



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

Facultad de Ciencias de la Electrónica

Licenciatura en Electrónica

**DISEÑO DE SISTEMA SCADA PARA
CONTROLAR Y MONITOREAR UN
PROCESO INDUSTRIAL A TRAVÉS DE
GRAFANA MEDIANTE UN PLC S7 1215C
DC/DC/DC**

Tesis presentada para obtener el grado de:

Licenciado en Electrónica

Presenta:

José Antonio Jiménez Sánchez

Asesor - Director:

M. en C. Marco Antonio Sobrevilla González



Puebla, Pue. Octubre 2024

AGRADECIMIENTOS

A mis Padres al Dr. Antonio Jiménez Gutiérrez y a la Lic. Janeth Sánchez Salinas, con todo mi amor y gratitud les dedico este trabajo, por el inalcanzable sacrificio y apoyo que me han brindado toda la vida, este trabajo es suyo como mío porque esto lo hicimos en equipo y gracias a ustedes he logrado este objetivo. Ustedes son mi inspiración más grande.

A mis hermanos Pedro y Nisaguie porque fueron parte de este equipo, sus sacrificios y apoyo fueron parte esencial para obtener este logro.

A mi asesor el Maestro en Ciencias Marco Antonio Sobrevilla González por su excepcional capacidad científica y tecnológica, por el apoyo brindado de manera comprometida en este proceso y por su entusiasmo y motivación hacia mi persona.

A mi abuelo Pedro Jiménez Orozco y a mi abuela Venancia Gutiérrez López por sus consejos y sus palabras de aliento hacia mi persona, así como su cariño brindado durante mi infancia y por su insistencia en titularme por tesis.

A mis abuelos maternos José Luis y María Lusmi por siempre estar al pendiente de mi persona, como también por todo lo que me han dado desde mi infancia.

A la Arquitecta Andrea Joucelyn de Aquino Cuaya por su apoyo y cariño incondicional, por confiar en mí y por motivarme a ser mejor persona.

A mis amigos en especial al Lic. Orlando Rodríguez Zárate, por las experiencias vividas durante la universidad, por su apoyo académico, por su motivación y consejos en esta bonita etapa.

A mi jurado de tesis: al Maestro Juan Carlos Rojas Bravo, al Doctor Osvaldo López Hernández y al Maestro Jesús Tuxpan Meneses. Por sus valiosas observaciones y recomendaciones que han mejorado la calidad de mi trabajo y ha enriquecido mi formación académica.

RESUMEN

La Automática es una ciencia lo cual estudia la automatización y sus aplicaciones tales como aquellas que tienen un fin de reducir y eliminar la intervención humana en un proceso de producción industrial.

Es aquí donde un sistema SCADA (Supervisory Control and Data Acquisition) entra en el proyecto ya que permite la adquisición de datos a través de los diferentes procesos en un entorno industrial lo cual permite el control y la supervisión de forma local o remota en tiempo real para proporcionar a los operadores una visión completa del proceso, permitiéndoles tomar decisiones informadas y controlar el sistema de manera más eficaz.

La tesis elaborada en este documento plantea el diseño de un sistema SCADA basado en el monitoreo de datos a través del software de GRAFANA mediante un PLC S7 1215C DC/DC/DC a través de un proceso industrial diseñado en Factory IO el cual a través de su entorno se utilizó para el control y la supervisión del proceso industrial mediante diferentes dispositivos tales como computadores, pantallas y dispositivos móviles.

Palabras Clave:

Sistema SCADA
Factory IO
GRAFANA
PLC

ÍNDICE GENERAL

AGRADECIMIENTOS.....	II
RESUMEN.....	III
ÍNDICE DE FIGURAS.....	VII
CAPÍTULO I: INTRODUCCIÓN.....	1
Introducción.....	2
Justificación.....	3
Objetivos.....	4
General:.....	4
Específicos:.....	4
Alcance.....	5
CAPÍTULO II: MARCO TEÓRICO.....	6
2.1 Automatización Industrial.....	7
2.11 Grados de la automatización industrial.....	7
2.12 La pirámide de la automatización.....	8
2.2 Sistemas SCADA.....	9
2.21 Componentes de un sistema SCADA:.....	10
2.22 Arquitectura y funcionamiento general de un sistema SCADA.....	11
2.3 Tecnología Operativa.....	11
2.31 Controlador Lógico Programable (PLC).....	12
Funciones básicas del PLC.....	12
2.32 Variador de frecuencia PowerFlex 4.....	13
Características del variador de frecuencia PowerFlex 4:.....	13
Ventajas de utilizar el variador de frecuencia PowerFlex 4:.....	15
Aplicaciones de los variadores de frecuencia Power Flex4:.....	15
2.4 Factory IO.....	16
2.5 Protocolos de Comunicación.....	16
2.51 Modbus.....	16

Modelo OSI para el protocolo Modbus	17
2.52 RS-485	18
2.53 Comunicación HTTP	19
2.6 Node-RED	19
2.7 Tecnologías de la Información	20
2.71 Grafana	20
2.72 InfluxDB	20
CAPÍTULO III: SISTEMA SCADA.....	21
3.1 Proceso industrial en Factory I/O	22
3.11 Paletizado	24
3.12 Elevador.....	26
3.13 Modbus RTU	27
3.14 Tanque de suministro de agua	28
3.2 Programación del PLC	28
3.21 Funciones de comunicación	30
3.22 Funciones de proceso	30
3.23 Funciones de procesamiento de datos	31
3.3 Node-RED	33
3.31 Conexión entre Node-RED y el PLC	34
3.32 Procesamiento de datos dentro de Node-RED	36
3.33 Conexión entre Node-RED e InfluxDB.....	40
3.4 InfluxDB.....	43
3.5 Grafana	46
3.51 Conexión entre Grafana e InfluxDB.....	46
3.52 Creación del dashboard en Grafana.....	48
3.6 Telegram.....	50
3.61 Conexión entre Telegram y Node-RED	51
CAPÍTULO IV: IMPLEMENTACIÓN	56
Funcionamiento del sistema SCADA.....	57
Implementación física.....	57
Prueba I: Desempeño normal del sistema.....	58
Prueba II: Paro de emergencia.....	62
Prueba III: Fallas en el variador de frecuencia	63
Prueba IV: Control del proceso desde Telegram.....	66

Prueba V: Reporte de funcionamiento del sistema SCADA	69
Ventajas del Proyecto	74
Desventajas del Proyecto.....	75
CONCLUSIONES.....	76
Importancia del Proyecto para el Mercado Industrial.	77
BIBLIOGRAFÍAS.....	78
ANEXOS	79
ANEXO 1	80
ANEXO 2	85
ANEXO 3	88

ÍNDICE DE FIGURAS

Figura 2.1. Barrientos, A y Gambao. (2014). La pirámide de la automatización.	8
Figura 2.2. Noblejas, D. (septiembre 2021). ¿Qué es un SCADA?	10
Control de supervisión y adquisición de datos. Nunsys. https://www.nunsys.com/scada/ ..	10
Figura 2.3. PLC s7 1215C DC/DC/DC.	12
Figura 2.4. Variador de Frecuencia PowerFlex 4.	13
Figura 2.5. Tabla de Datos del VFD PowerFlex 4.	14
Figura 2.6. Ejemplo de una red Modbus.	17
Figura 3.1. Sensores y actuadores conectados al PLC por medio de Factory IO.	22
Figura 3.2. Proceso industrial diseñado en Factory I/O.	23
Figura 3.3. Tablero de control del proceso industrial.	24
Figura 3.4. Subproceso de paletizado (vista frontal).	24
Figura 3.5. Subproceso de paletizado (vista posterior).	25
Figura 3.6. Acomodo de las seis cajas sobre un palet.	25
Figura 3.7. Organizado de cajas para el paletizado.	26
Figura 3.8. Subproceso del elevador.	27
Figura 3.9. Variador de frecuencia PowerFlex4 dentro del subproceso del elevador.	27
Figura 3.10. Subproceso del tanque de suministro de agua.	28
Figura 3.11. Diagrama de flujo del bloque de organización Main (OB1) del proceso industrial.	29
Figura 3.12. Bloque de datos (DB500) para enlazar a Node-RED.	33
Figura 3.13. Nodos de la librería node-red-contrib-s7.	34
Figura 3.14. Primera parte de la configuración del nodo s7 in.	35
Figura 3.15. Segunda parte de la configuración del nodo s7 in.	35
Figura 3.16. Configuración del nodo función para cada variable de lectura discreta.	36
Figura 3.17. Configuración del nodo intercambio para cada variable de lectura.	37
Figura 3.18. Configuración del nodo función para la velocidad del variador.	38
Figura 3.19. Configuración del nodo función para el nivel de agua del tanque.	39

Figura 3.20. Configuración del nodo función para el número de falla en el variador.....	39
Figura 3.21. Primera parte de la configuración del nodo influxdb out.....	40
Figura 3.22. Segunda parte de la configuración del nodo influxdb out.....	41
Figura 3.23. Red de conexión de variables discretas en Node-RED.....	42
Figura 3.24. Red de conexión de variables enteras en Node-RED.....	42
Figura 3.25. Creación de cuenta en InfluxDB Cloud 2.0.	43
Figura 3.26. Configuración de la base datos en InfluxDB Cloud 2.0.....	43
Figura 3.27. Creación del Bucket en InfluxDB.	44
Figura 3.28. Generación del token en InfluxDB para enlace externo.	44
Figura 3.29. Base de datos en InfluxDB funcionando correctamente.	45
Figura 3.30. Entorno de desarrollo de dashboards en Grafana.....	46
Figura 3.31. Establecer una conexión con InfluxDB en Grafana.	47
Figura 3.32. Primera parte de la configuración de la base de datos InfluxDB en Grafana. .	47
Figura 3.33. Segunda parte de la configuración de la base de datos InfluxDB en Grafana.	48
Figura 3.34. Ejemplo de la configuración de una visualización en Grafana.	48
Figura 3.35. Ejemplo de la descripción de una visualización en Grafana.....	49
Figura 3.36. Dashboard diseñado en Grafana para monitorear el proceso industrial.....	49
Figura 3.37. Creación del bot y el chat desde Telegram.	51
Figura 3.38. Interfaz del chat creado en Telegram para el control a distancia.	51
Figura 3.39. Nodos de la librería node-red-contrib-telegrambot.....	52
Figura 3.40. Primera parte de la configuración del nodo Command en Node-RED.....	53
Figura 3.41. Segunda parte de la configuración del nodo Command en Node-RED.....	53
Figura 3.42. Configuración del nodo de intercambio de datos que permite tres salidas.	54
Figura 3.43. Red de nodos para controlar el proceso mediante Telegram.	55
Figura 4.1. Diagrama a bloques del sistema SCADA diseñado.	57
Figura 4.2. Sistema SCADA implementado físicamente.	58
Figura 4.3. Proceso industrial en marcha manual desde Factory I/O.	59
Figura 4.4. Subproceso de paletizado en funcionamiento.	59
Figura 4.5. Subproceso del tanque de suministro de agua en funcionamiento.....	59
Figura 4.6. Subproceso del elevador y entrega de palet en funcionamiento.	60
Figura 4.7. Dashboard en Grafana con el sistema en marcha.....	61
Figura 4.8. Sistema SCADA físico en funcionamiento normal.	62
Figura 4.9. Paro de emergencia del proceso industrial.....	62
Figura 4.10. Alerta de paro de emergencia del proceso en Grafana.....	63

Figura 4.11. Falla en la alimentación del variador de frecuencia.....	64
Figura 4.12. Alerta de falla de alimentación del variador en Grafana.....	64
Figura 4.13. Falla en la comunicación Modbus RTU del variador de frecuencia.....	65
Figura 4.14. Alerta de falla de comunicación del variador en Grafana.....	65
Figura 4.15. Proceso industrial en modo automático para control a distancia.	66
Figura 4.16. Comando de marcha y reset enviado desde Telegram.	67
Figura 4.17. Sistema SCADA en funcionamiento controlado de manera remota.....	67
Figura 4.18. Comando de paro enviado desde Telegram.	68
Figura 4.19. Monitoreo del proceso después de un largo periodo de funcionamiento.....	68
Figura 4.20. Creación del reporte del proceso en Grafana	69
Figura 4.21. Rango de tiempo del reporte de datos en Grafana.	69
Figura 4.22. Formato del reporte en Grafana.	70
Figura 4.23. Cronograma del reporte en Grafana.	70
Figura 4.24. Elección del destino del reporte en Grafana.	71
Figura 4.25. Envío del reporte del proceso en Grafana.....	71
Figura 4.26. Recepción del reporte de Grafana vía correo electrónico.	72
Figura 4.27. Creación del reporte de almacenamiento de datos en InfluxDB.....	72
Figura 4.28. Ejemplo de reporte de datos generado por InfluxDB.....	73
Figura A1. Tabla de variables de entrada del PLC.....	82
Figura A2. Tabla de variables de salida del PLC.	84
Figura A3. Tabla de comandos de escritura en el registro 8192 de PowerFlex4 vía Modbus.(?).....	85
Figura A4. Tabla de comando de escritura en el registro 8193 de PowerFlex4 vía Modbus.(?).....	86
Figura A5. Tabla de comandos de lectura en el registro 8448 de PowerFlex4 vía Modbus.(?).....	86
Figura A6. Tabla de comando de lectura en el registro 8451 de PowerFlex4 vía Modbus.(?)	86
Figura A7. Tabla de comandos de lectura en el registro 8449 de PowerFlex4 vía Modbus.(?).....	87
Figura A8. Dashboard en Grafana cuando la eficiencia del proceso está en una etapa transitoria.....	88
Figura A9. Dashboard en Grafana durante un paro de emergencia del sistema.....	89
Figura A10. Dashboard en Grafana durante una falla en el variador de frecuencia.....	90
Figura A11. Dashboard en Grafana cuando la eficiencia del proceso está en una etapa estacionaria.	91

CAPÍTULO I:

INTRODUCCIÓN

Introducción

A nivel mundial, el sector industrial desempeña un papel más importante, ya que genera innovación, crecimiento y desarrollo social constante. Por lo que cada vez la competencia es mayor en la cual las empresas demandan productos de alta calidad con un tiempo de producción menor. Y solo aquellos proyectos que sean eficientes serán capaces de ser competitivos y revolucionar los sistemas de producción conforme lo demanda la industria 4.0 que a través de la implementación de tecnologías avanzadas como lo es la automatización, el internet industrial de las cosas (IIOT) y el análisis de datos, permite a las empresas el desarrollo de un conjunto de posibilidades para la supervisión y el control de las plantas de producción y así poder optimizar procesos, reducir costos y aumentar la eficiencia.

En este entorno, la importancia de un sistema SCADA (Supervisory Control and Data Acquisition) desempeña un papel crucial al adaptarse a cualquiera de las necesidades en las empresas para supervisar y controlar el funcionamiento de instalaciones remotas a través de la recopilación, transmisión y análisis de datos en tiempo real.

El proyecto de tesis radica en optimizar el monitoreo de las variables del sistema SCADA utilizando un software de visualización y análisis de datos como lo es Grafana, con el objetivo de sustituir o complementar el software y hardware que se utilizan comúnmente en los sistemas SCADA para dicha tarea. Además, Grafana maneja un ambiente de código abierto que permite adaptarse a cualquier entorno sin importar la marca de los dispositivos involucrados en la parte operativa.

En cuanto a la estructura del sistema SCADA desarrollado, se utilizó un PLC (Controlador Lógico Programable) de la marca SIEMENS que es el encargado de controlar todo el proceso industrial, tomando los datos que serán monitoreados en tiempo real utilizando diferentes protocolos de comunicación de acuerdo con los dispositivos y programas empleados.

Justificación

El internet industrial de las cosas juega un papel importante dado que, a través de la interconexión de dispositivos, sistemas y entornos industriales, su infraestructura permite la recopilación y el intercambio de datos a nivel industrial. Por otra parte, la supervisión y el control desempeñan un gran papel en la optimización de procesos industriales. Sin embargo, la tecnología evoluciona constantemente y eso exige que los sistemas SCADA sean más flexibles, más sofisticados y menos costosos, es aquí en donde entra GRAFANA y juega un papel muy importante.

Uno de los equipos más importantes en la automatización de procesos industriales son las HMI (Human Machine Interface), que constituyen la interacción entre la máquina y el hombre a través de una interfaz, los cuales son muy utilizados para monitorear los datos en tiempo real de la implementación de cualquier proyecto. Sin embargo, las pequeñas y medianas empresas no cuentan con suficientes fondos de inversión para comprar dichos equipos e invertir en el software necesario, ya que suelen ser muy costosos, es por eso que al utilizar Grafana reducirá notablemente el costo del proyecto sin la necesidad de una gran inversión puesto que al ser código abierto facilitara el uso de licencias.

Por otra parte, la industria moderna depende mucho de la eficiencia operativa, por lo cual la toma de decisiones basadas en datos es importante para reducir costos operativos, costos de mantenimiento y maximizar la productividad. Al igual que este proyecto está abierto a cualquier proceso industrial, pero para la monitorización de datos se empleará un entorno en el software de Factory IO.

Por último, el acceso remoto es sumamente importante en la industria, específicamente en la operación, la supervisión y casos de emergencia de procesos industriales, permite que los ingenieros y los operadores puedan acceder a los datos en tiempo real desde ubicaciones remotas, lo cual facilita la detección temprana de problemas en equipos, maquinaria y mantenimiento preventivo lo cual ayuda a reducir el tiempo de inactividad no planificado y a prolongar la vida útil de los componentes.

Considerando lo escrito anteriormente, la justificación de este proyecto es diseñar un sistema SCADA el cual le permita a las pequeñas y medianas empresas monitorear y controlar sus procesos industriales a un menor costo.

Objetivos

General:

Diseñar un sistema SCADA mediante el software de GRAFANA para monitorear un proceso Industrial desarrollado en Factory IO empleando un Variador de Frecuencia que estará controlado por un PLC físico S7-1215C con la finalidad de optimizar los recursos de una pequeña o mediana empresa, mejorar su eficiencia operativa y permitir un acceso remoto al proceso.

Específicos:

1. Diseñar un sistema SCADA para controlar y monitorear un proceso industrial a través de Grafana utilizando un PLC S7 1215 C DC/DC/DC y un variador de frecuencia Power Flex 4.
2. Implementar una red de comunicación industrial entre un PLC S7-1215C DC/DC/DC y un variador de frecuencia Power Flex 4 utilizando el protocolo de comunicación industrial Modbus para el envío de datos a la nube.
3. Implementar el sistema SCADA utilizando diferentes dispositivos conectados simultáneamente a la red de comunicación industrial, que proporcione ayuda al usuario y también sea controlado mediante acceso remoto.
4. Permitir la obtención de datos del proceso industrial en lapsos definidos de tiempo para el estudio y toma de decisiones que mejoren los procesos industriales de las pequeñas y medianas industrias.

Alcance

Teniendo presente que el objetivo general de este proyecto es el diseño de un sistema SCADA para el monitoreo de un proceso industrial en un área de aplicación enfocada a las pequeñas y medianas empresas que no cuentan con los recursos económicos suficientes para invertir en software y hardware.

La implementación del proyecto se ejemplifica en el entorno del software de Factory IO mediante el desarrollo de un escenario similar al de una pequeña o mediana empresa para la supervisión, es necesario que el sistema SCADA tiene la capacidad de adaptación y escalabilidad para ser aplicado en diversos procesos industriales.

En este proyecto se estudiarán conceptos técnicos, procedimientos y configuraciones que serán necesarias para comprender la metodología a seguir. Sin embargo, la implementación de las redes industriales necesita aplicar técnicas para su protección, las cuales para este proyecto no se tomarán en cuenta.

CAPÍTULO II:

MARCO TEÓRICO

2.1 Automatización Industrial

De acuerdo con el libro Introducción a la a Automatización Industrial de los autores Alberto Brunete, Pablo San Segundo y Rebeca Herrero: La automatización industrial se define como la aplicación de la automática a un proceso o a un dispositivo industrial.

Los objetivos de la automatización son:

- Incrementar la productividad y flexibilizar las herramientas
- Producir con calidad constante
- Dedicar a los humanos tareas creativas

Por otra parte, la automatización industrial ha transformado de manera significativa a los procesos de manufactura y producción permitiendo una mayor eficiencia y precisión en las operaciones industriales.

Según el libro de Groover (2007), la automatización industrial se refiere al “uso de sistemas de control, como computadoras o robots, y tecnologías de información para manejar procesos y maquinaria en una industria, reemplazando así el trabajo humano en la mayoría de las tareas” (p.1)

La automatización industrial ha evolucionado la producción y la manufactura en los procesos industriales para proporcionar innovación a la industria. Sin embargo, presenta grandes desafíos a medida que la tecnología avanza y se integra con la IIOT para transformar la industria y los proyectos sean más adaptables e inteligentes ajustándose a diversas necesidades.

2.11 Grados de la automatización industrial

De acuerdo con la información obtenida de Barrientos A. y Gambao (2014) la automatización está dividida en 2 ramas:

Automatización rígida: es aquella en la que la secuencia de las operaciones está fijada por la configuración de los equipos utilizados. La automatización rígida suele ser rentable en la producción de grandes series de productos iguales y se suele asociar a una distribución basada en el producto.

Automatización flexible: también denominada programable, permite cambiar el producto fabricado mediante un cambio en las ordenes de control del sistema, pero con alteraciones mínimas o nulas en el sistema de fabricación.

2.12 La pirámide de la automatización

Según el libro de Barrientos A. y Gambao (2014) la pirámide de la automatización es una representación de los diferentes niveles de automatización en una fábrica y está dividida de la siguiente manera:

- **Nivel 1:** Barrientos A. y Gambao (2014). Incluye a los actuadores, sensores y otros elementos de hardware que forman una máquina como también hace énfasis a que en este nivel se adquieren datos del proceso.
- **Nivel 2:** Esta formado por las máquinas individuales que intervienen en el proceso productivo. En este nivel están los controladores lógicos programables, los sistemas de control numérico, los robots industriales, las computadoras industriales, los que se conocen como controladores de máquinas.

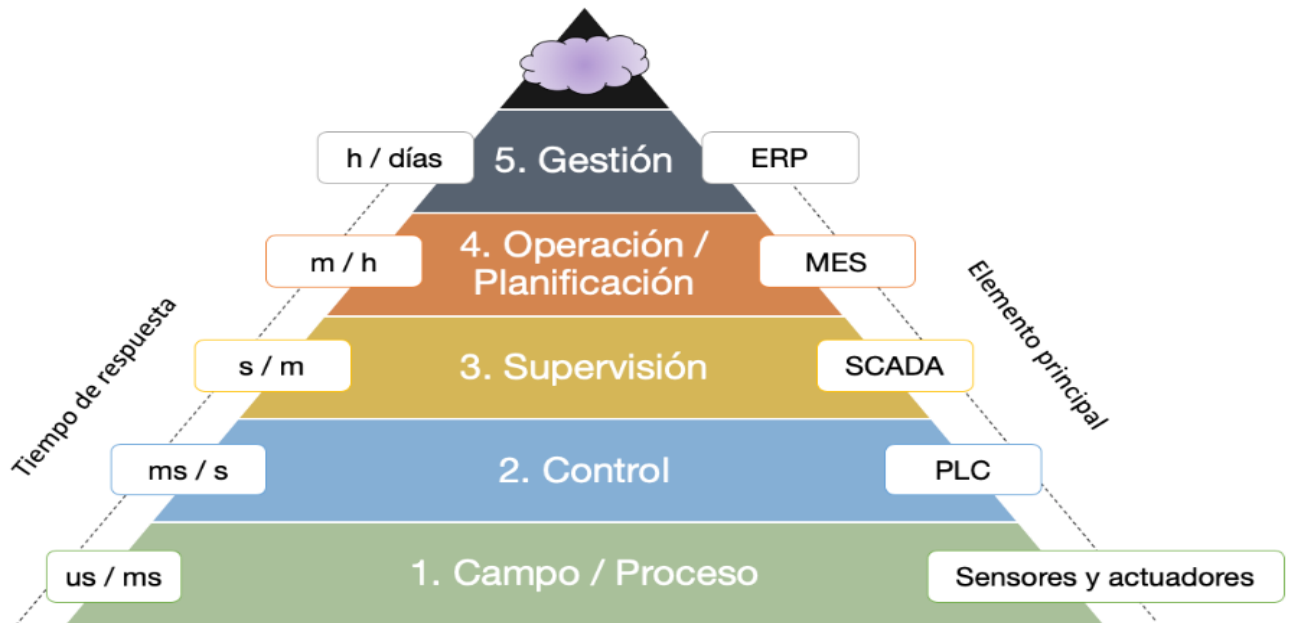


Figura 2.1. Barrientos, A y Gambao. (2014). La pirámide de la automatización.

- **Nivel 3:** En este nivel se encuentran los sistemas de control de supervisión y adquisición de datos (SCADA) así como los HMI. En esta capa los datos se supervisan a través de interfaces de usuario y se almacenan en bases de datos.
- **Nivel 4:** El conocido nivel de operación o de planificación ya que en este se monitoriza todo el proceso de fabricación en una planta o fabrica que va desde la materia prima hasta el producto concluido.
- **Nivel 5:** Por último, la cima de nivel conocida como el nivel de gestión es muy importante, ya que aquí la alta dirección de la empresa controla y controla sus

operaciones usando la tecnología de los niveles anteriores. Esto permite a la empresa monitorear todos los niveles del negocio.

2.2 Sistemas SCADA

Los sistemas SCADA son de suma importancia para la supervisión y control de procesos industriales a pequeña y a gran escala, mismos que permiten la recopilación y análisis de datos en tiempo real de diferentes procesos, lo que facilita la toma de decisiones informadas y respaldadas para mejorar el proceso industrial en diferentes entornos.

SCADA es el acrónimo de Supervisory Control and Data Acquisition. De acuerdo con Penin (2012) un sistema SCADA se refiere a un software que posibilita la supervisión y adquisición de datos de forma remota de cualquier proceso industrial. Además, debe posibilitar el control de dicho proceso.

Para diseñar un sistema SCADA, hay que considerar aspectos como la economía, la accesibilidad, el mantenimiento, la flexibilidad y el mantenimiento.

Los sistemas SCADA permitirán:

- El monitoreo continuo y en tiempo real de los procesos industriales. Lo cual ayudara a detectar problemas o fallas dentro del proceso y así poder responder rápidamente para resolver las diferentes anomalías.
- Automatizar el control de diversos procesos
- La gestión eficiente de recursos y materias primas contribuyendo a la reducción de costos operativos y a un uso más sostenible de los mismos.
- Los datos históricos y en tiempo real recopilados por el sistema permitirán implementar mejoras, tendencias y optimizar el rendimiento del sistema.
- El acceso remoto permitirá que los operadores puedan supervisar y controlar los procesos desde cualquier punto, mejorando la flexibilidad y la capacidad de respuesta.
- Podrán integrarse con otros sistemas de gestión al igual que pueden escalar conforme al crecimiento de las operaciones.
- El control preciso y continuo del proceso garantizará que los productos cumplan con los estándares de calidad, mejorando la satisfacción del cliente.

- Alertas visuales en caso de fallas.
- Garantizar la seguridad y acceso del sistema restringiendo aspectos de configuración, control, edición, y supervisión a usuarios no autorizados, incluso de ser necesario a cuartos de control.

Un sistema SCADA debe de ser escalable para así poder garantizar un correcto funcionamiento. Sin embargo, si el volumen de datos aumenta se pueden presentar problemas entorno al tráfico de datos ya que solo pasaría por un mismo punto, es por eso que al ser escalable puede estar en diferentes servidores a la vez.

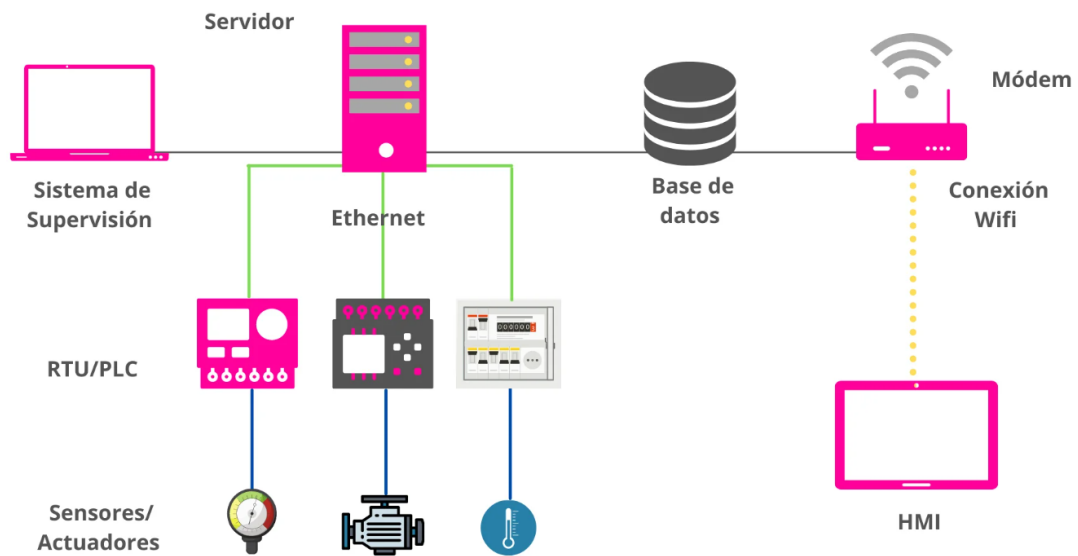


Figura 2.2. Noblejas, D. (septiembre 2021). ¿Qué es un SCADA? Control de supervisión y adquisición de datos. Nunsys. <https://www.nunsys.com/scada/>

2.21 Componentes de un sistema SCADA:

Un sistema SCADA se compone de diferentes módulos básicos los cuales son los siguientes:

- **Alarmas y Eventos:** Permiten al usuario operador ser alertado de errores o eventos inesperados del proceso industrial.

- **Comunicaciones:** Corresponde al software y hardware encargado de gestionar y controlar las comunicaciones.
- **Interfaz gráfica:** representa el dashboard y las múltiples pantallas de visualización y monitoreo de datos mediante múltiples imágenes.
- **Tendencias:** Permite representar la evolución y cambios en las variables de supervisión del sistema.
- **Registro:** Comprende funciones de almacenamiento temporal de datos de supervisión y control del proceso.
- **Configuración:** Define el entorno y las necesidades del proceso.

2.22 Arquitectura y funcionamiento general de un sistema SCADA.

1. Nivel de Campo:

- Sensores y actuadores
- Controladores Lógicos Programables (PLCs)
- Unidades Terminales Remotas (RTUs)

2. Nivel de Comunicación

- Redes de Comunicación
- Protocolos de comunicación

3. Nivel Empresarial

- Integración con Sistemas Empresariales

2.3 Tecnología Operativa

La tecnología operativa hace énfasis al hardware y software que detecta o causa cambios mediante el monitoreo y el control de sistemas industriales. Se centra en la administración y control de sistemas de infraestructura física como maquinaria, redes eléctricas entre otros. Su función principal es la recopilación de datos en tiempo real para la supervisión de los mismos y así detectar anomalías.

2.31 Controlador Lógico Programable (PLC)

Se entiende por Controlador Lógico Programable (PLC), a toda máquina electrónica diseñada para controlar en tiempo real y en un medio industrial procesos secuenciales.

El proyecto utiliza un PLC de la marca SIEMENS S7 1215c DC/DC/DC el cual según su manual: “ofrece la flexibilidad y potencia necesarias para controlar una gran variedad de dispositivos para las distintas necesidades de automatización. Gracias a su diseño compacto, configuración flexible y amplio juego de instrucciones” (p. 25).



Figura 2.3. PLC s7 1215C DC/DC/DC.

Funciones básicas del PLC

- La detección: lectura de la señal de los sensores distribuidos por el sistema de fabricación
- El mando: elabora y envía las acciones al sistema mediante los actuadores y preactuadores.
- El dialogo hombre-máquina: mantener un dialogo con los operarios de producción obedeciendo sus consignas e informando el estado del proceso.

- La programación: para introducir, elaborar y cambiar el programa de aplicación del PLC, el lenguaje de programación debe permitir modificar el programa incluso cuando el PLC se encuentra controlando el proceso industrial.

2.32 Variador de frecuencia PowerFlex 4

El variador de frecuencia PowerFlex 4 es un dispositivo utilizado para controlar la velocidad y funcionamiento de motores eléctricos.

Es fabricado por Allen Bradley, una marca de Rockwell Automation.

El PowerFlex 4 permite una amplia gama de control sobre la velocidad, la aceleración, la declaración y otros parámetros del motor. Algunas características de este variador incluyen:



Figura 2.4. Variador de Frecuencia PowerFlex 4.

Características del variador de frecuencia PowerFlex 4:



Figura 2.5. Tabla de Datos del VFD PowerFlex 4.

1. **Rango de Potencia:** El rango de potencia está disponible para soportar motores de 0.4 Kw o 0.5HP.
2. **Voltaje de Entrada:** Por estrategia se ocupó el modelo monofásico que va de 100-120 V AC.
3. **Frecuencia de Salida:** La frecuencia de salida que proporciona este variador va de 0 a 240 Hz.
4. **Voltaje de Salida:** El variador ofrece un voltaje de salida que va de 0 – 230 V
5. **Control del Motor:** Cuenta con la capacidad de un control V/Hz escalado como también algoritmos para un rendimiento optimo
6. **Interfaz y Programación:** El variador de frecuencia PowerFlex 4 incluye una interfaz que permite al usuario una programación sencilla de parámetros configurables.
7. **Tamaño Compacto:** Su diseño permite una fácil instalación en espacios reducidos.
8. **Funciones de Protección:** Cuenta con protección en contra de la sobrecarga del motor.
9. **Comunicaciones:** Contiene opciones de comunicación para conectarse a través de redes industriales tales como DeviceNet y Modbus RTU.
10. **Cumplimiento de Normas:** Cumple con múltiples estándares internacionales de seguridad y compatibilidad electromagnética.

El módulo de control del variador de frecuencia ejecuta todas las funciones por medio de un microprocesador que gestiona las rampas de aceleración y deceleración, los límites de

velocidad, los límites de corriente y otros datos de configuración que se definen utilizando el teclado integrado o mediante un dispositivo remoto como un PLC.

Ventajas de utilizar el variador de frecuencia PowerFlex 4:

La implementación de un variador de frecuencia en un motor eléctrico tiene las siguientes ventajas:

- Disminución en el consumo energético del motor.
- Prologar la vida útil de la maquinaria.
- Evita el alto consumo energético al iniciar el proceso de una máquina.
- Aumenta la precisión de control en diversos dispositivos como ventiladores, compresores etc.
- Disminuye los costos de mantenimiento y reparación de dispositivos.
- Reducción de estrés mecánico.
- Reducción de daño por cambios bruscos en cargar, pesos, flujos, etc.

Aplicaciones de los variadores de frecuencia Power Flex4:

- Los variadores de frecuencia o de velocidad tienen una amplia gama de aplicaciones industriales, por ejemplo:
- Bombas centrífugas: en este caso los variadores permiten un control ya sea de caudal determinado, de presión constante o de volumen variable.
- Ventiladores de aire acondicionado: aquí permite grandes ahorros de energía; se utilizan en extractores de aire, control de presurización, torres de enfriamiento, etc.
- Cintas o correas transportadoras: especialmente en el caso de procesos industriales donde las cintas llevan elementos que deben coordinarse con otras maquinarias.
- Extrusoras y prensas de tornillo: reemplazan sistemas hidráulicos tradicionales proporcionando una variación amplia de velocidad y control de torque.

- Separadores centrífugos: realizan un arranque suave y progresivo de la centrifuga evitando picos de corriente y velocidades de resonancia del sistema.
- Ascensores: permiten un arranque y parada suave del ascensor, pero manteniendo el torque, evitando que sufra movimientos bruscos.

2.4 Factory IO

Factory IO fue de vital importancia para desarrollar el proyecto ya que permitió crear y gestionar una fábrica con diferentes procesos industriales a través de su simulación en 3D. Es compatible con diferentes marcas y modelos de PLCs que en este caso al utilizar un PLC S7 1200 de la marca SIEMENS se pudo conectar al software de Factory IO.

Al desarrollar un entorno gráfico se pudo implementar el desarrollo de un entorno industrial de una pequeña o mediana empresa para así cumplir con el objetivo general de la tesis.

2.5 Protocolos de Comunicación

Los protocolos de comunicación son estándares que facilitan la transmisión de datos entre dispositivos de control y supervisión en entornos industriales. Estos protocolos son fundamentales para asegurar la eficiencia en la automatización de procesos.

2.5.1 Modbus

Modbus es un protocolo de comunicación serial desarrollado y publicado en el mercado industrial por Modicon en 1979. En su origen el uso de comunicación Modbus estaba direccionado exclusivamente al mando de los controladores lógicos programables de Modicon. Modbus es el protocolo de comunicación más utilizado en entornos industriales, sistemas de monitorización y telecontrol. La trama del protocolo Modbus se basa en la transmisión de información entre varios dispositivos electrónicos conectados a un mismo bus

de datos. Existiendo un solo dispositivo maestro (Master) y varios dispositivos esclavos (Slaves) conectados en red.

El protocolo Modbus es sumamente importante para el proyecto ya que es un protocolo de comunicación industrial de código abierto el cual está basado en el principio Maestro-Eslavo o Cliente-Servidor.

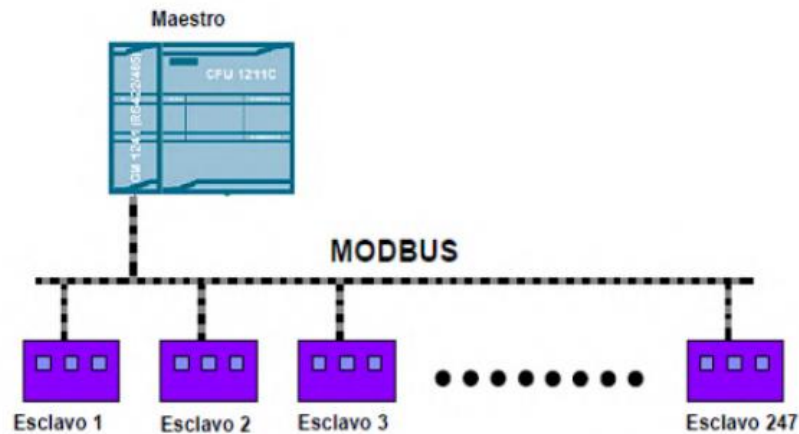


Figura 2.6. Ejemplo de una red Modbus.

Modelo OSI para el protocolo Modbus

El modelo de referencia OSI (Open Systems Interconnection) agrupa funciones de comunicación en siete niveles de proceso, mediante el cual los datos se empaquetan y se transmiten desde una aplicación emisora, viajando a través de medios físicos hasta llegar a una aplicación receptora. Es importante mencionar cuales son estos niveles ya que determina un panorama general en el cual se describen todas las capas. Y así determinar en que capa se encuentra modbus.

1. **Capa Física:** Esta capa toma los datos de la capa de enlace y los convierte en señales lógicas (1 y 0) que serán transmitidas a través de medios como el aire, cobre y el vacío.
2. **Capa de Enlace:** Facilita la comunicación entre dispositivos en una red local (LAN) mediante la conversión de datos en tramas y el uso de direcciones físicas del hardware.

3. **Capa Red:** Gestiona las direcciones y de hardware para determinar la ruta de envío de paquetes de un dispositivo a otro, asegurando que los paquetes lleguen al destino correcto.
4. **Capa Transporte:** Permite intercambiar datos entre sistemas finales, dividiendo el mensaje en varios fragmentos. El servicio de transporte puede ser orientado o no orientado a conexión, tomando en cuenta la unidad de transferencia máxima (MTU).
5. **Capa de Sesión:** Establece, gestiona y finaliza las sesiones de comunicación necesarias para la transmisión de datos, asegurando el control de inicio y fin de las mismas.
6. **Capa de Presentación:** Define el formato de los datos que se van a intercambiar entre las aplicaciones ofreciendo un conjunto de servicios para la transformación de estos.
7. **Capa de Aplicación:** Proporciona funciones y establece comunicaciones con las capas anteriores, definiendo los servicios que el dispositivo final necesitara para utilizar los datos según los requerimientos del usuario final.

Existen dos variantes del protocolo Modbus: ASCII y RTU; siendo este último escogido para las aplicaciones industriales por su eficiencia al ocupar un menor tiempo en el medio de transmisión y recepción. La modalidad ASCII básicamente usa caracteres ASCII para la codificación de sus mensajes con caracteres especiales al final de la trama, permitiendo al dispositivo conocer con certeza el final de esta, con esto se evita hacer vigilancia a las tramas con el uso de temporizadores. Esta modalidad es menos eficiente que RTU.

2.52 RS-485

Es un estándar de comunicaciones en bus para la capa física del modelo OSI y define las características eléctricas de los transmisores y receptores.

La transmisión es serial y asíncrona, lo cual quiere decir que los bits se van transmitiendo uno detrás de otro y sin una señal que sincronice al transmisor y al receptor. El medio físico es un par trenzado (A, B) que admite hasta 256 estaciones. La comunicación es semidúplex y se pueden cubrir hasta 1200 metros con un mismo bus sin pérdida de información gracias a la transmisión diferencial que cancela gran cantidad de ruido.

Las velocidades de transmisión oscilan entre los 300 y 19200 bit/s.

2.53 Comunicación HTTP

HTTP (Protocolo de Transferencia de Hipertexto) es el pilar fundamental para la transmisión de datos en la World Wide Web.

Este protocolo permite la petición de datos y recursos que funciona a través de la capa de aplicación del modelo OSI para poder permitir la comunicación entre un cliente y un servidor ya que principalmente su arquitectura se basa en ese modelo en donde el cliente envía solicitudes y el servidor responde con los recursos solicitados.

2.6 Node-RED

Node-RED es una innovadora herramienta que representa el desarrollo basado en flujos, diseñada para la integración de dispositivos, servicios y aplicaciones dentro ámbito del Internet de las Cosas (IoT). La cual fue desarrollada por IBM en 2013, Node-RED está fundamentada sobre Node.js, lo que le permite funcionar de manera eficiente en diversas plataformas y arquitecturas.

Un punto para considerar es la característica más sobresaliente de Node-RED, su interfaz gráfica la cual basada en web, permite a los usuarios crear aplicaciones mediante la simple acción de arrastrar y soltar nodos que representan diferentes funcionalidades. Estos nodos se conectan para formar flujos que definen cómo se procesan y transfieren los datos. Esta metodología esquematiza el desarrollo de aplicaciones complejas, ya que minimiza la necesidad de escribir código detallado.

2.7 Tecnologías de la Información

Las tecnologías de la información son sistemas electrónicos o tecnológicos que nos permiten intercambiar, transmitir y almacenar datos los cuales son esenciales para el desarrollo de proyectos como el que se hizo en esta tesis y para la sociedad en general.

2.71 Grafana

Grafana fue la interfaz de este proyecto es una herramienta de código abierto que ayudo a visualizar todos los datos y las variables principales del proceso industrial simulado en Factory IO.

Su aplicación abarca el monitoreo de sistemas, redes y aplicaciones, permitiendo a los usuarios crear, compartir y analizar gráficos de manera dinámica y eficiente.

- **Paneles (Dashboards):** Composición de gráficos, tablas y otros elementos visuales que permiten a los usuarios ver y analizar datos de múltiples fuentes en una sola interfaz.
- **Fuentes de Datos (Data Sources):** Orígenes de datos que Grafana puede intercalarse entre sí para extraer información. Esto contiene bases de datos, servicios de monitoreo, y APIs.
- **Alertas:** Sistema de notificaciones configurables que avisan a los usuarios sobre condiciones específicas en los datos, ayudando a la detección temprana de problemas.

2.72 InfluxDB

Fue el puente entre Node-RED y Grafana ya que al ser una base de datos su utilidad en el proyecto fue para almacenar y analizar los datos y darles seguimiento a través de reportes y tablas de datos del proceso industrial.

Se caracteriza por manejar grandes volúmenes de datos a través de marcas de tiempo que pueden variar desde días hasta meses.

A través de la creación de “buckets” InfluxDB permite dar de alta cualquier proceso industrial como también todas las variables que se encuentran en el mismo.

CAPÍTULO III:

SISTEMA SCADA

En este proyecto se desarrolla un sistema SCADA para un proceso que emula a una pequeña y mediana empresa, y que está conformado por dos tecnologías:

La tecnología operativa que consiste en todo el hardware y software que controla el desempeño del proceso y de donde se recopilarán todos los datos y variables a ser monitoreados, y la tecnología de la información que es la base de datos y la visualización en Grafana.

Ambas tecnologías serán enlazadas mediante el puente virtual que creará el software Node-RED y las interconectará a la aplicación Telegram para permitir el control remoto del proceso.

A continuación, se presentará el procedimiento y los resultados obtenidos para cumplir la tarea mencionada en el párrafo anterior, comenzando con la tecnología operativa.

3.1 Proceso industrial en Factory I/O

A través del simulador Factory I/O se diseñaron tres etapas interconectadas con elementos de campo muy comunes (sensores y actuadores) para cumplir una tarea sencilla. Este software permitió conectarse con un PLC físico (S7-1215C) para programar el control del proceso y del variador de frecuencia.

El proceso contiene un total de 26 señales de entrada para el PLC, 24 son de tipo booleano y 2 de tipo analógico (flujo y nivel del tanque de agua). Por parte de las salidas, son un total de 21 salidas discretas y 2 salidas analógicas (válvulas de llenado y vaciado del tanque de agua). En el Anexo 1 se muestra la lista de todas las señales del proceso y su descripción, así como en la figura 3.1 se aprecian sus conexiones al PLC S7-1215C.

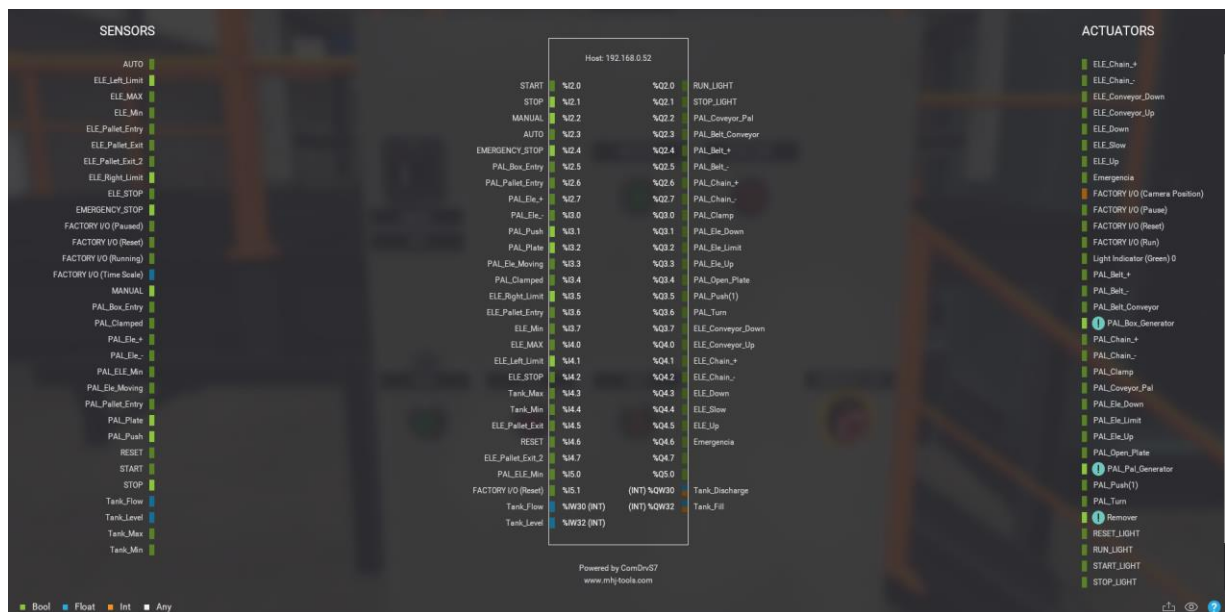


Figura 3.1. Sensores y actuadores conectados al PLC por medio de Factory IO.

En la figura 3.2 se aprecia una vista general del proceso industrial diseñado, está conformado por tres subprocesos sincronizados entre sí:

- a) Paletizado.
- b) Elevador.
- c) Tanque de suministro de agua.

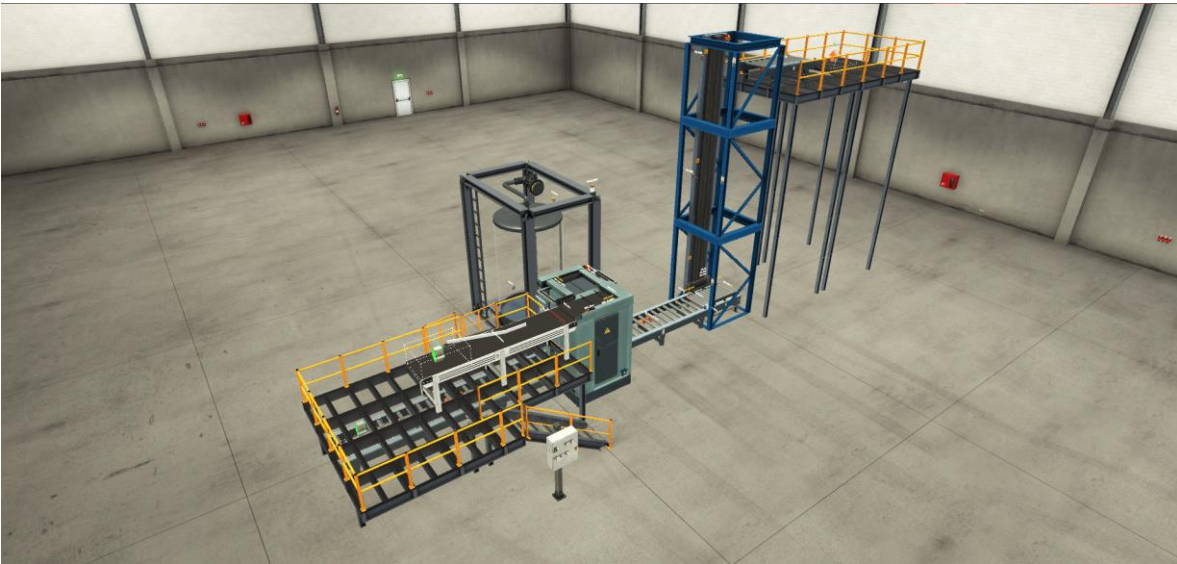


Figura 3.2. Proceso industrial diseñado en Factory I/O.

La tarea principal del proceso es mantener el suministro de agua en un nivel mayor al 10% mediante el llenado del tanque cuando sea requerido a lo largo del tiempo, ya que mientras el sistema esté en marcha se consume el líquido. Por otro lado, el proceso de paletizado se encarga de apilar seis cajas pequeñas sobre un palet que después será transportado hasta un elevador que lo llevará a la sección de entrega. El arranque, el paro, el reinicio y el paro de emergencia del proceso industrial se controla mediante un modo manual, es decir, mediante botones ubicados en un tablero dentro de Factory I/O como se muestra en la figura 3.3; y un modo automático, que permite al sistema ser controlado de manera remota por medio de la aplicación Telegram.

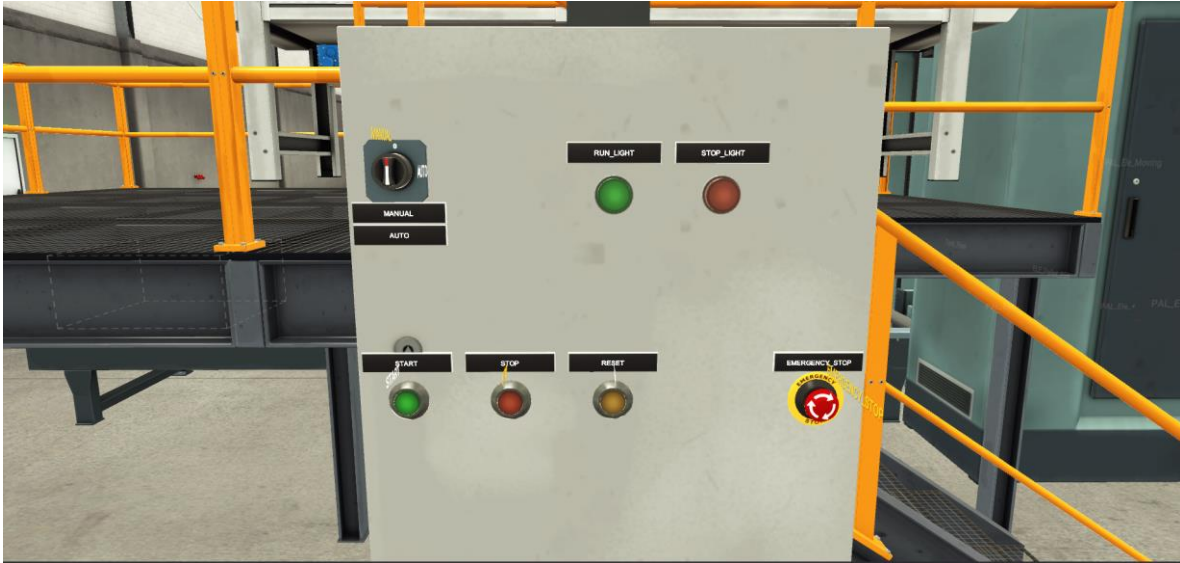


Figura 3.3. Tablero de control del proceso industrial.

3.11 Paletizado

Como se muestra en la figura 3.4 y 3.5 Este subproceso contiene un generador automático de cajas y palets sobre una cinta transportadora y un transportador de rodillos motorizados respectivamente. Por la parte inferior, los palets deben ser llevados por el transportador hacia un elevador que los subirá y colocará debajo de una compuerta automática. Por la parte superior, las cajas son movidas por la cinta transportadora hasta la superficie de un pistón que las empujará hacia la compuerta automática, donde otros dos pistones las comprimirán tratando de acomodarlas, de modo que cuando la compuerta se abra en el momento adecuado, las cajas reposen sobre el palet que espera debajo; y el elevador con el palet cargado, deberá descender y transportarlo hacia los rodillos motorizados que se dirigen al elevador de la siguiente etapa, mientras la compuerta se cierra, el elevador se vuelve a cargar con un palet vacío y el proceso reinicia el ciclo.



Figura 3.4. Subproceso de paletizado (vista frontal).



Figura 3.5. Subproceso de paletizado (vista posterior).

Las cajas apiladas sobre cada palet se acomodan formando dos columnas con tres cajas cada una, dando un total de seis cajas por cada palet. Para lograr esto, se requiere activar un brazo metálico que se encuentra sobre la cinta transportadora y que funciona como obstáculo a cada caja para que, al chocar con él cambie su posición de vertical a horizontal tal y como se ve en la figura 3.6.



Figura 3.6. Acomodo de las seis cajas sobre un palet.

Para lograr el todo cometido que se menciona en los párrafos anteriores, al final de la cinta transportadora se encuentra un sensor fotoeléctrico difuso que envía un pulso positivo cada que una caja pasa frente a él, así que esta señal es usada para saber que después de que han pasado seis cajas y el palet cargado ha descendido, el ciclo se ha completado y se debe reiniciar, de igual manera, sirve para detener la cinta y que no permita que las cajas continúen avanzando mientras el pistón que las empuja hacia la compuerta está extendido como expone la figura 3.7

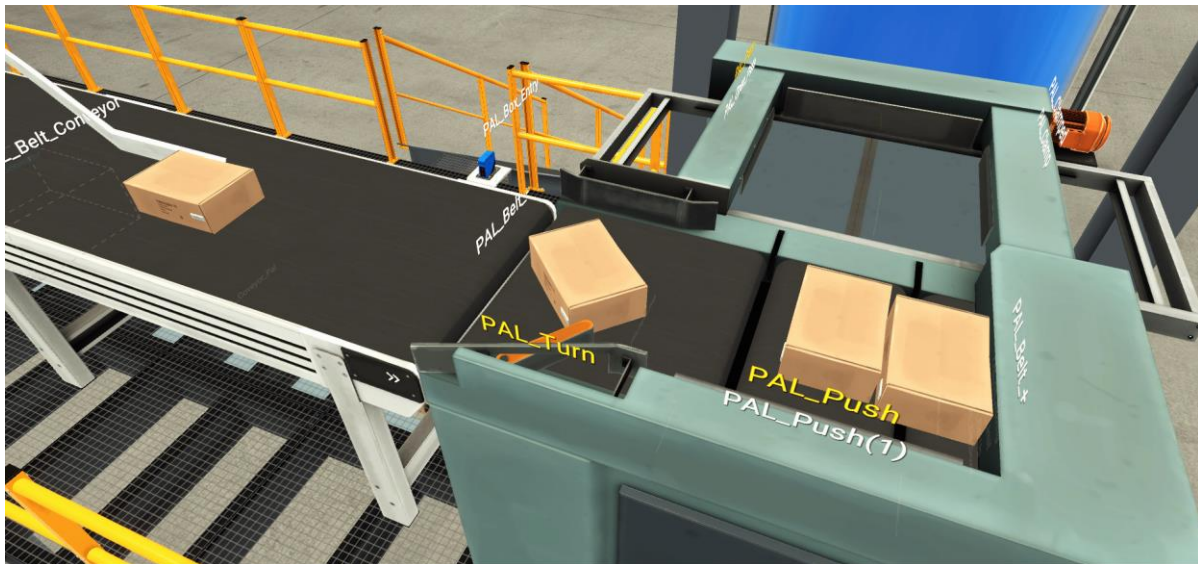


Figura 3.7. Organizado de cajas para el paletizado.

3.12 Elevador

Este es el segundo subproceso del sistema, la figura 3.8 muestra un elevador sencillo que en su cabina transporta el palet cargado con las seis cajas desde su límite inferior hasta su límite superior. Para hacer llegar el palet cargado al elevador, existe un transportador de rodillos motorizados que conecta la etapa de paletizado con la cabina del elevador, de modo que cuando el palet cargado termina de descender, el transportador se activa hasta que el palet ingresa en la cabina y una vez adentro, el elevador entra en marcha subiendo hasta llegar a su límite superior, en donde hay otro transportador de rodillos que permitirá retirar el palet de la cabina y llevarlo a la rampa de entrega, una vez que la cabina está vacía, el elevador desciende para reiniciar el ciclo con el siguiente palet que está abajo esperando a entrar.



Figura 3.8. Subproceso del elevador.

3.13 Modbus RTU

Para ir más allá de la simulación, se utiliza un motor trifásico controlado por el variador de frecuencia PowerFlex4 para que replique físicamente el comportamiento del elevador en Factory I/O, es decir, el variador de frecuencia se pondrá en marcha el motor cada vez que el elevador del simulador suba o baje, girando en sentido horario o antihorario, respectivamente. El programa está adaptado para que las mismas salidas -bajo sus condiciones dadas por los sensores- que controlan el movimiento del elevador en el simulador, controlen al mismo tiempo el arranque, el paro, el cambio de giro y la velocidad del variador.



Figura 3.9. Variador de frecuencia PowerFlex4 dentro del subproceso del elevador.

Debido a que Factory I/O no permite simular protocolos de redes industriales, la comunicación entre el PLC y el variador de frecuencia PowerFlex4 es de forma física mediante el protocolo Modbus RTU, tal como se muestra en la figura 3.9. Por consiguiente, la programación del protocolo Modbus RTU se realiza en una función aparte del PLC haciendo uso de los bloques de comunicación Modbus Master que ofrece TIA Portal.

3.14 Tanque de suministro de agua

El tercer subproceso se muestra en la figura 3.10. Está conformado por un tanque de agua, que simulará el suministro del líquido a la planta. El funcionamiento del tanque de agua comienza cuando inicia el sistema. Cuando se inicializa la simulación el tanque comienza a llenarse por medio de una bomba ubicada en la parte superior del mismo con las condiciones de vacío (la condición anterior no sucede en un proceso industrial real). Inmediatamente después de llegar a un nivel determinado, la válvula de vaciado se abre simulando el consumo de agua, mientras se continúa llenando hasta llegar al nivel máximo, una vez ahí, la bomba de llenado se apaga, y el consumo persiste, y al llegar al nivel mínimo, el ciclo se reinicia. Esto se logra con dos sensores capacitivos ubicados sobre la superficie externa del tanque (mínimo y máximo) que permitirán detectar el líquido en su interior y con ello tomar las decisiones correspondientes.



Figura 3.10. Subproceso del tanque de suministro de agua.

3.2 Programación del PLC

El PLC físico contiene y ejecuta el programa del proceso industrial de acuerdo con la lógica de cada etapa descrita en la sección anterior. El entorno de programación usado es TIA Portal V14 y cada segmento del bloque de organización principal [Main (OB1)] contiene una función que sigue al diagrama de flujo que se muestra en la figura 3.11.

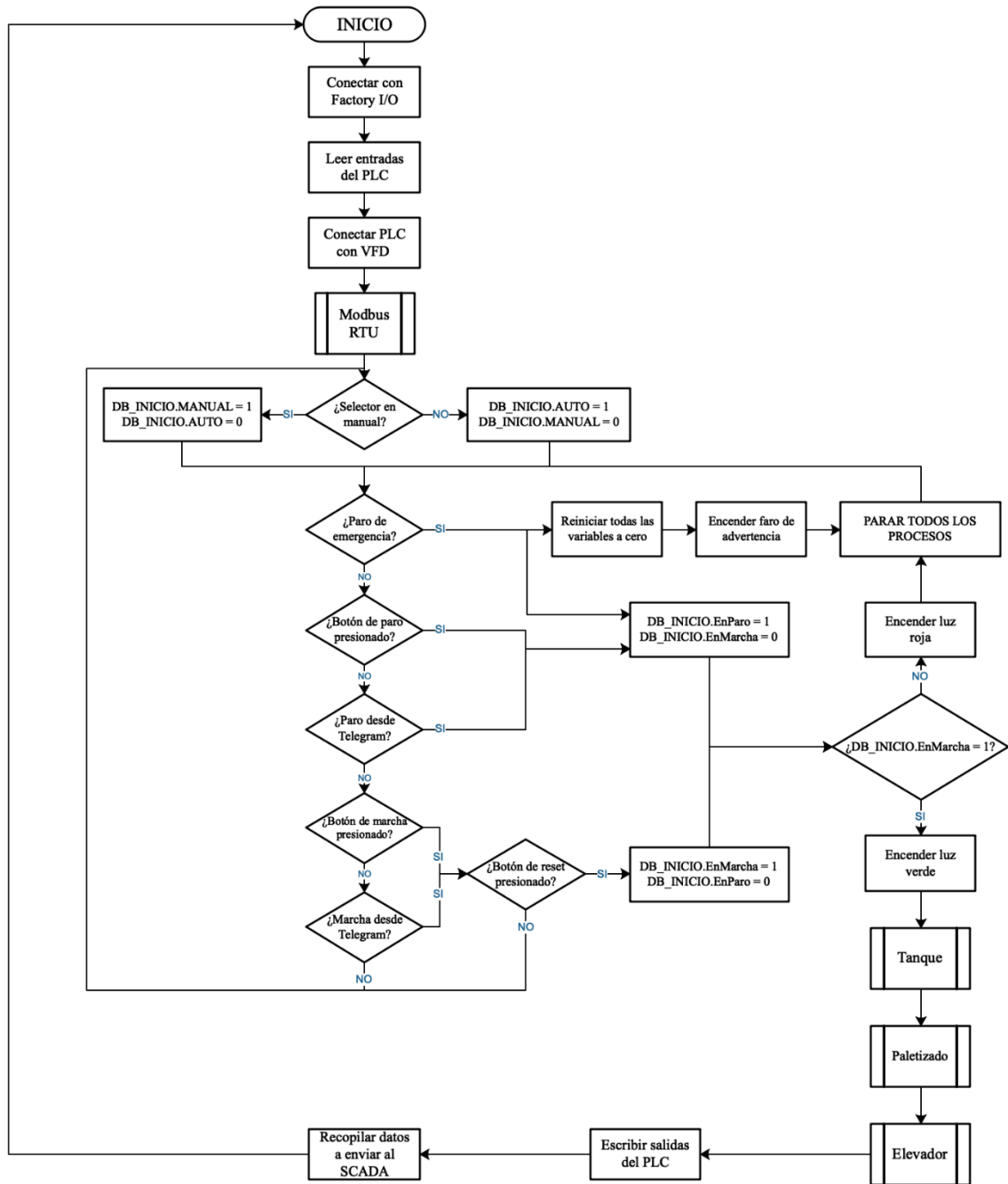


Figura 3.11. Diagrama de flujo del bloque de organización Main (OB1) del proceso industrial.

Los bloques de proceso y subproceso del diagrama de flujo anterior se describen detalladamente como funciones a continuación.

3.21 Funciones de comunicación

Dentro de estas funciones se concentran las instrucciones para establecer una correcta comunicación entre software y hardware externo al PLC, tal como Factory I/O y el variador de frecuencia PowerFlex4, respectivamente.

- a) *Conexión con Factory I/O*. Este software en su página oficial ofrece la herramienta de conexión a manera de un programa de ejemplo para la versión deseada de TIA Portal, se trata de una función programada en lenguaje SCL que sirve para establecer la conexión entre el entorno de Siemens y el de Factory I/O, enlazando las salidas y entradas mostradas en la Figura 3.3 con las del PLC físico.
- b) *Lectura de datos del variador*. En esta función se encuentran los bloques de comunicación Modbus Máster para poder ir a los registros de lectura del variador de frecuencia usando la función 03, su configuración se basa en el modo de transmisión, direcciones de registro y dirección en la red del esclavo.
- c) *Escritura de datos al variador*. De manera similar al segmento anterior, aquí se establecen los parámetros correctos del bloque Modbus para usar la función de escritura 06 y acceder a los registros del variador de frecuencia para controlarlo de manera remota de acuerdo con sus funciones disponibles descritas en el manual de usuario (Anexo 2).

3.22 Funciones de proceso

Estas funciones contienen la lógica de control del proceso general y de cada subproceso de acuerdo con las tareas que cada uno debe realizar.

- a) *Función inicio*. En este segmento se encuentran las condiciones para controlar todo el sistema de manera manual (botones del tablero en Factory I/O), o automática (vía Telegram).
 - a. *Marcha*. La marcha del sistema se logra presionando el botón verde del tablero de la Figura 3.1 y después presionando el botón de reset; el programa está diseñado para que el sistema después de un paro no arranque hasta cumplir con estas dos condiciones. Durante el tiempo que el sistema esté en marcha se cuentan los segundos, minutos y horas de trabajo, de igual manera, la luz verde del tablero principal enciende durante este periodo. Si el proceso está puesto en automático, la marcha del sistema ocurre enviando el comando /sistema marcha desde el chat de Telegram.
 - b. *Paro*. El paro simple del sistema pone en pausa el proceso por un tiempo indefinido hasta ponerlo de nuevo en marcha. Se diseñó el programa de tal manera que el paro no reinicia ni afecta la secuencia que el proceso esté

- llevando en el momento del cambio de estado, sino que solamente la interrumpe para poder continuar donde se quedó.
- c. *Paro de emergencia.* En este estado del sistema se interrumpe la marcha de todos los subprocesos, se enciende un faro de advertencia y no permite la marcha general hasta que la seta de emergencia esté liberada. Tampoco afecta la secuencia que el programa esté llevando en ese momento, se puede retomar al ponerlo en funcionamiento de nuevo.
 - d. *Tiempo de funcionamiento.* Como se mencionó anteriormente, se programaron instrucciones que realizan el conteo del tiempo que el proceso está funcionando y también cuando se encuentra en paro, esto con el fin de que cuando se calcule la eficiencia del proceso se tomen en cuenta los tiempos muertos también.
- b) *Subprocesos: tanque, elevador y paletizado.* Estos tres segmentos contienen las funciones con la lógica de trabajo de cada proceso de acuerdo con las tareas que deben realizar descritas en la sección anterior. Cada proceso es independiente, pero se encuentran sincronizados por medio de los sensores que los conectan para saber en qué momento deben funcionar. La marcha y el paro del sistema son condiciones principales de estos procesos y son las que deciden si cada subproceso tiene permitido arrancar o no.

3.23 Funciones de procesamiento de datos

En estos últimos segmentos se encuentran todas aquellas señales que controlan el proceso, tanto de salida como de entrada, así como las variables que serán monitoreadas en el sistema SCADA.

- a) *Lectura de entradas y escritura de salidas del PLC.* En estos dos segmentos se realiza la lectura de las entradas usadas por Factory I/O en donde están conectados los sensores del sistema, estos valores son almacenados en un bloque de datos para que posteriormente se usen en la programación de cada proceso; y se realiza la escritura de salidas transfiriendo los valores almacenados en otro bloque de datos hacia las salidas del PLC conectadas en el software para el funcionamiento de los actuadores.

- b) *Datos para enviar al SCADA*. Aquí se recolectan todas las variables que serán monitoreadas en el tablero de Grafana, el programa las lee de cada subproceso y las almacena en un solo bloque de datos para que sea más fácil acceder a ellas desde Node-RED. Son un total de 13 variables de lectura y 3 de escritura las cuales se describen a continuación y se muestran en la figura 3.12.
- a. *Proceso On-Off*. Esta variable de tipo booleano cambia de estado cada que el proceso está en marcha o en paro, valiendo 1 y 0, respectivamente.
 - b. *Proceso en emergencia*. Cuando el sistema está en un paro de emergencia, esta variable toma el valor de 1 y permanece así hasta liberar esa condición cambiando a 0.
 - c. *Elevador Up-Down*. Si el elevador en el simulador está subiendo, la variable cambia su estado lógico a verdadero y cuando el elevador baja toma el estado lógico falso.
 - d. *Tanque filling*. Cuando la válvula de llenado del tanque de agua está activa, la variable vale 1 y continúa en ese estado hasta que el tanque se llena por completo y la válvula de llenado se cierra tomando el valor de 0.
 - e. *Falla variador*. En esta variable se monitorea si el variador de frecuencia presenta alguna falla dentro de su funcionamiento, si esto se cumple el valor de esta será 1 y si el variador funciona correctamente permanece en 0.
 - f. *Número de falla*. Es una variable de tipo entero que guarda el número de la falla que presente el variador de acuerdo con la tabla de la Figura X en el manual de usuario de PowerFlex4.
 - g. *Número de cajas*. Almacena el conteo de cuántas cajas han pasado al proceso de paletizado durante todo el tiempo que el proceso esté en marcha.
 - h. *Numero de palets entregados*. Lleva el conteo de todos los palets cargados que han llegado hasta la rampa de entrega y que cuentan como aquellos que son entregados satisfactoriamente.
 - i. *Velocidad del variador*. Aquí se guarda la lectura de la velocidad actual del variador de frecuencia en unidades de Hertz.
 - j. *Nivel de agua*. Esta variable toma la lectura del sensor de nivel del tanque de suministro de agua convirtiéndola en porcentaje.
 - k. *Flujo de agua*. Almacena la lectura del sensor de flujo conectado a la válvula de vaciado del tanque de agua y ofrece una lectura en unidades de litros por minuto.
 - l. *Eficiencia*. La eficiencia del proceso está calculada en base a cuántos palets son entregados satisfactoriamente en un tiempo determinado, la meta del proceso, llamada *tasa de producción real* T_{pr} , es entregar 4 palets en 60 segundos, esto nos lleva a la siguiente relación ideal:

$$T_{pr} = \frac{4}{60} \approx 0.666666667 \quad \dots \text{Ecuación 3.1}$$

Por lo tanto, la *eficiencia de producción* E_p , se calcula con la relación entre los palets entregados y el tiempo de funcionamiento (incluyendo tiempos en paro), llamado *tasa de producción estándar* T_{pe} , dividido entre la relación ideal y multiplicado por 100 para obtenerlo en porcentaje:

$$T_{pe} = \frac{\text{Número de palets entregados}}{\text{Tiempo total}} \quad \dots \text{Ecuación 3.2}$$

$$E_p = \frac{T_{pe}}{T_{pr}} \times 100 \quad \dots \text{Ecuación 3.3}$$

m. *Marcha, reset y paro externo.* Estas son tres variables de escritura que Node-RED enlazará al chat de Telegram para controlar el proceso a distancia.

Name	Data type	Offset	Start value	Retain	Accessible f...	Write...	Visible in ...	Setpoint	Comment
Static									
PROCESO_OFF_ON	Bool	0.0	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
MARCHA_EXT	Bool	0.1	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	ESCRITURA NODE RED
PARO_EXT	Bool	0.2	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	ESCRITURA NODE RED
RESET_EXT	Bool	0.3	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	ESCRITURA NODE RED
PROCESO_EN_EMERGE...	Bool	0.4	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
ELEVADOR_UP_DOWN	Bool	0.5	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
TANQUE_FILLING	Bool	0.6	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
FALLA_VARIADOR	Bool	0.7	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
NUMERO_FALLA	Int	2.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
NUMERO_CAJAS	Int	4.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
NUMERO_PAL_ENT	Int	6.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
VELOCIDAD_VARIADOR	Int	8.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
SEGUNDOS_TRABAJO	Int	10.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
MINUTOS_TRABAJO	Int	12.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
HORAS_TRABAJO	Int	14.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
NIVEL_AGUA	Real	16.0	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
FLUJO_AGUA	Int	20.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
EFICIENCIA	Real	22.0	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Figura 3.12. Bloque de datos (DB500) para enlazar a Node-RED.

3.3 Node-RED

El software Node-RED, como se mencionó anteriormente, establecerá el puente entre la tecnología operativa y la tecnología de la información; accederá al bloque de datos del PLC llamado *DB_SCADA (DB500)* descrito anteriormente y transferirá esos valores a una base de datos creada en la plataforma InfluxDB.

3.31 Conexión entre Node-RED y el PLC

El primer paso es instalar el Software Node-RED en un ordenador de manera local e instalar la librería *node-red-contrib-s7* a través del Manage Palette del software. Esta librería contiene tres nodos de control tal y como se ve en la figura 3.13.

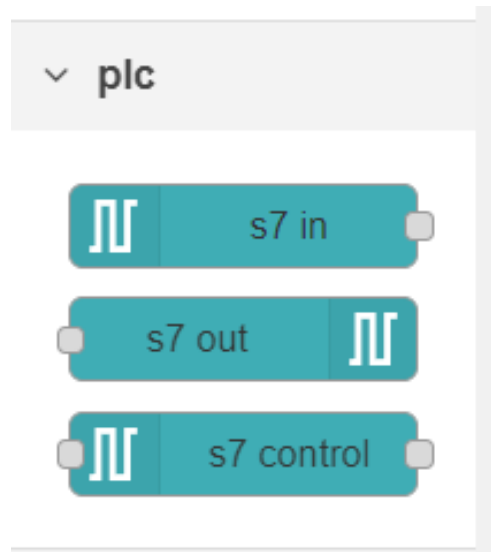


Figura 3.13. Nodos de la librería *node-red-contrib-s7*.

- 1) *s7 in*: Este nodo lee datos desde un PLC S7.
- 2) *s7 out*: Este nodo escribe a un área de memoria de un PLC S7.
- 3) *s7 control*: Habilita un control avanzando entre el PLC y la conexión.

El primer nodo por utilizar para leer las variables del bloque de datos *DB500* del PLC es el *s7 in* y se configura de la siguiente manera como se observa en la figura 3.14.

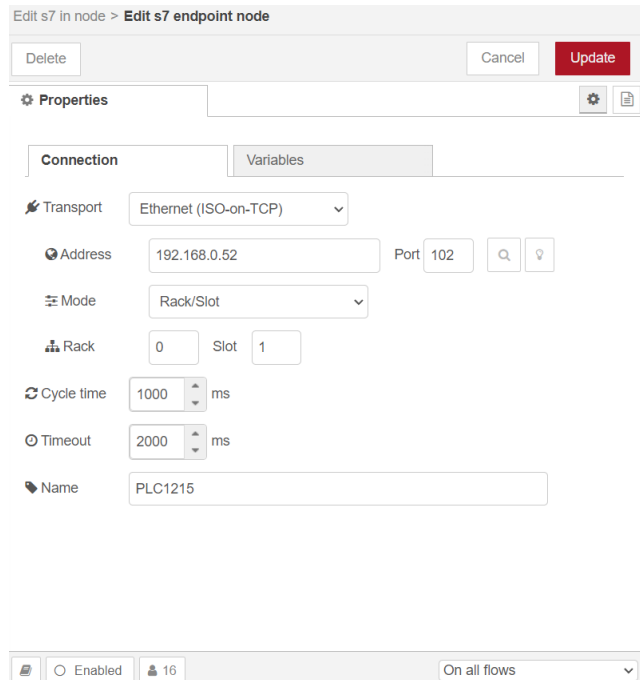


Figura 3.14. Primera parte de la configuración del nodo s7 in.

Como se observa en la figura anterior, en la pestaña *Connection* en el parámetro *Transport* se selecciona la opción *Ethernet (ISO-on-TCP)*. Posteriormente, en el parámetro *Address* se asigna la dirección del PLC físico, los demás parámetros se dejan por defecto y, al final se le asigna el nombre a la conexión que en este caso es *PLC1215*.

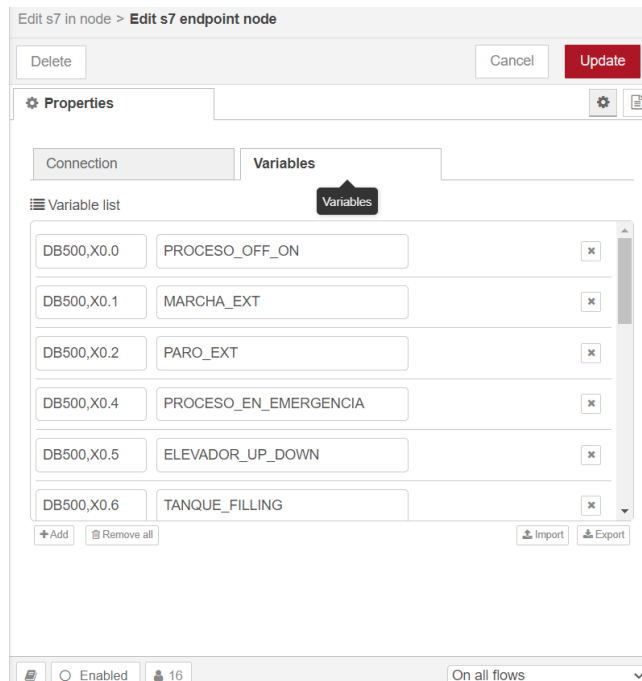


Figura 3.15. Segunda parte de la configuración del nodo s7 in.

En la pestaña *Variables* se asignan las direcciones y nombres de las variables del bloque de datos del PLC (*DB500*) según su tipo y ubicación dentro del mismo, tal como se muestra en la Figura 3.15.

3.32 Procesamiento de datos dentro de Node-RED

Una vez configurada la conexión y la creación de todas las variables, se procede a insertar un nodo *s7 in* por cada una de ellas. Se necesita tratar o modificar los valores de estas variables y adecuarlos correctamente para enviarlas a la base de datos en InfluxDB. Para ello se hace uso de algunas funciones e intercambio de datos en algunas variables dependiendo de su contenido.

Las señales digitales son adecuadas de tal manera que cuando su valor sea 0, en la consola de Node-RED muestre la palabra “*OFF*”; y cuando su valor sea 1 se muestre “*ON*”. Para llevar a cabo esto, se necesita un nodo de función con el siguiente código en su interior como se muestra en la figura 3.16.

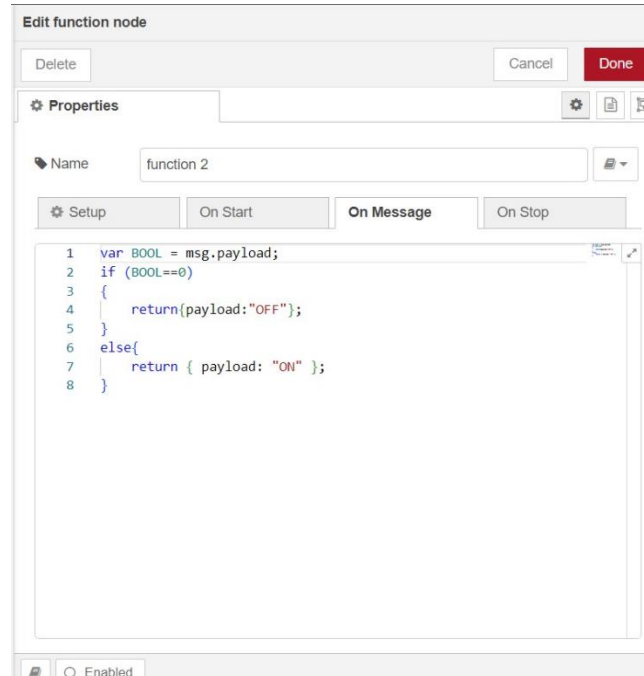


Figura 3.16. Configuración del nodo función para cada variable de lectura discreta.

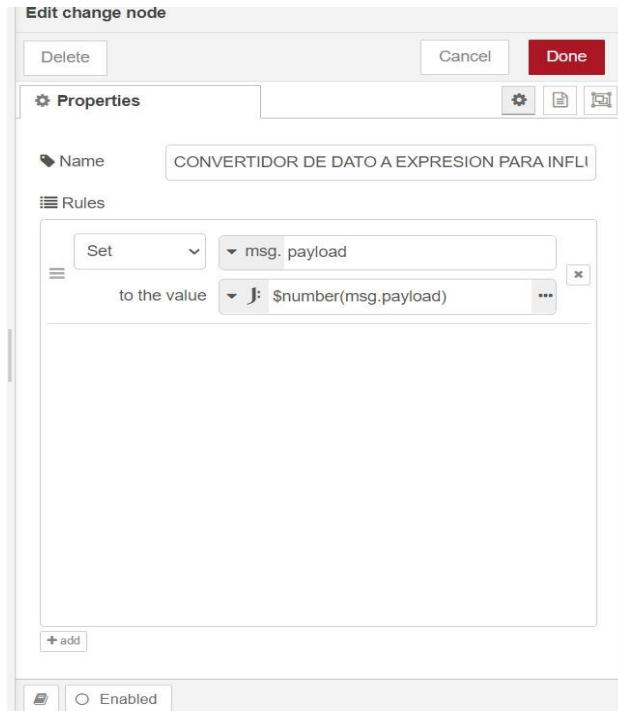


Figura 3.17. Configuración del nodo intercambio para cada variable de lectura.

Por otro lado, se requiere de un nodo de intercambio que convierta el dato proveniente del *DB500* del PLC en una expresión numérica de lenguaje JSON (el cual es un lenguaje versátil que facilita la comunicación entre diferentes sistemas) que permitirá que InfluxDB pueda leer y almacenar correctamente dicho dato. El interior del nodo se muestra en la Figura 3.17.

Algunas variables enteras requieren un tratamiento especial dependiendo de su contenido, y otras solamente necesitan del intercambio de datos mencionado en el párrafo anterior. Aquellas que requieren de modificar su contenido con una función son las siguientes:

- a) *Velocidad del variador*. Dado que el variador de frecuencia entrega la velocidad de giro actual en un formato de tres dígitos sin punto decimal, es necesario dividir ese valor entre diez para obtener la lectura real. Es decir, si la velocidad del variador fuera de 34.6 Hz, el dato que envía el mismo sería 346 y necesita ser convertido a punto decimal. Esta conversión se logra con la siguiente función como lo expresa la figura 3.18.

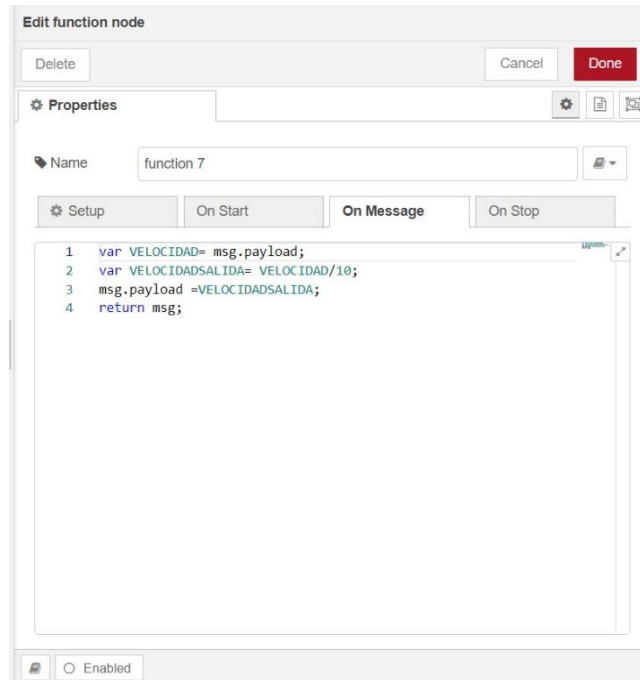


Figura 3.18. Configuración del nodo función para la velocidad del variador.

- b) *Nivel de agua*. Al ser una variable de tipo real, el valor del nivel de agua contiene más de cinco decimales, por lo tanto, es necesario redondear ese valor a dos decimales para una mejor visualización en Grafana. El interior de la función que logra esta tarea se muestra en la figura 3.19.

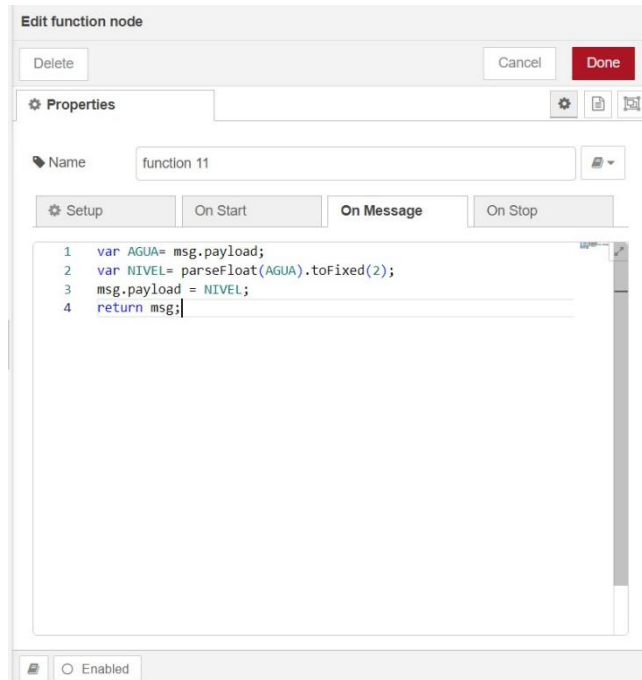


Figura 3.19. Configuración del nodo función para el nivel de agua del tanque.

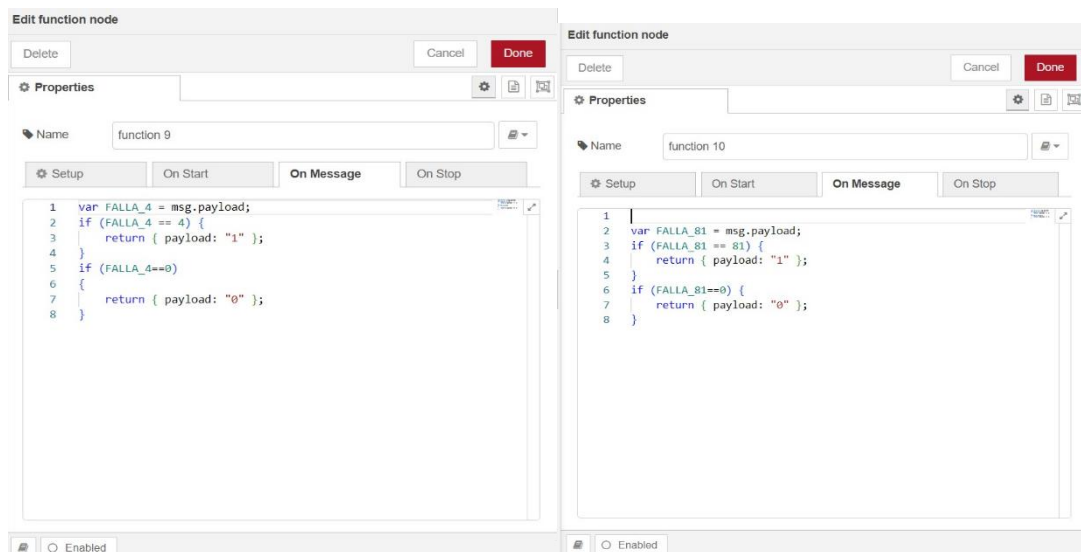


Figura 3.20. Configuración del nodo función para el número de falla en el variador.

- c) *Número de falla.* Por cuestiones prácticas y de simulación, las dos fallas disponibles para comprobar en el variador son la falla por falta de alimentación eléctrica y de comunicación con el PLC. Estas fallas son la número 4 y la 81 respectivamente. Dado que en Grafana se quiere mostrar el nombre de la falla que esté ocurriendo en ese momento, se necesita mandar una señal booleana para indicar a que falla hace referencia. Esto se logra modificando la señal del variador de entera a discreta con los códigos de la función de la Figura 3.20. En este caso solo se consideran 2 posibles fallas por cuestiones prácticas, de las que se pueden presentar en el variador de frecuencia.

3.33 Conexión entre Node-RED e InfluxDB

Una vez procesados y adecuados los valores de cada variable, se inserta el último nodo llamado *influxdb out* que permite la conexión entre Node-RED y la plataforma en línea de InfluxDB, dicho nodo se configura con los siguientes parámetros, como se muestra en la figura 3.21.

The image shows a configuration window for the 'influxdb out' node in Node-RED. The window has a title bar 'Edit influxdb out node' and three buttons: 'Delete', 'Cancel', and 'Done'. Below the buttons is a 'Properties' section with a gear icon and three small icons. The configuration fields are: 'Name' with the value 'ELEVADOR'; 'Server' with a dropdown menu showing '[v2.0] PROCESO INDUSTRIAL' and an edit icon; 'Organization' with the value 'SCADA'; 'Bucket' with the value 'PLC_S71215'; 'Measurement' with the value 'ELEVADOR'; and 'Time Precision' with a dropdown menu showing 'Milliseconds (ms)'. At the bottom left, there is a checkbox labeled 'Enabled' which is currently unchecked.

Figura 3.21. Primera parte de la configuración del nodo *influxdb out*.

Se usará un nodo por cada variable a monitorear, es decir, se le asigna el nombre de cada una a los campos *Name* y *Measurement*. Por otro lado, *Organization* y *Bucket* contienen el mismo nombre asignado en InfluxDB, los cuales se mostrarán en la siguiente sección. El parámetro *Server* se configura de la siguiente manera, como se observa en la figura 3.22.

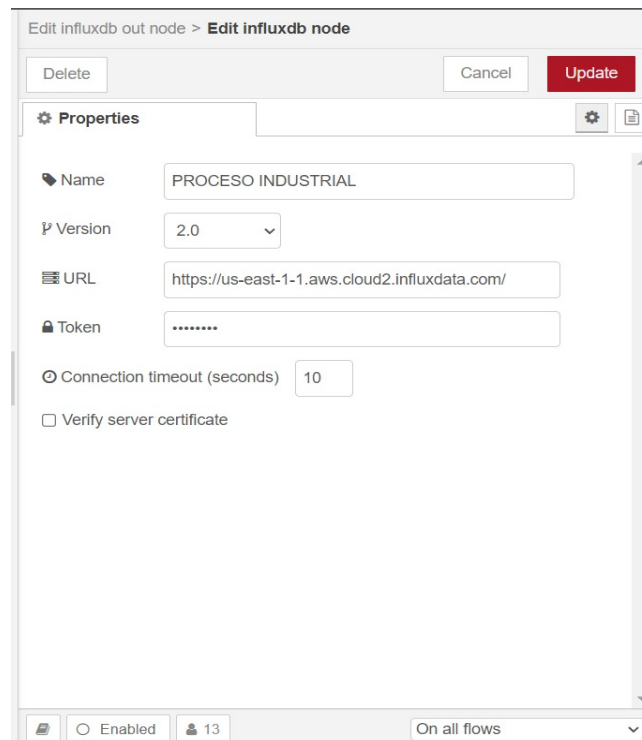


Figura 3.22. Segunda parte de la configuración del nodo influxdb out.

El nombre asignado al servidor a conectar con InfluxDB es *PROCESO INDUSTRIAL* y la versión usada de la plataforma es la 2.0. Los siguientes dos parámetros se obtienen de la plataforma InfluxDB vistos en el apartado de funciones de procesamiento de datos. Una vez configurados estos nodos, las redes para variables digitales y enteras quedan completas y funcionales, tal como se muestra en las figuras 3.23 y 3.24.

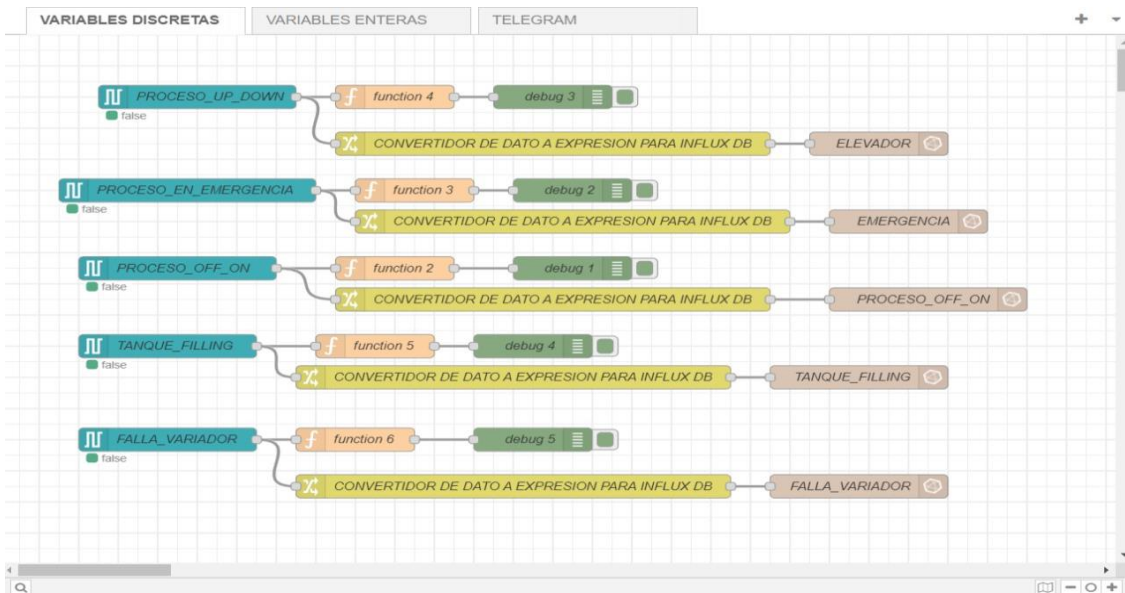


Figura 3.23. Red de conexión de variables discretas en Node-RED.

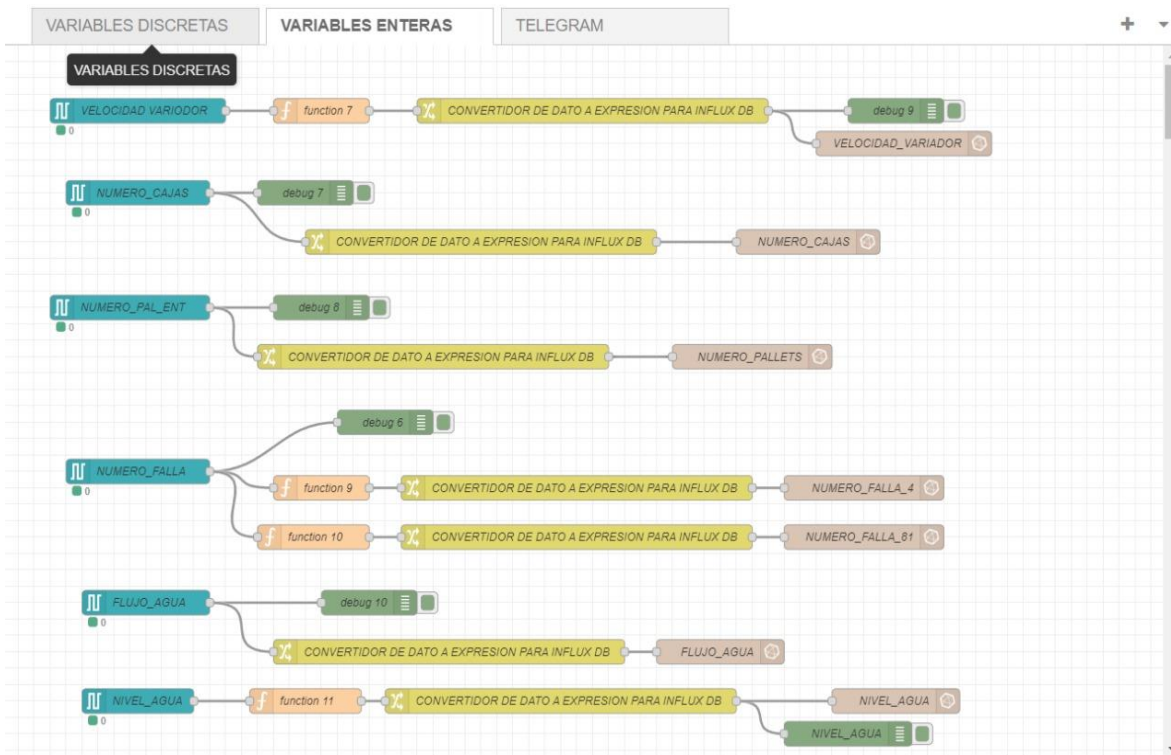


Figura 3.24. Red de conexión de variables enteras en Node-RED.

3.4 InfluxDB

El uso de esta plataforma consiste en crear una cuenta en la versión InfluxDB Cloud 2.0 y asignar un nombre de compañía y organización, así como elegir un proveedor de almacenamiento para la base de datos como lo muestran las Figuras 3.25 y 3.26.

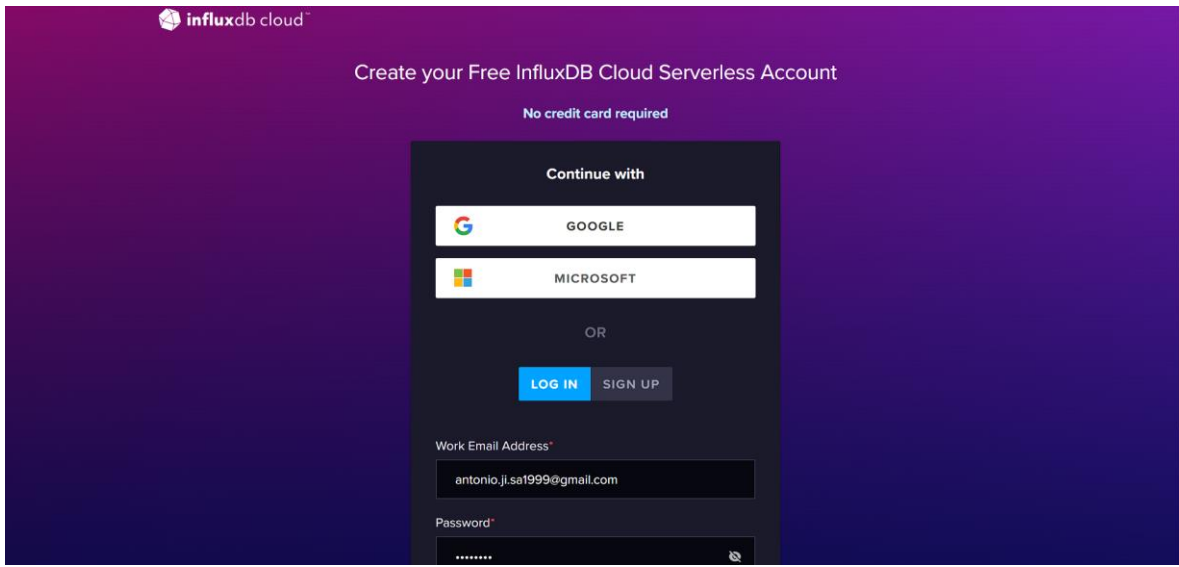


Figura 3.25. Creación de cuenta en InfluxDB Cloud 2.0.

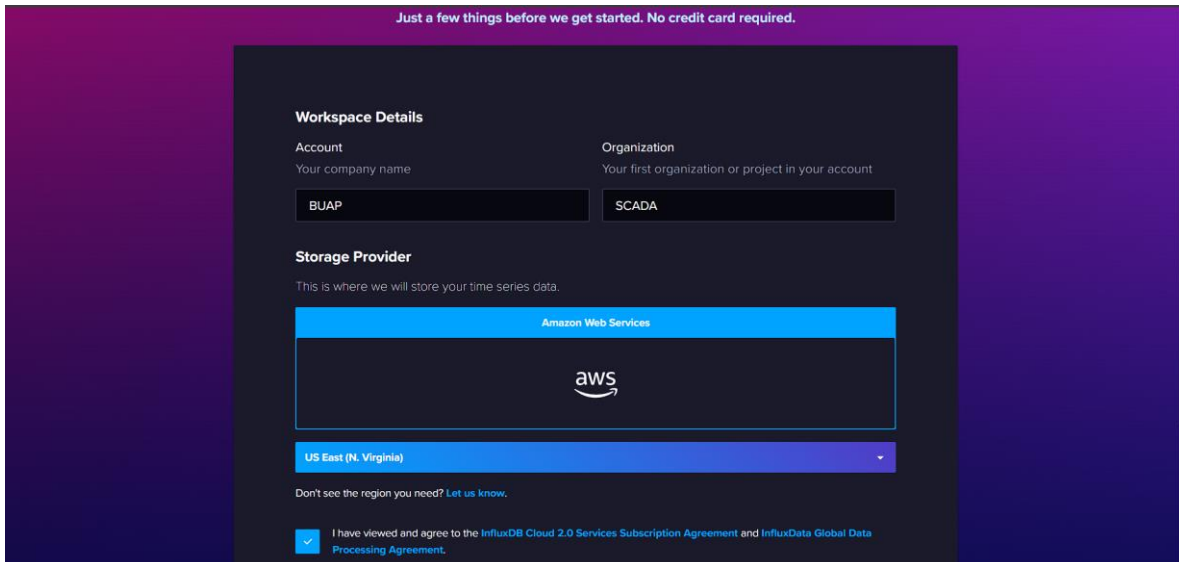


Figura 3.26. Configuración de la base datos en InfluxDB Cloud 2.0.

Posteriormente, se crea un espacio llamado *Bucket* que se encarga de enlazar cada variable proveniente de Node-RED y almacenarla en la base de datos para su posterior lectura en Grafana. Los nombres del *Bucket* y de la *Organization*, son los mismos que van en los parámetros de la Figura 3.21.

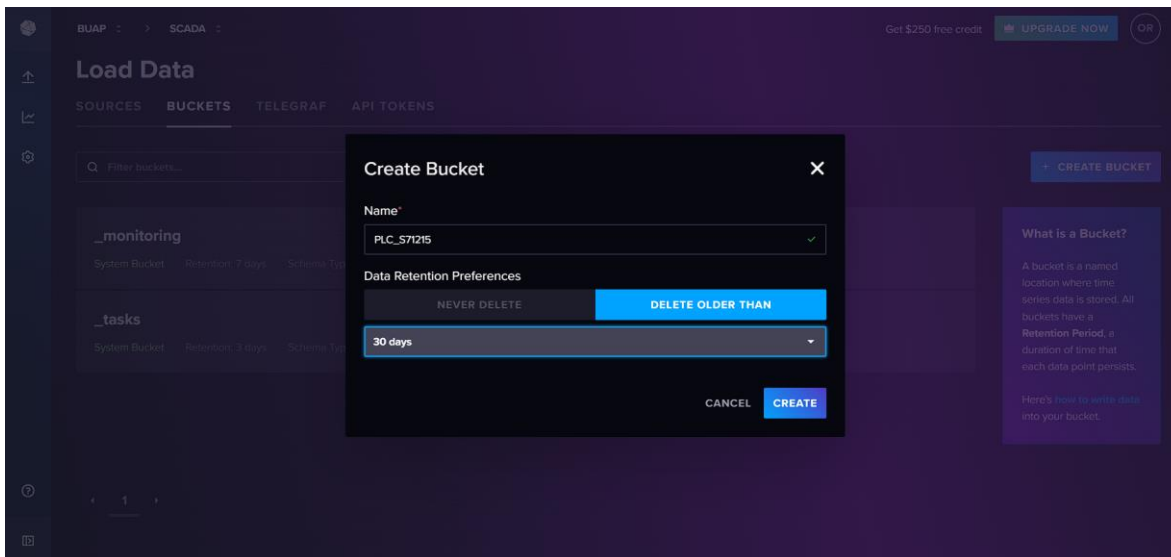


Figura 3.27. Creación del Bucket en InfluxDB.

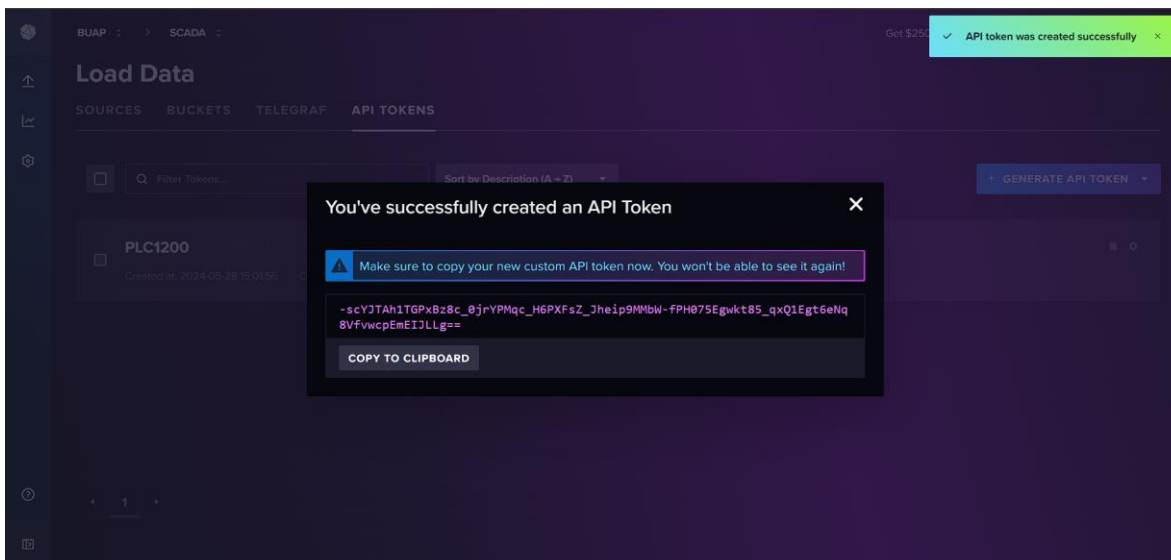


Figura 3.28. Generación del token en InfluxDB para enlace externo.

Un token es un único código de letras, números y símbolos que permite enlazar la base de datos con Node-RED y Grafana, en este caso, el token generado en InfluxDB permitirá conectarse con los nodos de salida y leer los datos que provienen de Node-RED. Este código, así como la URL de InfluxDB, se copian y se pegan en la configuración del nodo de la Figura 3.22.

Una vez ejecutados los diagramas en Node-RED, automáticamente dentro del *Bucket* de InfluxDB se cargarán las variables con sus respectivos nombres asignados en los nodos y sus valores deseados, esto indica que la conexión ha sido completada satisfactoriamente y se puede conectar a la base de datos con Grafana como se muestra en la figura 3.29.

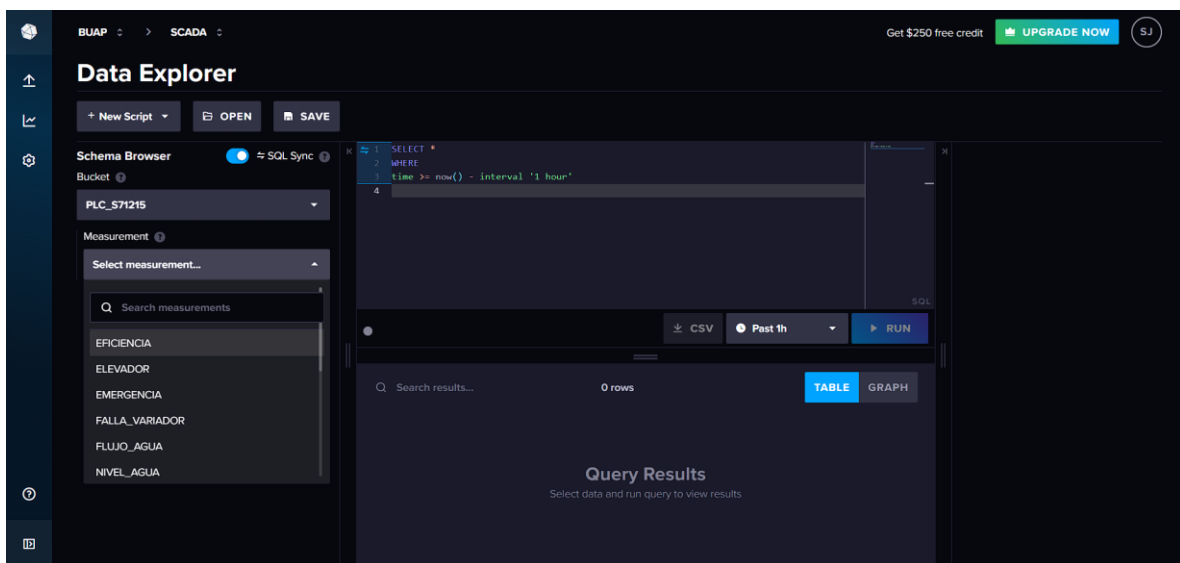


Figura 3.29. Base de datos en InfluxDB funcionando correctamente.

Se toma en cuenta que la versión gratuita de InfluxDB solamente ofrece treinta días de almacenamiento de datos, posterior a ese periodo se eliminan y carga los más recientes.

3.5 Grafana

La creación del dashboard en Grafana es el último paso para realizar la monitorización del proceso industrial, aquí se observarán los valores de las variables elegidas y almacenadas en la base de datos de InfluxDB, usando los diferentes tipos de visualizaciones que ofrece el software. Por lo anterior, la primera tarea a realizar es crear una cuenta gratuita en Grafana Cloud y ejecutar el entorno de visualización para creación de dashboards como se muestra en la figura 3.30.

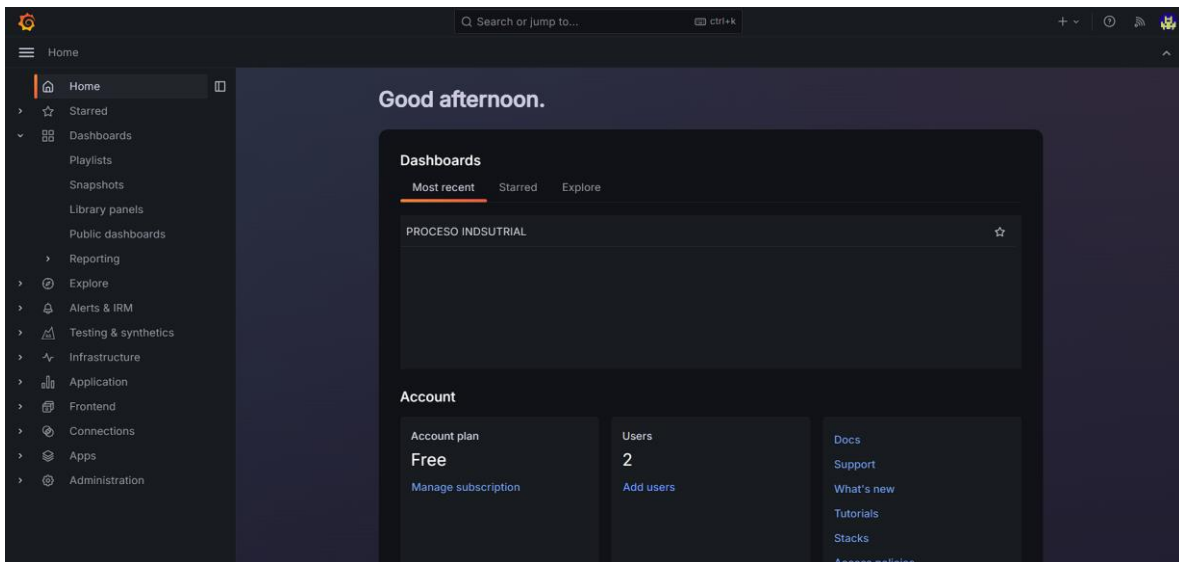


Figura 3.30. Entorno de desarrollo de dashboards en Grafana.

3.51 Conexión entre Grafana e InfluxDB

En la sección *Connections* del menú lateral de la Figura 3.30, se encuentra la opción para elegir la plataforma que almacena los datos a la cual se conectará Grafana para obtener las variables a monitorear. Para este caso se instala la opción de InfluxDB, posteriormente en la subsección *Data Sources* se debe crear una nueva base de datos de tipo InfluxDB y configurarla como se muestra en la Figura 3.32 y 3.33.

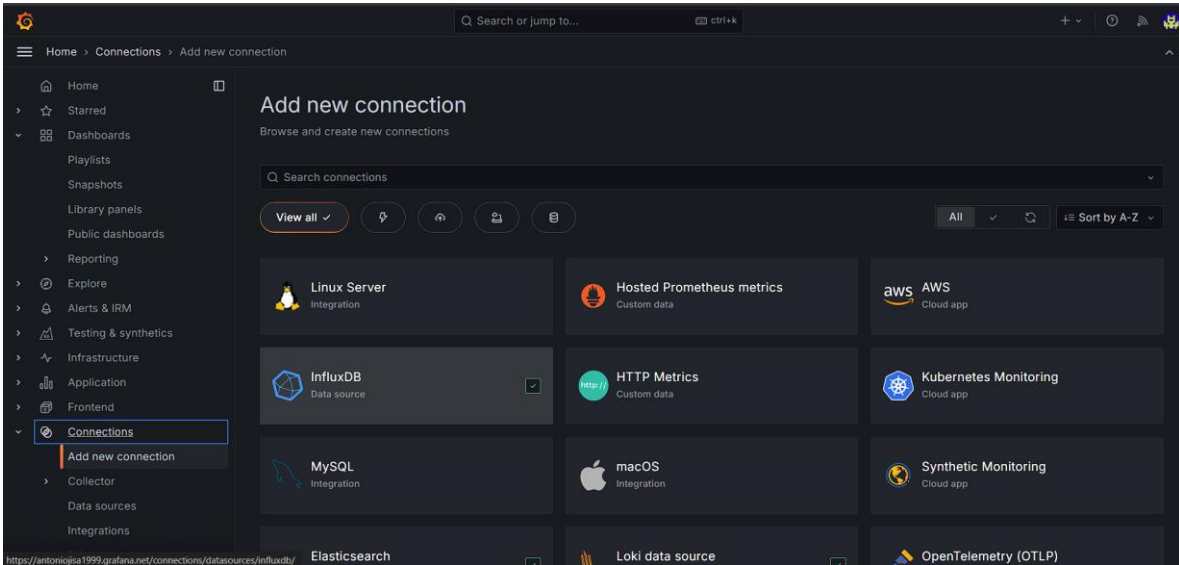


Figura 3.31. Establecer una conexión con InfluxDB en Grafana.

El lenguaje por defecto para programar la adquisición de datos dentro de Grafana hacia InfluxDB es Flux el cual se llama así porque se centra en el flujo de datos. Los demás parámetros son configurados con el URL, nombre de organización, token y nombre del *Bucket* creado en InfluxDB como se muestra en las figuras 3.32 y 3.33.

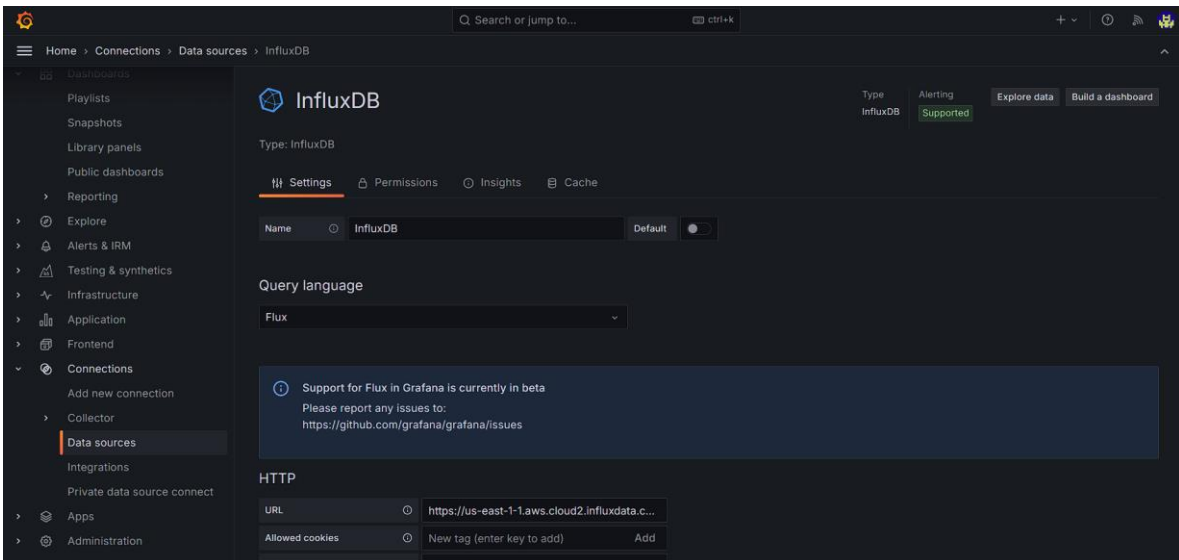


Figura 3.32. Primera parte de la configuración de la base de datos InfluxDB en Grafana.

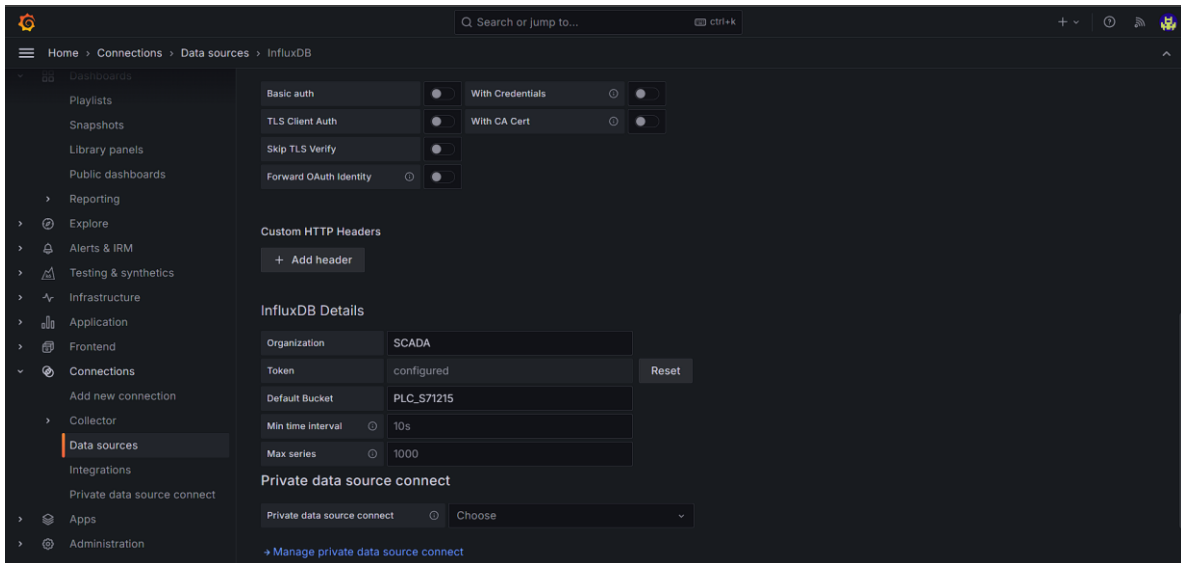


Figura 3.33. Segunda parte de la configuración de la base de datos InfluxDB en Grafana.

3.52 Creación del dashboard en Grafana

Una vez establecida la conexión entre Grafana e InfluxDB, se crea un nuevo dashboard para insertar todas las visualizaciones de cada variable. El proceso por seguir dentro de cada visualización consiste en elegir el tipo de gráfico de acuerdo con el tipo de variable a monitorear, así como el título de la visualización, color, texto, unidades, descripción y otras configuraciones que van de acuerdo con una la presentación como se muestra en la figura 3.34.

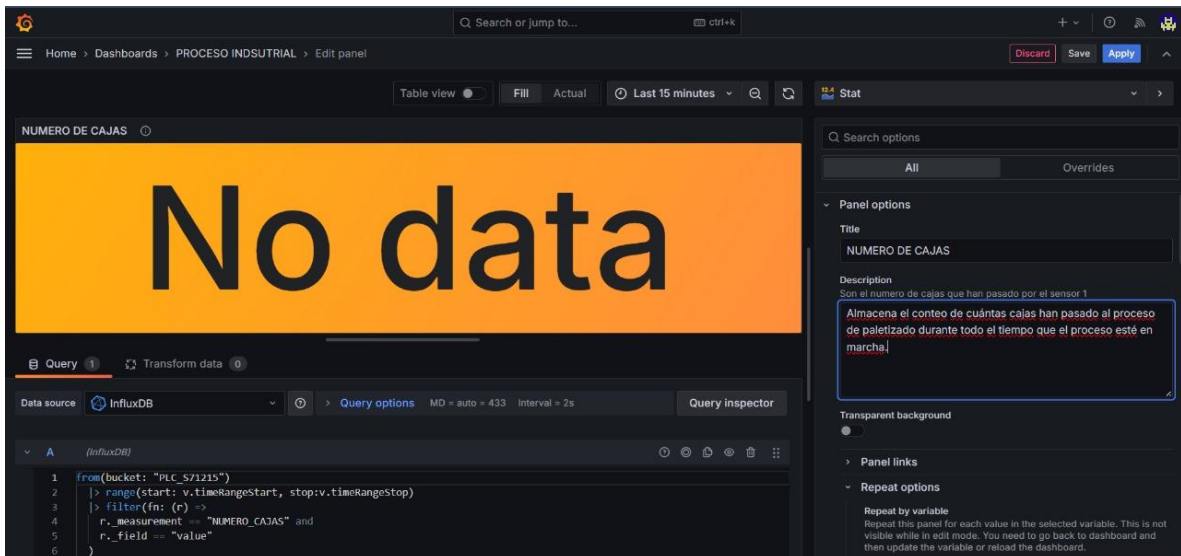


Figura 3.34. Ejemplo de la configuración de una visualización en Grafana.

El código de la parte inferior que se muestra en la Figura 3.34 Grafana propone un código por default que se encarga de realizar la petición a la base de datos de InfluxDB, para consultar el valor actual de la variable, los parámetros principales que incluye este código es el nombre del *Bucket*, el nombre de la variable (*Measurement*) y su valor (*value*).

El ejemplo de la figura mencionada es para agregar una visualización a la variable del número de cajas que pasan por el proceso.

El mismo procedimiento mencionado en el párrafo anterior se sigue para cada variable monitoreada, el único parámetro que cambia es *Measurement* y los respectivos gráficos para cada una. Por ello, el dashboard final queda diseñado de la siguiente manera, con la descripción de cada visualización y dividido en las siguientes áreas, como se muestra en las figuras 3.34 y 3.35.

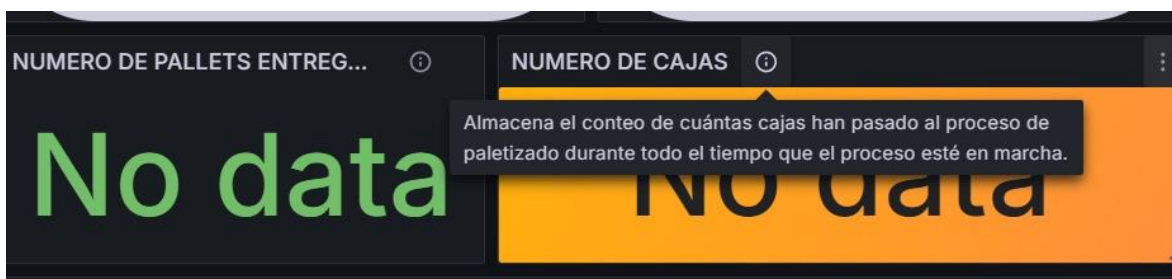


Figura 3.35. Ejemplo de la descripción de una visualización en Grafana.

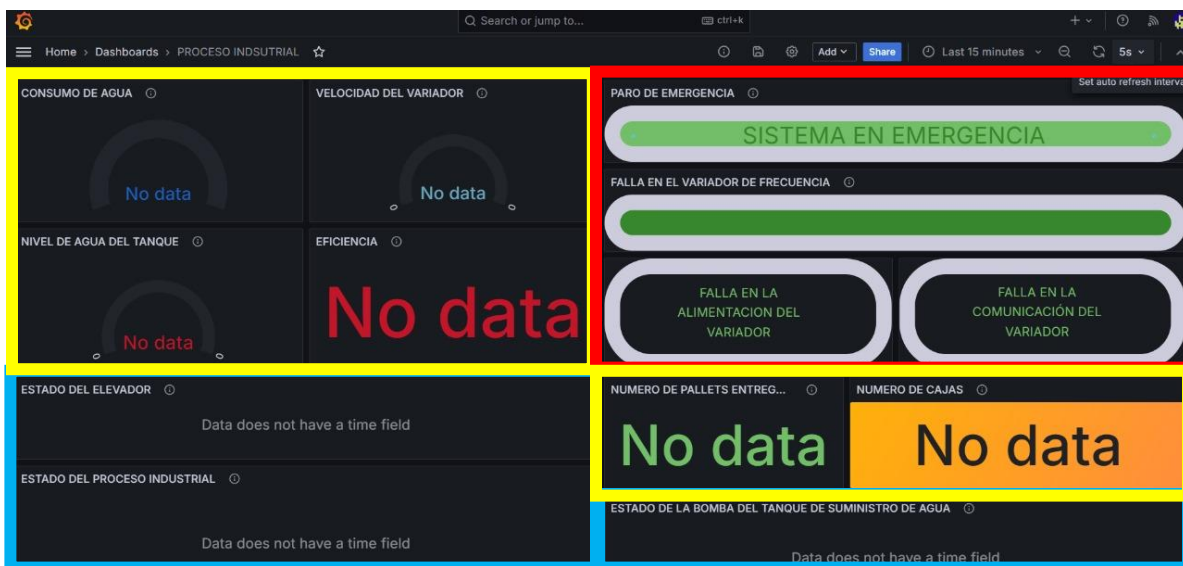


Figura 3.36. Dashboard diseñado en Grafana para monitorear el proceso industrial.

- Sección amarilla: Monitoreo de variables enteras y reales (nivel de agua, velocidad del variador, eficiencia, consumo de agua, número de cajas, número de palets).

- Sección azul: Monitoreo de variables discretas: (estado lógico [encendido/apagado] del sistema, del elevador, de la válvula del suministro de agua).
- Sección roja: Monitoreo de alarmas (paro de emergencia, falla de comunicación y falla de alimentación en el variador).

Se debe tomar en cuenta que las visualizaciones no se actualizan hasta que Node-RED se esté ejecutando y el proceso industrial esté funcionando.

3.6 Telegram

Como se mencionó al inicio del capítulo, el proceso industrial tiene la opción de colocar el selector en modo automático para permitir un control a distancia de la marcha, el paro y el reset. Este control se realiza por medio de la creación de una conversación dirigida por un *bot* en la aplicación de mensajería Telegram, la cual envía estos tres comandos a Node-RED y, haciendo uso de los nodos de *s7 out*, ahora se nos permita escribir en el bloque de datos del PLC (*DB500*) en las tres variables que controlan la marcha, el paro y el reset externo. El paro de emergencia queda completamente reservado para el tablero de Factory I/O y no depende de que si el proceso se encuentra en automático o manual.

Para crear el chat que será usado como medio de control del proceso industrial, la aplicación Telegram cuenta con un usuario llamado *BotFather*, este se encarga de crear este tipo de *bots* en un chat para una aplicación específica, dando también un token que funciona para enlazar esta aplicación con alguna externa como se muestra en la figura 3.37.

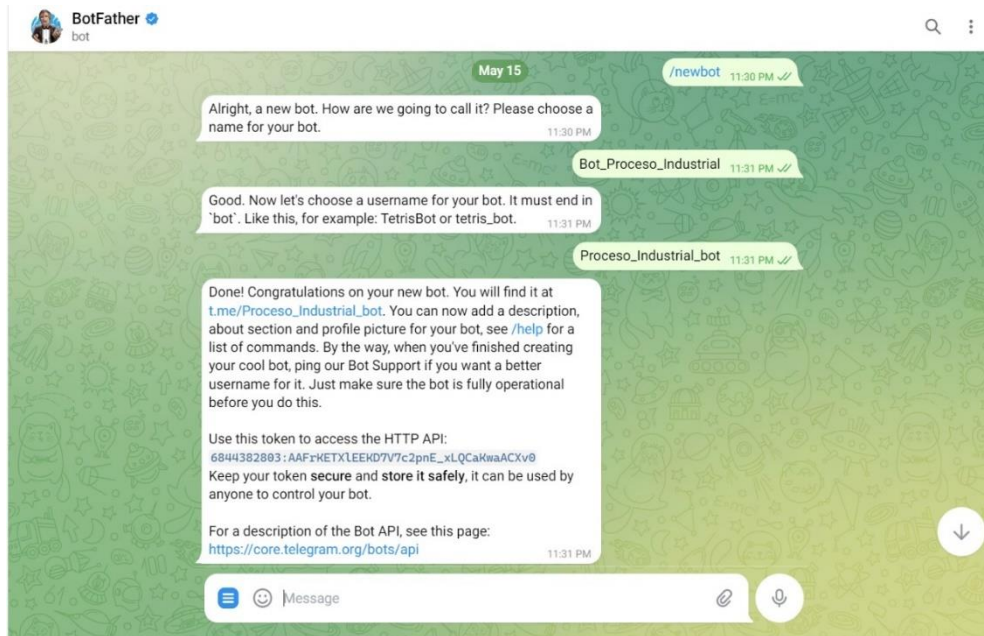


Figura 3.37. Creación del bot y el chat desde Telegram.

Al chat se le puede asignar una foto de perfil, un nombre, descripción y algunas otras funciones de personalización, para esta aplicación la descripción consiste en explicar los tres comandos posibles para realizar el control del proceso tal como se expone en la figura 3.38.

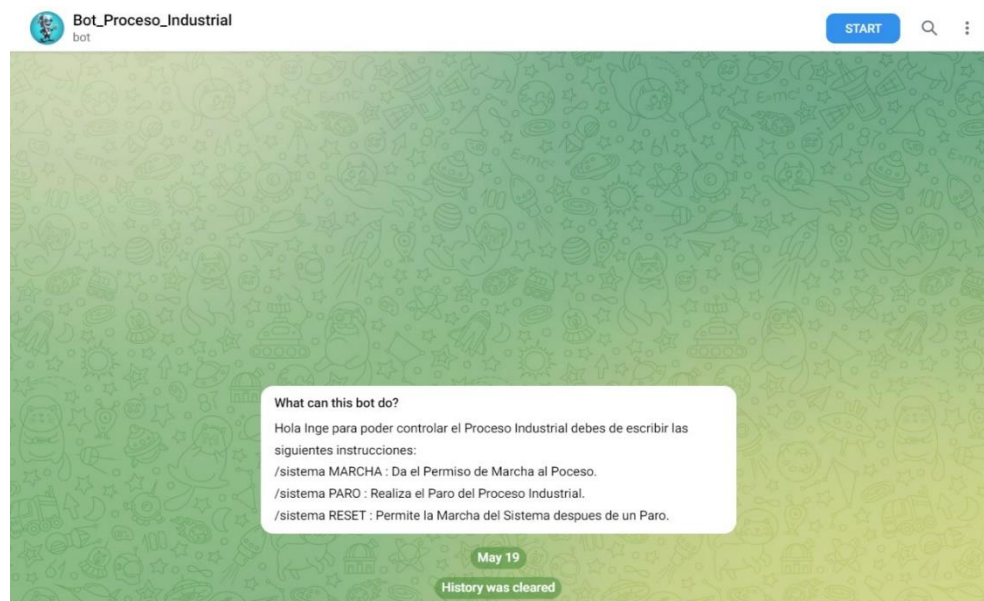


Figura 3.38. Interfaz del chat creado en Telegram para el control a distancia.

3.61 Conexión entre Telegram y Node-RED

Dentro de Node-RED se encuentra una librería llamada *node-red-contrib-telegrambot* que permite enlazar este software con la aplicación Telegram, esta librería contiene los siguientes nodos.

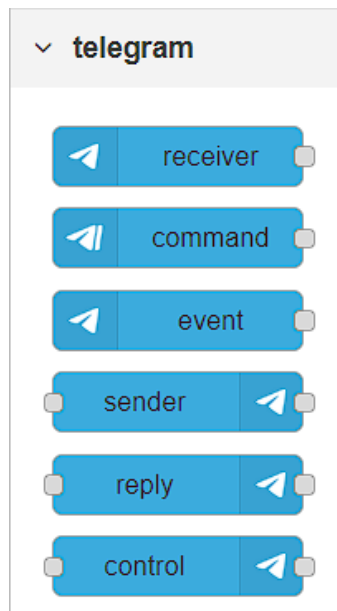


Figura 3.39. Nodos de la librería node-red-contrib-telegrambot.

- a) *Receiver*. Nodo que activa su salida cuando algún mensaje es recibido desde el chat.
- b) *Command*. Nodo que activa su salida cuando un comando específico es recibido desde el chat.
- c) *Event*. Nodo que activa su salida cuando un evento es recibido desde un chat.
- d) *Sender*. Nodo que envía el contenido del mensaje a su entrada hacia el chat.
- e) *Reply*. Nodo que es activado cuando alguien respondió a un mensaje específico.
- f) *Control*. Nodo que controla la configuración del nodo conectado a él.

Dado que la única tarea a realizar es recibir comandos desde el chat e interpretarlos para enviarlos al PLC y pueda controlar el proceso, el nodo a utilizar es *Command*. Su configuración se muestra a continuación.

Edit command node

Delete Cancel Done

Properties

Bot: Proceso_Industrial_bot

Name: CONTROL_PROCESO

Command: /sistema

Register at telegram server:

Description: Este comando es para poner en Marcha, Paro y

Language: (Optional language (see ISO-639-1). e.g.: de)

Scope: Default

Strict in group chats:

Has response output:

Enabled

Figura 3.40. Primera parte de la configuración del nodo Command en Node-RED.

Edit command node > **Edit telegram bot node**

Delete Cancel Update

Properties

Bot-Name: Proceso_Industrial_bot

Token: 6844382803:AAFkETXIEEKD7V7c2pnE_xLQCaKwaACXv0

Tip: If you don't have a token yet, you can create a new one here: @BotFather.

Users: (Optional list of authorized user names e.g.: hugo,sepp,egon)

ChatIds: (Optional list of authorized chat-ids e.g.: -1234567,2345678,-3456789)

Server URL: (Optional URL for proxying and testing e.g.: https://api.telegram.org)

Update Mode: Polling

Polling Options:

Poll Interval: 300

Tip: Polling mode is very robust and easy to set up. Nevertheless it

Enabled 1 On all flows

Figura 3.41. Segunda parte de la configuración del nodo Command en Node-RED.

Los parámetros por configurar son: el nombre del nodo que en este caso es *CONTROL_PROCESO* y la opción *Command* que permite asignar un menú de comandos para habilitarlo en el chat y evitar que algún otro mensaje mandado por error afecte el desempeño o la comunicación del proceso (Figura 3.39). Los demás parámetros permanecen por defecto a excepción de la asignación del *Bot*, el cual se configura de manera independiente, agregando el token asignado por Telegram y el nombre correcto del *bot* dado de alta en la misma aplicación (Figura 3.40).

Después de configurar el nodo de la comunicación con Telegram, es necesario tratar el dato recibido desde el chat para enviar el valor correcto hacia el PLC usando el nodo *s7 out*, por ello se requiere usar una función de intercambio de datos que permite tres salidas, una para cada comando recibido. Posteriormente, la lógica de control debe tener en cuenta que, al no ser señales instantáneas -como las de un botón pulsador- sino permanentes, se debe conmutar al estado anterior para cambiar de marcha a paro o viceversa. Es decir, si se requiere que el sistema pase de paro a marcha es necesario escribir un 0 en la variable paro y un 1 en la variable marcha, unido a un 1 en el reset. De igual manera tomar en cuenta que el paro de emergencia debe poner la marcha a 0 y el paro a 1, para que el sistema pueda reiniciar en cualquier momento de manera remota.

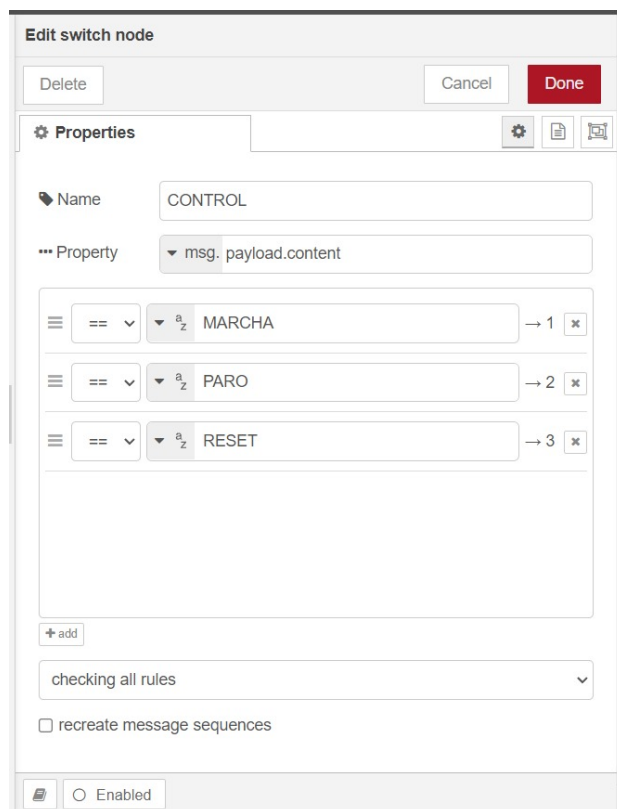


Figura 3.42. Configuración del nodo de intercambio de datos que permite tres salidas.

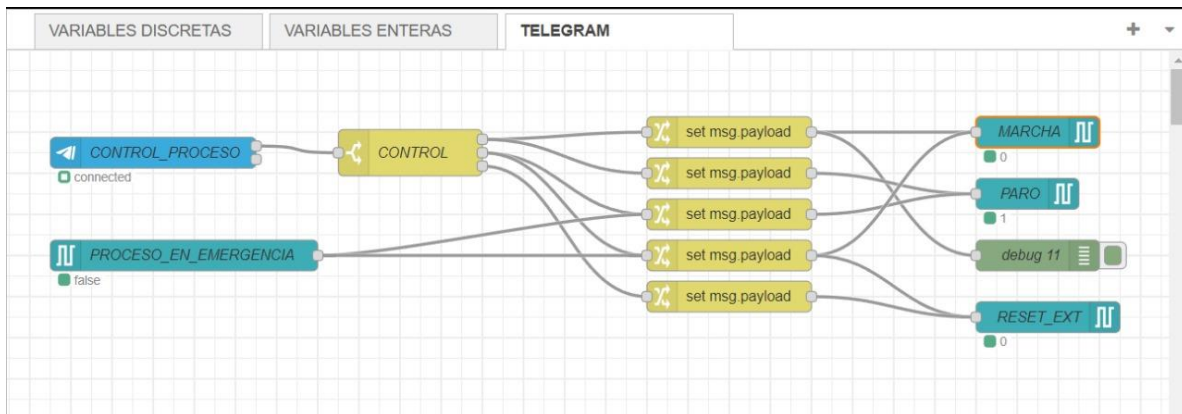


Figura 3.43. Red de nodos para controlar el proceso mediante Telegram.

Los cinco nodos llamados *set.msg.payload* asignan el valor de 1 o 0 al contenido del mensaje para que posteriormente se envíen al PLC por medio de los nodos de comunicación S7. Los comandos disponibles en Telegram para controlar el proceso son:

1. /sistema MARCHA
2. /sistema PARO
3. /sistema RESET

Esta es la última parte del diseño del sistema SCADA, se ha creado el proceso industrial en Factory I/O, se ha enlazado a un PLC físico que lo controla -junto con un variador de frecuencia- y guarda las variables a monitorear y controlar en un bloque de datos general sin acceso optimizado (*DB500*), al cual accederá Node-RED para recopilar los valores de las variables, tratarlos y adecuarlos para enviarlos a una base de datos creada en InfluxDB, misma que los almacena y permite su lectura desde Grafana, en donde se añaden visualizaciones personalizadas para cada variable en un dashboard. Finalmente, se crea y enlaza un chat de Telegram que permite el control remoto del proceso por medio de un *bot* comunicado con Node-RED que acopla las señales correctamente y las escribe en el PLC para lograr su cometido.

CAPÍTULO IV: IMPLEMENTACIÓN

Funcionamiento del sistema SCADA

Implementación física

El sistema SCADA completo queda conformado tal y como lo muestra el diagrama a bloques de la siguiente figura, destacando la división entre la Tecnología Operativa (OT) y la Tecnología de la Información (IT) unidos por Node-RED mediante diferentes protocolos de comunicación.

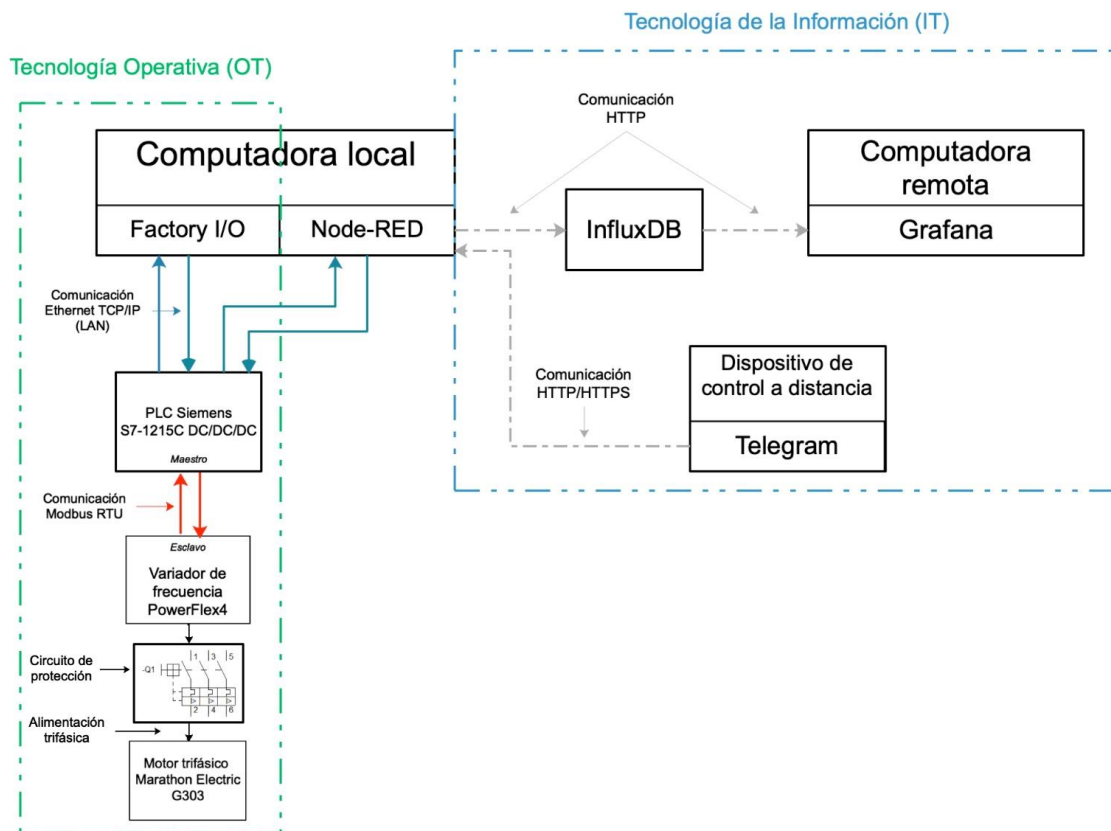


Figura 4.1. Diagrama a bloques del sistema SCADA diseñado.

En la implementación física del sistema SCADA de la Figura 4.2 es posible apreciar cada parte del diagrama a bloques anterior, logrando una comunicación efectiva y un desempeño

adecuado de acuerdo con lo propuesto en los objetivos del proyecto. A continuación, se presenta parte del desempeño y sus funciones más importantes.

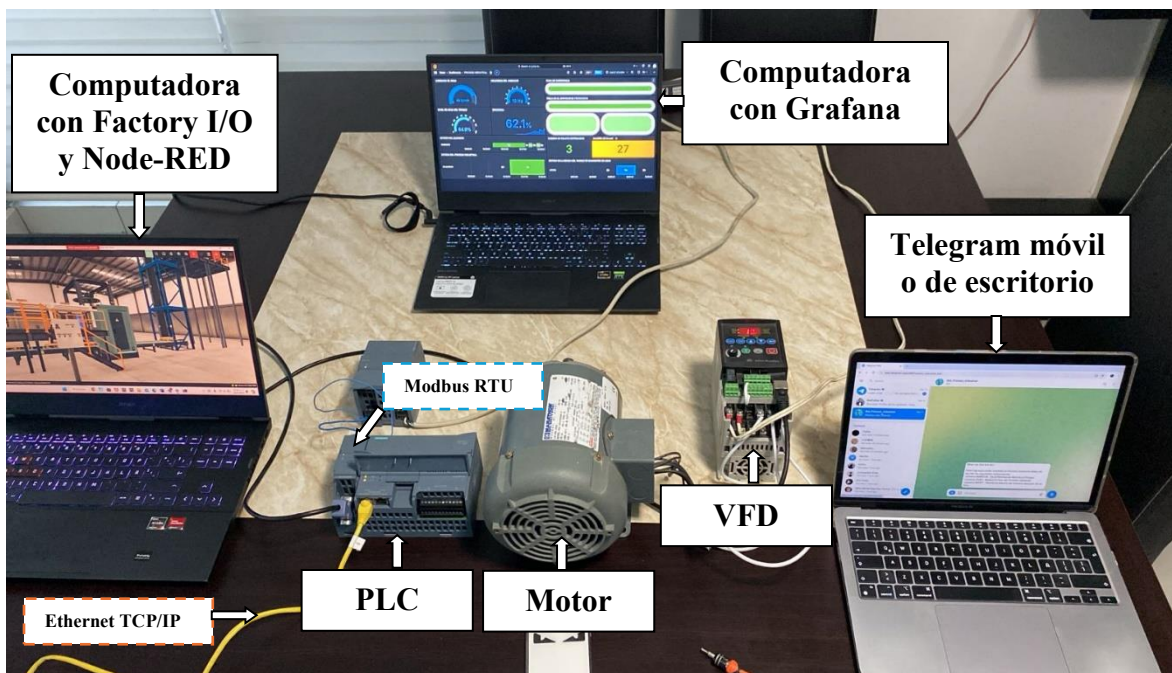


Figura 4.2. Sistema SCADA implementado físicamente.

Prueba I: Desempeño normal del sistema

Después de la conexión correcta de todos los elementos del sistema SCADA, la primera prueba a realizar es la puesta en marcha del proceso industrial desde Factory I/O en modo manual, es decir, desde el tablero dentro del software.



Figura 4.3. Proceso industrial en marcha manual desde Factory I/O.



Figura 4.4. Subproceso de paletizado en funcionamiento.

En un funcionamiento continuo y sin ningún contratiempo, los subprocesos trabajan sincronizadamente desempeñando cada una de sus tareas y aumentando la eficiencia del proceso.



Figura 4.5. Subproceso del tanque de suministro de agua en funcionamiento.



Figura 4.6. Subproceso del elevador y entrega de palet en funcionamiento.

Así mismo, se obtienen los siguientes datos en el monitoreo de variables desde Grafana.



Figura 4.7. Dashboard en Grafana con el sistema en marcha.

Como se observa en la figura anterior, el proceso después de un tiempo de trabajo y haber entregado un total de 13 palets, 89 cajas y una eficiencia del 83.9%, no presenta ninguna falla en el funcionamiento, ni operativo ni de comunicación, por ello es por lo que la eficiencia continúa en ascenso y las alarmas de la parte superior derecha no cambian su estado lógico. En las gráficas de la parte inferior derecha e izquierda se observan las variables discretas y su cambio de valor: si el elevador asciende o desciende, si el sistema está en marcha o en paro, así como si el tanque de suministro de agua se está llenando o no. Las gráficas de la parte superior derecha muestran el monitoreo de las variables enteras y reales: consumo de agua, velocidad del variador de frecuencia, nivel actual del tanque de suministro de agua y eficiencia del proceso.



Figura 4.8. Sistema SCADA físico en funcionamiento normal.

Prueba II: Paro de emergencia

La segunda prueba consiste en poner el proceso industrial en paro de emergencia e interrumpir la marcha de cada subprocesso, detener el variador de frecuencia y encender una advertencia en una de las visualizaciones de Grafana, así como dentro de Factory I/O.



Figura 4.9. Paro de emergencia del proceso industrial.

Como se mencionó en el capítulo anterior, el proceso en paro de emergencia detiene todos los subprocesos, reinicia las marchas de los actuadores y no permite la marcha general ni local ni remota, hasta que la seta de emergencia se libere y se presione el botón de reset. Dentro de Grafana también se observa la alerta del proceso en emergencia, además de que el consumo de agua y velocidad del variador caen a cero y la eficiencia comienza a descender.



Figura 4.10. Alerta de paro de emergencia del proceso en Grafana.

Prueba III: Fallas en el variador de frecuencia

Para efectos prácticos, las fallas disponibles en el variador de frecuencia para poner a prueba son: falta de alimentación eléctrica y error en la comunicación por su puerto RJ45.

La primera consiste en desconectar el cable de corriente de entrada del variador e inmediatamente se presenta el código *F004* en su pantalla, después de algunos segundos éste se apaga y el proceso industrial se detiene por completo y al mismo tiempo, se enciende una alarma en una visualización de Grafana.



Figura 4.11. Falla en la alimentación del variador de frecuencia.



Figura 4.12. Alerta de falla de alimentación del variador en Grafana.

La segunda falla presente en el variador consiste en perder la comunicación física vía Modbus RTU mostrando en su pantalla el código *F081*, inmediatamente el proceso de Factory I/O se detiene por completo y se enciende la alerta de falla por comunicación en Grafana.



Figura 4.13. Falla en la comunicación Modbus RTU del variador de frecuencia.



Figura 4.14. Alerta de falla de comunicación del variador en Grafana.

Prueba IV: Control del proceso desde Telegram

Una vez comprobado el funcionamiento del proceso de manera local, se procede a controlarlo de manera remota mediante el chat de Telegram creado, para ello se requiere colocar el selector del tablero de Factory I/O en automático y enviar el primer comando desde la aplicación de mensajería.



Figura 4.15. Proceso industrial en modo automático para control a distancia.

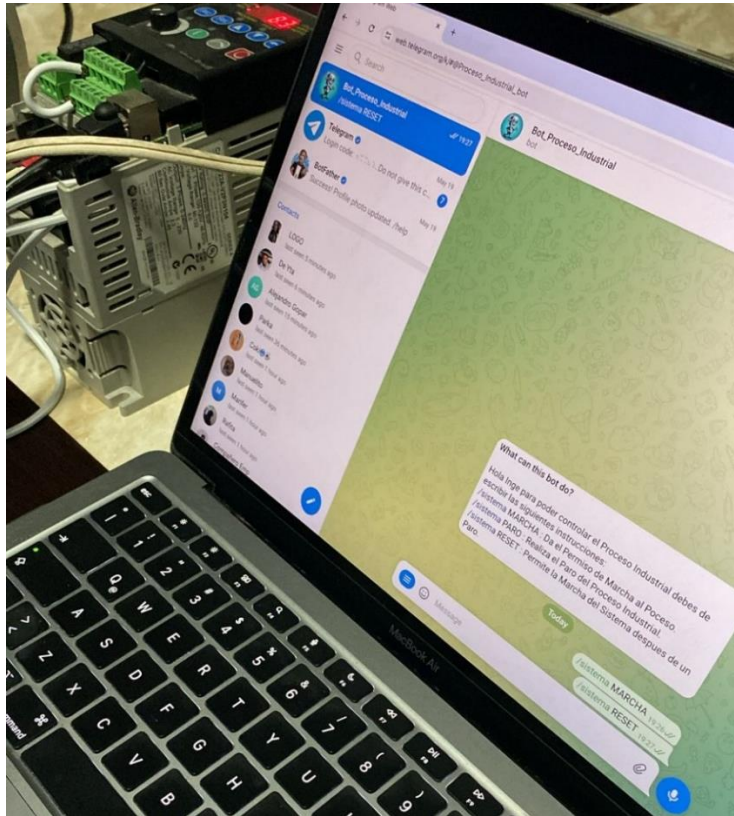


Figura 4.16. Comando de marcha y reset enviado desde Telegram.

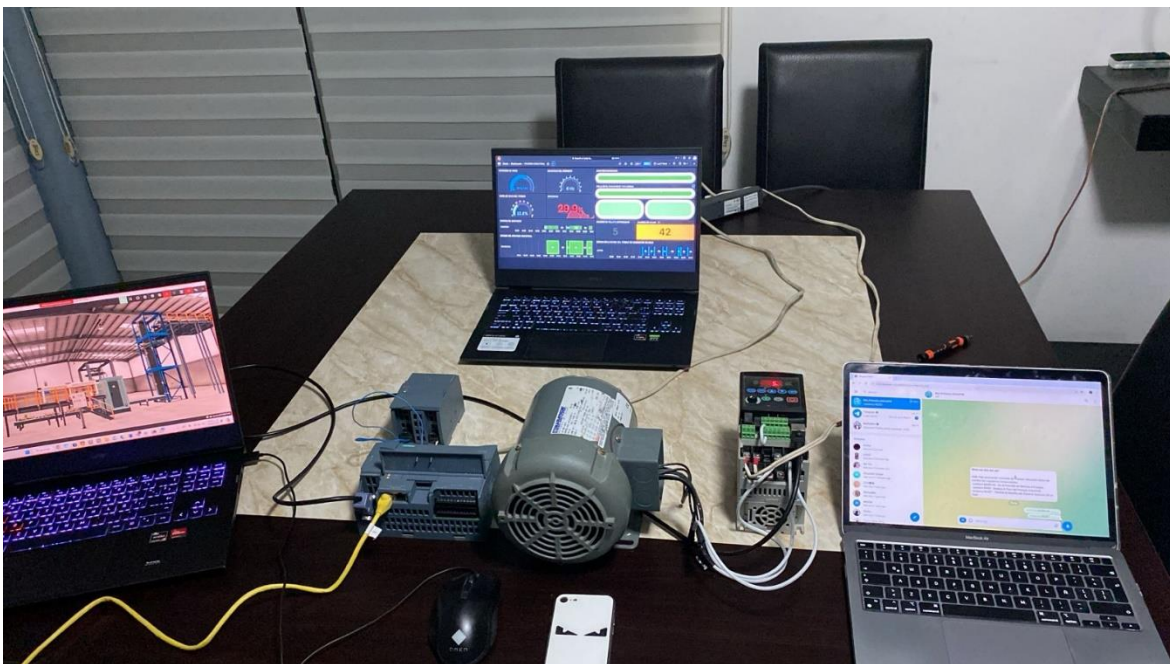


Figura 4.17. Sistema SCADA en funcionamiento controlado de manera remota.

El paro del sistema se realiza mediante el envío del comando `/sistema PARO` en el chat de Telegram, este es un paro normal que pone en pausa la secuencia del proceso y permite la posibilidad de reiniciarlo nuevamente con los comandos de marcha.

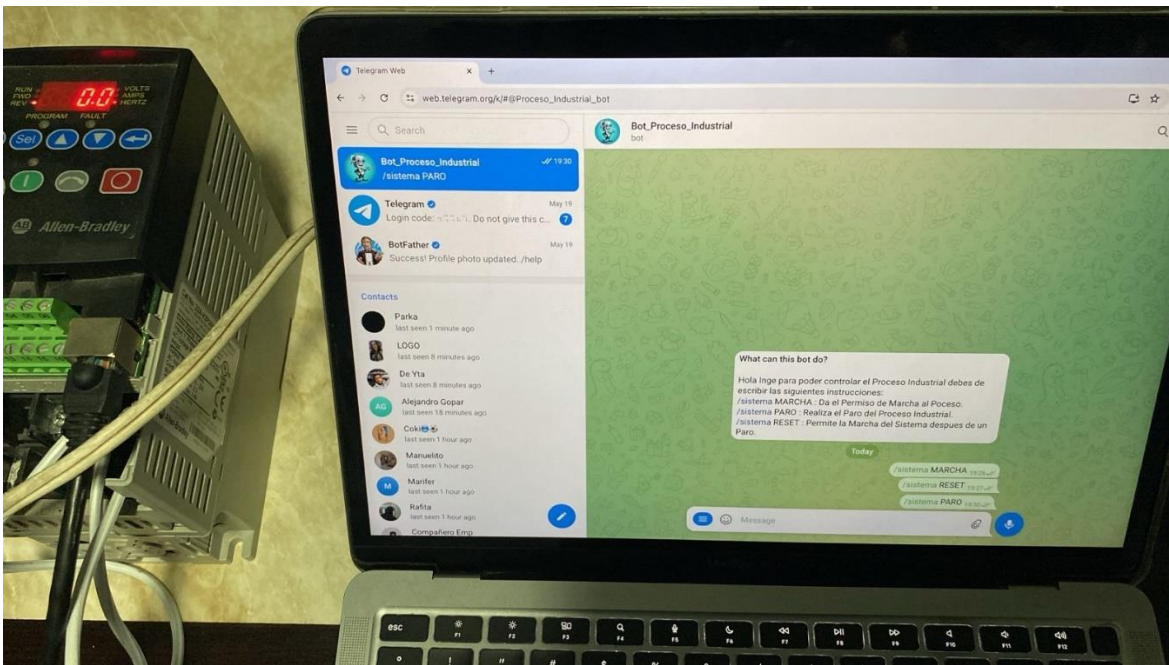


Figura 4.18. Comando de paro enviado desde Telegram.

Después de un largo tiempo de trabajo, aproximadamente veinte minutos, el proceso logra establecer una eficiencia de más del 90%, tomando en cuenta que no hay ningún inconveniente que haga detener la producción al menos durante un largo periodo de tiempo. Esto nos muestra que el proceso industrial, los protocolos de comunicación y el sistema de monitoreo en condiciones cercanas a las ideales se comporta de manera adecuada de acuerdo al objetivo planteado.



Figura 4.19. Monitoreo del proceso después de un largo periodo de funcionamiento.

Prueba V: Reporte de funcionamiento del sistema SCADA

Grafana ofrece la opción de compartir la información contenida en el dashboard a manera de un reporte técnico mediante diferentes medios, tales como correo electrónico, captura de pantalla, archivo PDF o la URL. La función principal es comunicar los resultados del monitoreo en un periodo de tiempo deseado con otras personas con el fin de tomar decisiones en cuanto al desempeño del proceso industrial. El procedimiento para realizar este reporte comienza con elegir la opción *Share* en el dashboard y dentro de ella la pestaña *Report* y crear uno nuevo, tal y como se muestra en la figura siguiente.

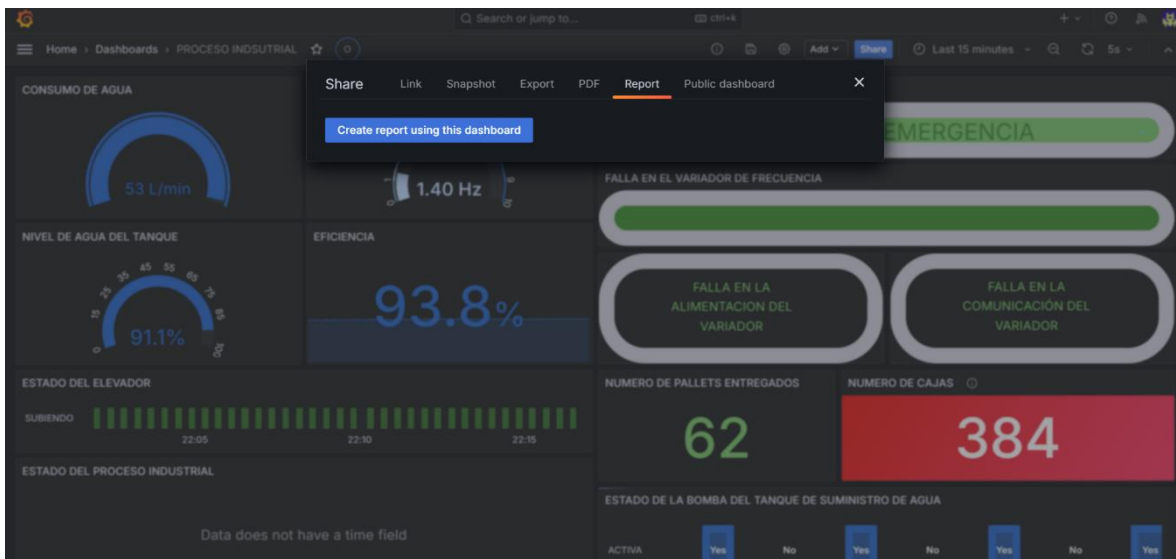


Figura 4.20. Creación del reporte del proceso en Grafana

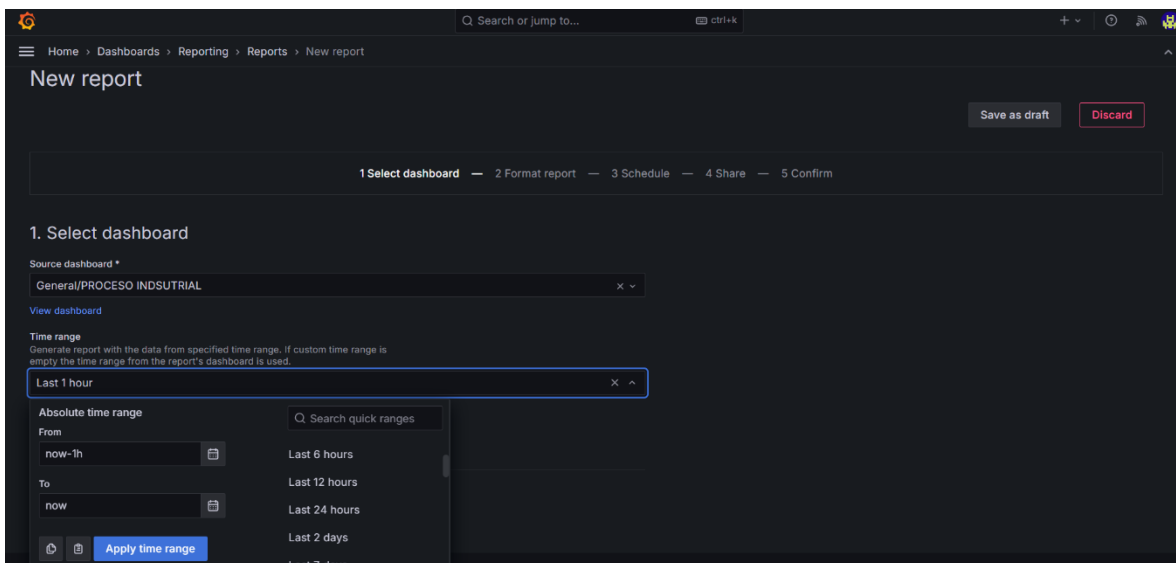


Figura 4.21. Rango de tiempo del reporte de datos en Grafana.

La siguiente opción por tomar en cuenta es elegir el dashboard y el periodo de tiempo del cual se quiere reportar el funcionamiento del proceso, en este caso, se eligen los datos almacenados en la última hora (Figura 4.21). Posteriormente se determina el formato y estilo del reporte en PDF.

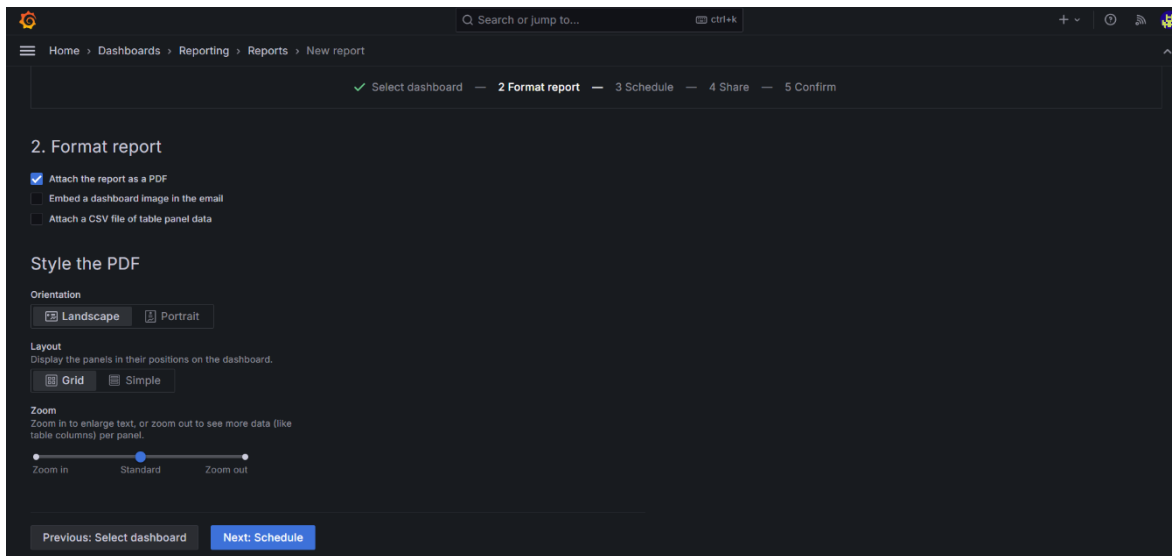


Figura 4.22. Formato del reporte en Grafana.

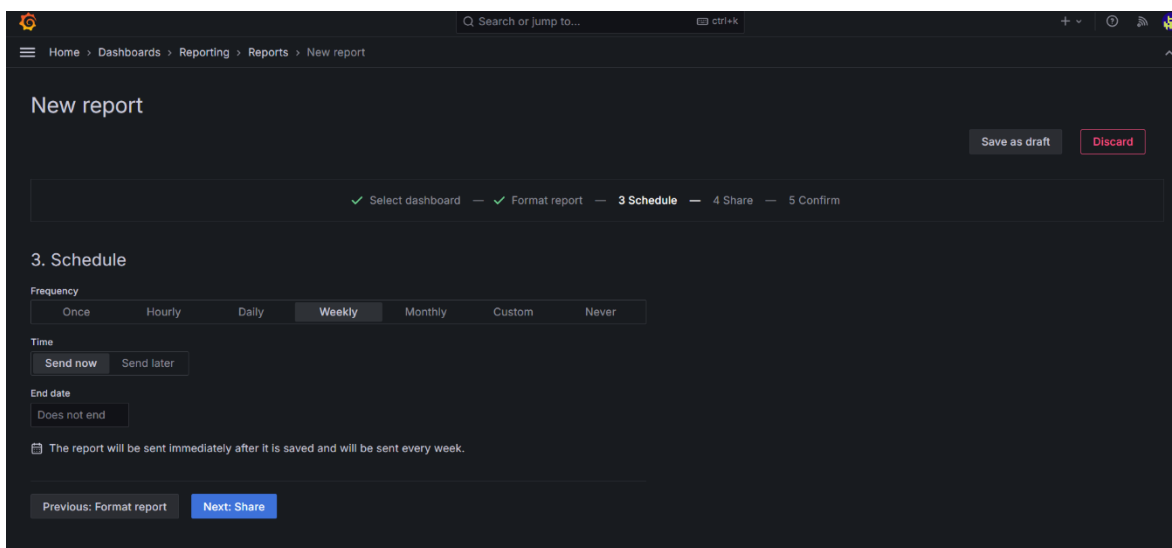


Figura 4.23. Cronograma del reporte en Grafana.

La Figura 4.23 muestra la configuración del cronograma del reporte, es decir, con qué frecuencia será enviado a su destino, hasta qué fecha se dejará de enviar y el momento para enviarlo. En este caso, se enviará semanalmente a la hora propuesta sin fecha de fin definida.

Finalmente, el cuarto es paso es elegir un nombre para el reporte, las direcciones de correo electrónico de los destinatarios, así como del propietario del dashboard, esto último con el fin de que las personas que reciben el reporte sepan a dónde y con quién comunicarse en caso de alguna situación. Como parte final de la configuración, se ofrece la opción de escribir un mensaje a los destinatarios para crear un ambiente de comunidad.

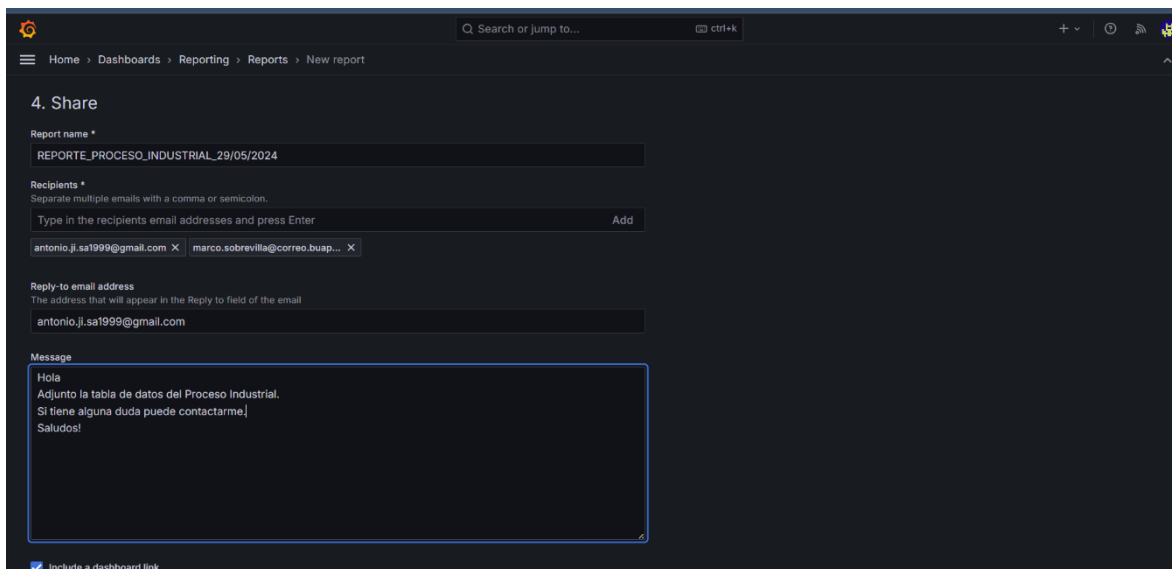


Figura 4.24. Elección del destino del reporte en Grafana.

Después de confirmar que toda la configuración y destinatarios son correctos, se comparte el reporte y ahora se observa en el menú principal que existe un reporte agendado para envío, cada semana a dos destinatarios hasta que se deje de compartir.

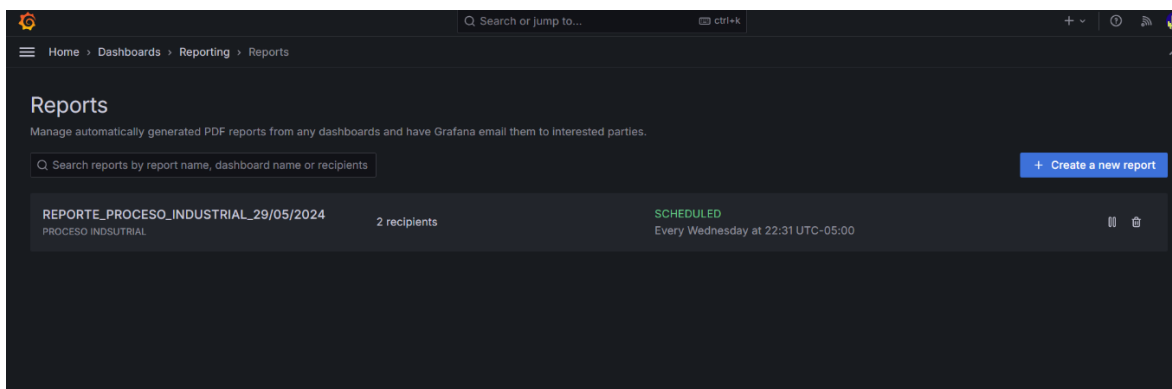


Figura 4.25. Envío del reporte del proceso en Grafana.

Se comprueba que el reporte creado es recibido por los destinatarios en su correo electrónico satisfactoriamente, como se muestra en la Figura 4.26.

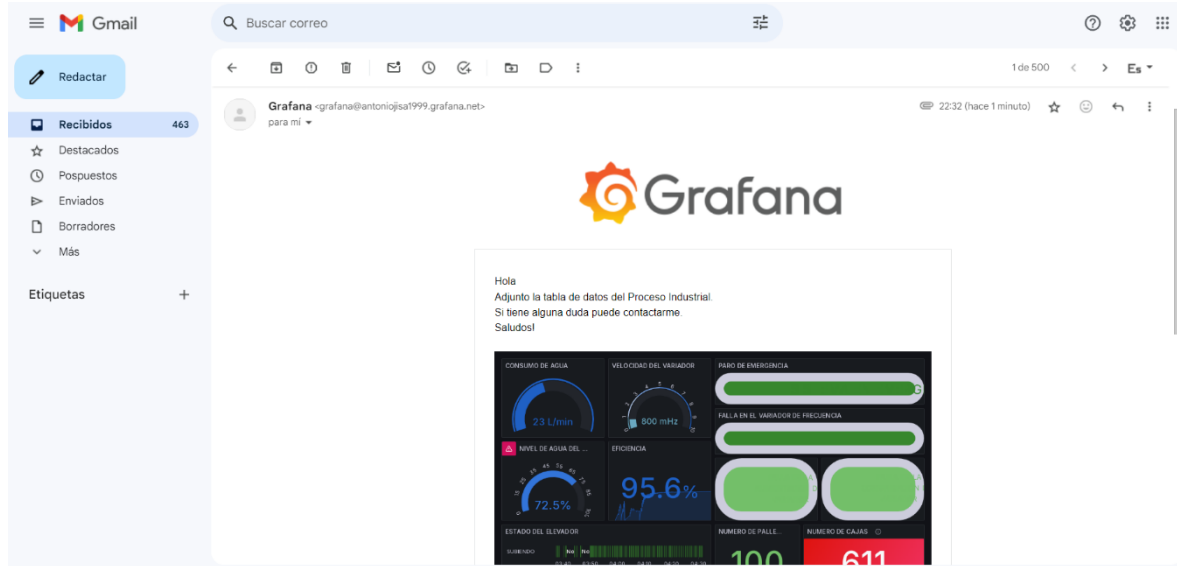


Figura 4.26. Recepción del reporte de Grafana vía correo electrónico.

Por otro lado, la plataforma InfluxDB que almacena y administra la base datos, ofrece la opción de crear un archivo en formato CSV para poder reportar la cantidad de información que se ha resguardado de cada variable durante un periodo de tiempo determinado, y así poder canalizar dicha información a los departamentos correspondientes para interpretar y analizar el comportamiento del sistema, así como actuar de manera preventiva ante una posible falla que ponga en riesgo la eficiencia del sistema.

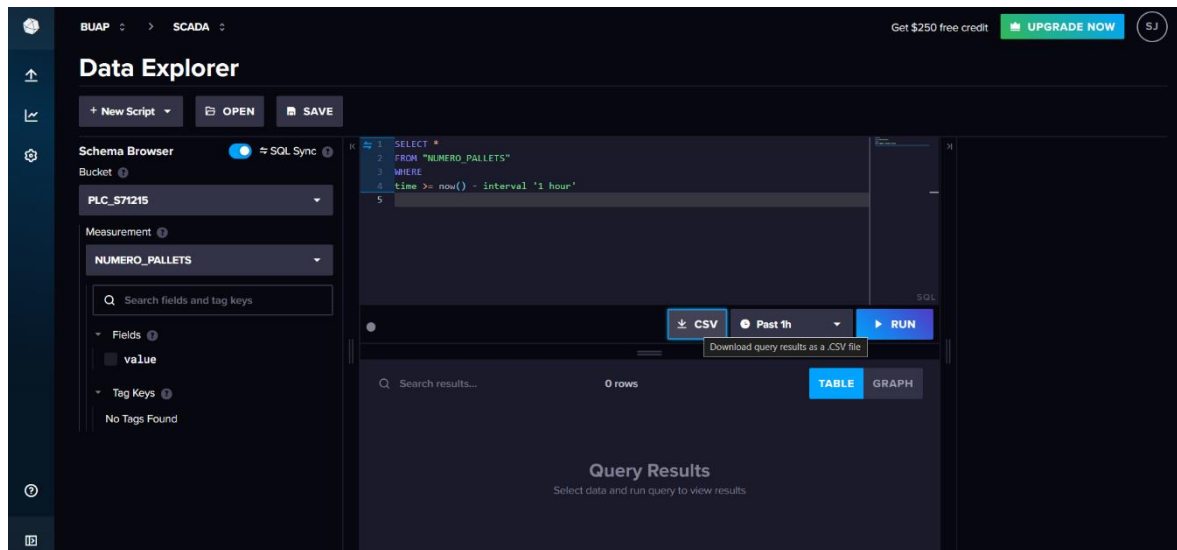


Figura 4.27. Creación del reporte de almacenamiento de datos en InfluxDB.

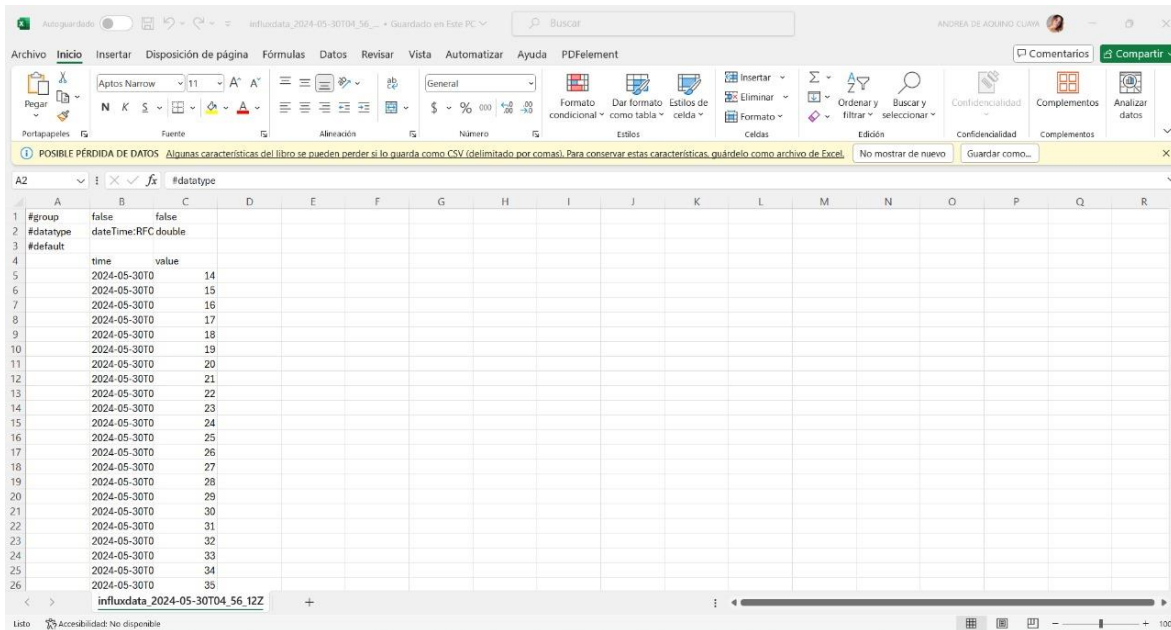


Figura 4.28. Ejemplo de reporte de datos generado por InfluxDB.

La figura 4.28 muestra un ejemplo de los valores almacenados en la base de datos del número de palets entregados, se puede apreciar que se adjunta el valor y la fecha en la cual se recibieron y guardaron dichos datos. Es decir, se pueden generar estos archivos de tipo CSV para cada variable y canalizarlos al departamento de análisis de datos para hacer una revisión del comportamiento pasado, presente y posiblemente una predicción del futuro de este.

Ventajas del Proyecto

1. El monitoreo en tiempo real del proceso industrial a través de Grafana, facilitará la detección inmediata de problemas y una rápida toma de decisiones.
2. Mediante un Dashboard el proyecto simplificará el monitoreo de datos a una sola pantalla, lo cual permitirá una supervisión efectiva de todos los estados del proceso industrial.
3. La facilidad de operación del proyecto permitirá un tiempo mínimo de capacitación para el usuario.
4. La gestión de usuarios integrada por Grafana permite la restricción de funciones a usuarios no dados de alta en el sistema.
5. Mediante Grafana la gestión de usuarios se dividirá en: Lectores, administradores y editores lo que permitirá a la empresa asignar esos cargos a personal estratégico.
6. Grafana permitirá al usuario interactuar y personalizar el dashboard de acorde a las necesidades específicas del proyecto.
7. El acceso al Dashboard de Grafana se podrá dar desde cualquier navegador, lo que permitirá a la empresa tener una amplia variedad de dispositivos sin importar su sistema operativo.
8. Cada panel del dashboard contiene la descripción del funcionamiento de la variable monitoreada de modo que facilitará al usuario conocer a detalle cada visualización.
9. El sistema SCADA proporciona información del proceso industrial mediante reportes programados que pueden ser compartidos con el personal indicado por la empresa.
10. La empresa podrá autorizar de manera remota a una persona para controlar el proceso industrial a través de Telegram.
11. Al estar desarrollado en un software de código abierto los costos por licenciamiento serán bajos.
12. El sistema SCADA permitirá a la empresa visualizar el comportamiento del proceso industrial en diferentes dispositivos, lo que facilitará la supervisión en tiempo real desde computadoras, tablets o smartphones. Esto permitirá a los operadores y

gerentes tomar decisiones informadas de manera rápida y eficiente, independientemente de su ubicación, mejorando así la productividad y reduciendo tiempos de respuesta ante posibles fallas o desviaciones en el proceso.

13. Telegram permitirá el control del proceso industrial (Marcha, Reset, Paro) a distancia.
14. En caso de una interrupción en la conectividad remota o la comunicación inalámbrica el proceso industrial puede mantenerse controlado y supervisado mediante una conexión por cable.

Desventajas del Proyecto

1. Para establecer una conexión remota con el sistema SCADA se requiere una red estable de internet.
2. Si las configuraciones de seguridad no están determinadas correctamente, los usuarios no autorizados podrían acceder al control del sistema.
3. Si el sistema operativo del ordenador que controla la comunicación entre la nube y los datos del PLC está obsoleto y no recibe actualizaciones de seguridad, puede ser vulnerable a ataques cibernéticos.

CONCLUSIONES

Actualmente los sistemas SCADA como el que se hizo en este proyecto juegan un papel importante en cualquier proceso industrial dentro de cualquier empresa, ya que a través de ellos se permite la supervisión continua y en tiempo real de dichos procesos, lo que ayuda a detectar y corregir anomalías dentro de todos los componentes para mejorar la eficiencia operativa de los diferentes proyectos a través de un tiempo mínimo de reacción. La importancia del diseño del sistema SCADA en el proyecto radica en la capacidad para combinar múltiples tecnologías y protocolos en un sistema integral que optimiza la operación y monitoreo de un proceso industrial. Por otra parte, el uso de nuevas tecnologías de información permitió una mayor eficiencia de monitoreo para el proceso industrial, lo que conlleva un ahorro económico al no comprar software y hardware que normalmente ocupan las grandes industrias que tienen el capital suficiente para adquirirlo.

Otro punto muy importante es que a través de las tecnologías de información implementadas se redujo el tiempo de transferencia de datos, se obtuvo una mejor visualización y la posibilidad de acceder y controlar el sistema de manera remoto.

Uno de los retos principales fue conectar las nuevas tecnologías con un proceso industrial tradicional el cual este compuesto de un PLC, un variador de frecuencia y un motor trifásico a través de comunicación alámbrica. Por otra parte, el PLC jugo un papel importante ya que su capacidad para automatizar tareas, reducir costos y adaptarse a las necesidades del ambiente lo hizo una herramienta esencial por su flexibilidad y escalabilidad que le ofrece al proyecto. Por otra parte, este proyecto me ayudo a desarrollar habilidades en torno a los protocolos de comunicación y a los estándares de seguridad para trabajar con equipo industrial.

A pesar de su antigüedad en el entorno industrial, el protocolo de comunicación Modbus sigue siendo ampliamente utilizado debido a su capacidad para manejar eficientemente los volúmenes de tráfico de datos que demandan los sistemas actuales. Además, al ser un protocolo de acceso libre, los costos asociados son considerablemente menores, lo cual fue muy benéfico para cumplir con el objetivo principal de este proyecto. Sin embargo, otro objeto de estudio para este proyecto es desarrollar un campo de gestión centralizado en proteger la red industrial contra el acceso no autorizado y el robo de información.

Al tener todo el proceso industrial conectado mediante el protocolo modbus una de las tareas a cumplir era investigar como enviar los datos emitidos del DB500 de TIA portal a la nube, en esa investigación nos encontramos con Node-RED que destaco y jugo un papel muy importante para la elaboración de este proyecto, fungió un puente entre el proceso industrial y la base de datos de InfluxDB. Simplificando el proceso de desarrollo a través de sus nodos

permitiendo una rápida interacción, facilitando la conectividad y envío de datos en nuestro sistema SCADA.

Grafana también evito más costos en el proyecto, al ser de libre acceso cualquier persona tiene ingreso a ella para la visualización de datos de sus respectivos proyectos.

Actualmente el análisis de datos es crucial en diversos ámbitos por favorables razones, gracias a ello se optimizan recursos en base a decisiones informadas y se identifican tendencias a través del rendimiento que permite evaluar el éxito de los procesos industriales. Por ello el proyecto a través de Influx DB y Grafana proporciona tablas de datos por lapsos definidos de tiempo.

El proyecto fue desarrollado utilizando dispositivos industriales que son ampliamente utilizados en la industria de automatización de procesos y planteando un caso de uso real en el entorno de Factory IO simulando un proceso industrial de una pequeña o mediana empresa cumpliendo así con los objetivos propuestos en esta tesis.

Importancia del Proyecto para el Mercado Industrial.

- La importancia de este proyecto al generar datos para su estudio lo avalan muchas investigaciones alrededor del mundo, ya que revistar como la de Harvard en su artículo **“The Evolution Making: How Leading Organization are Adopting a Data – Drive Culture”** publicado en el 2012. Destacan que cuando las pymes integran el análisis de datos en su toma de decisiones estas experimentan una mayor eficiencia operativa, toman decisiones un 5% más rápido y tienen un mejor rendimiento.
- La encuesta **“The Value of business intelligence sor SMEs”** publicada en el 2020 obtuvo como resultado que las pymes que generan datos y los analizan obtienen un aumento promedio del 8-10% en ingresos y una reducción del 10% en costos operativos.

El poder analizar los datos generados por un sistema SCADA ha sido ampliamente estudiado y se ha demostrado como un factor clave en el crecimiento y éxito de las pequeñas y medianas empresas. Los estudios y encuestas mencionados destacan que las pymes que adoptan una cultura de decisiones basadas en datos mejoran su rentabilidad, eficiencia y competitividad, permitiéndoles optimizar operaciones y así responder rápidamente a las necesidades del mercado. Por eso a través de los objetivos planteados se permite que el proyecto sea accesible monetariamente para las pequeñas y medianas empresas mediante la investigación y los resultados obtenidos en esta tesis.

BIBLIOGRAFÍAS

- [1] National Institute of Standards and Technology. (s.f.). Supervisory Control and Data Acquisition (SCADA) - Glossary | CSRC. https://csrc.nist.gov/glossary/term/supervisory_control_and_data_acquisition
- [2] Penin, A. R. (2012). Sistemas SCADA (3.a ed.). Marcombo.
- [3] National Institute of Standards and Technology. (s.f.). Master Terminal Unit (MTU) - Glossary | CSRC. National Institute of Standards and Technology. Recuperado 4 de septiembre de 2023, de https://csrc.nist.gov/glossary/term/master_terminal_unit
- [4] National Institute of Standards and Technology. (s.f.). Remote Terminal Unit - Glossary | CSRC. Recuperado 4 de septiembre de 2023, de https://csrc.nist.gov/glossary/term/remote_terminal_unit
- [5] Babel, W. (2022). Industry 4.0, China 2025, IoT: The Hype Around the World of Automation. Springer.
- [6] Tello, S. S. (2013). Sistemas automáticos industriales de eventos discretos (1.a ed.). Alfaomega.
- [7] Sicma21. (2023, marzo 8). Qué es un PLC, cómo funciona y por qué se utilizan. Soluciones Integrales para la Industria 4.0. Recuperado 6 de septiembre de 2023, de https://www.sicma21.com/que-es-un-plc/#Como_funciona_un_PLC
- [8] ABB. (s.f.). Qué es un variador de frecuencia. Definición, cómo funciona, características y ventajas. Convertidores. Recuperado 7 de septiembre de 2023, de <https://new.abb.com/drives/es/que-es-un-variador>
- [9] Martínez, L., Guerrero, V., & Yuste, R. (2009). Comunicaciones industriales. Alfaomega.
- [10] Bradley, A. (2017). Powerflex 4 adjustable frequency ac drive frn 7.xx [Manual de software informático].
- [11] MHEducation. (2012). Motores eléctricos. Mc Graw Hill.

ANEXOS

ANEXO 1

Tabla de variables de entrada del PLC					
Nombre	Descripción	Tipo	Rango	Unidad	Dirección
START	Botón de marcha	Pulsador	Encendido/Apagado	×	%I2.0
STOP	Botón de paro	Pulsador	Encendido/Apagado	×	%I2.1
MANUAL	Selector manual	Selector	Encendido/Apagado	×	%I2.2
AUTO	Selector automático	Selector	Encendido/Apagado	×	%I2.3
EMERGENCY_STOP	Seta de emergencia	Pulsador	Encendido/Apagado	×	%I2.4
PAL_Box_Entry	Detecta caja al final de la cinta transportadora	Sensor fotoeléctrico difuso	Encendido/Apagado	×	%I2.5
PAL_Pallet_Entry	Detecta palet al final del transportador de rodillos	Sensor fotoeléctrico difuso	Encendido/Apagado	×	%I2.6
PAL_Ele_+	Detecta el extremo final del palet dentro del elevador de paletizado	Sensor capacitivo	Encendido/Apagado	×	%I2.7
PAL_Ele_-	Detecta el extremo inicial del palet dentro del elevador de paletizado	Sensor capacitivo	Encendido/Apagado	×	%I3.0
PAL_Push	Detecta si el pistón que empuja las cajas está extendido o retraído	Sensor inductivo	Encendido/Apagado	×	%I3.1

PAL_Plate	Detecta si la compuerta está cerrada o abierta	Sensor inductivo	Encendido/Apagado	x	%I3.2
PAL_Ele_Moving	Detecta si el elevador de paletizado se está moviendo	Sensor de movimiento	Encendido/Apagado	x	%I3.3
PAL_Clamped	Detecta si los pistones que comprimen las cajas están extendidos	Sensor inductivo	Encendido/Apagado	x	%I3.4
ELE_Right_Limit	Detecta el extremo final del palet dentro del elevador	Sensor capacitivo	Encendido/Apagado	x	%I3.5
ELE_Pallet_Entry	Detecta si hay un palet a la entrada del elevador	Sensor fotoeléctrico difuso	Encendido/Apagado	x	%I3.6
ELE_Min	Detecta si el elevador está totalmente abajo	Sensor inductivo	Encendido/Apagado	x	%I3.7
ELE_MAX	Detecta si el elevador está totalmente arriba	Sensor inductivo	Encendido/Apagado	x	%I4.0
ELE_Left_Limit	Detecta el extremo inicial del palet dentro del elevador	Sensor capacitivo	Encendido/Apagado	x	%I4.1
ELE_STOP	Detecta cuando el elevador pasa por la mitad de su trayectoria y sirve para frenar el variador	Sensor inductivo	Encendido/Apagado	x	%I4.2
Tank_Max	Detecta el nivel máximo de agua dentro del tanque	Sensor capacitivo	Encendido/Apagado	x	%I4.3
Tank_Min	Detecta el nivel mínimo de agua dentro del tanque	Sensor capacitivo	Encendido/Apagado	x	%I4.4
Tank_Flow	Mide el flujo de agua que sale del tanque	Flujometro	0-60	L/min	%IW30
Tank_Level	Mide el nivel de agua dentro del tanque	Sonda hidrostática	0-100	%	%IW32
RESET	Botón de reset	Pulsador	Encendido/Apagado	x	%I4.6

ELE_Pallet_Exit	Detecta cuando un palet sale del proceso de paletizado	Sensor fotoeléctrico difuso	Encendido/Apagado	×	%I4.5
Ele_Pallet_Exit_2	Detecta cuando un palet llega a la zona de entrega	Sensor fotoeléctrico difuso	Encendido/Apagado	×	%I4.7
PAL_ELE_Min	Detecta cuando el elevador de paletizado está totalmente abajo	Sensor inductivo	Encendido/Apagado	×	%I5.0
RESET_FACTORY	Botón de reset desde Factory I/O	Pulsador	Encendido/Apagado	×	%I5.1

Figura A1. Tabla de variables de entrada del PLC.

Tabla de variables de salida del PLC					
Nombre	Descripción	Tipo	Rango	Unidad	Dirección
RUN_LIGHT	Luz de marcha	Lámpara	Encendido/Apagado	×	%Q2.0
STOP_LIGHT	Luz de paro	Lámpara	Encendido/Apagado	×	%Q2.1
PAL_Coveyor_Pal	Transportador de rodillos para palets	Motor	Encendido/Apagado	×	%Q2.2
PAL_Belt_Conveyor	Cinta transportadora de cajas	Motor	Encendido/Apagado	×	%Q2.3
PAL_Belt_+	Cinta transportadora de cajas hacia el interior del paletizado	Motor	Encendido/Apagado	×	%Q2.4
PAL_Belt_-	Cinta transportadora de cajas hacia el exterior del paletizado	Motor	Encendido/Apagado	×	%Q2.5
PAL_Chain_+	Cadena transportadora de palet hacia el interior del elevador de paletizado	Motor	Encendido/Apagado	×	%Q2.6
PAL_Chain_-	Cadena transportadora de palet hacia el exterior del elevador de paletizado	Motor	Encendido/Apagado	×	%Q2.7
PAL_Clamp	Pistones que comprimen las cajas	Pistón	Encendido/Apagado	×	%Q3.0
PAL_Ele_Down	Mueve el elevador de paletizado un paso hacia abajo	Motor	Encendido/Apagado	×	%Q3.1
PAL_Ele_Limit	Mueve el elevador de paletizado hasta su límite superior o inferior	Motor	Encendido/Apagado	×	%Q3.2
PAL_Ele_Up	Mueve el elevador de paletizado un paso hacia arriba	Motor	Encendido/Apagado	×	%Q3.3

PAL_Open_Plate	Abre o cierra la compuerta de paletizado	Pistón	Encendido/Apagado	×	%Q3.4
PAL_Push(1)	Pistón que empuja las cajas hacia la compuerta	Pistón	Encendido/Apagado	×	%Q3.5
PAL_Turn	Golpea las cajas para girarlas	Brazo metálico	Encendido/Apagado	×	%Q3.6
ELE_Conveyor_Down	Transportador de rodillos para palets hacia el elevador	Motor	Encendido/Apagado	×	%Q3.7
ELE_Conveyor_Up	Transportador de rodillos para palets hacia la zona de entrega	Motor	Encendido/Apagado	×	%Q4.0
ELE_Chain_+	Cadena transportadora de palet hacia el interior del elevador	Motor	Encendido/Apagado	×	%Q4.1
ELE_Chain_-	Cadena transportadora de palet hacia el exterior del elevador	Motor	Encendido/Apagado	×	%Q4.2
ELE_Down	Mueve el elevador hacia abajo	Motor	Encendido/Apagado	×	%Q4.3
ELE_Slow	Mueve el elevador lentamente hacia arriba o abajo	Motor	Encendido/Apagado	×	%Q4.4
ELE_Up	Mueve el elevador hacia arriba	Motor	Encendido/Apagado	×	%Q4.5
Tank_Discharge	Permite el paso del agua para vaciar el tanque	Válvula	0-10	V	%QW30
Tank_Fill	Permite el paso del agua para llenar el tanque	Válvula	0-10	V	%QW32
Emergencia	Faro amarillo de advertencia	Lámpara	Encendido/Apagado	×	%Q4.6

Figura A2. Tabla de variables de salida del PLC.

ANEXO 2

Logic Command			
Address (Decimal)	Bit(s)	Description	
8192	0	1 = Stop, 0 = Not Stop	
	1	1 = Start, 0 = Not Start	
	2	1 = Jog, 0 = No Jog	
	3	1 = Clear Faults, 0 = Not Clear Faults	
	5,4	00	No Command
		01	Forward Command
		10	Reverse Command
		11	No Command
	6	Not Used	
	7	Not Used	
	9,8	00	No Command
		01	Accel Rate 1 Enable
		10	Accel Rate 2 Enable
		11	Hold Accel Rate Selected
	11,10	00	No Command
01		Decel Rate 1 Enable	
10		Decel Rate 2 Enable	
11		Hold Decel Rate Selected	
14,13,12	000	No Command	
	001	Freq. Source = P036 [Start Source]	
	010	Freq. Source = A069 [Internal Freq]	
	011	Freq. Source = Comms (Addr 8193)	
	100	A070 [Preset Freq 0]	
	101	A071 [Preset Freq 1]	
	110	A072 [Preset Freq 2]	
	111	A073 [Preset Freq 3]	
15	Not Used		

Figura A3. Tabla de comandos de escritura en el registro 8192 de PowerFlex4 vía Modbus.(?)

Reference	
Address (Decimal)	Description
8193	A decimal value entered as xxx.x where the decimal point is fixed. For example, a decimal "100" equals 10.0 Hz and "543" equals 54.3 Hz.

Figura A4. Tabla de comando de escritura en el registro 8193 de PowerFlex4 vía Modbus.(?)

Logic Status		
Address (Decimal)	Bit(s)	Description
8448	0	1 = Ready, 0 = Not Ready
	1	1 = Active (Running), 0 = Not Active
	2	1 = Cmd Forward, 0 = Cmd Reverse
	3	1 = Rotating Forward, 0 = Rotating Reverse
	4	1 = Accelerating, 0 = Not Accelerating
	5	1 = Decelerating, 0 = Not Decelerating
	6	1 = Alarm, 0 = No Alarm
	7	1 = Faulted, 0 = Not Faulted
	8	1 = At Reference, 0 = Not At Reference
	9	1 = Reference Controlled by Comm
	10	1 = Operation Cmd Controlled by Comm
	11	1 = Parameters have been locked
	12	Digital Input 1 Status
	13	Digital Input 2 Status
	14	Not Used
15	Not Used	

Figura A5. Tabla de comandos de lectura en el registro 8448 de PowerFlex4 vía Modbus.(?)

Feedback ⁽¹⁾	
Address (Decimal)	Description
8451	A xxx.x decimal value where the decimal point is fixed. For example, a decimal "123" equals 12.3 Hz and "300" equals 30.0 Hz.

Figura A6. Tabla de comando de lectura en el registro 8451 de PowerFlex4 vía Modbus.(?)

Logic Status		
Address (Decimal)	Value (Decimal)	Description
8449	0	No Fault
	2	Auxiliary Input
	3	Power Loss
	4	Undervoltage
	5	Overvoltage
	6	Motor Stalled
	7	Motor Overload
	8	Heatsink Overtemperature
	12	HW Overcurrent (300%)
	13	Ground Fault
	29	Analog Input Loss
	33	Auto Restart Tries
	38	Phase U to Ground Short
	39	Phase V to Ground Short
	40	Phase W to Ground Short
	41	Phase UV Short
	42	Phase UW Short
	43	Phase VW Short
	63	Software Overcurrent
	64	Drive Overload
	70	Power Unit Fail
	80	AutoTune Fail
81	Communication Loss	
100	Parameter Checksum Error	
122	I/O Board Fail	

Figura A7. Tabla de comandos de lectura en el registro 8449 de PowerFlex4 vía Modbus.(?)

ANEXO 3



Figura A8. Dashboard en Grafana cuando la eficiencia del proceso está en una etapa transitoria.

PROCESO INDUSTRIAL 1/1



Figura A9. Dashboard en Grafana durante un paro de emergencia del sistema.

PROCESO INDSTRIAL 1/1

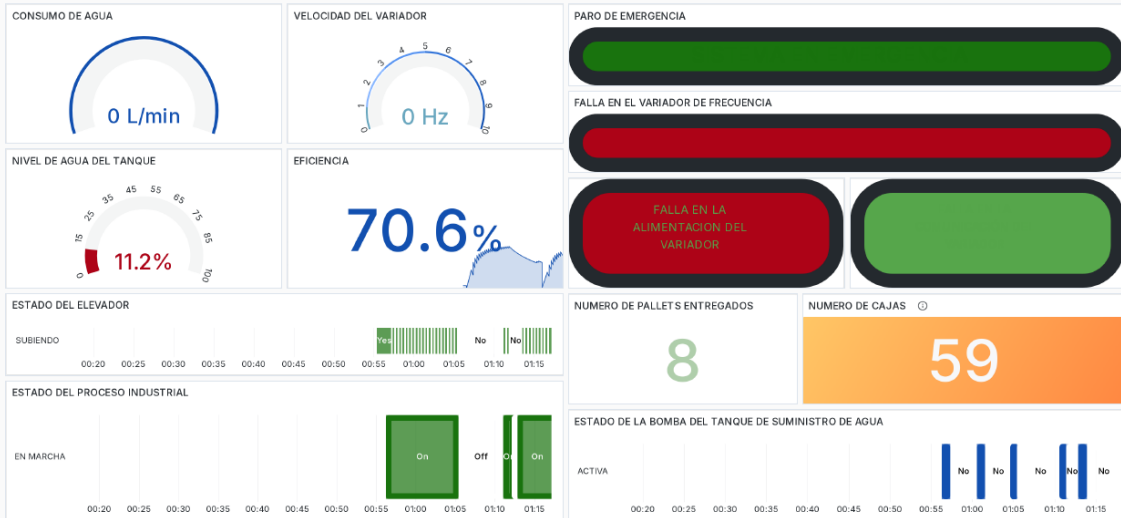


Figura A10. Dashboard en Grafana durante una falla en el variador de frecuencia.

PROCESO INDUSTRIAL 1/1

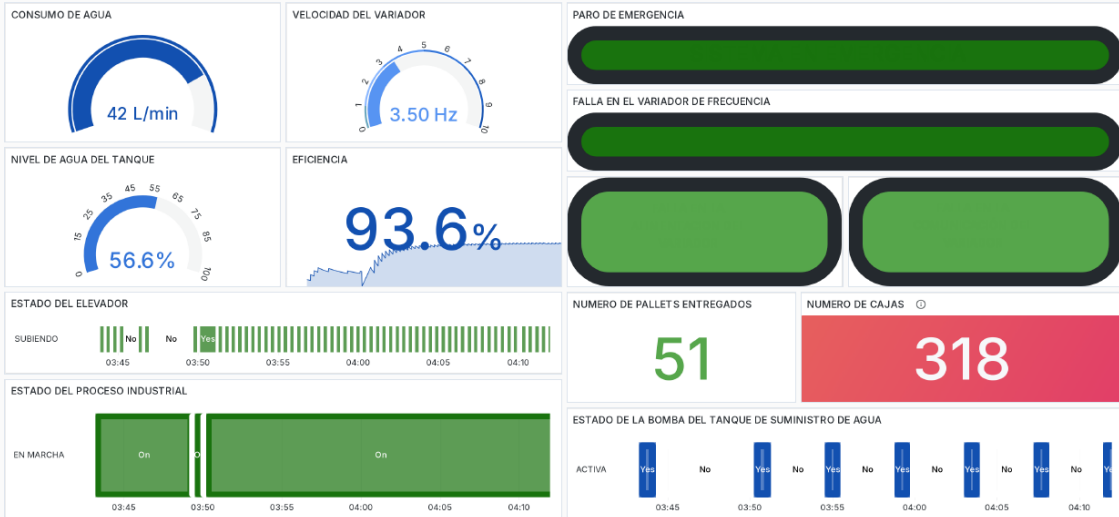


Figura A11. Dashboard en Grafana cuando la eficiencia del proceso está en una etapa estacionaria.