



**UNIVERSIDAD ESTATAL PENINSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN**

TRABAJO DE INTEGRACIÓN CURRICULAR

Propuesta tecnológica previo a la obtención del Título de:

INGENIERA EN ELECTRÓNICA Y AUTOMATIZACIÓN

**“MONITOREO Y CONTROL DE UN PROCESO DE FLUJO Y NIVEL CON
TECNOLOGÍA IIoT BASADO EN SOFTWARE LIBRE”**

AUTOR:

CHASI ZAMBRANO THAIZ FERNANDA

PROFESOR TUTOR

ING. ÓSCAR W. GÓMEZ MORALES PhD.

LA LIBERTAD-ECUADOR

2025

AGRADECIMIENTO

A mis padres; a mi papá sin él no podría haber logrado nada, gracias, papá por tu paciencia, a mi madre por acompañarme en este viaje, este logro es más suyo que mío.

A mis hermanas; a Sheyla por sostenerme todas las veces que quise renunciar y siempre recordándome lo capaz que soy, a Mishel por siempre sentirse orgullosa de mí y siempre darme palabras de aliento.

A mi sobrina Arianna Edith que, aunque ella no lo sepa fue motivo de nunca rendirme, Uds. mi familia fueron el motor indispensable para que yo logre esto.

Al Ing. Lizandro Tomala por haberme brindado su conocimiento en la realización de este proyecto, gracias por no desistir y ayudarme.

A mis amigos los ingenieros Erick y Guillermo por siempre velar por mí y estar pendientes en todo este proceso brindándome su mano para que yo no desistiera.

A mis amigas la Ing. Valeria Aguirre que nunca dudo en enseñarme algo cuando no entendía y a Belén Vallejo que siempre estuvo conmigo en este proceso dándome ánimos y nunca me dejo caer eres parte fundamental de mi vida gracias.

A Kevin Tigrero el primer amigo que hice en esta ciudad que me acompañó en cada tropiezo dándome ánimos, Uds. mis amigos que se convirtieron en mi paño de lágrimas estando lejos de casa y hacerme sentir que encontré mi familia, este logro también es suyo.

A Karla Inés Mariela y todos los amigos que hice en el camino a quienes me conocen de años y con quienes coincidí en estos últimos momentos gracias por las risas y los momentos que me hicieron olvidar que estaba sola en esta ciudad.

A Grace Torres por siempre confiar en mis capacidades y ser un espejo de la mujer que algún día quiero ser.

A todo aquel a quien acudí cuando no entendía alguna clase o necesitaba ayuda y nunca dudo en aclararme alguna duda y apartar de su tiempo un espacio para enseñarme a cada uno de Uds. les pertenece un pedacito de este logro.

Al Ing. Óscar Gómez quiero expresar mi más sincero agradecimiento por su gran flexibilidad y por la calidad humana que demostró en todo este proceso gracias por acompañarme con tanta paciencia y empatía, su apoyo me permitió seguir adelante.

DEDICATORIA

Este trabajo está dedicado antes que, nadie a mi padre, el Ing. Edison Chasi, quien jamás dejó de creer en mí, por tu paciencia y por ser esa fuerza que me sostuvo cuando sentía que ya no podía más, eres mi mayor inspiración, mi mejor compañía y el motor que impulsó cada paso de este camino, este nuevo peldaño en mi vida te pertenece, sin ti nada de esto sucediera.

Dedico también este logro a mi fiel compañera en esta travesía, mi Mazikeen, que en cada noche de desvelo estuvo ahí, acurrucada a mis pies en las madrugadas frías, acompañándome en silencio, que con su calor abrigo mi corazón y secó mis lágrimas cuando yo sentía que ya no daba más, fue mi mayor refugio en medio de tantos momentos difíciles.

A mis hermanas y a mi madre, quienes cada vez que hablan de mí lo hacen con orgullo, gracias por creer en la persona que soy incluso cuando yo lo dudo, su amor y su fe en mí han sido un abrazo constante que me devolvía la esperanza.

Y a mis amigos Belén, Erick, Guillermo, Valeria y Kevin quienes se convirtieron en mi familia lejos de casa, por su apoyo, por su cariño, por las risas y por estar presentes cuando más los necesité. Ustedes transformaron este camino en un viaje más ligero y mucho más agradable.

A todos Uds. les dedico este logro.

APROBACIÓN DEL TUTOR

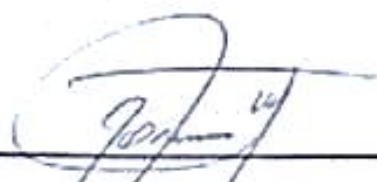
En mi calidad de tutor del trabajo de titulación denominado: “**Monitoreo y control de un proceso de flujo y nivel con tecnología IIoT basado en software libre**” elaborado por el estudiante Chasi Zambrano Thaiz Fernanda, de la carrera de Electrónica y Automatización de la Universidad Estatal Península de Santa Elena, me permito declarar que luego de haber orientado, estudiado y revisado, la apruebo en todas sus partes y autorizo al estudiante que inicie los trámites legales correspondientes.

La Libertad, 19 de noviembre del 2025



Ing. Óscar Gómez Morales, Ph.D.

TRIBUNAL DE GRADO



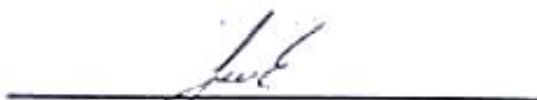
X Ing. Ronald Rovira Jurado, Ph.D.
DIRECTOR DE LA CARRERA



Ing. Carlos Saldaña Enderica, M.Sc.
DOCENTE ESPECIALISTA



Ing. Óscar Gómez Morales, Ph.D.
TUTOR



Ing. Luis Chuquimarca Jiménez, Mgtr.
DOCENTE GUÍA UIC



Ing. Corina Gonzabay de la A, Mgtr.
SECRETARIA DEL TRIBUNAL

RESUMEN

El sistema de monitoreo y control desarrollado en esta tesis se basa en la adquisición, visualización y análisis de las variables industriales de flujo y nivel mediante un módulo didáctico que integra tecnología IIoT, un microPLC Opta y una interfaz HMI, el proyecto permite obtener datos en tiempo real, procesarlos a través de software libre y enviarlos a la nube mediante una arquitectura IOT, lo que facilita el acceso remoto, la supervisión continua y la toma de decisiones basada en datos, la integración de sensores industriales, un variador de frecuencia y métodos de control ON/OFF y PID demuestra la capacidad del sistema para optimizar procesos; con ello, se evidencia cómo la incorporación de IIoT en este módulo aporta valor al aprendizaje práctico y promueve la implementación de soluciones modernas en entornos industriales.

La comunicación entre el microPLC Opta, el HMI Weintek y la plataforma IoT Cloud se realizó empleando protocolos industriales como Modbus y servicios de IoT seguros, garantizando una transmisión confiable y continua de datos. Esto permitió monitorear tendencias, validar el comportamiento dinámico del proceso y ajustar parámetros del controlador desde la interfaz, los resultados obtenidos evidencian que el sistema ofrece un control preciso del nivel del tanque, una respuesta estable bajo ambas estrategias de control y una visualización clara que facilita el análisis del proceso, el proyecto demuestra la viabilidad de implementar soluciones IIoT de bajo costo para optimizar el control de variables industriales y fortalecer la formación académica en automatización y control.

Palabras clave: IIoT, control PID, control ON/OFF, HMI, Opta PLC, software libre, monitoreo industrial.

ABSTRACT

The monitoring and control system developed in this thesis is based on the acquisition, visualization, and analysis of industrial flow and level variables through a didactic module that integrates IIoT technology, an Opta microPLC, and an HMI interface. The project allows real-time data acquisition, processing through open-source software, and cloud transmission using an IoT architecture, enabling remote access, continuous supervision, and data-driven decision-making. The integration of industrial sensors, a frequency inverter, and ON/OFF and PID control methods demonstrate the system's capability to optimize processes; consequently, the incorporation of IIoT technologies in this module enhances practical learning and promotes the implementation of modern solutions in industrial environments.

Communication between the Opta microPLC, the Weintek HMI, and the IoT Cloud platform was carried out using industrial protocols such as Modbus and secure IoT services, ensuring reliable and continuous data transmission. This allowed trend monitoring, validation of the process's dynamic behavior, and adjustment of controller parameters directly from the interface. The results show that the system provides precise control of the tank level, stable response under both control strategies, and clear visualization that facilitates process analysis. Overall, the project demonstrates the feasibility of implementing low-cost IIoT solutions to optimize the control of industrial variables and strengthen academic training in automation and control.

Keywords: IIoT, PID control, ON/OFF control, HMI, Opta PLC, open-source software, industrial monitoring.

INDICE GENERAL

AGRADECIMIENTO	2
DEDICATORIA	3
RESUMEN	6
ABSTRACT	7
INDICE GENERAL	8
INDICE FIGURAS	12
INDICE TABLAS	14
CAPÍTULO I	15
1.1 ANTECEDENTES	15
1.2 DESCRIPCION DEL PROYECTO	17
1.3 OBJETIVOS	18
1.3.1 Objetivo General	18
1.3.2 Objetivo Específicos	18
1.4 JUSTIFICACION	19
1.5 ALCANCE	20
1.6 METODOLOGIA DE LA INVESTIGACION	21
1.6.1 Investigación Fenomenológica.....	21
1.6.2 Investigación Hermenéutica.....	21
1.6.3 Investigación Práctico-Proyectual.	21
CAPÍTULO II	23
2. PROPUESTA	23
2.1 MARCO CONTEXTUAL	23
2.2 MARCO CONCEPTUAL	24
2.2.1 Automatización Industrial.....	24

2.2.2 Internet Industrial de las Cosas	25
2.2.3 Tipos de PLC:	26
2.2.4 Topologías de redes, para PLC.....	27
2.2.5 Arquitectura de redes inalámbricas con PLC a un servidor.	28
2.2.6 Interfaz Humano máquina	29
2.2.7 Principales Sistemas de Recolección de Datos en IoT:	29
2.2.8 Tipos de control para Nivel y flujo los más utilizados:	30
2.2.9 Tipos de plantas industriales que existe para la parte educativa.	31
2.2.10 Normativas aplicadas en la elaboración del módulo industrial para equipos o dispositivos.....	33
2.3 MARCO TEORICO	33
CAPÍTULO III.....	35
3. DESARROLLO DE LA PROPUESTA	35
3.1. COMPONENTES FÍSICOS	35
3.1.1 Opta Arduino Wifi.....	35
3.1.2 Sensor de flujo de tipo turbina	36
3.1.3 Transmisor de nivel KX-132V	38
3.1.4 HMI Weintek cMT3072X.....	39
3.1.5 Convertidor de frecuencia Invt Goodrive10 Series.....	39
3.1.6 Bomba de agua de 1 HP	41
3.1.7 Interruptor Automático Schneider Electric Easy9 C25.	41
3.1.8 Válvula de Bola de Latón de Paso Total para Agua.....	42
3.8.9 Switch de red TP-Link de 8 puertos	43
3.8.10 Elementos de control y señalización.....	43
3.2 COMPONENTES LÓGICOS	44
3.2.1 SolidWorks	44
3.2.2 IDE de PLC de Arduino.....	44

3.2.3 IoT CLOUD.....	45
3.2.4 EasyBuilder Pro	46
3.3 DISEÑO DE LA PROPUESTA.....	46
3.3.1 Diseño estructural y del tanque (Modelado 3D en Solidwork).....	46
3.3.2 Diseño Físico del Módulo.....	48
3.3.3 Construcción y ensamble del panel del control del sistema	49
3.3.4 Diseño diagramas de conexión.....	51
3.3.4.1 Diagrama de redes Modbus TCP I/P.....	51
3.3.4.2 Diagrama de redes Modbus RS485	52
3.3.4.3 Diagrama de control	53
3.3.4.4 Diagrama de potencia	54
3.3.5 Diagrama de tuberías e instrumentación (P&ID).....	56
3.3.6 Cableado diagramas de potencia y control.....	57
3.3.7 Montaje tubería	58
3.3.8 Diseño interfaz HMI.....	59
3.3.9 Partición de memoria del microPLC Arduino Opta.....	68
3.3.10 Actualización de firmware del PLC Arduino Opta	69
3.3.11 Conexión MODBUS	69
3.3.11.1 Conexión Modbus RS-485 entre el PLC y el variador de frecuencia	70
3.3.11.2 Conexión Modbus TCP/IP entre el PLC y HMI	71
3.3.12 Configuración IIoT con Arduino Cloud mediante PLC IDE	72
3.3.13 Controladores aplicados en el sistema	73
3.3.13.1 Controlador PID	74
3.3.13.2 Controlador ON-OFF	76
3.3.14 Pruebas aplicadas a los controladores.....	77
3.3.14.1 Pruebas aplicadas al controlador ON/OFF	78
3.3.14.2 Pruebas aplicadas al controlador PID	81

3.3.15 Integración IIoT con Arduino Cloud mediante PLC IDE.....	83
CAPÍTULO IV.....	88
4. CONCLUSIONES Y RECOMENDACIONES	88
4.1 Conclusiones.....	88
4.2 Recomendaciones.....	89
ANEXOS.....	91
Anexo 1 código WifiFirmwareUpdater.....	91
Anexo 2 directrices <i>WifiFirmwareUpdater</i>	103
Anexo 3 Programación del PLC IDE.....	104
Anexo 4 código para integrar el controlador PID con la plataforma Arduino IoT Cloud.....	107
Anexo 5 configuracion modbus rs485 en el plc ide.....	110
BIBLIOGRAFÍA.....	113

INDICE FIGURAS

Figura 1: Diagrama general de monitoreo y control del proceso.	18
Figura 2: Diagrama de flujo metodológico	22
Figura 3: <i>Diagramas de sistemas de lazo abierto y cerrado</i>	24
Figura 4: <i>Diagrama describiendo la estructura interna de un PLC</i> [18].	25
Figura 5: Topologías de redes, para PLC [25].....	27
Figura 6: Arquitectura de redes inalámbricas [31].	28
Figura 7: <i>Interfaz humano -máquina de uso industrial</i> [33].	29
Figura 8: Modulo Labvolt [46].	31
Figura 9: <i>Modulo PlantPAx™</i> [47].	32
Figura 10: LabVolt Series 6090-A [48].....	32
Figura 11: <i>Arduino Opta WiFi</i> [51].	36
Figura 12: sensor de flujo de pasillos de acero inoxidable [53].	37
Figura 13: Transmisor de sensor de nivel de líquido KX-132V [54].	38
Figura 14: Weintek CMT3072XH. [55].	39
Figura 15: Convertidor de frecuencia Invt Goodrive10 Series [56]	40
Figura 16: Bomba de agua de 1 HP [57].	41
Figura 17: Breaker Schneider [58].	42
Figura 18: Válvula [59].	42
Figura 19: Switch de red TP-Link [60].	43
Figura 20: Elementos de control y señalización [61]	43
Figura 21: SolidWorks [62].	44
Figura 22: IDE de Plc [63].	44
Figura 23: IoT CLOUD [64].	45
Figura 24: EasyBuilder Pro [65].	46
Figura 25: Medidas y diseño modulo.	47
Figura 26: Modulo en 3D.	47
Figura 27: Tanque en 3D	48
Figura 28: Esqueleto modulo.	48
Figura 29: Estructura final del modulo.	49
Figura 30: Instalación del HMI en el panel.	50
Figura 31: Panel de control finalizado.	50
Figura 32: Diagrama de redes modbus TCP I/P.	51
Figura 33: Diagrama de redes Modbus RS485	53
Figura 34: Diagrama de control	54
Figura 35: Diagrama de potencia/alimentación.	55
Figura 36: Diagrama P&ID	57
Figura 37: Alimentación 110-220v	57
Figura 38: Conexión actuadores.	58
Figura 39: Tanque reservorio.	58
Figura 40: Tanque principal	59
Figura 41: : Pantalla principal	60
Figura 42: Pestaña login.	61
Figura 43: Ventana modo de operación	61
Figura 44: Base de datos controlador PID	63
Figura 45: Base de datos controlador On/Off	64
Figura 46: Base de datos de límites de nivel.	64
Figura 47: Ventana proceso.	65
Figura 48: Ventana gráficas.	65
Figura 49: Ventana gráficas	67

Figura 50: Selección puerto.....	68
Figura 51: Comunicación Modbus.....	69
Figura 52: Control de lazo cerrado	74
Figura 53: Tanque acrílico principal	78
Figura 54: Prueba 1	79
Figura 55: Prueba 2	79
<i>Figura 56:</i> Prueba 3.....	80
Figura 57: Parámetros	81
Figura 58: Parámetros	82
Figura 59: Parámetros	82
Figura 60: Arduino Create Agent.	83
Figura 61: Monitoreo en tiempo real	84
Figura 62: Estado Online PLC.....	85
Figura 63: Estado Online PLC.....	85
Figura 64: Interfaz Arduino Cloud	86
Figura 65: Acceso remoto	86

INDICE TABLAS

Tabla 1 . Arduino Opta WiFi.....	36
Tabla 2 Características del sensor de flujo de pasillos de acero inoxidable de 1 pulgada	37
Tabla 3 Transmisor de sensor de nivel de líquido	38
Tabla 4 Weintek CMT3072XH.....	39
Tabla 5. Invt Goodrive10 Series	40
Tabla 6. Bomba de agua de 1 HP	41
Tabla 7 Schneider Electric Easy9 C25	42

CAPÍTULO I

1. FUNDAMENTACIÓN.

1.1 ANTECEDENTES

En esta propuesta se plantea el diseño e implementación de un sistema de monitoreo y control de flujo y nivel utilizando tecnología IIoT y herramientas de software libre, para comprender el contexto en el que se desarrolla el proyecto es importante considerar cómo la Industria 4.0 ha impulsado la transformación digital en los procesos industriales, el enfoque de este integra tecnologías de información, automatización y conectividad para mejorar la comunicación y coordinación dentro de las operaciones industriales.

Uno de los pilares de esta nueva etapa industrial es el Internet de las Cosas (IoT), que permite que diferentes dispositivos, sensores y equipos intercambien datos en tiempo real, su aplicación no se limita únicamente al sector industrial; también se ha extendido a entornos domésticos, educativos y comerciales, gracias al desarrollo de sensores más accesibles, sistemas embebidos, plataformas de comunicación y dispositivos capaces de procesar información de manera autónoma.

La primera revolución industrial surgió entre 1750 y 1840 como resultado del paso de la producción manual hacia la mecanización. La aparición de maquinaria movida por vapor permitió que sectores como textiles, metalurgia, transporte y manufactura adoptaran formas de trabajo más rápidas y eficientes. La automatización básica y el uso de energía mecánica marcaron la transición desde métodos artesanales hacia procesos cada vez más organizados y repetitivos.

Posteriormente, la segunda revolución industrial se desarrolló entre finales del siglo XIX y comienzos del siglo XX, su avance estuvo impulsado por el crecimiento científico y tecnológico, así como por la expansión de nuevas industrias en países como Francia, Alemania, Estados Unidos y la URSS. la electrificación, la producción en masa y la introducción de motores más avanzados permitieron elevar los niveles de producción y mejorar la disponibilidad de bienes.

Durante esa etapa surgieron inventos que modificaron profundamente la vida cotidiana y la actividad productiva, como el motor diésel diseñado por Rudolf Diésel en 1892, el radio desarrollado por Guillermo Marconi, el automóvil creado por Karl Benz, el avión de los hermanos Wright, el telégrafo de Samuel Morse, el teléfono introducido por Graham Bell y la bombilla perfeccionada por Thomas Alva Edison.

La tercera revolución industrial, conocida como la era de la información, comenzó aproximadamente entre los años 1960 y 1990. Su impacto se relacionó con la incorporación de tecnologías electrónicas, sistemas informáticos y métodos de automatización más avanzados. La digitalización permitió optimizar procesos, reducir tiempos de producción y establecer nuevas formas de comunicación y control dentro de las empresas.

La modernización de los procesos productivos ha avanzado de manera significativa gracias a la incorporación de tecnologías orientadas a la automatización y a la autonomía de equipos, las industrias han cambiado su modelo de operación, destinando mayores recursos a la adquisición de herramientas tecnológicas en lugar de depender únicamente de mano de obra humana.

En el ámbito social y económico también se han observado efectos importantes, la brecha entre países con mayor desarrollo y aquellos con menor acceso a tecnología se ha ampliado, debido principalmente al ritmo desigual con el que cada región adopta innovaciones y transforma sus procesos.

La rápida evolución de las tecnologías de información y comunicación, junto con el esfuerzo continuo de las empresas por modernizar sus procesos, ha impulsado un cambio profundo en la manera de ver los sistemas industriales, surge así la idea de una producción más flexible, conectada y autónoma, donde las fábricas se transforman en entornos inteligentes capaces de integrar dispositivos, sensores y plataformas de análisis en tiempo real.

La digitalización industrial también ha sido impulsada por la aparición de tecnologías disruptivas que permiten optimizar recursos, mejorar el control operativo y generar productos personalizados a un costo más competitivo. Esta tendencia se ha consolidado con proyecciones sobre el crecimiento del IoT a nivel global. Diversos estudios estimaron que para 2019 el número de dispositivos conectados superaría los 30 mil millones, incluyendo equipos como teléfonos inteligentes, tabletas, computadoras y relojes electrónicos, todos con capacidad de intercambiar información mediante redes digitales.

1.2 DESCRIPCION DEL PROYECTO

Este proyecto se basa en la implementación de un sistema de monitoreo y control para un proceso de flujo y nivel aprovechando la tecnología IoT y utilizando soluciones de software libre donde la implementación de tecnología IoT permite establecer una interconexión completa entre los dispositivos encargados de la medición y control que serán los sensores encargados de medir flujo y nivel, facilitando así la captura y transmisión de datos en tiempo real entonces esta interconexión posibilita una supervisión constante del proceso lo que conlleva a una comprensión más detallada y un control efectivo de las variables críticas, tales como el flujo y nivel.

El empleo de soluciones de software libre proporcionado por el microcontrolador a usar nos facilita el uso de este, estas soluciones suelen ser altamente personalizables y ajustables lo que proporciona una mayor flexibilidad y adaptabilidad a las necesidades cambiantes del entorno industrial.

El microPLC Arduino Opta es un sistema de comunicación inalámbrica y ha jugado un papel relevante dentro del proyecto, ya que permite enlazar los módulos de medición y control con la plataforma de supervisión. Su integración facilita el envío de datos entre equipos distribuidos y reduce la necesidad de cableado extensivo, lo que mejora la instalación y el mantenimiento del sistema, además de ofrecer mayor flexibilidad en la expansión del proceso. La representación de los datos del proceso es una herramienta valiosa ya que esta visualización directa permite una respuesta inmediata ante cualquier anomalía o situación imprevista en el proceso lo que contribuye a la optimización de la eficiencia operativa que ayudan a la centralización de los datos en la nube que permite un acceso remoto a la información del proceso desde cualquier ubicación con acceso a Internet, esto posibilita una supervisión continua y en tiempo real así como una colaboración efectiva entre los diferentes equipos de trabajo y la toma de decisiones basadas en datos actualizados.

La representación de los datos del proceso es una herramienta valiosa ya que esta visualización directa permite una respuesta inmediata ante cualquier anomalía o situación imprevista en el proceso lo que contribuye a la optimización de la eficiencia operativa.

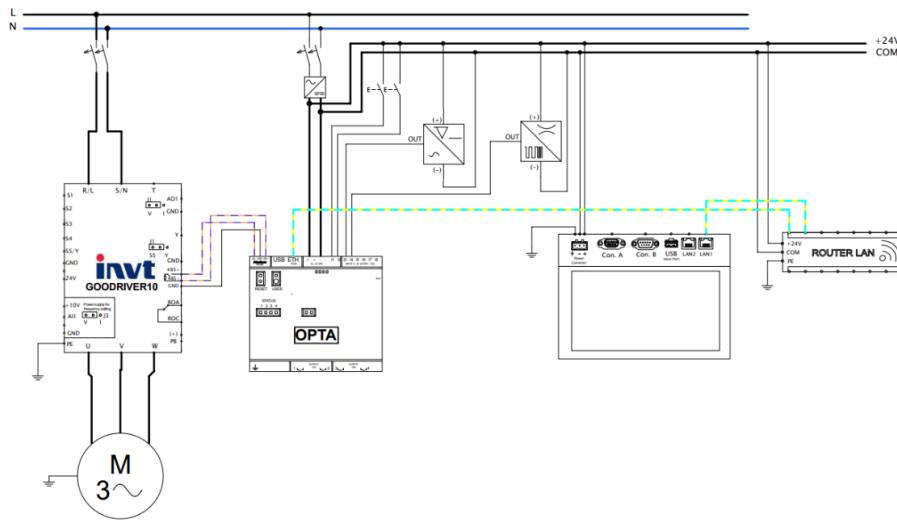


Figura 1: Diagrama general de monitoreo y control del proceso.

Fuente: T. Chasi, 2025

En la figura 1 se presenta el diagrama general del sistema, lo cual facilita la comprensión de su funcionamiento, este diagrama proporciona información de las diferentes partes y su interacción.

La combinación de tecnología IIoT y software de código abierto en este proyecto proporciona una manera completa y económica de supervisar y controlar el flujo y nivel en la industria por ende este enfoque no solo optimiza el proceso haciéndolo más preciso y eficiente, sino que también brinda mayor flexibilidad y accesibilidad lo que resulta en una mejora notable tanto en el proceso como en la calidad del sistema.

1.3 OBJETIVOS

1.3.1 Objetivo General

Implementar un sistema de monitoreo y control para un proceso de flujo y nivel mediante tecnología IIoT y soluciones de software libre permitiendo una supervisión constante un control efectivo de variables industriales, la visualización de datos en la nube y la representación de datos en tiempo real mediante HMI.

1.3.2 Objetivo Específicos

1. Diseñar e implementar un módulo de proceso de flujo y nivel
2. Configurar el microPLC OPTA con el sistema IIoT para la integración para permitir el intercambio de datos con el sistema de flujo y nivel.
3. Establecer la comunicación de las variables de flujo y nivel con el sistema IIoT.
4. Realizar pruebas y validación de la comunicación entre el microPLC OPTA con el sistema IIoT.

1.4 JUSTIFICACION

La implementación de un proceso de control y monitoreo utilizando tecnología IIoT y software libre para supervisar variables de flujo y nivel en entornos industriales nace por la necesidad de mejorar la eficiencia y la precisión en la gestión de estos procesos, esta iniciativa busca capturar y transmitir datos lo que facilitará un monitoreo continuo y detallado de las variables elegidas contribuyendo a una comprensión más profunda del proceso y a un control más preciso sobre los factores involucrados.

La selección del microPLC OPTA como componente integral del sistema de monitoreo y control destaca su importancia en el progreso de los controladores lógicos programables, el uso del microPLC representa un ahorro significativo en costos en comparación con PLCs de otras marcas establecidas además su reciente entrada en el mercado lo posiciona como un nuevo controlador prometedor.

La configuración adecuada de la conexión entre el microPLC OPTA y el sistema IIoT garantizará la integración eficiente de estos componentes en el sistema de monitoreo y control, esto se traducirá en un intercambio fluido de datos entre los diferentes dispositivos y sistemas involucrados en el proceso asegurando así la fiabilidad de la comunicación y la efectividad del sistema en la supervisión de variables de flujo y nivel.

La realización de pruebas y validación de la comunicación entre el microPLC OPTA y el sistema IIoT será fundamental para garantizar el funcionamiento correcto y confiable del sistema de monitoreo y control, estas pruebas y sus resultados darán el paso para el estudio de futuros proyectos de titulación además de afianzar conocimientos necesarios en el ámbito educativo.

La conexión del microPLC OPTA con un software libre en la nube permitirá proporcionar acceso remoto a la información del proceso desde cualquier ubicación esto facilitará la toma de decisiones informadas y la optimización de los recursos contribuyendo a una mayor eficiencia y competitividad en la industria entonces la implementación de este sistema de monitoreo y control representa un avance significativo en la modernización de los procesos industriales ofreciendo una solución nueva, accesible y colaborativa que puede impulsar la competitividad y la sostenibilidad en la industria.

1.5 ALCANCE

La finalidad de este proyecto radica en la obtención de datos de las variables industriales seleccionadas tales como el flujo y el nivel, estos datos se visualizarán a través de software libre que se integra con el sistema IIoT, junto con la implementación de una nueva opción como control lógico programable esta combinación ofrece soluciones innovadoras en la industria, facilitando nuevas investigaciones y fortaleciendo el tema mediante la incorporación de otros casos de estudio, esta iniciativa se presenta como una oportunidad para que estos resultados puedan complementar proyectos de titulación futuros.

Basándose en una investigación exhaustiva de las variables más usadas en la industria se ha decidido trabajar con dos de las más recurrentes, por ende, se llevará a cabo el diseño e implementación de un módulo para estudiar el comportamiento de estas variables industriales seleccionadas.

El módulo contará con la colaboración del microPLC OPTA para recopilar datos de las variables de flujo y nivel, estos datos serán transmitidos de forma inalámbrica a través del sistema IIoT, utilizando el Gateway como vía de transmisión que a su vez trabajará en conjunto con el software libre que permitirá visualizar los datos obtenidos.

El impacto positivo de este proyecto se manifestará tras la realización de pruebas y validación de datos los resultados obtenidos ampliarán el conocimiento para futuros estudiantes interesados en utilizar esta tecnología, se espera que este proyecto contribuya a consolidar conocimientos y abrir nuevas vías de investigación en base a sus hallazgos, beneficiando a la comunidad universitaria.

1.6 METODOLOGIA DE LA INVESTIGACION

1.6.1 Investigación Fenomenológica.

En este proyecto la investigación fenomenológica se aplicó para analizar el comportamiento real de las variables de flujo y nivel dentro del módulo, a través del monitoreo continuo se observaron y documentaron tendencias, variaciones, estabilidad del nivel del tanque y respuesta del sistema ante diferentes condiciones de operación. Este enfoque permitió comprender cómo se manifiestan las variables en un proceso físico real, más allá de la teoría, y fundamentar el diseño del controlador.

1.6.2 Investigación Hermenéutica.

La investigación hermenéutica se empleó para interpretar la información técnica disponible sobre IIoT, protocolos industriales (Modbus), controladores PLC, sensores y métodos de control, para esto se revisaron artículos, tesis, manuales técnicos y documentación oficial de los equipos utilizados, la interpretación de estas fuentes permitió seleccionar la tecnología apropiada, justificar su uso dentro del módulo y estructurar el proceso de comunicación entre PLC, HMI y la nube.

1.6.3 Investigación Práctico-Proyectual.

Esta investigación se aplicó en este proyecto como un proceso continuo de diseño, construcción del sistema de monitoreo y control de flujo y nivel, este método permitió abordar el problema desde una perspectiva operativa, comenzando por la identificación de la necesidad de contar con un módulo basado en tecnología IIoT y software libre, a partir de ello se desarrollaron las etapas de diseño estructural, selección de componentes, modelado 3D, diagramación eléctrica e hidráulica, así como la programación del Opta y la configuración del HMI, posteriormente se procedió al ensamblaje físico del módulo y a la integración de los sensores, actuadores y elementos de comunicación, finalmente se ejecutaron pruebas funcionales para evaluar el desempeño del controlador PID y del controlador ON/OFF, realizando los ajustes necesarios hasta asegurar un funcionamiento estable, este enfoque permitió mejorar progresivamente el sistema mediante ciclos de verificación y corrección, garantizando que el módulo cumpliera con los objetivos planteados.

Además, la Figura 2 ilustra la metodología propuesta de manera visual, en esta representación se destacan las fases esenciales del proyecto desde la investigación inicial hasta la implementación práctica, la representación gráfica ofrece una visión general de la estructura y ejecución de la metodología lo que facilita una comprensión más clara de la metodología empleada.

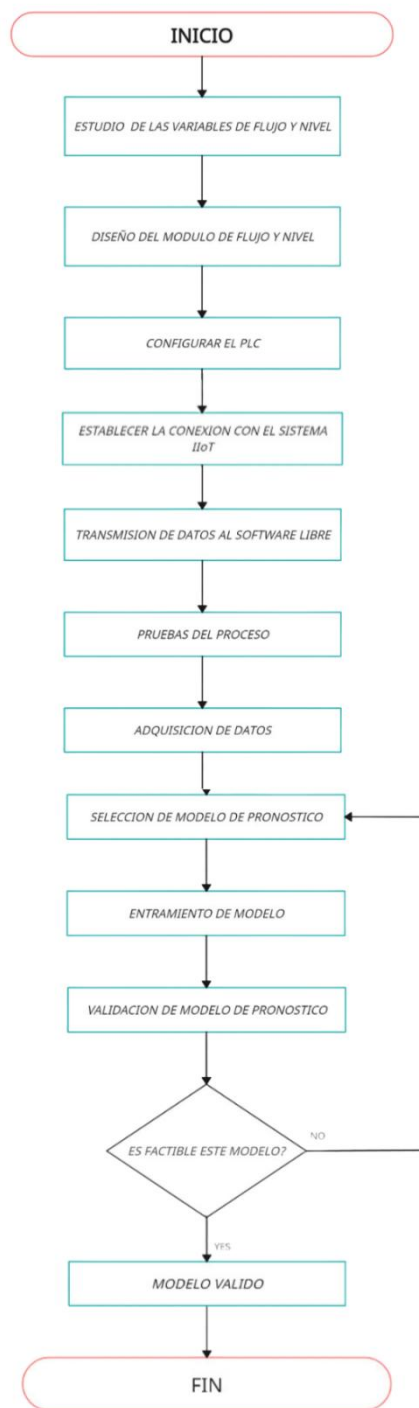


Figura 2: Diagrama de flujo metodológico

Fuente: T. Chasi, 2025

El diagrama metodológico presentado describe un sistema de monitoreo y control para un proceso de flujo y nivel, el sistema se compone de tres módulos principales: flujo y nivel, software libre y la utilización de tecnología IIoT, se capturan los datos que son procesados por el sistema IIoT luego los datos se transmiten al software libre donde se prueba y valida un modelo de pronóstico, si el modelo es válido se utiliza para controlar el sistema de relleno finalmente, se realizan pruebas para garantizar el correcto funcionamiento del sistema.

CAPÍTULO II

2. PROPUESTA

2.1 MARCO CONTEXTUAL

El proyecto se basa en desarrollar un sistema que permita supervisar y controlar el proceso de flujo y nivel mediante herramientas basadas en tecnología IIoT y software libre, la iniciativa busca integrar dispositivos de medición y control con una infraestructura digital que facilite la recopilación de datos y la interacción remota con las variables del sistema,

El proyecto se desarrolla en la Universidad Estatal Península de Santa Elena (UPSE), ubicada en La Libertad, Santa Elena, Ecuador. Esta región es conocida por su actividad industrial y educativa lo cual proporciona un entorno propicio para la implementación y prueba de tecnologías innovadoras, la infraestructura de la UPSE, junto con la colaboración de profesores y estudiantes de la Facultad de Sistemas y Telecomunicaciones ofrece los recursos necesarios para llevar a cabo este proyecto.

Sin embargo, el proyecto enfrenta varias limitaciones que pueden influir en su desarrollo y resultados dado que es una tecnología nueva, hay falta de información y se deben realizar varias pruebas de funcionamiento para garantizar su efectividad y adaptabilidad en diferentes entornos industriales, la correcta configuración y programación de sistemas IIoT y software libre requieren conocimientos técnicos avanzados los cuales demandan de una investigación exhaustiva.

Otro factor importante es la infraestructura de red ya que la efectividad del sistema de monitoreo y control depende de una infraestructura robusta para asegurar la transmisión de datos en tiempo real, en áreas con conectividad deficiente la transmisión de datos podría verse comprometida. Finalmente, el sistema diseñado debe ser escalable para adaptarse a diferentes tamaños y complejidades de procesos industriales que puedan ser aplicados a futuro además de su mantenimiento continuo que es crucial para asegurar su operatividad, lo que puede requerir recursos humanos y técnicos.

2.2 MARCO CONCEPTUAL

2.2.1 Automatización Industrial

Se define un sistema (máquina o proceso) automatizado como aquel capaz de reaccionar de forma automática (sin la intervención del operario) ante los cambios que se producen en el mismo, realizando las acciones adecuadas para cumplir la función para la que ha sido diseñado [13]. Un ejemplo de esto son los siguientes sistemas:

Lazo abierto: sucede cuando la variable de proceso no es comparada y se genera una acción independientemente de las condiciones de la misma [14].

Lazo cerrado: existe cuando la variable de proceso es medida, comparada con el setpoint y se genera una acción tendiente a corregir cualquier desviación respecto de este [14].

En la figura 3 se muestra un sistema de lazo abierto sin retroalimentación y un sistema de lazo cerrado con retroalimentación mediante sensores.

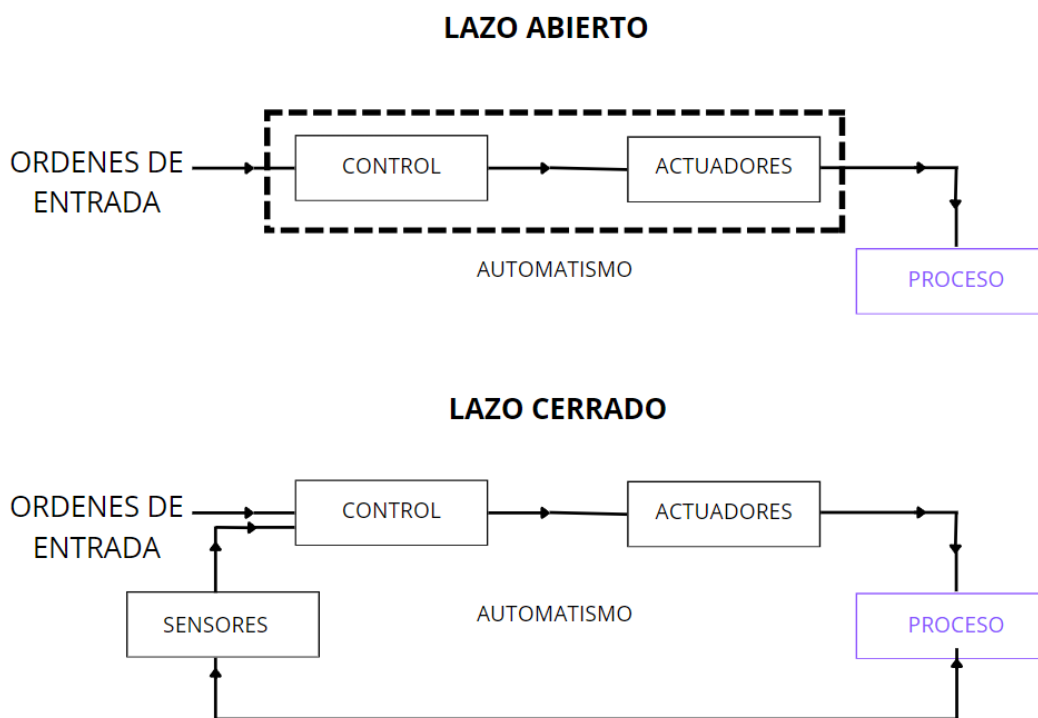


Figura 3: Diagramas de sistemas de lazo abierto y cerrado

Fuente: T. Chasi, 2025

2.2.2 Internet Industrial de las Cosas

El Internet Industrial de las Cosas (IIoT) hace posible la interconexión entre equipos, sensores y plataformas digitales dentro del entorno industrial, la capacidad de estos dispositivos para comunicarse entre sí y con sistemas externos ha impulsado nuevas formas de automatización, permitiendo que la información fluya de manera rápida y eficiente entre diferentes procesos.

La incorporación de sensores inteligentes y dispositivos distribuidos ha transformado el monitoreo industrial, ya que ahora es posible recopilar datos en tiempo real desde múltiples ubicaciones.

Software libre

Por software libre se puede entender el acceso ilimitado e irrestricto a la creación intelectual en el campo de los programas destinados a las tecnologías de la información y la comunicación (en adelante TIC), donde quiera que aquella se lleve a cabo y cualesquiera sean los propósitos para los que fue pensada [16].

PLC

Un Controlador Lógico Programable (PLC) es un dispositivo electrónico utilizado en la industria para gestionar y automatizar procesos mediante la ejecución de instrucciones en tiempo real, su función principal consiste en recibir señales de sensores, procesar la información a través de una lógica programada y enviar órdenes a actuadores, permitiendo controlar secuencias de operación de manera precisa y continua.

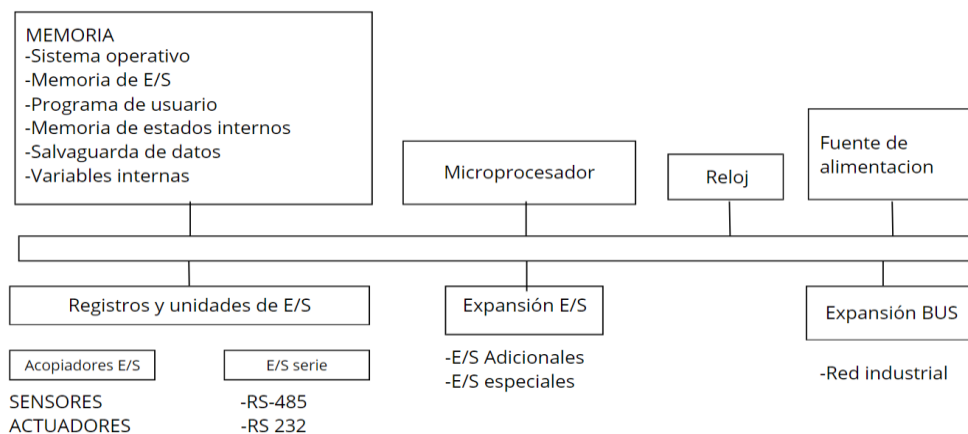


Figura 4: Diagrama describiendo la estructura interna de un PLC [18].

2.2.3 Tipos de PLC: Se clasifican en diferentes tipos dependiendo de su capacidad, tamaño y uso específico, estas características dependen mucho de los modelos y la capacidad de almacenamiento y expansión disponibles de cada autómata.

Micro PLC: Los Micro PLC son los que tienen de 15 a 128 puntos de E/S. Se utilizan más comúnmente en aplicaciones de control de baja complejidad, entre ellas algunos juegos mecánicos y dispositivos de pequeño formato.

Además, debido a que cada atracción solo puede agregar un par de dispositivos aquí y allá por motivos de seguridad, no requerirían una gran cantidad de puntos de E/S por lo que el Micro PLC ya será suficiente.

PLC Compacto: Consta de una construcción monolítica que se refiere a un enfoque en la arquitectura de sistemas donde todos los componentes necesarios están integrados en una única unidad, este diseño es comúnmente utilizado en sistemas monoprocesadores, donde un solo procesador maneja todas las tareas del sistema además, en estos sistemas es frecuente encontrar una conexión de bus de campo que facilita la comunicación y transferencia de datos entre diferentes componentes del sistema garantizando así una operación eficiente y coordinada.

PLC Modular: La construcción modular que utiliza un plano posterior, permite una mayor flexibilidad y escalabilidad en el diseño de sistemas, esta arquitectura puede soportar tanto sistemas de uno como de multiprocesador, adaptándose a las necesidades específicas de procesamiento.

Además, es habitual que estos sistemas tengan un bus de campo y una conexión LAN, lo cual hace más fácil la comunicación y la transmisión de datos, no solo entre los módulos a nivel interno, sino también a través de redes locales.

Safety PLC: Un PLC de seguridad es un controlador que realiza funciones críticas con tolerancia a fallos y siguiendo reglamentaciones de seguridad funcional, como la IEC 61508.

Soft-PLC: Son productos de automatización basados en Windows NT o CE Uso directo de CPU o coprocesadores [19].

2.2.4 Topologías de redes, para PLC.

Topología de Bus: En esta topología, todos los dispositivos están conectados a un único cable o bus. Es simple y económica, pero si el cable principal falla, toda la red se ve afectada [20].

Topología de Estrella: En la topología de estrella, todos los dispositivos están conectados a un dispositivo central, como un switch o hub. Esta configuración facilita la gestión y la solución de problemas, pero la falla del dispositivo central puede interrumpir toda la red [21].

Topología de Anillo: En la topología de anillo, cada dispositivo está conectado a otros dos dispositivos, formando un anillo. Los datos viajan en una dirección y pasan por cada dispositivo hasta llegar al destino. La falla de un solo dispositivo puede afectar la comunicación [22].

Topología de Malla: En la topología de malla, cada dispositivo tiene conexiones directas a otros dispositivos. Es altamente confiable debido a la redundancia de caminos, pero es costosa y compleja de implementar [23].

Topología Híbrida: La topología híbrida fusiona dos o más topologías fundamentales, sacando partido de sus beneficios y minimizando sus inconvenientes, puede ser difícil de diseñar y mantener, pero es escalable y flexible

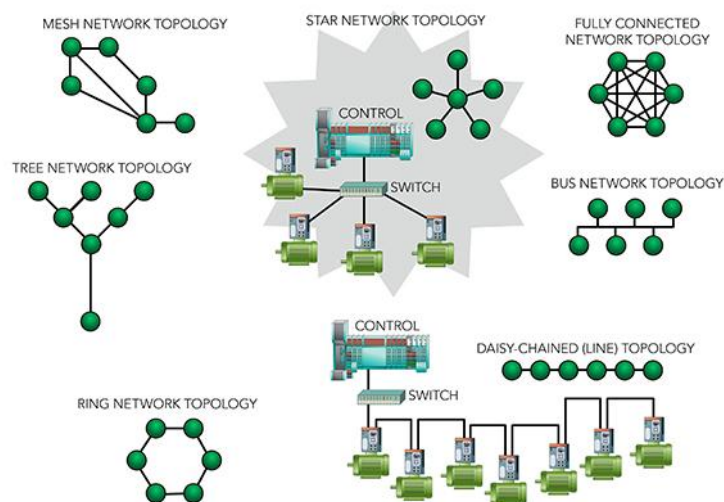


Figura 5: Topologías de redes, para PLC [25].

2.2.5 Arquitectura de redes inalámbricas con PLC a un servidor.

Controladores Lógicos Programables: Los PLC son dispositivos industriales robustos que controlan procesos y máquinas en aplicaciones industriales. Los PLCs en una red inalámbrica pueden comunicarse con sensores, actuadores y otros dispositivos para recopilar y procesar datos [26].

Módulos de Comunicación Inalámbrica: Los módulos de comunicación inalámbrica, como Wifi, Zigbee, y Bluetooth, permiten la transmisión de datos entre los PLCs y otros dispositivos en la red. Estos módulos pueden estar integrados en los PLCs o conectarse externamente [27].

Gateways: Las gateways actúan como intermediarios entre los PLCs y el servidor central. Recopilan datos de los PLCs y los transmiten al servidor, utilizando protocolos como MQTT o HTTP. Las gateways también pueden realizar un procesamiento preliminar de los datos para optimizar la transmisión [28].

Servidor Central: El servidor central recibe, almacena y procesa los datos enviados por las gateways. Este servidor puede estar ubicado en una instalación local (on-premise) o en la nube. El servidor ejecuta aplicaciones que analizan los datos y proporcionan información útil para la toma de decisiones [29].

Software de Supervisión y Control: El programa de control y supervisión, como SCADA (Supervisory Control and Data Acquisition), posibilita que los operadores supervisen y administren el sistema en tiempo real. el programa se comunica con el servidor para transmitir y recibir órdenes a los PLCs

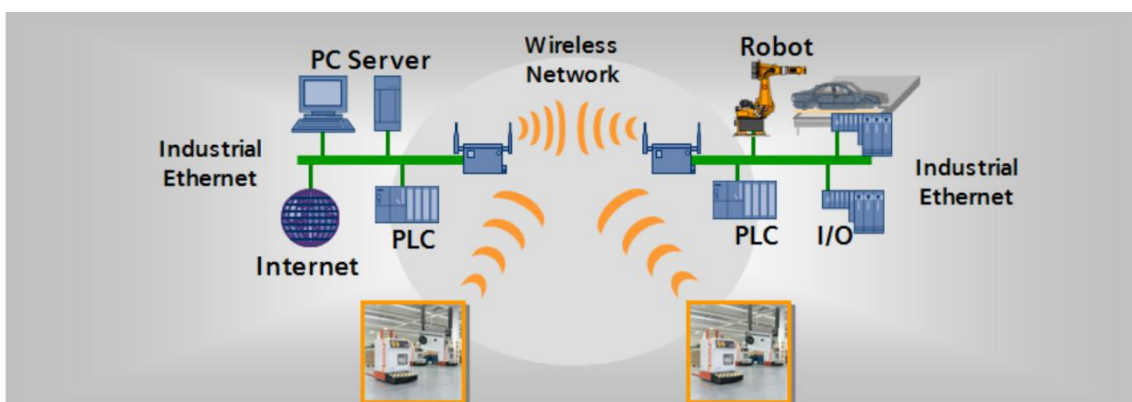


Figura 6: Arquitectura de redes inalámbricas [31]

2.2.6 Interfaz Humano máquina

Una interfaz humano-máquina es una herramienta que permite al operador visualizar información del proceso y comunicarse con los equipos de control. Su función principal es presentar datos relevantes del sistema y facilitar la supervisión y el manejo de variables industriales desde una pantalla o panel interactivo.



Figura 7: *Interfaz humano -máquina* [33].

2.2.7 Principales Sistemas de Recolección de Datos en IoT:

Sensores Inteligentes: Los sensores inteligentes son dispositivos que recopilan datos del entorno y los envían a un sistema central para su análisis. Estos sensores pueden medir una variedad de parámetros como temperatura, humedad, presión, luz, movimiento, etc. [34].

Redes de Sensores Inalámbricos (WSN): Las WSN consisten en múltiples sensores distribuidos que se comunican entre sí de forma inalámbrica para recopilar y transmitir datos a un sistema central. Estas redes son muy utilizadas en aplicaciones de IoT debido a su flexibilidad y escalabilidad [35].

Dispositivos de Pasarela (Gateways): Las pasarelas actúan como intermediarios entre los sensores y la nube o el sistema central. Recopilan datos de múltiples sensores y los envían a servidores para su procesamiento. También pueden realizar procesamiento de datos preliminar para reducir la carga de datos enviados a la nube [36].

Cloud Computing: La computación en la nube permite almacenar y procesar grandes volúmenes de datos recopilados por dispositivos IoT. Ofrece escalabilidad, flexibilidad y capacidad de análisis avanzado mediante técnicas de Big Data y Machine Learning [37].

Edge Computing: En el edge computing el procesamiento de datos se realiza cerca de la fuente de datos, es decir, en los propios dispositivos o en servidores locales. Esto reduce la latencia y el ancho de banda necesario para enviar datos a la nube [38].

Blockchain: Blockchain se utiliza para garantizar la seguridad y la integridad de los datos en IoT. Permite la creación de registros inmutables y seguros que pueden ser verificados por múltiples partes [39].

2.2.8 Tipos de control para Nivel y flujo los más utilizados:

Control On/Off: Es uno de los métodos más simples de control, utilizado principalmente en aplicaciones donde la precisión no es crucial. Funciona encendiendo o apagando un actuador (como una bomba o válvula) basado en si el nivel o flujo está por encima o por debajo de un valor de referencia [40].

Control Proporcional (P): En este tipo de control, la salida del controlador es proporcional al error de medición. Es decir, si la diferencia entre el valor deseado y el valor medido aumenta, la acción de control también aumenta [41].

Control Proporcional-Integral (PI): Este tipo de control combina la acción proporcional con una integral que se acumula con el tiempo, eliminando el error de estado estacionario que puede estar presente en un controlador proporcional puro [42].

Control Proporcional-Integral-Derivativo (PID): La razón por la que el control PID es uno de los más empleados en la industria es por su habilidad para ofrecer una respuesta rápida y estable, fusiona las acciones integral, derivativa y proporcional para optimizar la estabilidad y precisión del sistema. [43]

Control adaptativo: Para adecuarse a los cambios en las propiedades del proceso, este método modifica sus parámetros en tiempo real. En sistemas en los que las condiciones de operación varían a menudo, resulta útil [44].

Control Predictivo Basado en Modelo (MPC): Utiliza un modelo del proceso para predecir el comportamiento futuro y optimizar la acción de control en consecuencia. Es especialmente útil en sistemas multivariables y con restricciones [45].

2.2.9 Tipos de plantas industriales que existe para la parte educativa.

Lab-Volt: Sistema mecatrónica avanzado (AMS)

El sistema mecatrónica avanzado (AMS) es una plataforma educativa creada para formar y enseñar en el campo de la mecatrónica, que fusiona la electrónica, la mecánica, el control y la informática, el sistema AMS de Lab-Volt brinda un entorno modular y práctico donde los alumnos tienen la oportunidad de incorporar y utilizar principios ingenieriles en un marco realista. Comprende un conjunto de estaciones de trabajo y elementos, entre los cuales se encuentran robots, transportadores, sensores y sistemas de control que se pueden programar y configurar para llevar a cabo una serie de funciones industriales, el sistema es perfecto para adquirir capacidades en programación de PLCs, mantenimiento de aparatos mecatrónicos, diseño de sistemas integrados y automatización.



Figura 8: *Modulo Labvolt* [46].

Demostrador de sistema de control distribuido PlantPax™

El demostrador de sistema de control distribuido PlantPax™ de la Serie 3531-V de LabVolt es una herramienta educativa avanzada diseñada para la formación en sistemas de control industrial, este sistema permite a los estudiantes aprender sobre el control de procesos a través de la manipulación y monitoreo de variables como presión, caudal, nivel y temperatura.

Está equipado de software de control, sensores y actuadores que simulan un ambiente industrial real, ofreciendo una experiencia completa de aprendizaje en el área de la automatización y la regulación descentralizada.

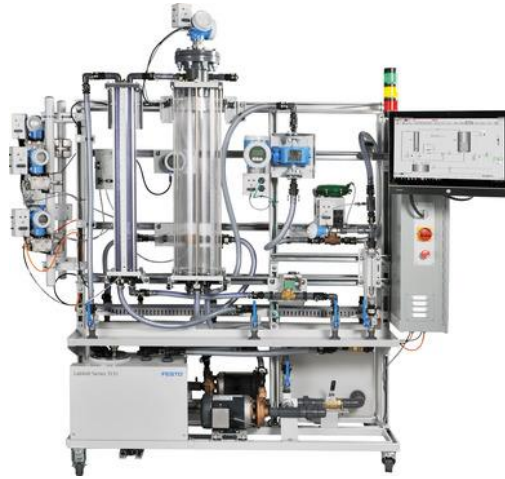


Figura 9: *Modulo PlantPAx™* [47].

Sistema didáctico de control de procesos de presión, caudal y nivel, con banco LabVolt Series 6090-A

El sistema didáctico de control de procesos de presión, caudal y nivel con banco de la Serie 6090-A, creado por LabVolt, es una herramienta pedagógica avanzada, desarrollada para impartir las bases esenciales del control de procesos, el sistema posibilita que los alumnos interactúen con un ambiente de laboratorio que reproduce procesos industriales auténticos, este equipo está dotado con sensores y actuadores que posibilitan supervisar y controlar variables críticas como la presión, el caudal y el nivel, es perfecto para adquirir habilidades prácticas en la optimización, operación y configuración de sistemas de control de procesos.



Figura 10: *LabVolt Series 6090-A* [48].

2.2.10 Normativas aplicadas en la elaboración del módulo industrial para equipos o dispositivos.

Para garantizar que el módulo funcione con seguridad, confiabilidad y buen desempeño dentro de un entorno industrial, los equipos involucrados deben diseñarse y aplicarse siguiendo normas y lineamientos técnicos reconocidos. A continuación, se incluyen algunas regulaciones y consideraciones.

IEC 60204-1: Seguridad de máquinas - Parte 1: Principios generales para el diseño de seguridad de máquinas

EN 61496-1: Equipos eléctricos de baja tensión - Protección contra el contacto eléctrico - Parte 1: Principios generales, métodos de ensayo y requisitos

IEC 61000-6-3: Seguridad eléctrica de los equipos de baja tensión - Parte 6-3: Límites para la emisión de campos electromagnéticos.

INEN 1 606:2006: Seguridad en las maquinarias, principios generales del diseño, evaluación del riesgo y disminución de este.

INEN 1 976:2008: Seguridad en sistemas electrónicos y eléctricos que se pueden programar condiciones y precauciones de seguridad funcional.

RTE INEN 005: Normas de seguridad para instalaciones eléctricas: Garantiza que las instalaciones eléctricas respeten las normas de seguridad.

2.3 MARCO TEORICO

En esta sección, se presentan los artículos y tesis relevantes que contribuyeron al desarrollo del presente trabajo de titulación.

En el año 2022, en Santiago de Chile, se realizó el trabajo “Diseño de interfaces modulares para facilitar proyectos de IoT”, que pretende crear un equipo electrónico buscando acercar a nuevos usuarios al ámbito de la electrónica y las tecnologías IoT mediante el uso de módulos que permitan conectar sensores de forma sencilla a una unidad principal, las mediciones obtenidas se envían automáticamente a una plataforma digital donde pueden visualizarse sin configuraciones complejas

En el año 2022, en Guayaquil - Ecuador, se realizó el trabajo “Automatización de un Sistema de Esclusas Empleando Tecnología del Internet de las Cosas Industrial (IIoT)”,

que pretende desarrollar un prototipo enfocado en la automatización de un sistema de esclusas de 4 compuertas. Este proyecto tiene como objetivo contener el agua entre secciones a través del uso de frenado dinámico y controladores lógicos programables. Las mediciones y el control del sistema se realizan mediante tecnología de la industria 4.0 y se visualizan a través de interfaces humano-máquina y bases de datos, asegurando una implementación efectiva de IoT en proyectos industriales [50].

En el año 2022, en la Universidad Politécnica Salesiana, se realizó el trabajo “Desarrollo de un sistema SCADA 4.0 utilizando herramientas IIOT para una planta de entrenamiento de procesos continuos controlada remotamente”, que pretende crear un sistema de control y monitoreo remoto empleando tecnologías de Internet de las Cosas (IIOT); a través de la implementación de una arquitectura cliente-servidor y el uso de una plataforma de comunicación en la nube, para que los datos de los procesos puedan ser gestionados y visualizados de manera eficiente y en tiempo real .

CAPÍTULO III

3. DESARROLLO DE LA PROPUESTA

Componentes de la propuesta

Esta sección describirá detalladamente los componentes tanto físicos como lógicos necesarios para la implementación del sistema de monitoreo y control de flujo y nivel basado en tecnología IIoT en este se emplean varios elementos eléctricos, controladores lógicos programables, sensores, estos se han clasificado para facilitar su comprensión y aplicación desde las primeras etapas hasta las fases finales del proyecto para cada componente físico, se incluirán las características o especificaciones técnicas más importantes, proporcionando así una visión más completa y detallada de cada uno de ellos.

3.1. COMPONENTES FÍSICOS

En este apartado se recopilan de manera generalizada todos los elementos físicos más relevantes que han sido adquiridos con anticipación para ser implementados en el desarrollo de la propuesta con el objetivo de contribuir a darle forma y complementar su ejecución de manera óptima.

3.1.1 Opta Arduino Wifi.

El Arduino Opta es un micro PLC diseñado para aplicaciones industriales con capacidad IoT y una estructura orientada a facilitar la integración con diferentes sistemas, su presentación incorpora puertos comunes como USB, LAN y terminales de entrada y salida que permiten conectar sensores, actuadores y dispositivos de comunicación sin complicaciones como se muestran en la Figura 11 fue diseñado en colaboración con Finder, el fabricante líder de dispositivos de automatización industrial y de edificios, Opta permite a los profesionales ampliar los proyectos de automatización aprovechando al mismo tiempo el ecosistema Arduino.

El microPLC Opta es fundamental en este proceso porque permite que el sistema IIoT y los dispositivos de medición compartan e integren datos, lo cual posibilita la captura y transmisión de información en tiempo real.

La Tabla 1 detalla las características técnicas relevantes del Opta Arduino Wifi.



Figura 11: *Arduino Opta WiFi* [51].

Tabla 1 . Arduino Opta Wifi

Características	Detalle
Modelo	AFX00002
RS-485	Half-duplex
Wi-Fi®	802.11 b/g/n
Voltaje de alimentación	12-24 VDC
Entradas	8 entradas analógicas/digitales
Salidas	4 relés (normalmente abiertos - NO)
Corriente máxima por relé	10A
Voltaje máximo del relé	400 VAC

Fuente: [52]

3.1.2 Sensor de flujo de tipo turbina

El sensor de flujo de pasillos de acero inoxidable de 1 pulgada que se muestra en la Figura 12 se compone principalmente de un cuerpo de válvula de latón, un conjunto de rotor de flujo de agua y un sensor Hall este emite la señal correspondiente y la devuelve al controlador, el dispositivo juzga el tamaño del flujo de agua y lo regula este sensor genera pulsos por litro de agua (595 pulsos/litro para la rosca de 1"), estos pulsos son fáciles de

procesar electrónicamente lo que permite integrarlo fácilmente con sistemas de control y monitoreo como microcontroladores o sistemas embebidos para registrar el flujo.

La Tabla 2 detalla las características técnicas relevantes del sensor de flujo de pasillos de acero inoxidable



Figura 12: sensor de flujo de pasillos de acero inoxidable [53].

Tabla 2 Características del sensor de flujo de pasillos de acero inoxidable de 1 pulgada

Características	Detalle
Roscas externas	1"
Temperatura de funcionamiento	-25 ~ +80°C
Presión permitida	1.75 Mpa
Rango de humedad	35% ~ 90% RH (sin escarcha)
Capacidad de carga	10 mA (DC 5V)
Rango de voltaje de trabajo	DC 5 ~ 18V
Corriente máxima de funcionamiento	15 mA (DC 5V)
Voltaje de trabajo nominal más bajo	DC 4.5 5V-24V

Fuente: [53]

3.1.3 Transmisor de nivel KX-132V

El transmisor de nivel de líquido que se muestra en la figura 13 se basa en el principio de que la presión hidrostática medida del nivel de líquido es proporcional a la altura del líquido este utiliza el efecto piezorresistivo de silicio difusivo para convertir la presión en una señal eléctrica, después de la compensación de temperatura y la calibración lineal, se convierte en un estándar. Las salidas de corriente, voltaje, RS485 y otras señales se pueden conectar directamente a PLC lo que permite integrarlo fácilmente con sistemas de control y monitoreo.

La Tabla 3 detalla las características técnicas relevantes del sensor de flujo de pasillos de acero inoxidable



Figura 13: Transmisor de sensor de nivel de líquido KX-132V [54].

Tabla 3 Transmisor de sensor de nivel de líquido

Características	Detalle
Rango de medición	0 ~ 0.5 ~ 1,640.4 ft H2O
Fuente de alimentación	24VDC
Señal de salida	4 ~ 20 mA

Fuente: [54]

3.1.4 HMI Weintek cMT3072X

La pantalla táctil industrial Weintek CMT3072XH mostrada en la figura 14 es una interfaz hombre-máquina (HMI) de 7 pulgadas diseñada para aplicaciones industriales nos ofrece una resolución de 1024 x 600 píxeles con tecnología IPS, asegurando una representación precisa de colores y un amplio ángulo de visión, está equipada con un procesador de cuatro núcleos, 4 GB de memoria flash y 1 GB de RAM, lo que permite un rendimiento eficiente en la visualización y el control de sistemas.

La Tabla 4 detalla las características técnicas relevantes del HMI Weintek CMT3072XH



Figura 14: *Weintek CMT3072XH.* [55].

Tabla 4 Weintek CMT3072XH.

Características	Detalle
Pantalla	7" IPS, resolución 1024 x 600 píxeles
Procesador	Quad-core ARM
Conectividad	RS-232, RS-485, CAN Bus
Certificaciones	CE, FCC, RoHS
Fuente de alimentación (DC)	24V

Fuente: [55]

3.1.5 Convertidor de frecuencia Invt Goodrive10 Series.

Este dispositivo que se muestra en la figura 15 permite ajustar la velocidad del motor en función de las demandas del sistema para poder realizar el control de este, también el

Goodrive10 Series utiliza el protocolo Modbus RTU que es compatible con el resto de los dispositivos facilitando la interacción de todos en el sistema.

La Tabla 5 describe de manera detallada los rasgos técnicos del convertidor de frecuencia Invt. Serie Goodrive10.



Figura 15: Convertidor de frecuencia Invt Goodrive10 Series [56]

Tabla 5. Invt Goodrive10 Series

Características	Detalle
Entrada	220-240 V monofásico/trifásico o 380-440 V trifásico
Salida	Frecuencia ajustable de 0 a 400 Hz
Capacidad de sobrecarga	150% por 1 minuto, 200% por 1 segundo
Entradas/salidas analógicas	1 entrada (0-10V/0-20mA), 1 salida configurable
Comunicación	Protocolo Modbus RTU
Control	Modos V/F y SVPWM

Fuente: [56]

3.1.6 Bomba de agua de 1 HP

La bomba que se muestra en la figura 16 suministra la presión adecuada para mover el flujo a través del proceso siendo clave para mantener las condiciones operativas óptimas. Su potencia es apropiada para aplicaciones de pequeña escala permitiendo un control eficiente del nivel de agua en tanques o sistemas cerrados.

La Tabla 6 detalla las características técnicas relevantes del Bomba de agua de 1HP.



Figura 16: Bomba de agua de 1 HP [57].

Tabla 6. Bomba de agua de 1 HP

Características	Detalle
potencia del motor en HP	1HP
Voltaje	Trifásica 380-440 V
PSI	30-50

Fuente: [57]

3.1.7 Interruptor Automático Schneider Electric Easy9 C25.

Este interruptor automático mostrado en la figura 17 es utilizado para proteger el circuito de alimentación del sistema de monitoreo y control, incluyendo componentes como el variador de frecuencia y la bomba trifásica, su función es garantizar la seguridad del sistema previniendo daños causados por sobrecargas eléctricas o cortocircuitos lo que resulta esencial para mantener la continuidad y la integridad operativa del proyecto.

La Tabla 7 detalla las características técnicas relevantes del Breaker Schneider.



Figura 17: Breaker Schneider [58].

Tabla 7 Schneider Electric Easy9 C25

Características	Detalle
Polos	2
Amperaje	25A
Corriente cortocircuito	20
Voltaje de entrada	220-440v

Fuente: [58]

3.1.8 Válvula de Bola de Latón de Paso Total para Agua

La función de esta válvula mostrada e en la figura 18, es regular el paso del agua por las tuberías, su diseño de paso total permite que el fluido pase sin restricciones cuando la válvula está completamente abierta, además es resistente a la corrosión, ya que está compuesta de latón.



Figura 18: Válvula [59].

3.8.9 Switch de red TP-Link de 8 puertos

El switch de red Ethernet de 8 puertos es un dispositivo activo clave en la capa de enlace de datos de la red local, su función esencial es gestionar el tráfico de información mediante el direccionamiento físico (direcciones MAC) para garantizar que los datos fluyan correctamente entre el microPLC, los sensores y la plataforma de monitoreo, por ende este componente cuya presentación física se muestra en la Figura 19 opera como un punto central de interconexión, facilitando la comunicación simultánea y minimizando las colisiones de paquetes, su uso es necesario en el sistema IIoT para asegurar la transferencia de datos en tiempo real con alta confiabilidad y baja latencia.

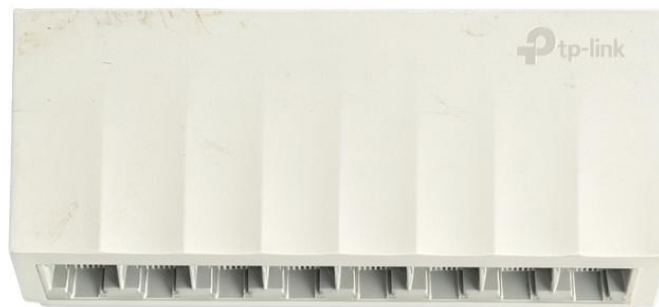


Figura 19: *Switch de red TP-Link* [60]

3.8.10 Elementos de control y señalización.

Estos componentes que se muestran en la figura 20 desempeñan un papel crucial dentro del sistema ya que proporcionan la señalización del estado operativo y mediante los cuales se inicia o finaliza el proceso de control y monitoreo de nuestra planta



Figura 20: *Elementos de control y señalización* [61]

3.2 COMPONENTES LÓGICOS

3.2.1 SolidWorks

Es un programa de software CAD, que significa diseño asistido por computadora, que se usa principalmente para modelar en tres dimensiones, este software es ampliamente utilizado en ingeniería, diseño industrial y manufactura para crear modelos precisos de piezas, ensamblajes y dibujos técnicos, se empleará para realizar el diseño de la estructura del módulo, el cual contendrá las cotas necesarias, además del tanque de agua que debe incluirse en el módulo.



Figura 21: *SolidWorks* [62].

3.2.2 IDE de PLC de Arduino

El Arduino IDE mostrado en la figura 22 es una plataforma de desarrollo que permite programar controladores como el arduino Opta de manera sencilla y accesible diseñado con un enfoque en la flexibilidad, el IDE es de código abierto y soporta diversos lenguajes de programación como ladder lo que facilita la creación de soluciones de automatización industrial y proyectos IIoT utilizando el arduino IDE se puede configurar y controlar el arduino Opta a través de una interfaz amigable simplificando la programación de funciones avanzadas como el monitoreo y control en tiempo real de procesos industriales.



Figura 22: *IDE de Plc* [63].

3.2.3 IoT CLOUD

Arduino IoT Cloud es una plataforma en la nube que facilita el desarrollo de soluciones IoT permitiendo la conexión y control de dispositivos desde cualquier lugar con acceso a Internet con su interfaz amigable se pueden visualizar y gestionar datos en tiempo real mediante dashboards personalizables, así como configurar reglas automáticas para activar acciones específicas en función de los valores recibidos de los sensores la Figura 23 ilustra tanto la identidad visual de la plataforma como su interfaz de monitoreo móvil, la plataforma utiliza protocolos como MQTT para una comunicación eficiente y segura entre los dispositivos y la nube.

IoT Cloud permite una integración perfecta con el microPLC Arduino Opta que ya forma parte del sistema la plataforma no solo facilita el monitoreo en tiempo real de las variables clave del proceso, como el flujo y el nivel sino que también ofrece la posibilidad de supervisar el sistema desde cualquier lugar, a través de una conexión a internet, esto resulta particularmente útil para garantizar una vigilancia continua sin la necesidad de estar presente físicamente en la planta, adicional al poder configurar alertas y acciones automáticas en la nube se puede reaccionar de inmediato ante variaciones críticas en las mediciones, optimizando la eficiencia del proceso y evitando fallos inesperados, la posibilidad de acceder a datos históricos y analizarlos también permite ajustar el sistema de manera más precisa, mejorando su rendimiento general y facilitando la toma de decisiones basadas en datos actualizados.



Figura 23: IoT CLOUD [64].

3.2.4 EasyBuilder Pro

El EasyBuilder Pro mostrado en la figura 24 es un software de diseño y programación de interfaces gráficas utilizado para configurar HMIs en proyectos de automatización industrial, su funcionalidad permite representar y controlar procesos en tiempo real a través de visualizaciones intuitivas y alarmas.

Permite diseñar una interfaz gráfica que visualiza variables críticas del proceso como flujo y nivel adicional facilita el control manual y automático de los elementos del sistema asegurando una supervisión eficiente, además su compatibilidad con protocolos como Modbus RTU y TCP asegura la integración con el resto de los dispositivos del sistema.



Figura 24: *EasyBuilder Pro* [65].

3.3 DISEÑO DE LA PROPUESTA

En esta sección se detalla el proceso de diseño y construcción del módulo de flujo y nivel siendo un componente central para la evaluación y el monitoreo de las variables industriales seleccionadas

3.3.1 Diseño estructural y del tanque (Modelado 3D en Solidwork)

En la figura 25 se muestra el diseño del módulo estructural, creado en SolidWorks, que sirve como base de referencia para la fabricación del módulo, este modelo 2D fue desarrollado con el objetivo de proporcionar una representación visual clara de la estructura, facilitando tanto la planificación y la ejecución del proceso de construcción.

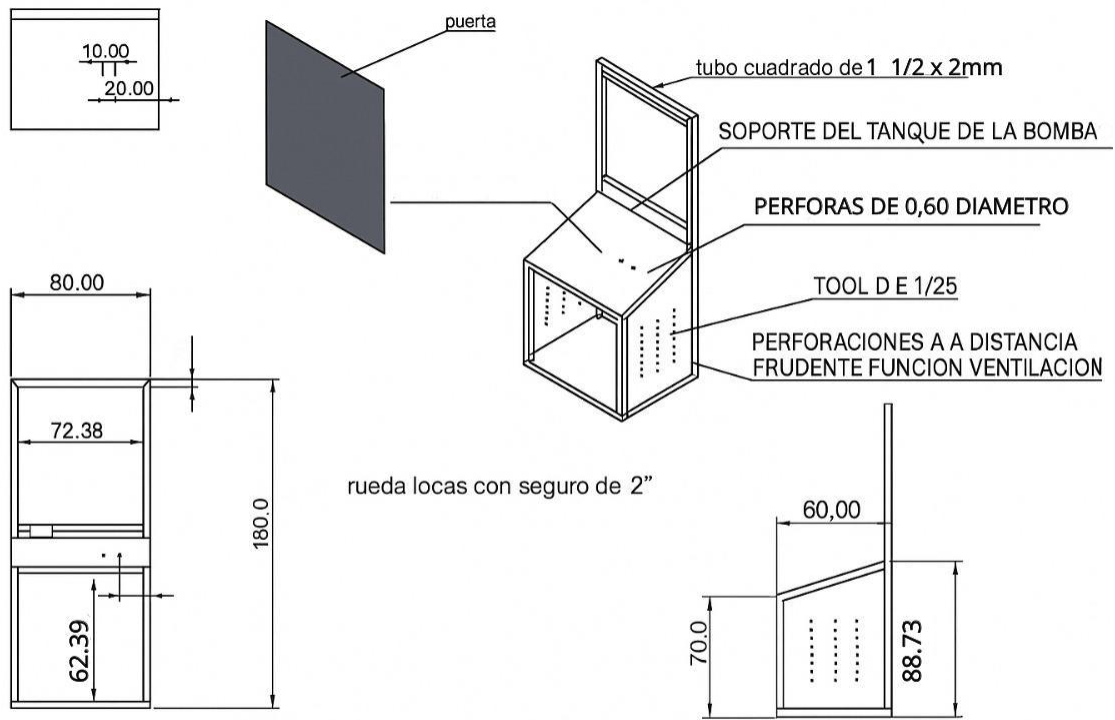


Figura 25: Medidas y diseño modulo.

Fuente: T. Chasi, 2025

La representación tridimensional (3D) de la estructura final en la Figura 26 ofrece una perspectiva visual completa y detallada de cómo se integra el módulo de flujo y nivel, este modelado nos permite validar la distribución espacial de los componentes, y como quedaría ya finalizado.

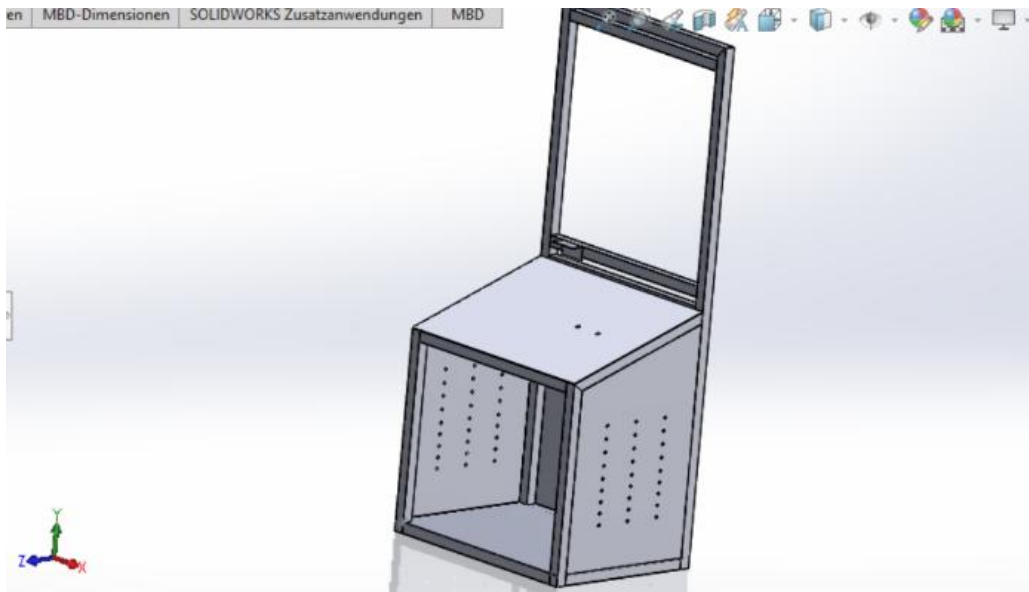


Figura 26: Modulo en 3D.

Fuente: T. Chasi, 2025

Además, en la figura 27 se presenta el diseño en SolidWorks del tanque cilíndrico que servirá como referencia para su fabricación, este modelo 3D ha sido diseñado para facilitar la construcción física del tanque que estará fabricado con material acrílico, seleccionado por sus propiedades de resistencia química, transparencia y ligereza.

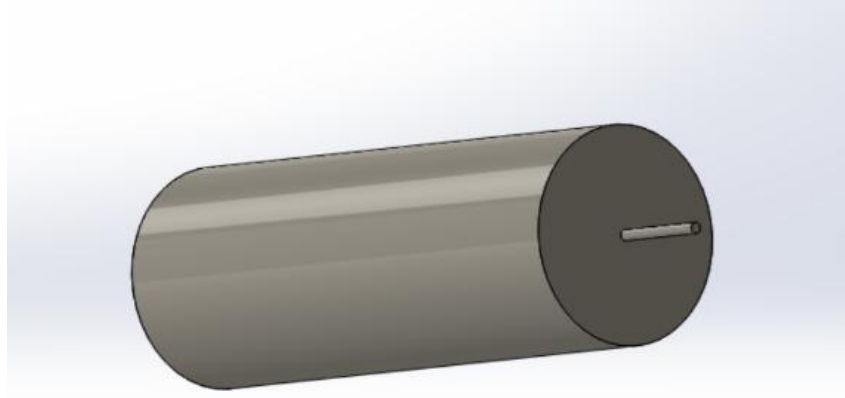


Figura 27: *Tanque en 3D*

Fuente: T. Chasi, 2025

3.3.2 Diseño Físico del Módulo

Para la realización del diseño, la estructura se construyó utilizando tubo cuadrado de hierro negro con dimensiones de 1 1/2 pulgada y 2 mm de espesor, este material fue seleccionado por su robustez puesto a que se trabaja con agua, se integraron ruedas con frenos en la base de la estructura como se aprecia en la figura 28.



Figura 28: *Esqueleto modulo*

Fuente: T. Chasi, 2025

La parte superior de la estructura incluye una base redonda diseñada específicamente para sostener el tanque cilíndrico de acrílico en esta base se incorpora un orificio central que facilita la conexión de la tubería inferior del tanque con el sistema de bombeo. Adicionalmente, se instaló una puerta frontal con un mecanismo de seguro para proporcionar un acceso controlado a la bomba de agua y conexiones eléctricas añadiéndole las ruedas locas con seguro.

Con el objetivo de garantizar que los equipos eléctricos ubicados en el interior funcionen bien, se hicieron agujeros en los costados de la estructura para fomentar la ventilación natural del sistema, la figura 29 ilustra el ensamblaje final de la estructura: las ruedas, la puerta y la base de soporte.



Figura 29: *Estructura final del módulo*

Fuente: T. Chasi, 2025

3.3.3 Construcción y ensamble del panel del control del sistema

Para esta parte del módulo se decidió usar una base de madera RH (material que resiste la humedad), se realizó primero el orificio para empotrar el HMI Weintek cMT3072X como se aprecia en la figura 30.

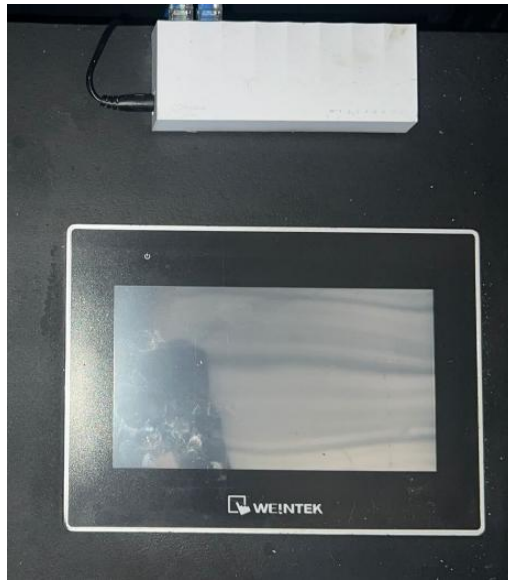


Figura 30: Instalación del HMI en el panel.

Fuente: T. Chasi, 2025

Se continuo con el convertidor de Frecuencia (VFD), el Arduino microPlc Opta, Switch de red TP-Link; y los elementos de control y señalización. Estos últimos incluyen los pulsadores de marcha (verde), paro (rojo) y emergencia (tipo hongo), junto con las luces piloto para indicar los estados del sistema además Adicionalmente, se dispuso de un espacio de trabajo al lado izquierdo junto al panel de control para facilitar el posicionamiento de una laptop durante las fases de programación, pruebas y monitoreo en tiempo real como se puede visualizar en la figura 31.



Figura 31: Panel de control finalizado.

Fuente: T. Chasi, 2025

3.3.4 Diseño diagramas de conexión

Este esquema es una representación completa de la interconexión de todos los componentes del sistema, proporcionando información detallada sobre sus partes y su interacción.

3.3.4.1 Diagrama de redes Modbus TCP I/P

El diagrama de la Figura 32 detalla la conexión de los componentes clave para la funcionalidad IIoT y de control local, centrándose en las conexiones de red y los buses de datos.

El microPLC Arduino Opta Wifi es el nodo principal de procesamiento y se ocupa de administrar las conexiones siguientes:

- Red de Ethernet (LAN): El Switch de red (SW1) asegura que el sistema esté conectado. El HMI (CMT1) y el microPLC OPTA se conectan al switch TP-Link mediante su puerto LAN para crear la red de control local (IIoT).
- Conexión a PC: El diagrama ilustra una conexión directa (ya sea por Ethernet o USB) a una Computadora (PC). Esta conexión es esencial para las tareas de programación, *debugging* y revisión del correcto funcionamiento del sistema y la configuración de la nube IIoT.
- Conexión inalámbrica: El microPLC Opta Wifi incorpora funcionalidades inalámbricas para transmitir datos a la nube (IoT Cloud), lo que posibilita el monitoreo en tiempo real y el acceso remoto.

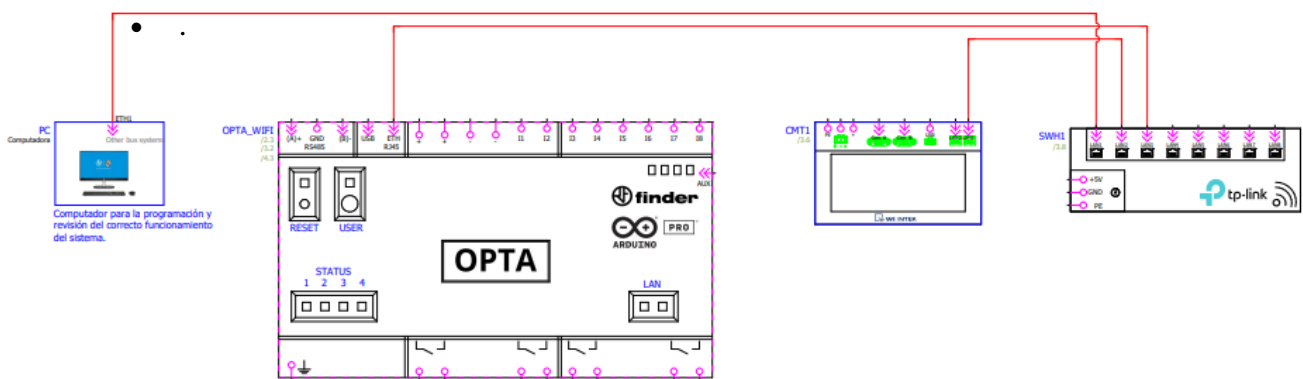


Figura 32: Diagrama de redes modbus TCP I/P.

Fuente: T. Chasi, 2025

3.3.4.2 Diagrama de redes Modbus RS485

La Figura 33 muestra el esquema detallado de cableado para la gestión de las Entradas/Salidas (I/O) y el control directo entre el convertidor de frecuencia (VDF1 - INVT Goodrive 10) y el microPLC Arduino Opta Wifi, este esquema proporciona una alternativa para las funciones críticas del VFD si no se utiliza el control exclusivamente por comunicación serial.

El esquema detalla lo siguiente:

- Entradas Digitales del VFD: Múltiples terminales de entrada digital del VFD (S1 a S5) están disponibles para ser cableados a salidas digitales del PLC o a pulsadores de control directo lo que permite configurar funciones de control básicas, como la habilitación o selección de velocidad.
- Señal de Control Analógico (Salida VFD/Entrada PLC): Se muestra la conexión de la salida de voltaje de 0-10V del VFD (terminal +10V y GND) a una de las entradas analógicas del microPLC OPTA (A/I), Aunque el control principal se realiza por Modbus, esta conexión permite al PLC monitorear la señal de referencia de frecuencia que está recibiendo el variador.
- Conexión de Control Manual: El pulsador B2 está vinculado con una entrada digital del OPTA, que suele estar destinada a comandos de alta prioridad, por ejemplo, la función de reinicio del sistema (-A1) o el paro de emergencia. Esto asegura una respuesta inmediata.
- Aislamiento y tierra: Es indispensable para la seguridad eléctrica y la integridad de las señales en el ambiente industrial que se verifique la conexión a tierra de protección (PE) en el microPLC y el VFD.

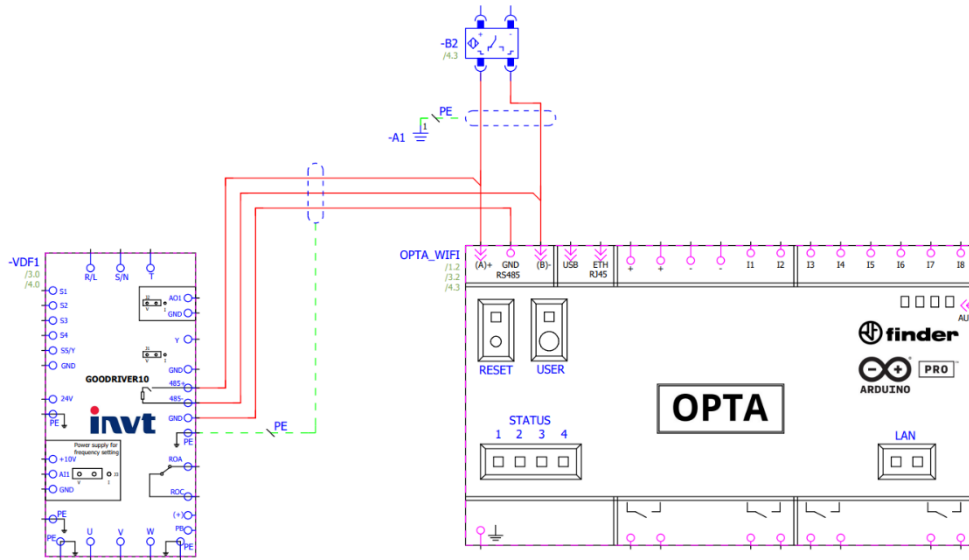


Figura 33: Diagrama de redes Modbus RS485

Fuente: T. Chasi, 2025

3.3.4.3 Diagrama de control

La Figura 34 presenta el esquema de cableado detallado de todos los elementos de entrada y salidas con el microPLC Arduino Opta Wifi, este diagrama es un punto clave ya que en este se visualiza como se reciben las señales de proceso y de operación y cómo trabajan los accionadores, la señalización cerrando el lazo de control.

El esquema se compone de:

- Sensores:
 - El Transmisor de Caudal se conecta a una entrada digital del OPTA, configurada para el conteo de pulsos.
 - El Transmisor de Nivel se conecta a una entrada analógica, recibiendo la señal de 4-20 mA.
- Accionadores:
 - Los pulsadores de Marcha (-S1) y Paro (-S2) se conectan a entradas digitales, permitiendo la marcha y paro manual del proceso.
 - El pulsador de emergencia (-B2), que tiene prioridad en el cableado, garantiza la desconexión total del sistema (-A1).

-Actuador de Potencia: Se mantiene la comunicación serial RS-485 con el Controlador Indicador de Frecuencia (VDF1) para la modulación de la velocidad de la bomba, el elemento final para el control de flujo.

- Señalización:

Las luces piloto de encendido (-P1), paro (-P2) y falla (-P3) son controladas por las salidas de relé del OPTA, estas luces se alimentan a través de la línea de tensión (L y N), lo que permite indicar el estado.

- La figura 34 muestra la topología de control en forma de estrella el microPLC OPTA se encarga de centralizar la obtención y el procesamiento lógico de los datos, así como las acciones de control. Se trata del punto donde todas las señales del proceso se encuentran.

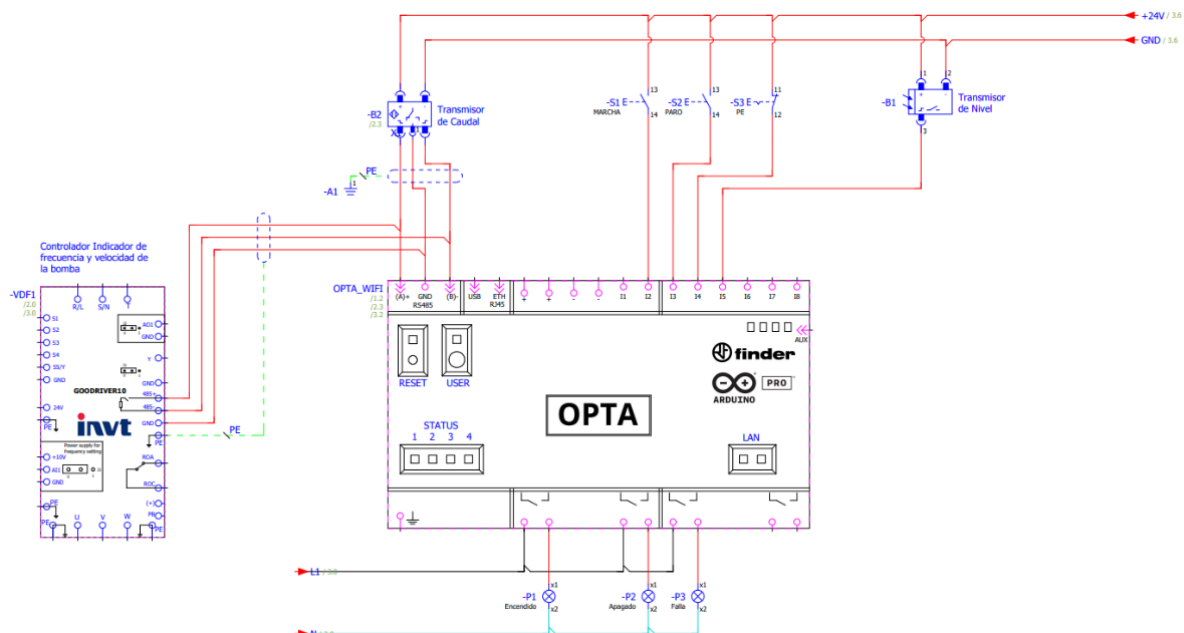


Figura 34: Diagrama de control

Fuente: T. Chasi, 2025

3.3.4.4 Diagrama de potencia

La Figura 35 representa el diagrama eléctrico general del módulo, juntando los circuitos de potencia y control en un solo esquema, este diseño valida la interconexión entre la alimentación principal, los sistemas de protección, los actuadores de potencia y los componentes de control digital.

El diagrama detalla las siguientes conexiones:

- **Circuito de Potencia:** La alimentación de 220 V (L1 y L2) ingresa al sistema protegida por interruptores automáticos (F1, F2). El Convertidor de Frecuencia (VFD1) se alimenta de esta línea y su salida (U, V, W) se conecta al Motor Trifásico (M1), permitiendo la modulación electrónica de la velocidad para el control de flujo.
- **Fuentes de Alimentación:** para conseguir las tensiones de bajo voltaje, dos transformadores reductores (T1 y T2) derivan la línea de 110 V. El microPLC OPTA y el HMI (CMT1) reciben +24 VDC del transformador T1, el transformador T2 provee +5 VDC al conmutador de red (SW1).
- **Comunicación y Control Central:** El microPLC OPTA es alimentado con +24V. La conectividad se muestra a través de LAN (para la red IIoT) y RS-485 (para el VFD).
- **Interfaces de Red:** El HMI (CMT1) y el Switch de red (SW1) están conectados a la red de control, el Switch es crucial para la comunicación Ethernet y la gestión del tráfico de datos que soporta la arquitectura IIoT.

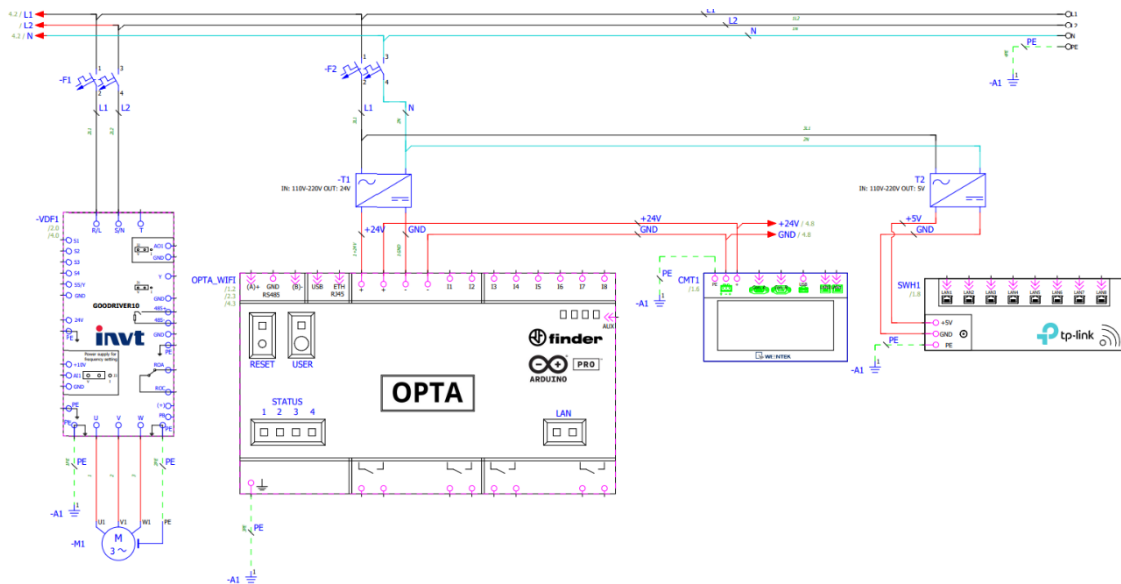


Figura 35: Diagrama de potencia/alimentación

Fuente: T. Chasi, 2025

El diseño de todos los diagramas presentados en esta sección nos ayuda a validar la arquitectura de control y la viabilidad de implementación del proyecto, el diseño cumple con las directrices de seguridad al establecer una clara diferencia entre los circuitos de potencia (220 V) y los de control (+24 VDC); el μ PLC Arduino Opta se consolida como el elemento central del sistema, gestionando el lazo de control cerrado mediante la adquisición de datos de los transmisores de flujo y nivel, el accionamiento de los pulsadores y la del actuador final (VFD) vía comunicación serial RS-485 y finalmente, la infraestructura de red visible en los diagramas garantiza la conectividad IIoT esencial para el monitoreo en la nube a través de Industrial Ethernet, concluyendo así la fase de diseño de hardware.

3.3.5 Diagrama de tuberías e instrumentación (P&ID)

El diagrama de tuberías e instrumentación mostrado en la figura 36 representa la distribución general de los componentes que conforman el sistema de monitoreo y control.

El sistema está compuesto por dos tanques: el tanque principal y el tanque de reservorio, el agua se bombea desde el tanque de reservorio hacia el tanque principal mediante una bomba centrífuga, la cual es controlada por un variador de frecuencia (VFD-01) que regula la frecuencia en función de la señal de control generada por el Opta (PLC-01).

En la línea de descarga de la bomba se incorporan válvulas de latón (V1 y V2) que permiten aislar o ajustar el flujo del proceso según las condiciones operativas además de provocar perturbaciones en el sistema. Posteriormente, el fluido atraviesa el sensor de flujo (TF-01), encargado de medir la cantidad de agua que circula por la tubería y enviar dicha información al PLC para su procesamiento.

En el tanque principal se ha instalado un transmisor de nivel (LT-01), que supervisa constantemente la altura del fluido, esta variable es crucial para controlar el proceso, pues el PLC se basa en esta lectura para regular el accionamiento de la bomba y mantener así el nivel dentro del rango definido.

El PLC (PLC-01) es el centro del sistema y se encarga de la comunicación entre el HMI (HMI-01), los sensores y el actuador (bomba), el operador tiene la posibilidad de observar en tiempo real las variables de flujo y nivel por medio del interfaz entre el ser humano y la máquina, así como también puede alterar los parámetros operativos, tales como el modo de operación o los valores de referencia.

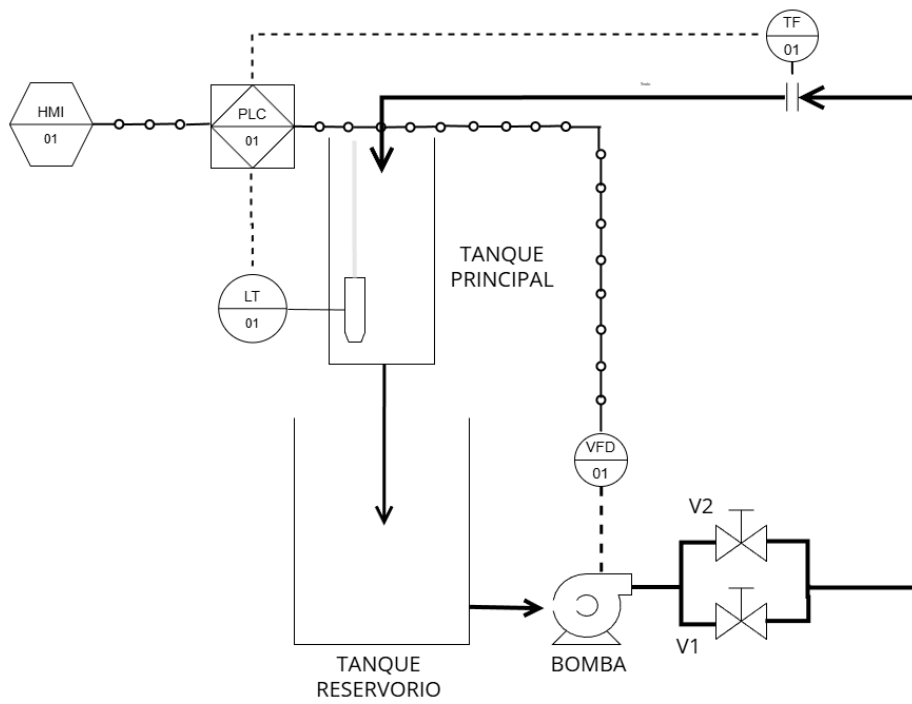


Figura 36: Diagrama P&ID

Fuente: T. Chasi, 2025

3.3.6 Cableado diagramas de potencia y control.

Primero se colocó la alimentación de 220v donde se realizó la instalación del interruptor automático para la protección y para energizar el sistema, además realizó la conexión del variador de frecuencia al motor de la bomba, tal como se muestra en la figura 37.



Figura 37: Alimentación 110-220v

Fuente: T. Chasi, 2025

Posteriormente como se muestra en la figura 38 se instaló el circuito de control derivando una línea de 110V para alimentar las fuentes que suministran +24 VDC y +5 VDC a todos los equipos electrónicos, luego se procedió al cableado final de los actuadores donde se instaló el HMI, los sensores, luces pilotos y los pulsadores de marcha, paro, emergencia.



Figura 38: *Conexión actuadores.*

Fuente: T. Chasi, 2025

3.3.7 Montaje tubería

Se realizó la instalación del circuito hidráulico utilizando tubería diseñada para la circulación del flujo, el sistema se compone; del tanque de reservorio (superior, figura 39) y el tanque principal de medición (inferior, figura 40).



Figura 39: *Tanque reservorio*

Fuente: T. Chasi, 2025



Figura 40: *Tanque principal*

Fuente: T. Chasi, 2025

La bomba se integró al sistema, conectando la línea de succión al tanque de reservorio mientras que la línea de descarga de la bomba se dirigió hacia el tanque principal de medición (superior), se instalaron válvulas de corte estratégicamente en la tubería de descarga como se muestra en la figura 40, para permitir el aislamiento de la bomba en caso de mantenimiento o falla, sin necesidad de vaciar el circuito completo.

La figura 40 muestra el detalle de la tubería de descarga superior, la cual se diseñó para que el caudal pasara directamente a través del sensor de flujo, esta ubicación en línea, justo antes de la descarga en el tanque principal, asegura que la medición de la variable de proceso sea precisa, además se observa el transmisor de nivel sumergido en el tanque principal de acrílico, la finalización de la instalación de la tubería concluye la fase de montaje de hardware, dejando el módulo completamente operativo en sus circuitos eléctrico e hidráulico.

3.3.8 Diseño interfaz HMI

El diseño de la Interfaz Humano-Máquina (HMI) es esencial porque constituye el principal punto de interacción entre el sistema de control y el operador, la interfaz fue creada con el software EasyBuilder Pro (componente lógico 3.2.3) para ofrecer una visualización exacta, clara y en tiempo real de las variables del proceso.

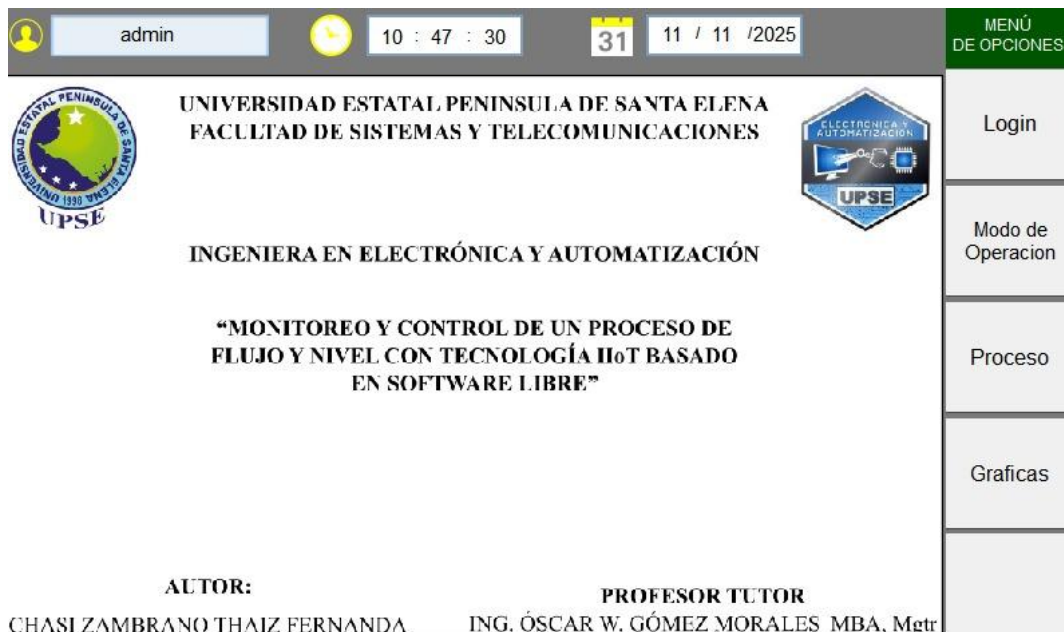


Figura 41: *Pantalla principal*

Fuente: T. Chasi, 2025

La pantalla principal mostrado en la figura 41 incluye:

- Identificación Institucional: Presentación de la Universidad Estatal Península de Santa Elena (UPSE), la Facultad y la Carrera, junto con el título del proyecto, destacando la autoría y la tutoría.
- Información de tiempo real: Indicadores de fecha y hora actualizados en la parte superior.

Además, en la primera pantalla existen botones de función, posicionados estratégicamente en el lado derecho de la pantalla, la misma que permite al operador interactuar con el sistema, mediante estos botones se asegura, un acceso rápido a las funciones, el primer botón de función login o autenticación que se superpone en la primera pantalla se implementó como un mecanismo de gestión de niveles de acceso dentro del HMI como se muestra en la figura 42, su propósito es esencialmente identificar al usuario que interactúa con el sistema y asignar los privilegios correspondientes para la edición de parámetros, esta gestión se establece mediante la definición de roles; al nivel superior se le otorga acceso total para modificar variables críticas del sistema, mientras que al operador se le limita el acceso únicamente a la vigilancia y el monitoreo de las variables y alarmas, impidiendo cualquier modificación no autorizada en el lazo de control.



Figura 42: Pestaña login.

Fuente: T. Chasi, 2025

La segunda ventana del interfaz HMI, mostrada en la figura 43, está diseñada para usuarios con acceso autorizado, permitiendo la modificación de parámetros críticos del sistema de control, esta ventana se despliega únicamente tras la autenticación exitosa mediante la pestaña login.



Figura 43: Ventana modo de operación

Fuente: T. Chasi, 2025

En esta pantalla se presentan múltiples secciones organizadas para facilitar la interacción del operador con el sistema:

La primera sección que se encuentra en la segunda ventana del HMI es el modo de operación donde encontramos:

- Selector de estado de tres posiciones: M (manual), OFF (apagado) y A (automático).
- Botones de marcha y paro y dos luces pilotos que nos indica el estado del sistema.

La segunda sección del interfaz corresponde a la unidad hidráulica, la cual opera a la par del modo manual permitiendo al usuario autorizado encender o apagar la bomba y asignar directamente parámetros esenciales como la frecuencia de operación y el setpoint de nivel, en esta misma ventana se visualizan en tiempo real las variables críticas del proceso, incluyendo la frecuencia (FREC) del variador, el nivel (NIVEL) del fluido en el sistema y el flujo (FLUJO) medido en m³/h, lo que proporciona al operador una supervisión inmediata y control directo sobre el comportamiento del sistema hidráulico.

La tercera sección de la interfaz corresponde a la configuración del controlador, donde el usuario autorizado puede seleccionar entre dos modos de control: PID u On/Off para ajustar sus respectivos parámetros según las necesidades del proceso. En el caso del controlador PID, se permite modificar los valores de ganancia proporcional (Kp), integral (Ki) y derivativa (Kd), lo que brinda un control más preciso y estable del sistema, mientras que el modo On/Off ofrece una alternativa más simple, basada en la activación o desactivación del actuador según el valor de asignado, además, cada uno de los modos de control disponibles en esta sección cuenta con una memoria interna que almacena hasta cuatro configuraciones distintas, lo que permite al usuario seleccionar entre los valores previamente guardados del controlador PID según las necesidades del proceso.

En esta ventana, se visualiza en la barra superior información relevante del sistema, incluyendo la hora, la fecha actual y el nombre del usuario que ha iniciado sesión, lo cual permite mantener un registro claro de la actividad y del responsable de las acciones realizadas; adicionalmente, en la parte inferior de la sección de configuración del controlador, se encuentra un ícono representado por una ventana con un engranaje, que al ser presionado despliega una pestaña que se superpone mostrada en la figura 44 donde

encontraremos una base de datos interna de parámetros PID, diseñada para facilitar la gestión y transferencia de configuraciones entre el sistema de recetas y el PLC.

Los registros que se encuentran en la base de datos están identificados por un número de controlador y contienen los valores concretos de los coeficientes de control: Kd (derivativo), Ki (integral) y Kp (proporcional), los valores son susceptibles de ser consultados y transmitidos entre dos secciones primordiales:

- Receta: Es la configuración escogida por el usuario, que tiene la posibilidad de ser editada o importada desde la base de datos.
- PLC: Muestra los parámetros actualmente activos en el controlador lógico programable.

La interfaz incluye botones de transferencia bidireccional:

- **RECETA → PLC:** Carga los valores seleccionados desde la receta al PLC, activando esa configuración en el sistema de control.
- **PLC → RECETA:** Permite guardar los parámetros actuales del PLC en la base de datos como una nueva receta.

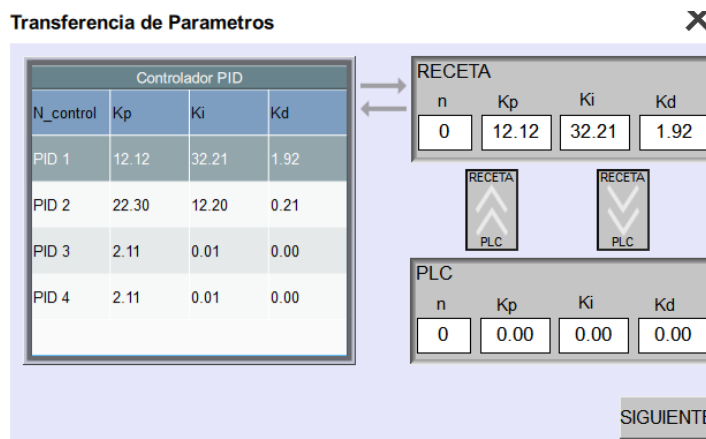


Figura 44: Base de datos controlador PID

Fuente: T. Chasi, 2025

Además, al seleccionar el botón siguiente dentro de la interfaz de esta pestaña, se accede de forma secuencial a las secciones correspondientes a la base de datos del controlador On/Off, ilustrado en la figura 45, se observa la configuración de los límites de nivel mostrada en la figura 47, en esta modalidad se registran parámetros como la frecuencia

de activación y el comportamiento del sistema durante los cambios de nivel y los límites de salida, mientras que en la sección de límites de nivel se definen rangos mínimos y máximos de saturación, todo ello organizado en tablas similares a la del PID y con funciones de transferencia entre la receta y el PLC.

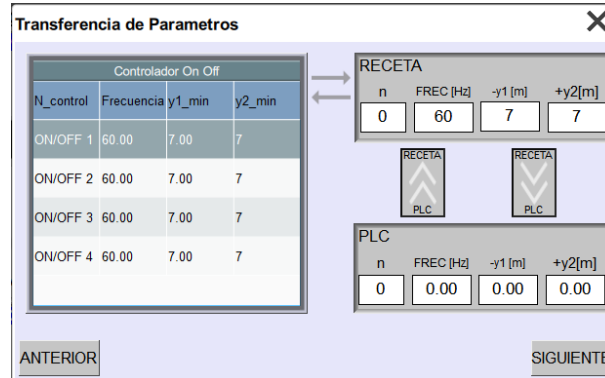


Figura 45: Base de datos controlador On/Off

Fuente: T. Chasi, 2025

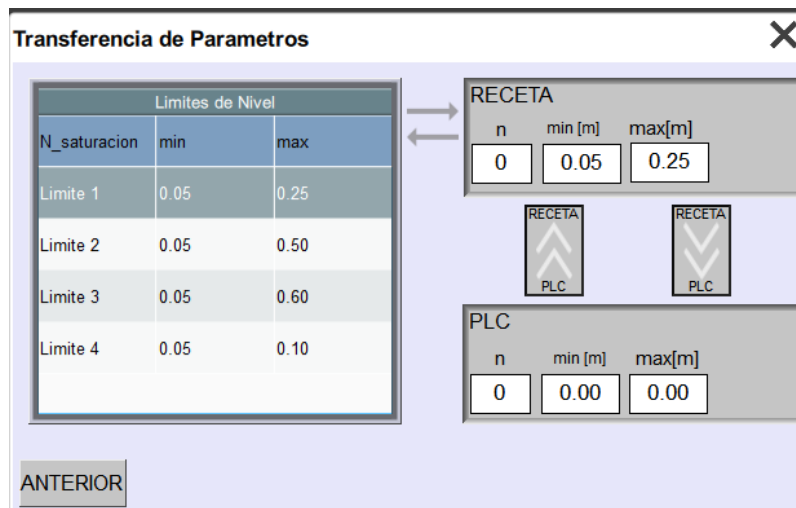


Figura 46: Base de datos de límites de nivel

Fuente: T. Chasi, 2025

En la tercera opción de menú el HMI presenta una última ventana destinada exclusivamente al operador, en la cual se permite la selección del modo de control On/Off o PID y la modificación del valor del setpoint de nivel, además de ofrecer una visualización clara y en tiempo real del comportamiento del proceso, esta interfaz representada por la figura 47, está diseñada con un enfoque práctico y seguro, limitando el acceso a funciones avanzadas y mostrando únicamente los elementos esenciales como el nivel del tanque, el flujo del sistema y el estado de los componentes.

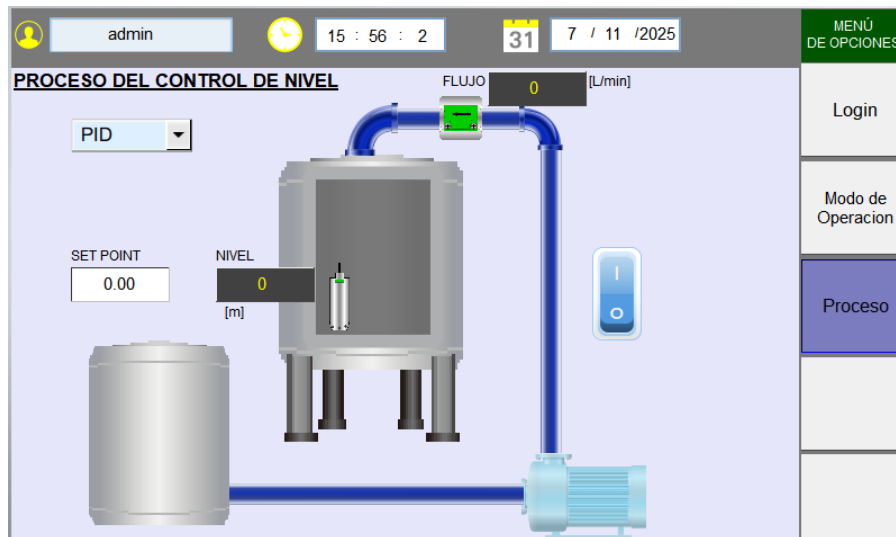


Figura 47: Ventana proceso

Fuente: T. Chasi, 2025

Finalmente, en la ventana de gráficas, mostrada en la Figura 48 nos permite observar de manera continua y en tiempo real la evolución de las principales variables del sistema: el nivel del tanque (m) y la frecuencia de la bomba (Hz), estas dos magnitudes son fundamentales para evaluar el comportamiento dinámico del proceso de control, la estabilidad del lazo PID y la respuesta del variador de frecuencia ante los cambios de consigna.



Figura 48: Ventana gráficas

Fuente: T. Chasi, 2025

En la parte superior de la interfaz se visualiza la gráfica correspondiente al nivel, lo que permite observar en tiempo real la variación de la señal medida. representada en unidades de metros, esta tendencia muestra la variación de la altura del fluido dentro del tanque principal permitiendo al operador verificar si el sistema mantiene el nivel dentro del rango establecido por el setpoint configurado en la pantalla de proceso, gracias a la actualización continua de los valores provenientes del transmisor de nivel KX-132V, la gráfica refleja en tiempo real las oscilaciones o perturbaciones generadas en el sistema, lo que facilita la evaluación de la respuesta del controlador PID o del modo On/Off, según el tipo de operación activo.

La segunda gráfica representa la frecuencia de operación de la bomba, medida en Hertz, esta tendencia está directamente vinculada con la señal de salida del variador de frecuencia y permite visualizar el comportamiento del actuador final en función de la demanda del sistema, mediante esta gráfica es posible identificar la relación directa entre la frecuencia aplicada a la bomba y el nivel alcanzado en el tanque por lo cual resulta esencial para el análisis del desempeño energético y de la eficiencia del control.

La tercera gráfica representa el flujo del sistema, que se expresa en metros cúbicos (m³). Esta variable representa el flujo de salida del tanque, que está directamente relacionado con la velocidad de la bomba controlada por el variador de frecuencia. Al realizar la operación, el valor del flujo se actualiza en tiempo real, esto posibilita que se vea cómo cambia a medida que la frecuencia es alterada por el controlador PID o por la lógica ON/OFF.

La supervisión del flujo es esencial para estudiar la eficiencia hidráulica del sistema y verificar que el caudal sea congruente con el nivel del tanque. Asimismo, este gráfico posibilita detectar potenciales pérdidas o inestabilidades en la línea, garantizando que el sistema conserve una conducta proporcional entre la frecuencia empleada y el flujo conseguido.

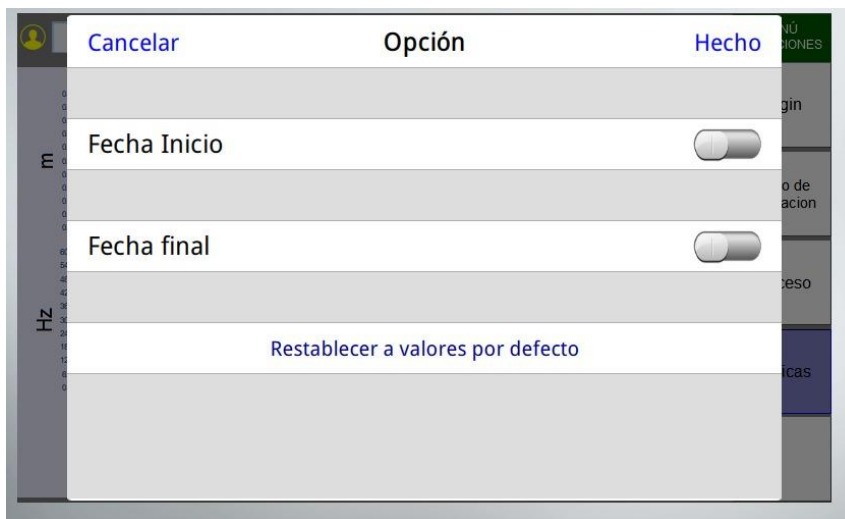


Figura 49: *Ventana gráficas*

Fuente: T. Chasi, 2025

Cada una de las gráficas cuenta con un ícono de engranaje ubicado en la esquina superior derecha el cual permite acceder a un panel de configuración avanzada mostrado en la figura 49, esta ventana emergente brinda la posibilidad de ajustar los parámetros de visualización temporal de las tendencias, ofreciendo opciones para definir la fecha de inicio y la fecha final del registro a analizar, mediante estos controles el operador puede seleccionar intervalos específicos de tiempo para estudiar el comportamiento histórico del sistema y comparar distintos periodos de operación.

Asimismo, dentro de esta misma ventana se dispone de la función “Restablecer a valores por defecto” que permite regresar rápidamente las gráficas a su estado original, mostrando únicamente los valores en tiempo real sin aplicar filtros de tiempo, esta característica es especialmente útil cuando se requiere reiniciar el monitoreo desde cero o cuando se han realizado varios ajustes previos en las fechas de visualización.

La versatilidad del HMI se fortalece con la incorporación de esta herramienta de configuración, ya que brinda tanto una visualización en tiempo real como una capacidad elemental para realizar análisis, sin requerir el uso de plataformas externas permitiendo al operador detectar de manera sencilla sucesos anómalos, tiempos de respuesta que se prolongan o variaciones no deseadas en la frecuencia y el nivel controlados, lo que ayuda a optimizar el sistema en general y a mantenerlo de forma predictiva.

3.3.9 Partición de memoria del microPLC Arduino Opta

Para el inicio del desarrollo del sistema se realizó la partición de memoria del Opta, con el objetivo de aprovechar los dos núcleos del PLC, esta configuración nos permite trabajar de forma paralela con las funciones del Opta y el Arduino IDE.



Figura 50: Selección puerto

Fuente: T. Chasi, 2025

El Opta utiliza una arquitectura de doble núcleo que optimiza el desempeño del sistema al separar las tareas de control y comunicación, el núcleo principal M4 se encarga de ejecutar la lógica de control desarrollada bajo los estándares IEC 61131-3, gestionando las entradas y salidas, el control PID y la comunicación local por Modbus RS-485.

El segundo núcleo M7 opera en paralelo y está dedicado a las funciones de conectividad, administración de la nube y servicios IIoT, desde este núcleo se gestionan las tareas de comunicación Wi-Fi, la transmisión de datos a la plataforma Arduino IoT Cloud y los procesos de actualización de firmware.

Esta división funcional garantiza que las operaciones críticas de control mantengan su estabilidad y tiempo de respuesta, incluso cuando se ejecutan procesos simultáneos de

transmisión o almacenamiento de datos. De esta manera, se logra un equilibrio eficiente entre rendimiento, seguridad y flexibilidad operativa dentro del entorno industrial.

La configuración de la partición de memoria se realiza desde el entorno de desarrollo del Arduino IDE, mediante la interfaz de configuración del dispositivo Opta como se muestra en la figura 50 en este paso se selecciona el tipo de puerto de comunicación y el modo de carga del firmware utilizando la opción “Automatic (BootLoader)”, la cual permite que el dispositivo sea detectado y programado sin necesidad de configuraciones manuales adicionales.

3.3.10 Actualización de firmware del PLC Arduino Opta

Como siguiente paso, se realizó la actualización del firmware del Opta usando un sketch de servicio cargado desde Arduino IDE, este sketch proporcionado por Finder (código mostrado en el anexo 1) se ejecuta en el núcleo M7 y automatiza el proceso de actualización: prepara la memoria, crea particiones de trabajo y copia el firmware Wifi necesarios para la comunicación.

3.3.11 Conexión MODBUS

Para el intercambio de datos entre los diferentes equipos del sistema, el variador de frecuencia, el PLC Arduino Opta y el interfaz HMI, se usa el protocolo Modbus, uno de los estándares de comunicación industrial más utilizados a nivel mundial por su simplicidad, confiabilidad y compatibilidad con múltiples dispositivos.

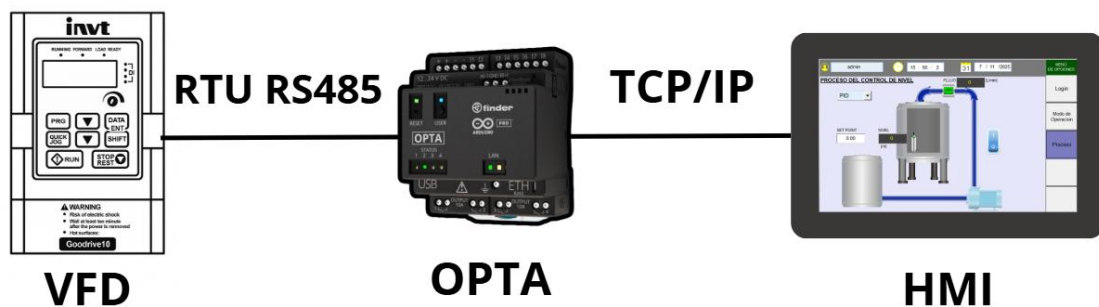


Figura 51: *Comunicación Modbus*

Fuente: T. Chasi, 2025

Modbus es un protocolo maestro-esclavo en este sistema, el Opta actúa como maestro, solicitando información o enviando comandos, mientras que los demás dispositivos

funcionan como esclavos como se muestra en la figura 51, respondiendo a esas peticiones, cada equipo del sistema se comunica mediante una capa física diferente del mismo protocolo, de acuerdo con sus funciones y requerimientos técnicos.

3.3.11.1 Conexión Modbus RS-485 entre el PLC y el variador de frecuencia

La comunicación entre el Opta y el variador de frecuencia INVT Goodrive10 Series se estableció utilizando el protocolo Modbus RTU sobre la capa física RS-485, con el objetivo de garantizar una transmisión estable, rápida y confiable de los datos de control.

En este sistema, el Opta funciona como equipo maestro; es responsable de mandar las órdenes de control y pedir datos al variador, mientras que el variador de frecuencia opera como esclavo, cumpliendo con las solicitudes y llevando a cabo las acciones según los parámetros establecidos.

La comunicación fue establecida con los siguientes parámetros estándar:

- Velocidad de transmisión (baud rate): 9600 bps
- Bits de datos: 8
- Paridad: Ninguna (N)
- Bits de parada: 1
- Modo de transmisión: RTU (Remote Terminal Unit)

La conexión física se llevó a cabo desde el puerto RS-485 del Opta hacia los terminales de comunicación del variador de frecuencia (A+ y B-), garantizando que la polaridad en las conexiones fuera la adecuada. El software del variador se configuró para que el Opta pueda reconocerlo dentro de la red al habilitar el modo de comunicación Modbus y asignarle una dirección esclavo ID5 específica.

El Opta utiliza este canal de comunicación para enviar órdenes de control que modifican la frecuencia de salida del variador y, en consecuencia, regulan la velocidad de la bomba. El variador también proporciona información como la frecuencia real y el estado operativo, que posteriormente se visualizan en la interfaz HMI.

3.3.11.2 Conexión Modbus TCP/IP entre el PLC y HMI

La comunicación entre el Opta y el interfaz HMI se estableció mediante el protocolo Modbus TCP/IP, que permite el intercambio de datos a través de una red Ethernet local, esta conexión posibilita que el HMI supervise en tiempo real las variables del proceso, envíe comandos al PLC y muestre el estado general del sistema.

El Modbus TCP/IP, en contraposición al Modbus RTU que se emplea con el variador de frecuencia, está fundamentado en paquetes de red que se envían a través del protocolo TCP/IP, asegurando una comunicación fiable, veloz y que puede integrarse sin dificultad con sistemas IIoT y plataformas basadas en la nube.

El Opta, que opera como servidor Modbus TCP/IP, presenta los registros donde están las variables del proceso, por su parte, el HMI tiene la función de ser el esclavo y es responsable de leer y escribir esos registros para actualizar o mostrar información en pantalla.

La configuración en el software IDE de PLC de Arduino (componente lógico 3.2.1) se llevó a cabo dentro del bloque ModbusTCP_cfg, en el que se establecieron las variables compartidas con HMI y los parámetros básicos de la red, simultáneamente, en el software EasyBuilder Pro, se eligió el controlador "Modbus TCP/IP (Client)" y se introdujeron los mismos datos de IP, puerto y tipo de registro.

En el HMI se configuraron palabras de memoria (LW) y bits (MX) que representan las principales variables del sistema; por ejemplo, el registro LW23 contiene el setpoint de nivel que es el valor de referencia que define la altura deseada del líquido en el tanque. El registro LW31 almacena la lectura del nivel medido proveniente del transmisor KX-132V, cuya señal de voltaje de 0-10 es convertida por el Opta a una variable digital de tipo float para su procesamiento.

De manera similar, la frecuencia de salida del variador se asocia al registro LW30, lo que permite visualizar en tiempo real la velocidad de la bomba y su comportamiento dentro del proceso. Los parámetros del controlador PID se distribuyeron en los registros LW5, LW7 y LW9, correspondientes a las ganancias proporcional (Kp), integral (Ki) y derivativa (Kd), estos valores pueden ser modificados desde el HMI para ajustar la respuesta del sistema según las condiciones operativas.

Adicionalmente, se utilizaron registros adicionales como el de LW15 para el modo de operación, que posibilita la elección entre los estados manual, automático u Off, y el bit MX0.0, que señala si la bomba está encendida o apagada.

Esta estructura de mapeo de variables establece un canal de comunicación fluido entre el HMI y el OPTA, permitiendo que las consignas enviadas desde la interfaz se reflejen inmediatamente en el controlador, y que las mediciones de flujo y nivel se actualicen de manera continua en la pantalla.

3.3.12 Configuración IIoT con Arduino Cloud mediante PLC IDE

Para lograr la comunicación IIoT y la visualización remota de las variables del proceso, se configuró la conexión del Opta con la plataforma Arduino Cloud, utilizando el entorno de programación Arduino PLC IDE, este entorno permite sincronizar las variables del proyecto con la nube sin necesidad de modificar la lógica de control principal, aprovechando la arquitectura de doble núcleo del Opta.

El primer paso consistió en registrar el Opta en la plataforma Arduino Cloud luego desde el panel principal se seleccionó la opción “Add device/Arduino Opta (PLC IDE)”. El sistema produjo automáticamente un identificador único (UUID) y las credenciales requeridas para una comunicación segura:

- ID del dispositivo: es la dirección exclusiva del Opta en la nube.
- Clave secreta: clave privada utilizada para la autenticación de TLS.

Estas credenciales se copiaron y guardaron para configurarlas posteriormente en el Arduino PLC IDE, en el Arduino PLC IDE, se abrió el proyecto principal y se accedió al menú “Resources/Cloud Connector”, donde se habilita el soporte de nube. Dentro de esta sección se ingresaron las credenciales obtenidas en el paso anterior:

ID: ad69005b-ba61-418c-826e-3df19e9ac4c9

FQBN: arduino:mbed_opta:opta

Serial Number: 002500283532511339353732

Dentro del PLC IDE, se definieron las variables que se enviarán o recibirán desde la nube, posteriormente en el apartado Cloud Variables Mapping, se vinculó cada variable local con su respectiva variable en la nube, previamente creada en el panel de Arduino Cloud/Thing/Variables, de esta forma, el Opta envía continuamente los valores

actualizados de proceso y recibe los setpoints o comandos de control desde el entorno web.

El Opta puede conectarse a internet mediante su puerto Ethernet o mediante Wi-Fi, dependiendo del modelo, en este proyecto, se utilizó ethernet por su mayor estabilidad en entornos industriales.

En el PLC IDE se establecieron los siguientes parámetros de la red:

IP Address: 192.168.1.10

Subnet Mask: 255.255.255.0

Gateway: 192.168.1.1

DNS: 8.8.8.8

Esta dirección IP es compatible con la configuración de la red local que alberga el variador de frecuencia y el HMI, lo que posibilita que el sistema de control y la conexión a la nube operen al mismo tiempo.

Una vez configuradas las variables y parámetros, se colocó el PLC en modo RUN y se verificó el estado de conexión en el panel Cloud Monitor del IDE cuando la conexión es exitosa el indicador muestra el estado “Connected”, y las variables comienzan a transmitirse en tiempo real.

El protocolo MQTT, que asegura un envío de datos fiable y con poco retardo, es la base de la comunicación, esta configuración posibilita la supervisión y el registro del comportamiento del proceso desde cualquier lugar, sin poner en riesgo la estabilidad del control local.

El núcleo M4 mantiene la gestión de las funciones ON/OFF y PID, mientras que el núcleo M7 se encarga de la difusión de datos en la nube.

3.3.13 Controladores aplicados en el sistema

En esta sección se presentan los controladores implementados dentro de este sistema, es importante destacar que el enfoque de este módulo no es el diseño del proceso físico en sí, sino la integración tecnológica basada en un entorno IIoT utilizando el Opta.

Los controladores utilizados se aplicaron con el propósito de demostrar las capacidades de procesamiento, comunicación y control del Opta, validando su desempeño dentro de una arquitectura moderna de automatización, ambos controladores permiten evidenciar

que el Opta puede ejecutar tareas de control industrial en tiempo real, comunicarse con diferentes protocolos y, a la vez transmitir datos hacia un entorno de supervisión HMI y subirlos la nube.

Variable controlada variable de control señal de proceso setpoint valor deseado ecuación general del tanque sintonización de PID prueba y error.

3.3.13.1 Controlador PID

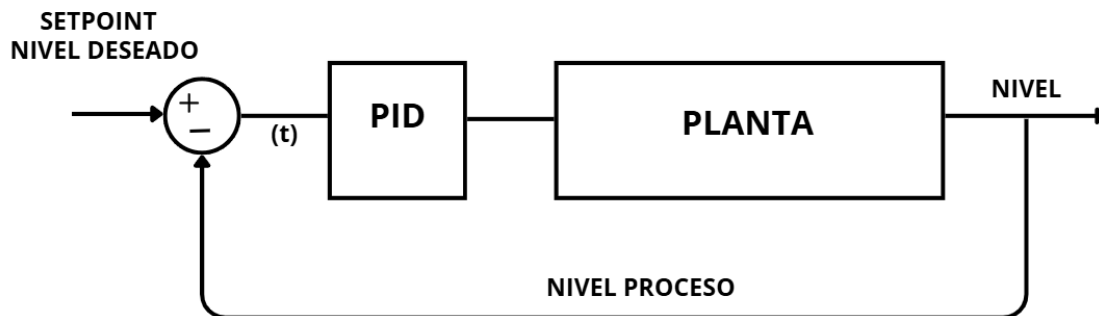


Figura 52: Control de lazo cerrado

Fuente: T. Chasi, 2025

El diagrama de la figura 52 representa el funcionamiento del lazo de control PID aplicado al sistema de nivel del tanque, el nivel deseado (setpoint) se compara con el nivel real del proceso medido por el sensor de nivel, la diferencia entre ambos valores genera una señal de error, la cual es procesada por el controlador Opta y posteriormente enviada hacia el controlador PID.

En este sistema:

- La variable controlada es el nivel del tanque.
- La variable de control es la frecuencia de salida del variador.
- El actuador es la bomba de agua.

El controlador determina la variable de control, que aquí es la frecuencia que se manda al variador, este regulador controla la bomba, actuando como elemento final de control o actuador, y modifica el flujo de agua hacia el tanque con el fin de mantener el nivel por debajo del valor de referencia.

El Opta compara el valor medido con el setpoint y obtiene la diferencia entre ambos, denominada error:

$$e(t) = SP - MV \quad (1)$$

A partir de ese error, el controlador calcula una señal de salida (CV) que se envía al variador de frecuencia, el variador ajusta la velocidad de la bomba lo que modifica el caudal y por ende el nivel del tanque. De esta forma el sistema corrige continuamente las desviaciones para mantener la variable controlada lo más cerca posible del valor de referencia.

El comportamiento del sistema de nivel puede modelarse con una ecuación diferencial de primer orden que representa la variación del volumen del tanque en función del flujo de entrada y salida:

$$A \frac{dh(t)}{dt} = Q_{in}(t) - Q_{out}(t) \quad (2)$$

donde:

- A : área transversal del tanque (m^2)
- $h(t)$: nivel del fluido (m)
- $Q_{in}(t)$: caudal de entrada (m^3/s)
- $Q_{out}(t)$: caudal de salida (m^3/s)

Las tres acciones que lo conforman determinan la conducta del controlador:

- **Acción Proporcional (P):** Produce una reacción que es proporcional al error en el momento. Si el error es grande, la rectificación también será mayor. el sistema responde con mayor rapidez al incrementar K_p aunque es posible que esto provoque oscilaciones.
- **Acción Integral (I):** Elimina el error persistente en estado estable y toma en cuenta la acumulación del error a lo largo del tiempo. Un K_i muy alto puede hacer que el sistema se vuelva inestable.

- Acción Derivativa (D): responde a la rapidez con que se modifica el error. Su función es prever los cambios repentinos y atenuar la reacción del sistema.

Estas tres acciones funcionan en conjunto en el Opta, que calcula la salida en tiempo real y la envía al VFD a través de Modbus TCP/I. El sistema se ajustó manualmente, calibrando K_p , K_i y K_d hasta conseguir una respuesta estable, con escaso sobre impulso y sin oscilaciones excesivas. El resultado es un control fluido y fiable del nivel o flujo, donde el Opta muestra su habilidad para implementar algoritmos de control sofisticados y comunicarse al mismo tiempo con otros dispositivos industriales.

El código del control PID se cargó dentro del sketch de Arduino alojado en la memoria destinada en el entorno IDE, donde se trabaja con lenguaje C utilizando la librería PID_V2 versión 1.0.8, esta librería permite calcular las tres acciones del controlador proporcional, integral y derivativa en tiempo real, de forma continua y precisa, durante la configuración, se establecieron los valores iniciales de K_p , K_i y K_d , además del tiempo de muestreo y los límites de salida del controlador.

3.3.13.2 Controlador ON-OFF

Se introdujo un controlador ON/OFF (todo o nada) como una opción de control sencilla dentro del mismo sistema. Este tipo de controlador se emplea para cotejar el comportamiento dinámico del proceso en comparación con un método de control continuo, y así demostrar la variación en la estabilidad y reacción del sistema.

El control ON/OFF trabaja también en un lazo cerrado, pero su acción de control se basa únicamente en dos estados encendido (ON) o apagado (OFF).

En el sistema creado, esta lógica se configuró en el microPLC Arduino Opta, fijando una condición de comparación entre la altura medida del tanque y la altura de referencia (setpoint).

Fundamento de operación

- El PLC pone en marcha la bomba (estado ON) cuando el tanque se encuentra por debajo del setpoint.
- Si el nivel llega al valor de referencia o lo supera, el PLC apaga la bomba (estado OFF).

La fluctuación natural alrededor del punto de consigna es provocada por esta conducta, dado que el nivel aumenta y disminuye sin cesar en un rango determinado.

Para evitar un número excesivo de encendidos y apagados, se añadió una banda de histéresis que define una zona muerta o margen de tolerancia, el sistema no cambia de estado hasta que la variable de proceso sobrepasa los límites de esta banda, protegiendo así los componentes eléctricos y mecánicos del sistema [66].

La relación de control ON/OFF con histéresis puede expresarse mediante la siguiente ecuación:

$$u(t) = \begin{cases} 1, & \text{si } h(t) < SP - \Delta h \\ 0, & \text{si } h(t) > SP + \Delta h \end{cases} \quad (3)$$

donde

- $u(t)$: salida del controlador (1 = bomba encendida, 0 = bomba apagada)
- $h(t)$: nivel actual del tanque
- SP: nivel de referencia o setpoint
- Δh : banda de histéresis

Esta función describe el comportamiento binario del sistema: cuando el nivel cae por debajo del valor inferior, la bomba se activa; cuando supera el valor superior, se desactiva. Como resultado, se produce una oscilación controlada que mantiene la variable de proceso dentro de los límites establecidos por el operador.

El comportamiento observado del controlador ON/OFF presenta una respuesta discontinua, el nivel del tanque oscila entre dos valores cercanos al setpoint, generando pequeñas fluctuaciones que dependen del tamaño de la banda de histéresis y de la inercia del proceso, aunque este método no logra un control tan estable ni suave como el PID, sí permite mantener la variable dentro de un rango aceptable y es suficiente para procesos simples o de baja sensibilidad [67].

3.3.14 Pruebas aplicadas a los controladores

Con el sistema completamente ensamblado y la configuración de los controladores finalizada, se procedió a realizar las pruebas de funcionamiento, estas pruebas tuvieron como finalidad comprobar la respuesta dinámica del sistema de nivel ante los distintos métodos de control implementados: ON/OFF y PID, verificando el comportamiento real del proceso respecto a las variables de flujo y nivel medidas a través de los sensores.

Durante las pruebas el tanque acrílico principal (Figura 53) fue el componente central del proceso, donde se observó el comportamiento del nivel de agua en función de las señales de control generadas por el Opta.

Se usó la plataforma IoT Cloud y el HMI para ver los datos recolectados en tiempo real, lo cual permitió examinar la exactitud, estabilidad y tiempo de respuesta de cada una de las estrategias de control.

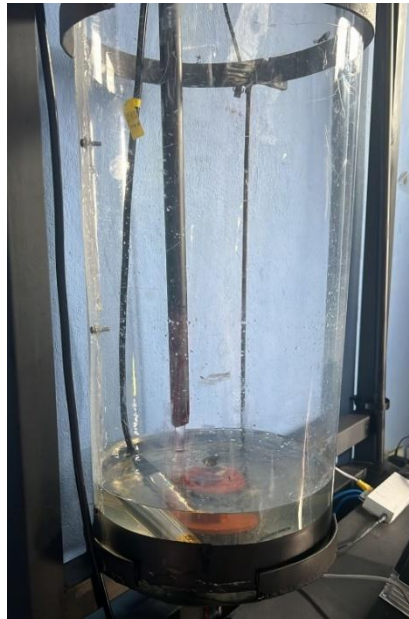


Figura 53: *Tanque acrílico principal*

Fuente: T. Chasi, 2025

3.3.14.1 Pruebas aplicadas al controlador ON/OFF

Para validar el funcionamiento de los controladores implementados en el sistema se realizaron pruebas experimentales con el módulo de flujo y nivel, observando el comportamiento de las variables críticas: nivel del tanque (m) y frecuencia de la bomba (Hz), las mediciones se visualizaron en tiempo real mediante el HMI el cual registró las tendencias de ambas variables.

Durante la prueba, se estableció un setpoint de nivel de 0.25 m, con una banda de histéresis configurada para evitar conmutaciones frecuentes de la bomba, el principio de funcionamiento consistió en que el sistema activa la bomba cuando el nivel cae por debajo del límite inferior y la desactiva al superar el límite superior del rango establecido.



Figura 54: Prueba 1

Fuente: T. Chasi, 2025

En la figura 54 se observa que el nivel del tanque aumenta de manera progresiva desde aproximadamente **0.18 m hasta 0.27 m**, momento en el cual el controlador desactiva la bomba, el comportamiento demuestra que el sistema responde adecuadamente al alcanzar el valor superior de la banda de histéresis, manteniendo el nivel dentro del rango deseado sin sobrepasar el setpoint.



Figura 55: Prueba 2

Fuente: T. Chasi, 2025

En la gráfica de la figura 55 tras el apagado de la bomba, el nivel del tanque desciende gradualmente debido al vaciado natural del sistema, llegando cerca de 0.10 m. Cuando se alcanza el límite inferior de la banda, el controlador reactiva automáticamente la bomba, lo cual se refleja en el aumento de la frecuencia de salida del variador, este comportamiento confirma que el lazo de control cerrado ON/OFF responde correctamente a las variaciones del proceso, restableciendo el nivel sin intervención manual.



Figura 56: Prueba 3

Fuente: T. Chasi, 2025

La figura 56 muestra el ciclo repetitivo del sistema una vez alcanzado el estado estacionario, el nivel del tanque oscila entre los límites definidos (aprox. 0.15 m y 0.30 m), mientras que la bomba alterna entre los estados de encendido y apagado de manera periódica.

Además, se determinó la banda de histéresis temporal (Δt_h), definida como la diferencia entre los tiempos de apagado y encendido de la bomba:

$$\Delta t_h = t_2 - t_1 \quad (4)$$

A partir de las gráficas obtenidas, se registró el apagado de la bomba en $t_1 = 10:06:00$ y su reactivación en $t_2 = 10:06:44$, por lo tanto:

$$\Delta t_h = 44 \text{ s} \quad (5)$$

Este intervalo de 44 segundos representa el tiempo en que el nivel del tanque desciende desde el límite superior hasta el inferior de la banda de control. Una respuesta controlada y estable es la que se señala con esta histéresis, evitando conmutaciones veloces del actuador y garantizando un funcionamiento fiable del sistema, la diferencia estimada entre ambos puntos, en términos de nivel, fue de 0.30 metros (máxima) y 0.10 metros (mínima), lo cual equivale a una banda de histéresis de 0.20 metros.

Esta amplitud es apropiada para la conducta hidráulica del módulo, pues conserva el proceso dentro de límites seguros y posibilita una observación nítida del ciclo de llenado y vaciado.

3.3.14.2 Pruebas aplicadas al controlador PID

Con el fin de evaluar el desempeño del controlador PID implementado en el sistema de control de nivel, se configuraron los parámetros de ganancia mediante el panel del HMI: $K_p = 10.00$, $K_i = 1.00$ y $K_d = 0.02$, como se muestra en la figura 57, estos valores fueron obtenidos por sintonización manual con el objetivo de lograr una estabilidad y usado como ejemplo la receta 0 del PID.

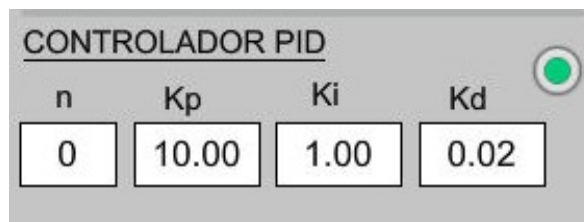


Figura 57: *Parámetros*

Fuente: T. Chasi, 2025

En el transcurso de la prueba, se fijó un punto ideal en 0.25 m para el nivel del tanque, mientras que el sistema estaba funcionando con condiciones normales de alimentación y carga. La figura 58 muestra que la línea azul (el nivel del tanque) se dirige con rapidez a alcanzar el valor de referencia (línea verde), estabilizándose en torno al punto de consigna con una oscilación mínima. La frecuencia de la bomba (línea roja) oscila gradualmente entre 6 Hz y 48 Hz, lo que demuestra el efecto modulante del controlador, que modifica la velocidad del motor.



Figura 58: *Parámetros*

Fuente: T. Chasi, 2025

El comportamiento dinámico del sistema muestra una respuesta rápida y estable, alcanzando el valor de referencia en aproximadamente veinte segundos, con un sobre impulso menor al tres por ciento y sin presencia de oscilaciones sostenidas, una vez alcanzado el régimen permanente, el nivel se mantiene constante dentro de un margen de variación de aproximadamente ± 0.01 m, lo cual evidencia la correcta acción del término integral, que elimina el error en estado estacionario, y del término derivativo, que contribuye a la amortiguación de las variaciones bruscas del proceso.



Figura 59: *Parámetros*

Fuente: T. Chasi, 2025

3.3.15 Integración IIoT con Arduino Cloud mediante PLC IDE

Para establecer esta comunicación se empleó el Opta, el cual actúa como unidad central de procesamiento encargada de ejecutar la lógica de control desarrollada en el entorno Arduino PLC IDE y, simultáneamente, gestionar el intercambio de información con la nube mediante el protocolo seguro MQTT.

Como parte inicial del proceso de integración, fue necesario realizar una actualización completa del firmware Wifi del Opta, asegurando la compatibilidad con las funciones de comunicación proporcionadas por Arduino IoT Cloud, este procedimiento se efectuó directamente desde el Arduino IDE, el código *WifiFirmwareUpdater* y las directrices a seguir se encuentra en el anexo 1 y 2, luego se verificó que el dispositivo no se encontrara reclamado en otra cuenta de Arduino Cloud, ya que un Opta registrado solo puede ser utilizado por el propietario que lo dio de registro inicialmente, esta restricción constituye una medida de seguridad esencial dentro del ecosistema IIoT, garantizando la integridad del acceso remoto.

Para agregar el dispositivo Opta a la cuenta de Arduino IoT Cloud es necesario realizar una conexión inicial mediante el cable USB directamente al Opta, una vez detectado el hardware, la plataforma Arduino Cloud solicita la descarga automática de una aplicación denominada Arduino create agent, esta herramienta es instalada en el equipo y permanece activa en segundo plano, mostrándose en la barra de tareas como un servicio en ejecución permanente como se muestra en la figura 61.

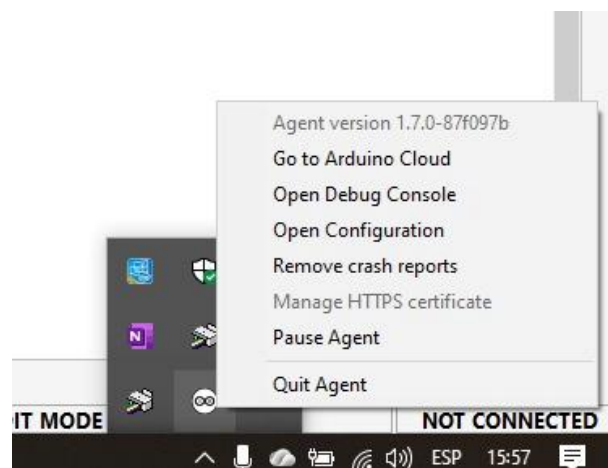


Figura 60: *Arduino Create Agent.*

Fuente: T. Chasi, 2025

La función del Arduino Create Agent es esencial dentro del proceso de integración IIoT, ya que actúa como puente de comunicación entre el hardware físico y la plataforma en la nube, gracias a esta aplicación, el Opta puede ser reconocido, configurado y vinculado correctamente con la cuenta del usuario en Arduino IoT Cloud.

Se procedió a la actualización individual de las librerías necesarias, tanto en Arduino IDE como en PLC IDE, ambas herramientas deben encontrarse sincronizadas esta se realiza mediante un código de PowerShell donde se va a actualizar el índice local que contiene las correcciones de seguridad obligatorias, si se utiliza la versión 1.0.8 o posterior del IDE de PLC (código se encuentra en el anexo 3), esta actualización se realiza automáticamente a través del IDE, puesto que el compilador del PLC depende de las mismas librerías que utiliza la plataforma IoT para gestionar funciones de conectividad, MQTT y provisión de dispositivos, durante este proceso se ejecutaron las líneas internas del Arduino IDE que sincronizan automáticamente los paquetes del Opta y los vinculan al entorno PLC IDE. Una vez completada la actualización el proyecto fue recompilado y cargado nuevamente en el Opta, habilitando correctamente la comunicación con Arduino IoT Cloud.

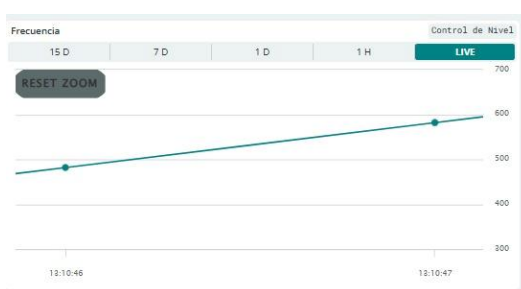


Figura 61: *Monitoreo en tiempo real*

Fuente: T. Chasi, 2025

En la figura 62 se presenta el panel de monitoreo en tiempo real del Arduino IoT Cloud, donde se visualiza la variable correspondiente a la frecuencia del variador de velocidad. Esta gráfica confirma que el Opta transmite datos de forma continua hacia la nube, permitiendo al usuario observar en vivo los cambios generados en el proceso físico, la tendencia ascendente registrada valida el funcionamiento del enlace MQTT y demuestra que la variable de frecuencia procesada y convertida previamente mediante el bloque TO_WORD en el PLC es sincronizada correctamente con la plataforma remota.

En la figura 63 muestra la información del dispositivo Opta registrado dentro del panel de administración de Arduino IoT Cloud, en este apartado se destacan datos relevantes

como: la red Wifi utilizada, la actividad reciente, el ID único del dispositivo, el número de serie, la versión de la librería IoT instalada y el firmware del módulo de conectividad.



Figura 62: Estado Online PLC

Fuente: T. Chasi, 2025

Esta información confirma que el dispositivo ha sido configurado correctamente y que se encuentra operativo dentro del entorno IIoT, el estado Online indica que la comunicación está establecida y lista para el intercambio de datos.

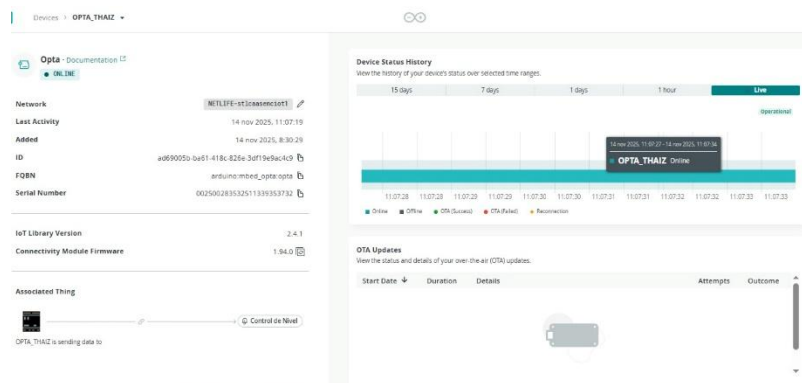


Figura 63: Estado Online PLC

Fuente: T. Chasi, 2025

En la figura 64 se presenta el historial de estado del Opta, donde se observa su conectividad continua durante la operación del sistema, este registro muestra el comportamiento del dispositivo frente a reconexiones, actualizaciones OTA y disponibilidad general, la presencia constante del estado Online confirma la estabilidad del enlace inalámbrico y demuestra que la actualización del firmware, la sincronización de librerías y la recopilación del proyecto fueron realizadas de manera adecuada, gracias a esta configuración el sistema consigue mantener un canal de comunicación entre el entorno físico y la nube.

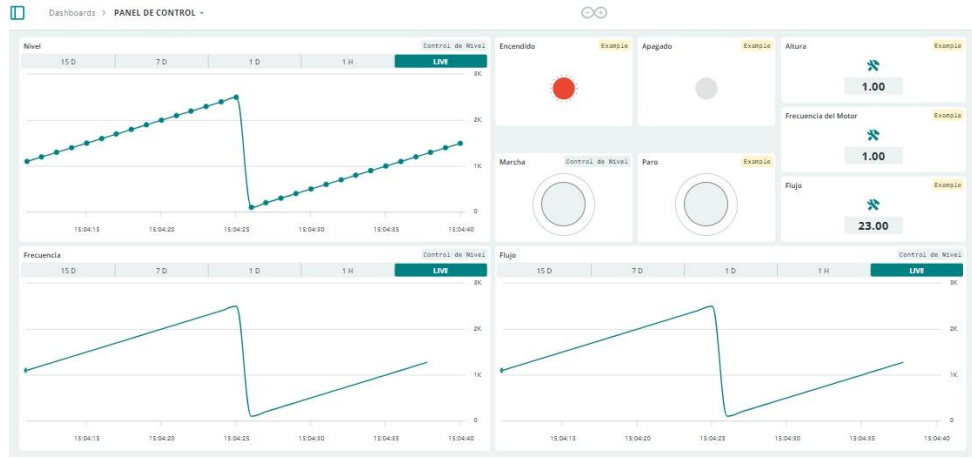


Figura 64: Interfaz Arduino Cloud

Fuente: T. Chasi, 2025

Finalmente, en la figura 65 se presenta la interfaz del croad donde se incluyen las gráficas en tiempo real del nivel del tanque, la frecuencia del variador y el caudal del sistema, permitiendo observar la dinámica completa del proceso durante las etapas de llenado, vaciado y estabilización.

El panel incluye indicadores de estado tales como encendido/apagado, marcha/paro y las lecturas numéricas de altura, frecuencia del motor y flujo, lo cual facilita la interpretación del comportamiento del sistema desde cualquier ubicación remota, esta vista consolidada demuestra la correcta integración del Opta con la plataforma IoT y confirma la operatividad del sistema IIoT implementado, posibilitando un monitoreo eficiente y continuo del proceso.



Figura 65: Acceso remoto

Fuente: T. Chasi, 2025

Asimismo, se comprobó el acceso remoto al sistema utilizando un teléfono móvil mediante la plataforma Arduino IoT Cloud, una vez que el Opta quedó registrado y vinculado a la cuenta correspondiente la visualización de las variables y gráficas en tiempo real se puede realizar desde la aplicación móvil sin necesidad de que el dispositivo estuviera conectado a la misma red local del PLC, esto se debe a que la comunicación se gestiona directamente a través de los servidores de Arduino Cloud, permitiendo que el usuario ingrese con sus credenciales desde cualquier red incluso con datos móviles, esta característica amplía el alcance del monitoreo, ya que el proceso puede supervisarse de forma continua desde ubicaciones remotas sin comprometer la estabilidad del control local.

CAPÍTULO IV

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- La comunicación entre el Arduino Opta y la plataforma Arduino IoT Cloud demostró ser estable y confiable durante todas las pruebas realizadas, el sistema mantuvo una conexión permanente, sin presentar desconexiones ni eventos de reconexión, lo que confirma la solidez del enlace MQTT y la estabilidad de la red utilizada, la latencia medida en la actualización de datos fue de aproximadamente 1 segundo, valor que se mantuvo constante durante el monitoreo continuo y que resulta adecuado para aplicaciones de supervisión remota en tiempo casi real.
- Los datos enviados desde el Opta hacia la nube se actualizaron con un retardo promedio de 1 segundo, tanto para variables críticas como nivel, frecuencia y flujo, este tiempo se encuentra dentro de los valores aceptables para sistemas IIoT que no requieren un control rígido en microsegundos, los comandos enviados desde la plataforma hacia el PLC mostraron un tiempo de respuesta inmediata seguido de un retardo natural de aproximadamente 1 segundo antes de observarse el cambio físico en los actuadores. Este comportamiento es típico en arquitecturas IIoT basadas en redes Wifi y protocolos MQTT.
- El variador de frecuencia respondió con estabilidad a las órdenes enviadas desde el PLC, requiriendo cerca de 1 segundo para reajustar su frecuencia ante un cambio de consigna, este tiempo coincide con la latencia global del sistema y se mantuvo constante durante las distintas pruebas, el proceso de llenado y vaciado del tanque operó de manera estable, sin presentar oscilaciones indeseadas, y mantuvo una dinámica suave durante los ciclos de control, lo que demuestra que la lógica implementada en el Opta es adecuada para esta aplicación.

- Se logró implementar un sistema de control de nivel funcional utilizando Arduino Opta, se estableció una comunicación IIoT completamente operativa con Arduino Cloud y se desarrolló un panel remoto que permitió monitorear en tiempo real las variables del proceso, se comprobó la viabilidad de integrar un variador de frecuencia con un microPLC de bajo costo, obteniendo un comportamiento estable y predecible del sistema.
- El sistema creado evidencia que es viable poner en marcha soluciones de automatización con acceso remoto, haciendo uso de tecnologías abiertas, asequibles y sencillas de integrar. La arquitectura sugerida es apropiada para ambientes industriales y educativos de pequeña magnitud, y su rendimiento demuestra que Arduino Opta puede llevar a cabo control básico y sincronización de datos con la nube sin interrupciones ni pérdidas. La estabilidad registrada, la latencia uniforme y la correcta respuesta del variador validan que el sistema puede ser empleado como base para aplicaciones IIoT de monitoreo en tiempo real.

4.2 Recomendaciones

- Para asegurar que el sistema funcione de manera estable, se aconseja dividir los núcleos del microPLC Arduino Opta y distribuir las tareas de comunicación y control de forma apropiada.
- Es necesario restablecer el código de fábrica del PLC antes de cargar un nuevo programa, con el fin de evitar errores de configuración o conflictos en la ejecución del proyecto.
- Se debe utilizar cableado apantallado en todas las conexiones de señal y comunicación, ya que el uso de cables sin protección puede generar interferencias electromagnéticas que afecten el correcto funcionamiento de los sensores y del controlador.
- Las tuberías del sistema deben estar selladas y correctamente ajustadas para prevenir vibraciones o fugas que puedan modificar las lecturas de nivel y flujo.
- Antes de iniciar la operación, se recomienda cebar las tuberías del sistema para asegurar que estén completamente llenas de líquido y así evitar daños en la bomba por funcionamiento en seco.

- Se recomienda usar tanques de forma cuadrada o rectangular en lugar de tanques cilíndricos, debido a que estos presentan mayores dificultades en el proceso de ensamblaje.
- El microPLC Arduino Opta puede tardar aproximadamente veinte minutos en compilar y cargar los programas, por lo tanto, se aconseja planificar las modificaciones del código y realizar pruebas por etapas para optimizar el tiempo de desarrollo.
- Verificar que los puertos COM no deben estar ocupados con otros dispositivos porque pueden generar errores en las subidas de códigos y bugs no deseados.
- Se recomienda para investigaciones futuras examinar el comportamiento del sistema a través de algoritmos de control más sofisticados, por ejemplo, el control difuso, PID adaptativo o el control predictivo fundamentado en modelos. Además, sería útil incorporar redes de sensores adicionales, experimentar con protocolos industriales por cable (Profinet, Modbus) y examinar cómo funciona el sistema en redes que son menos estables o que tienen mayores demandas de tiempo. Con estos avances se podría comparar el rendimiento del control actual con métodos más sólidos y extender las habilidades del sistema IIoT creado.
- Cuando el PLC se queda en modo running, no responde; se debe restablecer a fábrica con el programa Opta Arduino Factory reset.

ANEXOS

Anexo 1 código WifiFirmwareUpdater

```
#include "BlockDevice.h"#include "MBRBlockDevice.h"

#include "LittleFileSystem.h"
#include "FATFileSystem.h"
#include "wiced_resource.h"
#include "certificates.h"

#ifndef CORE_CM7
    #error Format QSPI flash by uploading the sketch to the M7 core instead of the M4 core.
#endif

using namespace mbed;

BlockDevice* root = BlockDevice::get_default_instance();
MBRBlockDevice wifi_data(root, 1);
MBRBlockDevice ota_data(root, 2);
MBRBlockDevice kvstore_data(root, 3);
MBRBlockDevice user_data(root, 4);
FATFileSystem wifi_data_fs("wlan");
FATFileSystem ota_data_fs("fs");
FileSystem * user_data_fs;

bool waitResponse() {
    bool confirmation = false;
    bool proceed = false;
    while (confirmation == false) {
        if (Serial.available()) {
            char choice = Serial.read();
            switch (choice) {
                case 'y':
                case 'Y':
                    confirmation = true;
                    proceed = true;
                    break;
                case 'n':
                case 'N':
                    confirmation = true;
            }
        }
    }
}
```

```

        proceed = false;
        break;
    default:
        continue;
    }
}
}
return proceed;
}

void printProgress(uint32_t offset, uint32_t size, uint32_t threshold, bool reset) {
    static int percent_done = 0;
    if (reset == true) {
        percent_done = 0;
        Serial.println("Flashed " + String(percent_done) + "%");
    } else {
        uint32_t percent_done_new = offset * 100 / size;
        if (percent_done_new >= percent_done + threshold) {
            percent_done = percent_done_new;
            Serial.println("Flashed " + String(percent_done) + "%");
        }
    }
}

void setup() {
    Serial.begin(115200);
    while (!Serial);

    Serial.println("\nWARNING! Running the sketch all the content of the QSPI flash
will be erased.");
    Serial.println("The following partitions will be created:");
    Serial.println("Partition 1: WiFi firmware and certificates 1MB");
    Serial.println("Partition 2: OTA 5MB");
    Serial.println("Partition 3: Provisioning KVStore 1MB");
    Serial.println("Partition 4: User data / OPTA PLC runtime 7MB"),
    Serial.println("Do you want to proceed? Y/[n]");
    if (true == waitResponse()) {

```

```

if (root->init() != BD_ERROR_OK) {
    Serial.println(F("Error: QSPI init failure."));
    return;
}

Serial.println("Do you want to perform a full erase of the QSPI flash before
proceeding? Y/[n]");
Serial.println("Note: Full flash erase can take up to one minute.");
bool fullErase = waitResponse();
if (fullErase == true) {
    Serial.println("Full erase started, please wait...");
    root->erase(0x0, root->size());
    Serial.println("Full erase completed.");
} else {
    // Erase only the first sector containing the MBR
    root->erase(0x0, root->get_erase_size());
}
MBRBlockDevice::partition(root, 1, 0x0B, 0, 1 * 1024 * 1024);
MBRBlockDevice::partition(root, 2, 0x0B, 1 * 1024 * 1024, 6 * 1024 * 1024);
MBRBlockDevice::partition(root, 3, 0x0B, 6 * 1024 * 1024, 7 * 1024 * 1024);
MBRBlockDevice::partition(root, 4, 0x0B, 7 * 1024 * 1024, 14 * 1024 * 1024);
// use space from 15.5MB to 16 MB for another fw, memory mapped
bool reformat = true;
if (!wifi_data_fs.mount(&wifi_data)) {
    Serial.println("\nPartition 1 already contains a filesystem, do you want to reformat it?
Y/[n]");
    wifi_data_fs.unmount();

    reformat = waitResponse();
}

if (reformat && wifi_data_fs.reformat(&wifi_data)) {
    Serial.println("Error formatting WiFi partition");
    return;
}

```

```

bool restore = true;
if (reformat || fullErase) {
    Serial.println("\nDo you want to restore the WiFi firmware and certificates? Y/[n]");
    restore = waitResponse();
}
if (reformat && restore) {
    flashWiFiFirmwareAndCertificates();
}
if (fullErase && restore) {
    flashWiFiFirmwareMapped();
}
reformat = true;
if (!ota_data_fs.mount(&ota_data)) {
    Serial.println("\nPartition 2 already contains a filesystem, do you want to reformat it?
Y/[n]");
    ota_data_fs.unmount();
    reformat = waitResponse();
}
if (reformat && ota_data_fs.reformat(&ota_data)) {
    Serial.println("Error formatting OTA partition");
    return;
}
Serial.println("\nDo you want to use LittleFS to format user data partition? Y/[n]");
Serial.println("If No, FatFS will be used to format user partition.");
Serial.println("Note: LittleFS is not supported by the OPTA PLC runtime.");
if (true == waitResponse()) {
    Serial.println("Formatting user partition with LittleFS.");
    user_data_fs = new mbed::LittleFileSystem("user");
} else {
    Serial.println("Formatting user partition with FatFS.");
    user_data_fs = new mbed::FATFileSystem("user");
}
reformat = true;
if (!user_data_fs->mount(&user_data)) {

```

```

Serial.println("\nPartition 4 already contains a filesystem, do you want to reformat it?
                Y/[n]");

user_data_fs->unmount();

reformat = waitResponse();

}

if (reformat && user_data_fs->reformat(&user_data)) {

    Serial.println("Error formatting user partition");

    return;

}

Serial.println("\nQSPI Flash formatted!");

}

Serial.println("It's now safe to reboot or disconnect your board.");

}

const uint32_t file_size = 421098;

extern const unsigned char wifi_firmware_image_data[];

void flashWiFiFirmwareAndCertificates() {

    FILE* fp = fopen("/wlan/4343WA1.BIN", "wb");

    uint32_t chunk_size = 1024;

    uint32_t byte_count = 0;

    Serial.println("Flashing WiFi firmware");

    printProgress(byte_count, file_size, 10, true);

    while (byte_count < file_size) {

        if(byte_count + chunk_size > file_size)

            chunk_size = file_size - byte_count;

        int ret = fwrite(&wifi_firmware_image_data[byte_count], chunk_size, 1, fp);

        if (ret != 1) {

            Serial.println("Error writing firmware data");

            break;

        }

        byte_count += chunk_size;

        printProgress(byte_count, file_size, 10, false);

    }

    fclose(fp);

    fp = fopen("/wlan/cacert.pem", "wb");

```

```

Serial.println("Flashing certificates");
chunk_size = 128;
byte_count = 0;
printProgress(byte_count, cacert_pem_len, 10, true);
while (byte_count < cacert_pem_len) {
    if(byte_count + chunk_size > cacert_pem_len)
        chunk_size = cacert_pem_len - byte_count;
    int ret = fwrite(&cacert_pem[byte_count], chunk_size, 1 ,fp);
    if (ret != 1) {
        Serial.println("Error writing certificates");
        break;
    }
    byte_count += chunk_size;
    printProgress(byte_count, cacert_pem_len, 10, false);
}
fclose(fp);
}

void flashWiFiFirmwareMapped() {
    uint32_t chunk_size = 1024;
    uint32_t byte_count = 0;
    const uint32_t offset = 15 * 1024 * 1024 + 1024 * 512;
    Serial.println("Flashing memory mapped WiFi firmware");
    printProgress(byte_count, file_size, 10, true);
    while (byte_count < file_size) {
        if (byte_count + chunk_size > file_size)
            chunk_size = file_size - byte_count;
        int ret = root->program(wifi_firmware_image_data, offset + byte_count, chunk_size);
        if (ret != 0) {
            Serial.println("Error writing memory mapped firmware");
            break;
        }
        byte_count += chunk_size;
        printProgress(byte_count, file_size, 10, false);
    }
}

```

```

}
#include "BlockDevice.h"
#include "MBRBlockDevice.h"
#include "LittleFileSystem.h"
#include "FATFileSystem.h"
#include "wiced_resource.h"
#include "certificates.h"
#ifndef CORE_CM7
#endif
using namespace mbed;

BlockDevice* root = BlockDevice::get_default_instance();
MBRBlockDevice wifi_data(root, 1);
MBRBlockDevice ota_data(root, 2);
MBRBlockDevice kvstore_data(root, 3);
MBRBlockDevice user_data(root, 4);
FATFileSystem wifi_data_fs("wlan");
FATFileSystem ota_data_fs("fs");
FileSystem * user_data_fs;

bool waitResponse() {
    bool confirmation = false;
    bool proceed = false;
    while (confirmation == false) {
        if (Serial.available()) {
            char choice = Serial.read();
            switch (choice) {
                case 'y':
                case 'Y':
                    confirmation = true;
                    proceed = true;
                    break;
                case 'n':
                case 'N':
                    confirmation = true;
                    proceed = false;
            }
        }
    }
}

```

```

        break;
    default:
        continue;
    }
}
}
return proceed;
}

void printProgress(uint32_t offset, uint32_t size, uint32_t threshold, bool reset) {
    static int percent_done = 0;
    if (reset == true) {
        percent_done = 0;
        Serial.println("Flashed " + String(percent_done) + "%");
    } else {
        uint32_t percent_done_new = offset * 100 / size;
        if (percent_done_new >= percent_done + threshold) {
            percent_done = percent_done_new;
            Serial.println("Flashed " + String(percent_done) + "%");
        }
    }
}

void setup() {
    Serial.begin(115200);
    while (!Serial);
    Serial.println("\nWARNING! Running the sketch all the content of the QSPI flash will
        be erased.");
    Serial.println("The following partitions will be created:");
    Serial.println("Partition 1: WiFi firmware and certificates 1MB");
    Serial.println("Partition 2: OTA 5MB");
    Serial.println("Partition 3: Provisioning KVStore 1MB");
    Serial.println("Partition 4: User data / OPTA PLC runtime 7MB"),
    Serial.println("Do you want to proceed? Y/[n]");
    if (true == waitResponse()) {

```

```

if (root->init() != BD_ERROR_OK) {
    Serial.println(F("Error: QSPI init failure."));
    return;
}
Serial.println("Do you want to perform a full erase of the QSPI flash before
proceeding? Y/[n]");
Serial.println("Note: Full flash erase can take up to one minute.");
bool fullErase = waitResponse();
if (fullErase == true) {
    Serial.println("Full erase started, please wait...");
    root->erase(0x0, root->size());
    Serial.println("Full erase completed.");
} else {
    // Erase only the first sector containing the MBR
    root->erase(0x0, root->get_erase_size());
}
MBRBlockDevice::partition(root, 1, 0x0B, 0, 1 * 1024 * 1024);
MBRBlockDevice::partition(root, 2, 0x0B, 1 * 1024 * 1024, 6 * 1024 * 1024);
MBRBlockDevice::partition(root, 3, 0x0B, 6 * 1024 * 1024, 7 * 1024 * 1024);
MBRBlockDevice::partition(root, 4, 0x0B, 7 * 1024 * 1024, 14 * 1024 * 1024);
// use space from 15.5MB to 16 MB for another fw, memory mapped
bool reformat = true;
if (!wifi_data_fs.mount(&wifi_data)) {
    Serial.println("\nPartition 1 already contains a filesystem, do you want to reformat it?
Y/[n]");
    wifi_data_fs.unmount();
    reformat = waitResponse();
}
if (reformat && wifi_data_fs.reformat(&wifi_data)) {
    Serial.println("Error formatting WiFi partition");
    return;
}
bool restore = true;
if (reformat || fullErase) {

```

```

Serial.println("\nDo you want to restore the WiFi firmware and certificates? Y/[n]");
restore = waitResponse();
}
if (reformat && restore) {
    flashWiFiFirmwareAndCertificates();
}
if (fullErase && restore) {
    flashWiFiFirmwareMapped();
}
reformat = true;
if (!ota_data_fs.mount(&ota_data)) {
Serial.println("\nPartition 2 already contains a filesystem, do you want to reformat it?
Y/[n]");
    ota_data_fs.unmount();
    reformat = waitResponse();
}
if (reformat && ota_data_fs.reformat(&ota_data)) {
    Serial.println("Error formatting OTA partition");
    return;
}
Serial.println("\nDo you want to use LittleFS to format user data partition? Y/[n]");
Serial.println("If No, FatFS will be used to format user partition.");
Serial.println("Note: LittleFS is not supported by the OPTA PLC runtime.");
if (true == waitResponse()) {
    Serial.println("Formatting user partition with LittleFS.");
    user_data_fs = new mbed::LittleFileSystem("user");
} else {
    Serial.println("Formatting user partition with FatFS.");
    user_data_fs = new mbed::FATFileSystem("user");
}
reformat = true;
if (!user_data_fs->mount(&user_data)) {
    Serial.println("\nPartition 4 already contains a filesystem, do you want to reformat it?
Y/[n]");
}

```

```

    user_data_fs->unmount();
    reformat = waitResponse();
}
if (reformat && user_data_fs->reformat(&user_data)) {
    Serial.println("Error formatting user partition");
    return;
}
Serial.println("\nQSPI Flash formatted!");
}
Serial.println("It's now safe to reboot or disconnect your board.");
}
const uint32_t file_size = 421098;
extern const unsigned char wifi_firmware_image_data[];
void flashWiFiFirmwareAndCertificates() {
    FILE* fp = fopen("/wlan/4343WA1.BIN", "wb");
    uint32_t chunk_size = 1024;
    uint32_t byte_count = 0;
    Serial.println("Flashing WiFi firmware");
    printProgress(byte_count, file_size, 10, true);
    while (byte_count < file_size) {
        if(byte_count + chunk_size > file_size)
            chunk_size = file_size - byte_count;
        int ret = fwrite(&wifi_firmware_image_data[byte_count], chunk_size, 1, fp);
        if (ret != 1) {
            Serial.println("Error writing firmware data");
            break;
        }
        byte_count += chunk_size;
        printProgress(byte_count, file_size, 10, false);
    }
    fclose(fp);
    fp = fopen("/wlan/cacert.pem", "wb");
    Serial.println("Flashing certificates");
    chunk_size = 128;

```

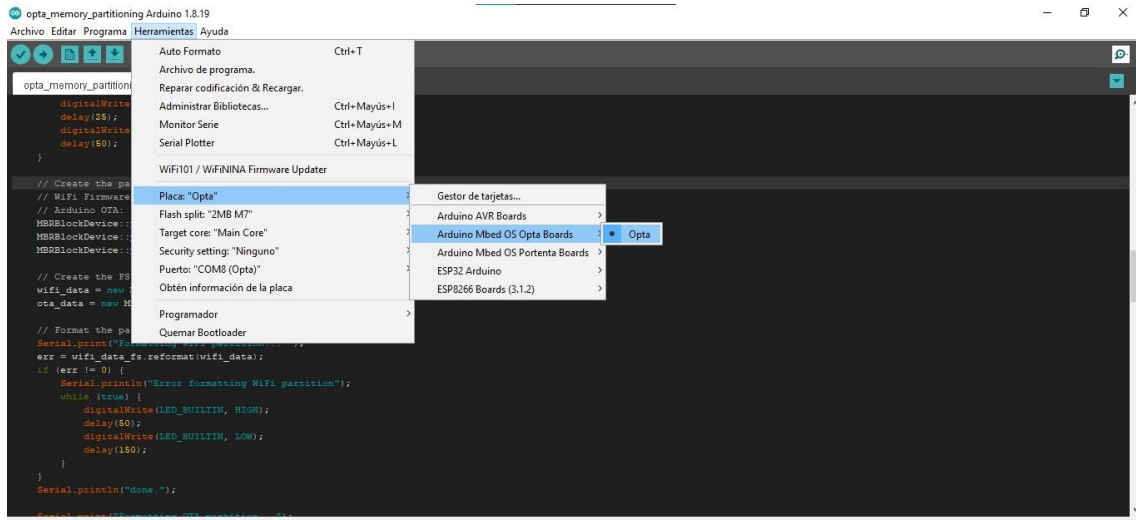
```

byte_count = 0;
printProgress(byte_count, cacert_pem_len, 10, true);
while (byte_count < cacert_pem_len) {
    if(byte_count + chunk_size > cacert_pem_len)
        chunk_size = cacert_pem_len - byte_count;
    int ret = fwrite(&cacert_pem[byte_count], chunk_size, 1 ,fp);
    if (ret != 1) {
        Serial.println("Error writing certificates");
        break.
    }
    byte_count += chunk_size;
    printProgress(byte_count, cacert_pem_len, 10, false);
}
fclose(fp);
}
void flashWiFiFirmwareMapped() {
    uint32_t chunk_size = 1024;
    uint32_t byte_count = 0;
    const uint32_t offset = 15 * 1024 * 1024 + 1024 * 512;
    Serial.println("Flashing memory mapped WiFi firmware");
    printProgress(byte_count, file_size, 10, true);
    while (byte_count < file_size) {
        if (byte_count + chunk_size > file_size)
            chunk_size = file_size - byte_count;
        int ret = root->program(wifi_firmware_image_data, offset + byte_count, chunk_size);
        if (ret != 0) {
            Serial.println("Error writing memory mapped firmware");
            break;
        }
        byte_count += chunk_size;
        printProgress(byte_count, file_size, 10, false);
    }
}
void loop() {

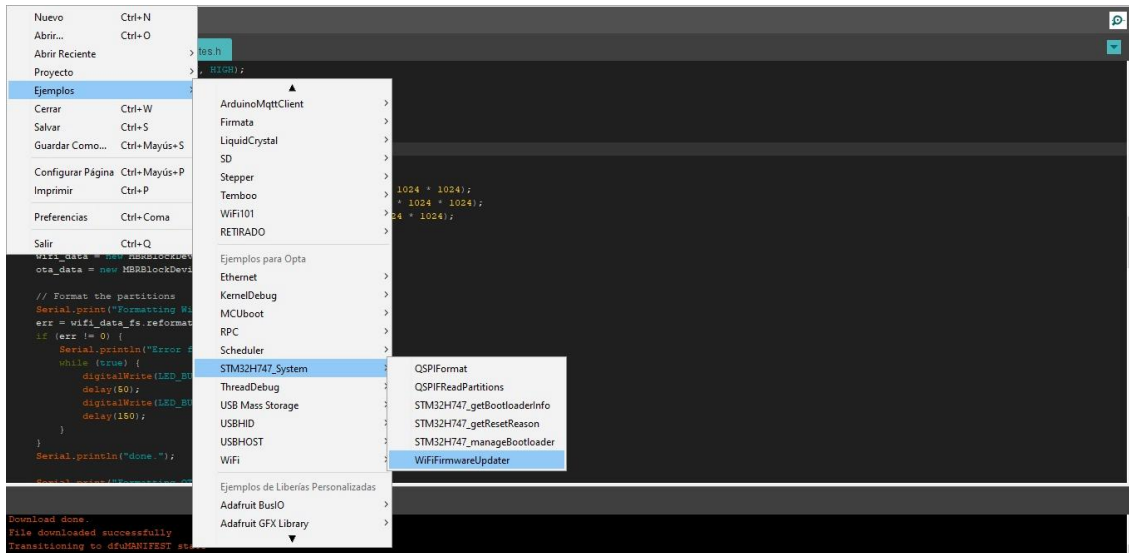
```

Anexo 2 directrices *WifiFirmwareUpdater*

Configuración del firmware



Seleccionar las tarjetas



Anexo 3 Programación del PLC IDE

Local variables

	Class	Pin	Name	Type	Array	Init value	Attribute
1	VAR_INPUT	0	INICIAR_BOMBA1	BOOL	No	..	
2	VAR_INPUT	1	SIST_INICIADO	BOOL	No	..	
3	RESULT		ARRANCAR_BOMBA_UINT	UINT	No	..	Function

```

0001 IF INICIAR_BOMBA1 = TRUE AND SIST_INICIADO = TRUE THEN
0002   ARRANCAR_BOMBA_F1 := 16#01 ;
0003 END_IF;
0004

```

Local variables

Name	Type	Address	Array	Init value	Attribute	Des
------	------	---------	-------	------------	-----------	-----

```

0001 out_enable := PID_ACTIVADO;
0002 out_kp := TO_REAL(HMI_KP)/100;
0003 out_ki := TO_REAL(HMI_KI)/100;
0004 out_kd := TO_REAL(HMI_KD)/100;
0005 out_windowsize := 5000;
0006 out_setpoint := TO_INT(SET_POINT);
0007
0008
0009
0010
0011
0012

```

Local variables

	Name	Type	Address	Array	Init value	Attribute
1	SENSOR_NIVEL_real	REAL	Auto	No	..	

```

0001 SENSOR_NIVEL_real := TO_REAL(SENSOR_NIVEL);
0002 NIVEL_AGUA_real := SENSOR_NIVEL_real * 2/65535;
0003 plc_hmi_NIVEL := TO_UINT(NIVEL_AGUA_real * 100.0);
0004 out_nivel := TO_INT(NIVEL_AGUA_real * 100.0);
0005

```

Local variables

	Class	Pin	Name	Type	Array	Init value	Attribute
1	VAR_INPUT	0	DETENER_BOMBA1	BOOL	No	..	Reci
2	VAR_INPUT	1	SIST_DETENIDO	BOOL	No	..	Si el
3	RESULT		DETENER_BOMBA_F1	UINT	No	..	Func

```

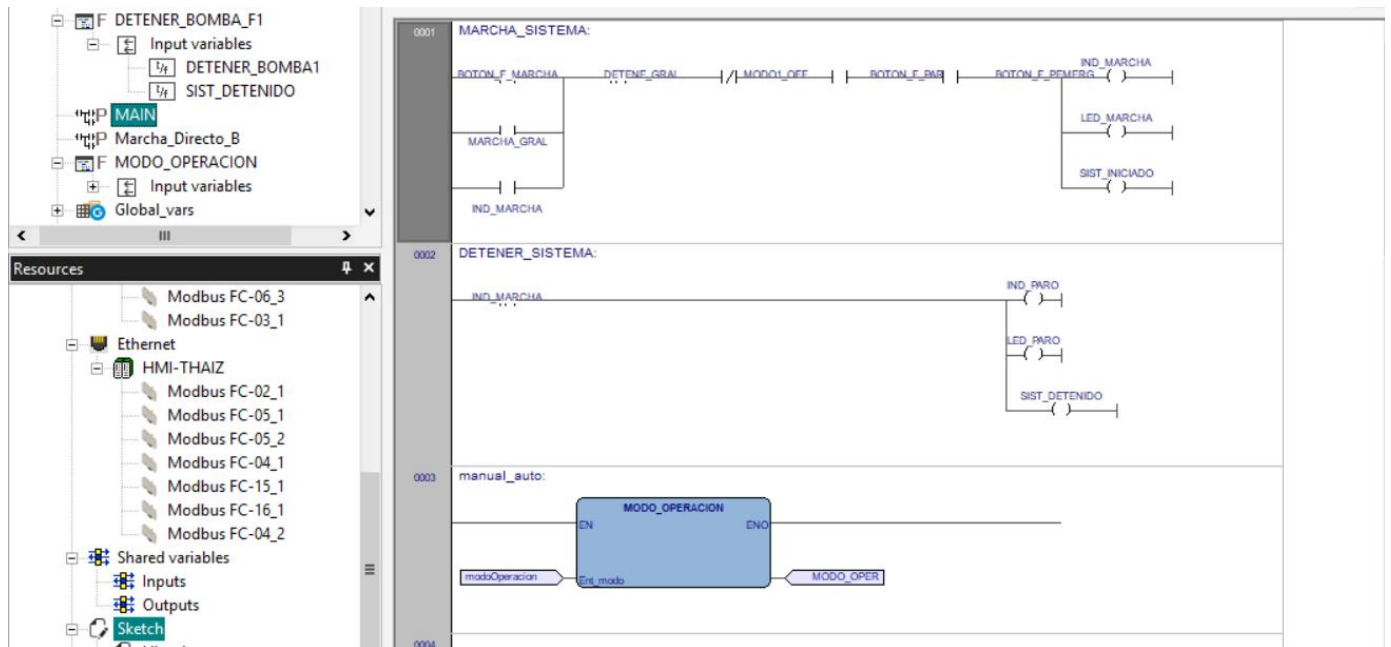
0001 IF DETENER_BOMBA1 = TRUE OR SIST_DETENIDO = TRUE THEN
0002   DETENER_BOMBA_F1 := 16#05;
0003 END_IF;
0004

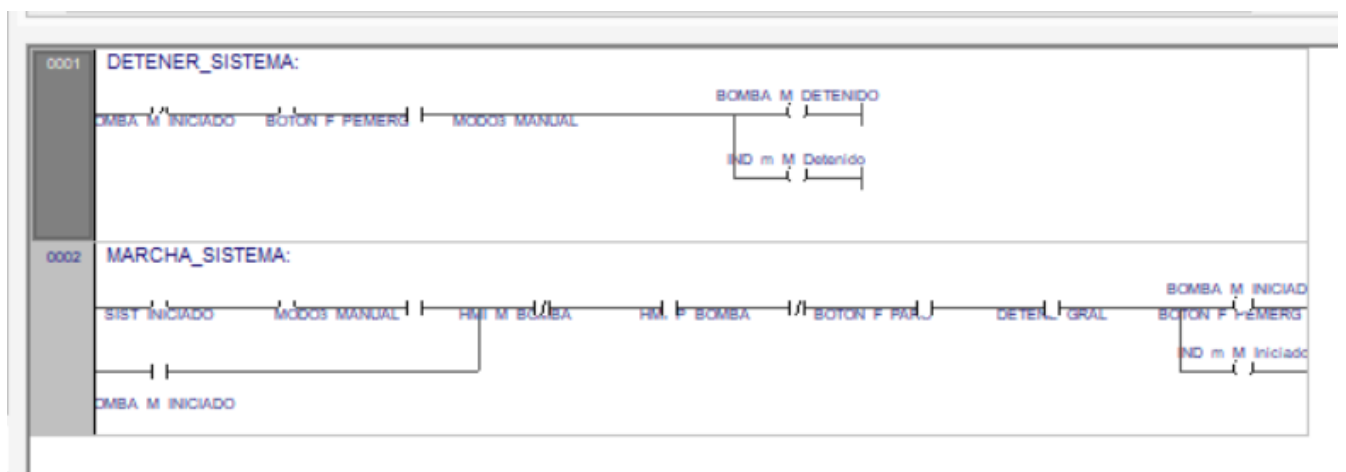
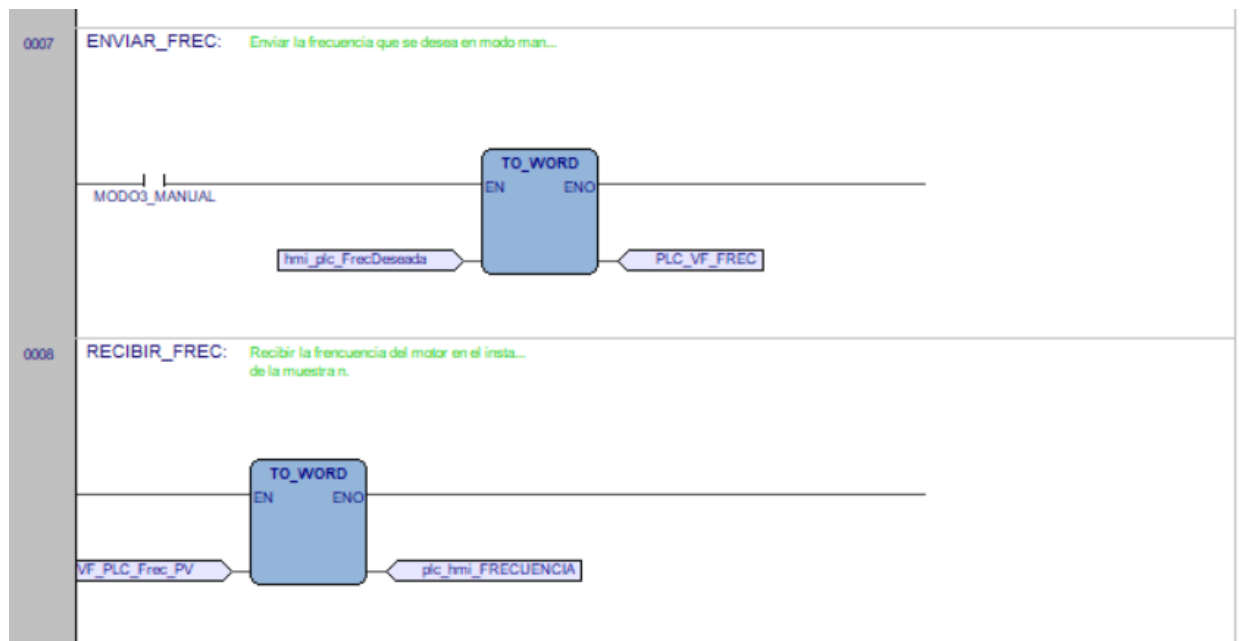
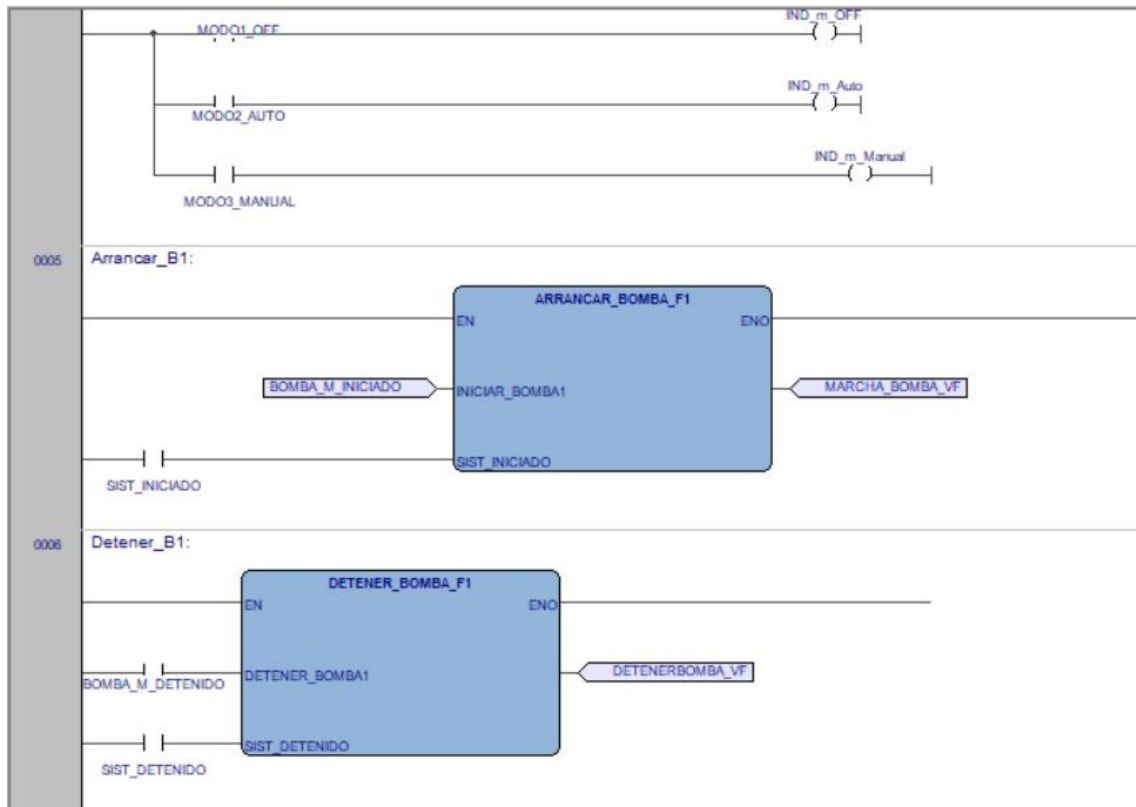
```

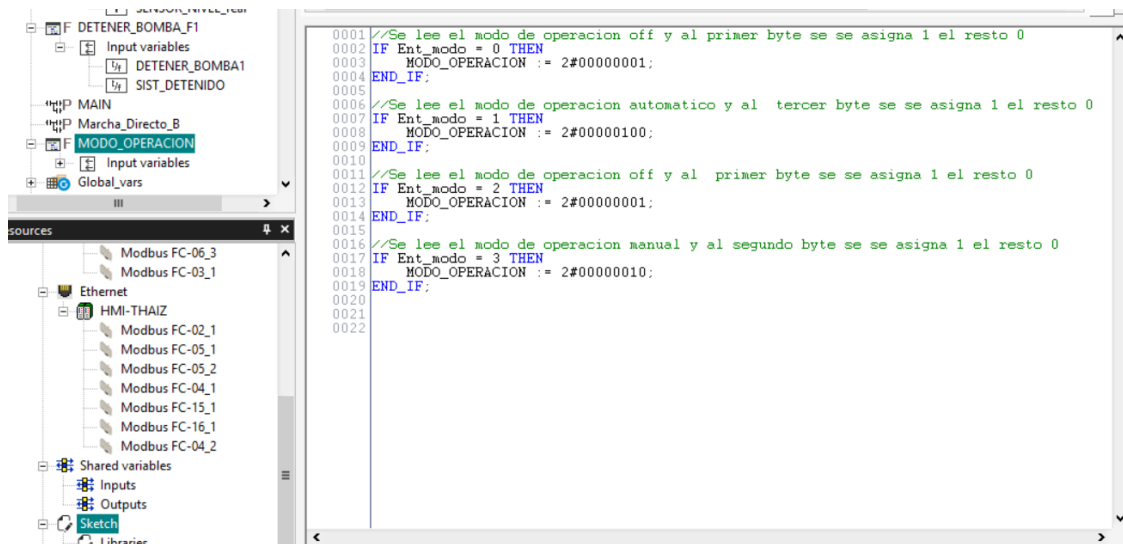
CONTROL PID DIAGRAMA EN ESCALERA



V







Anexo 4 código para integrar el controlador PID con la plataforma Arduino IoT Cloud.

```
#include <ArduinoIoTCloud.h>

#include <Arduino_ConnectionHandler.h>

#include "PID_v2.h"

#include <Arduino_NetworkConfigurator.h>

#include "configuratorAgents/agents/BLEAgent.h"

#include "configuratorAgents/agents/SerialAgent.h"

//parametros de la conexión a internet

const char SSID[] = "Maziken2.4G"; // Network SSID (name)

const char PASS[] = "8765432100";

//void onCloudButtonChange();

bool encendido;

WiFiConnectionHandler ArduinoIoTPreferredConnection(SSID, PASS);

double Kp = 2, Ki = 5, Kd = 1;

PID_v2 myPID(Kp, Ki, Kd, PID::Direct);

unsigned long windowStartTime;

void setup()

{

    Serial.begin(9600);

    delay(1500);

    // Configure static IP address

    IPAddress ip(192, 168, 2, 2);

    IPAddress dns(8, 8, 8, 8);
```

```

    IPAddress gateway(192, 168, 1, 0);
    IPAddress subnet(255, 255, 255, 0);
    //Si el cable no está conectado, esto bloqueará el inicio del PLC
    //con un tiempo de espera de aproximadamente 60 segundos.
    Ethernet.begin(ip, dns, gateway, subnet);
delay(5000);
    // Defined in thingProperties.h
    initProperties();
    // Connect to Arduino Cloud
    ArduinoCloud.begin(ArduinoIoTPreferredConnection);
    setDebugMessageLevel(2);
    ArduinoCloud.printDebugInfo();
    delay(5000);
    windowStartTime = millis();
    myPID.SetOutputLimits(0, 60);
    myPID.Start(PLCOut.out_nivel, 0, 100);
}
void loop()
{
    ArduinoCloud.update();
    const bool enabled = PLCOut.out_enable;
    const double input = PLCOut.out_nivel;
    const double Kp = PLCOut.out_kp;
    const double Ki = PLCOut.out_ki;
    const double Kd = PLCOut.out_kd;
    const int WindowSize = PLCOut.out_windowsize;
    const double setpoint = PLCOut.out_setpoint;
    // const double output = 0;
    double output = 0;
    myPID.SetOutputLimits(0, WindowSize);
    myPID.Setpoint(setpoint);
    myPID.SetTunings(Kp, Ki, Kd);
    if (enabled)
    {

```

```

        output = myPID.Run(input);
    }
else
{
    //myPID.Reset();          // Limpia integral y derivativo acumulados
    output = 0;
}
PLCIn.in_output = output*100;
while (millis() - windowStartTime > WindowSize) {
    windowStartTime += WindowSize;
}
unsigned long window = millis() - windowStartTime;
PLCIn.in_window = window;
if (output > 60)
    PLCIn.in_output = 6000;
else
    PLCIn.in_output = output*100;
encendido = enabled;
}

void initProperties(){
    NetworkConfigurator.addAgent(BLEAgent);
    NetworkConfigurator.addAgent(SerialAgent);
    NetworkConfigurator.setStorage(kvStore);

    // For changing the default reset pin uncomment and set your preferred pin. Use
    DISABLE_PIN for disabling the reset procedure.

    //NetworkConfigurator.setReconfigurePin(your_pin);
    ArduinoCloud.setConfigurator(NetworkConfigurator);
    ArduinoCloud.addProperty(encendido, READ, ON_CHANGE, NULL);
    //ArduinoCloud.addProperty(counter, READ, ON_CHANGE, NULL);
    //ArduinoCloud.addProperty(analog01, READ, ON_CHANGE, NULL);
}

//Final
//Final
//Final

```

Anexo 5 configuracion modbus rs485 en el plc ide

Modbus Configuration

Mode

Not used
 Modbus RTU Master
 Modbus RTU Slave

Baud rate

600 b/s
 1200 b/s
 2400 b/s
 4800 b/s
 9600 b/s
 19200 b/s
 38400 b/s
 57600 b/s
 76800 b/s
 115200 b/s



Serial Mode

N,8,1 (No parity, 8 data bits, 1 stop bit) ▼

Modbus FC 06(0x06) - Write Single Register

General **Single Reg.**



Settings

Address: (0 .. 65535)
Polling time: ms (0 = write on variation)
Time out: ms
Oneshot variable  

Modbus FC 06(0x06) - Write Single Register

General **Single Reg.**



Settings

Address: (0 .. 65535)
Polling time: ms (0 = write on variation)
Time out: ms
Oneshot variable  

Modbus FC 06(0x06) - Write Single Register

General **Single Reg.**

Settings

Address: (0 .. 65535)
Polling time: ms (0 = write on variation)
Time out: ms
Oneshot variable  

CONFIGURACION MODBUS TCP/IP EN EL PLC IDE

Generic Modbus node

General **Parametrization**

Settings

Name:

IP address:

Modbus address: (255 , or 1..247 for bridge)

Minimum polling time: ms

Address type: Modbus Jbus

Modbus FC 02(0x02) - Read Discrete Inputs

General **Inputs**

Settings

Start address: (1 .. 65536)

Polling time: ms (0 = Continuous Read)

Time out: ms

Modbus FC 05(0x05) - Write Single Coil

General **Coil**

Settings

Address: (1 .. 65536)

Polling time: ms (0 = write on variation)

Time out: ms

Modbus FC 05(0x05) - Write Single Coil

General **Coil**

Settings

Address: (1 .. 65536)

Polling time: ms (0 = write on variation)

Time out: ms

Modbus FC 03(0x03) - Read Holding Register



General **Holding Reg.**

Settings

Start address: (0 .. 65535)

Polling time: ms (0 = Continuous Read)

Time out: ms

Oneshot variable  

Modbus FC 04(0x04) - Read Input Registers

General

Input Reg.

Settings

Start address: (1 .. 65536)
Polling time: ms (0 = Continuous Read)
Time out: ms

Modbus FC 15(0x0F) - Write Multiple Coils

General

Coils

Settings

Start address: (1 .. 65536)
Polling time: ms (0 = write on variation)
Time out: ms

Modbus FC 16(0x10) - Write Multiple Register

General

Multiple Reg.

Settings

Start address: (1 .. 65536)
Polling time: ms (0 = write on variation)
Time out: ms

Modbus FC 04(0x04) - Read Input Registers

General

Input Reg.

Settings

Start address: (1 .. 65536)
Polling time: ms (0 = Continuous Read)
Time out: ms

BIBLIOGRAFÍA

- [1] D. G. J. C. J. G. H. V. y. M. T. L. L. F. Amaya Jimenez, *Método Fenomenológico Hermenéutico.*, 2021.
- [2] C. A. Ripoll, *Hermenéutica, pedagogía y praxeología.* Universidad Libre, Barranquilla, 2017.
- [3] M. Bruno, *Metodología proyectual*, 2013.
- [4] R. R. P. J. A. & V. A. C. Sanchis Llopis, *Automatización industrial*, 2010.
- [5] J. R. Vignoni, *Control de Procesos. Instrumentación y Comunicaciones Industriales.*, 2002.
- [6] J. V. Chavarría, *Software libre, alternativa tecnológica para la educación.* "Actualidades Investigativas en Educación 5.2, 2005.
- [7] MasterPLC, *Principales marcas de PLC*", 2024.
- [8] A. S. Tanenbaum, "*Computer Networks*", 5th ed., Pearson, 2011.
- [9] W. Stallings, *Data and Computer Communications*", 10th ed., 2013.
- [10] J. F. K. a. K. W. Ross, "*Computer Networking: A Top-Down Approach*", 6th ed..
- [11] A. Forouzan, "*Data Communications and Networking*", 2012.
- [12] R. T. L. a. T. S. Hwang, *Wireless technologies for industrial applications*, 2008.
- [13] A. C. D. D. a. R. V. C. Buratti, *An overview on wireless sensor networks technology and evolution*, 2009.
- [14] M. Rouse, *Gateway (IoT Gateway)*, 2016.
- [15] W. H. a. S. L. L. Da Xu, *Internet of Things in Industries: A Survey*, 2014.
- [16] U. P. [Online], *Tecnología inalámbrica PROFINET*.
- [17] S. Kim, "*Smart Sensors for Industrial Applications*.
- [18] T. M. a. K. R. C. I. F. Akyildiz, *A survey on wireless multimedia sensor networks*, 2007.
- [19] L. G. a. M. V. C. P. A. Chou, *Performance advantages of endpoint-based multimedia data collection and delivery.*, 2001.
- [20] M. A. e. al, *A view of cloud computing,*" *Communications of the ACM*, vol. 53,, 2010.
- [21] R. M. J. Z. a. S. A. F. Bonomi, *Fog Computing and Its Role in the Internet of Things*, 2012.
- [22] P. P. S. V. a. V. K. M. Crosby, "*Blockchain Technology: Beyond Bitcoin*.
- [23] J. Smith, *Introduction to Control Systems, 2nd ed*, New York, NY, USA: Wiley,, 2010.

- [24] S. R. C. Dorf and R. H. Bishop, *Modern Control Systems, 13th ed. Upper*, addle River, NJ, USA, 2017.
- [25] K. Ogata, *Modern Control Engineering*, Saddle River, NJ, USA: , 2009.
- [26] E. F. Camacho and C. Bordons, *Model Predictive Control*, London, UK:, 2007.
- [27] C. A. Zúñiga Reyes, *Automatización de un Sistema de Esclusas Empleando Tecnología del*, 2022.
- [28] [En línea].
- [29] ". A. Flow sensor 1/2 inch to 2 inch. [En línea]. Available: https://www.amazon.com/dp/B0D1QC838C?ref_=pe_386300_442618370_TE_sc_as_ri_0. [Último acceso: 18 sep 2024].
- [30] ". A. Flow sensor. [En línea]. Available: https://www.amazon.com/dp/B0CQFY9GPL?ref_=pe_386300_442618370_TE_sc_as_ri_0&th=1. [Último acceso: 19 sep 2024].
- [31] Trimantec, «Weintek cMTX: Advanced 7" HMI - cMT3072XP,» [En línea]. Available: <https://trimantec.com/products/weintek-cmtx-advanced-7-hmi-cmt3072xp..> [Último acceso: 25 11 2024].
- [32] INVT, «GD10 Series Mini variador de frecuencia económico,» [En línea]. Available: <https://www.invt.com/es/products/gd10-series-vfd-drive-13..> [Último acceso: 25 11 2024].
- [33] L. L. T. e. Línea., «Bomba Tipo Jet FW de 1/2HP 110-220V,» [En línea]. Available: Bomba Tipo Jet FW de 1/2HP 110-220V," <https://tienda.lallave.com.ec/tienda/bomba-de-agua-y-accesorios/bombas-tipo-jet/bomba-tipo-jet-fw-de-1-2hp-110-220v/..> [Último acceso: 25 SEP 2024].
- [34] CAMEI, «Breaker riel din 3 polos 6 amperios curva C Acti9 iC60N,» CAMEI.,» [En línea]. Available: <https://www.camei.com.ec/shop/product/se-a9f74306-breaker-riel-din-3-polos-6-amperios-curva-c-acti9-ic60n-1889?page=7>. [Último acceso: 08 10 2024].
- [35] P. ., F. G. Shinskey, *Control Systems: Application, Design, and Tuning*, New York, NY, USA: McGraw-Hill,; 4th ed., 1996..
- [36] K. J. Å. a. R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers.*, Princeton, NJ, USA: Princeton University Press, , 2010..
- [37] L. Q. J. M. V. G. J. & L. A. Viáfara, *Industria 4.0 Gestión del conocimiento Editorial Universidad Icesi*, 2021.
- [38] C. Lozada, *Desarrollo de un sistema basado en Internet Industrial de las Cosas para el monitoreo y control de un banco de pruebas de intercambiador de calor*, 2022.
- [39] J. Uslenghi,). *Diseño de aplicación IoT para la supervisión energética de una instalación frigorífica*, 2019.
- [40] U. Shrinivasa, "The evolution of diesel engines," , 2012.
- [41] F. Rozo-García, *Revisión de las tecnologías presentes en la industria 4.0. Revista UIS Ingenierías*, 2020.

- [42] M. directory, *Industrial Internet consortium*, 2019.
- [43] i. i. consortium, *industrial internet reference architecture*, 2015.
- [44] R. P. & F. J. J. Domiguez M., . *Laboratorio Remoto para la Enseñanza de la Automática en en la universidad de Leon*, España, 2005.
- [45] J. Cegarra Sánchez, *Metodología de la investigación científica y tecnológica.*, 2004.
- [46] J. I. R. Molano, *Metamodelo para la integración del internet de las cosas y redes sociales*, 2017.
- [47] J. M. Gallegos Ramírez y E. H. Delgado Guerrero, "Diseño e implementación de una red industrial Modbus para el control de actuadores trifásicos en el laboratorio de fabricación flexible, 2015.
- [48] P. M. P. Aguilera Martínez, *Programación de plc s (Doctoral dissertation, Universidad Autónoma de Nuevo León)*, 2002.
- [49] B. A. Forouzan and F. Mosharraf, "Computer Networks: A System Approach".
- [50] M. S. a. M. H. J. Soldatos, *Convergence of Utility Computing with the Internet-of-Things*, 2010.
- [51] J. D. P. a. A. E.-N. G. F. Franklin, *Feedback Control of Dynamic Systems*, 2015, Saddle River, NJ, USA.
- [52] K. J. Astrom and B. Wittenmark, *Adaptive Control.*, MA, USA, 1994.
- [53] P. Aguilera Martínez, *Programación de plc s (Doctoral dissertation, Universidad Autónoma de Nuevo León).*, 2002.
- [54] J. Inman, *What Are Proprietary Ring Topologies in Automation Networks,* DigiKey, 2022.
- [55] I. O. P. (HMI)., *Quin Systems Ltd.* [Online]. Available: <https://www.quin.co.uk/products/hmi-operator-panels/>, 2024.
- [56] M. E. E. Pereira, *Diseño de interfaces modulares para facilitar proyectos de IOT*, 2022.
- [57] F. (n.d.)., *istema didáctico de control de procesos de presión, caudal y nivel, con banco LabVolt Series 6090-A.* [Online]. Available: https://www.festo.com/co/es/p/sistema-didactico-de-control-de-procesos-de-presion-caudal-y-nivel-con-banco-id_PROD_DID_588652/?page=0.
- [58] F. (n.d.)., *Demostrador de sistema de control distribuido PlantPax™, presión/caudal/nivel/temperatura LabVolt Series 3531-V.* [Online]. Available: <https://www.festo.com/co/es/p/demostrador-de-sistema-de-control-distribuido-plantpax-tm-presion-caudal-nivel-temperatura->.
- [59] Coledidacticum., *Lab-Volt: Sistema mecatrónico avanzado (AMS).* [Online]. Available: <https://www.coledidacticum.com/wp/lab-volt-sistema-mecatronico-avanzado-ams/>. [Accessed: 07-Jul-2024]., (2018, Sept. 30).
- [60] «EVL, "Adaptador AC/DC Intel. Fijo 12V-2A Plug 2.5 - AD-122S," » [En línea]. Available: Available: <https://www.velasco.com.ec/velasco/producto.php?id=4440>. . [Último acceso: 18 Sep 2024].
- [61] ". W. A. S. Arduino. [En línea]. Available: <https://store.arduino.cc/products/opta-wifi>. [Último acceso: 18 Sep 2024].

- [62] ". H. K. B. m. P. S. I. M. Siemens. [En línea]. Available: <https://mall.industry.siemens.com/mall/es/ms/Catalog/Product/6AV6647-0AB11-3AX0>. [Último acceso: 24 SEP 2024].
- [63] CAMEI, «"Guardamotor Termomagnético GV2ME 3 polos 1.6 A - 2.5 A AC3," CAMEI.,» [En línea]. Available: <https://www.camei.com.ec/shop/product/se-gv2me07-guardamotor-termomagnetico-gv2me-3-polos-1-6-a-2-5-a-ac3-2168?category=225..> [Último acceso: 25 sep 2024].
- [64] «Contactor TeSys D 3P3 NA AC3, 440 V, 9 A, 24 V CC bobina,» [En línea]. Available: "Contactor TeSys D 3P3 NA AC3, 440 V, 9 A, 24 V CC bobina," [En línea]. <https://www.se.com/es/es/product/LC1D09BD/contactor-tesys-d-3p3-na-ac3-440-v-9-a-24-v-cc-bobina/>. [Último acceso: 08 10 2024].
- [65] KAS, «Válvula de bola de latón de paso total para agua PN 40," KAS.,» [En línea]. Available: <https://kas.com.tr/es/product/valvula-de-bola-de-laton-de-paso-total-para-agua-pn-40/>. [Último acceso: 08 10 2024].
- [66] 1. Marcas, «SolidWorks Logo," 1000 Marcas.,» [En línea]. Available: <https://1000marcas.net/solidworks-logo/>. [Último acceso: 8 10 2024].
- [67] Arduino, «"Software," Arduino,» [En línea]. Available: <https://www.arduino.cc/en/software..> [Último acceso: 10 08 2024].
- [68] How2Electronics, «Getting Started with Arduino IoT Cloud with ESP8266," How2Electronics.,» [En línea]. Available: [https://how2electronics.com/getting-started-with-arduino-iot-cloud-with-esp8266/..](https://how2electronics.com/getting-started-with-arduino-iot-cloud-with-esp8266/) [Último acceso: 08 10 2024].
- [69] Promart, «Tubo PVC 1 1/2"x6m 240psi Plastigama," Promart,» [En línea]. Available: <https://www.promart.ec/tubo-pvc-1-1-2x6m-240psi-plastigama/p..>
- [70] plc247.com, «EasyBuilder Pro V6.05.02 Weintek HMI Software,» [En línea]. Available: [https://plc247.com/download-easybuilder-pro-v6-05-02-weintek-hmi-software/..](https://plc247.com/download-easybuilder-pro-v6-05-02-weintek-hmi-software/) [Último acceso: 25 11 2025].
- [71] A. T. y. D. Wetherall, *Redes de Computadoras,,* 2012.
- [72] RoboticSec, «Luces Piloto 22mm 24V,» 26 10 2025. [En línea]. Available: <https://roboticsec.com/producto/luces-piloto-22mm-24v/>.