

**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA**  
**SEDE VIÑA DEL MAR – JOSÉ MIGUEL CARRERA**

**SIMULACIÓN DE SISTEMA DE CONTROL CON PLC AC500 E INTERFAZ GRÁFICA**

Trabajo de Titulación para optar al Título de  
Ingeniero de Ejecución en CONTROL E  
INSTRUMENTACIÓN INDUSTRIAL

Alumno:

Cristopher Mitchell Monsalve

Profesor Guía:

Mag. Loreto Marín Carcey

2025

## **RESUMEN**

**KEYWORDS:** PLC, HMI, Control, Simulacion

El proyecto desarrollado tiene como objetivo la simulación de un sistema de control industrial utilizando tecnología ABB, específicamente el PLC AC500 y la HMI CP600, junto con herramientas como Automation Builder, Panel Builder 600 y KEPServer. El sistema automatiza el control de nivel y temperatura en un tanque de mezcla, integrando sensores, actuadores y protocolos de comunicación como Profinet e IEC104.

El desarrollo incluyó la configuración del hardware y software, programación del PLC y diseño de la interfaz HMI. Se estableció comunicación eficiente entre los dispositivos, garantizando la supervisión y el control del proceso en tiempo real. Las pantallas de la HMI permiten a los operadores monitorear variables clave, gestionar alarmas y realizar ajustes en el sistema de forma intuitiva.

En las pruebas realizadas, el sistema mostró una operación confiable, activando alarmas ante condiciones críticas y permitiendo la recuperación segura tras eventos simulados. La integración de controladores PID permitió mantener las variables del proceso dentro de los rangos establecidos, optimizando la operación.

Los resultados evidenciaron que la tecnología ABB es competitiva frente a otras marcas líderes, destacándose por su flexibilidad y facilidad de implementación. Este proyecto sienta las bases para la aplicación de soluciones similares en la industria, mejorando la productividad, la seguridad y la eficiencia operativa.

## ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1: ANTECEDENTES GENERALES .....</b>	<b>2</b>
1 ANTECEDENTES GENERALES .....	3
1.1 REQUERIMIENTOS.....	3
1.1.1 Requerimientos de hardware .....	3
1.1.2 Requerimientos de software .....	4
1.1.3 Otros requerimientos .....	5
1.2 DESCRIPCIÓN DE HARDWARE Y SOFTWARE .....	5
1.2.1 Controlador lógico programable (PLC) AC500.....	5
1.2.2 Interfaz Hombre – Máquina (HMI) .....	6
1.2.3 Automation Builder .....	7
1.2.4 Panel Builder .....	8
1.2.5 KEP Server .....	9
1.3 DESCRIPCIÓN DEL PROCESO .....	10
1.3.1 Entradas analógicas .....	11
1.3.2 Entradas digitales.....	12
1.3.3 Funcionamiento del proceso .....	13
1.3.4 Comunicación de datos del sistema .....	13
1.3.5 Módulo de comunicación .....	14
1.3.6 Módulos de entradas y salidas .....	15
1.4 OBJETIVOS DEL PROYECTO .....	16
1.4.1 Objetivo general .....	17
1.4.2 Objetivos específicos .....	17
<b>CAPÍTULO 2: DESARROLLO DEL PROYECTO .....</b>	<b>18</b>
2 DESARROLLO DEL PROYECTO .....	19
2.1 AUTOMATION BUILDER .....	19

2.1.1	Creación del programa .....	20
2.1.2	Creación del hardware.....	23
2.1.3	Asignación de canales IO .....	29
2.1.4	Asignación de librerías.....	32
2.1.5	Creación de lógica de control .....	37
2.1.6	Configuración de alarmas .....	46
2.1.7	Configuración de símbolos .....	48
2.1.8	Comunicación de datos .....	51
2.2	PANEL 600 .....	61
2.2.1	Creación de Proyecto.....	61
2.2.2	Configuración de protocolo de comunicación con PLC AC500 .....	64
2.2.3	Asignación de variables Panel .....	65
2.2.4	Creación de usuarios y accesos .....	67
2.2.5	Pantallas de operación .....	68
2.2.6	KepServer.....	76
<b>CAPÍTULO 3: ANÁLISIS DE RESULTADOS.....</b>		<b>80</b>
3	ANÁLISIS DE RESULTADOS.....	81
3.1	PUESTA EN MARCHA DE LA PLANTA.....	81
3.1.1	Verificación del Hardware y Software .....	82
3.1.2	Pruebas de Comunicación .....	83
3.1.3	Ajustes de Configuración .....	84
3.1.4	Operación Inicial .....	84
3.2	RESPUESTA DEL SISTEMA.....	85
3.2.1	Respuesta en condiciones normales .....	85
3.2.2	Respuesta en Condiciones Críticas .....	86
3.2.3	Evaluación de Desempeño .....	90
<b>CONCLUSIONES.....</b>		<b>91</b>
<b>BIBLIOGRAFÍA .....</b>		<b>92</b>

## ÍNDICE DE FIGURAS

Figura 1-1 Controlador PM5630.....	6
Figura 1-2 Interfaz Hombre – Máquina CP6607.....	7
Figura 1-3 P&ID del Proceso.....	11
Figura 1-4 – Módulo de comunicación CM579-PNIO.....	14
Figura 1-5 Módulo CI502.....	15
Figura 1-6 Módulo DA501.....	16
Figura 2-1 Automation builder 2.7.....	20
Figura 2-2 Automation Builder 2.7, nuevo proyecto.....	20
Figura 2-3 Nuevo proyecto “Proyecto AC500”.....	21
Figura 2-4 Selección modelo AC500 PM5630-2ETH.....	21
Figura 2-5 Hardware del proyecto.....	22
Figura 2-6 Agregar objeto DA501.....	23
Figura 2-7 Hardware del proyecto, incluyendo DA501.....	24
Figura 2-8 Agregar objeto PROFINET IO CM579-PNIO.....	25
Figura 2-9 Hardware del proyecto, incluyendo CM579-PNIO.....	26
Figura 2-10 Agregar objeto CI502-PNIO-Device.....	27
Figura 2-11 Hardware del proyecto, incluyendo CI502.....	28
Figura 2-12 DA501 Configuración.....	29
Figura 2-13 Asignación de variables módulo DA501.....	30
Figura 2-14 Asignación de variables módulo CI502.....	31
Figura 2-15 Proyecto - Administrador de bibliotecas.....	32
Figura 2-16 Administrador de bibliotecas.....	33
Figura 2-17 PCO, 0.1.0.6.....	34
Figura 2-18 Macro PcoMot.....	34
Figura 2-19 Macro PcoValv.....	35
Figura 2-20 Util, 3.5.19.0.....	36
Figura 2-21 Macro PID_FIXCYCLE.....	36
Figura 2-22 Carpeta LÓGICA.....	37

Figura 2-23 TQ-03.....	39
Figura 2-24 TT-01.....	39
Figura 2-25 Variables digitales de entrada.....	40
Figura 2-26 Bloque PID a utilizar. ....	42
Figura 2-27 LIC-01.....	44
Figura 2-28 Alarm_Configuration.....	46
Figura 2-29 Alarm_Configuration, grupos de alarmas.....	47
Figura 2.30 AlarmGroup .....	47
Figura 2-31 Agregar objeto - Configuración de símbolos. ....	49
Figura 2-32 Configuración de símbolos.....	50
Figura 2-33 Configuración de símbolos, seleccionar GVL. ....	51
Figura 2-34 Extension_Bus.....	52
Figura 2-35 PNIO_Controller .....	53
Figura 2-36 CI502_PNIO_Device. ....	54
Figura 2-37 Bloque de función IEC60870_SEND_M_ME_16 .....	55
Figura 2-38 Bloque de función IEC60870_SEND_M_SP_16.....	56
Figura 2-39 Ethernet AC500.....	57
Figura 2-40 IEC 60870-5-104 Subestación. ....	57
Figura 2-41 Capa de enlace IEC104 .....	58
Figura 2-42 Objetos de Información. ....	58
Figura 2-43 Mapeo de señales. ....	59
Figura 2-44 IEC60870_5_104_Command_GVL .....	60
Figura 2-45 IEC60870_5_104_Connection_GVL .....	61
Figura 2-46 Creación del proyecto.....	62
Figura 2-47 Selección Panel HMI.....	62
Figura 2-48 Herramientas del Proyecto .....	63
Figura 2.49 CODESYS V3 ETH.....	64
Figura 2.50 Protocols Panel Builder 600. ....	65
Figura 2-51 Número de Tags agregados. ....	65

Figura 2-52 Variables agregadas a Panel Builder .....	66
Figura 2-53 Grupos de usuarios .....	67
Figura 2-54 Crear usuarios .....	68
Figura 2-55 Inicio de sesión.....	69
Figura 2-56 Proceso.....	70
Figura 2-57 Historial de alarmas. ....	71
Figura 2-58 Audit Trail.....	72
Figura 2-59 History Trends .....	73
Figura 2-60 Tendencias Tanques.....	74
Figura 2-61 Tendencias Válvulas. ....	74
Figura 2-62 Tendencias bombas.....	75
Figura 2-63 Menú. ....	76
Figura 2-64 Protocolo de comunicación. ....	77
Figura 2-65 Direccionamiento comunicación .....	78
Figura 2-66 Dirección común. ....	78
Figura 2-67 Tabla de variables. ....	79
Figura 3-1 Componentes del proyecto.....	81
Figura 3-2 PLC AC500. ....	82
Figura 3-3 Router Wi-Fi .....	83
Figura 3-4 Operación inicial.....	85
Figura 3-5 Proceso en marcha.....	86
Figura 3-6 alarmas en válvulas. ....	87
Figura 3-7 Alarmas en bombas.....	88
Figura 3-8 Parada de emergencia. ....	89

## SIGLAS Y SIMBOLOGÍA

### **A. SIGLAS:**

CFC	:Continuous Function Chart
FBD	: Diagrama de Bloques Funcionales
LD	: Diagrama de Contactos
CFC	: Diagrama de Funciones Continuas
SFC	: Diagrama Funcional Secuencial
E/S	: Entrada/Salida
HMI	: Interfaz Hombre-Máquina.
IP	: Internet Protocol
LabVIEW	: Laboratory Virtual Instrument Engineering Workbench.
NI	: National Instrument
PLC	: Controlador lógico programable
TCP	: Transmission Control Protocol
VI	: Virtual Instrument

### **B. SIMBOLOGÍA:**

V	: Volts.
%	: Por ciento.
cms	: Centímetros
s	: Segundos.
b	: Bit
°C	:Grados celsius

## **INTRODUCCIÓN**

La automatización industrial se ha consolidado como un elemento esencial en el panorama moderno, siendo un catalizador clave para mejorar la eficiencia operativa, incrementar la productividad y reducir costos en un entorno caracterizado por la constante evolución tecnológica y la alta competitividad. La implementación de sistemas automatizados no solo permite optimizar procesos, sino también adaptarse a las exigencias del mercado, mejorar la calidad del producto final y garantizar la sostenibilidad operativa.

En este contexto, la selección de herramientas tecnológicas adecuadas desempeña un papel crucial. Los controladores lógicos programables y las interfaces hombre-máquina son componentes esenciales en cualquier sistema automatizado, permitiendo el control preciso y la interacción intuitiva con los procesos industriales. Entre las marcas más destacadas en el mercado, Siemens, Rockwell y ABB han establecido un estándar de excelencia, ofreciendo soluciones innovadoras que se adaptan a las necesidades específicas de diferentes sectores industriales.

Este proyecto tiene como propósito principal simular un proceso industrial utilizando tecnología de la marca ABB, incluyendo un controlador lógico programable y su respectiva HMI. La simulación busca evaluar el desempeño de estos equipos en términos de eficiencia, flexibilidad y capacidad de adaptación, aspectos fundamentales en la automatización industrial moderna. Para llevar a cabo esta tarea, se emplearán herramientas especializadas como Automation Builder, para la programación del PLC Panel Builder, para el diseño y desarrollo de la interfaz HMI y KEP Server, para la integración de comunicación entre dispositivos mediante protocolos como IEC104 y Profinet.

El desarrollo de este sistema automatizado no solo permitirá una evaluación integral de las capacidades de ABB, sino que también ofrecerá una perspectiva comparativa frente a otras marcas líderes del mercado, como Siemens y Rockwell. De esta manera, el proyecto contribuirá a resaltar las ventajas competitivas de ABB en términos de innovación, versatilidad y eficiencia, proporcionando información valiosa para la selección de soluciones tecnológicas en la industria.

## **CAPÍTULO 1: ANTECEDENTES GENERALES**

## 1 ANTECEDENTES GENERALES

Este proyecto considera que, en la industria, la automatización de procesos es crucial para mejorar la productividad, la eficiencia operativa y la reducción de costos. Sin embargo, la selección de los controladores lógicos programables y las interfaces hombre-máquina (HMI) que gestionan estos procesos representa un desafío para las empresas, debido a la amplia variedad de marcas disponibles en el mercado, como Siemens, Rockwell y ABB, cada una con diferentes características, funcionalidades y niveles de integración.

Para este proyecto, se desarrollará una simulación de un sistema de control utilizando un controlador lógico programable de la marca ABB, programado con el software Automation Builder, que permite desarrollar la lógica de control en varios lenguajes. Además, se implementará una interfaz HMI programada con el software Panel Builder, que permitirá visualizar la actuación y la respuesta de los procesos. Estos equipos establecerán comunicación con el software KEP Server.

### 1.1 REQUERIMIENTOS

Para llevar a cabo la simulación de un proceso industrial utilizando un controlador ABB con su HMI, es necesario contar con una serie de recursos técnicos y humanos que garanticen el éxito de la implementación. Los requerimientos del proyecto se dividen en dos áreas principales hardware y software.

#### 1.1.1 Requerimientos de hardware

**Controlador ABB (PLC):** Se requiere un controlador lógico programable de la marca ABB, específicamente un modelo que permita la integración con sistemas HMI y soporte los protocolos de comunicación utilizados en la industria.

**Interfaz Hombre-Máquina (HMI):** Un dispositivo o software HMI compatible con el controlador ABB para facilitar la interacción y monitoreo en tiempo real del proceso simulado.

**PC o Estación de Trabajo:** Una computadora con capacidad suficiente para ejecutar el software de simulación y la plataforma de control, con un sistema operativo compatible.

**Sensores y Actuadores (opcional):** En caso de ser necesario, para simular un entorno más realista, se pueden incluir sensores y actuadores físicos conectados al PLC para replicar las condiciones de un proceso industrial real.

**Red de Comunicación Industrial:** Se debe contar con los elementos necesarios para implementar la red de comunicación entre el controlador ABB y otros dispositivos (HMI, sensores, actuadores), lo que puede incluir cables de comunicación, switches, routers o módulos de comunicación inalámbrica.

#### 1.1.2 Requerimientos de software

**Software de Programación de PLC (ABB Automation Builder):** Herramienta de software para la programación y configuración del controlador ABB. Este software permitirá crear la lógica de control necesaria para la simulación del proceso.

**Software de Simulación de Procesos:** Un entorno de simulación que permita replicar condiciones de un proceso industrial y que sea compatible con el controlador ABB.

**Software HMI:** Una plataforma para la creación de interfaces gráficas de control que permitan monitorear el proceso y gestionar alarmas, tendencias, entre otros.

**Licencias de Software:** Se deben contar con todas las licencias necesarias para el uso de los softwares mencionados, asegurando su legalidad y disponibilidad durante el proyecto.

### 1.1.3 Otros requerimientos

**Documentación Técnica:** Se deberá contar con la documentación técnica de los equipos y softwares utilizados para facilitar la configuración y programación del sistema.

**Espacio Físico:** Un área de trabajo adecuada donde se pueda instalar y probar el sistema, con acceso a la infraestructura de red y alimentación eléctrica necesaria para los equipos.

## 1.2 DESCRIPCIÓN DE HARDWARE Y SOFTWARE

A continuación, se describirán los equipos específicos a utilizar para el desarrollo del proyecto.

### 1.2.1 Controlador lógico programable (PLC) AC500

Un Controlador Lógico Programable (PLC) es un dispositivo electrónico utilizado para automatizar procesos industriales. El PLC recibe señales de entrada (de sensores o dispositivos de control), las procesa según un programa predefinido y luego genera señales de salida que activan equipos como motores, actuadores o válvulas.

Las características principales de un Controlador Lógico Programable (PLC) son:

**Entrada y salida (I/O):** Los PLCs tienen puertos de entrada y salida para interactuar con dispositivos externos, como sensores y actuadores.

**Capacidad de programación:** Los PLCs se programan generalmente utilizando lenguajes estandarizados, como el lenguaje de contactos (Ladder Logic), diagrama de bloques de funciones (FBD), o texto estructurado.

**Robustez:** Están diseñados para operar en entornos industriales exigentes, soportando altas temperaturas, vibraciones y condiciones de polvo o humedad.

**Modularidad:** Algunos PLCs son modulares, lo que permite agregar o quitar módulos de E/S según las necesidades del proceso.

Para este proyecto se utilizará un Controlador Lógico Programable (PLC) de la marca ABB específicamente el controlador PM5630 que se muestra en la figura 1-1 a continuación.



Fuente: <https://new.abb.com/products/es/3ABD00038893/pm5630-2eth>

Figura 1-1 Controlador PM5630.

### 1.2.2 Interfaz Hombre – Máquina (HMI)

Una Interfaz Hombre-Máquina (HMI) es un dispositivo o sistema de software que permite la interacción entre un operador humano y una máquina, sistema o proceso automatizado. Su función principal es facilitar la supervisión, el control y la visualización de datos de los procesos industriales en tiempo real.

Las características principales de una Interfaz Hombre-Máquina (HMI) son:

**Visualización:** Muestra información crucial del proceso, como variables, estados de operación, alarmas, gráficos y tendencias.

**Interacción:** Permite al operador controlar el sistema o máquina, ajustando parámetros, iniciando o deteniendo procesos, y respondiendo a alertas.

Para este proyecto se utilizará la Interfaz Hombre-Máquina (HMI) de la gama CP600 de la marca ABB específicamente la Interfaz Hombre-Máquina CP6607 como se muestra en la figura 1-2 a continuación.



Fuente: <https://new.abb.com/products/3ABD00038931/cp6607>

Figura 1-2 Interfaz Hombre – Máquina CP6607.

### 1.2.3 Automation Builder

Automation Builder es un software integral desarrollado por ABB que permite la programación, configuración y simulación de sistemas de automatización industrial. Está

diseñado para trabajar con productos de ABB, como PLCs (Controladores Lógicos Programables), HMI (Interfaz Hombre-Máquina), robots y otros dispositivos de automatización.

Las características principales de Automation Builder son:

**Programación de PLC:** Soporta la programación de los PLCs de ABB, como la serie AC500, utilizando lenguajes de programación estandarizados conforme a IEC 61131-3 (como Ladder Diagram, Function Block Diagram, Structured Text, etc.)

**Simulación:** Facilita la simulación del funcionamiento de un sistema antes de la implementación física, lo que ayuda a detectar y corregir errores en las primeras etapas.

**Comunicación y redes:** Soporta la configuración de protocolos de comunicación y redes industriales, como EtherCAT, PROFINET, Modbus, entre otros.

**Gestión centralizada:** Proporciona un entorno unificado para gestionar diferentes aspectos del sistema de automatización, desde la programación hasta la simulación y el monitoreo.

Automation Builder es especialmente útil para crear y gestionar sistemas complejos de automatización industrial, integrando controladores, interfaces, y dispositivos de manera eficiente.

#### 1.2.4 Panel Builder

Panel Builder es una herramienta de software de ABB diseñada para crear y configurar interfaces HMI (Interfaz Hombre-Máquina) en sistemas de automatización industrial. Se utiliza específicamente para diseñar pantallas que permitan a los operadores supervisar y controlar máquinas y procesos industriales de manera intuitiva. El software se integra con otros productos de ABB, como los PLCs y sistemas de control.

Las características principales de Panel Builder son:

**Diseño de pantallas HMI:** Permite crear interfaces gráficas personalizadas para visualizar y controlar procesos industriales en tiempo real. Los usuarios pueden agregar botones, gráficos, medidores, alarmas y otros componentes interactivos.

**Simulación de HMI:** Facilita la simulación del diseño antes de implementarlo en el hardware físico, lo que ayuda a verificar la funcionalidad y la experiencia del usuario.

**Comunicación PLC:** Está diseñado para integrarse fácilmente con controladores como los PLCs de ABB (por ejemplo, la serie AC500) para la supervisión y el control de procesos.

**Gestión de alarmas y eventos:** Proporciona herramientas para configurar alarmas, eventos y avisos que pueden alertar a los operadores sobre fallos o condiciones críticas en el proceso.

**Configuración de tendencias y reportes:** Permite configurar la recolección y visualización de datos históricos y en tiempo real, facilitando el análisis de tendencias y la generación de reportes.

**Compatibilidad con hardware HMI de ABB:** Está optimizado para funcionar con los paneles HMI de ABB, como la serie CP600. Para este proyecto se utilizará la versión de Panel Builder 600.

#### 1.2.5 KEP Server

KEPServerEX es un software de conectividad industrial desarrollado por Kepware Technologies que proporciona un entorno para la comunicación y el intercambio de datos entre

dispositivos de automatización y sistemas de supervisión. Es ampliamente utilizado en la industria para integrar dispositivos de diferentes fabricantes y protocolos en un solo sistema.

Las características principales de KEPServerEX son:

**Conectividad Multivendor:** Permite la comunicación con una variedad de dispositivos, como PLCs, sensores, controladores de procesos, HMI, y más, independientemente del fabricante, a través de múltiples protocolos industriales (como OPC, Modbus, Ethernet/IP, BACnet, entre otros).

**OPC DA, AE, y UA:** Ofrece soporte para los estándares de comunicación OLE for Process Control, permitiendo a las aplicaciones de software acceder a datos en tiempo real y a alarmas/eventos de manera estandarizada.

**Configuración de Canales y Dispositivos:** Proporciona una interfaz de usuario gráfica para configurar fácilmente canales de comunicación y dispositivos conectados, lo que simplifica la integración en sistemas de automatización.

**Gestión de Datos:** Permite la recopilación, gestión y visualización de datos en tiempo real, lo que es fundamental para la supervisión y el análisis del rendimiento de los procesos industriales.

KEPServerEX es una herramienta crucial en la automatización industrial, ya que permite la interoperabilidad de sistemas y dispositivos, optimizando la recopilación y el análisis de datos para una mejor toma de decisiones en tiempo real.

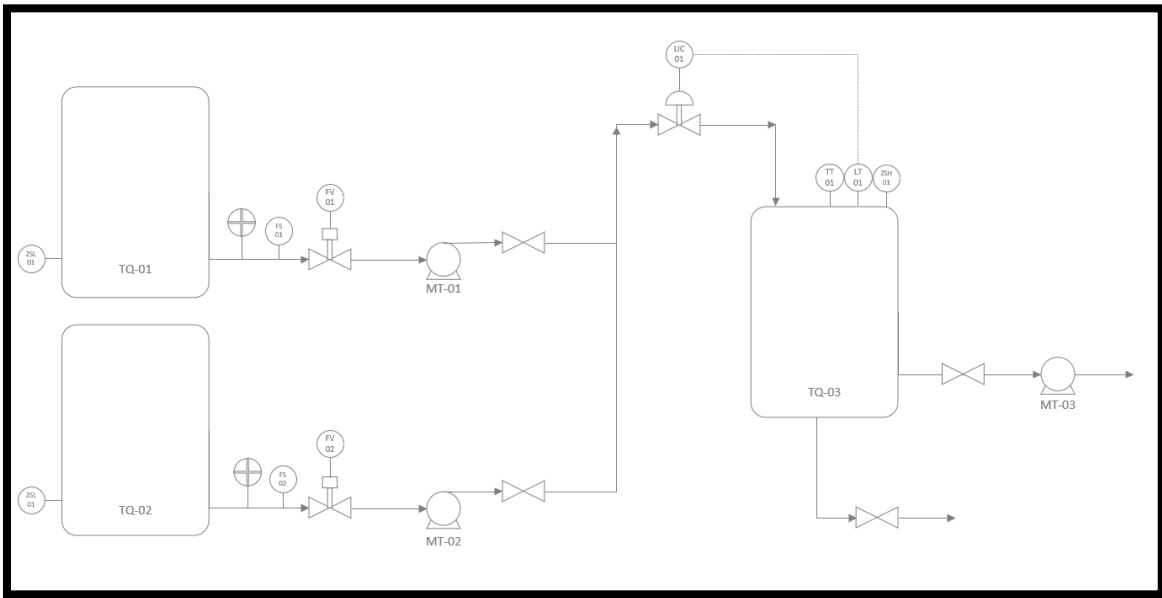
### 1.3 DESCRIPCIÓN DEL PROCESO

Se desarrollará un control de nivel y temperatura en un tanque de Mezcla.

Este proceso simula el control de un tanque de mezcla en una planta industrial. El tanque recibe y mezcla dos fluidos, controlando su nivel y temperatura, utilizando sensores y actuadores

conectados al controlador ABB. Las señales analógicas provienen de sensores que monitorean el nivel del líquido y la temperatura dentro del tanque, mientras que las señales digitales provienen de diferentes dispositivos de control y seguridad que garantizan un funcionamiento adecuado del sistema.

A continuación, se muestra en la Figura 1-3 el plano P&ID del proyecto a realizar.



Fuente: Elaboración propia.

Figura 1-3 P&ID del Proceso.

### 1.3.1 Entradas analógicas

**Sensor de Nivel LT-01 (AI\_01):** Un sensor de nivel ultrasónico o de presión mide la altura del líquido en el tanque. Esta señal se transmite al PLC como una variable continua que varía de 0 a 100%, representando el nivel mínimo y máximo del tanque, respectivamente. Rango de la señal: 0 – 10V DC.

**Sensor de Temperatura TT-01 (AI\_02):** Un termopar o sensor que mide la temperatura del líquido en el tanque. Esta señal analógica permite un monitoreo preciso de la temperatura,

asegurando que el líquido se mantenga dentro de los límites de seguridad establecidos. Rango de la señal: 0-10 V, representando una temperatura de 0°C a 100°C.

### 1.3.2 Entradas digitales

**Botón de Inicio de Proceso (DI\_01):** Un botón físico en el panel de control que, al ser activado, envía una señal digital para iniciar el proceso de mezcla.

Estado: 0 = Detenido, 1 = Iniciar proceso.

**Botón de Parada de Emergencia (DI\_02):** Un interruptor de parada de emergencia que, al presionarlo, detiene el sistema inmediatamente por razones de seguridad.

Estado: 0 = Normal, 1 = Parada de emergencia activada.

**Sensor de Presencia de Fluido A FS-01 (DI\_03):** Un sensor digital detecta si el fluido A ha llegado al tanque. Si no se detecta la presencia del fluido, el proceso no puede iniciar.

Estado: 0 = Fluido A no presente, 1 = Fluido A presente.

**Sensor de Presencia de Fluido B FS-02(DI\_04):** Similar al sensor de fluido A, este sensor digital detecta si el fluido B está en el tanque.

Estado: 0 = Fluido B no presente, 1 = Fluido B presente.

**Sensor de Nivel Mínimo ZSL-01 (DI\_05):** Este sensor digital asegura que el nivel del líquido en el tanque no esté por debajo de un valor mínimo, lo cual podría dañar las bombas o el sistema de mezcla del estanque 1.

Estado: 0 = Nivel normal, 1 = Nivel bajo (alarma).

**Sensor de Nivel Mínimo ZSL-02 (DI\_06):** Este sensor digital asegura que el nivel del líquido en el tanque no esté por debajo de un valor mínimo, lo cual podría dañar las bombas o el sistema de mezcla del estanque 2.

Estado: 0 = Nivel mínimo alcanzado, 1 = Nivel bajo (alarma).

**Sensor de Nivel Máximo ZSH-01 (DI\_07):** Sensor que envía una señal cuando el nivel del líquido en el tanque alcanza el máximo permitido, activando un sistema de alarma o parando la entrada de fluido.

Estado: 0 = Nivel normal, 1 = Nivel máximo (alarma).

### 1.3.3 Funcionamiento del proceso

El proceso comienza cuando el operador presiona el Botón de Inicio (DI\_01). El controlador ABB verifica las condiciones iniciales, como la presencia de los fluidos A y B en el tanque (DI\_03 y DI\_04) y que los niveles no estén fuera de los rangos definidos (DI\_05, DI\_06 y DI\_07).

Durante la operación, el Sensor de Nivel (AI\_01) monitorea continuamente el nivel de líquido en el tanque TQ-03, si el nivel alcanza el máximo permitido (DI\_07), el controlador detiene automáticamente la entrada de fluido y activa una alarma.

Simultáneamente, el Sensor de Temperatura (AI\_02) mide la temperatura del fluido. Si la temperatura sube demasiado, el sistema activa una alarma y detiene el proceso de calentamiento o mezcla.

El proceso puede ser detenido en cualquier momento por el Botón de Parada de Emergencia (DI\_02), asegurando que el sistema pueda ser detenido de manera segura ante cualquier contingencia.

### 1.3.4 Comunicación de datos del sistema

En el proyecto se utilizarán dos protocolos de comunicación industrial para garantizar la integración eficiente de los dispositivos y la recolección de datos en tiempo real: IEC104 y Profinet. Cada uno de estos protocolos desempeña un papel específico en la arquitectura del sistema.

**IEC104** es un protocolo de comunicación ampliamente utilizado en sistemas industriales para la transmisión de datos sobre redes Ethernet. En este proyecto, se empleará IEC104 para conectar el controlador ABB con KepServer, un servidor de comunicaciones que facilita la conexión entre distintos dispositivos y sistemas de control.

**Profinet** es un protocolo de comunicación industrial basado en Ethernet diseñado para integrar sistemas de automatización, especialmente cuando se requiere alta velocidad y confiabilidad en la transmisión de datos. En este proyecto, Profinet será utilizado para conectar el controlador ABB a una IO remota, lo que permite una interacción directa y eficiente con dispositivos distribuidos en la planta.

### 1.3.5 Módulo de comunicación

El módulo de comunicación a utilizar en el proyecto es el CM579 el cual se presenta en la Figura 1-4 a continuación.



Fuente: <https://new.abb.com/products/3ABD00038816/cm579-pnio>

Figura 1-4 – Módulo de comunicación CM579-PNIO.

El CM579 es un módulo de comunicación desarrollado por ABB para los controladores de la serie AC500, que proporciona conectividad en redes PROFINET IO Controller o PROFINET IO Device. Es fundamental para integrar los PLCs de ABB con otros dispositivos y sistemas en una red PROFINET, uno de los protocolos de comunicación más utilizados en la automatización industrial.

### 1.3.6 Módulos de entradas y salidas

Los módulos de entradas y salidas a utilizar en el proyecto son el módulo DA501 y el módulo CI502, los cuales se presentarán a continuación.

El **CI502**, presentado en la figura 1-5 es un módulo de comunicación de la serie AC500 de ABB. Está diseñado para expandir la capacidad de comunicación de los controladores lógicos programables (PLC) AC500, permitiendo la conexión a redes Profinet. Este módulo posee PROFINET IO RT device, 8 Digital inputs (DI) de 24VDC, 8 Digital output (DO) de 24V DC - 0.5A y 8 configurables Digital input/output (DI/DO) de 24VDC - 0.5A. Ampliable hasta 10 módulos de I/O.



<https://new.abb.com/products/3ABD00029508/ci502-pnio>

Figura 1-5 Módulo CI502.

El **DA501**, presentado en la figura 1-6 es un módulo de entradas y salidas (I/O) compacto diseñado para el sistema de controlador lógico programable (PLC) ABB AC500. Este posee 16 Digital inputs (DI) de 24VDC, 4 Análogos Input (AI), 2 Análogos Output (AO) y 8 configurables Digital input/output (DI/DO) de 24VDC - 0.5A.



<https://new.abb.com/products/3ABD00029513/da501>

Figura 1-6 Módulo DA501.

#### 1.4 OBJETIVOS DEL PROYECTO

A continuación, se presenta el objetivo general y los objetivos específicos, los cuales conforman este proyecto.

#### 1.4.1 Objetivo general

Realizar una simulación de un proceso industrial utilizando un controlador ABB y su HMI, con el fin de evaluar sus funcionalidades.

#### 1.4.2 Objetivos específicos

- a. Diseñar un proceso industrial simulado que permita demostrar las capacidades de control y automatización del PLC ABB.
- b. Programar la lógica de control del proceso utilizando el entorno de programación de ABB Automation Builder.
- c. Implementar una interfaz gráfica de usuario a través de la HMI de ABB para monitorear y gestionar el proceso.

## **CAPÍTULO 2: DESARROLLO DEL PROYECTO**

## **2 DESARROLLO DEL PROYECTO**

En este capítulo se describirá el desarrollo de la programación necesaria para llevar a cabo el proyecto de manera óptima. En primer lugar, se detallará cómo se realiza la programación en el PLC AC500 utilizando el software Automation Builder. En segundo lugar, se abordará la creación de las pantallas de operación mediante el software Panel Builder 600.

### **2.1 AUTOMATION BUILDER**

A continuación, se presentará la creación de la programación necesaria para el PLC AC500, considerando los siguientes aspectos clave:

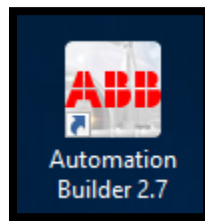
- Creación del programa.
- Configuración del hardware.
- Asignación de canales de I/O (entradas/salidas).
- Asignación de librerías.
- Desarrollo de la lógica de control.
- Definición de variables globales.
- Configuración de alarmas.
- Establecimiento de símbolos.
- Visualización.
- Comunicación de datos.

### 2.1.1 Creación del programa

Para realizar la creación del programa es indispensable contar con el software Automation Builder, específicamente la versión 2.7, que se ilustra en la Figura 2-1.

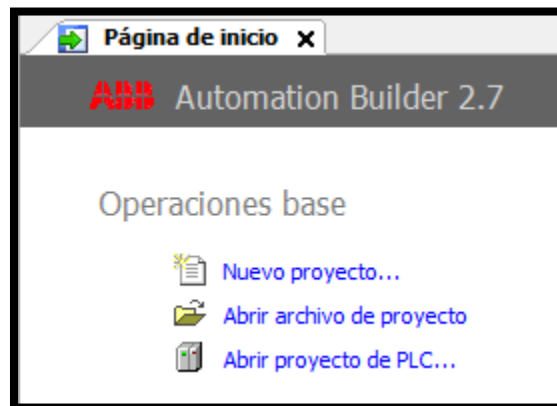
A continuación, se presenta una serie de pasos a seguir para la correcta creación del proyecto:

1. Iniciar el software Automation Builder y seleccionar la opción para crear un nuevo proyecto Figura 2-2.



Fuente: elaboración propia.

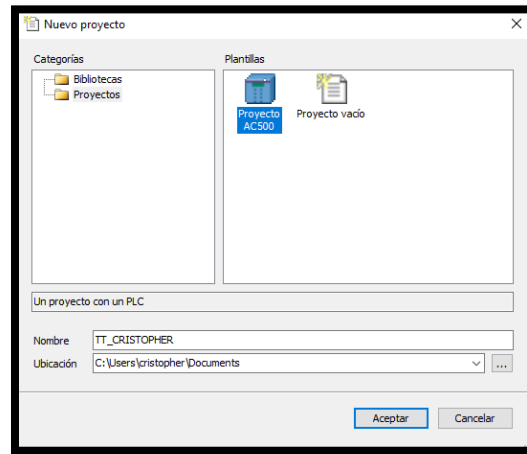
Figura 2-1 Automation builder 2.7



Fuente: elaboración propia.

Figura 2-2 Automation Builder 2.7, nuevo proyecto.

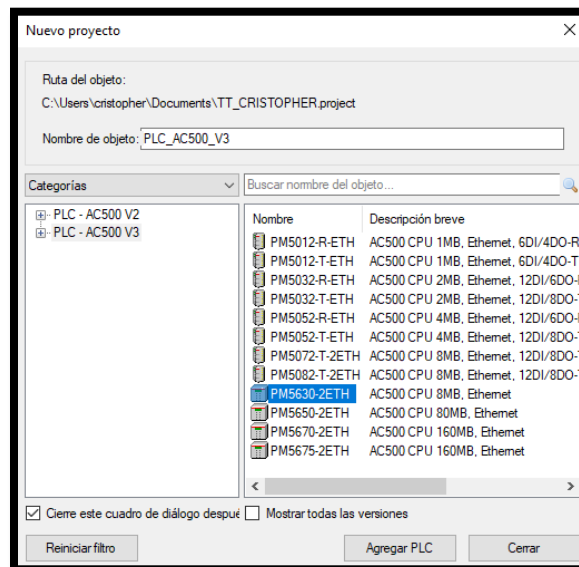
2. Seleccionar la opción de "Proyecto AC500", definir un nombre representativo para el proyecto y especificar la ubicación donde se almacenará.



Fuente: elaboración propia.

Figura 2-3 Nuevo proyecto “Proyecto AC500”

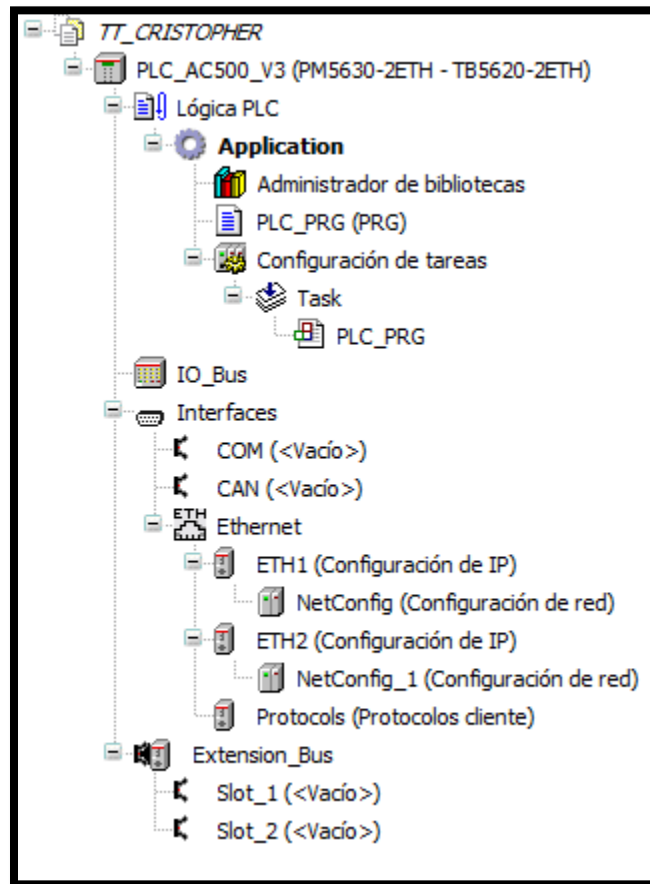
- Ya definido el nombre y la ubicación del proyecto, se debe seleccionar el controlador a utilizar. En este caso, se elegirá el modelo “**AC500 PM5630-2ETH**” mostrado en la Figura 2-4.



Fuente: elaboración propia.

Figura 2-4 Selección modelo AC500 PM5630-2ETH.

- Una vez creado el proyecto, se desplegará la información correspondiente a la Figura 2-5. Donde se deben definir los elementos de control, tales como: Módulos de entradas y salidas de señales, protocolos de comunicación, librerías a utilizar, programación lógica, entre otros aspectos necesarios para el desarrollo del proyecto.



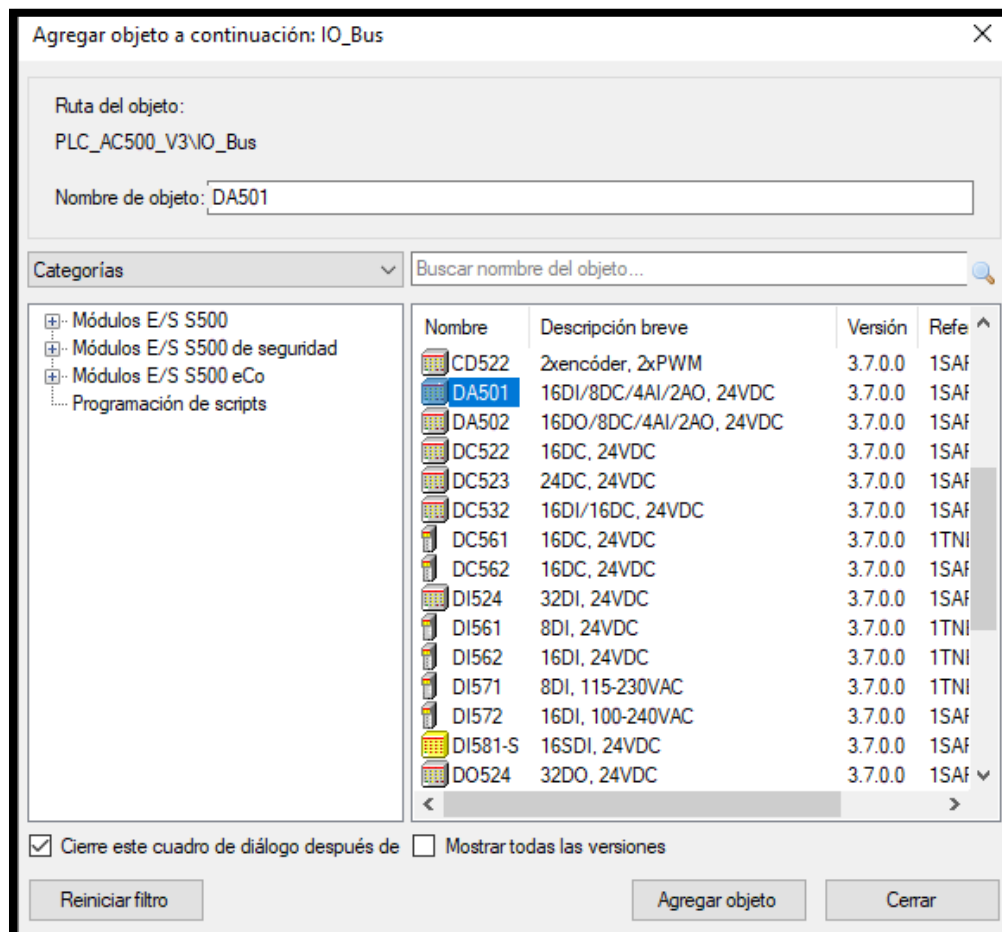
Fuente: Elaboración propia.

Figura 2-5 Hardware del proyecto.

### 2.1.2 Creación del hardware

A continuación, se detallan los pasos necesarios para integrar el hardware requerido:

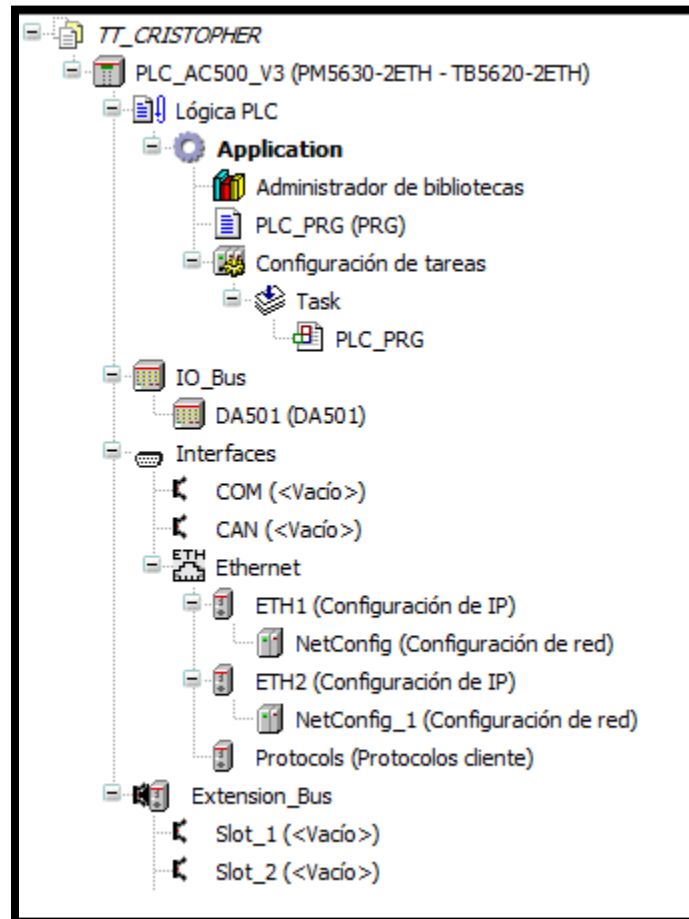
1. Para utilizar el hardware en la programación, es fundamental incluir los módulos correspondientes.
2. En la sección "IO\_BUS", mostrada en la Figura 2-5, debe añadirse el módulo DA501, el cual es esencial para gestionar las entradas y salidas analógicas, como se muestra en la Figura 2-6.



Fuente: Elaboración propia.

Figura 2-6 Agregar objeto DA501.

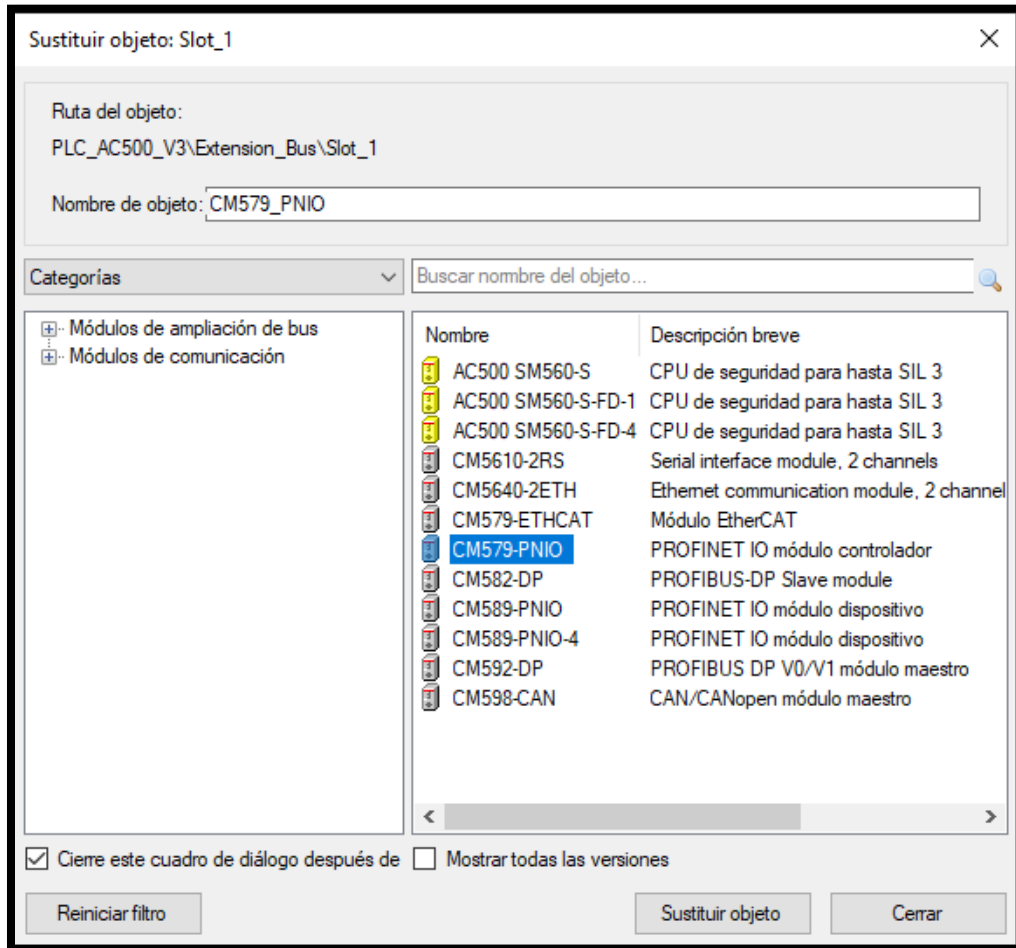
El hardware mostrado en la Figura 2-7 incluye el módulo IO DA501 ya agregado.



Fuente: Elaboración propia.

Figura 2-7 Hardware del proyecto, incluyendo DA501.

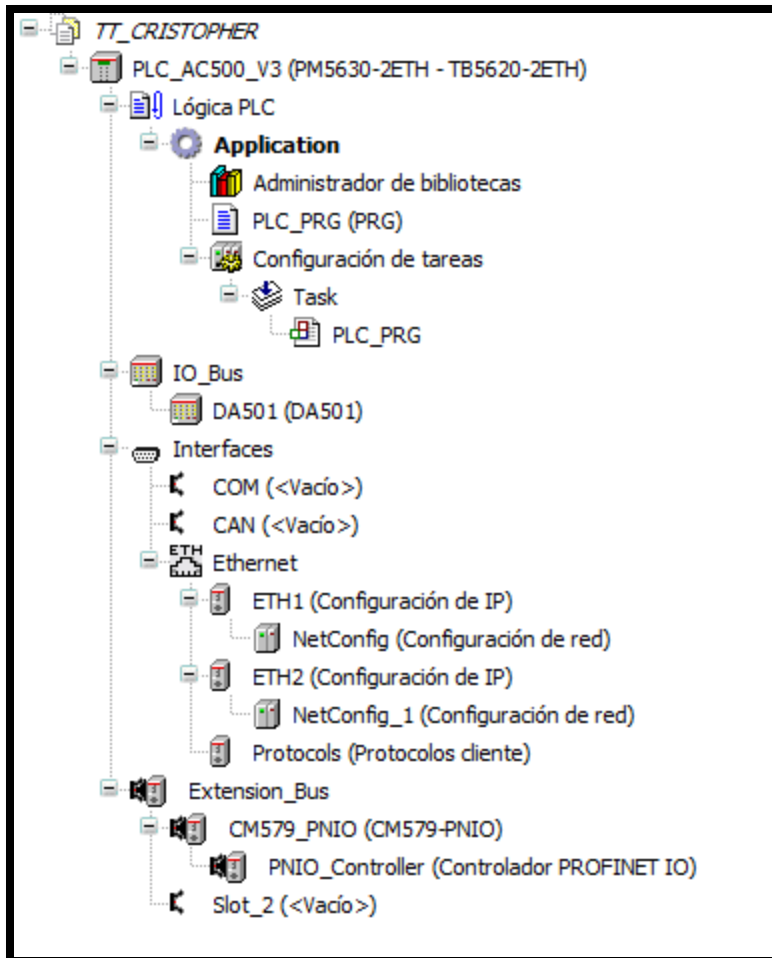
3. Para insertar el módulo CI502, es necesario dirigirse a la sección “Extensión\_Bus” mostrado en la Figura 2-7 anteriormente vista y agregar el módulo de comunicación PROFINET IO. En este caso, se añadirá el módulo CM579-PNIO mostrada en la figura2-8 para establecer la comunicación con el módulo de I/O remota CI502.



Fuente: Elaboración propia.

Figura 2-8 Agregar objeto PROFINET IO CM579-PNIO.

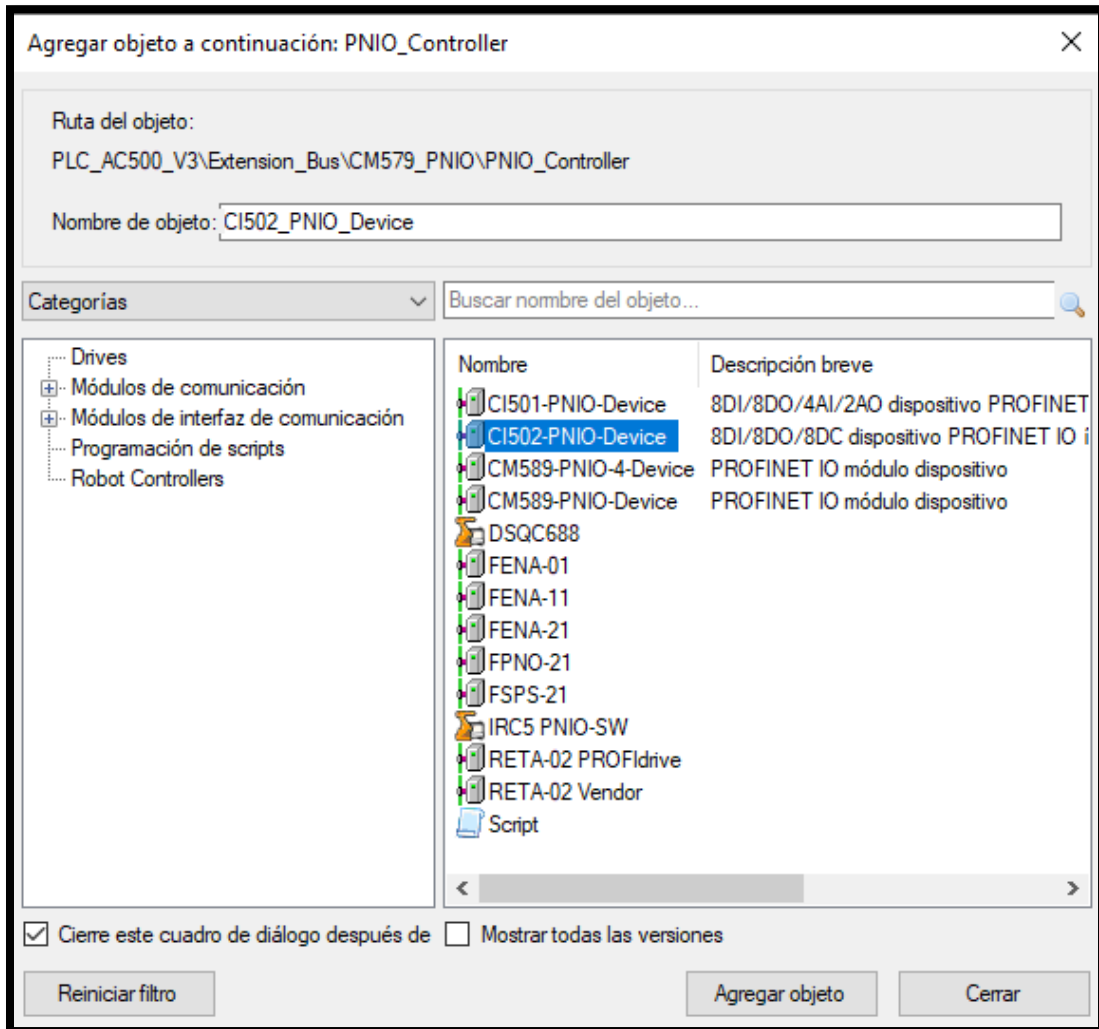
El hardware mostrado en la Figura 2-9 incluye el módulo CM579-PNIO ya agregado.



Fuente: Elaboración propia.

Figura 2-9 Hardware del proyecto, incluyendo CM579-PNIO.

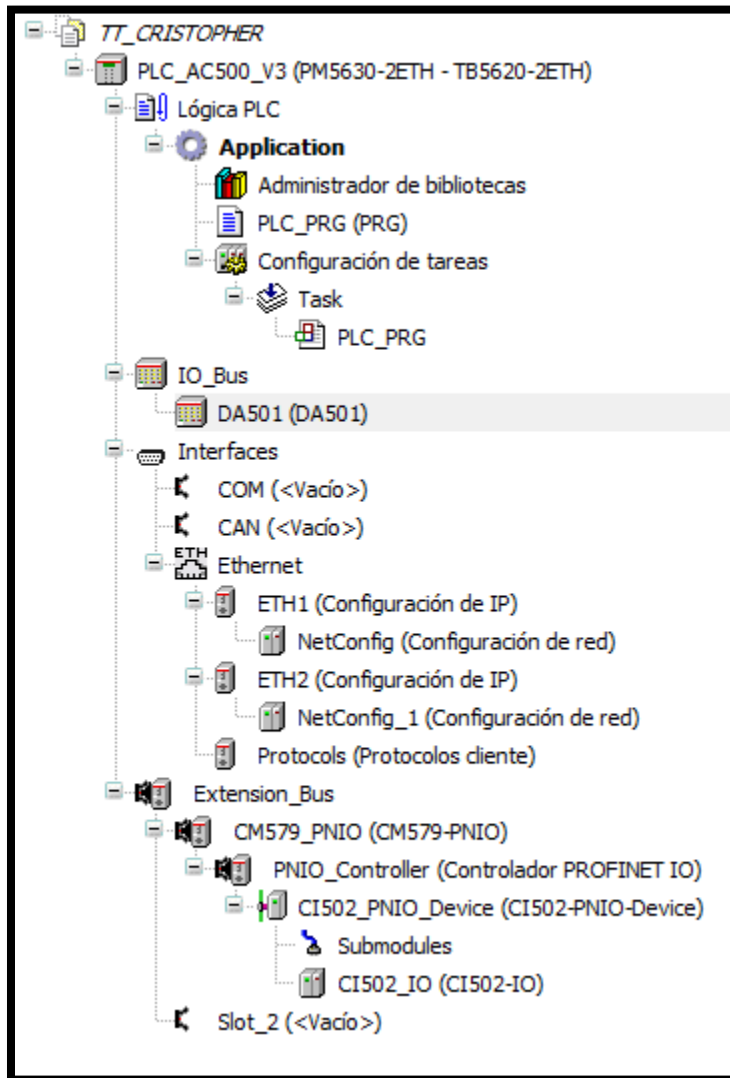
4. Una vez insertado el módulo PROFINET IO CM579-PNIO, se debe agregar el módulo CI502 debajo de éste. El módulo CM579-PNIO será el encargado de recibir las señales de IO provenientes del módulo CI502, mostrado en la figura 2-10.



Fuente: Elaboración propia.

Figura 2-10 Agregar objeto CI502-PNIO-Device.

Una vez completada la integración de los módulos de entrada y salida, el hardware debe quedar configurado como se muestra en la Figura 2-11.



Fuente: Elaboración propia.

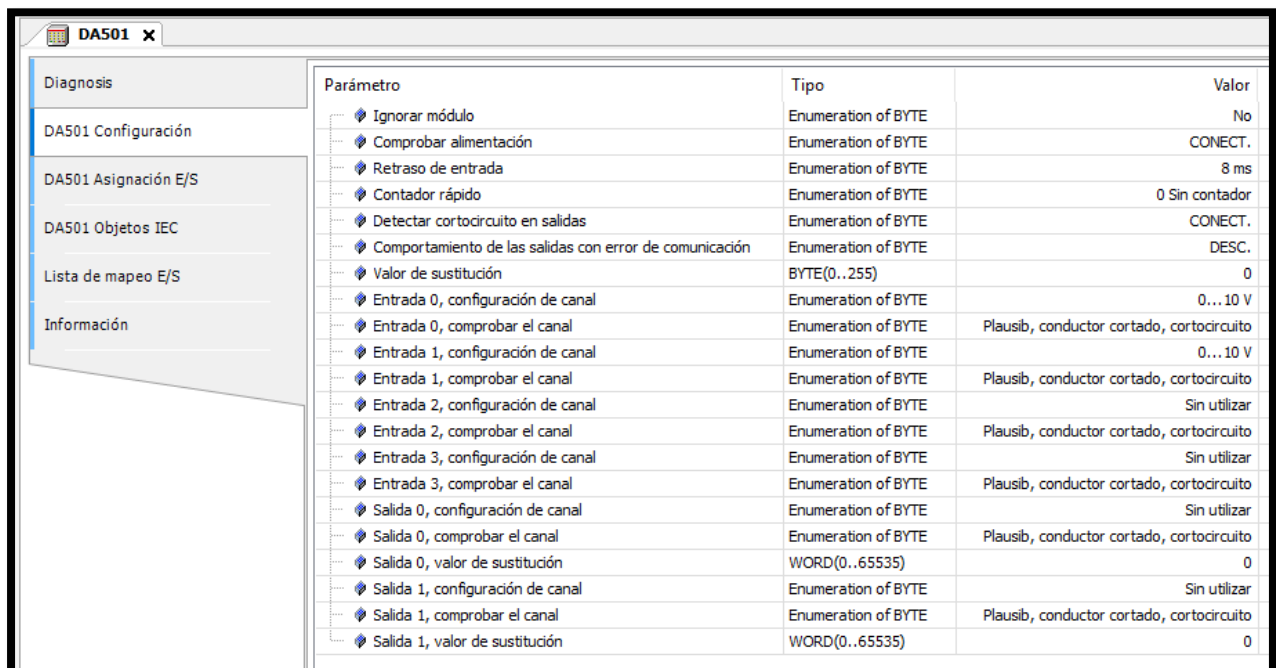
Figura 2-11 Hardware del proyecto, incluyendo CI502.

### 2.1.3 Asignación de canales IO

A continuación, se deben asignar los canales de IO a utilizar en los módulos DA501 y CI502.

Para ello, se seguirán los siguientes pasos:

1. Seleccionar el módulo DA501; al hacerlo, se abrirá la pestaña mostrada en la Figura 2-12.
2. En la pestaña mostrada en la Figura 2.12, dirigirse a la sección "DA501 Configuración" para habilitar los canales analógicos que se utilizarán.
3. En este caso, se habilitarán la entrada analógica 1 y la entrada analógica 2.
4. Configurar los canales con un rango de entrada de 0 a 10 V DC, ya que estos valores se utilizarán para realizar las mediciones correspondientes de nivel y temperatura.



Parámetro	Tipo	Valor
Ignorar módulo	Enumeration of BYTE	No
Comprobar alimentación	Enumeration of BYTE	CONNECT.
Retraso de entrada	Enumeration of BYTE	8 ms
Contador rápido	Enumeration of BYTE	0 Sin contador
Detectar cortocircuito en salidas	Enumeration of BYTE	CONNECT.
Comportamiento de las salidas con error de comunicación	Enumeration of BYTE	DESC.
Valor de sustitución	BYTE(0..255)	0
Entrada 0, configuración de canal	Enumeration of BYTE	0... 10 V
Entrada 0, comprobar el canal	Enumeration of BYTE	Plausib, conductor cortado, cortocircuito
Entrada 1, configuración de canal	Enumeration of BYTE	0... 10 V
Entrada 1, comprobar el canal	Enumeration of BYTE	Plausib, conductor cortado, cortocircuito
Entrada 2, configuración de canal	Enumeration of BYTE	Sin utilizar
Entrada 2, comprobar el canal	Enumeration of BYTE	Plausib, conductor cortado, cortocircuito
Entrada 3, configuración de canal	Enumeration of BYTE	Sin utilizar
Entrada 3, comprobar el canal	Enumeration of BYTE	Plausib, conductor cortado, cortocircuito
Salida 0, configuración de canal	Enumeration of BYTE	Sin utilizar
Salida 0, comprobar el canal	Enumeration of BYTE	Plausib, conductor cortado, cortocircuito
Salida 0, valor de sustitución	WORD(0..65535)	0
Salida 1, configuración de canal	Enumeration of BYTE	Sin utilizar
Salida 1, comprobar el canal	Enumeration of BYTE	Plausib, conductor cortado, cortocircuito
Salida 1, valor de sustitución	WORD(0..65535)	0

Fuente: Elaboración propia.

Figura 2-12 DA501 Configuración.

5. Luego, dirigirse a la sección "DA Asignación E/S", donde se asignarán las variables correspondientes a las entradas digitales DI8, DI9, DI10 y DI11, así como a las entradas analógicas AI0+ y AI1+, mostradas en la figura 2-13.
6. Asignar las siguientes variables a los canales correspondientes:
  - Canal digital DI8: **DI\_01**
  - Canal digital DI9: **DI\_02**
  - Canal digital DI10: **DI\_03**
  - Canal digital DI11: **DI\_04**
  - Canal analógico AI0+: **AI\_01**
  - Canal analógico AI1+: **AI\_02**

Variable	Asignación	Canal	Dirección	Tipo
		Entradas digitales DI0 - DI7	%IB0	BYTE
		Entradas digitales DI8 - DI15	%IB1	BYTE
DI_01		Entrada digital DI8	%IX1.0	BOOL
DI_02		Entrada digital DI9	%IX1.1	BOOL
DI_03		Entrada digital DI10	%IX1.2	BOOL
DI_04		Entrada digital DI11	%IX1.3	BOOL
		Entrada digital DI12	%IX1.4	BOOL
		Entrada digital DI13	%IX1.5	BOOL
		Entrada digital DI14	%IX1.6	BOOL
		Entrada digital DI15	%IX1.7	BOOL
AI_01		Entrada analógica AI0+	%IW1	INT
AI_02		Entrada analógica AI1+	%IW2	INT
		Entrada analógica AI2+	%IW3	INT
		Entrada analógica AI3+	%IW4	INT
		Salida analógica AO0+	%QW0	INT
		Salida analógica AO1+	%QW1	INT
		Entradas digitales DC16 - DC23	%IB10	BYTE
		Salidas digitales DC16 - DC23	%QB6	BYTE
		Contador rápido		

Fuente: Elaboración propia.

Figura 2-13 Asignación de variables módulo DA501.

7. Seleccionar el módulo CI502; al hacerlo, se abrirá la pestaña mostrada en la Figura 2-14.

8. En la pestaña mostrada en la Figura 2-14, dirigirse a la sección "PNIO Module Asignación E/S", donde se asignarán las variables correspondientes a las entradas digitales DI8, DI9, DI10 y DI11.
9. Asignar las siguientes variables a los canales correspondientes:
  - Canal digital DI8: **DI\_05**
  - Canal digital DI9: **DI\_06**
  - Canal digital DI10: **DI\_07**
  - Canal digital DI11: **DI\_08**

Variable	Asignación	Canal	Dirección	Tipo
		8-bit digital input	%IB28	USINT
		8-bit digital input	%IB29	USINT
DI_05		Channel 8	%IX29.0	BOOL
DI_06		Channel 9	%IX29.1	BOOL
DI_07		Channel 10	%IX29.2	BOOL
DI_08		Channel 11	%IX29.3	BOOL
		Channel 12	%IX29.4	BOOL
		Channel 13	%IX29.5	BOOL
		Channel 14	%IX29.6	BOOL
		Channel 15	%IX29.7	BOOL
		Fast Counter: Actual value 1	%ID8	UDINT
		Fast Counter: Actual value 2	%ID9	UDINT
		Fast Counter: State byte 1	%IB40	USINT
		Fast Counter: State byte 2	%IB41	USINT
		Input PS	%IB42	Enumeration of BYTE
		Output CS	%IB43	Enumeration of BYTE
		8-bit digital output	%QB32	USINT
		8-bit digital output	%QB33	USINT
		Fast Counter: Start value 1	%QD9	UDINT
		Fast Counter: End value 1	%QD10	UDINT
		Fast Counter: Start value 2	%QD11	UDINT
		Fast Counter: End value 2	%QD12	UDINT
		Fast Counter: Control byte 1	%QB52	USINT
		Fast Counter: Control byte 2	%QB53	USINT
		Output PS	%QB54	Enumeration of BYTE
		Input CS	%QB55	Enumeration of BYTE

Fuente: Elaboración propia.

Figura 2-14 Asignación de variables módulo CI502.

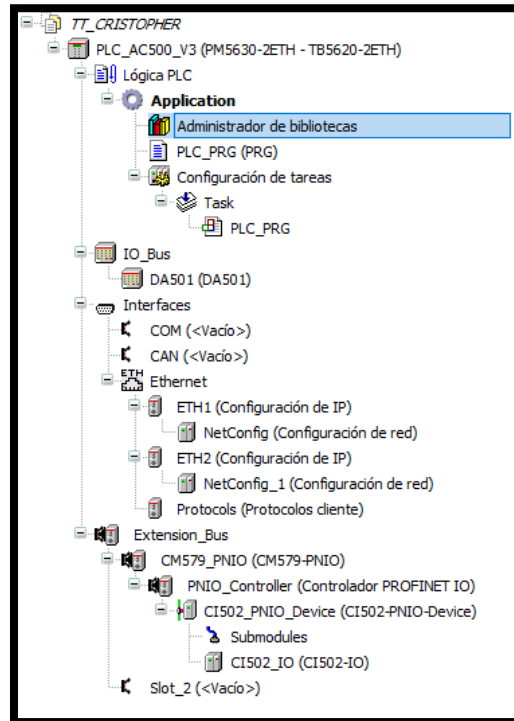
#### 2.1.4 Asignación de librerías

A continuación, se describirá la asignación de librerías necesarias para la implementación de la lógica de control del proceso. Estas librerías son fundamentales, ya que se utilizarán macros predefinidas proporcionadas por ABB para la configuración de los siguientes elementos:

- Bombas On/Off.
- Válvulas On/Off.
- Controlador PID.

Estas herramientas simplificarán el desarrollo del proyecto y garantizarán una configuración eficiente y estandarizada.

Para asignar las librerías, es necesario dirigirse a la sección de “Administrador de bibliotecas”, como se muestra en la Figura 2-15. Al acceder, se abrirá la pestaña reflejada en la Figura 2-16, donde se podrán gestionar las librerías necesarias para el proyecto.



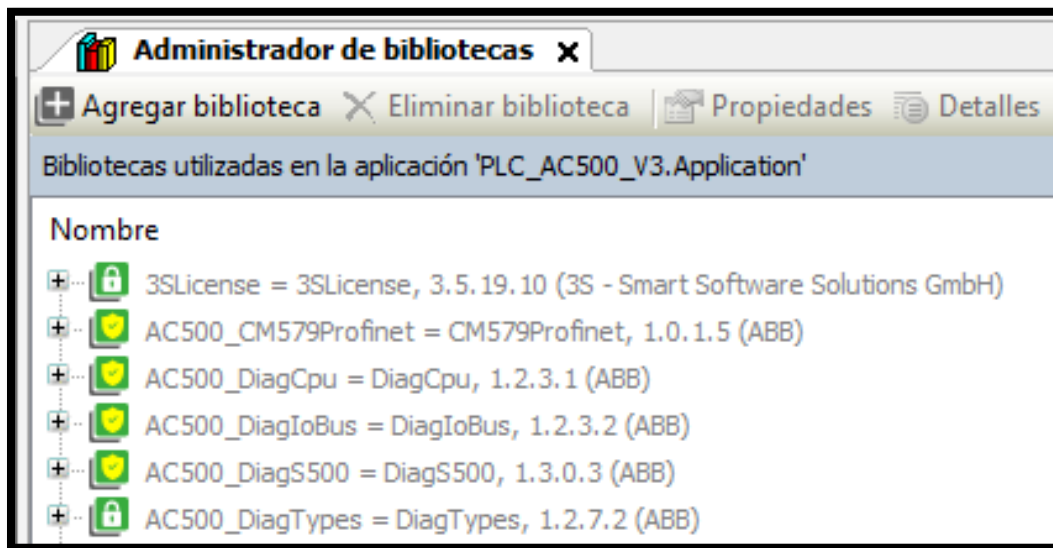
Fuente: Elaboración propia.

Figura 2-15 Proyecto - Administrador de bibliotecas.

En este caso, se agregarán dos librerías específicas:

- Librería UTIL.
- Librería PCO.

Estas librerías proporcionan las funciones y macros necesarias para la configuración y control del proceso.



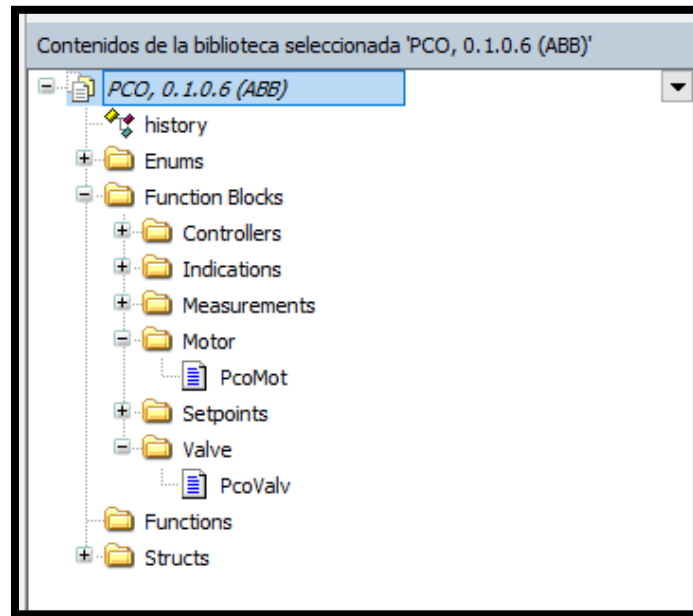
Fuente: Elaboración propia.

Figura 2-16 Administrador de bibliotecas.

En la Figura 2.17, se muestran las carpetas contenidas en esta librería PCO, destacando las macros que se utilizarán en la programación:

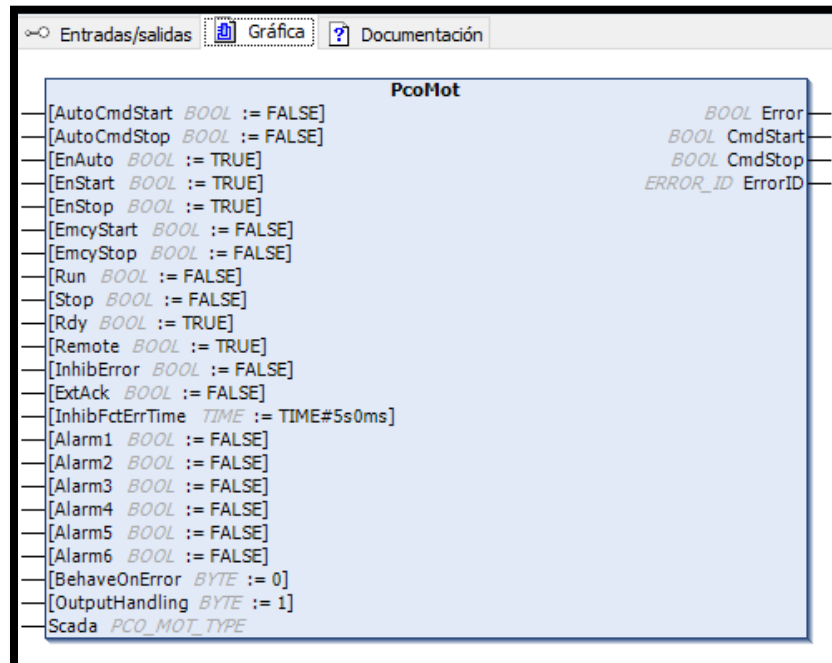
- **PcoMot**, una macro relacionada con el control de motores On/Off, ilustrada en la Figura 2-18.
- **PcoValv**, una macro asociada con el control de válvulas On/Off, ilustrada en la Figura 2-19.

Estas macros serán clave para la implementación eficiente de las funciones del proceso.



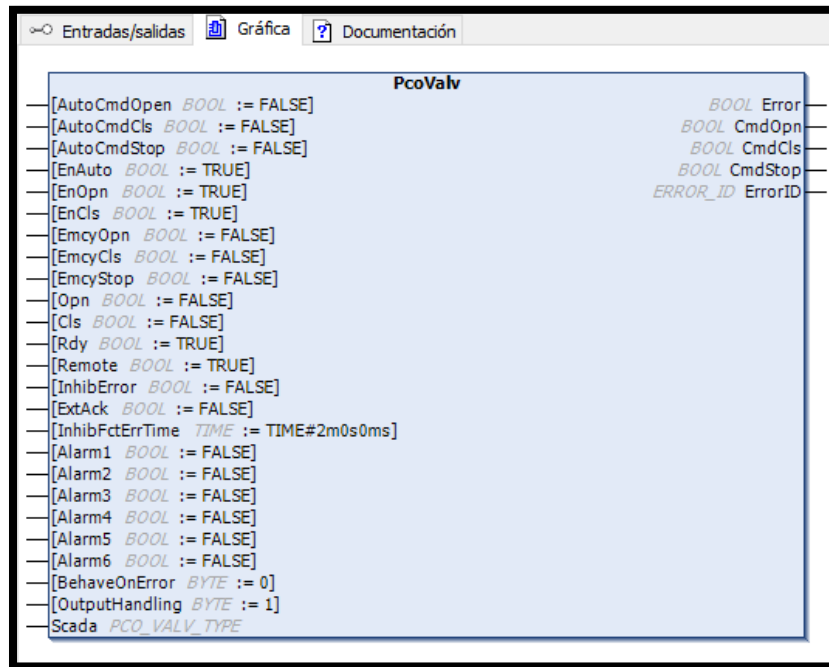
Fuente: Elaboración propia.

Figura 2-17 PCO, 0.1.0.6.



Fuente: Elaboración propia.

Figura 2-18 Macro PcoMot.



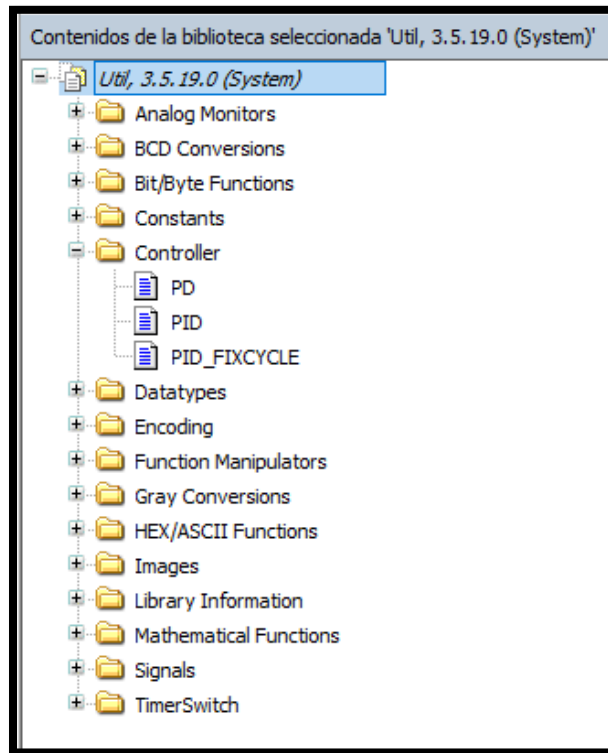
Fuente: Elaboración propia.

Figura 2-19 Macro PcoValv.

En la Figura 2-20, se presentan las carpetas contenidas en la librería UTIL, destacando la macro que se utilizará en la programación:

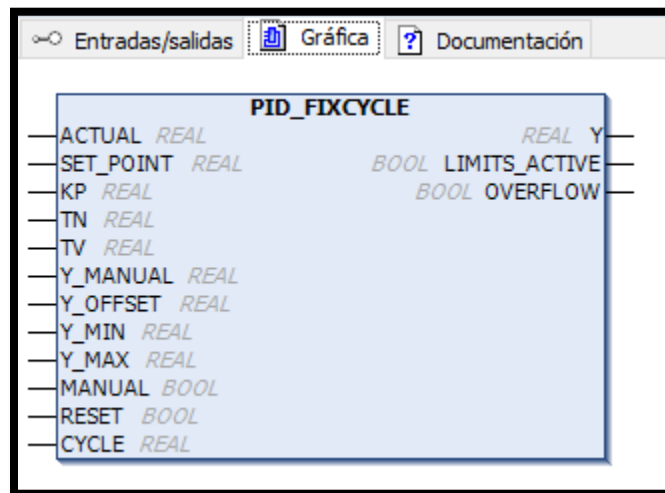
- **PID**, una macro relacionada con el control PID\_FIXCYCLE. En este caso, se empleará para controlar una válvula modulada que varía entre el 0% y el 100%. Esta macro está ilustrada en la Figura 2-21.

Esta macro será fundamental para la implementación eficiente de las funciones del proceso, garantizando un control preciso y ajustado a los requerimientos del sistema.



Fuente: Elaboración propia

Figura 2-20 Util, 3.5.19.0.



Fuente: Elaboración propia

Figura 2-21 Macro PID\_FIXCYCLE

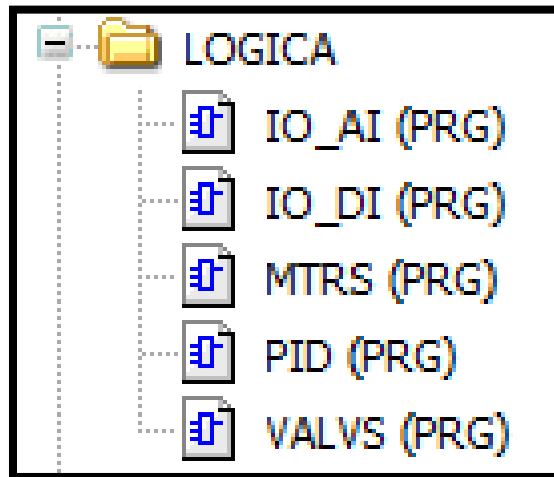
### 2.1.5 Creación de lógica de control

Para la creación de la lógica de control del proceso, se debe crear una carpeta en la sección “Application”, la cual contendrá toda la lógica de control del proceso. Esto se muestra en la Figura 2-22.

En este punto, se describirán los programas creados dentro de esta carpeta, que incluyen:

- **IO\_AI:** Manejo de entradas analógicas.
- **IO\_DI:** Manejo de entradas digitales.
- **MTRS:** Control de motores On/Off.
- **PID:** Configuración y control de procesos mediante PID.
- **VALVS:** Control de válvulas On/Off.

Esta organización facilita la estructuración del proyecto y mejora la comprensión y el mantenimiento del código.



Fuente: Elaboración propia

Figura 2-22 Carpeta LÓGICA.

En la carpeta creada “LOGICA”, se crear un Program Organization Unit (POU), se especifican las funciones del POU, el tipo de componente que representará (en este caso, será un Programa) y el lenguaje de programación a utilizar.

Entre las opciones disponibles para el lenguaje de programación se encuentran:

- Continuous Function Chart (CFC)
- Diagrama de Bloques Funcionales (FBD)
- Diagrama de Contactos (LD)
- Diagrama de Funciones Continuas (CFC)
- Diagrama Funcional Secuencial (SFC)
- Texto Estructurado (ST)

Para este proyecto, los programas creados utilizarán el lenguaje Diagrama de Bloques Funcionales (FBD), que es adecuado para desarrollar la lógica de control de manera gráfica y clara.

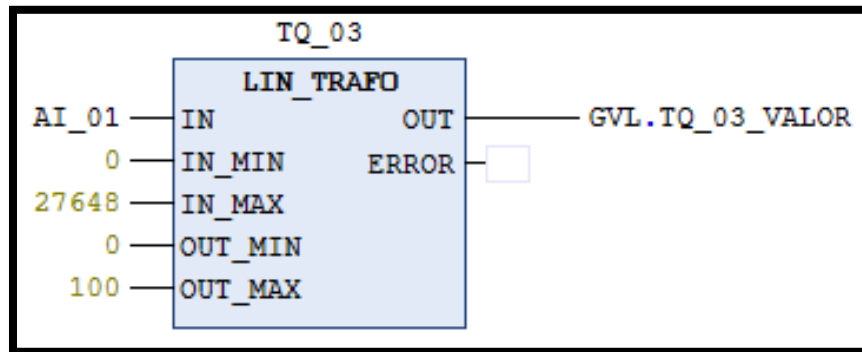
a) IO\_AI

En el programa IO\_AI creado, se gestionará la recepción de las variables de entrada analógicas y su respectiva escala para que puedan ser utilizadas en el control del proceso.

Se tiene en cuenta que el valor mínimo de las variables analógicas es 0, y el valor máximo es 27648. Para realizar esta transformación, se utiliza la macro LIN\_TRAFO, la cual escala los valores de entrada a un rango de salida comprendido entre 0 y 100.

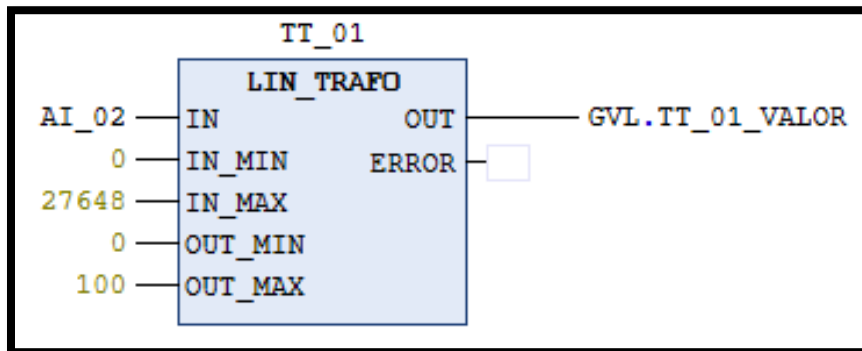
Esto asegura que las señales analógicas puedan interpretarse adecuadamente en las operaciones posteriores del sistema.

En las Figuras 2-23 y 2-24, se pueden observar las señales correspondientes al valor del nivel TQ-03 y la temperatura TT\_01.



Fuente: Elaboración propia

Figura 2-23 TQ-03.



Fuente: Elaboración propia

Figura 2-24 TT-01

## b) IO\_DI

En el programa IO\_DI creado, se gestionará la recepción de las variables de entrada digitales, las cuales serán procesadas para que puedan ser utilizadas en el control del proceso. Este programa se encargará de recibir las señales digitales y garantizar que los valores sean adecuados para la lógica de control del sistema, permitiendo su integración efectiva en el proceso de automatización.

Se asignan los DI respectivamente, como se muestra a continuación en la Figura 2-25.

DI_01	—	GVL.START_STOP
DI_02	—	GVL.PE_STOP
DI_03	—	GVL.PRES_FLU_A
DI_04	—	GVL.PRES_FLU_B
DI_05	—	GVL.LT_01_LL
DI_06	—	GVL.LT_02_LL
DI_07	—	GVL.LT_03_HH
DI_08	—	GVL.MODE_MANT

Fuente: Elaboración propia

Figura 2-25 Variables digitales de entrada.

c) MTRS

En el programa MTRS desarrollado, se gestiona el funcionamiento de las tres bombas del sistema.

**Bomba MT-01:** La bomba MT-01 operará en modo automático siempre que se cumplan las siguientes condiciones:

- Haya presencia de fluido A FS-01.
- El nivel del tanque TQ-01 no esté por debajo del límite establecido (ZSL-01).

- El proceso se encuentre en funcionamiento.
- La válvula FV-01 esté abierta.

La bomba MT-01 se detendrá si se presentan las siguientes condiciones:

- El nivel del tanque 3 supera el 90 %.
- La temperatura excede los 50 °C.
- El comando de parada (STOP) está activo.
- La válvula FV-01 está cerrada.

En caso de emergencia la bomba MT-01 se detendrá con la señal PE\_STOP, la cual corresponde a la parada de emergencia.

**Bomba MT-02:** La bomba MT-02 tiene un funcionamiento similar a la bomba MT-01 con la diferencia que esta operará en modo automático siempre que se cumplan las siguientes condiciones:

- Haya presencia de fluido B FS-02.
- El nivel del tanque TQ-02 no esté por debajo del límite establecido ZSL-02.
- El proceso se encuentre en funcionamiento.
- La válvula FV-02 esté abierta.

La bomba MT-02 se detendrá si se presentan las siguientes condiciones:

- El nivel del tanque 3 supera el 90 %.
- La temperatura excede los 50 °C.
- El comando de parada (STOP) está activo.
- La válvula FV-02 está cerrada.

En caso de emergencia la bomba MT-02 se detendrá con la señal PE\_STOP, la cual corresponde a la parada de emergencia.

**Bomba MT-03:** La bomba MT-03 operará en modo automático siempre que se cumplan las siguientes condiciones:

- La bomba MT-01 esté funcionando.
- La bomba MT-02 esté funcionando.
- El nivel del tanque TQ-03 no esté por debajo del límite establecido.
- La temperatura esté por encima de los 30 °C.

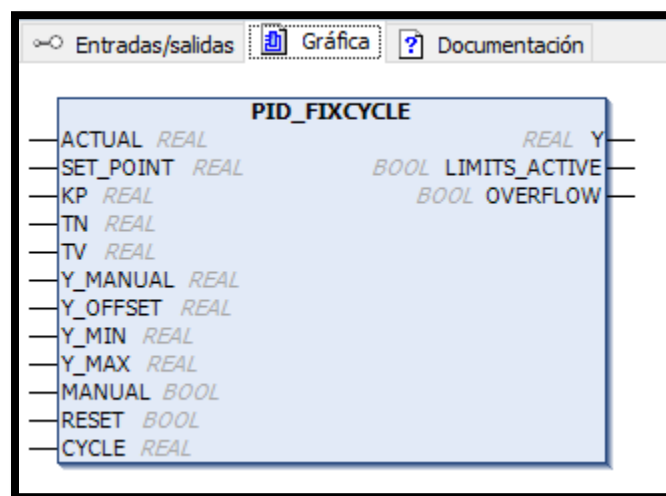
La bomba MT-03 se detendrá si se presentan las siguientes condiciones:

- La bomba MT-01 no estén funcionando.
- La bomba MT-02 no estén funcionando.
- El nivel del tanque 3 esté por debajo del 20 %.

En caso de emergencia la bomba MT-03 se detendrá con la señal PE\_STOP, la cual corresponde a la parada de emergencia.

#### d) PID

El controlador PID, mostrado en la figura 2-26, aplica una corrección basada en términos proporcionales, integrales y derivativos (denominados P, I y D respectivamente) que dan su nombre al tipo de controlador.



Fuente: Elaboración propia

Figura 2-26 Bloque PID a utilizar.

El término **P** representa los valores actuales del error. Por ejemplo, si el error es grande y positivo, la salida del controlador también será proporcionalmente grande y positiva.

El término **I** Este término tiene en cuenta el acumulado del error a lo largo del tiempo. Si el error persiste porque la acción proporcional no es suficiente, el término integral se incrementará continuamente hasta corregir el error por completo. Esto asegura que el sistema alcance finalmente el valor deseado, eliminando el error residual o "offset" que puede quedar cuando solo se usa el término proporcional.

El término **D** evalúa las posibles tendencias futuras del error, basándose en su tasa de cambio actual.

Dado que un controlador PID se basa únicamente en la medición de la variable de proceso y no en el conocimiento específico del proceso subyacente, su aplicación es extremadamente versátil. Al ajustar los tres parámetros del modelo (P, I y D), un controlador PID puede adaptarse a los requisitos específicos de diferentes procesos.

La respuesta del controlador se puede analizar en términos de su capacidad para responder a un error, el grado en que el sistema puede sobrepasar un punto de ajuste y la magnitud de cualquier oscilación del sistema. Sin embargo, es importante señalar que el uso del algoritmo PID no garantiza necesariamente un control óptimo ni la estabilidad del sistema.

La salida, representada como **Y**, es la variable manipulada y se calcula de la siguiente manera:

$$Y = KP \left( e + \frac{1}{TN} \int e dt + TV \frac{de}{dt} \right) + Y_{OFFSET}$$

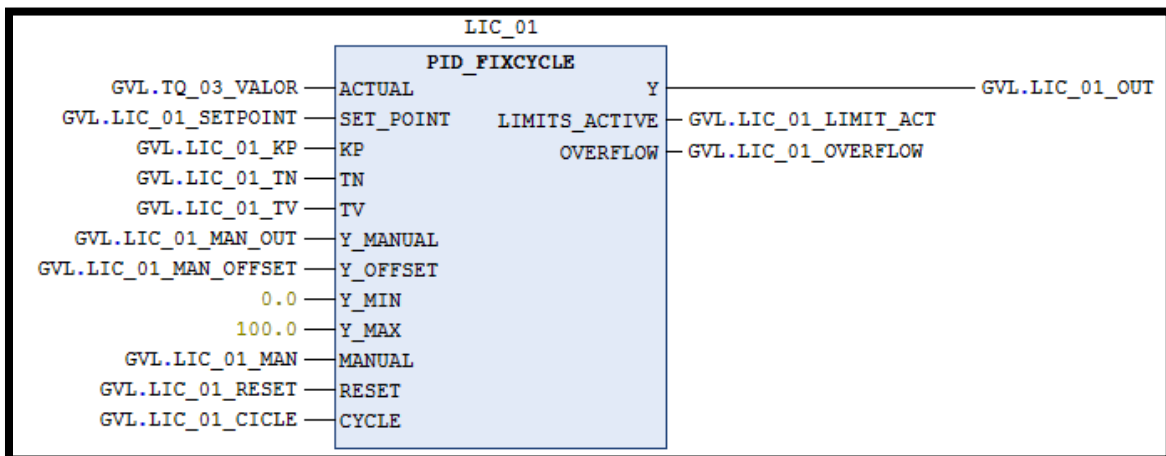
Para realizar la sintonización del sistema se utiliza el método empírico (prueba y error) Este método es adecuado cuando no se dispone de información detallada del sistema y los métodos analíticos no son posibles.

1. Incrementa KP gradualmente hasta observar una respuesta rápida pero estable.
2. Ajusta TN para eliminar el error estacionario (sin sobrepaso excesivo).
3. Introduce TV para mitigar oscilaciones.

En el programa PID desarrollado, se gestiona el control del nivel del tanque 3 (TQ-03). Este controlador opera de acuerdo con los parámetros seleccionados desde las pantallas de operación, los parámetros a intervenir en el cálculo del bloque PID son los siguientes:

- SET\_POINT.
- KP
- TN
- TV

Los cuáles serán escritos por el operador atreves de la interfaz HMI. En le bloque mostrado en la figura 2-27.



Fuente: Elaboración propia

Figura 2-27 LIC-01

#### e) VALVS

En el programa VALVS desarrollado, se gestiona el funcionamiento de las dos válvulas ON/OFF del sistema.

**La Válvula FV-01:** operará en modo automático siempre que se cumplan las siguientes condiciones:

- Haya presencia de fluido A.

- El nivel del tanque TQ-01 no esté por debajo del límite establecido.
- El proceso se encuentre en funcionamiento.

La Válvula VLV-01 se detendrá si se presentan las siguientes condiciones:

- El nivel del tanque 3 supera el 90 %.
- La temperatura excede los 50 °C.
- El comando de parada (STOP) está activo.

En caso de emergencia la Válvula FV-01 se cerrará con la señal PE\_STOP, la cual corresponde a la parada de emergencia. La lógica Válvula FV-01 se muestra en la figura 2-35 a continuación.

**La Válvula FV-02** operará en modo automático siempre que se cumplan las siguientes condiciones:

- Haya presencia de fluido B.
- El nivel del tanque TQ-02 no esté por debajo del límite establecido.
- El proceso se encuentre en funcionamiento.

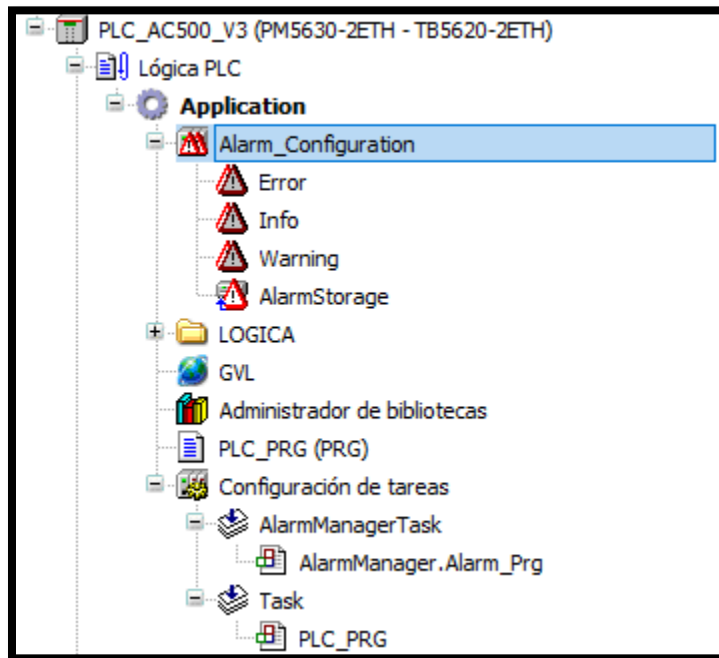
La Válvula FV-02 se detendrá si se presentan las siguientes condiciones:

- El nivel del tanque 3 supera el 90 %.
- La temperatura excede los 50 °C.
- El comando de parada (STOP) está activo.

En caso de emergencia la Válvula FV-02 se cerrará con la señal PE\_STOP, la cual corresponde a la parada de emergencia. La lógica Válvula VLV-02 se muestra en la figura 2-36.

### 2.1.6 Configuración de alarmas

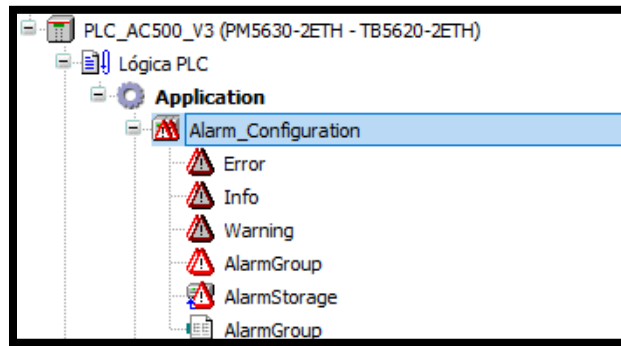
Es necesario agregar las configuraciones de las alarmas dentro del PLC AC500. Para ello, se debe crear la configuración de alarmas en la sección “Application”, la cual incluirá todas las alarmas y los grupos de alarmas del proceso. Esto se ilustra en la Figura 2-28.



Fuente: Elaboración propia

Figura 2-28 Alarm\_Configuration.

Se definen los grupos de alarmas dentro de esta configuración, como se muestra en la Figura 2-29 y Figura 2-30.



Fuente: Elaboración propia

Figura 2-29 Alarm\_Configuration, grupos de alarmas.

Después de crear el objeto Grupo de Alarmas, se debe configurar cada grupo. En este caso, se configuran dos grupos: ERROR y WARNING.

- Los errores se mostrarán en el listado de alarmas con letras de color rojo.
- Las advertencias (Warnings) se reflejarán con un color verde azulado.

Luego de configurar cada grupo de alarmas, es necesario asignar las alarmas a su grupo correspondiente. En este caso, se configuraron alarmas relacionadas con nivel, temperatura, parada de emergencia, fallas en motores, válvulas, presencia de fluidos, entre otras.

ID	Tipo de supervisión	Detalles	Desactivación	Clase	Mensaje
0	Digital	(GVL.LT_01_HH) = (TRUE)		Error	NIVEL TQ-01 MUY ALTO
1	Digital	(GVL.LT_01_LL) = (TRUE)		Error	NIVEL TQ-01 MUY BAJO
2	Digital	(GVL.LT_02_LL) = (TRUE)		Error	NIVEL TQ-02 MUY BAJO
3	Digital	(GVL.LT_03_HH) = (TRUE)		Error	NIVEL TQ-03 MUY ALTO
4	Digital	(GVL.MT_01_ERROR) = (TRUE)		Error	MT-01 FALLA
5	Digital	(GVL.MT_02_ERROR) = (TRUE)		Error	MT-02 FALLA
6	Digital	(GVL.MT_03_ERROR) = (TRUE)		Error	MT-03 FALLA
7	Digital	(GVL.PE_STOP) = (TRUE)		Error	PARADA DE EMERGENCIA
8	Digital	(GVL.TT_01_H) = (TRUE)		Warning	TEMPERATURA ALTA
9	Digital	(GVL.VLV_01_ERROR) = (TRUE)		Error	FV-01 FALLA
10	Digital	(GVL.VLV_02_ERROR) = (TRUE)		Error	FV-02 FALLA
11	Límite superior	GVL.TQ_03_VALOR > 90.0		Error	NIVEL TQ-03 MUY ALTO
12	Límite superior	GVL.TQ_03_VALOR > 80		Warning	NIVEL TQ-03 ALTO
13	Límite inferior	GVL.TQ_03_VALOR < 20		Warning	NIVEL TQ-03 BAJO
14	Límite inferior	GVL.TQ_03_VALOR < 10		Error	NIVEL TQ-03 MUY BAJO
15	Límite superior	GVL.TT_01_VALOR > 50		Error	TEMPERATURA TQ-03 MUY ALTA
16	Límite superior	GVL.TT_01_VALOR > 45		Warning	TEMPERATURA TQ-03 ALTA

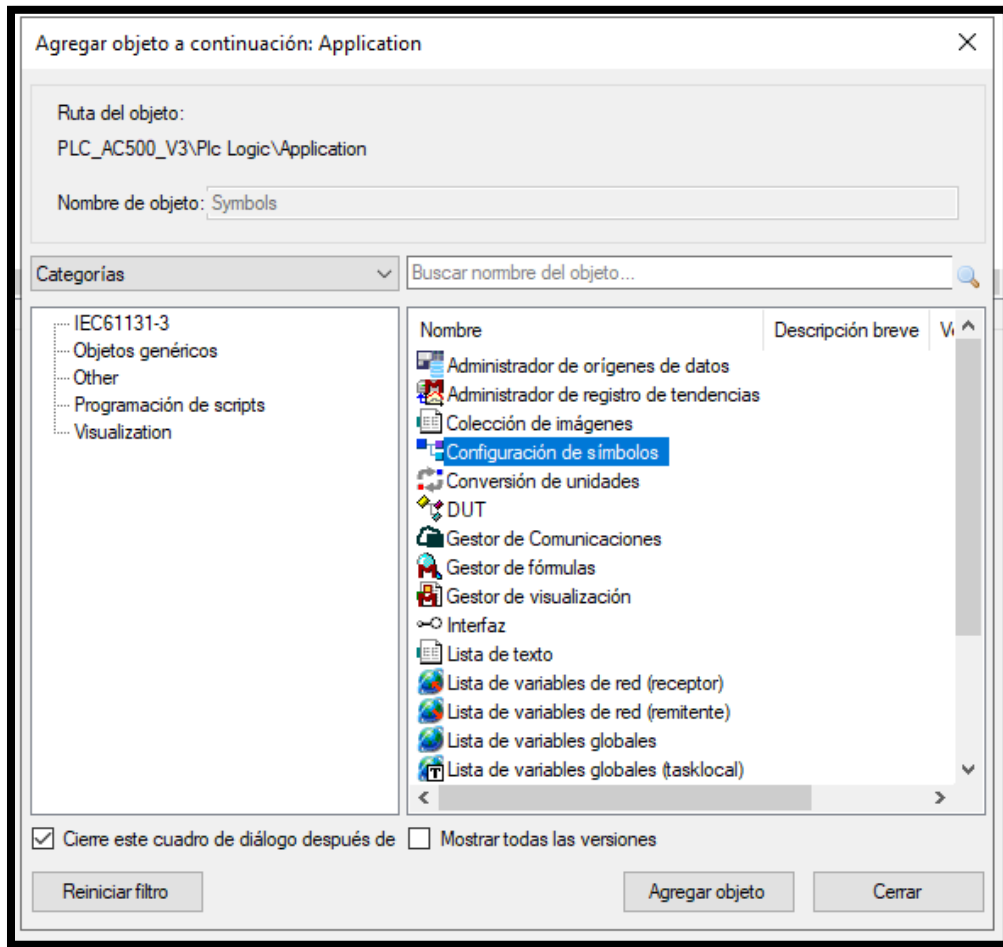
Fuente: Elaboración propia

Figura 2.30 AlarmGroup

### 2.1.7 Configuración de símbolos

Después de realizar todas las configuraciones mencionadas en este documento, es necesario crear las configuraciones de símbolos, ya que éstas permiten que el PLC AC500 se comunique con la HMI CP6607 mediante el protocolo CODESYS V3 ETH que se basa en protocolos TCP/IP.

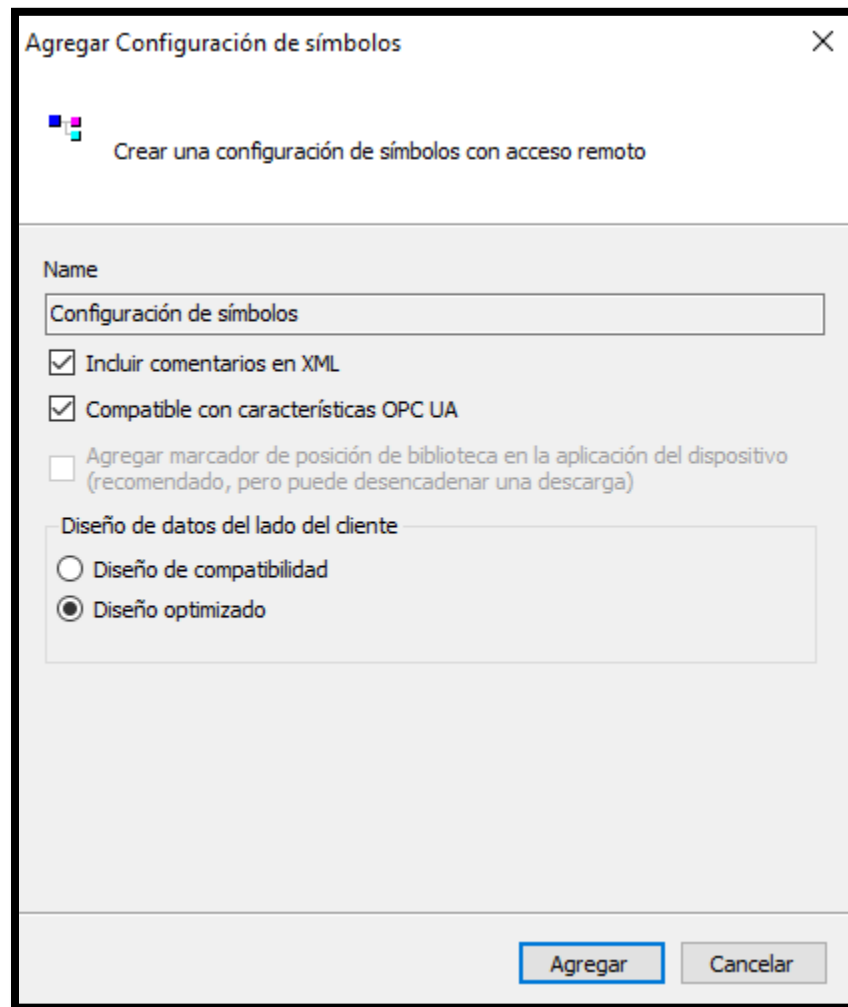
Para ello, se debe crear la configuración de símbolos en la sección “Application”, incluyendo todas las variables del proceso que se comunicarán con la HMI. Esto se ilustra en la Figura 2-31.



Fuente: Elaboración propia

Figura 2-31 Agregar objeto - Configuración de símbolos.

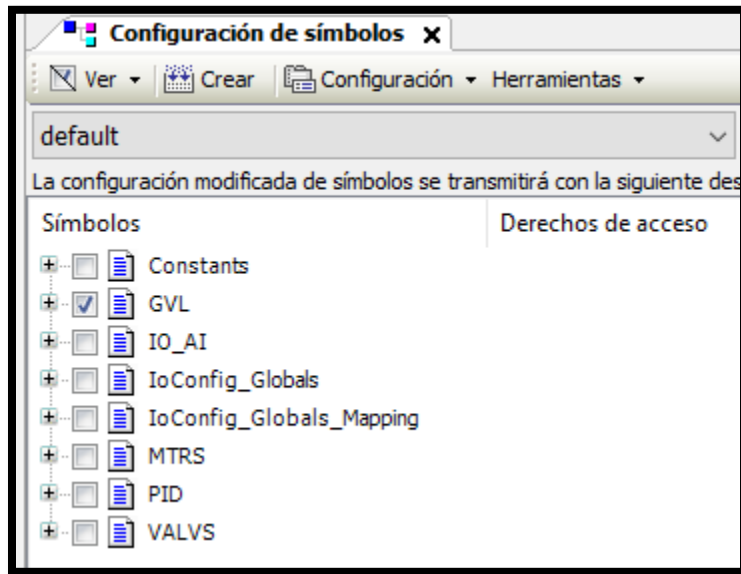
Una vez que se agrega el objeto, aparecerá la pestaña mostrada en la Figura 2-32. En esta sección, es importante seleccionar la opción Incluir comentarios XML, ya que se generará un documento en dicho formato, el cual deberá ser importado en las variables de la HMI.



Fuente: Elaboración propia

Figura 2-32 Configuración de símbolos.

Se deben abrir las **Configuraciones de Símbolos** y seleccionar el grupo de variables que se desea comunicar con la HMI. En este caso, se exportarán las variables globales (**GVL**) como se muestra en la Figura 2-33.



Fuente: Elaboración propia

Figura 2-33 Configuración de símbolos, seleccionar GVL.

Para finalizar las configuraciones de símbolos, una vez que se compile el programa y se genere el código, se creará el archivo XML en la carpeta donde se guarda el proyecto.

#### 2.1.8 Comunicación de datos

A continuación, se describen los dos tipos de comunicación implementados en este proyecto:

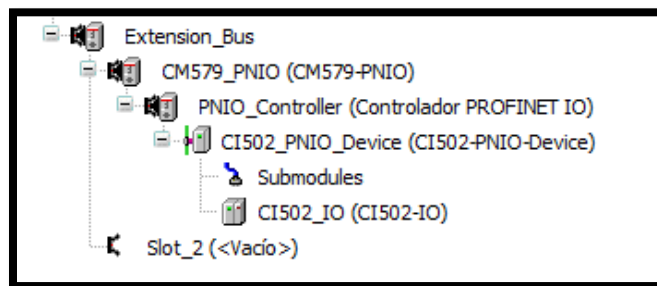
- **Profinet:** Este protocolo se utilizará para establecer comunicación entre el PLC AC500 y un módulo Profinet IO, que contiene un IO remoto. Esta configuración simula la comunicación con un gabinete de IO ubicado en el terreno, permitiendo el manejo de señales de entrada y salida desde un entorno descentralizado.
- **IEC 104:** Este protocolo se empleará para transmitir datos a un servidor. Para implementar esta comunicación, se utilizará el software KEPServer, cuya configuración y aplicación se detallará más adelante en este documento.

Estos protocolos son clave para garantizar la interoperabilidad y el flujo eficiente de datos dentro del sistema automatizado.

a) PROFINET IO

Para la configuración de Profinet IO, es necesario acceder al módulo PNIO\_Controller, que es donde se establecerá el rango de direcciones IP que se utilizarán en esta comunicación.

Este paso es fundamental para garantizar que los dispositivos conectados puedan comunicarse correctamente dentro de la red Profinet, asignando direcciones IP coherentes con la topología del sistema.

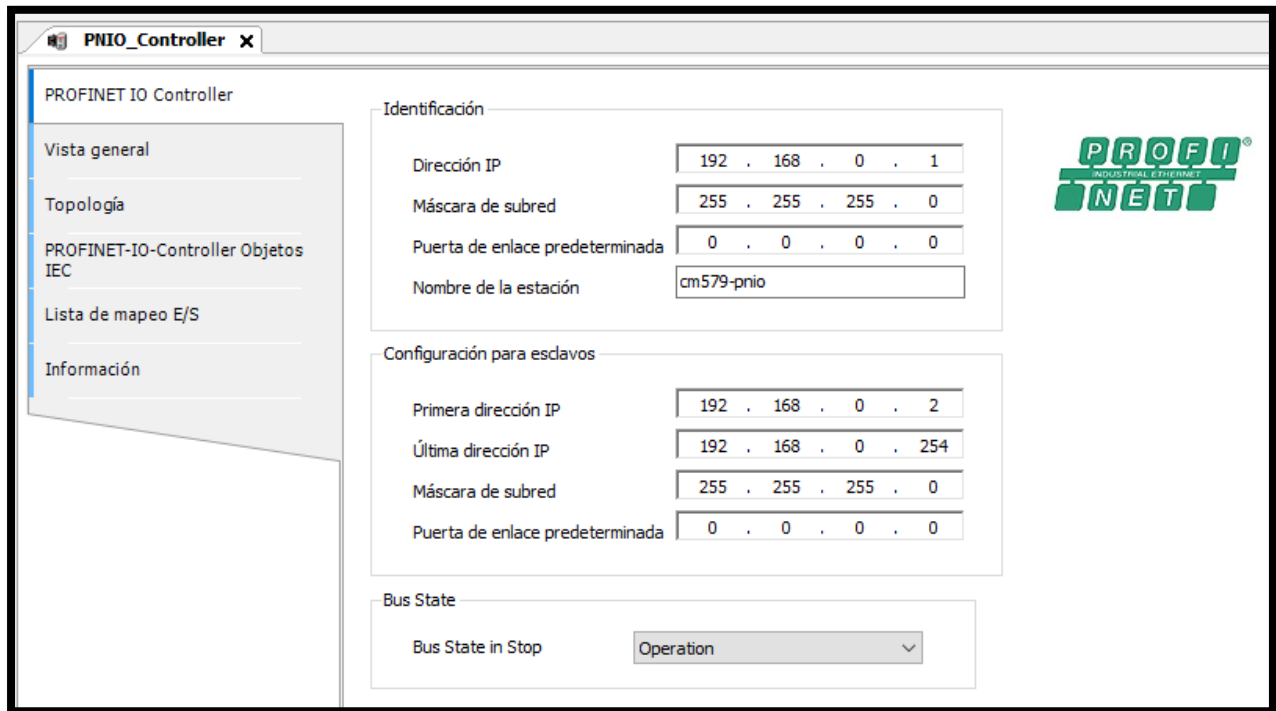


Fuente: Elaboración propia

Figura 2-34 Extension\_Bus.

En la Figura 2-58, se ilustra el rango de direcciones IP asignadas para los esclavos Profinet. En este caso, se utilizarán las IPs clase C entre 192.168.0.2 y 192.168.0.254.

Este rango define las direcciones disponibles para los dispositivos conectados al controlador Profinet, asegurando una comunicación organizada y evitando conflictos de direcciones en la red.

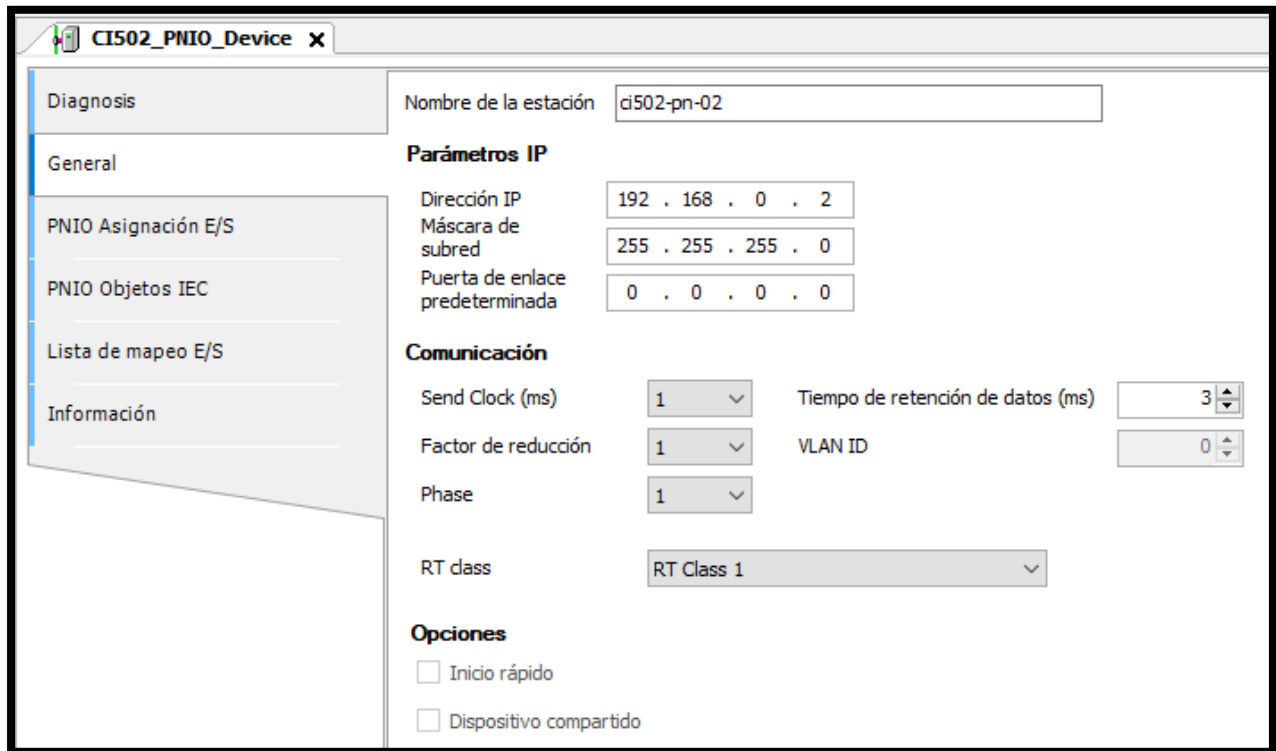


Fuente: Elaboración propia

Figura 2-35 PNIO\_Controller

Después de configurar el rango de direcciones IP, se debe acceder al módulo CI502\_PNIO\_Device, donde se definirán la dirección IP del esclavo y su nombre de dispositivo.

Es importante destacar que el nombre del módulo Profinet debe ser exactamente "ci502-pn-02". El número 02 dentro del nombre es crucial, ya que define la dirección de hardware del esclavo bajo el maestro en la red Profinet. Esto asegura una identificación única y precisa del dispositivo dentro del sistema.



Fuente: Elaboración propia

Figura 2-36 CI502\_PNIO\_Device.

#### b) IEC104

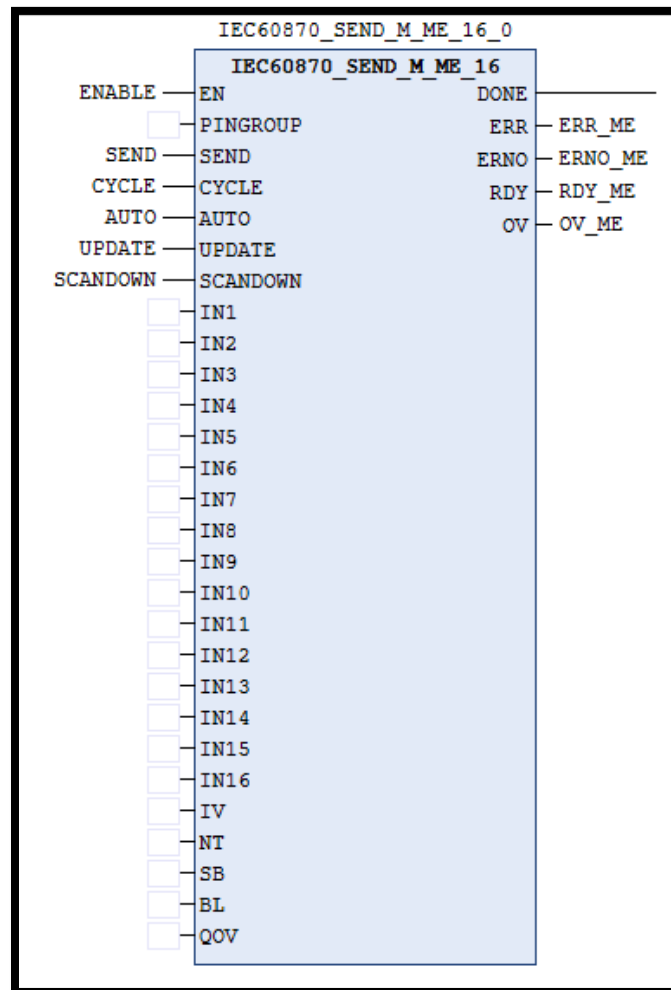
Para establecer la comunicación utilizando el protocolo IEC 104, es fundamental agregar la librería correspondiente, IEC 60870-5-104, en el administrador de bibliotecas del proyecto.

Esta librería proporciona las funciones y bloques necesarios para implementar la comunicación con el servidor, asegurando que los datos sean transmitidos y recibidos conforme al estándar IEC 104.

Una vez agregada la librería IEC 60870-5-104, se procederá a crear un programa específico para configurar, mediante bloques de función, las señales que se desean comunicar. Este programa permitirá asignar y mapear las señales requeridas, garantizando que los datos se transmitan correctamente al servidor conforme al protocolo establecido.

Para la comunicación de datos reales, se debe insertar el bloque de función IEC60870\_SEND\_M\_ME\_16 dentro del programa mostrado en la Figura 2-37.

Este bloque es utilizado para transmitir valores analógicos en formato de punto flotante (real), cumpliendo con el estándar del protocolo IEC 104.

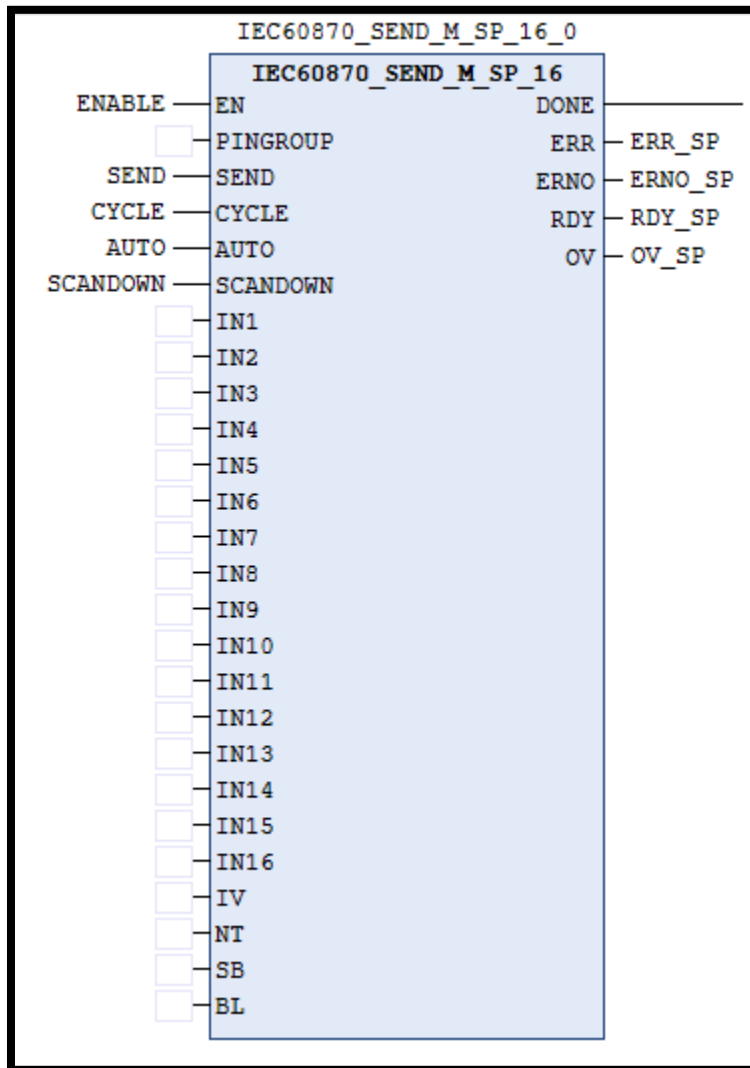


Fuente: Elaboración propia.

Figura 2-37 Bloque de función IEC60870\_SEND\_M\_ME\_16

Para la transmisión de datos booleanos, se debe insertar el bloque de función IEC60870\_SEND\_M\_SP\_16 dentro del programa mostrado en la Figura 2-38.

Este bloque se utiliza para enviar señales digitales (ON/OFF) o valores booleanos bajo el protocolo IEC 104.

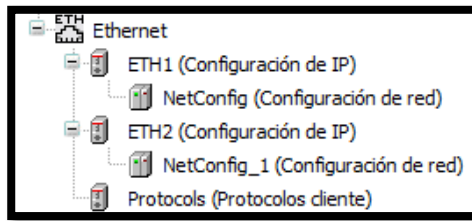


Fuente: Elaboración propia.

Figura 2-38 Bloque de función IEC60870\_SEND\_M\_SP\_16.

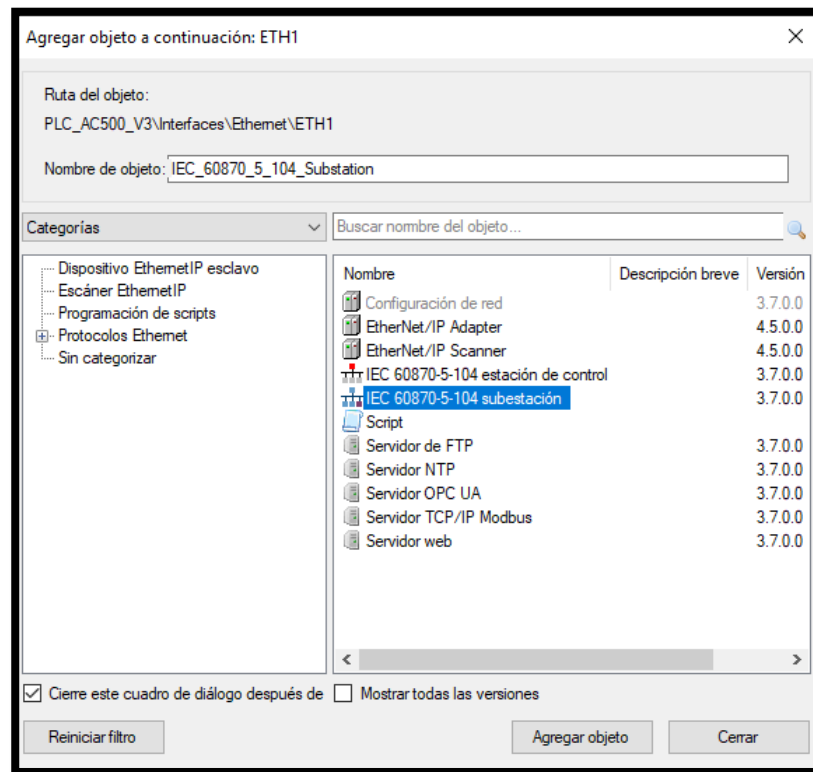
En el puerto correspondiente a la comunicación, es necesario configurar el protocolo a utilizar mostrado en la figura 2-39 los puertos. Para este caso, se seleccionará el protocolo "IEC 60870-5-104 Subestación". Mostrado en la figura 2-40.

Esta configuración permite que el sistema se comporte como una subestación bajo el estándar IEC 104, habilitando la comunicación bidireccional con el servidor y garantizando la correcta transmisión y recepción de datos según lo requerido.



Fuente: Elaboración propia.

Figura 2-39 Ethernet AC500.



Fuente: Elaboración propia.

Figura 2-40 IEC 60870-5-104 Subestación.

Una vez agregado el protocolo "IEC 60870-5-104 Subestación", es necesario configurar la capa de enlace, mostrada en la figura 2-41.

En esta sección, se debe especificar la dirección IP del equipo receptor que procesará las señales enviadas desde el sistema. Esta configuración asegura que los datos sean transmitidos al destino correcto dentro de la red, cumpliendo con los requisitos del protocolo IEC 104.

Fuente: Elaboración propia.

Figura 2-41 Capa de enlace IEC104

En la sección de Objetos de Información mostrada en la Figura 2-42, se configuran las señales que se desean enviar mediante el protocolo IEC 104.

En esta etapa, se asignan las direcciones de los objetos y las variables asociadas a las señales, definiendo los puntos de comunicación específicos para cada dato que será transmitido al servidor. Esta configuración es clave para garantizar la correcta correspondencia entre las señales del sistema y los datos recibidos en el destino.

Nombre ASDU	Tipo de datos	Tipo ASDU	Direc común	Obj info direc	Inicio normalizado	Fin normalizado	Umbral	Descripción	ID
-------------	---------------	-----------	-------------	----------------	--------------------	-----------------	--------	-------------	----

Fuente: Elaboración propia.

Figura 2-42 Objetos de Información.

En la sección de Objetos de Información, se debe hacer clic derecho sobre el panel y seleccionar la opción "Agregar objeto de información ASDU".

A continuación, se elige el tipo de objeto de información a configurar. Para este caso, se seleccionaron dos tipos de objetos de información con ASDU:

- (30) M\_SP\_TB\_1: Utilizado para la asignación de variables booleanas (señales digitales ON/OFF).
- (13) M\_ME\_NC\_1: Utilizado para la asignación de variables reales (valores analógicos).

Estos objetos permiten mapear las señales del sistema de forma adecuada para la transmisión de datos según las especificaciones del protocolo IEC 104, esta configuración se muestra en la Figura 2-43.

Nombre ASDU	Tipo de datos	Tipo ASDU	Direc común	Obj info direc
DATA_BOOL_01	IEC60870_SinglePointInformation	(30) M_SP_TB_1	10.0	1.0.0
DATA_BOOL_01	IEC60870_SinglePointInformation	(30) M_SP_TB_1	10.0	2.0.0
DATA_BOOL_01	IEC60870_SinglePointInformation	(30) M_SP_TB_1	10.0	3.0.0
DATA_BOOL_01	IEC60870_SinglePointInformation	(30) M_SP_TB_1	10.0	4.0.0
DATA_BOOL_01	IEC60870_SinglePointInformation	(30) M_SP_TB_1	10.0	5.0.0

Fuente: Elaboración propia.

Figura 2-43 Mapeo de señales.

A cada uno de los objetos de información ASDU agregados, se les asigna un nombre único, que posteriormente se integrará en los bloques de funciones correspondientes para enlazar las señales del bloque al objeto configurado.

Además, se debe configurar para cada objeto:

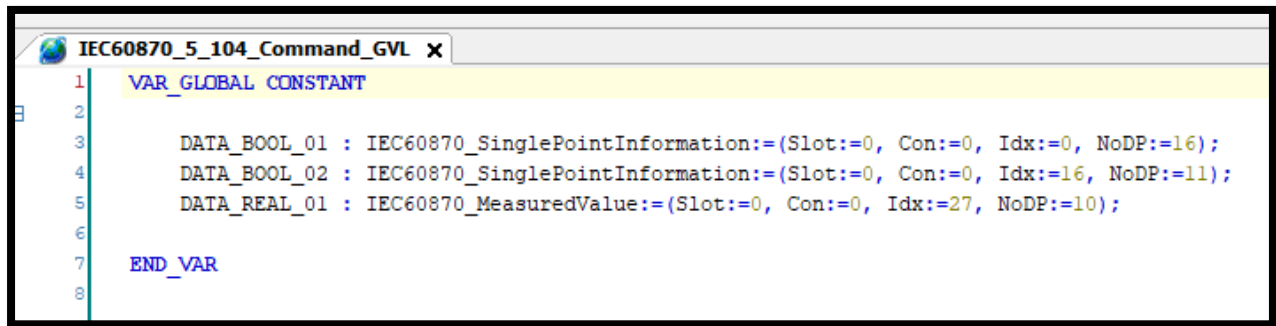
- **Dirección común:** Identifica el grupo o dispositivo en la red al que pertenece el objeto.
- **Dirección de objeto:** Especifica la ubicación exacta del objeto dentro del sistema.

Estas configuraciones son esenciales para garantizar una lectura correcta y precisa de la información transmitida, alineándose con los requerimientos del protocolo IEC 104.

Luego de completar las configuraciones de las variables y compilar el proyecto, se deben agregar dos asignaciones de variables globales:

- **IEC60870\_5\_104\_Command\_GVL:** En esta asignación, los nombres de los objetos ASDU configurados se agregan automáticamente. Esto facilita la integración y gestión de los objetos en la lógica de programación, mostrada en la Figura 2-44.
- **IEC60870\_5\_104\_Connection\_GVL:** En esta asignación, se especifica la conexión utilizada para la comunicación. En este caso, el objeto encargado de la comunicación es "IEC\_60870\_5\_104\_Substation", mostrada en la Figura 2-45.

Estas asignaciones son cruciales para establecer los vínculos entre las configuraciones realizadas y los bloques funcionales que implementan la comunicación en el sistema.



```
1  VAR_GLOBAL CONSTANT
2
3      DATA_BOOL_01 : IEC60870_SinglePointInformation:=(Slot:=0, Con:=0, Idx:=0, NoDP:=16);
4      DATA_BOOL_02 : IEC60870_SinglePointInformation:=(Slot:=0, Con:=0, Idx:=16, NoDP:=11);
5      DATA_REAL_01 : IEC60870_MeasuredValue:=(Slot:=0, Con:=0, Idx:=27, NoDP:=10);
6
7  END_VAR
8
```

Fuente: Elaboración propia.

Figura 2-44 IEC60870\_5\_104\_Command\_GVL

```

1  {attribute 'qualified_only'}
2
3  VAR_GLOBAL CONSTANT
4
5      IEC_60870_5_104_Substation : IEC60870_5_104_Connection:=(Slot:=0, Con:=0);
6
7  END_VAR
8

```

Fuente: Elaboración propia.

Figura 2-45 IEC60870\_5\_104\_Connection\_GVL

En el bloque de funciones correspondiente a las señales análogas, en el pin "PINGROUP" se debe ingresar el nombre del objeto ASDU correspondiente.

## 2.2 PANEL 600

A continuación, se presentará la creación de la programación necesaria para la HMI CP6607, considerando los siguientes aspectos clave:

- Creación de Proyecto.
- Configuración de protocolo de comunicación con PLC AC500.
- Asignación de variables Panel
- Pantallas de operación
- Creación de usuarios y accesos

### 2.2.1 Creación de Proyecto

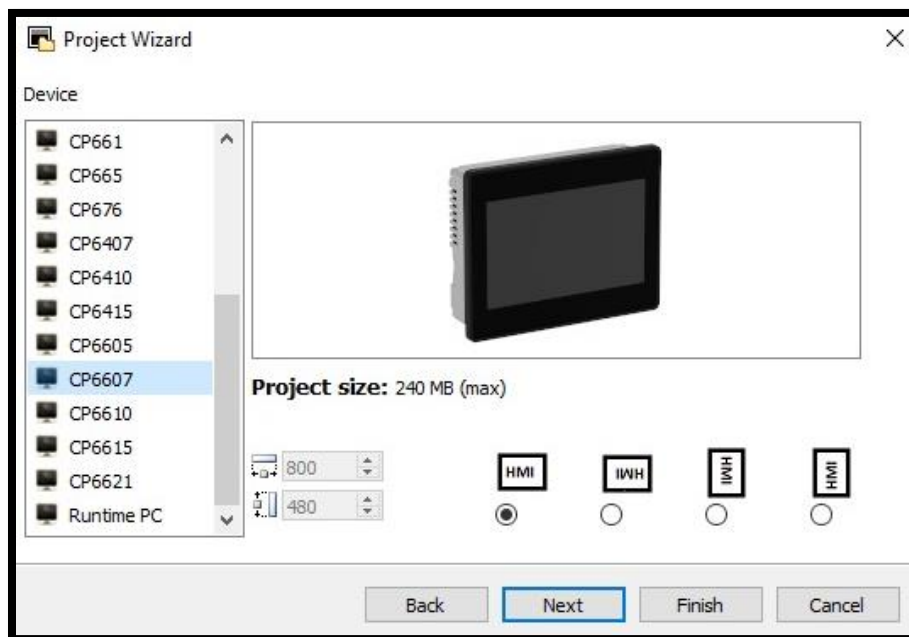
Para la creación del proyecto en Panel Builder 600, se debe abrir el programa, lo que mostrará la pestaña de la Figura 2-46. En esta pestaña, se debe seleccionar el nombre del proyecto y la ubicación donde se almacenará el proyecto. También seleccionar el modelo de HMI que se utilizará, mostrado en la Figura 2-47 en este caso se utilizará el modelo CP6607. Esta es la

primera configuración que establece la base para desarrollar la interfaz HMI (Interfaz Hombre-Máquina).



Fuente: Elaboración propia.

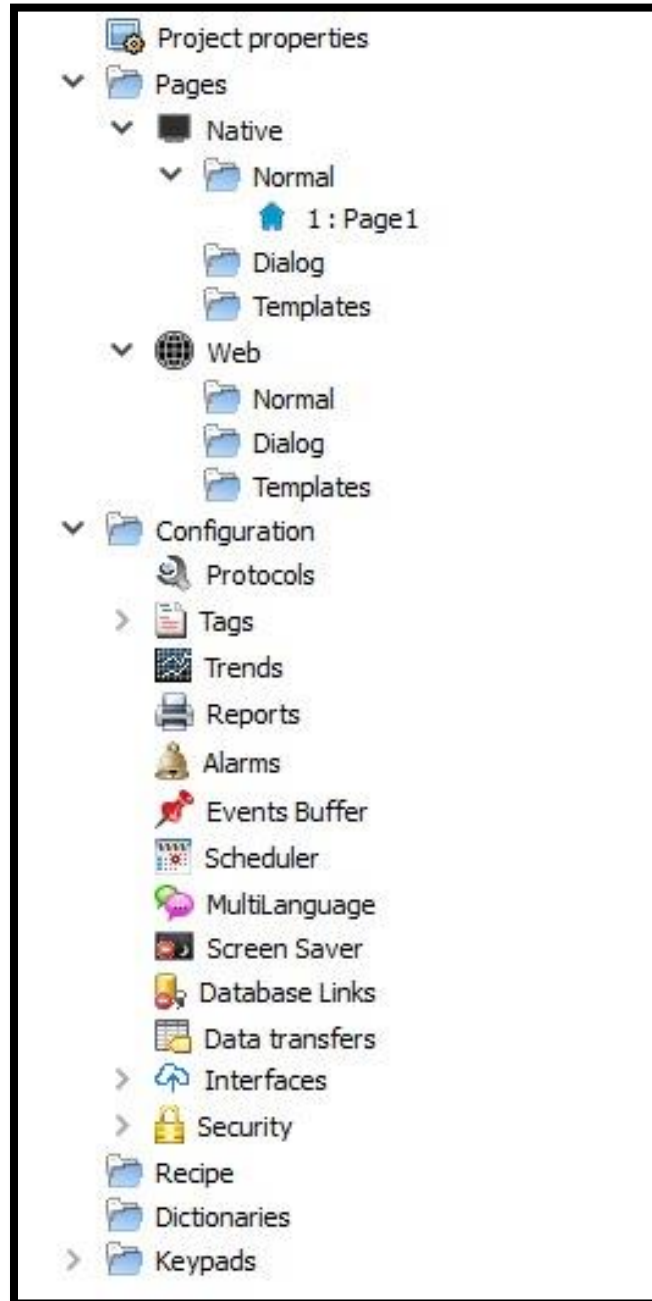
Figura 2-46 Creación del proyecto.



Fuente: Elaboración propia.

Figura 2-47 Selección Panel HMI.

Una vez creado el proyecto en Panel Builder 600, se desplegará la siguiente información, la cual se muestra en la Figura 2-48.



Fuente: Elaboración propia.

Figura 2-48 Herramientas del Proyecto

## 2.2.2 Configuración de protocolo de comunicación con PLC AC500

Una vez creado el proyecto en Panel Builder 600, se procede a realizar las configuraciones necesarias para la comunicación con el PLC AC500.

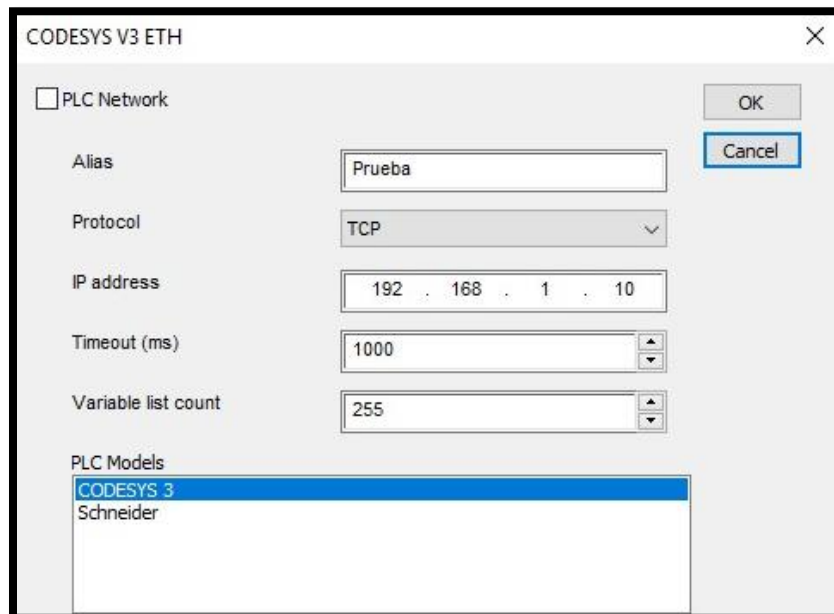
Para ello, se debe dirigir a Protocols, que se encuentra en la sección de Configuration, como se muestra en la Figura 2-49.

En esta sección, se configurará el protocolo de comunicación que permitirá al HMI intercambiar datos con el PLC AC500, asegurando una correcta transmisión de señales y datos entre ambos dispositivos.

Para esta ocasión, se utilizará el protocolo CODESYS V3 ETH, un protocolo basado en TCP/IP que permitirá la comunicación entre el HMI y el PLC AC500.

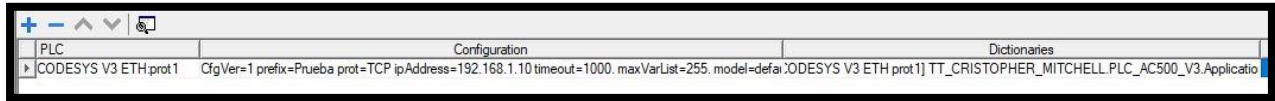
En esta configuración, se deberán especificar los parámetros del PLC, como su dirección IP y otros detalles de conexión. Esta configuración se muestra en la Figura 2-50.

Una vez configurado el protocolo, el HMI podrá establecer la comunicación con el PLC AC500 y acceder a las variables y datos necesarios para interactuar con el sistema.



Fuente: Elaboración propia.

Figura 2.49 CODESYS V3 ETH



Fuente: Elaboración propia.

Figura 2.50 Protocols Panel Builder 600.

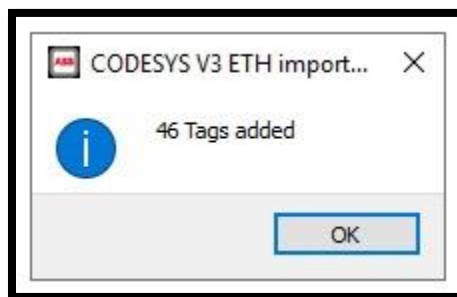
### 2.2.3 Asignación de variables Panel

Para la asignación de variables hacia la HMI desde el PLC AC500, se debe dirigir a Tags en el Panel Builder 600.

En esta sección, se procederá a cargar el archivo XML que se creó previamente durante las configuraciones en Automation Builder. Este archivo XML contiene las variables necesarias para el proceso y establece la comunicación entre el PLC y la HMI.

Una vez cargado el archivo, las variables del PLC estarán disponibles para ser utilizadas en la interfaz HMI, permitiendo su visualización y manipulación según las necesidades del proceso.

Una vez cargadas las variables se mostrará la pestaña de la Figura 2-51, identificando el número de Tags que contiene el archivo y se desplegará el listado de variables mostrado en la Figura 2-52.



Fuente: Elaboración propia.

Figura 2-51 Número de Tags agregados.

CODESYS V3 ETH:prot1		Container
Alias: Prueba		
Model: CODESYS 3		
Application/AlarmManager/AlarmGlobals/g_diAlarmStateTransitionCounter	int	Prueba/Application/AlarmManager/AlarmGlobals/g_diAlarmStateTransitionCounter
Application/AlarmManager/AlarmGlobals/g_diAlarmStorageCounter	int	Prueba/Application/AlarmManager/AlarmGlobals/g_diAlarmStorageCounter
Application/AlarmManager/AlarmGlobals/g_diAlarmStorageRowsCount	int	Prueba/Application/AlarmManager/AlarmGlobals/g_diAlarmStorageRowsCount
Application/AlarmManager/AlarmGlobals/g_iAlarmStorageClear	short	Prueba/Application/AlarmManager/AlarmGlobals/g_iAlarmStorageClear
Application/GVL/LIC_01_CICLE	float	Prueba/Application/GVL/LIC_01_CICLE
Application/GVL/LIC_01_KP	float	Prueba/Application/GVL/LIC_01_KP
Application/GVL/LIC_01_LIMIT_ACT	boolean	Prueba/Application/GVL/LIC_01_LIMIT_ACT
Application/GVL/LIC_01_MAN	boolean	Prueba/Application/GVL/LIC_01_MAN
Application/GVL/LIC_01_MAN_OFFSET	float	Prueba/Application/GVL/LIC_01_MAN_OFFSET
Application/GVL/LIC_01_MAN_OUT	float	Prueba/Application/GVL/LIC_01_MAN_OUT
Application/GVL/LIC_01_OUT	float	Prueba/Application/GVL/LIC_01_OUT
Application/GVL/LIC_01_OVERFLOW	boolean	Prueba/Application/GVL/LIC_01_OVERFLOW
Application/GVL/LIC_01_RESET	boolean	Prueba/Application/GVL/LIC_01_RESET
Application/GVL/LIC_01_SETPOINT	float	Prueba/Application/GVL/LIC_01_SETPOINT
Application/GVL/LIC_01_TN	float	Prueba/Application/GVL/LIC_01_TN
Application/GVL/LIC_01_TV	float	Prueba/Application/GVL/LIC_01_TV
Application/GVL/LT_01_HH	boolean	Prueba/Application/GVL/LT_01_HH
Application/GVL/LT_01_LL	boolean	Prueba/Application/GVL/LT_01_LL
Application/GVL/LT_02_LL	boolean	Prueba/Application/GVL/LT_02_LL
Application/GVL/LT_03_HH	boolean	Prueba/Application/GVL/LT_03_HH
Application/GVL/MODE_MANT	boolean	Prueba/Application/GVL/MODE_MANT
<b>Application/GVL/MT_01_ERROR</b>	<b>boolean</b>	<b>Prueba/Application/GVL/MT_01_ERROR</b>
Application/GVL/MT_01_START	boolean	Prueba/Application/GVL/MT_01_START
Application/GVL/MT_01_STOP	boolean	Prueba/Application/GVL/MT_01_STOP
<b>Application/GVL/MT_02_ERROR</b>	<b>boolean</b>	<b>Prueba/Application/GVL/MT_02_ERROR</b>
Application/GVL/MT_02_START	boolean	Prueba/Application/GVL/MT_02_START
Application/GVL/MT_02_STOP	boolean	Prueba/Application/GVL/MT_02_STOP
Application/GVL/MT_03_ERROR	boolean	Prueba/Application/GVL/MT_03_ERROR
Application/GVL/MT_03_START	boolean	Prueba/Application/GVL/MT_03_START
Application/GVL/MT_03_STOP	boolean	Prueba/Application/GVL/MT_03_STOP
Application/GVL/PE_STOP	boolean	Prueba/Application/GVL/PE_STOP
Application/GVL/PRES_FLU_A	boolean	Prueba/Application/GVL/PRES_FLU_A
Application/GVL/PRES_FLU_B	boolean	Prueba/Application/GVL/PRES_FLU_B
Application/GVL/START_STOP	boolean	Prueba/Application/GVL/START_STOP
Application/GVL/TQ_01_VALOR	float	Prueba/Application/GVL/TQ_01_VALOR
<b>Application/GVL/TQ_03_VALOR</b>	<b>float</b>	<b>Prueba/Application/GVL/TQ_03_VALOR</b>
Application/GVL/TT_01_H	boolean	Prueba/Application/GVL/TT_01_H
Application/GVL/TT_01_VALOR	float	Prueba/Application/GVL/TT_01_VALOR
Application/GVL/MLV_01_CLOSE	boolean	Prueba/Application/GVL/MLV_01_CLOSE
<b>Application/GVL/MLV_01_ERROR</b>	<b>boolean</b>	<b>Prueba/Application/GVL/MLV_01_ERROR</b>
Application/GVL/MLV_01_OPEN	boolean	Prueba/Application/GVL/MLV_01_OPEN
Application/GVL/MLV_01_STOP	boolean	Prueba/Application/GVL/MLV_01_STOP
Application/GVL/MLV_02_CLOSE	boolean	Prueba/Application/GVL/MLV_02_CLOSE
<b>Application/GVL/MLV_02_ERROR</b>	<b>boolean</b>	<b>Prueba/Application/GVL/MLV_02_ERROR</b>
Application/GVL/MLV_02_OPEN	boolean	Prueba/Application/GVL/MLV_02_OPEN
Application/GVL/MLV_02_STOP	boolean	Prueba/Application/GVL/MLV_02_STOP

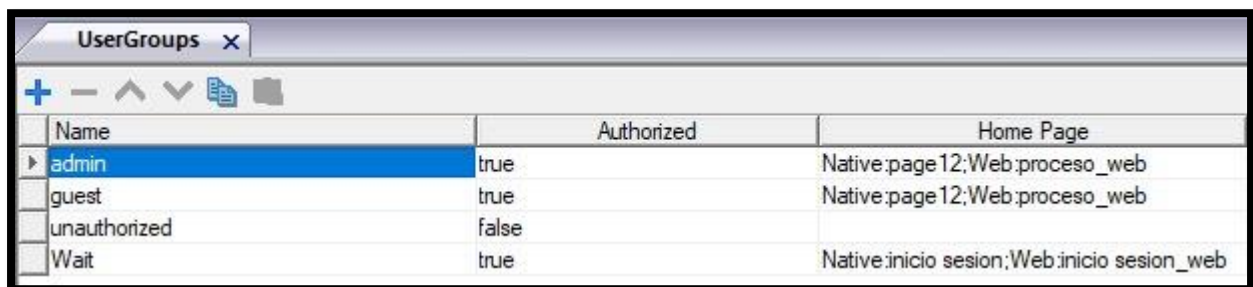
Fuente: Elaboración propia.

Figura 2-52 Variables agregadas a Panel Builder.

## 2.2.4 Creación de usuarios y accesos

Para configurar el acceso y control de usuarios en la HMI, se debe seguir el siguiente proceso:

- Definir grupos de usuarios: Los grupos de usuarios permiten organizar los distintos niveles de acceso. Por ejemplo, se pueden crear grupos como Administrador, Operador, Mantenimiento, etc., con diferentes permisos para acceder o modificar las configuraciones de la HMI. Esto se muestra en la Figura 2-53.
- Crear usuarios: Una vez definidos los grupos de usuarios, se deben crear los usuarios específicos que pertenecerán a esos grupos. A cada usuario se le asignarán credenciales (nombre de usuario y contraseña) y, opcionalmente, una política de acceso (como restricciones de visualización o control sobre ciertas funciones de la HMI). Esto se muestra en la Figura 2-54.
- Asignación de permisos: Después de crear los grupos de usuarios y los usuarios correspondientes, se deben asignar los permisos de acceso a las distintas pantallas y funciones del sistema. Esto garantiza que los usuarios solo puedan interactuar con las partes del sistema que se les haya permitido, manteniendo la seguridad y el control sobre el proceso. Esto se muestra en la Figura 2-54



Name	Authorized	Home Page
admin	true	Native:page 12;Web:proceso_web
guest	true	Native:page 12;Web:proceso_web
unauthorized	false	
Wait	true	Native:inicio sesion;Web:inicio sesion_web

Fuente: Elaboración propia.

Figura 2-53 Grupos de usuarios

Name	Default User	Inactive	Group	Password
admin	<input type="checkbox"/>	<input type="checkbox"/>	admin	*****
Cristopher	<input type="checkbox"/>	<input type="checkbox"/>	admin	*****
Espera	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Wait	*****
Loreto	<input type="checkbox"/>	<input type="checkbox"/>	guest	*****

Fuente: Elaboración propia.

Figura 2-54 Crear usuarios

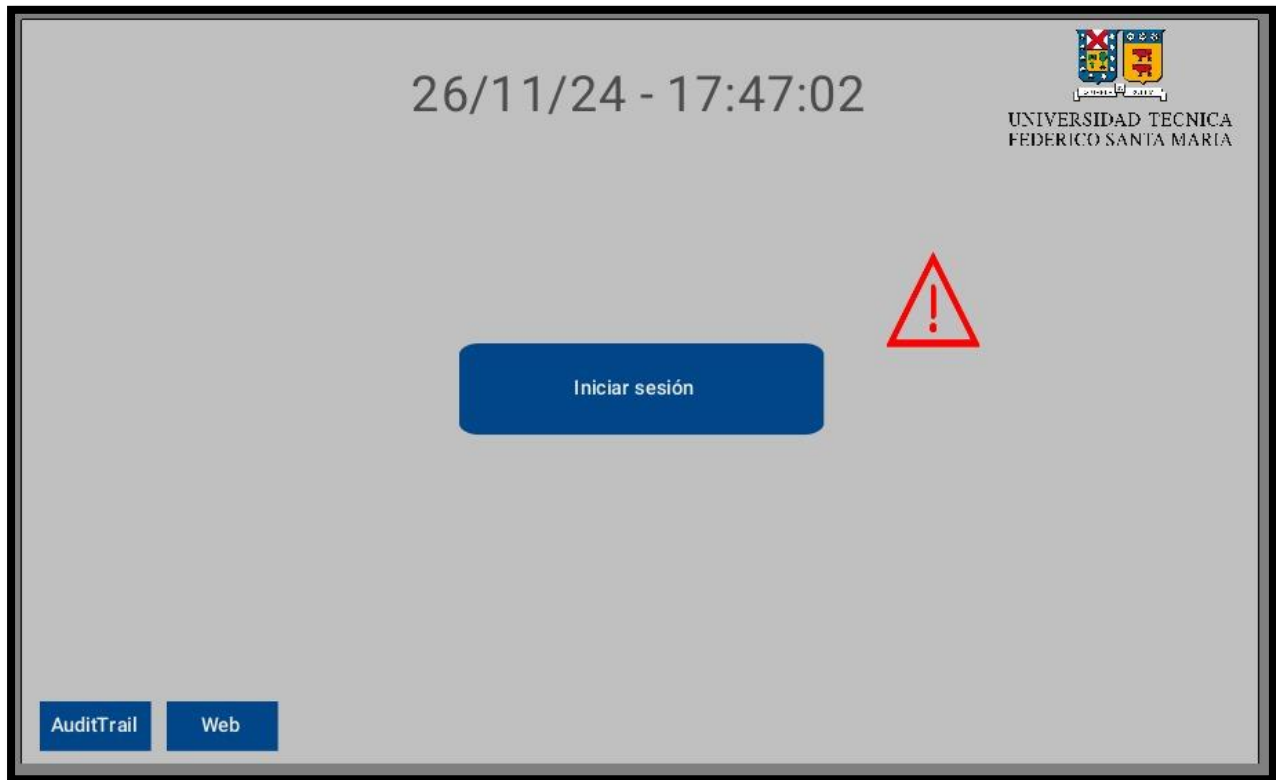
Una vez configurado, el HMI estará protegido por contraseñas y permisos de acceso según los grupos y usuarios definidos, asegurando que solo el personal autorizado tenga acceso a ciertas funcionalidades críticas del proceso.

### 2.2.5 Pantallas de operación

Las pantallas de operación creadas son las que podremos observar a continuación.

#### a) Inicio de sesión

En esta pantalla, el usuario ingresa su nombre de usuario y contraseña previamente configurados. El sistema valida las credenciales y asigna acceso según el grupo de usuario (Administrador, Operador, etc.). Dependiendo del nivel de acceso, el usuario podrá ver y controlar diferentes funcionalidades del HMI. Este proceso garantiza que solo los usuarios autorizados tengan acceso al sistema y sus operaciones, mostrada en la figura 2-55.



Fuente: Elaboración propia.

Figura 2-55 Inicio de sesión

b) Proceso


En esta pantalla se muestra el estado en tiempo real del proceso, incluyendo:

- Indicadores de estado de dispositivos como bombas y válvulas.
- Mediciones de variables como nivel y temperatura.
- Controles para ajustar parámetros o activar dispositivos.
- Alarmas y mensajes sobre fallos o eventos críticos.

Esta interfaz permite supervisar y controlar el proceso de manera eficiente, proporcionando información clave para la operación, mostrado en la figura 2-56.

26/11/2024 - 17:47:14

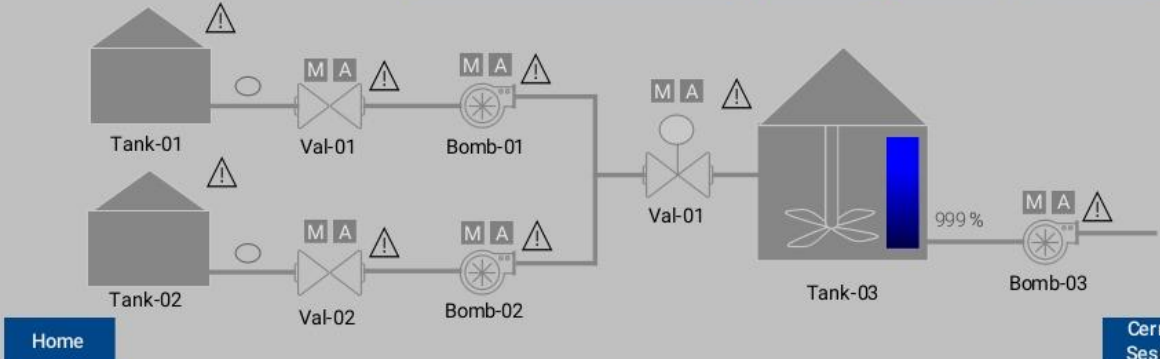
Proceso

  
 UNIVERSIDAD TECNICA  
 FEDERICO SANTA MARIA

**Active Alarms - Table**

Select	Name	State	Value	Time	Description	Priority
<input type="checkbox"/>	Label	Label	Label	11/26/24 - 17:47:14	Label	Label

Hide Not Triggered Toggle selection Ack Reset Save



Home Cerrar Sesión

Fuente: Elaboración propia.

Figura 2-56 Proceso

### c) Historial de alarmas

En esta pantalla, se muestra un registro detallado de todas las alarmas ocurridas durante el proceso, proporcionando información sobre:

- Tipo de alarma: Indica el tipo de evento que generó la alarma (por ejemplo, fallo de equipo, sobrepresión, etc.).
- Hora y fecha: Muestra el momento exacto en que la alarma fue activada.
- Descripción: Detalla la causa o el estado que generó la alarma.
- Estado: Indica si la alarma está activa o ha sido resuelta.

Este historial permite a los operadores y al personal de mantenimiento revisar los eventos pasados, facilitando la identificación de patrones y la toma de decisiones para mejorar la confiabilidad del sistema, mostrado e la figura 2-57.

26/11/2024 - 17:47:58

History Alarms

UNIVERSIDAD TECNICA  
FEDERICO SANTA MARIA

**Alarms History - Table**  
**From:** 26/11/24 - 17:47:58  
**To:** 26/11/24 - 17:47:58

Refresh 5 Mins

Filter: [dropdown] 99999 [X]

Timestamp	Name	State	Value	Description
Label	Label	Label	Label	Label

Home Cerrar Sesión

Fuente: Elaboración propia.

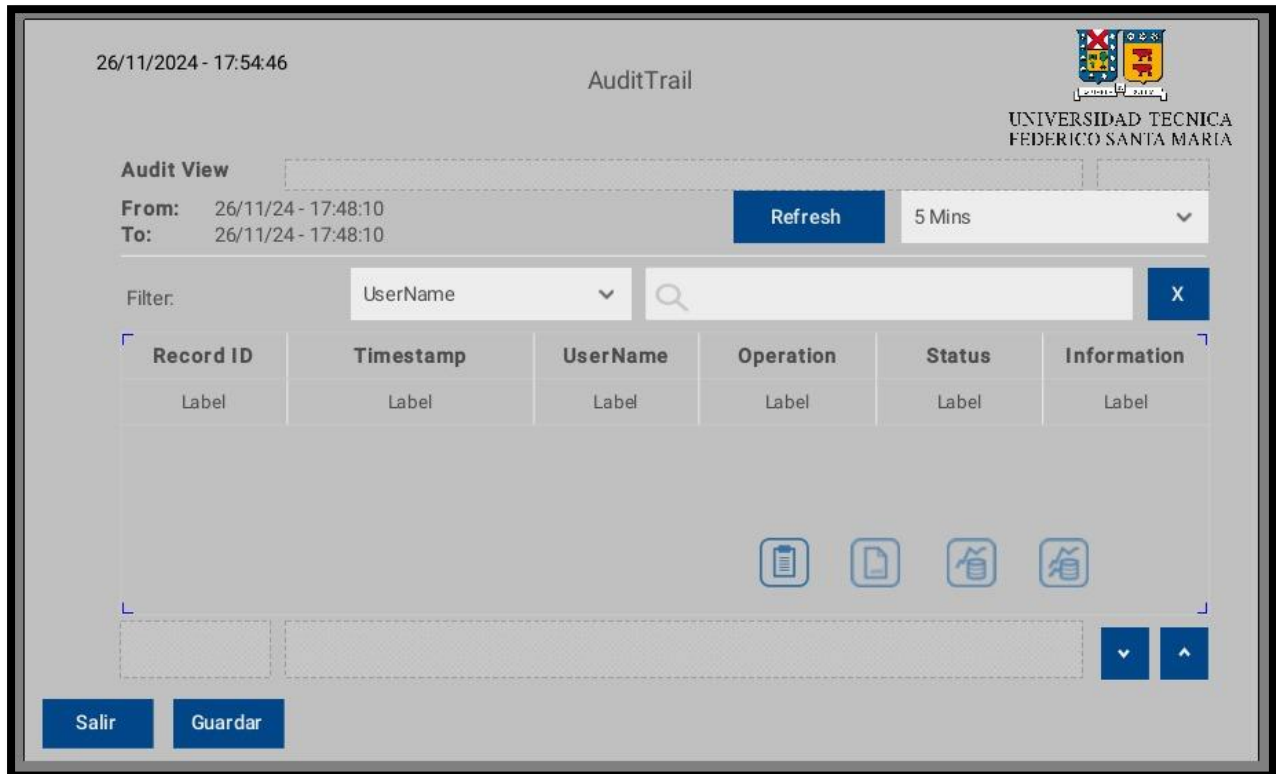
Figura 2-57 Historial de alarmas.

#### d) Audit Trail

En esta pantalla, se registra y muestra un rastro de auditoría (Audit Trail), que es una lista detallada de todas las acciones realizadas por los usuarios en el sistema. Los elementos clave incluyen:

- Usuario: Identifica al usuario que realizó la acción.
- Acción: Describe la operación realizada, como cambios en parámetros, ajustes de dispositivos o acceso a ciertas áreas del sistema.
- Fecha y hora: Registra el momento exacto en que se realizó cada acción.
- Resultado: Muestra si la acción fue exitosa o si ocurrió algún error.

El Audit Trail es esencial para mantener un registro de las actividades del sistema, lo que ayuda a garantizar la trazabilidad, la seguridad y el cumplimiento de normativas. Además, facilita la resolución de problemas y el análisis de la eficiencia operativa, mostrada en la figura 2-58.



Fuente: Elaboración propia.

Figura 2-58 Audit Trail

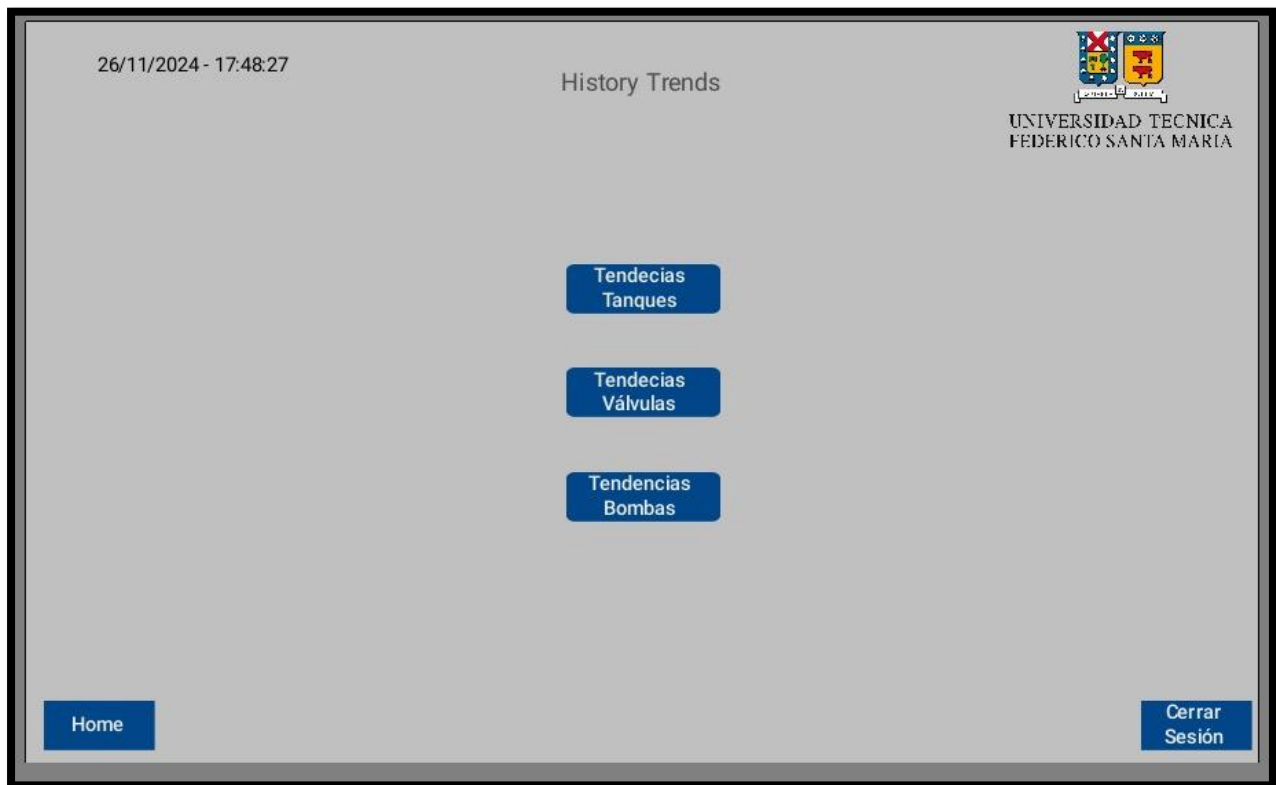
#### e) History Trends

En esta pantalla, se visualizan las tendencias históricas de las variables clave del proceso a lo largo del tiempo. Los elementos principales incluyen:

- Gráficos de tendencias: Muestran cómo han cambiado las variables (como temperatura, nivel, presión, etc.) a lo largo del tiempo, permitiendo una visión clara de los patrones operativos.

- Intervalo de tiempo: Se puede ajustar el rango temporal para visualizar las tendencias de días, semanas, meses o períodos específicos.

Esta pantalla es útil para analizar el comportamiento histórico del proceso, identificar tendencias y tomar decisiones basadas en datos históricos para mejorar la operación y prevenir posibles fallos u optimizaciones, mostrado e la figura 2-59, figura 2-60, figura 2-61 y figura 2-62.



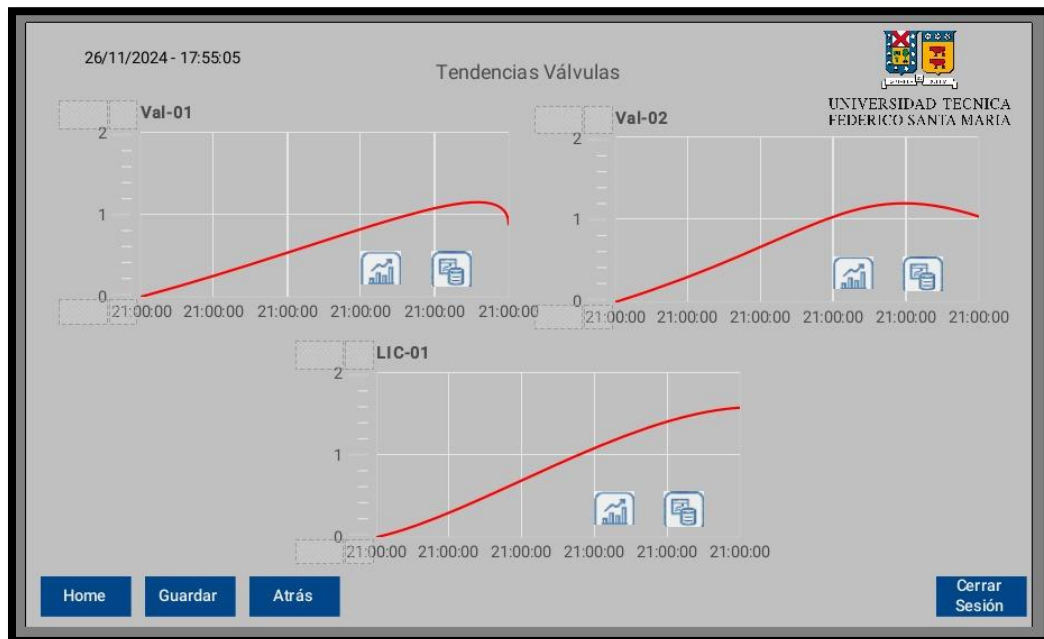
Fuente: Elaboración propia.

Figura 2-59 History Trends



Fuente: Elaboración propia.

Figura 2-60 Tendencia Tanque.



Fuente: Elaboración propia.

Figura 2-61 Tendencias Válvulas.



Fuente: Elaboración propia.

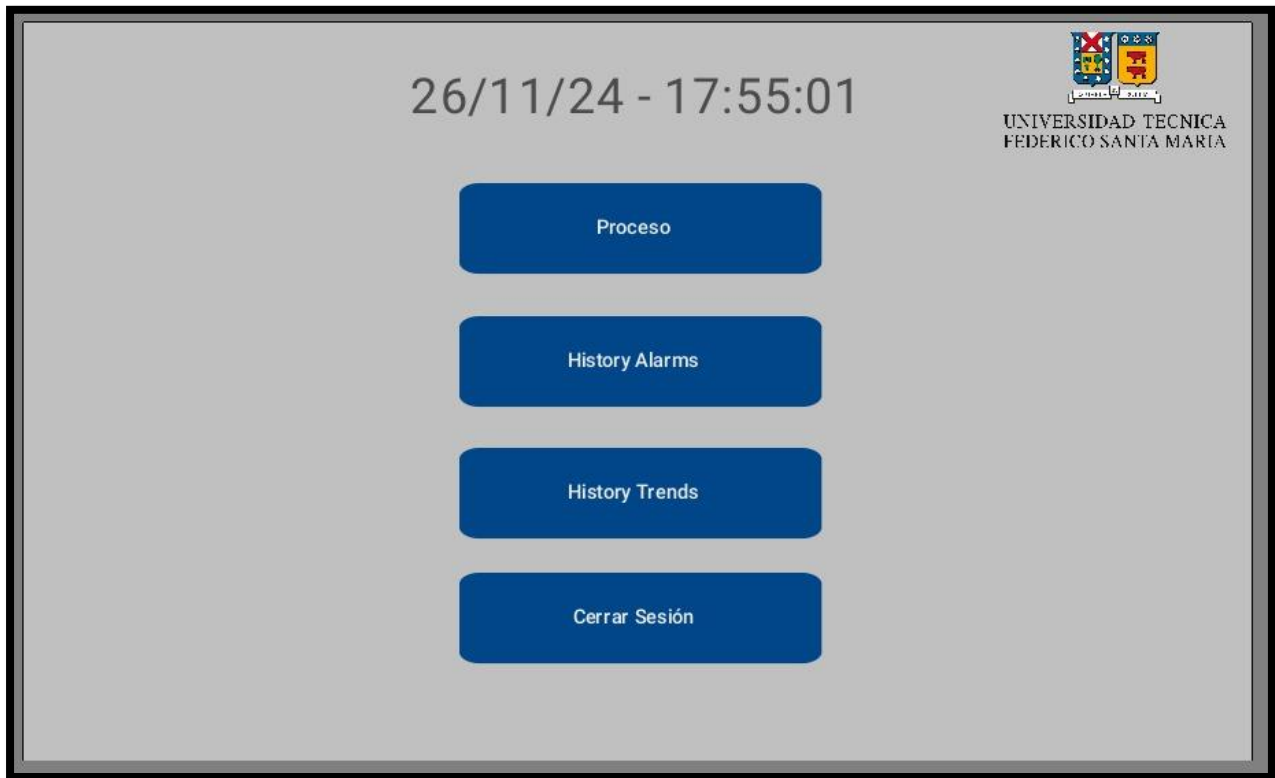
Figura 2-62 Tendencias bombas.

#### f) Menú

La pantalla del Menú proporciona una interfaz central para navegar entre las diferentes funciones y pantallas del sistema HMI. Los elementos clave incluyen:

- Opciones de navegación: Enlaces o botones para acceder rápidamente a las principales funciones del sistema, como el Inicio de sesión, Monitoreo de procesos, Historial de alarmas, Audit Trail, Tendencias históricas, entre otras.
- Salir: Opción para cerrar sesión o salir de la aplicación, asegurando que el sistema quede protegido cuando no esté en uso.

El Menú facilita la navegación del usuario por las distintas pantallas y funcionalidades del sistema HMI, garantizando un acceso rápido y organizado a la información y herramientas necesarias para operar y supervisar el proceso, mostrado en la figura 2-63.



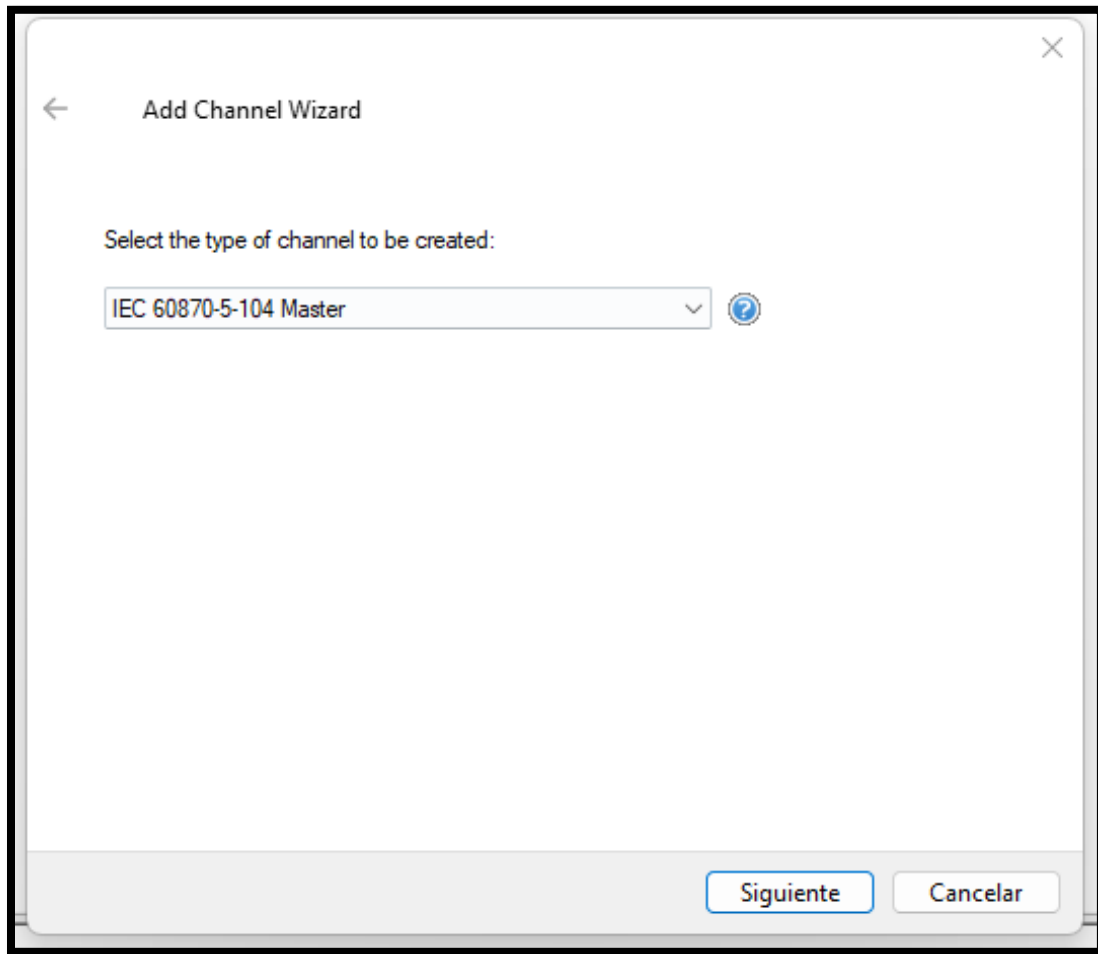
Fuente: Elaboración propia.

Figura 2-63 Menú.

### 2.2.6 KepServer

A continuación, se mostrará la configuración necesaria en el software KepServer para la recepción de señales transmitida por el PLC AC500, las cuales se emiten en el protocolo IEC104.

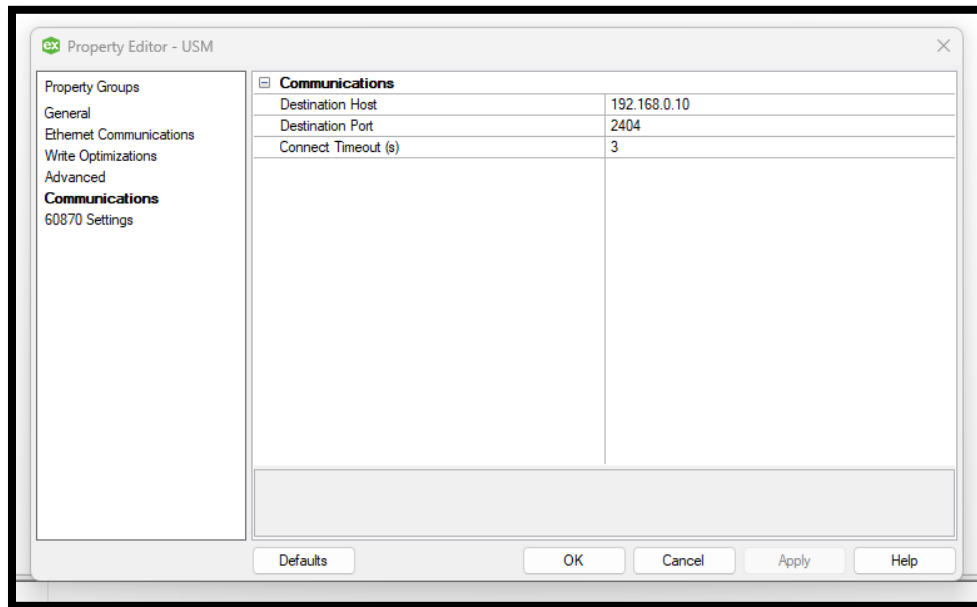
- Se crea un nuevo canal de comunicaciones con el protocolo de comunicación a utilizar, Figura 2-64.



Fuente: Elaboración propia.

Figura 2-64 Protocolo de comunicación.

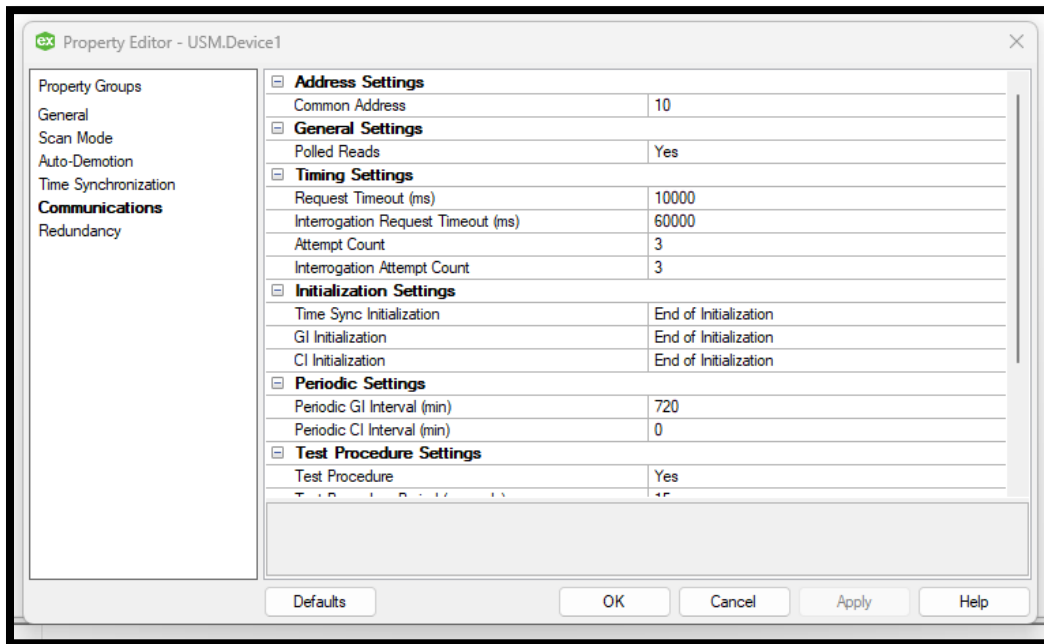
- Se configura la dirección IP del equipo el cual se va a comunicar, Figura 2-65.



Fuente: Elaboración propia.

Figura 2-65 Direccionamiento comunicación

- Se configura la dirección común de las señales las cuales se comunicarán, Figura 2-66.



Fuente: Elaboración propia.

Figura 2-66 Dirección común.

- Se integran las señales que se desean leer, Figura 2-67.

Tag Name	Address	Data Type
TT_01_VALOR	M_ME_FV.38.CURRENTVALUE	Float
TQ_03_VALOR	M_ME_FV.37.CURRENTVALUE	Float
LIC_01_TV	M_ME_FV.36.CURRENTVALUE	Float
LIC_01_TN	M_ME_FV.35.CURRENTVALUE	Float
LIC_01_SETPOINT	M_ME_FV.34.CURRENTVALUE	Float
LIC_01_OUT	M_ME_FV.33.CURRENTVALUE	Float
LIC_01_MAN_OUT	M_ME_FV.32.CURRENTVALUE	Float
LIC_01_MAN_OFFSET	M_ME_FV.31.CURRENTVALUE	Float
LIC_01_KP	M_ME_FV.30.CURRENTVALUE	Float
VLV_02_STOP	M_SP.26.CURRENTVALUE	Boolean
VLV_02_OPEN	M_SP.25.CURRENTVALUE	Boolean
VLV_02_ERROR	M_SP.24.CURRENTVALUE	Boolean
VLV_02_CLOSE	M_SP.23.CURRENTVALUE	Boolean
VLV_01_STOP	M_SP.22.CURRENTVALUE	Boolean
VLV_01_OPEN	M_SP.21.CURRENTVALUE	Boolean
VLV_01_ERROR	M_SP.20.CURRENTVALUE	Boolean
VLV_01_CLOSE	M_SP.19.CURRENTVALUE	Boolean
TT_01_H	M_SP.18.CURRENTVALUE	Boolean
START_STOP	M_SP.17.CURRENTVALUE	Boolean
PRESS_FLU_B	M_SP.16.CURRENTVALUE	Boolean
PRESS_FLU_A	M_SP.15.CURRENTVALUE	Boolean
PE_STOP	M_SP.14.CURRENTVALUE	Boolean
MT_03_STOP	M_SP.13.CURRENTVALUE	Boolean
MT_03_START	M_SP.12.CURRENTVALUE	Boolean
MT_03_ERROR	M_SP.11.CURRENTVALUE	Boolean
MT_02_STOP	M_SP.10.CURRENTVALUE	Boolean
MT_02_START	M_SP.9.CURRENTVALUE	Boolean
MT_02_ERROR	M_SP.8.CURRENTVALUE	Boolean
MT_01_STOP	M_SP.7.CURRENTVALUE	Boolean
MT_01_START	M_SP.6.CURRENTVALUE	Boolean
MT_01_ERROR	M_SP.5.CURRENTVALUE	Boolean
MODE_MANT	M_SP.4.CURRENTVALUE	Boolean
LT_03_HH	M_SP.3.CURRENTVALUE	Boolean
LT_02_LL	M_SP.2.CURRENTVALUE	Boolean
LT_01_LL	M_SP.1.CURRENTVALUE	Boolean

Fuente: Elaboración propia.

Figura 2-67 Tabla de variables.

### **CAPÍTULO 3: ANÁLISIS DE RESULTADOS**

### 3 ANÁLISIS DE RESULTADOS

La evaluación del sistema automatizado se centra en dos aspectos principales: la puesta en marcha y la respuesta del sistema bajo diversas condiciones. Este capítulo detalla los pasos realizados para garantizar la integración correcta de los componentes, así como el comportamiento del sistema en situaciones normales y críticas. Los resultados obtenidos permiten validar el diseño y la implementación del sistema.

#### 3.1 PUESTA EN MARCHA DE LA PLANTA

La puesta en marcha del sistema automatizado constituye una etapa crucial para validar la correcta integración y operación de todos los componentes involucrados, mostrados en la Figura 3-1.



Fuente: Elaboración propia

Figura 3-1 Componentes del proyecto.

### 3.1.1 Verificación del Hardware y Software

Se verificó el Hardware y software del sistema y se comprobó que todos los elementos del proyecto funcionaran de manera óptima, para ello se realizaron los siguientes puntos:

- Inspección y configuración de los módulos de entradas y salidas (DA501 y CI502) conectados al PLC AC500. Figura 3-2.
- Confirmación de la comunicación entre el PLC y la HMI mediante los protocolos Profinet y TCP/IP.
- Revisión de la conexión con KEPServer para garantizar la integración con los sistemas de supervisión.
- Configuración de los sensores y actuadores, asegurando que las entradas analógicas y digitales se correspondieran correctamente con las variables asignadas en el software.



Fuente: Elaboración propia

Figura 3-2 PLC AC500.

### 3.1.2 Pruebas de Comunicación

Para la comunicación entre los dispositivos, se simula un switch industrial utilizando un router, mostrado en la Figura 3-3, el cual proporciona una señal Wi-Fi que permite acceder al web server de los equipos.



Fuente: Elaboración propia.

Figura 3-3 Router Wi-Fi

- Se realizaron pruebas de intercambio de datos entre el PLC y la HMI, validando la sincronización de variables y la respuesta de las alarmas.
- Se evaluó la estabilidad de las conexiones Profinet con la IO remota y la transferencia de datos IEC104 hacia KEPServer.

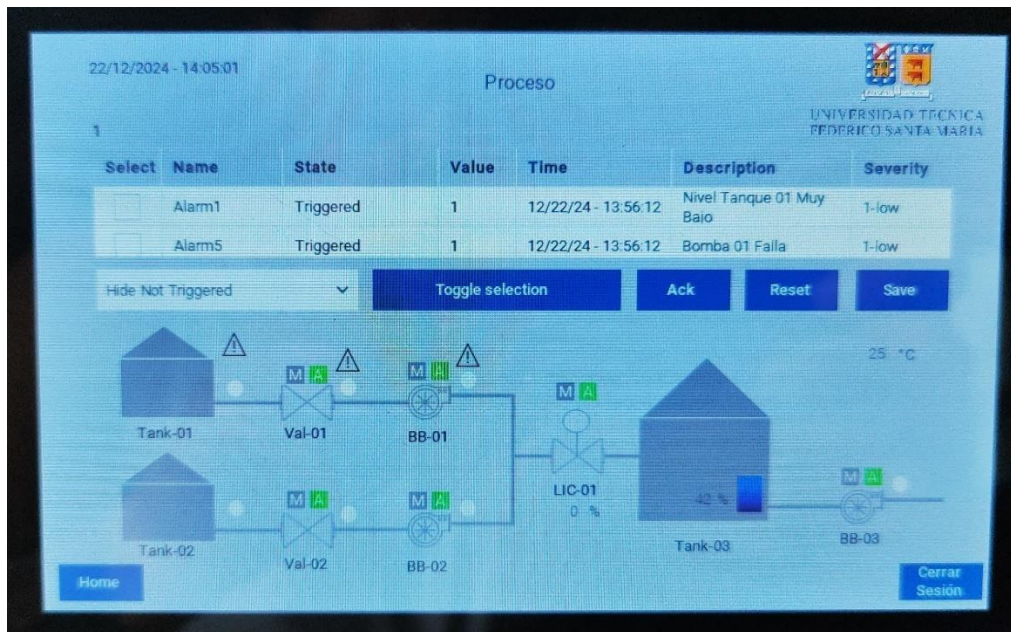
### 3.1.3 Ajustes de Configuración

En esta sección se detallan los ajustes realizados para optimizar el sistema de control para garantizar una operación precisa y eficiente.

- Se ajustaron las configuraciones de los controladores PID para obtener una respuesta óptima en el control del nivel del tanque.
- Se revisaron y corrigieron las lógicas de control en los programas creados en Automation Builder, como el control de bombas y válvulas.
- Refinamiento de los límites de las alarmas para garantizar una respuesta precisa y oportuna ante posibles fallos.

### 3.1.4 Operación Inicial

- Se puso en marcha el sistema con las condiciones iniciales establecidas para el proceso simulado, validando la correcta secuencia de operaciones y la visualización en tiempo real desde la HMI, mostrado en la figura 3-4.
- Observación y registro de datos operativos preliminares para identificar posibles ajustes en las configuraciones del sistema.



Fuente: Elaboración propia

Figura 3-4 Operación inicial

## 3.2 RESPUESTA DEL SISTEMA

La respuesta del sistema automatizado fue evaluada bajo distintos escenarios operativos con el objetivo de analizar su desempeño en condiciones normales y críticas.

### 3.2.1 Respuesta en condiciones normales

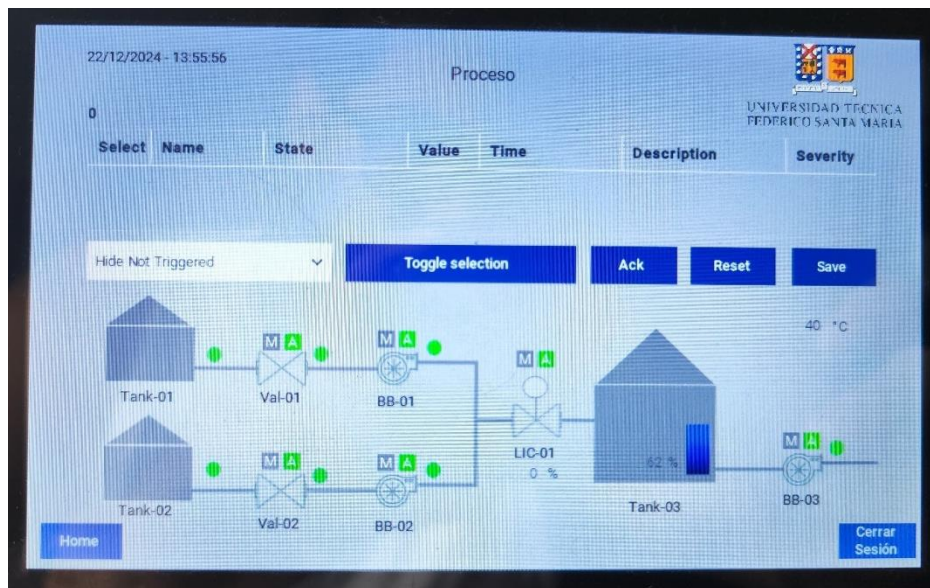
Durante la operación en estado estacionario, se observó lo siguiente:

#### **Precisión del Control:**

- El controlador PID mantuvo el nivel del tanque dentro del rango establecido (0-100%)
- Las bombas y válvulas operaron de forma sincronizada, asegurando una mezcla uniforme de los fluidos.

### Visualización HMI:

- Las pantallas mostraron correctamente el estado de los dispositivos, las variables clave (nivel y temperatura), figura 3-5, y las tendencias históricas de las mediciones.
- La interfaz respondió de manera fluida a las interacciones del usuario, permitiendo ajustes inmediatos en los parámetros del sistema.
- Las tendencias gráficas proporcionaron información útil para anticipar ajustes operativos.



Fuente: Elaboración propia

Figura 3-5 Proceso en marcha.

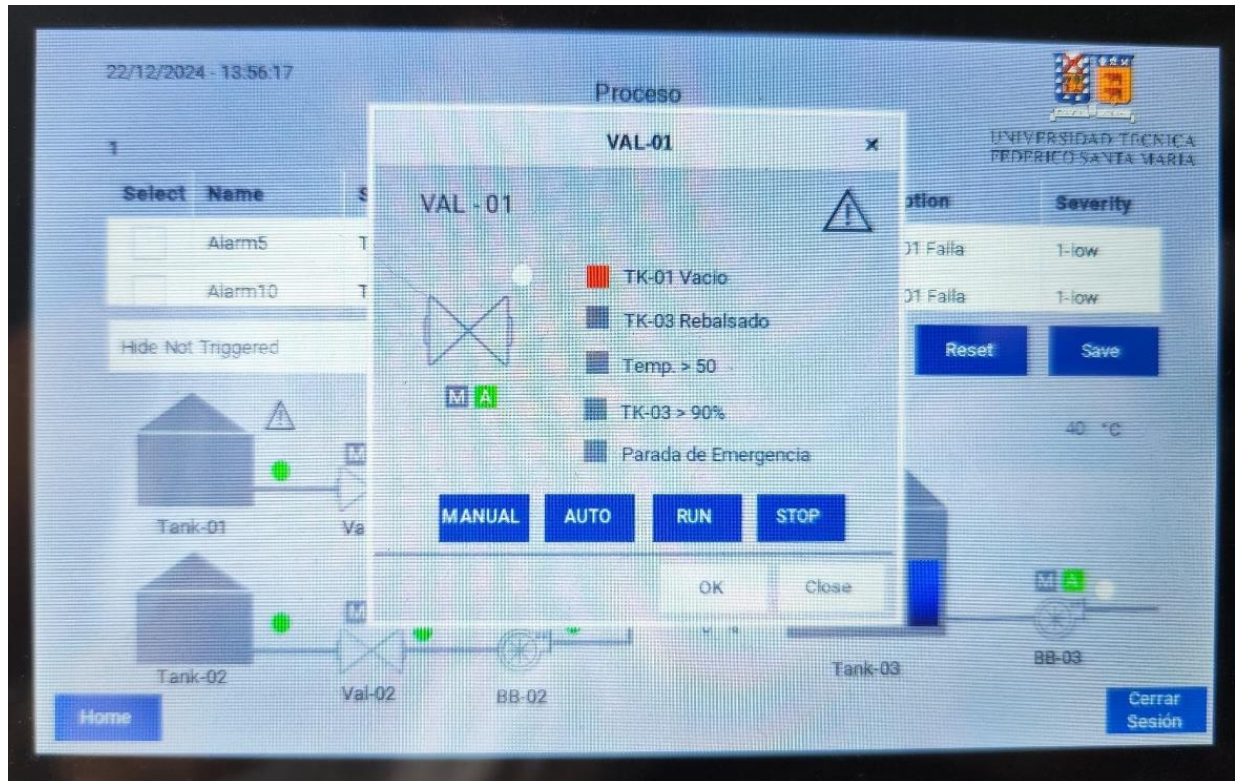
### 3.2.2 Respuesta en Condiciones Críticas

En escenarios simulados de falla, el sistema demostró una respuesta adecuada:

#### Alarmas:

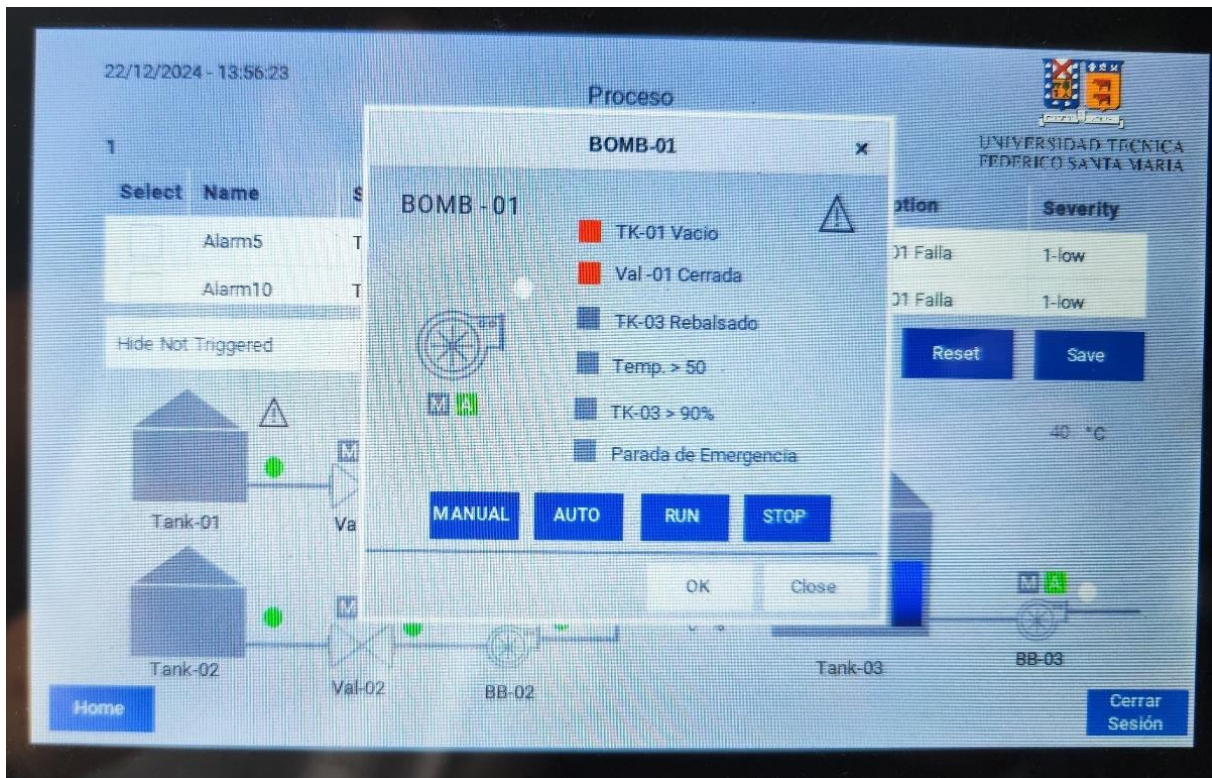
- Las alarmas se activaron inmediatamente ante eventos como sobrellenado del tanque, sobrecalentamiento o fallas en las bombas.

- La HMI registró y visualizó las alarmas en tiempo real, facilitando al operador la toma de decisiones.
- El historial de alarmas mostró información detallada sobre el tipo, fecha y hora de los eventos, lo que permitió un análisis efectivo para prevenir futuras ocurrencias.



Fuente: Elaboración propia

Figura 3-6 alarmas en válvulas.



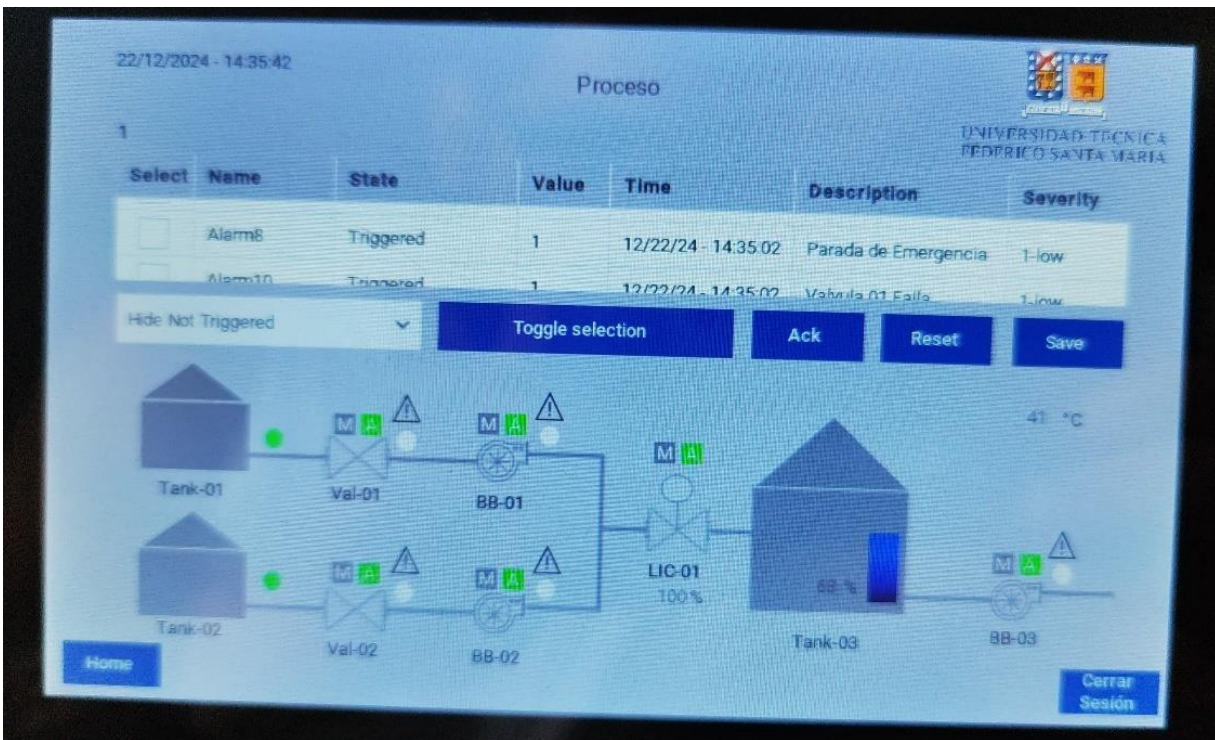
Fuente: Elaboración propia

Figura 3-7 Alarmas en bombas.

### Parada de Emergencia:

Al activar la parada de emergencia, el sistema detuvo todas las operaciones, asegurando la seguridad del proceso y protegiendo los equipos.

Se verificó que la parada no afectara la integridad de los datos registrados ni las configuraciones preexistentes.



Fuente: Elaboración propia

Figura 3-8 Parada de emergencia.

### Reinicio del Sistema:

Tras resolver las condiciones de falla, el sistema fue capaz de reiniciar las operaciones sin inconvenientes, gracias a la integración eficiente de los protocolos de comunicación y la configuración de las variables globales.

Se realizaron pruebas de reinicio en diferentes etapas del proceso para confirmar su recuperación completa.

### Resiliencia:

- El sistema demostró resiliencia al continuar operando con normalidad tras la resolución de condiciones críticas, sin pérdida de datos ni impactos negativos en la operación general.

### 3.2.3 Evaluación de Desempeño

**Confiabilidad:**

La detección de eventos y su resolución se llevaron a cabo en tiempos aceptables, minimizando interrupciones operativas.

**Flexibilidad:**

Los operadores pudieron realizar ajustes en tiempo real a través de la HMI, adaptando el sistema a nuevas condiciones sin afectar la estabilidad.

**Impacto Operativo:**

- La automatización del proceso permitió reducir los tiempos de operación manual, incrementando la eficiencia global del sistema.
- La integración de las herramientas de supervisión y control facilitó la toma de decisiones.

En general, los resultados obtenidos reflejan que el sistema automatizado cumple con los objetivos planteados, demostrando una operación estable y una respuesta eficiente ante condiciones normales y críticas. Este desempeño valida la solución propuesta como una herramienta viable para la optimización de procesos industriales.

## **CONCLUSIONES**

A continuación, se detallan las conclusiones alcanzadas durante la realización del proyecto:

El proyecto demostró que la combinación del PLC ABB AC500, la HMI CP600 y los softwares asociados ofrece una solución robusta y confiable para la automatización de procesos industriales. La integración de hardware y software facilitó una operación fluida y precisa.

El sistema respondió adecuadamente en escenarios normales y críticos, activando alarmas y procedimientos de seguridad de manera inmediata. La funcionalidad de parada de emergencia y la capacidad de reinicio sin pérdida de datos destacan su robustez.

La automatización del proceso permitió reducir la intervención manual, incrementando la eficiencia y minimizando riesgos de error humano. La implementación de controladores PID mejoró la estabilidad del sistema y el control de las variables críticas.

Las pantallas desarrolladas en el software Panel Builder 600 ofrecieron un entorno intuitivo y accesible para los operadores, garantizando un monitoreo eficiente y una toma de decisiones ágil.

La tecnología ABB utilizada en este proyecto se mostró como una opción competitiva frente a marcas líderes del mercado, destacándose por su flexibilidad, capacidad de personalización y facilidad de uso.

La arquitectura del sistema, basada en protocolos como Profinet e IEC104, permite la ampliación del alcance del proyecto para integrar más dispositivos o procesos, asegurando compatibilidad y rendimiento a largo plazo.

En conjunto, el proyecto no solo validó las capacidades de la tecnología ABB, sino que también sentó las bases para futuras aplicaciones en la optimización de procesos industriales automatizados.

## **BIBLIOGRAFÍA**

ABB. Página oficial de la empresa ABB que detalla los productos que ofrece la marca. [En línea]. <https://800xahardwareselector.automation.abb.com/> ; [Consulta: 19 de septiembre de 2024].

ABB Library. Pagina que proporciona información sobre PLC, Software, manuales, etc. [En línea]. <https://library.abb.com/> ; [Consulta: 20 septiembre de 2024].