

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

SEDE CONCEPCIÓN · REY BALDUINO DE BÉLGICA

[DEPARTAMENTO DE ELECTRONICA E INFORMATICA]

CONCEPCIÓN · CHILE



**SIMULACIÓN Y CONTROL AUTOMATIZADO DE UN SISTEMA
DE ALMACENAMIENTO CON TRANSELEVADORES Y
REGULACIÓN DE TEMPERATURA MEDIANTE PLC EN UN
ENTORNO DE SIMULACIÓN INDUSTRIAL**

Jordan Mauricio Gallardo Domínguez

Memoria de titulación para optar al título de
Ingeniero de Ejecución en Control e Instrumentación Industrial

Profesor guía: Ricardo Vera González

Año 2026



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: Simulación y control automatizado de un sistema de almacenamiento con transelevadores y regulación de temperatura mediante PLC en un entorno de simulación industrial

Nombre del candidato(a): Jordan Mauricio Gallardo Dominguez

Carrera / Grado: Ingeniería de Ejecución en Control e Instrumentación Industrial

Campus: Concepción Departamento: Electronica e Informatica

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Ricardo Luis González, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente DEJO CONSTANCIA que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO** contiene información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por (marcar una opción):

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 14/09/26

Firma: [Firma]

Estudiante o Candidato(a):

Fecha: 14/09/26

Firma: [Firma]

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

RESUMEN

Se presenta la integración de un sistema virtual de almacenamiento con transelevador y regulación de temperatura, desarrollado mediante TIA Portal, S7-PLCSIM Advanced, Factory I/O y una aplicación local en Python con Dash, Plotly y Snap7. El propósito es documentar la arquitectura, el intercambio de datos y el comportamiento del prototipo, estableciendo criterios para su evaluación sin disponer de una planta física.

Se describe una interfaz de 54 posiciones y un modelo térmico que incorpora enfriamiento, retorno ambiental y ruido. Se identifican los campos de comunicación del almacén y de temperatura y se analiza matemáticamente el equilibrio de la planta simulada bajo salida constante.

La evidencia visual muestra movimiento del almacén, visualización de variables y cambios de consigna. La revisión del código determina que la ocupación mostrada se acumula a partir de destinos observados y que la evolución térmica se activa desde la actualización del dashboard.

Se concluye que el prototipo constituye una base de integración virtual documentable, cuya validación cuantitativa requiere completar la configuración del PLC y ejecutar ensayos trazables. El alcance no incluye implementación física, certificación industrial ni validación sanitaria.

Palabras clave: almacenamiento automatizado; transelevador; PLC simulado; control de temperatura; supervisión local.

ABSTRACT

The integration of a virtual storage system with a stacker crane and temperature regulation is presented, developed using TIA Portal, S7-PLCSIM Advanced, Factory I/O, and a local Python application with Dash, Plotly, and Snap7. The purpose is to document the architecture, data exchange, and behavior of the prototype, establishing criteria for its evaluation without requiring a physical plant.

A 54-position interface is described, along with a thermal model that incorporates cooling, ambient return, and noise. The communication fields for the warehouse and temperature are identified, and the equilibrium of the simulated plant under constant output is mathematically analyzed.

Visual evidence shows warehouse movement, variable visualization, and setpoint changes. Code review determines that the occupation displayed accumulates based on observed destinations and that the thermal evolution is activated by the dashboard update.

It is concluded that the prototype constitutes a documentable virtual integration basis, whose quantitative validation requires completing the PLC configuration and executing traceable tests. The scope does not include physical implementation, industrial certification, or sanitary validation.

Keywords: automated storage; stacker crane; simulated PLC; temperature control; local supervision.

GLOSARIO Y SIMBOLOGÍA

Siglas y términos utilizados.

Término	Descripción
AS/RS	Sistema automatizado de almacenamiento y recuperación.
PLC	Controlador lógico programable.
HMI	Interfaz persona-máquina.
DB	Bloque de datos del PLC.
FB / OB	Bloque de función / bloque organizativo.
SP / PV	Set point / variable de proceso.
PID	Control proporcional, integral y derivativo.
OutputCool / OUT	Salida de enfriamiento; el código la interpreta como porcentaje.
BOOL / INT / REAL	Dato lógico / entero / real de punto flotante.
Offset	Desplazamiento de un dato dentro de una zona de memoria.
Callback	Función ejecutada ante un cambio de entrada de la interfaz.
Dashboard	Panel de supervisión; en este proyecto, servicio web local.
Snap7	Biblioteca utilizada para comunicación S7 desde Python.
MAE / RMSE	Error absoluto medio / raíz del error cuadrático medio.
T, u, Δt , η	Temperatura, salida normalizada, paso y ruido del modelo.

ÍNDICE DE CONTENIDO

RESUMEN	4
ABSTRACT	5
GLOSARIO Y SIMBOLOGÍA	6
1. INTRODUCCIÓN Y PLANTEAMIENTO DEL PROBLEMA	9
1.1 Contexto del proyecto	9
1.2 Problema de ingeniería	10
1.3 Justificación técnica y académica	11
1.4 Objetivo general	12
1.5 Objetivos específicos	12
1.6 Alcance y delimitaciones	13
1.7 Enfoque metodológico	13
2. FUNDAMENTOS TEÓRICOS Y ANTECEDENTES	14
2.1 Sistemas automatizados de almacenamiento	14
2.2 Controlador lógico programable y organización del programa	14
2.3 Simulación y puesta en marcha virtual	15
2.4 Factory I/O y PLCSIM Advanced	15
2.5 Memoria y comunicación S7 desde Python	16
2.6 Regulación de temperatura y variables del lazo	16
2.7 Modelo térmico simplificado	17
2.8 Supervisión mediante Dash y Plotly	18
3. DISEÑO DEL SISTEMA	19
3.1 Requisitos funcionales	19
3.2 Requisitos de información y reproducibilidad	19
3.3 Arquitectura y responsabilidades	20
3.4 Contrato de datos del almacén	21
3.5 Contrato de datos térmicos	22
3.6 Señales del proceso mecánico	23
3.7 Secuencia funcional de referencia	24
3.8 Representación de las posiciones	25
3.9 Diseño de supervisión térmica	26
4. IMPLEMENTACIÓN Y SIMULACIÓN	28
4.1 Preparación del entorno Siemens	28
4.2 Configuración de la escena y señales	29
4.3 Organización observable del control del almacén	30
4.4 Inicialización del cliente Python	33
4.5 Lectura y acumulación de destinos	34
4.6 Modelo de enfriamiento implementado	34
4.7 Equilibrio y rango alcanzable del modelo	35
4.8 Ejecución del lazo térmico	36

4.9 Integración con PID_Temp	37
4.10 Escritura de variables y concurrencia	39
4.11 Construcción y operación del dashboard	39
4.12 Incidencias documentadas y tratamiento	40
5. EVIDENCIAS, PRUEBAS Y DISCUSIÓN DE RESULTADOS	42
5.1 Criterio de evaluación	42
5.2 Registro de observaciones visuales	42
5.3 Resultados de inspección estática	43
5.4 Resultados analíticos del modelo térmico	44
5.5 Diseño de ensayos reproducibles	45
5.6 Protocolo de comunicación	46
5.7 Protocolo de almacenamiento	47
5.8 Protocolo de respuesta térmica	48
5.9 Indicadores de desempeño	48
5.10 Evaluación de objetivos	50
5.11 Discusión técnica	50
5.12 Proyección industrial y evaluación económica preliminar	51
CONCLUSIONES	52
REFERENCIAS	53
ANEXO A. MÓDULO DE COMUNICACIÓN Y MODELO TÉRMICO	55
ANEXO B. EXTRACTOS FUNCIONALES DEL DASHBOARD	58
ANEXO C. REDES Y CONFIGURACIÓN DEL PROYECTO	61
C.1 Inicio y reinicio de la secuencia	62
C.2 Recepción y transferencia de la carga	63
C.3 Temporización de elevación y avance	64
C.4 Conteo y asignación de posiciones	65
C.5 Depósito y orden de retorno	66
C.6 Modo y conexiones funcionales del PID	67
C.7 Escalado de salidas y reacción ante error	68
C.8 Límites y entrada de la variable de proceso	69
C.9 Permiso de comunicación con la CPU	70
C.10 Habilitación de simulación durante compilación	71

1. INTRODUCCIÓN Y PLANTEAMIENTO DEL PROBLEMA

1.1 Contexto del proyecto

El almacenamiento industrial exige coordinar el desplazamiento de unidades de carga, la disponibilidad de ubicaciones y la información que recibe el operador. En un almacén automatizado estas funciones se distribuyen entre dispositivos de campo, un controlador y sistemas de supervisión. El movimiento visible de una carga representa solamente una parte del proceso: también se requiere conocer el destino solicitado, reconocer la finalización de la maniobra y mantener consistencia entre la representación informática y la situación del almacén.

Cuando se incorpora una condición de conservación térmica, la integración adquiere una dimensión adicional. La secuencia de transporte es esencialmente discreta, mientras que la temperatura evoluciona como una variable continua. Ambas pueden supervisarse desde una misma interfaz, aunque sus modelos, tiempos de actualización y criterios de aceptación sean diferentes. Esta coexistencia permite estudiar un problema de automatización que combina lógica secuencial, intercambio de datos y regulación de una variable de proceso.

El presente trabajo se sitúa en ese ámbito y desarrolla documentalmente un prototipo de almacenamiento automatizado con transelevador, construido mediante TIA Portal V19, S7-PLCSIM Advanced, Factory I/O y una aplicación local en Python. La escena utilizada es «Automated Warehouse». El componente de supervisión representa 54 posiciones y permite visualizar la temperatura, su consigna y una tendencia histórica. La planta térmica está implementada matemáticamente en Python; no corresponde a una cámara frigorífica física ni a un sensor real.

1.2 Problema de ingeniería

La dificultad abordada consiste en construir una representación funcional en la que el controlador virtual pueda intercambiar señales con una planta tridimensional y, simultáneamente, compartir variables con una aplicación externa. Una configuración incorrecta puede interrumpir la comunicación; una interpretación equivocada de las direcciones puede producir valores inconsistentes; una interfaz puede representar como confirmado un estado que solo corresponde a una solicitud.

El problema no se reduce, por tanto, a conectar programas. Es necesario definir qué significa cada variable, quién la escribe, cuándo se considera válida y qué ocurre cuando una lectura falla. También debe establecerse cómo se relaciona la dinámica térmica calculada por Python con la salida del controlador PID. Sin esa definición, una demostración visual puede parecer satisfactoria y aun así carecer de la trazabilidad necesaria para evaluar su comportamiento.

La guía del proyecto registra dificultades relacionadas con el nombre de la instancia de simulación, los bloques de datos optimizados, los permisos de acceso y errores de dirección. El análisis del código agrega aspectos verificables: la ocupación se acumula a partir de la posición objetivo, las escrituras térmicas abarcan un bloque completo y el modelo de temperatura se ejecuta desde el callback del navegador. Estos elementos convierten la integración y su validación en el núcleo del problema de ingeniería.

La pregunta orientadora queda formulada del siguiente modo: ¿cómo integrar y documentar un sistema virtual de almacenamiento con transelevador y regulación térmica mediante PLC, de manera que sus señales, estados y resultados puedan interpretarse y verificarse de forma coherente?

1.3 Justificación técnica y académica

La simulación permite examinar la lógica antes de disponer de equipos físicos. Este proyecto ofrece un escenario para observar desplazamientos, seguir variables del PLC y relacionar una salida de control con un modelo térmico. Su utilidad concreta reside en hacer visibles las interacciones entre subsistemas y facilitar la repetición de pruebas bajo condiciones definidas.

Desde la formación en control e instrumentación, el proyecto articula competencias que habitualmente se estudian por separado: programación de secuencias, configuración de comunicaciones, interpretación de memoria, desarrollo de interfaces y análisis de sistemas dinámicos. La integración permite discutir por qué una variable de tipo REAL ocupa cuatro bytes, por qué una orden no equivale a una confirmación.

La justificación no supone una reducción económica cuantificada. No se cuenta con una línea base de tiempos de puesta en marcha, costos de equipos o pérdidas operacionales. En consecuencia, los beneficios se presentan como posibilidades del enfoque y como capacidades del banco virtual, sin asignar porcentajes de ahorro ni mejoras de productividad no medidas.

El aporte documental consiste además en transformar materiales dispersos en una descripción reproducible. El programa Python permite reconstruir el contrato de datos y el modelo térmico con mayor precisión que una captura aislada.

1.4 Objetivo general

Integrar un sistema automatizado de almacenamiento con transelevador y regulación de temperatura en un entorno de simulación industrial, mediante un PLC Siemens simulado, Factory I/O y una aplicación de supervisión local en Python, y evaluar su funcionamiento a partir del intercambio de señales, la representación de posiciones y la respuesta térmica.

1.5 Objetivos específicos

OE1. Definir la arquitectura del sistema, las herramientas utilizadas y las responsabilidades de sus componentes de control, simulación, comunicación y supervisión.

OE2. Documentar y verificar el intercambio de señales entre el PLC simulado y la escena Automated Warehouse, así como la organización del control del almacén.

OE3. Caracterizar la representación de las 54 posiciones del dashboard y establecer un procedimiento para contrastarlas con las operaciones del almacén.

OE4. Describir y analizar el modelo térmico implementado y su integración con el bloque PID_Temp mediante las variables de consigna, temperatura y salida de enfriamiento.

OE5. Documentar la comunicación Python/Snap7 y las funciones de supervisión y ajuste de consigna de la interfaz local desarrollada con Dash y Plotly.

OE6. Evaluar el funcionamiento del prototipo mediante la evidencia disponible y protocolos reproducibles de comunicación, almacenamiento, visualización y respuesta térmica.

OE7. Identificar las limitaciones del prototipo y proponer mejoras para su validación y una eventual implementación física.

1.6 Alcance y delimitaciones

El alcance comprende el sistema virtual y su documentación: programación del PLC referida por las fuentes, representación tridimensional, intercambio con Python, modelo térmico y visualización. El dashboard contiene una matriz de seis filas por nueve columnas. Esta distribución describe la interfaz; su orientación respecto de las ubicaciones físicas de la escena debe verificarse mediante el mapa de posiciones.

El prototipo incorpora una salida interpretada como porcentaje de enfriamiento y un modelo de retorno hacia una temperatura ambiente. No incluye dimensionamiento de evaporadores, selección de refrigerante, cálculo de cargas térmicas de una instalación ni instrumentación física. La temperatura simulada representa una variable agregada y no identifica gradientes entre posiciones o entre producto y aire.

No se consideran certificación de seguridad, validación sanitaria de alimentos, conexión efectiva a una nube, integración con un sistema empresarial de inventario ni evaluación económica de una instalación real. La pantalla accesible en 127.0.0.1:8050 corresponde a un servicio local. El uso previo del término «nube» en la presentación se corrige para evitar atribuir al prototipo una infraestructura que no está documentada.

1.7 Enfoque metodológico

El trabajo se organiza como un proyecto aplicado de integración y simulación. La unidad de análisis es el sistema virtual completo, con atención particular a sus interfaces. La revisión documental identifica el diseño declarado; la inspección estática del software determina comportamientos presentes en el código.

El desarrollo se estructura en seis etapas: delimitación de requisitos; revisión de fundamentos; diseño de arquitectura y datos; descripción de implementación; evaluación de evidencias y preparación de ensayos; discusión de limitaciones y mejoras.

2. FUNDAMENTOS TEÓRICOS Y ANTECEDENTES

2.1 Sistemas automatizados de almacenamiento

Un sistema automatizado de almacenamiento y recuperación, identificado habitualmente por la sigla AS/RS, coordina ubicaciones y equipos de movimiento para ingresar y retirar unidades de carga. Su análisis puede abordarse desde la mecánica, el control o la gestión logística. La memoria se concentra en el control y la supervisión, por lo que no resuelve el diseño estructural del rack ni una política óptima de secuenciación.

En la escena utilizada, las cargas son manipuladas por un transelevador. Este equipo combina desplazamiento a lo largo del pasillo, elevación y transferencia mediante horquillas. La coordinación depende de condiciones de llegada y de permisos de movimiento. Una lógica coherente impide, por ejemplo, iniciar un desplazamiento incompatible con una transferencia todavía en curso.

2.2 Controlador lógico programable y organización del programa

El PLC ejecuta una lógica de control a partir de entradas, memoria y condiciones internas, y actualiza salidas que gobiernan el proceso. En el proyecto su ejecución ocurre en una instancia simulada. TIA Portal actúa como entorno de ingeniería: permite configurar el dispositivo, preparar bloques, compilar y descargar el programa. No debe confundirse el entorno de edición con el componente que ejecuta efectivamente la lógica.

La organización por bloques facilita separar funciones. Un bloque de función puede conservar estados internos mediante un bloque de datos de instancia; un bloque de datos de intercambio puede exponer variables a una aplicación externa.

Los lenguajes gráficos permiten representar contactos, bobinas y bloques funcionales, mientras que un lenguaje estructurado facilita cálculos y tratamiento de datos.

2.3 Simulación y puesta en marcha virtual

La simulación construye una representación del proceso para estudiar comportamientos bajo hipótesis explícitas. En este caso la representación mecánica se delega a Factory I/O, la ejecución del PLC a PLCSIM Advanced y la dinámica térmica a Python. La utilidad del conjunto depende de la fidelidad necesaria para responder las preguntas del proyecto; no exige reproducir cada fenómeno físico, pero sí describir cuáles han sido simplificados.

El uso de un modelo virtual permite separar fallas lógicas de problemas de infraestructura. Una transición mal definida puede observarse aun cuando la conexión funcione; una lectura fallida puede impedir evaluar una secuencia que sería correcta. Esta separación orienta la metodología: primero verificar comunicación y semántica de señales, luego ejecutar maniobras y finalmente integrar el componente térmico.

Tampoco corresponde denominar al sistema un gemelo digital de una instalación existente. No se identifica un activo físico específico ni una sincronización con sensores reales. La expresión «modelo virtual» describe con mayor precisión la evidencia actual y mantiene abierta una evolución futura hacia un sistema conectado a una planta.

2.4 Factory I/O y PLCSIM Advanced

Factory I/O proporciona escenas tridimensionales con sensores y actuadores asociados a etiquetas. La escena «Automated Warehouse» se orienta al movimiento, almacenamiento y recuperación de cajas. La apariencia tridimensional facilita observar maniobras, pero la evaluación requiere además relacionarlas con señales del controlador.

El driver Siemens S7-PLCSIM permite enlazar la escena con distintas modalidades de simulación Siemens. La documentación oficial distingue la selección del driver y del modelo de PLC. Para el proyecto se documenta la modalidad S7-1500 con PLCSIM Advanced.

La guía oficial de configuración de PLCSIM Advanced establece «factoryio» como nombre de instancia predeterminado y permite cambiarlo mediante configuración de consola. Por lo tanto, el nombre no es una restricción universal e inmutable: debe coincidir con lo que espera el driver.

La misma documentación diferencia comunicación local mediante Softbus y comunicación mediante TCP/IP. Esa distinción explica por qué una conexión satisfactoria de Factory I/O no

demuestra necesariamente que Python pueda leer el PLC. Cada ruta necesita su configuración y sus verificaciones.

2.5 Memoria y comunicación S7 desde Python

La comunicación implementada utiliza la biblioteca Snap7 desde Python. El programa crea un cliente, configura una IP y lee o escribe zonas de bloques de datos. Para interpretar los bytes se emplean funciones como `get_bool`, `get_int` y `get_real`. La documentación de la biblioteca distingue estas conversiones y representa REAL mediante cuatro bytes.

Una dirección absoluta combina el bloque, el desplazamiento y el tipo. DB1.DBX0.0 designa un bit; DB1.DBW4 identifica una palabra a partir del byte 4; DB11.DBD4 designa cuatro bytes que el programa interpreta como REAL. El tipo no puede deducirse sólo del valor observado: los mismos bytes interpretados de otra forma pueden producir resultados sin significado para el proceso.

La compatibilidad del contrato de memoria requiere que el bloque exponga los offsets esperados. Para el correcto funcionamiento del proyecto se solicita desactivar el acceso optimizado para los bloques que se leen por dirección absoluta y habilitar el acceso externo correspondiente. La comprobación definitiva debe realizarse en el proyecto TIA utilizado, conservando capturas de atributos y permisos.

Una lectura exitosa tampoco garantiza coherencia temporal. Si una aplicación lee un bloque, modifica un campo y reescribe todos los bytes, puede volver a escribir valores que el PLC cambió después de la lectura.

2.6 Regulación de temperatura y variables del lazo

El control de temperatura compara una consigna con una variable de proceso y modifica una acción para reducir su diferencia. En el proyecto la consigna se denomina TEMP_SP, la temperatura TEMP_PV y la acción de enfriamiento PID_OUT. La variable de proceso proviene del modelo Python. El circuito de realimentación se cierra cuando esa temperatura se escribe en el PLC y se utiliza para calcular una nueva salida.

El error puede definirse como $e(t) = SP(t) - PV(t)$. En una representación PID ideal, la acción combina un término proporcional al error, uno integral y uno derivativo. Esa expresión ayuda

a explicar el principio de regulación, pero no equivale a la implementación interna exacta del bloque Siemens. El sentido de acción, la escala y los límites deben revisarse en la configuración efectiva.

$$u(t) = K_p \cdot e(t) + K_i \cdot \int e(t) dt + K_d \cdot de(t)/dt \quad (2.1)$$

En un sistema de enfriamiento, una temperatura mayor que la consigna requiere una acción que reduzca la temperatura. Si se conserva la definición de error anterior, esa situación produce error negativo. La configuración de la salida de frío debe transformar adecuadamente esa condición. No es correcto copiar una interpretación de calefacción sin comprobar cómo se utiliza OutputCool.

OutputCool entrega una variable REAL según la escala configurada. Las capturas del proyecto muestran enfriamiento habilitado y una salida de frío escalada de 0 % a 100 %, coherente con la normalización que emplea Python. Este porcentaje representa la acción de control del modelo; no equivale a una capacidad frigorífica en unidades de potencia sin una caracterización física adicional [P7, P8].

2.7 Modelo térmico simplificado

Un modelo agregado puede representar la evolución de una única temperatura a partir de efectos dominantes. El código recibido combina enfriamiento, retorno al ambiente y ruido aleatorio. Sus coeficientes no provienen de una identificación experimental documentada. Por ello, el modelo sirve como planta de ensayo del intercambio y del control, sin caracterizar una cámara comercial concreta.

El retorno ambiental introduce una tendencia hacia 8°C, mientras que el término de frío crece con la salida y con la distancia respecto de -5°C. Estos valores son parámetros de la simulación. La denominación MIN_TEMP del código no debe interpretarse como una garantía física de temperatura mínima: el programa no aplica un recorte estricto y el equilibrio depende también del retorno ambiental.

La formulación permite analizar equilibrio, sensibilidad y estabilidad numérica. Esas propiedades corresponden al modelo y al método discreto utilizado. La respuesta del sistema completo depende además del PID y de la secuencia de intercambio. El análisis matemático

puede anticipar consignas difíciles de alcanzar, pero no reemplaza la medición del lazo cerrado.

No se modelan masa de producto, apertura de puertas, infiltraciones, deshielo, humedad, potencia eléctrica, capacidad del evaporador ni distribución espacial. Una ampliación física necesitaría datos de geometría, materiales, cargas y condiciones de operación. En ausencia de tales antecedentes, agregar valores supuestos produciría una precisión aparente que no está justificada.

2.8 Supervisión mediante Dash y Plotly

Dash permite construir interfaces a partir de componentes y funciones de actualización. Los callbacks relacionan cambios de entradas con salidas visuales. En el prototipo una actualización periódica obtiene datos, reconstruye la matriz y agrega puntos a las curvas. Plotly representa las series de temperatura y consigna, mientras que componentes de entrada permiten enviar un nuevo valor al PLC.

El componente `dcc.Interval` incrementa un contador a intervalos configurados. Ese contador se utiliza en el eje horizontal del gráfico y está rotulado como ciclos. Su valor no constituye una marca de tiempo de adquisición. Si se producen retrasos, errores o diferentes condiciones de ejecución, la relación entre ciclos y segundos reales debe medirse.

El estado de posiciones y las series se guardan mediante `dcc.Store`. Dentro del código se describe este componente como almacenamiento de datos JSON en el navegador.

La documentación de Dash también advierte sobre el uso de variables globales mutables en aplicaciones con múltiples sesiones o procesos. El prototipo utiliza un cliente PLC y un simulador térmico globales. Por ello, se analiza como una aplicación local de demostración y se propone separar adquisición, simulación y presentación antes de ampliarla.

3. DISEÑO DEL SISTEMA

3.1 Requisitos funcionales

Los requisitos se formulan a partir del propósito del proyecto y de las funciones documentadas. Su inclusión no implica que todos estén completamente verificados. Cada requisito debe asociarse con una evidencia o una prueba, de modo que pueda distinguirse entre una función implementada, una función parcialmente observada y una mejora propuesta.

Tabla 3.1. Requisitos funcionales y criterio de verificación.

Identificador	Requisito	Criterio verificable
RF01	Intercambio PLC–Factory I/O	Cambio de entrada y salida coherente con la escena.
RF02	Movimiento hacia una ubicación	Destino válido y confirmación de maniobra terminada.
RF03	Representación de 54 posiciones	Etiquetas P01–P54 sin duplicación.
RF04	Lectura de variables térmicas	SP, PV y OUT coinciden con una tabla online.
RF05	Cambio de consigna	Valor escrito y lectura posterior concordantes.
RF06	Realimentación térmica	La salida del PLC modifica la PV calculada.
RF07	Tendencia de temperatura	Curvas identificadas y eje temporal interpretado.
RF08	Identificación de anomalías	Registro de fallas y datos inválidos distinguibles.

3.2 Requisitos de información y reproducibilidad

La reproducción del sistema requiere identificar versiones, archivos, configuración y condiciones iniciales. Las versiones utilizadas de los software son TIA Portal V19, PLCSIM Advanced V6.0, PLC S7-1500 simulado y Factory I/O v2.5.8. Los programas declaran la IP, rack, slot, tamaños de bloques y offsets.

Tabla 3.2. Configuración conocida y antecedentes pendientes.

Elemento	Dato disponible	Estado documental
TIA Portal	V19	Referido y visible en el entorno.
S7-PLCSIM Advanced	V6.0 Upd1	Visible en PLCSIMAdvanced.png [P7].
Factory I/O	v2.5.8, Automated Warehouse	Visible en el video.
PLC simulado	CPU 1511-1 PN, familia S7-1500	Identificado en capturas de TIA Portal [P7].
Python y bibliotecas		Código disponible, sin inventario de versiones.

3.3 Arquitectura y responsabilidades

La arquitectura integra dos procesos simulados: el movimiento del almacén y la dinámica térmica. El primero se representa en Factory I/O y recibe acciones del PLC; el segundo se calcula en Python a partir de la salida de enfriamiento. El PLC constituye el punto de intercambio de variables de control, mientras que el navegador presenta información y permite modificar la consigna.

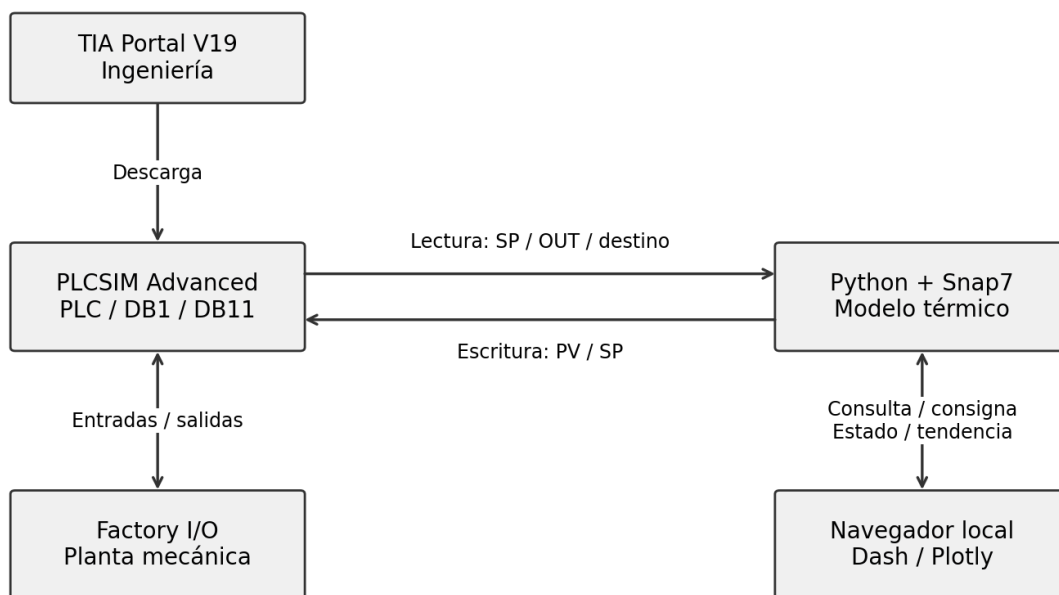


Figura 3.1. Arquitectura reconstruida del prototipo. Las flechas muestran intercambio operativo; TIA Portal configura y descarga el programa. Fuente: elaboración a partir de P2–P5.

La separación de responsabilidades es la siguiente: TIA Portal define el programa; PLCSIM Advanced lo ejecuta; Factory I/O actualiza la escena y sus señales; Snap7 transporta datos entre Python y los DB; el módulo `plc_almacen` implementa acceso y modelo térmico; `dashboard.py` organiza la presentación y activa el paso de simulación. Esta última responsabilidad mezcla visualización y dinámica, aspecto que debe considerarse al evaluar el prototipo.

La interfaz no se conecta directamente con los actuadores de Factory I/O. Lee variables del PLC y escribe consigna y temperatura simulada. Tampoco existe en el código un canal documentado de temperaturas por cada ubicación. La ocupación y el control térmico conviven

en el panel, pero no están acoplados mediante una carga térmica dependiente del número de pallets.

3.4 Contrato de datos del almacén

El módulo plc_almacen define DB_NUMBER = 1 y DB_SIZE = 20. Cada consulta lee veinte bytes desde el inicio de DB1 y extrae START, STOP y el destino. La captura «UBICACION CAJAS.png» identifica el bloque de instancia UBICACION DE CAJAS_DB, con entradas, salidas y variables estáticas. START y STOP ocupan los bits 0.0 y 0.1; la salida TARET POSITION es de tipo Int y comienza en el byte 4, en concordancia con la lectura de Python [P4, P7].

Tabla 3.3. Variables leídas por la interfaz desde DB1.

Variable	Dirección	Tipo	Uso en Python
START	DB1.DBX0.0	BOOL	Se devuelve al dashboard, sin condicionar su acumulación.
STOP	DB1.DBX0.1	BOOL	Se devuelve al dashboard, sin condicionar su acumulación.
TARGET_POSITION	DB1.DBW4	INT	Se agrega al conjunto si está entre 1 y 54.

La visualización actualiza las posiciones a partir del valor de destino recibido. Por ello, la ocupación representada debe interpretarse como un registro de destinos observados durante la ejecución.

La llamada de FB1 «UBICACION DE CAJAS» utiliza DB1 como bloque de instancia y conecta su salida TARGET POSITION a %QW30. La estructura de DB1 ubica esa salida en el byte 4. Ambas capturas permiten establecer el vínculo entre DB1.DBW4, leído por Python, y la palabra de salida enviada a la escena. Se trata de direcciones de áreas distintas que representan el mismo parámetro en esta llamada [P7].

3.5 Contrato de datos térmicos

DB11 «Temperatura_Dash» contiene TEMP_SP, TEMP_PV y PID_OUT, todos de tipo Real, en los offsets 0.0, 4.0 y 8.0. La captura «Temperatura_Dash_DB.png» coincide con los doce bytes interpretados por Python. La llamada de PID_Temp confirma la conexión de estos campos con Setpoint, Input y OutputCool, respectivamente [P4, P7].

Tabla 3.4. Variables térmicas y autoridad prevista.

Variable	Dirección	Magnitud	Escritor previsto
TEMP_SP	DB11.DBD0	Consigna en °C	Operador mediante Python o interfaz autorizada.
TEMP_PV	DB11.DBD4	Temperatura simulada en °C	Modelo de planta en Python.
PID_OUT	DB11.DBD8	Salida de frío interpretada en %	Controlador PID del PLC.

La autoridad prevista describe el diseño funcional. La implementación actual escribe el bloque completo al modificar SP o PV; por ello, físicamente retransmite también bytes correspondientes a otros campos. La diferencia entre autoridad funcional y alcance real de escritura es central para analizar la coherencia de datos.

Una variante más robusta mantendría operaciones de escritura restringidas al campo modificado, además de una política de exclusión o serialización del acceso. Para la consigna, se puede añadir una solicitud con identificación y una confirmación del PLC. Para PV, se requiere una periodicidad estable y una señal de validez. Estas medidas se proponen como diseño futuro y no como modificaciones ya ejecutadas. Con lo anterior se da a entender que si se requiere de un programa mas robusto se puede limitar la facilidad de modificar PV.

3.6 Señales del proceso mecánico

El mapeo se verificó comparando «PLC Tags .png», la llamada de FB1 en «MAIN.png» y «DRIVERS .png» de Factory I/O. Las tres capturas coinciden en nueve entradas booleanas, cinco salidas booleanas y una salida de posición. La tabla 3.5 corresponde a esta configuración del proyecto CAJA SLOCAS_V19 y sustituye el mapeo anterior. Se conservan

los nombres de TIA Portal tal como aparecen, incluidas sus abreviaturas y diferencias de escritura [P7].

Tabla 3.5. Señales verificadas en TIA Portal y en el driver de Factory I/O [P7].

Señal en TIA Portal / Factory I/O	Dirección y tipo	Sentido y función
START / Start	%I20.0 · Bool	Entrada: mando de inicio.
STOP / Stop	%I20.1 · Bool	Entrada: mando de parada.
AT MIDL / At Middle	%I20.2 · Bool	Entrada: detección de posición central.
AT ENTRY / At Entry	%I20.3 · Bool	Entrada: detección en la entrada.
at load / at load	%I20.4 · Bool	Entrada: detección en carga.
RESET / Reset	%I20.5 · Bool	Entrada: mando de reinicio.
MOVIN X / Moving X	%I20.6 · Bool	Entrada: señal de movimiento en X.
AT RIGHT / At Right	%I20.7 · Bool	Entrada: detección a la derecha.
AT LEFT / At Left	%I21.0 · Bool	Entrada: detección a la izquierda.
LOAD CONVEYOR / Load Conveyor	%Q20.0 · Bool	Salida: transportador de carga.
FORS RIGHT / Forks Right	%Q20.1 · Bool	Salida: horquillas hacia la derecha.
ENTRY CONVEYOR / Entry Conveyor	%Q20.2 · Bool	Salida: transportador de entrada.
FORS LEFT / Forks Left	%Q20.3 · Bool	Salida: horquillas hacia la izquierda.
LIFT / Lift	%Q20.4 · Bool	Salida: elevación.
TARGET POSITION / Target Position	%QW30 · Int	Salida: posición de destino.

La tabla 3.5 recoge las etiquetas cuya asignación es visible en las capturas revisadas. En particular, %I21.0 corresponde a At Left. [P7].

3.7 Secuencia funcional de referencia

La secuencia funcional de referencia considera espera, recepción, toma, desplazamiento, depósito y retorno. Organiza la explicación y los ensayos como una propuesta de formalización. Las seis redes de FB1 incorporadas en el anexo C permiten contrastar con la implementación: esta utiliza bits de paso enclavados, condiciones de sensores y temporizadores. No debe identificarse cada estado del esquema con un único bit del programa, porque varios pasos pueden permanecer activos simultáneamente [P8].

En espera, el equipo debe encontrarse en una condición conocida. La recepción verifica disponibilidad de carga y de destino. La toma habilita la transferencia hacia el transelevador. El desplazamiento conserva la identidad de la orden. El depósito requiere una confirmación coherente antes de declarar ocupada la ubicación. Finalmente, el retorno restablece una condición que permite aceptar otra orden.

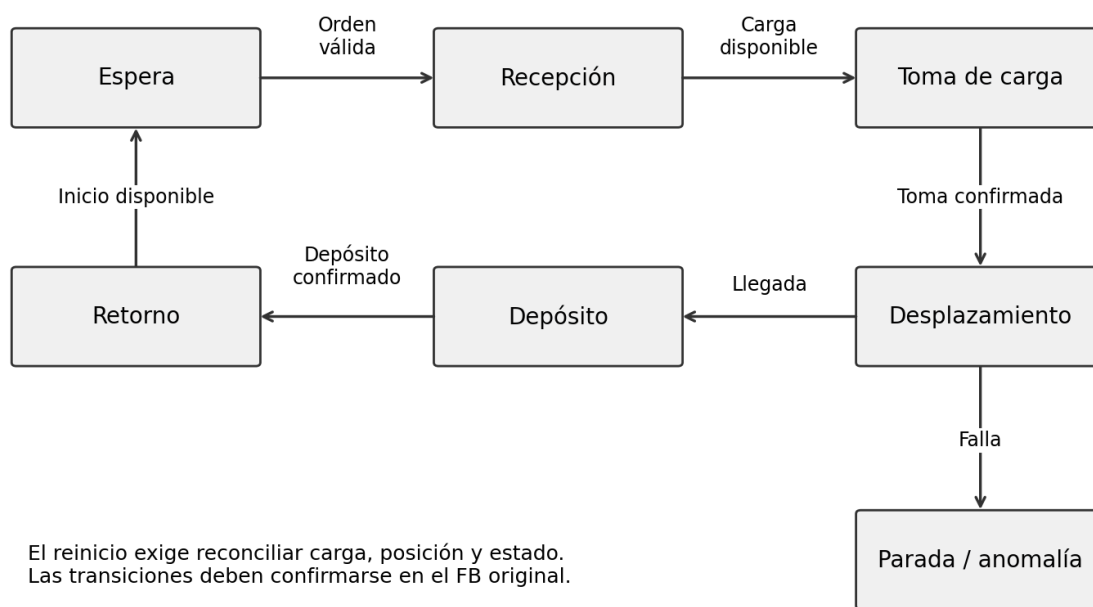


Figura 3.3. Secuencia funcional de referencia para documentar y validar una maniobra.
El esquema expresa etapas del proceso; las redes concretas y sus enclavamientos se analizan en la sección 4.3 y el anexo C [P8].

Cada transición debe asociarse con una condición observable. El uso exclusivo de temporizadores puede permitir una animación, pero no necesariamente demostrar que la carga

alcanzó la ubicación. El registro de las condiciones de transición ayuda a reconocer dónde se interrumpe el proceso y qué salidas deben desactivarse ante un error.

El reinicio del estado lógico debe mantener la coherencia con la escena: una carga puede permanecer en una ubicación aunque se reinicien las variables de control. Por ello, la interpretación de la ocupación requiere reconciliar el registro lógico con las posiciones observadas.

3.8 Representación de las posiciones

El dashboard construye celdas P01 a P54 en orden ascendente, con nueve columnas. Si p es el número de posición, la fila de la interfaz se obtiene dividiendo $p - 1$ por nueve y tomando la parte entera; la columna se obtiene con el resto de esa división. La numeración de filas y columnas puede expresarse desde uno.

$$\text{fila}(p) = \lfloor (p - 1)/9 \rfloor + 1; \text{columna}(p) = ((p - 1) \bmod 9) + 1 \quad (3.1)$$

Esta relación describe únicamente la malla visual. El primer elemento aparece arriba a la izquierda del panel, mientras que en un almacén las posiciones pueden numerarse desde otra esquina o seguir un orden propio del simulador. La correspondencia debe comprobarse con posiciones extremas e intermedias y registrarse en una tabla.

Para el conteo se utiliza un conjunto, por lo que un destino repetido no aumenta el total después de su primera incorporación. La propiedad evita duplicados dentro de una sesión, pero no valida el inventario. Si un destino cambia durante una maniobra fallida, también puede quedar incorporado aunque no se haya depositado una carga.

$$N_{\text{registradas}} = |S|; N_{\text{restantes}} = 54 - |S|; \text{Porcentaje} = 100 \cdot |S|/54 \quad (3.2)$$

La interfaz denomina «ocupadas» a las posiciones del conjunto S . En el análisis se utiliza «registradas» cuando se desea enfatizar que el dato proviene de destinos observados. Esta precisión mantiene la relación con las etiquetas reales sin atribuirles una confirmación que el software no implementa.

3.9 Diseño de supervisión térmica

El panel térmico integra cuatro elementos: valor actual, estado respecto de una banda, curvas de PV y SP, y formulario de cambio de consigna. La banda se calcula como una diferencia

absoluta de hasta 2°C respecto de SP. El criterio es informativo y se encuentra escrito directamente en el callback.

El ingreso de una nueva consigna se confirma mediante el retorno de la función de escritura. El código comprueba que exista un valor, lo convierte a número real y captura excepciones. No establece límites mínimo y máximo ni realiza una lectura posterior como confirmación funcional. La comprobación del valor efectivamente utilizado por el PLC debe añadirse al protocolo de pruebas.

La curva histórica guarda una muestra por actualización completada. En la implementación recibida se agregan puntos sin limitar la longitud de las listas. Para ensayos breves esto permite observar la evolución; para sesiones prolongadas puede aumentar el volumen de datos transferidos y procesados. No se dispone de una medición de rendimiento que permita fijar un límite seguro.

El rótulo «PLC Online (Simulado)» se crea como texto fijo en el diseño del panel. Su color no está conectado a una prueba dinámica de salud del enlace. Por ello, la ausencia de cambios en ese rótulo no demuestra continuidad de comunicación. Una mejora debería distinguir conexión, frescura de muestra y error de adquisición.

4. IMPLEMENTACIÓN Y SIMULACIÓN

4.1 Preparación del entorno Siemens

El entorno utiliza una CPU 1511-1 PN de la familia S7-1500, identificada en TIA Portal, y S7-PLCSIM Advanced V6.0 Upd1. El panel del simulador muestra la instancia factoryio en 192.168.0.1. Las propiedades del proyecto confirman la opción Support simulation during block compilation habilitada. [P7, P8].

La instancia se denomina «factoryio» según la configuración documentada. Para Factory I/O se selecciona el driver Siemens S7-PLCSIM y la modalidad correspondiente a S7-1500 con PLCSIM Advanced.

La comunicación de Python utiliza la ruta TCP/IP hacia 192.168.0.1, rack 0 y slot 1. La lectura de variables se realiza mediante las direcciones del contrato de datos descrito en este capítulo.

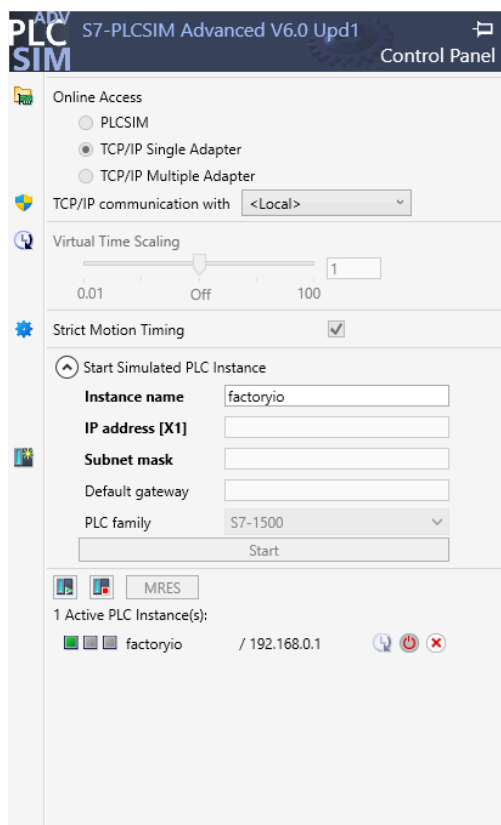


Figura 4.1. Panel de S7-PLCSIM Advanced V6.0 Upd1: acceso TCP/IP Single Adapter, instancia factoryio y dirección 192.168.0.1. Fuente: captura PLCSIMAdvanced.png [P7].

Las propiedades de la CPU muestran Full access (no protection) y el permiso de comunicación PUT/GET activado. Las propiedades de simulación del proyecto muestran Support simulation during block compilation seleccionado y Enable usage of S7-1500 blocks in virtual PLCs sin seleccionar. Ambas capturas se incorporan en las figuras C.11 y C.12 [P8].

Los ajustes de comunicación descritos corresponden al entorno simulado del proyecto.

4.2 Configuración de la escena y señales

La escena Automated Warehouse representa el rack, el transelevador y los transportadores de transferencia. La captura «Panel.png» permite reconocer la disposición del equipo y las ubicaciones del almacén. La imagen se utiliza para describir la configuración mecánica. [P7].



Figura 4.2a. Vista general del almacén y del transelevador en Factory I/O. Fuente: captura Panel.png, carpeta FACTORY IO [P7].

La configuración utiliza el driver Siemens S7-PLCSIM y el modelo S7-1500 con PLCSIM Advanced. En I/O Config se selecciona WORD. Los puntos booleanos tienen offset 20, con 15 entradas y 10 salidas configuradas; las palabras tienen offset 30, con cero entradas y una salida. Estos conteos definen la capacidad de intercambio configurada, no el número de etiquetas efectivamente asignadas. El mapeo visible utiliza nueve entradas y cinco salidas booleanas, además de Target Position [P7].

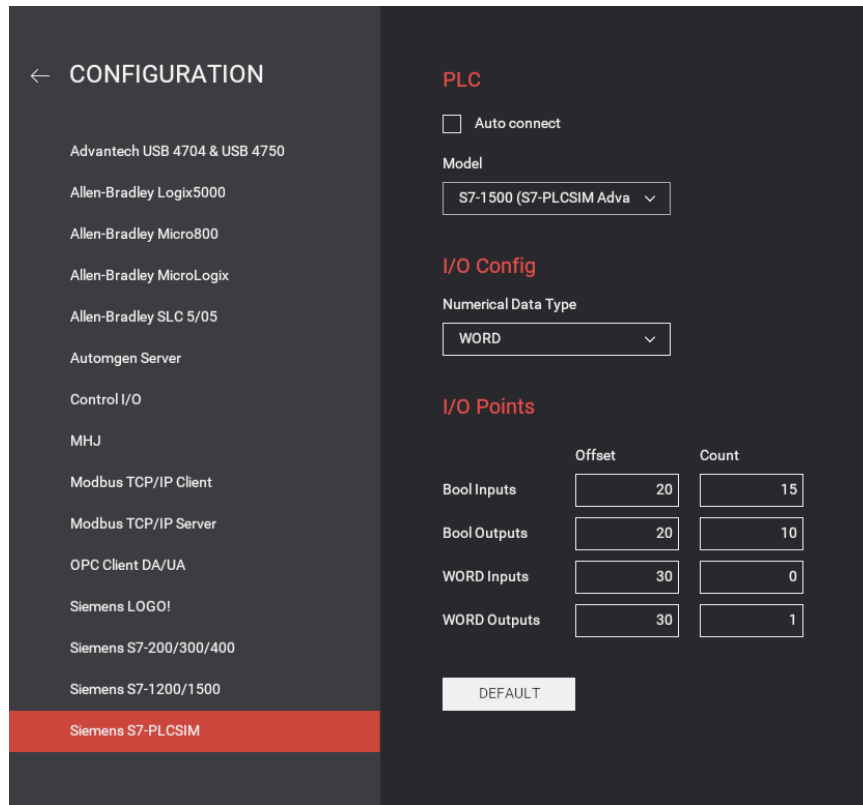


Figura 4.2b. Selección del driver, tipo numérico, offsets y cantidad de puntos de E/S. Fuente: captura CONFIGURACION .png [P7].

La correspondencia de etiquetas se presenta en la tabla 3.5. La salida Target Position se intercambia como una palabra y se declara Int en TIA Portal; para los destinos de 1 a 54 la representación es compatible. La verificación dinámica debe registrar cada orden y detección, identificando si procede de la secuencia, de un mando manual o de una señal forzada. Las capturas acreditan la configuración, pero no sustituyen esa prueba.

4.3 Organización observable del control del almacén

La captura MAIN.png muestra dos redes del programa principal. La primera llama a FC9000 «MHJ-PLC-Lab-Function-S71500». La segunda ejecuta FB1 «UBICACION DE CAJAS», asociado al bloque de instancia DB1 «UBICACION DE CAJAS_DB». Sus entradas reciben mandos y detecciones de la escena; sus salidas gobiernan transportadores, horquillas, elevación y posición de destino. La figura 4.3a permite seguir sus direcciones, sin recurrir a una representación de otra versión del proyecto [P7].

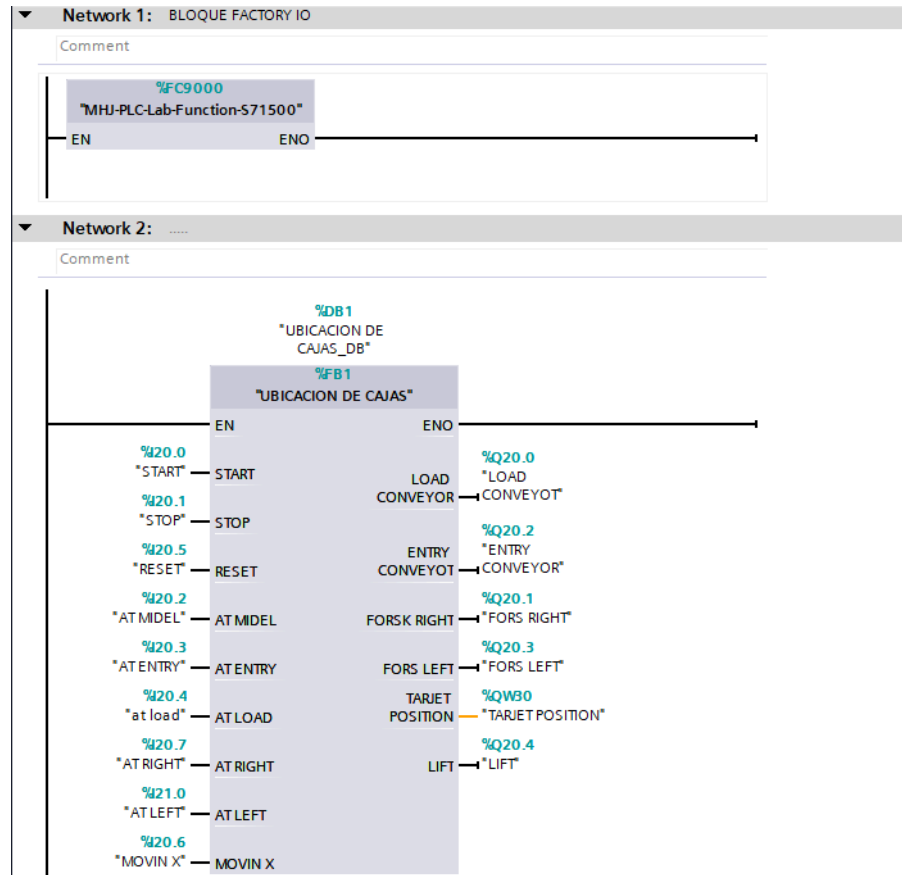


Figura 4.3a. Llamadas de FC9000 y FB1 en el programa principal. FB1 utiliza DB1 y entrega el destino en %QW30. Fuente: captura MAIN.png [P7].

La estructura de instancia contiene entradas de mando y detección, salidas de actuación y la salida de destino Int en el byte 4. La declaración de FB1 confirma paso 0 a paso 6 en los offsets 8.0 a 8.6, MOV1 y MOVEM en 10.0 y 12.0, SAL en 14.0 y auxiliares desde 18.0. Las seis redes recibidas permiten analizar el uso de paso 0 a paso 5. La declaración completa se reproduce en la figura C.1 [P7, P8].

UBICACION DE CAJAS_DB		Name	Data type	Offset	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	Input									
2	START	Bool	0.0	false						
3	STOP	Bool	0.1	false						
4	RESET	Bool	0.2	false						
5	ATMIDEL	Bool	0.3	false						
6	ATENTRY	Bool	0.4	false						
7	ATLOAD	Bool	0.5	false						
8	ATRIGHT	Bool	0.6	false						
9	ATLEFT	Bool	0.7	false						
10	MOVIN X	Bool	1.0	false						
11	Output									
12	LOAD CONVEYOR	Bool	2.0	false						
13	ENTRY CONVEYOT	Bool	2.1	false						
14	FORSK RIGHT	Bool	2.2	false						
15	FORS LEFT	Bool	2.3	false						
16	TARGET POSITION	Int	4.0	0						
17	LIFT	Bool	6.0	false						
18	InOut									
19	Static									
20	paso 0	Bool	8.0	false						
21	paso 1	Bool	8.1	false						
22	paso 2	Bool	8.2	false						
23	paso 3	Bool	8.3	false						
24	paso 4	Bool	8.4	false						
25	paso 5	Bool	8.5	false						
26	paso 6	Bool	8.6	false						
27	MOV1	Int	10.0	0						
28	MOVEM	Int	12.0	0						
29	SAL	Real	14.0	0.0						
30	AUX1	Bool	18.0	false						
31	AUX 2	Bool	18.1	false						
32	AUX 3	Bool	18.2	false						
33	AUX 4	Bool	18.3	false						

Figura 4.3b. Entradas, salidas y memoria estática de UBICACION DE CAJAS_DB. Fuente: captura DB UBICACION CAJAS.png [P7].

Las figuras C.2 a C.7 contienen las seis redes internas de FB1. La descripción siguiente se basa en la lectura estática de contactos, bobinas Set/Reset, temporizadores y transferencias. No equivale a una prueba online de tiempos o movimientos. Los nombres AT LOAD, AT MIDEL y MOVIN X se conservan como aparecen en el programa; un contacto negado se interpreta como condición lógica falsa, sin atribuirle una polaridad física no comprobada [P8].

Inicio y reinicio. En la red 1, START enclava paso 0. RESET borra paso 0 a paso 5, los transportadores, las órdenes de horquillas y LIFT, y transfiere cero a MOVEM y al destino.

Recepción y toma. En la red 2, paso 0 activa ENTRY CONVEYOR y LOAD CONVEYOR. Bajo ese paso, el contacto negado AT LOAD establece paso 1. A continuación, paso 1 borra paso 0, desactiva ambos transportadores y activa FORS LEFT y paso 2. Esta organización permite que una transición habilite otra red lógica durante la misma exploración; no establece exclusión mutua permanente entre todos los bits de paso.

Elevación y preparación del destino. En la red 3, paso 2 inicia TX (DB2), un TON de 3 s. Su salida borra paso 1; con AT MIDEL falso activa LIFT y, con AT LEFT verdadero, habilita T2 (DB3), de 2 s. La salida de T2 desactiva FORS LEFT y activa paso 3. En la red 4, paso 3

inicia T3 (DB4), de 3 s; al terminar, borra paso 2 y activa paso 4. Los tiempos son valores programados, no duraciones de ciclo medidas.

Asignación de destino. En la red 5, el contador CTU CONTADOR (DB5) recibe en CU la condición paso 4 y AT LOAD, tiene RESET conectado a R y entrega CV a MOV1. El valor PV es 1; no representa el límite de 54 ubicaciones. Con paso 4 activo y MOV1 mayor que cero, un MOVE copia MOV1 al destino. También se habilita TEM (DB7), TON de 7 s, pero su salida Q no gobierna ninguna acción visible en esta captura. No se observa una comparación que limite MOV1 a 54.

Depósito y retorno. En la red 6, paso 4 con MOVIN X falso inicia TH (DB6), de 4 s. Al finalizar se activa FORS RIGHT y se borra paso 3; con AT RIGHT verdadero se desactiva LIFT y se establece paso 5. Este paso inicia TK (DB8), de 4 s, cuya salida desactiva FORS RIGHT. Cuando además AT MIDEI es verdadero, se borra paso 4, se transfiere 21474 al destino y se borra paso 5. Finalmente, destino igual a 21474, AT MIDEI verdadero y MOVIN X falso habilitan TU (DB9), de 1 s, cuya salida vuelve a activar paso 0.

El valor 21474 funciona en la lógica como una orden especial de retorno, distinta de las 54 ubicaciones del panel. Su efecto mecánico debe confirmarse en la escena. Python lo excluye del conjunto de ocupación por estar fuera del intervalo 1–54; en cambio, incorpora los destinos válidos al observarlos, sin esperar la finalización de la red 6. Las nuevas redes, por tanto, mantienen la distinción entre destino ordenado y depósito confirmado.

La descripción del control se limita a la secuencia de almacenamiento visible en las seis redes. Los ensayos del capítulo 5 permiten contrastar su ejecución y la correspondencia entre los destinos lógicos y la escena.

4.4 Inicialización del cliente Python

La función `crear_cliente_plc` instancia `snap7.client.Client` ejecuta y conecta con IP 192.168.0.1, rack 0 y slot 1. Si la llamada finaliza, imprime un mensaje de conexión. El código no incorpora una secuencia explícita de reintentos ni una política de reconexión. La inicialización del cliente ocurre al cargar el módulo principal del dashboard.

Esta estructura establece una dependencia directa: si la conexión inicial falla, la aplicación puede no llegar a ofrecer su interfaz. Durante una demostración controlada se puede iniciar

primero la instancia del PLC y después Python. Para un sistema más robusto sería conveniente permitir que la interfaz informe el estado de desconexión y que un servicio de adquisición gestione la recuperación.

La conexión inicial tampoco representa una garantía permanente. La continuidad exige lecturas recientes, manejo de excepciones y reconocimiento del estado del PLC.

4.5 Lectura y acumulación de destinos

`leer_start_stop_target` solicita los veinte bytes de DB1 y extrae las variables definidas. `actualizar_estado` transforma la lista guardada en un conjunto, incorpora el destino válido y vuelve a ordenar los elementos. El rango admitido es inclusivo entre 1 y 54. Los valores fuera de ese intervalo no generan una nueva posición marcada.

La actualización es importante respecto de un destino repetido: observar varias veces la misma posición mantiene el mismo conjunto. Sin embargo, no es reversible: no existe una operación de eliminación asociada a recuperación. Tampoco se consulta una señal «Done», una confirmación de depósito o un identificador de transacción.

Por ejemplo, si se observan sucesivamente los destinos 3, 3 y 8, el conjunto termina con dos posiciones. Si el movimiento hacia 8 se cancela, el código no dispone de información para deshacer su incorporación. Este ejemplo es una deducción lógica del algoritmo, no un ensayo realizado con el PLC.

4.6 Modelo de enfriamiento implementado

El modelo recibe la temperatura interna y el porcentaje de `OutputCool`. Calcula un término de frío proporcional a la distancia respecto de `MIN_TEMP`, un retorno hacia `AMBIENT_TEMP` y una perturbación uniforme. La actualización se realiza con `DT_SECONDS = 1,0`. La expresión siguiente transcribe matemáticamente la función del código [P5].

$$T(k+1) = T(k) + \Delta t \cdot [-0,35 \cdot u(k) \cdot \max(T(k)+5, 0) + 0,03 \cdot (8-T(k))] + \eta(k) \quad (4.1)$$

Aquí $u(k) = \text{OutputCool}(k)/100$ y $\eta(k)$ se obtiene de $(\text{random.random()} - 0,5) \cdot 0,08 \cdot 2$. Bajo la distribución utilizada, el ruido se encuentra entre $-0,08$ y $0,08$ °C por actualización. El término aleatorio se suma fuera del producto por Δt , por lo que cambiar el periodo sin reconsiderar el ruido altera su efecto por unidad de tiempo.

Tabla 4.1. Parámetros extraídos del modelo Python.

Parámetro	Valor	Interpretación en el modelo
AMBIENT_TEMP	8,0 °C	Referencia del retorno ambiental.
MIN_TEMP	-5,0 °C	Referencia del término de enfriamiento.
COOLING_GAIN	0,35	Coefficiente multiplicado por salida normalizada.
RETURN_GAIN	0,03	Coefficiente de retorno ambiental.
NOISE_AMPL	0,08 °C	Semiamplitud del ruido por paso.
DT_SECONDS	1,0 s nominal	Paso usado en la ecuación y en el intervalo visual.

Para $T > -5$ °C y salida constante, la parte determinista se puede escribir como $T(k+1) = a \cdot T(k) + b$. Con $\Delta t = 1$, $a = 0,97 - 0,35u$ y $b = 0,24 - 1,75u$. Si se supone $0 \leq u \leq 1$, el coeficiente a queda entre 0,62 y 0,97, dentro del intervalo de estabilidad discreta de ese sistema de primer orden.

Este resultado demuestra una propiedad de la planta discretizada con salida fija, no la estabilidad del PID conectado. El controlador introduce otra dinámica y puede saturar o responder a lecturas retrasadas. Además, el código no limita explícitamente OutputCool al intervalo supuesto; la validez del análisis depende de verificar la escala y límites del PLC.

4.7 Equilibrio y rango alcanzable del modelo

Al igualar a cero el incremento determinista se obtiene la temperatura de equilibrio para una salida constante. El cálculo supone $T > -5$ °C, ruido nulo y parámetros sin variación. La ecuación permite interpretar la relación entre esfuerzo de enfriamiento y temperatura esperada.

$$T^* = (0,24 - 1,75u)/(0,03 + 0,35u) \quad (4.2)$$

Tabla 4.2. Equilibrios calculados de la planta, no mediciones experimentales.

OutputCool	u	T* calculada
0 %	0,00	8,000 °C
10 %	0,10	1,000 °C
25 %	0,25	-1,681 °C

OutputCool	u	T* calculada
50 %	0,50	-3,098 °C
100 %	1,00	-3,974 °C

El valor -5 °C utilizado en MIN_TEMP no coincide con el equilibrio a máxima salida. El retorno ambiental continúa aportando calentamiento, de modo que, bajo estas hipótesis, el equilibrio a 100 % es aproximadamente -3,974 °C. Por ello, una consigna de -5 °C no sería alcanzable en régimen determinista con salida limitada a 100 %. El ruido puede producir fluctuaciones, pero no convierte esa consigna en un equilibrio regulable.

Para una temperatura objetivo T*, la salida normalizada de equilibrio se obtiene despejando la misma relación. Esta expresión permite seleccionar consignas de ensayo compatibles con el modelo antes de evaluar el PID.

$$u^* = 0,03 \cdot (8 - T^*) / [0,35 \cdot (T^* + 5)] \quad (4.3)$$

A 5 °C se obtiene una salida de aproximadamente 2,571 % y a -1 °C una salida de aproximadamente 19,286 %. Son referencias matemáticas para interpretar un régimen estacionario ideal. No deben compararse con una captura transitoria como si existiera identidad instantánea: el estado integral, las perturbaciones y el momento de muestreo influyen en la salida observada.

El modelo también permite estimar una constante temporal discreta equivalente para salida fija mediante $\tau = -\Delta t / \ln(a)$. Para $u = 0$ resulta aproximadamente 32,8 s, mientras que para $u = 1$ resulta aproximadamente 2,09 s. Estas magnitudes caracterizan el retorno del modelo hacia su equilibrio y muestran que su rapidez cambia con la salida; no corresponden a tiempos de establecimiento medidos del sistema completo.

4.8 Ejecución del lazo térmico

La clase SimuladorTemperatura intenta inicializar su temperatura con PV del PLC. Si encuentra NaN o una excepción, utiliza AMBIENT_TEMP. En cada paso lee SP, PV y OUT; después actualiza su estado interno utilizando OUT y escribe la nueva temperatura en DB11. El valor PV leído durante el paso no sustituye al estado interno, por lo que la continuidad de la planta reside en el objeto Python.

Esta decisión implica que una modificación externa de PV después de la inicialización no reinicia necesariamente el modelo. Si se desea ensayar otra condición inicial, debe establecerse un procedimiento explícito para reinicializar el objeto o incorporar una función dedicada. Registrar únicamente el valor del DB puede ser insuficiente para conocer la condición interna del simulador.

El paso se invoca desde `update_dashboard`. Por tanto, la evolución de la planta depende de las solicitudes de actualización de la interfaz. En una sesión única el periodo nominal es un segundo, pero la programación no constituye una tarea independiente de tiempo real.

4.9 Integración con PID_Temp

La integración térmica utiliza DB11 «Temperatura_Dash» para el intercambio y DB12 «PID_Temp_1» como instancia del regulador. Se conserva la captura de variables de la figura 4.4: SP = 5,0 °C, PV = 4,943299 °C y OUT = 2,461716. Estos valores corresponden a un instante del desarrollo y no se consideran simultáneos a las nuevas capturas del bloque o del dashboard [P2, P7].

Name	Data type	Offset	Start value	Monitor value
▼ Static				
■ TEMP_SP	Real	0.0	3.0	5.0
■ TEMP_PV	Real	4.0	0.0	4.943299
■ PID_OUT	Real	8.0	0.0	2.461716

Figura 4.4. Variables térmicas visibles en la presentación. Fuente: P2, diapositiva 9.

La llamada documenta TEMP_SP en DB11.DBD0 conectado a Setpoint, TEMP_PV en DB11.DBD4 conectado a Input y OutputCool conectado a PID_OUT en DB11.DBD8, con DB12 como instancia. Las nuevas pantallas de configuración complementan esta conexión al mostrar enfriamiento habilitado, modo automático configurado tras reinicio y escalas de salida. [P7, P8].

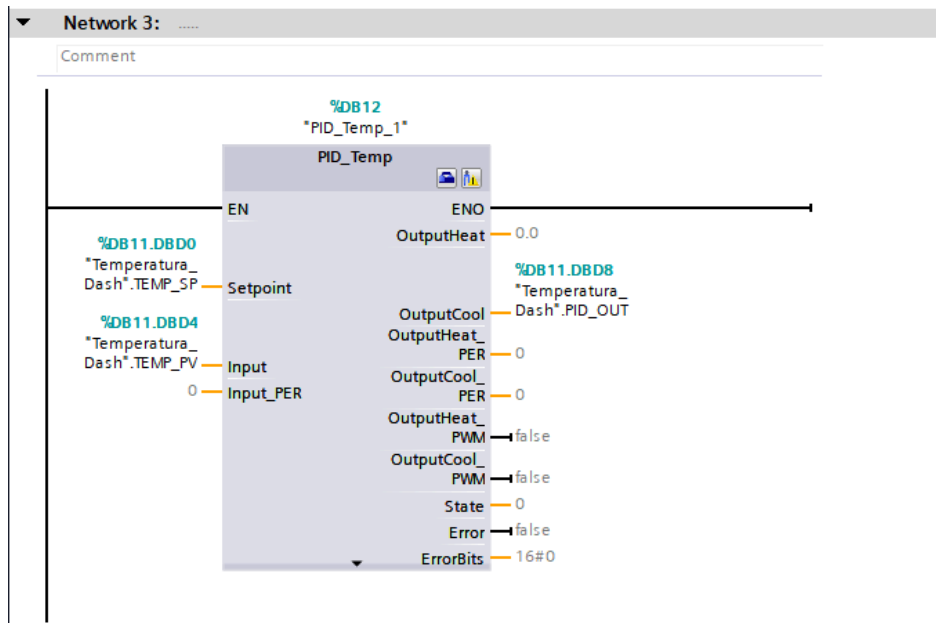


Figura 4.5. Llamada de PID_Temp con Setpoint, Input y OutputCool enlazados a DB11. Fuente: captura MAIN2.png, red 3 [P7].

La configuración básica selecciona Temperature y °C. Activate Mode after CPU restart está habilitado con Automatic mode y Activate cooling aparece seleccionado. No se selecciona funcionamiento maestro ni esclavo en cascada. Input se emplea como entrada del proceso y la pantalla de escalado indica Input_PER deshabilitado. Estos ajustes se reproducen en las figuras C.8 y C.10; describen la configuración, sin probar el modo efectivo en un instante de ejecución [P8].

La salida utiliza conmutación de parámetros para calentamiento y enfriamiento y cooling factor = 1,0. OutputCool y OutputHeat aparecen habilitadas. Para frío se representa el tramo de PidOutputSum de -100 % a 0 %, escalado a OutputCool entre 100 % y 0 %; para calor, de 0 % a 100 %, con salida de 0 % a 100 %. Se configura un valor sustitutivo de 0,0 % mientras exista un error. La escala de frío respalda la interpretación porcentual que utiliza Python [P8].

Los límites configurados de la variable de proceso son -5,0 °C y 10,0 °C. Estos valores pertenecen al regulador y no constituyen un recorte de la ecuación de la planta Python ni una banda sanitaria para productos. Aunque coincide numéricamente con el límite inferior del PID, MIN_TEMP = -5 °C tiene en el modelo la función específica descrita en su ecuación. Debe observarse el diagnóstico del regulador al aproximarse o excederse los límites de proceso [P8].

Esta sección documenta la integración del regulador con el intercambio de variables y el modelo térmico. La evaluación cuantitativa de su respuesta se plantea mediante los ensayos del capítulo 5.

La lectura de valores próximos a la consigna en una captura es compatible con regulación, pero no permite calcular sobreimpulso, establecimiento o error medio. Tales indicadores requieren una serie temporal y condiciones iniciales conocidas. La memoria conserva esta evidencia como antecedente y reserva los resultados cuantitativos para ensayos documentados.

4.10 Escritura de variables y concurrencia

Las funciones `escribir_temp_pv` y `escribir_temp_sp` utilizan un patrón de lectura, modificación y escritura del bloque completo. Para PV puede reutilizarse el buffer leído por paso; para SP se obtiene un buffer nuevo. En ambos casos `db_write` se realiza desde el byte cero con los doce bytes de temperatura.

Una secuencia posible ilustra el riesgo: Python lee SP, PV y OUT; el PLC calcula un OUT nuevo; Python modifica PV en el buffer previo y escribe los doce bytes. El OUT anterior puede retransmitirse. Si simultáneamente se modifica SP desde otro callback, una escritura basada en datos antiguos puede restaurar una consigna previa. Esta es una condición potencial derivada del diseño, no una pérdida medida durante los ensayos.

4.11 Construcción y operación del dashboard

El panel se divide en una zona de almacén y otra térmica. La primera presenta celdas, conteo de posiciones y porcentaje; la segunda muestra PV, banda, tendencia y consigna. El color verde indica pertenencia al conjunto registrado. La etiqueta visual no establece una segunda verificación contra la escena.



Figura 4.6. Dashboard local con matriz de 54 posiciones, monitor de temperatura y ajuste de consigna.
Fuente: captura DASHBOARD LOCAL.png [P7].

El callback principal devuelve once salidas visuales o de estado. Primero actualiza posiciones y conteos; luego ejecuta la planta térmica, agrega muestras y construye las curvas. Si una excepción interrumpe la función, la actualización puede no completarse, aunque el navegador mantenga la última imagen recibida. Por eso debe distinguirse una pantalla visible de una muestra reciente.

La captura muestra 0 posiciones registradas, 54 libres y 0,0 % de ocupación; en la zona térmica aparecen $PV = -2,9\text{ °C}$ y $SP = 3,0\text{ °C}$. La diferencia absoluta es $5,9\text{ °C}$, superior a la banda de $\pm 2,0\text{ °C}$, por lo que el indicador «Fuera de rango» es coherente con los valores mostrados. El eje horizontal representa ciclos. Esta imagen documenta la interfaz y su condición visible; no determina la causa de la desviación ni prueba por sí sola el desempeño del PID [P7].

4.12 Incidencias documentadas y tratamiento

La guía menciona fallas de conexión por nombre de instancia, bloques optimizados, permisos y direcciones fuera de rango. Se conservan como antecedentes relatados. Para afirmar una solución verificada en la memoria final se necesita registrar condición inicial, mensaje, cambio aplicado y repetición exitosa del ensayo.

Tabla 4.3. Incidencias y comprobación propuesta.

Incidencia	Fuente	Comprobación necesaria
Instancia no encontrada	Guía P3	Comparar nombre esperado y nombre activo.
Address out of range	Guía P3	Revisar número, tamaño y offset del DB descargado.
Acceso externo rechazado	Guía P3	Revisar permisos y configuración del bloque.
CLI : Job pending	Video P6	Registrar operaciones concurrentes y serializar ensayos.
Estado visual desactualizado	Riesgo derivado de P4	Añadir marca de tiempo y simular pérdida de lectura.

El diagnóstico debe avanzar desde el enlace hasta el significado del dato. Primero se comprueba si existe conexión; después si la lectura es permitida; luego si el tamaño y los tipos son correctos; finalmente si la variable representa el hecho deseado. Cambiar simultáneamente todas las configuraciones dificultaría identificar la causa y reproducir la solución.

Las incidencias forman parte del desarrollo porque muestran restricciones reales de integración. Su presentación técnica debe ser precisa: un problema resuelto y documentado aporta evidencia; un riesgo detectado en el código orienta mejoras; un síntoma en el video requiere diagnóstico. No se agrupan todos como fallas definitivas del sistema.

5. EVIDENCIAS, PRUEBAS Y DISCUSIÓN DE RESULTADOS

5.1 Criterio de evaluación

La evaluación distingue cuatro clases de afirmaciones. Una observación visual se apoya en una captura o segmento de video. Un resultado de inspección estática se obtiene del código. Un resultado analítico se deriva de una ecuación bajo hipótesis declaradas. Un resultado experimental requiere un ensayo ejecutado y registrado. Esta clasificación evita presentar una prueba propuesta como si ya estuviera realizada.

La grabación recibida permite observar alternancia entre TIA Portal, Factory I/O y el dashboard. No contiene un registro sincronizado exportado de todas las variables. Además, el gráfico del navegador incorpora una historia anterior al intervalo grabado. Sus ciclos no deben identificarse con los segundos transcurridos del archivo de video.

Los tiempos indicados en esta sección se refieren al inicio de la grabación, no al inicio del proceso ni al origen de las curvas. La revisión utiliza fotogramas con marcas temporales conocidas y comparación de escenas. No se utiliza el audio para atribuir resultados, instrucciones o declaraciones del autor.

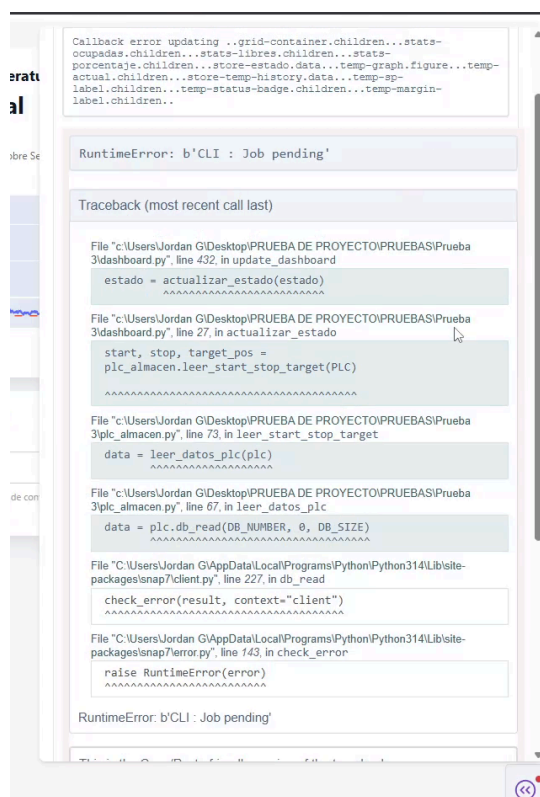
5.2 Registro de observaciones visuales

Tabla 5.1. Observaciones de la grabación P6.

Tiempo del video	Observación	Alcance de la evidencia
15 s	TIA Portal muestra una llamada PID conectada.	Presencia de integración visual; parámetros completos pendientes.
35–45 s	Transelevador y cargas visibles en Factory I/O.	Operación mecánica simulada en el segmento.
75 s	Dashboard: PV $-0,9\text{ }^{\circ}\text{C}$; SP indicado $-1,0\text{ }^{\circ}\text{C}$.	Muestra puntual cercana a la consigna.
85 s	PV $-0,3\text{ }^{\circ}\text{C}$; SP $5,0\text{ }^{\circ}\text{C}$; 26 posiciones, 48,1 %.	Visualización tras cambio de consigna; estado transitorio.
105 s	27 posiciones, 50,0 %; PV $0,8\text{ }^{\circ}\text{C}$; error CLI.	Actualización observada y anomalía de comunicación visible.

La diferencia entre las capturas de 75 y 85 segundos permite reconocer un cambio de consigna mostrada. La PV aumenta posteriormente en el segmento. Esto es consistente con la tendencia del modelo a volver al ambiente cuando disminuye la acción de enfriamiento. No demuestra la forma completa de la respuesta ni que la consigna final haya sido alcanzada durante la grabación.

El cambio de 26 a 27 posiciones produce, aritméticamente, el paso de 48,1 % a 50,0 %, coherente con una capacidad visual de 54. Esta concordancia comprueba el cálculo mostrado, pero no verifica que las 27 ubicaciones representen depósitos confirmados. El algoritmo analizado explica cómo un nuevo destino puede incrementar el conteo.



```

Callback error updating ..grid-container.children...stats-
ocupadas.children...stats-libres.children...stats-
porcentaje.children...store-estado.data...temp-graph.figure...temp-
actual.children...store-temp-history.data...temp-sp-
label.children...temp-status-badge.children...temp-margin-
label.children...

RuntimeError: b'CLI : Job pending'

Traceback (most recent call last)

File "c:\Users\Jordan G\Desktop\PRUEBA DE PROYECTO\PRUEBAS\Prueba
3\dashboard.py", line 432, in update_dashboard
    estado = actualizar_estado(estado)
    ~~~~~^~~~~~

File "c:\Users\Jordan G\Desktop\PRUEBA DE PROYECTO\PRUEBAS\Prueba
3\dashboard.py", line 27, in actualizar_estado
    start, stop, target_pos =
    plc_almacen.leer_start_stop_target(PLC)
    ~~~~~^~~~~~

File "c:\Users\Jordan G\Desktop\PRUEBA DE PROYECTO\PRUEBAS\Prueba
3\plc_almacen.py", line 73, in leer_start_stop_target
    data = leer_datos_plc(plc)
    ~~~~~^~~~~~

File "c:\Users\Jordan G\Desktop\PRUEBA DE PROYECTO\PRUEBAS\Prueba
3\plc_almacen.py", line 67, in leer_datos_plc
    data = plc.db_read(DB_NUMBER, 0, DB_SIZE)
    ~~~~~^~~~~~

File "C:\Users\Jordan G\AppData\Local\Programs\Python\Python314\Lib\site-
packages\snap7\client.py", line 227, in db_read
    check_error(result, context="client")
    ~~~~~^~~~~~

File "C:\Users\Jordan G\AppData\Local\Programs\Python\Python314\Lib\site-
packages\snap7\error.py", line 143, in check_error
    raise RuntimeError(error)
    ~~~~~^~~~~~

RuntimeError: b'CLI : Job pending'

```

Figura 5.1. Mensaje «CLI : Job pending» visible en t = 105 s de P6. Se conserva como evidencia de una anomalía observada; su causa específica requiere registro de llamadas.

El error aparece mientras la interfaz conserva valores previos o recientes. Esto demuestra la necesidad de una señal explícita de frescura. El rótulo fijo de conexión no permite establecer si el último dato es válido. Una implementación de diagnóstico debe informar el instante de la última lectura exitosa y el estado del enlace, sin borrar automáticamente la información útil para analizar la falla.

5.3 Resultados de inspección estática

La inspección confirma una separación modular entre interfaz y acceso/modelo. El dashboard importa `plc_almacen` y utiliza funciones de lectura y escritura. La configuración está concentrada en constantes al inicio del módulo de acceso. Esta organización facilita identificar el contrato con el PLC y ubicar las ecuaciones de la planta.

Se confirma también que el número de posiciones es 54, la cuadrícula tiene nueve columnas, el periodo configurado es 1 s y la banda térmica es ± 2 °C. Esas propiedades son reproducibles a partir del texto del programa. Su presencia no demuestra exactitud temporal real ni pertinencia del margen para un producto alimentario.

Los principales límites identificados son la ocupación inferida desde el destino, la ausencia de eliminación de posiciones, el estado de sesión en el navegador, la escritura completa de DB11, la falta de límites de consigna y salida en Python, la adquisición asociada al callback y el indicador fijo de conexión. La relevancia de cada punto depende del uso previsto: algunos son aceptables en una demostración breve, pero deben tratarse antes de usar el panel como fuente operacional de inventario.

Tabla 5.2. Hallazgos del código y efecto sobre la evaluación.

Hallazgo	Consecuencia técnica	Prioridad propuesta
Destino agregado sin confirmación	No equivale a inventario confirmado.	Alta.
DB11 reescrito completo	Posible interferencia entre escritores.	Alta.
Cliente y simulador globales	Requiere controlar acceso y sesiones.	Alta.
Sin marca temporal de muestra	Dificulta medir y detectar datos antiguos.	Alta.
Sin recorte de historial	Crecimiento de datos durante la sesión.	Media.
Sin límites de SP en el callback	Admite consignas fuera del rango del modelo.	Media.

Las prioridades se proponen según su impacto en la interpretación y coherencia del banco, no a partir de un análisis formal de riesgo industrial. Su utilidad es ordenar la siguiente revisión:

primero asegurar el significado y la integridad de los datos; después ampliar funciones y mejorar la presentación.

5.4 Resultados analíticos del modelo térmico

El análisis del capítulo 4 determina que el equilibrio de la planta depende de la salida de frío. Con ruido nulo y salida constante, el modelo admite una temperatura de 8 °C sin enfriamiento y aproximadamente -3,974 °C a 100 %. Estas cifras son resultados calculados del programa recibido, no medidas de una cámara real.

La tabla de equilibrios ayuda a identificar consignas adecuadas para pruebas. Una consigna de 5 °C requiere una acción de equilibrio pequeña, mientras que una de -1 °C requiere una mayor fracción de enfriamiento. El cambio observado en el video entre esas consignas es compatible con una variación de la demanda de frío, aunque la salida no se registra de forma continua en el material disponible.

El análisis con salida fija muestra que la planta discreta tiene un coeficiente entre 0,62 y 0,97 para una salida normalizada entre cero y uno. Las capturas del PID confirman una escala de frío de 0 % a 100 %. Este resultado caracteriza la planta bajo salida constante; la evaluación del lazo cerrado requiere los ensayos de respuesta previstos en el capítulo 5 [P8].

La presencia de ruido introduce variabilidad entre ejecuciones. El programa no fija una semilla aleatoria, por lo que no se espera reproducir exactamente cada punto de una curva. Para comparar configuraciones conviene registrar la semilla o ejecutar varias repeticiones, además de distinguir el comportamiento determinista del efecto de perturbaciones.

5.5 Diseño de ensayos reproducibles

Cada ensayo debe registrar objetivo, versión, condición inicial, secuencia aplicada, variables observadas y criterio de aceptación. Se recomienda conservar un archivo de datos y una captura representativa. El criterio se define antes de ejecutar la prueba, de manera que el análisis no se adapte retrospectivamente al resultado observado.

Para el banco virtual se propone comenzar con una única sesión de navegador y un único proceso de adquisición. Debe registrarse si la depuración está activa y si existe alguna lectura

adicional desde otro programa. Esta condición permite reducir ambigüedades al analizar errores de comunicación. Después pueden agregarse pruebas específicas de concurrencia.

Los registros temporales deberían incluir SP, PV, OUT, destino, estado de marcha y resultado de cada operación S7. Para ocupación confirmada se necesita una variable adicional de finalización o un vector del PLC. Mientras esa variable no exista, el ensayo debe registrar separadamente destino solicitado y depósito observado.

Tabla 5.3. Plan mínimo de ensayos; ejecución pendiente de registro.

Código	Ensayo	Criterio propuesto
C01	Lectura repetida de DB1 y DB11	Sin errores en 300 lecturas; valores concordantes.
C02	Cambio de SP y lectura posterior	Coincidencia con tolerancia numérica acordada.
A01	Depósito en posición inicial	Carga depositada y estado confirmado coherente.
A02	Destino repetido	Sin duplicación y con rechazo o tratamiento definido.
A03	Destino fuera de rango	Orden rechazada; sin movimiento indebido.
A04	Parada durante maniobra	Comportamiento documentado y recuperación controlada.
A05	Retiro de carga	Liberación confirmada; pendiente si no está implementado.
T01	Escalón de consigna	Serie SP/PV/OUT y métricas calculables.
T02	Consigna no alcanzable	Saturación reconocida; sin interpretación falsa de éxito.
D01	Recarga del navegador	Estado comparado antes y después del reinicio.
D02	Desconexión del PLC	Error visible y muestra antigua identificada.
D03	Dos sesiones simultáneas	Frecuencia y errores registrados; sin duplicar simulación.

El número de 300 lecturas en C01 es una duración propuesta para una primera prueba breve, no un umbral industrial de disponibilidad. La tolerancia de escritura debe definirse considerando representación REAL y visualización. Los ensayos de seguridad física quedan fuera del banco y requieren una evaluación específica si se implementa hardware.

5.6 Protocolo de comunicación

C01 comienza con el PLC en estado operativo y las dos estructuras descargadas. Se registra la configuración de conexión y se realizan lecturas periódicas, conservando hora, duración y resultado. En varios instantes se comparan los valores con una tabla online. Si aparece una excepción, se conserva el mensaje completo y la operación que la produjo.

C02 utiliza una consigna dentro del rango de análisis, con la planta en una condición conocida. Se registra el valor anterior, se envía la nueva consigna y se realiza una lectura posterior. La aceptación debe basarse en el valor utilizado por el PLC, no únicamente en el mensaje de la interfaz. También se observa si la escritura altera indebidamente PV u OUT.

Para analizar «Job pending» se propone repetir primero con acceso serial y una única sesión. Después se introduce una segunda fuente de solicitudes, registrando el orden de llamadas. Una reducción del error tras serializar constituye evidencia útil, pero la atribución causal necesita condiciones comparables. No corresponde declarar solucionada la anomalía antes de esa comprobación.

5.7 Protocolo de almacenamiento

A01 debe partir con escena y estado lógico reconciliados. Se selecciona una posición identificable, se registra la aceptación de la orden y se observa la maniobra hasta su finalización. El ensayo conserva destino, señales de llegada y estado confirmado. Si el panel marca la posición antes del depósito, se registra la diferencia temporal.

Las posiciones de prueba deben cubrir extremos e interior de la malla, por ejemplo P01, P09, P10 y P54, siempre que el controlador permita seleccionarlas. Estos casos ayudan a comprobar cambios de fila y correspondencia entre panel y escena. La selección propuesta no demuestra cobertura exhaustiva de las 54 ubicaciones.

A02 y A03 verifican tratamiento de solicitudes repetidas e inválidas. El código del dashboard ignora valores fuera de rango para su acumulación, pero eso no prueba que el PLC rechace un destino mecánicamente inválido. El ensayo debe observar por separado el controlador, el simulador y la interfaz.

A04 analiza parada y recuperación. Debe identificarse si la secuencia se pausa, se cancela o requiere reset, y qué ocurre con una carga en tránsito. A05 corresponde a retiro. Si no existe

esa función en el programa actual, se registra como no implementada y se retira de la declaración de capacidades de la versión evaluada.

5.8 Protocolo de respuesta térmica

T01 debe definir PV inicial, SP inicial, nueva SP, parámetros del modelo, configuración del PID y tiempo de adquisición. Se propone conservar un tramo previo al cambio y continuar hasta que la respuesta pueda caracterizarse. Deben registrarse al menos PV, SP y OUT con marcas de tiempo reales y sin reconstruir datos desde píxeles del gráfico.

La prueba se puede repetir con dos consignas alcanzables según el modelo, como 5 °C y -1 °C. Se deben considerar tanto disminución como aumento de consigna: el modelo solo dispone de enfriamiento activo y retorno ambiental, por lo que ambas respuestas no tienen por qué ser simétricas. La duración no se fija a partir del video, ya que su historia comienza antes de la grabación.

T02 utiliza una consigna deliberadamente fuera del rango determinista, únicamente dentro del entorno simulado. Su propósito es reconocer la saturación y explicar por qué una consigna no alcanzable no puede corregirse solo aumentando la acción integral. La prueba debe registrar límites, estado del PID y mensajes del panel. No se traslada este ensayo a un proceso alimentario real.

Para mejorar la comparabilidad se recomienda repetir los ensayos con condiciones iniciales equivalentes y conservar la semilla del ruido cuando se instrumente esa opción. Si se modifica el modelo, la nueva revisión debe identificarse y sus resultados mantenerse separados de los de esta versión.

5.9 Indicadores de desempeño

El error instantáneo se define como $e(k) = SP(k) - PV(k)$. Para una ventana de N muestras se puede calcular el error absoluto medio y la raíz del error cuadrático medio. Estos indicadores resumen la desviación, pero deben acompañarse de la ventana utilizada y de una distinción entre transitorio y régimen.

$$MAE = (1/N) \cdot \sum |e(k)|; RMSE = \sqrt{[(1/N) \cdot \sum e(k)^2]} \quad (5.1)$$

El tiempo de establecimiento requiere una banda previamente acordada y un criterio de permanencia. La banda ± 2 °C del dashboard puede utilizarse como una referencia de interfaz, pero no debe asumirse como exigencia de control o conservación. Para consignas próximas a cero, una banda absoluta evita ambigüedades de porcentajes referidos a la consigna.

El sobreimpulso debe definirse según el sentido del escalón. En un ensayo de enfriamiento puede interesar el descenso por debajo de la consigna; en un aumento, la elevación por encima de ella. La amplitud máxima de error y la duración de saturación de OUT complementan el análisis sin depender de una interpretación única de sobreimpulso porcentual.

Para comunicación se propone calcular tasa de operaciones exitosas y distribución de duración de llamadas. Para el almacén, tiempo de maniobra, órdenes completadas y discrepancias de inventario. Esos valores deben relacionarse con el número de ensayos; una sola maniobra exitosa no permite estimar confiabilidad general.

Tabla 5.4. Resultados cuantitativos por completar con registros.

Indicador	Unidad	Resultado de esta versión
MAE en régimen	°C	[PENDIENTE: serie temporal y ventana].
RMSE en régimen	°C	[PENDIENTE: serie temporal y ventana].
Tiempo de establecimiento	s	[PENDIENTE: banda y ensayo].
Máxima desviación transitoria	°C	[PENDIENTE: registro completo].
Duración de maniobra	s	[PENDIENTE: inicio y confirmación].
Operaciones S7 exitosas	%	[PENDIENTE: total y errores].
Discrepancias de ocupación	número	[PENDIENTE: comparación contra estado confirmado].

5.10 Evaluación de objetivos

Tabla 5.5. Estado de los objetivos según la evidencia disponible.

Objetivo	Evaluación documental	Pendiente principal
OE1 Arquitectura	Componentes y CPU identificados.	Registro de versiones del entorno.
OE2 Control del almacén	Mapeo, llamada y seis redes analizadas.	Ensayos de secuencia y correspondencia de destinos.
OE3 Posiciones	Matriz y algoritmo caracterizados.	Confirmación de depósitos y retiros.
OE4 Modelo térmico	Modelo, conexiones y escalas del PID documentados.	Ensayos de respuesta térmica.
OE5 Python y dashboard	Código revisado y servicio local observado.	Robustez de comunicación y persistencia.
OE6 Evaluación	Evidencias y protocolos documentados.	Ejecutar y registrar pruebas cuantitativas.
OE7 Limitaciones y mejoras	Identificadas y priorizadas.	Implementar una nueva revisión y validarla.

La evaluación permite reconocer un prototipo integrado con evidencia de funcionamiento parcial y una implementación Python analizable. No permite declarar validación integral de todos los requisitos. Esta formulación es más útil para la revisión académica porque indica qué evidencia respalda cada objetivo y qué actividad concreta cierra la brecha.

5.11 Discusión técnica

El resultado más sólido es la existencia de una arquitectura concreta que relaciona PLC, simulación mecánica y supervisión térmica. El código define direcciones y una planta matemática; las capturas muestran variables y paneles; el video registra operación y cambios. El valor del trabajo se encuentra en esa integración y en la posibilidad de analizarla con criterios de ingeniería.

La principal limitación funcional está en el significado de ocupación. Una interfaz puede calcular correctamente un porcentaje y aun así representar una magnitud distinta del

inventario físico. La revisión identifica esta diferencia de manera verificable y propone una solución basada en confirmaciones. Esta observación fortalece la tesis al convertir una limitación en un requisito de diseño explícito.

La principal limitación temporal es el acoplamiento entre callback y simulación. El periodo nominal de un segundo no basta para asegurar una dinámica reproducible si existen solicitudes concurrentes o errores. Separar el reloj de planta del navegador es una mejora estructural que permitiría evaluar con mayor rigor el PID y comparar ensayos.

5.12 Proyección industrial y evaluación económica preliminar

Una eventual aplicación física requeriría especificaciones del almacén, características de cargas, dimensionamiento mecánico, accionamientos, sensores, seguridad, red e instrumentación térmica. La migración no consiste únicamente en reemplazar el PLC simulado por uno real. Las hipótesis del modelo tendrían que contrastarse con mediciones y la supervisión debería integrarse con procedimientos operacionales.

En el contexto alimentario se necesitarían producto, rango térmico, tiempos admisibles, trazabilidad y responsabilidades. La temperatura única del modelo no permite inferir la temperatura del producto ni la uniformidad de una cámara. El diseño frigorífico debe realizarse sobre un balance de cargas y condiciones reales, no sobre los coeficientes de demostración.

El análisis económico se deja como una estructura de levantamiento. Deben separarse inversión inicial, licencias, ingeniería, instalación, mantenimiento y operación. Los beneficios solo pueden cuantificarse respecto de una situación base medible. Sin cotizaciones, consumos ni datos de productividad, no se calcula retorno de inversión ni periodo de recuperación.

CONCLUSIONES

El proyecto integra un almacén virtual con transelevador, un PLC Siemens simulado y un dashboard local en Python. Las fuentes muestran movimiento, intercambio térmico y visualización de 54 posiciones. El aporte consiste en articular control, simulación y supervisión sin una planta física.

El código y las capturas permiten establecer el contrato de DB1 y DB11, relacionar DB1.DBW4 con %QW30 y verificar las conexiones térmicas. Las seis redes de FB1 documentan una secuencia de almacenamiento con enclavamientos, temporizadores y contador. Los ajustes del PID acreditan enfriamiento habilitado y escalado porcentual. La evaluación dinámica se desarrolla mediante los ensayos propuestos en el capítulo 5.

La ocupación mostrada acumula destinos observados y no constituye, por sí sola, inventario confirmado. Para completar esa función se necesitan eventos de depósito y retiro o un vector de ocupación mantenido por el controlador. También se identificaron prioridades de mejora en la escritura compartida, la serialización del acceso S7 y la independencia del reloj térmico respecto del navegador.

El análisis matemático caracteriza la planta térmica bajo salida constante; MIN_TEMP no representa una temperatura mínima garantizada. La escala de frío de 0 % a 100 % y los límites de proceso de -5°C a 10°C están documentados en el regulador. El alcance de estos resultados corresponde a la integración simulada descrita; los protocolos del capítulo 5 permiten ampliar su validación dinámica.

En consecuencia, respalda una demostración funcional parcial y no una operación libre de fallas. Los protocolos propuestos permiten completar la evaluación mediante condiciones iniciales conocidas, muestras sincronizadas y criterios definidos antes del ensayo.

El siguiente paso consiste en consolidar el respaldo PLC, corregir las limitaciones prioritarias y ejecutar las pruebas antes de evaluar una migración a hardware. No se atribuyen ahorros, certificaciones ni desempeño industrial no medidos.

REFERENCIAS

Las fuentes del proyecto se identifican con P; las universitarias, con U; y las externas, con R. La documentación web fue consultada para esta revisión en septiembre de 2026. Las páginas de producto describen funciones y no certifican la configuración particular del prototipo.

- [P1] GALLARDO DOMÍNGUEZ, Jordan Mauricio. Trabajo de Título (1). Documento de trabajo [PDF]. Sin fecha de versión explícita. Material aportado por el autor; portada e introducción utilizadas como antecedente.
- [P2] GALLARDO DOMÍNGUEZ, Jordan Mauricio. Presentación Final de Título. Examen de título, 2025 [PDF, 16 diapositivas]. Material del proyecto. Se utilizan especialmente las diapositivas 7–11.
- [P3] Informe guía para retomar la escritura de tesis [PDF, 13 páginas]. Autoría y fecha no indicadas. Documento orientador aportado por el autor; no se utiliza como bibliografía científica independiente.
- [P4] GALLARDO DOMÍNGUEZ, Jordan Mauricio [autoría del archivo por confirmar]. dashboard.py [código Python]. PRUEBAS/Prueba 3; copia equivalente: import dash.txt. Archivo recibido para revisión; 549 líneas. Funciones actualizar_estado, update_dashboard y guardar_setpoint.
- [P5] GALLARDO DOMÍNGUEZ, Jordan Mauricio [autoría del archivo por confirmar]. plc_almacen.py [código Python]. PRUEBAS/Prueba 3; copia equivalente: import random.txt. Archivo recibido para revisión; 166 líneas. Comunicación y clase SimuladorTemperatura.
- [P6] GALLARDO DOMÍNGUEZ, Jordan Mauricio. 2025-12-20 09-07-25.mp4 [videgrabación de pantalla]. Fecha identificada por nombre y pantalla: 20 diciembre 2025. Duración aproximada: 113,6 s; resolución: 1920 × 1080. Material aportado por el autor.
- [P7] GALLARDO DOMÍNGUEZ, Jordan Mauricio. CAPTURAS PROGRAMAS [capturas de pantalla]. Proyecto CAJA SLOCAS_V19. Archivos consultados: TIA PORTAL — MAIN.png, MAIN2.png, DB UBICACION CAJAS.png, PLC Tags .png, Temperatura_Dash_DB.png, BLOQUES.png y PLCSIMAdvanced.png; FACTORY IO — Panel.png, CONFIGURACION .png y DRIVERS .png; CODIGO — DASHBOARD LOCAL.png. Material aportado por el autor, consultado en septiembre de 2026. Las imágenes no se asumen sincronizadas entre sí ni con la grabación P6.
- [P8] GALLARDO DOMÍNGUEZ, Jordan Mauricio. Redes de FB1 y configuración de control [capturas de pantalla]. Archivos: Network 1 FB1.png a Network 6 FB1.png; FB ENTRADA SALIDAS.png; PID_TEMP_BASIC.png; PID_TEMP_output.png; PID_TEMP_PROCEESS.png; PUT_GET.png; Bloques Optimizados.png. Material adjuntado por el autor para esta revisión. Se interpreta el contenido visible, sin asumir ejecución online ni simultaneidad entre las capturas. Reproducciones completas en el anexo C.
- [U1] UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA. Comité de Coordinación y Desarrollo Docente. Normas generales para la presentación de la memoria de titulación. 1 abril 2008 [PDF]. Archivo aportado: 061_Memorias-NormasGenerales.pdf.
- [U2] UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA. Citas bibliográficas [PDF]. Sin fecha confirmada. Archivo aportado: 061_Citas-Bibliográficas.pdf. Orientación de presentación de referencias.
- [U3] UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA. Constancia de validación y confidencialidad de monografía a repositorio académico [formulario PDF]. Sin fecha confirmada. Archivo oficial aportado; exige inserción como página 2 y firmas antes de entrega.
- [R1] ROODBERGEN, Kees Jan; VIS, Iris F. A. A survey of literature on automated storage and retrieval systems. European Journal of Operational Research, 2009, 194(2), 343–362. DOI:

- 10.1016/j.ejor.2008.01.038. Registro y resumen institucional consultados:
<https://pure.eur.nl/en/publications/a-survey-of-literature-on-automated-storage-and-retrieval-systems/>
- [R2] LEE, Chi G.; PARK, Sang C. Survey on the virtual commissioning of manufacturing systems. *Journal of Computational Design and Engineering*, 2014, 1(3), 213–222. DOI: 10.7315/JCDE.2014.021.
Registro editorial y resumen consultados:
<https://www.techno-press.org/content/?journal=cde&num=3&ordernum=7&page=article&volume=1>
- [R3] REAL GAMES. Automated Warehouse [documentación en línea]. Factory I/O.
<https://docs.factoryio.com/manual/scenes/automated-warehouse/>
- [R4] REAL GAMES. Siemens S7-PLCSIM [documentación en línea]. Factory I/O.
<https://docs.factoryio.com/manual/drivers/s7-plcsim/>
- [R5] REAL GAMES. Setting up S7-PLCSIM Advanced V4.0–8.0 [tutorial en línea]. Factory I/O.
<https://docs.factoryio.com/tutorials/siemens/setting-up-s7-plcsim-advanced-V4/>
- [R6] SIEMENS. Heating and cooling output value (S7-1200, S7-1500) [documentación en línea, TIA V20]. Referente funcional; verificar diferencias con V19 del proyecto.
https://docs.tia.siemens.cloud/r/en-us/v20/using-pid_temp-s7-1200-s7-1500/configuring-pid_temp-s7-1200-s7-1500/basic-settings-s7-1200-s7-1500/heating-and-cooling-output-value-s7-1200-s7-1500
- [R7] PYTHON-SNAP7. Util [documentación en línea, versión 2.1.1]. Conversiones de datos. No identifica la versión instalada del proyecto. <https://python-snap7.readthedocs.io/en/2.1.1/API/util.html>
- [R8] PLOTLY. Adding Interactivity with Callbacks [documentación en línea]. Dash.
<https://dash.plotly.com/basic-callbacks>
- [R9] PLOTLY. Live Updating Components [documentación en línea]. Dash.
<https://dash.plotly.com/live-updates>
- [R10] PLOTLY. dcc.Store [documentación en línea]. Dash.
<https://dash.plotly.com/dash-core-components/store>
- [R11] PLOTLY. Sharing Data Between Callbacks [documentación en línea]. Dash.
<https://dash.plotly.com/sharing-data-between-callbacks>
- [R12] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. ISO 12100:2010. Safety of machinery — General principles for design — Risk assessment and risk reduction. Descripción oficial consultada; no se realizó auditoría del texto normativo completo.
<https://www.iso.org/standard/51528.html>
- [R13] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. ISO 22000:2018. Food safety management systems — Requirements for any organization in the food chain. Descripción oficial consultada; no se realizó evaluación de conformidad. <https://www.iso.org/standard/65464.html>

ANEXO A. MÓDULO DE COMUNICACIÓN Y MODELO TÉRMICO

Transcripción del módulo plc_almacen.py aportado, equivalente a import random.txt. Se conserva el código, incluidos comentarios y decisiones de implementación. La numeración de línea y el ajuste visual de renglones son editoriales. No se presenta como una versión corregida.

```
001 import random
002 import snap7
003 from snap7.util import get_bool, get_int, set_bool, set_int, get_real, set_real
004
005 # =====
006 # Configuración PLC general
007 # =====
008
009 PLC_IP = '192.168.0.1'
010 RACK = 0
011 SLOT = 1
012
013 # =====
014 # DB del almacén (54 posiciones)
015 # =====
016
017 DB_NUMBER = 1      # DB UBICACION_DE_CAJAS
018 DB_SIZE = 20       # Tamaño del DB a leer en bytes
019
020 # START -> DB1.DBX0.0
021 # STOP -> DB1.DBX0.1
022 # TARGET -> DB1.DBW4 (INT)
023 OFFSET_START_BIT = (0, 0) # (byte, bit)
024 OFFSET_STOP_BIT = (0, 1)
025 OFFSET_TARGET_INT = 4      # byte de inicio del INT
026
027
028 # =====
029 # DB de temperatura (PID_Temp)
030 # =====
031
032 # OJO: esto asume que tu DB "Temperatura_Dash" es DB11 y:
033 # TEMP_SP -> %DB11.DBD0
034 # TEMP_PV -> %DB11.DBD4
035 # PID_OUT -> %DB11.DBD8 (conectado a OutputCool del PID_Temp)
036 DB_TEMP_NUMBER = 11
037 DB_TEMP_SIZE = 12
038
039 OFFSET_TEMP_SP = 0 # REAL
040 OFFSET_TEMP_PV = 4 # REAL
041 OFFSET_TEMP_OUT = 8 # REAL (OutputCool / PID_OUT)
042
043 # Parámetros del modelo de enfriamiento (simulación de planta)
044 AMBIENT_TEMP = 8.0 # °C ambiente
045 MIN_TEMP = -5.0 # °C mínimo alcanzable
046 COOLING_GAIN = 0.35 # qué tan fuerte enfría por % de OutputCool
047 RETURN_GAIN = 0.03 # tendencia a volver hacia AMBIENT_TEMP
048 NOISE_AMPL = 0.08 # ruido ±0.08°C
049 DT_SECONDS = 1.0 # intervalo de simulación (1 s, igual que el Interval del
    dashboard)
050
051
052 # =====
053 # Funciones básicas PLC
054 # =====
055
056 def crear_cliente_plc():
057     plc = snap7.client.Client()
058     plc.connect(PLC_IP, RACK, SLOT)
059     print('Conectado al PLC')
060     return plc
061
062
063 # ----- Almacén -----
064
065 def leer_datos_plc(plc):
066     """Lee el DB del PLC para el almacén."""
067     data = plc.db_read(DB_NUMBER, 0, DB_SIZE)
068     return data
069
070
071 def leer_start_stop_target(plc):
072     """Extrae START, STOP y TARGET_POSITION del DB del almacén."""
073     data = leer_datos_plc(plc)
074     start = get_bool(data, OFFSET_START_BIT[0], OFFSET_START_BIT[1])
075     stop = get_bool(data, OFFSET_STOP_BIT[0], OFFSET_STOP_BIT[1])
```

```

076 target_position = get_int(data, OFFSET_TARGET_INT)
077 return start, stop, target_position
078
079
080 # ----- Temperatura PID_Temp -----
081
082 def leer_db_temperatura(plc):
083     """Lee el DB de temperatura (Temperatura_Dash)."""
084     data = plc.db_read(DB_TEMP_NUMBER, 0, DB_TEMP_SIZE)
085     return data
086
087
088 def leer_sp_pv_out(plc):
089     """
090     Lee:
091     - SP (setpoint)
092     - PV (temperatura medida)
093     - OUT (PID_OUT, lo usamos como OutputCool %)
094     """
095     data = leer_db_temperatura(plc)
096     sp = get_real(data, OFFSET_TEMP_SP)
097     pv = get_real(data, OFFSET_TEMP_PV)
098     out = get_real(data, OFFSET_TEMP_OUT)
099     return sp, pv, out, data
100
101
102 def escribir_temp_pv(plc, nueva_pv, data=None):
103     """Escribe la temperatura PV simulada en el DB de temperatura."""
104     if data is None:
105         data = leer_db_temperatura(plc)
106     set_real(data, OFFSET_TEMP_PV, float(nueva_pv))
107     plc.db_write(DB_TEMP_NUMBER, 0, data)
108
109
110 def escribir_temp_sp(plc, nuevo_sp):
111     """Escribe el setpoint de temperatura en el DB de temperatura."""
112     data = leer_db_temperatura(plc)
113     set_real(data, OFFSET_TEMP_SP, float(nuevo_sp))
114     plc.db_write(DB_TEMP_NUMBER, 0, data)
115
116
117 # =====
118 # Simulador de Temperatura (enfriamiento)
119 # =====
120
121 class SimuladorTemperatura:
122     """
123     Simula una cámara fría controlada por OutputCool (PID_OUT) del bloque PID_Temp.
124     - Usa OutputCool (%) para bajar la temperatura hacia MIN_TEMP.
125     - Si OutputCool es bajo, la temperatura tiende a subir hacia AMBIENT_TEMP.
126     """
127
128     def __init__(self, plc):
129         self.plc = plc
130         # Inicializar PV con el valor actual o AMBIENT_TEMP si algo falla
131         try:
132             pv, _, _ = leer_sp_pv_out(self.plc)
133             if pv != pv: # NaN
134                 pv = AMBIENT_TEMP
135             self.temp_pv = pv
136         except Exception:
137             self.temp_pv = AMBIENT_TEMP
138
139     def modelo_enfriamiento(self, temp_actual, outcool_percent):
140         # Enfriamiento proporcional a la distancia a MIN_TEMP
141         cooling = COOLING_GAIN * (outcool_percent / 100.0) * max(temp_actual -
142             MIN_TEMP, 0.0)
143         # Calentamiento por ambiente si está por debajo de AMBIENT_TEMP
144         warming = RETURN_GAIN * (AMBIENT_TEMP - temp_actual)
145         # Ruido pequeño
146         noise = (random.random() - 0.5) * NOISE_AMPL * 2.0
147         dT = (-cooling + warming) * DT_SECONDS + noise
148         return temp_actual + dT
149
150     def paso(self):
151         """
152         Un paso de simulación:
153         1) Lee SP, PV (actual) y OUT (OutputCool %) desde el PLC.
154         2) Actualiza la temperatura simulada.
155         3) Escribe la nueva PV al PLC.
156         Devuelve (sp, pv_simulada, outcool).
157         """
158         sp, pv_actual, outcool, data = leer_sp_pv_out(self.plc)
159
160         # Actualiza modelo a partir del último valor simulado (no el PV que viene del
161         # PLC)
162         self.temp_pv = self.modelo_enfriamiento(self.temp_pv, outcool)
163         # Escribimos la nueva PV simulada al DB

```

```
164     escribir_temp_pv(self.plc, self.temp_pv, data)
165
166     return sp, self.temp_pv, outcool
```

ANEXO B. EXTRACTOS FUNCIONALES DEL DASHBOARD

Extractos de dashboard.py / import dash.txt. Se seleccionan las funciones de acumulación, actualización y consigna; se omite la mayor parte de los estilos visuales y las líneas en blanco. Los números corresponden a las líneas del archivo recibido. El código completo permanece en las fuentes del proyecto.

```
001 import dash
002 from dash import Dash, html, dcc, Output, Input, State
003 import plotly.graph_objs as go
005 import plc_almacen
007 NUM_POS = 54
009 # Cliente PLC y simulador de temperatura (enfriamiento)
010 PLC = plc_almacen.crear_cliente_plc()
011 SIM_TEMP = plc_almacen.SimuladorTemperatura(PLC)
014 # =====
015 # Estado del almacén
016 # =====
018 def init_state():
019     return {"occupied": []}
022 def actualizar_estado(estado):
023     """Lee TARGET_POSITION del PLC y marca esa posición como ocupada (anclada)."""
024     if estado is None:
025         estado = init_state()
027     start, stop, target_pos = plc_almacen.leer_start_stop_target(PLC)
029     if 1 <= target_pos <= NUM_POS:
030         ocupadas = set(estado.get("occupied", []))
031         ocupadas.add(target_pos)
032         estado["occupied"] = sorted(ocupadas)
033     return estado
414 @app.callback(
415     Output("grid-container", "children"),
416     Output("stats-ocupadas", "children"),
417     Output("stats-libres", "children"),
418     Output("stats-porcentaje", "children"),
419     Output("store-estado", "data"),
420     Output("temp-graph", "figure"),
421     Output("temp-actual", "children"),
422     Output("store-temp-history", "data"),
423     Output("temp-sp-label", "children"),
424     Output("temp-status-badge", "children"),
425     Output("temp-margin-label", "children"),
426     Input("interval-update", "n_intervals"),
427     State("store-estado", "data"),
428     State("store-temp-history", "data"),
429 )
430 def update_dashboard(n, estado, temp_history):
431     # ----- Almacén -----
432     estado = actualizar_estado(estado)
433     ocupadas = estado["occupied"]
435     total = NUM_POS
436     num_ocupadas = len(ocupadas)
437     libres = total - num_ocupadas
438     porc = (num_ocupadas / total) * 100 if total > 0 else 0.0
440     stats_ocupadas = f"{num_ocupadas}"
441     stats_libres = f"{libres}"
442     stats_porc = f"{porc:0.1f}%"
444     grid = build_grid(ocupadas)
446     # ----- Temperatura -----
447     if temp_history is None:
448         temp_history = {"t": [], "pv": [], "sp": []}
450     # Cierra el loop PID_Temp <-> simulación de frío
451     sp, pv, outcool = SIM_TEMP.paso()
453     temp_history["t"].append(n)
454     temp_history["pv"].append(pv)
```

```

455 temp_history["sp"].append(sp)
457 fig = go.Figure()
458 fig.add_trace(go.Scatter(x=temp_history["t"], y=temp_history["pv"],
459     mode="lines", name="PV", line=dict(width=2)))
460 fig.add_trace(go.Scatter(x=temp_history["t"], y=temp_history["sp"],
461     mode="lines", name="SP", line=dict(width=1.5,
    dash="dash"))))
462 fig.update_layout(
463     margin=dict(l=0, r=0, t=0, b=0),
464     xaxis_title="Tiempo (ciclos)",
465     yaxis_title="Temperatura [°C]",
466     legend=dict(orientation="h", yanchor="bottom", y=1.02, xanchor="right", x=1),
467 )
468 temp_actual_label = f"{pv:0.1f} °C Actual"
469 sp_label = f"Setpoint Actual: {sp:0.1f} °C"
470 # Estado dentro / fuera de rango
471 margen = 2.0
472 diff = abs(pv - sp)
473 dentro = diff <= margen
474 if dentro:
475     badge = html.Span(
476         "✓ Temperatura dentro de rango",
477         style={
478             "display": "inline-flex",
479             "alignItems": "center",
480             "padding": "2px 10px",
481             "borderRadius": "999px",
482             "backgroundColor": "#dcfce7",
483             "color": "#15803d",
484             "fontSize": "12px",
485             "fontWeight": "600",
486         },
487     )
488 else:
489     badge = html.Span(
490         "△ Fuera de rango",
491         style={
492             "display": "inline-flex",
493             "AlignItems": "center",
494             "padding": "2px 10px",
495             "borderRadius": "999px",
496             "backgroundColor": "#fee2e2",
497             "color": "#b91c1c",
498             "fontSize": "12px",
499             "fontWeight": "600",
500         },
501     )
502 margin_label = f"Margen permitido: ±{margen:0.1f}°C sobre Setpoint ({sp:0.1f}°C)."
503 return (
504     grid,
505     stats_ocupadas,
506     stats_libres,
507     stats_porc,
508     estado,
509     fig,
510     temp_actual_label,
511     temp_history,
512     sp_label,
513     badge,
514     margin_label,
515 )
516 # =====
517 # Callback: Guardar Setpoint
518 # =====
519 @app.callback(
520     Output("temp-sp-message", "children"),

```

```

529 Input("temp-sp-save", "n_clicks"),
530 State("temp-sp-input", "value"),
531 prevent_initial_call=True,
532 )
533 def guardar_setpoint(n_clicks, value):
534     if value is None:
535         return "Ingresa un valor de setpoint antes de guardar."
536     try:
537         plc_almacen.escribir_temp_sp(PLC, float(value))
538         return f"Setpoint actualizado a {float(value):0.1f} °C."
539     except Exception as exc:
540         return f"Error al escribir setpoint en PLC: {exc}"
543 # =====
544 # Main
545 # =====
547 if __name__ == "__main__":
548     print("Abre http://127.0.0.1:8050 en tu navegador")
549     app.run(debug=True)

```

ANEXO C. REDES Y CONFIGURACIÓN DEL PROYECTO

Este anexo incorpora las doce capturas recibidas para completar la descripción del control. Las imágenes se conservan sin modificar su contenido. Las descripciones nuevas se resaltan en amarillo; las limitaciones indicadas corresponden a información que no es visible en las capturas [P8].

		UBICACION DE CAJAS							
		Name	Data type	Offset	Default value	Accessible f...	Writa...	Visible in ...	Setpoint
CAJA SLOCAS_V19	1	Input							
Add new device	2	START	Bool	0.0	false	☒	☒	☒	☐
Devices & networks	3	STOP	Bool	0.1	false	☒	☒	☒	☐
PLC_1 [CPU 1511-1 PN]	4	RESET	Bool	0.2	false	☒	☒	☒	☐
Device configuration	5	AT MDEL	Bool	0.3	false	☒	☒	☒	☐
Online & diagnostics	6	AT ENTRY	Bool	0.4	false	☒	☒	☒	☐
Software units	7	AT LOAD	Bool	0.5	false	☒	☒	☒	☐
Program blocks	8	AT RIGHT	Bool	0.6	false	☒	☒	☒	☐
Add new block	9	AT LEFT	Bool	0.7	false	☒	☒	☒	☐
Main [OB1]	10	MOVIN X	Bool	1.0	false	☒	☒	☒	☐
MHU-PLC-Lab-Function-S715...	11	Output							
UBICACION DE CAJAS [FB1]	12	LOAD CONVEYOR	Bool	2.0	false	☒	☒	☒	☐
Temperatura_Dash [DB11]	13	ENTRY CONVEYOT	Bool	2.1	false	☒	☒	☒	☐
UBICACION DE CAJAS_DB [D...	14	FORSK RIGHT	Bool	2.2	false	☒	☒	☒	☐
System blocks	15	FORS LEFT	Bool	2.3	false	☒	☒	☒	☐
Technology objects	16	TARJET POSITION	Int	4.0	0	☒	☒	☒	☐
External source files	17	LIFT	Bool	6.0	false	☒	☒	☒	☐
PLC tags	18	InOut				☐	☐	☐	☐
PLC data types	19	<Add new>				☐	☐	☐	☐
Watch and force tables	20	Static				☐	☐	☐	☐
Online backups	21	paso 0	Bool	8.0	false	☒	☒	☒	☐
Traces	22	paso 1	Bool	8.1	false	☒	☒	☒	☐
OPC UA communication	23	paso 2	Bool	8.2	false	☒	☒	☒	☐
Web applications	24	paso 3	Bool	8.3	false	☒	☒	☒	☐
Device proxy data	25	paso 4	Bool	8.4	false	☒	☒	☒	☐
Program info	26	paso 5	Bool	8.5	false	☒	☒	☒	☐
PLC supervisions & alarms	27	paso 6	Bool	8.6	false	☒	☒	☒	☐
PLC alarm text lists	28	MOV1	Int	10.0	0	☒	☒	☒	☐
Local modules	29	MOVEM	Int	12.0	0	☒	☒	☒	☐
Ungrouped devices	30	SAL	Real	14.0	0.0	☒	☒	☒	☐
Security settings	31	AUX1	Bool	18.0	false	☒	☒	☒	☐
Settings	32	AUX 2	Bool	18.1	false	☒	☒	☒	☐
Users and roles	33	AUX 3	Bool	18.2	false	☒	☒	☒	☐
Cross-device functions	34	AUX 4	Bool	18.3	false	☒	☒	☒	☐
Common data									

Figura C.1. Declaraciones de FB1 UBICACION DE CAJAS: entradas, salidas, memoria de pasos y variables auxiliares [P8].

La interfaz reúne las entradas, las salidas y las variables internas del bloque. Su lectura complementa el análisis de las seis redes y permite relacionar las señales del proceso con la memoria utilizada por la secuencia.

C.1 Inicio y reinicio de la secuencia

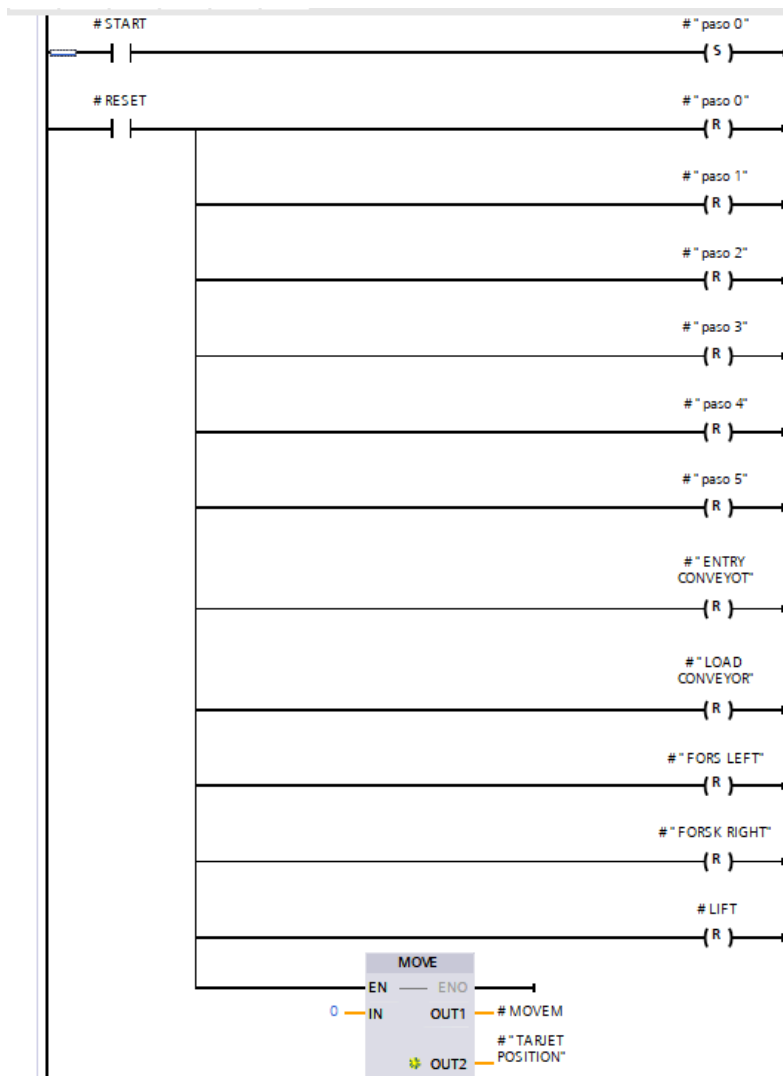


Figura C.2. Red 1 de FB1: activación por START, borrado por RESET y transferencia de cero a MOVEM y al destino [P8].

RESET borra paso 0 a paso 5 y las salidas enclavadas visibles. No se observa paso 6 en las bobinas de esta red. El contador y MOV1 se revisan además en la red 5; no debe interpretarse este MOVE como reinicio de todas las variables del bloque.

C.2 Recepción y transferencia de la carga

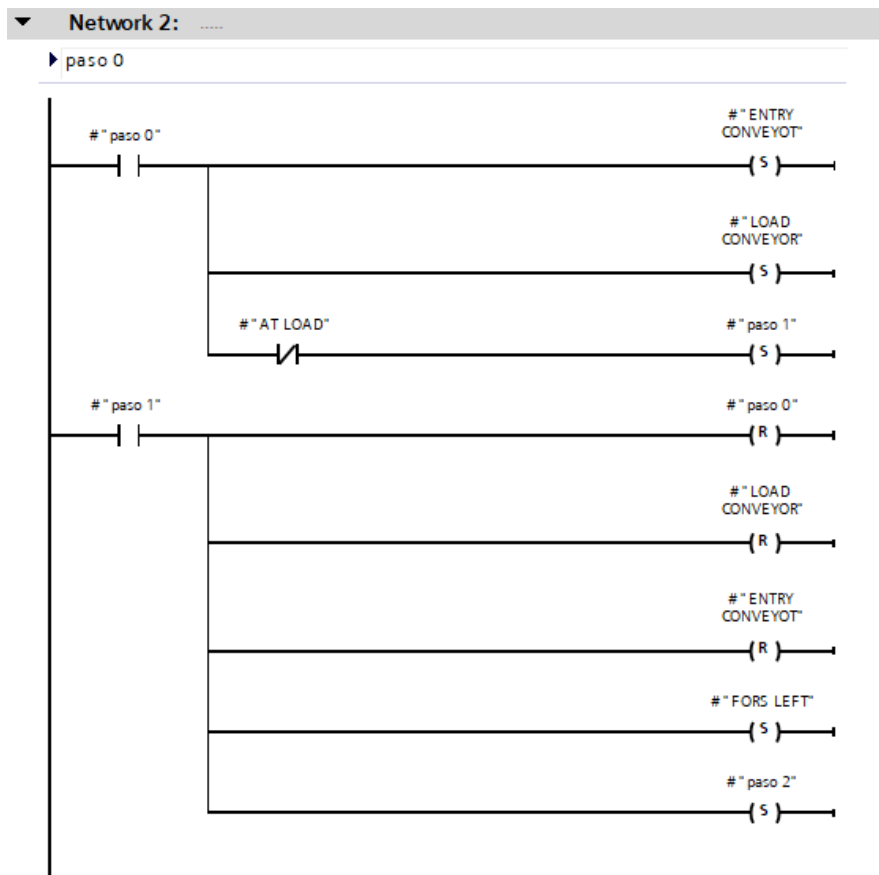


Figura C.3. Red 2: transportadores, condición negada AT LOAD y activación de FORS LEFT y paso 2 [P8].

El contacto AT LOAD aparece negado en la transición a paso 1. La condición eléctrica/lógica debe conservarse tal como está programada. Antes de describirla como presencia o ausencia física de carga es necesario contrastarla online con la señal de Factory I/O.

C.3 Temporización de elevación y avance

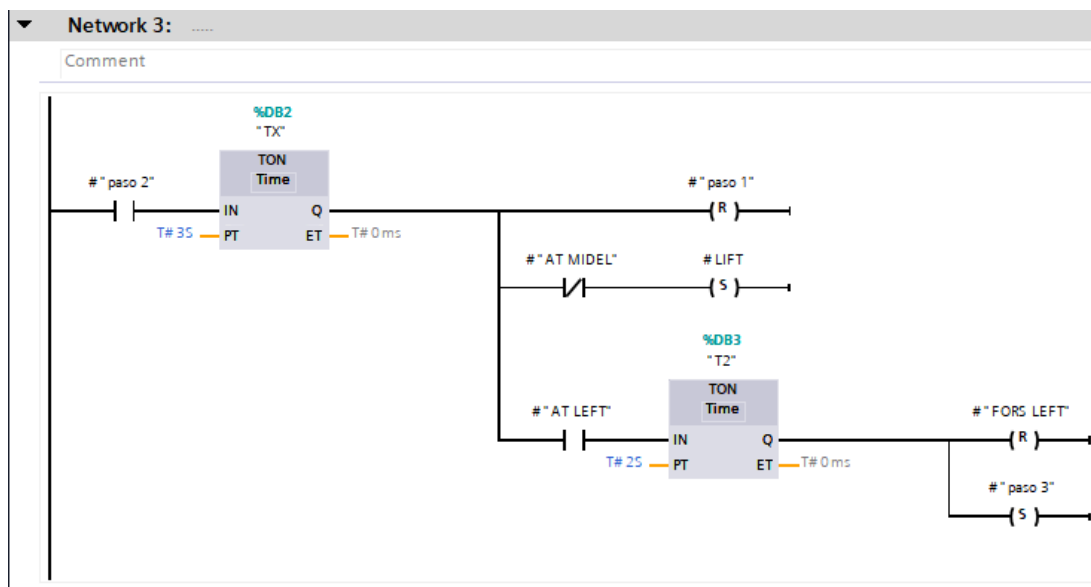


Figura C.4. Red 3: TX = 3 s y T2 = 2 s; condiciones AT MIDL y AT LEFT [P8].

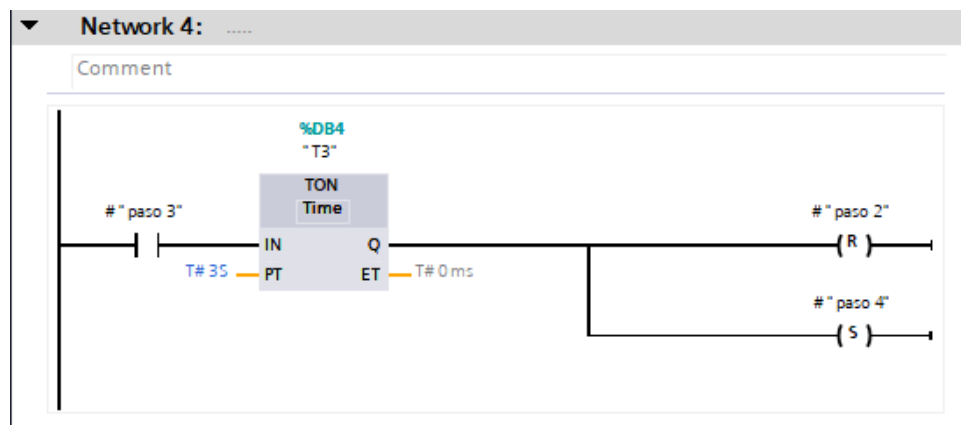


Figura C.5. Red 4: T3 = 3 s, borrado de paso 2 y activación de paso 4 [P8].

La salida de TX borra paso 1. La elevación depende además de AT MIDL negado; el avance a paso 3 requiere AT LEFT y el término de T2. T3 establece la transición posterior a paso 4. Estos valores son consignas de temporizadores, no tiempos medidos del movimiento.

C.4 Conteo y asignación de posiciones

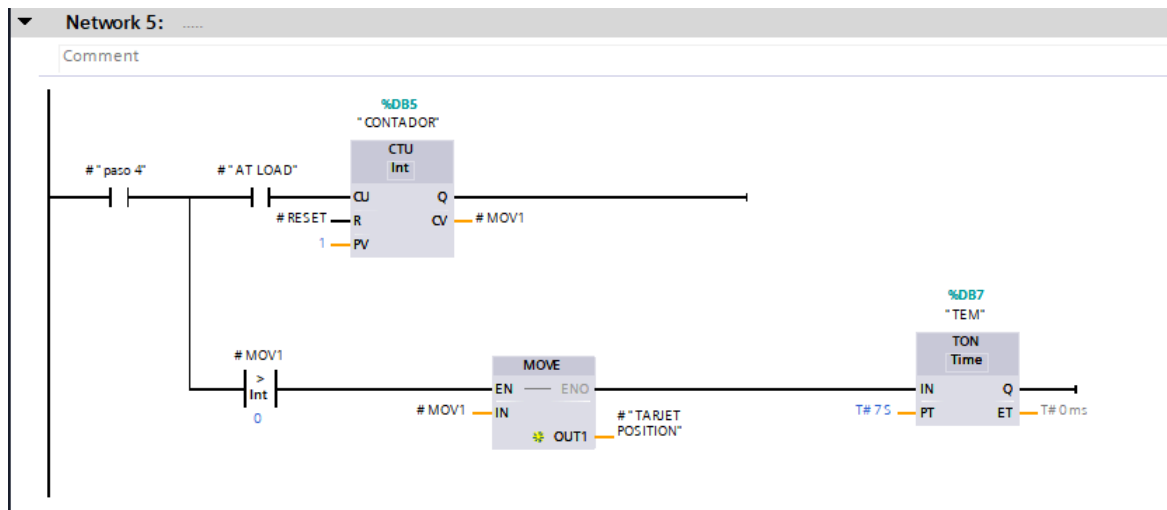


Figura C.6. Red 5: CTU CONTADOR, CV asociado a MOV1, MOVE al destino y TON TEM = 7 s [P8].

El conteo se vincula a la condición paso 4 y AT LOAD. El valor PV = 1 del CTU no limita el contador a las 54 ubicaciones. La captura no muestra una comparación superior para MOV1 ni una acción conectada a Q del temporizador TEM. Deben verificarse ambos puntos en el programa completo y en los ensayos.

C.5 Depósito y orden de retorno

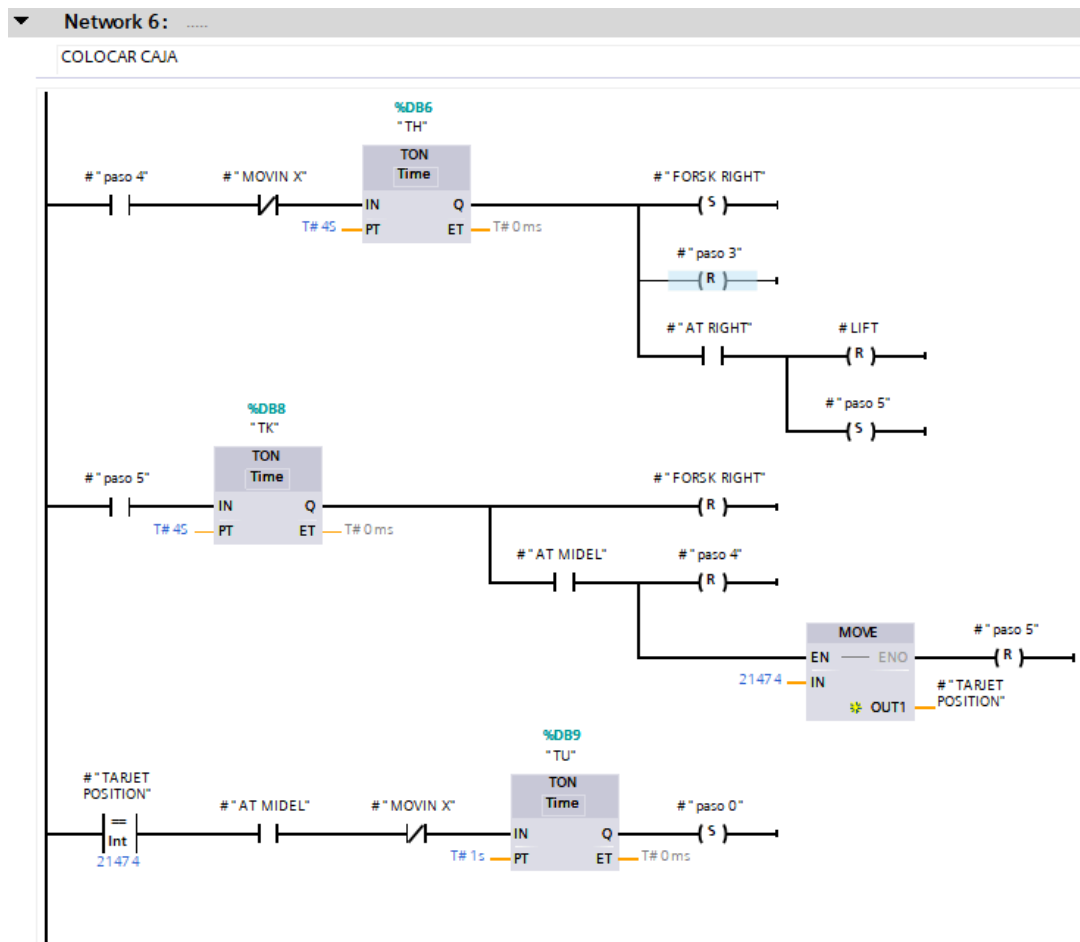


Figura C.7. Red 6: TH = 4 s, TK = 4 s, TU = 1 s y transferencia del destino especial 21474 [P8].

La secuencia combina ausencia de movimiento en X, AT RIGHT, AT MIDEL y temporizadores. El número 21474 se usa en el programa para la etapa de retorno y queda fuera del rango de posiciones que acumula Python. Su efecto sobre la escena y la finalización del depósito requieren comprobación online.

C.6 Modo y conexiones funcionales del PID

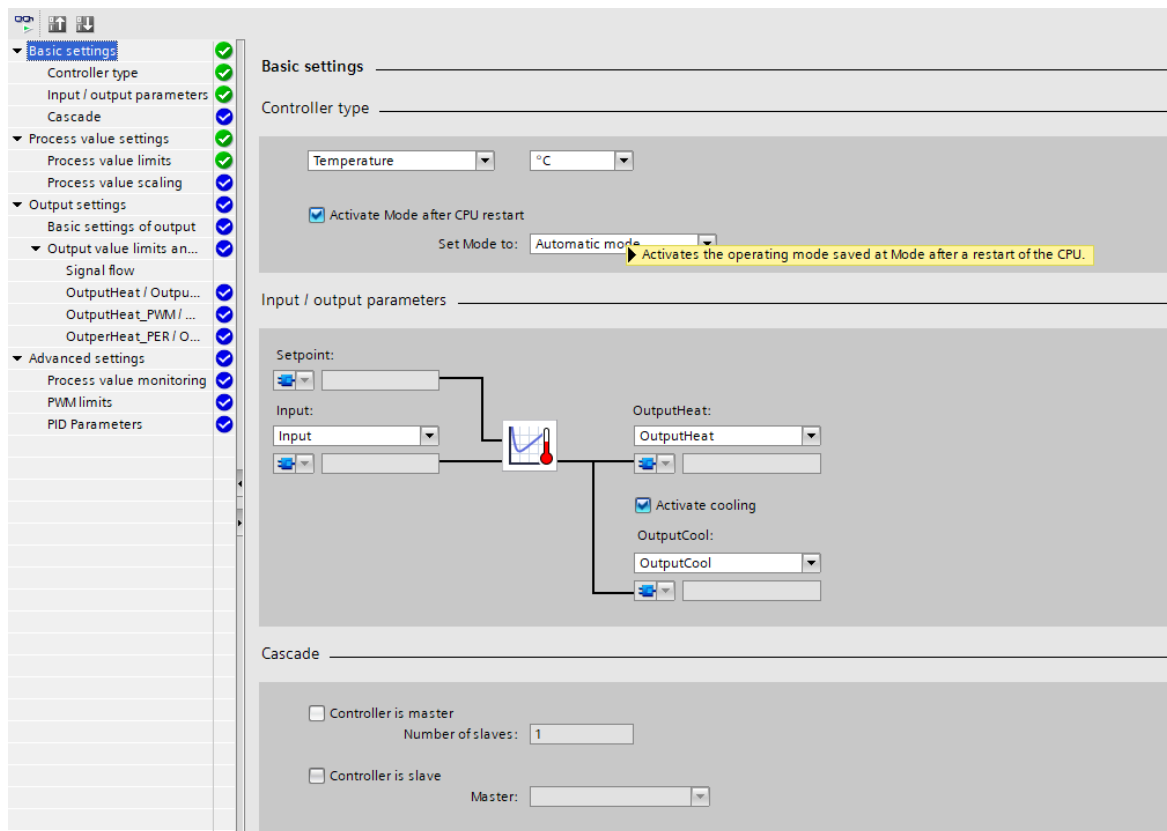


Figura C.8. PID_Temp: tipo Temperature, unidad °C, Automatic mode tras reinicio y Activate cooling seleccionado [P8].

El regulador utiliza Input y ofrece OutputHeat y OutputCool; no se selecciona funcionamiento maestro o esclavo en cascada. Las conexiones concretas con DB11 se muestran en la figura 4.5. La selección de modo tras reinicio no sustituye la comprobación del estado efectivo durante funcionamiento.

C.7 Escalado de salidas y reacción ante error

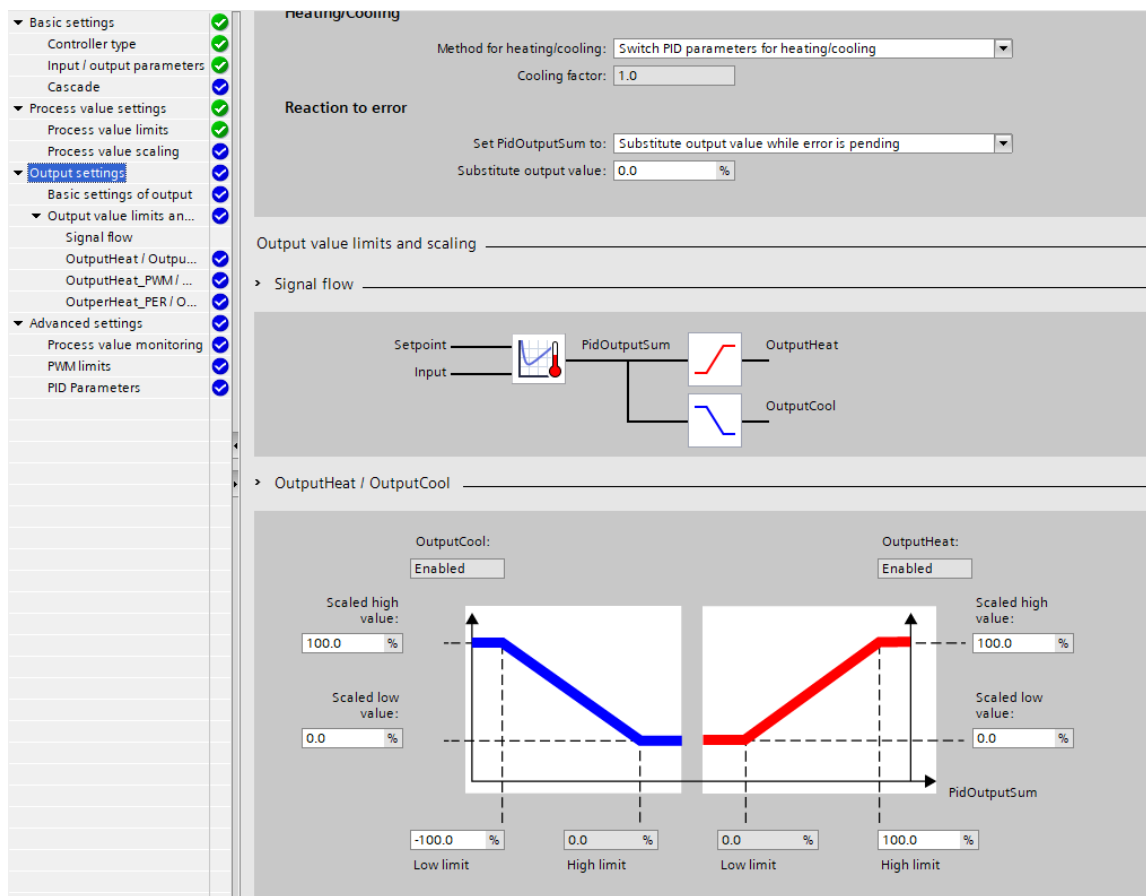


Figura C.9. PID_Temp: conmutación de parámetros calor/frío, factor 1,0, valor sustitutivo 0,0 % y escalas de OutputCool y OutputHeat [P8].

El tramo negativo de PidOutputSum se transforma en una salida de frío de 0 % a 100 %. Esta escala es consistente con la división por 100 utilizada por el modelo Python.

C.8 Límites y entrada de la variable de proceso

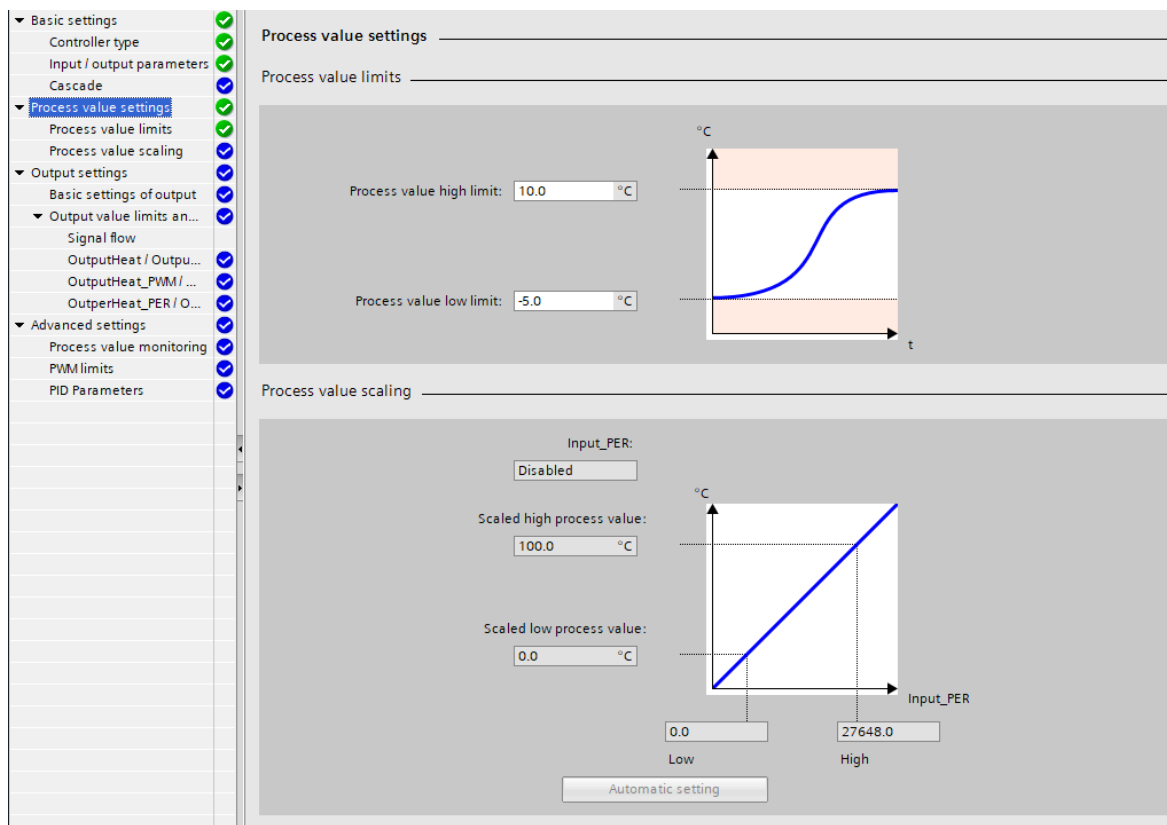


Figura C.10. PID_Temp: límites de proceso de $-5,0\text{ }^{\circ}\text{C}$ y $10,0\text{ }^{\circ}\text{C}$; Input_PER deshabilitado [P8].

La pantalla también muestra campos de escalado periférico, pero Input_PER está deshabilitado. La llamada utiliza Input con TEMP_PV. Los límites del regulador no cambian por sí mismos la ecuación del simulador térmico ni acreditan un rango válido de conservación de alimentos.

C.9 Permiso de comunicación con la CPU

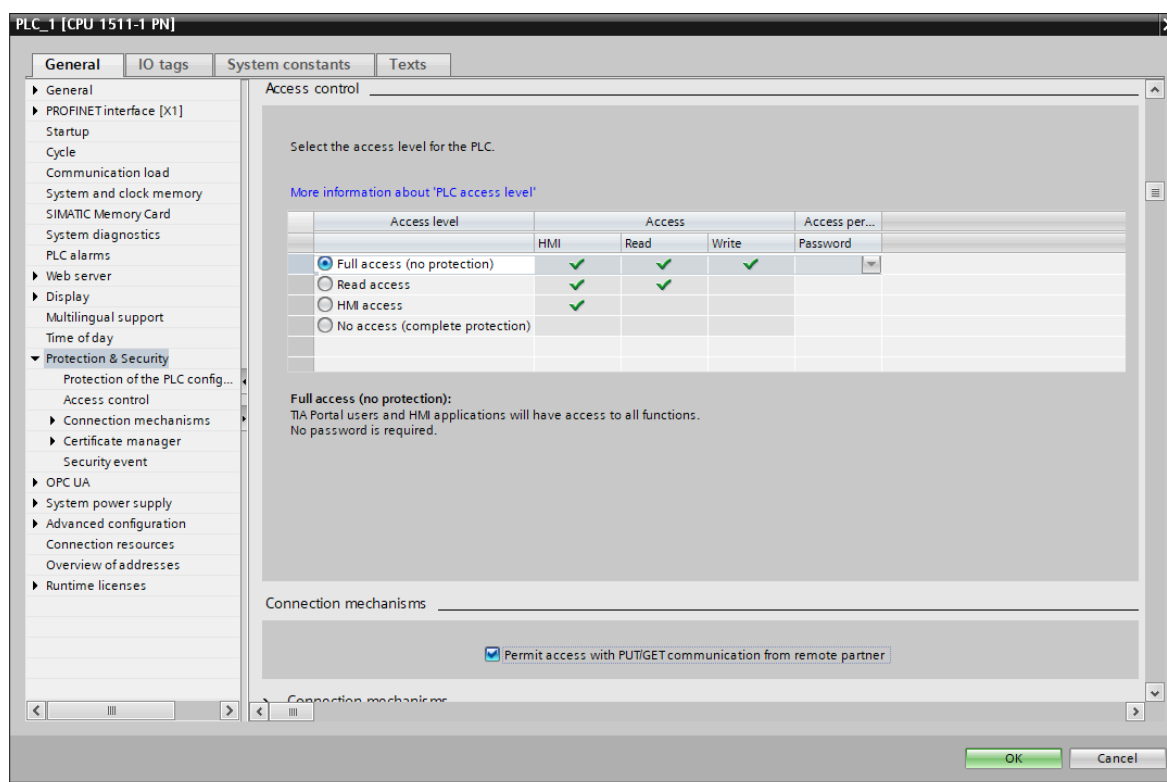


Figura C.11. Propiedades de CPU 1511-1 PN: Full access (no protection) y permiso PUT/GET seleccionado [P8].

La captura documenta los ajustes de acceso externo utilizados como referencia para la comunicación del banco de simulación.

C.10 Habilitación de simulación durante compilación

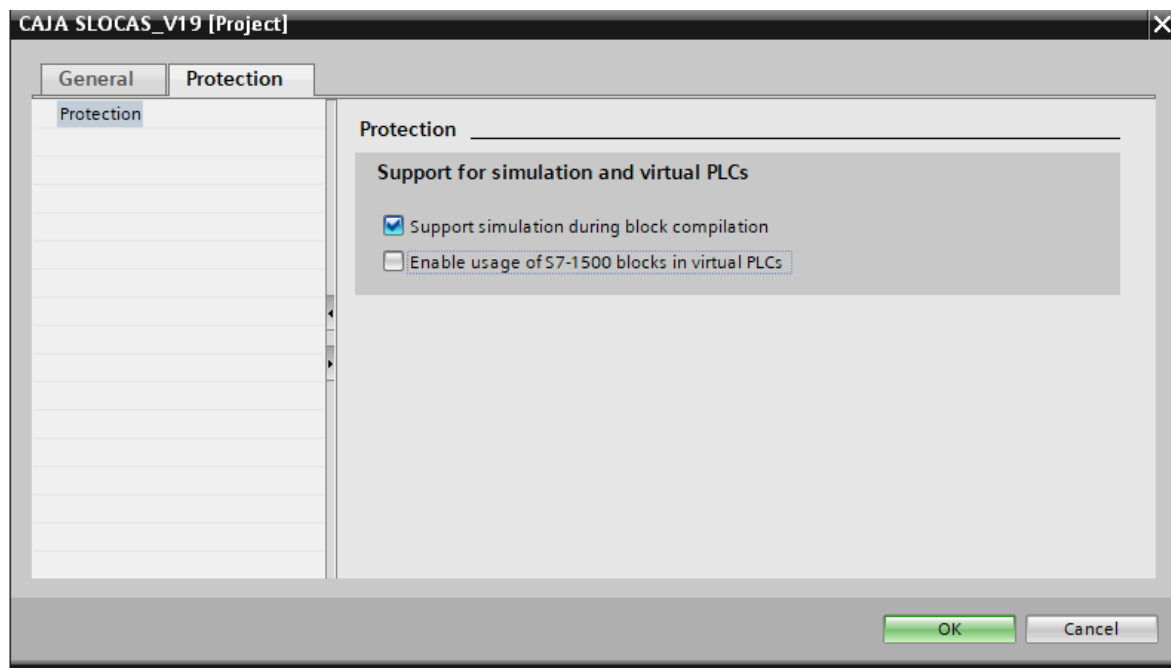


Figura C.12. Propiedades del proyecto CAJA SLOCAS_V19: Support simulation during block compilation seleccionado [P8].

La captura corresponde a las opciones de simulación del proyecto. Se observa habilitado el soporte de simulación durante la compilación de bloques; la segunda opción mostrada permanece sin seleccionar [P8].