ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DISEÑO E IMPLEMENTACIÓN DE DOS MÓDULOS DIDÁCTICOS QUE EMULEN UNA CELDA DE CONTROL INDUSTRIAL BASADA EN UNA RED PROFINET Y UNA PLATAFORMA DE CONTROL SUPERVISORIO

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE MAGISTER EN ELECTRÓNICA Y AUTOMATIZACIÓN, MENCIÓN REDES INDUSTRIALES

IVÁN ALEJANDRO DÁVILA ACOSTA

DIRECTOR: Dra. Ing. SILVANA GAMBOA BENÍTEZ
CODIRECTOR: Dr. Ing. LUIS ALBERTO MORALES ESCOBAR

Quito, agosto 2023

AVAL

Certifico que el presente trabajo fue desarrollado por Iván Alejandro Dávila Acosta, bajo nuestra supervisión.

Dra. Ing. SILVANA GAMBOA BENÍTEZ
DIRECTOR DEL TRABAJO DE TITULACIÓN

Dr. LUIS ALBERTO MORALES ESCOBAR
CODIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Iván Alejandro Dávila Acosta, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

Iván Alejandro Dávila Acosta

DEDICATORIA

El presente Trabajo de Titulación va dedicado a mi familia, quienes han sido mi fuente de fortaleza e impulso para enfrentar nuevos desafíos.

AGRADECIMIENTO

Quiero expresar mis más sinceros agradecimientos a todas las personas que contribuyeron a hacer posible este importante logro.

En primer lugar, quiero agradecer a mis padres, Iván y Sandra, por su amor incondicional y la confianza que depositaron en mí para alcanzar nuevos objetivos en mi vida. Han sido el pilar fundamental que me ha permitido crecer tanto personal como académicamente.
A mi querido hermano Emilio, gracias por estar a mi lado en cada paso de este camino y por ser mi cómplice y compañero en cada momento. Tu apoyo incondicional ha significado mucho para mí.

A mi sobrino Juan Martín, quien ha llenado de alegría a toda la familia con su amor y ternura.

A la Escuela Politécnica Nacional, por brindarme la oportunidad de crecer durante esta etapa de mi vida. Especialmente a la Dra. Silvana Gamboa, cuya dedicación y disposición han sido clave en la culminación exitosa de mi programa.

Iván

ÍNDICE DE CONTENIDO

AVAL I
DECLARACIÓN DE AUTORÍA II
DEDICATORIA III
AGRADECIMIENTO IV
ÍNDICE DE CONTENIDO V
RESUMEN VII
ABSTRACT VIII
1. INTRODUCCIÓN 1
1.1. Objetivo General 1
1.2. Objetivos Específicos 1
1.3. Alcance 1
1.4. Marco Teórico 3
1.4.1. Arquitectura CIM 3
1.4.1. Sistema SCADA 5
1.4.2. Redes de Campo 7
1.4.3. Profinet 8
1.4.4. Comunicación Profinet 8
1.4.5. Plataforma de Control Supervisorio 13
1.4.6. Seguridad en los ICS 17
1.4.7. HMI 19
2. METODOLOGÍA 31
2.1. Requerimientos ICS 31
2.2. Requerimientos Celda de Control 33
2.2.1. Características de los Componentes de Automatización 33
2.2.2. Instrucciones para Comunicación Profinet 37
2.2.3. Instrucción LED 39
2.3. Implementación Celda de Control 40
2.3.1. Configuración de la Red 40
2.3.2. Implementación de la Conexión TCP 41
2.4. Requerimientos Plataforma de Control Supervisorio 52
2.4.1. Arquitectura de los Componentes de la Plataforma 52
2.4.2. Aplicación de Control Supervisorio 53
2.4.3. Servidores de Datos 54
2.4.4. Software de Visualización de HMI 54

2.4.5. Planificación de la Aplicación 55
2.4.6. Sistema de Seguridad 56
2.4.7. Software de Historización 60
2.5. Implementación Plataforma de Control Supervisorio 61
2.5.1. Creación y Planificación del Servidor de Aplicación 61
2.5.2. Configuración Plantilla Objetos de Aplicación 64
2.5.3. Estaciones de la Plataforma 67
2.5.4. Estación de Ingeniería 68
2.5.5. Servidor de Datos 70
2.5.6. Implementación HMI 74
2.5.7. Alarmas 82
2.5.8. Histórico 88
2.5.9. Seguridad 95
3. RESULTADOS Y DISCUSIÓN 98
3.1. Resultados 98
3.1.1. Descripción de la Planta 98
3.1.2. Lógica Subproceso 1 101
3.1.3. Lógica Subproceso 2 102
3.1.4. Verificación de Lectura y Escritura de Variables 103
3.1.5. Verificación Historiador 120
4. CONCLUSIONES Y RECOMENDACIONES 121
4.1. CONCLUSIONES 121
4.2. RECOMENDACIONES 122
5. REFERENCIAS BIBLIOGRÁFICAS 124
6. ANEXOS 1
ANEXO A 2
ANEXO B 4
ANEXO C 10

RESUMEN

El presente trabajo de titulación propone el diseño e implementación de dos módulos didácticos. El primer módulo simula una celda de control industrial utilizando tres controladores lógicos programables. Dos de los controladores se configuran como unidades terminales remotas, mientras que el tercero actúa como una unidad terminal maestra. Estos controladores están integrados en una red de comunicaciones basada en el protocolo Profinet.

El segundo módulo emula una plataforma de control supervisorio que forma parte del sistema de control industrial. Este módulo se encarga de las tareas de adquisición y monitoreo de datos del primer módulo. El software asociado a la aplicación de control supervisorio de esta plataforma fue desarrollado utilizando la herramienta de ingeniería *ArchestrA* y consta de cuatro estaciones remotas para las tareas de ingeniería, operación, servidor de datos e historiador de proceso.

Para la simulación se utilizó una interfaz humano-máquina representada en la estación de operación, la cual muestra un proceso de producción de chocolate.

Estos módulos didácticos serán utilizados en el Laboratorio de Redes Industriales.

PALABRAS CLAVE: Profinet, ICS, ArchestrA, Plataforma de Control Supervisorio, Modelo CIM, Siemens, Wonderware

ABSTRACT

This report proposes the design and implementation of two didactic modules. The first module simulates an industrial control cell using three programmable logic controllers. Two of the controllers are configured as remote terminal units, while the third one acts as a master terminal unit. These controllers are integrated into a communication network based on the Profinet protocol.

The second module emulates a supervisory control platform that is part of the industrial control system. This module is responsible for data acquisition and monitoring tasks of the first module. The software associated with the supervisory control application of this platform was developed using the ArchestrA engineering tool and consists of four remote stations for engineering, operation, data server, and process historian tasks.

For the simulation, a human-machine interface represented by the operation station was used, which displays a chocolate production process.

These didactic modules will be used in the Industrial Networks Laboratory.

KEYWORDS: Profinet, ICS, ArchestrA, Supervisory Control Platform, CIM Model, Siemens, Wonderware

1. INTRODUCCIÓN

1.1. Objetivo General

Diseñar e implementar dos módulos didácticos que emulen una celda de control industrial basada en una red Profinet y una plataforma de control supervisorio.

1.2. Objetivos Específicos

- Realizar la búsqueda bibliográfica sobre redes de comunicación Profinet y plataforma de supervisión en sistemas de control industrial (ICS).
- Establecer los requerimientos del módulo de red de comunicación Profinet y el módulo de control supervisorio industrial.
- Establecer el software y el hardware con los cuales se desarrollarán los módulos propuestos para el ICS a implementarse.
- Diseñar e implementar los módulos del ICS acorde a los requerimientos establecidos durante la etapa de diseño.
- Realizar la validación del desempeño de los módulos desarrollados acorde a los requerimientos establecidos durante la etapa de diseño.

1.3. Alcance

En el presente trabajo de titulación se diseñará e implementará dos celdas que representarán dos secciones de un sistema de control industrial (ICS), que funcionarán como módulos didácticos para el Laboratorio de Redes Industriales. El primer módulo emulará una celda de control industrial cuya comunicación se basará en una red Profinet. El segundo módulo emulará una plataforma de control supervisorio industrial, para la adquisición y monitoreo del primer módulo.

Con este objetivo, se realizará una búsqueda bibliográfica sobre los ICS y sus componentes de la red de comunicación, dando énfasis a las redes Profinet, además de sus interfaces de operador, servidores de datos, plataforma de

supervisión y equipos de adquisición de datos a nivel de aplicación industrial, así como de normativas y/o estándares que rijan su diseño.

Luego, a partir de la búsqueda bibliográfica, se establecerán los requerimientos que deben cumplir las dos celdas a implementarse. Así también, se establecerá las herramientas de software y los elementos complementarios de hardware que se requerirán para que los módulos propuestos se ajusten a los requerimientos previamente establecidos, ajustados a utilizar el protocolo Profinet y de la misma forma utilizar herramientas para el diseño de la plataforma.

Por consiguiente, se diseñarán e implementarán las dos celdas que deberán funcionar acorde a lo especificado a continuación:

- El primer módulo por implementarse emulará una celda de control industrial, este módulo dispondrá de tres controladores lógicos programables (PLCs) modelo S7-1200, dos de ellos se configurarán como unidades terminales remotas (RTU) y el tercero como una unidad terminal maestra (MTU). Estos controladores realizarán la adquisición, monitoreo y control del proceso industrial que se vincule a esta celda. Los componentes de la celda de control estarán integrados en una red de comunicaciones utilizando el protocolo industrial Profinet.

- El segundo módulo didáctico que se implementará realizará la emulación de una plataforma de control supervisorio industrial, para la adquisición y monitoreo del primer módulo previamente mencionado. En esta sección se tomarán los datos que provienen de la unidad terminal maestra (MTU). Además, se realizará el diseño y la implementación de una aplicación de control supervisorio que constará de una red de al menos 3 computadores que contemplen al menos las siguientes funcionalidades, estaciones de operación, una estación de ingeniería, un servidor de datos y un historiador de proceso.

- Las estaciones de operación constarán de una interfaz humano-máquina (HMI) para la operación del proceso industrial por parte del operador. La estación de ingeniería permitirá la reconfiguración de los componentes del sistema de control industrial. Se utilizará un servidor de datos para acceder a la información proporcionada por la unidad MTU. Además, se implementará un historiador de proceso, el cual permitirá el almacenamiento de un histórico.

Finalmente, se validarán los módulos implementados emulando un proceso industrial usando los instrumentos disponibles en el Laboratorio de Redes Industriales. Dicha emulación estará asociada a la producción de chocolate.

1.4. Marco Teórico

En el presente capítulo se detalla la información teórica utilizada para el desarrollo del presente trabajo de titulación. Se divide en diferentes secciones las cuales están relacionadas con las propuestas para la implementación de un sistema de control industrial, una plataforma de supervisión y su arquitectura.

1.4.1. Arquitectura CIM

Para abordar un conjunto de componentes encargados de manejar una planta industrial automatizada, se plantea observar un modelo de referencia para la clasificación de estas estaciones. Uno de los más utilizados es el modelo *Computer Integrated Manufacturing* (CIM), traducido como Manufactura Integrada por Computadora. Este modelo comprende un conjunto de aplicaciones informáticas que tienen como objetivo automatizar diversos procesos, actividades y tareas en la industria, desde el ingreso de la materia prima, hasta la entrega del producto final. En la arquitectura CIM, los sistemas integran diferentes áreas funcionales a través de flujos de información y materiales, respaldados en plataformas de hardware y software.

El modelo CIM se basa en una estructura jerárquica con el objetivo de mejorar la eficiencia de los componentes, buscando los siguientes objetivos:

- Incremento en la flexibilidad.
- Mejora de la confiabilidad del sistema
- Reducción de costos.

1.4.1.1. Niveles del Modelo CIM

Cada uno de estos componentes se sitúa en una pirámide correspondiente a su función y cuenta con cinco divisiones que interactúan con los niveles adyacentes, como se muestra en la Figura 1.1. [1].

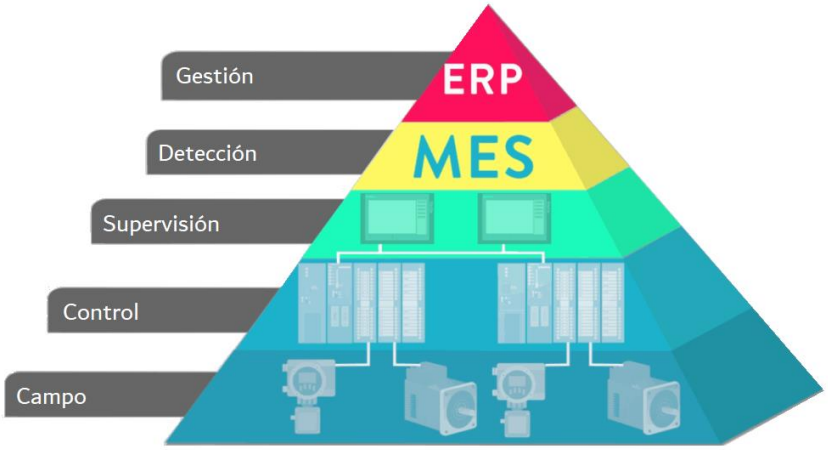


Figura 1.1. Pirámide CIM

Nivel de Campo

Este nivel corresponde a la base de la pirámide, donde se encuentran todos los equipos y componentes de hardware utilizados en los procesos de fabricación, y están en contacto directo con el proceso productivo. Los elementos característicos de este nivel incluyen sensores y actuadores.

Nivel de Control

El segundo nivel de la pirámide CIM es el nivel de control. Aquí se encuentran diversos dispositivos y controladores encargados del procesamiento de la información adquirida, realizando funciones de control y monitorización inicial. Estos dispositivos tienen la capacidad de gestionar los sensores y actuadores pertenecientes al nivel de campo. Tanto los componentes de este nivel como los

del nivel de campo tienen la capacidad de realizar tareas productivas de forma autónoma. Los dispositivos principales son los PLC, así como los controladores PID y los dispositivos electrónicos inteligentes (IEDs).

Nivel de Supervisión

En el tercer nivel se encuentran los sistemas utilizados para la supervisión de la información proveniente de la planta. Uno de los sistemas más comunes en este nivel es el sistema SCADA, el cual será detallado más adelante.

Nivel de Detección

El cuarto nivel, también conocido como nivel de detección, introduce sistemas de ejecución y planificación con el objetivo de reducir los costos de producción. Un ejemplo de sistema es el *Manufacturing Execution System* (MES).

Nivel de Gestión

En el quinto nivel se encuentran los sistemas de gestión integral relacionados con la planificación de recursos. El sistema ERP se destaca como el principal en este nivel.

1.4.1. Sistema SCADA

El sistema de Control con Supervisión y Adquisición de Datos (SCADA) se utiliza para hacer referencia a un conjunto de componentes que permiten el acceso a datos remotos y, al mismo tiempo, controlar el proceso. Vemos que se trata de un conjunto de utilidades diseñado para la monitorización y supervisión de una planta.

Los principales objetivos del sistema SCADA son la visualización sencilla de lo que ocurre en la instalación, el control de los parámetros de funcionamiento, la conectividad entre las diferentes estaciones, y la disponibilidad y accesibilidad inmediata de los datos.

El sistema está asociado a una aplicación de software con un funcionamiento optimizado en las estaciones de control, producción y comunicación, permitiendo el

control automático del proceso. Como principales prestaciones del sistema SCADA se destacan:

Monitorización

Consiste en la representación en tiempo real de las variables del proceso a los operadores. Estas variables pueden incluir datos de velocidad, temperatura, entre otros. Estos datos son utilizados para realizar un seguimiento y pueden ser preprocesadas.

Supervisión

Se refiere a las tareas de adquisición de datos y las herramientas de gestión que se utilizan para tomar decisiones óptimas en el proceso. La supervisión permite asegurar el correcto funcionamiento de la planta, así como realizar tareas de mantenimiento y prevención.

1.4.1.1. Arquitectura del sistema SCADA

Los usuarios del sistema SCADA utilizan herramientas de visualización y control para acceder al proceso, y generalmente emplean un ordenador como herramienta principal. Asimismo, los datos del proceso son adquiridos por los elementos de campo. La arquitectura del sistema SCADA se representa en la Figura 1.2 [2].

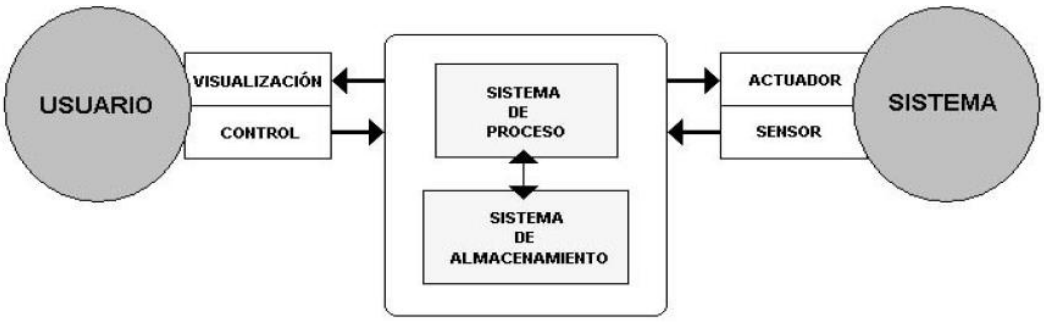


Figura 1.2. Arquitectura del sistema SCADA

Cada elemento del sistema implica la participación de varios subsistemas. Por ejemplo, en el proceso de adquisición de datos, es posible contar con uno o varios controladores que, a su vez, transportan los datos a través de diferentes servidores.

1.4.2. Redes de Campo

Los modelos y sistemas descritos anteriormente se basan en una comunicación jerárquica que se relaciona con sus funcionalidades. En este contexto, las redes de campo, también conocidas como buses de campo, son una tecnología utilizada en aplicaciones de comunicaciones en entornos industriales. Estas redes proporcionan un sistema de transmisión de los datos que facilita las instalaciones y las operaciones de las máquinas y equipos utilizados. Tienen como objetivo reemplazar las conexiones punto a punto entre los componentes de campo con los dispositivos de control [3].

Los buses de campo son generalmente redes digitales multipunto montadas en serie que se utilizan para gestionar procesos de control, mantenimiento o diagnóstico. En la Figura 1.3. se muestra una representación de la evolución de diversas redes de campo a lo largo de las últimas décadas.

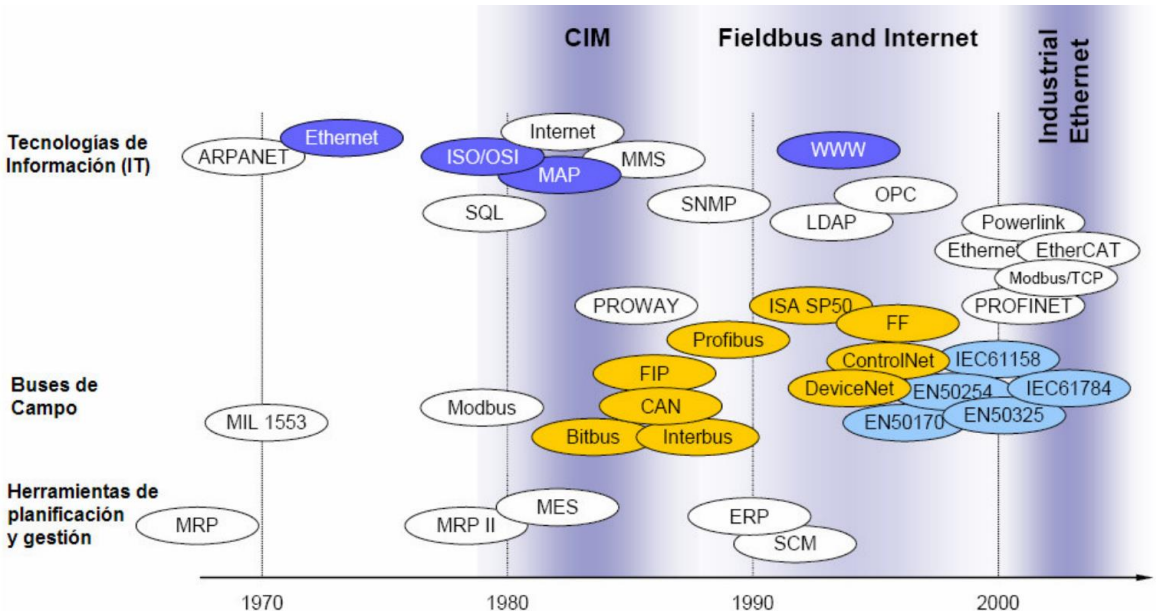


Figura 1.3. Evolución de tecnologías de comunicación y redes de campo.

1.4.3. Profinet

En la década de 1990, la implementación de tecnologías de buses de campo representó una innovación significativa al permitir la migración de sistemas de automatización centralizados a sistemas descentralizados. En este contexto, inicialmente se propuso el protocolo Profibus, el cual ha mantenido su posición como estándar líder en el mercado durante más de 20 años. Sin embargo, en la actualidad, las tecnologías de la información basadas en Ethernet, respaldadas por estándares como TCP/IP, han experimentado un considerable avance. Estas tecnologías se han incorporado de manera efectiva en el ámbito de la automatización industrial, lo que ha permitido una comunicación más eficiente entre sistemas, amplias configuraciones y mayores posibilidades de diagnóstico. Como resultado, surgió la necesidad de una transición hacia un protocolo más avanzado, conduciendo a la adopción de Profinet como el protocolo sucesor [4].

Profinet es un estándar abierto para Ethernet Industrial que cumple con todos los requerimientos de las tecnologías de automatización. Según los estándares de IEEE, Profinet es completamente compatible con Ethernet y está estandarizado en IEC 61158 y IEC 61784 [5].

La arquitectura modular de Profinet permite al usuario elegir la funcionalidad necesaria y define los siguientes requisitos para la comunicación de datos:

- Comunicación a 100 Mbps a través de cableado de cobre o fibra óptica (100 Base TX o 100 Base FX).
- Transmisión Full Dúplex

1.4.4. Comunicación Profinet

Para el intercambio de datos utilizando este protocolo, se utiliza tres canales de comunicación, como indica en la Figura 1.4. [6]:

- TCP/IP (o UDP/IP)
- Tiempo Real o *Real Time* (RT)
- Tiempo Real Isócrono o *Isochronous Real-Time* (IRT)

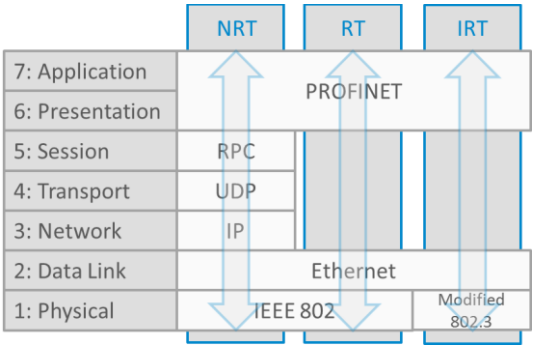


Figura 1.4. Canales Profinet

1.4.4.1. TCP/IP (o UDP/IP)

Este canal se utiliza para transportar datos que no requieren un tiempo crítico, como tareas de configuración, diagnóstico o parametrización. A través de este canal, se pueden incorporar protocolos adicionales, como son TCP/IP, utilizando funciones de transporte.

TCP/IP

Los protocolos principales que definen el funcionamiento de este conjunto son en el Protocolo de Internet (IP), el cual predomina en la capa de red del modelo OSI. Este protocolo proporciona funciones de direccionamiento, enrutamiento de datagramas y otras funciones de red. Por otro lado, otro protocolo importante es el Protocolo de Control de Transmisión (TCP), el cual es ampliamente utilizado en la cuarta capa del modelo OSI (capa de transporte) y se encarga de establecer, administrar la conexión y garantizar el transporte de datos entre los dispositivos [7].

Estos paquetes, al no ser críticos, contienen más información (datagrama más extenso), debido a que atraviesan una mayor cantidad de capas, como se observa en la Figura 1.5.

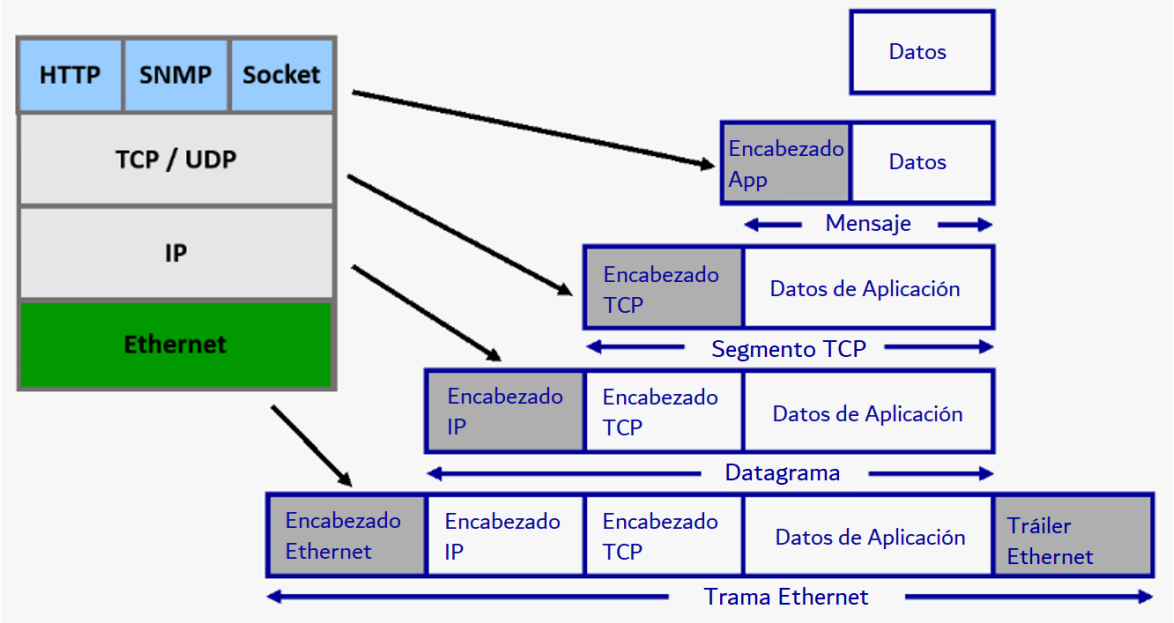


Figura 1.5. Trama Ethernet.

Esta información adicional resulta en telegramas Ethernet de mayor tamaño de datos en comparación con otros. El proceso de empaquetar y desempaquetar los datos se conoce como encapsulación.

Aunque los protocolos TCP/IP ofrecen ventajas en cuanto a funcionalidad, también presentan desventajas, como la fluctuación de retardo o *jitter*, que se describe como la variación en la latencia de los paquetes. Por estas razones, este canal no es el adecuado para tareas que requieran altas velocidades debido a su naturaleza crítica.

1.4.4.2. Tiempo Real (RT)

Este canal de comunicaciones se encarga del intercambio de datos críticos en tiempo real. Cuando se identifica que una trama Ethernet es de tiempo crítico, se dirige de forma directa desde la capa 2 hacia la capa 7, evitando las funciones de los protocolos de capas intermedias y eliminando el tiempo variable de procesamiento. Como resultado, se mejora significativamente la velocidad de comunicación y el determinismo. Numerosas tramas Profinet son enviadas por este canal.

1.4.4.3. Tiempo Real Isócrono (IRT)

Incluso el envío de datos en tiempo real puede presentar fluctuaciones en la latencia. Por esta razón, Profinet implementa un canal adicional llamado Tiempo Real Isócrono (IRT), el cual elimina estos retrasos mejorando las reglas de tráfico y estableciendo reglas especiales. Estos datos están sincronizados en el tiempo y se envían en segmentos determinados.

1.4.4.4. Profinet IO

La variante Profinet IO permite una interfaz directa con dispositivos de campo distribuidos. Todos estos dispositivos se conectan en una estructura de red uniforme, lo que facilita la comunicación de manera consistente en toda la planta. Profinet IO se utiliza para el intercambio de datos, la parametrización y el diagnóstico entre los controladores y los dispositivos. Está diseñado para lograr un intercambio rápido de información con tiempos de ciclo de bus de solo unos pocos milisegundos.

Además de las topologías de estrella, árbol y anillo, Profinet IO maneja también una topología de línea característica llamada bus de campo. Esta topología se implementa con la funcionalidad de interruptor, la cual está integrada en sus dispositivos.

Esta variación se asemeja a Profibus DP, con la diferencia de que Profinet IO utiliza el método productor/consumidor en lugar del maestro/esclavo. A través de este intercambio, se asigna un tiempo a cada componente IO en el cual produce sus datos y los coloca en el bus con un identificador único, lo que permite un acceso simultáneo. Se muestra este intercambio en la Figura 1.6.

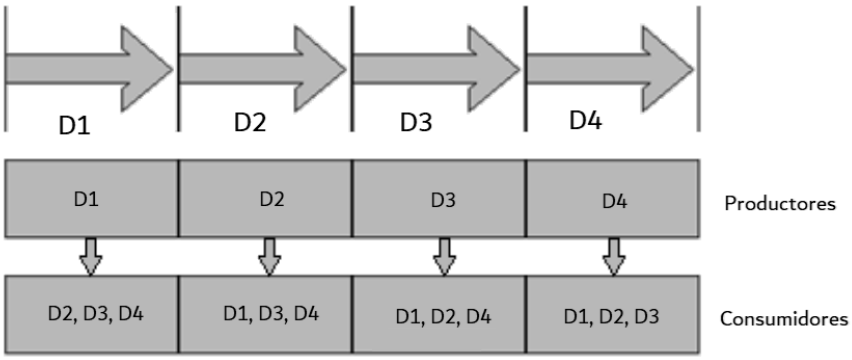


Figura 1.6. Intercambio Productor / Consumidor

Dentro de esta variante, existen diferentes clases de dispositivos, como se muestra en la Figura 1.7.

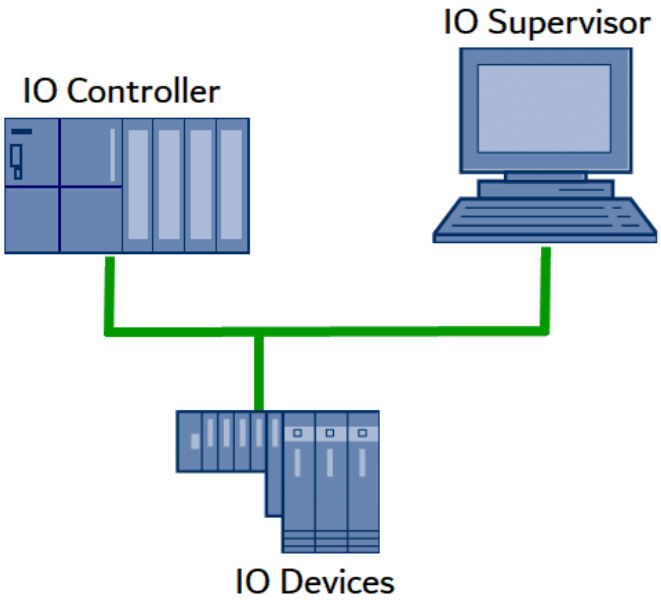


Figura 1.7. Clases de Dispositivos Profinet

A continuación, se detallan dichas clases:

Controlador IO

El *IO Controller* es un controlador programable, generalmente un PLC, en el cual se ejecuta la rutina de automatización. En todo sistema, siempre se cuenta con al menos uno de estos controladores IO.

Dispositivo IO

El *IO Device* es un dispositivo de campo distribuido que se encarga del intercambio de datos con uno o más controladores IO. En general, se cuenta con al menos un dispositivo IO en el sistema.

Supervisor IO

Es un dispositivo de ingeniería, como una PC, HMI o equipo de programación, que se utiliza para la puesta en servicio y diagnóstico de controladores IO y dispositivos IO.

Servidor de Parámetros IO

Es una estación de servidor, comúnmente una PC, la cual se encarga de cargar y guardar la configuración de los dispositivos IO.

1.4.5. Plataforma de Control Supervisorio

En las plantas de producción, la mejora continua es prioritaria. Para lograr este objetivo, es necesario utilizar un conjunto de elementos que incrementen la automatización en estos entornos de producción, aprovechando tecnologías que permitan la visualización de datos y la toma de decisiones de una forma más sencilla, agilizando así los procesos productivos. Existen herramientas integrales que nos permiten hacer frente a los constantes cambios y facilitar el desarrollo en ingeniería.

En el ámbito del control supervisorio, existen herramientas que proporcionan una plataforma única que abarca todo el sistema de control, incluyendo los componentes necesarios para la operación, configuración y procesamiento de información. La plataforma contiene aplicaciones que nos permiten llevar a cabo cada uno de estos procesos.

Dentro de la aplicación, se distinguen dos componentes: el programa de Desarrollo y el programa *Run-time*. El primer programa tiene herramientas y utilidades relacionadas con la creación y edición de cada componente de la plataforma,

mientras que el programa *run-time* se encarga de ejecutar la aplicación creada por el programa de desarrollo, manteniéndola en funcionamiento durante el desarrollo del proceso [2].

En la Figura 1.8. se observa una representación del funcionamiento interno de una plataforma de control supervisorio, en la cual los componentes están agrupados según su función.

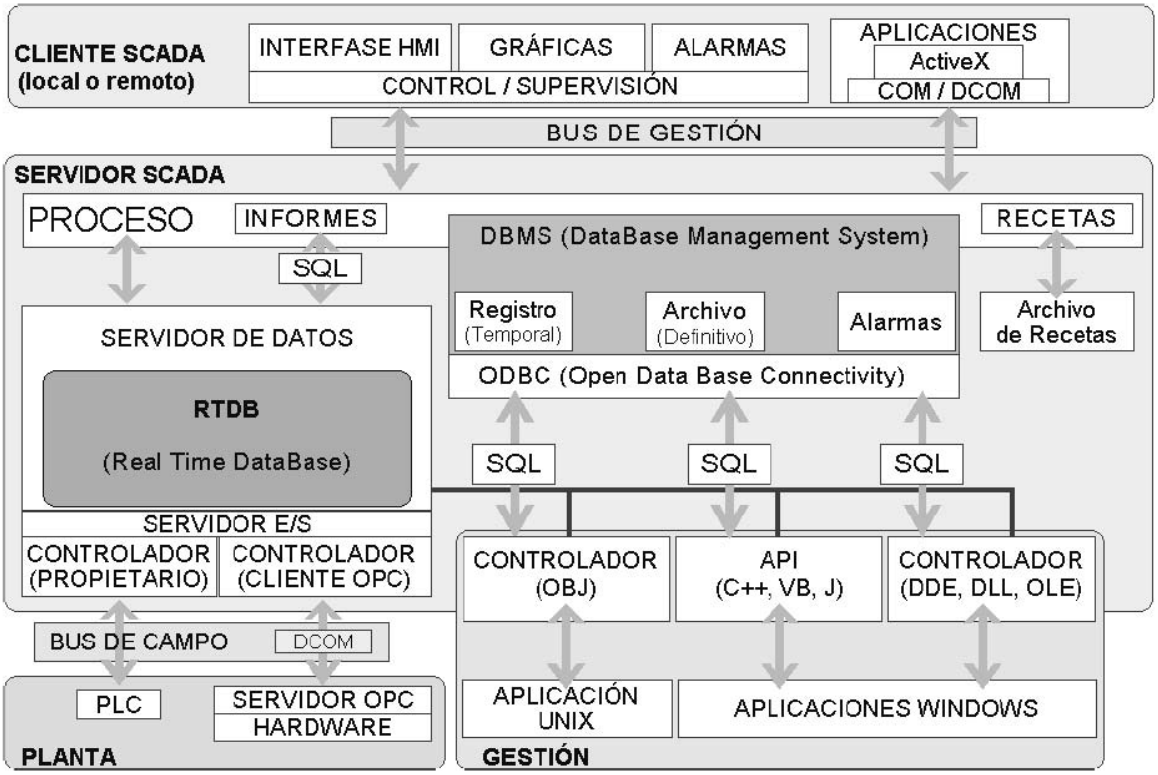


Figura 1.8. Arquitectura de la Plataforma de Control Supervisorio

Una característica importante en el uso de una plataforma de control supervisorio es contar con una arquitectura integral y escalable que sea fácil de implementar.

1.4.5.1. Enfoque Orientado a Objetos

El enfoque de desarrollo orientado a objetos se refiere a la capacidad de crear componentes basados en clases y derivarlos en una jerarquía. Este enfoque

permite construir un símbolo y replicarlo en objetos con funciones similares, mientras se mantienen sus características principales [8].

Los objetos utilizados permiten encapsular todas las configuraciones de cada componente del sistema, como la lógica de funcionamiento, la configuración del historiador, las alarmas y eventos, los atributos de entradas y salidas, el control de acceso y seguridad, así como los gráficos asociados, como se muestra en la Figura 1.9.

Figura 1.9. Objeto

Esta idea de autocontención reduce significativamente el tiempo de ingeniería relacionado con la creación, modificación y mantenimiento de cada uno de los componentes.

1.4.5.2. Plantillas e Instancias

Las herramientas de desarrollo de aplicaciones de control supervisorio relacionadas con objetos cuentan con una distribución de características que se propagan desde su forma base hasta cada elemento en particular, según su tipo [9].

Estas definiciones de alto nivel se denominan plantillas, mientras que las entidades que representan un elemento en específico, observadas en tiempo de ejecución (*run-time*), se llaman instancias, como se aprecia en la Figura 1.10.

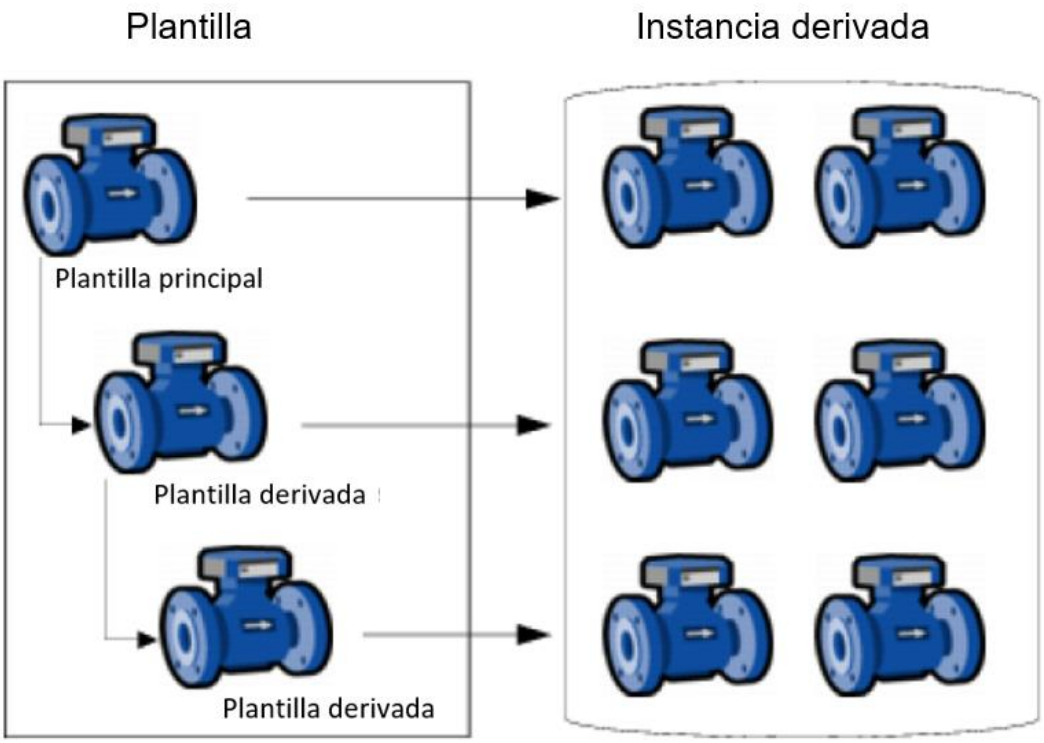


Figura 1.10. Plantillas e Instancias

- **Plantillas**

Una plantilla es una entidad que representa los requisitos funcionales comunes, como la información sobre las entradas y salidas, condiciones de alarma, configuraciones del historiador y seguridad. En este sentido, las configuraciones compartidas por las entidades derivadas (instancias hijas o plantillas derivadas) se definen aquí.

- **Instancias**

Las instancias son entidades creadas a partir de plantillas para funcionar en modo de ejecución. Son componentes individuales en el sistema y representan elementos únicos, como “PLC 01”, “HMI 02”, “Válvula 03”, entre otros.

1.4.5.3. Controlador de Comunicación

Un aspecto fundamental para el funcionamiento de una plataforma de control supervisiorio es la obtención de datos. Las interfaces HMI y las aplicaciones cliente que necesitan comunicarse con los dispositivos de campo requieren de programas conocidos como controladores o *drivers* de comunicación. Estos controladores tienen la función de traducir entre el lenguaje utilizado por la plataforma de control supervisorio y el PLC. Permite reconocer el dispositivo y utilizar una comunicación estandarizada para acceder y controlarlo. La configuración de los controladores se la realiza durante la elaboración del sistema [2].

1.4.6. Seguridad en los ICS

En comparación con otros sistemas de tecnología de la información (IT), las actualizaciones de los ICS han experimentado un crecimiento menos acelerado. Esto se debe a que realizar modificaciones en la infraestructura de los ICS implica cambios significativos con consecuencias importantes. Además, el aislamiento tradicional de las redes industriales ha contribuido a esta falta de actualización acelerada.

En las últimas décadas, la seguridad de las infraestructuras críticas relacionadas con los ICS se ha convertido en un tema recurrente debido a su conexión con las redes globales [10].

1.4.6.1. Autenticación

Uno de los métodos de seguridad internos utilizados en los ICS es la autenticación. La autenticación se define como el proceso de verificar la autenticidad de un usuario, un proceso o un dispositivo. Este proceso de verificación permite al usuario, proceso o dispositivo asegurarse de que es genuino. Durante el proceso de autenticación, el componente verificador se encarga de asegurarse de que la entidad examinada sea genuina. Además, el verificador participa activamente en el intercambio de información. Por lo general, el usuario utiliza información única

relacionada, como una contraseña o un certificado, para llevar a cabo este proceso de autenticación [11].

1.4.6.2. No-repudio

Otro término relacionado a la seguridad en los ICS es el repudio. Este se define como la negación de haber participado, total o parcialmente, en una acción por parte de una de las entidades involucradas. Por lo tanto, el no-repudio se refiere a la capacidad de protegerse contra la negación por parte de las entidades involucradas en la acción, es decir, asegurarse de que no puedan negar haber participado en la acción en un momento posterior [12]. Según la definición proporcionada por el Instituto Nacional de Estándares y Tecnología (NIST), el no-repudio también implica garantizar que el remitente tenga un comprobante de entrega y que el receptor tenga este comprobante para verificar la identidad del remitente. Esto asegura que ninguno de ellos pueda negar su participación en la acción en un momento posterior [13].

En resumen, el no-repudio es un método de seguridad que busca prevenir la negación de la participación en una acción, utilizando los comprobantes de entrega y verificación de la identidad.

1.4.6.3. Control de Acceso

Estos métodos de certificación se utilizan para gestionar el personal, asignando roles y funciones específicas a cada segmento del sistema. Es crucial contar con un sistema eficiente para agregar, eliminar o cambiar roles en las cuentas de usuario, evitando así procesos complejos y la creación de vulnerabilidades en los componentes de la red.

La autenticación se puede llevar a cabo a través de un proceso de control de acceso que verifica la identidad del usuario. Esto se puede implementar de dos formas: autenticación descentralizada y autenticación centralizada. En la autenticación descentralizada, cada componente de software realiza su propia verificación de autenticidad utilizando una base de datos independiente que almacena los datos

de los usuarios. Por otro lado, en la autenticación centralizada se utiliza un único punto de control de acceso y una base de datos centralizada para la autenticación. En el primer método, cada programa tiene su propia base de datos de usuarios, lo cual puede resultar poco flexible cuando se necesita escalar el número de cuentas de usuario. Por otro lado, la centralización de los controles de acceso agiliza y facilita estos inconvenientes, al utilizar una base de datos centralizada para la autenticación.

1.4.7. HMI

La interfaz humano-máquina, abreviada como HMI por sus siglas en inglés *Human-Machine Interface,* se define como un sistema que permite al usuario (ser humano) visualizar las variables relacionadas con un proceso determinado y controlarlo. Esta interfaz se encuentra presente en diversos dispositivos ubicados en proximidad al proceso, como paneles táctiles, y computadoras destinadas al operador [14].

Las funciones que ofrece incluyen la capacidad de monitorear y supervisar el proceso en tiempo real, la detección de eventos anómalos y mostrarlos como alarmas, aplicar algoritmos sencillos para realizar un preprocesamiento de los datos y la posibilidad de tomar muestras y almacenarlas en un histórico bajo ciertos criterios, estandarizados en la norma ANSI/ISA-101.01-2015 [15].

1.4.7.1. Componentes HMI

Existen varios componentes que forman parte del HMI, los cuales se detallan a continuación:

- **Consola:** Se trata de un conjunto de elementos compuestos de hardware, software y mobiliario utilizados por el usuario para llevar a cabo tareas de monitoreo y control del proceso. Estas consolas pueden incluir múltiples estaciones, dispositivos de comunicación y otros instrumentos, como estación de interruptores y cámaras. Como se observa en la Figura 1.11.

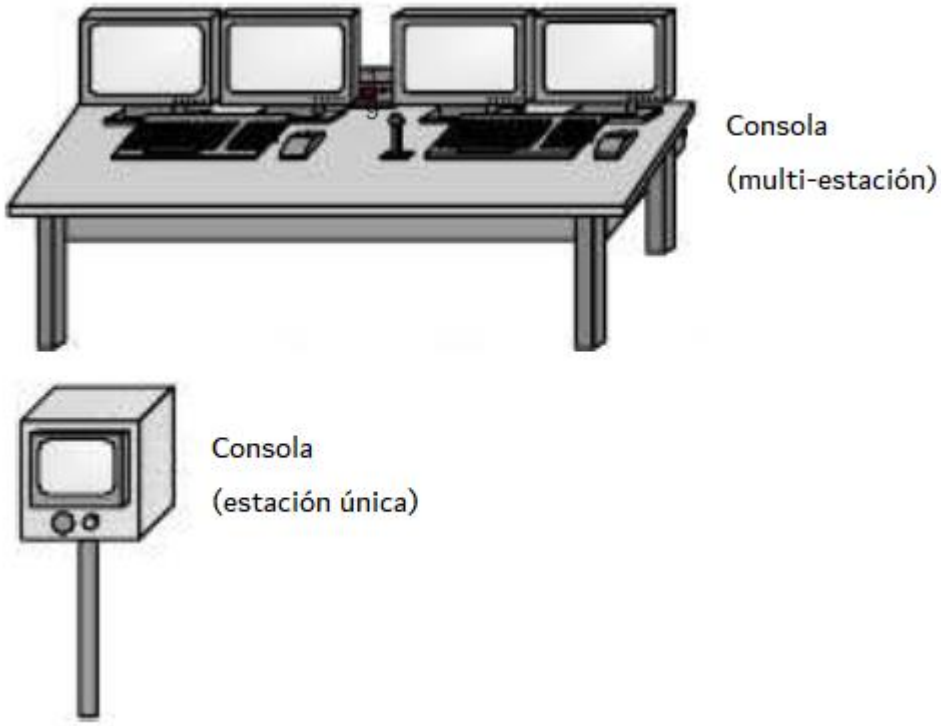


Figura 1.11. Consola

- **Estación (de operación):** Se refiere a la interfaz principal utilizada por el usuario, la cual consta de uno o más monitores y software de apoyo. Además, cuenta con periféricos de entrada, como teclado y dispositivo apuntador, así como periféricos de salida, como altavoces, impresoras. Esta configuración se puede observar en la Figura 1.12.

Figura 1.12. Estación

- **Monitor:** Es un dispositivo electrónico utilizado para visualizar información en forma de texto y/o gráficos, como una unidad de visualización de video (VDU), pantalla de computadora o pantalla LCD. Su función es mostrar información, tal como se muestra en la Figura 1.13.

  Además, el término “monitor” también puede utilizarse como verbo, refiriéndose a mantener la conciencia del estado del proceso mediante la observación de los datos y sus variaciones.

- **Pantalla:** Es la parte visible del monitor, como vista válida. Como se aprecia en la Figura 1.13.

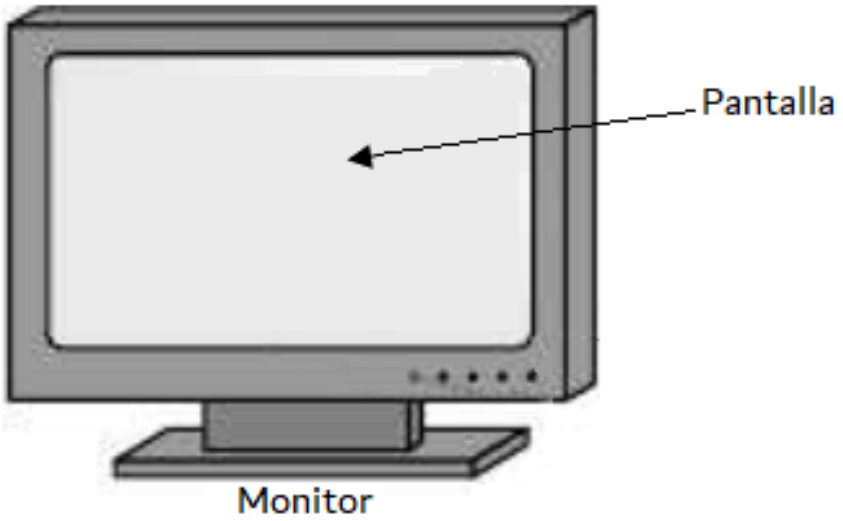


Figura 1.13. Monitor y Pantalla

- **Ventana:** Es una representación visual del proceso que se muestra en forma de área, conteniendo información relacionada. La ventana se visualiza en la pantalla, tal como se muestra en la Figura 1.14.

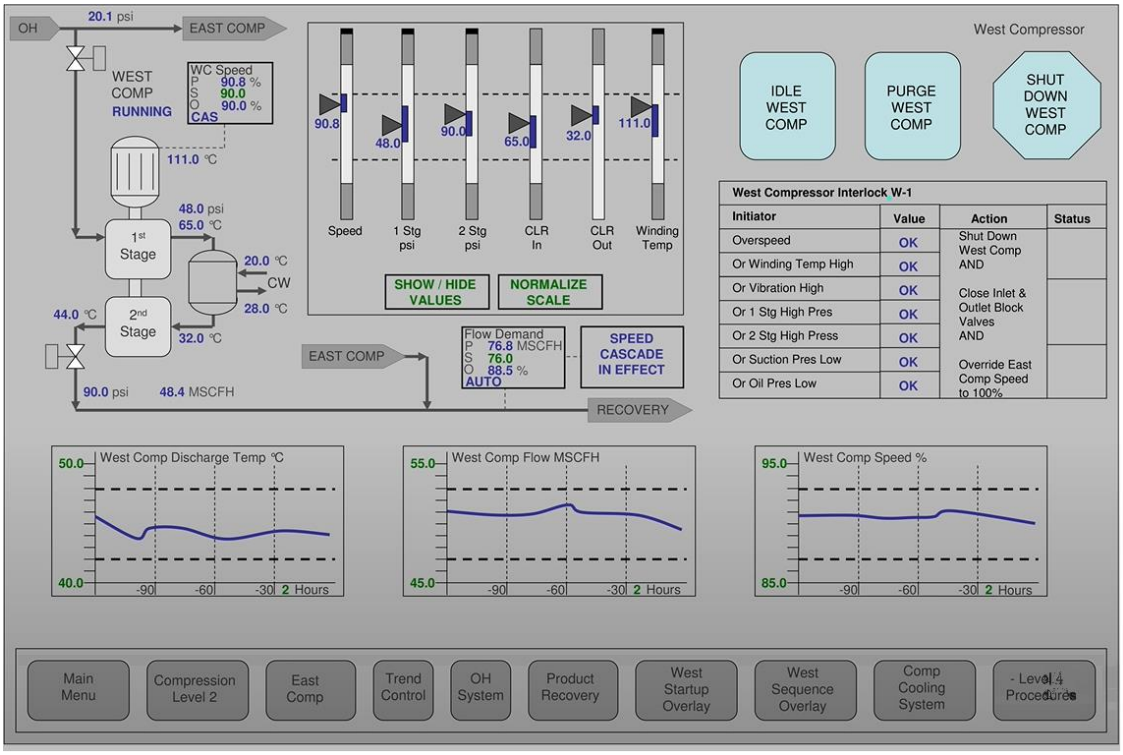


Figura 1.14. Ventana HMI

- **Ventana emergente (*Pop-up*):** Es un tipo de ventana que aparece en primer plano en la pantalla, pudiendo obstruir parcial o totalmente la visualización de otras ventanas. Se utiliza para mostrar alertas, elementos de suma importancia o el detalle de un proceso, como se aprecia en la Figura 1.15.

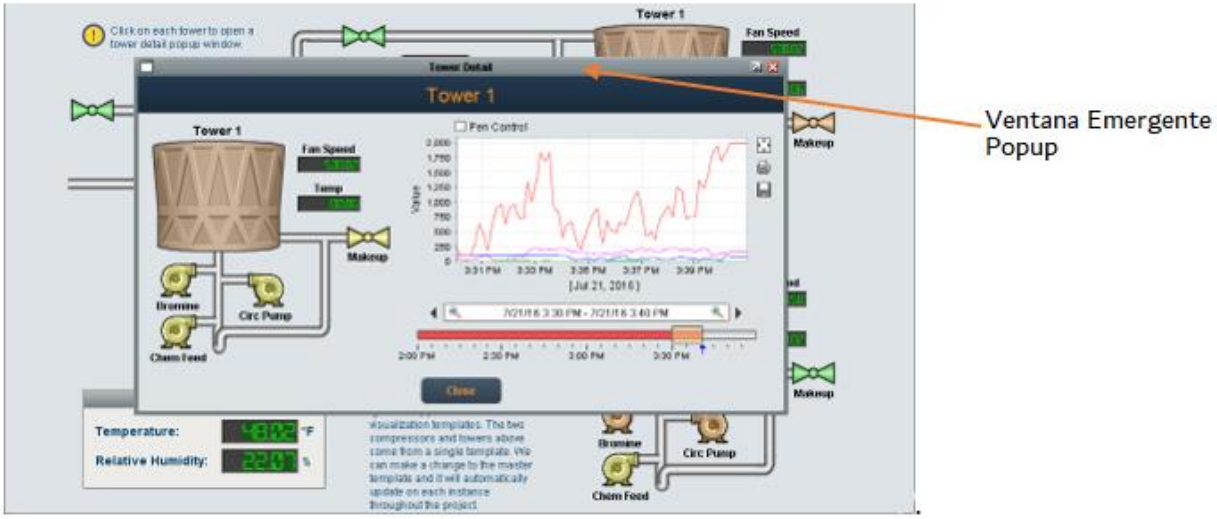


Figura 1.15. Ventana Emergente

- **Símbolos y Elementos Gráficos:** Se refiere a representaciones visuales de componentes o condiciones del proceso en la pantalla, y están compuestos por una combinación de elementos gráficos simples. Un símbolo es una representación completa, mientras que un elemento gráfico es una parte individual de este símbolo, como una figura geométrica, como se muestra en la Figura 1.16.

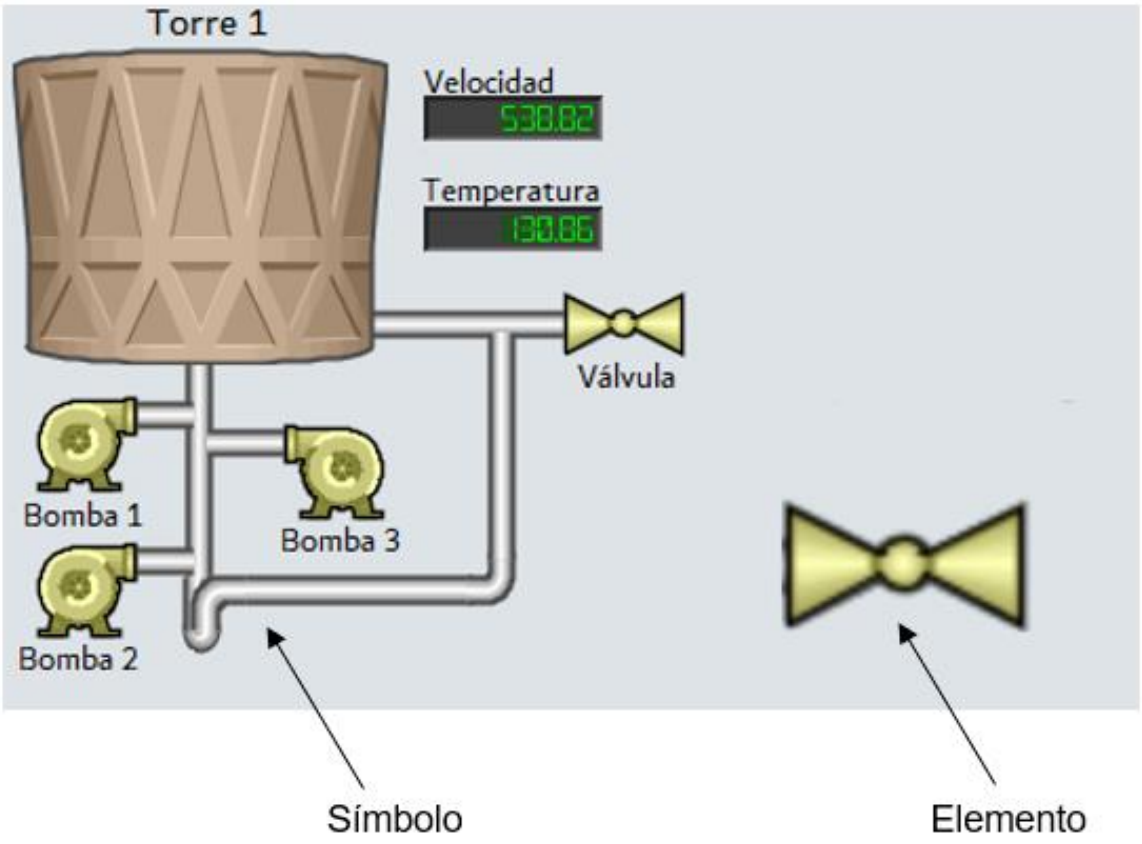


Figura 1.16. Símbolos y Elementos Gráficos

1.4.7.2. Jerarquía de Ventanas

Se recomienda establecer una jerarquía de ventanas que proporcione al operador una vista estructurada de sus responsabilidades y le permita acceder a diferentes niveles de control.

Nivel 1

Estas ventanas muestran una descripción general o un resumen de los parámetros clave, alarmas y condiciones generales del proceso, así como una visualización del espacio de navegación. En la Figura 1.17. se puede observar un ejemplo de este nivel.

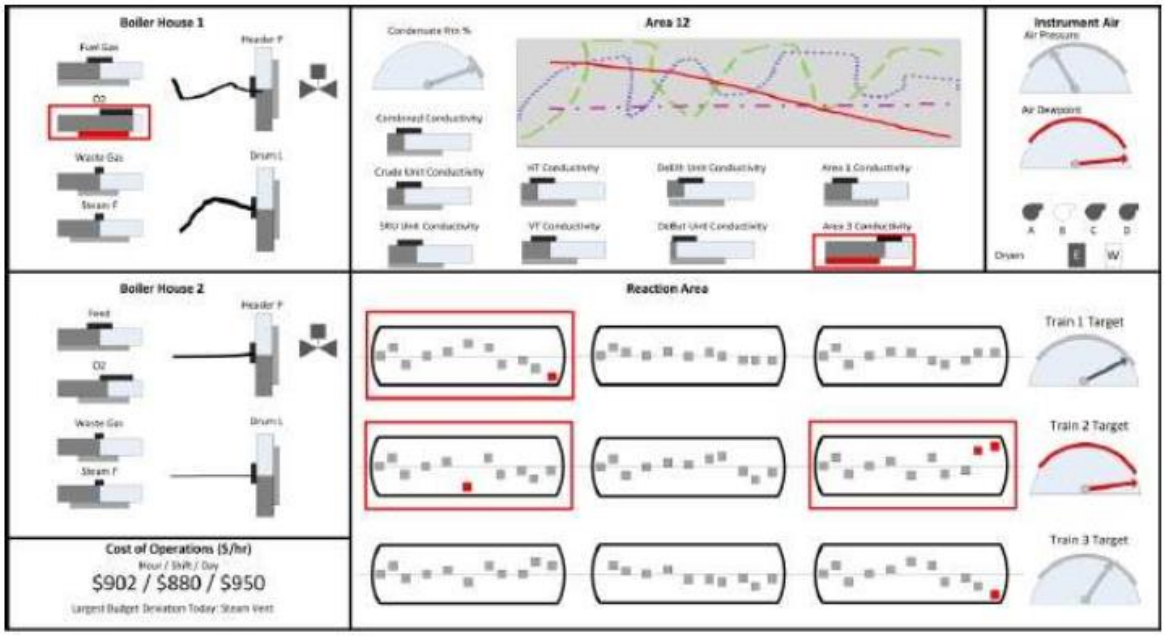


Figura 1.17. Ejemplo de una ventana Nivel 1

Nivel 2

También conocidas como ventanas de proceso con alto nivel, ofrecen más detalles que las ventanas de nivel 1. Estas ventanas son las más utilizadas durante las operaciones normales, para cambios en las rutinas y para el monitoreo continuo.

Nivel 3

Estas ventanas se utilizan con fines de diagnóstico y su objetivo no es el control del proceso.

Para la navegación entre las distintas pantallas, se recomienda un desplazamiento vertical entre niveles y horizontal entre ventanas del mismo nivel. En la Figura 1.18. se muestra un ejemplo de este nivel.

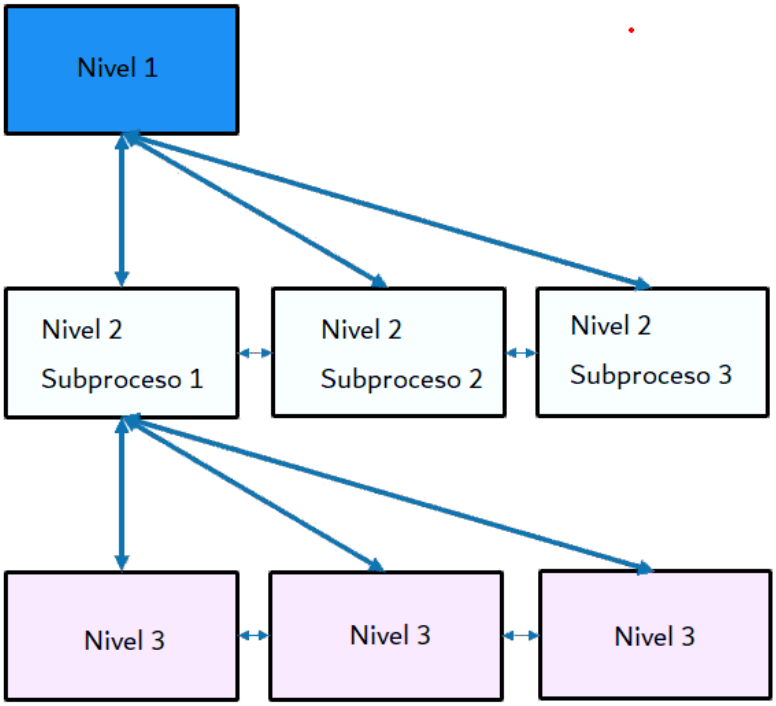


Figura 1.18. Navegación del HMI

1.4.7.3. Conciencia de la Situación

La conciencia situacional implica estar alerta y consciente de lo que está ocurriendo en el proceso, comprendiendo su estado presente y anticipando las posibles consecuencias futuras. Su objetivo es tomar acciones correctivas si es necesario. El HMI desempeña un papel fundamental al proporcionar esta conciencia y comprensión del estado del sistema. Cuando el proceso funciona correctamente, la interfaz debe generar una mínima cantidad de estímulos sensoriales. Sin embargo, cuando el proceso presenta un comportamiento anormal, el HMI debe proporcionar señales visuales y/o auditivas en función de la gravedad de la situación anómala [16]. Un diagrama descriptivo de la conciencia situacional es el mostrado en la Figura 1.19.

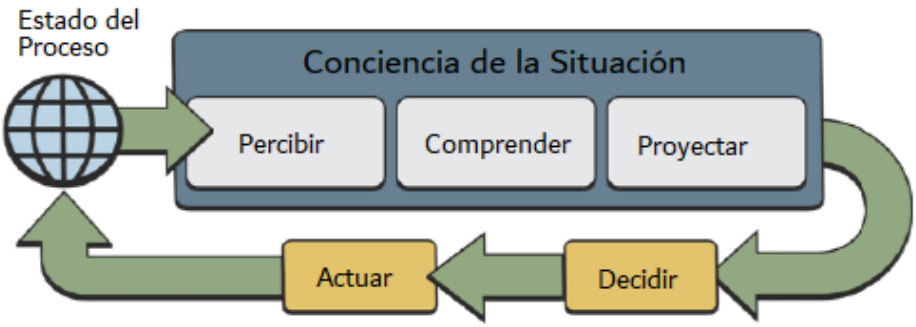


Figura 1.19. Conciencia de la Situación

Se requiere que la interfaz HMI sea diseñada de manera efectiva para facilitar la percepción y la compresión de la situación del proceso. Además, debe proporcionar información relevante de forma clara, concisa y eficiente.

1.4.7.4. Alarmas

En el proceso de monitorización y supervisión, es fundamental contar con mecanismos de alerta que permitan al operador priorizar su atención en situaciones que presenten un comportamiento anormal o que se encuentre fuera de los límites establecidos para el proceso. Para ello, se utiliza un sistema de alarmas y eventos.

Las alarmas desempeñan la función de detectar y notificar automáticamente al usuario sobre fallas en los equipos, desviaciones en el proceso o comportamientos anómalos que requieran una respuesta inmediata. Estas alarmas son esenciales para mantener las condiciones normales de funcionamiento y desempeñan un papel crucial en la seguridad de la planta al proporcionar un medio para reducir riesgos y prevenir daños en situaciones peligrosas [17].

El estándar ANSI/ISA-18.2 “Gestión de Sistemas de Alarmas para Procesos Industriales” ofrece una guía para el diseño, implementación y mantenimiento de sistemas de alarmas que maximizan la seguridad del operador [18]. En los sistemas de control, se asignan conjuntos de valores que determinan el comportamiento para el cual se pueden asignar diferentes tipos de alarmas. Por ejemplo, para la señal de nivel, se pueden establecer los siguientes tipos de alarmas:

- Valor máximo
- Valor alto
- Valor bajo
- Valor mínimo
- Tasa de cambio o velocidad de cambio

Existen ciertas características que tienen las alarmas dentro de su funcionamiento que actúan de forma independiente y se describen a continuación:

- **Modo de la Alarma:** Esta característica es utilizada para habilitar o deshabilitar la alerta de la alarma.
- **Alarma Encendida:** Este es un indicador que se activa cuando los requerimientos configurados en la alarma están presentes.
- **Alarma Reconocida:** Es una propiedad de la alarma que se activa cuando el usuario reconoce o tiene conocimiento de la existencia de la situación anómala.
- **Mensaje en Alarma:** Consiste en un comentario o conjunto de palabras suministradas por el operador al reconocer la alarma.

En las configuraciones de las alarmas, muchas herramientas de desarrollo incluyen como parte de su configuración el estado de la alarma durante el tiempo de ejecución. Estos estados pueden ser los siguientes:

- **Habilitada:** La alarma emite la señal de activación a los clientes configurados.
- **Deshabilitada:** La señal no es enviada a los clientes y, por lo tanto, no se genera ninguna alerta.
- **Silenciada:** La señal es emitida a los clientes y se genera un registro de emisión, pero el cliente no emite una alerta.

Prioridad

Otra configuración importante de las alarmas es su prioridad. El nivel de urgencia de cada alarma se determina en función de su importancia, y se le asigna un valor numérico que indica dicha prioridad. Por ejemplo, cuando la temperatura de un fluido alcanza niveles críticos que pueden tener un impacto significativo, se debe generar una alarma con alta prioridad. Por otra parte, una alarma que simplemente indique un cambio en un elemento puede considerarse menos severa.

En la Tabla 1.1. se muestra un ejemplo de cómo se relaciona la severidad de una alarma con un rango numérico correspondiente.

Tabla 1.1. Prioridad de la Alarmas

Severidad de la Alarma	Estimación Rango
Crítica	1 – 249
Mayor	250 – 499
Menor	500 – 749
Informativa	750 – 999

Además, existen otras situaciones que pueden ocurrir en condiciones normales y no requieren una atención prioritaria por parte del operador. Estas situaciones se denominan eventos y pueden registrarse automáticamente sin generar ningún tipo de aviso.

De esta manera, se observa que las alarmas son un tipo especial de evento que indica un comportamiento anómalo. Se muestran ejemplos de alarmas y eventos en la Tabla 1.2.

Tabla 1.2. Ejemplos de alarmas y eventos

Alarmas	Eventos
Un valor de proceso ha excedido el límite predefinido por ejemplo de Nivel o Temperatura	Cuando se modifica una consigna por parte del operador.
Cuando el valor de proceso no se ajusta a la consigna configurada.	Cuando se modifica un valor discreto. Por ejemplo, cuando se enciende un motor

Un dispositivo de campo no se encuentra en un estado deseado p.ej. una válvula se encuentra abierta cuando se la configuró para estar cerrada	Al momento de realizar un inicio de sesión en el sistema.

Distinguir entre alarmas y eventos es importante para el operador, ya que las alarmas requieren una acción inmediata debido a su naturaleza crítica, mientras que los eventos pueden ser monitoreados sin necesidad de intervención inmediata. Esto permite al operador enfocar su atención en los aspectos más relevantes y priorizar sus acciones de acuerdo con la importancia de cada situación.

1.4.7.5. Historización

Uno de los componentes importantes de una plataforma de control supervisorio es el sistema de historización, el cual se encarga de almacenar los datos históricos del proceso. Las variables que representan los valores del proceso se generan continuamente y sus cambios a lo largo del tiempo reflejan el comportamiento del sistema. Por lo tanto, es fundamental guardar estas señales en un sistema de historización.

Dentro de la plataforma de control supervisorio, existen herramientas específicas que capturan estas señales y las almacenan en un espacio de memoria. Los historiadores tienen la capacidad de recopilar datos de múltiples componentes interconectados en red (pertenecientes a un mismo servidor) y presentar esta información a los clientes de manera eficiente.

En entornos industriales, es recomendable que el almacenamiento de los datos históricos se realice en un nodo independiente, dedicado exclusivamente a este propósito.

Tendencias

Las tendencias son herramientas que permiten representar la evolución de las variables de proceso en un gráfico. Estas gráficas pueden tener parámetros predefinidos o pueden modificarse durante la ejecución.

Es posible representar diferentes variables con sus respectivos valores de forma simultánea en el mismo gráfico. Sin embargo, es importante tener en cuenta la legibilidad de la información cuando se muestran muchas variables o líneas en el gráfico. Un exceso de información puede dificultar la interpretación de los datos.

2. METODOLOGÍA

Para el planteamiento de la estructura metodológica en el presente trabajo se desarrollan dos secciones principales asociadas a cada uno de los módulos propuestos. Por una parte, se describirán los parámetros y restricciones del módulo de control industrial basado en comunicación y, por otra parte, se presentarán las consideraciones para el módulo de la plataforma de control supervisorio.

Los conceptos e información presentados previamente son usados en el diseño e implementación en las dos celdas propuestas. Siguiendo los conceptos, teorías y normativas correspondientes, se procede a desarrollar los módulos.

2.1. Requerimientos ICS

En la presente sección, se propone una arquitectura para el diseño del sistema ICS, el cual consta de diversos componentes que abarcan los diferentes niveles del modelo CIM.

En el nivel de campo y control, se encuentran dos subprocesos ubicados de manera remota. Por lo tanto, se plantea la utilización de dos dispositivos de control o RTUs en cada subproceso, los cuales llevarán a cabo las tareas de control y adquisición de datos. Estas RTUs mantienen una conexión en red Profinet con la unidad central MTU, la cual adquiere las variables de cada unidad remota para posteriormente enviarlas a la plataforma de control supervisorio.

En el siguiente nivel de automatización, se plantea el desarrollo de la plataforma de control supervisorio, encargada de los distintos medios de control, supervisión, adquisición e historización de las variables obtenidas, así como del estado de la red y diagnóstico de los componentes del sistema de control. La arquitectura mencionada se puede observar en la Figura 2.1. la cual propone un sistema ICS integrado.

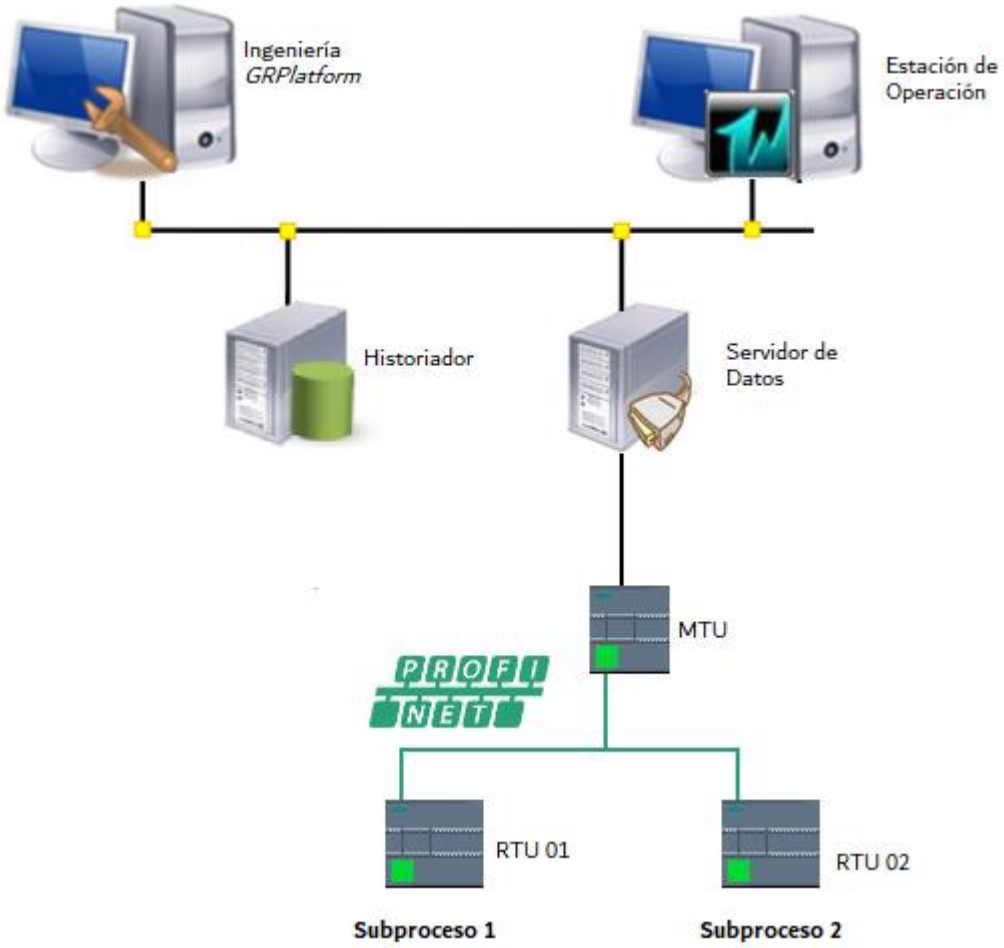


Figura 2.1. Arquitectura ICS

Los componentes de hardware utilizados para realizar el control, la adquisición, supervisión y análisis de datos del sistema se presentan en la Tabla 2.1.

Tabla 2.1. Dispositivos del ICS

Nombre	**Dispositivo**
RTU 01	S7-1200 1214C DC/DC/DC
RTU 02	S7-1200 1214C AC/DC/Rly
MTU	S7-1200 1212C AC/DC/Rly
Switch	SCALANCE X005
Estación de Ingeniería	PC Windows
Estación Servidor de Datos	PC Windows
Estación de Operación	PC Windows

Los requerimientos de los dispositivos de hardware describen en las secciones siguientes.

2.2. Requerimientos Celda de Control

Para llevar a cabo el proceso descrito anteriormente, es necesario contar con un sistema de control distribuido (DCS) que permita controlar cada uno de los subprocesos de forma independiente. Cada sección cuenta con un controlador descentralizado que funciona como una RTU. Las señales adquiridas son centralizadas por un dispositivo que actúa como una MTU.

Cada controlador tiene asignada una dirección IP para su identificación dentro de la red. A continuación, se muestran en la Tabla 2.2. las direcciones correspondientes a cada interfaz de red ubicada en los PLCs.

Tabla 2.2. Direcciones IP

Dispositivo	Dirección
MTU	192.168.10.3 / 24
RTU 01	192.168.10.2 / 24
RTU 02	192.168.10.10 / 24

Estas direcciones pertenecen a la red 192.168.10.0/24.

2.2.1. Características de los Componentes de Automatización

2.2.1.1. Controlador Lógico Programable

La RTU 01 utilizada es un PLC del fabricante Siemens, específicamente el modelo S7-1200, 1214C DC/DC/DC. Para la RTU 02, se emplea otro controlador Siemens, modelo S7-1200, 1214C AC/DC/Rly. Por último, el dispositivo utilizado como MTU para centralizar los datos es un PLC Siemens S7-1200, 1212C AC/DC/Rly. Se muestra una representación del PLC en la Figura 2.2.

Figura 2.2. Siemens S7-1200

Estos modelos de controladores fueron seleccionados debido a su flexibilidad y las capacidades necesarias para la automatización de procesos. Además, presentan una estructura compacta al contar con una fuente de alimentación integrada y un conjunto de entradas y salidas acopladas al dispositivo. Además de estas características, las CPU de los controladores ofrecen ventajas adicionales, como su capacidad de comunicación Profinet, su robustez en entornos industriales y su velocidad tanto el control del circuito como las comunicaciones [19].

A continuación, se muestra una ficha técnica en la Tabla 2.3.

Tabla 2.3. Datos Técnicos PLC S7-1200

Datos Técnicos	**1214C DC/DC/DC (RTU 1)**	**1214C AC/DC/Rly (RTU 2)**	**1212C AC/DC/Rly (MTU)**
Rango de tensión	De 20,4 V DC a 28,8 V DC	De 85 a 264 V AC (47 a 63 Hz)	
Entradas Digitales			
Número	14		8
Tensión nominal	24 V DC a 4 mA, nominal		
Salidas Digitales			
Número	10		6
Tipo	MOSFET	Relé mecánico	
Entradas Analógicas			
Entradas Analógicas	2		
Tipo	Tensión		
Rango Total	De 0 a 10 V		

Rango Desbordamiento	De 11,760 a 11,852 V		
Rango Total (palabras)	De 0 a 27648		
Rango Desbordamiento (palabras)	De 32512 a 32767		
Protocolos			
Controlador Profinet IO	Sí	Sí	Sí
Dispositivo Profinet IO	Sí	Sí	Sí

2.2.1.2. Módulo SM 1234

Para adquirir señales analógicas en el segundo subproceso, se utiliza el módulo de entradas y salidas analógicas de la marca Siemens, modelo SM 1234 como se muestra en la Figura 2.3. El módulo está acoplado al CPU, lo que incrementa su capacidad para leer y escribir señales analógicas.

Figura 2.3. Módulo SM 1234

Los datos técnicos se muestran en la Tabla 2.4.

Tabla 2.4. Módulo de E/S analógicas SM 1234

Datos Técnicos SM 1234	
Tensión de alimentación	24 V
Entradas Analógicas	
Número	4 (Corriente o Tensión)
Tensión Máxima	35 V
Intensidad Máxima	40 mA
Rango Tensión	±10V, ±5V, ±2,5V

Rango Intensidad	4 a 20 mA, 0 a 20 mA
Salidas Analógicas	
Número	2
Rango Tensión	-10 V a +10 V
Rango Intensidad	0 a 20 mA 4 a 20 mA

2.2.1.3. Switch SCALANCE X005

Para establecer la conexión entre los controladores, se emplea un switch industrial que prioriza el protocolo de comunicaciones. El modelo utilizado es el SCALANCE X005, el cual cuenta con cinco puertos RJ45 como se muestra en la Figura 2.4. Este switch se encarga de facilitar la comunicación Profinet entre los PLCs y se utiliza para transmitir los datos hacia el servidor de datos. Los detalles técnicos del switch se muestran en la Tabla 2.5.

Tabla 2.5. Switch SCALANCE X005

Datos Técnicos SCALANCE X005	
Tensión de alimentación	24 V
Terminales de red	5: RJ45 con collar de sujeción
Velocidad de transferencia	10 Mbit/s, 100 Mbit/s
Protección IP	IP30

Figura 2.4. SCALANCE X005

2.2.2. Instrucciones para Comunicación Profinet

El protocolo utilizado para el módulo de control es Profinet, un estándar de comunicación industrial que facilita el tráfico de datos entre los dispositivos de campo.

Las CPU seleccionadas permiten la comunicación TCP estándar. Para establecer la red de comunicación, se utiliza el software de desarrollo TIA Portal, que proporciona las instrucciones TSEND_C y TRCV_C para enviar y recibir datos, respectivamente. Estas instrucciones engloban un conjunto de comandos que se detallarán a continuación.

2.2.2.1. Instrucción TSEND_C

Esta instrucción tiene la función de enviar datos a través de Ethernet y combina las funciones de las instrucciones TCON, TDISCON y TSEND, las cuales se detallan a continuación. TSEND_C permite establecer una conexión TCP con un interlocutor, enviar los datos y, posteriormente, cerrar la conexión.

La ejecución de esta instrucción es asíncrona y lleva a cabo las siguientes funciones en secuencia:

- Configura y establece la conexión de comunicación cuando se detecta un flanco ascendente en el parámetro REQ. Una vez establecida la conexión, la CPU la mantiene y supervisa automáticamente. La configuración del enlace se describe en el parámetro CONNECT.

- La transmisión de los datos se realiza cuando se detecta un flanco ascendente en el parámetro REQ, después de la configuración. Los datos que se transmitirán se especifican en el parámetro DATA.

- La conexión se cierra al finalizar la transmisión de los datos o el parámetro CONT = “0”, de lo contrario la conexión se mantendrá.

El bloque de la instrucción TSEND_C se muestra en la Figura 2.5.

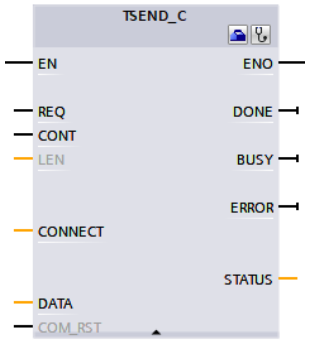


Figura 2.5. TSEND_C

Las especificaciones detalladas de los parámetros se describen en el Anexo A.

2.2.2.2. Instrucción TRCV_C

Esta instrucción se encarga de recibir datos a través de Ethernet y combina las funciones de las instrucciones TCON, TDISCON y TRCV, las cuales se detallarán a continuación. TRCV_C permite establecer una conexión TCP con un interlocutor, recibir los datos y, posteriormente, cerrar la conexión.

Esta instrucción también se ejecuta de forma asíncrona y realiza las siguientes funciones de forma secuencial:

- Configura y establece la conexión cuando el parámetro EN_R tiene el valor “1” y no hay una conexión activa. Una vez establecida la conexión, la CPU la mantiene y supervisa automáticamente.
- Los datos recibidos se almacenan en un área de recepción que se configura en el parámetro DATA.
- Para cerrar la conexión, se espera a que concluya la recepción de los datos y que el parámetro CONT tenga el valor “0”. En caso contrario, la conexión se mantendrá.

El bloque de la instrucción TRCV_C se muestra en la Figura 2.6.

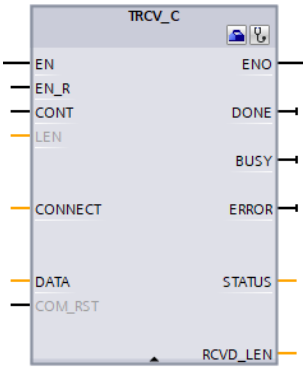


Figura 2.6. TRCV_C

Las especificaciones de los parámetros se encuentran en el Anexo A.

2.2.3. Instrucción LED

Por último, se realiza la lectura del estado de los indicadores de diagnóstico utilizando la instrucción LED, la cual se emplea para diagnosticar el dispositivo, como se muestra en la Figura 2.7.

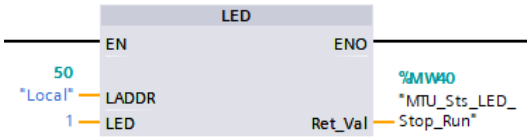


Figura 2.7. Instrucción LED

Los resultados proporcionados por esta instrucción (Ret_Val) indican si un LED está apagado, encendido con un color en específico o parpadeando a una frecuencia determinada con su respectiva tonalidad. Los detalles de las instrucciones detectadas en el parámetro RET_VAL se encuentran en la Tabla 2.6.

Tabla 2.6. Descripción parámetro RET_VAL

RET_VAL (INT)	Descripción
0	LED no disponible
1	Apagado
2	Primer color encendido
3	Segundo color encendido
4	Primer color parpadeante a 2 Hz
5	Segundo color parpadeante a 2 Hz

6	Primer y segundo color parpadeando alternativamente a 2 Hz
7	LED encendido con el primer color
8	LED encendido con el segundo color
9	LED inexistente

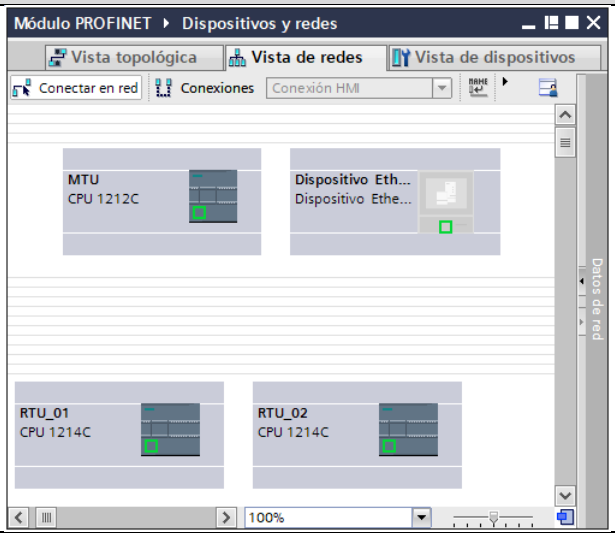
2.3. Implementación Celda de Control

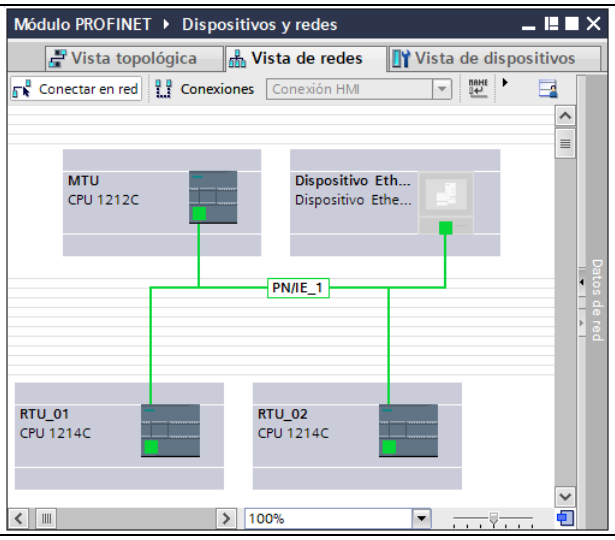
Para la implementación del módulo de control, se presentan las siguientes configuraciones para la implementación de la red y las instrucciones necesarias para enviar tramas utilizando el protocolo Profinet. Un instructivo para una conexión más sencilla se la muestra en el Anexo C.

2.3.1. Configuración de la Red

Para realizar la configuración de la red se procede al portal “Dispositivo y redes” y en la “Vista de red” de “Configuración de dispositivos” se realiza la conexión, como se muestra en la Tabla 2.7.

Tabla 2.7. Datos de Configuración Conexiones TCP

Procedimiento	Resultado
En “Vista de redes” y “Conectar en red” se visualiza los dispositivos de hardware existentes	

Se selecciona el puerto de cada dispositivo y se arrastra hasta el switch de red para crear la conexión. De esta forma se crea la Red PN/IE_1 (Profinet)	

2.3.2. Implementación de la Conexión TCP

Para configurar la red, se propone establecer cuatro conexiones lógicas distintas entre los controladores para el envío y recepción de datos, tal como se muestra en la Tabla 2.8. Estas conexiones se encuentran en la red Profinet PN/IE_1.

Tabla 2.8. Puertos Lógicos – Conexión TCP

Desde	Hacia	Puerto
MTU	RTU 01	2001
RTU 01	MTU	2002
MTU	RTU 02	2003
RTU 02	MTU	2004

Esta configuración se lleva a cabo mediante las instrucciones TSEND_C y TRCV_C implementadas en cada PLC. Para la instrucción TSEND_C, se utiliza una señal de reloj con una frecuencia de 10Hz en el parámetro REQ para iniciar la solicitud de transmisión en cada flanco ascendente.

La programación realizada en la MTU se centra principalmente en el intercambio de datos a través de los puertos mencionados anteriormente, permitiendo el envío y recepción mediante Profinet desde y hacia los controladores RTU 01 y RTU 02.

Las instrucciones TSEND_C programadas en la MTU envían datos TCP a la RTU 01 y RTU 02, utilizando los puertos lógicos 2001 y 2003 respectivamente, como se muestra en la Figura 2.8.

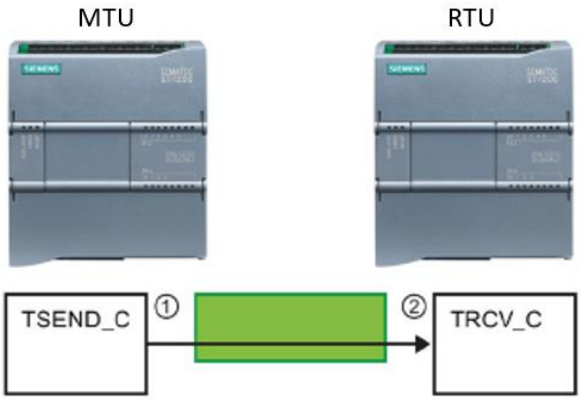


Figura 2.8. TSEND_C

Asimismo, las instrucciones TRCV_C ubicadas en la MTU reciben datos de las RTUs utilizando los puertos locales 2002 y 2004 respectivamente, como se muestra en la Figura 2.9.

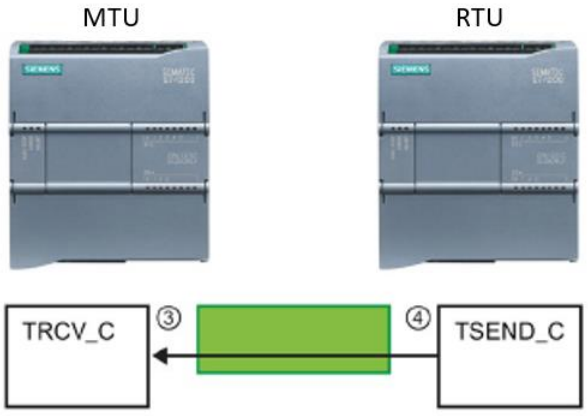


Figura 2.9. TRCV_C

Estas instrucciones de envío y recepción se encargan de transferir Bloques de Datos (DB) que contienen todas las variables. Por esta razón, se crearon cuatro DB en la MTU, como se muestra en la Tabla 2.9.

Tabla 2.9. Bloque de Datos MTU

Nombre	DB
"DatosEnviar_RTU_01"	[DB1]
"DatosRecibir_RTU_01"	[DB2]
"DatosEnviar_RTU_02"	[DB3]
"DatosRecibir_RTU_02"	[DB4]

Para enviar información, se realiza el movimiento de las variables locales hacia los DB1 y DB3 respectivamente. Por otro lado, las variables recibidas se mueven desde los DB2 y DB4 hacia las variables locales.

De esta manera, podemos observar que cada una de las variables, tanto las enviadas como las recibidas, ocupan un espacio de memoria local en la MTU (en forma de Marcas). Estas marcas serán escritas o leídas por el servidor de datos.

En las unidades RTU, se configuran DB con la misma longitud que los realizados en la unidad MTU. Estos DB deben tener características similares para el proceso de transmisión. Los detalles de los DB establecidos se presentan en la Tabla 2.10.

Tabla 2.10. Bloque de Datos RTU

RTU	Nombre	DB
RTU 01	“Datos Enviar”	[DB1]
	“Datos Recibir”	[BD2]
RTU 02	“Datos Enviar”	[DB1]
	“Datos Recibir”	[BD2]

2.3.2.1. Conexión Puerto 2001

El Puerto 2001 se utiliza para la transmisión de las variables de escritura desde la unidad MTU hacia el dispositivo RTU 01 ubicado en el subproceso 1.

TSEND_C_RTU_01 (MTU)

La configuración de instrucción TSEND_C_RTU_01 ubicada en la MTU se muestra en la Figura 2.10.

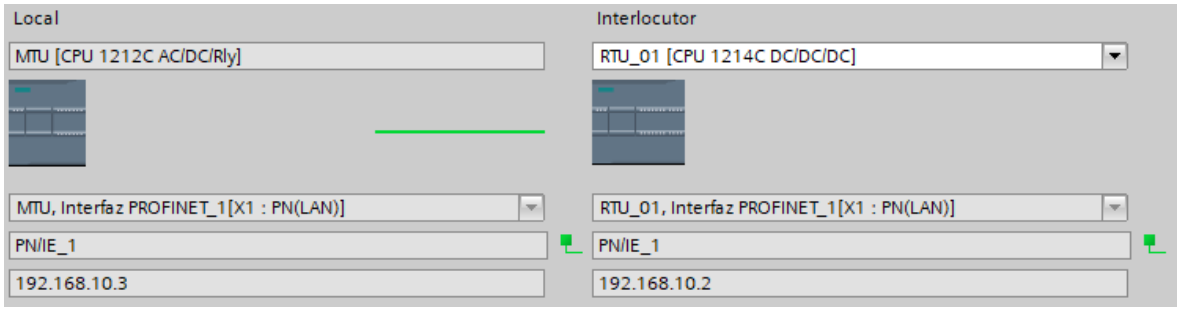


Figura 2.10. MTU: TSEND_C_RTU_01

En la Tabla 2.11. Se detalla la instrucción TSEND_C ubicada en el dispositivo MTU.

Tabla 2.11. Instrucción en MTU

MTU		
TSEND_C (RTU_01)		
Descripción	**Nombre**	**Abreviación**
General	TSEND_C_DB_RTU_01	%DB100
Datos de Conexión	MTU_Send_DB_RTU_01	%DB110
DATA	DatosEnviar_RTU_01	%DB1
ID conexión	1 (2001)	

TRCV_C (RTU 01)

La configuración de instrucción TRCV_C ubicada en la RTU 01 se muestra en la Figura 2.11.

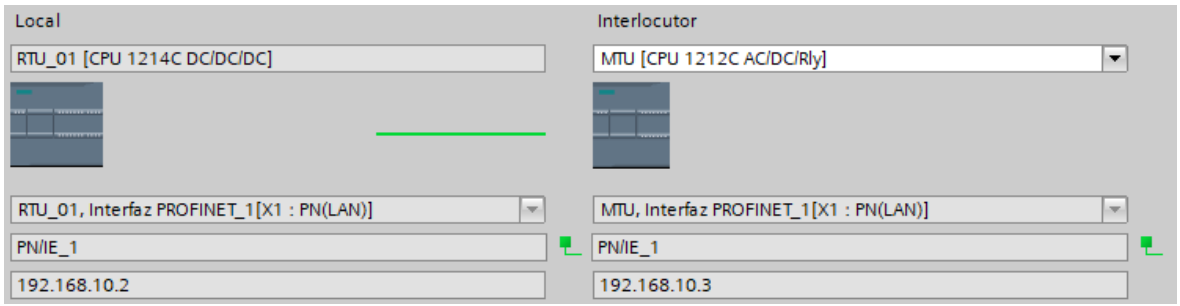


Figura 2.11. RTU_01: TRCV_C

En la Tabla 2.12. Se detalla la instrucción TRCV_C ubicada en el dispositivo RTU 01.

Tabla 2.12. Instrucción en RTU 01

RTU_01		
TRCV_C		
Descripción	**Nombre**	**Abreviación**
General	TRCV_C_DB	%DB200
Datos de Conexión	RTU_01_Receive_DB	%DB210
DATA	DatosRecibir	%DB2
ID conexión	1 (2001)	

Las variables configuradas en los dos DB para la conexión 2001 se detallan en la Tabla 2.13.

Tabla 2.13. DB1 (MTU) – DB2 (RTU 01)

%DB1 (MTU) / % DB2 (RTU 1)	
Name	**Data type**
Reset_Error_TSEND_C	Bool
Reset_Error_TRCV_C	Bool
ZV_101A_CmdOpen	Bool
ZV_101A_CmdClose	Bool
ZV_101B_CmdOpen	Bool
ZV_101B_CmdClose	Bool
ZV_102A_CmdOpen	Bool
ZV_102A_CmdClose	Bool
ZV_102B_CmdOpen	Bool
ZV_102B_CmdClose	Bool
ZV_103_CmdOpen	Bool
ZV_103_CmdClose	Bool
EY_104_Cmd	Bool
PN_Bool_1_Out	Bool
PN_Bool_2_Out	Bool
Cacao_Milk_SP_Level	Real
PN_Real_Out	Real

2.3.2.2. Conexión Puerto 2002

El Puerto 2002 realiza la lectura de las variables del subproceso 1 y del dispositivo RTU 01, transmitiéndolas hacia la unidad MTU.

TSEND_C (RTU 01)

La configuración de instrucción TSEND_C ubicada en la RTU 01 se muestra en la Figura 2.12.

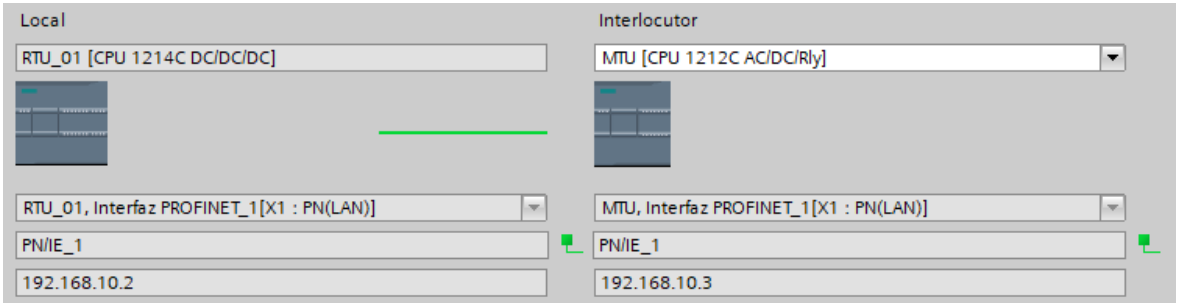


Figura 2.12. RTU_01: TSEND_C

En la Tabla 2.14. Se detalla la instrucción TSEND_C ubicada en el dispositivo RTU 01.

Tabla 2.14. Instrucción en RTU 01

RTU_01		
TSEND_C		
Descripción	**Nombre**	**Abreviación**
General	TSEND_C_DB	%DB100
Datos de Conexión	RTU_01_Send_DB	%DB110
DATA	DatosEnviar	%DB1
ID conexión	2 (2002)	

TRCV_C_RTU_01 (MTU)

La configuración de instrucción TRCV_C_RTU_01 ubicada en la MTU se muestra en la Figura 2.13.

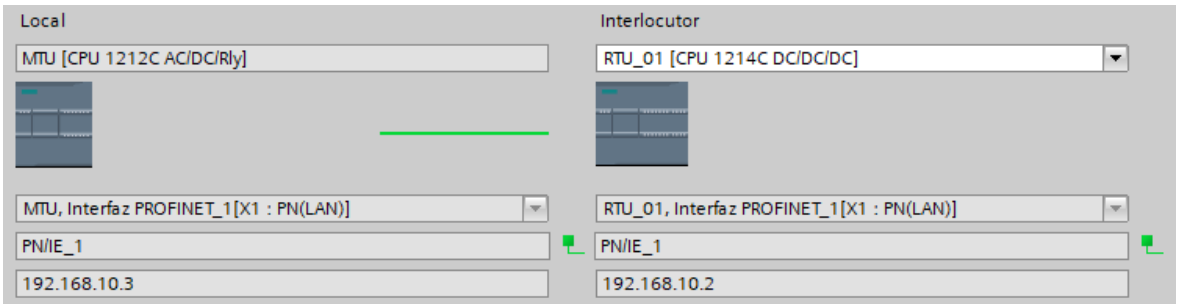


Figura 2.13. MTU: TRCV_C_RTU_01

En la Tabla 2.15. Se detalla la instrucción TRCV_C ubicada en el dispositivo MTU.

Tabla 2.15. Instrucción en MTU

MTU		
TRCV_C (RTU_01)		
Descripción	**Nombre**	**Abreviación**
General	TRCV_C_DB_RTU_01	%DB200
Datos de Conexión	MTU_Receive_RTU_01_DB	%DB210
DATA	DatosRecibir_RTU_01	%DB2
ID conexión	2 (2002)	

Las variables configuradas en los dos DB para la conexión 2002 se detallan en la Tabla 2.16.

Tabla 2.16. DB2 (MTU) – DB1 (RTU 01)

%DB2 (MTU) / % DB1 (RTU 1)	
Name	**Data type**
TSEND_C_DONE	Bool
TSEND_C_BUSY	Bool
TSEND_C_ERROR	Bool
TSEND_C_ERROR_DETECTED	Bool
TSEND_C_STATUS	Word
TSEND_C_STATUS_DETECTED	Word
TRCV_C_DONE	Bool
TRCV_C_BUSY	Bool
TRCV_C_ERROR	Bool
TRCV_C_ERROR_DETECTED	Bool
TRCV_C_STATUS	Word
TRCV_C_STATUS_DETECTED	Word
Sts_LED_Stop_Run	Word
Sts_LED_ERROR	Word
ZV_101A_CS	Bool
ZV_101A_OS	Bool
ZV_101B_CS	Bool
ZV_101B_OS	Bool

2.3.2.3. Conexión Puerto 2003

El Puerto 2003 es utilizado para realizar la transmisión de las variables de escritura desde la unidad MTU hacia el dispositivo RTU 02 ubicado en el subproceso 2.

TSEND_C_RTU_02 (MTU)

La configuración de instrucción TSEND_C_RTU_02 ubicada en la MTU se muestra en la Figura 2.14.

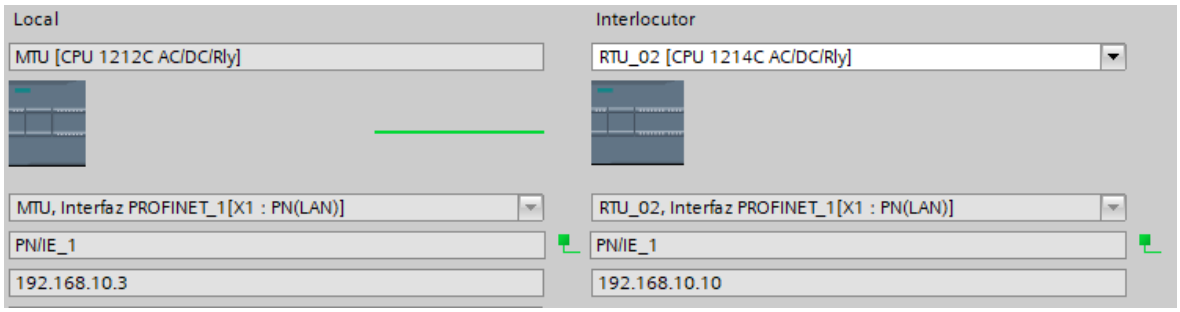


Figura 2.14. MTU: TSEND_C_RTU_02

En la Tabla 2.17. Se detalla la instrucción TSEND_C ubicada en el dispositivo MTU.

Tabla 2.17. Instrucción en MTU

MTU		
TSEND_C (RTU_02)		
Descripción	**Nombre**	**Abreviación**
General	TSEND_C_DB_RTU_02	%DB300
Datos de Conexión	MTU_Send_DB_RTU_02	%DB310
DATA	DatosEnviar_RTU_02	%DB3
ID conexión	3 (2003)	

TRCV_C (RTU 02)

La configuración de instrucción TRCV_C ubicada en la RTU 02 se muestra en la Figura 2.15.

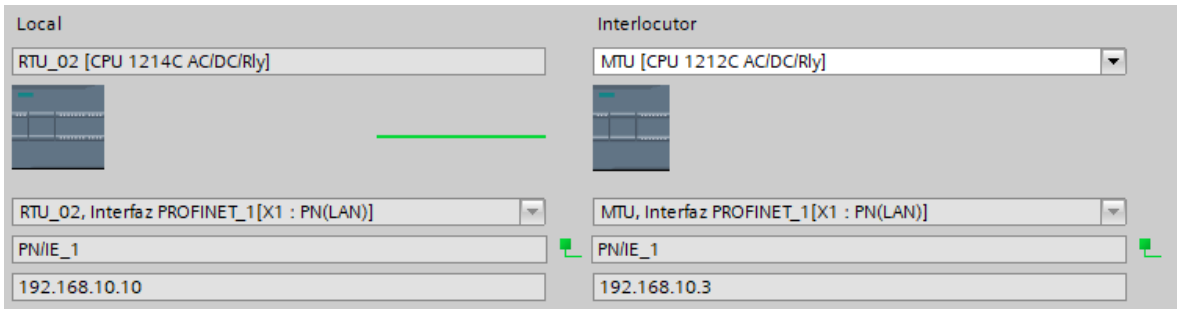


Figura 2.15. RTU_02: TRCV_C

En la Tabla 2.18. Se detalla la instrucción TRCV_C ubicada en el dispositivo RTU 02.

Tabla 2.18. Instrucción en RTU 02

RTU_02		
TRCV_C		
Descripción	**Nombre**	**Abreviación**
General	TRCV_C_DB	%DB200
Datos de Conexión	RTU_02_Receive_DB	%DB210
DATA	DatosRecibir	%DB2
ID conexión	3 (2003)	

Las variables configuradas en los dos DB para la conexión 2003 se detallan en la Tabla 2.19.

Tabla 2.19. DB3 (MTU) – DB2 (RTU 02)

%DB3 (MTU) / % DB2 (RTU 2)	
Name	**Data type**
Reset_Error_TSEND_C	Bool
Reset_Error_TRCV_C	Bool
ZV_201A_CmdOpen	Bool
ZV_201A_CmdClose	Bool
ZV_201B_CmdOpen	Bool
ZV_201B_CmdClose	Bool
EY202_Cmd	Bool
YV_200A_SP	Word
YV_200B_SP	Word
SP_Level_Tank1	Real
SP_Temp_Tank2	Real
PN_Real_Out	Real
EMERGENCY_STOP	Bool

2.3.2.4. Conexión Puerto 2004

El Puerto 2004 realiza la lectura de las variables del subproceso 2, del dispositivo RTU 02, y se las transmite hasta la unidad MTU.

TSEND_C (RTU 02)

La configuración de instrucción TSEND_C ubicada en la RTU 02 se muestra en la Figura 2.16.

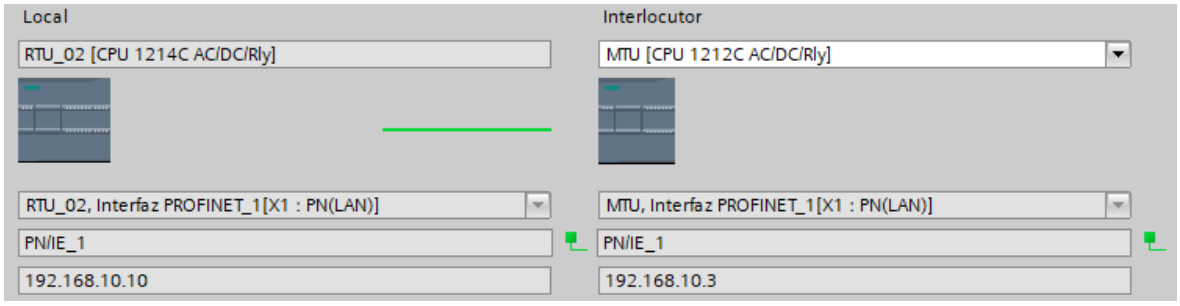


Figura 2.16. RTU_02: TSEND_C

En la Tabla 2.20. Se detalla la instrucción TSEND_C ubicada en el dispositivo RTU 02.

Tabla 2.20. Instrucción en RTU 02

RTU_02		
TSEND_C		
Descripción	**Nombre**	**Abreviación**
General	TSEND_C_DB	%DB100
Datos de Conexión	RTU_02_Send_DB	%DB110
DATA	DatosEnviar	%DB1
ID conexión	4 (2004)	

TRCV_C_RTU_02 (MTU)

La configuración de instrucción TRCV_C_RTU_02 ubicada en la MTU se muestra en la Figura 2.17.

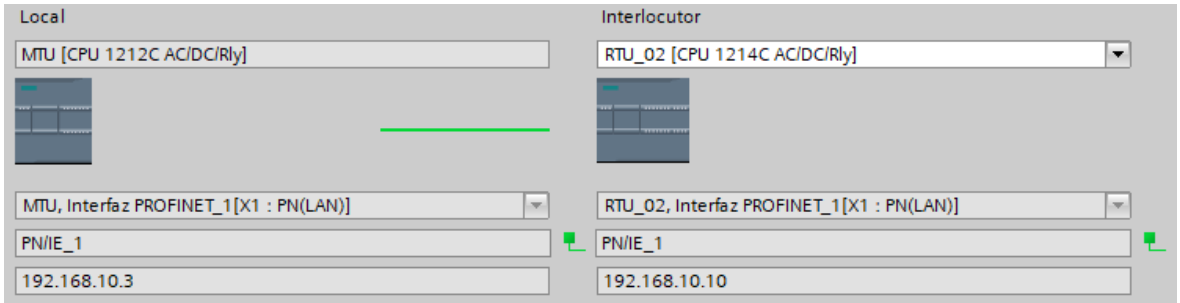


Figura 2.17. MTU: TRCV_C_RTU_02

En la Tabla 2.21. Se detalla la instrucción TRCV_C ubicada en el dispositivo MTU.

Tabla 2.21. Instrucción en MTU

MTU		
TRCV_C (RTU_02)		
Descripción	**Nombre**	**Abreviación**
General	TRCV_C_DB_RTU_02	%DB400
Datos de Conexión	MTU_Receive_DB_RTU_02	%DB410
DATA	DatosRecibir_RTU_02	%DB4
ID conexión	4 (2004)	

Las variables configuradas en los dos DB para la conexión 2004 se detallan en la Tabla 2.22.

Tabla 2.22. DB4 (MTU) – DB1 (RTU 02)

%DB4 (MTU) / % DB1 (RTU 2)	
Name	**Data type**
TSEND_C_DONE	Bool
TSEND_C_BUSY	Bool
TSEND_C_ERROR	Bool
TSEND_C_ERROR_DETECTED	Bool
TSEND_C_STATUS	Word
TSEND_C_STATUS_DETECTED	Word
TRCV_C_DONE	Bool
TRCV_C_BUSY	Bool
TRCV_C_ERROR	Bool
TRCV_C_ERROR_DETECTED	Bool
TRCV_C_STATUS	Word
TRCV_C_STATUS_DETECTED	Word
Sts_LED_Stop_Run	Word
Sts_LED_ERROR	Word
ZV_201A_CS	Bool
ZV_201A_OS	Bool
ZV_201B_CS	Bool
ZV_201B_OS	Bool
YV_200A_CS	Bool
YV_200A_OS	Bool
YV_200B_CS	Bool
YV_200B_OS	Bool
LT_202_Sts	Bool
EY202_Sts	Bool
PN_Bool_1_In	Bool
PN_Bool_2_In	Bool
YV_200A_PV	Word

YV_200B_PV	Word
LT_200_PV	Real
TT_202_PV	Real
PN_Real_IN	Real

2.4. Requerimientos Plataforma de Control Supervisorio

2.4.1. Arquitectura de los Componentes de la Plataforma

Para las tareas de control supervisorio se proponen diversos componentes relacionados con la adquisición de datos, el registro histórico de variables y operación de la planta. Por lo tanto, es necesario disponer de una arquitectura que nos permita cumplir con estas funciones. A continuación, se describen las estaciones propuestas.

La primera estación propuesta se encarga del desarrollo y configuración de la plataforma, debe contar con el hardware y software necesario para realizar dichas funciones y establecer una conexión con los distintos componentes remotos. Esta estación de ingeniería realiza la creación y las modificaciones a los componentes presentes en la plataforma y posteriormente desplegará dicho desarrollo en las siguientes estaciones.

La siguiente estación corresponde al servidor de datos, esta estación independiente se encarga de la comunicación con los dispositivos ubicados en el módulo de control y también transmiten dichas variables a las aplicaciones cliente presentes en la plataforma de control supervisorio.

La siguiente estación propuesta es la de operación, aquí se despliega la interfaz HMI con la cual los operadores realizan el control de la planta. La estación de operación es una parte fundamental en la arquitectura ya que realiza un acercamiento directo de los usuarios con el sistema de control.

Por último, se dispone de la estación del historiador. En esta estación se realiza el almacenamiento de algunas de las variables adquiridas, las cuales luego pueden ser accedidas por las aplicaciones clientes del historiador. Una de estas aplicaciones permite la observación de las variables almacenadas en el histórico a través de gráficas de tendencias.

La arquitectura de las cuatro estaciones mencionadas se las observa en la Figura 2.18.

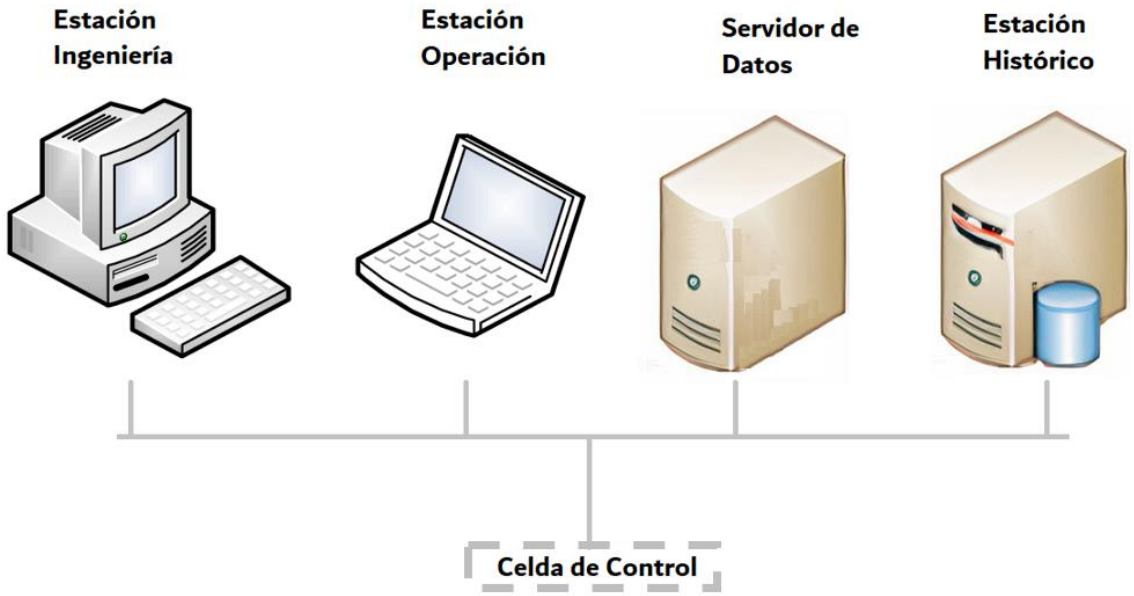


Figura 2.18. Estaciones Plataforma de Control Supervisorio

2.4.2. Aplicación de Control Supervisorio

La aplicación de control supervisorio cumple varias funciones en el sistema, incluyendo la adquisición de datos en tiempo real, administra los eventos y alarmas, centraliza la seguridad, procesa los datos, realiza lanzamientos remotos a los equipos en red y permite una ingeniería colaborativa.

La herramienta utilizada para implementar esta aplicación de control supervisorio es *ArchestrA* de *Wonderware*, la cual tiene las prestaciones necesarias para cumplir con los requerimientos planteados. Esta aplicación de software permite el acceso

a la configuración, la elaboración de código, visualización, historización, y documentación.

Uno de los términos utilizados por la herramienta *ArchestrA* es el denominado *Galaxy*. Este es el nombre asignado la aplicación de control supervisorio perteneciente a una planta específica y su información de configuración, y define el nombre del espacio en el cual todos los objetos y componentes se encuentran. El *Galaxy* puede alojarse en una o varias PCs conectadas en red.

El *Galaxy* se puede visualizar desde dos perspectivas: el modo de configuración y el modo de ejecución. El primero se almacena en una estructura de archivos, y se ubica en una PC denominada *Galaxy Repository* (GR). Esta es una PC única y por su naturaleza, es necesaria solo una por planta. Por otra parte, el modo de ejecución se efectúa durante el *runtime* o tiempo de ejecución de la planta [20].

2.4.3. Servidores de Datos

Los servidores de acceso a datos están diseñados para proporcionar una conectividad simultánea entre las aplicaciones cliente y los dispositivos que proveen los datos. Para esta funcionalidad, se utilizan controladores de comunicación conocidos como *Data Access Server* (*DA Servers*). Estos controladores están diseñados específicamente para establecer la conectividad entre el dispositivo de campo MTU y las aplicaciones cliente que requieren acceder a dichos datos.
En la implementación de la plataforma propuesta, se emplea un DA Server específico para acceder y controlar las variables del proceso emulado. Esta característica será detallada en la descripción de la implementación de la plataforma.

2.4.4. Software de Visualización de HMI

Para realizar un óptimo control y manejo de la planta se propone utilizar la herramienta InTouch. Este software es un interfaz humano-máquina utilizado para

el control, la visualización del proceso y la gestión de operaciones. Se utiliza dicha herramienta debido a su arquitectura, innovación, conectividad e integración de dispositivos.

Cuenta con la integración de la conectividad hacia los *drivers* de comunicación, enlace entre los atributos de los componentes, representación gráfica de elementos para su visualización, manejo de la seguridad, alarmas e instrucciones programadas.

2.4.5. Planificación de la Aplicación

Al implementar un proyecto en una aplicación de control supervisorio, se sugiere seguir una planificación para obtener un modelo funcional de la planta. Esta información sirve como guía para la creación de la aplicación. En la Figura 2.19. se muestran las etapas de la planificación.

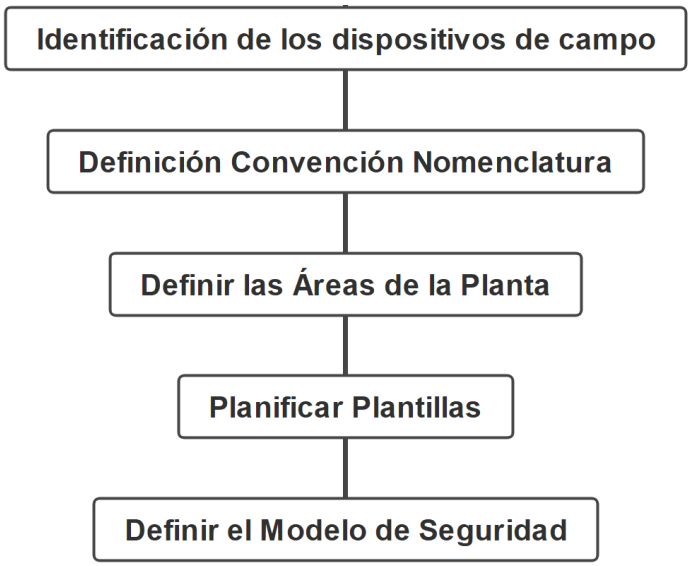


Figura 2.19. Planificación de Aplicación

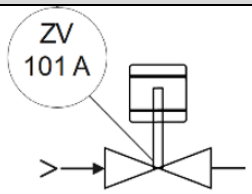
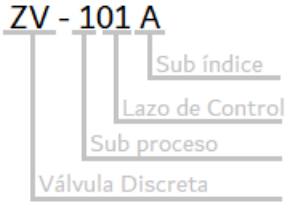
- **Identificación de los dispositivos de campo**

Se busca identificar los dispositivos que se utilizarán en campo y su funcionalidad. Se crea una lista de cada componente y se agrupan, identificando los requisitos funcionales, como alarmas, seguridades, gráficos, entre otros.

- **Definición de una Convención de Nomenclatura**

Se define una convención de nomenclatura basada en normativas o estándares internos de la empresa. Siguiendo estas normativas, se propone el uso de los nombres mostrados en los diagramas P&ID para nombrar las instancias dentro de la aplicación de control supervisorio. A continuación, se muestra un ejemplo en la Tabla 2.23. para asociar cada elemento con su respectivo nombre.

Tabla 2.23. Nomenclatura

Símbolo P&ID	Nomenclatura Instancia	Detalle del Objeto
		Inlet Valve Cacao

- **Definir las Áreas de la Planta**

Las áreas representan agrupaciones lógicas de las secciones físicas de la planta, y se incluye una sección que incluye los elementos del sistema de control dentro de la aplicación de control supervisorio.

- **Planificar Plantillas**

Se generan las instancias de cada plantilla, representando cada dispositivo específico. Es importante recordar que los cambios realizados en las plantillas se propagarán a sus instancias derivadas.

- **Definir el Modelo de Seguridad**

En esta etapa, se habilita la clasificación de seguridad en cada entidad. Este proceso se describe más adelante.

2.4.6. Sistema de Seguridad

Para garantizar la seguridad en la Plataforma de Control Supervisorio, se propone utilizar el modelo presentado por *ArchestrA*, el cual está diseñado para restringir las

actividades no autorizadas por parte de los usuarios. Este sistema de seguridad respalda el funcionamiento normal de los sistemas de automatización. Las contraseñas son encriptadas y se almacenan en una base de datos asequible.

Entre los diferentes modos autenticación que ofrece la herramienta, se propone utilizar aquel que se basa en las configuraciones guardadas localmente dentro del servidor de aplicación.

2.4.6.1. Clasificación de Seguridad en los Atributos

Un atributo se refiere a una señal o característica específica de un objeto que puede ser modificado de diferentes formas según la configuración de seguridad. La forma en que se pueden cambiar los valores del atributo está relacionada con los roles de cada usuario. A continuación, se presentan las diferentes clasificaciones de seguridad propuestas por las herramientas de desarrollo de las aplicaciones de control supervisorio:

- **Acceso libre:** Permite que cualquier usuario, sin importar su privilegio, pueda escribir directamente el valor del atributo. Se utiliza en tareas de emergencia o cuando hay restricciones de tiempo que podrían verse afectadas por un proceso de inicio de sesión.
- **Operación**: Asociada a los atributos que son escritos por los operadores durante un proceso rutinario, como consignas, control de un PID, comandos discretos, entre otros.
- **Escritura segura**: También está asociada a los atributos escritos por los operadores, pero con la diferencia de que se requiere una re-autenticación del usuario antes de ejecutar el cambio, solicitándole que introduzca nuevamente sus credenciales.
- **Escritura Verificada**: Son escritas igualmente por los operarios de forma rutinaria, pero para ejecutar el cambio, se solicita la autenticación de un segundo usuario el cual tenga el privilegio correspondiente.

- **Sintonizar**: La escritura en los atributos configurados con esta clasificación se considera una actividad de afinación o sintonización. Algunos ejemplos son los ajustes de las consignas en las alarmas y la sensibilidad en los controladores PID, etc.
- **Configuración**: Cuando se realiza un cambio en estos atributos, se considera un cambio significativo en las configuraciones. Por esta razón, se requiere que el objeto esté en modo de configuración.
- **Solo lectura**: Los atributos configurados con esta clasificación no pueden ser escritos durante el modo de ejecución.

2.4.6.2. Roles y Grupos de Seguridad

En la configuración de los atributos, se ha discutido previamente cómo permitir a los usuarios escribir o modificar dichos atributos. Dado que cada usuario desempeña funciones diferentes dentro de la planta, es necesario establecer un esquema para la administración de los permisos y privilegios asignados a cada usuario. Aquí es donde entran en juego los conceptos de grupos de seguridad y roles.

Los grupos de seguridad son conjuntos de elementos que comparten comportamientos relacionados con la seguridad. Por ejemplo, los elementos de un subproceso están contenidos en un grupo de seguridad específico, y cada elemento debe pertenecer a algún grupo de seguridad. Se muestra un ejemplo en la Figura 2.20.

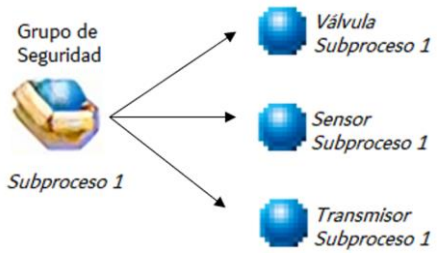


Figura 2.20. Ejemplo Grupos de Seguridad

Por otro lado, los roles están asociados a funciones o permisos específicos. Por ejemplo, el rol de “operadores” tiene permisos de operación, mientras que el rol de “supervisores” tiene permisos de operación y supervisión. En otras palabras, a los

roles se les asigna permisos en función de la función que desempeñan. Un ejemplo del rol de “operadores del subproceso 1” se muestra en la Figura 2.21.

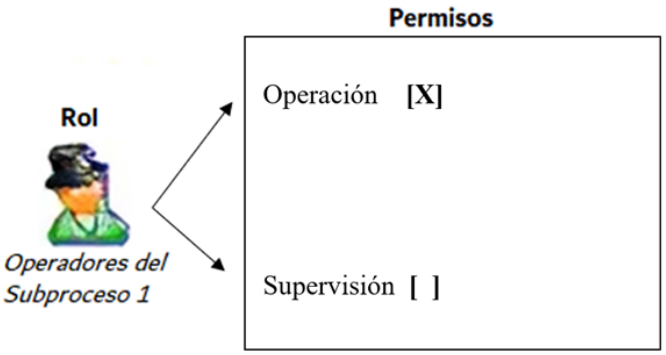


Figura 2.21. Ejemplo Rol

Además, un usuario puede tener uno o más roles. Por ejemplo, un usuario llamado “Supervisor” puede tener asociado dos roles: el de “supervisores”, el de “operadores del subproceso 1” y el de “operadores del Subproceso 2”. Un ejemplo del usuario “Supervisor” se muestra en la Figura 2.22.

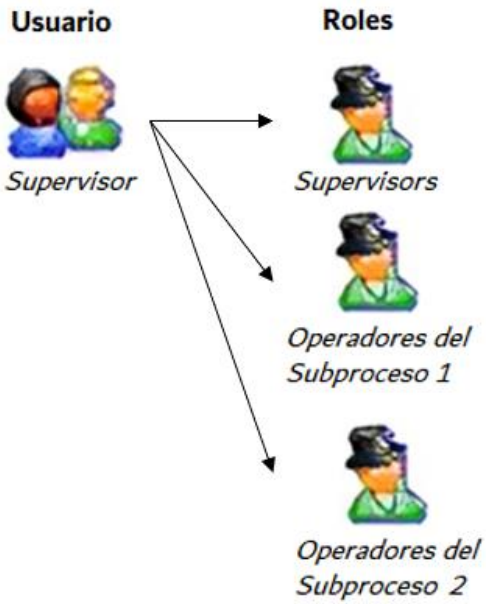


Figura 2.22. Ejemplo Usuarios

2.4.7. Software de Historización

El software utilizado para implementar los requerimientos de historización es el *Wonderware Historian*. Esta herramienta proporciona las funcionalidades necesarias para almacenar datos históricos y posee una base de datos de alto rendimiento específicamente diseñada para un historiador de procesos. Combina las capacidades de una base de datos relacional con el manejo del historiador y es compatible con los demás componentes de la plataforma de control supervisorio.

El *Wonderware Historian* está diseñado para recopilar una gran cantidad de datos de la planta con la máxima resolución y una alta tasa de transferencia de datos.

Para almacenar estas señales en una base de datos del histórico, es necesario la instalación de la herramienta *Wonderware Historian* en un nodo de la misma red.

El historiador puede almacenar diferentes categorías o tipos de datos de los atributos, incluyendo datos numéricos (flotantes, dobles, enteros) y no numéricos (booleanos, cadenas de caracteres).

2.4.7.1. Visualización de Gráficas

Dentro de la plataforma de control, es necesario contar una aplicación cliente que permita visualizar los valores historiados. La herramienta *Historian Client Trend* cumple con esta función.

Estas herramientas suelen incluir dos tipos de gráficos diferentes: una curva de tendencia regular y un diagrama de dispersión bidimensional [21].

Al agregar etiquetas (valores de la base de datos) a los gráficos, es posible manipularlos de diversas formas, como realizar paneo, acercamiento y escalamiento para explorar los datos con mayor detalle y precisión.

2.5. Implementación Plataforma de Control Supervisorio

En las siguientes subsecciones se describe la implementación de la plataforma de control supervisorio utilizada para el monitoreo de la planta. El módulo implementado consta de diversas estaciones utilizadas para una función específica. Las estaciones presentes en la red industrial son las siguientes:

- Estación de Ingeniería (Configuración, en donde se almacena el GR)
- Estación de Operación
- Servidor de Datos
- Histórico del Proceso

Es importante destacar que la herramienta *ArchestrA* se refiere a la aplicación de control supervisorio como el servidor de aplicación.

2.5.1. Creación y Planificación del Servidor de Aplicación

Para la planificación de un modelo funcional dentro de la plataforma, se propone distribuir la aplicación de control supervisorio de la siguiente manera, como se muestra en la Figura 2.23.

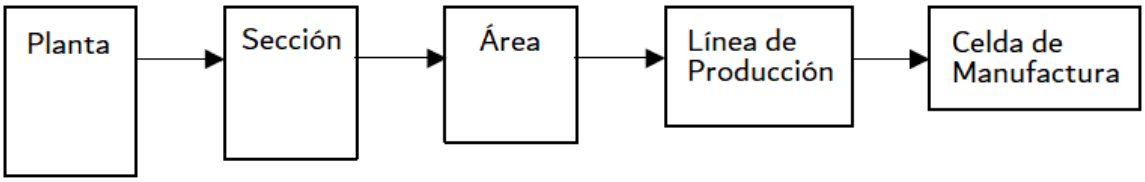


Figura 2.23. Distribución Modelo

Para crear un nuevo servidor de aplicación, se comienza por crear un nuevo Galaxy. En la aplicación de desarrollo *ArchestrA IDE*, iniciamos con un nuevo *Galaxy* de tipo *Base_Application_Server.cab*, como se muestra en la Figura 2.24.

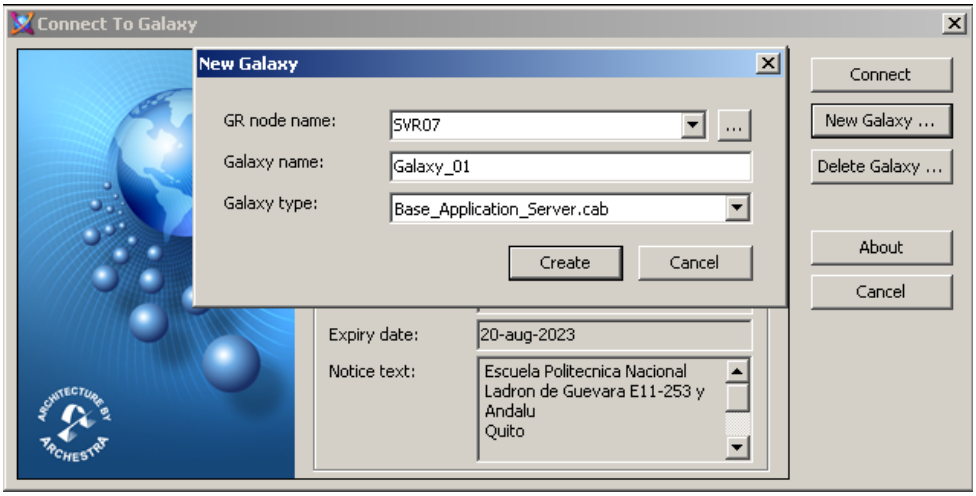


Figura 2.24. Creación Galaxy_01

Al crear un nuevo servidor de aplicación y establecer la conexión con él, se despliega la interfaz de la plataforma.

A continuación, se asigna cada uno de los componentes físicos a sus respectivos objetos dentro del servidor de aplicación. A cada componente se le asignó previamente su nomenclatura. Debido a su naturaleza, estos elementos se clasifican según una Plantilla de Objeto, como se muestra en la Tabla 2.24.

Tabla 2.24. Instancias asociadas a la Plantilla

Plantilla Objeto	Nomenclatura Instancia	Detalle del Objeto
$Valve_Discrete	ZV-101A	*Inlet Valve Cacao*
	ZV-101B	*Outlet Valve Cacao*
	ZV-102A	*Inlet Valve Milk*
	ZV-102B	*Outlet Valve Milk*
	ZV-103	*Outlet Valve Chocolate*
	ZV-201A	*Inlet Valve*
	ZV-201B	*Outlet Valve*
$Valve_Analog	YV-200A	*Inlet Level Tank*
	YV-200B	*Inlet Heater Tank*
$Actuator_Discrete	EY-104	*Cacao Mixer*
	EY-202	*Heater*
$Level_Sensor	LT-101	*Level Sensor Cacao*
	LT-102	*Level Sensor Milk*
	LT-200	*Level Sensor*
$Temp_Sensor	TT-202	*Temp Sensor*

$Sensor_Discrete	PT-103	*Chocolate Position Sensor*
	PT-104	*Cacao Position Sensor*
	LT-202	*Sensor Position Level*

* Elementos en Subproceso 01 con color

Algunas de estas Plantillas se derivan de otras de mayor nivel, que son más generales. En la Tabla 2.25 se presentan las Plantillas de las cuales se derivan, y se han creado ciertas instancias de una Plantilla principal, las cuales se muestran en toda la fila.

Tabla 2.25. Plantillas Derivadas

Plantilla Principal	Plantilla Derivada	2da Plantilla Derivada
$gAnalogDevice	$Sensor_Analog	$Level_Sensor
		$Temp_Sensor
	$Valve_Analog	
$gDiscreteDevice	$Actuator_Discrete	
	$Sensor_Discrete	
	$Valve_Discrete	
$gUserDefined	$SP_Obj	

Además, se generan nuevas Plantillas derivadas de cada Objeto que se utiliza, y se las mueve a otras carpetas dentro de la caja de herramientas de Plantillas, como se ilustra en la Figura 2.25.

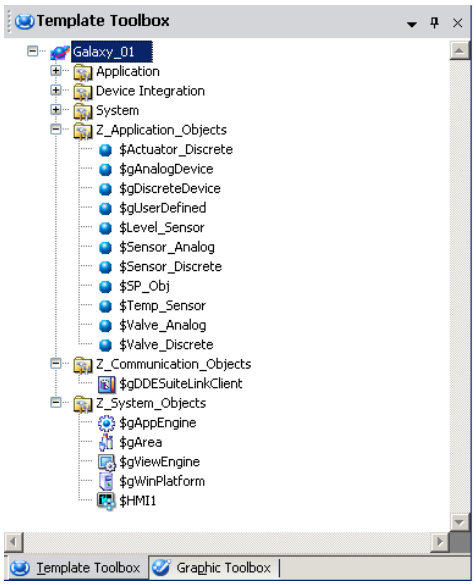


Figura 2.25. Nuevas Plantillas Derivadas

2.5.2. Configuración Plantilla Objetos de Aplicación

Se realizan configuraciones específicas, teniendo en cuenta la naturaleza de cada componente que se creará posteriormente. Estas configuraciones se aplican a tres Plantillas principales, las cuales se identifican por un símbolo y una letra ($g) antes de su nombre.

2.5.2.1. Dispositivo Analógico

Como primera Plantilla principal, tenemos a al dispositivo analógicos denominado *$gAnalogDevice*. Esta Plantilla representa la lectura y escritura de señales analógicas provenientes de campo. Se utiliza para representar tanto los sensores con variables analógicas como las válvulas con control analógico.

Sensor Analógico (*$Sensor_Analog*)

Se utiliza para los sensores de nivel y temperatura. Tienen habilitado la configuración de alarmas en caso de que el valor esté fuera del rango de funcionamiento normal, como se muestra en la Tabla 2.26.

Tabla 2.26. Alarmas Sensores Analógicos

Nombre Alarma	Sensor	Activación
HiHi	Temperatura	Mayor a 95.0 °C
	Nivel	Mayor a 95.0 %
LoLo	Temperatura	Menor a 5.0 °C
	Nivel	Menor a 5.0 %

Válvula Analógica (*$Valve_Analog*)

Se utiliza para representar válvulas con control analógico. En esta clasificación, el valor de comando de posición es un valor porcentual que se aplica a la posición del vástago. Estas válvulas también tienen dos componentes de escritura discreta, que indican al usuario si la válvula está completamente abierta o cerrada, como se muestra en la Figura 2.26.

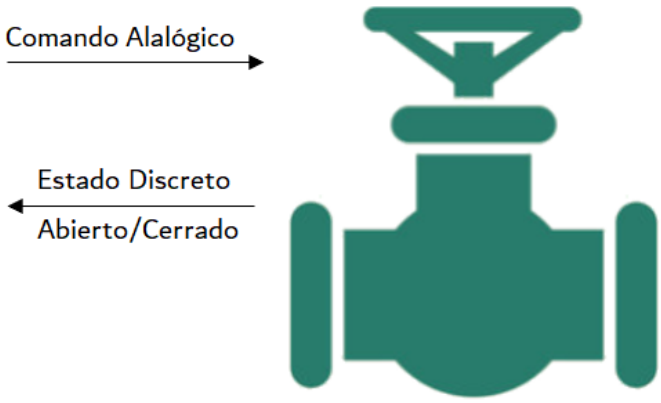


Figura 2.26. Válvula Analógica

2.5.2.2. Dispositivo Discreto

La Plantilla denominada como *$gDiscreteDevice* se clasifica en tres Plantillas derivadas y tiene como característica principal el manejo de entradas y/o salidas con una combinación de señales discretas.

Actuador Discreto

También llamado *$Actuator_Discrete*, tiene las configuraciones de las instancias de los actuadores con un comando binario. Se clasifican para activarse cuando la señal está en estado VERDADERO, como se muestra en la Figura 2.27.

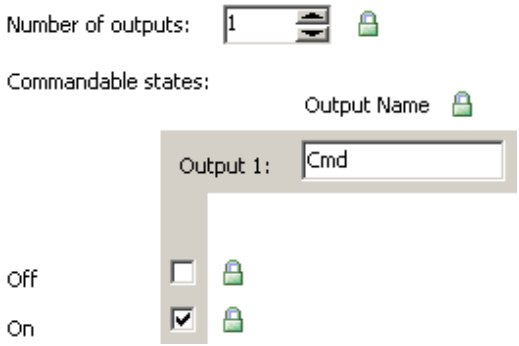


Figura 2.27. Configuración de salidas del *$Actuator_Discrete*

Otra característica configurada en esta Plantilla es la detección del estado de encendido. Es decir, cuando el valor de estado coincide con el valor de comando, como se observa en la Figura 2.28.

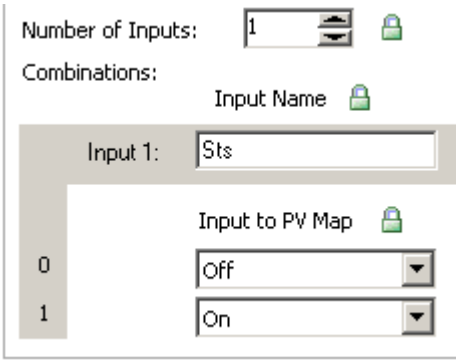


Figura 2.28. Configuración de entradas del *$Actuator_Discrete*

Las entradas y salidas de este Objeto se pueden comprender mejor visualizando la Figura 2.29.

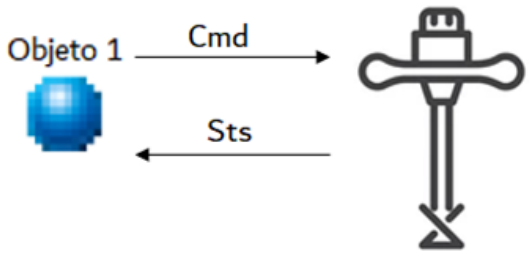


Figura 2.29. Actuador Discreto

Sensor Discreto

En esta Plantilla, solo se activan las variables de entrada, ya que los sensores utilizados no tienen variables de escritura. La configuración de la tabla de entradas se basa en una única variable binaria de estado, donde "No detectado" se representa como cero y "Detectado" como uno, como se muestra en la Figura 2.30.

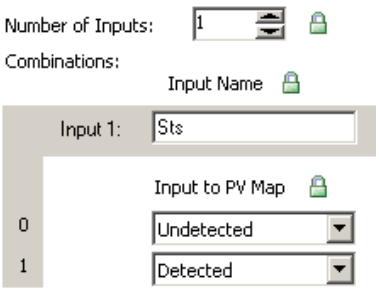


Figura 2.30. Configuraciones de entrada *$Sensor_Discrete*

Válvula Discreta

Esta última Plantilla, basada en dispositivos discretos, se denomina *$Valve_Discrete*. En la sección de salidas (escritura), se tienen dos variables de comando discretas que activan las salidas de "Abierto" y "Cerrado", como se muestra en la Figura 2.31.

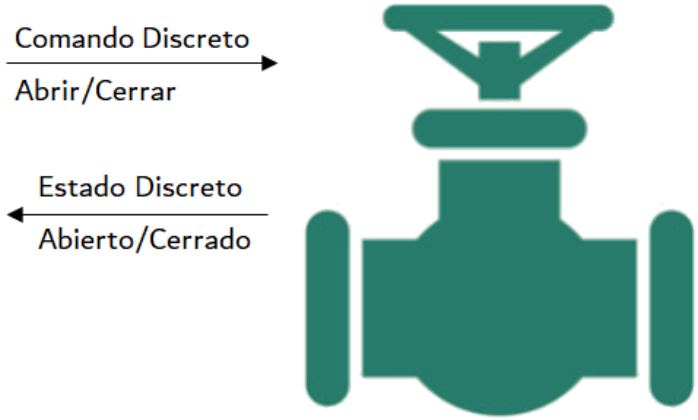


Figura 2.31. Válvula Discreta

2.5.2.3. Definido por el Usuario

Para otros elementos presentes en el proceso, también se configura la Plantilla *$gUserDefined*. Se utiliza para ubicar las variables de consigna de nivel y temperatura como Objetos, como se muestra en la Figura 2.32.

Figura 2.32. Consigna como Objeto

2.5.3. Estaciones de la Plataforma

Cada estación se representa mediante una instancia de la Plantilla *$gWinPlatform*, y se distribuyen de acuerdo a lo mostrado en la Tabla 2.27. Dado que cada objeto representa una estación diferente, es necesario asignar una dirección IP correspondiente a cada una de ellas.

Tabla 2.27. Instancias de las Estaciones

Estaciones	Objeto *$WinPlatform*	Dirección IP
Estación de Ingeniería	*GRPlatform*	192.168.10.140
Estación Servidor de Datos	*Platform_DataServer*	192.168.10.141
Estación de Operación	*Platform_Operation*	192.168.10.142
Estación Historiador	*Platform_Historian*	192.168.10.143

2.5.4. Estación de Ingeniería

Comenzamos con la configuración de la Estación de Ingeniería, también conocida como *GRPlatform*. Siendo la primera instancia creada, se representa con un ícono de color amarillo distinto, ya que esta estación representa la PC de configuración y debe adoptar las configuraciones específicas de la máquina en la que se realizará el almacenamiento del GR asociado al *Galaxy*.

Dado que la planta consta de dos secciones físicamente separadas, procedemos a crear dos instancias relacionadas con cada una de ellas, correspondientes a los subprocesos. Además, se creará una instancia adicional que contendrá los componentes lógicos del sistema de control, como se muestra en la Figura 2.33.

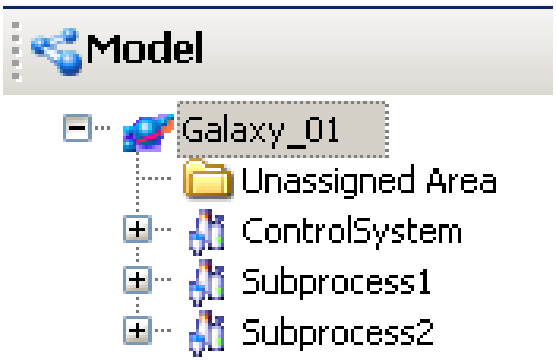


Figura 2.33. Objetos Área – Vista Modelo

Como es de esperar, cada uno de los elementos individuales está contenido en su respectivo subproceso, como se muestra en la Tabla 2.28.

Tabla 2.28. Objetos por Subproceso

Objeto de Aplicación	
Subproceso 1	**Subproceso 1**
ZV-101A	ZV-201A
ZV-101B	ZV-201B
ZV-102A	YV-200A
ZV-102B	YV-200B
ZV-103	EY202
PT-104	LT200
LT-101	TT202
LT102	LT202
PT-103	SP Nivel Tanque 1
EY-104	SP Temperatura Tanque 2
SP Nivel Cacao	

Objeto de Aplicación

En esta sección se detalla la configuración de los Objetos de Aplicación creados como Instancias dentro de los subprocesos. En este punto, los elementos individuales se muestran en la Vista de Implementación con el propósito de direccionar cada una de las señales o atributos hacia los elementos presentes en el campo.

A modo de ejemplo, se muestra el Objeto que representa la válvula analógica ubicada en el subproceso 2 (YV-200B, *Intel_Tank1*). En sus configuraciones, se especifica el directorio de la variable de comando y del estado analógico, como se muestra en la Figura 2.34.

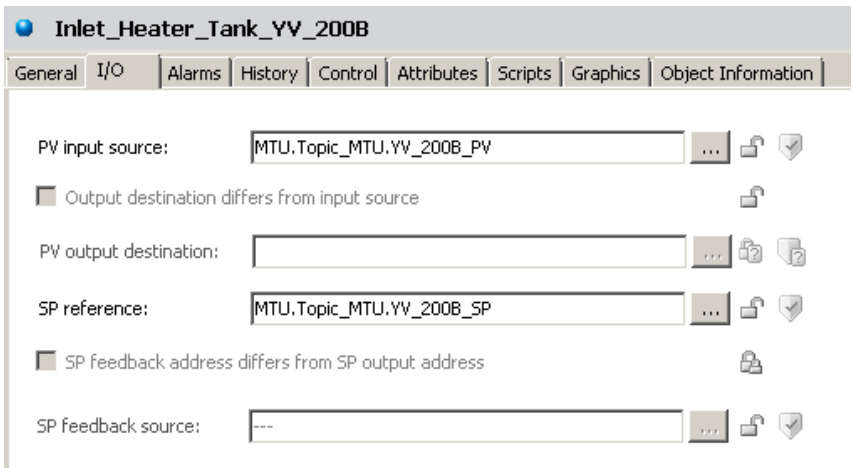


Figura 2.34. Direccionamiento Objeto de Aplicación

El detalle de la dirección mostrada se explicará más adelante.

2.5.5. Servidor de Datos

Para garantizar el adecuado manejo del servidor de aplicación, se ha implementado una estación adicional dedicada a la obtención y envío de datos. Este servidor se encarga de configurar los diferentes *DAServers* que se comunican con los controladores en planta.

El dispositivo encargado de proporcionar las señales a la Plataforma es un único MTU. Este módulo centralizador de datos utiliza un *DAServer* específico según el fabricante y modelo.

2.5.5.1. Configuración en el *DAServer Manager*

Dado que el servidor de datos está conectado en red, las configuraciones de los *DAServers* se pueden realizar utilizando cualquier PC con acceso al *System Management Console*.

Dentro del administrador, se pueden observar los diferentes *DAServers* disponibles para su configuración, como se muestra en la Figura 2.35.

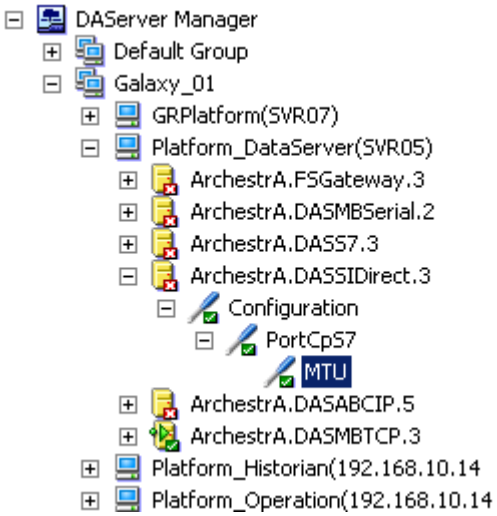


Figura 2.35. *DAServer Manager*

El servidor específico para el intercambio de datos con el MTU es el *SIDirect*, cuyo nombre es *ArchestrA.DASSIDirect.3.* En este servidor se deben configurar los diferentes parámetros requeridos.

Inicialmente, se agrega un puerto *PortCpS7*, el cual representa el tablero de red de la computadora que se comunica con el PLC. Para agregarlo, se hace clic derecho sobre la rama de Configuración dentro de la jerarquía del *DAServer*, posteriormente se selecciona "*Add PortCpS7*" y se crea el objeto. Este puerto utiliza los protocolos TCP/IP.

La conexión S7 tiene los siguientes tres parámetros de configuración, que se muestran en la Tabla 2.29:

- Dirección de Red: Es la dirección IP o el nombre del host del PLC remoto.
- TSAP Local: Se configura el número de recursos de conexión de la computadora.
- TSAP Remoto: Corresponde a cómo se configuró el TSAP para el PLC, que incluye el Rack y el Slot remotos.

Tabla 2.29. Parámetros S7Cp (MTU)

Parámetros			
Dirección de Red	192.168.10.3		
TSAP Local	01 - 00		
TSAP Remoto	Rack No. 0	Slot No. 1	Recurso de conexión: 1

Para el envío de datos hacia el PLC centralizador, se procede a crear un Tópico que sirve como punto de inicio para el direccionamiento de los datos. Se le asigna el nombre de "*Topic_MTU*". Para establecer una relación entre las variables de este Tópico, se ingresan cada uno de los tags provenientes del MTU con su nombre y su respectiva dirección local en la memoria del PLC. Esto proporciona una ruta hacia las diferentes variables, como se muestra en la Tabla 2.30.

Tabla 2.30. Parámetros S7Cp (MTU)

Tipo de Dato	Dirección Local S7	Referencia *DAServer*
Bool	*%M0.3*	*MX0.3*
Byte	*%MB1*	*MB1*
Word	*%MW2*	*MW2*
DWord	*%MD4*	*MD4*
INT	*%MW6*	*MINT6*
DINT	*%MD10*	*MDINT10*
REAL	*%MD14*	*MREAL14*

En la Figura 2.36 se puede observar un ejemplo del direccionamiento desde la variable "*Cacao_Milk_SP_Level*" con dirección local *%MD150*, configurada en el SMC como *MREAL14*.

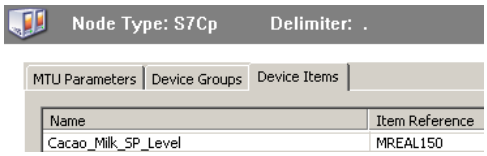


Figura 2.36. Configuración de una señal en el SMC

2.5.5.2. Configuración Objeto de Comunicación

Por otra parte, para acceder a los datos, se procede a crear un Objeto de Integración de Dispositivos dentro del software de desarrollo *ArchestrA IDE*. Dado que estamos utilizando un *DAServer* como servidor de datos, utilizaremos un protocolo del mismo fabricante, en este caso, el Objeto es un cliente *DDESuiteLink*.

Este Objeto debe estar contenido dentro de un *Engine*, por lo tanto, se procede a crear un nuevo *AppEngine* que se encuentra dentro de la Estación de Servidor de Datos, como se muestra en la Figura 2.37.

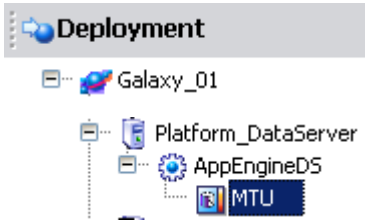


Figura 2.37. Cliente *SuiteLink* en la Estación de Servidor de Datos

Inicialmente, se debe configurar el Objeto llamado MTU con las configuraciones de red y el nombre del *DAServer* utilizado, como se indica en la Figura 2.38.

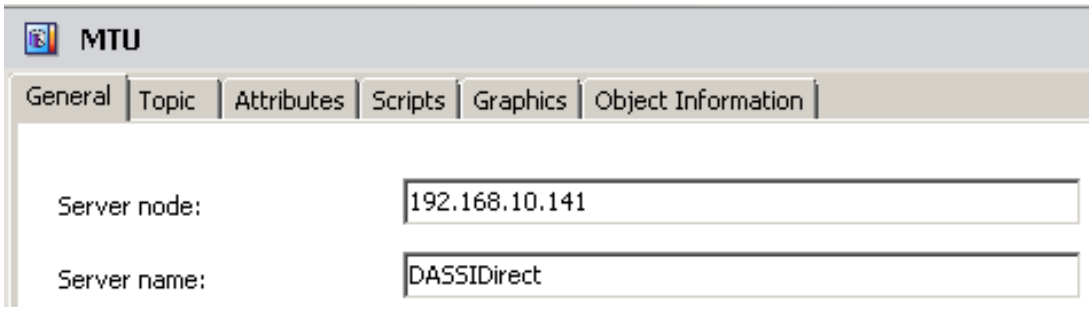


Figura 2.38. Configuración general del cliente *SuiteLink* (MTU)

Esta configuración permitirá establecer la conexión con el *DAServer* presente en la máquina con la dirección mostrada. Una vez establecida la conexión, se procede a especificar el nombre del Tópico creado y las referencias a cada una de las variables, completando así la ruta de las variables desde la fuente de datos (PLC) hasta el Objeto de Comunicación.

En resumen, la dirección creada para cada variable se compone de los siguientes componentes separados por puntos, como se muestra en la Figura 2.39:

- Nombre del Objeto de Comunicación.
- Nombre del Tópico.
- Nombre del Atributo asociado a dicho Tópico.

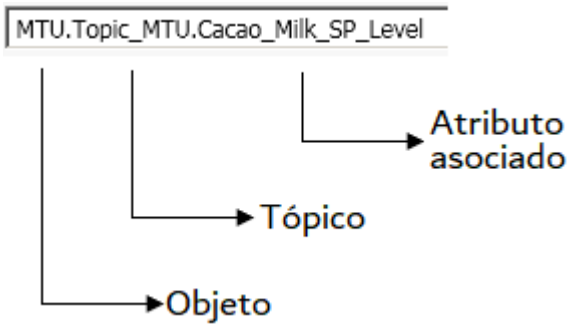


Figura 2.39. Ejemplo dirección al atributo

Esta dirección se debe ingresar en cada Objeto de Aplicación que requiera acceder a una variable específica. Para facilitar esta tarea, se puede utilizar el navegador del *Galaxy*, como se detalla en la Figura 2.40.

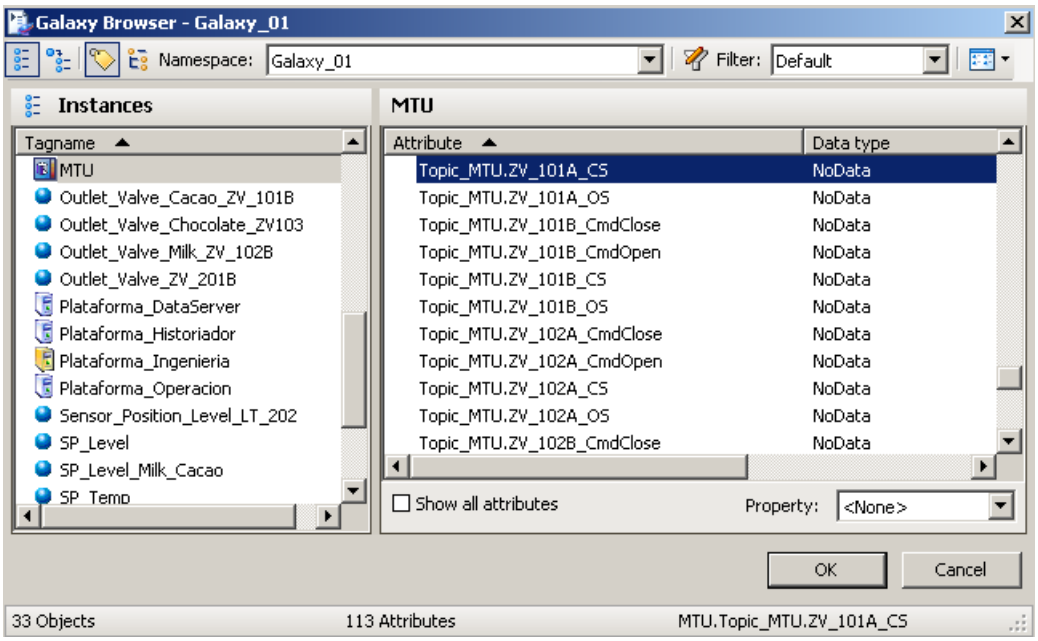


Figura 2.40. Galaxy Browser

2.5.6. Implementación HMI

Dentro del sistema de control y monitorización, los operadores desempeñan un papel clave en el desarrollo del proceso. Por esta razón, se lleva a cabo la implementación de la estación correspondiente de operación, la cual se encarga de facilitar la interacción entre los datos de la planta y el personal.

2.5.6.1. Arquitectura del HMI

La interfaz con el usuario está compuesta por diversos procesos y componentes de automatización que deben estar organizados de manera estructurada. El esquema propuesto clasifica cada sección según su función. A continuación, se muestra un diagrama que ilustra dicho esquema, junto con las ventanas y la navegación entre ellas, tal como se presenta en la Figura 2.41.

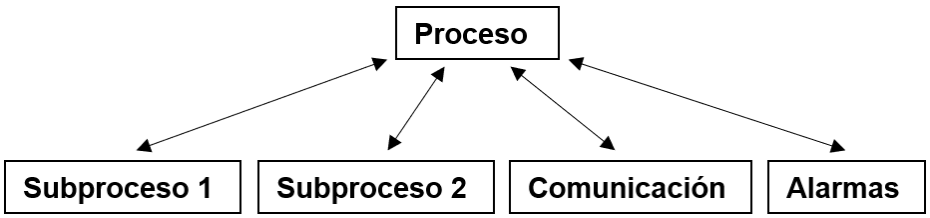


Figura 2.41. Esquema HMI

2.5.6.2. Implementación del Objeto

El software *InTouch* tiene la capacidad de funcionar de forma independiente, estableciendo una conexión directa de red con los servidores. En este trabajo de titulación, se adopta un enfoque basado en el servidor de aplicación.

El Objeto *$InTouchViewApp* representa el software *InTouch* dentro del servidor de aplicación y, como tal, administra su ejecución en diferentes ubicaciones, como se muestra en la Figura 2.42.

Figura 2.42. Objeto *$InTouchViewApp*

Una característica particular de *$InTouchViewApp* es que todas las configuraciones se realizan en su Plantilla, lo que significa que la instancia creada no tiene ninguna configuración habilitada. Por lo tanto, se crea una Plantilla derivada llamada *$HMI* y se accede al *WindowMaker*, que está integrado en el software.

2.5.6.3. Desarrollo en *WindowMaker*

Al acceder a las configuraciones, se observa una interfaz que es muy similar a la del software *InTouch*, (ver Figura 2.43).

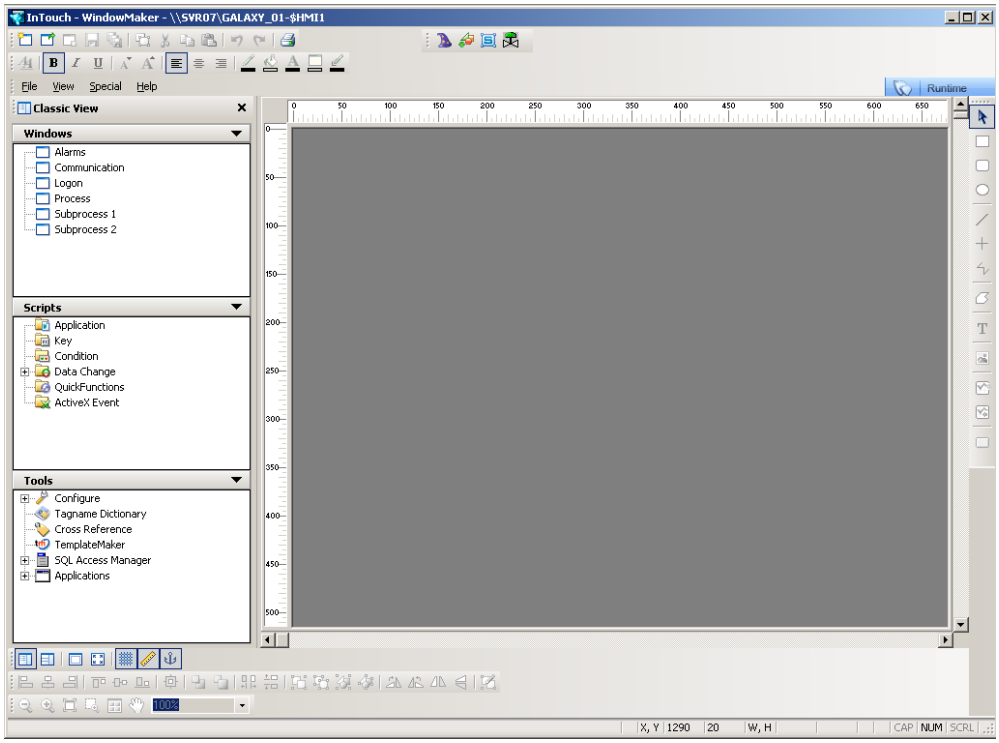


Figura 2.43. *InTouch – WindowMaker*

Configuraciones de Seguridad

Las configuraciones de seguridad del servidor de aplicación se detallarán más adelante. Por ahora, se habilita dicho sistema en *InTouch* seleccionando el tipo de seguridad implementado en el menú de herramientas, como se muestra en la Figura 2.44.

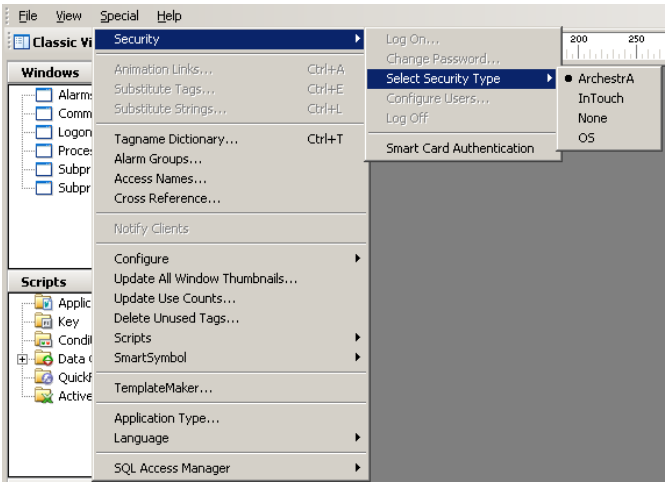


Figura 2.44. Tipo de seguridad en *InTouch*

Diseño de las Ventanas

Las ventanas implementadas siguen un estándar común en cuanto a su visualización. Para mayor comodidad y ergonomía de los operadores, tienen una relación de aspecto 4:3 y un fondo de pantalla suave para evitar la fatiga visual, siguiendo los criterios establecidos en los estándares. Las propiedades de la ventana se detallan en la Figura 2.45.

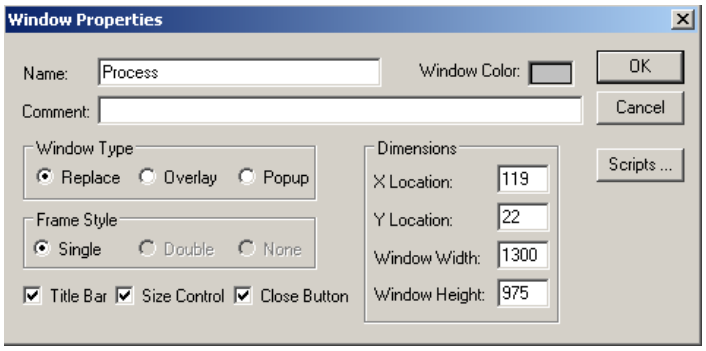


Figura 2.45. Propiedades Ventana

Para la navegación, se utiliza un componente pulsador que tiene la función de abrir una ventana. Estos pulsadores se encuentran en la parte izquierda, en la sección del menú de navegación. La ventana que se encuentra activa muestra texto en lugar de un pulsador para mantener el orden en la navegación vertical, como se muestra en la Figura 2.46.

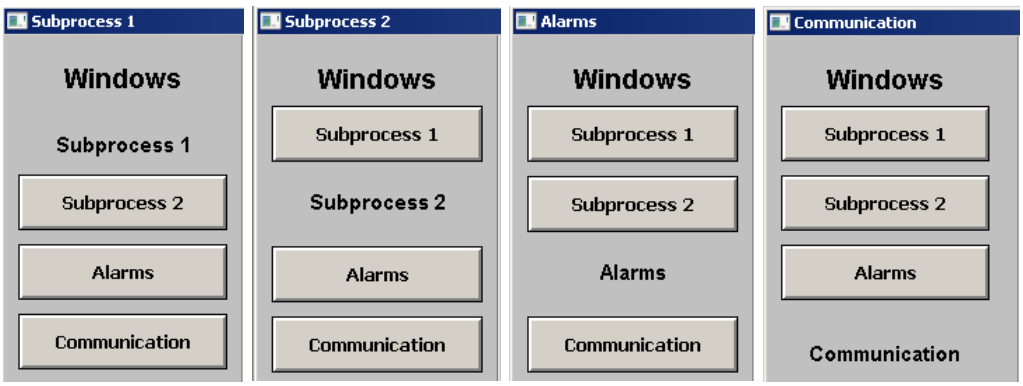


Figura 2.46. Navegación Horizontal Segundo Nivel

En la parte inferior, hay una sección utilizada para el posicionamiento de diversos componentes. El primer elemento permite abrir la pantalla principal (Proceso), el

segundo elemento es un pulsador programado para cerrar la sesión del usuario que esté utilizando la interfaz, y por último, se muestra un reloj y calendario digital, como se aprecia en la Figura 2.47.

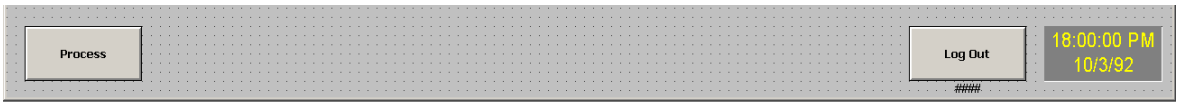


Figura 2.47. Menú inferior

Las ventanas del segundo nivel se muestran en las Figuras 2.48 a 2.51. Para obtener información detallada sobre el manejo de cada una de estas ventanas del HMI, consulte el Anexo C.

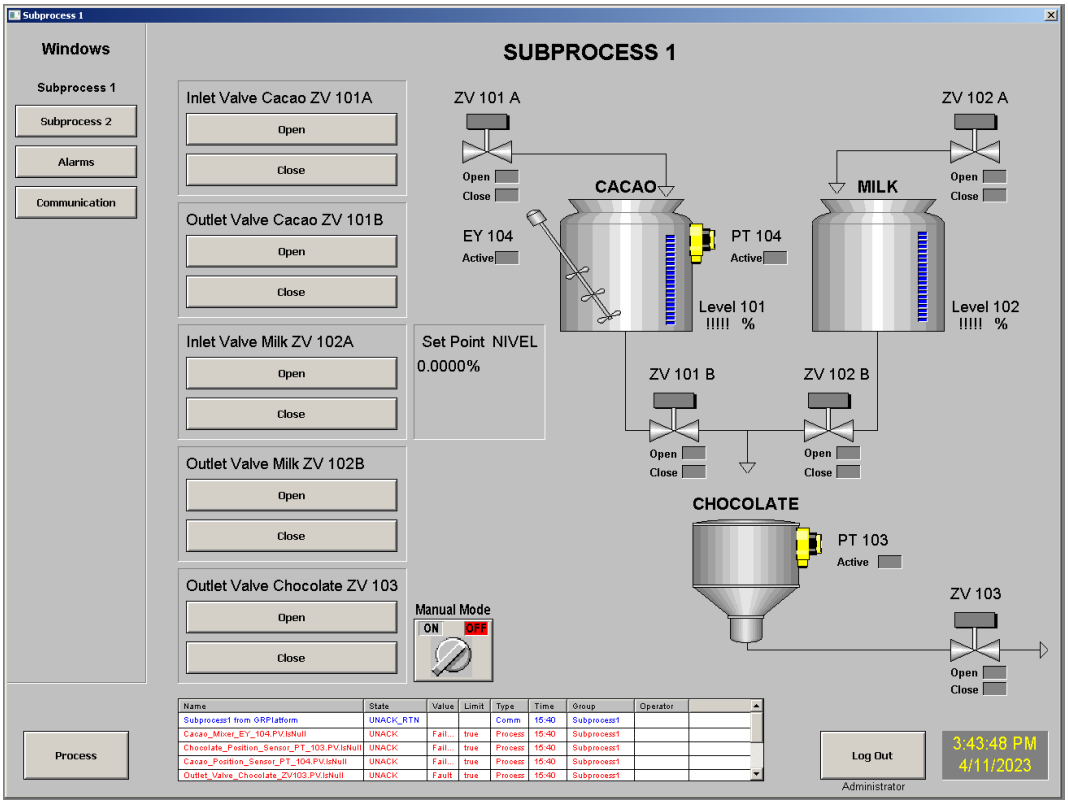


Figura 2.48. Subproceso 1

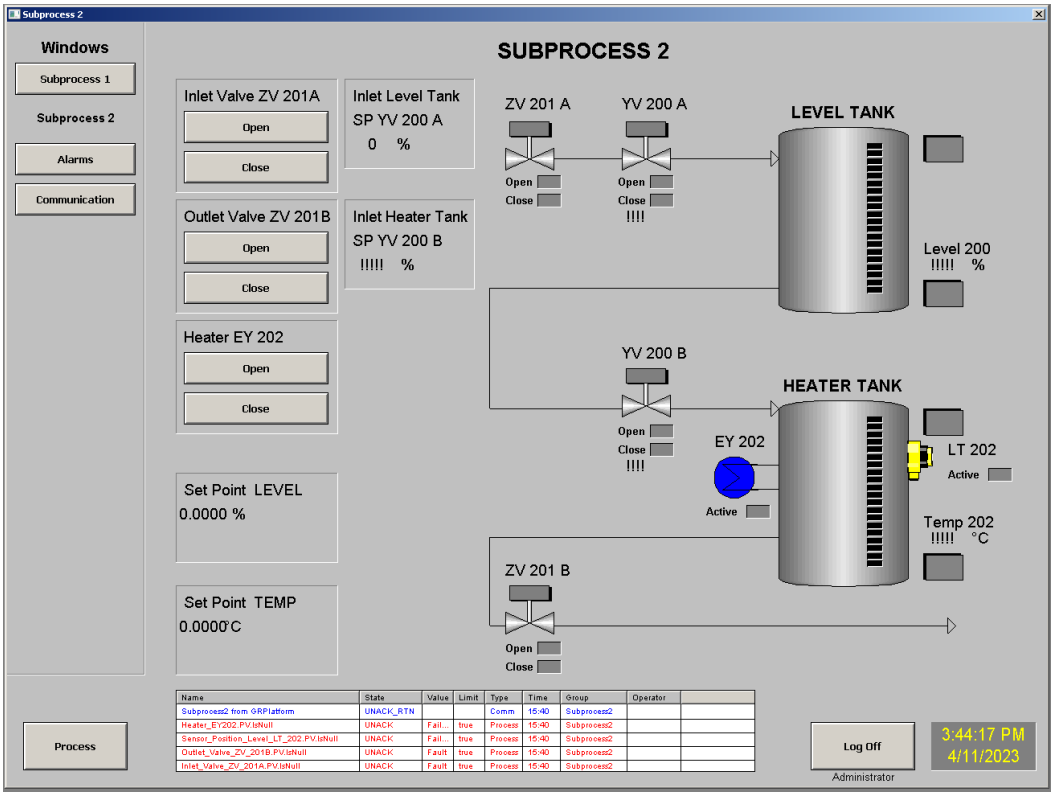


Figura 2.49. Subproceso 2

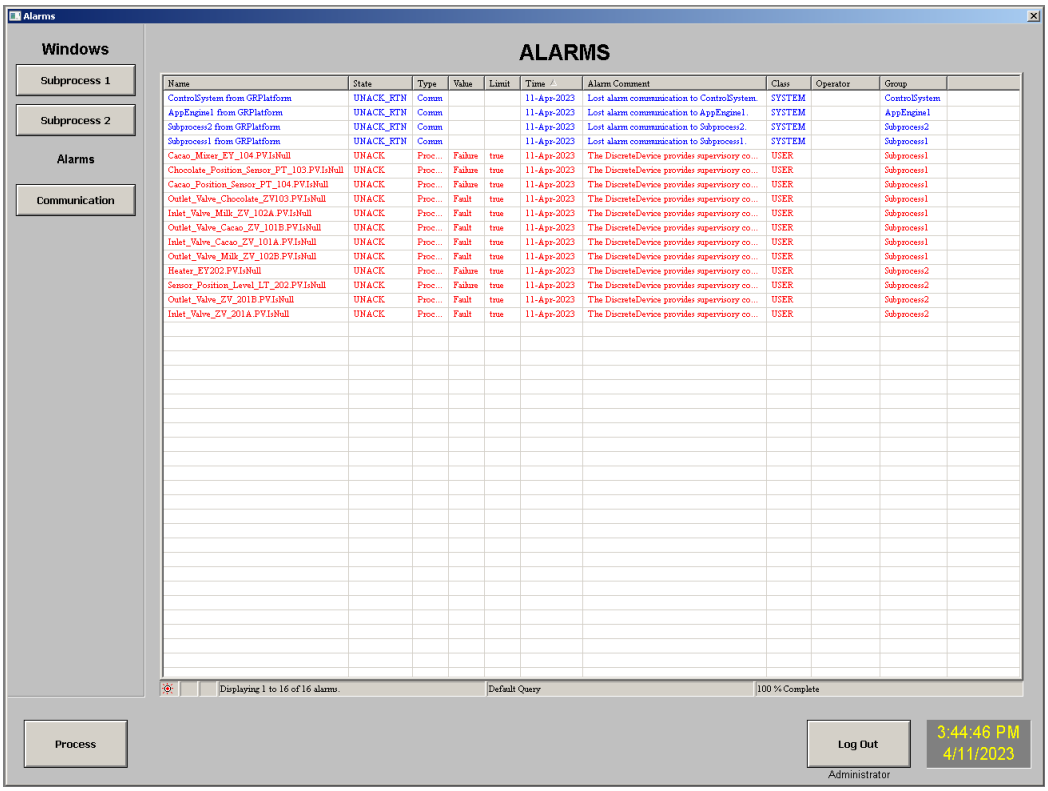


Figura 2.50. Alarmas

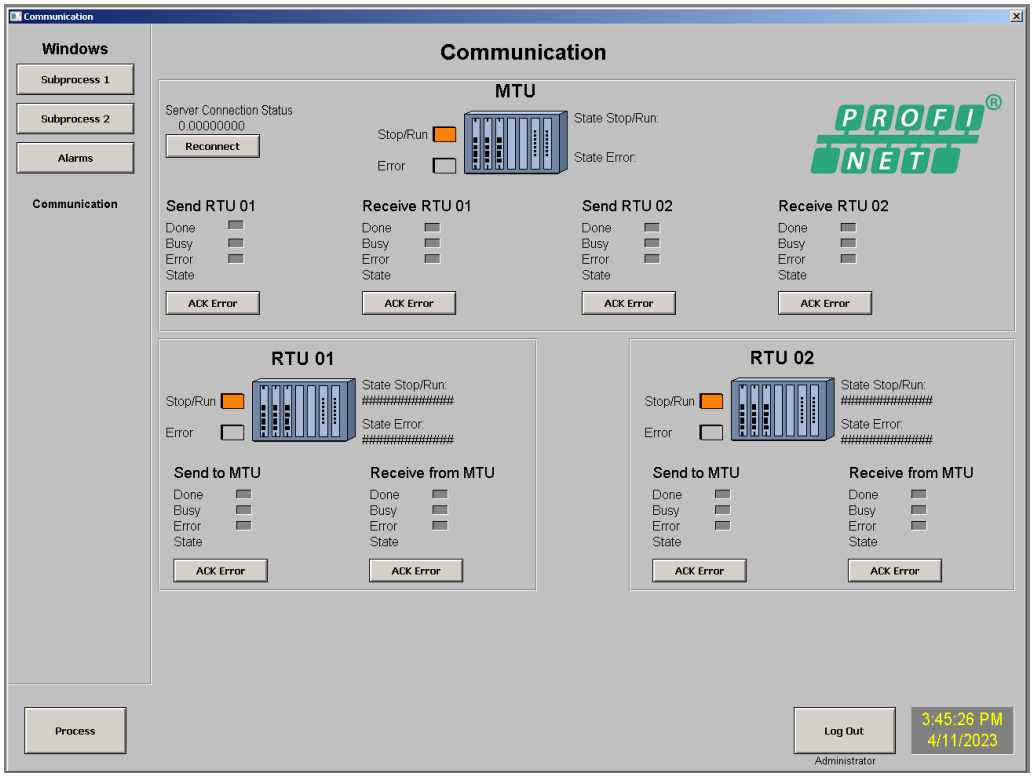


Figura 2.51. Comunicación

2.5.6.4. Relación con Objetos de Aplicación

Para vincular cada componente y elemento gráfico con sus respectivas variables, ya sea para escritura (por ejemplo, consigna) o lectura (por ejemplo, variables de proceso o estado), se realiza esta asociación con cada Objeto de Aplicación al que pertenecen.

Manipulación Atributos

A modo de ejemplo, se tomará la válvula de ingreso de la materia prima de cacao para explicar cómo funciona esta asociación del componente con la variable. Este dispositivo está representado por el Objeto de Aplicación "*Inlet_Valve_Cacao_ZV101A*". Los componentes y elementos gráficos se relacionan directamente con los atributos del objeto. Este objeto interpreta y procesa la información, transmitiéndola al servidor de datos a través de sus variables I/O, como se muestra en la Figura 2.52.

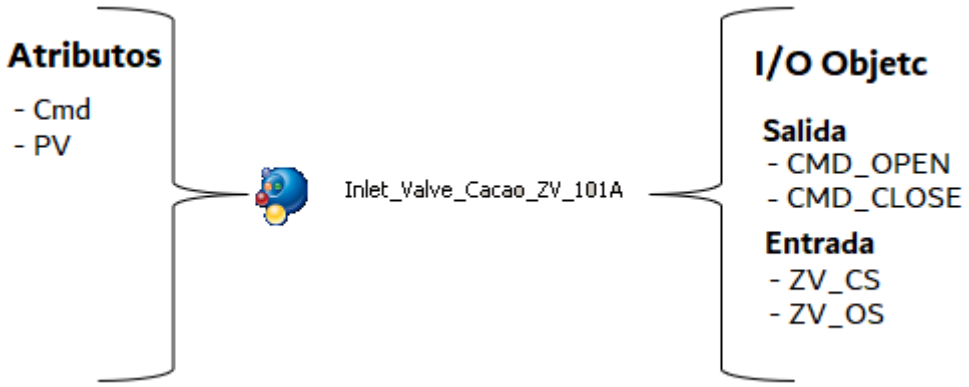


Figura 2.52. Objeto ZV_101A

Para escribir en una variable (comando sobre el actuador), se utiliza un botón de apertura que cambia el valor del atributo "Cmd" a "*Opened*" mediante un script. Del mismo modo, al presionar el botón de cierre, este atributo cambia a "*Closed*", como se observa en la Figura 2.53.

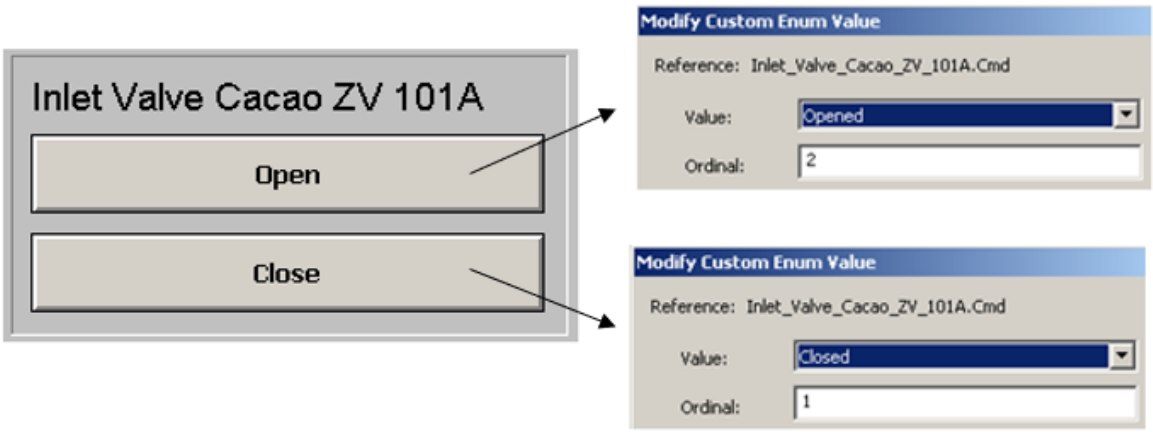


Figura 2.53. Cambio en el atributo *Cmd*

Por otro lado, para la lectura de las variables, se sigue un proceso inverso. Los componentes de visualización leen las características I/O del Objeto y las representan gráficamente, como se aprecia en la Figura 2.54.

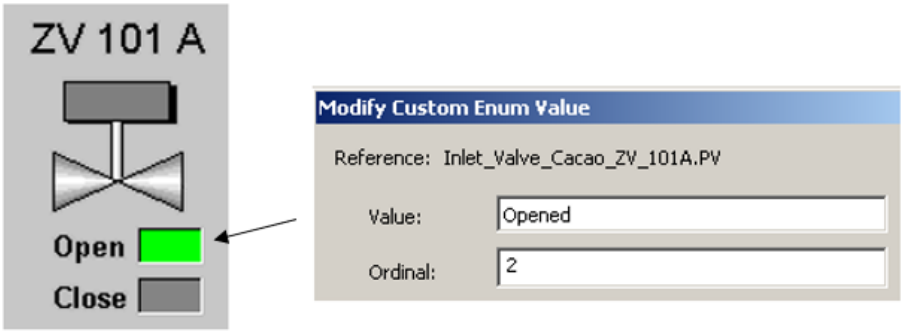


Figura 2.54. Lectura PV

2.5.7. Alarmas

A continuación, procedemos a la implementación de las alarmas necesarias en los Objetos de Automatización.

2.5.7.1. Descripción de Alarmas en la Plataforma

Proveedores de Alarmas

Al utilizar el servidor de aplicación de *ArchestrA*, el Objeto *WinPlatform* es uno de los componentes que actúa como proveedor de alarmas. Se debe habilitar la configuración en dicho Objeto, el cual suministrará a los Clientes las alarmas presentes dentro de los Objetos contenidos, como se muestra en la Figura 2.55.

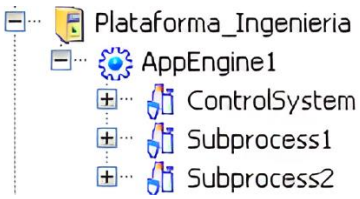


Figura 2.55. Plataforma como Proveedor de Alarmas

Atributos Generales

Cada alarma tiene varios atributos manejados de forma independiente, que se presentan a continuación:

- ***AlarmMode:*** Indica el estado actual del modo de la alarma del Objeto y si la alarma está habilitada o deshabilitada. El atributo en sí no permite un cambio de modo, para esto se utiliza el comando *AlarmModeCmd*.

- ***AlarmInhibit***: Otro atributo binario que, cuando está activado, deshabilita las alarmas. Este atributo cambia el *AlarmMode* al estar activado.
- ***InAlarm***: Se activa cuando alguna de las alarmas del Objeto se encuentra en estado activo.

Algunas de las configuraciones a realizar en las alarmas para modificar sus atributos se presentan a continuación:

- ***<Atributo>.Acked***: Se utiliza para especificar si la alarma ha sido reconocida. Se actualiza cuando se configura el atributo AckMsg. Es FALSO cuando se detecta una nueva alarma (cambio de *InAlarm* de FALSO a VERDADERO)
- ***<Atributo>.AckMsg***: Es el comentario del operador al reconocer la alarma, y se guarda como texto. El cambio en este atributo configura Acked como VERDADERO.
- ***<Atributo>.AlarmMode***: Muestra la configuración de la alarma en los valores: Habilitado, Deshabilitado o Silenciado.
- ***<Atributo>.AlarmModeCmd***: Configura la alarma en los valores: Habilitado, Deshabilitado o Silenciado.
- ***<Atributo>.InAlarm***: Muestra el estado de la alarma.

Las alarmas pueden estar habilitadas, deshabilitadas o silenciadas durante el tiempo de ejecución.

- **Habilitada**: Las alarmas de un objeto se reportan a las aplicaciones Cliente y se guardan en un histórico.
- **Deshabilitada**: Las alarmas de Objeto no se detectan y se regresan a un estado normal hasta que se habilite la alarma nuevamente.
- **Silenciada:** Las alarmas de un objeto se detectan, se guardan en un histórico y se envían a las aplicaciones Cliente, pero los Clientes no muestran dichas alarmas.

2.5.7.2. Configuración de la Plataforma

Para obtener las alarmas, se debe configurar inicialmente el proveedor de las mismas. El Objeto *WinPlatform* puede realizar estas funciones de suministro de alarmas. Por lo tanto, se inicia habilitando dicha característica en las configuraciones del objeto, como se muestra en la Figura 2.56.

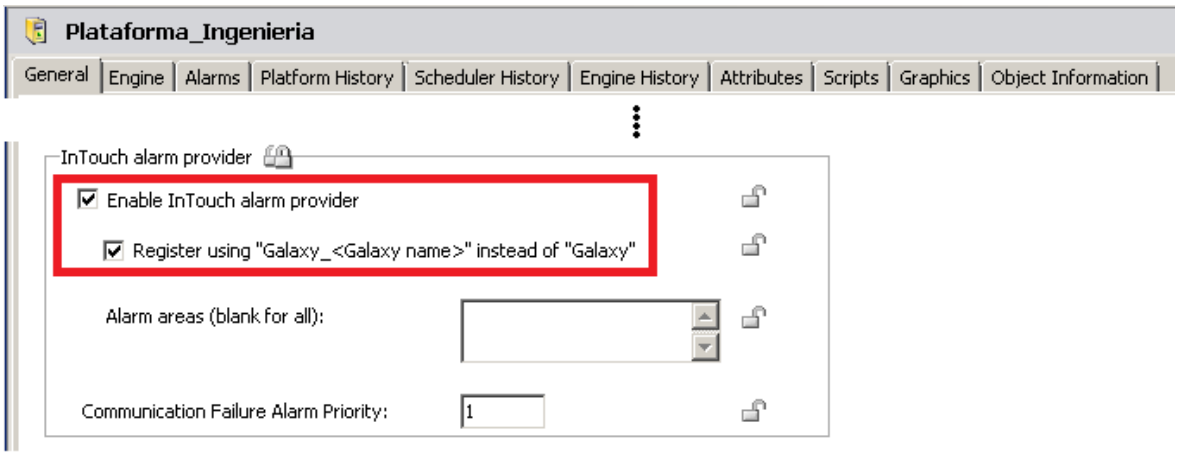


Figura 2.56. Habilitar *Enable InTouch alarm provider*

Al tener esta opción habilitada, la plataforma proveerá las alarmas de los objetos contenidos dentro las Áreas (*ControlSystem*, *Subprocess1*, *Subprocess2*).

2.5.7.3. Configuración de los Objetos de Aplicación

Los objetos que tengan valores críticos y necesiten mostrar alarmas serán configurados como tales. Como muchas de las configuraciones, las alarmas se configurarán en las Plantillas:

$Level_Sensor

Los sensores analógicos de nivel están basados en esta Plantilla, y la unidad de ingeniería es el porcentaje. Se procede a la configuración de las alarmas correspondientes. La primera es la habilitación de las alarmas de nivel, y se requiere una alerta cuando el componente transmita un valor mayor al 95% o un valor menor al 5%. Dentro de las configuraciones de dicha Plantilla, se habilita esta alarma y se asignan los valores correspondientes. Estas alarmas de nivel tienen

una severidad crítica y un valor de prioridad de 100, como se muestra en la Figura 2.57.

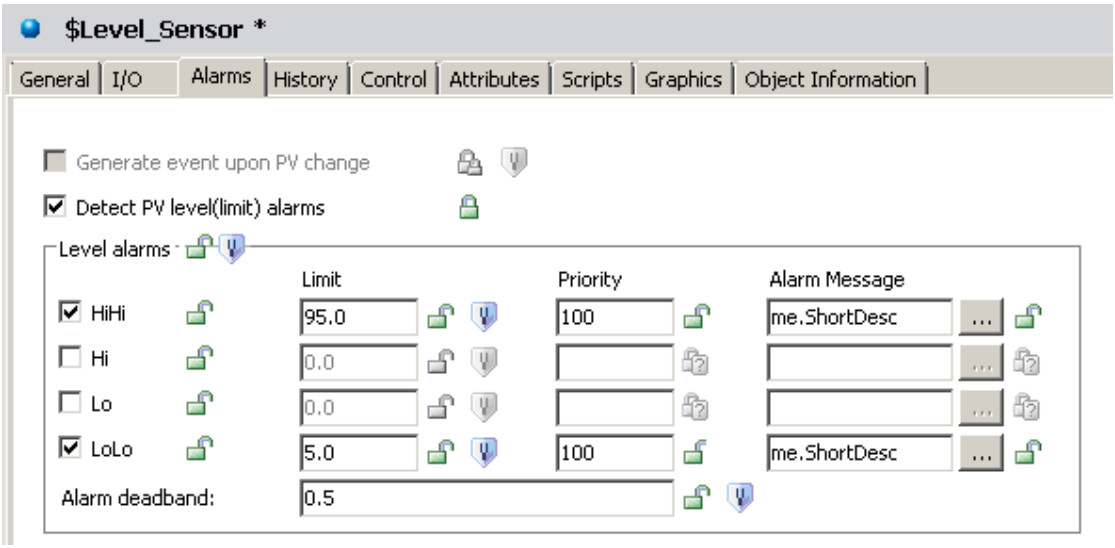


Figura 2.57. Alarma *$Level_Sensor*

$Temp_Sensor

Los objetos que representan los sensores de temperatura se basan en dicha Plantilla, y tienen como unidad de medida los grados centígrados. Se procede a activar la alarma de nivel, teniendo en cuenta que se considera un valor crítico cuando se sobrepasa los 90°C, como se muestra en la Figura 2.58.

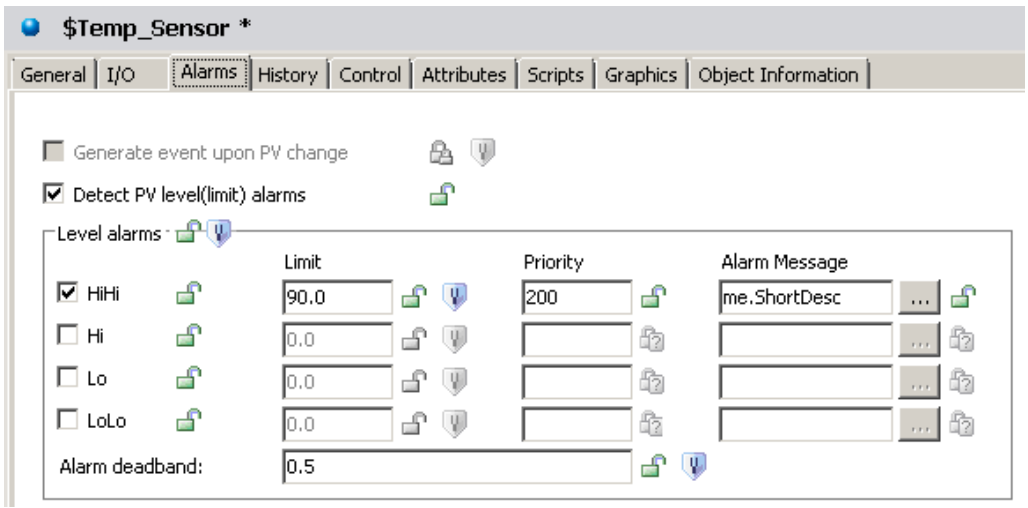


Figura 2.58. Alarma *$Temp_Sensor*

$Actuador_Discrete:

En esta plantilla se ubican los objetos que representan actuadores digitales (*Heater* y *Mixer*). Se procede a habilitar las alertas de detección de cambios no realizados (discrepancia con la consigna) y alertas pasadas un tiempo de espera de 10 segundos, con prioridades de 600 al tener una severidad menor, como se muestra en la Figura 2.59.

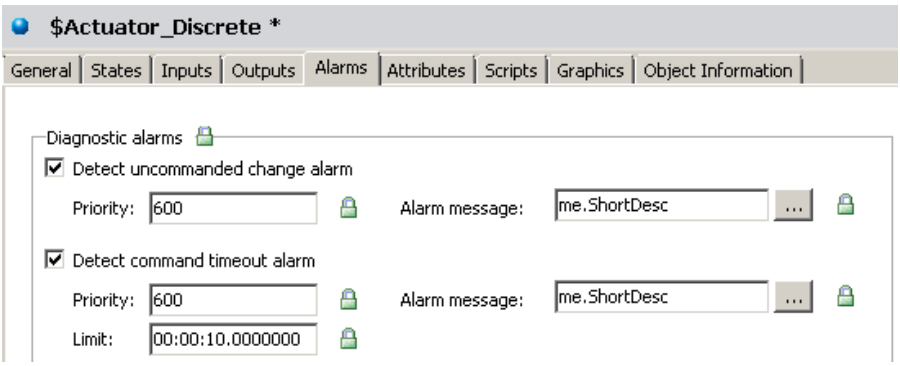


Figura 2.59. Alarma *$Actuador_Discrete*

$Valve_Discrete:

Al igual que la Plantilla anterior, se procede a activar las mismas alarmas para los objetos que representan las válvulas con comando discreto. Al tener un nivel de severidad mayor, se configuran con un valor de prioridad de 300, como se muestra en la Figura 2.60.

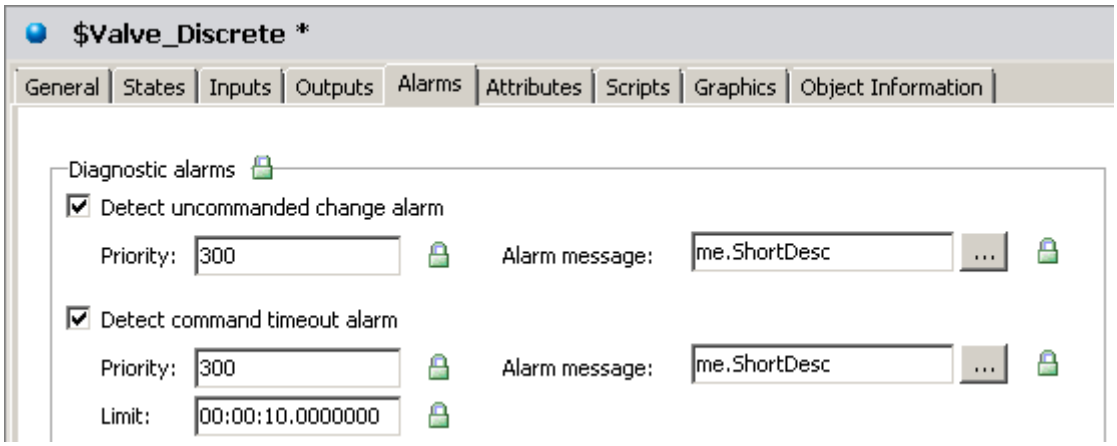


Figura 2.60. Alarma *$Valve_Discrete*

2.5.7.4. Implementación de Alarmas

Una vez que las alarmas y el proveedor están configurados, procedemos a implementar dichos elementos en las ventanas del HMI. Dentro del software *InTouch*, existen dos objetos relacionados con las alarmas que se utilizan en esta aplicación.

Alarm Viewer Control

Este objeto se utiliza para visualizar las alarmas. Cuenta con barras de desplazamiento, columnas modificables, diferentes secciones de alarmas y una barra de estado de actualizaciones.

Las columnas que se utilizan en el visualizador son: nombre, estado, tipo de alarma, valor, límite, fecha, contenido de la alarma, clase, operador y grupo. El *Alarm Viewer Control* se encuentra en la ventana de *Alarms* dentro del HMI y mostrará las alarmas de las dos áreas del proceso. El direccionamiento configurado en el objeto es el siguiente:

\\SVR07\Galaxy_Galaxy_01!Subprocess1

\\SVR07\Galaxy_Galaxy_01!Subprocess2

Ubicando el nodo remoto "*SVR07*", el Galaxy con nombre "*Galaxy_01*" y los nombres de los subprocesos "*Subprocess1*" y "*Subprocess2*", como se muestra en la Figura 2.61.

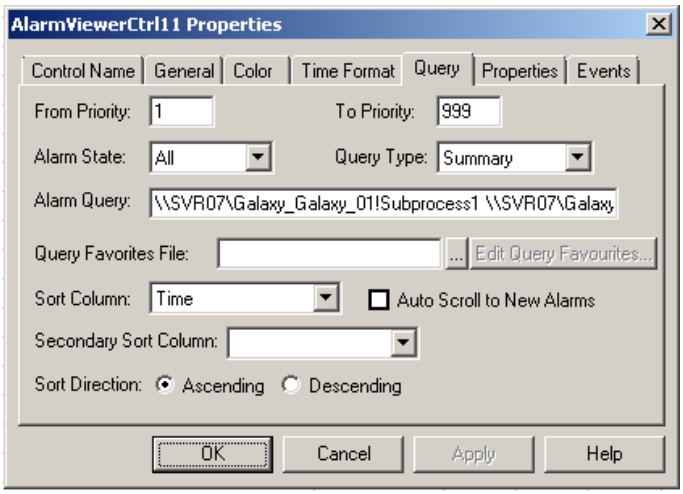


Figura 2.61. Propiedades de *AlarmViewerCtrl*

El objeto mencionado tiene la posibilidad de realizar cambios en las alarmas, como por ejemplo, realizar un reconocimiento, como se muestra en la Figura 2.62.

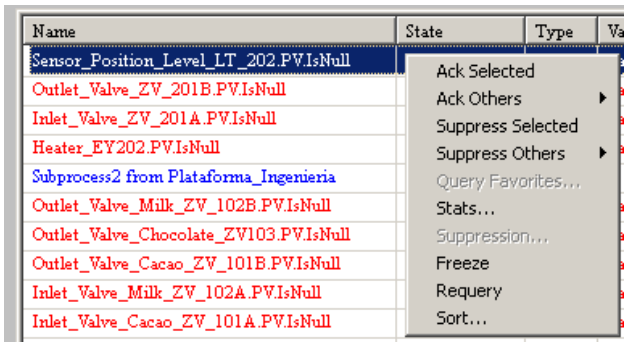


Figura 2.62. Comandos en *AlarmViewerCtrl*

2.5.8. Histórico

A continuación, se detalla la implementación del Histórico utilizado para almacenar los datos del proceso. En esta aplicación, se utiliza una estación remota que se encarga de visualizar y almacenar estos datos.

Los Objetos de Automatización configurados para crear y almacenar los datos se encuentran ubicados dentro de la Plataforma de Configuración (*GRPlatform*). Para esto, se debe acceder a sus configuraciones y crear un directorio en el cual se almacenan los datos *Store Forward*, como se muestra en la Figura 2.63.

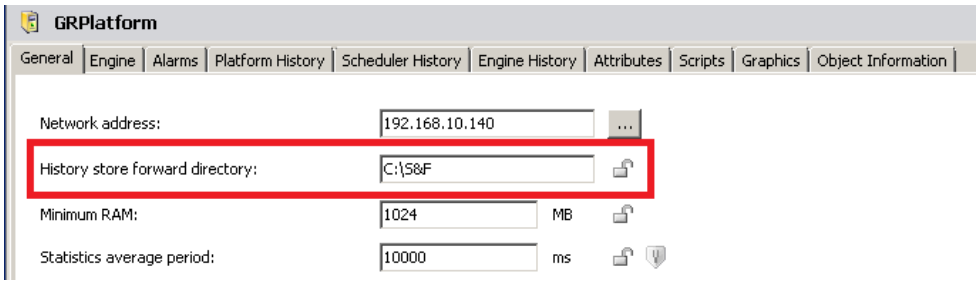


Figura 2.63. Directorio *Store and Forward*

Adicionalmente, se habilita la configuración del *Engine* que contiene los objetos encargados de enviar las señales al Historiador. Se utiliza el direccionamiento del nodo utilizado en la estación Historiador, como se observa en la Figura 2.64.

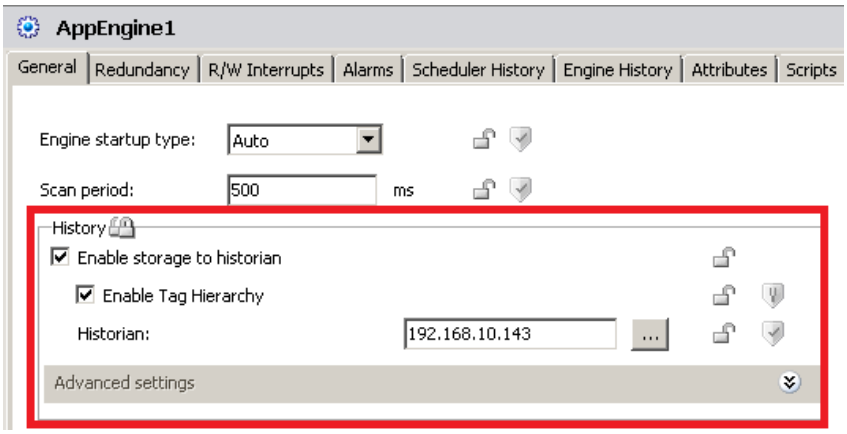


Figura 2.64. Historiador *Engine*

2.5.8.1. Configuración en los Objetos de Aplicación

Se procede a habilitar y configurar las propiedades de historización de los objetos pertinentes. Al igual que muchas de las configuraciones en los Objetos de Automatización, y particularmente en los Objetos de Aplicación, estas configuraciones se realizan en las Plantillas, ya que al ser una herramienta de desarrollo orientada a objetos, se propagan a cada uno de sus derivados.

Sensor Analógico

La Plantilla *$Sensor_Analog* tiene habilitada la entrada PV. Por lo tanto, se procede a marcar dicha configuración para la historización. Se configura para que el dato se envíe cada 20 segundos, con una banda muerta del 0.5% o °C, como se muestra en la Figura 2.65.

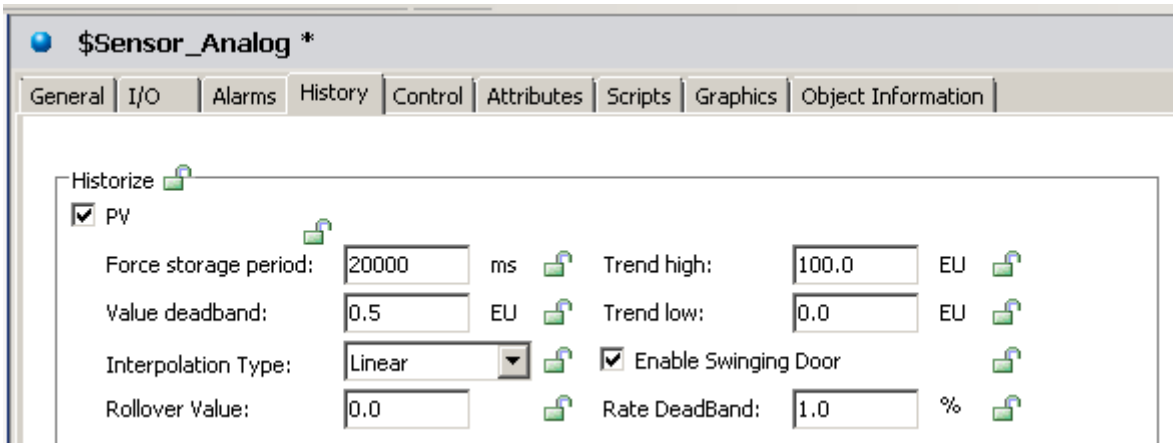


Figura 2.65. Historización *$Sensor_Analog*

Válvula Analógica

En el caso de la Plantilla *$Valve_Analog*, se configura la historización para los dos valores: PV y las señales de estado. Para las configuraciones del PV, también se realiza un almacenamiento cada 20 segundos. Además, se envía al historiador cuando se supera una banda muerta del 1.0%, como se muestra en la Figura 2.66.

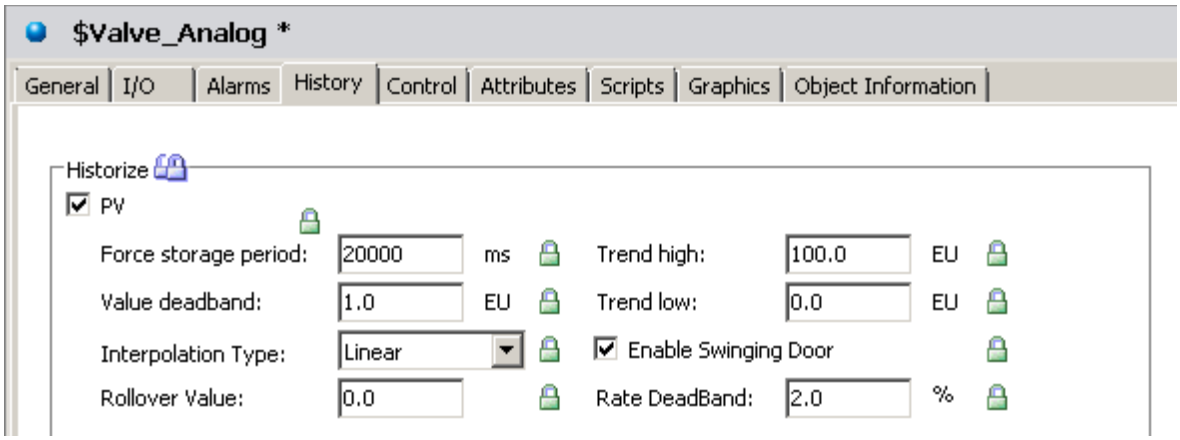


Figura 2.66. Historización *$Valve_Analog "PV"*

Para los atributos de las señales de estado (UDA), se habilita la historización cuando se produce un cambio en la variable discreta, como se observa en la Figura 2.67.

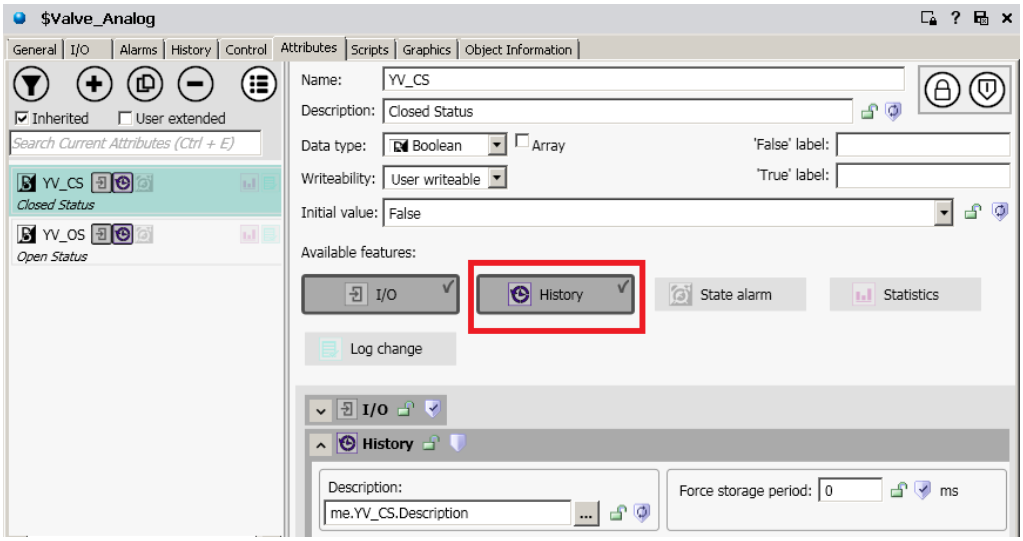


Figura 2.67. Historización *$Valve_Analog "Status"*

Actuador Discreto

En la Plantilla *$Actuator_Discrete* se habilita la configuración de la historización para los cambios de entrada (estado) y salida (comando) discretos, como se aprecia en la Figura 2.68.

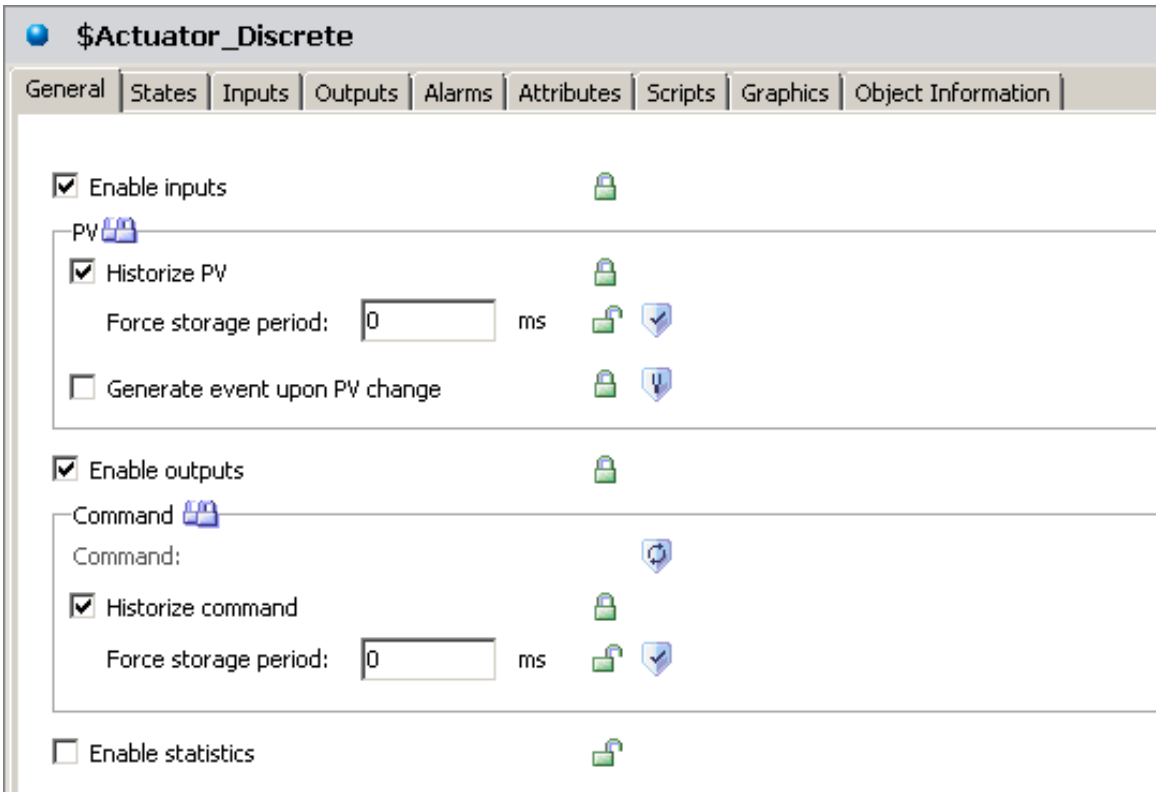


Figura 2.68. Historización *$Actuator_Discrete*

Sensor Discreto

La Plantilla *$Sensor_Discrete* contiene una sola entrada discreta (posición). Por esta razón, se habilita la historización para dicha señal, como se observa en la Figura 2.69.

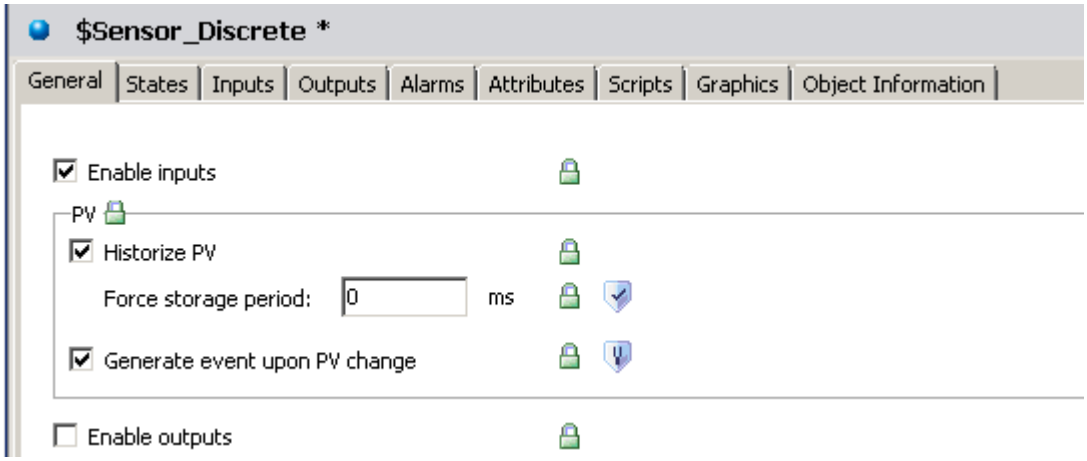


Figura 2.69. Historización *$Sensor_Discrete*

Válvula Discreta

En la Plantilla *$Valve_Discrete* se tienen habilitadas las salidas y entradas (comando y estado, respectivamente). Por lo tanto, se habilita la historización para las dos señales mencionadas, como se observa en la Figura 2.70.

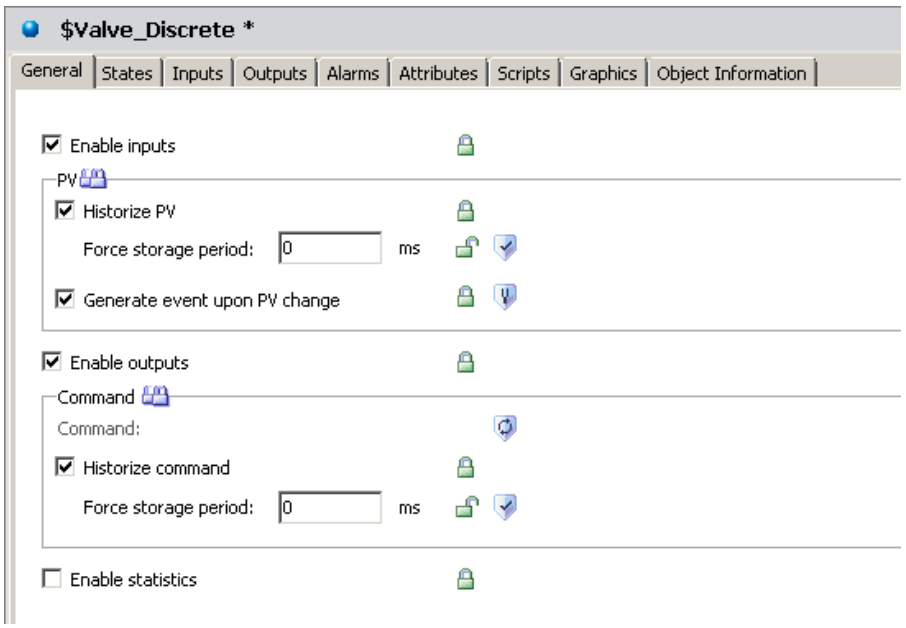


Figura 2.70. Historización *$Valve_Discrete*

Objeto SP

Por último, se encuentra el Objeto de Aplicación SP, que representa las señales de consigna de nivel y de temperatura. Se habilita la configuración de historización en el UDA creado, como se aprecia en la Figura 2.71.

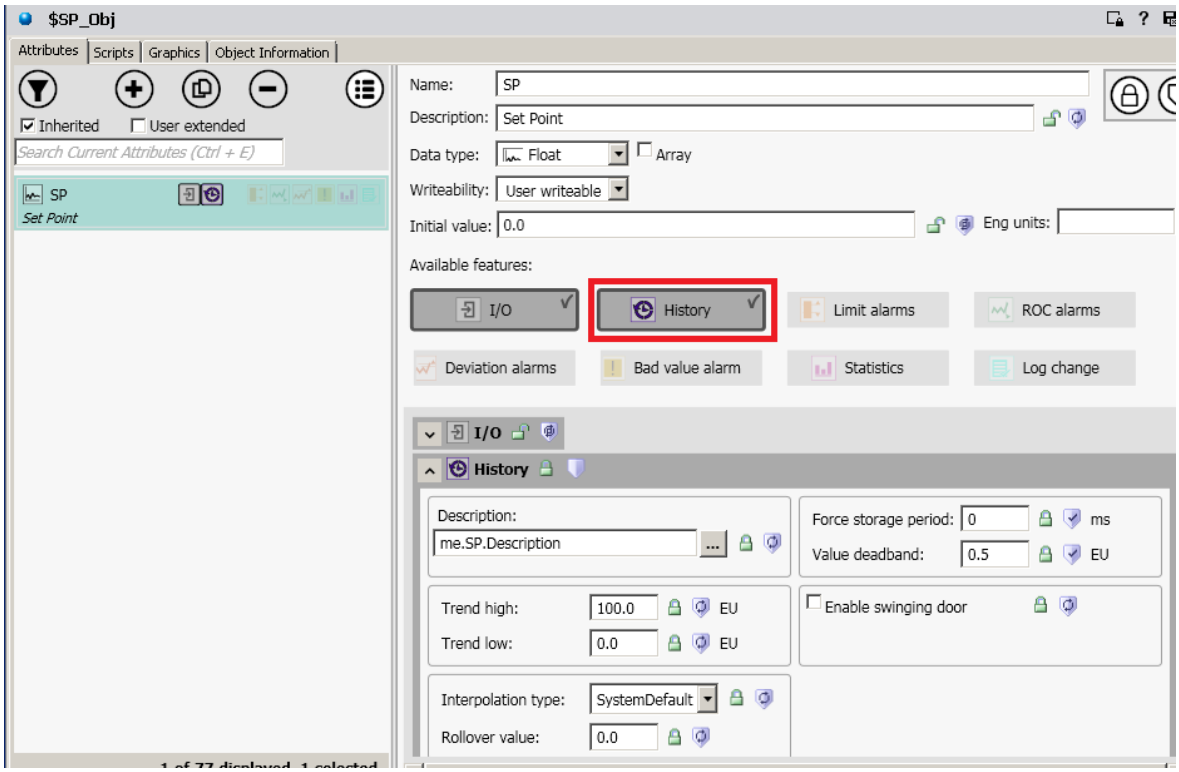


Figura 2.71. Historización *$SP_Obj "SP"*

2.5.8.2. Visualización en *Historian Client Trend*

Para visualizar los gráficos de tendencia, se procede a crear el servidor que contiene las señales historizadas. Dentro de la herramienta *Historian Client Trend*, se selecciona la opción de *Servers*, ubicada en la pestaña *Tools*. Se configura el nombre del servidor como se muestra en la Figura 2.72.

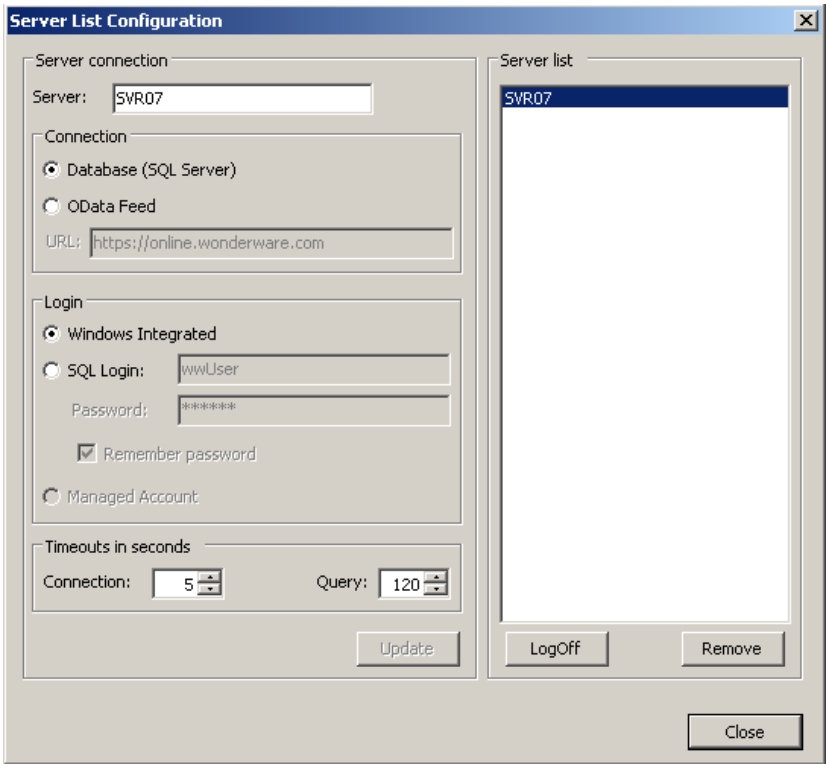


Figura 2.72. Conexión Servidor en *Historian Client Trend*

Al agregar el servidor correspondiente, se despliega un árbol de componentes a la izquierda de la herramienta, mostrando los grupos y señales existentes en la base de datos, como se aprecia en la Figura 2.73.

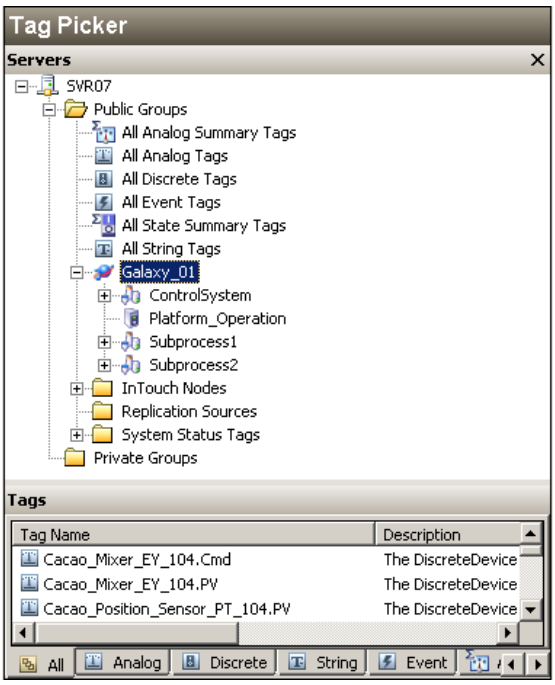


Figura 2.73. *Tag Picker*

2.5.9. Seguridad

Dentro de la plataforma de control supervisorio se encuentran representados todos los elementos de control y el esquema que representa a la planta industrial. Por esta razón, es necesario implementar un sistema de seguridad capaz de restringir y autenticar diferentes usuarios y softwares al realizar eventos, configuraciones o manipulaciones.

2.5.9.1. Configuración de Seguridad

Como se mencionó en la descripción teórica, las herramientas de desarrollo aplicaciones de control supervisorio tiene diferentes modos de autenticación. En este trabajo, se utilizará el modo denominado Galaxy para este servidor de aplicación.

Dentro de las configuraciones de seguridad, se selecciona el modo propuesto en la pestaña correspondiente, como se muestra en la Figura 2.74.

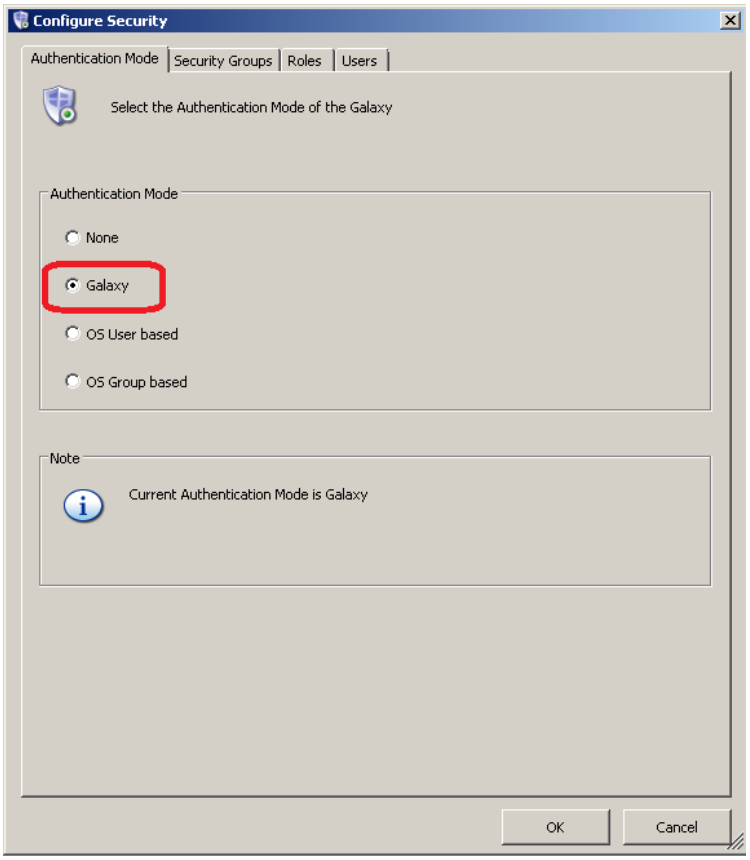


Figura 2.74. Comandos en *AlarmViewerCtrl*

Luego, se procede a crear diferentes grupos de seguridad en la pestaña siguiente. Estos grupos contienen objetos con funciones de seguridad similares. Los grupos y sus objetos correspondientes se describen en la Tabla 2.31.

Tabla 2.31. Grupos de Seguridad

Grupos de Seguridad	Objetos de Automatización
ControlSystem	Objetos que representan componentes del sistema de control.
	Objetos de Sistema (*AppEngine, Area, InTouchViewApp, ViewEngine y WinPlatform*)
	Objetos de Integración de Dispositivos (*DDESuiteLinkClient*)
	Plantillas de los Objetos de Aplicación (*$Actuator_Discrete*, *$gAnalogDevice*, etc.)
Subprocess1	Instancias de los Objetos de Aplicación presentes en el Subproceso 1: (*Cacao_Mixer_EY_104*, *Inlet_Valve_Cacao_ZV_101A*, etc.)
	instancia Area *Subprocess1*
Subprocess2	Instancias de los Objetos de Aplicación presentes en el Subproceso 2: (*Heater_EY202*, *Inlet_Heater_Tank_YV_200B*, etc.)
	instancia Area *Subprocess2*

Una vez creados los grupos de seguridad, se procede a la creación de los diferentes roles que tendrán los usuarios. Los criterios de distribución para estos roles son los siguientes:

Administrator:

Los usuarios con este rol son los desarrolladores de todo el servidor de aplicación, y tienen autorización para realizar cualquier cambio, modificación o configuración. Este rol cuenta con un nivel de accesibilidad completo.

Supervisors:

Los usuarios encargados de la supervisión de los procesos industriales cuentan con este rol. Tienen la posibilidad de realizar cambios en los atributos y cualquier modificación en las alarmas. No poseen privilegios de configuración.

Operators_Subprocess_1:

Los usuarios pertenecientes a este rol son los operadores del Subproceso 1. Tienen permiso para realizar cambios en los atributos configurados en modo '*Operate'*, y pueden reconocer las alarmas.

Operators_Subprocess_2:

Los usuarios pertenecientes a este rol tienen los mismos privilegios que el rol anterior, pero para el Subproceso 2.

Default:

Este rol no posee ningún privilegio.

Con el modo de autenticación *Galaxy*, la creación y modificación de los usuarios se realiza a nivel de la herramienta *ArchestrA IDE*. Como se detalló en secciones anteriores, cada usuario puede tener más de un rol a la vez. Para este sistema de seguridad, se crea un conjunto de usuarios que representan las diferentes funciones dentro del servidor de aplicación. Los usuarios, contraseñas y roles se describen en la Tabla 2.32.

Tabla 2.32. Usuarios y Roles

Usuarios y Contraseñas	**Roles**
Administrator *ww*	*Administrator*, *Supervisors*, *Operators_Subprocess1*, *Operators_Subprocess2*, *Default*
Supervisor *ww*	*Supervisors*, *Operators_Subprocess1*, *Operators_Subprocess2*, *Default*
Operator1 *ww*	*Operators_Subprocess1*, *Default*
Operator2 *ww*	*Operators_Subprocess2*, *Default*
DefaultUser *ww*	*Default*

3. RESULTADOS Y DISCUSIÓN

3.1. Resultados

En esta sección, se presentan los resultados obtenidos de las pruebas realizadas. Se incluye una descripción de la planta simulada y la lógica de programación utilizada en el proceso.

3.1.1. Descripción de la Planta

El objetivo de la planta simulada es la producción de chocolate como producto final. El proceso de producción se divide en dos áreas.

En la primera etapa, se utilizan habas de cacao y leche como materias primas. Las habas de cacao se ingresan y trituran, mientras que la leche se introduce en la línea de producción. Una vez que se alcanza un nivel específico, se realiza la mezcla de los dos productos para obtener el chocolate final, como se muestra en la Figura 3.1.

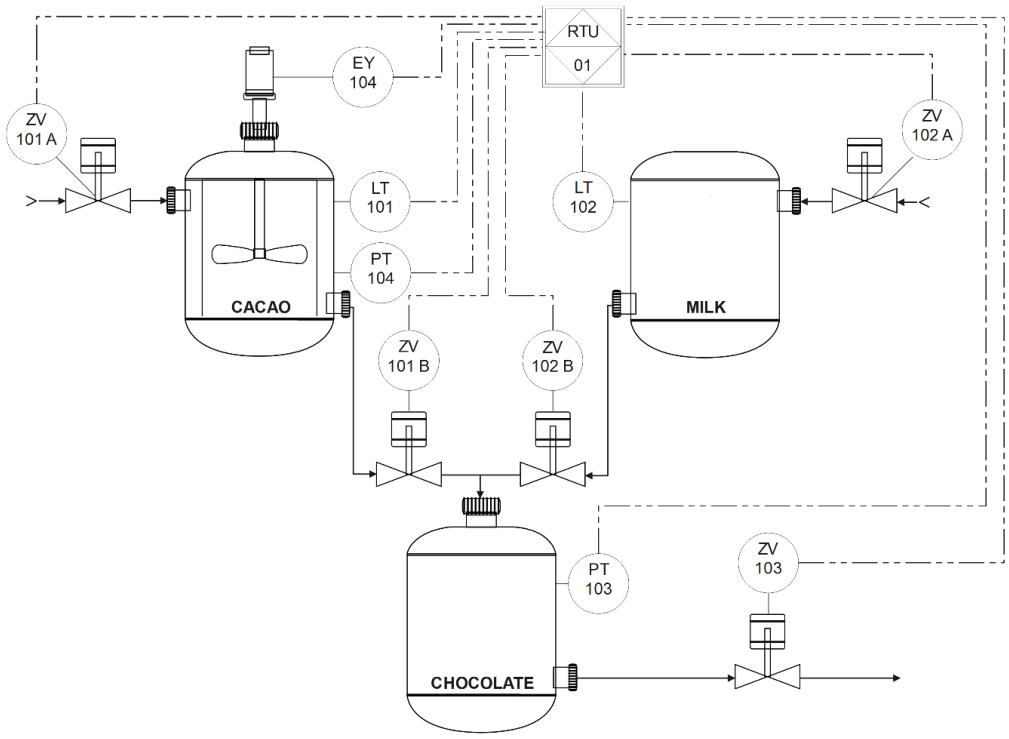


Figura 3.1. P&ID del subproceso 1.

En la segunda etapa de producción, el chocolate se transporta a un área de almacenamiento y se calienta para lograr una salida controlada del producto final, como se puede apreciar en la Figura 3.2.

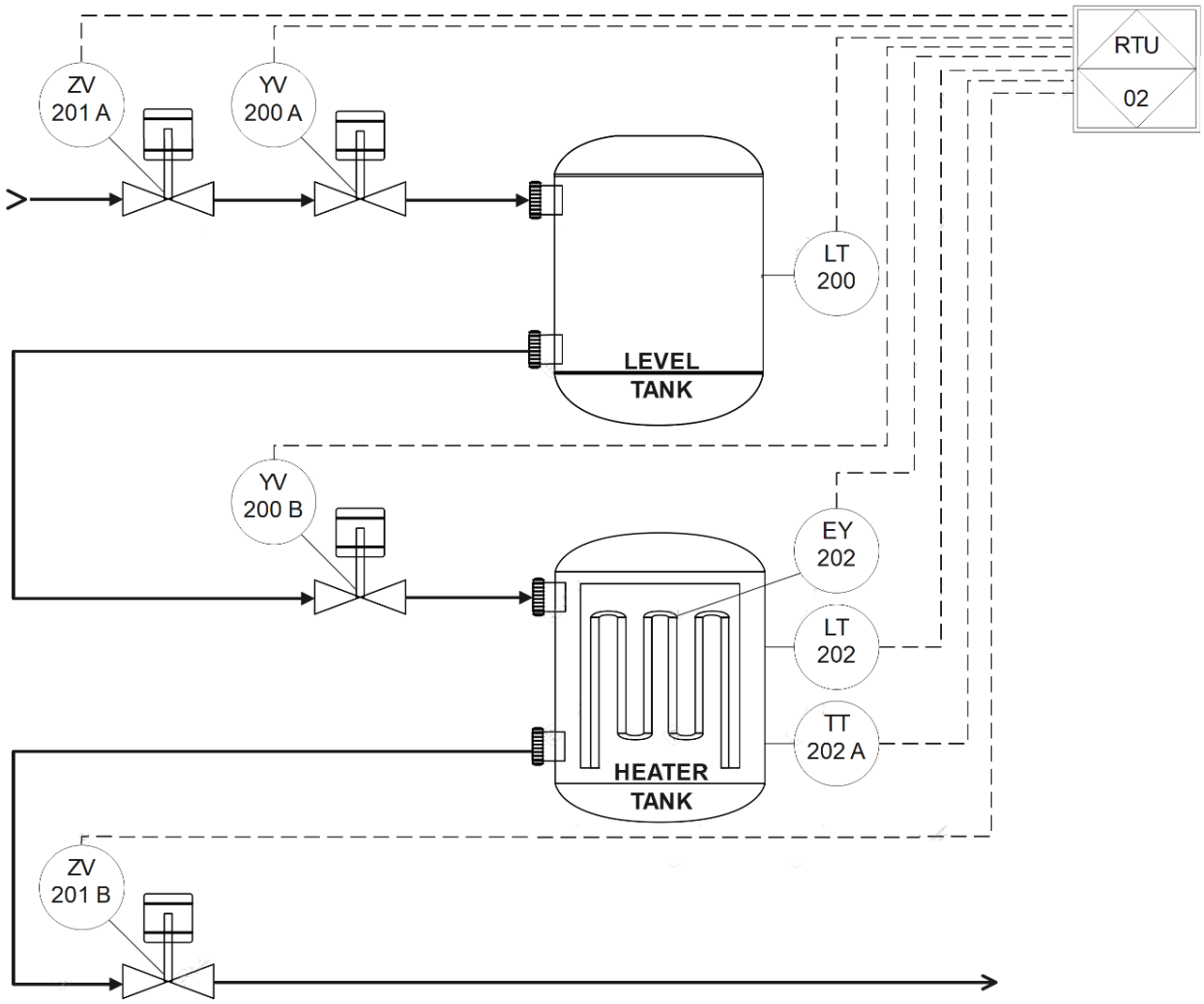


Figura 3.2. P&ID del subproceso 2.

3.1.1.1. Componentes de la planta

Para llevar a cabo el proceso de producción, se requieren diversos instrumentos industriales. A continuación, se presenta la nomenclatura utilizada según el estándar vigente [22], como se muestra en la Tabla 2.33, además de la descripción del significado de los mismos.

Tabla 2.33. Nomenclatura Instrumentos

Identificación Funcional	Número Lazo de Control	Nombre del Instrumento
ZV	101A	*Inlet Valve Cacao*
	101B	*Outlet Valve Cacao*
	102A	*Inlet Valve Milk*
	102B	*Outlet Valve Milk*
	103	*Outlet Valve Chocolate*
	201A	*Inlet Valve*
	201B	*Outlet Valve*
YV	200A	*Inlet Level Tank*
	200B	*Inlet Heater Tank*
PT	103	*Position Transmitter Chocolate*
	104	*Position Transmitter Cacao*
LT	101	*Level Sensor Cacao*
	102	*Level Sensor Milk*
	200	*Level Sensor (Level Tank)*
EY	104	*Cacao Mixer*
	202	*Heater*
TT	202	*Temperature Transmitter (Heater Tank)*

Válvula Discreta (ZV):

Válvula con dos señales discretas de comando (apertura y cierre) y dos variables discretas de estado (completamente abierto y cerrado).

Válvula Analógica (YV):

Válvula con dos señales análogas de comando y dos variables discretas de estado (completamente abierto y cerrado).

Sensor Discreto (PT):
Sensor de posición con una señal discreta de estado.

Sensor de Nivel (LT):
Sensor de nivel con una señal analógica.

Actuador Discreto (EY):
Actuador con una señal discreta de comando y una señal discreta de estado.

Sensor de Temperatura (TT):
Sensor de temperatura con una señal analógica.

3.1.2. Lógica Subproceso 1

La programación del subproceso uno se realiza en el controlador correspondiente a la RTU 01. En esta sección del proceso, se controlan las válvulas de entrada y salida de producto de cada tanque de almacenamiento. En el modo automático, las válvulas se encargan de controlar el nivel de cada tanque siguiendo un valor de consigna ingresado, abriendo y cerrando las válvulas mediante comandos discretos.

Por otro lado, las válvulas también se pueden controlar de forma manual, lo que permite al operador tener control independiente sobre el estado de cada actuador. El esquema se muestra en la Figura 3.3.

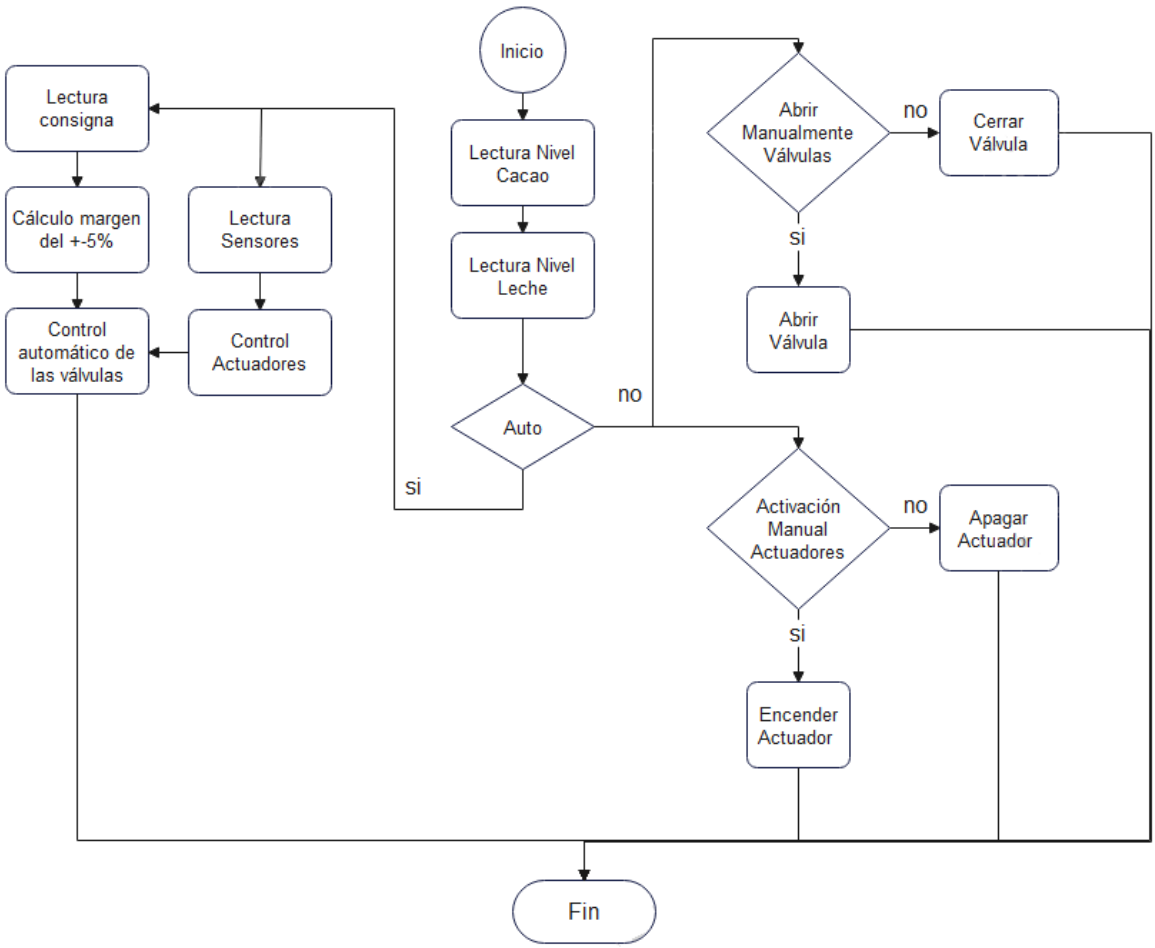


Figura 3.3. Diagrama de Flujo RTU 01

Además, el controlador se encarga del envío y recepción de datos, actuando como interlocutor de la MTU. El PLC de la RTU 01 consta de dos bloques de datos (DB) para el intercambio de datos Profinet, los cuales se detallan en la Tabla 3.1. como se mostró en el capítulo de la metodología.

Tabla 3.1. Bloque de Datos RTU 01

Nombre	DB
“DatosEnviar”	DB1
"DatosRecibir"	DB2

3.1.3. Lógica Subproceso 2

Para la segunda parte del proceso, se utiliza el PLC correspondiente a la RTU 02, y su programación consta de lo siguiente.

En los externos de la línea del proceso, se dispone de dos válvulas con comando discreto, las cuales se pueden activar o desactivar manualmente desde la interfaz HMI, estableciendo un enclavamiento de estas salidas. Además, se cuenta con dos válvulas analógicas que se encargan de controlar el nivel en los tanques de almacenamiento y la temperatura.

Adicionalmente, se dispone de un calentador eléctrico que se controla de forma discreta. Se enciende cuando se registra un valor por debajo de la consigna establecida, y se apaga cuando el sensor de temperatura proporciona un valor superior al 105% de la consigna. Un margen de histéresis del 5% es considerado aceptable para el proceso. El algoritmo de control se muestra en la Figura 3.4.

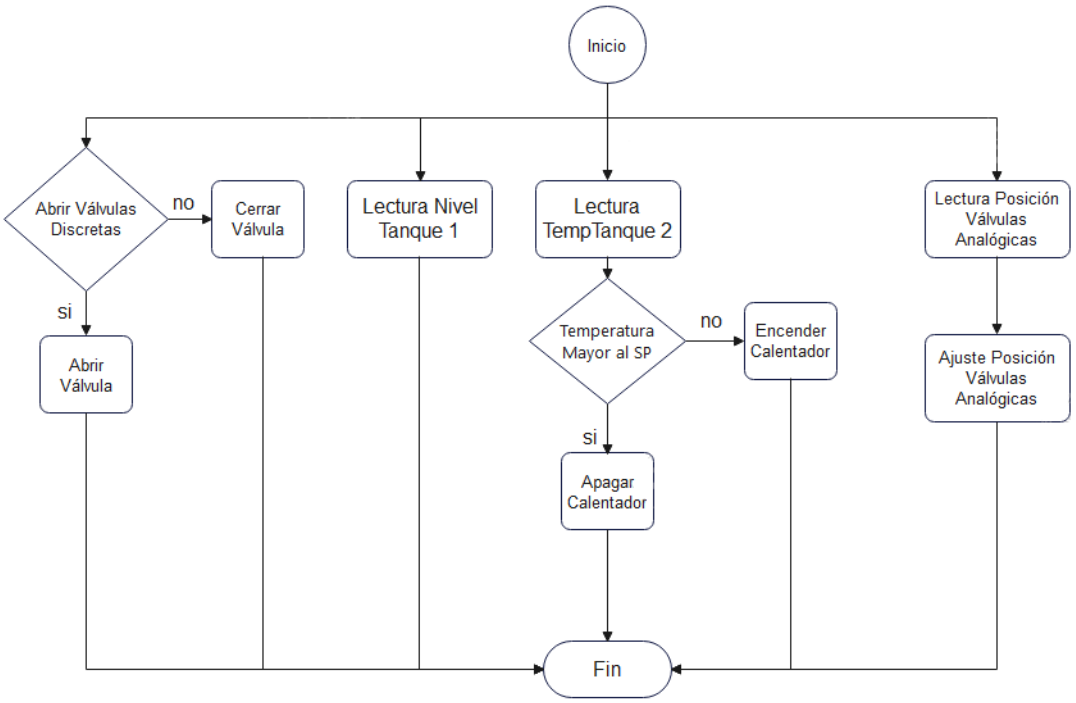


Figura 3.4. Diagrama de Flujo RTU 02

3.1.4. Verificación de Lectura y Escritura de Variables

3.1.4.1. Variables de Proceso

Escritura Variables Discretas

En esta sección, se realiza una verificación de las variables de escritura que son de tipo discreto, es decir, aquellas que tienen un comportamiento binario en sus estados. Las variables a verificarse están destinadas al proceso en las dos áreas de la planta.

Para escribir en estas variables, se observa inicialmente un comando mediante la pulsación de un botón o el cambio de un elemento gráfico discreto dentro del HMI. Estos componentes gráficos se resaltan en la Figura 3.5 y la Figura 3.6.

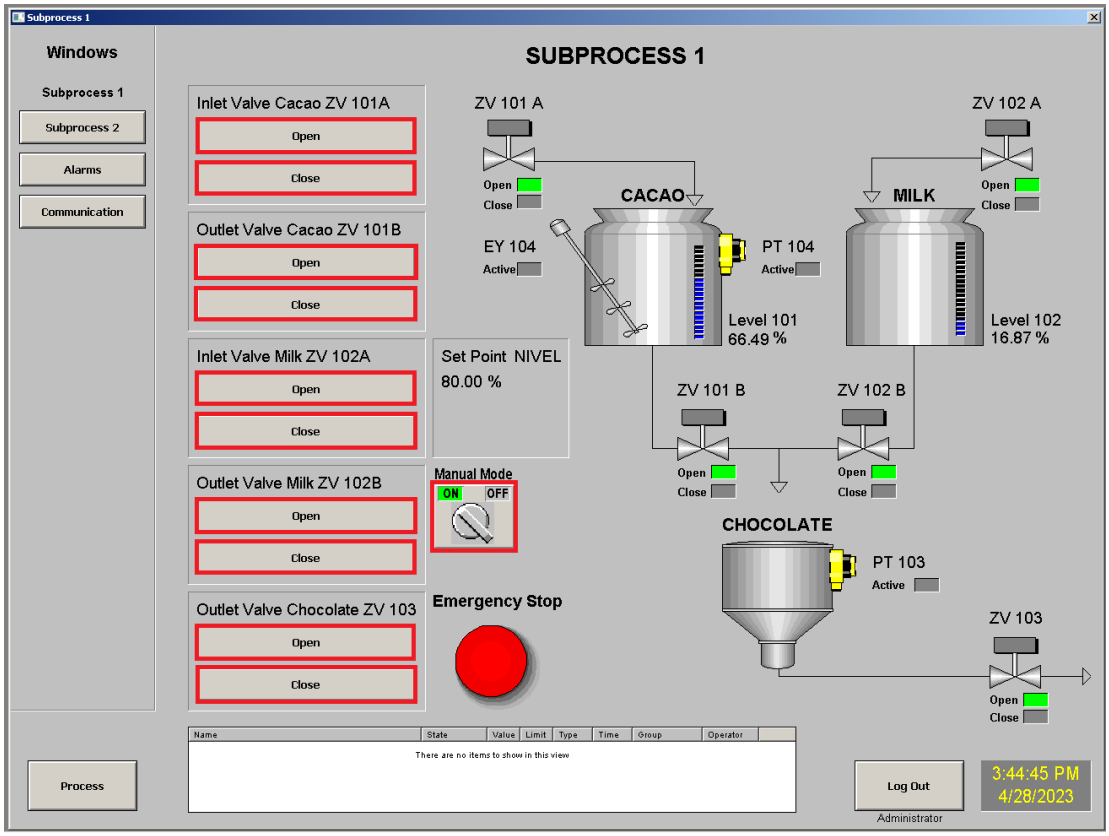


Figura 3.5. Elementos Gráficos de Comando Discretos – Subproceso 1

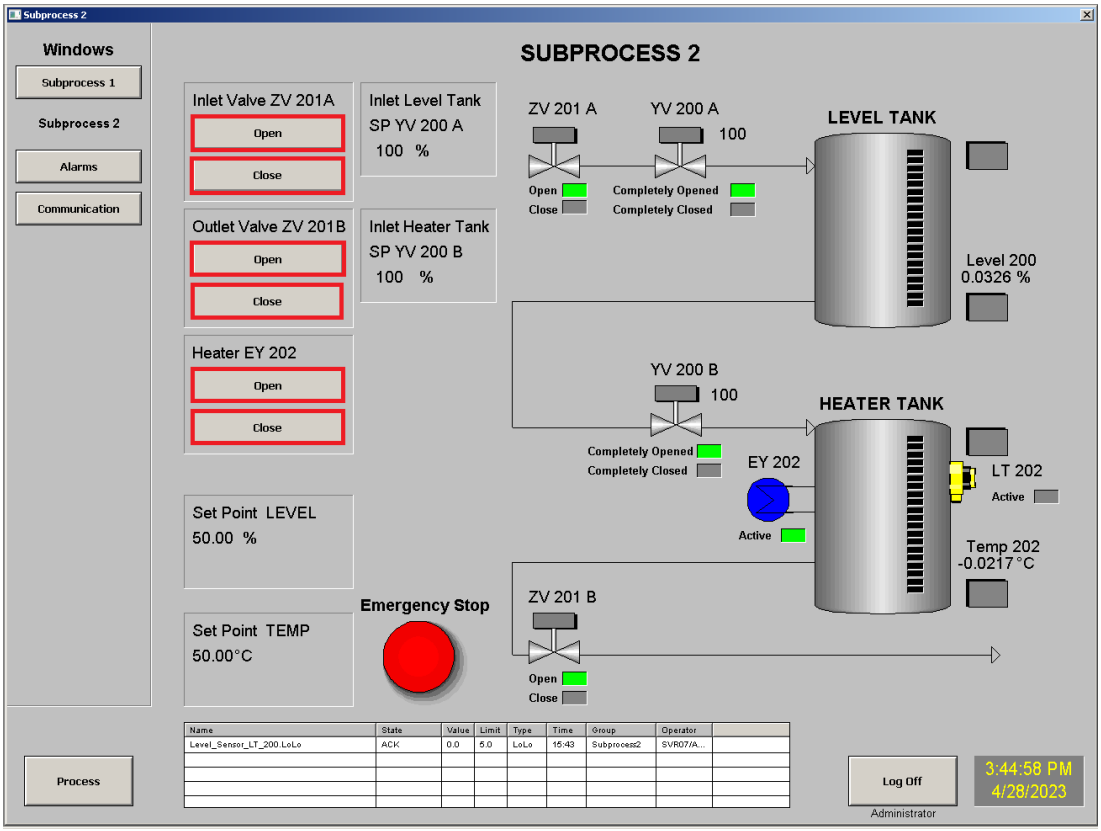


Figura 3.6. Elementos Gráficos de Comando Discretos – Subproceso 2

Los cambios realizados en dichos elementos provocan modificaciones o realizan cambios en los objetos que representan a los componentes del proceso. Los valores resultantes de estas modificaciones se pueden observar en la Figura 3.7.

AttributeReference	Value	Quality	Status
Inlet_Valve_Cacao_ZV_101A.Cmd	Opened	C0:Good	Ok
Outlet_Valve_Cacao_ZV_101B.Cmd	Opened	C0:Good	Ok
Inlet_Valve_Milk_ZV_102A.Cmd	Opened	C0:Good	Ok
Outlet_Valve_Milk_ZV_102B.Cmd	Opened	C0:Good	Ok
Outlet_Valve_Chocolate_ZV103.Cmd	Opened	C0:Good	Ok
MTU.Topic_MTU.Manual_Mode	true	C0:Good	Ok
Inlet_Valve_ZV_201A.Cmd	Opened	C0:Good	Ok
Outlet_Valve_ZV_201B.Cmd	Opened	C0:Good	Ok
Heater_EY202.Cmd	On	C0:Good	Ok

AttributeReference	Value	Quality	Status
Inlet_Valve_Cacao_ZV_101A.Cmd	Closed	C0:Good	Ok
Outlet_Valve_Cacao_ZV_101B.Cmd	Closed	C0:Good	Ok
Inlet_Valve_Milk_ZV_102A.Cmd	Closed	C0:Good	Ok
Outlet_Valve_Milk_ZV_102B.Cmd	Closed	C0:Good	Ok
Outlet_Valve_Chocolate_ZV103.Cmd	Closed	C0:Good	Ok
MTU.Topic_MTU.Manual_Mode	false	C0:Good	Ok
Inlet_Valve_ZV_201A.Cmd	Closed	C0:Good	Ok
Outlet_Valve_ZV_201B.Cmd	Closed	C0:Good	Ok
Heater_EY202.Cmd	Off	C0:Good	Ok

Figura 3.7. Valores de Escritura Discreta en Objetos de Proceso

En el proceso de escritura, cada variable o atributo está enlazado a un *tag* ubicado en el servidor de datos. El valor de dicho *tag* se visualiza en la Figura 3.8.

Name	Value
ZV_101A_CmdClose	TRUE
ZV_101A_CmdOpen	FALSE
ZV_101B_CmdClose	TRUE
ZV_101B_CmdOpen	FALSE
ZV_102A_CmdClose	TRUE
ZV_102A_CmdOpen	FALSE
ZV_102B_CmdClose	TRUE
ZV_102B_CmdOpen	FALSE
ZV_103_CmdClose	TRUE
ZV_103_CmdOpen	FALSE
Manual_Mode	FALSE
ZV_201A_CmdClose	TRUE
ZV_201A_CmdOpen	FALSE
ZV_201B_CmdClose	TRUE
ZV_201B_CmdOpen	FALSE
EY202_Cmd	FALSE

Figura 3.8. Valores de Escritura Discreta de Proceso en el Servidor de Datos

Estas variables están direccionadas a elementos locales (marcas) en el dispositivo MTU. Los valores observados en estos elementos discretos destinados al proceso se presentan en la Figura 3.9.

Nombre	Dirección	Format...	Valor de observac.
"ZV_101A_CmdOpen"	%M100.2	BOOL	TRUE
"ZV_101A_CmdClose"	%M100.3	BOOL	FALSE
"ZV_101B_CmdOpen"	%M100.4	BOOL	TRUE
"ZV_101B_CmdClose"	%M100.5	BOOL	FALSE
"ZV_102A_CmdOpen"	%M100.6	BOOL	TRUE
"ZV_102A_CmdClose"	%M100.7	BOOL	FALSE
"ZV_102B_CmdOpen"	%M101.0	BOOL	TRUE
"ZV_102B_CmdClose"	%M101.1	BOOL	FALSE
"ZV_103_CmdOpen"	%M101.2	BOOL	TRUE
"ZV_103_CmdClose"	%M101.3	BOOL	FALSE
"Manual_Mode"	%M101.5	BOOL	TRUE
"ZV_201A_CmdOpen"	%M300.2	BOOL	TRUE
"ZV_201A_CmdClose"	%M300.3	BOOL	FALSE
"ZV_201B_CmdOpen"	%M300.4	BOOL	TRUE
"ZV_201B_CmdClose"	%M300.5	BOOL	FALSE
"EY202_Cmd"	%M300.6	BOOL	TRUE

Nombre	Dirección	Format...	Valor de observac.
"ZV_101A_CmdOpen"	%M100.2	BOOL	FALSE
"ZV_101A_CmdClose"	%M100.3	BOOL	TRUE
"ZV_101B_CmdOpen"	%M100.4	BOOL	FALSE
"ZV_101B_CmdClose"	%M100.5	BOOL	TRUE
"ZV_102A_CmdOpen"	%M100.6	BOOL	FALSE
"ZV_102A_CmdClose"	%M100.7	BOOL	TRUE
"ZV_102B_CmdOpen"	%M101.0	BOOL	FALSE
"ZV_102B_CmdClose"	%M101.1	BOOL	TRUE
"ZV_103_CmdOpen"	%M101.2	BOOL	FALSE
"ZV_103_CmdClose"	%M101.3	BOOL	TRUE
"Manual_Mode"	%M101.5	BOOL	FALSE
"ZV_201A_CmdOpen"	%M300.2	BOOL	FALSE
"ZV_201A_CmdClose"	%M300.3	BOOL	TRUE
"ZV_201B_CmdOpen"	%M300.4	BOOL	FALSE
"ZV_201B_CmdClose"	%M300.5	BOOL	TRUE
"EY202_Cmd"	%M300.6	BOOL	FALSE

Figura 3.9. Valores de Escritura Discreta de Proceso en Marcas locales MTU

Los valores de estas marcas son trasladados a las variables pertenecientes a los DB. Los elementos presentes en los DB, destinados a enviarse vía Profinet hacia los dispositivos remotos, se visualizan en la Figura 3.10.

DatosEnviar_RTU_01

	Nombre	Tipo...	...	...	Valor de observación
1	Static				
4	ZV_101A_CmdOpen	Bool	...	fal:	FALSE
5	ZV_101A_CmdClose	Bool	...	fal:	TRUE
6	ZV_101B_CmdOpen	Bool	...	fal:	FALSE
7	ZV_101B_CmdClose	Bool	...	fal:	TRUE
8	ZV_102A_CmdOpen	Bool	...	fal:	FALSE
9	ZV_102A_CmdClose	Bool	...	fal:	TRUE
10	ZV_102B_CmdOpen	Bool	...	fal:	FALSE
11	ZV_102B_CmdClose	Bool	...	fal:	TRUE
12	ZV_103_CmdOpen	Bool	...	fal:	FALSE
13	ZV_103_CmdClose	Bool	...	fal:	TRUE
15	PN_Bool_1_Out	Bool	...	fal:	FALSE

DatosEnviar_RTU_02

	Nombre	Tip...	...	...	Valor de observación
1	Static				
4	ZV_201A_CmdOpen	Bool	...	fal:	FALSE
5	ZV_201A_CmdClose	Bool	...	fal:	TRUE
6	ZV_201B_CmdOpen	Bool	...	fal:	FALSE
7	ZV_201B_CmdClose	Bool	...	fal:	TRUE
8	EY202_Cmd	Bool	...	fal:	FALSE

DatosEnviar_RTU_01

	Nombre	Tipo...	...	...	Valor de observación
1	Static				
4	ZV_101A_CmdOpen	Bool	...	fal:	TRUE
5	ZV_101A_CmdClose	Bool	...	fal:	FALSE
6	ZV_101B_CmdOpen	Bool	...	fal:	TRUE
7	ZV_101B_CmdClose	Bool	...	fal:	FALSE
8	ZV_102A_CmdOpen	Bool	...	fal:	TRUE
9	ZV_102A_CmdClose	Bool	...	fal:	FALSE
10	ZV_102B_CmdOpen	Bool	...	fal:	TRUE
11	ZV_102B_CmdClose	Bool	...	fal:	FALSE
12	ZV_103_CmdOpen	Bool	...	fal:	TRUE
13	ZV_103_CmdClose	Bool	...	fal:	FALSE
15	PN_Bool_1_Out	Bool	...	fal:	TRUE

DatosEnviar_RTU_02

	Nombre	Tip...	...	...	Valor de observación
1	Static				
4	ZV_201A_CmdOpen	Bool	...	fal:	TRUE
5	ZV_201A_CmdClose	Bool	...	fal:	FALSE
6	ZV_201B_CmdOpen	Bool	...	fal:	TRUE
7	ZV_201B_CmdClose	Bool	...	fal:	FALSE
8	EY202_Cmd	Bool	...	fal:	TRUE

Figura 3.10. Variables de Escritura Discreta de Proceso en los DB

Una vez que los datos son recibidos por las unidades RTU, se almacenan en marcas internas, como se muestra en la Figura 3.11.

RTU 01

	Nombre	Dirección	For...	Valor de ...
1	"ZV_101A_CmdOpen"	%M400.0	BOOL	TRUE
2	"ZV_101A_CmdClose"	%M400.1	BOOL	FALSE
3	"ZV_101B_CmdOpen"	%M400.2	BOOL	TRUE
4	"ZV_101B_CmdClose"	%M400.3	BOOL	FALSE
5	"ZV_102A_CmdOpen"	%M400.4	BOOL	TRUE
6	"ZV_102A_CmdClose"	%M400.5	BOOL	FALSE
7	"ZV_102B_CmdOpen"	%M400.6	BOOL	TRUE
8	"ZV_102B_CmdClose"	%M400.7	BOOL	FALSE
9	"ZV_103_CmdOpen"	%M401.0	BOOL	TRUE
10	"ZV_103_CmdClose"	%M401.1	BOOL	FALSE
11	"Manual_Mode"	%M402.3	...	TRUE

RTU 02

	Nombre	Dirección	For...	Valor de ...
1	"ZV_201A_CmdOpen"	%M400.0	...	TRUE
2	"ZV_201A_CmdClose"	%M400.1	BOOL	FALSE
3	"ZV_201B_CmdOpen"	%M400.2	BOOL	TRUE
4	"ZV_201B_CmdClose"	%M400.3	BOOL	FALSE
5	"EY202_Cmd"	%M400.4	BOOL	TRUE
6		<Agregar>		

	Nombre	Dirección	For...	Valor de ...
1	"ZV_101A_CmdOpen"	%M400.0	BOOL	FALSE
2	"ZV_101A_CmdClose"	%M400.1	BOOL	TRUE
3	"ZV_101B_CmdOpen"	%M400.2	BOOL	FALSE
4	"ZV_101B_CmdClose"	%M400.3	BOOL	TRUE
5	"ZV_102A_CmdOpen"	%M400.4	BOOL	FALSE
6	"ZV_102A_CmdClose"	%M400.5	BOOL	TRUE
7	"ZV_102B_CmdOpen"	%M400.6	BOOL	FALSE
8	"ZV_102B_CmdClose"	%M400.7	BOOL	TRUE
9	"ZV_103_CmdOpen"	%M401.0	BOOL	FALSE
10	"ZV_103_CmdClose"	%M401.1	BOOL	TRUE
11	"Manual_Mode"	%M402.3	...	FALSE

	Nombre	Dirección	For...	Valor de ...
1	"ZV_201A_CmdOpen"	%M400.0	...	FALSE
2	"ZV_201A_CmdClose"	%M400.1	BOOL	TRUE
3	"ZV_201B_CmdOpen"	%M400.2	BOOL	FALSE
4	"ZV_201B_CmdClose"	%M400.3	BOOL	TRUE
5	"EY202_Cmd"	%M400.4	BOOL	FALSE
6		<Agregar>		

Figura 3.11. Valores de Escritura Discreta de Proceso en Marcas locales RTU

Estos valores discretos verificados anteriormente se utilizan para modificar las variables físicas mediante las salidas Q del controlador o realizan un procesamiento previo en la lógica del controlador.

Escritura Variables Analógicas

En esta verificación, se realiza un seguimiento de las variables analógicas utilizadas para comandar o configurar una consigna dentro del proceso de la planta. Las variables verificadas corresponden a la consigna de nivel en el subproceso 1, así como las consignas de nivel y temperatura en el subproceso 2. Además, se verifican las señales analógicas que sirven como entrada de comando en las válvulas.

Inicialmente, se observan los valores de las variables de escritura analógica, como se muestra en la Figura 3.12 y la Figura 3.13. Posteriormente, se lleva a cabo una comparación de estos valores a lo largo de la ruta de comunicación.

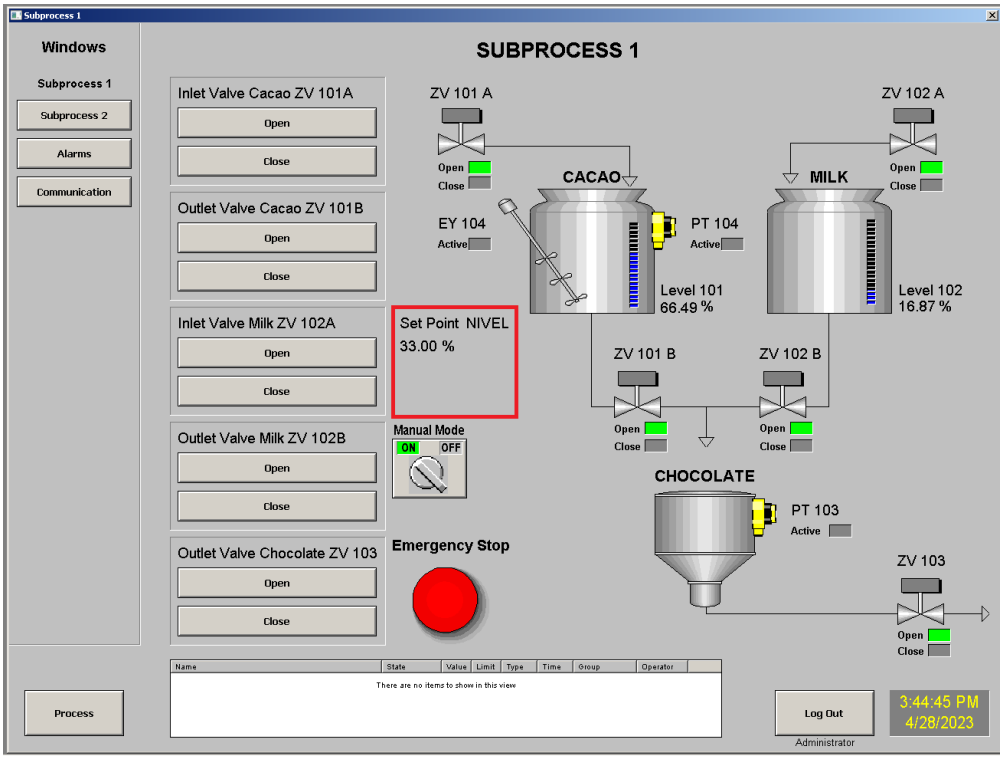


Figura 3.12. Variable de Consigna de Nivel en HMI – Subproceso 1

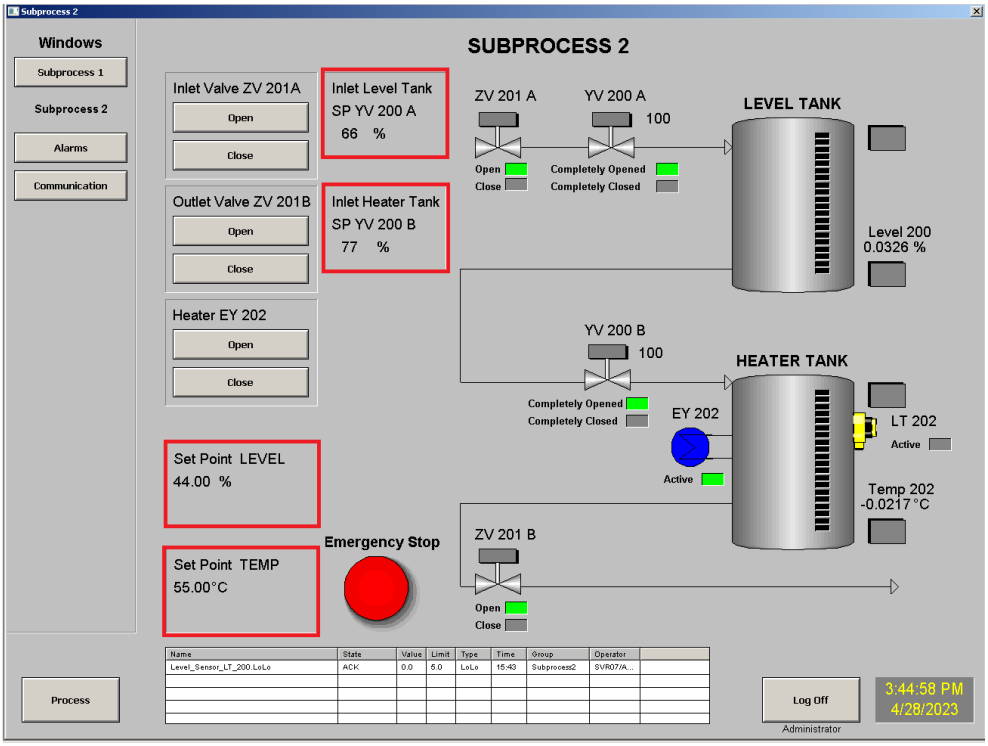


Figura 3.13. Variables Consignas y Posición de Válvulas en HMI – Subproceso 2

Los valores presentes en los componentes del HMI ocasionan modificaciones en los atributos de los objetos asociados. Estos cambios se pueden observar en la Figura 3.14.

AttributeReference	Value
SP_Level_Milk_Cacao.SP	33.0
SP_Level.SP	44.0
SP_Temp.SP	55.0
Inlet_Level_Tank_YV_200A.PV	66.0
Inlet_Heater_Tank_YV_200B.PV	77.0

Figura 3.14. Valores de Escritura Analógica en Objetos de Proceso

De manera similar, estos valores ocasionan cambios en los atributos direccionados a las variables ubicadas en el dispositivo de control (MTU). Dichos atributos se pueden observar en el diagnóstico de las comunicaciones presentes en el servidor de datos, tal como se muestra en la Figura 3.15.

Name	Value
Cacao_Milk_SP_Level	33
SP_Level_Tank1	44
SP_Temp_Tank2	55
YV_200A_SP	66
YV_200B_SP	77

Figura 3.15. Valores de Escritura Analógica de Proceso en el Servidor de Datos

Estas variables están direccionadas a elementos locales (marcas) en el dispositivo MTU. Los valores observados en estos elementos analógicos destinados al proceso se presentan en la Figura 3.16.

Nombre	Dirección	...	Valor de observación
"Cacao_Milk_SP_Level"	%MD150	...	33.0
"SP_Level_Tank1"	%MD354	...	44.0
"SP_Temp_Tank2"	%MD358	...	55.0
"YV_200A_SP"	%MW350	...	66
"YV_200B_SP"	%MW352	...	77

Figura 3.16. Valores de Escritura Analógica de Proceso en Marcas locales MTU

Estos valores de las marcas se trasladan a las variables pertenecientes a los DB. La visualización de los elementos presentes en los DB, destinados a enviarse vía Profinet hacia los dispositivos remotos, se muestra en la Figura 3.17.

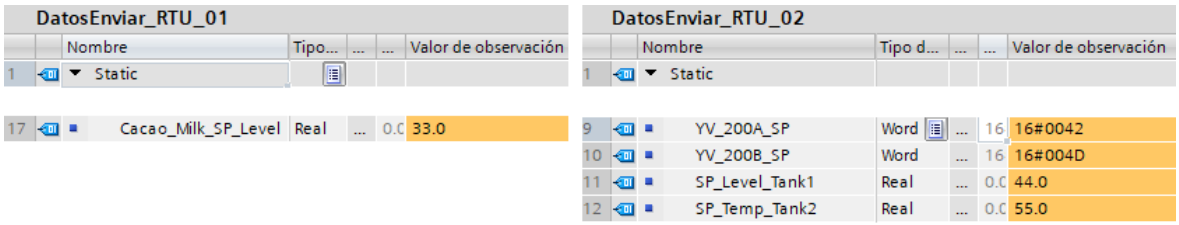


Figura 3.17. Variables de Escritura Analógica de Proceso en los DB

Después de haber recibido las variables mediante Profinet, las RTU almacenan los valores en marcas internas, tal como se observa en la Figura 3.18.

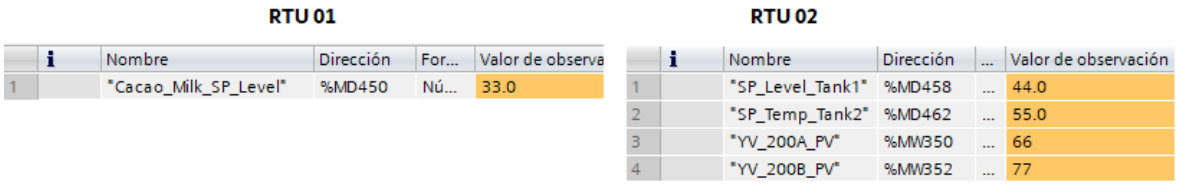


Figura 3.18. Valores de Escritura Anagógica de Proceso en Marcas locales RTU

Finalmente, se realiza un cómputo de dichas señales y se procede a la salida de los dispositivos.

Lectura Variables Discretas

En esta sección, se realiza la verificación de las variables binarias obtenidas en el proceso. Estas señales corresponden a los estados de apertura y cierre de las válvulas, así como el estado de los dispositivos discretos.

En la Figura 3.19, se observan las marcas locales pertenecientes a los dispositivos remotos RTU. Los valores de estas variables se han trasladado desde las entradas lógicas *%I*.

RTU 01

	Nombre	Dirección	Formato visualización	Valor de observación
1	"ZV_101A_CS"	%M300.0	BOOL	FALSE
2	"ZV_101A_OS"	%M300.1	BOOL	TRUE
3	"ZV_101B_CS"	%M300.2	BOOL	FALSE
4	"ZV_101B_OS"	%M300.3	BOOL	TRUE
5	"ZV_102A_CS"	%M300.4	BOOL	FALSE
6	"ZV_102A_OS"	%M300.5	BOOL	TRUE
7	"ZV_102B_CS"	%M300.6	BOOL	FALSE
8	"ZV_102B_OS"	%M300.7	BOOL	TRUE
9	"ZV_103_CS"	%M301.0	BOOL	FALSE
10	"ZV_103_OS"	%M301.1	BOOL	TRUE
11	"PT_104_Sts"	%M301.2	BOOL	TRUE
12	"PT_103_Sts"	%M301.3	BOOL	TRUE
13	"EY_104_Sts"	%M301.4	BOOL	TRUE

RTU 02

	Nombre	Dirección	Formato visualización	Valor de observación
1	"ZV_201A_CS"	%M300.0	BOOL	FALSE
2	"ZV_201A_OS"	%M300.1	BOOL	TRUE
3	"ZV_201B_CS"	%M300.2	BOOL	FALSE
4	"ZV_201B_OS"	%M300.3	BOOL	TRUE
5	"YV_200A_CS"	%M300.4	BOOL	FALSE
6	"YV_200A_OS"	%M300.5	BOOL	TRUE
7	"YV_200B_CS"	%M300.6	BOOL	FALSE
8	"YV_200B_OS"	%M300.7	BOOL	TRUE
9	"LT_202_Sts"	%M301.0	BOOL	TRUE
10	"EY202_Sts"	%M301.1	BOOL	TRUE

Figura 3.19. Valores de Lectura Discreta de Proceso en Marcas locales RTU

Estas variables se trasladan a los DB de cada unidad RTU y se envían mediante Profinet hasta el dispositivo MTU. Los DB recibidos por dicho dispositivo se pueden observar en la Figura 3.20.

DatosRecibir_RTU_01

	Nombre	Tipo..	...	...	Valor de observación
1	Static				
16	ZV_101A_CS	Bool	...	fal:	FALSE
17	ZV_101A_OS	Bool	...	fal:	TRUE
18	ZV_101B_CS	Bool	...	fal:	FALSE
19	ZV_101B_OS	Bool	...	fal:	TRUE
20	ZV_102A_CS	Bool	...	fal:	FALSE
21	ZV_102A_OS	Bool	...	fal:	TRUE
22	ZV_102B_CS	Bool	...	fal:	FALSE
23	ZV_102B_OS	Bool	...	fal:	TRUE
24	ZV_103_CS	Bool	...	fal:	FALSE
25	ZV_103_OS	Bool	...	fal:	TRUE
26	PT_104_Sts	Bool	...	fal:	TRUE
27	PT_103_Sts	Bool	...	fal:	TRUE
28	EY_104_Sts	Bool	...	fal:	TRUE

DatosRecibir_RTU_02

	Nombre	Tipo ...	...	...	Valor de observación
1	Static				
16	ZV_201A_CS	Bool	...	fal:	FALSE
17	ZV_201A_OS	Bool	...	fal:	TRUE
18	ZV_201B_CS	Bool	...	fal:	FALSE
19	ZV_201B_OS	Bool	...	fal:	TRUE
20	YV_200A_CS	Bool	...	fal:	FALSE
21	YV_200A_OS	Bool	...	fal:	TRUE
22	YV_200B_CS	Bool	...	fal:	FALSE
23	YV_200B_OS	Bool	...	fal:	TRUE
24	LT_202_Sts	Bool	...	fal:	TRUE
25	EY202_Sts	Bool	...	fal:	TRUE

Figura 3.20. Variables de Lectura Discreta de Proceso en los DB

Las variables presentes en los DB se trasladan a las marcas internas del dispositivo MTU, como se muestra en la Figura 3.21.

	i	Nombre	Dirección	...	Valor de observación
1		"ZV_101A_CS"	%M216.0	...	FALSE
2		"ZV_101A_OS"	%M216.1	...	TRUE
3		"ZV_101B_CS"	%M216.2	...	FALSE
4		"ZV_101B_OS"	%M216.3	...	TRUE
5		"ZV_102A_CS"	%M216.4	...	FALSE
6		"ZV_102A_OS"	%M216.5	...	TRUE
7		"ZV_102B_CS"	%M216.6	...	FALSE
8		"ZV_102B_OS"	%M216.7	...	TRUE
9		"ZV_103_CS"	%M217.0	...	FALSE
10		"ZV_103_OS"	%M217.1	...	TRUE
11		"PT_104_Sts"	%M217.2	...	TRUE
12		"PT_103_Sts"	%M217.3	...	TRUE
13		"EY_104_Sts"	%M217.4	...	TRUE
14		"ZV_201A_OS"	%M416.1	...	TRUE
15		"ZV_201A_CS"	%M416.0	...	FALSE
16		"ZV_201B_OS"	%M416.3	...	TRUE
17		"ZV_201B_CS"	%M416.2	...	FALSE
18		"YV_200A_CS"	%M416.4	...	FALSE
19		"YV_200A_OS"	%M416.5	...	TRUE
20		"YV_200B_CS"	%M416.6	...	FALSE
21		"YV_200B_OS"	%M416.7	...	TRUE
22		"LT_202_Sts"	%M417.0	...	TRUE
23		"EY202_Sts"	%M417.1	...	TRUE

	i	Nombre	Dirección	...	Valor de observación
1		"ZV_101A_CS"	%M216.0	...	TRUE
2		"ZV_101A_OS"	%M216.1	...	FALSE
3		"ZV_101B_CS"	%M216.2	...	TRUE
4		"ZV_101B_OS"	%M216.3	...	FALSE
5		"ZV_102A_CS"	%M216.4	...	TRUE
6		"ZV_102A_OS"	%M216.5	...	FALSE
7		"ZV_102B_CS"	%M216.6	...	TRUE
8		"ZV_102B_OS"	%M216.7	...	FALSE
9		"ZV_103_CS"	%M217.0	...	TRUE
10		"ZV_103_OS"	%M217.1	...	FALSE
11		"PT_104_Sts"	%M217.2	...	TRUE
12		"PT_103_Sts"	%M217.3	...	TRUE
13		"EY_104_Sts"	%M217.4	...	FALSE
14		"ZV_201A_OS"	%M416.1	...	FALSE
15		"ZV_201A_CS"	%M416.0	...	TRUE
16		"ZV_201B_OS"	%M416.3	...	FALSE
17		"ZV_201B_CS"	%M416.2	...	TRUE
18		"YV_200A_CS"	%M416.4	...	TRUE
19		"YV_200A_OS"	%M416.5	...	FALSE
20		"YV_200B_CS"	%M416.6	...	TRUE
21		"YV_200B_OS"	%M416.7	...	FALSE
22		"LT_202_Sts"	%M417.0	...	TRUE
23		"EY202_Sts"	%M417.1	...	FALSE

Figura 3.21. Valores de Lectura Discreta de Proceso en Marcas locales MTU

A continuación, el servidor de aplicación procede a la lectura de dichas variables discretas de proceso, las cuales se evidencian en la Figura 3.22.

Name	R/W Status	Value
ZV_101A_CS	R/W	FALSE
ZV_101A_OS	R/W	TRUE
ZV_101B_CS	R/W	FALSE
ZV_101B_OS	R/W	TRUE
ZV_102A_CS	R/W	FALSE
ZV_102A_OS	R/W	TRUE
ZV_102B_CS	R/W	FALSE
ZV_102B_OS	R/W	TRUE
ZV_103_CS	R/W	FALSE
ZV_103_OS	R/W	TRUE
PT_104_Sts	R/W	TRUE
PT_103_Sts	R/W	TRUE
EY_104_Sts	R/W	TRUE
ZV_201A_CS	R/W	FALSE
ZV_201A_OS	R/W	TRUE
ZV_201B_CS	R/W	FALSE
ZV_201B_OS	R/W	TRUE
YV_200A_CS	R/W	FALSE
YV_200A_OS	R/W	TRUE
YV_200B_CS	R/W	FALSE
YV_200B_OS	R/W	TRUE
LT_202_Sts	R/W	TRUE
EY202_Sts	R/W	TRUE

Figura 3.22. Valores de Lectura Discreta de Proceso en el Servidor de Datos

Los atributos de los objetos direccionados a dichas variables también cambian de estado, y este cambio se puede observar en la Figura 3.23.

AttributeReference	Value	Quality
Inlet_Valve_Cacao_ZV_101A.PV	Opened	C0:Good
Outlet_Valve_Cacao_ZV_101B.PV	Opened	C0:Good
Inlet_Valve_Milk_ZV_102A.PV	Opened	C0:Good
Outlet_Valve_Milk_ZV_102B.PV	Opened	C0:Good
Outlet_Valve_Chocolate_ZV103.PV	Opened	C0:Good
Cacao_Position_Sensor_PT_104.PV	Detected	C0:Good
Chocolate_Position_Sensor_PT_103.PV	Detected	C0:Good
Cacao_Mixer_EY_104.PV	On	C0:Good
Inlet_Valve_ZV_201A.PV	Opened	C0:Good
Outlet_Valve_ZV_201B.PV	Opened	C0:Good
Inlet_Level_Tank_YV_200A.YV_CS	false	C0:Good
Inlet_Level_Tank_YV_200A.YV_OS	true	C0:Good
Inlet_Heater_Tank_YV_200B.YV_CS	false	C0:Good
Inlet_Heater_Tank_YV_200B.YV_OS	true	C0:Good
Sensor_Position_Level_LT_202.PV	Detected	C0:Good
Heater_EY202.PV	On	C0:Good

AttributeReference	Value	Quality
Inlet_Valve_Cacao_ZV_101A.PV	Closed	C0:Good
Outlet_Valve_Cacao_ZV_101B.PV	Closed	C0:Good
Inlet_Valve_Milk_ZV_102A.PV	Closed	C0:Good
Outlet_Valve_Milk_ZV_102B.PV	Closed	C0:Good
Outlet_Valve_Chocolate_ZV103.PV	Closed	C0:Good
Cacao_Position_Sensor_PT_104.PV	Detected	C0:Good
Chocolate_Position_Sensor_PT_103.PV	Detected	C0:Good
Cacao_Mixer_EY_104.PV	Off	C0:Good
Inlet_Valve_ZV_201A.PV	Closed	C0:Good
Outlet_Valve_ZV_201B.PV	Closed	C0:Good
Inlet_Level_Tank_YV_200A.YV_CS	true	C0:Good
Inlet_Level_Tank_YV_200A.YV_OS	false	C0:Good
Inlet_Heater_Tank_YV_200B.YV_CS	true	C0:Good
Inlet_Heater_Tank_YV_200B.YV_OS	false	C0:Good
Sensor_Position_Level_LT_202.PV	Detected	C0:Good
Heater_EY202.PV	Off	C0:Good

Figura 3.23. Valores de Lectura Discreta en Objetos de Proceso

Finalmente, estos valores se visualizan en las pantallas del HMI, como se muestra en la Figura 3.24 y la Figura 3.25

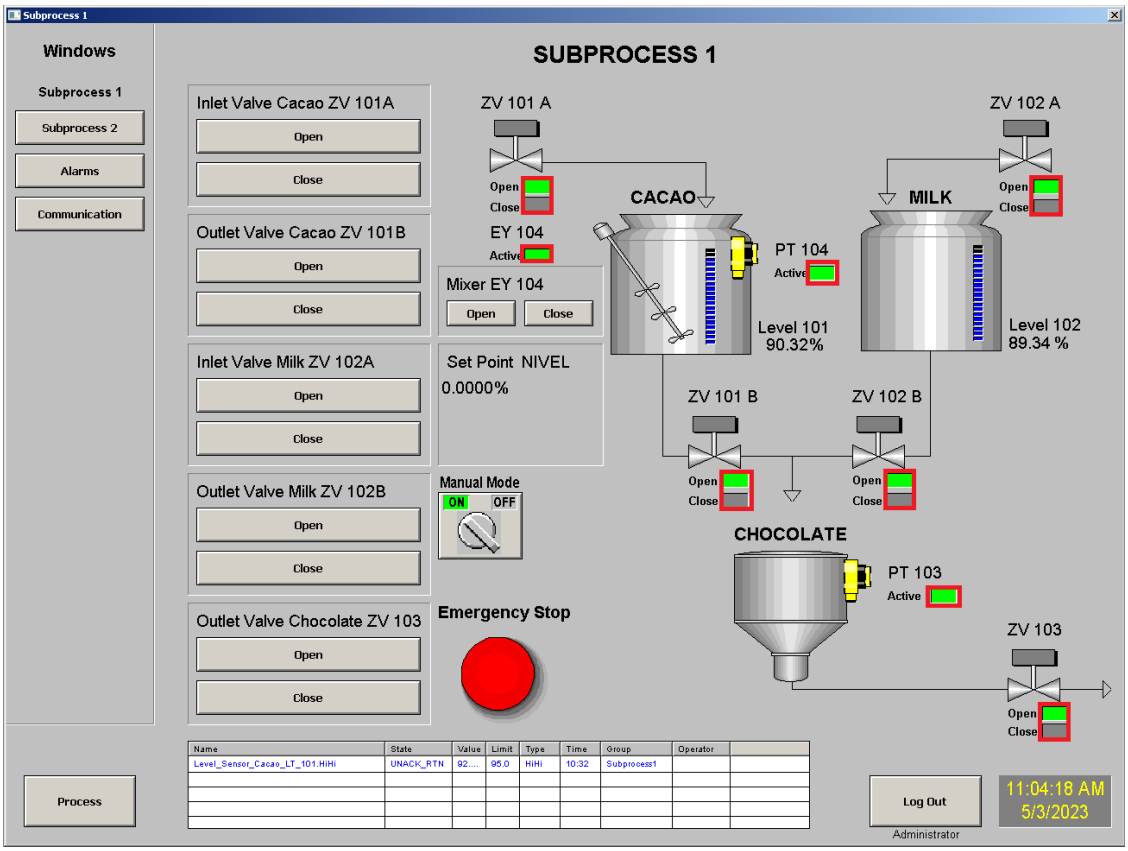


Figura 3.24. Visualización de Elementos Discretos – Subproceso 1

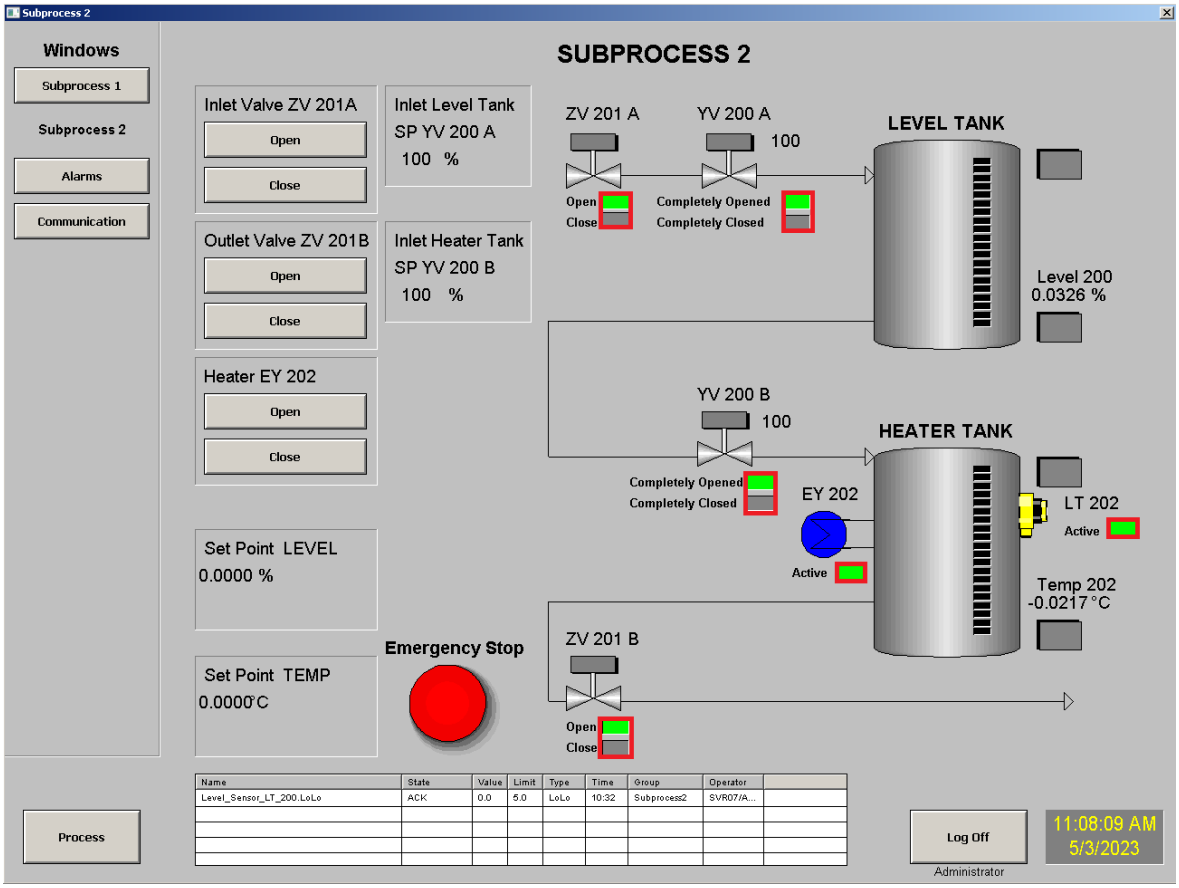


Figura 3.25. Visualización de Elementos Discretos – Subproceso 2

Lectura Variables Analógicas

En esta verificación, se examinan las señales de lectura analógica, que incluyen las variables de proceso (PV) y el estado de la posición del vástago de las válvulas con comando analógico.

En la Figura 3.26, se visualizan las marcas locales ubicadas en los dispositivos RTU. Estos valores de proceso se trasladan desde las entradas analógicas *%IW.*

RTU 01

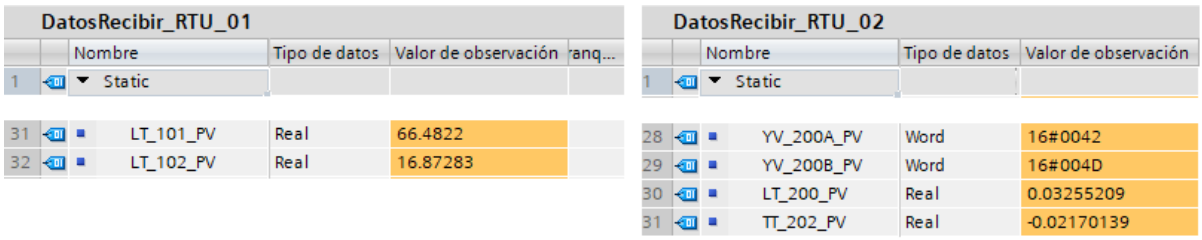
	Nombre	Dirección	...	Valor de observación
1	"LT_101_PV"	%MD350	...	66.48944
2	"LT_102_PV"	%MD354	...	16.84389

RTU 02

	Nombre	Dirección	...	Valor de observación
1	"YV_200A_PV"	%MW350	...	66
2	"YV_200B_PV"	%MW352	...	77
3	"LT_200_PV"	%MD354	...	0.03255209
4	"TT_202_PV"	%MD358	...	-0.01808449

Figura 3.26. Valores de Lectura Analógica de Proceso en Marcas locales RTU

Estas variables de proceso se trasladan a los DB de cada RTU respectivamente y posteriormente son enviadas hacia la unidad MTU mediante Profinet. Los DB recibidos por dicho dispositivo se pueden observar en la Figura 3.27.



DatosRecibir_RTU_01

	Nombre	Tipo de datos	Valor de observación	ranq...
1	Static			
31	LT_101_PV	Real	66.4822	
32	LT_102_PV	Real	16.87283	

DatosRecibir_RTU_02

	Nombre	Tipo de datos	Valor de observación
1	Static		
28	YV_200A_PV	Word	16#0042
29	YV_200B_PV	Word	16#004D
30	LT_200_PV	Real	0.03255209
31	TT_202_PV	Real	-0.02170139

Figura 3.27. Variables de Lectura Analógica de Proceso en los DB

Estas variables se trasladan a las marcas internas del dispositivo MTU, como se muestra en la Figura 3.28.

	Nombre	Dirección	...	Valor de observación
1	"LT_101_PV"	%MD250	...	66.4822
2	"LT_102_PV"	%MD254	...	16.86198
3	"YV_200A_PV"	%MW450	...	66
4	"YV_200B_PV"	%MW452	...	77
5	"LT_200_PV"	%MD454	...	0.03255209
6	"TT_202_PV"	%MD458	...	-0.01446759

Figura 3.28. Valores de Lectura Analógica de Proceso en Marcas locales MTU

En la Figura 3.29, se observan los componentes del HMI que representan el nivel correspondiente en los tanques de cacao (101) y leche (102) del subproceso 1.

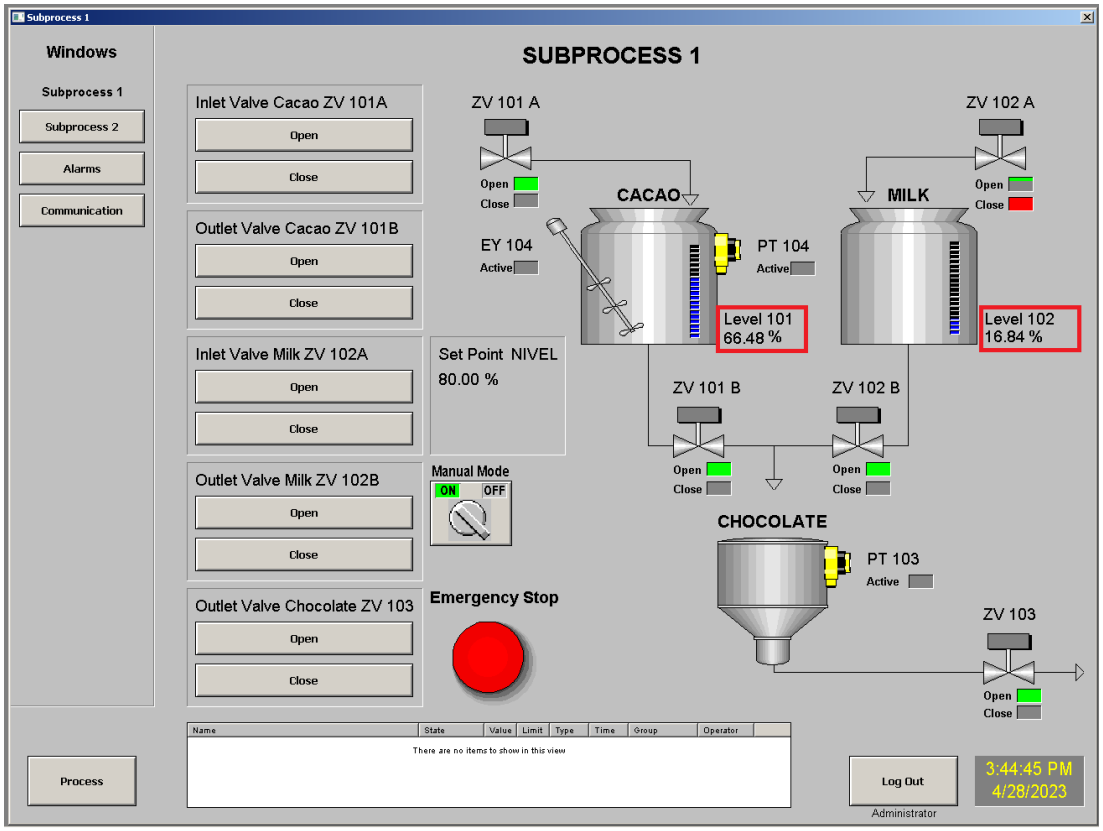


Figura 3.29. Indicadores de Valores de Proceso de Nivel en el Subproceso 1

Además, en la Figura 3.30 se resaltan los indicadores que representan la posición actual de las válvulas de comando analógico, así como los valores de nivel en porcentaje y temperatura en grados centígrados.

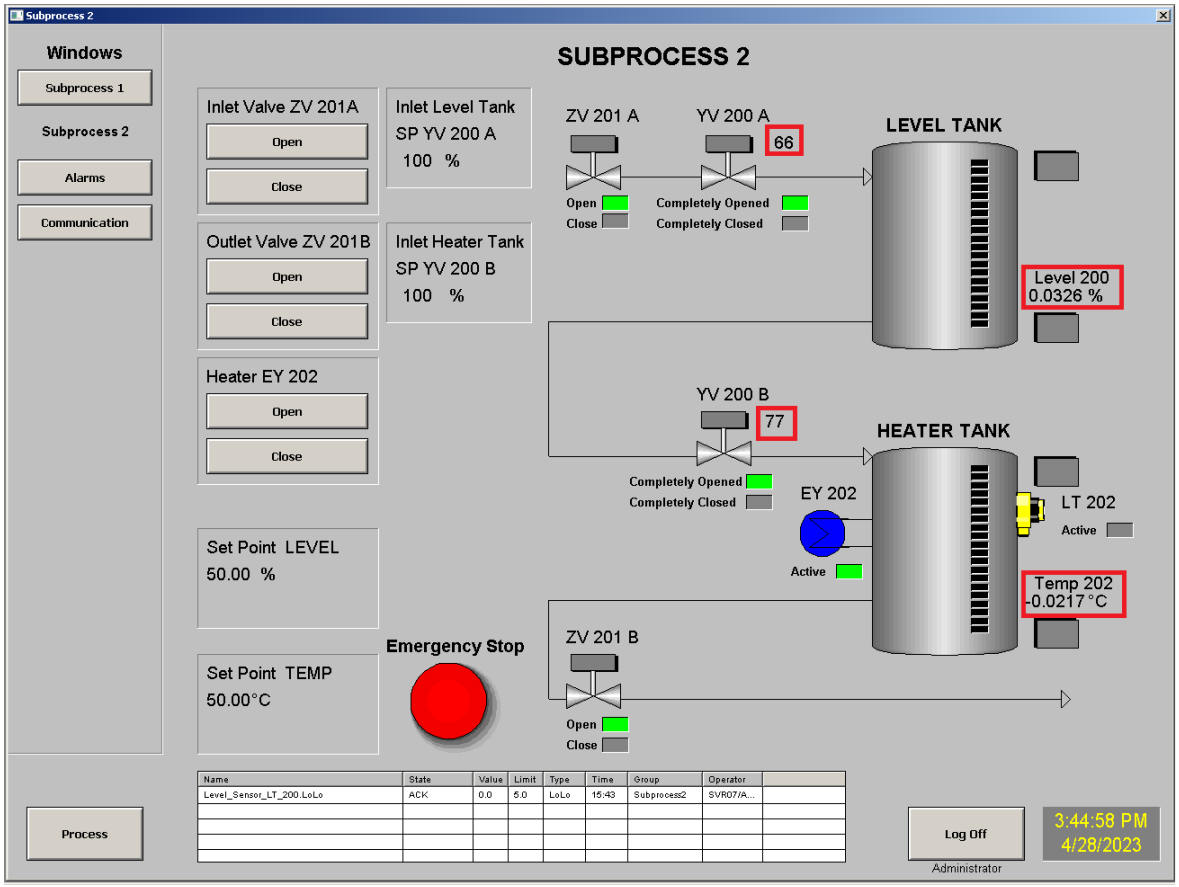


Figura 3.30. Valores de Proceso de Nivel y Temperatura y Posición de Válvulas en el Subproceso 2

3.1.4.2. Variables de Comunicación

En la ventana de comunicación, se encuentran varios indicadores que representan el estado de la red Profinet. En esta sección, se verifica el origen y representación de estas variables desde su generación hasta su visualización en el HMI. Estas variables de estado y digitales se generan en las instrucciones de envío y recepción TCP descritas anteriormente.

Las señales generadas se almacenan en marcas locales de los dispositivos, como se observa en la Figura 3.31.

	i	Nombre	Dirección	Formato visualiza..	Valor de observac..
3		"TSEND_C_RTU_01_Reset_Error"	%M500.0	BOOL	FALSE
4		"TRCV_C_RTU_01_Reset_Error"	%M500.1	BOOL	FALSE
5		"TSEND_C_RTU_01_DONE"	%M500.2	BOOL	FALSE
6		"TSEND_C_RTU_01_BUSY"	%M500.3	BOOL	FALSE
7		"TSEND_C_RTU_01_ERROR"	%M500.4	BOOL	FALSE
8		"TSEND_C_RTU_01_ERROR_DETECTED"	%M500.5	BOOL	FALSE
9		"TSEND_C_RTU_01_STATUS"	%MW502	Hex	16#7004
10		"TSEND_C_RTU_01_STATUS_DETECTED"	%MW504	Hex	16#0000
11		"TRCV_C_RTU_01_DONE"	%M506.0	BOOL	FALSE
12		"TRCV_C_RTU_01_BUSY"	%M506.1	BOOL	TRUE
13		"TRCV_C_RTU_01_ERROR"	%M506.2	BOOL	FALSE
14		"TRCV_C_RTU_01_ERROR_DETECTED"	%M506.3	BOOL	FALSE
15		"TRCV_C_RTU_01_STATUS"	%MW508	Hex	16#7002
16		"TRCV_C_RTU_01_STATUS_DETECTED"	%MW510	Hex	16#0000
17		"TSEND_C_RTU_02_Reset_Error"	%M550.0	BOOL	FALSE
18		"TRCV_C_RTU_02_Reset_Error"	%M550.1	BOOL	FALSE
19		"TSEND_C_RTU_02_DONE"	%M550.2	BOOL	FALSE
20		"TSEND_C_RTU_02_BUSY"	%M550.3	BOOL	FALSE
21		"TSEND_C_RTU_02_ERROR"	%M550.4	BOOL	FALSE
22		"TSEND_C_RTU_02_ERROR_DETECTED"	%M550.5	BOOL	FALSE
23		"TSEND_C_RTU_02_STATUS"	%MW552	Hex	16#7004
24		"TSEND_C_RTU_02_STATUS_DETECTED"	%MW554	Hex	16#0000
25		"TRCV_C_RTU_02_DONE"	%M556.0	BOOL	FALSE
26		"TRCV_C_RTU_02_BUSY"	%M556.1	BOOL	TRUE
27		"TRCV_C_RTU_02_ERROR"	%M556.2	BOOL	FALSE
28		"TRCV_C_RTU_02_ERROR_DETECTED"	%M556.3	BOOL	FALSE
29		"TRCV_C_RTU_02_STATUS"	%MW558	Hex	16#7002
30		"TRCV_C_RTU_02_STATUS_DETECTED"	%MW560	Hex	16#0000

Figura 3.31. Variables de Estado de Comunicación en Marcas MTU

Estas variables también se obtienen bajo demanda a través del servidor de datos y están direccionadas a los objetos de aplicación que tienen dichas variables de estado como atributos. Los valores de los atributos se muestran en la Figura 3.32.

AttributeReference	Value	Quality
Communication_MTU.TSEND_C_RTU_01_DONE	true	C0:Good
Communication_MTU.TSEND_C_RTU_01_BUSY	false	C0:Good
Communication_MTU.TRCV_C_RTU_01_DONE	false	C0:Good
Communication_MTU.TRCV_C_RTU_01_BUSY	true	C0:Good
Communication_MTU.TSEND_C_RTU_02_DONE	false	C0:Good
Communication_MTU.TSEND_C_RTU_02_BUSY	false	C0:Good
Communication_RTU_01.RTU_01_TSEND_C_DONE	false	C0:Good
Communication_RTU_01.RTU_01_TSEND_C_BUSY	true	C0:Good
Communication_RTU_01.RTU_01_TRCV_C_DONE	false	C0:Good
Communication_RTU_01.RTU_01_TRCV_C_BUSY	true	C0:Good
Communication_RTU_02.RTU_02_TSEND_C_DONE	false	C0:Good
Communication_RTU_02.RTU_02_TSEND_C_BUSY	true	C0:Good
Communication_RTU_02.RTU_02_TRCV_C_DONE	false	C0:Good
Communication_RTU_02.RTU_02_TRCV_C_BUSY	true	C0:Good
Communication_MTU.TSEND_C_RTU_01_STATUS	0	C0:Good
Communication_MTU.TRCV_C_RTU_01_STATUS	28674	C0:Good
Communication_MTU.TSEND_C_RTU_02_STATUS	28676	C0:Good
Communication_MTU.TRCV_C_RTU_02_STATUS	28674	C0:Good

Figura 3.32. Variables del Estado de la Comunicación - Objetos de Aplicación

Por último, se pueden observar dichos estados en la ventana de comunicación, como se resalta en la Figura 3.33.

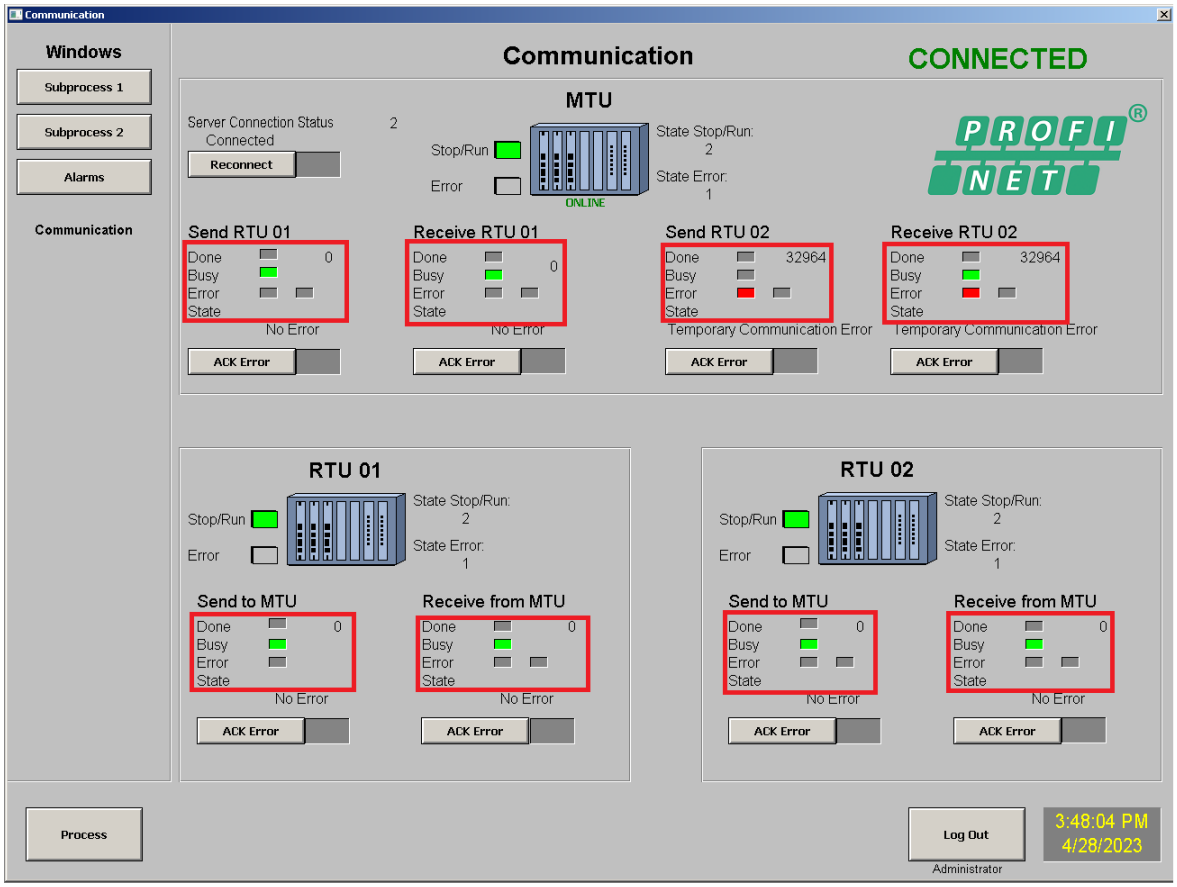


Figura 3.33. Estado de la Comunicación de Red en Ventana Comunicación

3.1.4.3. Variables de Diagnóstico de Dispositivo

El siguiente conjunto de señales corresponde al diagnóstico de los dispositivos de control y al estado de la conexión del Objeto de Integración de Dispositivo hacia el dispositivo MTU.

Las señales de diagnóstico se obtienen de la misma manera que el resto de las variables, a través del servidor de datos. Por otra parte, el origen de la variable de conexión se encuentra en el propio objeto, el cual proporciona información sobre la

conexión al servidor. Los valores de los atributos descritos se pueden observar en la Figura 3.34.

AttributeReference	Value
Communication_MTU.MTU_Sts_LED_Stop_Run	2
Communication_MTU.MTU_Sts_LED_ERROR	1
Communication_RTU_01.RTU_01_Sts_LED_Stop_Run	2
Communication_RTU_01.RTU_01_Sts_LED_ERROR	1
Communication_RTU_02.RTU_02_Sts_LED_Stop_Run	2
Communication_RTU_02.RTU_02_Sts_LED_ERROR	1
MTU.ConnectionStatus	Connected
MTU.Topic_MTU.ItemErrorCntAlarm	false
MTU.Topic_MTU.ItemErrorCnt	0

Figura 3.34. Variables de diagnóstico de dispositivo y conexión del Objeto

3.1.5. Verificación Historiador

Para verificar las variables almacenadas en el historiador, se llevó a cabo la configuración de un intervalo de tiempo dentro de la herramienta de visualización.

En esta pantalla se presentan las variables que se han configurado para ser almacenadas en el historiador de procesos, y se proporciona una gráfica de los valores guardados. La Figura 3.35. muestra la representación gráfica del historiador.

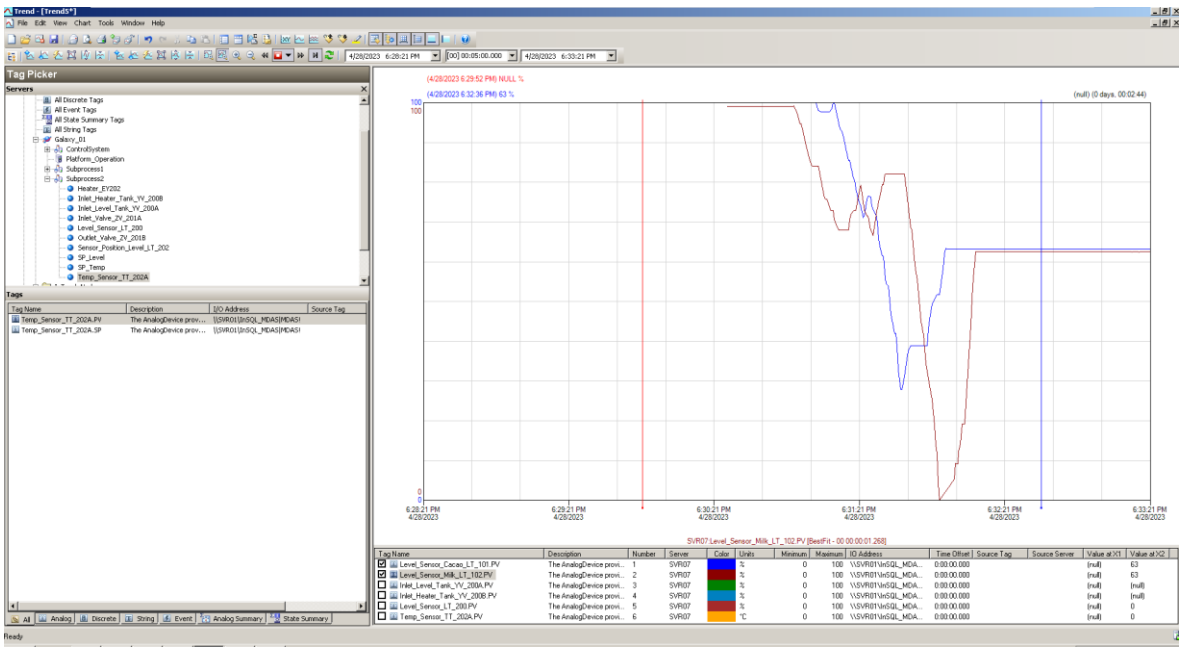


Figura 3.35. Grafica del Historiador de Procesos

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- Se ha realizado un estudio sobre los diferentes tipos de sistemas de control industrial, para posteriormente establecer el que se ajuste a las necesidades de la implementación requerida para el laboratorio de Redes Industriales. El enfoque principal se centró en la arquitectura CIM, que sirvió como base para la implementación de los componentes en los dos módulos didácticos. Además, se realizó una revisión bibliográfica del protocolo de comunicación Profinet y de las herramientas utilizadas en una plataforma de supervisión para el sistema de control industrial, tomando en consideración las normativas y especificaciones correspondientes.
- Se ha establecido una red de control Profinet con tres controladores. Dos de ellos se ubicaron en secciones de la planta, actuando como unidades terminales remotas para controlar estas secciones geográficamente separadas. El tercer PLC se está a cargo de la adquisición de datos de las unidades remotas, centralizando la información y transmitiéndola a la plataforma de control supervisorio.
- Se establecieron los requerimientos y funcionalidades necesarios para el diseño e implementación de los módulos del sistema de control industrial. Estos requerimientos se basaron en la arquitectura y componentes discutidos en la sección del marco teórico, y posteriormente se utilizaron las herramientas necesarias, tanto de hardware como de software, para cumplir con los requerimientos propuestos del sistema de control industrial.
- Durante la revisión de los protocolos de comunicación industrial, se ha determinado que el protocolo Profinet cumple con las características necesarias para lograr una comunicación óptima entre los controladores. Esto se debe a su alta velocidad y rendimiento, lo que permite una rápida y eficiente transferencia de datos. Además, Profinet es compatible con Ethernet y está ampliamente disponible, lo que facilita su integración con las redes existentes en la planta. La escalabilidad y flexibilidad de Profinet son características clave que lo hacen adecuado para su implementación en

entornos industriales. Basándonos en estas consideraciones, se ha desarrollado un programa en los controladores lógicos programables (PLC) de la celda de control para intercambiar y transmitir información a través de Profinet. Esto permite una comunicación eficiente y confiable entre los diferentes dispositivos de la planta, mejorando así la eficiencia operativa y facilitando la toma de decisiones basada en datos precisos y oportunos.

- Durante la programación de control, se ha determinado que se requerían cuatro puertos lógicos para la transmisión de datos: dos para enviar información a las unidades remotas y otros dos para recibir variables de lectura comunicadas desde la unidad centralizadora.
- La interfaz HMI se ajustó a los requisitos de la planta de producción de chocolate, definiendo la arquitectura de la interfaz en términos de procesos, comunicación y control de variables.
- Se llevó a cabo la validación de cada uno de los componentes de los módulos para asegurar el cumplimiento de los requerimientos establecidos. Se realizó un control en la planta y se validó la transmisión de datos en tres grupos: variables de proceso, variables de comunicación y variables de diagnóstico de dispositivos. Esta validación de señales se realizó en todo el recorrido de transmisión, desde la interfaz HMI hasta su aplicación en el proceso industrial.
- Se ha realizado la validación de la representación de las variables almacenadas en la base de datos del histórico, mostrándolas en forma de curvas de tendencia histórica.

4.2. RECOMENDACIONES

- Se recomienda adicionar mecanismos de duplicación en los ICS con el objetivo de aumentar la confiabilidad en el proceso de automatización de la planta, especialmente si se trata de una infraestructura crítica. Para lograr esto, se sugiere incorporar componentes y elementos de redundancia en los diferentes componentes del sistema, lo que generará una mayor confianza durante el control del sistema.

- Se sugiere implementar otro método de autenticación en caso de que la cantidad de usuarios aumente, centralizando los datos en una estación independiente. Los métodos de autenticación basados en el sistema operativo se recomiendan para almacenar información y realizar configuraciones cuando hay un gran número de usuarios.
- Se recomienda utilizar el instructivo adjunto para llevar a cabo la implementación de un programa de control que se encargue de la comunicación Profinet entre dos PLC.

5. REFERENCIAS BIBLIOGRÁFICAS

[1] A. Venkateshwaran, *Computer Integrated Manufacturing*. Delhi: PHI Learning, 2010.

[2] A. Rodríguez, *Sistemas SCADA*, Tercera. México D.F.: Alfaomega Grupo Editor, 2013.

[3] H. Kaschel C. y E. Pinto L., «ANALISIS DEL ESTADO DEL ARTE DE LOS BUSES DE CAMPO APLICADOS AL CONTROL DE PROCESOS INDUSTRIALES», Santiago.

[4] «PROFINET Website». https://us.profinet.com/tecnologia/profinet-es/ (accedido 8 de febrero de 2023).

[5] INTERNATIONAL ELECTROTECHNICAL COMMISSION, «IEC 61784-2 Industrial communication networks». 2019.

[6] M. Bowne, «Canales de Comunicación Profinet», *https://us.profinet.com/canales-de-comunicacion-profinet/*, 2020.

[7] C. Kozierok, *The TCP/IP Guide*. San Francisco, CA: No Starch Press, 2005.

[8] Wonderware, «Application Server User's Guide». Lake Forest, CA, 2016.

[9] P. Fritzson y V. Engelson, «A Unified Object-Oriented Language for System Modeling and Simulation», *Linköping University*.

[10] M. Krotofil y D. Gollmann, «Industrial control systems security: What is happening?», en *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*, Hamburg: IEEE, 2013, pp. 670-675. doi: 10.1109/INDIN.2013.6622964.

[11] U. Makhsud, «Identification and Authentication», *International Journal of Academic Pedagogical Research (IJAPR)*, vol. 5, 2021.

[12] E. Schiavone, A. Ceccarelli, y A. Bondavalli, «Continuous Authentication and Non-repudiation for the Security of Critical Systems», *35th Symposium on Reliable Distributed Systems*, 2016.

[13] M. Swanson, J. Hash, y P. Bowen, «Guide for Developing Security Plans for Federal Information Systems», *NIST Special Publication 800-18*. National Institute of Standards and Technology (NIST), 2006.

[14] Hollifield Bill, D. Oliver, I. Nimmo, y E. Habibi, *The High Performance HMI Handbook A Comprehensive Guide to Designing, Implementing and Maintaining Effective HMIs for Industrial Plant Operations*. 2008.

[15] «Human Machine Interfaces for Process Automation Systems ANSI/ISA-101.01-2015».

[16] Y. Reddy y B. Mehta, *Industrial Process Automation systems: Design and implementation*. Butterworth-Heinemann, 2014.

[17] T. Stauffer y P. Clarke, «Using alarms as a layer of protection», *Process Safety Progress*, vol. 35, n.º 1, pp. 76-83, mar. 2016, doi: 10.1002/prs.11739.

[18] AVEVA, «Alarms and Events Guide», 2021.

[19] SIEMENS, «SIMATIC S7 Controlador programable S7-1200 Manual de sistema». Postfach, 2018.

[20] Wonderware, «Training Manual System Platform», 2009.

[21] Wonderware, «Historian Client User Guide». Schneider Electric, Lake Forest, CA, 2016.

[22] «Norma ISA S5.1».

6. ANEXOS

ANEXO A. Detalle de los Parámetros de TSEND_C

ANEXO B. Instructivo del HMI

ANEXO C. Instructivo de conexión TCP – Profinet

ANEXO A

Se detalla la descripción de cada uno de los Parámetros de TSEND_C

Parámetro	I/O	Tipo datos	Memoria	Descripción
REQ	Input	BOOL	I, Q, M, D, L	Se inicia petición de transmisión con un flanco ascendente.
CONT	Input	BOOL	I, Q, M, D, L	Controla conexión: • 0: Deshacer la conexión • 1: Establecer y mantener la conexión Cuando se envían los datos (flanco ascendente en REQ), el parámetro CONT deberá estar en TRUE para establecer o mantener la conexión.
LEN	Input	UINT	I, Q, M, D, L o constante	Número máximo (en bytes) enviados con la petición. Si utiliza solo valores simbólicos en DATA, el parámetro LEN debe ser "0".
CONNECT	InOut	TCON_Param	D	Puntero hacia la descripción de la conexión
DATA	InOut	VARIANT	I, Q, M, D, L	Puntero hacia el área de transmisión que contiene la dirección y la longitud de los datos para enviar (máxima: 8192 bytes). Estas deben ser idénticas tanto en el lado de transmisión como en el de recepción.
COM_RST	InOut	BOOL	I, Q, M, D, L	Reinicia la instrucción: • 0: Irrelevante • 1: Reinicia la instrucción (deshace la conexión y la reinicia)
DONE	Output	BOOL	I, Q, M, D, L	Parámetro del estado: • 0: Petición no iniciada aún o está en ejecución • 1: Petición finalizada (sin errores)
BUSY	Output	BOOL	I, Q, M, D, L	Parámetro de estado:

				• 0: Petición no iniciada aún o ya finalizada • 1: Petición no finalizada aún. (No se puede iniciar una nueva petición).
ERROR	Output	BOOL	I, Q, M, D, L	Parámetro de estado: • 0: Sin error • 1: Errores
STATUS	Output	WORD	I, Q, M, D, L	Estado de la instrucción

Para la Instrucción TRCV_C se tiene como parámetro:

Parámetro	**I/O**	**Tipo datos**	**Memoria**	**Descripción**
EN_R	Input	BOOL	I, Q, M, D, L	Habilitación de la recepción
RCVD_LEN	Output	UINT	I, Q, M, D, L	Cantidad de datos realmente recibidos (en bytes)

En la siguiente tabla se muestra la relación entre los parámetros BUSY, DONE y ERROR de TSEND_C y TRCV_C

BUSY	**DONE**	**ERROR**	**Descripción**
1	-	-	Se está procesando la petición
0	1	0	Se ha ejecutado correctamente la petición
0	0	1	La petición ha finalizado con un error
0	0	0	Ninguana petición nueva se ha asignado

ANEXO B

En este anexo se presenta un instructivo sobre el uso y manejo de las ventanas en la interfaz de operación (HMI).

Inicio Sesión

La primera pantalla que aparece al iniciar el sistema es la de inicio de sesión. Esta pantalla contiene un botón que permite abrir una ventana donde se ingresan el nombre de usuario y la contraseña, estas credenciales se ajustan a las descritas anteriormente. Ver Figura B.1

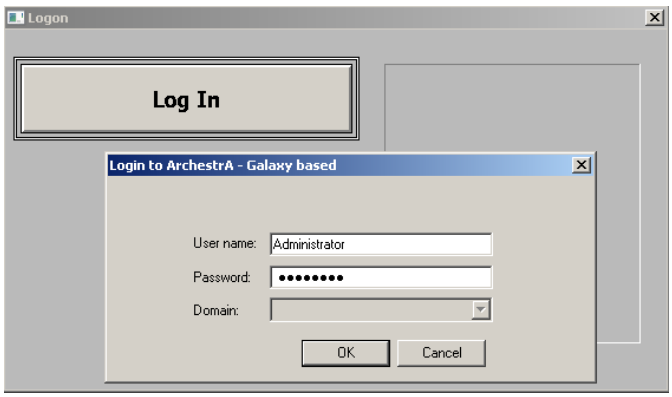


Figura B.1. Inicio Sesión ArchestrA

Una vez ingresado el nombre de usuario, se muestra junto con su Nivel de Acceso, lo que le permite al usuario acceder a la interfaz principal o cerrar la sesión. Ver Figura B.2.

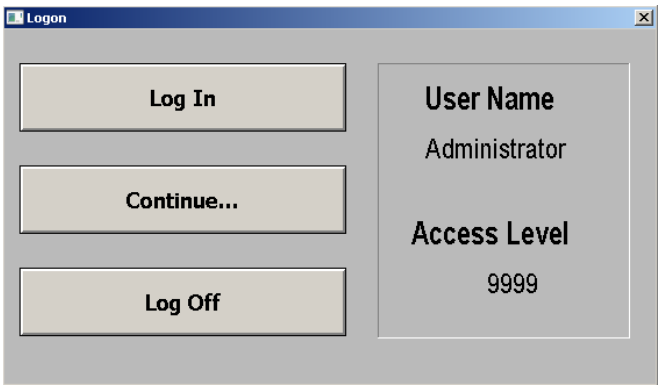


Figura B.2. Usuario Iniciado ArchestrA

Proceso

La pantalla de Proceso muestra un resumen de las secciones principales de la fábrica, permitiendo un análisis general de la operación. También proporciona una visualización de las alarmas principales y permite una navegación sencilla entre las ventanas.

En la vista principal se presentan gráficos simplificados de los dos subprocesos y su interconexión. Para el primer subproceso, se muestran las variables de nivel en los dos tanques y el estado del dispositivo de mezcla. Para el siguiente subproceso, se muestra una visión general del nivel y la temperatura de los tanques, así como el estado del dispositivo de calentamiento. Al tratarse de una ventana de estado, no es posible ingresar comandos ni realizar configuraciones.

Además, se muestran los principales componentes de comunicación con la unidad centralizadora MTU. Por último, se incluye una sección que muestra las alarmas principales, como se puede apreciar en la Figura B.3.

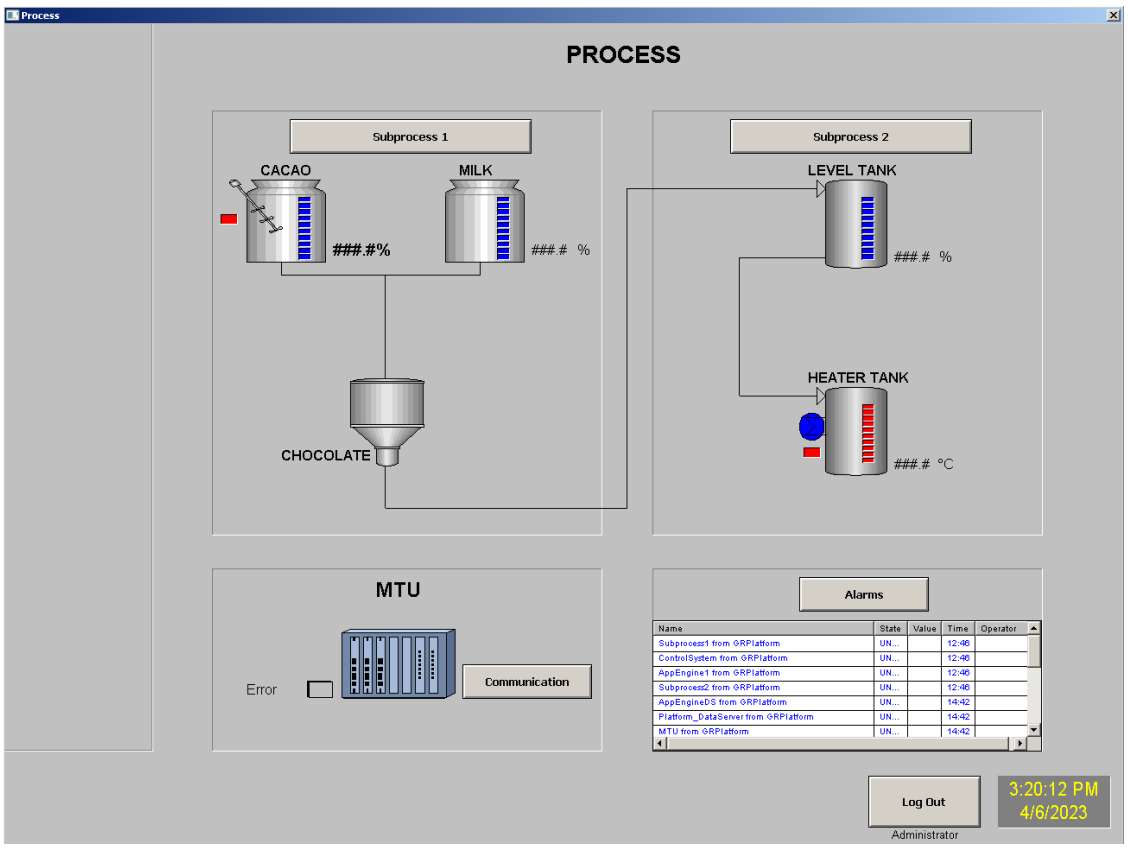


Figura B.3. Ventana Proceso

Subproceso 1

En esta pantalla se muestran específicamente los componentes del subproceso 1. Al ser una ventana de control, el operario puede ingresar comandos para controlar los actuadores. En la sección izquierda de la vista principal se presentan diferentes comandos para la manipulación de las válvulas, que incluyen:

- Apertura y cierre de las válvulas de paso.
- Valor de consigna deseado para el nivel en los tanques.
- Selección del modo manual o automático.

En la sección central y derecha se visualizan los elementos gráficos del subproceso en su totalidad, mostrando cada una de las variables de proceso. Ver Figura B.4.

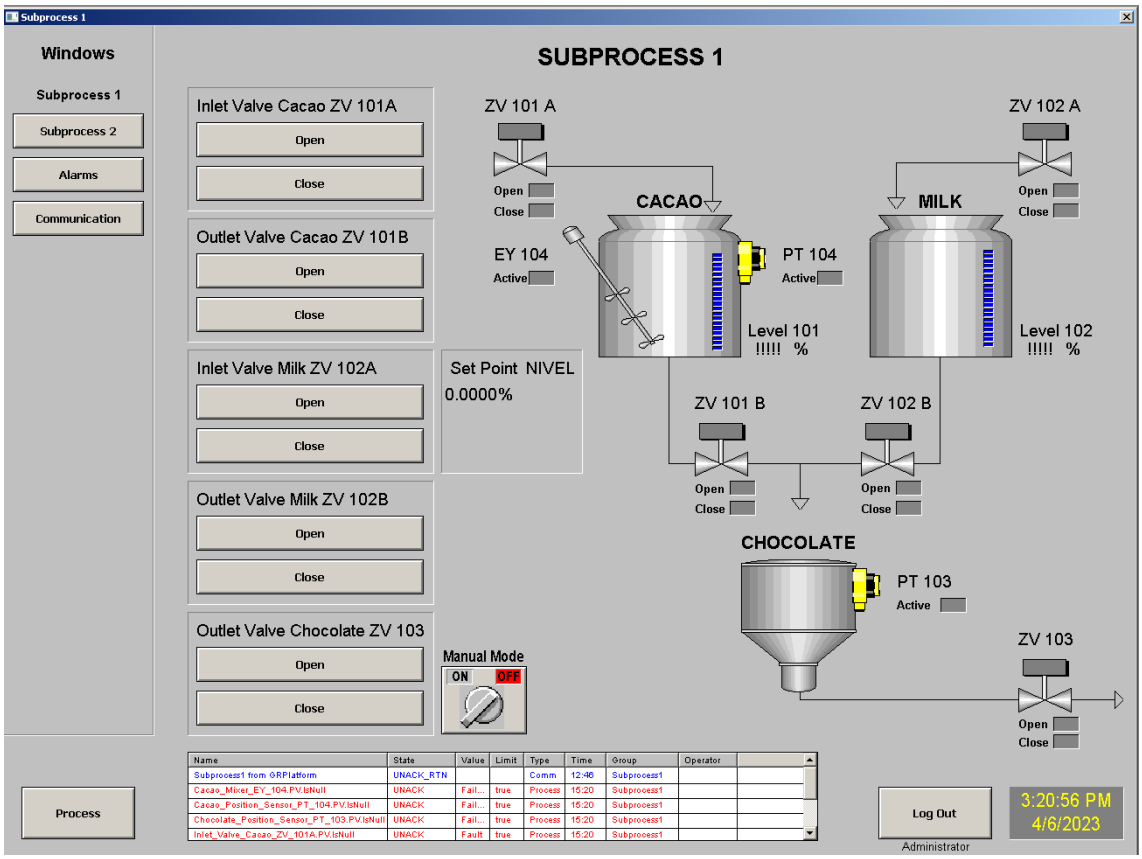


Figura B.4. Ventana Subproceso 1

Subproceso 2

La estructura de la pantalla para el subproceso 2 es muy similar a la del subproceso anterior. En la sección de comandos, se puede controlar adicionalmente la

temperatura del segundo tanque y el posicionamiento de las válvulas. Ver Figura B.5.

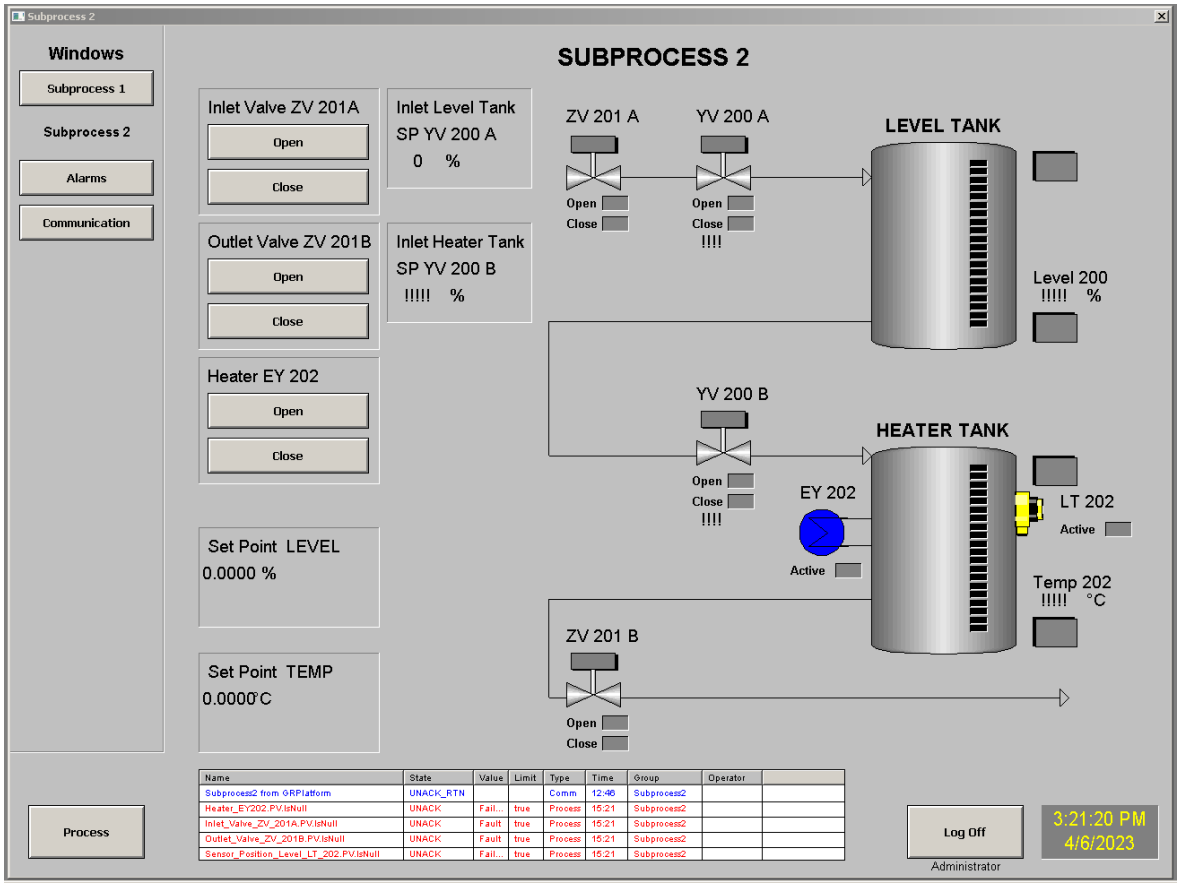


Figura B.5. Ventana Subproceso 2

Alarmas

Esta ventana muestra las alarmas con mayor detalle, proporcionando una vista completa de la tabla. Además, se incluye un conjunto de pulsadores que permiten al usuario manipular las configuraciones y parámetros de las alarmas. Ver Figura B.6.

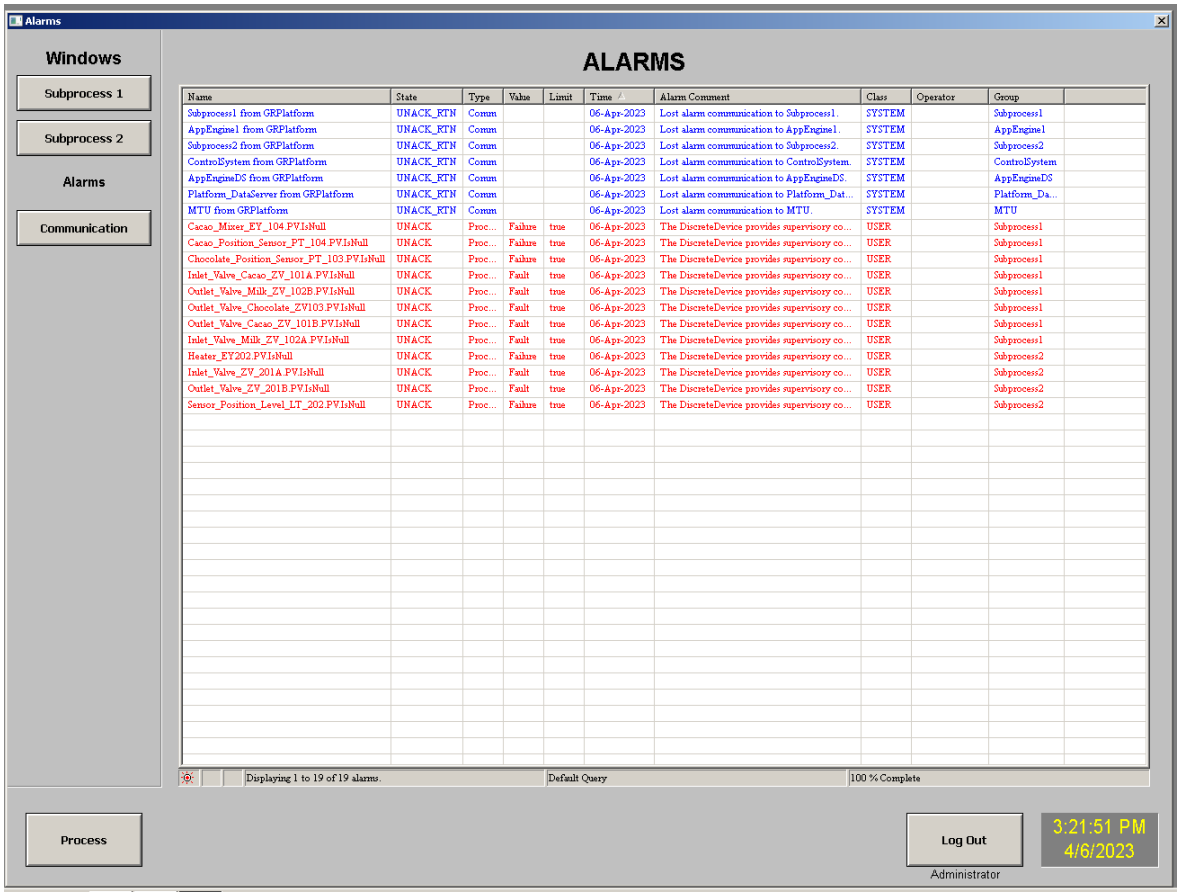


Figura B.6. Ventana Alarmas

Comunicación

En la ventana de comunicación se presentan de manera detallada los parámetros de los dispositivos de comunicación. En el caso del controlador MTU, se pueden observar los estados de error, ejecución y parada, así como el estado de error en caso de que se produzca. En esta sección también se muestran los estados de cada instrucción de comunicación Profinet utilizada (TSEND_C y TRCV_C), incluyendo las variables "*done*", "*busy*", "*error*" y "*state*".

Para las unidades RTU también se disponen datos de comunicación de manera similar al dispositivo anterior. Ver Figura B.7.

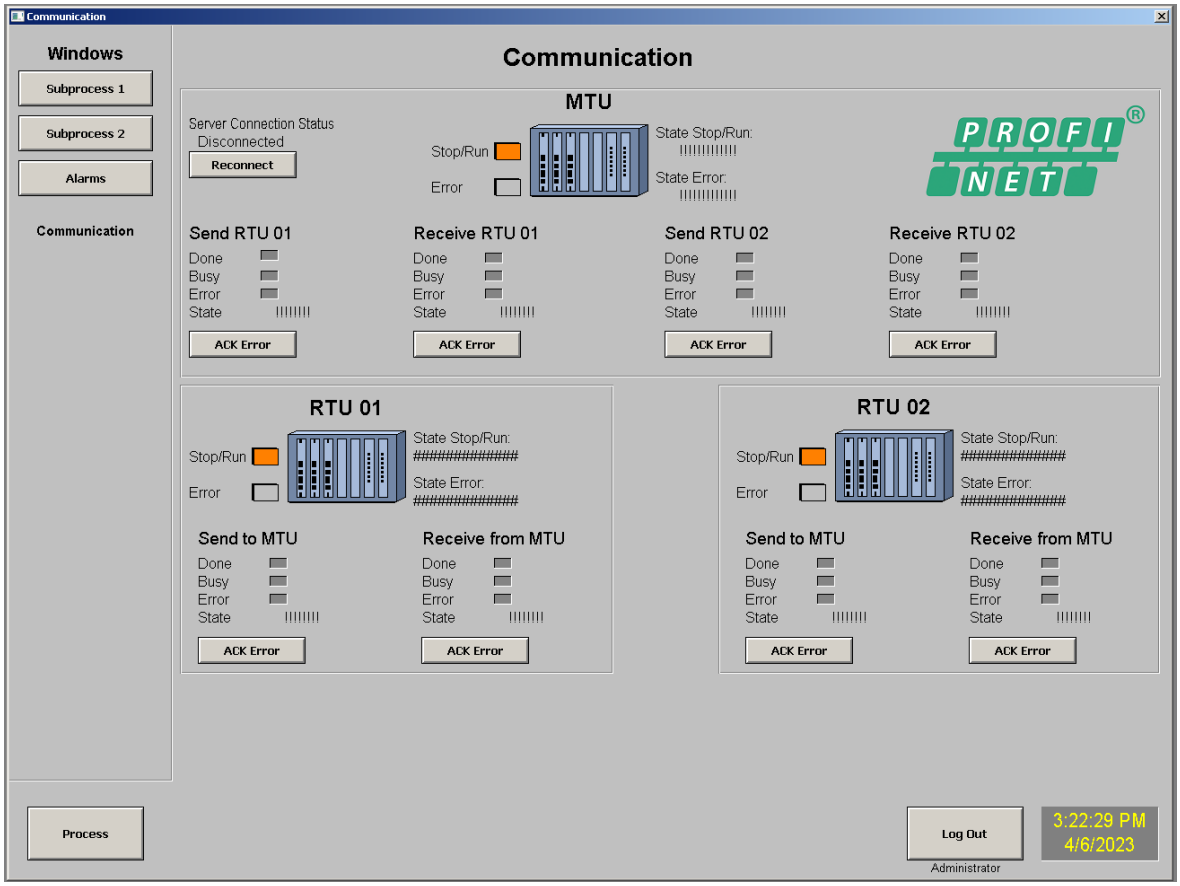


Figura B.7. Ventana Comunicación

ANEXO C

En este anexo se describe una serie de pasos para realizar el intercambio de información mediante una conexión TCP entre dos PLC S7-1200. El intercambio de información se lleva a cabo entre dos CPU denominadas "PLC_01" y "PLC_02", como se muestra en la Tabla C.1 y Tabla C.2.

Tabla C.1. PLC_01

Especificaciones PLC_01	
Nombre	“PLC_01”
Modelo CPU	1214C DC/DC/DC
Versión de Firmware	V2.2
Referencia	6ES7 214-1AE30-0XB0
Dirección IP	192.168.10.2
Másc. Subred	255.255.255.0

Tabla C.2. PLC_02

Especificaciones PLC_02	
Nombre	“PLC_02”
Modelo CPU	1214C AC/DC/Rly
Versión de Firmware	V2.2
Referencia	6ES7 214-1BE30-0XB0
Dirección IP	192.168.10.10
Másc. Subred	255.255.255.0

A continuación, se presenta un instructivo que detalla los pasos para enviar datos desde "PLC_01" a "PLC_02".

Paso 1. Agregar Dispositivos

Al iniciar el proyecto, se deben añadir las dos CPU utilizando la información proporcionada en las tablas anteriores. En el Árbol del proyecto, selecciona "Agregar dispositivo" como se muestra en la Figura C.1. Luego, agrega el segundo CPU siguiendo el mismo procedimiento (Figura C.2).

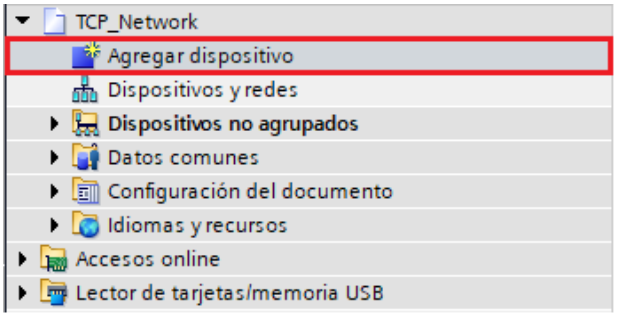


Figura C.1. Agregar dispositivo

Se seleccionan los modelos y la versión de *Frimware* respectiva.

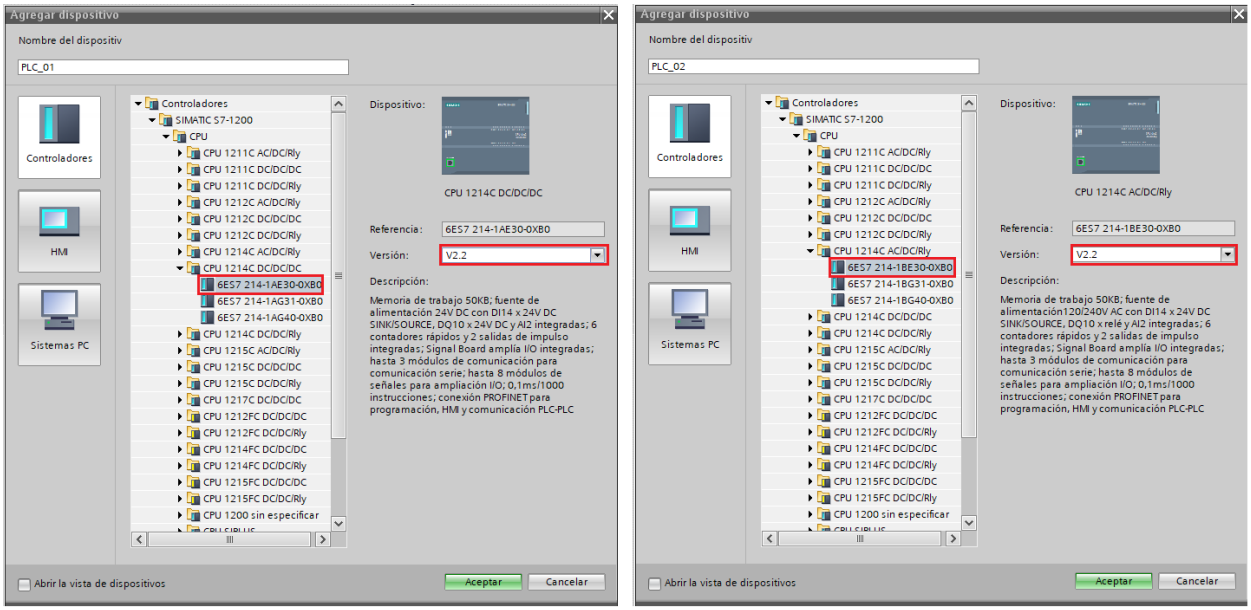


Figura C.2. Selección PLC_01 y PLC_02

Paso 2. Asignación de las Direcciones IP

Procede a asignar la dirección IP y la máscara de subred. En la Configuración de dispositivo se realiza clic derecho en propiedades en la representación de la Interfaz PROFINET del controlador, y se asigna la dirección correspondiente al CPU, como se muestra en la Figura C.3. Repite este proceso para el PLC_02 (Figura C.4).

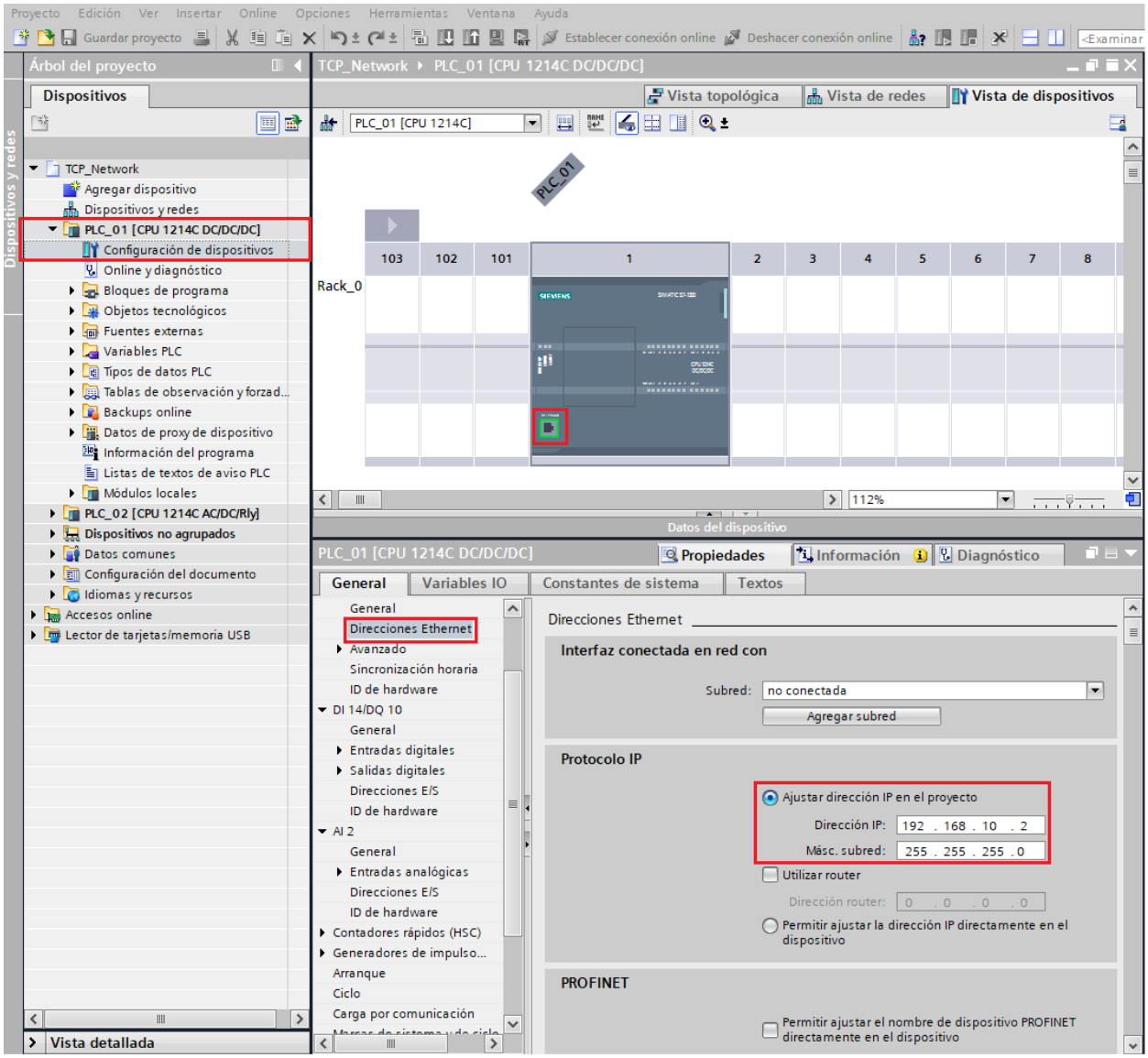


Figura C.3. Dirección IP 192.168.10.2 – PLC_01

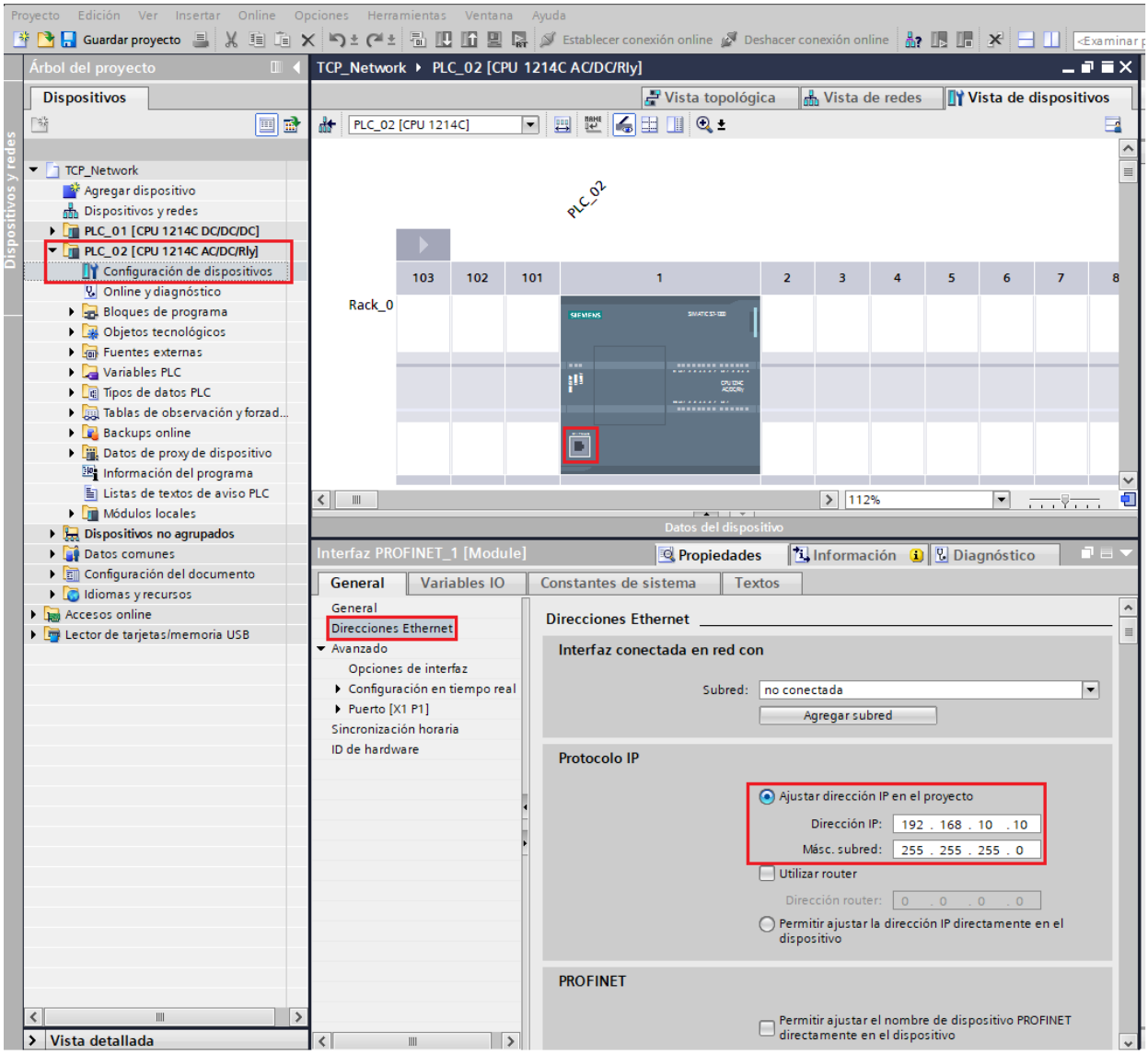


Figura C.4. Dirección IP 192.168.10.10 – PLC_02

Paso 3. Conexión Vista de Redes

En la vista de Dispositivos y redes, se establece la conexión entre las dos interfaces PROFINET (Figura C.5).

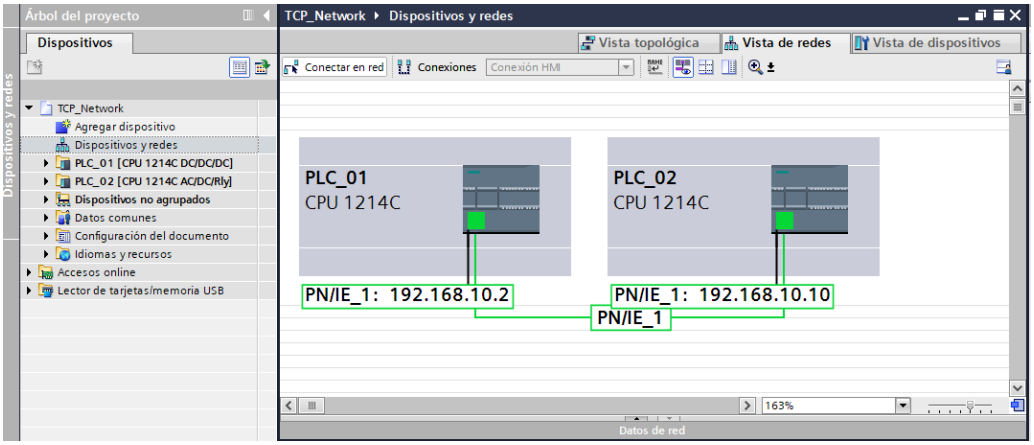


Figura C.5. Conexión PN/IE – Vista de redes

Paso 4. Asignar Instrucciones TSEND_C en PLC_01

Esta instrucción envía el mensaje desde el PLC_01 al PLC_02. La instrucción TSEND_C se ubica en un bloque de programa (por ejemplo, *Main[OB1]*). La instrucción se encuentra en la carpeta "Open user communication" en la pestaña "Comunicación" y la sección de "Instrucciones" (Figura C.6). Al asignar esta instrucción, se te pedirá crear un DB asociado, simplemente haz clic en "Aceptar" (Figura C.7).

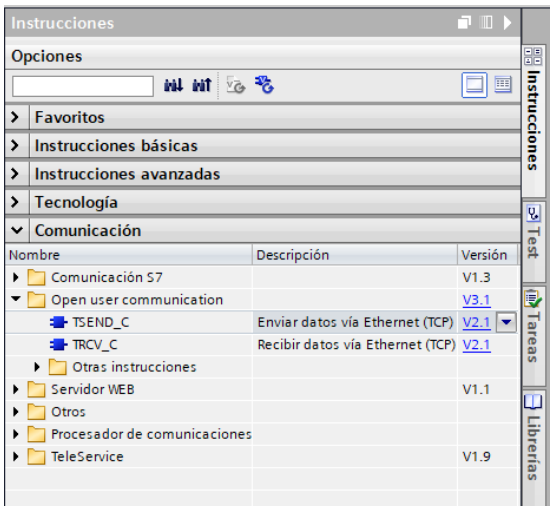


Figura C.6. Ubicación TSEND_C

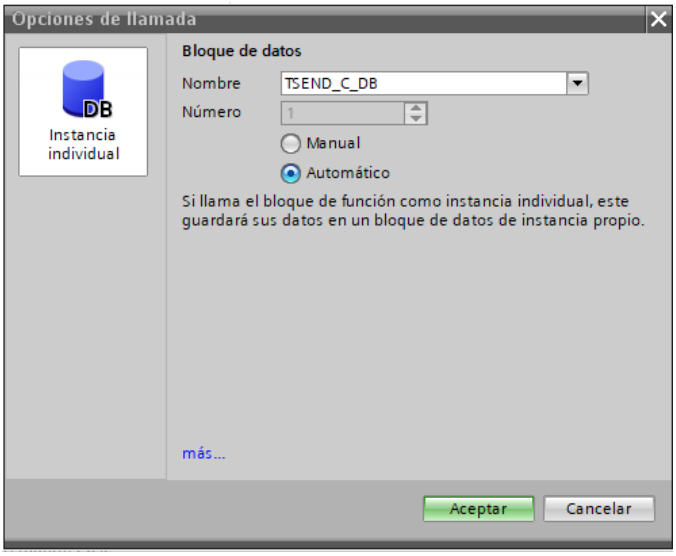


Figura C.7. DB TSEND_C_DB

Paso 5. Configuración TSEND_C en PLC_01

Con el bloque ubicado, realiza las configuraciones haciendo clic en el icono de configuración (Figura C.8).

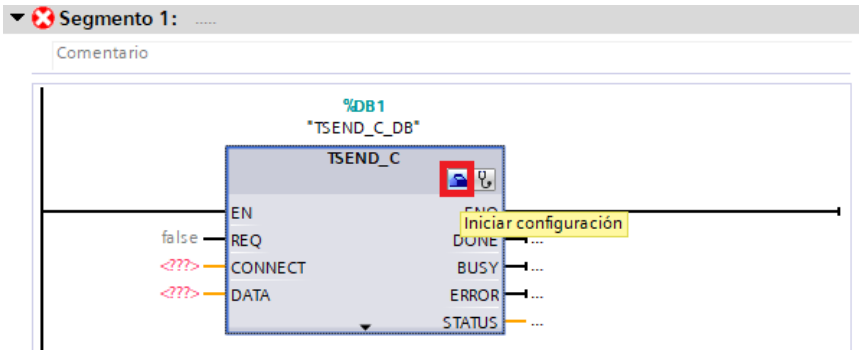


Figura C.8. Ícono Configuración – TSEND_C

En la ventana desplegada, se selecciona el interlocutor (CPU al cual se realizará el envío), en este caso, se selecciona el PLC_02. Detectará que ya existe una red creada (Paso 3) y se mostrarán las direcciones IP correspondientes.

Es necesario crear un DB, el cual contiene las configuraciones de esta instrucción, llamado "PLC_01_Send_DB". También se crea otro DB para los datos de

configuración de la instrucción TRCV_C, llamado "PLC_02_Receive_DB" (Figura C.9).

Otro dato se debe seleccionar el tipo de conexión.

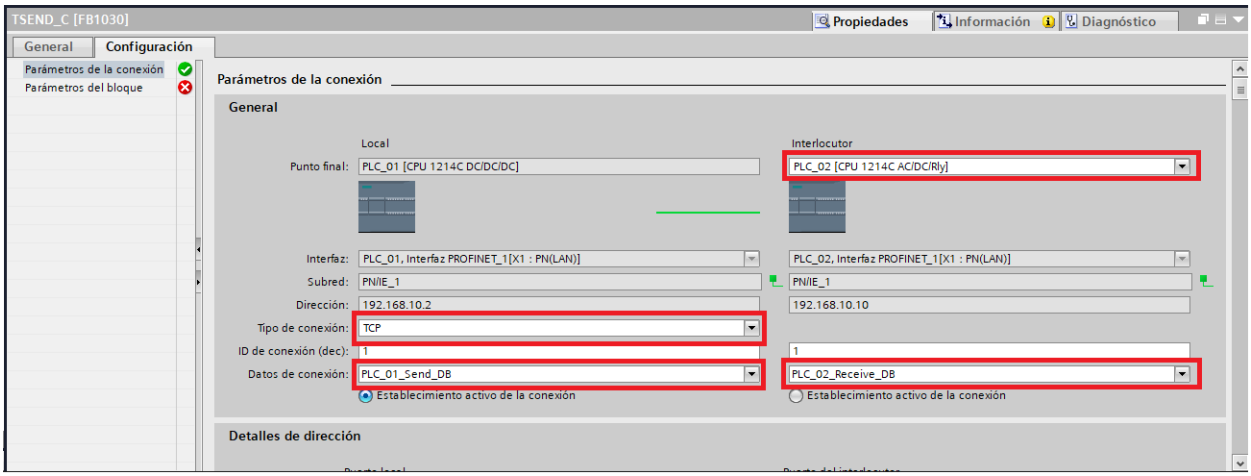


Figura C.9. Configuración – TSEND_C

El DB creado en las configuraciones se agregará automáticamente en el parámetro CONNECT (Figura C.10).

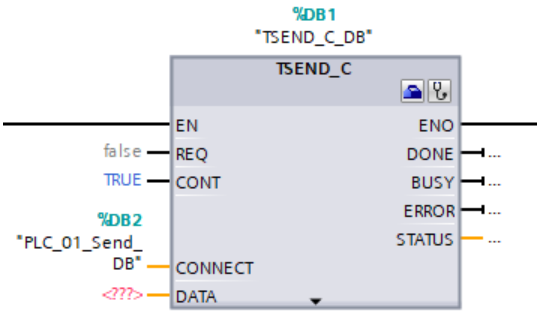


Figura C.10. Agregar “PLC_01_Send_DB” en CONNECT – TSEND PLC_01

Paso 6. Creación DB de Datos en PLC_01

A continuación, se crea un DB que contendrá las señales a ser transmitidas. El DB se lo creará inicialmente en el PLC_01.

Se da clic en “Agregar nuevo bloque” perteneciente al PLC_01, (Figura C.11).

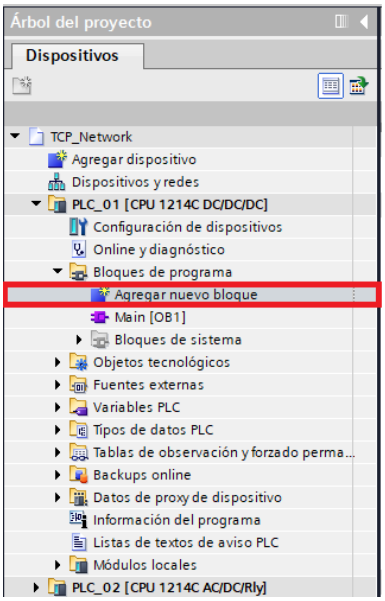


Figura C.11. Agregar nuevo bloque – PLC_01

Se agrega un DB con el nombre "*DB_Data_Send1*" (Figura C.12).

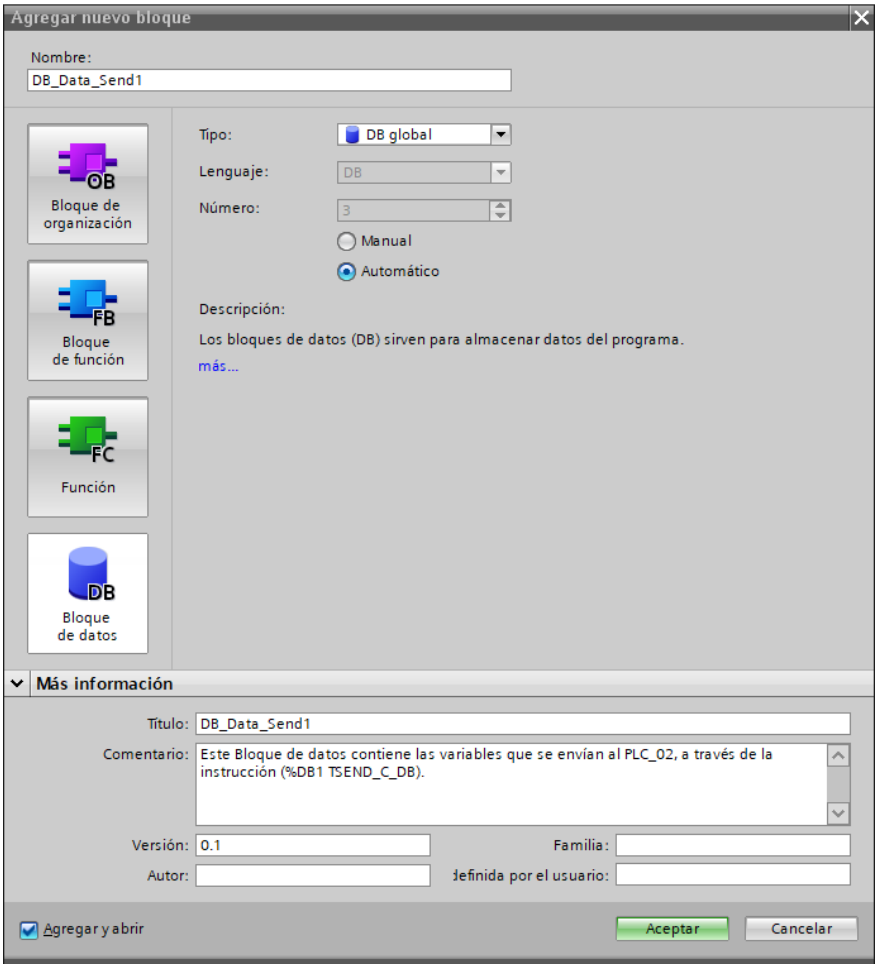


Figura C.12. Creación “DB_Data_Send1” en PLC_01

Paso 7. Crear variables en “DB_Data_Send1”

Dentro de la ventana del DB, se observa una tabla en blanco. En esta tabla se crea una lista de las variables que serán transmitidas. Para este instructivo, se crearán diferentes variables con distintos tipos de datos (Figura C.13).

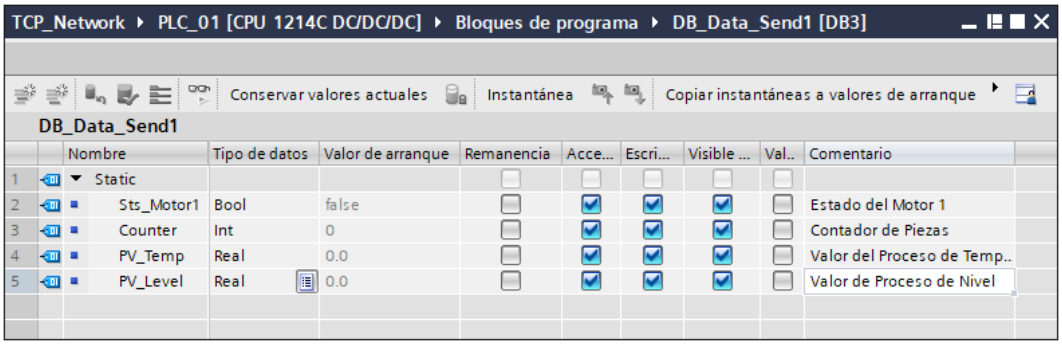


Figura C.13. Estructura “DB_Data_Send1” – PLC_01

Es importante destacar que este DB no debe tener la optimización del bloque habilitada. Para desactivarla, se accede a las propiedades del DB y se desmarca la casilla de verificación "Acceso optimizado al bloque" (Figura C.14).

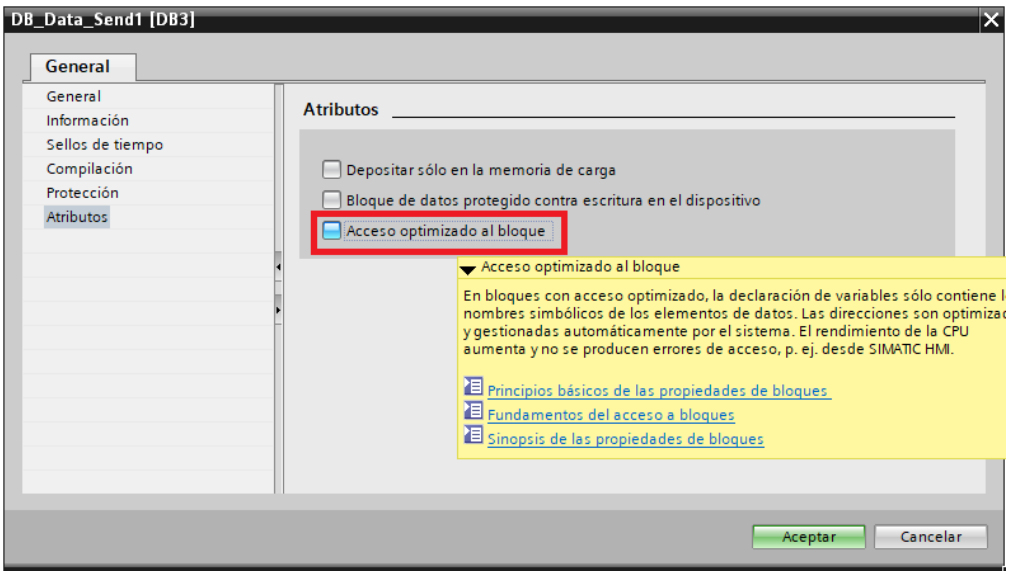


Figura C.14. Acceso optimizado al bloque

A continuación, compila el DB (Figura C.15).

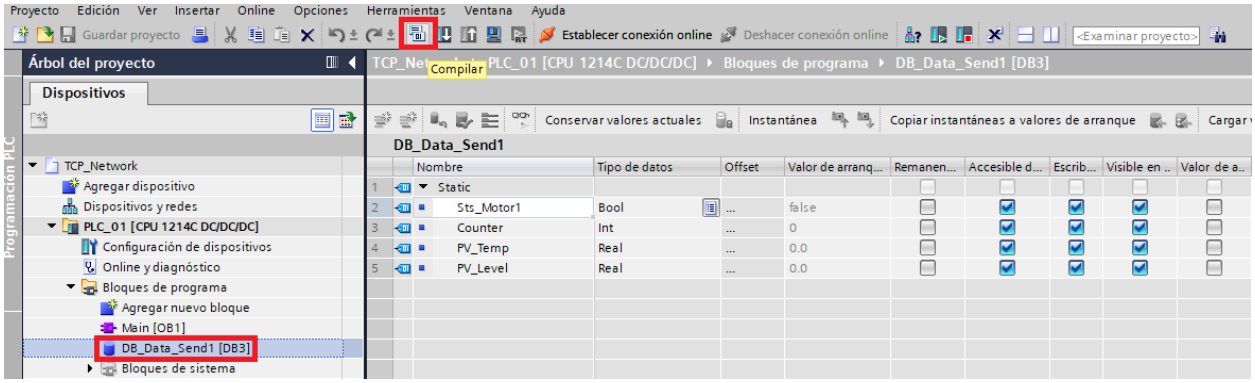


Figura C.15. Compilar – DB_Data_Send1

Una vez configurado el DB, se lo asigna dentro del parámetro DATA de la instrucción "TSEND_C_DB" escribiendo el nombre del DB: "DB_Data_Send1" en el segmento del PLC_01 (Figura C.16).

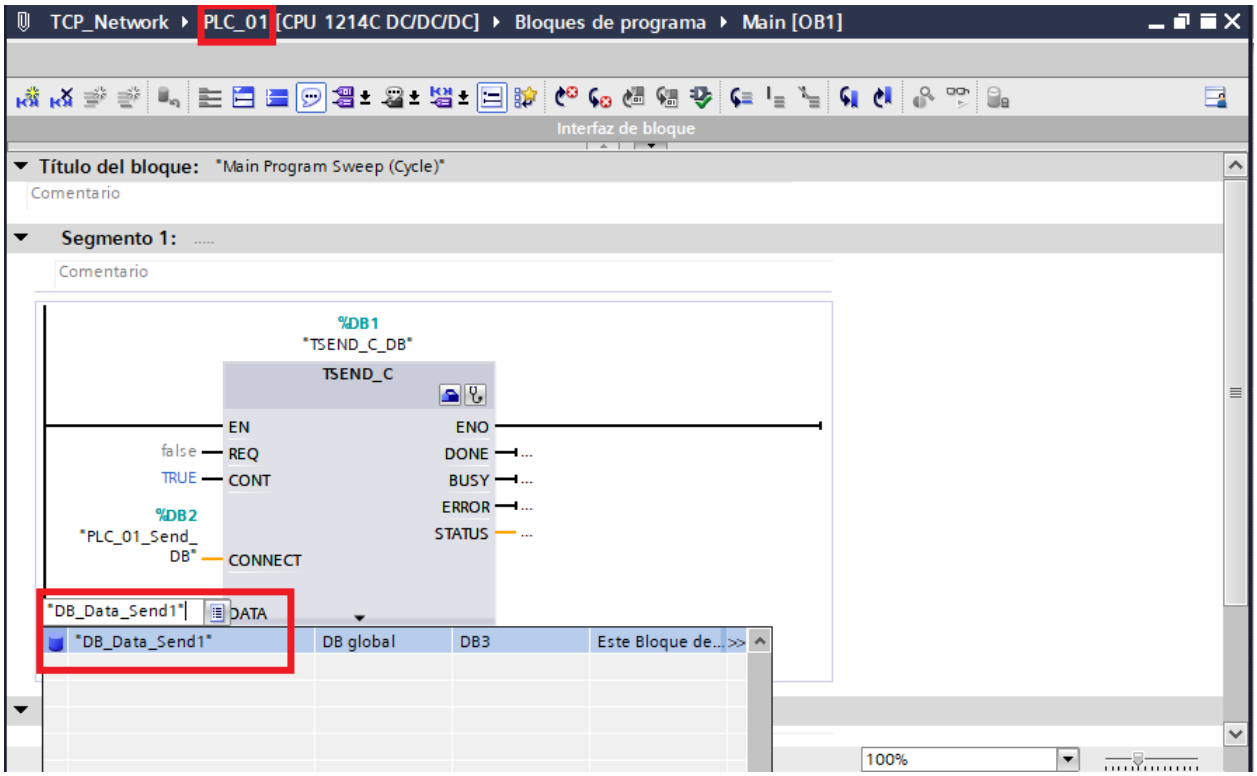


Figura C.16. Parámetro DATA - TSEND

Paso 8. Asignación de estados TSEND_C_DB

Se asigna las variables ubicadas en el DB de configuraciones "TSEND_C_DB" a los parámetros de salida de la instrucción (Figura C.17).

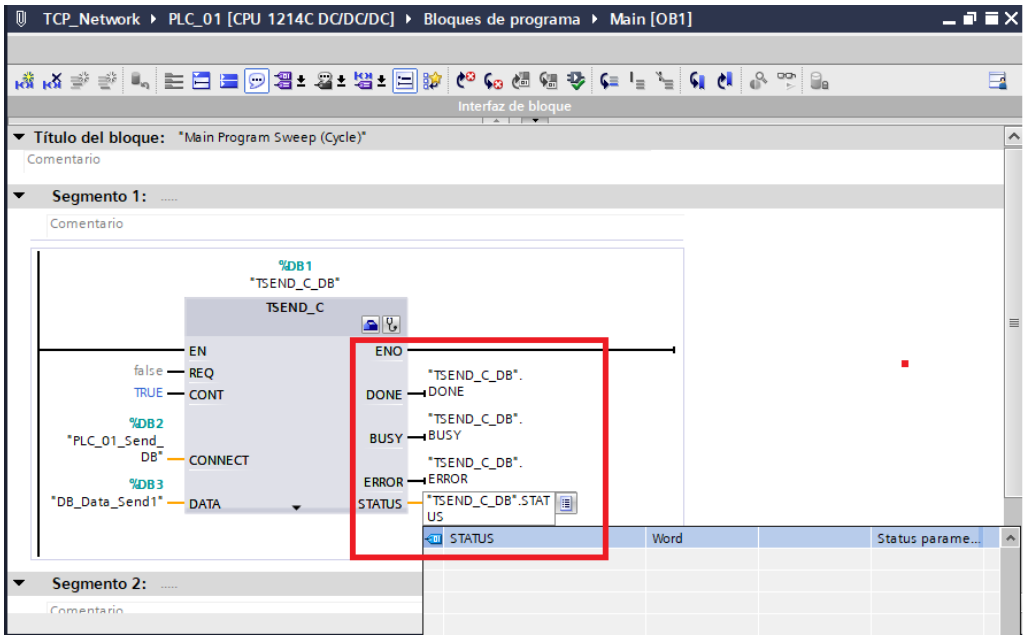


Figura C.17. Parámetros de Salida – TSEND_C

Paso 9. Asignación en Parámetro REQ – TSEND

El parámetro REQ inicia la petición de transmisión cuando detecta un flanco ascendente. Para habilitar esta función, se habilita las variables de marcas de ciclo para realizas este envío.

En las propiedades del PLC_01, en la sección de Maras de sistema y de ciclo, se activa las casillas de verificación “Activar la utilización del byte de marcas de sistema” y “Activar la utilización de byte de marcas de ciclo”, Figura C.18.

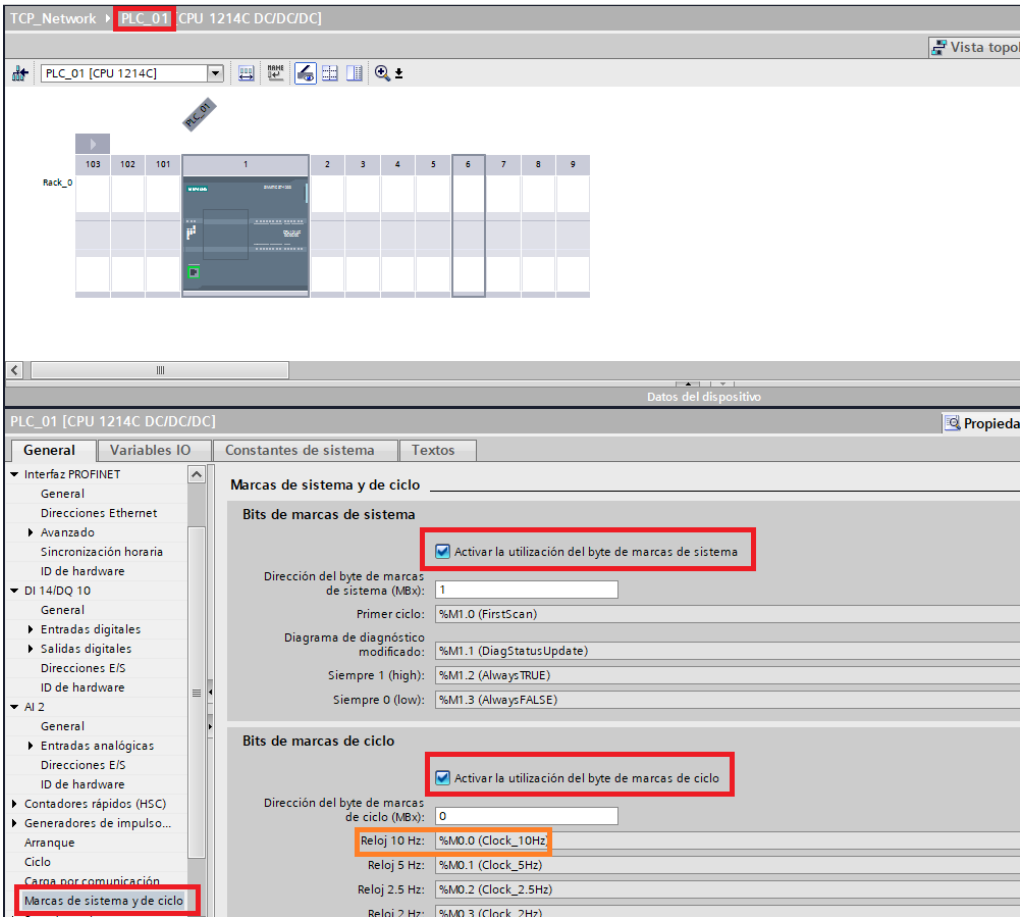


Figura C.18. Activación de bytes de marcas de sistema y de marcas de ciclo – PLC_01

Se visualiza la ubicación del bit que marca el ciclo a 10 Hz y se lo ubica en el parámetro REQ de la instrucción TSEND_C del PLC_01, Figura C.19.

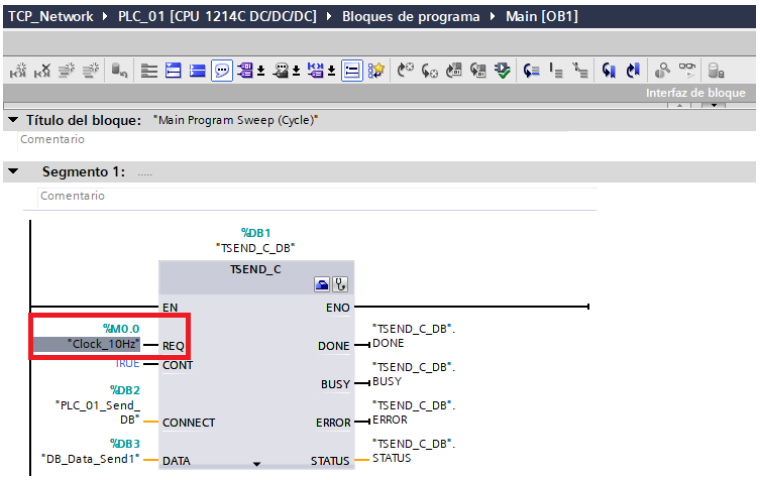


Figura C.19. "Clock_10Hz" en REQ de TSEND_C – PLC_01

Paso 10. Asignar Instrucciones TRCV_C en PLC_02

Esta instrucción establece la conexión, recibe los datos y deshace la conexión. Se ubica la instrucción TRCV_C en el PLC_02. La instrucción se encuentra en la carpeta "Open user communication" en la pestaña "Comunicación" y la sección de "Instrucciones" (Figura C.20).

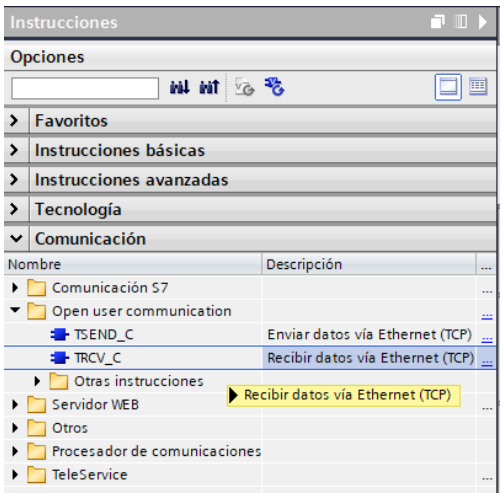


Figura C.20. Ubicación TRCV_C

Al asignar la instrucción, se le pedirá crear un DB asociado, se da clic en "Aceptar" (Figura C.21).

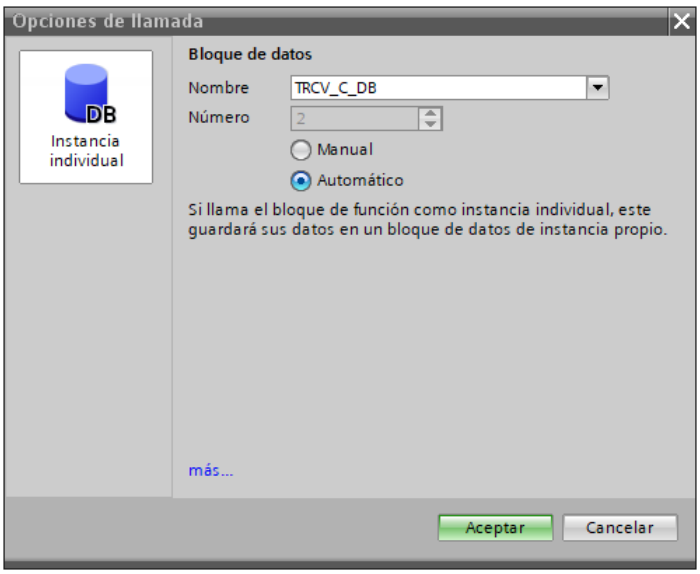


Figura C.21. DB TRCV_C_DB

Paso 11. Configuración TRCV_C en PLC_02

Con la instrucción ubicada en el segmento del PLC_02, se realiza las configuraciones correspondientes (Figura C.22).

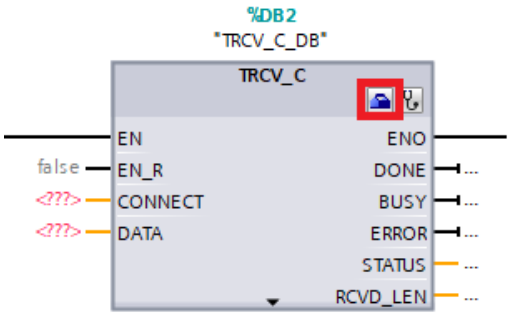


Figura C.22. Ícono Configuración – TRCV_C

En la ventana desplegada, se selecciona el interlocutor de esta conexión (CPU de la cual se recibirán los datos), en este caso, selecciona el PLC_01. Se la red creada previamente (Paso 3) y se mostrará las direcciones IP correspondientes.

Se crear un DB que contenga las configuraciones de esta instrucción, llamado "PLC_02_Receive_DB", y se asignará automáticamente el DB de las configuraciones del interlocutor (Paso 5). También se observa que la instrucción del interlocutor establecerá activamente la conexión (Figura C.23).

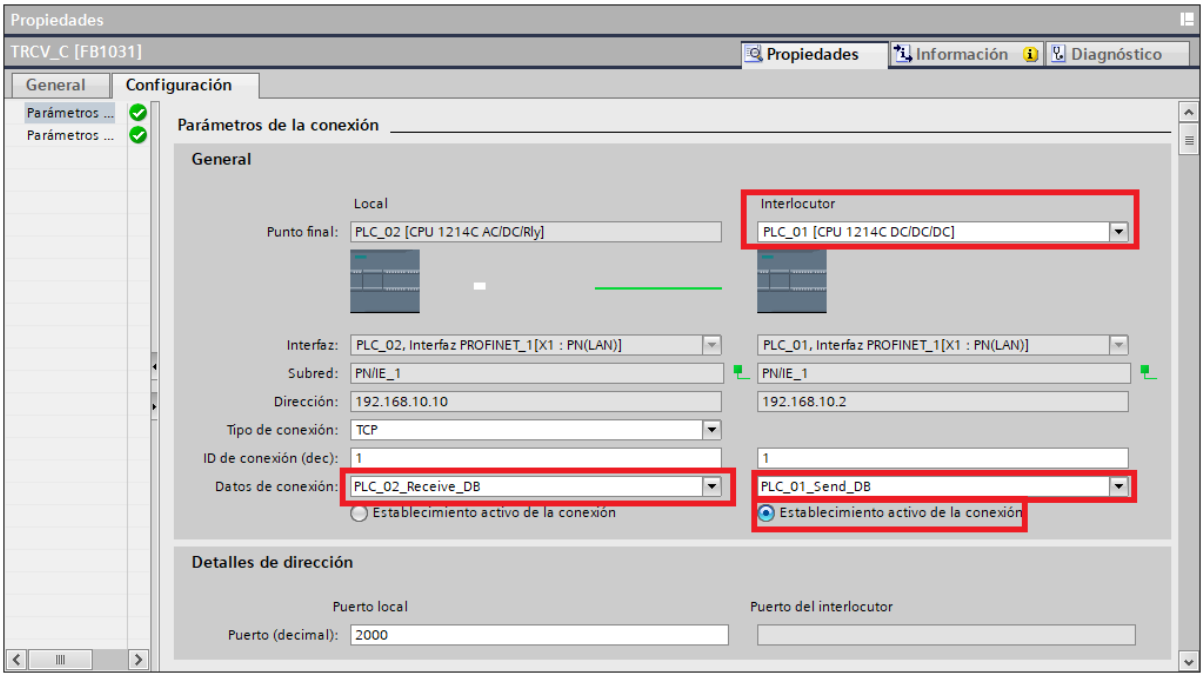


Figura C.23. Configuración – TRCV_C

En el parámetro EN_R de la instrucción TRCV_C, se asigna TRUE (Figura C.24).

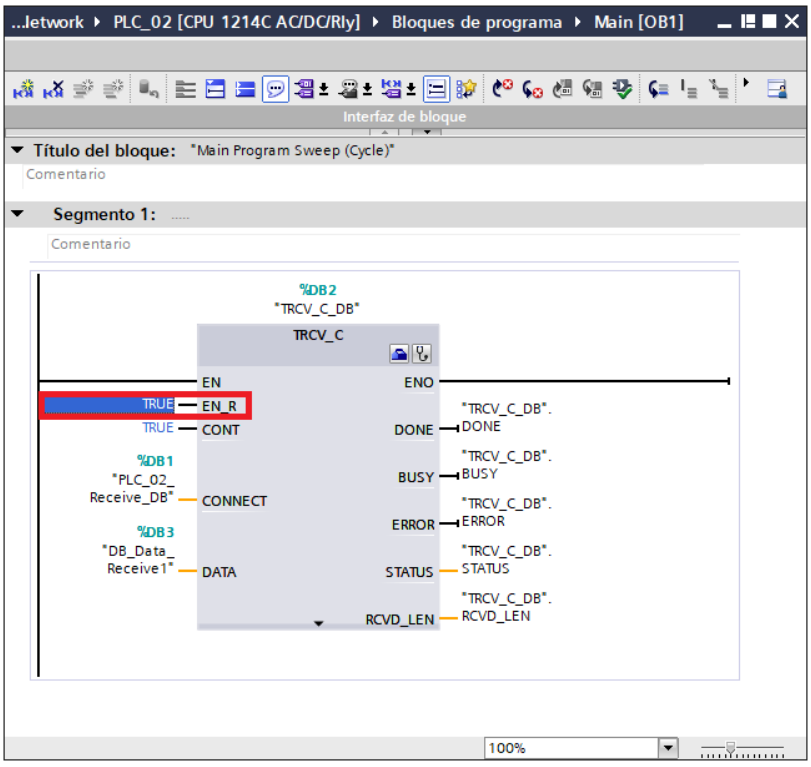


Figura C.24. True en EN_R – TRCV_C

Paso 12. Creación DB de Datos en PLC_02

Se crear un DB que contendrá las variables de la transmisión. Se da clic en "Agregar nuevo bloque" perteneciente al PLC_02 (Figura C.25).

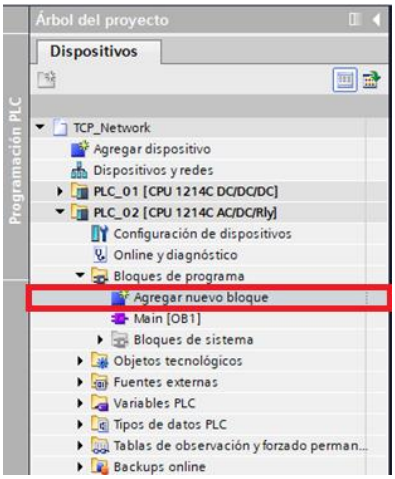


Figura C.25. Agregar nuevo bloque – PLC_02

Luego, se agrega un DB con el nombre "DB_Data_Receive1" (Figura C.26).

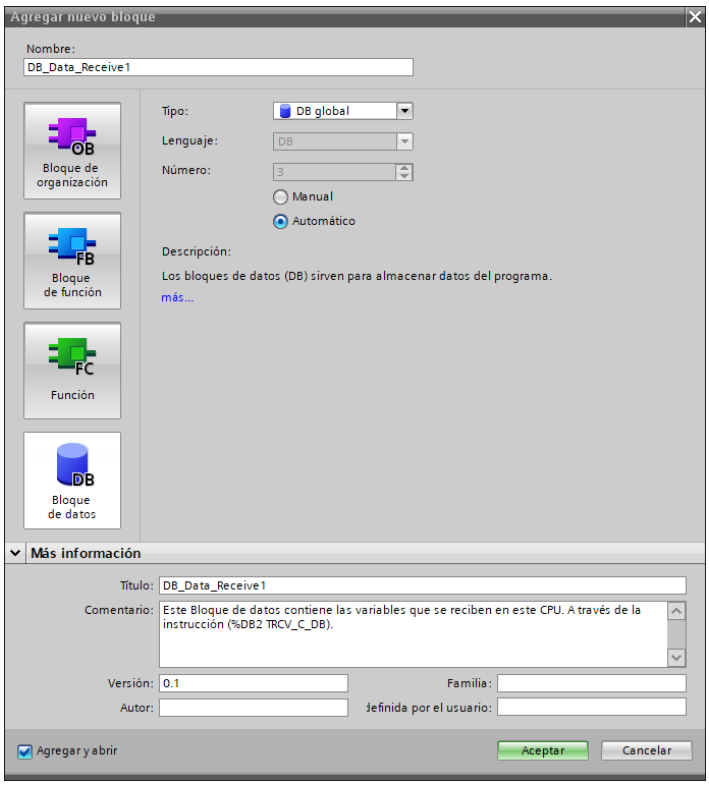


Figura C.26. Creación “DB_Data_Receive1” en PLC_02

Paso 13. Crear variables en “DB_Data_Receive1”

La estructura del "DB_Data_Receive1" debe ser igual a la del "DB_Data_Send1", ya que estas variables son las mismas que serán transmitidas. No es necesario que compartan el mismo nombre, pero sí el tipo de datos (Figura C.27).

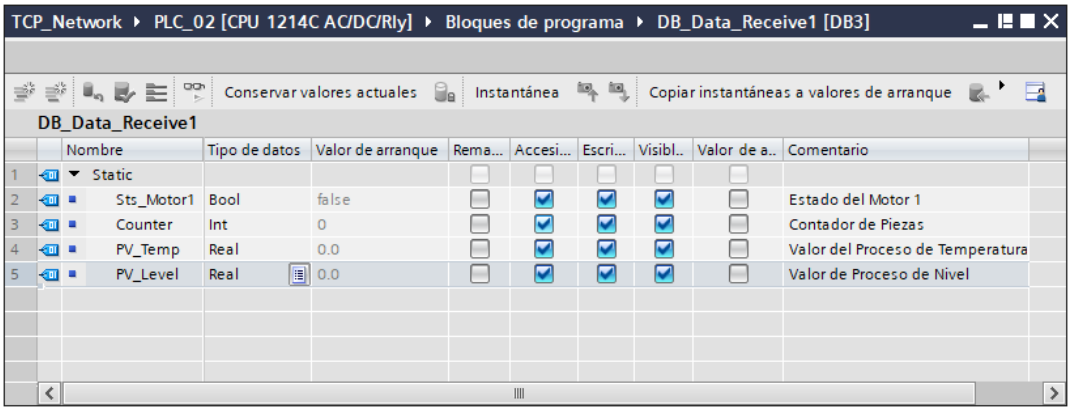


Figura C.27. Estructura “DB_Data_Receive1” – PLC_02

Asimismo, el DB no debe tener la optimización de bloque activada. Por lo tanto, se deshabilita la casilla de verificación "Acceso optimizado al bloque" en las propiedades del DB (Figura C.28).

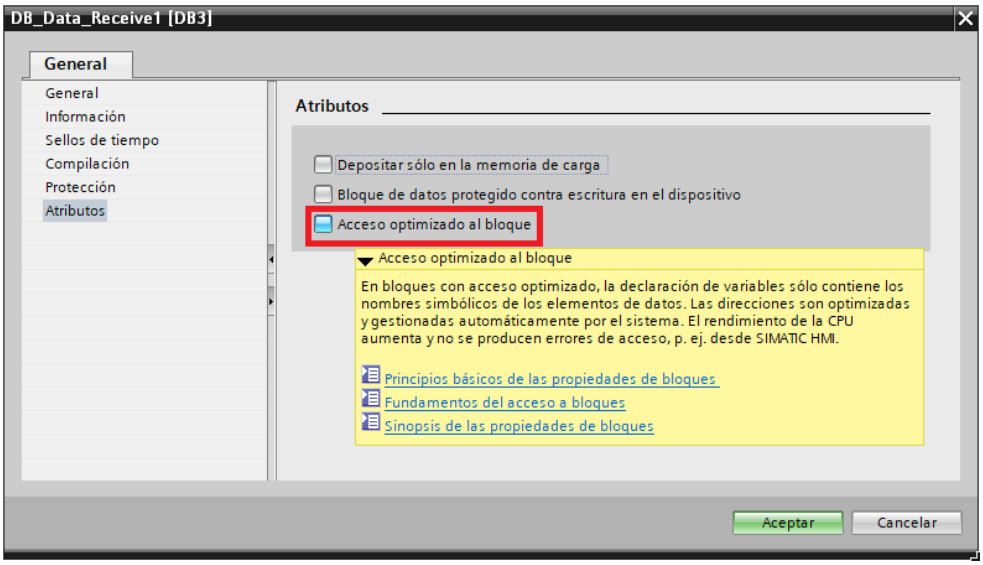


Figura C.28. Acceso optimizado al bloque

Se procede a compilar el DB (Figura C.29).

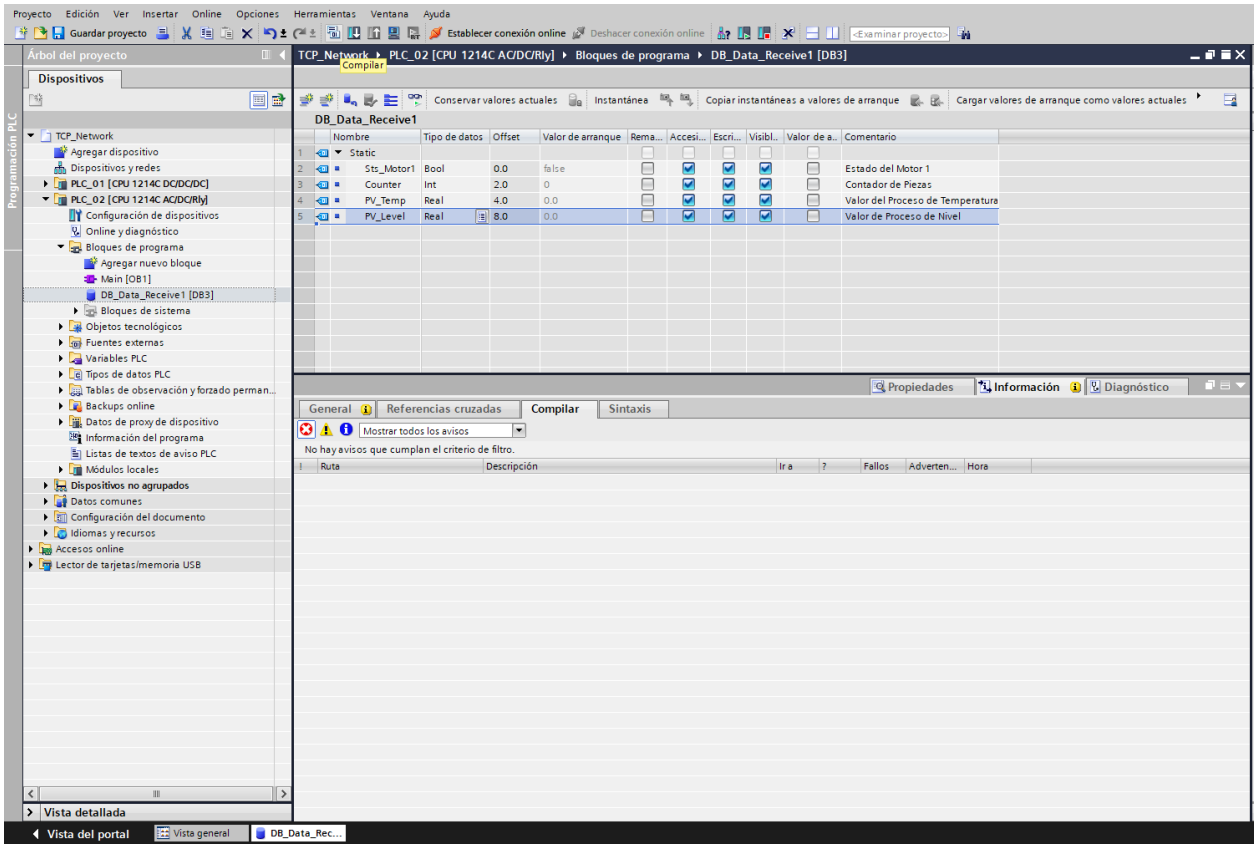


Figura C.29. Compilación - “DB_Data_Receive1”

A continuación, se asigna el DB al parámetro DATA de la instrucción "TRCV_C_DB" escribiendo el nombre del DB: "DB_Data_Receive1" en el segmento del PLC_02 (Figura C.30).

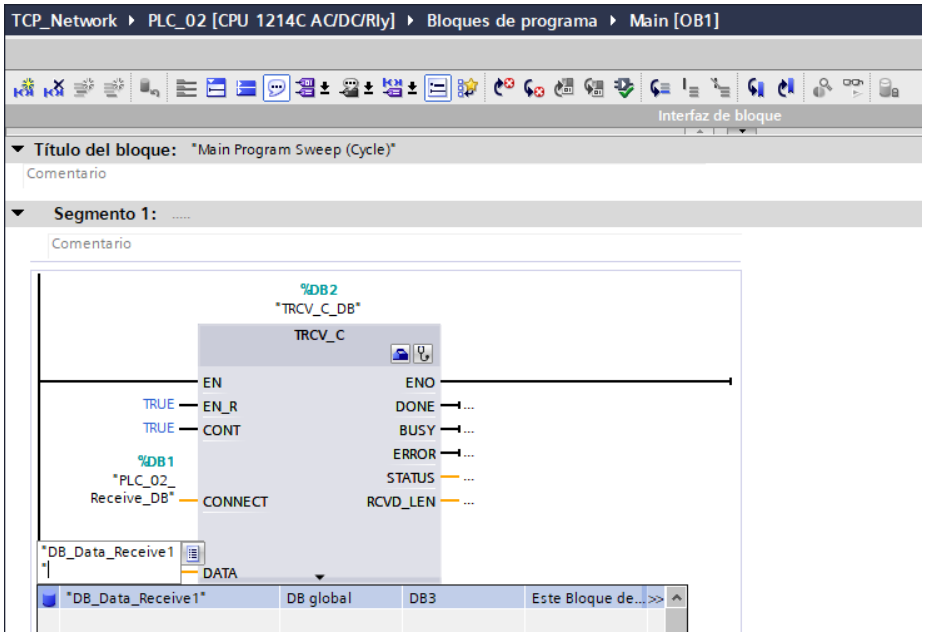


Figura C.30. DB_Data_Receive1 en DATA

Paso 14. Asignación de estados TRCV_C_DB

Se asigna las variables ubicadas en el DB de configuraciones "TRCV_C_DB" a los parámetros de salida de la instrucción (Figura C.31).

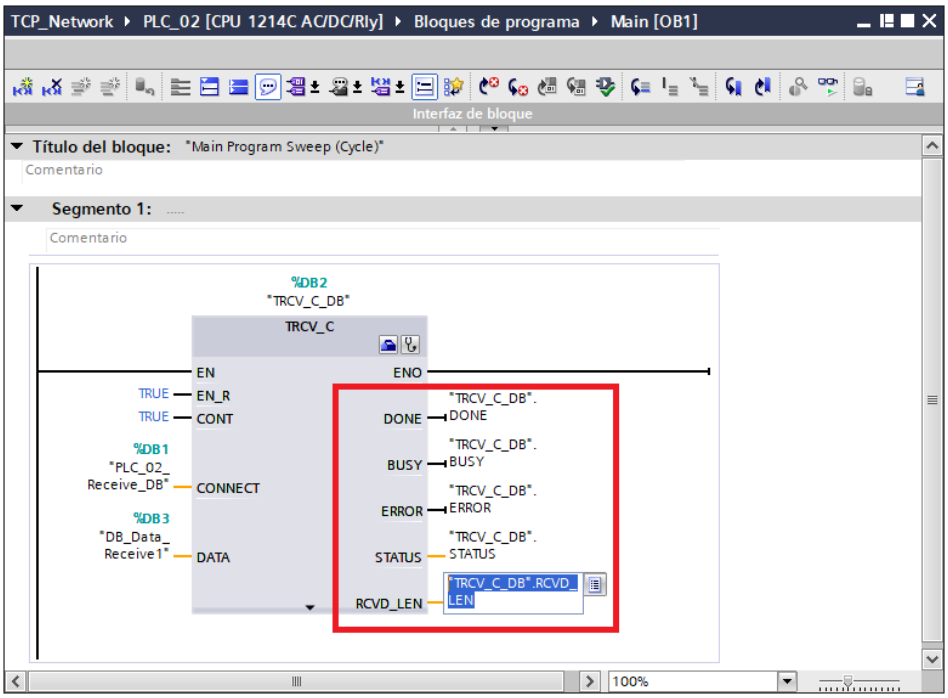


Figura C.31. Parámetros de Salida – TRCV_C

Paso 15. Verificación de la Conexión

Se compila el hardware y el software, y procede a descargar los programas a las CPUs. Se abre los DB que contienen los datos y modifica las variables para verificar que se reflejen en el DB de destino (Figuras C.32 y C.33).

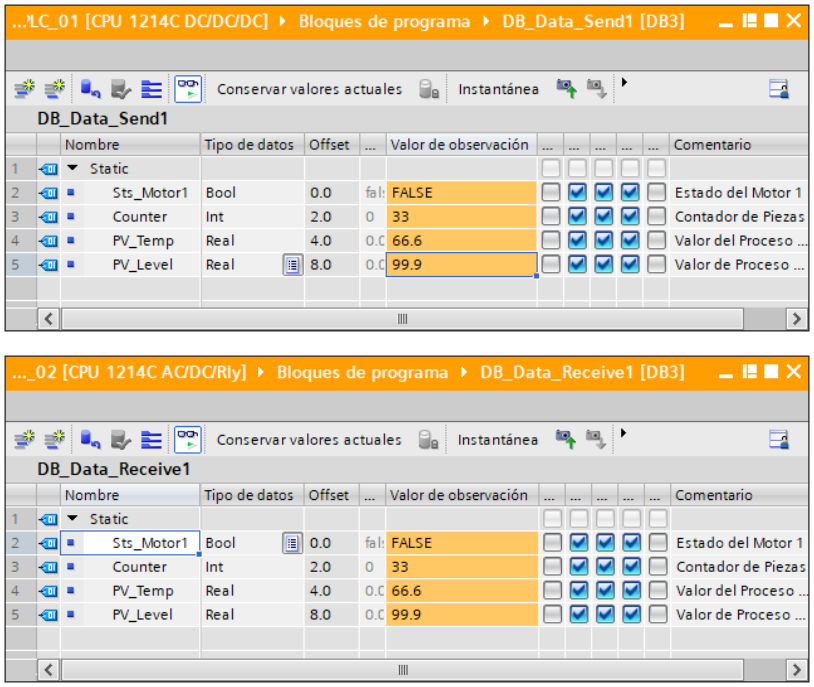


Figura C.32. Verificación 1

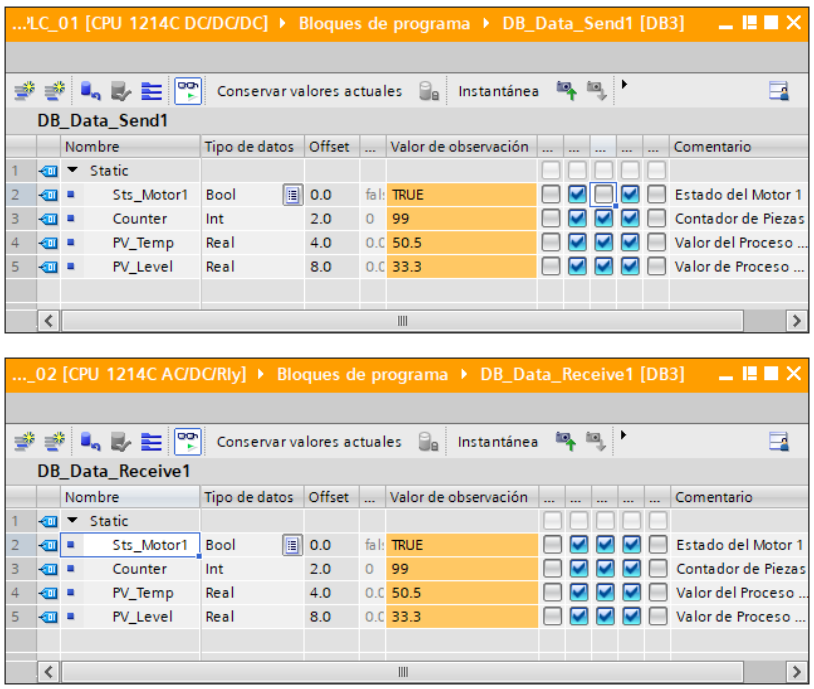


Figura C.33. Verificación 2