



UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD EN CIENCIAS DE LA INGENIERÍA Y APLICADA

CARRERA DE INGENIERÍA EN ELECTRICIDAD

PROPUESTA TECNOLÓGICA

DESARROLLO DE UN CONTROLADOR LÓGICO PROGRAMABLE
CON CAPACIDADES IOT MEDIANTE EL USO DE SOFTWARE
LIBRE

Propuesta tecnológica presentada previo a la obtención del Título de
Ingeniero en Electricidad

Autores:

Cristian Fernando Alvear Romero
Lenin Alexander Arellano Sangacha

Tutor:

Ing. Byron Paúl Corrales Bastidas

LATACUNGA – ECUADOR

FEBRERO - 2025

DECLARACIÓN DE AUTORÍA

Alvear Romero Cristian Fernando, con cédula de ciudadanía No. 2100886890, Arellano Sangacha Lenin Alexander, con cédula de ciudadanía No. 2100864681 declaramos ser autores de la presente **PROPUESTA TECNOLÓGICA: “DESARROLLO DE UN CONTROLADOR LÓGICO PROGRAMABLE CON CAPACIDADES IOT MEDIANTE EL USO DE SOFTWARE LIBRE”**, siendo el Ing. Byron Paúl Corrales Bastidas, Tutor del presente trabajo; y, eximo expresamente a la Universidad Técnica de Cotopaxi y a sus representantes legales de posibles reclamos o acciones legales.

Además, certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Latacunga, 20 de febrero de 2025



Cristian Fernando Alvear Romero
CC: 2100886890



Lenin Alexander Arellano Sangacha
CC: 2100864681

AVAL DEL TUTOR DE LA PROPUESTA TECNOLÓGICA

En calidad de Tutor de la Propuesta Tecnológica sobre el título:

“DESARROLLO DE UN CONTROLADOR LÓGICO PROGRAMABLE CON CAPACIDADES IOT MEDIANTE EL USO DE SOFTWARE LIBRE”, de: Alvear Romero Cristian Fernando y Arellano Sangacha Lenin Alexander, de la Carrera de Ingeniería en Electricidad, considero que dicha Propuesta Tecnológica es merecedor del aval de aprobación al cumplir las normas técnicas, traducción y formatos previstos, así como también ha incorporado las observaciones y recomendaciones propuestas en la pre-defensa.

Latacunga, 20 de febrero de 2025



Byron Paúl Corrales Bastidas
CC: 0502347768
TUTOR

AVAL DE APROBACIÓN DEL TRIBUNAL DE TITULACIÓN

En calidad de Tribunal de Lectores, aprueban el presente Propuesta Tecnol de acuerdo a las disposiciones reglamentarias emitidas por la Universidad Técnica de Cotopaxi, y, por la Facultad de Ciencias de la Ingeniería y Aplicadas; por cuanto, los postulantes: Alvear Romero Cristian Fernando y Arellano Sangacha Lenin Alexander, con el título de la Propuesta Tecnológica: **“DESARROLLO DE UN CONTROLADOR LÓGICO PROGRAMABLE CON CAPACIDADES IOT MEDIANTE EL USO DE SOFTWARE LIBRE”**, ha considerado las recomendaciones emitidas oportunamente y reúne los méritos suficientes para ser sometido al acto de sustentación del trabajo de titulación.

Por lo antes expuesto, se autoriza grabar los archivos correspondientes en un CD, según la normativa institucional.

Latacunga, 20 de febrero de 2025

Para constancia firman:



Diego Leonardo Jiménez Jiménez
CC: 0503493702
LECTOR 1 (PRESIDENTE)



Marco Aníbal León Segovia
CC: 0502305402
LECTOR 2 (MIEMBRO)



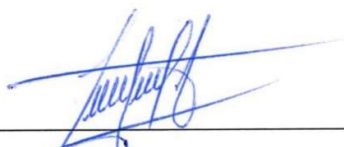
Manuel Ángel León Segovia
CC: 0502041353
LECTOR 3 (MIEMBRO)

AVAL DE IMPLEMENTACIÓN

En calidad de Tutor de la Propuesta Tecnológica sobre el título:

“DESARROLLO DE UN CONTROLADOR LÓGICO PROGRAMABLE CON CAPACIDADES IOT MEDIANTE EL USO DE SOFTWARE LIBRE”, de: Alvear Romero Cristian Fernando, con cédula de ciudadanía No. 2100886890, Arellano Sangacha Lenin Alexander, con cédula de ciudadanía No. 2100864681, de la Carrera de Ingeniería en Electricidad realizan la entrega del proyecto de titulación en pleno funcionamiento.

Latacunga, febrero de 2025


Byron Paúl Corrales Bastidas
CC: 0502347768
TUTOR

AGRADECIMIENTO

Con profundo respeto y gratitud, quiero dedicar este logro a mis padres, Luz Romero y Fernando Alvear y a mi hermana, Tania Alvear, quienes han sido mis pilares fundamentales, brindándome su amor, apoyo incondicional y guía en cada paso de mi camino. Gracias a ellos, he podido perseguir y alcanzar mis metas con determinación y perseverancia.

A mi querida novia, Keyla Flores, por su amor, comprensión y apoyo inagotable. Su presencia ha sido una fuente de inspiración y fortaleza en este proceso.

A la Universidad Técnica de Cotopaxi, por abrirme sus puertas y permitirme crecer académicamente en sus aulas, y a sus docentes, quienes con su dedicación y conocimientos han contribuido de manera invaluable a mi formación profesional.

A todos quienes, de una u otra manera, han sido parte de este camino, les expreso mi más sincero agradecimiento.

Cristian Alvear

DEDICATORIA

Dedico este trabajo y esfuerzo a las personas más importantes de mi vida.

A mis padres, Luz Romero y Fernando Alvear y mi hermana, Tania Alvear, quienes con su amor, sacrificio y enseñanzas me han guiado en cada paso de mi camino. Su apoyo incondicional ha sido el motor que me impulsa a seguir adelante y luchar por mis sueños.

A mi amada novia, Keyla Flores, por ser mi refugio, mi motivación y mi fuente constante de inspiración. Su amor y compañía han sido fundamentales en este proceso, y cada logro es también suyo.

A mis docentes y amigos, quienes con sus enseñanzas y apoyo han enriquecido mi formación y han sido parte de esta etapa tan importante de mi vida.

Con gratitud y cariño, dedico este trabajo a todos ustedes.

Cristian Alvear

AGRADECIMIENTO

En primer lugar, doy gracias a Dios, quien ha sido mi guía y me ha brindado la fortaleza necesaria para seguir adelante en cada etapa de este proceso.

A mis padres, Vinicio Arellano y Doris Sangacha por su amor incondicional, su apoyo incesante y por ser el pilar fundamental en mi vida personal.

A mis herman@s, Alexandra Arellano, Daniela Arellano, Jaqueline Arellano y Jhan Carlos Arellano por su comprensión, estímulo constante y por acompañarme en cada paso de este camino.

A la Universidad Técnica de Cotopaxi, por brindarme las herramientas, conocimientos y oportunidades que han hecho posible mi formación profesional.

A todos quienes, de una u otra manera, han sido parte de este camino, les expreso mi más sincero agradecimiento.

Lenin Arellano

DEDICATORIA

A mis padres, Vinicio Arellano y Doris Sangacha pilares fundamentales de mi vida, por su amor incondicional, su sacrificio silencioso y su apoyo constante en cada paso de mi formación.

A mis herman@s, Alexandra Arellano, Daniela Arellano, Jaqueline Arellano y Jhan Carlos Arellano compañeros de vida, por su alegría, su comprensión y su apoyo incondicional.

Les dedico este logro como un pequeño reflejo de mi gratitud y admiración hacia ustedes.

Lenin Arellano

ÍNDICE GENERAL

DECLARACIÓN DE AUTORÍA.....	i
AVAL DEL TUTOR DE LA PROPUESTA TECNOLÓGICA.....	ii
AVAL DE APROBACIÓN DEL TRIBUNAL DE TITULACIÓN	iii
AVAL DE IMPLEMENTACIÓN	iv
ÍNDICE GENERAL.....	ix
ÍNDICE DE FIGURAS.....	xiii
ÍNDICE DE TABLAS	xv
RESUMEN.....	xvi
ABSTRACT.....	xvii
AVAL DE TRADUCCIÓN	xviii
1. INFORMACIÓN GENERAL	1
2. INTRODUCCIÓN.....	2
2.1. SITUACIÓN PROBLEMÁTICA	3
2.2. FORMULACIÓN DEL PROBLEMA.....	3
2.3. OBJETO Y CAMPO DE ACCIÓN	3
2.3.1. Objeto de investigación:.....	3
2.3.2. Campo de acción:.....	3
2.4. BENEFICIARIOS.....	3
2.4.1. Directo.....	3
2.4.2. Indirecto	3
2.5. JUSTIFICACIÓN.....	4
2.6. OBJETIVOS	5
2.6.1. General.....	5
2.6.2. Específicos	5
2.6.3. Sistemas de tareas.....	5
3. FUNDAMENTACIÓN TEÓRICA	6
3.1. ANTECEDENTES.....	6
3.2. DEFINICIÓN DE UN CONTROLADOR LÓGICO PROGRAMABLE.....	7
3.3. PLC EN EL CAMPO DE AUTOMATIZACIÓN	7
3.4. AUTOMATIZACIÓN CON LÓGICA CABLEADA	8

3.4.1.	Ventajas.....	8
3.4.2.	Desventajas.....	8
3.5.	VENTAJAS DE AUTOMATIZACIÓN CON PLC	8
3.6.	DESVENTAJAS DE AUTOMATIZACIÓN CON PLC.....	9
3.7.	COMPARACIÓN LÓGICA CABLEADA VS PLC	9
3.8.	ESTRUCTURA BÁSICA DEL PLC	10
3.8.1.	La unidad central de proceso.....	10
3.8.2.	Memorias.....	11
3.8.2.1.	Memoria del programa	11
3.8.2.2.	Memoria de datos	11
3.8.3.	Entradas.....	11
3.8.4.	Salidas	11
3.8.5.	Fuentes de alimentación	12
3.8.6.	Puertos de comunicación.....	12
3.9.	TIPOS DE SEÑALES.....	12
3.10.	TIPOS DE PLC	17
3.10.1.	Clasificación PLC	18
3.11.	LENGUAJE DE PROGRAMACIÓN PLC	19
3.12.	ARDUINO IDE.....	20
3.13.	SOFTWARE OPENPLC.....	21
3.14.	BASE DE DATOS FIREBASE	22
3.15.	SOFTWARE EASYEDA.....	22
3.16.	INTERNET DE LAS COSAS (IOT)	23
3.17.	APLICACIONES Y BENEFICIOS DEL IOT.....	23
4.	MÉTODOS Y PROCEDIMIENTOS	24
4.1.	DESCRIPCIÓN DEL PROYECTO	24
4.2.	DECLARACIÓN DE VARIABLES	24
4.3.	ESQUEMA GENERAL DE FUNCIONAMIENTO	25
4.4.	DETALLE DE COMPONENTES DEL SISTEMA	26
4.4.1.	Microcontrolador Arduino Nano.....	26
4.4.2.	Microcontrolador ESP32.....	26

4.4.3.	LM2596 Módulo Regulador.....	26
4.4.4.	Circuitos integrados y componentes pasivos.....	27
4.5.	DISEÑO DEL SISTEMA	27
4.5.1.	Cálculos y dimensionamientos de los componentes	28
4.5.1.1.	Entradas digitales	28
4.5.1.2.	Entradas Analógicas.....	28
4.5.1.3.	Salida Analógica	29
4.5.1.4.	Entradas analógicas al ESP32	29
4.5.1.5.	Dimensionamiento del fusible de protección	30
4.5.2.	Diagrama de flujo del sistema.....	31
4.5.3.	Diagrama de conexión.....	32
4.6.	DESARROLLO DEL SISTEMA.....	36
4.6.1.	Creación de la base de datos en Firebase	36
4.6.2.	Diseño de la placa del sistema.....	38
4.6.3.	Programación del ESP32.....	39
4.6.4.	Desarrollo de una App para interactuar con el PLC.....	41
4.6.5.	Diseño de la carcasa del sistema	43
4.6.5.1.	Nivel de protección IP	45
5.	ANÁLISIS DE RESULTADOS.....	46
5.1.	PRUEBA DE VALIDACIÓN DEL PROTOTIPO	46
5.1.1.	Eficacia de la programación.....	46
5.1.2.	Pruebas de comunicación con Firebase.....	47
5.1.3.	Programación con OpenPLC.....	47
5.2.	FUNCIONALIDAD DE LA APLICACIÓN DESARROLLADA EN APPINVENTOR	
	48	
5.2.1.	Experiencia de uso	48
5.2.2.	Capacidad de control y monitoreo remoto	48
5.3.	PRUEBAS DE CAMPO Y ESTUDIOS DE CASO	49
5.3.1.	Pruebas y Resultados.....	49
5.3.1.1.	Encendido de una lámpara modo blink	49
5.3.1.2.	Encendido de una lámpara por un pulsador	50
5.3.1.3.	Enclavamiento de un contactor	50
5.3.1.4.	Control de un semáforo de una vía.....	51

5.3.1.5.	Contador de eventos	53
5.3.1.6.	Control de nivel de un tanque (On/Off)	54
5.3.2.	Aplicaciones IoT	57
5.3.2.1.	Control y monitoreo mediante IoT	57
5.3.2.2.	Envío de datos IoT	59
5.3.3.	Evaluación del Prototipo en Laboratorio con Estudiantes	60
5.3.3.1.	Descripción de la práctica de laboratorio	60
5.3.3.2.	Resultados y análisis	60
5.4.	IDENTIFICACIÓN DE ÁREAS DE MEJORA	67
5.4.1.	Posibilidad de añadir un protocolo de comunicación Modbus.....	67
5.4.2.	Optimización del consumo energético	68
5.4.3.	Incremento de la seguridad en la comunicación IoT	68
5.4.4.	Escalabilidad del sistema	69
5.5.	ANÁLISIS DE COSTOS	69
5.5.1.	Costos de materiales	69
5.5.2.	Costo de desarrollo.....	70
5.5.3.	Gastos totales del PLC	70
6.	CONCLUSIONES Y RECOMENDACIONES	71
6.1.	CONCLUSIONES	71
6.2.	RECOMENDACIONES	71
7.	REFERENCIAS	72

ÍNDICE DE FIGURAS

Figura 1. Estructura básica del PLC.	10
Figura 2. Conexión de un sensor PNP.	13
Figura 3. Conexión de un sensor NPN.	13
Figura 4. Entrada de corriente.	14
Figura 5. Salida tipo voltaje.	14
Figura 6. Diagrama de conexión de un sensor inductivo.	15
Figura 7. Diagrama de conexión de un sensor NPN.	16
Figura 8. Diagrama de conexión tipo corriente.	16
Figura 9. Diagrama de conexión de un relé.	17
Figura 10. Tipos de PLC.	18
Figura 11. Clasificación de los PLCs Siemens.	19
Figura 12. Interfaz del software OpenPLC.	21
Figura 13. Ventana para cargar el programa en el PLC.	22
Figura 14. Esquema general de funcionamiento.	25
Figura 15. Diagrama de flujo del sistema.	31
Figura 16. Alimentación del PLC.	32
Figura 17. Diagrama para las entradas digitales del PLC.	32
Figura 18. Diagrama para el aislamiento de ruido en las entradas digitales.	33
Figura 19. Diagrama para las entradas analógicas.	34
Figura 20. Diagrama del Arduino Nano.	34
Figura 21. Diagrama de las salidas digitales.	35
Figura 22. Diagrama para la salida analógica.	35
Figura 23. Diagrama de conexión del ESP32.	36
Figura 24. Base de datos creadas para trabajar con el PLC.	37
Figura 25. Plan gratuito de Rea Time Database.	37
Figura 26. Diseño de la PCB del PLC.	38
Figura 27. Selección de la placa ESP32 Dev Module.	39
Figura 28. Librerías utilizadas para la programación del ESP32.	39
Figura 29. Variables declaradas en el código antes del Setup.	40
Figura 30. Inicialización del código en el ESP32.	40
Figura 31. Bucle del código en el ESP32.	41
Figura 32. Proyecto creado para realizar la App.	41
Figura 33. Construcción de la interfaz gráfica de la App.	42
Figura 34. Programación por bloques usando AppInventor.	42

Figura 35. Interfaz de la App vista en el celular.....	43
Figura 36. Diseño de la base de la caja del PLC.	44
Figura 37. Diseño de la tapa que cubre al PLC.	44
Figura 38. Impresión de la caja utilizando la impresora 3D.....	44
Figura 39. Monitor serial que muestra las impresiones serial del ESP32.	46
Figura 40. Datos guardados en Real Time Database de Firebase.	47
Figura 41. Interfaz y programación en OpenPLC.	47
Figura 42. Diagrama de conexión encendido de lámpara dos modos.	49
Figura 43. Programa del encendido de una lampara (Blink).....	50
Figura 44. Programa del encendido de una lampara (Pulsador).....	50
Figura 45. Diagrama de conexión enclavamiento de un contactor.	51
Figura 46. Programa de enclavamiento de un contactor.	51
Figura 47. Diagrama de conexión (Semáforo).	52
Figura 48. Programa de control del semáforo una vía.....	52
Figura 49. Diagrama de conexión contador de eventos.....	53
Figura 50. Programa de contador de eventos.	54
Figura 51. Diagrama de conexión control ON/OFF.....	55
Figura 52. Nivel del tanque (Probado).	56
Figura 53. Programa control ON/OFF.....	56
Figura 54. Programa control PID.	57
Figura 55. Monitoreo de entradas digitales.	58
Figura 56. Monitoreo horario.	58
Figura 57. Control y Monitoreo de entradas y salidas.....	59
Figura 58. Monitoreo en tiempo real.	59

ÍNDICE DE TABLAS

Tabla 1. Sistema de tareas utilizadas en el desarrollo del documento.....	5
Tabla 2. Comparación de la funcionabilidad.....	10
Tabla 3. Descripción de los componentes del PLC.....	27
Tabla 4. Comparación de los diferentes tipos de filamentos.....	45
Tabla 5. Mediciones del voltaje en el sensor.....	55
Tabla 6. Costo de materiales para la construcción del PLC.....	69
Tabla 7. Costos de desarrollo del PLC.....	70
Tabla 8. Gastos totales del PLC.....	70

UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

TÍTULO: “DESARROLLO DE UN CONTROLADOR LÓGICO PROGRAMABLE CON CAPACIDADES IOT MEDIANTE EL USO DE SOFTWARE LIBRE”

Autores:

Alvear Romero Cristian Fernando

Arellano Sangacha Lenin Alexander

RESUMEN

Este proyecto desarrolló un PLC IoT utilizando hardware de bajo costo y software de código abierto, con el objetivo de proporcionar una alternativa asequible para implementar procesos de automatización. Se utilizó un Arduino Nano para el procesamiento, OpenPLC para la programación y un ESP32 para crear conexiones IoT con los datos almacenados en Firebase de Google. Se sabe que los PLCs comunes son costosos, carecen de funcionalidades IoT y no pueden implementarse fácilmente en entornos educativos y laborales. Este proyecto se centra en la creación de un PLC rentable, lo cual es un enfoque metodológico importante para mejorar la capacitación de los estudiantes y también para utilizar aplicaciones de automatización industrial. Con el PLC, se realizaron pruebas que ayudaron a validar la funcionalidad del sistema en la automatización de procesos; se obtuvo una transmisión de datos estable mediante el protocolo de comunicación UDP desarrollado, así como una integración efectiva con OpenPLC y Firebase. En las pruebas iniciales, el PLC controló exitosamente tanto las entradas y salidas digitales como las analógicas, respondió rápidamente a los cambios cuando se operó de forma remota y permitió la supervisión en tiempo real a través de una aplicación móvil. Entre las pruebas realizadas se verificó el porcentaje de error de la señal analógica equivalente como parte del IoT, con un error máximo del 10% en el signo entre la señal real registrada en la entrada del PLC. Se realizaron experimentos junto con estudiantes de la Universidad Técnica de Cotopaxi para validar la experiencia del usuario con el PLC, utilizando una guía y manuales que se desarrollaron para integrar el PLC con OpenPLC y la aplicación para automatizar procesos. Estas prácticas permitieron identificar mediante encuestas posibles mejoras en la interfaz y funcionalidades, demostrando una gran facilidad en el manejo de software, hardware y el IoT por parte de los usuarios.

Palabras clave: PLC, IoT, OpenPLC, Firebase, Automatización.

COTOPAXI TECHNICAL UNIVERSITY

FACULTY ENGINEERING AND APPLIED SCIENCES

TOPIC: “DEVELOPMENT OF A PROGRAMMABLE LOGIC CONTROLLER WITH IOT CAPABILITIES USING OPEN-SOURCE SOFTWARE”

Authors:

Alvear Romero Cristian Fernando

Arellano Sangacha Lenin Alexander

ABSTRACT

This project developed an IoT PLC using low-cost hardware and open-source software, to provide an affordable alternative for implementing automation processes. An Arduino Nano was used for processing, OpenPLC for programming, and an ESP32 to create IoT connections to data stored in Google's Firebase. Common PLCs are known to be expensive, lack IoT functionalities, and cannot be easily implemented in educational and work environments. This project focuses on creating a cost-effective PLC, which is an important methodological approach to improve student training and also to use industrial automation applications. With the PLC, tests were conducted that helped validate the system's functionality in process automation; stable data transmission was obtained using the developed UDP communication protocol, as well as effective integration with OpenPLC and Firebase. In initial tests, the PLC successfully controlled both digital and analog inputs and outputs, responded quickly to changes when operated remotely, and enabled real-time monitoring via a mobile application. Among the tests performed, the percentage error of the equivalent analog signal was verified as part of the IoT, with a maximum error of 10% in the sign between the real signal recorded at the PLC input. Experiments were conducted together with students from the Technical University of Cotopaxi to validate the user experience with the PLC, using a guide and manuals that were developed to integrate the PLC with OpenPLC and the application to automate processes. These practices allowed to identify through surveys possible improvements in the interface and functionalities, demonstrating great ease in the management of software, hardware, and IoT by the users.

Keywords: PLC, IoT, OpenPLC, Firebase, Automation.

UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

CARRERA: INGENIERÍA EN ELECTRICIDAD


Autores: Cristian Fernando Alvear Romero, Lenin Alexander Arellano Sangacha

AVAL DE TRADUCCIÓN- Profesional Externo

Yadira Elizabeth Quiroz Chacán con cédula de identidad número:0250278330 Licenciada/o; Magíster en:, solo Pedagogía del idioma inglés con número de registro de la SENESCYT:1010-2023-2634372; **CERTIFICO** haber revisado y aprobado la traducción al idioma Inglés del resumen de la propuesta tecnológica con el título: **“DESARROLLO DE UN CONTROLADOR LÓGICO PROGRAMABLE CON CAPACIDADES IoT MEDIANTE EL USO DE SOFTWARE LIBRE ”** de: **Cristian Fernando Alvear Romero, Lenin Alexander Arellano Sangacha**, egresados de la carrera de **Ingeniería en Electricidad**, perteneciente a la Facultad de: CIENCIAS DE LA INGENIERIA Y APLICADAS.

En virtud de lo expuesto y para constancia de lo mismo se registra la firma respectiva.

Latacunga, febrero del 2025


.....
Lic. Quiroz Chacán Yadira Elizabeth
CI: 0250278330

1. INFORMACIÓN GENERAL

Tema del proyecto: Desarrollo de un controlador lógico programable con capacidades IoT mediante el empleo de hardware y software libre.

Modalidad de Titulación:

Propuestas Tecnológicas	<input checked="" type="checkbox"/>
Proyectos de Investigación	<input type="checkbox"/>

Carrera: Carrera de Ingeniería en Electricidad.

Trabajo de Titulación Vinculado al Proyecto: Desarrollo de sistemas eficientes para el abastecimiento y uso de energía eléctrica a nivel local, regional o nacional.

Equipo de Trabajo del Trabajo de Titulación:

Alvear Romero Cristian Fernando

Arellano Sangacha Lenin Alexander

Tutor de Titulación:

Corrales Bastidas Byron Paúl

Área de conocimiento: 07 Ingeniería, Industria y Construcción / 071 Ingeniería y Profesiones Afines / 0713 Electricidad y Energía.

Línea de investigación: 04. Procesos industriales.

Sublínea de investigación de la carrera:

Automatización, control y protección de sistemas eléctricos en el sector productivo y los servicios.

2. INTRODUCCIÓN

Los Controladores Lógicos Programables son dispositivos fundamentales en la automatización industrial moderna que ayudan a administrar y controlar procesos de una manera precisa, rápida, efectiva y confiable. Desde que se presentaron por primera vez como alternativa a los sistemas convencionales basados en relés, los PLC revolucionaron la forma en que operan las industrias, permitiendo una flexibilidad y adaptabilidad nunca antes vista en procesos a menudo extremadamente precisos.

Los PLC no solo controlan, sino que actualmente se conectan con tecnologías de nuevas como IoT (Internet de las cosas). Por lo tanto, los PLCs pueden comunicarse con otros sistemas en tiempo real, recopilar datos operativos y controles para aprovechar la eficacia y la transformación a través del análisis de conjuntos de datos. Esto hace que se integren a la Industria 4.0.

En esta propuesta tecnológica se desarrolla un controlador lógico programable con capacidades IoT basado en un microcontrolador Atmega328P del Arduino Nano que es el cerebro del PLC propuesto al que se conectarán las entradas y salidas, tanto digitales como analógicas, aquí se encontrará cargado el código que será previamente realizado en el software OpenPLC, ya que consta con las funciones necesarias para realizar los programas de automatización y control. Además, cumple con la normativa IEC 61131-3 para diseñar los proyectos de PLC en lógica de escalera[1], texto estructurado, lista de instrucciones, diagrama de bloques de funciones o diagrama de funciones secuenciales. Además, cuenta con un módulo que permite la conectividad con una red, llevar los datos a la nube y poder manejarlos correctamente según se disponga. El ESP32 es el módulo ideal para la implementación de capacidades IoT al PLC por la integración de Wifi y Bluetooth que este posee. Las capacidades IoT en un PLC recoge datos de sensores, los procesa y toma decisiones. Conectado a Internet, transmite datos a una plataforma IoT en la nube, permitiendo monitoreo y control remoto, análisis en tiempo real, y mantenimiento predictivo. Mejora la eficiencia operativa y optimiza procesos industriales.

Este diseño de PLC con capacidades IoT es un aporte tecnológico importante para los usuarios ya que esto integra recolección y procesamiento de datos con conectividad a Internet, permitiendo el monitoreo, control remoto y el análisis en tiempo real.

2.1. SITUACIÓN PROBLEMÁTICA

Actualmente se utilizan PLCs convencionales que están limitados en su capacidad de conectividad y funcionalidad IoT. Además, el uso de estos representa altos costos en software y hardware en el que se incluye su configuración y mantenimiento.

En el ámbito educativo, la falta de PLCs en los laboratorios de la universidad limita la experiencia práctica de los estudiantes. La escasez de recursos financieros y el soporte específico para estas plataformas representan un desafío, impactando negativamente la calidad del aprendizaje en el área de automatización.

Frente a estas dificultades, se busca una solución que permita desarrollar un controlador lógico programable que cumpla con los estándares de la industria, pero a un costo accesible, y que además cuente con capacidades IoT. Sin embargo, con esta propuesta surge la tarea de encontrar las herramientas adecuadas, así como la necesidad de capacitar a las personas en el uso de estas tecnologías. Además, deben asegurarse de que la solución propuesta sea confiable, segura y cumpla con los estándares.

2.2. FORMULACIÓN DEL PROBLEMA

La falta de controladores lógicos programables de bajo costo con capacidades IoT, limita el desarrollo de aplicaciones, especialmente en ambientes experimentales de laboratorio orientados tanto a la industria como a la educación, impidiendo mejoras en eficiencia y prácticas educativas de calidad.

2.3. OBJETO Y CAMPO DE ACCIÓN

2.3.1. Objeto de investigación:

Desarrollo del PLC con capacidades IoT.

2.3.2. Campo de acción:

Desarrollo del PLC con capacidades IoT

2.4. BENEFICIARIOS

2.4.1. Directo

- Usuarios del laboratorio PLC y automatización de la facultad CIYA

2.4.2. Indirecto

- Comunidad Científica

2.5. JUSTIFICACIÓN

El desarrollo tecnológico globalizado permite el acceso al conocimiento, permitiendo que áreas como la electricidad, que antes dependían de recursos limitados a universidades o centros de investigación especializados, ahora puedan aprovechar el conocimiento colaborativo para resolver problemáticas locales. En este contexto, la carrera de Electricidad juega un papel clave, ya que forma profesionales capaces de diseñar, implementar y mantener sistemas eléctricos y de automatización, fundamentales para el avance industrial y tecnológico.

El proyecto de investigación propuesto, enfocado en desarrollar un controlador lógico programable de bajo costo para aplicaciones específicas, está alineado directamente con los objetivos de la carrera de Electricidad. Este tipo de dispositivos son esenciales en la automatización de procesos industriales, ayudando a optimizar la productividad de las empresas locales y a reducir costos operativos. Al crear un PLC accesible y adaptable a las necesidades locales, no solo se fomenta la innovación tecnológica, sino que también se fortalece la capacidad de los profesionales de electricidad para ofrecer soluciones técnicas eficientes y sostenibles.

Además, el uso de tecnologías libres, tanto en software como en hardware, es un aspecto crucial en este proyecto. Estas tecnologías permiten a los profesionales de electricidad y a las empresas locales independizarse de licencias propietarias, reduciendo costos y promoviendo la adaptabilidad de los sistemas a diversas aplicaciones. Este enfoque no solo favorece la libertad de uso y modificación, sino que también estimula la colaboración entre desarrolladores, resultando en soluciones más robustas y accesibles para la industria.

Al concluir este proyecto, se tendrá una visión clara sobre la viabilidad de producir controladores lógicos programables para el mercado local, ofreciendo soluciones personalizadas con garantía y servicio técnico cercano. Esto beneficiará a las empresas locales y, al mismo tiempo, sentará las bases para futuras investigaciones en el campo de la electricidad y la automatización. Los profesionales de esta área podrán usar los resultados de este trabajo para mejorar sus diseños, optimizar procesos y contribuir al crecimiento de la industria tanto local como nacional.

2.6. OBJETIVOS

2.6.1. General

Desarrollar un controlador lógico programable con capacidades IoT, utilizando hardware de bajo costo y software libre para aplicaciones de automatización.

2.6.2. Específicos

- Investigar toda la documentación relacionada a PLCs, sistemas IoT y el uso de hardware y software libre en la automatización.
- Seleccionar los componentes adecuados para el desarrollo del PLC con capacidades IoT.
- Diseñar el diagrama del prototipo y su configuración para la implementación del hardware y software.
- Validar el funcionamiento del PLC en diferentes aplicaciones.

2.6.3. Sistemas de tareas

Tabla 1. Sistema de tareas utilizadas en el desarrollo del documento.

Objetivo	Actividad	Resultado de la actividad	Técnica e instrumentos
Investigar toda la documentación relacionada a PLCs, sistemas IoT y el uso de hardware y software libre en la automatización.	Recopilación de información sobre estándares de PLC, tecnologías IoT y herramientas de hardware/software libre.	Identificación de las tecnologías disponibles y recomendaciones preliminares.	Revisión bibliográfica, bases de datos académicas, foros especializados, consulta de manuales técnicos.
Seleccionar los componentes adecuados para el desarrollo del PLC con capacidades IoT.	Evaluación diferentes opciones de hardware (microcontroladores, sensores, actuadores y software, lenguajes).	Lista de componentes seleccionados, justificando su elección con criterios técnicos y económicos.	Análisis comparativo, consultas con proveedores, simulaciones preliminares.
Diseñar el diagrama del prototipo y su configuración para la implementación del hardware y software.	Desarrollo de diagramas eléctricos y de flujo del sistema. Documentar configuraciones de hardware y software.	Diseño esquemático del prototipo y manual de configuración.	Herramientas de diseño como AutoCAD y EasyEDA para su respectivo diseño.
Realizar pruebas de funcionamiento y análisis de resultados del sistema.	Implementación del prototipo, ejecutar pruebas de conectividad, funcionalidad y rendimiento. Analizar datos obtenidos.	Identificación de resultados de pruebas, análisis de desempeño y recomendaciones para mejoras.	Equipos de medición (multímetros), herramientas de análisis de datos como Excel.

3. FUNDAMENTACIÓN TEÓRICA

En este capítulo se presentan los conceptos básicos del proyecto, explorando aspectos esenciales como los Softwares libres, los distintos sistemas de control, el IoT, los PLCs y el entorno de programación, entre otros temas relevantes.

3.1. ANTECEDENTES

En la Expo Ibero Otoño de la Universidad Iberoamericana Puebla propuso el desarrollo de un PLC de bajo costo aplicando innovación frugal, con la finalidad de ofrecer una solución accesible para sectores con menor capacidad adquisitiva. Este prototipo de PLC buscó cumplir con los estándares industriales (Normas IP ANSI), asegurando resistencia, empotramiento e impermeabilidad, a la vez que implementaba tecnologías como Raspberry Pi 3 y microcontroladores ATMEGA 2560 para garantizar flexibilidad y compatibilidad con diversos entornos [2].

Este proyecto establece un importante precedente en la creación de PLCs accesibles, evidenciando la factibilidad de desarrollar soluciones económicas que no sacrifican las funciones esenciales necesarias para los sistemas de automatización. En este contexto, la investigación actual tiene como objetivo profundizar en estos conceptos al investigar nuevas tecnologías y perfeccionar la implementación de PLCs en una variedad de entornos, a compartir desde el ámbito industrial hasta el sector residencial.

En [3] Jonathan Westhues inició el proyecto LDMICRO para enseñar cómo programar los microcontroladores PIC y AVR utilizando el lenguaje Ladder. Este método se utiliza ampliamente en la automatización industrial y los controladores lógicos programables. El objetivo principal de LDMICRO es facilitar el procedimiento de codificación para los maquinistas y técnicos conocidos con los PLC, permitiéndoles así transferir su experiencia en este entorno alternativo. LDMICRO proporciona a los usuarios la facultad de crear programas que emplean razonamiento paso a paso, luego crear código que se puede procesar de acuerdo con las capacidades del microcontrolador. El software muestra un diseño visual básico y fácil de entender, lo que hace que sea fácil de hacer y cambiar rápidamente los programas informáticos. También incluye herramientas para imitar cómo el programa funciona para que podamos comprobarlo antes de realmente usarlo en las máquinas reales. Esto ayuda a identificar y corregir errores dentro del programa lógico.

Aunque el desarrollo del proyecto de Jonathan Westhues se desvanece, LDMICRO mantiene un conjunto dinámico de personas involucradas en activos compartidos y estrategias dentro de reuniones virtuales. Micro Company desarrolla tecnología que es

útil para las personas involucradas en la programación de microcontroladores para aplicaciones automatizadas e industriales.

3.2. DEFINICIÓN DE UN CONTROLADOR LÓGICO PROGRAMABLE

Un PLC es un dispositivo electrónico programable a través de un lenguaje que controla en tiempo real varios tipos de operación secuencial relacionada con el contexto industrial. A medida que el programador señala el comportamiento requerido el controlador funciona según las instrucciones contenidas dentro de un programa guardado en su memoria interna [4]. Las funciones de un PLCs ejecutan a través de las entradas del dispositivo que de acuerdo con la lógica de programación envían señales hacia la salida las cuales serán usadas para cumplir las órdenes de control desarrolladas por usuario [5].

3.3. PLC EN EL CAMPO DE AUTOMATIZACIÓN

Los PLC son utilizados en diversas aplicaciones incluyendo plantas de energía, redes eléctricas inteligentes, unidades de fabricación y el monitoreo de herramientas o equipos para maquinaria. Además, desempeñan un papel fundamental en el diseño y control de procesos [6]:

Automatización de procesos industriales: en sectores como la manufactura, química, alimentaria y farmacéutica, los PLC se encargan del control y supervisión de diversos procesos que abarcan desde la producción hasta el llenado, empaquetado y mezcla de ingredientes; también regulan parámetros críticos como temperatura y presión.

Control de maquinaria: estos sistemas son esenciales para gestionar maquinarias dentro del entorno industrial. Esto incluye robots industriales, cintas transportadoras e impresoras entre otros dispositivos. Su implementación permite realizar tareas repetitivas que pueden ser peligrosas con mayor seguridad y eficiencia.

Automatización de edificios: dentro del ámbito edificatorio se utilizan los PLC en sistemas relacionados con climatización, iluminación o controles de acceso así como medidas de seguridad. Facilitan un manejo centralizado junto a una programación inteligente para estos aparatos.

Control de energía: se aplican en sistemas diseñados para optimizar la distribución eléctrica al monitorizar su consumo energético; esto contribuye a mejorar la eficacia energética además coordinar las operaciones en instalaciones renovables.

Automatización de procesos en la industria petrolera y del gas: durante actividades relacionadas con exploración o refinación tanto del petróleo como gas natural los PLC

gestionan múltiples procedimientos tales como perforaciones u operaciones sobre flujo además asegurando estándares operativos seguros.

Automatización de sistemas de transporte: son empleados ampliamente dentro de oficinas logísticas vinculados a trenes urbanos o aeropuertos donde permiten controlar señalizaciones vehiculares así manejar tráfico eficiente sumándose asimismo automatismos asociados al movimiento de carga.

3.4. AUTOMATIZACIÓN CON LÓGICA CABLEADA

La lógica de cableado, también conocida como control cableado o lógica cableada, es un sistema de control en el que las funciones lógicas se implementan mediante la conexión física de componentes electrónicos, como relés, interruptores, temporizadores y otros dispositivos. Este enfoque fue ampliamente utilizado en la automatización industrial antes de la llegada de los controladores lógicos programables. A continuación, se presentan las ventajas y desventajas de la lógica de cableado [7]:

3.4.1. Ventajas

- Ideal para aplicaciones donde el tiempo de respuesta es crítico.
- Resistente a interferencias electromagnéticas y condiciones adversas
- Económico para aplicaciones simples con pocas entradas y salidas.
- Fácil de implementar para personal con conocimientos de electricidad, pero sin experiencia en software.
- No hay riesgo de fallos por interrupciones o fallas de programas.

3.4.2. Desventajas

- Cualquier cambio en la lógica requiere modificar el cableado físico, lo que es costoso.
- Dificultad para diagnosticar fallos.
- Limitado para sistemas complejos o en crecimiento.
- Mayor costo a largo plazo.
- Vida útil limitada de componentes mecánicos.

3.5. VENTAJAS DE AUTOMATIZACIÓN CON PLC

Dentro de la automatización con los PLCs presentan una gran variedad de ventajas que ofrecen a los procesos que requieren automatización en la industria [8]:

- Están diseñados y contruidos específicamente para su uso en entornos industriales.
- Son equipos flexibles debido a su carácter programable.
- Son fáciles de instalar y reutilizar.
- Su construcción facilita el mantenimiento y la localización de fallas.
- Pueden ser utilizados en diversos tipos de tareas de control dentro de una misma planta, lo que facilita el aprendizaje y optimiza el aprovechamiento de sus funciones.
- Su capacidad de comunicación permite integrarse eficazmente en un sistema global de control o en un sistema de producción integrado.

3.6. DESVENTAJAS DE AUTOMATIZACIÓN CON PLC

Las desventajas iniciales de tener que manejar dos o tres sistemas diferentes, posiblemente con lenguajes y equipos de programación propios, están desapareciendo debido a la creciente similitud de los equipos actuales, la posibilidad de programación a través de ordenadores personales, la disponibilidad de módulos de interconexión entre unidades de distintos fabricantes, y la interconectividad de Redes Locales impulsada por los esfuerzos de normalización en el ámbito de las Redes de Comunicación [8].

La evolución hacia estándares más universales y compatibles, junto con las mejoras en las herramientas de desarrollo y la interconexión de equipos, está reduciendo la complejidad y las desventajas de tener que dominar múltiples sistemas distintos en el campo de la tecnología y la informática.

Además, las desventajas dentro de la automatización con PLCs se muestran las principales que se presenta en la instalación [8]:

- Para procesos pequeños el costo es muy elevado
- Remplazo total del PLC ante fallas en el mismo.
- Personal capacitado para la programación y mantenimiento
- Su uso no es viable en procesos con tiempos de respuesta cortos o en señales con transiciones entre niveles lógicos que ocurren a altas frecuencias.

3.7. COMPARACIÓN LÓGICA CABLEADA VS PLC

La lógica cableada y los Controladores Lógicos Programables (PLC), destacando sus diferencias en términos de funcionamiento, aplicaciones, costos, flexibilidad y mantenimiento mostradas en la Tabla 2.

Tabla 2. Comparación de la funcionabilidad.

CARACTERÍSTICA	LÓGICA CABLEADA	PLC
Flexibilidad	Baja. Cambios requieren modificar el cableado físico.	Alta. Cambios se realizan en el software sin alterar el cableado.
Complejidad de Lógica	Limitada a funciones simples.	Maneja lógicas complejas (cálculos, control PID, etc.).
Tiempo de Respuesta	Respuesta inmediata y determinista.	Depende del ciclo de escaneo (pequeño retardo).
Costo Inicial	Bajo para sistemas pequeños.	Alto (incluye hardware y software).
Costo a Largo Plazo	Alto (mantenimiento y cambios costosos).	Bajo (modificaciones y mantenimiento económicos).
Espacio Físico	Ocupa más espacio (relés, temporizadores, etc.).	Compacto (múltiples funciones en un solo dispositivo).
Mantenimiento	Complicado y costoso (dificultad para diagnosticar fallos).	Sencillo y económico (diagnóstico mediante software).
Escalabilidad	Difícil de escalar (aumento exponencial de conexiones).	Fácil de escalar (agregar módulos de E/S y actualizar software).
Integración	Limitada (no se integra con sistemas modernos).	Alta integración (SCADA, redes industriales, IoT).
Aplicaciones Típicas	Sistemas simples y estáticos (luces, motores básicos).	Sistemas complejos y dinámicos (líneas de producción, robótica).
Robustez	Alta en ambientes extremos (resistente a interferencias).	Depende del modelo (algunos PLC son robustos).
Vida Útil	Componentes mecánicos (relés) se desgastan con el tiempo.	Componentes electrónicos más duraderos.
Programación	No requiere programación (solo conexiones físicas).	Requiere programación (lenguajes como Ladder, bloques funcionales, etc.).
Diagnóstico de Fallos	Difícil (revisión manual de conexiones).	Fácil (herramientas de diagnóstico integradas).

3.8. ESTRUCTURA BÁSICA DEL PLC

En la estructura básica del PLC se muestran los principales elementos en la Figura 1 para su funcionamiento, que se describen a continuación:

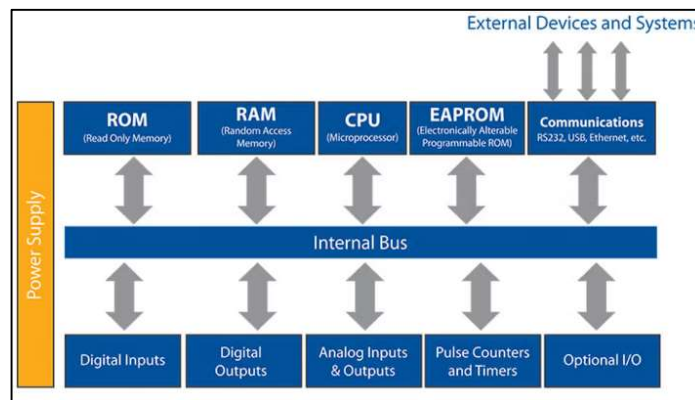


Figura 1. Estructura básica del PLC [5].

3.8.1. La unidad central de proceso

La Unidad Central de Proceso de un autómata consta principalmente de dos elementos: el procesador y la memoria [9]. Su diseño puede variar, desde un solo módulo que incluye ambos componentes, hasta una configuración con un módulo para el procesador y otro para la memoria, o incluso en formato rack o armario. En ciertos modelos, la unidad también integra la fuente de alimentación, y en los mini autómatas, generalmente incluye también parte del sistema de E/S [10].

La Unidad Central de Proceso ejerce el control de todas las tareas del autómata, tanto en lo que se refiere a la toma de datos y control del proceso, como a las funciones internas.

3.8.2. Memorias

Un PLC tiene dos tipos de memorias:

3.8.2.1. Memoria del programa

Es el componente encargado de almacenar el código del programa que define la lógica de control del PLC. Su estructura se basa en un bloque principal denominado *Main*, al que se pueden incorporar bloques adicionales, como subrutinas, funciones y bloques de datos, permitiendo una organización modular y eficiente del sistema [11].

3.8.2.2. Memoria de datos

Es el área de memoria destinada al almacenamiento de datos resultantes de la lectura de entradas, escritura en salidas y procesamiento de operaciones lógicas o aritméticas dentro del programa del PLC. Admite diferentes tipos de datos, como *bits*, *bytes* (8 bits), *words* (16 bits), *double words* (32 bits), valores en punto flotante (real), temporizadores y contadores. En generaciones anteriores, la memoria se organizaba en segmentos específicos según el tipo de dato de cada bloque. Hoy en día, la arquitectura de memoria es de propósito general, lo que permite una asignación más flexible y eficiente de los datos compatibles con el PLC [9].

3.8.3. Entradas

Transmiten al PLC el estado de dispositivos de medición, pulsadores, interruptores, sensores o cualquier elemento capaz de proporcionar información del entorno lo cual requieren una conexión física con el dispositivo a monitorear y pueden ser de tipo digital o analógico, según su resolución y aplicación. Las entradas analógicas están diseñadas para la conexión de sensores como termocuplas, celdas de carga, galgas extensiométricas y potenciómetros, entre otros. Además, incorporan protecciones opto eléctricas internas para garantizar el aislamiento y la protección del sistema [5].

3.8.4. Salidas

Se vinculan y se conectan con los actuadores, que están destinados a controlar o alterar el estado de una variable a regular, como motores, niquelinas, lámparas, luces piloto, electroválvulas, entre otros. Al igual que las entradas, pueden ser tanto digitales como analógicas. Internamente, el sistema utiliza un ULN2803, el cual incorpora diodos de

protección contra tensiones inversas, esenciales para proteger el equipo al manejar cargas inductivas como relés o motores [5].

3.8.5. Fuentes de alimentación

Para la fuente de alimentación es necesario un dispositivo que suministre energía a una carga o grupo de cargas eléctricas. Su propósito principal es convertir la corriente alterna (CA) en una corriente continua (CC) estable y adecuada para alimentar los componentes electrónicos. Las fuentes de alimentación son esenciales en una amplia gama de aplicaciones, desde dispositivos electrónicos pequeños hasta grandes sistemas industriales [9].

En la automatización de procesos industriales para que un PLC funcione correctamente, requiere una fuente de alimentación adecuada que suministre la energía necesaria para sus operaciones.

3.8.6. Puertos de comunicación

Hoy en día, la mayoría de los PLCs incluyen múltiples puertos de comunicación que facilitan la interacción con los entornos de programación, que generalmente están integrados en el propio PLC. El puerto Ethernet se ha convertido en el estándar principal, reemplazando poco a poco las interfaces seriales RS-232 y RS-485, que eran comunes en las generaciones anteriores de estos dispositivos [12].

3.9. TIPOS DE SEÑALES

El PLC es capaz de recibir y emitir dos tipos de señales hacia y desde el entorno como señales digitales binarias, representadas por estados 1 o 0 y señales analógicas dentro de un rango de voltaje (0 a 10 V) o corriente (4 a 20 mA). La forma en que estas señales son procesadas depende tanto del tipo de actuador o dispositivo a controlar como del módulo específico utilizado lo cual es fundamental considerar que cada módulo requiere una conexión a la fuente de alimentación y cuenta con un sistema interno de aislamiento galvánico, acondicionamiento, filtrado y protección. Este diseño previene interferencias por ruido, conexiones incorrectas o señales inadecuadas que podrían afectar las entradas, salidas o el CPU del PLC [11].

En cuanto a las entradas pueden ser:

- **Tipo voltaje continuo PNP o fuente**, el terminal común se conecta al polo negativo de la fuente de alimentación (0 VCC), mientras que el polo positivo (24 VCC) es suministrado a través del dispositivo externo conectado, como un

interruptor, sensor o pulsador, hasta el terminal de entrada del módulo. Aunque 24 VCC es el voltaje más utilizado en las entradas, también existen configuraciones que operan con voltajes de 12 VCC o 50 VCC [11]. Ver en la Figura 2.

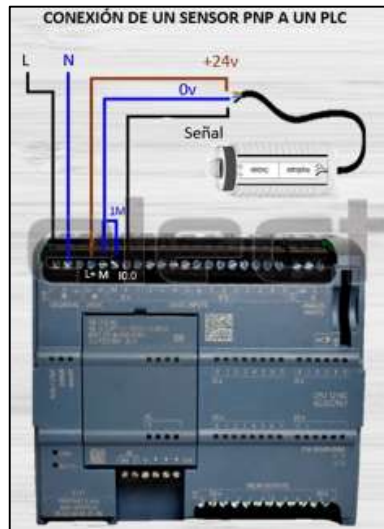


Figura 2. Conexión de un sensor PNP [11].

- **Tipo voltaje continuo NPN o sumidero**, el terminal común se vincula al polo positivo de la fuente de alimentación (24 VCC), mientras que el polo negativo (0 VCC) es conducido a través del dispositivo externo conectado, como un interruptor, sensor o pulsador, hasta el terminal de entrada del módulo, asegurando la correcta generación de la diferencia de potencial necesaria para su funcionamiento [11]. Ver en la Figura 3.

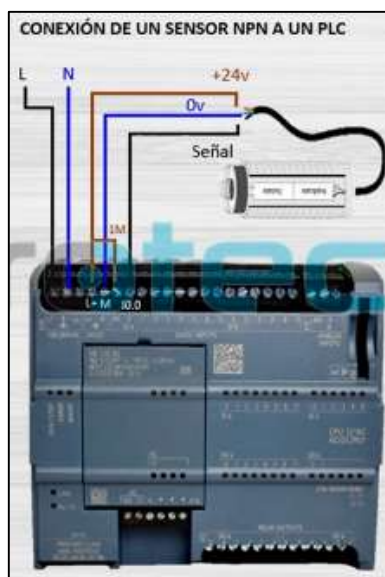


Figura 3. Conexión de un sensor NPN [11].

- **Tipo corriente (4 o 20 mA):** el terminal común del módulo se conecta al polo negativo de la fuente de alimentación (0 VCC), mientras que la corriente de entrada al módulo varía entre 4 mA para representar un estado lógico 0 y 20 mA para un estado lógico 1. Este tipo de conexión es ideal para aplicaciones con distancias relativamente largas entre el módulo y el dispositivo de entrada. Además, permite la detección de fallos en la conexión, ya que una señal de 0 mA puede indicar una desconexión o un problema en el circuito [11]. Ver en la Figura 4.

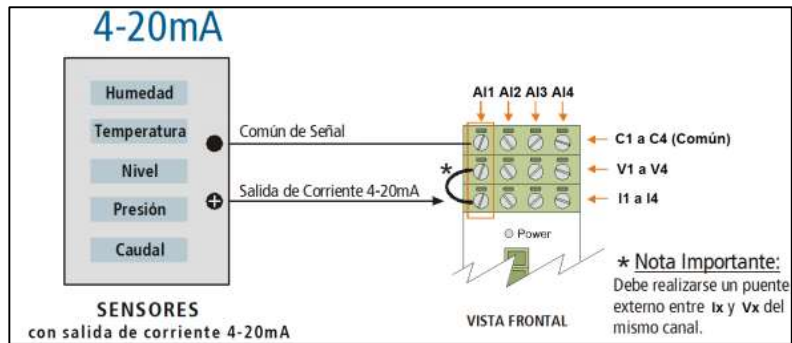


Figura 4. Entrada de corriente [11].

- **Tipo voltaje alterno:** el terminal común del módulo se conecta al neutro de la fuente de alimentación, mientras que la fase es suministrada a través del dispositivo externo conectado, como un interruptor, sensor o pulsador, hasta el terminal de entrada del módulo. Este tipo de conexión está disponible en distintos niveles de voltaje, incluyendo 24, 48, 110 o 220 VAC.[11]. Ver en la Figura 5.

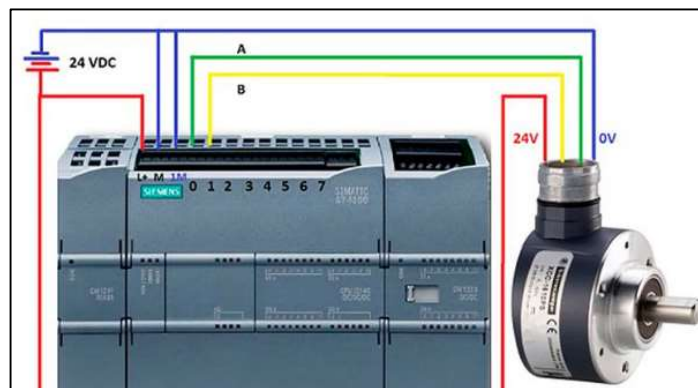


Figura 5. Salida tipo voltaje [11].

Además, existen módulos de entradas especiales, como los contadores rápidos, que se utilizan para sensores que emiten señales de alta frecuencia, como los *encoders*.

Las entradas analógicas pueden operar con señales de voltaje en rangos de 0 a 10 V o -10 a 10 V, así como con señales de corriente de 4 a 20 mA. Estas entradas convierten

internamente las señales analógicas en valores digitales mediante conversores analógico-digitales (ADC). Una característica clave de este tipo de entradas es su resolución, que puede ser de 10, 12 o 16 bits; a mayor resolución, mayor precisión en la medición, ya que permite detectar cambios más pequeños en la señal. Generalmente, el valor procesado se entrega en un *Word* como un número entero o flotante. Además, existen módulos de entradas analógicas especializados diseñados para la conexión de sensores específicos, proporcionando el acondicionamiento, filtrado y transmisión de señales necesarios, lo que elimina la necesidad de procesamiento adicional. Ejemplos de estos módulos incluyen aquellos utilizados para termocuplas o celdas de carga [11].

En las entradas digitales, generalmente es necesario conectar el cable que transporta la señal negativa (0 V) al módulo donde se ubican las entradas, ya que estas reciben el polo positivo y requieren una diferencia de potencial interna para su activación, siempre que se utilice lógica positiva.

En cuanto a las salidas, estas pueden ser [11]:

- **Tipo voltaje continuo PNP colector abierto:** el terminal común del módulo se conecta al polo negativo de la fuente de alimentación (0 VDC), mientras que el terminal positivo (24 VDC) suministra voltaje al dispositivo externo conectado, como una luz piloto, motor, electroválvula, relé u otro actuador, a través del terminal de salida del módulo [11]. Ver en la Figura 6.

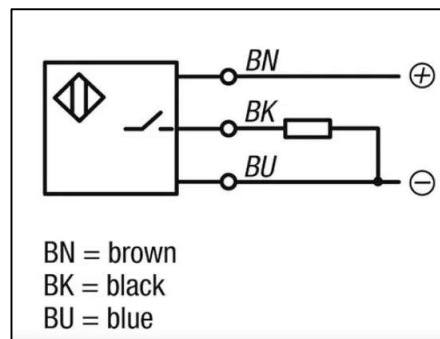


Figura 6. Diagrama de conexión de un sensor inductivo [11].

- **Tipo voltaje continuo NPN colector abierto:** el terminal común del módulo se conecta al polo positivo de la fuente de alimentación (24 VDC), mientras que el terminal negativo (0 VDC) es enviado al dispositivo externo conectado, como una luz piloto, motor, electroválvula, relé u otro tipo de salida, a través del terminal de salida del módulo [11]. Ver en la Figura 7

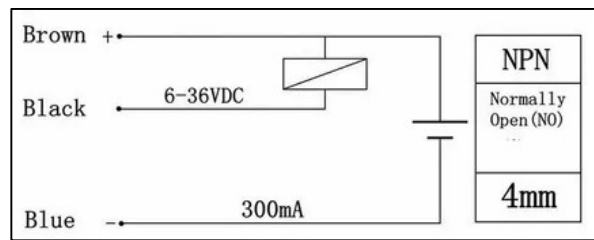


Figura 7. Diagrama de conexión de un sensor NPN [11].

- Tipo corriente (4 o 20 mA):** el terminal común del módulo se vincula al polo negativo de la fuente de alimentación (0 VDC), mientras que el terminal de salida del módulo suministra la corriente correspondiente, con 4 mA para representar un estado lógico 0 y 20 mA para un estado lógico 1. Este método es ideal para aplicaciones en las que el módulo y el dispositivo de salida están separados por distancias relativamente largas. Además, permite detectar posibles fallos en la conexión, ya que una señal de 0 mA en el dispositivo puede indicar una desconexión o un problema en el circuito [11].

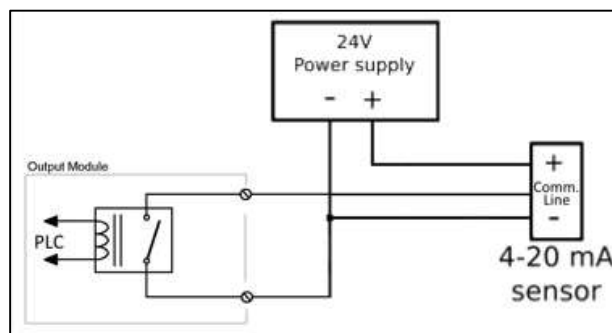


Figura 8. Diagrama de conexión tipo corriente [11].

- Tipo voltaje alterno TRIAC:** el terminal común del módulo se conecta al neutro de la fuente de alimentación, mientras que la fase es entregada al dispositivo externo, como un motor, relé o lámpara, a través del terminal de salida del módulo [11]. Ver en la Figura 9.
- Tipo relé:** es uno de los tipos de salida más utilizados, ya que puede operar con señales tanto de corriente alterna (AC) como de corriente continua (DC). En esta configuración, el terminal común del módulo se conecta a un terminal de la fuente de alimentación, mientras que el terminal de salida se encarga de cerrar el circuito. Generalmente, dispone de un solo contacto normalmente abierto, aunque algunos PLCs incluyen tanto un contacto normalmente abierto como uno cerrado. Sus principales desventajas son el tiempo de respuesta, ya que al tratarse de un componente electromecánico, la conmutación es más

lenta en comparación con salidas NPN, PNP o TRIAC, y el desgaste de los contactos, lo que puede afectar su vida útil [11]. Ver en la Figura 9.

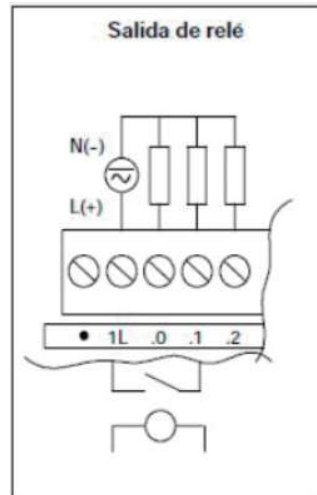


Figura 9. Diagrama de conexión de un relé [11].

Además, hay módulos de salidas especiales, como la modulación por ancho de pulsos (PWM), que se emplean para controlar elementos como motores.

Las salidas analógicas pueden operar con señales de voltaje en rangos de 0 a 10 V o -10 a 10 V, así como con señales de corriente de 4 a 20 mA. Estas salidas convierten internamente los valores digitales de sus registros en señales analógicas mediante convertidores digital-analógico (DAC). Una característica clave de este tipo de salidas es su resolución, que puede ser de 10, 12 o 16 bits, lo que influye directamente en la precisión y calidad de la señal generada [11].

3.10. TIPOS DE PLC

Los PLCs se clasifican de la siguiente forma [13]:

- **PLC nano:** es un dispositivo que es notable principalmente por ser increíblemente pequeño y en comparación con otros PLC, de capacidad limitada. Un dispositivo así se implementa en tareas simples de automatización donde la automatización y el monitoreo solo necesitan las funciones básicas. Por lo general, este tipo de PLC se implementa en aplicaciones, en las que se necesita un mínimo de E / S y posiciones de memoria. Ver en la Figura 10.
- **PLC compacto:** este tipo de PLC también se implementa con la máxima integración: la CPU, las E / S y la fuente de alimentación se combinan en una unidad. Como regla general, este controlador se utiliza en ámbito de

aplicación cuando el espacio está limitado y la solución de control tiene que ser compacta y fácil de instalar. Ver Figura 10.

- **PLC modular:** se implementa como una combinación de diferentes módulos, cada uno de ellos realizando su función específica. Los módulos en este PLC se conectan al chasis o base promoviendo una gran flexibilidad y compatibilidad en la elaboración e implementación de soluciones de control. Ver Figura 10.



Figura 10. Tipos de PLC [9].

3.10.1. Clasificación PLC

La clasificación de estos dispositivos de diferentes fabricantes se fundamenta en la capacidad de procesamiento, el número de entradas o entradas y salidas E/S, las funciones disponibles, la posibilidad de la ampliación y el costo como se muestra en la Figura 11.

- **Gama baja:** se ven definidos por su bajísima capacidad de memoria en datos, programa y entradas o E/S. A veces no se les considera PLC en toda la regla de la recepción del término y es más correcto hablar de relés inteligentes. Suelen ser dispositivos compactos o semi compactos; un ejemplo de esta categoría es la serie Logo de Siemens [14].
- **Gama media:** una excelente alternativa para todo tipo de aplicaciones de tamaño medio en industria y comercio. Representan un equilibrio perfecto entre el costo y la funcionalidad y permiten la ejecución de proyectos complejos en las dimensiones óptimas, la ampliación y las funciones más avanzadas para mejorar el control y la supervisión de los procesos de trabajo. La comunicación y la flexibilidad para adaptarse a todo tipo de necesidades hacen que resulten perfectos para la automatización de sistemas industriales y de comercio hasta la gestión de sistemas de climatización, máquinas y mucho más [14].

- **Gama alta:** dispositivos diseñados para aplicaciones críticas y complejas en la industria. Superior en sus prestaciones y tamaños, superior son su capacidad de control de la comunicación y la seguridad son esenciales para la automatización de cualquier planta industrial, sistemas de control distribuidos y el funcionamiento de aplicaciones críticas [14].



Figura 11. Clasificación de los PLCs [5].

3.11. LENGUAJE DE PROGRAMACIÓN PLC

Los fabricantes de Controladores Lógicos Programables proporcionan entornos de selección de programación para proyectos IEC 61131-3. Para ello, Siemens proporciona TIA Portal y Rockwell Automation ofrece Studio 5000, entre otras. Estos entornos accedidos por el programador le permiten seleccionar diferentes lenguajes que incluyen el lenguaje de diagrama de escalera, LAD; el lenguaje de diagrama de bloques de función, FBD; el lenguaje de Lenguaje Estructurado, SLC; y GRAFCET, dependiendo del PLC..modelo específico y serie [14].

Las características comunes de estos entornos de programación incluyen la capacidad de crear proyectos, agregar módulos de expansión, diagnosticar el sistema y los procesos, simular programas, configurar recursos de comunicación, y documentar los proyectos de automatización. Además, estos entornos permiten la conexión con otros dispositivos como PLCs y HMIs, lo que facilita la integración y el control de sistemas complejos de automatización industrial [11].

Dentro del funcionamiento del lenguaje programación existen:

- **Diagrama de escalera (*Ladder LD*):** es un lenguaje de programación gráfico estandarizado según la norma IEC 61131-3, utilizado en la programación de PLCs. Emplea una estructura basada en dos barras verticales que representan los conductores de alimentación, entre las cuales se disponen "peldaños"

horizontales que contienen contactos (normalmente abiertos o cerrados) y bobinas, los cuales simbolizan entradas, salidas y operaciones lógicas [11].

- **El Diagrama de Bloques Funcionales (FBD):** es un lenguaje de programación gráfico muy utilizado en aplicaciones que involucran el flujo de información o datos entre componentes de control. En este lenguaje, las funciones y bloques funcionales se representan como circuitos integrados interconectados, lo que permite visualizar claramente la relación entre las entradas, salidas y operaciones lógicas. Es utilizado de manera extensa en Europa y presenta similitudes con la implementación de circuitos digitales que se fundamentan en compuertas lógicas y flip-flops. Esta característica lo convierte en una opción excelente para aplicaciones que exigen operaciones complejas relacionadas con el procesamiento de señales y el control secuencial [11].
- **Lenguaje de programación gráfica (GRAPH SFC):** este estándar es regulado por la IEEE 61131-3 se utiliza para la programación de PLC en tal contexto de uso donde este dispositivo está encargado de realizar control secuencial. Se basa en la representación gráfica de procesos divididos en pasos y transiciones. Estos se organizan en diagramas de flujo, que muestran las secuencias de acciones. Cada paso refleja el sistema y sus estados, mientras que las transiciones definen las condiciones seguidas de las cuales hay un cambio en el proceso. GRAPH SFC es extremadamente útil en aplicaciones que involucran varios pasos y deben ordenarse. Se aplica en la industria para sistemas y maquinaria automatizados complejos que funcionan de acuerdo con el principio de lotes, así como sistemas industriales y de producción complejos con múltiples etapas. Funciona de manera más lógica, lo que simplifica en gran medida la comprensión, programación y corrección de sistemas secuenciales y los hace fácilmente comprensibles para los especialistas [11].

3.12. ARDUINO IDE

El Arduino IDE es un entorno de desarrollo integrado diseñado para escribir, compilar y cargar código en placas Arduino [15]. Es fácil de usar y compatible con múltiples sistemas operativos como Windows, macOS y Linux. Facilita la programación en el lenguaje de Arduino (una variante de C++) y ofrece una interfaz intuitiva para la depuración y carga de código. El software está disponible para descarga gratuita en el sitio oficial de Arduino.

3.13. SOFTWARE OPENPLC

OpenPLC es un software ideal para programar el PLC, ya que consta con las funciones necesarias para realizar los programas de automatización y control [16]. Además, cumple con la normativa IEC 61131-3 para diseñar los proyectos de PLC en lógica de escalera, texto estructurado, lista de instrucciones, diagrama de bloques de funciones o diagrama de funciones secuenciales.

El software es compatible para varias plataformas como Windows, Linus y MacOS, se lo puede descargar de manera gratuita desde su página oficial y la instalación es como cualquier otro programa para el caso de Windows [16].

Para cargar el programa al PLC se necesita tener realizado un proyecto, conectar mediante USB con la PC, seleccionar el botón “*Transfer Program to PLC*”.

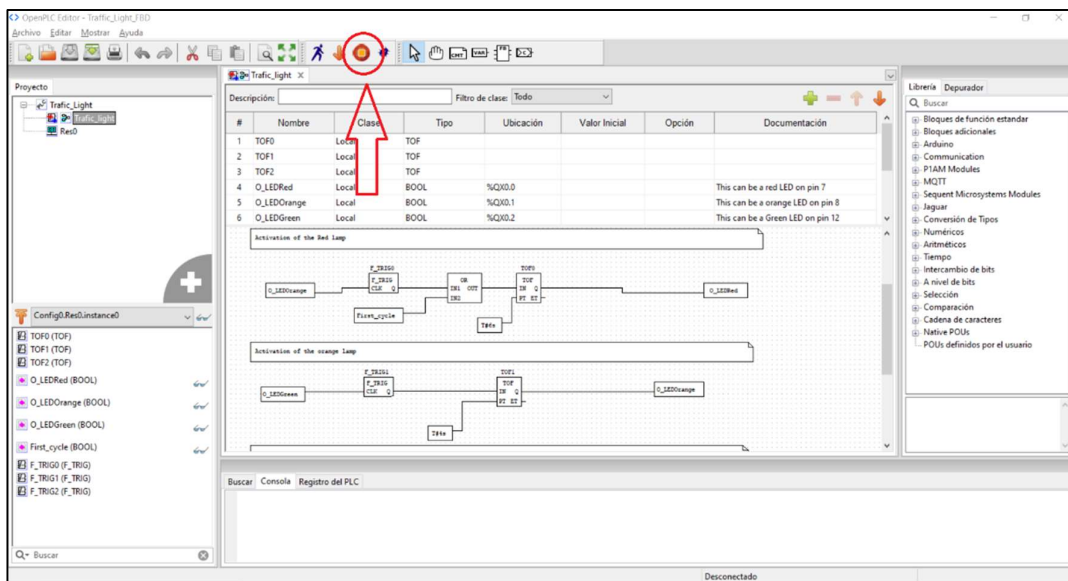


Figura 12. Interfaz del software OpenPLC.

Cuando aparezca la ventana, se elige el tipo de placa visto en la Figura 13, que en este caso sería el Arduino UNO, si es la primera vez que se utiliza el programa o una nueva placa, se debe esperar a que se instalen las librerías correspondientes.

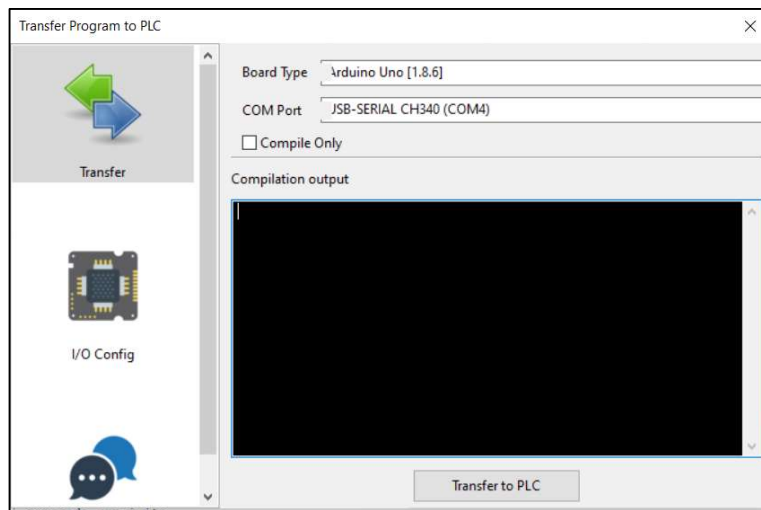


Figura 13. Ventana para cargar el programa en el PLC.

3.14. BASE DE DATOS FIREBASE

Firestore es una plataforma de desarrollo para aplicaciones web y móviles, creada por Google, que proporciona diversas herramientas para el almacenamiento de datos en tiempo real, autenticación de usuarios, análisis y otras funcionalidades. En este proyecto, Firestore se utiliza para almacenar y gestionar los datos de las entradas y salidas del PLC de manera remota. Con esto es posible monitorear y controlar el sistema desde cualquier lugar con acceso a internet [17].

Para la integración de Firestore con el ESP32, se debe configurar una base de datos en Firestore y utilizar las bibliotecas de Firestore para Arduino con el objetivo de hacer una conexión y envíos de datos desde el ESP32 a la base de datos en tiempo real más sencillos.

3.15. SOFTWARE EASYEDA

EasyEDA es una herramienta en línea que se puede utilizar para diseñar esquemas eléctricos y placas de circuito impreso (PCB). La aplicación ideal para diseñar los esquemas eléctricos y las PCB que se presentarán a continuación. Aparte de eso, EasyEDA es una herramienta de diseño palanca ya que permite a los usuarios simular circuitos y generar listas de materiales (BOM), al igual que hace posible fabricar las PCB utilizadas para la solución.

Para usar EasyEDA, necesitas una cuenta que puedas crear en el sitio web oficial de EasyEDA, habilitándote de todas las herramientas que se necesitan para redactar tus diagramas y simular tus circuitos. Además de eso, posee una amplia biblioteca con componentes alabables incluidos que se pueden usar en el proyecto.

3.16. INTERNET DE LAS COSAS (IOT)

Como todos sabemos, una de las tecnologías de más rápido crecimiento para automatizar funcionalidades es el Internet de las Cosas (IoT). El IoT se refiere a un objeto físico (el objeto físico que tiene incorporadas partes electrónicas para realizar sus capacidades) cuyas comunicaciones externas les dan la capacidad de interactuar, y la capacidad de las cosas individuales para detectar sus interacciones con otros objetos o el entorno en el que operan [12].

La interconexión de nuestros objetos cotidianos y su comunicación a través de la web es la base del IoT. Esto permite que los objetos físicos se conviertan en "inteligentes" mediante la integración de sensores, actuadores y otros dispositivos que aprovechan la inteligencia para retener y difundir información sin la necesidad de intervención humana. Descúbrelos conectándose a través de comunicaciones inalámbricas como WiFi y Bluetooth, o redes dedicadas de bajo consumo y largo alcance (LPWAN). Esta red permite que los objetos 'envíen' y 'reciban' pequeños fragmentos de información, creando un ecosistema en el que la toma de decisiones inteligentes puede ser automatizada y ajustada a procesos específicos [19].

La arquitectura del IoT se divide en cuatro capas. La capa de detección de objetos se encarga de identificar los objetos físicos y recoger los datos correspondientes. La capa de intercambio de información se ocupa de transmitir los datos de manera legible. La capa de integración de información se encarga de recombinar, limpiar y fusionar los datos inciertos obtenidos de las redes, transformándolos en conocimiento útil. Por último, la capa de servicios de aplicaciones proporciona servicios de contenido para diferentes usuarios [20].

3.17. APLICACIONES Y BENEFICIOS DEL IOT

El IoT no solo está cambiando todo a nuestro alrededor, ayudándonos a optimizar procesos, reducir costos, crear nuevas experiencias y lograr más eficiencia energética, mientras nos proporciona información sobre el entorno. Sus usos principales son los siguientes [20]:

- **Ciudades Inteligentes:** permite mejor utilización de los recursos urbanos como el tráfico, las luces públicas, el suministro de agua y la eliminación de residuos. La ciudad inteligente o entorno inteligente se refiere al uso de IoT y dispositivos interconectados para conservar energía y mejorar la calidad de vida de los ciudadanos.

- **Industria 4.0:** es la cuarta revolución industrial en la que se utiliza IoT para monitorear máquinas en tiempo real, predecir fallos continuos de equipos y mantener máquinas para tomar medidas preventivas, para aumentar la productividad en toda la industria e incluso reducir el tiempo de inactividad.
- **Agricultura de Precisión:** los sensores IoT inteligentes y asequibles envían datos instantáneos sobre las condiciones del suelo, el clima y los campos durante los registros, para empoderar a los agricultores a lograr más con menos: un modo de agricultura basado en la calidad, económico y sostenible.
- **Salud y Bienestar:** estos incluyen dispositivos portátiles y sistemas de monitoreo remoto que funcionan como chequeos externos y en tiempo real del estado del paciente, facilitando más opciones de tratamiento para los pacientes con un costo administrativo menor para el sistema de salud.

4. MÉTODOS Y PROCEDIMIENTOS

Este capítulo explica los métodos, procesos, materiales y condiciones experimentales que respaldan el diseño y la puesta en marcha de un PLC con capacidades de IoT, empleando software de código libre.

4.1. DESCRIPCIÓN DEL PROYECTO

Este proyecto está basado en PLC con IoT, utilizando Arduino Nano y ESP32. El PLC se adapta tanto a entornos universitarios como industriales, centrándose en tecnologías accesibles y de bajo costo.

OpenPLC se carga en el Arduino Nano, que es el controlador principal que procesa las entradas y salidas del sistema. La interfaz de señal y comunicación se obtiene mediante un conjunto de circuitos integrados. ESP32 se utiliza para controlar la conectividad IoT, que transfiere los datos a Firebase para verificar los valores y controlar el sistema de forma remota a través de internet.

4.2. DECLARACIÓN DE VARIABLES

- **Variable independiente**
Desarrollo del Controlador Lógico Programable con capacidades IoT.
- **Variable dependiente**
Funcionalidad del Hardware y Software.

4.3. ESQUEMA GENERAL DE FUNCIONAMIENTO

El PLC funciona con una alimentación principal de 24VDC, que puede provenir de una fuente o un adaptador del mismo voltaje con una capacidad de al menos 1A. Esta misma fuente también puede utilizarse para alimentar las entradas digitales, que operan a 24VDC.

Para suministrar energía a los componentes de baja tensión del sistema, se emplea un módulo regulador *step-down* DC-DC LM2596, el cual tiene una eficiencia cercana al 85% y proporciona una salida de 5VDC estable. Este voltaje es utilizado para alimentar los módulos ESP32 y Arduino Nano, donde el primero se encarga de la funcionalidad IoT y el segundo gestiona las entradas y salidas, de acuerdo con la programación en OpenPLC.

El ESP32 se conecta a una red WiFi para interactuar con la base de datos Firebase, facilitando la gestión de los datos requeridos. Por su parte, el Arduino Nano controla relés para las salidas y utiliza arreglos de resistencias para la lectura de entradas digitales y analógicas. Además, se emplea un amplificador operacional LM358 para generar una salida analógica de 0-10VDC.

Finalmente, se requiere una computadora para desarrollar los programas mediante el software OpenPLC, el cual proporciona la interfaz de programación. La carga del código al Arduino Nano se realiza a través de USB, convirtiéndolo en el controlador principal del PLC.

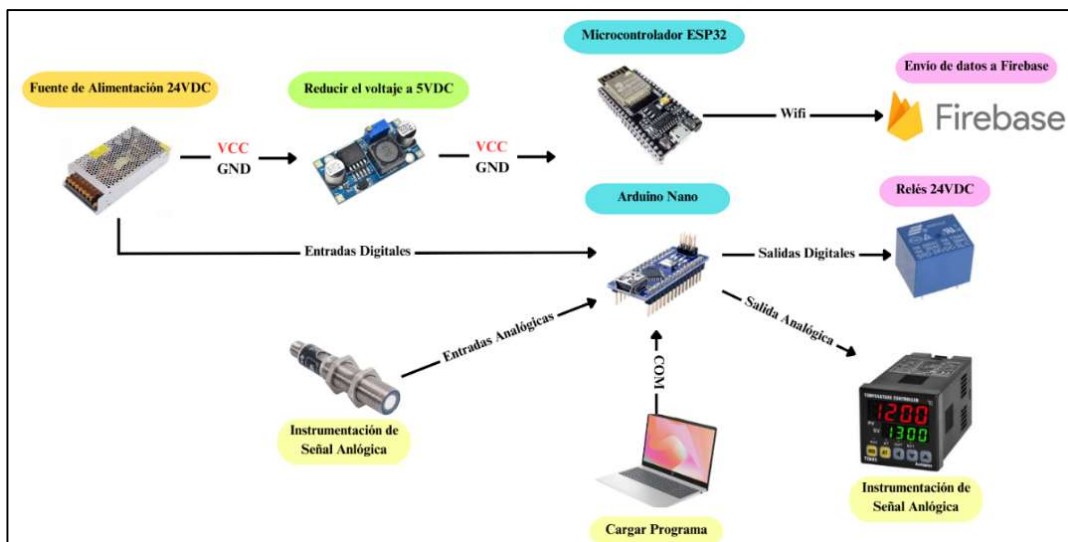


Figura 14. Esquema general de funcionamiento.

4.4. DETALLE DE COMPONENTES DEL SISTEMA

En este apartado se describirán las principales características de los dispositivos utilizados en el desarrollo del PLC con capacidades IoT, con el objetivo de seleccionar los dispositivos más adecuados para el proyecto.

4.4.1. Microcontrolador Arduino Nano

El microcontrolador será el cerebro del PLC a desarrollar, a este estarán conectadas las entradas y salidas, tanto digitales como analógicas, aquí se encontrará cargado el código que será previamente realizado en el software OpenPLC. Cuenta con un procesador Atmega328p con una velocidad de reloj de 16MHz lo cual resulta ideal para realizar los procesos que conlleva un PLC, su voltaje de alimentación es de 5VDC por lo que es necesario suministrar un voltaje para el microcontrolador y todos los PICS usados para el manejo de entradas y salidas, además de los circuitos que ayudan a aislar el ruido con los voltajes diferentes al Arduino Nano.

4.4.2. Microcontrolador ESP32

Es necesario un módulo que permita la conectividad con una red y poder llevar los datos a la nube y poder manejarlos correctamente según se disponga. El ESP32 es ideal para la implementación de capacidades IoT al PLC por la integración de Wifi y Bluetooth que este posee en su placa, también el módulo permite alimentarlo con 5VDC al igual que el Arduino Nano cuenta con un doble núcleo y su frecuencia de reloj es ajustable de 80MHz a 240 MHz. Tiene la posibilidad de apagar el CPU y utilizar el coprocesador de baja potencia y poder seguir supervisando los pines que estarán conectados paralelamente al Arduino Nano mediante el integrado ULN2803 que aísla los diferentes voltajes a los que estos microcontroladores trabajan.

4.4.3. LM2596 Módulo Regulador

Los convertidores DC-DC son módulos que permiten convertir el nivel de voltaje a uno de menor nivel. Para alimentar el PLC se va a utilizar un convertidor de nivel conmutado por su fácil uso y su excelente eficiencia energética.

Se requiere un convertidor de voltaje DC-DC *Step Down* para proporcionar al circuito un voltaje de salida constante y específico. Este dispositivo es capaz de reducir el voltaje de entrada de 24VDC a 5VDC, necesario para el funcionamiento del Arduino Nano, el ESP32 y otros componentes integrados que forman parte del PLC.

4.4.4. Circuitos integrados y componentes pasivos.

También se utilizó componentes electrónicos adicionales para brindar un funcionamiento completo del módulo. En la Tabla 3 se detallan los componentes.

Tabla 3. Descripción de los componentes del PLC.

Elemento	Descripción
ULN2803	Array de transistores Darlington para manejar cargas.
74LS04	Inversor lógico hexagonal.
74LS245	Transceptor bidireccional con bus de datos.
PC817	Optoacoplador con diodo emisor y fototransistor.
LM358	Amplificador operacional dual.
Resistencia	Limita corriente eléctrica.
Diodo	Permite el paso de corriente en una dirección.
Capacitor	Almacena energía en forma de campo eléctrico.
Pulsador	Interruptor momentáneo.
Fusible	Protección contra sobre corrientes.
Relé	Interruptor electromecánico.
Bornera	Conexión eléctrica segura.
Potenciómetro	Resistencia ajustable.

4.5. DISEÑO DEL SISTEMA

El desarrollo del sistema abarca varias etapas que incluyen la selección de hardware, la programación y configuración del software, así como la integración de todos los componentes para asegurar un funcionamiento óptimo. A continuación, se presenta el diagrama de flujo que detalla el proceso completo del sistema, desde la inicialización hasta la interacción con un aplicativo móvil a través de Firebase.

El sistema comienza con la inicialización del Arduino Nano y el ESP32. Una vez logrado, el Arduino Nano comienza a leer las señales de las entradas digitales y analógicas. Si hay señales presentes, se procesan los datos de acuerdo con la lógica de control programada en OpenPLC, y se actualizan las salidas digitales y analógicas.

Los datos se envían de forma paralela al ESP32, que se encarga de transmitir esta información a Firebase a través de una conexión WiFi. El sistema verifica continuamente el estado de la conectividad WiFi para asegurar una transmisión de datos estable. En caso de una desconexión, el sistema intentará reconectar automáticamente.

Además, los datos almacenados en Firebase son accesibles y manejables a través de un aplicativo móvil, permitiendo la monitorización y control remoto del sistema desde cualquier lugar con acceso a internet. El aplicativo móvil puede visualizar los datos de las entradas y salidas del PLC, además de controlar estas últimas, proporcionando una interfaz amigable para el usuario.

4.5.1. Cálculos y dimensionamientos de los componentes

Se realizaron cálculos para determinar que componentes utilizar para las entradas y salidas del PLC, y además para la señal que entra a los microcontroladores.

4.5.1.1. Entradas digitales

Se utiliza el optoacoplador PC817 para aislar la entrada de 24VDC del controlador que es a 5VDC, se ubica una resistencia en serie para limitar la corriente que ingresa al LED interno para activar el transistor fotodetector. La corriente directa del LED típicamente está entre 10mA y 20mA, entonces:

$$R = \frac{V_{in} - V_F}{I_F} = \frac{24 - 1.2}{0.01} = 2280\Omega$$

El valor comercial más cercano sería de 2.2 kΩ. Se procede a calcular la nueva corriente que va a pasar por el led del optoacoplador.

$$I = \frac{V}{R} = \frac{24 - 1.2}{2200} = 10.36mA$$

Se obtiene una corriente de 10.36mA, que activaría correctamente el fotodetector. Asumiendo que todas las entradas están activadas, sumarían unos 52mA de consumo en los optoacopladores.

4.5.1.2. Entradas Analógicas

Para las entradas analógicas que va de un rango de 0-10V es necesario que al microcontrolador solo llegue 0-5V, para ello se utilizará un divisor de voltaje con resistencias que brindará este cambio de voltaje de manera lineal. Esto implica que:

$$V_{salida} = V_{entrada} \cdot \frac{R_2}{R_1 + R_2}, V_{salida} = 5V, V_{entrada} = 10V$$

Entonces:

$$\frac{R_2}{R_1 + R_2} = \frac{5}{10} = 0.5$$

Esto significa que las dos resistencias deben ser iguales. Se eligen resistencias de $4.7k\Omega$ que es relativamente alta para limitar la corriente y tener lecturas más precisas. Se agrega una resistencia al inicio de 330Ω para limitar la corriente y tener la posibilidad de aumentar o disminuir su valor en un futuro.

4.5.1.3. Salida Analógica

El PLC cuenta con una salida analógica y se utiliza la salida PWM del microcontrolador para variar el voltaje de 0 a 5V, como para la salida se requiere un voltaje de 0-10V se utiliza el amplificador operacional LM358 con el que se realiza una operación de amplificación no inversora. Se procede a realizar los cálculos involucrados para lograrlo empezando con la ganancia:

$$G = 1 + \frac{R1}{R2} = 2$$

Cuando la ganancia es 2, el voltaje de salida será el doble que el de entrada, por lo que se obtendrá 0-10V en la salida analógica.

4.5.1.4. Entradas analógicas al ESP32

El microcontrolador ESP32 que enviará los datos a la nube trabaja a 3.3V, por lo que hay que volver a reducir el voltaje de la entrada que ya está en un rango de 0-5V, para ello se realiza el siguiente arreglo de resistencias:

$$\frac{R_2}{R_1 + R_2} = \frac{V_{salida}}{V_{entrada}} = \frac{3.3V}{5V} = 0.66$$

$$R_2 = 0.66 \cdot R_1 + 0.66 \cdot R_2$$

$$R_2 - 0.66 \cdot R_2 = 0.66 \cdot R_1$$

$$R_2 \cdot (1 - 0.66) = 0.66 \cdot R_1$$

$$R_2 \cdot 0.34 = 0.66 \cdot R_1$$

$$R_1 = 0.515 \cdot R_2$$

Como solo se usa el voltaje como referencia y al tener el ESP32 una alta impedancia se va a trabajar con corrientes bajas. Entonces se asigna a R_2 un valor de $10k\Omega$.

$$R_1 = 0.515 \cdot 10k\Omega = 5.15k\Omega$$

Redondeando, $R_1 = 5.1k\Omega$

4.5.1.5. Dimensionamiento del fusible de protección

Se va a realizar el cálculo de los consumos más significativos dentro del sistema. Primero se calculará para los relés.

Hay un total de 4 relés que funcionan a 24VDC, cada uno tiene un consumo aproximado de 18.7mA como se muestra en el Anexo 1.3, entonces:

$$I_{relés} = 4 \cdot 18.7mA = 74.8mA$$

Para las cargas de 5VDC asumidas por el LM2596 se procede a realizar los cálculos correspondientes. El Arduino Nano tiene un consumo típico de 20mA por cada pin según el Anexo 1.1, como no va a tener cargas significativas en sus salidas, se asume un valor de 100mA por las cuatro salidas digitales a los relés y los leds integrados en la placa.

Tomando el Anexo 1.2, el ESP32 consume un promedio de 80mA, pero en picos de WiFi puede alcanzar 500mA.

Entonces asumiendo un consumo máximo total a 5VDC y tomando la corriente también utilizada por los optoacopladores que sumarían unos 60mA:

$$I_{salida\ 5VDC} = 100mA + 500mA + 60mA = 660mA$$

Corriente en la entrada del LM2596 a 24VDC, considerando una eficiencia del regulador del 85%:

$$I_{entrada} = \frac{V_{salida} \cdot I_{salida}}{V_{entrada} \cdot eficiencia} = \frac{5V \cdot 660mA}{24V \cdot 0.85} \approx 162mA$$

Sumando las corrientes principales, se obtiene un consumo total máximo aproximado de:

$$I_{total} = I_{relés} + I_{entrada} = 74.8mA + 162mA = 236.8mA$$

Para el dimensionamiento del fusible se recomienda un fusible con un margen de seguridad del 150%-200% sobre la corriente nominal del circuito. Esto es para evitar disparos innecesarios por picos momentáneos. Por lo tanto:

$$I_{fusible} = I_{total} \cdot 1.5 \approx 236.8mA \cdot 1.5 = 355.2mA$$

Como hay pérdidas pequeñas en los demás componentes que completan el sistema, pero son muy pequeñas, se tomará un valor un poco hacia arriba, entonces el fusible que se ocupará será de 0.5A.

4.5.2. Diagrama de flujo del sistema

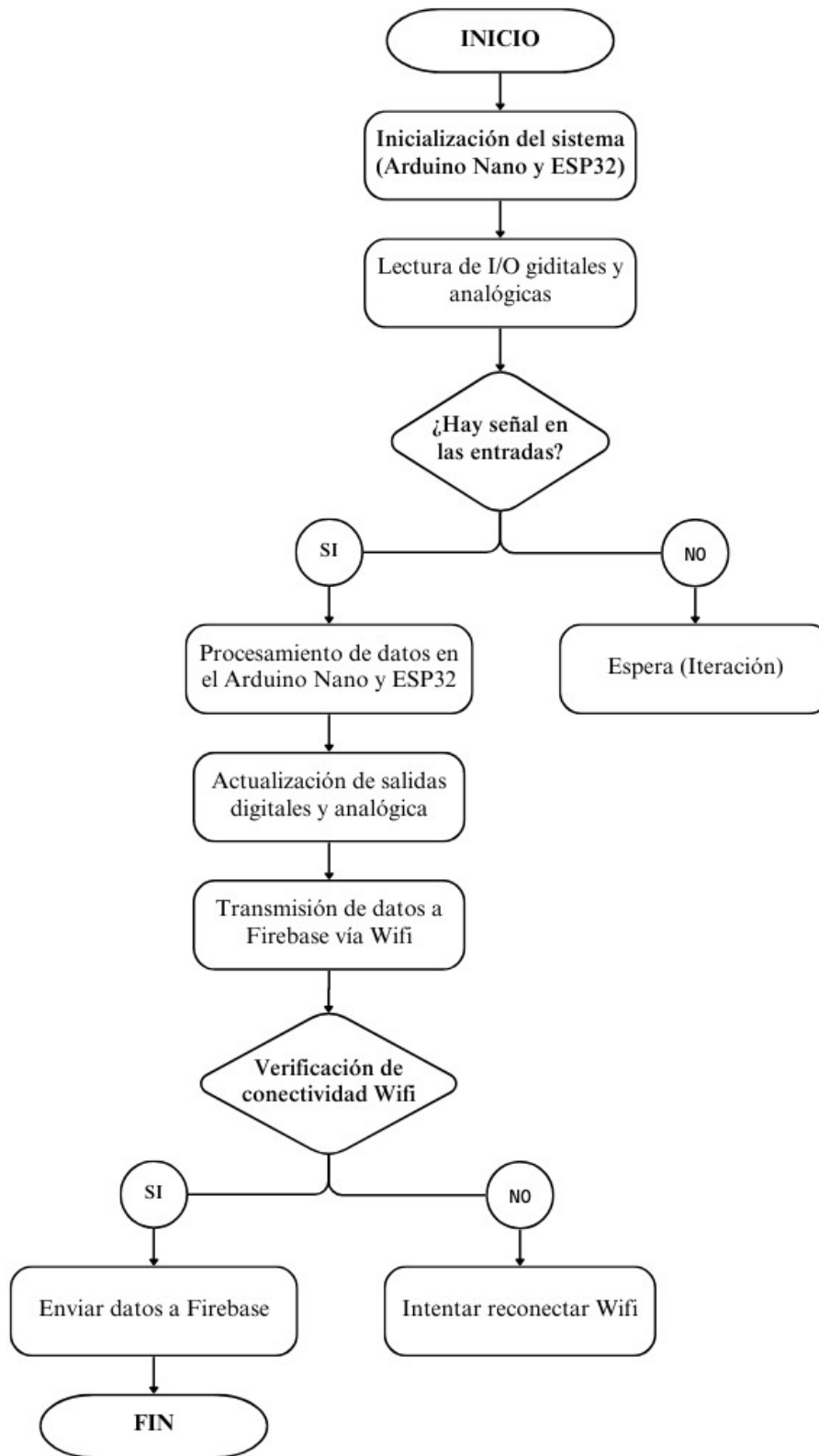


Figura 15. Diagrama de flujo del sistema.

4.5.3. Diagrama de conexión

A continuación, se presenta el diagrama de las conexiones del sistema, que muestra cómo se interconectan todos los componentes desde la fuente de alimentación hasta cada uno de los módulos utilizados. Este diagrama es ayudará a entender la distribución de la energía y la comunicación entre los elementos del sistema, asegurando un funcionamiento coherente y eficiente.

En la Figura 16 se muestra el diagrama de conexión del convertor DC-DC LM2596 para la alimentación de los componentes a 5VDC, cuenta con un fusible de 1A para evitar una sobre corriente y un diodo para prevenir una inversión de polaridad, además se toma de la alimentación de 24VDC que servirá para alimentar los relés.

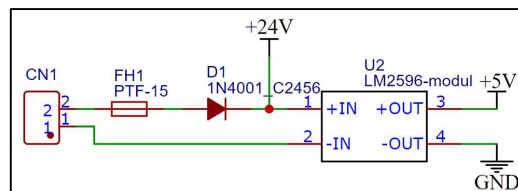


Figura 16. Alimentación del PLC.

En la Figura 17 se muestra la conexión para las entradas digitales, que consta de la bornera de 6 pines para 5 entradas digitales y un pin de GND. Como las entradas serán de 24VDC, es necesario una resistencia de 10kΩ para limitar la corriente que entra al optoacoplador PC817, con este además se logra una entrada aislada que va a abrir o cerrar el paso de GND hacia el siguiente integrado, la compuerta lógica 74LS04. Se consideró una bornera con salida de 24VDC para ocuparla en pulsadores o interruptores externos, y además de un GND adicional que servirá de referencia por si se utiliza una fuente externa de alimentación.

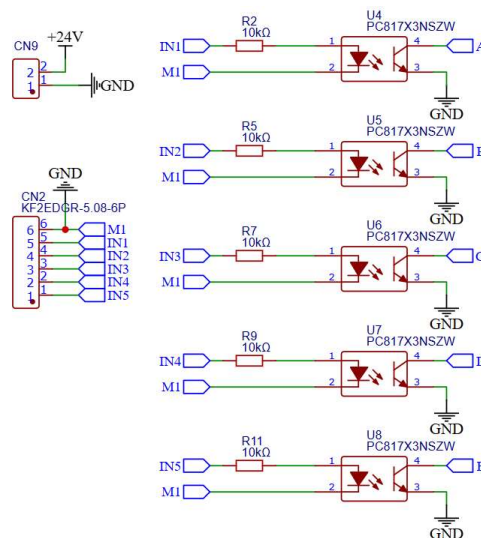


Figura 17. Diagrama para las entradas digitales del PLC.

Para el circuito presente en la Figura 18, en la salida de los optoacopladores se implementó un *pull-up* para mantener en nivel alto los pines del 74LS04, que procederá a invertir la señal a un estado bajo. Al colocar el inversor entre la señal de entrada y el microcontrolador, cualquier ruido presente en la señal será evadido ya que, sin importar los picos en la entrada, la salida mantendrá su estado apagado. A continuación, antes de la llegada de la señal al controlador pasará por el integrado 74LS245, que es un transeceptor de bus bidireccional que puede proporcionar aislamiento y amplificación de señales en sistemas digitales. Ofrece beneficios adicionales para el manejo de señales digitales, especialmente en la protección contra el ruido y la integridad de las señales. Además, con este integrado es posible activar o desactivar las entradas digitales mediante el pin, por si en algún momento no es necesario su uso. Finalmente, en su salida se tienen los 5VDC que irán al microcontrolador para su lectura adecuada.

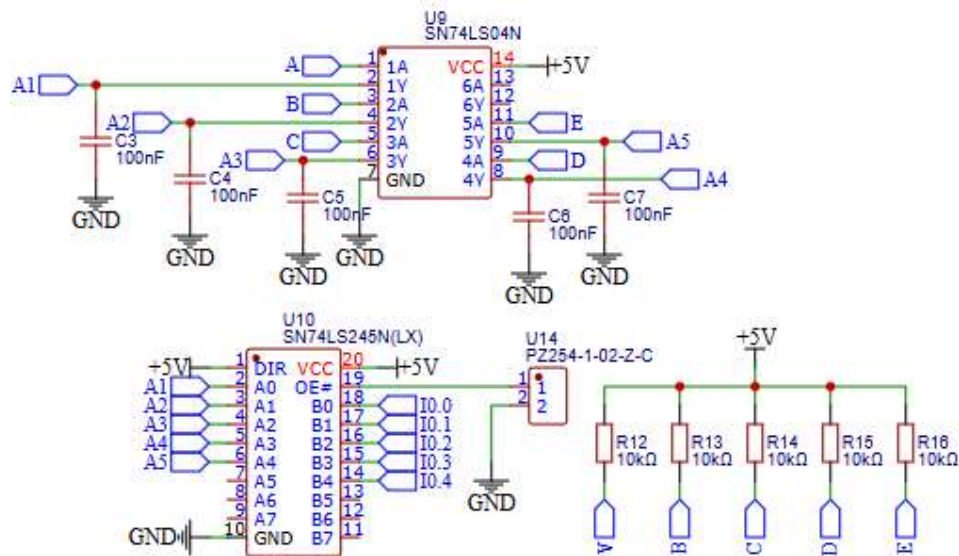


Figura 18. Diagrama para el aislamiento de ruido en las entradas digitales.

Como se muestra en la Figura 19 también se consideró dos entradas analógicas que aceptarán 10VDC, para ello solo fue necesario realizar una división de voltaje con resistencias en paralelo de una misma magnitud y además para proteger el circuito ante una conexión accidental de un mayor voltaje al soportado, se incluyó un diodo Zener que al sobrepasar su voltaje de ruptura de 10VDC, este permitirá el paso de corriente a GND para mantener el voltaje máxima designado. En la salida se tiene los 5VDC que soporta el Arduino en sus entradas y pueda realizar las lecturas correspondientes.

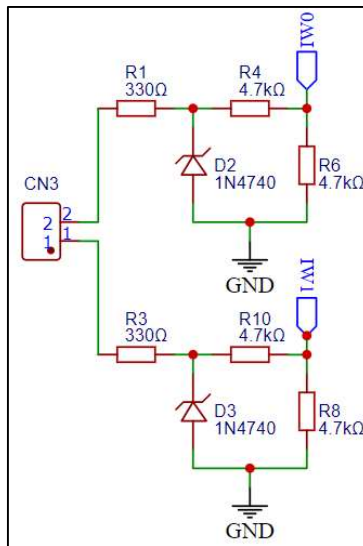


Figura 19. Diagrama para las entradas analógicas.

En la Figura 20 está el cerebro del PLC, el Arduino Nano que recibe las entradas digitales y analógicas antes mencionadas y dependiendo el programa cargado previamente desde el software EasyEDA controlará cuatro salidas digitales y una salida analógica, se incluyó un pulsador para resetear el Arduino Nano ante cualquier situación.

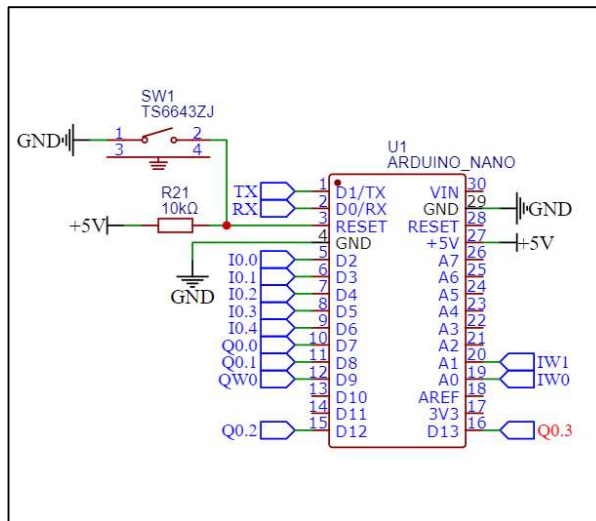


Figura 20. Diagrama del Arduino Nano.

Para controlar las salidas digitales se muestra en la Figura 21 los pines que ingresan al integrado ULN2803. Es un array de transistores Darlington de alta corriente que se utiliza para controlar cargas de alto voltaje y corriente. Para este caso controlará cuatro relés a 24VDC y debido a su construcción, ahorra varios componentes para controlarlos, y solo se referencia a GND las bobinas de los relés para activarlos y desactivarlos.

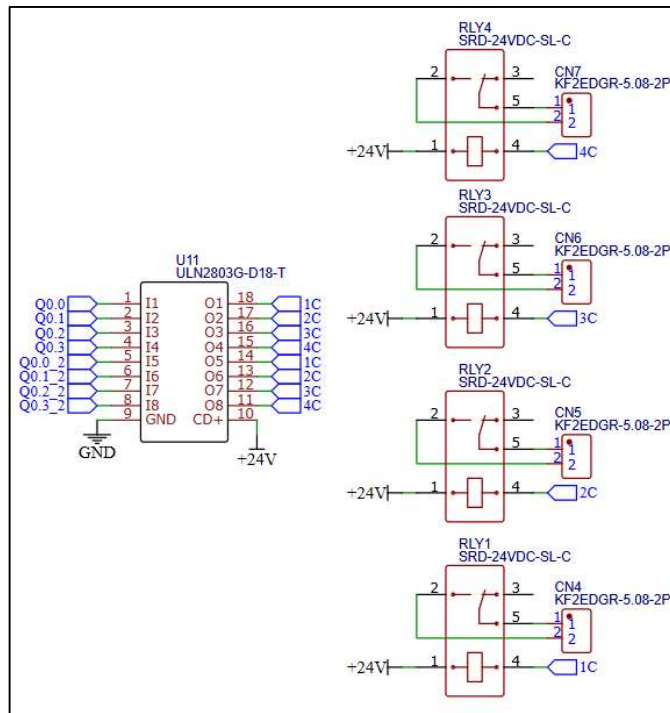


Figura 21. Diagrama de las salidas digitales.

En la Figura 22 se observa el circuito para la salida analógica. Al tener la necesidad de elevar el voltaje de 5VDC a 10VDC de manera lineal se ocupó un amplificador operacional con una configuración adecuada para lograrlo, con ayuda de un potenciómetro se podrá ajustar el voltaje de salida y que coincida con lo adecuado para controlar la instrumentación externa.

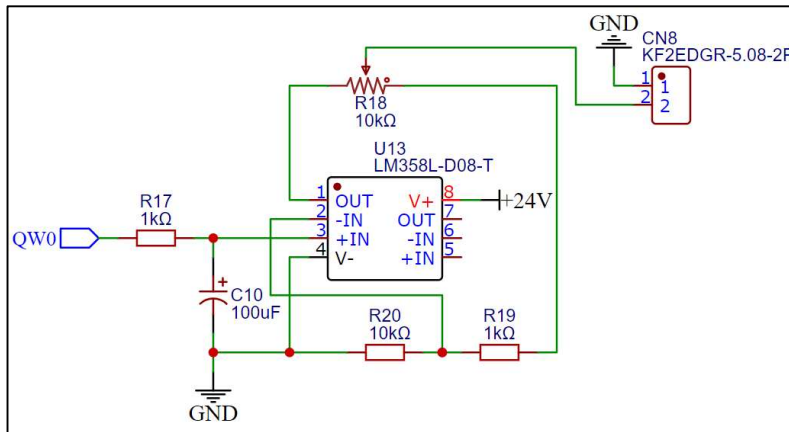


Figura 22. Diagrama para la salida analógica.

En la Figura 23 se muestra el último circuito del PLC y esta parte está relacionada con la implementación IoT. Consta de un ULN2803 para aislar las entradas de 5VDC para manejar un nivel de 3.3VDC que soporta el ESP32. El ESP32 es un controlador que implementa la conectividad Wifi para mandar estos datos a la nube y poder ser manejados posteriormente. Cuatro pines van hacia el ULN2803 del Arduino Uno para controlar las

salidas digitales remotamente, además consta de un arreglo de resistencias para leer también las dos entradas y salida analógicas. Al igual que el Arduino Nano, tiene un pulsador para resetear el módulo. Además, se puede con un puente inhabilitar la alimentación del ESP32 por si no es necesario su uso.

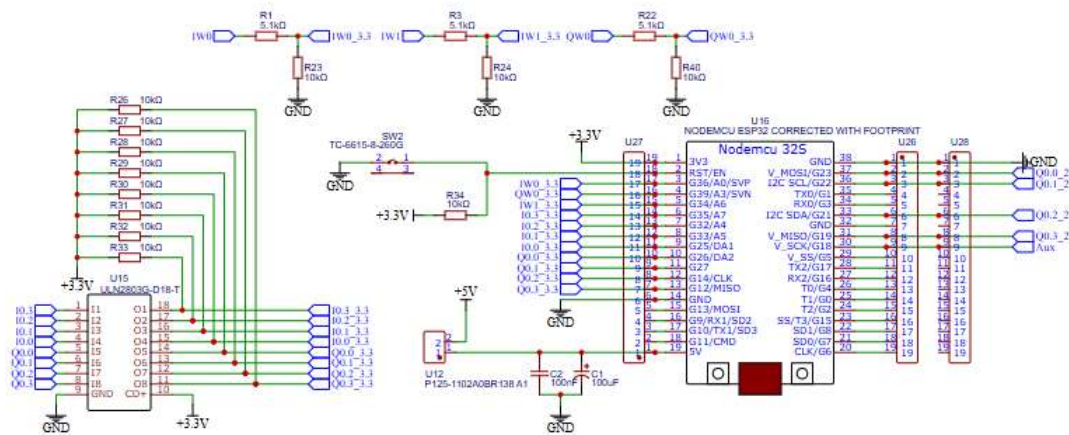


Figura 23. Diagrama de conexionado del ESP32.

En el Anexo 2 se muestra el diagrama de conexión completo realizado en EasyEDA, con todas las etiquetas correspondientes para cada uno de los nodos.

4.6. DESARROLLO DEL SISTEMA

El desarrollo del sistema de Controlador Lógico Programable (PLC) con capacidades IoT se compone de varias etapas esenciales que abarcan desde la selección y configuración del hardware, la programación y configuración del software, hasta la integración de todos los componentes para asegurar un funcionamiento óptimo y eficiente.

4.6.1. Creación de la base de datos en Firebase

Para guardar los datos y poder manipularlos con gran facilidad, se utilizará la plataforma Firebase de Google, para ello, se deberá crear un proyecto con nuestra cuenta de Google, y dentro del proyecto construir una base de datos, en el Anexo 3 se muestra todo el proceso a seguir. Posteriormente se procede a crear en la raíz un documento, el primer PLC se llamará “PLC_0001”, y en el interior todos los identificadores para las entradas y salidas tanto digitales como analógicas. Como se ve en la Figura 24.

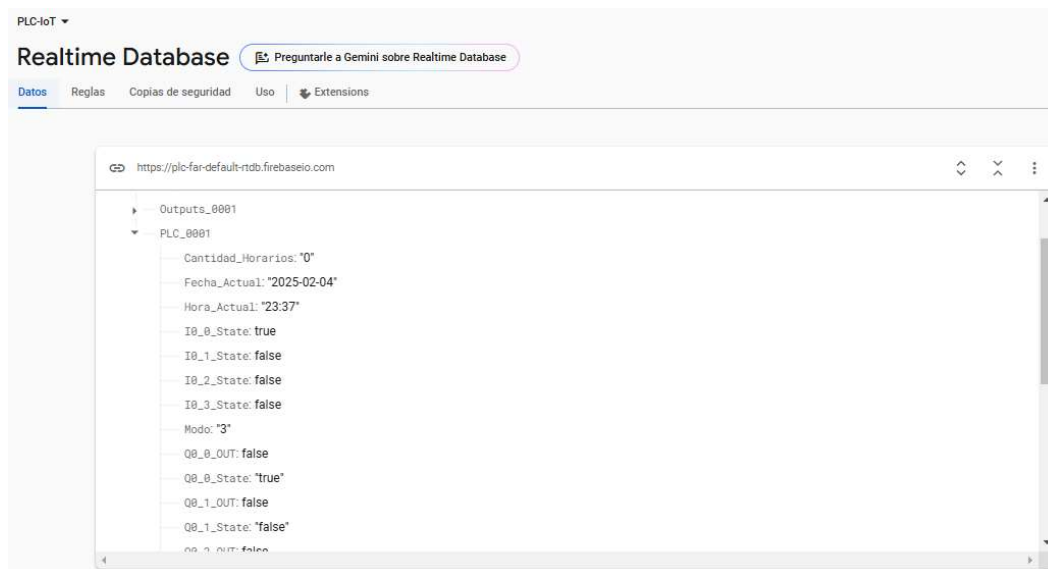


Figura 24. Base de datos creadas para trabajar con el PLC.

Se ha utilizado la función llamada Real Time Database (base de datos en tiempo real) que brindará una forma sencilla y rápida de manejar los datos para el PLC, cuenta con una raíz que será el nombre asignado al PLC y todos los indicadores van por debajo.

Firestore cuenta con varios planes que otorgan diferentes beneficios, el plan básico que no tiene ningún costo tiene los siguientes límites presentados en la Figura 25.

Base de datos en tiempo real

Conexiones simultáneas	100
GB almacenados	1 GB
GB descargados	10 GB por mes
Varias bases de datos por proyecto.	

Figura 25. Plan gratuito de Real Time Database.

Esto es más que suficiente para manejar varios PLCs y no tener que pagar por el servicio, pero ya si se necesita mejorar el plan, solo suma lo que se consume a la facturación, no es necesario pagar si no se cumplen los límites establecidos por el plan.

Cabe recalcar que se ha creado la base de datos en modo de prueba y no es seguro el manejo de datos, pero para este proyecto es conveniente realizarlo de este modo, solo es necesaria la URL hacia la base de datos para escribir y leer los datos.

4.6.2. Diseño de la placa del sistema

Una vez concluido todo el proceso previo se procede a realizar la placa del circuito, para el diseño se utilizó el software web EasyEDA. En la plataforma además también se realizó el esquemático del prototipo que fácilmente se traslada a una PCB. En la Figura 26 se muestra el PCB que se ha desarrollado. Cuenta con medidas de 16.65cm de largo y 10cm de alto.

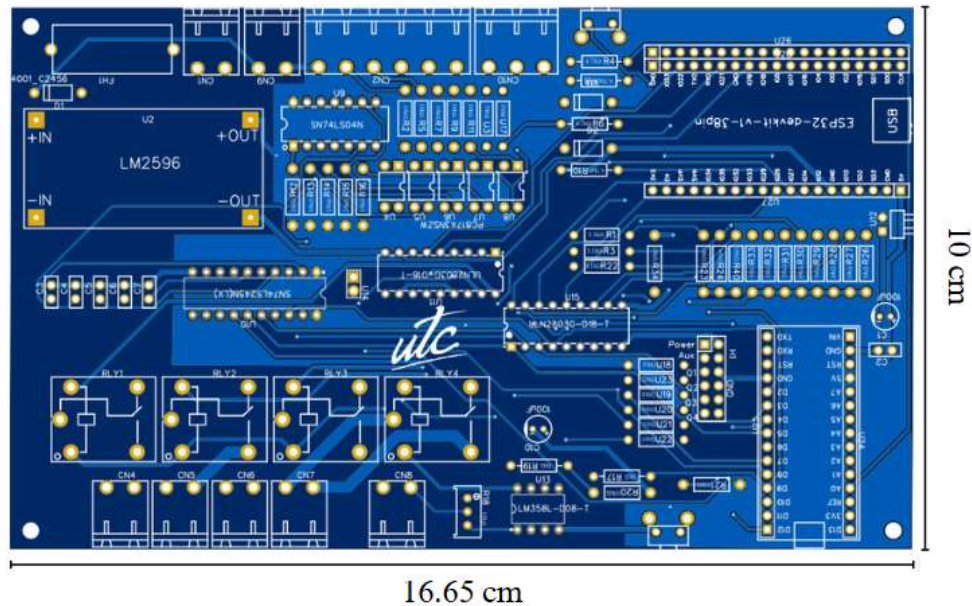


Figura 26. Diseño de la PCB del PLC.

El PCB consta de dos capas, una superior y una inferior, la superior tiene la malla de los 5VDC y la inferior el GND. De esta manera se tiene un fácil enrutamiento de las pistas y reduce el ruido eléctrico, distribuye uniformemente la alimentación, reduce caídas de tensión, mejora la integridad de la señal, minimiza las interferencias electromagnéticas y asegura un retorno de corriente más eficiente, optimizando el rendimiento del circuito.

La mayoría de pistas tienen diámetros similares de 0.6mm en su mayoría para un seguro flujo de corriente a los componentes que da un máximo de 1A, las pistas de los relés son las de mayor grosor entre sus pines y las borneras, ya que van a trabajar con grandes corrientes, la anchura de estas pistas será de 2mm que permite un paso de corriente aproximado de 7A, lo que es suficiente para servir de interruptor para muchos componentes.

Se ubicó cuatro puntos de perforación en cada una de las esquinas de 3mm de diámetro que perfectamente encajan los tornillos M3, así en un futuro pueda servir de anclaje para su propia caja, montarse en algún cajetín o ponerle soportes para riel din. En el Anexo 4 se muestra la PCB completa en 3D.

4.6.3. Programación del ESP32

Una vez esté lista la base de datos y la PCB lista con los elementos soldados en ella. Se procede a escribir un programa en Arduino IDE para posteriormente cargarlo al ESP32 y brindar de la capacidad IoT al PLC.

Para que el IDE pueda trabajar con la placa de desarrollo ESP32, que ha sido seleccionada previamente, es necesaria la instalación de componentes como las librerías y la placa que de forma nativa no se encuentra. Este proceso se encuentra detallado en el Anexo 5. Una vez completada la instalación, se procede a escoger la placa como se muestra en la Figura 27.

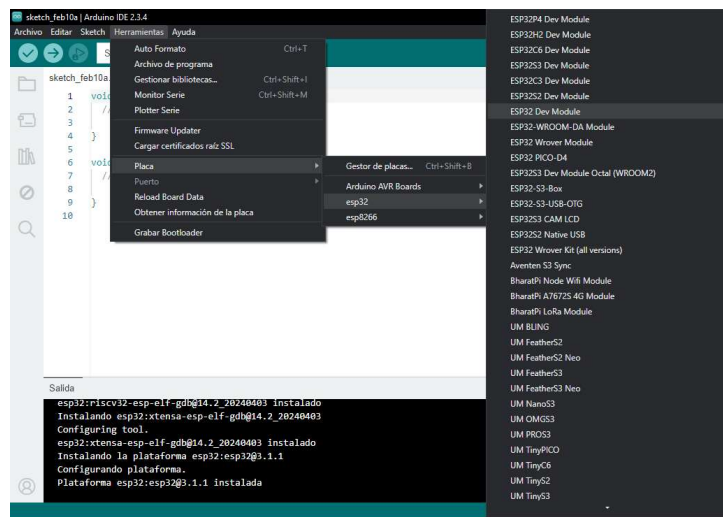


Figura 27. Selección de la placa ESP32 Dev Module.

Para lograr el funcionamiento correcto del código es necesario incluir las siguientes bibliotecas señaladas en la Figura 28 que ya habían sido incluidas en el procedimiento del Anexo 5 y el resto ya vienen en el IDE por defecto.

```
1 #include <Wifi.h>
2 #include "FirebaseESP32.h"
3 #include "time.h"
4 #include "esp_sntp.h"
5 #include <WifiManager.h>
```

Figura 28. Librerías utilizadas para la programación del ESP32.

Primeramente, se inicializan todas las variables públicas del código, que corresponden a las credenciales de Firebase, los pines de entradas y salidas tanto digitales como analógicas y por último las variables de control. En la Figura 29 se presenta el esquema correspondiente.

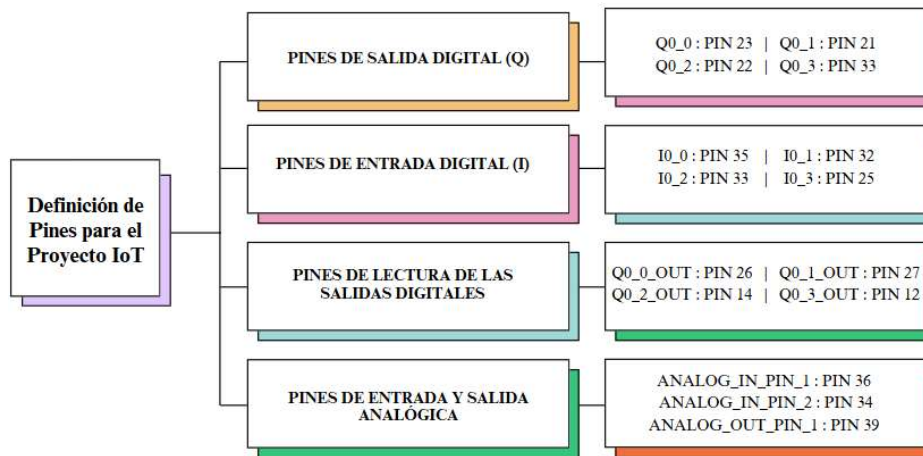


Figura 29. Variables declaradas en el código antes del *Setup*.

El *Setup* sirve para realizar todos los procesos previos a correr el código principal en bucle, en este apartado se inicializa el *baud rate* de comunicación a 115200 *bauds* que comúnmente suele trabajar el ESP32, se conectará a internet si previamente se guardó la red WiFi, sino ve a abrir el AP que permite realizar la conexión con cualquier dispositivo a la red Wifi más cercana. Por último, configura la conectividad con Firebase y realiza la configuración del tiempo actual. En la Figura 30 se muestra lo descrito.

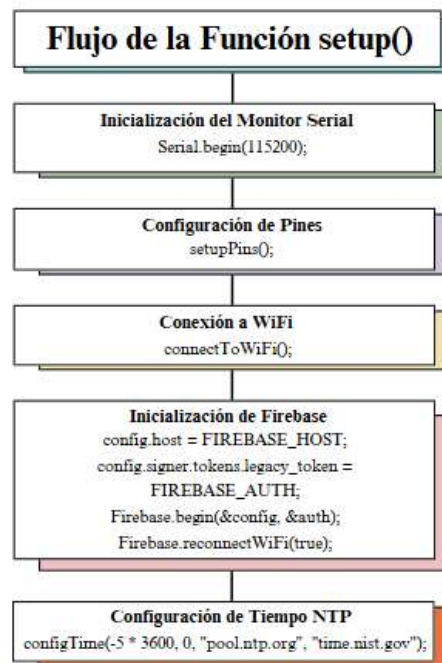


Figura 30. Inicialización del código en el ESP32.

A continuación, viene la parte del código que se va a repetir en bucle, por lo que se realizó las funciones fuera de este para facilitar la programación, y solamente se trabaja con modos de funcionamiento seleccionados y permanentemente va a estar leyendo las

entradas y salidas sin importar que se esté haciendo en la App. En la Figura 31 se muestra el código del *Loop*.

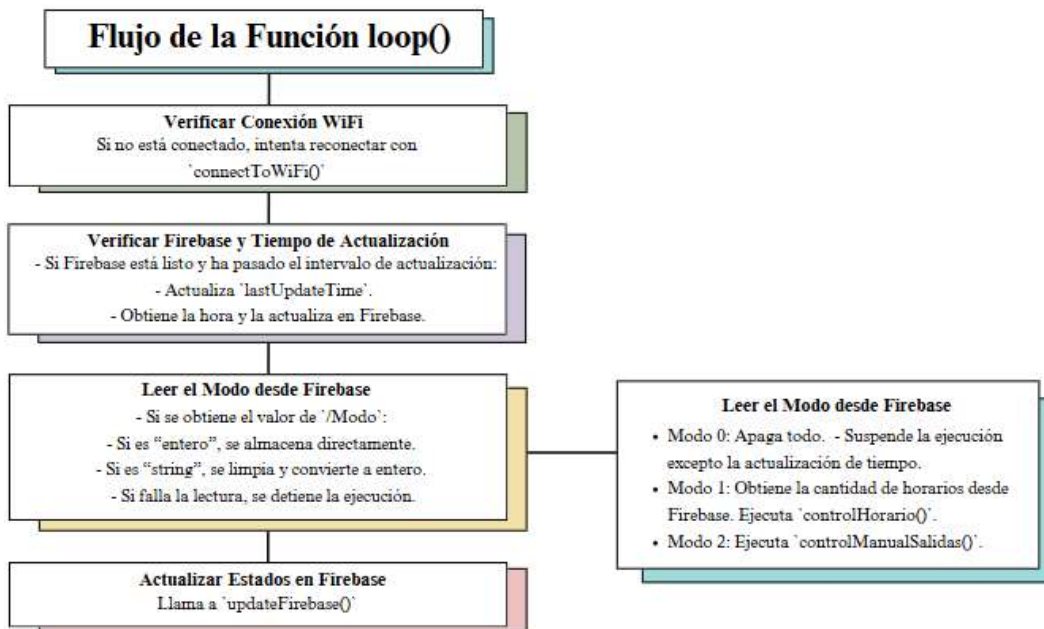


Figura 31. Bucle del código en el ESP32.

Como se observó anteriormente el código trabaja con funciones externas que son llamadas por el *Loop*.

4.6.4. Desarrollo de una App para interactuar con el PLC

En esta sección, se describirá el proceso de creación de una aplicación móvil utilizando AppInventor. La app permitirá la manipulación y lectura de datos desde la Firebase Real Time Database, proporcionando una interfaz gráfica amigable para el control y monitoreo del sistema PLC con capacidades IoT.

La aplicación interactuará con la base de datos en tiempo real, permitiendo a los usuarios acceder a datos de entradas y salidas, así como enviar comandos para controlar el PLC de manera remota.

En la Figura 32 se procede a crear un proyecto en la web App Inventor.

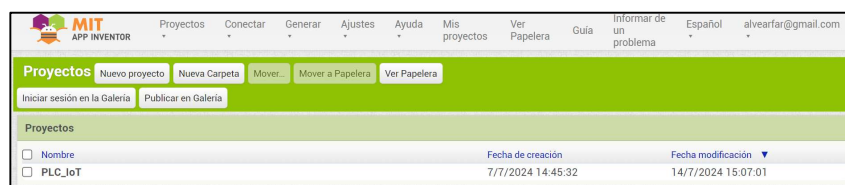


Figura 32. Proyecto creado para realizar la App.

Se procede a crear la interfaz de la aplicación de la manera que se desee, en la Figura 33 se muestra cómo se construiría la interfaz. Y se agrega en los componentes la extensión FirebaseDB que brinda una comunicación sencilla con la base de datos que se ha creado previamente.

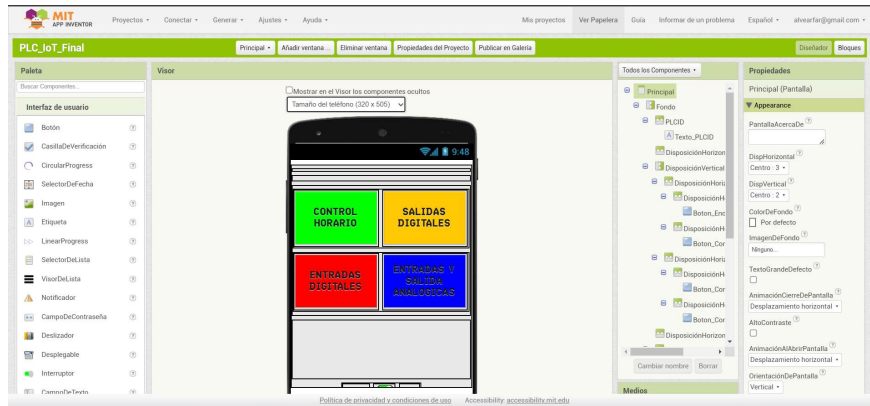


Figura 33. Construcción de la interfaz gráfica de la App.

Para que funcione la aplicación es necesario programarla usando bloques, en donde se tomará los datos de las lecturas cada vez que se actualicen en la base de datos y ubicar los valores en la app. En la Figura 34 se observa el diagrama de bloques de la app.

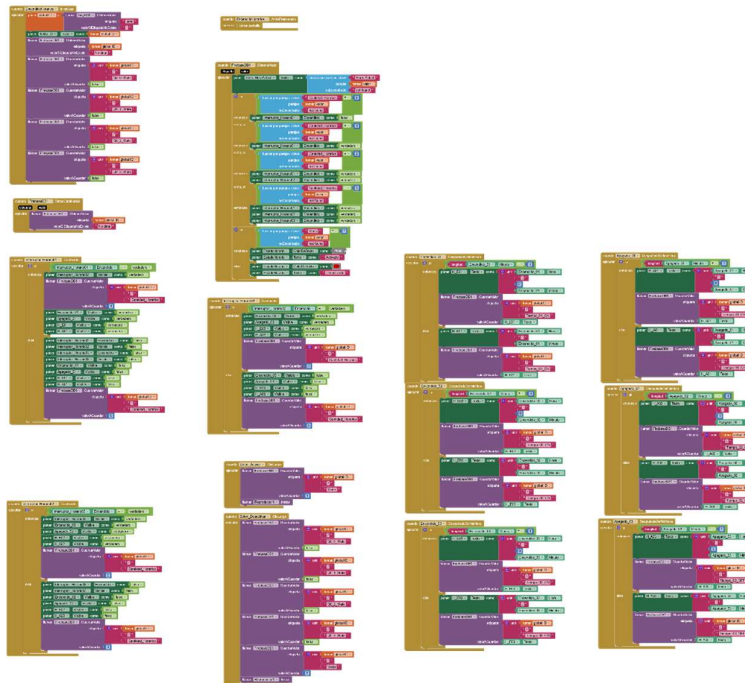


Figura 34. Programación por bloques usando AppInventor.

Finalmente se instala la aplicación en un dispositivo móvil para visualizar los datos y también realizar escrituras. En la Figura 35 se muestra la aplicación donde con círculos verdes se indica que una lectura indica estado activa y un círculo rojo que está apagado, se usa selectores para encender y apagar las salidas digitales y para la lectura de entradas

analógicas se tiene puede leer el valor y además obsérvalo en una barra que indica el porcentaje total medido.

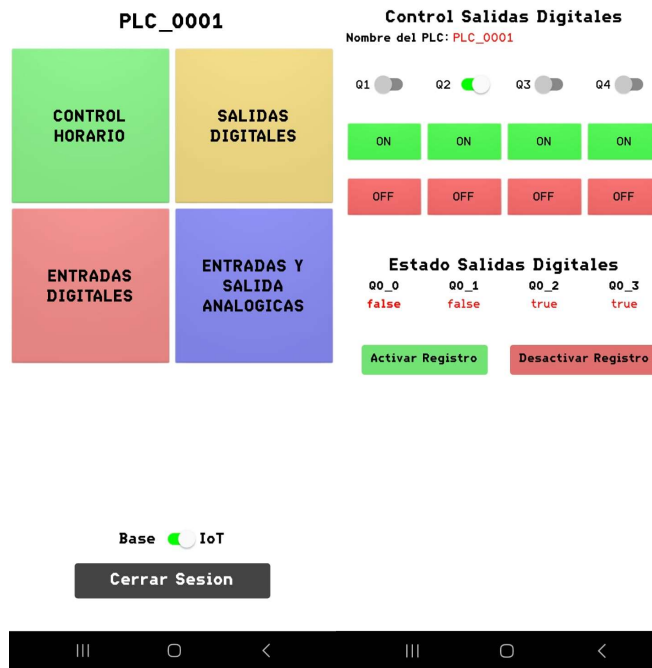


Figura 35. Interfaz de la App vista en el celular.

Así como se ha manejado los datos desde un celular es posible hacerlo desde cualquier otra plataforma, como un programa, una página web, con un chat de voz, un *ChatBot* y otras tantas opciones. En el Anexo 6 se encontrarán todos los bloques utilizados para cada pantalla de la aplicación creada.

4.6.5. Diseño de la carcasa del sistema

Una vez realizada la PCB y soldado los componentes necesarios para su funcionamiento, es importante protegerlo del exterior para que no sufra golpes o sea afectado por agentes externos como polvo, líquidos, etc. Para ello se va a realizar el diseño de una caja en el software Autodesk Fusión para luego imprimir el modelo usando una impresora 3D, para realizarlo se utilizará la impresora Anycubic Kobra.

Primero se va a realizar el diseño de una base donde irá montado el PCB mediante tornillos M3 y a su vez esta base se le podrá acoplar soportes para riel rin que se imprimirán después. Se incluye unas cejas que serán necesarias para que se acople con la tapa superior. En la Figura 36 se muestra la base terminada.

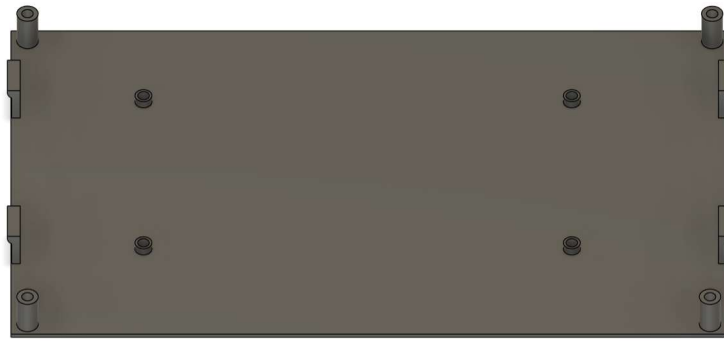


Figura 36. Diseño de la base de la caja del PLC.

La tapa tendrá los orificios necesarios para poder presionar los botones, observar las luces internas y para todas las borneras que contiene la placa. Por último, se realizan muescas para los indicadores, se ubica el nombre de la universidad y el nombre del PLC. Se muestra en la Figura 37 como se verá la tapa. En el Anexo 7 se encuentran los modelos y planos de la base y tapa respectivamente.

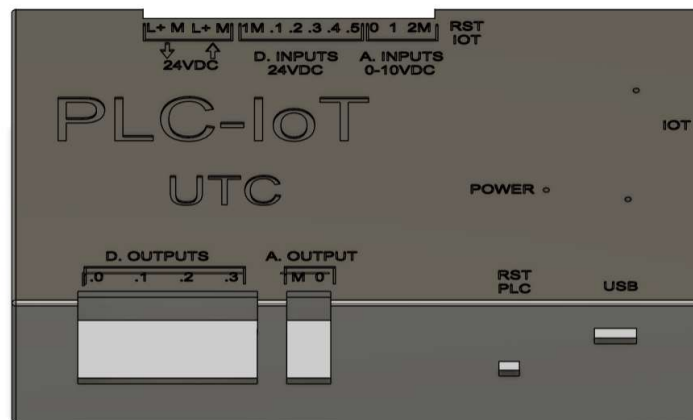


Figura 37. Diseño de la tapa que cubre al PLC.

En la Figura 38 se muestra el proceso de impresión de una de las partes de la caja que se usará para proteger la placa del PLC.



Figura 38. Impresión de la caja utilizando la impresora 3D.

4.6.5.1. Nivel de protección IP

Los PLC comerciales generalmente cuentan con una clasificación de protección IP20, lo que indica que están protegidos contra el ingreso de objetos sólidos mayores a 12.5 mm de diámetro, pero no tienen resistencia al agua. En este proyecto, la carcasa impresa en 3D ha sido diseñada considerando este estándar, asegurando un nivel de protección equivalente mediante las siguientes características:

Diseño estructural cerrado: La caja impresa en 3D cubre completamente la electrónica del PLC, evitando el acceso directo a los componentes internos y protegiéndolos contra el contacto accidental con herramientas u otros objetos mayores a 12.5 mm, cumpliendo con la exigencia de IP20. Las únicas aberturas grandes son donde irán las borneras de conexiones, pero estas solo serán vulnerables si se retiran y no se vuelven a colocar en su sitio.

Materiales resistentes: La impresión 3D se realizó con PLA, asegurando una estructura robusta y con su facilidad de impresión permite reemplazar la caja fácilmente ante algún siniestro en ella.

Tabla 4. Comparación de los diferentes tipos de filamentos.

Material	Resistencia Mecánica	Resistencia Térmica	Resistencia Química	Facilidad de impresión	Costo
PLA (Ácido Poliláctico)	Media	Baja	Baja	Muy Fácil	Bajo
ABS (Acrilonitrilo Butadieno Estireno)	Alta	Media (hasta 100°C)	Media	Difícil (requiere cama caliente)	Medio
PETG (Polietileno Tereftalato Glicol)	Alta	Media (hasta 80°C)	Alta	Fácil	Medio

Ajuste y ensamble adecuado: El diseño de la caja incluye un sistema de cierre adecuado, asegurando que no existan espacios abiertos por los cuales puedan ingresar elementos que afecten el funcionamiento interno del PLC.

Con estas características, la carcasa diseñada ofrece un nivel de protección similar a IP20, adecuado para su uso en ambientes controlados sin exposición directa a polvo fino o líquidos, cumpliendo con los requisitos básicos de resguardo de los PLC comerciales.

5. ANÁLISIS DE RESULTADOS

En esta sección se presenta un análisis detallado de los resultados obtenidos a lo largo del desarrollo e implementación del PLC con capacidades IoT. El análisis se centrará en evaluar el rendimiento del sistema, la efectividad de la comunicación con Firebase, la funcionalidad de la aplicación desarrollada en AppInventor, y otras formas de control y monitoreo implementadas.

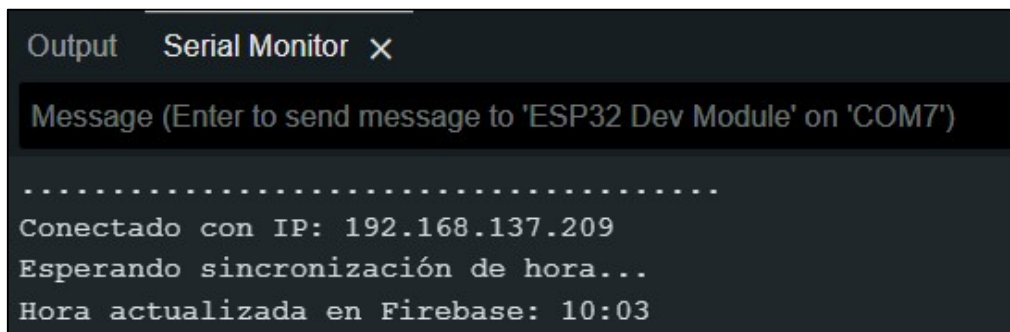
5.1. PRUEBA DE VALIDACIÓN DEL PROTOTIPO

En esta sección se detallan las pruebas de validación realizadas para asegurar el correcto funcionamiento del prototipo. Se verificaron todos los componentes y funcionalidades, desde la integración del hardware hasta la comunicación con Firebase y el control mediante la aplicación móvil, asegurando que el prototipo cumpla con los requisitos y objetivos establecidos en el proyecto.

5.1.1. Eficacia de la programación

En este apartado se verificará si el ESP32 se conecta a la red Wifi cuando se la guarda dentro de la memoria flash del ESP32 utilizando la librería WifiManager y además puede leer y escribir en la base de datos de Firebase que daría la capacidad IoT al PLC que se va a construir. En la Figura 39 se muestra el monitor serial del ESP32 donde se guarda la hora y se muestra la IP de la red.

En el Anexo 6 se adjunta el código completo de la programación, donde se abarca todo lo necesario para controlar las salidas, realizar todas las lecturas necesarias tanto digitales como analógicas y los procesos de comunicación con la base de datos para un correcto funcionamiento.



```
Output Serial Monitor ×
Message (Enter to send message to 'ESP32 Dev Module' on 'COM7')
.....
Conectado con IP: 192.168.137.209
Esperando sincronización de hora...
Hora actualizada en Firebase: 10:03
```

Figura 39. Monitor serial que muestra las impresiones serial del ESP32.

5.1.2. Pruebas de comunicación con Firebase

El ESP32 debe ser capaz de enviar la información a la base de datos y esta se refleje correctamente en la misma. En la Figura 40 se muestra la ventana donde se guardan los datos en la base de datos.

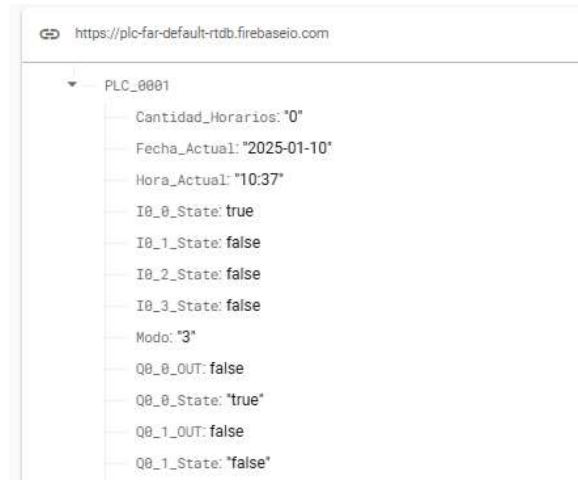


Figura 40. Datos guardados en Real Time Database de Firebase.

5.1.3. Programación con OpenPLC

El Arduino Nano funciona como cerebro del PLC, por ello es importante que tenga compatibilidad con el software usado para programar, y es necesario cargar algún ejemplo que pueda evidenciar el correcto funcionamiento del PLC. En la Figura 41 se muestra la programación en Ladder utilizando OpenPLC, que luego se cargará al microcontrolador del PLC. Además, se puede consultar el manual de usuario en el Anexo 11.

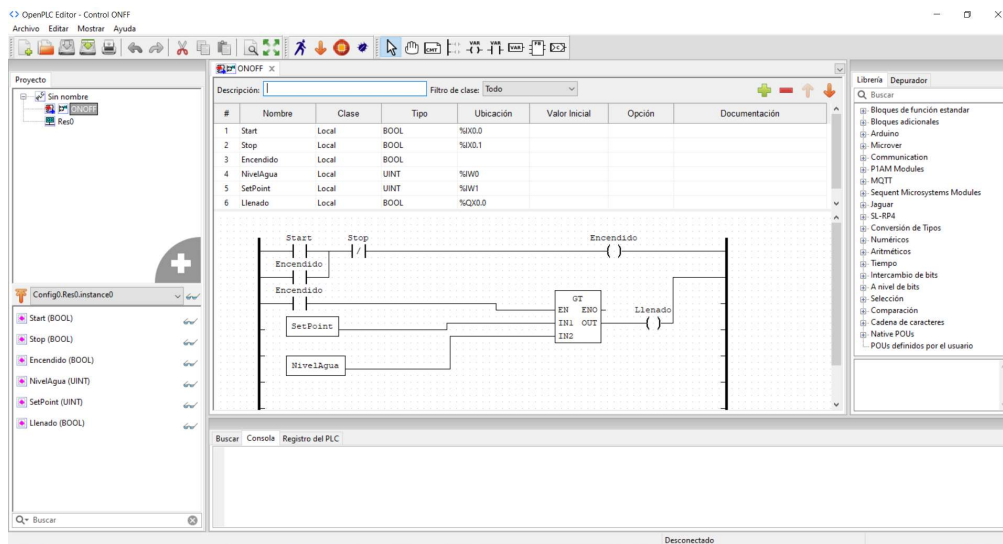


Figura 41. Interfaz y programación en OpenPLC.

5.2. FUNCIONALIDAD DE LA APLICACIÓN DESARROLLADA EN APPINVENTOR

5.2.1. Experiencia de uso

La experiencia de uso de la aplicación desarrollada para el control y monitoreo del sistema basado en PLC con capacidades IoT se evaluó a través de diversas pruebas de interacción con el usuario. El enfoque está en diseñar una interfaz de usuario fácil de usar y amigable, que permita a los usuarios operar y rastrear el sistema sin problemas. Los resultados se sintieron completamente naturales durante las pruebas, tan fácil como la navegación o la buena posición de las opciones en la pantalla principal, donde se puede acceder rápidamente a las funciones del sistema disponible (encender y apagar salidas, visualizar entradas o la configuración de parámetros).

Otras características incluyeron botones de control e indicadores de estado, junto con la opción de exportar datos recopilados durante un período de tiempo en formato de hoja de cálculo de Google. La integración con Firebase permitía que los datos se reflejaran en tiempo real en el sistema, mientras ofrecía control total.

Al desarrollar la aplicación, la interfaz y los elementos se ajustaron adecuadamente a diferentes tamaños de pantalla, lo que proporcionó una adaptación adecuada de los elementos para una amplia gama de dispositivos móviles.

5.2.2. Capacidad de control y monitoreo remoto

La aplicación, junto con el sistema, permitió al usuario controlar las funcionalidades de los PLC desde cualquier lugar debido a la disponibilidad de conexión a Internet, permitiendo al usuario interactuar con los dispositivos conectados.

Se observó que la aplicación era suficiente y adecuada, y monitoreaba las entradas y salidas del PLC, donde reflejaba activamente los datos de los sensores y actuadores. Además de eso, una persona tenía la opción de controlar si los otros dispositivos estaban conectados a un relé específico, un elemento que originalmente estaba en el diseño del sistema real.

Los escenarios industriales y de automatización son relevantes para esta funcionalidad de control remoto. El sistema de control remoto también era estable, con baja latencia y sin desconexiones del dispositivo durante las pruebas. Esto demuestra la fiabilidad del PLC, asegurando que el monitoreo y control del sistema puedan realizarse de manera continua y sin interrupciones significativas.

5.3. PRUEBAS DE CAMPO Y ESTUDIOS DE CASO

Las pruebas en el desarrollo de un PLC son importantes para verificar que el sistema de automatización funcione correctamente antes de su implementación final. Estas pruebas aseguran que el programa, el hardware y la integración del sistema operen de acuerdo a lo especificado. A continuación, se detallan las principales pruebas que se realizan durante el desarrollo de un PLC para su aplicación:

5.3.1. Pruebas y Resultados

A continuación, se presentan algunas aplicaciones de control y automatización realizados en el laboratorio de la UTC para sus respectivos análisis de resultados:

5.3.1.1. Encendido de una lámpara modo blink

En la Figura 42 se muestra la conexión del PLC para el encendido de lámpara aplicando los dos modos de funcionamiento.

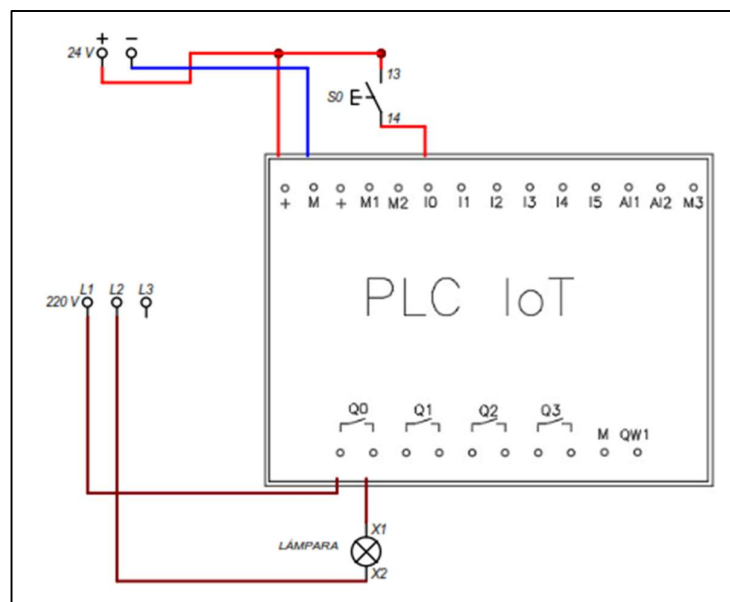


Figura 42. Diagrama de conexión encendido de lámpara dos modos.

La lámpara parpadeará (Blink) de manera periódica, encendiéndose y apagándose en intervalos de tiempo definidos en la Figura 43. Esto demuestra el uso correcto de temporizadores (TON y TP) donde el TP es un temporizador de pulso es usado para generar un pulso de salida con una duración determinada de 2 [s] y el TON es un temporizador con retraso es usado para activar una salida, un tiempo fijo de 2 [ms] después de la activación de una entrada.

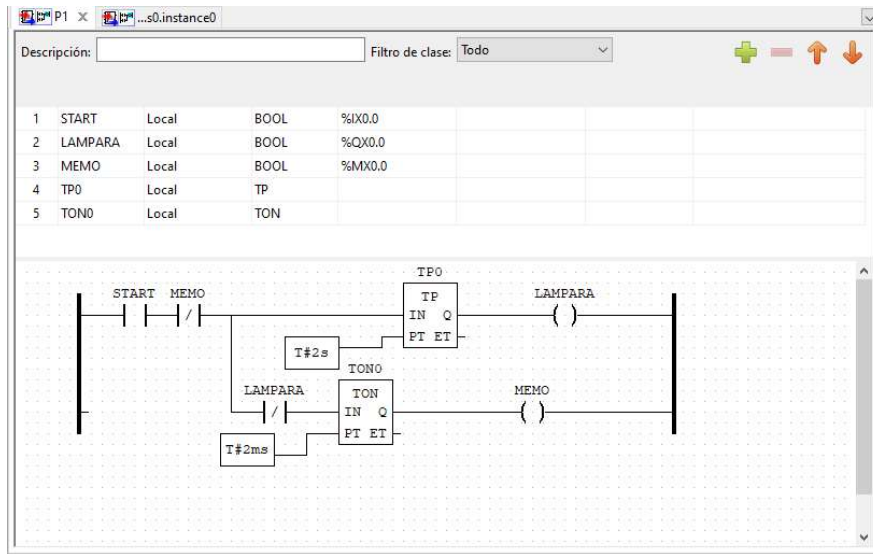


Figura 43. Programa del encendido de una lámpara (Blink).

5.3.1.2. Encendido de una lámpara por un pulsador

La lámpara se encenderá únicamente cuando el pulsador esté presionado, y se apagará cuando se suelte. Esto confirma que el PLC está leyendo correctamente las entradas digitales (Pulsador) y actuando sobre las salidas digitales (Lámpara) en función de la lógica programada en la Figura 44.

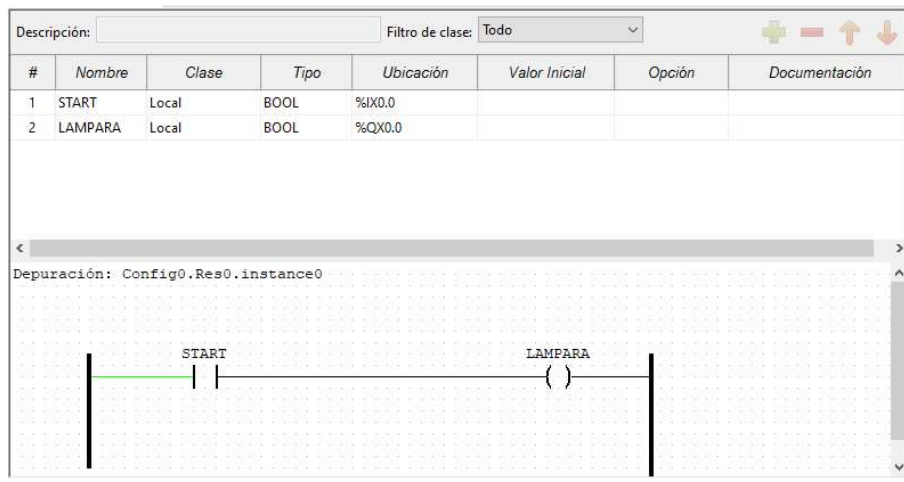


Figura 44. Programa del encendido de una lámpara (Pulsador).

5.3.1.3. Enclavamiento de un contactor

En la Figura 45 se muestra la conexión del PLC que describe el enclavamiento del contactor mediante dos pulsadores, uno abierto para activar el enclavamiento y otro cerrado para desactivarlo.

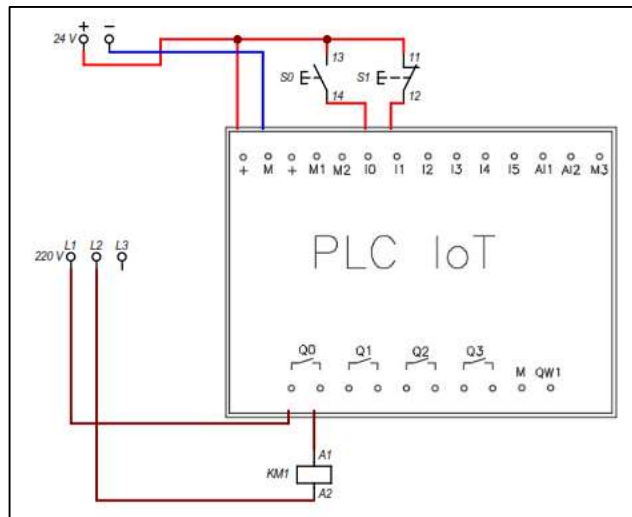


Figura 45. Diagrama de conexión enclavamiento de un contactor.

El enclavamiento de un contactor en el PLC se implementó con éxito, demostrando un funcionamiento estable y seguro. Al activar la entrada I0.0 (Pulsador de START), la salida Q0.0 se energizó y mantuvo su estado de funcionamiento gracias a la lógica de autoretención programada en el PLC de la Figura 46. La desenergización fue inmediata al presionar I0.1 (Pulsador de STOP), cumpliendo con los requisitos de seguridad. El sistema operó sin fluctuaciones durante 15 minutos, validando su confiabilidad para aplicaciones industriales. Esta practica verifica la eficacia del PLC en el control automatizado de contactores, sentando las bases para aplicaciones más complejas.

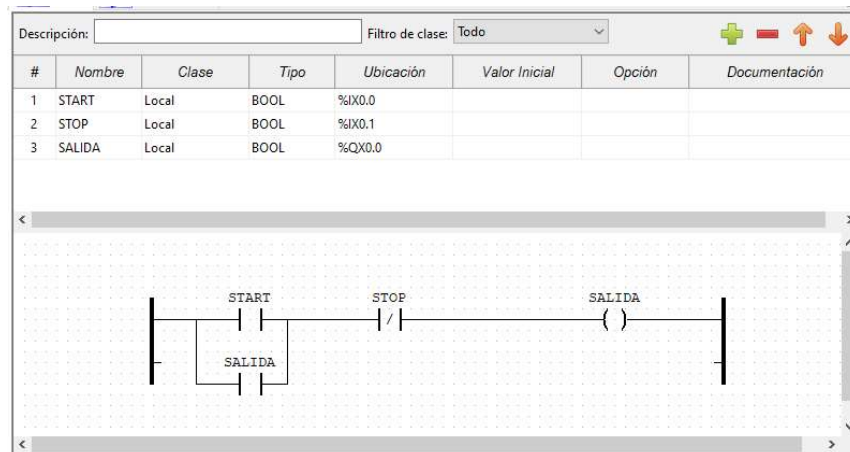


Figura 46. Programa de enclavamiento de un contactor.

5.3.1.4. Control de un semáforo de una vía

En la Figura 47 muestra las conexiones de los dispositivos en un sistema de control de semáforo de una vía utilizando el PLC.

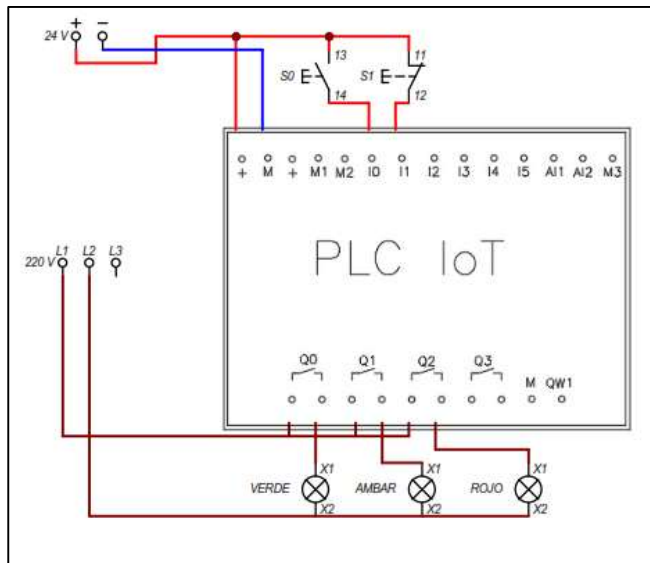


Figura 47. Diagrama de conexión (Semáforo).

La Figura 48 muestra un diagrama de programación en lenguaje Ladder para controlar un semáforo utilizando el PLC. En el esquema, se utilizan entradas Start y Stop para iniciar y detener el sistema, junto con una memoria interna M para mantener el estado. Los temporizadores TON como T_1, T_2 y T_3, controlan los intervalos de tiempo: T_1 con 4 segundos, T_2 con 3 segundos y T_3 con 4 segundos, activando secuencialmente las salidas para las luces verde, ámbar y roja. La lógica asegura que cada luz se active en el orden correcto, utilizando los contactos NO (Normalmente abierto) y bobinas para controlar las salidas. Este diseño permite un control preciso y cíclico del semáforo, demostrando la aplicación de temporizadores y lógica secuencial en sistemas de automatización.

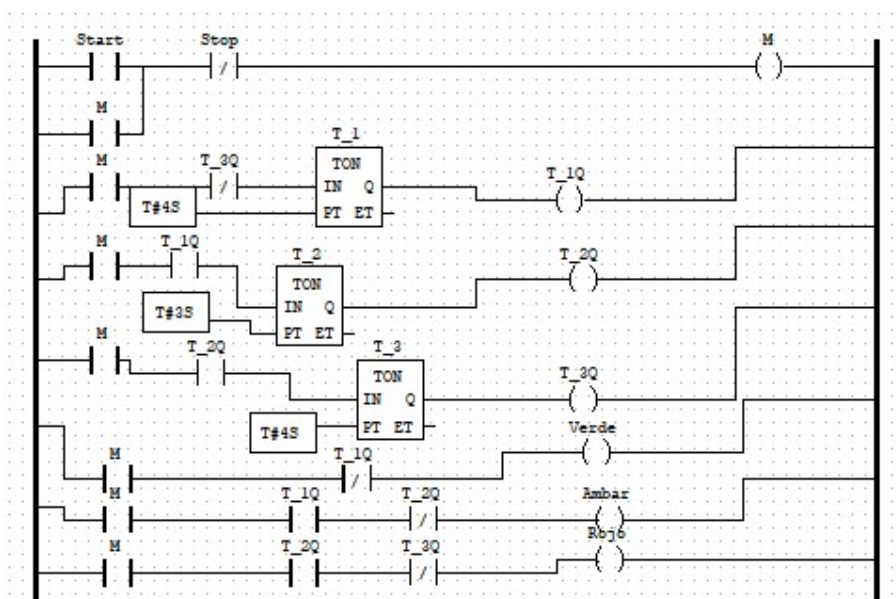


Figura 48. Programa de control del semáforo una vía.

La Figura 49 muestra las variables utilizadas para el diagrama de programación en lenguaje Ladder para controlar un Semáforo.

#	Nombre	Clase	Tipo	Ubicación
1	Start	Local	BOOL	%IX0.0
2	Stop	Local	BOOL	%IX0.1
3	M	Local	BOOL	%MX0.0
4	Verde	Local	BOOL	%QX0.0
5	Ambar	Local	BOOL	%QX0.1
6	Rojo	Local	BOOL	%QX0.2
7	Contactor	Local	BOOL	%QX0.3
8	T_1	Local	TON	
9	T_1Q	Local	BOOL	
10	T_2Q	Local	BOOL	
11	T_3Q	Local	BOOL	
12	T_ContactorQ	Local	BOOL	
13	T_ContactorQ'	Local	BOOL	
14	T_2	Local	TON	
15	T_3	Local	TON	

Figura 49. Entradas y salidas en Open PLC.

5.3.1.5. Contador de eventos

En la Figura 50 muestra las conexiones de los dispositivos en un sistema de control de contador de eventos utilizando el PLC.

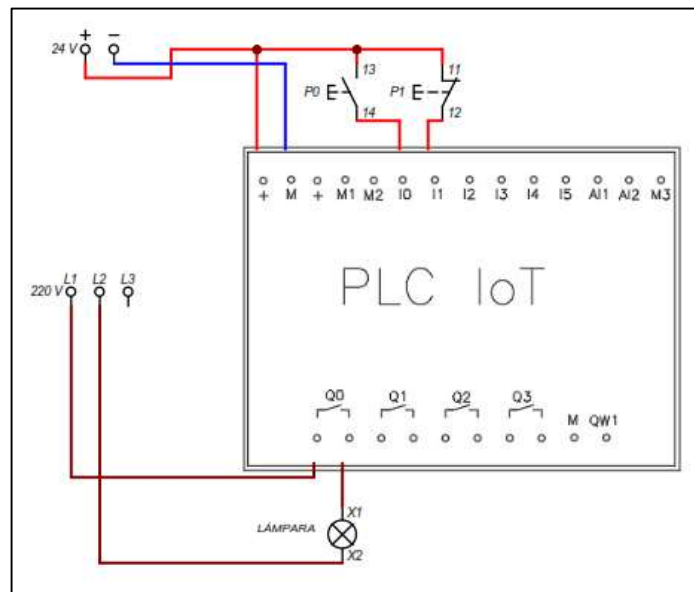


Figura 50. Diagrama de conexión contador de eventos.

La Figura 50 presenta un diagrama de programación en lenguaje Ladder para el PLC, donde PULSADOR 1 actúa como una entrada digital que incrementa un contador ascendente (CTU0) cada vez que es activado. El contador tiene un valor preestablecido (PV) que al ser alcanzado por el valor actual (CV), activa la salida LÁMPARA. Además, PULSADOR 2 funciona como una entrada de reinicio (R), que restablece el contador a cero cuando es presionado. Esta lógica permite controlar el encendido de la lámpara en función del número de pulsaciones registradas por PULSADOR 1, proporcionando un mecanismo de conteo y control basado en condiciones predefinidas.

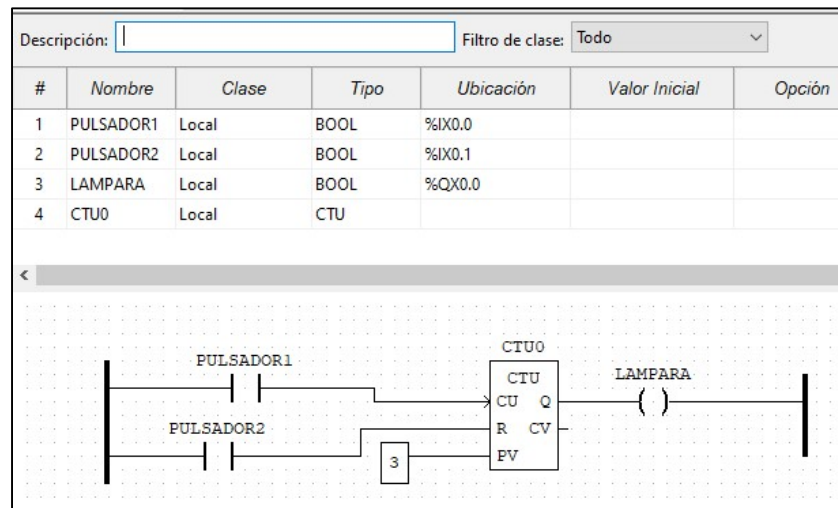


Figura 51. Programa de contador de eventos.

5.3.1.6. Control de nivel de un tanque (On/Off)

En este tipo de control se utiliza la entrada analógica para gestionar el llenado y vaciado de un tanque de agua, manteniendo el nivel del líquido dentro de un rango definido por la entrada analógica, lo cual ayuda a verificar si la entrada analógica actúa con señal transmitida por el sensor ultrasónico.

A continuación, en la Figura 51 se describe el esquema de conexión para la implementación de un sistema de control de nivel de agua en un tanque utilizando un sensor ultrasónico que se desarrolla en laboratorio de automatización y control de la Universidad Técnica de Cotopaxi.

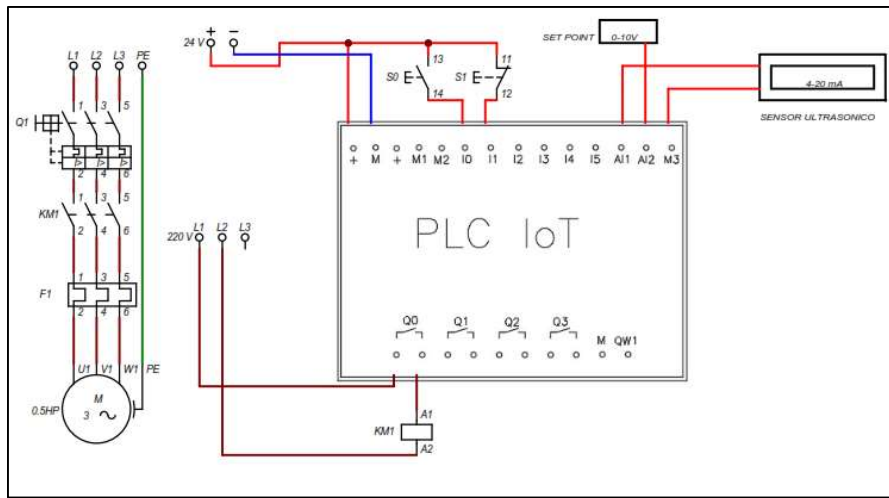


Figura 52. Diagrama de conexión control On/Off.

Para la prueba realizada se tomaron los diferentes valores de voltaje de respuesta del sensor que ingresa al PLC y se tomó también el valor para diferentes alturas (Diferentes niveles de líquido). Con los valores obtenidos se realizó una tabla verificando el funcionamiento correcto del PLC al recibir la señal transmitida por el sensor ultrasónico. Esto comprueba que el sistema es capaz de interpretar correctamente las señales del sensor para reflejar los diferentes niveles de líquido, lo que asegura la confiabilidad y precisión del sistema en condiciones.

También en la Tabla 5 se obtiene el error calculado una vez realizadas las pruebas de funcionamiento del PLC. Ecuación 1.

$$error = \frac{h_{medida} - h_{real}}{h_{real}} \times 100 \quad (1)$$

Tabla 5. Mediciones del voltaje en el sensor.

Nº de registro	Altura real [cm]	Voltaje del sensor (Multímetro) [V]	Altura medida [cm]	Error [%]
1	10	3,704	11	10
2	20	4,925	21	5
3	30	6,195	31	3.33
4	40	7,395	41	2.5
5	50	8,669	51	2
6	60	9,946	61	1.66

En la Figura 52 muestra la medida del tanque de nivel de agua de referenciada utilizada para la prueba con el sensor ultrasónico midiendo la distancia del tanque.



Figura 53. Nivel del tanque de prueba.

En la Figura 53 muestra el diagrama Ladder lo cual representa un sistema de control para el llenado de un tanque de agua. El circuito se activa con el botón Start y se desactiva con el botón Stop. Una vez encendido, se mantiene activo gracias a la realimentación del Contacto Encendido (Enclavamiento). El sistema incluye un control de llenado que compara el nivel de agua con un valor de referencia (SetPoint). Si el nivel de agua es inferior al SetPoint, se activa la salida "Llenado", iniciando el proceso de llenado del tanque. El bloque GT (Mayor que) realiza la comparación entre el nivel de agua y el SetPoint, y su salida activa la bobina Llenado cuando se cumple la condición.

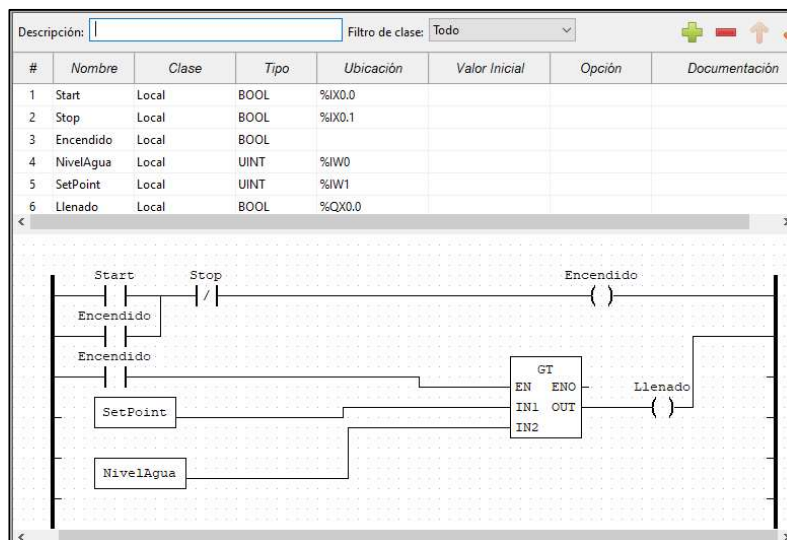


Figura 54. Programa control On/Off.

La programación realizada en el software OpenPLC, utilizando un diagrama Ladder, define el algoritmo de control destinado a la regulación del nivel de agua. Este programa

fue cargado con correctamente en el PLC, donde su ejecución demostró un funcionamiento adecuado, asegurando que las instrucciones implementadas responden de manera precisa y eficiente a las variaciones del nivel de agua, cumpliendo con los parámetros estables.

5.3.2. Aplicaciones IoT

A continuación, se desarrolló un control y monitoreo basado en IoT para una aplicación de un control PID del nivel de un tanque de agua:

5.3.2.1. Control y monitoreo mediante IoT

En esta prueba se desarrolló el control PID del nivel de un tanque de agua, en la Figura 54 se muestra un diagrama de programación en lenguaje Ladder para cargar al PLC que implementa un control PID (Proporcional Integral Derivativo) para controlar el llenado de un tanque de agua. El sistema utiliza entradas (Start) y (Stop) para iniciar y detener el proceso, mientras que (NivelAgua) mide el nivel actual del agua, convirtiendo la señal de un valor entero (UINT) a un valor real (REAL) mediante el bloque UINT_TO_REAL. El bloque PID0 ajusta el proceso de llenado comparando el valor actual (PV) con el valor deseado (SP), utilizando parámetros como KP (Ganancia proporcional), TR (Tiempo de respuesta) y TD (Tiempo derivativo). La salida del PID, convertida de nuevo a un valor entero mediante REAL_TO_UINT, controla la salida Llenado, que regula el mecanismo de llenado del tanque. Este sistema asegura un control preciso y dinámico del nivel de agua, manteniéndolo en el valor deseado.

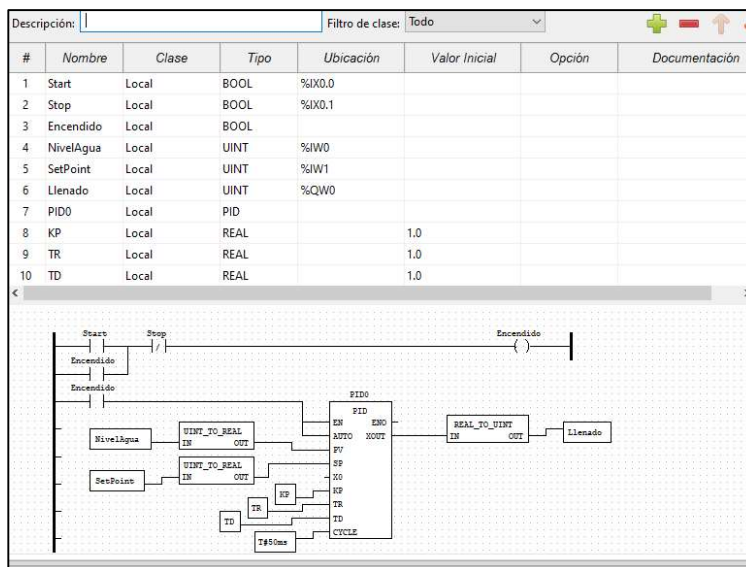


Figura 55. Programa control PID.

El control y monitoreo se ejecuta desde un celular que tenga conectividad a internet donde tenga la aplicación instalada lo que permite establecer el control de un tanque de agua permite medir en tiempo real el nivel del agua usando sensores conectados al PLC. En la Figura 55 una se muestra la información de las entradas digitales lo cual le permite al operador monitorear las entradas digitales que están activadas.



Figura 56. Monitoreo de entradas digitales.

También se puede establecer un encendido horario en función de lo que requiera el proceso para su automatización controlando las salidas del PLC como se muestra en la Figura 56.

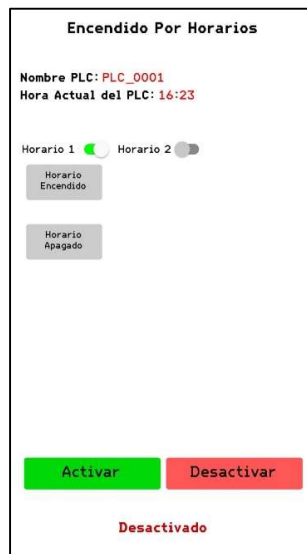


Figura 57. Monitoreo horario.

En la Figura 57 muestra una una interfaz de control para un PLC llamado "PLC_0001", diseñada para controlar y monitorear salidas digitales (01, 02, 03, 04). Estas salidas pueden activarse (ON) o desactivarse (OFF), y su estado se muestra en formato de interruptor y booleano (true/false). En este caso muestra que la salida Q1 esta activada, todas las demas salidas están apagadas (OFF).



Figura 58. Control y Monitoreo de entradas y salidas.

5.3.2.2. Envió de datos IoT

En un sistema IoT para el control del nivel de agua, los datos recolectados pueden abarcar múltiples variables clave para el monitoreo y la gestión eficiente del recurso, En la Figura 58 muestra combinación del nivel de agua (Voltaje), la fecha y hora de medición, proporciona una visión completa del estado del sistema y permite tomar decisiones informadas para optimizar su funcionamiento.

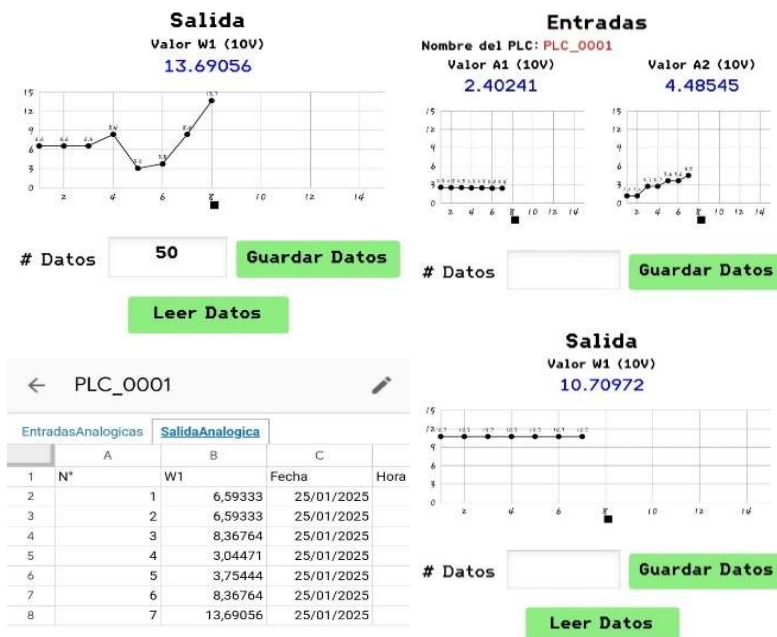


Figura 59. Monitoreo en tiempo real.

5.3.3. Evaluación del Prototipo en Laboratorio con Estudiantes

Como evaluación final del PLC con capacidades IoT, se procede a realizar una práctica con estudiantes de electricidad de la UTC que actualmente toman la materia de control industrial. Posteriormente a la práctica se pedirá a los estudiantes llenar una encuesta presentada en el Anexo 13.

5.3.3.1. Descripción de la práctica de laboratorio

Para la práctica se pretende que los estudiantes conozcan físicamente y se familiaricen con los elementos que integran el PLC con capacidades IoT y poder realizar procesos de control con el mismo.

Las actividades a desarrollar en la práctica son

- Identificación de las terminales del PLC y sus conexiones de entradas y salidas.
- Cableado de los actuadores (luces del semáforo) a las salidas del PLC.
- Conexión de la fuente de alimentación de 24VDC al sistema.
- Establecer comunicación entre el PLC y una aplicación móvil para supervisión remota.
- Verificación de conexiones y pruebas de continuidad antes de energizar el sistema.
- Carga del programa desarrollado en OpenPLC para el control del semáforo.
- Uso del IoT para monitoreo de la práctica y control manual de salidas para encender o apagar las luces del semáforo.
- Encuesta realizada a los estudiantes acerca de la práctica.

La guía completa de la práctica se adjunta en el Anexo 12 donde se detallan los preparativos y las actividades a desarrollar se muestran detalladamente.

5.3.3.2. Resultados y análisis

Mediante la encuesta se valorará el criterio de los estudiantes con respecto al uso del software y hardware que involucran el PLC con capacidades IoT. La primera pregunta permite conocer el nivel de conocimiento trabajando con PLCs.

La gráfica obtenida se muestra en la Figura 59, donde la mayoría de estudiantes recién están aprendiendo del tema.

¿Cual es tu nivel de experiencia con PLCs?

6 respuestas

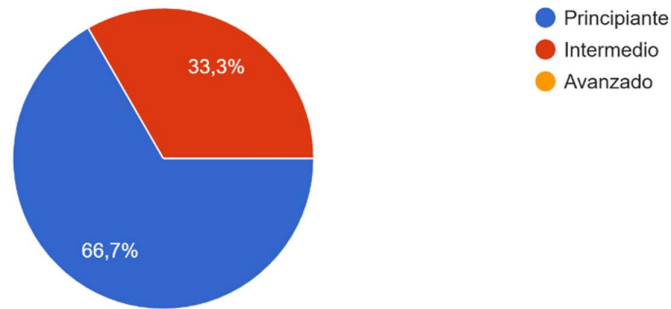


Figura 60. Gráfica nivel de experiencia en PLCs.

Las siguientes preguntas tratan sobre la experiencia en el uso del software, hardware e IoT.

- **Manejo del software OpenPLC.**

Las siguientes preguntas están directamente relacionadas con el uso del programa OpenPLC que permitirá diseñar los programas para posteriormente cargarlos al PLC.

La primera pregunta demostró una dificultad media en instalar el programa, lo que es muy bueno comparado a otros programas más pesados y complejos de instalar.

¿Cómo calificarías la facilidad de instalación y configuración del software OpenPLC?

6 respuestas

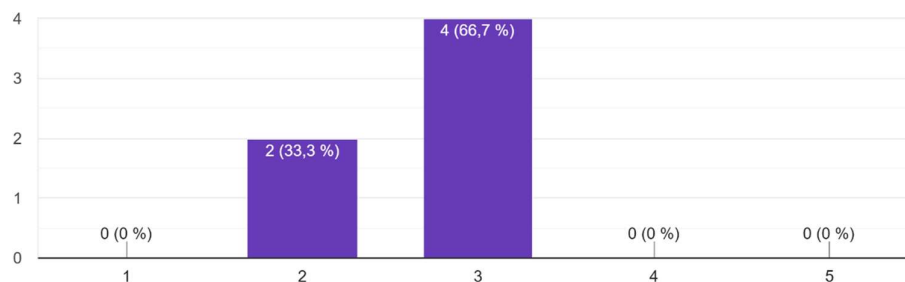


Figura 61. Gráfica sobre la dificultad de instalación de OpenPLC.

La siguiente pregunta busca conocer la dificultad en la comprensión del interfaz, mostrando un resultado medio, lo que es muy bueno sabiendo que es un software completamente nuevo para el estudiante.

¿Cómo evaluarías la interfaz gráfica del software OpenPLC?
6 respuestas

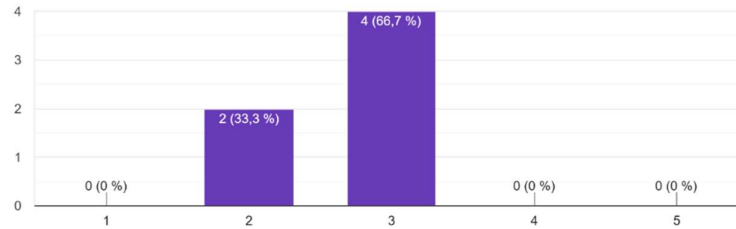


Figura 62. Gráfica sobre la dificultad de comprensión del interfaz.

A continuación está una pregunta cerrada para conocer si la declaración de variables y el uso de bloques es parecido al software que han utilizado antes para programar el PLC. Mostrando que prácticamente similar ya que OpenPLC maneja la normativa IEC 61131-3, siendo la misma de la mayoría de softwares que realizan el mismo propósito.

¿La declaración de variables y el uso de bloques es similar al software que has utilizado antes?
6 respuestas

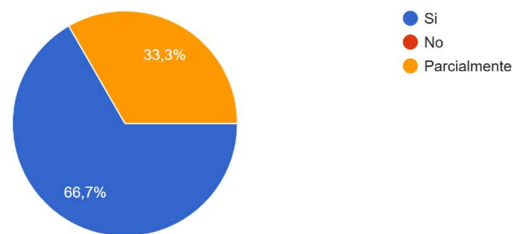


Figura 63. Gráfica sobre la similitud de variables y bloques con otro software utilizado antes.

Un detalle muy importante a la hora de compilar el programa para subirlo al PLC, es la facilidad de realizarlo con este software. En la Figura 63 se muestra la gráfica las respuestas a esta pregunta, demostrando que fue muy fácil cargar el programa.

¿Fue fácil cargar el programa al PLC?
6 respuestas

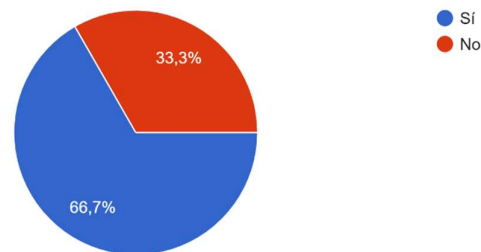


Figura 64. Gráfica sobre la facilidad de cargar el programa al PLC.

Todo software requiere contar con una documentación completa y clara, que ayudará a los usuarios a manejar fácilmente el programa. La siguiente pregunta solicita conocer la dificultad de la comprensión de estos documentos sobre OpenPLC, obteniendo una

moderada facilidad, esto podría ser debido a que la documentación, videos y foros se encuentran todos en inglés.

¿Qué difícil fue la comprensión de la documentación/tutoriales del OpenPLC?
6 respuestas

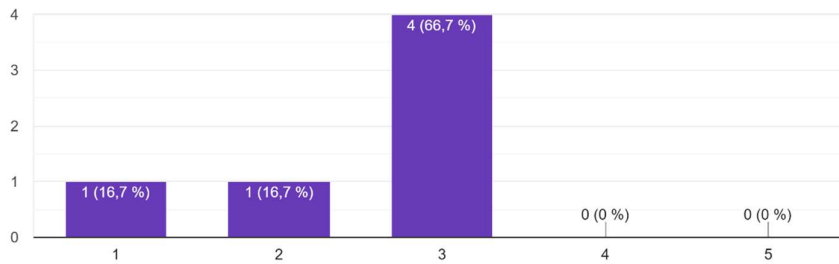


Figura 65. Gráfica sobre la comprensión de la documentación de OpenPLC.

- **Manejo del hardware**

La parte que más se requiere cuidado, es el manejo del hardware del PLC, que involucra la alimentación del y todas las conexiones para el sistema que se requiera a realizar.

La primera pregunta trata sobre la facilidad de realizar las conexiones y configurar el PLC, para que funcione con el programa y sus capacidades IoT. Las respuestas muestran una facilidad moderada con una tendencia a fácil, esto debido a las borneras extraíbles que cuenta el PLC y un atornillado sencillo con los cables.

¿Fue fácil conectar y configurar el PLC?
6 respuestas

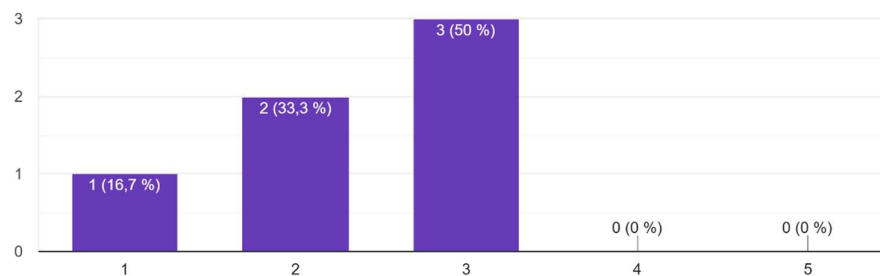


Figura 66. Gráfica de resultados sobre la facilidad de puesta a marcha del PLC.

La siguiente pregunta demuestra una gran facilidad en identificar las entradas y salidas del PLC, debido a que maneja un sistema de conexión similar a la mayoría de los PLCs comerciales, además que cuenta con indicadores en la parte superior para facilitar la comprensión.

¿Que tan difícil fue familiarizarte con las entradas y salidas del PLC?

6 respuestas

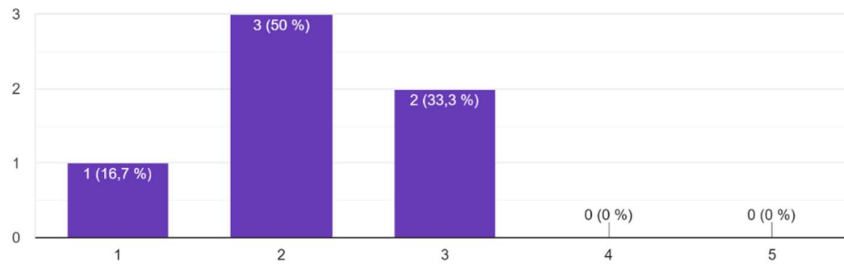


Figura 67. Gráfica sobre la dificultad en entender las entradas y salidas del PLC.

Un aspecto fundamental es que el PLC responda correctamente al programa cargado, en las respuestas de los estudiantes se evidencia un correcto funcionamiento del PLC, sin tener ninguna novedad.

¿Crees que el PLC respondió correctamente a las prácticas realizadas?

6 respuestas

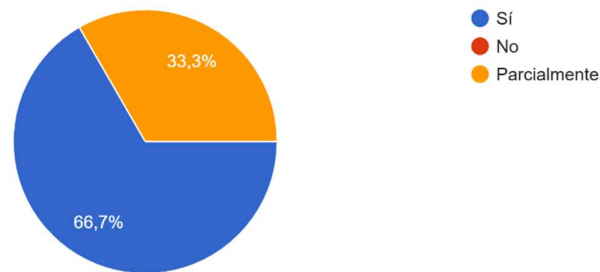


Figura 68. Gráfica sobre la respuesta del PLC en las prácticas.

Otra parte importante es saber que opinan los alumnos sobre el estado físico del PLC, demostrando que les parece adecuado para utilizarlo en entornos industriales.

¿Consideras que el hardware es robusto y confiable para aplicaciones industriales?

6 respuestas

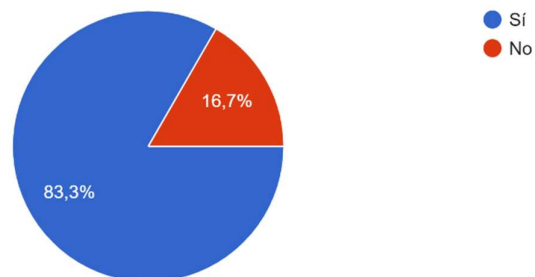


Figura 69. Gráfica sobre la opinión en la robustez el PLC.

La última pregunta trata sobre como el manual ayudó a entender el PLC para poder realizar las conexiones y cargar el programa al PLC correctamente. Las respuestas

mostradas en la Figura 69 nos demuestra un excelente manual para los usuarios que requieran su uso.

¿El manual fue suficiente para entender el hardware del PLC?
6 respuestas

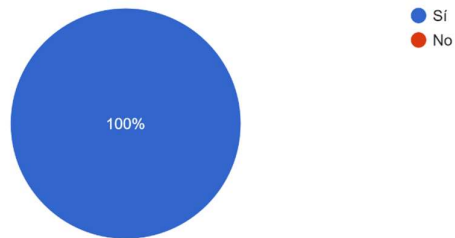


Figura 70. Gráfica sobre la comprensión del PLC usando el manual.

- **Uso del IoT.**

Las últimas preguntas tratan sobre el uso de la aplicación para el celular y que tan amigable es el interfaz de la misma.

La primera pregunta trata sobre la dificultad en la instalación de la aplicación, lo que resulta muy fácil ya que solo se instala como una aplicación cualquiera en el celular y ya estaría lista.

¿Fue fácil la instalación de la aplicación móvil?
6 respuestas

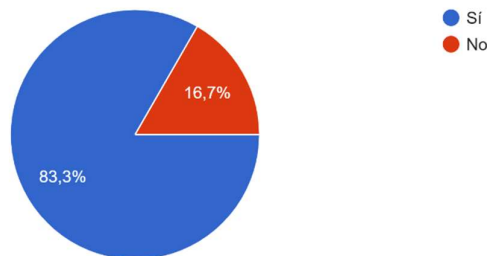


Figura 71. Gráfica sobre la facilidad en la instalación de la App.

A continuación, en la Figura 71 se pregunta sobre qué tan amigable resulta la interfaz de la App para el uso de las capacidades IoT. Se trabajó en una aplicación que contenga un menú simple y una fácil visualización de los parámetros del PLC, lo que les resultó muy fácil comprender a los estudiantes.

¿Te parece amigable el interfaz de la aplicación?

6 respuestas

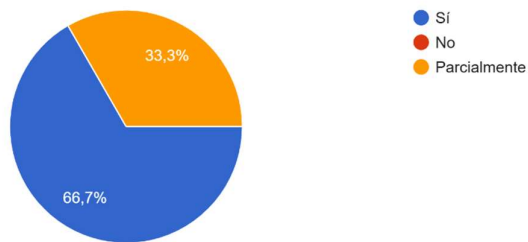


Figura 72. Gráfica sobre el entendimiento del interfaz de la App.

En esta pregunta se pedirá a los estudiantes responder que tan fácil fue conectar el PLC a la red WiFi. Esta acción les resultó muy sencilla debido a la librería utilizada en el código del ESP32, esta cuenta con un interfaz que te busca automáticamente la red y solo necesitas poner la contraseña y ya el PLC procede a conectarse y recordar esta red.

¿Qué tan fácil fue conectar el PLC a la red y transmitir datos?

6 respuestas

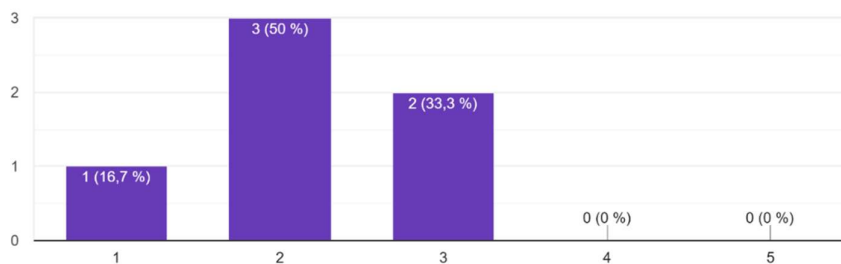


Figura 73. Gráfica de barras sobre la facilidad de conectar el PLC a la red.

El tiempo de respuesta puede resultar crítico para ciertas aplicaciones de automatización, en la Figura 73 se muestra la opinión de los estudiantes acerca del tiempo de respuesta de las acciones IoT que tiene el PLC. Para la mayoría les parece que el retardo es regular, esto debido a que el programa cargado en el ESP32 cuenta con un *delay* por defecto para realizar todas las funciones que tiene el PLC, esto solo puede representar un retarde de un par de segundos.

¿Cómo calificarías el tiempo de respuesta en la visualización y ejecución de datos en la App?

6 respuestas

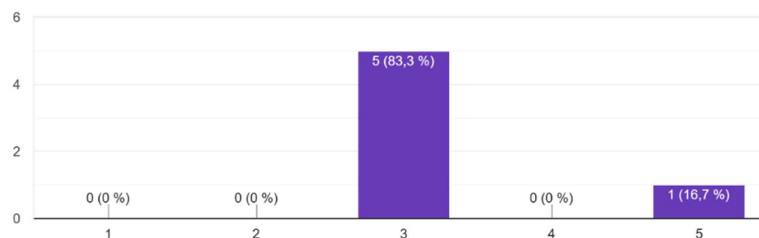


Figura 74. Gráfica sobre el tiempo de respuesta del PLC.

La última pregunta demuestra que la implementación del IoT en los PLCs para el uso en procesos industriales les parece muy necesario por los estudiantes, esto debido al rápido acceso que se tiene al PLC desde cualquier parte, y con un dispositivo que casi todo el mundo tiene en su bolsillo.

¿Consideras que el uso de IoT en el PLC es útil para aplicaciones industriales reales?
6 respuestas

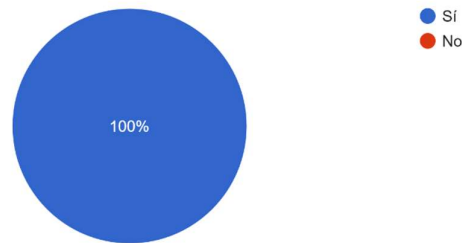


Figura 75. Gráfica sobre el uso del IoT en entornos industriales.

5.4. IDENTIFICACIÓN DE ÁREAS DE MEJORA

Durante el desarrollo del proyecto de un controlador lógico programable con capacidades IoT, se identificaron varias áreas de mejora y se omitieron justificadamente capacidades adicionales que podrían ayudar en la funcionalidad del sistema. A continuación, se detallan algunos aspectos que podrían potenciar el proyecto en futuras versiones.

5.4.1. Posibilidad de añadir un protocolo de comunicación Modbus

Durante las etapas iniciales del desarrollo del PLC, se exploró la posibilidad de incorporar el protocolo Modbus para el sistema de comunicación. Finalmente, esto no se incluyó porque el proyecto se centró en un mundo inalámbrico utilizando técnicas más modernas, como WiFi y computación en la nube. Tales soluciones tienen beneficios distintivos al ser más flexibles, accesibles y escalables.

A pesar de ser uno de los estándares más comunes en la industria, Modbus fue creado principalmente para sistemas cableados, lo que conlleva algunas limitaciones físicas y requiere hardware adicional, como módulos RS-485. En un entorno IoT, donde la conectividad remota es crítica y se espera que se rompan las barreras físicas, usar un protocolo cableado no era una opción viable.

Una solución inalámbrica fue la tecnología elegida para reducir la complejidad y el costo del hardware, al mismo tiempo que se permitía un sistema nuevo, más flexible y evolucionado. Al integrar tecnologías como Firebase, logramos una comunicación en

tiempo real desde cualquier lugar con acceso a internet, cumpliendo con todos los requerimientos sin comprometer la funcionalidad.

Por estas razones, se determinó que el protocolo Modbus no era necesario en el desarrollo del proyecto, ya que su inclusión habría añadido restricciones que contradicen el sentido de un sistema IoT, que su propósito es evitar limitaciones físicas.

5.4.2. Optimización del consumo energético

Como medio principal para la implementación del IoT se agregó el microcontrolador ESP32 que es ajeno al Arduino Nano que contendrá el código del programa que se realice, se podría agregar un único microcontrolador que contenga todo para disminuir considerablemente el consumo energético, pero sería muy frustrante su reprogramación constante cada vez que se requiera, ya que deberá contener el programa base para su funcionamiento y el que se realice en el software OpenPLC.

Para realizar un fácil armado de la placa y la posibilidad de realizar sustituciones de elementos averiados en un futuro, se omitió el uso de elementos de montaje superficial (SMD) que tienen un tamaño y consumo energético mucho menor a los de montaje por orificio pasante que se usaron para realizar la placa.

Se utilizaron relés electromecánicos en las salidas digitales del PLC por su manejo de altas corrientes y la facilidad de obtenerlos en el mercado a comparación de los de estado sólido. Su consumo energético es mayor y emite un sonido en cada activación, cosa que no sucede con los relés de estado sólido que funcionan con dispositivos semiconductores.

5.4.3. Incremento de la seguridad en la comunicación IoT

A la hora de realizar las pruebas de funcionamiento y como primera versión del prototipo se debería incrementar la seguridad en la comunicación IoT mediante Firebase, se pueden implementar varias medidas clave. En primer lugar, es viable utilizar Firebase *Authentication* para garantizar que solo usuarios o dispositivos autorizados puedan acceder a los datos. Esto puede lograrse mediante métodos robustos de autenticación, como correo electrónico y contraseña, o mediante autenticación a través de plataformas como Google, Facebook o Apple. Además, es necesario establecer reglas de seguridad en Firebase para limitar el acceso a los datos, garantizando que solo los usuarios autenticados y con los permisos adecuados puedan leer o modificar la información en las bases de datos.

5.4.4. Escalabilidad del sistema

Se podría diseñar un sistema compatible con Firebase para permitir que el sistema se adapte a mayores demandas o se integre con otros dispositivos y plataformas en el futuro.

Esto daría hincapié al desarrollo de otros dispositivos que interactúen con el PLC, como también se podría crear otras maneras de comunicarse con el mismo, ya sea mediante la web, control remoto, programas de computadora, Alexa, sistemas Scada, etc.

5.5. ANÁLISIS DE COSTOS

En este apartado se presenta un análisis detallado de los costos involucrados en el desarrollo e implementación del PLC con capacidades IoT.

5.5.1. Costos de materiales

En la Tabla 5 se muestran todos los costos asociados a los materiales utilizados para el desarrollo del prototipo.

Tabla 6. Costo de materiales para la construcción del PLC.

N.º	Detalle	Cantidad	Precio Unitario	Total
1	Microcontrolador Arduino Nano	1	\$ 10.00	\$ 10.00
2	Microcontrolador ESP32	1	\$ 11.00	\$ 11.00
3	Relé 24VDC 10A 5 pines	4	\$ 1.25	\$ 5.00
4	LM2596 Modulo Regulador Step Down	1	\$ 2.50	\$ 2.50
5	ULN2803	2	\$ 0.90	\$ 0.90
6	74LS245 Transceptor de Bus Octal	1	\$ 1.75	\$ 1.75
7	74LS14 NOT	1	\$ 1.10	\$ 1.10
8	PC817 Optoacoplador	5	\$ 0.55	\$ 2.75
9	LM358 Amplificador Operacional	1	\$ 0.60	\$ 0.60
10	Diodo Zener 10V 1W	2	\$ 0.25	\$ 0.50
11	Porta Fusible para Placa de 20mm	1	\$ 0.50	\$ 0.50
12	Fusible de Vidrio Pequeño 1A 250V 20mm	1	\$ 0.15	\$ 0.15
13	1N4007	1	\$ 0.10	\$ 0.10
14	Resistencias de carbón 1/4w	37	\$ 0.03	\$ 1.10
15	Capacitor Cerámico 0,1uF 100nF (104)	6	\$ 0.10	\$ 0.60
16	Condensador Electrolítico 100uF 16V	2	\$ 0.18	\$ 0.36
17	Pulsador 2 Pines De Chasis para T.V.	2	\$ 0.25	\$ 0.50
18	Bornera De Terminal Ángulo Recto 6 pines	1	\$ 0.80	\$ 0.80
19	Bornera De Terminal Ángulo Recto 2 pines	6	\$ 0.60	\$ 3.60
20	Regleta Simple Hembra 40 Pines 2mm	2	\$ 0.50	\$ 1.00
21	PCB impresa	1	\$ 4.00	\$ 4.00
22	Caja impresa en 3D con filamento ABS	1	\$ 8.00	\$ 8.00
TOTAL				\$ 56.81

5.5.2. Costo de desarrollo

El costo de desarrollo es el tiempo dedicado a diseñar, ensamblar, programar y probar el prototipo del sistema PLC con capacidades de IoT. Este análisis incluye tanto la actividad directa asociada con la fabricación de hardware como las actividades relacionadas con la programación y configuración del sistema.

La Tabla 5 muestra una división tentativa de las horas de trabajo, así como los datos de costo para cada actividad:

Tabla 7. Costos de desarrollo del PLC.

N°	Actividad	Tiempo estimado	Costo por hora	Costo total
1	Diseño del esquema del sistema	10 horas	\$8.00	\$80.00
2	Diseño y fabricación de la PCB	4 horas	\$8.00	\$56.00
3	Ensamblaje de componentes y pruebas	6 horas	\$8.00	\$48.00
4	Programación del controlador (Arduino Nano y ESP32)	24 horas	\$8.00	\$192.00
5	Integración de la comunicación IoT (Firebase, WiFi)	48 horas	\$8.00	\$384.00
6	Pruebas de funcionamiento y ajustes	10 horas	\$8.00	\$80.00
7	Documentación	4 horas	\$8.00	\$32.00
TOTAL				\$ 872

5.5.3. Gastos totales del PLC

En la Tabla 7 se muestran el costo del valor total del desarrollo del PLC.

Tabla 8. Gastos totales del PLC.

Gastos indirectos totales	\$ 872
Gastos directos totales	\$ 56.81
Total	\$ 928.81

Estos valores indican la totalidad de la inversión requerida para la implementación del PLC, incluyendo tanto los costos directos de los materiales y componentes empleados, como los costos indirectos asociados a la logística, pruebas y otros recursos vitales para su desarrollo.

6. CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

- Los componentes adoptados para elaborar el PLC con capacidades IoT fueron seleccionados según los factores tecnológicos aleatorios (compatibilidad, rendimiento, escalabilidad y costo). Además, se empleó software de libre distribución para garantizar que el sistema sea asequible y adaptable. Dentro de partes seleccionadas se incluyen microcontroladores o microprocesadores adecuados para IoT, sensores, actuadores, módulos de comunicación inalámbrica (Wi-Fi, Bluetooth, etc.) y herramientas de software de código abierto para la programación y gestión del sistema.
- El prototipo del PLC con capacidades de IoT fue comprobado en diversas aplicaciones, demostrando su desempeño y efectividad en escenarios reales. Se llevaron a cabo ensayos para asegurar la adecuada comunicación entre dispositivos, la realización de tareas de automatización y la habilidad para supervisar y controlar a distancia. Los hallazgos confirmaron que el sistema es factible y puede ajustarse a diferentes requerimientos.
- Se elaboraron una guía de laboratorio y un manual de manejo para el software libre OpenPLC y el módulo del PLC, complementando así la guía de manejo. Esto garantiza un correcto funcionamiento y simplifica el análisis del control y progreso de las prácticas.
- Al contrastar las alturas reales con las medidas, se notó que el error se reduce conforme se incrementan los valores de altura. El error inició en un 10% para la medición más baja y disminuyó a un 1.67% para la más alta, con una media de error de cerca del 4.08%. Esto señala que, pese a las variaciones, el sistema demuestra un rendimiento óptimo en cuanto a exactitud, particularmente en las mediciones de mayor magnitud.

6.2. RECOMENDACIONES

- Para mejorar la seguridad en la transferencia de datos entre dispositivos IoT, se sugiere implementar protocolos de cifrado que protejan la transmisión de información desde ESP32 a Firebase. Antes de enviar cualquier información, también se debe considerar configurar la autenticación mediante claves API para evitar accesos no autorizados.

- Se recomienda diseñar una versión modular del PLC para asegurar su capacidad de expansión y así no tener problemas con el rendimiento del sistema al aumentar entradas y salidas. Por lo tanto, si su aplicación de automatización es más compleja y exigente, debería optar por un microcontrolador con mayor capacidad de procesamiento.
- Para simplificar la experiencia del usuario, se ha estilizado el desarrollo de una versión más completa de la aplicación que se pueda instalar en dispositivos móviles, tales como la generación de informes históricos, alertas, gráficos personalizados, etcétera.
- Con el objetivo de proteger el PLC de factores externos como humedad, líquidos, interferencias electromagnéticas y golpes, se recomienda instalarlo en un tablero adecuado según el entorno en el que se utilizará.

7. REFERENCIAS

- [1] M. T. Karl-Heinz John, IEEE 61131-3:Programming Industrial Automation Systems.
- [2] M. C. J. M. A. L. M. S. Cano Padrón, “Desarrollo de un Controlador Lógico Programable (PLC) para aplicaciones universales aplicando innovación frugal con un enfoque en PYMES y sector doméstico,” IBERO PUEBLA, pp. 1–5, 2018.
- [3] “LDmicro: Ladder Logic for PIC and AVR.” Accessed: Jul. 20, 2024. [Online]. Available: <https://cq.cx/ladder.pl>
- [4] “¿Qué es un PLC y cómo funciona? – Industrias GSL.” Accessed: Jul. 20, 2024. [Online]. Available: <https://industriasgsl.com/blogs/automatizacion/que-es-un-plc-y-como-funciona>
- [5] Mayol Albert, Autómatas Programables. BARCELONA, 1988.
- [6] “¿Para qué sirve un PLC? - AUTYCOM.” Accessed: Jul. 20, 2024. [Online]. Available: <https://www.autycom.com/para-que-sirve-un-plc/>
- [7] “Siemens.” Accessed: Jul. 20, 2024. [Online]. Available: <https://www.siemens.com/global/en.html>
- [8] “Ventajas y desventajas de la automatización industrial | Análisis profundo.” Accessed: Jul. 20, 2024. [Online]. Available: <https://automatizacionindustrial360.com/ventajas-y-desventajas/>

- [9] P. A. Daneri, PLC Automatización y Control Industrial, vol. 1. Buenos Aires, 2008.
- [10] “PLC.” Accessed: Jul. 20, 2024. [Online]. Available: http://blog.espol.edu.ec/johnvilla/?wmp_theme_mode=desktop
- [11] T. L. T. E. Zapata Mireya, Fundamentos de Automatización y Redes Industriales, vol. 1. 2021.
- [12] M. Bartoň, R. Budjač, P. Tanuška, P. Schreiber, and T. Horák, “Industry Communication Based on TCP/IP Protocol,” Research Papers Faculty of Materials Science and Technology Slovak University of Technology, vol. 29, no. 49, 2021, doi: 10.2478/rput-2021-0025.
- [13] P. Ing and N. Molinari, “Curso sobre Controladores Lógicos Programables (PLC).,” Molinari Norberto, vol. 12, 2018.
- [14] Moreno M, CONTROLADOR LOGICO PROGRAMABLE (PLC). Buenos Aires.
- [15] “Software | Arduino.” Accessed: Jul. 20, 2024. [Online]. Available: <https://www.arduino.cc/en/software>
- [16] “Autonomía: software PLC de código abierto.” Accessed: Jul. 20, 2024. [Online]. Available: <https://autonomylogic.com/>
- [17] “Firebase | Plataforma de desarrollo de aplicaciones web y móviles de Google.” Accessed: Jul. 20, 2024. [Online]. Available: <https://firebase.google.com/?hl=es>
- [18] “EasyEDA – Simulador de circuitos y diseño de circuitos impresos online.” Accessed: Jul. 20, 2024. [Online]. Available: <https://easyeda.com/es>
- [19] B. Op, S. Fmt, S. Rp, and A. Fl, “System integration between PLCs and IoT devices System integration between PLCs and IoT devices,” CONGRESSO BRASILEIRO DE INSTRUMENTAÇÃO, SISTEMAS E AUTOMAÇÃO, no. October, 2019, doi: 10.17648/cobisa-2019-101098.
- [20] Y. J. Lin, C. F. Tan, and C. Y. Huang, “Integration of logic controller with IoT to form a manufacturing edge computing environment: A premise,” in Procedia Manufacturing, 2019. doi: 10.1016/j.promfg.2020.01.383.