



**UNIVERSIDAD CATÓLICA DE CUENCA**

*Comunidad Educativa al Servicio del Pueblo*

**UNIDAD ACADÉMICA DE INGENIERÍA,  
INDUSTRIA Y CONSTRUCCIÓN**

**CARRERA DE INGENIERÍA ELECTRÓNICA**

**SISTEMA SCADA DE BAJO COSTO PARA LA  
INTEGRACIÓN DE DISPOSITIVOS HETEROGÉNEOS DEL  
LABORATORIO DE PLC'S DE LA UNIVERSIDAD  
CATÓLICA DE CUENCA**

**TRABAJO DE TITULACIÓN O PROYECTO DE INTEGRACIÓN  
CURRICULAR PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO ELECTRÓNICO**

**AUTOR: WILSON ISMAEL CARANGUI RODRÍGUEZ**

**DIRECTOR: ING. PAUL MATA QUEVEDO**

**AZOGUES - ECUADOR**

**2020**

*Yo me gradúe en los  
50 años de La Cato!*

©Copyright Wilson I. Carangui.  
All Rights reserved

# Resumen

La supervisión y el control en el proceso de producción son funciones primordiales en la industria. Para lograr este objetivo, generalmente es necesario utilizar herramientas informáticas dedicadas y el costo suele ser alto. Por lo tanto, para poder comunicarse con los equipos de planta y realizar una adecuada monitorización del proceso, se debe contar con un sistema SCADA (Sistema de adquisición de datos y control supervisado); esto significa una fuerte inversión en *hardware* y *software*, lo que limita su implementación en algunas empresas. Este proyecto presenta el diseño e implementación del sistema SCADA para realizar la intercomunicación de dispositivos de control heterogéneos en el laboratorio de ingeniería electrónica, el cual está integrado por PLCs (controlador lógico programable), Raspberry y Arduino. Una investigación acerca de la comunicación, topología y las actuales plataformas que brindan las prestaciones requeridas para el sistema, fue puesta en marcha, y se definió el uso de Node-RED, debido a que es de uso libre y sus características cumplen con los requisitos. Una vez creada la subred, se realizó la programación para que los dispositivos se comuniquen entre sí y se diseñó una interfaz de usuario para poder interactuar con el sistema. Además, se creó una base de datos para almacenar los diferentes registros de los sensores que participan en este proceso. Una vez finalizado el diseño y la implementación del sistema, se ejecutó pruebas de transmisión de datos y programación. Con el fin de validar el trabajo, realizar una correcta gestión y lograr el mejor desempeño del proceso, se creó una guía de prácticas.

**Palabras claves:** SCADA, PLC, *hardware*, *software*, Node-RED

# Abstract

Supervision and control of the production process is a vital function for any industry. In order to achieve this goal, it is generally necessary to use specialized computer tools, but their cost is often high. Therefore, to communicate with plant equipment and perform appropriate process monitoring, the SCADA (Supervisory Control and Data Acquisition) system must be available. This means a strong investment in hardware and software, which limits its implementation in some companies. This project introduces the design and implementation of the SCADA system to intercommunicate heterogeneous control devices in the electronic engineering laboratory, which consists of PLCs (programmable logic controllers), Raspberry, and Arduino. Research on communication, topology, and current platforms that provide the required performance for the system was launched, and Node-RED usage was defined because it is free to use and its features meet the requirements. After the subnet was created, programming was performed so that the devices would communicate with each other and a user interface was designed to interact with the system. Additionally, a database was created to store the different records of the sensors involved in this process. After system design and deployment was completed, data transmission and programming tests were run. Finally, an internship guide was created to validate the work, perform proper management, and to achieve the best performance of the process.

**Keywords:** SCADA, PLC, *hardware*, *software*, Node-RED

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Introducción	1
1.2. Definición del problema	2
1.3. Línea de investigación	2
1.4. Objeto de estudio	2
1.5. Objetivos	2
1.6. Hipótesis	3
1.7. Contribuciones	3
<b>2. Sustento teórico</b>	<b>4</b>
2.1. Sistemas SCADA	4
2.1.1. Definición	4
2.1.2. Objetivos de los sistemas SCADA	4
2.1.3. Arquitectura de un sistema SCADA	5
2.1.4. Funciones de un sistema SCADA	6
2.1.5. Software del sistema SCADA	7
2.1.6. Hardware del sistema SCADA	7
2.2. Comunicaciones industriales	8
2.2.1. Sistemas distribuidos	9
2.3. Pirámide CIM ( <i>Computer Integrated Manufacturing</i> )	9
2.3.1. Nivel 0 o de proceso	10
2.3.2. Nivel 1 o de campo	10
2.3.3. Nivel 2 o de célula	10
2.3.4. Nivel 3 o de planta	10
2.3.5. Nivel 4 o de fábrica	11
2.4. Modelo OSI ( <i>Open Systems Interconnection</i> ) de comunicación	11
2.5. Buses de campo	11
2.5.1. Ventajas de los buses de campo	12
2.5.2. Ethernet/IP	12
2.5.3. DeviceNet	13
2.5.4. AS-Interface	14
2.5.5. Modbus	14
2.5.6. Profibus	15

2.5.7. Profinet	16
2.5.8. Lonworks	17
2.6. Controlador lógico programable (PLC)	18
2.6.1. PLC Simatic S7-1200	19
2.6.2. Step 7 Basic	20
2.7. Raspberry Pi	20
2.8. Arduino UNO	22
2.9. JSON	22
2.9.1. Node.js	22
2.9.2. Node-RED	23
2.9.3. Oracle MySQL Database Service	24
<b>3. Diseño e implementación del sistema</b>	<b>26</b>
3.1. Arquitectura del proyecto	26
3.2. Desarrollo de la red	27
3.2.1. Configuración de router (Cisco RV320)	27
3.2.2. Direccionamiento de la red	28
3.3. Configuración de PLCs	28
3.4. Configuración de Arduino	30
3.5. Creación del servicio de Node-RED	31
3.5.1. Instalación de Node.js en Raspberry Pi	31
3.5.2. Instalación de Node-RED en Raspberry Pi	31
3.5.3. Ejecutar NodeRed como un servicio	31
3.6. Desarrollo del sistema SCADA en Node-RED	32
3.6.1. Conexión de PLC con Node-RED	32
3.6.2. Conexión del arduino con Node-RED	36
3.6.3. Control de pines GPIO de Raspberry con Node-RED	38
3.7. Base de datos	40
3.7.1. Servidor MySQL	40
3.7.2. MySQL en Node-RED	42
3.8. Interfaz gráfica	43
3.8.1. Dashborad en Node-RED	43
3.8.2. Administración de usuarios	45
3.8.3. Protección de la interfaz UI de Node-RED	47
3.9. Repotenciación de tableros de control	47
<b>4. Pruebas de funcionamiento y análisis de resultados</b>	<b>50</b>
4.1. Inicio de sesión en el editor de Node-RED y en la interfaz SCADA	50
4.2. Validación de usuarios	52
4.3. Dashboard	52
4.3.1. Ventana MAIN	53
4.3.2. Ventana WRITE DATA	53
4.3.3. Ventana READ DATA	53
4.3.4. Ventana HISTORICAL TREND	54

4.3.5. Ventana DB SCADA	55
4.4. Registro de datos en DB	56
4.5. Pruebas de control de salidas de los controladores	58
4.5.1. Prueba de control de salidas de los PLCs	58
4.5.2. Prueba de control de salidas del Arduino	59
4.5.3. Prueba de control de salidas de la Raspberry	60
4.6. Eventos y alarmas	61
4.7. Acceso remoto al sistema	65
<b>5. Conclusiones</b>	<b>67</b>
5.1. Conclusiones	67
5.2. Trabajo futuro	68
<b>A. Práctica No. 1</b>	<b>72</b>
A.1. Objetivos	72
A.2. Descripción	72
A.3. Equipos y materiales	72
A.4. Desarrollo de la práctica	73
A.5. Reporte	78
<b>B. Práctica No. 2</b>	<b>79</b>
B.1. Objetivos	79
B.2. Descripción	79
B.3. Equipos y materiales	79
B.4. Desarrollo de la práctica	80
B.5. Reporte	85
<b>C. Tablas de la base de datos nodedata</b>	<b>86</b>

# Índice de figuras

2.1. Esquema elemental del sistema SCADA	4
2.2. Arquitectura básica de un sistema SCADA	6
2.3. Estructura básica del <i>hardware</i> de un sistema SCADA	7
2.4. Diagrama de comunicación industrial	8
2.5. Pirámide CIM	10
2.6. Esquema del Ethernet Industrial	13
2.7. Esquema del protocolo ModBus	14
2.8. Esquema del protocolo Profibus	15
2.9. Esquema del protocolo Profinet	16
2.10. PLC S7-1200	19
2.11. <i>Software</i> Setp 7	20
2.12. Raspberry Pi 3 B+	21
2.13. Arduino UNO	22
2.14. Interfaz gráfica de Node-RED	23
2.15. MySQL Workbench	25
3.1. Arquitectura del proyecto	26
3.2. Configuración de VLAN en router CISCO	27
3.3. Direccionamiento y configuración de DHCP en router CISCO	27
3.4. Topología de red	28
3.5. Interconeccción de PLC's en TIA PORTAL	29
3.6. Primer bloque de programa en TIA PORTAL 15 con el esquema de contactores	29
3.7. Segundo bloque de programa en TIA PORTAL 15 con el esquema de diagrama de funciones	30
3.8. Ubicación de sketch de FIRMATA en el IDE de Arduino	31
3.9. Instalación Node-RED en la terminal de Raspberry	32
3.10. Nodos S7 en editor de Node-RED	33
3.11. Ventana de configuración del nodo S7	33
3.12. 1 Parámetros de comunicación del nodo S7 (izquierda), 2 Variables del PLC (derecha)	34
3.13. Configuración de protección del CPU en TIA Portal	36
3.14. Configuración de atributos de bloques de programa en TIA Portal	36
3.15. Nodos Arduino en editor de Node-RED	37
3.16. 1. Configuración del nodo ArduinoInput (izquierda), 2. Configuración del nodo ArduinoOnput (derecha)	38
3.17. Nodos Raspberry Pi	39

3.18. Ventana de configuración del nodo rpi-gpio out en Node-RED	39
3.19. Ventana de configuración del nodo DHT en Node-RED	40
3.20. Tablas para almacenamiento de datos en MySQL	41
3.21. Nodo MySQL en Node-RED	42
3.22. Ventana de configuración del nodo mysql en Node-RED	43
3.23. Nodos Dashboard en Node-RED	44
3.24. Flujo para control de LED's y realizar consultas de base de datos en Node-RED	44
3.25. Ventana de grupos de dashboard en Node-RED	45
3.26. Archivo de configuración de Node-RED	46
3.27. Creación de hash de contraseña en Raspberry	46
3.28. Configuración de seguridad HTTP en el archivo de configuración de Node-RED	47
3.29. Tablero de control	48
3.30. Panel de contactos del tablero de control	48
3.31. Panel de contactos del tablero de control	49
3.32. Panel desensamblado para acceder a contactores	49
4.1. Ventana de inicio de sesión en el editor Node-RED	51
4.2. Ventana de inicio de sesión <i>dashboard</i> SCADA	51
4.3. Ventana MAIN del sistema SCADA	52
4.4. Ventana WRITE DATA del sistema SCADA	53
4.5. Ventana READ DATA del sistema SCADA	54
4.6. Ventana HISTORICAL TREND del sistema SCADA	55
4.7. Ventana DB SCADA del sistema SCADA	56
4.8. Historiador de los datos de humedad de la DB	57
4.9. Historiador de los datos de temperatura almacenados en la DB	57
4.10. Historiador de los datos de voltaje de la entrada analógica del PLC3 almacenados en la DB	58
4.11. Control de salidas de PLCs desde Node-RED	59
4.12. Control de salidas del Arduino desde Node-RED	60
4.13. Control de salidas de la Raspberry desde Node-RED	60
4.14. Evento entre Arduino y LED de Raspberry	61
4.15. Mensaje de alarma de voltaje muy alto censado en el Arduino	62
4.16. Historiador sobre el evento entre Arduino y LED5 de Raspberry	63
4.17. Evento entre PLC3 y LED de Raspberry	63
4.18. Mensaje de alarma de voltaje muy alto censado en el PLC	64
4.19. Historiador sobre el evento entre Arduino y LED5 de Raspberry	65
4.20. Acceso al sistema desde una computadora	66
4.21. Acceso al sistema desde un teléfono inteligente	66
A.1. Nodos designados para la práctica	74
A.2. Ejemplo de parametrización del nodo S7	74
A.3. Ejemplo de parametrización del nodo arduino in	75
A.4. Ejemplo de parametrización de los nodos gpio output y dht22	75
A.5. Ubicación de sketch de FIRMATA en el IDE de Arduino	76
A.6. Nodos Dashboard en Node-RED	77

A.7. Ejemplo conexión de nodos con switch del dashboard . . . . .	77
A.8. Ejemplo conexión de nodos gauge y chart . . . . .	78
B.1. Nodo MySQL en Node-RED . . . . .	81
B.2. Ventana de configuración del nodo mysql en Node-RED . . . . .	82
B.3. Ejemplo de programación para realizar consultas en MySQL . . . . .	82
B.4. Archivo de configuración de Node-RED . . . . .	83
B.5. Configuración de seguridad HTTP en el archivo de configuración de Node-RED . . . . .	84
B.6. Nodo notification en Node-RED . . . . .	84

# Índice de tablas

2.1. Modelos de sistemas distribuidos	9
2.2. Funciones de las capas del modelo OSI	11
2.3. Análisis de características entre buses	18
2.4. Especificaciones técnicas del Raspberry 3 B+	21
3.1. Direccionamiento de variables	35
3.2. Tipo de nodos nodeArduino	37
4.1. Usuarios y permisos del sistema SCADA	52
4.2. Pruebas de eventos sobre el sensor de la placa Arduino	62
4.3. Pruebas de eventos sobre el sensor del PLC3	64
C.1. Datos de humedad registrados en la base de datos	88
C.2. Datos de temperatura registrados en la base de datos	91
C.3. Datos de voltaje registrados en la base de datos	93

# Dedicatoria

*A la vida, a la naturaleza y a los seres que me aman.*

# Agradecimiento

*Mi formación académica la debo a la Universidad Católica de Cuenca sede Azogues y sus docentes de la Facultad de Ingeniería Electrónica, al ser Supremo que me permitió ver muy claro mi norte y potenció mis capacidades para cumplir mi objetivo.*

# Capítulo 1

## Introducción

### 1.1. Introducción

El proceso tecnológico y la innovación son las vías más claras de la evolución. Éstas tienen una evidente repercusión en el ámbito de la dinámica competitiva y la viabilidad de las organizaciones. La flexibilidad que brindan las nuevas tecnologías está descrita en acciones tangibles y/o intangibles. Pueden estar sujetas a mediciones de forma directa o indirecta, dependiendo de la naturaleza del proceso. Por ejemplo, una disminución en el tiempo de ejecución del proceso, se traduce directamente en ahorro financiero de la empresa (Alvarez Echeverría and Antonio, 2015).

El uso de sistemas SCADA permite monitorear y controlar de manera remota diferentes procesos utilizando una interfaz gráfica de usuario, posibilitando al operador detener o modificar varias etapas del sistema a la vez. La producción automatizada ha experimentado cambios tremendos a partir de la llegada de los controladores programables. Este equipo hace que los procesos industriales sean más precisos y eficientes. De esta forma se reduce el gran costo que comprende el reemplazo del enredado sistema de control convencional basado en contactores y relés (José M. Hurtado, 2017).

Debido al costo de las licencias y los paquetes de *software*, los sistemas SCADA no siempre son factibles para ciertos tipos de empresas. Aún así, los dispositivos heterogéneos deben vincularse para formar un solo sistema. Por ello, suelen ser varias las empresas que renuncian o se niegan a integrar sistemas, y también se resisten a los beneficios de la tecnología en el proceso (Merchán Piedra and Peralta Sarmiento, 2017).

La implementación de la tecnología SCADA permite un alto nivel de automatización para resolver los problemas de desarrollo, almacenamiento, transmisión, procesamiento y visualización de información. Las principales ventajas de los sistemas SCADA incluyen: reducir el tiempo para desarrollar sistemas grandes y distribuidos, así como la conveniencia y la facilidad de las modificaciones del sistema operativo local a través de herramientas de configuración integradas (Prokhorov et al., 2018). Este sistema promete a los estudiantes de la Facultad de Ingeniería Electrónica nuevos métodos de control de procesos. Así, la institución adquiere un beneficio pedagógico y actualización en su infraestructura y sus sistemas.

## 1.2. Definición del problema

Actualmente el laboratorio de PLC's de la Universidad Católica de Cuenca sede Azogues, cuenta con diferentes módulos y equipos para realizar prácticas de automatización y control. No obstante, estos no son aprovechados al máximo debido a la ausencia de un sistema SCADA que interconecte todos los dispositivos.

Los esquemas de licenciamiento de SCADAs comerciales son, generalmente, costosos; esto limita, entre otras cosas, su adquisición a empresas pequeñas y medianas. Además se genera una dependencia permanente de sus desarrolladores (Padilla Julca and Toro Ydrogo, 2018). Es así como surge la propuesta de diseño e implementación de un sistema SCADA de *software* libre, capaz de integrar dispositivos heterogéneos, para el laboratorio de la Facultad de Ingeniería Electrónica.

Existen varias alternativas para la integración de instrumentos mediante un sistema SCADA. Por ejemplo, (Faedo et al., 2015) plantean un sistema SCADA de tercera generación. Este sistema puede distribuirse por una red LAN (*Local Area Network*, en inglés), aumentando considerablemente el rendimiento. La implementación del protocolo WAN (*Widel Area Network*, en inglés) es la mayor contribución del sistema SCADA de tercera generación. (Almas et al., 2014) proponen una actualización sobre la integración del código abierto SCADA BR en el Laboratorio de Sistemas de Transmisión Inteligente (*SmarTS-Lab*). Cuentan con dispositivos que son compatibles con DNP3 y que pueden configurarse como estaciones externas DNP3 para transmitir su estado analógico y digital. Todas estas PMU (*Phasor Measurement Unit*) se pueden acoplar con el simulador en tiempo real eMEGAsim de Opal-RT para adquirir mediciones de sincrofasores para un escenario simulado. SCADA BR está configurado con SmarTS-Lab en una de las estaciones de trabajo en el laboratorio para adquirir las actualizaciones de estado de estas PMU y relés que usan DNP3 para fines de monitoreo y control. Existe otra posibilidad de implementación, (Minchala et al., 2016) ha desarrollado un sistema que utiliza software gratuito para mejorar las capacidades de fabricación avanzadas. El proyecto se compone de un host asignado como servidor de comunicaciones OPC (*OLE for Process Control*) y de equipos de automatización tradicionales. Los dispositivos se mantienen comunicados con el servidor OPC por medio del protocolo Ethernet, lo que brinda un registro de entradas y salidas desde el sistema SCADA.

## 1.3. Línea de investigación

Ciencias exactas, naturales y tecnológicas.

## 1.4. Objeto de estudio

Control industrial, monitoreo y automatización.

## 1.5. Objetivos

### Objetivo general

- Diseñar e implementar un sistema de comunicación de red industrial capaz de integrar diversos dispositivos para prácticas de comunicación industrial de control y automatización mediante un sistema SCADA de bajo costo para el laboratorio de automatización de la Universidad Católica de Cuenca sede Azogues.

**Objetivos específicos**

- Recopilar información sobre el estado del arte del proyecto.
- Desarrollar una red que interconecte los dispositivos existentes en el laboratorio de electrónica de la Universidad Católica de Cuenca sede Azogues y cualquier otro equipo de automatización que se desee integrar.
- Configurar los equipos de administración de la red industrial: switch y router.
- Repotenciar los tableros/estaciones de los PLC's.
- Validar los procesos mediante la implementación de dos prácticas de laboratorio con conexión remota.

**1.6. Hipótesis**

¿En qué medida un sistema SCADA de software libre ayuda a solucionar problemas de comunicación de equipos industriales del laboratorio de PLC's de la facultad de Ingeniería Electrónica? El sistema SCADA de software permite la interconexión de dispositivos heterogéneos de control y automatización, sin faltar a las normas de regulación internacional.

**1.7. Contribuciones**

Este proyecto de investigación brinda las siguientes contribuciones:

- Mediante el uso de un sistema de código abierto, solventa los problemas de comunicación de equipos industriales heterogéneos.
- Disminución de costos y mejora en la flexibilidad de sistemas de monitoreo en la industria.
- Brindar nuevas áreas de investigación y trabajo en automatización y sistemas de software libre para monitorear y controlar procesos industriales.

# Capítulo 2

## Sustento teórico

### 2.1. Sistemas SCADA

#### 2.1.1. Definición

(Penin, 2011) asigna el nombre SCADA a los sistemas que permiten procesar datos remotos y admiten el uso de las herramientas de comunicación necesarias para controlarlo, es decir, es un sistema diseñado para monitorear y controlar instalaciones de forma remota, que puede integrar datos recopilados de diferentes sensores, autómatas (PLC) y dispositivos que utilizan diferentes protocolos en un solo lugar. La Figura 2.1 muestra un esquema básico de los sistemas SCADA.

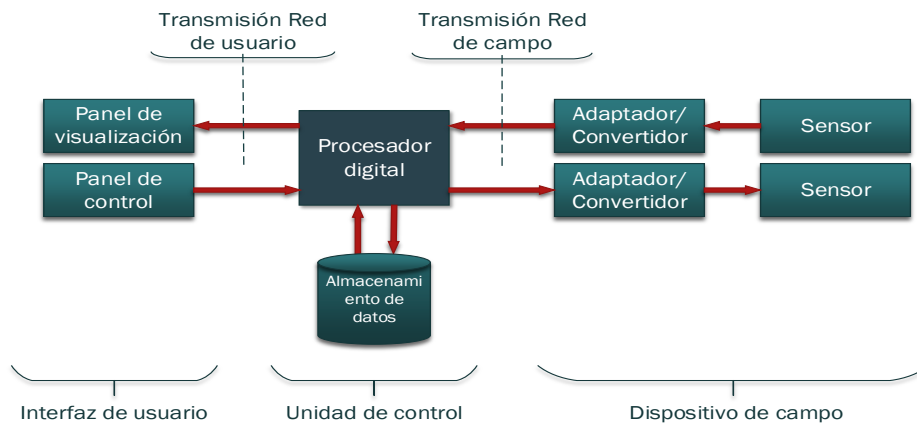


Figura 2.1: Esquema elemental del sistema SCADA

Fuente: (Castro and Lozano, 2017)

#### 2.1.2. Objetivos de los sistemas SCADA

Los sistemas SCADA, según (Penin, 2011), están diseñados para la supervisión y el control de procesos industriales, por lo que apuntan a cumplir con los siguientes objetivos:

- **Accesibilidad**

Permite al operador controlar y manipular todo el proceso. Desde cualquier ubicación utilizando la interfaz HMI.

- **Mantenimiento**

A través de la base de datos generada por el proceso industrial, se puede generar una serie de informes desde la planta y, a través de estos, se puede retener el control de cada elemento relacionado con el proceso industrial para el mantenimiento programado.

- **Economía**

Usando el sistema SCADA, es posible verificar lo que sucede en la fábrica donde se concentra todo el contenido en lugar de permitir que el operador realice el mismo trabajo

- **Gestión**

Los datos obtenidos por el sistema pueden procesarse de diferentes maneras, como datos estadísticos, gráficos, etc.

- **Flexibilidad**

Se pueden realizar cambios en el sistema de visualización sin incurrir en costos de tiempo y de infraestructura del sistema.

### 2.1.3. Arquitectura de un sistema SCADA

En la automatización industrial, el control desde la PC hasta la planta de proceso se divide en tres partes principales: sistema de adquisición de datos, comunicación y software SCADA (Penin, 2011).

La arquitectura básica de un sistema SCADA se puede observar en la Figura 2.2

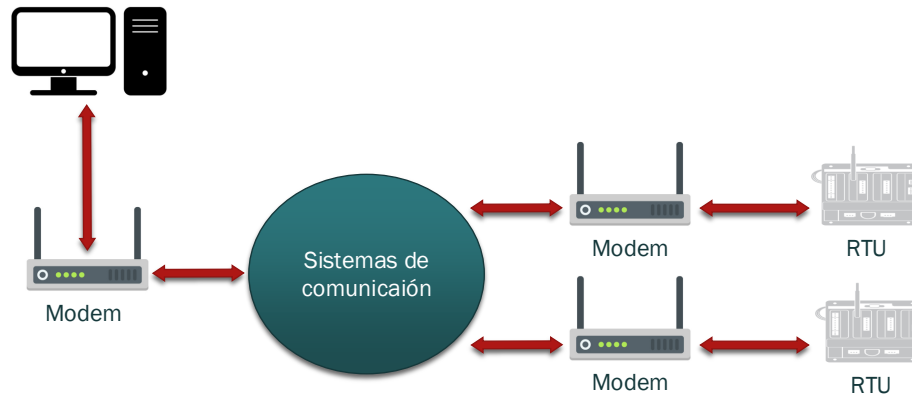


Figura 2.2: Arquitectura básica de un sistema SCADA

Fuente: (Penin, 2011)

El sistema SCADA está diseñado para trabajar en ordenadores en control de producción y proporciona comunicación entre dispositivos de campo (también conocidos como RTU (Unidad Terminal Remota)), aquí se puede encontrar autómatas programables y unidades centrales (MTU) que controlan automáticamente los procesos. Los usuarios pueden acceder al sistema a través de una interfaz visual, generalmente una computadora (sistema de servidor) con aplicaciones de control y monitoreo instaladas (Penin, 2011).

#### 2.1.4. Funciones de un sistema SCADA

(Moreano Abata, 2019) señaló en su documentación que las funciones básicas del sistema SCADA en un proceso industrial particular deben garantizar que funcione, y observar los siguientes parámetros:

- **Supervisión remota de procesos**

Utiliza las señales recibidas de los dispositivos de campo para monitorear el estado de los mismos y planificar el trabajo de mantenimiento.

- **Control de procesos remotamente**

Gestiona de forma remota un proceso durante el cual permite que los elementos del actuador se activen o desactiven desde un ordenador o la pantalla HMI.

- **Procesamiento de información**

Los datos adquiridos por cada actuador o dispositivo de campo permiten el análisis y la comparación con datos anteriores para obtener información importante, confiable y efectiva.

- **Almacenamiento de datos**

Los datos y la información recibidos se almacenan en el disco duro de la computadora para su posterior análisis.

### 2.1.5. Software del sistema SCADA

En el software SCADA, se pueden distinguir dos tipos de bloques de programa.

El primero es el programa de desarrollo, que incluye utilidades relacionadas con la creación y la versión del sistema de aplicación, así como las características principales como texto, gráficos y colores. El segundo es la ubicación donde se ejecuta SCADA o Run-Time, que le permite ejecutar aplicaciones creadas con programas de desarrollo (Gualan Sarango and Pintado, 2019).

### 2.1.6. Hardware del sistema SCADA

(Gualan Sarango and Pintado, 2019) explica que la principal característica del sistema SCADA es que se puede programar en el sistema informático, pero como todo software, requiere hardware para funcionar de manera efectiva. En este caso, la parte del hardware se puede dividir en 4 partes importantes como muestra la Figura 2.3

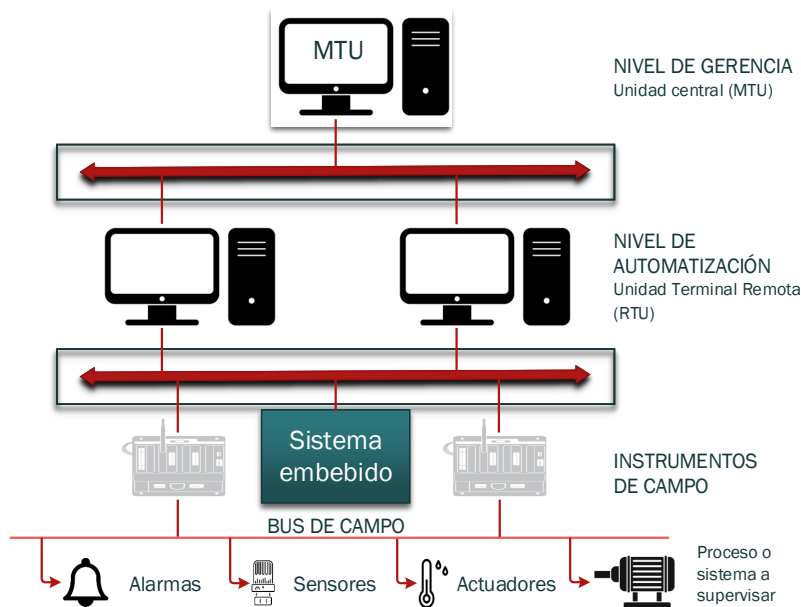


Figura 2.3: Estructura básica del *hardware* de un sistema SCADA

Fuente: (Romagosa et al., 2004)

- **Nivel de Gerencia**

Este nivel se puede considerar como el nivel principal, porque aquí está el ordenador o MTU, que es responsable de recopilar toda la información de las subestaciones restantes

- **Nivel de Automatización**

Aquí se encuentran computadoras remotas o RTU. Reciben señales de sensores en el campo y luego ejecutan SCADA para ordenar los componentes de control finales. Estos elementos suelen ser PCs, pero ahora también se consideran los PLCs.

- **Instrumentos de Campo**

Estos elementos permiten el control y la automatización del sistema de ejecución, en este nivel suelen estar sensores, alarmas, PLCs y controladores de procesos industriales.

- **Red de Comunicaciones**

Permite que la información se transfiera entre la fábrica y la arquitectura de hardware que admite el sistema SCADA a través de cable o inalámbricamente, ya que puede utilizar cualquier protocolo industrial. CANbus, Fieldbus, Modbus, etc.

## 2.2. Comunicaciones industriales

En el último siglo, el progreso tecnológico se desarrolló junto con la revuelta industrial. El corazón de la industria y sus procesos comprendía los sistemas mecánicos. Las técnicas de automatización toman fuerza en los últimos años del siglo XX, optimizando los sistemas establecidos en relés.

La inserción de controladores programables hace que los sistemas industriales sean más eficientes. Esto genera una reducción de costos al sustituir los de sistemas de panel de control complejos (Albán Guerrero and Paguay Llamuca, 2017).

Las redes de comunicación industrial, como se puede apreciar en la Figura 2.4, son una parte indispensable del proceso de producción de una empresa porque transmiten información entre varios dispositivos en un sistema determinado, por lo que el tipo de comunicación industrial debe considerar muchos aspectos específicos para cumplir con los requisitos de la empresa en equipos y procesos.

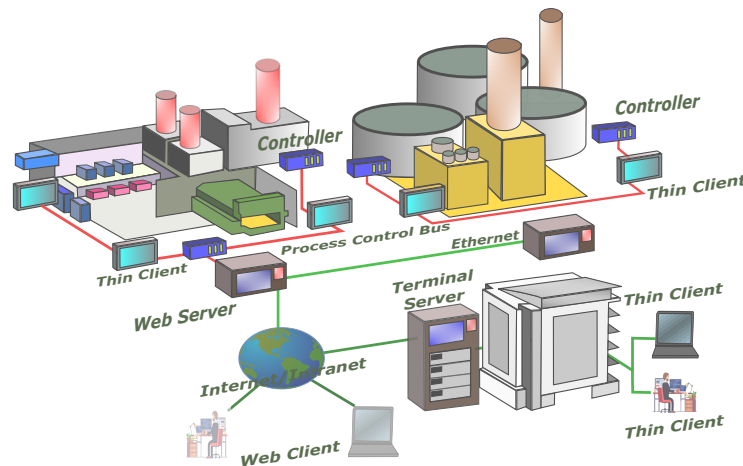


Figura 2.4: Diagrama de comunicación industrial

Fuente: (Mundologia, 2015)

### 2.2.1. Sistemas distribuidos

Los equipos independientes inteligentes que trabajan con objetos específicos, conforman un sistema distribuido (SD). Estos equipos desempeñan funciones ordenadas para intercambiar información a través de la red. Se considera un sistema distribuido cuando existen nodos que ejecutan labores de control (Castro Gil et al., 2017).

En el campo de las comunicaciones industriales, según (Penin, 2011), se utilizan los siguientes tres procesos o modelos:

- Modelo cliente-servidor
- Modelo productor-consumidor
- Modelo de publicación-suscripción

La Tabla 2.1 muestra las diferentes características de cada modelo.

Tabla 2.1: Modelos de sistemas distribuidos

Palabra a enviar	Modelo cliente-servidor	Modelo productor-consumidor	Modelo de publicación-suscripción
<b>Tipo de comunicación</b>	Entre iguales ( <i>Peer-to-peer</i> )	Difusión ( <i>Broadcast</i> )	Multidifusión ( <i>Multicast</i> )
<b>Estilo de comunicación</b>	Orientada a la conexión	Sin conexión explícita	Sin conexión explícita
<b>Relación maestro-esclavo</b>	Uno o varios maestros	Varios maestros	Varios maestros
<b>Servicios de comunicaciones</b>	Confirmados, sin confirmar, con confirmación	Sin confirmar, con confirmación	Sin confirmar, con confirmación
<b>Clases de aplicaciones</b>	Transferencia de parámetros, comunicación cíclica	Motificación de eventos, alarmas, eventos, sincronización	Cambios de estado y notificación de eventos

Fuente: (Penin, 2011)

### 2.3. Pirámide CIM (*Computer Integrated Manufacturing*)

En las redes industriales, las comunicaciones se encuentran concentradas jerárquicamente de acuerdo a la información. Un subsistema mantiene comunicación directa con el mismo nivel de la pirámide y subsistemas de nivel superior o inferior. En el ámbito empresarial esto corresponde a una distribución piramidal jerarquizada. Esta estructura permite tomar decisiones empresariales desde lo más alto de la pirámide (Sarmiento Sánchez, 2016).

La Figura 2.5 describe los niveles jerárquicos de la pirámide CIM.



Figura 2.5: Pirámide CIM

Fuente: Autor

Según (Salazar Serna and Correa Ortiz, 2013), los niveles jerárquicos de la pirámide CIM son los siguientes:

### 2.3.1. Nivel 0 o de proceso

Consiste en elementos de medida y control denominados actuadores dispersos en el sistema de producción. Los sensores envían las señales adquiridas a los dispositivos del siguiente nivel superior para que tomen acciones de control y expidan respuestas acertadas a los actuadores. En este nivel se comunican los elementos de campo o *field* con los controladores de nivel de célula por medio de los conocidos buses de campo.

### 2.3.2. Nivel 1 o de campo

Los elementos de mando y control se encuentran en este nivel. Pueden administrar sensores y actuadores de nivel 0, tales como: PLC de rango bajo y medio, transporte automático, sistemas CNC, controladores y proporcionar información de actuación y estado de nivel 0 y 2, respectivamente. Aquí, las variables de campo son procesadas de acuerdo a estas se ejecutan funciones asignadas en el sistema.

### 2.3.3. Nivel 2 o de célula

Las órdenes ejecutadas en el primer nivel son emitidas por este nivel y reciben el estado de esta capa. Además, recibe planes de producción y mantenimiento de nivel 3 que generan retroalimentación en la fábrica. En este nivel se encuentran los equipos de control y las computadoras industriales que reciben información y toman medidas.

### 2.3.4. Nivel 3 o de planta

En esta capa están ubicados los equipos de control de planta que tienen como función la sucesión y dirección de recursos sobre capas inferiores. Cuenta con sistemas de supervisión y control SCADA. Este nivel tiene como objetivo el control y la planificación de la producción.

### 2.3.5. Nivel 4 o de fábrica

La fábrica es gestionada completamente en este nivel. Aquí se comunican varias plantas y se generan relaciones con clientes y proveedores. Computadoras de oficina y de ingeniería, servidores y estaciones de trabajo son implementadas en esta capa. Además, existe un intercambio de información muy grande.

## 2.4. Modelo OSI (*Open Systems Interconnection*) de comunicación

Este modelo se puede utilizar como base para sincronizar sistemas abiertos. El modelo OSI cuenta con 7 capas, como se muestra en la Tabla 2.2. El usuario puede intervenir solamente en la capa física y de aplicación (Espinoza Castro and Baño Soto, 2018).

Tabla 2.2: Funciones de las capas del modelo OSI

MODELO OSI	
Capa	Función
<b>Aplicación</b>	Otorga a los usuarios acceso al entorno OSI y a los servicios de información asignados..
<b>Presentación</b>	La diferencia en la representación de datos (sintaxis) hace que el proceso de aplicación sea independiente.
<b>Sesión</b>	La comunicación entre aplicaciones es controlada en esta capa. Las sesiones entre acciones colaboradoras son gestionadas aquí.
<b>Transporte</b>	Genera Transparencia y seguridad en los datos de los puntos finales, rescata errores y gestiona el flujo hasta el destino.
<b>Red</b>	Otorga un mayor nivel de independencia en relación con la tecnología de conmutación y transmisión utilizada para conectar el sistema. Es responsable de establecer, mantener y cerrar conexiones.
<b>Enlace de datos</b>	Permite la transmisión segura de datos a través de enlaces físicos. También envía tramas de datos que realizan sincronización, control de errores y transmisión.
<b>Físico</b>	Transmite cadenas de datos bits sin estructurar. Se relaciona con procedimientos, características y funciones de instrucciones eléctricas y mecánicas del entorno físico.

Fuente: (Espinoza Castro and Baño Soto, 2018)

### 2.5. Buses de campo

Estos buses permiten enviar y recibir datos, reduciendo así la instalación de sistemas industriales y máquinas en el proceso productivo. Están diseñados para reemplazar las conexiones punto a punto y proporcionar un cierto grado de independencia al sistema a través de equipos de procesamiento inteligente. Estos elementos pueden realizar

funciones de control, mantenimiento y monitorización sin olvidar que cuentan con comunicación bidireccional. (Albán Guerrero and Paguay Llamuca, 2017).

### 2.5.1. Ventajas de los buses de campo

Pueden suplir sistemas centralizados de control por medio de una red de control, esto mejora significativamente el rendimiento del proceso optimizando recursos y disminuyendo costos. La información enviada de dispositivo a otro, es digital; esto genera precisión en el sistema (Albán Guerrero and Paguay Llamuca, 2017).

Según (Albán Guerrero and Paguay Llamuca, 2017), las principales ventajas de los buses de campo son:

- Disminución de costos de ensamblaje, mantenimiento y operatividad del sistema.
- Reduce el cableado para sistemas de control o instalaciones.
- Mejora la confiabilidad del sistema o proceso.
- Mayor flexibilidad del usuario en la interfaz del sistema.
- Los sistemas de control son más efectivos, gracias a la simplificación en la adquisición de datos.
- Le permite al operador monitorear el equipo que compone el sistema y le permite explicar las interacciones entre ellos, haciéndolo más fácil y práctico.
- Los sistemas de control y los dispositivos de campo están conectados de forma bidireccional.

### 2.5.2. Ethernet/IP

Este protocolo es utilizado con frecuencia en procesos industriales. Se encuentra fundado en el protocolo TCP/IP y sus elementos de *hardware* y *software* establecen capas para controlar y acceder a dispositivos de automatización. Los nodos, en Ethernet, se clasifican con respecto al tipo de dispositivo y las funciones que realiza (Roque and Antonio, 2018).

La Figura 2.6 muestra el sistema integrado entre la red central y la fábrica que brinda Ethernet.

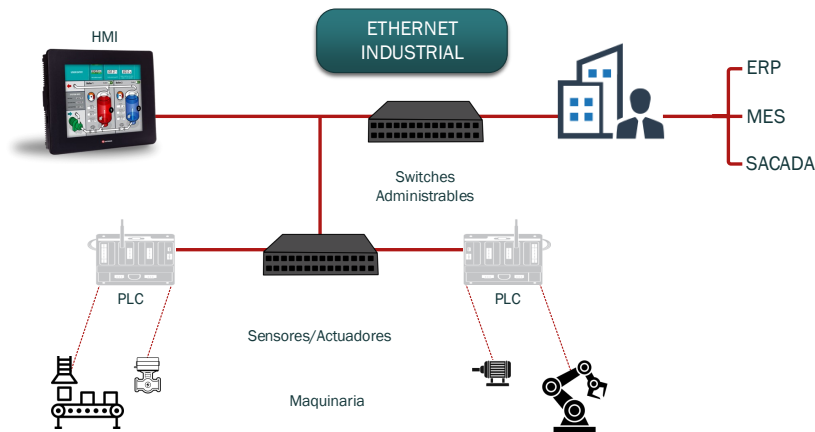


Figura 2.6: Esquema del Ethernet Industrial

Fuente: (Aula21, 2015)

Ethernet / ip está diseñado para implementar su tecnología tradicional y proporcionar acceso a la tecnología de señalización en la tarjeta de interfaz.

### 2.5.3. DeviceNet

Protocolo orientado al nivel bajo y medio de automatización. Se encuentra instaurado en el nivel de fábrica, de acuerdo a la pirámide CIM. Allen-Bradley fue quien diseñó, pero en la actualidad es un sistema abierto. Por esta razón muchos fabricantes basan sus prototipos en este protocolo. (Rosado Muñoz, 2010).

Las principales características de DeviceNet, según (Rosado Muñoz, 2010), son:

- Cuenta con un máximo de 64 nodos.
- 100 m y 500 m es la distancia máxima de transmisión.
- Brinda una velocidad de transferencia de datos de 125, 250 y 500 kbit/s.
- Su estructura de comunicación es de línea principal y puede derivar líneas a nodos.
- Emplea un par trenzado para energizar con un máximo de 8A y otro par para el transporte de datos de hasta 3A.
- Necesita terminación de línea con una impedancia de aproximadamente 120 ohmios.
- Tamaño máximo de mensaje: 8 bytes por nodo
- El sistema de transmisión se basa en el modelo productor/consumidor, por lo que admite el modelo maestro/esclavo, el modelo multimaestro y el modelo de igual a igual para ofrecer diferentes métodos.

### 2.5.4. AS-Interface

Es un sistema de cableado avanzado diseñado para modificar cables discretos conectados a controladores programables como sensores y actuadores. Este protocolo permite la construcción de un sistema de una sola línea para el equipo más simple, que es fácil de usar, fácil de conectar y operar (Roque and Antonio, 2018).

El protocolo se puede construir con topologías de árbol, bus, anillo o estrella. Cada segmento puede interconectar hasta 31 estaciones esclavas con una longitud máxima de 100 metros. Su velocidad de transmisión es de 167 kbps y su repetidor puede enlazar hasta tres segmentos (Salazar Serna and Correa Ortiz, 2013).

La característica principal de AS-Interface es la conexión directa y sencilla con el PLC, o como un enlace descentralizado con el sistema de bus ascendente.

### 2.5.5. Modbus

Fue desarrollado por Modicon para proporcionar comunicación maestro/esclavo entre los dispositivos de control y de campo. Su función principal es enviar y recibir señales analógicas y digitales, y generar registros entre ellas. La comunicación es de tipo maestro/esclavo, como se puede observar en la Figura 2.7. La estación maestra Modbus puede ser una computadora central o panel de operación, y la estación esclava puede ser un controlador programable (Penin, 2011).

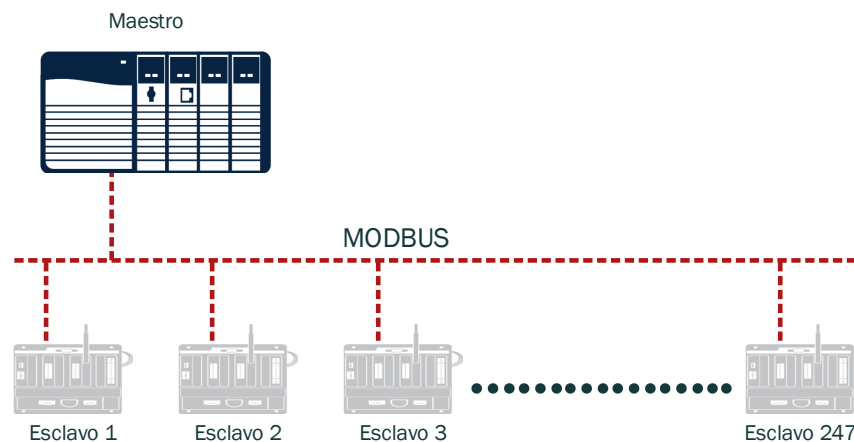


Figura 2.7: Esquema del protocolo Modbus

Fuente: (Marín, 2018)

Es el protocolo ideal para el monitoreo remoto por radio a través de elementos de campo (RTU), como los utilizados en estaciones de tratamiento de agua, instalaciones de gas natural o petróleo. Actualmente, se ha implementado en otros campos más allá de su concepto original, como la domótica o el control de procesos. (Penin, 2011).

El protocolo establece de que forma cada controlador identifica la dirección y cuando enviar mensajes, obtener datos o/y ejecutar operaciones. En otras redes, los mensajes que contienen el protocolo Modbus se incluyen en la estructura del paquete de datos que utiliza la red (Penin, 2011).

La comunicación convencional del protocolo es a través de una línea serial que puede ser RS-232 o RS-485, aunque en la actualidad se puede contar con otros tipos de conexión, como por ejemplo Modbus TCP, que encapsula tramas de Ethernet TCP/IP. Los dispositivos de marcas diferentes, pueden ser vinculados gracias a este protocolo (Espinoza Castro and Baño Soto, 2018).

### 2.5.6. Profibus

(Salazar Serna and Correa Ortiz, 2013) en su documento exhibe que, el gobierno alemán financió la creación de este protocolo y luego fue estandarizado por DIN E 19245 en Alemania y EN 50170 en Europa. Posteriormente AEG, ABB, Siemens, entre otros, obtuvieron el permiso para desarrollar y vender este producto.

La Figura 2.8 muestra el esquema del protocolo profibus.

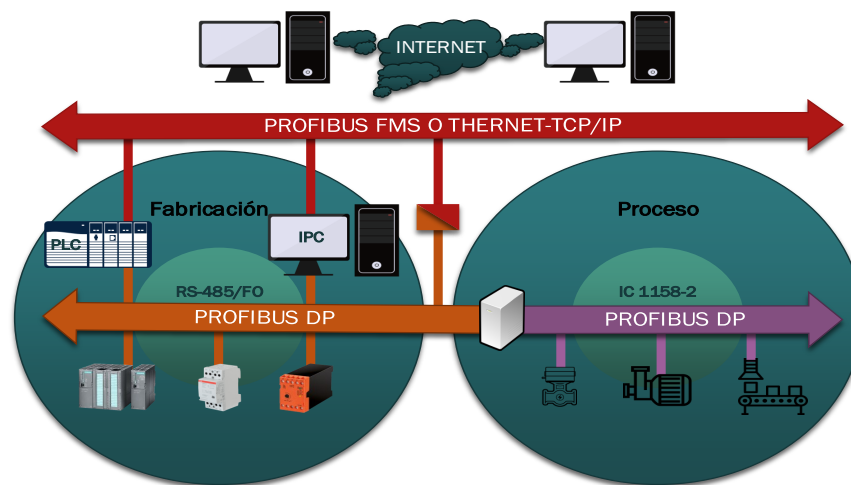


Figura 2.8: Esquema del protocolo Profibus

Fuente: (Salazar Serna and Correa Ortiz, 2013)

Existe tres versiones de Profibus, y cada versión está dedicada al campo de las comunicaciones:

- **Profibus-DP (Decentralized Periphery)**

Está optimizado para aplicaciones de velocidad y de bajo costo, enfocando sensores / actuadores conectados a un procesador (PLC) o terminal.

- **Profibus-PA (Process Automation)**

Está especialmente diseñado para el control de procesos y cumple con las normas especiales de seguridad (IEC 1 1 15 8-2) para entornos peligrosos y explosivos (como la industria química), con una velocidad de 31,25 Kbps y apta para una distancia máxima de 1,9 kilómetros.

- **Profibus-FMS (Fieldbus Message Specification)**

Es una solución para la comunicación entre unidades de proceso o equipos de automatización, y el desarrollo de Profibus es utilizar el protocolo TCP/IP para enlazar a nivel de proceso, lo que significa que el archivo de configuración ya no es importante. Cuando la distancia es superior a 100 km, puede proporcionar una alta velocidad de 9,6 Kbps a 1500 Kbps.

### 2.5.7. Profinet

TCP/IP, Ethernet y otros modelos de comunicación conforman la base del protocolo Profinet. Proporciona Ethernet en tiempo real y cooperación de los dispositivos que integran el bus de campo. (Roque and Antonio, 2018).

La Figura 2.9 expone el esquema del protocolo Profinet.

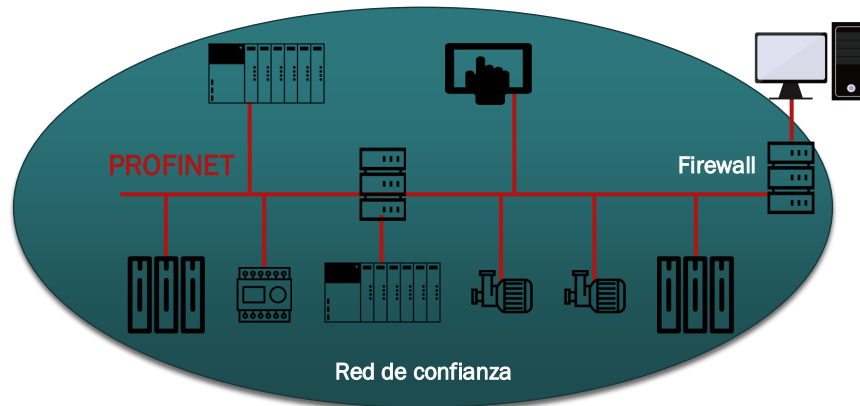


Figura 2.9: Esquema del protocolo Profinet

Fuente: (José M. Hurtado, 2017)

Se definen varios protocolos en el contexto de PROFINET, por ejemplo:

- **Profinet/CBA**

Protocolos relacionados con aplicaciones de automatización distribuida en ambientes industriales.

- **ProfinetT/DCP**

Descubrimiento y configuración básica. Protocolo basado en la capa de enlace, que se utiliza para configurar el nombre del dispositivo y la dirección IP.

- **Profinet/IO**

Algunas veces llamado PROFINET-RT (RealTime), se usa para comunicarse con E/S descentralizadas.

- **Profinet/MRP**

Protocolo para redundancia de medios. Cuando la red tiene una topología de anillo, si ocurre una falla, utilizará principios básicos para reorganizar la red.

- **Profinet/MRRT**

Tiene como propósito prestar soluciones de redundancia de medios para PROFINET/RT.

- **Profinet/PTCP**

El protocolo de control de precisión de tiempo basado en la capa de enlace se utiliza para sincronizar las señales de reloj/hora en varios PLC.

- **Profinet/RT**

Transmisión de datos en tiempo real.

- **Profinet/IRT**

Transmisión de datos síncronos en tiempo real.

### 2.5.8. Lonworks

El bus de campo se basa en el protocolo LonTalk y es distribuido por Echelon. Contempla una estructura completa de servicios y productos, y tiene como objetivo cubrir los mercados de domótica, control industrial y edificios inteligentes. Se han instalado millones de nodos y la plataforma está creciendo exponencialmente gracias al apoyo de miles de empresas que trabajan con LonWorks (Salazar Serna and Correa Ortiz, 2013).

Lonworks ofrece una amplia gama de opciones, incluida una variedad de medios físicos, como radio, fibra óptica, etc. Y topologías libres y en anillo. El par trenzado es el medio más utilizado, con una velocidad de 38 a 78 Kbps, y proporciona múltiples servicios con una amplia gama de estructuras y una gran cantidad de nodos (Salazar Serna and Correa Ortiz, 2013).

La Tabla 2.3 enseña las principales características de algunos buses de campo.

Tabla 2.3: Análisis de características entre buses

Nombre	Topología	Soporte	Distancia Max Km	Comunicación
<b>Profibus DP</b>	Línea, estrella, anillo	Par trenzado, fibra óptica	0.1 segm 24fibra	Master/Slave, Peer to peer
<b>Profibus PA</b>	Línea, estrella, anillo	Par trenzado, fibra óptica	0.1 segm 24fibra	Master/Slave, Peer to peer
<b>Profibus-FMS</b>		Par trenzado, fibra óptica		Master/Slave, Peer to peer
<b>LonWorks</b>	Bus, anillo, estrella, lazo	Par trenzado, fibra óptica, coaxial, radio	2	Master/Slave, Peer to peer
<b>DeviceNet</b>	Trocal/ puntal/ bifurcación	Par trenzado, fibra óptica	0.5 6c/rep	Master/Slave, Multi master, Peer to peer
<b>ASI</b>	Bus, estrella, anillo	Par trenzado, fibra óptica	0.1, 0.3 c/rep 24fibra	Master/slave
<b>Modbus</b>	Línea, estrella, árbol, red con segmentos	Par tenzado, coaxial, radios	0.35	Master/salve
<b>Ethernet</b>	Bus, estrella, malla, cadena	Coaxial, par tenzados, fibra óptica	0.1 1000 mono c/swtich	Master/Slave, Peer to peer

Fuente: (Salazar Serna and Correa Ortiz, 2013)

## 2.6. Controlador lógico programable (PLC)

Según (Moreano Abata, 2019), un PLC es un dispositivo electrónico que cuenta con memoria accesible al programa que puede almacenar instrucciones. Le permiten implementar varias funciones, como funciones de lógica, sincronización y secuencia. Su objetivo principal es el control de máquinas y procesos industriales.

El PLC es una parte importante de la automatización industrial. Varias fábricas se encuentran inmersas en la investigación de nuevas tecnologías con PLC, puesto que, son ampliamente utilizados en la industria, lo que ayuda a reducir los costos de producción y aumentar la producción e incrementa la calidad. Aunque también se puede programar usando KOP y FUP, la mayoría de los PLC usan un lenguaje llamado lógica de escalera en su programación. Existen diferentes tipos de PLC. Dependiendo de la marca o modelo, cada PLC utiliza diferentes tipos de software de programación (Cherrez, 2018).

El PLC tiene una estructura básica fácil de entender, con CPU, entradas y salidas analógicas y digitales, y puede comunicarse con dispositivos en un entorno industrial. Los componentes de estos dispositivos se pueden dividir en elementos de entrada, elementos de salida y relés internos, que se utilizan para activar o desactivar los elementos del proceso (Cherrez, 2018).

### 2.6.1. PLC Simatic S7-1200

El controlador básico SIMATIC S7-1200, mostrado en la Figura 2.10, es ideal para realizar tareas de automatización con la máxima flexibilidad y eficiencia en el rango de rendimiento bajo y medio. Con una amplia gama de entradas/salidas integradas y funciones técnicas, y un diseño compacto que ahorra espacio, pueden proporcionar resultados convincentes. Con un protocolo de control remoto estandarizado, puede conectar el controlador SIMATIC S7 1200 directamente al centro de control sin ninguna programación (Siemens, 2018)

Otro beneficio decisivo es la integración de todos los controladores SIMATIC en el *Totally Integrated Automation Portal* (TIA Portal): todos los controladores SIMATIC tienen acceso a bases de datos compartidas, conceptos operativos estandarizados y servicios integrados, como protocolos de comunicación (Siemens, 2018).

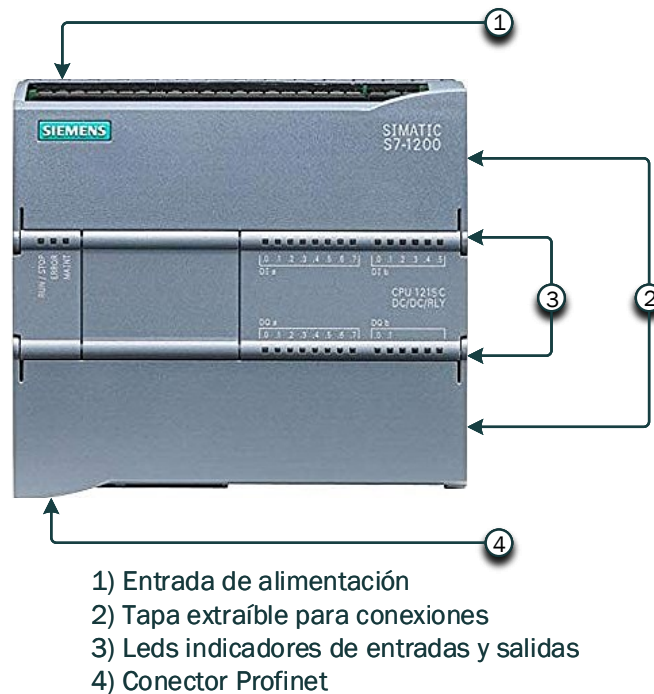


Figura 2.10: PLC S7-1200

Fuente: (Siemens, 2018)

Este tipo de PLC puede ejecutar de manera rápida y eficiente la lógica de programación, incluidas 8 entradas y 6 salidas digitales, y 2 entradas analógicas, y puede expandir los módulos de entrada y salida y un puerto Ethernet para comunicarse fácilmente con la PC. Todas las CPU en este tipo de PLC están protegidas con contraseña, lo que permite controlar el acceso a las funciones del dispositivo (Cherrez, 2018).

### 2.6.2. Step 7 Basic

El STEP 7 BASIC, que se muestra en la Figura 2.11, es el software de programación del PLC de la serie S7-1200-1500, y proporciona un entorno fácil de entender en términos de configuración del dispositivo, introduciendo lógica de programación en lenguaje Ladder, KOP y FUP, incluido el diseño de HMI y herramientas muy útiles para el desarrollo de procesos. Además, puede usar el sistema de PC HMI para ver el proceso, que es una herramienta gráfica muy útil que puede contener informes o alarmas definidos por el usuario (Siemens, 2018).

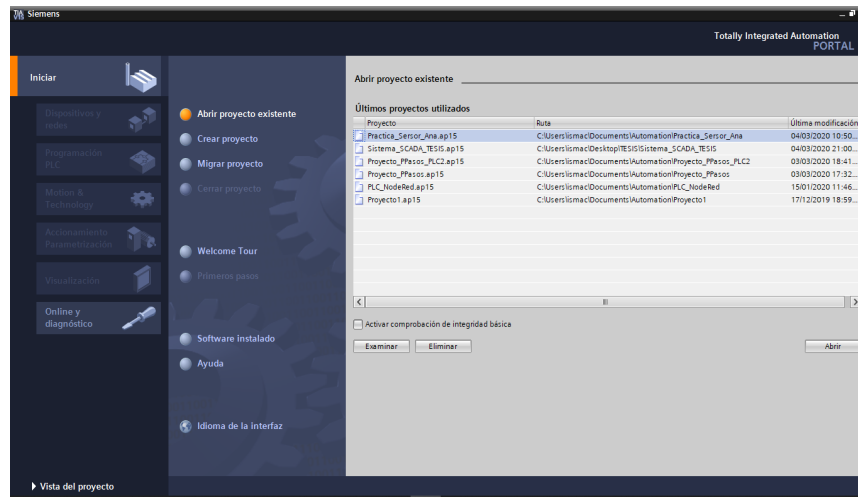


Figura 2.11: Software Setp 7

Fuente: Autor

## 2.7. Raspberry Pi

Este pequeño dispositivo fue desarrollado en 2006 por los ingenieros Eben Upton, Robmullins, Jack Lang y Alan Mycroft mientras trabajaban en un laboratorio de la Universidad de Cambridge, en Inglaterra. Las pequeñas computadoras realizan cálculos en las escuelas para enseñar a los jóvenes cómo programar en el último período de la escuela secundaria. Estas computadoras son muy baratas y pueden ser utilizadas por cualquier estudiante. Con el desarrollo de los procesadores móviles alrededor de 2008, la idea de crear microcomputadoras se volvió más factible, por lo que crearon la base Raspberry Pi. Comenzaron a utilizar este dispositivo para diseñar y vender el dispositivo, al que llamaron arquitectura abierta, y también animaron a otros fabricantes a copiar la idea (Areiza et al., 2015).

El Raspberry Pi 3 modelo B +, mostrado en la Figura 2.12, cuenta con un procesador de cuatro núcleos de 64 bits que trabaja en un rango de 1,4 a 2,4 GHz y LAN inalámbrica, además de Bluetooth 4.2/BLE y la una interfaz Ethernet más rápida (RaspberryPi, 2018).

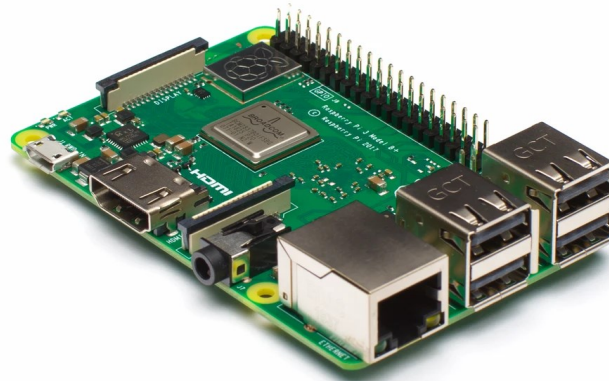


Figura 2.12: Raspberry Pi 3 B+

Fuente: (RaspberryPi, 2018)

La Tabla 2.4 indica las características de el Raspberry Pi 3 modelo B +, según (RaspberryPi, 2018).

Tabla 2.4: Especificaciones técnicas del Raspberry 3 B+

<b>Raspberry PI 3+</b>	
<b>Procesador</b>	Broadcom BCM 2837B0, Cortex-A53 64-bit SoC 1.4GHz
<b>Memoria</b>	1 GB LPDDR2 SDRAM
<b>Conectividad</b>	2.4 GHz y 5 GHz IEEE 802.11 wireless, Bluetooth 4.2, BLE, LAN; Gigabit Ethernet sobre USB 2.0; 4 puertos USB 2.0
<b>Acceso</b>	40 pines Gpio
<b>Sonido y video</b>	1 x HDMI; puerto MIPI DSI display; puerto MIPI CSI camera; puerto de video compuesto y salida estéreo
<b>Multimedia</b>	H.264, MPEG-4 y H.264 encode); Graphics OpenGL ES 1.1, 2.0
<b>Soporte tarjeta SD</b>	Micro SD para almacenamiento de datos
<b>Potencia de entrada</b>	5V/2.5A DC por conector micro USB; 5V DC por GPIO header; Alimentación a través de Ethernet (PoE) habilitada
<b>Tiempo de producción</b>	El Raspberry Pi 3 Model B+ seguirá en producción hasta al menos enero de 2023

Fuente: (RaspberryPi, 2018)

## 2.8. Arduino UNO

Arduino es un microcontrolador de código abierto. Es una placa de desarrollo simple y barata. Utiliza un lenguaje de programación de procesamiento basado en Java en el entorno de desarrollo y es fácil de aprender. Con entradas y salidas (analógicas y digitales), Arduino nos permite crear proyectos electrónicos de forma sencilla y eficaz. Actualmente, existe una sólida comunidad de desarrollo para presentar proyectos innovadores de código abierto, y hay muchos documentos que nos respaldan cuando escribimos código. (Proserquisa, 2016).

Ésta placa está establecida en Atmega328P. Cuenta con un total de 14 pines para entradas y salidas digitales, 6 de ellas pueden establecerse como salidas PWM. Con respecto a las entradas analógicas, esta microcontroladora cuenta con 6 pines, además de un resonador cerámico, conector de alimentación y conexión USB. Su nombre, UNO, se debe a que ésta fue la primera placa que se utilizó con el lanzamiento de Arduino Software (IDE) 1.0, por lo tanto, es la tarjeta insignia de Arduino (Arduino, 2019).

En la Figura 2.13 se observa el aspecto de la placa.

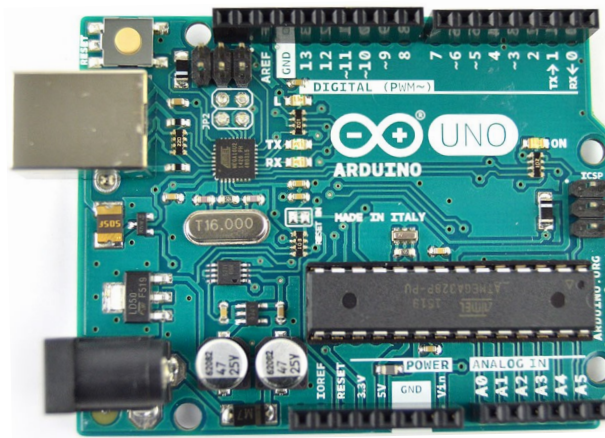


Figura 2.13: Arduino UNO

Fuente: (Arduino, 2019)

## 2.9. JSON

JSON es un formato de texto derivado de objetos de texto JavaScript para la serialización de datos estructurados. Puede representar cuatro tipos primitivos, como cadenas, números, valores booleanos y valores nulos, y se define en la tercera edición del estándar de lenguaje de programación ECMAScript (Moreano Abata, 2019).

### 2.9.1. Node.js

Node.js se considera un tiempo de ejecución de JavaScript asíncrono controlado por eventos diseñado para crear aplicaciones web escalables. Para cada conexión, se activará una devolución de llamada o *callback*, pero si no

hay trabajo por hacer, Node.js entrará en suspensión. Esto contrasta con el modelo de simultaneidad más común que usa subprocesos del sistema operativo en la actualidad. Las redes cableadas son relativamente ineficientes y difíciles de usar. Además, los usuarios de Node.js tienen la libertad de preocuparse por los bloqueos del proceso porque no existe. Existen pocas funciones en Node.js que realizan E/S directamente, por lo que el proceso nunca se bloqueará (NodeJS, 2019).

## 2.9.2. Node-RED

Node-RED Es una plataforma de programación que permite vincular, de una forma innovadora, hardware, servicios en línea y API.

### ■ Edición de flujos basado en navegador

Node-RED proporciona un editor de flujo basado en navegador, como se muestra en la Figura 2.14, que simplifica el flujo de conexión al usar múltiples nodos en la paleta. Con un solo clic, el proceso se puede implementar en *runtime*. Puede utilizar el editor de texto enriquecido para crear funciones de JavaScript en el editor. Además, su biblioteca incorporada le permite guardar funciones, plantillas o procesos útiles para uso repetido (Node-RED, 2019).

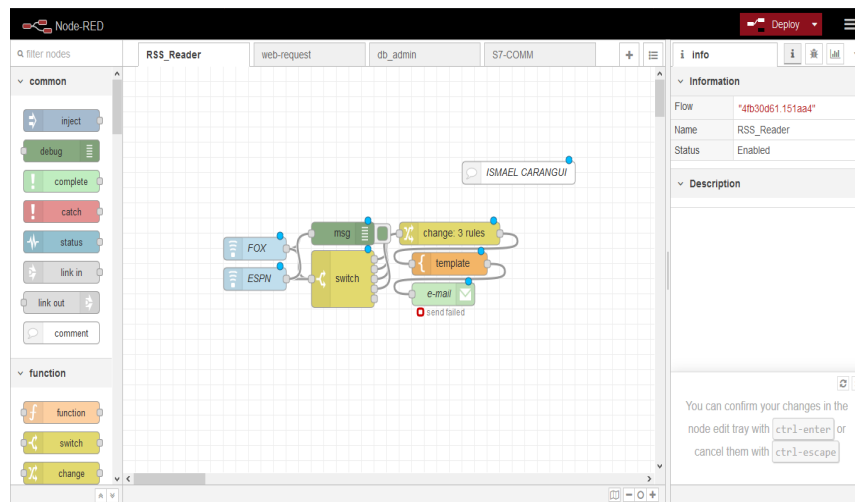


Figura 2.14: Interfaz gráfica de Node-RED

Fuente: Autor

### ■ Construcción en Node.js

Dado que está basado en Node.js, un modelo sin bloqueo basado en el control de eventos, se puede utilizar por completo, lo que significa operaciones de bajo costo de *hardware* en el borde de la red. Cuenta con más de 225.000 módulos y una gran cantidad de colaboradores en la comunidad (Node-RED, 2019).

### ■ Desarrollo social

Los flujos creados en Node-RED se almacenan en JSON y se pueden importar y exportar fácilmente para compartir con otros. La biblioteca de transmisión en línea le permite compartir las mejores transmisiones con el mundo (Node-RED, 2019).

### 2.9.3. Oracle MySQL Database Service

MySQL es un servicio de gestión de bases de datos SQL totalmente administrado que brinda a las organizaciones la oportunidad de trabajar sobre la base de datos más conocida en el mundo para efectuar aplicaciones nativas en la nube (Oracle, 2020).

MySQL como sistema gestor de base de datos, según (Alberto et al., 2015), brinda las siguientes prestaciones:

- Está desarrollado en C/C++.
- Los archivos ejecutables se distribuyen en aproximadamente 19 distintas plataformas.
- Python, Java, Eiffel, Perl, PHP, C, C ++, TCL y Ruby pueden ejecutar la API.
- Está optimizado para dispositivos multiprocesador.
- Se puede usar como servidor cliente o incrustado en una aplicación.
- Admite diferentes estrategias de almacenamiento para tablas con diferentes ventajas para mejorar la gestión de datos para cada situación específica.
- Además de ser totalmente compatible con ODBC, sus opciones de conexión también incluyen TCP/IP, sockets UNIX y sockets NT.
- Su gestión se establece en estructura de usuarios y privilegios.
- Hay evidencia de que procesó 50 millones de registros, 60,000 tablas y 5 millones de columna.

La Figura 2.15 muestra la herramienta visual de diseño de MySQL.

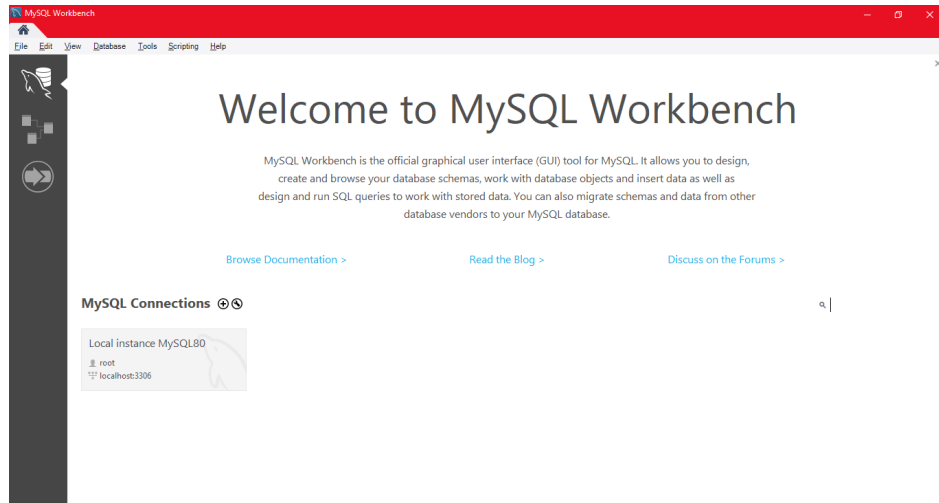


Figura 2.15: MySQL Workbench

Fuente: Autor

# Capítulo 3

## Diseño e implementación del sistema

### 3.1. Arquitectura del proyecto

La Figura 3.1 indica la estructura del sistema en términos de obtención de datos, comunicación, control y procesamiento de información desde el sistema SCADA de equipos heterogéneos, basado en los tres primeros niveles de la pirámide CIM.

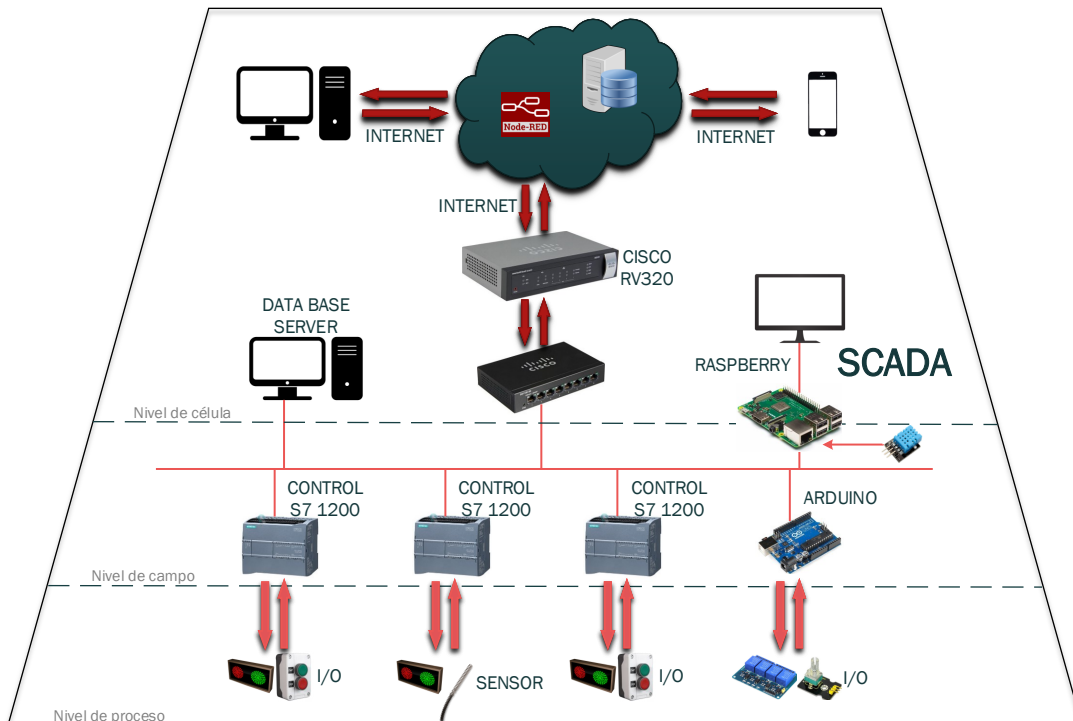


Figura 3.1: Arquitectura del proyecto

Fuente: Autor

## 3.2. Desarrollo de la red

### 3.2.1. Configuración de router (Cisco RV320)

Este es un enrutador para pequeñas empresas y se administra localmente. La creación de VLAN (*Virtual Local Area Network*), permite segmentar la red. Además, este router ofrece 2 interfaces WAN con equilibrio de cargas, configuración de VPN (*Virtual Private Network*) y otras herramientas.

En la Figura 3.2 puede ver la configuración de VLAN implementada en el enrutador.

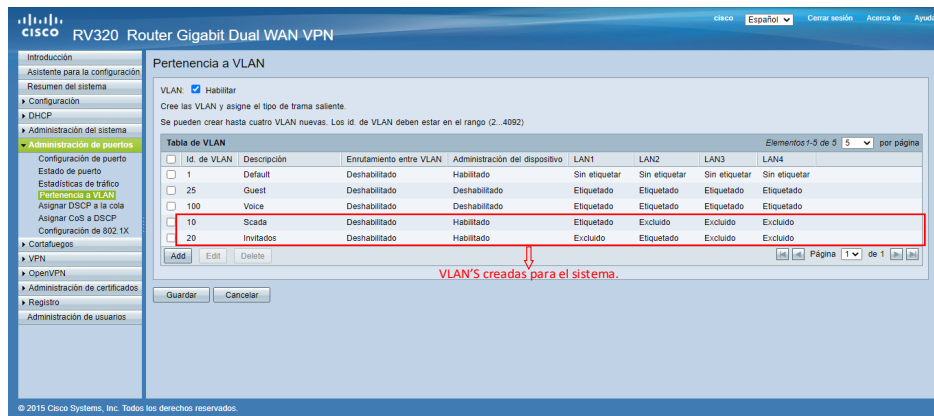


Figura 3.2: Configuración de VLAN en router CISCO

Fuente: Autor

Primero, creamos dos VLAN, "Scada" e "Invitados" para dividir la red. Para generar membresía, cada VLAN se asigna a un puerto determinado para su funcionamiento.

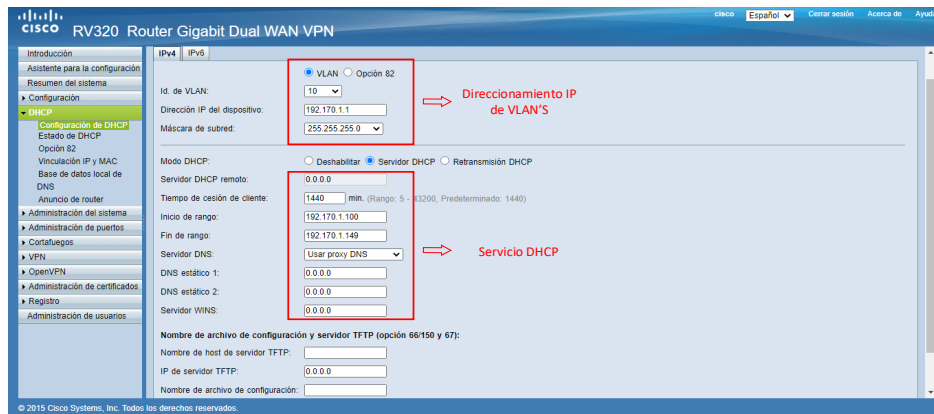


Figura 3.3: Direccionamiento y configuración de DHCP en router CISCO

Fuente: Autor

Luego, en la configuración DHCP, como se muestra en la Figura 3.3, se establece la dirección IP para cada VLAN. También, se habilita un servicio DHCP para que los equipos conectados puedan recibir direccionamiento dinámico.

### 3.2.2. Direccionamiento de la red

Luego de configurar el enrutador, conectamos los dispositivos pertenecientes a cada VLAN, como indica la Figura 3.4

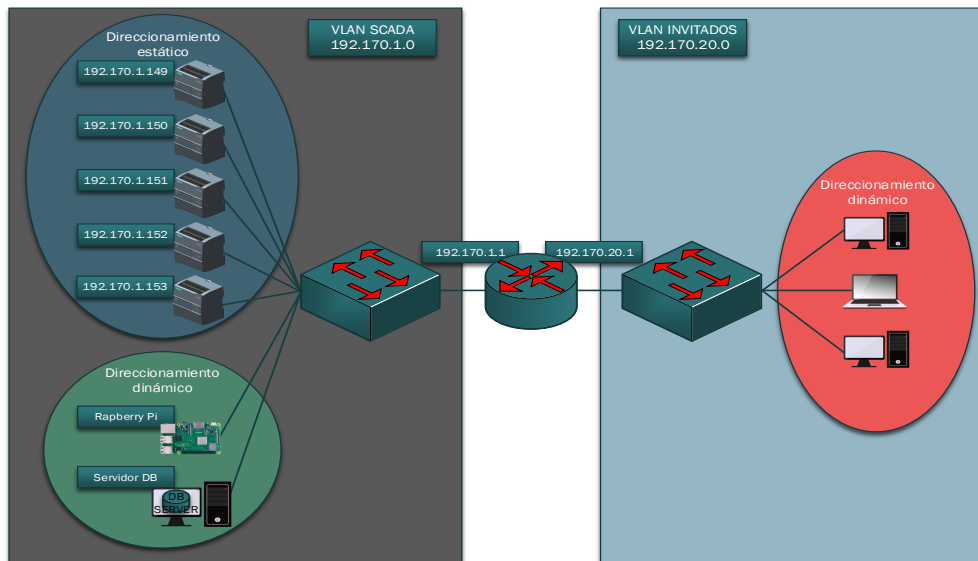


Figura 3.4: Topología de red

Fuente: Autor

Cada subred, al ser de máscara /24, puede albergar un total de 254 hosts con un rango de direcciones IP desde 192.170.1.1 hasta 192.170.1.254 para la VLAN "Scada" y un rango de direcciones IP desde 192.170.20.1 hasta 192.170.20.254 para la VLAN "Invitados".

Dado que la red no es extensa, se asigna direcciones IP estáticas a los PLC para poder gestionarlos de forma segura y mantener siempre la misma ruta al destino. Por otro lado, otros dispositivos (como Raspberry, servidores de bases de datos y otras computadoras) reciben direccionamiento DHCP, ajustando así automáticamente la topología de la red para redirigir el tráfico en caso de falla.

### 3.3. Configuración de PLCs

Se utiliza tres PLC Siemens S7-1200 interconectados por medio de un Switch CISCO. Además, se hace uso de la herramienta TIA PORTAL v15 de Siemens para la programación de estos dispositivos.

Como configuración inicial, se debe establecer los PLC's dentro de la misma red y comprobar su interconexión, como indica la Figura 3.5.

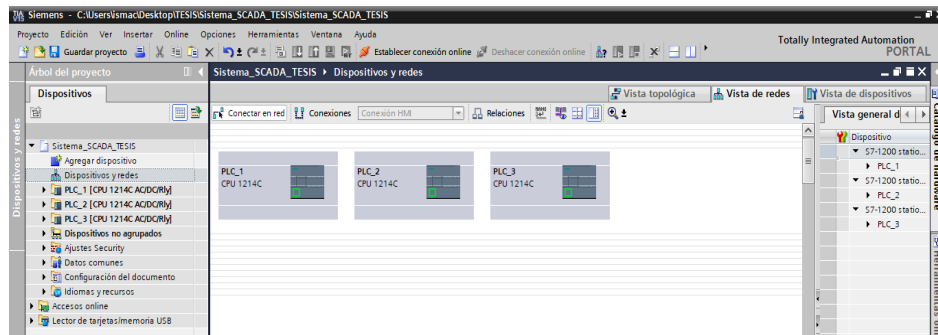


Figura 3.5: Interconexión de PLC's en TIA PORTAL

Fuente: Autor

La Figura 3.6 expone la programación KOP para el control de salidas del sistema. A cada contactor se le asigna una etiqueta responsable de controlar el estado de la salida del PLC asignada.

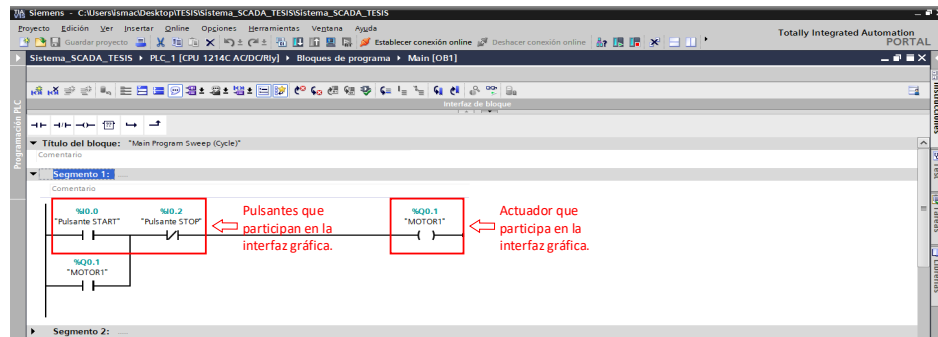


Figura 3.6: Primer bloque de programa en TIA PORTAL 15 con el esquema de contactores

Fuente: Autor

De manera similar, la Figura 3.7 muestra la programación basada en diagramas de funciones, donde cada bloque realiza una función determinada y recibe parámetros de entrada para procesar su salida.

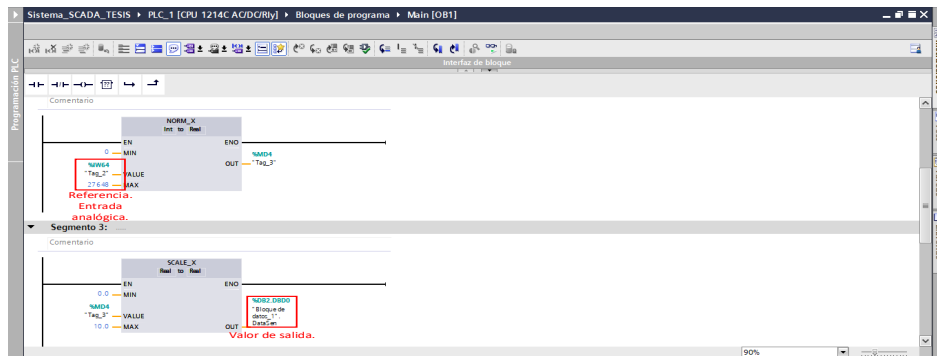


Figura 3.7: Segundo bloque de programa en TIA PORTAL 15 con el esquema de diagrama de funciones  
Fuente: Autor

En el primer bloque, se configura el control de encendido y apagado del relés, y las salidas del PLC son monitoreadas en Node-Red. En cambio, el segundo bloque es utilizado para recibir una señal analógica de un sensor. El valor de ingreso debe ser normalizado y escalado para ser presentado en función de voltaje de salida.

### 3.4. Configuración de Arduino

Para la sincronización de Arduino con el servidor de Node-RED es necesario configurar el protocolo genérico de comunicación con microcontroladores: **FIRMATA**. Este protocolo permite controlar íntegramente el Arduino desde Node-RED sin escribir una sola línea de código de Arduino.

Aplicando el modelo más común de uso de FIRMATA, se carga el Sketch de propósito general llamado **StandardFirmata**. Esto nos permitirá controlar gráficamente las entradas y salidas requeridas por el sistema a través del editor de Node-RED. La Figura 3.8 muestra la ubicación del sketch en el IDE de Arduino, que es necesario para que FIRMATA establezca la comunicación.

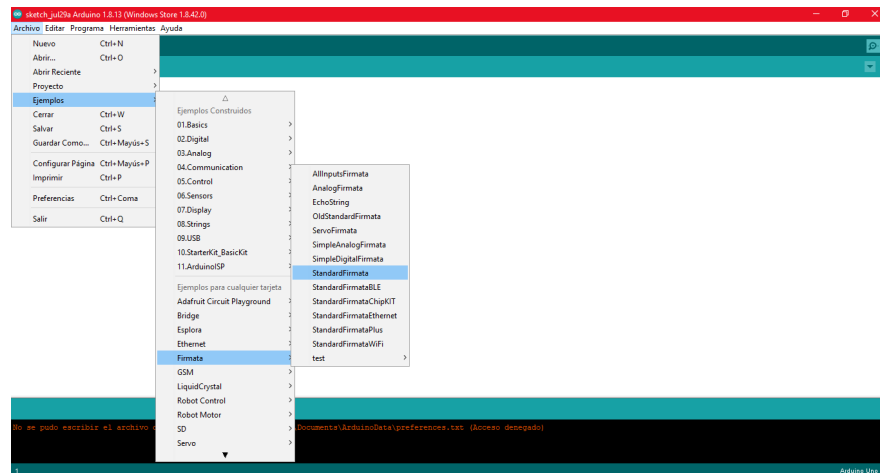


Figura 3.8: Ubicación de sketch de FIRMATA en el IDE de Arduino

Fuente: Autor

## 3.5. Creación del servicio de Node-RED

### 3.5.1. Instalación de Node.js en Raspberry Pi

Debemos recordar que Node-RED trabaja bajo el motor Node.js, que es un servidor capaz de ejecutar código javascript. Sabiendo esto, lo primero que debemos hacer es instalar Node.js mediante el siguiente comando ejecutado en la terminal de la Raspberry:

```
>>> sudo apt install nodejs
```

### 3.5.2. Instalación de Node-RED en Raspberry Pi

Una vez instalado Node.js, podemos usar su administrador de paquetes y desde allí instalar NodeRed con el siguiente comando:

```
>>> sudo npm install -g --unsafe-perm node-red
```

Ahora, solo necesitamos ejecutar Node-RED desde el terminal Raspberry o configurarlo como un servicio.

### 3.5.3. Ejecutar NodeRed como un servicio

Desde la consola de Node-RED es posible configurar para que se establezca como un servicio. Es decir, Node-RED se ejecutará en segundo plano y se habilitará automáticamente cuando arranca el equipo.

La Figura 3.9 indica los comandos a ejecutarse en la consola, para habilitar o deshabilitar Node-RED como servicio.

```

Node-RED console
Archivo  Editar  Pestañas  Ayuda

Start Node-RED

Once Node-RED has started, point a browser at http://192.170.1.116:1880
On Pi Node-RED works better with the Firefox or Chrome browser

Use  node-red-stop           to stop Node-RED
Use  node-red-start         to start Node-RED again
Use  node-red-log           to view the recent log output
Use  sudo systemctl enable nodered.service to autostart Node-RED at every boot
Use  sudo systemctl disable nodered.service to disable autostart on boot

To find more nodes and example flows - go to http://flows.nodered.org
Starting as a systemd service.

```

Figura 3.9: Instalación Node-RED en la terminal de Raspberry

Fuente: Autor

## 3.6. Desarrollo del sistema SCADA en Node-RED

### 3.6.1. Conexión de PLC con Node-RED

Para establecer una conexión entre Node-RED y los PLCs Siemens S7, es necesario instalar el nodo *node-red-contrib-s7*. Este nodo está diseñado exclusivamente para la comunicación con los PLC SIMATIC S7-300/400/1200/1500 y se basa en el protocolo de comunicación Ethernet Siemens RFC1006.

NodeS7 se encarga de la reconexión de forma autónoma. De modo que, si se pierde el enlace por una desconexión del PLC, puede seguir solicitando datos sin tomar ninguna otra medida. Devuelve valores **incorrectos** y, finalmente, la conexión se repondrá automáticamente (Node-RED, 2019).

La instalación del nodo S7, se puede realizar por medio de comando en la terminal de la Raspberry o directamente desde el editor de Node-RED en la sección de **Manage palette**.

Como se muestra en la Figura 3.10, cada nodo de configuración S7 representa una conexión con un PLC. En este caso, tenemos 3 nodos de salida (utilizados para controlar los relés) y un nodo de entrada (para comunicarse con el sensor conectado a la entrada analógica de uno de los PLC).

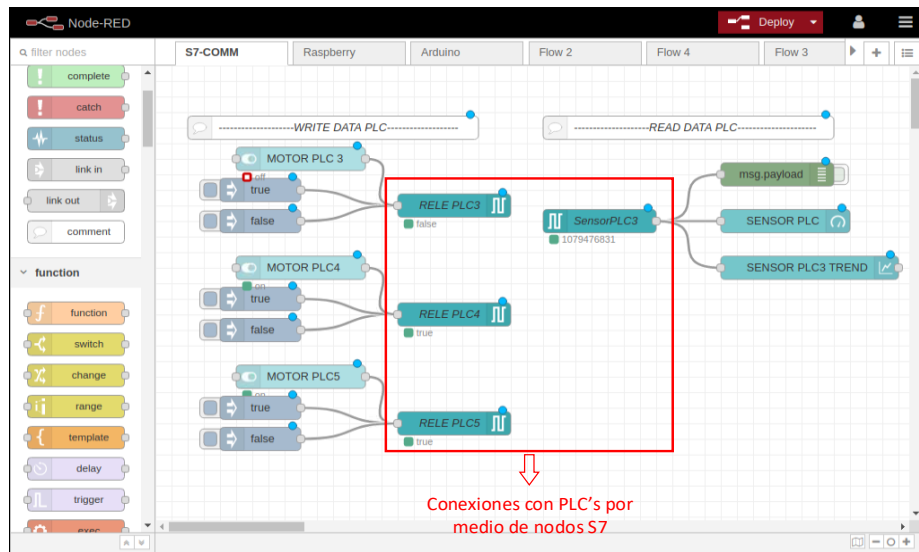


Figura 3.10: Nodos S7 en editor de Node-RED

Fuente: Autor

Para poder configurar el nodo S7, es necesario ingresar a dicho nodo y crear una nueva instancia para de comunicación con el PLC. La Figura 3.11 muestra la ventana que se despliega para realizar la parametrización.

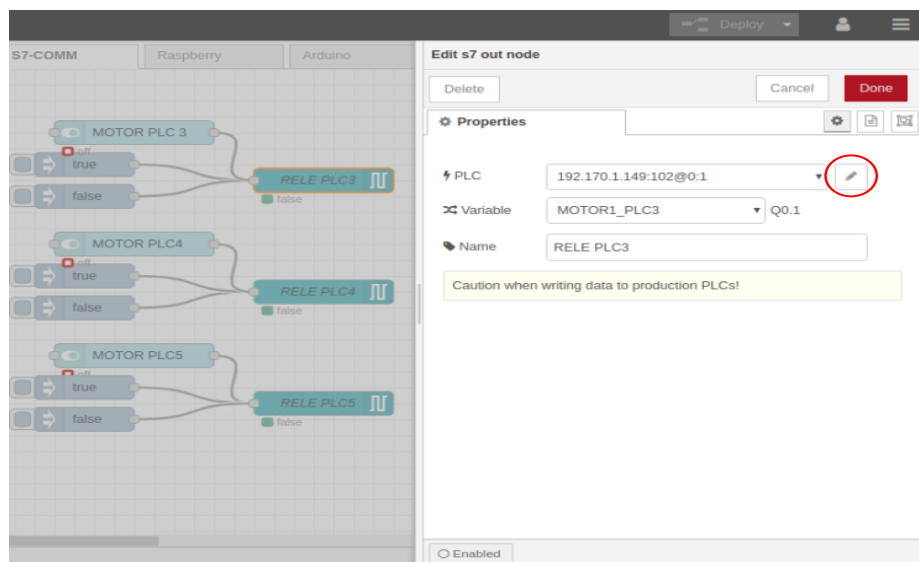


Figura 3.11: Ventana de configuración del nodo S7

Fuente: Autor

Al ingresar al botón de configuración del nodo, encontraremos dos ventanas. Como se puede observar en la Figura 3.12, en la primera ventana (izquierda) aparecen los parámetros de configuración para la conexión (protocolo de comunicación, IP del PLC, Port, Slot). La segunda (derecha) muestra la lista de variables vinculadas con el PLC.

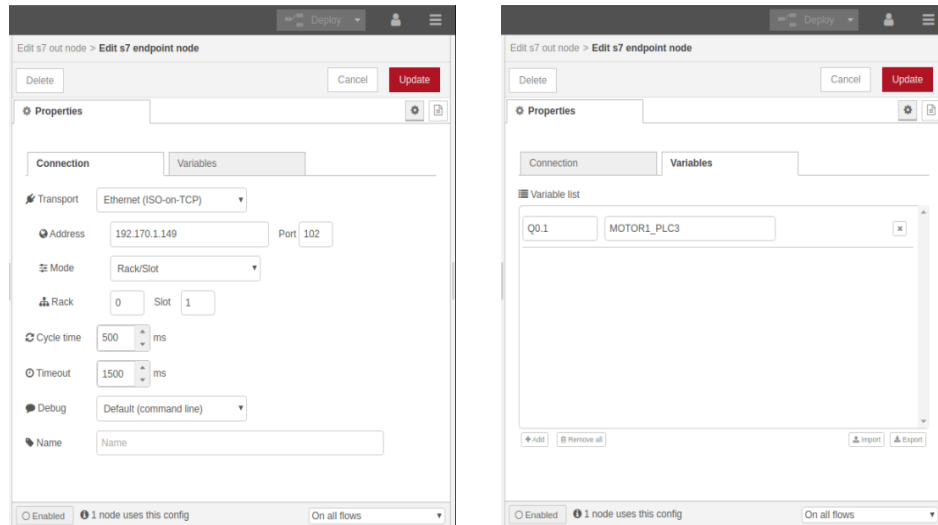


Figura 3.12: 1 Parámetros de comunicación del nodo S7 (izquierda), 2 Variables del PLC (derecha)

Fuente: Autor

Es importante mencionar que el direccionamiento de variables en el nodo S7 sigue un esquema sutilmente diferente al utilizado en TIA Portal. La Tabla 3.1 especifica algunos ejemplos con sus equivalencias.

Tabla 3.1: Direccionamiento de variables

Address	Step7 equivalent	JS Data type	Description
DB5,X0.1	DB5.DBX0.1	Boolean	Bit 1 of byte 0 of DB 5
DB23,B1 or DB23,BYTE1	DB23.DBB1	Number	Byte 1 (0-255) of DB 23
DB100,C2 or DB100,CHAR2	DB100.DBB2	String	Byte 2 of DB 100 as a Char
DB42,I3 or DB42,INT3	DB42.DBW3	Number	Signed 16-bit number at byte 3 of DB 42
DB57,WORD4	DB57.DBW4	Number	Unsigned 16-bit number at byte 4 of DB 57
DB13,DI5 or DB13,DINT5	DB13.DBD5	Number	Signed 32-bit number at byte 5 of DB 13
DB19,DW6 or DB19,DWORD6	DB19.DBD6	Number	Unsigned 32-bit number at byte 6 of DB 19
DB21,R7 or DB21,REAL7	DB21.DBD7	Number	Floating point 32-bit number at byte 7 of DB 21
I1.0 or E1.0	I1.0 or E1.0	Boolean	Bit 0 of byte 1 of input area
Q2.1 or A2.1	Q2.1 or A2.1	Boolean	Bit 1 of byte 2 of output area
M3.2	QM3.2	Boolean	Bit 2 of byte 3 of memory area
QB5 or AB5	QB5 or AB5	Number	Byte 5 (0 -255) of output area
MB6	MB6	Number	Byte 6 (0 -255) of memory area

Fuente: (Node-RED, 2019)

Dado que los PLC más nuevos proporcionan una versión más amplia del protocolo S7, y NodeS7 tiene su versión básica, se requieren los siguientes ajustes en el PLC.

- En la configuración del CPU y en el apartado de **Protección**, como se muestra en la Figura 3.13, es necesario marcar la casilla **Sin protección** para otorgar acceso completo de lectura y escritura.

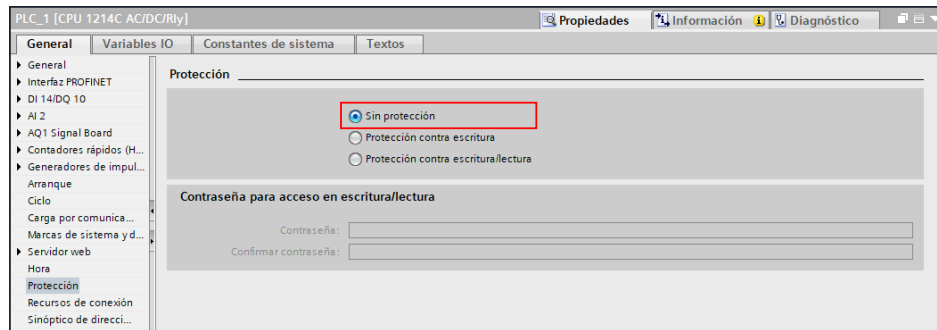


Figura 3.13: Configuración de protección del CPU en TIA Portal

Fuente: Autor

- En los DB, en la sección de **Atributos**, como indica la Figura 3.14, debemos deshabilitar la casilla **Acceso optimizado al bloque**.

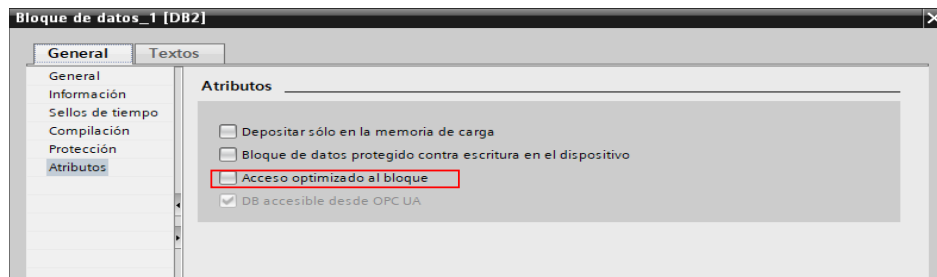


Figura 3.14: Configuración de atributos de bloques de programa en TIA Portal

Fuente: Autor

### 3.6.2. Conexión del arduino con Node-RED

Como primer paso, debemos instalar el nodo Arduino en Node-RED. Para ello, podemos ir a la sección **Manage palette**, en el editor de Node-RED y buscar *node-red-node-arduino* o ejecutar el siguiente código directamente en la terminal Raspberry:

```
>>> npm i --unsafe --perm node-red-node-arduino
```

Luego se debe cargar el firmware Firmata en el Arduino como se muestra en la Figura 3.8, lo que nos permitirá acceder directamente a los pines de entrada y salida.

Para nuestro sistema, se han implementado 3 nodos Arduino para controlar la salida de diferentes pines vinculados a los relés. Además de un nodo de entrada, el cual relaciona el sensor conectado a la entrada analógica del Arduino con Node-RED. La implementación de los nodos se puede ver en la Figura 3.15

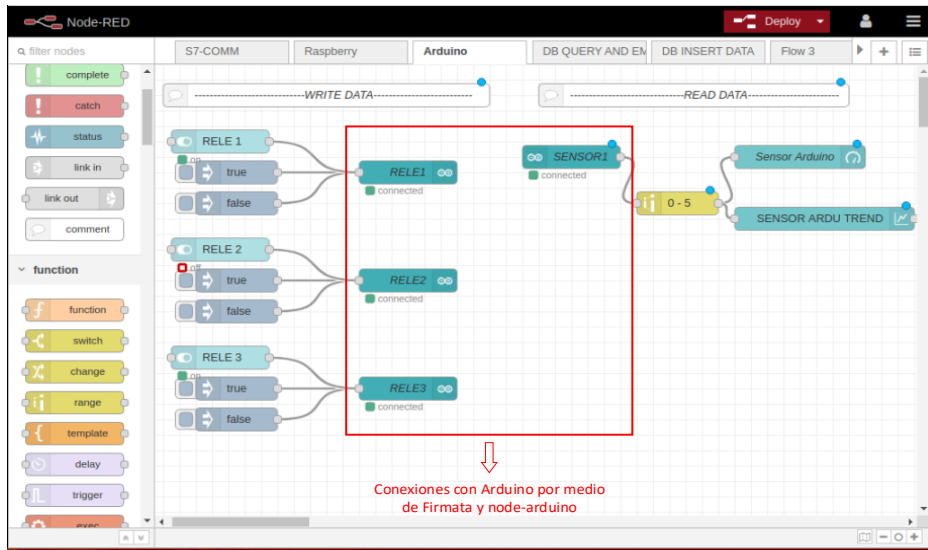


Figura 3.15: Nodos Arduino en editor de Node-RED

Fuente: Autor

Al ingresar a la configuración del nodo, encontramos los parámetros (puerto, tipo de dato, pin y etiqueta del nodo) que deben configurarse para establecer la comunicación. Dependiendo del nodo, puede configurar el tipo de entrada (digital, pull-up, analógica o cadena) o el tipo de salida (digital, analógica, servo o cadena) vinculado al pin Arduino.

En la Tabla 3.2, puede observar con más detalle el tipo de nodo y los datos asignados por cada tipo.

Tabla 3.2: Tipo de nodos nodeArduino

Node Type	Data Type
<b>Input Node</b>	Digital
	Pullup
	Analogue
	String
<b>Output Node</b>	Digital (accepts 0, 1, true, false, on, off)
	Analogue (PWM - accepts Integer 0 to 255)
	Servo (accepts Integer 0 - 180)
	String (to send a String to the Arduino)

Fuente: (Node-RED, 2019)

La Figura 3.16 muestra los ejemplos de configuración de los nodos **ArduinoInput** y **ArduinoOutput**, en los que se han definido puertos, tipos de datos y pines de interacción.

En el nodo de salida se define un dato de tipo **Digital (0/1)**, lo que permite abrir y cerrar el relé conectado al pin 9. Para otros relés asignados a la salida, este proceso se repetirá y se asignarán a diferentes pines del Arduino. Para el nodo de entrada, los datos son del tipo **Analogue pin**, y el pin A0 del Arduino se asigna a un sensor analógico, el cual envía un valor de 0 a 1024, y debe ser normalizado posteriormente para mostrarse en la unidad correcta.

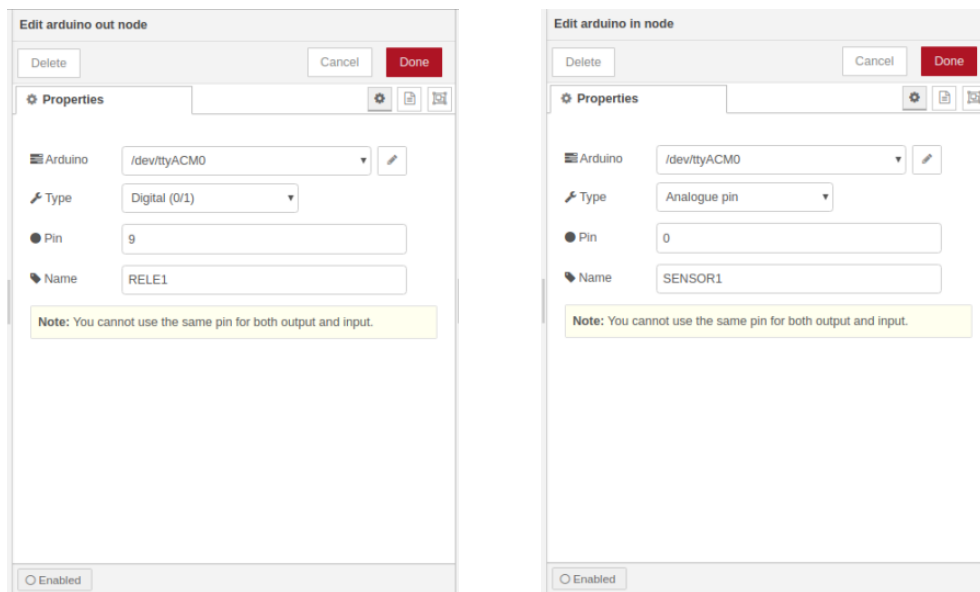


Figura 3.16: 1. Configuración del nodo ArduinoInput (izquierda), 2. Configuración del nodo ArduinoOutput (derecha)  
Fuente: Autor

Es muy importante saber que no puede utilizar el IDE de Arduino y nodos Arduino al mismo tiempo, ya que esto provocará conflictos. Para reprogramar el Arduino desde el IDE, es necesario detener Node-RED.

### 3.6.3. Control de pines GPIO de Raspberry con Node-RED

Node-RED en Raspbian cuenta con un nodo preconfigurado llamado *node-red-node-gpio*. Esta librería permite controlar y monitorear fácilmente los pines GPIO.

En los nodos de colores del lado izquierdo de la interfaz, encontraremos el nodo rotulado como **Raspberry Pi**, como se muestra en la Figura 3.17. Aquí, veremos dos nodos etiquetados como **rpi gpio**. Son estos los que usaremos para comunicarnos con los pines GPIO en la Raspberry Pi.

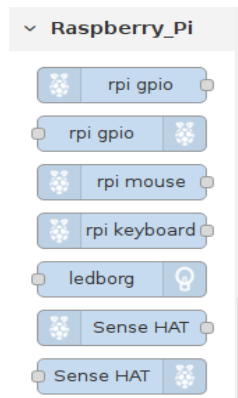


Figura 3.17: Nodos Raspberry Pi

Fuente: Autor

El segundo nodo, que tiene el icono de Raspberry a la derecha, es utilizado para las salidas. Pueden ser utilizados para encender o apagar LEDs o controlar relés, como es el caso de este sistema. Luego de ingresar al nodo, aparecerá un cuadro, como indica la Figura 3.18, que nos permitirá configurarlo. Seleccionamos en el diagrama de pines el GPIO con el que deseamos interactuar y agregamos una etiqueta de ser necesario.

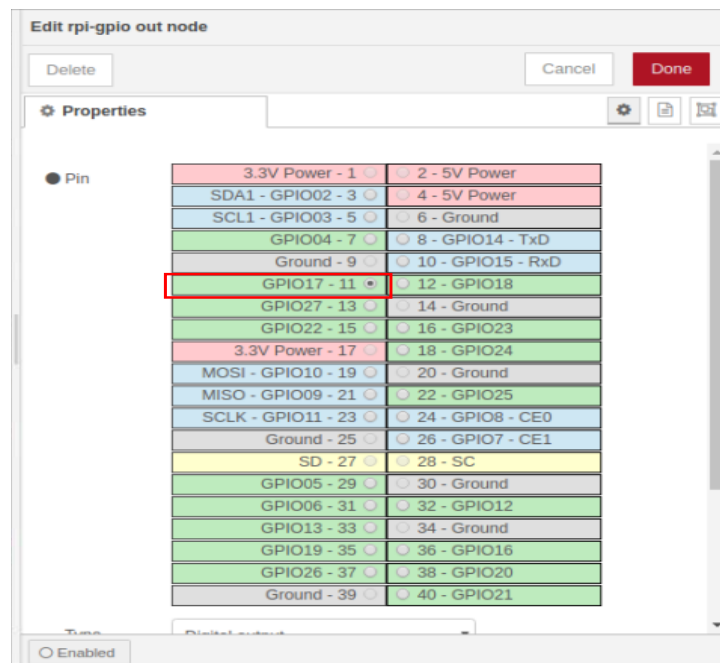


Figura 3.18: Ventana de configuración del nodo rpi-gpio out en Node-RED

Fuente: Autor

Para el nodo de entrada tenemos un sensor DHT11, que nos proporcionará valores de temperatura y humedad. La biblioteca DHT tiene nodos de entrada específicos para estos sensores. Para instalarlo debemos ir a la sección **Manage panel** en el editor y buscar el nodo *node-red-contrib-dht-sensor*.

Después de instalar el nodo, ingresamos a él para configurar los parámetros de conexión. En las propiedades escogemos el modelo del sensor, en este caso es el DHT11. Además, elegimos el pin de la Raspberry donde estará conectado el sensor, como muestra la Figura 3.19.

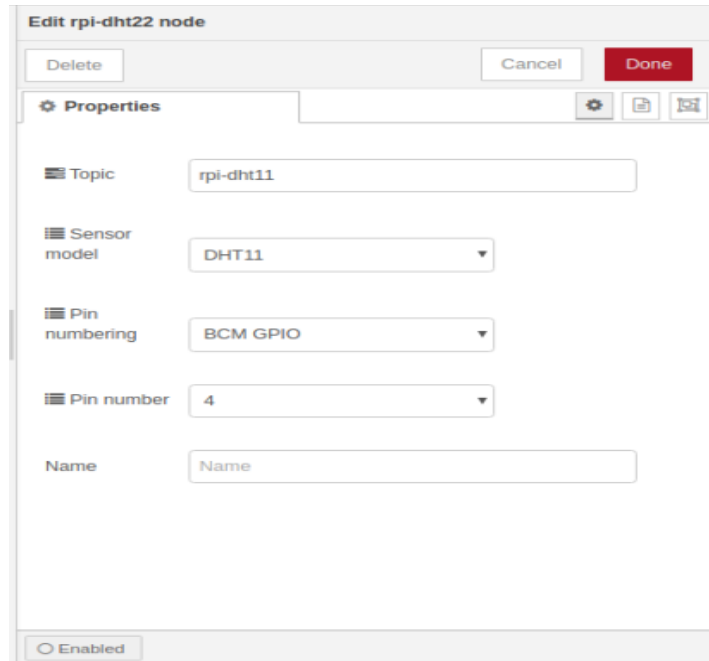


Figura 3.19: Ventana de configuración del nodo DHT en Node-RED

Fuente: Autor

## 3.7. Base de datos

### 3.7.1. Servidor MySQL

Para registrar los datos de los sensores del sistema, se ejecuta el servidor de base de datos MySQL desde un *host* externo a la Raspberry. Una vez instalado MySQL en el host asignado para esta función, en una nueva base de datos se crean tablas para el registro del sensor del PLC1 y del sensor DHT11 de la Raspberry. Para hacer esto, en **MySQL Workbench**, se ejecutan las líneas de comando en que se muestran a continuación:

```
CREATE TABLE 'nodedata'. 'dataplc1' (
  'id' INT NOT NULL AUTO_INCREMENT,
  'Fecha' DATETIME NOT NULL,
  'SensorAna' FLOAT NOT NULL,
  PRIMARY KEY ('id'));
```

```
CREATE TABLE 'nodedata'. 'datahumras' (
  'id' INT NOT NULL AUTO_INCREMENT,
  'Fecha' DATETIME NOT NULL,
  'HumDHT11' FLOAT NOT NULL,
  PRIMARY KEY ('id'));
```

```
CREATE TABLE 'nodedata'. 'datatempras' (
  'id' INT NOT NULL AUTO_INCREMENT,
  'Fecha' DATETIME NOT NULL,
  'TempDHT11' FLOAT NOT NULL,
  PRIMARY KEY ('id'));
```

La Figura 3.20 muestra las tablas que gestionan los datos de los sensores.

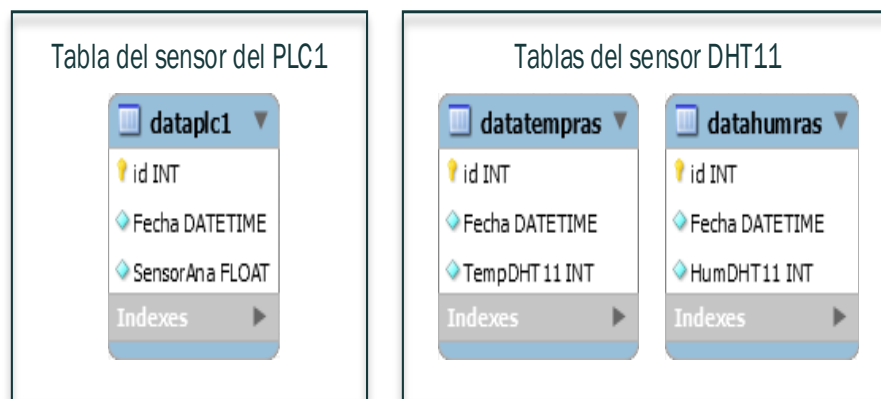


Figura 3.20: Tablas para almacenamiento de datos en MySQL

Fuente: Autor

Debido a que la base de datos se ejecuta en un *host* distinto a la Raspberry desde donde se realiza la lectura y escritura, debe ejecutarse los siguientes comandos en la base de datos para otorgar privilegios al usuario para que admita *hosts* externos.

```
GRANT ALL PRIVILEGES ON *.* TO 'root' '@' '%'
WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

Donde “root” es el usuario y “%” es el parámetro para identificar el *host* donde se encuentra el servidor MySQL.

### 3.7.2. MySQL en Node-RED

Una vez que se crea las tablas para almacenar los datos en MySQL, debe instalar el paquete *node-red-node-mysql* desde el editor Node-RED o ejecutar el siguiente comando en la terminal de Raspberry:

```
>>> npm i node-red-node-mysql
```

Este nodo otorga acceso básico a la base de datos MySQL. Utiliza operaciones de consulta en la base de datos configurada, lo cual permite funciones de lectura y escritura.

En la sección nodos del editor Node-RED, encontraremos el nodo llamado **mysql**. Su icono se muestra en la Figura 3.21.



Figura 3.21: Nodo MySQL en Node-RED

Fuente: Autor

Después de ingresar a este nodo, se mostrará la ventana de configuración, en la que se creará una nueva instancia de comunicación. Aquí debe registrar las credenciales (*Host*, *Port*, *User*, *Password* y *Database*) de la base de datos creada en MySQL. La Figura 3.22 muestra la configuración de la base de datos para este sistema.

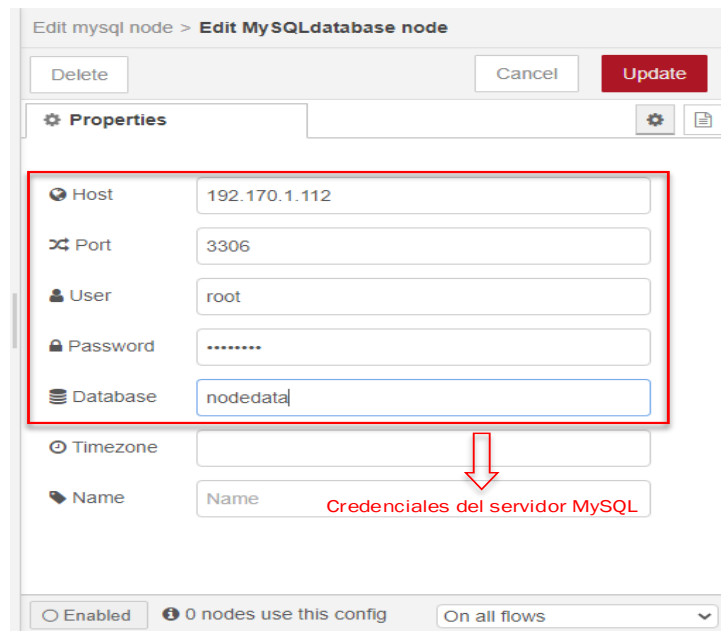


Figura 3.22: Ventana de configuración del nodo mysql en Node-RED

Fuente: Autor

## 3.8. Interfaz gráfica

Permite el intercambio de datos obtenidos del sistema y de los usuarios, que serán visualizados de forma sencilla y práctica, y cuenta con las siguientes funciones:

- Pantalla de control de salidas.
- Lectura de sensores en tiempo real.
- Visualiza historiales de variables en tiempo real.
- Actúa sobre base de datos, escribiendo o solicitando datos.

### 3.8.1. Dashborad en Node-RED

Se utiliza la librería *node-red-dashboard* en Node-RED para el control del sistema SCADA. El propósito de esta interfaz es permitir que los operadores manipulen el sistema de control a nivel de planta.

Después de instalar la biblioteca desde el **Manage panel**, en la sección Nodos en el lado izquierdo del editor Node-RED, encontramos todas las herramientas, mostrados en la Figura 3.23, utilizadas para generar la interfaz.

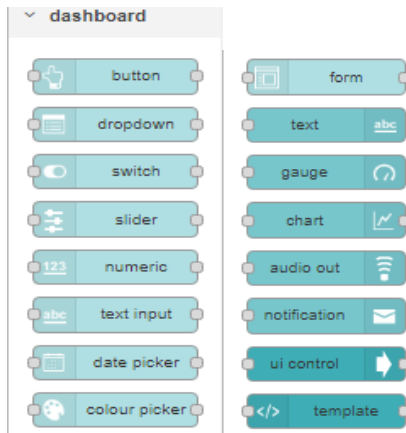


Figura 3.23: Nodos Dashboard en Node-RED

Fuente: Autor

El flujo mostrado en la Figura 3.24, permite al usuario actuar sobre los pines de la Raspberry (los nodos de colores de la parte derecha), utilizando los nodos **button** y asignando cada uno a un PIN, enciende o apaga los LED's del sistema. También soporta el control de la base de datos de MySQL (nodos de colores de la parte izquierda), que, al hacer clic sobre los botones, puede realizar consultas o vaciar la base de datos. Además, muestra un gráfico de lectura de datos en tiempo real y un historiadur del sensor DHT11 conectado a la Raspberry, utilizando los nodos **gauge** y **chart**, respectivamente.

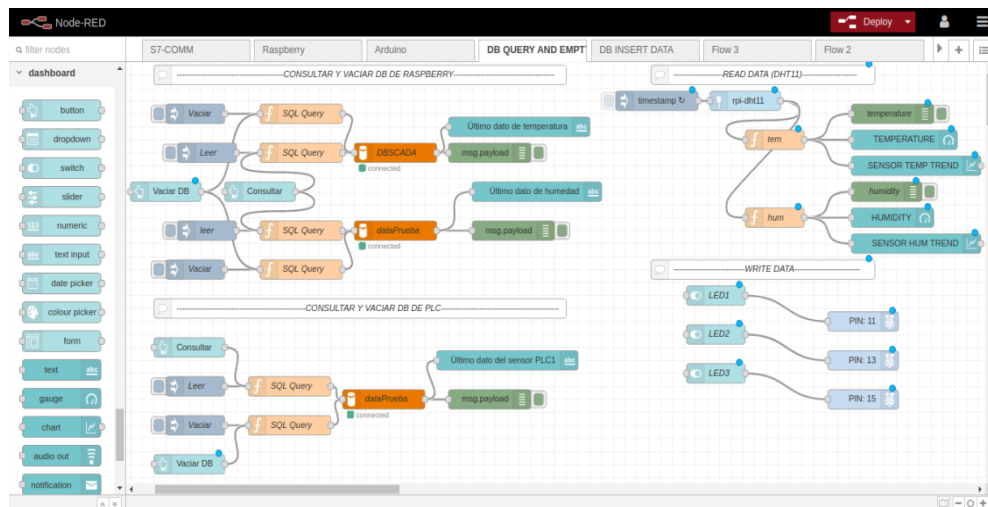


Figura 3.24: Flujo para control de LED's y realizar consultas de base de datos en Node-RED

Fuente: Autor

Como se muestra en la Figura 3.25, todos estos nodos deben organizarse en grupos para que los usuarios puedan encontrar fácilmente el tablero. Los grupos de paneles determinan cómo se muestran en el selector de paneles.

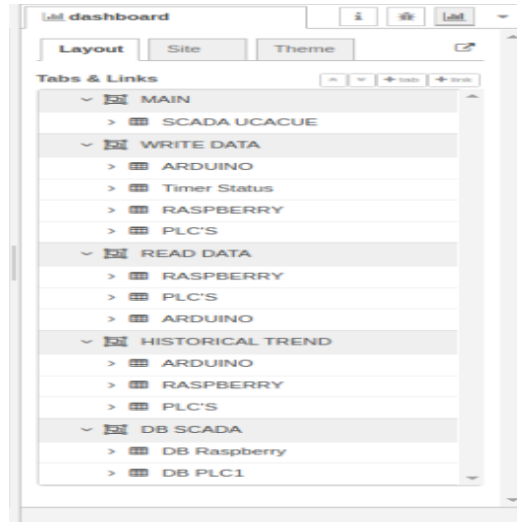


Figura 3.25: Ventana de grupos de dashboard en Node-RED

Fuente: Autor

Si se utiliza las configuraciones predeterminadas, la interfaz UI se encuentra disponible en <http://localhost:1880/ui>. En caso de solicitar acceso desde un equipo externo, debemos digitar la dirección IP de la Raspberry donde se encuentra el servidor Node-RED y mantener el puerto **1880** con la extensión **/ui**.

### 3.8.2. Administración de usuarios

La administración de usuarios y grupos es una parte importante de la gestión del sistema dentro de una organización. Las cuentas de usuario son una forma de identificar y autenticar a las personas con del sistema. Las cuentas de usuario tienen diferentes componentes. Primero, hay un nombre de usuario, luego está la contraseña y finalmente la información de control de acceso.

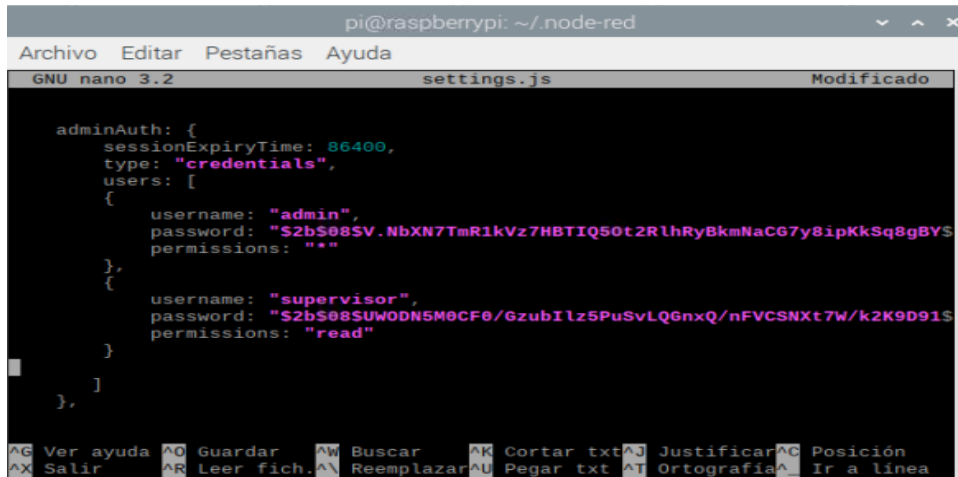
Según (Node-RED, 2019), la interfaz de administración y el editor asienten dos métodos de autenticación:

- Basada en credenciales de nombre de usuario/contraseña.
- Contra cualquier proveedor OAuth/OpenID.

En esta ocasión utilizamos la autenticación basada en nombre de usuario/contraseña. Para poder habilitar esta herramienta, debemos ubicarnos al archivo de configuración de Node-RED **settings.js**, ejecutando las siguientes instrucciones en la línea de comando de Raspberry.

```
pi@raspberrypi:~ $ cd .node-red
pi@raspberrypi:~/node-red $ nano settings.js
```

Como se muestra en la Figura 3.26, se han creado dos usuarios. El primero definido como **admin**, tiene permiso para realizar todas las operaciones en el editor y tiene una contraseña de **password**. El otro, llamado **supervisor**, tiene acceso de solo lectura.



```

pi@raspberrypi: ~/.node-red
Archivo  Editar  Pestañas  Ayuda
GNU nano 3.2  settings.js  Modificado

adminAuth: {
  sessionExpiryTime: 86400,
  type: "credentials",
  users: [
    {
      username: "admin",
      password: "$2b$08$V.NbXN7TmR1kVz7HBTIQ50t2RlhRyBkmNacG7y8ipKkSq8gBYS",
      permissions: "*"
    },
    {
      username: "supervisor",
      password: "$2b$08$UWODN5M0CF8/6zubI1z5PuSvLQGnxQ/nFVCSNXt7W/k2K9D91S",
      permissions: "read"
    }
  ]
}
),
]
),

AG Ver ayuda  AO Guardar  AW Buscar  AK Cortar txt  AJ Justificar  AC Posición
AX Salir  AR Leer fich.  AN Reemplazar  AU Pegar txt  AT Ortografía  AL Ir a línea

```

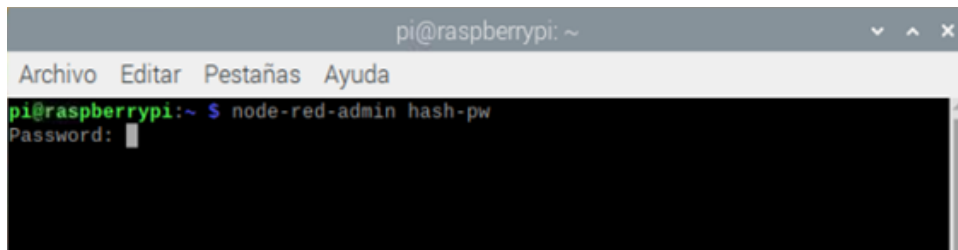
Figura 3.26: Archivo de configuración de Node-RED

Fuente: Autor

Para una correcta configuración de los usuarios, es necesario generar el hash de las contraseñas. Para esto es preciso instalar la herramienta **Command-line Administration** de Node-RED ejecutando el siguiente comando en terminal de Raspberry:

```
>>> npm install -g --unsafe-perm node-red-admin
```

Luego de instalar esta herramienta, a través del comando que se muestra en la Figura 3.27, generamos un valor hash de la contraseña, que será pegado en la sección de password al crear un usuario.



```

pi@raspberrypi: ~
Archivo  Editar  Pestañas  Ayuda
pi@raspberrypi:~ $ node-red-admin hash-pw
Password:

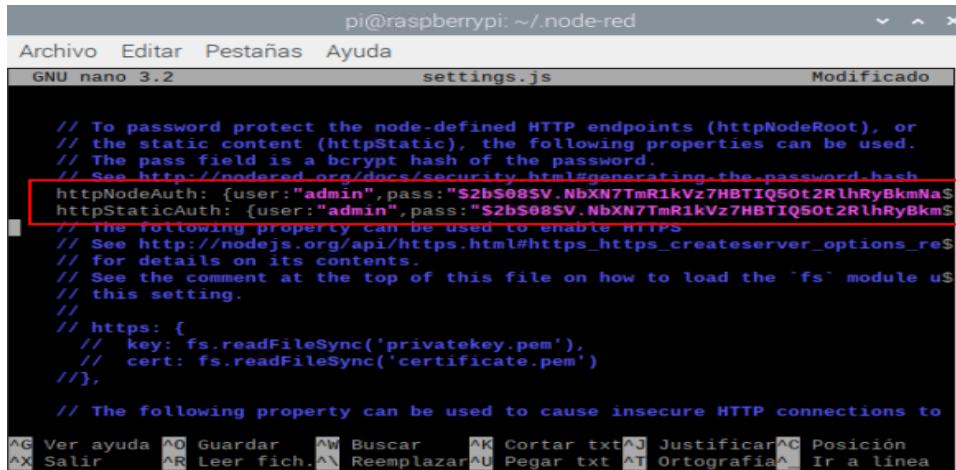
```

Figura 3.27: Creación de hash de contraseña en Raspberry

Fuente: Autor

### 3.8.3. Protección de la interfaz UI de Node-RED

Es posible proteger la interfaz UI de NodeRed, para ello necesitamos descomentar **httpNodeAuth:** y **httpStaticAuth:**. Mediante la declaración de una contraseña y usuario, estas propiedades permiten limitar el acceso a las rutas. Luego se crea el usuario que administrará la interfaz, como se muestra en la Figura 3.28, y se agrega un hash de contraseña.



```
pi@raspberrypi: ~/.node-red
Archivo Editar Pestañas Ayuda
GNU nano 3.2 settings.js Modificado
// To password protect the node-defined HTTP endpoints (httpNodeRoot), or
// the static content (httpStatic), the following properties can be used.
// The pass field is a bcrypt hash of the password.
// See http://nodered.org/docs/security.html#generating-the-password-hash
httpNodeAuth: {user:"admin",pass:"S2bS08$V.NbXN7TmR1kVz7HBTIQ50t2RlhRyBkmNa$"}
httpStaticAuth: {user:"admin",pass:"S2bS08$V.NbXN7TmR1kVz7HBTIQ50t2RlhRyBkmNa$"}
// The following property can be used to enable HTTPS
// See http://nodejs.org/api/https.html#https_https_createserver_options_reS
// for details on its contents.
// See the comment at the top of this file on how to load the 'fs' module us
// this setting.
//
// https: {
//   key: fs.readFileSync('privatekey.pem'),
//   cert: fs.readFileSync('certificate.pem')
// },
// The following property can be used to cause insecure HTTP connections to
^C Ver ayuda ^O Guardar ^W Buscar ^K Cortar txt ^J Justificar ^G Posición
^X Salir ^AR Leer fich. ^A Reemplazar ^AU Pegar txt ^AI Ortografía ^A Ir a línea
```

Figura 3.28: Configuración de seguridad HTTP en el archivo de configuración de Node-RED

Fuente: Autor

## 3.9. Repotenciación de tableros de control

El laboratorio de ingeniería electrónica, cuenta con seis tableros de control. Cada tablero cuenta con un PLC, un tablero de contactos, tres contactores, luces, interruptores y pulsadores. Para iniciar la repotenciación de los tableros, mostrado en la Figura 3.29, primero se determina el estado actual de los tableros de control para determinar el mantenimiento y modificaciones que se deben realizar.

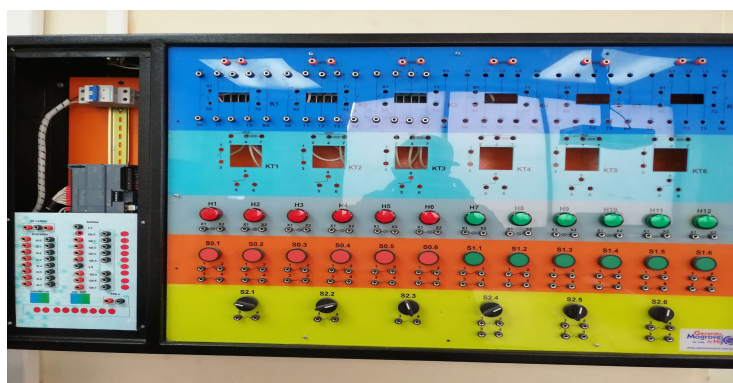


Figura 3.29: Tablero de control

Fuente: Autor

Para verificar el funcionamiento del PLC, se cargó un programa básico desde TIA PORTAL, con el cual se pudo comprobar el estado de cada entrada y salida del controlador. Además, se soldó las terminales dañadas del panel de contactos, expuesto en la Figura 3.30, que conectan las entradas/salidas del PLC con los actuadores.



Figura 3.30: Panel de contactos del tablero de control

Fuente: Autor

En cuanto al cableado, se comprobó continuidad línea por línea, y se reemplazó ciertos conductores considerados obsoletos. También, se ordenó el cableado mediante serpentinas de empaquetado, mejorando significativamente su estética. La Figura 3.31 muestra el resultado final del mantenimiento de esta sección del tablero.

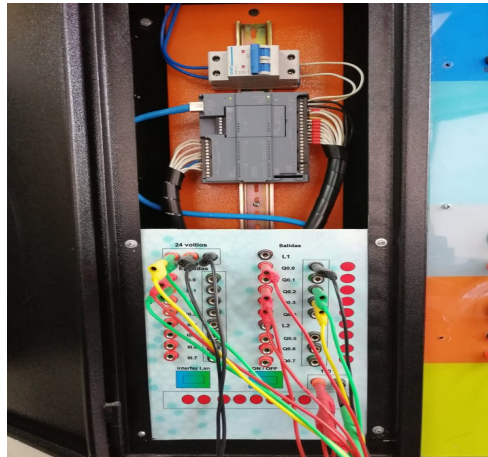


Figura 3.31: Panel de contactos del tablero de control

Fuente: Autor

En cuanto a los contactores, se abrió el tablero para acceder a ellos, como se puede observar en la Figura 3.32. Para comprobar su funcionamiento se realiza las siguientes acciones:

- Comprobar esquemas de conexión.
- Instalación correcta de conexiones defectuosas.
- Verificar la alimentación de la bobina.
- Limpieza de los de piezas internas del contactor.
- Comprobar el libre juego de todas las partes móviles.
- Limpieza de superficies polares.

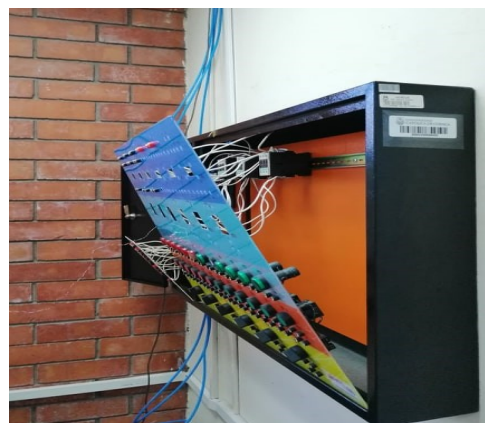


Figura 3.32: Panel desensamblado para acceder a contactores

Fuente: Autor

## Capítulo 4

# Pruebas de funcionamiento y análisis de resultados

### 4.1. Inicio de sesión en el editor de Node-RED y en la interfaz SCADA

Una vez levantado el servidor, los usuarios pueden iniciar sesión en el editor para configurar el sistema. La Figura 4.1 muestra los cuadros de las credenciales que deben ser completados para acceder al editor. Estos campos son:

- **Username:** Campo obligatorio. Los usuarios asignados a cada cliente deben registrarse aquí.
- **Password:** Aquí es donde debe registrarse la contraseña de cada cliente, y este campo no deberá estar vacío.
- **Botón Login:** Este botón intenta iniciar sesión. Si los datos registrados son válidos, ingresa al editor Node-RED. De lo contrario, se mostrará el mensaje **Login failed**.

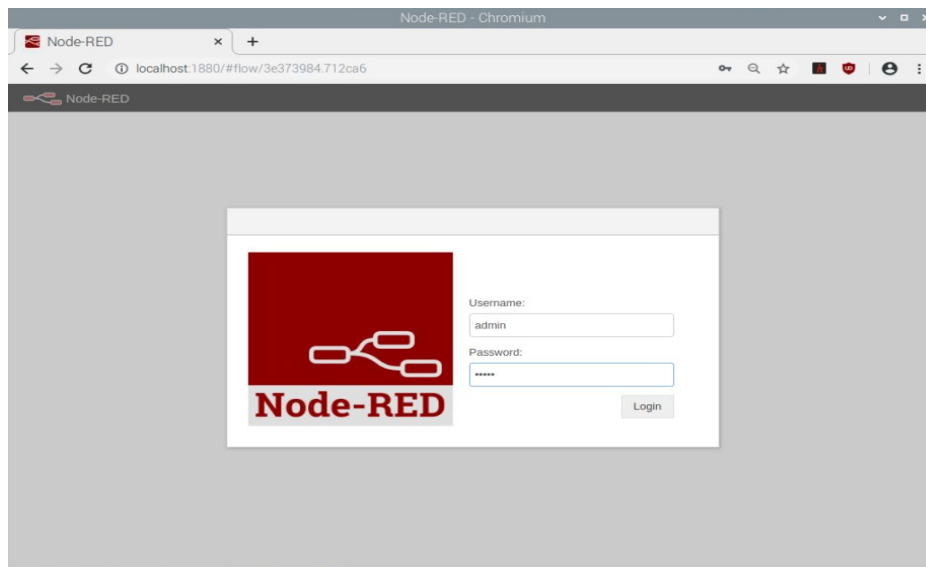
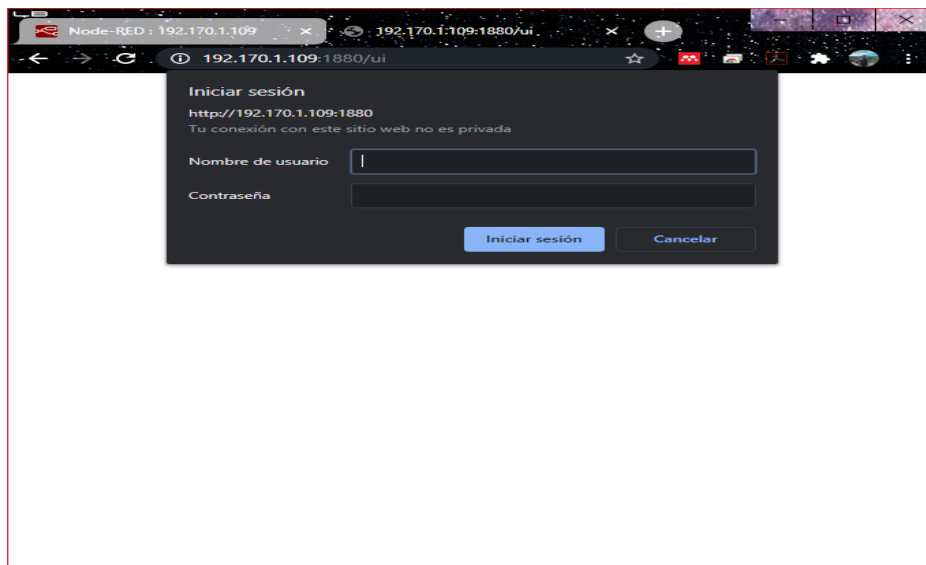


Figura 4.1: Ventana de inicio de sesión en el editor Node-RED

Fuente: Autor

Para ingresar a la interfaz del sistema, es necesario registrar las credenciales en los campos que se muestran en la Figura 4.2. Estos son los mismos campos expuestos en el editor de Node-RED.

Figura 4.2: Ventana de inicio de sesión *dashboard* SCADA

Fuente: Autor

## 4.2. Validación de usuarios

El perfil de usuario puede ser reconocido por el sistema. La Tabla 4.1 muestra un registro de los usuarios, sus contraseñas y permisos asignados a cada uno. Los usuarios **ismael** y **admin** cuentan con acceso completo al sistema. Mientras que el usuario **supervisor** cuenta un permiso de solo lectura, lo que significa que este usuario no tiene la capacidad de modificar el sistema desde el editor.

Tabla 4.1: Usuarios y permisos del sistema SCADA

Usuario	Contraseña	Permisos
admin	admin	Acceso completo
ismael	ismael123	Acceso completo
supervisor	supervisor	Solo lectura

Fuente: Autor

## 4.3. Dashboard

Después de iniciar sesión correctamente, el usuario encontrará la ventana principal del sistema como se muestra en la Figura 4.3.

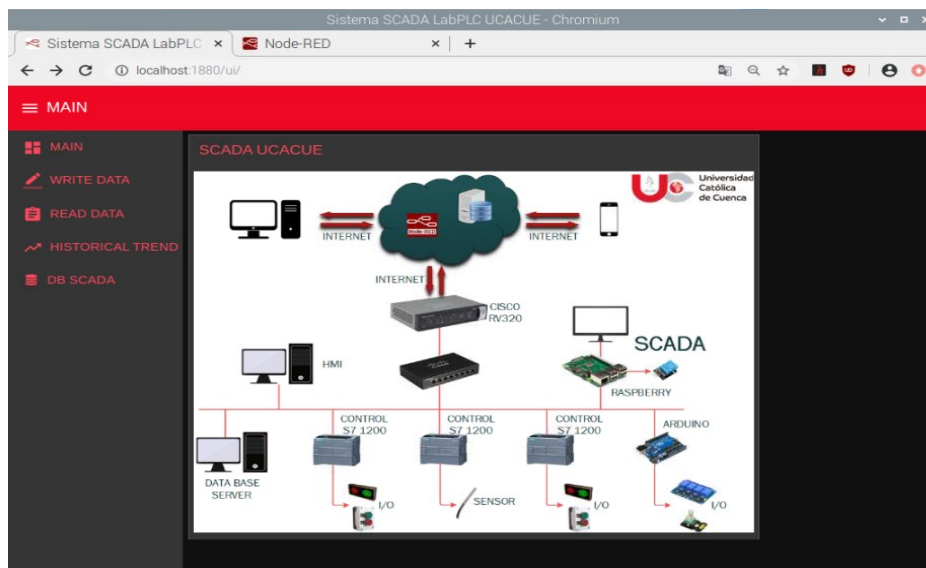


Figura 4.3: Ventana MAIN del sistema SCADA

Fuente: Autor

En la parte superior izquierda se encuentra el botón que despliega la barra de pestañas del sistema. El sistema cuenta con las siguientes ventanas:

### 4.3.1. Ventana MAIN

Esta es una ventana de presentación, mostrada en la Figura 4.3. A través de ella, es posible reconocer la arquitectura implementada, identificar cada elemento y reconocer la función que desempeña en el sistema.

### 4.3.2. Ventana WRITE DATA

Esta ventana está asignada para la escritura de datos sobre los equipos de control. La ventana consta de tres secciones mostradas en la Figura 4.4. Estas son:

- **ARDUINO OUTPUT:** Esta sección está situada en la zona izquierda de la ventana. Este apartado contiene botones para controlar los tres relés conectados a la salida Arduino.
- **RASPBERRY OUTPUT:** Sección emplazada en parte central de la ventana. Muestra los botones que intervienen sobre los LEDs conectados en las salidas de la Raspberry.
- **PLCs OUTPUT:** Se encuentra en el lado derecho de la ventana y muestra los botones que actúan sobre las salidas de los PLCs del sistema.

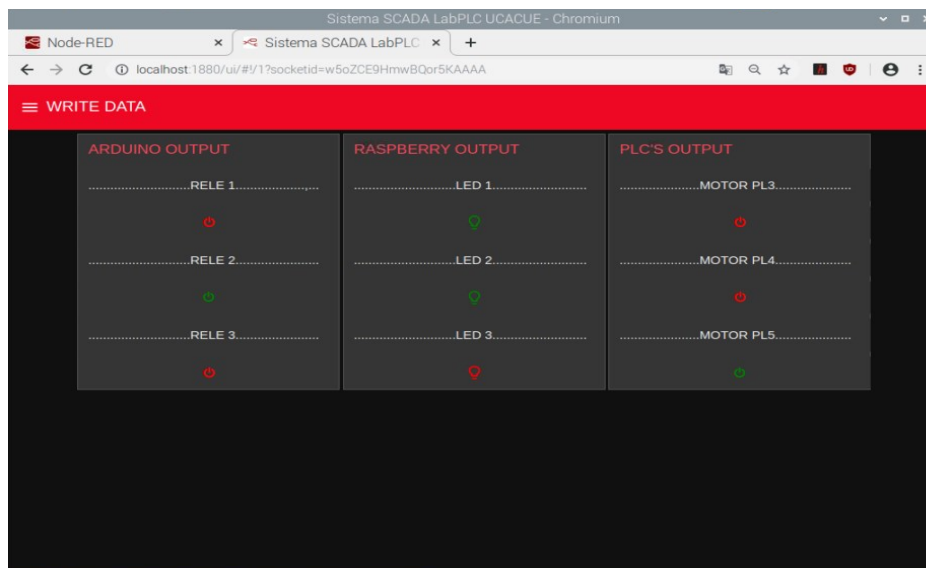


Figura 4.4: Ventana WRITE DATA del sistema SCADA

Fuente: Autor

### 4.3.3. Ventana READ DATA

Tiene como función mostrar las lecturas de los sensores de cada dispositivo controlador en tiempo real. Además, como se puede observar en la Figura 4.5, cuenta con un botón que permite almacenar en cualquier momento los datos registrados sin esperar el tiempo programado para guardar dichos datos. La ventana se divide en tres secciones.

- **RASPBERRY INPUT:** Se encuentra en la parte superior e imprime los datos de temperatura y humedad obtenidos por el sensor DTH11 conectado al pin de entrada de la Raspberry.
- **PLC3 INPUT:** Esta sección está en la parte inferior izquierda. Su función es visualizar los datos adquiridos por el sensor conectado a la entrada analógica PLC3.
- **ARDUINO INPUT:** Apartado de la sección inferior derecha. Se utiliza para publicar los datos obtenidos por el sensor conectado a la entrada analógica del Arduino.

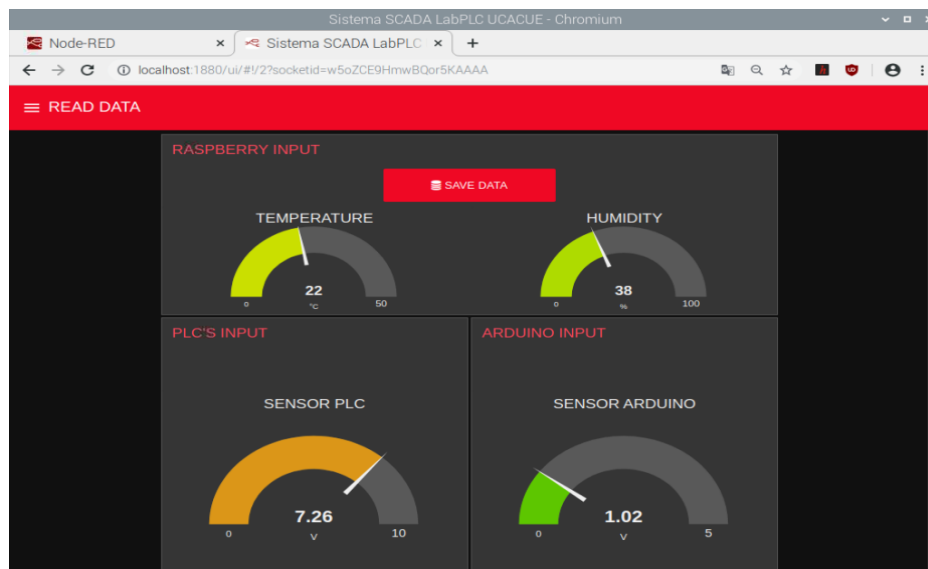


Figura 4.5: Ventana READ DATA del sistema SCADA

Fuente: Autor

#### 4.3.4. Ventana HISTORICAL TREND

La Figura 4.6 indica las secciones distribuidas en la ventana. Aquí encontramos las tendencias de los sensores de cada controlador. Cuenta con una herramienta dinámica para observar los datos; ubicando el cursor sobre la curva en la hora deseada mostrará el valor registrado en ese momento. Posee las siguientes secciones:

- **ARDUINO TREND:** Esta sección se encuentra en el lado izquierdo de la ventana. Muestra la tendencia de los sensores vinculados a la entrada analógica Arduino.
- **RASPBERRY TREND:** Se ubica en la zona central y muestra la tendencia de los datos de temperatura y humedad obtenidos por el sensor DTH11 conectado a la Raspberry Pi.
- **PLC TREND:** Enseña la tendencia del sensor asociado a la entrada analógica del PLC3 y se encuentra en la parte derecha de la ventana.

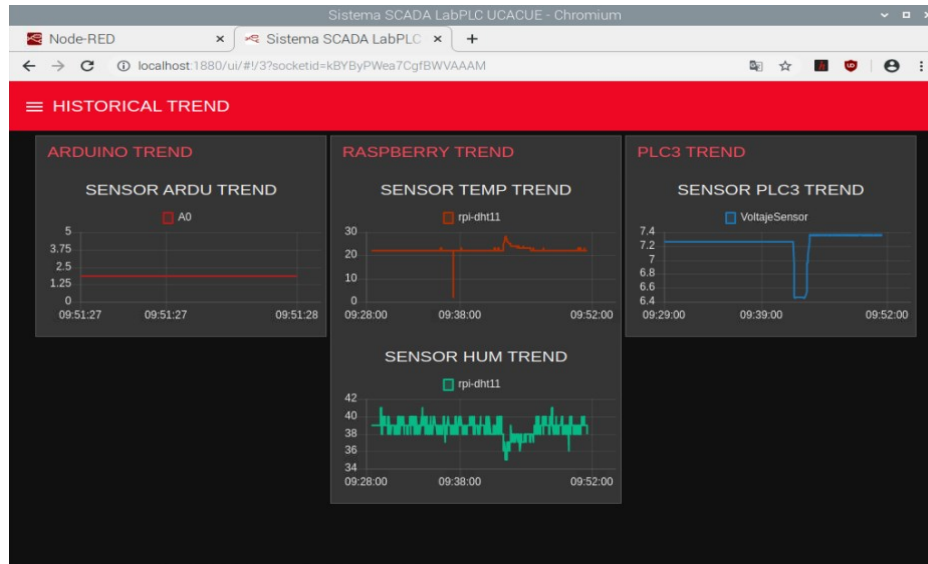


Figura 4.6: Ventana HISTORICAL TREND del sistema SCADA

Fuente: Autor

#### 4.3.5. Ventana DB SCADA

En esta ventana, encontramos dos secciones, como se muestra en la Figura 4.7. Cada sección tiene un botón para consultar los últimos datos almacenados en la base de datos del servidor MySQL. También cuentan con un botón que permite vaciar la base de datos cuando el usuario lo necesite.

Las secciones creadas en esta ventana son:

- **DB PLC3:** Situada en la zona izquierda de la ventana. A través del botón **CONSULTAR**, puede mostrar el último dato guardado en la base de datos del PLC3. También puede vaciar esta base de datos por medio del botón **VACIAR DB**.
- **DB RASPBERRY:** Sección ubicada a la derecha, permite consultar y vaciar la base de datos de la Raspberry.

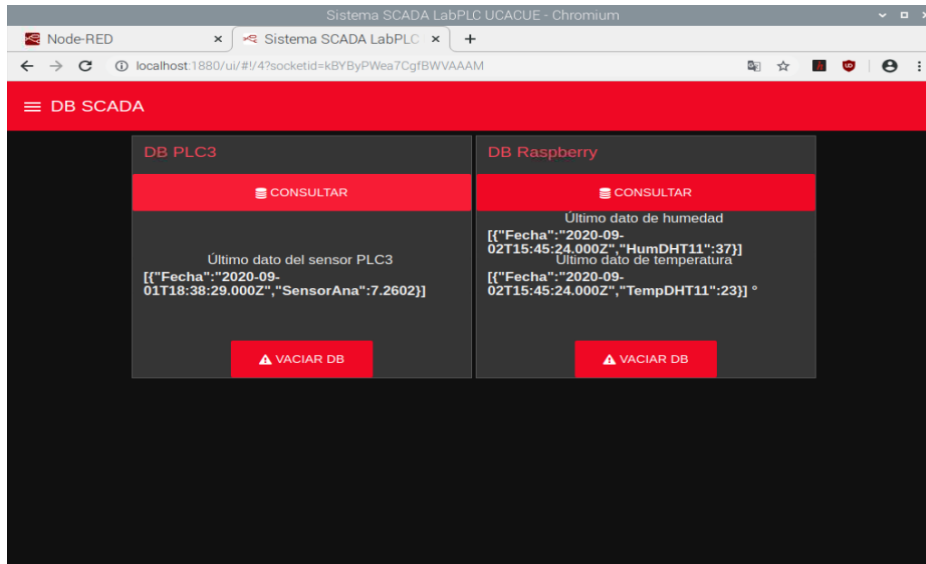


Figura 4.7: Ventana DB SCADA del sistema SCADA

Fuente: Autor

#### 4.4. Registro de datos en DB

La base de datos está diseñada para cumplir los requerimientos básicos y el propósito del sistema. Está constituida para que los usuarios puedan tener acceso a información fundamental; ignorando datos redundantes y manteniendo precisión de los mismos. De esta forma la DB ahorra recursos de *hardware* y *software*.

La base de datos está ubicada en un host externo. Esto con el motivo de no sobrecargar la Raspberry, donde está funcionando el sistema SCADA.

En la programación en el nodo MySQL en Node-RED, para la base de datos de temperatura y humedad del sensor Raspberry, el registro de datos se ejecuta cada minuto. Por otro lado, el tiempo de registro del sensor PLC3 es una vez cada treinta segundos. El Anexo [C](#) muestra una tabla más detallada de las tabas de la DB.

Las Figuras [4.8](#) y [4.9](#) muestran los historiadores de cien datos tomados del sensor DHT11 almacenados en la base de datos. El sensor fue puesto a prueba conjuntamente con un ventilador, este generó un cambio en la entrada de datos doble la humedad y temperatura en el sistema.

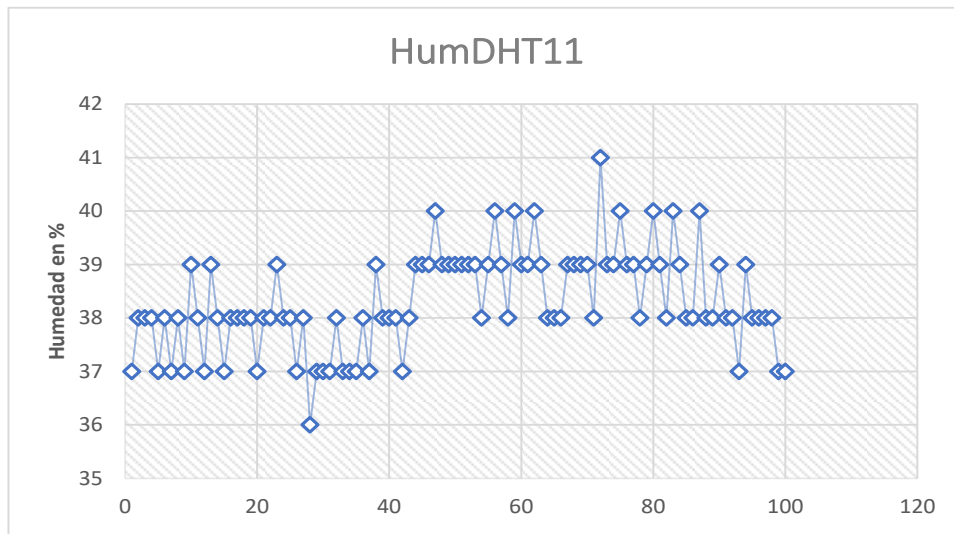


Figura 4.8: Historiador de los datos de humedad de la DB  
Fuente: Autor

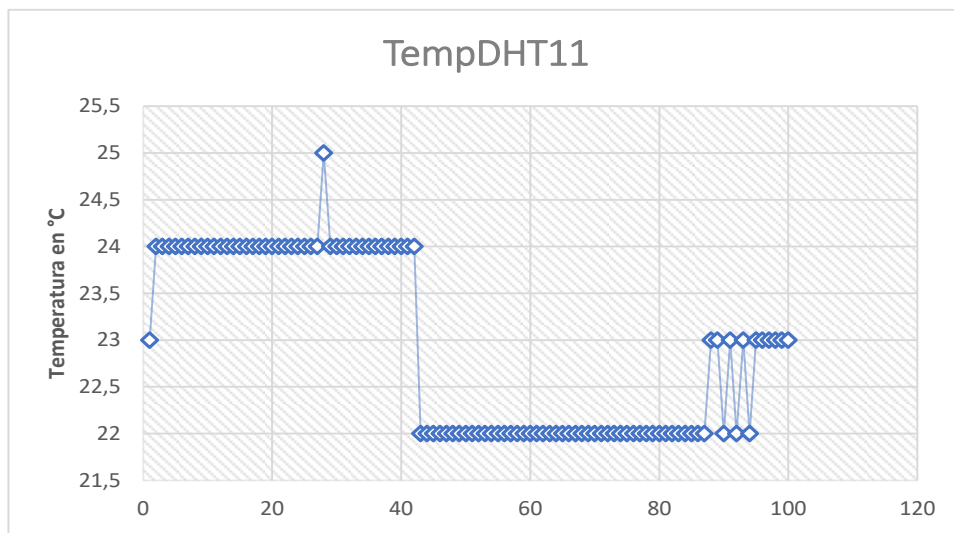


Figura 4.9: Historiador de los datos de temperatura almacenados en la DB  
Fuente: Autor

También la Figura 4.10 muestra el historiador de cien datos de voltaje captados en la entrada analógica del PLC3 y que se encuentran almacenados en la base de datos del sistema.

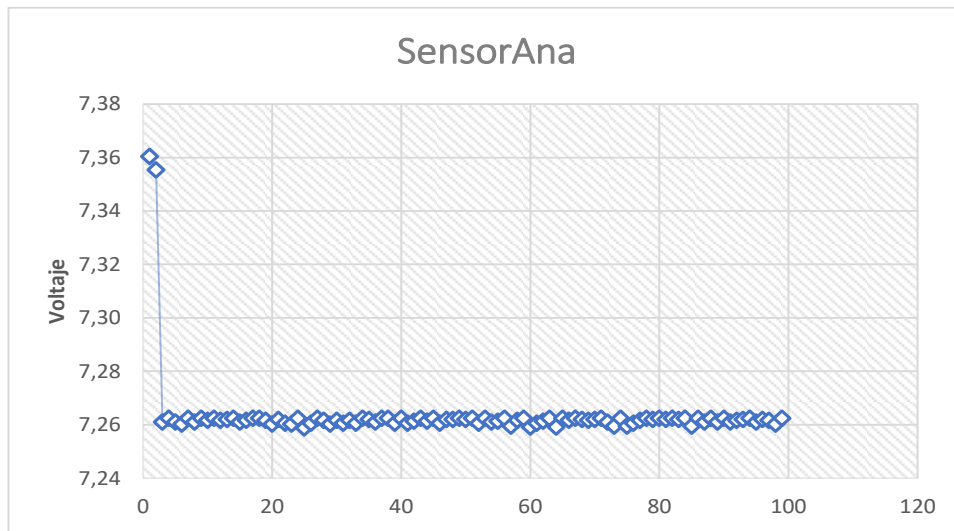


Figura 4.10: Historiador de los datos de voltaje de la entrada analógica del PLC3 almacenados en la DB  
Fuente: Autor

## 4.5. Pruebas de control de salidas de los controladores

Las pruebas fueron realizadas desde equipos conectados a la misma red del Laboratorio de Ingeniería Electrónica. Se inició con pruebas básicas para comprobar los requerimientos del sistema.

### 4.5.1. Prueba de control de salidas de los PLCs

Los PLCs son controlados mediante pulsos emitidos por los botones programados en Node-RED. En la interfaz del sistema SCADA, encontramos los botones que permiten que cada salida del PLC cambie de estado.

Se encuentra asignada una salida para cada PLC, como indica la Figura 4.11. Esta encenderá o apagará un foco, que puede ser reemplazado por un relé o contactor, cada vez que cambia de estado el botón. Además, es posible controlar esta salida manualmente desde el tablero de control y el sistema reconocerá el estado actual de la salida.



Figura 4.11: Control de salidas de PLCs desde Node-RED

Fuente: Autor

#### 4.5.2. Prueba de control de salidas del Arduino

Los pines digitales 9, 10 y 11 del Arduino se encuentran conectados al módulo de relés. Estos relés están inicialmente desactivados y para cambiar su estado es necesario encender los botones asignados para esta función en la interfaz de usuario de Node-RED.

En la Figura [4.12](#), podemos ver el funcionamiento de los relés. La luz indicadora aquí está apagada; esto significa que los contactos COMÚN y NO (normalmente abierto) están unidos, por lo que el relé está activado.

La velocidad de transmisión de datos que procesa **Firmata** es muy eficiente. Esto se refleja en el momento en que cambia el estado del relé, lo que resulta ser de inmediato.

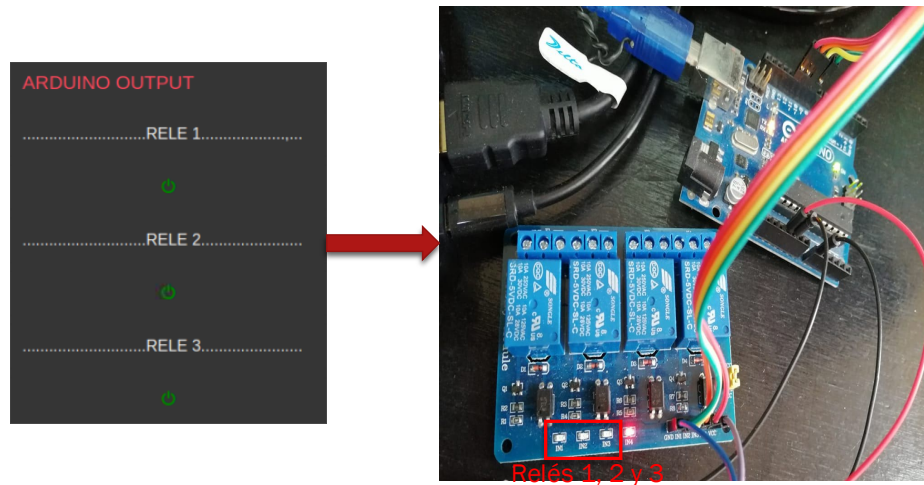


Figura 4.12: Control de salidas del Arduino desde Node-RED

Fuente: Autor

### 4.5.3. Prueba de control de salidas de la Raspberry

En esta prueba, los tres LED conectados a los pines 11, 13 y 15 de la Raspberry Pi, se encienden y apagan. Como se muestra en la Figura [4.13](#) en la sección **RASPBERRY OUTPUT**, se encuentran los botones que actúan sobre cada LED.

Debido a que el sistema SCADA está configurado en este controlador, la velocidad de transmisión de datos es muy alta y no hay interferencia.

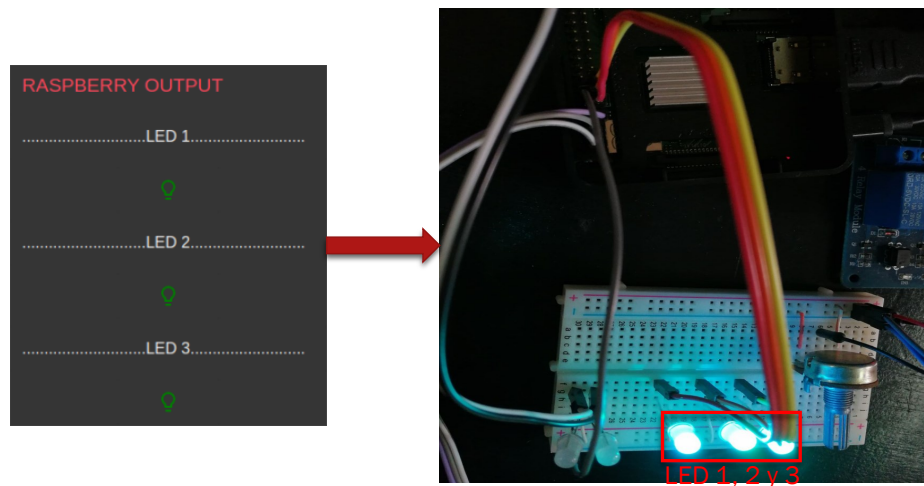


Figura 4.13: Control de salidas de la Raspberry desde Node-RED

Fuente: Autor

## 4.6. Eventos y alarmas

Para verificar la comunicación entre los controladores, se establecieron dos eventos en la Raspberry PI. Estos eventos están relacionados con los datos obtenidos de los sensores del PLC3 y Arduino.

El primer evento se ejecutará siempre y cuando el valor del sensor analógico del Arduino supere los 4 V. Cuando se excede este límite, el LED5 de la Raspberry PI se encenderá, como se muestra en la Figura 4.14. Una vez el valor censado sea inferior a 4 V, el LED5 se apagará.

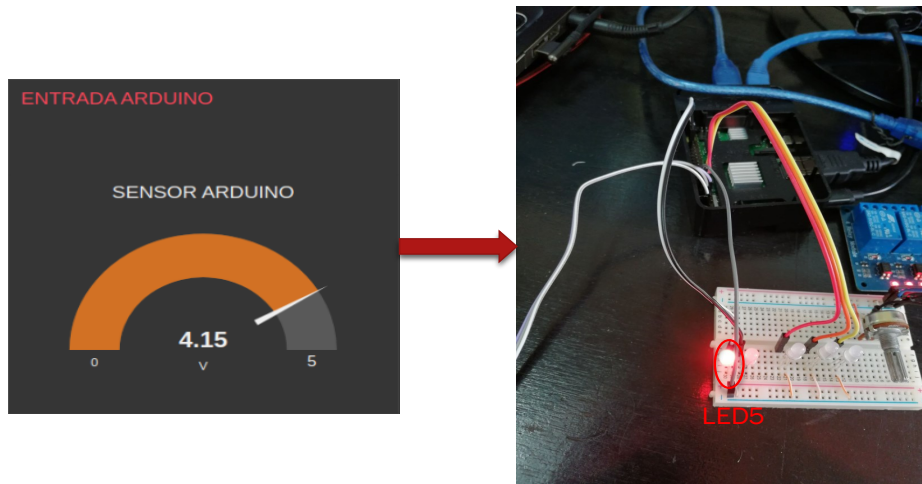


Figura 4.14: Evento entre Arduino y LED de Raspberry

Fuente: Autor

Mientras se enciende el LED5 de alarma, se mostrará un mensaje de alerta en la parte inferior derecha para informar al usuario que el voltaje, censado por el Arduino, es muy alto, como indica la Figura 4.15

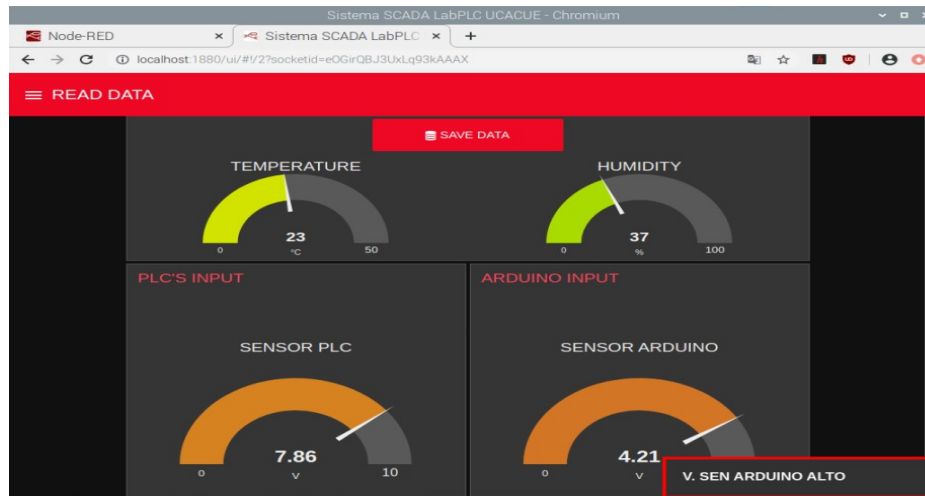


Figura 4.15: Mensaje de alarma de voltaje muy alto censado en el Arduino  
Fuente: Autor

La Tabla 4.2 indica el estado del LED5 y del mensaje de alerta para cada valor registrado en la prueba de eventos entre el sensor del Arduino y la Raspberry Pi.

Tabla 4.2: Pruebas de eventos sobre el sensor de la placa Arduino

Voltaje sensor Arduino	Estado del LED5	Mensaje de alarma
3,65	off	desactivado
3,76	off	desactivado
3,94	off	desactivado
4,1	on	activado
4,35	on	activado
4,56	on	activado
4,4	on	activado
4,25	on	activado
4,07	on	activado
3,98	off	desactivado
3,8	off	desactivado
3,62	off	desactivado

Fuente: Autor

En la Figura 4.16, puede observar los datos históricos registrados por el sensor del Arduino y el rango de voltaje en donde actúa el LED5.

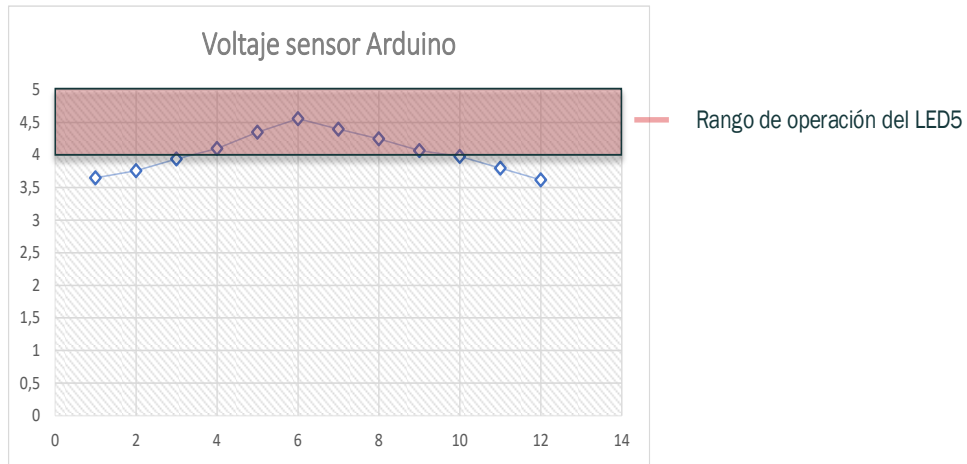


Figura 4.16: Historiador sobre el evento entre Arduino y LED5 de Raspberry  
 Fuente: Autor

La Figura 4.17 muestra otro evento en donde participa el PLC3 con su sensor y la Raspberry Pi con su LED4 como actuador. Aquí, el LED4 permanecerá apagado hasta que el valor de voltaje cesando por el PLC3 sobrepase los 8 V. Cuando esto sucede, el estado del LED cambiará y se iluminará. Además, se mostrará un mensaje de alarma, indicando que el valor de voltaje es muy alto, como indica la Figura 4.18.

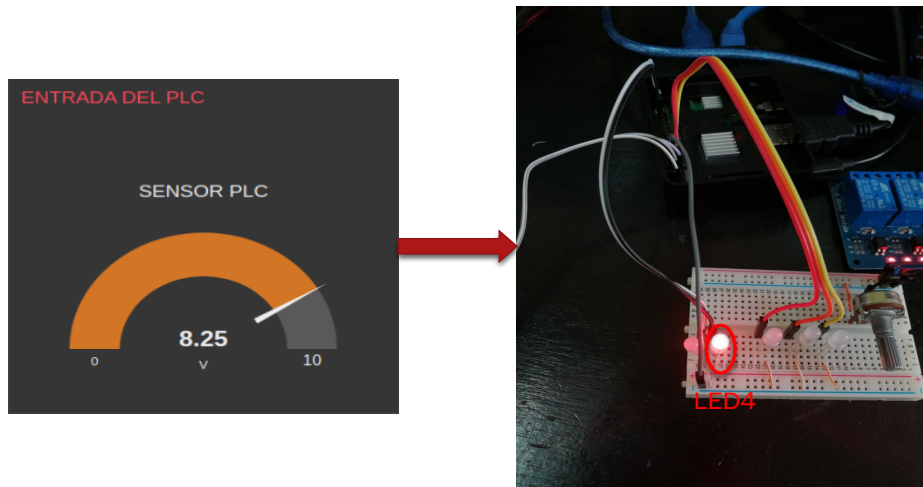


Figura 4.17: Evento entre PLC3 y LED de Raspberry  
 Fuente: Autor

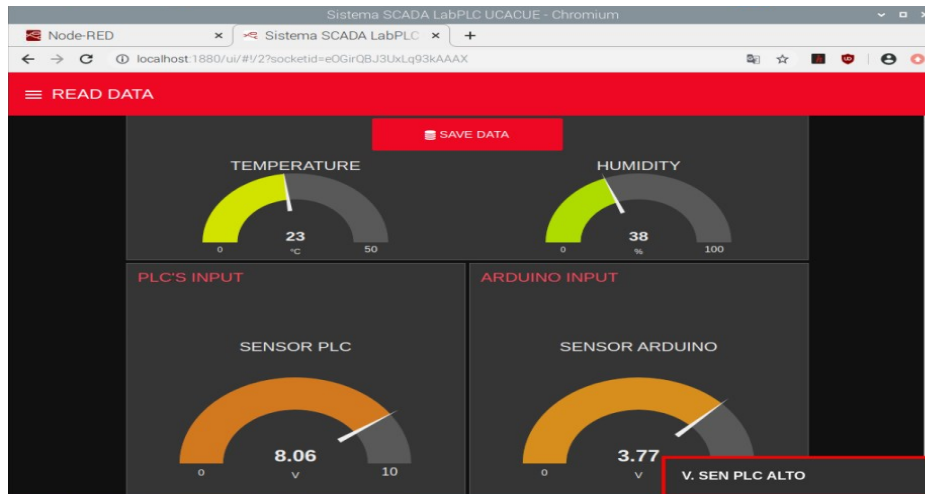


Figura 4.18: Mensaje de alarma de voltaje muy alto censado en el PLC  
Fuente: Autor

En la Tabla 4.3 se enumera el estado de LED5 y el mensaje de alarma para cada valor registrado en la prueba de eventos entre el sensor del PLC3 y Raspberry Pi.

Tabla 4.3: Pruebas de eventos sobre el sensor del PLC3

Voltaje sensor PLC3	Estado del LED6	Mensaje de alarma
7,58	off	desactivado
7,76	off	desactivado
7,94	off	desactivado
8,22	on	activado
8,41	on	activado
9,2	on	activado
8,97	on	activado
8,57	on	activado
8,09	on	activado
7,87	off	desactivado
7,68	off	desactivado
7,46	off	desactivado

Fuente: Autor

En la Figura 4.19, puede ver los datos históricos registrados por el sensor PLC3 y el rango de voltaje de operación del LED5.

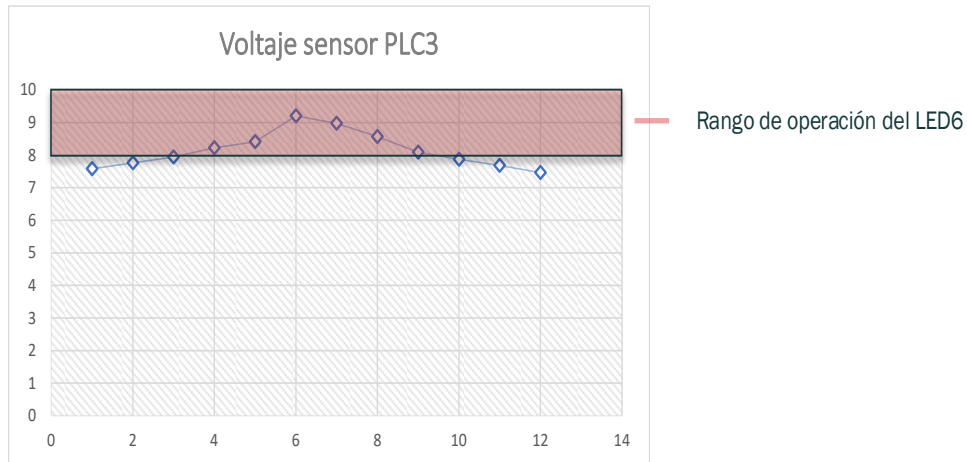


Figura 4.19: Historiador sobre el evento entre Arduino y LED5 de Raspberry

Fuente: Autor

Los mensajes de alerta se mostrarán en todas las ventanas del sistema. Esto quiere decir que sin importar en que sección estemos trabajando, los mensajes siempre serán visibles para el usuario.

## 4.7. Acceso remoto al sistema

Para acceder de forma remota desde otro dispositivo conectado a la misma red, la dirección IP de la Raspberry PI, donde se encuentra el sistema SCADA, debe colocarse utilizando el puerto asignado por Node-RED y la extensión /ui.

Como se puede observar en la Figura 4.20, se inicia sesión desde una computadora diferente por medio de la dirección <http://192.170.1.109:1880/ui>. Luego de ingresar correctamente las credenciales de usuario, tendremos acceso completo a la interfaz SCADA.

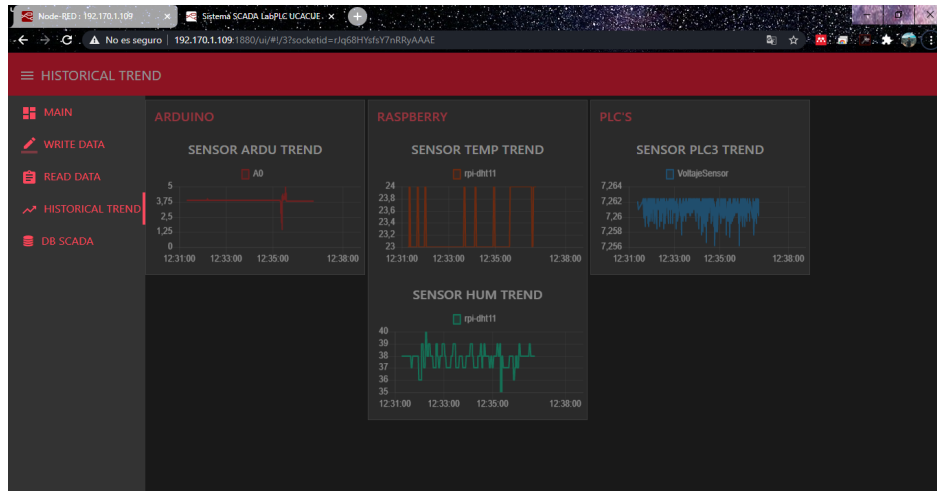


Figura 4.20: Acceso al sistema desde una computadora

Fuente: Autor

Al acceder desde un teléfono inteligente, se debe realizar el mismo proceso desde el navegador de su elección, como se muestra en la Figura [4.21](#).

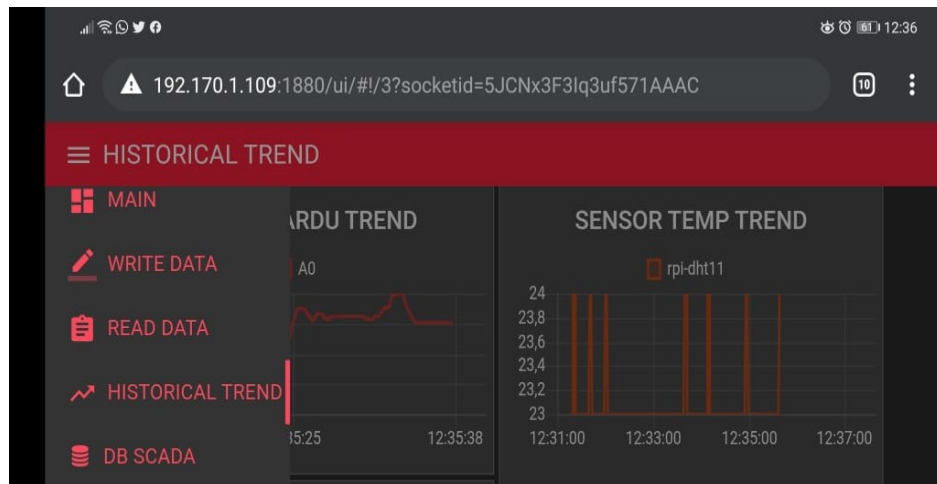


Figura 4.21: Acceso al sistema desde un teléfono inteligente

Fuente: Autor

# Capítulo 5

## Conclusiones

### 5.1. Conclusiones

Como se muestra en este proyecto, en la automatización industrial, las herramientas informáticas utilizadas para implementar los sistemas SCADA son muy importantes. Varios programas SCADA ocupan diferentes tipos de licencias, desde la versión de prueba hasta las versiones completas que requieren precios elevados. Por tanto, esto se considera uno de los puntos más importantes a la hora de seleccionar herramientas informáticas para el proyecto.

El sistema diseñado en base al modelo CIM, puede realizar la automatización del proceso de producción industrial. Las capas que componen esta estructura piramidal facilitan la identificación de las diferentes etapas involucradas en el proceso de automatización.

Se diseñó e implementó un sistema SCADA que trabaja sobre los tres niveles básicos del modelo CIM: nivel de proceso, nivel de campo y nivel de célula. En el nivel de proceso, como sensores pilotos se utiliza el sensor DHT11 y otros analógicos, y relés, focos y LED como actuadores. El nivel de control maneja cinco controladores lógicos programables PLC s7-1200, una placa Arduino y una Raspberry PI; siendo todos estos dispositivos heterogéneos, el sistema logró comunicar los equipos a pesar de que cada uno maneja un protocolo de comunicación diferente. En el nivel de célula, correspondiente al sistema SCADA, se implementan los servicios de Node-RED; desde este nivel se supervisa y controla íntegramente el sistema.

La particularidad de Node-RED de ser un software libre, permite reducir significativamente el costo total del proyecto. Además, cumple con los requisitos de comunicación, control y visualización requeridos para este trabajo.

El sistema SCADA está en la capacidad de generar alarmas y eventos, y crear datos históricos sobre el funcionamiento de los elementos. Mantiene la comunicación entre todos los equipos y puede controlar y supervisar el proceso a través de su amigable interfaz HMI (interfaz hombre-máquina), mostrando su desempeño. Además, genera procesamiento de datos, su posible gestión y generación de reportes mediante la base de datos implementada. El acceso remoto al sistema tiene grandes ventajas y puede proporcionar una gran cantidad de pantallas en el sistema a un costo muy reducido.

El hecho de que el sistema SCADA esté desarrollado en Node-RED ayuda a implementarlo en cualquier plataforma. El uso de código libre es la mayor característica del sistema porque no está vinculado a una licencia, por lo que puede crecer indefinidamente debido a las contribuciones de los colaboradores de esta herramienta.

## **5.2. Trabajo futuro**

- Vincular nuevos dispositivos de control al sistema.
- Probar nuevos métodos para mejorar la seguridad del sistema y su gestión de usuarios.
- Probar la operatividad del sistema en el ámbito industrial.
- Comprobar el funcionamiento del sistema con otros protocolos de comunicación y nuevos medios de transmisión (como la fibra óptica).

# Bibliografía

- Albán Guerrero, J. E. and Paguay Llamuca, N. J. (2017). Diseño e implementación de un sistema SCADA con comunicación PROFIBUS para el control y monitoreo de procesos industriales en el laboratorio de automatización de la Facultad de Mecánica.
- Alberto, L., Santillán, C., Gibert, M., Óscar, G., and Mora, P. (2015). Bases de datos en MySQL. Technical report, UOC.
- Almas, M. S., Vanfretti, L., Lovlund, S., and Gjerde, J. O. (2014). Open source SCADA implementation and PMU integration for power system monitoring and control applications. In *2014 IEEE PES General Meeting | Conference & Exposition*, pages 1–5. IEEE.
- Alvarez Echeverría, F. A. and Antonio, F. (2015). Implementación de nuevas tecnologías : valuación, variables, riesgos y escenarios tecnológicos.
- Arduino (2019). Arduino Uno Rev3.
- Areiza, Y. A., Charry, O., and Garzón Castro, Y. F. (2015). RASPBERRY PI. Revisión técnica: guía de uso y programación.
- Aula21 (2015). Qué es el protocolo Ethernet Industrial y cómo funciona | Aula21.
- Castro, C. and Lozano, C. (2017). Introducción a SCADA. Technical report, Universidad de Córdoba, Cordoba.
- Castro Gil, M.-A., Díaz Orueta, G., and Mur Pérez, F. (2017). *Comunicaciones industriales : sistemas distribuidos y aplicaciones*. UNED - Universidad Nacional de Educación a Distancia.
- Cherrez, Henry; Galabay, R. (2018). DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA SCADA PARA EL CONTROL DE CARGA Y EL MONITOREO DE LA ENERGÍA ELÉCTRICA OBTENIDA POR UN GENERADOR HÍBRIDO (EÓLICOSOLAR). Technical report, ESPE, Latacunga.
- Espinoza Castro, M. L. and Baño Soto, J. P. (2018). Integración de redes industriales mediante un software de monitoreo y adquisición de datos en el laboratorio de control y manipulación automática de la carrera de Ingeniería de Mantenimiento.
- Faedo, N. E., Lucero, M., Mazzone, V., Suarez, M., and Rojas, N. L. (2015). Low cost SCADA for a laboratory-scale bioreactor. In *2015 XVI Workshop on Information Processing and Control (RPIC)*, pages 1–6. IEEE.
- Gualan Sarango, M. A. and Pintado, M. A. (2019). Elaboración de un módulo de guías de práctica para aprendizaje y el entrenamiento en el desarrollo de sistemas SCADA. Technical report, Universidad Politécnica Saieciana, Cuenca.

- José M. Hurtado (2017). Introducción a las Redes de Comunicación Industrial. Technical report, I.E.S Himilce, Linares.
- Marín, A. (2018). Control de PLCs Siemens S7-1200 mediante el protocolo MODBUS a través del programa LABVIEW para realización de prácticas de comunicación industrial. Technical report, Universidad Pública de Navarra, Pamplona.
- Merchán Piedra, D. F. and Peralta Sarmiento, J. A. (2017). Diseño e implementación de un control tolerante a fallos activo utilizando plataformas de bajo costo para control, comunicación y supervisión de un proceso mediante programas de código libre y controladores lógicos programables.
- Minchala, L. I., Ochoa, S., Velecela, E., Astudillo, D. F., and Gonzalez, J. (2016). An open source SCADA system to implement advanced computer integrated manufacturing. *IEEE Latin America Transactions*, 14(12):4657–4662.
- Moreano Abata, A. R. (2019). DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA CARRERA DE INGENIERÍA ELECTROMECAÁNICA TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO ELECTROMECAÁNICO. Technical report, Universidad de las Fuerzas Armadas, Riobamba.
- Mundologia (2015). El Mundo De La Automatización Industrial: Comunicaciones Industriales.
- Node-RED (2019). Node-RED.
- NodeJS (2019). Acerca | Node.js.
- Oracle (2020). MySQL Database Service | Oracle.
- Padilla Julca, C. E. and Toro Ydrogo, G. E. (2018). Diseño e implementación de un sistema SCADA con PLC siemens S7-200 para el control PID de la planta de nivel de la escuela profesional de ingeniería electrónica de la Universidad Nacional Pedro Ruiz Gallo, utilizando software libre. *Universidad Nacional Pedro Ruiz Gallo*.
- Penin, A. (2011). *Sistemas SCADA - Marcombo, S.A. (ediciones técnicas)*. 3 edition.
- Prokhorov, A. S., Chudinov, M. A., and Bondarev, S. E. (2018). Control systems software implementation using open source SCADA-system OpenSCADA. In *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pages 220–222. IEEE.
- Proserquisa (2016). Curso de Arduino. Technical report, San Salvador.
- RaspberryPi (2018). Raspberry Pi 3 Model B+. Technical report.
- Romagosa, J., Gallegos, D., and Pacheco, R. (2004). *MINIPROYECTO AUTOMATIZACIÓN INDUSTRIAL (AUTI)*. PhD thesis, Universitat Politècnica de Catalunya, Barcelona.
- Roque, E. and Antonio, J. (2018). Protocolos de comunicaciones industriales Logicbus S.A de C.V. Technical report, Logicbus SA de CV.
- Rosado Muñoz, A. (2010). SISTEMAS INDUSTRIALES DISTRIBUIDOS: Una filosofía de automatización APUNTES TEORÍA 3º ITT-SE. Universidad de Valencia. Dpto. Ingeniería Electrónica. Technical report, Universidad de Valencia.

Salazar Serna, C. A. and Correa Ortiz, L. C. (2013). Buses de campo y protocolos en redes industriales. *Ventana Informatica*, (25).

Sarmiento Sánchez, D. J. (2016). Implementación de los protocolos de comunicación industrial mediante Simatic S7-1200 para el laboratorio de automatización de la Universidad del Azuay.

Siemens (2018). Engineered with TIA Portal. Technical report, SIEMENS.

# Apéndice A

## Práctica No. 1

### Lectura y escritura de datos del sistema SCADA en Node-RED

#### A.1. Objetivos

- Conocer el funcionamiento de los nodos de entrada y salida del sistema Node-RED.
- Crear una interfaz amigable para el usuario.
- Controlar las entradas y salidas desde un equipo remoto.

#### A.2. Descripción

Esta práctica tiene como objetivo crear un sistema que pueda integrar tres controladores diferentes y administrar sus entradas y salidas. Además, propone la implementación de la interfaz desde donde se gestiona el proceso. Para realizar esta práctica, debe tener conocimientos de programación en el TIA Portal.

#### A.3. Equipos y materiales

- Un PLC S7-1200.
- Un controlador Arduino.
- Un controlador Raspberry Pi.
- Sensor DHT11.
- Un módulo relé Arduino.
- Un juego de LEDs y resistencias.
- Potenciómetro.
- Una fuente variable DC.

- Cables de conexión.
- Software TIA Portal.

## A.4. Desarrollo de la práctica

### CONTROL DE ENTRADAS Y SALIDAS DE LOS DISPOSITIVOS CONTROLADORES

#### Instalación de Node-RED

Existen instrucciones que nos permiten instalar esta herramienta de programación visual desde la terminal de Raspberry PI.

Las instrucciones a ejecutarse son:

- `sudo apt install nodejs`
- `sudo npm install -g --unsafe-perm node-red`

Para levantar el servicio de Node-RED, debe escribir en la terminal el siguiente código:

- `node-red start`

#### Instalación y configuración de nodos para los controladores

Node-RED cuenta con nodos específicos para cada dispositivo controlador. Para hacer uso de ellos, es necesario instalar los siguientes paquetes desde la sección **Manage palette** en el editor de Node-RED:

- `node-red-contrib-s7`
- `node-red-node-arduino`
- `node-red-node-gpio`
- `node-red-contrib-dht-sensor`
- `node-red-dashboard`

Después de realizar la instalación, los nodos aparecerán en la sección izquierda del editor.

Utilice un nodo S7 de entrada y uno de salida para el control de las entradas y salidas del PLC. De igual forma, emplee un nodo rpi gpio de salida y uno dht para el sensor de la Raspberry PI. Por último, utilice un nodo arduino out y un nodo arduino in para intervenir sobre la placa Arduino. Después de colocar los nodos en el editor, continúe configurando los parámetros requeridos por cada uno.

En el caso de los nodos S7, la Figura [A.2](#) muestra un ejemplo de los parámetros que deben ser llenados para levantar la comunicación con el PLC.

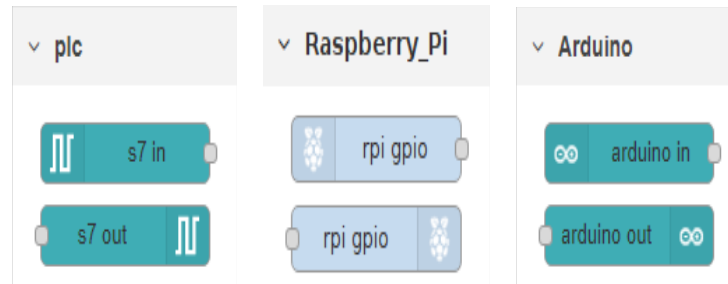


Figura A.1: Nodos designados para la práctica

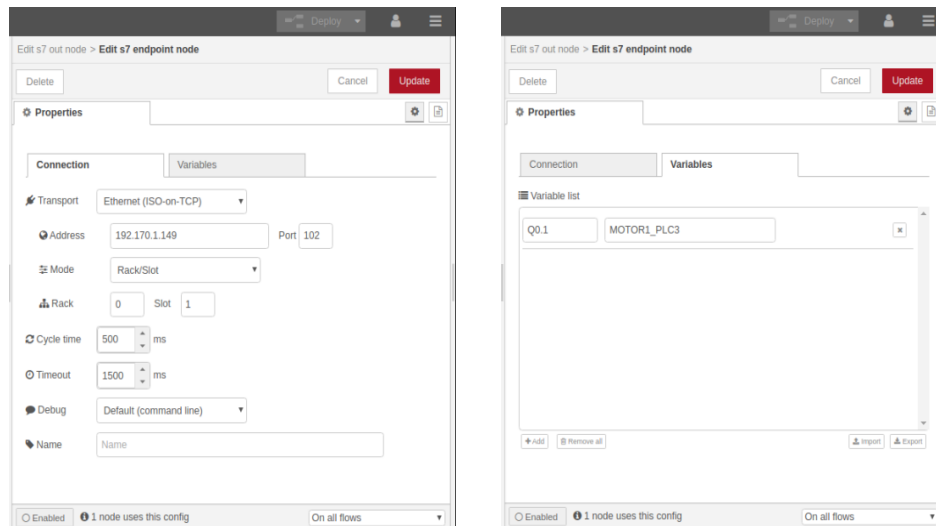


Figura A.2: Ejemplo de parametrización del nodo S7

Para los nodos arduino in, siga el ejemplo presentado en la Figura [A.3](#) sobre los campos que tienen que ser llenados para controlar las salidas.

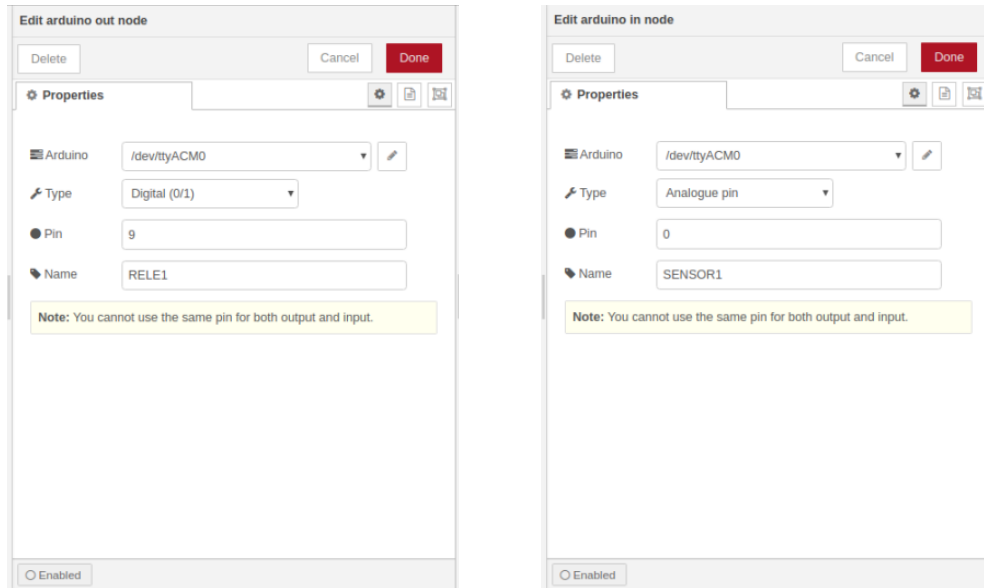


Figura A.3: Ejemplo de parametrización del nodo arduino in

Utilice, de ser necesario, la parametrización mostrada en la Figura A.4 como base para configurar los nodos gpio output y dht22.

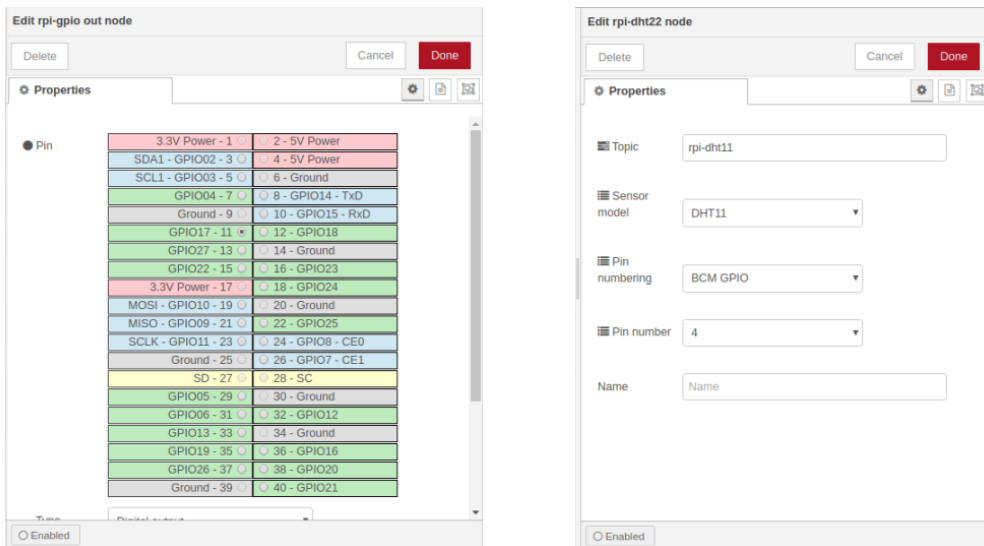


Figura A.4: Ejemplo de parametrización de los nodos gpio output y dht22

Después de configurar el nodo, conecte la fuente variable DC a la entrada analógica del PLC. A través de TIA

PORTAL, realice la programación necesaria para escalar y normalizar esta tensión de entrada, y luego almacenar el valor normalizado del voltaje de en una variable perteneciente a una DB del programa, la cual estará vinculada a la variable programada en el nodo S7. También realice la programación en escalera para controlar una salida del PLC y enlace esta variable con la asignada en el nodo S7 out.

Luego conecte el potenciómetro a la entrada analógica establecida en el Arduino. En el IDE de Arduino, busque y cargue el sketch Firmata, como indica la Figura A.5 Para el control de las salidas, conecte el pin, definido en el nodo arduino out en la configuración de Nore-RED, con el relé.

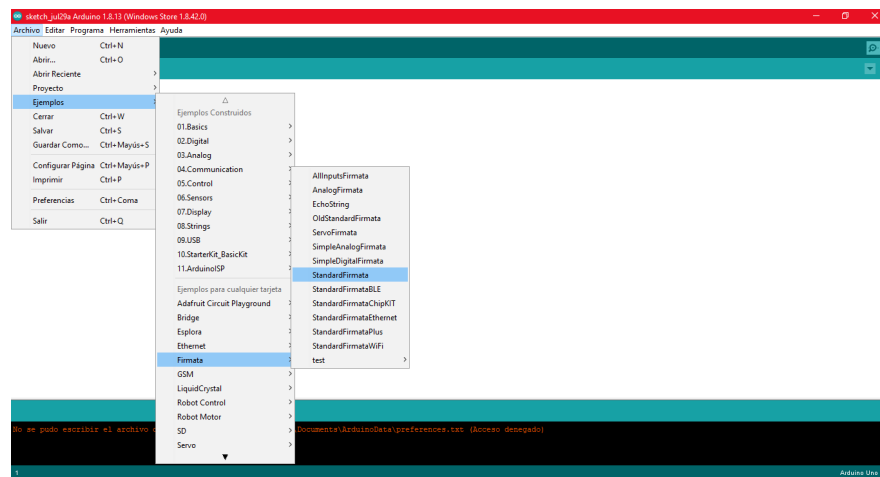


Figura A.5: Ubicación de sketch de FIRMATA en el IDE de Arduino

Por último, enlace el sensor DHT11 y el LED con el GPIO asignado en la configuración de los nodos.

### Creación de interfaz gráfica

Para crear una interfaz, previamente debe instalar el paquete Dashboard indicado con antelación.

Una vez que el paquete se haya instalado correctamente, encontraremos los nodos dashboard en el lado izquierdo del editor de Node-RED, como se muestra en la Figura A.6.

Para controlar las salidas desde la interfaz, es necesario vincular un interruptor o botón al nodo de salida de cada controlador. Para que esto se cumpla, arrastre el nodo **switch**, asigne un nombre y conéctelo al nodo **rpi gpio** de salida, como indica el ejemplo de la Figura A.7. Repita este procedimiento para la salida del PLC y del Arduino.



Figura A.6: Nodos Dashboard en Node-RED

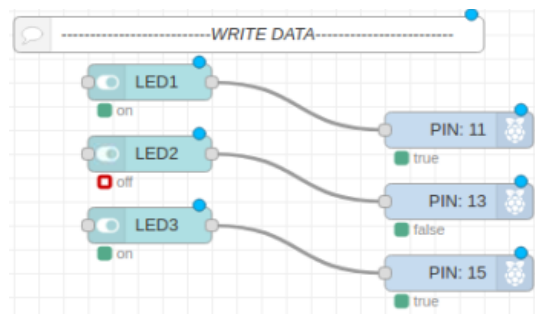


Figura A.7: Ejemplo conexión de nodos con switch del dashboard

En el caso de las entradas de los sensores, utilice los nodos **gauge** y **chart**. Conecte los nodos mencionados a las salidas de los nodos correspondientes a los sensores de los equipos. La Figura [A.8](#) muestra un ejemplo de conexión.

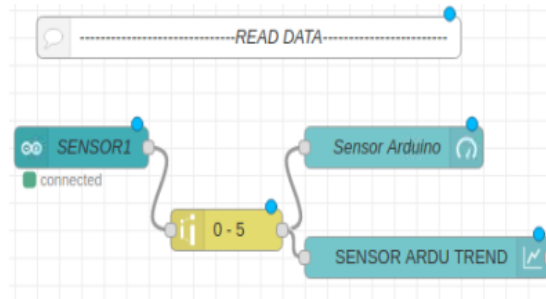


Figura A.8: Ejemplo conexión de nodos gauge y chart

Todos los nodos, pertenecientes al Dashboard, deben estar organizados en pestañas para ser mostrados correctamente en la interfaz.

#### Acceso remoto al sistema

Para acceder remotamente a la interfaz del sistema, en un nuevo dispositivo debe ingresar al navegador Web de su preferencia y en la barra de navegación colocar la dirección IP de la Raspberry seguido del puerto 1880 más la extensión /ui. Esto le enviará a la interfaz creada anteriormente.

Utilice los switches configurados para comprobar el funcionamiento de las salidas de los controladores. Navegue por las pestañas y supervise los valores obtenidos por los sensores y las tendencias que estos generan.

## A.5. Reporte

- Adjunte una imagen de la interfaz creada, demostrando el funcionamiento del sistema.
- Reporte sus conclusiones y comentarios.
- Proponga una o más aplicaciones de este sistema en la industria.

# Apéndice B

## Práctica No. 2

### Base de datos, usuarios, alarmas y eventos del sistema SCADA en Node-RED

#### B.1. Objetivos

- Generar una base de datos para la gestión de los valores obtenidos por los sensores.
- Brindar seguridad por medio de administración de usuarios.
- Establecer alarmas y eventos relacionando los controladores.

#### B.2. Descripción

Partiendo de la Práctica No1. Este trabajo completará las herramientas necesarias para asumir un sistema SCADA íntegro.

Esta práctica sugiere vincular el sistema con la base de datos configurada en MySQL. Esto permite gestionar los datos obtenidos por los sensores de cada controlador. Además, tiene como objetivo generar seguridad mediante la creación y gestión de usuarios con los respectivos permisos para realizar operaciones en el sistema. Como herramienta complementaria, se establecerán alarmas y eventos para vincular los controladores. Para esta experiencia necesita tener conocimiento de programación de base de datos.

#### B.3. Equipos y materiales

- Un PLC S7-1200.
- Un controlador Arduino.
- Un controlador Raspberry Pi.
- Sensor DHT11.

- Un módulo relé Arduino.
- Un juego de LEDs y resistencias.
- Potenciómetro.
- Una fuente variable DC.
- Cables de conexión.
- Software TIA Portal.
- Software MySQL
- Software Workbench

## **B.4. Desarrollo de la práctica**

### **CREACIÓN DE UNA BASE DE DATOS, ADMINISTRACIÓN DE USUARIOS, CONFIGURACIÓN DE ALARMAS Y EVENTOS**

Esta práctica inicia a partir del trabajo realizado en la Práctica No. 1. Manteniendo el desarrollo y sus resultados.

#### **Configuración de la base de datos en MySQL**

El servidor de MySQL debe ser configurado en un host diferente a la Raspberry Pi para tener una gestión remota.

Como primer paso, en MySQL debe crear una nueva base de datos con las tablas donde se almacenarán los registros. En este caso, habrá tres tablas, una por cada sensor correspondiente a cada controlador.

Las líneas de comando que se muestra a continuación, servirá como base para crear una tabla en el software MySQL Workbench.

```
CREATE TABLE 'nodedata'. 'dataplc1' (
  'id' INT NOT NULL AUTO_INCREMENT,
  'Fecha' DATETIME NOT NULL,
  'SensorAna' FLOAT NOT NULL,
  PRIMARY KEY ('id'));
```

```
CREATE TABLE 'nodedata'. 'datahumras' (
  'id' INT NOT NULL AUTO_INCREMENT,
  'Fecha' DATETIME NOT NULL,
  'HumDHT11' FLOAT NOT NULL,
  PRIMARY KEY ('id'));
```

```
CREATE TABLE 'nodedata'. 'datatempras' (
  'id' INT NOT NULL AUTO_INCREMENT,
```

```
'Fecha ' DATETIME NOT NULL,  
'TempDHT11 ' FLOAT NOT NULL,  
PRIMARY KEY ( 'id ' ));
```

Tenga en cuenta que el nombre de las tablas se utilizarán más adelante en los nodos de funciones en Node-RED. Por lo tanto, debe administrar las etiquetas correctamente.

Para poder escribir en la base de datos desde un host externo, el usuario debe tener privilegios para admitir la conexión. Para resolver este problema, se debe ejecutar el siguiente comando en la base de datos en Workbench.

```
GRANT ALL PRIVILEGES ON *.* TO 'root '@' %'  
WITH GRANT OPTION;  
FLUSH PRIVILEGES;
```

Donde “root” es el usuario y “%” es el parámetro para identificar el *host* donde se encuentra el servidor MySQL.

Para contar con el nodo MySQL, debe instalar desde la sección **Manage palette** el paquete *node-red-node-mysql*. Si la instalación es exitosa, el nodo mostrado en la Figura B.1 aparecerá en la sección izquierda del editor de Node-RED.



Figura B.1: Nodo MySQL en Node-RED

Arrastre el nodo hacia un nuevo flujo y realice las configuraciones necesarias para obtener conexión con el servidor MySQL. La Figura B.2 brinda una muestra de la correcta parametrización del nodo.

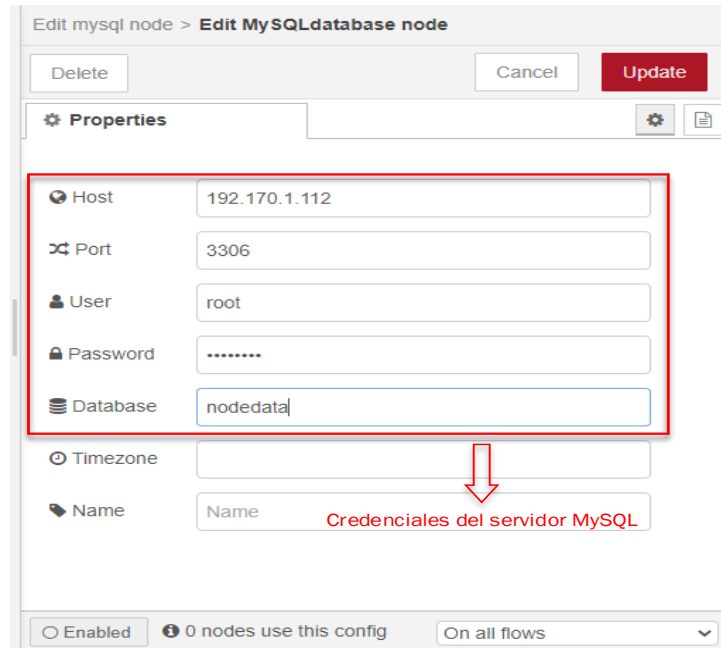


Figura B.2: Ventana de configuración del nodo mysql en Node-RED

La Figura B.3 muestra un ejemplo de programación para almacenar los datos obtenidos por el sensor PLC. Agregue un nodo de inyección al proceso y configure el intervalo de tiempo para introducir el mensaje que se utilizará para activar la función de almacenamiento. El nodo mencionado anteriormente se conectará al nodo S7, que se ha configurado de acuerdo con el procedimiento de la Práctica 1. Luego, adjuntamos un nodo de función en donde se programa una consulta de tipo INSERT INTO y debe estar conectada al nodo MySQL que se configuró anteriormente.

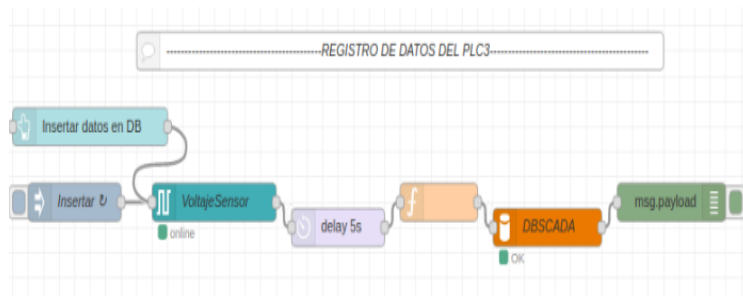


Figura B.3: Ejemplo de programación para realizar consultas en MySQL

Repita este procedimiento para los sensores del Arduino y de la Raspberry PI, considerando el nombre de la tabla correspondiente para cada caso.

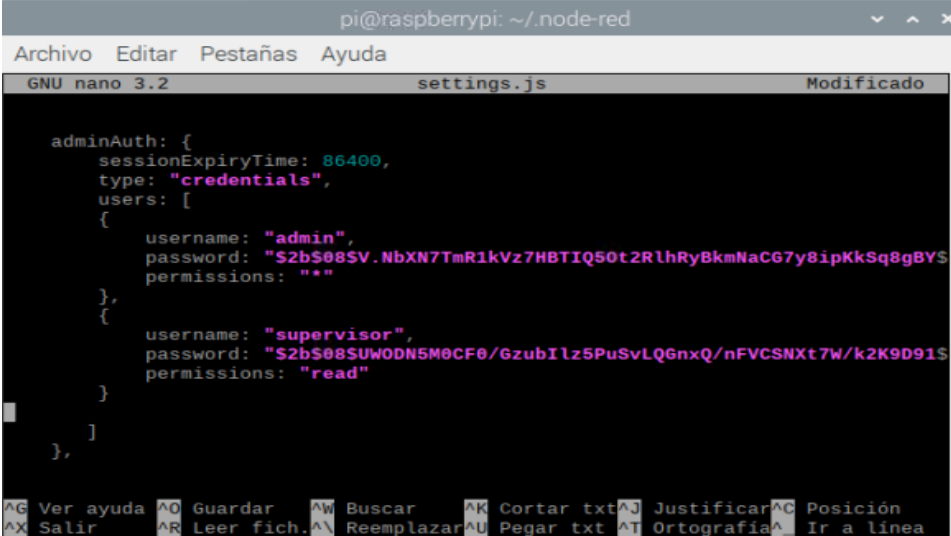
### Administración de usuarios

Las cuentas de usuario son una forma de identificar y autenticar a las personas con del sistema. Para esta práctica, se utilizará la autenticación basada en nombre de usuario/contraseña.

Para habilitar esta herramienta, debe ubicarse en el archivo de configuración de Node-RED **settings.js**, ejecutando las siguientes instrucciones en la línea de comando de la Raspberry PI.

```
pi@raspberrypi:~ $ cd .node-red
pi@raspberrypi:~/node-red $ nano settings.js
```

Esto le dirigirá al archivo de configuración donde debe descomentar la sección **adminAuth**. Agregue dos usuarios, como muestra el ejemplo de la Figura B.4, y administre el tipo de permiso para cada uno.



```
pi@raspberrypi: ~/node-red
Archivo Editar Pestañas Ayuda
GNU nano 3.2 settings.js Modificado

adminAuth: {
  sessionExpiryTime: 86400,
  type: "credentials",
  users: [
    {
      username: "admin",
      password: "$2b$08$V.NbXN7TmR1kVz7HBTIQ50t2RlhRyBkmNaCG7y8ipKkSq8gBYS",
      permissions: "*"
    },
    {
      username: "supervisor",
      password: "$2b$08$SUW0DN5M0CF0/GzubIlz5PuSvLQGnxQ/nFVCSNXt7W/k2K9D91S",
      permissions: "read"
    }
  ]
},
```

Figura B.4: Archivo de configuración de Node-RED

Para la protección de la interfaz del sistema, es necesario modificar el archivo de configuración, descomentando las secciones **httpNodeAuth:** y **httpStaticAuth:**. Estas propiedades permiten definir un nombre de usuario y contraseña únicos que podrán acceder a las rutas. Luego configure el usuario que administrará la interfaz, como se muestra el ejemplo de la Figura B.5, y se agrega un hash de contraseña.

```

pi@raspberrypi: ~/node-red
Archivo  Editar  Pestañas  Ayuda
GNU nano 3.2  settings.js  Modificado

// To password protect the node-defined HTTP endpoints (httpNodeRoot), or
// the static content (httpStatic), the following properties can be used.
// The pass field is a bcrypt hash of the password.
// See http://nodered.org/docs/security.html#generating-the-password-hash
httpNodeAuth: {user:"admin", pass:"$2b$08$V.NbXN7TmR1kVz7HBTIQ50t2RlhRyBkmNa$
httpStaticAuth: {user:"admin", pass:"$2b$08$V.NbXN7TmR1kVz7HBTIQ50t2RlhRyBkm$
// The following property can be used to enable HTTPS
// See http://nodejs.org/api/https.html#https\_https\_createserver\_options\_reS
// for details on its contents.
// See the comment at the top of this file on how to load the `fs` module us
// this setting.
//
// https: {
//   key: fs.readFileSync('privatekey.pem'),
//   cert: fs.readFileSync('certificate.pem')
// },
//
// The following property can be used to cause insecure HTTP connections to

AG Ver ayuda  AO Guardar  AW Buscar  AK Cortar txt  AJ Justificar  AC Posición
AX Salir     AR Leer fich.  AA Reemplazar  AU Pegar txt  AT Ortografía  AA Ir a línea

```

Figura B.5: Configuración de seguridad HTTP en el archivo de configuración de Node-RED

### Alarmas y eventos

Configure un evento donde esté relacionado el sensor del PLC con un LED actuador de la Raspberry PI. Para realizar este procedimiento, debe agregar al nodo de salida del plc un nodo **change**, que permitirá enviar un dato boleando, el cual activará el actuador de la Raspberry siempre y cuando se cumplan las condiciones establecidas. Para este caso, establezca la condición para que el LED se encienda cuando el sensor obtenga un valor igual o superior a 8 v.

También utilice el nodo **notification**, mostrado en la Figura B.6. Al conectarlo a la salida del nodo **change** se activará conjuntamente con el LED y mostrará mensajes emergentes mientras el LED de alarma esté funcionando. Ingrese al nodo y configure un mensaje acorde a la alarma indicada.

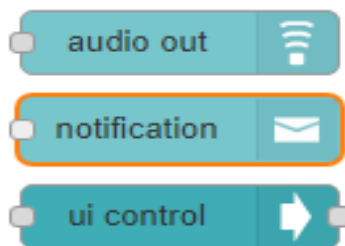


Figura B.6: Nodo notification en Node-RED

## **B.5. Reporte**

- Adjunte una imagen de las tablas de la base de datos en MySQL con los datos registrados por los sensores.
- Reporte sus conclusiones y comentarios.
- Reporte las imágenes de las alarmas y los eventos del sistema.

## Apéndice C

### Tablas de la base de datos nodedata

id	Fecha	HumDHT11
1	31/8/2020 12:52	37
2	31/8/2020 12:53	38
3	31/8/2020 12:54	38
4	31/8/2020 12:55	38
5	31/8/2020 12:56	37
6	31/8/2020 12:57	38
7	31/8/2020 12:58	37
8	31/8/2020 12:59	38
9	31/8/2020 13:00	37
10	31/8/2020 13:01	39
11	31/8/2020 13:02	38
12	31/8/2020 13:03	37
13	31/8/2020 13:04	39
14	31/8/2020 13:08	38
15	31/8/2020 13:09	37
16	31/8/2020 13:11	38
17	31/8/2020 13:12	38
18	31/8/2020 13:13	38
19	31/8/2020 13:15	38
20	31/8/2020 13:16	37
21	31/8/2020 13:17	38
22	31/8/2020 13:18	38
23	31/8/2020 13:19	39
24	31/8/2020 13:21	38
25	31/8/2020 13:22	38
26	31/8/2020 13:23	37
27	31/8/2020 13:25	38

Sigue en la página siguiente.

id	Fecha	HumDHT11
28	31/8/2020 13:26	36
29	31/8/2020 13:31	37
30	31/8/2020 13:32	37
31	31/8/2020 13:33	37
32	31/8/2020 13:34	38
33	31/8/2020 13:35	37
34	31/8/2020 13:36	37
35	31/8/2020 13:37	37
36	31/8/2020 13:38	38
37	31/8/2020 13:39	37
38	31/8/2020 13:40	39
39	31/8/2020 13:41	38
40	31/8/2020 13:42	38
41	31/8/2020 13:43	38
42	31/8/2020 13:44	37
43	1/9/2020 8:53	38
44	1/9/2020 8:54	39
45	1/9/2020 8:55	39
46	1/9/2020 8:56	39
47	1/9/2020 8:57	40
48	1/9/2020 8:58	39
49	1/9/2020 8:59	39
50	1/9/2020 9:00	39
51	1/9/2020 9:01	39
52	1/9/2020 9:02	39
53	1/9/2020 9:03	39
54	1/9/2020 9:04	38
55	1/9/2020 9:05	39
56	1/9/2020 9:06	40
57	1/9/2020 9:07	39
58	1/9/2020 9:08	38
59	1/9/2020 9:09	40
60	1/9/2020 9:10	39
61	1/9/2020 9:11	39
62	1/9/2020 9:12	40
63	1/9/2020 9:13	39
64	1/9/2020 9:14	38
65	1/9/2020 9:15	38
66	1/9/2020 9:16	38
67	1/9/2020 9:17	39
68	1/9/2020 9:18	39

Sigue en la página siguiente.

id	Fecha	HumDHT11
69	1/9/2020 9:19	39
70	1/9/2020 9:20	39
71	1/9/2020 9:21	38
72	1/9/2020 9:22	41
73	1/9/2020 9:23	39
74	1/9/2020 9:24	39
75	1/9/2020 9:25	40
76	1/9/2020 9:26	39
77	1/9/2020 9:27	39
78	1/9/2020 9:29	38
79	1/9/2020 9:30	39
80	1/9/2020 9:31	40
81	1/9/2020 9:33	39
82	1/9/2020 9:34	38
83	1/9/2020 9:35	40
84	1/9/2020 9:36	39
85	1/9/2020 9:37	38
86	1/9/2020 9:38	38
87	1/9/2020 9:39	40
88	1/9/2020 9:40	38
89	1/9/2020 9:41	38
90	1/9/2020 9:42	39
91	1/9/2020 9:43	38
92	1/9/2020 9:44	38
93	1/9/2020 9:45	37
94	1/9/2020 9:46	39
95	1/9/2020 9:47	38
96	1/9/2020 9:48	38
97	1/9/2020 9:49	38
98	1/9/2020 9:50	38
99	1/9/2020 9:51	37
100	1/9/2020 9:52	37

Tabla C.1: Datos de humedad registrados en la base de datos

id	Fecha	TempDHT11
1	31/8/2020 12:52	23
2	31/8/2020 12:53	24
3	31/8/2020 12:54	24
4	31/8/2020 12:55	24

Sigue en la página siguiente.

id	Fecha	TempDHT11
5	31/8/2020 12:56	24
6	31/8/2020 12:57	24
7	31/8/2020 12:58	24
8	31/8/2020 12:59	24
9	31/8/2020 13:00	24
10	31/8/2020 13:01	24
11	31/8/2020 13:02	24
12	31/8/2020 13:03	24
13	31/8/2020 13:04	24
14	31/8/2020 13:08	24
15	31/8/2020 13:09	24
16	31/8/2020 13:11	24
17	31/8/2020 13:12	24
18	31/8/2020 13:13	24
19	31/8/2020 13:15	24
20	31/8/2020 13:16	24
21	31/8/2020 13:17	24
22	31/8/2020 13:18	24
23	31/8/2020 13:19	24
24	31/8/2020 13:21	24
25	31/8/2020 13:22	24
26	31/8/2020 13:23	24
27	31/8/2020 13:25	24
28	31/8/2020 13:26	25
29	31/8/2020 13:31	24
30	31/8/2020 13:32	24
31	31/8/2020 13:33	24
32	31/8/2020 13:34	24
33	31/8/2020 13:35	24
34	31/8/2020 13:36	24
35	31/8/2020 13:37	24
36	31/8/2020 13:38	24
37	31/8/2020 13:39	24
38	31/8/2020 13:40	24
39	31/8/2020 13:41	24
40	31/8/2020 13:42	24
41	31/8/2020 13:43	24
42	31/8/2020 13:44	24
43	1/9/2020 8:53	22
44	1/9/2020 8:54	22
45	1/9/2020 8:55	22

Sigue en la página siguiente.

id	Fecha	TempDHT11
46	1/9/2020 8:56	22
47	1/9/2020 8:57	22
48	1/9/2020 8:58	22
49	1/9/2020 8:59	22
50	1/9/2020 9:00	22
51	1/9/2020 9:01	22
52	1/9/2020 9:02	22
53	1/9/2020 9:03	22
54	1/9/2020 9:04	22
55	1/9/2020 9:05	22
56	1/9/2020 9:06	22
57	1/9/2020 9:07	22
58	1/9/2020 9:08	22
59	1/9/2020 9:09	22
60	1/9/2020 9:10	22
61	1/9/2020 9:11	22
62	1/9/2020 9:12	22
63	1/9/2020 9:13	22
64	1/9/2020 9:14	22
65	1/9/2020 9:15	22
66	1/9/2020 9:16	22
67	1/9/2020 9:17	22
68	1/9/2020 9:18	22
69	1/9/2020 9:19	22
70	1/9/2020 9:20	22
71	1/9/2020 9:21	22
72	1/9/2020 9:22	22
73	1/9/2020 9:23	22
74	1/9/2020 9:24	22
75	1/9/2020 9:25	22
76	1/9/2020 9:26	22
77	1/9/2020 9:27	22
78	1/9/2020 9:29	22
79	1/9/2020 9:30	22
80	1/9/2020 9:31	22
81	1/9/2020 9:33	22
82	1/9/2020 9:34	22
83	1/9/2020 9:35	22
84	1/9/2020 9:36	22
85	1/9/2020 9:37	22
86	1/9/2020 9:38	22

Sigue en la página siguiente.

id	Fecha	TempDHT11
87	1/9/2020 9:39	22
88	1/9/2020 9:40	23
89	1/9/2020 9:41	23
90	1/9/2020 9:42	22
91	1/9/2020 9:43	23
92	1/9/2020 9:44	22
93	1/9/2020 9:45	23
94	1/9/2020 9:46	22
95	1/9/2020 9:47	23
96	1/9/2020 9:48	23
97	1/9/2020 9:49	23
98	1/9/2020 9:50	23
99	1/9/2020 9:51	23
100	1/9/2020 9:52	23

Tabla C.2: Datos de temperatura registrados en la base de datos

id	Fecha	SensorAna
1	31/8/2020 12:52	7,36
2	31/8/2020 12:53	7,36
3	31/8/2020 12:53	7,26
4	31/8/2020 12:55	7,26
5	31/8/2020 12:56	7,26
6	31/8/2020 12:56	7,26
7	31/8/2020 12:57	7,26
8	31/8/2020 12:58	7,26
9	31/8/2020 12:58	7,26
10	31/8/2020 12:59	7,26
11	31/8/2020 12:59	7,26
12	31/8/2020 13:00	7,26
13	31/8/2020 13:01	7,26
14	31/8/2020 13:01	7,26
15	31/8/2020 13:02	7,26
16	31/8/2020 13:03	7,26
17	31/8/2020 13:03	7,26
18	31/8/2020 13:04	7,26
19	31/8/2020 13:04	7,26
20	31/8/2020 13:05	7,26
21	31/8/2020 13:06	7,26
22	31/8/2020 13:07	7,26

Sigue en la página siguiente.

id	Fecha	SensorAna
23	31/8/2020 13:09	7,26
24	31/8/2020 13:09	7,26
25	31/8/2020 13:10	7,26
26	31/8/2020 13:11	7,26
27	31/8/2020 13:11	7,26
28	31/8/2020 13:13	7,26
29	31/8/2020 13:13	7,26
30	31/8/2020 13:14	7,26
31	31/8/2020 13:16	7,26
32	31/8/2020 13:16	7,26
33	31/8/2020 13:17	7,26
34	31/8/2020 13:17	7,26
35	31/8/2020 13:18	7,26
36	31/8/2020 13:18	7,26
37	31/8/2020 13:19	7,26
38	31/8/2020 13:20	7,26
39	31/8/2020 13:21	7,26
40	31/8/2020 13:21	7,26
41	31/8/2020 13:22	7,26
42	31/8/2020 13:23	7,26
43	31/8/2020 13:25	7,26
44	31/8/2020 13:25	7,26
45	31/8/2020 13:26	7,26
46	31/8/2020 13:26	7,26
47	31/8/2020 13:27	7,26
48	31/8/2020 13:30	7,26
49	31/8/2020 13:31	7,26
50	31/8/2020 13:34	7,26
51	31/8/2020 13:34	7,26
52	31/8/2020 13:35	7,26
53	31/8/2020 13:35	7,26
54	31/8/2020 13:35	7,26
55	31/8/2020 13:35	7,26
56	31/8/2020 13:35	7,26
57	31/8/2020 13:35	7,26
58	31/8/2020 13:35	7,26
59	31/8/2020 13:35	7,26
60	31/8/2020 13:35	7,26
61	31/8/2020 13:35	7,26
62	31/8/2020 13:35	7,26
63	31/8/2020 13:35	7,26

Sigue en la página siguiente.

id	Fecha	SensorAna
64	31/8/2020 13:35	7,26
65	31/8/2020 13:35	7,26
66	31/8/2020 13:35	7,26
67	31/8/2020 13:35	7,26
68	31/8/2020 13:35	7,26
69	31/8/2020 13:35	7,26
70	31/8/2020 13:35	7,26
71	31/8/2020 13:35	7,26
72	31/8/2020 13:35	7,26
73	31/8/2020 13:35	7,26
74	31/8/2020 13:35	7,26
75	31/8/2020 13:35	7,26
76	31/8/2020 13:35	7,26
77	31/8/2020 13:35	7,26
78	31/8/2020 13:35	7,26
79	31/8/2020 13:35	7,26
80	31/8/2020 13:35	7,26
81	31/8/2020 13:35	7,26
82	31/8/2020 13:35	7,26
83	31/8/2020 13:35	7,26
84	31/8/2020 13:35	7,26
85	31/8/2020 13:35	7,26
86	31/8/2020 13:35	7,26
87	31/8/2020 13:35	7,26
88	31/8/2020 13:35	7,26
89	31/8/2020 13:35	7,26
90	31/8/2020 13:35	7,26
91	31/8/2020 13:35	7,26
92	31/8/2020 13:35	7,26
93	31/8/2020 13:35	7,26
94	31/8/2020 13:35	7,26
95	31/8/2020 13:35	7,26
96	31/8/2020 13:35	7,26
97	31/8/2020 13:35	7,26
98	31/8/2020 13:35	7,26
99	31/8/2020 13:35	7,26

Tabla C.3: Datos de voltaje registrados en la base de datos

# Tesis - Ismael Carangui Rev2

---

## INFORME DE ORIGINALIDAD

---

8%

INDICE DE SIMILITUD

6%

FUENTES DE  
INTERNET

1%

PUBLICACIONES

4%

TRABAJOS DEL  
ESTUDIANTE

---

ENCONTRAR COINCIDENCIAS CON TODAS LAS FUENTES (SOLO SE IMPRIMIRÁ LA FUENTE SELECCIONADA)

---

1%

★ [dspace.ucuenca.edu.ec](https://dspace.ucuenca.edu.ec)

Fuente de Internet

---

Excluir citas

Activo

Excluir coincidencias

< 8 words

Excluir bibliografía

Activo

## PERMISO DEL AUTOR DE TESIS PARA SUBIR AL REPOSITORIO INSTITUCIONAL

Yo, **WILSON ISMAEL CARANGUI RODRIGUEZ**, portador (a) de la cédula de ciudadanía Nro. **0302045091**. En calidad de autor/a y titular de los derechos patrimoniales del trabajo de Titulación “**SISTEMA SCADA DE BAJO COSTO PARA LA INTEGRACIÓN DE DISPOSITIVOS HETEROGÉNEOS DEL LABORATORIO DE PLC’s DE LA UNIERSIDAD CATÓLICA DE CUENCA**” de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de Los Conocimientos, Creatividad e Innovación, reconozco a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos, Así mismo; autorizo a la Universidad para que realice la publicación de éste trabajo de titulación en Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior.

Azogues, 08 de septiembre de 2020

F: 

.....  
**WILSON ISMAEL CARANGUI RODRIGUEZ**

**0302045091**

# El Bibliotecario de la Sede Azogues

## CERTIFICA:

Que: **CARANGUI RODRÍGUEZ WILSON ISMAEL**, con cédula de ciudadanía Nro. **0302045091**, de la Carrera de: **INGENIERÍA ELECTRÓNICA**

No adeuda libros, a esta fecha: **08 de septiembre del 2020**.



Byron Alonso Torres Romo  
**Bibliotecario**

Biblioteca Universitaria  
MONS. "FROILAN POZO QUEVEDO"