



UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA,
ELECTRÓNICA Y SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



**DISEÑO DE UN MONITOREO REMOTO DE LA SALA DE
CONTROL DE PRESIÓN CONSTANTE UTILIZANDO LA
TECNOLOGÍA DE LA INDUSTRIA 4.0.**

TESIS

PRESENTADA POR:

JAMIL HENRY SACACA RAMOS

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRÓNICO

PUNO – PERÚ

2024



NOMBRE DEL TRABAJO

DISEÑO DE UN MONITOREO REMOTO DE
LA SALA DE CONTROL DE PRESIÓN CO
NSTANTE UTILIZANDO LA TECNOLOGÍA
DE LA INDUSTRIA 4.0

AUTOR

JAMIL HENRY SACACA RAMOS

RECUESTO DE PALABRAS

19946 Words

RECUESTO DE CARACTERES

117870 Characters

RECUESTO DE PÁGINAS

147 Pages

TAMAÑO DEL ARCHIVO

6.9MB

FECHA DE ENTREGA

Jan 25, 2024 4:38 PM GMT-5

FECHA DEL INFORME

Jan 25, 2024 4:40 PM GMT-5

● **11% de similitud general**

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base

- 9% Base de datos de Internet
- Base de datos de Crossref
- 7% Base de datos de trabajos entregados
- 2% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

● **Excluir del Reporte de Similitud**

- Material bibliográfico
- Material citado
- Material citado
- Coincidencia baja (menos de 9 palabras)


Marco Antonio Quispe Barra
CIP 57501.
Director de Tesis.


Dr. Ing. Henry Ivan Condori Alejo
CIP: 56684
Dr. Ing. Juan Flores
V.B.

Resumen



DEDICATORIA

A Dios y a mis padres, Joaquín y Lucía, cuyo amor incondicional y sacrificio han sido la motivación a lo largo de formación académico. Sus valores y consejos han sido importante, y este logro es también de ellos.

A mis hermanas Ana y Yeni por estar siempre acompañándome en todo momento de mi vida. A mi hermano Wilver por su apoyo permanente en mi vida universitaria.

A mis amigos, de una u otra manera, compartieron este viaje conmigo, brindándome apoyo emocional. Este logro importante no hubiera sido posible sin el apoyo positivo de cada uno de ustedes.

Jamil Henry Sacaca Ramos



AGRADECIMIENTO

Quiero expresar mi más sincero agradecimiento a la Dr.Sc. Liceli Peñarrieta Bedoya, quien desempeñó un papel fundamental como mi maestra y mentora a lo largo de mi carrera académica. Su sabiduría y su guía experta fueron elementos significativos en el éxito de este proyecto.

La realización de esta tesis de pregrado no hubiera sido posible sin la ayuda de mi asesor, agradezco al Dr. Marco Quispe Barra por proporcionar los recursos y entorno propicio para realizar el proyecto. La infraestructura y el acceso de laboratorios fueron indispensable para llevar la investigación. También, expreso mi gratitud al Mg. Gavino Rey Vidangos Ponce por su guía experta. Su orientación y paciencia ha sido crucial para dar forma a la investigación.

Me gustaria extender mi agradecimiento a la empresa E2I y a su CEO Mg. Eduardo Zárate Cáceres por formarme y guiarme a lo largo de mi formación profesional. Esta tesis no hubiera sido posible sin el apoyo y la colaboración de todo el equipo E2I, que confió en mí para lograr este objetivo. Estoy agradecido por el compromiso continuo de la empresa E2I con la excelencia académica y la promoción de la investigación.

Quiero expresar mi más sincero agradecimiento a las personas que han contribuido de manera significativa a la realización de este trabajo de tesis.

Jamil Henry Sacaca Ramos



ÍNDICE GENERAL

	Pág.
DEDICATORIA	
AGRADECIMIENTO	
ÍNDICE GENERAL	
ÍNDICE DE TABLAS	
ÍNDICE DE FIGURAS	
ÍNDICE DE ANEXOS	
ÍNDICE DE ACRÓNIMOS	
RESUMEN	17
ABSTRACT.....	18
CAPÍTULO I	
INTRODUCCIÓN	
1.1 PROBLEMA.....	20
1.2 OBJETIVO	20
1.3 HIPÓTESIS	21
1.3.1 Hipótesis específicas.....	21
CAPÍTULO II	
REVISIÓN DE LITERATURA	
2.1 ANTECEDENTES DE LA INVESTIGACIÓN	22
2.1.1 A nivel internacional.....	22
2.1.2 A nivel nacional.....	25
2.2 REFERENCIAS TEÓRICAS	26



2.2.1	Industria 4.0	26
2.2.1.1	Internet de las Cosas	26
2.2.2	Automatización industrial.....	27
2.2.3	Sistema de control presión constante.....	28
2.2.3.1	Controlador lógico programables (PLC).....	28
2.2.3.2	Actuadores y transmisores Industriales.....	29
2.2.3.3	Software de diseño (CAD).....	29
2.2.3.4	Software programación industrial.....	30
2.2.4	Protocolo Modbus.....	30
2.2.4.1	Trama Modbus	32
2.2.4.2	Funciones Modbus	33
2.2.4.3	Modbus RTU.....	34
2.2.4.4	Modbus ASCII	36
2.2.4.5	Modbus TCP/IP.....	36
2.2.4.6	Modbus PLUS.....	38
2.2.5	Protocolo MQTT.....	39
2.2.6	Sistemas embebidos.....	40
2.2.6.1	Sistema de control: Raspberry PI.....	41
2.2.6.2	Sistema de comunicación: Enrutador.....	42
2.2.7	Frameworks	42
2.2.7.1	JavaScript.....	43
2.2.7.2	Python	43
2.2.8	Computación en la nube	44
2.2.8.1	Amazon Web Services (AWS)	46
2.2.8.2	Microsoft Azure	47
2.2.8.3	Google Cloud Platform (GCP).....	48



CAPÍTULO III

MATERIALES Y MÉTODOS

3.1	METODOLOGÍA	49
3.1.1	Diseño de investigación	49
3.1.1.1	Variables	49
3.1.2	Población	50
3.1.3	Muestra	50
3.1.4	Técnica de recolección de datos	51
3.1.5	Instrumento de recolección de datos.....	51
3.2	MODELO Y ANÁLISIS.....	52
3.2.1	Elicitación de requisitos.....	53
3.2.1.1	Abstracción de caja negra	54
3.2.1.2	Agrupación de funciones	55
3.2.1.3	Estructura de funciones.....	57
3.2.2	Análisis morfológico.....	59
3.2.2.1	Matriz morfológica del sistema integrado	60
3.2.2.2	Matriz morfológica del sistema eléctrico.....	63
3.2.3	Evaluación técnica – económica.....	65
3.2.3.1	Criterios técnicos.....	66
3.2.3.2	Criterios económicos.....	67
3.2.4	Concepto de solución optimo	68
3.2.5	Arquitectura y diseño del sistema.....	69
3.2.5.1	Arquitectura del proceso de control.....	69
3.2.5.2	Arquitectura de software.....	72
3.2.5.3	Arquitectura de AWS IoT Core	77
3.3	INGENIERÍA DE REQUISITOS.....	78



3.3.1	Ubicación del proyecto	79
3.3.2	Planta de control de presión.....	79
3.3.3	Enrutador Industrial TRB245	80
3.3.4	Raspberry PI 3 Plus.....	81
3.3.5	Amazon Web Services.....	82
3.3.6	DAIREL.....	83
3.4	INTEGRACIÓN DEL SISTEMA	84
3.4.1	Almacenamiento de datos	84
3.4.2	Control remoto del proceso.....	89
3.4.3	Configuración TRB245.....	93
3.4.4	Programación en Raspberry PI	96
3.4.5	Estructura de base de datos	99
3.4.6	Consulta de datos	102
3.4.7	Monitoreo en DAIREL	104

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1	OPTIMIZACIÓN DEL PROCESO DE MONITOREO.....	106
4.2	ANÁLISIS DE DATOS	108
4.2.1	Análisis de datos presión constante	108
4.2.2	Análisis de datos variador.....	109
4.3	ANÁLISIS DE VARIABLES.....	111
4.3.1	Correlación de variables	112
4.3.1.1	Interpretación de procesos	112
4.3.2	Correlación de Pearson	118
4.3.3	Visualización de variables	119
4.4	ARQUITECTURA DEL SISTEMA.....	120



4.4.1	Arquitectura de la primera solución IIoT	121
4.4.2	Arquitectura de la segunda solución IIoT.....	121
4.5	DISCUSIONES.....	123
V.	CONCLUSIONES.....	125
VI.	RECOMENDACIONES.....	127
VII.	REFERENCIAS BIBLIOGRÁFICAS	128
	ANEXOS.....	132

Área: Automatización e Instrumentación.

Tema: Industria 4.0.

FECHA DE SUSTENTACIÓN: 31 de enero de 2024



ÍNDICE DE TABLAS

	Pág.
Tabla 1 Código de funciones Modbus	33
Tabla 2 MBAP (Modbus Application Protocol Header)	37
Tabla 3 Lista de Exigencias	54
Tabla 4 Aspectos Posibles de Evaluación.....	58
Tabla 5 Criterios de calificación	59
Tabla 6 Matriz morfológica del sistema integrado	61
Tabla 7 Evaluación de la matriz morfológica del sistema integrado	62
Tabla 8 Matriz morfológica del sistema eléctrico.....	63
Tabla 9 Evaluación de la matriz morfológica del sistema eléctrico	64
Tabla 10 Ponderado de criterios	65
Tabla 11 Evaluación técnica	66
Tabla 12 Evaluación económica	67
Tabla 13 Criterios de solución óptimo.....	69
Tabla 14 Configuración del diagrama de flujo	75
Tabla 15 Comparación de correlaciones de Pearson	118
Tabla 16 Elementos de control de presión constante.....	132
Tabla 17 Hoja de especificación del TRB245	143
Tabla 18 Hoja de especificación del Raspberry PI	144
Tabla 19 Hoja de especificación de MateBook D 15.....	145



ÍNDICE DE FIGURAS

	Pág.
Figura 1 Esquema general de internet de las cosas.....	27
Figura 2 Modbus maestro - esclavo.....	31
Figura 3 Arquitectura general del protocolo Modbus.....	32
Figura 4 Trama de Modbus Serial	33
Figura 5 Conector DB9.....	35
Figura 6 Arquitectura de conexión del protocolo Modbus RTU	35
Figura 7 Arquitectura de conexión de Modbus ASCII	36
Figura 8 Trama de Modbus TCP/IP.....	37
Figura 9 Arquitectura de conexión del protocolo Modbus TCP/IP	38
Figura 10 Protocolo MQTT	39
Figura 11 Servicios en la nube.....	45
Figura 12 Modelo VDI 2206	53
Figura 13 Caja negra del sistema.....	55
Figura 14 Agrupación de funciones.....	57
Figura 15 Estructura integral de las funciones	58
Figura 16 Gráfico de comparación valor Técnico - Económico.....	68
Figura 17 Diagrama de lazo cerrado de presión constante	70
Figura 18 Esquema general de lazo cerrado	71
Figura 19 Distribución de la sala de control de presión constante	72
Figura 20 Diagrama de secuencia.....	74



Figura 21	Diagrama de flujo.....	76
Figura 22	Arquitectura del código.....	77
Figura 23	Arquitectura AWS IoT Core	78
Figura 24	Universidad Nacional del Altiplano.....	79
Figura 25	Laboratorio de la planta de presión constante.....	80
Figura 26	Enrutador TRB245 - Teltonika	81
Figura 27	Raspberry PI 3 Plus.....	82
Figura 28	Arquitectura de solución IoT -AWS	82
Figura 29	Aplicación de DAIREL.....	83
Figura 30	Conexión TIA Portal - PLC SIEMENS	85
Figura 31	Presión escalado.....	86
Figura 32	Almacenamiento de datos en Hold Register	86
Figura 33	Modbus Server	87
Figura 34	Conexión Modbus	88
Figura 35	Datos en Hold Register	88
Figura 36	Asignación de variables	89
Figura 37	Variador y Presión - Ubidots.....	90
Figura 38	Estado del sistema - Ubidots	90
Figura 39	Inicio y parada - Ubidots.....	91
Figura 40	Node-red.....	92
Figura 41	Control remoto en Ubidots.....	92
Figura 42	Topología de envío de datos - TRB245	93
Figura 43	Solicitud de datos Modbus TCP/IP	94



Figura 44	Formato JSON.....	95
Figura 45	Certificados AWS	95
Figura 46	Configuración de acceso a AWS.....	96
Figura 47	Topología envío de datos a la nube – segunda solución	97
Figura 48	Librerías de Python	97
Figura 49	Configuración Modbus Server	98
Figura 50	Configuración AWS IoT Core	98
Figura 51	Payload.....	99
Figura 52	Tópico AWS IoT Core	99
Figura 53	Reglas IoT	100
Figura 54	AWS Timestream.....	101
Figura 55	Acción Query	101
Figura 56	Accesos externos - IAM.....	102
Figura 57	Conexión AWS y Grafana.....	103
Figura 58	Consulta de Datos - Grafana	103
Figura 59	Dashboards del monitoreo remoto	104
Figura 60	Registro de usuario - DAIREL.....	104
Figura 61	Consulta de datos en DAIREL.....	105
Figura 62	Validación de datos local - remoto.....	106
Figura 63	Validación de proceso de monitoreo	107
Figura 64	Datos de la presión constante	108
Figura 65	Datos del variador	110
Figura 66	Interpretación de un sub evento	111



Figura 67	Estabilización de datos	113
Figura 68	Correlación de la estabilización	113
Figura 69	Suavizado exponencial de estabilización	114
Figura 70	Anomalía de perdida de datos	115
Figura 71	Correlación de perdida de datos	115
Figura 72	Inestabilidad de datos	116
Figura 73	Correlación de la inestabilidad	117
Figura 74	Suavizado exponencial de inestabilidad.....	117
Figura 75	Visualización de variables presión - variador	120
Figura 76	Arquitectura del sistema de monitoreo.....	122
Figura 77	Leyenda del módulo de control	134
Figura 78	Circuito de alimentación y protección.....	135
Figura 79	Esquemático - puerta del tablero	136
Figura 80	Esquemático- tablero de control.....	137
Figura 81	Variador de velocidad	138
Figura 82	Módulo de entradas digitales.....	139
Figura 83	Salidas digitales.....	140
Figura 84	Módulo de entradas analógica.....	141
Figura 85	Distribución de borneras	142



ÍNDICE DE ANEXOS

	Pág.
ANEXO 1 Elementos de control de presión constante	132
ANEXO 2 Planos eléctricos	134
ANEXO 3 Hoja de especificación del TRB245	143
ANEXO 4 Hoja de especificación del Raspberry PI.....	144
ANEXO 5 Hoja de especificación de MateBook D15	145
ANEXO 6 Declaración jurada de autenticidad de tesis	146
ANEXO 7 Autorización para el depósito de tesis en el Repositorio Institucional.....	147



ÍNDICE DE ACRÓNIMOS

ASCII:	American Standard Code for Information Interchange
AWS:	Amazon Web Services
DAaaS:	Data Analyst as a Services
GCP:	Google Cloud Platform
HMI:	Human Machine Interface
HTTPS:	Hyper Text Transfer Protocol Secure
IAM:	Identity and Access Management
IIoT:	Industrial Internet of Things
IoT:	Internet of Things
IP:	Internet Protocol
JSON:	JavaScript Object Notation
LoRaWAN:	Long Range Wide Area Network
MQTT:	Message Queuing Telemetry Transport
PID:	Proportional Integral Derivative
PLC:	Programmable Logic Controller
RTU:	Remote Terminal Units
SCADA:	Supervisor Control and Data Acquisition
SSL:	Secure Sockets Layer
TCP:	Transmission Control Protocol
TLS:	Transport Layer Security
VDI:	Verein Deutscher Ingenieure



RESUMEN

En la actualidad, la industria 4.0, impulsada por los avances tecnológicos como la Internet de la Cosas (IoT), inteligencia artificial y automatización industrial, han transformando los procesos de control y la toma de decisiones basada en datos en tiempo real. La presente tesis se centra en diseñar un monitoreo remoto de una sala de control de presión constante, haciendo uso de tecnologías de Industria Internet de la Cosas (IIoT). El objetivo principal es mejorar los reportes para la toma de decisiones basados en Análisis de Datos como Servicio (DAaaS). Para llevar a cabo la investigación, se desarrolló una arquitectura de software que interactuar entre el proceso de control de presión constante y la plataforma IIoT DAIREL. Para la validación de la investigación, se aplicó la metodología VDI (sociedad alemana de ingenieros), la directriz 2206 modela el diseño mediante matrices morfológicas, analiza los aspectos técnicos – económicos para una solución óptima; y la ingeniería de requisitos permite seleccionar y configurar elementos para la transmisión, almacenamiento y envío de datos a la nube. Para el análisis de datos como servicio (DAaaS), se consultó datos de 30 días, donde se analizó procesos (muestras) en series de tiempo; estabilización, anomalía y la inestabilidad. Los resultados obtenidos mediante la correlación de Pearson y el suavizado exponencial de datos generó reportes basado en datos a través de la lectura, almacenamiento y consulta de datos; permitiendo la detección temprana del encendido, apagado y fallas durante su operación de la planta de control. Las conclusiones resaltan la integración de las tecnologías de la industria 4.0 en la optimización del proceso de control de presión constante. Se resalta la adaptabilidad del diseño propuesto, la cual puede ser aplicado en otros procesos de control industriales.

Palabras clave: Industria 4.0, Modbus, AWS, DAaaS.



ABSTRACT

Nowadays, Industry 4.0, driven by technological advances such as the Internet of Things (IoT), artificial intelligence, and industrial automation, is revolutionizing control processes and data-driven decision-making in real-time. This thesis work focuses on designing remote monitoring for a constant pressure control room, utilizing Industrial Internet of Things (IIoT) technologies. The main objective is to enhance reporting for decision-making based on Data Analytics as a Service (DAaaS) To conduct the research, a software architecture was developed that allows interaction between the constant pressure control process and platform IIoT DAIREL. For research validation, the VDI methodology (German Society of Engineers) and guideline 2206 are used. This allows modeling the design through morphological matrices, analyzing the technical-economic aspects for an optimal solution. Requirement's engineering enables the selection and configuration of elements for data transmission, storage, and sending to the cloud. For the Data Analysis as a Service (DAaaS), data from the past 30 days was queried, involving the analysis of processes (samples) within time series: stabilization, anomaly, and instability. The results, obtained through Pearson correlation and exponential data smoothing, generated reports based on the data. This is achievable through the requiring, storage and consulting, allowing for early detection of start-up, shut-down, and operational failures. The conclusions highlight the integration of Industry 4.0 technologies in optimizing the constant pressure control process. The adaptability of the proposed design is emphasized, as it can be applied to other industrial control processes.

Keywords: Industry 4.0, Modbus, AWS, DAaaS.



CAPÍTULO I

INTRODUCCIÓN

La industria 4.0 ha experimentado un aumento debido a los avances tecnológicos como el internet de las cosas (IoT), la inteligencia artificial, la automatización industrial, los procesos y análisis de datos. Asimismo, la cuarta revolución industrial enfatiza la toma de decisiones basada en datos, la cual posibilita una mejor gestión de la cadena de suministros (equipos, materiales y recursos) mediante el monitoreo en tiempo real.

Una sala de control industrial es sumamente importante, ya que proporciona información a través de los procesos de control. Este hecho se logra gracias a la interconexión de dispositivos, sensores, máquinas mediante protocolos de comunicación industriales y hacer el uso del Internet de la Cosa; lo que conduce a un proceso de control inteligente en la Industria Internet de las Cosas (IIoT). Sin embargo, esta automatización industrial toma tiempo, debido a que aún está en adaptación.

En este proyecto de investigación, se diseña un sistema de un monitoreo remoto en una sala de control aplicando la tecnología de la IIoT, un diseño que permite interactuar los servicios de la nube y la sala de control industrial; en bajo coste económico y aplicable en otros procesos de control inteligente. Además, de la posibilidad de obtener datos adicionales y aprovecharlos en la toma de decisiones más certera en análisis de datos como servicios (DAaaS). El objetivo de la investigación es adaptar un diseño de un monitoreo remoto con la tecnología de la industria 4.0, el diseño no se utilizará únicamente en un laboratorio de control, ya que puede adaptarse y aplicarse en muchos campos de la industria, y podría mejorar en la minería, refinería y centros hidroeléctricos.



1.1 PROBLEMA

En la industria, la falta de un sistema eficiente de monitoreo en tiempo real podría afectar en la operación de la planta de presión, lo que genera gastos inmensos a futuro. Además, la ausencia de métodos de almacenar y analizar datos de la presión constante y de más elementos de la planta puede dificultar la identificación temprana de problemas y la toma de decisiones. El diseño de un sistema de monitoreo basado en los servicios de la nube podría mejorar las limitaciones de la planta de presión constante.

El primer obstáculo a superar, es adquirir datos de una planta de control de presión constante mediante el protocolo industria Modbus, la planta de control proporciona datos para la visualización en HMI y no en los servicios en la nube. Se debe desarrollar una arquitectura de software para la adquisición (Modbus) y envío (MQTT) de datos. La arquitectura de software deberá respaldar los paquetes de datos en formato JSON con los certificados de la nube. Finalmente, los datos deben ser publicados y almacenados en base de datos; que permita consultar en series de tiempo.

La validación del diseño es otro desafío, que comprende un diseño de una planta de control de presión constante, la arquitectura de software – hardware y la computación en la nube. Dado que el diseño es complejo, será necesario aplicar una metodología de investigación que permita determinar las soluciones industriales y académicas para un proceso de estandarización y adaptarse en otras plantas de control industrial.

1.2 OBJETIVO

Diseñar un monitoreo remoto de la sala de control de presión constante haciendo el uso de la tecnología de la industria 4.0, que permita adquirir datos a través de protocolos de comunicación y visualizar en una plataforma elástica.



- Diseñar un sistema de comunicación MQTT para monitorear la sala de control de presión constante.
- Desarrollar una arquitectura de software en función del protocolo de comunicación Modbus.
- Estructurar una base de datos en series temporales en Amazon Web Services (AWS).
- Relacionar las variables de la planta de presión en series de tiempo.

1.3 HIPÓTESIS

El diseño de un monitoreo remoto integrado con la tecnología de la industria 4.0 mejorara el mantenimiento de los equipos de la sala de control de presión constante, y generara reportes en tiempo real, la detección de anomalías y predice fallas.

1.3.1 Hipótesis específicas

- El diseño de un sistema de comunicación MQTT permite monitorear la sala de control de presión constante.
- La arquitectura de software del protocolo de comunicación Modbus lee registros y genera el formato JSON.
- La estructura de datos permite almacenar datos en series temporales en AWS y hacer las consultas.
- La relación entre las variables de la planta permite hacer el análisis de datos como servicio (DAaaS)



CAPÍTULO II

REVISIÓN DE LITERATURA

2.1 ANTECEDENTES DE LA INVESTIGACIÓN

Para dar con el diseño del sistema, es necesario investigar si existe soluciones en industrias y/o laboratorios sobre el diseño que se desea desarrollar, y esto conlleva revisar las investigaciones sobre protocolos de comunicación Modbus y plantas de presión; recopilar información relacionado al diseño del sistema y agruparlo las investigaciones a nivel internacional y nacional.

2.1.1 A nivel internacional

Sobre la industria 4.0, se revisaron direcciones web como proyectos y/o artículos de investigación de cómo ha cambiado en el mercado y las nuevas tecnologías a favor de las industrias.

El artículo de investigación titulado “Industria internet de las cosas y sus aplicaciones en la industria 4.0: Estado de Arte”, este artículo sirve como punto de partida, donde destaca las nuevas tecnologías que brinda oportunidades significativas para la mejora de la vida, industria y los procesos empresariales. En el artículo también aborda las aplicaciones en las industrias, dispositivos integrados, monitoreo, agricultura, redes inteligentes y atención médica, por lo cual se destaca el uso de IIoT y sus beneficios. (Malik et al., 2021)

La visión hacia el futuro es un punto a considerar, ya que toda tecnología normalmente proyecta para el cambio y la adaptabilidad, tal como se titula el artículo “Creando una visión para el futuro de las salsas de control industriales bajo la industria 4.0, Suecia” analiza el impacto posible de Industria 4.0; solución



basada en Internet que tiene el potencial para producciones autónomas más flexibles capaces de administrarse automáticamente, en las industrias metalúrgicas. Por lo tanto, el autor detalla los cambios que son deseables para lograr una industria sostenible, eficaz e igualitaria y representar los posibles efectos positivos y negativos de implementar la Industria 4.0. (Sundström, 2019)

También se desarrolló en India el artículo titulado “Tecnologías de la Industria 4.0 incorporadas con la Automatización del Hogar Inteligente basada en Delta PLC para el Desarrollo Rural” donde analiza el desarrollo de un sistema de automatización del hogar utilizando un controlador lógico programable (PLC); y conectarse con otros dispositivos como GSM para proporcionar control de voz y protección para personas mayores y discapacitadas. Lo interesante es que el uso de un controlador PLC ofrece soluciones para domótica, y es un detalle para considerar en la presente investigación ya que el artículo concluye que a la automatización de hogares minimiza el desperdicio de energía. (Sarangapani, 2021)

Para conceptualizar en que se basa la industria 4.0, se revisó artículos relacionados a una industria; es decir una sala de control que esté relacionado a una palta de control de presión tal como el artículo titulado “Diseño constructivo de sistema de control de presión constante en sistema de distribución de agua con método PID mediante PLC basado en IoT”. Este artículo de investigación tiene como objetivo de implementa un sistema de control presión contaste para la distribución de agua; utiliza el método de control PID mediante un controlador lógico programable Siemns S7-1200, obtenido un sistema simple que compone de una bomba trifásica que su función es suministrar agua desde en un tanque; mediante la comunicación de Wi-Fi los datos recolectados del controlador son



enviados a un servidor nube, esto es posible mediante el Raspberry Pi 3 y el protocolo MQTT. (Syufrijal, 2019)

Este artículo titulado “Domótica de bajo costo mediante Arduino y protocolo Modbus, Irán” da la importancia de los hogares inteligentes y la automatización; para dar un alto rendimiento en la calidad de vida y reducir el consumo de energía gradualmente. Este sistema domótico tiene un diseño para el costo bajo, usa el microcontrolador Arduino y el protocolo Modbus. El sistema se basa en un diseñado para controlar y monitorear los dispositivos eléctricos, lo que reduce las pérdidas de energía. (Hassanpour et al., 2017)

La tesis titulada “Diseño y simulación de un sistema de monitoreo para un proceso de control de temperatura mediante PLC e IoT para la empresa Educativa, Colombia”, diseña de un proceso de control de temperatura con integración de PLC y IoT, que busca capacitar a personas en la industria y la Revolución 4.0. Este banco permite conocer conceptos de programación de PLC, trabajar con instrumentos de medición de procesos y tecnología IoT para enviar información a la nube y registrar datos en tiempo real en plataformas como Ubidots. (Muñoz Díaz, 2021)

Este proyecto de tesis titulada “Diseño y desarrollo de un prototipo de módulo de alarma contra incendios para la universidad Politécnica salesiana sede cuenca” tiene como objetivo de diseñar y desarrollar un módulo de alarmas contra incendios para mejorar la seguridad en la Universidad Politécnica Salesiana en Cuenca, Ecuador. Este proyecto se desarrolla debido a que el sistema de alarma que éste ha fallado debido a la falta de mantenimiento y conexiones defectuosas. El prototipo desarrollado de módulo de alarma, utiliza dispositivos como



iniciadores y paneles de control que monitorea y detecta incendios mediante comunicación Wi-Fi y Ethernet. La programación se basa en secuencia de nodos - Node-RED -, mantenido por IBM, y el tiempo de respuesta es rápido. (Merchan Damian & Damian Tapa, 2022)

2.1.2 A nivel nacional

La investigación que se desarrolló en Perú titulada “diseño SCADA para monitorear alarmas contra incendio del hospital regional de Lambayeque Chiclayo 2017”; propone que el desarrollo de un sistema SCADA para el sistema contra incendios del Hospital Regional de Lambayeque. Dónde, realiza un inventario de todos los equipos sensores y actuadores en los seis pisos del dicho hospital, y diseña el sistema SCADA con el objetivo de monitorear y controlar varios parámetros, como la visualización y reset de las luces estroboscópicas, las presiones y caudal en las tuberías principales de cada sector, y las válvulas solenoides de apertura y cierre total en las tuberías principales. El sistema SCADA utiliza un controlador para la comunicación y supervisión. (Rodriguez Cera, 2018)

La tesis “Sistema de control de flujo para presión constante de suministro de agua utilizando PLC en un edificio del instituto de Ciencias y Humanidades” moderniza el sistema de distribución de agua, y ejecuta un sistema de presión constante, reduciendo el consumo de energía que en consecuencia conservar los equipos en condiciones estables mientras está en uso (mantenimiento preventivo); incluye el uso de un controlador, variador de frecuencia, un sensor de presión marca DANNFOSS y motores de posición vertical marca WE. (Díaz Rodas & Trujillo Aramburú, 2019)

2.2 REFERENCIAS TEÓRICAS

2.2.1 Industria 4.0

El concepto de la industria 4.0, surge en Alemania 2011, donde hace referencia a una nueva política económica basada en el uso de la energía de alta tecnología; la inteligencia artificial, la digitalización de procesos de control, y el uso dispositivo electrónicos avanzados. Igualmente, por el volar agregado y la personalización de la producción, la prestación de servicios y el intercambio de información entre maquina y humano. (Mosconi, 2015)

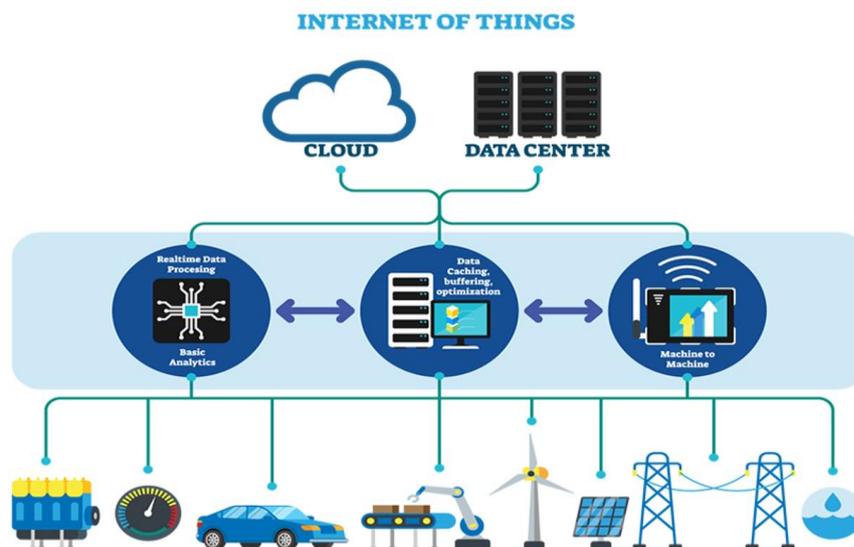
Durante los últimos años el término del internet de las cosas (IOT) se ha convertido un gran interés para los académicos e industrias. El término de la industria IoT tomó importancia cuando la TI (tecnología de la información) fue integridad con la automaticen industrial y sistema de control (Kamarka et al., 2019). La industria internet de las cosas ayuda la complejidad en la manufactura de proceso, es decir incrementa la productividad y disminuye tiempo – costo, lo que conlleva tener un ingreso mayor de lo convencional. También, permite realizar un análisis de datos económico de la producción y evaluación.

2.2.1.1 Internet de las Cosas

La tendencia en los últimos años es conectar dispositivos a la nube, esto es conocido como IoT (Internet of Things); los avances del hardware y las capacidades de la computación en la nube hacen posible la comunicación fluida entre personas y los dispositivos.

Figura 1

Esquema general de internet de las cosas



Nota: Esquema general de IoT. Adaptado de *Introducción general a IoT*, por A. Bassi, 2021. (https://www.gotoiot.com/pages/articles/iot_intro/index.html). Dominio público.

Kevin Ashton en 1999 introduce la expresión de “Internet de las Cosas” (traducida) donde se interpreta como la interconexión de dispositivos físicos a través del internet a un servidor que permita controlar y monitorear desde los servicios de la nube (figura 1), esto es posible mediante los datos que emite los equipos conectados. El costo de su implantación es bajo, algunas veces puede ser un ahorro significativo para la economía (Bassi et al., 2013)

2.2.2 Automatización industrial

La automatización industrial es un proceso mediante el cual se utiliza tecnologías entre equipos o dispositivos, con el objetivo de llevar a cabo una tarea o un proceso de control. Una sala de control es muy importante dentro de este proceso de la automatización industrial; un espacio físico dentro de una instalación en una industria donde se pueden mostrar equipos y sistemas necesarios para supervisar, monitorear y controlar diversas operaciones y procesos. Además de la toma de



decisiones administrativas industriales, por este espacio se puede monitorear y analizar la producción de cada planta de control; la seguridad de cada proceso, la seguridad del personal y las medidas a cualquier fallo dentro del proceso. (Mehta & Reddy, 2015)

2.2.3 Sistema de control presión constante

El concepto de una planta hace referencia a un conjunto de elementos de una maquina o dispositivos electrónicos y/o eléctricos que trabaja en conjunto, cada dispositivo y/o maquina tiene una función particular; su objetivo principal es ejecutar una operación con los elementos físicos como sensores y actuadores. (Ogata, 2010)

Una planta de control de presión constante es un espacio físico dentro de una instalación en una industria donde se puede mostrar equipos y sistemas necesarios para mantener la presión de un ambiente o una sala de manera constante. Los equipos típicamente incluyen sistema de ventilación y aire acondicionad, compuertas, filtros de aire y válvulas de control de flujo, sensores y medidores de presión. Estos equipos componen una sala de control de presión constante, asegurando que ese mantenga dentro de los parámetros establecidos. (Díaz Rodas & Trujillo Aramburú, 2019)

2.2.3.1 Controlador lógico programables (PLC)

Para el desarrollo de las industrias es muy común usar controladores sea tipo industrial (PLC) o comercial de uso académico (Rasberry PI), un controlador es indispensable en una planta de control, su función principal es leer, escuchar, procesar datos de los equipos electrónicos o eléctricos.



Los PLCs son ampliamente usados en las industrias, es de uso sencillo y a media que el tiempo estos equipos van reducido su costo. Estos equipos son robusto y confiable, y es muy adecuado para las industrias que ya hace décadas se ha convertido una herramienta esencial en la automatización industrial. (SIEMENS, 2018)

2.2.3.2 Actuadores y trasmisores Industriales

Los actuadores son dispositivos electromagnéticos que utilizan una señal de control en una acción física, su función principal es adaptar cambio en un sistema mecánico, eléctrico e hidráulico. (Creus Solé, 2011)

Son dispositivos que reciben señal generada de los sensores y acondiciona las señales para transmite a controladores. Dentro de la evolución de la instrumentación, la aparición del transmisor fue en 1983, por parte de la firma Honeywell. Los transmisores son componentes que se usan para la automatización industrial y control de procesos. (Creus Solé, 2011)

2.2.3.3 Software de diseño (CAD)

Para dar con un diseño con los estándares internacional, la compañía AutoDesk sé dedicada a desarrollar softwares para industrias e ingeniería como el desarrollo de prototipos, planos eléctricos y diseño 3D. Para la industria, los diseños eléctricos son desarrollador en software CAD (*Computer-Aided Design*). Debido a su flexibilidad y las normas establecidas en una industrial.

Auto Cad Electrical: Desarrollado por el grupo de AutoDesk, es ampliamente utilizado en las industrias, especialmente para diseños y documentación que incluyen planos eléctricos. Este software permite a la



ingeniería rubro de la automatización e instrumentación en la industria eléctrica; crea diagramas y esquemas eléctricos, simulación y análisis eléctricos, generación de lista de materiales (BOM) e incluye bibliotecas de símbolos y componentes eléctricos estandarizado.

2.2.3.4 Software programación industrial

Los softwares de programación industrias son esenciales para desarrollar y/o aplicar, cada compañía que está abocado en fabricar controladores los desarrolla conjuntamente con un software, como Siemens o Shneider Electrical entre otros. La gran ventaja de estos softwares que ya están integrados y esto facilita la programación y adquisición de datos. (Siemens TIAPortal, 2023)

- TIA Portal: Software desarrollado por Siemens, que permite programar, configurar, y ejecutar en los PLC marca Siemens, y esto incluye otros equipos de automatización industrial.
- S7-Siemenes: Son PLCs dentro del software TIA Portal con diferentes modelos como el S7-1200, S7-1500 y los S7-xxx estos PLCs se pueden programar en TIA Portal de Siemens, que permite programar, configurar y poner en marcha los productos de Siemens.

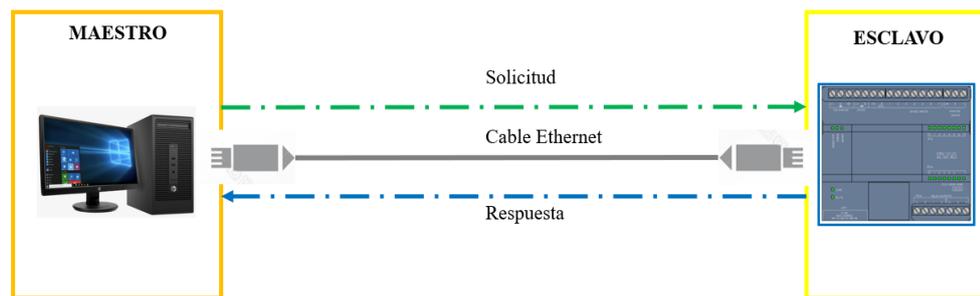
2.2.4 Protocolo Modbus

La comunicación entre equipos en la automatización industrial se basa en protocolos, es decir que cada equipo tiene un protocolo de comunicación y un conector que los enlaza entre sí. Modbus es uno de ellos, por ser práctico y desarrollado para industrias.

Modbus es un protocolo de comunicación abierta, y su uso es muy frecuente en la automatización industrial; establecer comunicación entre dispositivos electrónicos es la función principal. Modbus permite el intercambio de información entre un dispositivo maestro y un esclavo basándose en la comunicación serial como RS-232, RS -485 y ethernet (TCP/IP), Modbus sigue una arquitectura de cliente servidos, donde el dispositivo maestro inicia con las solicitudes (leer – escribir datos) hacia los dispositivos esclavos (solicitud – respuesta). (Mehta & Reddy, 2015)

Figura 2

Modbus maestro - esclavo

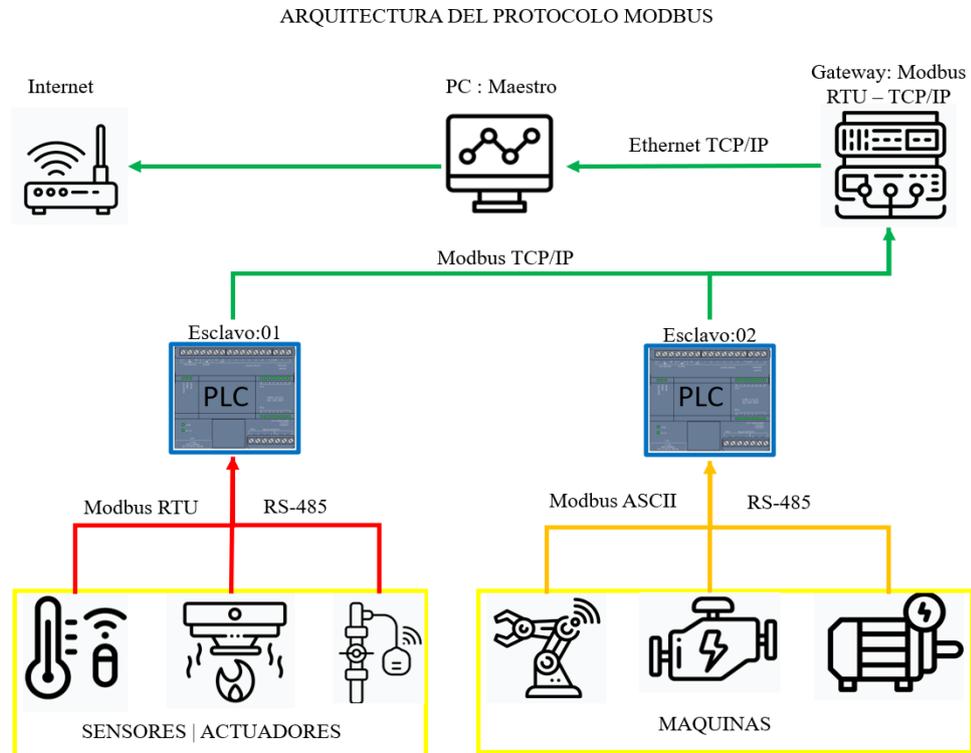


Nota: Adaptado de MODICON Inc., Industrial Automation Systems, 1996. (https://www.modbus.org/docs/PI_MBUS_300.pdf). En dominio público.

Existen diferentes variantes de Modbus, que incluyen Modbus RTU (formato binario sobre comunicación en serie), Modbus ASCII (formato ASCII sobre comunicación en serie) y Modbus TCP (usando Ethernet y TCP/IP). Este protocolo de comunicación define tipo de datos y códigos de función, específicamente tabla de registros para determinar la operación que debe de realizar. (Ruiz Olaya et al., 2004)

Figura 3

Arquitectura general del protocolo Modbus



Nota: Adaptado de MODICON Inc., Industrial Automation Systems, 1996, por el Centro de Formación Técnica para la Industrial, 2023. En dominio público.

2.2.4.1 Trama Modbus

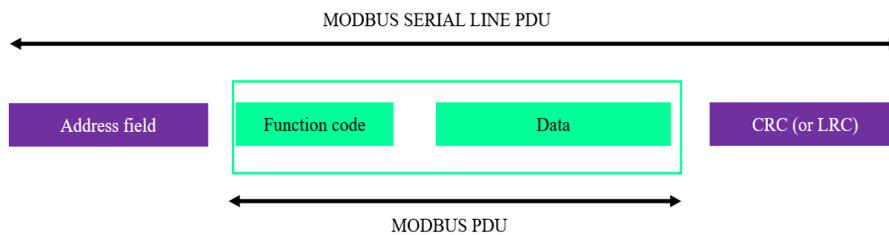
La trama de Modbus se utiliza para enviar información en paqueteada entre el maestro y esclavo. Modbus serial en línea PDU; es una parte muy importante de la trama de Modbus, que contiene los datos muy específicos de la solicitud o respuesta a través del protocolo Modbus. En la figura se puede observar de cómo se forma una trama Modbus.

- **Dirección del dispositivo del esclavo (Address field):** la trama empieza con 1 bit, es el bit de la solicitud entre esclavo y maestro.
- **Código de función (Function code):** es la función para escribir o leer dentro de una solicitud.

- **Datos (data):** es el valor específico que se desea leer o escribir.
- **Checksum (CRC):** es para que trama sea íntegro. Por lo visto el CRC se calcula en función de los datos transmitidos.

Figura 4

Trama de Modbus Serial



Nota: Adaptado de MODICON Inc., Industrial Automation Systems, 1996. (<https://slideplayer.com/slide/1594282/>). En dominio público.

2.2.4.2 Funciones Modbus

Las funciones de Modbus son estándares, es decir que equipos que tenga protocolo de comunicación de Modbus puede conectarse entre sí, no importa la marca que sea. Las funciones de Modbus, también son conocidas como código de función.

Tabla 1

Código de funciones Modbus

Código	Función	Nombre de la tabla
01 (01 hex)	Read	Discrete output coils
05 (05 hex)	Write single	Discrete output coil



15 (0F)	Write multiples coils	Discrete output coils
02 (02 hex)	Read Discrete Inputs	Discrete input contacts
04 (04 hex)	Read Input Registers	Analog input registers
03 (03 hex)	Read Holding Register	Analog output holding register
06 (06 hex)	Write single	Analog output holding register
16 (10 he)	Write multiple	Analog output holding registers

Nota: Adaptado de *Las funciones del protocolo de Modbus son estándares*, de Mehta y Reddy, 2015. Dominio público.

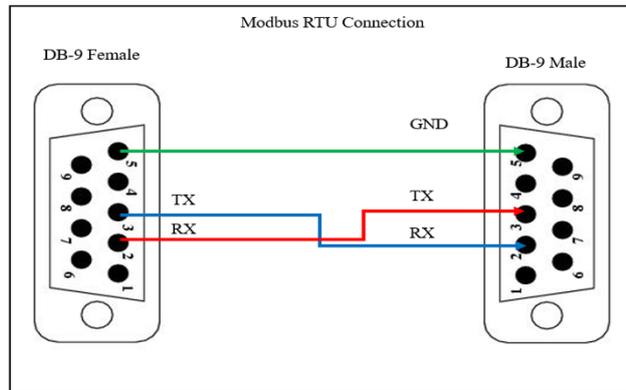
2.2.4.3 Modbus RTU

Modbus Unidad Terminal Remota o Modbus RTU es un protocolo ampliamente utilizado en la automatización industrial, la comunicación es serial; normalmente el serial utilizado es RS-485 para la comunicación entre los dispositivos electrónicos. (Modbus, 2023)

La conexión del Modbus RTU permite el intercambio de datos entre dos dispositivos o más (maestro – esclavo). Este protocolo de comunicación usa la tecnología UART.

Figura 5

Conector DB9

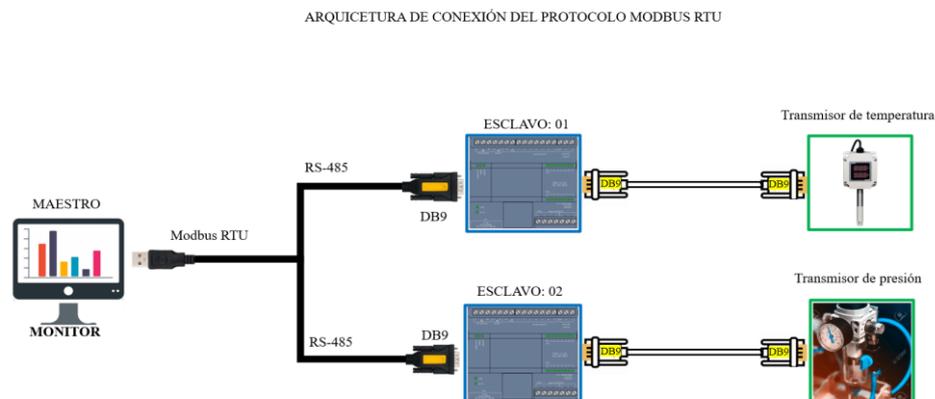


Nota: Adaptado de MODICON. Conector DB9

Para hacer la conexión de maestro y esclavo en Modbus RTU se sugiere usar el conector DB9 (Figura 5), ampliando el concepto del conector, puede ser macho o hembra según a la conexión que se realizará.

Figura 6

Arquitectura de conexión del protocolo Modbus RTU



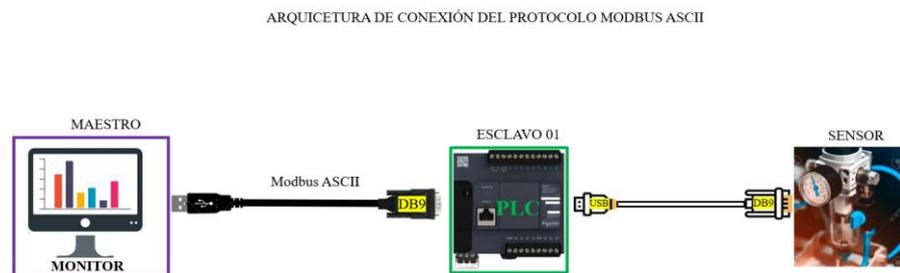
Nota: Adaptado de MODICON Inc., Industrial Automation Systems, 1996. (https://www.modbus.org/docs/PI_MBUS_300.pdf). En dominio público.

2.2.4.4 Modbus ASCII

En este protocolo de comunicación se utiliza los caracteres ASCII legibles para representar datos, cada valor de datos se codifica en una secuencia ASCII que está representada en un valor hexadecimal.

Figura 7

Arquitectura de conexión de Modbus ASCII



Nota: Adaptado de MODICON Inc., Industrial Automation Systems, 1996. (https://www.modbus.org/docs/PI_MBUS_300.pdf). En dominio público.

2.2.4.5 Modbus TCP/IP

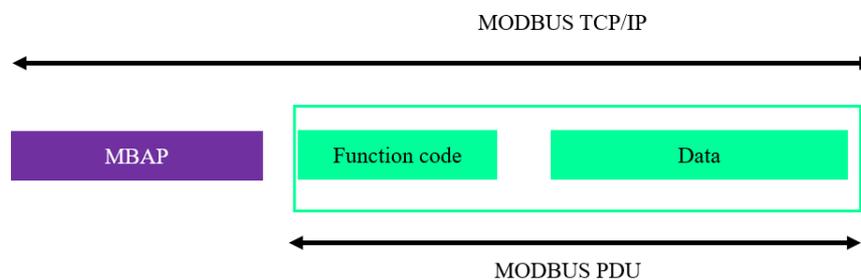
Este protocolo de comunicación envía datos entre equipos a través de una red Ethernet. Es decir, usar la capa física de Ethernet y el Protocolo de control de transmisión (TCP). Al usar esta comunicación se tiene una arquitectura cliente – servidor. Para detallar el cliente Modbus TCP/IP inicia las solicitudes para leer datos del servidor Modbus TCP/IP. Para entrar en detalle se puede dar concepto que el Modbus TCP/IP cliente es maestro y el Modbus TCP/IP servidor es el esclavo. Generalmente el protocolo de comunicación Modbus TCP/IP usa un puerto 502 para la comunicación, este número puede variar según el equipo y/o la configuración. Y el cliente se conecta mediante la dirección IP del servidor, es así como se establece la conexión del cliente – servidor. (Modbus Organization, 2006)

Trama de Modbus TCP/IP:

- La trama usa la arquitectura de red TCP/IP para hacer la solicitud y envío de datos. Tal como se ve en la figura, el MBAP (Modbus Application Protocolo Header) es como el campo de la dirección.
- MBAP (Modbus Application Protocolo Header): consta con 7 bytes, es el bit de la solicitud entre esclavo y maestro.
- Código de función (Function code): es la función para escribir o leer entro de una solicitud.
- Datos (data): es el valor especifico que se desea leer o escribir.

Figura 8

Trama de Modbus TCP/IP



Nota: Adaptado de MODICON Inc., Industrial Automation Systems, 1996. (https://www.modbus.org/docs/PI_MBUS_300.pdf). En dominio público.

Tabla 2

MBAP (Modbus Application Protocolo Header)

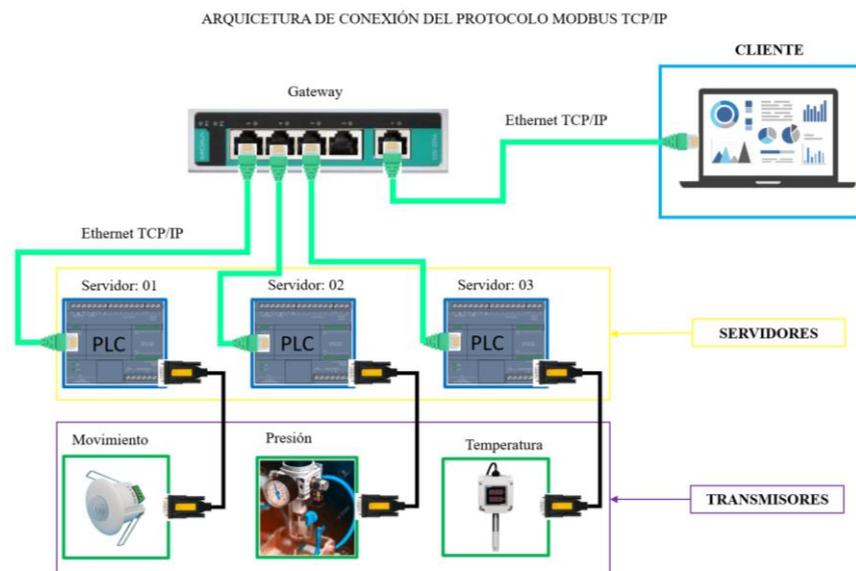
	Tamaño	Descripción	Cliente	Servidor
Identificador de Transacción	2 bytes	Identifica la solicitud	Inicializa	Solicitud recibida
Identificador de protocolo	2 bytes	0= MODBUS protocolo	Inicializa	Solicitud recibida

Tamaño	2 bytes	Número del siguiente byte	Inicializa (solicitud)	Inicializa
Identificador de unidad	1 bytes	Identificación del esclavo	Inicializa	Solicitud recibida

Nota: adaptado de Modicon (1996).

Figura 9

Arquitectura de conexión del protocolo Modbus TCP/IP



Nota: Adaptado de MODICON Inc., Industrial Automation Systems, 1996. (https://www.modbus.org/docs/PI_MBUS_300.pdf). En dominio público.

2.2.4.6 Modbus PLUS

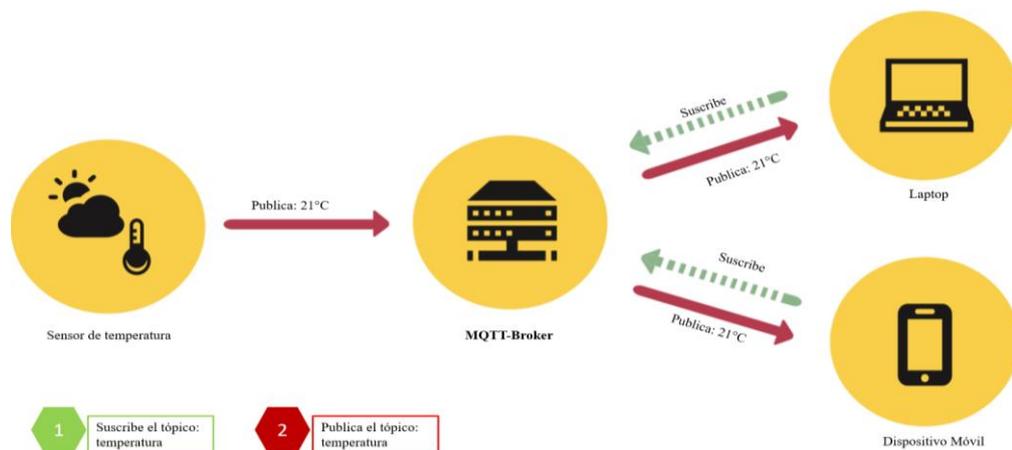
Modbus plus es una versión mejorada del Modbus, al igual que sus anteriores versiones, también fue desarrollada por Modicon, su desarrollo fue impulsado para mayor velocidad entre equipos de una misma marca, es decir que este Modbus usa un token para su acceso con equipos que tenga el mismo protocolo de comunicación, esto permite que tenga mayor velocidad al transmitir datos, corregirlos y retransmitirlos. (Scheider Electric, 2023)

2.2.5 Protocolo MQTT

Para que sea posible la conexión entre dispositivos electrónicos como los gadgets a un servidor – nube existen protocolos de comunicación como el MQTT, que permite conectar. Este protocolo de comunicación fue desarrollado por IBM y Cirrus Link Solution en 1999. El MQTT se base en un modelo de publicación – suscripción, y los mensajes se envían mediante el bróker MQTT. Este protocolo se adecua al entorno de Internet de la cosa (IoT), es decir una comunicación eficiente y un bajo consumo de energía. Tal como se muestra en la imagen. (HiveMQ, 2023)

Figura 10

Protocolo MQTT



Nota: Esta topología representa la conexión entre el suscriptor y publicador. MQTT Essentials, 2023. (<https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt/>). En dominio público.

Las principales características del protocolo de comunicación MQTT:

- **Publicación/suscripción:** Publicador (dispositivos IoT) que publican mensajes a través de un tópico “topics”, tanto como el publicador y el suscriptor debe tener el mismo tópico.



- **Broker:** Es responsable de que exista la comunicación entre el publicador y el suscriptor.
- **QoS (Calidad del servicio):** Al configurar un dispositivo se debe tener en cuenta la calidad de fiabilidad, por lo general son tres:
- **QoS 0:** Los mensajes se envían sin garantías de entrega.
- **QoS 1:** Los mensajes se envían y se repiten hasta que se confirma su recepción.
- **QoS 2:** Los mensajes se envían y se confirman tanto por el emisor como por el receptor para asegurar que se entreguen exactamente una vez.
- **Mensajes retain:** permite que los mensajes esten retenidos en el bróker.
- **Puerto:** Predeterminando 1883, 8883 y SSL/TLS.

2.2.6 Sistemas embebidos

Los sistemas embebidos son tecnologías de mini computadoras integrados con microcontroladores y está diseñado para funciones dedicadas a la IoT; realizan funcione específicas, donde funcionan prolongadamente sin interrupciones. Destaca por su fiabilidad, donde son eficientes energéticamente, están construidos para consumir menor cantidad de engeriría. Tiene recursos limitados en la capacidad de almacenamiento y procesamiento, esto permite que inicie rápidamente debido a los componentes de un sistema embebido (CEUPF MAGAZINE, 2023)



- **Microcontrolador:** controla toda la operación, ejecuta las instrucciones y deriva las gestiones de trabajo de un proceso.
- **Memoria:** integrado con la memoria RAM (Random Access Memory) que cumple la función de almacenar datos temporales; la memoria flash permite almacenar programas dentro del sistema.
- **Periféricos:** permite la interacción con otros equipos electrónicos, sensores, actuadores y los puertos de entrada y salida.
- **Software:** comprende el conjunto de programas que ponen en marcha los procesos y funciones específicas; es la parte inteligente del sistema de embebidos.
- **Fuente de energía:** proporciona energía reducida para que el sistema funcione correctamente; baterías y/o energía renovable.

2.2.6.1 Sistema de control: Raspberry PI

Son sistemas que se componen por entradas, salidas y puertos, que cumple una función de controlar y procesar dispositivos electrónicos, equipos eléctricos. El Raspberry PI es un embebido para desarrollar proyectos de tecnologías, informática y la domótica más simple al igual que una computadora de escritorio. Es muy usado en las aplicaciones de IoT. Su sistema operativo está basado en Linux y adaptándose para el desarrollo de proyectos de robótica y electrónica. (Raspberry Pi, 2023)

Características de un Raspberry PI

- Puerto GPIO



- Tarjeta microSD
- Procesador ARM
- RAM
- Puertos de vídeo HDMI
- Conectividad a internet

2.2.6.2 Sistema de comunicación: Enrutador

Es una de las partes más importantes de la industria 4.0, gracias a los sistemas de comunicación como los enrutadores industriales, que sirven como puerta de enlace entre una planta y un servidor en la nube. Estos enrutadores desempeñan un rol muy importante en garantizar la comunicación confiable y estable entre máquinas, señores digitales y un sistema de control industrial. (TELTONIKA, 2023)

También son dispositivos esenciales para la gestión de adquisición de datos y transferencia de datos; la confiabilidad de adaptarse a las redes Ethernet permite la conexión de equipos mediante protocolos de comunicación.

2.2.7 Frameworks

Los marcos de trabajado (*frameworks*) son conjunto de herramientas y bibliotecas de programación estructuradas que proporcionan una base de desarrollo de aplicaciones de softwares. Para leer los registros del protocolo Modbus, es necesario y trabajar en frameworks que se base en una hoja de especificaciones en industrias, debido que en su mayoría las conexiones son los



puertos de los controladores. Para ello, los frameworks son de una cantidad reducida.

2.2.7.1 JavaScript

Es un software de desarrollo web, ligero con función de primera clase con bibliotecas y módulos con la cual dota una dinámica e interactivas a las páginas web (front-end). (GitHub, 2022)

Es un entorno de tiempo de ejecución, programación versátil y fácil. Está basado en V8 de Chrome que permite ejecutar códigos de JavaScript y también en consolas de un sistema operativo como Linux, Windows y macOS.

Formato JSON: JavaScript Object Notation es un formato de intercambio de datos ligeros para máquinas y usuarios.

2.2.7.2 Python

Fue desarrollado por Guido van Rossum y lanzado en 1991. Es considerado un lenguaje de programación de alto nivel; utiliza en una amplia variedad de áreas, como el desarrollo web y aplicaciones de escritorio, inteligencia artificial, análisis de datos y ciencia de datos. (Python, 2022)

Python como lenguaje programación se pueden ejecutar a través de diferentes sistemas operativos Windows, Linux y macOS, también en IDE Pichar, Atón, Sublime, Visual Studio Conde. Esto permite programar entorno de aplicaciones de escritorio, web y scripts. Python es un lenguaje:

- Interpretado
- Multiplataforma



- Libre
- Programación funcional, imperativa y orientación a objetos

Pymodbus (2023) es una librería de Python desarrollada para comunicación Modbus y es una complementación que usa un síncrona o asíncrono. Basado en protocolos de comunicación de Modbus: TCP, RTU – TCP, UDP, SERIAL, TLS.

2.2.8 Computación en la nube

La computación en la nube, conceptúa a la entrega de servicio de computación, ya sea almacenamiento, servidores, base de datos, etc. Dentro de ello se tiene diferentes modelos de servicios en la nube. Ver la Figura 11.

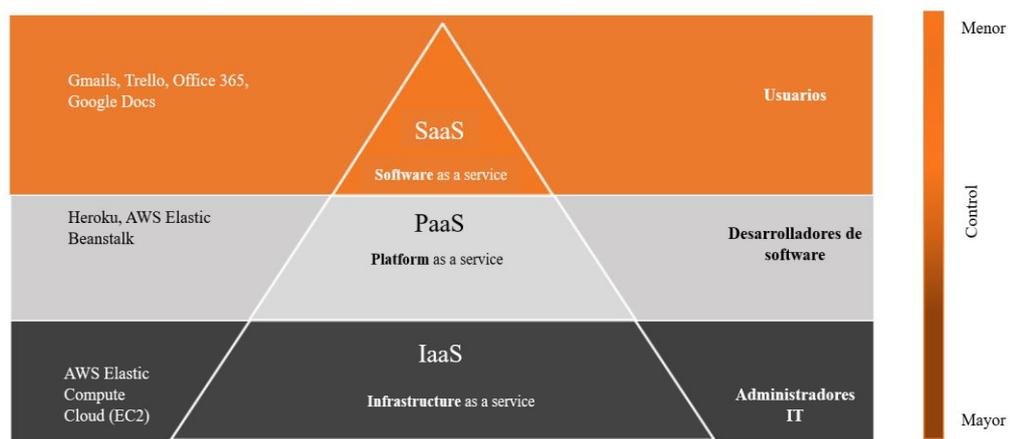
- **Infraestructura como servicio (IaaS):** Gestiona una estructura física propia de tecnología de información, es decir que los usuarios pueden alquilar y utilizar recursos informáticos, como redes, servidores, espacio de almacenamiento y máquinas virtuales. En este mercado de la computación en la nube podemos mencionar a los más importantes proveedores que ofrecen infraestructura como servicio son Amazon Web Service (AWS), Microsoft Azure, Google Cloud Platform (GCP), IBM Cloud y Oracle Cloud Infraestructura (OCI), en si hay algunas más, pero aún están recién introduciendo al mercado virtual.
- **Plataforma como servicio (PaaS):** es una plataforma completa que permite desarrollar, desplegar y gestionar aplicaciones y

lógica empresarial. Proporciona un entorno de desarrollo integrado; herramientas de desarrollo, bibliotecas, lenguajes de programación. Algunas plataformas como servicios tenemos a Google App Engine es una plataforma de desarrollo de aplicaciones web y móviles.

- **Software como Servicio (SaaS):** Es un modelo de servicio que el usuario adquiere a través de un navegador web, aplicación o software mediante instalación, actualización e interacción con el entorno y/ o colaboración. Se puede acceder en cualquier momento con conexión o sin conexión a internet. Estos algunos servicios como infraestructura tenemos el Microsoft 365 y Google Workspace.

Figura 11

Servicios en la nube



Nota: Servicios en la nube, adaptado de AWS, por Gleb B y Dmitry G. 2023, (<https://rubygarage.org/blog/iaas-vs-paas-vs-saas>). En dominio público.



2.2.8.1 Amazon Web Services (AWS)

Amazon Web Services (2023) ofrece un amplio conjunto de productos globales basados en la nube; recursos informáticos, bases de datos, análisis, herramientas para desarrolladores, IoT, seguridad y aplicaciones para empresa. los servicios que ofrece son:

- **AWS IoT Core:** Es un servicio de nube, que permite a millones de dispositivos que estén conectados e interactuar, mediante mensajes que son procesados y dirigidos a puntos de enlace de AWS. La mejor funcionalidad de este servicio IoT Core es realizar seguimiento a todo el dispositivo y comunicar con ellos en tiempo real, incluso cuando estos hayan perdido la conexión. Acuerdo a la guía oficial de Amazon Web Services (2023), AWS IoT Core permite conectar dispositivos IoT o nuevos dispositivos desarrollados que cuente con un protocolo de comunicación compatible que se menciona lo siguiente:
 - MQTT (Message Queuing and Telemetry Transport)
 - MQTT sobre WSS (Websockets Secure)
 - HTTPS (Hypertext Transfer Protocol - Secure)
 - LoRaWAN (Long Range Wide Area Network)
- **AWS Timestream:** Está gestionado para almacenar datos para el proceso de análisis de datos mediante series temporales. Las series temporales son un conjunto de datos recompilados de dispositivos externos de la nube (sensores, plcs, etc.).



- **Amazon Managed Grafana:** es una plataforma de visualización de datos que permite diseñar a los usuarios graficas a través de paneles interactivos y tener la información en tiempo real. Es una herramienta que es comúnmente usado para la supervisión y análisis de sistemas y servicios.

Según AWS (2023) Grafana permite:

- Visualización de datos.
- Conexión a diversas fuentes para la adquisición de datos.
- Alerta y notificaciones.

2.2.8.2 Microsoft Azure

Es una plataforma de servicio, que proporciona servicios y herramienta de desarrollo y gestión en la nube. Es decir, un conjunto de servicios en la nube que permite construir, desarrollar, administrar con herramientas en la nube. (Microsoft Azure, 2023)

Algunas características claves de Azure son:

- **Computación en la nube:** Comprar máquinas virtuales para desarrollar y administrar. También proporcionan almacenamiento estable y seguras para datos. Es muy frecuente el uso de empresa.
- **Base de datos SQL Azure:** ofrece una gran variedad de base de datos como SQL data base, Cosmos DB y Azure SQL.
- **Azure IoT Hub:** es un servicio que permite conectar millones de dispositivos y equipos a la vez, junto con datos y análisis de la IoT.



Los procesos de conexión con la nube de Azure constan con tres simples pasos:

- Equipos externos y/o dispositivos electrónicos que tenga protocolos de comunicación de conectividad – nube.
- Certificados generados por Azure para los dispositivos externos.
- Almacenamiento de datos mediante series de tiempo.

2.2.8.3 Google Cloud Platform (GCP)

Es una plataforma de servicios en la nube. Es un conjunto de herramientas y servicios desarrollados para mejorar los procesos de las empresas en el comercio e innovación.

GCP despliega gran variedad de servicios en la nube como:

- **Cloud SQL:** Almacenamiento escalable de base de datos, para obtener datos externos de equipos y/o dispositivos electrónicos. También permite generar análisis en series temporales.
- **Cloud IoT Core:** Conecta, administra y transfiere datos con facilidad mediante certificados de GCP. Práctico para adquirir datos de equipos y dispositivos electrónicos con protocolos de comunicación como el MQTT o Lora WAN.

Estos servicios permiten al usuario acceder, almacenar, procesar y amistar datos de manera virtual y remota.



CAPÍTULO III

MATERIALES Y MÉTODOS

3.1 METODOLOGÍA

El sistema de un monitoreo remoto de la sala de control de presión constante utilizando las tecnologías de la industria 4.0, se diseñó teniendo como guía la directriz VDI 2206, desarrollada por la Asociación Alemana de Ingeniería. La principal ventaja reside en su forma de organización del proyecto según los criterios técnicos y económicos, basando en la tendencia de las nuevas tecnologías. (Graessler & Hentze, 2020)

3.1.1 Diseño de investigación

El diseño de esta investigación es tipo no experimental, descriptivo - correlacional, donde no se manipulará la variable y el enfoque es cuantitativo donde se analizará la veracidad y certeza de las hipótesis formuladas. (Sampieri et al., 2014)

3.1.1.1 Variables

Las variables de esta investigación son correlacionales, los datos de la presión constante serán analizados mediante servicio en la nube, y el efecto será generar reportes de comportamientos de las variables.

Variable 01: valores de presión constante

Variable 02: valores del variador de frecuencia

3.1.2 Población

La población de datos de esta investigación está constituida de 30 días calendarios del 30 de noviembre a 30 de diciembre del 2023, cabe mencionar que los datos están en series de tiempo. Todos los datos fueron recolectados por la plataforma digital DAIREL (<https://dairrel.com/>).

3.1.3 Muestra

La clase de la muestra es probabilística donde se selecciona mediante procedimiento basado en series de tiempo. De la población general se analizó en tres eventos que son muy significativos para la investigación. El evento 01 describe el comportamiento del PDI estable y el evento 2 describe la anomalía de funcionamiento del proceso de control en series de tiempo. Donde se realizó una Correlación Línea de Pearson interpretado en Python, tomado la siguiente fórmula:

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}$$

Donde:

- x_i e y_i son los valores individuales de las variables x e y respectivamente.
- \bar{x} e \bar{y} son las medias de las variables x e y respectivamente.

El coeficiente de Pearson puede tomar valores en el rango de -1 a 1:

- $r = 1$: Correlación positiva perfecta.
- $r = -1$: Correlación negativa perfecta.



- $r = 0$:No hay correlación lineal.

Para hacer el tratamiento de datos se hará el análisis de suavizamiento exponencial de datos en series de tiempo.

Donde:

$$f_{t+1} = \alpha \cdot y_t + (1 - \alpha) \cdot f_t$$

Donde:

- f_{t+1} : predicción para el próximo periodo
- y_t : observación actual
- f_t : suavización anterior
- α : factor suavizado

3.1.4 Técnica de recolección de datos

La recolección de datos es posible mediante el protocolo MQTT, donde se generó una trama de datos en formato JSON para la primera solución instrumentado con el enrutador industrial y para la segunda solución instrumento con un sistema de embebido se armó una trama Payload. Todos estos datos son recolectados y almacenados en la plataforma DAIREL

3.1.5 Instrumento de recolección de datos

El formato JSON de la primera solución contiene la siguiente sintaxis:

```
{"time": %t, "device":"PLC_SIEMENS", "%r": %a}
```

Donde:



- *%t*: timestamp
- *device*: es el controlador del proceso
- *%r*: solicitud de variables
- *%a*: registro de datos

El Payload que es la segunda solución contiene la siguiente sintaxis:

```
payload = f{{"time": {ts},"device": "{EQUIPO}", "pressure_value":  
{valor_1}, "variator": {valor_2}}}
```

donde:

- *ts*: timestamp
- *device*: controlador del proceso
- *pressure_value*: valor de la presión en el registro 1
- *variator*: valor del variador en el registro 2

3.2 MODELO Y ANÁLISIS

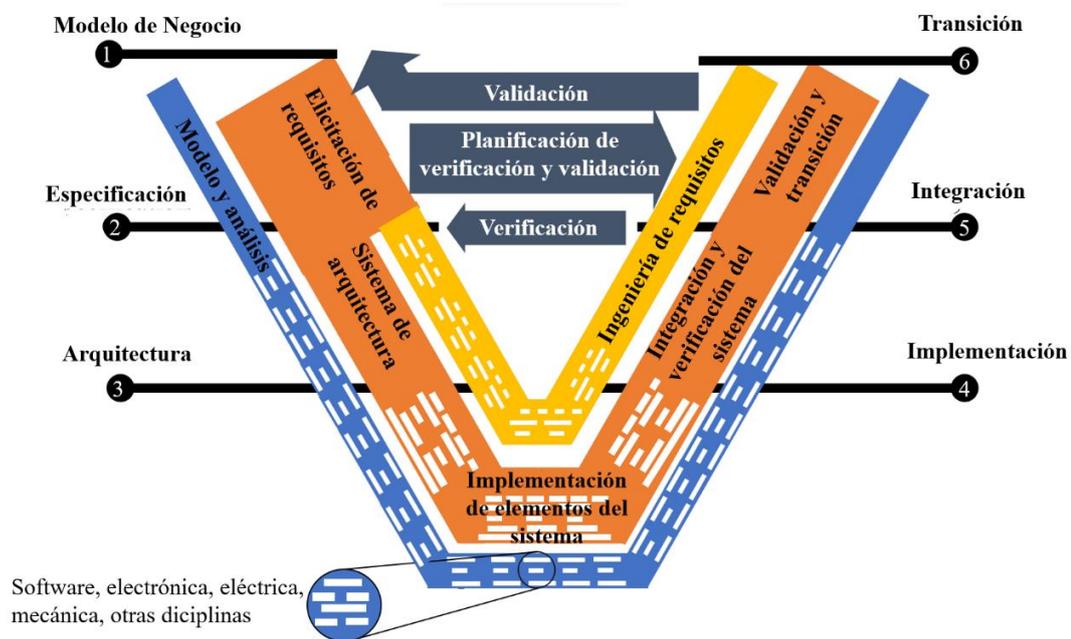
La asociación alemana de ingenieros ha estado integrando y adaptando el nuevo V-Model que escribe el desarrollo de un producto basándose en investigación o aplicación. Representa la idea de interconectar todas las disciplinas involucradas en la tarea de ingenieros; electrónica, eléctrica, mecatrónica y sistemas.

El modelo V fue introducido para integración de disciplinas, como se ve en la Figura 12. Consta con tres capítulos. El hilo central, anaranjado, describe las principales actividades. La línea interior, amarilla, describe el trabajo con requisitos. Por último, la

línea azul exterior representa las actividades de análisis. Esto se subraya mediante el fondo grafico coherente en cada color de hilo, ya que la coordinación y cooperación entre disciplinas es un factor realmente decisivo para el proyecto. Las disciplinas más asociadas para el modelo V son la mecatrónica clásica, electrónica, electricidad y software, de esta manera, el V- Model se puede aplicar tanto en proyectos científicos e investigación, gestionados de forma clásica y ágiles.

Figura 12

Modelo VDI 2206



Nota: Adaptado del nuevo modelo VDI 2206, de Graessler y Hentze, 2020 (<https://doi.org/10.1515/auto-2020-0015>).

3.2.1 Elicitación de requisitos

Para la obtención, estructuración y análisis de requisitos según requerido que permita cumplir el objetivo de la investigación se basó en la directriz VDI 2221, donde conceptualiza la descripción del producto mediante criterios y agrupación de funciones. (Jänsch & Birkhofer, 2006)

Tabla 3

Lista de Exigencias

Deseo Exigencia	o	Descripción
E		Función principal: Monitoreo remoto de la sala de control de presión constante, utilizando tecnologías 4.0
E		Recursos: Se trabajará con equipos accesibles
E		Control: La adquisición de datos y envío será por medio de un controlador PLC y un comunicador industrial.
E		Seguridad: Los paquetes de envío de datos (Formato JSON) deben estar protegidos para su almacenamiento en la nube
E		Mantenimiento: Los elementos de visualización son compresibles para el usuario y posteriores cambios.
D		Implementación: El diseño debe contar con los presupuestos y la aprobación de un laboratorio.
D		Energía: Para garantizar el buen funcionamiento deberá contar con tableros industrias según el diseño.
D		Transporte: Se optará en un diseño para un laboratorio, los equipos de comunicación deben ser transportables.
E		Ergonomía: El diseño de estar incluido en los planos eléctricos según las normas de una industria.

Nota. Adaptado de la directriz VDI 2221

3.2.1.1 Abstracción de caja negra

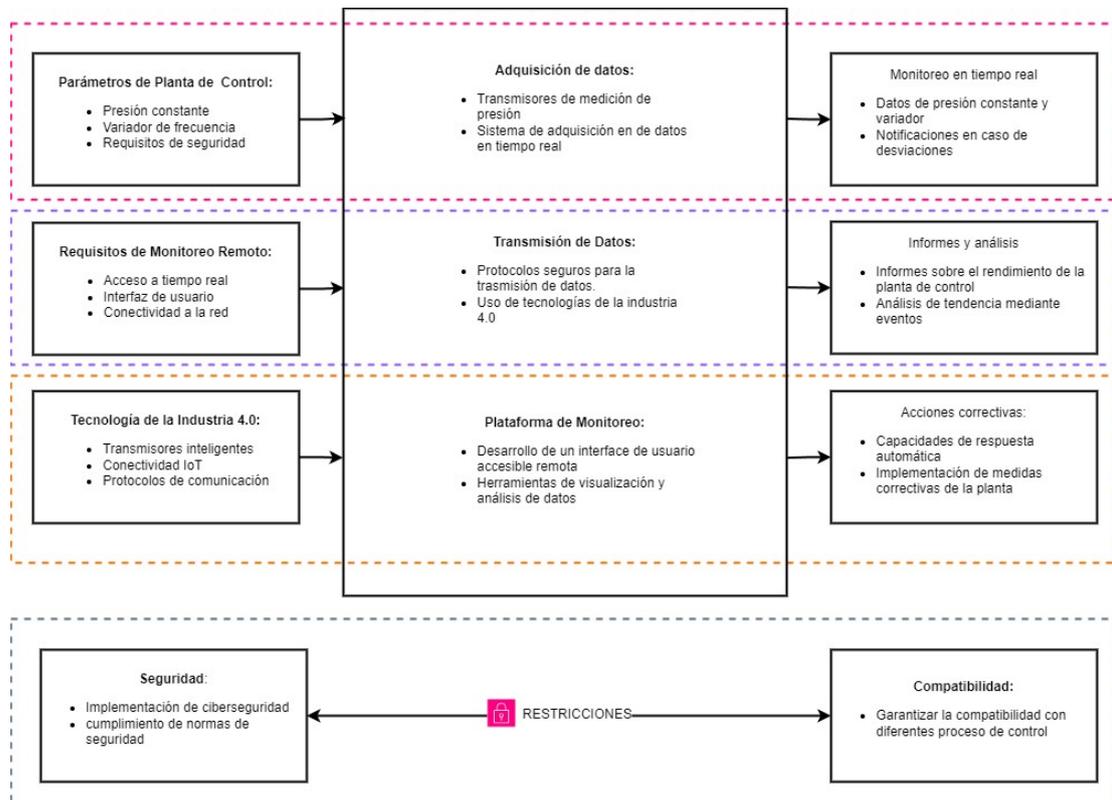
La caja negra en VDI 2221 es una herramienta de diseño utilizada en la ingeniería. En esta investigación la caja describe y analiza el sistema funcional dividida en tres secciones y las restricciones del proyecto.

- **Entradas:** se considera los parámetros de proceso de control, los requisitos del sistema de monitoreo y la integración de las tecnologías de la industria 4.0.

- **Procesos:** se considera la adquisición transmisión de datos mediante una plataforma de monitoreo en tiempo real.
- **Salidas:** se considera monitoreo en tiempo real, reportes y acciones correctivas mediante datos.
- **Restricciones:** se considera la seguridad del sistema y la compatibilidad con los demás procesos de control.

Figura 13

Caja negra del sistema



Nota: La caja negra se centra en las funciones del sistema sin abordar a detalle. Adaptado de VDI 2221

3.2.1.2 Agrupación de funciones

A partir de la lista de exigencias y deseos, se puede agrupar las funciones en un diagrama de bloques, tal como se muestra en la Figura 14. Esta sección

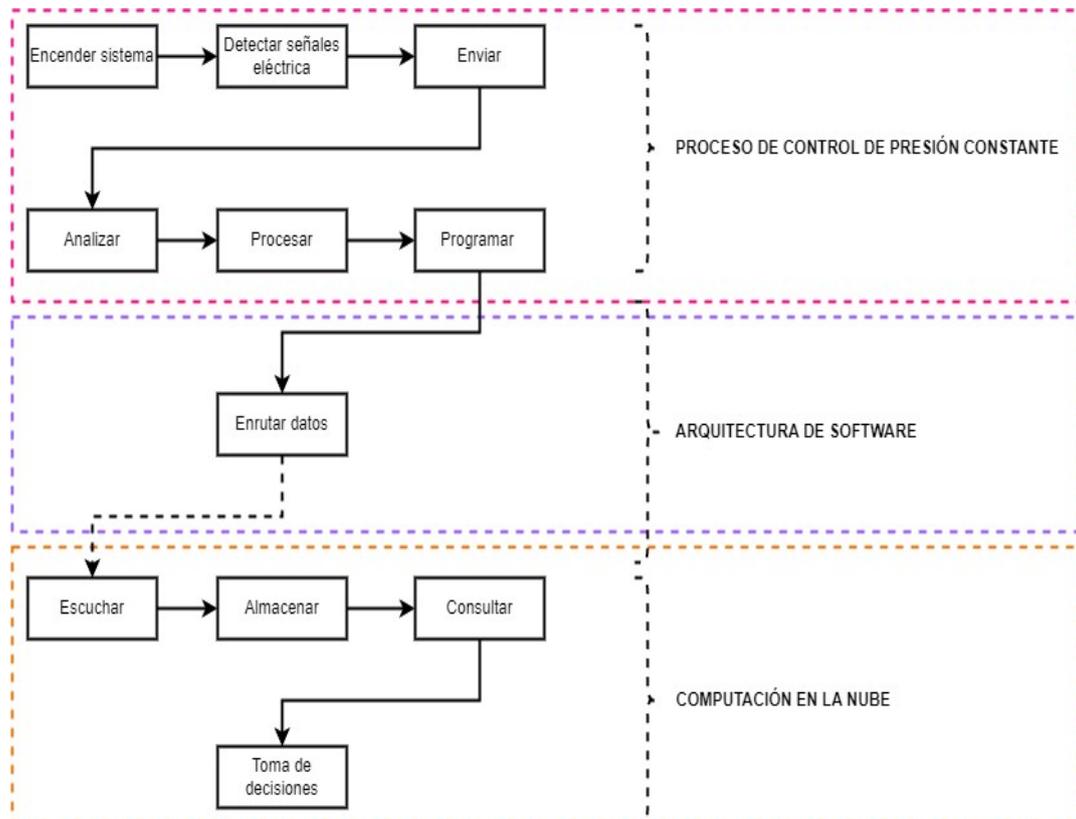


permite entender el diseño de un sistema del funcionamiento que tendrá en esta investigación.

- **Proceso de control de presión constante:** la primera acción del proceso es encender el sistema, el transmisor de presión detecta la señal para enviar a un controlador mediante las entradas analógicas. El controlador mediante un software analiza y procesa la señal emitida por el transmisor.
- **Arquitectura de software:** es el principal medio de enlace entre la nube y el proceso de control. La arquitectura adquiere los datos registrados del transmisor de presión en el software del controlador. También se debe tener los certificados de accesos a la nube mediante un protocolo de comunicación.
- **Computación en la nube:** registrarse en servicios de nube, para generar los accesos y las reglas de IoT, esto permitirá escuchar los datos emitidos por la Arquitectura de software. Luego ejecutar la ingesta en una base de datos en series temporales. Finalmente, hacer las consultas de datos en Grafana que permitirá el monitoreo en tiempo real y la toma de decisiones.

Figura 14

Agrupación de funciones



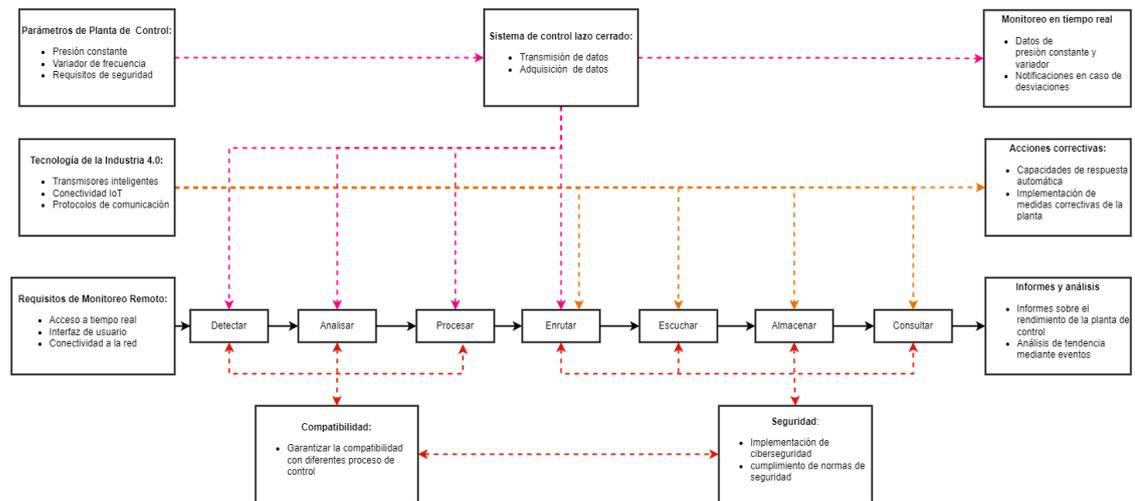
Nota: Las funciones son las acciones principales en cada proceso.

3.2.1.3 Estructura de funciones

La estructura de funciones de esta investigación, está seccionado en tres niveles; el proceso de control de presión constante, detectar, analizar; arquitectura de software, programar y enrutar; computación en la nube, escuchar, almacenar y mostrar. La matriz morfológica permite visualizar las posibles soluciones del diseño, para luego analizar con los criterios y aspectos posibles técnicos – económicos según la norma VDI 2225.

Figura 15

Estructura integral de las funciones



Nota: las restricciones mencionadas en la caja negra, se integra como proceso en la estructura integral, siendo fundamenta en la investigación. Adaptado del VDI 2221

Tabla 4

Aspectos Posibles de Evaluación

Aspectos técnicos	Aspectos Económicos
Uso de energía	Facilidad de adquisición de materiales
Seguridad	Costo de tecnología
Rapidez	Productividad
Estabilidad	Facilidad de mantenimiento
Manipulación	Costo de operación
Portabilidad	
Resistencia a interface extrema	

Nota: Los aspectos técnicos y económicos pueden variar según la investigación.

Los aspectos técnicos y económicos deben ser analizado y calificados críticamente para establecer las posibles soluciones, tal como se muestra en la Tabla 5. los criterios se evalúan no satisface (0) hasta el puntaje idea (4), se debe

tener en cuenta que si en un aspecto técnico y/o económico tiene un puntaje 0; la soluciones se considera no viable.

Tabla 5

Criterios de calificación

Puntaje	Apreciación
0	No satisface
1	Aceptable
2	Suficiente
3	Bien
4	Solución ideal

Nota: Adaptado de la directriz 2225.

3.2.2 Análisis morfológico

Es un enfoque sistemático para obtener soluciones y posibles respuestas de problemas complejos. En ingeniería, se aplica para abordar problemas multidisciplinarios y facilitar las respuestas para la toma de decisiones. Zwicky (1969) desarrolló la "Morfología General", este método descompone un problema complejo en segmentos pequeños para analizar las interacciones y los segmentos para encontrar la solución ideal. Para evaluar los conceptos de las posibles soluciones se analiza la matriz morfológica, donde permite visualizar dispositivos electrónicos, softwares y los servicios en la nube. Además, permite construir dos o más posibles soluciones aplicables en diferentes contextos de la ingeniería. Para hacer el análisis morfológico, se debe hacer investigaciones exhaustivas de elementos que compondrá las posibles soluciones y se debe considera los siguientes criterios:



- **Procesos:** son las divisiones de la investigación, debido a que cada proceso es diferente en esta investigación.
- **Funciones:** son acciones del sistema integrado que se obtiene mediante la caja negra. Las funciones determinan las acciones que cumplirla los elementos de la investigación.
- **Soluciones:** son el estudio de elementos que ayudarán a generar soluciones, esto es posible a la integración de las funciones en cada proceso que se divide la investigación.

3.2.2.1 Matriz morfológica del sistema integrado

La matriz morfológica del sistema del monitoreo remoto de una sala de control de presión constante que usa las tecnologías de la industria 4.0; está seccionada en tres procesos, control de presión constante, arquitectura de software y computación en la nube, tal como se muestra en la Tabla 6.

- **Control de presión constante:** Este proceso conforma los equipos electrónicos, como sensores, transmisores (digitales y analógicos), y transmisores Modbus. La función principal es detectar señales eléctricas, luego enviar a los controladores, y permita analizar los valores de cada transmisor para procesarlos.
- **Arquitectura de software:** Es el proceso de la adquisición y procesamientos de datos en un código script. Este proceso generará los mensajes en formato JSON. El desarrollado back-end se llevará en un software de programación libre.

- **Servicios en la nube:** Conforman los servicios en la nube como Azure, AWS, GCP y IBM. Cabe resaltar que existe más servicios. Esto dependerá del desarrollador y el proyecto que se desarrollará.

Tabla 6

Matriz morfológica del sistema integrado

Proceso	Función	Solución 01	Solución 02	Solución 03	Solución 04
Control de presión constante	Detectar	Sensor Analógico	Sensor digital	Transmisor	Transmisor Modbus
	Analizar	Raspberry PI	Beaglebone	PLC	Laptop
Arquitectura de software	Programar	Python	C++	Ladder	Matlab
	Enviar	HTTP HTTPS	MQTT	CoAP Constrained Application Protocol	SFTP
Computación en la nube	Escuchar	IoT Core	Azure IoT	GCP IoT Core	IBM IoT
	Almacenar	Timestream	Cosmos DB	FireStore	TimescaleDB



Nota: La matriz morfología plantea proporciona tres posibles soluciones viables, sin embargo, podría generar más soluciones. Adaptado de la directriz VDI 2225

Tabla 7

Evaluación de la matriz morfológica del sistema integrado

Nro.	Aspectos técnicos y económicos	Solución 01	Solución 02	Solución 03	Solución ideal
1	Función	3	3	4	4
2	Precisión	3	3	4	4
3	Velocidad	3	3	4	4
4	Productividad	3	3	4	4
5	Confiabilidad	3	3	4	4
6	Portabilidad	4	4	2	4
7	Robustez	1	1	4	4
8	Seguridad	2	2	4	4
9	Ergonomía	4	1	4	4
10	Fabricación	4	3	2	4
11	Montaje	4	4	3	4
12	Mantenimiento	4	4	3	4
13	Costo de tecnología	4	4	3	4
14	Costo de operación	4	4	2	4
Suma total		46	42	47	56

Nota. Adaptado de la directriz VDI 2225

Según la evaluación de la matriz morfológica, la solución 3 es la más óptima para la investigación.

3.2.2.2 Matriz morfológica del sistema eléctrico

Tabla 8

Matriz morfológica del sistema eléctrico

Proceso	Función	Solución 01	Solución 02	Solución 03	Solución 04
Energizar					
		Alimentación 5v	Alimentación 24V	Fuente Regulable	
Detectar					
		Sensor Analógico	Sensor digital	Transmisor	Transmisor Modbus
Analizar					
		Raspberry Pi	Beaglebone	PLC	Laptop
Comunicar					
		Usb – modbus	Usb-DB9	Tcp/ip	Usb – hmi
Enrutar					
		RUT950	RUT240	TRB245	Router ur35
IIoT					
		Android/IOS	HMI	Monitor	Rugged Dell

Nota: La matriz morfológica plantea proporciona tres posibles soluciones viables, sin embargo, podrá generar más soluciones. Adaptado de la directriz VDI 2225

La matriz morfológica del sistema eléctrico se divide en:

- **Instrumentación de presión constante:** Es el proceso de la energización de equipos electrónicos, para la funcionabilidad de los controladores y transmisores.
- **Comunicación industrial:** Son las conexiones (conectores y enrutadores) de protocolos de comunicación y envío de datos a la nube.
- **IIoT (Industria Internet de las cosas):** Sistema de monitoreo de la planta de control de presión constante a través de los datos adquiridos.

Tabla 9

Evaluación de la matriz morfológica del sistema eléctrico

Nro.	Aspectos técnicos y Económicos	Solución 01	Solución 02	Solución 03	Solución ideal
1	Función	3	3	4	4
2	Precisión	3	3	4	4
3	Velocidad	3	3	4	4
4	Productividad	2	2	4	4
5	Confiabilidad	3	3	4	4
6	Portabilidad	4	4	2	4
7	Robustez	2	2	4	4
8	Seguridad	3	2	4	4
9	Ergonomía	4	2	4	4
10	Fabricación	4	4	3	4

11	Montaje	4	4	3	4
12	Mantenimiento	4	4	4	4
13	Costo de tecnología	3	3	2	4
14	Costo de operación	4	4	2	4
Suma total		46	43	48	56

Nota: Adaptado de la directriz VDI 2225

Según la evaluación de la matriz morfológica, la solución 3 es la más óptima para la investigación.

3.2.3 Evaluación técnica – económica

En esta sección se evalúa tres conceptos de solución, formuladas en las anteriores evaluaciones, según la norma de VDI 2225. Cada concepto de solución se puntuará de 0 a 4 por criterio. Esto permite elegir la solución más óptima posible. Se elaboró la siguiente tabla de muestra los ponderados de criterio.

Tabla 10

Ponderado de criterios

Ponderado: g	Valor
4	Muy importante
3	Importante
2	Considerable
1	Menos considerable

Nota: Adaptado de la directriz VDI 2225

3.2.3.1 Criterios técnicos

Los criterios técnicos para la evaluación de los tres conceptos de solución son los siguientes:

- **Seguridad:** Se considera muy importante (4) y factor importante para esta investigación.
- **Monitoreo remoto:** Se muy importante (4), para el proceso de datos.
- **Automatización:** Se considera importante (3), para un proceso inteligente.
- **Manipulación:** Es considerable (2), para la toma de decisiones y el mantenimiento.
- **Complejidad de implementación:** se considera importante (2) para las plantas de control muy aparte de la planta de presión constante.

Tabla 11

Evaluación técnica

Criterios técnicos		Solución 1		Solución 2		Solución 3		Solución ideal		
No	Criterio	g	P	$g \times p$	p	$g \times p$	p	$g \times p$	p	$g \times p$
1	Seguridad	4	2	8	2	8	3	12	4	16
2	Monitoreo remoto	4	3	12	3	12	3	12	4	16
3	Automatización	3	2	6	2	6	2	6	4	12
4	Manipulación	2	3	6	2	4	3	6	4	8

5	Complejidad de implementación	2	3	6	3	6	2	4	4	8
	Suma		13	38	12	36	13	40	20	60
	Suma relativa		0.65	0.63	0.6	0.6	0.65	0.66	1.00	1.00

Nota: p: Puntaje de 0 a 4 donde 0: no satisface, 1: menos aceptable, 2: suficiente, 3: bien, 4: muy bien. g: peso ponderado según la importancia del criterio. Adaptado de la directriz 2225

3.2.3.2 Criterios económicos

Los criterios técnicos para la evaluación de los tres conceptos de solución son los siguientes:

- **Costo de tecnología:** Se considera muy importante (4) para el monitoreo y envío de datos.
- **Costo de mantenimiento:** Se considera muy importante (4) debido a que los equipos siempre deben estar óptimos.
- **Costo de equipos:** Es considerable (3), para la instrumentación, los materiales son accesibles en el mercado nacional.
- **Consumo de energía:** Se considera importante (3), hay equipos que se adaptan al bajo consumo de energía.

Tabla 12

Evaluación económica

Criterios técnicos		Solución 1		Solución 2		Solución 3		Solución ideal		
No	Criterio	g	p	$g \times p$	p	$g \times p$	p	$g \times p$	p	$g \times p$
1	Costo de tecnología	4	3	12	2	8	2	8	4	16

2	Costo de mantenimiento	4	2	8	2	8	4	16	4	16
3	Costo de equipos	3	3	9	3	9	2	6	4	12
4	Consumo de energía	3	3	9	3	9	3	9	4	12
	Suma		11	38	10	34	12	39	16	56
	Suma relativa		0.68	0.67	0.62	0.60	0.75	0.69	1.00	1.00

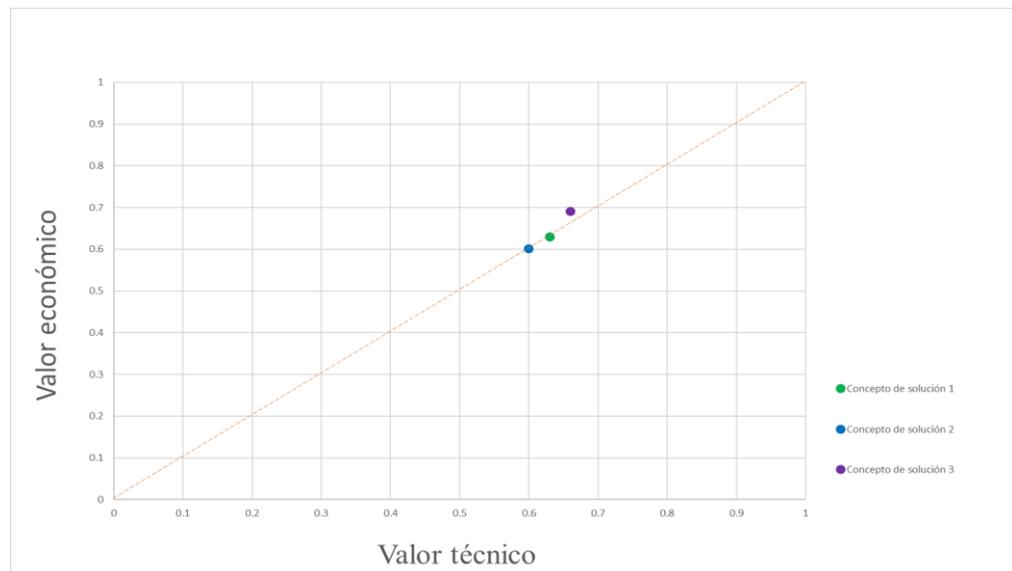
Nota: p: Puntaje de 0 a 4 donde 0: no satisface, 1: menos aceptable, 2: suficiente, 3: bien, 4: muy bien. g: peso ponderado según la importancia del criterio. Adaptado la directriz VDI 2225.

3.2.4 Concepto de solución óptimo

Mediante los criterios económicos y técnicos de la Tabla 11 y Tabla 12 se construye la gráfica de comparación de los conceptos de solución, la cual se muestra en la Figura 16.

Figura 16

Gráfico de comparación valor Técnico - Económico



Nota: Adaptado de la directriz VDI 2221

A partir de la gráfica se concluye que la solución 3 es la más óptima, debido a se acerca a la solución idea que es 1. Sin embargo, se puede considera la solución 2, por su cercanía a la solución óptima.

Tabla 13

Criterios de solución óptimo

Valor técnico	Valor económico	Calificación
0.8	0.8	Muy buena solución
0.7	0.7	Buena solución
0.6 o menos	0.6 o menos	Solución deficiente

Nota: Fuente: Según VDI 2225

3.2.5 Arquitectura y diseño del sistema

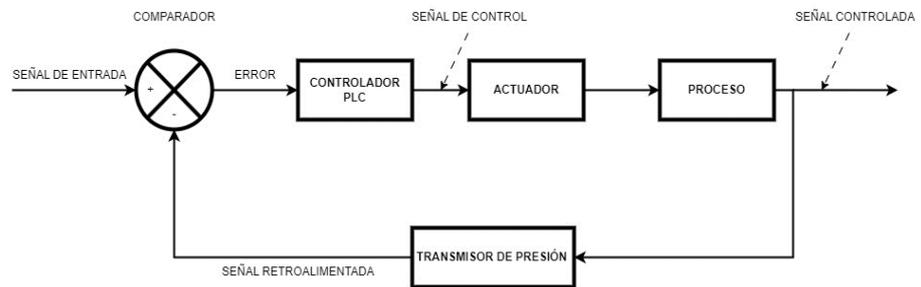
En la arquitectura del sistema se desarrolla la estructura de solución general interdisciplinaria del sistema. La arquitectura del sistema genera las alternativas estructurales, se comparan sobre la base de criterios comprensibles, como la realización de interdependencias de cada proceso, análisis funcionales del diseño y la consideración de criterios de exclusión. (Graessler & Hentze, 2020)

3.2.5.1 Arquitectura del proceso de control

A partir de la matriz morfológica del sistema integrado y la matriz del sistema eléctrico, se obtiene la distribución del sistema de control. Para ello se propone que el sistema de control de presión constante de lazo cerrado (control PID)

Figura 17

Diagrama de lazo cerrado de presión constante

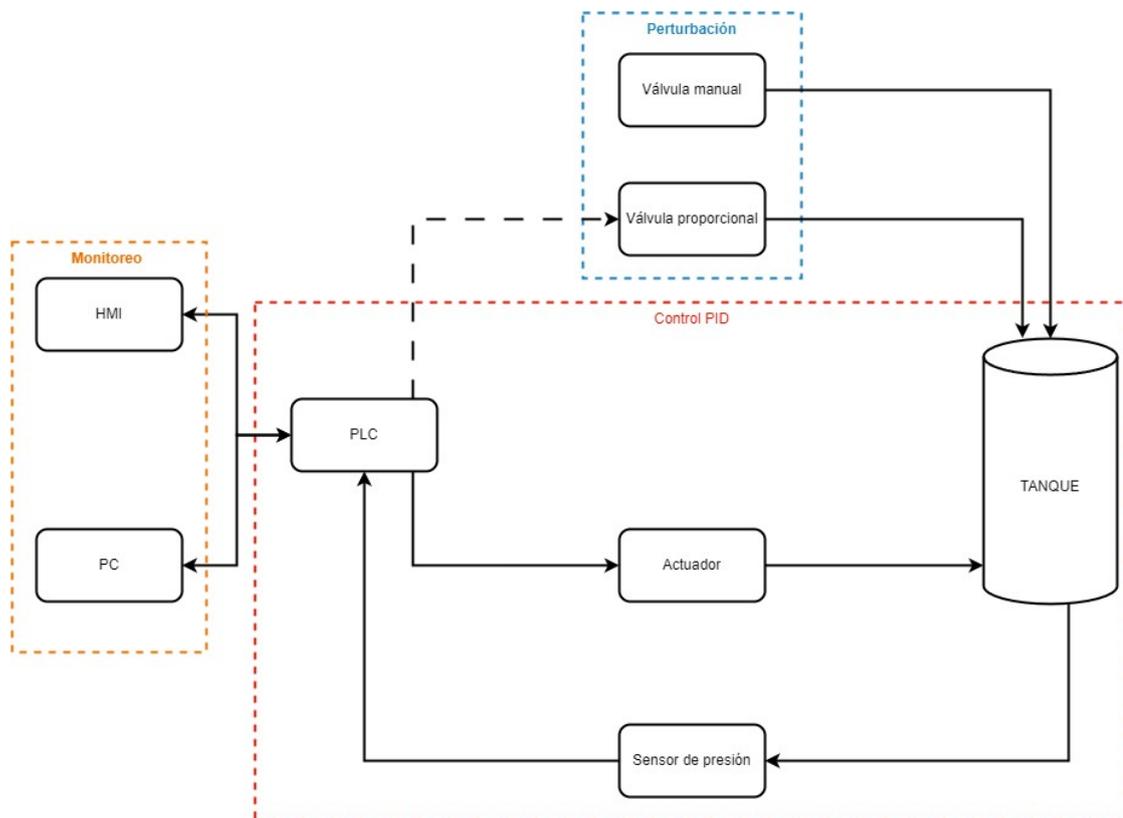


Nota: Está basado en el sistema de control de presión constante, el sistema de control es un PDI.

El sistema de control emplea algoritmos PDI, de manera que la computadora y el HMI (sección monitoreo) está relacionado para supervisar y configurar parámetro de control. El PLC es el encargado de accionar al actuador según el grado de los parámetros PID. El motor trifásico representa el mando de la potencia y es controlado mediante el variador, con una señal resultante del PDI enviada por el PLC, donde la variable (presión) es modificada según el accionar del actuador. La válvula proporcional ayuda a presurizar al sistema, también las perturbaciones en el taque son generadas por las válvulas proporcional y manual.

Figura 18

Esquema general de lazo cerrado



Nota: Esquema de lazo cerrado en un sistema de control de presión. Adaptado de *Manual de Usuario Planta de Control de Presión*, por INDUCONTROL S.A.C, 2020

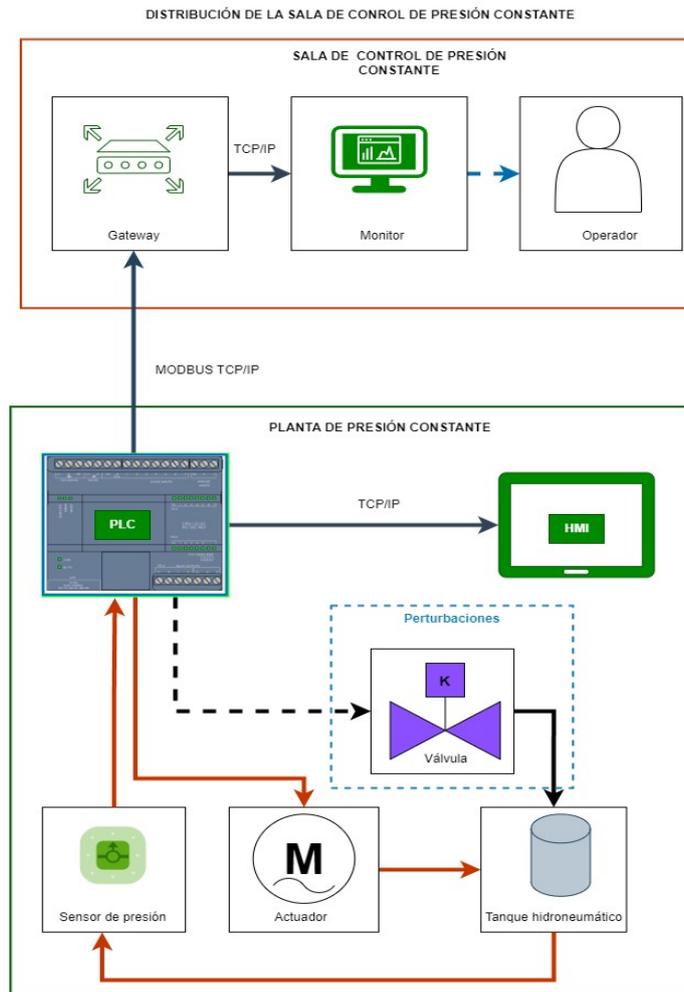
En esta investigación propone un sistema de distribución de control. El proceso de control llevara las acciones organizadas con el fin de mejorar la producción, administrar el sistema y tomar decisiones con respecto a la planta. Este sistema se compone con los siguientes elementos; los sensores, actuadores y el controlador para que cumpla una función única o dos, es necesario proponer la distribución del proceso de control tal como se puede observar en la Figura 19.

- **Sala de control de presión constante:** Es el uso del operario con la planta de control mediante un monitor enlazado con un Gateway.

- **Planta de presión constante:** En esta sección están los elementos del proceso de control, como el sensor de presión constante, los actuadores, el controlador y el HMI.

Figura 19

Distribución de la sala de control de presión constante



Nota: Este proceso permite controlar las demandas de agua de menor escala, estableciendo flujo constante.

3.2.5.2 Arquitectura de software

La arquitectura está basada en operación back-end, esto permite leer registros inscritos en la memoria del controlador del proceso de control de presión constante. También se encarga de leer y procesar. Finalmente, la operación



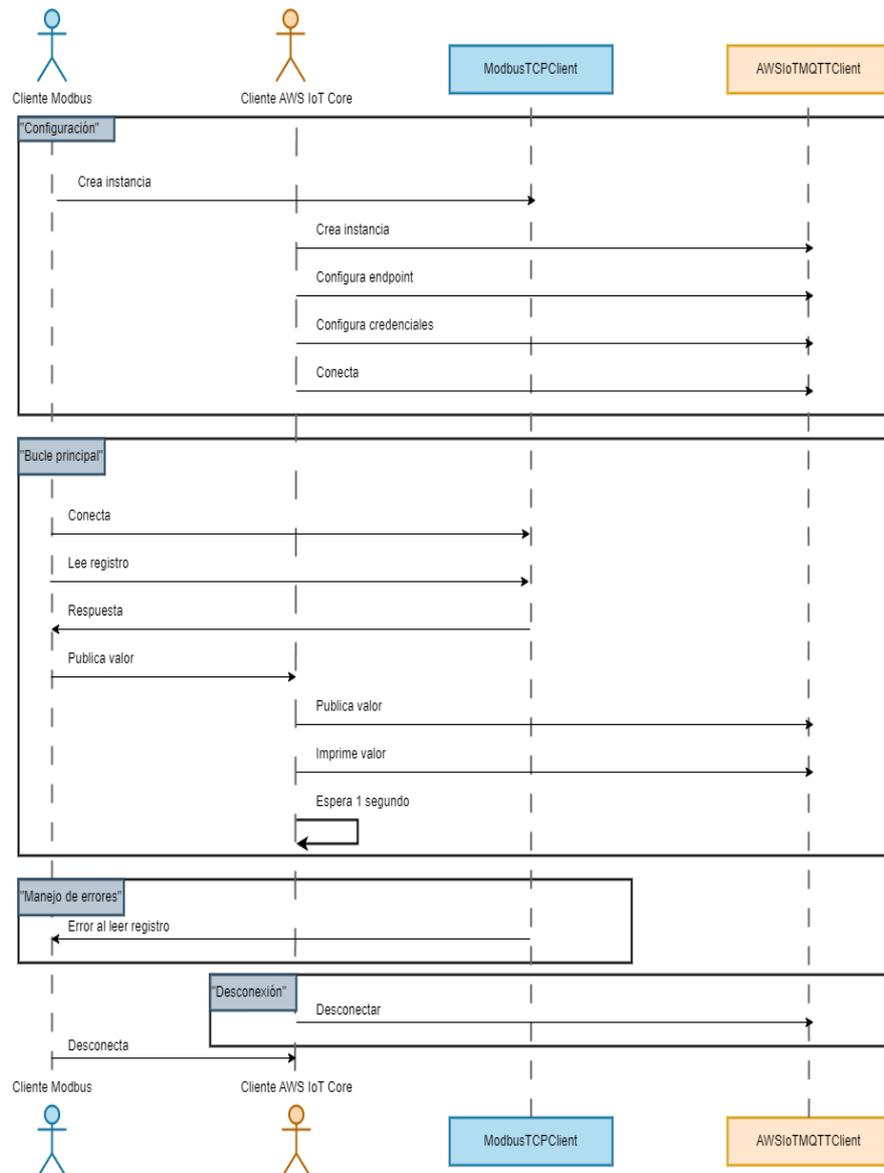
permite la comunicación directa con hardware y generar datos empaquetados (Payload) para el envío a la nube.

La operación back-end tendrá un diagrama de secuencia (representación gráfica de interacción de elementos de un sistema a lo largo del tiempo), donde se representa la interacción entre Modbus y AWS IoT Core, tal como se puede observar en la Figura 20.

- **Configuración:** En esta parte, ambos actores crean instancias de sus respectivos clientes. Se configura las direcciones y las credenciales de AWS IoT Client.
- **Bucle principal:** esta es la función principal del programa. El cliente Modbus se conecta a un servicio en la nube, luego solicita registros al PLC, y enviarlos en formatos JSON, y con un retraso de 1 minuto.
- **Manejo de errores:** Si dentro de la función existe error mientras se solicita la petición (registro), El cliente Modbus envía error al actor del cliente Modbus.
- **Desconexión:** En el final del programa, ambos actores se desconectan, tanto como el cliente Modbus y AWS IoT Core.

Figura 20

Diagrama de secuencia



Nota: Desarrollado en software de dibujo grafico Draw.io.

Se puede establecer el diagrama de flujo. Considerando las cuatro partes del diagrama de secuencia.

- **Configuración:** Donde, comienza con la configuración del PLC Modbus y de AWS IoT Core, seguido de la creación de las instancias de los clientes Modbus TCP y AWS IoT Core. luego, se establece la comunicación con el cliente AWS IoT Core.

- **Bucle principal:** En esta sección, se establece la conexión con el cliente Modbus, se solicita los registros especificados y se verifica si la respuesta es válida. Si la respuesta es válida, se obtiene el valor del registro, se crea el payload para formato JSON, luego la publicación.
- **Manejo de error:** Si la respuesta no es válida, se imprime un mensaje de error. También se imprime la acción de parar la operación.
- **Desconexión:** Después de cada 1 segundo de interacción, se cierra la conexión con el PLC Modbus y se espera la siguiente lectura. Al final del programa, se desconecta el cliente AWS IoT Core.

El diagrama de flujo contempla dos configuraciones importantes tal como se muestra en la siguiente Tabla 14.

Tabla 14

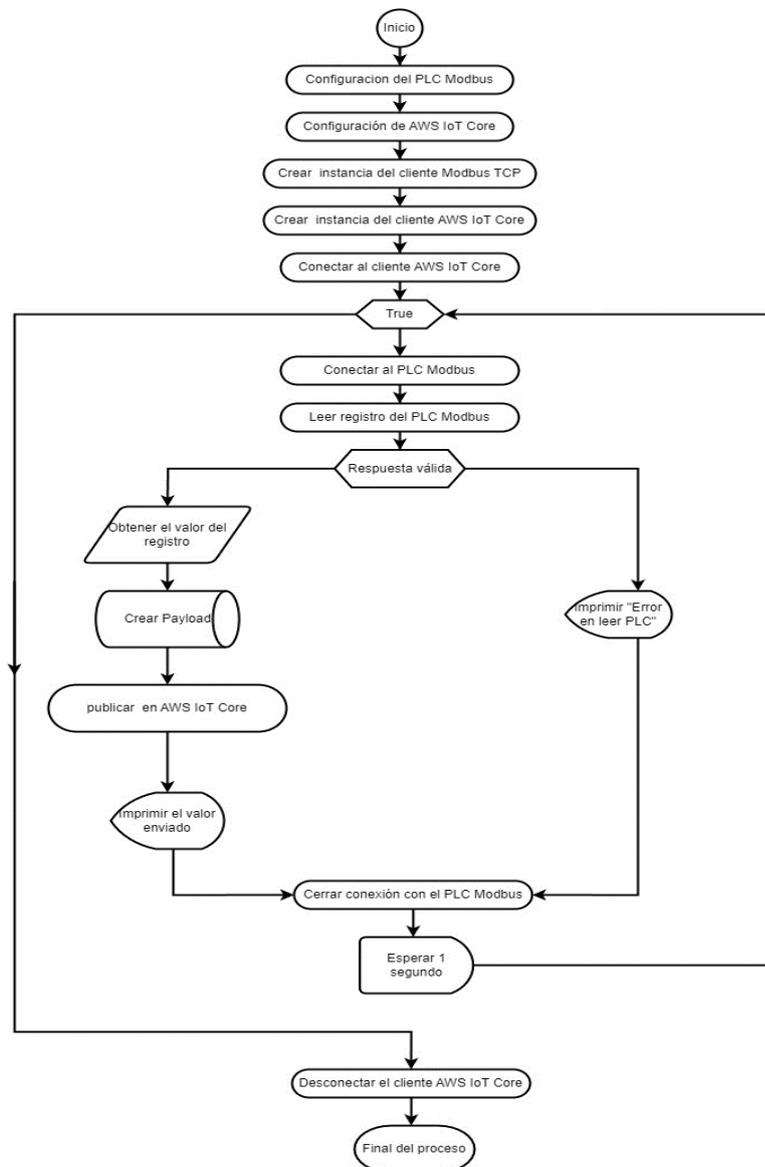
Configuración del diagrama de flujo

Configuración	Descripción	Acción
Modbus	Establecer parámetros de conexión del controlador	Configurar parámetros Modbus; IP, puerto, dirección del dispositivo, registros a leer.
AWS IoT Core	Establecer credenciales y configuraciones necesarias para conectarse a AWS IoT Core.	Configura credenciales y rutas de certificados del dispositivo y endpoint de AWS IoT Core.

Nota: Esta configuración se puede adaptar en diferentes lenguajes de programación.

Figura 21

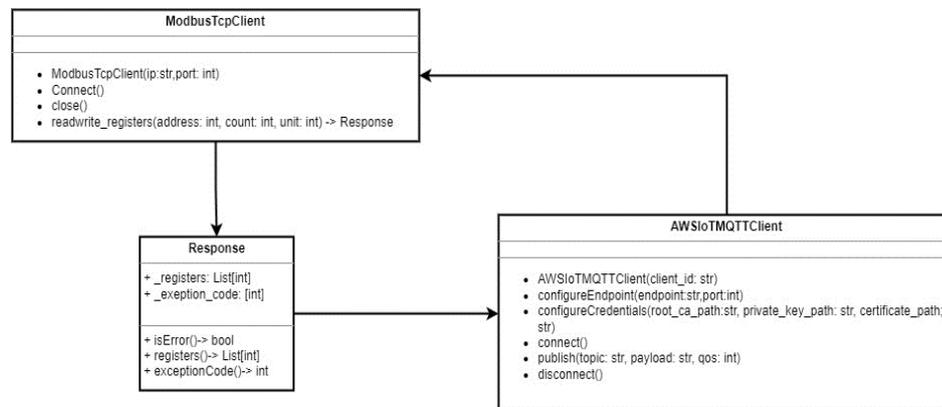
Diagrama de flujo



Nota: Desarrollado en software de dibujo grafico Draw.io

Figura 22

Arquitectura del código



Nota: Desarrollado en software de dibujo gráfico Draw.io y código UML.

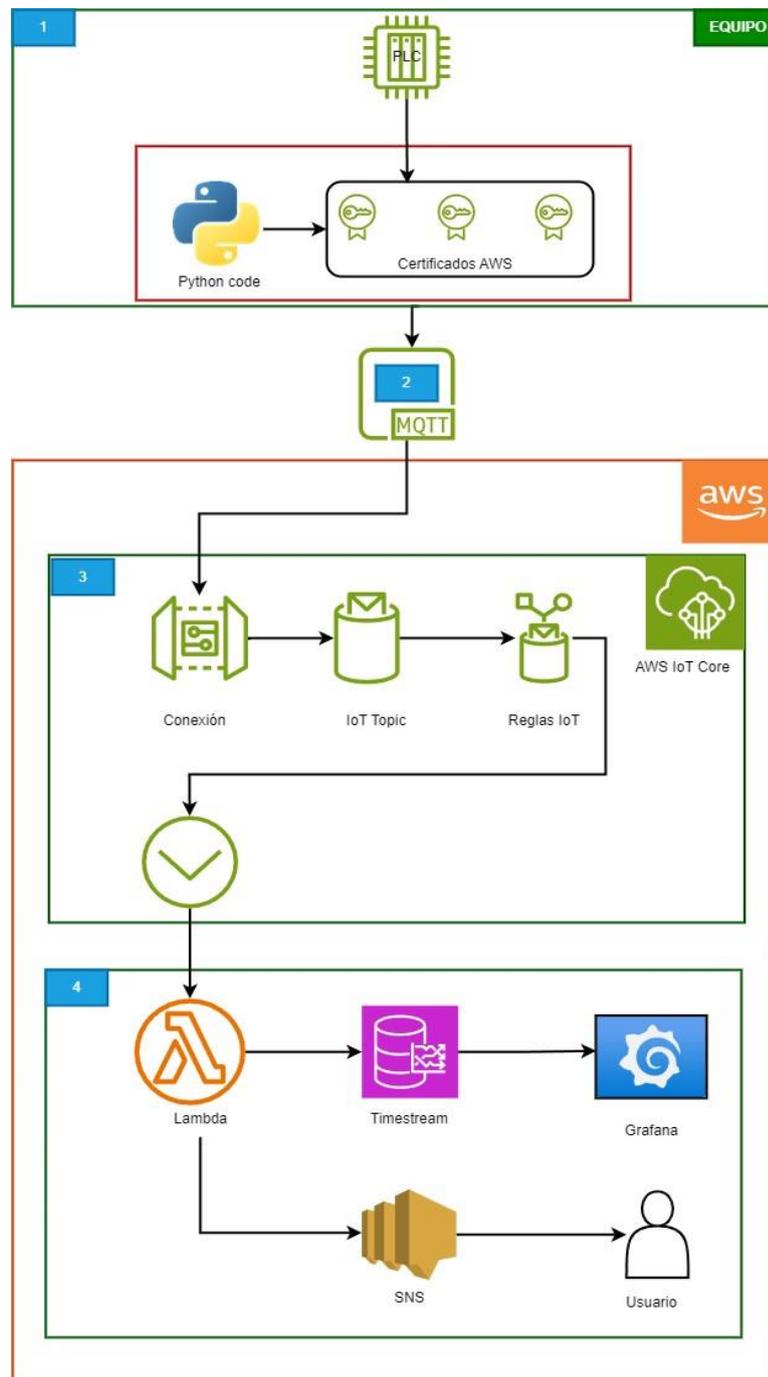
3.2.5.3 Arquitectura de AWS IoT Core

La arquitectura para AWS IoT Core, está elaborada en cuatro pasos:

1. **Equipos:** para la conexión de equipos con los servicios de la nube se deben tener certificados, que son generados por la nube. Estos certificados sirven como enlace y verificación entre el equipo y la nube.
2. **Comunicación MQTT:** Es enlace entre el equipo y AWS, conjuntamente con el envío de los mensajes en formato JSON.
3. **AWS IoT Core:** Es un servicio de AWS que permite escuchar los datos mediante un tópicos que es declarado en el equipo (en la operación back-end). Y esto reconoce los certificados ya generados.
4. **Estructura de base de datos:** la función lambda permite configurar, si es que los datos requieren. Luego, se almacena en series temporales para la ingesta de datos en Grafana.

Figura 23

Arquitectura AWS IoT Core



Nota: Desarrollado en software de dibujo grafico Draw.io

3.3 INGENIERÍA DE REQUISITOS

La ingeniería de requisitos describe el manejo y estructuración de recursos, asignación de presupuesto, análisis e integración de elementos que se debe de gestionarse

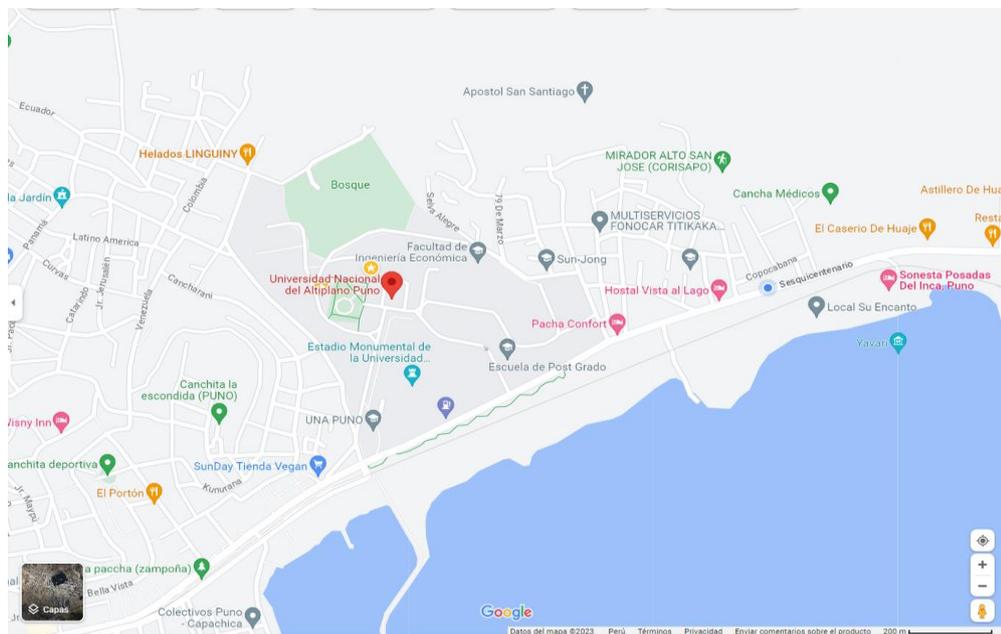
para el proyecto en desarrollo. Para enfatizar la importancia de la obtención como base para un proyecto exitoso, también se maneja el termino de gestión de requisitos en el Modelo V. El manejo de los requisitos está respaldado por diversas herramientas, bases de datos y métodos. La gestión de requisitos es relevante tanto en el nivel superior del sistema como en el nivel de los elementos del sistema (Graessler & Hentze, 2020).

3.3.1 Ubicación del proyecto

El estudio y la validación de la tesis se realizó en la Universidad Nacional del Altiplano de Puno (Figura 24), laboratorio de control II (Figura 25) de la Escuela Profesional de Ingeniería Electrónica.

Figura 24

Universidad Nacional del Altiplano



Nota: Generado por Google Map.

3.3.2 Planta de control de presión

La planta de presión constante cuenta con servicio de internet local mediante red LAN, es óptimo para validar la investigación, cuenta con los

correspondientes elementos del proceso de control, esto incluye los planos eléctricos y las ubicaciones físicas de cada elemento según el Anexo A.

El computador personal permitirá programar en TIA-Portal los registros Modbus TCP/IP, y activar el código principal en python3 a través de MobaxTerm (terminal) mediante el protocolo SSH. Para el control de remoto es necesario instalar VNC Viewer.

Figura 25

Laboratorio de la planta de presión constante



Nota: Fotografía del laboratorio de control II de la Escuela Profesional de Ingeniería Electrónica – UNA PUNO.

3.3.3 Enrutador Industrial TRB245

Los enrutadores industriales son elementos de sistema de embebidos, en esta tesis, proporcionarán la conexión y envío de datos a los servicios de la nube. Además, un enrutador industrial es programable en el formato JSON (Java Script

Object Notation) y ejecutable mediante la consola interna (sistema operativo Linux).

Figura 26

Enrutador TRB245 - Teltonika



Nota: Enrutador TRB245 (Ver ANEXO 3) dispositivo industrial, permite solicitar datos al PLC y formar los paquetes JSON para luego enviar a AWS mediante el protocolo y los certificados generados AWS IoT Core.

3.3.4 Raspberry PI 3 Plus

Según la matriz morfológica del sistema eléctrico (tabla19) y los conceptos de solución técnico – económico, el uso del Raspberry PI 3 (solución 1) aprovecha la red local. El dispositivo lee los registros del PLC mediante el protocolo Modbus TCP/IP con la operación back-end desarrollado en Python. Para el monitoreo y control del dispositivo se debe de activar los protocolos SSH y VNC.

Figura 27

Raspberry PI 3 Plus



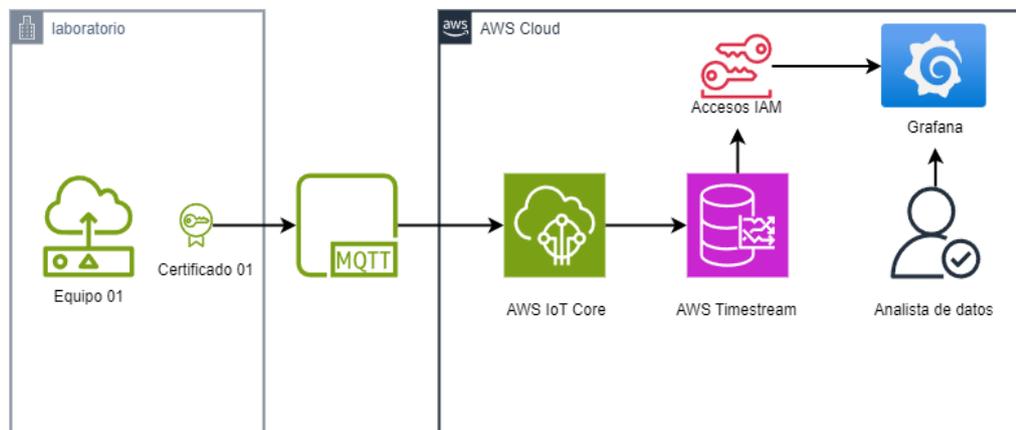
Nota. Consola Linux y python3.

3.3.5 Amazon Web Services

los servicios de AWS son fundamentales en esta investigación tal como se muestra en la arquitectura de solución (Figura 28), los servicios seleccionados permiten adquirir datos de equipos mediante el protocolo de comunicación MQTT. También permite la ingesta datos en series temporales y mediante los permisos de IAM se genera las consultas en otros servicios en la nube.

Figura 28

Arquitectura de solución IoT -AWS



Nota: El Proyecto se desarrollará en buenas prácticas de la computación en la nube.

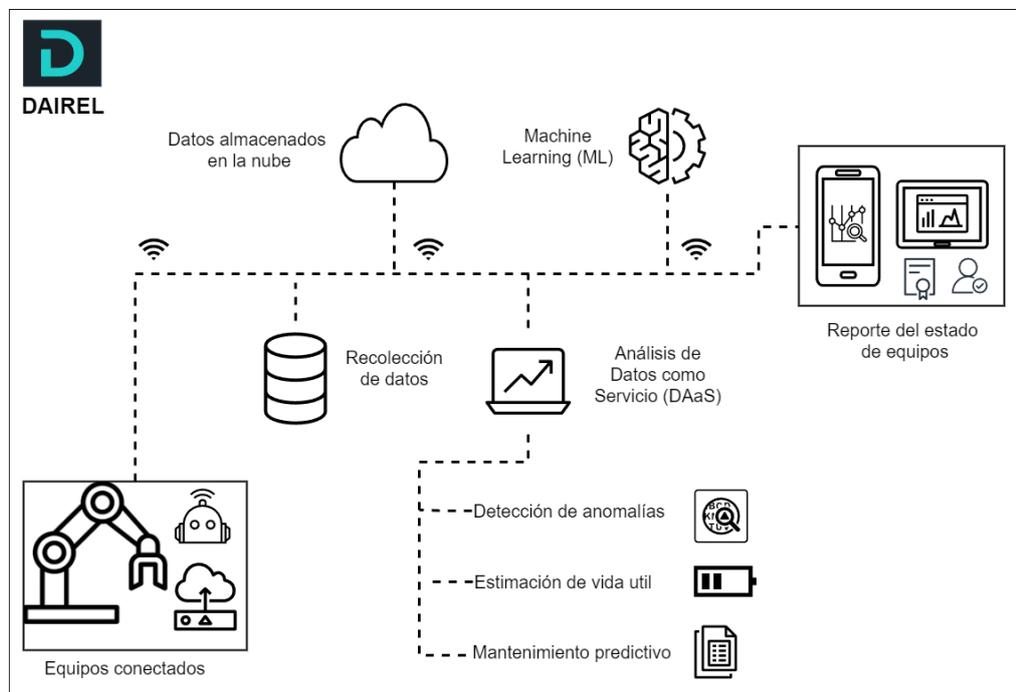
3.3.6 DAIREL

Es una plataforma elástica de IIoT y de DAaaS (*Data analytics as a service*) soportada en técnica de inteligencia artificial, especialmente diseñada y desarrollada para trabajar con datos de procesos industriales.

DAIREL permite conectar dispositivos mediante el internet, luego recolecta datos para ordenarlos, centralizarlos y almacenarlos en la nube. Mediante los datos se generan los Dashboards, donde se analiza los estados del equipo, la estimación de la vida útil, detección de anomalías y predice las fallas. Con estas informaciones se generan patrones que permite aplicar el mantenimiento predictivo en tiempo real.

Figura 29

Aplicación de DAIREL



Nota: Desarrollado por E2I



3.4 INTEGRACIÓN DEL SISTEMA

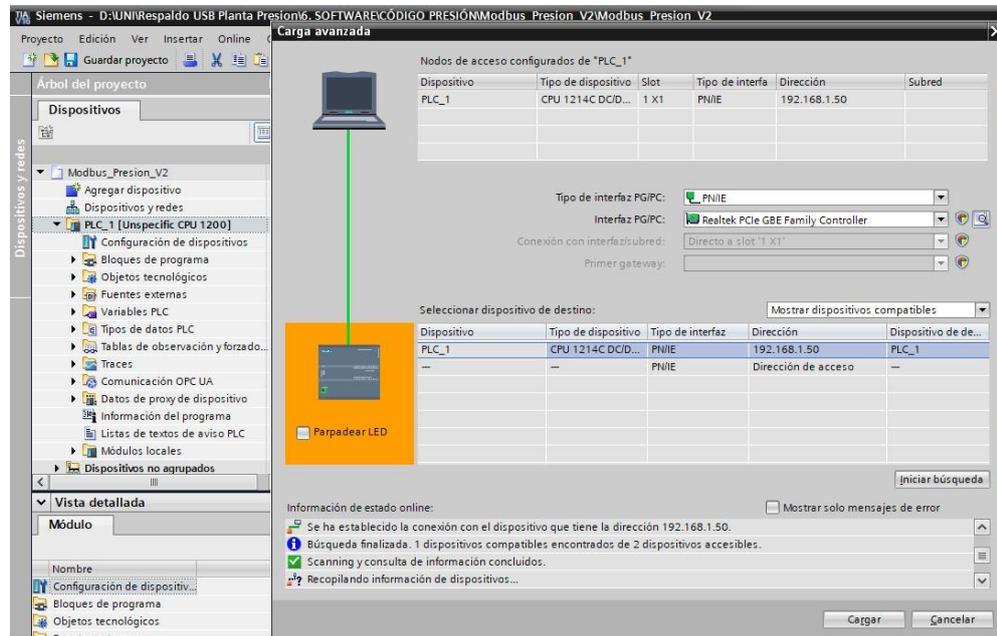
La principal ventaja del diseño propuesto, es aprovechar las tecnologías de un proceso de control para mejorar las tomas de soluciones, y accesible para el usuario (desarrolladores y/o encargados de laboratorios). En consecuencia, permite adquirir datos de un proceso de control, en este caso es la planta de precisión constante. Los datos son solicitados por un enrutador industrial de la planta de control y también mediante un minicomputador. Posteriormente, se envían mediante el protocolo de MQTT, conjuntamente con la integración de los certificados de AWS IoT Core. Además, los datos están en formato JSON, esto permite procesarlos en la nube. Un diseño que mejorará bastante a las investigaciones que se estudió antes del proyecto de tesis.

3.4.1 Almacenamiento de datos

El controlador (PLC) adquiere los datos del transmisor de presión constante por las entradas analógica (ver ficha técnica Anexo A), de las cuales debe procesar y declarar en una de las entradas analógicas o digitales; y conjuntamente con los valores emitidos por los actuadores; el objetivo es almacenar en base de datos Modbus. El primer paso es verificar el óptimo funcionando, y ver las características de cada componente que conforman el proceso de control. El controlador Siemens proporciona un software TIA Portal para ejecutar y programar, es necesario establecer conexión entre el controlador Siemens y el software a través del cable ethernet TCP/IP, que también servirá como comunicación Modbus, tal como se observa en la Figura 30.

Figura 30

Conexión TIA Portal - PLC SIEMENS

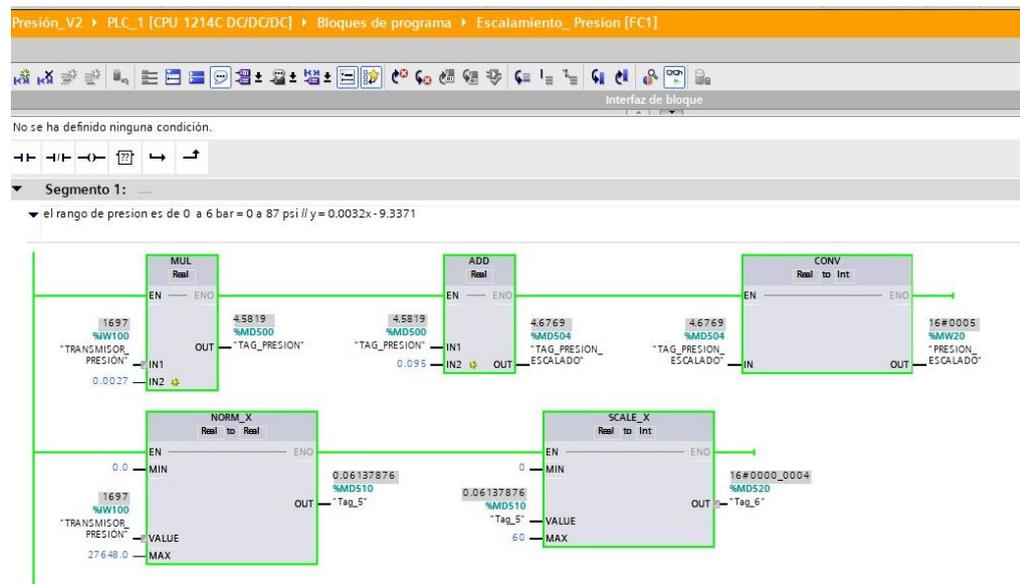


Nota. El software TIA Portal permite reconocer el controlador, la dirección IP y el cable de conexión. Por lo cual, es práctico descargar el programa ya guardar en la memoria del controlador.

En el proceso de la programación es necesario ubicar el transmisor de presión, y ver la configuración para direccionar los datos entrantes a un base de datos Modbus. El transmisor de presión es de tipo analógico y está declarado con el nombre de “TRANSMISOR_PRESIÓN” y como variable %W100. Después, sus valores analógicos son convertidos a valores enteros, esta acción se le nombra como “PRESIÓN_ESCALADO” y su variable es %MW20.

Figura 31

Presión escalado

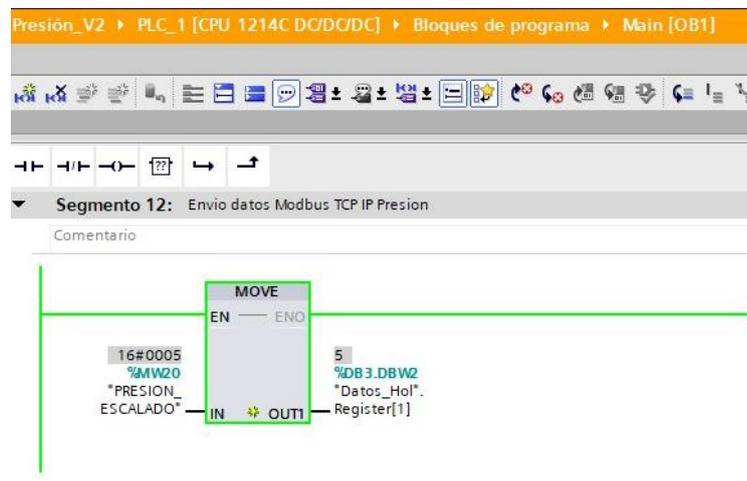


Nota: Programación en Ladder.

Los datos emitidos por el transmisor de presión, ya escalados, debe ser direccionados a una base de datos Modbus TCP/IP. Este segmento 12 direcciona los valores escalados del transmisor de presión de la entrada %MW20, la acción es mover los datos y almacenarlos en un Holding Register de Modbus.

Figura 32

Almacenamiento de datos en Hold Register

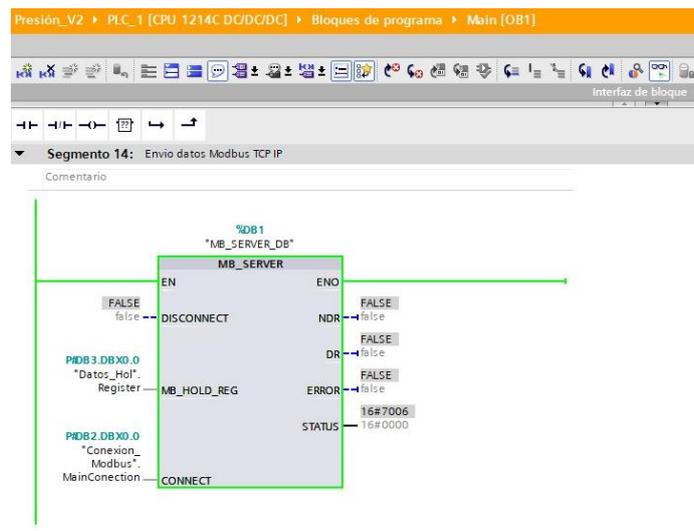


Nota: Programación en Ladder.

Para configurar el protocolo de comunicación Modbus, se agrega la comunicación Modbus TCP/IP server. Esto permite registrar los datos direccionados de la presión escalado. El “MB_SERVER” permite establecer conexión con otro dispositivo externo.

Figura 33

Modbus Server



Nota: Programación en Ladder.

Para establecer conexión con un dispositivo mediante Modbus se debe configurar la conexión “Conexión Modbus”. Estos parámetros pueden variar según el programa, es decir, en esta investigación tiene parámetros únicos como la dirección IP y en la dirección del esclavo.

Figura 34

Conexión Modbus

Presión_V2 ▶ PLC_1 [CPU 1214C DC/DC] ▶ Bloques de programa ▶ Conexion_Modbus [DB2]

Conservar valores actuales Instantánea Copiar instantáneas a valores de arranque Cargar valores de arranque como valores actuales

Conexion_Modbus											
Nombre	Tipo de datos	Offset	Valor de arranque	Valor de observación	Remanen...	Accesible d...	Escrib...	Visible en ...	Valor de a...	Comentario	
Static											
MainConection	TCON_IP_V4	0.0				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
Interfaceld	HW_ANY	0.0	64	64		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		HW-identifier of IE-interface submodule	
ID	CONN_OUC	2.0	10	16#000A		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		connection reference / identifier	
ConnectionType	Byte	4.0	11	16#08		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		type of connection: 11=TCP/IP, 19=UDP (17=TCP)	
ActiveEstablished	Bool	5.0	false	FALSE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		active/passive connection establishment	
RemoteAddress	IP_V4	6.0				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		remote IP address (IPv4)	
ADDR	Array[1..4] of Byte	6.0				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address	
ADDR[1]	Byte	6.0	0	16#00		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address	
ADDR[2]	Byte	7.0	0	16#00		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address	
ADDR[3]	Byte	8.0	0	16#00		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address	
ADDR[4]	Byte	9.0	0	16#00		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address	
RemotePort	UInt	10.0	0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		remote UDP/TCP port number	
LocalPort	UInt	12.0	502	502		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		local UDP/TCP port number	

Nota: Programación en Ladder.

En esta sección de la configuración del “Datos_Hol” se agrega los registros que se enviarán y la cantidad de registro que tomara el Modbus cliente. En esta investigación se designa un registro para enviar los valores escalados del transmisor de presión, toma el registro 1 tal como se observa en la Figura 35.

Figura 35

Datos en Hold Register

Presión_V2 ▶ PLC_1 [CPU 1214C DC/DC] ▶ Bloques de programa ▶ Datos_Hol [DB3]

Conservar valores actuales Instantánea Copiar instantáneas a valores de arranque Cargar valores de arranque como valores actuales

Datos_Hol										
Nombre	Tipo de datos	Offset	Valor de arranque	Valor de observación	Remanen...	Accesible d...	Escrib...	Visible en ...	Valor de a...	
Static										
Register	Array[0..10] ...	0.0				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Register[0]	Int	0.0	0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Register[1]	Int	2.0	0	5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Register[2]	Int	4.0	0	25		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Register[3]	Int	6.0	0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Register[4]	Int	8.0	0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Register[5]	Int	10.0	0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Register[6]	Int	12.0	0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Register[7]	Int	14.0	0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Register[8]	Int	16.0	0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Register[9]	Int	18.0	0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Register[10]	Int	20.0	0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Nota: Los registros se almacenan en array de 0-10. Los valores son enteros de cada registro.

3.4.2 Control remoto del proceso

Para hacer el control remoto de la planta se debe de agregar un base de datos, en esta sección se asigna “Datos_U”. la subsección de “Variable_INT” se asigna para los valores de la presión y el variador. Mientras tanto en la otra subsección nombrado “Variable_BOOL” se asigna el estado del sistema, el inicio y la parada del proceso.

Figura 36

Asignación de variables

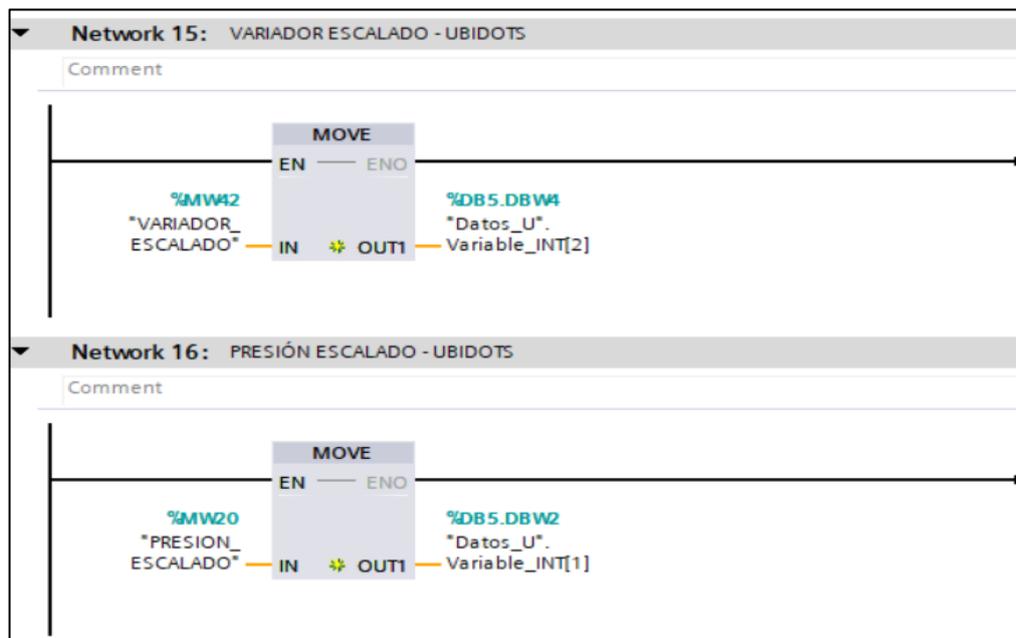
Datos_U									
	Name	Data type	Offset	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	Variable_INT	Array[0..5] o...	0.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	Variable_INT[0]	Int	0.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	Variable_INT[1]	Int	2.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	Variable_INT[2]	Int	4.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	Variable_INT[3]	Int	6.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	Variable_INT[4]	Int	8.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	Variable_INT[5]	Int	10.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	Variable_BOOL	Array[0..5] of Bool	12.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	Variable_BOOL[0]	Bool	12.0	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	Variable_BOOL[1]	Bool	12.1	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	Variable_BOOL[2]	Bool	12.2	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	Variable_BOOL[3]	Bool	12.3	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	Variable_BOOL[4]	Bool	12.4	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15	Variable_BOOL[5]	Bool	12.5	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Nota: Variables almacenados enteros y booleanos.

Despues de estructurar la base de datos, se agrega un segmento 15 con la acción MOVE, para almacenar datos de variador escalado y también para la presión escalado.

Figura 37

Variador y Presión - Ubidots

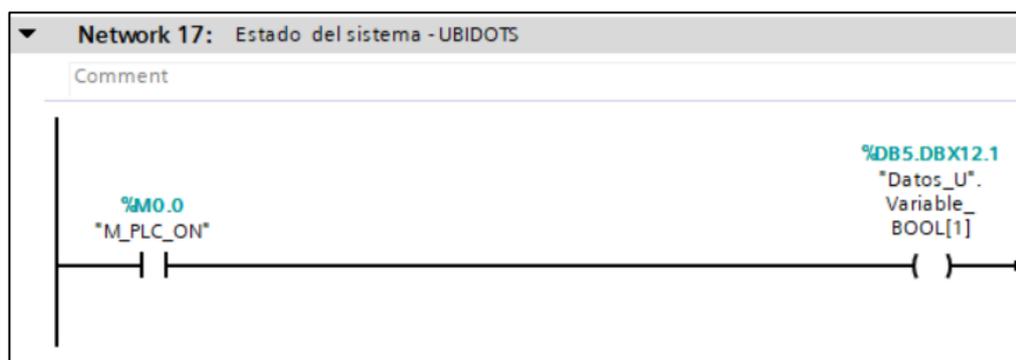


Nota: Direccionamiento de variables.

En el segmento 17 se configura el estado del sistema con la variable booleana %M_PLC_ON.

Figura 38

Estado del sistema - Ubidots

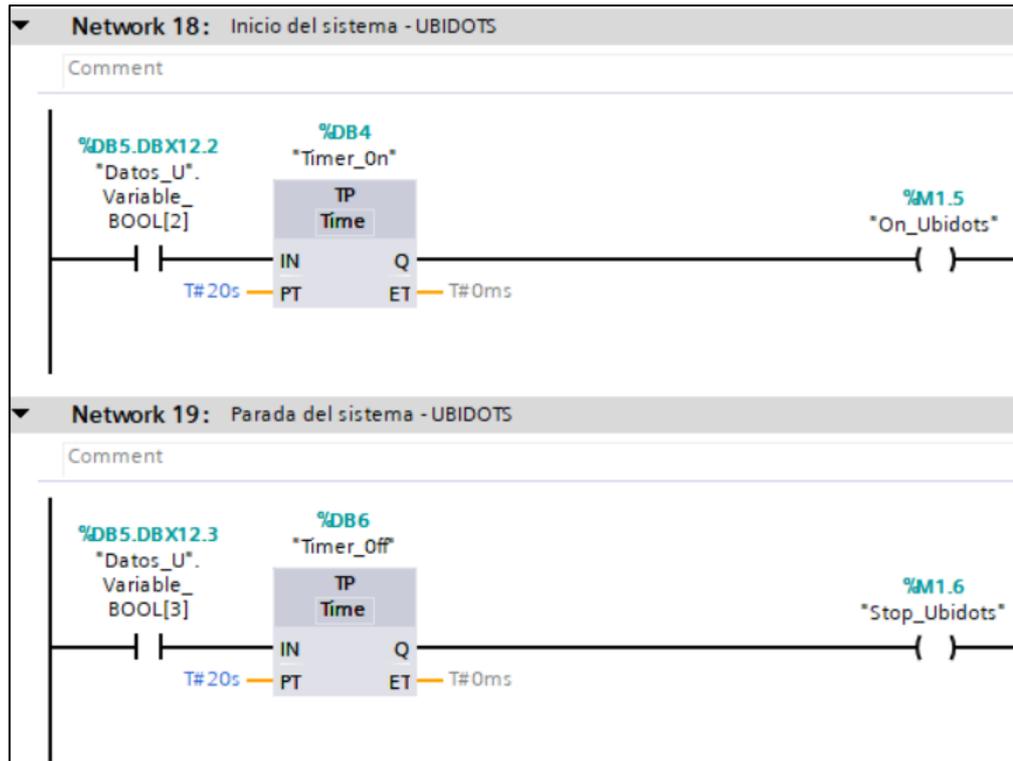


Nota: Estado del sistema del encendido y parada.

Para el inicio y parada del sistema se registra como variables booleanas. A través de la acción “Time” es posible controlar remotamente el inicio y la parada.

Figura 39

Inicio y parada - Ubidots

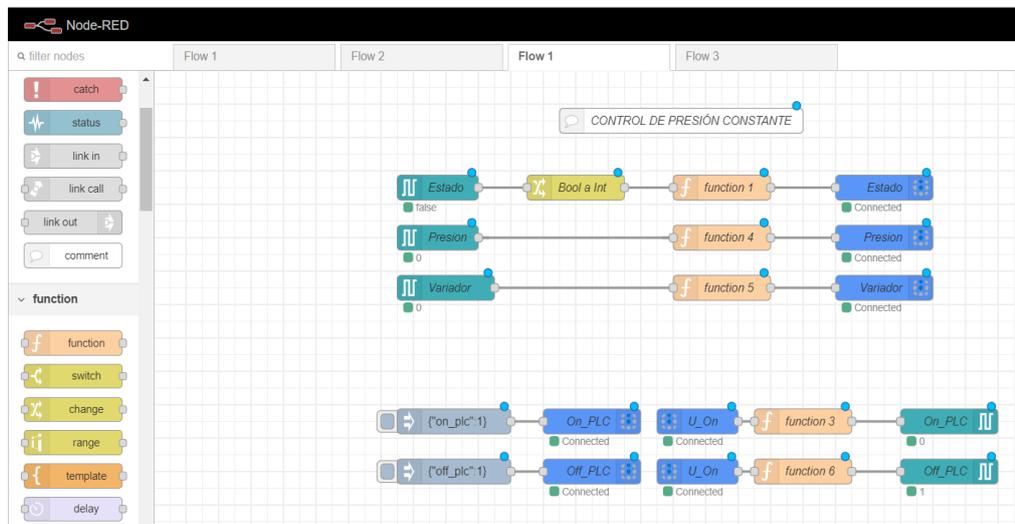


Nota: Segmentos direccionados para el control mediante Ubidots.

Para la interacción entre la plataforma Ubidots y la planta de presión es necesario armar el Flow en Node-Red. Los nodos más importantes son el S7 y Ubidots para PLC. La plataforma Ubidots genera los Tokens para el control y monitoreo de la planta.

Figura 40

Node-red

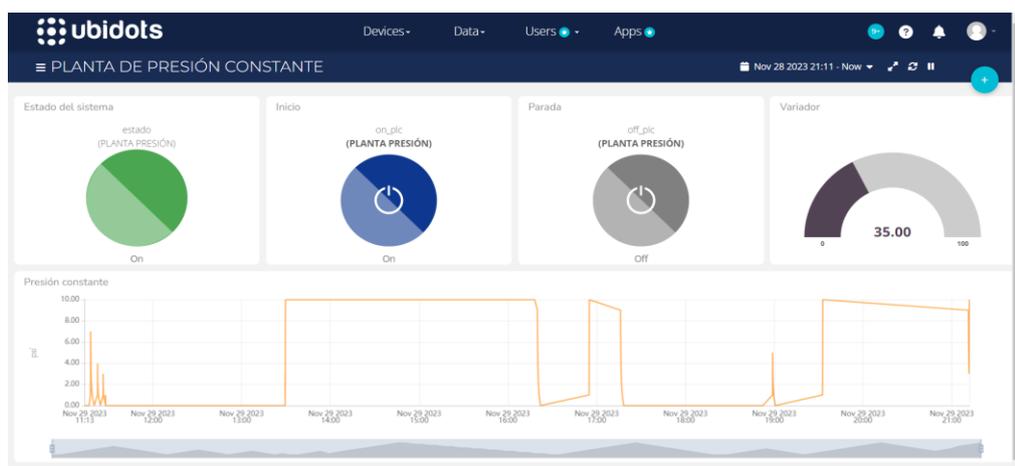


Nota: Diagrama de nodos generado por Node -Red.

También Ubidots proporciona un control remoto de inicio y parada de la planta. Esto es muy práctico para realizar pruebas en distancias remotas.

Figura 41

Control remoto en Ubidots



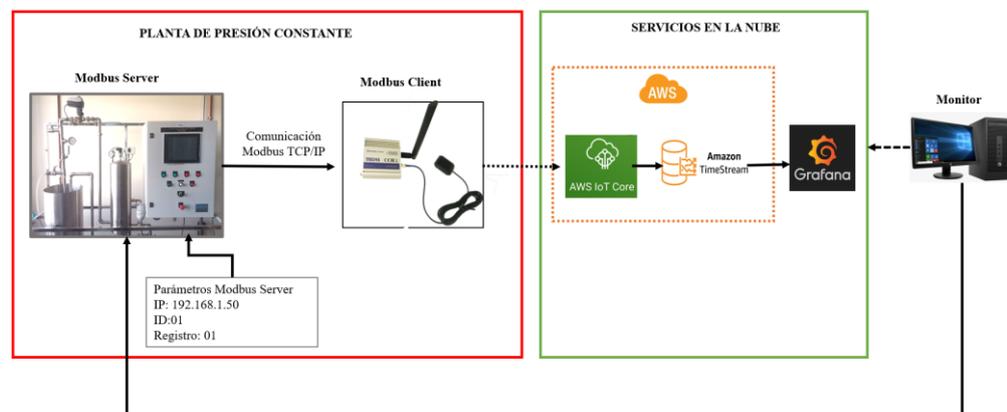
Nota: Panel de visualización del sistema de control en Ubidots.

3.4.3 Configuración TRB245

Para la adquisición y envío de datos, se debe tener consideración las soluciones según las matrices morfológicas del sistema integrado y eléctrico (tabla 6 y 8). En base en la solución 1, el enrutador industrial (Modbus cliente) mediante el protocolo Modbus TCP/IP solicita los datos de la planta y luego armar el formato JSON para él envío en AWS IoT Core.

Figura 42

Topología de envío de datos - TRB245



Nota: la topología es la solución 1, según los criterios técnicos y económicos (Tabla 11 y Tabla 12)

Para leer los datos de los registros Modbus del controlador de la planta de presión constante, es necesario tener los parámetros ya programados, conjuntamente la configuración Modbus TCP/IP tal como se muestra en la Figura 42 (arriba). Luego de la configuración del Modbus, se dispone los registros que se desea leer y guardar en la memoria del TRB245, esto proceso se puede mostrar en la siguiente Figura 43.

Figura 43

Solicitud de datos Modbus TCP/IP

The screenshot shows a web interface for configuring Modbus requests. It is divided into three main sections:

- REQUESTS CONFIGURATION:** A table with columns for NAME, DATA TYPE, FUNCTION, FIRST REGISTER NUMBER, REGISTER COUNT / VALUES, and BRACKETS. Two rows are visible: 'Pressure' and 'Variatior'. Each row has a 'Use brackets' dropdown and a toggle switch (currently 'off').
- ADD NEW REQUEST:** A section with a 'NEW REQUEST NAME' input field and an 'ADD' button.
- REQUEST CONFIGURATION TESTING:** A section with a 'Requests' dropdown menu (set to 'Pressure') and a 'TEST' button.

At the bottom of the interface, there is a large empty text area with a placeholder '[10]'.

Nota: La configuración de cada parámetro varía según las marcas dispositivos. Teltonika web.

En la sección de “ADVANCED SENDER SETTING”, se configura el protocolo MQTT que permite el envío del formato JSON; el enrutador por especificación propia, muestra las sintaxis para generar un mensaje el formato JSON lo cual se puede observar en la Figura 44. Para armar el formato JSON se debe tener en consideración que variable se seleccionará para él envío a la nube. Para el caso de esta investigación se denotará: {"time": %t, "device": "PLC_SIEMENS", "%r": %a}

Figura 44

Formato JSON

Date (Linux timestamp) - %t	Date (binary Linux timestamp, UINT32BE) - %T	Date (Day/Month/Year Hour:Minute:Second) - %d
MODBUS slave ID - %i	MODBUS slave IP - %p	Binary Modbus slave ID (UINT8) - %I
Start register number - %s	Binary start register address (UINT16BE) - %S	Register data (JSON object) - %a
Full MODBUS register address - %z	Request name - %r	Slave name - %n

Nota: La configuración de cada parámetro varía según las marcas dispositivos. Adaptado de Teltonika web.

Para la conexión entre el enrutador TRB245 y los servicios de AWS se debe generar los certificados. Para ello se debe de acceder a la consola de AWS IoT Core; en la sección de “Manage” y luego selecciona “Things” y finalmente “Create things”. En esta sección se podrá generar los certificados.

Figura 45

Certificados AWS

Configure device certificate - optional

Auto-generate a new certificate (recommended)
Generate a certificate, public key, and private key using AWS IoT's certificate authority.

Use my certificate
Use a certificate signed by your own certificate authority.

Upload CSR
Register your CA and use your own certificates on one or many devices.

Skip creating a certificate at this time
You can create a certificate for this thing and attach a policy to the certificate at a later time.

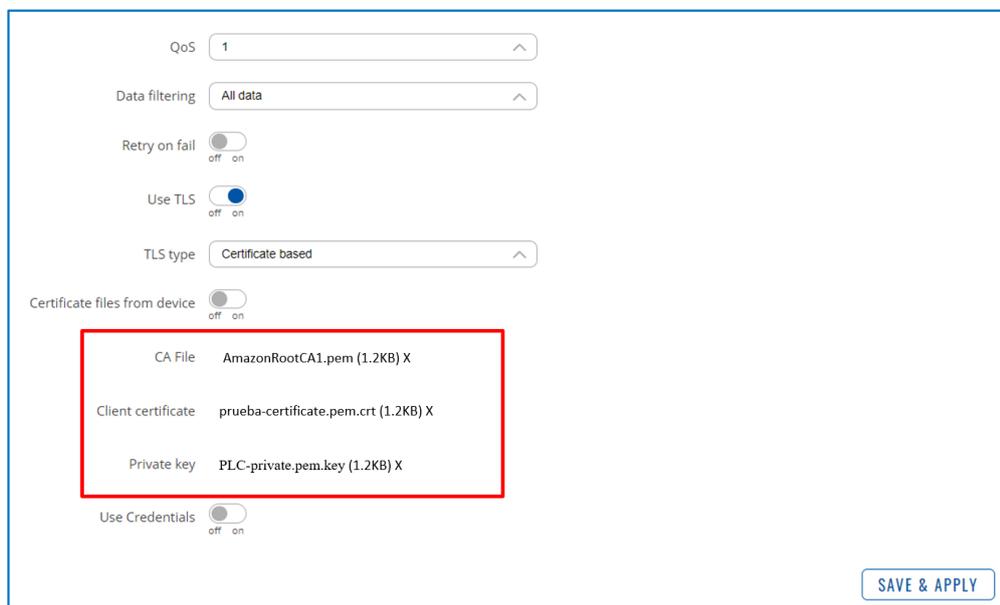
Nota: Los certificados son los accesos de los equipos a la nube. Generado por AWS

Los certificados generados por AWS son:

- **Certificado root:** AmazonRootCA1.pem
- **Certificado:** prueba-certificate.pem.crt.
- **Certificado privado:** PLC-private.pem.key.
- **Certificado público:** PLC-public.pem.key

Figura 46

Configuración de acceso a AWS



QoS 1

Data filtering All data

Retry on fail

Use TLS

TLS type Certificate based

Certificate files from device

CA File	AmazonRootCA1.pem (1.2KB) X
Client certificate	prueba-certificate.pem.crt (1.2KB) X
Private key	PLC-private.pem.key (1.2KB) X

Use Credentials

SAVE & APPLY

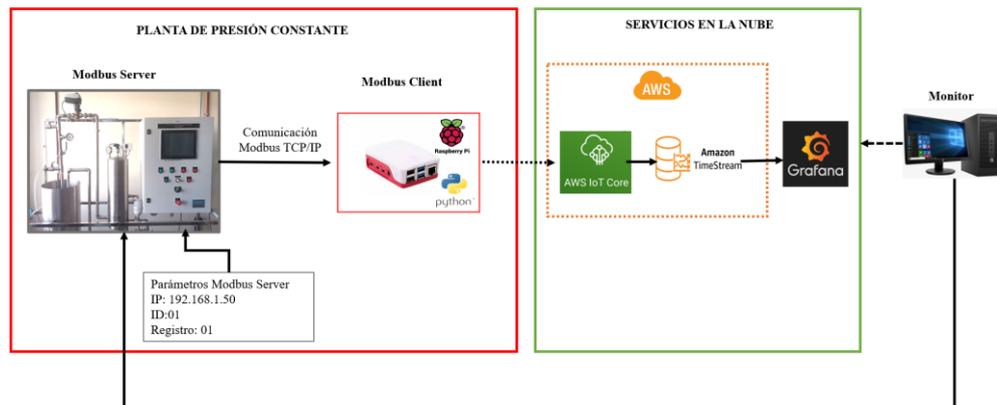
Nota: Los certificados son los accesos de los equipos a la nube. Generado por AWS

3.4.4 Programación en Raspberry PI

Según la matriz morfológica del sistema eléctrico (tabla 8) y los conceptos de solución técnico – económico, el uso del Raspberry PI 3 (solución 1) aprovecha la red local. El dispositivo lee los registros del PLC mediante el protocolo Modbus TCP/IP con la operación back-end desarrollado en Python. Para el monitoreo y control del dispositivo se debe de activar los protocolos SSH y VNC.

Figura 47

Topología envió de datos a la nube – segunda solución



Nota: La topología es la solución 1, según los criterios técnicos y económicos (Tabla 11 y Tabla 12)

El Raspberry PI opera un sistema operativo Linux, por ende se configura en la terminal. Primeramente, se debe descargar el software de programación libre Python y la librería Pymodbus, esta librería permite leer y escribir datos. Luego, descargar la librería de AWSIoTPythonSDK para conectar con los servicios de AWS, tal como se muestra en la Figura 48.

Figura 48

Librerías de Python

```
1 pip sudo apt install python3-pip
2 pip install pymodbus
3 pip install AWSIoTPythonSDK
```

Nota: adaptado de Pymodbus (2023)

En el código se debe configurar los parámetros del PLC con la dirección del dispositivo, la dirección IP y el puerto. Luego los registros de Modbus Server con el registro que se leerá la cantidad de registros que se leerán. Ver Figura 49.

Figura 49

Configuración Modbus Server

```
1 # Configuración de Modbus
2 PLC_IP = '192.168.1.50'           # Dirección IP PLC
3 PLC_PORT = 502                   # Puerto PLC
4 MODBUS_DEVICE_ADDRESS = 10      # Dirección del Modbus Server
5 REGISTRO_A_LEER = 1             # Registro del transmisor
6 REGISTRO_A_LEER_2 = 2          # Registro del variador
7 CANTIDAD_REGISTROS = 2          # Cantidad de registros a leer
8 EQUIPO = "PLC_SIEMENS"         # Nombre del equipo
```

Nota: adaptado de Amazon Web Services (2023).

Para la conexión e interacción entre el Raspberry PI y los servicios de AWS se debe generar los certificados. Los certificados constan con nombres muy extenso, que es necesario editarlos, para poder introducirlo al código. Esta investigación solo se usará tres certificados, ya que se está basando en el protocolo TLS.

Figura 50

Configuración AWS IoT Core

```
1 # Configuración de AWS IoT Core
2 IoT_ENDPOINT = 'a2ce3st7q61nm4-ats.iot.us-east-2.amazonaws.com' # Configuración de endpoint
3 IoT_ROOT_CA = 'root-ca.pem' # Certificado ROOT AWS IoT Core
4 IoT_CERTIFICATE = 'prueba-certificate.pem.crt' # Certificado AWS IoT Core
5 IoT_PRIVATE_KEY = 'PLC-private.pem.key' # Certificado privado AWS IoT Core
6 IoT_CLIENT_ID = 'topic/modbus' # Tópico AWS IoT Core / Modbus Server
```

Nota: Es recomendable configurar los certificados solo una vez.

Luego de configurar los certificados, es necesario formar el Payload, que es un paquete del mensaje que se desea enviar y procesar, ya que contiene el mensaje entre el cliente y el suscriptor.

Figura 51

Payload

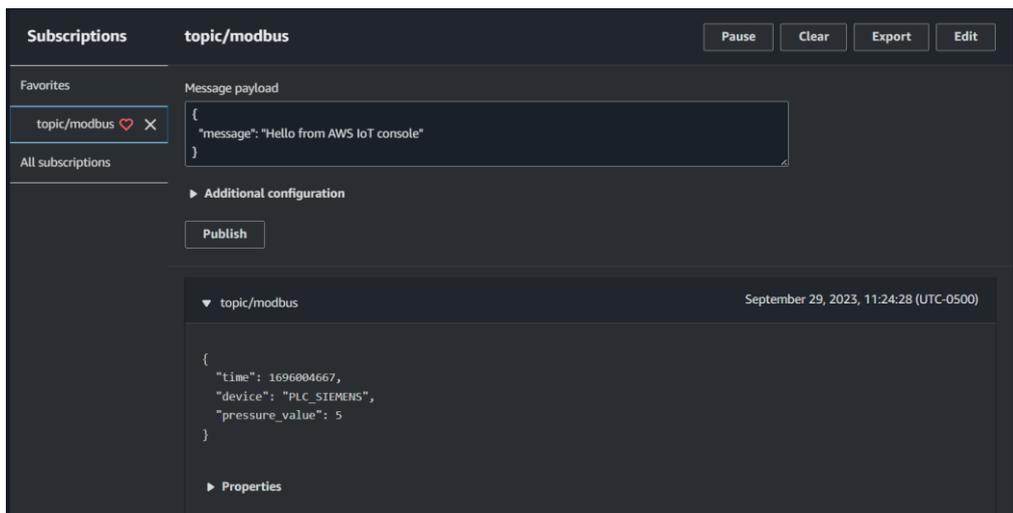
```
1 payload = f'{{"time": {ts}, "device": "{EQUIPO}", "pressure_value": {valor_1}, "variator": {valor_2}}}'
```

Nota: el formato Payload es también el formato JSON.

El AWS IoT adquiere los datos con el tópico “topic/modbus”; esto es posible a los certificados establecidos en el la operación back.end. Los mensajes enviados a través del **payload** llegan en un formato JSON tal como se aprecia la Figura 52. Y posteriormente hacer la ingesta de datos a una tabla.

Figura 52

Tópico AWS IoT Core



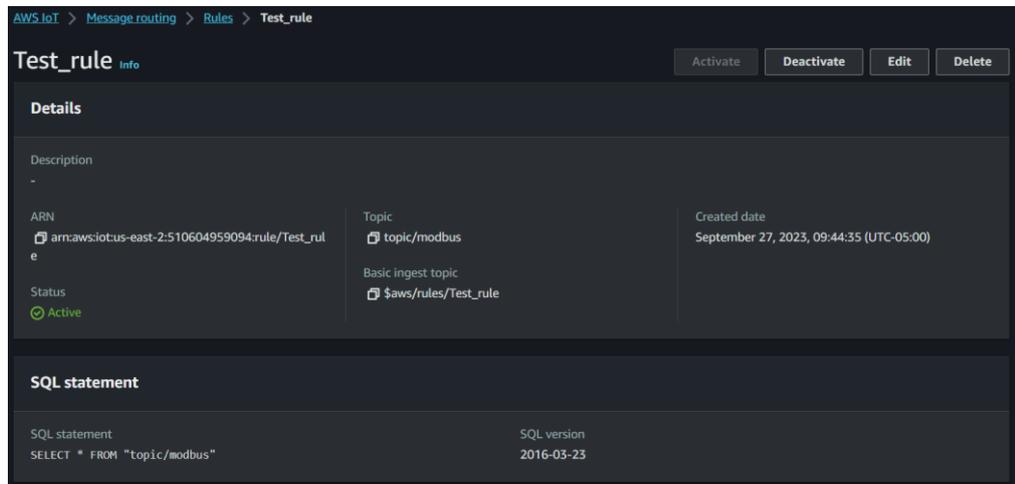
Nota: la Figura 52. solo hace el test si en caso que el formato JSON esté bien estructurado.

3.4.5 Estructura de base de datos

Para la ingesta de datos, se debe de crear una base de datos y dentro de ella una tabla que contendrá los mensajes captados por AWS IoT. Crear una regla de IoT Core en la sección de “Act” se selecciona Rules donde se genera una regla entre los dos servicios de AWS.

Figura 53

Reglas IoT



Nota: Las reglas de IoT son fundamentales para la ingesta de datos.

La configuración de Amazon de Timestream permite seleccionar el nombre de base de datos y la tabla proviene de la acción de AWS IoT Core. Se asigna las dimensiones que se mostrarán en la tabla. También se debe asignar en que espacio de tiempo son los mensajes, en esta investigación son en milisegundos. Luego seleccionar la regla generada anteriormente que es “Test_rule” tal como se observa en la Figura 54.

Figura 54

AWS Timestream

The screenshot shows the AWS Timestream console configuration for 'Action 1'. The configuration includes:

- Timestream table:** Write a message into a Timestream table. Includes a 'Remove' button.
- Database name:** TestDB. Includes a refresh button and a 'View' link. A 'Create Timestream database' button is also present.
- Table name:** test1_table. Includes a refresh button and a 'View' link. A 'Create Timestream table' button is also present.
- Dimensions:** Each record contains an array of dimensions (minimum 1). Dimensions represent the metadata attributes of a time series data point. A table with 'Dimensions name' (Device) and 'Dimension value' (\$device) is shown, with a 'Remove' button. An 'Add new dimension' button is also present.
- Timestamp value - optional:** \$time.
- Timestamp unit:** SECONDS.
- IAM role:** test_role. Includes a refresh button, a 'View' link, and a 'Create new role' button. A note states: 'AWS IoT will automatically create a policy with a prefix of "aws-iot-rule" under your IAM role selected.'

Nota: configuración de base de datos en series temporales.

Para la ingesta de datos es necesario configurar el Query editor. En esta sección, se elige en que base de datos se desea almacenar. En la figura proporcionada se elige “TestDB” que se creó inicialmente. Dentro de la sección de Query 1 se observa la sección de que base de datos obtendrá.

Figura 55

Acción Query

The screenshot shows the AWS Query Editor interface. The 'Database' dropdown is set to 'TestDB'. The 'Tables (1)' section shows 'test1_table'. The SQL query editor contains the following query:

```
-- Get the 10 most recently added data points in the past 15 minutes. You can change the time period  
if you're not continuously ingesting data  
SELECT * FROM 'TestDB'.test1_table WHERE time between ago(15m) and now() ORDER BY time DESC LIMIT  
10
```

The 'Run' button is highlighted in orange.

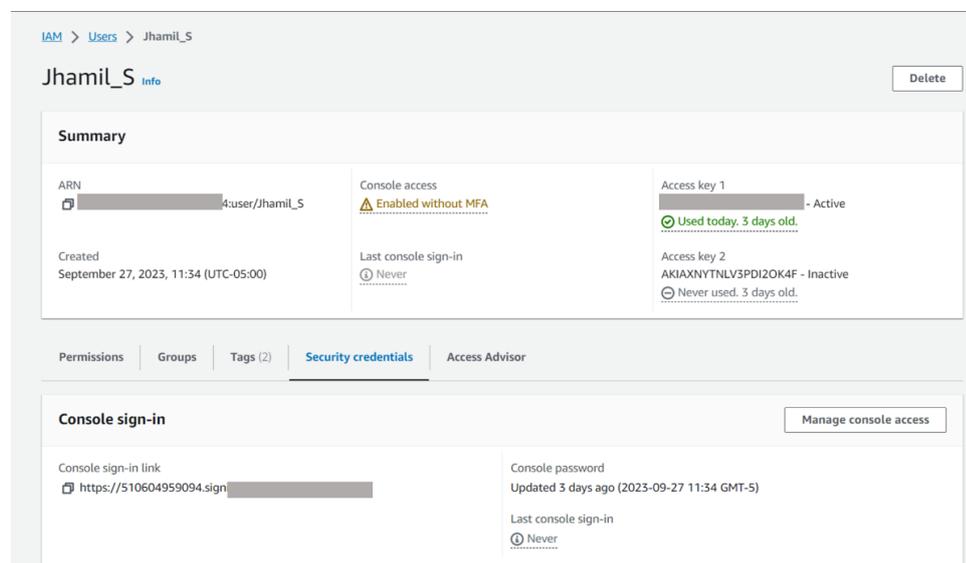
Nota: la acción Query hará las consultas para la consulta de datos.

3.4.6 Consulta de datos

Para la consulta de datos, primeramente, se debe de generar un usuario IAM en AWS para aplicaciones externas. Es necesario buscar en la consola el IAM para generar el acceso que permitirá interactuar entre AWS y Grafana. Todo acceso exterior generará solo un usuario, tal como se observa en la Figura 56.

Figura 56

Accesos externos - IAM

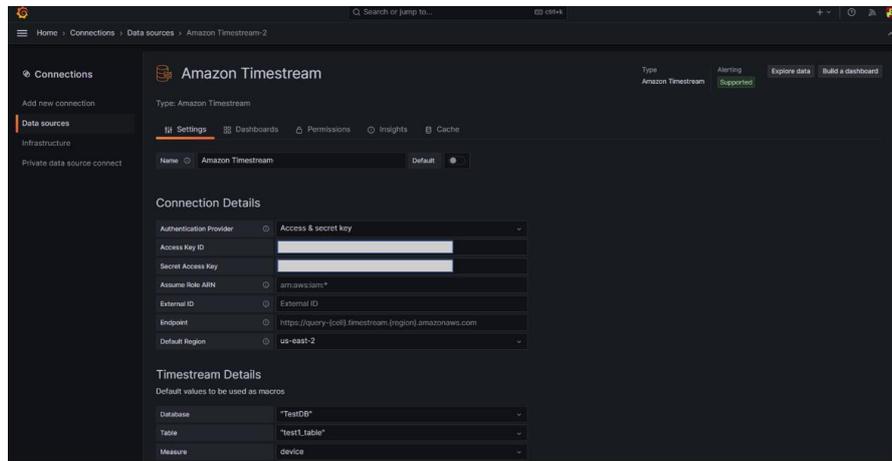


Nota: los accesos son de uso personal y deben ser restringidos cada vez que no se use.

La configuración en Grafana es principalmente seleccionar el tipo de base de datos se visualizará. Para lograr el objetivo se debe descargar Amazon Timestream en la sección de CONECTIONS y agregar datos. Esta sección se agrega los permisos correspondientes que fue generado de IAM de AWS. Esta sección enlazara el dato de base automáticamente. tal como se puede observar en la Figura 57.

Figura 57

Conexión AWS y Grafana



Nota: la configuración de acceso AWS y Grafana es de solo para el desarrollador.

Los datos que se almacena en series de tiempo, son consultas y permite mostrar la gráfica del monitoreo remoto de la planta de presión constante.

Figura 58

Consulta de Datos - Grafana



Nota: Consulta de datos de la planta de presión constante. Desarrollado en Grafana.

Mediante las herramientas de Grafana se genera las de tablas de visualización según el formato de JSON recibido del PLC.

Figura 59

Dashboards del monitoreo remoto



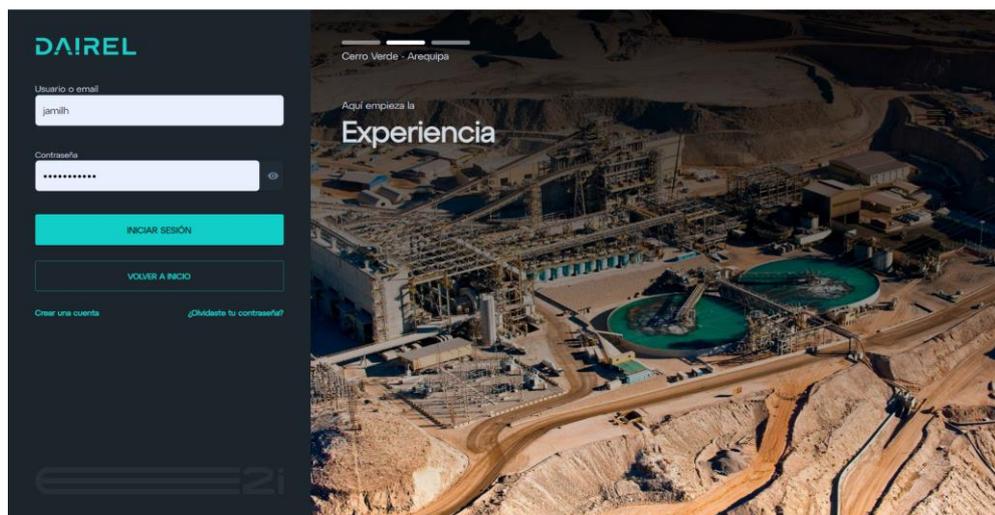
Nota: Grafana permite insertar la medidas y tipos de grafico en series de tiempo.

3.4.7 Monitoreo en DAIREL

Para hacer las consultas en la plataforma DAIREL, primeramente, se debe de registrar el usuario, tal como se muestra en la Figura 60. Es necesario contactare con los desarrolladores antes de subir el proyecto de tesis. DAIREL está desarrollado para industrias que generan datos permanentemente y tiene una demanda elevada.

Figura 60

Registro de usuario - DAIREL



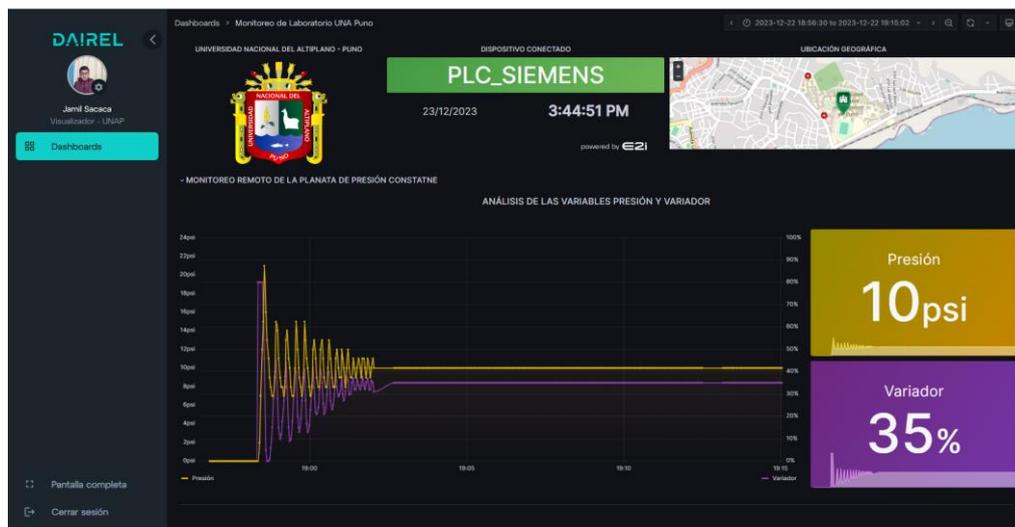
Nota: Visualización exterior de DAIREL. Desarrollado por E2I

Después de registrarse a la plataforma, Los desarrolladores de DAIREL proporcionará los certificados y los permisos. Para hacer las consultas del variador y del trasmisor de presión, es necesario configurar los certificados de DAIREL en el enrutador TRB245 en la solución 01, esto basado en la matriz morfológica (tabla 01).

Para la solución 02 según la matriz morfológica (tabla 10), se debe editar el código en el script **main.py** y colocar los certificados de DAIREL. Estos cambios son necesarios para hacer las consultas en tiempo real.

Figura 61

Consulta de datos en DAIREL



Nota: Visualización exterior de DAIREL. Desarrollado por E2I.

CAPÍTULO IV

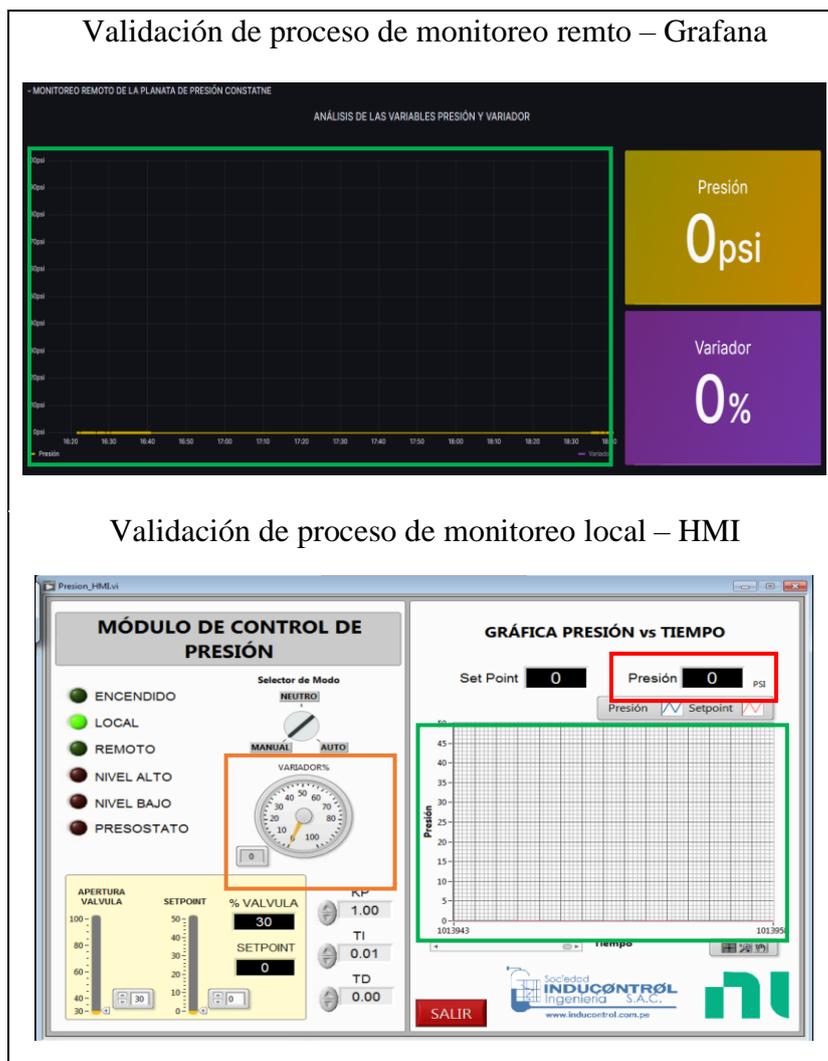
RESULTADOS Y DISCUSIÓN

4.1 OPTIMIZACIÓN DEL PROCESO DE MONITOREO

La tesis demostró la validación de los datos mostrados entre el HMI y Grafana de manera exitosa en la Figura 62 (comparativa), Donde los valores del transmisor se represan en PSI y el variador en porcentaje (%), ambos indican que están en reposo.

Figura 62

Validación de datos local - remoto

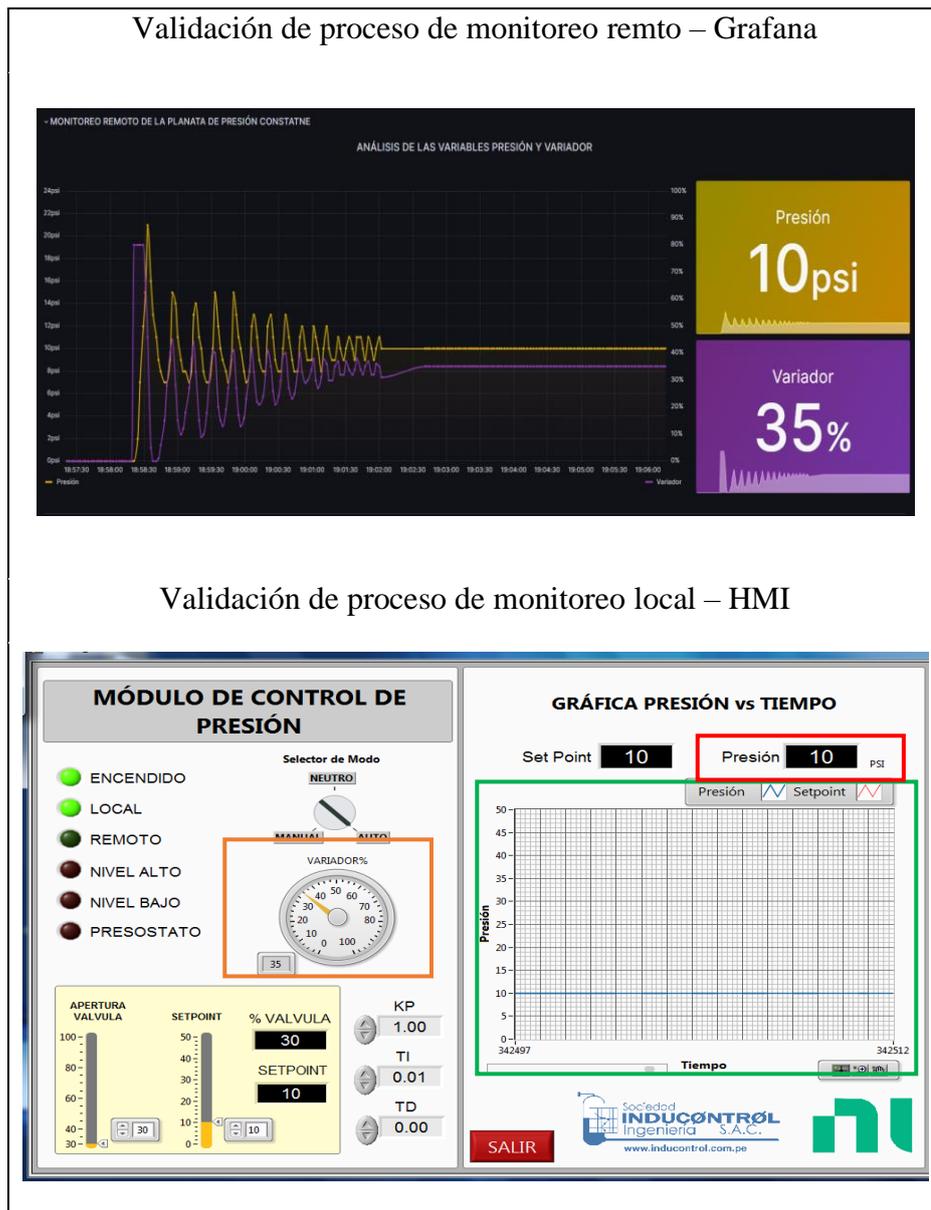


Nota: Representación de valores; verde, representa de la presión en tiempo real; rojo, indicador de la presión; naranja, variador en porcentaje.

Los valores de la presión constante empiezan a fluctuar en un sistema PDI. Según la Figura 63, el sistema de control tarda 10 minutos para estabilizarse en 10 psi. El punto máximo fue 15 psi.

Figura 63

Validación de proceso de monitoreo



Nota: El tiempo de la estabilidad puede variar, esto depende de cada planta de control de presión constante.

4.2 ANÁLISIS DE DATOS

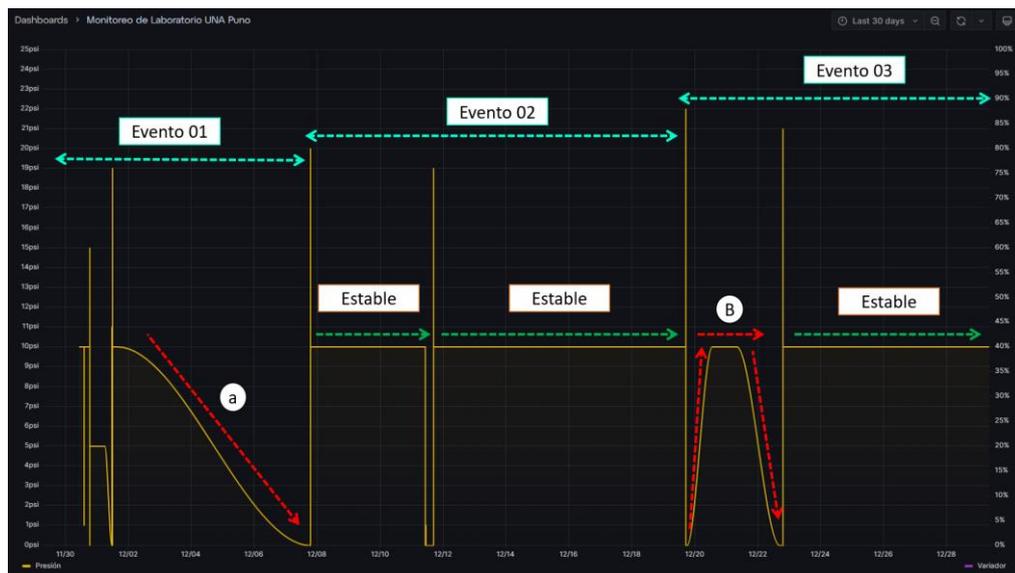
Se hizo el análisis de datos de 30 días del año 2023, donde DAIREL generó los reportes adecuados y de cada variable en series temporales, con diferentes unidades de medición.

4.2.1 Análisis de datos presión constante

Los datos almacenados en Timestream, permitieron generar los reportes de la plata de presión. También se obtuvo tres eventos diferentes tal como se muestra la Figura 64.

Figura 64

Datos de la presión constante



Nota: los datos mostrados se pueden actualizar permanentemente a través del tiempo. a: en este periodo de 6 días, la planta de presión constante estuvo en mantenimiento y no se obtuvo datos. b: en este periodo de 2 días, se hizo las pruebas de conexión y configuración de equipos, para optimizar.

Se consulto datos de 30 días atrás, lo que sería de 1 mes de prueba y tenemos los siguientes eventos:



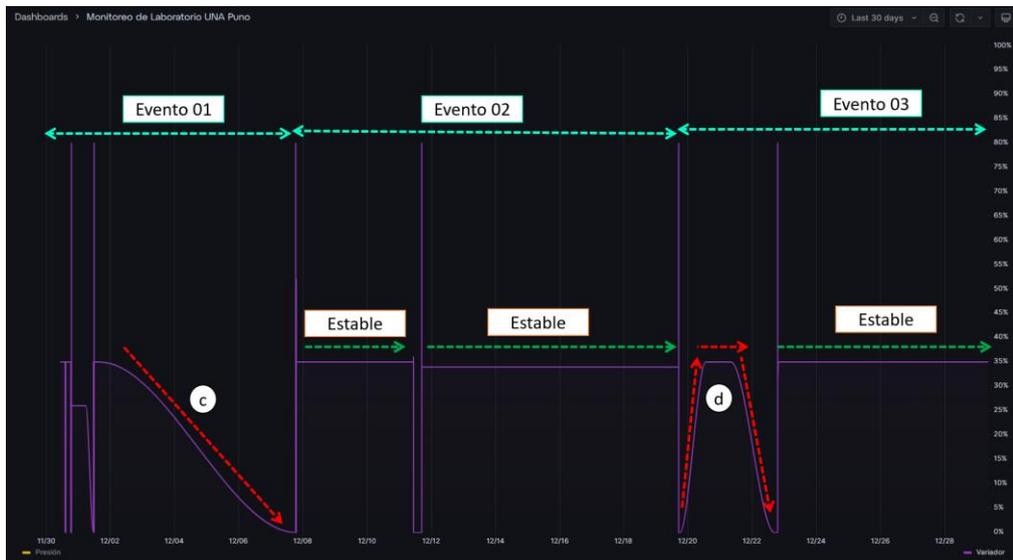
- **Evento 01:** en este enveto se generó el primer reporte, donde el 30 de noviembre se hizo la primera consulta y monitoreo en DAIREL. También se obtuvo la validación de datos de la planta de presión constante, donde a partir de la fecha 02/12 – 08/12 tal como se muestra en la Figura 64, se detectó una falla de conexión de los equipos, por lo cual se optó hacer el manteamiento correspondiente.
- **Evento 02:** en este evento no se reportó fallas de la planta de presión, si hubo manipulaciones y pruebas experimentales por lo cual no hubo reporte.
- **Evento 03:** en este evento se reportó un comportamiento no apropiado de la planta, donde se reportó, perdida de conexión a internet, y corte de energía eléctrica en el laboratorio, esto llevó que la planta de presión activase la parada de emergencia. También se realizaron las pruebas de los equipos después del manteamiento.

4.2.2 Análisis de datos variador

Se hizo la consulta de datos del variador de frecuencia, el variador cuenta con tres eventos similares a los datos de la presión constante tal como se muestra en la Figura 65. Se consulto datos de 30 días atrás, lo que sería de 1 mes.

Figura 65

Datos del variador



Nota: los datos mostrados se pueden actualizar permanentemente a través del tiempo. c: periodo de 6 días, mantenimiento. d: periodo de 2 días, se hizo las pruebas de conexión y configuración de equipos.

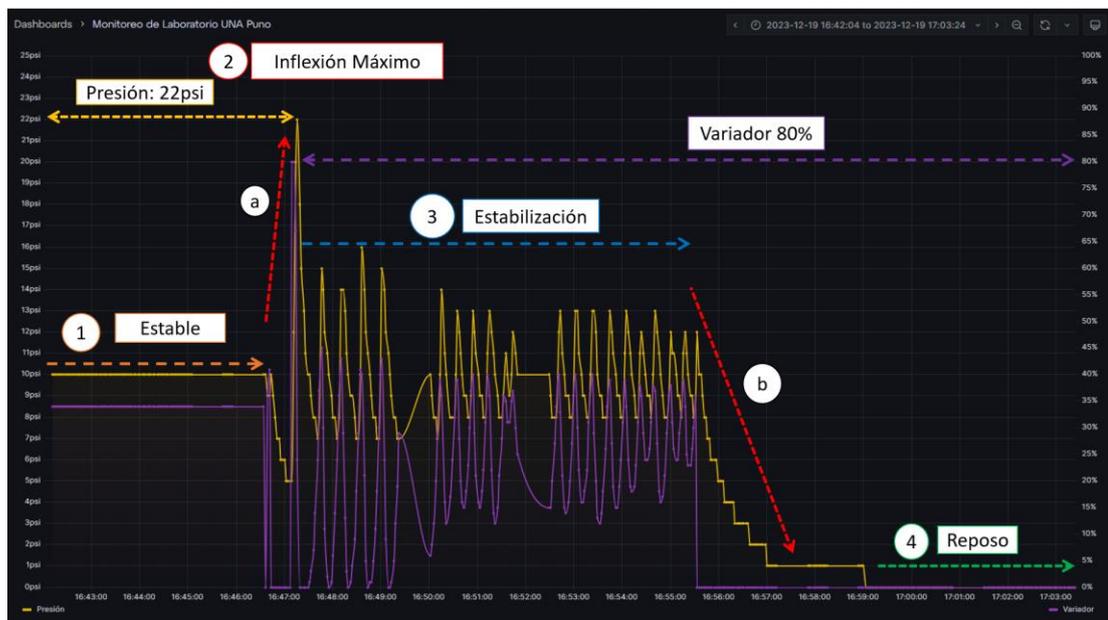
- **Evento 01:** en este evento, se generó el primer reporte, donde el 30 de noviembre se hizo la primera consulta y monitoreo en DAIREL. También se validaron los datos del variador, la fecha correspondiente es 02/12 – 08/12, donde se hizo el mantenimiento correctivo y configuración al variador.
- **Evento 02:** en este evento no se reportó fallas de la planta de presión.
- **Evento 03:** en este evento se reportó una anomalía, donde se reportó, corte de energía eléctrica en el laboratorio, esto llevó al variador activar la parada de emergencia, accionado al máximo al motor trifásico para evitar daños de los elementos de la planta de presión constante.

4.3 ANÁLISIS DE VARIABLES

Para hacer la interpretación y análisis de datos, se identificó patrones de datos y se tuvo que analizar un evento, donde hubo variaciones de procesos, tal como se muestra en la Figura 66. El umbral máximo consta con la acción de parada de emergencia, donde la presión empieza a acceder a un punto máximo de 22 psi (solo es representativo, pueden variar).

Figura 66

Interpretación de un sub evento



Nota: Cada evento es diferente, esto dependerá el proceso de control. Este resultado es obtenido del laboratorio de presión constante ya automatizado.

1. **Estable:** El sistema es estable debido que es un PDI, en esta parte se denota 10 psi, es decir es lo más óptimo y el varedor a 35%.
2. **Inflexión Máximo:** cuando el variador es manipulado al máximo alcanza un 80% de trabajo y la presión llega a 22 psi, el variador activar la parada de emergencia, haciendo que el varedor se apague de manera automática tal como se muestra la Figura 66.



3. **Estabilización:** representa el PDI del proceso de control, el sistema buscará estabilizarse según la manipulación. El punto máximo en estabilización es 15 psi, mientras tanto el variador indica 45%, es muy frecuente que se tarde de estabilizarse de 10 a 20 minutos.
4. **Reposo:** el sistema está en 0 psi, debido a la parada de emergencia del vareador, pero aún no está en marcha. Este evento será repetitivo cada vez que se reinicia un dispositivo por emergencia.

4.3.1 Correlación de variables

Para la correlación de variable se tomó el proceso más repetitivo de toda la población. Para lo cual se aplica la correlación lineal y la suavización exponencial en los tres procesos; la estabilización, anomalía e inestabilidad de datos.

4.3.1.1 Interpretación de procesos

En este proceso de estabilización tomaron 373 datos de la presión y el variador ambos almacenado en un archivo CSV (Comma-Separated Values).

Figura 67

Estabilización de datos

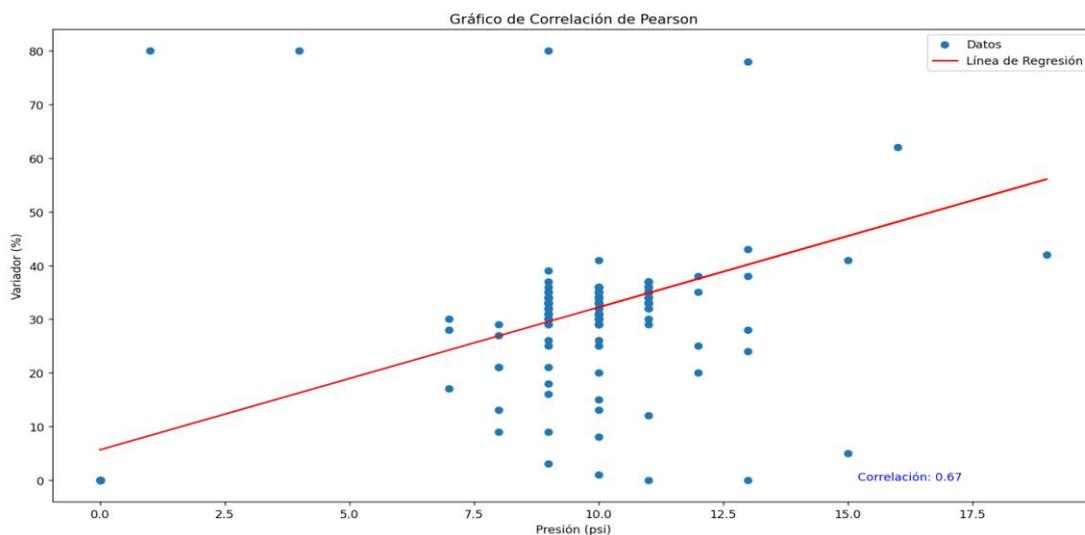


Nota: Tomado el 16:19 – 16:31 (12 minutos) de 11 de diciembre de 2023.

La correlación de Pearson es $r = 0.67$ (correlación positiva), donde las variables variador y la presión tienen una relación positiva.

Figura 68

Correlación de la estabilización

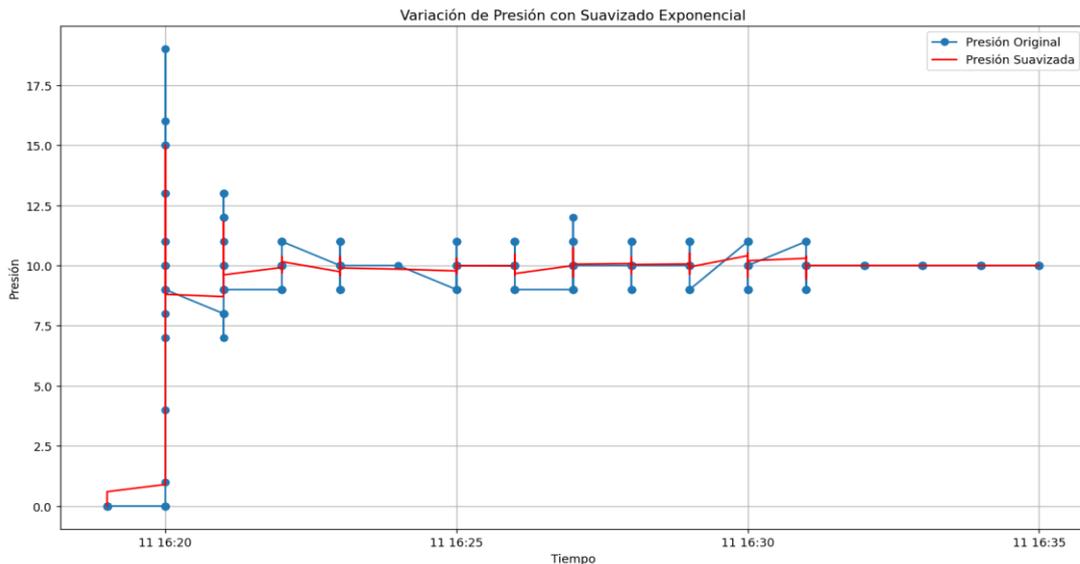


Nota: Tomado el 16:19 – 16:31 (12 minutos) de 11 de diciembre de 2023. Su correlación es de 0.67.

En la Figura 69 se muestra el análisis de suavizado exponencial. Este análisis muestra los datos posibles que debería de obtenerse al variar parámetros de control.

Figura 69

Suavizado exponencial de estabilización



Nota: Suavización exponencial de 373 datos.

Para llevar la correlación del proceso de anomalía se tomó 2799 datos de la población general.

Figura 70

Anomalía de pérdida de datos

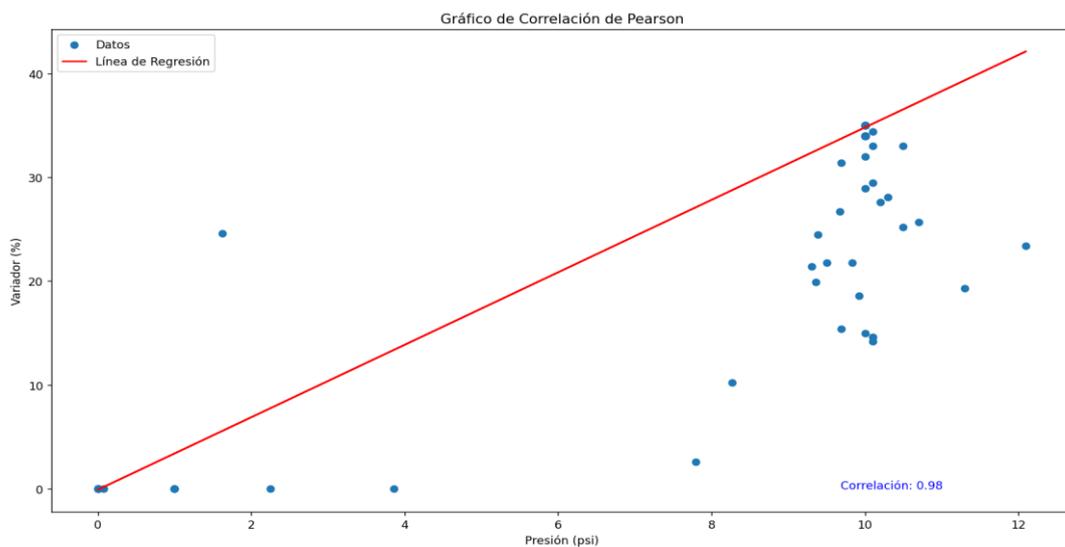


Nota: Tomado 19 de diciembre al 22 de diciembre.

La correlación tiene de $r = 0.98$, tal como se muestra en la figura 70. La correlación es tratable. La anomalía representa pérdida de datos.

Figura 71

Correlación de pérdida de datos



Nota: Tomado 19 de diciembre al 22 de diciembre. Correlación positiva de 0.98.

También se obtuvo datos del proceso de inestabilidad temprana de la plata, tal como se muestra la Figura 71.

Figura 72

Inestabilidad de datos

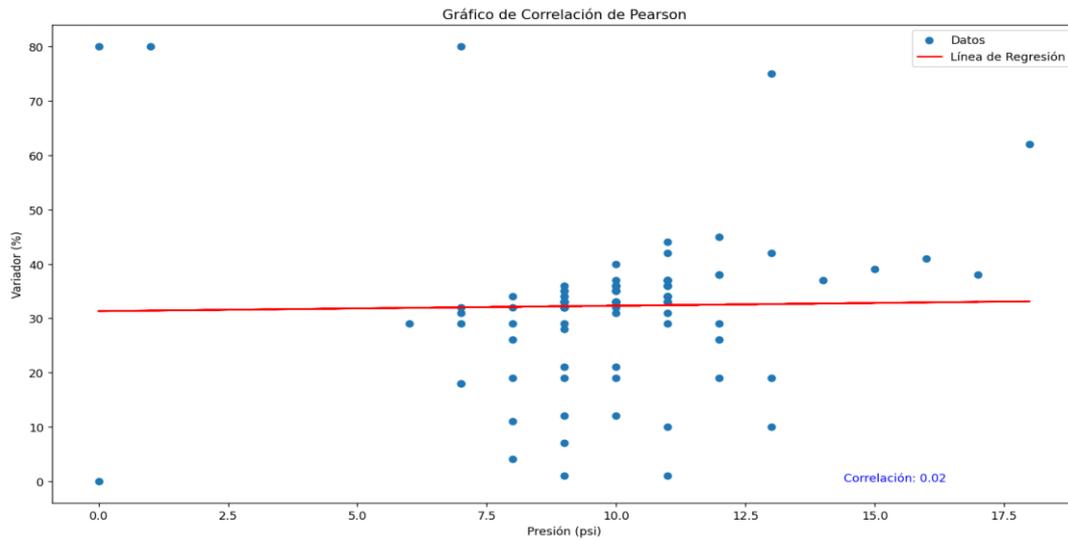


Nota. Tomado el 15:10 a 15:14 del 29 de diciembre.

La correlación de la inestabilidad es de tendencia $r = 0.02$, es relativamente muy bajo, por lo que no es estable, lo que afecta permanentemente a los equipos.

Figura 73

Correlación de la inestabilidad

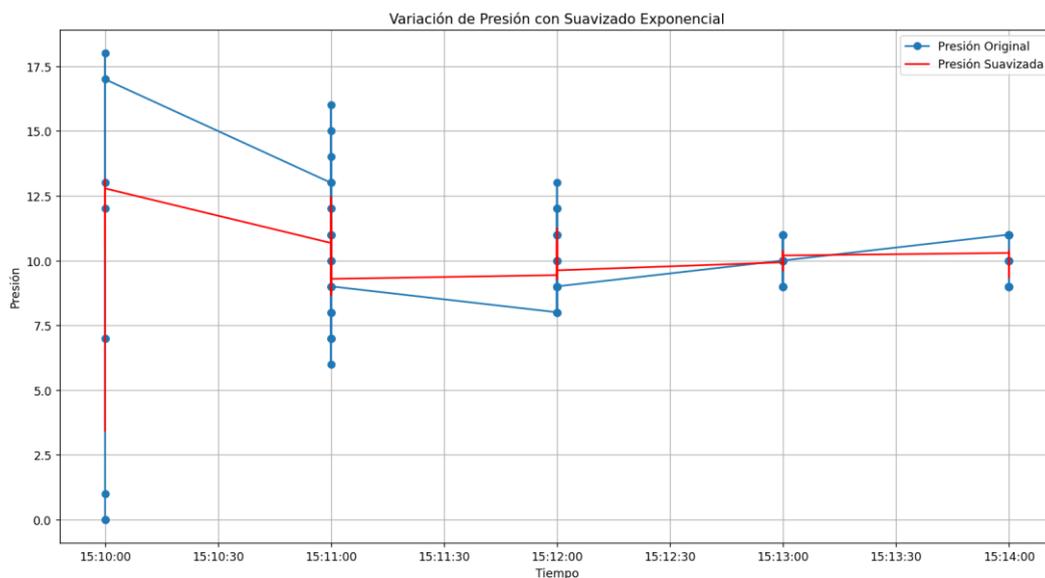


Nota: Tomado el 15:10 a 15:14 del 29 de diciembre. Correlación de 0.02.

En la Figura 73 se muestra la suavización de datos de la inestabilidad, los datos tomados son 107, lo que implica que no es predecible.

Figura 74

Suavizado exponencial de inestabilidad



Nota: Suavización exponencial de 107 datos.

4.3.2 Correlación de Pearson

La Tabla 15 muestra el resumen de los tres procesos; estabilización, anomalía y la inestabilidad. La anomalía y la estabilización tiene una significancia de 0.001, es decir son correlaciones positivas en sus respectivos procesos. Mientras tanto la significancia de la inestabilidad es de 0.848 su correlación tiende a 0.019, por lo que no tratable.

Tabla 15

Comparación de correlaciones de Pearson

		Proceso					
		Estabilización		Anomalía		Inestabilidad	
		Presión	Variador	Presión	Variador	Presión	Variador
Presión	Correlación de Pearson	1	.671**	1	.985**	1	.019
	Sig. (bilateral)		<.001		<.001		.848
	N	373	373	2799	2799	107	107
Variador	Correlación de Pearson	.671**	1	.985**	1	.019	1
	Sig. (bilateral)	<.001		<.001		.848	
	N	373	373	2799	2799	107	107

Nota: **La correlación es significativa en el nivel 0.01 (bilateral)

Según los datos obtenidos, se tiene los siguientes reportes basado en DAaaS:

1. **Calibración de los parámetros:** los datos no es lo adecuado cada vez que el tiempo transcurre, lo que se recomienda es calibrar los



parámetros PID esto permitirá tener datos correctos y correlaciones entre variador y presión.

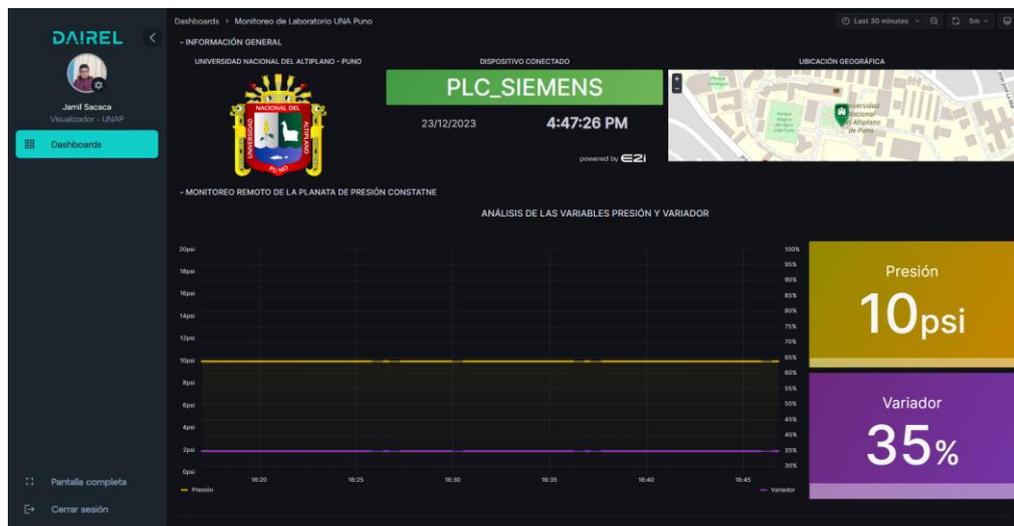
2. **Manteamiento correctivo:** esto es debido a que el motor accionado mediante el variador requiere un mantenimiento o calibrar sus parámetros. Esto mejorará las señales obtenidas de esta investigación.
3. **Programación PLC:** debido a los datos mostrados en la posición “Automático”, se debe de restablecer la programación principal y dar nuevos parámetros y así preservar la vida útil de los equipos.

4.3.3 Visualización de variables

La plataforma DAIREL permitió monitorear el estado de operación de la planta de presión constante, donde se obtuvo las variables de medición de la presión constante y el variador en series de tiempo. Según la Figura 75, los valores de la presión se estabilizan de manera proporcional con la velocidad ejercida por el variador. Los valores son sistemas PID. Los valores de la presión están representados por el color amarillo (mostaza) y la velocidad (color morado) ejercida.

Figura 75

Visualización de variables presión - variador



Nota: Generado en la plataforma DAIREL.

4.4 ARQUITECTURA DEL SISTEMA

Se ha desarrollado una arquitectura del sistema de monitoreo remoto del proceso de control. La arquitectura del sistema integra dos soluciones óptimas, previamente analizadas en el aspecto técnico – económico basado en el análisis morfológico de las soluciones. En la Figura 76, se observa el proceso de una planta automatizada y la computación en la nube. A la planta de presión constante, se le agregó dos soluciones, con el objetivo de leer los registros de las variables mencionada en la anterior Figura 33 y enviar por el protocolo MQTT. La computación en la nube hace el proceso de adquisición, almacenamiento y consulta de datos.

La arquitectura del sistema integra la planta presión constante con la computación en la nube. Donde se registró el proyecto de tesis en la plataforma DAIREL, eso ayudó a reducir el costo del monitoreo drásticamente. También se generó reportes mediante mediato los datos y correlacionarlos para encontrar fallas del proceso de control de presión constante.



4.4.1 Arquitectura de la primera solución IIoT

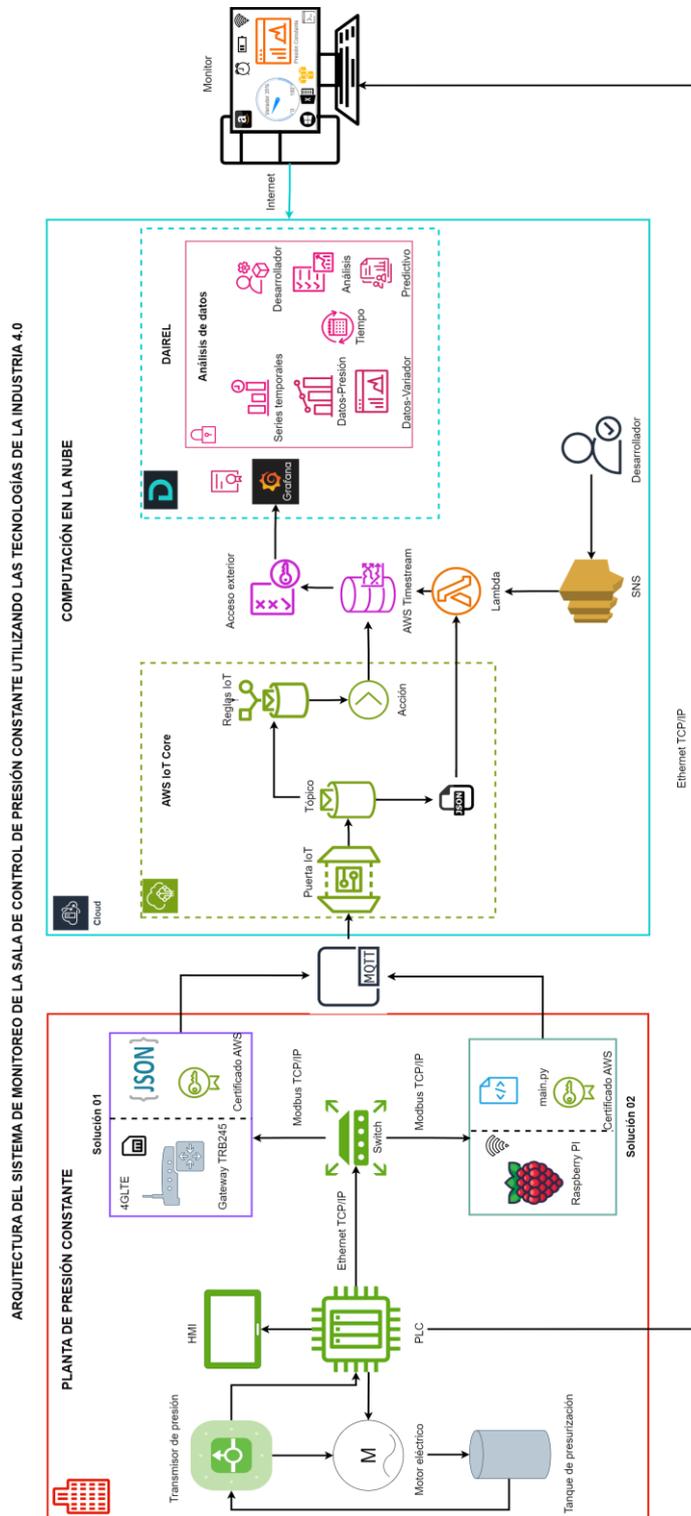
Esta la solución es la más óptima para la adquisición y envío de datos a la nube. Según la arquitectura del sistema, utiliza el enrutador industrial TRB245 para leer, almacenar y enviar a la nube. Esta solución permite enviar datos mediante la tecnología de 4G LTE; no es necesario tener un acceso de internet local debido a la tecnología incorporada dentro del dispositivo. Para esta validación, se hizo las configuraciones del protocolo Modbus TCP/IP y se cargó los certificados de AWS.

4.4.2 Arquitectura de la segunda solución IIoT

En esta solución, se desarrolló una arquitectura de software en un script de Python, la función principal es adquirir y enviar datos a DAIREL. Esta solución es viable si cuenta con acceso a internet local dentro del laboratorio de control. También es viable por el aspecto económico y académico.

Figura 76

Arquitectura del sistema de monitoreo



Nota: La arquitectura está basado en las prácticas de la computación en la nube.

4.5 DISCUSIONES

La investigación aborda de manera integral la industria 4.0, los procesos de la automatización industrial, los sistemas de comunicación hardware y software que son fundamental para las transferencias de datos con los servicios de la nube y por último el tratamiento de datos basado en análisis e interpretación en ciencia de datos.

Malik et al. (2021) aborda las nuevas tecnologías que mejoran significativamente los procesos de la industria y de los demás procesos como el monitoreo de la agricultura, atenciones médicas y negocios, en donde destaca la importancia de IIoT. Sin embargo, los resultados obtenidos de esta investigación proporcionan una visión más detallada del monitoreo e interpretación de datos para el mantenimiento predictivo de una planta industrial, donde evita riesgo de pérdidas económicas y las posibles fallas de los equipos.

Syufrijal (2019) en su investigación desarrolló un sistema de control de presión constante para distribución de agua, esta investigación integra sistemas de embebidos y el IoT para recolectar datos lo que sus resultados dan un monitoreo remoto. Mientras tanto los resultados de esta investigación son de mayor confianza que el de Syufrijal (2019), ya que el diseño de esta investigación tiene dos soluciones basadas en IIoT, por lo cual mejora drásticamente el proceso de control de presión constante y es adaptable a otros procesos de control.

La investigación de Muñoz Díaz (2021), donde integra la plataforma Ubidots para el monitoreo remoto de la planta de temperatura basándose en IoT en una industria. Sin embargo la presente investigación hizo uso de embebidos, donde no es elástico y poco seguro para monitorear y hacer el análisis de datos, por lo cual esta investigación se integró la plataforma DAIREL, donde la plataforma permite que los datos del usuario y del proyecto estén protegidos, también cuenta con un respaldo técnico en desarrollo de la



investigación haciendo que Ubidots sea limitado. Los resultados evidencian una optimización adaptable, sencilla y eficiente en supervisar en tiempo real, así como la toma de decisiones a distancia han reducido los tiempos de respuesta ante los cambios de la planta.

La modelización del sistema de distribución de agua mediante el sistema de presión constante propuesto por Diaz Rodas y Trujillo Aramburú (2019), donde tiene menor consumo de energía. Mientras tanto esta investigación está basada en integrar procesos de la industria 4.0, por lo que no se puede aseverar la reducción del consumo de energía, ya que son sistemas de control industrial y el mantenimiento es relativamente mayor de lo que proponen Diaz y Trujillo.

Además de los resultados obtenidos, existen algunos retos encontrados durante la validación, como la capacitación del personal a cargo del proceso de control, la capacitación del mantenimiento adecuado, la interpretación de datos y los servicios en la nube. Además, la protección de datos mediante la integración de la ciberseguridad, la protección de datos y la restauración de datos perdidos durante la transferencia y el almacenamiento de datos.



V. CONCLUSIONES

PRIMERO: Se abordó de manera integral el diseño del monitoreo remoto de la sala presión contante, utilizando tecnologías de la industria 4.0. Los resultados de esta investigación, empleando la metodología VDI 2206, indica que la validación e interacción remota entre los servicios de DAIREL y la planta de presión constante se llevaron a cabo de manera exitosa.

SEGUNDO: Las soluciones obtenidas son viables con base sólidas para la implementación en otros procesos de control industrial. También, se adaptan a la arquitectura de software Modbus desarrollado, encargado de leer los registros del transmisor, mientras tanto, el sistema de comunicación MQTT envía los registros en formato JSON, debidamente validados con los certificados de AWS. Las soluciones proporcionan una estructura de base de datos en series temporales para la consulta en Grafana.

TERCERO: La relación de las variables presión – variador permitieron hacer el análisis de datos como servicios, donde se realizaron las correlaciones de Pearson en tres procesos de sub eventos; estabilización, anomalía y estabilización. Donde se reportó fallas y perdidas de datos del proceso de control estudiado. También se aplicó el análisis de suavizado exponencial a los procesos donde se obtuvo las predicciones del posible tratamiento de datos.

CUARTO: La integración en la plataforma DAIREL ha sido fundamental para el monitoreo en tiempo real y ha facilitado la transición de un enfoque de un proceso automatizado a un enfoque de Industria 4.0. La capacidad de



aprovechar los datos y generar reportes a través el sistema de monitoreo remoto, el estado de los equipos, la detección de anomalías y la predicción ha sido crucial para la planificación estratégica y la mejora significativamente del proceso de control de presión.



VI. RECOMENDACIONES

PRIMERO: Se sugiere establecer convenios con empresas tecnológicas que podría proporcionar accesos, actualizaciones y soporte técnico. Un convenio con una empresa facilitará las adaptaciones de nuevos diseños basados en las tecnologías de la industria 4.0.

SEGUNDO: Considerar la posibilidad de adaptar el alcance del diseño y la implementación en otros procesos industriales. La aplicación del diseño proporcionaría una visión más completa de la industria 4.0 y la automatización, donde proporcionaría reportes basado en tiempo real.

TERCERO: Dada la interacción del diseño entre un proceso y la computación en la nube, se recomienda investigar continuamente las tecnologías de la industria 4.0, que puedan mejorar la eficiencia del sistema de monitoreo. La integración de la inteligencia artificial, conjuntamente con transmisores avanzados podría ser algunas posibilidades a considerar. Para la protección de datos críticos de los procesos, se sugiere integrar la ciberseguridad en los procesos industriales.



VII. REFERENCIAS BIBLIOGRÁFICAS

- Sarangapani, E. (2021). "Industry 4.0 technologies incorporated with Delta PLC based Smart Home Automation for Rural". *IOP Publishing*.
<https://doi.org/10.1088/1757-899X/1084/1/012112>
- Sundström, E. (2019). Creating a vision for the future of industrial control rooms under Industry 4.0. *Control 4.0*. Retrieved from www.diva-portal.org/smash/get/diva2:1323543/FULLTEXT01.pdf
- Amazon Web Services. (2023). AWS IoT Core. *Developer Guide*.
<https://docs.aws.amazon.com/pdfs/iot/latest/developerguide/iot-dg.pdf#what-is-aws-iot>
- Amazon Web Services. (2023). Información general sobre Amazon Web Services. *Documento técnico de AWS*.
https://docs.aws.amazon.com/es_es/whitepapers/latest/aws-overview/aws-overview.pdf
- Bassi , A., Bauer, M., & Fiedler , M. (2013). *Enabling Things to Talk*. Springer.
https://doi.org/DOI 10.1007/978-3-642-40403-0_1
- Bernal Hernández, R. (2013). Automatización del sistema de refrigeración para el proceso de extrusión de tubería de polietileno. Colombia.
- CEUPF MAGAZINE. (2023). *Tecnología* . <https://www.ceupe.com/blog/sistema-embebido.html>
- Creus Solé, A. (2011). *Instrumentación industrial*. Barcelona: Marcombo.
- Díaz Rodas, H., & Trujillo Aramburú, G. (25 de Octubre de 2019). *Sistema de control de flujo para presión constante de suministro de agua utilizando PLC en un edificio del Instituto de Ciencias y Humanidade* [Tesis de pregrado, Universidad de Ciencias y Humanidades]. Repositorio UCH.
<https://repositorio.uch.edu.pe/handle/20.500.12872/407>
- GitHub. (2022). *Cloud-Automation/node-modbus*. (CodeCreator, Ed.)
<https://github.com/Cloud-Automation/node-modbus>



- Graessler, I., & Hentze, J. (2020). The new V-Model of VDI 2206 and its validation. *at - Automatisierungstechnik*, 68(5). <https://doi.org/10.1515/auto-2020-0015>
- Hassanpour, V., Shayan, Z., Rajabi, S., & Mehdi Arefi, M. (2017, Noviembre). *Low-cost home automation using Arduino and Modbus protocol*. ResearchGate. <https://doi.org/10.1109/ICCIAutom.2017.8258694>
- HiveMQ. (2023). *MQTT Essentials*. HiveMQ: <https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt/>
- HiveMQ. (2023). *Transferring Data from Modbus to MQTT Broker for Advanced IIoT Data Use Case*. <https://www.hivemq.com/>
- INDUCONTROL S.A.C. (2020). Manual de usuario planta de control de presión. www.inducontrol.com.pe
- Jänsch , J., & Birkhofer , H. (2006). The development of the guideline VDI 2221 -the change of direction. *INTERNATIONAL DESIGN CONFERENCE - DESIGN 2006*, 49. <https://www.pmd.tu-darmstadt.de/pmd/index.de.jsp>
- Kamarka, A., Naiwrita, D., Baral, T., Manojee, C., & Md., R. (2019). Industrial Internet of Things: A Review. *2019 International Conference on Opto-Electronics and Applied Optics (Optronix)*, 1-6. <https://doi.org/10.1109/OPTRONIX.2019.8862436>
- Malik, P. K., Sharma, R., Singh, R., Gehlot, A., Satapathy, S., Alnumay, W., . . . Nayak, J. (2021). Industrial Internet of Things and its Applications in Industry 4.0: State of The Art. *Computer Communications*, 166, pp. 125-139. <https://doi.org/doi.org/10.1016/j.comcom.2020.11.016>
- Mehta, R., & Reddy, J. (2015). *Industrial process automation systems*. Oxford: ELSEVIER.
- Merchan Damian, A. R., & Damian Tapa, C. A. (2022). Diseño y desarrollo de un prototipo de módulo de alarma contra incendios para la universidad politécnica salesiana sede cuenca. <https://dspace.ups.edu.ec/bitstream/123456789/23495/1/UPS-CT010094.PDF>



- Microsoft Azure. (2023). *Documentación de Azure IoT*. <https://learn.microsoft.com/es-es/azure/iot/iot-introduction>
- Modbus. (2023). Obtenido de <https://modbus.org/>
- Modbus Organization. (2006). *Modbus messaging on TCP/IP implementation guide V1.0b*.
- MODICON. (1996). *Modicon Modbus Protocol Reference Guide*. https://www.modbus.org/docs/PI_MBUS_300.pdf
- Mosconi, F. (2015). *The New European Industrial Policy*. London: Routledge. <https://doi.org/doi.org/10.4324/9781315761756>
- Muñoz Díaz, C. A. (05 de 06 de 2021). *Diseño y simulación de un sistema de monitoreo para un proceso de control de temperatura mediante PLC e IoT para la empresa Educatia* [Tesis, Universidad Antonio Nariño]. Repositorio Institucional, Colombia. <http://repositorio.uan.edu.co/handle/123456789/4958>
- Pymodbus. (21 de Jul de 2023). *Pymodbus documentation*. <https://pypi.org/project/pymodbus/>
- Python. (2022). Obtenido de <https://docs.python.org/es/3/tutorial/>
- Raspberry Pi. (2023). *Raspberry Pi Documentation*. <https://www.raspberrypi.com/documentation/>
- Rodriguez Cera, N. F. (2018). *Diseño scada para monitorear alarmas contra incendio del hospital regional de Lambayeque Chiclayo 2017* [Tesis, Universidad Cesar Vallejo]. Repositorio Institucional, Chiclayo, Perú. https://repositorio.ucv.edu.pe/bitstream/handle/20.500.12692/25916/Rodriguez_CNF.pdf?sequence=1&isAllowed=y
- Ruiz Olaya, A., Barandica López, A., & Guerrero Moreno, F. (2004, Febrero 24). Implementación de una Red MODBUS/TCP. Madrid, España. https://revistaingenieria.univalle.edu.co/index.php/ingenieria_y_competitividad/article/view/2277/3027



Sampieri, R. H., Collado, C. F., & Baptista, M. d. (2014). *Metodología de la investigación*. México: McGRAW-HILL.

Scheider Electric. (2023). *Modbus Plus*. www.se.com

SIEMENS. (2018). S7-1200 Programmable controlle. <https://www.siemens.com/>

Siemens TIAPortal. (2023). *Software en TIAPortal*. www.siemens.com

Syufrijal. (2019). Construction design system of constant pressure control in water distribution system with PID method using PLC based on IoT. *Journal of Physics: Conference Series*. <https://doi.org/10.1088/1742-6596/1402/2/022060>

TELTONIKA. (2023). *RUTX50 THE NEW EVOLUTION OF CONNECTIVITY*.
Obtenido de <https://teltonika-networks.com/>



ANEXOS

ANEXO 1. Elementos de control de presión constante

Tabla 16

Elementos de control de presión constante

No	Equipo	Marca/fabricante	Modelo	Características
1	Transmisor de presión	DANFOSS	060G1101	Alimentación: 10 – 30v DC Rango de medición: 0 – 6 Bar
2	Bomba eléctrica tipo: centrifuga	PENTAX	Ultra U5-120/3T	Potencia: 1.2HP Frecuencia: 60HZ
3	Presostato	DANFOSS	060-500266	Rango de medida: -0-20 – 8.00 bar
4	Válvula automática	DANFOSS	DF082H0161	Configurable mediante interruptores DIP Presión: máx. 10 bar
5	Sensor de nivel	KOBOLD	NKP	Máx. temperatura: 80°C / 175°F Máx. presión: 10 bar / 145 psi
6	Variador de velocidad	SCHNEIDER ELECTRIC	LTIVAR 12	Potencia: 1.5 kW Comando 2 hilos/3 hilos.



7	HMI	NATIONAL INSTRUMENTS	TPC-2230	Resolución Máxima: 1024 × 768 XGA. CPU: Intel Atom Processor E3827 1.75 GHz. Controlador S7- 1200, 14 entradas digitales, 2 entradas analógicas, 10 salidas digitales, 24VDC
8	PLC	SIEMENS	6ES7214- 1AG40- 0XB0	4 canales de entrada. 2 canales de salida Voltaje de operación: 690 V
9	MÓDULO DE ENTRADAS/SALIDAS ANALÓGICAS	SIEMENS	6ES7234- 4HE32- 0XB0	
10	GUARDAMOTOR	SCHNEIDER ELECTRIC	GV2ME10	Rango de Ajuste para disparo térmico: 4 – 6.3 A

Nota: adaptado de INDUCONTROL S.A.C (2020).

ANEXO 2. Planos eléctricos

Figura 77

Leyenda del módulo de control

SIMBOLO		DESCRIPCION
Q_01	Interruptor Termomagnético IQ (SCHNEIDER)	
Q_02	Interruptor Termomagnético IQ (SCHNEIDER)	
ID_01	Interruptor Diferencial (SCHNEIDER)	
GV_01	Guardamotor (SCHNEIDER)	
V_01	Variador de Velocidad (Altivar 312 - Schneider Electric)	
M	Motor 3Ø220AC 1.2HP PENTAX UFS-120	
57-1200	Controlador Siemens 6ES7212-1AE60-0XB0	
SM-1234	Modulo I/O Analógicas 6ES7234-4HE32-0XB0	
F_01	Fuente de Alimentacion Siemens 24V DC 5Amp	
SW_01	SWTCH TP-LINK ## # Maps	
HMI	Interfaz Hombre - Maquina (NI - TPC 2230)	
X_0	Borneras	

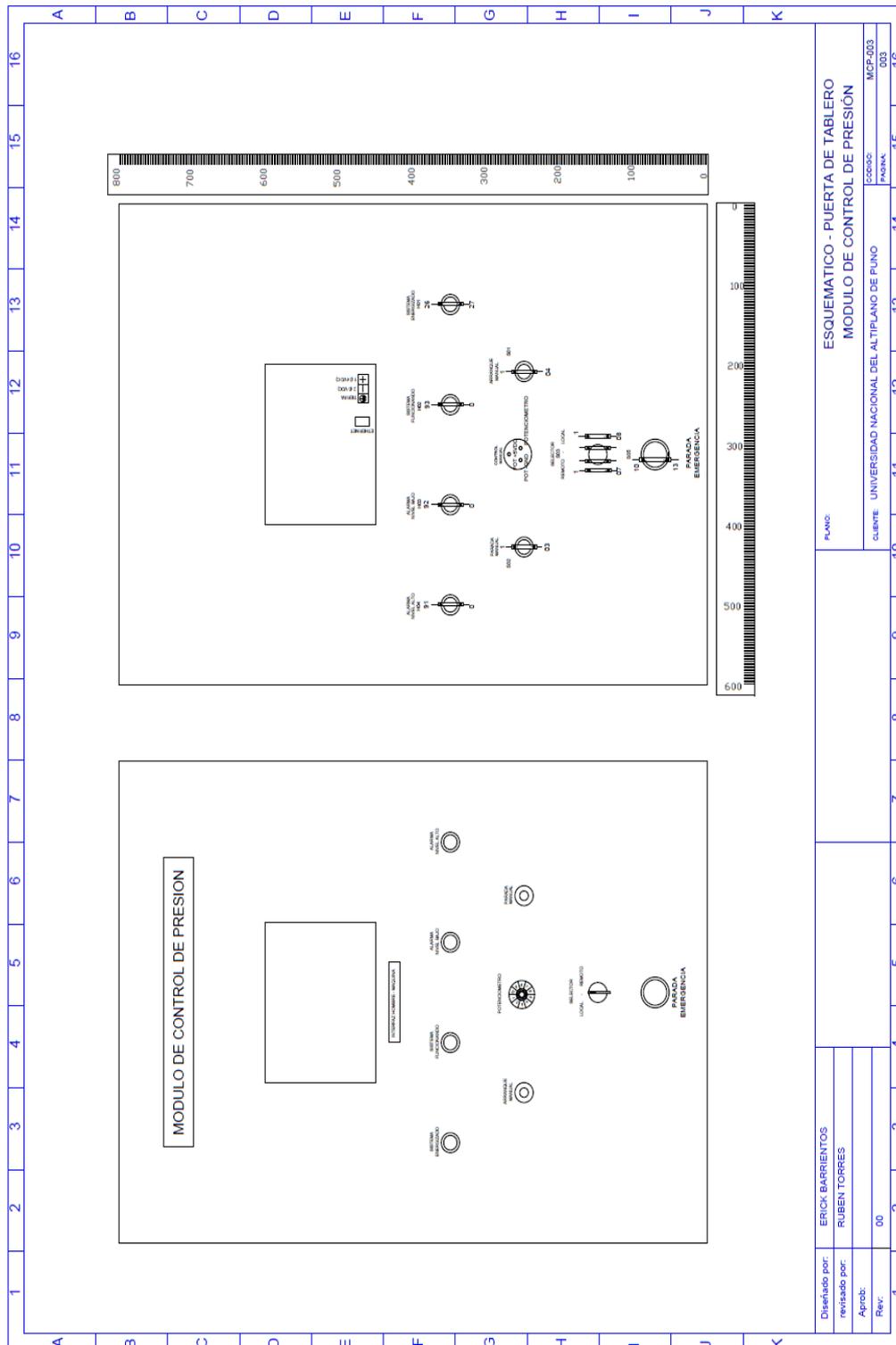
Diseñado por:		ERICK BARRIENTOS	
revisado por:		RUBEN TORRES	
Aprobi:			
Rev:		00	
1	2	3	4

LEYENDA	
MÓDULO DE CONTROL DE PRESIÓN	
CLIENTE:	UNIVERSIDAD NACIONAL DEL ALTIPLANO DE PUNO
ESCALA:	1:1
PROYECTO:	MCS-001
FECHA:	01/16

Nota: adaptado de INDUCONTROL S.A.C (2020).

Figura 79

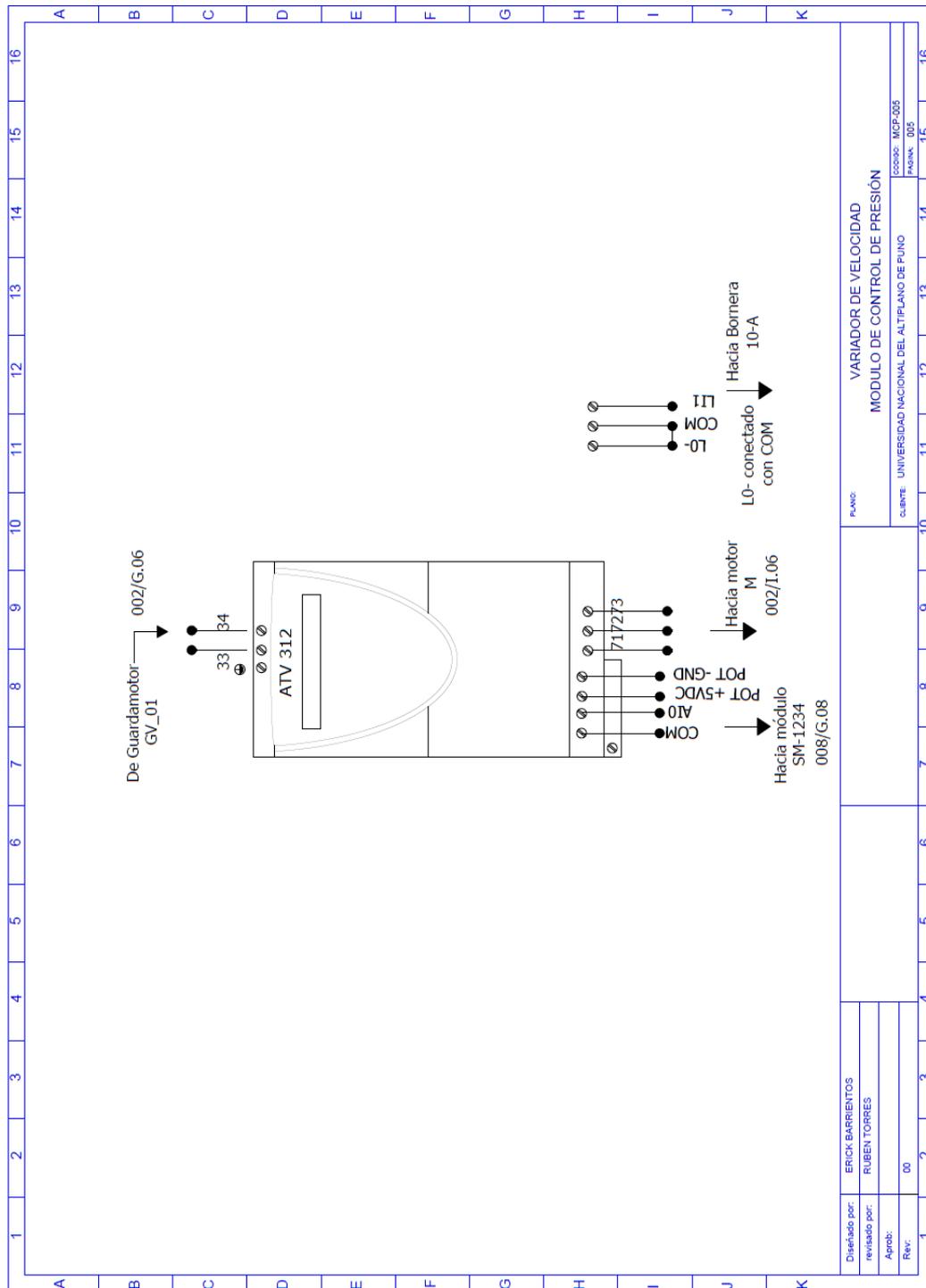
Esquemático - puerta del tablero



Nota: adaptado de INDUCONTROL S.A.C (2020).

Figura 81

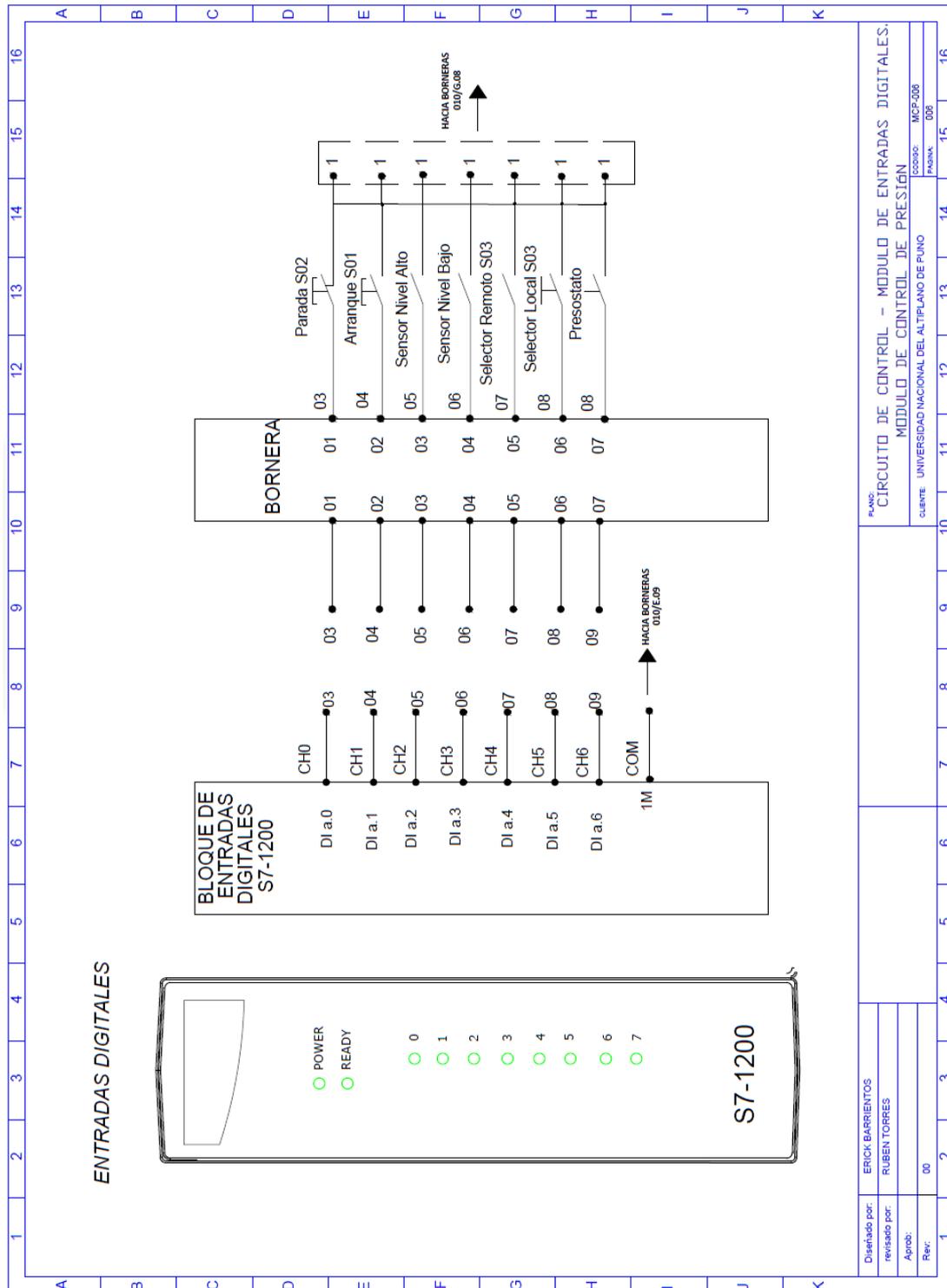
Variador de velocidad



Nota: adaptado de INDUCONTROL S.A.C (2020).

Figura 82

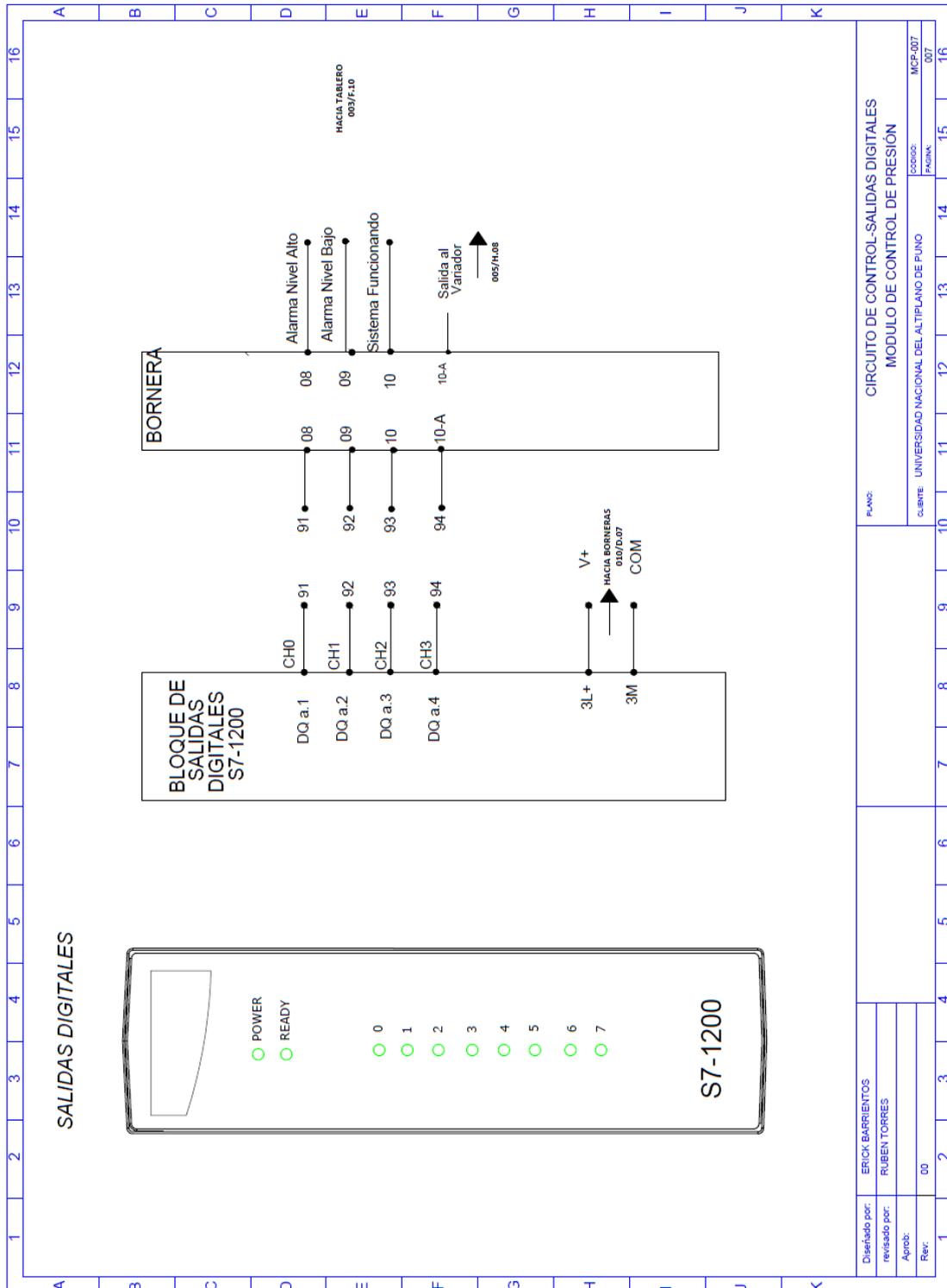
Módulo de entradas digitales



Nota: adaptado de INDUCONTROL S.A.C (2020).

Figura 83

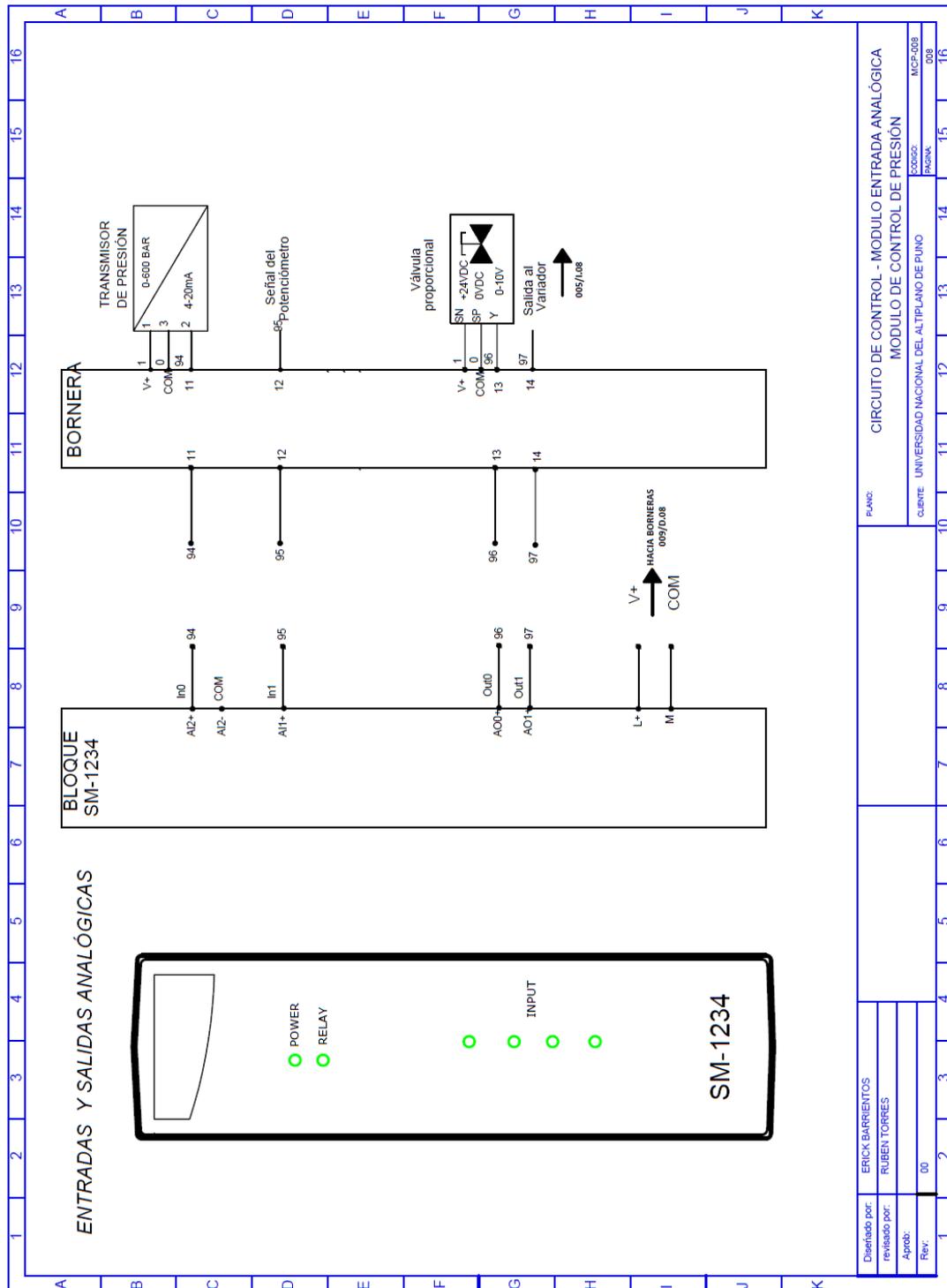
Salidas digitales



Nota: adaptado de INDUCONTROL S.A.C (2020).

Figura 84

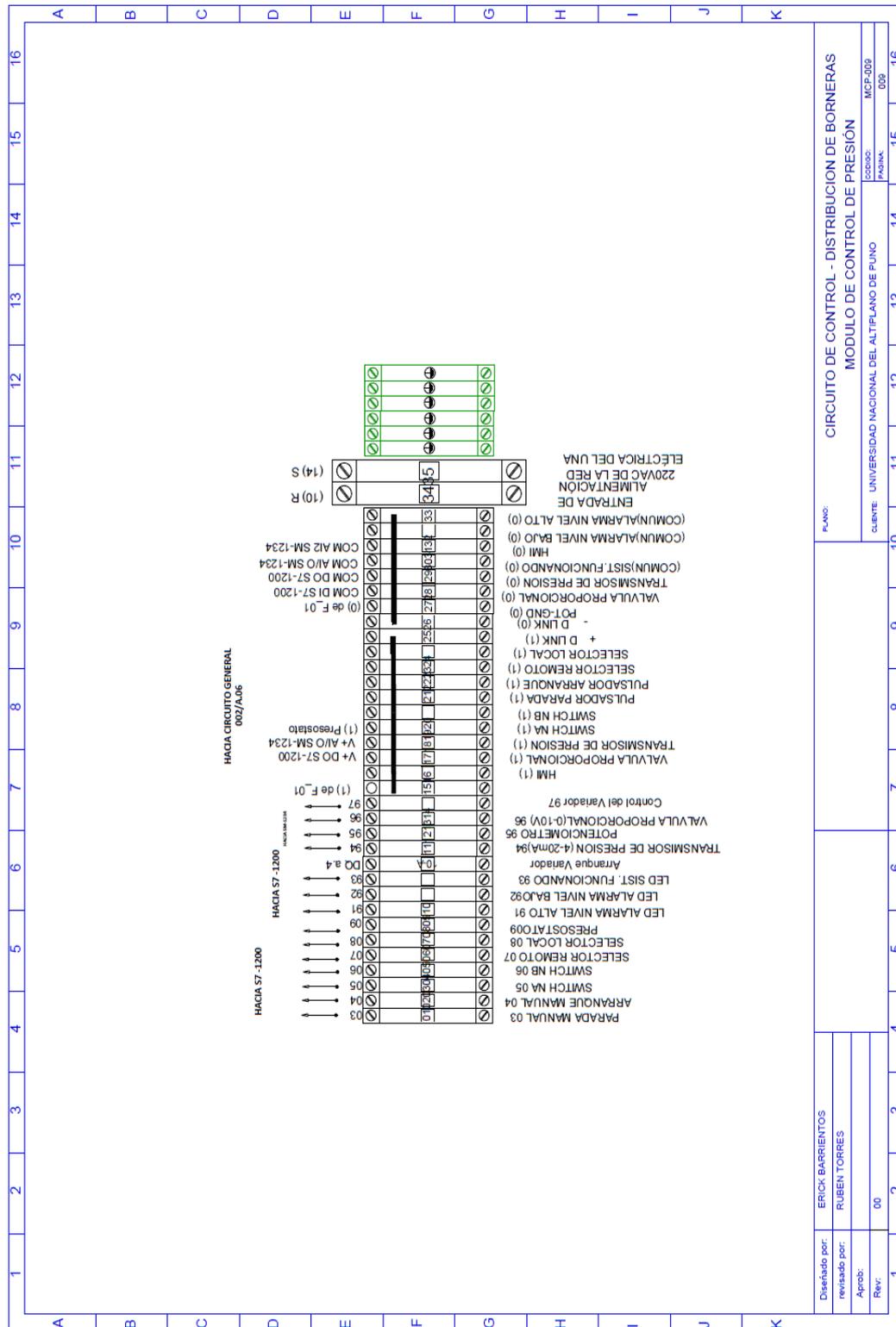
Módulo de entradas analógicas



Nota: adaptado de INDUCONTROL S.A.C (2020).

Figura 85

Distribución de borneras



Nota: adaptado de INDUCONTROL S.A.C (2020).



ANEXO 3. Hoja de especificación del TRB245

Tabla 17

Hoja de especificación del TRB245

Mobile	
<hr/>	
SIM switch	2 SIM cards, auto-switch cases: weak signal, data limit, SMS limit, on roaming, no network, network denied, data connection fail
Modulo Mobil	4G (LTE)
ETHERNET	
LAN	1 x LAN port 10/100 Mbps, (Configurable to work as WAN port)
Comunicación serial	
Modos	MODBUS RTU master, MODBUS gateway
Modbus TCP master	
Formato de datos admitidos	8 bit: INT, UINT; 16 bit: INT, UINT (MSB or LSB first); 32 bit: float, INT, UINT (ABCD (big-endian), DCBA (little-endian), CDAB, BADC), HEX, ASCII
MQTT GATEWAY	
Gateway	Allows sending commands and receiving data from MODBUS Master trough MQTT broker
DATA TO SERVER	



Protocolos	HTTP(S), MQTT, Azure MQTT, Kinesis
------------	------------------------------------

Nota. Los equipos teltonika pueden variar, esto dependerá de los productos.

ANEXO 4. Hoja de especificación del Raspberry PI

Tabla 18

Hoja de especificación del Raspberry PI

Procesador	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Memoria	1GB
Conectividad	<ul style="list-style-type: none"> • 2.4 GHz and 5 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 4.2, BLE • Gigabit Ethernet over USB 2.0
Multimedia:	H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
SD card	32GB
Entradas	<ul style="list-style-type: none"> • 5V/2.5A DC via micro-USB connector • 5V DC via GPIO header • Power over Ethernet (PoE)-enabled (requires separate PoE HAT)
Temperatura de funcionamiento	0-50°C

Nota: proporciona por las fuentes de Raspberry PI



ANEXO 5. Hoja de especificación de MateBook D15

Tabla 19

Hoja de especificación de MateBook D 15

MateBook D 15	
Procesador	AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GH
RAM	8.00 GB
Sistema	64 bits procesador x64
Windows	
Edición	Windows 11 Home Single Language
Versión	22H2

Nota: Proporcionada de la computadora personal.



ANEXO 6. Declaración jurada de autenticidad de tesis

 Universidad Nacional del Altiplano Puno  VRI Vicerrectorado de Investigación  Repositorio Institucional

DECLARACIÓN JURADA DE AUTENTICIDAD DE TESIS

Por el presente documento, Yo **Jamil Henry Sacaca Ramos**, identificado con DNI **62206509** en mi condición de egresado de:

Escuela Profesional, **Programa de Segunda Especialidad**, **Programa de Maestría o Doctorado**

informo que he elaborado el/la **Tesis** o **Trabajo de Investigación** denominada:

"DISEÑO DE UN MONITOREO REMOTO DE LA SALA DE CONTROL DE PRESIÓN CONSTANTE UTILIZANDO LA TECNOLOGÍA DE LA INDUSTRIA 4.0"

Es un tema original.

Declaro que el presente trabajo de tesis es elaborado por mí persona y **no existe plagio/copia** de ninguna naturaleza, en especial de otro documento de investigación (tesis, revista, texto, congreso, o similar) presentado por persona natural o jurídica alguna ante instituciones académicas, profesionales, de investigación o similares, en el país o en el extranjero.

Dejo constancia que las citas de otros autores han sido debidamente identificadas en el trabajo de investigación, por lo que no asumiré como tuyas las opiniones vertidas por terceros, ya sea de fuentes encontradas en medios escritos, digitales o Internet.

Asimismo, ratifico que soy plenamente consciente de todo el contenido de la tesis y asumo la responsabilidad de cualquier error u omisión en el documento, así como de las connotaciones éticas y legales involucradas.

En caso de incumplimiento de esta declaración, me someto a las disposiciones legales vigentes y a las sanciones correspondientes de igual forma me someto a las sanciones establecidas en las Directivas y otras normas internas, así como las que me alcancen del Código Civil y Normas Legales conexas por el incumplimiento del presente compromiso

Puno 24 de enero del 2024



Jamil Henry Sacaca Ramos
DNI: 62206509



Huella



ANEXO 7. Autorización para el depósito de tesis en el Repositorio Institucional

 Universidad Nacional del Altiplano Puno  Vicerrectorado de Investigación  Repositorio Institucional

AUTORIZACIÓN PARA EL DEPÓSITO DE TESIS O TRABAJO DE INVESTIGACIÓN EN EL REPOSITORIO INSTITUCIONAL

Por el presente documento, Yo **Jamil Henry Sacaca Ramos** identificado, con DNI **62206509** en mi condición de egresado de:

Escuela Profesional, Programa de Segunda Especialidad, Programa de Maestría o Doctorado

de **Ingeniería Electrónica**, informo que he elaborado el/la Tesis o Trabajo de Investigación denominada:

"DISEÑO DE UN MONITOREO REMOTO DE LA SALA DE CONTROL DE PRESIÓN CONSTANTE UTILIZANDO LA TECNOLOGÍA DE LA INDUSTRIA 4.0"

para la obtención de Grado, Título Profesional o Segunda Especialidad.

Por medio del presente documento, afirmo y garantizo ser el legítimo, único y exclusivo titular de todos los derechos de propiedad intelectual sobre los documentos arriba mencionados, las obras, los contenidos, los productos y/o las creaciones en general (en adelante, los "Contenidos") que serán incluidos en el repositorio institucional de la Universidad Nacional del Altiplano de Puno.

También, doy seguridad de que los contenidos entregados se encuentran libres de toda contraseña, restricción o medida tecnológica de protección, con la finalidad de permitir que se puedan leer, descargar, reproducir, distribuir, imprimir, buscar y enlazar los textos completos, sin limitación alguna.

Autorizo a la Universidad Nacional del Altiplano de Puno a publicar los Contenidos en el Repositorio Institucional y, en consecuencia, en el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto, sobre la base de lo establecido en la Ley N° 30035, sus normas reglamentarias, modificatorias, sustitutorias y conexas, y de acuerdo con las políticas de acceso abierto que la Universidad aplique en relación con sus Repositorios Institucionales. Autorizo expresamente toda consulta y uso de los Contenidos, por parte de cualquier persona, por el tiempo de duración de los derechos patrimoniales de autor y derechos conexas, a título gratuito y a nivel mundial.

En consecuencia, la Universidad tendrá la posibilidad de divulgar y difundir los Contenidos, de manera total o parcial, sin limitación alguna y sin derecho a pago de contraprestación, remuneración ni regalía alguna a favor mío; en los medios, canales y plataformas que la Universidad y/o el Estado de la República del Perú determinen, a nivel mundial, sin restricción geográfica alguna y de manera indefinida, pudiendo crear y/o extraer los metadatos sobre los Contenidos, e incluir los Contenidos en los índices y buscadores que estimen necesarios para promover su difusión.

Autorizo que los Contenidos sean puestos a disposición del público a través de la siguiente licencia:

Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional. Para ver una copia de esta licencia, visita: <https://creativecommons.org/licenses/by-nc-sa/4.0/>

En señal de conformidad, suscribo el presente documento.

Puno 24 de enero del 2024



Jamil Henry Sacaca Ramos
DNI: 62206509


Huella