**ESCUELA SUPERIOR DE INGENIERÍA Y TECNOLOGÍA**

*Grado en Ingeniería Electrónica Industrial y Automática*

# *TRABAJO FIN DE GRADO*

# IMPLEMENTACIÓN DE UN SERVIDOR OPC INTEGRADO EN UN SISTEMA SCADA PARA LA AUTOMATIZACIÓN DE CÉLULAS DE FABRICACIÓN

**Autor:**

Jhon Sebastián Rodríguez Quintero

**Tutor:**

Roberto Luis Marichal Plasencia

# ÍNDICE MEMORIA

# ÍNDICE FIGURAS

# ÍNDICE TABLAS

# 1. Abstract

The objective of this project is to implement a server in a *SCADA* (Supervisory Control and Data Acquisition) system for the automation of manufacturing cells. To achieve this, we rely on the *SCADA* system developed by engineers Javier de la Rosa and Alexander Epifanio Corona Ledesma.

A *SCADA* system is a comprehensive solution that combines hardware, software, and communications to supervise, control, and optimize industrial processes. It provides a complete view of the plant, facilitating decision-making, improving efficiency and safety, and enabling more efficient resource management.

Within a *SCADA* system, the *OPC* (Open Protocol Communication) server plays a fundamental role. It acts as an intermediary between field devices and the central supervision and control system. The main function of the *OPC* server in this project is to collect, process, and transmit data through standard communication protocols. Together, the *OPC* server enhances the efficiency and reliability of the *SCADA* system by enabling effective real-time monitoring and control of industrial processes.

In addition to the *OPC* server, there are other servers applicable in different contexts. For example, the *HTTP* (Hypertext Transfer Protocol) server is primarily used to deliver web content and facilitate interaction with online services. In the case of this project, *HTTP* communication tests will be conducted to verify its functionality. On the other hand, the *VNC* (Virtual Network Computing) server is used for remote communication and control of computers. It allows users to remotely access and control devices from another location, which is especially useful for remote technical support.

# 2. Resumen

El objetivo de este proyecto es implementar un servidor *OPC* en un sistema *SCADA* (Supervisory Control and Data Acquisition) para la automatización de células de fabricación. Para ello, este trabajo está basado en un sistema *SCADA* desarrollado previamente por los ingenieros Javier Rodríguez de la Rosa y Alexander Epifanio Corona Ledesma.

Un sistema *SCADA* es una solución integral que combina hardware, software y comunicaciones para supervisar, controlar y optimizar procesos industriales. Proporciona una visión completa de la fábrica, facilitando la toma de decisiones, mejorando la eficiencia y la seguridad, y permitiendo una gestión más eficiente de los recursos.

Dentro de un sistema *SCADA* el servidor *OPC* (Open Protocol Communication) desempeña un papel fundamental. Actúa como un intermediario entre los dispositivos de campo y el sistema central de supervisión y control. La función principal del servidor *OPC* en este proyecto es recopilar, procesar y transmitir datos a través de protocolos de comunicación estándar. Además, el servidor *OPC* mejora la eficiencia y confiabilidad del sistema *SCADA* al permitir una supervisión y control eficaz de los procesos industriales en tiempo real.

Por otra parte, además del servidor *OPC* existen otros servidores que son aplicables en diferentes contextos, como puede ser el servidor *HTTP* (Hypertext Transfer Protocol) utilizado para entregar contenido web, facilitando la interacción con servicios en línea. En el caso de este proyecto, se realizarán pruebas de comunicación *HTTP* para verificar su funcionamiento. Por otro lado, se ha implementado el servidor *VNC* (Virtual Network Computing), se utiliza para la comunicación y control remoto de ordenadores. Permite a los usuarios acceder y controlar dispositivos de forma remota desde otra ubicación, lo que resulta especialmente útil para el soporte técnico a distancia.

# 3. Objeto

El objetivo de este trabajo es implementar un servidor *OPC* en la planta *FESTO*, así como investigar e implementar diferentes tipos de servidores. Los objetivos específicos incluyen:

- Aprender y desarrollar habilidades en el software *TIAPortal* (Totally Integrated Automation Portal).
- Programar en el lenguaje *KOP* (Kontaktplan).
- Interactuar con pantallas *HMI* (Human-Machine Interface).
- Realizar correcciones y mejoras en la programación de la planta *FESTO*.
- Creación un servidor *OPC* para manipular variables y transferir datos.
- Simular pantallas *HMI* para realizar pruebas con servidores *HTTP* y servidores *VNC*.
- Probar el software *PLCSIM Advanced* para simular *PLC* (Programmable Logic Controller).

# 4. Sistema SCADA

*SCADA* [1] es un sistema de control y adquisición de datos utilizado en diversas industrias para supervisar y controlar procesos en tiempo real. El sistema *SCADA* se compone de software y hardware que trabajan en conjunto para recopilar, monitorizar y analizar datos de dispositivos y equipos distribuidos en una planta, instalación o red.

El sistema *SCADA* permite a los operadores supervisar y controlar de forma remota distintos procesos industriales. Utiliza una arquitectura cliente-servidor, donde los dispositivos de campo (sensores, actuadores, controladores) están conectados a una red de comunicación, que a su vez está conectada a un servidor central.

El servidor *SCADA* recibe y procesa los datos enviados desde los dispositivos de campo presentando la información de manera visual a los operadores a través de una interfaz gráfica. Esta interfaz permite supervisar el estado de los equipos, recibir alarmas, notificaciones, realizar ajustes y controlar los procesos.

Además de monitorizar y controlar, los sistemas *SCADA* también pueden almacenar datos históricos para su posterior análisis y generación de informes. Estos datos pueden utilizarse para detectar patrones, realizar análisis de tendencias y optimizar los procesos.

El esquema del *SCADA* de este proyecto es el siguiente:

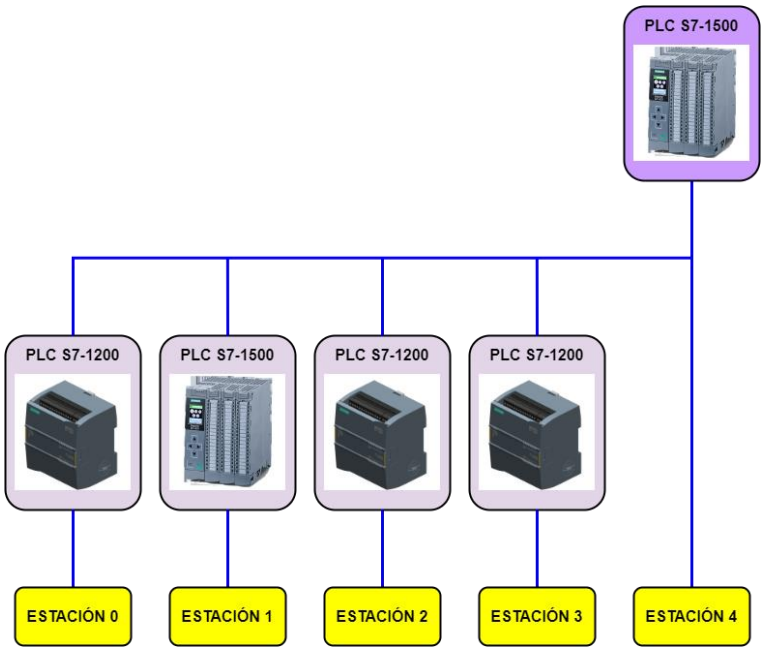



*Figura 1. Esquema del SCADA*

Este *SCADA* se puede dividir en tres niveles:

1. El **nivel Actuador/Sensor** es el nivel más bajo del sistema *SCADA*, donde se encuentran los dispositivos físicos encargados de la recopilación de datos y la actuación sobre el entorno controlado (estaciones *FESTO*). Estos dispositivos son fundamentales para el funcionamiento del sistema *SCADA*, ya que proporcionan la entrada y salida de información entre el mundo físico y el sistema de control.

2. En el **nivel de campo** se encuentran los *PLC´*s, desempeñan un papel vital en el sistema *SCADA* al controlar los procesos, adquirir datos, administrar las interfaces de E/S, programarse según la lógica de control y generar alarmas y eventos cuando sea necesario. Su funcionalidad es esencial para la automatización y supervisión eficiente del proceso industrial.

3. El **nivel medio** en un sistema *SCADA* se ubica por encima del nivel de campo, y se encarga de realizar funciones de procesamiento,

almacenamiento y control de los datos adquiridos. En este caso se utiliza el *S7-1500* de la estación *4* en la comunicación con los otros autómatas.

## 4.1. FESTO

El proyecto gira entorno al sistema de producción modular *FESTO* [2]. El sistema está compuesto por cinco estaciones, las cuales simulan células de fabricación de una instalación industrial real. Las plantas se encuentran en las instalaciones de la *ESIT* (Escuela Superior de Ingeniería y Tecnología), concretamente en el Laboratorio Profesor Lorenzo Moreno Ruiz.

El sistema a contralar está conformado por las siguientes estaciones: Estación *0*: *"Almacén y Distribución de Piezas",* Estación *1*: *"Testeo de Piezas",* Estación *2*: *"Procesado de Piezas",* Estación *3*: *"Acarreo de Piezas",* Estación *4*: *"Clasificación de Piezas"* [3].

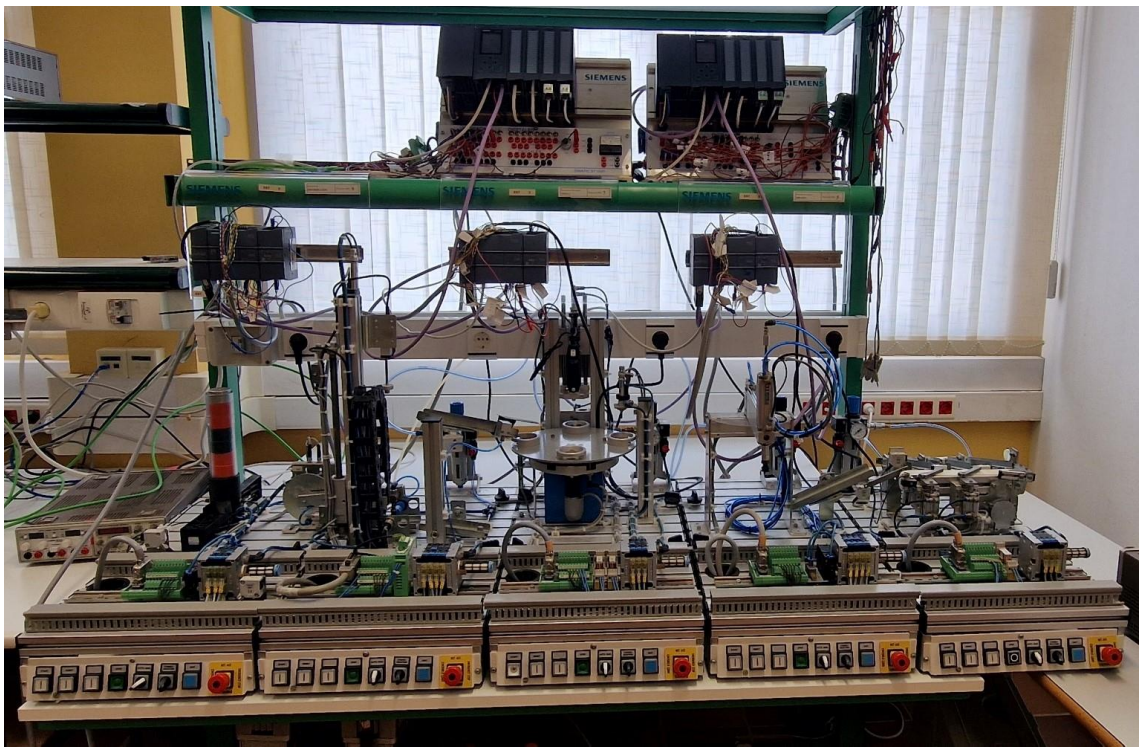



*Figura 2. Estaciones FESTO*

### 4.1.1. Estaciones

#### 4.1.1.1. Estación 0

En esta estación se hallan las piezas apiladas en un almacén y su función es, por una parte, sacar las piezas del almacén para después transportar las piezas a la estación *1* mediante un brazo mecánico rotativo. Para controlar esta planta se utiliza un autómata *S7-1200*.

Las piezas son extraídas del almacén mediante un pistón de expulsión. Cuando el brazo se encuentre en la posición del almacén y tiene una pieza disponible para transportar, se activa una ventosa que succiona la pieza y se mueve a la siguiente estación. Al llegar a la estación siguiente, el brazo desactiva la ventosa y deja caer la pieza sobre la plataforma de elevación de la estación *1*. Una vez depositada la pieza, el brazo regresa a la posición anterior para extraer otra pieza del almacén.

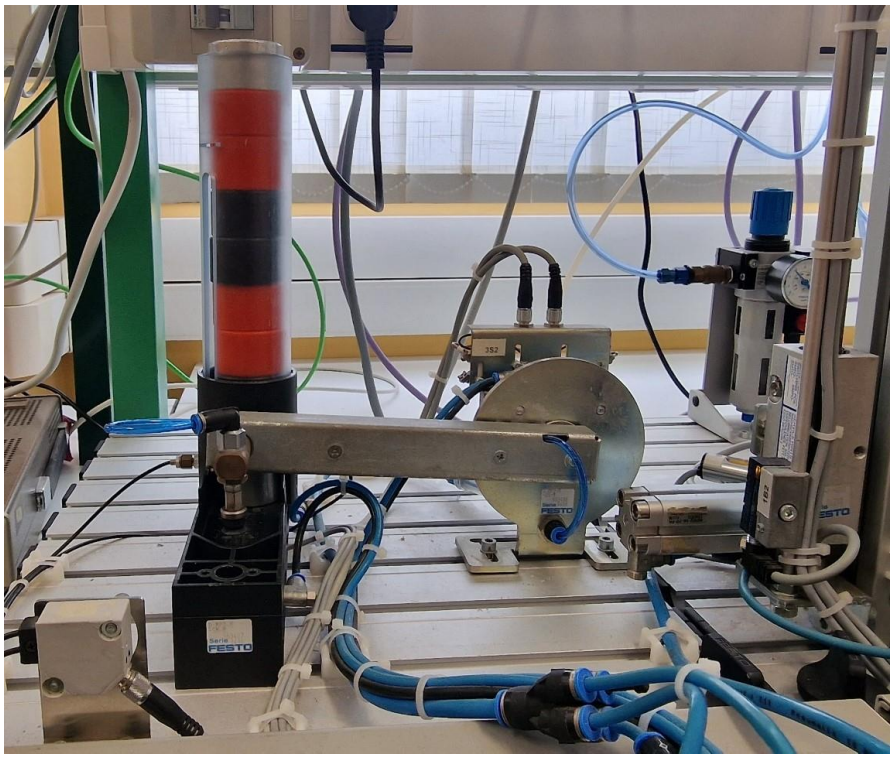



*Figura 3. Estación 0*

#### 4.1.1.2. Estación 1

La estación test tiene como propósito medir el tamaño de cada pieza. Si la pieza medida se encuentra dentro del tamaño considerado 'normal', se procede a enviarla a la siguiente estación. Sin embargo, si la pieza es más grande o pequeña que el tamaño 'normal', se descarta. Para supervisar y controlar esta planta, se utiliza un autómata *S7-1500*.

Cuando la estación *1* recibe la pieza de la estación *0*, se coloca en una plataforma que puede ser colocada en dos ubicaciones diferentes (posición *1*: abajo; posición *2*: arriba). Cada una de estas posiciones cuentan con una rampa que cumplen una función específica.

En la posición *1* se encuentra un sensor que detecta la presencia de una pieza en la plataforma. Una vez se detecta la pieza, la plataforma sube hasta la posición *2*; al llegar arriba, baja un sensor que mide la longitud de la pieza.

Esta medida se compara con el valor de 'longitud normal'. Si la pieza tiene el tamaño considerado 'normal', se procede a enviarla a la siguiente estación mediante la rampa *2* (ver ***Figura 4***). Con la pieza ya enviada, la plataforma debe bajar a la posición *1* a la espera de una nueva pieza.

Por otro lado, si la longitud de la pieza medida resulta ser mayor o menor que la longitud 'normal', se considera que es una pieza defectuosa, en tal caso, la plataforma debe descender a la posición *1* con la pieza defectuosa. Con ayuda de un pistón se desecha la pieza por la rampa *1* (ver ***Figura 4***). Una vez que se ha empujado la pieza defectuosa, la plataforma debe permanecer a la espera de una nueva pieza procedente de la estación *0*.

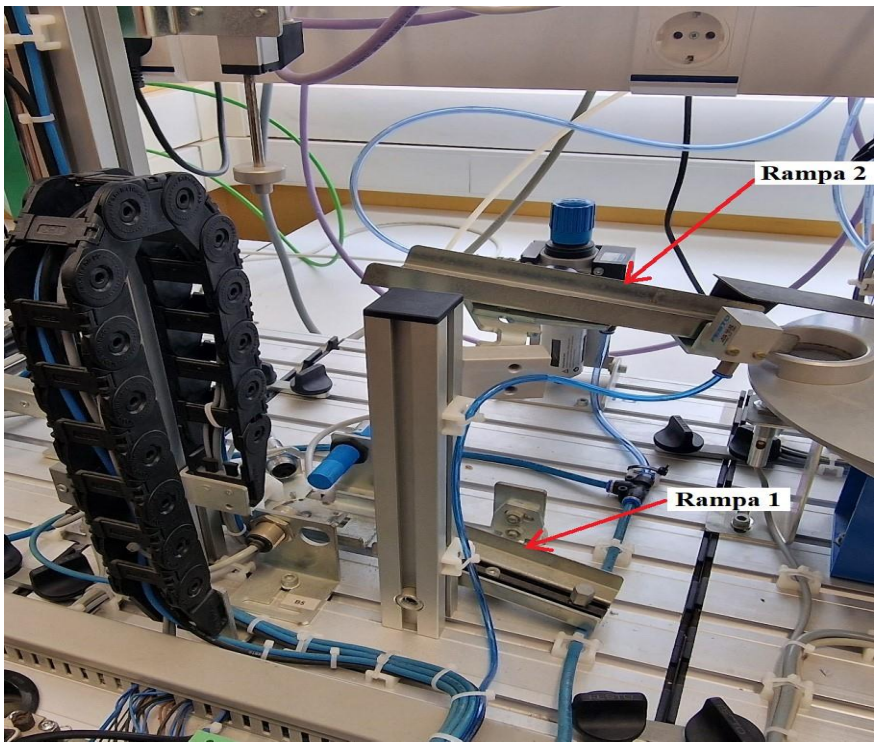



*Figura 4. Estación 1*

#### 4.1.1.3. Estación 2

Cuando llega una pieza de la estación anterior, la estación de procesado de piezas realiza una serie de operaciones de transformación sobre la pieza. Para ello, se utiliza un autómata *S7-1200*.

Este proceso se realiza con una mesa giratoria controlada por un motor. Dicha mesa posee cuatro posiciones de funciones específicas.

- **Posición *1*:** Recibe la pieza proveniente de la estación anterior, una vez la pieza este ubicada correctamente, la mesa gira *90º* para colocarse en la siguiente posición.

- **Posición *2*:** Un cilindro sujeta la pieza para que un taladro realice un orificio en la parte superior. Después de la perforación, el cilindro se retrae y el taladro se levanta. Un giro adicional de *90º* lleva la pieza a la posición *3*.

- **Posición *3*:** En la posición de verificación o posición *3*, un cilindro extensible comprueba si el orificio realizado anteriormente es correcto o por lo contrario hay que desecharla al almacén de piezas defectuosas. Tras la verificación, la mesa gira *90º*.

- **Posición *4*:** Finalmente, en la posición *4* la pieza espera ser recogida por la estación *3*. Una vez la pieza es recogida, la mesa vuelva a la posición *1*.

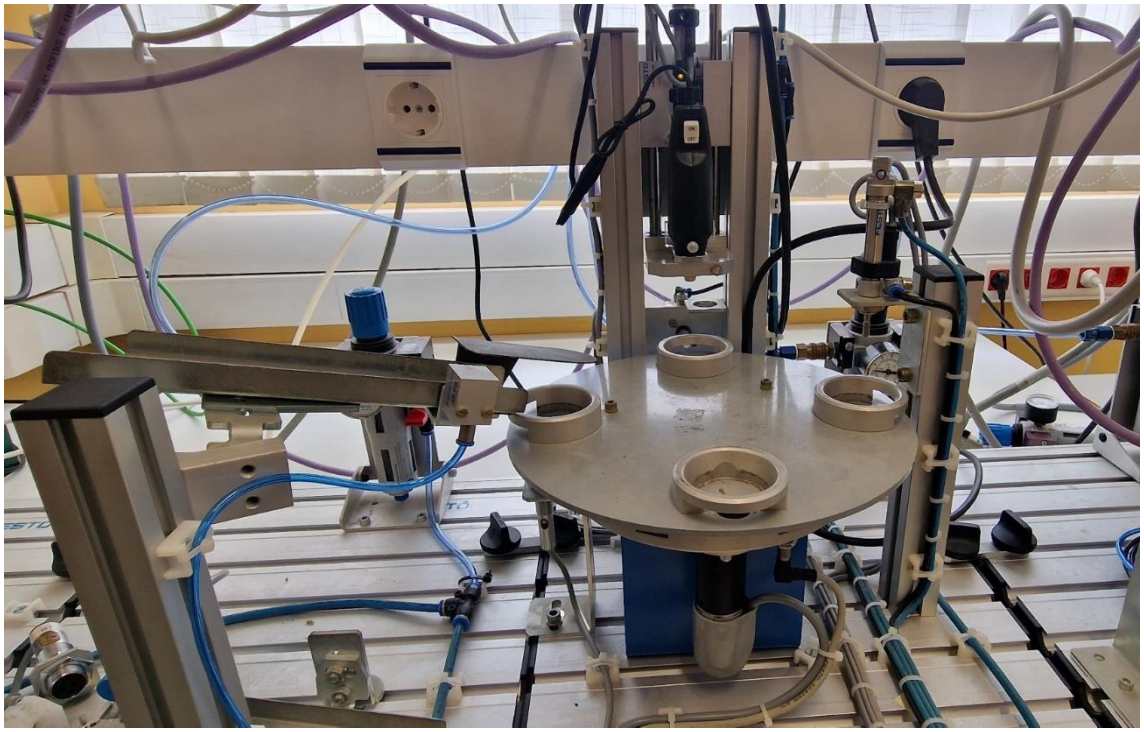


*Figura 5. Estación 2*

#### 4.1.1.3. Estación 3

Esta planta permite el traslado de una pieza desde la estación anterior de procesamiento hacia el almacén de piezas defectuosas en caso de que el taladro en la estación *2* no se haya realizado correctamente. Además, tiene la capacidad de enviar la pieza a la siguiente estación, conocida como estación clasificadora. Esta estación es controlada por un *S7-1200*.

El funcionamiento normal de esta estación solo puede llevarse a cabo si la estación previa y la estación siguiente han concedido el permiso correspondiente.

Un brazo giratorio con la capacidad de extenderse de forma vertical y horizontal se acerca a las estaciones adyacentes para coger o dejar las piezas.

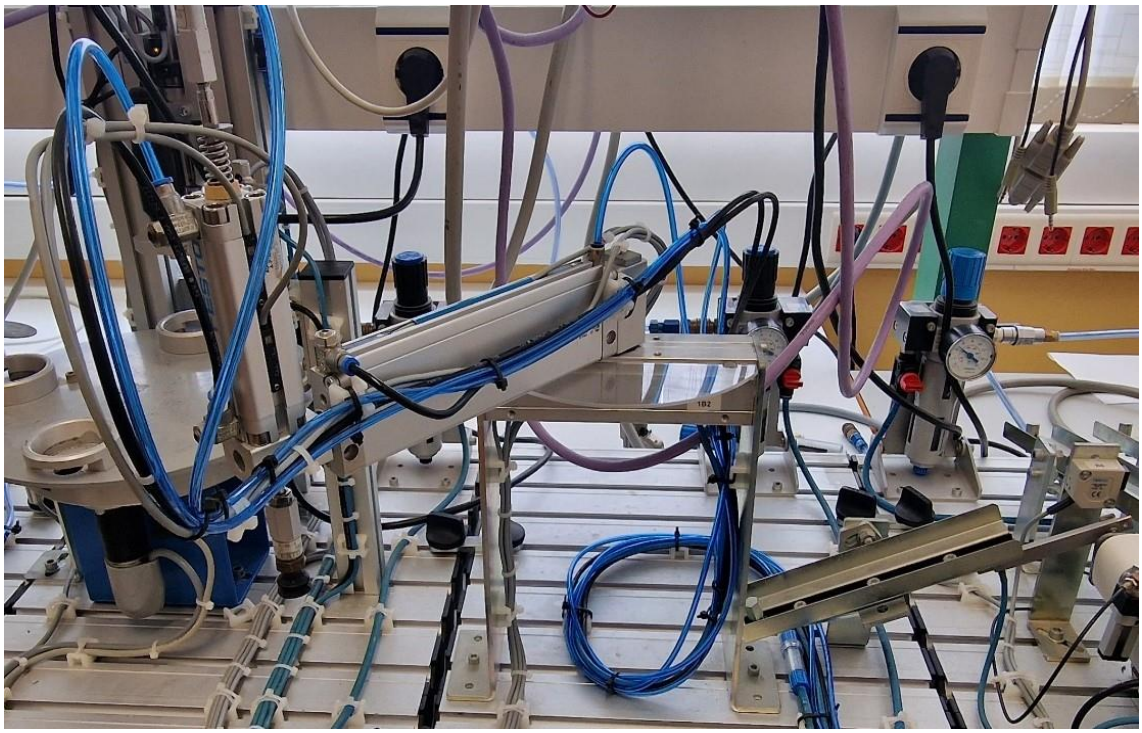


*Figura 6. Estación 3*

#### 4.1.1.5. Estación 4

En esta etapa del proceso, se realiza la clasificación de las piezas en tres categorías diferentes y se envían a distintos almacenes. Para controlar esta planta y establecer la comunicación con los otros autómatas, se utiliza un *S7-1500*. La función principal de este autómata es coordinar la automatización de los cincos estaciones mediante lectura y escritura en la memoria de los demás *PLC*'s.

La estación *3* deposita la pieza al principio de una cinta transportadora de un solo sentido de giro. La cinta empieza a moverse cuando se detecta una pieza en su inicio y dejará de moverse automáticamente cuando la pieza entre en alguno de los almacenes.

Los actuadores neumáticos son los encargados de enviar la pieza al almacén correspondiente.

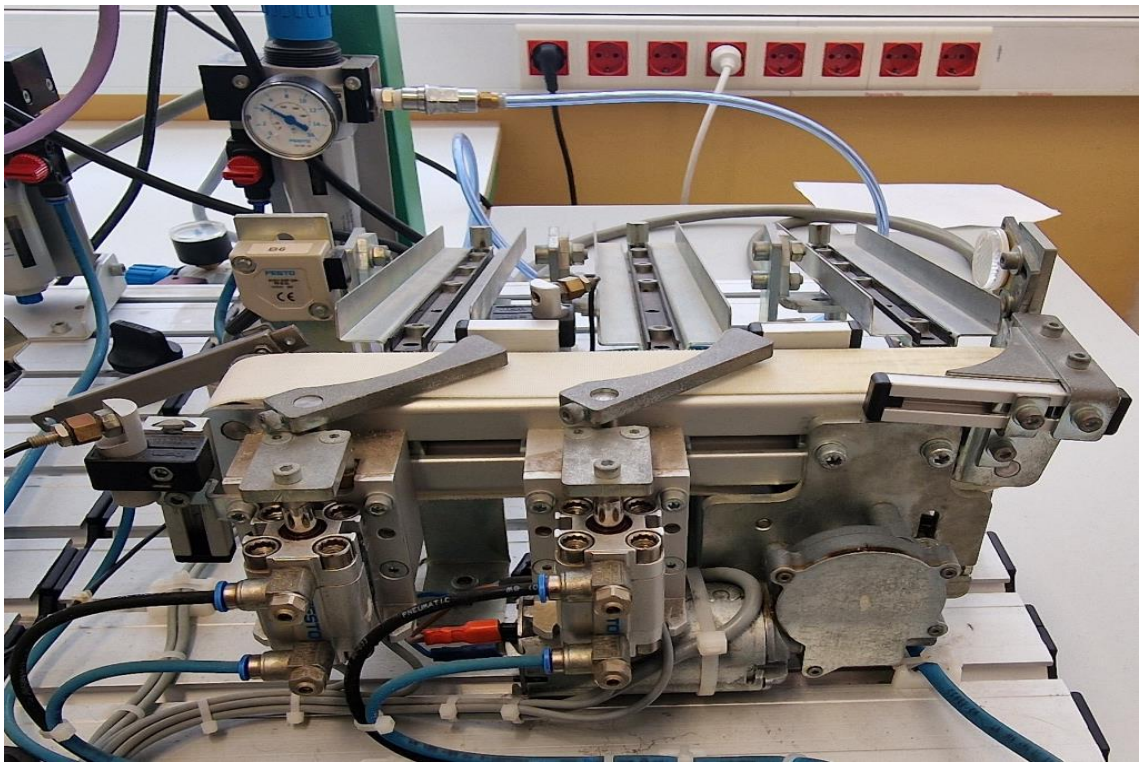


*Figura 7. Estación 4*

## 4.2. Software

*TIAPortal* [4] es el software principal de este proyecto, es la abreviatura de "Totally Integrated Automation Portal", es una plataforma de software desarrollada por Siemens que ofrece un entorno de ingeniería integrado para la configuración, programación, visualización y diagnóstico de sistemas de automatización industrial, se utiliza para programar controladores lógicos programables (*PLC*s), controladores de automatización industrial, sistemas de visualización de operador (*HMI*), control de motores y otros dispositivos relacionados con la automatización.

El *TIAPortal* permite a los ingenieros y programadores trabajar en un entorno centralizado, lo que simplifica el desarrollo de proyectos, reduciendo tiempo y esfuerzo.

Ofrece una interfaz gráfica intuitiva que facilita la configuración de hardware, la programación de lógica y la visualización de datos en tiempo real.

Además, el *TIAPortal* proporciona herramientas de diagnóstico avanzadas que ayudan a identificar y solucionar problemas de manera eficiente, lo que resulta en un mantenimiento más rápido.

En este proyecto se emplea el *TIAPortal* para las siguientes funciones:

- Programación de *PLC*'s en lenguaje *KOP*.
- Configuración Servidor *OPC UA*.
- Creación Servidor *HMI HTTP* con *WinCC* (Windows Control Center*)*.
- Configuración del SmartServer y SmartClient.
- Simulación de programas con *PLCSIM Advanced*.

### 4.2.1. WinCC

*WinCC* es un software de supervisión y control para ordenadores desarrollado por Siemens, se centra en el manejo de procesos, máquinas, instalaciones y líneas de fabricación. El *WinCC* en un sistema *SCADA* implica configurar y personalizar las funcionalidades de acuerdo con las necesidades específicas del proceso industrial.

Esta herramienta de tipo *IHMI* (Integrated Human Machine Interface) se integra al software del *PLC* para la automatización efectiva de la planta. El operador puede interactuar con la aplicación desde la máquina o desde el controlador.

El software *WinCC* permite programar la pantalla *HMI* desde el *TIAPortal*. Una vez transferido el programa al panel *HMI*, se ejecuta la rutina denominada "*Runtime*", la cual permite trabajar con la pantalla de forma dinámica.

*WinCC* se divide en diferentes categorías que ofrecen funcionalidades específicas según la necesidad del proyecto. Algunas de las categorías principales de *WinCC* son:

- ***WinCC Basic***: Esta categoría se centra en aplicaciones de nivel básico y es ideal para proyectos de automatización pequeños. Proporciona una funcionalidad esencial para la supervisión y control de procesos simples. Este modelo solo permite configurar y programar paneles básicos *HMI.*

- ***WinCC Comfort***: Esta categoría está diseñada para aplicaciones de mediana escala. Ofrece una interfaz de usuario intuitiva y funcionalidades esenciales para la supervisión y control de procesos industriales. El modelo *Comfort* es algo más complejo que el *Basic*, permitiendo trabajar con paneles móviles y multipantallas.

- ***WinCC Runtime Advanced***: Esta categoría está diseñada para la ejecución de proyectos de visualización en tiempo real. Proporciona una interfaz de usuario interactiva para supervisar y controlar procesos industriales. Permite el trabajo con sistemas monopuestos basados en *PC's*.

- ***WinCC Professional***: Esta categoría se enfoca en aplicaciones de supervisión y control de alto nivel. Proporciona una amplia gama de herramientas y características avanzadas para el monitoreo, la visualización y el control de procesos industriales complejos. Este modelo permite definir sistemas multipuestos basados en *PC's*, además permite la funcionalidad de los *SCADA's*.

### 4.2.2. PLCSIM Advanced

*PLCSIM Advanced* [9] es un software de simulación de controladores lógicos programables (*PLC*) desarrollado por Siemens. Está diseñado para simular y probar programas de control en entornos industriales sin necesidad de hardware físico.

Con *PLCSIM Advanced*, es posible simular diferentes modelos de controladores SIMATIC *S7-1500* y ejecutar programas de control en un entorno virtual. Proporciona una interfaz gráfica intuitiva que permite configurar y controlar la simulación, monitorizar variables, establecer puntos de ruptura y depurar programas de control.

Además, ofrece funcionalidades avanzadas, como la simulación de señales de entrada y salida, la integración con herramientas de ingeniería de Siemens y la posibilidad de realizar pruebas de diagnóstico.

## 4.3. Hardware

Como componentes de hardware principales del proyecto contamos con cinco autómatas, un panel *HMI* y un Switch Ethernet.

### 4.3.1. *PLC*'s

Un *PLC* [4] (Programmable Logic Controller) o Controlador Lógico Programable es un dispositivo electrónico utilizado en la automatización industrial. Actúa como el controlador central del sistema de control, recibe señales de entradas de sensores, realizando cálculos y lógica de control, para generar señales de salidas para activar actuadores. Los *PLC´*s son programables, lo que significa que se pueden cargar programas en ellos para definir el comportamiento y la lógica de control deseada.

Se cuenta con dos modelos de autómata: el *SIMATIC S7-1200* se utiliza en las estaciones *0*, *2* y *3*, mientras que el *SIMATIC S7-1500* se emplea en las estaciones *1* y *4*. El *PLC* de la estación *4* asume la responsabilidad de iniciar y detener el proceso de todos los autómatas restantes, desempeñando un papel de Maestro-Esclavo.

A continuación, se presenta una tabla que muestra las especificaciones técnicas de ambos modelos. Esta tabla fue elaborada por Javier Rodríguez de la Rosa [5] en su Trabajo de Fin de Grado (TFG) titulado "*Desarrollo de Sistema SCADA para la automatización de células de fabricación*", *2021*.

| CARACTERÍSTICAS | S7-1200 | *S7-1500* |
|---|---|---|
| Tensión de alimentación | 24V DC | 24V DC |
| Consumo (valor nominal) | 0,85 A | 0,85 A |
| Memoria de trabajo para programa | 75 Kbyte | 1 Mbyte |
| Memoria de trabajo para programa | 4 Mbyte | 5 Mbyte |
| Tiempo de ejecución de la CPU para operaciones a bits | 0,085 µs | 10 ns |
| Tiempo de ejecución de la CPU para operaciones a palabras | 1,7 µs | 12 ns |
| Nº de bloques (total) | 6000 | 6000 |
| Nº de interfaces Profinet | 1 | 2 |
| Permite lengUAje *KOP* | Sí | Sí |
| Permite lengUAje AWL | No | Sí |

*Tabla 1. Características S7-1200 Y S7-1500*

### 4.3.2. Panel *HMI*

Panel *HMI* [6] (Human-Machine Interface) es un dispositivo utilizado en sistemas de automatización industrial para permitir la interacción entre los operadores humanos y la maquinaria o sistema automatizado. También se conoce como "Interfaz Hombre-Máquina". Está compuesto por una pantalla táctil que permite a los operadores visualizar y controlar el proceso básico. Proporciona información en tiempo real sobre el estado del sistema, permite el seguimiento de variables y alarmas, y permite a los operadores interactuar con la máquina o sistema mediante toques, gestos o entradas. Además, puede tener capacidades de conectividad para comunicarse con otros dispositivos y sistemas, como controladores lógicos programables (*PLC*s) o sistemas de control distribuido (DCS).

Para el proyecto se dispone de un panel *SIMATIC HMI TP700 Comfort Panel* con las siguientes características principales:

| REFERENCIA | |
|---|---|
| 6ES7 124-0GC01-0AX0 | |
| VERSIÓN | |
| 16.0.0.0. | |
| DISPLAY | |
| Diagonal de pantalla | 7 in |
| Anchura del Display | 152,4 mm |
| Altura del Display | 91,4 |
| RESOLUCIÓN (PIXELES) | |
| Resolución de imagen horizontal | 800 pixeles |
| Resolución de imagen vertical | 480 pixeles |
| TENSIÓN DE ALIMENTACIÓN | |
| Tipo de tensión de la alimentación | DC |
| Valor nominal (DC) | 24 V |
| Rango admisible, límite inferior (DC) | 19,2 V |
| Rango admisible, límite superior (DC) | 28,8 V |
| INTENSIDAD DE ENTRADA | |
| Consumo (Valor nominal) | 0,5 A |
| POTENCIA | |
| Consumo de potencia activa | 12 W |

*Tabla 2. Características Panel HMI*

### 4.3.3. Switch Ethernet

Un switch Ethernet, también conocido simplemente como switch, es un dispositivo de red utilizado para interconectar dispositivos en una red local (LAN) mediante el protocolo Ethernet. Su función principal es facilitar la comunicación entre los dispositivos conectados, como ordenadores, impresoras, servidores, cámaras IP, entre otros.

La utilización de un switch Ethernet en una red local permite una mayor capacidad de transmisión de datos y una mejor gestión del tráfico entre los ordenadores del laboratorio, los *PLC*'s y la pantalla *HMI*.

## 4.4. Esquema de la planta de automatización

En la jerarquía del sistema, se puede distinguir dos roles en el comportamiento de los autómatas, *Maestro-Esclavo*. En este caso, el autómata de la estación *4* asume el rol del *Maestro*, este se encarga de adquirir y gestionar todas las variables de las demás estaciones. Una vez recopila esta información, se envía al panel *HMI* y/o *PC*. Para facilitar esta comunicación, se utiliza el Switch Ethernet, que actúa como punto de conexión.

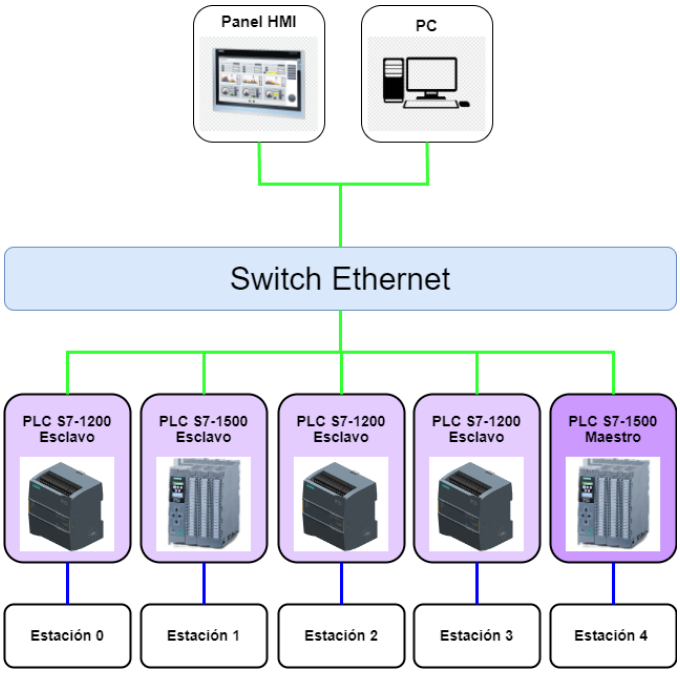



*Figura 8. Esquema de la planta*

# 5. Servidor OPC UA

Para asegurar un correcto funcionamiento del sistema *SCADA*, es fundamental recopilar y almacenar la información generada por cada estación de la *FESTO*. En este sentido, lo ideal sería disponer de un servidor *OPC* dedicado a recopilar esta información. Sin embargo, no se dispone de este recurso específico, por lo que se ha optado a utilizar el *PLC S7-1500* de la estación *4* como servidor *OPC UA*. Este autómata transferirá toda la información recopilada de la planta *FESTO* a un cliente, en este caso, será un ordenador y/o una pantalla *HMI*. El uso de este protocolo de servidor permite establecer comunicación bidireccional entre el servidor y el cliente, lo que permite tanto la lectura como la escritura de datos en ambos sentidos.

Como punto de inicio para la implementación de un servidor *OPC*, se comienza sobre el trabajo elaborado por los ingenieros Javier Rodríguez de La Rosa [5] y Alexander Epifanio Corona Ledesma [8] en sus respectivos Trabajos de Final de Grado (TFG).

Un servidor *OPC* UA [7] (Object Linking and Embedding for Process Control Unified Architecture) es un componente de software que proporciona conectividad y comunicación entre sistemas y dispositivos en entornos de automatización industrial. *OPC UA* es un estándar de comunicación abierto y ampliamente utilizado en la industria para el intercambio de datos y la interoperabilidad entre diferentes equipos y sistemas.

El servidor *OPC UA* actúa como un intermediario que permite a los clientes *OPC UA*, como aplicaciones de supervisión, controladores de automatización, dispositivos de campo u otros sistemas, acceder y compartir datos de manera segura y confiable.

Las principales características y funciones de un servidor *OPC UA* incluyen:

- **Conectividad multiplataforma**: Los servidores *OPC UA* pueden ejecutarse en diferentes sistemas operativos, como Windows, Linux o incluso sistemas embebidos, lo que facilita su implementación en una amplia gama de entornos.

- **Seguridad**: *OPC UA* ofrece mecanismos de seguridad robustos para proteger la integridad, autenticidad y confidencialidad de los datos transmitidos. Incluye cifrado de datos, autenticación de clientes y servidores, y o*pc*iones de autorización para controlar el acceso a los recursos.

- **Modelado de información**: *OPC UA* utiliza un modelo de información estructurado y extensible, basado en objetos y nodos, para representar los datos y servicios disponibles en un sistema. Esto permite una descripción clara y organizada de los elementos de automatización y facilita la interoperabilidad entre diferentes sistemas.

- **Comunicación escalable**: Los servidores *OPC UA* utilizan una arquitectura cliente-servidor, donde los clientes pueden solicitar datos y servicios al servidor. La comunicación se realiza a través de protocolos como TCP/IP, lo que permite la transmisión eficiente de grandes volúmenes de datos.

- **Integración de datos en tiempo real**: El servidor *OPC UA* puede proporcionar datos en tiempo real, lo que permite la monitorización, el control y la toma de decisiones rápidas en sistemas de automatización.

En resumen, la jerarquía de comunicación entre un servidor *OPC* y un cliente *OPC* abarca desde el nivel físico de la conexión hasta el nivel de datos, pasando por los niveles de red. Permitiendo la interacción y el intercambio de información en un entorno de control.

Por otro lado, a un nivel similar al cliente *OPC*, se encuentra el panel *HMI*, que establece comunicación con la estación *4*. A través de esta comunicación, el panel *HMI* puede enviar comandos y recibir los datos de la estación *4*, lo que permite supervisar las operaciones de la planta.

La comunicación entre la pantalla *HMI* y el cliente *OPC* se establece mediante una conexión de datos para intercambiar información. El cliente *OPC* puede enviar solicitudes al servidor *OPC* para obtener datos específicos y la pantalla *HMI* puede recibir y mostrar esos datos en su interfaz gráfica.

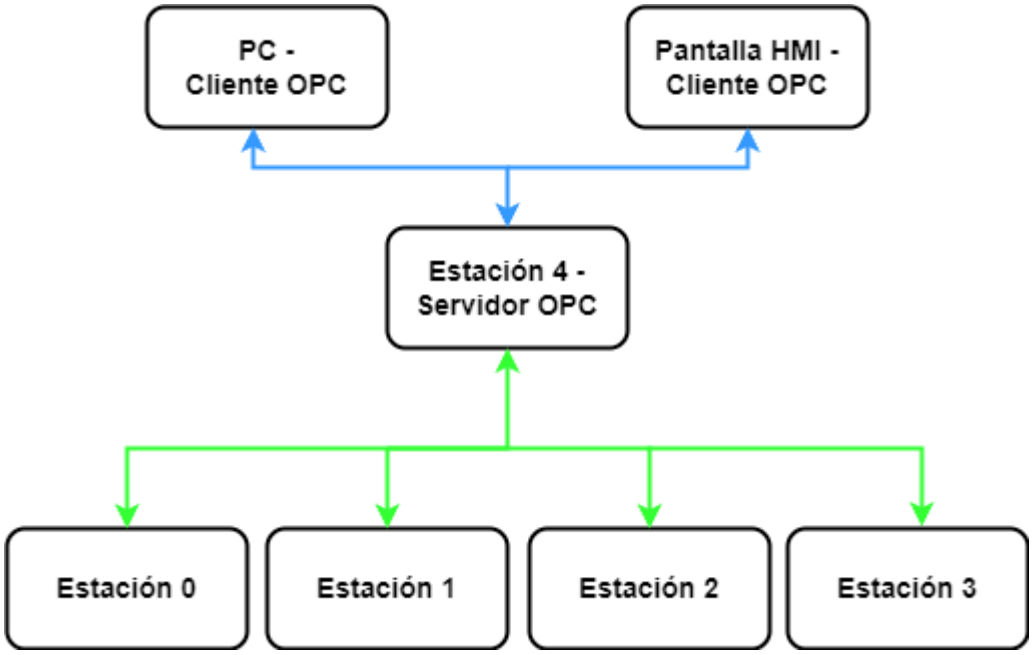



*Figura 9. Jerarquía de comunicación OPC*

## 5.1. Configuración Servidor OPC UA

El *S7-1500* de la estación *4*, puede actuar como cliente o como servidor *OPC*. En este servidor es importante tener en cuenta la versión del software de programación *TIA Portal* y el firmware del *PLC*.

Para implementar el servidor *OPC* en el autómata *S7-1500* es necesario actualizar el firmware, ya que los autómatas *S7-1500* del laboratorio cuentan actualmente con la versión 2.6.0 y el servidor requiere como mínimo 2.8.0.

En este caso, se actualiza el firmware desde el *TIAPortal*, aunque también es posible realizarlo desde el propio autómata. Lo primero es bajarse el firmware desde la página oficial de Siemens, para ello tenemos en cuenta que la referencia del autómata es *"6ES7 516-3AN01-0AB0"*.

Una vez descargado el archivo y el *PLC* esté conectado en la misma red local, se acede al menú: "Online y diagnostico" > "Funciones" > "Actualización de firmware". Se abre el fichero del firmware e iniciamos la actualización.

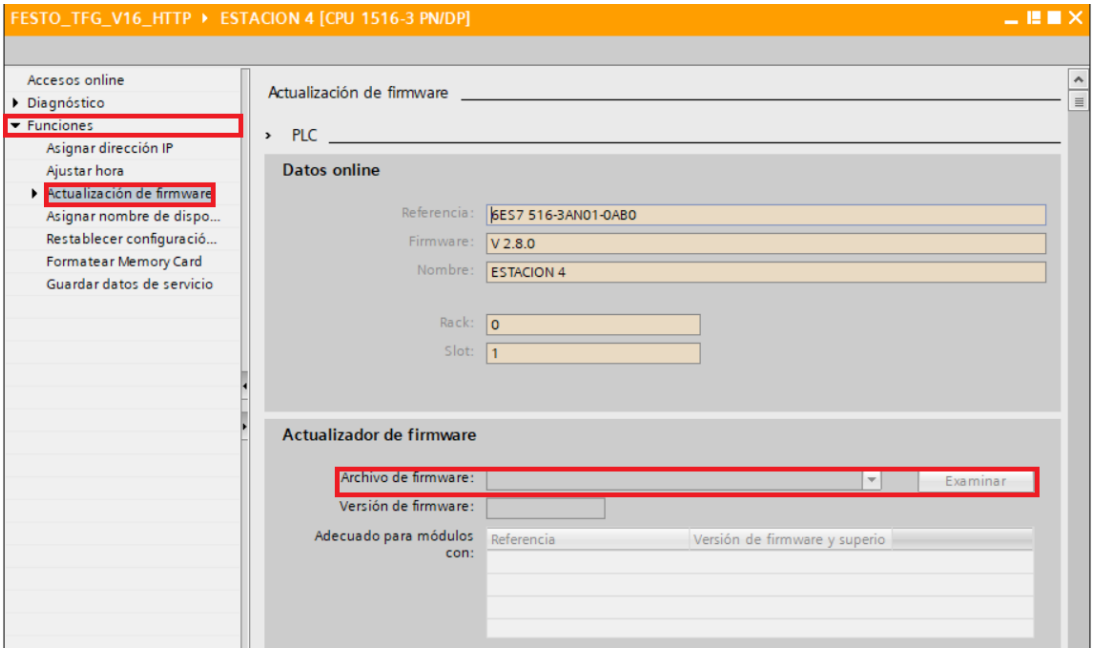



*Figura 10. Actualización Firmware*

Tras finalizar la actualización, se procede a habilitar el *OPC* Server en el dispositivo. Para ello seguimos la siguiente ruta: "Configuración de dispositivo" > "Propiedades" > "*OPC UA*" > "Servidor" > "General" > "Activar servidor *OPC UA*".

Por otro lado, la dirección de servidor nos permite la comunicación con un cliente *OPC*, el cual tiene asignado el número de puerto de 4840.

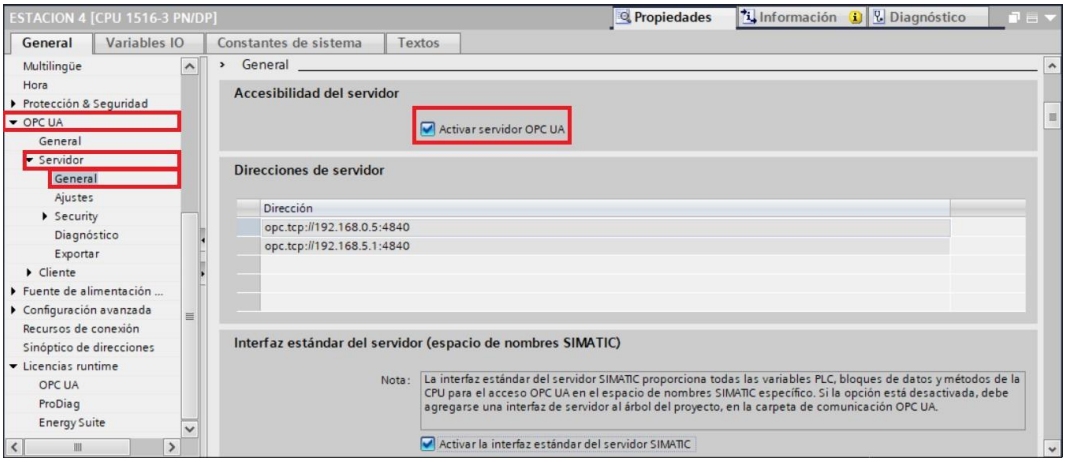



*Figura 11. Activación OPC UA*

Para trabajar con *OPC* UA en este *PLC*, es necesario activar la licencia de tiempo de ejecución (runtime license). Esta licencia autoriza la ejecución, operación y supervisión de los procesos industriales. El autómata requiere al menos una licencia *SIMATIC OPC UA S7-1500* medium.

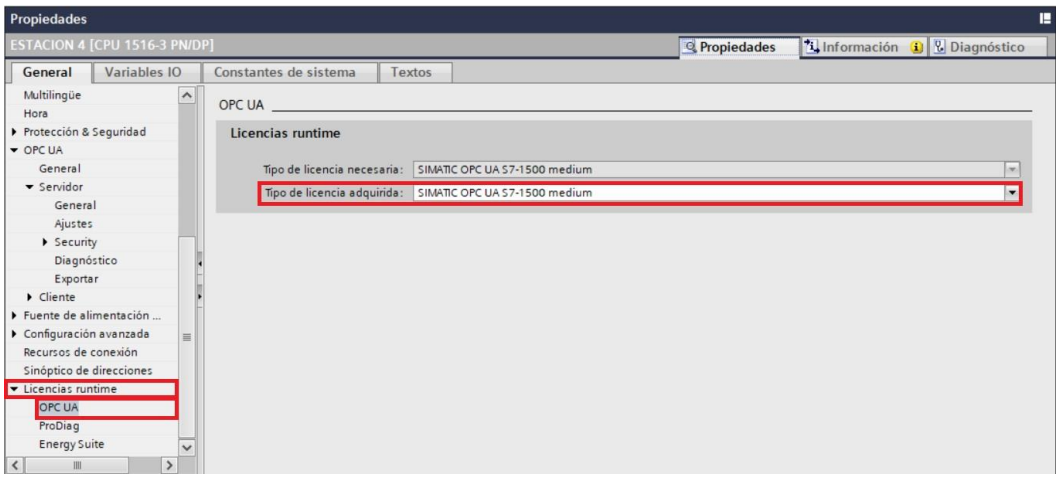



*Figura 12. Licencia runtime OPC UA*

El servidor *OPC UA* necesita un bloque de datos (*DB*) para almacenar y organizar la información que será compartida con los clientes. En el contexto de este proyecto, el *DB* se genera en el autómata de la estación *4*. En este bloque de datos se almacenan todas las variables de cada una de las estaciones, es decir, tendremos los valores de las entradas, salidas, marcas, contadores y temporizadores de toda la planta *FESTO*. El *DB* se actualiza en tiempo real a medida que se producen cambios en las variables, esto asegura que los clientes del servidor *OPC UA* siempre tengan acceso a los datos más actualizados de la *FESTO*

## 5.2. Bloque de datos para la comunicación

Para el control y almacenamiento de los datos de todos los autómatas se ha generado un bloque de datos (*DB*). En este caso se ha empleado un *DB Global*. Una base de datos global con la que se pueden almacenar y organizar diferentes tipos de información de manera centralizada. En el contexto de *TIAPortal*, la *DB Global* se utiliza para compartir datos entre diferentes bloques de programa.

La ventaja de utilizar una *DB Global* en *TIAPortal* es que permite una gestión centralizada de los datos compartidos. En lugar de tener que duplicar y mantener variables en múltiples bloques de programa, se pueden almacenar en una *DB Global* y acceder a ellas desde diferentes partes del proyecto. Esto facilita la coordinación y el intercambio de información entre los diferentes componentes del sistema de automatización.

Además, la *DB Global* también puede ser utilizada para compartir datos entre distintos dispositivos. Esto permite la sincronización y la comunicación de datos entre diferentes sistemas conectados, mejorando la eficiencia y la flexibilidad de la automatización.

El bloque de datos que contienen la información de todos los autómatas y el cual está habilitado a transmitirlo mediante el protocolo *OPC UA* es el creado por el ingeniero Alexander Epifanio Corona Ledesma [8] en su Trabajo de Fin de Grado (*TFG*) titulado "*Desarrollo de sistema multipantalla SCADA para la automatización de células de fabricación*", 2022.

El siguiente paso consiste en agregar el nuevo servidor al *PLC S7-1500*. Dentro del menú desplegable de la estación *4*, se deben seguir los siguientes pasos: "Comunicación *OPC UA*" > "Interfaces del servidor" > "Agregar un nuevo interfaz".

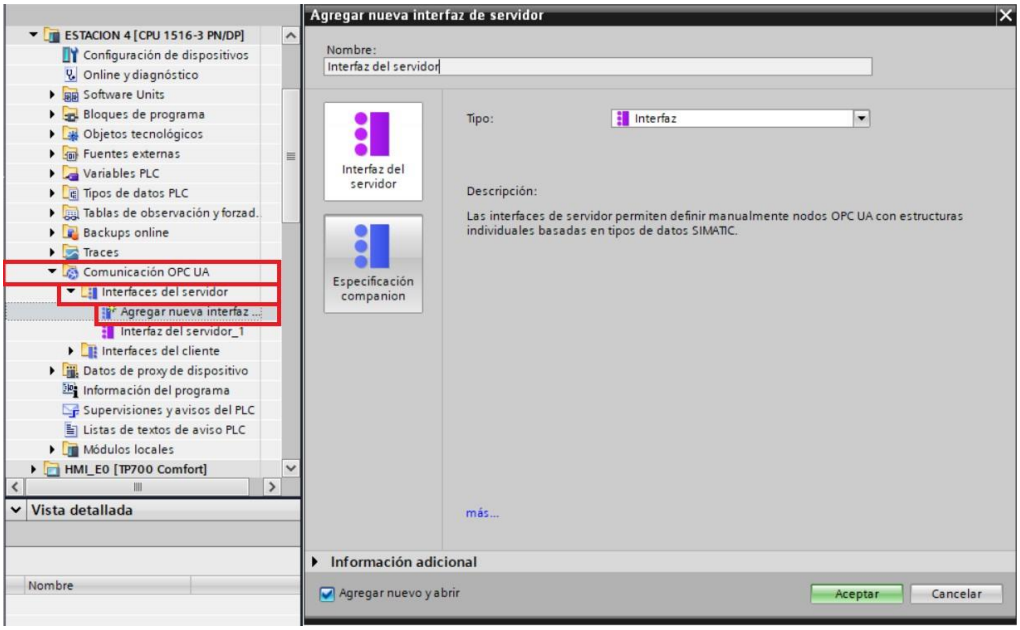



*Figura 13. Interfaz del servidor*

Una vez creado el Interfaz, se toman los elementos del bloque de datos (*DB*) y se adicionan a la interface.

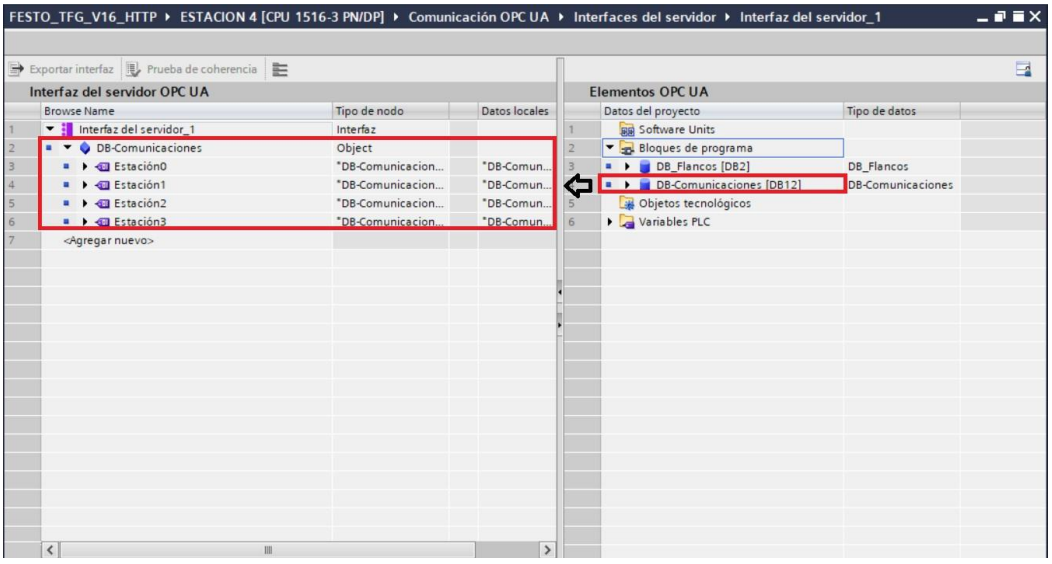



*Figura 14. Servidor OPC UA*

Para comprobar que la comunicación se establece correctamente, se procede a configurar un cliente *OPC UA* que leerá las variables que estamos enviando a través del protocolo servidor *OPC UA*. Para esta prueba, se utilizará el simulador de controladores lógicos programables *PLCSIM Advanced 3.0*, con el objetivo de establecer una conexión entre dos ordenadores: uno actuará como servidor y el otro como cliente. El propósito principal es verificar y validar el funcionamiento del servidor antes de su implementación en la planta *FESTO*.

En este simulador es importante tener en cuenta la configuración de las tarjetas de red para establecer conexión entre los dispositivos. En este caso, se tendrán tres tarjetas de red: una en el ordenador servidor, otra en el ordenador cliente y una tercera que será la de Siemens *PLCSIM Virtual Ethernet*.

Para que los dispositivos pueden comunicarse entre sí, es necesario asignar direcciones *IP* que se encuentren en la misma subred. Esto garantizará que estén en el rango de direcciones *IP* válidas y puedan establecer una conexión adecuada.

El primer objetivo es simular un *PLC* real utilizando *PLCSIM Advanced*. Para ello, se debe configurar en *PLCSIM Advanced* la misma dirección *IP* que ha sido asignada al autómata en el proyecto, así como la misma mascara subred. Además, debemos seleccionar el tipo de comunicación *TCP/IP*, que en este caso será con el ordenador que funcionará como servidor.

Finalmente, se carga el proyecto utilizando el adaptador del Siemens *PLCSIM Virtual Ethernet Adapter* (Ver ***Figura 15***), estableciendo conexión con el *PLC virtual*.

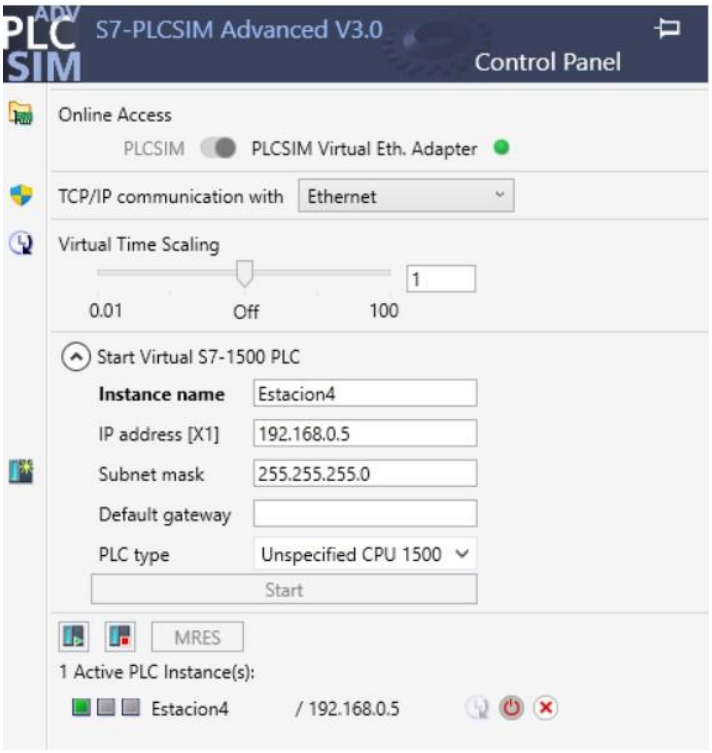



*Figura 15. Estación 4 PLCSIM Advanced*

## 5.3. Cliente OPC UA

EL *PLC* de la estación *4* se encarga de recibir y almacenar la información de las demás estaciones de la planta. Una vez recibida esta información, el *PLC* la almacena en el bloque de datos. Esta información está disponible para ser solicitada por el cliente *OPC UA*. El cliente podrá leer y modificar las variables de la planta *FESTO*, como cliente *OPC UA* se utilizará una aplicación especifica instalada en uno de los ordenadores del laboratorio permitiendo la transmisión de datos entre el cliente y el autómata que actúa como servidor.

Un cliente *OPC UA* es un software o aplicación que se utiliza para conectarse y comunicarse con servidores *OPC UA*. Su función principal es solicitar y recibir datos de los servidores *OPC UA*, así como interactuar con ellos para leer, escribir y monitorear variables, recibir notificaciones y ejecutar acciones en tiempo real. Los clientes *OPC UA* son utilizados en aplicaciones de automatización industrial, sistemas de control y supervisión, y en general, en entornos donde se requiere acceder a datos y funcionalidades

proporcionados por servidores *OPC UA*. Estos clientes facilitan la interoperabilidad y la comunicación segura entre los sistemas y dispositivos industriales que utilizan el estándar *OPC UA*.

### 5.3.1. UAExpert

*UAExpert* es una aplicación desarrollada por Unified Automation, una empresa especializada en soluciones *OPC UA*. Se trata de un cliente de prueba multiplataforma para *OPC UA*. Diseñado como un cliente de prueba de propósito general, *UAExpert* admite varias características del estándar *OPC UA*, como DataAccess, Alarmas y Condiciones y acceso histórico.

### 5.3.2. Configuración UAExpert

Una vez instalado el programa en el ordenador cliente, se procede a la configuración del servidor *OPC UA*. Para hacerlo, se sigue estos pasos:

1. Se abre el programa y se va al menú “File” (Archivo).
2. Selecciona “New Server Connection” (Nueva conexión de servidor”).
3. En la ventana que se abre, se introduce los parámetros de conexión requeridos, como la dirección *IP* y el puerto de comunicación. Estos datos son proporcionados por el administrador del servidor *OPC UA*.

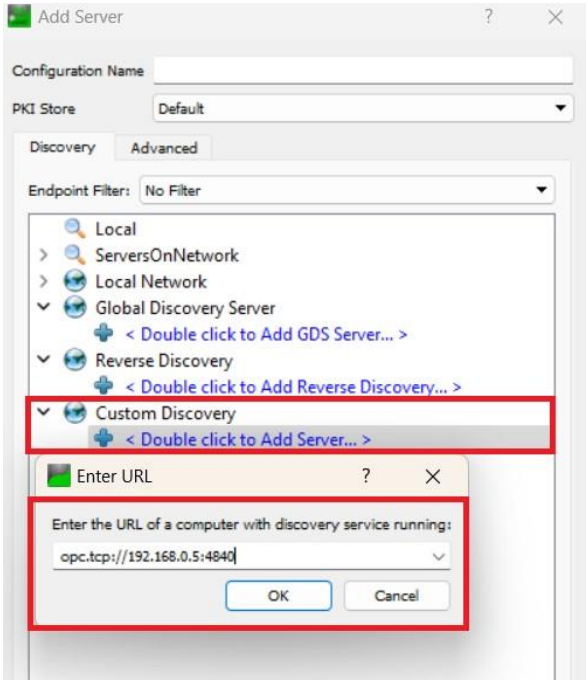


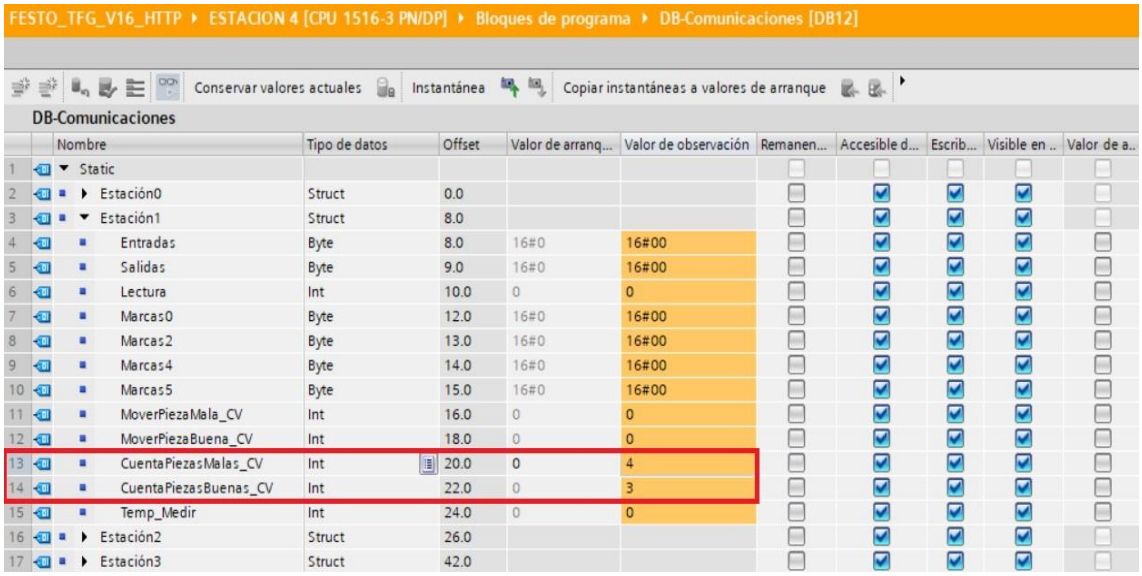

*Figura 16. Configuración* UA*Expert*

## 5.5. Comunicación entre Servidor OPC y Cliente OPC

### 5.5.1. Verificación de resultados Servidor OPC

Finalmente, se establece la comunicación entre ordenador servidor y el ordenador cliente, se transfiere los datos de las variables en ambos sentidos (bidireccionalmente). Esto permite que los datos sean intercambiados y actualizados entre el servidor y el cliente, garantizando una comunicación efectiva y la sincronización de la información.

Para realizar la prueba, se estableció un valor forzado de *4* en las piezas malas en el ordenador servidor, mientras que en el ordenador cliente se estableció un valor de *3* en las piezas buenas. Esto permitió simular y evaluar el comportamiento del sistema en diferentes escenarios, comprobando cómo se gestionan y procesan los datos correspondientes a las piezas malas en el servidor y los datos de las piezas buenas en el cliente.



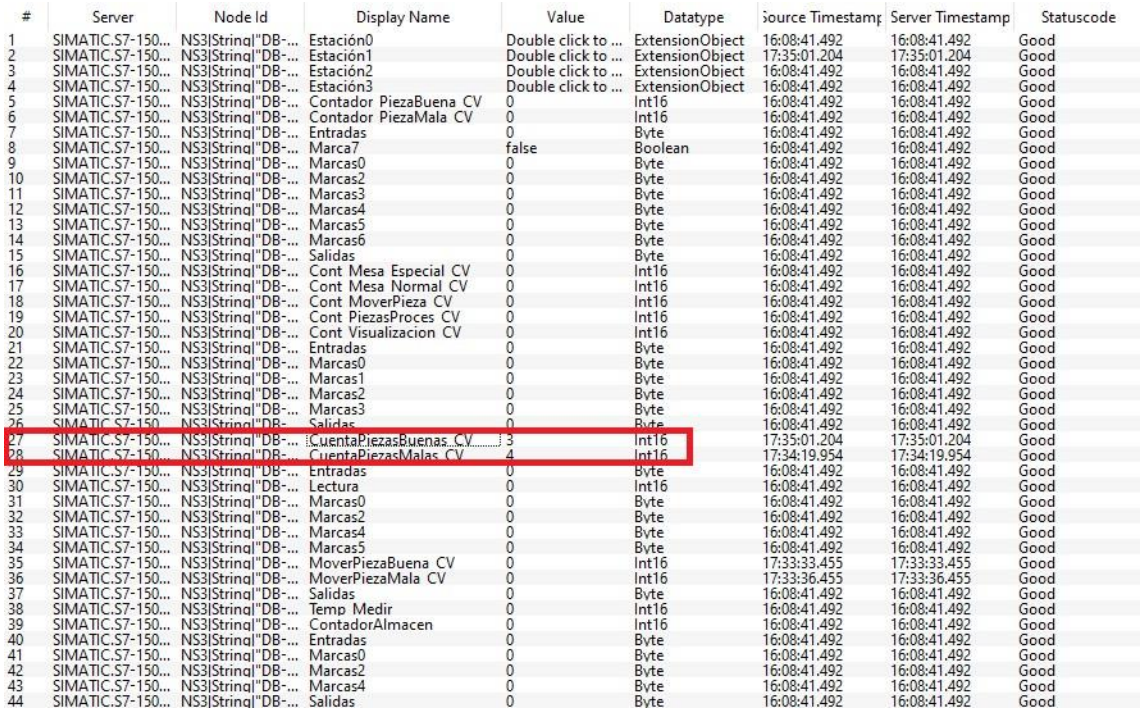
FESTO_TFG_V16_HTTP ▸ ESTACION 4 [CPU 1516-3 PN/DP] ▸ Bloques de programa ▸ DB-Comunicaciones [DB12]

Conservar valores actuales | Instantánea | Copiar instantáneas a valores de arranque

DB-Comunicaciones

| | Nombre | Tipo de datos | Offset | Valor de arranq... | Valor de observación | Remanen... | Accesible d... | Escrib... | Visible en ... | Valor de a... |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Static | | | | | | | | | |
| 2 | Estación0 | Struct | 0.0 | | | | | | | |
| 3 | Estación1 | Struct | 8.0 | | | | | | | |
| 4 | Entradas | Byte | 8.0 | 16#0 | 16#00 | | | | | |
| 5 | Salidas | Byte | 9.0 | 16#0 | 16#00 | | | | | |
| 6 | Lectura | Int | 10.0 | 0 | 0 | | | | | |
| 7 | Marcas0 | Byte | 12.0 | 16#0 | 16#00 | | | | | |
| 8 | Marcas2 | Byte | 13.0 | 16#0 | 16#00 | | | | | |
| 9 | Marcas4 | Byte | 14.0 | 16#0 | 16#00 | | | | | |
| 10 | Marcas5 | Byte | 15.0 | 16#0 | 16#00 | | | | | |
| 11 | MoverPiezaMala_CV | Int | 16.0 | 0 | 0 | | | | | |
| 12 | MoverPiezaBuena_CV | Int | 18.0 | 0 | 0 | | | | | |
| 13 | CuentaPiezasMalas_CV | Int | 20.0 | 0 | 4 | | | | | |
| 14 | CuentaPiezasBuenas_CV | Int | 22.0 | 0 | 3 | | | | | |
| 15 | Temp_Medir | Int | 24.0 | 0 | 0 | | | | | |
| 16 | Estación2 | Struct | 26.0 | | | | | | | |
| 17 | Estación3 | Struct | 42.0 | | | | | | | |

*Figura 17. PC servidor OPC*



| # | Server | Node Id | Display Name | Value | Datatype | Source Timestamp | Server Timestamp | Statuscode |
|---|---|---|---|---|---|---|---|---|
| 1 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Estación0 | Double click to ... | ExtensionObject | 16:08:41.492 | 16:08:41.492 | Good |
| 2 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Estación1 | Double click to ... | ExtensionObject | 17:35:01.204 | 17:35:01.204 | Good |
| 3 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Estación2 | Double click to ... | ExtensionObject | 16:08:41.492 | 16:08:41.492 | Good |
| 4 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Estación3 | Double click to ... | ExtensionObject | 16:08:41.492 | 16:08:41.492 | Good |
| 5 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Contador PiezaBuena_CV | 0 | Int16 | 16:08:41.492 | 16:08:41.492 | Good |
| 6 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Contador PiezaMala_CV | 0 | Int16 | 16:08:41.492 | 16:08:41.492 | Good |
| 7 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Entradas | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 8 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Marca7 | false | Boolean | 16:08:41.492 | 16:08:41.492 | Good |
| 9 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Marcas0 | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 10 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Marcas2 | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 11 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Marcas3 | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 12 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Marcas4 | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 13 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Marcas5 | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 14 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Marcas6 | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 15 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Salidas | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 16 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Cont Mesa Especial_CV | 0 | Int16 | 16:08:41.492 | 16:08:41.492 | Good |
| 17 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Cont Mesa Normal_CV | 0 | Int16 | 16:08:41.492 | 16:08:41.492 | Good |
| 18 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Cont MoverPieza_CV | 0 | Int16 | 16:08:41.492 | 16:08:41.492 | Good |
| 19 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Cont PiezasProces_CV | 0 | Int16 | 16:08:41.492 | 16:08:41.492 | Good |
| 20 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Cont Visualizacion_CV | 0 | Int16 | 16:08:41.492 | 16:08:41.492 | Good |
| 21 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Entradas | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 22 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Marcas0 | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 23 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Marcas1 | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 24 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Marcas2 | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 25 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Marcas3 | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 26 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Salidas | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 27 | SIMATIC.S7-150... | NS3\|String\|"DB-... | CuentaPiezasBuenas_CV | 3 | Int16 | 17:35:01.204 | 17:35:01.204 | Good |
| 28 | SIMATIC.S7-150... | NS3\|String\|"DB-... | CuentaPiezasMalas_CV | 4 | Int16 | 17:34:19.954 | 17:34:19.954 | Good |
| 29 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Entradas | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 30 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Lectura | 0 | Int16 | 16:08:41.492 | 16:08:41.492 | Good |
| 31 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Marcas0 | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 32 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Marcas2 | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 33 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Marcas4 | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 34 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Marcas5 | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 35 | SIMATIC.S7-150... | NS3\|String\|"DB-... | MoverPiezaBuena_CV | 0 | Int16 | 17:33:33.455 | 17:33:33.455 | Good |
| 36 | SIMATIC.S7-150... | NS3\|String\|"DB-... | MoverPiezaMala_CV | 0 | Int16 | 17:33:36.455 | 17:33:36.455 | Good |
| 37 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Salidas | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 38 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Temp_Medir | 0 | Int16 | 16:08:41.492 | 16:08:41.492 | Good |
| 39 | SIMATIC.S7-150... | NS3\|String\|"DB-... | ContadorAlmacen | 0 | Int16 | 16:08:41.492 | 16:08:41.492 | Good |
| 40 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Entradas | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 41 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Marcas0 | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 42 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Marcas2 | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 43 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Marcas4 | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |
| 44 | SIMATIC.S7-150... | NS3\|String\|"DB-... | Salidas | 0 | Byte | 16:08:41.492 | 16:08:41.492 | Good |

*Figura 18. PC cliente OPC*

# 6. Servidor HMI HTTP

Para complementar el servidor *OPC*, se va a utilizar un servidor *HMI HTTP* que proporciona una interfaz gráfica de usuario basada en web. En este proyecto, se realizará un primer acercamiento a este sistema, lo que permitirá explorar sus características y evaluar su potencial en la optimización de los procesos. Esta implementación consiste en configurar un servidor *HMI HTTP* en la estación *2* de la planta *FESTO*, así como su cliente *HMI HTTP* correspondiente. El cliente *HMI HTTP* se comunicará con el servidor *HMI HTTP*, y este con el *PLC* que controla la estación. Esta combinación de servidores mejora la supervisión, el control y la eficiencia de los sistemas de automatización.

Un servidor *HMI HTTP* [10] (Hypertext Transfer Protocol) se refiere a un servidor que utiliza el protocolo de transferencia de hipertexto (*HTTP*) para proporcionar acceso y visualización de la interfaz hombre-máquina (*HMI*) a través de un navegador web.

El servidor *HMI HTTP* actúa como una plataforma que almacena y sirve páginas web con contenido gráfico y de control para interactuar con el sistema de automatización. Los usuarios pueden acceder a la interfaz *HMI* utilizando un navegador web estándar desde cualquier dispositivo compatible con conexión a Internet. En este caso se realizará un cliente *HTTP* que se comunique con el servidor.

El servidor *HMI HTTP* se encarga de procesar las solicitudes *HTTP* enviadas por los clientes y proporciona los contenidos gráficos, controles, alarmas y otra información visual relacionada con el sistema de automatización. El servidor *HMI HTTP* puede tener capacidades adicionales, como el registro de datos, la generación de informes y la gestión de usuarios.

Este enfoque basado en *HTTP* permite una mayor flexibilidad y accesibilidad, ya que los usuarios pueden interactuar con la interfaz *HMI* desde cualquier ubicación con conexión a Internet, utilizando dispositivos como ordenadores, tablets y móviles, sin necesidad de instalar software específico en cada dispositivo.

## 6.1. Sistemas Pc

Los sistemas *PC* en *TIAPortal* se refieren a la utilización de odenadores personales (*PC*) como plataforma de hardware para ejecutar y gestionar el software de programación y configuración de Siemens *TIAPortal*.

Los *sistemas PC* en *TIAPortal* son ordenadores personales que cumplen con los requisitos de hardware y software recomendados por Siemens para ejecutar el *TIAPortal*. Estas *PC* se utilizan como estaciones de ingeniería desde las cuales los ingenieros y programadores pueden diseñar y desarrollar proyectos de automatización utilizando el entorno *TIAPortal*.

Además de ejecutar el software *TIAPortal*, el sistema *PC* puede estar conectado a dispositivos de automatización y redes industriales para realizar tareas de programación, monitoreo, diagnóstico y visualización. También puede proporcionar conectividad a través de interfaces de comunicación industrial, como Profinet, Profibus o Ethernet/IP.

### 6.1.1. Configuración sistema PC

La idea es tomar la pantalla *HMI* de la estación 2 creada por Javier Rodriguez de la Rosa [5] y transferir las imágenes, plantillas, variables *HMI* y conexiones al *HMI RT* (runtime) del *sistema PC*. Para lograr este objetivo, el primer paso es configurar el *sistema PC*: “Agregar dispositivo” > “Sistemas PC” > “SIMATIC *HMI* Application” > “*WinCC RT Advanced*”.

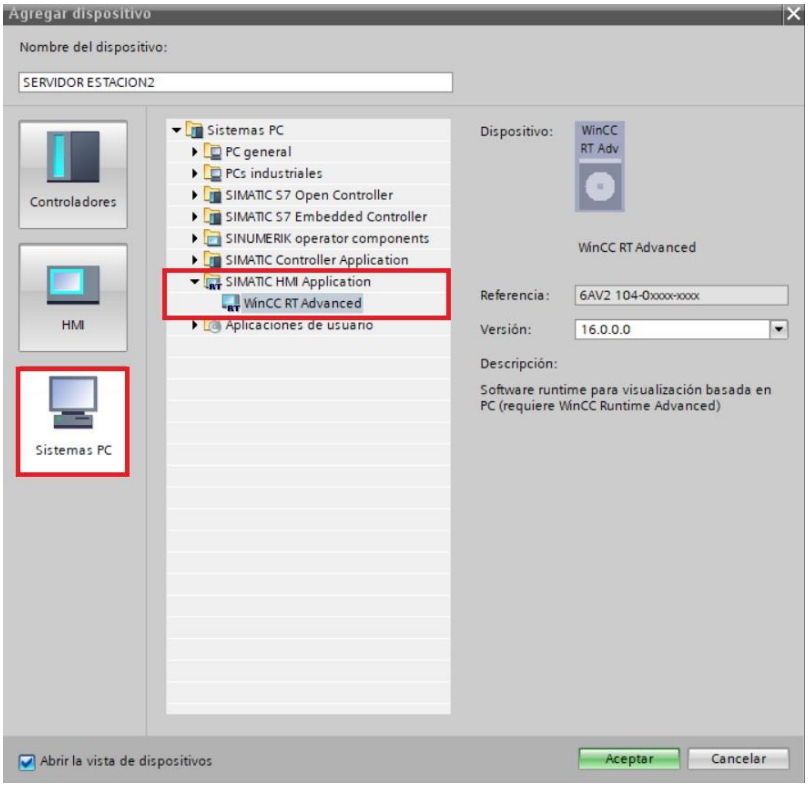



*Figura 19. Sistema PC*

Una vez que el sistema *PC* ha sido creado y se han transferido los datos del *HMI* de la estación *2*, es necesario habilitar el Servidor *HTTP Channel*. Para hacer esto, debemos acceder a la configuración de runtime y activar el servidor en la sección de servicios.

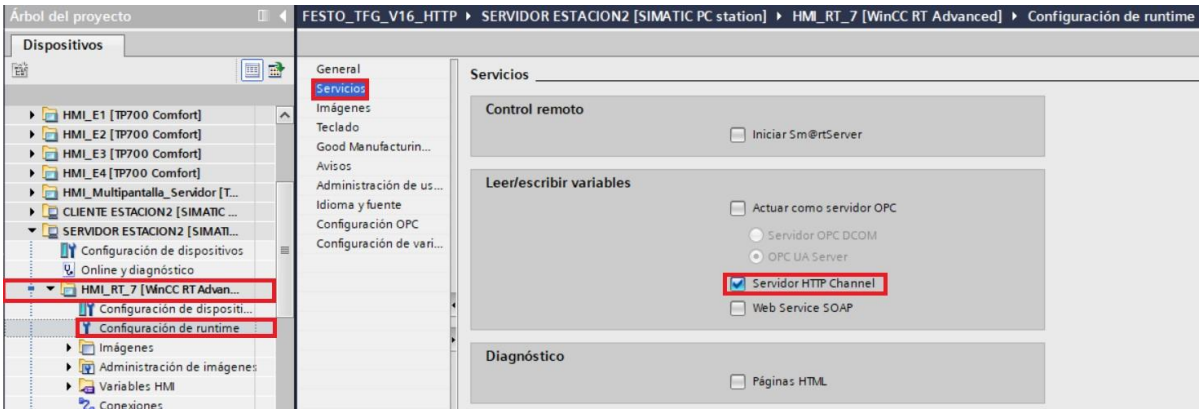



*Figura 20. Activación Servidor HTTP*

## 10.2. Cliente *HMI HTTP*

El cliente *HTTP* [10] se encarga de enviar la solicitud al servidor *HTTP*, las cuales pueden ser solicitudes para obtener información, enviar datos, actualizar o eliminar recursos, entre otras acciones definidas por el protocolo *HTTP*. El cliente será una copia exacta de la pantalla del Servidor *HMI HTTP*, se le añadirá un texto que especifiqué claramente cuál es el cliente y cuál es el servidor, para distinguirlos de manera inequívoca.

La conexión que se debe establecer en la planta sigue la siguiente configuración: el servidor *HMI HTTP* se conecta a la subred mediante una conexión *HMI*, es importante asegurarse de que el sistema *PC* tenga la misma dirección *IP* que el ordenador que se esté utilizando. En cambio, el Cliente *HTTP* no requiere ninguna conexión directa con el autómata. La conexión que se debe establecer en la planta es la siguiente, el servidor *HMI HTTP* se conecta mediante conexión *HMI* a la subred, en cambio el Cliente *HTTP* no requiere ningún tipo de conexión ya que no se conecta directamente con el autómata.

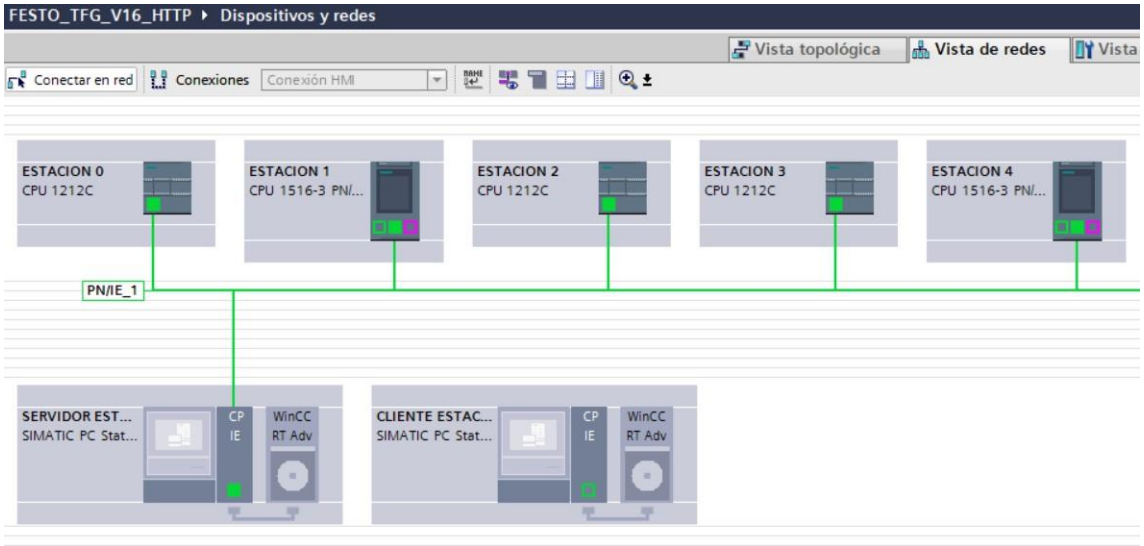



*Figura 21. Conexión Servidor HTTP*

### 10.2.1. Configuración Cliente *HMI HTTP*

#### 10.2.1.1. Conexión Cliente *HMI HTTP*

Se debe cambiar la conexión del cliente *HTTP* de *SIMATIC S7 1200* a *HTTP*, para ello, debemos realizar los siguientes pasos:

1. Acceder a la conexión de la pantalla *HMI* y cambiar la conexión de *SIMATIC S7 1200* a *HTTP*.

2. Además, para que el cliente *HTTP* puede comunicarse correctamente con el servidor, el nombre del servidor web del panel operador, debemos colocar el nombre del ordenador. En este caso, como trabajamos en el laboratorio Profesor Lorenzo Moreno Ruiz, los ordenadores tienen el nombre de "DESKTOP-TGS97DM".

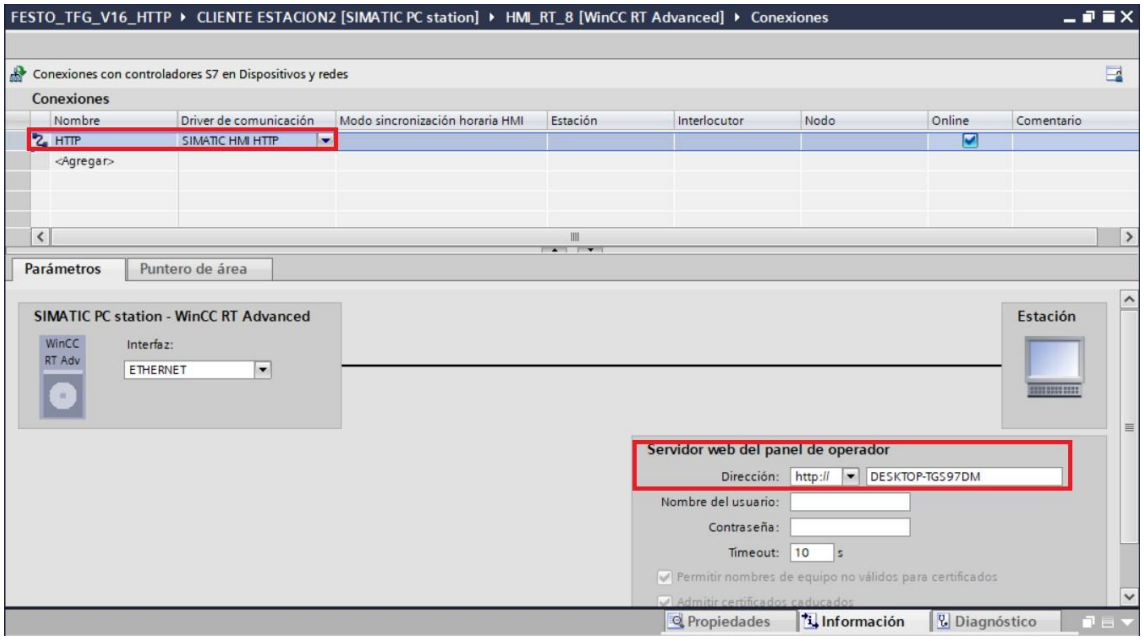



*Figura 22. Configuración conexión Cliente HTTP*

#### 10.2.1.2. Variables *HMI* Cliente *HMI HTTP*

Se debe modificar el tipo de conexión en las variables *HMI*, es necesario ajustarlas a la conexión *HTTP*. En el caso del cliente *HTTP*, dado que no se conecta directamente con el autómata, debemos modificar la dirección de cada variable. En lugar de utilizar la dirección del autómata, debemos sustituirla por el nombre específico de cada variable. Esta modificación asegurará que el cliente *HTTP* pueda acceder correctamente a las variables requeridas sin necesidad de una conexión directa con el autómata.

En la figura siguiente se muestra un ejemplo de cómo debería verse el resultado después de realizar estos cambios:


Tabla de variables estándar

| | Nombre | Tipo de datos | Conexión | Nombre del PLC | ... | Dirección |
|---|---|---|---|---|---|---|
| | ActivarTalad | Bool | HTTP | | ... | ActivarTalad |
| | AUX_Panel | Bool | HTTP | | ... | AUX_Panel |

*Figura 23. Variables HMI Cliente HTTP*

## 6.3. Transferir el sistema PC

Es necesario transferir el sistema PC al ordenador, para ello se utilizará el *WinCC Runtime Loader*. Este software es una herramienta especializada en el entorno de automatización industrial, diseñada para cargar y ejecutar aplicaciones de visualización y control creadas con el software *WinCC Runtime Advanced* o *WinCC Runtime* Professional. Su principal función es permitir la ejecución de proyecto de visualización en tiempo real, actuando como intermediario entre el hardware de la máquina y el software de visualización.

El *WinCC Runtime Loader* se encarga de cargar el proyecto de visualización en tiempo de ejecución, comunicarse con los controladores y dispositivos de campo, recopilar y procesar los datos de la planta, y proporcionar una interfaz de usuario interactiva para supervisar y controlar los procesos industriales. Además, ofrece funciones avanzadas como la gestión de alarmas, el registro de datos y la generación de informes.

La implementación implica los siguientes pasos:

1. Abrir el *WinCC Runtime Loader* en el ordenador.
2. Seleccionar la ruta de ubicación donde se desea crear el archivo.
3. Hacer clic en el botón "Transferir" para iniciar el proceso de transferencia.
4. Finalmente, cargar en el *TIAPortal* el sistema *PC* que contiene el cliente *HTTP*.

Una vez creado el archivo de transferencia, debemos proceder a transferirlo a otro ordenador que funcionará como cliente. Esta transferencia se realiza con el fin de comprobar el funcionamiento del servidor en dicho cliente.

## 6.4. Comunicación entre Servidor *HMI HTTP* y Cliente *HMI HTTP*

### 6.4.1. Verificación de resultados Servidor *HMI HTTP*

Para comprobar los resultados, realizamos los siguientes pasos: Iniciamos el runtime del servidor *HTTP* en el ordenador designado como servidor y, en el ordenador cliente, abrimos el archivo previamente creado y ejecutamos el simulador que emula la pantalla *HMI* del cliente.

Con ambos sistemas en funcionamiento, se llevó a cabo pruebas y verificaciones del correcto desempeño del sistema. Interactuamos con la interfaz del cliente para activar el cilindro de sujeción, bajar taladro y activar el motor taladro, lo que corresponde a la posición *2* (taladro de las piezas) de la estación *2* (ver ***Figura 26***). El servidor recibe está información y actualiza sus valores (ver ***Figura 25***), confirmando que las comunicaciones entre el servidor y el cliente se realizan correctamente.

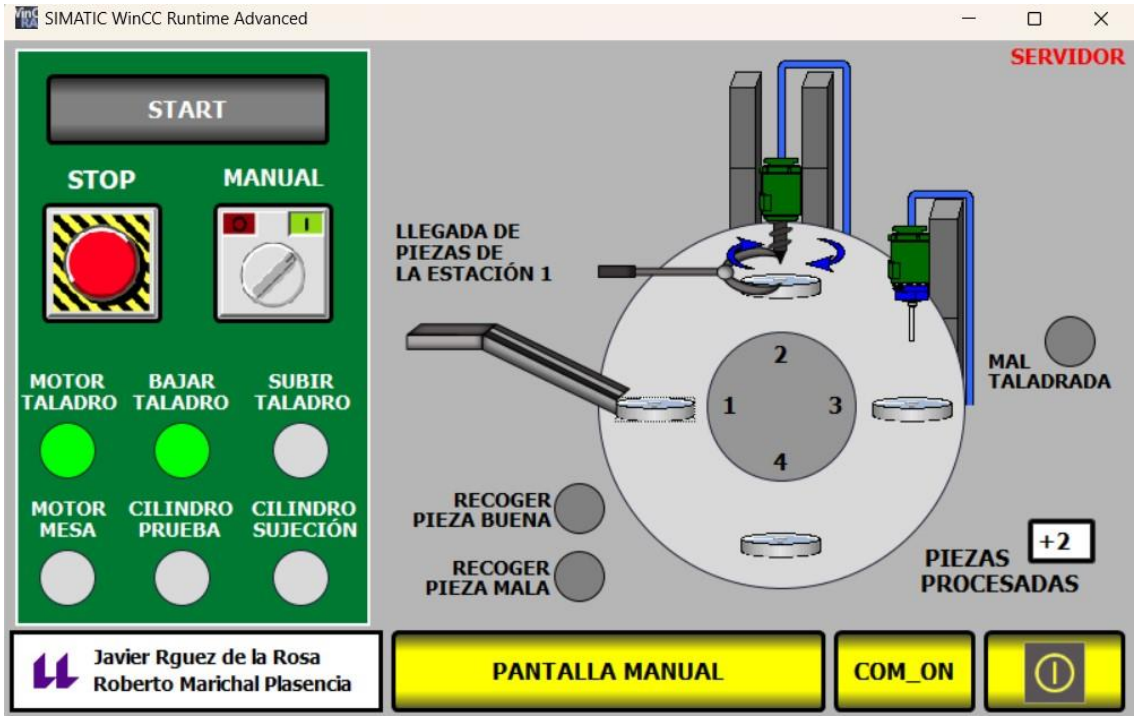



*Figura 24. Servidor HTTP*

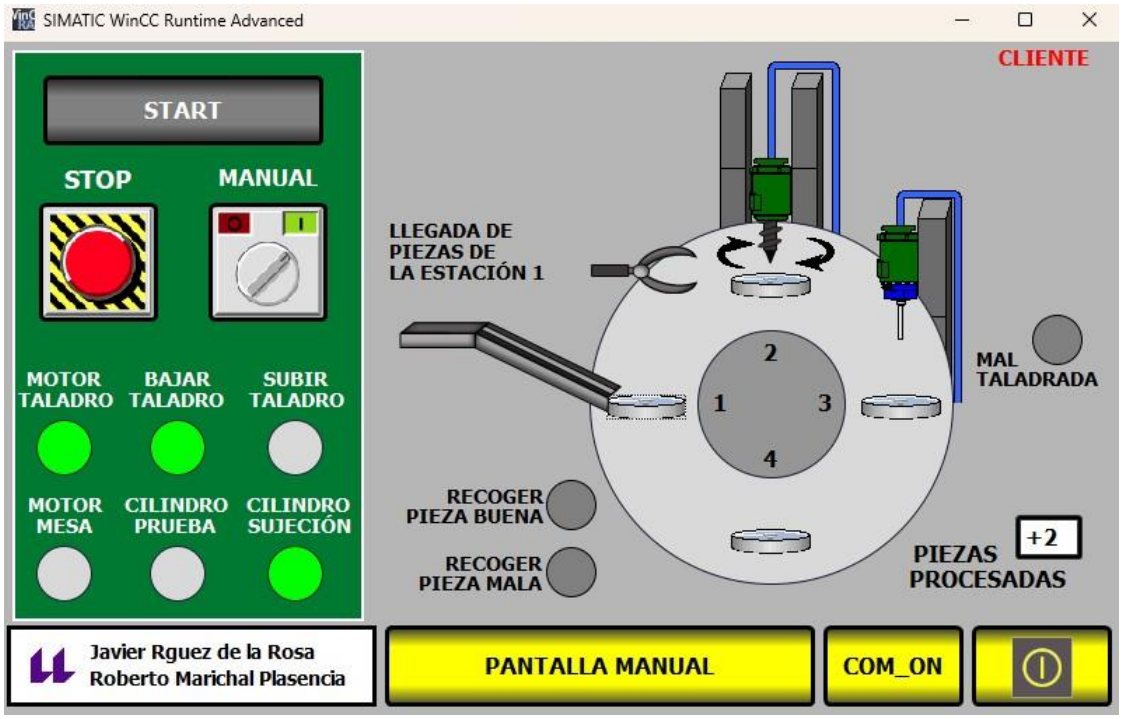



*Figura 25. Cliente HTTP*

### 6.4.2. Jerarquía de comunicaciones Servidor *HMI HTTP*

La jerarquía de un servidor *HMI HTTP* y un cliente *HMI HTTP* se puede entender de la siguiente manera:

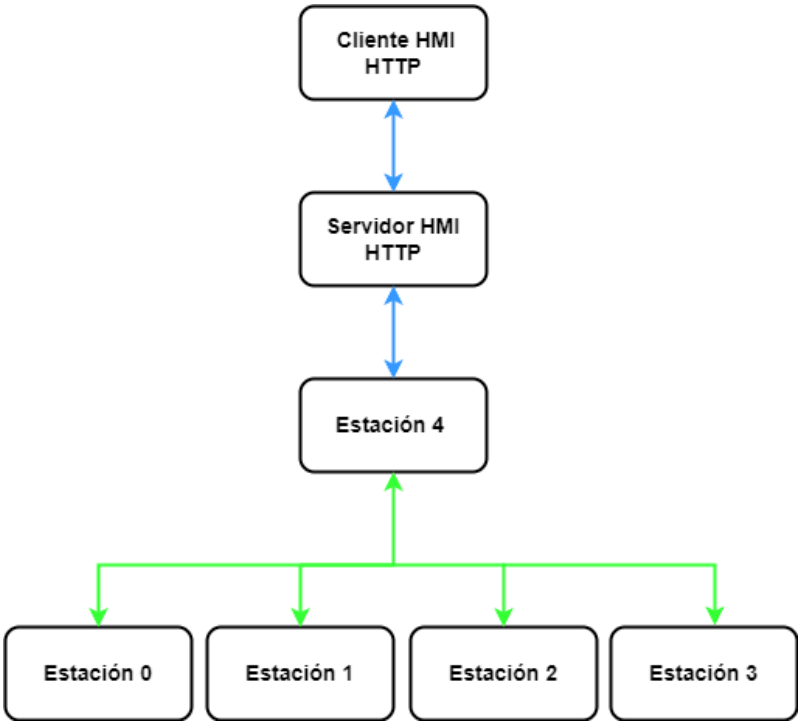



*Figura 26. Jerarquía comunicación HTTP*

Los clientes *HMI HTTP* son las entidades que acceden y utilizan la interfaz proporcionada por el servidor *HMI HTTP*. Estos clientes *HMI HTTP* pueden ser aplicaciones o dispositivos que se conectan al servidor para visualizar y controlar los procesos industriales. Los clientes pueden ser ejecutados en ordenadores, paneles táctiles, dispositivos móviles u otros dispositivos compatibles.

Por otro lado, El servidor *HMI HTTP* se encuentra en la parte superior de la jerarquía y actúa como la entidad central encargada de proporcionar y gestionar la interfaz de usuario y la visualización de datos. Este servidor *HMI HTTP* se comunica con la estación *4* de la planta *FESTO*, para recopilar información y enviar comandos. También es responsable de procesar los datos, generar alarmas, registrar eventos y llevar a cabo otras funciones relacionadas con la visualización y el control.

# 7. SmartServer

Para complementar y potenciar las capacidades de un servidor *OPC*, se puede utilizar el *SmartServer*. Este actúa como un puente entre los dispositivos de campo y el servidor *OPC*, facilitando una comunicación eficiente y confiable entre ellos.

En este proyecto, nos centraremos en la función de gestión remota para establecer una comunicación utilizando el protocolo *VNC* (Virtual Network Computing) con el fin de controlar la multipantalla *HMI* creada por Alexander Epifanio Corona Ledesma desde cualquier ordenador conectado a la subred del Laboratorio Profesor Lorenzo Moreno Ruiz.

El protocolo *VNC* se basa en la arquitectura cliente-servidor, donde el servidor *VNC* se ejecuta en la máquina que se desea controlar de forma remota (panel *HMI*), mientras que el cliente *VNC* se ejecuta en el equipo desde el cual se realiza la conexión remota (ordenador).

La comunicación en el entorno SmartServer sigue un diagrama de flujo similar al de un servidor *HTTP*. El cliente, en este caso el cliente *VNC*, se comunica con el Panel *HMI*, el cual actúa como servidor en la comunicación remota. A su vez, el panel *HMI* establece una conexión con la Estación *4*, que desempeña el papel de maestro en la jerarquía de comunicación de la planta. Esta conexión permite la transferencia de datos y el control de las demás estaciones en la planta.

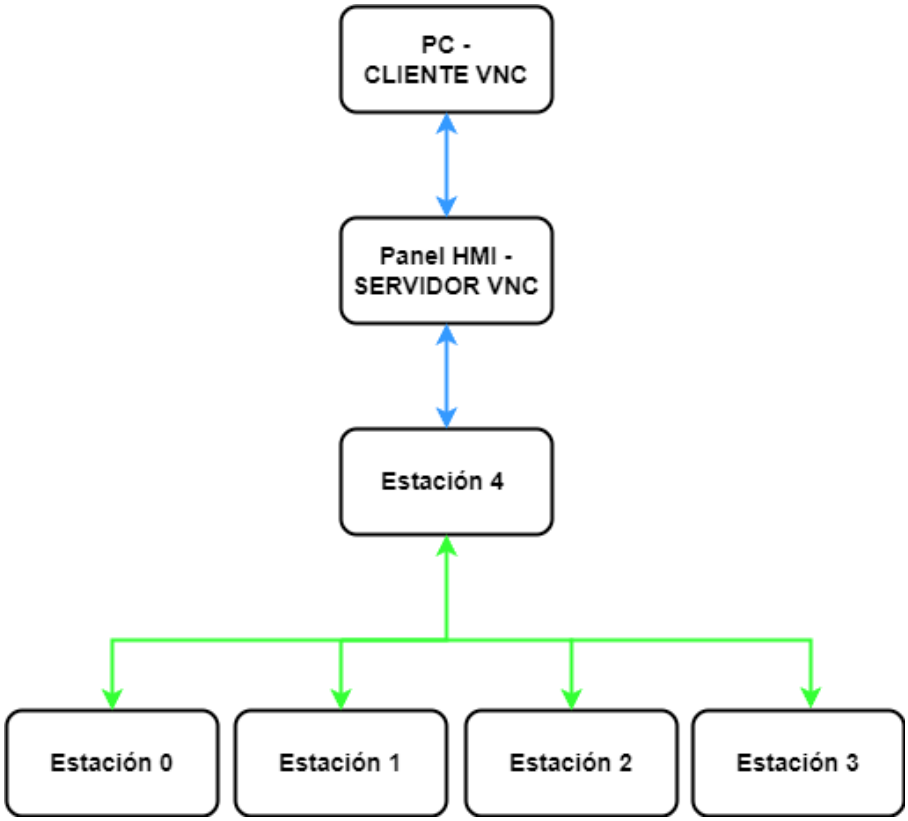



*Figura 27. Jerarquía de comunicación con SmartServer*

Un SmartServer [11]es un dispositivo de hardware y software que actúa como un servidor inteligente capaz de gestionar y controlar diversos aspectos de un entorno de automatización.

El SmartServer tiene la capacidad de recopilar datos de sensores, dispositivos y sistemas conectados, procesar y analizar dichos datos, y proporcionar servicios y funcionalidades avanzadas. Está diseñado para facilitar la conectividad, la interoperabilidad y la gestión eficiente de los dispositivos y datos en un entorno automatizado.

Entre las características comunes de un SmartServer se encuentran:

1. **Conectividad**: Puede soportar diferentes protocolos de comunicación, como *Modbus*, *OPC*, *BACnet*, *MQTT*, entre otros, para permitir la integración y comunicación con una amplia gama de dispositivos y sistemas.

2. **Procesamiento de datos**: Tiene capacidad de procesamiento y almacenamiento para realizar análisis en tiempo real, permitiendo la toma de decisiones basadas en datos y el control de los sistemas conectados.

3. **Seguridad**: Proporciona mecanismos de seguridad para proteger la integridad y confidencialidad de los datos, así como para controlar el acceso a los dispositivos y sistemas conectados.

4. **Interoperabilidad**: Permite la integración con otros sistemas y plataformas, como sistemas de gestión de edificios (*BMS*), sistemas *SCADA*, sistemas de gestión energética, entre otros, para crear un entorno de automatización completo y colaborativo.

5. **Gestión remota**: Permite la configuración, supervisión y control remoto de los dispositivos y sistemas conectados, facilitando la gestión centralizada y eficiente de la infraestructura.

## 7.1. Configuración SmartServer

Para habilitar el servicio de SmartServer en la multipantalla *HMI*, debemos seguir la siguiente ruta: “Configuración de runtime” > “Servicios” > “Iniciar SmartServer”. Al hacer esto, activamos la funcionalidad que nos permite visUAlizar la pantalla a través de la red.

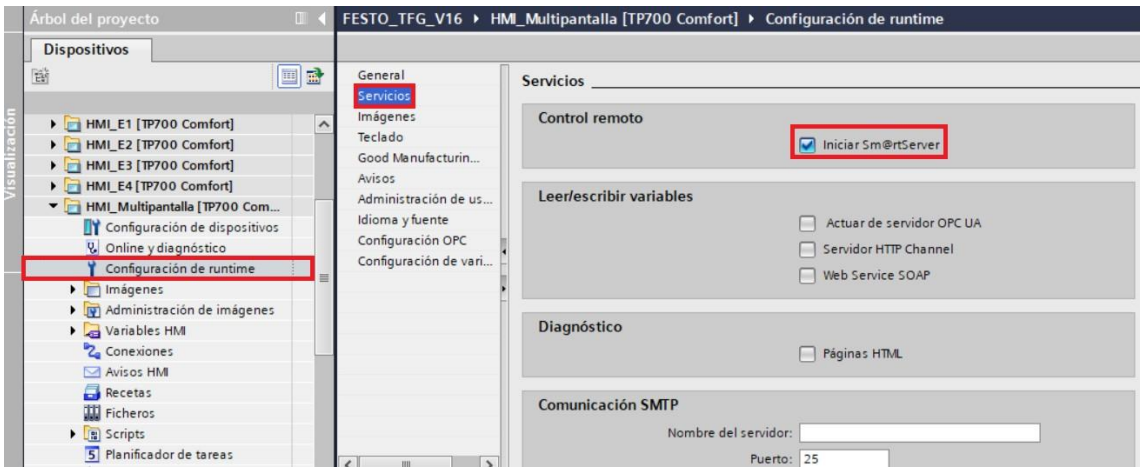



*Figura 28. Configuración SmartServer*

Una vez habilitado el servicio de SmartServer, debemos configuras la conexión remota en la pantalla física *HMI*. Para ello, seguimos los siguientes pasos:

1. Accedemos a "Setting" de la pantalla *HMI*.

2. Navegamos hasta "*WinCC* Internet Settings".

3. Dentro de esta sección, seleccionamos la *Opci*ón "Remote".

4. Antes de entrar en "Change Configuration", aseguramos de activar la pestaña "Start Automatically after Booting". Esto garantiza que el SmartServer se active cada vez que reiniciamos la pantalla.

5. En la sección de "Server", encontramos dos contraseñas:

   "Password 1" y "Password 2".

   - Activamos "Password 1" para permitir la interacción y control de la pantalla. Como medida de prueba, se asignará la contraseña "1234".

   - Por otro lado, "Password 2" viene activado por defecto y solo muestra la imagen de la pantalla sin permitir la interacción.

Con esta configuración, podemos establecer una conexión remota y controlar la pantalla *HMI* para realizar pruebas y ajustes necesarios.

## 7.2. SmartClient

SmartClient [11] es un término utilizado en el contexto de los sistemas de automatización industrial y las soluciones de supervisión y control. Se refiere a un cliente de software que se utiliza para visualizar los datos y gráficos generados por un sistema de control o supervisión.

En un entorno de automatización, el SmartClient permite a los usuarios acceder de forma remota a un sistema de control desde diferentes dispositivos, como ordenadores, tablets o teléfonos móviles. El SmartClient ofrece una interfaz de usuario intuitiva y funcionalidades que permiten visualizar y controlar los procesos en tiempo real.

Se conecta a un servidor que almacena y procesa los datos de control y supervisión. A través del SmartClient, los usuarios pueden monitorizar variables, recibir alarmas y notificaciones, interactuar con gráficos y pantallas de operación, y realizar ajustes y modificaciones en el sistema de control.

### 7.2.1. Configuración SmartClient

Para establecer una conexión remota con el panel *HMI*, debemos abrir la aplicación SmartClient e ingresar la dirección *IP* del *HMI* al cual se ha activado el SmartServer. En este caso, se trata del *HMI* multipantalla. Luego, procedemos a conectar la aplicación con el *HMI* e introducir la contraseña para establecer la comunicación remota y acceder a las funciones de control y visualización que proporciona el *HMI*.

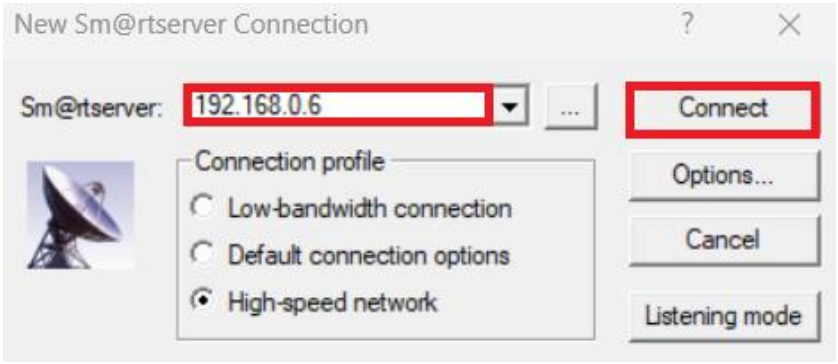



*Figura 29. Dirección de la Multipantalla HMI*

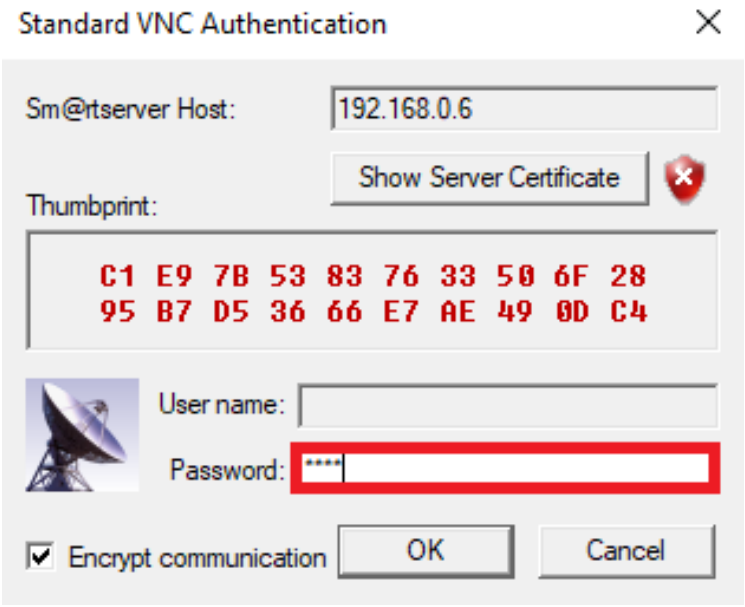



*Figura 30. Comunicación remota*

## 7.3. Comunicación entre el SmartClient y SmartServer

### 7.3.1. Verificación de resultados del SmartServer

Como resultado, podemos utilizar cualquier ordenador conectado a la misma red para establecer una conexión con el panel *SIMATIC HMI TP700 Comfort Panel* (*ver* ***Figura 31***). La gestión remota nos permitirá acceder y controlar el panel *HMI* de forma remota, sin necesidad de estar físicamente cerca del mismo. Esta funcionalidad ha sido de especialmente útil para solucionar problemas de la planta *FESTO*, como por ejemplo

corregir un error en la medida en la estación 1, y para probar nuevas funcionalidades, como la de colocar automáticamente la mesa en posición correcta en la estación 2.

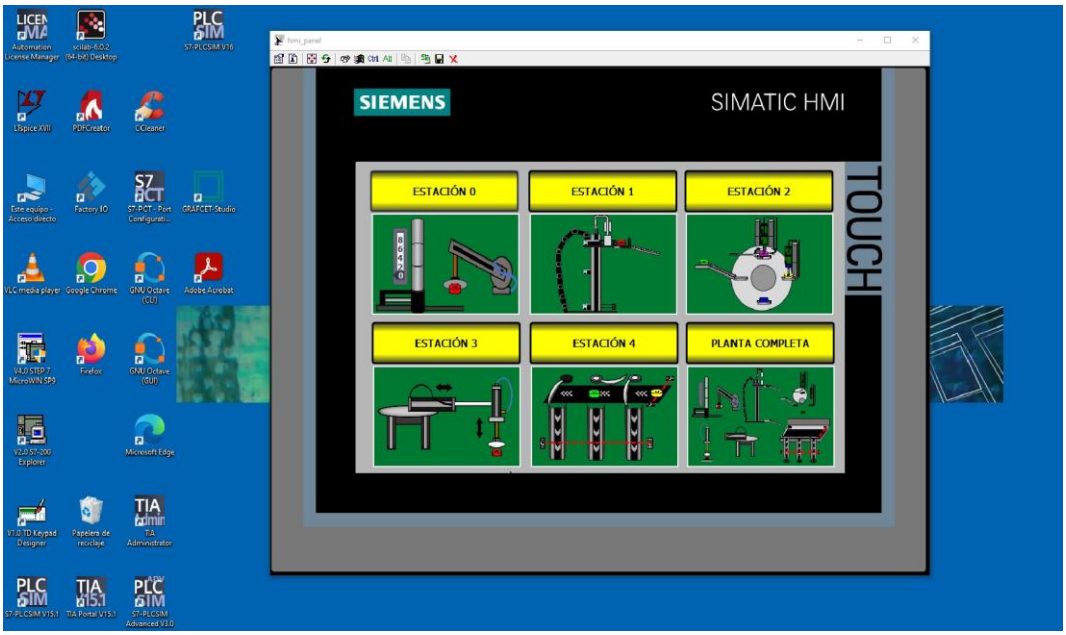



*Figura 31. Cliente VNC*

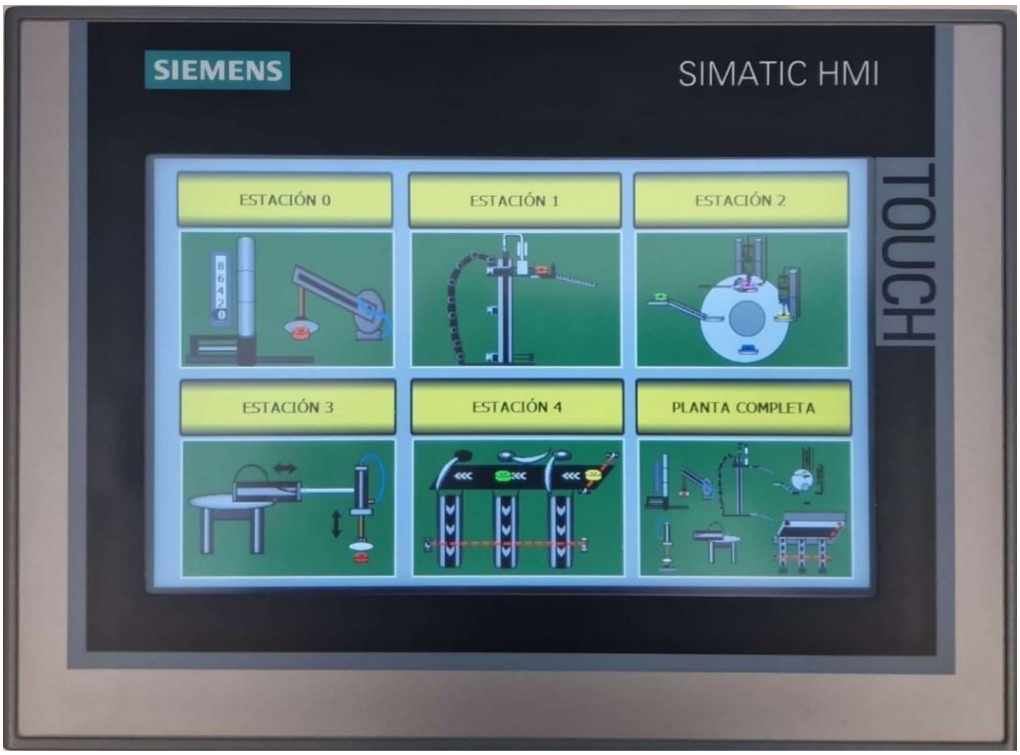



*Figura 32. Servidor Pantalla HMI*

# 8. Presupuesto

| Material | Unidades (u) | Precio Unitario(€/u) | Precio (€) |
|---|---|---|---|
| PLC S7-1500. Modelo CPU 1516-3 PN/DP (con módulos incluidos) | 2 | 1400 | 2800 |
| PLC S7-1200. Modelo CPU 1212C AC/DC/Rly (sin módulos) | 3 | 230 | 690 |
| Módulo SM1223 DI8/DQ8 x relé (S7-1200) | 3 | 175 | 525 |
| Módulo de comunicaciones PROFINET/PROFIBUS | 3 | 350 | 1050 |
| HMI TP700 Comfort Panel | 1 | 600 | 600 |
| Switch D-LINK | 1 | 25 | 25 |
| SIMATIC WinCC Advanced con TIA Portal integrado | 1 | 2930 | 2930 |
| SIMATIC S7-PLCSIM Advanced V3.0 | 1 | 3460 | 3460 |
| Horas de trabajo invertidas | 300 | 20 | 6000 |
| **TOTAL** | **-** | **-** | **18080** |

*Tabla 3. Presupuesto*

# 9. Conclusión

This final degree project focuses on the development and implementation of an *OPC* server in the *SCADA* system of the *FESTO* plant. To ensure the proper functioning of the *SCADA* system, it is essential to collect and store the information generated by each station. For this purpose, the *S7-1500 PLC* from station *4* was used as an *OPC UA* server. This *PLC* was responsible for transferring the collected information to a client, which could be a computer and/or an *HMI* screen. Thanks to the *OPC* server protocol, a bidirectional communication was established, allowing the reading and writing of data in both directions.

To conduct *OPC* protocol testing, the *PLCSIM Advanced V3.0* simulator was used to simulate real-time operation of the *PLC* in station *4*. This allowed for the identification and correction of operational errors prior to implementation in the actual plant.

Furthermore, in this project, the implementation of an *HMI HTTP* server was carried out as a complement to the *OPC* server. This implementation was performed on the *HMI* screen of station *2*, developed by Javier Rodríguez de la Rosa. To establish communication with the *HMI HTTP* client, the client's connection system was modified to use the *HTTP* protocol. Lastly, WinCC Runtime Advanced was employed to simulate the server-client screens on the laboratory computers, enabling the verification of proper communication.

Additionally, to enhance the capabilities of the *OPC* server, the SmartServer's remote management function based on the *VNC* protocol was utilized. This protocol allowed the control of the multi-screen *HMI* developed by Alexander Epifanio Corona Ledesma. To enable this functionality, the *HMI* panel was configured as a *VNC* server, and the corresponding settings were made in the *TiaPortal* project. Through this remote connection, any computer connected to the subnetwork of the Professor Lorenzo Moreno Ruiz Laboratory could act as a client.

In conclusion, simultaneous work with the *OPC* server and the *VNC* server was achieved. The real-time operation of the *FESTO* plant could be controlled from one

computer using the *VNC* protocol, while the plant information was displayed on another computer through the implementation of the *OPC* server.

# 10.Conclusión

Este trabajo de fin grado se centra en el desarrollo e implementación de un servidor *OPC* en el sistema *SCADA* de la planta *FESTO*. Para asegurar el correcto funcionamiento del sistema *SCADA*, es esencial recopilar y almacenar la información generada por cada estación. Para ello, se utilizó el *PLC S7-1500* de la estación *4* como servidor *OPC UA*. Este *PLC* se encargó de transferir la información recopilada en el bloque de datos (*DB*). El cliente, ya sea un ordenador y/o una pantalla *HMI* recibe dicha información y gracias al protocolo de servidor *OPC*, se establece una comunicación bidireccional que permite la lectura y escritura de datos en ambos sentidos.

Para realizar pruebas del protocolo *OPC*, se utilizó el simulador *PLCSIM Advanced V3*.0 para simular el funcionamiento en tiempo real del *PLC* de la estación *4*. Esto permitió corregir errores de funcionamiento antes de su implementación en la planta real.

Por otro lado, en este proyecto se llevó a cabo la implementación de un servidor *HMI HTTP* que actúa como complemento al servidor *OPC*. Esta implementación se realizó sobre la pantalla *HMI* de la estación *2* creada por Javier Rodríguez de la Rosa. Para establecer la comunicación con el cliente *HMI HTTP* se modificó el sistema de conexión del cliente al protocolo *HTTP*. Por último, se empleó el *WinCC Runtime Advanced* para simular las pantallas servidor-cliente en los ordenadores del laboratorio, lo que permitió verificar la correcta comunicación.

Finalmente, para complementar y potenciar las capacidades del servidor *OPC*, se utilizó la función de gestión remota que ofrece el SmartServer basado en el protocolo *VNC*. Este protocolo permitió controlar la multipantalla *HMI* desarrollada por Alexander Epifanio Corona Ledesma. Para habilitar esta funcionalidad, se configuró el panel *HMI* como servidor *VNC* y se realizaron las configuraciones correspondientes en el proyecto de *TiaPortal*. A través de esta conexión remota, cualquier ordenador conectado a la subred del Laboratorio Profesor Lorenzo Moreno Ruiz actúa como cliente.

Como conclusión, se logró trabajar simultáneamente con el servidor *OPC* y el servidor *VNC*. Se pudo controlar el funcionamiento real de la planta *FESTO* desde un ordenador

utilizando el protocolo *VNC*, mientras que en otro ordenador se visualizó la información de la planta gracias a la implementación del servidor *OPC*.

# 11.Plan de mejora

Como plan de mejora, sería beneficioso integrar un servidor web [12] en el sistema. *TIAPortal* ofrece la capacidad de crear y configurar páginas web que se ejecutan en un servidor web incorporado en el sistema. Estas páginas web pueden mostrar información en tiempo real, como datos de producción, alarmas, estado de dispositivos y permitir la interacción con el sistema de automatización. Incluso se podría incluir pantallas *HMI* programada en *HTML*.

El servidor web en *TIAPortal* proporciona una interfaz de usuario basada en web, lo que significa que se puede acceder y controlar el sistema de automatización desde cualquier dispositivo con un navegador web. Esta flexibilidad y facilidad de acceso permiten a los usuarios supervisar y controlar el sistema en cualquier momento y desde cualquier ubicación conveniente.

Además, el servidor web puede ofrecer funciones de seguridad como autenticación de usuarios y control de acceso para proteger el sistema de automatización contra accesos no autorizados. Esto garantiza la confidencialidad y la integridad de los datos y garantiza que solo las personas autorizadas tengan acceso al sistema.

# 12.Bibliografía


**[1]** **«COPADATA,» [En línea]. Available: *HTTP*s://www.copadata.com/es/productos/zenon-software-platform/vis**UA**lizacion-control/que-es-*SCADA*/.[Último acceso: Junio 2023]**

**[2]** **«*FESTO*,» [En línea]. Available: *HTTP*s://www.*FESTO*.com/es/es/. [Último acceso: Junio 2023].**

**[3]** **M. SIGUT SAAVEDRA y R. L. MARICHAL PLASENCIA, «Guiones de las estaciones *FESTO*.,» [En línea]. [Último acceso: Junio 2023].**

**[4]** **«SIEMENS,» [En línea]. Available: *HTTP*s://www.siemens.com/mx/es/productos/automatizacion/industry-software/automation-software/tia-portal/software.html. [Último acceso: Junio 2023].**

**[5]** **J. Rodríguez de la Rosa, «Repositorio Insitucional de la ULL (RIULL),» [En línea]. Available: *HTTP*s://riull.ull.es/xmlui/handle/915/24769. [Último acceso: Junio 2023].**

**[6]** **«SIEMENS,» [En línea]. Available: *HTTP*s://www.siemens.com/es/es/productos/automatizacion/sistemas/simatic/*HMI*.html. [Último acceso: Junio 2023].**

**[7]** **«Opiron,» [En línea]. Available: *HTTP*s://www.opiron.com/que-es-*OPC*-UA/. [Último acceso: Junio 2023].**

**[8]** **A. E. Corona Ledesma, «Repositorio Institucional de la ULL (RIULL),» [En línea]. Available: *HTTP*s://riull.ull.es/xmlui/handle/915/30315. [Último acceso: Junio 2023].**

**[9]** **«SIEMENS,» [En línea]. Available: *HTTP*s://support.industry.siemens.com/cs/document/109772889/descarga-del-simatic-s7-*PLC*sim-advanced-v3-0-de-prueba-(trial)?dti=0&lc=es-ES. [Último acceso: Junio 2023].**

**[10] «*PLC-HMI- SCADA*s,» [En línea]. Available: *HTTP*s://*PLC-HMI-SCADA*s.com/115.php. [Último acceso: Junio 2023].**

**[11] «SIEMENS,» [En línea]. Available: *HTTP*s://support.industry.siemens.com/cs/document/109476153/acceso-remoto-a-los-paneles-de-operador-simatic-*HMI*?dti=0&lc=es-UY. [Último acceso: Junio 2023].**

**[12] «SIEMENS,» [En línea]. Available: *HTTP*s://cache.industry.siemens.com/dl/files/560/59193560/att_109205/v1/s71500_webserver_function_manUAl_es-ES_es-ES.pdf. [Último acceso: Julio 2023].**

# 13.ANEXOS

# 13.1. ANEXO 1 Bloque DB de Comunicaciones

# DB-Comunicaciones [DB12]

| DB-Comunicaciones Propiedades | | | | | | | |
|---|---|---|---|---|---|---|---|
| **General** | | | | | | | |
| **Nombre** | DB-Comunicaciones | **Número** | 12 | **Tipo** | DB | **Idioma** | DB |
| **Numeración** | Automático | | | | | | |
| **Información** | | | | | | | |
| **Título** | | **Autor** | | **Comentario** | | **Familia** | |
| **Versión** | 0.1 | **ID personalizado** | | | | | |

| Nombre | Tipo de datos | Offset | Valor de arranque | Remanencia | Accesible desde HMI/OPC UA/Web API | Escribible desde HMI/OPC UA/Web API | Visible en HMI Engineering | Valor de ajuste | Supervisión | Comentario |
|---|---|---|---|---|---|---|---|---|---|---|
| ▼ Static | | | | | | | | | | |
| ▼ Estación0 | Struct | 0.0 | | False | True | True | True | False | | |
| Entradas | Byte | 0.0 | 16#0 | False | True | True | True | False | | |
| Salidas | Byte | 1.0 | 16#0 | False | True | True | True | False | | |
| Marcas0 | Byte | 2.0 | 16#0 | False | True | True | True | False | | |
| Marcas2 | Byte | 3.0 | 16#0 | False | True | True | True | False | | |
| Marcas4 | Byte | 4.0 | 16#0 | False | True | True | True | False | | |
| ContadorAlmacen | Int | 6.0 | 0 | False | True | True | True | False | | |
| ▼ Estación1 | Struct | 8.0 | | False | True | True | True | False | | |
| Entradas | Byte | 8.0 | 16#0 | False | True | True | True | False | | |
| Salidas | Byte | 9.0 | 16#0 | False | True | True | True | False | | |
| Lectura | Int | 10.0 | 0 | False | True | True | True | False | | |
| Marcas0 | Byte | 12.0 | 16#0 | False | True | True | True | False | | |
| Marcas2 | Byte | 13.0 | 16#0 | False | True | True | True | False | | |

| Nombre | Tipo de datos | Offset | Valor de arranque | Remanencia | Accesible desde HMI/OPC UA/Web API | Escribible desde HMI/OPC UA/Web API | Visible en HMI Engineering | Valor de ajuste | Supervisión | Comentario |
|---|---|---|---|---|---|---|---|---|---|---|
| Marcas4 | Byte | 14.0 | 16#0 | False | True | True | True | False | | |
| Marcas5 | Byte | 15.0 | 16#0 | False | True | True | True | False | | |
| MoverPiezaMala_CV | Int | 16.0 | 0 | False | True | True | True | False | | |
| MoverPiezaBuena_CV | Int | 18.0 | 0 | False | True | True | True | False | | |
| CuentaPiezasMalas_CV | Int | 20.0 | 0 | False | True | True | True | False | | |
| CuentaPiezasBuenas_CV | Int | 22.0 | 0 | False | True | True | True | False | | |
| Temp_Medir | Int | 24.0 | 0 | False | True | True | True | False | | |
| ▼ Estación2 | Struct | 26.0 | | False | True | True | True | False | | |
| Entradas | Byte | 26.0 | 16#0 | False | True | True | True | False | | |
| Salidas | Byte | 27.0 | 16#0 | False | True | True | True | False | | |
| Marcas0 | Byte | 28.0 | 16#0 | False | True | True | True | False | | |
| Marcas1 | Byte | 29.0 | 16#0 | False | True | True | True | False | | |
| Marcas2 | Byte | 30.0 | 16#0 | False | True | True | True | False | | |
| Marcas3 | Byte | 31.0 | 16#0 | False | True | True | True | False | | |
| Cont_Mesa_Especial_CV | Int | 32.0 | 0 | False | True | True | True | False | | |
| Cont_Mesa_Normal_CV | Int | 34.0 | 0 | False | True | True | True | False | | |
| Cont_MoverPieza_CV | Int | 36.0 | 0 | False | True | True | True | False | | |
| Cont_PiezasProces_CV | Int | 38.0 | 0 | False | True | True | True | False | | |
| Cont_Visualizacion_CV | Int | 40.0 | 0 | False | True | True | True | False | | |
| ▼ Estación3 | Struct | 42.0 | | False | True | True | True | False | | |
| Entradas | Byte | 42.0 | 16#0 | False | True | True | True | False | | |
| Salidas | Byte | 43.0 | 16#0 | False | True | True | True | False | | |
| Marcas0 | Byte | 44.0 | 16#0 | False | True | True | True | False | | |
| Marcas2 | Byte | 45.0 | 16#0 | False | True | True | True | False | | |
| Marcas3 | Byte | 46.0 | 16#0 | False | True | True | True | False | | |
| Marcas4 | Byte | 47.0 | 16#0 | False | True | True | True | False | | |

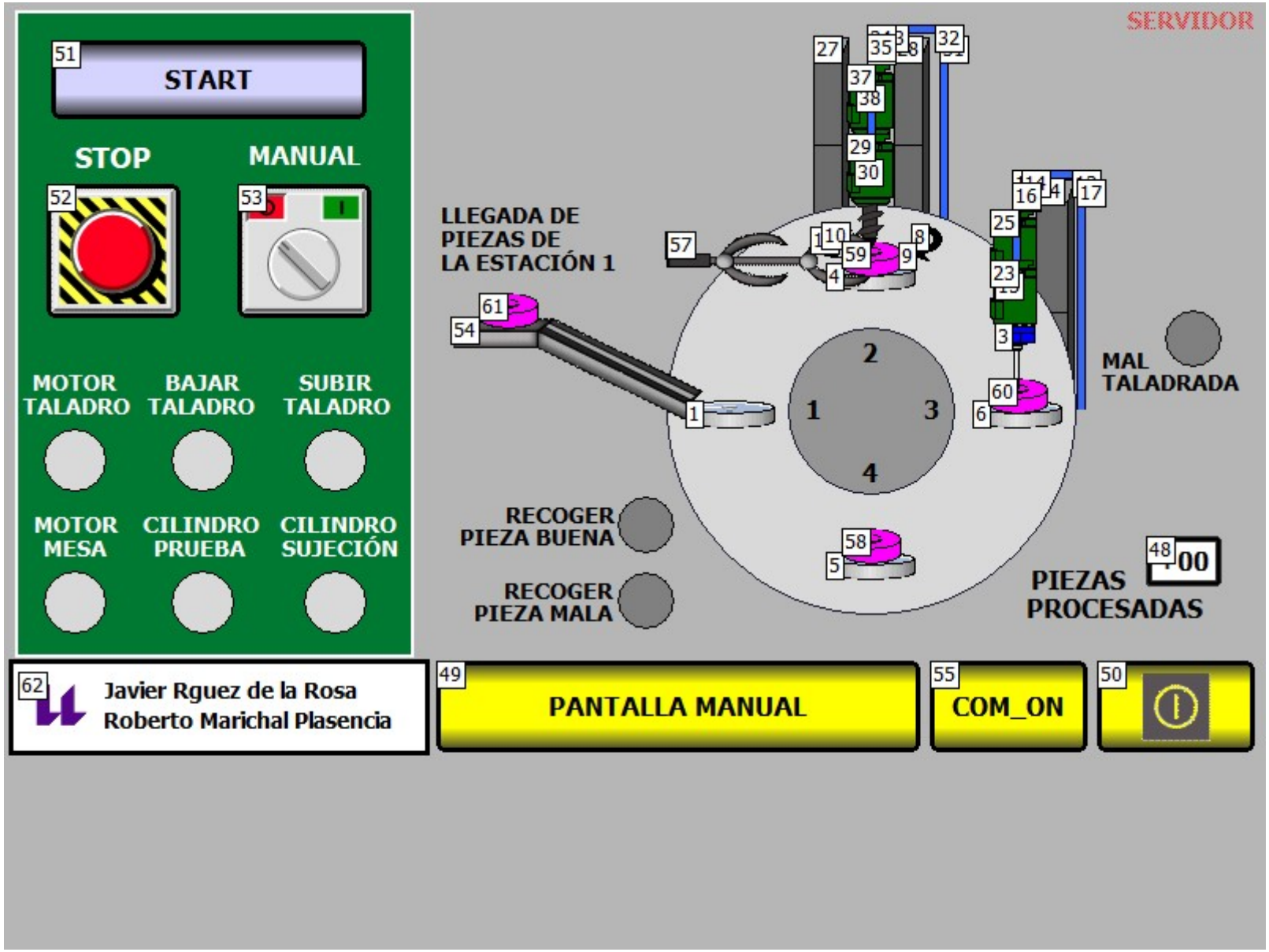
| Nombre | Tipo de datos | Offset | Valor de arranque | Remanencia | Accesible desde HMI/OPC UA/Web API | Escribible desde HMI/OPC UA/Web API | Visible en HMI Engineering | Valor de ajuste | Supervisión | Comentario |
|---|---|---|---|---|---|---|---|---|---|---|
| Marcas5 | Byte | 48.0 | 16#0 | False | True | True | True | False | | |
| Marcas6 | Byte | 49.0 | 16#0 | False | True | True | True | False | | |
| Contador_PiezaMala_CV | Int | 50.0 | 0 | False | True | True | True | False | | |
| Contador_PiezaBuena_CV | Int | 52.0 | 0 | False | True | True | True | False | | |
| Marca7 | Bool | 54.0 | false | False | True | True | True | False | | |

# 13.2. ANEXO 2

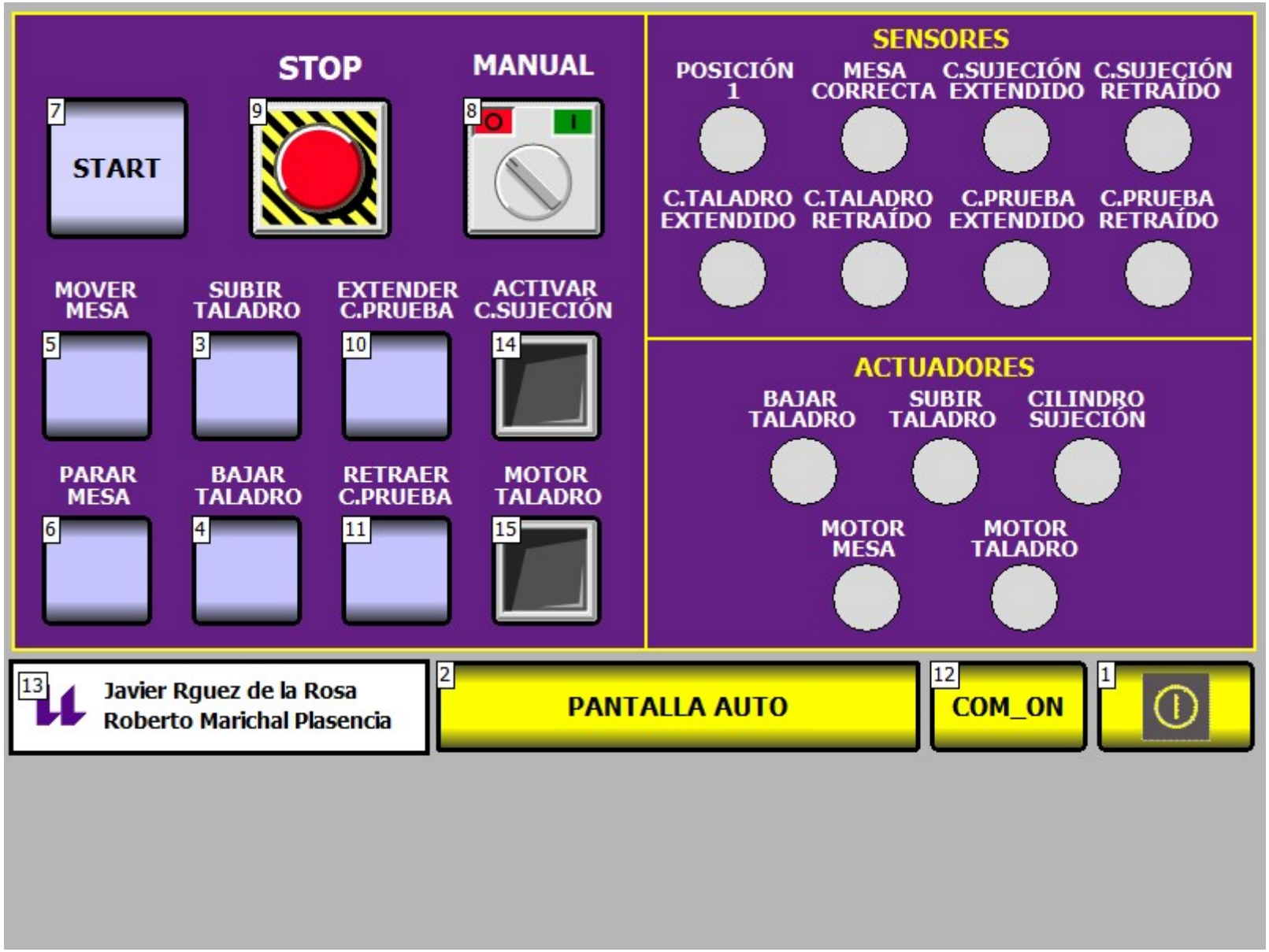
# Pantallas Servidor

# Estación 2

**Copia impresa de Pantalla_Auto**



**Copia impresa de Pantalla_Manual**



# 13.3. ANEXO 3

# Variables HMI

# Estación 2

# Tabla de variables estándar [47]

## ActivarTalad

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | ActivarTalad | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**ActivarTalad**

## AUX_Panel

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | AUX_Panel | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**AUX_Panel**

## BajarTaladro

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | BajarTaladro | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**BajarTaladro**

## COM_Off

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | COM_Off | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**COM_Off**

## Cont_Mesa_Especial.CV

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Cont_Mesa_Especial.CV | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Int | **Longitud** | 2 |

### Cont_Mesa_Normal.CV

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Cont_Mesa_Normal.CV | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Int | **Longitud** | 2 |

### Cont_MoverPieza.CV

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Cont_MoverPieza.CV | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Int | **Longitud** | 2 |

### Cont_PiezasProces.CV

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Cont_PiezasProces.CV | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Int | **Longitud** | 2 |

### Cont_Visualizacion.CV

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Cont_Visualizacion.CV | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Int | **Longitud** | 2 |

### CPExt

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | CPExt | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**CPExt**

### CPRet

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | CPRet | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**CPRet**

## CPrueba

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | CPrueba | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**CPrueba**

## CSExt

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | CSExt | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**CSExt**

## CSRet

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | CSRet | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**CSRet**

## CSujecion

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | CSujecion | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**CSujecion**

## CTExt

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | CTExt | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

## CTExt

### CTRet

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | CTRet | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

## CTRet

### Graf_pinza_sujec_1

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Graf_pinza_sujec_1 | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | <Variable interna> | **Tipo de datos** | Bool | **Longitud** | 1 |

### Graf_pinza_sujec_2

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Graf_pinza_sujec_2 | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | <Variable interna> | **Tipo de datos** | Bool | **Longitud** | 1 |

### Graf_rampa_pieza

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Graf_rampa_pieza | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | <Variable interna> | **Tipo de datos** | Bool | **Longitud** | 1 |

### Graf_taladro1

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Graf_taladro1 | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | <Variable interna> | **Tipo de datos** | Bool | **Longitud** | 1 |

### Graf_taladro2

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Graf_taladro2 | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | <Variable interna> | **Tipo de datos** | Bool | **Longitud** | 1 |

### Graf_ULL

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Graf_ULL | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | <Variable interna> | **Tipo de datos** | Bool | **Longitud** | 1 |

### HMI_ActTaladro

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | HMI_ActTaladro | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**HMI_ActTaladro**

### HMI_BajarTal

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | HMI_BajarTal | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**HMI_BajarTal**

### HMI_ExtPrueba

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | HMI_ExtPrueba | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**HMI_ExtPrueba**

### HMI_MoverMesa

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | HMI_MoverMesa | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**HMI_MoverMesa**

### HMI_PararMesa

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | HMI_PararMesa | **Nombre de visualización** | | **Dirección** | |

| | | | | | |
|---|---|---|---|---|---|
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**HMI_PararMesa**

### HMI_RetPrueba

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | HMI_RetPrueba | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**HMI_RetPrueba**

### HMI_START

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | HMI_START | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**HMI_START**

### HMI_STOP

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | HMI_STOP | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**HMI_STOP**

### HMI_SubirTal

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | HMI_SubirTal | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**HMI_SubirTal**

### HMI_Sujecion

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | HMI_Sujecion | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**HMI_Sujecion**

## MesaCorrecta

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | MesaCorrecta | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**MesaCorrecta**

## MODO_MANUAL

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | MODO_MANUAL | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**MODO_MANUAL**

## MotorMesa

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | MotorMesa | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**MotorMesa**

## MotorTaladro

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | MotorTaladro | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**MotorTaladro**

## Orden_PiezaBuena_E3

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Orden_PiezaBuena_E3 | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**Orden_PiezaBuena_E3**

## Orden_PiezaMala_E3

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Orden_PiezaMala_E3 | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**Orden_PiezaMala_E3**

## PARADA

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | PARADA | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**PARADA**

## Posicion1

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Posicion1 | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**Posicion1**

## Recibir_PiezaMala_E1

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Recibir_PiezaMala_E1 | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**Recibir_PiezaMala_E1**

## ResetNormal

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | ResetNormal | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**ResetNormal**

### STOP

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | STOP | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**STOP**

### SubirTaladro

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | SubirTaladro | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Bool | **Longitud** | 1 |

**SubirTaladro**

### Temp_MedirPieza.ET

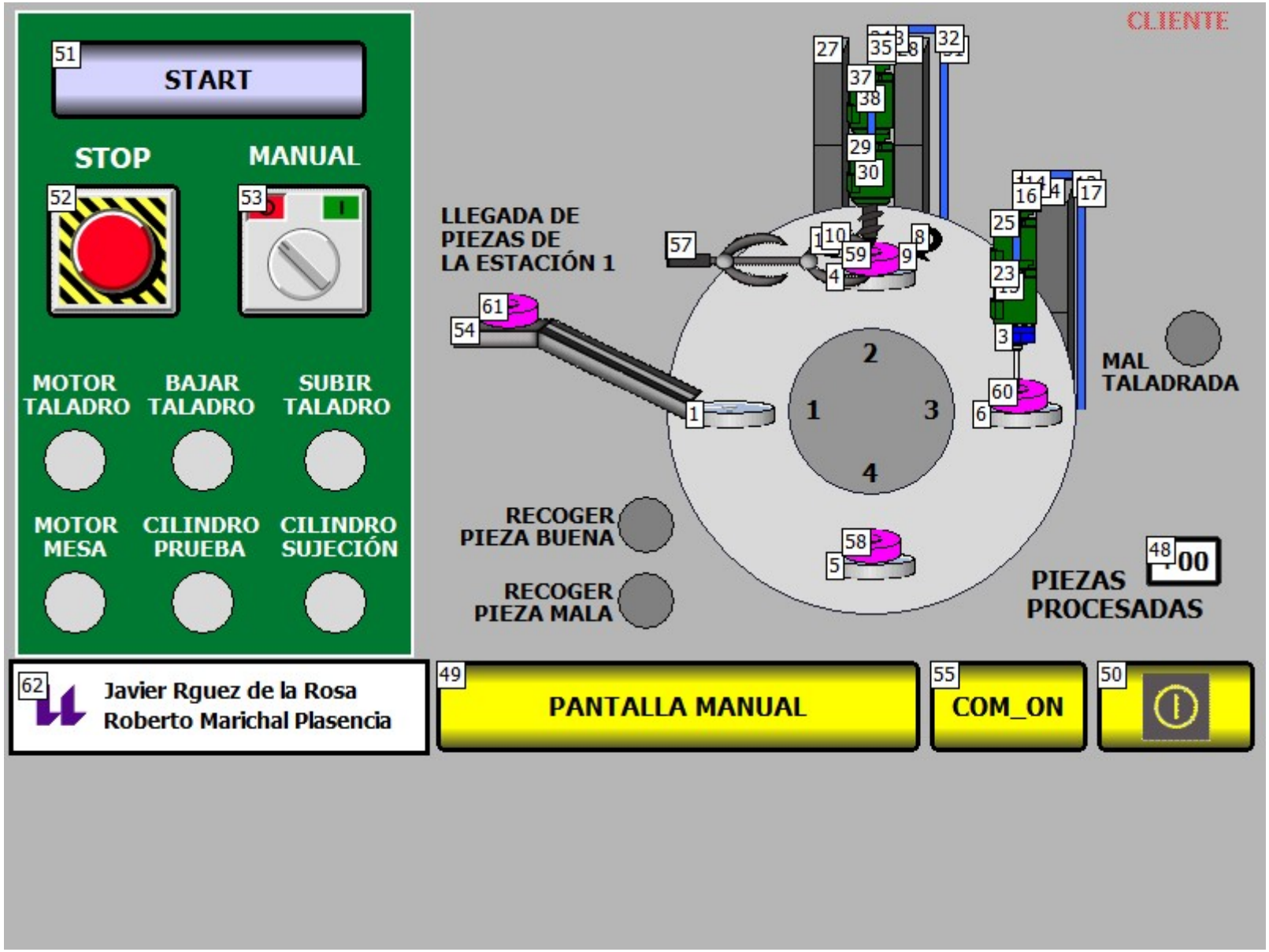
| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Temp_MedirPieza.ET | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | HMI_Conexión_7 | **Tipo de datos** | Time | **Longitud** | 4 |

# 13.4. ANEXO 4

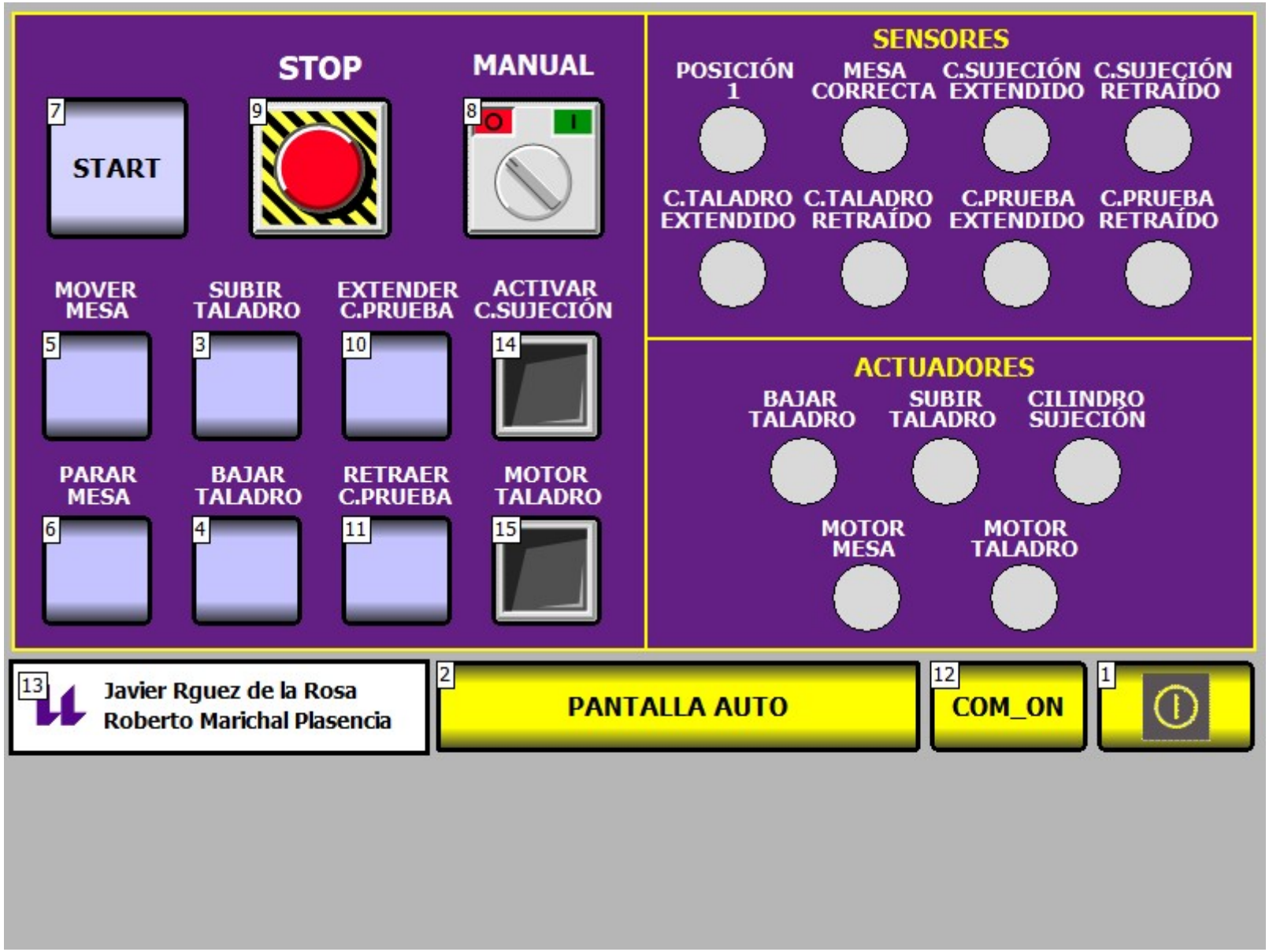
# Pantallas Cliente

# Estación 2

**Copia impresa de Pantalla_Auto**



**Copia impresa de Pantalla_Manual**



# 13.5. ANEXO 5

# Variables HMI

# HTTP Estación 2

# Variables HMI

## Tabla de variables estándar [47]

### ActivarTalad

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | ActivarTalad | **Nombre de visualización** | | **Dirección** | ActivarTalad |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### AUX_Panel

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | AUX_Panel | **Nombre de visualización** | | **Dirección** | AUX_Panel |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### BajarTaladro

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | BajarTaladro | **Nombre de visualización** | | **Dirección** | BajarTaladro |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### COM_Off

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | COM_Off | **Nombre de visualización** | | **Dirección** | COM_Off |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### Cont_Mesa_Especial.CV

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Cont_Mesa_Especial.CV | **Nombre de visualización** | | **Dirección** | Cont_Mesa_Especial.CV |
| **Conexión** | HTTP | **Tipo de datos** | Int | **Longitud** | 2 |

### Cont_Mesa_Normal.CV

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Cont_Mesa_Normal.CV | **Nombre de visualización** | | **Dirección** | Cont_Mesa_Normal.CV |
| **Conexión** | HTTP | **Tipo de datos** | Int | **Longitud** | 2 |

### Cont_MoverPieza.CV

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Cont_MoverPieza.CV | **Nombre de visualización** | | **Dirección** | Cont_MoverPieza.CV |
| **Conexión** | HTTP | **Tipo de datos** | Int | **Longitud** | 2 |

### Cont_PiezasProces.CV

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Cont_PiezasProces.CV | **Nombre de visualización** | | **Dirección** | Cont_PiezasProces.CV |
| **Conexión** | HTTP | **Tipo de datos** | Int | **Longitud** | 2 |

### Cont_Visualizacion.CV

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Cont_Visualizacion.CV | **Nombre de visualización** | | **Dirección** | Cont_Visualizacion.CV |
| **Conexión** | HTTP | **Tipo de datos** | Int | **Longitud** | 2 |

### CPExt

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | CPExt | **Nombre de visualización** | | **Dirección** | CPExt |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### CPRet

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | CPRet | **Nombre de visualización** | | **Dirección** | CPRet |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### CPrueba

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | CPrueba | **Nombre de visualización** | | **Dirección** | CPrueba |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### CSExt

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | CSExt | **Nombre de visualización** | | **Dirección** | CSExt |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### CSRet

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | CSRet | **Nombre de visualización** | | **Dirección** | CSRet |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### CSujecion

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | CSujecion | **Nombre de visualización** | | **Dirección** | CSujecion |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### CTExt

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | CTExt | **Nombre de visualización** | | **Dirección** | CTExt |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### CTRet

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | CTRet | **Nombre de visualización** | | **Dirección** | CTRet |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### Graf_pinza_sujec_1

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Graf_pinza_sujec_1 | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | <Variable interna> | **Tipo de datos** | Bool | **Longitud** | 1 |

### Graf_pinza_sujec_2

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Graf_pinza_sujec_2 | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | <Variable interna> | **Tipo de datos** | Bool | **Longitud** | 1 |

### Graf_rampa_pieza

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Graf_rampa_pieza | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | <Variable interna> | **Tipo de datos** | Bool | **Longitud** | 1 |

### Graf_taladro1

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Graf_taladro1 | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | <Variable interna> | **Tipo de datos** | Bool | **Longitud** | 1 |

### Graf_taladro2

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Graf_taladro2 | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | <Variable interna> | **Tipo de datos** | Bool | **Longitud** | 1 |

### Graf_ULL

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Graf_ULL | **Nombre de visualización** | | **Dirección** | |
| **Conexión** | <Variable interna> | **Tipo de datos** | Bool | **Longitud** | 1 |

### HMI_ActTaladro

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | HMI_ActTaladro | **Nombre de visualización** | | **Dirección** | HMI_ActTaladro |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### HMI_BajarTal

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | HMI_BajarTal | **Nombre de visualización** | | **Dirección** | HMI_BajarTal |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### HMI_ExtPrueba

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | HMI_ExtPrueba | **Nombre de visualización** | | **Dirección** | HMI_ExtPrueba |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### HMI_MoverMesa

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | HMI_MoverMesa | **Nombre de visualización** | | **Dirección** | HMI_MoverMesa |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### HMI_PararMesa

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | HMI_PararMesa | **Nombre de visualización** | | **Dirección** | HMI_PararMesa |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### HMI_RetPrueba

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | HMI_RetPrueba | **Nombre de visualización** | | **Dirección** | HMI_RetPrueba |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### HMI_START

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | HMI_START | **Nombre de visualización** | | **Dirección** | HMI_START |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### HMI_STOP

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | HMI_STOP | **Nombre de visualización** | | **Dirección** | HMI_STOP |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### HMI_SubirTal

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | HMI_SubirTal | **Nombre de visualización** | | **Dirección** | HMI_SubirTal |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### HMI_Sujecion

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | HMI_Sujecion | **Nombre de visualización** | | **Dirección** | HMI_Sujecion |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

### MesaCorrecta

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | MesaCorrecta | **Nombre de visualización** | | **Dirección** | MesaCorrecta |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

## MODO_MANUAL

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | MODO_MANUAL | **Nombre de visualización** | | **Dirección** | MODO_MANUAL |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

## MotorMesa

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | MotorMesa | **Nombre de visualización** | | **Dirección** | MotorMesa |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

## MotorTaladro

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | MotorTaladro | **Nombre de visualización** | | **Dirección** | MotorTaladro |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

## Orden_PiezaBuena_E3

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Orden_PiezaBuena_E3 | **Nombre de visualización** | | **Dirección** | Orden_PiezaBuena_E3 |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

## Orden_PiezaMala_E3

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Orden_PiezaMala_E3 | **Nombre de visualización** | | **Dirección** | Orden_PiezaMala_E3 |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

## PARADA

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | PARADA | **Nombre de visualización** | | **Dirección** | PARADA |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

## Posicion1

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Posicion1 | **Nombre de visualización** | | **Dirección** | Posicion1 |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

## Recibir_PiezaMala_E1

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Recibir_PiezaMala_E1 | **Nombre de visualización** | | **Dirección** | Recibir_PiezaMala_E1 |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

## ResetNormal

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | ResetNormal | **Nombre de visualización** | | **Dirección** | ResetNormal |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

## STOP

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | STOP | **Nombre de visualización** | | **Dirección** | STOP |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

## SubirTaladro

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | SubirTaladro | **Nombre de visualización** | | **Dirección** | SubirTaladro |
| **Conexión** | HTTP | **Tipo de datos** | Bool | **Longitud** | 1 |

## Temp_MedirPieza.ET

| | | | | | |
|---|---|---|---|---|---|
| **Nombre** | Temp_MedirPieza.ET | **Nombre de visualización** | | **Dirección** | Temp_MedirPieza.ET |
| **Conexión** | HTTP | **Tipo de datos** | DInt | **Longitud** | 4 |