



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

UNIDAD DE GESTIÓN DE  TECNOLOGÍAS

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y

AVIÓNICA

MONOGRAFÍA: PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
TECNÓLOGO EN ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y

AVIÓNICA

TEMA: “IMPLEMENTACIÓN DE UN SISTEMA DE COMUNICACIÓN

MODBUS TCP ENTRE EL PLC S7-300 Y S7-1200”

AUTOR: CACHAGO GÓMEZ, JONATHAN JAVIER

DIRECTOR: ING. SANDOVAL VIZUETE, PAOLA NATALY

LATACUNGA

2020



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**

**CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN & AVIÓNICA**

**CERTIFICACIÓN**

Certifico que la monografía, “**IMPLEMENTACIÓN DE UN SISTEMA DE COMUNICACIÓN MODBUS TCP ENTRE EL PLC S7-300 Y S7-1200**” fue realizado por el señor **Cachago Gómez, Jonathan Javier** el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Latacunga, 30 de Enero del 2020

Una firma manuscrita en tinta azul, que parece ser la de Paola Nataly Sandoval Vizuite.

.....  
**Ing. Paola Nataly Sandoval Vizuite**

C. C. 0503254005



## DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

### CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN & AVIÓNICA

#### AUTORÍA DE RESPONSABILIDAD

Yo, *Cachago Gómez, Jonathan Javier*, declaro que el contenido, ideas y criterios de la monografía: ***IMPLEMENTACIÓN DE UN SISTEMA DE COMUNICACIÓN MODBUS TCP ENTRE EL PLC S7-300 Y S7-1200***, es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Latacunga, 30 de Enero del 2020

Una firma manuscrita en tinta azul que parece ser 'Jonathan Cachago Gómez'. Debajo de la firma hay una línea de puntos.

**Jonathan Javier Cachago Gómez**

C. C. 1727363267



## DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

### CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN & AVIÓNICA

#### AUTORIZACIÓN

*Yo, Cachago Gómez, Jonathan Javier autorizo a la Universidad de las Fuerzas Armadas ESPE publicar la monografía: “IMPLEMENTACIÓN DE UN SISTEMA DE COMUNICACIÓN MODBUS TCP ENTRE EL PLC S7-300 Y S7-1200” en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.*

Latacunga, 30 de Enero del 2020

Una firma manuscrita en tinta azul que parece decir 'Jonathan Cachago', sobre una línea de puntos horizontal.

**Jonathan Javier Cachago Gómez**

C. C. 1727363267

## DEDICATORIA

*Este logro en mi vida va dedicado a mis padres Julio y María, quienes con su amor, confianza, esfuerzo y perseverancia durante todo este tiempo inculcándome valores y principios que ayudaron a lograr esta gran etapa en mi vida.*

*A mis hermanas Jhenny y Vivian por sus consejos y su apoyo incondicional.*

## AGRADECIMIENTO

*En primer lugar agradezco a mis padres por ser pilar y promotores de mis sueños, por la confianza depositada en mí en todo momento y por su amor incondicional.*

*A mis hermanas que con sus palabras de aliento me han impulsado cada día a ser una mejor persona.*

*A la Universidad de las fuerzas armadas – ESPE y a sus docentes quienes compartieron sus conocimientos en el transcurso de mi educación superior, de manera especial a la Ingeniera Paola Sandoval tutora de mi proyecto de investigación, quien con su amplio conocimiento supo guiarme para finalizarlo con éxito.*

## ÍNDICE DE CONTENIDO

### CARÁTULA

CERTIFICACIÓN ..... i

AUTORÍA DE RESPONSABILIDAD ..... ii

AUTORIZACIÓN ..... iii

DEDICATORIA ..... iv

AGRADECIMIENTO ..... v

ÍNDICE DE CONTENIDO ..... vi

ÍNDICE DE FIGURAS ..... xi

ÍNDICE DE TABLAS ..... x

RESUMEN ..... xv

ABSTRACT ..... xvi

## CAPÍTULO I

### PLANTEAMIENTO DEL PROBLEMA DE INVESTIGACIÓN

1.1. Antecedentes ..... 1

1.2. Planteamiento del problema ..... 2

1.3. Justificación ..... 3

1.4. Alcance ..... 3

1.5. Objetivos ..... 4

1.5.1. Objetivo general ..... 4

1.5.2. Objetivos específicos.....	4
-----------------------------------	---

## **CAPÍTULO II**

### **MARCO TEÓRICO**

2.1. Sistema de comunicación.....	5
2.1.1. Sistema de comunicación Industrial.....	5
2.2. Pirámide de automatización CIM ( <i>Computer Integrated Manufacturing</i> ) .....	6
2.2.2. Nivel de Proceso.....	7
2.3. Modbus dentro de la Pirámide de Automatización .....	9
2.4. Redes Industriales. ....	10
2.5. Modbus.....	11
2.5.1. Características Generales. ....	13
2.5.2. Ventajas y desventajas de Modbus.....	14
2.6. Estructura de mensaje.....	15
2.7. Códigos de funciones. ....	16
2.8. Protocolo MODBUS .....	17
2.9. Red Modbus TCP/IP ( <i>Transmission Control Protocol/Internet Protocol</i> ).....	17
2.9.1. Características y elementos. ....	18
2.9.2. Encapsulamiento de trama Modbus sobre el protocolo TCP .....	20
2.9.3. Modelo Cliente- Servidor.....	21
2.10. Configuración de la Red.....	22
2.10.1. Red .....	22
2.10.2. Configuración de las conexiones.....	23
2.10.3. Instrucciones Modbus TCP.....	23



2.11.	PLC ( <i>Programmable Logic Controller</i> ).....	24
2.12.	PLC S7 300 - CPU 315-2 PN/DP.....	25
2.12.1.	Características .....	26
2.13.	PLC S7 1200 – CPU 1215C AC/DC/RLY.....	27
2.13.1.	Características .....	27

### **CAPÍTULO III**

#### **DESARROLLO DEL TEMA**

3.1.	Configuración de la red PROFINET.....	30
3.2.	Programación del Servidor.....	35
3.2.1.	Crear bloque de datos tipo Array del Cliente 1.....	36
3.2.2.	Programación de segmentos del Servidor. ....	37
3.3.	Programación del Cliente 1.....	45
3.3.1.	Crear bloque de datos tipo Array del Cliente 1.....	48
3.3.2.	Programación de segmentos del Cliente 1. ....	49
3.4.	Programación del Cliente 2.....	52
3.4.1.	Crear variables para los datos a transmitir del Cliente_2.....	53
3.4.2.	Programación de segmentos del Cliente_2. ....	53
3.5.	Programación de pantalla HMI en Win CC. ....	57
3.5.1.	Crear plantilla.....	57
3.5.2.	Crear imágenes.....	57
3.6.	Pruebas de transmisión de datos en la red Modbus TCP/IP.....	63
3.6.1.	Transmisión de datos del Servidor a los Clientes a la vez. ....	63
3.6.2.	Transmisión de datos de los Clientes al Servidor. ....	64

3.6.3. Transmisión de datos de Clientes al Servidor con valores independientes.....	64
3.6.4. Transmisión de datos del Servidor a Clientes con valores independientes.....	64
3.6.5. Control de lectura o escritura. ....	65
3.6.6. HMI con todos los datos transmitidos en la Red Modbus TCP. ....	65
3.6.7. Comprobación de la estabilidad de la comunicación Modbus TCP.....	66

## **CAPÍTULO IV**

### **CONCLUSIONES Y RECOMENDACIONES**

4.1. Conclusiones .....	67
4.2. Recomendaciones.....	68

### **REFERENCIAS BIBLIOGRÁFICAS ..... 69**

### **ANEXOS ..... 74**

## ÍNDICE DE TABLAS

<b>Tabla 1</b> <i>Características principales de Modbus.</i> .....	14
<b>Tabla 2</b> <i>Principales códigos de función Modbus.</i> .....	16
<b>Tabla 3</b> <i>Características TCP</i> .....	18
<b>Tabla 4</b> <i>Estructura del mensaje en Modbus TCP/IP.</i> .....	21
<b>Tabla 5</b> <i>Datos técnicos del PLC S7 300- CPU 315-2 PN/DP.</i> .....	26
<b>Tabla 6</b> <i>Propiedades técnicas del PLC S7 1200 - CPU 1215C AC/DC/RLY.</i> .....	27
<b>Tabla 7</b> <i>Propiedades de comunicación del PLC S7 1200.</i> .....	28
<b>Tabla 8</b> <i>Dirección IP de dispositivos.</i> .....	35

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Pirámide de automatización.....	7
<b>Figura 2.</b> Pirámide de automatización por Niveles.....	9
<b>Figura 3.</b> Tipos de redes Industriales .....	11
<b>Figura 4.</b> Red Modbus.....	12
<b>Figura 5.</b> Estructura de mensaje Modbus.....	15
<b>Figura 6.</b> Encapsulamiento de la trama ModBus en TCP/IP .....	20
<b>Figura 7.</b> Modelo Cliente-Servidor .....	22
<b>Figura 8.</b> Arquitectura básica de un PLC .....	25
<b>Figura 9.</b> PLC S7 300 - CPU 315-2 PN/DP.....	26
<b>Figura 10.</b> PLC S7 1200 – CPU 1215C AC/DC/RLY.....	29
<b>Figura 11.</b> Crear Proyecto en TIA PORTAL .....	30
<b>Figura 12.</b> Agregar dispositivos PLC S7 300 Y S7 1200 .....	30
<b>Figura 13.</b> Configuración General y Dirección Ethernet del PLC S7 300.....	31
<b>Figura 14.</b> Activación de la marca de ciclo del PLC S7 300 .....	32
<b>Figura 15.</b> Activación de las marcas de ciclo del PLC S7 1200 .....	32
<b>Figura 16.</b> Agregar interfaz de usuario Win CC .....	33
<b>Figura 17.</b> Agregar módulo de comunicación integrado en Win CC.....	33
<b>Figura 18.</b> Configuración de la subred y la dirección IP de Win CC.....	34
<b>Figura 19.</b> Conexión PROFINET- Comunicación HMI .....	34
<b>Figura 20.</b> Bloque de comunicación MB_SERVER .....	35
<b>Figura 21.</b> Estructura TCON_IP_v4 del Servidor.....	36

<b>Figura 22.</b> Estructura TCON_IP_v4 del Servidor.....	37
<b>Figura 23.</b> Estructura TCON_IP_v4 del Servidor.....	38
<b>Figura 24.</b> Bloque de comunicación PUT en el servidor .....	38
<b>Figura 25.</b> Parámetros de comunicación S7 (PUT) en el Servidor. ....	39
<b>Figura 26.</b> Variables para la transmisión de datos (PUT) del servidor .....	40
<b>Figura 27.</b> Configuración de la instrucción PUT del Servidor.....	40
<b>Figura 28.</b> Bloque de comunicación GET en el servidor. ....	41
<b>Figura 29.</b> Parámetros de comunicación S7 (GET) en el Servidor .....	42
<b>Figura 30.</b> Variables para la transmisión de datos (GET).....	42
<b>Figura 31.</b> Configuración del bloque de comunicación GET del PLC 2. ....	43
<b>Figura 32.</b> Mover los datos Array a una variable.....	44
<b>Figura 33.</b> Variables del PLC 2.....	44
<b>Figura 34.</b> Bloque Modbus TCP, MB_CLIENT.....	45
<b>Figura 35.</b> Bloque de datos para configurar de conexión TCP. ....	46
<b>Figura 36.</b> Estructura TCON_IP_v4 del Cliente 1.....	47
<b>Figura 37.</b> Parámetros configurados del Cliente 1. ....	47
<b>Figura 38.</b> Bloque de datos, DATOS_TCP_C1 .....	48
<b>Figura 39.</b> Bloque de datos para la condición de leer o escribir. ....	49
<b>Figura 40.</b> Configuración de la instrucción MB_CLIENT .....	50
<b>Figura 41.</b> Condición para la señal de comunicación establecida.....	51
<b>Figura 42.</b> Condición que permite la selección de lectura o escritura. ....	51
<b>Figura 43.</b> Tabla de variables del PLC 1.....	52
<b>Figura 44.</b> Parámetros de comunicación S7 (GET) en el Cliente_2. ....	53

<b>Figura 45.</b> Variables para la transmisión de datos (GET) en el Cliente_2.....	53
<b>Figura 46.</b> Configuración del bloque de comunicación GET del PLC 3. ....	54
<b>Figura 47.</b> Variables para la transmisión de datos (PUT) en el Cliente_2.....	54
<b>Figura 48.</b> Configuración del bloque de comunicación PUT del PLC 3. ....	55
<b>Figura 49.</b> Mover los valores enviados a una variable del PLC3.....	56
<b>Figura 50.</b> Tabla de variables del PLC3.....	56
<b>Figura 51.</b> Plantilla_ModbusTCP.....	57
<b>Figura 52.</b> Configuración de plantilla. ....	58
<b>Figura 53.</b> Presentación de interfaz HMI.....	58
<b>Figura 54.</b> Imagen HMI para establecer comunicación. ....	59
<b>Figura 55.</b> Transmisión de datos entre el servidor y los dos clientes.....	60
<b>Figura 56.</b> Icono campo de texto.....	61
<b>Figura 57.</b> Icono Botón .....	61
<b>Figura 58.</b> Configuración del botón 1 .....	62
<b>Figura 59.</b> Configurar el botón para añadir una imagen. ....	62
<b>Figura 60.</b> Imagen deslizable a la derecha. ....	63
<b>Figura 61.</b> Transmisión de datos del Servidor a Clientes 1 y 2.....	63
<b>Figura 62.</b> Transmisión de datos del Servidor a Clientes 1 y 2.....	64
<b>Figura 63.</b> Transmisión de datos del Servidor a Clientes 1 y 2.....	64
<b>Figura 64.</b> Transmisión de datos del Servidor a Clientes 1 y 2.....	65
<b>Figura 65.</b> Selector de lectura y escritura en la interfaz HMI de Win CC .....	65
<b>Figura 66.</b> Transmisión de datos del servidor a clientes y viceversa.....	65
<b>Figura 67.</b> Estado de la red Modbus TCP .....	66

**Figura 68.** Imagen para establecer comunicación con el status de cada PLC.....66

## RESUMEN

La presente monografía describe las características de comunicación Modbus TCP/IP en los autómatas programables marca SIEMENS de la gama S7 300 y S7 1200, en esta red el primer autómata S7 1200 CPU 1215C AC/DC/RLY estará configurado como Servidor, el segundo autómata con las mismas características que el anterior está configurado como cliente 1 y por último un autómata de la serie S7 300 CPU 315-2 PN/DP está configurado como cliente 2 que en conjunto están en una red Ethernet bajo una conexión PROFINET. Este sistema de comunicación está basado en el modelo maestro – esclavo, razón por la cual las solicitudes realizadas por el cliente 1 y por el cliente 2 serán asistidas de forma independiente por el autómata configurado como servidor. Para la implementación de esta red Modbus TCP/IP se utilizó el software TIA PORTAL V15, este contiene las características necesarias para realizar diferentes tipos de redes industriales, entre estas esta Modbus, con la instrucción MB\_SERVER se configura como servidor y con la instrucción MB\_CLIENT se configura como cliente, esto permite realizar la transmisión de datos entre los autómatas programables y a su vez se integra a la red una interfaz HMI Win CC la cual mostrará los datos enviados o recibidos independientemente de cada autómata.

## PALABRAS CLAVE

- **MODBUS TCP/IP**
- **PROFINET**
- **PROTOCOLO CLIENTE SERVIDOR**
- **REDES INDUSTRIALES**



## **ABSTRACT**

This monograph describes the characteristics of Modbus TCP/IP communication in the SIEMENS range of programmable controllers S7 300 and S7 1200. In this network, the first controller S7 1200 CPU 1215C AC/DC/RLY is configured as a server, the second controller with the same characteristics as the previous one is configured as client 1 and finally a controller of the S7 300 CPU 315-2 PN/DP series is configured as client 2 which together are in an Ethernet network under a PROFINET connection. This communication system is based on the master-slave model, which means that requests from client 1 and client 2 are handled independently by the controller configured as a server. For the implementation of this Modbus TCP/IP network the software TIA PORTAL V15 was used, this contains the necessary characteristics to make different types of industrial networks, among these this Modbus, with the instruction MB\_SERVER it is configured as a server and with the instruction MB\_CLIENT it is configured as a client, this allows to make the data transmission between the programmable controllers and at the same time a Win CC HMI interface is integrated to the network which will show the data sent or received independently of each controller.

## **KEY WORDS**

- **MODBUS TCP/IP**
- **PROFINET**
- **CLIENT-SERVER PROTOCOL**
- **INDUSTRIAL NETWORKS**

## CAPÍTULO I

### PLANTEAMIENTO DEL PROBLEMA DE INVESTIGACIÓN

#### 1.1. Antecedentes

Previo a desarrollar la implementación de un sistema de comunicación Modbus TCP entre el PLC S7-300 y S7-1200, fue necesario realizar la investigación técnica que permita sustentar de forma clara y precisa la factibilidad para la ejecución del tema. Entre los trabajos analizados y que se consideraron relevantes fueron:

Los PLC son controladores robustos y resistentes a los ambientes hostiles que se encuentran en la industria (ISAAC RAMÍREZ ENRÍQUEZ, 2017) quien desarrolló el proyecto de grado con el tema: “CONTROL PID DE TEMPERATURA CON PLC SIEMENS S7-300 Y ALLEN BRADLEY SLC 500”, argumentando que “Un PLC es un dispositivo electrónico digital que usa una memoria programable para guardar instrucciones y llevar a cabo funciones lógicas, de secuencia, de sincronización, de conteo y aritméticas para controlar máquinas y procesos”.

El segundo trabajo revisado pertenece al autor ENCARNACIÓN SÁNCHEZ GARCÍA, 2017, quien desarrolló el proyecto de posgrado con el tema “AUTOMATIZACIÓN DE UNA FÁBRICA BASADO EN LA HERRAMIENTA TIA PORTAL”, donde describe la programación de PLCs con el software TIA PORTAL V13, y argumenta que: “La programación de estos autómatas se realiza mediante el uso de un software específico del fabricante denominado TIA PORTAL. Este software es de gran tamaño y permite programar autómatas con un lenguaje de programación muy fácil de programar, el lenguaje KOP.”

Es obligatorio conocer al modelo TCP/IP pues la gran mayoría de las redes a nivel mundial funcionan bajo este protocolo, (DÍAZ MOROCHO JORGE WLADIMIR y VIZCAÍNO

ASIMBAYA FRANCISCO, 2016) quienes desarrollaron el proyecto de grado con el tema “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE ENTRENAMIENTO EN REDES INDUSTRIALES UTILIZANDO TECNOLOGÍA SIEMENS”, en el cual argumentan que “Si un dispositivo de una red industrial debe conectarse por medio de una red tipo Ethernet o Internet a otro dispositivo en otra red, deberá necesariamente encapsular sus datos con los estándares que se maneja en TCP/IP, en particular deberá usar las direcciones IP.”

## **1.2. Planteamiento del problema**

El desarrollo de la tecnología en los equipos de redes de comunicación industrial ha incrementado dentro de las instituciones tanto públicas como privadas, en este caso en la Unidad de Gestión de Tecnologías (UGT), el laboratorio de Automatización y control de procesos está equipado con diversos equipos y estaciones entre ellos, módulo electro neumático, estación de temperatura, estación de caudal, etc. No obstante se pretende mejorar la tecnología de estudio con autómatas de vanguardia, implementando una red de comunicación ModBus TCP utilizando los PLCs S7 300 y S7 1200.

La inexperiencia y desconocimiento de esta tecnología que permite enlazar autómatas de dos equipos de la familia SIEMENS, provocará que los estudiantes de la Carrera de Tecnología Superior en Automatización e Instrumentación se encuentren con inconvenientes en el ámbito laboral al manejar este tipo de PLCs que surgen con el avance de la tecnología ya que poseen características mejoradas de sus antecesores.

Por tal motivo la implementación en el laboratorio de automatización y control de procesos de una red de comunicación Modbus, ayudará a que los estudiantes se familiaricen con protocolos de

comunicación que han surgido en PLCs de alta gama que a nivel profesional brindará oportunidades de liderazgo.

### **1.3. Justificación**

Hoy en día las Redes de Comunicación Industrial son de vital importancia en las diferentes industrias, es por ello que en el ámbito de la educación superior debe contener prácticas relacionadas con estas para que los estudiantes puedan obtener conocimientos óptimos e innovadores en el ámbito laboral y poder tomar decisiones mejorando la eficiencia en cualquier tipo de comunicación industrial.

La adquisición de dispositivos innovadores son esenciales en el campo de Redes de Comunicación Industrial, es por ello que se emplea un PLC de alta gama el cual permite implementar una comunicación Modbus TCP para la enseñanza dentro de la Unidad de Gestión de Tecnologías (UGT), permitiendo a los estudiantes interactuar de manera práctica y dinámica con autómatas que poseen tecnología que hoy en día son utilizadas en pequeñas, medianas y grandes industrias, de esta manera se complementaran los laboratorios con equipos que permitan elaborar redes de comunicación.

### **1.4. Alcance**

Este proyecto tiene como finalidad implementar una red de comunicación Modbus TCP entre un PLC S7 300 y un PLC S7 1200, que permitirá fortalecer los conocimientos prácticos de los estudiantes de la Carrera de Tecnología Superior en Automatización e Instrumentación en el área de redes de comunicación industrial que proporcionará una formación integral de acuerdo al perfil profesional para desenvolverse de una mejor manera con nuevas tecnologías que se presentan al pasar de los años.

El PLC S7-300 será adquirido y donado al laboratorio de automatización de la UNIDAD DE GESTIÓN DE TECNOLOGÍAS - ESPE para fines académicos en donde los estudiantes podrán relacionarse con las redes de comunicación industriales existentes dentro de las diferentes industrias.

## **1.5. Objetivos**

### **1.5.1. Objetivo general**

Implementar un sistema de comunicación Modbus TCP entre el PLC S7-300 y S7-1200 para mejorar el nivel de aprendizaje técnico de los estudiantes de la UGT.

### **1.5.2. Objetivos específicos**

- Realizar una investigación bibliográfica de las características técnicas de comunicación que ofrece el PLC S7-300 y el PLC S7-1200.
- Implementar la comunicación con protocolos Modbus TCP que brinde obtener un enlace de calidad.
- Analizar la eficiencia y eficacia de la red Ethernet con la conexión PROFINET entre el PLC S7-300 y S7-1200.

## CAPÍTULO II

### MARCO TEÓRICO

#### **2.1. Sistema de comunicación.**

Un sistema de comunicación básicamente es un conjunto de dispositivos utilizados con el fin de transmitir audio, video, voz y datos en forma de impulsos, señales electromagnéticas o eléctricas de un lugar a otro. Está compuesto por un emisor a cargo de generar y transmitir la información, un medio de transmisión por el cual transitaran los datos transmitidos y por ultimo un receptor que tiende a ser el que recibe la información. (Conacyt, 2017) & (Ferreira, 2016)

##### **2.1.1. Sistema de comunicación Industrial.**

Está basado principalmente en la trasmisión de datos e información en sistemas y circuitos electrónicos destinados con el fin de cumplir tareas de gestión, control y supervisión de procesos, permitiendo al controlador transmitir datos de información a los diferentes dispositivos o elementos que involucran el proceso industrial que pueden ser: sensores, detectores, actuadores, etc. (Murcia & Strack, 2016) & (Sarmiento, 2016)

Desde la perspectiva de (Sarmiento, 2016) hoy en día un sistema de comunicación industrial tiende a ser más exigente mientras la distancia con el proceso físico sea más reducida, la aplicación de las diferentes redes de comunicación son determinadas por tres características principales:

- Velocidad de transmisión
- Volumen de datos
- Velocidad de respuesta

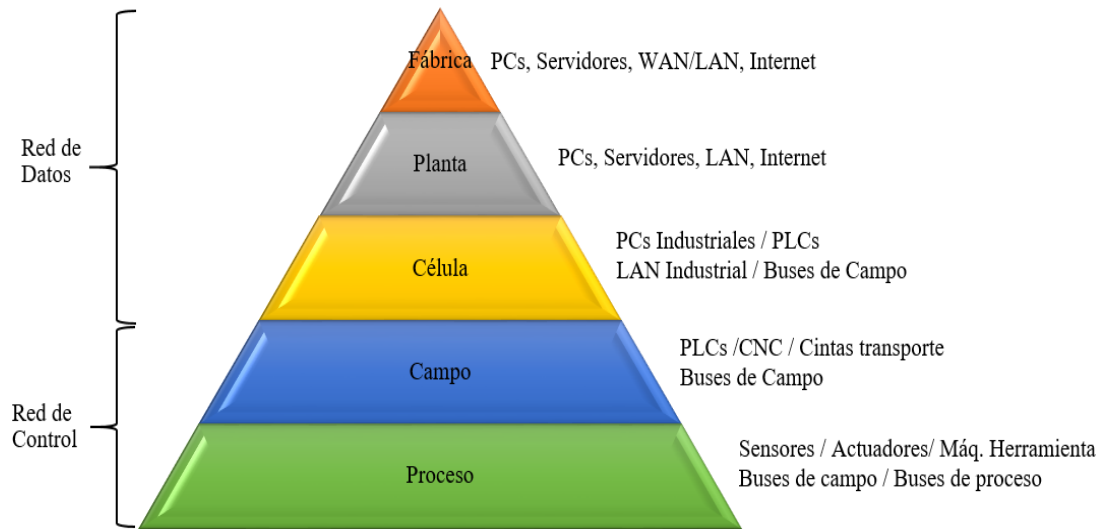
## **2.2. Pirámide de automatización CIM (*Computer Integrated Manufacturing*)**

Desde el punto de vista de (Hernández, 2016); la pirámide CIM está representada por 5 niveles jerárquicamente organizada, en donde en los niveles bajos se tiene actuadores y sensores, en los niveles intermedios de la pirámide estos elementos son interconectados para trabajar en conjunto para un fin común, en cuanto al nivel superior surge la red informática técnico-administrativa encargada de recolectar registros históricos, informaciones de estado, datos de partida; que permiten tomar las mejores decisiones o estrategias empresariales para mejorar el rendimiento industrial.

### **2.2.1. Características**

Manufactura integrada por computador tiene como propósito mejorar la eficiencia de todos los diferentes elementos que comprende la empresa principalmente que tengan relación con la producción para tener un mayor desempeño industrial, puntualizando los siguientes propósitos. (Aresi & Ardoli, 2015)

- Aumentar la flexibilidad.
- Mejorar la calidad del producto.
- Reducir los costos.
- Reducir el tiempo y el número de pasos empleados en la fabricación.
- Aumentar la confiabilidad del sistema.



**Figura 1.** Pirámide de automatización

Fuente: (López & Mora, 2017)

### 2.2.2. Nivel de Proceso

Identificado también como el nivel de instrumentación, conformado por componentes de mando “actuadores” y medida “sensores” que comparten un fin común. En consecuencia tienden a ser los elementos ligados que resaltan en el proceso de producción. Entre los sensores más comunes se tiene a los que permiten medir el nivel de temperatura, presión, líquidos, caudal, etc. Como ejemplos de actuadores se tiene: válvulas, motores, calentadores, taladros, etc. Todos estos elementos tanto actuadores como sensores necesitan ser controlados por un dispositivo. (Zambrano & Caballero, 2018)

### 2.2.3. Nivel de campo

Los dispositivos de mando y control están situados en este nivel, tales como: Equipos basados en microprocesadores, PLCs, controladores de robots, sistemas de control de movimiento, sistemas



embebidos y basados en PC; en este nivel también se encuentra la interfaz hombre-máquina con la capacidad de gestionar sensores y actuadores todo esto dentro del nivel de campo. En consecuencia este nivel está a cargo de emitir órdenes mediante la programación de sus diferentes dispositivos hacia el nivel inferior. (Sarmiento, 2016)

#### **2.2.4. Nivel de célula**

Este nivel posee elementos de control tales como PCs y PLCs industriales, los mismos que aportan a la recolección de datos del nivel de campo. Máquinas y operaciones ejecutan funciones coordinadas, este nivel es el responsable del sistema de control que secuencia y controla un trabajo específico. (Aresi & Ardoli, 2015)

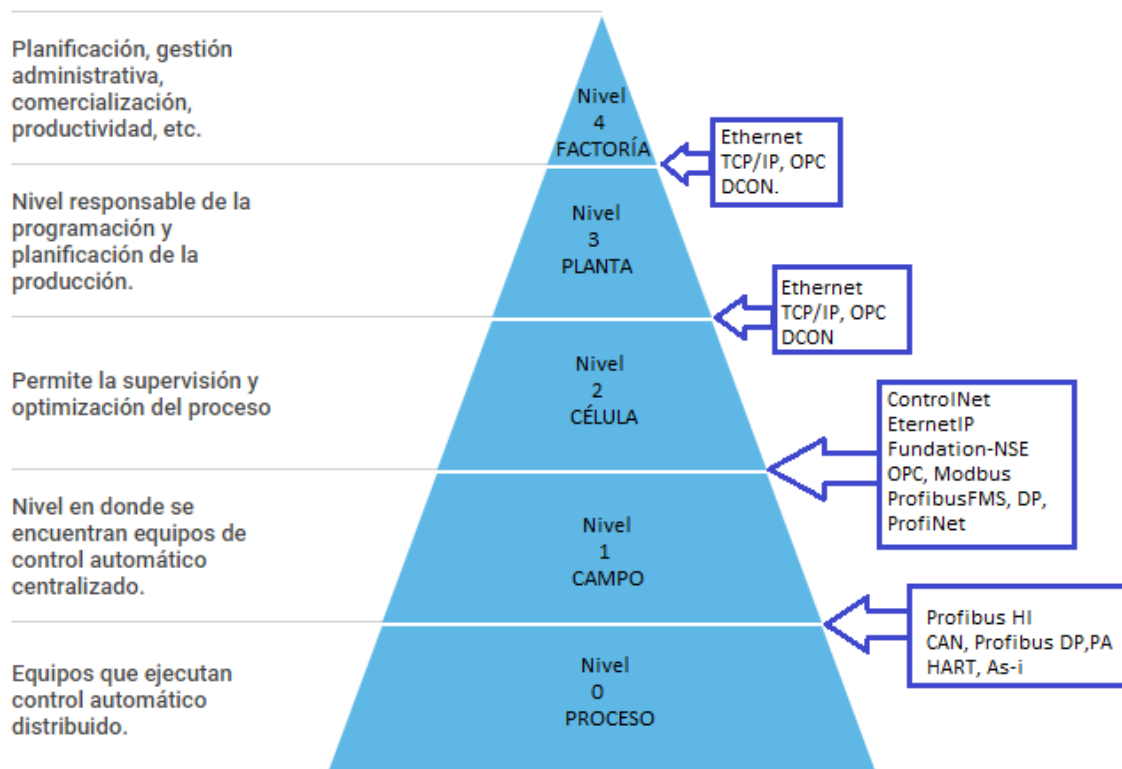
#### **2.2.5. Nivel de Planta**

El proceso de planta como se está llevando a cabo puede ser visualizado en este nivel, tener una imagen virtual de la planta a través de entornos SCADA (Supervisión, Control y Adquisición de Datos) de manera detallada de cada paso del proceso, además esto permite visualizar alguna falla de forma gráfica en un lugar específico con posibles soluciones inmediatas. (MECATRON, 2016)

#### **2.2.6. Nivel de Fábrica u Oficina**

En este nivel la supervisión de ventas y *stocks* son tareas procesadas de asunto corporativo. Maneja una red tipo LAN (*Local Area Network*) la cual consigue acceder a todos los puntos de la red con el objetivo de transmitir nuevas consignas de producción y recolectar datos de proceso. (Gutierrez, 2015)

### 2.3. Modbus dentro de la Pirámide de Automatización



**Figura 2.** Pirámide de automatización por Niveles.

Fuente: (Cassiolato, 2020)

Modbus el protocolo que utilizare para el desarrollo del presente proyecto se encuentra en el nivel 2 de la pirámide CIM por manipular dispositivos y controladores de planta que tienen control automático de los dispositivos de campo, tales como PLCs, en el Nivel 3 también es utilizado el Modbus TCP/IP ya que se tiene que comunicar con pantallas mediante este protocolo para la supervisión de procesos. Por otra parte se utiliza Modbus RTU para dispositivos específicamente de planta como por ejemplo variadores de frecuencia. Es importante destacar que las redes industriales dentro de Modbus permite la comunicación y conexión de un PLC con

componentes a través de las salidas y entradas analógicas sin embargo para el siguiente nivel se realiza a través de la conexión Ethernet del dispositivo. (López & Mora, 2017)

#### **2.4. Redes Industriales.**

Se puede definir a una red como una sucesión de nodos o dispositivos interconectados entre sí con el objetivo de obtener información disponible en cada dispositivo de la red o en el número de dispositivos que se desee, clasificada por redes deterministas y no deterministas mostradas en la figura 3 (Smartech, 2017)

Por ende, las redes industriales están constituidas por equipos de control, como:

- Controladores.
- Transductores y actuadores.
- Interfaces de Operador.
- PCs industriales.
- Sistemas de control distribuido.
- Módulos inteligentes.

Las redes industriales constituyen un modo de unir o enlazar diferentes dispositivos que permitan aumentar el rendimiento, entre estos están: ordenadores de diseño y gestión, autómatas, actuadores, sensores, etc. (Cedillo & Concepción, 2014)

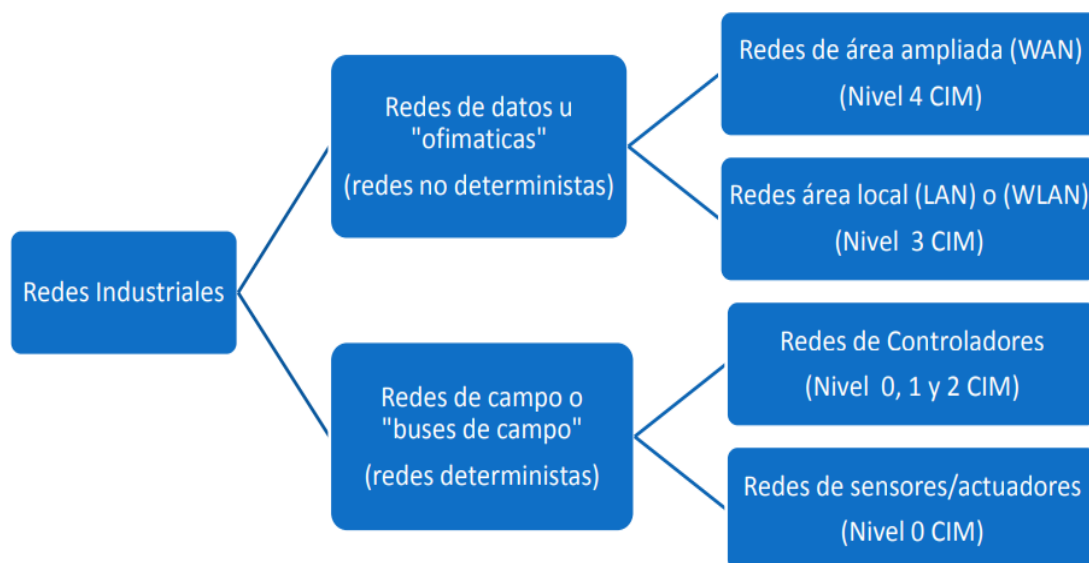
Entre las ventajas más destacadas en una red industrial son:

- Programación a distancia, sin necesidad de estar a pie de fábrica.
- Mejora del rendimiento general de todo el proceso.
- Visualización y supervisión de todo el proceso productivo.

- Posibilidad de intercambio de datos entre sectores del proceso y entre departamentos.
- Toma de datos del proceso más rápida o instantánea.

### 2.4.1. Tipos de redes industriales.

Las redes industriales están clasificadas por: redes de datos u ofimáticas y en redes de campo o buses de campo, cada una clasificada y con referencia en la pirámide CIM de la figura 1.



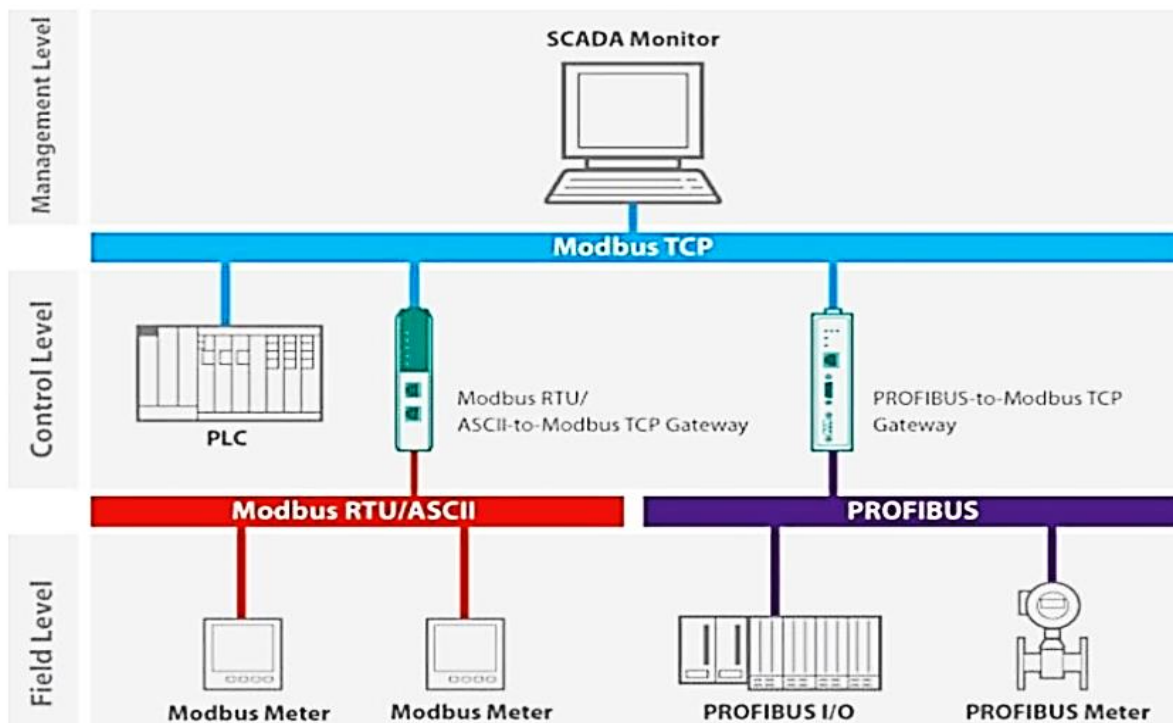
**Figura 3.** Tipos de redes Industriales

Fuente: (Murcia & Strack, 2016)

### 2.5. Modbus.

Desarrollado en el año 1979 por la empresa Modicon. Modbus es un protocolo de comunicación industrial, fue creada con el objetivo de unir o vincular los elementos de campo con los dispositivos controladores, trabaja con el modelo cliente-servidor o maestro esclavo utilizado para la transmisión de señales. Hoy en día se trabaja con una tecnología estándar libremente abierta siendo

perfecta para elementos de campo que necesitan monitorización remota RTU (*Remote Terminal Unit*, “Unidad terminal Remota”), diseñado para quipos industriales como: Computadores, PLCs y algún otro tipo de equipos físicos de entrada/salida que necesiten comunicarse en una red. Para la transmisión de datos Modbus utiliza distintas variantes considerando el ambiente en el que se desarrolle la aplicación, las variantes que la constituyen son: Modbus TCP/IP, RTU y ASCII; utilizando tanto la *extranet* que admiten operaciones de mando y monitorización, por otro lado la *intranet* es utilizado con el fin de integrar los elementos de planta. (Ver figura 4). (Sarmiento, 2016)



**Figura 4.** Red Modbus

Fuente: (Sarmiento, 2016)

### 2.5.1. Características Generales.

Modbus admite dos tipos de comunicación, la primera utilizando comunicación serial RS-232 y RS-485 y la segunda mediante Ethernet (Modbus TCP/IP), ambos transmiten información bajo el modelo Maestro-esclavo o Cliente-Servido. Las características que más destacan entre los tres modelos dentro de Modbus (TCP/IP, RTU y ASCII) se puede observar en la tabla 1. (Torres & Vega, 2015)

Hoy en día Modbus es un protocolo utilizado en diversos procesos industriales gracias a las ventajas que dispone, al momento de realizar control sobre dispositivos es considerado un protocolo de alta confiabilidad. (Defas & Guzmán, 2016)

Según (Defas & Guzmán, 2016) las características que más destacan de Modbus son:

- Al funcionar en modo maestro/esclavo, el maestro tiene el absoluto control de las comunicaciones con los esclavos.
- En toda la red, se puede disponer desde 1 hasta 247 esclavos. Cada esclavo posee una dirección única.
- La técnica de transmisión entre el maestro y los esclavos puede ser: *Broadcast* (uno a todos) o *Unicast* (uno a uno).
- Los códigos de funciones es usado por Modbus para el envío de operaciones a realizar por uno o más esclavos, y a la vez los clientes establecen la operación y envían de regreso la información o datos hacia el dispositivo maestro.
- Es un protocolo libre y público, brinda compatibilidad con una gran variedad de dispositivos electrónicos.

- Es fácil y sencillo de implementarlo, motivo por el cual es un protocolo que no requiere de un elevado costo de instalación.

**Tabla 1**

*Características principales de Modbus.*

<b>Características</b>	<b>Descripción</b>
Denominación	Modbus RTU/ASCII, Modbus TCP/IP.
Soporte	Modbus-IDA.
Topología	Bus, árbol, estrella.
Medio	Par trenzado, RS-232, RS-485 o Fibra.
Elementos	RTU/ASCII: 250 por segmento.
Distancia	RTU/ASCII: 350 metros. TCP/IP: 100 mts entre switches.
Comunicación	Maestro/Esclavo o Cliente/Servidor.
Velocidad	RTU/ASCII: 300bits/s-38.4Kbits/s. TCP/IP: 100Mbits/s.
Datos/Paquete	RTU/ASCII: 0-254 bytes. TCP/IP: 1500 bytes.

Fuente: (Torres & Vega, 2015)

### 2.5.2. Ventajas y desventajas de Modbus

Según (López & Mora, 2017) se consideran ventajas y desventajas.

#### a) Ventajas

- Es un software libre y disponible para la mayoría de fabricantes.
- Su implementación es relativamente fácil siempre y cuando se tenga la capacitación en redes de comunicación.
- Este protocolo no aplica excesivas restricciones en la manipulación de datos.
- Permite un enlace entre el nivel 2 y el nivel 3 en la pirámide de automatización.

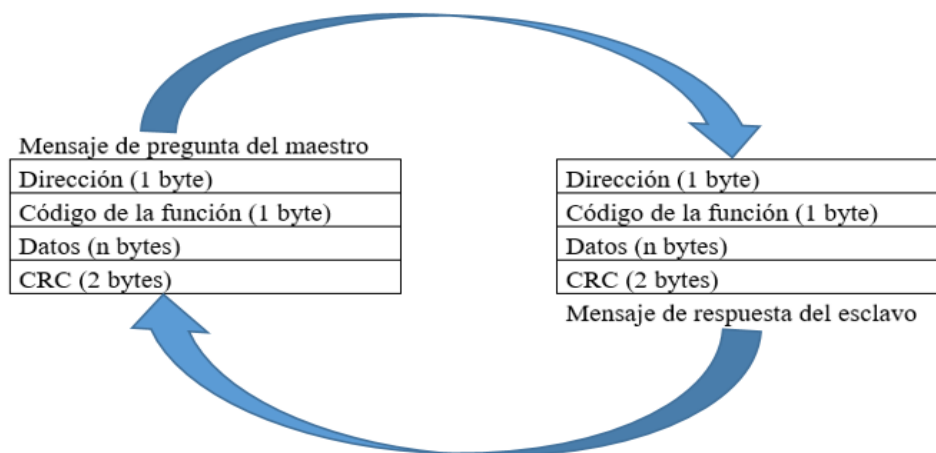
#### b) Desventajas

- Los avances de este protocolo solo pertenecen a la marca Schneider Electric, razón por la cual cuesta.
- Posee limitaciones al conectar algunos elementos por exponerse al ruido.

- En el estándar solo se permiten 254 nodos como máximo.
- Establecido en la jerarquía Maestro-Esclavo razón por la cual no permite comunicaciones no solicitadas.

## 2.6. Estructura de mensaje

En una red Modbus cuando los mensajes o datos son enviados desde el maestro bajo el concepto de petición o consulta, o en todo caso desde el esclavo bajo el concepto de respuesta; tienen una estructura del mensaje compuesta por cuatro partes: Dirección, Código de Función, Datos, CRC. Misma estructura que está compuesta de la misma manera para todos los mensajes independientemente de la red, siendo el campo de datos el único que varía dependiendo del caso. (Sarmiento, 2016)



**Figura 5.** Estructura de mensaje Modbus  
Fuente (Sarmiento, 2016)

### 2.6.1. Dirección.

El número máximo de esclavos soportados puede ir de 0 a 247 ya que estas son las direcciones validas en Modbus.



### 2.6.2. Código de función.

Los valores permitidos van entre 1-255 (decimal), el esclavo reconoce el tipo de acción que debe desarrollar cuando el maestro envía un mensaje que especifica el código de función.

### 2.6.3. Campo de Datos.

La longitud o tamaño de los datos enviados puede variar llegando incluso a ser cero, el esclavo cumple la acción requerida con los datos enviados desde el maestro y que este campo los contiene.

### 2.6.4. Comprobación

Este campo de la parte final del mensaje está destinado para la comprobación de errores.


## 2.7. Códigos de funciones.

Dentro de Modbus existen códigos de función que permiten al maestro intercambiar información con los esclavos, ya sea para leer o escribir datos, considerando que cada código (ver la tabla 2) pertenece a una tarea específica que el equipo subordinado debe responder. (Bartolomé, 2011)

**Tabla 2**

*Principales códigos de función Modbus.*

Código	Nombre	Descripción
1	Leer estado de bit.	Permite leer el valor de los bits de salida.
2	Leer estado de entrada.	Permite leer el valor de los bits de entrada.
3	Leer registros de retención.	Permite leer el valor de palabra de los registros de retención.
4	Leer registro de entrada.	Permite leer el valor de palabra de los registros de entrada.
5	Escribir bit de salida.	Permite escribir un bit de salida.
6	Escribir registro de retención.	Permite escribir un registro de retención.
7	Leer estado de excepción.	Permite leer el estado de 8 bits de excepción.
8	Funciones de diagnóstico.	Permite a través de subsunciones cumplir tareas de diagnóstico de los esclavos.
11	Contador de eventos.	Permite reconocer si el esclavo ha manejado correctamente los mensajes a través de un registro contador de eventos.
12	Registrador de eventos.	Retorna una palabra de estado, contador de eventos, contador de mensajes y un campo de eventos de byte.

**CONTINÚA** 

15	Escribir varios bits de salida.	Permite escribir varios bits de salida.
16	Escribir varios registros.	Permite escribir 1 o más registros de retención.
17	Reporte de ID de esclavo.	Retorna una descripción con el tipo de controlador (Micro84, 484, 184/384, 584, 884, 984. Cada tipo retorna información específica).
20	Leer referencia general.	Retorna el contenido de los archivos de referencia de la memoria extendida.
21	Escribir referencia general.	Permite escribir el contenido de los archivos de referencia de la memoria extendida.

Fuente: (Bartolomé, 2011)

## 2.8. Protocolo MODBUS

Modbus siendo un protocolo de comunicación industrial tiende a estar en la capa 7 del modelo OSI (*Open Systems Interconnection*, “Interconexión de sistemas abiertos”), vinculado con la arquitectura maestro/esclavo. Actualmente dentro de la industria es un protocolo estándar que posee gran disponibilidad de enlace de dispositivos industriales. La arquitectura de tipo maestro/esclavo posee un nodo maestro el cual cumple con la función de enviar instrucciones explícitas a cada nodo de los esclavos los mismos que procesan la respuesta solicitada. Una gran ventaja que posee este tipo de bus de campo es que mientras no exista una petición del nodo maestro, los nodos no transmiten información considerando que el resto de nodos esclavos no se enlazan dentro de la red. (Zambrano & Caballero, 2018).

## 2.9. Red Modbus TCP/IP (*Transmission Control Protocol/Internet Protocol*)

Modbus TCP/IP brinda servicios especificados por los códigos de función mediante el protocolo solicitud/respuesta. Este protocolo provee un conjunto de funciones para leer y escribir datos en los equipos de campo, tolera transferencias de datos bit y word.

Para garantizar un mayor rendimiento en Modbus TCP se debe considerar que se depende esencialmente del tipo y diseño de la red Ethernet que se vaya a utilizar, asimismo las interfaces de comunicación son esenciales para el óptimo rendimiento de los procesadores. (López & Mora, 2017)

Dentro de la familia del protocolo Modbus se ubica la versión o variante conocida como el protocolo Modbus TCP/IP que admite la implementación en las redes Ethernet. Tiende a ser muy utilizada para la transmisión de mensajes Modbus conforme el protocolo TCP/IP con el objetivo de supervisar y controlar el dispositivo de automatización. (Defas & Guzmán, 2016)

### 2.9.1. Características y elementos.

La comunicación de los dispositivos Modbus se lleva a cabo bajo la técnica maestro-esclavo, misma técnica en la que solo un dispositivo cliente o maestro tiene la potestad de iniciar transacciones o consultas. En relación a los dispositivos servidores o esclavos tienden a responder considerando el suministro de los datos solicitados al maestro/cliente o a través de la adopción de las medidas requeridas en la consulta. (Automation Networks, 2017)

En cuanto al tipo de red, topología, velocidad de datos entre otros, en la tabla 3 se puede ver los detalles.

**Tabla 3**

*Características TCP.*

<b>Característica</b>	<b>Descripción</b>
Tipo de red:	Ethernet-TCP/IP basado en la red Cliente/Servidor
Topología:	Muy flexible, con estructuras en estrella, árbol o bus. Todas las topologías que pueden ser implementados con la tecnología Ethernet estándar.
Instalación:	La tecnología de Ethernet 10, 100 o 1.000 Mbits/s estándar basada en alambres de cobre, fibra óptica o accesorios inalámbricos se

**CONTINÚA** 

	pueden utilizar.
Velocidad de datos:	10, 100 o 1.000 Mbits/s.
Max. Estaciones:	Casi ilimitada
Datos	Desde 1500 Byte por estructura Total: Casi ilimitada
Funciones de red:	Red simple Cliente/Servidor basada en tecnología Ethernet, estándar y los protocolos TCP/UDP/IP en la capa 3-4.

Fuente: (López & Mora, 2017)

TCP/IP y Ethernet son utilizados por Modbus para el transporte de los datos de la estructura del mensaje Modbus a través de equipos compatibles. Dicho de otra manera Modbus TCP/IP trabaja combinando una creación de redes estándar (TCP/IP), una red física denominada Ethernet y finalmente con un método estándar de presentación de datos considerando a Modbus como el protocolo de aplicación. En conclusión el dato Modbus TCP/IP viene a ser una comunicación Modbus que tiende a estar encapsulado en una red Ethernet TCP/IP. (Hernández, 2016)

## Ventajas

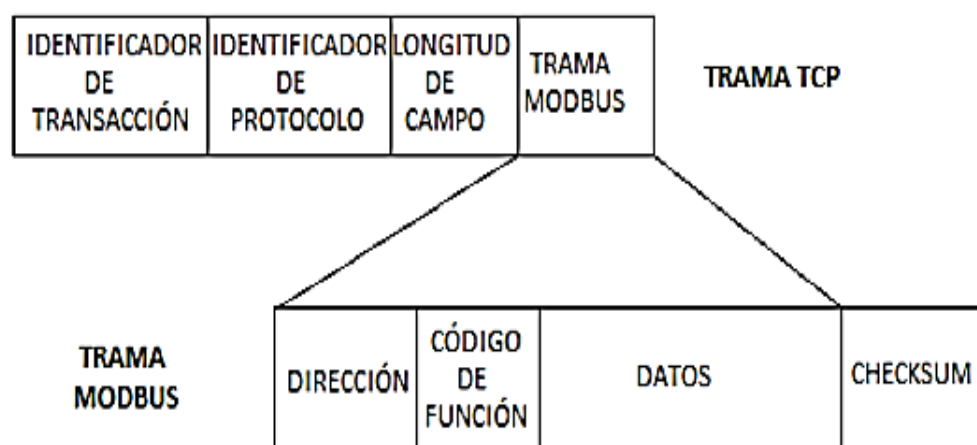
Modbus desarrolla esta versión con el objetivo de establecer comunicación utilizando el internet, por tal razón Modbus TCP/IP ofrece ventajas como:

- Se puede implementar sobre cualquier equipo o dispositivo que soporta sockets TCP/IP, razón por la cual aumenta la conectividad.
- Desde cualquier parte del mundo se puede direccionar un dispositivo.
- Es simple de expandir ya que no es necesario utilizar herramientas con compleja configuración.
- Existen convertidores Modbus a Modbus TCP/IP que permiten establecer comunicación con una red Modbus ya existente.

Cabe recalcar que Modbus TCP/IP es un protocolo libre y abierto que tiene un bajo costo a nivel de hardware, razón por la cual su implementación va creciendo cada día más a nivel industrial. (Hernández, 2016)

### 2.9.2. Encapsulamiento de trama Modbus sobre el protocolo TCP

Modbus/TCP en concreto abarca un marco Modbus en el interior de un marco TCP de una forma sencilla mostrada en la figura 6.



**Figura 6.** Encapsulamiento de la trama Modbus en TCP/IP  
Fuente: (Castro & Alvarado, 2017)

Es indispensable apuntar que la arquitectura del cuerpo de la solicitud y respuesta, iniciando desde el código de función hasta el final de los datos, considerando que a diferencia de las otras variantes de Modbus posee en exactitud el mismo significado y misma disposición, siendo el patrón de verificación de error, los delimitadores (inicial y final) del mensaje y siendo la forma de interpretar la dirección sus únicas diferencias. En el puerto registrado 502 todas las solicitudes son

enviadas vía TCP. Normalmente las solicitudes en una conexión dada son enviadas de forma half-duplex, o dicho de otra manera mientras una respuesta está pendiente no existe ventaja al enviar alguna solicitud adicional en una única conexión. (Castro & Alvarado, 2017)

La estructura de un mensaje Modbus TCP/IP completo se puede observar en la tabla 4.

**Tabla 4**

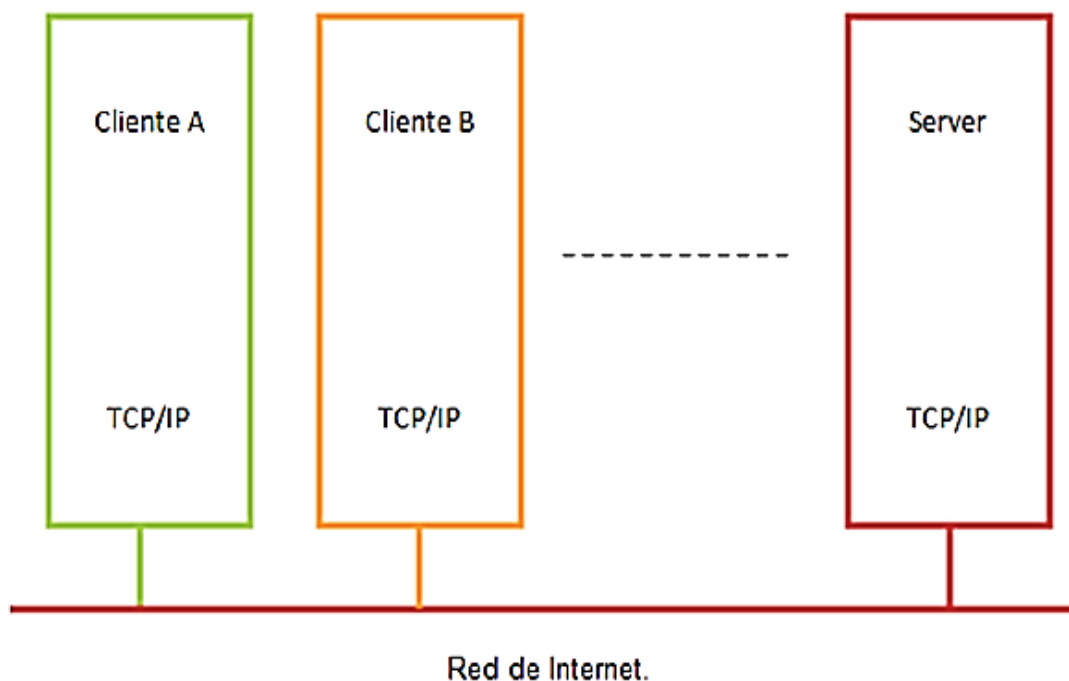
*Estructura del mensaje en Modbus TCP/IP.*

Posición del Byte	Significado
Byte 0	Identificador de transacción. Copiado por el servidor-normalmente 0.
Byte 1	Identificador de transacción. Copiado por el servidor-normalmente 0.
Byte 2	Identificador de protocolo=0.
Byte 3	Identificador de protocolo=0.
Byte 4	Campo de longitud (byte alto)= 0. Ya que los mensajes son menores a 256.
Byte 5	Campo de longitud (byte bajo). Número de bytes siguientes.
Byte 6	Identificador de unidad (previamente, “dirección esclavo”).
Byte 7	Código de función Modbus.
Byte 8 y más	Los datos necesarios.

Fuente: (Castro & Alvarado, 2017)

### 2.9.3. Modelo Cliente- Servidor

Siendo el modelo cliente-servidor utilizado generalmente en diversas aplicaciones para ejecutar las comunicaciones, en la figura 7 se puede apreciar un ejemplo simple del modelo cliente-servidor. Considerado como un servidor a una aplicación que brinda un servicio a los consumidores de internet, viene a ser una aplicación receptora de peticiones de servicios y los resultados retornan en forma de réplica. Por otra parte se tiene al cliente que viene a ser el solicitante del servicio, usando el protocolo TCP/IP como medio de transporte el cliente hace la petición de un servicio al servidor, hay que tomar en cuenta que el servidor logra realizar mencionada acción con varios clientes a la vez. (Gutierrez, 2015)



**Figura 7.** Modelo Cliente-Servidor  
Fuente: (Gutierrez, 2015)

## 2.10. Configuración de la Red.

Para la configuración de esta red se requiere la utilización de tres controladores lógicos programables PLC, un PLC S7 300 configurado como cliente, un PLC S7 1200 configurado como esclavo y otro PLC S7 1200 configurado como Servidor.

### 2.10.1. Red

Se establece una red Modbus TCP/IP compuesta por un CPU Simatic S7-300 denominado cliente 2, por un CPU Simatic S7-1200 denominado cliente 1 y un CPU Simatic S7-1200 que configurara como Servidor. Los tres PLCs establecen una red *Ethernet*, tanto a clientes como a servidores se asigna una única dirección IP. Una red Modbus RTU posee un solo maestro a

diferencia de una red Modbus TCP/IP que puede tener más de uno, la conexión multi-maestro es posible gracias a la tecnología TCP en donde fácilmente pueden ser conectados a la red más equipos en la configuración de cliente o servidor.

### **2.10.2. Configuración de las conexiones**

Para las conexiones en la red Modbus TCP/IP se debe manejar mediante las direcciones IP de los equipos PLCs y de los puertos de las CPUs, cuando se integra una instrucción de comunicación para un equipo configurado como esclavo se genera un DB (Bloque de datos) el mismo que posee cada uno de los parámetros que se necesitan para establecer una conexión.

Utilizando una sola dirección y puerto IP también se puede realizar de manera coordinada varias peticiones Modbus, en el caso de requerir los resultados de las peticiones estos pueden ser almacenados dependiendo de su naturaleza en diferentes áreas de memoria.

### **2.10.3. Instrucciones Modbus TCP.**

En la parte derecha de TIA PORTAL dentro de la categoría Modbus TCP se puede encontrar las instrucciones requeridas para establecer la comunicación a través del protocolo TCP, tales controles permiten configurar, recibir y enviar datos entre 2 dispositivos interconectados bajo el modelo Cliente-Servidor, a continuación se detalla la función de la estructura que maneja una comunicación Modbus vía Ethernet.

- Instrucción MB\_SERVER: Esta instrucción establece una comunicación bajo la posición de servidor Modbus mediante una red *Ethernet*. A través de MB\_SERVER se puede configurar el puerto necesario y el canal de comunicación, permite procesar peticiones bajo la función de



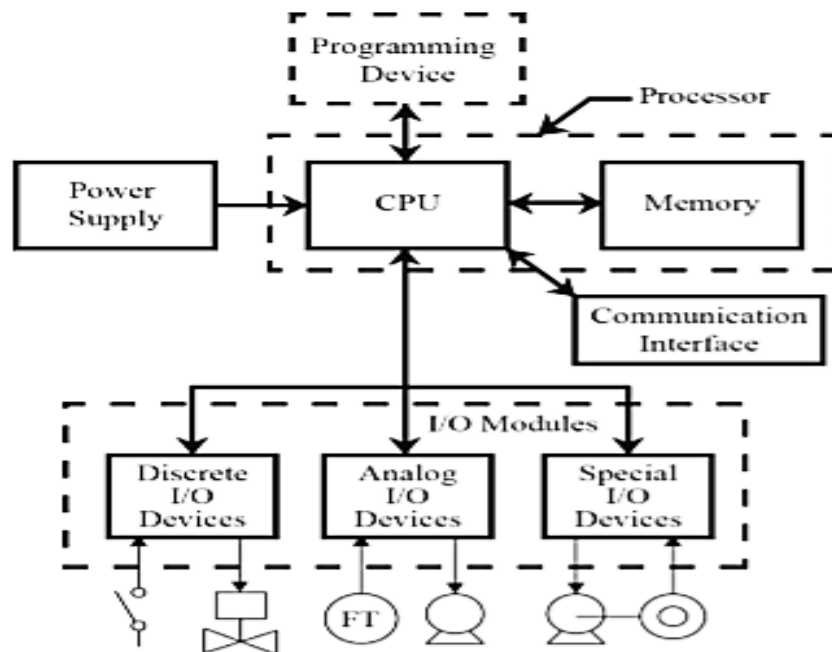
cliente Modbus y también permite enviar los oportunos mensajes de respuesta.

- Instrucción MB\_CLIENT: Es la que opera en si las comunicaciones de la red Modbus TCP/IP, esta permiten a una CPU operar como Cliente Modbus, entre las funciones que cumple son las de establecer una comunicación entre el cliente y servidor, asimismo mediante el código de funciones adecuado permite enviar peticiones, deshacer los canales creados y del equipo subordinado procesa las respuestas. (Sarmiento, 2016)

### **2.11. PLC (*Programmable Logic Controller*)**

Un autómata programable también conocido como PLC (Controlador Lógico Programable) en esencia es una computadora industrial misma que transfiere y procesa los datos de una maquina tales como: sensores, temporizadores, botones y entre otras señales de entrada con el objetivo de controlar los actuadores como: motores, válvulas, pistones y entre otros o básicamente es el encargado de controlar cualquier proceso industrial de forma automática. (Mecafenix, 2018)

Los principales elementos que componen la arquitectura básica de un PLC se puede apreciar en la figura 2.10, entre estos elementos esta la fuente de energía, el modulo procesador y los módulos de entrada y salida. El modulo procesador está compuesto por una CPU y una memoria. (Camposano & Rivera, 2016)



**Figura 8.** Arquitectura básica de un PLC

Fuente: (Camposano & Rivera, 2016)

La Unidad Central de Proceso posee una interface dirigida a un dispositivo programable y posiblemente puede tener interfaces a entradas y salidas remotas incluso a otras redes de comunicación. Usualmente el módulo de la fuente de alimentación y los módulos de entrada y salida tienden a venir por separado del procesador. Los ejemplares de los módulos de entrada y salida contienen tipos: analógicas como variables continuas, discretas (ON/OFF) y contadores rápidos. (Camposano & Rivera, 2016)

## 2.12. PLC S7 300 - CPU 315-2 PN/DP

Según (Siemens, Siemens, 2019) Argumenta que “La CPU 315-2 PN/DP es una CPU con memoria de programa de capacidad media. Se utiliza en instalaciones que, además de unidades

periféricas centrales, también cuentan con una topología de automatización descentralizada. En SIMATIC S7-300 puede aplicarse como PROFINET IO Controller y como maestro estándar PROFIBUS DP. La CPU 315-2 PN/DP también se utiliza como inteligencia descentralizada (esclavo PROFIBUS DP o PROFINET I-Device).”



**Figura 9.** PLC S7 300 - CPU 315-2 PN/DP  
Fuente: (Siemens, Siemens, 2019)

### 2.12.1. Características

**Tabla 5**

*Datos técnicos del PLC S7 300- CPU 315-2 PN/DP.*

Datos técnicos	
Tensiones, Intensidades	
Tensión de alimentación (valor nominal)	24 V DC
Consumo (en marcha en vacío), típ.	150 mA
Consumo de corriente (valor nominal)	750 mA
<b>Memoria</b>	
<b>Memoria de trabajo</b>	
Integrados	384 kB

**CONTINÚA** →

Ampliable	No
Tamaño de la memoria remanente para bloques de datos remanentes, máx.	128 kB
<b>Memoria de carga</b>	
Insertable (MMC)	Sí
Insertable (MMC), máx.	8 MB
Conservación de datos en la Micro Memory Card (tras la última programación), mín.	10 años
<b>Canales digitales</b>	
Entradas	16384
Salidas	16384
Entradas, de ellas centralizadas	1024
Salidas, de ellas centralizadas	1024
<b>Canales analógicos</b>	
Entradas	1024
Salidas	1024
Entradas, de ellas centralizadas	256
Salidas, de ellas centralizadas	256
<b>Funciones de comunicación</b>	
Comunicación PG/OP	Sí
TCP/IP	Sí (a través de la interfaz PROFINET integrada y FBs cargables)
Número de enlaces, máx.	8
Longitud de datos con el tipo de enlace 01H, máx.	1460 bytes
Longitud de datos con el tipo de enlace 11H, máx.	32768 bytes
Varios enlaces pasivos posibles por puerto soportado (Multipuerto)	Sí

Fuente: (Siemens, Simatic, 2011)

## 2.13 PLC S7 1200 – CPU 1215C AC/DC/RLY

### 2.13.1 Características

**Tabla 6**

*Propiedades técnicas del PLC S7 1200 - CPU 1215C AC/DC/RLY.*

<b>Datos técnicos</b>	<b>Descripción</b>
Rango de tensión.	De 85 a 264 V AC.
Frecuencia de línea.	De 47 a 63 Hz.
Intensidad de entrada (Solo CPU).	100 mA a 120 V AC - 50 mA a 240 V AC.

**CONTINÚA** →

Intensidad de entrada (CPU con todos los accesorios de ampliación).	300 mA a 120 V AC - 150 mA a 240 V AC.
Memoria de trabajo.	100KB.
Memoria de carga.	4 MB, interna, ampliable hasta tamaño de tarjeta SD.
Memoria remanente.	10 KB.
E/S digitales integradas.	14 entradas/10 salidas.
E/S analógicas integradas.	2 entradas/2 salidas.
Tamaño de la memoria imagen de proceso.	1024 bytes de entradas (I)/1024 bytes de salidas (Q).
Área de marcas (M).	8192 bytes.
Ampliación con módulos de señales.	8 SM máx.
Ampliación con módulos de comunicación.	3 CM máx.
Memory Card.	SIMATIC Memory Card (opcional).
Precisión del reloj en tiempo real.	MAS/- 60 segundos/mes.

Fuente: (Siemens, Simatic, 2014)

**Tabla 7**

*Propiedades de comunicación del PLC S7 1200.*

Datos Técnicos	Descripción
Número de puertos	2
Tipo	Ethernet
Dispositivo HMI	3
Programadora (PG)	1
Conexiones	<ul style="list-style-type: none"> <li>• 8 para Open User Communication (activa o pasiva): TSEND_C, TRCV_C, TCON, TDISCON, TSEND y TRCV.</li> <li>• 3 para comunicaciones S7 GET/PUT (CPU a CPU) de servidor</li> <li>• 8 para comunicaciones S7 GET/PUT (CPU a CPU) de cliente</li> </ul>
Transferencia de datos	10/100 Mb/s
Aislamiento	Aislado por transformador, 1500 V AC, solo para seguridad frente a defectos breves
Tipo de cable	CAT5e apantallado

Fuente: (Siemens, Simatic, 2014)



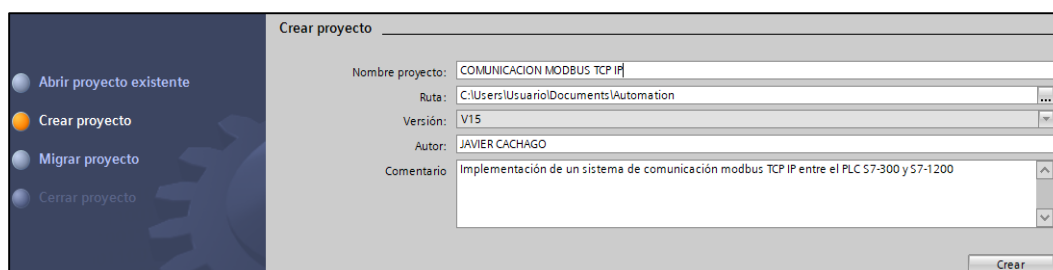
**Figura 10.** PLC S7 1200 – CPU 1215C AC/DC/RLY.  
Fuente: (Siemens, Siemens, 2019)

## CAPÍTULO III

### DESARROLLO DEL TEMA

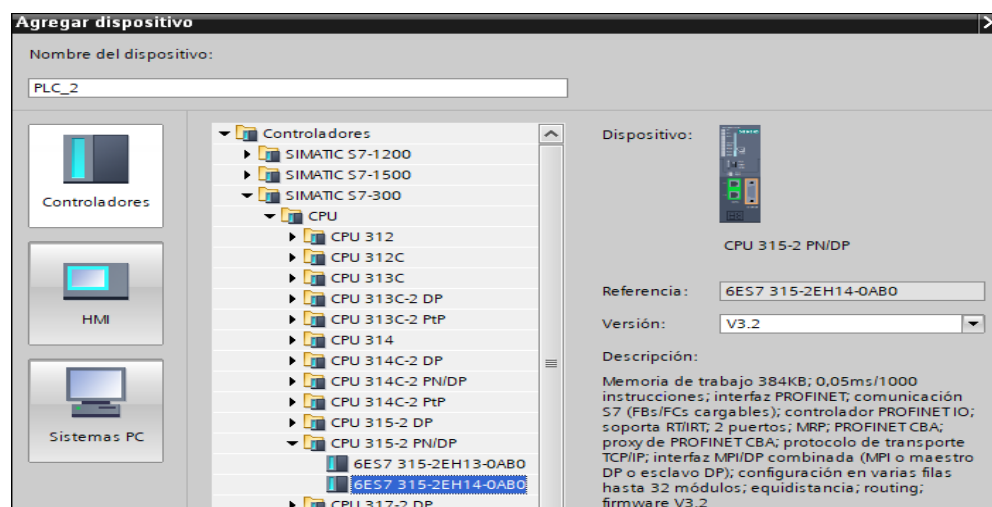
#### 3.1. Configuración de la red PROFINET.

1. Abrir el software TIA PORTAL, clic en Crear proyecto, escribir el nombre del proyecto y clic en Crear como se muestra en figura 11.



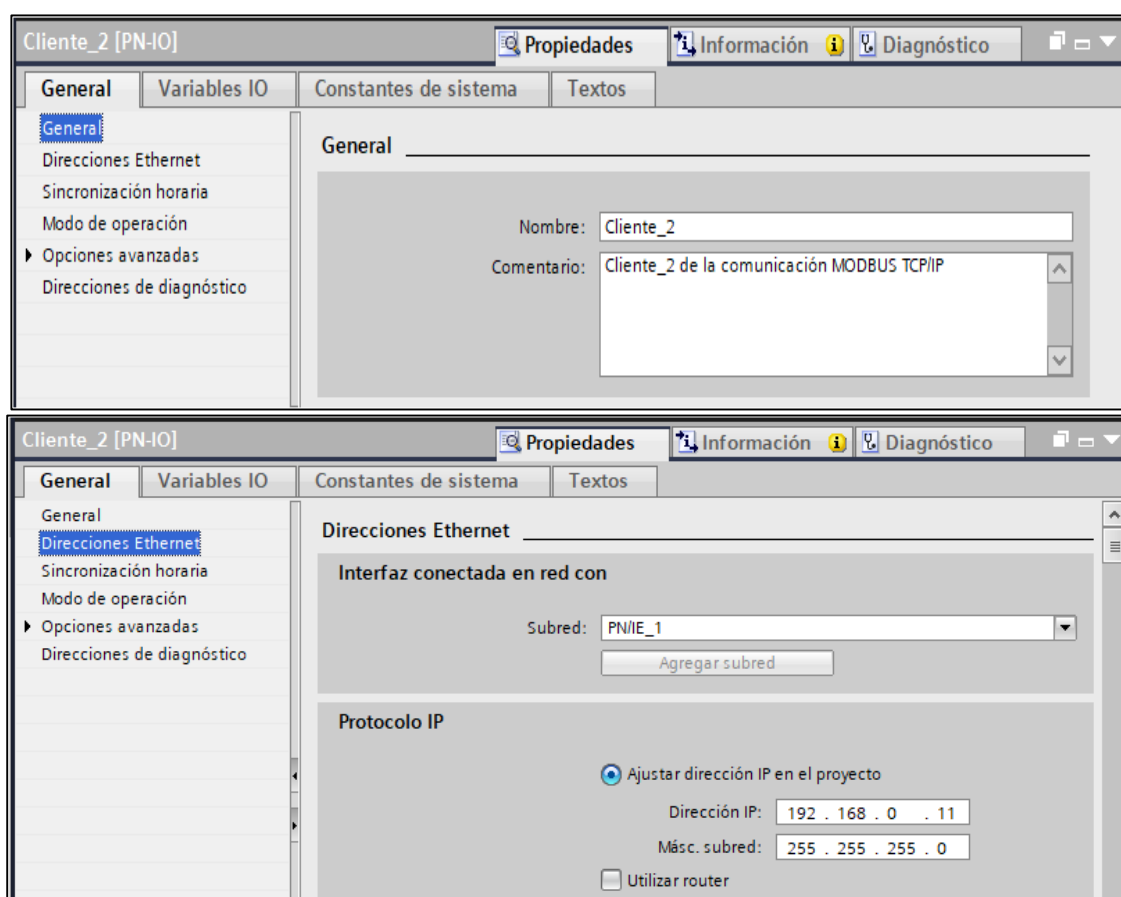
*Figura 11.* Crear Proyecto en TIA PORTAL

2. En la siguiente ventana que se muestra en la figura 12 dar clic en **Configurar un dispositivo**> **Agregar dispositivo**> **Controladores**, elegir los PLCs a utilizar que en este caso es el PLC S7 300 CPU 315-2 PN/DP referencia 6ES7 315-2EH14-0AB0 y 2PLCs s7 1200 CPU 1215C AC/DC/Rly referencia 6ES7 215-1BG40-0XB0 y clic en **Agregar**.



*Figura 12.* Agregar dispositivos PLC S7 300 Y S7 1200

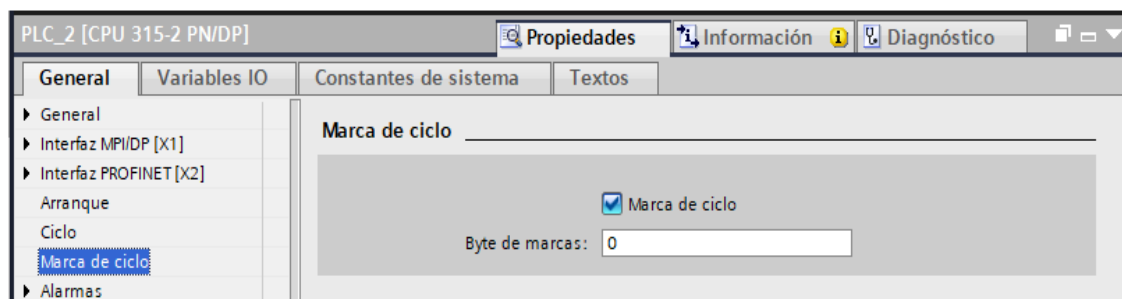
3. En el apartado de dispositivos, clic en **Dispositivos y redes**, Clic en el PLC para proceder a configurar, clic en **Interfaz PROFINET \_1** como se muestra en la figura 13. Automáticamente se abrirá una ventana, clic en **Propiedades> General> General**, ventana en la cual se cambiara el nombre del PLC a *PLC3\_Cliente2*. Posteriormente se configura la subred y la dirección IP, para ello clic en **Propiedades> General> Direcciones Ethernet**, en esta ventana clic en agregar subred que automáticamente es asignada con el nombre de **PN/IE\_1**, fijar la dirección IP: 192.168.0.11.



**Figura 13.** Configuración General y Dirección Ethernet del PLC S7 300

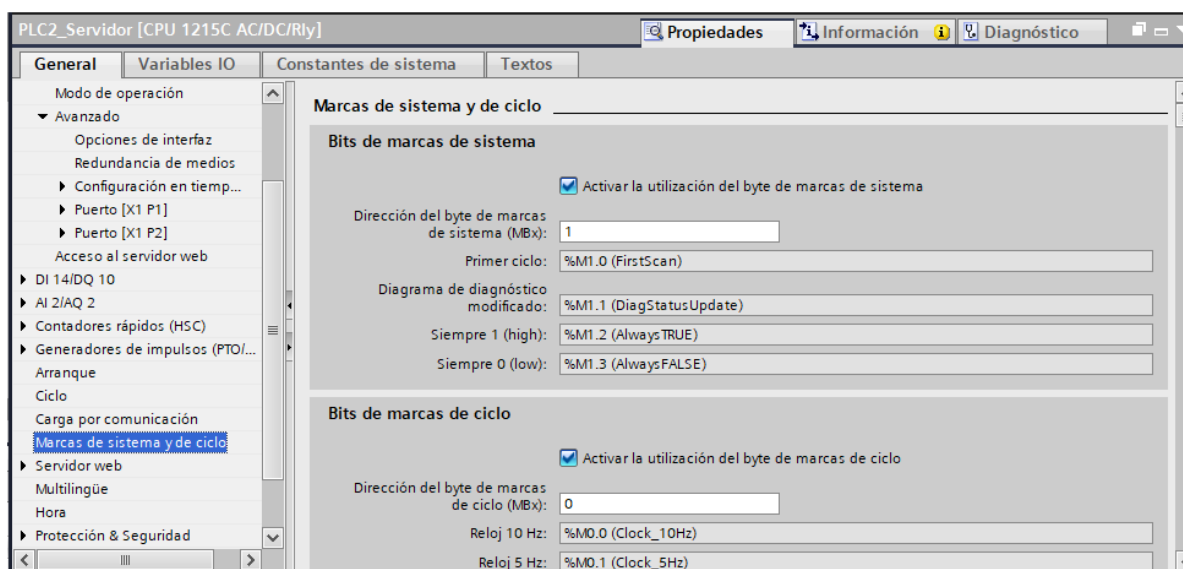
Para el PLC S7 300 se activa las marcas de ciclo del PLC detallada en la figura 14, clic en **Propiedades> General> Marca de ciclo**.





**Figura 14.** Activación de la marca de ciclo del PLC S7 300

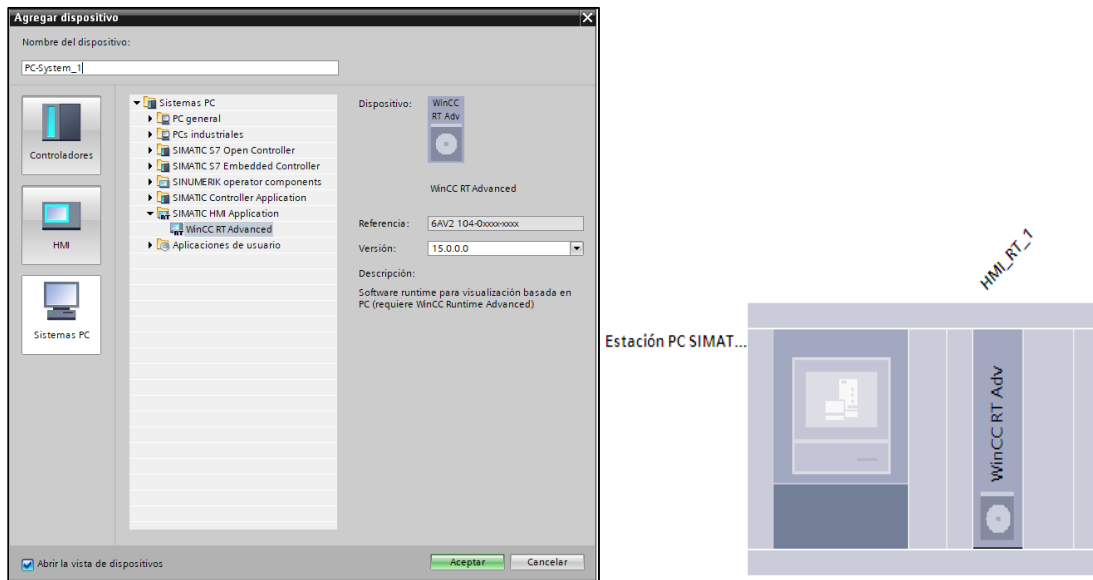
4. Para los PLCs S7 1200 se realiza los mismos pasos de configuración que en el PLC S7 300 a excepción de las marcas de ciclo, dentro de la ventana de dispositivos y redes como se muestra en la figura 15, seleccionar el PLC S7 1200, clic en **Propiedades > General > Marcas de sistema y de ciclo** se debe activar las pestañas: *Activar la utilización del byte de marcas del sistema* y *Activar la utilización del byte de marcas de ciclo*.



**Figura 15.** Activación de las marcas de ciclo del PLC S7 1200

5. Una vez terminada la configuración de los 3PLCs, agregar una interfaz de usuario Win CC en TIA PORTAL que permite observar todos los datos que se transmiten mutuamente entre los PLCs. En el apartado de *Dispositivos* que se muestra en la figura 16, clic en **Agregar dispositivo >**

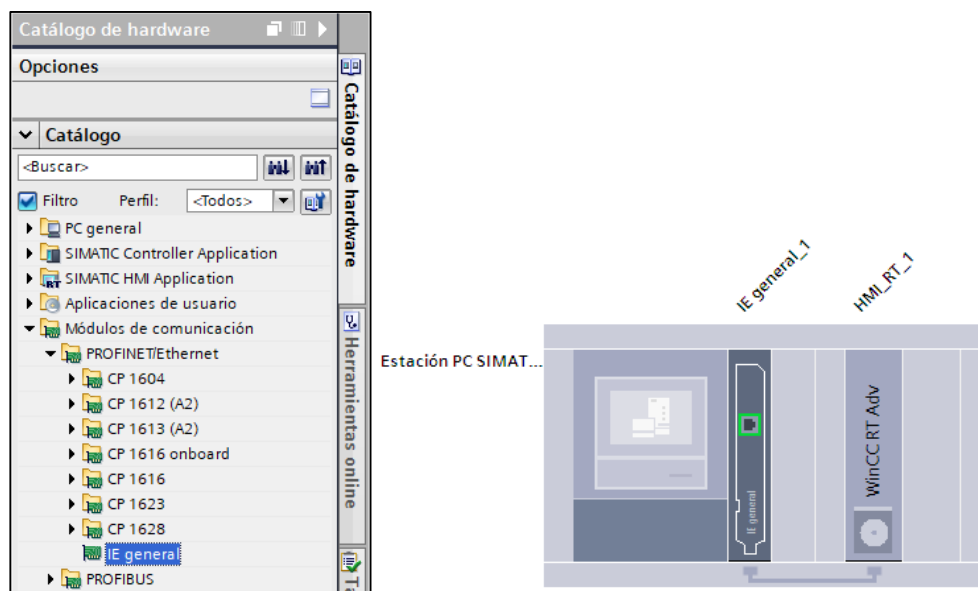
**Sistemas PC**, en la carpeta de *SIMATIC HMI Application* seleccionar *Win CC RT Advanced* y clic en **Aceptar**.



**Figura 16.** Agregar interfaz de usuario Win CC

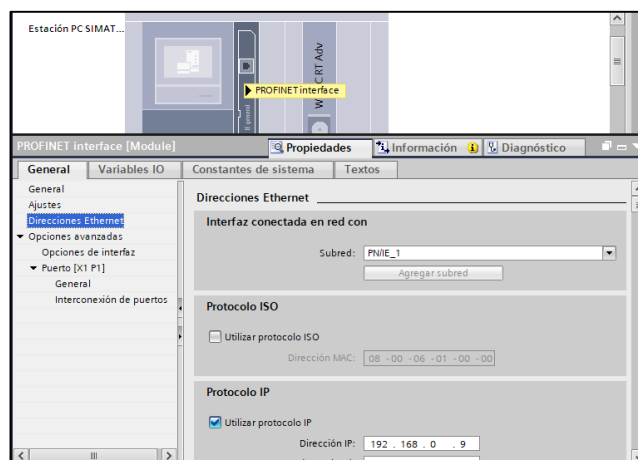
6. En la figura 17 se muestra en el apartado derecho un *Catálogo de hardware* clic en **Catálogo**

**Módulos de comunicación** > **PROFINET/Ethernet** y seleccionar *IE general*.



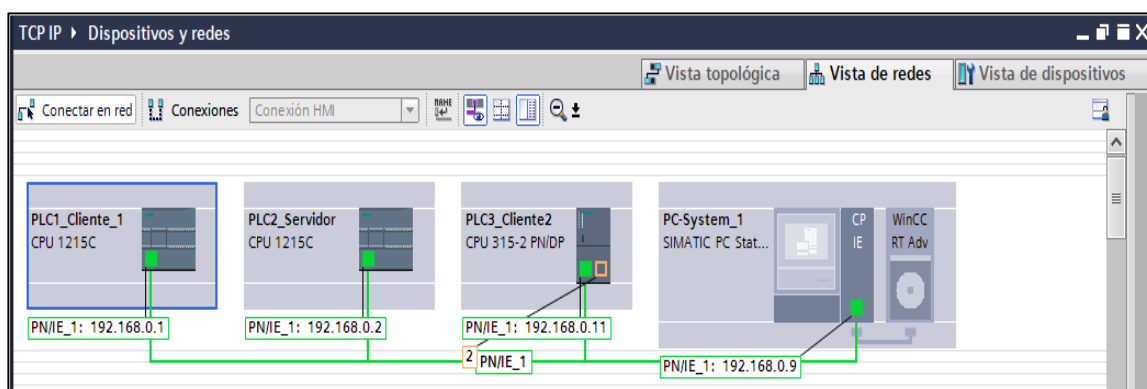
**Figura 17.** Agregar módulo de comunicación integrado en Win CC

7. En el módulo de comunicación integrado en Win CC dar clic en **PROFINET interface**, se abrirá una ventana de configuración, clic en **Propiedades> Direcciones Ethernet**, se despliega una ventana de propiedades que se muestra en la figura 18, en ella agregar Win CC a la misma subred de los PLCs denominada *PN/IE\_1*, es muy importante que el módulo este configurado con la misma dirección IP que tenga la PC que en este caso es IP 192.168.0.9.



**Figura 18.** Configuración de la subred y la dirección IP de Win CC.

8. Una vez terminado de agregar todos los dispositivos se puede apreciar toda la conexión PROFINET en una red Ethernet, en el apartado de *Dispositivos y redes* detallado en la figura 19 con sus respectivas IPs descrita en la tabla 10.



**Figura 19.** Conexión PROFINET- Comunicación HMI

**Tabla 8**

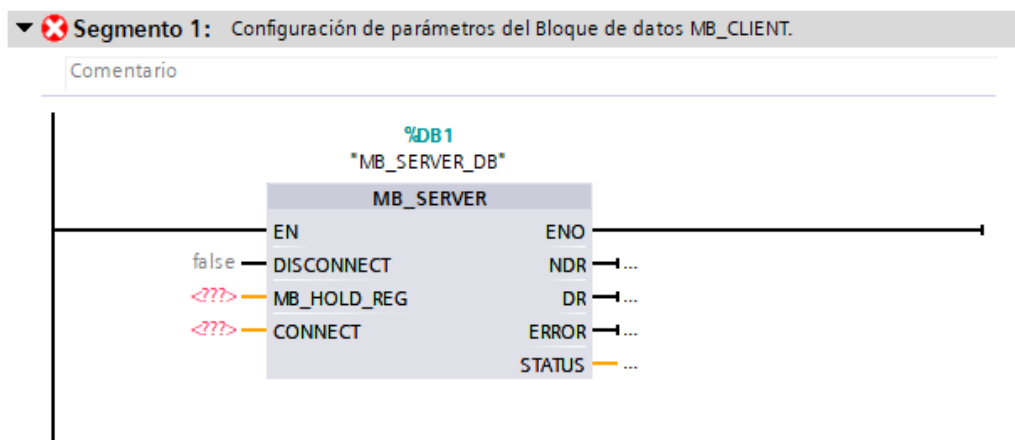
*Dirección IP de dispositivos.*

Dispositivo	IP
PLC1_Cliente_1	192.168.0.1
PLC2_Servidor	192.168.0.2
PLC3_Cliente_2	192.168.0.11
Win CC	192.168.0.9

### 3.2. Programación del Servidor

1. El *PLC2\_Servidor* se configurara como servidor de la red Modbus TCP. **MB\_SERVER** es un bloque integrado en TIA PORTAL en el apartado de comunicación, el mismo que permite realizar la comunicación de una forma más sencilla entre dos PLCs S7 1200 sin la necesidad de ninguna licencia y de ningún modulo adicional.

En el apartado de *Dispositivos* clic en **PLC2\_Servidor**> **Bloques de programas**> **Servidor** **[OB1]**, aparecerá una ventana en la cual se escriben las condiciones a cumplir en el servidor. El segmento 1 estará compuesto por el bloque de comunicación que trabajará como Servidor, en el apartado de *Instrucciones* clic en **Comunicación**> **Otros**> **Modbus TCP** y seleccionar el bloque **MB\_SERVER**.



**Figura 20.** Bloque de comunicación MB\_SERVER

2. Al igual que en el bloque MB\_CLIENT en MB\_SERVER también se debe crear una estructura TCON\_IP\_v4 el cual integra los parámetros a configurar para la comunicación Modbus TCP, para ello se crea un bloque de datos global que en este caso se denomina *MB\_Config\_M*, se crea una variable llamada *Connect* y en el tipo de dato se escribe la estructura TCON\_IP\_v4, estos valores de detallan en la figura 21.

■	▼ Connect	TCON_IP_v4	0.0		
■	InterfaceId	HW_ANY	0.0	64	Identificador de hardware de la interfaz local.
■	ID	CONN_OUC	2.0	1	Identifica una conexión en la CPU de modo unívoco.
■	ConnectionType	Byte	4.0	11	11=TCP/IP, 19=UDP (17=TCP/IP)
■	ActiveEstablished	Bool	5.0	false	Establecimiento de la conexión activa/pasiva
■	▼ RemoteAddress	IP_V4	6.0		Dirección IP remota (IPv4)
■	▼ ADDR	Array[1..4] of Byte	6.0		Dirección IPv4
■	ADDR[1]	Byte	6.0	192	192
■	ADDR[2]	Byte	7.0	168	168
■	ADDR[3]	Byte	8.0	0	0
■	ADDR[4]	Byte	9.0	1	1
■	RemotePort	UInt	10.0	0	Número de puerto UDP/TCP remoto
■	LocalPort	UInt	12.0	502	Número de puerto UDP/TCP local

**Figura 21.** Estructura TCON\_IP\_v4 del Servidor

### 3.2.1. Crear bloque de datos tipo Array del Cliente 1.

1. El parámetro interfaz, por defecto para PROFINET es 64, el ID debe ser igual para todos los dispositivos de la red, en el tipo de conexión para TCP siempre debe ser el valor 11 en decimal, la dirección IP debe ser la del cliente 1 que en este caso es IP: 192.168.0.1 y por último el puerto local para esta comunicación el valor estándar es 502.

Para enviar o recibir datos por el bloque MB\_SERVER en el parámetro MB\_HOLD\_REG este debe remitir siempre a un área de memoria mayor de dos bytes, para esto se crea un bloque de datos con una variable Array con un tipo de dato INT de 0..10. (Ver figura 22).

DATOS_TCP_S				
	Nombre	Tipo de datos	Offset	Valor de arranq...
1	Static			
2	DATOS	ay[0..10] of Int	0.0	
3	DATOS[0]			
4	DATOS[1]			
5	DATOS[2]			
6	DATOS[3]			
7	DATOS[4]			
8	DATOS[5]			
9	DATOS[6]			
10	DATOS[7]	Int	14.0	0
11	DATOS[8]	Int	16.0	0
12	DATOS[9]	Int	18.0	0
13	DATOS[10]	Int	20.0	0

Tipo de datos:

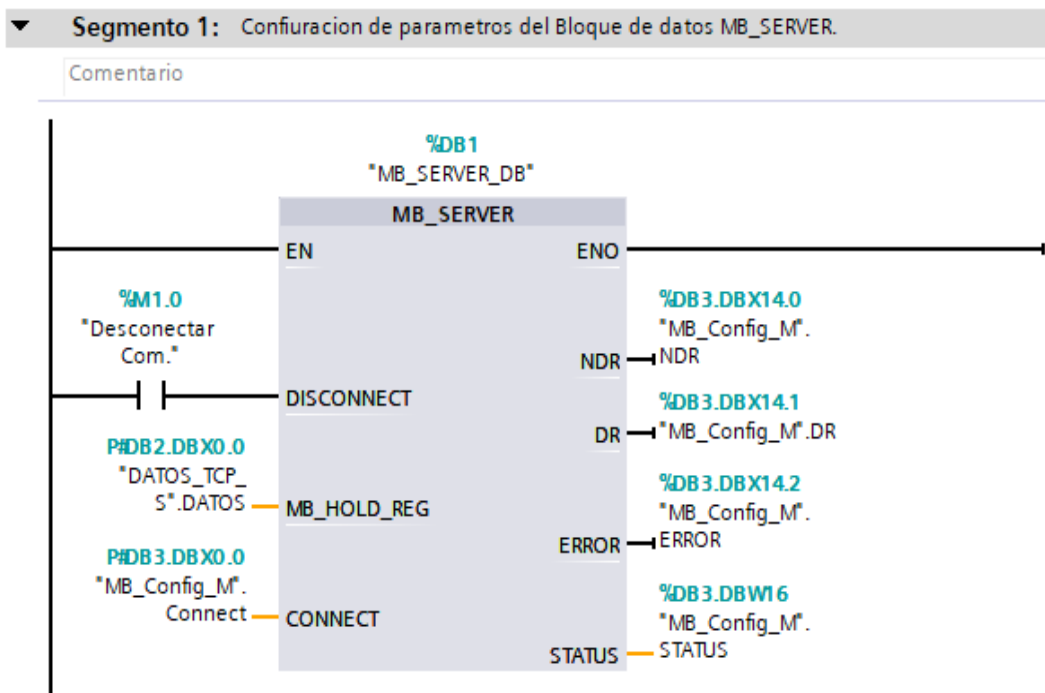
Límites de matriz:

Ejemplos: 0..99 ó 0..99,0..10

**Figura 22.** Estructura TCON\_IP\_v4 del Servidor

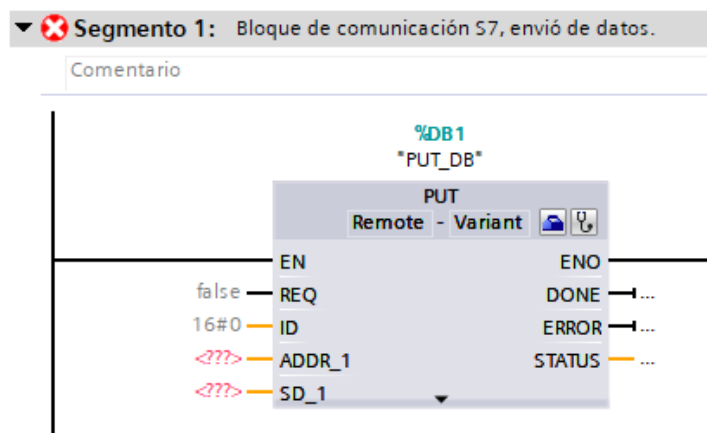
### 3.2.2. Programación de segmentos del Servidor.

1. En el segmento 1 se configuran los parámetros de comunicación integrados en el bloque MB\_SERVER. El parámetro DISCONNECT permite controlar cuándo se aceptará una solicitud de conexión, el parámetro MB\_HOLD\_REG se establece el Array de datos que en este caso esta denominado como *DATOS*, el parámetro CONNECT se establece con la estructura TCON\_IP\_v4 que está dentro del bloque de datos global.



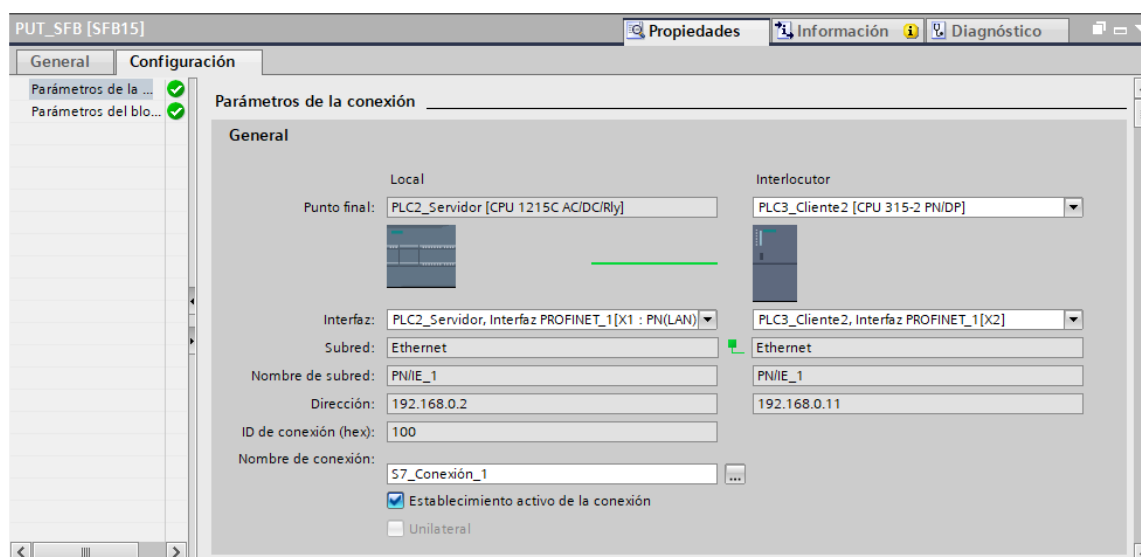
**Figura 23.** Estructura TCON\_IP\_v4 del Servidor

2. En el segmento 2 se configura el bloque PUT el cual envía datos al *PLC3\_Cliente\_2*, para obtener este bloque en el apartado de *Instrucciones* clic en **Comunicación**> **Comunicación S7** y seleccionar el bloque **PUT**, aparecerá una ventana para asignar el nombre y clic en **Aceptar**. (Ver figura 24).



**Figura 24.** Bloque de comunicación PUT en el servidor

3. Para configurar la comunicación entre el *PLC2\_Servidor* y el *PLC3\_Cliente\_2*, seleccionar el bloque, hacer clic derecho y seleccionar **Propiedades**, se despliega una ventana en la cual por defecto ya está configurado el PLC Local, en el apartado del interlocutor seleccionar el *PLC3\_Cliente\_2* ya que es el autómatas al cual se enviara la información, al seleccionar la interfaz de cada PLC la IP y la ID de conexión se establece automáticamente.



**Figura 25.** Parámetros de comunicación S7 (PUT) en el Servidor.

4. El bloque PUT trabaja con una dirección ADDR\_1 y SD\_1. ADDR\_1 es el puntero hacia las áreas de la CPU interlocutora en las que debe escribirse. SD\_1 es el puntero hacia las áreas de la CPU propia que contienen los datos que deben transmitirse.

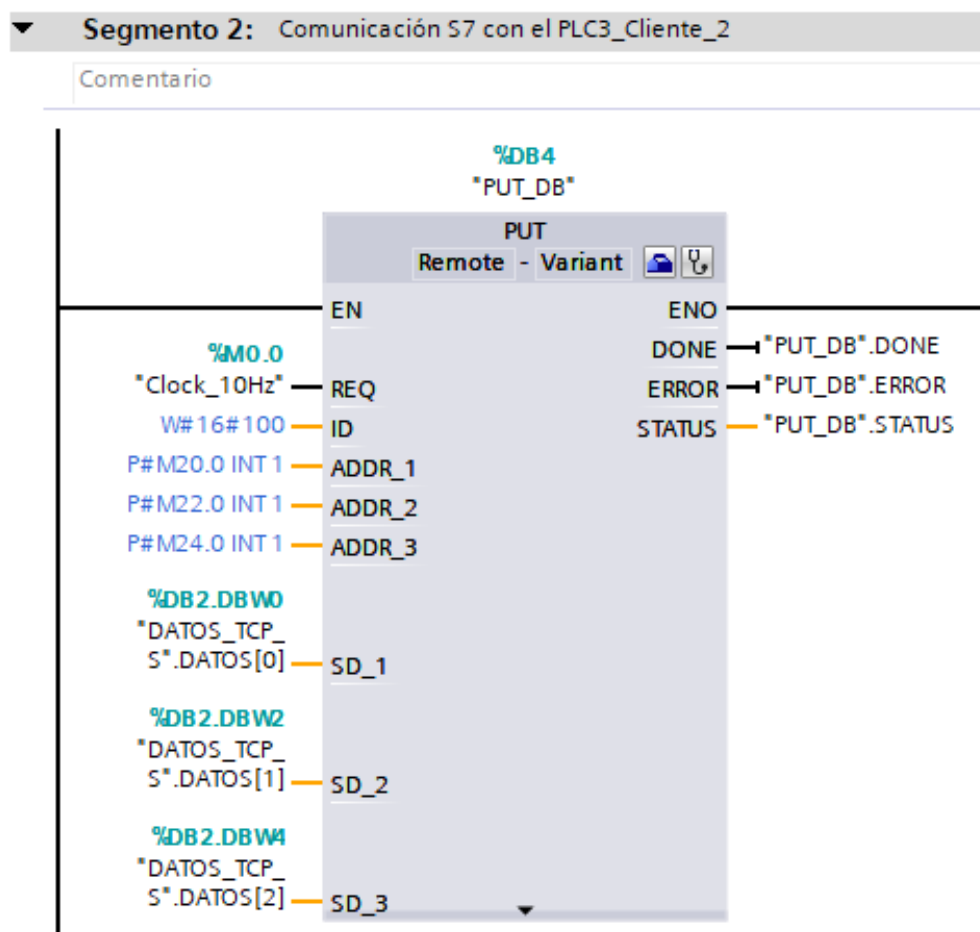
El parámetro REQ estará con una señal de reloj de 10Hz, esta señal permite recibir datos siempre que se encuentre en un flanco positivo, para el parámetro de la dirección primero se debe crear las variables que sean necesarias con una dirección que en este caso van a ser tres datos que se detallan en la figura 36.



10		dato_S1	Tabla de variables e..	Int	%MW20
11		dato_S2	Tabla de variables e..	Int	%MW22
12		dato_S3	Tabla de variables e..	Int	%MW24

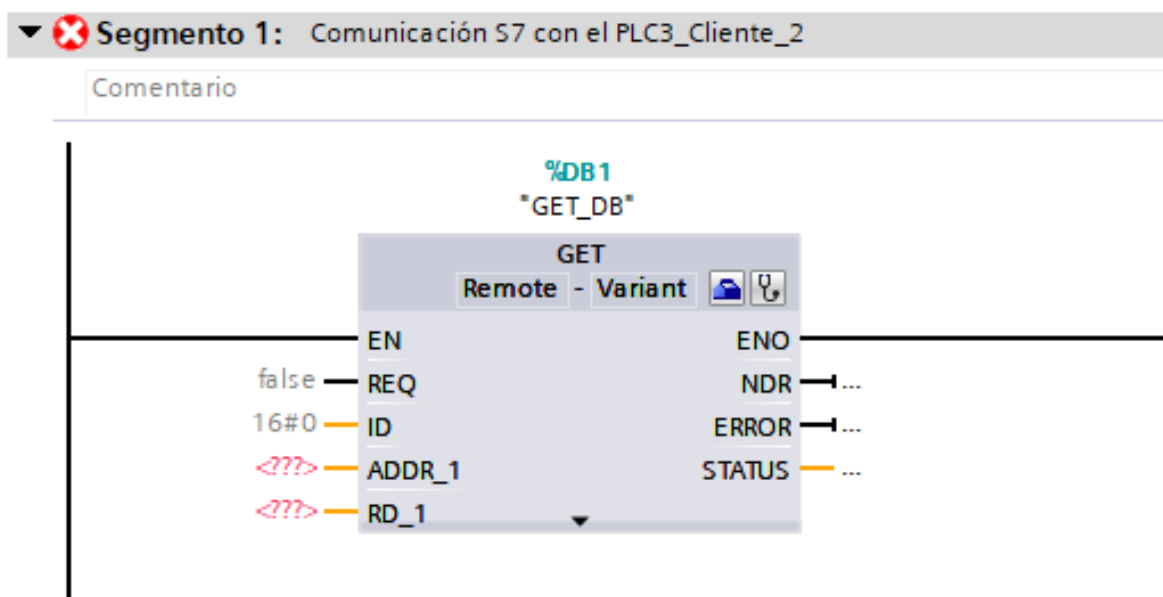
**Figura 26.** Variables para la transmisión de datos (PUT) del servidor

5. En este caso se tiene las direcciones MW20, MW22, MW 24, las mismas que deben ser escritas de la siguiente manera **P#M20.0 INT 1** (**M20.0** es la dirección, **INT** es el tipo de dato que en este caso es de tipo INT, **1** es la ID de conexión local), en el parámetro SD\_1 se configura con la dirección de los datos que se desean transmitir siguiendo el orden de las direcciones. (Ver figura 27.)



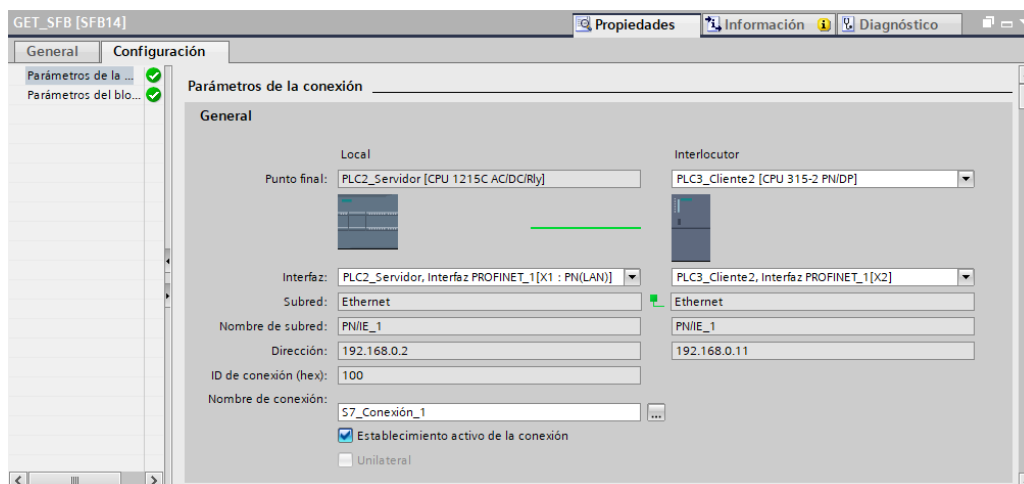
**Figura 27.** Configuración de la instrucción PUT del Servidor

6. En el segmento 3 se configura el bloque GET el cual recibe los datos del *PLC3\_Cliente\_2*, para obtenerlo, en el apartado de *Instrucciones* clic en **Comunicación**> **Comunicación S7** y seleccionar el bloque **GET**, aparecerá una ventana para asignar el nombre y clic en **Aceptar**. (Ver figura 28).



**Figura 28.** Bloque de comunicación GET en el servidor.

7. Para configurar la comunicación entre el *PLC2\_Servidor* y el *PLC3\_Cliente\_2*, seleccionar el bloque, hacer clic derecho y seleccionar **Propiedades**, se despliega una ventana en la cual por defecto ya está configurado el PLC Local, en el apartado del interlocutor seleccionar el *PLC3\_Cliente\_2* ya que es el PLC de donde el bloque va a recibir información, la interfaz de cada PLC, la IP y la ID de conexión se establece automáticamente. (Ver figura 29).



**Figura 29.** Parámetros de comunicación S7 (GET) en el Servidor

8. El bloque GET trabaja de forma similar que el bloque PUT, contiene los parámetros ADDR\_1 Y RD\_1, ADDR\_1 es el puntero hacia las áreas de la CPU interlocutora que se deben leer y RD\_1 es el puntero hacia las áreas de la CPU propia en las que se depositan los datos leídos.

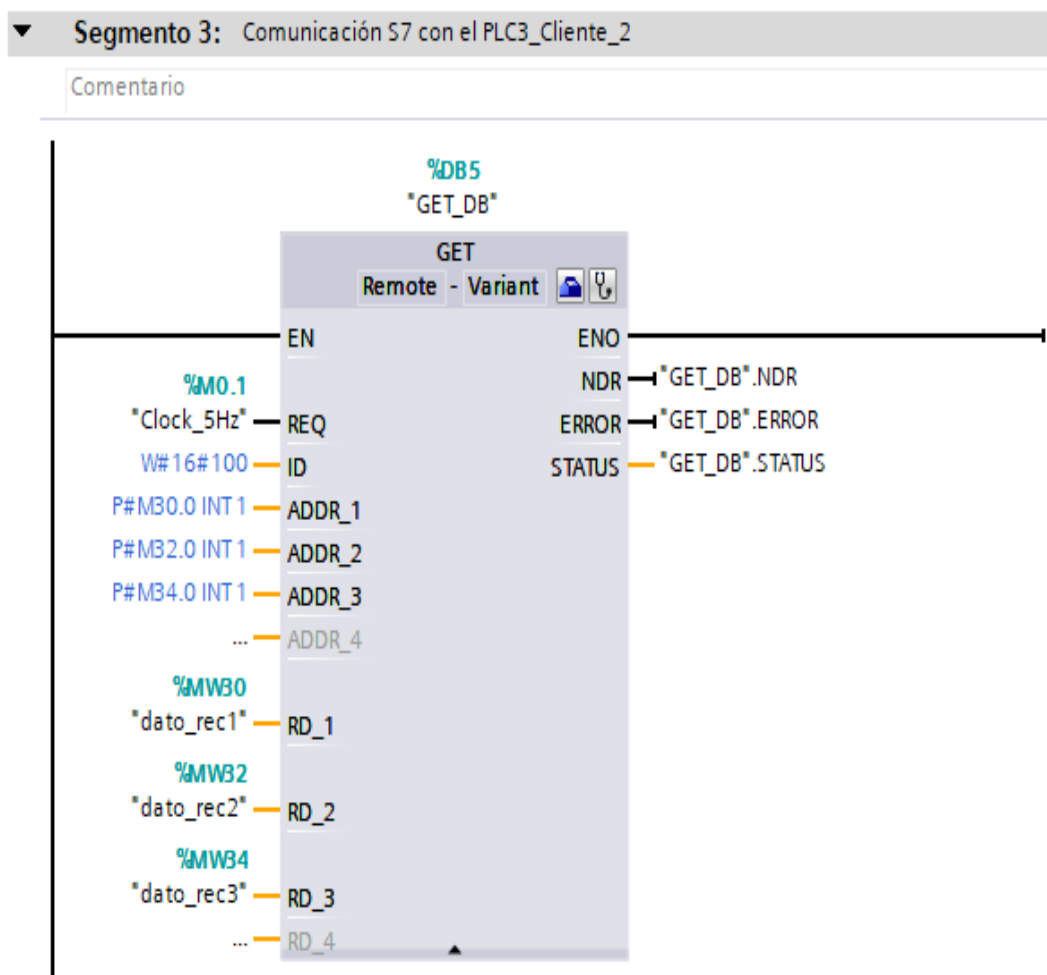
Se crea tres variables de tipo INT para depositar los datos adquiridos del PLC3\_Cliente\_2, estas variables están detalladas en la figura 30.

15		dato_rec1	Tabla de variables estándar	Int	%MW30
16		dato_rec2	Tabla de variables estándar	Int	%MW32
17		dato_rec3	Tabla de variables estándar	Int	%MW34

**Figura 30.** Variables para la transmisión de datos (GET)

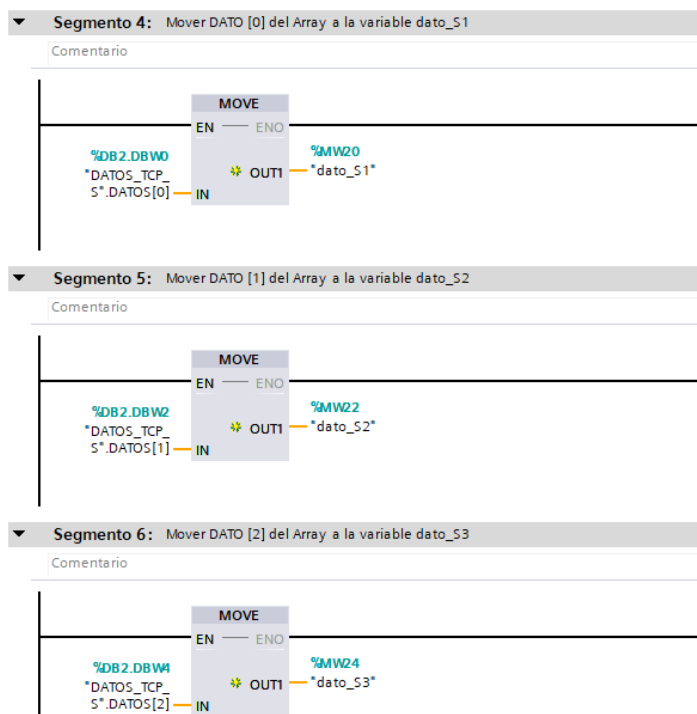
Los datos a recibir se van a depositar en la dirección MW30, MW32 y MW34, las mismas que ayudan a establecer las direcciones en el bloque GET. En el parámetro RD\_1 se asigna la dirección de las variables de la figura 30.

En la figura 31 se puede observar los parámetros con las direcciones asignadas para el depósito de datos adquirida por el bloque GET.



**Figura 31.** Configuración del bloque de comunicación GET del PLC 2.

9. Para finalizar la programación en el PLC2 se integra el segmento 4, 5, 6 con la finalidad de mover los datos del Array a la tabla de variables para ser observadas en conjunto.



**Figura 32.** Mover los datos Array a una variable

10. Al terminar de realizar las condiciones en el *PLC\_Server* se observa todas las variables que se han creado para comprobar las direcciones que manejaran los bloques GET y PUT. (Ver figura 33).

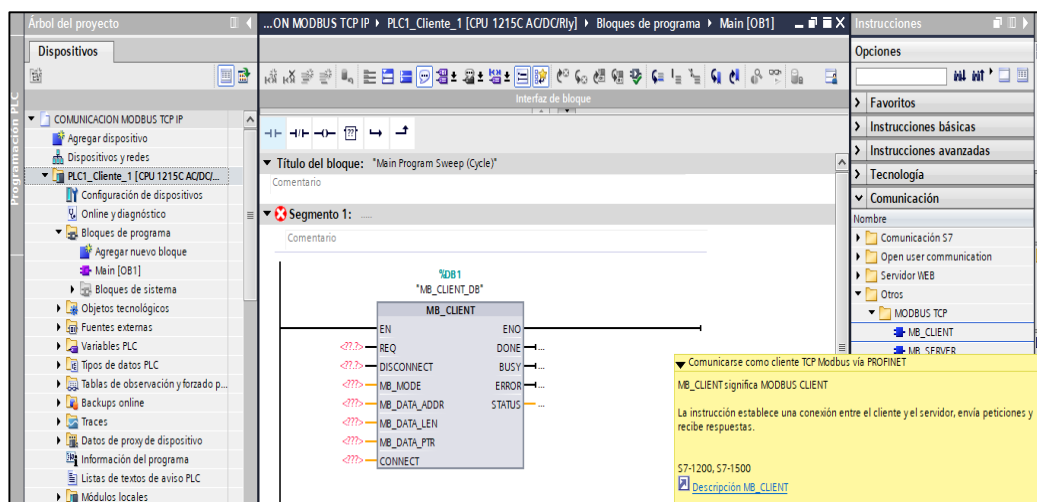
1	Clock_Byte	Tabla de variabl...	Byte	%MBO
2	Clock_10Hz	Tabla de variables e..	Bool	%M0.0
3	Clock_5Hz	Tabla de variables e..	Bool	%M0.1
4	Clock_2.5Hz	Tabla de variables e..	Bool	%M0.2
5	Clock_2Hz	Tabla de variables e..	Bool	%M0.3
6	Clock_1.25Hz	Tabla de variables e..	Bool	%M0.4
7	Clock_1Hz	Tabla de variables e..	Bool	%M0.5
8	Clock_0.625Hz	Tabla de variables e..	Bool	%M0.6
9	Clock_0.5Hz	Tabla de variables e..	Bool	%M0.7
10	dato_S1	Tabla de variables e..	Int	%MW20
11	dato_S2	Tabla de variables e..	Int	%MW22
12	dato_S3	Tabla de variables e..	Int	%MW24
13	Desconectar Com.	Tabla de variables e..	Bool	%M1.0
14	Tag_1	Tabla de variables e..	Bool	%M1.3
15	dato_rec1	Tabla de variables e..	Int	%MW30
16	dato_rec2	Tabla de variables e..	Int	%MW32
17	dato_rec3	Tabla de variables e..	Int	%MW34

**Figura 33.** Variables del PLC 2

### 3.3. Programación del Cliente 1.

1. Verificar la conexión PROFINET y la IP configurada a cada dispositivo para iniciar con la programación del *PLC1\_Cliente\_1*.

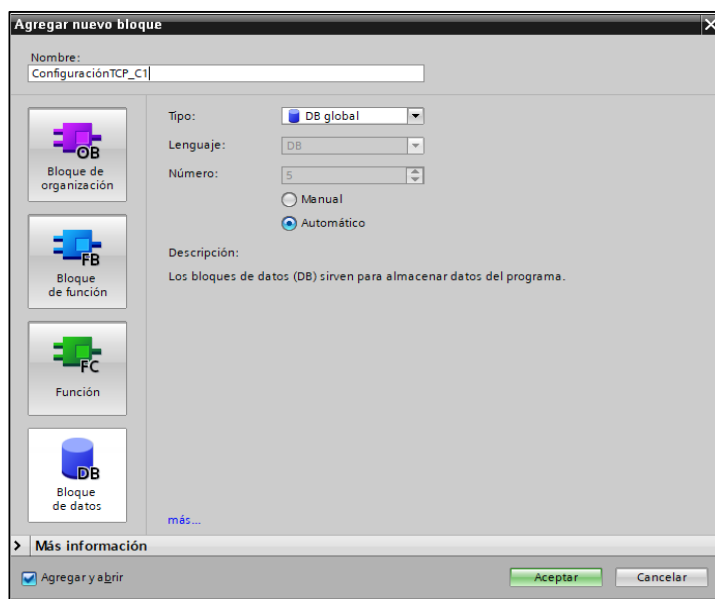
El *PLC1\_Cliente\_1* se configurara como cliente 1 en la red Modbus TCP. TIA PORTAL; posee bloques compactos que permiten realizar esta comunicación de forma más sencilla entre dos PLCs S7 1200 sin la necesidad de ninguna licencia y de ningún modulo adicional. En el apartado de *Dispositivos* detallada en la figura 34 seleccionar **PLC1\_Cliente\_1** > **Bloques de programa** > **Main [OB1]**, posteriormente se despliega una ventana de instrucciones ubicada en la parte derecha, clic en **Comunicación** > **Otros** > **MODBUS TCP**, seleccionar el bloque *MB\_CLIENT*



**Figura 34.** Bloque Modbus TCP, MB\_CLIENT.

2. Los parámetros de comunicación Modbus TCP deben estar dentro de un bloque de datos, bloque en el cual bajo la estructura *TCON\_IP\_v4* se debe indicar solamente conexiones del tipo TCP. Para la configuración de conexión TCP, en el apartado de dispositivos, clic en **PLC1\_Cliente\_1** > **Bloque de programa** > **Agregar nuevo bloque**, aparecerá una ventana en la

cual se debe seleccionar *Bloque de datos*, se asigna el nombre *ConfiguraciónTCP\_C1* y clic en **Aceptar**. (Ver figura 35).



**Figura 35.** Bloque de datos para configurar de conexión TCP.

3. Dentro del bloque de datos global denominado *ConfiguraciónTCP\_C1*, se crea un parámetro con el nombre de *CONNECT* y en el tipo de dato se asigna la estructura *TCON\_IP\_v4* detallada en la figura 36.

En la interfaz PROFINET de los PLCs por defecto es 64, es muy importante que la ID de todos los PLCs conectados la red sea igual a 1, el tipo de conexión que plantea este bloque de comunicación aplica solo para TCP con el valor decimal 11, si se desea realizar otra comunicación como UDP este parámetro se debe dejar vacío, en el parámetro *ActiveEstablished* se asina un valor booleano (TRUE) para el establecimiento activo de la conexión, posteriormente se asina la dirección IP del interlocutor (Servidor) que en este caso es la IP:192.168.0.2, en el puerto remoto TCP trabaja con un valor estándar del puerto 502.

■	▼ CONNECT	TCON_IP_v4	8.0		<input type="checkbox"/>	
■	Interfaceld	HW_ANY	8.0	64	<input type="checkbox"/>	Local-Interfaz_PROFINET_1.
■	ID	CONN_OUC	10.0	1	<input type="checkbox"/>	Identifica una conexión en la CPU de modo unívoco.
■	ConnectionType	Byte	12.0	11	<input type="checkbox"/>	11=TCP/IP, 19=UDP (17=TCP/IP)
■	ActiveEstablished	Bool	13.0	true	<input type="checkbox"/>	Establecimiento de la conexión activa/pasiva
■	▼ RemoteAddress	IP_V4	14.0		<input type="checkbox"/>	Dirección IP remota (IPv4)
■	▼ ADDR	Array[1..4] of Byte	14.0		<input type="checkbox"/>	Dirección IPv4
■	ADDR[1]	Byte	14.0	192	<input type="checkbox"/>	192
■	ADDR[2]	Byte	15.0	168	<input type="checkbox"/>	168
■	ADDR[3]	Byte	16.0	0	<input type="checkbox"/>	0
■	ADDR[4]	Byte	17.0	2	<input type="checkbox"/>	2
■	RemotePort	UInt	18.0	502	<input type="checkbox"/>	Número de puerto UDP/TCP remoto
■	LocalPort	UInt	20.0	0	<input type="checkbox"/>	Número de puerto UDP/TCP local

**Figura 36.** Estructura TCON\_IP\_v4 del Cliente 1.

En este bloque de datos también se integra el resto de parámetros a configurar para el bloque MB\_CLIENT detallada en la figura 37, en las entradas: DISCONNECT, MB\_MODE, DATA\_ADDR, DATA\_LEN y salidas: DONE, BUSY, ERROR y STATUS.

1		▼ Static				
2		■ DISCONNECT	Bool	0.0	false	
3		■ MB_MODE	USInt	1.0	1	
4		■ DATA_ADDR	UDInt	2.0	40001	
5		■ DATA_LEN	UInt	6.0	10	
6		■ ► CONNECT	TCON_IP_v4	8.0		
7		■ DONE	Bool	22.0	false	
8		■ BUSY	Bool	22.1	false	
9		■ ERROR	Bool	22.2	false	
10		■ STATUS	Word		24.0	16#0

**Figura 37.** Parámetros configurados del Cliente 1.



### 3.3.1. Crear bloque de datos tipo Array del Cliente 1.

1. Los datos a transmitir van a ser de estructura Array, en el apartado de *Dispositivos* clic en **PLC1\_Cliente\_1 > Bloques de programa > Agregar nuevo bloque**, aparecerá una ventana en donde se debe seleccionar *Bloque de datos*, se asigna el nombre del bloque como **DATOS\_TCP\_C1** y clic en **Aceptar**. Dentro del bloque se asigna una variable denominada **DATOS**, y en el tipo de dato seleccionar Array de tipo **INT** y por último en límites de matriz en este caso seleccionar de **0..10**. Estos parámetros se pueden observar en la figura 38.

DATOS_TCP_C1				
	Nombre	Tipo de datos	Offset	Valor de arranq...
1	Static			
2	DATOS	Array[0..10] of Int	0.0	
3	DATOS[0]			
4	DATOS[1]			
5	DATOS[2]			
6	DATOS[3]			
7	DATOS[4]			
8	DATOS[5]			
9	DATOS[6]			
10	DATOS[7]	Int	14.0	0
11	DATOS[8]	Int	16.0	0
12	DATOS[9]	Int	18.0	0
13	DATOS[10]	Int	20.0	0

**Figura 38.** Bloque de datos, DATOS\_TCP\_C1

2. Para proporcionar una elección al usuario en la interfaz Win CC si se desea leer o escribir desde el cliente al servidor se debe configurar el estado del parámetro **MB\_MODE** integrado en el bloque **MB\_CLIENT**, mismo que posee un tipo de dato **USInt**, 8 bits, enteros sin signo, para ello se ha creado un bloque de datos global denominado **LEER O ESCRIBIR** en donde el valor de 0 (cero) es para leer y el valor de 1 (uno) es para escribir. Para esto se integra un nuevo bloque de datos global con el nombre **LEER O ESCRIBIR**, estos datos se pueden observar en la figura 39.

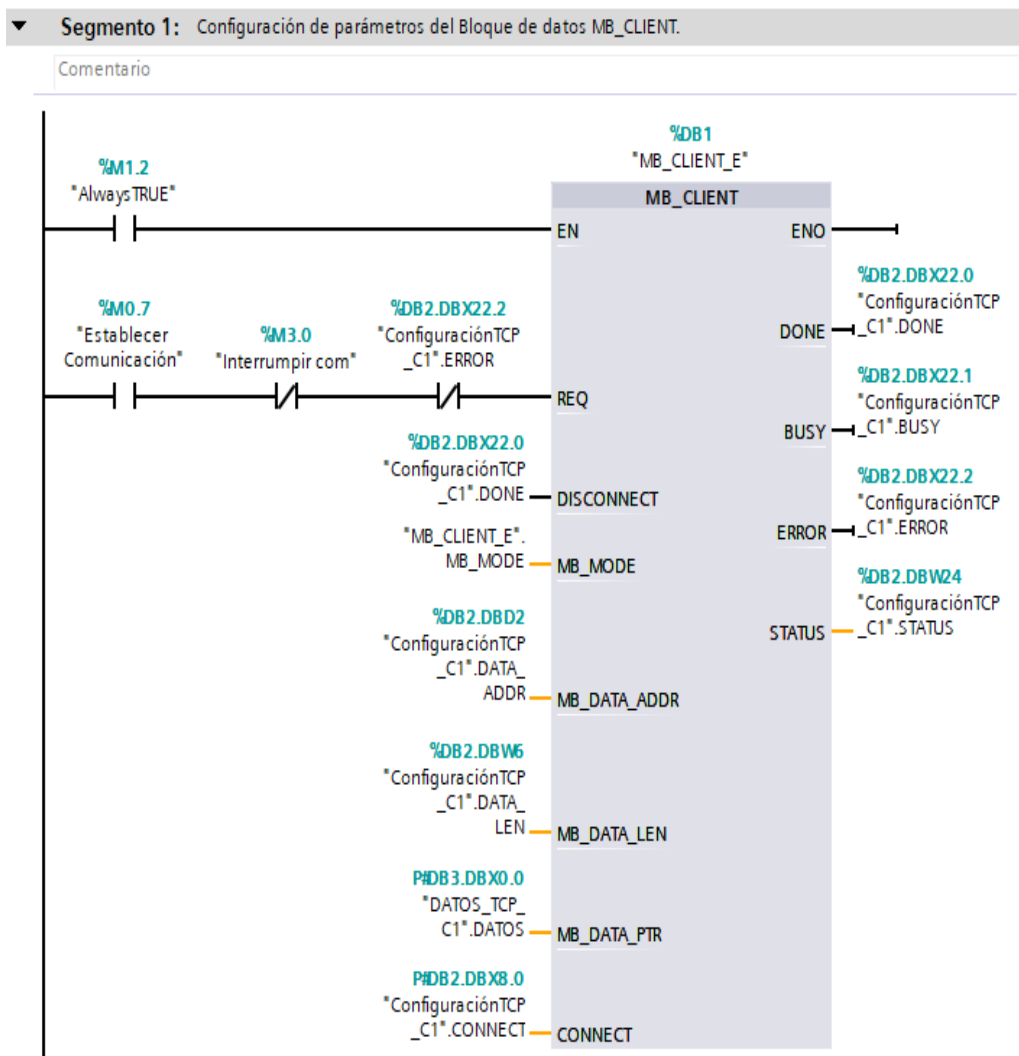
LEER O ESCRIBIR			
	Nombre	Tipo de datos	Valor de arranq...
1	Static		
2	LEER	USInt	0
3	ESCRIBIR	USInt	1

*Figura 39.* Bloque de datos para la condición de leer o escribir.

### 3.3.2. Programación de segmentos del Cliente 1.

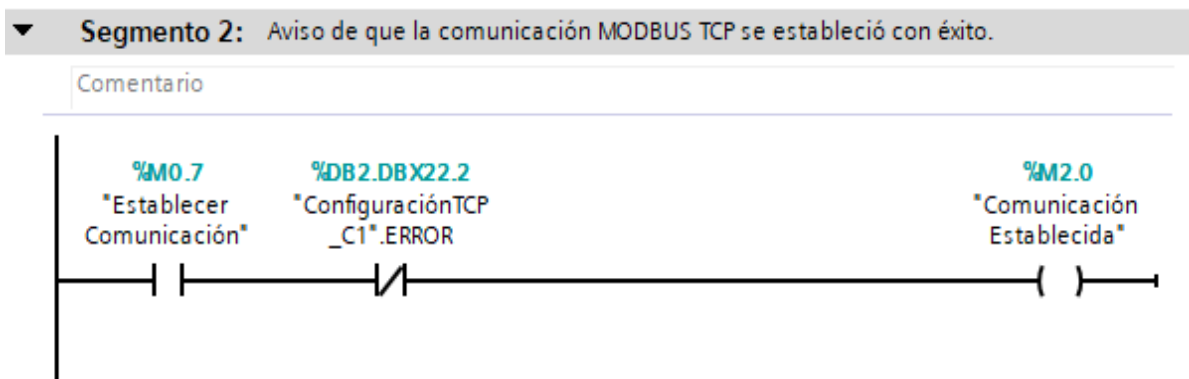
1. Después de haber terminado de crear todos los bloques de datos globales con las respectivas configuraciones, dentro de bloques de programa *Main [OBI]* se empieza a programar en los segmentos. En el segmento 1 se encuentra el bloque de comunicación MB\_CLIENT del cual los parámetros ya se han detallado anteriormente, el parámetro EN de la entrada del bloque está conectado en serie con el contacto de la marca M1.2 (Always TRUE), esta variable es creada automáticamente al momento que se activa la pestaña *Marcas del sistema*. El parámetro REQ es controlado por niveles, es decir que mientras la entrada este activa (REQ=TRUE) esta instrucción estará enviando peticiones de comunicación, estos parámetros están detallados en la figura 40.

Para este parámetro se ha integrado un contacto normalmente abierto (NA) y dos contactos normalmente cerrados (NC) todos estos conectados en serie, el contacto NA nos permite establecer la comunicación al momento que pase del estado 0 (cero) al estado 1 (uno), a la vez esta señal booleana será interrumpida con dos contactos NC, el primero denominado *Interrumpir comunicación* es un contacto que ayudara en la interfaz Win CC a deshacer la comunicación y el segundo es un contacto de la salida del bloque MB\_CLIENT denominado *ERROR* el cual al detectar un error automáticamente la comunicación será interrumpida y desconectada hasta que el usuario de solución al código de error que se presenta en la salida *STATUS*.



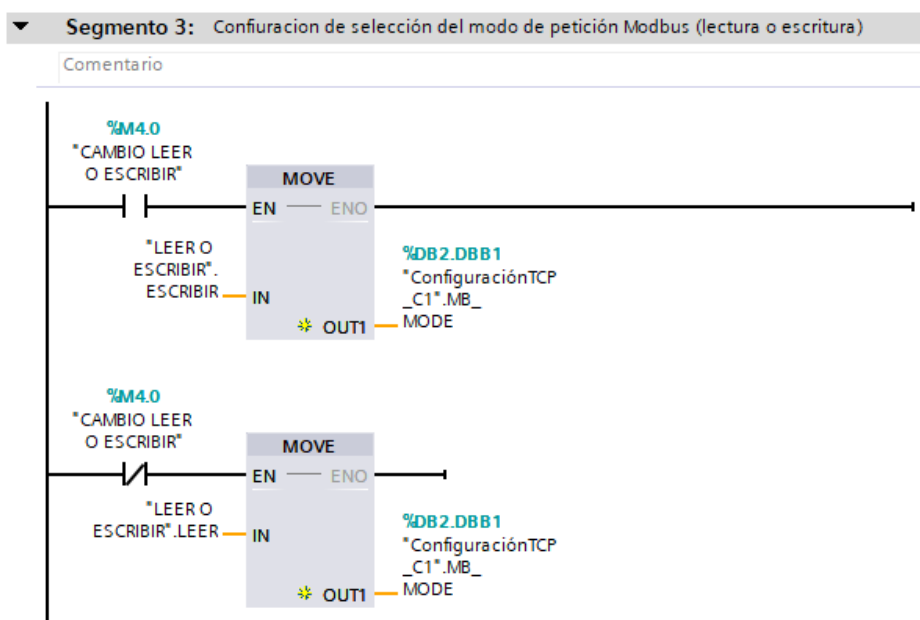
**Figura 40.** Configuración de la instrucción MB\_CLIENT

2. En el segmento 2 se realiza una señal de aviso bajo una marca conectada en serie con el contacto encargado de establecer la comunicación y con el contacto NC quien interrumpe la comunicación al detectar algún error. Esta marca M2.0 denominada *Comunicación Establecida* será visualizada en la interfaz Win CC para facilitar el estado de la comunicación al operador. (Ver figura 41).



**Figura 41.** Condición para la señal de comunicación establecida.

3. En este segmento se establece la condición que permite seleccionar con mayor facilidad la lectura o escritura en la interfaz Win CC. Esta condición se ha realizado con un contacto NA y un contacto NC, los dos contactos con la misma marca M4.0 así cuando se activa el contacto la condición de leer a escribir se activara automáticamente y viceversa, se ha utilizado el bloque MOVE el cual nos ayuda a mover un dato de un lugar a otro ya que este parámetro trabaja con un tipo de dato USInt. (Ver figura 42).



**Figura 42.** Condición que permite la selección de lectura o escritura.

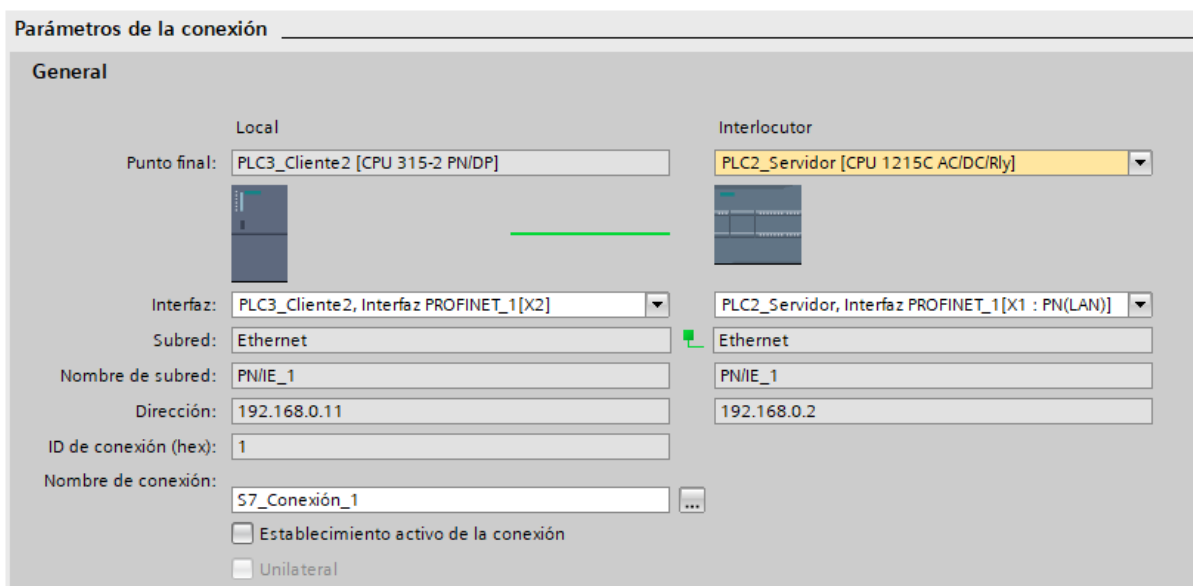
4. Al terminar de programar los segmentos del PLC las variables creadas se pueden observar en el apartado de *Dispositivos*, clic en **PLC1\_Cliente\_1** > **Variables PLC** > **Mostrar todas las variables**, allí se pueden observar todas las variables que se han creado para el cliente 1. (Ver figura 43).

	Nombre	Tabla de variables	Tipo de datos	Dirección
1	System_Byte	Tabla de variabl... ▼	Byte	%MB1 ▼
2	FirstScan	Tabla de variables e..	Bool	%M1.0
3	DiagStatusUpdate	Tabla de variables e..	Bool	%M1.1
4	AlwaysTRUE	Tabla de variables e..	Bool	%M1.2
5	AlwaysFALSE	Tabla de variables e..	Bool	%M1.3
6	Establecer Comunicación	Tabla de variables e..	Bool	%M0.7
7	Comunicación Establecida	Tabla de variables e..	Bool	%M2.0
8	Interrumpir com	Tabla de variables e..	Bool	%M3.0
9	CAMBIO LEER O ESCRIBIR	Tabla de variables e..	Bool	%M4.0

*Figura 43.* Tabla de variables del PLC 1.

### 3.4. Programación del Cliente 2.

En el PLC3\_Cliente\_2 se establece la comunicación con el Servidor con el bloque GET y PUT, para extraer el bloque GET revisar la figura 28, para establecer la comunicación ir a propiedades del bloque en donde el PLC local ya está configurado por defecto, en el interlocutor seleccionar el PLC2\_Servidor, la dirección IP es reconocida y la ID de conexión se crea automáticamente. (Ver figura 44).



**Figura 44.** Parámetros de comunicación S7 (GET) en el Cliente\_2.

### 3.4.1. Crear variables para los datos a transmitir del Cliente\_2.

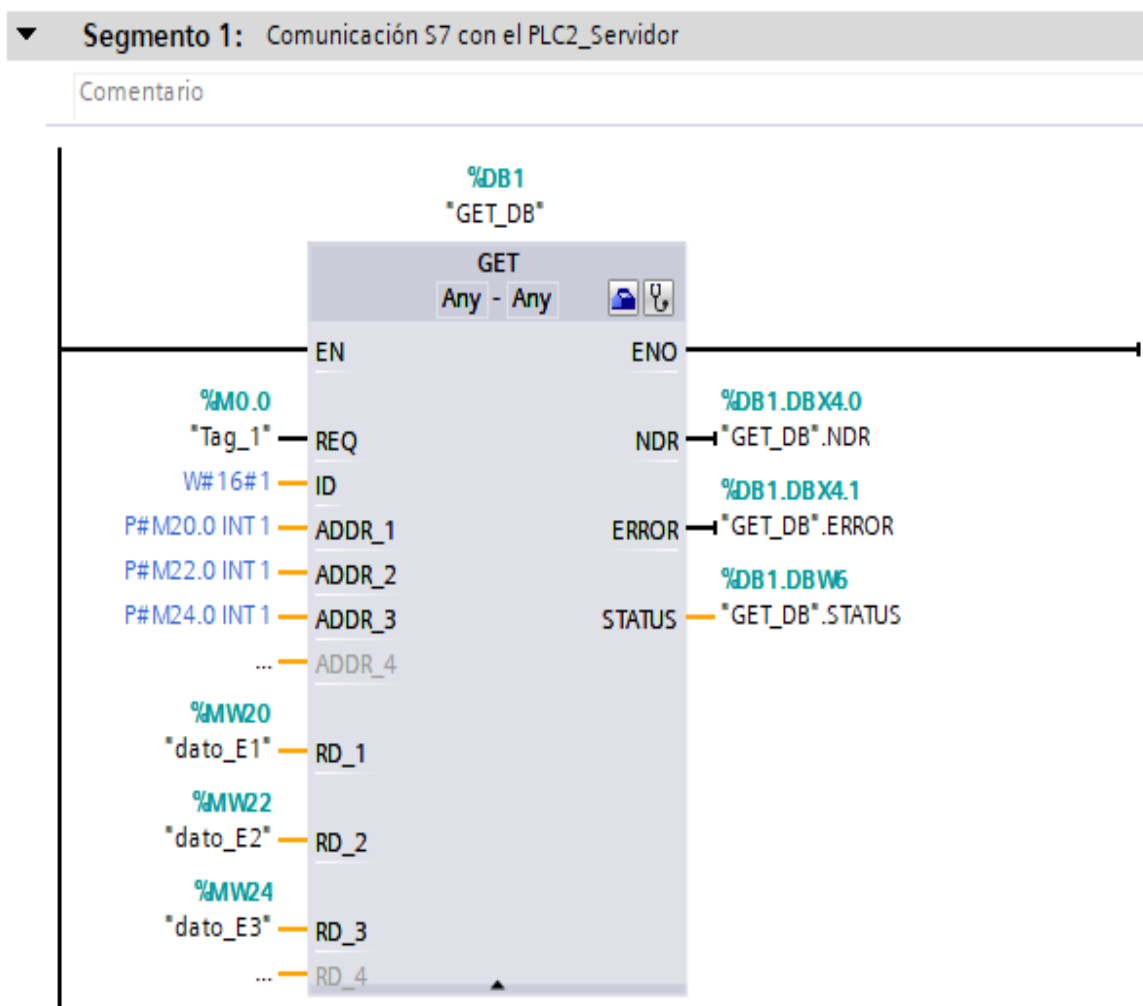
Para configurar los parámetros del bloque GET necesariamente se debe crear variables para depositar los datos a recibir. Estas variables contienen un tipo de dato INT en la dirección MW20, MW22 y MW24. (Ver figura 45).

2		dato_E1	Int	Tabla de variables estándar	%MW20
3		dato_E2	Int	Tabla de variables estándar	%MW22
4		dato_E3	Int	Tabla de variables estándar	%MW24

**Figura 45.** Variables para la transmisión de datos (GET) en el Cliente\_2

### 3.4.2. Programación de segmentos del Cliente\_2.

1. Las direcciones asignadas son establecidas en los parámetros de dirección ADDR\_1 y los parámetros RD\_1 contienen la dirección de las variables de depósito de los datos recibidos, estos valores están detallados en la figura 46.



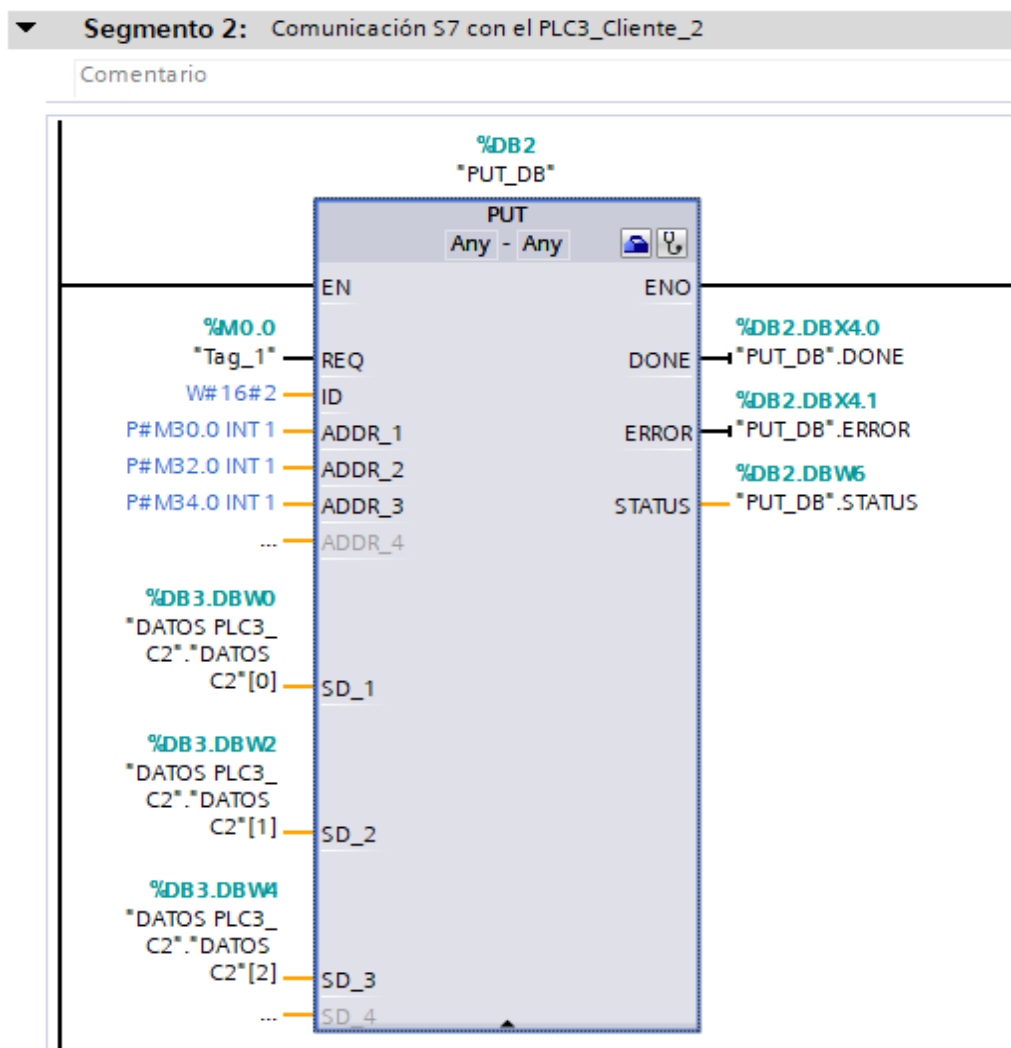
**Figura 46.** Configuración del bloque de comunicación GET del PLC 3.

2. Para extraer el bloque PUT revisar la figura 24, se crea variables para adquirir las direcciones y establecerlas en el bloque PUT en el PLC3\_Cliente\_2, estas variables se pueden observar en la figura 47.

5		dato_Env1	Int	Tabla de variables estándar	%MW30
6		dato_Env2	Int	Tabla de variables estándar	%MW32
7		dato_Env3	Int	Tabla de variables estándar	%MW34

**Figura 47.** Variables para la transmisión de datos (PUT) en el Cliente\_2

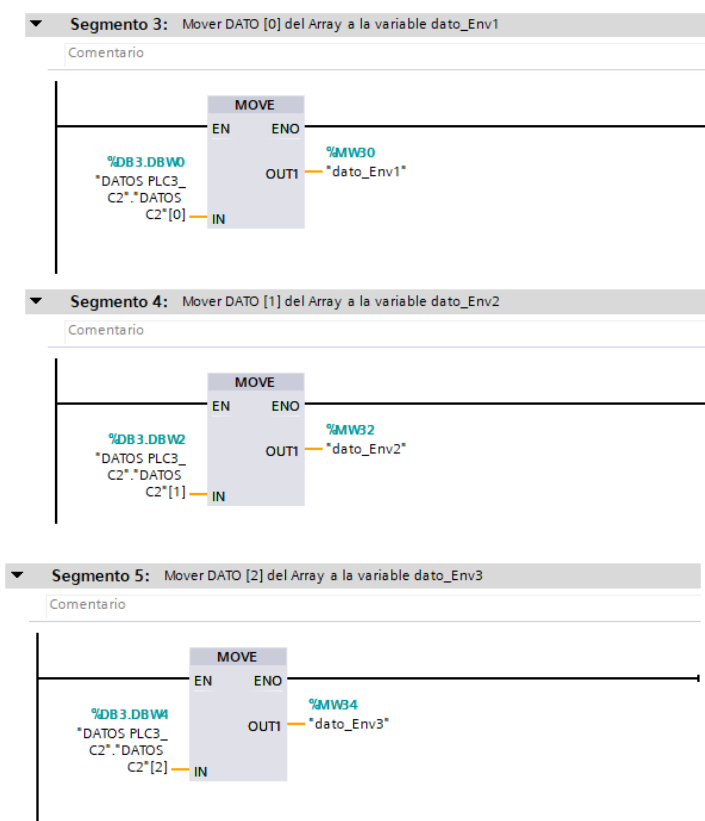
3. Las direcciones MW30, MW 32 y MW34 son establecidas en el bloque de comunicación PUT detallada en la figura 48.



**Figura 48.** Configuración del bloque de comunicación PUT del PLC 3.

4. El segmento 3, 4 y 5 condicionan el movimiento de datos a una variable para observar toda la información transmitida de manera conjunta.





**Figura 49.** Mover los valores enviados a una variable del PLC3.

5. Las variables establecidas en el PLC3\_Cliente\_2 se pueden observar en la figura 50, las direcciones se tienen que verificar con los parámetros de los bloques GET y PUT.

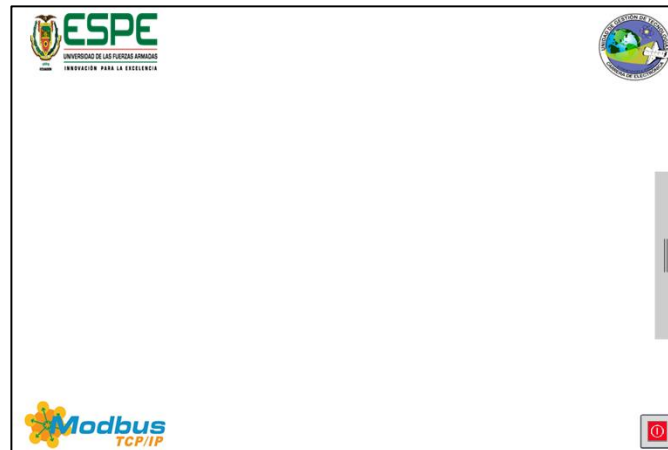
1		Tag_1	Bool		Tabla de variables está...	%M0.0
2		dato_E1	Int		Tabla de variables estándar	%MW20
3		dato_E2	Int		Tabla de variables estándar	%MW22
4		dato_E3	Int		Tabla de variables estándar	%MW24
5		dato_Env1	Int		Tabla de variables estándar	%MW30
6		dato_Env2	Int		Tabla de variables estándar	%MW32
7		dato_Env3	Int		Tabla de variables estándar	%MW34
8		Clock_5Hz	Bool		Tabla de variables estándar	%M0.1

**Figura 50.** Tabla de variables del cliente 2

### 3.5. Programación de pantalla HMI en Win CC.

#### 3.5.1. Crear plantilla

Una vez creada todas las variables en el servidor y en los dos clientes, los datos a transmitirse van a ser visualizados en una pantalla HMI Win CC integrada en TIA PORTAL. Primero se crea una plantilla, en el apartado de *Dispositivos* clic en **PC-System\_1> HMI\_ModbusTCP> Administración de imágenes> Plantillas> Agregar plantilla**, automáticamente se crea una plantilla a la cual se asigna el nombre *Plantilla\_ModbusTCP*, adicional se inserta 3 imágenes y un botón ExitRuntime, este botón se encuentra en el apartado de *Librerías* clic en **Librerías globales> Buttons-and-Switches> Plantillas maestras> SystemButtons> Rectangular\_large** y seleccionar *SB\_ExitRuntime\_85x64*, la plantilla se puede observar en la figura 51.

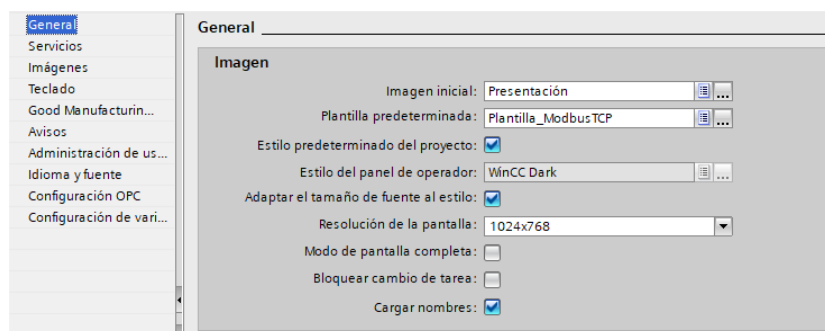


*Figura 51.* Plantilla\_ModbusTCP

#### 3.5.2. Crear imágenes.

1. En el apartado de Dispositivos clic en **PC-System\_1> HMI\_ModbusTCP> Imágenes> Agregar imagen**, en la interfaz Win CC se necesitara tres imágenes: Presentación, Establecer Comunicación y Modbus TCP/IP. Estas tres imágenes serán representadas en un menú dentro de una imagen deslizable a la derecha, para configurar la imagen, en el apartado de *Dispositivos* clic

en **PC-System\_1> HMI\_ModbusTCP> Configuración de runtime**, se despliega una ventana con las configuraciones de la imagen, en el menú **General> Imagen** se establece la imagen inicial que en este caso es **Presentación**, en plantilla predeterminada seleccionar *Plantilla\_ModbusTCP*, en la resolución de la pantalla seleccionar el tamaño de la imagen deseada que en este caso es de 1024x768, desactivar la pestaña de pantalla completa y cerrar la ventana. (Ver figura 52).



**Figura 52.** Configuración de plantilla.

2. La configuración de runtime se realiza una sola vez para todas las imágenes, al finalizar la imagen quedara con la plantilla seleccionada y adicional a esto se inserta texto. (Ver figura 53).



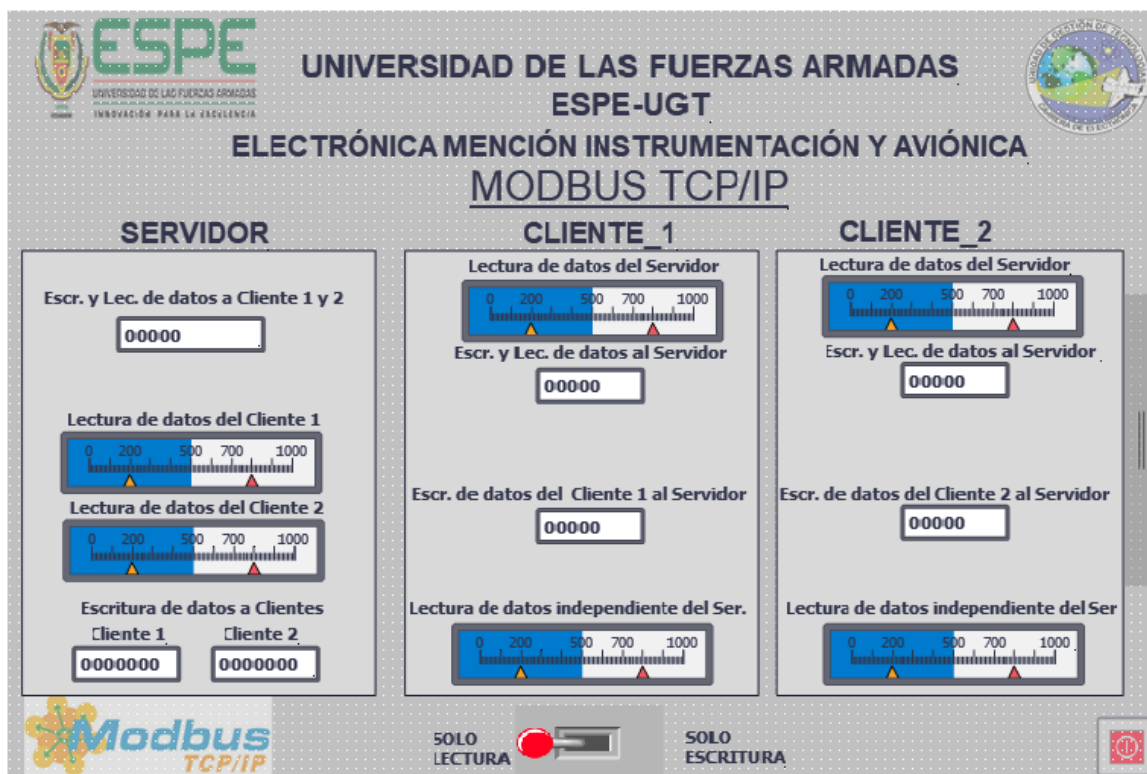
**Figura 53.** Presentación de interfaz HMI

3. Insertar una nueva imagen (ver paso 39) a la cual se asigna el nombre de *Establecer comunicación*. En esta imagen se inserta un botón encargado de establecer la comunicación con un indicador visual en la parte inferior denominado *Comunicación Establecida* y otro botón para *Deshacer la Comunicación*, cada PLC tiene un indicador de estado, el cual muestra el tráfico de datos según los estados de cada PLC. (Ver figura 54).



**Figura 54.** Imagen HMI para establecer comunicación.

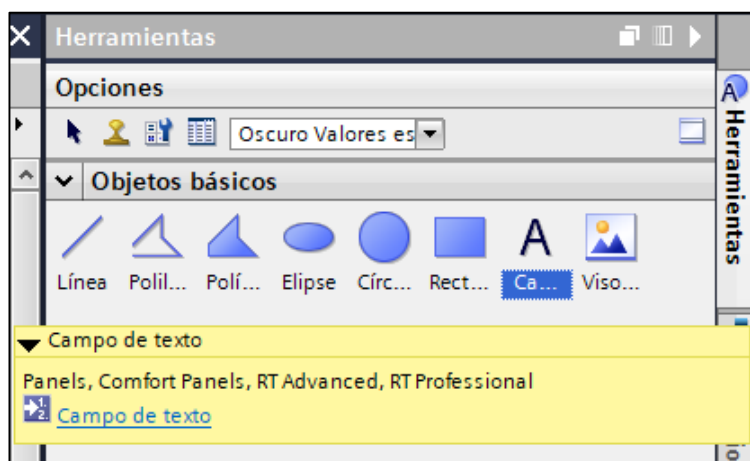
4. La imagen *MODBUS TCP/IP* integra todos los datos de lectura y escritura representados mediante controles e indicadores dentro de la imagen *MODBUS TCP*, estos datos están clasificados verticalmente según los datos de cada PLC. (Ver figura 55). Adicional a esto se suma un selector de lectura o escritura ubicado en la parte inferior de la imagen.



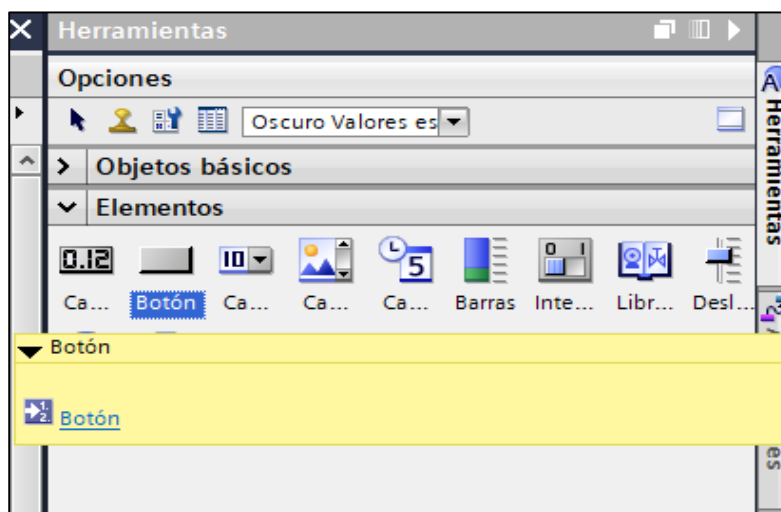
**Figura 55.** Transmisión de datos entre el servidor y los dos clientes

5. Insertar una imagen deslizable, para ello en el apartado de *Dispositivos* clic en **PC-System\_1**> **HMI\_ModbusTCP**> **Administración de imágenes**> **Imágenes deslizables** y seleccionar **Imagen deslizable a la derecha**, editar esta imagen con un menú de *Inicio* de 3 botones (Presentación, Establecer Comunicación y Modbus TCP).

Para añadir texto, seleccionar el icono de *Campo de texto* en el apartado de Herramientas sección *Objetos básicos*, (Ver figura 56). Adicional se agregan tres botones que se encuentran en el apartado de **Herramientas**> **Elementos** y seleccionar *Botón*, (Ver figura 57).

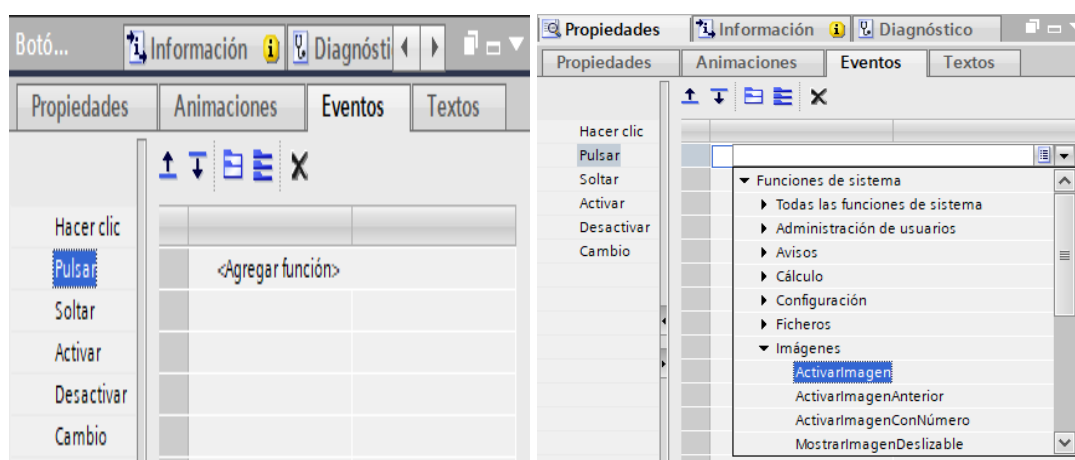


*Figura 56.* Icono campo de texto



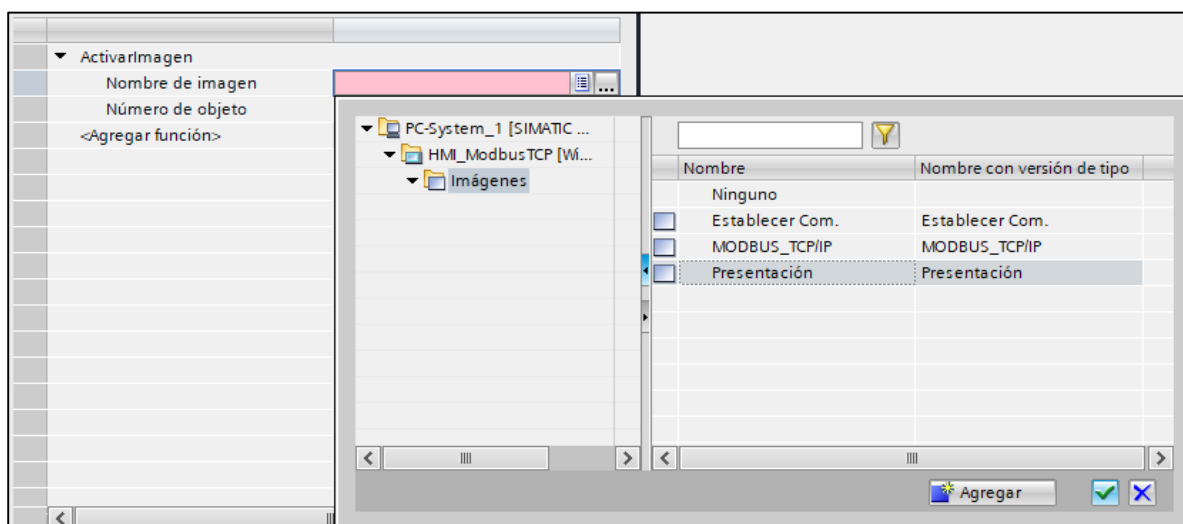
*Figura 57.* Icono Botón

6. Para elegir la posición del texto o el botón basta con arrastrar el icono al lugar deseado, la intención con los botones es abrir una imagen cuando este sea presionado, Para el botón 1 denominado *Presentación*, seleccionar el botón y hacer clic en **Propiedades**> **Eventos** seleccionar **Pulsar**. (Ver figura 58), clic en **Agregar función**> **Funciones del sistema**> **Imágenes**> **Activar imagen**. (Ver figura 59),



**Figura 58.** Configuración del botón 1

7. Automáticamente aparecen dos parámetros (Nombre de la imagen y Número del objeto), en el nombre del objeto seleccionar la imagen **Presentación** y clic en el icono del **visto**, (Ver figura 59). Repetir este paso para los dos botones restantes.



**Figura 59.** Configurar el botón para añadir una imagen.

8. Después de concluir con la configuración de los tres botones de inicio, se obtendrá una imagen que se muestra en la figura 60.

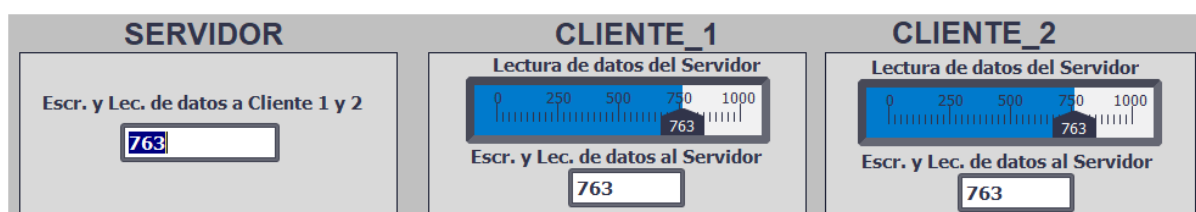


**Figura 60.** Imagen deslizable a la derecha.

### 3.6. Pruebas de transmisión de datos en la red Modbus TCP/IP.

#### 3.6.1. Transmisión de datos del Servidor a los Clientes a la vez.

Con todas las configuraciones descritas anteriormente, se inicia con la transmisión de datos, la imagen *MODBUS TCP/IP* también está clasificada horizontalmente para una mejor visualización del operador, los valores a enviar por ser tipo INT pueden ser de (-32768 a +32767), en los indicadores se ha establecido un valor máximo de 1000, el dato enviado en este ejemplo es de 763, valor que es representado en los dos clientes, considerar que la configuración (MB\_MODE=0). (Ver figura 61).

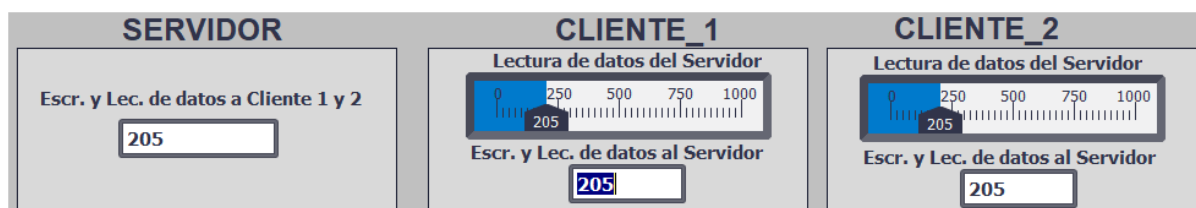


**Figura 61.** Transmisión de datos del Servidor a Clientes 1 y 2.



### 3.6.2. Transmisión de datos de los Clientes al Servidor.

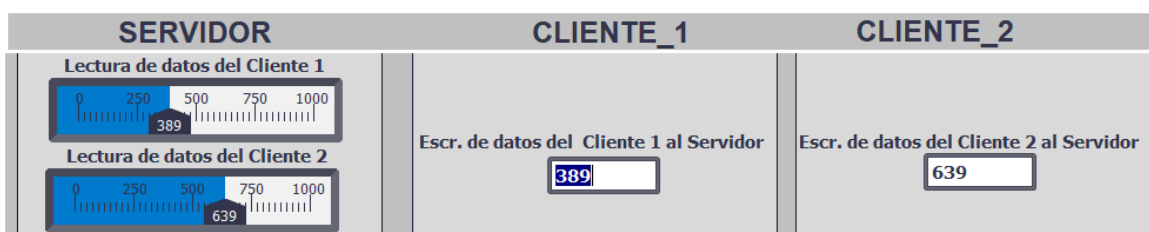
De igual manera se puede leer los datos enviados desde cualquier cliente con la configuración (MB\_MODE=1), en la figura 62 se observa la lectura del dato 205 en el servidor y escrita desde cualquier cliente.



*Figura 62.* Transmisión de datos del Servidor a Clientes 1 y 2.

### 3.6.3. Transmisión de datos de Clientes al Servidor con valores independientes.

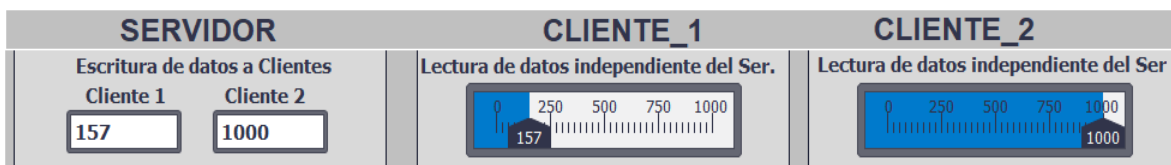
El servidor también puede recibir datos independientemente de cada PLC, en la figura 63 se observa el valor 389 que es enviado desde el CLIENTE\_1 al SERVIDOR y el valor 639 enviado desde el CLIENTE\_2 al SERVIDOR.



*Figura 63.* Transmisión de datos del Servidor a Clientes 1 y 2

### 3.6.4. Transmisión de datos del Servidor a Clientes con valores independientes.

El servidor puede enviar datos independientemente a cada cliente, en este caso el valor 157 es enviado al CLIENTE\_1 y el valor 1000 es enviado AL CLIENTE\_2. (Ver figura 64).



*Figura 64.* Transmisión de datos del Servidor a Clientes 1 y 2

### 3.6.5. Control de lectura o escritura.

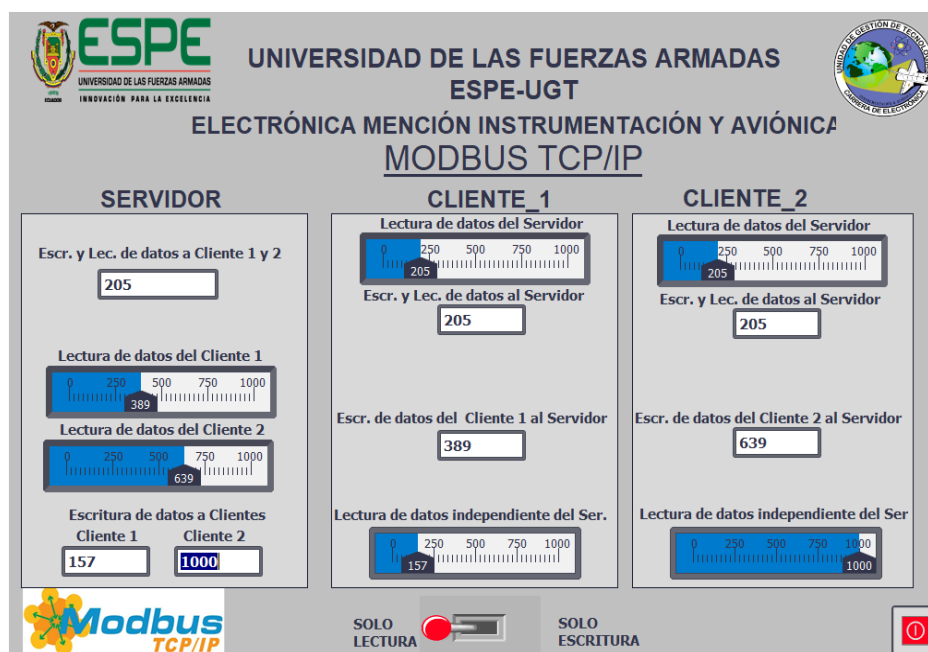
Para mayor facilidad del operador se ha integrado un selector de lectura y escritura que facilitara la operación a realizar en el sistema de comunicación Modbus TCP/IP. (Ver figura 65).



*Figura 65.* Selector de lectura y escritura en la interfaz HMI de Win CC

### 3.6.6. HMI con todos los datos transmitidos en la Red Modbus TCP.

Como parte final de la lectura y escritura de datos se presenta la Interfaz completa con todos los valores transmitidos. (Ver figura 66).



*Figura 66.* Transmisión de datos del servidor a clientes y viceversa

### 3.6.7. Comprobación de la estabilidad de la comunicación Modbus TCP.

Para la comprobación de la comunicación entre el servidor y los dos clientes se puede observar con la función, *Establecer conexión online*, en el apartado de Dispositivos acceder en **Dispositivos y redes**, (Ver figura 67), o en la interfaz HMI de Win CC en la imagen *Establecer comunicación* se observa el STATUS de cada PLC.



Nombre de conexión local	Punto final local	ID local (hex)	ID del interloc...	Interlocutor	Tipo de conexión
S7_Conexión_1	PLC2_Servidor [CPU...	100	1	PLC3_Cliente2 [...	Conexión S7
S7_Conexión_1	PLC3_Cliente2 [CPU... 1		100	PLC2_Servidor [CPU...	Conexión S7
S7_Conexión_2	PLC3_Cliente2 [CPU... 2			Desconocido	Conexión S7
S7_Conexión_3	PLC2_Servidor [CPU... 101			Desconocido	Conexión S7
HMI_Conexión_1	PC-System_1   HMI...			PLC2_Servidor [CPU...	Conexión HMI
HMI_Conexión_2	PC-System_1   HMI...			PLC3_Cliente2 [CPU...	Conexión HMI
HMI_Conexión_3	PC-System_1   HMI...			PLC1_Cliente [CPU ...	Conexión HMI
Conexión PG_192.168.0.9	PLC1_Cliente			192.168.0.9	Conexión PG
Open User Comunicacio...	PLC2_Servidor	1		192.168.0.1	Open User Communi...
Otra conexión_192.168.0...	PLC2_Servidor			192.168.0.11	Otra conexión
Conexión PG_192.168.0.9	PLC2_Servidor			192.168.0.9	Conexión PG
Open User Comunicacio...	PLC1_Cliente	1		192.168.0.2	Open User Communi...

Figura 67. Estado de la red Modbus TCP



Figura 68. Imagen para establecer comunicación con el status de cada PLC

## CAPÍTULO IV

### CONCLUSIONES Y RECOMENDACIONES

#### 4.1. Conclusiones

Modbus trabaja con los protocolos TCP, ASII y RTU, sobresaliendo TCP por la facilidad de conexión con prestaciones de relevancia como el envío de paquetes de datos de 1500 bytes a una velocidad de transmisión de 10 a 100Mbits/s y por su versatilidad física que permite integrar hasta 247 dispositivos, se verificó la calidad y velocidad de respuesta de ésta red al conectar tres dispositivos, uno como servidor y dos como clientes, teniendo una comunicación con envío de datos en paralelo evidenciada en una respuesta inmediata y real.

Modbus TCP maneja el modelo cliente-servidor, razón por la cual en esta red de comunicación industrial la solicitud del servidor será asistida de forma independiente ya sea por el cliente 1 o por el cliente 2 con datos a transferir, en este caso se creó un Array que contiene una estructura de 10 datos, este bloque tiene todos los datos que están presentes en la red, obteniendo así la eficiencia de la velocidad de transmisión por el tamaño de datos enviados, lo cual se evidenció en un enlace de calidad, considerando que los datos no deben ser menores a un área de memoria de 2 Bytes.

El protocolo de comunicación Modbus TCP/IP maneja un estándar en el campo industrial para automatización de procesos, siendo eficiente su implementación con cualquier dispositivo que tenga o soporte sockets TCP/IP, el laboratorio de Automatización y Control de la ESPE-UGT cuenta con módulos compatibles que permite realizar prácticas de redes industriales bajo este protocolo, esto es en beneficio del conocimiento que obtendrán los estudiantes de la carrera.

#### **4.2. Recomendaciones.**

Investigar las propiedades de comunicación Modbus TCP IP de las dos series de los autómatas para conocer los datos y parámetros a configurar; de forma que se pueda activar marcas del ciclo y marcas del sistema de los PLC para el complemento de la programación en el lenguaje ladder, permitiendo establecer una comunicación sólida con los contactos del PLC y señales de reloj.

Configurar la dirección IP del HMI Win CC con la misma IP de la PC que esté conectada a la RED, y a la vez comprobar la conexión PROFINET para la transferencia de datos.

Crear un bloque de datos global para insertar la información a transmitir de tipo Array de 0 a 10 ya que de esta manera se cumple con los parámetros de longitud de dato de las instrucciones Modbus en Tia Portal.

## REFERENCIAS BIBLIOGRÁFICAS

- Aresi, A., & Ardoli, F. (2015). PLANTA INDUSTRIAL A ESCALA ACADÉMICA PARA EL ESTUDIO DE CONCEPTOS Y APLICACIONES EN REDES DE CONTROLADORES (PLCs) Y REDES DE CAMPO INDUSTRIAL. *Tesis de pregrado*. UNIVERSIDAD NACIONAL DE ROSARIO., Rosario. Recuperado el 3 de Octubre de 2019, de <https://www.dsi.fceia.unr.edu.ar/images/downloads/CIM/ProyectoComunicacionesIndustriales.pdf>
- Automation Networks. (2017). *Protocolo Modbus*. Recuperado el 9 de Octubre de 2019, de Automation Networks: <http://automation-networks.es/glossary/modbus-tcpip>
- Bartolomé, J. (2011). *TOALEMON*. Recuperado el 9 de Noviembre de 2019, de El protocolo MODBUS: <http://www.tolaemon.com/docs/modbus.htm>
- Camposano, J., & Rivera, J. (2016). Tablero para el aprendizaje de comunicaciones industriales con equipos Siemens. (*Tesis de pregrado*). Escuela Superior Politécnica del Litoral, Guayaquil. Recuperado el 12 de Diciembre de 2019, de <https://www.dspace.espol.edu.ec/retrieve/97671/D-103365.pdf>
- Cassiolato, C. (2020). *SMAR*. Recuperado el 20 de Enero de 2020, de CIM: <http://www.smar.com/espanol/articulos-tecnicos/redes-industriales-parte-1>
- Castro, C., & Alvarado, J. (2017). Implementación de un red Modbus TCP Wireless con integración de tecnología Siemens y Schneider para prácticas estudiantiles del laboratorio de automatización industrial. (*Tesis de pregrado*). Universidad Politécnica Salesiana, Guayaquil. Recuperado el 11 de Noviembre de 2019, de <https://dspace.ups.edu.ec/bitstream/123456789/14191/1/UPS-GT001882.pdf>

- Cedillo, M., & Concepción, R. (2014). AUTOMATIZACIÓN DE UN MÓDULO DE ENTRENAMIENTO APLICADO A SISTEMAS DE TRASLADO DE MATERIALES. *Tesis de Pregrado*. INSTITUTO POLITÉCNICO NACIONAL, México. Recuperado el 4 de Octubre de 2019, de <https://tesis.ipn.mx/jspui/bitstream/123456789/14995/1/AUTOMATIZACI%C3%93N%20DE%20UN%20M%C3%93DULO%20DE%20ENTRENAMIENTO%20APLICADO%20A%20SISTEMAS%20DE%20TRASLADO%20DE%20MATERIALES.pdf>
- Conacyt. (2017). *Prociencia*. Recuperado el 13 de Noviembre de 2019, de PROYECTO DE INVESTIGACIÓN DISEÑO E IMPLEMENTACIÓN DE SENSORES DE RADIACIÓN SOLAR Y TRANSMISIÓN INALÁMBRICA DE DATOS: [https://www.conacyt.gov.py/sites/default/files/upload\\_editores/u294/Protocolo\\_tecnico\\_cientifico\\_sensores.pdf](https://www.conacyt.gov.py/sites/default/files/upload_editores/u294/Protocolo_tecnico_cientifico_sensores.pdf)
- Defas, R., & Guzmán, A. (2016). Diseño e implementación de un módulo didáctico para la integración de redes de campo industrial: Modbus, Profibus, para actuadores eléctricos. (*Tesis de pregrado*). Escuela Politécnica Nacional, Quito. Recuperado el 12 de Diciembre de 2019, de [http://ciecfie.epn.edu.ec/wss/VirtualDirectories/80/JIEE/historial/XXVII/Contenido/MEMORIAS\\_XXVII-145-151.pdf](http://ciecfie.epn.edu.ec/wss/VirtualDirectories/80/JIEE/historial/XXVII/Contenido/MEMORIAS_XXVII-145-151.pdf)
- Ferreira, M. (Abril de 2016). *Telecomunicaciones*. Recuperado el 13 de Noviembre de 2019, de Sistema de comunicaciones: <http://24tes.blogspot.com/2016/04/establecimiento-de-comunicacion-en-un.html>
- Gutierrez, F. (2015). Sistema de seguridad con visualizadores de programación gráfica para procesos de control. (*Tesis de pregrado*). Universidad Nacional Autónoma de México,

México. Recuperado el 14 de Noviembre de 2019, de <https://docplayer.es/78266354-Tesis-sistema-de-seguridad-con-visualizadores-de-programacion-grafica-para-procesos-de-control.html>

Hernández, M. (2016). Desarrollo e implementación de una red de datos basada en Modbus y Ethernet para autómatas industriales. (*Tesis de pregrado*). Universidad de Sevilla, Sevilla. Recuperado el 20 de Diciembre de 2019, de <https://idus.us.es/xmlui/bitstream/handle/11441/53165/marhertin.pdf?sequence=1&isAllowed=y>

López, R., & Mora, E. (2017). Diseño e implementación de un módulo didáctico para una red de comunicación Industrial Utilizando protocolo abierto Modbus RTU-TCP/IP para monitoreo, control local y remoto de la estación de multivariables físicas, en el laboratorio de hidrónica. (*Tesis de pregrado*). Universidad de las Fuerzas Armadas Extensión Latacunga, Latacunga. Recuperado el 10 de Octubre de 2019, de <http://repositorio.espe.edu.ec/xmlui/handle/21000/13436?show=full>

Mecafenix, F. (2018). *Ingeniería Mecafenix*. Recuperado el 5 de Diciembre de 2019, de Que es y para que sirve un PLC: <https://www.ingmecafenix.com/automatizacion/que-es-un-plc/>

MECATRON. (2016). *Principios de automatización*. Recuperado el 30 de Septiembre de 2019, de Piramide CIM: <http://automatizacion2008.blogspot.com/2008/03/piramide-cim.html>

Murcia, G., & Strack, L. (2016). *Instrumentación avanzada*. Recuperado el 3 de Diciembre de 2019, de Introduccion a las comunicaciones Industriales Scada con LabVIEW: [http://www3.fi.mdp.edu.ar/electrica/instrumentacion/recursos/ia\\_2016\\_clase7.pdf](http://www3.fi.mdp.edu.ar/electrica/instrumentacion/recursos/ia_2016_clase7.pdf)

Sarmiento, D. (2016). Implementación de los protocolos de comunicación industrial mediante Simatic S7-1200 para el laboratorio de automatización de la Universidad del Azuay. (*Tesis*



*de pregrado*). Universidad del Azuay, Cuenca. Recuperado el 13 de Noviembre de 2019, de <http://dspace.uazuay.edu.ec/handle/datos/5479>

Siemens. (2011). *Simatic*. Recuperado el 4 de Enero de 2020, de S7 300-CPU 32xC y CPU 31x:

Datos técnicos:  
[https://cache.industry.siemens.com/dl/files/149/36305149/att\\_1849/v1/s7300\\_cpu\\_31xc\\_and\\_cpu\\_31x\\_manual\\_es-ES\\_es-ES.pdf](https://cache.industry.siemens.com/dl/files/149/36305149/att_1849/v1/s7300_cpu_31xc_and_cpu_31x_manual_es-ES_es-ES.pdf)

Siemens. (Marzo de 2014). *Simatic*. Recuperado el 15 de Octubre de 2019, de S7 Controlador programable S7-1200:

[https://media.automation24.com/manual/es/91696622\\_s71200\\_system\\_manual\\_es-ES\\_es-ES.pdf](https://media.automation24.com/manual/es/91696622_s71200_system_manual_es-ES_es-ES.pdf)

Siemens. (2019). *Siemens*. Recuperado el 18 de Enero de 2020, de CPU 315-2 PN/DP:

<https://mall.industry.siemens.com/mall/es/WW/Catalog/Products/10026479?tree=CatalogTree>

Siemens. (2019). *Siemens*. Recuperado el 15 de Diciembre de 2019, de CPU 1215C:

<https://mall.industry.siemens.com/mall/en/us/Catalog/Product/6ES72151BG400XB0>

Smartech. (2017). *Smartech*. Recuperado el 5 de Diciembre de 2019, de Redes Industriales:

<http://smartech.com.ec/index.php/redes-insdustriales/>

Torres, J., & Vega, A. (2015). Diseño e implementación de un laboratorio de redes de comunicación industrial para la universidad politécnica Salesiana, Cuenca. (*Tesis de pregrado*). Universidad Politécnica Salesiana Sede Cuenca, Cuenca. Recuperado el 6 de Noviembre de 2019, de <https://dspace.ups.edu.ec/bitstream/123456789/7764/1/UPS-CT004621.pdf>

Zambrano, R., & Caballero, C. (2018). Diseño e implementación de una red Modbus/RTU entre dos autómatas programables s7-1200 basado en el estándar RS485. (*Tesis de pregrado*). Universidad Politécnica Salesiana Sede Guayaquil, Guayaquil. Recuperado el 23 de Noviembre de 2019, de <https://dspace.ups.edu.ec/bitstream/123456789/16357/1/UPS-GT002374.pdf>

# ANEXOS



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**  
**CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y**  
**AVIÓNICA**  
**CERTIFICACIÓN**

Se certifica que la presente monografía fue desarrollada por el Señor Cachago Gómez, Jonathan Javier bajo mi supervisión.

En la ciudad de Latacunga, a los 31 días del mes de enero del 2020

**Aprobado por:**

Ing. Paola Sandoval.  
Director del proyecto

Ing. Pablo Pilatasig.  
Director de carrera

Abg. Sarita Plaza.  
Secretaria académica