la red, según el objeto se le ha otorgado un gráfico ya sea tipo velocímetro o tabla para mostrar la información, para los gráficos tipo velocímetro la aguja muestra el valor obtenido, el grafico posee valor máximos (color rojo) y mínimos (color verde) para una mejor percepción en la Figura 52 se muestra el grafico del ping do donde existe valores máximos de respuesta del ping (>1000 ms) o valores mínimos que serían valores de respuesta normales (1 - 100 ms).

Figura 52

Indicador tipo velocímetro de tiempo de respuesta de ping



Fuente: elaborado por el autor

Y para los objetos que poseen información en cadenas de texto se presentara como se muestra en la Figura 53 donde se puede apreciar una tabla donde se muestra estado de dispositivo y el contenido del mensaje que se recepto por el gestor.

Figura 53

Valores de objetos establecidos para cada NodoSensor

| Pos | Sensor | Estado | Mensaje |
|-----|---|--------|------------------------------|
| 1. | <input checked="" type="checkbox"/> Ping | OK | OK |
| 2. | <input checked="" type="checkbox"/> Tiempo de actividad | OK | OK |
| 3. | <input checked="" type="checkbox"/> SysContact | OK | WILMER BENAVIDES |
| 4. | <input checked="" type="checkbox"/> SysName | OK | NodeMCUEsp32 |
| 5. | <input checked="" type="checkbox"/> SysLocation | OK | Invernadero # 1 |
| 6. | <input checked="" type="checkbox"/> SysDescr | OK | SNMP Agent NODO 1 |
| 7. | <input checked="" type="checkbox"/> SysObjectID | OK | 1.3.6.1.2.1.experimental.WSN |

Fuente: elaborado por el autor

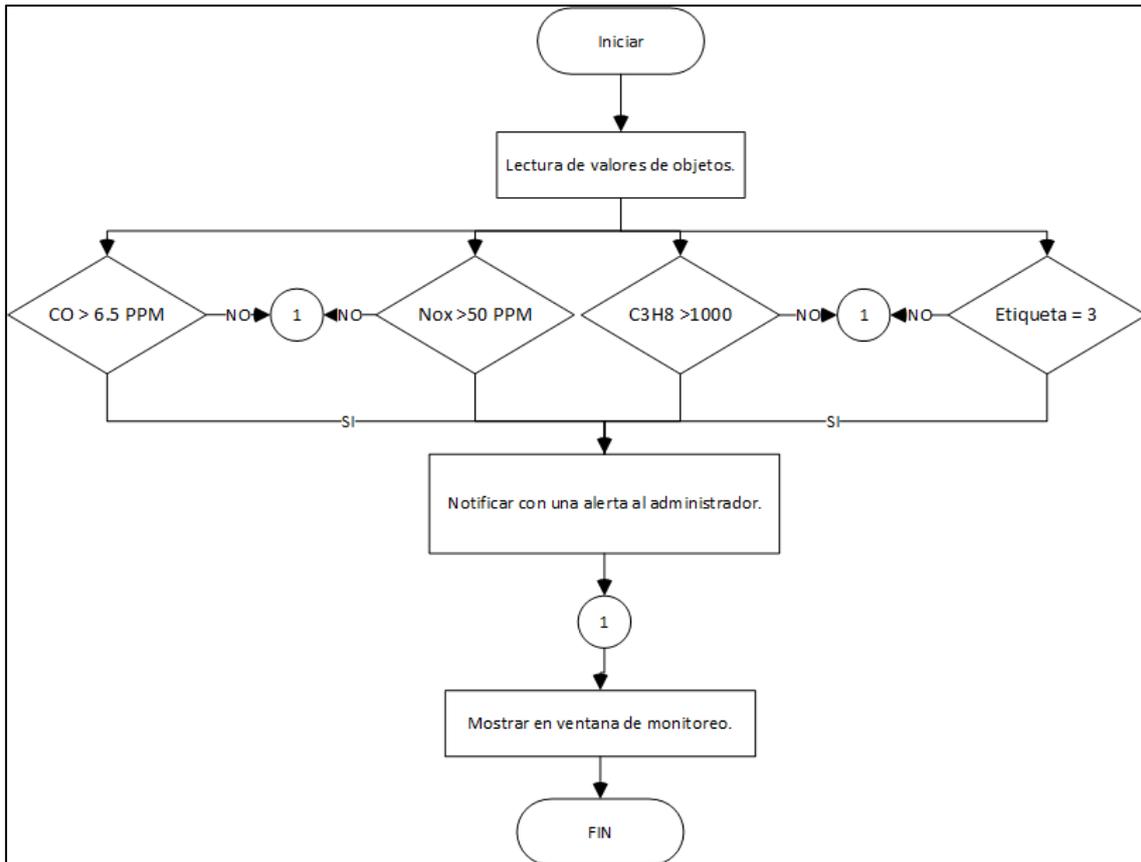
3.7.4.5.2. Diagrama de flujo del bloque de visualización

El diagrama de flujo muestra el ciclo de funcionamiento del bloque de visualización, como se muestra en la Figura 54 se inicia con la lectura de los valores de cada objeto enviado por los agentes, los valores de los objetos corresponden la los valores en ppm de la concentración de gas en el ambiente obtenido por el bloque de adquisición

de datos(sensores), se realiza la comparación con la tabla de concentración y se muestra de manera grafica los valores obtenidos, en caso de superar los límites permitidos se notifica al administrador.

Figura 54

Diagrama de flujo del ciclo de funcionamiento del bloque de visualización



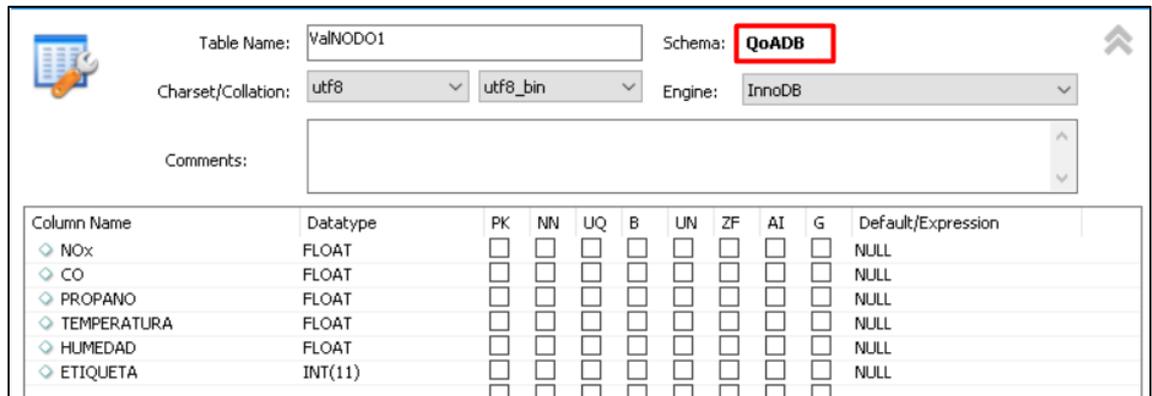
Fuente: elaborado por el autor

3.7.4.5.3. Base De Datos

Los datos obtenidos mediante el bloque de adquisición de datos, se los almacena en una base de datos MYSQL denominada QoADB, donde todos los valores obtenidos por cada sensor y etiqueta obtenida por el algoritmo de clasificación supervisada (NOx, CO, PROPANO, TEMPERATURA, HUMEDAD, ETIQUETA) y por cada nodo (NODO1, NODO2, NODO3), en la Figura 55 se muestra la creación de cada columna y tipo de dato (FLOAT, INTEGER) para cada uno de los valores obtenidos por los sensores.

Figura 55

Creación de tablas para la base de datos



Fuente: elaborado por el autor

3.7.5. Diseño De La Red WSN

La red WSN está conformada por cuatro nodos sensores y el Gateway que es el encargado de recepción y retransmisión de los datos entre nodos sensores y la estación base, Para la implementación en el invernadero es necesario conocer las dimensiones el invernadero y el área de cobertura del protocolo de comunicación IEEE 802.11, por lo cual, teniendo las dimensiones del invernadero de 50 metros de largo por 25 metros de ancho, se analiza cual sería la ubicación ideal del Gateway. En la figura 56 se muestra las dimensiones del invernadero capturada por medio de Google maps.

Figura 56

Vista del invernadero por medio de Google maps

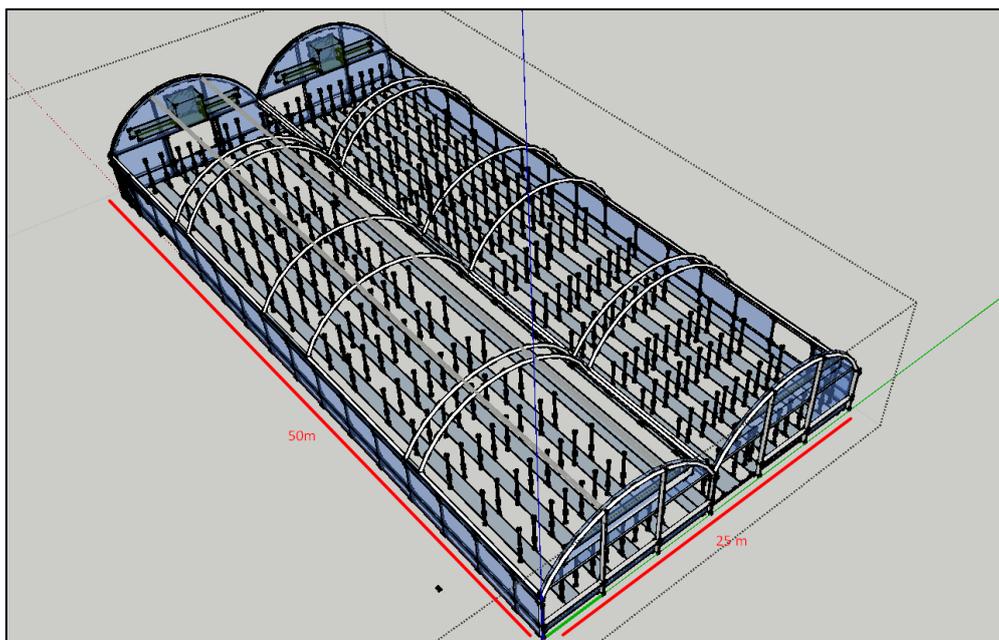


Fuente: Adaptado de Google maps (2022), Finca florícola X

Con las dimensiones del invernadero se procede a modelarlo en un software de modelado 3D SketchUp, para lograr determinar el mejor lugar para la ubicación de los nodos. En la Figura 57 se muestra el invernadero modelado en 3D.

Figura 57

Invernadero modelado en 3D

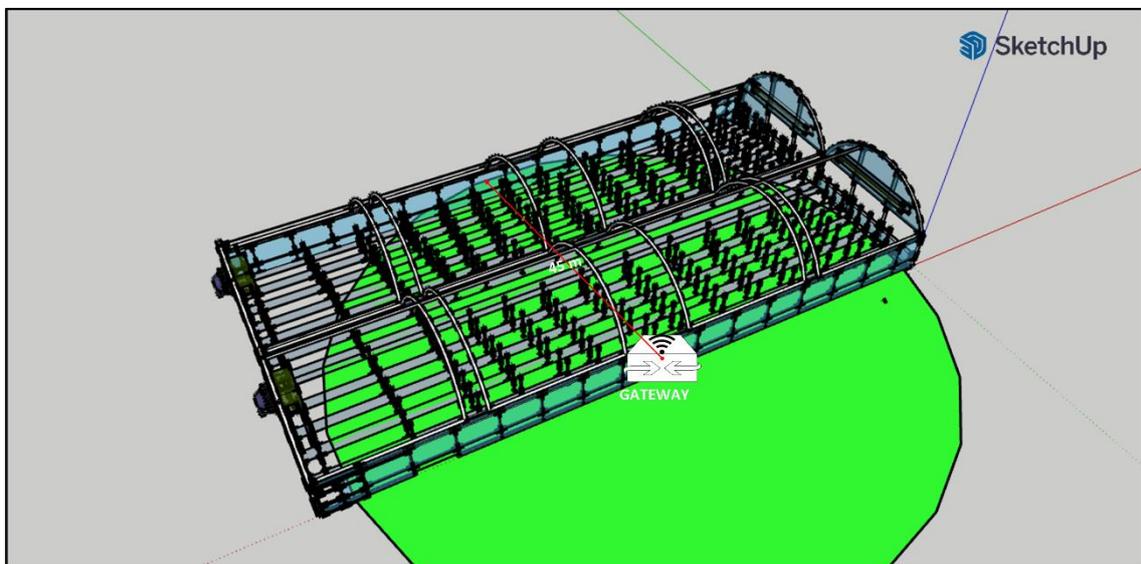


Fuente: elaborado por el autor

Con las dimensiones del invernadero y de acuerdo con los requerimientos antes mencionados el protocolo de comunicación inalámbrica se tiene al protocolo IEEE 802.11g el cual proporciona un rango de cobertura menor de 100 metros en espacios libres, se procede a verificar la cobertura, esto se realizó verificando la potencia de la señal al alejarse del Gateway y se llega a una distancia de 40 metros desde el Gateway hasta la estación con una potencia aceptable, por lo cual se procede a determinar que el Gateway debe ubicarse en el centro del invernadero. En la Figura 58 se muestra la ubicación del Gateway.

Figura 58

Ubicación del Gateway en el invernadero

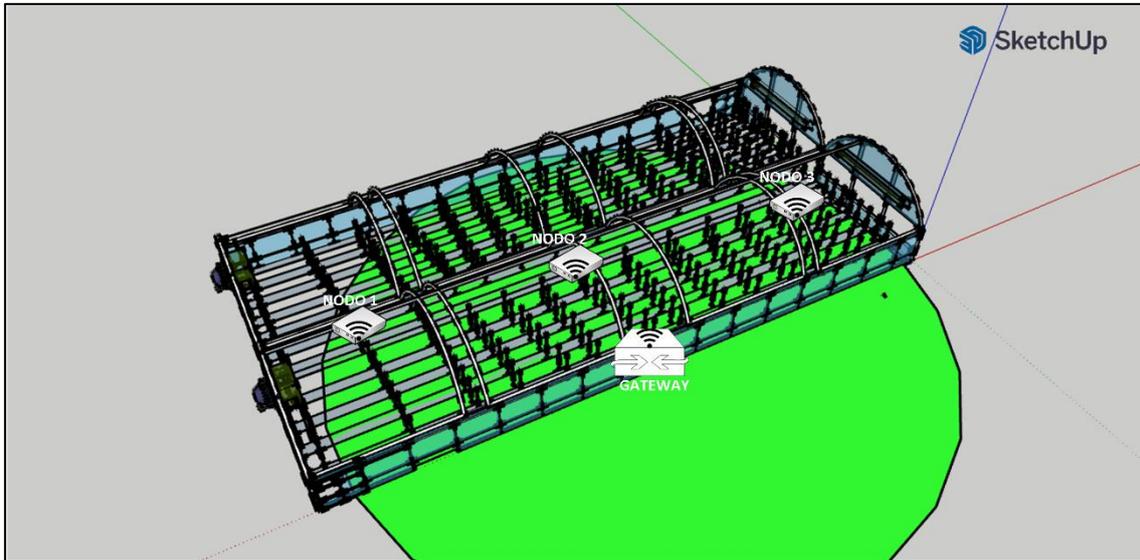


Fuente: elaborado por el autor

Al poseer una radiación omnidireccional, es decir, se irradia a todos lados se puede suponer como una circunferencia el radio de radiación de la antena del Gateway, por lo cual se debe ubicar los nodos de la manera que formen parte de la celda del Gateway. En la Figura 59 se muestra la ubicación de los tres nodos sensores.

Figura 59

Distribución de nodos en el invernadero



Fuente: elaborado por el autor

3.7.5.1. Direccionamiento

Como se especifica en la arquitectura el protocolo usado para la capa de red es el protocolo de internet versión 4 IPv4, el direccionamiento se realizará de acuerdo con cada elemento que compone la red WSN. En la Tabla 30 se muestra cómo se realizó la asignación de direcciones ip.

Tabla 30

Direccionamiento

| Equipo | Interfaz | Subred | Mascara de subred | Dirección Ipv4 |
|----------------|------------------|-------------|-------------------|----------------|
| Gateway | Wi-Fi | 192.168.4.0 | /24 | 192.168.4.1 |
| Nodo 1 | Wi-Fi | 192.168.4.0 | /24 | 192.168.4.2 |
| Nodo 2 | Wi-Fi | 192.168.4.0 | /24 | 192.168.4.3 |
| Nodo 3 | Wi-Fi | 192.168.4.0 | /24 | 192.168.4.4 |
| Estación base | Wi-Fi | 192.168.4.0 | /24 | 192.168.4.5 |
| Servidor MySQL | Conector puente. | 192.168.4.0 | /24 | 192.168.4.6 |

Fuente: elaborado por el autor

3.8.Implementación

En esta sección se realiza la implementación de hardware como software en cada uno de los bloques dando como resultado el sistema final.

3.8.1. Implementación De Hardware

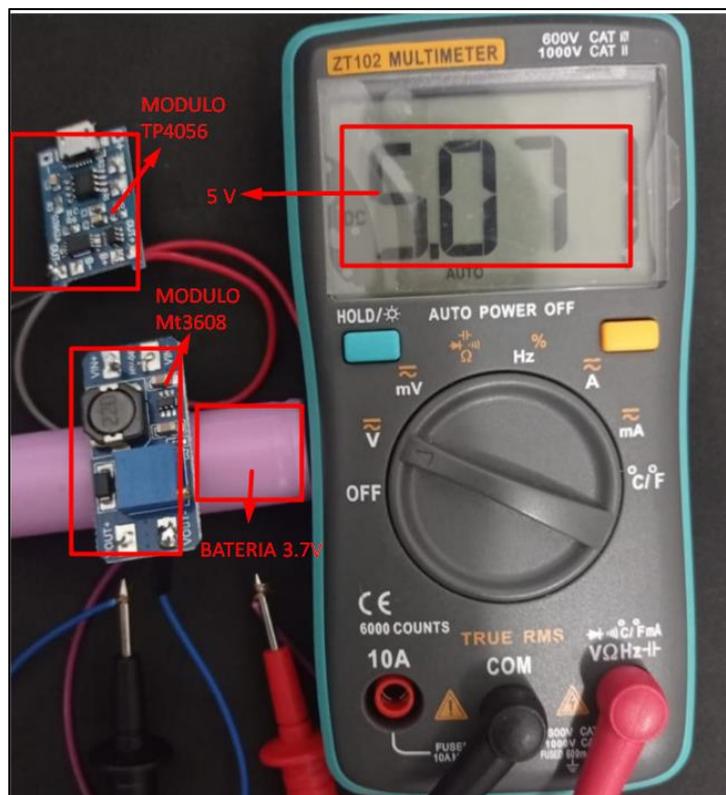
En la implementación del hardware se procede a la construcción y adaptación de cada uno de los dispositivos que formaran parte del sistema.

3.8.1.1.Bloque De Alimentación

Para el bloque de alimentación se procede a realizar la conexión entre todos los dispositivos y verificar con un multímetro el valor del voltaje final de 5 voltios. En la Figura 60 se muestra la distribución y conexión de los dispositivos.

Figura 60

Distribución y conexión de bloque de alimentación



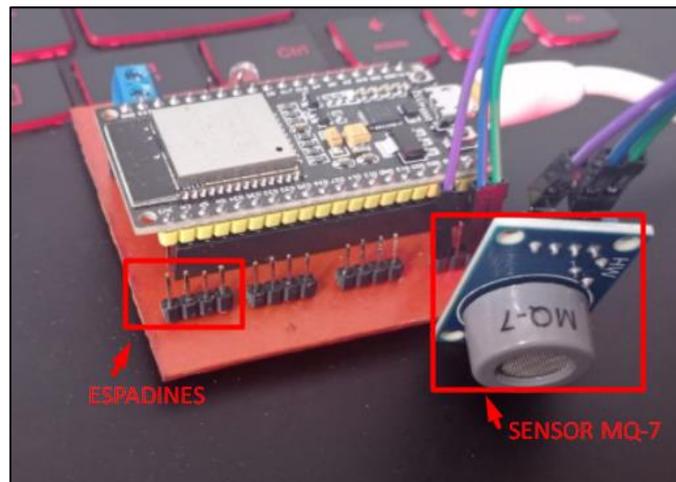
Fuente: elaborado por el autor

3.8.1.2. Bloque De Adquisición De Datos

Para la parte de adquisidor de datos que corresponde a los sensores se ha diseñado una placa impresa en una baquelita virgen, donde mediante el uso de espadines se conectarán las líneas VCC, GND Y DATA. En la Figura 61 se muestra como se realizó la conexión de cada una de las líneas.

Figura 61

Conexión de líneas de sensor con la PCB



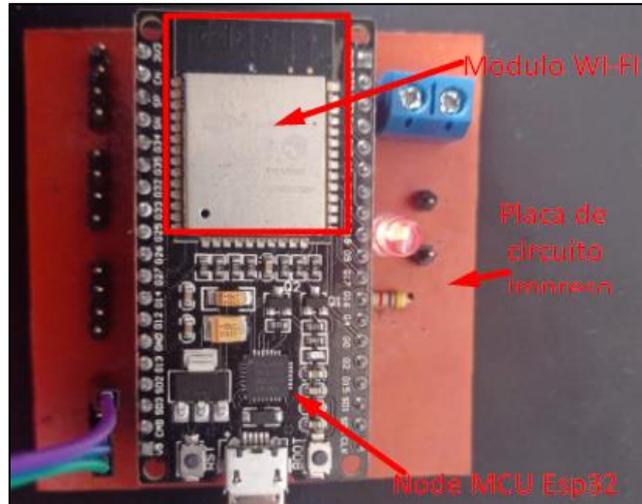
Fuente: elaborado por el autor

3.8.1.3. Bloque De Procesamiento Y Comunicación

Para el bloque de procesamiento y comunicación se agregó en la PCB los espadines hembra para acoplar al Node MCU Esp32 con la placa y las líneas de alimentación y comunicación con los sensores y para la comunicación el Node posee un módulo de comunicación WI-FI bajo el protocolo IEEE 802.11g, en la Figura 62 se puede observar la adaptación del Node MCU Esp32 con la PCB.

Figura 62

Anclaje del Node MCU esp32 con la placa



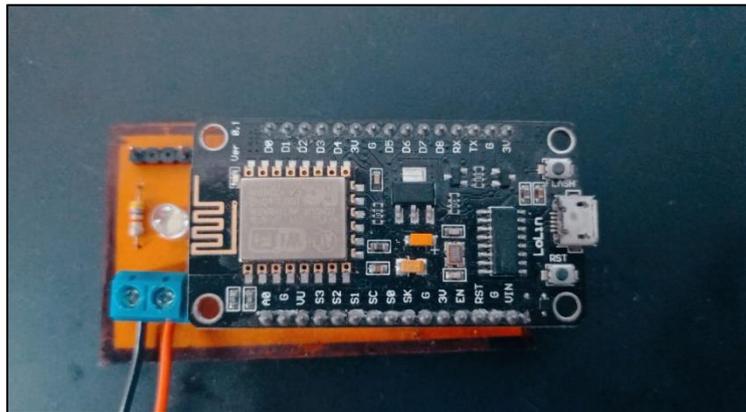
Fuente: elaborado por el autor

3.8.1.4. Bloque De Encaminamiento De Datos

Para este bloque se realizó la placa de circuito impreso donde se agregó espadines hembra para anclar al Node MCU esp8266 con la placa y las líneas de alimentación, en la figura 63 se muestra como el anclaje del Node MCU esp8266 con la placa.

Figura 63

Anclaje del Node MCU esp8266 con la placa de circuito impreso



Fuente: elaborado por el autor

3.8.2. Implementación De Software

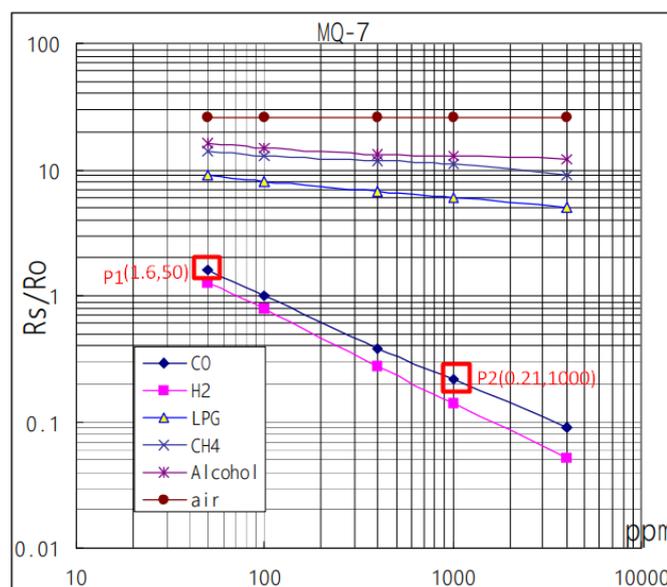
En la implementación de software se realiza la programación en C++ del software IDE ARDUINO donde se establece todos los parámetros y características necesarias para el funcionamiento de sensores, módulo de comunicación y modelo de gestión de red SNMP.

3.8.2.1. Bloque De Adquisición De Datos

Para la correcta adquisición de datos es necesario la calibración de los sensores de tal forma que los datos obtenidos sean fiables, en este caso se usa la gráfica de la curva de sensibilidad proporcionada por el fabricante de los sensores MQ, donde dependiendo de la gráfica se puede tomar los valores para encontrar ($x =$ eje de las abscisas (R_s/R_o), $y =$ eje de las ordenadas (ppm)). Con estos valores se procede a utilizar la *Ecuación 13* especificada anteriormente. Para realizar los cálculos se tomará dos puntos de la gráfica de sensibilidad establecida en el datasheet del sensor MQ-7 para CO, como se muestra en la Figura 64.

Figura 64

Selección de dos puntos en la tabla de sensibilidad del sensor MQ7



Se procede a calcular y, x y m con los puntos establecidos.

$$x = \log\left(\frac{Rs}{Ro}\right) = \log(50) = 1.69$$

$$y = \log(ppm) = \log(1.6) = 0.20$$

$$m = \frac{\log(1000/50)}{\log(0.20/1.69)}$$

$$m = -1.4753$$

Encontrados los valores se procede a crear un método que ayude a la calibración de cada sensor, primero se crea las variables con los valores calculados como se muestra en la Figura 65.

Figura 65

Declaración de variables para calibración de sensores.

```

//*****define la entrada analogica para el sensor
#define MQ2 (27)
#define MQ7 (14)
#define MQ135 (12)
//*****define el valor de la resistencia RL kohms
#define PROPANO_VALOR (5)
#define CO_VALOR (20)
#define Nox_VALOR (20)
//*****variable para valor del gas en ppm
#define GAS_Propano (0)
#define GAS_CO (0)
#define GAS_NOx (0)
//*****valores calculados(x,y,m)
float PROPANOCurve[3] = {2.3, 0.32, -0.481};
float COCurve[3] = {1.69, 0.20, -1.4753};
float NOxCurve[3] = {1, 0.38, -0.366};
//***** Resistencia RO estimada
float RoMQ2 = 1000;
float RoMQ7 = 100;
float RoMQ135 = 100;

```

Fuente: elaborado por el autor

Posteriormente se crea un método para la lectura del puerto analógico del sensor, por lo cual se toma 5 muestras que ayudaran a suavizar la señal ingresada como un filtro digital, para obtener el valor del gas en partes por millón (ppm) se aplica la *Ecuación 9* donde los valores obtenidos por medio de la ecuación de la función logarítmica antes

mencionada. En la Figura 66 se muestra la implementación de la ecuación a sketch de Arduino.

Figura 66

Métodos creados para la calibración, suavizado de la señal y calculo ppm del gas

```
//lectura de la entrada analogica del sensor
float lecturaMQ7(int mq_pinMQ7) {
  int i;
  float rsMQ7 = 0;
  for (i = 0; i < 5; i++) {
    rsMQ7 += calc_res7(analogRead(mq_pinMQ7));
    delay(50);
  }
  rsMQ7 = rsMQ7 / 5;
  return rsMQ7;
}
//aplicacion de la ecuacion para calcular el porcentaje de gas en PPM=10^{log [(RS/RO)-b]/m}
int porcentaje_gasCO(float rs_ro_ratioMQ7, float *COcurve) {
  return (pow(10, (((log(rs_ro_ratioMQ7) - COcurve[1]) / COcurve[2]) + COcurve[0])));
}
```

Fuente: elaborado por el autor

3.8.2.2. Bloque De Procesamiento Y Comunicación

En este bloque se desarrolla el algoritmo de clasificación supervisada KNN, el mismo que utiliza una matriz de entramiento para poder clasificar los valores obtenidos en base a la cercanía que tenga cada muestra con el valor establecido en la matriz de entrenamiento.

3.8.2.2.1. Selección De Datos

Para la selección de datos se procede a tomar muestras sometiendo a los sensores a varios ambientes como aire limpio y con presencia de productos agroquímicos utilizados a la hora de realizar la fumigación de las rosas. Donde los datos que se recolectados son NOx, CO, Propano, temperatura y humedad, en la Figura 67 se muestra un resumido de los datos obtenidos.

Figura 67

Datos adquiridos por los sensores en diferentes escenarios

| Propano | CO | Nox | Temperatura | Humedad |
|---------|------|-------|-------------|---------|
| 78.18 | 1.53 | 66.72 | 14 | 63 |
| 75.02 | 1.51 | 65.01 | 14 | 63 |
| 70.92 | 1.49 | 64.26 | 14 | 63 |
| 68.74 | 1.48 | 61.80 | 14 | 63 |
| 65.71 | 1.48 | 57.72 | 14 | 63 |
| 66.56 | 1.48 | 56.45 | 14 | 63 |
| 69.71 | 1.50 | 54.40 | 14 | 63 |
| 70.39 | 1.52 | 54.40 | 14 | 63 |
| 66.56 | 1.53 | 52.41 | 14 | 63 |
| 266.57 | 1.47 | 15.22 | 12 | 62 |
| 272.17 | 1.50 | 15.24 | 12 | 62 |
| 275.75 | 1.53 | 15.24 | 12 | 62 |
| 275.75 | 1.55 | 15.14 | 12 | 62 |
| 276.36 | 1.56 | 15.11 | 12 | 62 |
| 274.85 | 1.59 | 15.16 | 12 | 62 |
| 271.28 | 1.61 | 15.14 | 12 | 62 |
| 266.28 | 1.61 | 15.03 | 12 | 62 |
| 262.81 | 1.63 | 14.93 | 12 | 62 |
| 255.42 | 1.64 | 14.93 | 12 | 62 |
| 248.77 | 1.65 | 14.72 | 12 | 62 |
| 243.08 | 1.65 | 14.93 | 12 | 62 |
| 236.99 | 1.67 | 14.64 | 12 | 62 |
| 230.52 | 1.66 | 14.64 | 12 | 62 |
| 223.96 | 1.67 | 14.46 | 12 | 62 |
| 217.09 | 1.68 | 14.38 | 12 | 62 |
| 211.82 | 1.68 | 14.38 | 12 | 62 |
| 204.13 | 1.66 | 14.36 | 12 | 62 |
| 198.70 | 1.67 | 14.13 | 12 | 62 |

Fuente: elaborado por el autor

3.8.2.2.2. Limpieza de datos

Para la clasificación de los datos y filtrado es necesario la asignación de etiquetas que indiquen el nivel de concentración de gas en el aire como está establecido en la Tabla 13, la cual proporciona los niveles de toxicidad de los componentes dependiendo del valor en partes por millón(ppm). En la Tabla 31 se muestra los rangos y niveles que se estableció a cada uno de los datos adquiridos.

Tabla 31

Valores límites de cada contaminante común en la atmosfera en partes por millón (PPM)

| CO | NO _x | PROPANO | Etiqueta | Valor |
|-----------|-----------------|------------|----------|-----------|
| 0 – 4.4 | 0 – 30 | 0- 500 | 1 | Normal |
| 4.5 – 6.4 | 30 – 50 | 500 – 1000 | 2 | Aceptable |
| >6.5 | >50 | >1000 | 3 | Alerta |

Fuente: elaborado por el autor

Después de la asignación de las etiquetas se procede limpiar los datos usando el software RStudio, se trabajará con diferentes filtros CNN, AENN, BBNR, DROP1, DROP2 y DROP3 que corresponden a la librería NoiseFiltersR los cuales se encargan de la selección las instancias más relevantes, para este proceso se cuenta con una base de datos de 508 instancias. Al aplicar los diferentes filtros se tiene que el filtro CNN proporciona un mejor porcentaje selección, dando como resultado la filtración del 95.2755 % de la base de datos. En la Figura 68 se muestra la resultante de la aplicación del filtro CNN.

Figura 68

Resultados del filtrado CNN

```
Call:
CNN(x = training_set, k = 5)

Results:
Number of removed instances: 242 (95.27559 %)
Number of repaired instances: 0 (0 %)
~matriz_con ~matriz_confiltranada
```

Fuente: elaborado por el autor

Esta matriz resultante se la utilizara como la matriz de entrenamiento para el algoritmo de clasificación supervisada KNN. En la Figura 69 se muestra una porción de la matriz de entrenamiento.

Figura 69

Matriz resultante del proceso de limpieza.

| | Propano | CO | Nox | Temperatura | Humedad | Etiqueta |
|-----|---------|------|-------|-------------|---------|----------|
| 249 | 448.10 | 2.86 | 38.56 | 12 | 66 | 1 |
| 260 | 461.50 | 3.38 | 39.91 | 11 | 66 | 1 |
| 261 | 465.50 | 3.38 | 39.96 | 11 | 66 | 1 |
| 262 | 466.64 | 3.38 | 39.05 | 11 | 66 | 1 |
| 264 | 467.23 | 4.19 | 39.20 | 11 | 66 | 1 |
| 271 | 476.56 | 4.20 | 39.45 | 11 | 66 | 1 |
| 277 | 484.30 | 4.21 | 39.60 | 11 | 67 | 1 |
| 294 | 523.33 | 4.53 | 41.45 | 11 | 67 | 2 |
| 296 | 525.98 | 4.53 | 41.55 | 11 | 67 | 2 |
| 297 | 526.64 | 4.55 | 41.65 | 11 | 67 | 2 |
| 300 | 539.44 | 4.56 | 41.90 | 11 | 67 | 2 |
| 316 | 595.44 | 4.63 | 44.97 | 11 | 68 | 2 |
| 317 | 600.62 | 4.64 | 45.30 | 11 | 68 | 2 |
| 320 | 606.87 | 4.65 | 47.43 | 11 | 68 | 2 |
| 321 | 615.59 | 4.65 | 47.85 | 11 | 68 | 2 |
| 327 | 631.84 | 4.67 | 50.59 | 12 | 68 | 2 |
| 330 | 636.81 | 4.68 | 53.24 | 12 | 68 | 2 |
| 334 | 650.84 | 4.68 | 54.40 | 12 | 68 | 2 |
| 340 | 683.75 | 4.69 | 57.24 | 12 | 68 | 2 |

Fuente: elaborado por el autor

3.8.2.2.3. Minería de datos

Para realizar la minería de datos se procede a implementar el algoritmo de clasificación supervisada KNN, este algoritmo se hace uso de la matriz de entrenamiento, para este algoritmo es fundamental la elección del número total de vecinos a utilizar, por lo cual se realizó pruebas para determinar el número de vecinos que proporciona la mejor eficiencia del algoritmo. En la Tabla 32 se muestra los resultados obtenidos.

Tabla 32

Pruebas de numero óptimo de vecinos para algoritmo KNN

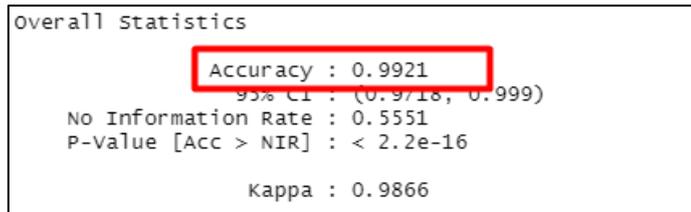
| Semilla | Número de vecinos óptimo para algoritmo KNN | | |
|---------|---|-------|-------|
| | K=1 | K=3 | K=5 |
| 5008 | 98.82 | 99.21 | 98.43 |
| 2054 | 95.92 | 98.98 | 97.96 |

Fuente: elaborado por el autor

En la Figura 70 se muestra la eficiencia de utilizar un número de vecinos $K=3$, con 5008 datos de semilla, estableciéndose como el número ideal para el algoritmo de clasificación.

Figura 70

Resultado de aplicar algoritmo KNN con número de vecinos $K=3$



Fuente: elaborado por el autor

Una vez establecido el número de vecinos y probado su efectividad, se procede a implementar el algoritmo de clasificación supervisada KNN en el microcontrolador, donde la matriz de entrenamiento y el algoritmo de clasificación procesaran los datos recopilados por el bloque de adquisición de datos y proporcionaran un índice de calidad del aire.

Se procede a asignar las variables que servirán para el proceso de clasificación, donde se crea una matriz que comprende a los valores obtenidos por los sensores MQ-2, MQ7 y MQ135, estos datos ingresados en esta matriz serán los que se clasificarán en base a la distancia de k vecinos y la matriz de entrenamiento. En la Figura 71 se muestra la asignación de variables.

Figura 71

Asignación de variables para algoritmo KNN

```
/* algoritmo de clasificación supervisada KNN *****
float matriz_prueba [3] = {entValorSensorMQ135, entValorSensorMQ7,
entValorSensorMQ2};
int fil = 49;
int columna = 4;
int vecinos = 3; // numero optimo de vecinos cercanos
int etiquetas = 3;
int col;
int fila = 0;
float sumatoria = 0;
float distancia;
float matriz_k [2][vecinos]; //Matriz que alberga los vecinos
int i = 0;
int j;
float etiqueta; // Valor que retorna del método Knn
float mayor = 0;
float aux;
float aux1; // variables auxiliares para ordenamiento del vector
float matriz_etiquetas [2][etiquetas]; //Matriz que alberga las
etiquetas
```

Fuente: elaborado por el autor

Se establece ciclos repetitivos que se encargan de ir ordenando y clasificando valor por valor, para posteriormente encontrar la distancia entre dos puntos, a los valores de la matriz de entrenamiento corresponden a punto uno y los datos entrantes serán el punto dos, la resultante de este proceso se guarda en una matriz. En la Figura 72 se muestra el proceso de clasificación.

Figura 72

Clasificación en base a distancia entre dos puntos.

```
for (; i < vecinos; i++) { //Matriz que alberga el número de vecinos
matriz_k[0][i] = 3000 + i;
matriz_k[1][i] = 0;
}
i = 0;
for (; i < etiquetas; i++) {
matriz_etiquetas[0][i] = i + 1;
matriz_etiquetas[1][i] = 0;
}
i = 0;

for (; fila < fil; fila++) { // ciclo de lectura de filas
for (col = 0; col < columna - 1; col++) {
//Diferencia de cuadrados entre dos puntos
sumatoria = sumatoria + pow(matriz_prueba[col] - matriz[fila][col], 2);
}
distancia = sqrt(sumatoria); // distancia Euclidiana
sumatoria = 0; // reinicio de variable para siguiente lectura
//comparación de nuevo dato con solo la distancia mayor
if (distancia < matriz_k[0][vecinos - 1]) {
matriz_k[0][vecinos - 1] = distancia;
matriz_k[1][vecinos - 1] = matriz[fila][columna - 1];
}
}
```

Fuente: elaborado por el autor

Con los datos clasificados es necesario ordenarlos de menor a mayor por lo cual se utiliza el algoritmo de clasificación llamado método de la burbuja que ordena los valores uno por uno intercambiándolos de acuerdo con el valor de cada elemento. En la figura 73 se muestra el algoritmo para ordenar los elementos.

Figura 73

Algoritmo para ordenar los elementos anteriormente clasificados

```
// ordenamiento del vector de menor a mayor utilizando el método "BURBUJA"
for (i = 0; i < vecinos; i++) {
    for (j = i + 1; j < vecinos; j++) {
        if (matriz_k[0][i] > matriz_k[0][j]) {
            aux = matriz_k[0][i];
            matriz_k[0][i] = matriz_k[0][j];
            matriz_k[0][j] = aux;
            //etiquetas
            aux1 = matriz_k[1][i];
            matriz_k[1][i] = matriz_k[1][j];
            matriz_k[1][j] = aux1;
        }
    }
}
```

Fuente: elaborado por el autor

Con los datos ordenados y clasificados se procede a asignar la etiqueta que corresponde a el índice de calidad del aire. En la Figura 74 se muestra el proceso de asignación de etiquetas.

Figura 74

Proceso de asignación de etiquetas

```
}
//Asignación de la etiqueta
for (i = 0; i < etiquetas; i++) {
    for (j = 0; j < vecinos; j++) {
        if (matriz_etiquetas[0][i] == matriz_k[1][j]) {
            matriz_etiquetas[1][i] = 1 + matriz_etiquetas[1][i];
        }
    }
}

for (i = 0; i < etiquetas; i++) {
    if (matriz_etiquetas[1][i] > mayor) { //Condición para evaluar y
asignar la etiqueta
        mayor = matriz_etiquetas[1][i];
        etiqueta = matriz_etiquetas[0][i]; //Variable que retorna la
función KNN
    }
}
entValorEtiqueta = etiqueta;
```

Fuente: elaborado por el autor

3.8.2.2.4. Envío De Datos

Para el envío de datos se ha adaptado el mensaje *Get.Response* del agente SNMP de tal forma que la información obtenida por los sensores y el valor resultante del algoritmo de clasificación supervisada KNN se los denominó un objeto más que poseen un OID específico para cada uno y de esta manera se envían los datos en un mensaje SNMP y a su vez encapsulados en un datagrama UDP.

En la Figura 75 se muestra cómo se asignó objetos que poseen su OID para cada uno de los sensores, donde se especifica el tipo de sensor, nombre, valor y el estado.

Figura 75

Asignación de objetos para cada uno de los sensores

```
char* oidentNodeDescripcion = ".1.3.6.1.3.1.1.1.0"; // Descripción general del nodo sensor
char* oidentNodeSwVersion = ".1.3.6.1.3.1.1.4.1.0"; // Versión del software instalada
char* oidentNodeSwFechaActualizacion = ".1.3.6.1.3.1.1.4.2.0"; // fecha de actualización
char* oidentTipoSensorMQ135 = ".1.3.6.1.3.1.1.2.1.1.0"; // Tipo de sensor
char* oidentNameSensorMQ135 = ".1.3.6.1.3.1.1.2.2.1.0"; // Nombre del sensor
char* oidentValorSensorMQ135 = ".1.3.6.1.3.1.1.2.3.1.0"; // Valor del sensor en ppm
char* oidentStatusSensorMQ135 = ".1.3.6.1.3.1.1.2.4.1.0"; // Estado del sensor
char* oidentTipoSensorMQ7 = ".1.3.6.1.3.1.1.2.1.2.0"; // Tipo de sensor
char* oidentNameSensorMQ7 = ".1.3.6.1.3.1.1.2.2.2.0"; // Nombre del sensor
char* oidentValorSensorMQ7 = ".1.3.6.1.3.1.1.2.3.2.0"; // Valor del sensor en ppm
char* oidentStatusSensorMQ7 = ".1.3.6.1.3.1.1.2.4.2.0"; // Estado del sensor
char* oidentTipoSensorMQ2 = ".1.3.6.1.3.1.1.2.1.3.0"; // Tipo de sensor
char* oidentNameSensorMQ2 = ".1.3.6.1.3.1.1.2.2.3.0"; // Nombre del sensor
char* oidentValorSensorMQ2 = ".1.3.6.1.3.1.1.2.3.3.0"; // Valor del sensor en ppm
char* oidentStatusSensorMQ2 = ".1.3.6.1.3.1.1.2.4.3.0"; // Estado del sensor
char* oidentTipoSensorDTH11Temperatura = ".1.3.6.1.3.1.1.2.1.4.0"; // Tipo de sensor
char* oidentNameSensorDTH11Temperatura = ".1.3.6.1.3.1.1.2.2.4.0"; // Nombre del sensor
char* oidentValorSensorDTH11Temperatura = ".1.3.6.1.3.1.1.2.3.4.0"; // Valor de Temperatura
char* oidentStatusSensorDTH11Temperatura = ".1.3.6.1.3.1.1.2.4.4.0"; // Estado del sensor
char* oidentTipoSensorDTH11HumedadRelativa = ".1.3.6.1.3.1.1.2.1.5.0"; // Tipo de sensor
char* oidentNameSensorDTH11HumedadRelativa = ".1.3.6.1.3.1.1.2.2.5.0"; // Nombre del sensor
char* oidentValorSensorDTH11HumedadRelativa = ".1.3.6.1.3.1.1.2.3.5.0"; // Valor humedad
char* oidentStatusSensorDTH11HumedadRelativa = ".1.3.6.1.3.1.1.2.4.5.0"; // Estado del sensor
char* oidentValorEtiqueta = ".1.3.6.1.3.1.1.1.1.0"; // Valor de la etiqueta
```

Fuente: elaborado por el autor

Además de usar objetos SNMP para envío de datos se ha implementado el envío directo a un servidor MYSQL desde los nodos sensores, donde se utilizó la librería *<MySQL_connection.h>* desarrollada específicamente para realizar la negociación y comunicación bidireccional entre el servidor MYSQL y el nodo sensor. Para el ingreso y registro de valores, se establece un arreglo de valores donde se especifica el tipo de dato y las variables en el orden que se encuentren asignadas en la base de datos, como se muestra en la Figura 76.

Figura 76

Envío de datos a servidor SQL.

```
/******Cargar librerías para conexión con servidor SQL* *****/
#include <MySQL_Connection.h>
#include <MySQL_Cursor.h>

/******Establecimiento de parámetros para conexión con servidor SQL* *****/

IPAddress server_addr(192, 168, 4, 6); // IP of the MySQL *server* here
char user[] = "root"; // MySQL user login username
char password[] = "admin123"; // MySQL user login password
WiFiClient client;
MySQL_Connection conn((Client *)&client);
MySQL_Cursor* cursor;

/******establecimiento de conexión con el servidor SQL* *****/
Serial.print("Connecting to SQL... ");
if (conn.connect(server_addr, 3306, user, password)) {
    Serial.println("OK.");
}
else {
    Serial.println("FAILED.");
}
/******Cargar valores en variable para registrarlos en el servidor SQL* *****/

char INSERT_SQL[] = "INSERT INTO QoADB.ValNODO3 (NOx, CO, PROPANO, TEMPERATURA, HUMEDAD,
ETIQUETA) VALUES (%s,%s,%s,%d,%d,%d)";
char buffer[128];

sprintf(buffer, INSERT_SQL, valor_NOx2, valor_CO2, valor_PROPANO2, valor_TEMP,
valor_HUM, valor_etiqueta);

if (conn.connected()) {
    cursor->execute(buffer);
}
```

Fuente: elaborado por el autor

3.8.2.2.5. *Agente SNMP*

Para el agente SNMP se le han incluido varios objetos que proporcionan la información del software implementado y objetos para verificar el estado del equipo y tiempo activo, además de a incluido un grupo System especificado en el RFC 1213 (MIB-II) ya que es un grupo obligatorio para cada sistema. En la figura 77 se muestra los OID asignados para el nodo sensor.

Figura 77

Creación de objetos de acuerdo con el RFC 1213

```

/***** RFC1213-MIB*****/
char* oidSysDescr = ".1.3.6.1.2.1.1.1.0"; // SysDescr
char* oidSysObjectID = ".1.3.6.1.2.1.1.2.0"; // SysObjectID
char* oidSysUptime = ".1.3.6.1.2.1.1.3.0"; // sysUptime (seconds)
char* oidSysContact = ".1.3.6.1.2.1.1.4.0"; // SysContact
char* oidSysName = ".1.3.6.1.2.1.1.5.0"; // SysName
char* oidSysLocation = ".1.3.6.1.2.1.1.6.0"; // SysLocation
char* oidSysServices = ".1.3.6.1.2.1.1.7.0"; // sysServices

/***** información de gestión MIB //RFC1213*****/
std::string sysDescr = "SNMP Agent NODO 1";
std::string sysObjectID = "1.3.6.1.2.1.experimental.WSN";
uint32_t sysUptime = 0;
std::string sysContactValue = " WILMER BENAVIDES";
std::string sysNameValue = " NodeMCUEsp32";
std::string sysLocationValue = "Invernadero # 1";
int sysServices = 2;

```

Fuente: elaborado por el autor

3.8.2.3. Bloque De Recepción Y Retransmisión

Para el bloque de encaminamiento de datos se ha establecido en modo AP al Node MCU esp32, ya que posee un módulo de comunicación WI-FI bajo el protocolo IEEE 802.11 b/g/n, en la Figura 78 se muestra la configuración para establecer en modo AP.

Figura 78

Configuración de Gateway

```
#include <ESP8266WiFi.h> // librerías para el módulo de comunicación
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
// SSID y la contraseña para la autenticación
#ifndef APSSID
#define APSSID "GATEWAY-WSN"
#define APPSK "123456789"
#endif

const char *ssid = APSSID;
const char *password = APPSK;

ESP8266WebServer server (80);

// mensaje de servidor http http://192.168.4.1

void handleRoot() {
  server.send(200, "text/html", "<h1>Conectado a WSN inervadora 1
</h1>");
}

void setup() {
  delay(1000);
  Serial.begin(115200);
  Serial.println();
  Serial.print("Configurando Access point...");
  WiFi.softAP(ssid, password);
  IPAddress myIP = WiFi.softAPIP();
  server.on("/", handleRoot);
  server.begin();
  Serial.println("HTTP server started");
}

void loop() {
  server.handleClient();
}
```

Fuente: elaborado por el autor

En el Gateway también se ha alojado el agente SNMP para que proporcione la información necesaria del software implementado, estado del equipo y tiempo activo, los objetos creados hacen referencia al grupo System del MIB-II establecido en el RFC 1213. En la Figura 79 se muestra los objetos creados.

Figura 79

Creación de objetos de acuerdo con el RFC 1213 para Agente SNMP Gateway

```
/****** RFC1213-MIB*****  
char* oidSysDescr = ".1.3.6.1.2.1.1.1.0"; // OctetString SysDescr  
char* oidSysObjectID = ".1.3.6.1.2.1.1.2.0"; // OctetString SysObjectID  
char* oidSysUptime = ".1.3.6.1.2.1.1.3.0"; // TimeTicks sysUptime (hundredths  
of seconds)  
char* oidSysContact = ".1.3.6.1.2.1.1.4.0"; // OctetString SysContact  
char* oidSysName = ".1.3.6.1.2.1.1.5.0"; // OctetString SysName  
char* oidSysLocation = ".1.3.6.1.2.1.1.6.0"; // OctetString SysLocation  
char* oidSysServices = ".1.3.6.1.2.1.1.7.0"; // Integer sysServices  
/****** informacion de gestion MIB //RFC1213*****  
std::string sysDescr = "SNMP Agent GATEWAY";  
std::string sysObjectID = "1.3.6.1.2.1.experimental.WSN";  
uint32_t sysUptime = 0;  
std::string sysContactValue = "WSN_GATEWAY";  
std::string sysNameValue = "NodeMCUEsp8266";  
std::string sysLocationValue = "Invernadero # 1";
```

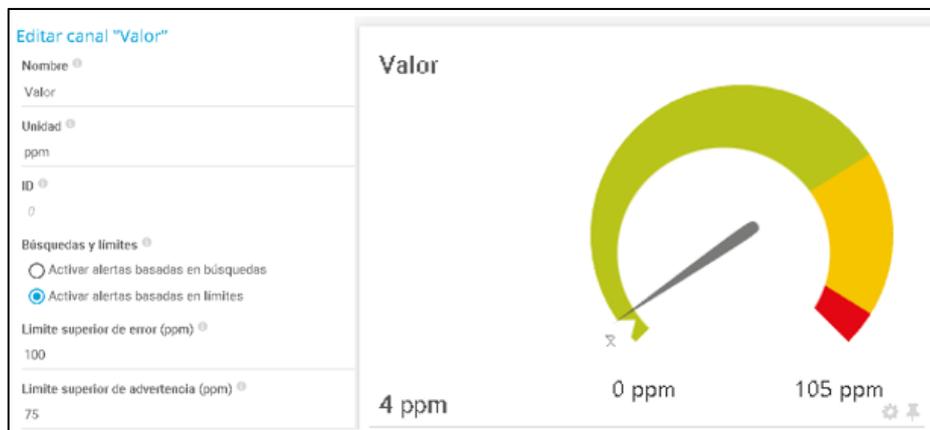
Fuente: elaborado por el autor

3.8.2.4. Bloque De Gestión

Para el bloque de gestión mediante el software PRTG NETWORK MONITOR se crea sensores para cada objeto que se desee monitorizar, donde para los objetos que poseen un valor como por ejemplo los sensores (NOx, CO, PROPANO y etiqueta) se puede establecer límites los cuales cambian de color dependiendo los valores máximos de advertencia (verde= normal, amarillo= advertencia y rojo= fallo), en la Figura 80 se muestra cómo se establece los límites.

Figura 80

Configuración de alertas



Fuente: elaborado por el autor

De igual manera para los TRAPS se configura un sensor para cada uno de las alertas que se puedan acontecer, donde se filtra por la concurrencia de los mensajes enviados por el agente SNMP. En la Figura 81 se muestran los mensajes enviados desde el nodo 3.

Figura 81

Traps enviados desde el NODO 3

| Mensajes trap | | | | |
|--------------------|-------------|-------|------------|---|
| | Source | Agent | Enterprise | Bindings |
| 18/8/2022 20:14:23 | 192.168.4.4 | | | RFC1213-MIB::sysObjectID.0 = SNMPv2-SMI::experimental.1.2.3.0 |
| 18/8/2022 20:14:23 | 192.168.4.4 | | | RFC1213-MIB::sysObjectID.0 = SNMPv2-SMI::experimental.1.2.3.0 |
| 18/8/2022 20:14:23 | 192.168.4.4 | | | RFC1213-MIB::sysObjectID.0 = SNMPv2-SMI::experimental.1.2.3.0 |
| 18/8/2022 20:14:23 | 192.168.4.4 | | | RFC1213-MIB::sysObjectID.0 = SNMPv2-SMI::experimental.1.2.3.0 |
| 18/8/2022 20:14:23 | 192.168.4.4 | | | RFC1213-MIB::sysObjectID.0 = SNMPv2-SMI::experimental.1.2.3.0 |
| 18/8/2022 20:14:23 | 192.168.4.4 | | | RFC1213-MIB::sysObjectID.0 = SNMPv2-SMI::experimental.1.2.3.0 |

Fuente: elaborado por el autor

En la Figura 82 se puede observar cómo los sensores asignados para los TRAPS (SENSOR NO READ y SENSOR DOWN) están recibiendo mensajes de error, por lo cual el color rojo de fallo.

Figura 82

Alerta de fallo a través de un mensaje trap

The screenshot shows a network monitoring interface with three nodes: NODO 1, NODO 2, and NODO 3. Each node has a list of sensors with their status. Node 3 is highlighted with a red box. Two sensors in Node 3 are shown with red error icons: 'TRAP SENSOR NO READ' and 'TRAP SENSOR DOWN'. A red arrow points to these two sensors.

| Node | Sensor | Status | Value |
|--------|---------------------|--------|----------|
| NODO 1 | Ping | OK | 20 mseg |
| | Humedad Relat... | OK | 40 % |
| NODO 2 | Ping | OK | 10 mseg |
| | Humedad Relat... | OK | 39 % |
| NODO 3 | Ping | OK | 3 mseg |
| | Humedad Relat... | OK | 38 % |
| NODO 3 | TRAP SENSOR NO READ | ERROR | 2.27 #/s |
| | TRAP SENSOR DOWN | ERROR | 2.27 #/s |

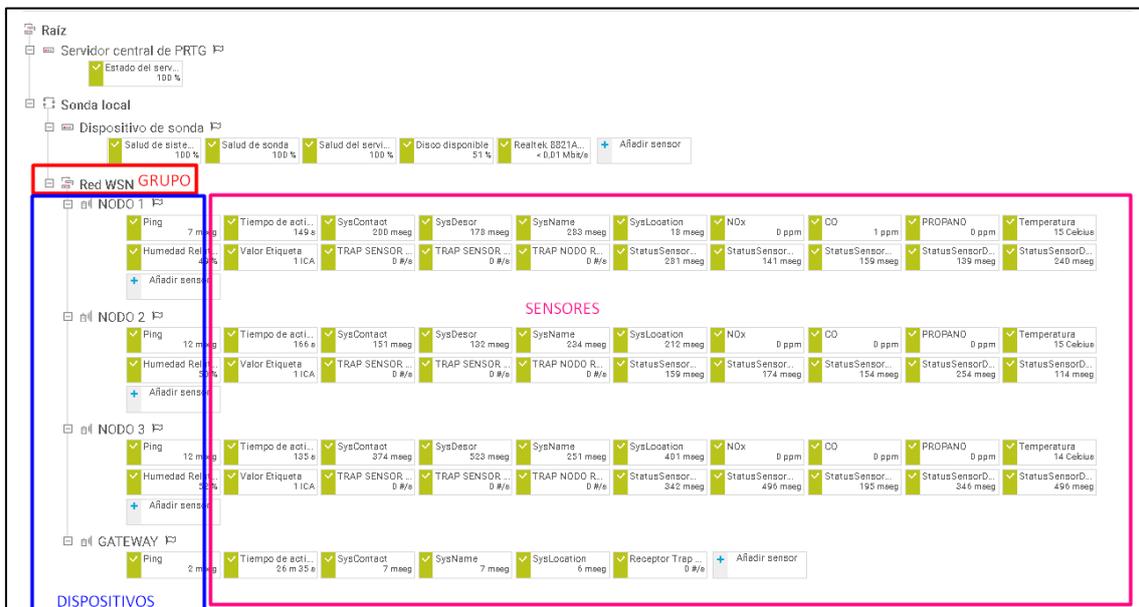
Fuente: elaborado por el autor

3.8.2.5. Bloque de visualización

En el bloque de visualización que se ha implementado el software PRTG NETWORK MONITOR el cual permite la implementación de grupos, dispositivos y sensores, donde el grupo de le asigno red WSN, los dispositivos serán los NODO 1, NODO 2, NODO 3 y GATEWAY, cada nodo posee sensores “objetos” que ayudan a obtener y observar la información de los recursos del dispositivo. En la Figura 83 se muestra la distribución en la ventana de monitoreo del software de monitoreo.

Figura 83

Distribución de elementos dentro de PRTG NETWORK MONITOR

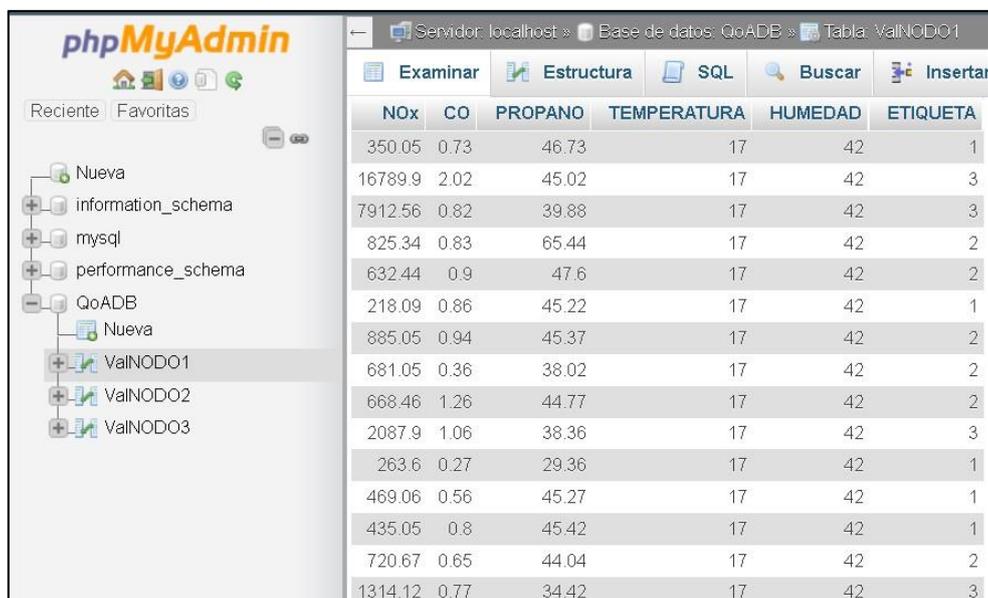


Fuente: elaborado por el autor

Para la base de datos en MYSQL se envían directamente desde el nodo y se guardan en tablas asignadas con el nombre del nodo y cada tabla posee las columnas de cada valor de los sensores e incluido la etiqueta. En la Figura 84 se muestra la visualización de los valores en la base de datos.

Figura 84

Valores guardados en base de datos



The screenshot shows the phpMyAdmin interface. On the left, a tree view shows the database structure with 'QoADB' expanded to show three tables: 'ValNODO1', 'ValNODO2', and 'ValNODO3'. The main area displays the 'ValNODO1' table with the following data:

| NOx | CO | PROPANO | TEMPERATURA | HUMEDAD | ETIQUETA |
|---------|------|---------|-------------|---------|----------|
| 350.05 | 0.73 | 46.73 | 17 | 42 | 1 |
| 16789.9 | 2.02 | 45.02 | 17 | 42 | 3 |
| 7912.56 | 0.82 | 39.88 | 17 | 42 | 3 |
| 825.34 | 0.83 | 65.44 | 17 | 42 | 2 |
| 632.44 | 0.9 | 47.6 | 17 | 42 | 2 |
| 218.09 | 0.86 | 45.22 | 17 | 42 | 1 |
| 885.05 | 0.94 | 45.37 | 17 | 42 | 2 |
| 681.05 | 0.36 | 38.02 | 17 | 42 | 2 |
| 668.46 | 1.26 | 44.77 | 17 | 42 | 2 |
| 2087.9 | 1.06 | 38.36 | 17 | 42 | 3 |
| 263.6 | 0.27 | 29.36 | 17 | 42 | 1 |
| 469.06 | 0.56 | 45.27 | 17 | 42 | 1 |
| 435.05 | 0.8 | 45.42 | 17 | 42 | 1 |
| 720.67 | 0.65 | 44.04 | 17 | 42 | 2 |
| 1314.12 | 0.77 | 34.42 | 17 | 42 | 3 |

Fuente: elaborado por el autor

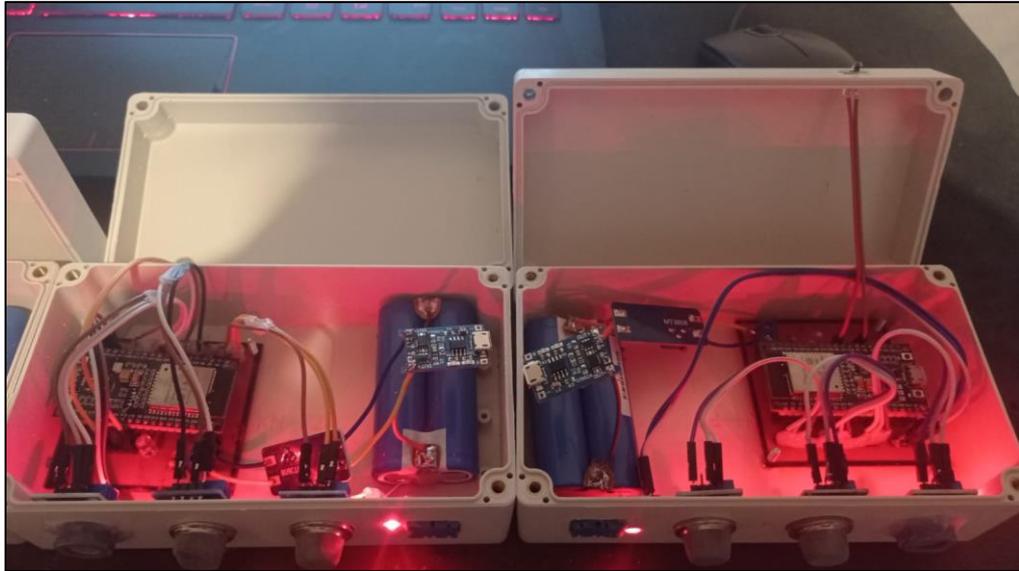
3.8.3. Implementación en el case al Gateway y nodos sensores

Después de diseñar y construir las placas de circuito impreso y acoplar todos los elementos se procede a adaptar los circuitos y elementos en un case que sea pequeño y hermético ya que se implementara en un lugar donde se utiliza agua para realizar la disolución de productos agroquímicos y atomizadores para dispersarlo en las plantas florícolas.

En la Figura 85 se muestra la implementación de los diferentes bloques (bloque de adquisición, bloque de alimentación y el bloque de procesamiento y comunicación) los cuales forman la estructura funcional del Nodo Sensor en un case que permitirá la fácil movilidad y ubicación de este en los invernaderos.

Figura 85

Nodo sensor adaptado en el case

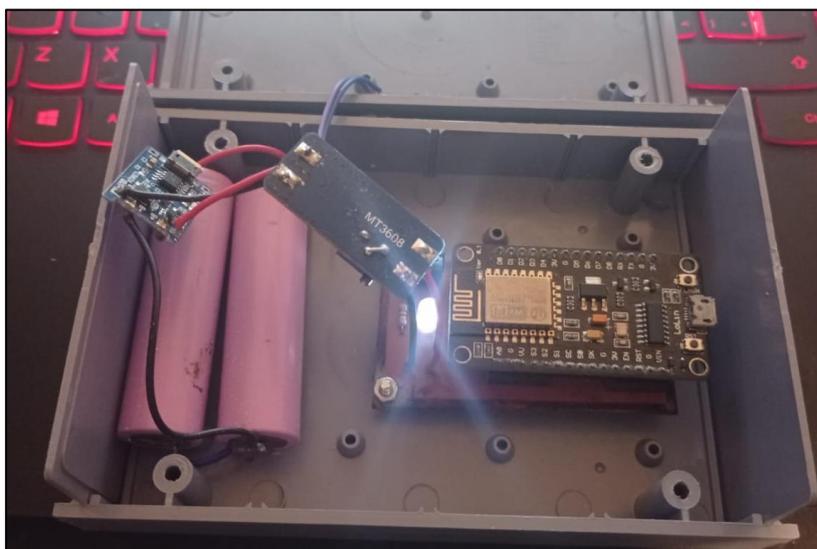


Fuente: elaborado por el autor

En la Figura 86 se muestra la implementación de los diferentes bloques (bloque de alimentación y bloque de recepción y retransmisión) que conforman la parte funcional del Gateway.

Figura 86

Implementación del Gateway en el case



Fuente: elaborado por el autor

3.9.Integración Del Sistema

Una vez culminado el diseño e implementación de las diferentes etapas para la elaboración del sistema de monitoreo y alerta, se procede a realizar la integración de los diferentes nodos sensores, Gateway y estación base en el invernadero florícola donde se realizar las diferentes pruebas de funcionamiento. En la Figura 87 se muestra la ubicación del nodo 1 en el invernadero florícola.

Figura 87

Nodos ubicados dentro del invernadero florícola.



Fuente: elaborado por el autor

Una vez ubicado los nodos sensores en la ubicación correspondiente, se procede a ubicar al Gateway en el invernadero florícola de tal manera que brinde cobertura a todos los nodos sensores y estación base, en la Figura 88 se muestra la integración del Gateway.

Figura 88

Integración de Gateway en florícola



Fuente: elaborado por el autor

CAPITULO IV. Procesamiento de la información y pruebas de funcionamiento

En el presente capítulo se aborda la verificación del sistema implementado realizando test de funcionalidad a cada bloque de la arquitectura del sistema. El propósito del proceso de Verificación es proporcionar evidencia objetiva de que un sistema o elemento del sistema cumple con los requisitos y características especificados, el proceso de verificación identifica las anomalías (errores, defectos o fallas) de los elementos del sistema implementados. Este proceso proporciona la información necesaria para determinar la resolución de las anomalías identificadas (ISO / IEC / IEEE 15288, 2015).

4.1.Verificación y Validación Del Funcionamiento Del Sistema, Mediante

Recomendaciones Especificadas En El Estándar ISO/IEC 30141:2018

El estándar ISO/IEC 30141:2018 está enfocado en proporcionar una arquitectura de referencia para IoT, donde se estandarizada las mejores prácticas en la industria. Para reforzar la seguridad y la protección, permitiendo desplegar sistemas fiables. Para el presente proyecto se utiliza las características funcionales para sistemas IoT descritas en el **capítulo 7.4** del estándar. A continuación, se realiza la verificación y validación del funcionamiento enfocado en comprobar el cumplimiento de características funcionales del sistema.

Se procede a comprobar el cumplimiento de los requerimientos establecidos mediante la verificación y validación de cada uno de los requerimientos. Realizando una serie de pruebas de funcionamiento al sistema en el invernadero, para la verificación exitosa fue necesario probar el funcionamiento en diferentes días.

4.1.1. Bloque De Alimentación

Se valida el cumplimiento de los requerimientos de sistema y de arquitectura especificados para el bloque de alimentación mediante una evaluación de cumplimiento

como se muestra en la Tabla 33, con el propósito de verificar su funcionalidad al momento de alimentar los diferentes componentes del sistema.

Tabla 33

Tabla de verificación de requerimientos de sistema y arquitectura para el bloque de alimenta

| Tabla de validación de requerimientos sistema y arquitectura | | | | | |
|---|----|--------------|---------------|---------------|---------------|
| Requerimientos | | SYSR2 | SYSH28 | SYSH30 | SYSH32 |
| Cumple | SI | ✓ | ✓ | ✓ | ✓ |
| | NO | | | | |

Fuente: elaborado por el autor

El sistema cumple en diseño y en funcionamiento los requerimientos establecidos, proporcionando línea VCC a 5 voltios estables y línea a tierra GND para cada uno de los Nodos y Gateway, además de poseer un batería de LiPo de 4600 mAh.

4.1.1.1. Test eléctrico

Para la verificación del bloque de alimentación se procede a realizar un test eléctrico como se muestra en la Tabla 34, con la finalidad de hacer la verificar del correcto encendido de los equipos y verificar el correcto valer del voltaje mediante la medición con un multímetro.

Tabla 34

Test Eléctrico para bloque de alimentación

| Test eléctrico | |
|-------------------------|---|
| Bloque de prueba | Bloque de alimentación |
| Descripción | Prueba de funcionamiento para bloque de alimentación. |
| Prerrequisitos: | |
| ✓ | Adquisición de batería de Lipo de 4600mAh 3.7 V |
| ✓ | Adquisición de módulo de carga de batería TP4056. |
| ✓ | Adquisición de modulo estabilizador y elevador de voltaje Mt3608. |

-
- ✓ Elaboración de placa PCB con líneas de para VCC y GND.
 - ✓ Adquisición de multímetro.

Pasos:

- ✓ Verificar valor de voltaje 5 voltios en la salida del bloque de alimentación.
- ✓ Verificar continuidad en placa PCB en líneas VCC y GND mediante el uso de multímetro.
- ✓ Verificar encendido del Node MCU Esp32.
- ✓ Verificar encendido del Node MCU Esp8266.
- ✓ Verificar encendido del sensor MQ-2.
- ✓ Verificar encendido del sensor MQ-7.
- ✓ Verificar encendido del sensor MQ-135.
- ✓ Verificar encendido del sensor DTH11.
- ✓ Verificar las subidas o caídas de voltaje en una línea de temporal.

Resultados esperados:

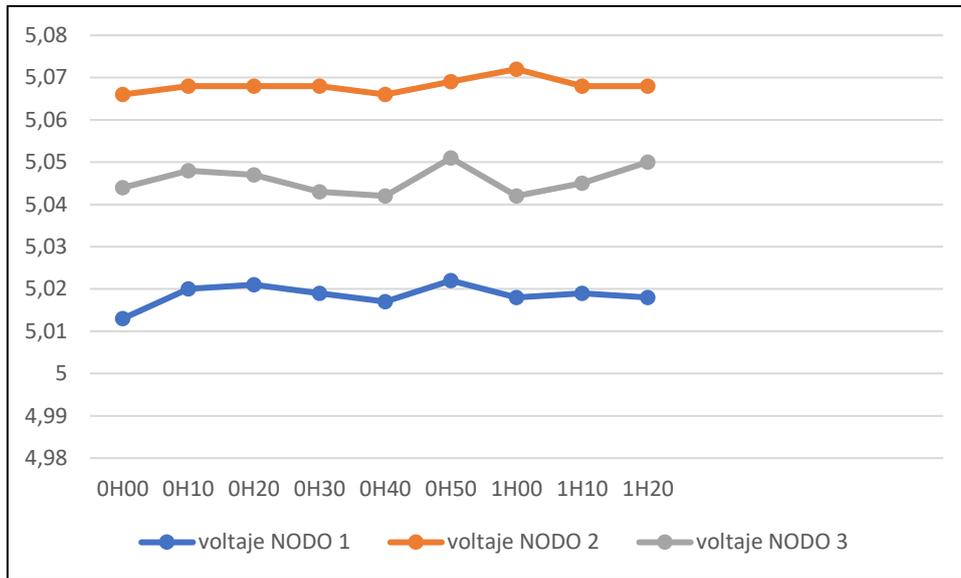
Tener un sistema de alimentación estable con la capacidad necesaria para mantener en funcionamiento de los elementos del sistema embebido por un tiempo prolongado.

Fuente: elaborado por el autor

Resultado obtenido: el bloque de alimentación funciona correctamente proporcionando 5 voltios con caídas y subidas de voltaje muy minimas. Para lo cual se realizó una medición durante periodos de tiempo. En la Figura 89 se muestra el resultando mediante grafico de línea del comportamiento del bloque de alimentación en una línea de tiempo, donde se puede apreciar los tiempos establecidos y los valores de voltaje obtenido en la medición de los tres nodos, este valor se lo obtuvo mediante la medición con el multímetro en las líneas de alimentación del circuito, donde se tiene una resultante lineal lo que indica que el voltaje suplementado posee unas caídas y subidas de voltaje mínimas y el voltaje de alimentación del sistema permanece estable y continuo.

Figura 89

Línea temporal de voltaje



Fuente: elaborado por el autor

La medición de los valores de voltaje se los realizo con el multímetro en las líneas de alimentación de la placa PCB para verificar los valores de voltaje en la Figura 90 se muestra el valor obtenido en la pantalla del multímetro.

Figura 90

Comprobación de voltaje



Fuente: elaborado por el autor

4.1.2. Bloque De Adquisición De Datos

Se valida el cumplimiento de los requerimientos de sistema y de arquitectura especificados para el bloque de adquisición de datos mediante un test de lectura de datos, el mismo que ayuda a la evaluación de cumplimiento como se muestra la Tabla 35. Con el propósito de verificar el cumplimiento de los requisitos en el diseño e implementación del sistema.

Tabla 35

Tabla de verificación de requerimientos de sistema y arquitectura para el bloque de adquisición de datos

| Tabla de validación de requerimientos sistema y arquitectura | | | | | | | | | | |
|---|--------------|--------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| Requerimientos | SYSR1 | SRSH3 | SRSH21 | SRSH21 | SRSH22 | SRSH23 | SRSH24 | SRSH25 | SRSH26 | SRSH27 |
| Cumple | SI | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | NO | | | | | | | | | |

Fuente: elaborado por el autor

El sistema al poseer 4 sensores es capaz de detectar una variedad de concentraciones de diferentes compuestos químicos relacionados con el cultivo de flores, con la calibración y precalentamiento especificado en las hojas técnicas, se puede obtener resultados altamente acertados.

4.1.2.1. Test de lectura de datos

Según el estándar ISO/IEC 30141 los datos recolectados deben poseer las "5V de datos" de volumen, velocidad, veracidad, variabilidad y variedad a menudo se aplican a los sistemas de IoT. Las 5V de datos se derivan de los grandes sistemas de datos, pero a menudo ocurre que los sistemas de IoT son la fuente de datos de gran volumen, entregada a gran velocidad a través de enlaces de red, cuya veracidad debe ser validada (ISO/IEC 30141, 2018).

Por lo cual se realiza el test de lectura de datos al bloque de adquisición de datos, mediante la captura y lectura de valores de cada sensor después de la calibración adecuada y el correcto almacenamiento de los datos en la base de datos. En la Tabla 36 se muestra el test aplicado al bloque de adquisición de datos.

Tabla 36

Test de adquisición de datos

| Test de lectura de datos | |
|--|---|
| Bloque de prueba | Bloque de adquisición de datos |
| Descripción | Prueba de funcionamiento para bloque de adquisición de datos. |
| Prerrequisitos: | |
| <ul style="list-style-type: none"> ✓ Adquisición de sensor MQ-2. ✓ Adquisición de sensor MQ-7. ✓ Adquisición de sensor MQ-135. ✓ Adquisición de sensor DTH11. ✓ Conexión de sensores a las líneas VCC, GND Y ANALÓGICO de la placa PCB ✓ Pre calentamiento de sensores MQ. | |
| Pasos: | |
| <ul style="list-style-type: none"> ✓ Verificar encendido de sensores. ✓ Calibración de sensores MQ-2, MQ-7, MQ-135, DTH11. ✓ Lectura de sensores MQ-2, MQ-7, MQ-135, DTH11. ✓ Verificar la las 5V de datos. ✓ Verificación de envío, recepción y almacenamiento de datos. | |
| Resultados esperados: | |
| Funcionamiento correcto de las líneas de alimentación, placa PCB y los sensores para permitir una lectura correcta de todos los sensores y una comunicación correcta entre sensores y microcontrolador. | |

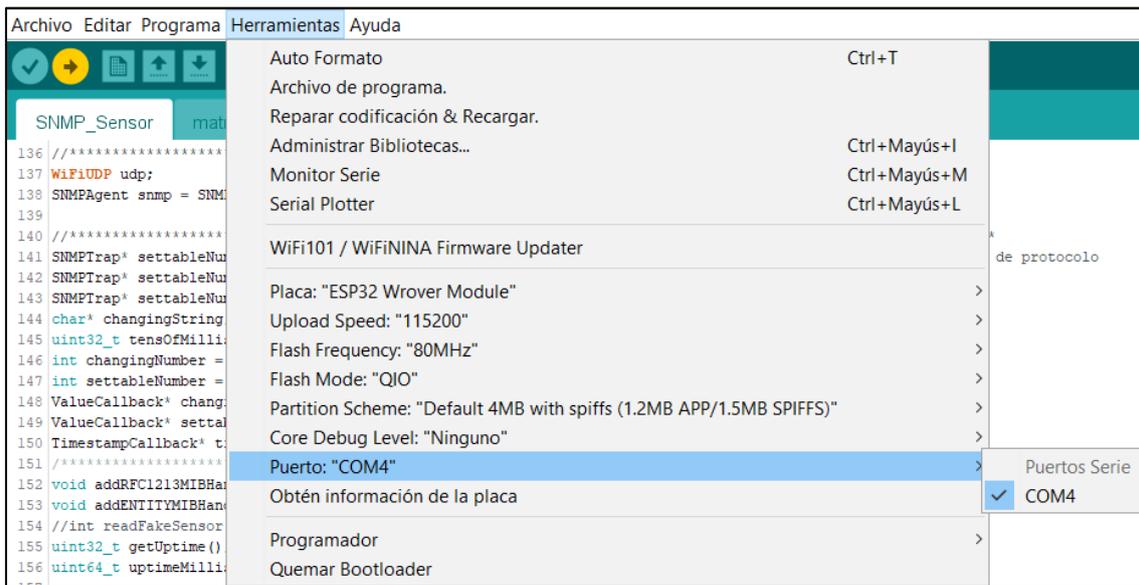
Fuente: elaborado por el autor

Resultados obtenidos: Se verifico el correcto acoplamiento de los sensores, la placa PCB y el Node Mcu Esp32, la autocalibración de los sensores y lectura de los datos

cada 500 ms, la impresión de los valores de cada sensor a través del puerto COM 4 en el Serial Plotter de Arduino para visualizar los datos en tiempo real es correcta y la verificación de envío, recepción y almacenamiento de datos en la base de datos MySQL es exitosa. En la Figura 91 la correcta lectura del puerto COM con Arduino.

Figura 91

Comunicación serial por puerto COM de nodo sensor y ordenador

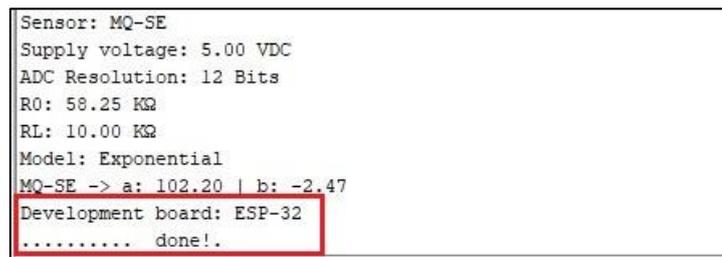


Fuente: elaborado por el autor

La autocalibración de los sensores, es decir cuando el nodo se enciende lo primero que realiza el nodo es la calibración de cada sensor conectado, en la Figura 92 se muestra la calibración automática de un nodo.

Figura 92

Auto calibración de sensores



Fuente: elaborado por el autor

En la Figura 93 se muestra la lectura de los datos obtenidos por los sensores cada 5 segundos, posteriormente se los guardara en variables y se procede a imprimir en el Serial Plotter del IDE ARDUINO, para validar la lectura y escritura de datos.

Figura 93

Impresión por serial plotter de valores obtenidos por sensores

```
COM4
Model: Exponential
MQ-SE -> a: 99.04 | b: -1.52
Development board: ESP-32
NH4:4.15 ppm    CO:0.63 ppm    Propano:2.39 ppm    TEMP:17 *C,    HUMEDAD R:52 H
iniciando SNMP Trap .....
Se envio SNMP Trap Sensor DOWN
NH4:4.61 ppm    CO:0.63 ppm    Propano:3.37 ppm    TEMP:17 *C,    HUMEDAD R:52 H
NH4:4.61 ppm    CO:0.65 ppm    Propano:3.33 ppm    TEMP:17 *C,    HUMEDAD R:52 H
NH4:4.59 ppm    CO:0.65 ppm    Propano:3.35 ppm    TEMP:17 *C,    HUMEDAD R:52 H
NH4:4.37 ppm    CO:0.65 ppm    Propano:3.34 ppm    TEMP:17 *C,    HUMEDAD R:52 H
NH4:4.38 ppm    CO:0.65 ppm    Propano:3.34 ppm    TEMP:17 *C,    HUMEDAD R:52 H
NH4:4.38 ppm    CO:0.67 ppm    Propano:3.38 ppm    TEMP:17 *C,    HUMEDAD R:52 H
NH4:4.38 ppm    CO:0.67 ppm    Propano:3.33 ppm    TEMP:17 *C,    HUMEDAD R:52 H
NH4:4.13 ppm    CO:0.67 ppm    Propano:3.42 ppm    TEMP:17 *C,    HUMEDAD R:52 H
NH4:4.15 ppm    CO:0.67 ppm    Propano:3.40 ppm    TEMP:17 *C,    HUMEDAD R:52 H
NH4:4.14 ppm    CO:0.65 ppm    Propano:3.40 ppm    TEMP:17 *C,    HUMEDAD R:52 H
```

Fuente: elaborado por el autor

En la Figura 94 se muestra la conexión exitosa del nodo y el servidor de base de datos MySQL.

Figura 94

Texto de confirmación de conexión entre nodo y el servidor MySQL

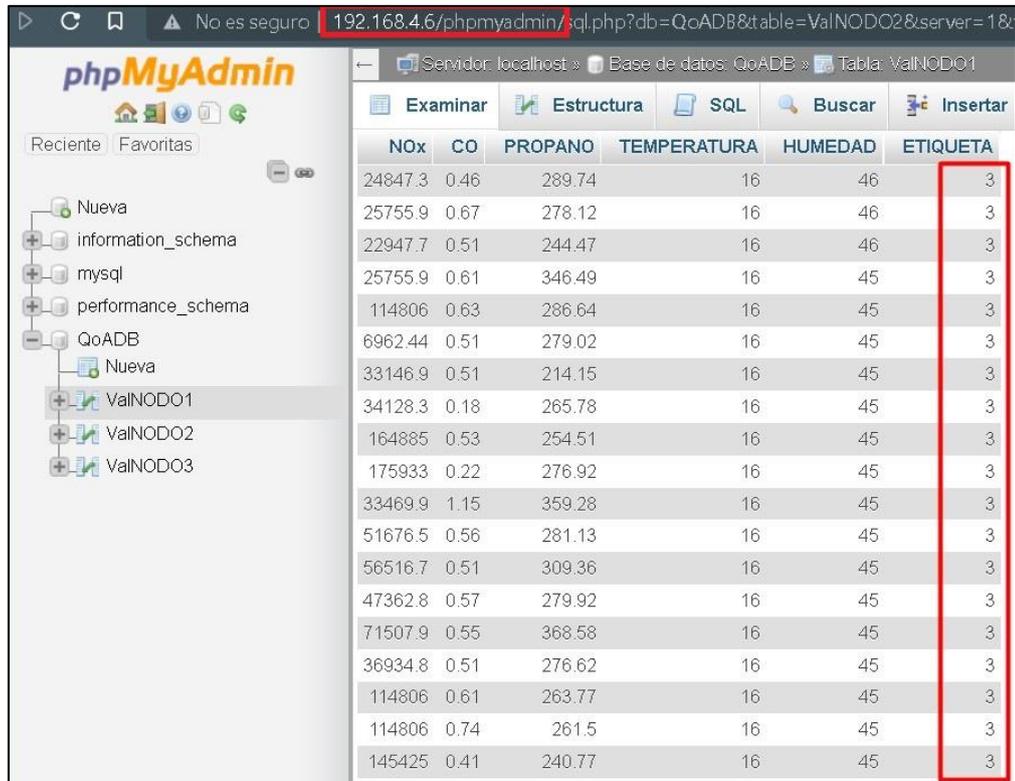
```
Connecting to WiFi .....Connected to SSID: GATEWAY-WSN
IP address: 192.168.4.3
Connecting to SQL... ..trying...
Connected to server version 5.6.51
OK.
Calibrando ..... done!
```

Fuente: elaborado por el autor

En la Figura 95 se muestra el correcto almacenamiento de los datos, accediendo a la base de datos MySQL mediante la dirección ip del servidor en un navegador web y utilizando la herramienta phpadmin utilizada para la administración de MySQL.

Figura 95

Verificación de datos guardados en base de datos MySQL



The screenshot shows the phpMyAdmin interface. The browser address bar contains the URL: 192.168.4.6/phpmyadmin/sql.php?db=QoADB&table=ValNODO2&server=1&... The interface shows a tree view on the left with databases like 'QoADB' and tables 'ValNODO1', 'ValNODO2', and 'ValNODO3'. The main area displays a table with the following data:

| NOx | CO | PROPANO | TEMPERATURA | HUMEDAD | ETIQUETA |
|---------|------|---------|-------------|---------|----------|
| 24847.3 | 0.46 | 289.74 | 16 | 46 | 3 |
| 25755.9 | 0.67 | 278.12 | 16 | 46 | 3 |
| 22947.7 | 0.51 | 244.47 | 16 | 46 | 3 |
| 25755.9 | 0.61 | 346.49 | 16 | 45 | 3 |
| 114806 | 0.63 | 286.64 | 16 | 45 | 3 |
| 6962.44 | 0.51 | 279.02 | 16 | 45 | 3 |
| 33146.9 | 0.51 | 214.15 | 16 | 45 | 3 |
| 34128.3 | 0.18 | 265.78 | 16 | 45 | 3 |
| 164885 | 0.53 | 254.51 | 16 | 45 | 3 |
| 175933 | 0.22 | 276.92 | 16 | 45 | 3 |
| 33469.9 | 1.15 | 359.28 | 16 | 45 | 3 |
| 51676.5 | 0.56 | 281.13 | 16 | 45 | 3 |
| 56516.7 | 0.51 | 309.36 | 16 | 45 | 3 |
| 47362.8 | 0.57 | 279.92 | 16 | 45 | 3 |
| 71507.9 | 0.55 | 368.58 | 16 | 45 | 3 |
| 36934.8 | 0.51 | 276.62 | 16 | 45 | 3 |
| 114806 | 0.61 | 263.77 | 16 | 45 | 3 |
| 114806 | 0.74 | 261.5 | 16 | 45 | 3 |
| 145425 | 0.41 | 240.77 | 16 | 45 | 3 |

Fuente: elaborado por el autor

4.1.3. Bloque De procesamiento y comunicación

Se valida el cumplimiento de los requerimientos de sistema y de arquitectura especificados para el bloque de procesamiento y comunicación mediante la aplicación de un test de rendimiento del algoritmo de clasificación y un test de comunicación, el cual permitirá realizar una evaluación de cumplimiento como se muestra en la Tabla 37. Con el propósito de verificar el cumplimiento de los requisitos de diseño e implementación del sistema.

Tabla 37

Tabla de verificación de cumplimiento de requerimientos para bloque de procesamiento y comunicación

| Tabla de validación de requerimientos sistema y arquitectura | | | | | | | | | | | | | | | | |
|---|--|--------------|--------------|--------------|--------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|------------------|---------------|---------------|
| Requerimientos | | SYSR2 | STSR4 | SYSR5 | SYSR9 | SYSR10 | SYSR11 | SYSR12 | SYSR13 | SRSH18 | SRSH19 | SRSH20 | SYSH28 | SYSH21-27 | SYSH29 | SYSH32 |
| SI | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Cumple | | | | | | | | | | | | | | | | |
| NO | | | | | | | | | | | | | | | | |

Fuente: elaborado por el autor

Al usar un Node MCU se tiene la ventaja de poseer un circuito con módulo de procesamiento y módulo de comunicación, los cuales brindan alto rendimiento al momento de procesamiento de datos y compatibilidad con bandas de frecuencia ISM y estándares de comunicación inalámbrica. Logrando así cumplir con los requerimientos que se requería para el diseño del sistema.

4.1.3.1. Test de rendimiento

Se procede a realizar un test precisión como se muestra en la Tabla 38, con la finalidad de la comprobación de rendimiento del algoritmo de clasificación supervisada implementado en el sistema embebido.

Tabla 38

Test de rendimiento

| Test de rendimiento | |
|----------------------------|--|
| Bloque de prueba | Bloque de procesamiento y comunicación |
| Descripción | Test de rendimiento. |
| Prerrequisitos: | |
| | ✓ Adquisición de Node MCU esp32 |
| | ✓ Diseño de placa PCB. |

-
- ✓ Bloque de alimentación.
 - ✓ Bloque de adquisición de datos.
 - ✓ Algoritmo de clasificación supervisada KNN
 - ✓ Matriz de entrenamiento

Pasos:

- ✓ Montaje de bloque de alimentación, adquisición de datos, procesamiento y comunicación.
- ✓ Cargar sketch de Arduino con el código del algoritmo de clasificación y matriz de entrenamiento.
- ✓ Cargar los valores obtenidos por el bloque de adquisición de datos a un array para el algoritmo de clasificación.
- ✓ Aplicar clasificación supervisada a los valores adquiridos.
- ✓ Crear matriz de confusión.
- ✓ Calcular la precisión
- ✓ Calcular la tasa de error
- ✓ Calcular la exactitud

Resultados esperados:

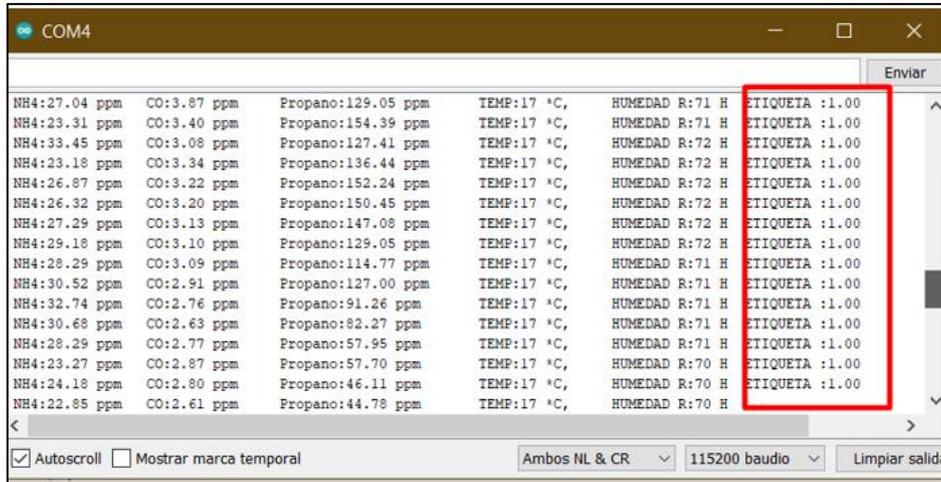
Acoplamiento correcto con todos los elementos del sistema embebido, correcto rendimiento del algoritmo de clasificación.

Fuente: elaborado por el autor

Resultados obtenidos: correcto acoplamiento entre los sensores y la placa PCB, autocalibración de sensores y lectura de datos, niveles de precisión, exactitud y tasa de errores aceptable en la Figura 96 se muestra la impresión correcta de la etiqueta en el serial plotter de IDE ARDUINO.

Figura 96

Impresión correcta de la etiqueta correspondiente al resultado del algoritmo KNN



Fuente: elaborado por el autor

Con el correcto funcionamiento del bloque de adquisición de datos y el microcontrolador se procede a verificar el correcto rendimiento del algoritmo de clasificación supervisada.

Matriz de confucion

En la Tabla 39 se muestra la matriz de confusion con la cual se procede a calcular el desempeño de el algoritmo de clasficacion supervisada KNN con K vecinos igual a 3.

Tabla 39

Matriz de confusión

| | 1 | 2 | 3 |
|----------|----------|----------|----------|
| 1 | 142 | 0 | 0 |
| 2 | 1 | 61 | 3 |
| 3 | 0 | 0 | 47 |

Fuente: elaborado por el autor

Con los valores dados por la matriz de Confusión evaluada en el software RStudio y la Ecuación 5 se procede a verificar la exactitud, donde se puede decir que el algoritmo implementado es tiene una exactitud de clasificación de 98.4%.

$$Exactitud = \frac{142 + 61 + 47}{142 + 61 + 47 + 3 + 1}$$

$$Exactitud = 0.984 * 100$$

$$Exactitud = 98.4\%$$

Utilizando la Ecuación 6 se puede determinar que el algoritmo tiene un 24.0% de datos clasificados incorrectamente

$$Tasa de error = \frac{0 + 61 + 0}{142 + 61 + 47 + 3 + 1}$$

$$Tasa de error = 0.24015 * 100$$

$$Tasa de error = 24.015\%$$

Utilizando la Ecuación 8 se puede determinar que el algoritmo utilizado posee una probabilidad de precisión del 94% es decir que de los valores que datos por la matriz de entrenamiento existe una gran probabilidad que los valores encontrados y clasificados sean correctos.

$$Presición = \frac{47}{47 + 3}$$

$$Presición = 0.94 * 100$$

$$Presición = 94\%$$

La precisión es una característica esencial en el diseño de un sistema IoT, debido a que en la mayoría de los nodos están constituidos por sensores que miden las propiedades del mundo físico. Precisión de estos sensores es la proximidad de la concordancia entre los valores medidos y los valores reales de esas propiedades, los servicios de software que son implementados para hacer cálculos con los datos de entrada como el algoritmo de clasificación supervisada debe poseer una precisión expresada en

porcentaje. (ISO/IEC 30141, 2018) Para medir la precisión de clasificación del sistema se ha tomado varias muestras de datos y se calculó la precisión de clasificación del algoritmo implementado, en la Tabla 40 se muestra los resultados obtenidos.

Tabla 40

Accuracy

| Semilla | Número de vecinos óptimo para algoritmo KNN | | |
|-------------|---|-------|-------|
| | K=1 | K=3 | K=5 |
| 5008 | 98.82 | 99.21 | 98.43 |
| 1504 | 95.92 | 98.98 | 97.96 |

Fuente: elaborado por el autor

Como resultados se obtuvo que existe una mejor precisión implementado un valor de K vecinos igual a 3, de igual manera que a un valor más alto de semilla la precisión aumenta. Verificando una aceptable precisión de clasificación para el reconocimiento de material particulado en el aire.

4.1.3.2. Test de comunicación

Se procede a realizar un test comunicación como se muestra en la Tabla 41, con la finalidad de la comprobación el envío y recepción de los datos, bit rate y paquetes SNMP enviados desde nodos a estación base.

Tabla 41

Test de comunicación

| Test de comunicación | |
|-------------------------|--|
| Bloque de prueba | Bloque de procesamiento y comunicación |
| Descripción | Test de comunicación. |
| Prerrequisitos: | |
| | ✓ Node MCU esp32 |

-
- ✓ Node MCU esp8266 modo Gateway
 - ✓ Estación de gestión activa.
 - ✓ Diseño de placa PCB.
 - ✓ Sistema de alimentación.
 - ✓ Bloque de adquisición de datos.
 - ✓ Bloque de procesamiento

Pasos:

- ✓ Cargar sketch de Arduino con el código de habilitación de módulo de comunicación inalámbrico.
- ✓ Autenticación con Gateway
- ✓ Verificación de bit rate.
- ✓ Verificación de envío y recepción de paquetes SNMP.
- ✓ Análisis de la red.

Resultados esperados:

Módulo de comunicación trabajando correctamente bajo el protocolo IEEE 802.11g, latencia y bit rate aceptable, envío y recepción de paquetes y evitar la pérdida de paquetes.

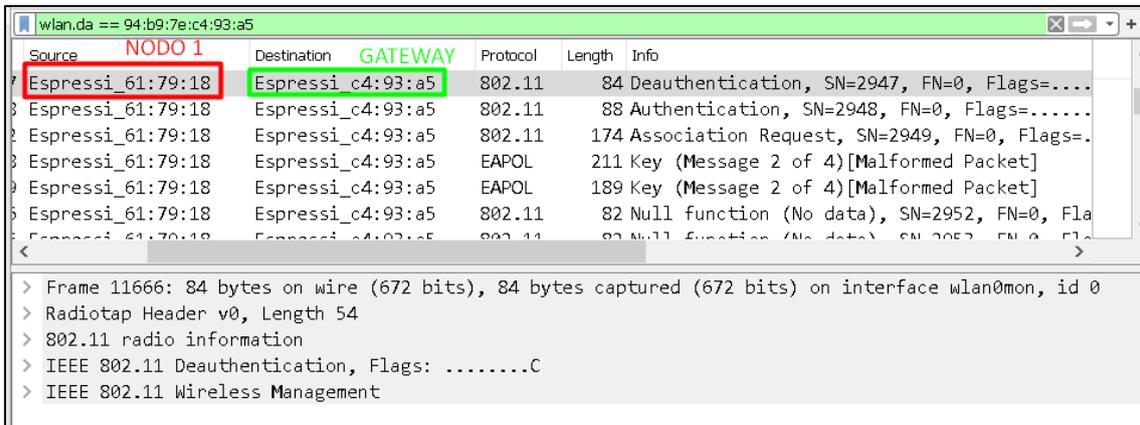
Fuente: elaborado por el autor

Resultados obtenidos: autenticación y asociación correcto entre todos los elementos de la red WSN, latencia aceptable < 200 ms, un máximo de 1400 bytes a una distancia de 50 metros, autenticación y asociación exitosa con el Gateway, correcto envío y recepción de paquetes.

Para verificar la autenticación y asociación de los elementos de red, se procedió a utilizar un sniffer y proceder a capturar las tramas que se emiten desde los nodos hacia el Gateway, en este caso se capturo la deautenticacion, autenticación y asociación entre nodo 1 (Transmitter address: Espressi_61:79:18 (7c:9e:bd:61:79:18)) y el Gateway (Destination address: Espressi_c4:93:a5 (94:b9:7e:c4:93:a5)). En la Figura 97 se muestra las tramas capturadas.

Figura 97

Tramas de autenticación y asociación

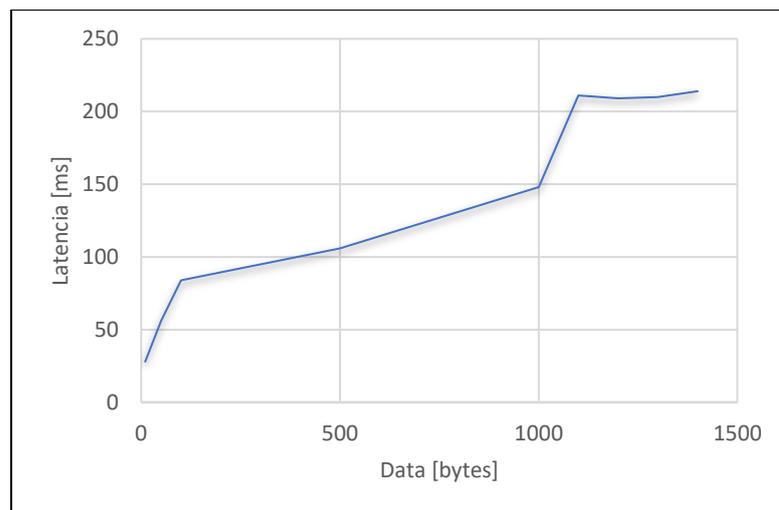


Fuente: elaborado por el autor

Para verificar el bit rate se realizó mediante el protocolo ICMP para medir la latencia que existe entre el nodo y la estación de gestión, además se utilizó diferentes tamaños de paquetes en bytes variable. En la Figura 98 se muestra como a mayor cantidad de bytes mayor es la latencia, esta prueba se la realizo en campo abierto.

Figura 98

Relación entre tamaño de paquete [bytes] y latencia[ms]



Fuente: elaborado por el autor

En la Figura 99 se muestra el comando utilizado para variar la cantidad de bytes en el mensaje ICMP, para verificar el tiempo estimado de ida y retorno del paquete. En

este caso se asignó la dirección IPv4 de los diferentes nodos y utilizando el comando -l se modifica y se varió el tamaño del paquete enviado en bytes, hasta llegar a al límite soportado por los nodos, en este caso en los 1500 bytes existe una pérdida de paquetes del 100%.

Figura 99

Protocolo ICMP para verificación de latencia y bit rate.

```
C:\Users\Wilmer Benavides>ping -l 1450 192.168.4.2
Haciendo ping a 192.168.4.2 con 1450 bytes de datos:
Respuesta desde 192.168.4.2: bytes=1450 tiempo=181ms TTL=255
Respuesta desde 192.168.4.2: bytes=1450 tiempo=202ms TTL=255
Respuesta desde 192.168.4.2: bytes=1450 tiempo=216ms TTL=255
Respuesta desde 192.168.4.2: bytes=1450 tiempo=25ms TTL=255

Estadísticas de ping para 192.168.4.2:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 25ms, Máximo = 216ms, Media = 156ms

C:\Users\Wilmer Benavides>ping -l 50 192.168.4.4
Haciendo ping a 192.168.4.4 con 50 bytes de datos:
Respuesta desde 192.168.4.4: bytes=50 tiempo=113ms TTL=255
Respuesta desde 192.168.4.4: bytes=50 tiempo=132ms TTL=255
Respuesta desde 192.168.4.4: bytes=50 tiempo=147ms TTL=255
Respuesta desde 192.168.4.4: bytes=50 tiempo=163ms TTL=255

Estadísticas de ping para 192.168.4.4:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 113ms, Máximo = 163ms, Media = 138ms

C:\Users\Wilmer Benavides>ping -l 1500 192.168.4.3
Haciendo ping a 192.168.4.3 con 1500 bytes de datos:
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.

Estadísticas de ping para 192.168.4.3:
    Paquetes: enviados = 2, recibidos = 0, perdidos = 2
    (100% perdidos),
```

Fuente: elaborado por el autor

Para verificar que le agente SNMP esté trabajando correctamente se procede a capturar los paquetes enviados desde el Nodo 1 y que ingresan a la estación de gestión mediante el software Wireshark, en la Figura 100 se muestra los paquetes capturados,

donde se puede apreciar la dirección IP origen y destino, protocolo, tipo de mensaje SNMP y el OID del objeto.

Figura 100

Capturas de paquetes en estación de gestión

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------|-------------|-------------|----------|--------|------------------------------------|
| 131 | 40.668628 | 192.168.4.3 | 192.168.4.2 | SNMP | 85 | get-request 1.3.6.1.3.1.1.2.3.3.0 |
| 132 | 41.076583 | 192.168.4.2 | 192.168.4.3 | SNMP | 100 | get-response 1.3.6.1.3.1.1.2.3.3.0 |
| 133 | 41.077517 | 192.168.4.3 | 192.168.4.2 | SNMP | 83 | get-request 1.3.6.1.2.1.1.3.0 |
| 144 | 42.220609 | 192.168.4.2 | 192.168.4.3 | SNMP | 98 | get-response 1.3.6.1.2.1.1.3.0 |
| 150 | 43.669743 | 192.168.4.3 | 192.168.4.2 | SNMP | 85 | get-request 1.3.6.1.3.1.1.2.4.3.0 |
| 166 | 44.505709 | 192.168.4.2 | 192.168.4.3 | SNMP | 97 | get-response 1.3.6.1.3.1.1.2.4.3.0 |
| 186 | 46.669684 | 192.168.4.3 | 192.168.4.2 | SNMP | 85 | get-request 1.3.6.1.3.1.1.2.1.4.0 |
| 190 | 46.788292 | 192.168.4.2 | 192.168.4.3 | SNMP | 114 | get-response 1.3.6.1.3.1.1.2.1.4.0 |
| 207 | 49.667989 | 192.168.4.3 | 192.168.4.2 | SNMP | 85 | get-request 1.3.6.1.3.1.1.2.3.4.0 |
| 208 | 49.668889 | 192.168.4.3 | 192.168.4.1 | SNMP | 84 | get-request 1.3.6.1.2.1.25.1.1.0 |
| 211 | 49.673570 | 192.168.4.1 | 192.168.4.3 | SNMP | 95 | get-response 1.3.6.1.2.1.25.1.1.0 |
| 212 | 49.674272 | 192.168.4.3 | 192.168.4.1 | SNMP | 83 | get-request 1.3.6.1.2.1.1.3.0 |
| 213 | 49.679777 | 192.168.4.2 | 192.168.4.3 | SNMP | 98 | get-response 1.3.6.1.2.1.1.3.0 |

Fuente: elaborado por el autor

En la Figura 101 se muestra la estructura de un mensaje GET-REQUEST encapsulada en un datagrama UDP enviado desde el Nodo 1 a la estación de gestión, donde se puede apreciar la versión del protocolo SNMP, la comunidad y el OID en capa de aplicación, en capa transporte se puede apreciar el protocolo UDP donde se encuentran el puerto origen y destino, en capa de internet se encuentran la versión de protocolo IP usado, direcciones IPv4 de origen y destino, en capa de acceso a la red se encuentran las direcciones MAC de origen y destino.

Figura 101

Estructura de un mensaje get-requets encapsulado en un datagrama UDP.

```
131 40.668628 192.168.4.3 192.168.4.2 SNMP 85 get-request 1.3.6.1.3.1.1.2.3.3.0
132 41.076583 192.168.4.2 192.168.4.3 SNMP 100 get-response 1.3.6.1.3.1.1.2.3.3.0

<
> Frame 131: 85 bytes on wire (680 bits), 85 bytes captured (680 bits) on interface \Device\NPF {65805709-B87F-46A2-AB4E-366225ACE073}, id 0
> Ethernet II, Src: CyberTAN b3:f7:51 (60:14:b3:b3:f7:51), Dst: Espressi c4:93:a4 (94:b9:7e:c4:93:a4)
  > Internet Protocol Version 4, Src: 192.168.4.3, Dst: 192.168.4.2
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      Total Length: 71
      Identification: 0x02ab (683)
    > Flags: 0x00
      ...0 0000 0000 0000 = Fragment Offset: 0
      Time to Live: 128
      Protocol: UDP (17)
      Header Checksum: 0xaea5 [validation disabled]
      [Header checksum status: Unverified]
      Source Address: 192.168.4.3
      Destination Address: 192.168.4.2
  > User Datagram Protocol, Src Port: 52593, Dst Port: 161
    > Simple Network Management Protocol
      version: v2c (1)
      community: public
      > data: get-request (0)
        > get-request
          request-id: 8001
          error-status: noError (0)
          error-index: 0
          > variable-bindings: 1 item
            > 1.3.6.1.3.1.1.2.3.3.0: Value (Null)
[Response In: 132]
```

Fuente: elaborado por el autor

Análisis de la red

Para la verificación del bloque de comunicación se procede a realizar un análisis del tráfico enviado desde los nodos a la estación base, para este aparatado se realizara un análisis de los tamaños de las cabeceras que posee cada uno de los protocolos de red utilizados, para determinar y verificar el tamaño del paquete y el tiempo en el medio, en este caso el medio aire ya que se trabaja con una red Wireless, a continuación, se analiza cada uno de las capas en relación a la arquitectura tcp/ip.

Cabecera SNMP

En base a el fundamento teórico visto en el capítulo II, se tiene diferentes capas según el modelo de referencia OSI, donde para las capas aplicación, presentación y sesión se tiene el protocolo SNMP, el cual posee tres cabeceras fundamentales:

- ✓ **Versión**, con un tamaño fijo de 4 bytes.
- ✓ **Comunidad**, con un tamaño variable máximo de 255 octetos.

✓ **PDU**, con un valor variable máximo de 484 octetos.

Para este caso se ha capturado un mensaje GET-RESPONSE, en la Figura 102 se muestra las diferentes cabeceras SNMP.

Figura 102

Mensaje SNMP get-response

| Source | Destination | Protocol | Length | Info |
|-------------|-------------|----------|--------|------------------------------------|
| 192.168.4.2 | 192.168.4.5 | SNMP | 109 | get-response 1.3.6.1.2.1.1.6.0 |
| 192.168.4.5 | 192.168.4.2 | SNMP | 85 | get-request 1.3.6.1.3.1.1.2.3.1.0 |
| 192.168.4.2 | 192.168.4.5 | SNMP | 100 | get-response 1.3.6.1.3.1.1.2.3.1.0 |
| 192.168.4.5 | 192.168.4.2 | SNMP | 83 | get-request 1.3.6.1.2.1.1.3.0 |
| 192.168.4.2 | 192.168.4.5 | SNMP | 98 | get-response 1.3.6.1.2.1.1.3.0 |


```

> User Datagram Protocol, Src Port: 161, Dst Port: 53292
  Simple Network Management Protocol
    version: v2c (1)
    community: public
    data: get-response (2)
      get-response
        request-id: 29626
        error-status: noError (0)
        error-index: 0
        variable-bindings: 1 item
          1.3.6.1.3.1.1.2.3.1.0: 33
            Object Name: 1.3.6.1.3.1.1.2.3.1.0 (iso.3.6.1.3.1.1.2.3.1.0)
            Value (Integer32): 33
  
```

Fuente: elaborado por el autor

A continuación, en la Tabla 42 se realiza el cálculo del tamaño en bytes de las diferentes cabeceras y payload.

Tabla 42

Cálculo del tamaño total de datos

| Campo | Tamaño (bytes) |
|--------------|-----------------------|
| Versión | 4 |
| Comunidad | 6 |
| PDU | 59 |
| Total | 69 |

Fuente: elaborado por el autor

Como se puede observar en la Figura 103 y la tabla 49 el tamaño de datos la capa aplicación es este caso bajo el protocolo SNMP es de 69 bytes.

Cabecera UDP

Para la capa de transporte que trabaja con el protocolo UDP, el cual posee una cabecera de fija 8 bytes, que en su estructura consta de los siguientes campos:

- ✓ Puerto origen, con un tamaño de 2 bytes
- ✓ Puerto destino, con un tamaño de 2 bytes
- ✓ Longitud UDP, con un tamaño de 2 bytes
- ✓ Checksum, con un tamaño de 2 bytes.
- ✓ Payload, con un tamaño variable.

En la Figura 103 se muestra la estructura de un segmento UDP, donde se puede apreciar el puerto por el cual se realizará la comunicación entre nodo y estación base, el puerto 161 el cual utiliza el protocolo SNMP por defecto.

Figura 103

Cabeceras UDP

| | | | | | |
|-------------|-------------|------|-----|--------------|-----------------------|
| 192.168.4.5 | 192.168.4.2 | SNMP | 85 | get-request | 1.3.6.1.3.1.1.2.3.1.0 |
| 192.168.4.2 | 192.168.4.5 | SNMP | 100 | get-response | 1.3.6.1.3.1.1.2.3.1.0 |
| 192.168.4.5 | 192.168.4.2 | SNMP | 83 | get-request | 1.3.6.1.2.1.1.3.0 |
| 192.168.4.2 | 192.168.4.5 | SNMP | 98 | get-response | 1.3.6.1.2.1.1.3.0 |
| 192.168.4.5 | 192.168.4.2 | SNMP | 85 | get-request | 1.3.6.1.3.1.1.2.3.1.0 |

<

> Internet Protocol Version 4, Src: 192.168.4.1 (192.168.4.1), Dst: 192.168.4.5 (192.168.4.5)

✓ User Datagram Protocol, Src Port: 161, Dst Port: 62197

Source Port: 161

Destination Port: 62197

Length: 76

Checksum: 0xdc9a [unverified]
[Checksum Status: Unverified]
[Stream index: 129]

> [Timestamps]

UDP payload (69 bytes)

✓ Simple Network Management Protocol

Fuente: elaborado por el autor

Tomando el valor de cada campo se procede a realizar el cálculo del tamaño del segmento UDP como se muestra en la Tabla 43.

Tabla 43

Calculo cabecera UDP

| Campo | Tamaño (bytes) |
|--------------|-----------------------|
| Cabecera UDP | 8 |
| UDP payload | 69 |
| Total | 77 |

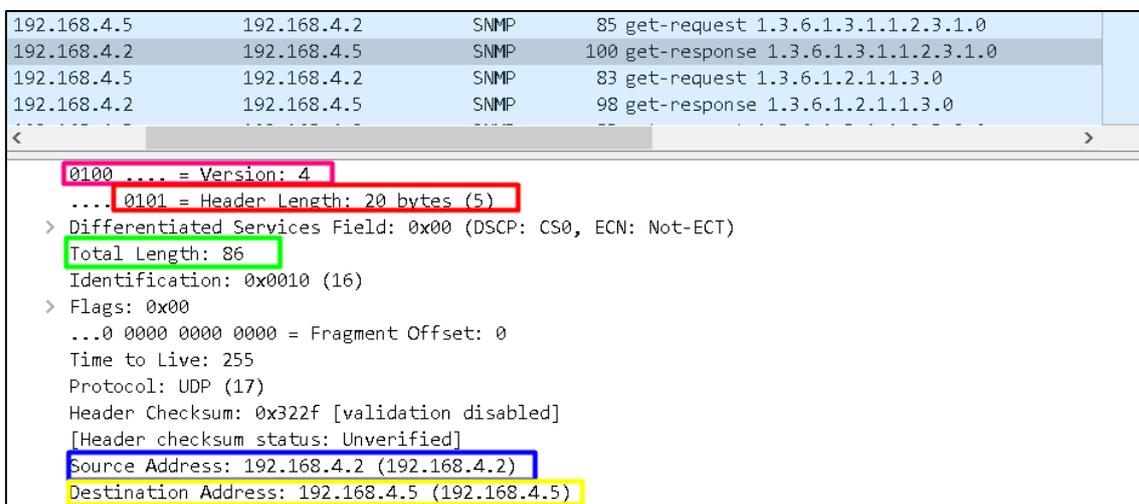
Fuente: elaborado por el autor

Cabecera IPv4

La capa de red trabaja con el protocolo de internet, donde se encuentra la cabecera ipv4 posee un tamaño de 20 bytes no fijos debido a que si se utilizan los campos “opciones” puede tener un tamaño máximo de 60 bytes, en el cual se puede apreciar los campos como versión de protocolo, tamaño total de cabecera, dirección ip origen y dirección ip destino entre otros campos como se muestra en la Figura 104.

Figura 104

Cabecera IPv4



Fuente: elaborado por el autor

En la Tabla 44 se muestra el cálculo del tamaño de la cabecera IPv4, tomando los valores en bytes de la cabecera IPv4 y payload.

Tabla 44

Calculo cabecera ipv4

| Campo | Tamaño (bytes) |
|---------------|-----------------------|
| Cabecera IPv4 | 20 |
| Payload | 77 |
| Total | 97 |

Fuente: elaborado por el autor

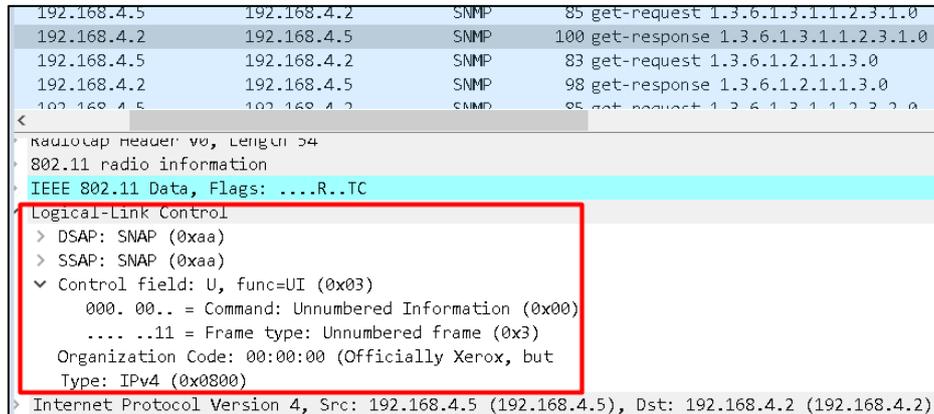
Cabeceras IEEE 802.11g

El protocolo 802.11g trabaja a nivel de capa enlace de datos que a su vez se divide en dos subcapas (subcapa LLC, subcapa MAC) y capa física que a su vez se divide en dos subcapas (la subcapa PLCP, subcapa PMD).

La subcapa LLC es la subcapa que se encarga de control del enlace lógico y flujo de los datos, donde se especifican los SNAP que son una extensión del protocolo LLC para lograr distinguir un mayor número de protocolos de capas superiores, está compuesta por un tamaño de 8 bytes. En la Figura 105 se muestra la estructura de la cabecera LLC.

Figura 105

Cabecera LLC

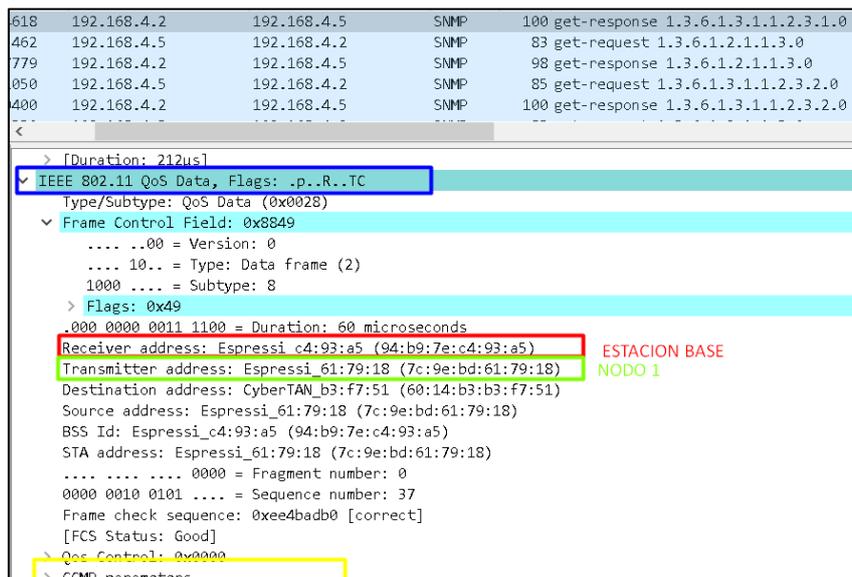


Fuente: elaborado por el autor

En la cabecera MAC es donde se agregan las direcciones MAC del transmisor, receptor, destinatario, origen y la dirección de BSSID entre otros campos, con un tamaño de 30 bytes, además se agrega una cabecera donde se especifican parámetros del protocolo de autenticación de mensaje que pertenece a la encriptación de los datos. En la Figura 106 se muestra la estructura de la cabecera MAC.

Figura 106

Cabecera MAC.



Fuente: elaborado por el autor

A continuación, en la Tabla 47 se muestra el cálculo del tamaño en bytes del MPDU, tomando en cuenta el tamaño del PSDU, la cabecera LLC y MAC.

Tabla 45

Cálculo del tamaño total MPDU

| Campo | Tamaño (bytes) |
|-------------------|-----------------------|
| Cabecera LLC | 8 |
| Cabecera MAC | 30 |
| FCS | 4 |
| PSDU | 97 |
| TOTAL MPDU | 139 |

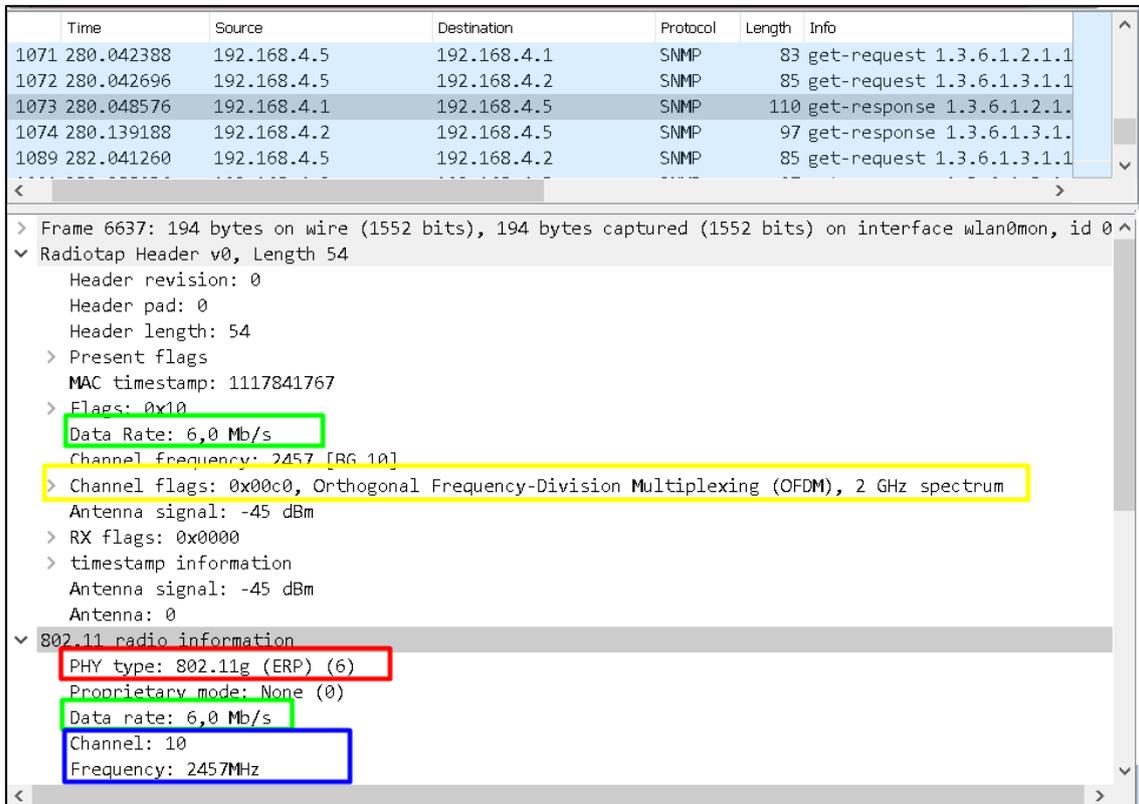
Fuente: elaborado por el autor

Cabecera PLCP

La cabecera PLCP posee un tamaño de 40 bits y está compuesto por los campos Señal, Servicio, longitud y una comprobación CRC de la capa PLCP como se muestra en la Figura 107.

Figura 107

Cabecera PLCP



Fuente: elaborado por el autor

A continuación, en la Tabla 48 se muestra el cálculo total del tamaño del PPDU, tomando el valor de la cabecera PLCP y valor del campo Tail con un tamaño de 6 bits.

Tabla 46

Calculo tamaño PPDU

| Campo | Tamaño (bytes) |
|----------------------|-----------------------|
| MPDU | 138 |
| Tail | 0.75 |
| Cabecera PLCP | 5 |
| Total PPDU | 144.75 |

Fuente: elaborado por el autor

Cálculo de tiempo en el aire

Para el cálculo de tiempo en el aire hay que transformar a bits el valor total del PPDU.

$$\#Total\ bits = 144.75 * 8$$

$$\#Total\ bits = 1158$$

Para 802.11g con un bit rate de 6 Mb/s utiliza una modulación BPSK, una tasa de codificación de $\frac{1}{2}$ y 24 bits de datos por símbolo OFDM.

$$\#Simbolos_OFDM = 1158/24$$

$$\#Simbolos_OFDM \cong 49$$

Sabiendo que la duración de cada símbolo OFDM es de $4\ \mu s$, y sabiendo que la duración para el preámbulo ERP_OFDM es de $16\ \mu s$, se puede calcular el tiempo de la trama en el aire.

$$AirTime = (4 * 49) + 16$$

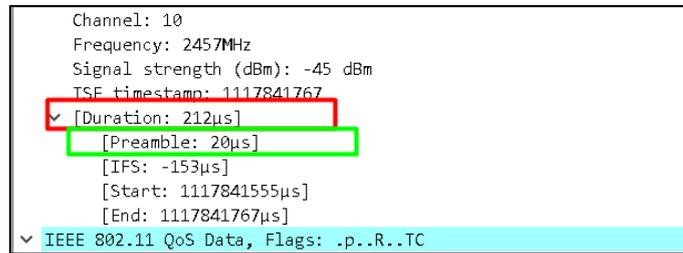
$$AirTime = 196 + 16$$

$$AirTime = 212\ \mu s$$

En la Figura 108 se muestra la duración de $212\ \mu s$ que coincide con la trama capturada por el sniffer.

Figura 108

Duración de la trama en el aire



Fuente: elaborado por el autor

4.1.4. Bloque De Recepción Y Retransmisión

Se valida el cumplimiento de los requerimientos de sistema y de arquitectura especificados para el bloque de recepción y retransmisión mediante la aplicación de un test de comunicación el cual ayudara a realizar la evaluación de cumplimiento como se muestra en la Tabla 47. Con el propósito de verificar el cumplimiento de los requisitos en el diseño e implementación del sistema.

Tabla 47

Tabla de verificación de cumplimiento de requerimientos para bloque de adquisición y retransmisión

| Tabla de validación de requerimientos sistema y arquitectura | | | | | | | | | | | | |
|--|-------|-------|-------|-------|--------|--------|--------|--------|-------|--------|--------|--------|
| Requerimientos | STSR2 | STSR5 | SYSR3 | SYSR4 | SYSR10 | SYSR11 | SYSR13 | SYSR14 | SRSH7 | SRSH12 | SRSH14 | SYSH28 |
| SI | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ |
| Cumple | | | | | | | | | ✓ | | | |
| NO | | | | | | | | | | | | |

Fuente: elaborado por el autor

El Gateway al poseer recursos necesarios para el buen funcionamiento, logra cumplir la mayoría de los requerimientos, pero no obstante al no trabajar protocolos Low power, el consumo de energía es mayor y la recarga de las baterías se debe realizar más constantemente.

4.1.4.1. Test de comunicación

Se procede a realizar un test comunicación como se muestra en la Tabla 48, con la finalidad de la comprobación el envío, recepción, retransmisión de los datos y paquetes SNMP enviados desde Gateway a la estación base.

Tabla 48

Test de comunicación

| Test de comunicación | |
|--|-------------------------------------|
| Bloque de prueba | Bloque de recepción y retransmisión |
| Descripción | Test de comunicación. |
| Prerrequisitos: | |
| <ul style="list-style-type: none">✓ Node MCU esp8266 modo Gateway✓ Estación de gestión activa.✓ Diseño de placa PCB.✓ Sistema de alimentación.✓ Bloque de adquisición de datos. | |
| Pasos: | |
| <ul style="list-style-type: none">✓ Cargar sketch de Arduino con el código de habilitación de módulo de comunicación inalámbrico.✓ Verificación de SSID activo✓ Verificar autenticación y asociación de nodos y estación base al Gateway.✓ Verificar recepción y retransmisión de datos entre nodos y estación base.✓ Envío y recepción de paquetes. | |
| Resultados esperados: | |
| Gateway proporciona una correcta autenticación y asociación de los nodos, además de la recepción y retransmisión de datos correctamente y el envío de la información requerida por el gestor SNMP. | |

Fuente: elaborado por el autor

Resultados obtenidos: el Gateway muestra el SSID de la red denominada “GATEWAY WSN”, además de proporcionar 8 estaciones conectadas y enviando datos al mismo tiempo, el envío y recepción de datos por parte de nodos y estación base

aceptable, además de las correspondientes notificaciones SNMP por parte del Gateway a la estación de gestión correctos.

En la Figura 109 se muestra el resultado de la captura del tráfico al realizar el envío de datos desde una estación de la red hacia la estación base, donde las tramas son reenviadas desde la estación al Gateway y después retransmitidas hacia la estación base, donde se puede apreciar al SSID y protocolo de comunicación inalámbrica 802.11g, el canal y el BSSID correspondiente al Gateway.

Figura 109

Captura de tráfico entre nodo, Gateway y estación base

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-------------|-------------------|-------------------|----------|--------|--|
| 4 | 0.020574380 | ee:fa:bc:08:04:d7 | Broadcast | 802.11 | 137 | Beacon frame, SN=1431, FN=0, Flags=....., BI=100, SSID=GATEWAY-WSN |
| 60 | 0.123252131 | ee:fa:bc:08:04:d7 | Broadcast | 802.11 | 137 | Beacon frame, SN=1432, FN=0, Flags=....., BI=100, SSID=GATEWAY-WSN |
| 80 | 0.225388852 | ee:fa:bc:08:04:d7 | Broadcast | 802.11 | 137 | Beacon frame, SN=1433, FN=0, Flags=....., BI=100, SSID=GATEWAY-WSN |
| 84 | 0.327669616 | ee:fa:bc:08:04:d7 | Broadcast | 802.11 | 137 | Beacon frame, SN=1434, FN=0, Flags=....., BI=100, SSID=GATEWAY-WSN |
| 88 | 0.429944419 | ee:fa:bc:08:04:d7 | Broadcast | 802.11 | 137 | Beacon frame, SN=1435, FN=0, Flags=....., BI=100, SSID=GATEWAY-WSN |
| 102 | 0.532273623 | ee:fa:bc:08:04:d7 | Broadcast | 802.11 | 137 | Beacon frame, SN=1436, FN=0, Flags=....., BI=100, SSID=GATEWAY-WSN |
| 118 | 0.635079123 | ee:fa:bc:08:04:d7 | Broadcast | 802.11 | 137 | Beacon frame, SN=1437, FN=0, Flags=....., BI=100, SSID=GATEWAY-WSN |
| 134 | 0.725018533 | ee:fa:bc:08:04:d7 | CyberTAN_b3:f7:51 | 802.11 | 122 | Data, SN=904, FN=0, Flags=.p...F. |
| 136 | 0.725483191 | ee:fa:bc:08:04:d7 | CyberTAN_b3:f7:51 | 802.11 | 122 | Data, SN=905, FN=0, Flags=.p...R.F. |
| 138 | 0.736821518 | ee:fa:bc:08:04:d7 | Broadcast | 802.11 | 137 | Beacon frame, SN=1438, FN=0, Flags=....., BI=100, SSID=GATEWAY-WSN |
| 149 | 0.840230430 | ee:fa:bc:08:04:d7 | Broadcast | 802.11 | 137 | Beacon frame, SN=1439, FN=0, Flags=....., BI=100, SSID=GATEWAY-WSN |
| 162 | 0.942460265 | ee:fa:bc:08:04:d7 | Broadcast | 802.11 | 137 | Beacon frame, SN=1440, FN=0, Flags=....., BI=100, SSID=GATEWAY-WSN |
| 170 | 1.044982258 | ee:fa:bc:08:04:d7 | Broadcast | 802.11 | 137 | Beacon frame, SN=1441, FN=0, Flags=....., BI=100, SSID=GATEWAY-WSN |
| 188 | 1.147048308 | ee:fa:bc:08:04:d7 | Broadcast | 802.11 | 137 | Beacon frame, SN=1442, FN=0, Flags=....., BI=100, SSID=GATEWAY-WSN |


```

> Frame 134: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface wlan0mon, id 0
> Radiotap Header v0, Length 18
> 802.11 radio information
  > PHY type: 802.11g (ERP) (6)
  > Proprietary mode: None (0)
  > Data rate: 54.0 Mb/s
  > Channel: 11
  > Frequency: 2462MHz
  > Signal strength (dBm): -59 dBm
  > [Duration: 36us]
  > IEEE 802.11 Data, Flags: .p...F.
    > Type/Subtype: Data (0x0020)
    > Frame Control Field: 0x0842
      > .000 0000 0011 1100 = Duration: 60 microseconds
      > Receiver address: CyberTAN_b3:f7:51 (60:14:b3:b3:f7:51)
      > Transmitter address: ee:fa:bc:08:04:d7 (ee:fa:bc:08:04:d7)
      > Destination address: CyberTAN_b3:f7:51 (60:14:b3:b3:f7:51)
      > Source address: ee:fa:bc:08:04:d7 (ee:fa:bc:08:04:d7)
      > BSS Id: ee:fa:bc:08:04:d7 (ee:fa:bc:08:04:d7)
      > STA address: CyberTAN_b3:f7:51 (60:14:b3:b3:f7:51)
      > ..... 0000 = Fragment number: 0
      > 0011 1000 1000 .... = Sequence number: 904
  
```

Fuente: elaborado por el autor

Con el uso de sistema operativo Linux y el uso de un software capturador de paquetes Wireshark se procedió a configurar el módulo de comunicación inalámbrica del ordenador en modo monitor para escuchar todos los paquetes que se comparten entre el Gateway y la estación base. En la Figura 110 se muestra los mensajes PROBE REQUEST

usada por el nodo para una sonda de prueba y PROBE RESPONSE que le responde el Gateway para indicar parámetros para la comunicación.

Figura 110

Scanning activo entre Nodo y Gateway

| | | | | | |
|------|--------------|-------------------------|-------------------------|---------------------------------|--|
| 1740 | 13.824673651 | ee:47:20:86:5e:65 | zte_db:b6:a3 | 802.11 | 42 Null function (No data), SN=637, FN=0, Flags=.....T |
| 1741 | 13.824916357 | MAC: ESTACION BASE | ee:47:20:86:5e:65 (...) | 802.11 | 28 Acknowledgement, Flags=..... |
| 1742 | 13.826079051 | CyberTAN_b3:f7:51 | Broadcast | 802.11 | 127 Probe Request, SN=1179, FN=0, Flags=...P...., SSID=cat-passpoint-profile |
| 1743 | 13.826423855 | CyberTAN_b3:f7:51 | CyberTAN_b3:f7:51 (...) | 802.11 | 28 Acknowledgement, Flags=..... |
| 1744 | 13.828082909 | CyberTAN_b3:f7:51 | Broadcast | 802.11 | 127 Probe Request, SN=1180, FN=0, Flags=...P...., SSID=cat-passpoint-profile |
| 1745 | 13.829300993 | CyberTAN_b3:f7:51 | Broadcast | 802.11 | 113 Probe Request, SN=1181, FN=0, Flags=...P...., SSID=eduroam |
| 1746 | 13.829747758 | CyberTAN_b3:f7:51 | Broadcast | 802.11 | 113 Probe Request, SN=1182, FN=0, Flags=...P...., SSID=eduroam |
| 1747 | 13.830019833 | CyberTAN_b3:f7:51 | CyberTAN_b3:f7:51 (...) | 802.11 | 28 Acknowledgement, Flags=..... |
| 1748 | 13.831129799 | CyberTAN_b3:f7:51 | Broadcast | 802.11 | 106 Probe Request, SN=1183, FN=0, Flags=...P...., SSID=wildcard (Broadcast) |
| 1749 | 13.831351409 | MAC: GATEWAY | CyberTAN_b3:f7:51 (...) | 802.11 | 28 Acknowledgement, Flags=..... |
| 1750 | 13.832545459 | ee:fa:bc:08:04:d7 | CyberTAN_b3:f7:51 | 802.11 | 131 Probe Response, SN=1573, FN=0, Flags=....., BI=100, SSID=GATEWAY-WSN |
| 1751 | 13.833839741 | ee:fa:bc:08:04:d7 | CyberTAN_b3:f7:51 | 802.11 | 131 Probe Response, SN=1573, FN=0, Flags=...R...., BI=100, SSID=GATEWAY-WSN |
| 1752 | 13.834132299 | ee:fa:bc:08:04:d7 (...) | 802.11 | 28 Acknowledgement, Flags=..... | |

Fuente: elaborado por el autor

En la Figura 111 se muestra el envío y recepción de paquetes por parte de los nodos y la estación base.

Figura 111

Captura de paquetes

| | | | | | |
|-----|-----------|-------------------|-------------------|-------|--|
| 401 | 78.050330 | 192.168.4.6 | 192.168.4.2 | MySQL | 65 Response OK |
| 402 | 78.060693 | 192.168.4.2 | 192.168.4.6 | TCP | 54 57479 → 3306 [ACK] Seq=532 Ack=134 Win=5611 Len=0 |
| 403 | 78.406491 | CyberTAN_b3:f7:51 | Broadcast | ARP | 42 Who has 192.168.4.4? Tell 192.168.4.5 |
| 404 | 78.470710 | 192.168.4.2 | 192.168.4.6 | MySQL | 171 Request Query |
| 405 | 78.474342 | 192.168.4.6 | 192.168.4.2 | MySQL | 65 Response OK |
| 406 | 78.499740 | CyberTAN_b3:f7:51 | Espressi_61:79:18 | ARP | 42 Who has 192.168.4.2? Tell 192.168.4.5 |
| 407 | 78.503030 | Espressi_61:79:18 | CyberTAN_b3:f7:51 | ARP | 42 192.168.4.2 is at 7c:9e:bd:61:79:18 |
| 408 | 78.560354 | 192.168.4.2 | 192.168.4.6 | TCP | 54 57479 → 3306 [ACK] Seq=649 Ack=145 Win=5600 Len=0 |
| 409 | 78.919155 | 192.168.4.2 | 192.168.4.6 | MySQL | 171 Request Query |
| 410 | 78.922841 | 192.168.4.6 | 192.168.4.2 | MySQL | 65 Response OK |
| 411 | 79.000208 | CyberTAN_b3:f7:51 | Broadcast | ARP | 42 Who has 192.168.4.4? Tell 192.168.4.5 |
| 412 | 79.062643 | 192.168.4.2 | 192.168.4.6 | TCP | 54 57479 → 3306 [ACK] Seq=766 Ack=156 Win=5589 Len=0 |
| 413 | 79.343819 | 192.168.4.2 | 192.168.4.6 | MySQL | 171 Request Query |
| 414 | 79.347060 | 192.168.4.6 | 192.168.4.2 | MySQL | 65 Response OK |
| 415 | 79.383020 | 192.168.4.5 | 192.168.4.1 | SNMP | 83 get-request 1.3.6.1.2.1.1.3.0 |
| 416 | 79.388262 | 192.168.4.1 | 192.168.4.5 | SNMP | 98 get-response 1.3.6.1.2.1.1.3.0 |
| 417 | 79.564808 | 192.168.4.2 | 192.168.4.6 | TCP | 54 57479 → 3306 [ACK] Seq=883 Ack=167 Win=5578 Len=0 |
| 418 | 79.767428 | 192.168.4.2 | 192.168.4.6 | MySQL | 171 Request Query |

| | |
|--|--|
| > Frame 413: 171 bytes on wire (1368 bits), 171 bytes captured (1368 bits) on interface \Device\NPF_{7F64592C-D635-4DF5-B139-5AF694E8DF1B}, id 0 > Ethernet II, Src: Espressi_61:79:18 (7c:9e:bd:61:79:18), Dst: CyberTAN_b3:f7:51 (60:14:b3:b3:f7:51) > Internet Protocol Version 4, Src: 192.168.4.2, Dst: 192.168.4.6 > Transmission Control Protocol, Src Port: 57479, Dst Port: 3306, Seq: 766, Ack: 156, Len: 117 | |
| MySQL Protocol Packet Length: 113 Packet Number: 0 Request Command Query Command: Query (3) Statement: INSERT INTO QoADB.ValNODO1 (NOx, CO, PROPANO, TEMPERATURA, HUMEDAD, ETIQUETA) VALUES (10.34, 0.68, 5.15,13,48,1) | |

Fuente: elaborado por el autor

4.1.5. Bloque De Gestión

Se valida el cumplimiento de los requerimientos de sistema y de arquitectura especificados para el bloque de gestión mediante un test de gestión, el mismo que

permitirá realizar la evaluación de cumplimiento como se muestra en la Tabla 49. Con el propósito de verificar el cumplimiento de los requisitos en el diseño e implementación del sistema.

Tabla 49

Tabla de verificación de cumplimiento de requerimientos para bloque de gestión

| Tabla de validación de requerimientos sistema y arquitectura | | | | | | |
|---|--------------|--------------|---------------|---------------|---------------|---------------|
| Requerimientos | SYSR6 | SYSR7 | SRSH14 | SRSH15 | SRSH16 | SRSH17 |
| Cumple | SI | ✓ | ✓ | ✓ | ✓ | ✓ |
| | NO | | | | | |

Fuente: elaborado por el autor

4.1.5.1. Test de gestión

Se realiza el test de gestión como se muestra en la Tabla 50, mediante la creación de dispositivos (Nodos) que contienen sensores (Objetos) de acuerdo con los OID creados en cada uno de los Nodos, además se verifica la petición de la MIB de cada sensor mediante los GET-RESQUEST y la respuesta de los nodos mediante los mensajes GET-RESPONSE encapsulados en datagramas UDP, también se verifica la recepción de TRAPS.

Tabla 50

Test para el bloque de gestión

| Test de gestión | |
|-------------------------|---|
| Bloque de prueba | Bloque de gestión |
| Descripción | Prueba de funcionamiento para bloque de gestión |
| Prerrequisitos: | |
| | ✓ Descargar PRTG NETWORK MONITOR |
| | ✓ Instalar PRTG NETWORK MONITOR |
| | ✓ Agregar agentes SNMP en todos los elementos de la red WSN |

Pasos:

- ✓ Crear usuario y contraseña para administrador.
- ✓ Crear grupo denominado WSN.
- ✓ Crear dispositivos denominado Nodos y Gateway.
- ✓ Agregar sensores (Objetos)
- ✓ Verificar envío y recepción de mensajes SNMP.

Resultados esperados:

Creación correcta de elementos de la red WSN en el software PRTG NETWORK MONITOR, donde cada dispositivo realice el envío correcto de MIBs correspondiente y TRAPs dependiendo del evento acontecido, correcta petición por parte de la estación de gestión de MIBs a los agentes SNMP y notificación de eventos registrados por fallos o TRAPs por notificaciones instantáneas.

Fuente: elaborado por el autor

Resultado obtenido: la instalación y creación de credenciales de autenticación de administrador, se procedió a agregar dispositivos al grupo WSN donde se agregó sensores correspondientes a cada OID que se especificó en la Tabla 34 correspondiente a los OID de subárbol del nodo WSN a cada dispositivo de la red WSN, donde se aprecia que todos los dispositivos están funcionando correctamente, los mensajes GET-REQUEST y GET RESPONSE se envía correctamente, además los TRAPS son enviados satisfactoriamente. En la Figura 112 se muestra los mensajes GET-REQUEST, GET-RESPONSE y TRAPS enviados desde los NODOS al gestor capturados por un capturados de paquetes Wireshark, en respuesta a las peticiones hechas por la estación de gestión.

Figura 112

Trafico SNMP generado entre nodos y estación de gestión

| No. | Time | Source | Destination | Protocol | Length | Info |
|------|------------|-------------|-------------|----------|--------|---|
| 1894 | 288.347582 | 192.168.4.6 | 192.168.4.4 | SNMP | 83 | get-request 1.3.6.1.2.1.1.6.0 |
| 1895 | 288.770518 | 192.168.4.4 | 192.168.4.6 | SNMP | 109 | get-response 1.3.6.1.2.1.1.6.0 |
| 1897 | 289.348019 | 192.168.4.6 | 192.168.4.3 | SNMP | 83 | get-request 1.3.6.1.2.1.1.4.0 |
| 1898 | 289.348163 | 192.168.4.6 | 192.168.4.2 | SNMP | 83 | get-request 1.3.6.1.2.1.1.5.0 |
| 1899 | 289.638674 | 192.168.4.2 | 192.168.4.6 | SNMP | 107 | get-response 1.3.6.1.2.1.1.5.0 |
| 1900 | 289.685333 | 192.168.4.3 | 192.168.4.6 | SNMP | 111 | get-response 1.3.6.1.2.1.1.4.0 |
| 1903 | 290.552232 | 192.168.4.3 | 192.168.4.6 | SNMP | 120 | snmpV2-trap 1.3.6.1.2.1.1.3.0 1.3.6.1.2.1.1.2.0 |
| 1905 | 291.349894 | 192.168.4.6 | 192.168.4.4 | SNMP | 85 | get-request 1.3.6.1.3.1.1.2.3.1.0 |
| 1906 | 291.453273 | 192.168.4.4 | 192.168.4.6 | SNMP | 100 | get-response 1.3.6.1.3.1.1.2.3.1.0 |
| 1907 | 291.454105 | 192.168.4.6 | 192.168.4.4 | SNMP | 83 | get-request 1.3.6.1.2.1.1.3.0 |
| 1908 | 291.578701 | 192.168.4.4 | 192.168.4.6 | SNMP | 98 | get-response 1.3.6.1.2.1.1.3.0 |
| 1909 | 292.348318 | 192.168.4.6 | 192.168.4.3 | SNMP | 83 | get-request 1.3.6.1.2.1.1.5.0 |
| 1910 | 292.349127 | 192.168.4.6 | 192.168.4.1 | SNMP | 83 | get-request 1.3.6.1.2.1.1.4.0 |
| 1911 | 292.349276 | 192.168.4.6 | 192.168.4.2 | SNMP | 83 | get-request 1.3.6.1.2.1.1.6.0 |
| 1912 | 292.357037 | 192.168.4.1 | 192.168.4.6 | SNMP | 106 | get-response 1.3.6.1.2.1.1.4.0 |
| 1913 | 292.475611 | 192.168.4.2 | 192.168.4.6 | SNMP | 109 | get-response 1.3.6.1.2.1.1.6.0 |
| 1914 | 292.488994 | 192.168.4.3 | 192.168.4.6 | SNMP | 120 | snmpV2-trap 1.3.6.1.2.1.1.3.0 1.3.6.1.2.1.1.2.0 |

Fuente: elaborado por el autor

En la Figura 113 se muestra que las notificaciones se muestran y cada uno de los sensores activados en el software de monitoreo funcionan correctamente cambiando de color de acuerdo con los valores establecidos.

Figura 113

Verificación cambio de color de acuerdo con los límites establecidos

| Red WSN | | | | | | | | | | | | | | | | | | | |
|------------------|---------|------------------|---------|-----------------|----------|-----------------|----------|----------------|----------|-----------------|----------|-----------------|-----------|-----------------|----------|------------------|----------|------------------|------------|
| NODO 1 | | | | | | | | | | | | | | | | | | | |
| Ping | 35 mseg | Tiempo de act... | 5 m 1 s | SysContact | 215 mseg | SysDesor | 364 mseg | SysName | 92 mseg | SysLocation | 244 mseg | NOx | 6.322 ppm | CO | 1 ppm | PROPANO | 152 ppm | Temperatura | 16 Celsius |
| Humedad Relat... | 36 % | Valor Etiqueta | 3 ICA | TRAP SENSOR ... | 0 #/s | TRAP SENSOR ... | 0 #/s | TRAP NODO R... | 0 #/s | StatusSensor... | 490 mseg | StatusSensor... | 195 mseg | StatusSensor... | 347 mseg | StatusSensorD... | 499 mseg | StatusSensorD... | 225 mseg |
| NODO 2 | | | | | | | | | | | | | | | | | | | |
| Ping | 13 mseg | Tiempo de act... | 117 s | SysContact | 573 mseg | SysDesor | 307 mseg | SysName | 495 mseg | SysLocation | 251 mseg | NOx | 4 ppm | CO | 0 ppm | PROPANO | 0 ppm | Temperatura | 16 Celsius |
| Humedad Relat... | 37 % | Valor Etiqueta | 1 ICA | TRAP SENSOR ... | 1,47 #/s | TRAP SENSOR ... | 2,28 #/s | TRAP NODO R... | 0 #/s | StatusSensor... | 423 mseg | StatusSensor... | 179 mseg | StatusSensor... | 372 mseg | StatusSensorD... | 124 mseg | StatusSensorD... | 325 mseg |
| NODO 3 | | | | | | | | | | | | | | | | | | | |
| Ping | 18 mseg | Tiempo de act... | 131 s | SysContact | 585 mseg | SysDesor | 321 mseg | SysName | 80 mseg | SysLocation | 271 mseg | NOx | 1.220 ppm | CO | 20 ppm | PROPANO | 0 ppm | Temperatura | 19 Celsius |
| Humedad Relat... | 33 % | Valor Etiqueta | 3 ICA | TRAP SENSOR ... | 2,27 #/s | TRAP SENSOR ... | 2,28 #/s | TRAP NODO R... | 0 #/s | StatusSensor... | 455 mseg | StatusSensor... | 216 mseg | StatusSensor... | 410 mseg | StatusSensorD... | 170 mseg | StatusSensorD... | 365 mseg |

Fuente: elaborado por el autor

4.1.6. Bloque De Visualización

Se valida el cumplimiento de los requerimientos de sistema y de arquitectura especificados para el bloque de visualización mediante un test de visualización de datos que permitirá establecer los resultados para la evaluación de cumplimiento como se

muestra en la Tabla 51. Con el propósito de verificar el cumplimiento de los requisitos en el diseño e implementación del sistema.

Tabla 51

Tabla de verificación de cumplimiento de requerimientos para bloque de visualización

| Tabla de validación de requerimientos sistema y arquitectura | | | |
|---|---------------|---------------|---------------|
| Requerimientos | SYSR12 | SYSR13 | SRSR17 |
| SI | ✓ | ✓ | ✓ |
| Cumple | | | |
| NO | | | |

Fuente: elaborado por el autor

La visualización de los datos, tanto de los sensores, nivel de concentración de material particulado o índice de calidad del aire, cumple con los requerimientos y se realiza un test de visualización de datos para verificar el correcto funcionamiento.

4.1.6.1. Test de visualización de datos

Se realiza el test de visualización como se muestra en la Tabla 52, mediante la visualización de los valores obtenidos por el bloque de adquisición de datos y el nivel de contaminación del aire proporcionado por el algoritmo de clasificación supervisada, además se verifica la recepción de TRAPs y las notificaciones instantáneas enviadas.

Tabla 52

Test para el bloque de gestión

| Test de visualización de datos | |
|---------------------------------------|---|
| Bloque de prueba | Bloque de visualización de datos |
| Descripción | Test de visualización de datos |
| Prerrequisitos: | |
| ✓ | PRTG NETWORK MONITOR |
| ✓ | Agentes SNMP en todos los elementos de la red WSN |
| Pasos: | |

-
- ✓ Acceder a la interfaz web mediante la dirección ip de la estación de gestión.
 - ✓ Inspeccionar los dispositivos y sensores.
 - ✓ Verificar la visualización de los porcentajes de concentración de gases.
 - ✓ Verificar la visualización de nivel de contaminación.
 - ✓ Verificar el estado de todos los objetos de los dispositivos.
 - ✓ Verificación de mensajes correspondientes a TRAPS.
 - ✓ Verificación de notificaciones.
 - ✓ Verificar datos guardados en base de datos.

Resultados esperados:

Visualización correcta de todos los valores de los sensores y nivel de contaminación, además correcta visualización de mensajes para identificar el tipo de evento imprevisto y visualización de notificaciones de alerta además de visualizar los datos registrados en la base de datos.

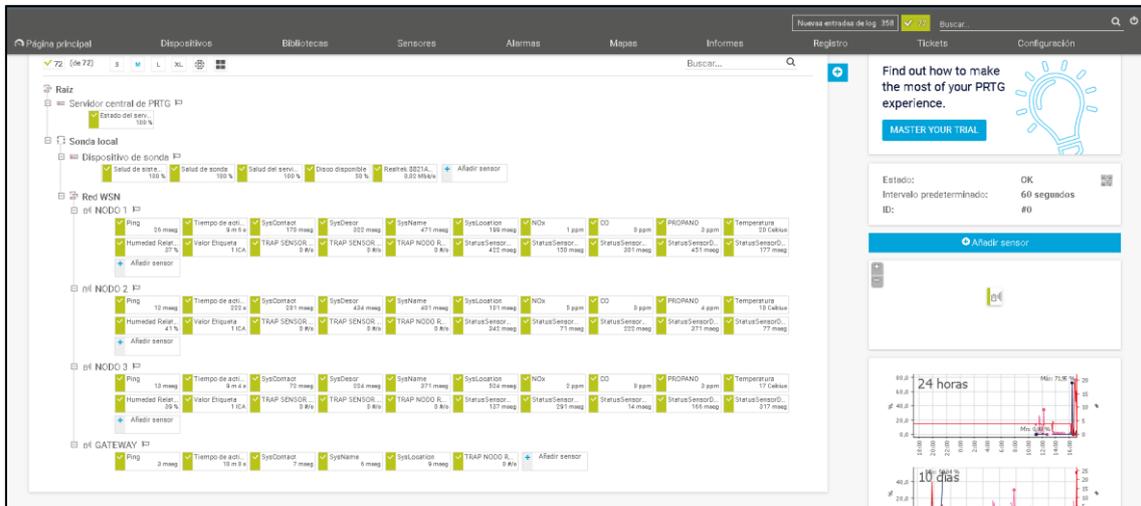
Fuente: elaborado por el autor

Resultado obtenido: mediante el software PRTG NETWORK MONITOR se logra acceder a la interfaz web del servidor, donde se puede visualizar todos los dispositivos agregados, además de cada sensor agregado, donde dependiendo del color (verde = OK, amarillo= Advertencia, rojo= Fallo y azul = Pausado) se puede verificar el estado de funcionamiento de los dispositivos. Además, para el nivel de concentración de gases se ha creado indicadores tipo velocímetro que adoptan el tipo de color antes mencionado, la base de datos presenta los datos de cada uno de los sensores tipo histograma para una visualización más fácil para el usuario.

En la Figura 114 se muestra que todos los dispositivos poseen un estado normal establecido por el color verde y un objeto en rojo que indica que existe un problema en el sensor, es decir recibieron correctamente los datos solicitados por el gestor.

Figura 114

Red WSN activa

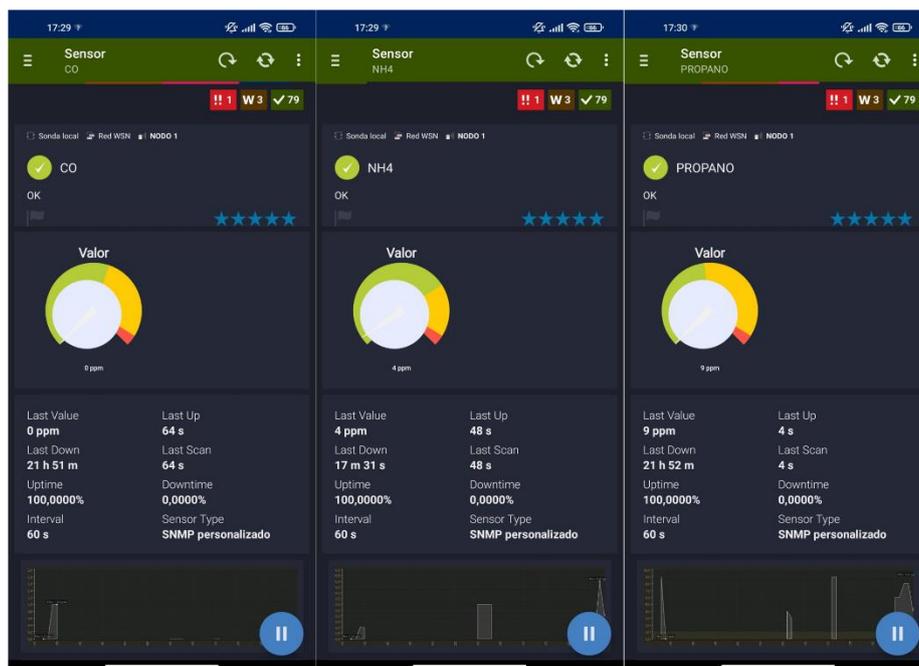


Fuente: elaborado por el autor

En la Figura 115 se puede apreciar los valores en ppm del bloque de adquisición de datos, dependiendo de cada sensor desde la APP. Donde se ha configurado un límite superior para notificar cuando los niveles superen los establecidos en la Tabla 40.

Figura 115

Valores de sensores desde APP



Fuente: elaborado por el autor

En la Figura 116 se muestra los traps enviados desde los diferentes nodos a la estación de gestión, donde se puede apreciar el OID que le corresponde dependiendo a la Tabla 36 donde se especifica el OID para traps de DOWN, NO READ Y REBOOTING.

Figura 116

Recepción y visualización de mensajes TRAP.

| Mensajes trap | | | | |
|--------------------|-------------|-------|------------|---|
| | Source | Agent | Enterprise | Bindings |
| 18/8/2022 20:14:23 | 192.168.4.4 | | | RFC1213-MIB::sysObjectID.0 = SNMPv2-SMI::experimental.1.2.3.0 |
| 18/8/2022 20:14:23 | 192.168.4.4 | | | RFC1213-MIB::sysObjectID.0 = SNMPv2-SMI::experimental.1.2.3.0 |
| 18/8/2022 20:14:23 | 192.168.4.4 | | | RFC1213-MIB::sysObjectID.0 = SNMPv2-SMI::experimental.1.2.3.0 |
| 18/8/2022 20:14:23 | 192.168.4.4 | | | RFC1213-MIB::sysObjectID.0 = SNMPv2-SMI::experimental.1.2.3.0 |
| 18/8/2022 20:14:23 | 192.168.4.4 | | | RFC1213-MIB::sysObjectID.0 = SNMPv2-SMI::experimental.1.2.3.0 |
| 18/8/2022 20:14:23 | 192.168.4.4 | | | RFC1213-MIB::sysObjectID.0 = SNMPv2-SMI::experimental.1.2.3.0 |

Fuente: elaborado por el autor

Cuando un sensor del bloque de adquisición de datos no tenga lecturas enviara a la estación base un TRAP SENSOR NO READ, en la Figura 117 se muestra la alerta en pantalla de monitorización

Figura 117

Notificación de SENSOR NO READ



Fuente: elaborado por el autor

Cuando un sensor del bloque de adquisición de datos no tenga lecturas es decir el valor de lectura igual a cero y ha transcurrido 1 minuto desde al ultima lectura de un valor diferente de cero se activa y se procede a enviar a la estación base un TRAP SENSOR DOWN que indica que el sensor esta apagado, en la Figura 118 se muestra la alerta en pantalla de monitorización.

Figura 118

Notificación SENSOR DOWN



Fuente: elaborado por el autor

En la Figura 119 se muestra los datos guardados en forma de histogramas, donde se puede apreciar los valores enviados por los nodos sensores a la estación base en diferentes rangos de tiempo (último día, últimos 10 días y últimos 100 días).

Figura 119

Resúmenes de registros tipo histograma

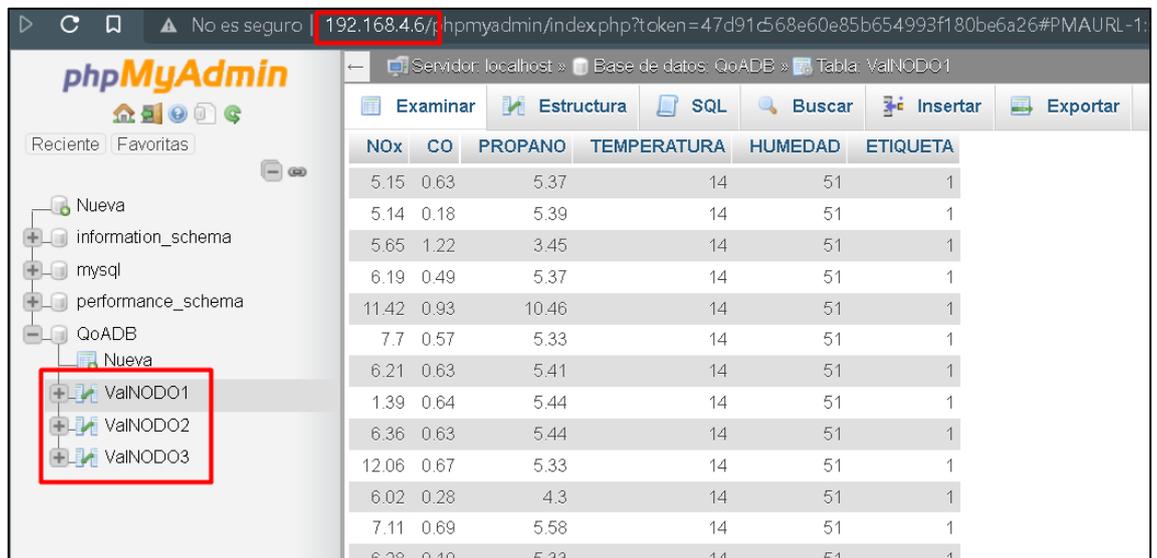


Fuente: elaborado por el autor

En la Figura 120 se muestra los datos guardados en la base de datos MySQL, de cada uno de los nodos sensores, en tablas individuales.

Figura 120

Visualización de la base de datos MySQL

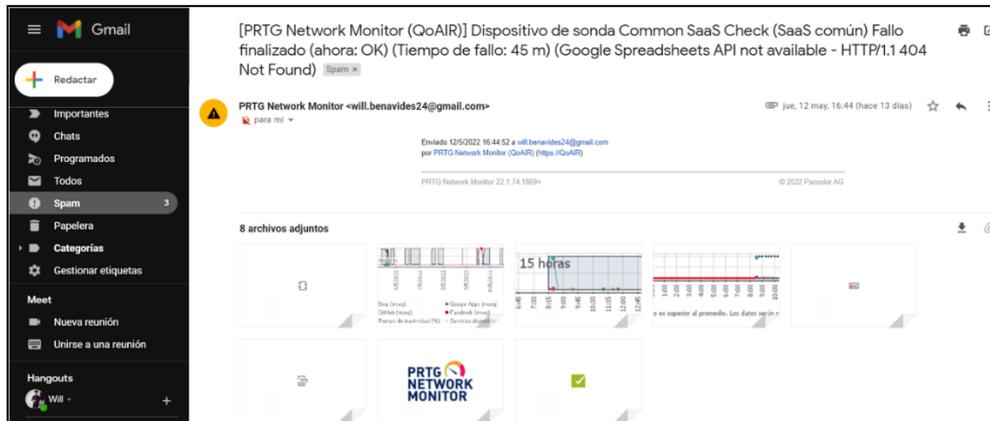


Fuente: elaborado por el autor

En la Figura 121 se muestra las notificaciones enviadas al correo, notificaciones en la página web y notificaciones en la aplicación celular.

Figura 121

Notificación por correo

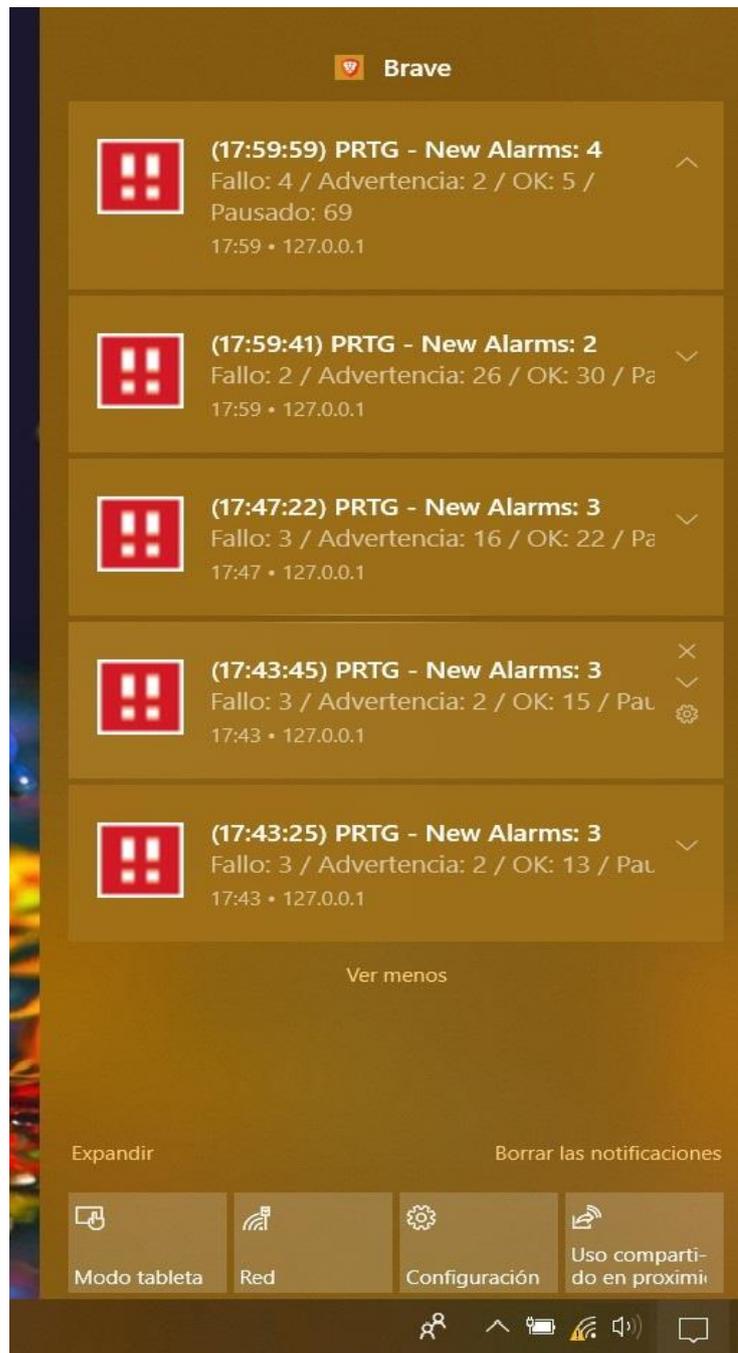


Fuente: elaborado por el autor

En la Figura 122 se muestra cómo se muestran las notificaciones que se activan de acuerdo con los parámetros establecidos, en este caso las notificaciones en pantalla visualizado desde un ordenador sistema operativo Windows 10.

Figura 122

Notificación de eventos del software de monitoreo

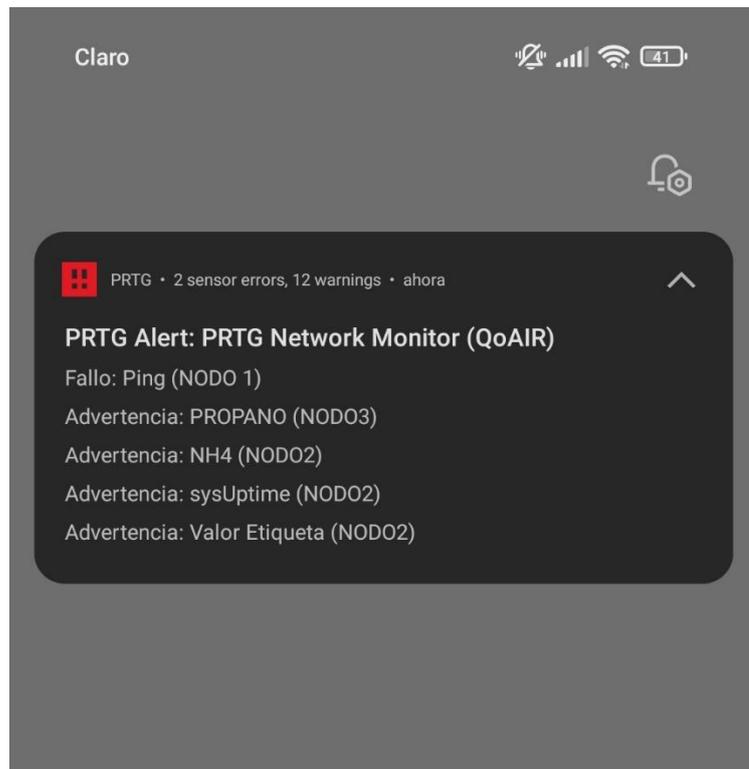


Fuente: elaborado por el autor

En la Figura 123 se muestra cómo se muestran las notificaciones que se activan de acuerdo con los parámetros establecidos, en este caso las notificaciones en pantalla de bloqueo del dispositivo móvil, visualizado desde un dispositivo móvil Xiaomi sistema operativo Android 11.

Figura 123

Visualización de notificaciones en dispositivo móvil.



Fuente: elaborado por el autor

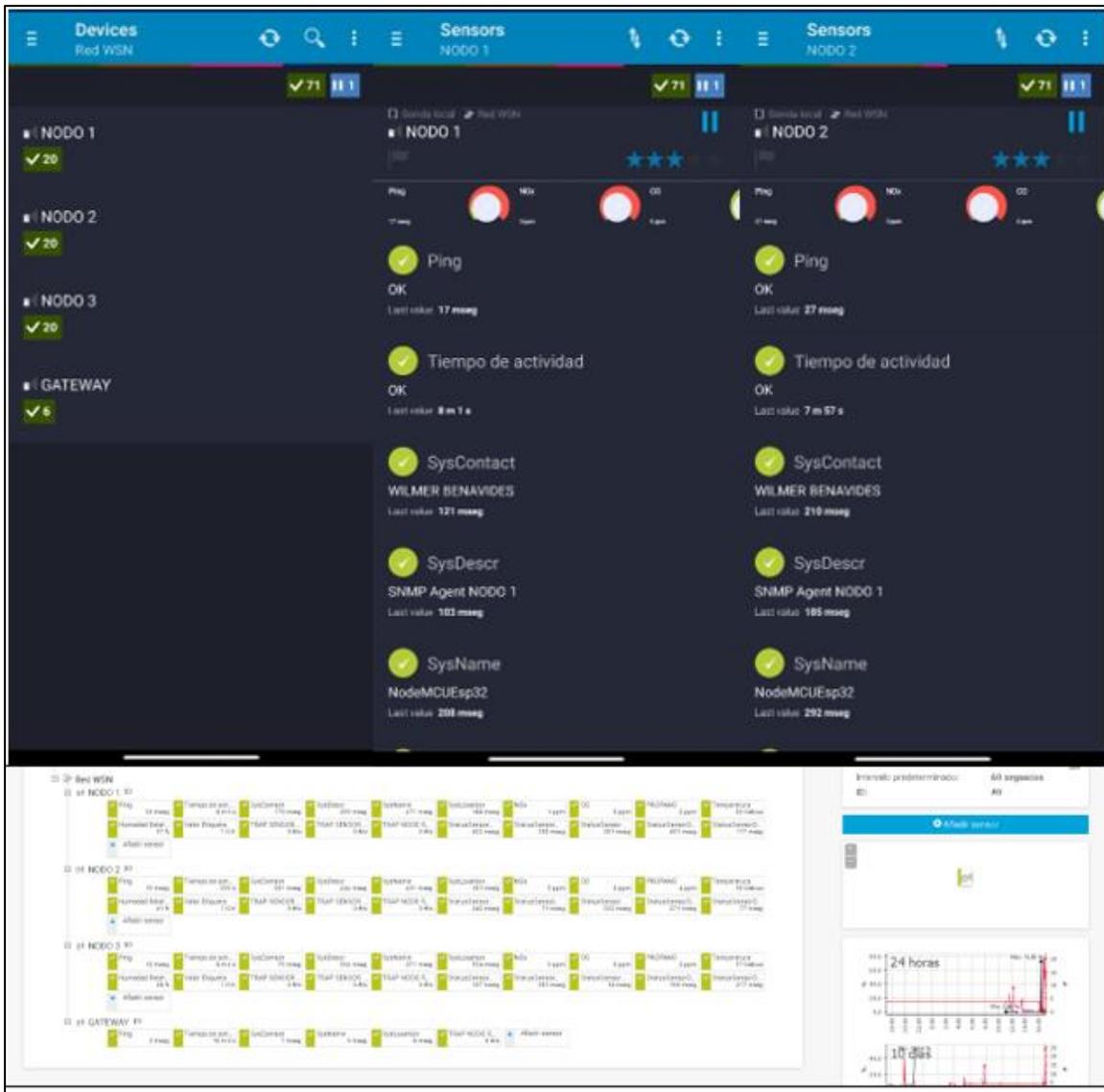
CAPITULO V. Resultados, Conclusiones y Recomendaciones

5.1.Resultados

Para la demostración de los resultados primero se ubica los nodos sensores y el Gateway, posteriormente se verifica que el sistema se encuentre funcionando de manera correcta, de tal manera que se logre visualizar los valores de los sensores en estado activo en la interfaz gráfica del servidor y APP. En la Figura 124 se muestra todos los sensores activos tanto en la interfaz gráfica como en la APP y la interfaz de visualización para ordenadores.

Figura 124

Red WSN activa monitorizada desde aplicación móvil y la interfaz de visualización para ordenadores.



Fuente: elaborado por el autor

Verificado que el sistema esté funcionando correctamente, se inicia la fumigación por parte de un trabajador de la florícola debidamente equipado, y se procede a medir las concentraciones de materiales particulados producidos por los residuos de los químicos utilizados y la combustión de derivados de petróleo usado en la moto bomba. En la Figura 125 se observa al trabajador realizando la fumigación y al nodo sensor.

Figura 125

Fumigación del bloque con sistema de monitoreo activo

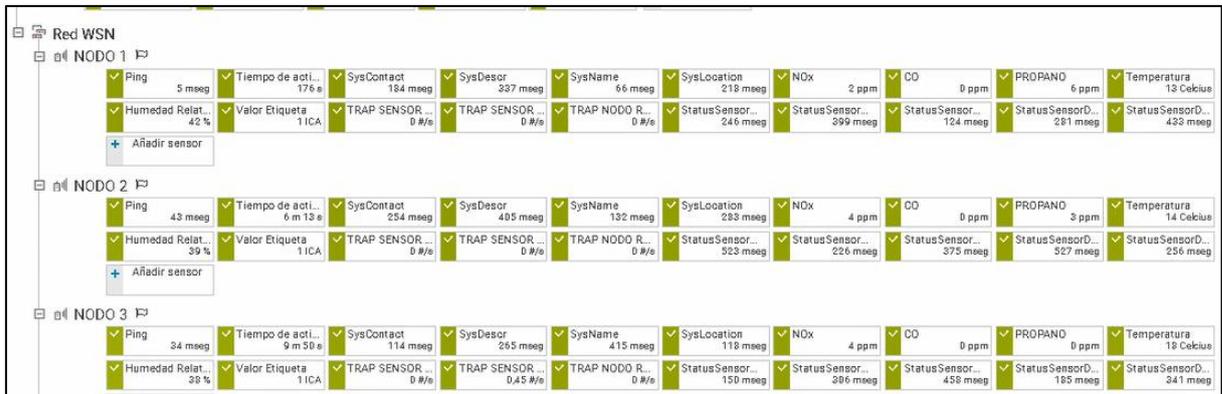


Fuente: elaborado por el autor

Se ha realizado el mapeo del invernadero antes, durante y después de la fumigación, para lograr determinar el estado en el que se encuentra, al inicio de la fumigación se puede observar que los niveles se mantenían en valores normales, como se muestra en la Figura 126.

Figura 126

Niveles de concentración de material particulado normal.



Fuente: elaborado por el autor

Quando empieza la fumigación el nodo más cercado empieza a elevar los valores de NOx y PROPANO, al utilizar combustibles fósiles para la motobomba también los niveles de CO empiezan a incrementar. En la Figura 127 se muestra como los niveles de NOx. PROPANO y CO empiezan a incrementar.

Figura 127

Valores de NoX, PROPANO y CO en incremento durante la fumigación.



Fuente: elaborado por el autor

Al transcurrir un tiempo desde el inicio de la fumigación los valores de concentración de material particulado empiezan a incrementar de igual manera el índice de calidad de aire empieza a cambiar de normal = 1 a aceptable =2, en la Figura 128 se muestra la etiqueta de ICA.

Figura 128

Valor de ICA

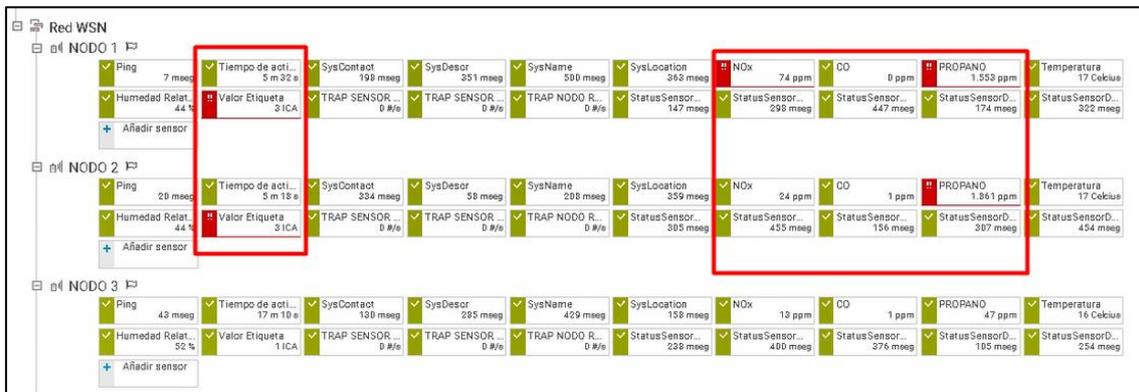


Fuente: elaborado por el autor

Cuando el fumigador pasa por el Nodo los niveles de concentración se elevan hasta superar los niveles aceptables, esto se puede apreciar en los colores de cada objeto que cambian de verde a amarillo y después a rojo. En la Figura 129 se muestra el incremento en la concentración y el cambio de color de cada objeto.

Figura 129

Incremento en la concentración y cambio de color en cada objeto.



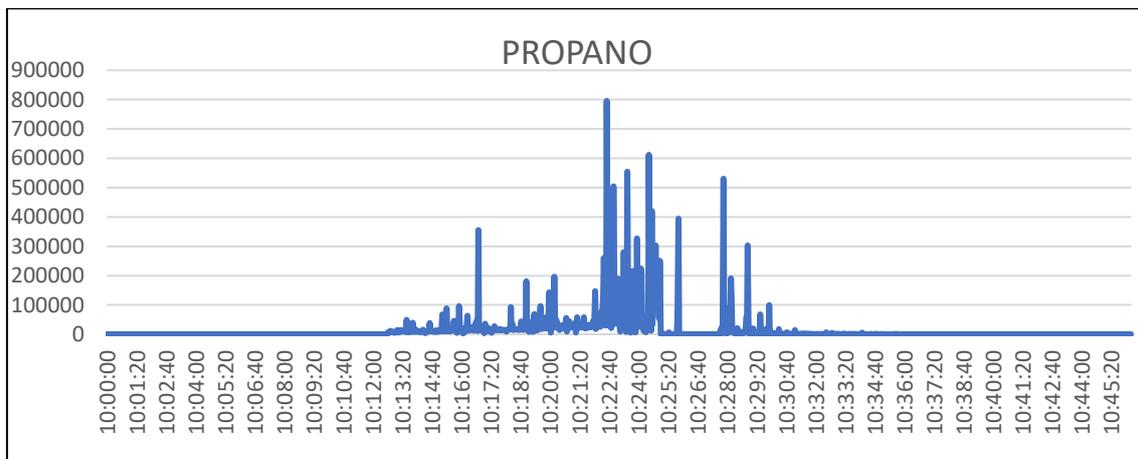
Fuente: elaborado por el autor

Durante la fumigación se registraron los datos obtenidos por el bloque de adquisición de datos de cada nodo, para posteriormente realizar un análisis del tiempo que transcurre desde que los niveles de concentración de propano empiezan a detectarse, hasta cuando los niveles regresan a la normalidad, desde el minuto cero que empieza la fumigación transcurren alrededor de 12 minutos hasta que el bloque de adquisición de

datos muestra los primeros incrementos en los niveles de propano, teniendo en el minuto 22 el valor más alto que es cuando el fumigador pasa por el nodo y van variando los niveles hasta el minuto 34, donde los niveles de propano empieza a normalizarse. En la Figura 130 se muestra la gráfica de línea para observar la línea temporal de variación de niveles de propano.

Figura 130

Línea temporal de la variación de PROPANO durante la fumigación.

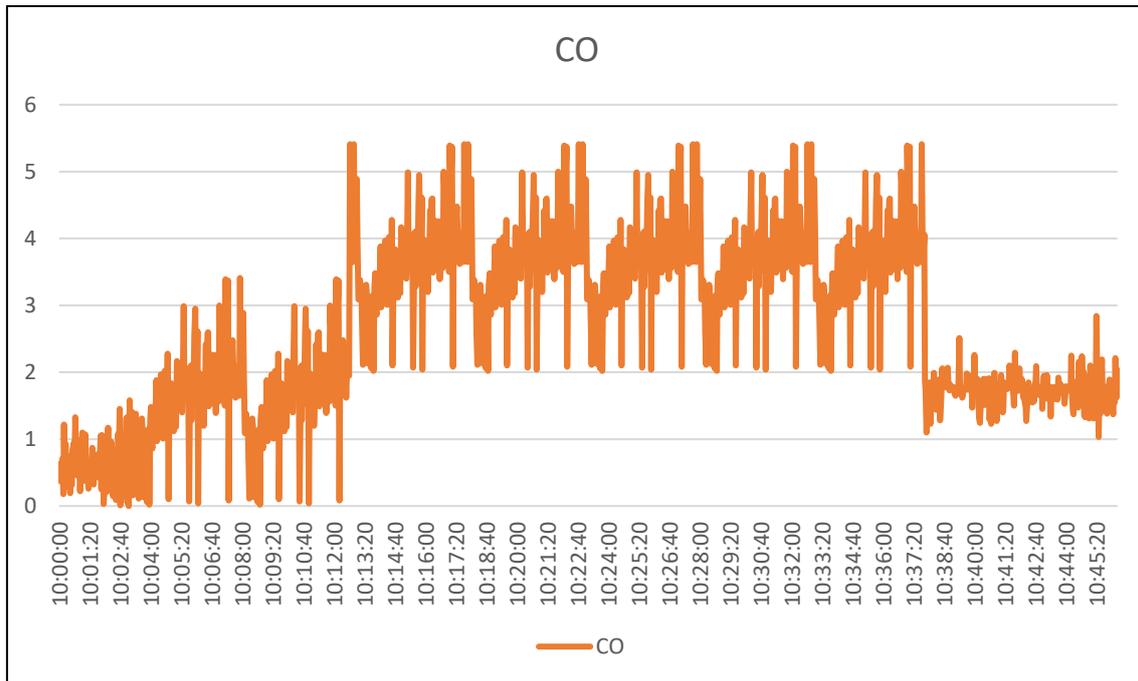


Fuente: elaborado por el autor

Al usar una motobomba que su funcionamiento es a base de combustibles fósiles, se tiene desde el minuto que se enciende la motobomba presencia de monóxido de carbono CO, teniendo que en los momentos que la motobomba aumenta la presión para la dispersión del producto químico los niveles de CO aumentan, en la Figura 131 se muestra una línea temporal de la variación en los valores de CO.

Figura 131

Línea temporal de la variación de CO durante la fumigación.

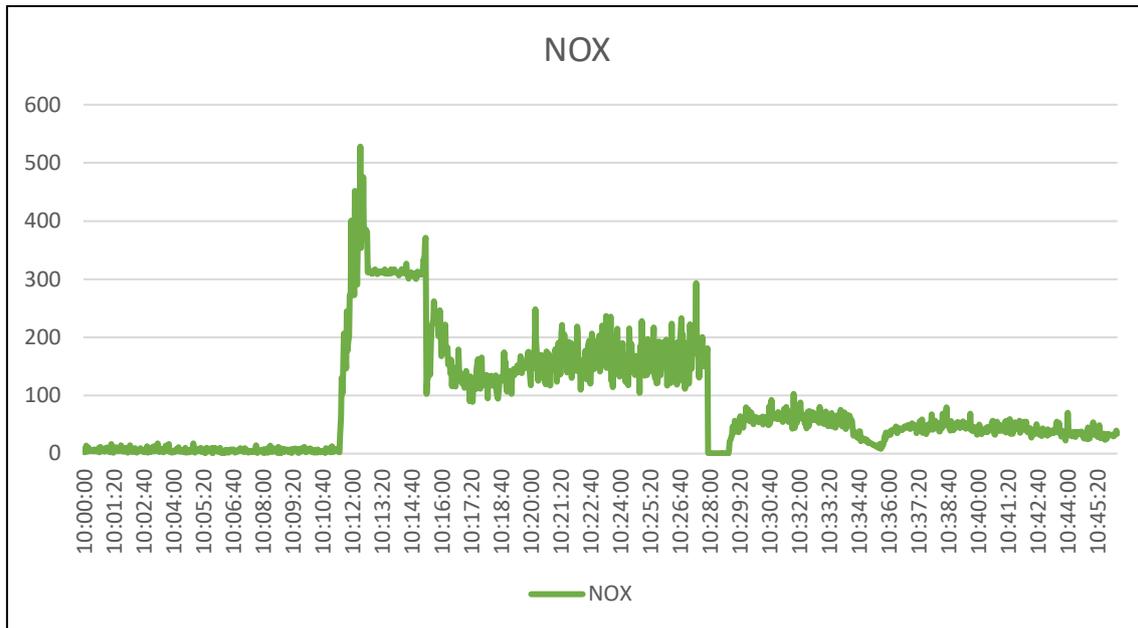


Fuente: elaborado por el autor

Para los óxidos de nitrógeno NO_x, se tiene que al minuto 10 desde empezar la fumigación los valores de NO_x empiezan a variar, manteniendo la variación por un periodo comprendido desde el minuto 10 hasta el minuto 28, posteriormente se manteniéndose en un nivel normal, en la Figura 132 se muestra la línea temporal de la variación de NO_x.

Figura 132

Línea temporal de la variación de NOx durante la fumigación.



Fuente: elaborado por el autor

Con las gráficas obtenidas de propano y óxidos de nitrógeno (NOx) se logra establecer que existe un tiempo de referencia desde (t=0) donde empieza la fumigación hasta transcurrir un tiempo (t=10 min) se puede apreciar como los niveles de concentración se mantienen en niveles de alerta, es decir propano (>1000 ppm) y NOx(>50 ppm), por un tiempo de aproximado de 20 minutos hasta llegar a un tiempo (t=28 min) donde los valores empiezan a decrecer y se establecen en niveles aceptables, es decir propano entre (500 – 1000 ppm) y NOx entre (30 – 50 ppm) durante un tiempo aproximado de 15 min (t=45 min), esto quiere decir, que para el ingreso de los trabajadores al invernado se debe esperar al menos 35 minutos, lo ideal y recomendado es esperar hasta que los niveles de concentración bajen a niveles normales, es decir propano (>0 y < 500) y NOx(>0 y < 50 ppm) que se logra al transcurrir una hora desde que se termina la fumigación. Para el monóxido de carbono (CO) depende del estado del sistema de combustión del motor, ya que, por la mala combustión de la gasolina se

produce el CO produciendo un humo gris, en este caso el motor utilizado en la fumigación está en buen estado, mantenido los niveles de CO no mayor a los 5.5 ppm, lo cual entra en los niveles aceptables de concentración de CO que se mantiene entre (4.5 – 6.4 ppm).

5.2.Conclusiones

Se logró la implantación del sistema de alerta y monitoreo con un nivel muy aceptable a niveles de hardware como software, logrando una integración de los diferentes protocolos como los de capa aplicación (SNMP), transporte (UDP), internet (IP) y acceso a la red (802.11G) en referencia a la arquitectura TCP/IP, los cuales forman parte de cada uno de los bloques que constituyen el sistema.

Mediante el estudio bibliográfico y la entrevista realizada a el ingeniero agrónomo encargado de la florícola, se logró identificar que no existe aún un sistema que ayude a medir y alertar la presencia de material particulado toxico, esto conlleva a que los trabajadores se exponen directamente al contacto de material particulado, ocasionando afectaciones graves en la salud a largo plazo, por lo cual es necesaria la implementación de un sistema que pueda medir y alertar cuando las concentraciones de material particulado excedan los límites permitidos.

Al establecer semillas con un número considerable de muestras se logra crear una matriz de entrenamiento que logre generar al algoritmo de clasificación supervisada una precisión de clasificación superior al 99%, lo cual permite al usuario tener un sistema electrónico optimo con lecturas acertadas al momento de censar el medio.

Durante la medición de la concentración de material particulado en el proceso de fumigación con plaguicidas se logró establecer una mayor concentración de propano debido a el uso de productos derivados del petróleo como plaguicidas, fungicidas y herbicidas, además al usar combustible fósil para el funcionamiento de la motobomba se

tiene niveles aceptables de monóxido de carbono, estableciendo el mayor foco mientras se realiza el proceso de fumigación y reduciendo gradualmente la concentración, el tiempo de fumigación se tarda un lapso de más o menos una hora y hasta que la concentración baja a niveles aceptables existe un lapso de alrededor de 30 minutos.

El sistema de alerta y monitoreo al trabajar bajo un modelo de gestión de red Internet, proporciona el usuario una mayor flexibilidad ya que integra las dos principales necesidades, administrar la red y la visualización de las concentraciones de material particulado del invernadero, ya que al ser usar un modelo experimental se logra adaptar objetos en los cuales se puede incluir los información de los recursos de los dispositivos y niveles de concentración, teniendo en la interfaz de visualización que abarca todos los requerimientos establecidos.

Al desarrollar un sistema basado en un modelo de gestión de red internet, bajo el protocolo SNMP, donde en su estructura permite crear nuevos objetos con su respectivo OID, sin ocasionar problemas de reconocimiento en los softwares de monitoreo de redes, proporciona una escalabilidad en el número de recursos gestionables en nuevas tecnologías y sistemas basados en software libre como lo es Arduino.

Al usar un estándar enfocado a proporcionar las arquitecturas necesarias para el despliegue de redes IOT como lo es el estándar ISO/IEC/IEEE 15288, se logra establecer parámetros de verificación y validación para el sistema desarrollado, tomando en cuenta el estándar se desarrolló pruebas de verificación y validación a cada uno de los bloques que constituyen el sistema de alerta y monitoreo, tanto a nivel de software como hardware, obteniendo un puntaje aceptable en cada uno de los puntos evaluados.

5.3.Recomendaciones

Con el diseño de un stock de sensores que permitan medir las concentraciones de material particulado, es necesaria el precalentamiento en aire limpio para mejorar la eficiencia en la medición de cada sensor y posteriormente realizar la calibración de acuerdo con la gráfica de sensibilidad de cada sensor especificada en el datasheet del fabricante.

Al generar la matriz de entrenamiento se debe tomar varias muestras en varios ambientes para lograr establecer valores en lo más cercano a el ambiente en el cual se va a implementar el sistema, con un número de muestras considerables, debido a que la matriz de entrenamiento son los datos que se van a tomar de referencia para clasificar los datos de entrada de los sensores.

Para el diseño de la red WSN se debe considerar como parámetros fundamentales dimensiones del invernadero, protocolo de comunicación inalámbrico, área de cobertura y data rate, para la distribución correcta de los nodos, ya que al ubicarlos muy alejados se puede causar atenuación y pérdida de datos.

Al usar un modelo de gestión de red Internet se debe seguir la estructura general SMI para establecer el nombre de identificación de objeto OID, para no causar problemas al crear sensores en la estación de gestión ya que si no se sigue la nomenclatura SMI no podrá identificar los objetos de los agentes SNMP.

En la estructura del NodeMCU ESP32 posee 38 pines en los cuales existen una distribución especial si se pretende usa el módulo de comunicación inalámbrico, es decir no se puede usar todos los puertos ADC si se usa el módulo Wi-Fi. Ya que si se los usa las lecturas obtenidas por esos pines tienden a infinito.

BIBLIOGRAFÍA

- Paessler AG. (2022). *PRTG NETWORK MONITOR*. Obtenido de <https://www.paessler.com/es/prtg/features>
- Aakvaag, N., & Frey, J.-E. (2016). Redes de sensores inalámbricos Nuevas soluciones de interconexión para la automatización industrial. *ABB*, 39-42.
- Alhojely, S. (2016). Sentiment Analysis and Opinion Mining: A Survey. *International Journal of Computer Applications*, 150(6):22-25.
- ANSI/IEEE sta 802.11. (2003). Wireless LAN Medium AccessControl (MAC) and Physical Layer (PHY) Specifications.
- ARDUINO. (2022). *What Is Arduino?* Obtenido de <https://www.arduino.cc/>
- BALLESTEROS, D. M., & TORRES, D. R. (2018). *INTRODUCCIÓN A LOS FILTROS DIGITALES UN ENFOQUE TEÓRICO-PRÁCTICO*. REDIPE Red Iberoamericana de Pedagogía.
- Bayas, J. I. (2015). *SERVIDOR DE CONTROL DE DISPOSITIVOS Y SERVICIOS MEDIANTE EL PROTOCOLO SNMP PARA LA RED DE DATOS EN CELEC .E.P. UNIDAD DE NEGOCIO HIDROAGOYAN*. Ambato.
- Bonet, M. (2016). *El imperio del aire: Espectro radioeléctrico y radiodifusión*. (E. UOC, Ed.) Obtenido de <https://elibro.net/es/ereader/utnorte/58583?page=58>
- Cabrera, W. (2020). Fundamentos de Redes y Telecomunicaciones. Obtenido de <http://instipp.edu.ec/instipp/assets/pdf/guias/redes/rt-s2-fundamento.pdf>
- Carballeira, F. G. (2015). *El lenguaje de programación C: diseño e implementación de programas*.

- Chacín, R. B. (2015). *PROTOCOLO DE GESTIÓN INTERNET*. Maracaibo,.
- Córdoba, D. M., & Buitrago, F. A. (2013). *STATE OF THE ART OF WIRELESS SENSOR NETWORKS*. Obtenido de <https://revistas.udistrital.edu.co/tia/article/download>
- CWAP. (2018). 802.11 PHY Layers.
- Dahshan, M. H. (2017). *Computer Network Management SNMPv2*.
- Emiliano, G. P. (2018). *DESARROLLO DE UNA HERRAMIENTA DE MINERÍA DE DATOS PARA EL ANÁLISIS DE INFLUENCIA DE CUENTAS AUTOMATIZADAS EN TEMAS DE TENDENCIA SOBRE LA OPINIÓN DE LOS USUARIOS DE TWITTER EN ECUADOR*. Quito.
- Espínola, N., Córdova, A., Díaz, S., & Damas, R. (2015). Contaminación del aire por productos químicos y su repercusión en la Salud. 17-19.
- Espressif Systems. (2021). *ESP32 Series*. Obtenido de https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- Espressif Systems IOT Team. (2015). *ESP8266EX Datasheet*. Obtenido de <https://bbs.espressif.com/>
- Fiallos, S. F. (2017). *Estudio y análisis para la actualización de red wlan de la seps utilizando tecnologías basadas en el estándar ieee802.11 ac*. Quito.
- Fundación PLAGBOL. (2018). *Manual de Diagnóstico Tratamiento y Prevención de Intoxicaciones Agudas por plaguicidas*. La Paz, Bolivia . Obtenido de <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwiVy4udwMP0AhXHRDABHR2dBcMQFnoEAcQAQ&url=http%3>

A%2F%2Fwww.enfermeriaaps.com%2Fportal%2Fdownload%2FURGENCIAS

-

INTOXICACIONES%2FManual%2520DiagnosticoTratamiento%2520Intoxicaciones%252

Garbarino, J. (2017). *Protocolos para redes inalámbricas de sensores*. Buenos Aires.

Gill, K., Yang, S.-H., Yao, F., & Lu, X. (2009). A ZigBee-Based Home Automation System. *IEEE Transactions on Consumer Electronics VOL 55*.

Gutiérrez, O. E. (2020). *Comunicaciones móviles y redes inalámbricas*. Jorge Sarmiento

.

HANWEI ELETRONICS CO.,LTD. (s.f.). *TECHNICAL DATA MQ-2 GAS SENSOR*.

Huang, G., Song, S., Gupta, J. N., & Wu, C. (2014). Semi-Supervised and Unsupervised Extreme Learning Machines. *IEEE Transactions on Cybernetics*, 44(12):1-1.
doi:10.1109/TCYB.2014.2307349

Huawei. (10 de Febrero de 2021). *What is SNMP*. Obtenido de

<https://support.huawei.com/enterprise/en/doc/EDOC1100086963>

ISO / IEC / IEEE 15288. (2015). System and software engineering - system life cycle processes. *IEEE Xplore*.

ISO/IEC 30141. (2018). Internet Of Things (IoT) -Reference Architecture.

ISO/IEC-8824-1. (2015). *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation — Part 1*. Recuperado el 16 de Noviembre de 2021, de <https://www.iso.org/standard/68350.html>

- Jose, P. (2018). *Data Science and Machine Learning Bootcamp with R*. Obtenido de <https://www.udemy.com/data-science-and-machine-learning-bootcamp-with-r/>.
- Joshi, G. P., & Nam, S. Y. (2018). Cognitive Radio Wireless Sensor Networks: Applications, Challenges and Research Trends. doi:10.3390/s130911196
- Khanduri, R., Rattan, S. S., & Uniyal, A. (2013). Understanding the Features of IEEE 802.11g in High Data Rate Wireless LANs. *International Journal of Computer Applications*.
- Londoño, C. A., & Vasco, G. J. (2008). RELACIÓN ENTRE LAS PARTÍCULAS FINAS (PM2.5) Y RESPIRABLES (PM10) EN LA CIUDAD DE MEDELLÍN. *Revista Ingenierías Universidad de Medellín*, 23-42.
- Lopez, S. R. (2018). Aplicación de sistema de control y mantenimiento del nivel de CO2 en espacios cerrados.
- Marchán, M. V. (2018). *Análisis de la interferencia provocada por cámaras inalámbricas de video a 2,4 ghz en la coexistencia con redes wlan*. RIOBAMBA, ECUADOR. Obtenido de <http://dspace.esPOCH.edu.ec/bitstream/123456789/4054/1/20T00454.pdf>
- Marlene, M. C. (2016). *Problemas de salud en los trabajadores expuestos a fungicidas de la Florícola Rosas*. Ibarra.
- Martínez, F. R., & Meré, J. O. (2014). *Redes Inalámbricas de sensores: teoría y aplicación práctica*. Universidad de la Rioja.
- Microsoft.com. (2021). *Protocolos y tecnologías de IoT*. Obtenido de <https://azure.microsoft.com/es-es/overview/internet-of-things-iot/iot-technology-protocols/>

- Microsoft.com. (7 de Diciembre de 2021). *Protocolos y tecnologías de IoT*. Obtenido de <https://azure.microsoft.com/es-es/overview/internet-of-things-iot/iot-technology-protocols/>
- MIGUEL, P. A. (2016). *Electrónica aplicada 2.ª edición*.
- Molina, C. M. (2014). *Políticas integrales para mejorar la calidad del aire en la ZMVN programa hoy no circula*. doi:<https://doi.org/10.1787/9789264210448-en>
- Monatañana, R. (2012). *Redes inalámbricas 802.11*.
- Morera, P. B., & Araya, C. C. (2017). Intoxicación por monóxido de carbono. *Med. leg. Costa Rica vol.34*.
- Municipio del Distrito Metropolitano de Quito. (2017). *INFORME ANUAL 2017 • LA CALIDAD DEL AIRE EN QUITO*.
- Murphy, K. P. (2015). *Machine Learning A Probabilistic Perspective*. Obtenido de http://noiselab.ucsd.edu/ECE228/Murphy_Machine_Learning.pdf
- Novoa, A. K., & Suarez, D. M. (2019). *Revisión sistemática del estándar ISO / IEC 30141: 2018 como arquitectura de referencia para la seguridad en entornos IoT*. Bogotá D.C.
- Ogawa, N. (14 de Noviembre de 2018). *Society 5.0*. Obtenido de <https://www.eesc.europa.eu/en/news-media/presentations/society-50-co-creating-future>
- Organización Mundial de la Salud (OMS). (2019). Clasificación recomendada por la OMS de los plaguicidas por el peligro que presentan y Directrices para la clasificación. *WHO recommended*.

Organización Mundial de la Salud (OMS). (22 de Septiembre de 2021). *Contaminación del aire ambiente (exterior)*. Obtenido de [https://www.who.int/es/news-room/fact-sheets/detail/ambient-\(outdoor\)-air-quality-and-health](https://www.who.int/es/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health)

Organización Mundial de la Salud. (2017). *Prevención de los riesgos para la salud derivados del uso de plaguicidas en la agricultura*. Obtenido de https://www.who.int/occupational_health/publications/es/pwh1sp.pdf

Organización Mundial de la Salud. (2021). *Contaminación del aire de interiores y salud*. Obtenido de <https://www.who.int/es/news-room/fact-sheets/detail/household-air-pollution-and-health>

Orozco, J., & Siles, G. (2019). Radio study and problems in a WiFi network with high user density. *Acta Nova*, 32-52. Obtenido de http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S1683-07892019000100003&lng=es&tlng=es

Perez, S. A. (2015). *ESTUDIO COMPARATIVO DE LOS SISTEMAS DE GESTIÓN Y MONITOREO BASADOS EN LOS REQUERIMIENTOS GENERALES DE LA RED DE UN CAMPUS UNIVERSITARIO*. PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR, Quito.

Prayashi, B., & Hemant, P. (2015). Understanding and formulation of various kernel techniques for support vector machines. *IEEE*, 1-6.
doi:10.1109/ICCIC.2015.7435804

Ramírez, C. L. (2016). *Diseño de una arquitectura para redes de sensores con soporte para aplicaciones de detección de eventos*.

RFC-2271. (s.f.). SNMPv3 Architecture.

- Ribes, R. C. (2013). Redes locales. Obtenido de <https://elibro.net/es/ereader/utnorte/43257?page=258>
- Romero, M. d. (2017). Sistemas Avanzados de Comunicaciones Gestión de Redes. *Dpto. de Tecnología Electrónica.*
- S., P., Mallapur, J. D., & Bendigeri, K. Y. (2017). Remote Sensing and Controlling of Greenhouse Agriculture Parameters based on IoT. *IEEE Vishwakarma Institute of Technology*, 20-22.
- S.Tanenbaum, A., & Wetherall, D. J. (2017). *Redes de computadoras. Séptima Edición.* Pearson.
- Salcedo, L. (31 de Enero de 2018). *Introducción al Machine Learning #9 - K Vecinos más cercanos (Clasificación y Regresión)*. Obtenido de <https://pythondiario.com/2018/01/introduccion-al-machine-learning-9-k.html>
- Sosa, J. V. (2016). *SNMP*. CINVESTAV-Tamaulipas. Recuperado el 15 de Noviembre de 2021, de <https://www.tamps.cinvestav.mx/~vjsosa/clases/redes/SNMP.pdf>
- Stallings, W. (1998). SNMP and SNMPv2: The Infrastructure for Network Management. *IEEE* , 37-43.
- Stallings, W. (2014). Data .
- Stallings, W. (2014). *Data and Computer Communications Tenth Edition.*
- Subramanian, d. M., Gonsalves, T. A., & Rani, N. U. (2015). *Network Management: Principles and Practice* (Vol. II). Pearson Education India.
- Tanenbaum, A. S., & Wetherall, D. J. (2012). *Redes de computadoras*. Naucalpan de Juárez: Pearson.

The MathWorks, I. (2021). Support vector machine (SVM). Obtenido de <https://la.mathworks.com/discovery/support-vector-machine.html>

Tituaña, R. A. (2016). *SISTEMA DE GESTIÓN DE RED BASADO EN EL MODELO FUNCIONAL SNMP DE LA IETF PARA MONITOREAR LOS RECURSOS DE LA RED LAN EN EL EDIFICIO DE EMAPA-I DE LA CIUDAD DE IBARRA*. Ibarra.

Venegas, E. V., Cortés, S. L., Baltazar, R. G., & Serrano, M. P. (2016). Exposición laboral a plaguicidas y efectos en la salud de trabajadores florícolas de Ecuador.

Villavicenci, C. C., & Carrión, C. G. (2021). El sector florícola del Ecuador y su aporte a la Balanza Comercial Agropecuaria: período 2009 – 2020. *Revista Científica y Tecnológica UPSE*, 74-82.

Yigit, H. (2013). classifier, A weighting approach for KNN.
doi:10.1109/ICECCO.2013.6718270

ZAMBRANO, O. P. (2016). *CARACTERIZACION DE MATERIAL PARTICULADO, PLOMO Y ARSENICO PARA LA EVALUACION DE LA PLOMO Y ARSENICO PARA LA EVALUACION DE LA MATARANI*". AREQUIPA – PERU: UNIVERSIDAD NACIONAL DE SAN AGUSTIN.

ANEXOS

ANEXO A. Entrevista realizada al ingeniero agropecuario DE LA CRUZ

PEDRO jefe de la finca PLANTEC.

| REGISTRO DE ENTREVISTA | |
|-------------------------|-----------------------|
| Nombre del entrevistado | PEDRO DE LA CRUZ |
| Cargo: | JEFE DE FINCA PLANTEC |
| Fecha: | 28/5/2022 |

| |
|--|
| Pregunta 1. ¿Cuál es la frecuencia con la que se realiza la fumigación en un invernadero? |
| 3 VECES POR SEMANA |
| Pregunta 2. ¿Cuáles son los horarios más comunes de fumigación? |
| DESDE LAS 7 AM HASTA LAS 12 AM |
| Pregunta 3. ¿Qué plaguicidas se utilizan frecuentemente al momento de fumigar un invernadero? |
| FUNGICIDAS 2 VECES POR SEMANA. PROPAMOCARB, INSECTICIDAS 1 VEZ POR SEMANA. SPINOSAD |
| Pregunta 4. ¿Cuál es el tiempo estimado durante la fumigación en un bloque? |
| 1 HORA |
| Pregunta 5. ¿Cuál es el tiempo estimado para ingresar después de realizar la fumigación? |
| |

DEPENDE DE LAS INSTRUCCIONES DEL FABRICANTE, CON RESPECTO A LA SALUD. 6 HORAS, 12 HORAS, 24 HORAS.

Pregunta 6. La plantación: ¿posee algún sistema electrónico que permita medir la concentración de diferentes partículas tóxicas, antes de ingresar al invernadero?

NO, SOLAMENTE SE BASA A LOS LIMITES PROPORCIONADOS EN LAS ETIQUETAS DEL LOS FABRICANTES

Pregunta 7. ¿Usted cree que sea necesario implementar un sistema electrónico para medir las diferentes concentraciones de partículas tóxicas en un invernadero?

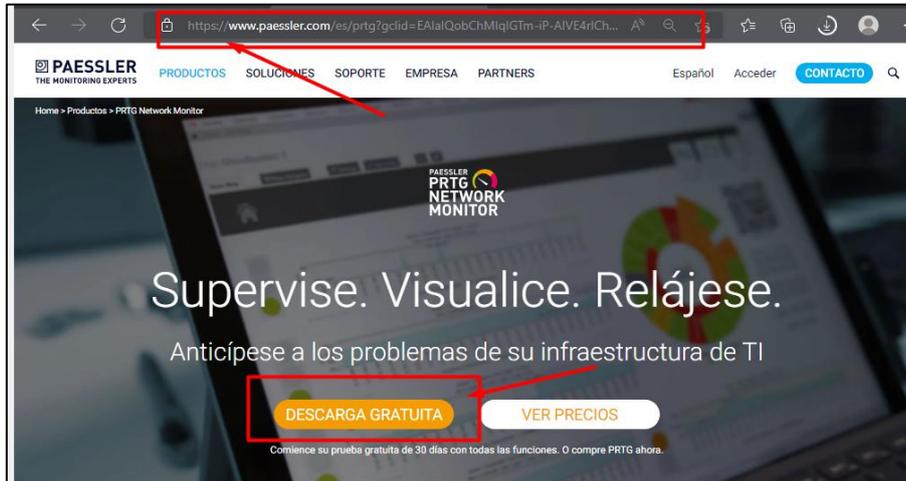
SI, Y ADEMÁS EL SISTEMA DEBERÍA SER FÁCIL INSTALACIÓN Y DE BAJO COSTO DE IMPLEMENTACIÓN.

Pregunta 8. ¿Cuáles características operacionales cree usted que debería poseer este sistema electrónico de alerta y monitoreo de partículas tóxicas?

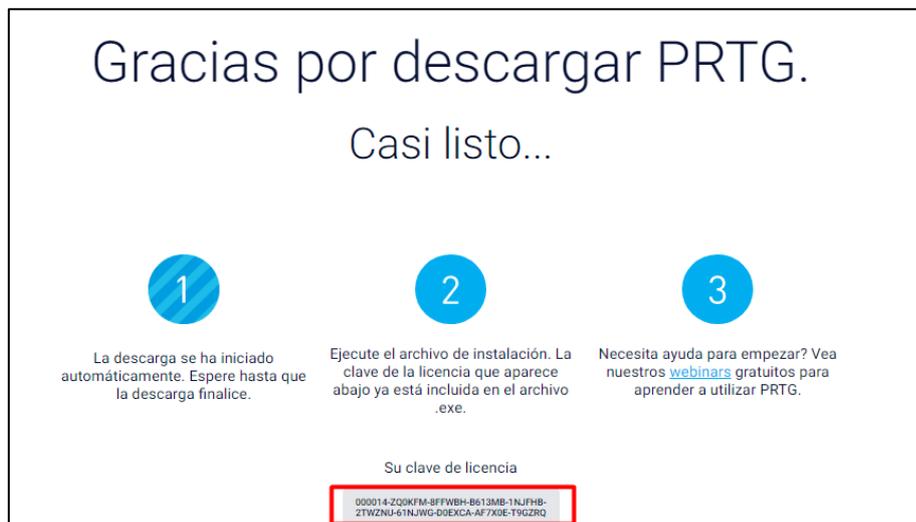
ESTE SISTEMA DEBERÍA MEDIR, EL TIPO DE INGREDIENTE ACTIVO, LA CONCENTRACIÓN DEL INGREDIENTE ACTIVO, TENER UNA VISUALIZACIÓN FÁCIL PARA EL USUARIO Y ALERTAR CUANDO SE SUPERE LOS LIMITES PERMITIDOS.

ANEXO B. manual de instalación de PRTG NETWORK MONITOR

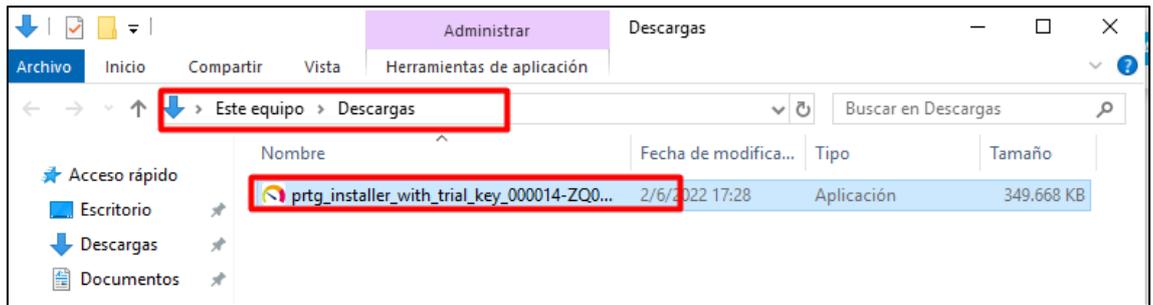
1. Se inicia ingresando a la página principal de PRTG NETWORK MONITOR y clic en descarga gratuita.



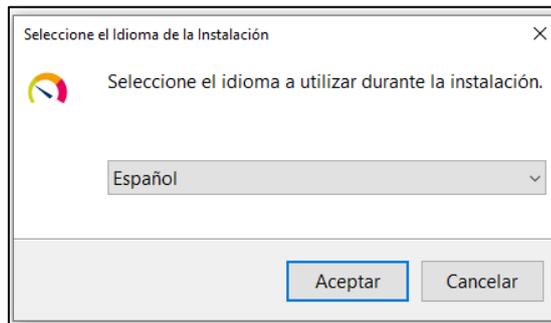
2. La descarga empezara y proporciona la clave de licencia, se debe copiar y guardar para el posterior uso



3. Al acabar la descarga, en la carpeta descargas del ordenador se encuentra el archivo, se lo ejecuta.



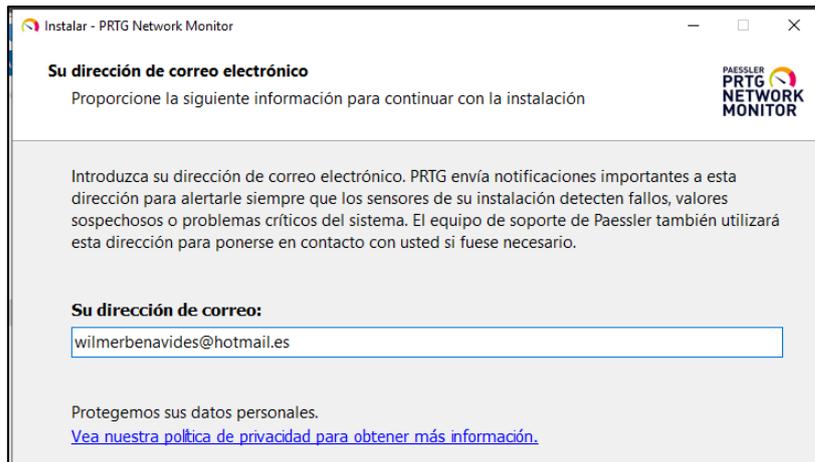
4. Se selecciona el idioma



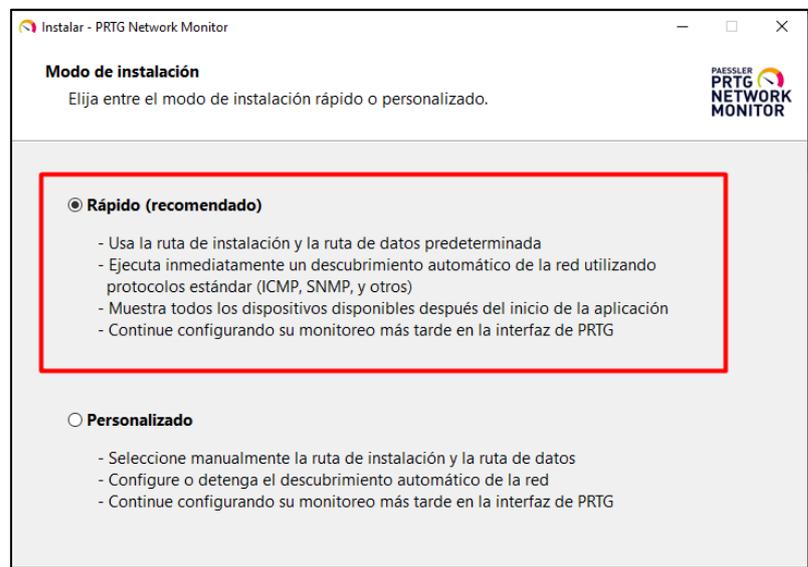
5. Se acepta los términos y condiciones



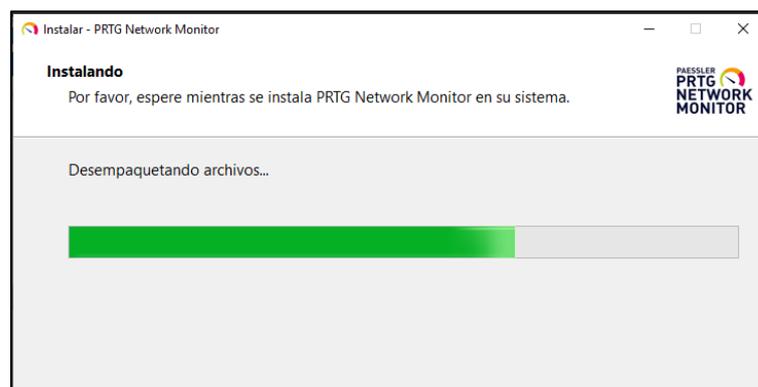
6. Se ingresa el correo del administrador



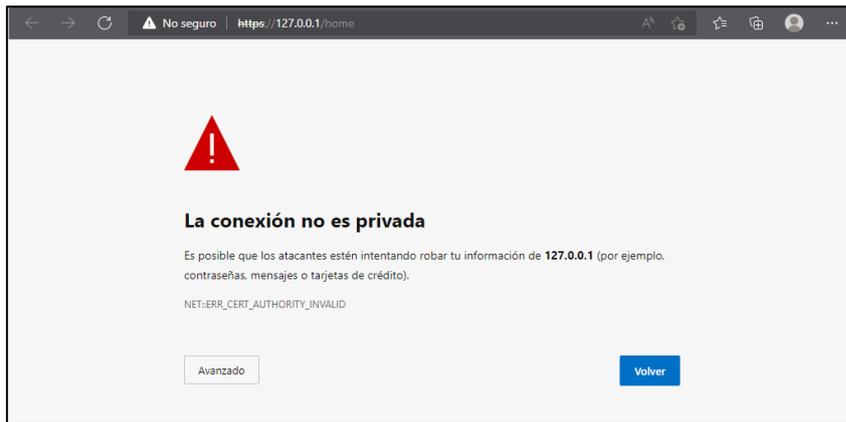
7. Se selecciona el modo recomendado para la instalación



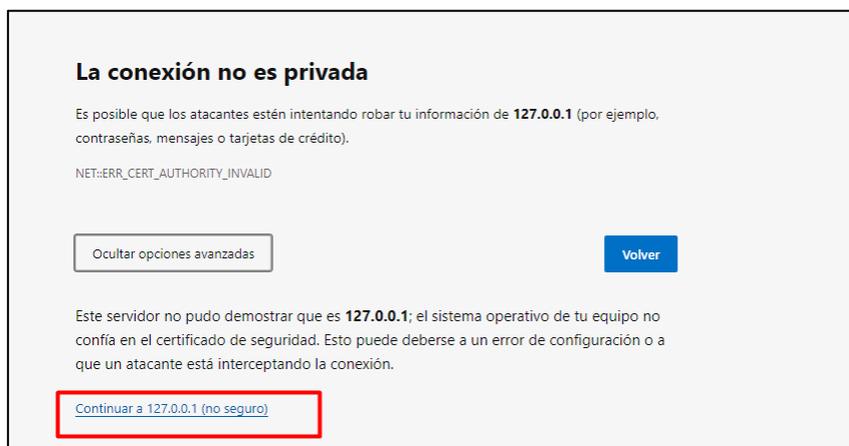
8. Se procede a instalar



9. Una vez instalado se procede a ejecutar el servidor y se abre el navegador asignando la dirección de localhost.



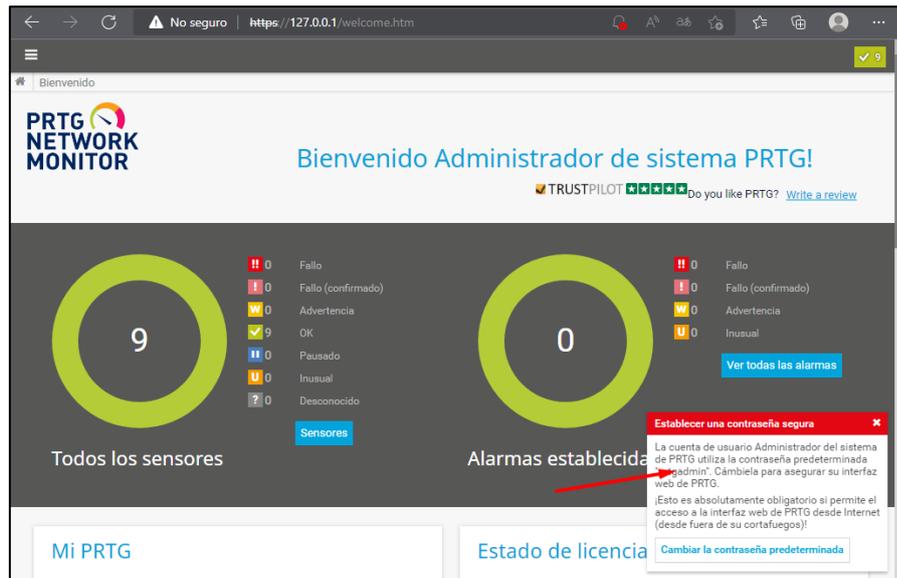
10. Damos en avanzado y continuar



11. Se despliega el login del servidor con los parámetros de autenticación por defecto

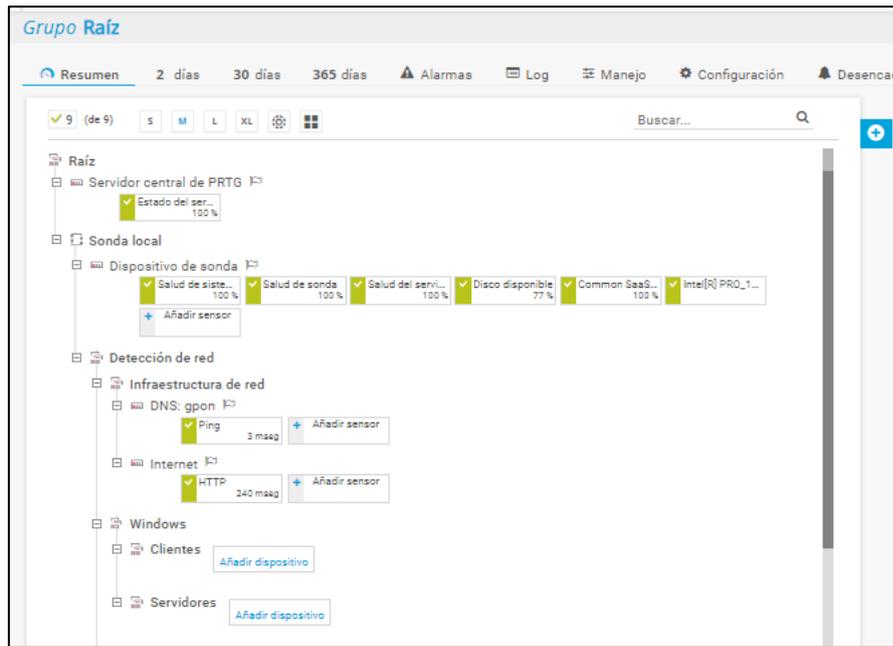


12. Se despliega la ventana principal de PRTG y se notifica que se cambie la contraseña a una segura

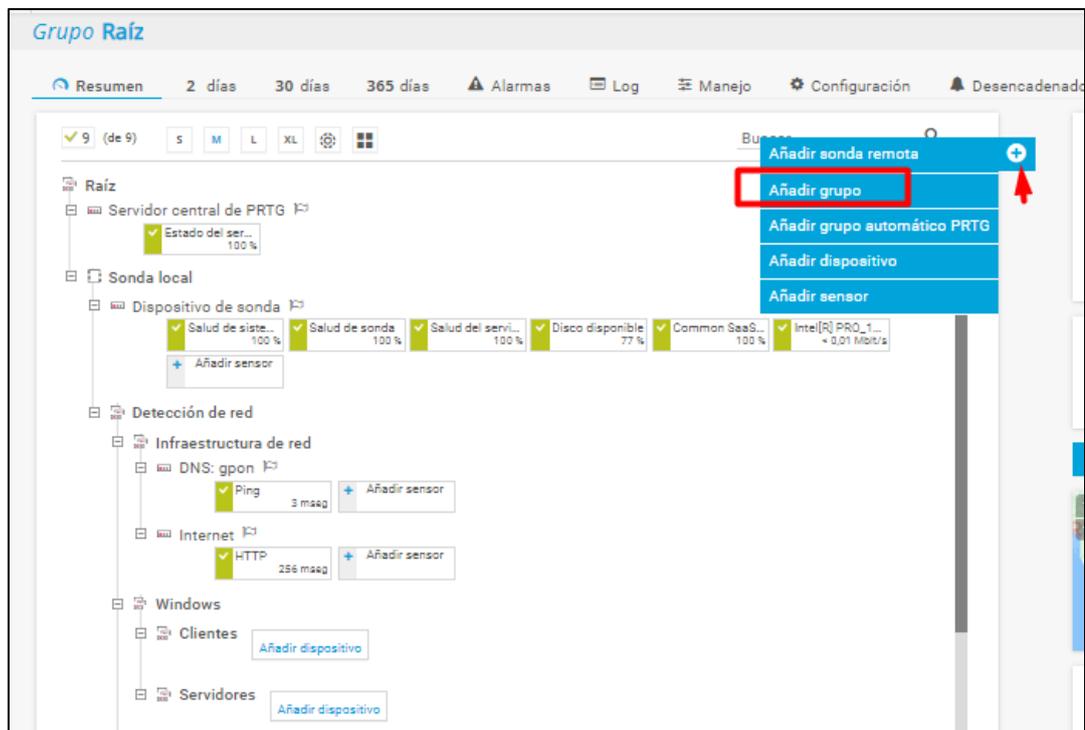


13. Se cambia el usuario de inicio y la contraseña

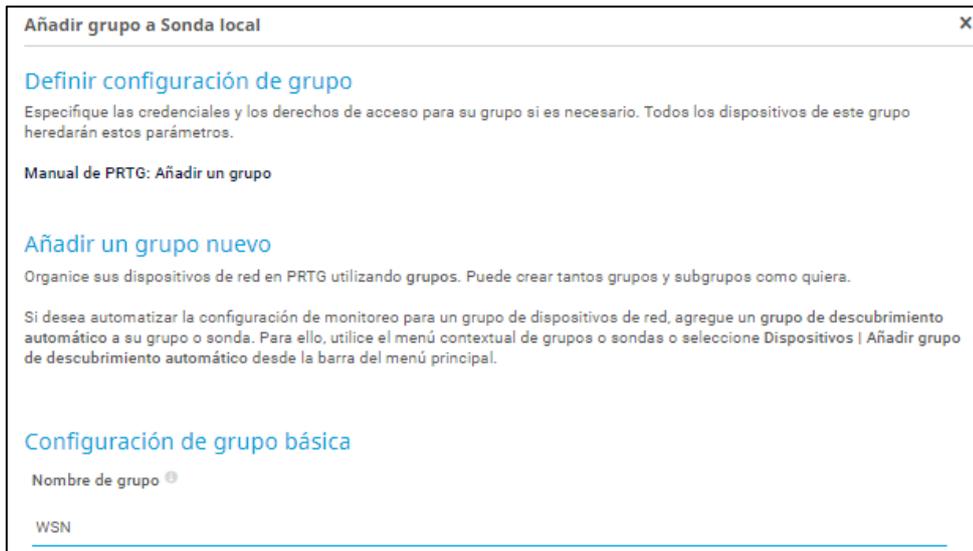
14. En la ventana principal se puede ver el grupo raíz donde se ha generado automáticamente sensores para el servidor denominado sonda local



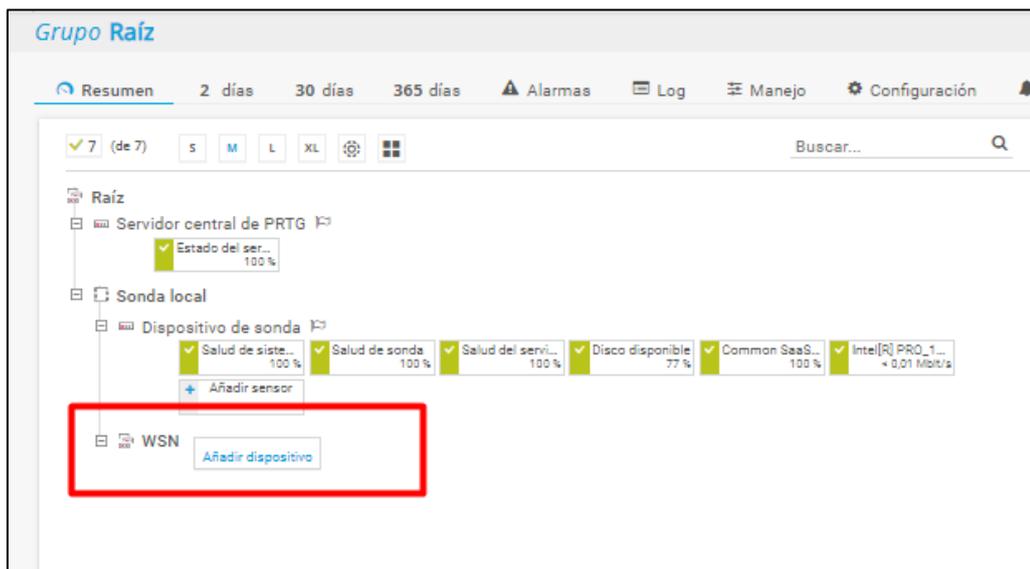
15. Se procede a generar el grupo denominado como WSN, en la parte superior derecha existe está el icono para agregar grupos, dispositivos y sensores, clic en añadir grupo



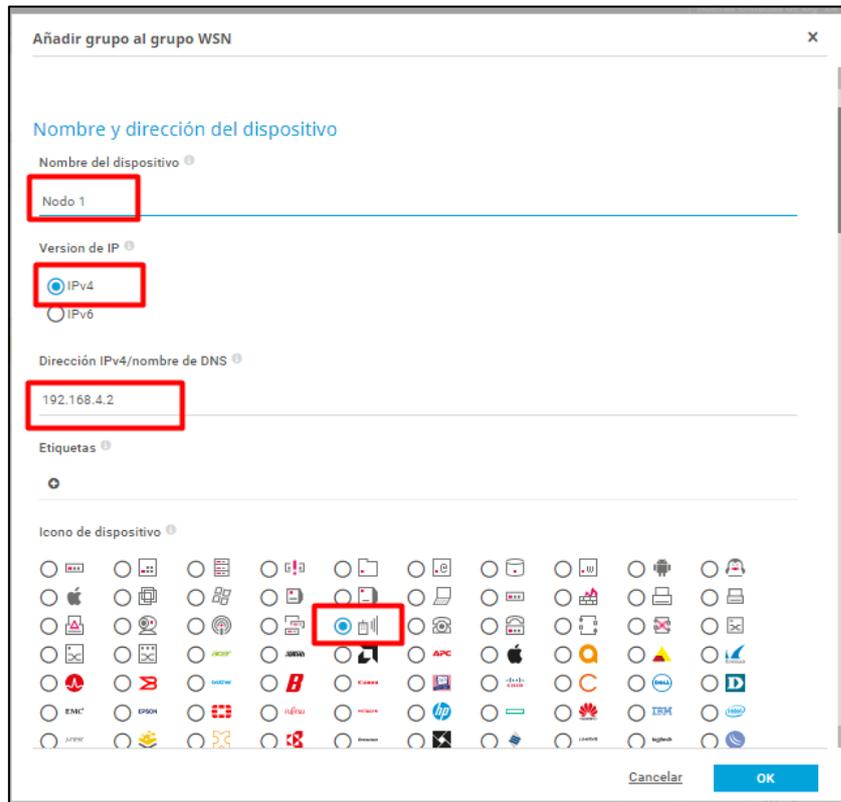
16. Se asigna el nombre al grupo



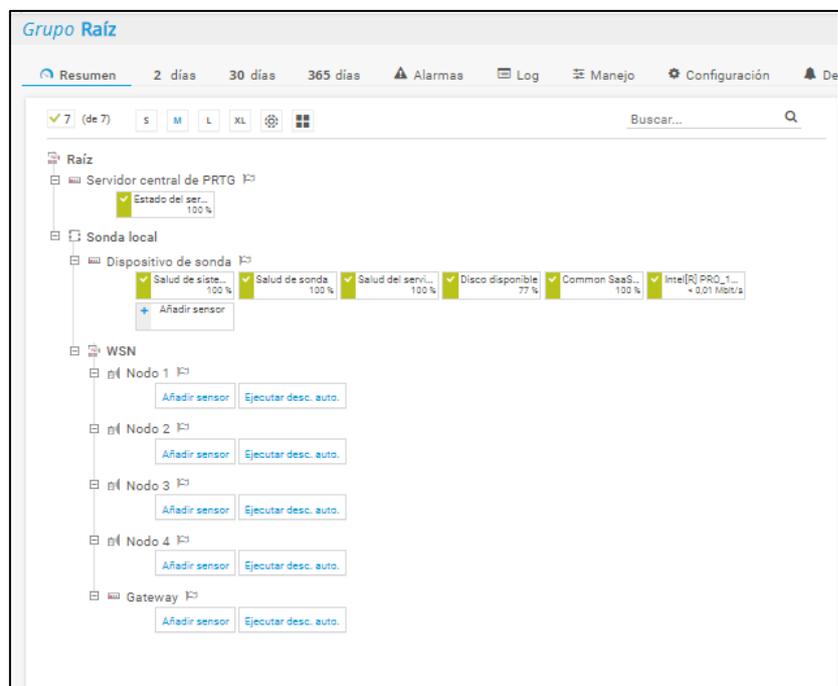
17. Se crea el grupo y se procede a agregar dispositivos, en este caso corresponde a nodo 1, nodo 2, nodo 3 y Gateway



18. Se asigna el nombre, si se va a trabajar con IPv4 o IPv6, la dirección IPv4 y un icono para cada dispositivo



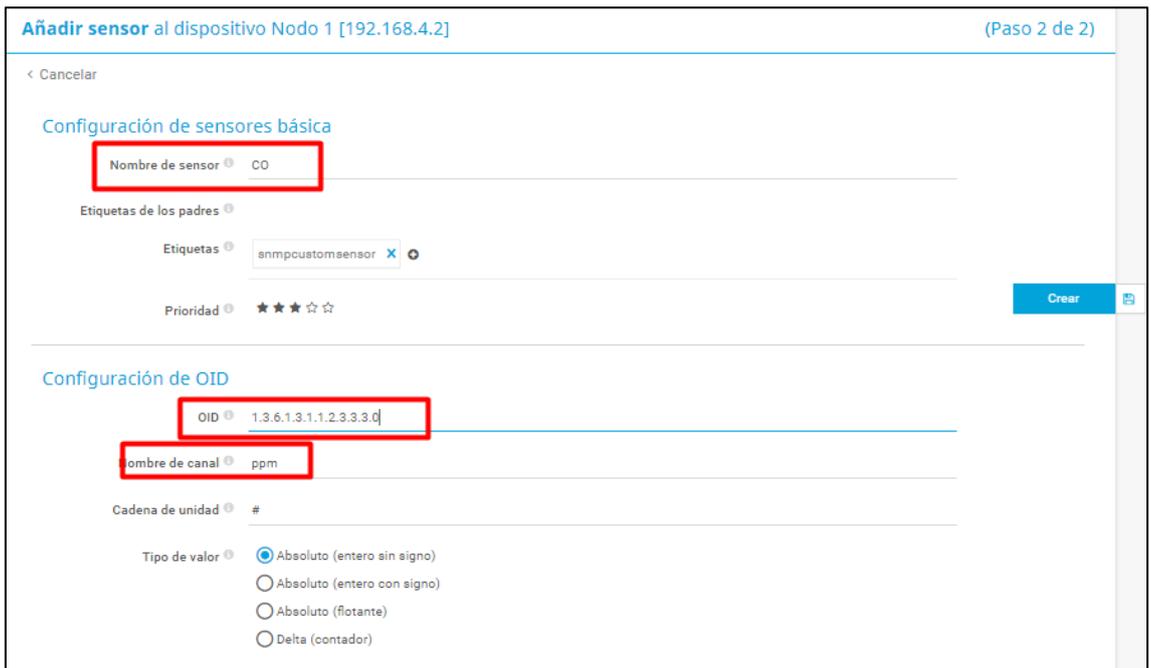
19. Se ejecuta el mismo proceso para todos los dispositivos



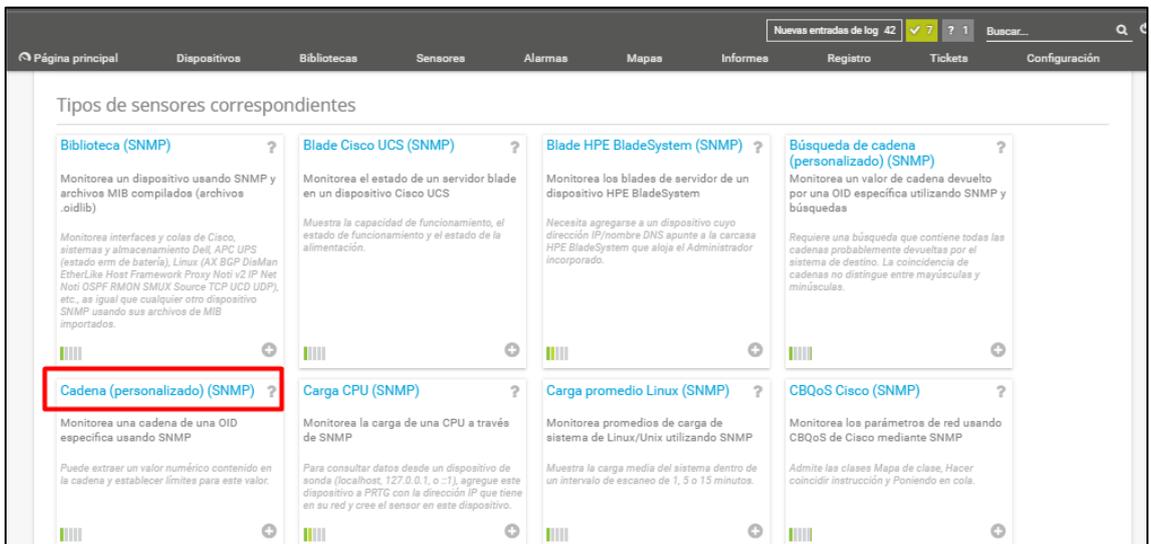
20. Se procede a asignar sensores a cada dispositivo, en este caso se debe tener en cuenta el OID de cada objeto de la MIB a monitorear, se da clic en agregar sensor y se busca SNMP personalizado para los OID con valor tipo Integer



21. Y se procede a ingresar el nombre del sensor el OID y el tipo de valor en el nombre del canal.



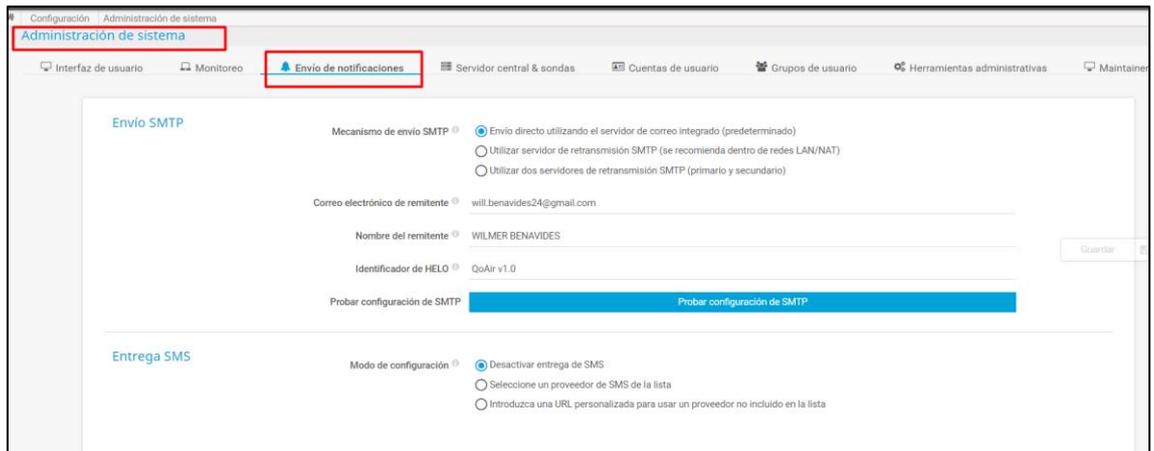
22. Para los OID con valor cadena de texto o String se selecciona cadena personalizada SNMP



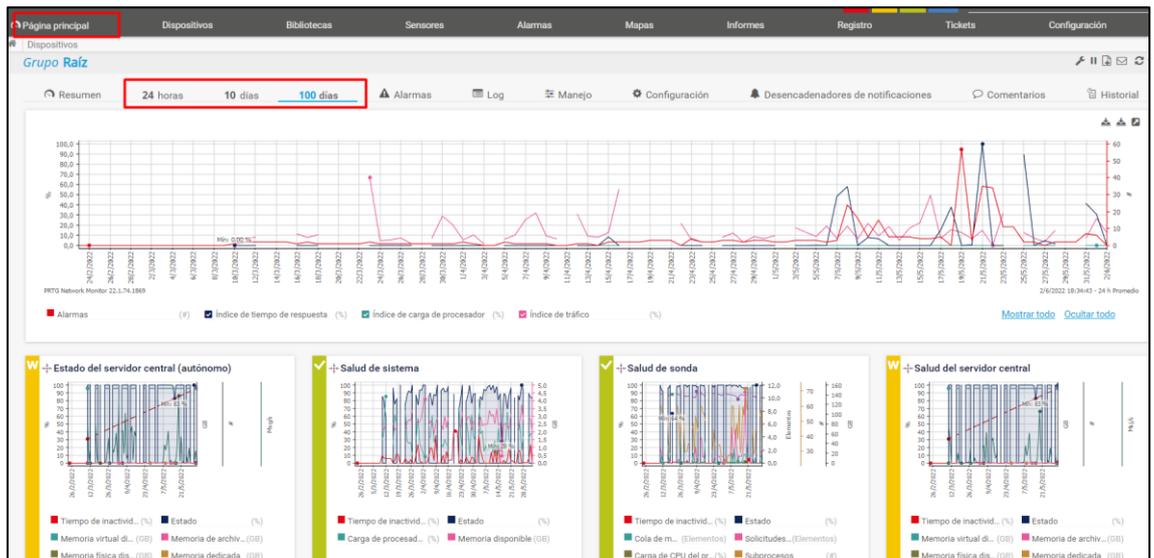
23. Se asigna el nombre al sensor y el OID

24. Y de esta manera se va agregando solo los objetos que se desee monitorear

25. Para las configuraciones de notificar al correo o por SMS se ingresa a administración de sistema y envío de notificaciones, se agrega el correo a cuál se quiere se enví las notificaciones.



26. Para acceder a la base de datos, se debe ir a página principal y seleccionar el resumen de 24 horas, 10 días o 100 días. Se puede apreciar que los datos se presentan en manera de histogramas, cada sensor de cada dispositivo se va guardando.



27. Se puede escoger el sensor que se desea analizar, para determinar sus variaciones en el tiempo.

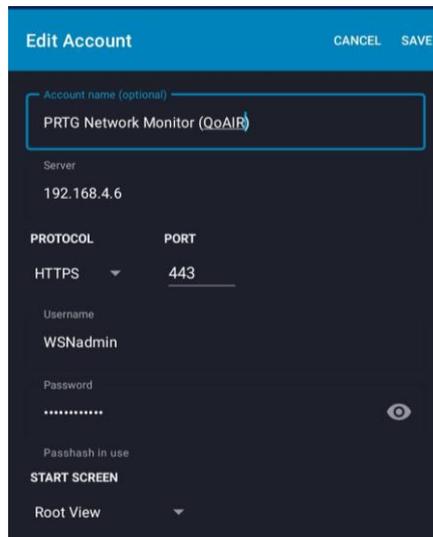


28. Para poder acceder a la aplicación móvil de PRTG NETWORK MONITOR se procede a ingresar a configuraciones y descargas opcionales.

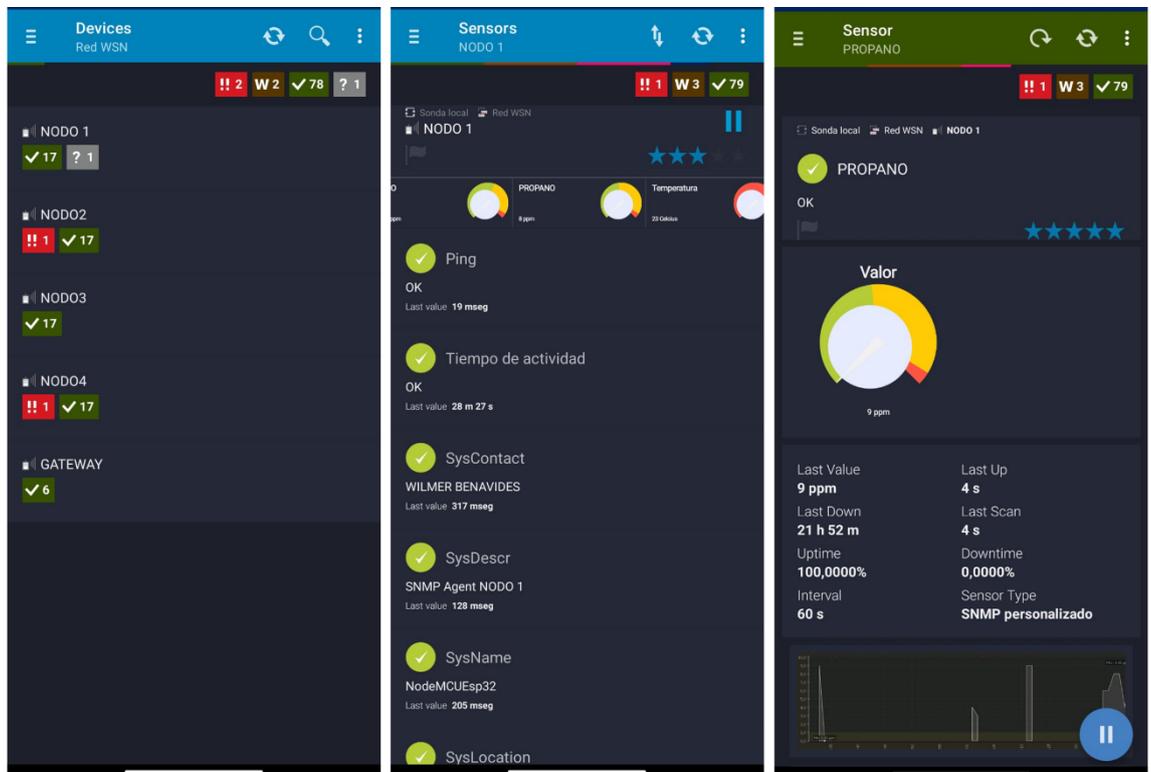
29. Se despliega un código QR el cual al escanearlo con el dispositivo móvil le direcciona a PLAY STORE

Se descarga y se instala, al abrir la aplicación se despliega el editor de cuentas, donde se puede acceder al servidor asignando un nombre de usuario, la dirección

IPv4 del servidor, el protocolo, el nombre de usuario y contraseña que se creó en el servidor.



30. Y se accede al servidor desde la aplicación móvil, donde se puede observar todo el grupo con los dispositivos y sensores agregados. Además, que se activan automáticamente las notificaciones cuando se notifiquen al servidor.



ANEXO C. Sketch de Arduino del nodo sensor

```
//+++++
//                                UNIVERSIDAD TÉCNICA DEL NORTE                +
//                                FICA-CIERCOM                                  +
//                                PROYECTO DE TITULACIÓN                       +
//                                WILMER BENAVIDES                             +
//+++++
/*****librerías de módulo de comunicación Wi-Fi*****/
#if defined (ESP8266)
#include <ESP8266WiFi.h>           // ESP8266 Core WiFi Library
#else
#include <WiFi.h>                 // ESP32 Core WiFi Library
#endif

/*****matriz de entrenamiento para algoritmo KNN*****/
#include "matrizEntrenamiento.h"// matriz de entrenamiento para algoritmo KNN

/*****Librerías para el protocolo SNMP*****/
#include <WiFiUdp.h> // Libreria para manejo del protocolo UDP
#include <SNMP_Agent.h> // libreria para el agente SNMP
#include <SNMPTrap.h> // libreria para los traps SNMP

/*****librerías MYSQL*****/
#include <MySQL_Connection.h>
#include <MySQL_Cursor.h>
#define FORMAT_LITTLEFS_IF_FAILED true

/*****Librerías para sensores*****/
#include <DHT.h> // libreria para sensor de temperatura y humedad relativa
#include <MQUnifiedsensor.h>
/*****Distribucion de pines*****/
#define Placa ("ESP-32")
#define PinMQ135 (39)
#define PinMQ2 (34)
#define PinMQ7 (35)
#define PinDHT11 (32)
#define led (16) // led indicador de funcionamiento

/*****Distribución de parámetros para sensores*****/
#define Type ("MQ-SENSORS")
#define Voltage_Resolution (5)
#define ADC_Bit_Resolution (12) // For esp32
#define RatioMQ2CleanAir (9.83) //RS / R0 = 9.83 ppm
#define RatioMQ7CleanAir (27.5) //RS / R0 = 27.5 ppm
#define RatioMQ135CleanAir (3.6) //RS / R0 = 3.6 ppm
/*****Ingresar pin al método de sensor DHT*****/
MQUnifiedsensor MQ2(Placa, Voltage_Resolution, ADC_Bit_Resolution, PinMQ2, Type);
MQUnifiedsensor MQ7(Placa, Voltage_Resolution, ADC_Bit_Resolution, PinMQ7, Type);
MQUnifiedsensor MQ135(Placa, Voltage_Resolution, ADC_Bit_Resolution, PinMQ135, Type);
DHT dht(PinDHT11, DHT11);

/*****parámetros wifi*****/
const char* ssid = "GATEWAY-WSN";
const char* pass = "WSN_ciercom2022";
IPAddress ip(192, 168, 4, 4); // Dirección ip NODO 3
IPAddress gateway(192, 168, 4, 1);
IPAddress subnet(255, 255, 255, 0);

/*****MYSQL*****/
IPAddress server_addr(192, 168, 4, 6); // IP of the MySQL *server* here
char user[] = "root"; // MySQL user login username
char password[] = "admin123"; // MySQL user login password
WiFiClient client;
MySQL_Connection conn((Client *)&client);
MySQL_Cursor* cursor;

/*****configuración de permisos SNMP*****/
const char* rocommunity = "public"; // permiso de solo lectura a la comunidad
const char* rwcommunity = "public"; // permiso de lectura y escritura a la comunidad

/***** RFC1213-MIB*****/
// RFC1213-MIB
char* oidSysDescr = ".1.3.6.1.2.1.1.1.0"; // SysDescr
char* oidSysObjectID = ".1.3.6.1.2.1.1.2.0"; // SysObjectID
```

```

char* oidSysUptime = ".1.3.6.1.2.1.1.3.0"; // sysUptime (seconds)
char* oidSysContact = ".1.3.6.1.2.1.1.4.0"; // SysContact
char* oidSysName = ".1.3.6.1.2.1.1.5.0"; // SysName
char* oidSysLocation = ".1.3.6.1.2.1.1.6.0"; // SysLocation
char* oidSysServices = ".1.3.6.1.2.1.1.7.0"; // sysServices

/***** información de gestión MIB //RFC1213*****/
std::string sysDescr = "SNMP Agent NODO 3";
std::string sysObjectID = "1.3.6.1.2.1.experimental.WSN";
uint32_t sysUptime = 0;
std::string sysContactValue = " WILMER BENAVIDES";
std::string sysNameValue = " NodeMCUEsp32";
std::string sysLocationValue = "Invernadero # 1";
int sysServices = 2;

/***** ENTITY-MIB .1.3.6.1.3.1.1.1 - implementada en el nodo sensor de la wsn*****/
/*****.iso.org.dod.internet.experimental.wsn.nodo.sensores*****/
/***** OID/entidades para los Sensores cada nodo*****/
char* oidentNodeDescripcion = ".1.3.6.1.3.1.1.1.0"; // Descripción general del nodo
sensor
char* oidentNodeSwVersion = ".1.3.6.1.3.1.1.4.1.0"; // Versión del software instalada
char* oidentNodeSwFechaActualizacion = ".1.3.6.1.3.1.1.4.2.0"; // fecha de actualización
del software
char* oidentTipoSensorMQ135 = ".1.3.6.1.3.1.1.2.1.1.0"; // Tipo de sensor
char* oidentNameSensorMQ135 = ".1.3.6.1.3.1.1.2.2.1.0"; // Nombre del sensor
char* oidentValorSensorMQ135 = ".1.3.6.1.3.1.1.2.3.1.0"; // Valor del sensor en ppm
char* oidentStatusSensorMQ135 = ".1.3.6.1.3.1.1.2.4.1.0"; // Estado del sensor
char* oidentTipoSensorMQ7 = ".1.3.6.1.3.1.1.2.1.2.0"; // Tipo de sensor
char* oidentNameSensorMQ7 = ".1.3.6.1.3.1.1.2.2.2.0"; // Nombre del sensor
char* oidentValorSensorMQ7 = ".1.3.6.1.3.1.1.2.3.2.0"; // Valor del sensor en ppm
char* oidentStatusSensorMQ7 = ".1.3.6.1.3.1.1.2.4.2.0"; // Estado del sensor
char* oidentTipoSensorMQ2 = ".1.3.6.1.3.1.1.2.1.3.0"; // Tipo de sensor
char* oidentNameSensorMQ2 = ".1.3.6.1.3.1.1.2.2.3.0"; // Nombre del sensor
char* oidentValorSensorMQ2 = ".1.3.6.1.3.1.1.2.3.3.0"; // Valor del sensor en ppm
char* oidentStatusSensorMQ2 = ".1.3.6.1.3.1.1.2.4.3.0"; // Estado del sensor
char* oidentTipoSensorDTH11Temperatura = ".1.3.6.1.3.1.1.2.1.4.0"; // Tipo de sensor
char* oidentNameSensorDTH11Temperatura = ".1.3.6.1.3.1.1.2.2.4.0"; // Nombre del sensor
char* oidentValorSensorDTH11Temperatura = ".1.3.6.1.3.1.1.2.3.4.0"; // Valor de
Temperatura en Celcius
char* oidentStatusSensorDTH11Temperatura = ".1.3.6.1.3.1.1.2.4.4.0"; // Estado del
sensor
char* oidentTipoSensorDTH11HumedadRelativa = ".1.3.6.1.3.1.1.2.1.5.0"; // Tipo de sensor
char* oidentNameSensorDTH11HumedadRelativa = ".1.3.6.1.3.1.1.2.2.5.0"; // Nombre del
sensor
char* oidentValorSensorDTH11HumedadRelativa = ".1.3.6.1.3.1.1.2.3.5.0"; // Valor humedad
char* oidentStatusSensorDTH11HumedadRelativa = ".1.3.6.1.3.1.1.2.4.5.0"; // Estado del
sensor
char* oidentValorEtiqueta = ".1.3.6.1.3.1.1.1.1.0"; // Valor del sensor de humedad
relativa del ambiente

/***** asignación de valores variables para cada valor de cada entidad*****/
std::string entNodeDescripcion = "NodeMCUEsp32";
std::string entNodeSwVersion = "QoAIR.v1";
std::string entNodeSwFechaActualizacion = "03/marzo/2022";
std::string entTipoSensorMQ135 = "Sensor gas CO2";
std::string entNameSensorMQ135 = "MQ-135";
int entValorSensorMQ135 = 0;
std::string entStatusSensorMQ135 = " ";
std::string entTipoSensorMQ7 = "Sensor gas CO";
std::string entNameSensorMQ7 = "MQ-7";
int entValorSensorMQ7 = 0;
std::string entStatusSensorMQ7 = " ";
std::string entTipoSensorMQ2 = "Sensor gas H2";
std::string entNameSensorMQ2 = "MQ-2";
int entValorSensorMQ2 = 0;
std::string entStatusSensorMQ2 = " ";
std::string entTipoSensorDTH11Temperatura = "Sensor Temperatura";
std::string entNameSensorDTH11Temperatura = "DTH11";
int entValorSensorDTH11Temperatura = 0;
std::string entStatusSensorDTH11Temperatura = " ";
std::string entTipoSensorDTH11HumedadRelativa = "Sensor Humedad Relativa";
std::string entNameSensorDTH11HumedadRelativa = "DTH11";
int entValorSensorDTH11HumedadRelativa = 0;
std::string entStatusSensorDTH11HumedadRelativa = " ";
int entValorEtiqueta = 0;
int entPhySensorType_1 = 8; // Celsius
int entPhySensorScale_1 = 9; // Units

```

```

int entPhySensorPrecision_1 = 0;
int entPhySensorValue_1 = 0; // Value to be updated
int entPhySensorOperStatus_1 = 1; // OK
std::string entPhySensorUnitsDisplay_1 = "Celsius";
int entPhySensorValueTimeStamp_1 = 0;
int entPhySensorValueUpdateRate_1 = 0; // Unknown at declaration, set later.
/*****Variables globales para actualización de tiempo*****/
static const unsigned long UPTIME_UPDATE_INTERVAL = 1000; // ms = 1 segundos
static unsigned long lastUptimeUpdateTime = 0;
static const unsigned long SENSOR_UPDATE_INTERVAL = 5000; // ms = 5 Segundos
static unsigned long lastSensorUpdateTime = 0;
/***** SNMP Objects*****/
WiFiUDP udp;
SNMPAgent snmp = SNMPAgent(rocommunity, rwcommunity); // Creacion de la instancia para
el agente SNMP

/***** configuración para los SNMPTrap *****/
SNMPTrap* settableNumberTrap = new SNMPTrap("public", SNMP_VERSION_2C); // asignacion de
la comunidad y version de protocolo
SNMPTrap* settableNumberTrap1 = new SNMPTrap("public", SNMP_VERSION_2C);
SNMPTrap* settableNumberTrap2 = new SNMPTrap("public", SNMP_VERSION_2C);
char* changingString;
uint32_t tensOfMillisCounter = 0;
int changingNumber = 1;
int settableNumber = 0;
ValueCallback* changingNumberOID;
ValueCallback* settableNumberOID;
TimestampCallback* timestampCallbackOID;
/*****contadores TRAP *****/
int cont_NOx = 0;
int cont_CO = 0;
int cont_PROPANO = 0;
int cont_TEMP = 0;
int cont_HUM = 0;
/***** métodos declarados *****/
void addrRFC1213MIBHandler();
void addENTITYMIBHandler();
int readFakeSensor();
uint32_t getUptime();
uint64_t uptimeMillis();

/***** método de comunicación de Wi-Fi *****/
void initWiFi() {
    WiFi.mode(WIFI_STA);
    WiFi.config(ip, gateway, subnet);
    WiFi.begin(ssid, pass);
    Serial.print("Connecting to WiFi ..");
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print('.');
        delay(1000);
    }
    Serial.print("Connected to SSID: "); // imprimir el nombre de la red conectada
    Serial.println(ssid);
    Serial.print("IP address: "); // imprimir la direccion ip asignada
    Serial.println(WiFi.localIP());
}

void setup()
{
    /***** Comunicación serial *****/
    Serial.begin(115200);
    while (!Serial);

    /***** led de aviso de funcionamiento *****/
    pinMode(led, OUTPUT);

    /***** Habilitar librerías DHT *****/
    dht.begin();

    /***** llamar a método de conexión WIFI *****/
    initWiFi();

    /***** establecimiento de conexión con el servidor MYSQL *****/
    Serial.print("Connecting to SQL... ");
    if (conn.connect(server_addr, 3306, user, password)) {
        Serial.println("OK.");
    }
}

```

```

else {
    Serial.println("FAILED.");
}
/***** configuraciones para SNMP TRAP*****/
trap_conf();

/*****
// INICIA CALIBRACIÓN DE SENSORES
*****/
calibracion();
/***** metodos de armado de cabeceras para mensajes SNMP*****/
addRFC1213MIBHandler(); // RFC1213-MIB (System)
addENTITYMIBHandler(); // ENTITY-MIB

/***** activar los snmpwalk*****/
snmp.sortHandlers();

/*****CARGAR SERVICIOS SNMP*****/
snmp.setUDP(&udp);
snmp.begin();
cursor = new MySQL_Cursor(&conn);
}

void loop()
{
/*****Inicio bucle SNMP *****/
snmp.loop();

/*****encendido del led para indicar el estado del nodo*****/
digitalWrite( led, HIGH);

/***** // lectura de datos con sensores*****/

// SENSOR MQ-135
MQ135.update();
int valor_NOx = MQ135.readSensor();
float valor_NOx1 = MQ135.readSensor();
char valor_NOx2[10];
dtostrf(valor_NOx1, 5, 2, valor_NOx2);
entValorSensorMQ135 = valor_NOx;
// Serial.print("NOx:");
// Serial.print(valor_NOx2);
// Serial.print("ppm");
// Serial.print(" ");

/*****
// SENSOR MQ- 7
MQ7.update();
int valor_CO = MQ7.readSensor();
float valor_CO1 = MQ7.readSensor();
char valor_CO2[10];
dtostrf(valor_CO1, 5, 2, valor_CO2);
entValorSensorMQ7 = valor_CO;
// Serial.print("\t CO:");
// Serial.print(valor_CO2);
// Serial.print("ppm");
// Serial.print(" ");

/*****
// SENSOR MQ- 2//
MQ2.update(); // Update data, the arduino will read the voltage from the analog pin
int valor_PROPANO = MQ2.readSensor();
float valor_PROPANO1 = MQ2.readSensor();
char valor_PROPANO2[10];
dtostrf(valor_PROPANO1, 5, 2, valor_PROPANO2);
entValorSensorMQ2 = valor_PROPANO;
// Serial.print("\t PROPANO:");
// Serial.print(valor_PROPANO2);
// Serial.print("ppm");
// Serial.print(" ");
/*****
*****
//SENSOR DE TEMPERATURA
entValorSensorDTH11HumedadRelativa = dht.readHumidity();
int valor_HUM = entValorSensorDTH11HumedadRelativa;
entValorSensorDTH11Temperatura = dht.readTemperature() - 10;

```

```

int valor_TEMP = entValorSensorDTH11Temperatura;
if (isnan(entValorSensorDTH11HumedadRelativa) ||
isnan(entValorSensorDTH11Temperatura))
{
    Serial.println("Failed to read from DHT sensor!");
    return;
}
// Serial.print("\t Humedad:");
// Serial.print(entValorSensorDTH11HumedadRelativa);
// Serial.print("%");
// Serial.print(" ");
// Serial.print("\t Temp:");
// Serial.print(entValorSensorDTH11Temperatura);
// Serial.print("° Celsius");
// Serial.println(" ");

//***** configuracion del TRAP
SNMP*****

if (sysUptime == 0) {
    IPAddress destinationIP = IPAddress(192, 168, 4, 5);
    //Serial.println("iniciando SNMP Trap .....");
    snmp.sendTrapTo(settableNumberTrap, destinationIP, true, 2, 5000);
    // Serial.println("Se envio SNMP Trap Nodo Rebooting");
}

if ( valor_NOx1 == 0) {
    IPAddress destinationIP = IPAddress(192, 168, 4, 5);
    //Serial.println("iniciando SNMP Trap .....");
    //delay (500);

    snmp.sendTrapTo(settableNumberTrap1, destinationIP, true, 2, 5000);
    //Serial.println("Se envio SNMP Trap SENSOR NO READ");

    cont_NOx = cont_NOx + 1;

    if (cont_NOx > 100) {
        snmp.sendTrapTo(settableNumberTrap2, destinationIP, true, 2, 5000);
        // Serial.println("Se envio SNMP Trap Sensor DOWN");
    }
}

if ( valor_CO1 == 0) {
    IPAddress destinationIP = IPAddress(192, 168, 4, 5);
    // Serial.println("iniciando SNMP Trap .....");
    // delay (500);

    snmp.sendTrapTo(settableNumberTrap1, destinationIP, true, 2, 5000);
    // Serial.println("Se envio SNMP Trap SENSOR NO READ");

    cont_CO = cont_CO + 1;

    if (cont_CO > 100) {
        snmp.sendTrapTo(settableNumberTrap2, destinationIP, true, 2, 5000);
        // Serial.println("Se envio SNMP Trap Sensor DOWN");
    }
}

if ( valor_PROPANO1 = MQ2.readSensor() == 0) {
    IPAddress destinationIP = IPAddress(192, 168, 4, 5);
    //Serial.println("iniciando SNMP Trap .....");
    //delay (500);

    snmp.sendTrapTo(settableNumberTrap1, destinationIP, true, 2, 5000);
    //Serial.println("Se envio SNMP Trap NO READ");
    cont_PROPANO = cont_PROPANO + 1;

    if (cont_PROPANO > 50) {
        snmp.sendTrapTo(settableNumberTrap2, destinationIP, true, 2, 5000);
        Serial.println("Se envio SNMP Trap Sensor DOWN");
    }
}

if ( entValorSensorDTH11Temperatura == 0) {
    IPAddress destinationIP = IPAddress(192, 168, 4, 5);
    //Serial.println("iniciando SNMP Trap .....");
}

```

```

//delay (500);

snmp.sendTrapTo(settableNumberTrap1, destinationIP, true, 2, 5000);
// Serial.println("Se envio SNMP Trap SENSOR NO READ");

cont_TEMP = cont_TEMP + 1;

if (cont_TEMP > 100) {
    snmp.sendTrapTo(settableNumberTrap2, destinationIP, true, 2, 5000);
    // Serial.println("Se envio SNMP Trap Sensor DOWN");
}
}
if ( entValorSensorDTH11HumedadRelativa == 0) {
    IPAddress destinationIP = IPAddress(192, 168, 4, 5);
    //Serial.println("iniciando SNMP Trap .....");
    //delay (500);

    snmp.sendTrapTo(settableNumberTrap1, destinationIP, true, 2, 5000);
    // Serial.println("Se envio SNMP Trap SENSOR NO READ");

    cont_HUM = cont_HUM + 1;

    if (cont_HUM > 100) {
        snmp.sendTrapTo(settableNumberTrap2, destinationIP, true, 2, 5000);
        // Serial.println("Se envio SNMP Trap Sensor DOWN");
    }
}
/*****configuración para el tiempo activo del nodo*****/
if (millis() - lastUptimeUpdateTime >= UPTIME_UPDATE_INTERVAL)
{
    lastUptimeUpdateTime += UPTIME_UPDATE_INTERVAL;
    sysUptime = getUptime();
}
if (millis() - lastSensorUpdateTime >= SENSOR_UPDATE_INTERVAL)
{
    lastSensorUpdateTime += SENSOR_UPDATE_INTERVAL;
    // entPhySensorValue_1 = readFakeSensor();
    entPhySensorValueTimeStamp_1 = sysUptime;
}

/***** algoritmo de clasificación supervisada KNN*****/
float matriz_prueba [3] = {entValorSensorMQ135, entValorSensorMQ7, entValorSensorMQ2}
;
int fil = 75 ;
int columna = 4;
int vecinos = 3;
int etiquetas = 3;
int col;
int fila = 0;
float sumatoria = 0; //Variable que alberga la sumatoria de diferencia de cuadrados
entre dos putnos
float distancia; // Variable que alberga la distancia Euclideana
float matriz_k [2][vecinos]; //Matriz que alberga los vecinos
int i = 0;
int j;
float etiqueta; // Valor que retorna del método Knn
float mayor = 0;
float aux ;
float aux1; // variables auxiliares para ordenamiento del vector
float matriz_etiquetas [2][etiquetas]; //Matriz que alberga las etiquetas
/*
Matriz de k-vecinos
[1][2][3]----->número de vecinos
[0.1][0.2][0.3]--> distancia
[2][1][1]----->asignación del vecino por Etiquetas
*/
*/
for (; i < vecinos; i++) { //Matriz que alberga el número de vecinos y se encera
    matriz_k[0][i] = 3000 + i;
    matriz_k[1][i] = 0;
    /*
        [1][2][3]
        [0][0][0]
        [3001][3002][3003]
    */
}

```

```

}
i = 0;
for (; i < etiquetas; i++) { //Asignar etiquetas y se encera el contador
    matriz_etiquetas[0][i] = i + 1;
    matriz_etiquetas[1][i] = 0;
    /*
        [1][2][3]
        [0][0][0]
    */
}
i = 0;
/*
    PASOS PARA K-NN
    leer cada fila de la matriz de entrenamiento
    distancia entre la fila de la matriz con la nueva instancia (vector de prueba)
    ordenar la matriz_K (determinar los k vecinos e menor instancia
    contar las etiquetas en la matrizK y asignar en matriz etiquetas
    elegir al mayor número de matriz etiquetas
    retorno la etiqueta resultante
*/
for (; fila < fil; fila++) { // ciclo de lectura de filas
    for (col = 0; col < columna - 1; col++) { //-1 por que ultima columna es la etiqueta
        sumatoria = sumatoria + pow(matriz_prueba[col] - matriz[fila][col], 2);
//Diferencia de cuadrados entre dos puntos
    }
    distancia = sqrt(sumatoria); //Variable que alberga la distancia Euclidiana
    sumatoria = 0; // reinicio de variable para siguiente lectura
    //comparacion de nuevo dato con solo la distancia mayor almacenada en matriz_k
    if (distancia < matriz_k[0][vecinos - 1]) {
        matriz_k[0][vecinos - 1] = distancia;
        matriz_k[1][vecinos - 1] = matriz[fila][columna - 1];
    }
    // ordenamiento del vector de menor a mayor utilizando el metodo "BURBUJA"
    for (i = 0; i < vecinos; i++) {
        for (j = i + 1; j < vecinos; j++) {
            if (matriz_k[0][i] > matriz_k[0][j]) {
                aux = matriz_k[0][i];
                matriz_k[0][i] = matriz_k[0][j];
                matriz_k[0][j] = aux;
                //etiquetas
                aux1 = matriz_k[1][i];
                matriz_k[1][i] = matriz_k[1][j];
                matriz_k[1][j] = aux1;
            }
        }
    }
}
//Asignacion de la etiqueta
for (i = 0; i < etiquetas; i++) {
    for (j = 0; j < vecinos; j++) {
        if (matriz_etiquetas[0][i] == matriz_k[1][j]) {
            matriz_etiquetas[1][i] = 1 + matriz_etiquetas[1][i];
        }
    }
}

for (i = 0; i < etiquetas; i++) {
    if (matriz_etiquetas[1][i] > mayor) { //Condición para evaluar y asignar la etiqueta
        mayor = matriz_etiquetas[1][i];
        etiqueta = matriz_etiquetas[0][i]; //Variable que retorna la función KNN
    }
}
entValorEtiqueta = etiqueta;
int valor_etiqueta = entValorEtiqueta;
// Serial.println (etiqueta) ;
// delay(5000);

/*****Agregar datos en buffer para enviar a la base
de datos *****/
char INSERT_SQL[] = "INSERT INTO QoADB.ValNODO3 (NOx, CO, PROPANO, TEMPERATURA,
HUMEDAD, ETIQUETA) VALUES (%s,%s,%s,%d,%d,%d)";
char buffer[128];

sprintf(buffer, INSERT_SQL, valor_NOx2, valor_CO2, valor_PROPANO2, valor_TEMP,
valor_HUM, valor_etiqueta);

if (conn.connected()) {

```

```

        cursor->execute(buffer);
        // Serial.println (buffer) ;
        // delay(1000);
    }
}
int readFakeSensor()
{
    int min = -50;
    int max = 100;
    return min + esp_random() % ((max + 1) - min);
}

#ifdef ESP32
uint64_t uptimeMillis()
{
    return (esp_timer_get_time() / 1000);
}
#else
uint64_t uptimeMillis()
{
    static uint32_t low32, high32;
    uint32_t new_low32 = millis();
    if (new_low32 < low32)high32++;
    low32 = new_low32;
    return (uint64_t)high32 << 32 | low32;
}
#endif

uint32_t getUptime()
{
    return (uint32_t)(uptimeMillis() / 10); // Convercion de milisegundos a segundos
}

void addRFC1213MIBHandler()
{
    // Add SNMP Handlers of correct type to each OID
    snmp.addReadOnlyStaticStringHandler(oidSysDescr, sysDescr);
    snmp.addReadOnlyStaticStringHandler(oidSysObjectID, sysObjectID);
    snmp.addIntegerHandler(oidSysServices, &sysServices);
    snmp.addTimestampHandler(oidSysUptime, &sysUptime);
    // Add Settable Handlers
    snmp.addReadOnlyStaticStringHandler(oidSysContact, sysContactValue);
    snmp.addReadOnlyStaticStringHandler(oidSysName, sysNameValue);
    snmp.addReadOnlyStaticStringHandler(oidSysLocation, sysLocationValue);
}

void addENTITYMIBHandler()
{
    snmp.addReadOnlyStaticStringHandler(oidentNodeDescripcion, entNodeDescripcion);
    snmp.addReadOnlyStaticStringHandler(oidentNodeSwVersion, entNodeSwVersion);
    snmp.addReadOnlyStaticStringHandler(oidentNodeSwFechaActualizacion,
entNodeSwFechaActualizacion);
    snmp.addReadOnlyStaticStringHandler(oidentTipoSensorMQ135, entTipoSensorMQ135);
    snmp.addReadOnlyStaticStringHandler(oidentNameSensorMQ135, entNameSensorMQ135);
    snmp.addIntegerHandler(oidentValorSensorMQ135, &entValorSensorMQ135);
    snmp.addReadOnlyStaticStringHandler(oidentStatusSensorMQ135, entStatusSensorMQ135);

    snmp.addReadOnlyStaticStringHandler(oidentTipoSensorMQ7, entTipoSensorMQ7);
    snmp.addReadOnlyStaticStringHandler(oidentNameSensorMQ7, entNameSensorMQ7);
    snmp.addIntegerHandler(oidentValorSensorMQ7, &entValorSensorMQ7);
    snmp.addReadOnlyStaticStringHandler(oidentStatusSensorMQ7, entStatusSensorMQ7);

    snmp.addReadOnlyStaticStringHandler(oidentTipoSensorMQ2, entTipoSensorMQ2);
    snmp.addReadOnlyStaticStringHandler(oidentNameSensorMQ2, entNameSensorMQ2);
    snmp.addIntegerHandler(oidentValorSensorMQ2, &entValorSensorMQ2);
    snmp.addReadOnlyStaticStringHandler(oidentStatusSensorMQ2, entStatusSensorMQ2);

    snmp.addReadOnlyStaticStringHandler(oidentTipoSensorDTH11Temperatura,
entTipoSensorDTH11Temperatura);
    snmp.addReadOnlyStaticStringHandler(oidentNameSensorDTH11Temperatura,
entNameSensorDTH11Temperatura);
    snmp.addIntegerHandler(oidentValorSensorDTH11Temperatura,
&entValorSensorDTH11Temperatura);
    snmp.addReadOnlyStaticStringHandler(oidentStatusSensorDTH11Temperatura,
entStatusSensorDTH11Temperatura);
}

```

```

    snmp.addReadOnlyStaticStringHandler(oidentTipoSensorDTH11HumedadRelativa,
entTipoSensorDTH11HumedadRelativa);
    snmp.addReadOnlyStaticStringHandler(oidentNameSensorDTH11HumedadRelativa,
entNameSensorDTH11HumedadRelativa);
    snmp.addIntegerHandler(oidentValorSensorDTH11HumedadRelativa,
&entValorSensorDTH11HumedadRelativa);
    snmp.addReadOnlyStaticStringHandler(oidentStatusSensorDTH11HumedadRelativa,
entStatusSensorDTH11HumedadRelativa);

    snmp.addIntegerHandler(oidentValorEtiqueta, &entValorEtiqueta);
}

void calibracion() {
  /*****CALIBRACIÓN MQ-2 PROPANO *****/

  /***/

  //Establezca un modelo matemático para calcular la concentración de PPM y el valor de
las constantes
  //-----MQ2-----

  MQ2.setRegressionMethod(1); // _PPM = a*ratio^b
  MQ2.setA(658.71); MQ2.setB(-2.168); // Configure la ecuación para calcular la
concentración del gas
  MQ2.init();
  /*
  Regresión exponencial:
  Gas | a | b
  H2 | 987.99 | -2.162
  LPG | 574.25 | -2.222
  CO | 36974 | -3.109
  Alcohol| 3616.1 | -2.675
  Propano| 658.71 | -2.168
  */
  /*
  // -----MQ7-----
  */

  MQ7.setRegressionMethod(1); // _PPM = a*ratio^b
  MQ7.setA(99.042); MQ7.setB(-1.518); // Configuración para concentración CO
  MQ7.init();
  /*
  Exponential regression:
  GAS | a | b
  H2 | 69.014 | -1.374
  LPG | 700000000 | -7.703
  CH4 | 6000000000000 | -10.54
  CO | 99.042 | -1.518
  Alcohol | 4000000000000000 | -12.35
  */
  /*
  //-----MQ-----/
  */

  MQ135.setRegressionMethod(1); // _PPM = a*ratio^b
  MQ135.setA(102.2); MQ135.setB(-2.473); // Configuración para concentración NH4
  MQ135.init();
  /*
  Exponential regression:
  GAS | a | b
  CO | 605.18 | -3.937
  Alcohol | 77.255 | -3.18
  CO2 | 110.47 | -2.862
  Toluen | 44.947 | -3.445
  NH4 | 102.2 | -2.473
  Aceton | 34.668 | -3.369
  */
  Serial.print("Calibrando ...");
  float calcROMQ135 = 0;
  for (int i = 1; i <= 10; i++)
  {
    MQ135.update(); // Update data, the arduino will read the voltage from the analog
pin
    calcROMQ135 += MQ135.calibrate(RatioMQ135CleanAir);
    Serial.print(".");
  }
}

```

```

MQ135.setR0(calcROMQ135 / 10);
Serial.println(" done!.");

//if(isinf(calcROMQ135)) {Serial.println("Warning: Conection issue, ROMQ135 is
infinite (Open circuit detected) please check your wiring and supply"); while(1);}
// if(calcROMQ135 == 0){Serial.println("Warning: Conection issue found, ROMQ135 is
zero (Analog pin shorts to ground) please check your wiring and supply"); while(1);}
/***** MQ CALibration *****/
MQ135.serialDebug(true);
/***** MQ CALibration *****/
float calcROMQ2 = 0;
for (int i = 1; i <= 10; i ++)
{
    MQ2.update(); // Update data, the arduino will read the voltage from the analog pin
    calcROMQ2 += MQ2.calibrate(RatioMQ2CleanAir);
    Serial.print(".");
}
MQ2.setR0(calcROMQ2 / 10);
Serial.println(" done!.");

//if(isinf(calcROMQ2)) {Serial.println("Warning: Conection issue, ROMQ2 is infinite
(Open circuit detected) please check your wiring and supply"); while(1);}
// if(calcROMQ2 == 0){Serial.println("Warning: Conection issue found, ROMQ2 is zero
(Analog pin shorts to ground) please check your wiring and supply"); while(1);}
/***** MQ CALibration *****/

MQ2.serialDebug(true);
/*****

float calcROMQ7 = 0;
for (int i = 1; i <= 10; i ++)
{
    MQ7.update(); // Update data, the arduino will read the voltage from the analog pin
    calcROMQ7 += MQ7.calibrate(RatioMQ7CleanAir);
    Serial.print(".");
}
MQ7.setR0(calcROMQ7 / 10);
Serial.println(" done!.");

//if(isinf(calcROMQ7)) {Serial.println("Warning: Conection issue, ROMQ7 is infinite
(Open circuit detected) please check your wiring and supply"); while(1);}
//if(calcROMQ7 == 0){Serial.println("Warning: Conection issue found, ROMQ7 is zero
(Analog pin shorts to ground) please check your wiring and supply"); while(1);}
/***** MQ CALibration *****/
MQ7.serialDebug(true);
}

void trap_conf() {
    timestampCallbackOID = (TimestampCallback*)snmp.addTimestampHandler(".1.3.6.1.3.1.0",
&sysUptime);
    settableNumberTrap->setUDP(&udp); // protocolo de uso para el envío del trap
    settableNumberTrap->setTrapOID(new OIDType(".1.3.6.1.3.1.2.1.0")); // OID del trap
    settableNumberTrap->setSpecificTrap(0); // número de mensaje del trap
    settableNumberTrap->setUptimeCallback(timestampCallbackOID); //Configuración del
contador de tiempo de actividad para usar en la trap
    settableNumberTrap->setIP(WiFi.localIP()); // agregar la dirección ip local
    settableNumberTrap->setInform(true);

    settableNumberTrap1->setUDP(&udp); // protocolo de uso para el envío del trap
    settableNumberTrap1->setTrapOID(new OIDType(".1.3.6.1.3.1.2.2.0")); // OID del trap
    settableNumberTrap1->setSpecificTrap(2); // número de mensaje del trap
    settableNumberTrap1->setUptimeCallback(timestampCallbackOID); //Configuración del
contador de tiempo de actividad para usar en la trap
    settableNumberTrap1->setIP(WiFi.localIP()); // agregar la dirección ip local
    settableNumberTrap->setInform(true);

    settableNumberTrap2->setUDP(&udp); // protocolo de uso para el envío del trap
    settableNumberTrap2->setTrapOID(new OIDType(".1.3.6.1.3.1.2.3.0")); // OID del trap
    settableNumberTrap2->setSpecificTrap(2); // número de mensaje del trap
    settableNumberTrap2->setUptimeCallback(timestampCallbackOID); //Configuración del
contador de tiempo de actividad para usar en la trap
    settableNumberTrap2->setIP(WiFi.localIP()); // agregar la dirección ip local
    settableNumberTrap->setInform(true);
}

```