

REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD DEL ZULIA  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA ELÉCTRICA



**"GUÍA DE PRÁCTICAS DE LABORATORIO DE REDES DE AUTOMATIZACIÓN  
INDUSTRIAL PARA LA INTEGRACION DE PLCS SIMATIC S7-1200 Y  
CONTROLLOGIX"**

Rodríguez Barrios, Amílcar José

Socorro Acosta, Eduardo José

Maracaibo, Enero del 2016

## **PRÁCTICA N° 1**

*Introducción a LabVIEW*

## ÍNDICE GENERAL

	Pág.
1. OBJETIVO GENERAL.....	1
2. LISTA DE MATERIALES Y EQUIPOS .....	1
3. FUNDAMENTACIÓN TEÓRICA.....	1
3.1. ¿Qué es LabVIEW? .....	1
3.2. Programación en LabVIEW.....	2
3.2.1. Tipo de datos .....	3
3.2.2. Estructuras.....	5
4. PROCEDIMIENTO .....	7
4.1. Crear un proyecto.....	7
4.2. Crear un VI y definir sus partes.....	10
4.3. Herramientas.....	12
4.3.1. Paleta de herramientas <i>Controls</i> .....	12
4.3.2. Paleta de herramientas <i>Functions</i> .....	13
4.4. Crear VI de prueba.....	14
4.5. Crear VI haciendo uso de las estructuras <i>Case</i> y <i>While Loop</i> .....	18
4.5.1. <i>Case</i> .....	18
4.5.2. <i>While Loop</i> .....	22
4.6. Implementación de la herramienta <i>Image Navigator</i> para el diseño de HMIs .	29

## ÍNDICE DE FIGURAS

	Pág.
Figura 1. Estructura <i>While Loop</i> .....	5
Figura 2. Estructura <i>While Loop</i> .....	6
Figura 3. Ventana de gestión de proyectos de LabVIEW.....	7
Figura 4. Ventana <i>Create Project</i> .....	8
Figura 5. Ventana <i>Untitled Project 1 - Project Explorer</i> .....	9
Figura 6. Guardar proyecto <i>Untitled Project 1</i> .....	9
Figura 7. Guardar proyecto <i>Untitled Project 1</i> como Proyecto 1.....	10
Figura 8. Crear nuevo VI desde Proyecto 1.....	10
Figura 9. Ventana <i>Front Panel</i> o Panel Frontal.....	11
Figura 10. Ventana <i>Block Diagram</i> o Diagrama de Bloque.....	12
Figura 11. Paleta de herramientas <i>Controls</i> .....	13
Figura 12. Paleta de herramientas <i>Functions</i> .....	13
Figura 13. Control e Indicador booleano dispuestos en el <i>Front Panel</i> .....	14
Figura 14. Control e Indicador booleano modificados en el <i>Front Panel</i> .....	15
Figura 15. Selección de la propiedad mecánica del Control (Por defecto <i>Switch When Pressed</i> ).....	15
Figura 16. Diagrama de bloques del VI.....	16
Figura 17. Modos de ejecución de VI.....	17
Figura 18. VI ejecutado (una sola vez).....	17
Figura 19. Selección y despliegue del control <i>Enum</i> .....	18
Figura 20. Selección del indicador <i>String Indicator</i> .....	19
Figura 21. Estructura <i>Case</i> desplegada en el <i>Block Diagram</i> .....	19
Figura 22. Configuración del <i>Control selector de caso</i> .....	20
Figura 23. Conexión del <i>Control selector de caso</i> con la estructura <i>Case</i> .....	20
Figura 24. Configuración y conexión de la <i>String Constant</i> del caso “Código 1”.....	21
Figura 25. Configuración y conexión de la <i>String Constant</i> del caso “Código 2”.....	21
Figura 26. Prueba del código para el caso “Código 1”.....	22

Figura 27. Prueba del código para el caso “Código 2” .....	22
Figura 28. Estructura <i>While Loop</i> desplegada. ....	23
Figura 29. Creación del control condicional de la estructura <i>While Loop</i> .....	23
Figura 30. Control dispuesto dentro de la estructura. ....	24
Figura 31. <i>Front Panel</i> del VI con estructura <i>While Loop</i> .....	24
Figura 32. <i>Block Diagram</i> del VI con estructura <i>While Loop</i> . ....	25
Figura 33. Creación de control de la herramienta <i>Wait Until Next ms Multiple</i> .....	26
Figura 34. Herramienta <i>Wait Until Next ms Multiple</i> y <i>Control de velocidad del bucle (ms)</i> dispuesto en el <i>Block Diagram</i> .....	26
Figura 35. <i>Control de velocidad del bucle (ms)</i> dispuesto en el <i>Front Panel</i> . ....	27
Figura 36. Creación de Indicador numérico asociado a terminal <i>Loop iteration</i> .....	27
Figura 37. Indicador <i>Conteo de bucle</i> dispuesto en el <i>Block Diagram</i> . ....	28
Figura 38. Indicador <i>Conteo de bucle</i> dispuesto en el <i>Front Panel</i> .....	28
Figura 39. VI finalizado con el indicador <i>Conteo de bucle</i> funcionando. ....	29
Figura 40. Creación de una nueva ventana <i>Control</i> .....	30
Figura 41. Ventana <i>Control 1</i> .....	30
Figura 42. Ventana <i>Image Navigator</i> .....	31
Figura 43. Control Booleano <i>Horizontal Toogle Switch</i> y símbolos <i>Selector switch 3</i> desplegados en <i>Control 1</i> .....	32
Figura 44. <i>Change to Customize Mode</i> (personalización) y <i>Change to Edit Mode</i> ...	33
Figura 45. Símbolo <i>Selector switch 3 (rigth)</i> asociado en control <i>Boolean</i> en estado <i>false</i> . ....	33
Figura 46. Control <i>Boolean</i> personalizado desplegado en <i>Front Panel</i> del nuevo VI.	34
Figura 47. Indicador Booleano <i>Round LED</i> y símbolo <i>Motor 1</i> desplegados en <i>Control 1</i> .....	35
Figura 48. Símbolo <i>Motor 1</i> con color verde sólido desplegado en <i>Control 1</i> .....	35
Figura 49. <i>Block Diagram</i> del VI encendido y apagado de un motor. ....	36
Figura 50. <i>Front Panel</i> del VI encendido y apagado de un motor. ....	36
Figura 51. <i>Front Panel</i> del VI encendido y apagado de un motor en ejecución <i>Run Continuously</i> . ....	36

## ÍNDICE DE TABLAS

	Pág.
Tabla 1. Tipos de datos numéricos.....	4

## 1. OBJETIVO GENERAL

Instruir al estudiante en el manejo básico del software LabVIEW para ser implementado posteriormente en el diseño de Interfaces Humano Maquina.

## 2. LISTA DE MATERIALES Y EQUIPOS

- Ordenador con el software LabVIEW instalado.

## 3. FUNDAMENTACIÓN TEÓRICA

### 3.1. ¿Qué es LabVIEW?

LabVIEW constituye un revolucionario sistema de programación gráfica para aplicaciones que involucren adquisición, control, análisis y presentación de datos.

LabVIEW es un entorno de programación destinado al desarrollo de aplicaciones, similar a los sistemas de desarrollo comerciales que utilizan el lenguaje C o BASIC. Sin embargo, LabVIEW se diferencia de dichos programas en un importante aspecto: los citados lenguajes de programación se basan en líneas de texto para crear el código fuente del programa, mientras que LabVIEW emplea la programación gráfica o lenguaje G para crear programas basados en diagramas de bloques.

Para el empleo de LabVIEW no se requiere gran experiencia en programación, ya que se emplean iconos, términos e ideas familiares a científicos e ingenieros, y se apoya sobre símbolos gráficos en lugar de lenguaje escrito para construir las aplicaciones. Por ello resulta mucho más intuitivo que el resto de lenguajes de programación convencionales. LabVIEW posee extensas librerías de funciones y

subrutinas. Además de las funciones básicas de todo lenguaje de programación. Las ventajas que proporciona el empleo de LabVIEW se resumen en las siguientes:

- Se reduce el tiempo de desarrollo de las aplicaciones al menos de 4 a 10 veces, ya que es muy intuitivo y fácil de aprender.
- Dota de gran flexibilidad al sistema, permitiendo cambios y actualizaciones tanto del hardware como del software.
- Da la posibilidad a los usuarios de crear soluciones completas y complejas.
- Con un único sistema de desarrollo se integran las funciones de adquisición, análisis y presentación de datos.
- El sistema está dotado de un compilador gráfico para lograr la máxima velocidad de ejecución posible.
- Tiene la posibilidad de incorporar aplicaciones escritas en otros lenguajes.

### 3.2. Programación en LabVIEW

Los programas desarrollados con LabVIEW se llaman Instrumentos Virtuales, o VIs (por sus siglas en inglés *Virtual Instruments*), llamados así porque su apariencia y comportamiento simula a instrumentos físicos, como: osciloscopios, multímetros etc. Los VIs utilizan funciones que manipulan las entradas del usuario o datos provenientes de otras fuentes y muestran la información procesada o la pasan a ficheros u otros ordenadores. Hoy en día los VIs se han expandido ampliamente no sólo al control de

todo tipo de electrónica (Instrumentación electrónica) sino también a su programación embebida, comunicaciones, matemáticas, etc.

### 3.2.1. Tipo de datos

LabVIEW ofrece una gran variedad de tipos de datos con los que podemos trabajar respondiendo a las necesidades reales con las que nos encontramos. Uno de los aspectos más significativos de LabVIEW es la diferenciación que efectúa en el diagrama de bloques entre los diferentes tipos de controles o indicadores, basada en que cada uno de ellos tiene un color propio. De esta manera nos será fácil identificarlos y reconocer inmediatamente si estamos trabajando con el tipo de dato adecuado.

Los tipos de datos para la implementación en este curso son:

- **Boolean (Booleano):** es el tipo de dato que representa dos estados lógicos, *true* (verdadero) o *false* (falso). Los tipos de datos booleanos son enteros de 16 bits. El bit más significativo contiene el valor booleano. Si el bit 15 se pone a 1, entonces el valor del control o indicador es *true*; por el contrario, si este bit 15 vale 0, el valor de la variable booleana será *false*. Se representan con el color verde claro.
- **Numeric (Numérico):** es el tipo de dato principal de LabVIEW, representa cualquier tipo de número según su tipo (real, imaginario, complejo, con signo, entero), para más detalle véase la **Tabla 1**.

**Tabla 1.** Tipos de datos numéricicos.

	Tipo de dato	Bits de almacenamiento	Dígitos decimales (Aprox.)	Límites	Color
	Precisión extendida Coma flotante	128	33/15 (20)	6,48e-4966 a 1,19e+4932 -6,48e-4966 a -1,19e+4932	Naranja
	Precisión doble Coma flotante	64	15	4,94e-324 a 1,79e+308 -4,94e-324 a -1,79e+308	Naranja
	Precisión simple coma flotante	32	6	1,40e-45 a 3,40e+38 -1,40e-45 a -3,40e+38	Naranja
	Largo Entero con signo	32	9	-2.147.483.648 a 2.147.483.647	Azul
	Palabra Entero con signo	16	4	-32.768 a 32.767	Azul
	Byte Entero con signo	8	2	-128 a 127	Azul
	Largo Entero sin signo	32	9	0 a 4.294.967.295	Azul
	Palabra Entero sin signo	16	4	0 a 65.535	Azul
	Byte Entero sin signo	8	2	0 a 255	Azul
	Precisión Extendida Complejo	256	33/15	Igual que EXT para cada parte (real e imaginario)	Naranja
	Precisión doble Complejo	128	15	Igual que DBL para cada parte (real e imaginario)	Naranja
	Precisión simple Complejo	64	6	Igual que SGL para cada parte (real e imaginario)	Naranja

- **String (Cadena de caracteres):** este tipo de dato se identifica con el color rosa y representa letras, palabras y oraciones, las cuales serán representadas por signos de acuerdo al idioma, cada letra o signo tiene su equivalente ASCII con el cual es manejado por el programa.

### 3.2.2. Estructuras

Una estructura es un elemento de control del programa. Las estructuras controlan el flujo de datos en un VI. G tiene varias estructuras: Bucles *While*, Bucle *For*, Estructura *Case*, Bloque secuencia, Nodo Formula, etc.

- **While Loop (Mientras):** el ciclo *While* ejecuta un subdiagrama hasta que una condición es cumplida, es decir, mientras se cumpla la condición. Dicha condición se encuentra en la terminal *Loop condition*, el cual es un valor booleano (verdadero o falso). El comportamiento predeterminado para la terminal es *Stop if True* (Parar si es verdadero), donde el sub diagrama se ejecuta hasta que la terminal recibe un valor verdadero. Cuando una terminal de condición es configurada como *Continue if True*, el ciclo *While* ejecuta el sub diagrama hasta que la terminal reciba un valor falso. Debido a que la estructura revisa el valor presente en la terminal después de ejecutar cada iteración, el ciclo *While* se ejecuta por lo menos una vez, lo cual es igual que un *Do-While* en los lenguajes de texto convencionales, véase la **Figura 1**.

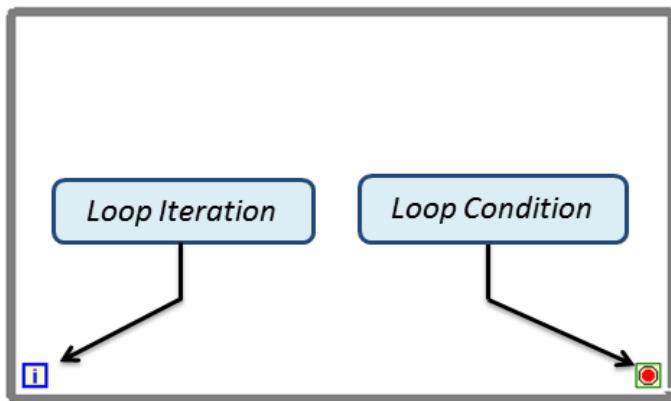
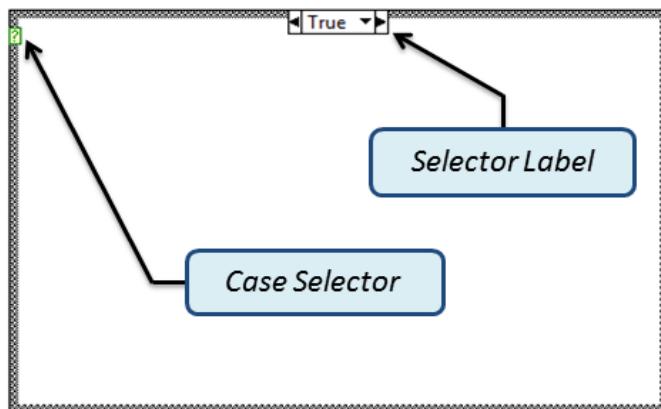


Figura 1. Estructura *While Loop*.

La terminal de iteraciones (*Loop iteration*) proporciona el número de iteraciones completadas; siempre empieza en cero y regresa a cada inicio de la iteración el valor de cero.

Cabe señalar que si la terminal condicionante (*Loop condition*) no se conecta a un valor, el VI no operara.

- **Case (Caso):** una estructura Case tiene uno o más sub diagramas, o casos. Solamente un sub diagrama esta visible y la estructura ejecuta solo un caso a la vez. Un dato de entrada determina cual subdiagrama se ejecutara. Esta estructura es similar al *Switch* o *If-Then*, *If-Else If*, *If-Else*, de los lenguajes de programación tradicionales, ver **Figura 2**.



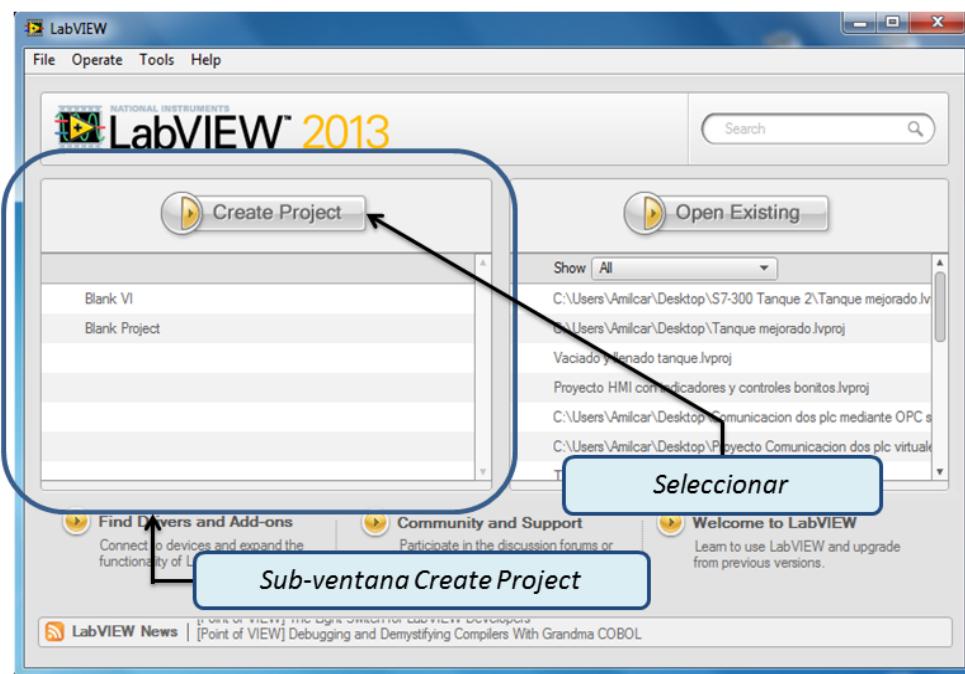
**Figura 2.** Estructura *While Loop*.

En la parte superior de la estructura se encuentra la etiqueta del selector de casos, esta etiqueta contiene los valores para los cuales cambiará cada sub diagrama, aplicando así cada caso. Las flechas a los costados permite desplazarse por cada condición, la flecha hacia abajo indica que casos están presentes en la estructura. La estructura Case puede llegar a tener diferentes tipos de datos de entrada, pudiendo ser booleanos, enteros, cadena de caracteres o del tipo enumerado.

## 4. PROCEDIMIENTO

### 4.1. Crear un proyecto

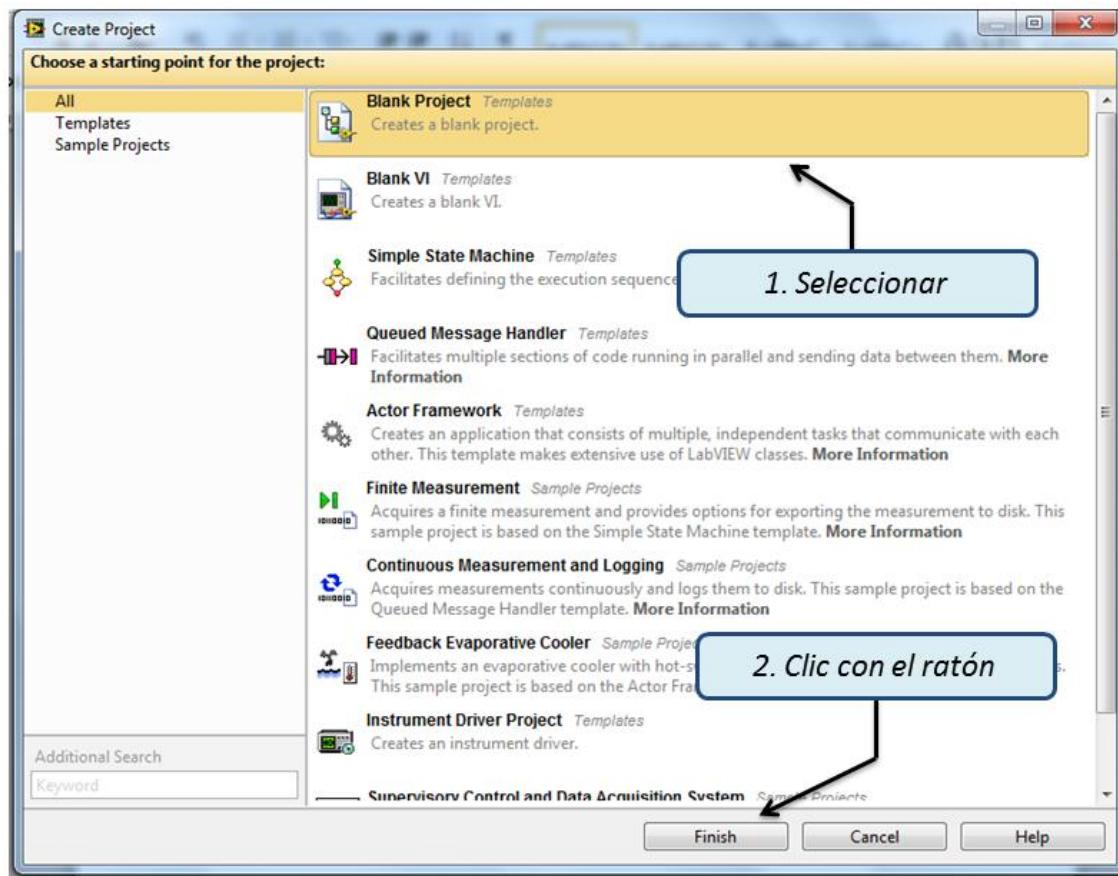
Ejecute LabVIEW, una vez cargado aparecerá una ventana para la versión 2013 como se muestra en la **Figura 3**.



**Figura 3.** Ventana de gestión de proyectos de LabVIEW.

En la ventana abierta (**Figura 3**) se encuentra la opción de *Create Project* (Crear Proyecto) que se utilizará para configurar un nuevo proyecto y *Open Existing* para cargar un proyecto creado anteriormente. Enfoque su atención hacia la sub ventana *Create Project* y obsérvese que aparecen debajo las opciones *Blank VI* y *Blank Project*, dichas son atajos para la creación de nuevos VIs y proyectos en blanco respectivamente.

Pulse la opción *Create Project* y emergerá una nueva ventana como es mostrado en la **Figura 4**.

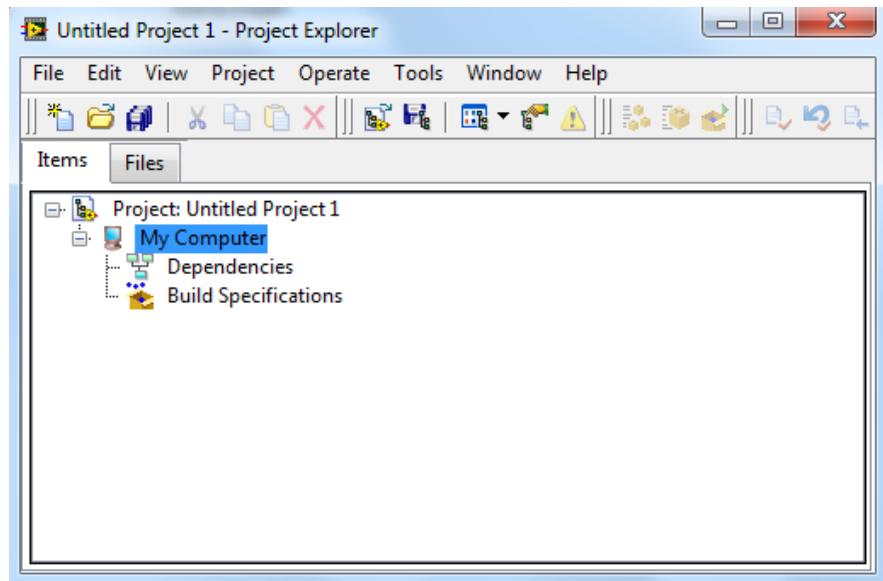


**Figura 4.** Ventana *Create Project*.

En la nueva ventana abierta (*Create Project* [**Figura 4**]) se tienen varias opciones para la creación de distintos tipos de trabajos, para fines de este manual se trabajará con la creación de *Blank Projects* (Proyectos en blanco) y VIs. Seleccione la opción *Blank Project* y volverá a cargarse una nueva ventana como es mostrada en la **Figura 5**.

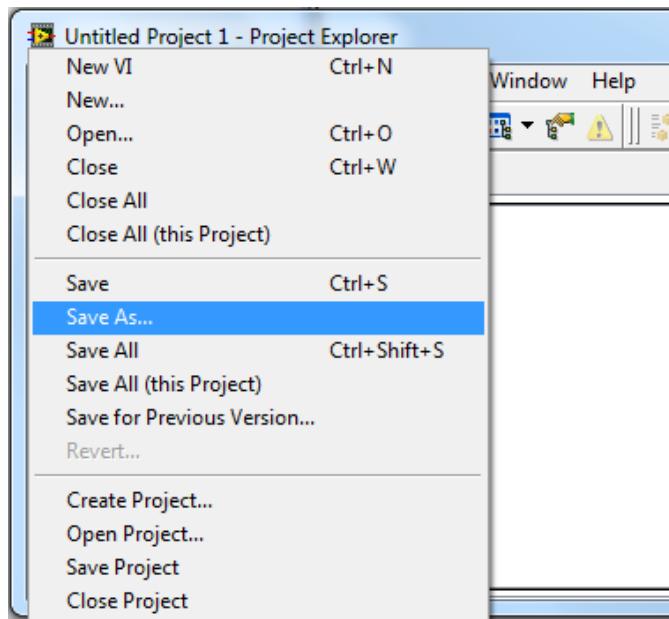
La nueva ventana abierta en la **Figura 5** es lo que se conoce como *Project Explorer* o explorador de proyecto, este entorno será de suma importancia para el desarrollo de

futuros proyectos en este software de programación. En el *Project Explorer* podrá administrar la creación de VIs, variables, *I/O servers*, entre otras funciones.

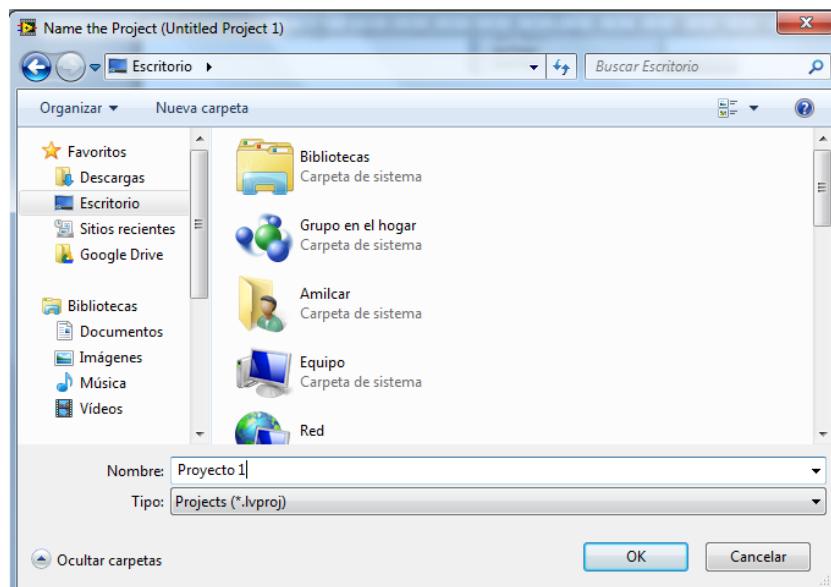


**Figura 5.** Ventana *Untitled Project 1 - Project Explorer*.

Guarde el proyecto haciendo clic en *File / Save as*, véase las **Figuras 6 y 7**.



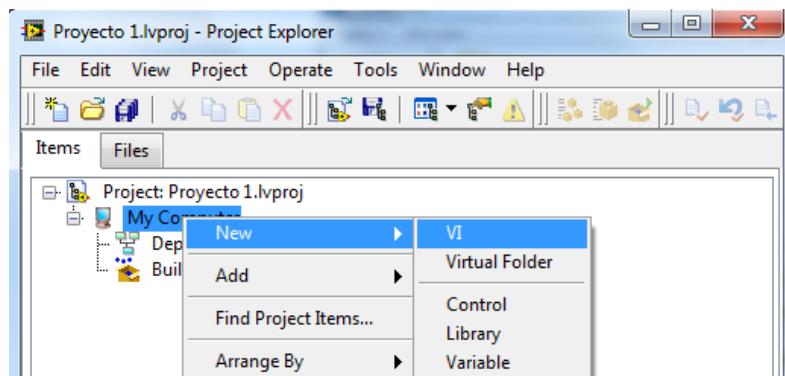
**Figura 6.** Guardar proyecto *Untitled Project 1*.



**Figura 7.** Guardar proyecto *Untitled Project 1* como Proyecto 1.

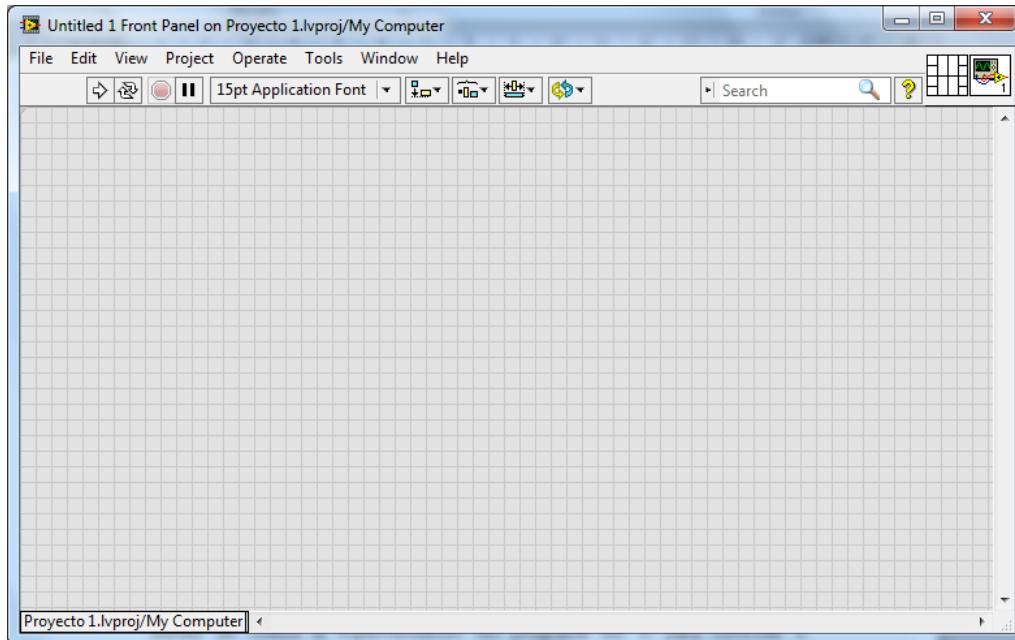
#### 4.2. Crear un VI y definir sus partes.

Partiendo de la apertura previa de un *Blank Project* se creará un nuevo VI, seleccione *My Computer* y haga clic derecho en *New / VI*, vea la **Figura 8**. Una vez abierto el nuevo VI aparecerán automáticamente dos entornos como los que se muestran en la **Figura 9** y **10**.



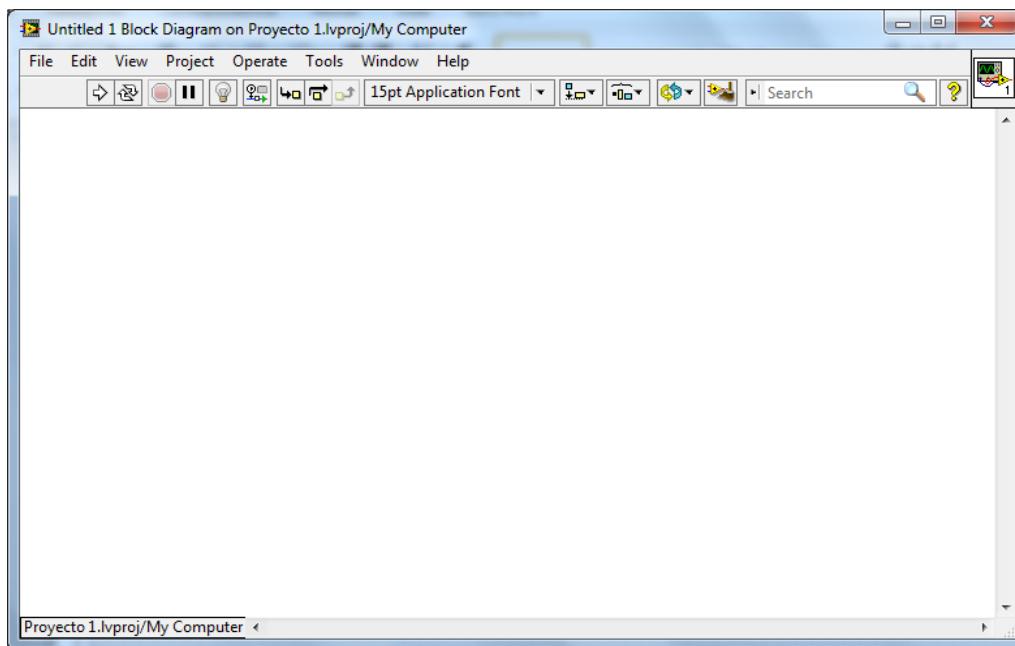
**Figura 8.** Crear nuevo VI desde Proyecto 1.

La ventana de la **Figura 9** es llamada *Front Panel* o Panel Frontal, el cual representa la interfaz gráfica con el usuario. Un panel frontal estará conformado por una serie de botones, pulsadores, potenciómetros, gráficos, etc., cada uno de ellos puede estar definido como un control o indicador. Los controles se emplean para introducir parámetros al VI, mientras que los indicadores se emplean para mostrar los resultados producidos, ya sean datos adquiridos o resultados de alguna operación.



**Figura 9.** Ventana *Front Panel* o Panel Frontal.

La ventana de la **Figura 10** es denominada *Block Diagram* o Diagrama de Bloque, constituye el código fuente del VI. En el diagrama de bloques es donde se realiza la implementación del programa del VI para controlar o realizar cualquier procesado de las entradas y salidas que se crearon en el panel frontal.

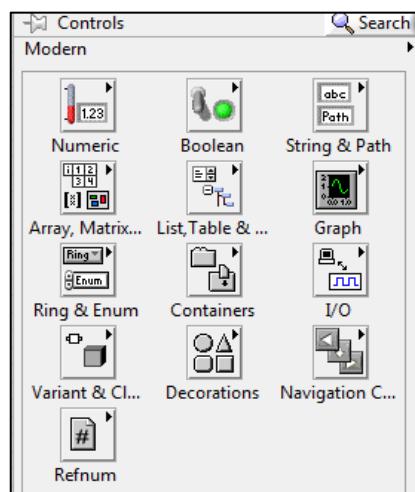


**Figura 10.** Ventana *Block Diagram* o Diagrama de Bloque.

### 4.3. Herramientas

#### 4.3.1. Paleta de herramientas *Controls*

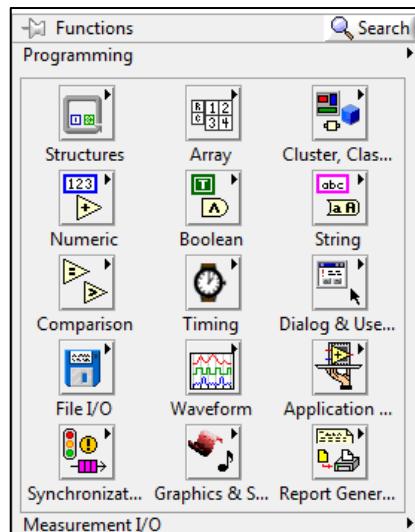
Se utiliza únicamente en el *Front Panel*. Contiene todos los controles e indicadores que se emplearán para crear la interfaz del VI con el usuario. Para acceder a esta paleta haga clic derecho con el ratón sobre el área de trabajo del *Front Panel*, vea la paleta en la **Figura 11**.



**Figura 11.** Paleta de herramientas *Controls*.

#### 4.3.2. Paleta de herramientas *Functions*

Se emplea en el diseño de diagrama de bloques. Contiene todos los objetos que se requieren para la implementación del programa del VI, ya sean funciones aritméticas, de entrada/salida de señales, entrada/salida de datos a fichero, adquisición de señales, temporización de la ejecución del programa, etc. Para acceder a esta paleta haga clic derecho con el ratón sobre el área de trabajo del *Block Diagram*, véase la **Figura 12**.

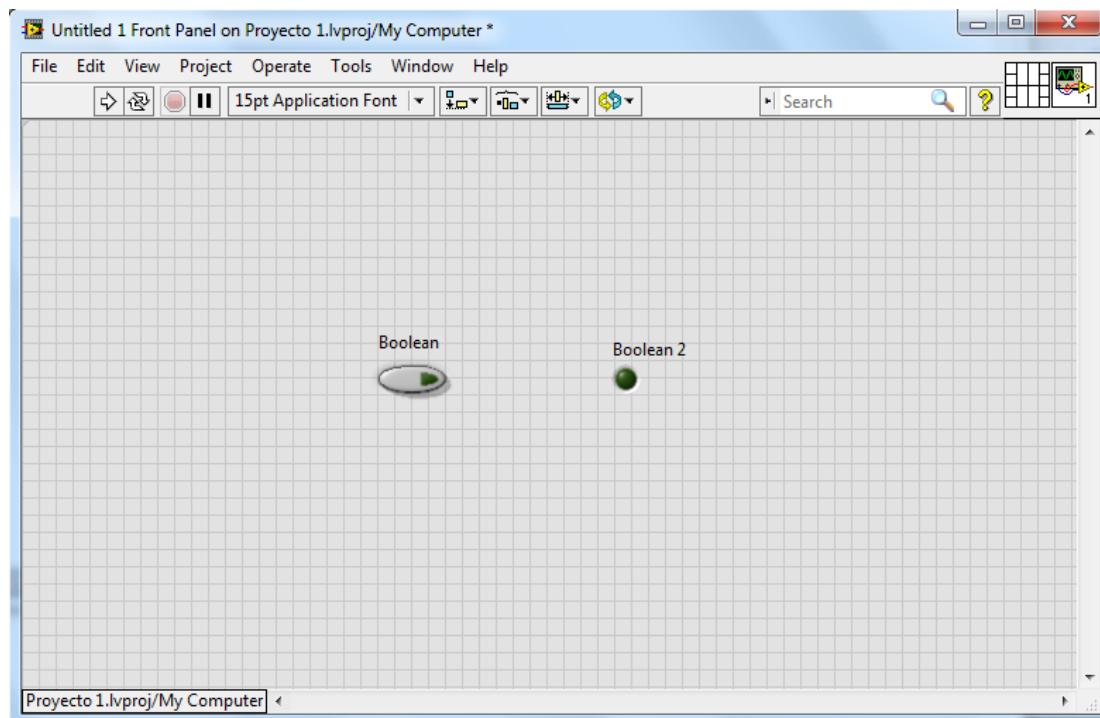


**Figura 12.** Paleta de herramientas *Functions*.

#### 4.4. Crear VI de prueba

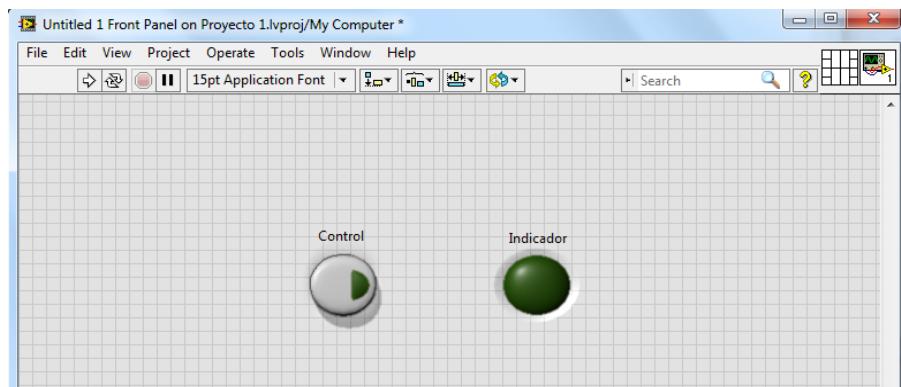
Desde el *Front Panel* seleccione la categoría *Boolean* presente en la paleta de herramientas *Controls*, dentro aparecerán los controles e indicadores correspondientes al funcionamiento de datos exclusivamente Booleanos.

Seleccione un *Push Button* (control) y arrástrelo hacia el área de trabajo del *Front Panel*, posteriormente repita la acción con un *Round LED* (indicador). Véase la **Figura 13**.



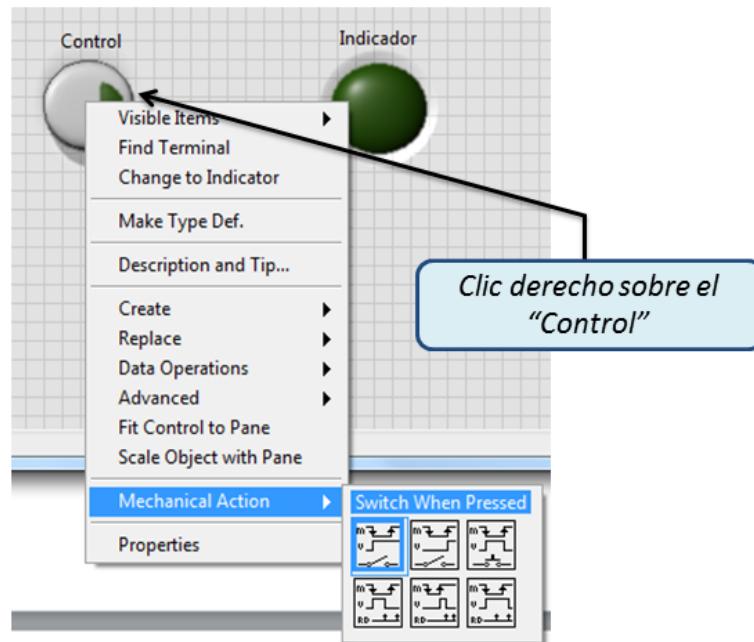
**Figura 13.** Control e Indicador booleano dispuestos en el *Front Panel*.

Es posible realizar modificaciones (tamaño, colores, etc.) de cada control e indicador accediendo a las *Properties* (Propiedades) de cada uno (hacer clic derecho sobre el elemento). Se modificarán nombre (etiqueta de identificación) y tamaño de los elementos como es mostrado en la **Figura 14**.



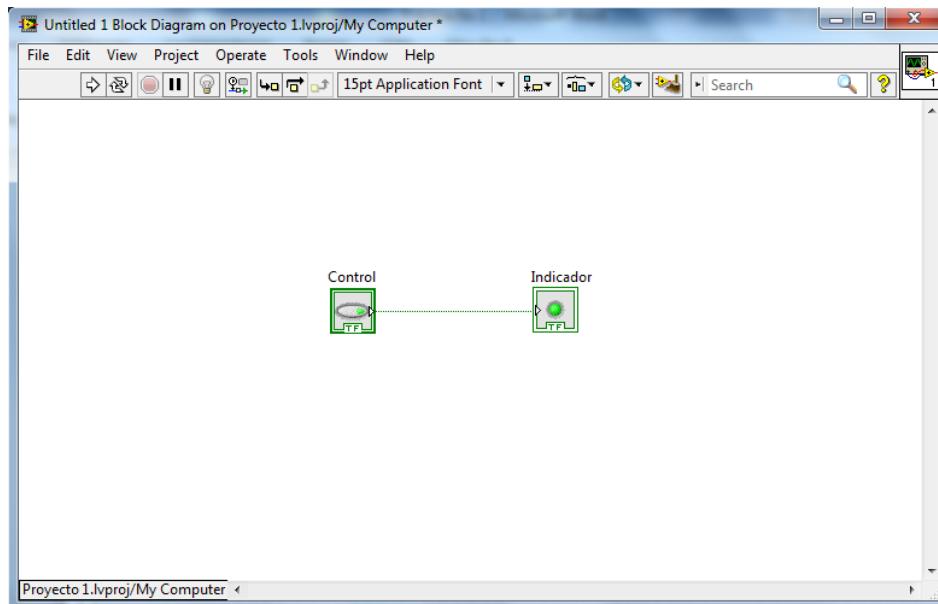
**Figura 14.** Control e Indicador booleano modificados en el *Front Panel*.

También se pueden modificar las propiedades mecánicas de cada control, es decir, su funcionamiento ya sea como pulsador o interruptor, para ello seleccione y haga clic derecho sobre el *Control* desde el *Front Panel* y elija la opción *Mechanical Action*, en ella podrá elegir la modalidad de funcionamiento del control, en este caso se trabajará en modo *Switch When Pressed*, es decir, cambiar estado cuando es pulsado (interruptor), véase la **Figura 15**.



**Figura 15.** Selección de la propiedad mecánica del Control (Por defecto *Switch When Pressed*).

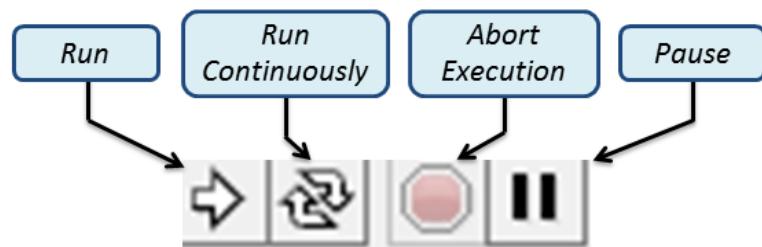
Resta llevar a cabo la programación del VI, por consiguiente, vaya al *Block Diagram* y aparecerán los respectivos bloques de cada elemento presentes en el *Front Panel*, en este caso el control y el indicador, haga clic con el ratón sobre la pequeña flecha del control y conecte ambos bloques como aparece en la **Figura 16**.



**Figura 16.** Diagrama de bloques del VI.

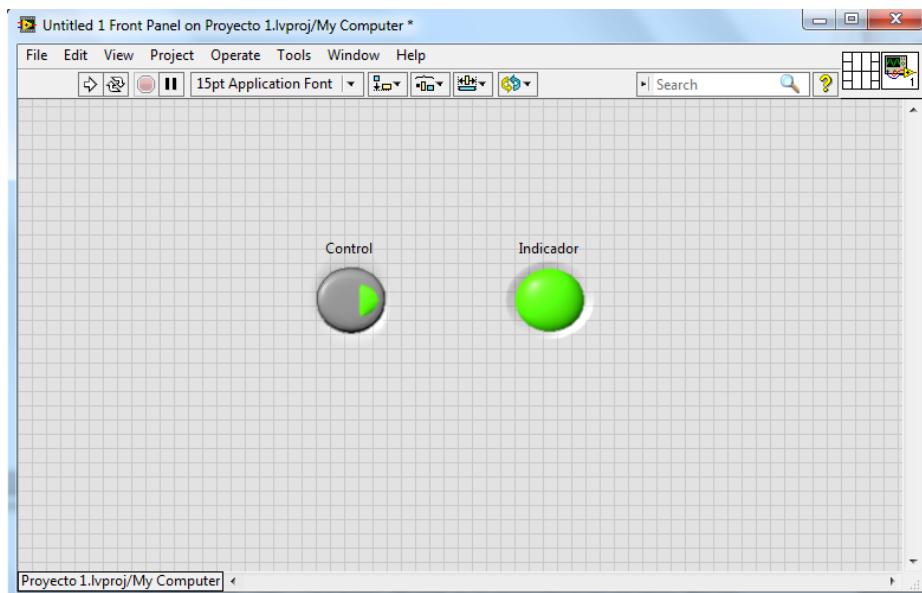
Para manipular la ejecución de los VIs se tienen los modos: *Run*, *Run Continuously*, *Abort Execution* y *Pause* (véase **Figura 17**), estos modos son accesibles desde el *Block Diagram* como del *Front Panel*, sus funciones son las siguientes:

- **Run:** ejecuta el código del VI una sola vez.
- **Run Continuously:** ejecuta el código del VI “N” veces hasta ser detenido (ciclo).
- **Abort Execution:** detiene permanentemente la ejecución del código del VI.
- **Pause:** detiene temporalmente la ejecución del código del VI hasta que el usuario decida reanudar la ejecución.



**Figura 17.** Modos de ejecución de VI.

Conocido el funcionamiento de los modos de operación ejecute el VI realizado, para esto desde el *Front Panel* pulse el botón Control (*true*), luego haga clic con el ratón sobre el botón *Run* y note como se enciende el Indicador, ver la **Figura 18**.



**Figura 18.** VI ejecutado (una sola vez).

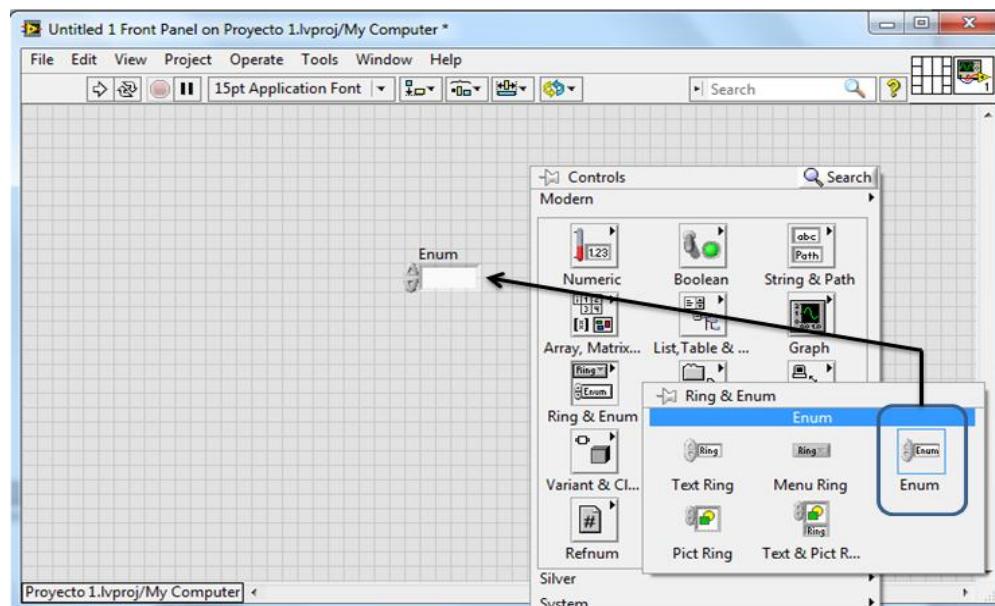
Obsérvese que el VI fue ejecutado solo una vez ya que se usó el modo *Run*, para apagar el Indicador pulse el Control (*false*) y vuelva a seleccionar *Run*. Para ejecutar el VI continuamente, es decir, que los cambios que se realicen en los estados del Control (*true o false*) puedan verse en tiempo real en el Indicador use el modo *Run Continuously*.

## 4.5. Crear VI haciendo uso de las estructuras Case y While Loop

### 4.5.1. Case

Se realizará un ejercicio básico haciendo uso de otros tipos de datos para demostrar la practicidad e importancia de la estructura Case. Elimine del *Front Panel* los elementos presentes del ejemplo anterior *Crear Primer VI*.

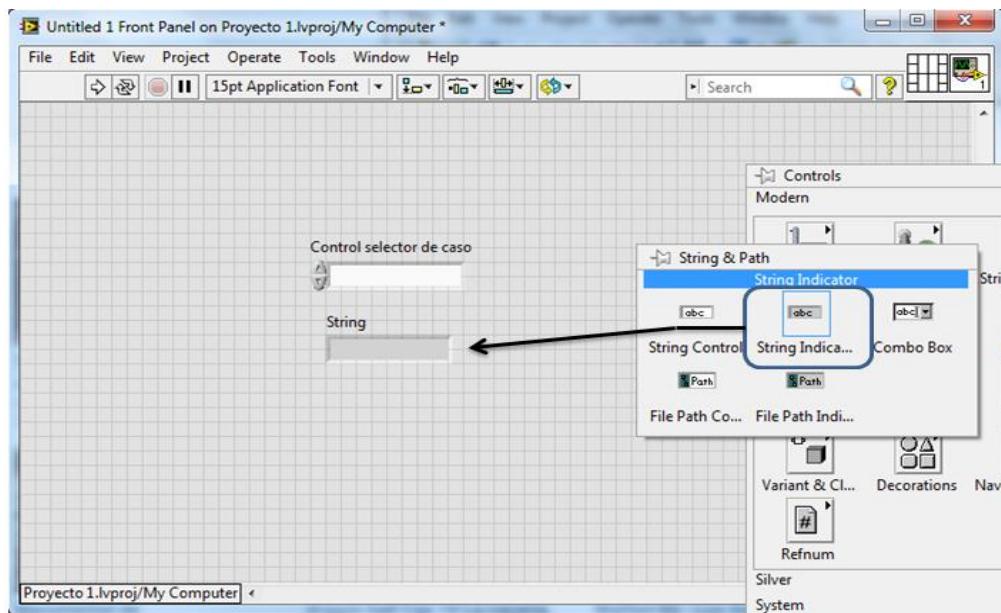
Diríjase al *Front panel / Controls / Ring & Enum* y seleccione el control *Enum*, despliéguelo sobre el área de trabajo del *Front Panel*, véase la **Figura 19**.



**Figura 19.** Selección y despliegue del control *Enum*.

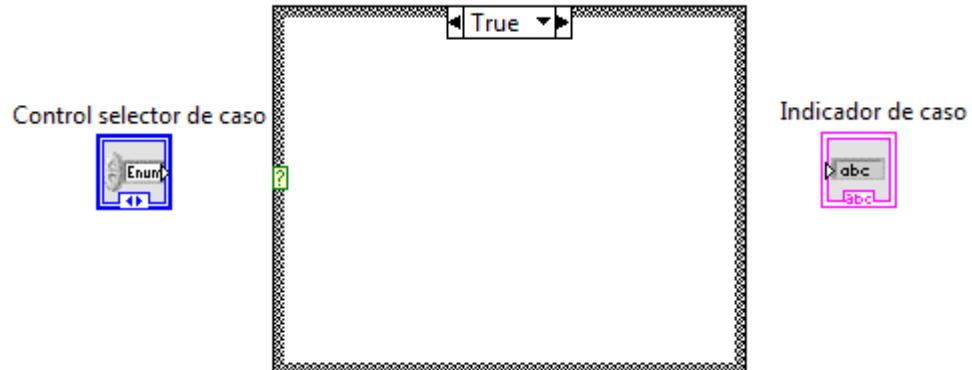
*Enum* es un control selector de caso, proporciona al usuario la posibilidad de poder seleccionar o elegir un caso basado en un nombre constante en lugar de utilizar un dato tipo *string* (cadena de caracteres) o *numeric* (numérico).

Seleccione ahora el indicador *String Indicator* (presente en *Controls / String & Path*) y arrástrelo hasta el área de trabajo del *Front Panel*, ver la **Figura 20**.



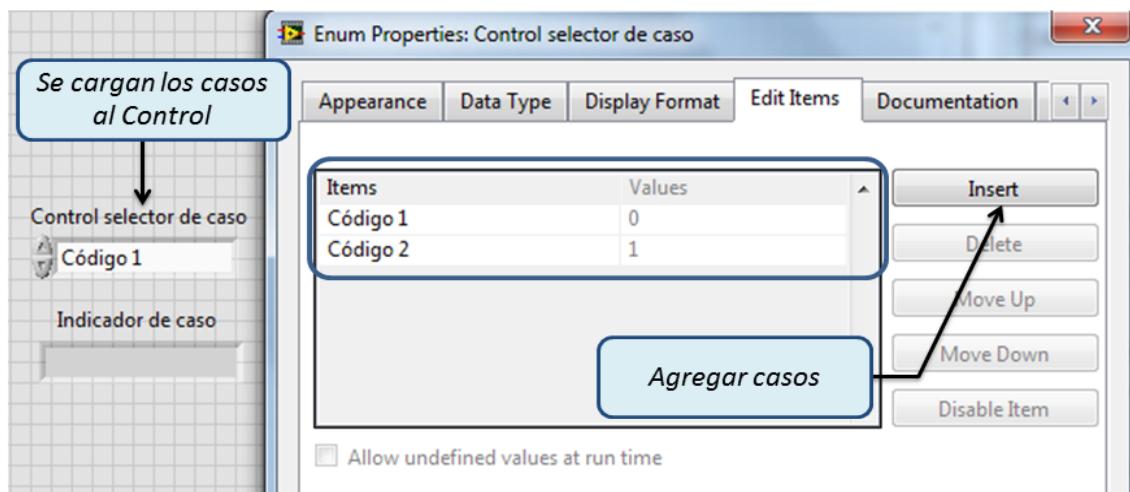
**Figura 20.** Selección del indicador *String Indicator*.

Desde el *Block Diagram* seleccione la estructura *Case structure* (disponible en la paleta de herramientas *Functions / Structures*), despliéguela en el área de trabajo del *Block Diagram* tal como se muestra en la **Figura 21**.



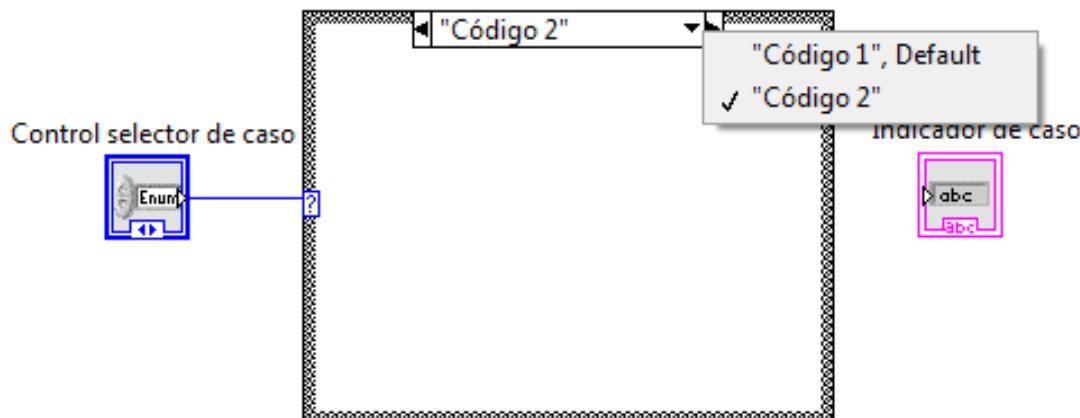
**Figura 21.** Estructura *Case* desplegada en el *Block Diagram*.

Configure el *Control selector de caso*, para ello haga clic derecho sobre el control y elija *Properties / Edit Items*, configure los casos tal como es mostrado en la **Figura 22**. En la pestaña *Edit Items* es posible configurar los casos que se seleccionarán.



**Figura 22.** Configuración del *Control selector de caso*.

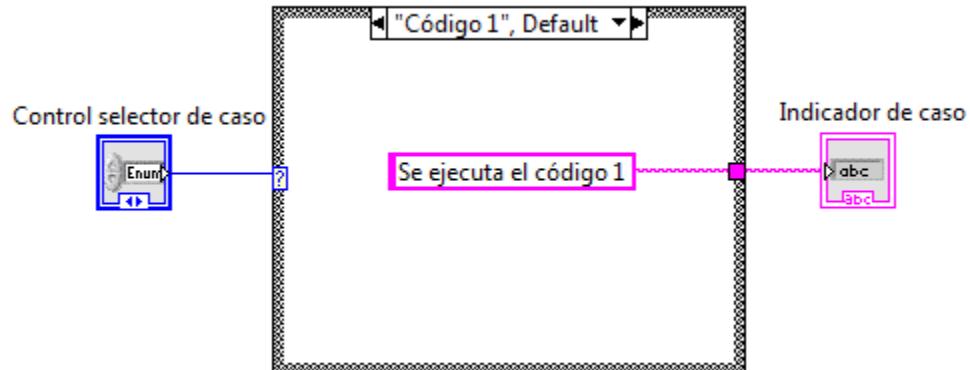
Conecte el *Control selector de caso* con el selector de casos de la estructura *Case*, vea en la **Figura 23** como se actualizan los parámetros de las etiquetas de selección en la parte superior de la estructura.



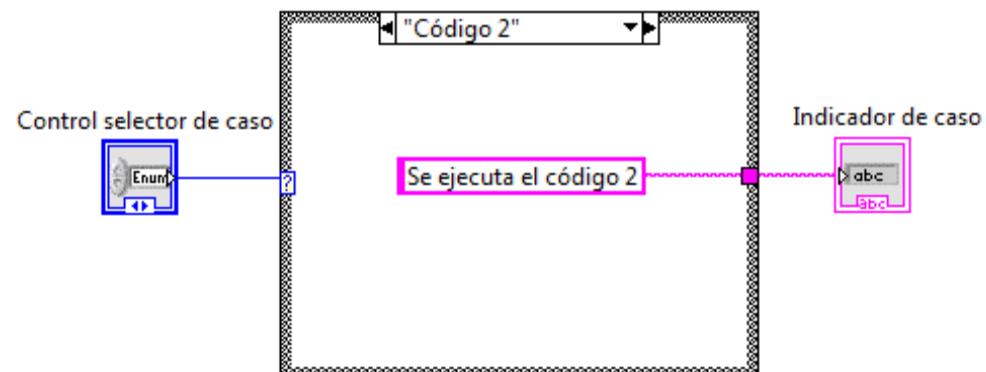
**Figura 23.** Conexión del *Control selector de caso* con la estructura *Case*.

Insera dos *String Constant* dentro de la estructura *Case*, una para la condición “Código 1” y otra para la condición “Código 2”, esta constante se encuentra ubicada en la paleta de herramientas *Functions / String*.

Una vez colocadas las *String Constant* dentro de los casos, ajústelas y conéctelas como es mostrado en la **Figura 24 y 25**.

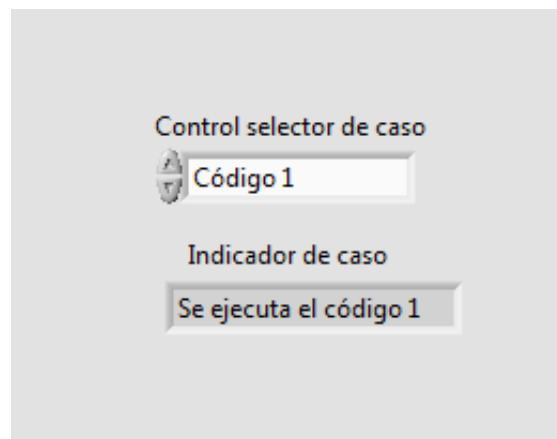


**Figura 24.** Configuración y conexión de la *String Constant* del caso “Código 1”.

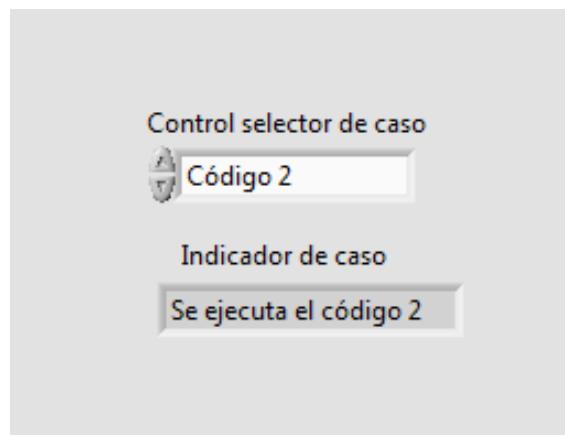


**Figura 25.** Configuración y conexión de la *String Constant* del caso “Código 2”.

Finalizado el VI, ejecútelo en el modo *Run Continuously*, nótese como para el cambio de condición en el *Control selector de caso*, la salida (*Indicador de caso*) cambia de mensaje, ver **Figuras 26 y 27**.



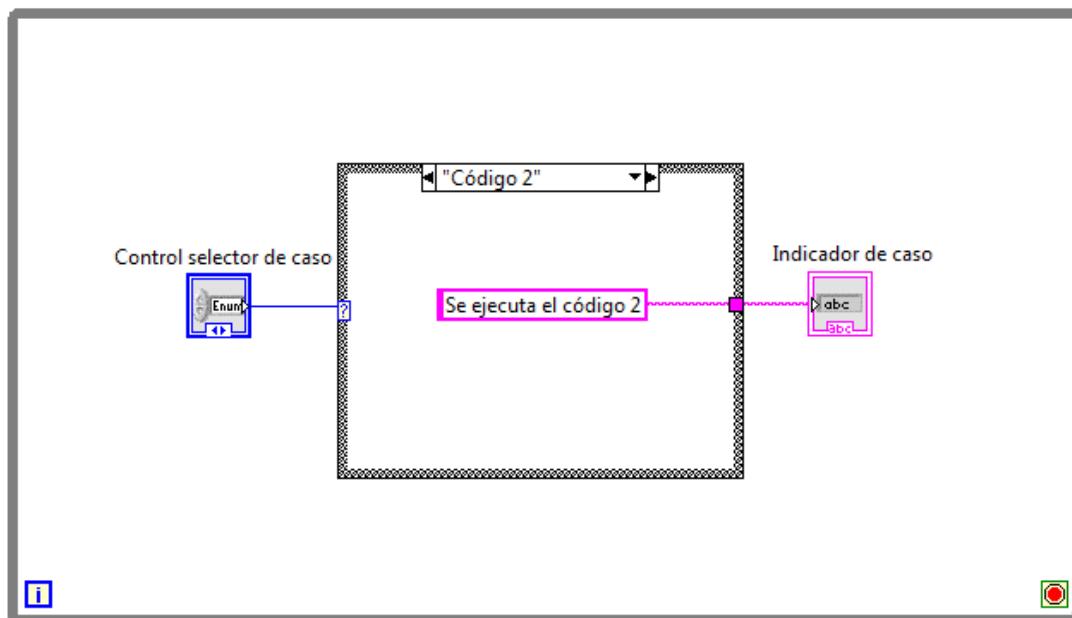
**Figura 26.** Prueba del código para el caso “Código 1”.



**Figura 27.** Prueba del código para el caso “Código 2”.

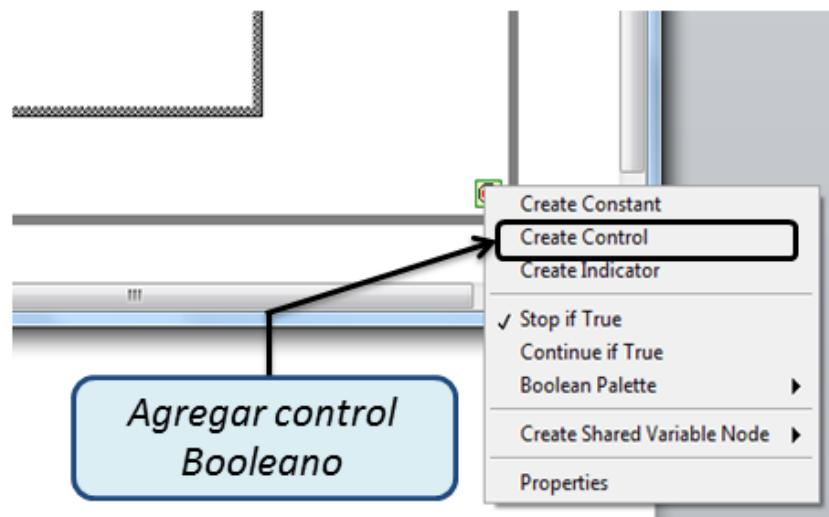
#### 4.5.2. *While Loop*

Para exemplificar el funcionamiento de ésta estructura tan práctica se seguirá trabajando con el ejemplo anterior. Diríjase a la paleta de herramientas *Functions / Structures* (desde el *Block Diagram*) y despliegue sobre el código ya realizado la estructura *While Loop* tal como se muestra en la **Figura 28**.

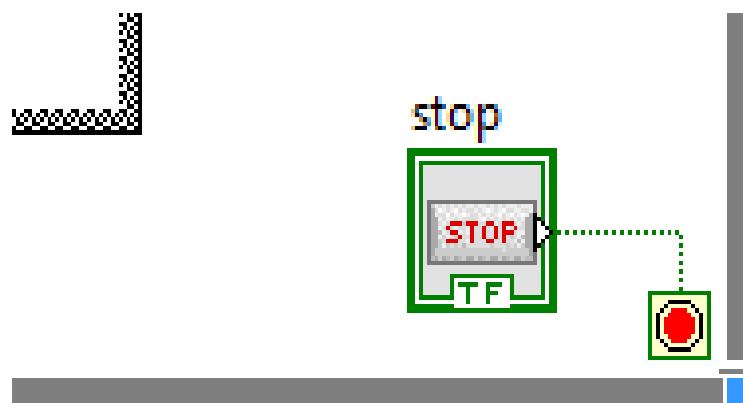


**Figura 28.** Estructura *While Loop* desplegada.

Haga clic derecho con el ratón sobre el terminal condición (*Loop condition*) de la estructura *While Loop*, y escoja la opción *Create Control*, véase las **Figuras 29 y 30**.



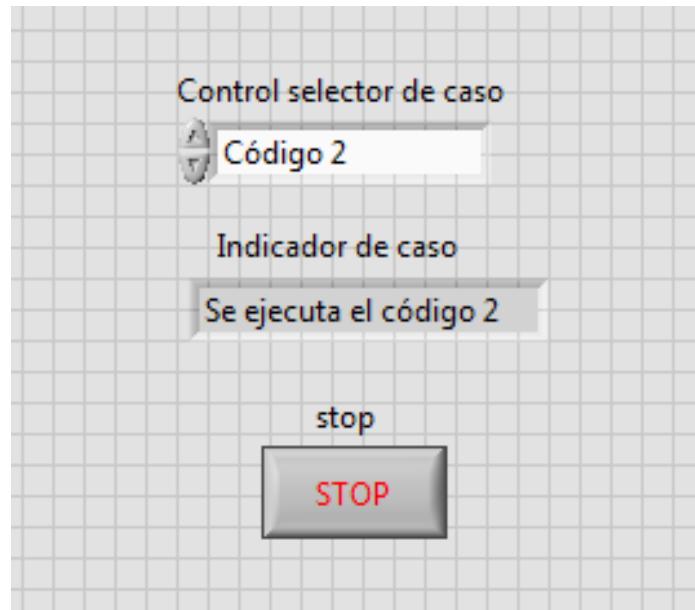
**Figura 29.** Creación del control condicional de la estructura *While Loop*.



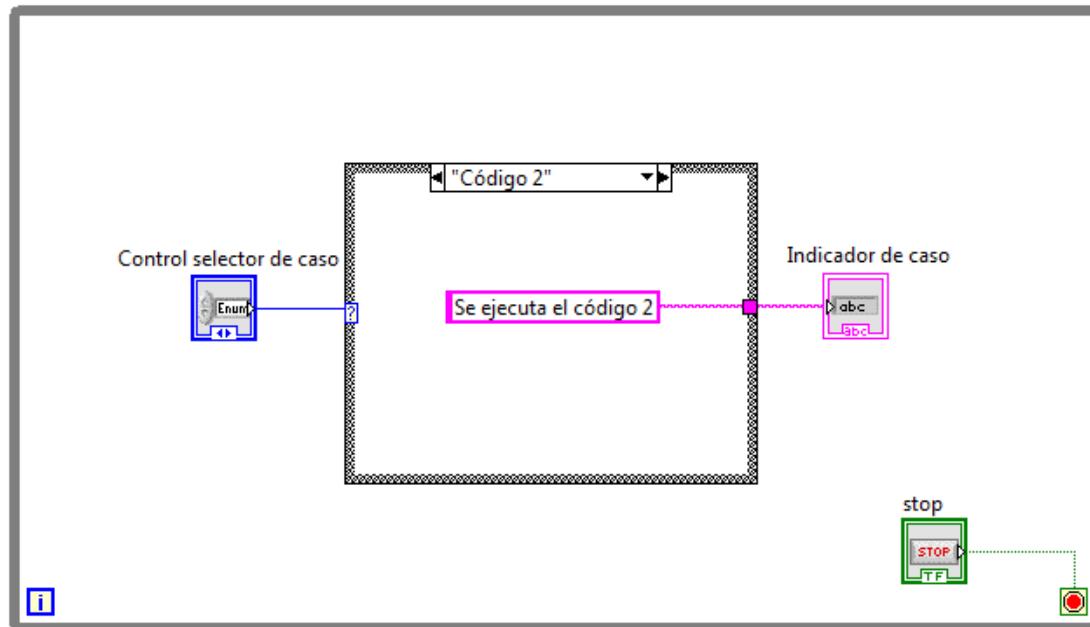
**Figura 30.** Control dispuesto dentro de la estructura.

En las **Figuras 31** y **32**, son mostrados los progresos en el VI tanto desde el *Front Panel* como desde el *Block Diagram*.

Ejecute el código pero haciendo uso del modo de ejecución *Run*, observe como el VI se mantiene ejecutado hasta que la condición de parada (*Stop*) sea activada.



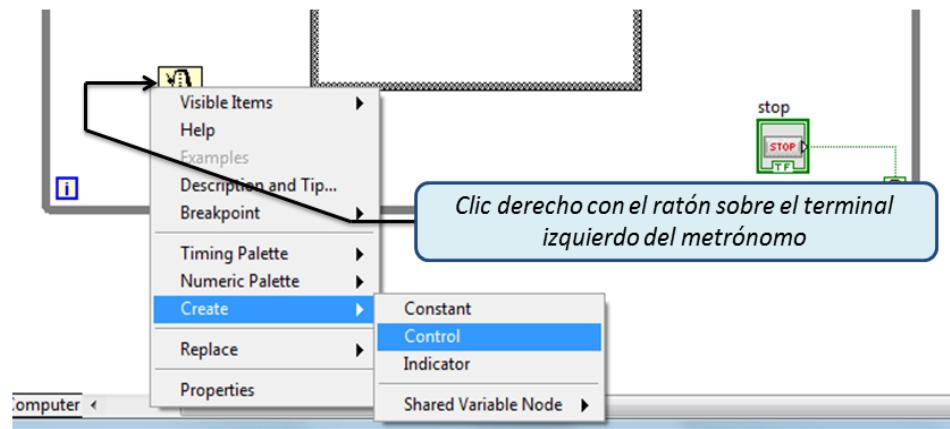
**Figura 31.** *Front Panel* del VI con estructura *While Loop*.



**Figura 32.** Block Diagram del VI con estructura *While Loop*.

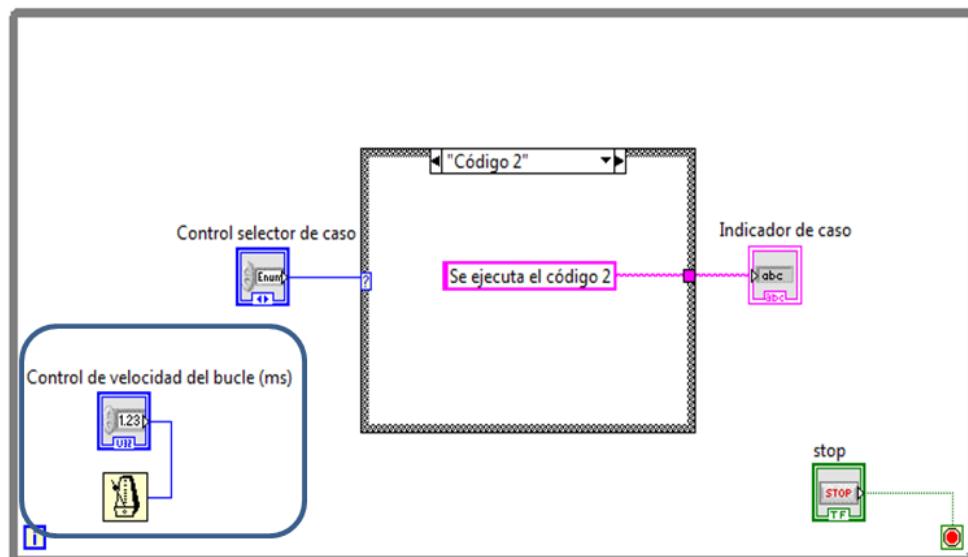
Observe que los cambios realizados por el *Control selector de caso* se ven reflejados instantáneamente en el *Indicador de caso*, esto es debido a que la estructura *While Loop* se encuentra ejecutada cíclicamente a una alta velocidad.

Es posible establecer o controlar la velocidad de las iteraciones del bucle (velocidad de actualización) haciendo uso de la herramienta con “forma de metrónomo” *Wait Until Next ms Multiple* presente en el *Functions / Timing*, por lo tanto inserte dicha herramienta dentro de la estructura *While Loop* y posteriormente creé un control asociado a ella, vea la **Figura 33**.

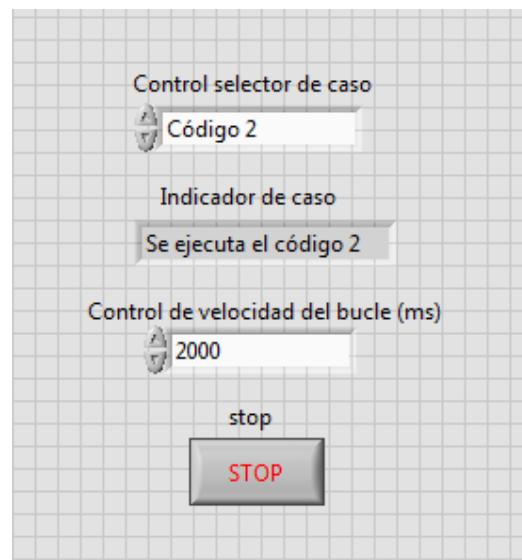


**Figura 33.** Creación de control de la herramienta *Wait Until Next ms Multiple*.

El control asociado a la herramienta *Wait Until Next ms Multiple* es un control numérico de datos largos enteros sin signo (32 bits), con el que será posible insertar la velocidad del paso de iteraciones del bucle (en milisegundos); en la **Figura 34** se puede observar el control dispuesto en el *Block Diagram* ya configurado con su etiqueta y en la **Figura 35** dispuesto en el *Front Panel* con ajuste de tiempo de 2000 ms (2 segundos).



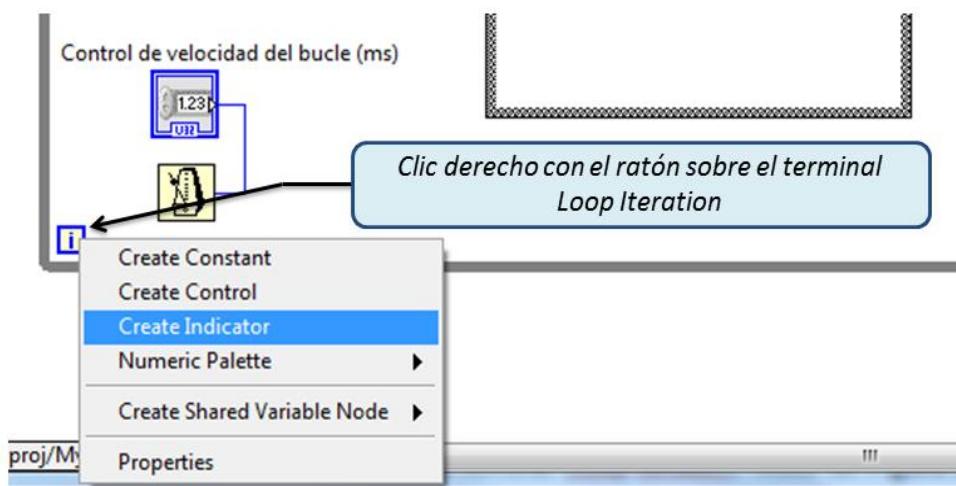
**Figura 34.** Herramienta *Wait Until Next ms Multiple* y *Control de velocidad del bucle (ms)* dispuesto en el *Block Diagram*.



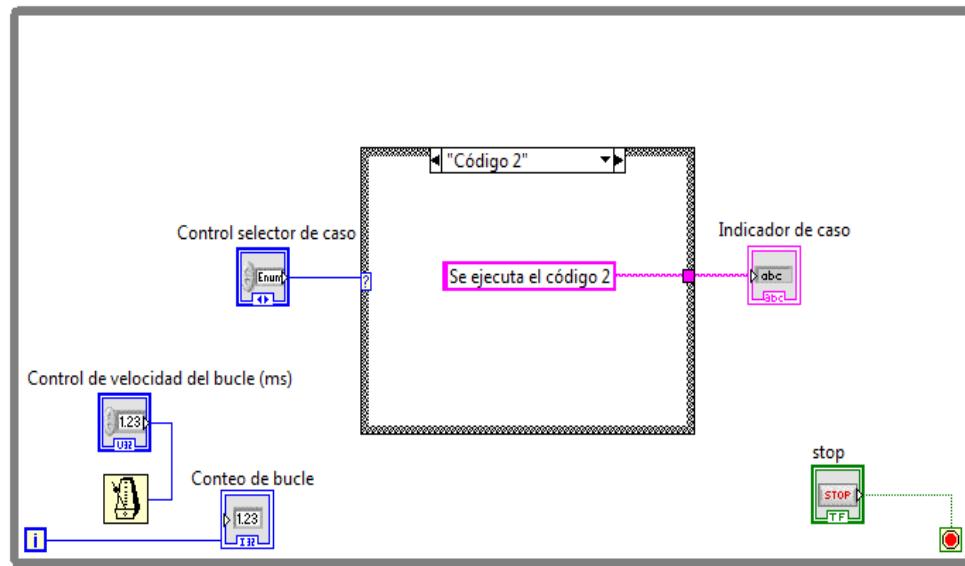
**Figura 35.** Control de velocidad del bucle (ms) dispuesto en el *Front Panel*.

Ejecute el VI y observe como al cambiar el caso en el *Control selector de caso* existe un retardo de 2 segundos en la actualización del *Indicador de caso*.

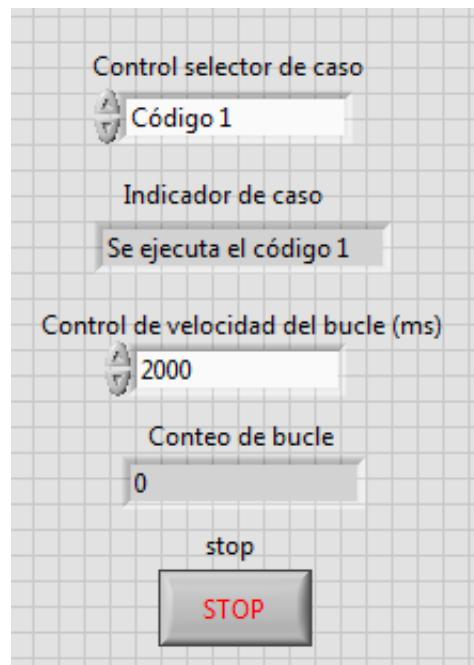
Para evidenciar de manera más adecuada la velocidad del paso del bucle asocie un indicador numérico al terminal *Loop iteration*, vea las **Figura 36, 37 y 38**.



**Figura 36.** Creación de Indicador numérico asociado a terminal *Loop iteration*.

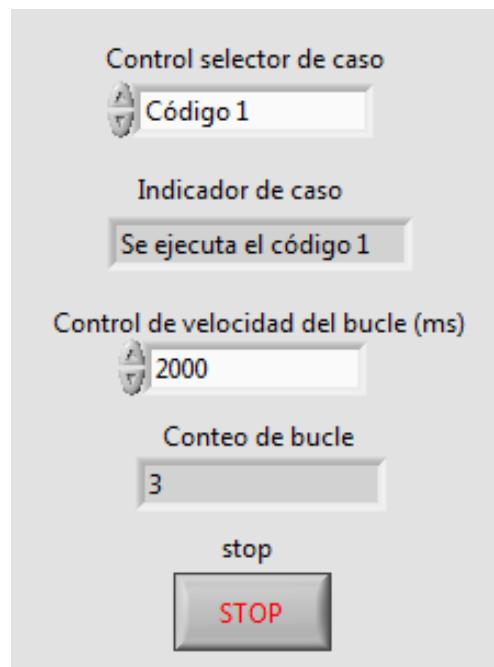


**Figura 37.** Indicador *Conteo de bucle* dispuesto en el *Block Diagram*.



**Figura 38.** Indicador *Conteo de bucle* dispuesto en el *Front Panel*.

Ejecute el VI haciendo uso del modo *Run*, note como el indicador *Conteo de bucle* cuenta las iteraciones cada 2 segundos, ver **Figura 39**.



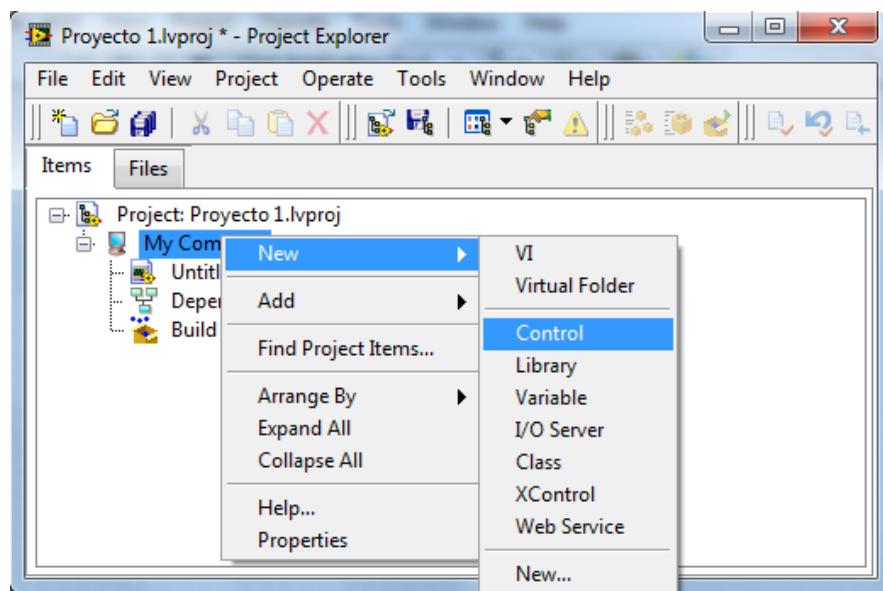
**Figura 39.** VI finalizado con el indicador *Conteo de bucle* funcionando.

#### 4.6. Implementación de la herramienta *Image Navigator* para el diseño de HMIs.

Al estar integrado en LabVIEW el módulo *Datalogging and Supervisory Control* (DSC) es posible acceder a la herramienta *Image Navigator*, con esta será posible diseñar el estilo visual de los VIs que posteriormente serán las Interfaces Humano-Maquina (HMIs) para las prácticas posteriores.

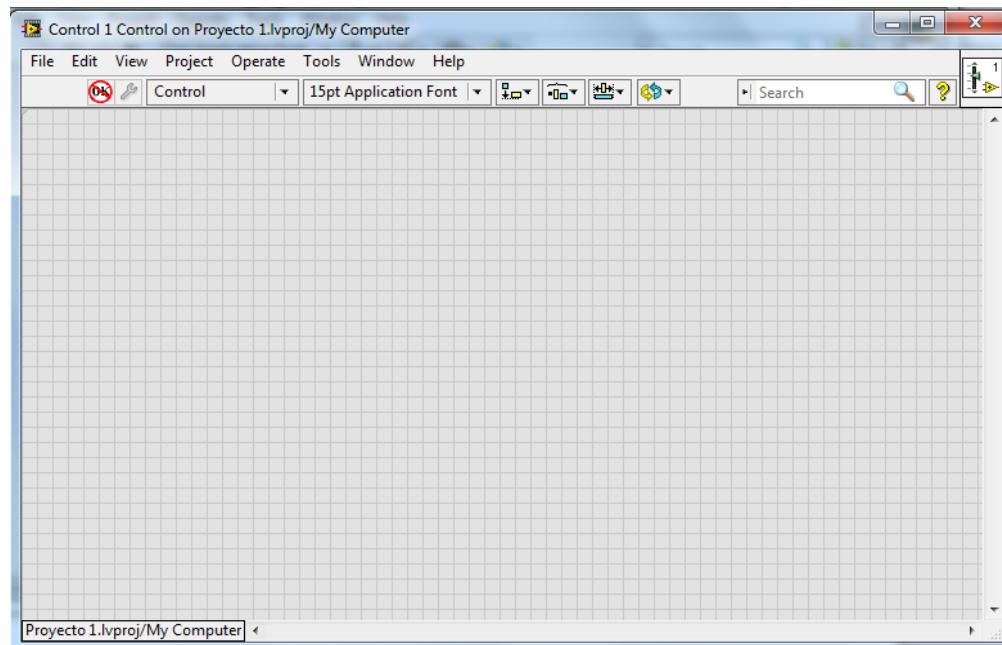
Para exemplificar la utilidad de la herramienta *Imagen Navigator* se creará un nuevo VI donde se podrá visualizar el encendido y apagado de un motor.

Seleccione *Project Explorer / My Computer / New / Control*, ver **Figura 40**.



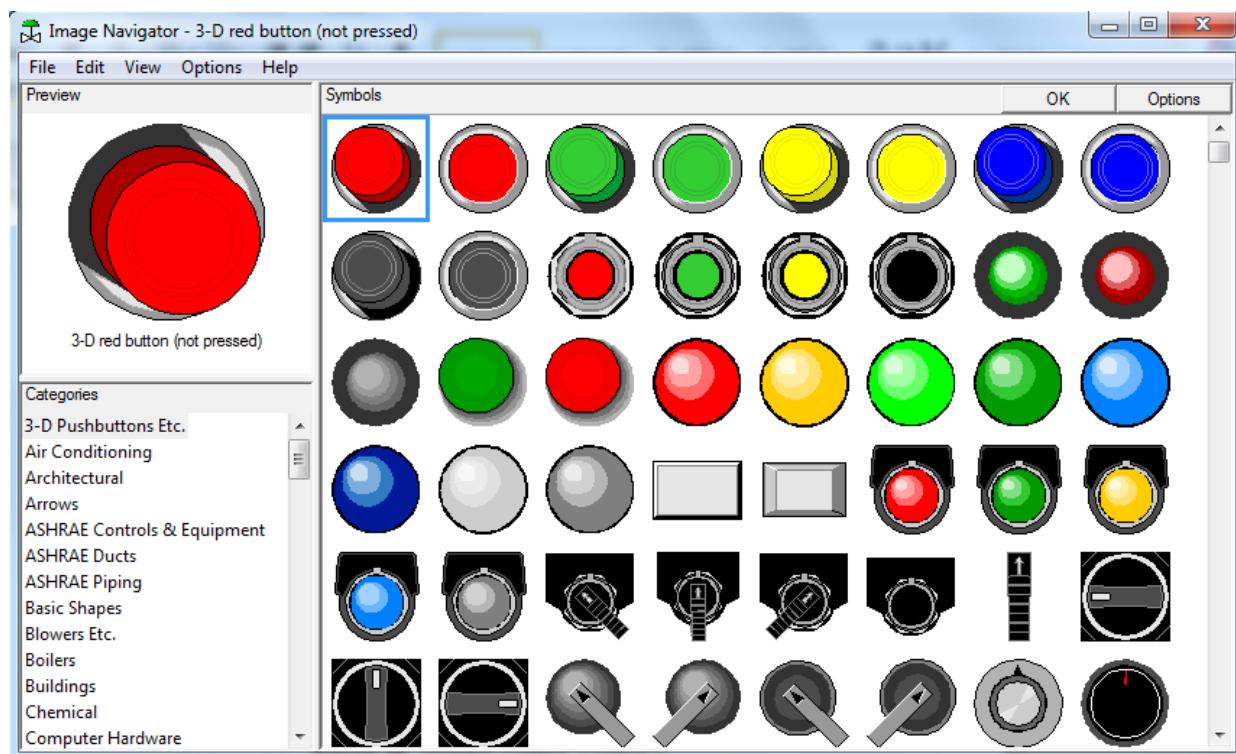
**Figura 40.** Creación de una nueva ventana *Control*.

Se abrirá una nueva ventana llamada *Control 1*, esta es similar al *Front Panel* de un VI, en ella se configurarán los elementos visuales (controles e indicadores) del VI, ver **Figura 41**.



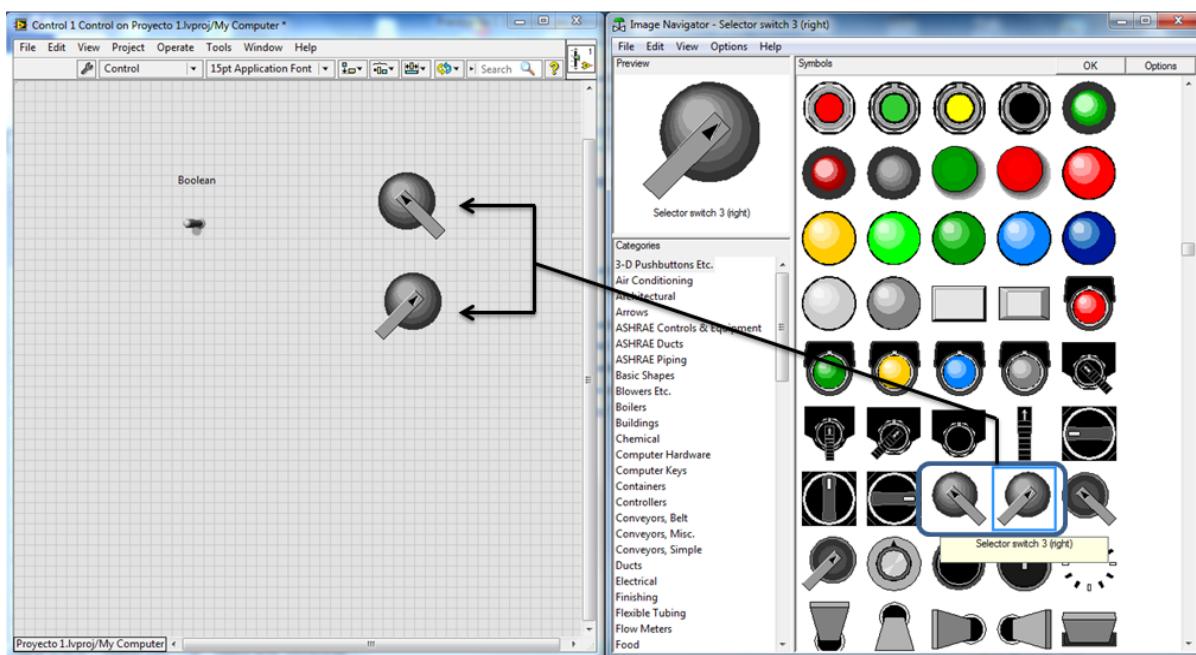
**Figura 41.** Ventana *Control 1*.

Vaya a *Tools / DSC Module / Image Navigator*, esta acción abrirá la herramienta *Image Navigator*, ver **Figura 42**.



**Figura 42.** Ventana *Image Navigator*.

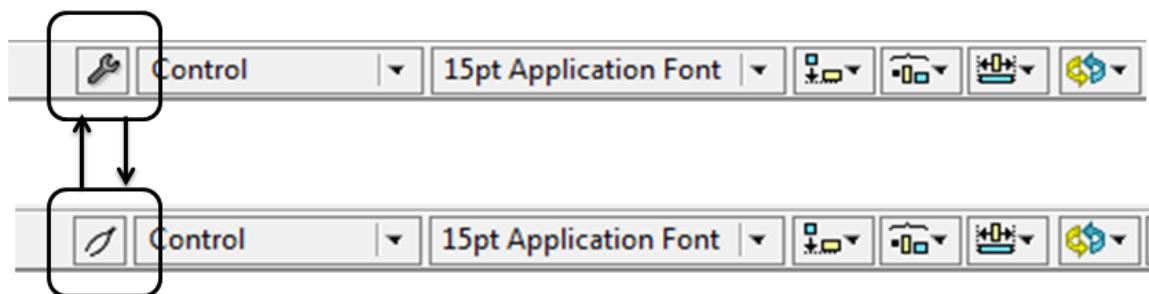
Vuelva a *Control 1*, en esta ventana también es posible acceder a la paleta de herramientas *Controls*, despliegue un control booleano *Horizontal Toogle Switch* sobre el área de trabajo, luego desde *Image Navigator*, seleccione *Categories / 3-D Pushbuttons Etc.*, escoja y arrastre hacia el área de trabajo los *Symbols* (*símbolos*) *Selector switch 3 (left)* y *Selector switch 3 (right)*, en la **Figura 43** se visualiza este procedimiento.



**Figura 43.** Control Booleano *Horizontal Toggle Switch* y símbolos *Selector switch 3* desplegados en *Control 1*.

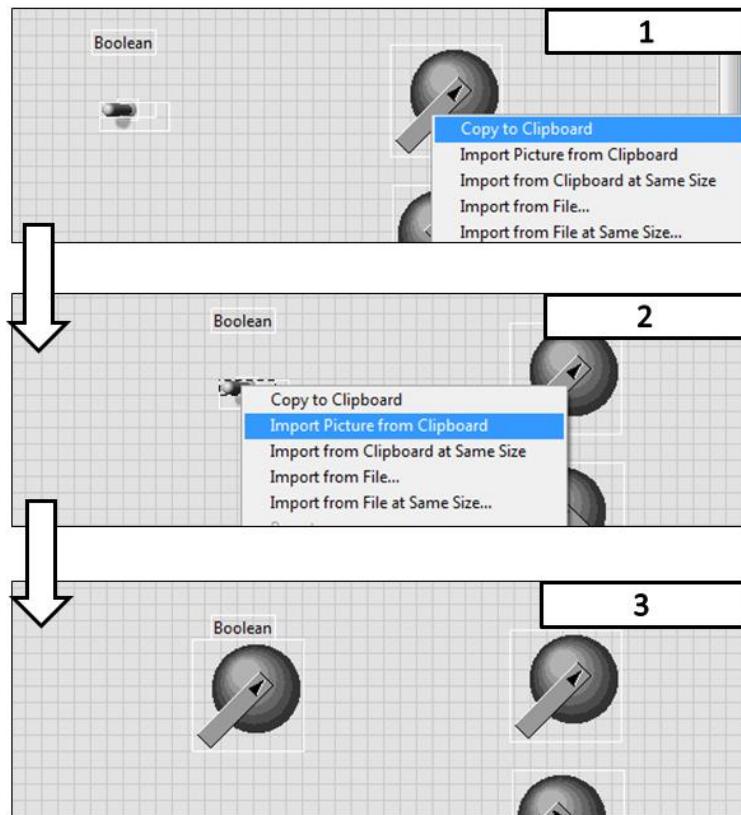
Ahora se asociarán los símbolos con el control *Boolean*, para realizar esta operación hay que tener en cuenta que por cada estado (*true* y *false*) del control irá asociado un símbolo, es decir, para el estado *false* deberá aparecer el símbolo *Selector switch 3 (right)* y para el estado *true* el símbolo *Selector switch 3 (left)*.

En *Control 1* haga clic con el ratón sobre el botón de llave ajustable *Change to Customize Mode*, hecho esto se cambiará al modo de “personalización” y además se modificará el botón para ahora ser una pinza, durante este modo es posible enlazar o mejor dicho copiar los símbolos al control booleano, ver **Figura 44**.



**Figura 44.** Change to Customize Mode (personalización) y Change to Edit Mode.

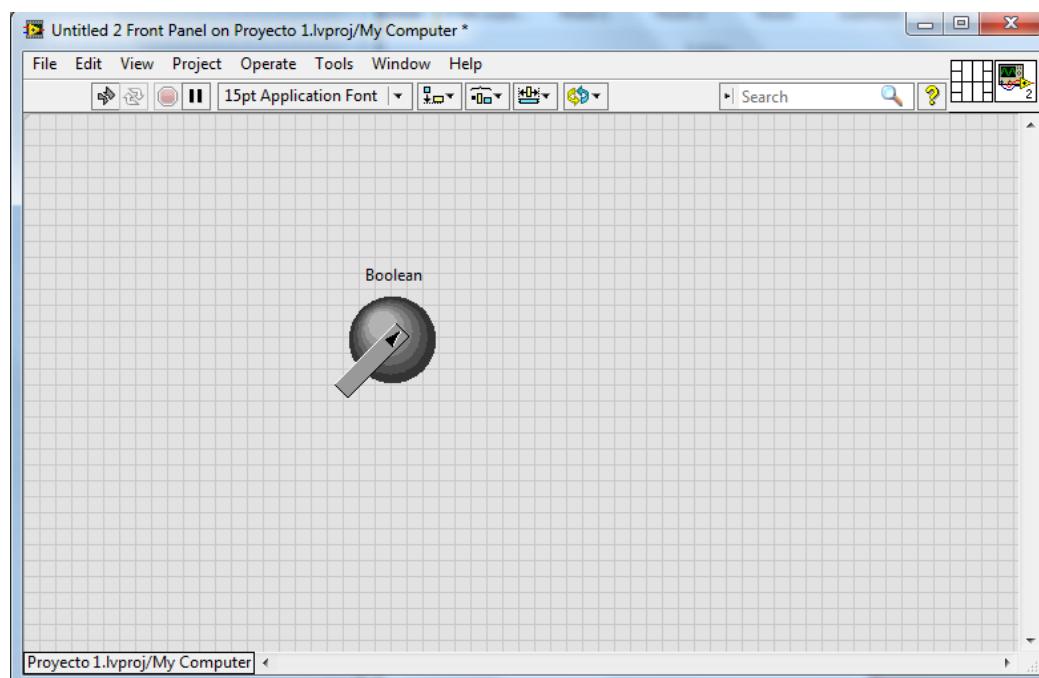
Haga clic derecho sobre el símbolo *Selector switch 3 (right)*, seleccione la opción *Copy to Clipboard*, posteriormente haga clic derecho sobre el control booleano y seleccione *Import Picture from Clipboard*, de esta manera se copiará el símbolo al estado actual del control (*false*), vea la **Figura 45** donde se puede observar este procedimiento.



**Figura 45.** Símbolo *Selector switch 3 (right)* asociado en control *Boolean* en estado *false*.

Seguidamente cambie de modo *Change to Edit Mode*, pulse el control *Boolean* y cambiará de estado, vuelva a cambiar de modo *Change to Customize Mode* y repita el procedimiento de la **Figura 45**, pero ahora con el símbolo *Selector switch 3 (left)*.

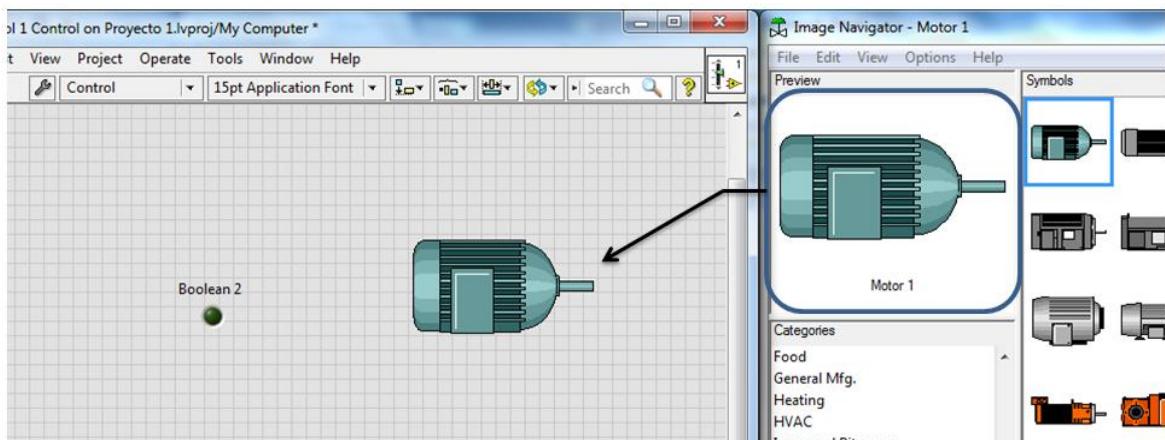
Finalizada la personalización del nuevo control, cree un nuevo VI y despliegue el control programado en el *Front Panel*, ver **Figura 46**.



**Figura 46.** Control *Boolean* personalizado desplegado en *Front Panel* del nuevo VI.

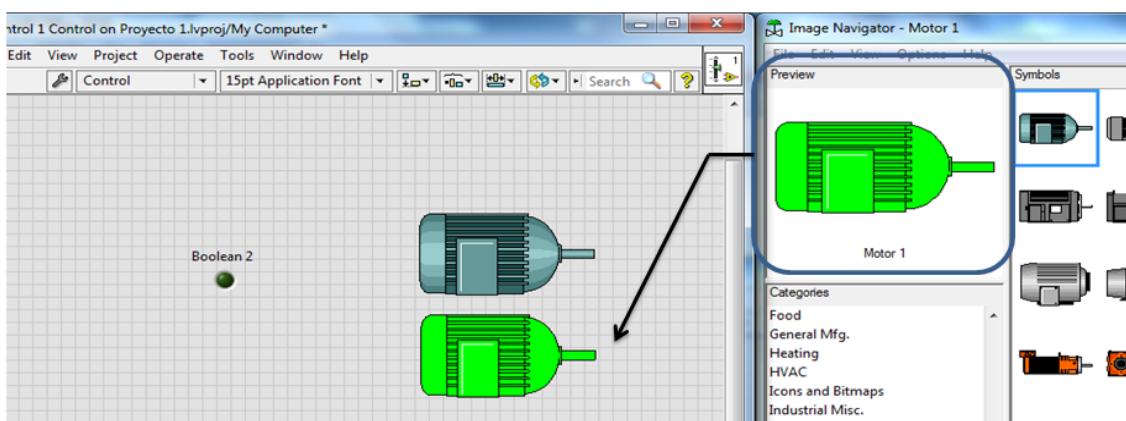
Vuelva a la ventana *Control 1* y elimine el control *Boolean*, inmediatamente despliegue un indicador booleano *Round LED*, el cual irá asociado a la figura de un motor.

Diríjase a *Image Navigator / Categories / Motors* y seleccione *Motor 1*, arrástrelo con el mouse hacia el área de trabajo de la ventana *Control 1*, ver **Figura 47**.



**Figura 47.** Indicador Booleano *Round LED* y símbolo *Motor 1* desplegados en *Control 1*.

Posteriormente, desde *Image Navigator* haga clic derecho sobre *Motor 1* y seleccione *Symbol Options*, se abrirá una ventana llamada *Symbol Options*, en el campo *Fill Color Mode* seleccione *Solid* y escoja en la paleta de colores una tonalidad verde (en este caso), note como toda la figura del motor cambia a color verde sólido, luego seleccione *Ok* e inmediatamente despliegue desde la ventana *Preview* el símbolo del *Motor 1* hacia el área de trabajo de *Control 1*, ver **Figura 48**.



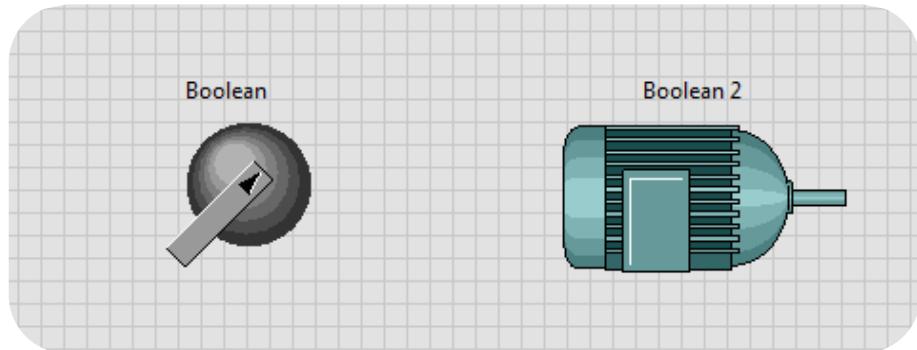
**Figura 48.** Símbolo *Motor 1* con color verde sólido desplegado en *Control 1*.

Asocie el indicador *Boolean 2 (Round LED)* y los símbolos *Motor 1* para cada estado respectivo implementando la metodología planteada ya en el control *Boolean*.

Finalizada la personalización del indicador *Boolean 2* cópielo en el VI, luego conecte desde el *Block Diagram* al control con el indicador. En las **Figura 49** y **50** se muestran el *Block Diagram* y el *Front Panel* del VI finalizado.

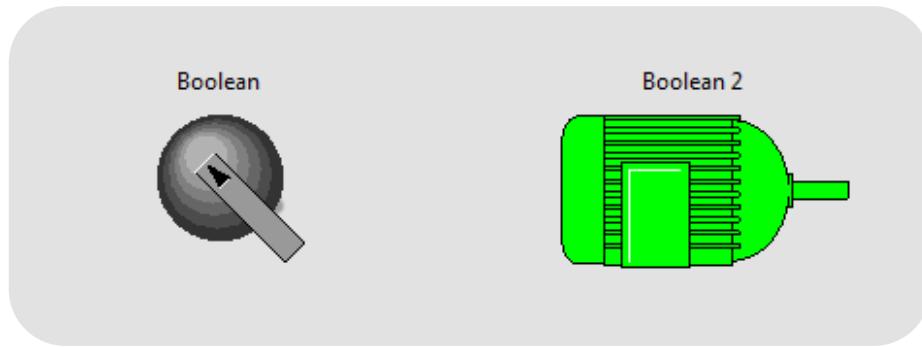


**Figura 49.** *Block Diagram* del VI encendido y apagado de un motor.



**Figura 50.** *Front Panel* del VI encendido y apagado de un motor.

Compruebe el funcionamiento del VI haciendo uso del modo *Run Continuously*, en la **Figura 51** aparece el motor encendido cuando el control *Boolean* se encuentra en estado *true*.



**Figura 51.** *Front Panel* del VI encendido y apagado de un motor en ejecución *Run Continuously*.

## **PRÁCTICA N° 2**

*Integración entre un PLC Siemens S7-1200 y LabVIEW para el desarrollo de Interfaces  
Humano Maquina.*

## ÍNDICE GENERAL

	Pág.
1. OBJETIVO GENERAL .....	1
2. LISTA DE MATERIALES Y EQUIPOS.....	1
3. FUNDAMENTACION TEORICA .....	1
3.1. PLC Siemens SIMATIC S7-1200 .....	1
3.1.1. Lenguajes de programación .....	3
3.1.2. Bloques de Organización y contactos .....	5
3.1.3. Temporizadores y contadores .....	7
3.1.4. Direccionamiento de áreas de memoria .....	12
3.2. Servidor OPC .....	14
3.3. Interfaz Humano Maquina (HMI).....	15
4. PROCEDIMIENTO .....	16
4.1. Creación de un proyecto en TIA PORTAL .....	16
4.2. Configuración del servidor OPC.....	24
4.3. Configuración de LabVIEW.....	37
4.4. Crear VI del proyecto .....	42
4.5. Adicionar al proyecto un temporizador controlado y supervisado desde LabVIEW .....	46
4.6. Adicionar al proyecto un contador controlado y supervisado desde LabVIEW .....	53
4.7. Actividad (alternativa).....	60

## ÍNDICE DE FIGURAS

	Pág.
Figura 1. PLC Siemens SIMATIC S7-1200 (1214C) .....	2
Figura 2. Esquema de contactos KOP .....	4
Figura 3. Esquema de contactos KOP con flujo invertido de corriente .....	5
Figura 4. Esquema de contactos KOP con rama causando un cortocircuito en el contacto “B”.....	5
Figura 5. TON (temporizador con retardo al conectar).....	7
Figura 6. Diagrama de tiempo de un TON. ....	8
Figura 7. TOF (temporizador con retardo al desconectar). ....	9
Figura 8. Diagrama de tiempo de un TOF.....	9
Figura 9. CTU (Contador ascendente).....	10
Figura 10. Diagrama de funcionamiento de un CTU. ....	11
Figura 11. CTD (Contador descendente). ....	11
Figura 12. Diagrama de funcionamiento de un CTD. ....	12
Figura 13. Esquema de memoria y direccionamientos. ....	13
Figura 14. Esquema de funcionamiento de un servidor OPC. ....	15
Figura 15. Vista del portal ventana <i>Iniciar</i> . ....	16
Figura 16. Vista del portal <i>Primeros pasos</i> . ....	17
Figura 17. Agregar dispositivo ( <i>CPU 1200 sin especificar</i> ). ....	17
Figura 18. Vista del proyecto, selección de la opción <i>determinar</i> , para posterior detección.....	18
Figura 19. Ventana <i>Detección de Hardware</i> . ....	19
Figura 20. Detección finalizada.....	19
Figura 21. Vista del proyecto con el PLC SIMATIC S7-1200 detectado. ....	20
Figura 22. Selección del <i>Main [OB1]</i> .....	20
Figura 23. Primer Programa creado en lenguaje KOP.....	21
Figura 24. <i>Establecer conexión online</i> con el PLC.....	22
Figura 25. Descargar el programa al PLC.....	22

Figura 26. Selección del botón del modo <i>Run</i> .....	22
Figura 27. Monitoreo del programa descargado al PLC modo <i>Run</i> con <i>Salida</i> inactiva. .....	23
Figura 28. Monitoreo del programa descargado al PLC modo <i>Run</i> con <i>Salida</i> activa. .....	23
Figura 29. NI OPC Server. ....	24
Figura 30. Ventana de identificación del canal ( <i>Identification</i> ). ....	25
Figura 31. Ventana de selección del driver correspondiente al PLC ( <i>Device Driver</i> ). 25	25
Figura 32. Ventana de selección del adaptador de red del ordenador ( <i>Network Interface</i> ). ....	26
Figura 33. Ventana <i>Write Optimizations</i> .....	26
Figura 34. Ventana <i>Non-Normalized Float Handling</i> .....	27
Figura 35. Ventana de resumen de configuraciones ( <i>Summary</i> ). .....	27
Figura 36. Añadir dispositivo (PLC).....	28
Figura 37. Ventana de asignación del nombre del controlador ( <i>Name</i> ). .....	28
Figura 38. Ventana de selección del modelo de controlador ( <i>Model</i> ). .....	29
Figura 39. Ventana de identificación del dispositivo en la red (ID). ....	30
Figura 40. Ventana de configuración de barrido de datos ( <i>Scan Mode</i> ). .....	30
Figura 41. Ventana parámetros de temporización ( <i>Timing</i> ). ....	31
Figura 42. Ventana <i>Auto-Demotion</i> .....	31
Figura 43. Ventana de parámetros de comunicaciones ( <i>Communications Parameters</i> ). .....	32
Figura 44. Ventana de parámetros de configuración de controladores Siemens ( <i>S7 Communications Parameters</i> ).....	32
Figura 45. Ventana de opciones de direccionamiento ( <i>Addressing Options</i> ). ....	33
Figura 46. Ventana de resumen ( <i>Summary</i> ). .....	33
Figura 47. Añadir nuevo <i>tag</i> . ....	34
Figura 48. Ventana <i>Tag Properties</i> (Propiedades del Tag).....	35
Figura 49. <i>Tags</i> creados. ....	35
Figura 50. Botón de herramienta de monitoreo <i>Quick Client</i> .....	36
Figura 51. <i>Quick Client</i> ejecutado y monitoreando los <i>tags</i> creados.....	36

Figura 52. Cambio de valor (estado) del <i>tag Salida</i> .....	37
Figura 53. Selección de un nuevo <i>I/O Server</i> .....	38
Figura 54. Selección de <i>OPC Client</i> .....	38
Figura 55. Ventana de configuración del <i>OPC Client</i> . ....	39
Figura 56. <i>Untitled Library 1</i> y <i>OPC1</i> agregados.....	39
Figura 57. Creación de la variable <i>Arranque Remoto</i> . ....	40
Figura 58. Enlace de la variable <i>Arranque Remoto</i> .....	41
Figura 59. Variable <i>Arranque Remoto</i> configurada. ....	41
Figura 60. Variables creadas y configuradas dispuestas en la librería. ....	42
Figura 61. <i>Block Diagram</i> del VI en creación. ....	43
Figura 62. Despliegue de las variables en el <i>Front Panel</i> . ....	44
Figura 63. Cambio de tipo de acceso de <i>Read</i> a <i>Write</i> .....	44
Figura 64. <i>Block Diagram</i> del VI.....	45
Figura 65. <i>Front Panel</i> del VI. ....	45
Figura 66. <i>Front Panel</i> del VI ejecutado con <i>Salida</i> (Q0.0) activada.....	46
Figura 67. Temporizador TON integrado a la lógica. ....	47
Figura 68. Creación del tag <i>Variable de Temporización 1</i> . ....	48
Figura 69. <i>Tags Variable de Temporización 1</i> y <i>Tiempo de Habilitación</i> añadidos al servidor OPC.....	49
Figura 70. Creación de la variable <i>Tiempo de Habilitación</i> . ....	49
Figura 71. Creación de <i>Variable de Temporización 1</i> .....	50
Figura 72. Variables <i>Tiempo de Habilitación</i> y <i>Variable de Temporización 1</i> agregadas a la librería. ....	50
Figura 73. <i>Front Panel</i> del VI con control e indicador de temporización integrados a la interfaz gráfica (VI). ....	51
Figura 74. <i>Block Diagram</i> del VI con control e indicador de temporización integrados. ....	52
Figura 75. Contador CTU integrado a la lógica. ....	53
Figura 76. Creación del <i>tag Valor de Conteo actual</i> .....	55
Figura 77. Creación del <i>tag Reiniciar Contador</i> . ....	55

Figura 78. Tags <i>Valor de Conteo actual</i> y <i>Reiniciar Contador</i> añadidos al servidor OPC.....	56
Figura 79. Creación de la variable <i>Valor de Conteo actual</i> .....	56
Figura 80. Creación de la variable <i>Reiniciar Contador</i> .....	57
Figura 81. Variables <i>Valor de Conteo Actual</i> y <i>Reiniciar Contador</i> anexadas a la librería.....	57
Figura 82. <i>Front Panel</i> del VI con control e indicador del contador integrados a la interfaz.....	58
Figura 83. <i>Block Diagram</i> del VI con control e indicador de conteo integrados. ....	59
Figura 84. VI Finalizado en modo de ejecución (contador agregado). ....	59
Figura 85. Segmento 1.....	62
Figura 86. Segmento 2.....	62
Figura 87. Segmento 3.....	63
Figura 88. Segmento 4.....	63
Figura 89. Segmento 4.....	64
Figura 90. Segmento 6.....	64
Figura 91. Segmento 7.....	65
Figura 92. Segmento 8.....	65
Figura 93. Segmento 9.....	66
Figura 94. Segmento 10.....	66
Figura 95. Segmento 11.....	67
Figura 96. Segmento 12.....	67
Figura 97. Segmento 13.....	68
Figura 98. Segmento 14.....	68
Figura 99. Segmento 15.....	69
Figura 100. Segmento 16.....	69
Figura 101. Segmento 17.....	70
Figura 102. Segmento 18.....	70
Figura 103. Segmento 19.....	71
Figura 104. Segmento 20.....	71
Figura 105. Segmento 21.....	72

Figura 106. <i>Tags</i> creados en el servidor OPC .....	72
Figura 107. Proyecto diseñado en LabVIEW.....	73
Figura 108. <i>Block Diagram</i> de la interfaz.....	74
Figura 109. HMI (VI) del mezclador.....	75

## ÍNDICE DE TABLAS

	Pág.
Tabla 1. Especificaciones de las diferentes CPUs del PLC SIMATIC S7-1200. ....	3
Tabla 2. Tipos de datos.....	14

## 1. OBJETIVO GENERAL

Comunicar el PLC Siemens SIMATIC S7-1200 con el programa LabVIEW para el control y supervisión de variables de proceso.

## 2. LISTA DE MATERIALES Y EQUIPOS

- Ordenador con los programa TIA Portal, National Instruments OPC Server y LabVIEW con los módulos Datalogging and Supervisory Control (DSC) y Real Time integrados.
- PLC Siemens SIMATIC S7-1200 1214C AC/DC/RLY.
- Cable UTP RJ45/RJ45.

## 3. FUNDAMENTACION TEORICA

### 3.1. PLC Siemens SIMATIC S7-1200

El controlador SIMATIC S7-1200 ofrece la flexibilidad y capacidad de controlar una gran variedad de dispositivos para las distintas tareas de automatización, posee un diseño compacto, su CPU incorpora un microprocesador, una fuente de alimentación integrada, así como circuitos de entrada y salida en una carcasa compacta, conformando así un potente PLC. Una vez cargado el programa en el CPU, ésta contiene la lógica necesaria para vigilar y controlar los dispositivos de la aplicación. La CPU vigila las entradas y cambia el estado de las salidas según la lógica del programa de usuario, que puede incluir lógica booleana, instrucciones de contaje y temporización,

funciones matemáticas complejas, así como comunicación con otros dispositivos inteligentes. En la **Figura 1** es mostrada la apariencia física del controlador.



**Figura 1.** PLC Siemens SIMATIC S7-1200 (1214C).

Para ser más específicos el controlador SIMATIC S7-1200 incluye:

- PROFINET Incorporado
- E/S de alta velocidad aptas para el control de movimiento, entradas analógicas incorporadas para minimizar el espacio requerido y excluir la necesidad de E/S adicionales, dos (2) generadores de impulsos para aplicaciones con modulación de ancho de pulso, y seis (6) contadores rápidos
- Diferentes entradas y salidas incorporadas en los módulos CPU que proporcionan entre seis (6) y catorce (14) salidas dependiendo del tipo de CPU que use.
- Módulos de señales para DC, relé o E/S analógicas amplían el número de E/S, mientras que las innovadoras *SignalBoards* integradas en el frontal de la CPU proporcionan entradas y salidas adicionales.

Las especificaciones por cada CPU presente en la gama SIMATIC S7-1200 pueden verse en la **Tabla 1**.

**Tabla 1.** Especificaciones de las diferentes CPUs del PLC SIMATIC S7-1200.

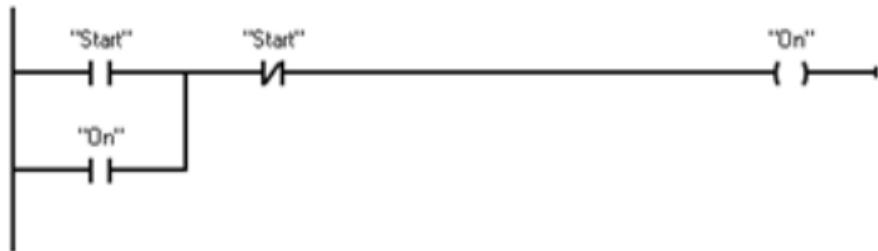
Función	CPU 1211C	CPU 1212C	CPU 1214C
Dimensiones físicas (mm)	90 x 100 x 75	90 x 100 x 75	110 x 100 x 75
Memoria de usuario			
• Memoria de trabajo	• 25 KB	• 25 KB	• 50 KB
• Memoria de carga	• 1 MB	• 1 MB	• 2 MB
• Memoria remanente	• 2 KB	• 2 KB	• 2 KB
E/S Integradas locales			
• Digital	• 6 entradas 4 salidas	• 8 entradas 6 salidas	• 14 entradas 10 salidas
• Analógico	• 2 entradas	• 2 entradas	• 2 entradas
Tamaño de la memoria imagen de proceso			
• Entradas	• 1024 bytes	• 1024 bytes	• 1024 bytes
• Salidas	• 1024 bytes	• 1024 bytes	• 1024 bytes
Área de marcas (M)	4096 bytes	4096 bytes	8192 bytes
Ampliación con módulos de señales	Ninguno	2	8
Signal Board	1	1	1
Módulos de comunicación	3	3	3
Contadores rápidos	3	4	6
• Fase simple	• 3 a 100 kHz	• 3 a 100 kHz 1 a 30 kHz	• 3 a 100 kHz 3 a 30 kHz
• Fase en cuadratura	• 3 a 80 kHz	• 3 a 80 kHz 1 a 20 kHz	• 3 a 80 kHz 3 a 20 kHz
Salidas de impulsos <sup>1</sup>	2	2	2
Memory Card (opcional)	Sí	Sí	Sí
Tiempo de respaldo del reloj en tiempo real	Típico: 10 días / Mínimo: 6 días a 40 °C		
Velocidad de ejecución de funciones matemáticas con números reales	18 µs/instrucción		
Velocidad de ejecución booleana	0,1 µs/instrucción		

### 3.1.1. Lenguajes de programación

Existen varios lenguajes compatibles con el controlador SIMATIC S7-1200, entre los cuales se tienen, el KOP (esquema de contactos), FUP (diagramas de bloques), SCL

(lenguaje de control estructurado), AWL (lista de instrucciones), SFC (secuencial), para efectos de este material instruccional se implementará exclusivamente el lenguaje KOP.

- **KOP (esquema de contactos):** es un lenguaje de programación gráfico. Su representación es similar a los esquemas de circuitos, ver **Figura 3**.



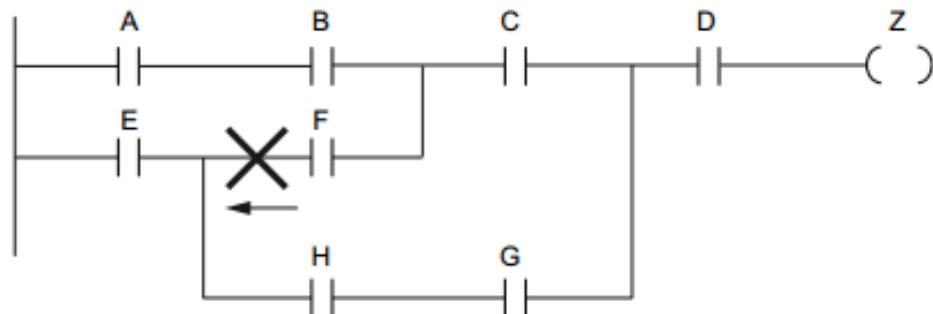
**Figura 2.** Esquema de contactos KOP.

Para crear la lógica de operaciones complejas, es posible insertar ramas para los circuitos paralelos. Las ramas paralelas se abren hacia abajo o se conectan directamente a la barra de alimentación. Las ramas se terminan hacia arriba.

KOP ofrece instrucciones con cuadros para numerosas funciones, p. ej. Matemáticas, temporizadores, contadores y transferencia.

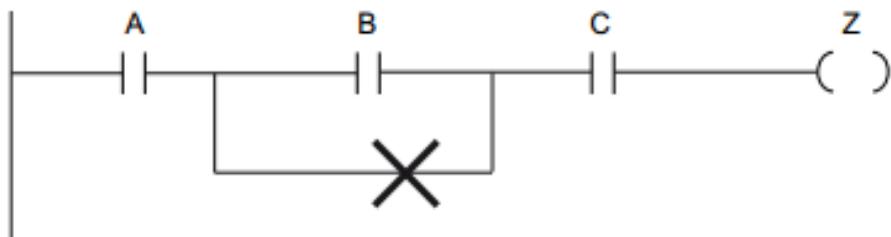
Tenga en cuenta las reglas siguientes al crear segmentos KOP:

- ✓ Todo segmento KOP debe terminar con una bobina o cuadro. No termine un segmento con una instrucción de comparación ni de detección de flancos (ascendentes o descendentes).
- ✓ No se permite programar ramas que puedan ocasionar un flujo invertido de la corriente, ver Figura 3.



**Figura 3.** Esquema de contactos KOP con flujo invertido de corriente.

- ✓ No se permite programar ramas que causen cortocircuitos, ver **Figura 4.**



**Figura 4.** Esquema de contactos KOP con rama causando un cortocircuito en el contacto “B”.

### 3.1.2. Bloques de Organización y contactos.

- **Bloque de organización (OB):** los bloques de organización permiten estructurar el programa. Estos bloques sirven de interfaz entre el sistema operativo y el programa de usuario. Los OBs son controlados por eventos. Un evento (por ejemplo una alarma de diagnóstico o un intervalo) hace que la CPU ejecute un OB.

El OB de ciclo contiene el programa principal. Es posible incluir más de un OB de ciclo en el programa de usuario. En estado operativo Run (el estado Run se refiere a cuando la CPU del PLC está encendida) los OBs de ciclo se ejecutan en el nivel de prioridad más bajo y pueden ser interrumpidos por todos los demás tipos de procesamientos del programa. El OB de arranque no interrumpe el OB

de ciclo, puesto que la CPU ejecuta el OB de arranque antes de pasar al estado operativo Run.

Tras finalizar el procesamiento de los OBs de ciclo, la CPU vuelve a ejecutarlos inmediatamente. Esta ejecución cíclica es el tipo de procesamiento "normal" que se utiliza para los controladores lógicos programables. En numerosas aplicaciones el programa de usuario entero está contenido en un solo OB de ciclo.

Es posible crear otros OBs para ejecutar funciones específicas, tales como el procesamiento de alarmas y el tratamiento de errores, o la ejecución de un código de programa específico en determinados intervalos. Estos OBs interrumpen la ejecución de los OBs de ciclo.

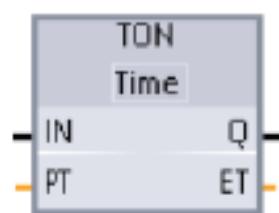
- **Contactor:** es un aparato mecánico de conexión y desconexión eléctrica, asociado por cualquier forma de energía, menos manual, capaz de establecer, soportar e interrumpir corrientes en condiciones normales del circuito, incluso las de sobrecarga. Para realizar el programa se utilizaran los siguientes contactores:
  - ✓ **Contacto normalmente abierto** ( ---| |--- ): la activación de un contacto normalmente abierto depende del estado lógico del operando correspondiente. Si el estado lógico del operando es uno (1), se cierra el contacto normalmente abierto y el estado lógico de la salida de la instrucción se pone en uno (1). Si el estado lógico del operando es cero (0), el contacto normalmente abierto no se activa y el estado lógico de la salida de la instrucción se pone en cero (0).
  - ✓ **Contacto normalmente cerrado** ( ---| / |--- ): la activación de un contacto normalmente cerrado depende del estado lógico del operando correspondiente. Si el estado lógico del operando es uno (1), se "abre" el

contacto y se interrumpe el flujo de corriente hacia la barra de alimentación derecha. En este caso, la salida de la operación devuelve el estado lógico cero (0).

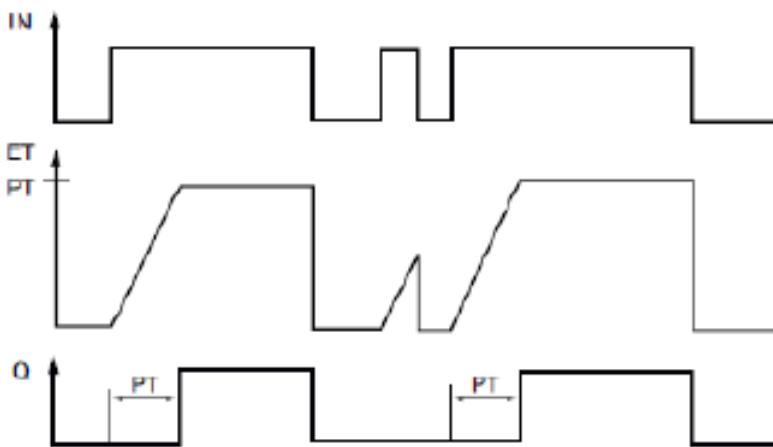
- ✓ **Asignación (---( )---**): la instrucción Asignación permite activar el bit de un operando indicado. Si el resultado lógico en la entrada de la bobina es uno (1), el operando indicado adopta el estado lógico uno (1). Si el estado lógico en la entrada de la bobina es cero (0), el bit del operando indicado se coloca en cero (0).

### 3.1.3. Temporizadores y contadores

- **Temporizadores:** las instrucciones con temporizadores se utilizan para crear retardos programados. Todos los temporizadores utilizan una estructura almacenada en un bloque de datos para mantener los datos. El bloque de datos se asigna al colocar la instrucción de temporización en el editor. Se procederá a explicar el funcionamiento de dos tipos de temporizadores, los cuales son, el TON (Retardo al conectar) y el TOF (Retardo al desconectar).
- ✓ **Temporizador con retardo al conectar (TON):** se utiliza para activar una salida (Q) tras un retardo preseleccionado, véase las **Figuras 5 y 6**.



**Figura 5.** TON (temporizador con retardo al conectar).



**Figura 6.** Diagrama de tiempo de un TON.

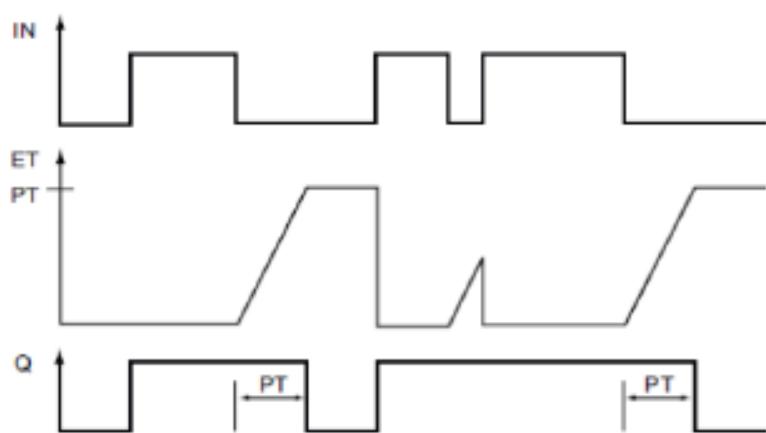
Cuando se activa el temporizador ( $IN = 1$ ), el temporizador de retardo al conectar espera un tiempo preseleccionado *Preset Time* (*PT*) hasta activar su salida ( $Q = 1$ ). La salida permanecerá activada mientras la entrada esté activada. El temporizador de retardo a la conexión utiliza un bloque de datos (DB) para conservar sus datos. Este bloque de datos se asigna al insertar la instrucción TON en el segmento.

Se puede introducir una constante para el parámetro del tiempo preseleccionado *Preset Time* (*PT*). En la aplicación también se puede introducir una dirección de memoria del tipo Time (valor de 4 bytes con signo) para almacenar el valor preseleccionado (ej. MD20). En caso necesario, esto permite al programa de usuario cambiar el valor de preselección conforme a las condiciones operativas. El tiempo transcurrido *Enable Time* (*ET*) también se puede almacenar en una dirección de memoria Time a la que pueden acceder otros elementos del programa de usuario.

- ✓ **Temporizador con retardo al desconectar (TOF):** se utiliza para mantener activada una salida ( $Q$ ) durante un tiempo preseleccionado tras haberse desactivado la entrada ( $IN$ ), vea **Figura 7** (bloque TOF) y **Figura 8** (diagrama de tiempo TOF).



**Figura 7.** TOF (temporizador con retardo al desconectar).

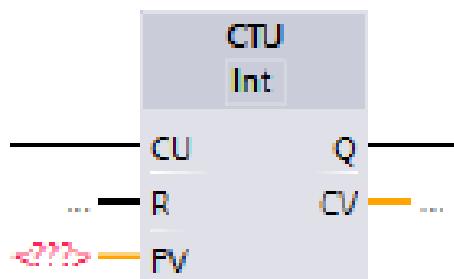


**Figura 8.** Diagrama de tiempo de un TOF.

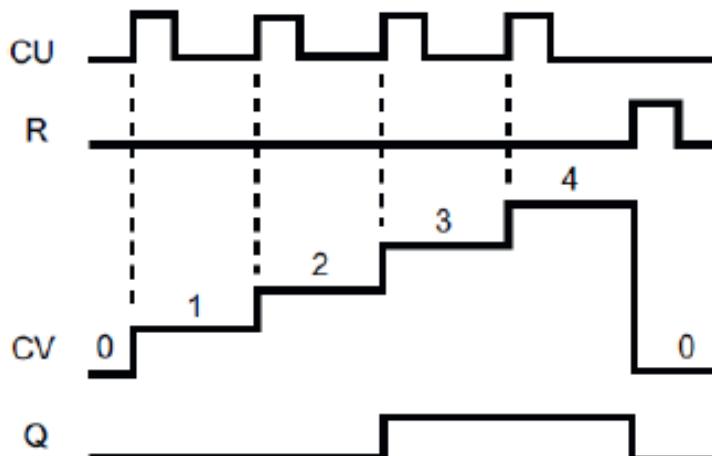
Al igual que el temporizador de retardo a la conexión, el temporizador de retardo a la desconexión utiliza un bloque de datos (DB) para conservar sus datos. Este bloque de datos se asigna al insertar la instrucción TOF introducirá una constante para el parámetro del tiempo preseleccionado (PT).

También es posible introducir una dirección de memoria Time para almacenar el valor de preselección. Esto permite al programa de usuario cambiar el valor de preselección según las condiciones operativas, si es necesario. El tiempo transcurrido (ET) también se puede almacenar en una dirección Time a la que pueden acceder otros elementos del programa de usuario.

- **Contadores:** las instrucciones con contadores se utilizan para contar eventos del programa internos y eventos del proceso externos. Todo contador utiliza una estructura almacenada en un bloque de datos para conservar sus datos. El bloque de datos se asigna al colocar la instrucción de conteo en el editor. Estas instrucciones utilizan contadores por software cuya frecuencia de conteo máxima limitada por la frecuencia de ejecución del OB en el que están contenidas. El OB en el que se depositan las instrucciones debe ejecutarse con suficiente frecuencia para detectar todas las transiciones de las entradas CU o CD. Se procederá a explicar el funcionamiento de dos tipos de contadores, los cuales son, el CTU (Contador ascendente) y el CTD (Contador descendente).
- ✓ **Contador ascendente (CTU):** CTU se incrementa en uno (1) cuando el valor del parámetro CU cambia de cero (0) a uno (1). Si el valor del parámetro CV (valor de conteo actual) es mayor o igual al valor del parámetro PV (valor de conteo predeterminado), el parámetro de salida del contador Q = 1. Si el valor del parámetro de reset (R) cambia de cero (0) a uno (1), el valor de conteo actual se pone. La **Figura 10** muestra un cronograma de la instrucción CTU con una valor de conteo de entero sin signo (donde PV = 3).

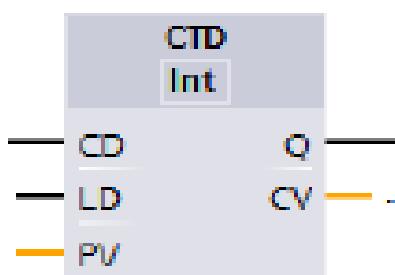


**Figura 9.** CTU (Contador ascendente).

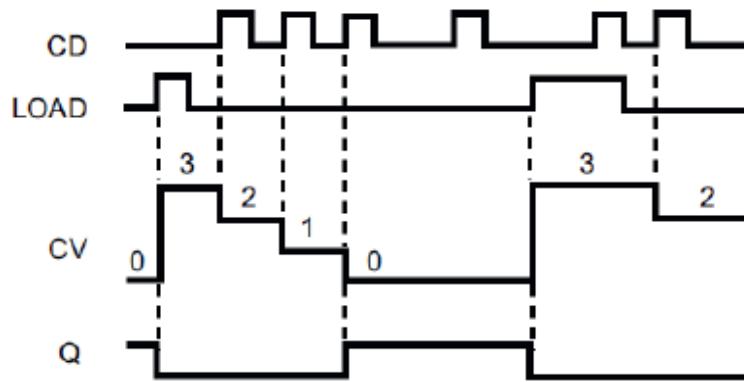


**Figura 10.** Diagrama de funcionamiento de un CTU.

- ✓ **Contador descendente (CTD):** CTD se decrementa en uno (1) cuando el valor del parámetro CD cambia de cero (0) a uno (1). Si el valor del parámetro CV (valor de contaje actual) es menor o igual a cero (0), el parámetro de salida del contador Q = 1. Si el valor del parámetro Load cambia de 0 a 1, el valor del parámetro PV (valor predeterminado) se carga en el contador como nuevo CV (valor de contaje actual). La **Figura 12** muestra un cronograma de la instrucción CTD con un valor de contaje de entero sin signo (donde PV = 3).



**Figura 11.** CTD (Contador descendente).



**Figura 12.** Diagrama de funcionamiento de un CTD.

### 3.1.4. Direccionamiento de áreas de memoria

Step 7 facilita la programación simbólica. Se crean nombres simbólicos o variables para las direcciones de los datos, ya sea como variables PLC asignada a direcciones de memoria y E/S o como variables locales utilizadas dentro de un bloque lógico. Para utilizar estas variables en el programa de usuario basta con introducir el nombre de variable para el parámetro de instrucción. Para una mejor comprensión de como la CPU estructura y direcciona las áreas de memoria, los siguientes párrafos explicarán el direccionamiento “absoluto” al que se refieren las variables PLC. La CPU ofrece varias opciones para almacenar datos durante la ejecución del programa de usuario:

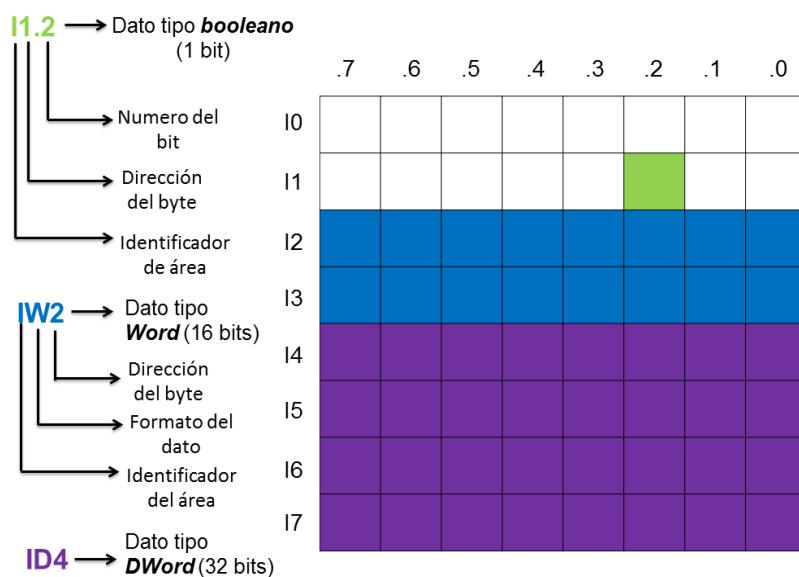
- **Memoria global:** la CPU ofrece distintas áreas de memoria, incluyendo entradas (I), salidas (Q) y marcas (M). Todos los bloques lógicos pueden acceder sin restricción alguna a esta memoria.
- **Bloque de datos (DB):** es posible incluir DBs en el programa de usuario para almacenar los datos de los bloques lógicos. Los datos almacenados se conservan cuando finaliza la ejecución del bloque lógico asociado. Un DB “global” almacena datos que pueden ser utilizados por todos los bloques lógicos,

mientras que un DB de instancia almacena datos para un bloque de función (FB) específico y está estructurado según los parámetros del FB.

- **Memoria temporal:** cada vez que se llama un bloque lógico, el sistema operativo de la CPU asigna la memoria temporal o local (L) que debe utilizarse durante la ejecución del bloque. Cuando finaliza la ejecución del bloque lógico, la CPU reasigna la memoria local para la ejecución de otros bloques lógicos.

Toda posición de memoria diferente tiene una dirección unívoca. El programa de usuario utiliza estas direcciones para acceder a la información de la posición de memoria.

- **Tipos de Datos:** los tipos de datos se utilizan para determinar el tamaño de un elemento de datos y como deben interpretarse los datos. Todo parámetro de instrucción soporta como mínimo un tipo de datos. Algunos parámetros soportan varios tipos de datos. En la **Figura 13** se muestra un esquema de memoria y direccionamientos, y en la **Tabla 2** pueden evidenciarse los tipos de datos.



**Figura 13.** Esquema de memoria y direccionamientos.

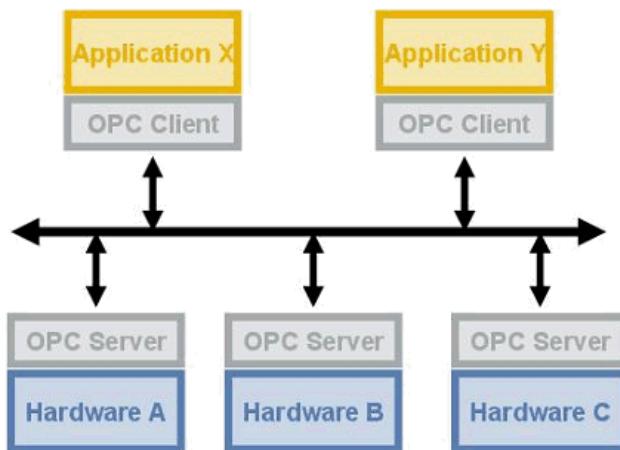
**Tabla 2.** Tipos de datos.

Tipos de datos	Descripción
Tipos de datos de bits y secuencias de bits	<ul style="list-style-type: none"> <li>Bool es un valor de bit o booleano.</li> <li>Byte es un valor de byte (8 bits).</li> <li>Word es un valor de 16 bits.</li> <li>DWord es un valor de palabra doble (32 bits).</li> </ul>
Tipos de datos enteros	<ul style="list-style-type: none"> <li>USInt (entero sin signo de 8 bits) y SInt (entero con signo de 8 bits) son enteros "cortos" (8 bits o 1 byte de memoria) que pueden tener o no signo.</li> <li>UInt (entero sin signo de 16 bits) e Int (entero con signo de 16 bits) son enteros (16 bits o 1 palabra de memoria) que pueden tener o no signo.</li> <li>UDInt (entero de 32 bits sin signo) y DInt (entero de 32 bits con signo) son enteros dobles (32 bits o 1 palabra doble de memoria) que pueden tener o no signo.</li> </ul>
Tipos de datos de números reales	<ul style="list-style-type: none"> <li>Real es un valor de número real de 32 bits o en coma flotante.</li> <li>LReal es un valor de número real de 64 bits o en coma flotante.</li> </ul>
Tipos de datos de fecha y hora	<ul style="list-style-type: none"> <li>Date es un valor de fecha de 16 bits (similar a un valor UInt) que contiene el número de días desde el 1 de enero de 1900. El valor de fecha máximo es 65535 (16#FFFF), que corresponde al 6 de junio de 2169. Todos los posibles valores de Date son válidos.</li> <li>DTL (Date and Time Long) es una estructura de 12 bytes que almacena información de fecha y hora en una estructura predefinida. <ul style="list-style-type: none"> <li>Año (UInt): de 1970 a 2554</li> <li>Mes (USInt): de 1 a 12</li> <li>Día de la semana (USInt): de 1 (domingo) a 7 (sábado)</li> <li>Horas (USInt): de 0 a 23</li> <li>Minutos (USInt): de 0 a 59</li> <li>Segundos (USInt): de 0 a 59</li> <li>Nanosegundos (UDInt): de 0 a 999999999</li> </ul> </li> <li>Time es un valor de tiempo CEI de 32 bits (parecido al valor Dint) que almacena el número de milisegundos (de 0 a 24 días, 20 horas, 31 minutos, 23 segundos y 647 ms). Todos los posibles valores de Time son válidos. Los valores de Time se pueden usar para cálculos, y se pueden obtener tiempos negativos.</li> <li>TOD (Time of Day) es un valor de hora de 32 bits (parecido al valor Dint) que contiene el número de milisegundos desde medianoche (de 0 a 86399999).</li> </ul>
Tipos de datos de caracteres y cadenas	<ul style="list-style-type: none"> <li>Char es un carácter simple de 8 bits.</li> <li>String es una cadena de longitud variable de hasta 254 caracteres.</li> </ul>

### 3.2. Servidor OPC

Un servidor OPC es una aplicación de software (*driver*) que cumple con una o más especificaciones definidas por la OPC Foundation. El servidor OPC hace una interfaz comunicando por un lado con una o más fuentes de datos utilizando sus protocolos

nativos (típicamente PLCs, DCSs, basculas, Módulos I/O, controladores, etc.) y por otro lado con clientes OPC (típicamente SCADAs, HMIs, generadores de informes, generadores de gráficos, aplicaciones de cálculos, etc.). En una arquitectura Cliente OPC/Servidor OPC, el servidor OPC es el esclavo mientras que el cliente OPC es el maestro. Las comunicaciones entre cliente OPC y el servidor OPC son bidireccionales, lo que significa que los clientes pueden leer y escribir en los dispositivos a través del servidor OPC.



**Figura 14.** Esquema de funcionamiento de un servidor OPC.

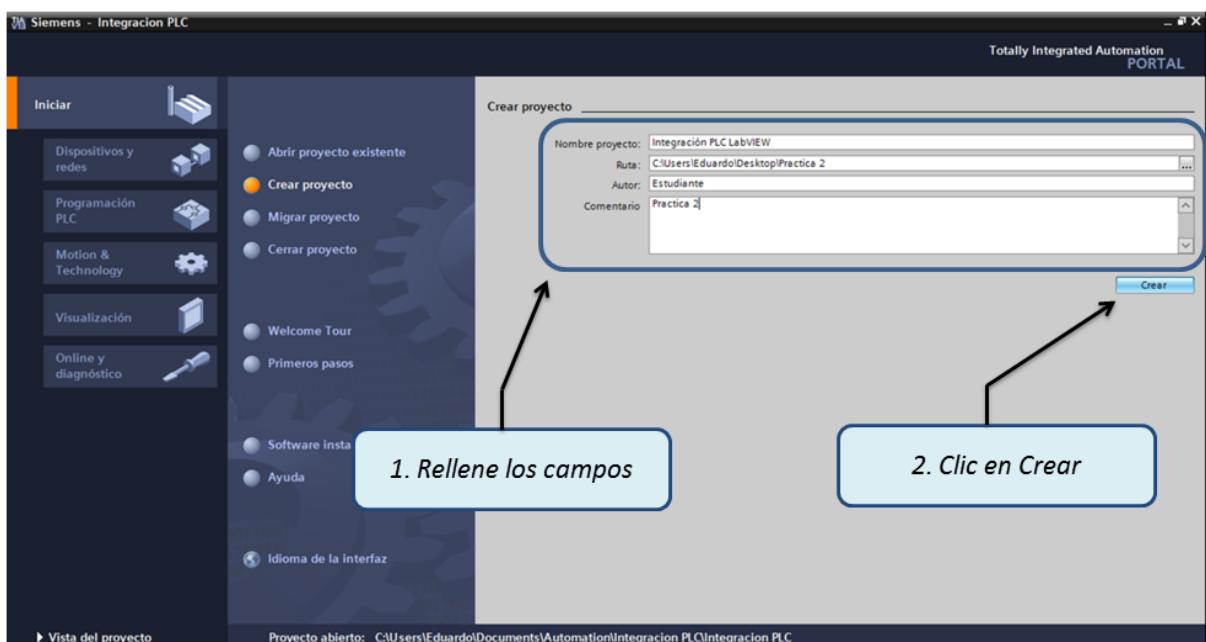
### 3.3. Interfaz Humano Maquina (HMI)

HMI significa *Human Machine Interface* (Interfaz Humano Maquina), es decir, es el dispositivo o sistema que permite la interfaz entre el humano y la máquina. Tradicionalmente estos sistemas consistían en paneles compuestos por indicadores y comandos, tales como luces pilotos indicadores digitales y análogos, registradores, pulsadores, selectores y otros que se interconectaban con la máquina o proceso. En la actualidad, dado que las máquinas y procesos en general están implementadas con controladores y otros dispositivos electrónicos que dejan disponibles puertas de comunicación, es posible contar con sistemas HMI más poderosos y eficaces, además de permitir una conexión más sencilla y económica con el proceso o maquinas.

## 4. PROCEDIMIENTO

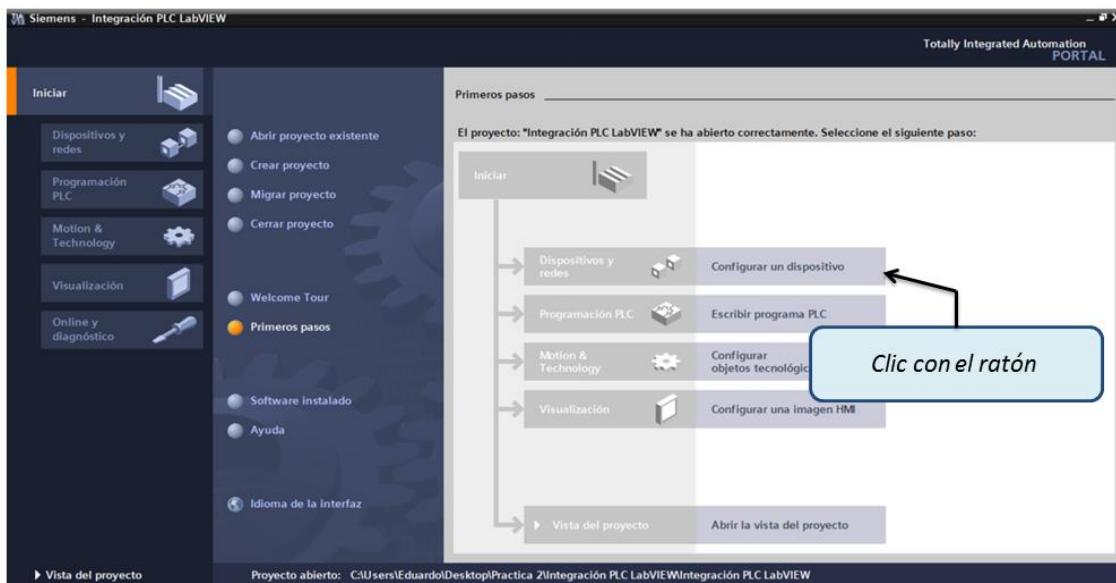
### 4.1. Creación de un proyecto en TIA PORTAL

Ejecute el TIA PORTAL, hecho esto se abrirá lo que se conoce como *Vista del Portal*, por defecto aparecerá seleccionada la sub ventana *Iniciar*, esta puede gestionar la apertura, creación y migración de proyectos entre otras funciones en esta interfaz de programación. Seleccione *Crear proyecto* y rellene los campos requeridos tal como se muestra en la **Figura 15**. Rellenados los campos presione el botón *Crear*.

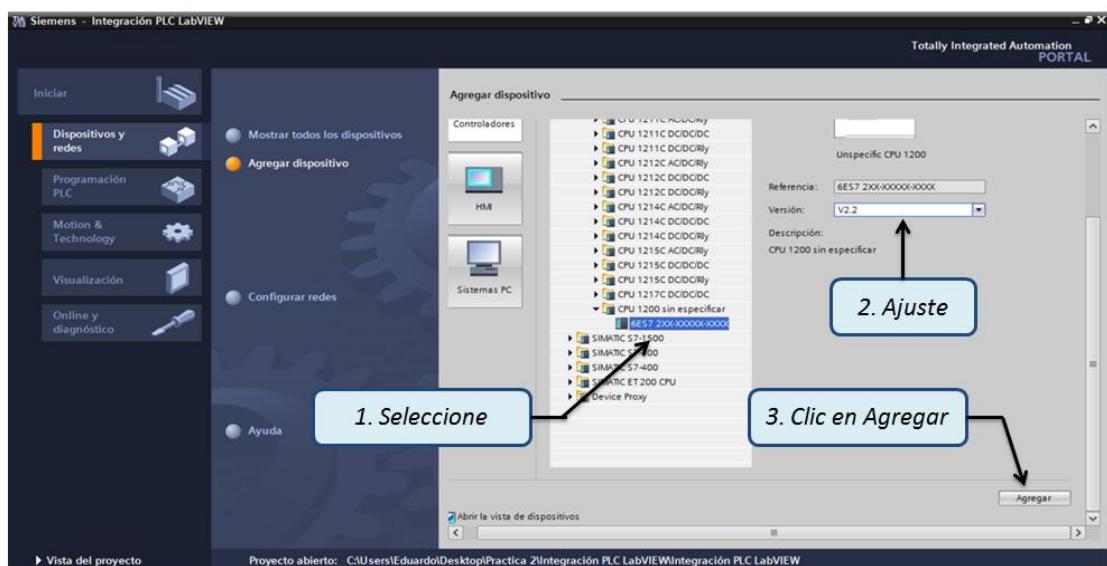


**Figura 15.** Vista del portal ventana *Iniciar*.

Creado el proyecto será direccionado al ítem *Primeros pasos* (dentro de la misma sub ventana *Iniciar*), seleccione la opción *Configurar un dispositivo*, tal como es mostrado en la **Figura 16**. Posteriormente será direccionado a la sub ventana *Dispositivos y redes* donde tendrá la posibilidad de agregar la CPU del PLC, seleccione la *CPU sin especificar / 6ES7 2XX-XXXXX-XXXX*, con versión “V2.2” y pulse *Agregar*, ver **Figura 17**.



**Figura 16.** Vista del portal *Primeros pasos*.

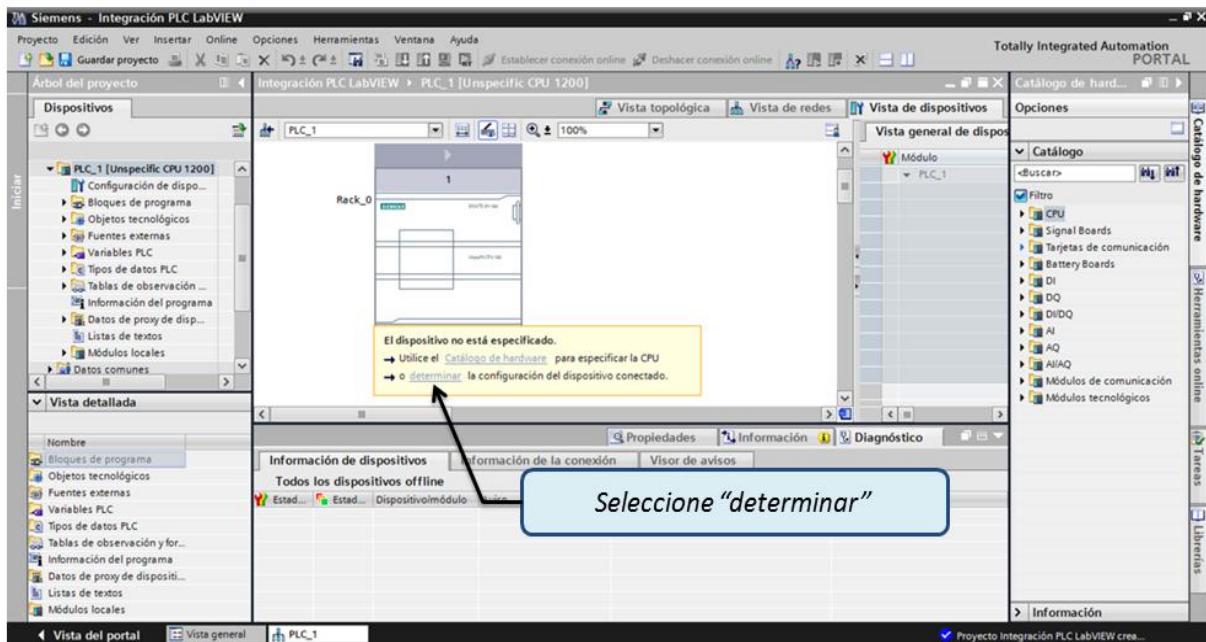


**Figura 17.** Agregar dispositivo (CPU 1200 sin especificar).

Agregada la CPU (sin especificar) del PLC aparecerá lo que se denomina *Vista del proyecto*, dentro se encuentran todas las herramientas para la configuración y programación del PLC.

Encienda y conecte el controlador al ordenador mediante el Cable UTP RJ45/RJ45.

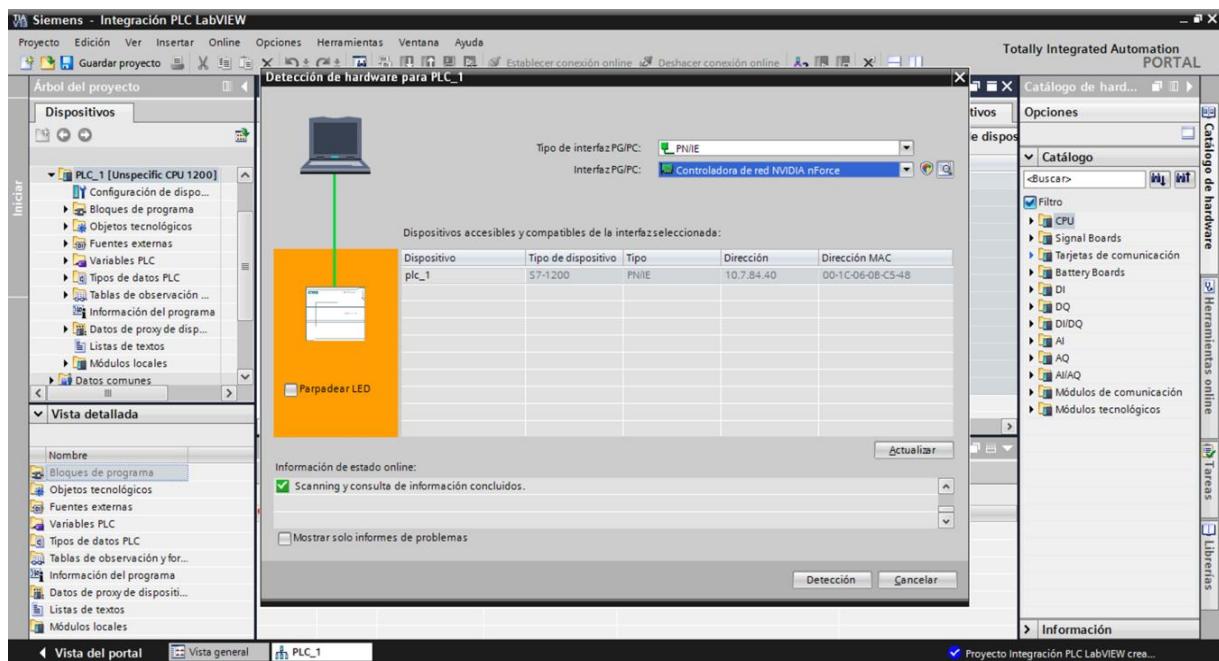
Como es mostrado en la **Figura 18**, seleccione dentro del “recuadro amarillo” *El dispositivo no está especificado*, la opción **determinar**.



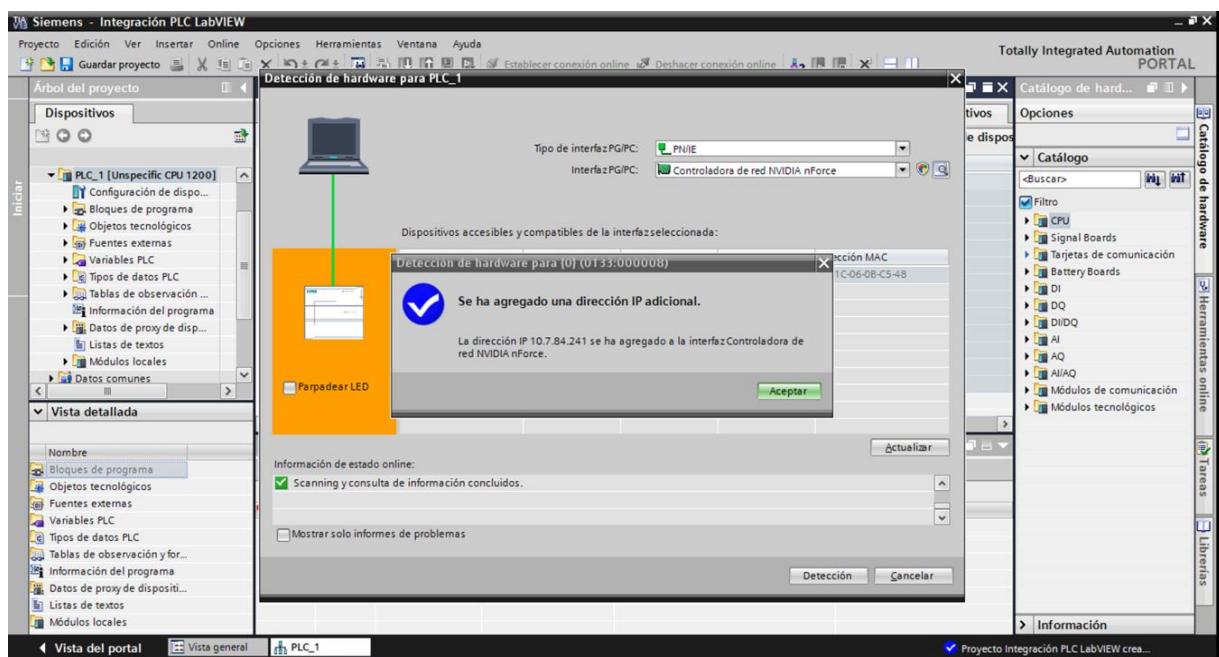
**Figura 18.** Vista del proyecto, selección de la opción **determinar**, para posterior detección.

Seguidamente se desplegará una ventana llamada *Detección de hardware* para *PLC\_1*, configure las opciones *Tipo de interfaz PG/PC*: en *PN/IE* y para la *Interfaz PG/PC* seleccione su adaptador de red, después como alternativa al usuario, pulse la opción *Parpadear LED*, de esta manera se comprobará la conexión entre ordenador y controlador, por último seleccione el botón *Detección*, véase las **Figuras 19 y 20**.

En la **Figura 20** es mostrado como se agrega una dirección IP adicional (ordenador), la cual en si permite realizar la “integración” entre el ordenador y el PLC.

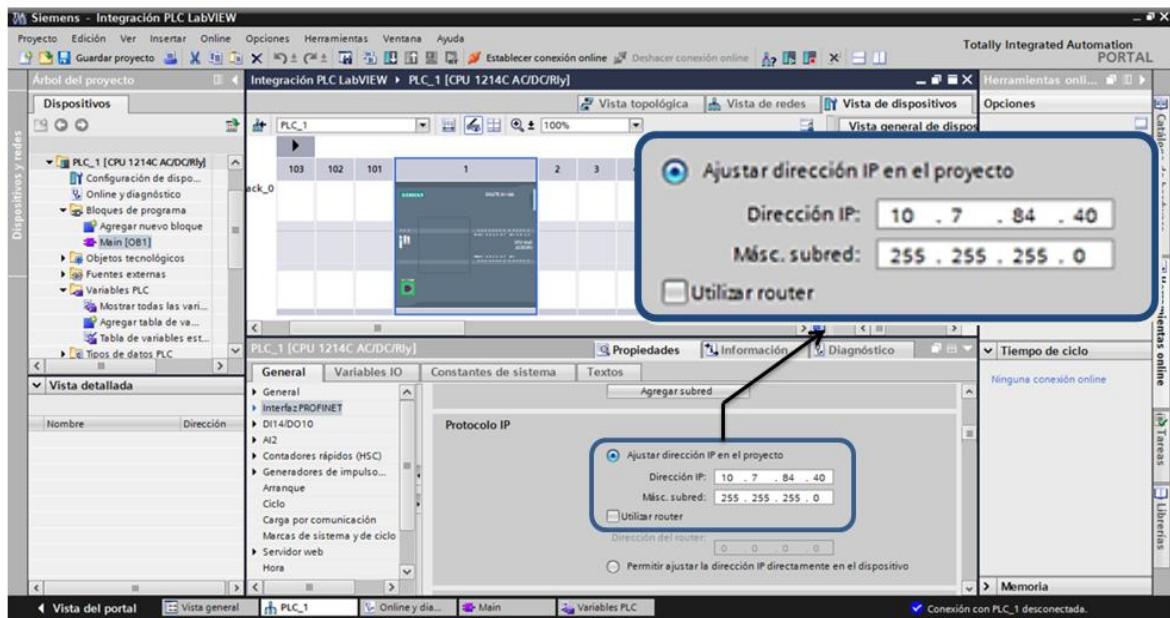


**Figura 19.** Ventana Detección de Hardware.



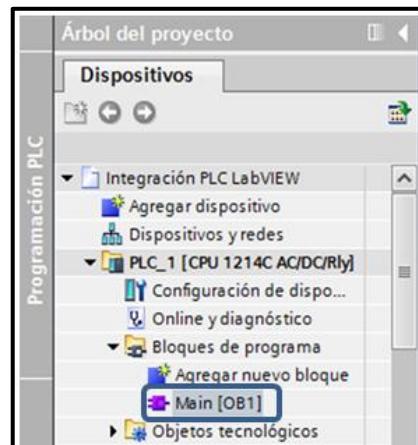
**Figura 20.** Detección finalizada.

Detectado el controlador es posible realizar cualquier ajuste y programación en él. En la **Figura 21** aparece el controlador detectado y puede observarse la dirección IP de este (**10.7.84.40**).



**Figura 21.** Vista del proyecto con el PLC SIMATIC S7-1200 detectado.

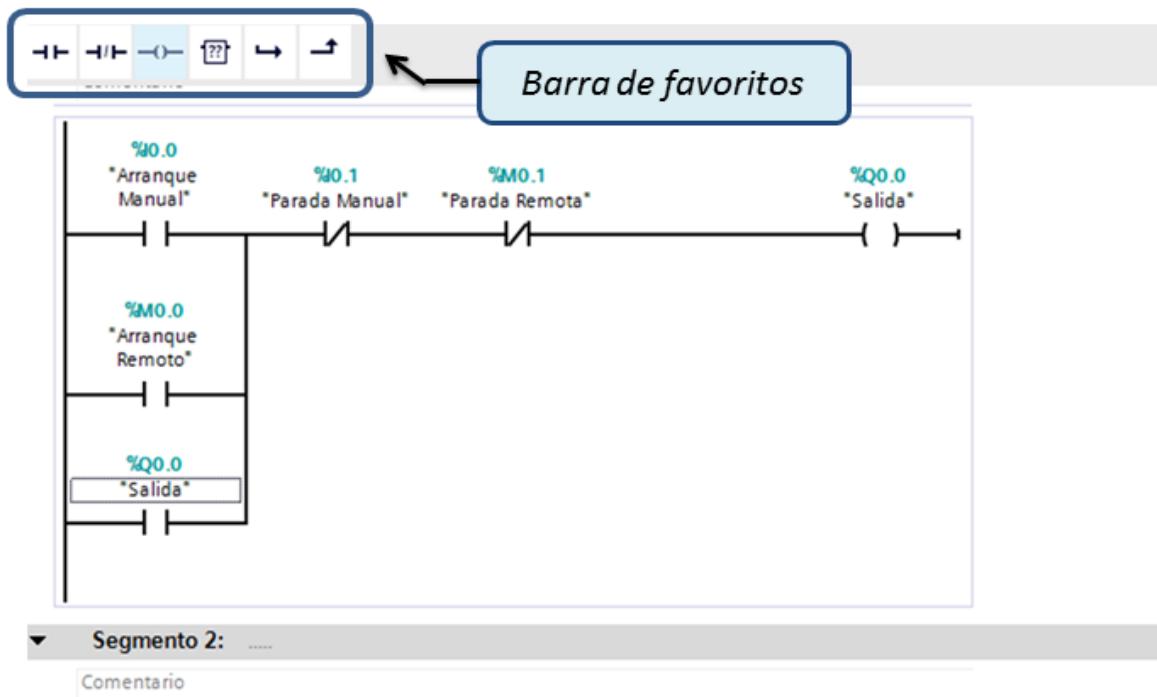
Luego deberá programar el controlador, para ello diríjase al *Árbol del proyecto* / *Dispositivos* / *PLC\_1* / *Bloques de programa* / *Main [OB1]*, ver **Figura 22**.



**Figura 22.** Selección del *Main [OB1]*.

Abierto el *Main [OB1]* aparecerá el primer segmento para la programación en lenguaje de esquema de contactos (KOP). Haciendo uso de los contactos definidos en la *Barra de favoritos*, repita la lógica visualizada en la **Figura 23**. Para cada contacto agregado a la lógica, es posible modificar sus características tales como nombre y

direccionalamiento accediendo a *Árbol del proyecto / Dispositivos / PLC\_1 / Variables PLC.*



**Figura 23.** Primer Programa creado en lenguaje KOP.

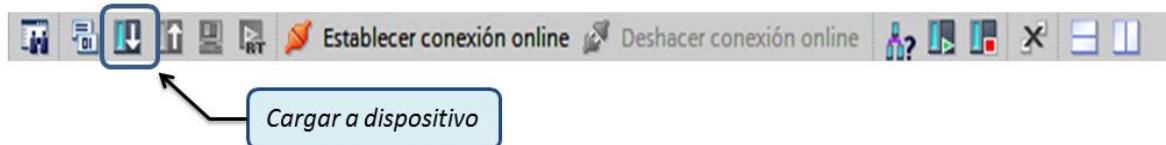
El programa creado (ver **Figura 23**) corresponde a la activación y desactivación de una salida física del PLC implementado auto enganche o auto enclavamiento (debido al uso de pulsadores), la activación (arranque) se encuentra programada para ejecutarse de forma manual o lógica (remota) y de igual manera para la desactivación (parada), el hecho de esta programación es para aportar seguridad, flexibilidad y maniobra a la lógica en el uso posterior de una HMI.

Seleccione *Establecer conexión online*, ver **Figura 24**, de este modo se establecerá conexión online con el PLC.



**Figura 24.** Establecer conexión online con el PLC.

Posteriormente, seleccione el botón de *Cargar a dispositivo* para descargar el programa en el PLC, ver **Figura 25**.



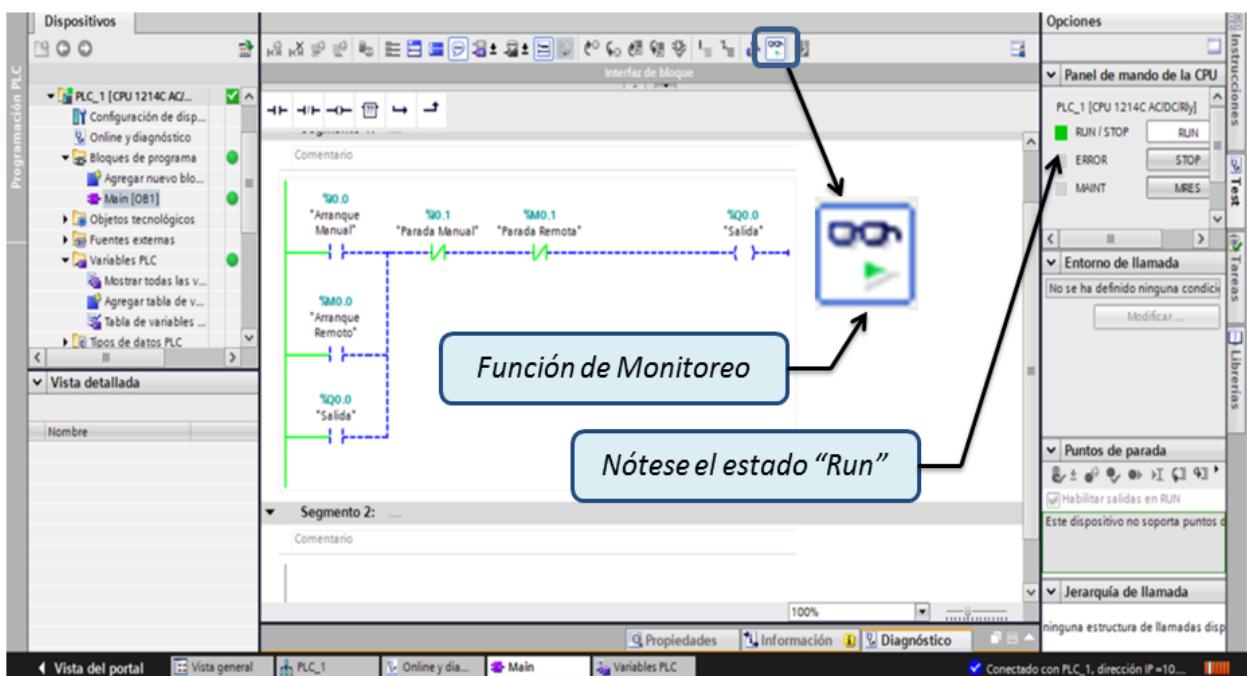
**Figura 25.** Descargar el programa al PLC.

EL PLC tiene dos modos de operación *Run* y *Stop*, en modo *Run* el PLC podrá ejecutar el programa que ha sido descargado en él (La CPU se enciende), en modo *Stop* el PLC se mantiene sin ejecutar alguna acción (La CPU está apagada), conociendo esto y ya descargado el programa en el PLC, pulse el botón *Run* para que el PLC pueda ejecutar la lógica descargada en él, véase **Figura 26**.



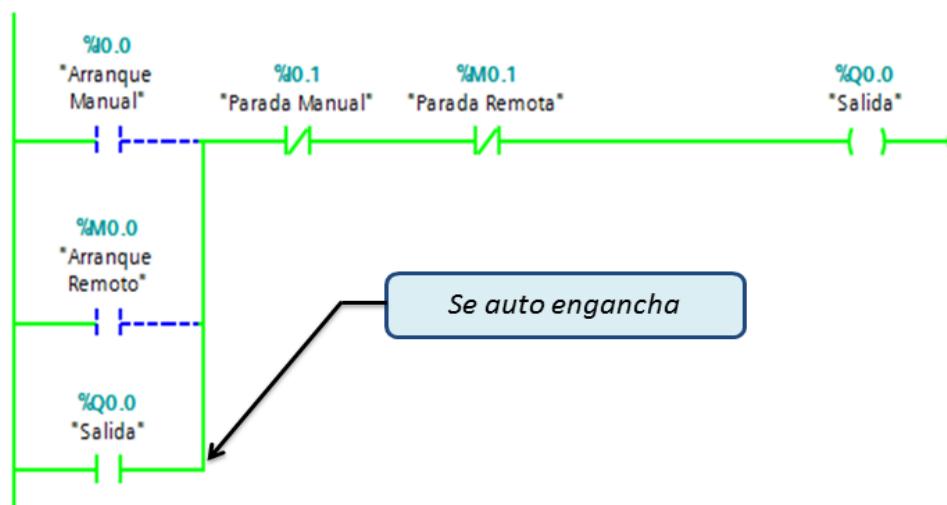
**Figura 26.** Selección del botón del modo *Run*.

Es posible monitorear la ejecución y funcionamiento del programa, para esto seleccione el botón de *Monitoreo* como es mostrado en la **Figura 27**.



**Figura 27.** Monitoreo del programa descargado al PLC modo *Run* con *Salida* inactiva.

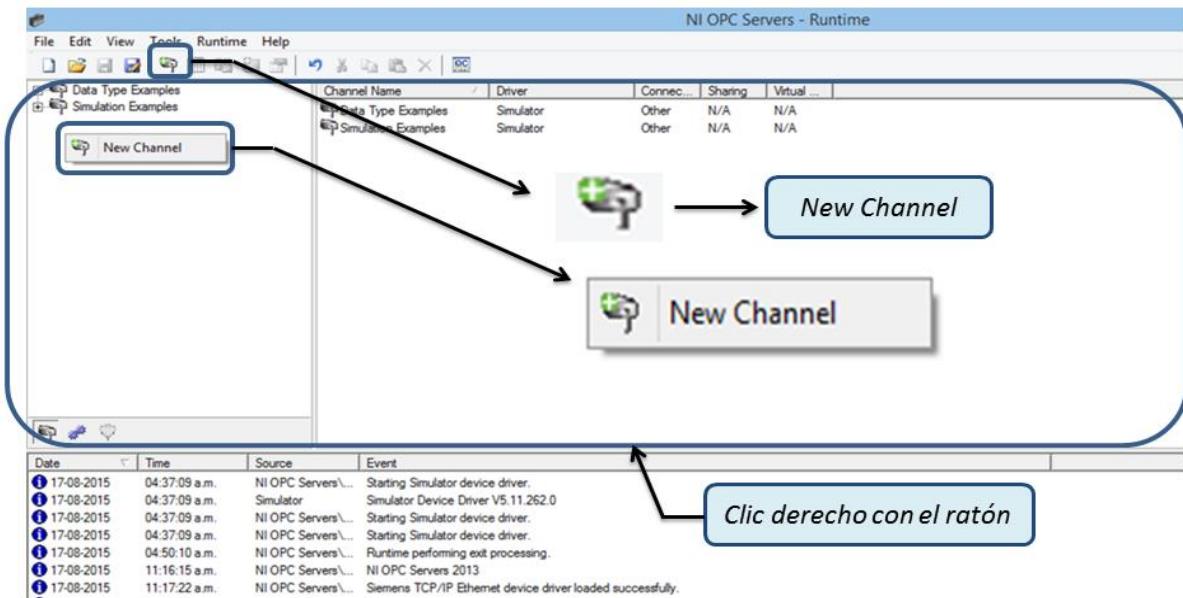
Active alguno de los *Arranques* (manual o remoto) para observar el cambio de estado de la *Salida* (Q0.0), ver **Figura 28**.



**Figura 28.** Monitoreo del programa descargado al PLC modo *Run* con *Salida* activa.

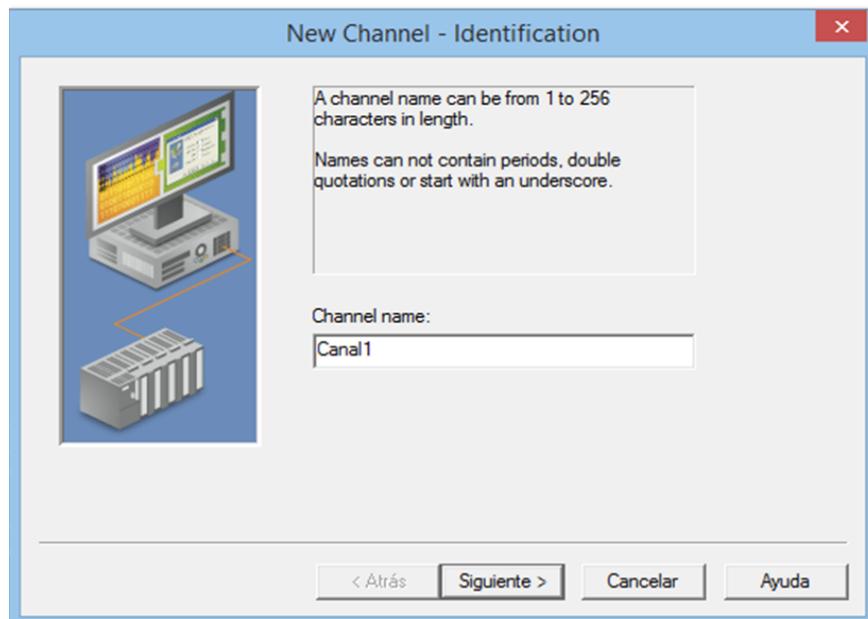
## 4.2. Configuración del servidor OPC

Ejecute el NI OPC Server, se abrirá una ventana principal como es mostrado en la **Figura 29**. Seleccione el botón *New Channel* o haga clic derecho con el ratón sobre cualquier lugar dentro del recuadro especificado en la **Figura 29**, hecho esto se desplegará una pestaña llamada *New Channel* (Nuevo Canal) que permitirá configurar el canal de comunicación (driver, adaptador de red, etc.) entre el PLC y el Servidor OPC.



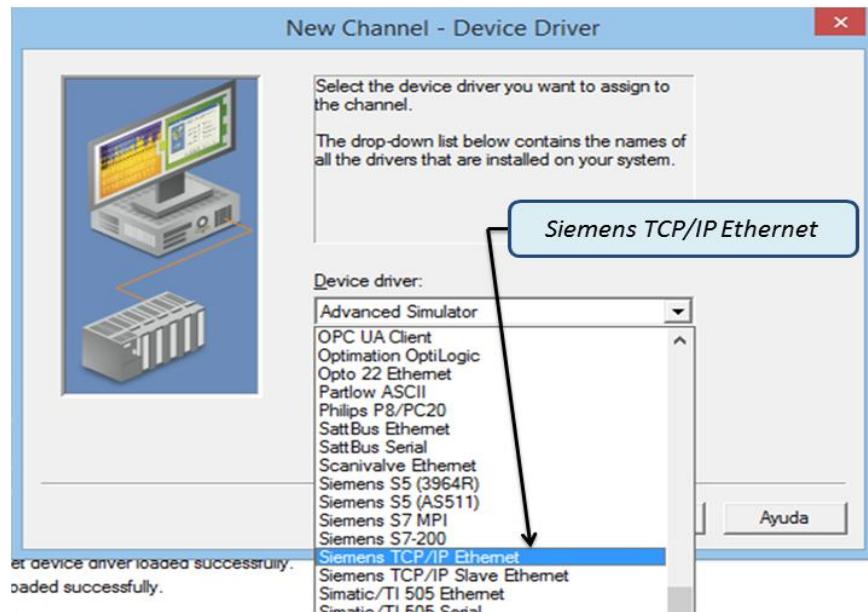
**Figura 29.** NI OPC Server.

Configure el *New Channel* como es mostrado desde la **Figura 30** hasta la **35**. Primero que todo se le pedirá insertar la identificación del canal (**Figura 30**).



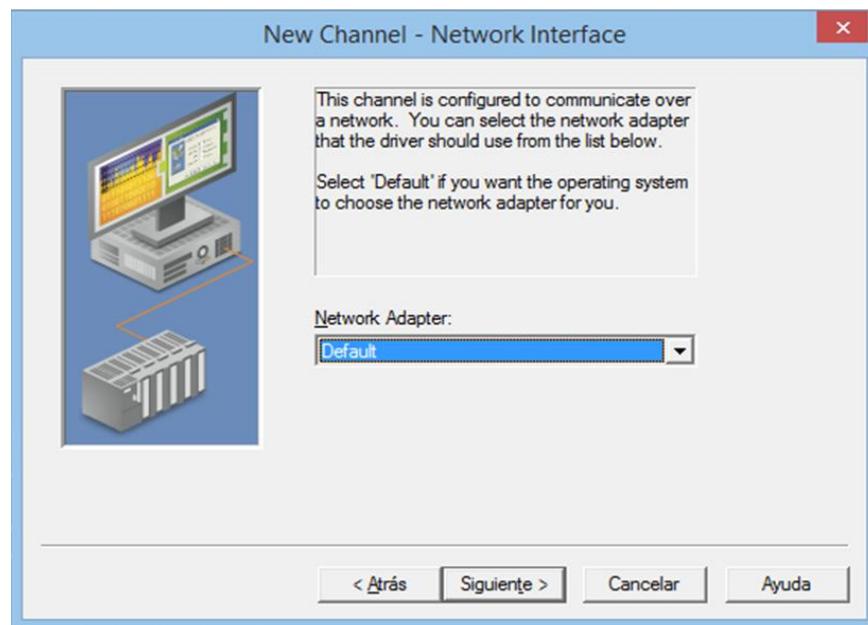
**Figura 30.** Ventana de identificación del canal (*Identification*).

Luego seleccione *Siemens TCP/IP Ethernet* en *Device Driver* (**Figura 31**).



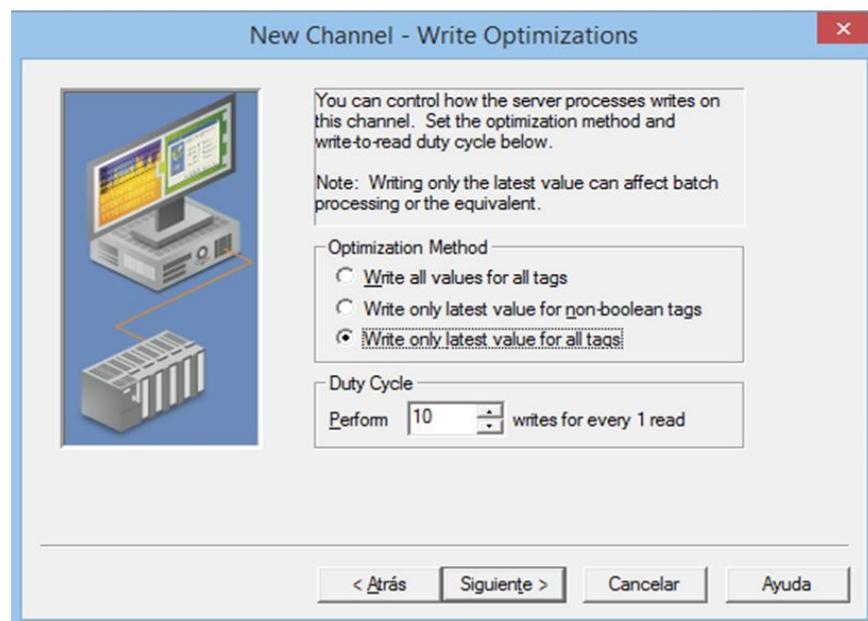
**Figura 31.** Ventana de selección del driver correspondiente al PLC (*Device Driver*).

Elija el adaptador de red, en este caso *Default* (**Figura 32**).



**Figura 32.** Ventana de selección del adaptador de red del ordenador (*Network Interface*).

En la ventana *Write Optimizations* (**Figura 33**) establezca los valores por defecto y pulse *Siguiente*, haga lo mismo con la ventana *Non-Normalized Float Handling* (**Figura 34**).

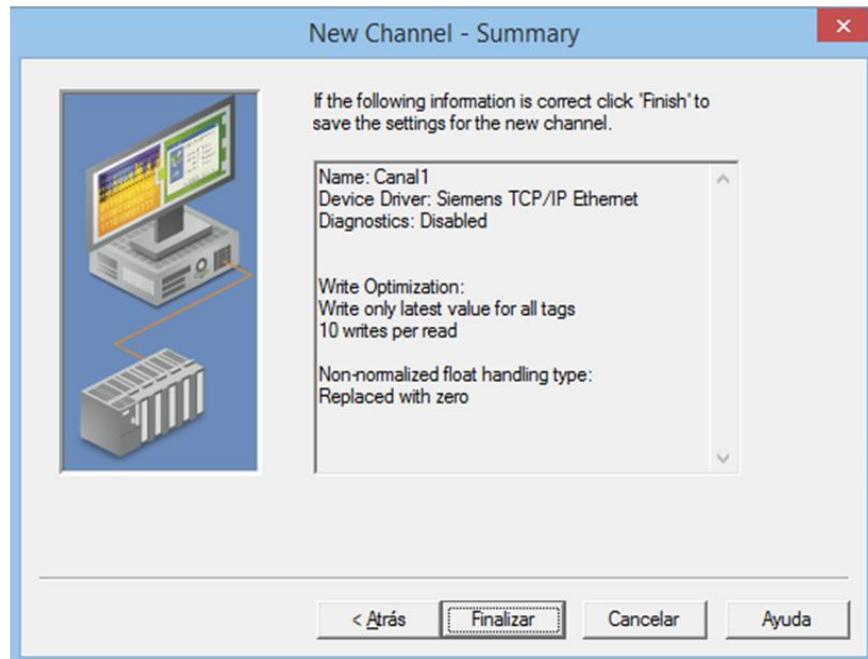


**Figura 33.** Ventana *Write Optimizations*.



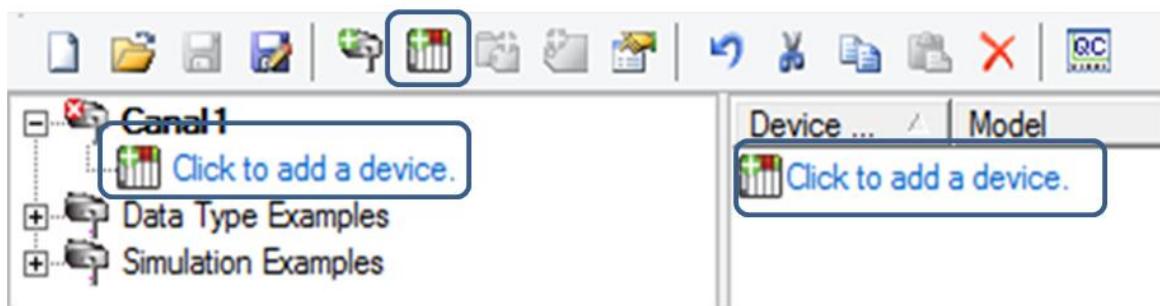
**Figura 34.** Ventana Non-Normalized Float Handling.

Por ultimo aparecerá un resumen de todas las configuraciones hechas previamente (**Figura 35**).



**Figura 35.** Ventana de resumen de configuraciones (Summary).

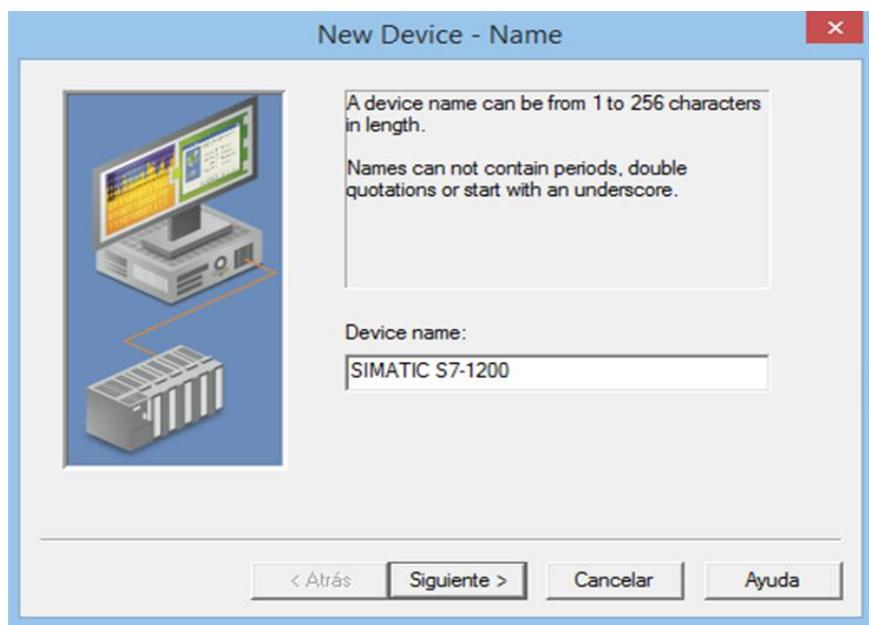
Finalizada la configuración del *Canal 1* se agregará el dispositivo (PLC SIMATIC S7-1200) dentro del canal, por lo tanto, seleccione la opción *Click to add a device*, como es mostrado en la **Figura 36**.



**Figura 36.** Añadir dispositivo (PLC).

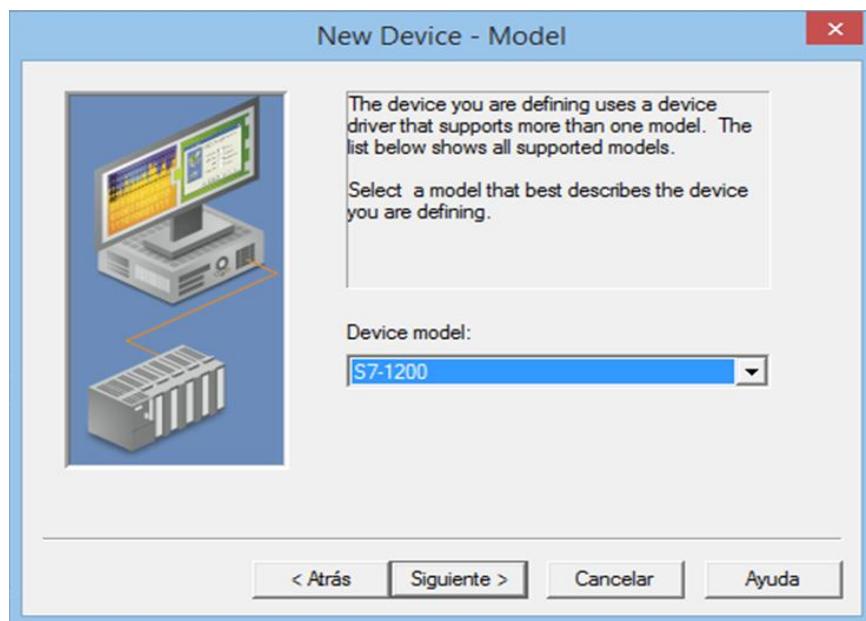
Seguidamente deberá configurar el controlador como se muestra desde la **Figura 37** hasta la **Figura 46**.

En primer lugar, asigne el nombre al nuevo dispositivo (**Figura 37**).



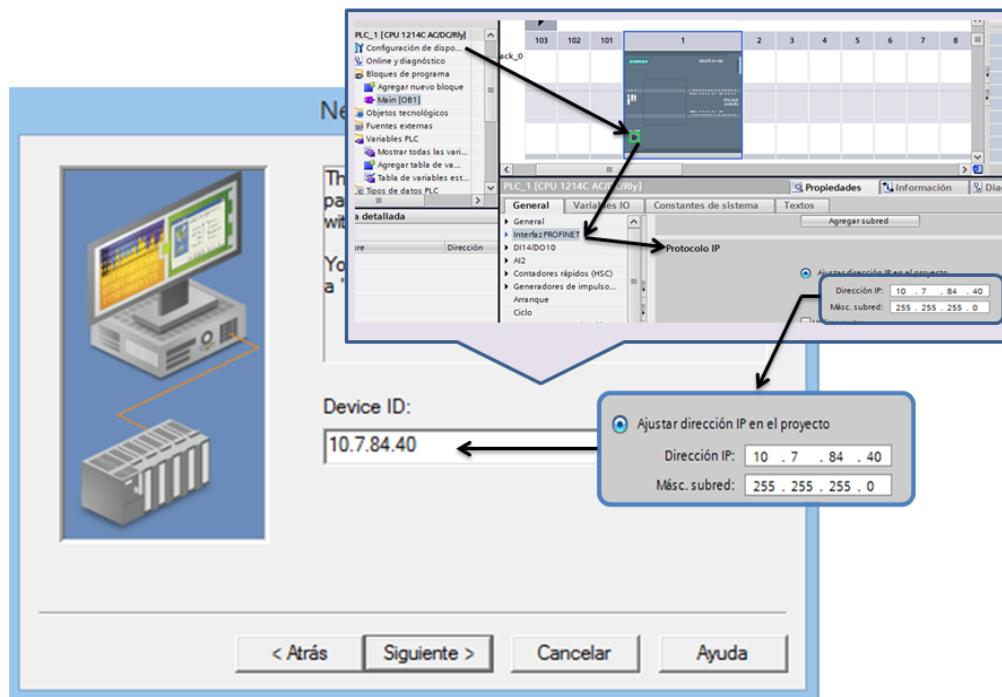
**Figura 37.** Ventana de asignación del nombre del controlador (*Name*).

Luego elija el modelo del dispositivo (*S7-1200*), ver **Figura 38**.



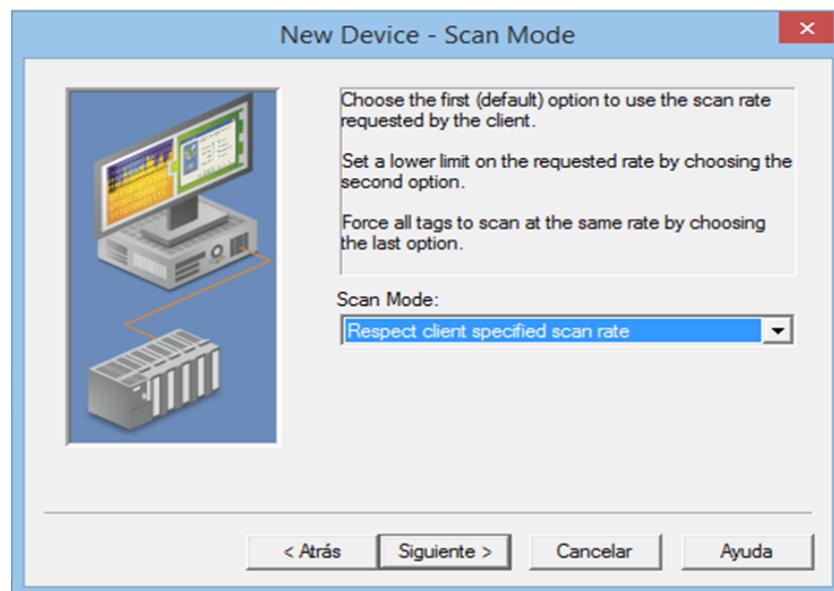
**Figura 38.** Ventana de selección del modelo de controlador (*Model*).

En la ventana *ID* (**Figura 39**) rellene el campo con la dirección IP del PLC, puede acceder a esta desde el TIA PORTAL con la siguiente ruta: *Configuración de dispositivo / Interfaz PROFINET / Protocolo IP*.



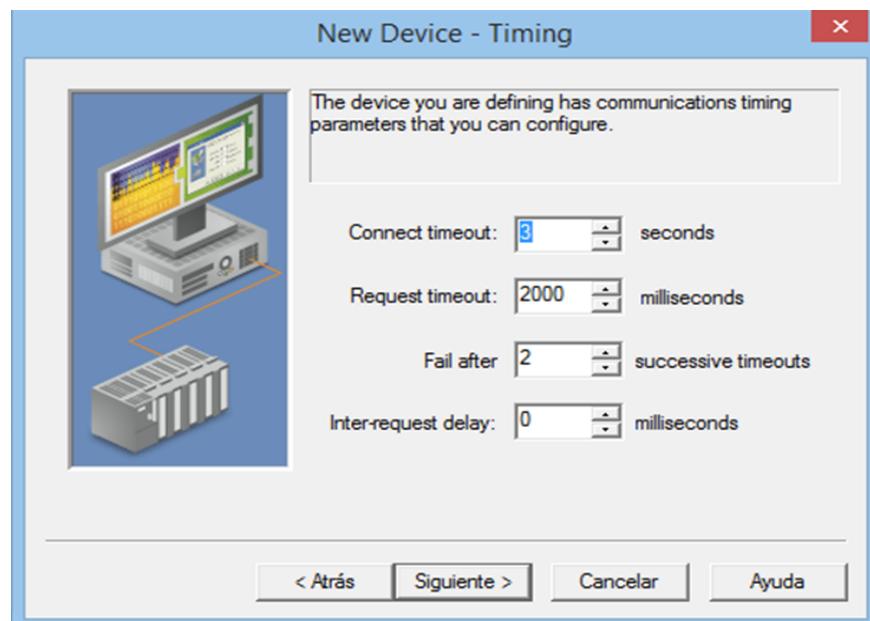
**Figura 39.** Ventana de identificación del dispositivo en la red (ID).

Después, en la ventana *Scan Mode* establezca por defecto la selección *Respect client specified scan rate* (**Figura 40**).



**Figura 40.** Ventana de configuración de barrido de datos (*Scan Mode*).

En *Timing* deje por defecto los valores ya especificados, ver **Figura 41**.



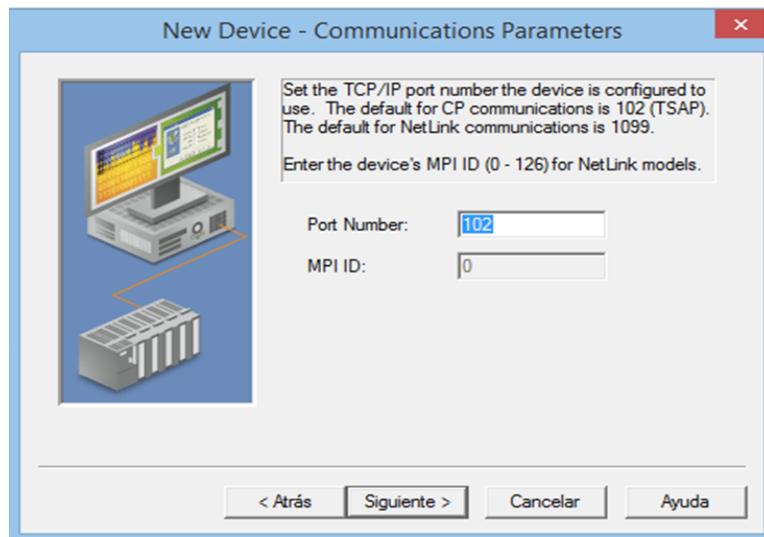
**Figura 41.** Ventana parámetros de temporización (*Timing*).

La ventana *Auto-Demotion* quedará configurada por defecto, ver **Figura 42**.



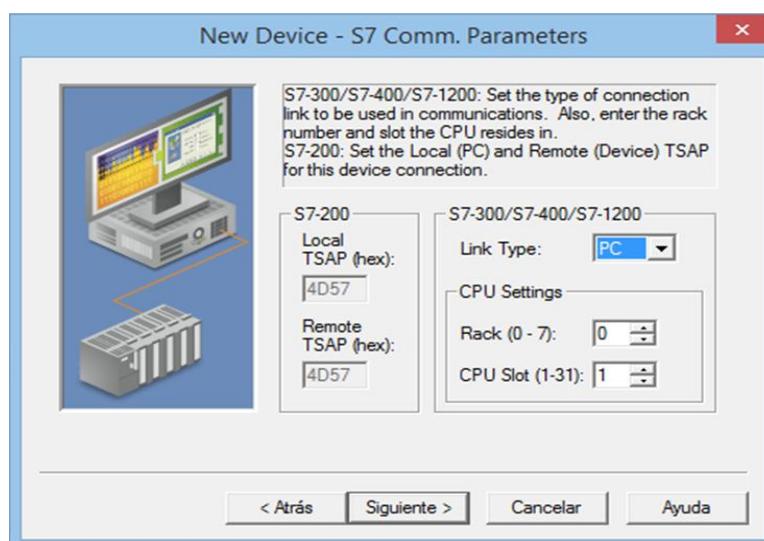
**Figura 42.** Ventana *Auto-Demotion*.

Posteriormente, en *Communications Parameters* establezca por defecto el valor ya especificado para el número del puerto (102), ver **Figura 43**.



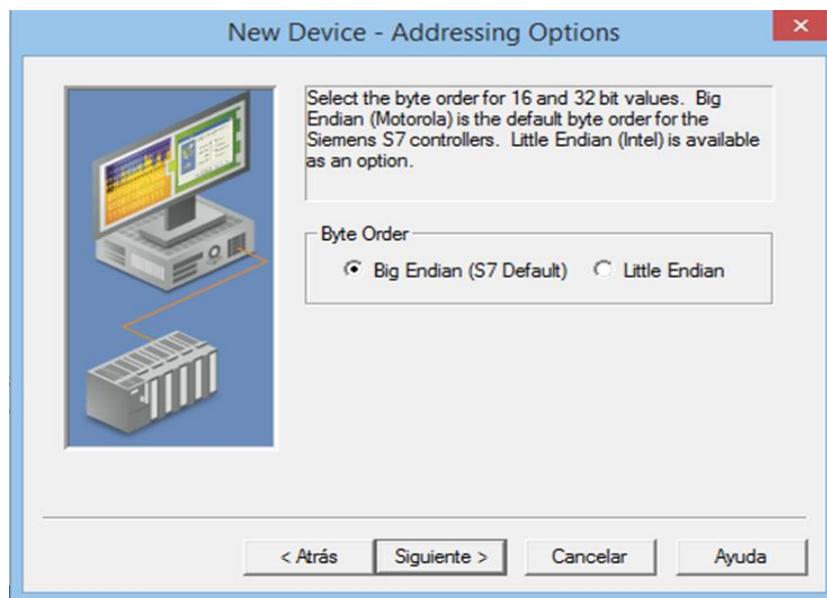
**Figura 43.** Ventana de parámetros de comunicaciones (*Communications Parameters*).

En S7 *Communications Parameters* las configuraciones que se establecerán irán por defecto (**Figura 44**).



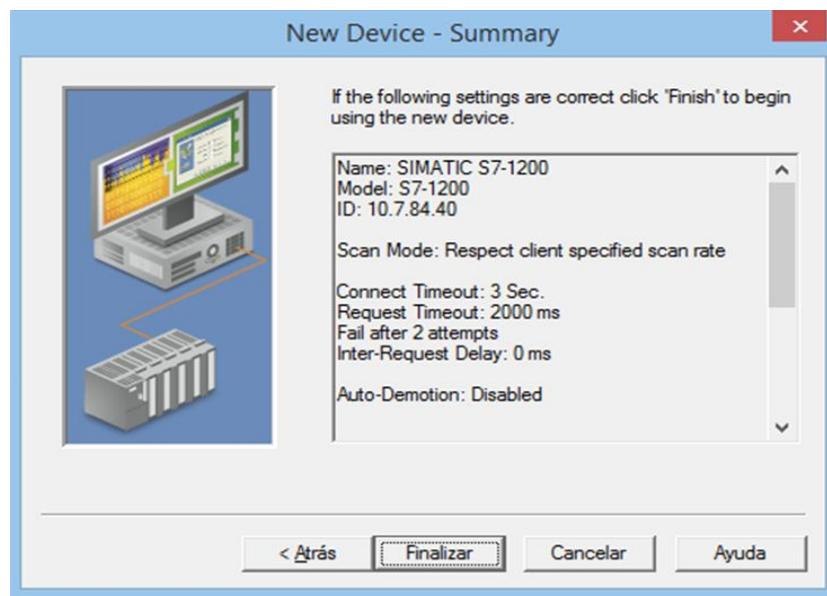
**Figura 44.** Ventana de parámetros de configuración de controladores Siemens (S7 *Communications Parameters*).

*Addressing Options* quedará configurada por defecto, ver **Figura 45.**



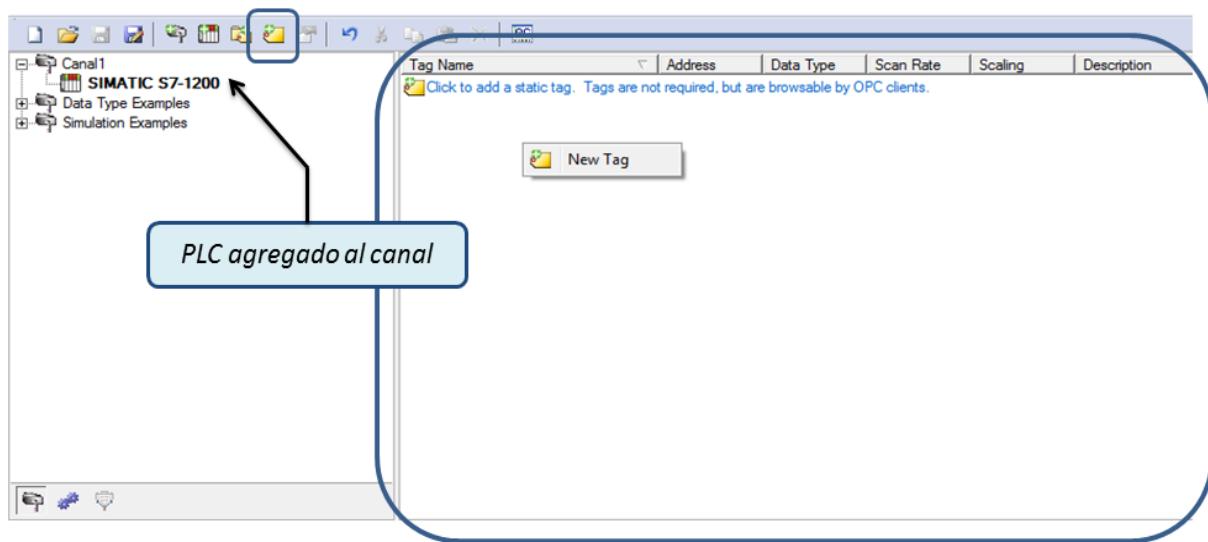
**Figura 45.** Ventana de opciones de direccionamiento (*Addressing Options*).

Por último, aparecerá un resumen de todas las configuraciones hechas previamente (**Figura 35**).



**Figura 46.** Ventana de resumen (*Summary*).

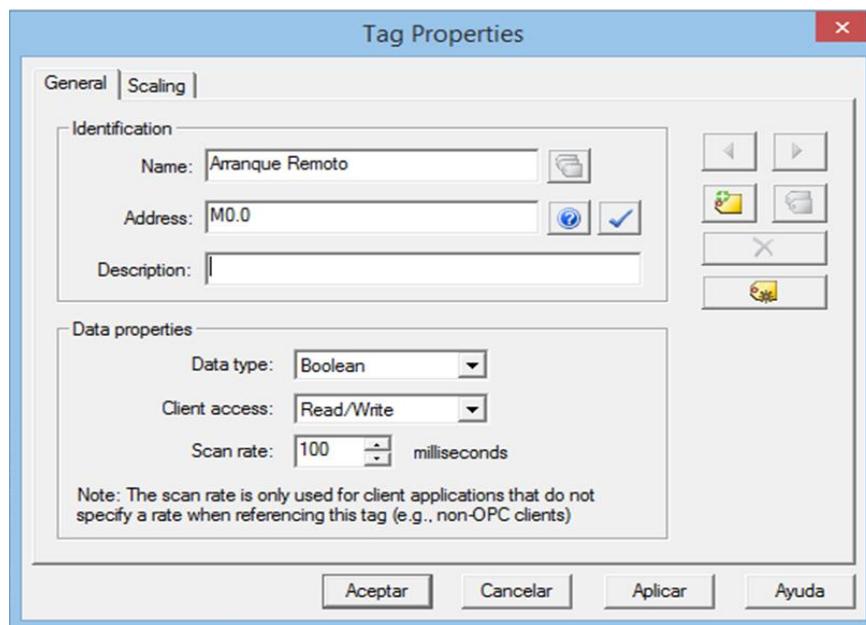
Agregado con éxito el PLC al canal se añadirán los *tags* relacionados al programa que se encuentra cargado en el PLC, por consiguiente, pulse el botón *New Tag* o haga clic derecho con el ratón sobre el área especificada por el recuadro de la **Figura 47** para acceder a la misma opción.



**Figura 47.** Añadir nuevo tag.

Cabe destacar que solo se agregarán tres (3) *tags*: *Arranque Remoto*, *Parada Remota* y *Salida*.

El primer *tag* en ser añadido será el que se encuentra direccionado a M0.0, llamado *Arranque Remoto*; rellene los campos solicitados por la ventana *Tag Properties* (propiedades del *tag*) como se visualiza en la **Figura 48**, una vez llenado los campos *Name* y *Address* es recomendable verificar el tipo de dato del *tag*, para realizar esto pulse el botón *Check Addres* y de esta manera se comprobará automáticamente. Los campos *Client access* y *Scan rate* quedarán configurados por defecto.



**Figura 48.** Ventana *Tag Properties* (Propiedades del Tag).

**Nota:** el campo *Client Access* es de suma importancia que este configurado en **Read / Write** para que se pueda leer y escribir el valor del dato desde un cliente del servidor OPC, como lo será LabVIEW en el posterior desarrollo de esta práctica. El *Scan rate* se trabajará en 100 ms para todos los *tags*.

De la misma manera que se agregó el *tag Arranque Remoto* se añadirán los demás *tags*.

Agregados los *tags* quedarán asignados como es mostrado en la **Figura 49**.

Tag Name	Address	Data Type	Scan Rate	Scaling
Arranque Remoto	M0.0	Boolean	100	None
Parada Remota	M0.1	Boolean	100	None
Salida	Q0.0	Boolean	100	None

**Figura 49.** Tags creados.

EL PLC SIMATIC S7-1200 desde el proyecto creado en TIA PORTAL había sido colocado en modo de operación *Run* (de no ser así, coloque el PLC en *Run*) con la finalidad de comprobar la correcta comunicación entre el servidor OPC y el PLC.

Para corroborar la correcta comunicación (lectura de datos) entre el servidor OPC y el PLC utilice la herramienta *Quick Client* presente en el mismo servidor OPC, véase la **Figuras 50 y 51**.



**Figura 50.** Botón de herramienta de monitoreo *Quick Client*.



**Figura 51.** *Quick Client* ejecutado y monitoreando los *tags* creados.

En la **Figura 51** desde el ítem *Quality* puede observarse la correcta comunicación (*Good*) entre el servidor OPC y PLC, además vea que aparecen los valores (*Value*) de los estados (Booleanos) de cada *tag*, en este caso se encuentran en cero (0) debido a que no ha sido activada la marca *Arranque Remoto* por lo que la salida no ha cambiado de estado.

Activando y desactivando (simulando la acción mecánica de un pulsador) la marca *Arranque Remoto* desde el TIA PORTAL en estado de conexión online y observación

con el PLC SIMATIC S7-1200, se comprobará el correcto funcionamiento de la conexión OPC-PLC, véase que en la **Figura 52** como la salida cambio de estado cero (0) a uno (1) (dato booleano).

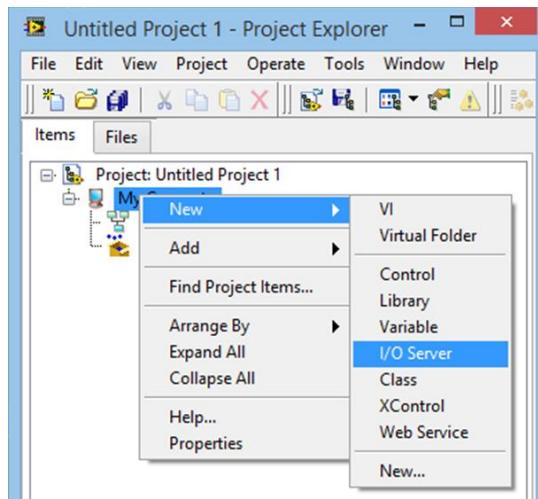
Item ID	Data Type	Value	Timestamp	Quality	Update C
Canal1.SIMATIC S7-1200._Rack	Byte	0	15:43:49.557	Good	1
Canal1.SIMATIC S7-1200._Slot	Byte	1	15:43:49.579	Good	1
Canal1.SIMATIC S7-1200.Arriague Remoto	Boolean	0	15:43:49.870	Good	1
Canal1.SIMATIC S7-1200.Parada Remota	Boolean	0	15:43:49.870	Good	1
<b>Canal1.SIMATIC S7-1200.Salida</b>	Boolean	<b>1</b>	15:44:55.563	Good	2

**Figura 52.** Cambio de valor (estado) del tag Salida.

#### 4.3. Configuración de LabVIEW

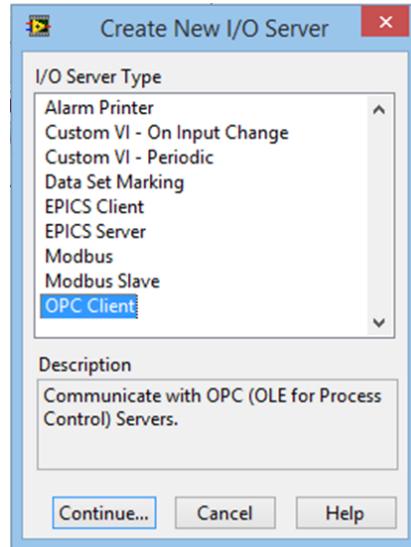
Configurado correctamente el servidor OPC se ajustará como “cliente OPC” al programa LabVIEW para la posterior creación de HMIs.

Ejecute LabVIEW y cree un nuevo proyecto en blanco (*Blank Project*), por ende se abrirá un nuevo *Project Explorer*, en él pulse clic derecho sobre *My Computer* y seleccione *New / I/O Server*, tal como se muestra en la **Figura 53**.



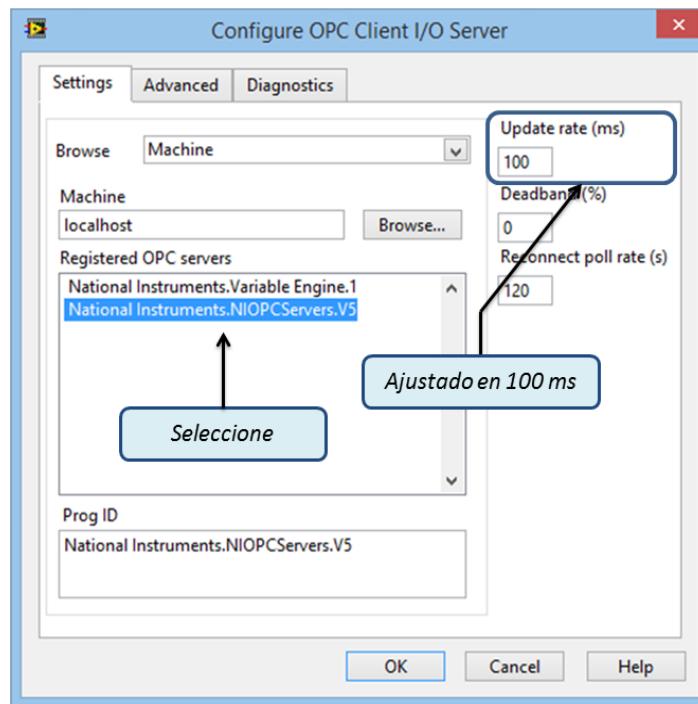
**Figura 53.** Selección de un nuevo I/O Server.

En I/O Server será posible crear el *OPC client* (Cliente OPC), véase la **Figura 54**.



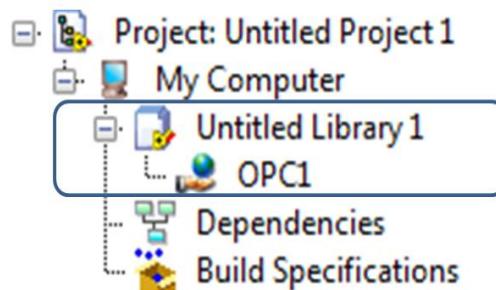
**Figura 54.** Selección de OPC Client.

Seleccionado *OPC Client* se desplegará una ventana llamada *Configure OPC Client I/O Server*, configúrela como se muestra en la **Figura 55** (el *Update rate (ms)* ajústelo en **100 ms**).



**Figura 55.** Ventana de configuración del OPC Client.

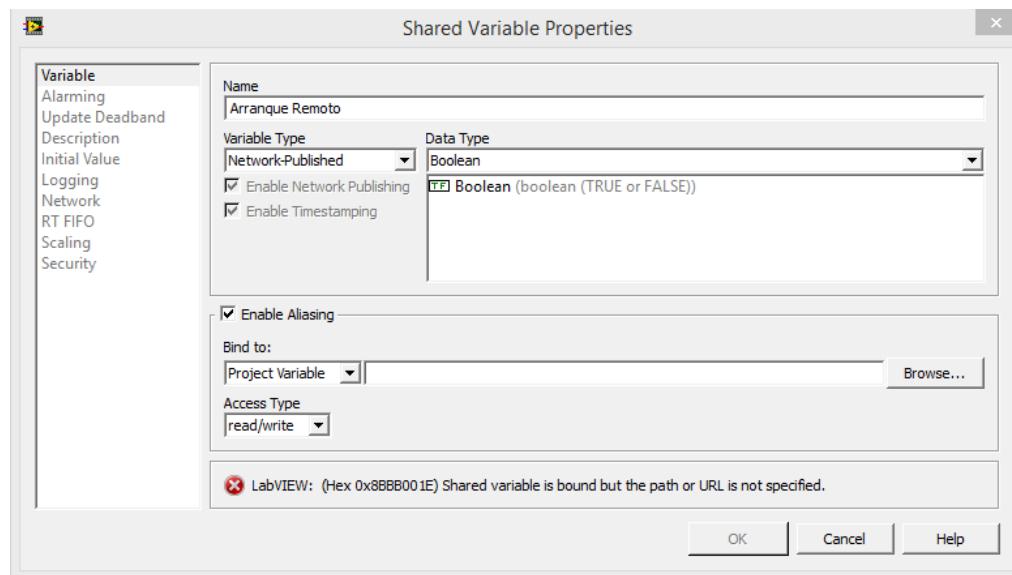
Finalizada la operación anterior se creará dentro del Proyecto *Untitled Project 1* el archivo (librería) *Untitled Library 1* y a su vez dentro de este el OPC1 (OPC Client), ver **Figura 56.**



**Figura 56.** *Untitled Library 1* y OPC1 agregados.

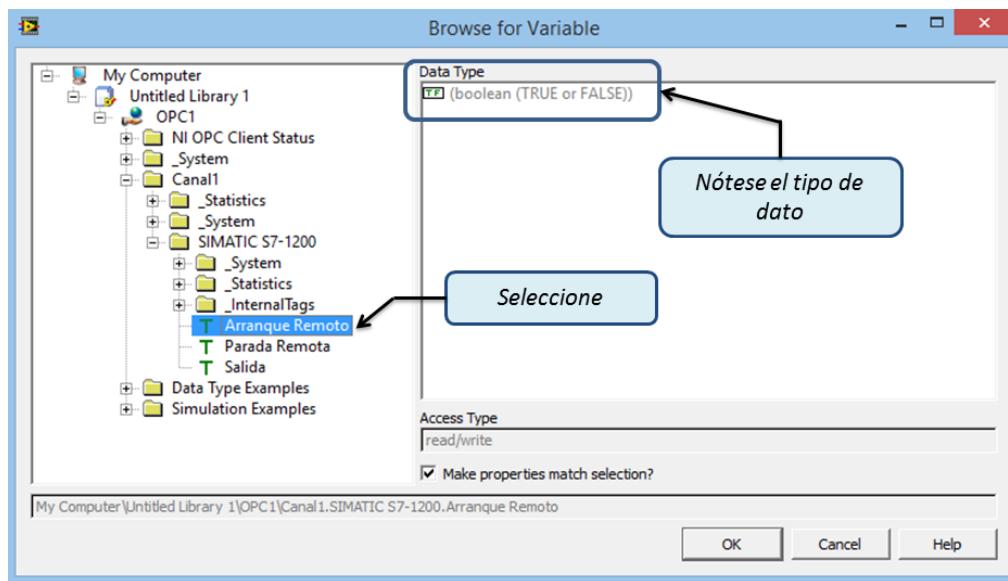
Posteriormente, agregue a la librería *Untitled Library 1* las variables que se requerirán para la futura creación de la interfaz gráfica, para materializar esto último, haga clic derecho con el ratón sobre *Untitled Library 1* y siga la ruta *New / Variable*,

inmediatamente se desplegará una ventana llamada *Share Variable Properties*, donde será posible crear las variables participantes en la interfaz, véase la **Figura 57**.

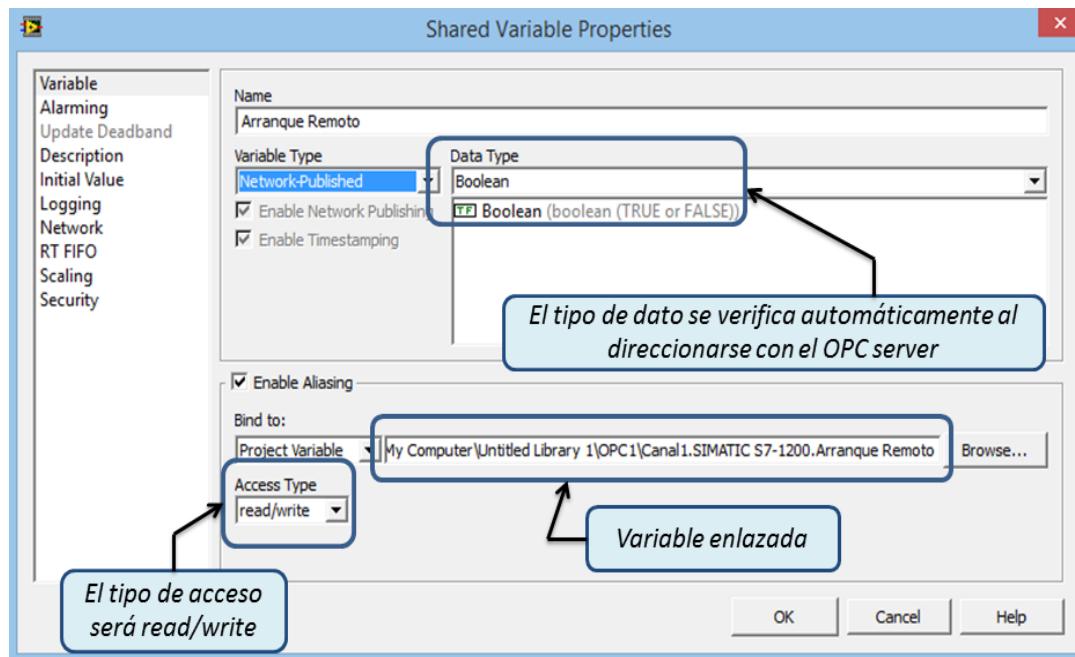


**Figura 57.** Creación de la variable *Arranque Remoto*.

La primera variable en ser creada (en este caso) que ejemplificará este procedimiento será *Arranque Remoto*, nótese que en la **Figura 57** ya aparece rellenado el campo *Name*, posteriormente de haber rellenado dicho campo vaya al campo *Bind to* y pulse *Browse*, esto le permitirá enlazar o direccionar dicha variable con la respectiva creada (*tag*) en el servidor OPC, ver **Figuras 58 y 59**.

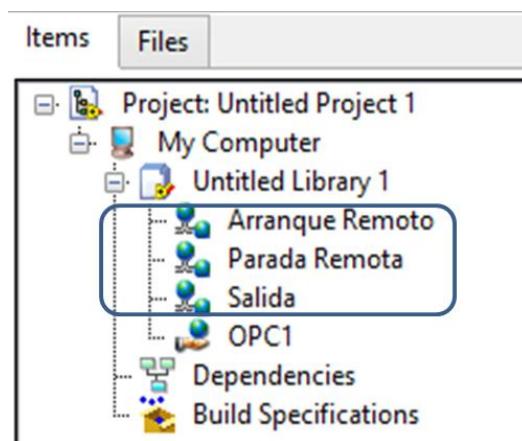


**Figura 58.** Enlace de la variable *Arranque Remoto*.



**Figura 59.** Variable *Arranque Remoto* configurada.

Configurada la variable *Arranque Remoto*, el mismo procedimiento se realizará con las otras dos (*Parada Remota* y *Salida*). En la **Figura 60** aparecen todas las variables ya configuradas y dispuestas dentro de *Untitled Library 1*.

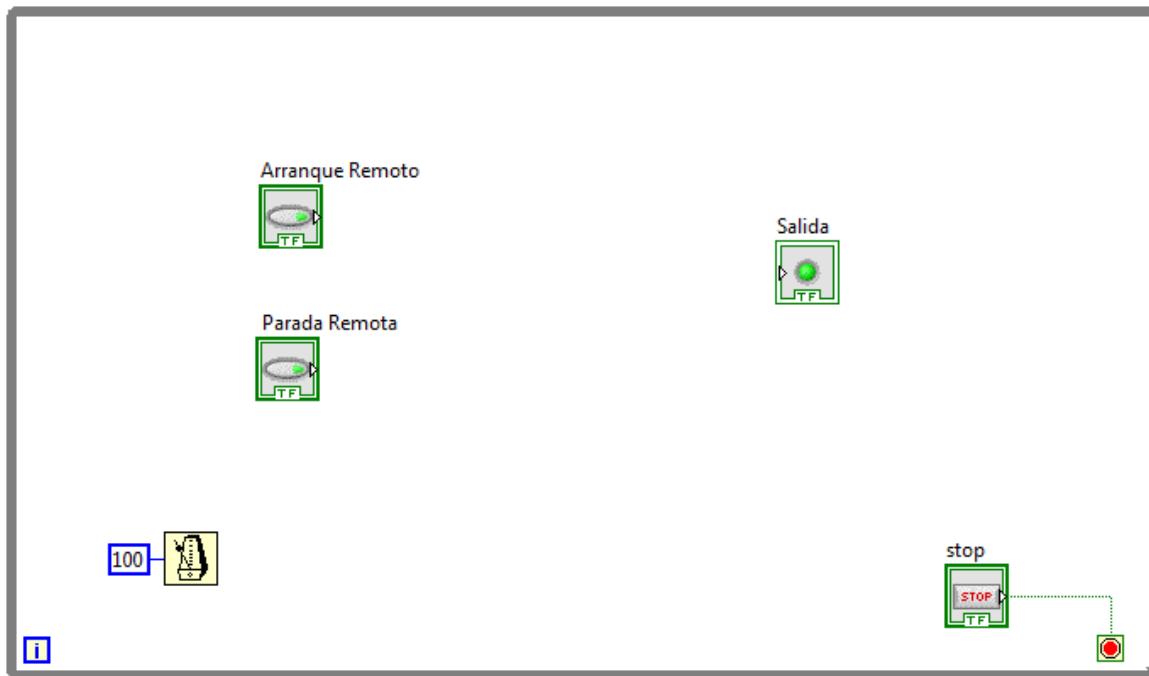


**Figura 60.** Variables creadas y configuradas dispuestas en la librería.

#### 4.4. Crear VI del proyecto

Cree un nuevo VI, seguidamente seleccione dos (2) *Push Button* y un (1) *Round LED*, agréguelos al área de trabajo del *Front Panel*, identifique los controles como *Arranque Remoto*, *Parada Remota* y el indicador como *Salida*. A los controles *Arranque Remoto* y *Parada Remota* configúrelos para que funcionen como pulsadores, para esto seleccione en *Mechanical Action* (Acción mecánica) la opción *Switch Until Released*.

Desde el *Block Diagram* haga uso de una estructura *While Loop* y encierre los controles e indicadores; configure dicha estructura con su respectivo control de ejecución *Stop* y velocidad del bucle *Wait Until Next ms Multiple* ajustado mediante una constante en **100 milisegundos**, tal como es evidenciado en la **Figura 61**.

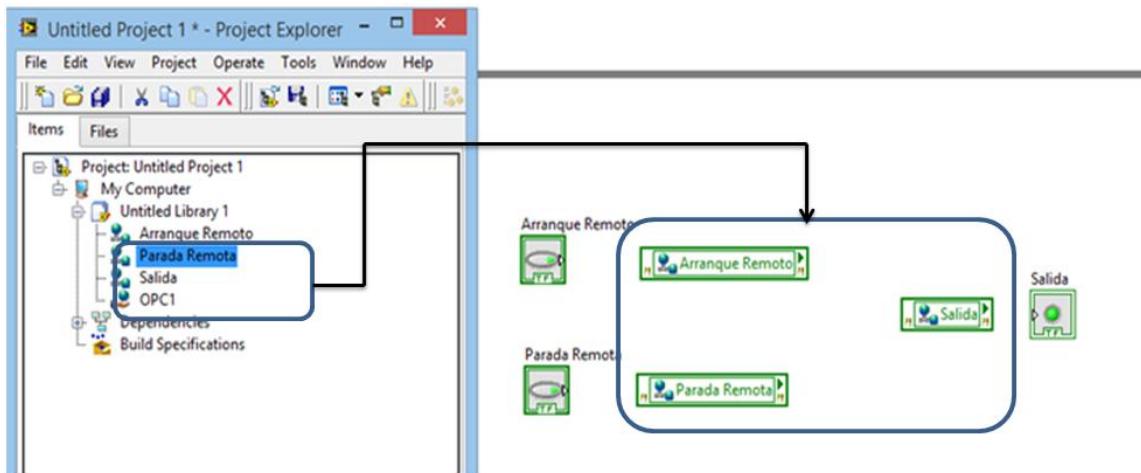


**Figura 61.** Block Diagram del VI en creación.

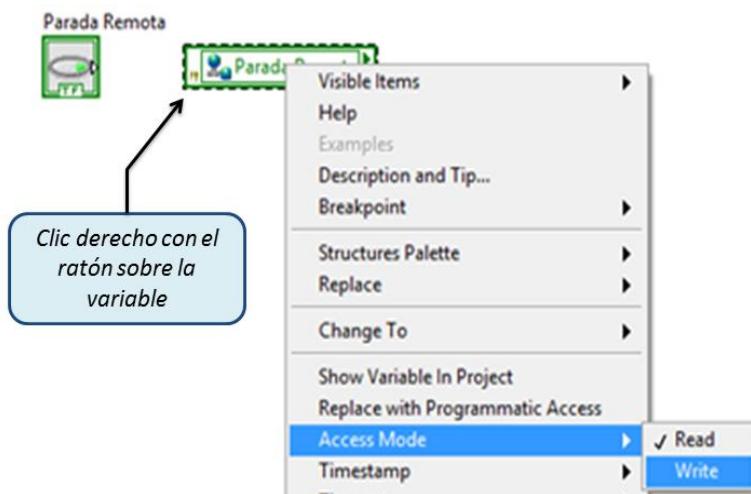
Posteriormente, conecte los controles (*Arranque Remoto* y *Parada Remota*) y el indicador (*Salida*) con sus respectivas variables a escribir y leer respectivamente, las cuales ya han sido creadas en *Untitled Library 1*, para ello existen dos alternativas:

La primera alternativa es haciendo clic derecho con el ratón sobre el (en este caso) control *Arranque Remoto* y siguiendo la ruta *Create / Share Variable Node / My Computer / Untitled Library 1 / Arranque Remoto*, por defecto la variable será dispuesta y enlazada con el tipo de acceso adecuado para el control (*Write*).

La segunda alternativa es ubicarse en el *Project Explorer* y arrastrar la variable hacia la ventana del *Block Diagram*, después configurar el tipo de acceso de la variable para ser enlazada con el control o indicador dependiendo del caso, véase la **Figura 62 y 63** donde se ejemplifica esta última alternativa.



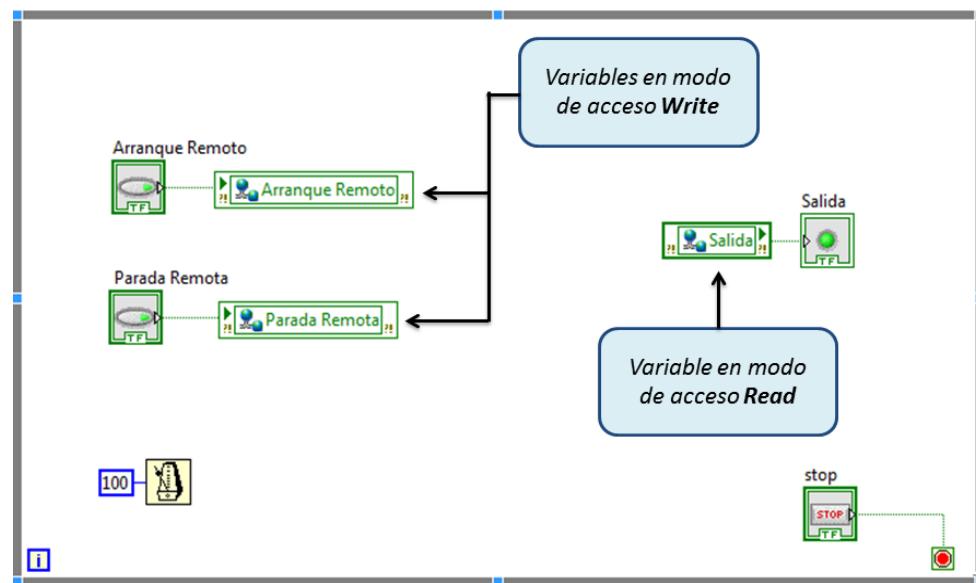
**Figura 62.** Despliegue de las variables en el *Front Panel*.



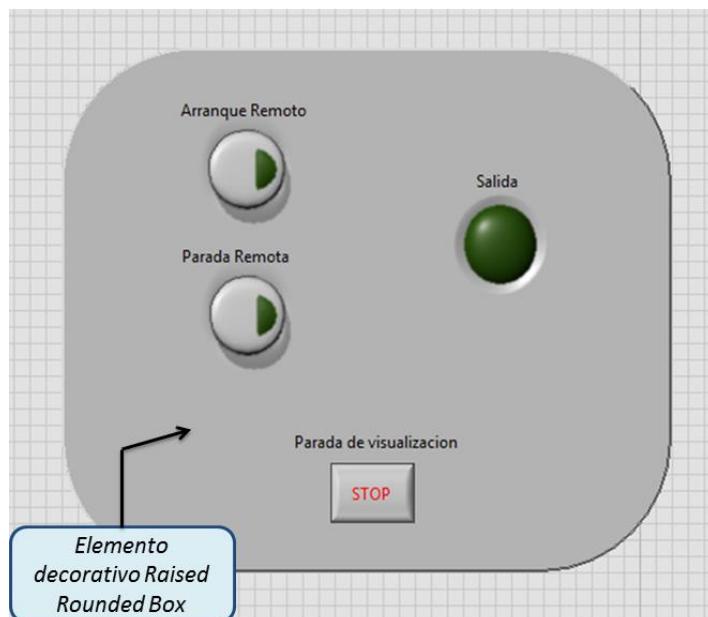
**Figura 63.** Cambio de tipo de acceso de *Read* a *Write*.

**Nota:** Los **Controles escriben** sobre las variables, los **indicadores leen** desde las variables.

Configurado el modo de acceso de las variables, el enlace entre las variables, los instrumentos en el *Block Diagram* y ajustada la dimensión de los instrumentos en el *Front Panel*, puede verse el VI terminado para esta etapa de la práctica en las **Figuras 64 y 65**.



**Figura 64.** Block Diagram del VI.



**Figura 65.** Front Panel del VI.

Ejecute el VI en el modo *Run* de LabVIEW y pulse el control *Arranque Remoto*, nótese como se enciende el indicador *Salida* del VI y la *Salida* (Q0.0) en el PLC, ver **Figura 66**. Para desactivar la *Salida*, simplemente pulse *Parada Remota* en el VI o pulse *Parada manual* desde las entradas físicas del PLC.

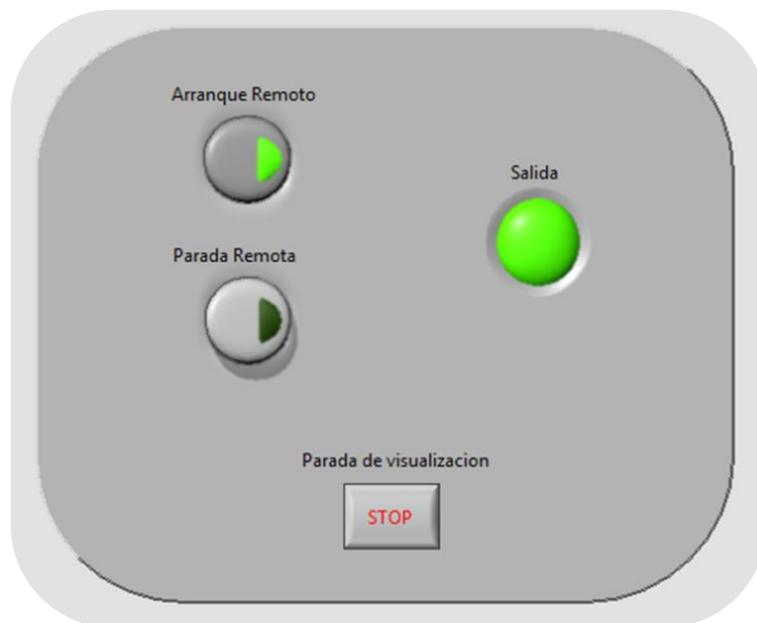


Figura 66. *Front Panel* del VI ejecutado con *Salida* (Q0.0) activada.

#### 4.5. Adicionar al proyecto un temporizador controlado y supervisado desde LabVIEW.

Remítase al proyecto creado en TIA PORTAL, específicamente a la lógica creada en *Main [OB1]*, a dicha lógica se le integrará un temporizador TON (Temporizador con retardo al conectar) con la finalidad de temporizar la activación de la *Salida* (Q0.0).

Diríjase a *Instrucciones / Instrucciones básicas / Temporizadores*, seleccione el TON y arrástrelo con el ratón hacia el Segmento 1 del código, dispóngalo y configúrelo tal como es mostrado en la **Figura 67**.

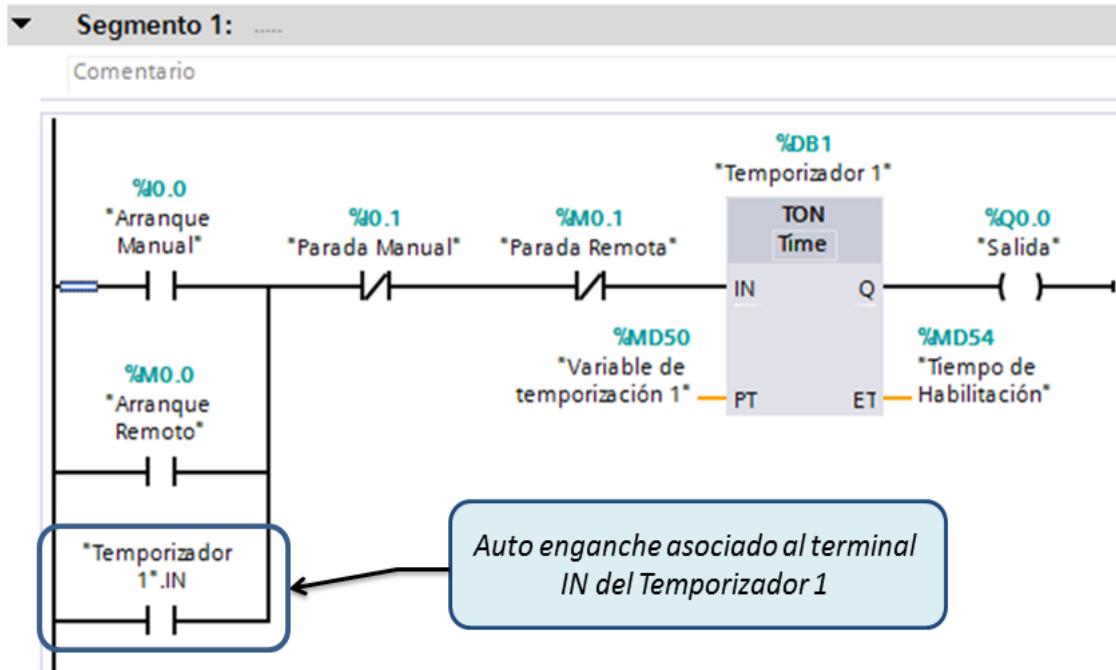


Figura 67. Temporizador TON integrado a la lógica.

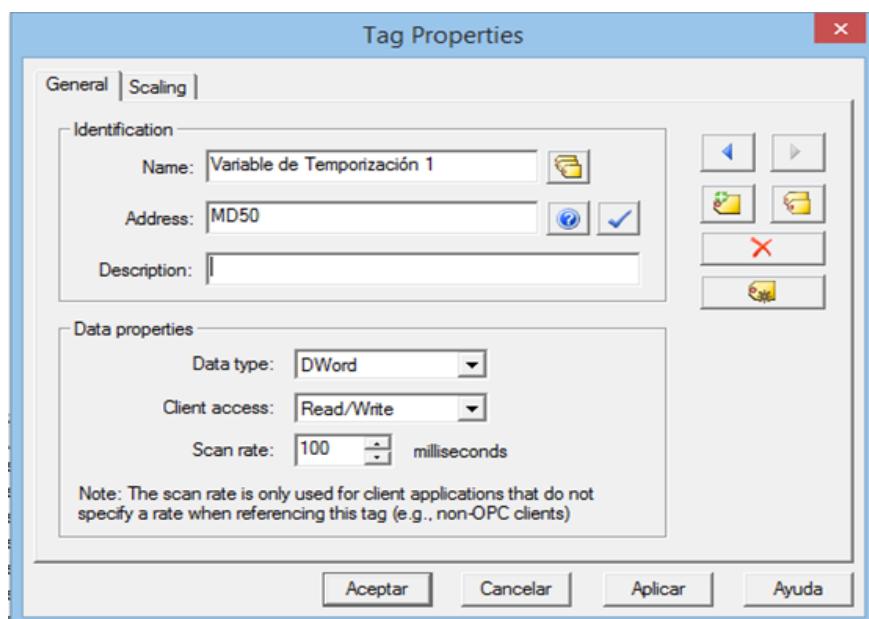
En la **Figura 67**, pueden observarse importantes variaciones con respecto a la lógica diseñada anteriormente (**Figura 8**), las cuales son:

- Inclusión y configuración del temporizador, obsérvese que en los parámetros tiempo preseleccionado PT (*Preset time*) y tiempo transcurrido ET (*Enable time*) se encuentran especificadas las variables *Variable de temporización 1* (MD50) y *Tiempo de Habilitación* (MD54) respectivamente, tales hacen referencia a el Área “M” (marcas en la memoria), “D” formato de doble palabra (32 bits) y por último el primer byte de la variable. Es importante acotar que dichas variables de doble palabra (DWord) son necesarias para almacenar datos del tipo *Time*.
- Sustitución del auto enganche *Salida* (Q0.0) por “*Temporizador 1*.IN”.

Modificada la lógica descárguela en el dispositivo (PLC), en el caso de que el dispositivo se encuentra en modo de ejecución *Run* cámbielo a *Stop* y pulse el botón *Cargar a dispositivo*, posteriormente vuelva a colocar el PLC en modo *Run*.

Descargado el programa se configurarán las nuevas variables agregadas al programa (*Variable de temporización 1* y *Tiempo de Habilitación*) en el servidor OPC.

Vaya al servidor OPC *Canal1 / SIMATIC S7-1200* y cree los nuevos *tags*, para exemplificar “rápidamente” este proceso se mostrará la creación del tag *Variable de Temporización 1*. Al ser verificada la dirección el campo *Data type* se ajusta al tipo de dato (*Dword*), véase **Figura 68**.



**Figura 68.** Creación del tag *Variable de Temporización 1*.

Agregados los nuevos *tags* se tiene lo mostrado en la **Figura 69**.

Tag Name	Address	Data Type	Scan Rate	Scaling
Variable de Temporización 1	MD50	DWord	100	None
Tiempo de Habilitación	MD54	DWord	100	None
Salida	Q0.0	Boolean	100	None
Parada Remota	M0.1	Boolean	100	None
Arranque Remoto	M0.0	Boolean	100	None

Figura 69. Tags Variable de Temporización 1 y Tiempo de Habilitación añadidos al servidor OPC.

Diríjase ahora a *Untitled Library 1* (LabVIEW) y añada las nuevas variables, véase las Figuras 70 y 71.

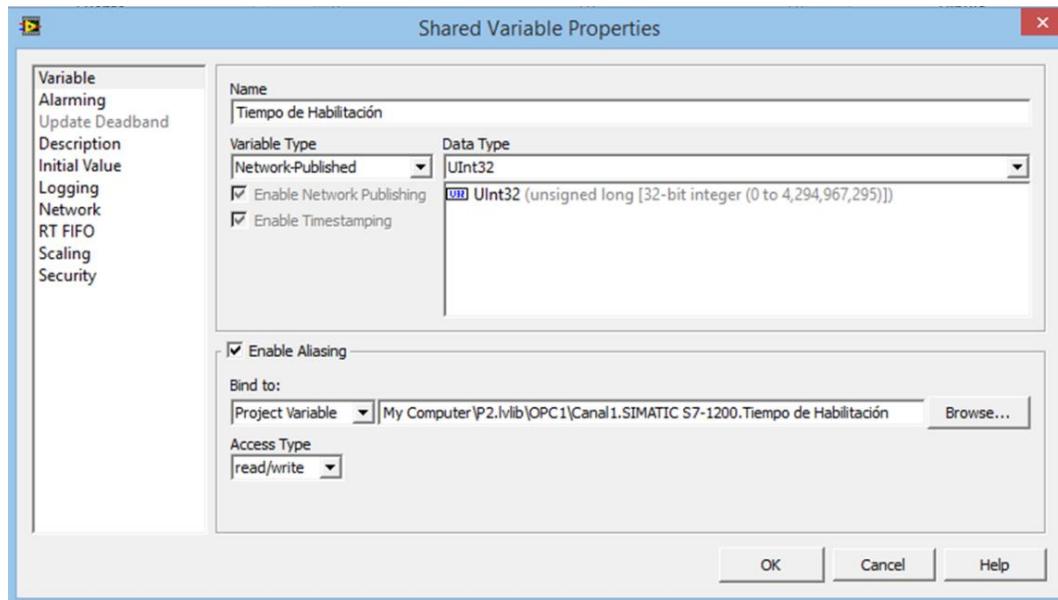
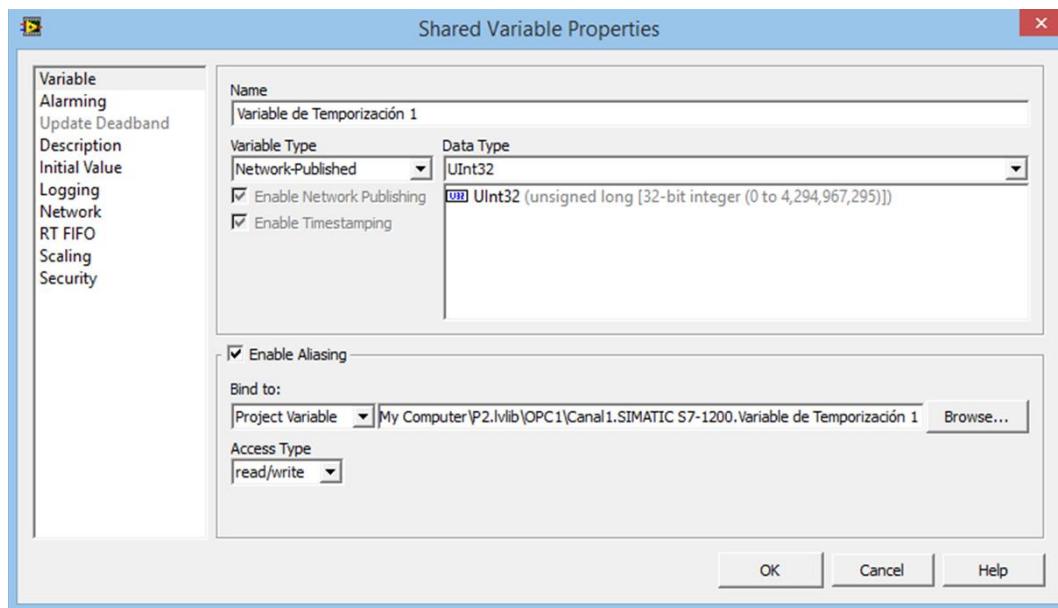
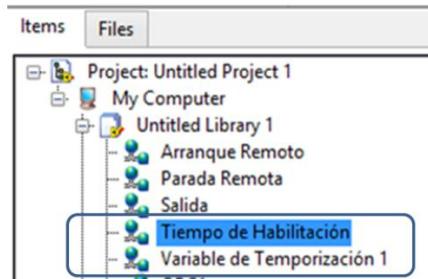


Figura 70. Creación de la variable Tiempo de Habilitación.



**Figura 71.** Creación de *Variable de Temporización 1*.

Configuradas las variables, estas pueden verse dispuestas desde el *Project Explorer*, ver **Figura 72**.

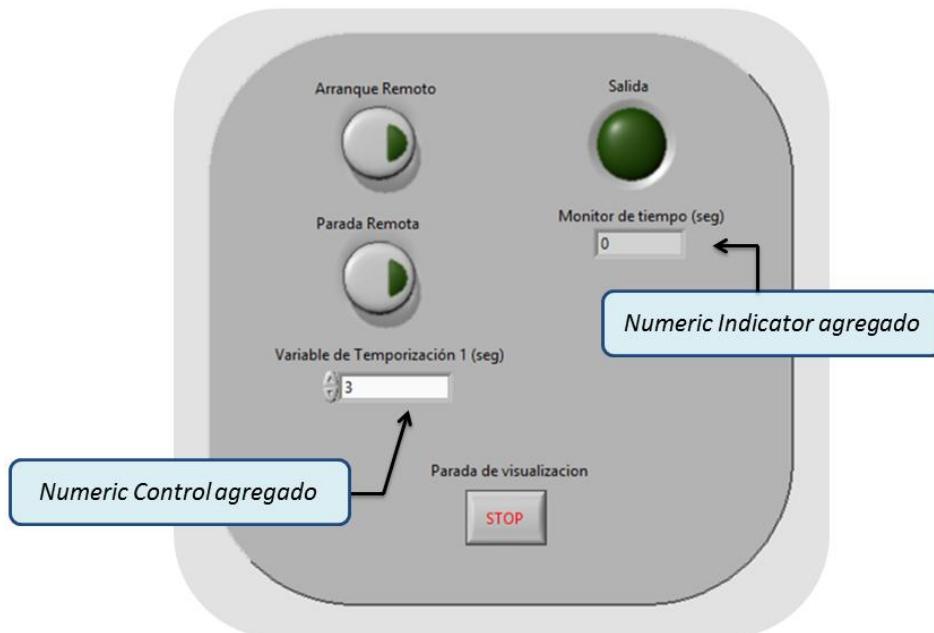


**Figura 72.** Variables *Tiempo de Habilitación* y *Variable de Temporización 1* agregadas a la librería.

Ahora, haga las modificaciones pertinentes en el VI realizado anteriormente insertando el control y monitoreo del temporizador encargado de controlar la activación de la *Salida* (Q0.0) en el PLC.

Vaya al *Front Panel* y seleccione un control *Numeric Control* ubicado en la ruta *Controls / Numeric*, despliéguelo sobre el área de trabajo del *Front Panel*, de igual

manera despliegue un indicador *Numeric Indicator* presente en la misma ruta, configure el nuevo *Front Panel* del VI como es mostrado en la **Figura 73**.



**Figura 73.** *Front Panel* del VI con control e indicador de temporización integrados a la interfaz gráfica (VI).

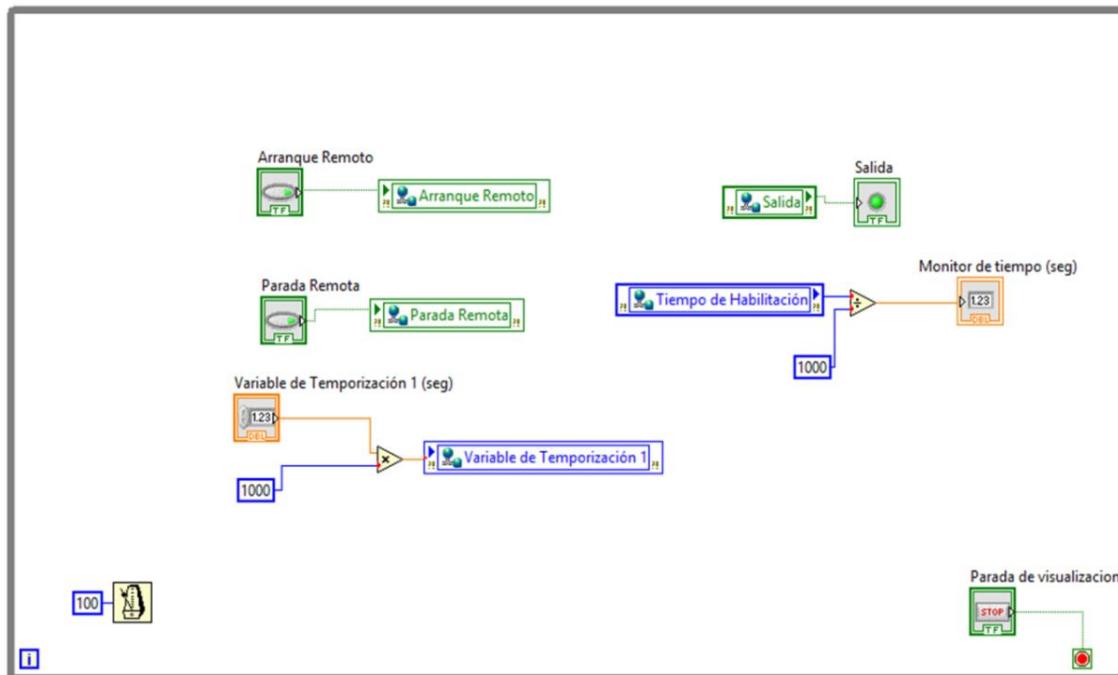
Después deberá programar el control *Variable de temporización 1* y el indicador *Monitor de tiempo*, para esto remítase al *Block Diagram* y programe los diagramas para cada instrumento tal como pude verse en la **Figura 74**.

**Programación del control *Variable de Temporización 1*:** para programar correctamente este control son necesarios 3 elementos, la variable que será asociada al control la cual ya ha sido creada en *Untitled Library 1* como *Variable de Temporización 1* , un multiplicador numérico *Multiply* y una constante numérica *Numeric Constant*, estos dos últimos elementos se encuentran presenten en *Functions / Numeric*.

En la **Figura 74** puede observase como el valor dado por el control *Variable de temporización 1* se multiplica por el valor de la *Numeric Constant*, en este caso en

particular con un valor de 1000, esto es debido a que tanto LabVIEW como el PLC trabajan por defecto los datos asociados a tiempo en milisegundos, por lo tanto para ordenarle al PLC que temporice la activación de la *Salida* (Q0.0) en 3 segundos es necesario aportarle este valor (al servidor OPC y subsecuentemente al PLC) en milisegundos, es decir, 3000 ms.

**Programación del indicador *Monitor de tiempo*:** para programar este indicador se aplica la lógica inversa que se utilizó para el control *Variable de Temporización 1*, el dato recibido *Tiempo de Habilitación* desde el servidor OPC (cuya fuente es el PLC) viene dado en milisegundos, por esa razón, para visualizar dicho valor en segundos se hace uso de una constante numérica *Numeric Constant* cuyo valor es de 1000 y un divisor numérico *Divide* que se conectan entre la variable *Tiempo de Habilitación* y el indicador *Monitor de tiempo*.



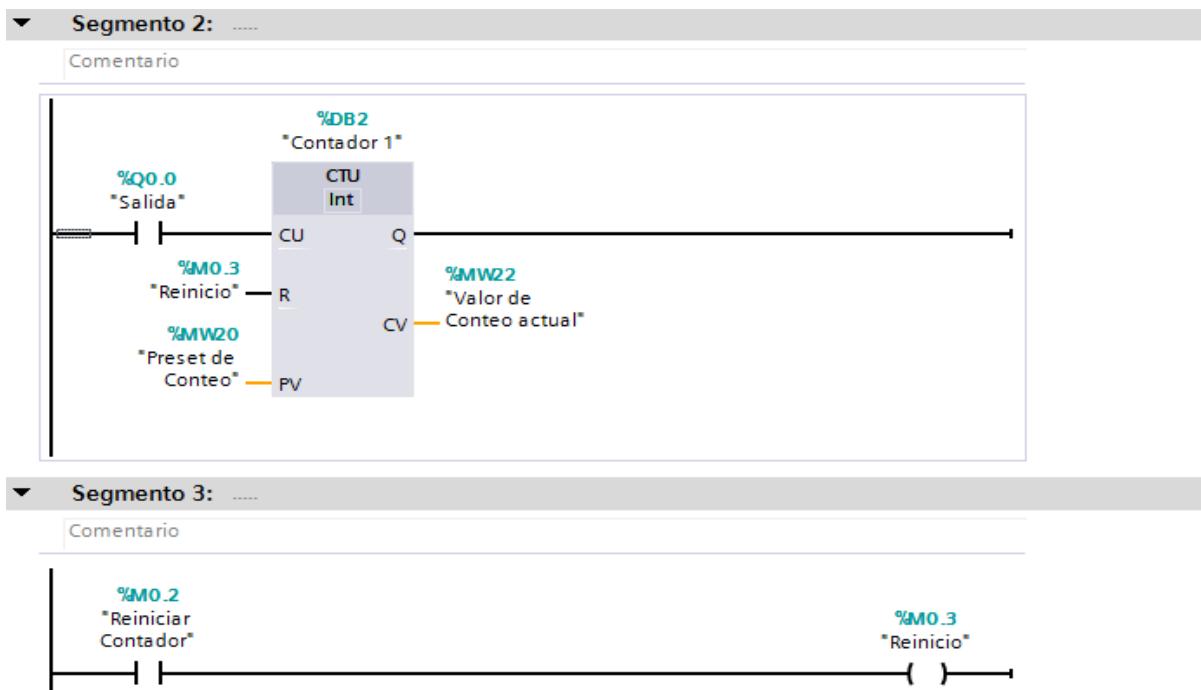
**Figura 74.** Block Diagram del VI con control e indicador de temporización integrados.

Ejecute y verifique el VI haciendo uso de la metodología planteada anteriormente.

#### 4.6. Adicionar al proyecto un contador controlado y supervisado desde LabVIEW.

Diríjase al proyecto creado en TIA PORTAL específicamente a la lógica creada en *Main [OB1]*, a dicha lógica se le integrará un contador CTU (Contador ascendente o de incremento) con la finalidad de contabilizar las veces que será activada la *Salida* (Q0.0).

Vaya a *Instrucciones / Instrucciones básicas / Contadores*, seleccione el CTU arrástrelo y dispóngalo con el ratón en el Segmento 2 del código, además añada en el Segmento 2 y 3 un contacto normalmente abierto por cada segmento y una instrucción de asignación en el Segmento 3. Dispuesto el contador, contactos e instrucción de asignación configúrelas como es mostrado en la **Figura 75**.



**Figura 75.** Contador CTU integrado a la lógica.

En la **Figura 75** desde los Segmentos 2 y 3 puede evidenciarse la nueva etapa de código insertada, en el Segmento 2 se encuentra un contacto normalmente abierto asociado a la variable *Salida* (Q0.0), esta al activarse (cambia de estado false a true) energizará el terminal CU del CTU y de esta forma el contador empezará a almacenar

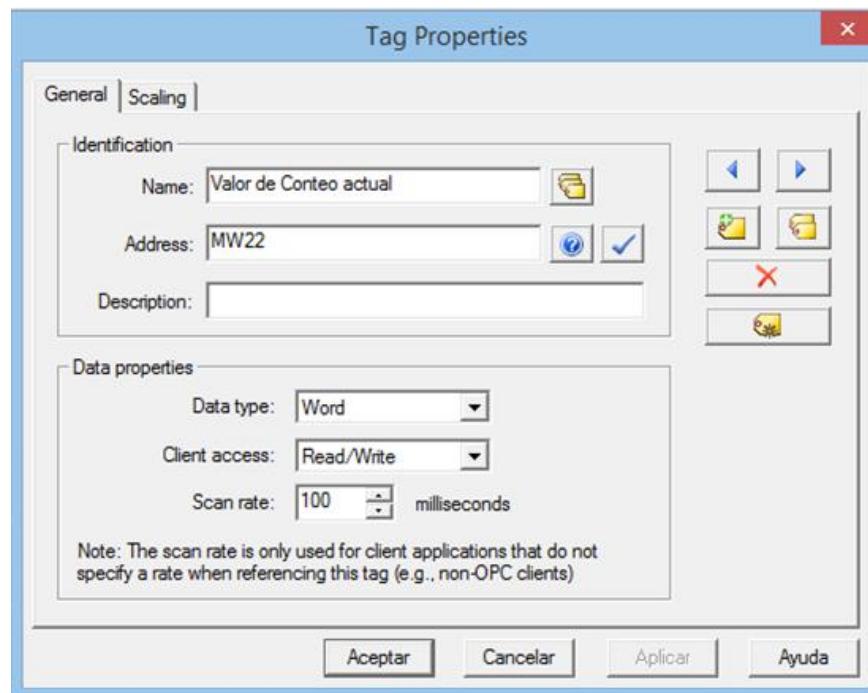
la cuenta de modo ascendente en la variable *Valor de Conteo actual* (MW22) asociada al terminal CV (*Count Value*). En el Terminal PV se encuentra asociada la variable *Preset de Conteo* (MW20) que por defecto está ajustada a un valor de 0, dicha variable no será asociada a la interfaz gráfica..

Desde el segmento 3 se puede observar la presencia de un contacto normalmente abierto llamado *Reiniciar Contador* (M0.2) y una bobina asociada a la variable *Reinicio* (M0.3), al activarse el contacto se energizará la bobina, lo que genera un cambio de estado en dicha bobina de false a true, dicho estado es direccionado al terminal R (*Reset*) del contador CTU, por lo que el contador reiniciará su cuenta al percibir el cambio de estado.

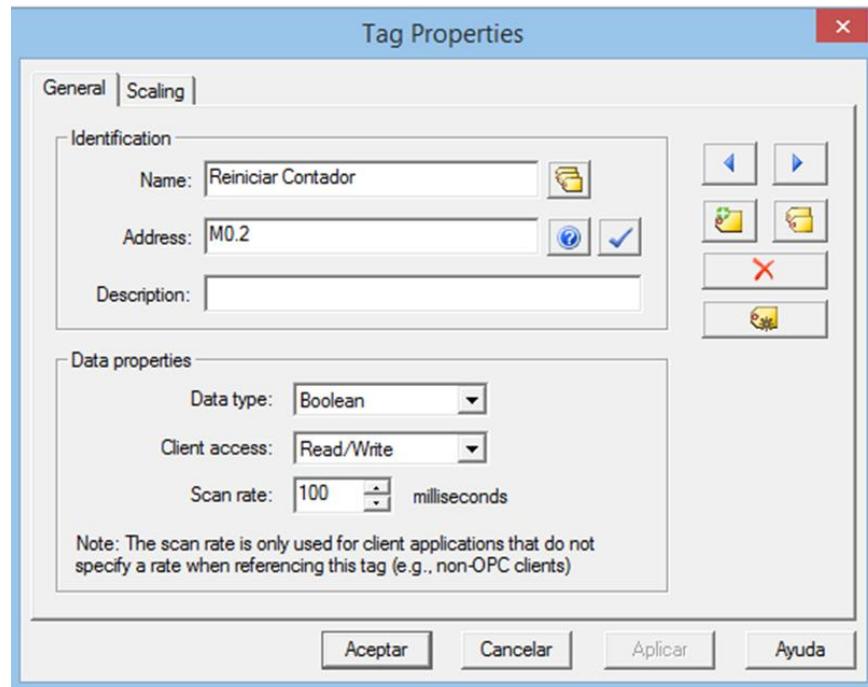
Modificada la lógica proceda a descargarla en el PLC.

Descargado el programa, configure las nuevas variables agregadas al programa (*Valor de Conteo actual* y *Reiniciar Contador*) en el servidor OPC.

De la **Figura 76 a 78** se puede evidenciar la configuración e inserción de los nuevos **tags**.



**Figura 76.** Creación del tag *Valor de Conteo actual*.

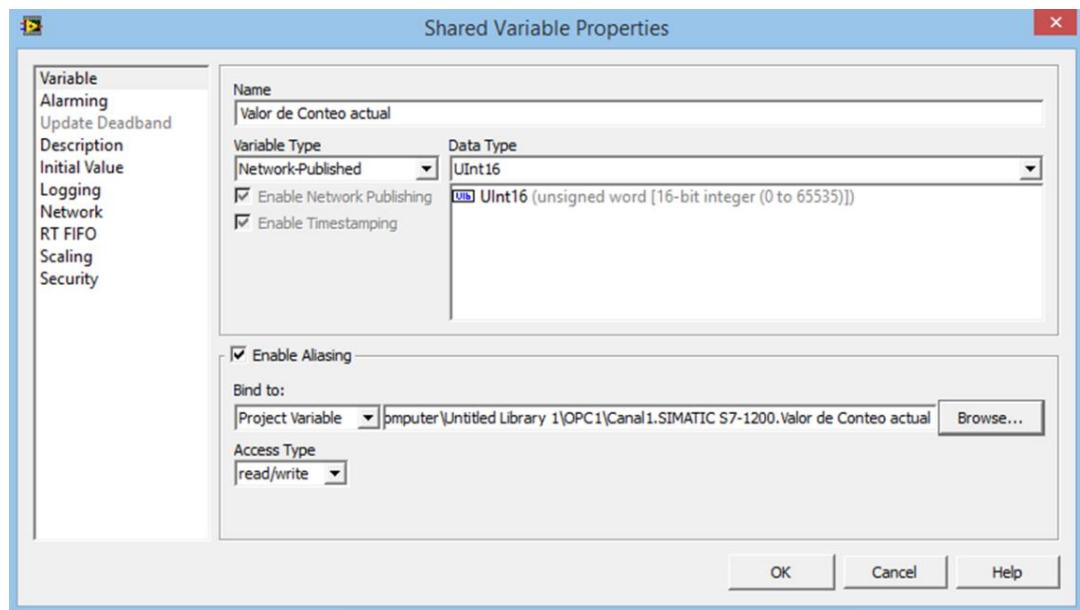


**Figura 77.** Creación del tag *Reiniciar Contador*.

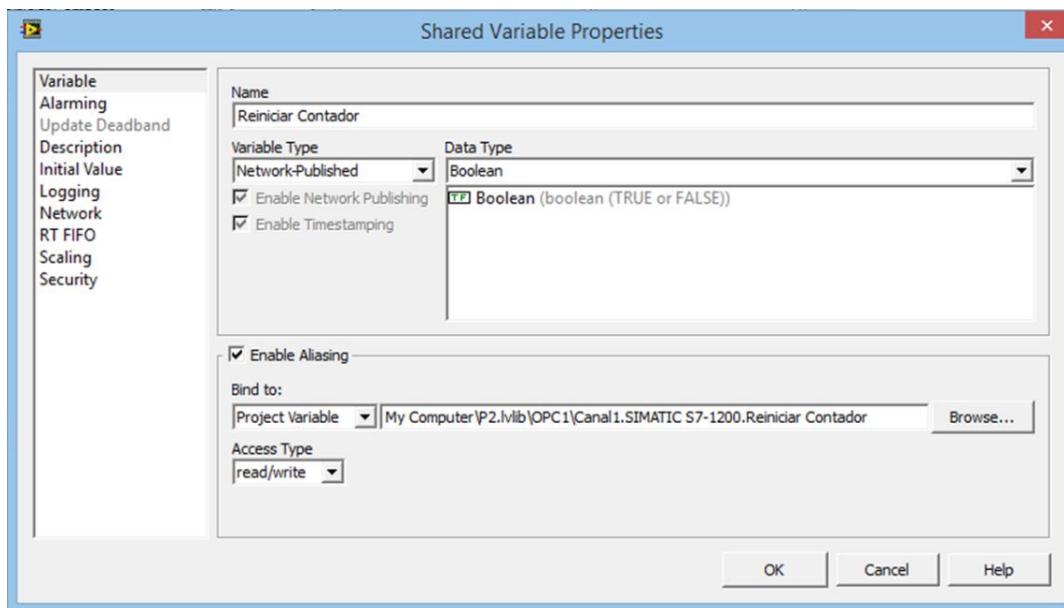
Tag Name	Address	Data Type	Scan Rate	Scaling
Variable de Temporización 1	MD50	DWord	100	None
Valor de Conteo actual	MW22	Word	100	None
Tiempo de Habilitación	MD54	DWord	100	None
Salida	Q0.0	Boolean	100	None
Reiniciar Contador	M0.2	Boolean	100	None
Parada Remota	M0.1	Boolean	100	None
Arranque Remoto	M0.0	Boolean	100	None

**Figura 78.** Tags *Valor de Conteo actual* y *Reiniciar Contador* añadidos al servidor OPC.

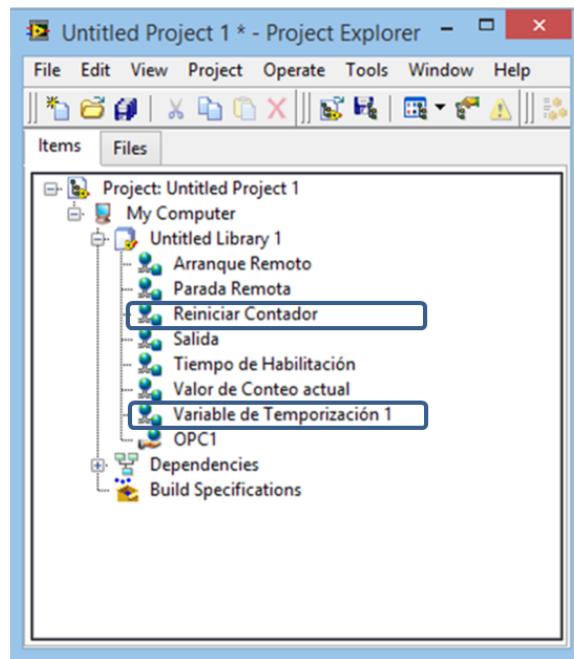
Diríjase ahora a *Untitled Library 1* (LabVIEW), en el añada y configure las nuevas variables, véase la **Figura 79, 80 y 81**.



**Figura 79.** Creación de la variable *Valor de Conteo actual*.



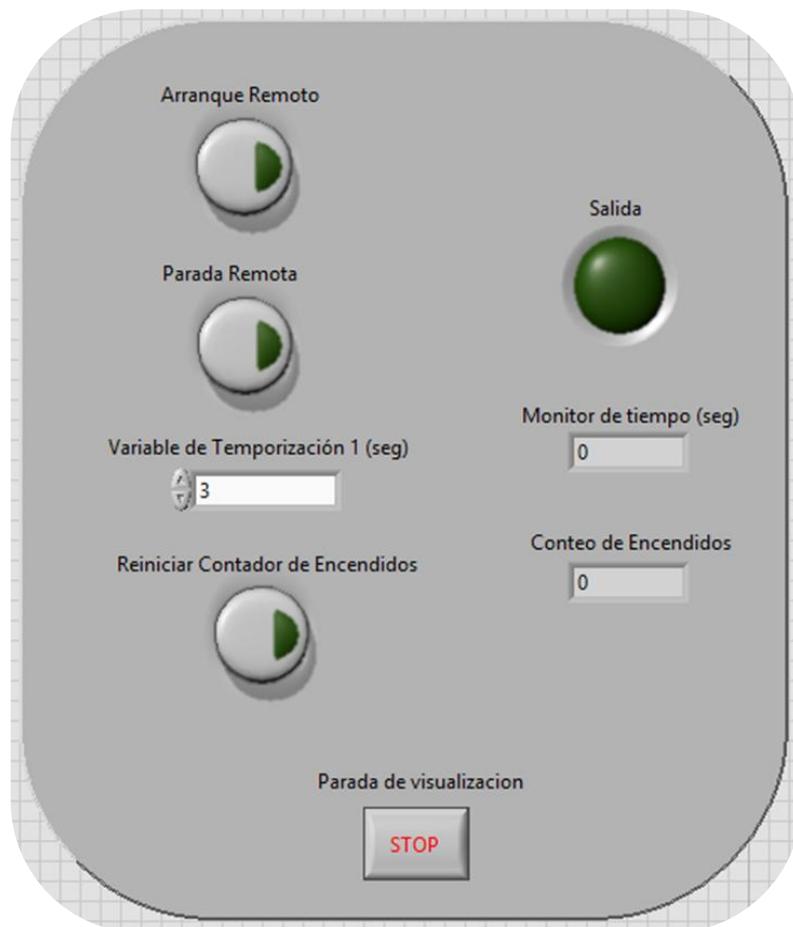
**Figura 80.** Creación de la variable *Reiniciar Contador*.



**Figura 81.** Variables *Valor de Conteo Actual* y *Reiniciar Contador* anexadas a la librería.

Ahora, haga las modificaciones pertinentes en el VI realizado anteriormente para insertar el control y monitoreo del contador encargado de contabilizar el número de veces la activación de la *Salida* (Q0.0) en el PLC.

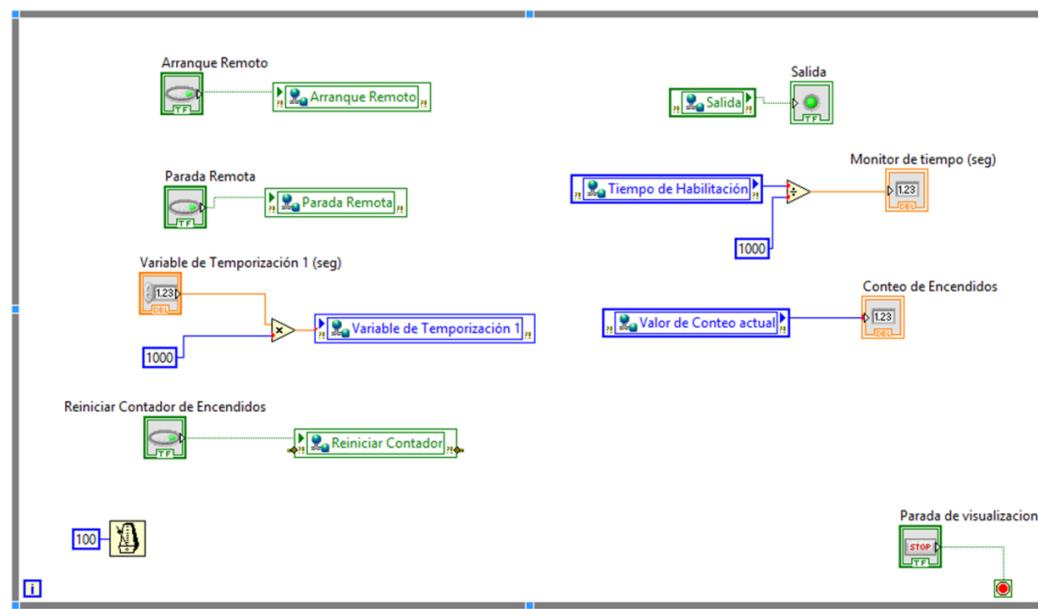
Vaya al *Front Panel* y seleccione un control *Push Button*, desplíéguelo sobre el área de trabajo del *Front Panel*, de igual manera despliegue un indicador *Numeric Indicator*, configure el nuevo *Front Panel* del VI como es mostrado en la **Figura 82**.



**Figura 82.** *Front Panel* del VI con control e indicador del contador integrados a la interfaz.

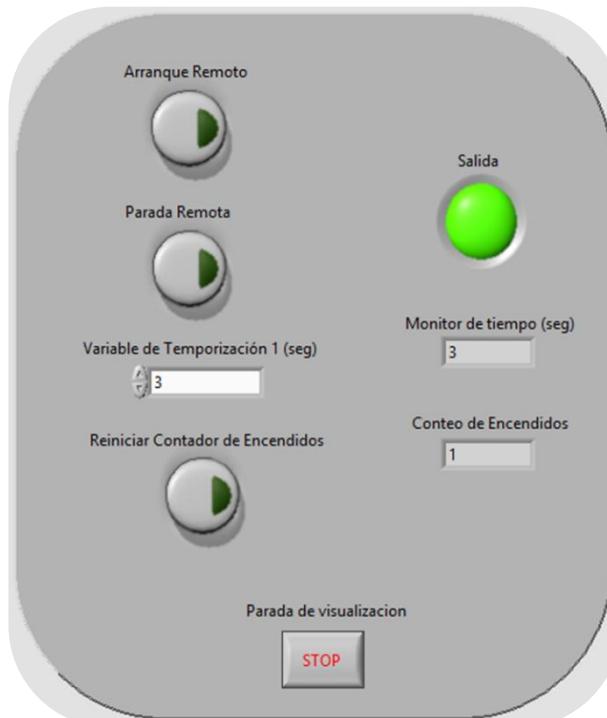
Configure el control *Reiniciar Contador de Encendidos* en modo pulsador (*Switch Until Released*) tal como fue hecho con los controles *Arranque Remoto* y *Parada Remota*.

Luego, programe el control *Reiniciar Contador de Encendidos* y el indicador *Conteo de Encendidos* como se muestra en la **Figura 83**.



**Figura 83.** Block Diagram del VI con control e indicador de conteo integrados.

Finalizado el VI ejecútelo teniendo al PLC en modo *Run*, en la **Figura 84** puede observarse el VI en ejecución.



**Figura 84.** VI Finalizado en modo de ejecución (contador agregado).

#### 4.7. Actividad (alternativa)

Diseñe el programa y la HMI para la “**simulación**” de un mezclador con las siguientes pautas:

- Se mezclarán dos ingredientes (sustancias). El vaciado de cada ingrediente en el tanque será controlado por una válvula normalmente cerrada, es decir, por cada ingrediente (dos en este caso) irá una válvula de control.
- La capacidad de llenado del tanque será de 10000 litros, el límite de llenado de cada ingrediente será de 5000 litros, este será controlado por un sensor de nivel (simulado) que determinará la cantidad de cada uno en el tanque para la continuidad del proceso. La cantidad de llenado por ingrediente deberá ser controlada por el usuario a través de la HMI.
- No se asumirán tiempos de reposo entre el llenado de cada ingrediente, es decir, al finalizar el llenado del primer ingrediente comenzará inmediatamente el llenado del segundo ingrediente.
- Mientras el proceso esté en el estado de llenado del segundo ingrediente, y el valor de llenado del primer ingrediente se modifique incrementándolo, al finalizarse el llenado del segundo ingrediente, el proceso deberá detenerse para reiniciarse manualmente con la continuación llenado del primer ingrediente. Finalizado el relleno del primer ingrediente si se incrementó el valor de llenado del segundo ingrediente, automáticamente se activará la válvula 2 y se continuará con el rellenado del segundo ingrediente.
- Finalizado el llenado de ambos ingredientes inmediatamente se iniciará la etapa de mezclado, el tiempo de duración de la etapa de mezclado deberá ser

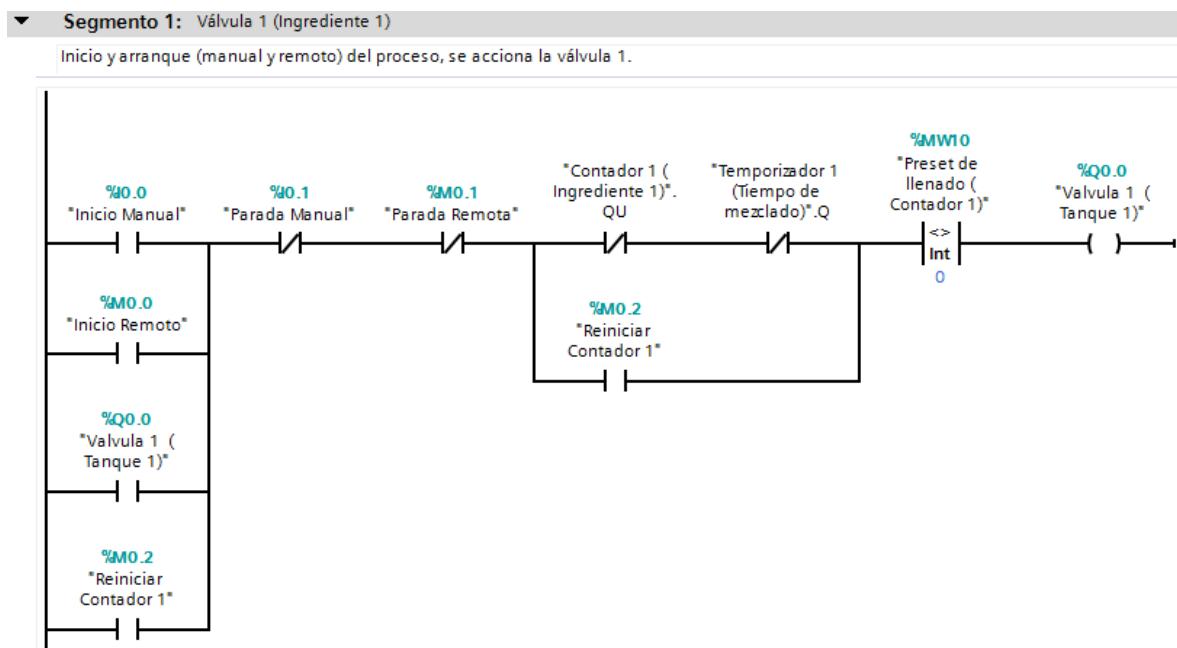
controlada por el operario a través de la HMI, este tiempo será controlado segundos. El mezclador deberá detenerse al dar la orden de parada (manual o remota) y el tiempo de mezclado se reiniciará.

- Finalizada la etapa de llenado y mezclado, se descargará la mezcla resultante mediante la implementación de una tercera válvula normalmente cerrada, terminada la descarga el proceso se volverá a repetir.
- El proceso contará con las ordenes inicio y parada manual (directamente en el PLC) e inicio y parada remota (desde la HMI). La orden de parada deberá detener el proceso en cualquiera de los estados sin alterar ninguno de ellos, de igual manera con la orden de inicio.

El programa se realizará en lenguaje de contactos (escalera), implemente las funciones de bits, temporización, conteo y de ser necesario, funciones matemáticas, de comparación y de transferencia.

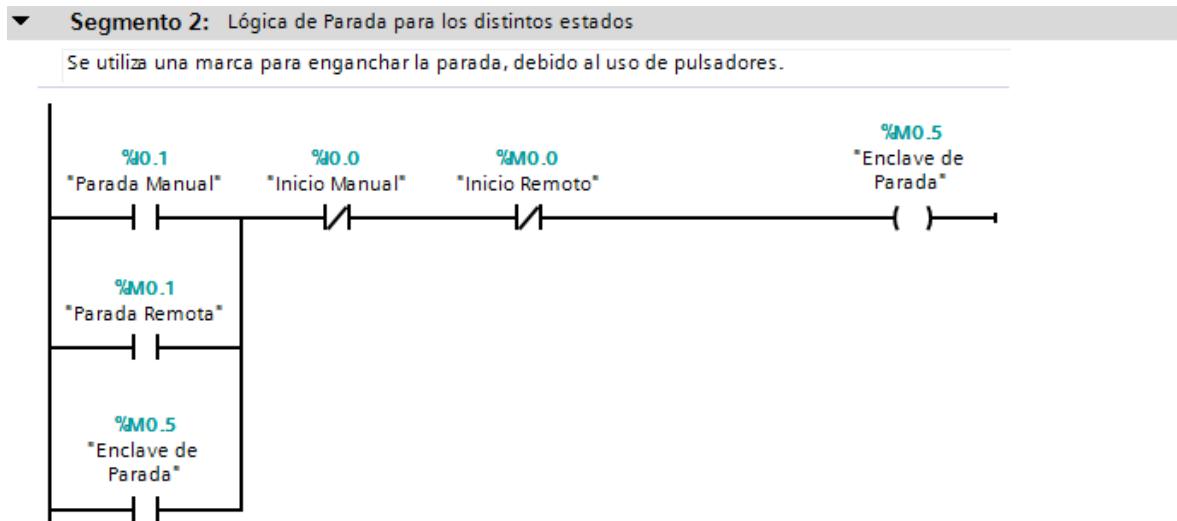
- Solución:

**Segmento 1:** válvula 1 (ingrediente 1), ver **Figura 85.**



**Figura 85.** Segmento 1.

**Segmento 2:** lógica de parada para los distintos estados, ver **Figura 86.**

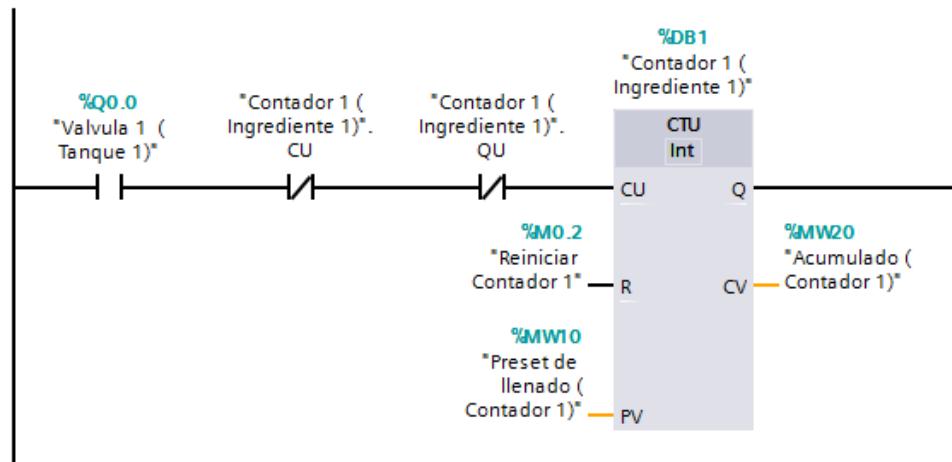


**Figura 86.** Segmento 2.

**Segmento 3:** contador que simula el llenado del tanque, asumiendo flujo constante (válvula 1 activa), ver **Figura 87**.

Segmento 3: Contador que simula el llenado del tanque asumiendo flujo constante (válvula 1 activa)

### Lógica para activar, enclavar y parar el conteo.

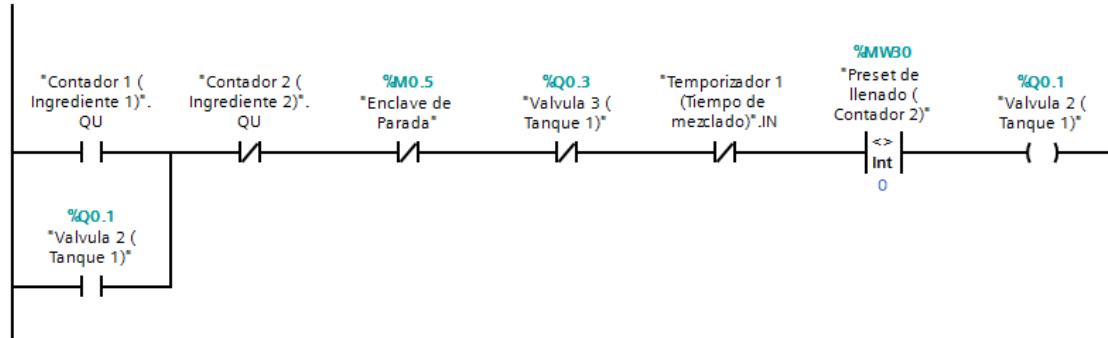


**Figura 87.** Segmento 3.

**Segmento 4:** válvula 2 (Ingrediente 2), ver **Figura 88.**

▼ Segmento 4: Válvula 2 (Ingrediente 2)

Lógica para activar y parar (manual y automáticamente) la válvula 2.

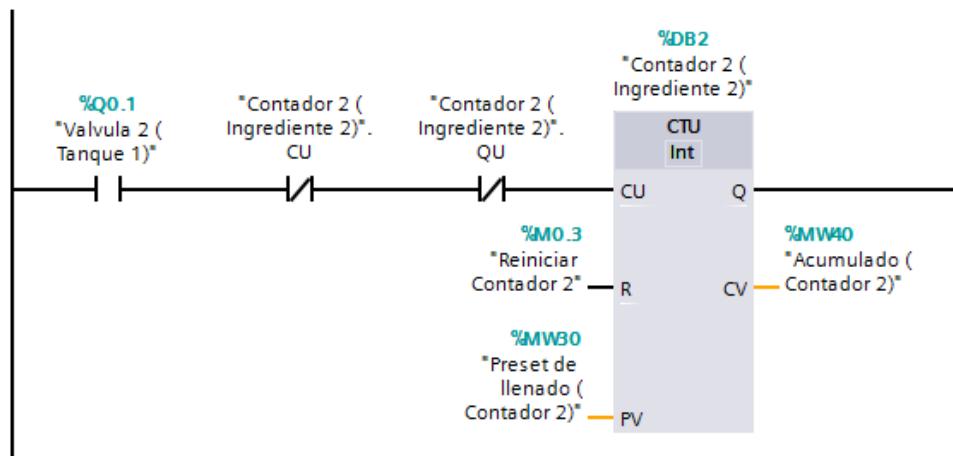


**Figura 88.** Segmento 4.

**Segmento 5:** contador que simula el llenado del tanque, asumiendo flujo constante, ver **Figura 89**.

▼ **Segmento 5:** Contador que simula el llenado del tanque a flujo constante (válvula 2 activa)

Lógica para activar, enclavar y parar el conteo.

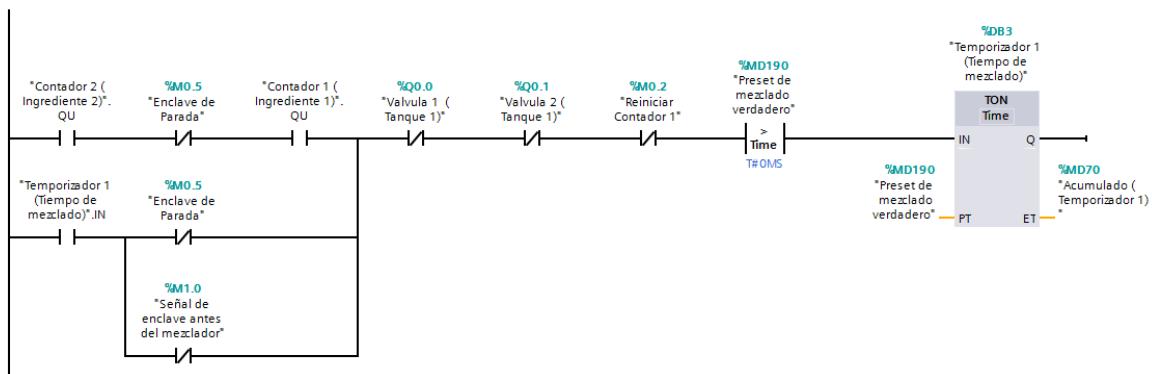


**Figura 89.** Segmento 4.

**Segmento 6:** temporización del tiempo de mezclado, ver **Figura 90.**

▼ **Segmento 6:** Temporización del tiempo de mezclado

Lógica para activar, enclavar y reiniciar el temporizador.

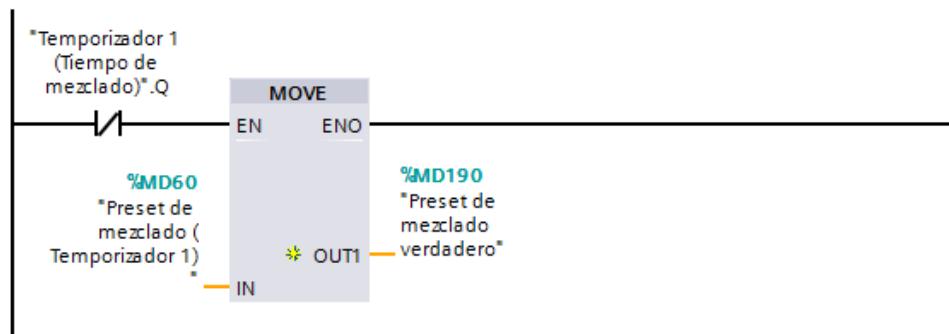


**Figura 90.** Segmento 6.

**Segmento 7:** lógica para la asignación del preset de mezclado en el temporizador (1), ver **Figura 91.**

▼ **Segmento 7:** Lógica para la asignación del preset de mezclado en el temporizador

Antes del mezclador

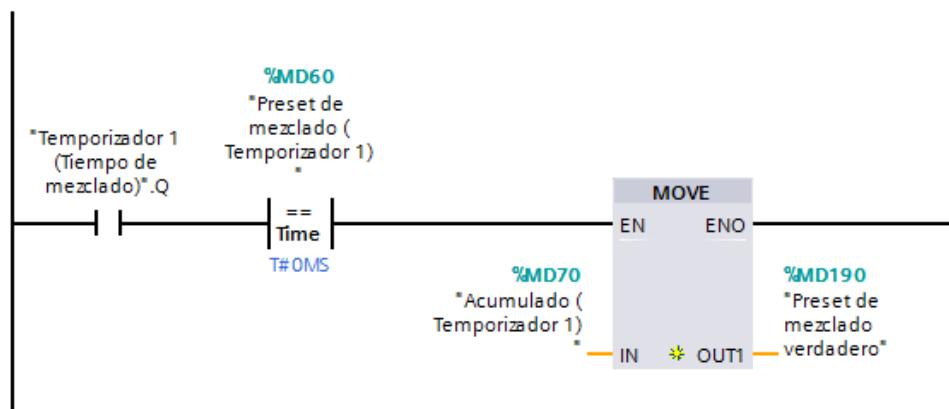


**Figura 91.** Segmento 7.

**Segmento 8:** lógica para la asignación del preset de mezclado en el temporizador (2), ver **Figura 92.**

▼ **Segmento 8:** Lógica para la asignación del preset de mezclado en el temporizador

▼ Despues del mezclador del mezclador y en caso de adquirir un valor de 0 se carga el valor acumulado.

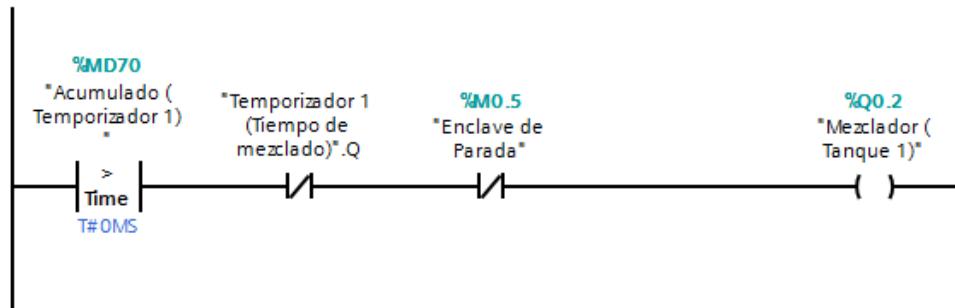


**Figura 92.** Segmento 8.

**Segmento 9:** mezclador, ver **Figura 93.**

▼ **Segmento 9:** Mezclador

- ▼ Condicionada la activación del mezclador para un valor mayor a 0 segundos. Si el temporizador esta ajustado en 0 segundos el proceso de detiene.



**Figura 93.** Segmento 9.

**Segmento 10:** mezclador, ver **Figura 94.**

▼ **Segmento 10:** Válvula 3 (vaciado) y bomba 1

Lógica para activar, parar (automaticamente y manualmente) a la válvula 3 y bomba 1.

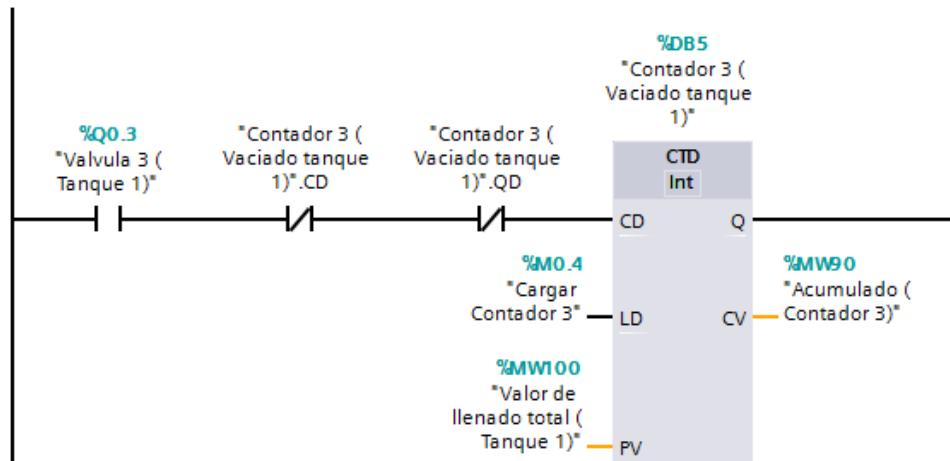


**Figura 94.** Segmento 10.

**Segmento 11:** contador que simula el vaciado del tanque a flujo constante (válvula 3 activa), ver **Figura 95.**

▼ **Segmento 11:** Contador que simula el vaciado del tanque a flujo constante (válvula 3 activa)

Logica para activar, enclavar y parar el conteo.

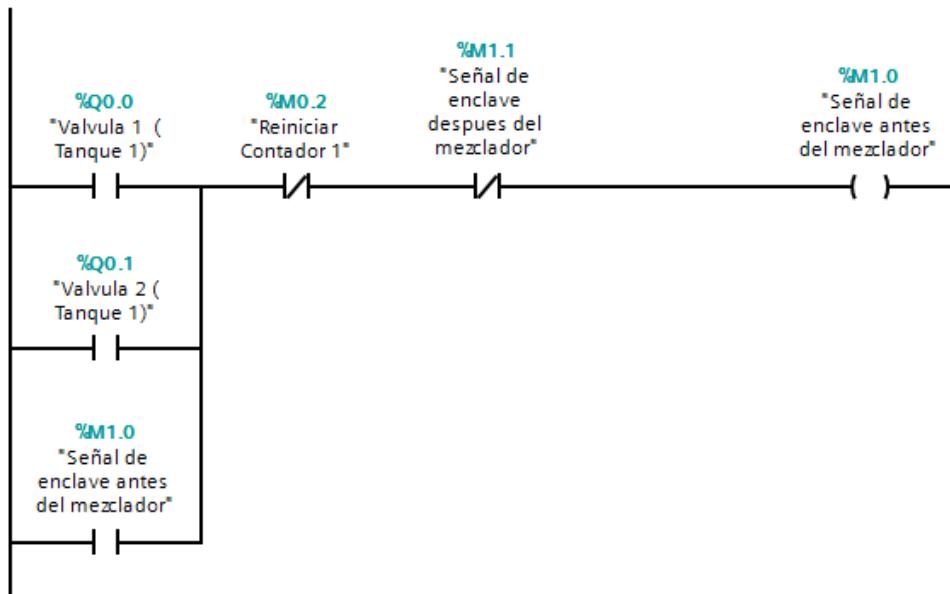


**Figura 95.** Segmento 11.

**Segmento 12:** señal de enclave antes del mezclador, ver **Figura 96.**

▼ **Segmento 12:** Señal de enclave antes del mezclador

► Enclave que define el estado en el que se encuentran las variables de temporización y conteo ant...

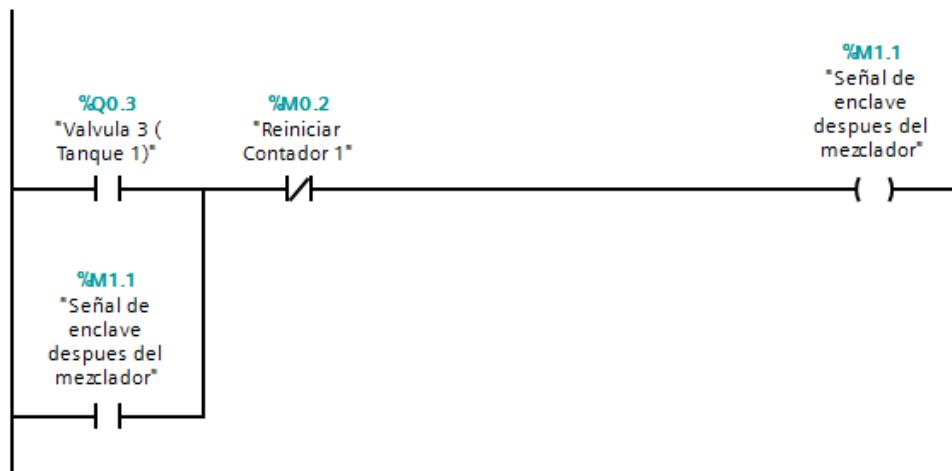


**Figura 96.** Segmento 12.

**Segmento 13:** señal de enclave después del mezclador, ver **Figura 97.**

▼ **Segmento 13:** Señal de enclave despues del mezclador

► Enclave que define el estado en el que se encuentran las variables de temporización y conteo des...

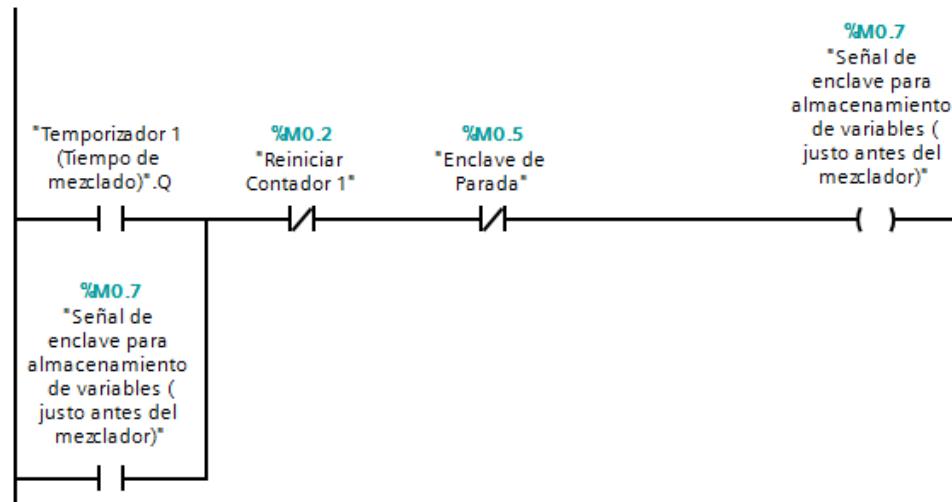


**Figura 97.** Segmento 13.

**Segmento 14:** señal de enclave para almacenamiento de variables justo antes del mezclador, ver **Figura 98**.

▼ **Segmento 14:** Señal de enclave para almacenamiento de variables justo antes del mezclador

► Enclave que define el estado en el que se encuentran las variables de temporización y conteo just...

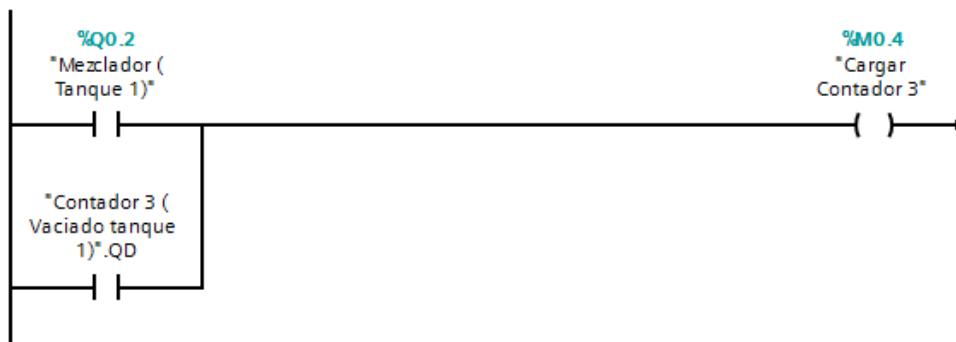


**Figura 98.** Segmento 14.

**Segmento 15:** carga de preset en el contador 3, ver **Figura 99**.

▼ Segmento 15: Carga de preset en el contador 3

Comentario

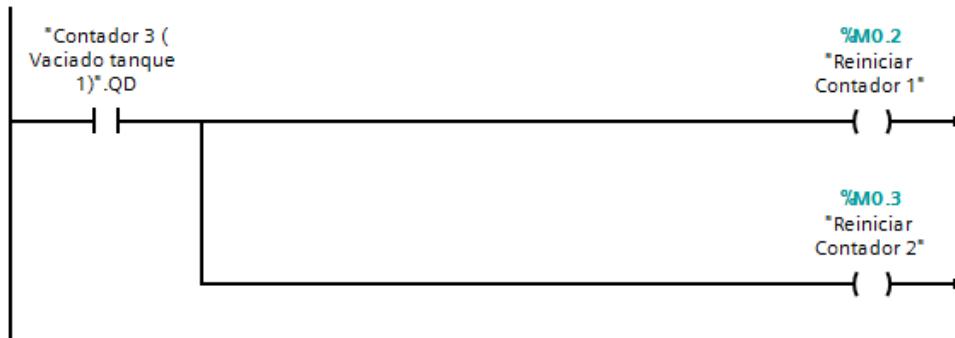


**Figura 99.** Segmento 15.

**Segmento 16:** reinicio del contador 1 y 2, ver **Figura 100**.

▼ Segmento 16: Reinicio del contador 1 y 2

► Finalizado el ciclo, se reinician los contadores 1 y 2

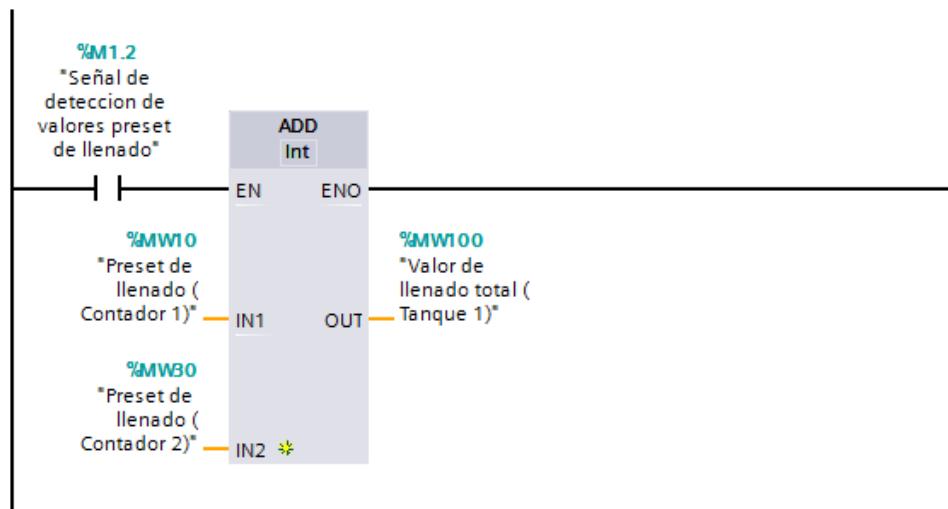


**Figura 100.** Segmento 16.

**Segmento 17:** suma y almacenamiento de presets de llenado, ver **Figura 101**.

▼ Segmento 17: Suma y almacenamiento de presets de llenado

Condición 1

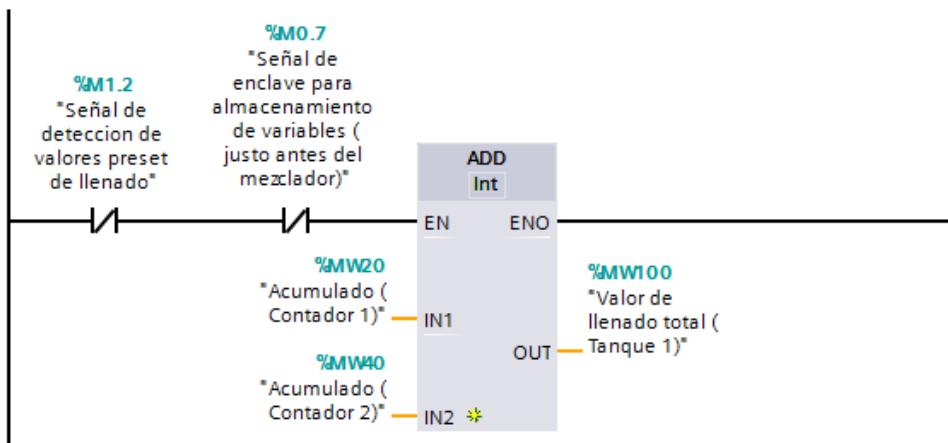


**Figura 101.** Segmento 17.

**Segmento 18:** suma y almacenamiento de acumulados, ver **Figura 102.**

▼ Segmento 18: Suma y almacenamiento de acumulados

Condición 2

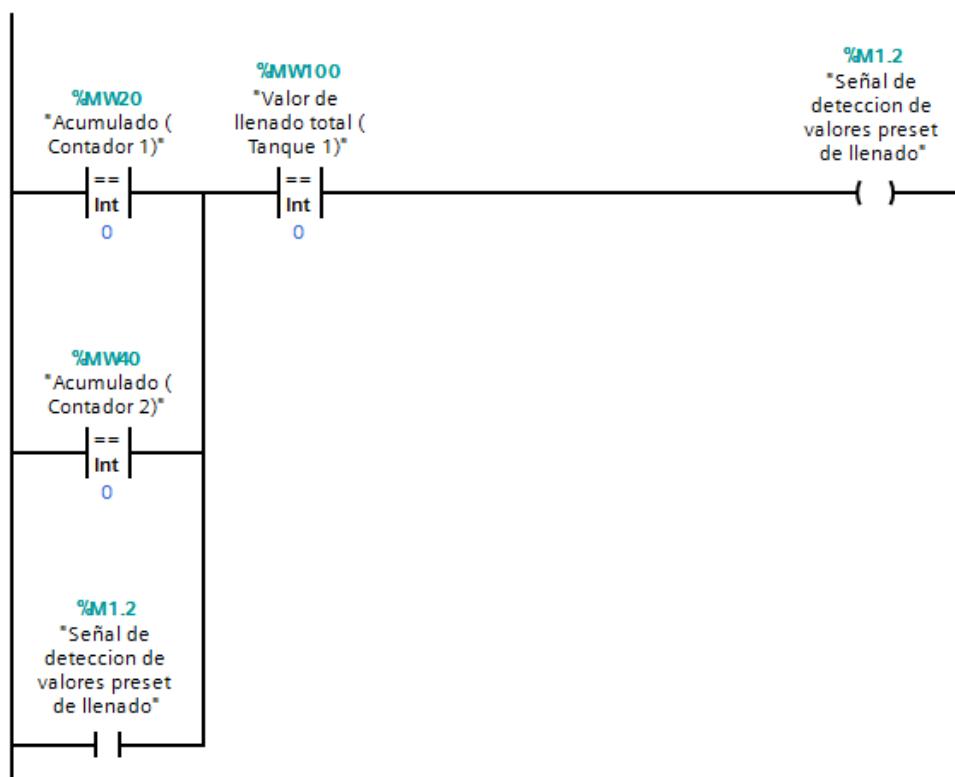


**Figura 102.** Segmento 18.

**Segmento 19:** señal de detección de valores de preset de llenado, ver **Figura 103.**

▼ **Segmento 19:** Señal de detección de valores de preset de llenado

lógica condicional para la determinación de valores de preset y arranque del proceso.

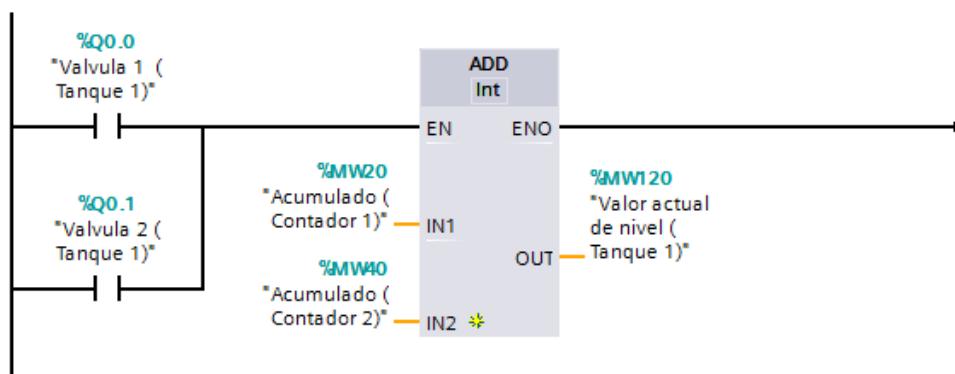


**Figura 103.** Segmento 19.

**Segmento 20:** valor de nivel actual (Con propósitos para la HMI) (1), ver **Figura 104**.

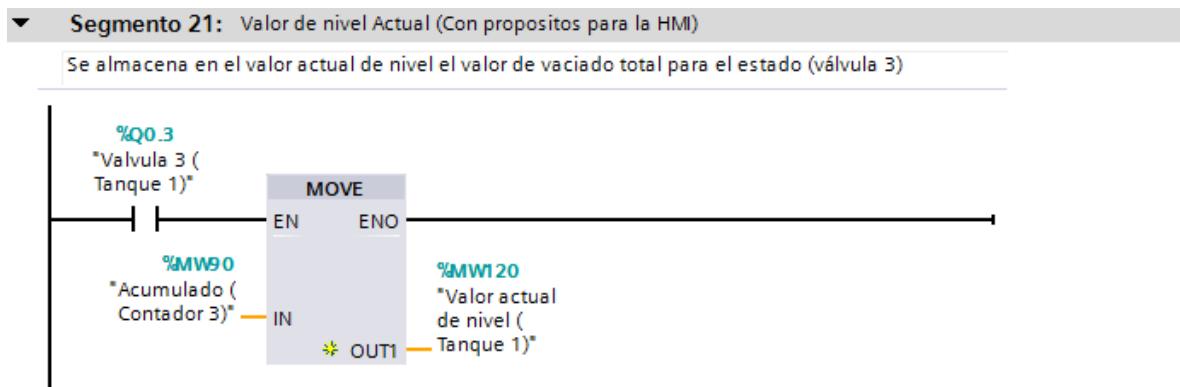
▼ **Segmento 20:** Valor de nivel Actual (Con propósitos para la HMI)

► Se almacena en el valor actual de nivel el valor de llenado total para los dos estados (válvula 1 y v...



**Figura 104.** Segmento 20.

**Segmento 21:** valor de nivel actual (Con propósitos para la HMI) (2), ver **Figura 105.**



**Figura 105.** Segmento 21.

Tags creados en el canal / dispositivo (Servidor OPC):

Tag Name	Address	Data Type	Scan Rate	Scaling	Description
Valvula 3	Q0.3	Boolean	100	None	
Valvula 2	Q0.1	Boolean	100	None	
Valvula 1	Q0.0	Boolean	100	None	
Valor actual de nivel (Tanque 1)	MW120	Word	100	None	
Preset de mezclado (Temporizador 1)	MD60	DWord	100	None	
Preset de llenado (Contador 2)	MW30	Word	100	None	
Preset de llenado (Contador 1)	MW10	Word	100	None	
Parada Remota	M0.1	Boolean	100	None	
Mezclador	Q0.2	Boolean	100	None	
Inicio Remoto	M0.0	Boolean	100	None	
Acumulado (Temporizador 1)	MD70	DWord	100	None	

**Figura 106.** Tags creados en el servidor OPC.

Proyecto creado en LabVIEW:

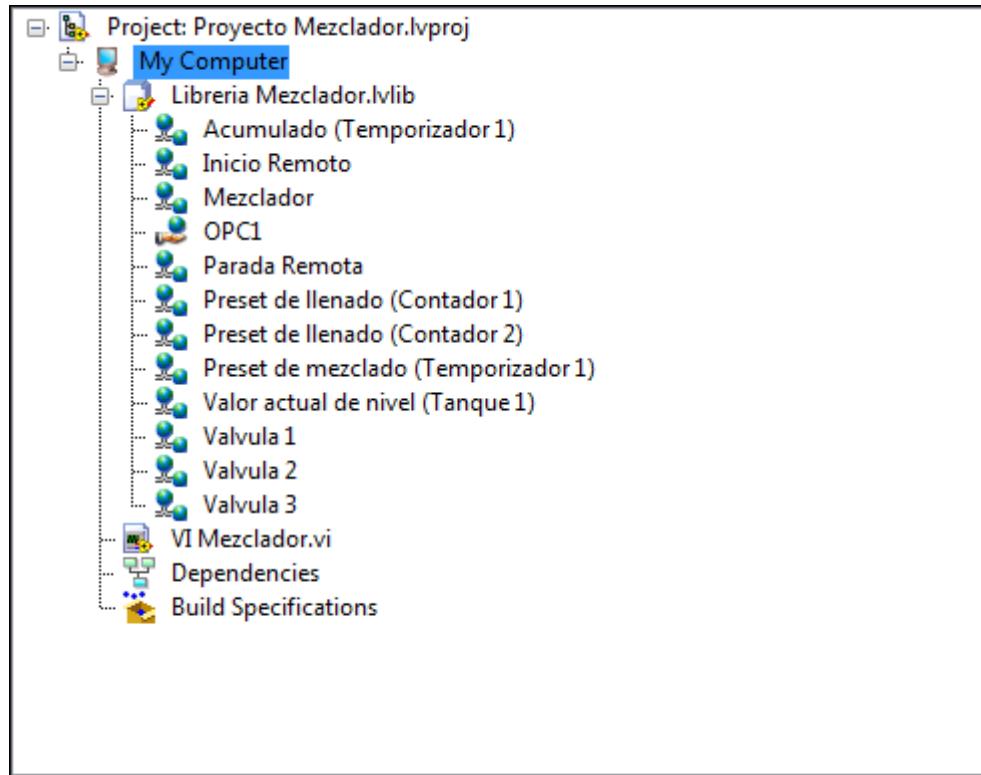
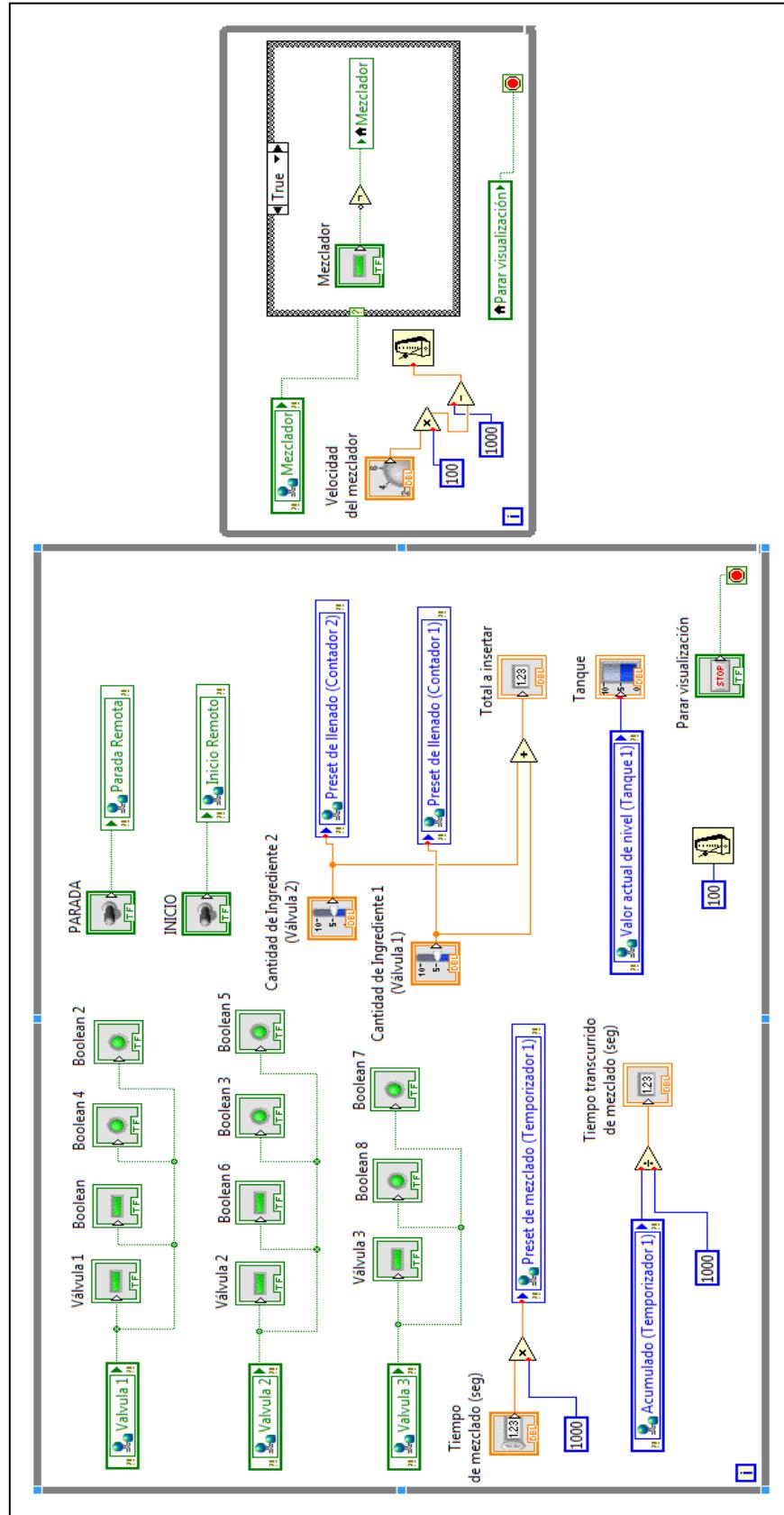
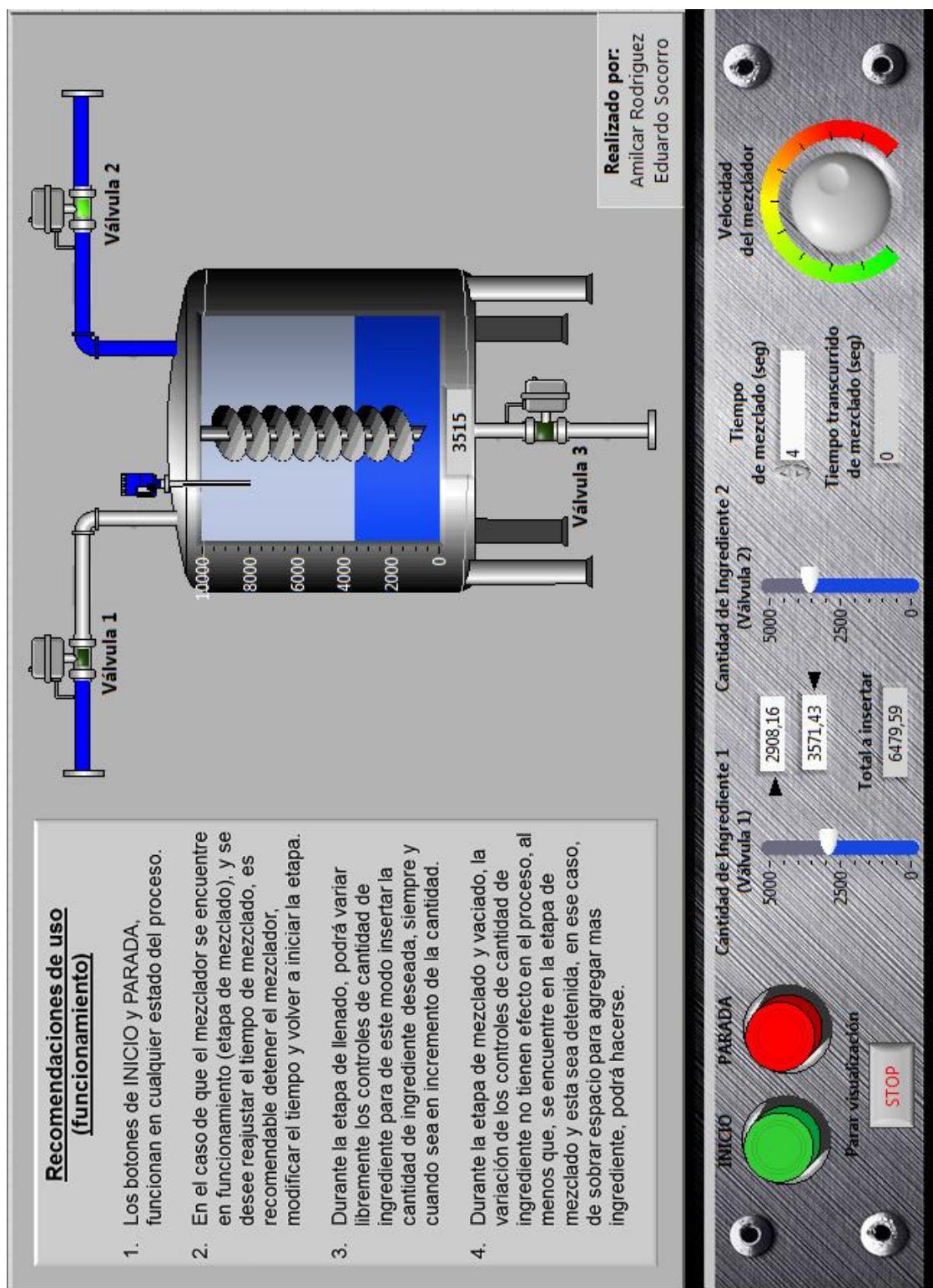


Figura 107. Proyecto diseñado en LabVIEW.



**Figura 108. Block Diagram de la interfaz.**



**Figura 109. HMI (VI) del mezclador.**

## **PRÁCTICA N° 3**

*Integración entre un PLC Allen Bradley ControlLogix y LabVIEW para el desarrollo de Interfaces Humano Maquina.*

## ÍNDICE GENERAL

	Pág.
1. OBJETIVO GENERAL.....	1
2. LISTA DE MATERIALES Y EQUIPOS .....	1
3. FUNDAMENTACIÓN TEÓRICA.....	1
3.1. PLC Allen Bradley ControlLogix .....	1
3.1.1. Módulos de E/S ControlLogix .....	2
3.1.2. Controladores ControlLogix .....	5
3.1.3. Fuente de alimentación.....	5
3.1.4. Lenguajes de programación .....	6
3.1.5. Instrucciones de básicas de programación.....	6
3.1.6. Tags y tipo de datos .....	13
3.2. Servidor OPC .....	15
3.3. Interfaz Humano Maquina (HMI) .....	15
4. PROCEDIMIENTO .....	17
4.1. Enlace entre Ordenador y PLC ControlLogix .....	17
4.2. Creación de un proyecto en RSLogix 5000 .....	19
4.3. Configuración del servidor OPC .....	29
4.4. Configuración de LabVIEW .....	40
4.5. Crear VI del proyecto .....	41
4.6. Adicionar al proyecto un temporizador y contador controlado y supervisado desde LabVIEW.....	43

## ÍNDICE DE FIGURAS

	Pág.
Figura 1. Sistema ControlLogix sencillo.....	2
Figura 2. TON (temporizador de retardo a la conexión).....	7
Figura 3. Diagrama de tiempo de un TON.....	7
Figura 4. TOF (temporizador de retardo a la desconexión). .....	8
Figura 5. Diagrama de tiempo de un TOF.....	9
Figura 6. CTU (contador ascendente).....	9
Figura 7. Diagrama de funcionamiento de un CTU.....	10
Figura 8. CTD (contador descendente).....	11
Figura 9. Diagrama de funcionamiento de un CTD.....	12
Figura 10. Esquema de funcionamiento de un OPC Server.....	15
Figura 11. RSLinx Classic con driver de comunicación configurado y hardware detectado. ....	18
Figura 12. Hardware detectado en el <i>backplane</i> del controlador.....	18
Figura 13. Ventana <i>New Controller</i> .....	19
Figura 14. Ventana <i>Controller Organizer</i> .....	20
Figura 15. Ventana <i>Select Module</i> .....	21
Figura 16. Ventana <i>Select Major Revision</i> .....	21
Figura 17. Ventana de propiedades del Módulo.....	22
Figura 18. Ventana <i>New Module</i> (configuración del módulo 1756-IA16/A).....	22
Figura 19. Ventana <i>Module Properties</i> (configuración del módulo 1756-IA16/A).....	23
Figura 20. Módulos de entradas y salidas configurados. ....	23
Figura 21. Crear nuevo <i>tag</i> .....	24
Figura 22. Creación del <i>tag Arranque_manual</i> .....	25
Figura 23. Creación del <i>tag Salida</i> .....	25
Figura 24. Programa realizado ( <i>Arranque_manual / Salida</i> ). .....	26
Figura 25. Descarga del programa al PLC (1).....	26
Figura 26. Descarga del programa al PLC (2).....	27

Figura 27. Parámetros de monitoreo del PLC.....	27
Figura 28. Observación del programa en <i>Run Mode</i> con entrada inactiva. ....	28
Figura 29. Observación del programa en <i>Run Mode</i> con entrada activa. ....	28
Figura 30. Lógica modificada.....	28
Figura 31. Ventana de identificación del canal ( <i>Identification</i> ).....	30
Figura 32. Ventana de selección del driver correspondiente al PLC ( <i>Device Driver</i> ).	30
Figura 33. Ventana de selección del adaptador de red del ordenador ( <i>Network Interface</i> ). ....	31
Figura 34. <i>Canal 1</i> agregado.....	31
Figura 35. Ventana de asignación del nombre del controlador ( <i>Name</i> ). ....	32
Figura 36. Ventana de selección del modelo de controlador ( <i>Model</i> ). ....	32
Figura 37. Dispositivos ControlLogix soportados por el NI OPC Server. ....	33
Figura 38. Ventana de identificación del dispositivo en la red (ID).....	33
Figura 39. Ventana de configuración de barrido de datos ( <i>Scan Mode</i> ). ....	34
Figura 40. Ventana parámetros de temporización ( <i>Timing</i> ). ....	34
Figura 41. Ventana <i>Auto-Demotion</i> .....	35
Figura 42. Ventana de creación de <i>tags</i> ( <i>Database Creation</i> ). ....	35
Figura 43. Ventana de parámetros de conexión ( <i>Logix Communications Parameters</i> ). ....	36
Figura 44. Ventana opciones varias ( <i>Logix Options</i> ).....	36
Figura 45. Ventana de importación de <i>tags</i> ( <i>Logix Database Setting</i> ). ....	37
Figura 46. Ventana de resumen ( <i>Summary</i> ). ....	37
Figura 47. Selecciona de <i>Properties</i> para la posterior importación de <i>tags</i> . ....	38
Figura 48. Importar el archivo “.L5K”.....	39
Figura 49. Creación de los <i>tags</i> (importados) .....	39
Figura 50. <i>Tags</i> creados. ....	40
Figura 51. Proyecto configurado con su respectiva librería, variables y Cliente OPC. ....	41
Figura 52. <i>Block Diagram</i> del VI.....	42
Figura 53. <i>Front Panel</i> del VI. ....	42

Figura 54. Inserción del temporizador <i>Temporizador_1</i> y cambio de segmento para <i>Salida</i> .....	43
Figura 55. Re direccionado del contacto de auto enganche en <i>Temporizador_1.EN</i> y colocación del contacto <i>Temporizador_1.DN</i> .....	44
Figura 56. Adición de contacto asociado a <i>Salida</i> y <i>Contador_1</i> .....	44
Figura 57. Tag tipo base <i>Reser_contador_1</i> e instrucción de reset <i>Contador_1</i> . ....	45
Figura 58. Programa con temporización y conteo de la activación de <i>Salida</i> . .....	45
Figura 59. Carpetas contenedoras de tags <i>Contador_1</i> y <i>Temporizador_1</i> agregadas al <i>Prgm_MainProgram</i> . .....	46
Figura 60. Tags contenidos en <i>Contador_1</i> .....	46
Figura 61. Tags contenidos en <i>Temporizador_1</i> .....	47
Figura 62. Monitoreo de todos los tags del <i>Prgm_MainProgram</i> . .....	47
Figura 63. Nuevas variables añadidas a la librería. .....	48
Figura 64. <i>Block Diagram</i> del VI con las etapas de temporización y conteo añadidas.	49
Figura 65. <i>Front Panel</i> del VI con las etapas de temporización y conteo añadidas. ..	49

## ÍNDICE DE TABLAS

	Pág.
Tabla 1. Parámetros del TON.....	7
Tabla 2. Estructura del TON.....	7
Tabla 3. Parámetros del TOF.....	8
Tabla 4. Estructura del TOF.....	8
Tabla 5. Parámetros del CTU.....	10
Tabla 6. Estructura del CTU.....	10
Tabla 7. Parámetros del CTD.....	11
Tabla 8. Estructura del CTD.....	11
Tabla 9. Funcionamiento de la instrucción RES.....	12
Tabla 10. Clasificación de tipos de <i>tags</i> .....	13
Tabla 11. Tipo de datos en el controlador ControlLogix.....	14
Tabla 12. Tamaños de datos en el controlador ControlLogix.....	14

## 1. OBJETIVO GENERAL

Comunicar el PLC Allen Bradley ControlLogix con el programa LabVIEW para el control y supervisión de variables de proceso.

## 2. LISTA DE MATERIALES Y EQUIPOS

- Ordenador con los programa TIA Portal, National Instruments OPC Server y LabVIEW con los módulos Datalogging and Supervisory Control (DSC) y Real Time integrados.
- PLC Allen Bradley ControlLogix 5555 1756-L55.
- Cable UTP RJ45/RJ45.

## 3. FUNDAMENTACIÓN TEÓRICA

### 3.1. PLC Allen Bradley ControlLogix

El sistema ControlLogix proporciona control secuencial de proceso, de movimiento y de variador además de comunicaciones y E/S avanzadas en un pequeño y económico paquete. El sistema es modular, lo que permite el diseño, la construcción y la modificación eficaz para ahorrar los gastos necesarios para la captación e ingeniería. Un sistema ControlLogix sencillo consiste en un controlador autónomo y módulos de E/S en un solo chasis.



**Figura 1.** Sistema ControlLogix sencillo.

### 3.1.1. Módulos de E/S ControlLogix

Los Módulos de entrada y salida trabajan directamente con el controlador y se definen de la siguiente manera:

- Módulos de entrada

Un módulo de entrada responde a la señal de entrada del modo siguiente:

- ✓ Los filtros de entrada limitan los fenómenos transitorios de voltaje debidos a rebotes de contactos y/o el ruido eléctrico. Si no se han filtrado, los efectos de los fenómenos transitorios de voltaje pueden producir datos falsos. Todos los módulos de entrada usan filtros de entrada.
- ✓ El aislamiento óptico protege los circuitos lógicos contra posibles daños debidos a fenómenos transitorios eléctricos.
- ✓ Los circuitos lógicos procesan la señal.

- ✓ Un indicador LED que se activa y desactiva indica el estado del dispositivo de salida correspondiente.

- Módulos de Salida

Un módulo de salida controla la señal de salida del modo siguiente:

- ✓ Los circuitos lógicos determinan el estado de salida.
- ✓ Un indicador LED de salida indica el estado de la señal de salida.
- ✓ El aislamiento óptico separa la lógica de módulo y los circuitos de bus de la alimentación de campo.
- ✓ El controlador de salida activa y desactiva la salida correspondiente.

La línea ControlLogix abarca un amplio rango de módulos de entradas y salidas para diversas aplicaciones, desde aplicaciones discretas de alta velocidad hasta el control de procesos.

Los Módulos de E/S ControlLogix que se requieran ocupar se deben montar en un Chasis ControlLogix y se necesita un RTB (bloque de terminales extraíble) o un IFM 1492 (módulo de interface), para conectar el alambrado de campo.

Existen varios tipos de módulos de E/S, así como los módulos para conectar sensores y la ubicación de los módulos, y se listan a continuación con una breve acotación.

- Módulos de E/S digitales 1756

Existen:

- ✓ Módulos de entrada digital AC.
- ✓ Módulos de salida digital AC.
- ✓ Módulos de entrada digital DC.
- ✓ Módulos de salida digital DC.
- ✓ Módulos de salida de contacto digital.

Los módulos de E/S digitales 1756 soportan:

- ✓ Variedad de capacidades de interface de voltaje.
- ✓ Tipos de módulo aislados y no aislados.
- ✓ Estados de fallo de salida de nivel de punto.
- ✓ Posibilidad de seleccionar comunicaciones de rack optimizadas o de conexión directa diagnósticos de lado de campo en determinados módulos.
- Módulos de E/S Análogos

Los módulos de E/S análogos soportan:

- ✓ Alarma de datos incorporada.

- ✓ Escalado a unidades de ingeniería.
- ✓ Muestreo de canal en tiempo real.
- ✓ Formatos de datos enteros de 16 bits o punto flotante de 32 bits IEEE.

La mayoría de módulos análogos son de entrada, sólo algunos vienen con entrada y salida integrada.

### 3.1.2. Controladores ControlLogix

Los PLCs ControlLogix proporcionan una solución de controlador escalable que es capaz de direccionar una gran cantidad de puntos de E/S (128,000 digitales máx. / 4000 analógicos máx.). El ControlLogix puede colocarse en cualquier ranura de un chasis de E/S ControlLogix y pueden instalarse múltiples controladores en el mismo chasis. Múltiples controladores en el mismo chasis se comunican entre sí mediante el *backplane*, pero funcionan independientemente. Los controladores ControlLogix pueden monitorear y controlar E/S a través del *backplane* ControlLogix así como mediante vínculos de E/S. Los controladores ControlLogix pueden comunicarse con computadoras u otros procesadores a través de RS-232-C (protocolo DF1/DH-485) y las redes DeviceNet, DH+, ControlNet y EtherNet/IP.

### 3.1.3. Fuente de alimentación

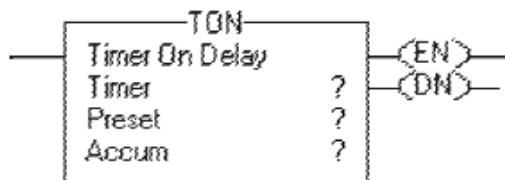
Es la unidad que suministra la energía para alimentar el *backplane* donde van empotrados los módulos de entrada, salida, módulos de comunicación y CPU. En algunas fuentes pueden tener diferentes derivaciones en cuanto la salida de su voltaje y/o amperaje dependiendo de las necesidades del usuario y/o su aplicación.

### 3.1.4. Lenguajes de programación

Existen varios lenguajes compatibles con el controlador ControlLogix, entre los cuales se tienen, Escalera de relé (esquema de contactos), texto estructurado, bloque de funciones y SFC. Para efectos de este material instruccional se implementará exclusivamente el lenguaje escalera de relé.

### 3.1.5. Instrucciones de básicas de programación

- Instrucciones de bit
  - ✓ **Contacto normalmente abierto XIC** (→ [ ]): la instrucción XIC examina el bit de datos para determinar si está establecido.
  - ✓ **Contacto normalmente cerrado XIO** (→/[ ]): la instrucción XIO examina el bit de datos para determinar si esta borrado.
  - ✓ **Activación de salida o bobina no retentiva OTE** (←○○): la instrucción OTE establece o borra el bit de datos.
- Instrucciones de temporización
  - ✓ **Temporizador de retardo a la conexión (TON)**: la instrucción TON es un temporizador no retentivo que acumula el tiempo cuando la instrucción está habilitada, (la condición de entrada del renglón es verdadera), en la **Figura 2** puede observarse el bloque de temporización TON.



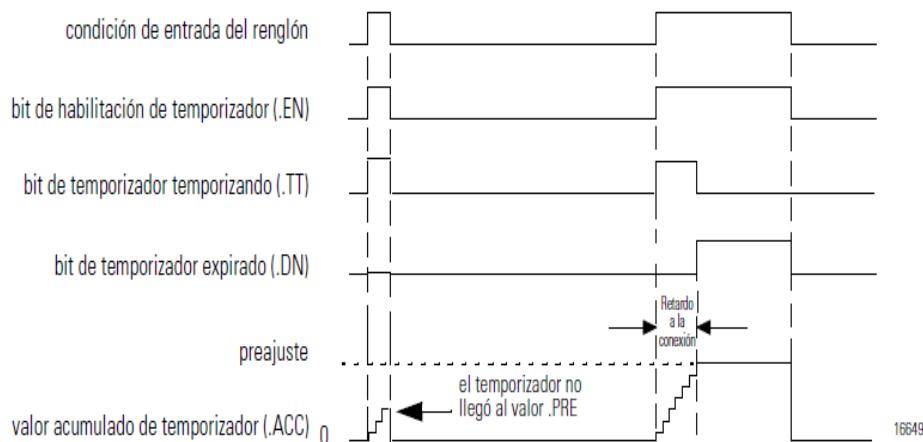
**Figura 2.** TON (temporizador de retardo a la conexión).

**Tabla 1.** Parámetros del TON.

Operando	Tipo	Formato	Descripción
Timer	TIMER	tag	estructura de temporizador
Preset	DINT	inmediato	período de retardo (acumulación de tiempo)
Acum	DINT	inmediato	el tiempo en ms contado por el temporizador el valor inicial es típicamente 0

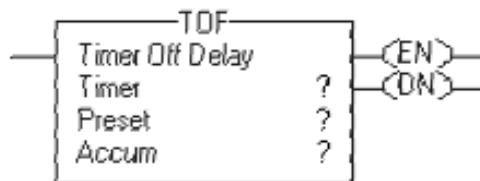
**Tabla 2.** Estructura del TON.

Mnemónico	Tipo de datos	Descripción
.EN	BOOL	El bit de habilitación indica que la instrucción TON está habilitada.
.TT	BOOL	El bit de temporización indica que se está ejecutando una operación de temporización.
.DN	BOOL	El bit de efectuado se establece cuando .ACC ≥ .PRE.
.PRE	DINT	El valor de preajuste especifica el valor (en unidades de 1 ms) al que debe llegar el valor acumulado para que la instrucción establezca el bit .DN.
.ACC	DINT	El valor acumulado especifica el número de milisegundos que han transcurrido desde que se habilitó la instrucción TON.



**Figura 3.** Diagrama de tiempo de un TON.

- ✓ **Temporizador de retardo a la desconexión (TOF):** la instrucción TOF es un temporizador no retentivo que acumula el tiempo cuando la instrucción está habilitada (la condición de entrada del renglón es falsa). En la **Figura 4** puede observarse el bloque de temporización TON.



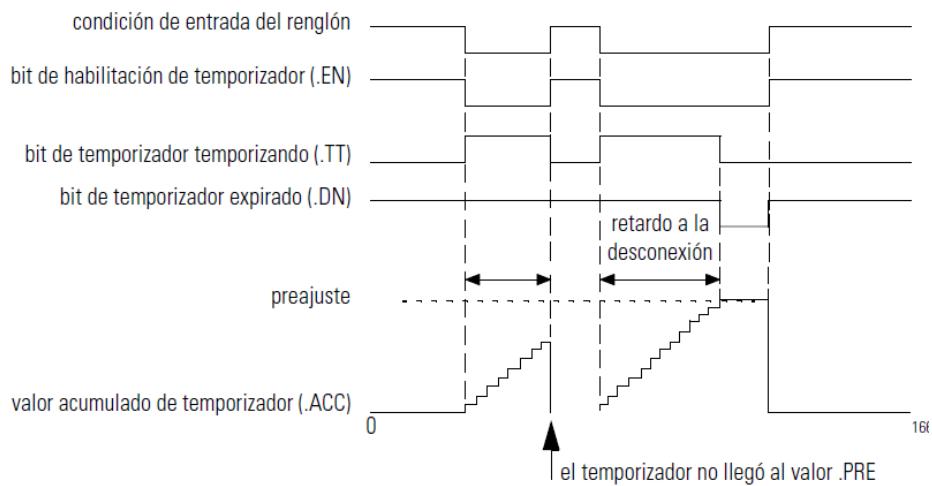
**Figura 4.** TOF (temporizador de retardo a la desconexión).

**Tabla 3.** Parámetros del TOF.

Operando	Tipo	Formato	Descripción
Timer	TIMER	tag	estructura de temporizador
Preset	DINT	inmediato	período de retardo (acumulación de tiempo)
Acum	DINT	inmediato	total de ms que contó el temporizador el valor inicial es típicamente 0

**Tabla 4.** Estructura del TOF.

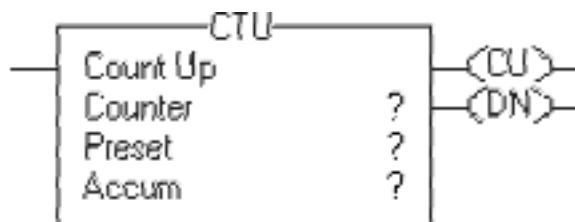
Mnemónico	Tipo de datos	Descripción
.EN	BOOL	El bit de habilitación indica que la instrucción TOF está habilitada.
.TT	BOOL	El bit de temporización indica que se está ejecutando una operación de temporización.
.DN	BOOL	El bit de efectuado se borra cuando .ACC ≥ .PRE.
.PRE	DINT	El valor de preajuste especifica el valor (en unidades de 1 ms) al que debe llegar el valor acumulado para que la instrucción borre el bit .DN.
.ACC	DINT	El valor acumulado especifica el número de milisegundos que han transcurrido desde que se habilitó la instrucción TOF.



**Figura 5.** Diagrama de tiempo de un TOF.

- Instrucciones de conteo

✓ **Contador ascendente (CTU):** cuando se habilita el contador, la instrucción CTU incrementa el contador en uno, alcanzado el valor de cuenta preestablecido se habilita el bit .DN, la cuenta puede seguir incrementándose hasta que se reinicie mediante la instrucción RES o se edite en el mismo bloque del CTU.



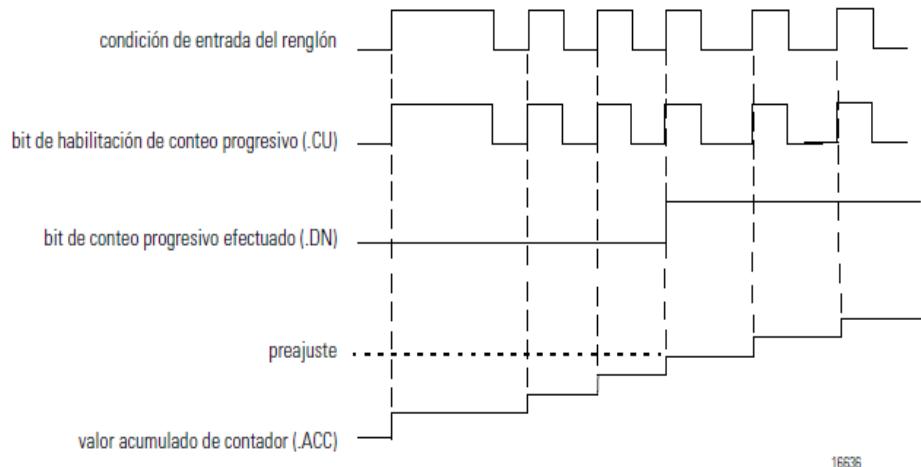
**Figura 6.** CTU (contador ascendente).

**Tabla 5.** Parámetros del CTU.

Operando	Tipo	Formato	Descripción
Counter	COUNTER	tag	estructura de contador
Preset	DINT	inmediato	valor superior hasta donde contar
Acum	DINT	inmediato	número de veces que contó el contador el valor inicial es típicamente 0

**Tabla 6.** Estructura del CTU.

Mnemónico	Tipo de datos	Descripción
.CU	BOOL	El bit de habilitación de conteo progresivo indica que la instrucción CTU está habilitada.
.DN	BOOL	El bit de efectuado indica que .ACC $\geq$ .PRE.
.OV	BOOL	El bit de overflow indica que el contador excedió el límite superior de 2,147,483,647, por lo que el contador regresa a -2,147,483,648 y comienza nuevamente el conteo progresivo.
.UN	BOOL	El bit de underflow indica que el contador excedió el límite inferior de -2,147,483,648, por lo que el contador regresa a 2,147,483,647 y comienza nuevamente el conteo regresivo.
.PRE	DINT	El valor de preajuste especifica el valor al que debe llegar el valor acumulado para que la instrucción establezca el bit .DN.
.ACC	DINT	El valor acumulado especifica el número de transiciones que contó la instrucción.

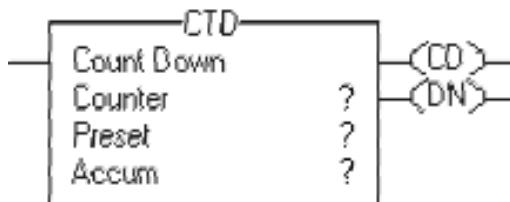


16636

**Figura 7.** Diagrama de funcionamiento de un CTU.

- ✓ **Contador descendente (CTD):** la instrucción CTD cuenta regresivamente, normalmente se usa con una instrucción CTU que referencia la misma estructura de contador. Cuando se habilita y el bit .CD se borra, la instrucción CTD decrementa el contador en uno. Cuando se habilita y el bit .CD se

establece, o cuando se inhabilita la instrucción CTD retiene su valor .ACC. El valor acumulado continúa decrementándose, aun después de que se establece el bit .DN. Para borrar el valor acumulado, use una instrucción RES o escriba cero (0) en el valor acumulado.



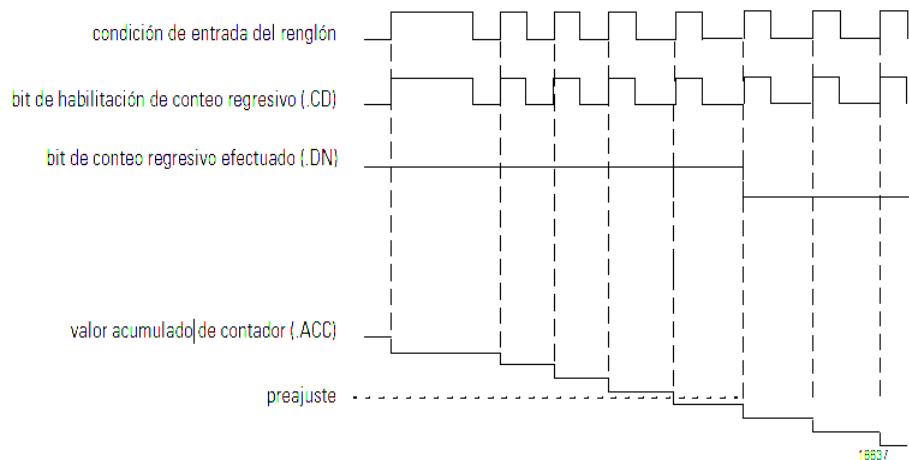
**Figura 8.** CTD (contador descendente).

**Tabla 7.** Parámetros del CTD.

Operando	Tipo	Formato	Descripción
Counter	COUNTER	tag	estructura de contador
Preset	DINT	inmediato	valor inferior hasta donde contar
Acum	DINT	inmediato	número de veces que contó el contador el valor inicial es típicamente 0

**Tabla 8.** Estructura del CTD.

Mnemónico	Tipo de datos	Descripción
.CD	BOOL	El bit de habilitación de conteo regresivo indica que la instrucción CTD está habilitada.
.DN	BOOL	El bit de efectuado indica que .ACC $\geq$ .PRE.
.OV	BOOL	El bit de overflow indica que el contador excedió el límite superior de 2,147,483,647, por lo que el contador regresa a -2,147,483,648 y comienza nuevamente el conteo progresivo.
.UN	BOOL	El bit de underflow indica que el contador excedió el límite inferior de -2,147,483,648, por lo que el contador regresa a 2,147,483,647 y comienza nuevamente el conteo regresivo.
.PRE	DINT	El valor de preajuste especifica el valor al que debe llegar el valor acumulado para que la instrucción establezca el bit .DN.
.ACC	DINT	El valor acumulado especifica el número de transiciones que contó la instrucción.



**Figura 9.** Diagrama de funcionamiento de un CTD.

- **Instrucción restablecer RES (–RES–):** RES restablece una estructura TIMER, COUNTER o CONTROL.

**Tabla 9.** Funcionamiento de la instrucción RES.

Cuando se usa una instrucción RES para	La instrucción borra
TIMER	el valor .ACC
	los bits de estado de control
COUNTER	el valor .ACC
	los bits de estado de control
CONTROL	el valor .POS
	los bits de estado de control

### 3.1.6. Tags y tipo de datos

- Tags

Los *tags* o etiquetas son el método para asignar y hacer referencia a posiciones de memoria en Allen Bradley controladores Logix5000, dependiendo de la aplicación o uso pueden clasificarse como es mostrado en la **Tabla 10**.

**Tabla 10.** Clasificación de tipos de *tags*.

Si desea que el tag	Entonces seleccione este tipo
Almacene uno o varios valores para que la lógica los use en el proyecto	Base
Represente otro tag	Alias
Envíe datos a otro controlador	Producido
Reciba datos de otro controlador	Consumido

- Tipo de datos

Un tipo de datos es una definición del tamaño y el diseño de memoria asignada a un *tag* o etiqueta creada. Los tipos de datos definen el número de bits, bytes o palabras de datos de una etiqueta a utilizar, en la **Tabla 11** se encuentran especificados los tipos de datos manejados por el sistema ControlLogix.

**Tabla 11.** Tipo de datos en el controlador ControlLogix.

Para	Seleccione
Dispositivo analógico en modo con punto flotante	REAL
Dispositivo analógico en modo con número entero (para frecuencias de muestreo muy rápidas)	INT
Caracteres ASCII	Cadena
Bit	BOOL
Contador	CONTADOR
Punto de E/S digital	BOOL
Número con punto flotante	REAL
Número entero	DINT
Secuenciador	CONTROL
Temporizador	TIMER

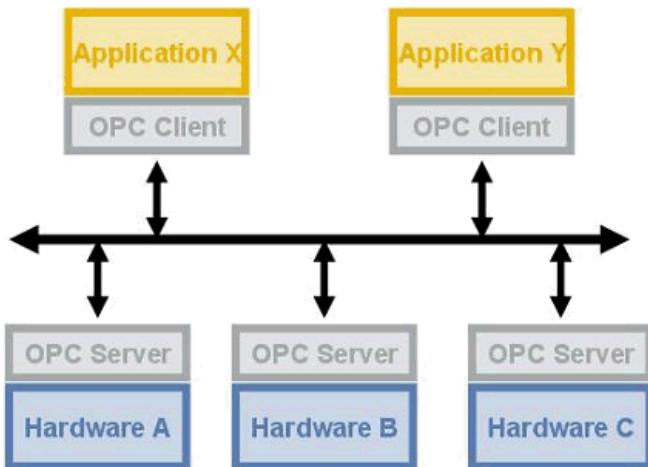
La asignación mínima de memoria para un *tag* es de cuatro (4) bytes. Cuando se crea un *tag* con datos que requieren menos de cuatro (4) bytes, el controlador asigna cuatro (4) bytes pero los datos solo ocupan la parte que necesitan.

**Tabla 12.** Tamaños de datos en el controlador ControlLogix.

Tipo de datos	Bits							
	31	16	15	8	7	1	0	
BOOL	no se usa							0 o 1
SINT	no se usa							-128...+127
INT	no se usa							-32,768...+32,767
DINT								-2,147,483,648...+2,147,483,647
REAL								-3.40282347E <sup>38</sup> ...-1.17549435E <sup>38</sup> (valores negativos)
								0
								1.17549435E <sup>38</sup> ...3.40282347E <sup>38</sup> (valores positivos)

### 3.2. Servidor OPC

Un servidor OPC es una aplicación de software (driver) que cumple con una o más especificaciones definidas por la OPC Foundation. El servidor OPC hace una interfaz comunicando por un lado con una o más fuentes de datos utilizando sus protocolos nativos (típicamente PLCs, DCSs, basculas, Módulos I/O, controladores, etc.) y por otro lado con clientes OPC (típicamente SCADAs, HMIs, generadores de informes, generadores de gráficos, aplicaciones de cálculos, etc.). En una arquitectura Cliente OPC/Servidor OPC, el servidor OPC es el esclavo mientras que el cliente OPC es el maestro. Las comunicaciones entre cliente OPC y el servidor OPC son bidireccionales, lo que significa que los clientes pueden leer y escribir en los dispositivos a través del servidor OPC.



**Figura 10.** Esquema de funcionamiento de un OPC Server.

### 3.3. Interfaz Humano Maquina (HMI)

HMI significa “*Human Machine Interface*” (Interfaz Humano Maquina), es decir, es el dispositivo o sistema que permite la interfaz entre el humano y la máquina. Tradicionalmente estos sistemas consistían en paneles compuestos por indicadores y

comandos, tales como luces pilotos indicadores digitales y análogos, registradores, pulsadores, selectores y otros que se interconectaban con la maquina o proceso. En la actualidad, dado que las máquinas y procesos en general están implementadas con controladores y otros dispositivos electrónicos que dejan disponibles puertas de comunicación, es posible contar con sistemas HMI más poderosos y eficaces, además de permitir una conexión más sencilla y económica con el proceso o maquinas.

## 4. PROCEDIMIENTO

### 4.1. Enlace entre Ordenador y PLC ControlLogix

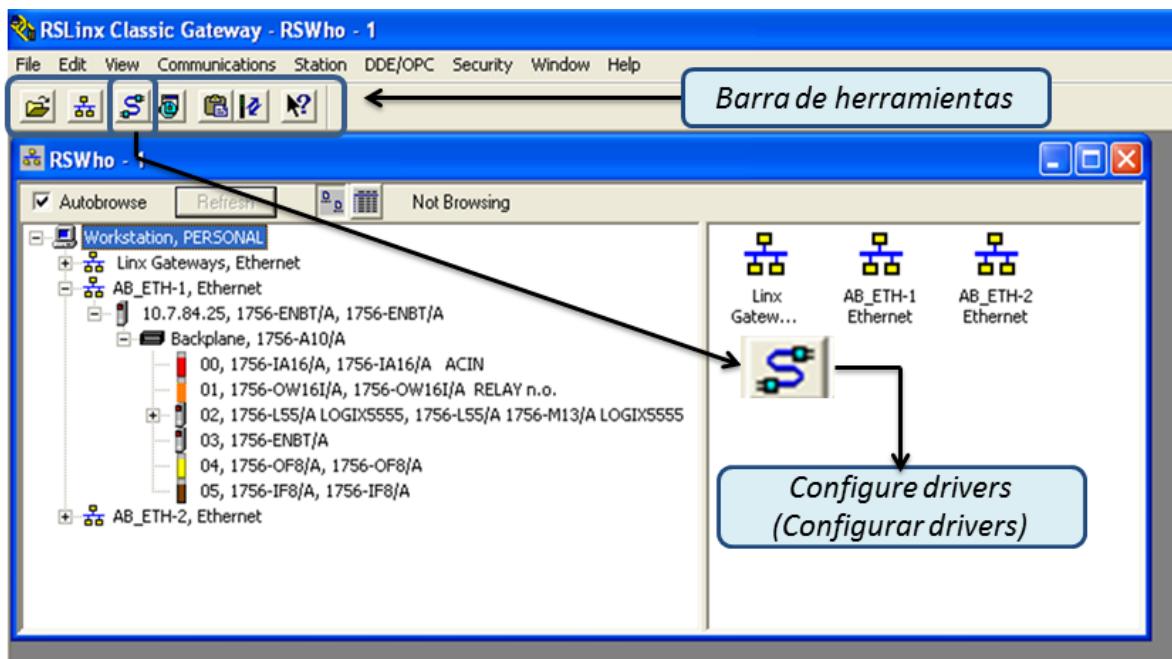
Para programar el PLC ControlLogix es necesario primero que todo establecer comunicación entre el ordenador y el PLC, para hacer esto se requiere de la implementación del servidor de comunicación RSLinx, por lo tanto se explicará el manejo básico de este programa orientado al enlace entre el ordenador y el autómata.

En primer lugar, cerciórese que el módulo de comunicaciones Ethernet/IP 1756-ENBT/A se encuentre conectado al *backplane* del controlador, hecho esto último, encienda el PLC, espere unos segundos y podrá observar la dirección IP del módulo de comunicaciones a través del *display* presente en el módulo, dicha IP se encuentra configurada en este caso particular en **10.7.84.25**.

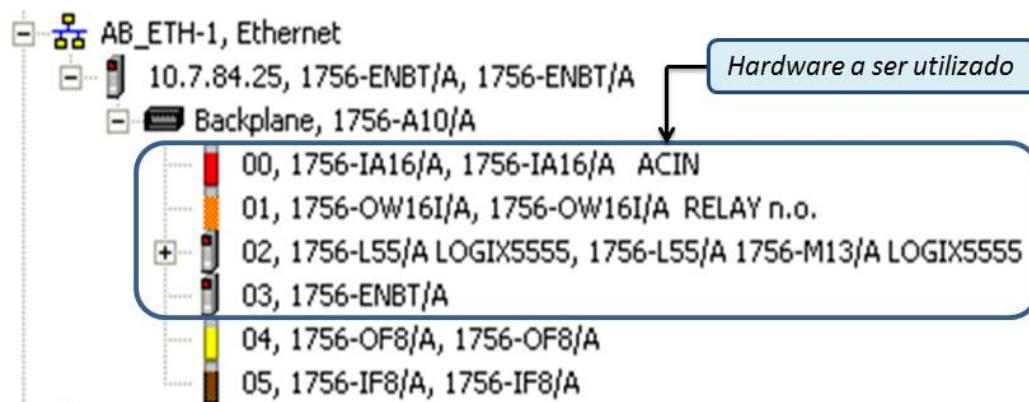
Conecte el PLC (Módulo Ethernet/IP) y el ordenador mediante el cable UTP RJ45/RJ45.

Ejecute el RSLinx Classics, abierto el programa pulse el botón *Configure drivers* presente en la *Barra de herramientas*, inmediatamente se desplegará una ventana llamada *Configure Drivers*, en ella se llevará a cabo la configuración del driver requerido para ejecutar el enlace. En el campo en blanco *Available Driver Types* seleccione *Ethernet devices* y pulse el botón *Add New*, configure el driver insertando el nombre, se dejará en este caso por defecto *AB\_ETH-1* y posteriormente la dirección IP del controlador en el campo *Host Name* que en este caso es 10.7.84.25, luego pulse *Aceptar* y seguidamente *Close*, con esto último se ha configurado exitosamente el driver, vea la **Figura 11** con el driver ya ajustado y por consiguiente detección del hardware del controlador.

En las **Figuras 11** y **Figura 12** específicamente esta última, puede observarse el hardware detectado (módulos) en el *backplane* del PLC, para efectos de este material instruccional solo se hará uso de cuatro (4) de los seis (6) módulos presentes en el controlador: el módulo de la CPU (1756-L55/A), el módulo de comunicaciones (1756-ENBT/A) y los módulos de entradas digitales (1756-IA16/A) y salidas *relay* (1756-OW16I/A).



**Figura 11.** RSLinx Classic con driver de comunicación configurado y hardware detectado.

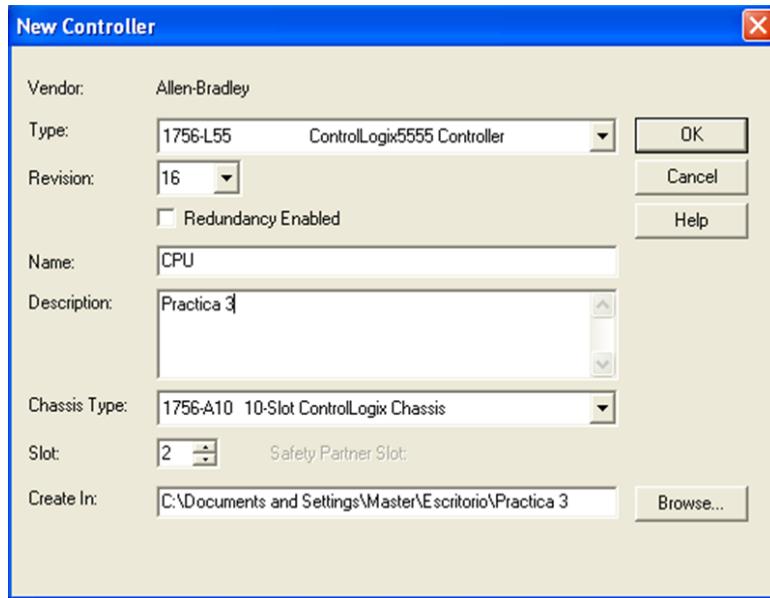


**Figura 12.** Hardware detectado en el *backplane* del controlador.

## 4.2. Creación de un proyecto en RSLogix 5000

Ejecute el RSLogix 5000, posteriormente cree un nuevo controlador siguiendo la ruta *File / New*, instantáneamente surgirá una ventana llamada *New Controller* (**Figura 13**).

En la ventana *New Controller* se configurará lo referente al módulo de CPU y chasis o *backplane* del controlador. Seleccione en el campo *Type* el modelo de CPU, en este caso se trata del “1756-L55/A”, después escoja la *Revision* (16), el *Slot* en el que se encuentra el módulo CPU (2) y el modelo del *Chassis Type* (1756-A10), hecho esto seleccione “OK”, en la **Figura 13** aparecen hechas dichas configuraciones.



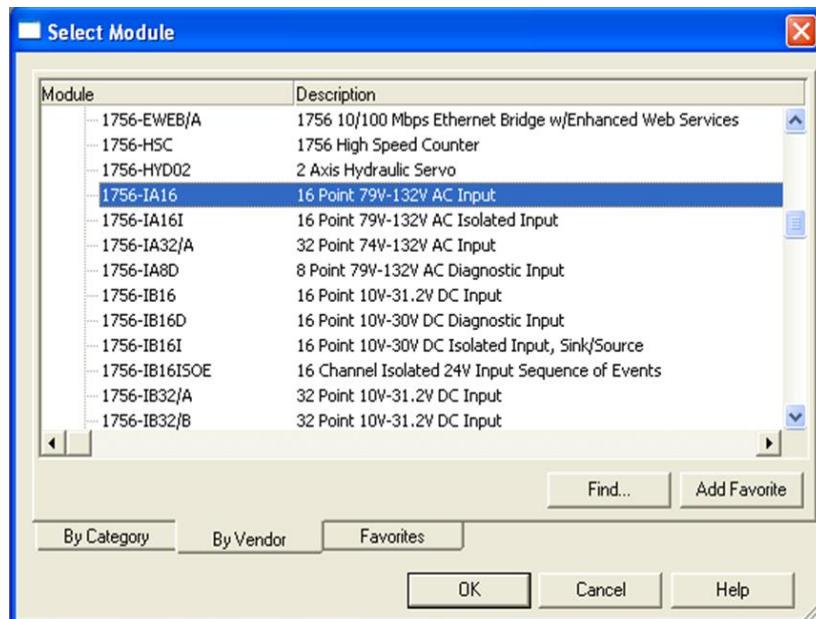
**Figura 13.** Ventana *New Controller*.

Configurada la CPU y *backplane* es habilitada del lado izquierdo de la pantalla la ventana *Controller Organizer* (organizador del controlador), en ella diríjase a *I/O Configuration / 1756 Backplane 1756-A10*, sobre este último haga clic derecho con el ratón y seleccione la pestaña *New Module*, ver **Figura 14**.

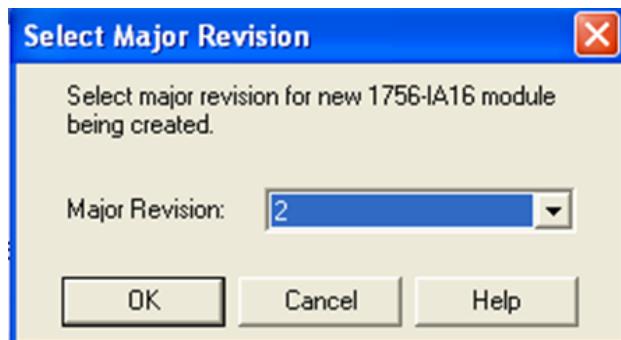


**Figura 14.** Ventana Controller Organizer.

Pulsado *New Module* aparecerá la ventana *Select Module*, el primer módulo en ser agregado será el “1756-IA16/A” (entradas digitales), pulse “OK” (ver **Figura 15**) e instantáneamente se abrirá la ventana *Select Major Revision* la cual se ajustará en un valor de dos (2), ver **Figura 16**.

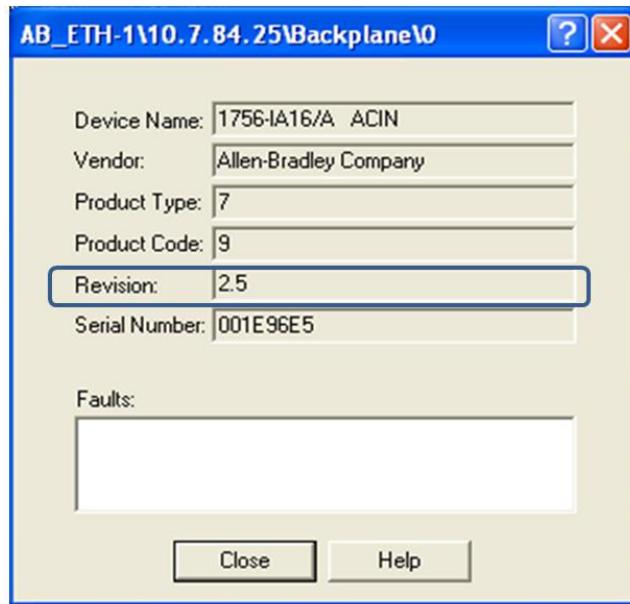


**Figura 15.** Ventana Select Module.



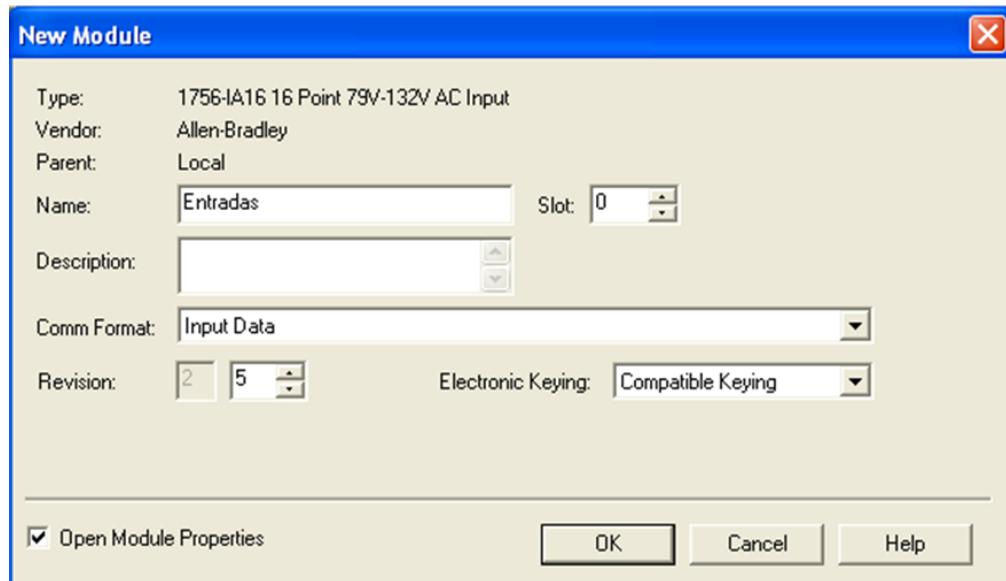
**Figura 16.** Ventana Select Major Revision.

Para saber el valor de la revisión del módulo vaya al RSLinx y haga clic derecho con el ratón sobre el módulo o dispositivo y seleccione *Device Properties*, se abrirá la ventana con la información requerida, en este caso de interés “la revisión”, ver **Figura 17**.



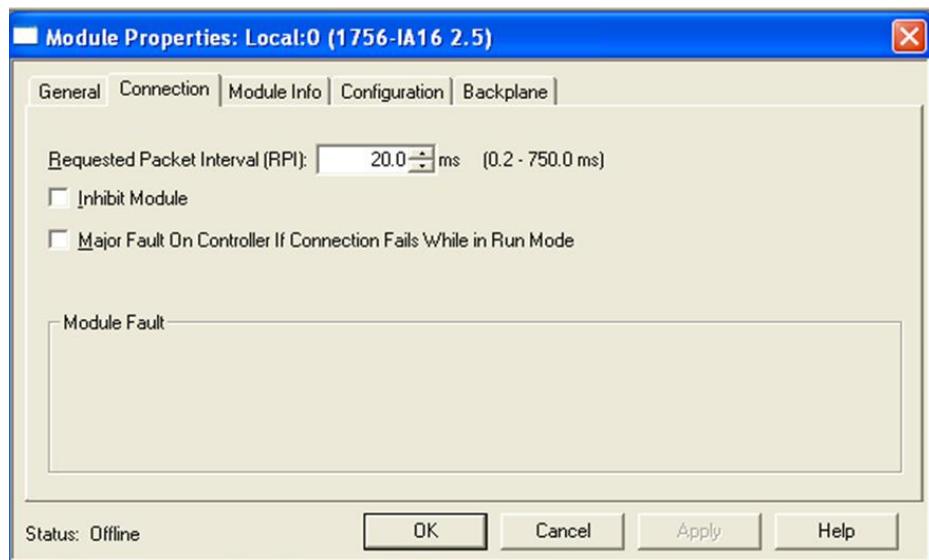
**Figura 17.** Ventana de propiedades del Módulo.

Posterior a la ventana *Select Major Revision*, se accederá a *New Module* donde será posible continuar con la configuración del módulo (1756-IA16/A), rellene el campo de *Name* y ajuste los parámetros *Slot* y *Revision* como es mostrado en la **Figura 18**.



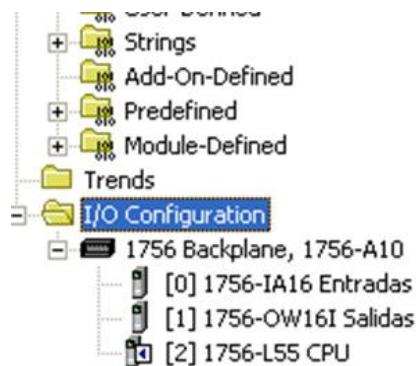
**Figura 18.** Ventana *New Module* (configuración del módulo 1756-IA16/A).

Realizados los ajustes del módulo 1756-IA16/A, aparecerá la ventana *Module Properties* específicamente en la pestaña *Connection*, en ella configure por defecto el parámetro *Requested Packet Interval (RPI)* en **20 milisegundos**, ver **Figura 19**.



**Figura 19.** Ventana *Module Properties* (configuración del módulo 1756-IA16/A).

Finalizada la configuración esencial del módulo de entradas (1756-IA16/A), proceda de la misma forma con el módulo de salidas (1756-OW16I/A). En la **Figura 20** desde el *Controller Organizer* se observan los tres (3) módulos configurados y dispuestos.

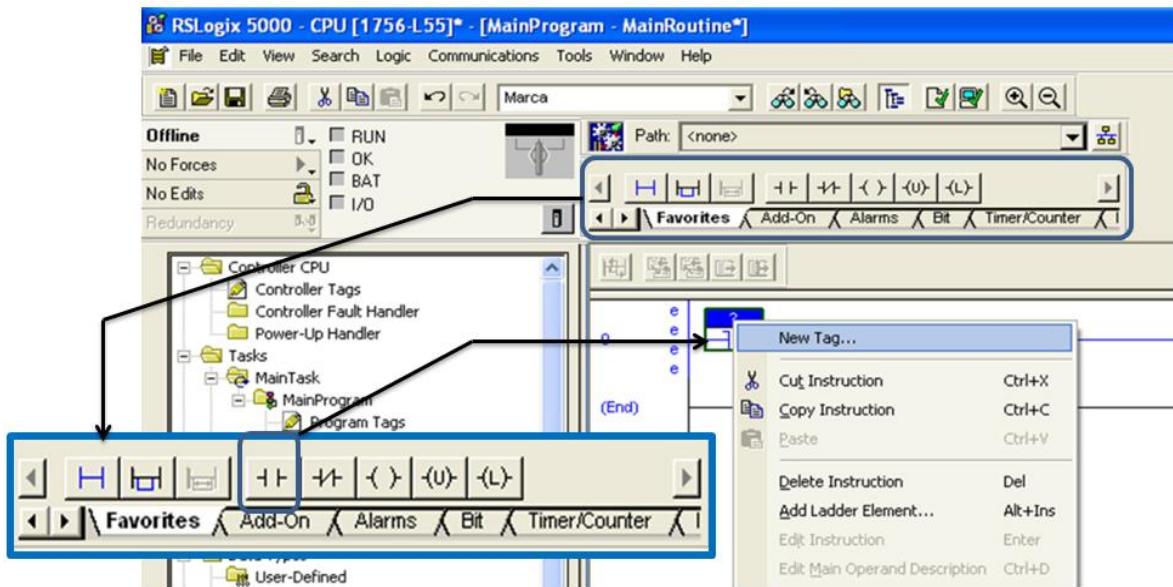


**Figura 20.** Módulos de entradas y salidas configurados.

Configurados los módulos se desarrollará la programación pertinente al proyecto.

Diríjase a *Task / Main Task / MainProgram / MainRoutine*. En *MainRoutine* se realizará la programación, particularmente en lenguaje de contactos (escalera).

Vaya a la pestañas de instrucciones, por defecto se encuentra seleccionada en *Favorites* (favoritos), seleccione un contacto normalmente abierto y arrástrelo hacia el segmento cero (0) y único disponible, posteriormente, haga clic derecho con el ratón sobre el contacto dispuesto en el segmento y seleccione *New Tag*, véase la **Figura 21**.



**Figura 21.** Crear nuevo tag.

Pulsado *New Tag*, se abrirá la ventana *New Tag* donde podrá rellenar todos los parámetros concernientes a la identificación y configuración del tag. Al tag se le asignará el nombre (*Name*) de *Arranque\_manual* y será del tipo (*Type*) Alias direccionado en “Local:0:I.Data.1”, véase la **Figura 22**.

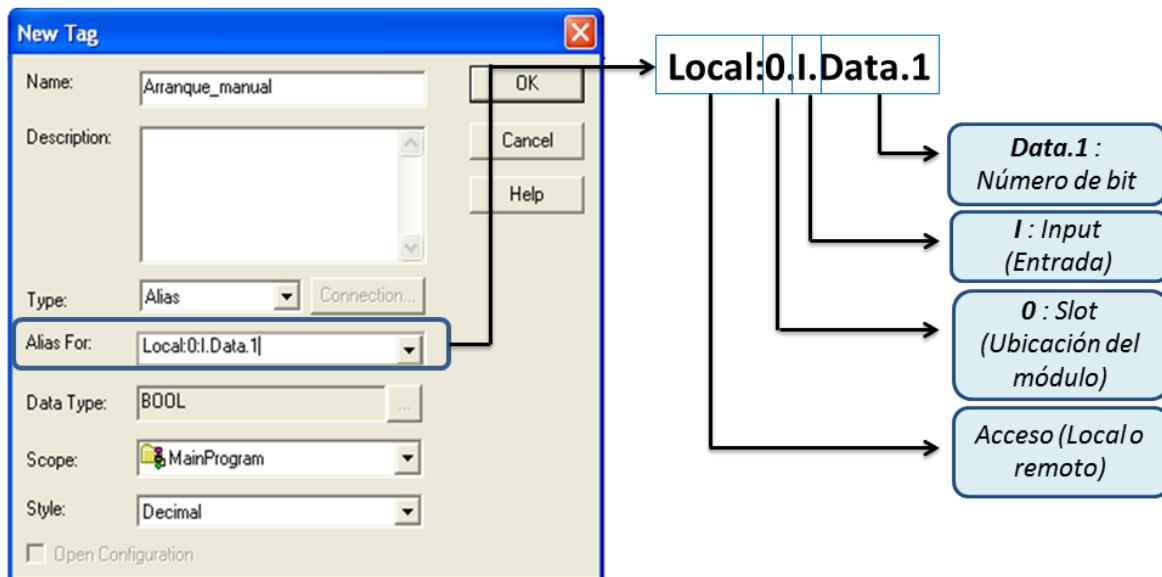


Figura 22. Creación del tag *Arranque\_manual*.

Aplicando la misma metodología empleada para la configuración del tag *Arranque\_manual*, cree el tag *Salida* asociado a la instrucción de asignación insertada al final del segmento cero (0), ver **Figura 23**.

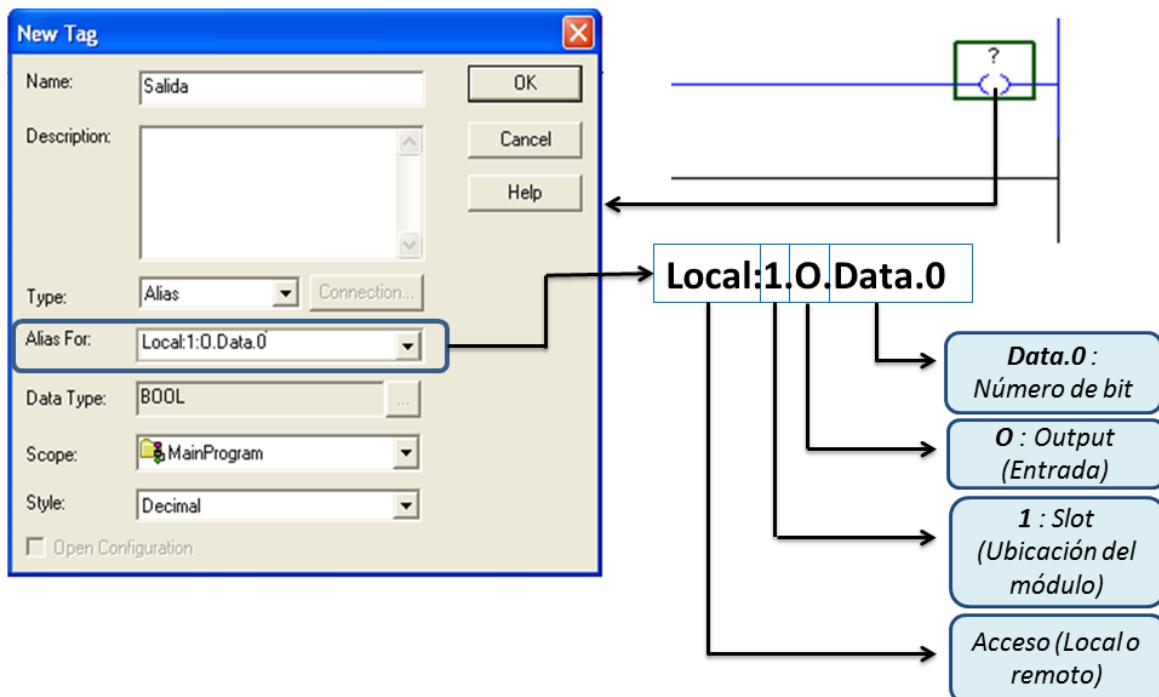
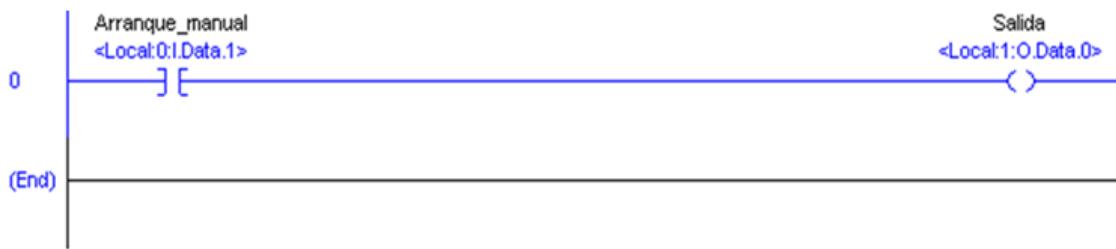


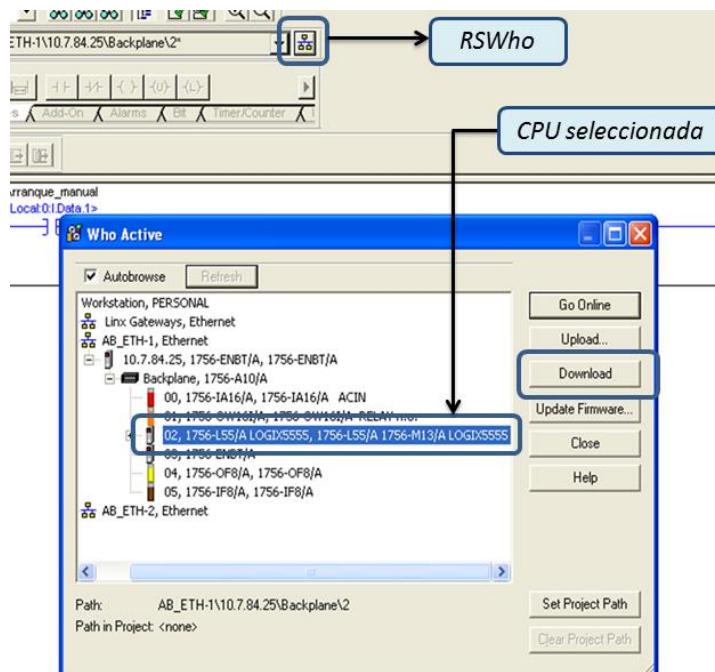
Figura 23. Creación del tag *Salida*.

En la **Figura 24** es mostrado el programa finalizado, este consiste en una entrada (*Arranque\_manual*) y una salida (*Salida*), al encender la entrada (mediante un botón con acción mecánica de interruptor) se activará la salida debido a la continuidad del flujo de poder.



**Figura 24.** Programa realizado (*Arranque\_manual / Salida*).

Finalizado el programa descárguelo al controlador, para ello remítase al botón *RSWho* lo que abrirá la ventana *Who Active* y tal como en el RSLinx tendrá visibilidad del hardware detectado, seleccione la CPU “1756-L55/A” y pulse el botón *Download*, ver **Figura 25**.



**Figura 25.** Descarga del programa al PLC (1).

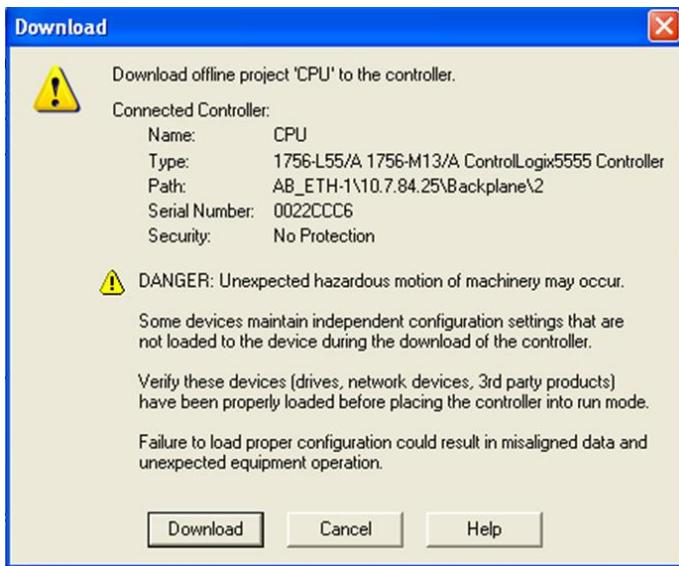


Figura 26. Descarga del programa al PLC (2).

Al descargar el programa al controlador (**Figura 26**) se comprueba el estado de ciertos parámetros, tales como: "OK" (estado de conexión online), "BAT" (estado de la batería del PLC) e "I/O" (estado de los módulos de entradas y salidas); en la **Figura 27** puede observarse el estado de dichos indicadores de estado posteriormente a la descarga del programa.

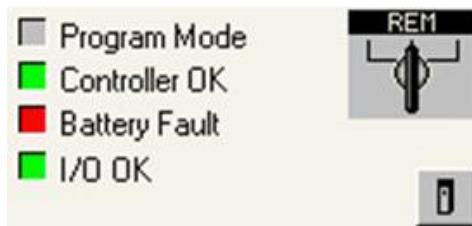
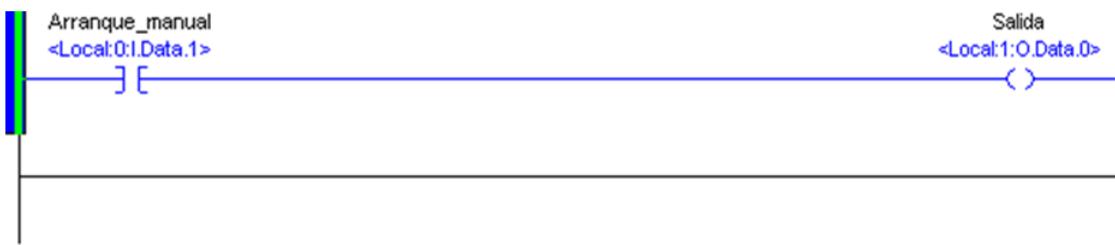


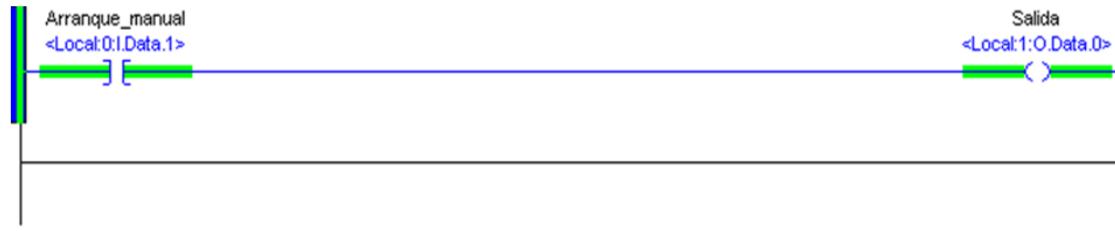
Figura 27. Parámetros de monitoreo del PLC.

Descargado el programa en el controlador se colocará en modo de operación *Run Mode*, para ello diríjase a *Rem Prog* y seleccione *Run Mode*, note como en los parámetros de monitoreo del PLC el indicador *Program Mode* cambia a *Run Mode*, en la **Figura 28** puede verse como los rieles del programa se energizan (color verde).



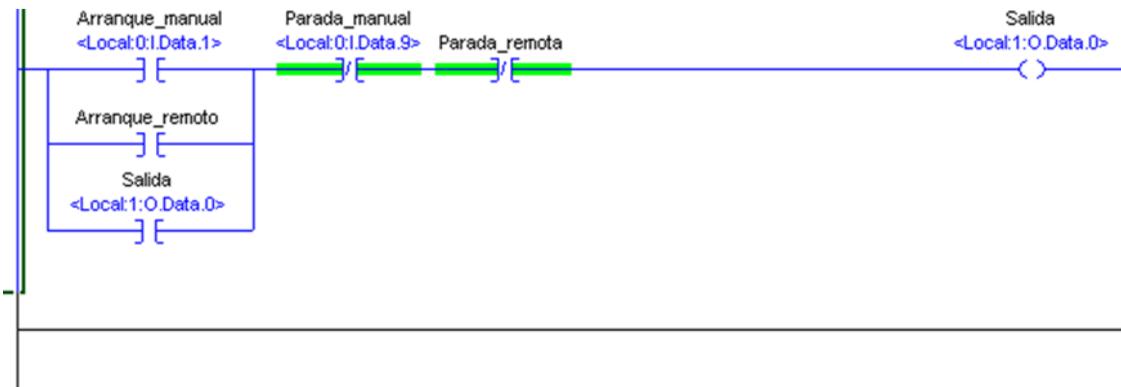
**Figura 28.** Observación del programa en *Run Mode* con entrada inactiva.

Active la entrada física asociada al *tag Arranque\_manual* y note como se activa la salida (**Figura 29**).



**Figura 29.** Observación del programa en *Run Mode* con entrada activa.

Finalizada la verificación del programa cambie el estado de operación del PLC a *Program Mode*, esto es debido a que se llevará a cabo un rediseño en la lógica creada previamente, el resultado de estos cambios puede verse en la **Figura 30**.



**Figura 30.** Lógica modificada.

El programa rediseñado (**Figura 30**) corresponde a la activación y desactivación de una salida física del PLC implementado auto enganche o auto enclavamiento (debido al uso de pulsadores), la activación (arranque) se encuentra programada para ejecutarse de forma manual o lógica (remota), de igual manera para la desactivación (parada), el hecho de esta programación es para aportar seguridad, flexibilidad y maniobra a la lógica en el uso posterior de una HMI.

#### 4.3. Configuración del servidor OPC

En esta sección se configurará el servidor OPC para la posterior implementación de LabVIEW como "cliente" para el desarrollo de la HMI.

La metodología aplicada para la configuración del servidor OPC es similar a la de la práctica 2. Cree un nuevo canal seleccionando *New Channel*, seguidamente configure una por una las ventanas que se desplegarán. Lo más resaltante para la configuración del canal son: el tipo de driver a ser utilizado (*Device Driver*), en este caso será seleccionado como "Allen-Bradley ControlLogix Ethernet" y el adaptador de red (*Network Interface*) que se seleccionará en *Default*, las demás configuraciones irán por defecto, véase la **Figura 31** (identificación del canal), **Figura 32** (selección del driver) y **Figura 33** (selección del adaptador de red).

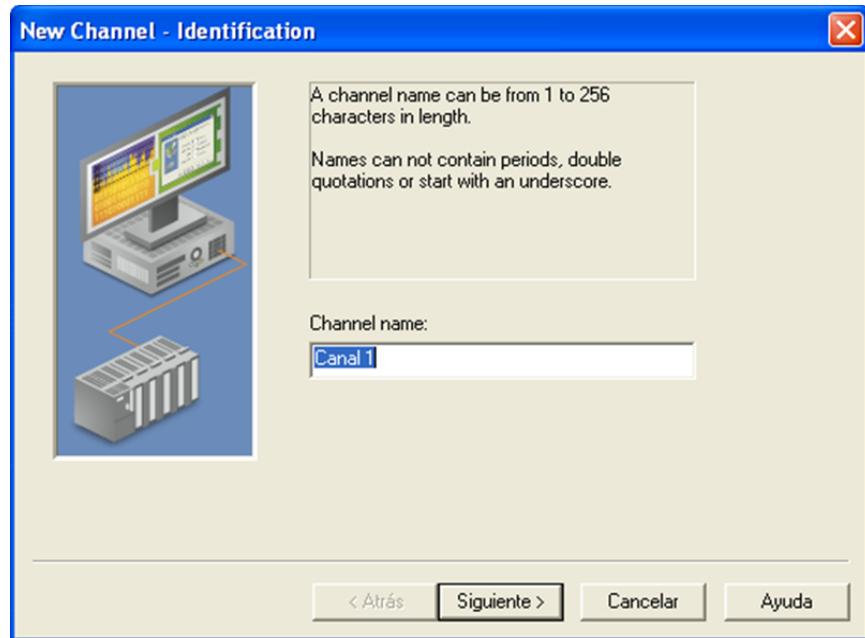


Figura 31. Ventana de identificación del canal (*Identification*).

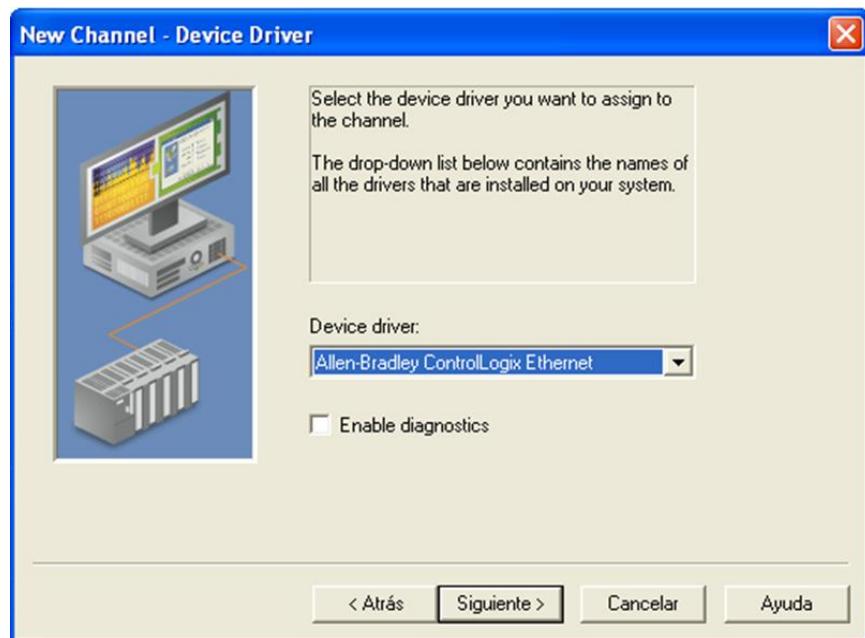
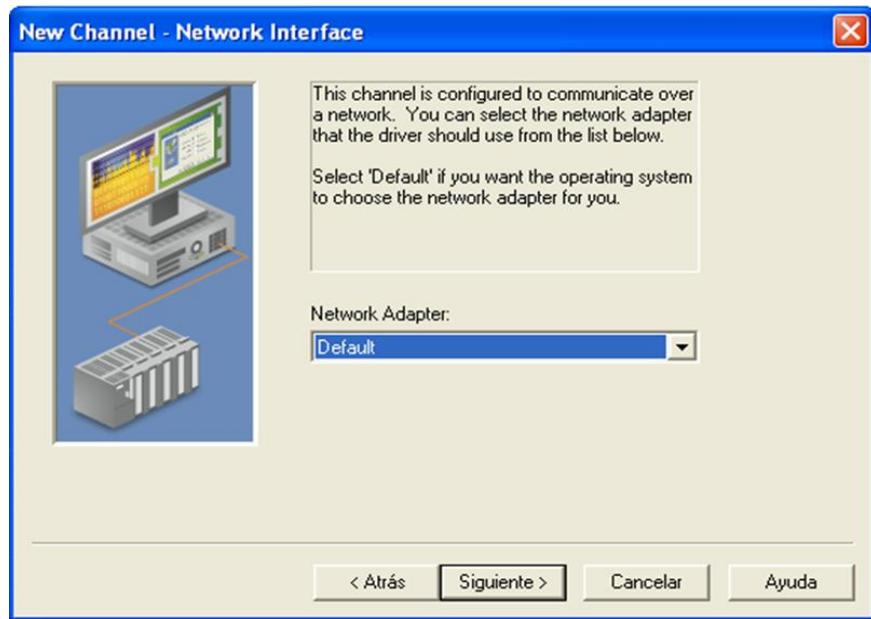
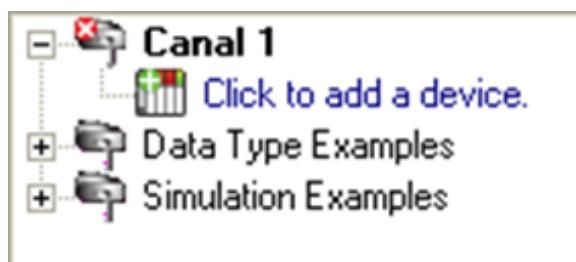


Figura 32. Ventana de selección del driver correspondiente al PLC (*Device Driver*).



**Figura 33.** Ventana de selección del adaptador de red del ordenador (*Network Interface*).

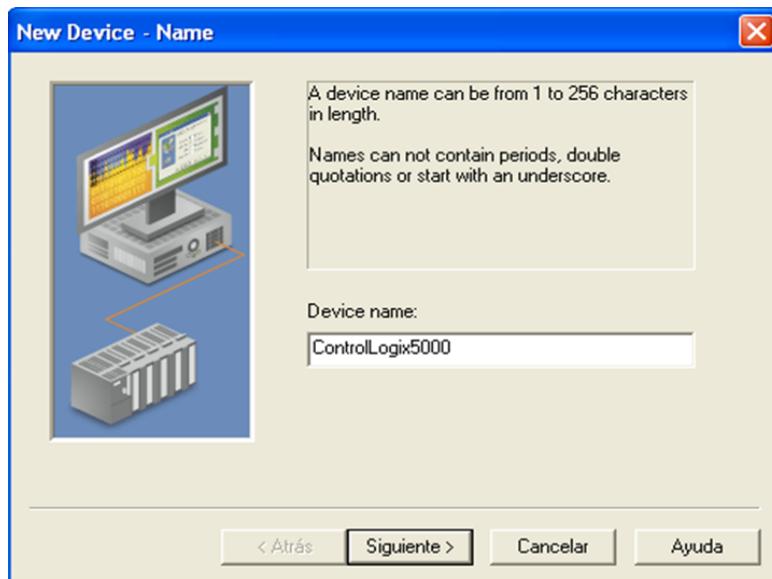
Finalizado la configuración del canal puede verse este creado en la **Figura 34**.



**Figura 34.** *Canal 1* agregado.

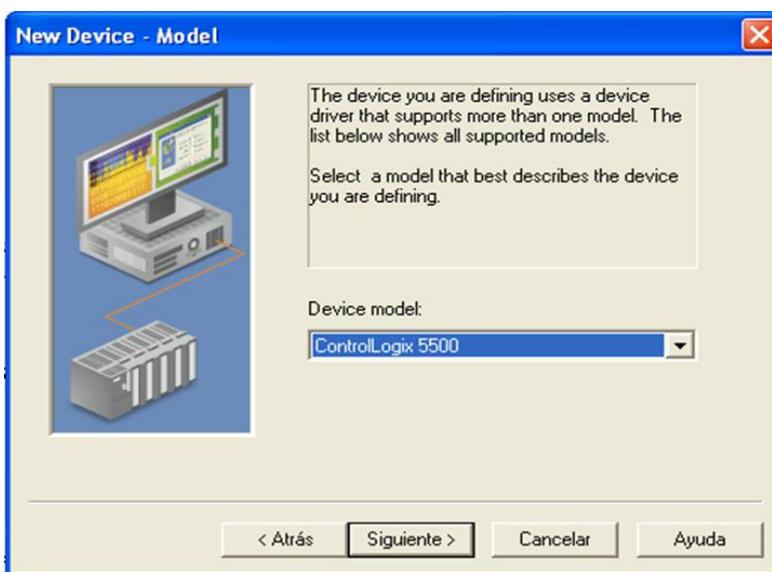
Creado el canal se agregará el controlador a él, véase en las siguientes figuras la configuración establecida para el controlador.

En la **Figura 35** se define el nombre del dispositivo, en este caso: *ControlLogix 5000*.



**Figura 35.** Ventana de asignación del nombre del controlador (*Name*).

Identificado el controlador se abrirá la ventana *Model* donde se elegirá el modelo del dispositivo, en este caso seleccione *ControlLogix 5500* el cual no genera ningún tipo de incompatibilidad con la CPU “1756-L55 ControlLogix 5555”, en la **Figura 36** puede verse la selección del modelo y en la **Figura 37** la compatibilidad del NI OPC Server con ciertos modelos ControlLogix.

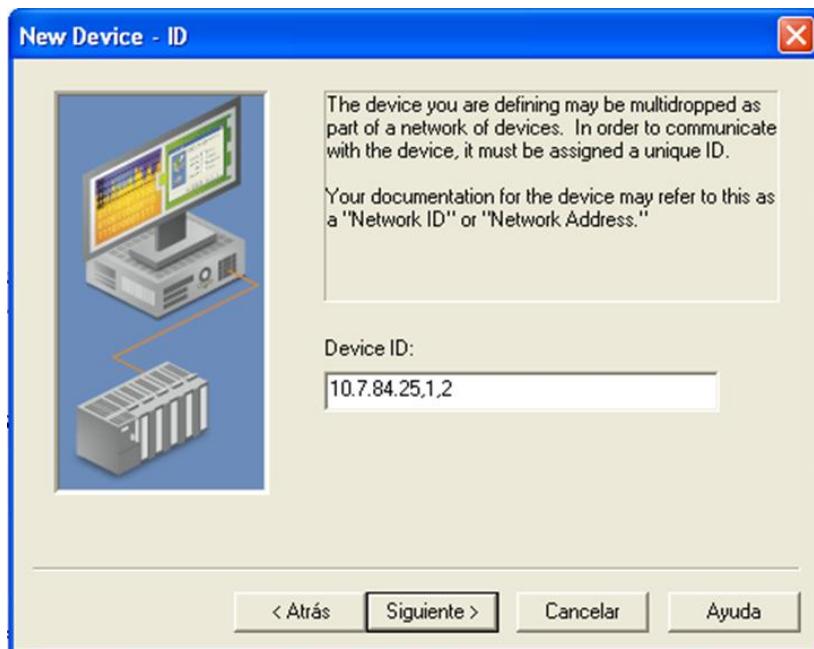


**Figura 36.** Ventana de selección del modelo de controlador (*Model*).

- Allen-Bradley ControlLogix OPC Server Supported Devices**
- ControlLogix 5550
  - ControlLogix 5553
  - ControlLogix 5555
  - ControlLogix 5561
  - ControlLogix 5562
  - ControlLogix 5563

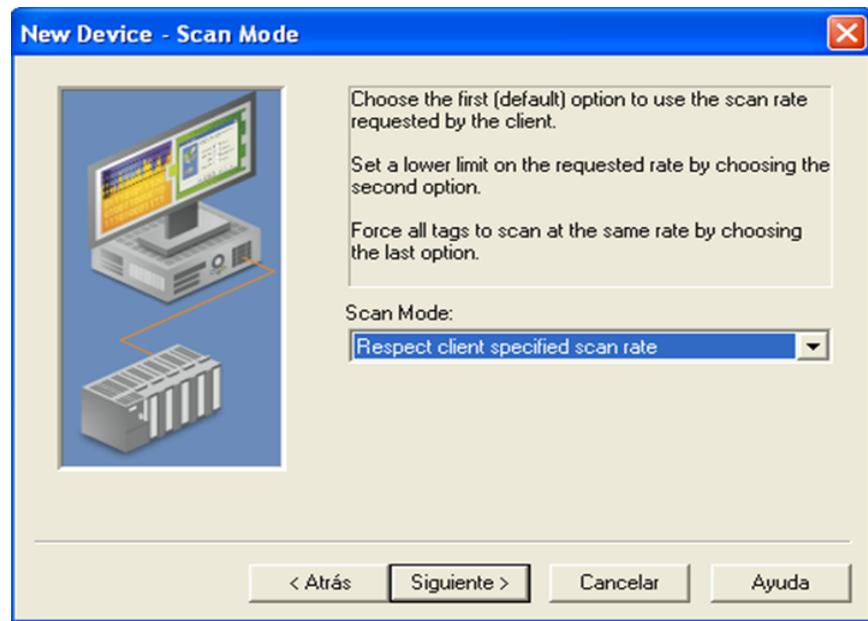
**Figura 37.** Dispositivos ControlLogix soportados por el NI OPC Server.  
(Fuente: [www.ni.com](http://www.ni.com)).

Posteriormente proceda a introducir la ID del PLC que estará establecida por la secuencia: “<IP or Hostname>, 1, [<Optional Routing Path>], <CPU Slot>”, véase la **Figura 38**.



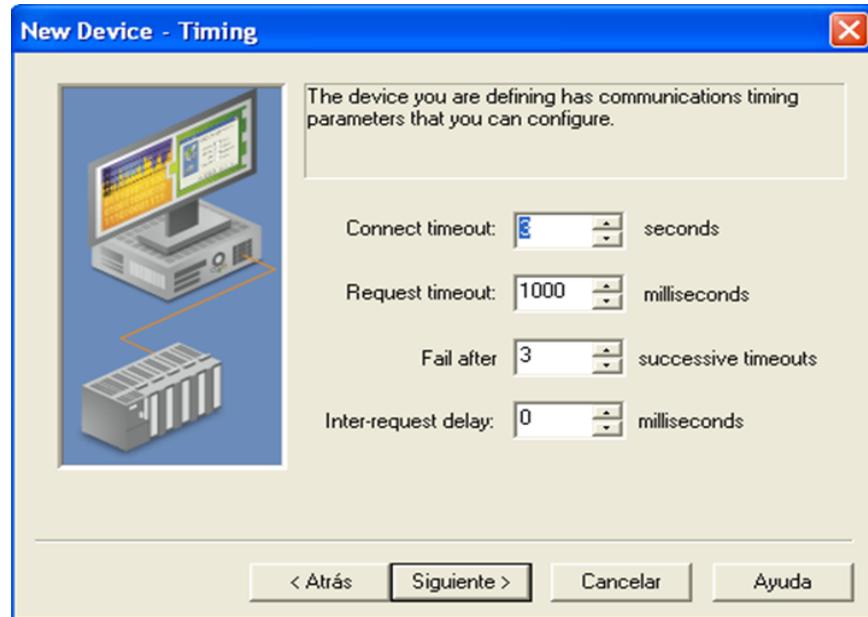
**Figura 38.** Ventana de identificación del dispositivo en la red (ID).

En la ventana *Scan Mode* deje por defecto *Respect client specified scan rate* (**Figura 39**).



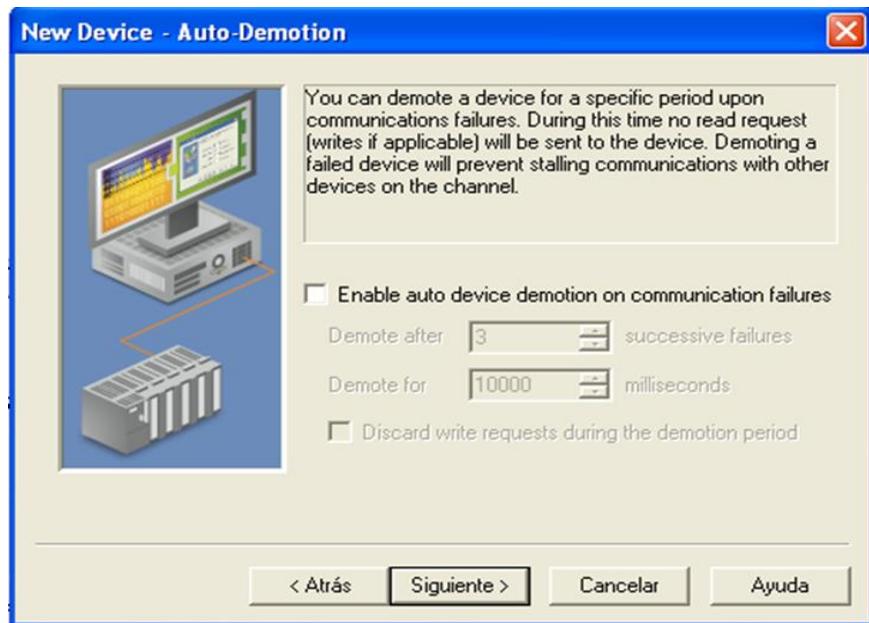
**Figura 39.** Ventana de configuración de barrido de datos (*Scan Mode*).

En *Timing* establezca por defecto los valores especificados, ver **Figura 40**.



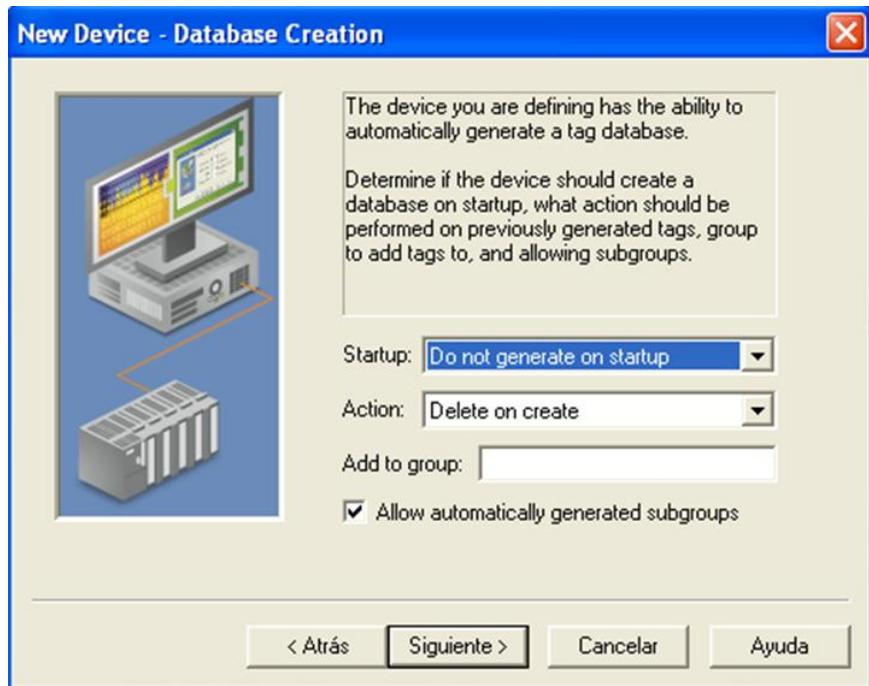
**Figura 40.** Ventana parámetros de temporización (*Timing*).

La ventana *Auto-Demotion* quedará configurada por defecto, ver **Figura 41**.



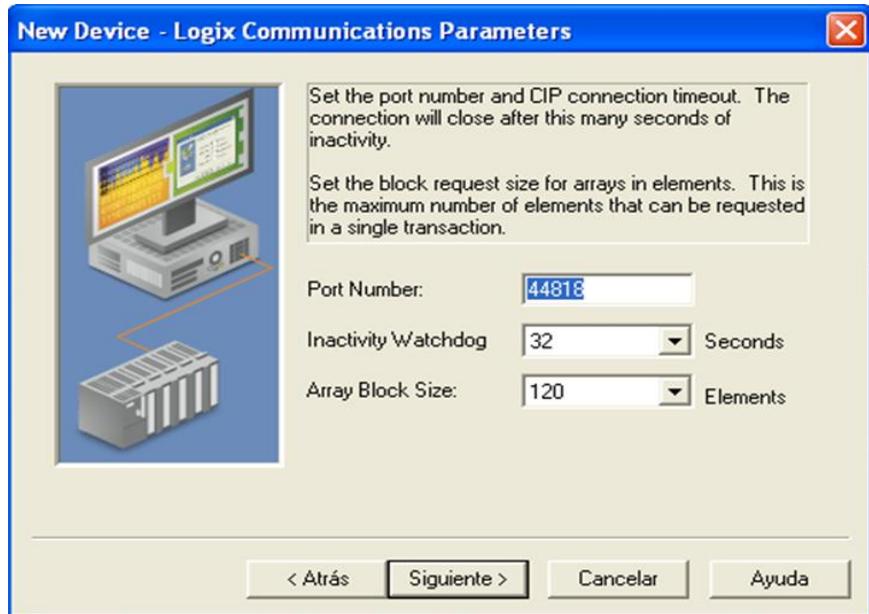
**Figura 41.** Ventana *Auto-Demotion*.

Posteriormente, en *Database-Creation* la ventana irá configurada por defecto, ver **Figura 42.**

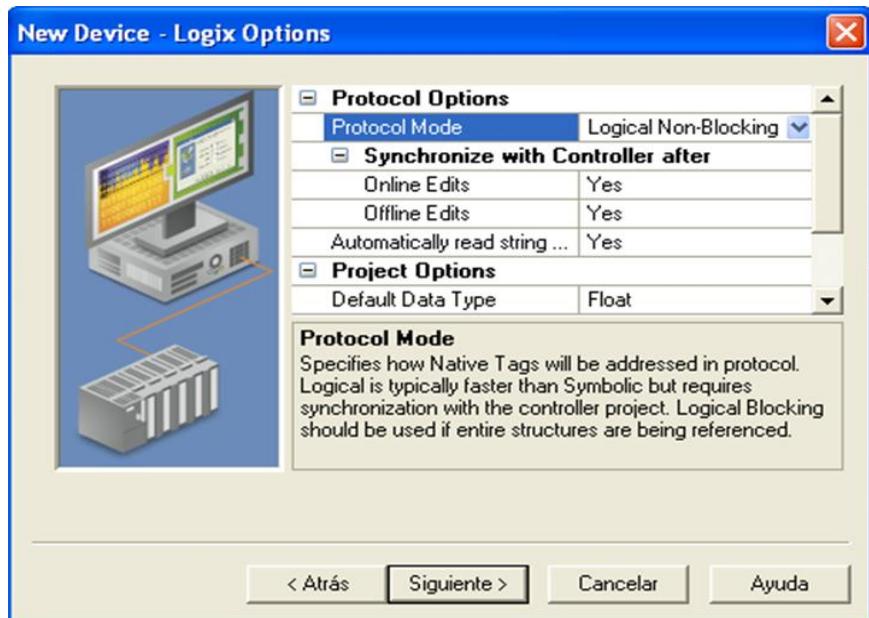


**Figura 42.** Ventana de creación de tags (*Database Creation*).

En *Logix Communications Parameters*, las configuraciones que se establecerán serán por defecto (**Figura 43**), de igual manera con *Logix Options* (**Figura 44**).

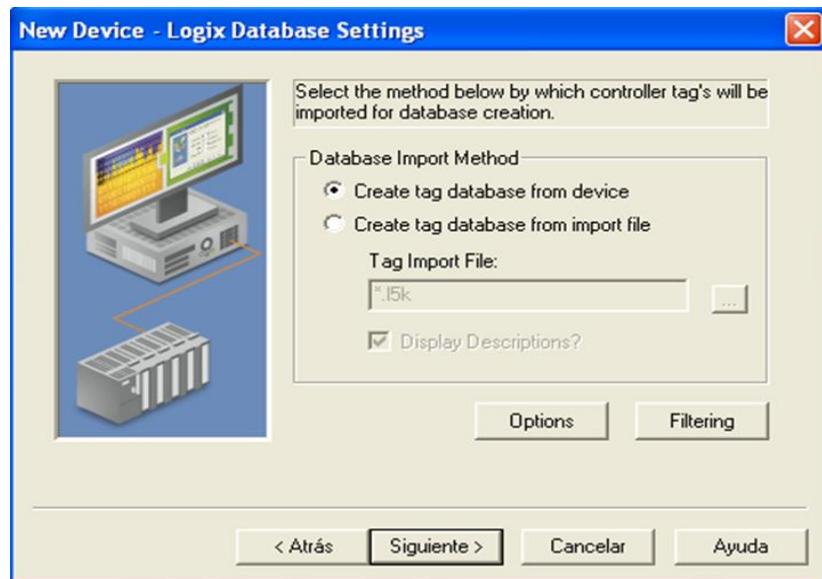


**Figura 43.** Ventana de parámetros de conexión (*Logix Communications Parameters*).



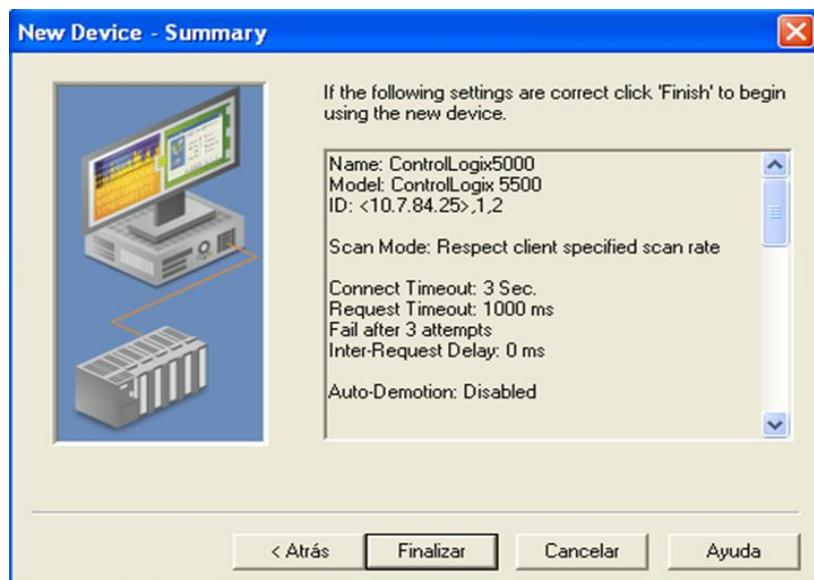
**Figura 44.** Ventana opciones varias (*Logix Options*).

Luego en *Logix Database Setting*, deje la ventana configurada por defecto, ver **Figura 45.**



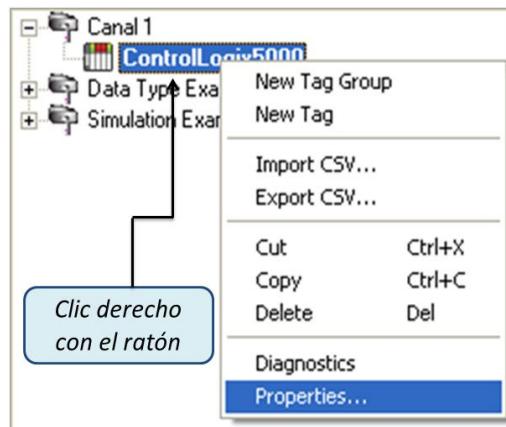
**Figura 45.** Ventana de importación de *tags* (*Logix Database Setting*).

Por último, aparecerá un resumen de todas las configuraciones hechas previamente (**Figura 46**).

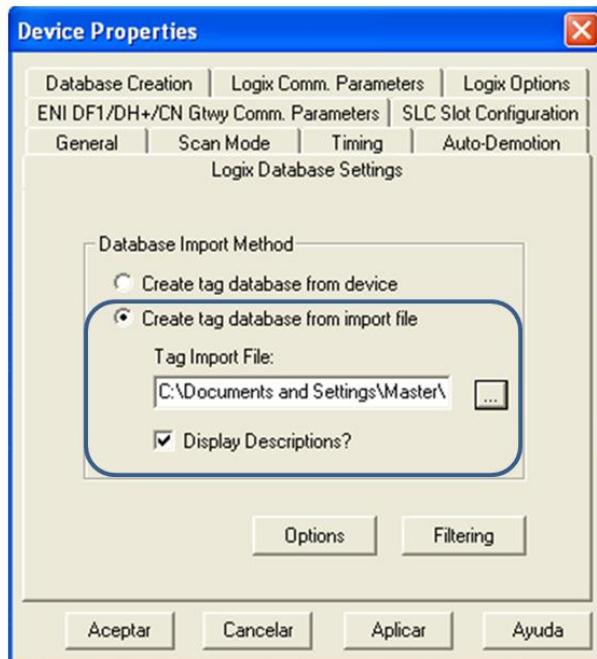


**Figura 46.** Ventana de resumen (*Summary*).

Finalizada la configuración del dispositivo (controlador) dentro del canal se agregarán los *tags* requeridos para el posterior diseño de la HMI, para realizar esto último vaya al RSLogix 5000 y guarde el proyecto con extensión “.L5K”, en este caso se guardará como “Practica 3.L5K”, posteriormente, seleccione *Properties* del PLC agregado en el canal del servidor OPC y elija *Logix Database Settings*, dentro seleccione la opción *Create tag data base from import file*, para así importar los *tags* del programa, seguidamente, elija importar el archivo “Practica 3.L5K” para realizar dicha operación, en las **Figuras 47 y 48** se ilustra este procedimiento.

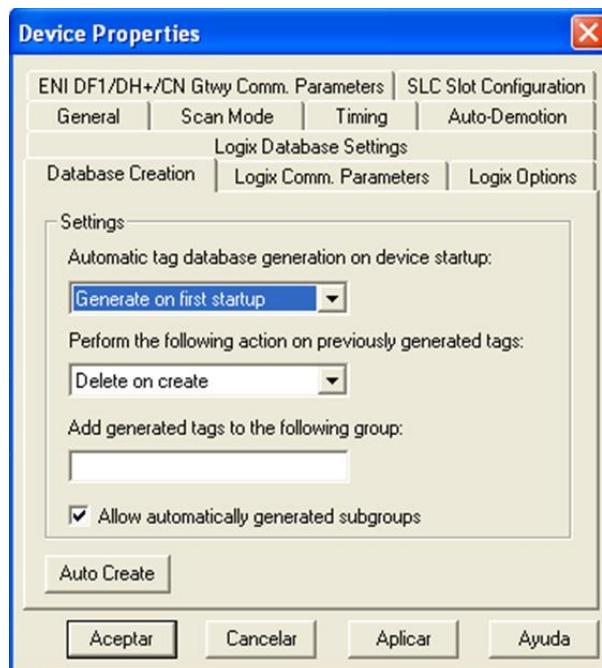


**Figura 47.** Selecciona de *Properties* para la posterior importación de *tags*.



**Figura 48.** Importar el archivo “.L5K”.

Ahora, diríjase a la pestaña *Database Creation* y seleccione la opción *Generate on first startup*, pulse *Auto Create*, ver la **Figura 49**.



**Figura 49.** Creación de los tags (importados).

En la **Figura 50**, pueden verse creados todos los *tags* asociados al programa.

The screenshot shows the ControlLogix 5000 software interface. On the left, the project tree displays 'Canal 1', 'ControlLogix5000', 'Prgm\_MainProgram' (which is expanded to show 'MAINPROGRAM.LAD'), 'Data Type Examples', and 'Simulation Examples'. On the right, a table lists the created tags:

Tag Name	Address	Data Type	Scan Rate
Arranque_manual	PROGRAM:MAINPROGRAM.A...	Boolean	100
Arranque_remoto	PROGRAM:MAINPROGRAM.A...	Boolean	100
Parada_manual	PROGRAM:MAINPROGRAM.P...	Boolean	100
Parada_remota	PROGRAM:MAINPROGRAM.P...	Boolean	100
Salida	PROGRAM:MAINPROGRAM.S...	Boolean	100

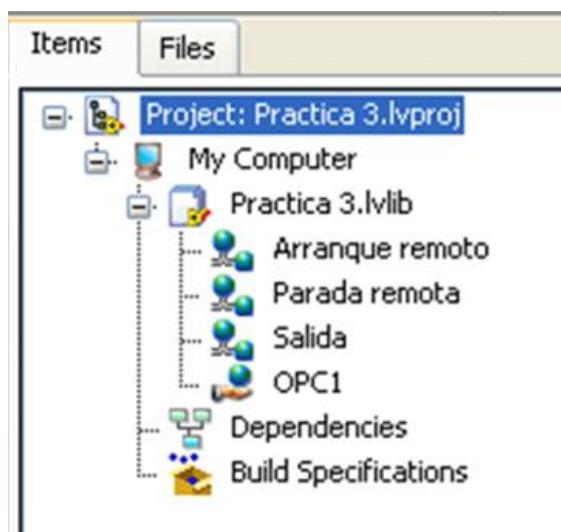
**Figura 50.** Tags creados.

Utilice el *Quick Client* para verificar la correcta lectura de los *tags* importados, para ello recuerde tener el PLC en *Run Mode*.

#### 4.4. Configuración de LabVIEW

Configurado correctamente el servidor OPC se ajustará como “cliente OPC” a LabVIEW para la posterior creación de HMIs.

La metodología empleada para la creación del proyecto, librerías, variables y configuración del cliente OPC es la misma implementada en la **práctica 2**, por lo tanto en la **Figuras 51** es mostrado el proyecto con todo lo que será necesario para la posterior creación de la interfaz gráfica.



**Figura 51.** Proyecto configurado con su respectiva librería, variables y Cliente OPC.

#### 4.5. Crear VI del proyecto

El VI que vendría siendo la HMI será creado tal como el realizado en la práctica 2, dispondrá de dos (2) *Push Button* configurados como pulsadores, los cuales irán asociados o conectados a las variables de *Arranque\_remoto* y *Parada\_remota*, además llevará un (1) indicador *Round LED* que irá conectado a la variable *Salida*, en las **Figuras 52 y 53** se muestra el diseño del VI desde el *Front Panel* y *Block Diagram*.

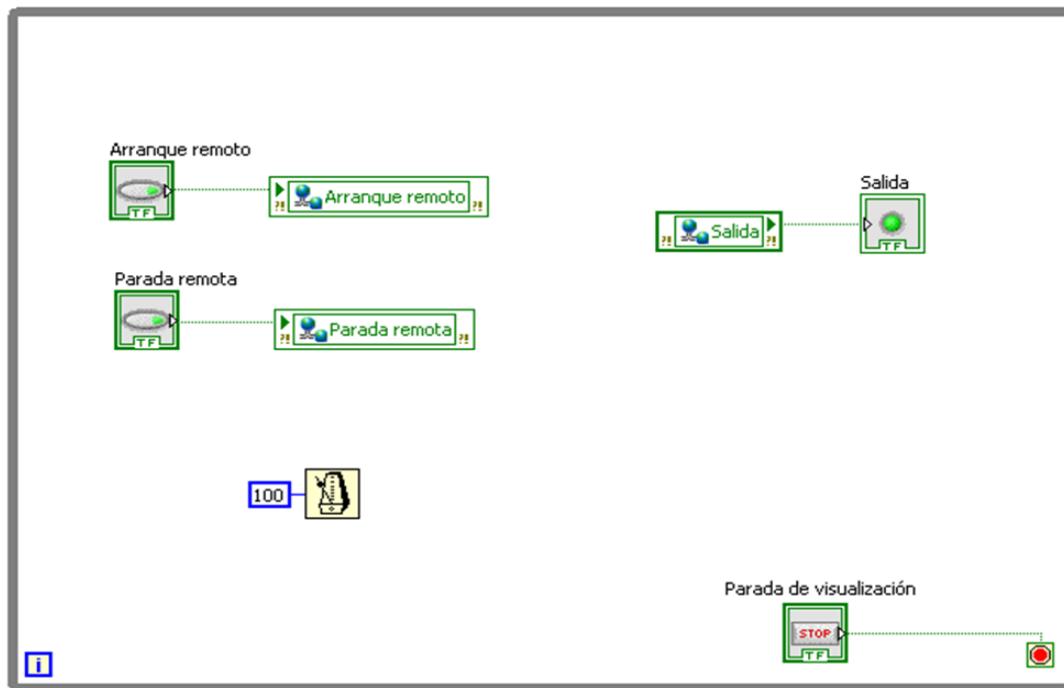


Figura 52. *Block Diagram* del VI.



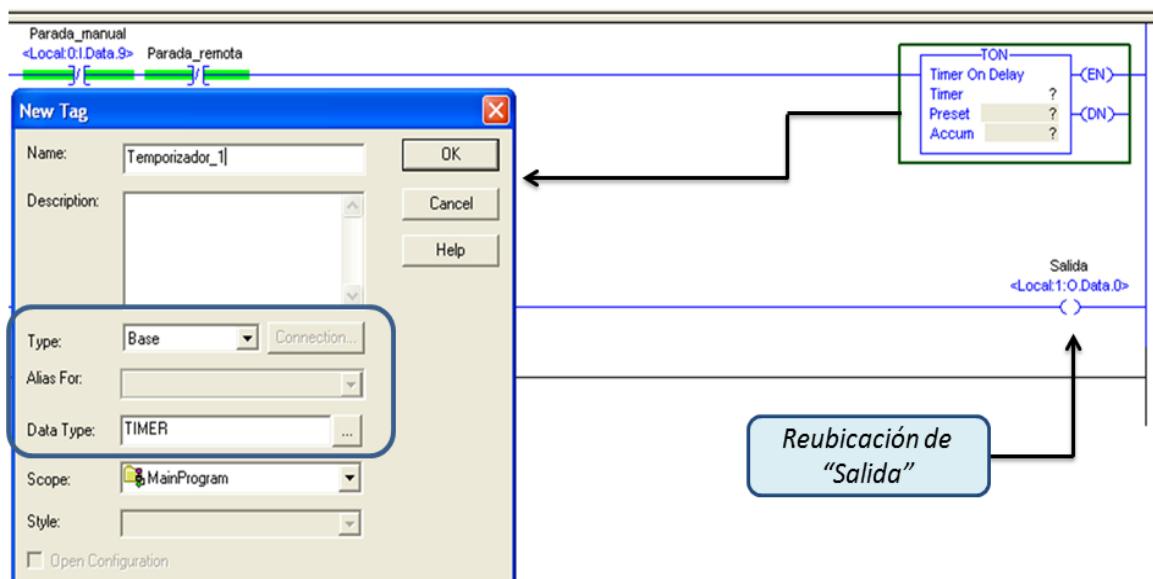
Figura 53. *Front Panel* del VI.

#### **4.6. Adicionar al proyecto un temporizador y contador controlado y supervisado desde LabVIEW.**

Diríjase a *MainRoutine* en el RSLogix 5000, desde la barra de instrucciones *Favorites* seleccione la instrucción agregar nuevo segmento en tres (3) ocasiones, esto añadirá tres (3) nuevos segmentos al programa.

Seleccione la instrucción de asignación *Salida* disponible en el segmento cero (0) y arrástrelo al segmento uno (1). Posteriormente, en las pestañas de instrucciones seleccione *Timer / Counter* y elija la instrucción TON (temporizador con retardo a la conexión), agréguela en el segmento cero (0).

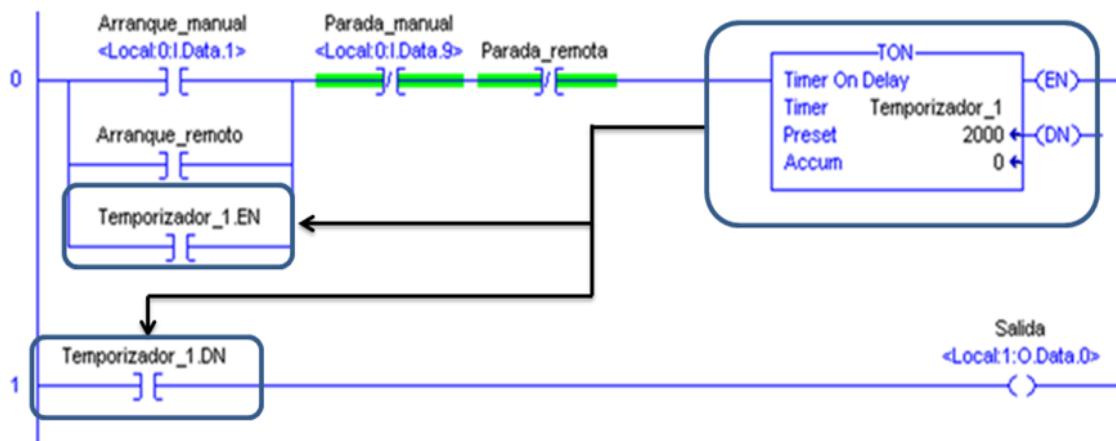
Haga clic derecho sobre el temporizador TON y seleccione la opción *New Tag*, se abrirá la ventana *New Tag*, en ella identifique al temporizador, en este caso como *Temporizador\_1* y los demás parámetros déjelos por defecto, ver **Figura 54**.



**Figura 54.** Inserción del temporizador *Temporizador\_1* y cambio de segmento para Salida.

Agregado el temporizador *Temporizador\_1* desde el segmento cero (0) redireccionó el contacto asociado a *Salida* (auto enganche) en *Temporizador\_1.EN*, este es el bit de habilitación del temporizador que sustituirá a *Salida* para efectos del auto enganche.

Posteriormente, inserte un contacto normalmente abierto en el segmento uno (1) y direcciónelo en *Temporizador\_1.DN*, este corresponde al bit de efectuado y cambia de estado (*false* a *true*) cuando el valor acumulado (*Accum*) del temporizador sea mayor o igual al valor del preestablecido (*Preset*), ver **Figura 55**.



**Figura 55.** Re direcciónado del contacto de auto enganche en *Temporizador\_1.EN* y colocación del contacto *Temporizador\_1.DN*.

Después, añada en el segmento dos (2) un contador implementado la misma metodología que en el temporizador anterior, inmediatamente, inserte un contacto normalmente abierto y asócielo a la instrucción *Salida*, véase la **Figura 56**.



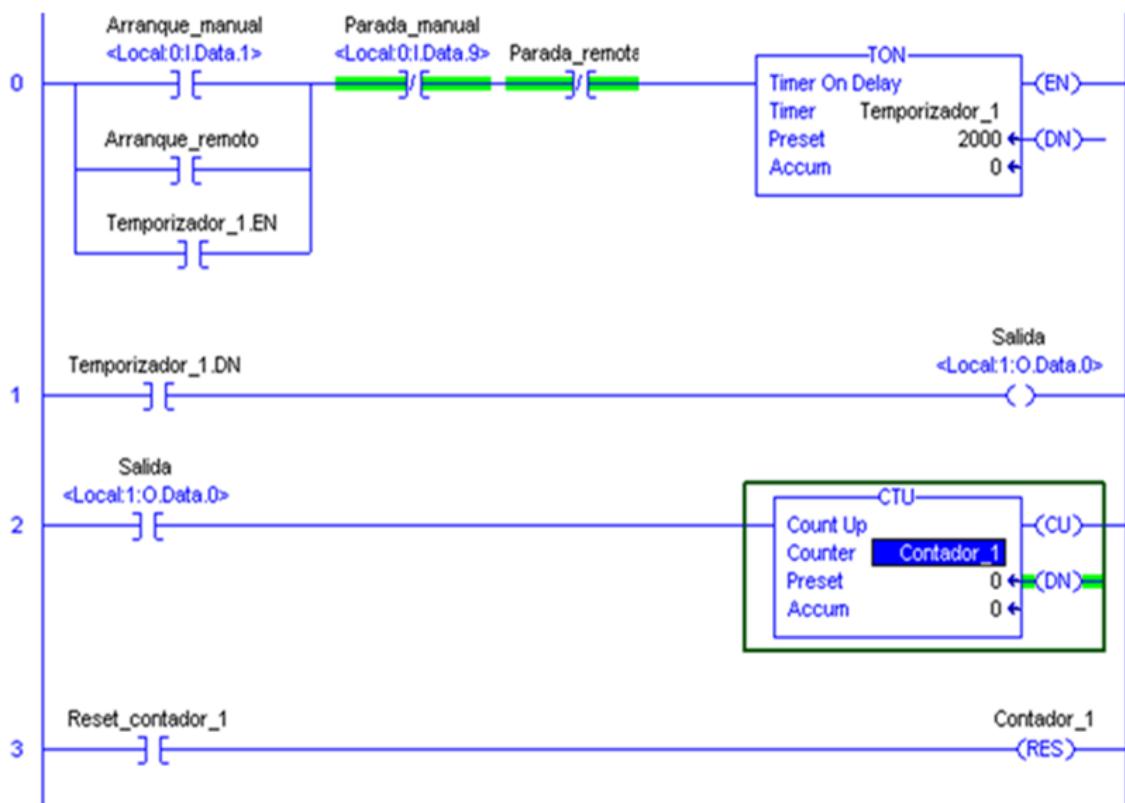
**Figura 56.** Adición de contacto asociado a *Salida* y *Contador\_1*.

Luego, en el segmento tres (3) introduzca otro contacto normalmente abierto, pero asociado a un *tag* tipo base llamado *Reset\_contador\_1*, seguidamente, añada la instrucción RES la cual se encuentra disponible en la pestaña de instrucciones *Timer / Counter*, la instrucción RES será asociada al *tag* (*Contador\_1*) del contador y esta se encargará de reiniciar la cuenta de dicho contador, ver **Figura 57**.



**Figura 57.** Tag tipo base *Reser\_contador\_1* e instrucción de *reset Contador\_1*.

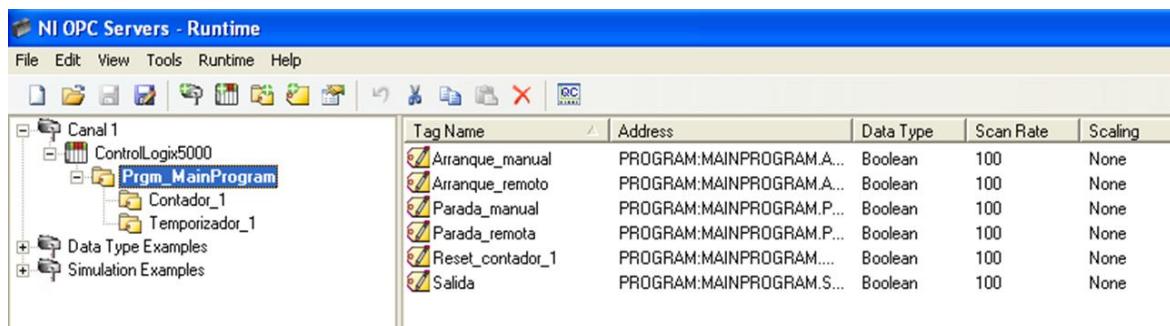
El programa ha sido finalizado, la programación que se ha hecho adicional en esta sección permite temporizar y contar el número de activaciones de la *Salida*. Véase en la **Figura 58** el programa completo con todos sus segmentos.



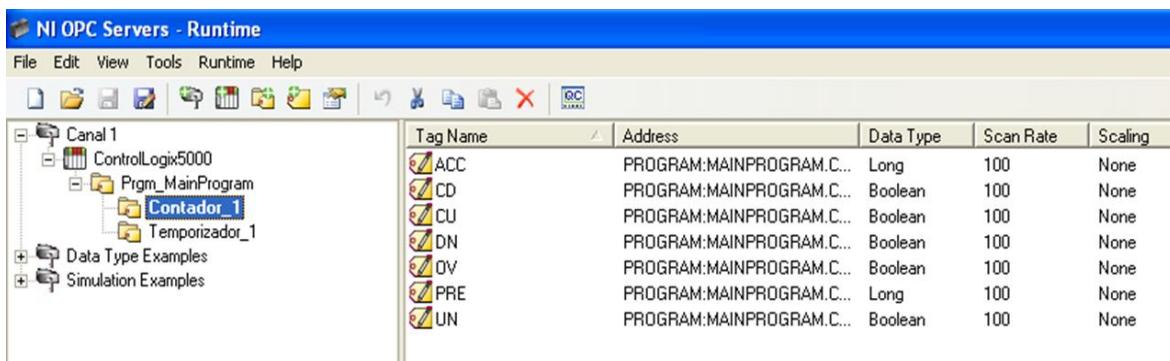
**Figura 58.** Programa con temporización y conteo de la activación de *Salida*.

Descargue el programa en el PLC y pruébelo para verificar su correcto funcionamiento.

Corroborado el funcionamiento del programa, guarde el proyecto en formato “L5K” y cargue los **nuevos tags** al servidor OPC haciendo uso de la metodología explicada previamente en “Configuración del servidor OPC”, en las **Figuras 59, 60 y 61** se muestran cargadas dentro *Prgm\_MainProgram* las carpetas contenedoras de *tags* del *Contador\_1* y *Temporizador\_1*.



**Figura 59.** Carpetas contenedoras de *tags Contador\_1* y *Temporizador\_1* agregadas al *Prgm\_MainProgram*.



**Figura 60.** Tags contenidos en *Contador\_1*.

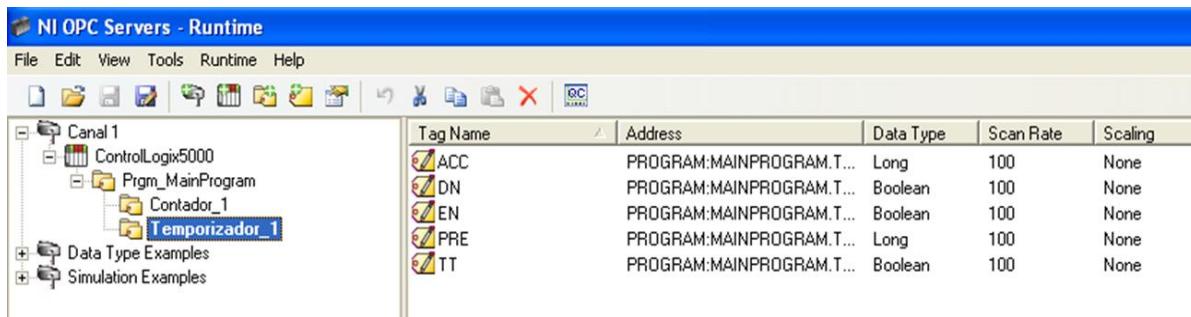


Figura 61. Tags contenidos en *Temporizador\_1*.

Utilice el *Quick Client* para verificar la correcta lectura de los nuevos *tags* importados, para ello recuerde tener el PLC en modo *Run Mode*, ver **Figura 62**.

Item ID	Data Type	Value	Timestamp	Quality
Canal 1.ControlLogix5000.Prgm_MainProgram.Arranque_manual	Boolean	0	02:42:31.218	Good
Canal 1.ControlLogix5000.Prgm_MainProgram.Arranque_remoto	Boolean	0	02:42:31.218	Good
Canal 1.ControlLogix5000.Prgm_MainProgram.Contador_1.ACC	Long	0	02:42:31.218	Good
Canal 1.ControlLogix5000.Prgm_MainProgram.Contador_1.CD	Boolean	0	02:42:31.218	Good
Canal 1.ControlLogix5000.Prgm_MainProgram.Contador_1.CU	Boolean	0	02:42:31.218	Good
Canal 1.ControlLogix5000.Prgm_MainProgram.Contador_1.DN	Boolean	1	02:42:31.218	Good
Canal 1.ControlLogix5000.Prgm_MainProgram.Contador_1.OV	Boolean	0	02:42:31.218	Good
Canal 1.ControlLogix5000.Prgm_MainProgram.Contador_1.PRE	Long	0	02:42:31.218	Good
Canal 1.ControlLogix5000.Prgm_MainProgram.Contador_1.UN	Boolean	0	02:42:31.218	Good
Canal 1.ControlLogix5000.Prgm_MainProgram.Parada_manual	Boolean	0	02:42:31.218	Good
Canal 1.ControlLogix5000.Prgm_MainProgram.Parada_remota	Boolean	0	02:42:31.218	Good
Canal 1.ControlLogix5000.Prgm_MainProgram.Reset_contador_1	Boolean	0	02:42:31.218	Good
Canal 1.ControlLogix5000.Prgm_MainProgram.Salida	Boolean	0	02:42:31.218	Good
Canal 1.ControlLogix5000.Prgm_MainProgram.Temporizador_1.ACC	Long	0	02:42:31.218	Good
Canal 1.ControlLogix5000.Prgm_MainProgram.Temporizador_1.DN	Boolean	0	02:42:31.218	Good
Canal 1.ControlLogix5000.Prgm_MainProgram.Temporizador_1.EN	Boolean	0	02:42:31.218	Good
Canal 1.ControlLogix5000.Prgm_MainProgram.Temporizador_1.PRE	Long	2000	02:42:31.218	Good
Canal 1.ControlLogix5000.Prgm_MainProgram.Temporizador_1.TT	Boolean	0	02:42:31.218	Good

Figura 62. Monitoreo de todos los *tags* del *Prgm\_MainProgram*.

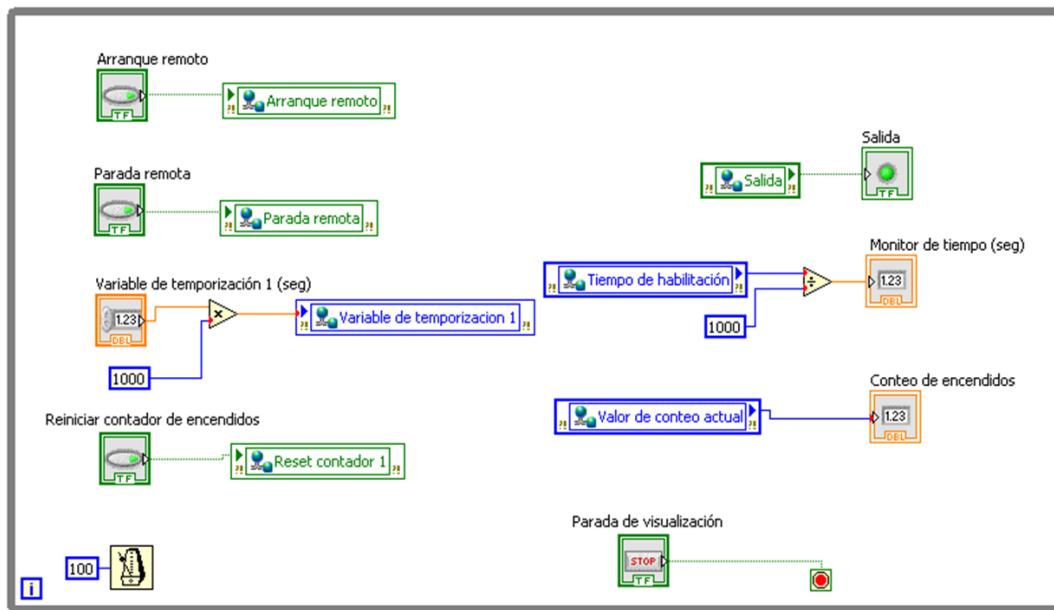
Finalizado lo referente al servidor OPC, diríjase a LabVIEW y agregue las nuevas variables que serán implementadas en el VI, dichas variables son: *Tiempo de habilitación* direccionada al tag “*Temporizador\_1.ACC*”, *Variable de temporización 1* direccionada al tag “*Temporizador\_1.PRE*”, *Valor de conteo actual* direccionada al tag “*Contador\_1.ACC*” y por ultimo *Reset contador 1* direccionada al tag “*Reset\_contador\_1*”.

En la **Figura 63** puede verse las nuevas variables anexadas a la librería.



**Figura 63.** Nuevas variables añadidas a la librería.

Remítase tanto al *Front Panel* y *Block Diagram* para realizar el rediseño y programación del nuevo VI en función de las nuevas variables y aplicando la misma metodología enseñada en las prácticas posteriores, en las **Figuras 64** y **65** puede verse finalizado el VI.



**Figura 64.** *Block Diagram* del VI con las etapas de temporización y conteo añadidas.



**Figura 65.** *Front Panel* del VI con las etapas de temporización y conteo añadidas.

## **PRÁCTICA N° 4**

*Integración de dos PLC Siemens S7-1200.*

## ÍNDICE GENERAL

	Pág.
1. OBJETIVO GENERAL .....	1
2. LISTA DE MATERIALES Y EQUIPOS.....	1
3. FUNDAMENTACIÓN TEÓRICA .....	1
3.1. Dispositivos de redes de comunicación industrial.....	1
3.2. Tipos de redes según su topología .....	3
3.3. Tipos de redes según su extensión .....	6
3.4. Modos de dialogo.....	7
3.5. PROFINET .....	8
3.5.1. Conexión Local/Interlocutor.....	10
3.5.2. ID de conexión para las instrucciones PROFINET .....	11
3.5.3. TCP y ISO on TCP .....	12
3.5.4. Instrucciones TSEND_C y TRCV_C .....	13
3.6. Instrucciones de transferencia y comparación .....	17
4. PROCEDIMIENTO .....	20
4.1. Configuración de la red .....	20
4.2. Comunicación unidireccional entre los PLCs .....	25
4.3. Comunicación bidireccional entre los PLCs .....	40
4.4. Transmisión de paquetes controlada y supervisada mediante una HMI.....	53
4.5. Diseño de un secuenciador doble controlado y supervisado mediante una HMI.....	62

## ÍNDICE DE FIGURAS

	Pág.
Figura 1. Bloque de transferencia MOVE.....	18
Figura 2. Bloque de comparación “igual”. .....	18
Figura 3. Bloque de comparación IN_RANGE .....	19
Figura 4. Bloque de comparación IN_RANGE .....	19
Figura 5. Elementos conectados en red (topología en estrella). ....	20
Figura 6. Configuración de la dirección IP del ordenador. ....	21
Figura 7. Configuración de la dirección IP del router. ....	22
Figura 8. Configuración de la subred y dirección IP del PLC_1. ....	23
Figura 9. Cargar configuraciones hechas en PLC_1.....	23
Figura 10. Configuración de la subred y dirección IP del PLC_2 .....	24
Figura 11. Conexión PROFINET (subred configurada) entre los PLCs. ....	24
Figura 12. Elementos conectados y configurados en red. ....	25
Figura 13. Configuración del byte de marcas de ciclo PLC_1.....	25
Figura 14. Comunicación unidireccional entre los PLCs. ....	26
Figura 15. Bloque TSEND_C dispuesto en <i>Main [OB1]</i> del PLC_1. ....	27
Figura 16. Configuración de <i>Parámetros de la conexión</i> del bloque TSEND_C <i>Envío de datos PLC_1</i> . ....	28
Figura 17. Configuración de los parámetros Locales (PLC_1).....	29
Figura 18. Configuración de los parámetros del Interlocutor (PLC_2). ....	29
Figura 19. Configuración del parámetro <i>REQ</i> .....	30
Figura 20. Configuración del parámetro <i>CONT</i> .....	30
Figura 21. Configuración del parámetro <i>CONNECT</i> .....	30
Figura 22. Configuración del parámetro <i>DATA</i> .....	31
Figura 23. Configuración de <i>Parámetros del bloque TSEND_C Envío de datos PLC_1</i> .....	31
Figura 24. Bloque TSEND_C <i>Envío de datos PLC_1</i> , configurado.....	32
Figura 25. Bloque TRCV_C dispuesto en <i>Main [OB1]</i> del PLC_2.....	32

Figura 26. Configuración de <i>Parámetros de la conexión</i> del bloque TRCV_C <i>Recepción de datos PLC_2</i> .....	33
Figura 27. Configuración de los parámetros Locales (PLC_2).....	34
Figura 28. Configuración de los parámetros del Interlocutor (PLC_1). .....	34
Figura 29. Configuración del parámetro <i>EN_R</i> .....	35
Figura 30. Configuración del parámetro <i>CONT</i> .....	35
Figura 31. Configuración del parámetro <i>CONNECT</i> . .....	35
Figura 32. Configuración del parámetro <i>DATA</i> .....	36
Figura 33. Configuración de <i>Parámetros del bloque TRCV_C Recepción de datos PLC_2</i> .....	36
Figura 34. Bloque TRCV_C <i>Recepción de datos PLC_2</i> , configurado. ....	37
Figura 35. Cambio de formato de visualización del parámetro <i>DATA</i> .....	38
Figura 36. Visualización del dato en formato Decimal y Hex. .....	38
Figura 37. Forzado de operando (MW10). .....	39
Figura 38. Comunicación unidireccional establecida y verificada. .....	39
Figura 39. Comunicación bidireccional entre los PLCs. .....	40
Figura 40. Bloque TSEND_C dispuesto en <i>Main [OB1]</i> del PLC_2. ....	41
Figura 41. Configuración de <i>Parámetros de la conexión</i> del bloque TSEND_C <i>Envío de datos PLC_2</i> .....	42
Figura 42. Configuración de los parámetros Locales (PLC_2).....	42
Figura 43. Configuración de los parámetros del Interlocutor (PLC_1). .....	43
Figura 44. Configuración del parámetro <i>REQ</i> .....	43
Figura 45. Configuración del parámetro <i>CONT</i> .....	44
Figura 46. Configuración del parámetro <i>CONNECT</i> . .....	44
Figura 47. Configuración del parámetro <i>DATA</i> .....	44
Figura 48. Configuración de <i>Parámetros del bloque TSEND_C Envío de datos PLC_2</i> .....	45
Figura 49. Bloque TSEND_C <i>Envío de datos PLC_2</i> , configurado.....	45
Figura 50. Bloque TRCV_C dispuesto en <i>Main [OB1]</i> del PLC_1.....	46
Figura 51. Configuración de <i>Parámetros de la conexión</i> del bloque TRCV_C <i>Recepción de datos PLC_1</i> .....	47

Figura 52. Configuración de los parámetros Locales (PLC_1).....	48
Figura 53. Configuración de los parámetros del Interlocutor (PLC_2). .....	48
Figura 54. Configuración del parámetro <i>EN_R</i> .....	49
Figura 55. Configuración del parámetro <i>CONT</i> .....	49
Figura 56. Configuración del parámetro <i>CONNECT</i> . ....	49
Figura 57. Configuración del parámetro <i>DATA</i> . ....	49
Figura 58. Configuración de <i>Parámetros del bloque TRCV_C Recepción de datos PLC_1</i> .....	50
Figura 59. Comunicación bidireccional establecida y verificada entre los PLCs (1)...	51
Figura 60. Comunicación bidireccional establecida y verificada entre los PLCs (2)...	52
Figura 61. Transmisión unidireccional de un arreglo de datos.....	53
Figura 62. Creación de <i>Bloque de datos_1</i> .....	54
Figura 63. Ajuste de optimización del <i>Bloque de datos_1</i> .....	55
Figura 64. Creación del arreglo <i>DatoPLC1</i> dentro del <i>Bloque de datos_1</i> (PLC_1). .	55
Figura 65. Creación del arreglo <i>DatoPLC2</i> dentro del <i>Bloque de datos_1</i> (PLC_2). .	56
Figura 66. Bloque TSEND_C <i>Envío de datos PLC_1</i> para la transferencia del arreglo <i>DatoPLC1</i> .....	56
Figura 67. Bloque TRCV_C <i>Recepción de datos PLC_2</i> .....	57
Figura 68. <i>Tabla de observación_1</i> del PLC_1. ....	57
Figura 69. <i>Tabla de observación_1</i> del PLC_2. ....	58
Figura 70. PLCs dispuestos y configurados en el servidor OPC.....	58
Figura 71. <i>Tags</i> creados en PLC_1. ....	59
Figura 72. <i>Tags</i> creados en PLC_2. ....	59
Figura 73. Variables agregadas a la librería.....	60
Figura 74. Interfaz para el control y supervisión del paquete de datos. ....	60
Figura 75. <i>Block Diagram</i> de la Interfaz para el control y supervisión del paquete de datos enviados/recibidos.....	61
Figura 76. Verificación del correcto monitoreo de los <i>tags</i> del PLC_1.....	61
Figura 77. Verificación del correcto monitoreo de los <i>tags</i> del PLC_1.....	62
Figura 78. Interfaz para el control y supervisión del paquete de datos en ejecución.	62

Figura 79. Esquema de funcionamiento del secuenciado doble (transmisión bidireccional).....	64
Figura 80. Segmento 1 (PLC_1). .....	65
Figura 81. Segmento 2 (PLC_1). .....	65
Figura 82. Segmento 3 (PLC_1). .....	66
Figura 83. Segmento 3 (PLC_1). .....	66
Figura 84. Segmento 5 (PLC_1). .....	66
Figura 85. Segmento 6 (PLC_1). .....	67
Figura 86. Segmento 7 (PLC_1). .....	67
Figura 87. Segmento 8 (PLC_1). .....	67
Figura 88. Segmento 9 (PLC_1). .....	68
Figura 89. Segmento 10 (PLC_1). .....	68
Figura 90. Segmento 11 (PLC_1). .....	69
Figura 91. Segmento 12 (PLC_1). .....	70
Figura 92. Segmento 13 (PLC_1). .....	71
Figura 93. Segmento 14 (PLC_1). .....	72
Figura 94. Segmento 15 (PLC_1). .....	72
Figura 95. Segmento 1 (PLC_2). .....	74
Figura 96. Segmento 2 (PLC_2). .....	74
Figura 97. Segmento 3 (PLC_2). .....	75
Figura 98. Segmento 4 (PLC_2). .....	75
Figura 99. Segmento 5 (PLC_2). .....	75
Figura 100. Segmento 6 (PLC_2). .....	76
Figura 101. Segmento 7 (PLC_2). .....	76
Figura 102. Segmento 8 (PLC_2). .....	76
Figura 103. Segmento 9 (PLC_2). .....	77
Figura 104. Segmento 10 (PLC_2). .....	77
Figura 105. Segmento 11 (PLC_2). .....	78
Figura 106. Segmento 12 (PLC_2). .....	78
Figura 107. Segmento 13 (PLC_2). .....	79
Figura 108. Segmento 14 (PLC_2). .....	80

Figura 109. Canal 1 y dispositivos (PLCs) configurados.....	81
Figura 110. Tags creados en PLC_1.....	82
Figura 111. Tags creados en PLC_2.....	83
Figura 112. Variables agregadas a la librería (1).....	83
Figura 113. Variables agregadas a la librería.....	84
Figura 114. Interfaz para el control y supervisión del proceso (secuenciador doble) y comunicación entre los PLCs.....	85
Figura 115. Block Diagram 1 de la interfaz para el control y supervisión del proceso (secuenciador doble) y comunicación entre los PLCs.....	86
Figura 116. Block Diagram 2 de la interfaz para el control y supervisión del proceso (secuenciador doble) y comunicación entre los PLCs.....	87
Figura 117. Interfaz para el control y supervisión del proceso (secuenciador doble) y comunicación entre los PLCs en ejecución.....	88

## ÍNDICE DE TABLAS

	Pág.
Tabla 1. Instrucciones TSEND_C y TRCV_C .....	14
Tabla 2. Tipos de datos TSEND_C y TRCV_C para los parámetros. ....	15
Tabla 3. Códigos de condición TSEND_C y TRCV_C para ERROR y STATUS (1)..	16
Tabla 4. Códigos de condición TSEND_C y TRCV_C para ERROR y STATUS (2)..	17
Tabla 5. Descripciones de comparaciones. ....	18

## 1. OBJETIVO GENERAL

Comunicar, supervisar y controlar los PLC Siemens SIMATIC S7-1200.

## 2. LISTA DE MATERIALES Y EQUIPOS

- Ordenador con los programa TIA Portal, National Instruments OPC Server y LabVIEW con los módulos Datalogging and Supervisory Control (DSC) y Real Time integrados.
- Dos (2) PLCs Siemens SIMATIC S7-1200 1214C AC/DC/RLY.
- Cable UTP RJ45/RJ45.
- Router.

## 3. FUNDAMENTACIÓN TEÓRICA

### 3.1. Dispositivos de redes de comunicación industrial

Estos dispositivos son utilizados para procesar señales de los equipos en los procesos industriales. Entre los dispositivos se tienen:

- **Repetidor:** equipo que actúa a nivel físico, prolongando la longitud de la red mediante la unión de dos segmentos de red y amplificando la señal. Junto con la señal se amplifica el ruido; por esta razón el número de estaciones que pueden compartir el medio es limitado.

- **Concentrador (*Hub*):** consiste en una serie de puntos de conexión que son bidireccionales y cuya función es la de reenvío de los paquetes de información que llegan por un puerto cualquiera hacia el resto de los puertos, de forma tal que todos los equipos que estén conectados a este dispositivo reciban los paquetes de datos. Los *Hubs* se utilizan para crear un punto de conexión central para los medios de cableado (red en estrella). Los *Hubs* pueden ser activos (amplifican la señal, permitiendo conectar equipos a distancias de hasta 609 metros) o pasivos (reparten la señal conectando equipos a distancias de hasta 30 metros).
- **Puente (*Bridge*):** equipo que interconecta dos segmentos de red, utilizando el mismo protocolo de comunicación para pasar datos de un segmento hacia el otro segmento con base en la dirección física de destino de cada paquete de datos. Un *bridge* solo pasa las tramas pertenecientes a cada segmento de la red; por esta característica se mejora el rendimiento de las redes al disminuir el tráfico de datos inútiles.
- **Comutador (*Switch*):** es un *bridge* con múltiples puertos. Analiza las tramas que le llegan y genera unas tablas de tráfico que le permiten dirigir cada trama únicamente al puerto de salida donde se encuentre el equipo destinatario optimizando así el ancho de banda de la red y evitando las colisiones.
- **Encaminador o enrutador (*Router*):** se basa en el funcionamiento de los *bridges*; es un equipo de interconexión de redes que actúa a nivel de los protocolos de red (capa tres) del modelo de referencia OSI, determinando las rutas más eficientes que deberá seguir la información. Cuando llega un paquete a un router, este lee la dirección del destinatario y envía el paquete al siguiente nodo próximo a la dirección de destino (por ejemplo, otro router que realiza el mismo proceso). Existen router que son también *switch* con cuatro (4) puertos y punto de acceso WIFI.

- **Puerta de enlace (Gateway):** son equipos para interconectar redes con protocolos de comunicación y arquitecturas diferentes. Su propósito es traducir la información del protocolo utilizando en la red de origen al protocolo usado en la red de destino.

### 3.2. Tipos de redes según su topología

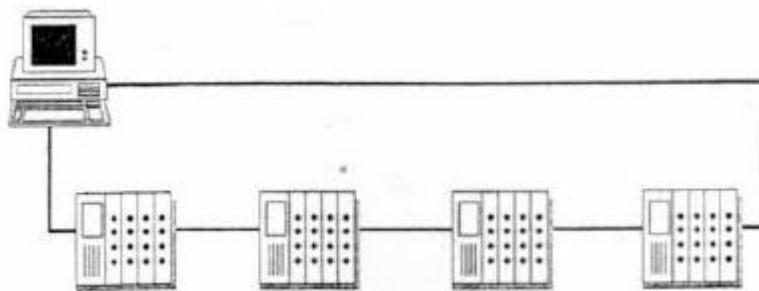
La topología define la disposición de diferentes equipos alrededor del medio de transmisión de datos determinando unas estructuras de red características. Entre las topologías más comunes se encuentran:

- **Anillo:** el medio de transmisión forma un circuito cerrado (anillo) al que se conectan los equipos, ver **Figura 1**. Las principales ventajas de este tipo de topología son:
  - ✓ Los requerimientos de cable son mínimos, similares a los de la topología bus.
  - ✓ Se basa en una serie de conexiones punto a punto de una estación con la siguiente.
  - ✓ El modo de transmisión se organiza por turnos mediante el paso de un permiso de transmisión de una estación a otra.
  - ✓ El mensaje vuelve al emisor.
  - ✓ El tráfico de información va en un sentido único a lo largo del soporte de transmisión.

- ✓ Es una estructura activa, la señal se regenera en cada nodo.

Los posibles puntos débiles son:

- ✓ La caída de un equipo interrumpe el tráfico de información.
- ✓ Añadir y quitar nodos afecta la red.



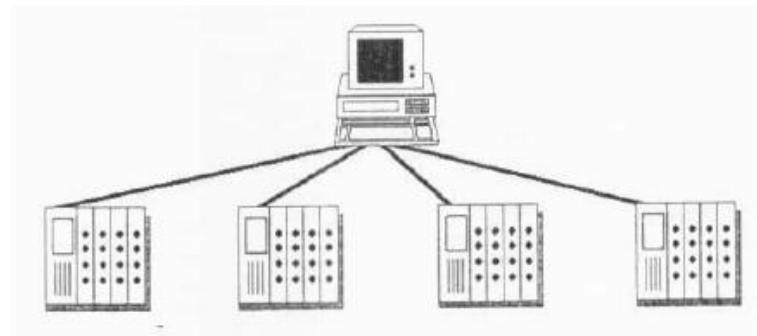
**Figura 1.** Topología Anillo.

- **Estrella:** en esta configuración todos los equipos están conectados a un equipo o nodo central (HUB del inglés *Host Unit Broadcast*) que realiza las funciones de control y coordinación, ver **Figura 2**.

Las características principales son:

- ✓ La transferencia de información es punto a punto.
- ✓ La sencillez de su mantenimiento.
- ✓ El equipo central, el HUB, controla toda la red.
- ✓ El rendimiento de la red (velocidad) dependerá del HUB.

- ✓ Si el HUB se para, la red queda inutilizada.
- ✓ La ampliación del sistema está limitada por la capacidad del nodo central o HUB.



**Figura 2.** Topología Estrella.

- **Bus:** en esta topología la distribución básica se realiza alrededor de un segmento de cable al cual se conectan los equipos, ver **Figura 3**.

El modo de transmisión es aleatorio, un equipo transmite cuando lo necesita. Si hay transmisiones simultáneas (colisiones), unos algoritmos especiales solventan el problema.

Entre las características más destacables se encuentran las siguientes:

- ✓ Necesita menor longitud de cable en comparación con otras redes similares en extensión.
- ✓ Las conexiones de alta impedancia permiten conectar y desconectar elementos de forma sencilla. Por tanto, la caída de un equipo no afecta al resto de la red.

- ✓ Elevada velocidad de transmisión.

Las posibles desventajas son:

- ✓ Falta de seguridad, pues cualquier nodo puede variar cualquier mensaje (aunque no sea destinatario)
- ✓ El diagnóstico puede ser difícil debido a la estructura física, un fallo eléctrico puede estar en cualquier punto del bus.
- ✓ En casos de sobrecarga de tráfico puede bajar el rendimiento.

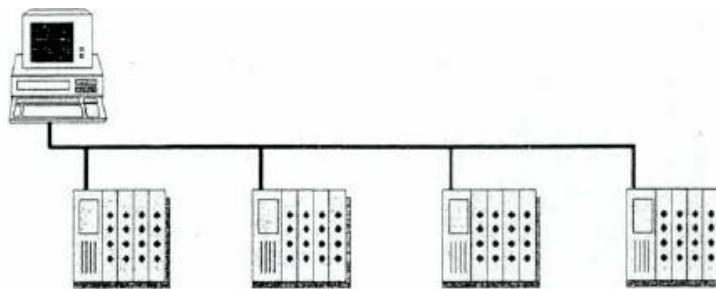


Figura 3. Topología Bus.

### 3.3. Tipos de redes según su extensión

- **WAN (Wide Area Network):** cubre necesidades internacionales (reserva de vehículos de alquiler) o nacionales (seguridad nacional).
- **MAN (Metropolitan Area Network):** cubre necesidades a escala de una ciudad (gestión de edificios municipales).

- **LAN (Local Area Network):** son las conocidas redes locales. Son geográficamente limitadas (sobre 1km de radio), y permite interconectar de forma sencilla ordenadores situados en edificios próximos, que pueden ser de uso industrial, terciario o doméstico (Ethernet o FDDI).

También hay redes locales para distancias muy pequeñas (centímetros). Es el caso de los ordenadores personales, en los cuales los elementos están conectados mediante enlace paralelo de alta velocidad (PCI, AGP, VME).

### 3.4. Modos de dialogo

Entre las posibilidades dialogo se encuentran:

- **Modo de comunicación Simplex:** el modo simplex es aquel en el cual hay un emisor y un receptor de datos, y la información solamente fluye del primero al segundo. No es de interés industrial en el sentido en que se trata de un sistema en lazo abierto (no se reciben datos del elemento al cual se le manda la información).
- **Modo de comunicación Half-Duplex:** las comunicaciones de este tipo definen aquel dialogo que se realiza entre dos puntos, en las dos direcciones, pero no de forma simultánea, sino por turnos.
- **Modo de comunicación Duplex (Full-Duplex):** con este método, el intercambio de información es bidireccional y simultaneo. Esto es gracias a que hay un canal para emitir y otro para recibir.

### 3.5. PROFINET

Permite una alta gama de conectividad entre equipos de diferentes niveles cubriendo la total comunicación en una planta industrial. Sus prestaciones son muy flexibles, gracias a esto ha podido ser utilizado en la industria automotriz, industria de alimentos y bebidas, entre otras que a pesar de ser procesos de diferentes pueden solucionarse con PROFINET. A continuación se presentan ciertas características del bus.

- En cuanto a fiabilidad
  - ✓ Redundancia de anillo rápida (< 200 ms).
  - ✓ WLAN con reserva de ancho de banda, *rapid roaming*.
  - ✓ Equipos modulares con sustituciones en caliente.
  - ✓ Consta de una total integración, diagnosis de red integrada en PLC, HMI.
- Características de aplicación industrial
  - ✓ Consta de una total integración, diagnosis de red integrada en PLC, HMI.
  - ✓ Facilidad de mantenimiento
  - ✓ Sustitución de equipos por personal no especializado (elementos extraíbles).
  - ✓ Sistema de cableado industrial rápido e inmune a ruidos.
- Uso fiable en entornos severos

- ✓ No se producen reflexiones en cables ni problemas de cableado.
- ✓ Los problemas de conexión solo afectan a dos puntos de un enlace y no a la línea entera (fácil localización de fallas).
- Tecnologías robustas para PROFINET
  - ✓ El uso de Ethernet conmutada siempre permite una conexión punto a punto
  - ✓ El uso de cables apantallados garantiza inmunidad frente a influencias electromagnéticas.
  - ✓ La auto-negación y el auto-crossing reducen los errores potenciales durante la puesta en marcha y mantenimiento.
- Velocidad de transmisión
  - ✓ Debido a su amplio ancho de banda, aplicaciones intensivas en datos se pueden usar paralelamente sin que ello afecte a la transmisión de datos de periferia.
  - ✓ Usando 100 Mbit/s Ethernet, PROFINET alcanza una velocidad de transmisión significativamente alta.
  - ✓ 100 Mbit/s es una prestación suficientemente alta para el nivel de campo.

### 3.5.1. Conexión Local/Interlocutor

Una conexión local/interlocutor (remoto) define una asignación lógica de dos interlocutores para establecer servicios de comunicación. Una conexión define lo siguiente:

- Interlocutores involucrados (uno activo, otro pasivo).
- Tipo de conexión (por ejemplo una conexión de PLC, HMI o dispositivo).
- Ruta de conexión.

Los interlocutores ejecutan las instrucciones necesarias para configurar y establecer la conexión. El punto final activo y el punto final pasivo de la comunicación se especifican median parámetros. Una vez configurada y establecida la conexión, la CPU la mantiene y vigila automáticamente.

La CPU se puede comunicar con otras CPUs, con programadoras, con dispositivos HMI y con dispositivos no Siemens que utilicen protocolos de comunicación TCP estándar.

El puerto PROFINET de las CPU 1211C, 1212C y 1214C no contiene ningún switch Ethernet. Una conexión directa entre una programadora o HMI y una CPU no requiere un comutador Ethernet. Sin embargo, una red con más de dos CPUs o dispositivos HMI si requiere un comutador Ethernet.

### 3.5.2. ID de conexión para las instrucciones PROFINET

Cuando se insertan las instrucciones PROFINET TSEND\_C y TRCV\_C en el programa de usuario, STEP 7 crea un DB de instancia para configurar el canal de comunicaciones (o conexión) entre los dispositivos. Utilice las propiedades de la instrucción para configurar los parámetros de la conexión. Entre los parámetros se encuentra la ID de conexión.

- La ID de conexión debe ser unívoca para la CPU. Cada conexión que se cree debe tener un DB y una ID de conexión distintos.
- Tanto la CPU local como la CPU interlocutora pueden utilizar el mismo número de ID de conexión para la misma conexión, aunque los números de ID de conexión no tienen que coincidir necesariamente. El número de ID de conexión solo es relevante para las instrucciones PROFINET dentro del programa de usuario de la CPU individual.
- Pueden utilizarse cualquier número para la ID de conexión de la CPU. Sin embargo, si se configuran las ID de conexión secuencialmente empezando por "1", es más fácil saber el número de conexión que se están utilizando en una CPU concreta.
- Cada instrucción TSEN\_C y TRCV\_C del programa de usuario crea una conexión nueva. Es importante utilizar la ID de conexión correcta para cada conexión.

### 3.5.3. TCP y ISO on TCP

Transport Control Protocol (TCP) es un protocolo estándar descrito por RFC 793: Transmission Control Protocol. El objetivo principal de TCP es ofrecer un servicio de conexión seguro y fiable entre pares de procesos. Este protocolo tiene las características siguientes:

- Protocolo de comunicación eficiente puesto que está vinculado estrechamente al hardware.
- Adecuado para cantidades de datos medianas y grandes (hasta 8192 bytes).
- Ofrece numerosas prestaciones más a las aplicaciones, además de una recuperación de errores, control de flujo y fiabilidad considerables.
- Protocolo orientado a la conexión.
- Puede utilizarse muy flexiblemente con sistemas de terceros que soporten únicamente TCP.
- Apto para routing.
- Son aplicables sólo las longitudes de datos estáticas.
- Los mensajes se acusan.
- Las aplicaciones se dirigen usando números de puerto.

- Es necesario programar la gestión de datos debido a la interfaz de programación SEND / RECEIVE.

International Standards Organization (ISO) on Transport Control Protocol (TCP) (RFC 1006) (ISO on TCP) es un mecanismo que permite portar aplicaciones ISO a la red TCP/IP. Este protocolo tiene las características siguientes:

- Protocolo de comunicación eficiente vinculado estrechamente al hardware.
- Adecuado para cantidades de datos medianas y grandes (hasta 8192 bytes)
- A diferencia de TCP, los mensajes tienen un indicador de fin y están orientados a los mensajes.
- Apto para routing; puede utilizarse en WAN.
- Las longitudes de datos dinámicas son posibles.
- Es necesario programar la gestión de datos debido a la interfaz de programación SEND / RECEIVE.

### 3.5.4. Instrucciones TSEND\_C y TRCV\_C

La instrucción TSEND\_C combina las funciones de las instrucciones TCON, TDISCON y TSEND. La instrucción TRCV\_C combina las funciones de las instrucciones TCON, TDISCON y TRCV.

El tamaño mínimo de los datos que pueden transmitirse (TSEND\_C) o recibirse (TRCV\_C) es de un byte; el tamaño máximo es de 8192 bytes. TSEND\_C no soporta la transmisión de datos desde posiciones booleanas y TRCV\_C no recibe datos en posiciones booleanas.

En las **Tablas 1 y 2** aparecen descritas las instrucciones TSEND\_C y TRCV\_C.

**Tabla 1.** Instrucciones TSEND\_C y TRCV\_C.

KOP / FUP	SCL	Descripción
	<pre>"TSEND_C_DB" TSEND_C - EN           ENO - REQ          DONE - CONT         BUSY - LEN          ERROR - CONNECT      STATUS - DATA - COM_RST</pre>	<p>TSEND_C establece una conexión de comunicación TCP o ISO-on-TCP con un interlocutor, envía datos y puede deshacer la conexión. Una vez configurada y establecida la conexión, la CPU la mantiene y la vigila automáticamente.</p>
	<pre>"TRCV_C_DB" TRCV_C - EN           ENO - EN_R         DONE - CONT         BUSY - LEN          ERROR - CONNECT      STATUS - DATA - COM_RST</pre>	<p>TRCV_C establece una conexión de comunicación TCP o ISO-on-TCP con una CPU interlocutora, recibe datos y puede deshacer la conexión. Una vez configurada y establecida la conexión, la CPU la mantiene y la vigila automáticamente.</p>

Es de importancia mencionar que la instrucción TSEND\_C requiere una transición de bajo a alto (pulso) en el parámetro de entrada REQ para iniciar una tarea de transmisión.

**Nota:** La CPU soporta un número máximo de 8 conexiones asíncronas simultáneas TSEND\_C y TRCV\_C.

**Tabla 2.** Tipos de datos TSEND\_C y TRCV\_C para los parámetros.

Parámetro y tipo	Tipo de datos	Descripción
REQ (TSEND_C)	IN	Bool El parámetro de control REQ inicia la tarea de transmisión con la conexión descrita en CONNECT cuando se detecta un flanko ascendente.
EN_R (TRCV_C)	IN	Bool Parámetro de control habilitado para recibir: Si EN_R = 1, TRCV_C está preparado para recibir. La tarea de recepción se procesa.
CONT	IN	Bool <ul style="list-style-type: none"> <li>• 0: Desconectar</li> <li>• 1: Establecer y mantener la conexión</li> </ul>
LEN	IN	UInt Número máximo de bytes que deben enviarse (TSEND_C) o recibirse (TRCV_C): <ul style="list-style-type: none"> <li>• Predeterminado = 0: El parámetro DATA determina la longitud de datos transmitidos (TSEND_C) o recibidos (TRCV_C).</li> <li>• Modo Ad hoc = 65535: Para la recepción se especifica una longitud de datos variable (TRCV_C).</li> </ul>
CONNECT	IN_OUT	TCON_Param Puntero hacia la descripción de la conexión
DATA	IN_OUT	Variant <ul style="list-style-type: none"> <li>• Contiene la dirección y longitud de los datos que se van a enviar (TSEND_C).</li> <li>• Contiene la dirección de inicio y la longitud máxima de los datos recibidos (TRCV_C).</li> </ul>
COM_RST	IN_OUT	Bool Permite reiniciar la instrucción: <ul style="list-style-type: none"> <li>• 0: Irrelevante</li> <li>• 1: Reinicio completo del bloque de función; se deshace la conexión existente.</li> </ul>
DONE	OUT	Bool <ul style="list-style-type: none"> <li>• 0: Tarea no iniciada aún o en proceso.</li> <li>• 1: Tarea finalizada sin errores.</li> </ul>
BUSY	OUT	Bool <ul style="list-style-type: none"> <li>• 0: Tarea finalizada.</li> <li>• 1: Tarea no finalizada aún. No se puede iniciar una tarea nueva.</li> </ul>
ERROR	OUT	Bool Parámetros de estado con los valores siguientes: <ul style="list-style-type: none"> <li>• 0: No hay error</li> <li>• 1: Ha ocurrido un error durante el procesamiento. STATUS proporciona información detallada sobre el tipo de error.</li> </ul>
STATUS	OUT	Word Información de estado, incluida información de error. (Véase la tabla "Parámetros de error y estado" a continuación.)
RCVD_LEN (TRCV_C)	OUT	Int Cantidad de datos (en bytes) recibida realmente

En las **Tablas 3** y **4** aparecen descritos los códigos de condición TSEND\_C y TRCV\_C para ERROR y STATUS.

**Tabla 3.** Códigos de condición TSEND\_C y TRCV\_C para ERROR y STATUS (1).

ERROR	STATUS	Descripción
0	0000	Tarea ejecutada sin errores
0	7000	No se está procesando ninguna tarea
0	7001	Iniciar procesamiento de la tarea, estableciendo la conexión, esperando al interlocutor
0	7002	Enviando o recibiendo datos
0	7003	Deshaciendo la conexión
0	7004	Conexión establecida y vigilada, no se está procesando ninguna tarea
1	8085	El parámetro LEN es mayor que el valor máximo admisible.
1	8086	El parámetro CONNECT está fuera del rango admisible.
1	8087	Se ha alcanzado el número máximo de conexiones; no es posible establecer más conexiones.
1	8088	El parámetro LEN no es válido para el área de memoria especificada en DATA.
1	8089	El parámetro CONNECT no apunta a un bloque de datos.
1	8091	Se ha excedido la profundidad de anidamiento máxima.
1	809A	El parámetro CONNECT apunta a un campo que no concuerda con la longitud de la descripción de la conexión.
1	809B	La "local_device_id" indicada en la descripción de la conexión no concuerda con la CPU.
1	80A1	Error de comunicación: <ul style="list-style-type: none"> <li>• La conexión indicada no se ha establecido todavía</li> <li>• La conexión indicada se está deshaciendo; no es posible transferir a través de esta conexión</li> <li>• La interfaz se está reinicializando</li> </ul>
1	80A3	Se está intentando deshacer una conexión no existente
1	80A4	La dirección IP del interlocutor remoto de la conexión no es válida. Por ejemplo, la dirección IP del interlocutor remoto es igual a la dirección IP del interlocutor local.
1	80A5	ID de conexión en uso.
1	80A7	Error de comunicación: TDISCON se ha llamado antes de finalizar TSEND_C.
1	80B2	El parámetro CONNECT apunta a un bloque de datos generado con la palabra clave UNLINKED

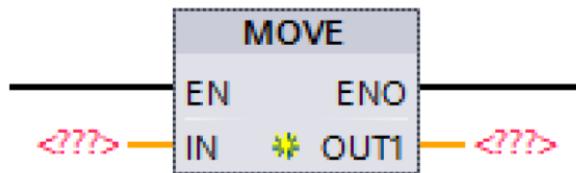
**Tabla 4.** Códigos de condición TSEND\_C y TRCV\_C para ERROR y STATUS (2).

ERROR	STATUS	Descripción
1	80B3	Parámetros incoherentes: <ul style="list-style-type: none"><li>• Error en la descripción de la conexión</li><li>• El puerto local (parámetro local_tsap_id) ya existe en una descripción de conexión diferente.</li><li>• La ID indicada en la descripción de la conexión no concuerda con la ID especificada como parámetro</li></ul>
1	80B4	Si se utiliza ISO on TCP (connection_type = B#16#12) para establecer una conexión pasiva, el código de condición 80B4 advierte de que el TSAP introducido no cumple uno de los siguientes requisitos para la dirección: <ul style="list-style-type: none"><li>• Si la longitud del TSAP local es 2 y el valor de la ID TSAP es E0 o E1 (hexadecimal) para el primer byte, el segundo byte deberá ser 00 ó 01.</li><li>• Si la longitud del TSAP local es 3 o superior y el valor de la ID TSAP es E0 o E1 (hexadecimal) para el primer byte, el segundo byte deberá ser 00 ó 01 y todos los demás bytes deberán ser caracteres ASCII válidos.</li><li>• Si la longitud del TSAP local es 3 o superior y el valor del primer byte de la ID TSAP no es E0 ni E1 (hexadecimal), todos los demás bytes de la ID TSAP deberán ser caracteres ASCII válidos.</li></ul> Los caracteres ASCII válidos son valores de byte comprendidos entre 20 y 7E (hexadecimal).
1	80B7	El tipo de datos y/o la longitud de los datos transmitidos no cabe en el área de la CPU interlocutora en la que deben escribirse.
1	80C3	Se están utilizando todos los recursos de conexión.
1	80C4	Error de comunicación temporal: <ul style="list-style-type: none"><li>• La conexión no se puede establecer en estos momentos</li><li>• La interfaz está recibiendo nuevos parámetros</li><li>• La instrucción TDISCON está deshaciendo la conexión configurada.</li></ul>
1	8722	Parámetro CONNECT: Área de origen no válida: el área no existe en el DB.
1	873A	Parámetro CONNECT: Imposible acceder a la descripción de la conexión (p. ej. porque el DB no está disponible).
1	877F	Parámetro CONNECT: Error interno, p. ej. referencia no válida a ANY
1	893A	El parámetro contiene el número de un DB que no está cargado.

### 3.6. Instrucciones de transferencia y comparación

En el posterior desarrollo de la práctica se hará uso de las siguientes instrucciones las cuales:

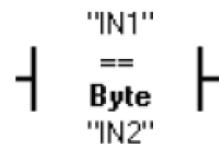
- **MOVE (transferencia):** la instrucción MOVE copia un elemento de datos individual de la dirección de origen que indica el parámetro IN en la dirección de destino que indica el parámetro OUT.



**Figura 4.** Bloque de transferencia MOVE.

- **Comparación**

- ✓ **Instrucción de comparación:** compara varios elementos del mismo tipo de datos. Si la comparación de contactos KOP es true (verdadera), se activa el contacto. Véase la **Figura 5** (bloque de comparación) y **Tabla 5** (descripciones de comparaciones).



**Figura 5.** Bloque de comparación "igual".

**Tabla 5.** Descripciones de comparaciones.

Tipo de relación	La comparación se cumple si ...
=	IN1 es igual a IN2
<>	IN1 es diferente de IN2
>=	IN1 es mayor o igual a IN2
<=	IN1 es menor o igual a IN2
>	IN1 es mayor que IN2
<	IN1 es menor que IN2

- ✓ **Instrucciones IN\_RANGE (dentro del rango) y OUT\_RANGE (fuera del rango):** comprueba si un valor de entrada está dentro o fuera de un rango de valores específicos. Véase la **Figura 6** (bloque IN\_RANGE) y **Figura 7** (bloque OUT\_RANGE).

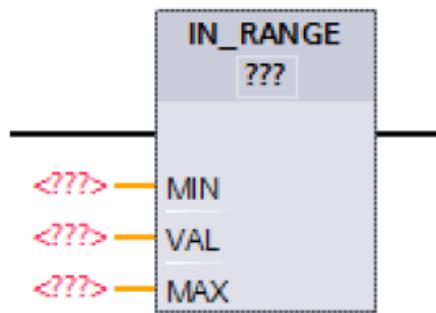


Figura 6. Bloque de comparación IN\_RANGE.

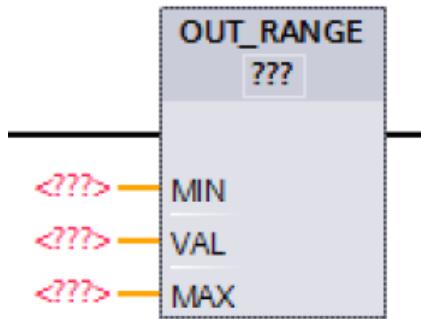
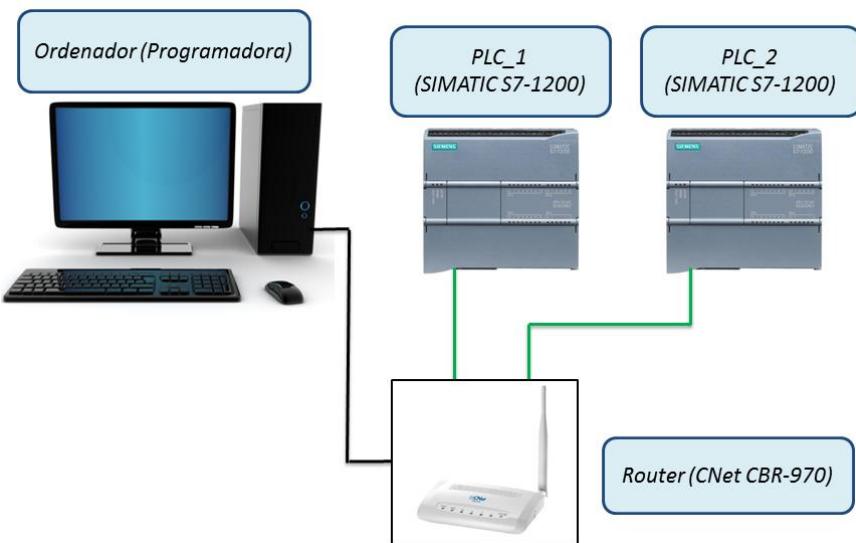


Figura 7. Bloque de comparación IN\_RANGE.

## 4. PROCEDIMIENTO

### 4.1. Configuración de la red

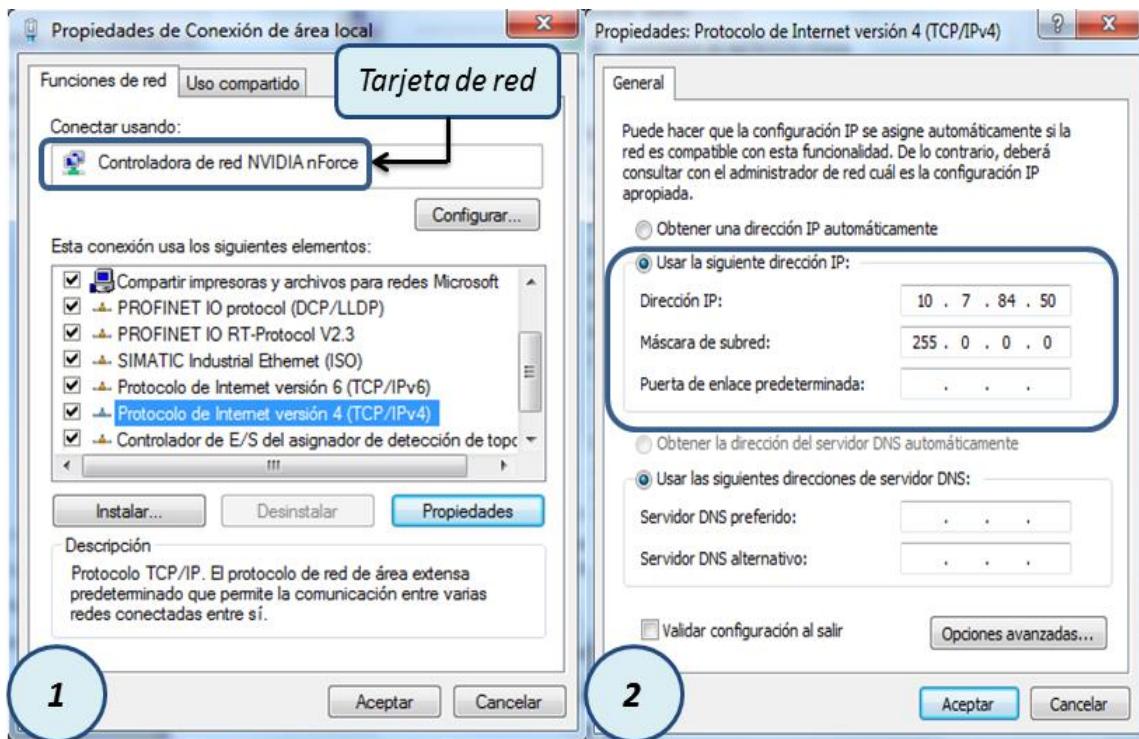
La red sobre la cual se trabajará en el desarrollo de esta práctica estará compuesta de cuatro (4) elementos: dos (2) PLCs SIMATIC S7-1200, un ordenador con tarjeta de red integrada y un enrutador o encaminador (*router CNet CBR-970*). La red será definida por una topología en estrella debido al uso del enrutador, en la **Figura 8** puede observarse el conexionado de los distintos elementos de la red.



**Figura 8.** Elementos conectados en red (topología en estrella).

Conocida la topología y el conexionado de los distintos elementos, se procederá a configurar uno por uno cada elemento para el correcto funcionamiento de la red.

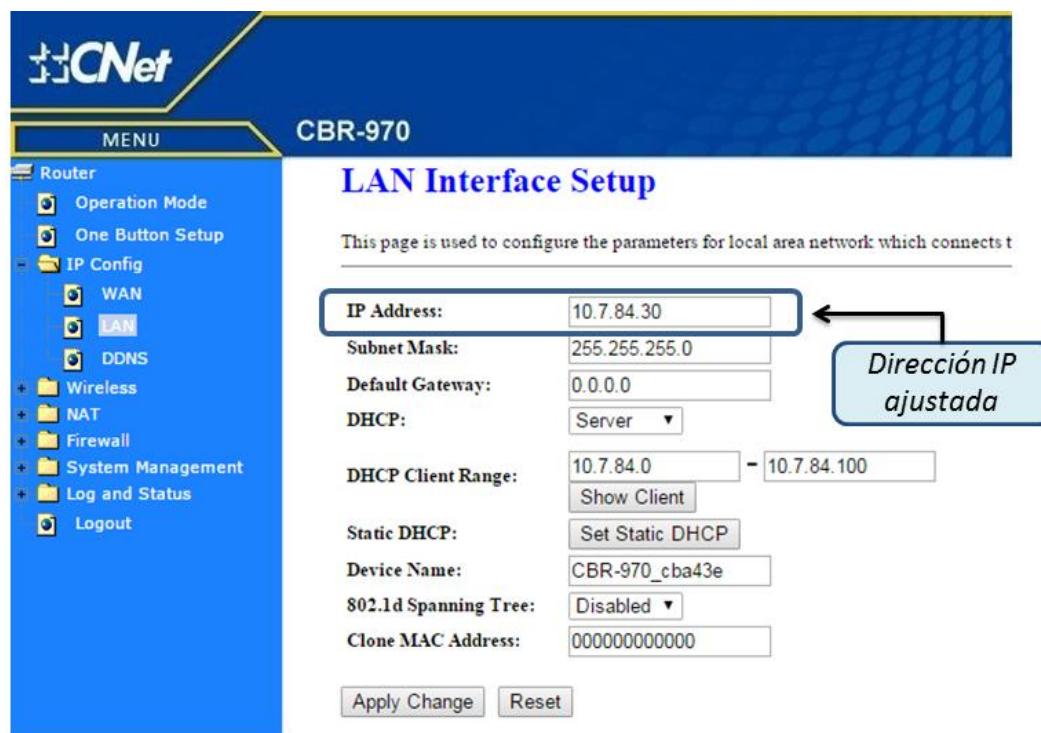
- **Configuración del ordenador (programadora):** acceda a las configuraciones de su tarjeta de red y ajuste en *Protocolo de Internet (TCP/IP)* la dirección IP en **10.7.84.50**, en la **Figura 9** aparece este procedimiento en entorno Microsoft Windows 7.



**Figura 9.** Configuración de la dirección IP del ordenador.

- **Configuración del router CNet CBR-970:** en este caso particular, la red será unificada mediante un *router* de la marca CNet modelo CBR-970, el procedimiento de asignación de la dirección IP en este router es similar al de cualquier otro (marca y modelo).

Remítase al panel de configuraciones del router insertando la dirección IP por defecto del router en algún navegador web (vía http), posteriormente, acceda a las configuraciones de IP de área local (LAN), en este caso *IP Config / LAN* y ajuste la dirección en **10.7.84.30**, en la **Figura 10**, puede evidenciarse parte de este procedimiento.



**Figura 10.** Configuración de la dirección IP del router.

- **Configuración de los PLC SIMATIC S7-1200:** una vez detectado el PLC\_1 independientemente de la dirección IP que posea (pero dentro de la red), acceda a las *Propiedades / General / Interfaz PROFINET / Protocolo IP*, en ella:

En *Interfaz conectada en red con*, pulse el botón *Agregar subred*, por defecto se agregará la subred *PN/IE\_1*.

Ajuste la dirección IP en **10.7.84.40** y seleccione la opción *Utilizar router*, donde deberá agregar la dirección IP del router (previamente definida en **10.7.84.30**).

En la **Figura 11** aparecen los procedimientos descritos anteriormente.

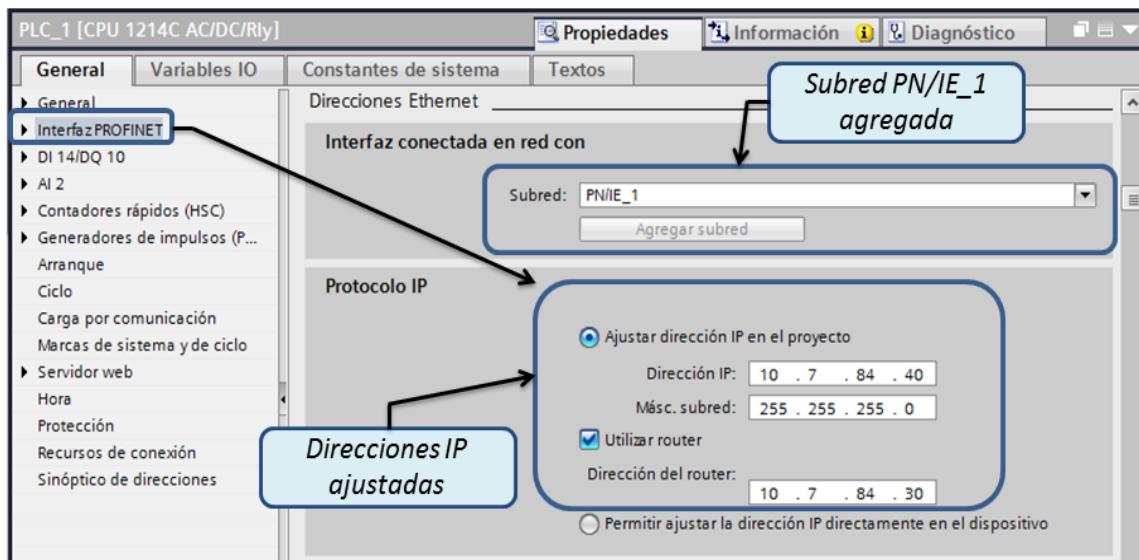


Figura 11. Configuración de la subred y dirección IP del PLC\_1.

Posteriormente, deberá cargar en el dispositivo las modificaciones realizadas, para ello remítase al *Árbol de proyecto / Dispositivos / PLC\_1*, en este último haga clic derecho con el ratón y vaya a *Cargar en dispositivo / Hardware y software (solo cambios)*, siga el procedimiento de carga respectivo y finalice la operación, en la **Figura 12**, puede evidenciarse lo descrito.

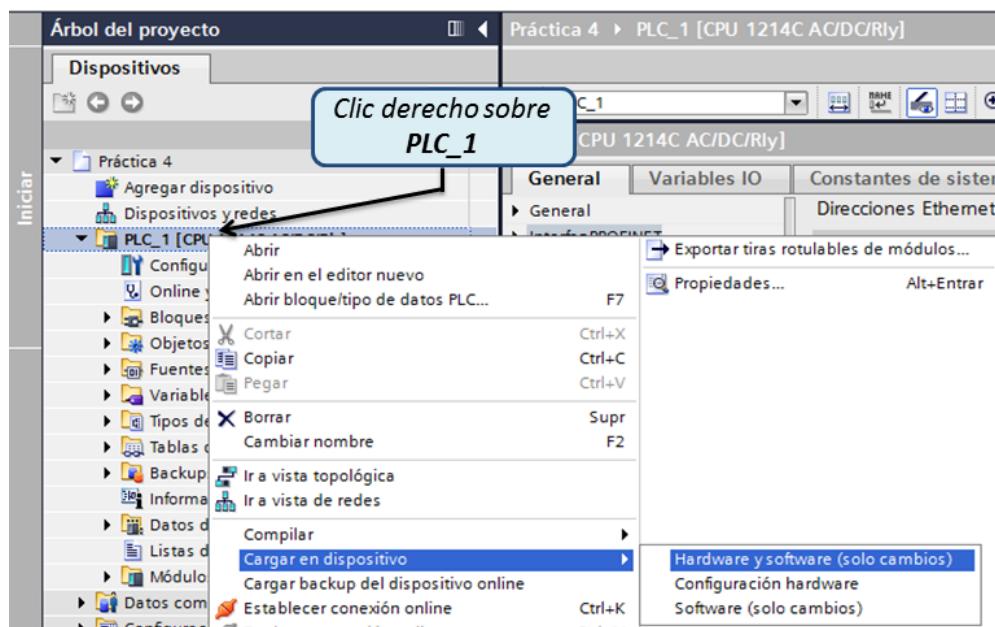
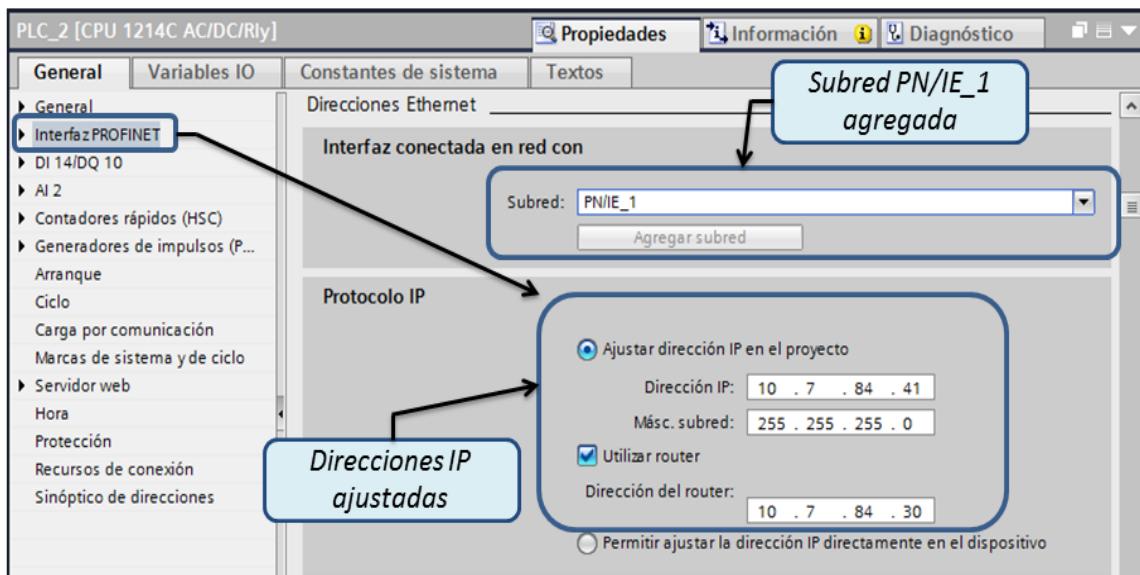


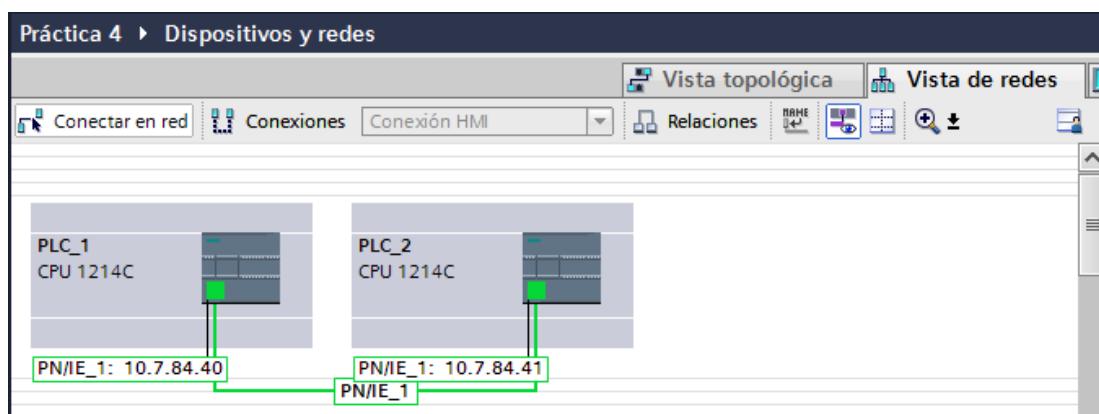
Figura 12. Cargar configuraciones hechas en PLC\_1.

Seguidamente agregue el PLC\_2 (Aregar dispositivo), después vaya a *Interfaz PROFINET* y seleccione en *Interfaz conectada en red con la subred PN/IE\_1*, la modificación del direccionamiento IP es análogo al realizado para el PLC\_1, en este caso el PLC\_2 será configurado con una dirección IP en **10.7.84.41**, ver **Figura 13**.

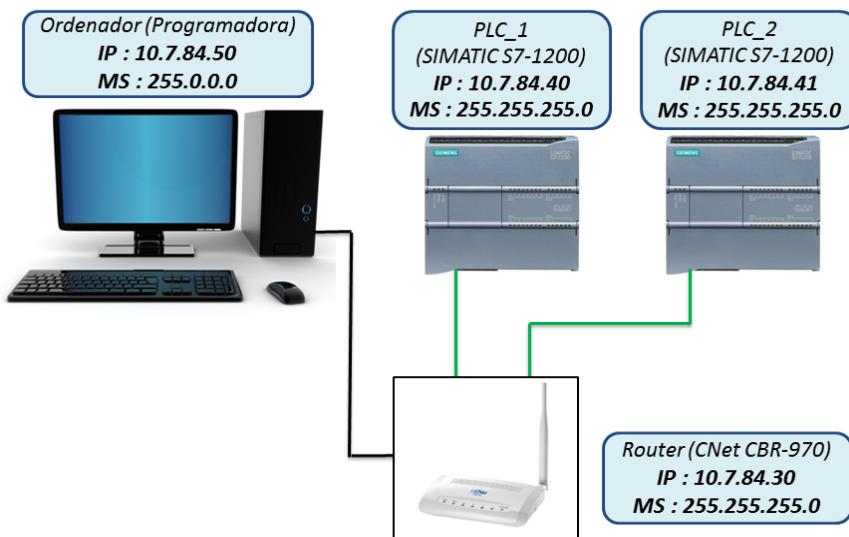


**Figura 13.** Configuración de la subred y dirección IP del PLC\_2.

Finalizada la configuración del PLC\_2 ha logrado integrar con éxito la red, en la **Figura 14** se muestra la conexión PROFINET establecida entre los dos PLCs y en la **Figura 15** es mostrada la red con todos sus equipos integrados.



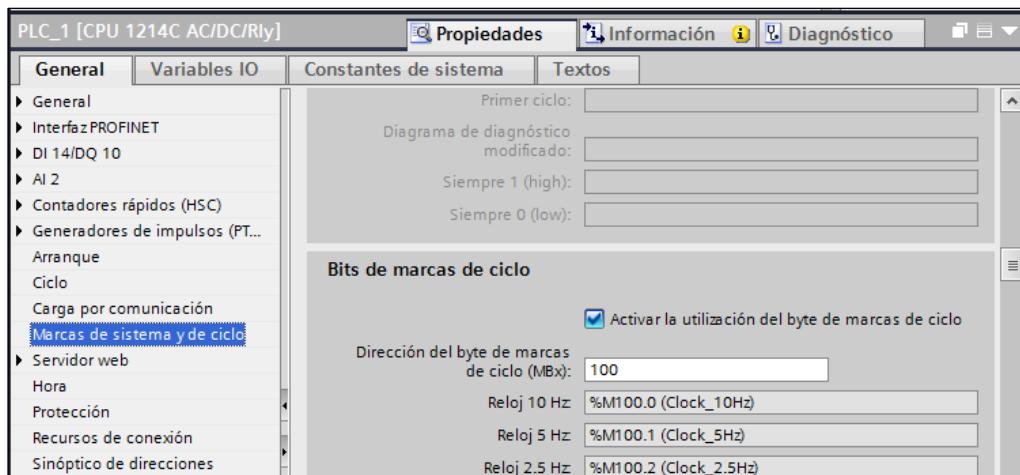
**Figura 14.** Conexión PROFINET (subred configurada) entre los PLCs.



**Figura 15.** Elementos conectados y configurados en red.

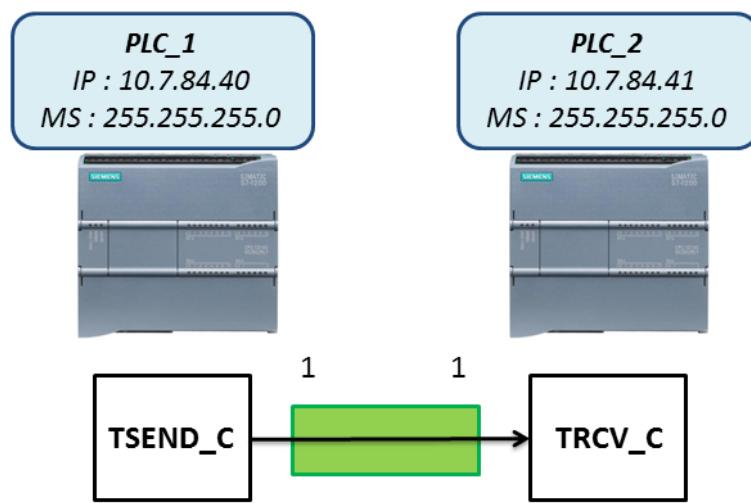
## 4.2. Comunicación unidireccional entre los PLCs

Para comenzar, vaya a *Propiedades / General / Marca de sistema y de ciclo / Bits de marcas de ciclo* y tilde la casilla *Activar la utilización de byte de marcas de ciclo*, aunado a esto en *Dirección del byte de marcas de ciclo (MBx)* coloque cien (100) (usado en este caso, puede seleccionar cualquier otro byte), observe este procedimiento en la **Figura 16**.



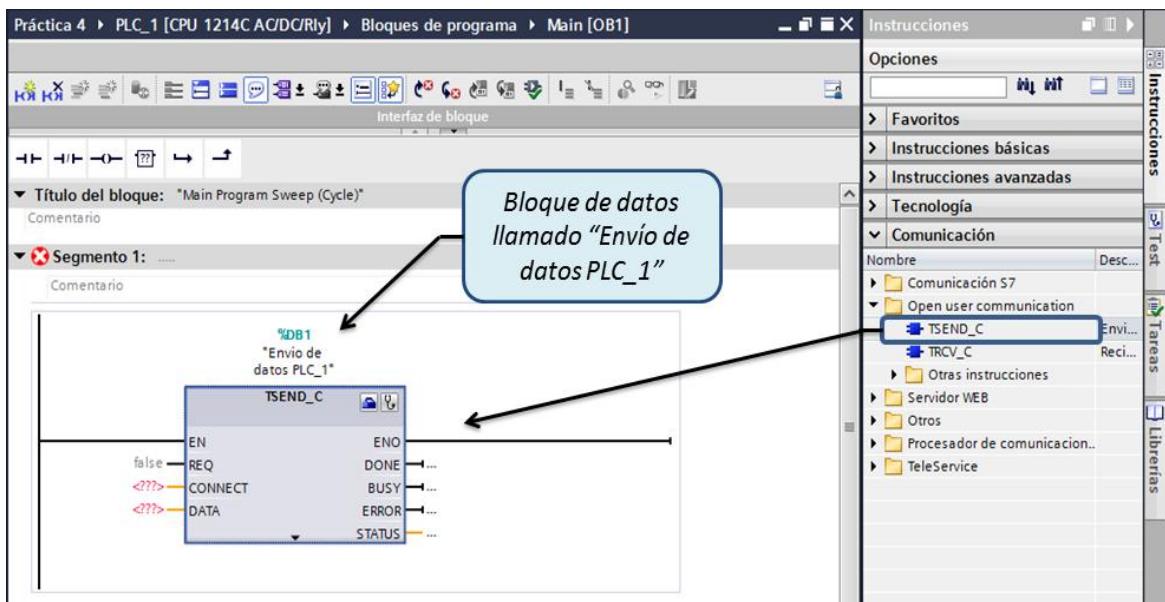
**Figura 16.** Configuración del byte de marcas de ciclo PLC\_1.

Repita el procedimiento anterior pero ahora con el PLC\_2 para su uso en el futuro del desarrollo de esta práctica. Configurado el byte de marcas de ciclo deberá programar la comunicación de manera unidireccional, es decir, en un solo sentido desde el PLC\_1 hacia el PLC\_2, para lograr esto se hará uso de las instrucciones TSEND\_C y TRCV\_C, ver **Figura 17**.



**Figura 17.** Comunicación unidireccional entre los PLCs.

Diríjase al *Main [OB1]* del **PLC\_1**, arrastre o coloque una instrucción TSEND\_C en el bloque de organización, esta se encuentra presente en *Instrucciones / Comunicación / Open user communication*, ver **Figura 18**.



**Figura 18.** Bloque TSEND\_C dispuesto en *Main [OB1]* del PLC\_1.

Acceda a las propiedades del bloque *Envío de datos PLC\_1*, específicamente en **Parámetros de la conexión** y configure los parámetros de la siguiente manera:

**Interlocutor:** PLC\_2.

**Datos de conexión (Local):** <nuevo>, lo que conlleva a agregarse por defecto al bloque de datos *PLC\_1\_Send\_DB* (presente en *PLC\_1 / Bloques de programa / Bloques de sistema / Recursos de programa*).

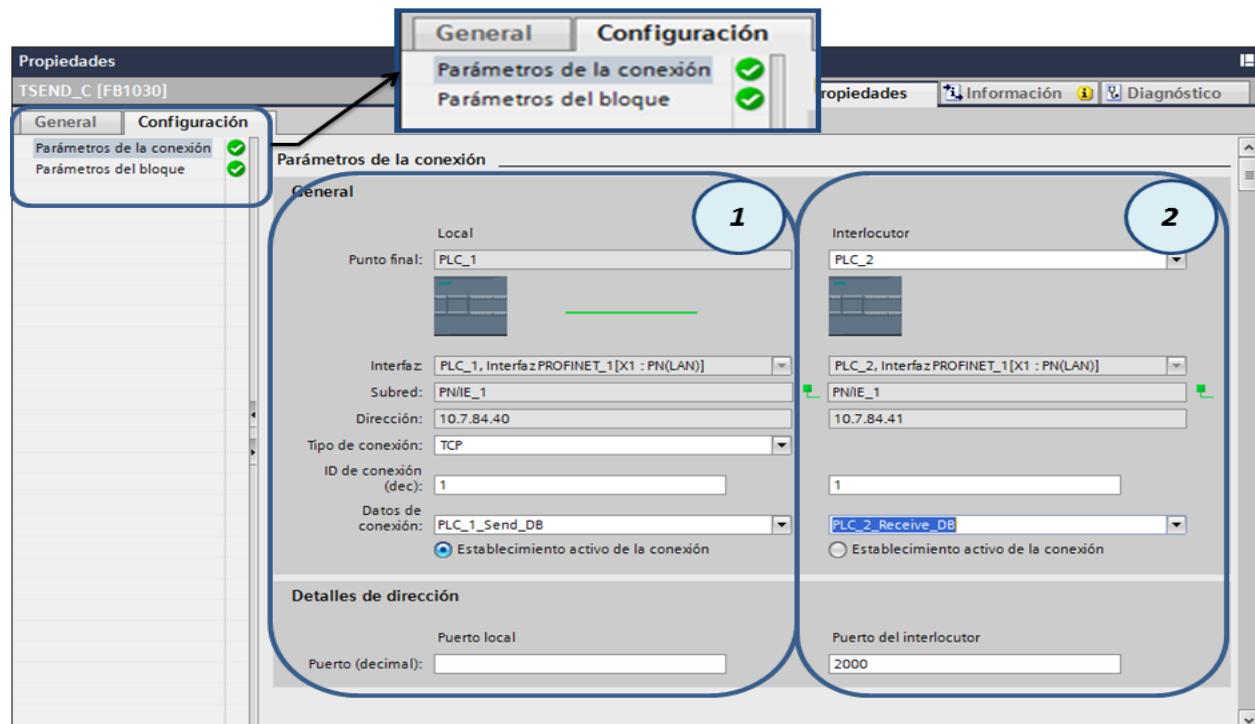
**Datos de conexión (Interlocutor):** <nuevo>, lo que conlleva a agregarse por defecto al bloque de datos *PLC\_2\_Receive\_DB* (presente en *PLC\_2 / Bloques de programa / Bloques de sistema / Recursos de programa*).

**Tipo de conexión:** TCP.

**ID de conexión (dec):** 1 (en la **Figura 17** puede evidenciarse esta ID).

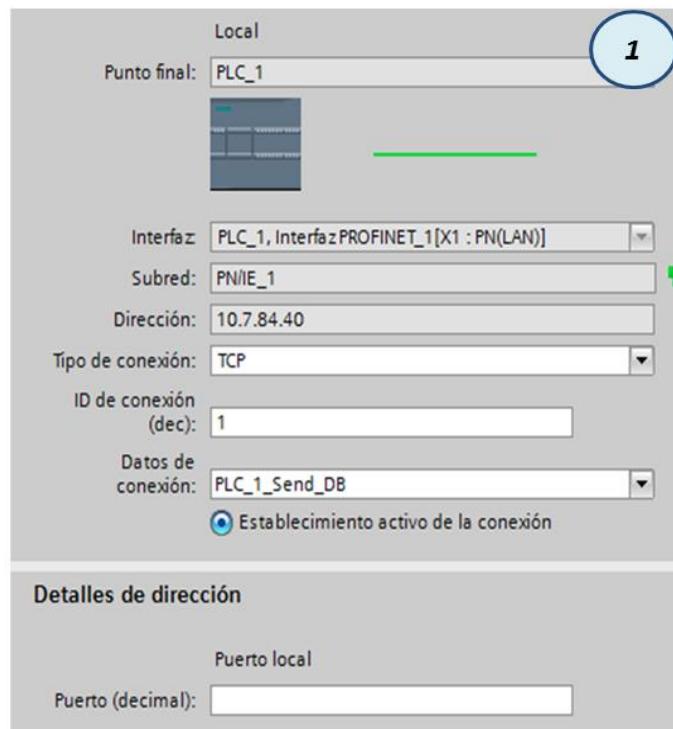
**Detalles de dirección:** 2000 (Tipo de conexión TCP entre 2000 - 5000).

En las **Figura 19** pueden verse las configuraciones realizadas anteriormente.

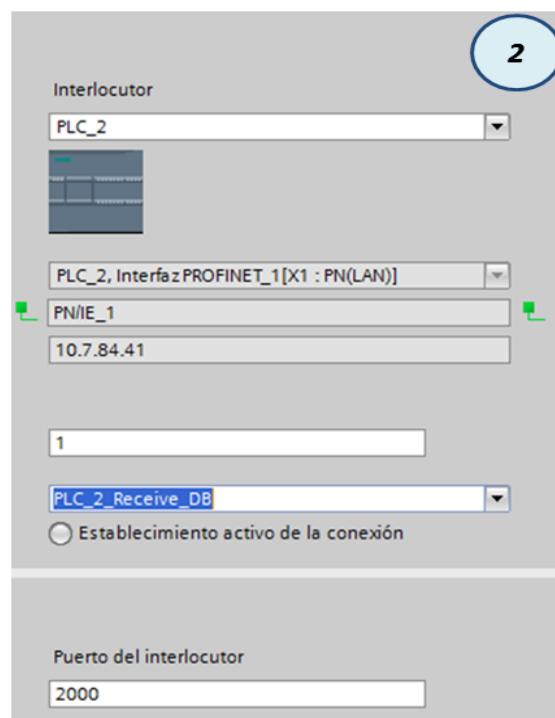


**Figura 19.** Configuración de *Parámetros de la conexión* del bloque TSEND\_C  
Envió de datos PLC\_1.

En las **Figuras 20 y 21** puede apreciarse mejor por partes la **Figura 19**.



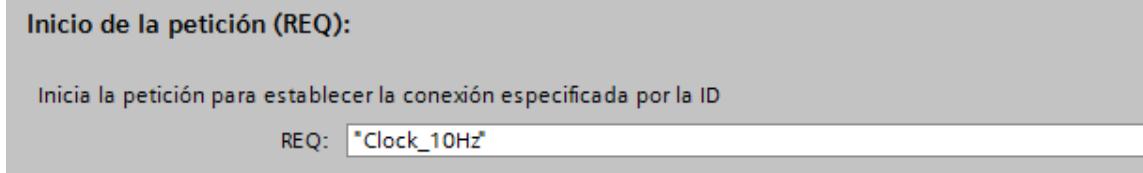
**Figura 20.** Configuración de los parámetros Locales (PLC\_1).



**Figura 21.** Configuración de los parámetros del Interlocutor (PLC\_2).

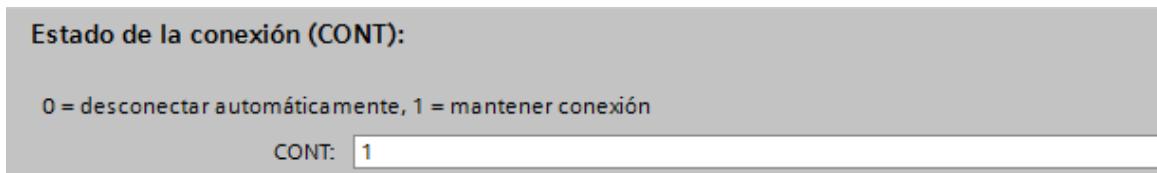
Configurados los *Parámetros de la conexión*, en **Parámetros del bloque** configure los parámetros de la siguiente manera:

**Inicio de la petición (REQ):** M100.0 (Clock\_10 Hz), ver **Figura 22.**



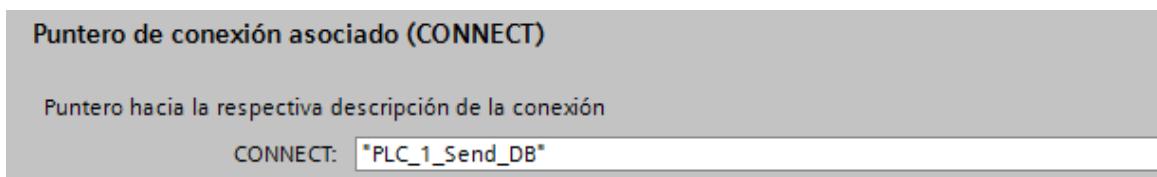
**Figura 22.** Configuración del parámetro *REQ*.

**Estado de la conexión (CONT):** 1, ver **Figura 23.**



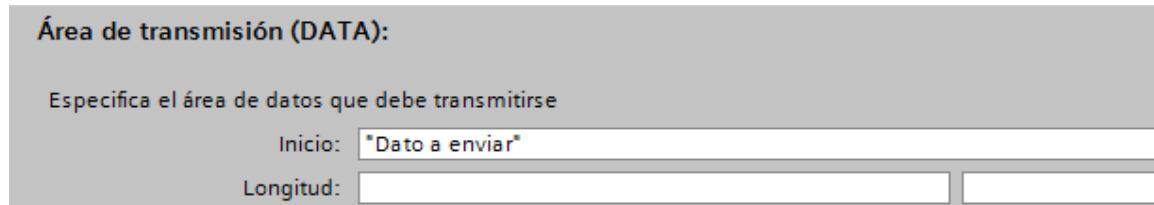
**Figura 23.** Configuración del parámetro *CONT*.

**Puntero de conexión asociado (CONNECT):** PLC\_1\_Send\_DB, ver **Figura 24.**



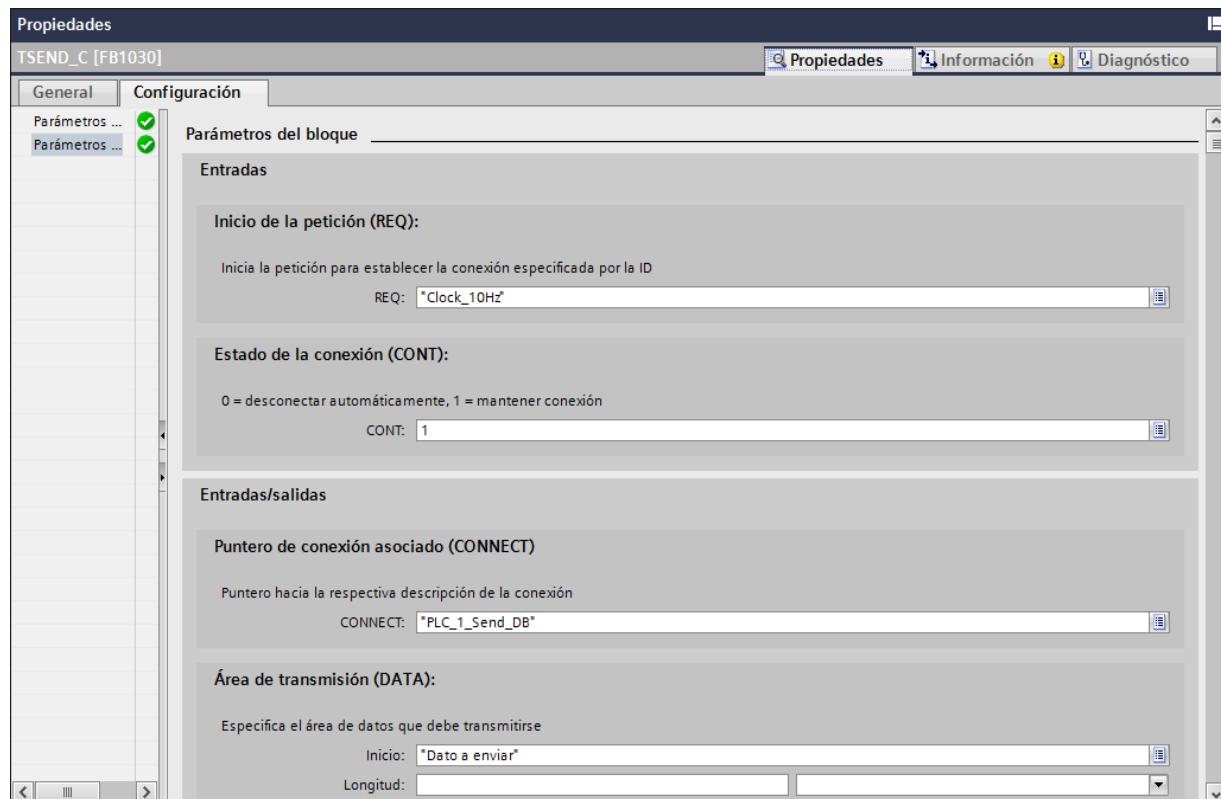
**Figura 24.** Configuración del parámetro *CONNECT*.

**Área de transmisión (DATA):** MW10 (Dato a enviar), ver **Figura 25.**



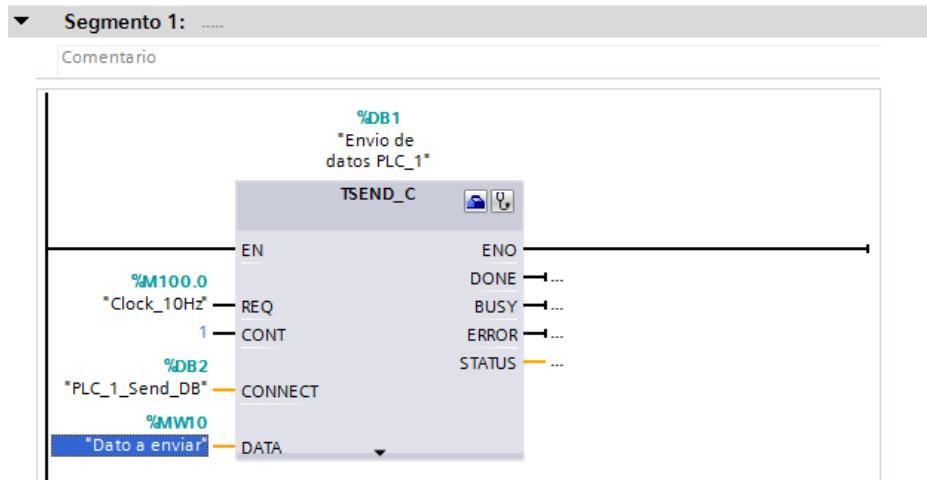
**Figura 25.** Configuración del parámetro DATA.

En la **Figura 26** se muestra la ventana de **Parámetros del bloque** ya configurada.



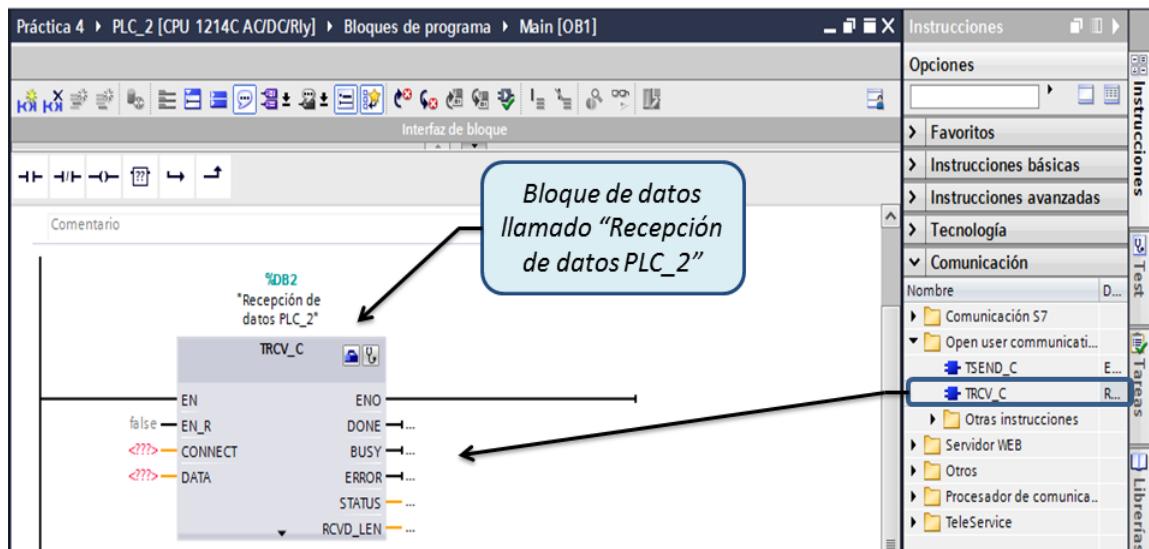
**Figura 26.** Configuración de *Parámetros del bloque TSEND\_C Envío de datos PLC\_1*.

En la **Figura 27** se puede apreciar el bloque *Envío de datos PLC\_1* configurado desde el *Main [OB1]*.



**Figura 27.** Bloque TSEND\_C Envío de datos PLC\_1, configurado.

Finalizada la configuración del bloque TSEND\_C Envío de datos PLC\_1, vaya al Main [OB1] del PLC\_2 y disponga sobre un segmento de código al bloque TRCV\_C, ver Figura 28.



**Figura 28.** Bloque TRCV\_C dispuesto en Main [OB1] del PLC\_2.

Acceda a las propiedades del bloque Recepción de datos PLC\_2 específicamente en **Parámetros de la conexión** y configure los parámetros de la siguiente manera:

**Interlocutor:** PLC\_1.

**Datos de conexión (Local):** PLC\_2\_Receive\_DB (presente en PLC\_2 / Bloques de programa / Bloques de sistema / Recursos de programa).

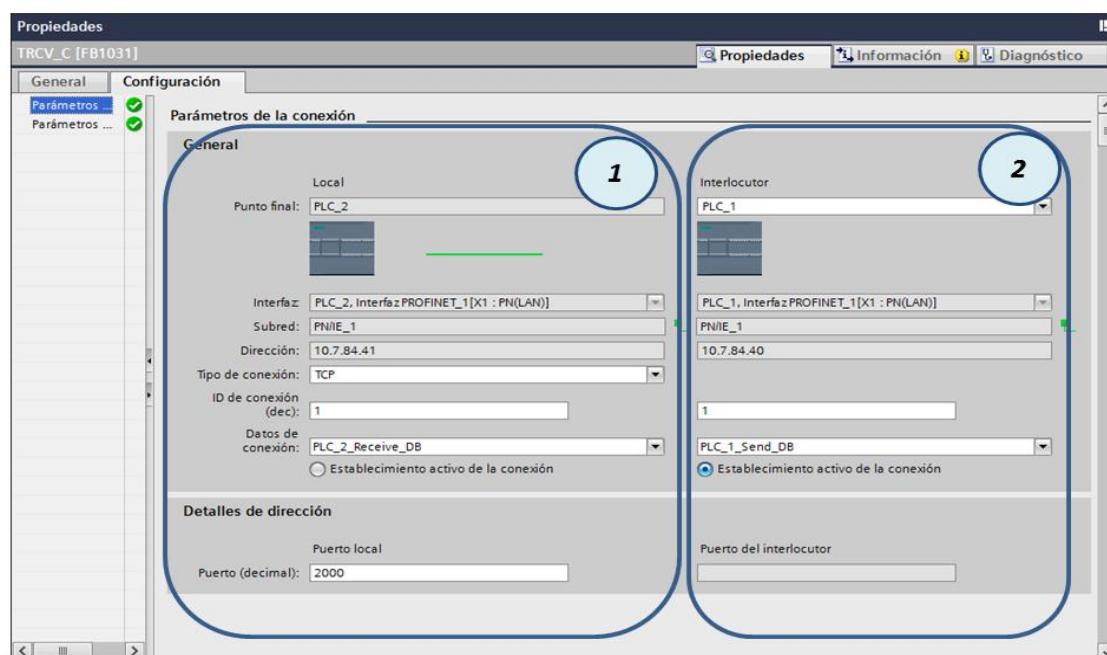
**Datos de conexión (Interlocutor):** PLC\_1\_Send\_DB (presente en PLC\_1 / Bloques de programa / Bloques de sistema / Recursos de programa).

**Tipo de conexión:** TCP.

**ID de conexión (dec):** 1 (en la **Figura 17** puede evidenciarse esta ID).

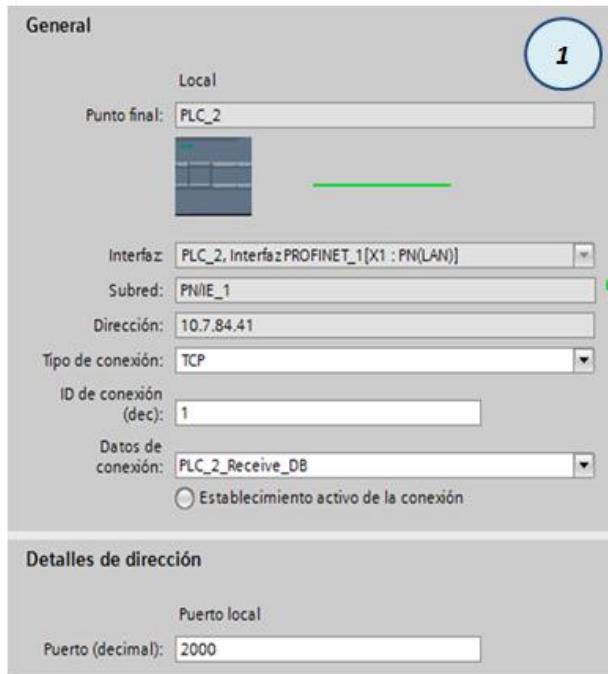
**Detalles de dirección:** 2000 (Tipo de conexión TCP entre 2000 - 5000).

En las **Figura 29** pueden verse las configuraciones realizadas anteriormente.

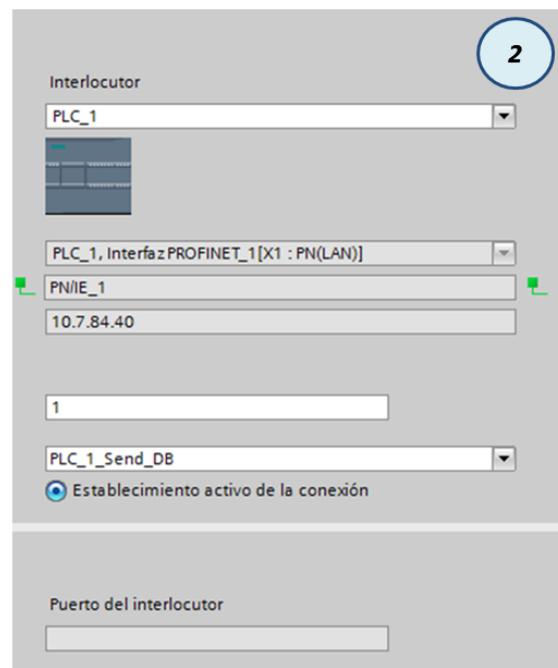


**Figura 29.** Configuración de *Parámetros de la conexión* del bloque TRCV\_C *Recepción de datos PLC\_2*.

En las **Figuras 30 y 31** puede apreciarse mejor por partes la **Figura 29**.



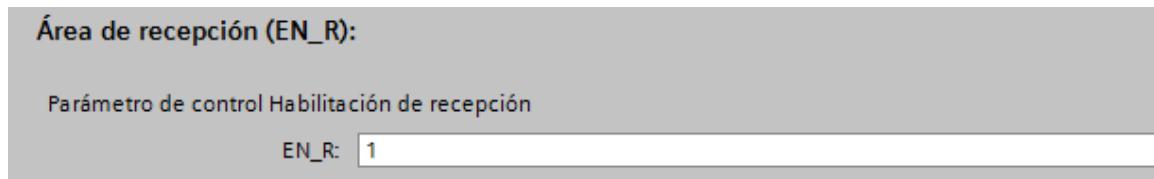
**Figura 30.** Configuración de los parámetros Locales (PLC\_2).



**Figura 31.** Configuración de los parámetros del Interlocutor (PLC\_1).

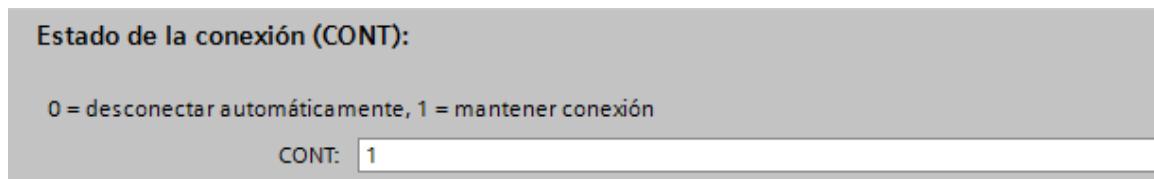
Configurados los **Parámetros de la conexión**, en **Parámetros del bloque** configure los parámetros de la siguiente manera:

**Área de recepción (EN\_R):** 1, ver **Figura 32.**



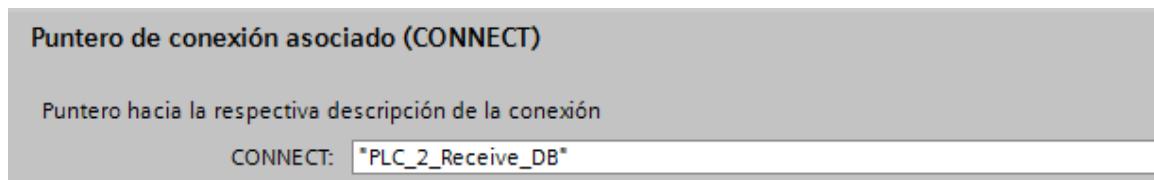
**Figura 32.** Configuración del parámetro *EN\_R*.

**Estado de la conexión (CONT):** 1, ver **Figura 33.**



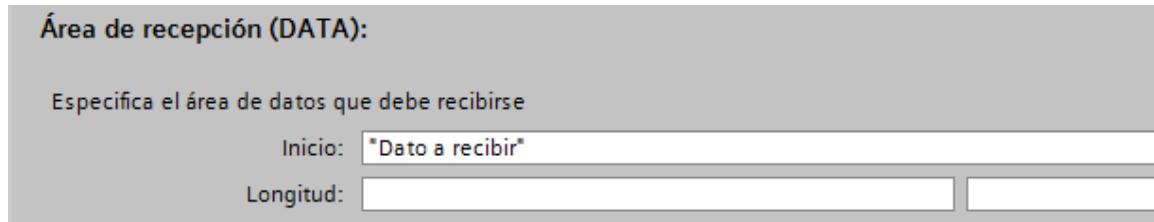
**Figura 33.** Configuración del parámetro *CONT*.

**Puntero de conexión asociado (CONNECT):** *PLC\_2\_Receive\_DB*, ver **Figura 34.**



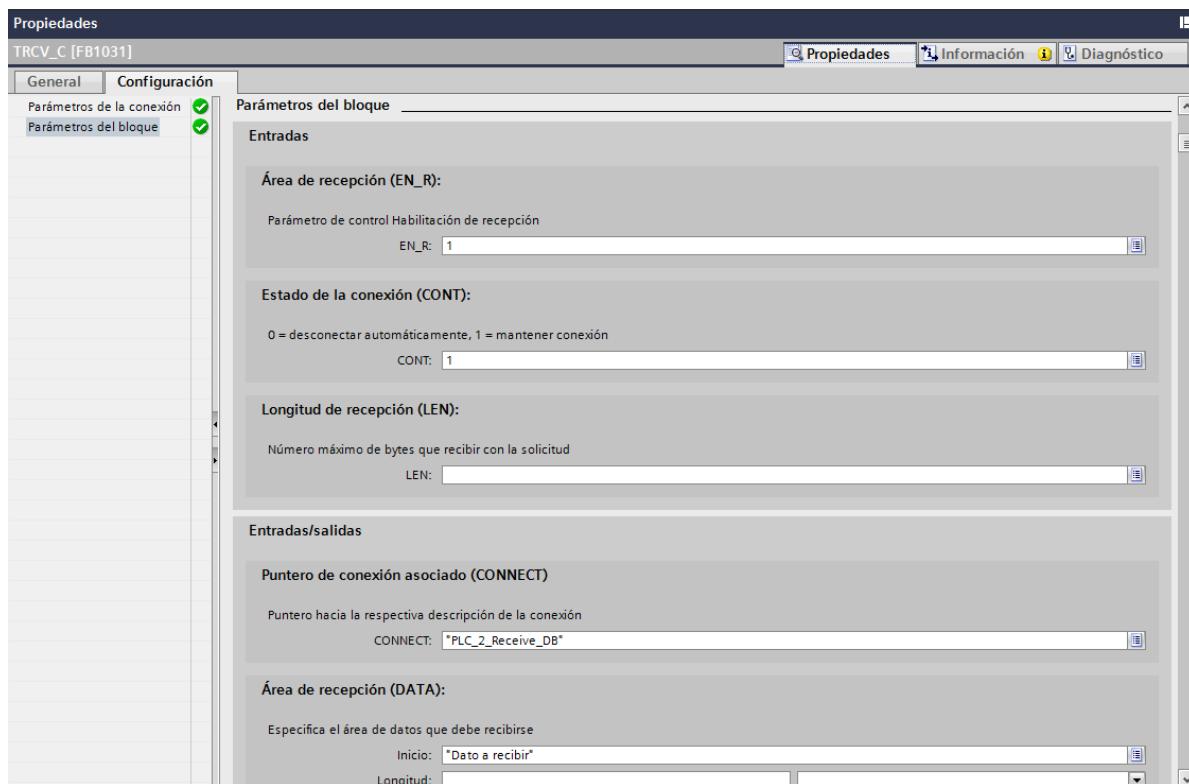
**Figura 34.** Configuración del parámetro *CONNECT*.

**Área de transmisión (DATA):** MW10 (Dato a recibir), ver **Figura 35.**



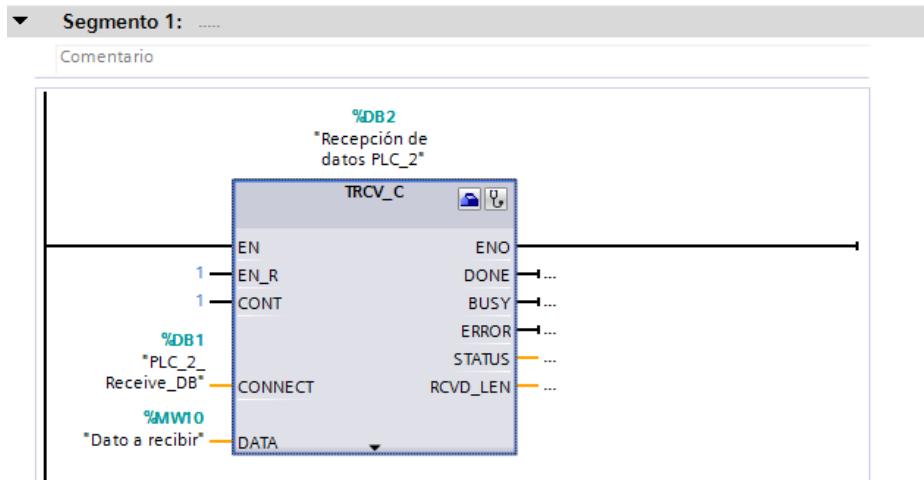
**Figura 35.** Configuración del parámetro DATA.

En la **Figura 36** se muestra la ventana de **Parámetros del bloque** ya configurada.



**Figura 36.** Configuración de Parámetros del bloque TRCV\_C Recepción de datos PLC\_2.

En la **Figura 37** se puede apreciar el bloque *Recepción de datos PLC\_2* configurado desde el *Main [OB1]* del PLC\_2.

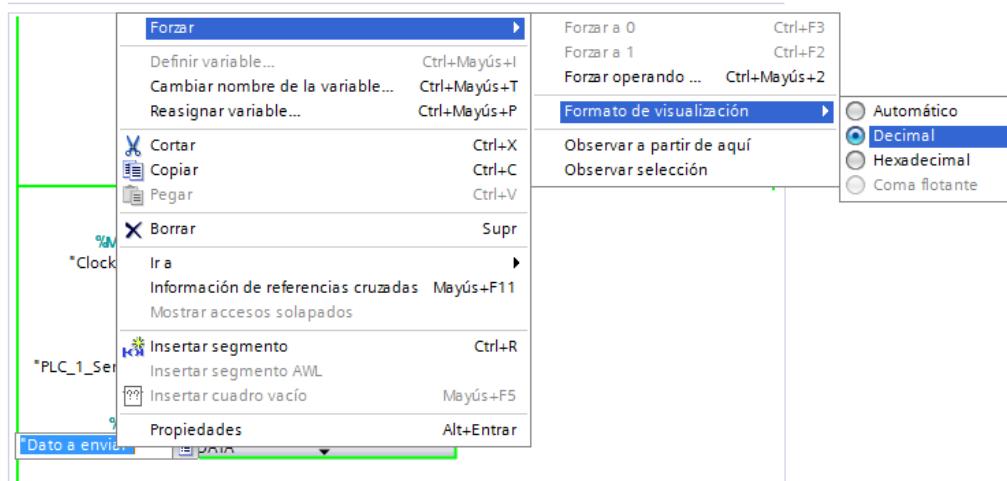


**Figura 37.** Bloque TRCV\_C *Recepción de datos PLC\_2*, configurado.

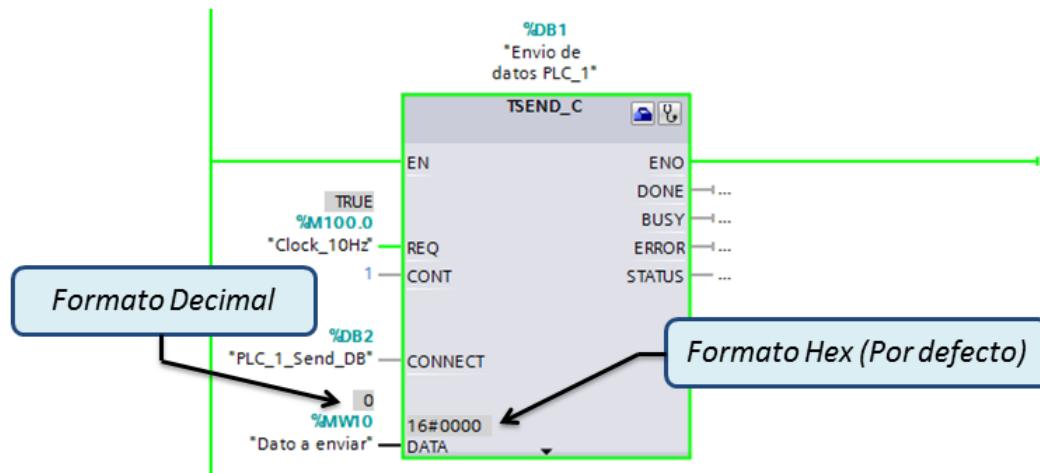
Finalizada la configuración de los bloques para cada PLC, dispóngase a cargar ambos programas a cada PLC respectivamente.

Los respectivos programas creados y cargados tienen como objetivo establecer comunicación entre ambos PLCs, la transferencia de datos es de forma unidireccional desde el PLC\_1 hacia el PLC\_2 y el dato enviado / recibido (en este caso) es de la longitud de una palabra (*Word*, 16 bits).

Coloque ambos PLCs en modo *Run* y desde el parámetro DATA (MW10 *Dato a enviar*) del bloque TSEND\_C *Envío de datos PLC\_1* seleccione la opción *Forzar / Formato de visualización / Decimal*, por defecto el bloque muestra el parámetro data en formato *Hex* (Hexadecimal), de este modo es posible evidenciar el dato a ser transmitido tanto en formato Decimal como Hexadecimal, ver **Figura 38** (Selección del formato) y **Figura 39** (Ambos formatos). De manera análoga repita el procedimiento anterior con el bloque TRCV\_C *Recepción de datos PLC\_2*.

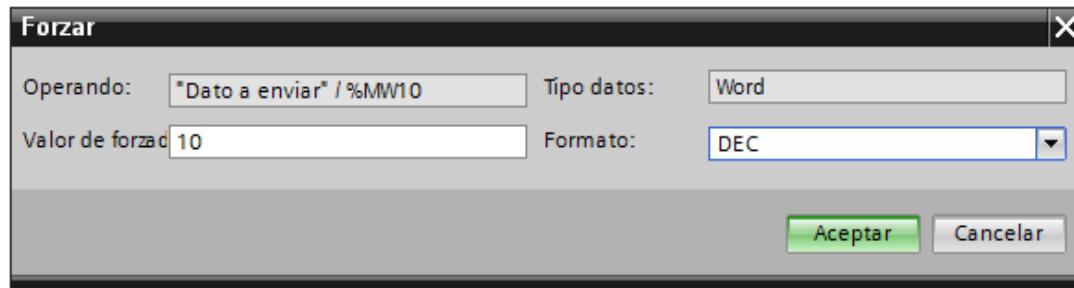


**Figura 38.** Cambio de formato de visualización del parámetro *DATA*.



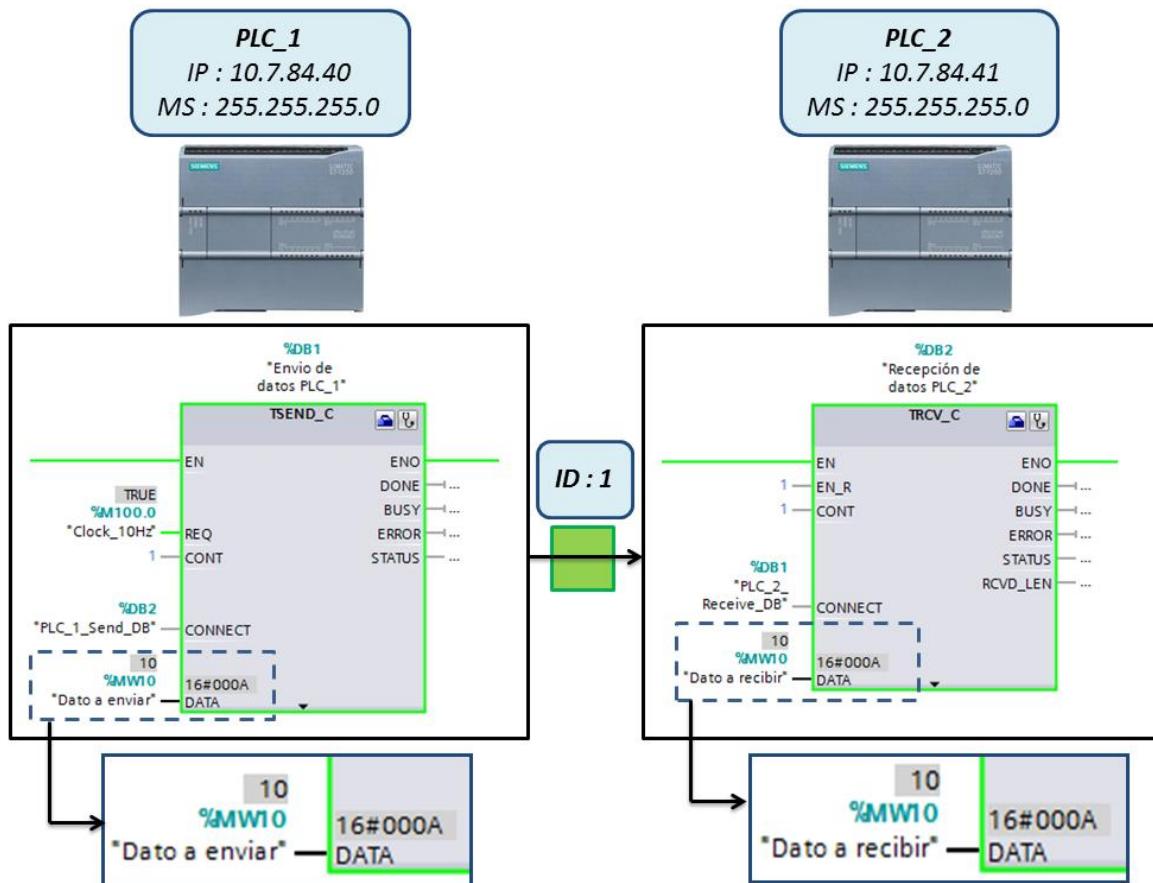
**Figura 39.** Visualización del dato en formato Decimal y Hex.

Seleccione la opción *Forzar operando* e introduzca el valor de diez (10), véase la **Figura 40**.



**Figura 40.** Forzado de operando (MW10).

En la **Figura 41** se aprecia la comunicación establecida y verificada por la transmisión exitosa del valor de diez (10) desde el PLC\_1 hacia el PLC\_2.

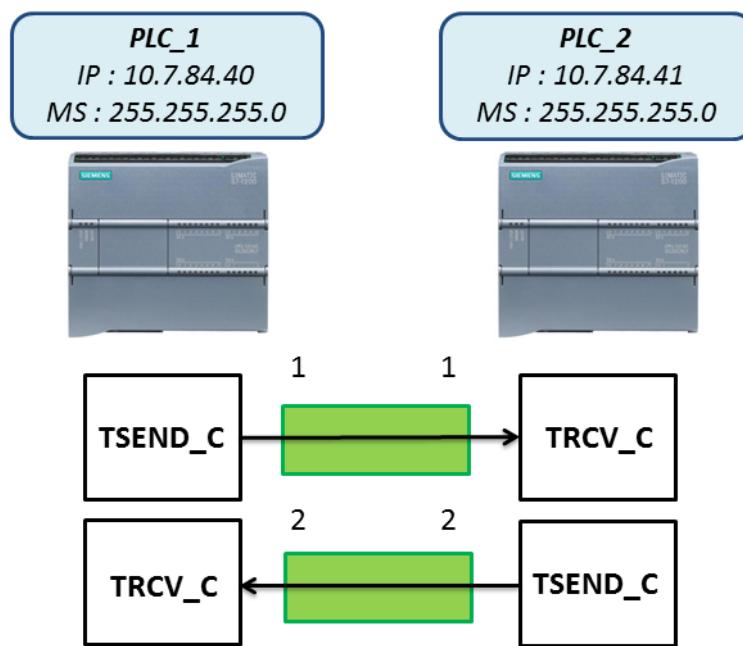


**Figura 41.** Comunicación unidireccional establecida y verificada.

**Nota:** Alternativamente TIA PORTAL incluye la tabla de observación, la cual puede ser utilizada para cada PLC en la observación del dato suscrito a MW10.

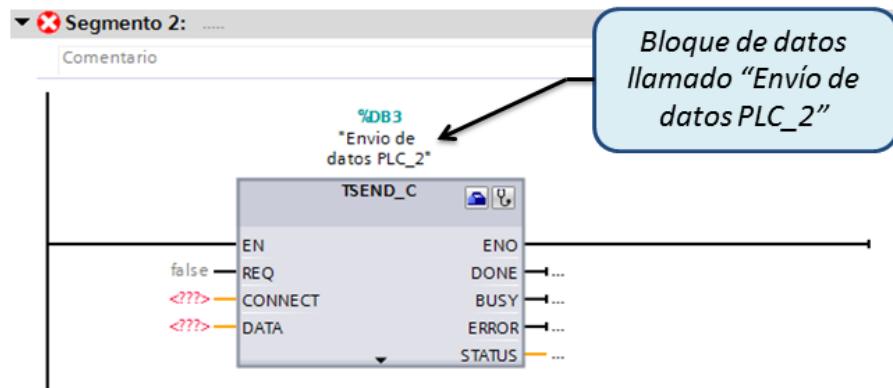
#### 4.3. Comunicación bidireccional entre los PLCs

Para llevar a cabo este tipo de comunicación se complementará el programa anterior (comunicación unidireccional entre los PLCs); primero se necesitará otra ID de conexión que permita establecer una vía alterna de encaminamiento de los datos entre los PLCs (específicamente del PLC\_2 al PLC\_1), véase la **Figura 42**.



**Figura 42.** Comunicación bidireccional entre los PLCs.

Diríjase al *Main [OB1]* del **PLC\_2** y arrastre o coloque una instrucción TSEND\_C en el bloque de organización, ver **Figura 43**.



**Figura 43.** Bloque TSEND\_C dispuesto en *Main [OB1]* del PLC\_2.

Configure los **Parámetros de la conexión** del bloque TSEND\_C *Envío de datos PLC\_2* de la siguiente forma:

**Interlocutor:** PLC\_1.

**Datos de conexión (Local):** <nuevo>, lo que conlleva a agregarse por defecto al bloque de datos *PLC\_2\_Send\_DB* (presente en *PLC\_2 / Bloques de programa / Bloques de sistema / Recursos de programa*).

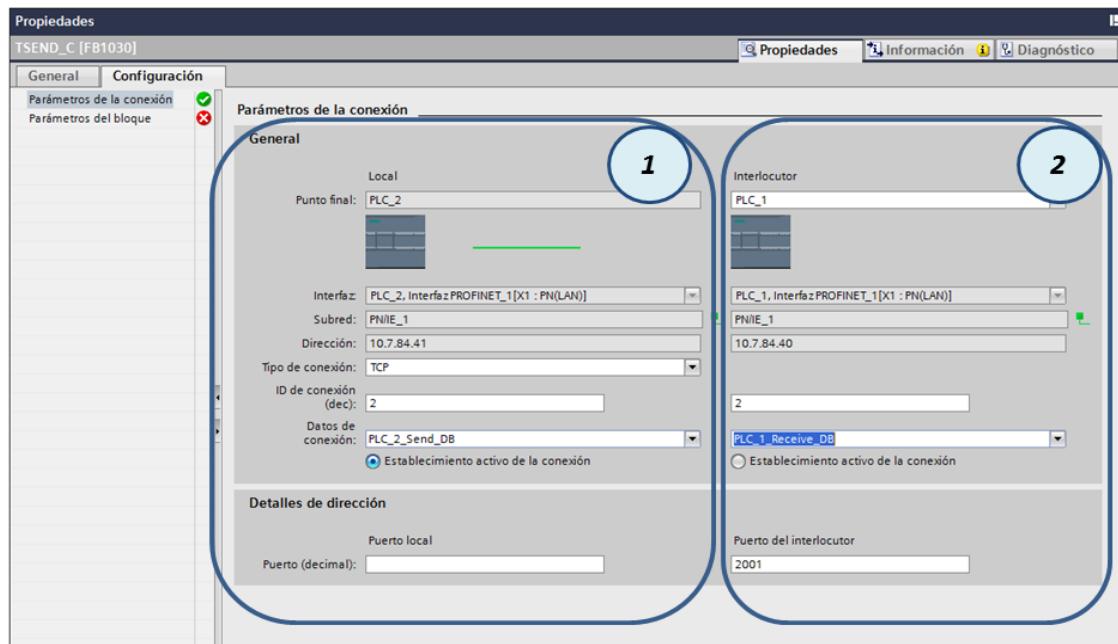
**Datos de conexión (Interlocutor):** <nuevo>, lo que conlleva a agregarse por defecto al bloque de datos *PLC\_1\_Receive\_DB* (presente en *PLC\_1 / Bloques de programa / Bloques de sistema / Recursos de programa*).

**Tipo de conexión:** TCP.

**ID de conexión (dec):** 2 (por defecto, en la **Figura 42** puede evidenciarse esta ID).

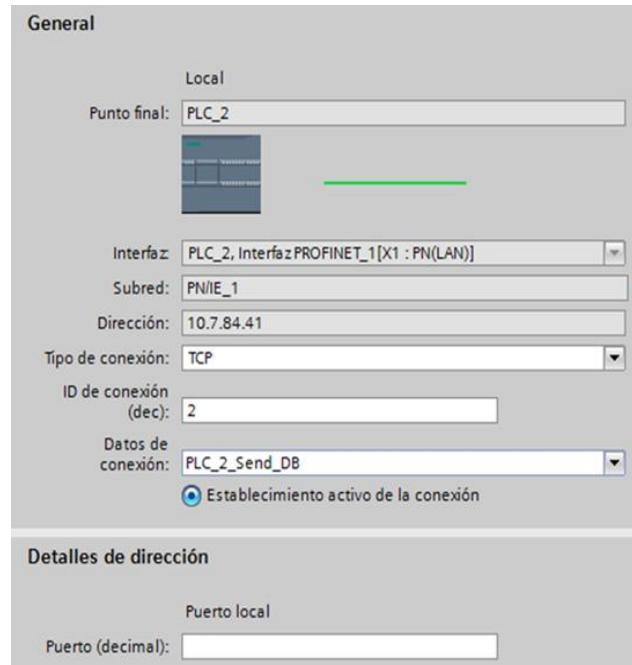
**Detalles de dirección:** 2001 (Tipo de conexión TCP entre 2000 - 5000).

En las **Figura 44** pueden verse las configuraciones realizadas anteriormente.

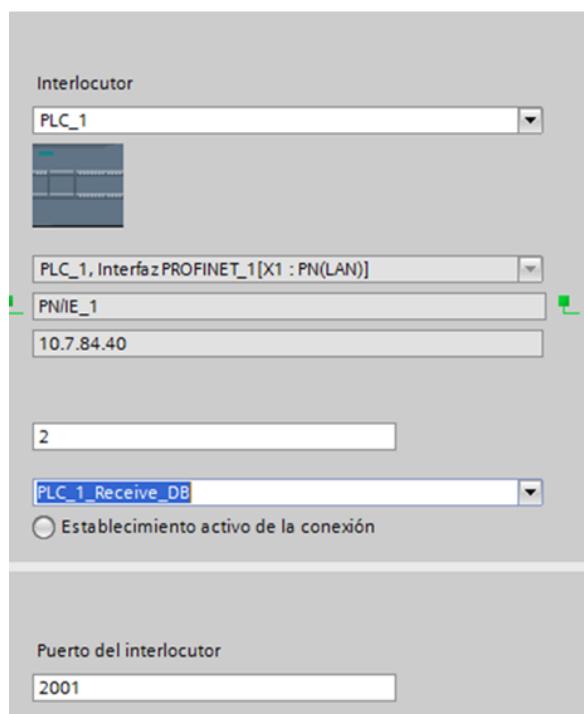


**Figura 44.** Configuración de *Parámetros de la conexión* del bloque TSEND\_C Envió de datos PLC\_2.

En las **Figuras 45 y 46** puede apreciarse mejor por partes la **Figura 44**.



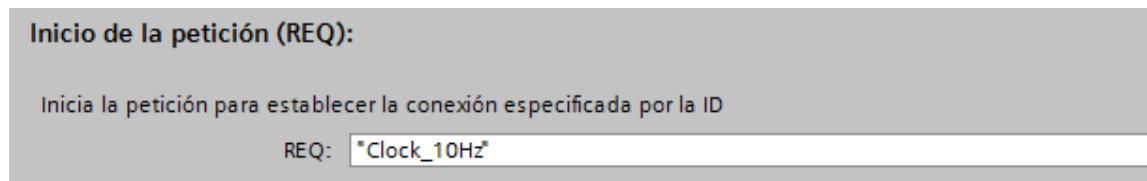
**Figura 45.** Configuración de los parámetros Locales (PLC\_2).



**Figura 46.** Configuración de los parámetros del Interlocutor (PLC\_1).

Configurados los **Parámetros de conexión**, en **Parámetros del bloque** configurre los parámetros de la siguiente manera:

**Inicio de la petición (REQ):** M100.0 (Clock\_10 Hz), ver **Figura 47.**



**Figura 47.** Configuración del parámetro *REQ*.

**Estado de la conexión (CONT):** 1, ver **Figura 48.**

**Estado de la conexión (CONT):**

0 = desconectar automáticamente, 1 = mantener conexión

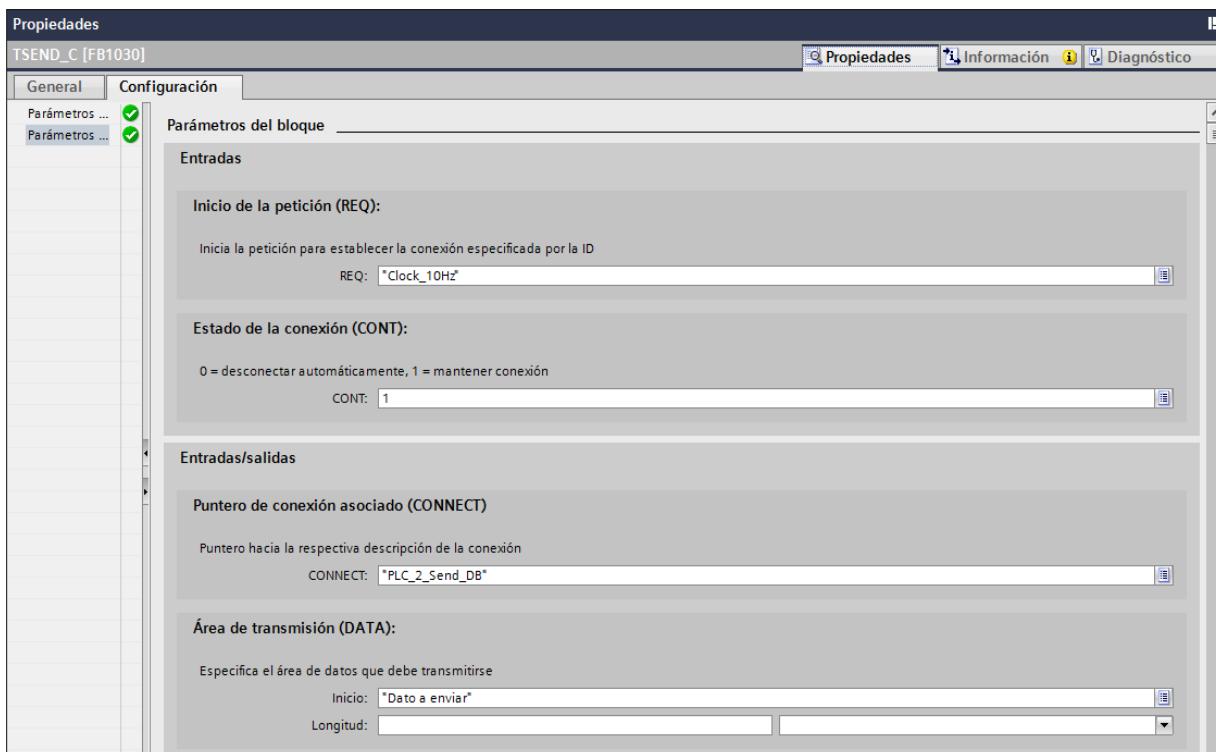
CONT: **Figura 48.** Configuración del parámetro CONT.**Puntero de conexión asociado (CONNECT): PLC\_2\_Send\_DB, ver Figura 49.****Puntero de conexión asociado (CONNECT)**

Puntero hacia la respectiva descripción de la conexión

CONNECT: **Figura 49.** Configuración del parámetro CONNECT.**Área de transmisión (DATA): MW20 (Dato a enviar), ver Figura 50.****Área de transmisión (DATA):**

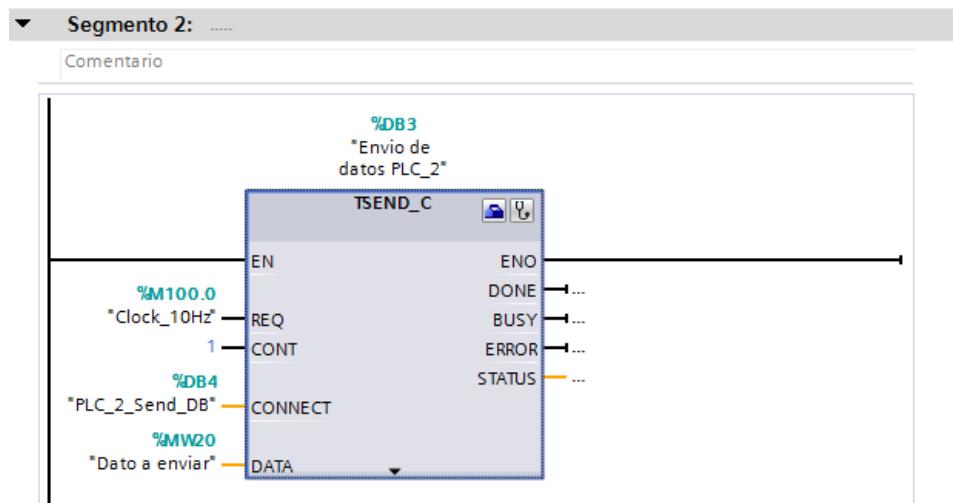
Especifica el área de datos que debe transmitirse

Inicio: Longitud: **Figura 50.** Configuración del parámetro DATA.En la **Figura 51**, se muestra la ventana de **Parámetros del bloque** ya configurada.



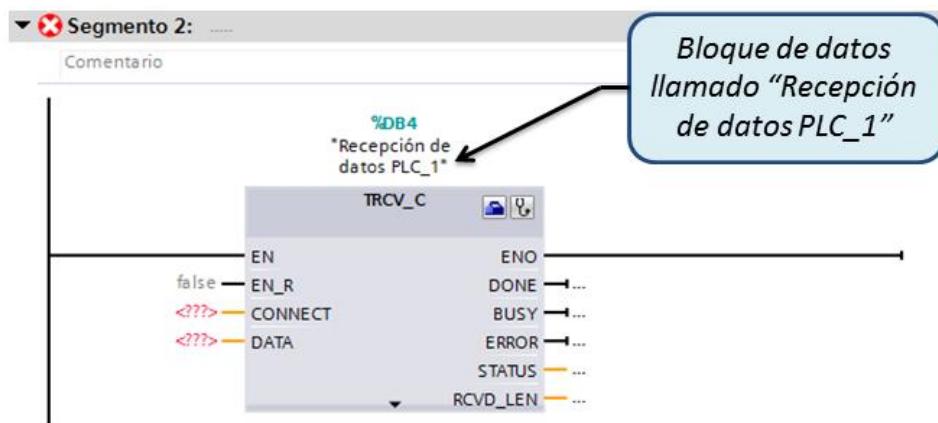
**Figura 51.** Configuración de *Parámetros del bloque TSEND\_C Envío de datos PLC\_2*.

En la **Figura 52** se puede apreciar el bloque *Envío de datos PLC\_2* configurado desde el *Main [OB1]*.



**Figura 52.** Bloque TSEND\_C *Envío de datos PLC\_2*, configurado.

Finalizada la configuración del bloque TSEND\_C *Envío de datos PLC\_2*, vaya al *Main [OB1]* del **PLC\_1** y disponga sobre un segmento de código al bloque TRCV\_C, ver **Figura 53.**



**Figura 53.** Bloque TRCV\_C dispuesto en *Main [OB1]* del PLC\_1.

Acceda a las propiedades del bloque *Recepción de datos PLC\_1* específicamente en **Parámetros de la conexión** y configure los parámetros de la siguiente manera:

**Interlocutor:** PLC\_2.

**Datos de conexión (Local):** PLC\_1\_Receive\_DB (presente en *PLC\_1 / Bloques de programa / Bloques de sistema / Recursos de programa*).

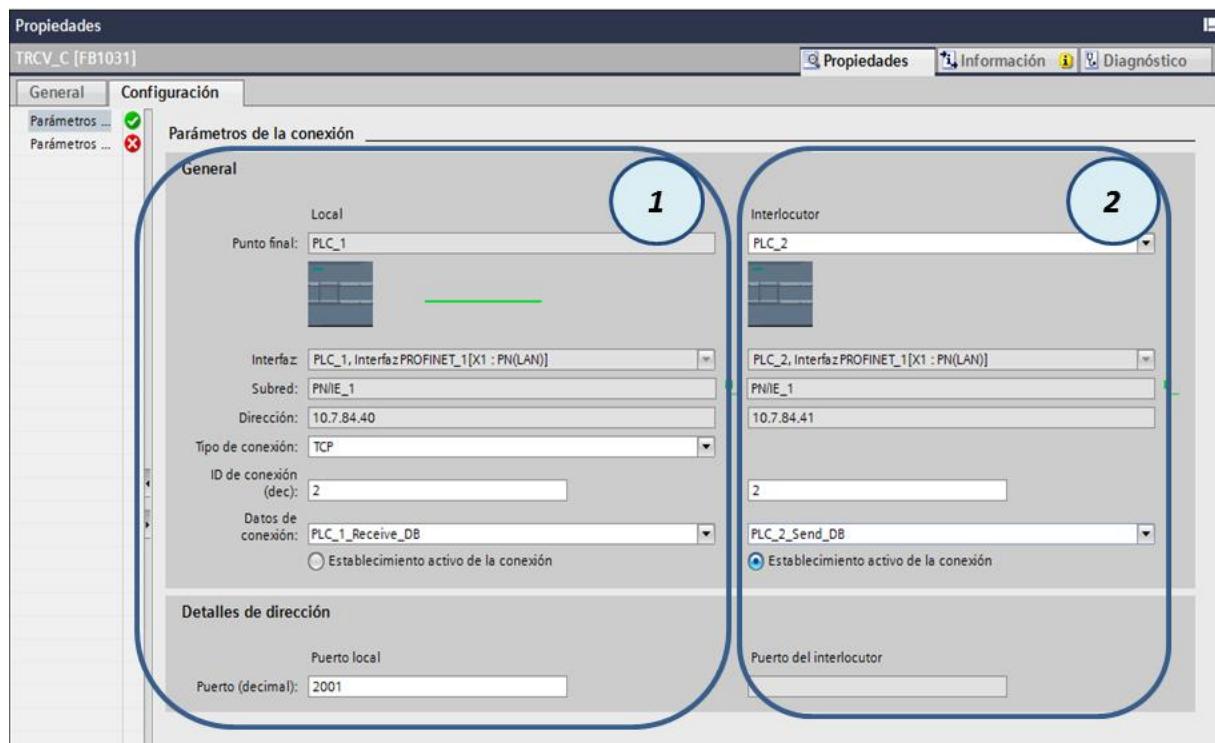
**Datos de conexión (Interlocutor):** PLC\_2\_Send\_DB (presente en *PLC\_2 / Bloques de programa / Bloques de sistema / Recursos de programa*).

**Tipo de conexión:** TCP.

**ID de conexión (dec):** 2 (por defecto, en la **Figura 42** puede evidenciarse esta ID).

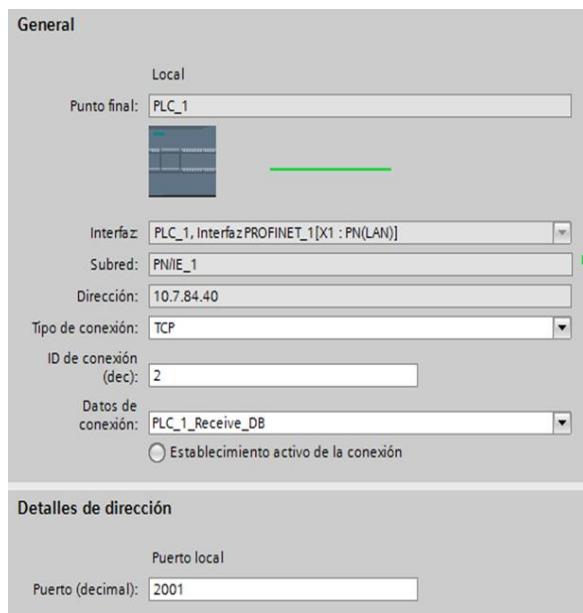
**Detalles de dirección:** 2001 (Tipo de conexión TCP entre 2000 - 5000).

En las **Figura 54** pueden verse las configuraciones realizadas anteriormente.

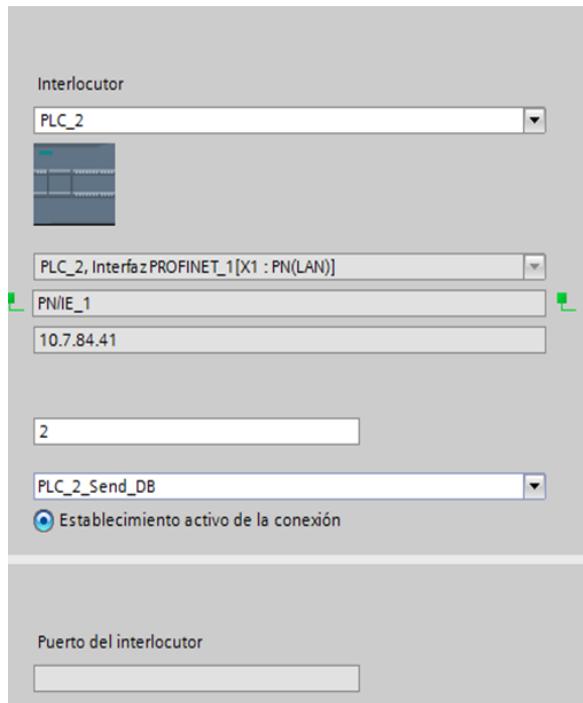


**Figura 54.** Configuración de *Parámetros de la conexión* del bloque TRCV\_C  
Recepción de datos PLC\_1.

En las **Figuras 55 y 56** puede apreciarse por partes la **Figura 54**.



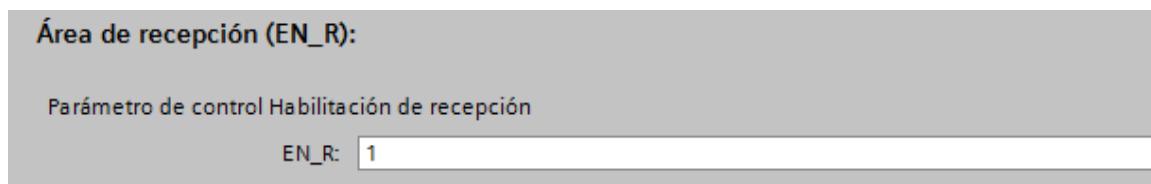
**Figura 55.** Configuración de los parámetros Locales (PLC\_1).



**Figura 56.** Configuración de los parámetros del Interlocutor (PLC\_2).

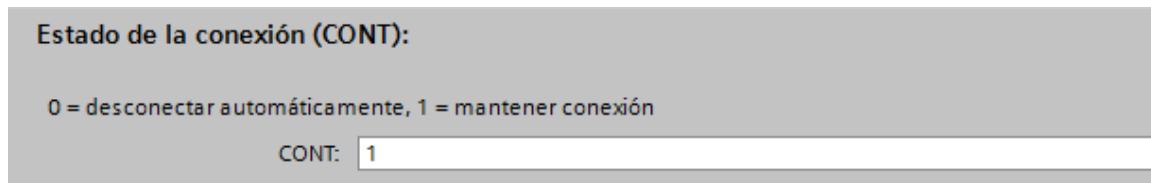
Configurados los **Parámetros de conexión**, en **Parámetros del bloque** ajuste los parámetros de la siguiente manera:

**Área de recepción (EN\_R):** 1, ver **Figura 57.**



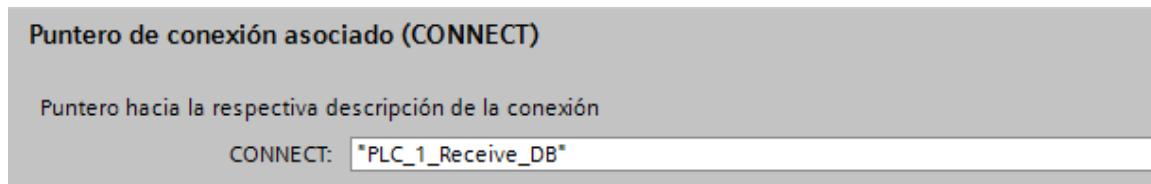
**Figura 57.** Configuración del parámetro *EN\_R*.

**Estado de la conexión (CONT):** 1, ver **Figura 58.**



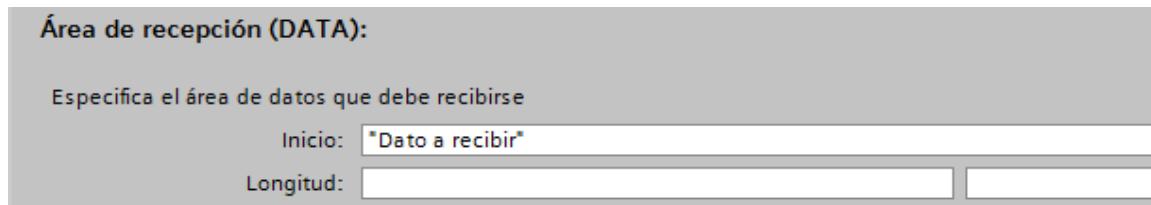
**Figura 58.** Configuración del parámetro *CONT*.

**Puntero de conexión asociado (CONNECT):** *PLC\_1\_Receive\_DB*, ver **Figura 59.**



**Figura 59.** Configuración del parámetro *CONNECT*.

**Área de transmisión (DATA):** MW20 (Dato a recibir), ver **Figura 60.**



**Figura 60.** Configuración del parámetro *DATA*.

En la **Figura 61** se muestra la ventana de **Parámetros del bloque** ya configurada.



**Figura 61.** Configuración de *Parámetros del bloque* TRCV\_C *Recepción de datos PLC\_1*.

Finalizada la configuración de los nuevos bloques para cada PLC dispóngase a cargar ambos programas a cada PLC respectivamente.

Los respectivos programas creados y cargados tienen como objetivo establecer comunicación entre ambos PLCs, la transferencia de datos será bidireccional desde el PLC\_1 hacia el PLC\_2 y desde el PLC\_2 hacia el PLC\_1, los datos enviados / recibidos (en este caso) son de la longitud de una palabra (*Word*, 16 bits).

Ahora, coloque ambos PLCs en modo *Run* y utilice la función *Forzar operando* para transmitir un diez (10) desde el PLC\_1 hacia el PLC2 y un quince (15) desde el PLC\_2 hacia el PLC\_1, ver **Figura 62**.

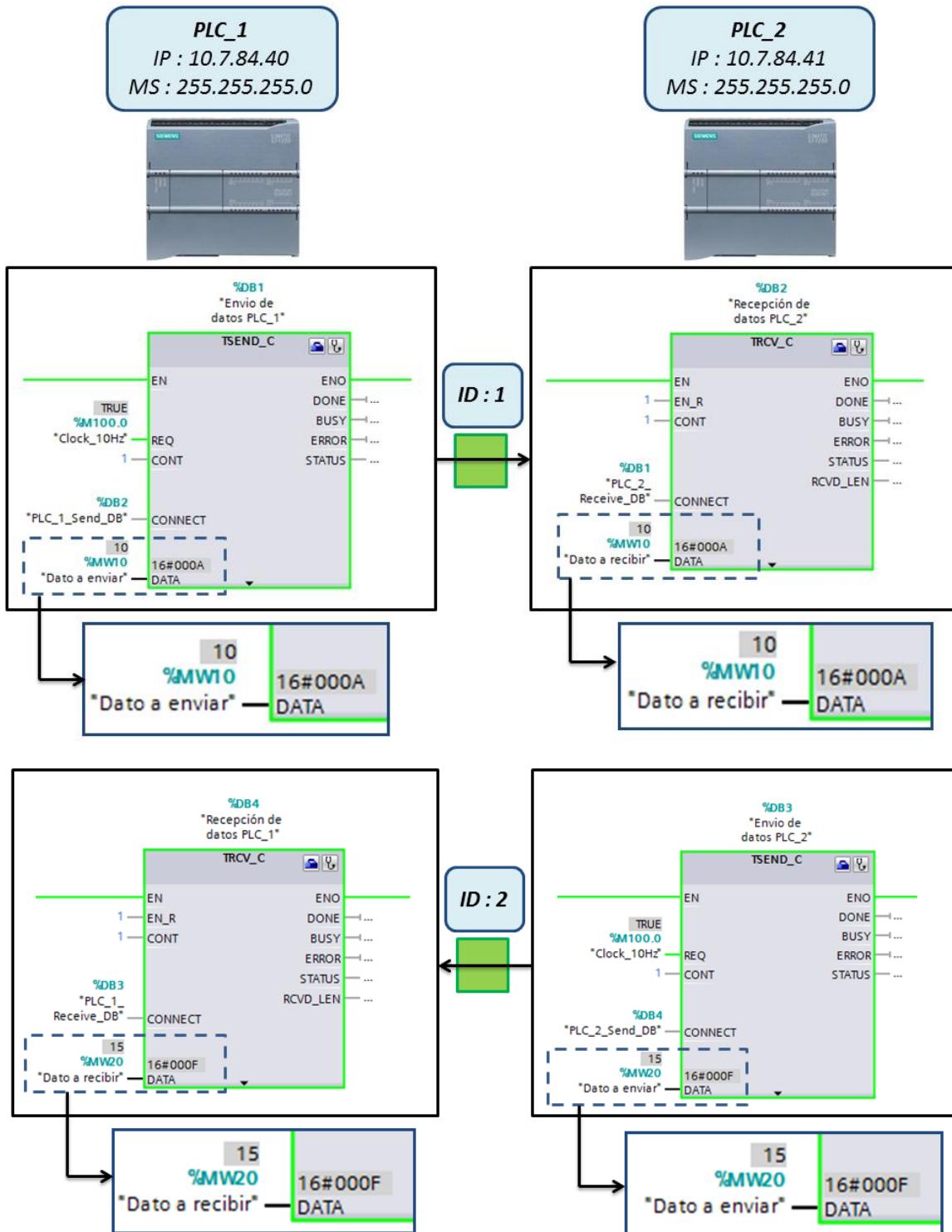
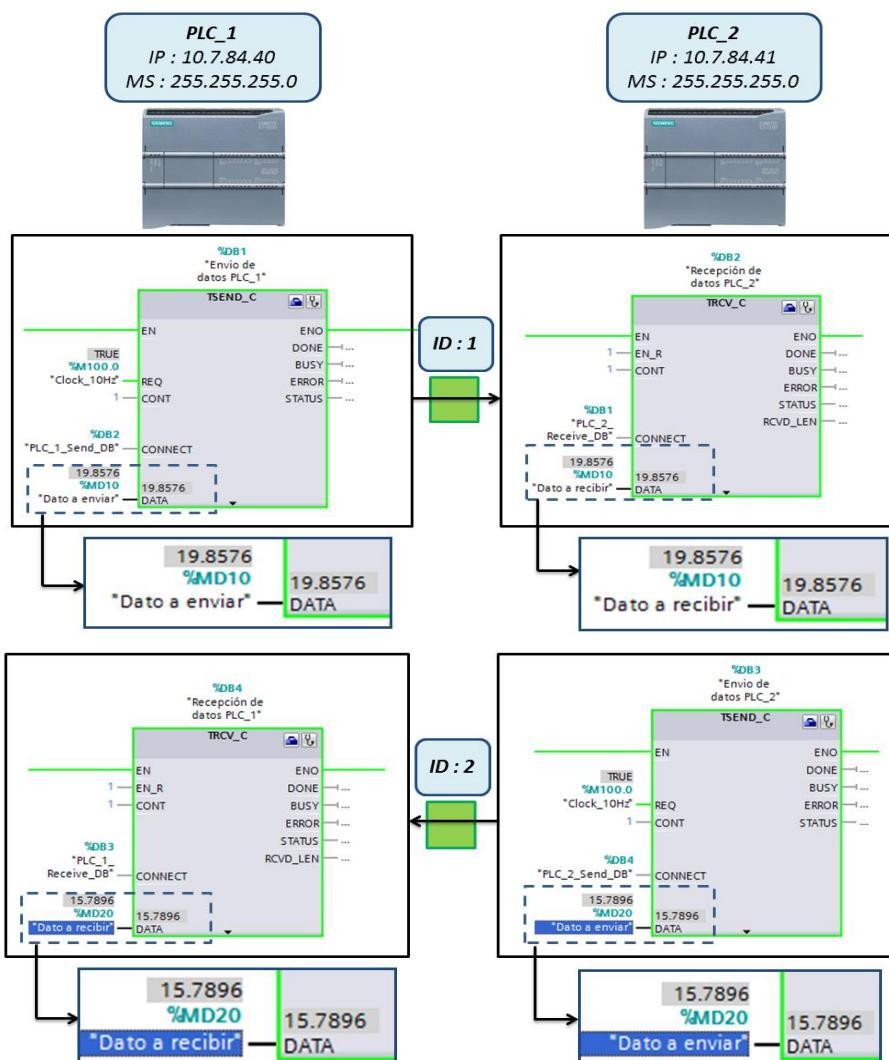


Figura 62. Comunicación bidireccional establecida y verificada entre los PLCs (1).

Realizada la transferencia de datos del tipo *Word* (MW10 y MW20), ajuste los parámetros *Dato a enviar* y *Dato a recibir* (en cada bloque de comunicación) desde la opción *Reasignar variable* (clic derecho sobre la variable en el parámetro) para un nuevo tamaño de memoria de doble palabra *Dword* (MD) y tipo de dato *Real*.

Luego, coloque ambos PLCs en modo *Run* y utilice la función *Forzar operando* para transmitir 19.8576 desde el PLC\_1 hacia el PLC2 y un 15.7896 desde el PLC\_2 hacia el PLC\_1, ver **Figura 63**.

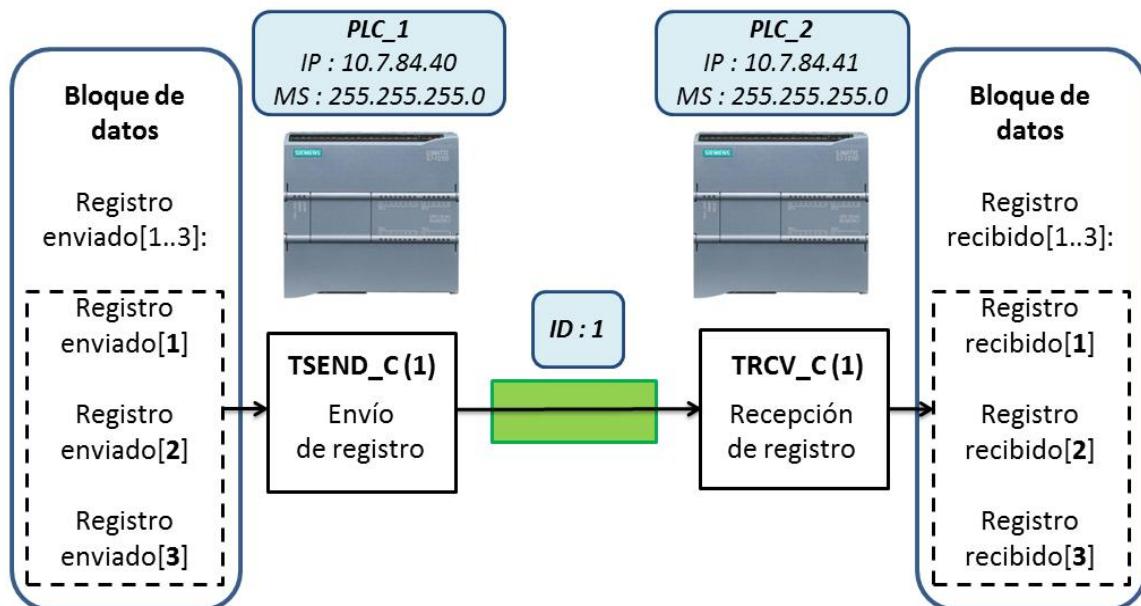


**Figura 63.** Comunicación bidireccional establecida y verificada entre los PLCs (2).

#### 4.4. Transmisión de paquetes controlada y supervisada mediante una HMI.

En función del material desarrollado en las prácticas anteriores más la implementación de bloques de datos, se utilizarán las instrucciones TSEND\_C y TRCV\_C para la transmisión de un pequeño arreglo de datos, adicionalmente se le incorporará una HMI para complementar y visualizar externamente al TIA PORTAL la transmisión de datos.

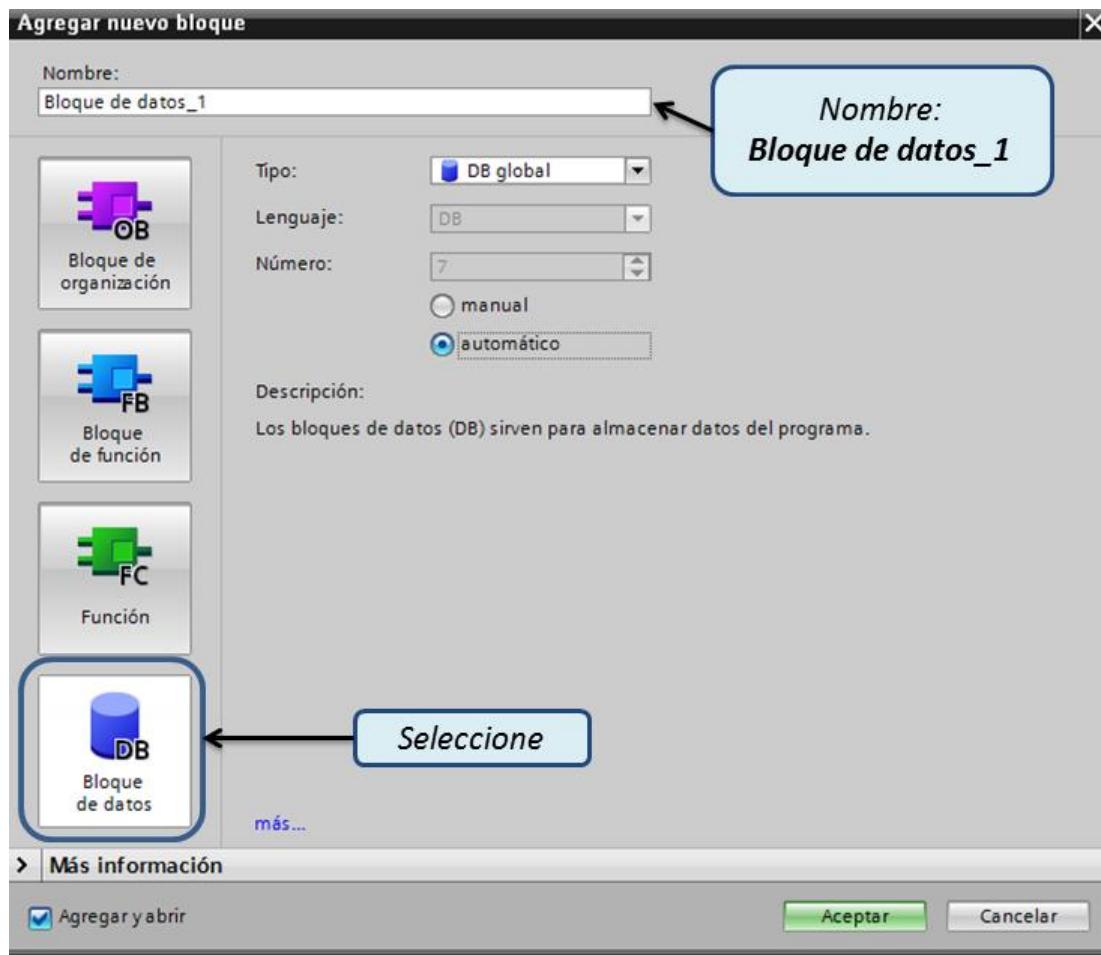
Se desea transmitir un arreglo de tres datos mediante una sola instrucción TSEND\_C y TRCV\_C (una sola ID), en la **Figura 64** se encuentra esquematizado el principio de funcionamiento como base para la posterior programación.



**Figura 64.** Transmisión unidireccional de un arreglo de datos.

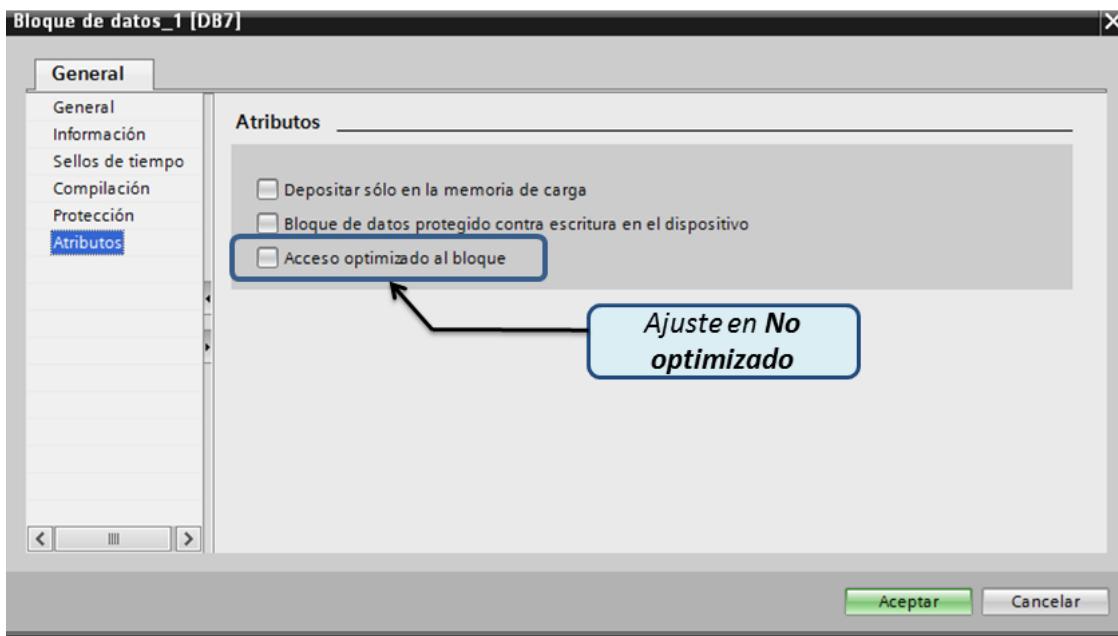
A continuación se realizará el procedimiento pertinente para la configuración y programación de los PLCs:

Diríjase hacia *Árbol de proyecto / Dispositivos / PLC\_1 / Bloques de programas / Agregar nuevo bloque*, en este último seleccione la opción Bloque de datos (DB) y cree un nuevo bloque de datos llamado *Bloque de datos\_1*, ver **Figura 65**.



**Figura 65.** Creación de *Bloque de datos\_1*.

Agregado el *Bloque de datos\_1*, acceda a sus propiedades y ajuste el acceso del bloque a *no optimizado*, ver **Figura 66**.



**Figura 66.** Ajuste de optimización del *Bloque de datos\_1*.

Acceda al *Bloque de datos\_1*, dentro del bloque agregue un nuevo arreglo de datos llamado *DatoPLC1* del tipo *Word* y con límites en [1...3], ver **Figura 67**.

The screenshot shows the SIMATIC Manager interface with the title bar 'Práctica 4 > PLC\_1 [CPU 1214C AC/DC/Rly] > Bloques de programa > Bloque de datos\_1 [DB7]'. Below is a table titled 'Bloque de datos\_1' with columns: Nombre, Tipo de datos, Offset, Valor de arr..., Remanen..., Accesible d..., Visible en .., Valor de a... . The table contains five rows:

	Nombre	Tipo de datos	Offset	Valor de arr...	Remanen...	Accesible d...	Visible en ..	Valor de a...
1	Static							
2	DatoPLC1	Array[1..3] of Word	0.0			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	DatoPLC1[1]	Word	0.0	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	DatoPLC1[2]	Word	2.0	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	DatoPLC1[3]	Word	4.0	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

**Figura 67.** Creación del arreglo *DatoPLC1* dentro del *Bloque de datos\_1* (PLC\_1).

Posteriormente, repita el procedimiento anterior en el PLC\_2 donde el bloque de datos será llamado *Bloque de datos\_1* (respectivo al PLC\_2), en el agregue un arreglo de datos llamado *DatoPLC2* del tipo *Word* y con límites en [1...3], ver **Figura 68**.

	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ...	Valor...
1	Static							
2	DatoPLC2	Array[1..3] of Word	0.0			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	DatoPLC2[1]	Word	0.0	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	DatoPLC2[2]	Word	2.0	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	DatoPLC2[3]	Word	4.0	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figura 68. Creación del arreglo *DatoPLC2* dentro del *Bloque de datos\_1* (PLC\_2).

Luego, en base a las configuraciones realizadas previamente en el bloque TSEND\_C del apartado *Comunicación unidireccional entre los PLCs*, agregue y configure en el *Main [OB1]* del **PLC\_1** un bloque de comunicaciones con las mismas características. En la **Figura 69** es mostrado el bloque TSEND\_C (llamado *Envío de datos PLC1*) ajustado en cada uno de sus parámetros, y como peculiaridad en *DATA* ha sido dispuesto el arreglo *DatoPLC1* perteneciente a *Bloque de datos\_1*.

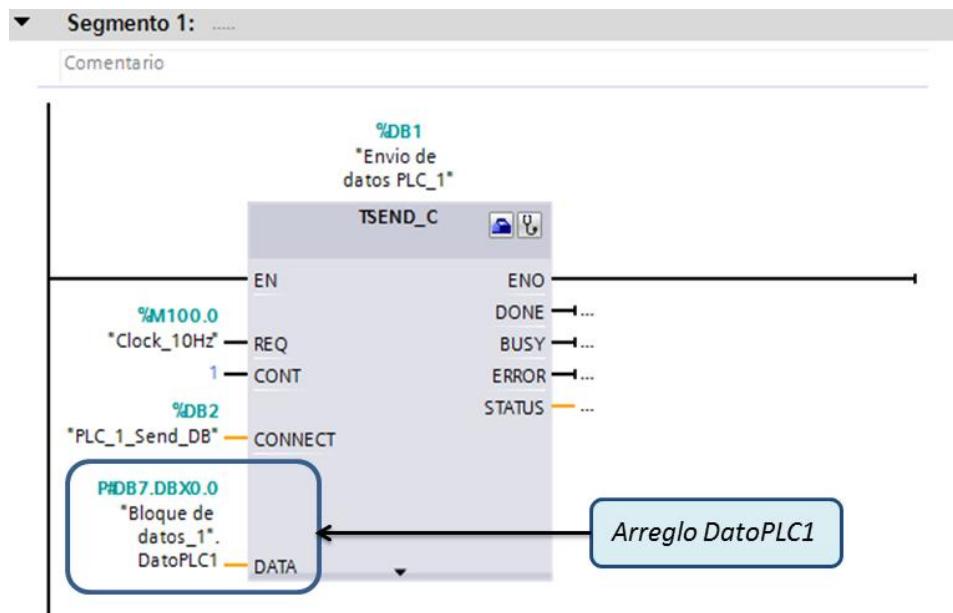
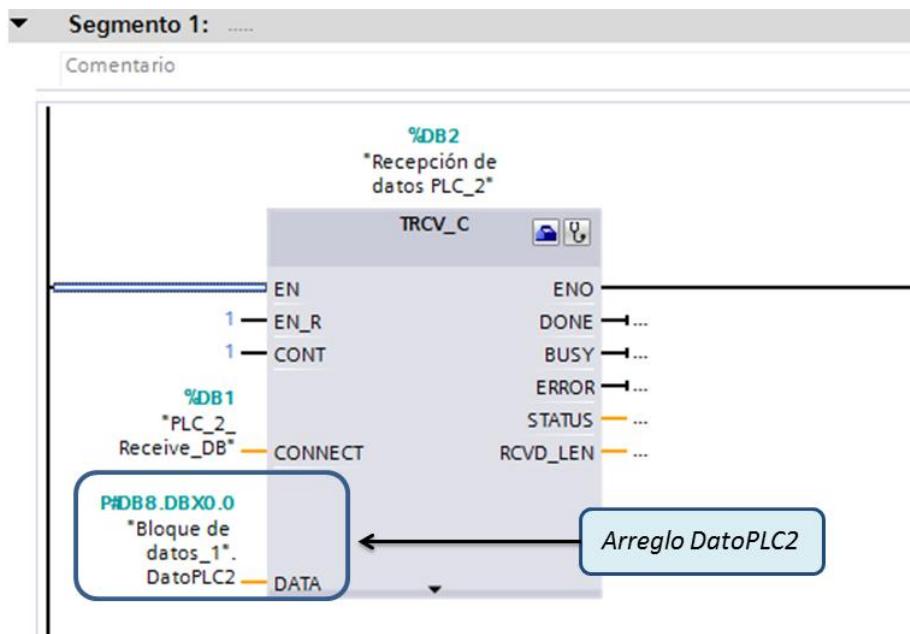


Figura 69. Bloque TSEND\_C *Envío de datos PLC\_1* para la transferencia del arreglo *DatoPLC1*.

Análogamente, repita el procedimiento anterior pero con un bloque TRCV\_C para la recepción del arreglo en el *Main [OB1]* del **PLC\_2**, ver **Figura 70**.



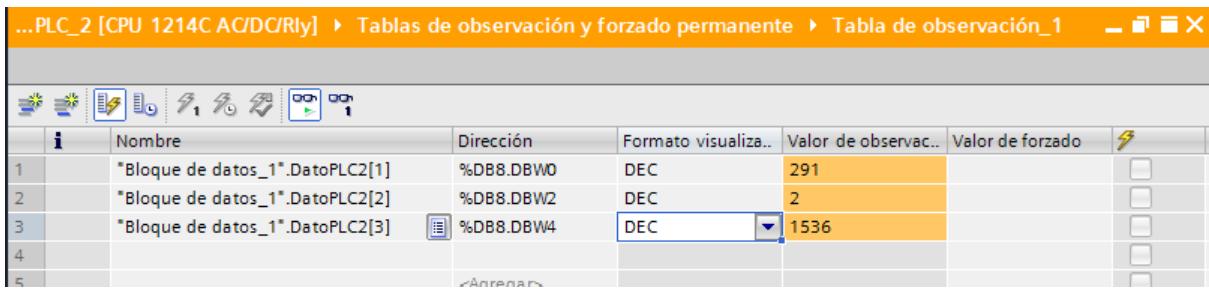
**Figura 70.** Bloque TRCV\_C Recepción de datos PLC\_2.

Finalizados los ajustes y configuraciones, descargue los programas a cada PLC y ejecútelos. En las **Figuras 71** y **72** se evidencian las pruebas concernientes a la transferencia y recepción del arreglo de datos mediante el uso de *Tablas de observación y forzado permanente*.

	i	Nombre	Dirección	Formato visualiza..	Valor de observac..	Valor de forzado	...	C...
1		*Bloque de datos_1*.DatoPLC1[1]	%DB7.DBW0	DEC	291	291	<input checked="" type="checkbox"/>	
2		*Bloque de datos_1*.DatoPLC1[2]	%DB7.DBW2	DEC	2	2	<input checked="" type="checkbox"/>	
3		*Bloque de datos_1*.DatoPLC1[3]	%DB7.DBW4	DEC	1536	1536	<input checked="" type="checkbox"/>	
4			<Agregar>					

**Figura 71.** Tabla de observación\_1 del PLC\_1.

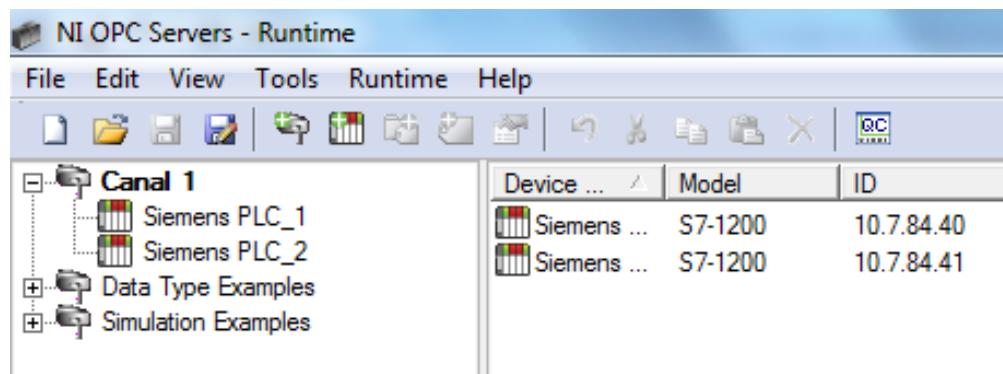
...PLC\_2 [CPU 1214C AC/DC/Rel] ▶ Tablas de observación y forzado permanente ▶ Tabla de observación\_1



	i	Nombre	Dirección	Formato visualiza..	Valor de observac..	Valor de forzado	⚡
1		"Bloque de datos_1".DatoPLC2[1]	%DB8.DBW0	DEC	291		
2		"Bloque de datos_1".DatoPLC2[2]	%DB8.DBW2	DEC	2		
3		"Bloque de datos_1".DatoPLC2[3]	%DB8.DBW4	DEC	1536		
4							
5							

**Figura 72.** Tabla de observación\_1 del PLC\_2.

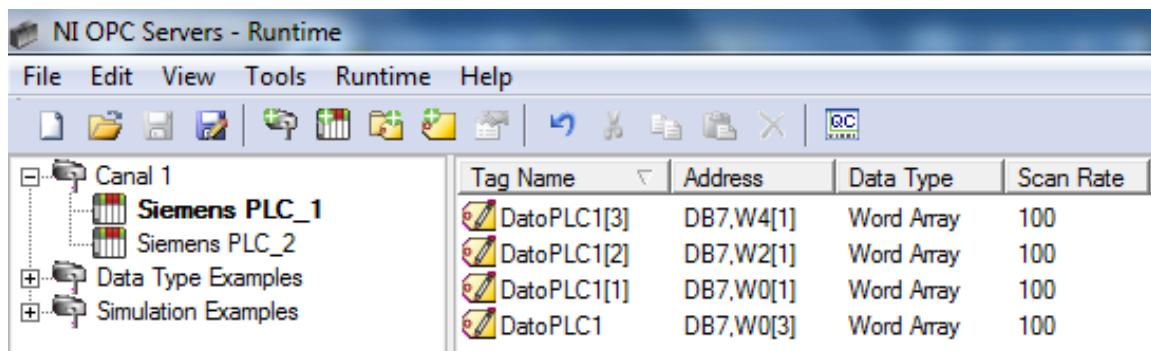
Comprobado el funcionamiento del programa se llevará a cabo el diseño e implementación de la HMI. Ejecute al NI OPC Server y configure un nuevo canal con los respectivos PLCs. En la **Figura 73** es mostrado el canal con los PLCs agregados y configurados.



**Figura 73.** PLCs dispuestos y configurados en el servidor OPC.

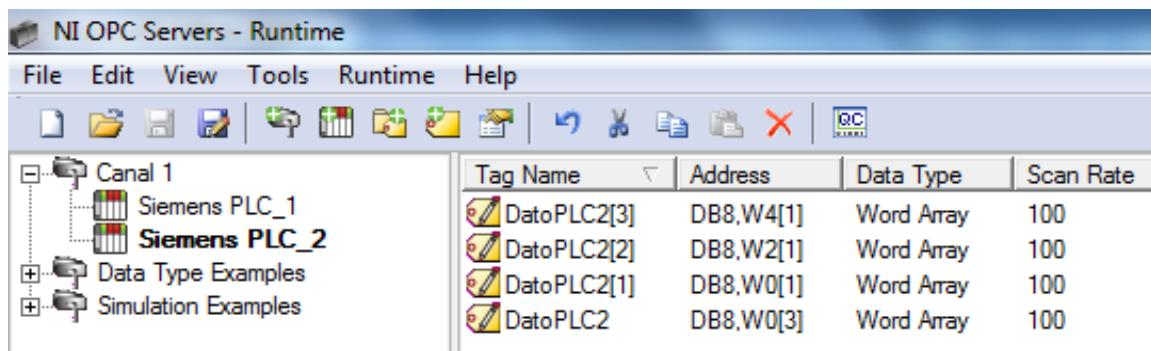
Posteriormente, agregue los *tags* correspondientes al programa, los cuales son:

**PLC\_1:** *DatoPLC1[1]* (DB7,W0[1]), *DatoPLC1[2]* (DB7,W2[1]), *DatoPLC1[3]* (DB7,W4[1]), *DatoPLC1* (DB7,W0[3]). Véase la **Figura 74**, donde aparecen los *tags* agregados.



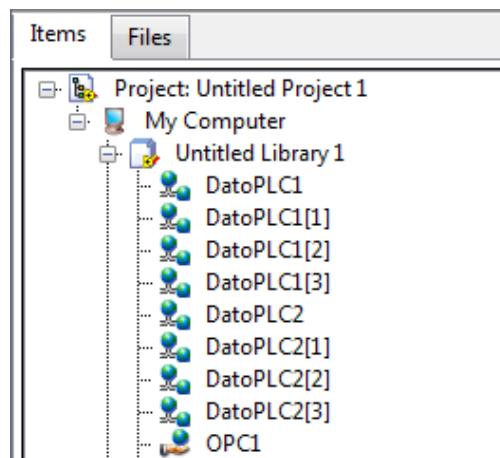
**Figura 74.** Tags creados en PLC\_1.

**PLC\_1:** *DatoPLC2[1]* (DB8,W0[1]), *DatoPLC2[2]* (DB8,W2[1]), *DatoPLC2[3]* (DB8,W4[1]), *DatoPLC2* (DB8,W0[3]). Véase la **Figura 75**, donde aparecen los *tags* agregados.



**Figura 75.** Tags creados en PLC\_2.

Configurado el servidor OPC ejecute LabVIEW, dentro cree un *Blank Project* y configure la instancia *OPC Client*, de tal manera se creará la librería contenedora de las variables, luego agregue las variables en función de los *tags* creados previamente en el servidor OPC, ver **Figura 76**.



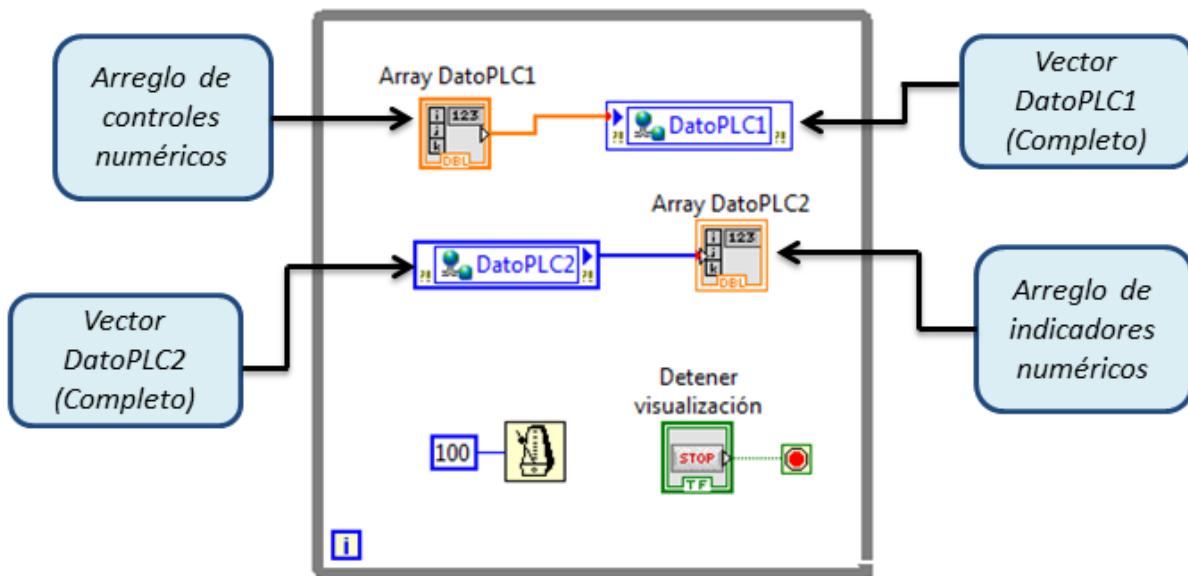
**Figura 76.** Variables agregadas a la librería.

Finalizada la creación de las distintas variables para lectura y escritura en la interfaz, cree la interfaz tal como el que se muestra en la **Figura 77 (Front Panel)** y **Figura 78 (Block Diagram)**, donde se utilizarán *Numeric Controls* (datos de entrada) y *Numeric Indicators* (datos de salida) bajo una configuración de arreglo (Array) encontrada en *Controls / Moderns / Array, Matrix & Cluster*.

**Nota:** La decoración es alternativa, en ella puede implementar elementos presentes en *Controls / Decorations*.

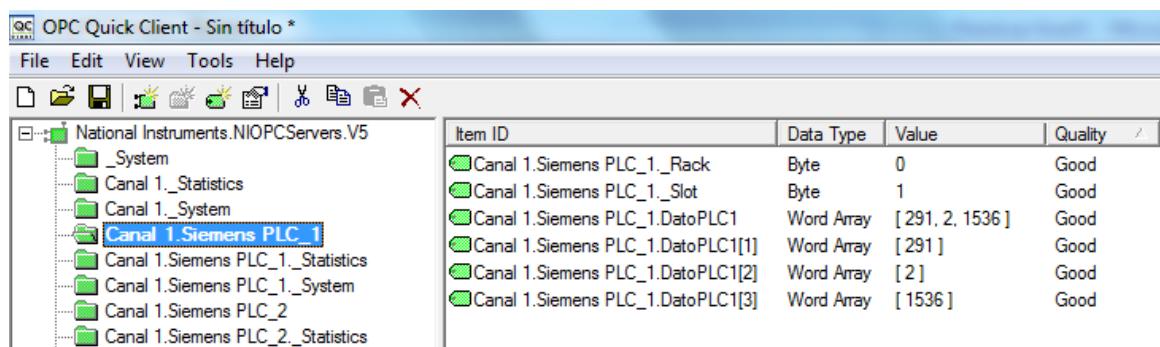


**Figura 77.** Interfaz para el control y supervisión del paquete de datos.



**Figura 78.** Block Diagram de la Interfaz para el control y supervisión del paquete de datos enviados/recibidos.

Terminado el diseño de la interfaz, verifique el correcto monitoreo de los *tags* (por cada PLC) en el *Quick Client*, ver **Figura 79** (PLC\_1) y **Figura 80** (PLC\_2).



**Figura 79.** Verificación del correcto monitoreo de los *tags* del PLC\_1.

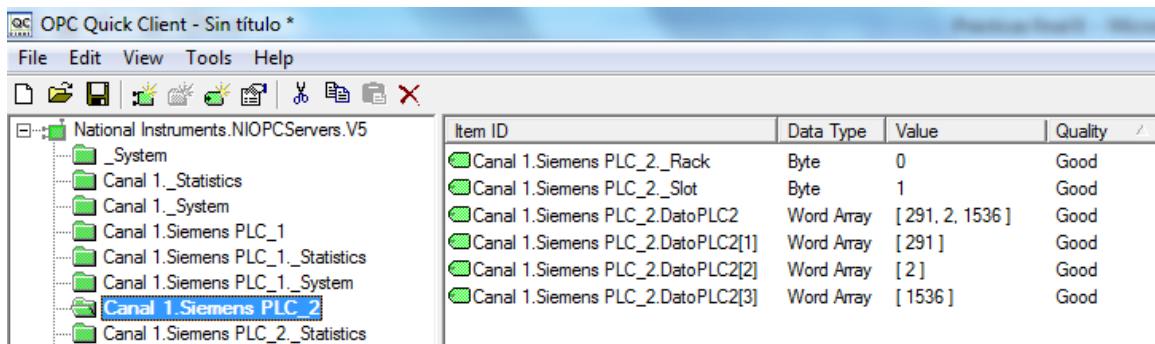


Figura 80. Verificación del correcto monitoreo de los *tags* del PLC\_1.

Finalmente ejecute la interfaz (ver **Figura 81**).

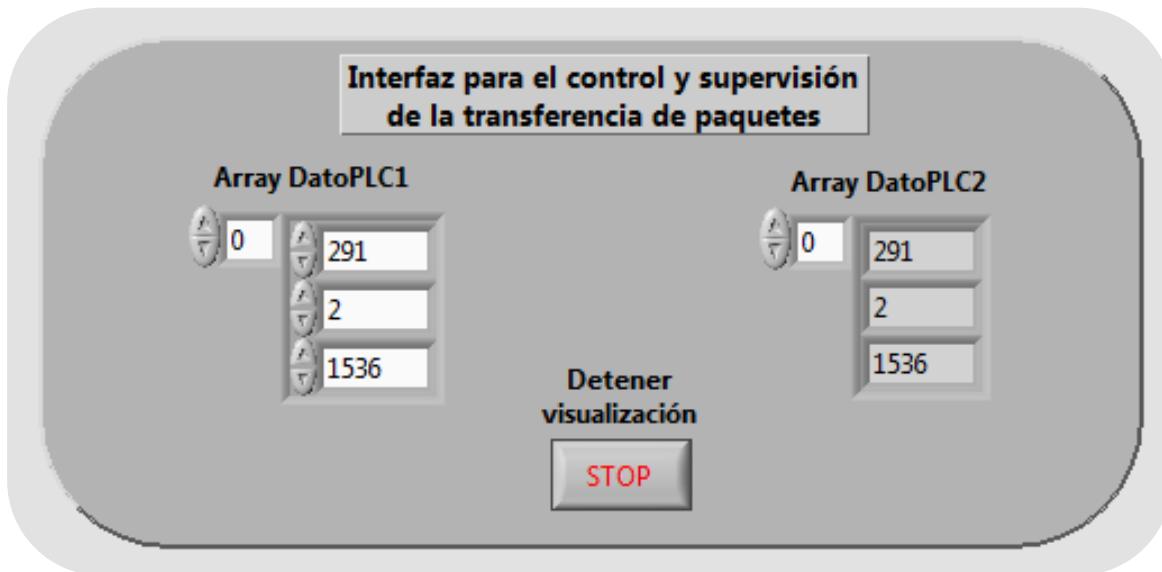


Figura 81. Interfaz para el control y supervisión del paquete de datos en ejecución.

#### 4.5. Diseño de un secuenciador doble controlado y supervisado mediante una HMI.

Con el fin de poner en práctica la comunicación entre los dispositivos en una aplicación de mayor alcance ilustrativo y didáctico, se diseñará un secuenciador doble, con las siguientes características a considerar:

- Ambos PLCs ejecutarán un algoritmo que permita la activación y desactivación secuencial de tres (3) salidas (Primero el PLC\_1 y luego el PLC\_2) de forma cíclica. Además cada PLC deberá ser capaz de contar el número de veces que se ha ejecutado la secuencia, y finalizado el conteo (establecido por el usuario), se activará una cuarta salida (en cada PLC).
- Se hará uso de una comunicación bidireccional entre los PLCs para establecer el funcionamiento cíclico.
- El PLC\_1 será el encargado de iniciar y detener el proceso (Maestro).
- Deberán asociarse variables a PT (tiempo preseleccionado) en los temporizadores y CV (valor de conteo actual) y PV (valor de conteo predeterminado) en los contadores para su posterior uso y control en la HMI
- Asocie marcas de bits al parámetro CONT de cada bloque de envío o recepción de datos para activar o desactivar la comunicación.
- Asocie variables al parámetro STATUS para almacenar los estados de funcionamiento del bloque (comunicación).

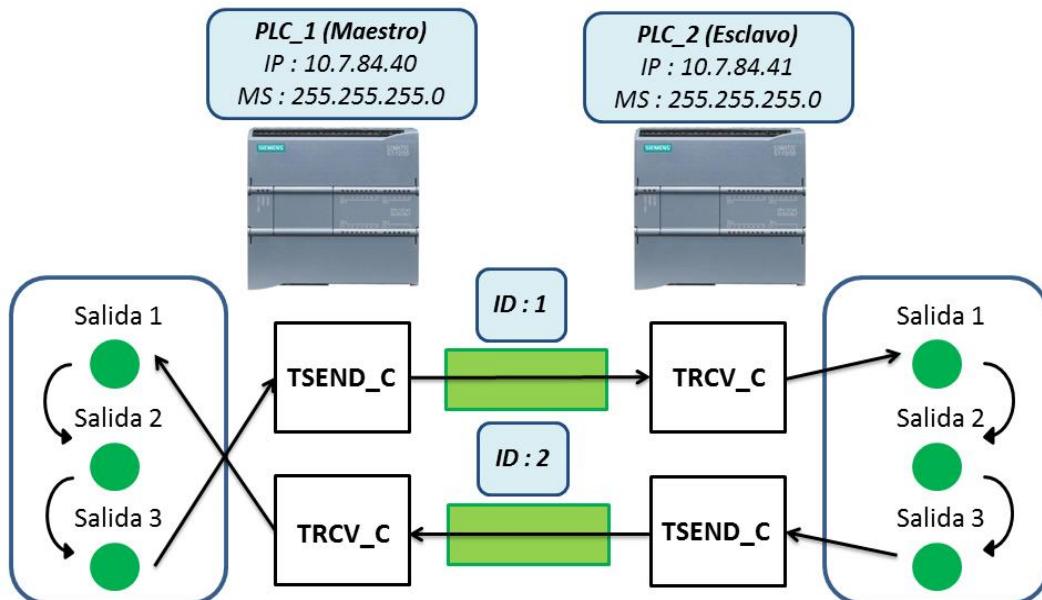
En cuanto a la HMI, considere lo siguiente:

- Cada PLC tendrá que contar con sus respectivos controles e indicadores de proceso y comunicación. Controles de proceso y comunicación: INICIO, PARADA (únicamente PLC\_1), Ajuste de tiempo (velocidad del secuenciador), Ajuste de cuenta (conteo de repeticiones de secuencia), Reiniciar conteo, Activar comunicación, Desactivar comunicación. Indicadores de proceso y comunicación: Indicador 1, Indicador 2, Indicador 3, Indicador de Finalización de cuenta, Conteo

actual, Estado del PLC (programado con un *String Indicator*), Comunicación activa, Envió y recepción

- Haga uso de los datos extraídos de cada bloque de comunicación (parámetro STATUS), estructuras *While Loop* y *Case*, para programar el comportamiento (velocidad) del indicador Envió y recepción. Considere los siguientes valores de STATUS: 7000hex (28672dec, encendido), 7001hex (28673dec, ciclo lento), 7002hex (28674dec, ciclo rápido), 7004hex (28676, ciclo lento), 80C4hex (32964dec, apagado).

En la **Figura 82** aparece un esquema básico de funcionamiento del secuenciador doble.



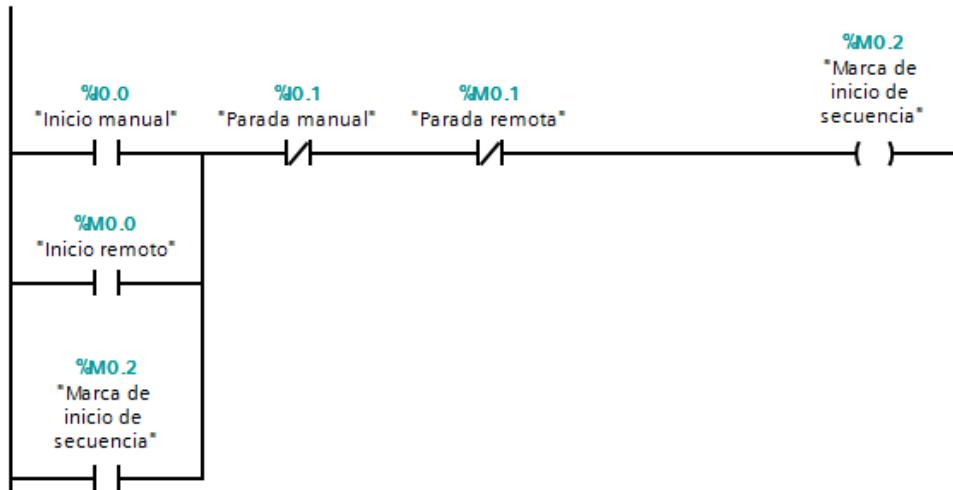
**Figura 82.** Esquema de funcionamiento del secuenciador doble (transmisión bidireccional).

Para empezar, vaya al *Main [OB1]* del **PLC\_1** y programe en el la siguiente lógica:

**Segmento 1:** lógica de inicio y detención del proceso, ver **Figura 83**.

**Segmento 1:** Lógica de inicio y detención del proceso

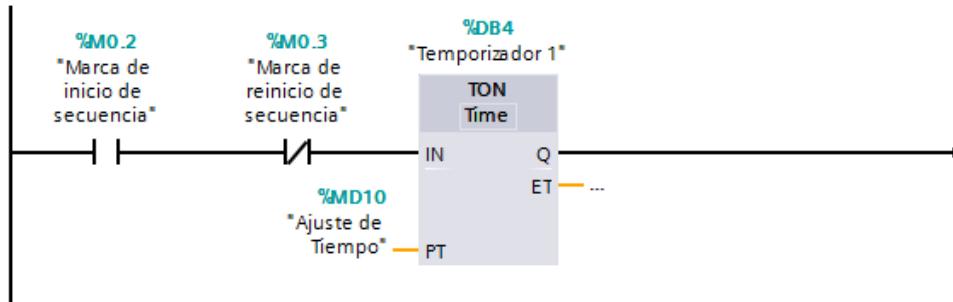
- Se pueden evidenciar los contactos de Inicio (manual y remoto) y Parada (manual y remoto) dispuestos en una lógica de auto enganche para la activación y desactivación del bit "Marca de inicio de secuencia"


**Figura 83.** Segmento 1 (PLC\_1).

**Segmento 2:** lógica de secuencia, ver **Figura 84**.

**Segmento 2:** Lógica de secuencia

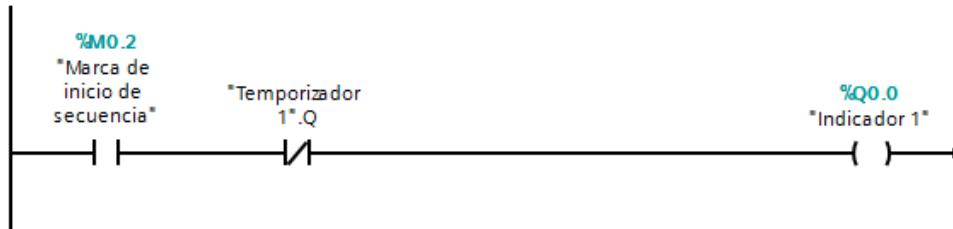
Activación del "Temporizador 1", asociado al "Indicador 1"


**Figura 84.** Segmento 2 (PLC\_1).

**Segmento 3:** lógica de secuencia, ver **Figura 85**.

**Segmento 3:** Lógica de secuencia

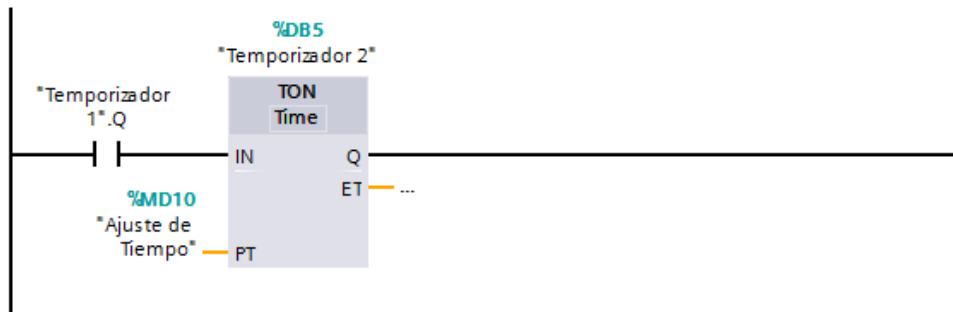
Activación del "Indicador 1", asociado al "Temporizador 1"


**Figura 85.** Segmento 3 (PLC\_1).

**Segmento 4:** lógica de secuencia, ver **Figura 86**.

**Segmento 4:** Lógica de secuencia

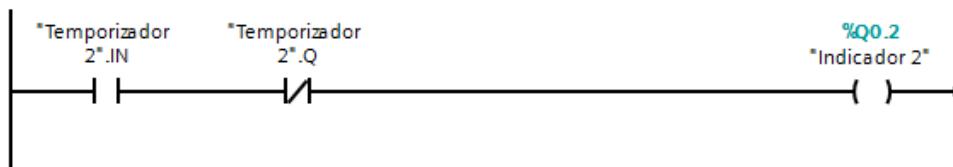
Activación del "Temporizador 2" en función del "Temporizador 1"


**Figura 86.** Segmento 3 (PLC\_1).

**Segmento 5:** lógica de secuencia, ver **Figura 87**.

**Segmento 5:** Lógica de secuencia

Activación del "Indicador 2" en función del "Temporizador 2"


**Figura 87.** Segmento 5 (PLC\_1).

**Segmento 6:** lógica de secuencia, ver **Figura 88**.

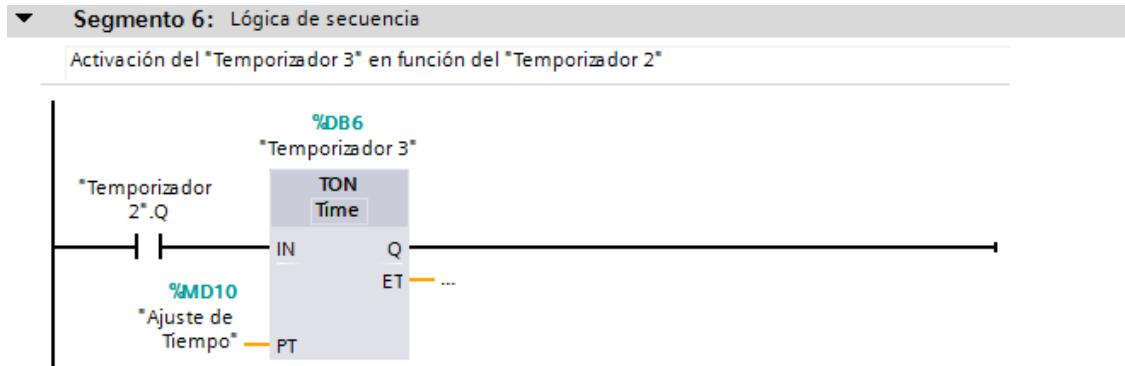


Figura 88. Segmento 6 (PLC\_1).

**Segmento 7: lógica de secuencia, ver Figura 89.**

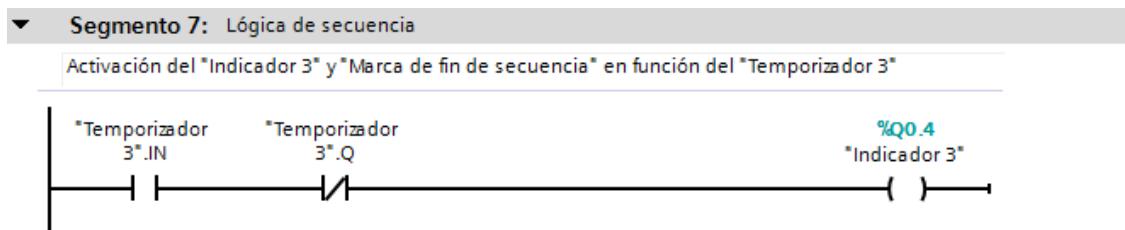


Figura 89. Segmento 7 (PLC\_1).

**Segmento 8: lógica de conteo de secuencias, ver Figura 90.**

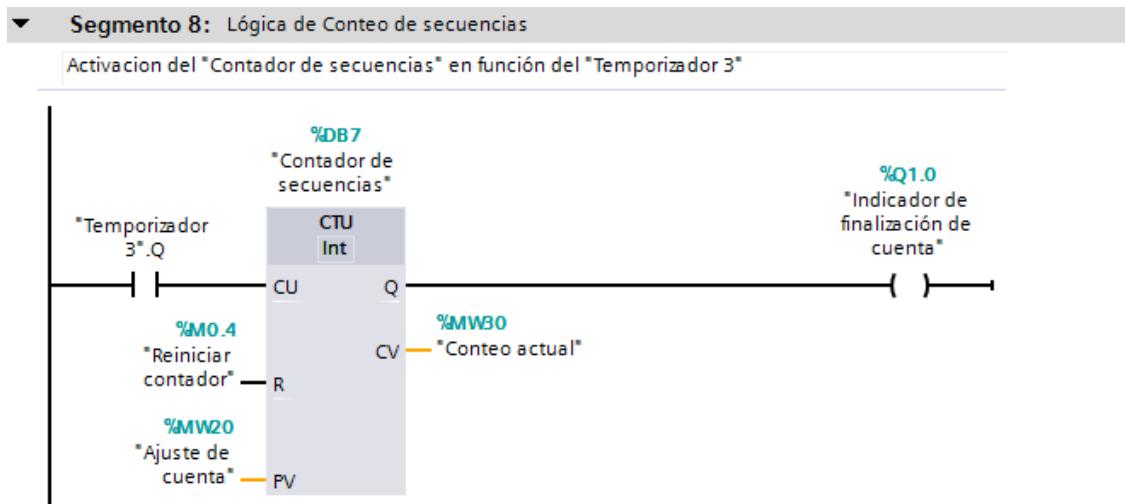
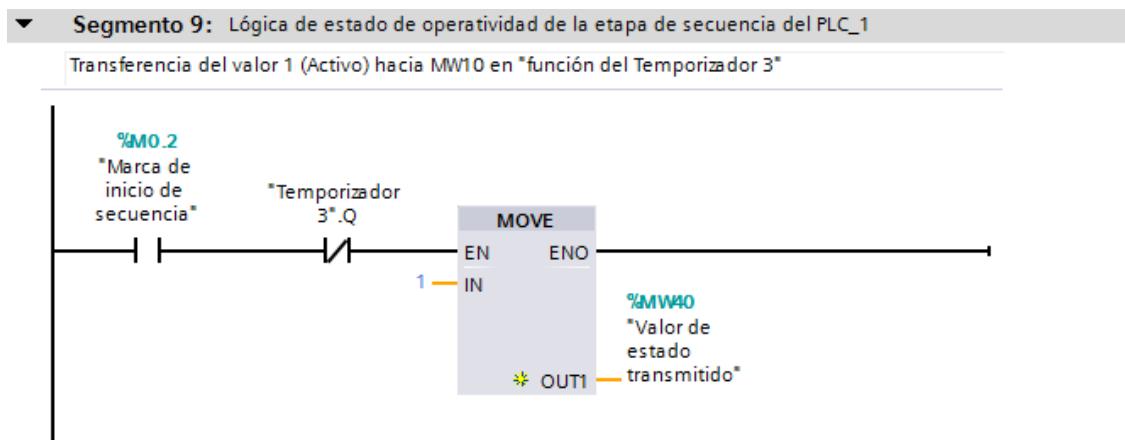


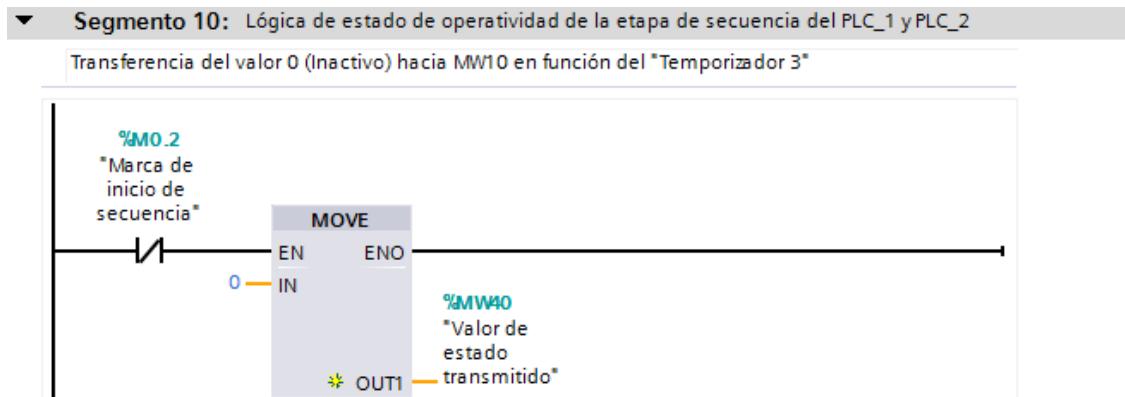
Figura 90. Segmento 8 (PLC\_1).

**Segmento 9:** lógica de estado de operatividad de la etapa de secuencia del PLC\_2, ver **Figura 91**.



**Figura 91.** Segmento 9 (PLC\_1).

**Segmento 10:** lógica de estado de operatividad de la etapa de secuencia del PLC\_1 y PLC\_2, ver **Figura 92**.

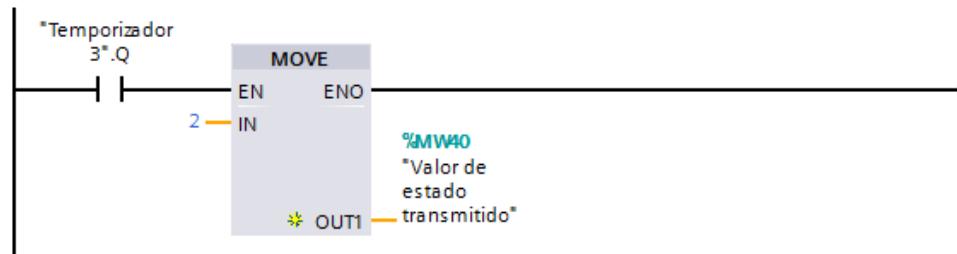


**Figura 92.** Segmento 10 (PLC\_1).

**Segmento 11:** lógica de estado de operatividad de la etapa de secuencia del PLC\_2, ver **Figura 93.**

**Segmento 11:** Lógica de estado de operatividad de la etapa de secuencia del PLC\_2

Transferencia del valor 2 (Activo) hacia MW10 en "función del Temporizador 3"


**Figura 93.** Segmento 11 (PLC\_1).

**Segmento 12:** bloque de transferencia de datos (en MW40) al PLC\_2, ver **Figura 94**.

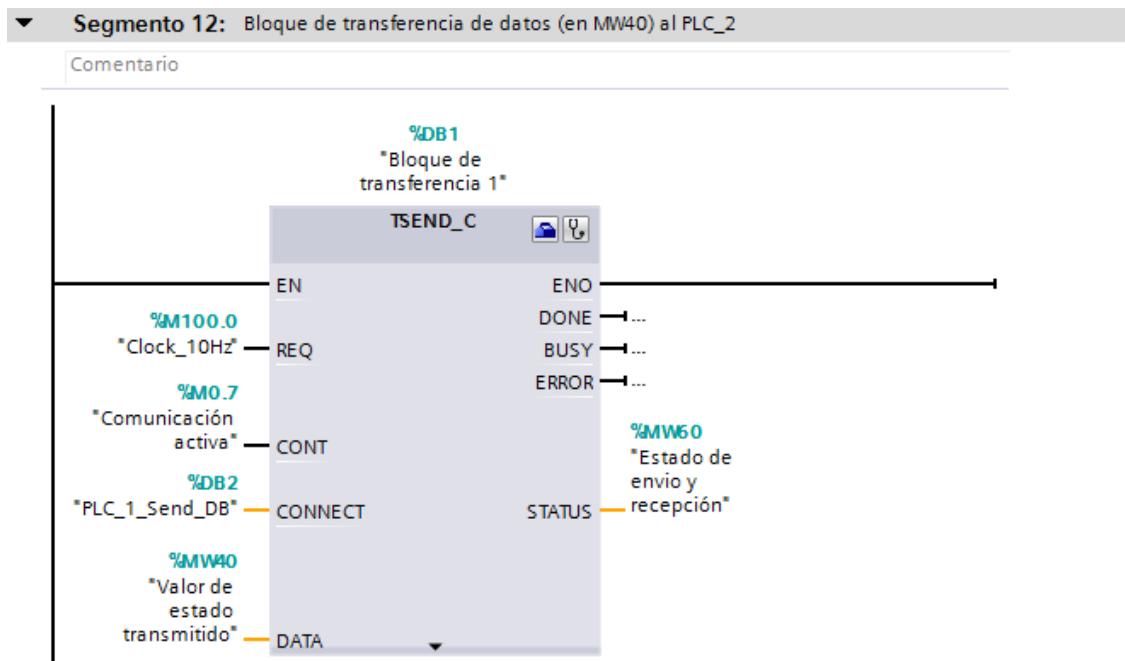
Configuraciones de **Parámetros de la conexión**:

- Interlocutor: PLC\_2.
- Datos de conexión (Local): <nuevo>, *PLC\_1\_Send\_DB*.
- Datos de conexión (Interlocutor): <nuevo>, *PLC\_2\_Receive\_DB*.
- Tipo de conexión: TCP.
- ID de conexión (dec): 1.
- Detalles de dirección: 2000.

Configuraciones de **Parámetros del bloque**:

- Inicio de la petición (REQ): M100.0 (Clock\_10 Hz).

- Estado de la conexión (CONT): M0.7 (Comunicación activa).
- Puntero de conexión asociado (CONNECT): *PLC\_1\_Send\_DB*.
- Área de transmisión (DATA): MW40 (Valor de estado transmitido).

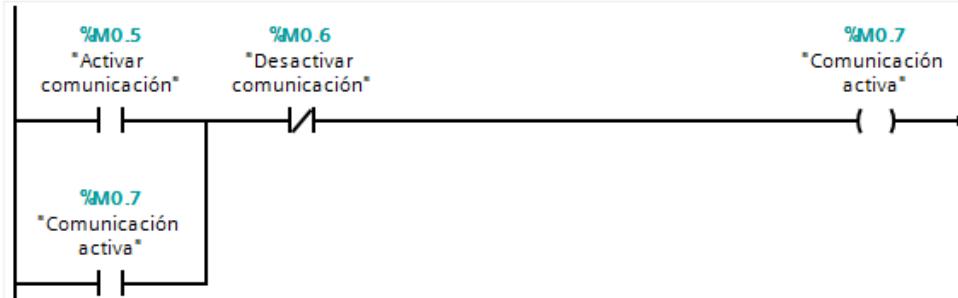


**Figura 94.** Segmento 12 (PLC\_1).

**Segmento 13:** lógica para activar establecer conexión (estado de la conexión), ver **Figura 95**.

**Segmento 13:** Lógica para activar establecer conexión (estado de la conexión)

Comentario


**Figura 95.** Segmento 13 (PLC\_1).

**Segmento 14:** bloque de recepción de datos (en MW50) desde el PLC\_2, ver **Figura 96.**

Configuraciones de **Parámetros de la conexión:**

- Interlocutor: PLC\_2.
- Datos de conexión (Local): <nuevo>, PLC\_1\_Receive\_DB.
- Datos de conexión (Interlocutor): <nuevo>, PLC\_2\_Send\_DB.
- Tipo de conexión: TCP.
- ID de conexión (dec): 2.
- Detalles de dirección: 2001.

Configuraciones de **Parámetros del bloque:**

- Área de recepción (EN\_R): 1.

- Estado de la conexión (CONT): M0.7 (Comunicación activa).
- Puntero de conexión asociado (CONNECT): *PLC\_1\_Receive\_DB*.
- Área de transmisión (DATA): MW50 (Valor de estado recibido).

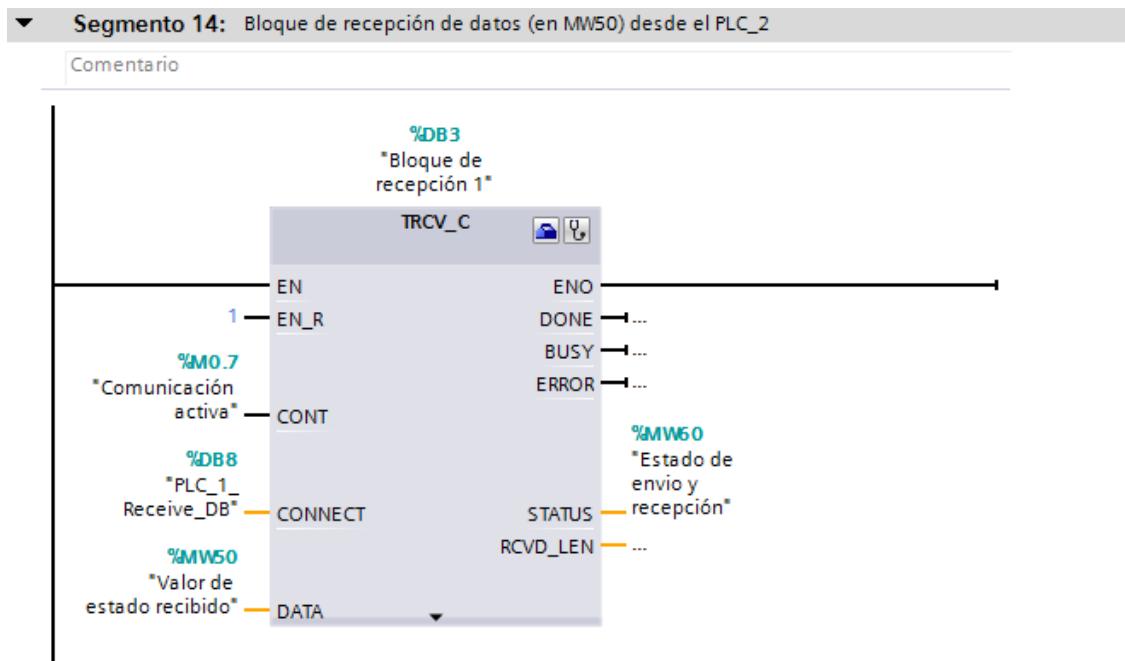


Figura 96. Segmento 14 (PLC\_1).

**Segmento 15:** lógica de reinicio de secuencia en función del PLC\_2, ver Figura 97.

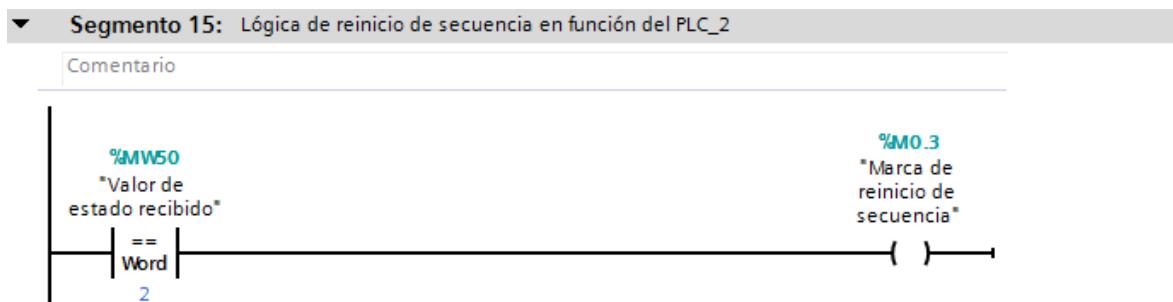


Figura 97. Segmento 15 (PLC\_1).

Finalizado el programa en el **PLC\_1** vaya al *Main[OB1]* del **PLC\_2** y desarrolle el programa como se muestra a continuación:

**Segmento 1:** bloque de recepción de datos (en MW10) desde el PLC\_1, ver **Figura 98.**

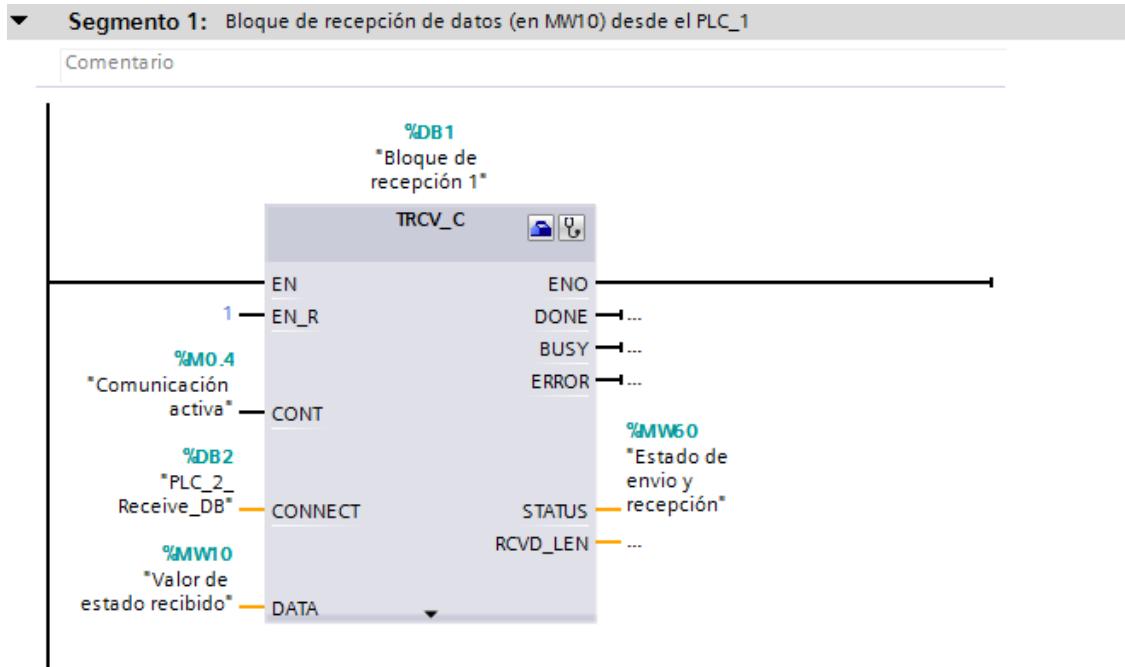
Configuraciones de **Parámetros de la conexión:**

- Interlocutor: PLC\_1.
- Datos de conexión (Local): *PLC\_2\_Receive\_DB*.
- Datos de conexión (Interlocutor): *PLC\_1\_Send\_DB*.
- Tipo de conexión: TCP.
- ID de conexión (dec): 1.
- Detalles de dirección: 2000.

Configuraciones de **Parámetros del bloque:**

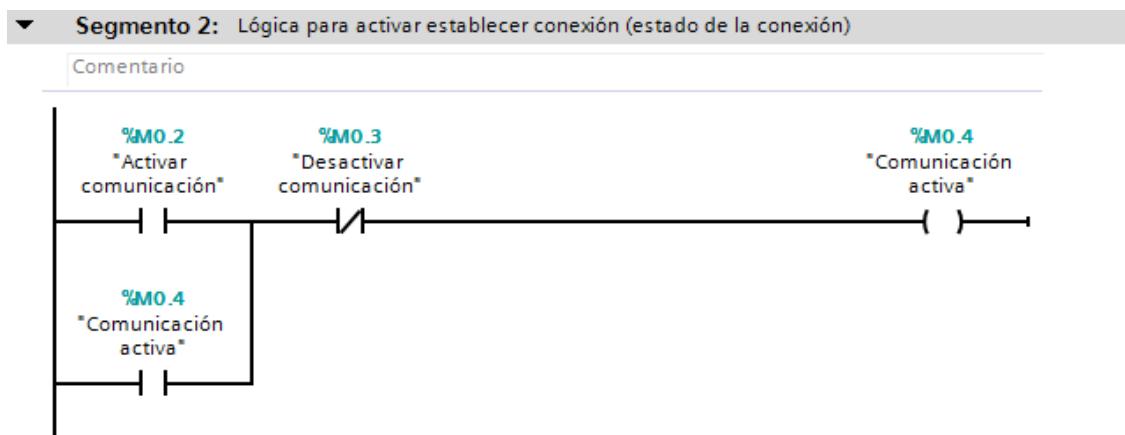
- Área de recepción (EN\_R): 1.
- Estado de la conexión (CONT): M0.4 (*Comunicación activa*).
- Puntero de conexión asociado (CONNECT): *PLC\_1\_Receive\_DB*.

- Área de transmisión (DATA): MW10 (*Valor de estado recibido*).



**Figura 98.** Segmento 1 (PLC\_2).

**Segmento 2:** lógica para activar establecer conexión (estado de la conexión), ver **Figura 99.**

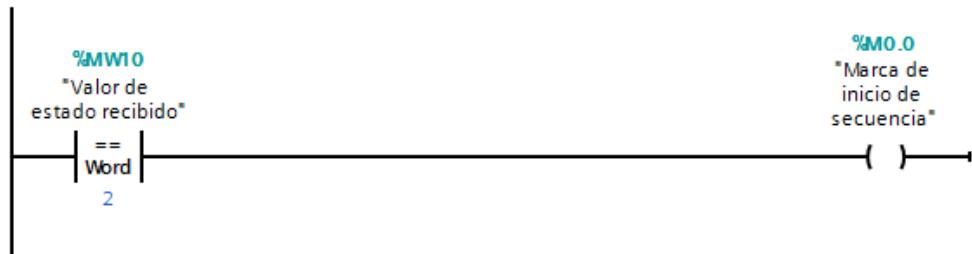


**Figura 99.** Segmento 2 (PLC\_2).

**Segmento 3:** lógica de inicio de secuencia en función del PLC\_1, ver **Figura 100.**

**Segmento 3:** Lógica de inicio de secuencia en función del PLC\_1

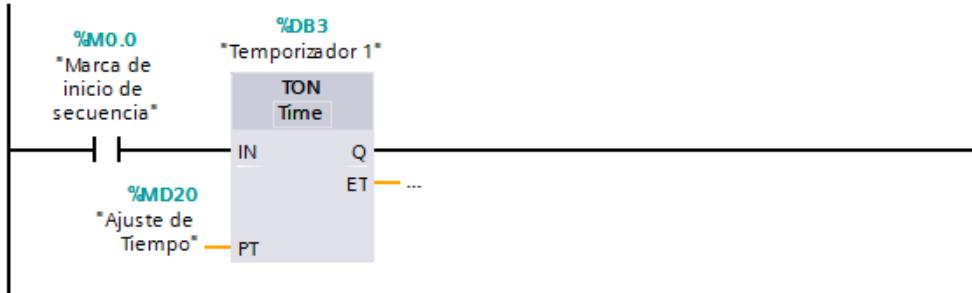
Comentario


**Figura 100.** Segmento 3 (PLC\_2).

**Segmento 4:** lógica de secuencia, ver **Figura 101**.

**Segmento 4:** Lógica de secuencia

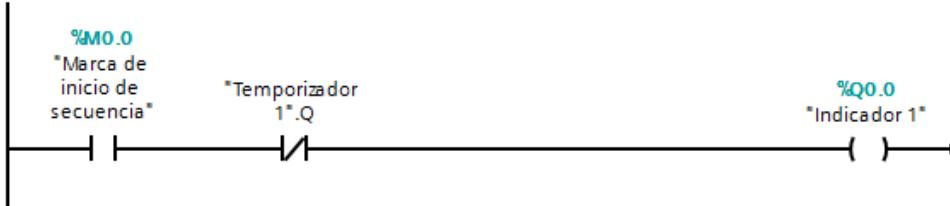
Activación del "Temporizador 1" en función del bit "Marca de inicio de secuencia"


**Figura 101.** Segmento 4 (PLC\_2).

**Segmento 5:** lógica de secuencia, ver **Figura 102**.

**Segmento 5:** Lógica de secuencia

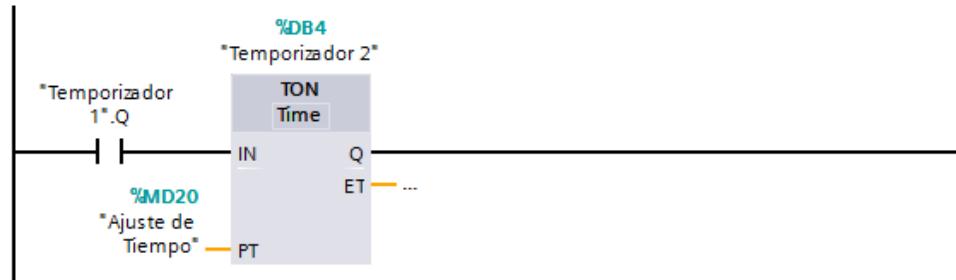
Activación del "Indicador 1" en función del bit "Marca de inicio de secuencia"


**Figura 102.** Segmento 5 (PLC\_2).

**Segmento 6:** lógica de secuencia, ver **Figura 103**.

**Segmento 6:** Lógica de secuencia

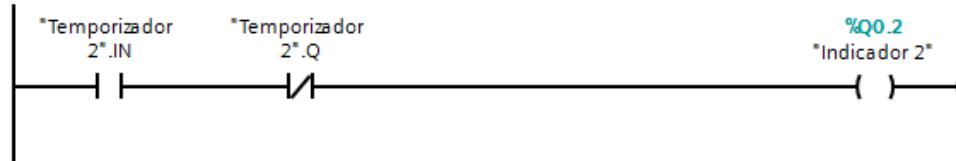
Activación del "Temporizador 2" en función del "Temporizador 1"


**Figura 103.** Segmento 6 (PLC\_2).

**Segmento 7:** lógica de secuencia, ver **Figura 104**.

**Segmento 7:** Lógica de secuencia

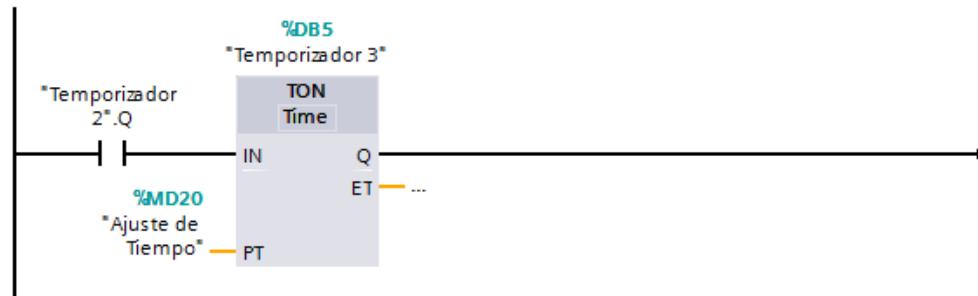
Activación del "Indicador 2" en función del "Temporizador 2"


**Figura 104.** Segmento 7 (PLC\_2).

**Segmento 8:** lógica de secuencia, ver **Figura 105**.

**Segmento 8:** Lógica de secuencia

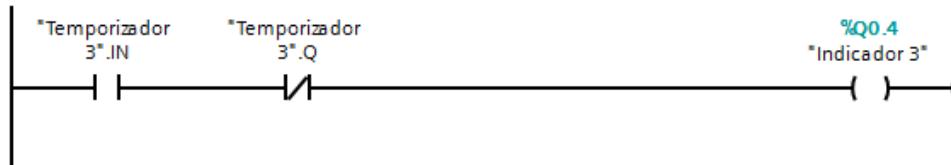
Activación del "Temporizador 3" en función del "Temporizador 2"


**Figura 105.** Segmento 8 (PLC\_2).

**Segmento 9:** lógica de secuencia, ver **Figura 106**.

**Segmento 9:** Lógica de secuencia

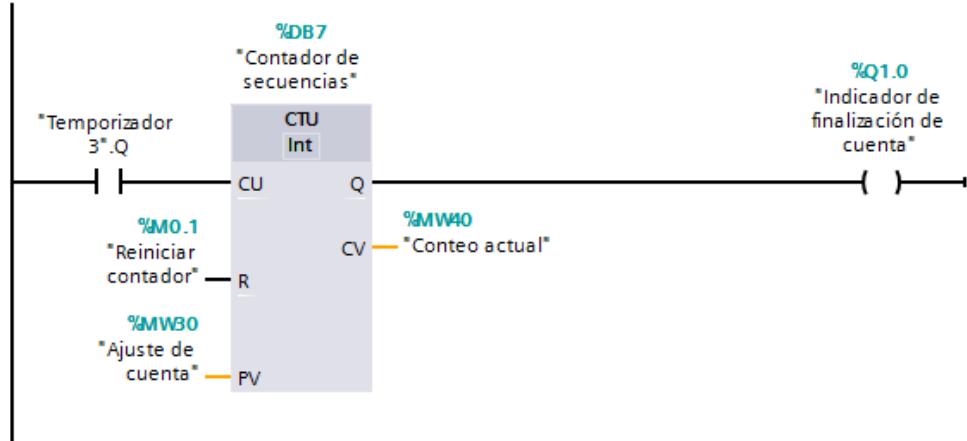
Activación del "Indicador 3" en función del "Temporizador 3"


**Figura 106.** Segmento 9 (PLC\_2).

**Segmento 10:** lógica de conteo de secuencias, ver **Figura 107**.

**Segmento 10:** Lógica de Conteo de secuencias

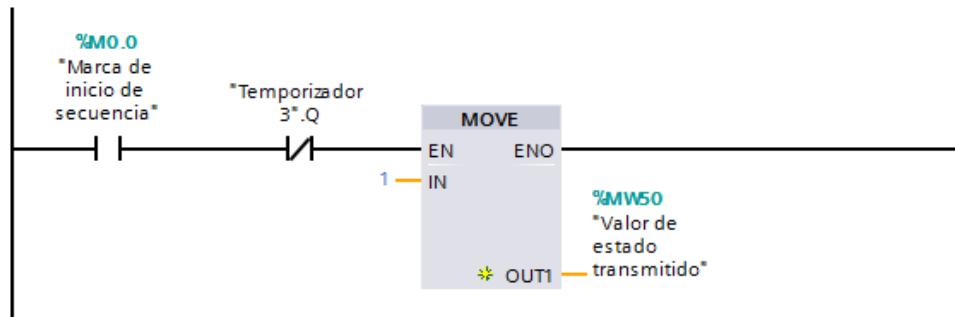
Activación del "Contador de secuencias" en función del "Temporizador 3"


**Figura 107.** Segmento 10 (PLC\_2).

**Segmento 11:** lógica estado de operatividad de la etapa de secuencia del PLC\_2, ver **Figura 108**.

**Segmento 11:** Lógica estado de operatividad de la etapa de secuencia del PLC\_2

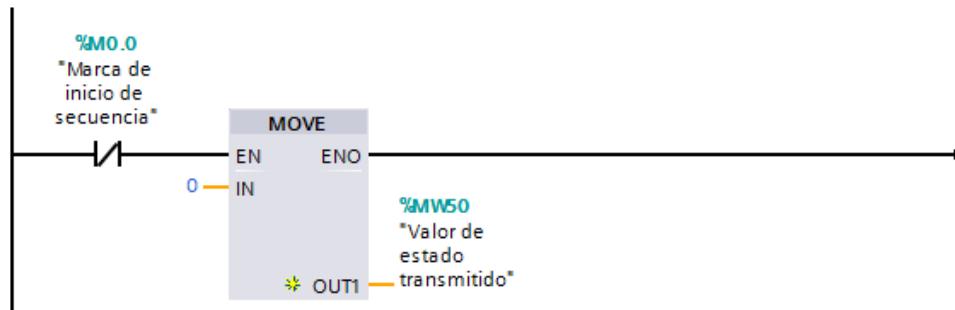
Transferencia del valor 1 (Activo) hacia MW50 en función del "Temporizador 3"


**Figura 108.** Segmento 11 (PLC\_2).

**Segmento 12:** lógica estado de operatividad de la etapa de secuencia del PLC\_2, ver **Figura 109**.

**Segmento 12:** Lógica estado de operatividad de la etapa de secuencia del PLC\_2

Transferencia del valor 0 (Inactivo) hacia MW50 en función del "Temporizador 3"


**Figura 109.** Segmento 12 (PLC\_2).

**Segmento 13:** lógica estado de operatividad de la etapa de secuencia del PLC\_2, ver **Figura 110**.

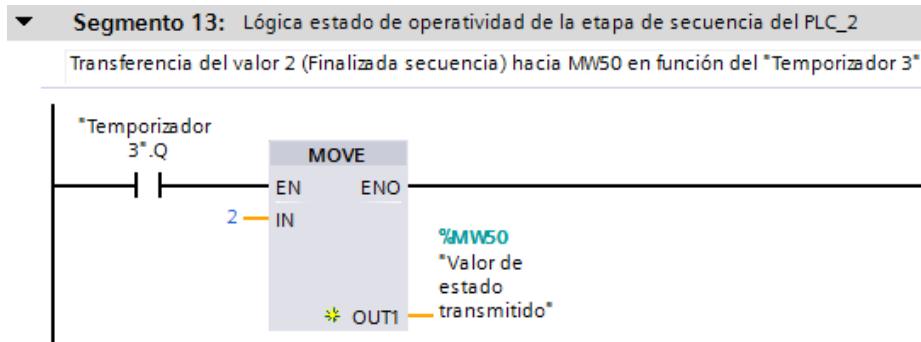


Figura 110. Segmento 13 (PLC\_2).

**Segmento 14:** bloque de transferencia de datos (en MW50) al PLC\_1, ver **Figura 111.**

Configuraciones de **Parámetros de la conexión:**

- Interlocutor: PLC\_1.
- Datos de conexión (Local): PLC\_2\_Send\_DB.
- Datos de conexión (Interlocutor): PLC\_1\_Receive\_DB.
- Tipo de conexión: TCP.
- ID de conexión (dec): 1.
- Detalles de dirección: 2001.

Configuraciones de **Parámetros del bloque:**

- Inicio de la petición (REQ): M100.0 (Clock\_10 Hz).

- Estado de la conexión (CONT): M0.4 (Comunicación activa).
- Puntero de conexión asociado (CONNECT): PLC\_2\_Send\_DB.
- Área de transmisión (DATA): MW50 (Valor de estado transmitido).

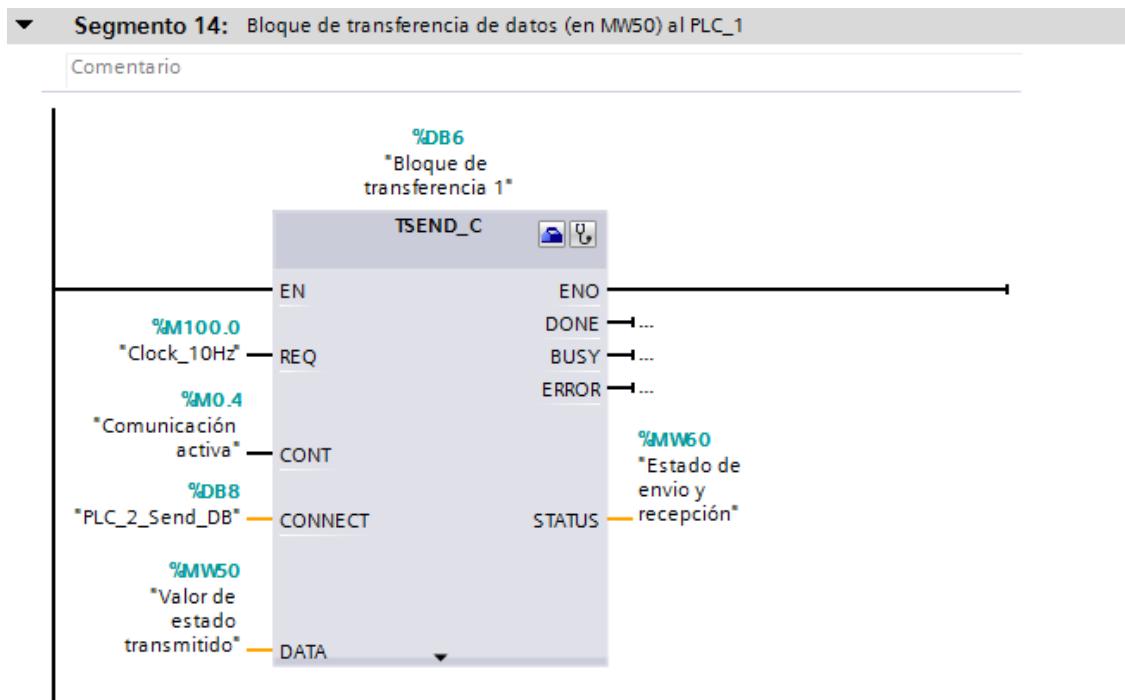


Figura 111. Segmento 14 (PLC\_2).

Finalizada la construcción del programa, se llevará a cabo el diseño e implementación de la HMI. Ejecute al NI OPC Server y configure un nuevo canal con los respectivos PLCs. En la **Figura 112** es mostrado el canal creado con sus respectivos PLCs agregados y configurados.

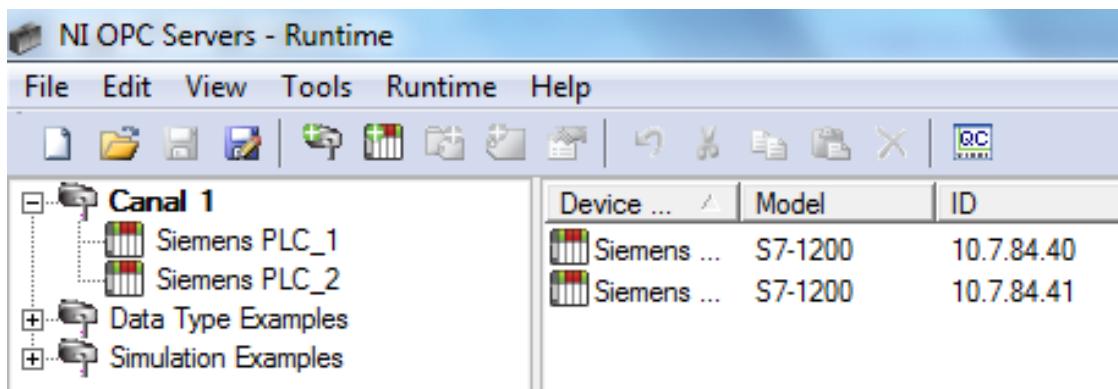


Figura 112. Canal 1 y dispositivos (PLCs) configurados.

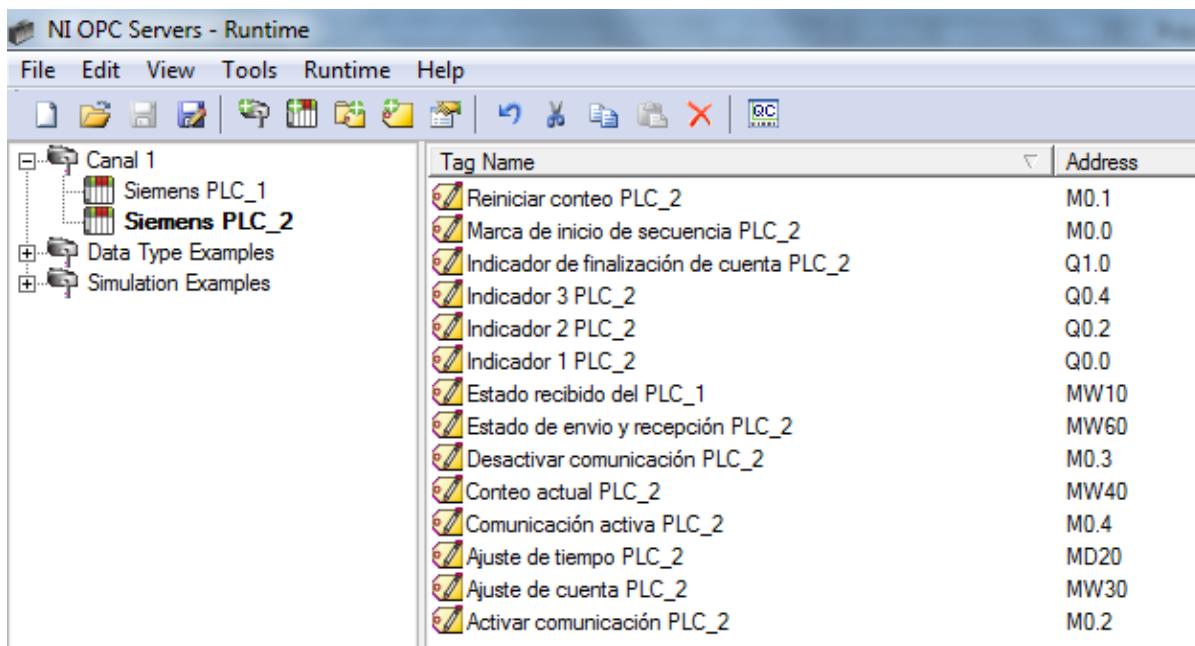
Posteriormente agregue los *tags* correspondientes al programa los cuales son:

**PLC\_1:** Reiniciar conteo PLC\_1 (M0.4), Parada remota PLC\_1 (M0.1), Marca de inicio de secuencia PLC\_1 (M0.2), Inicio remoto PLC\_1 (M0.0), Indicador de finalización de cuenta PLC\_1 (Q1.0), Indicador 3 PLC\_1 (Q0.4), Indicador 2 PLC\_1 (Q0.2), Indicador 1 PLC\_1 (Q0.0), Estado de envío y recepción PLC\_1 (MW60), Desactivar comunicación PLC\_1 (M0.6), Conteo actual PLC\_1 (MW30), Comunicación activa PLC\_1 (M0.7), Ajuste de tiempo PLC\_1 (MD10), Ajuste de cuenta PLC\_1 (MW20), Estado recibido del PLC\_2 (MW50), Activar comunicación PLC\_1 (M0.5). Véase la Figura 113 donde aparecen los *tags* agregados.

NI OPC Servers - Runtime	
File Edit View Tools Runtime Help <span style="float: right;">QC</span>	
<span style="float: left; border: 1px solid #ccc; padding: 2px;">Canal 1</span> <span style="float: left; border: 1px solid #ccc; padding: 2px;">Siemens PLC_1</span> <span style="float: left; border: 1px solid #ccc; padding: 2px;">Siemens PLC_2</span> <span style="float: left; border: 1px solid #ccc; padding: 2px;">Data Type Examples</span> <span style="float: left; border: 1px solid #ccc; padding: 2px;">Simulation Examples</span>	
<span style="border: 1px solid #ccc; padding: 2px;">Tag Name</span>	<span style="border: 1px solid #ccc; padding: 2px;">Address</span>
Reiniciar conteo PLC_1	M0.4
Parada remota PLC_1	M0.1
Marca de inicio de secuencia PLC_1	M0.2
Inicio remoto PLC_1	M0.0
Indicador de finalización de cuenta PLC_1	Q1.0
Indicador 3 PLC_1	Q0.4
Indicador 2 PLC_1	Q0.2
Indicador 1 PLC_1	Q0.0
Estado recibido del PLC_2	MW50
Estado de envío y recepción PLC_1	MW60
Desactivar comunicación PLC_1	M0.6
Conteo actual PLC_1	MW30
Comunicación activa PLC_1	M0.7
Ajuste de tiempo PLC_1	MD10
Ajuste de cuenta PLC_1	MW20
Activar comunicación PLC_1	M0.5

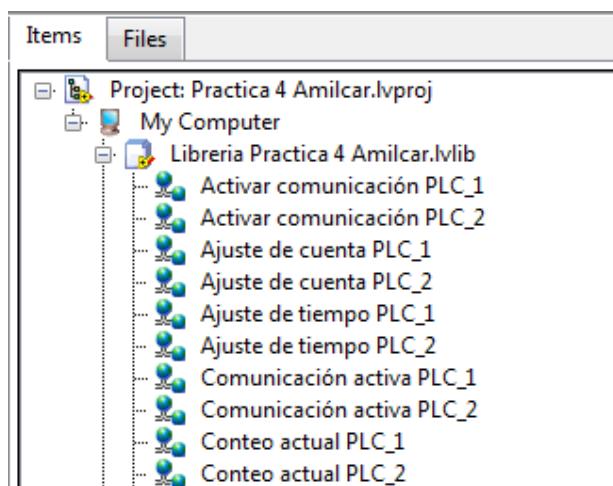
Figura 113. Tags creados en PLC\_1.

**PLC\_1:** Reiniciar conteo PLC\_2 (M0.1), Marca de inicio de secuencia PLC\_2 (M0.0), Indicador de finalización de cuenta PLC\_2 (Q1.0), Indicador 3 PLC\_2 (Q0.4), Indicador 2 PLC\_2 (Q0.2), Indicador 1 PLC\_2 (Q0.0), Estado de envío y recepción PLC\_2 (MW60), Desactivar comunicación PLC\_1 (M0.3), Conteo actual PLC\_2 (MW40), Comunicación activa PLC\_1 (M0.4), Ajuste de tiempo PLC\_2 (MD20), Ajuste de cuenta PLC\_1 (MW20), Estado recibido del PLC\_1 (MW10), Activar comunicación PLC\_2 (M0.2). Véase la **Figura 114** donde aparecen los tags agregados.

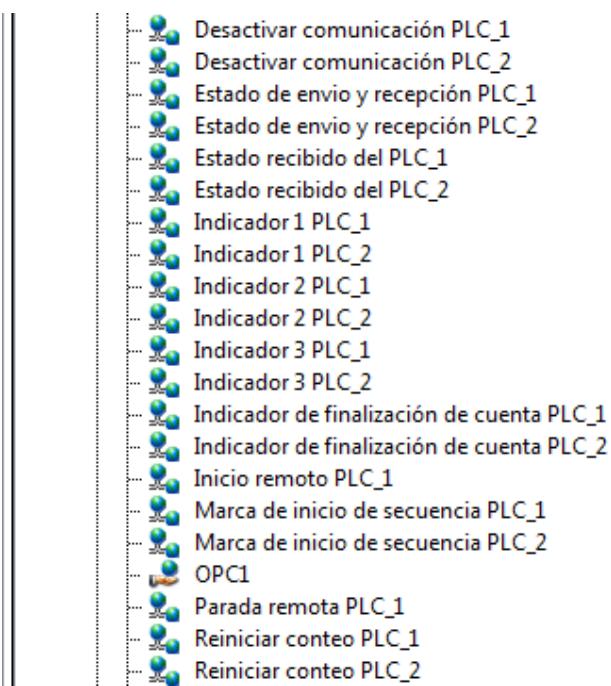


**Figura 114.** Tags creados en PLC\_2.

Configurado el servidor OPC, ejecute LabVIEW, en el cree un *Blank Project* y configure el *OPC Client* lo que generará la librería contenedora de las variables, luego agregue las variables en función de los *tags* creados previamente en el servidor OPC, ver **Figuras 115 y 116**.



**Figura 115.** Variables agregadas a la librería (1).



**Figura 116.** Variables agregadas a la librería (2).

Finalizada la creación de las distintas variables para lectura y escritura en la interfaz, cree la interfaz tal como el que se muestra en las **Figura 117 (Front Panel)**, **Figura 118 (Block Diagram 1)** y **Figura 119 (Block Diagram 2)** donde se implementarán *Push Buttons*, *Round Leds*, *Numeric Controls* y *Numeric Indicators*, *String Indicators* y *String Constants*.

Finalizado el diseño de la interfaz, ejecútela, en la **Figura 120** es mostrado el *Front Panel* en ejecución

**Nota:** La decoración es alternativa, en ella puede implementar elementos presentes en *Controls / Decorations*.

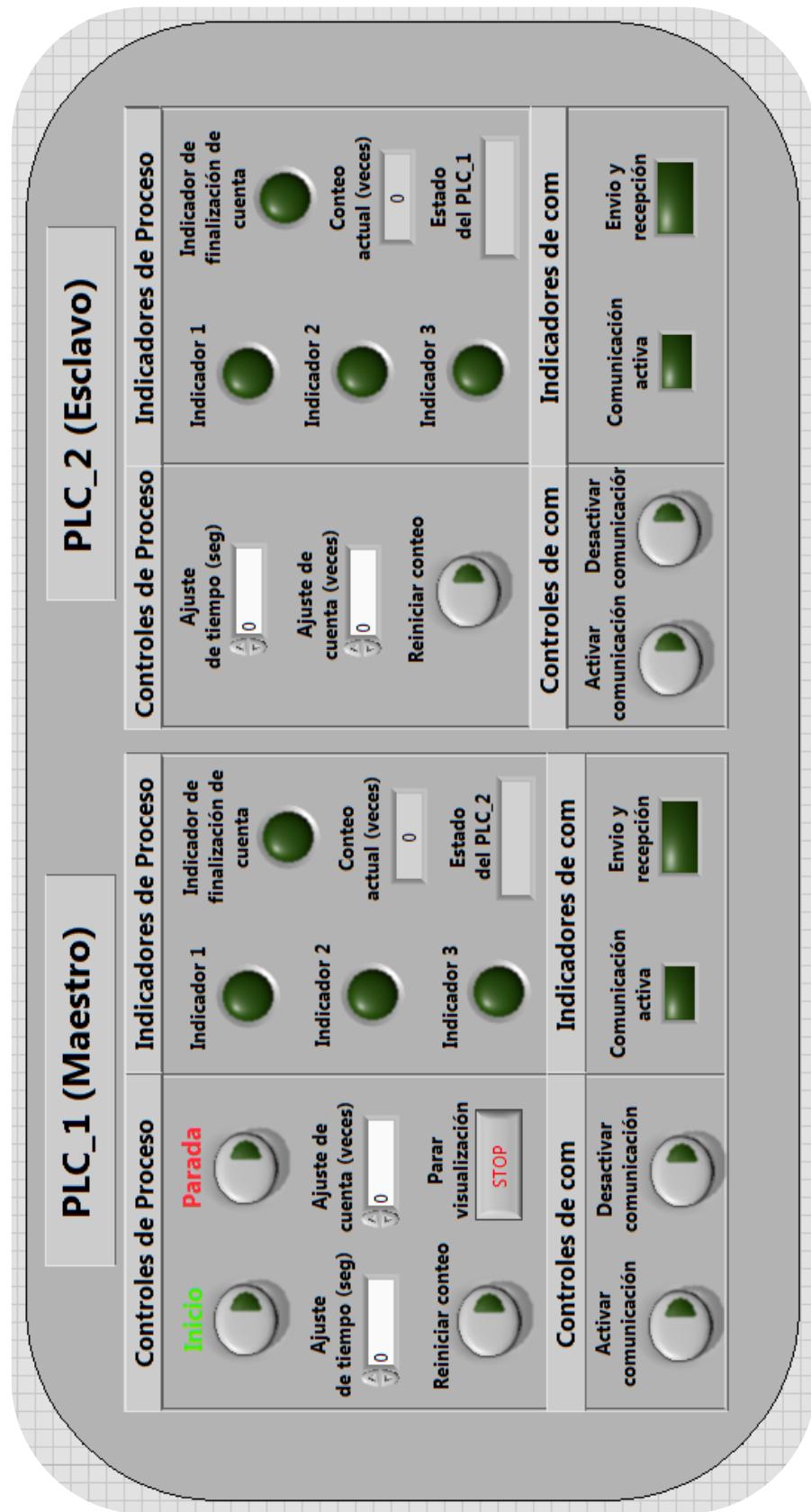


Figura 117. Interfaz para el control y supervisión del proceso (secuenciador doble) y comunicación entre los PLCs.

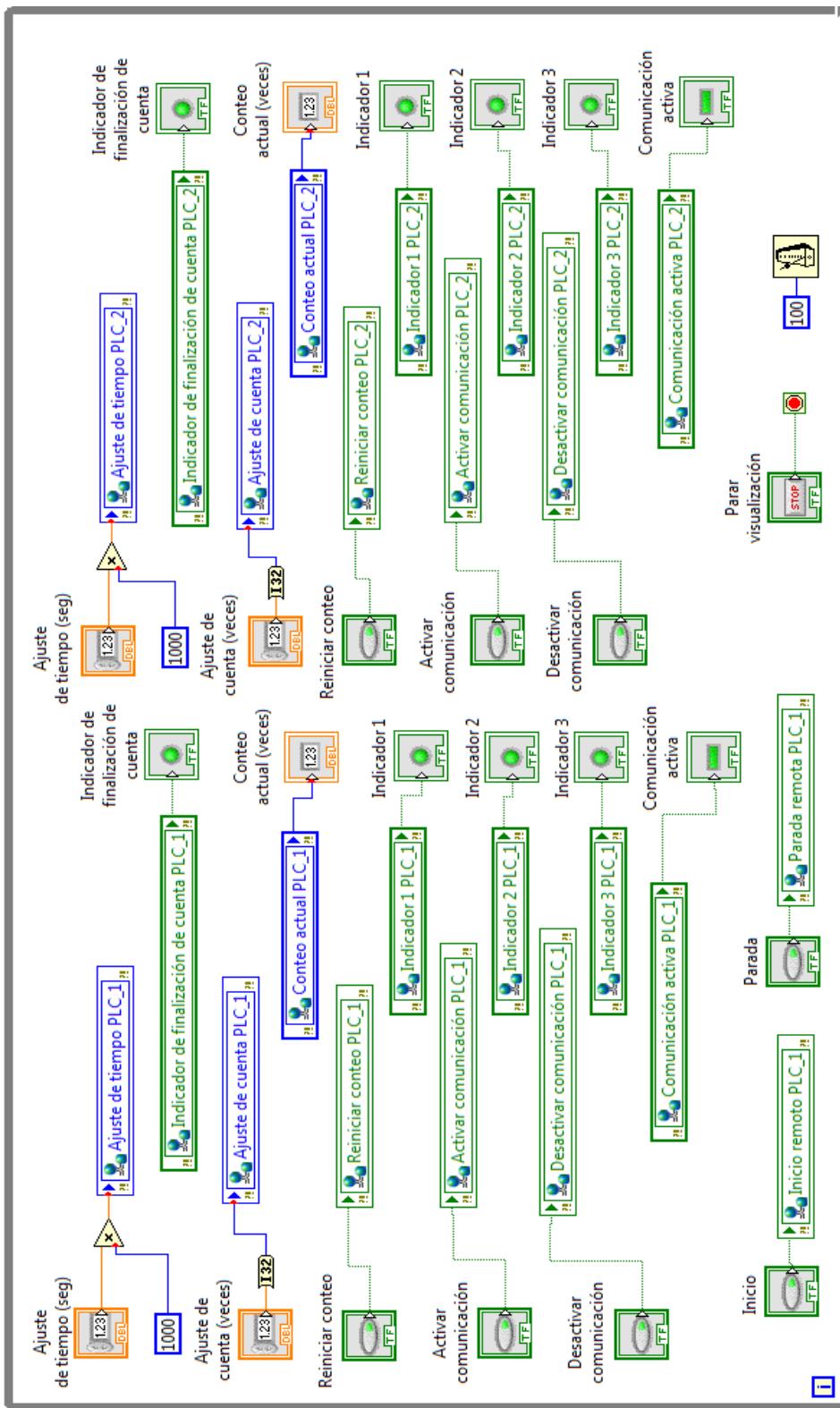


Figura 118. Block Diagram 1 de la interfaz para el control y supervisión del proceso (secuenciador doble) y comunicación entre los PLCs.

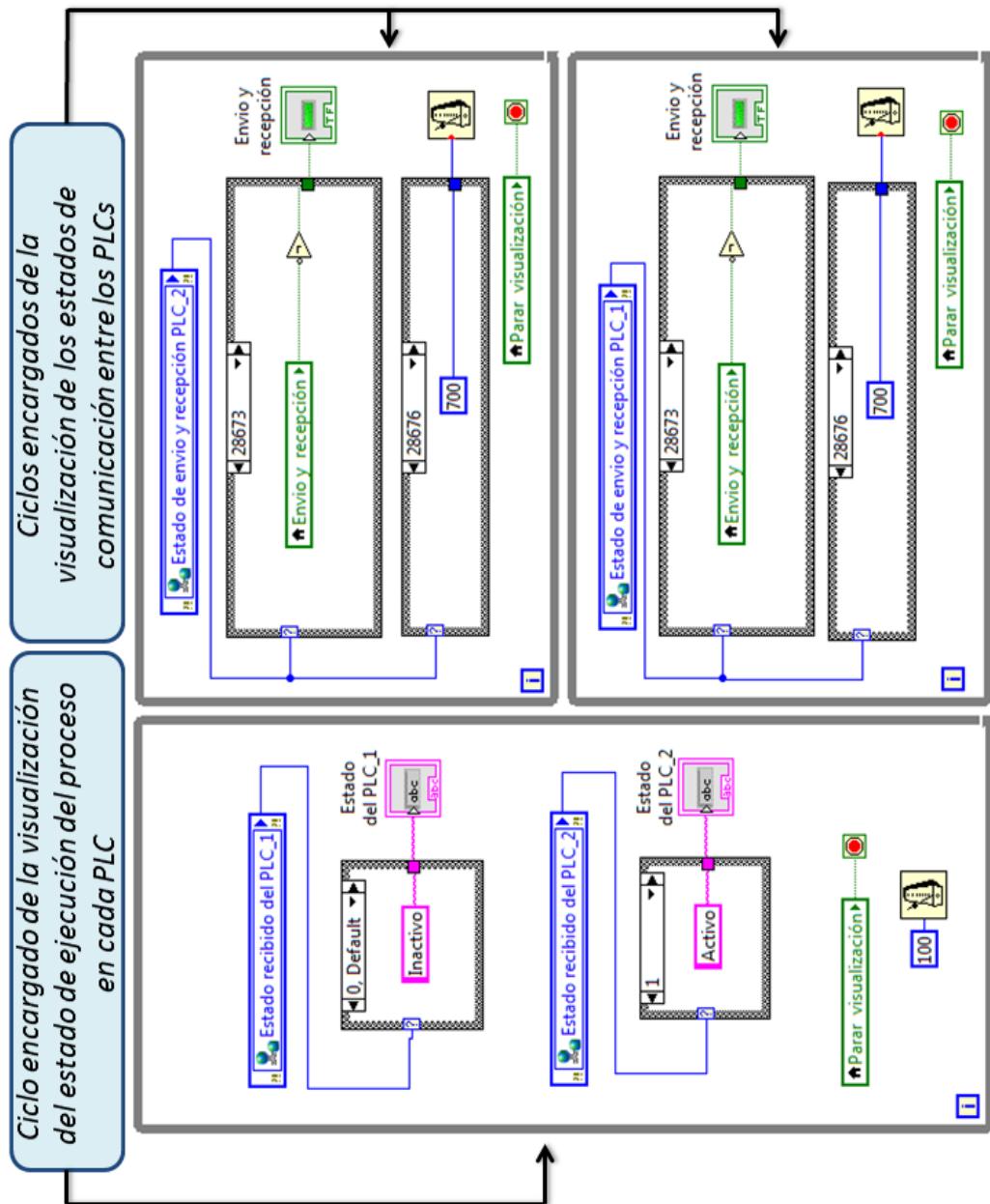


Figura 119. Block Diagram 2 de la interfaz para el control y supervisión del proceso (secuenciador doble) y comunicación entre los PLCs.

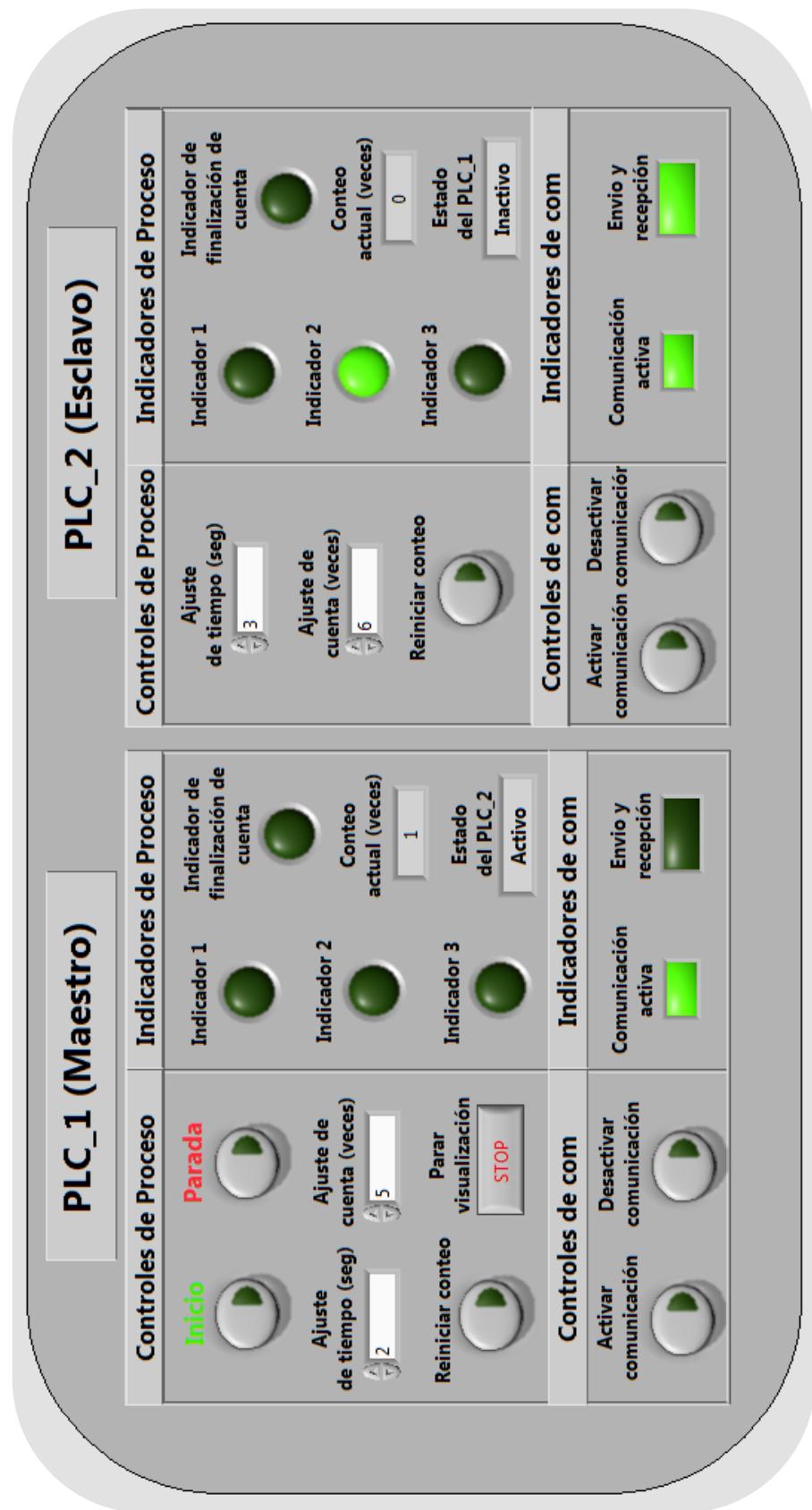


Figura 120. Interfaz para el control y supervisión del proceso (secuenciador doble) y comunicación entre los PLCs en ejecución.

## **PRÁCTICA N° 5**

*Integración de un PLC Siemens S7-1200 y un Allen Bradley ControlLogix.*

## ÍNDICE GENERAL

	Pág.
1. OBJETIVO GENERAL.....	1
2. LISTA DE MATERIALES Y EQUIPOS .....	1
3. FUNDAMENTACIÓN TEÓRICA.....	1
3.1. Modelo OSI .....	1
3.2. Modelo TCP/IP .....	3
3.3. PROFINET .....	5
3.4. Ethernet/IP .....	6
3.5. Compatibilidad entre PROFINET y Ethernet/IP .....	7
4. PROCEDIMIENTO .....	12
4.1. Configuración de la red .....	12
4.2. Comunicación bidireccional entre los PLCs controlada y supervisada desde una HMI .....	16
4.3. Diseño de un secuenciador doble controlado y supervisado mediante una HMI.....	23

## ÍNDICE DE FIGURAS

	Pág.
Figura 1. Modelo ISO/OSI.....	3
Figura 2. Modelo TCP/IP .....	5
Figura 3. PROFINET en el modelo TCP/IP.....	6
Figura 4. Ethernet/IP en el modelo TCP/IP.....	7
Figura 5. Comparación entre PROFINET y Ethernet/IP.....	8
Figura 6. Modelo para la comunicación entre un PLC Siemens SIMATIC S7-1200 y Allen Bradley ControlLogix.....	10
Figura 7. Capa de aplicación del Ordenador (Programadora) ( <i>Gateway</i> ).....	11
Figura 8. Elementos conectados en red (topología en estrella). ....	12
Figura 9. Configuración de la dirección IP del ordenador. ....	13
Figura 10. Configuración de la dirección IP del router. ....	14
Figura 11. Configuración de la dirección IP del PLC_S. ....	14
Figura 12. CPU y módulo ENBT/A dispuestos en <i>Controller Organizer / I/O Configuration</i> . ....	15
Figura 13. Elementos conectados y configurados en red. ....	16
Figura 14. Comunicación bidireccional entre los PLCs. ....	17
Figura 15. Variables creadas desde <i>Variables PLC</i> .....	18
Figura 16. Variables agregadas a <i>Tabla de observación_1</i> .....	18
Figura 17. Variables agregadas a <i>MainProgram</i> .....	19
Figura 18. Canal 1 con el PLC SIMATIC S7-1200 configurado y variables agregadas. .....	19
Figura 19. Canal 2 con el PLC AB ControlLogix configurado y variables agregadas. ....	19
Figura 20. Variables agregadas a la librería.....	20
Figura 21. Interfaz (VI) para el control y supervisión de la transferencia de datos....	20
Figura 22. <i>Block Diagram</i> de la interfaz para la transferencia de datos. ....	21
Figura 23. Interfaz (VI) para el control y supervisión de la transferencia de datos ejecutándose.....	22

Figura 24. Tabla de observación_1 del PLC SIMATIC S7-1200 en ejecución.....	22
Figura 25. MainProgram del PLC AB ControlLogix en ejecución.....	23
Figura 26. Esquema de funcionamiento del secuenciado doble (transmisión bidireccional).....	25
Figura 27. Segmento 1 (PLC_S).....	26
Figura 28. Segmento 2 (PLC_S).....	26
Figura 29. Segmento 3 (PLC_S).....	27
Figura 30. Segmento 3 (PLC_S).....	27
Figura 31. Segmento 5 (PLC_S).....	27
Figura 32. Segmento 6 (PLC_S).....	28
Figura 33. Segmento 7 (PLC_S).....	28
Figura 34. Segmento 8 (PLC_S).....	29
Figura 35. Segmento 9 (PLC_S).....	29
Figura 36. Segmento 10 (PLC_S).....	30
Figura 37. Segmento 11 (PLC_S).....	30
Figura 38. Segmento 11 (PLC_S).....	30
Figura 39. Segmento 0 (PLC_AB).....	31
Figura 40. Segmento 1 (PLC_AB).....	31
Figura 41. Segmento 2 (PLC_AB).....	31
Figura 42. Segmento 3 (PLC_AB).....	32
Figura 43. Segmento 4 (PLC_AB).....	32
Figura 44. Segmento 5 (PLC_AB).....	32
Figura 45. Segmento 6 (PLC_AB).....	32
Figura 46. Segmento 7 (PLC_AB).....	33
Figura 47. Segmento 8 (PLC_AB).....	33
Figura 48. Segmento 9 (PLC_AB).....	33
Figura 49. Segmento 10 (PLC_AB).....	33
Figura 50. Segmento 11 (PLC_AB).....	34
Figura 51. Segmento 11 (PLC_AB).....	34
Figura 52. Canal 1 con el PLC SIMATIC S7-1200 configurado y variables agregadas.	
	35

Figura 53. Canal 2 con el PLC AB ControlLogix configurado y variables agregadas en <i>Prgm_MainProgram</i> .....	35
Figura 54. Canal 2 con el PLC AB ControlLogix configurado y variables agregadas en <i>Prgm_MainProgram / Contador_1</i> .....	36
Figura 55. Canal 2 con el PLC AB ControlLogix configurado y variables agregadas en <i>Prgm_MainProgram / Temporizador_1</i> .....	36
Figura 56. Variables agregadas a la librería.....	37
Figura 57. Interfaz ( <i>Front Panel</i> ) para el control y supervisión del proceso (secuenciador doble).....	38
Figura 58. <i>Block Diagram</i> de la interfaz para la comunicación, control y supervisión del proceso entre los PLCs (secuenciador doble).....	40
Figura 59. Interfaz ( <i>Front Panel</i> ) para el control y supervisión del proceso (secuenciador doble) en ejecución.....	41
Figura 60. Verificación del correcto monitoreo de los <i>tags</i> del PLC_S .....	42
Figura 61. Verificación del correcto monitoreo de los <i>tags</i> del PLC_AB .....	42

## 1. OBJETIVO GENERAL

Comunicar, supervisar y controlar los PLC Siemens SIMATIC S7-1200 y Allen Bradley ControlLogix.

## 2. LISTA DE MATERIALES Y EQUIPOS

- Ordenador con los programa TIA Portal, National Instruments OPC Server y LabVIEW con los módulos Datalogging and Supervisory Control (DSC) y Real Time integrados.
- PLC Siemens SIMATIC S7-1200 1214C AC/DC/RLY.
- PLC Allen Bradley ControlLogix 5555 1756-L55.
- Cable UTP RJ45/RJ45.
- Router.

## 3. FUNDAMENTACIÓN TEÓRICA

### 3.1. Modelo OSI

El modelo OSI del inglés *Open Systems Interconnection* que significa modelo de interconexión de sistemas abiertos, se desarrolló por la Organización Internacional de Estandarización ISO (por sus siglas en inglés International Organization for

*Standarization)* como una arquitectura para comunicaciones entre computadores, con el objetivo de ser el marco de referencia en el desarrollo de protocolos estándares.

El modelo OSI consta de siete capas:

- **Capa de aplicación:** proporciona el acceso al entorno OSI para los usuarios, también proporciona servicios de información distribuida.
- **Capa de presentación:** proporciona a los procesos de aplicación independencia respecto a las diferencias en la representación de los datos (sintaxis).
- **Capa de sesión:** proporciona el control de la comunicación entre las aplicaciones; establece, gestiona y cierra las conexiones (sesiones) entre las aplicaciones cooperadas.
- **Capa de transporte:** proporciona seguridad, transferencia transparente de datos entre los puntos finales; proporciona además procedimientos de recuperación de errores y control de flujo origen-destino.
- **Capa de red:** proporciona independencia a los niveles superiores respecto a las técnicas de commutación y de transmisión utilizadas para conectar los sistemas; es responsable del establecimiento, mantenimiento y cierre de las conexiones.
- **Capa de enlace de datos:** proporciona un servicio de transferencia de datos seguro a través del enlace físico; envía bloques de datos (tramas) llevando a cabo la sincronización, el control de errores y de flujo necesarios

- **Capa física:** se encarga de la transmisión de cadenas de bits no estructurados sobre el medio físico; está relacionada con las características mecánicas, eléctricas, funcionales y de procedimiento para acceder al medio físico.

En la **Figura 1** se puede ver gráficamente el modelo OSI.



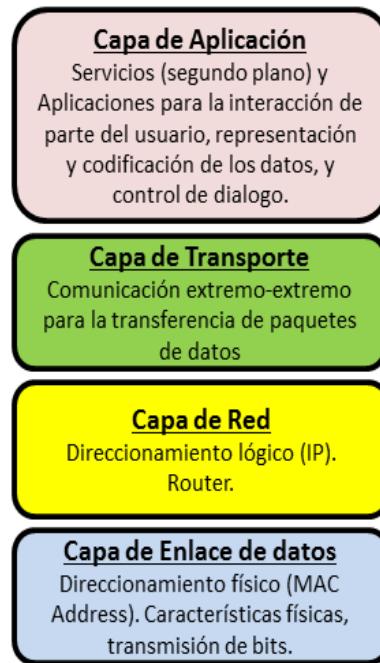
**Figura 1.** Modelo ISO/OSI.

### 3.2. Modelo TCP/IP

El modelo TCP/IP está compuesto por cuatro capas o niveles, cada nivel se encarga de determinados aspectos de la comunicación y a su vez brinda un servicio específico a la capa superior. A continuación se describen una por una las capas:

- **Capa de Aplicación:** proporciona la comunicación entre procesos o aplicaciones de computadores separados. En este nivel se definen los protocolos de aplicación TCP/IP y como se conectan los programas de host a los servicios de nivel de transporte para utilizarlos en la red. Protocolos: HTTP, Telnet, FTP, TFTP, SNMP, DNS, SMTP, X Windows y otros protocolos de aplicación.
- **Capa de transporte:** proporciona un servicio de transferencia de datos extremo a extremo. Esta capa puede incluir mecanismos de seguridad. Oculta los detalles de la red, o redes subyacentes, a la capa de aplicación. Este nivel permite administrar las sesiones de comunicación entre equipos host. Define el nivel de servicio y estado de la conexión utilizada por el transportador de datos. Protocolos: TCP, UDP, RTP.
- **Capa de internet (Red):** esta capa está relacionada con el encaminamiento de los datos del computador origen al destino a través de una o más redes conectadas por dispositivos de encaminamiento. Este nivel se encarga de empaquetar los datos en datagrama IP, que contienen información de las direcciones de origen y destino utilizadas para reenviar los diagramas entre hosts y a través de redes. Realiza el enrutamiento de datagramas IP. Protocolos: IP,I CMP, ARP, RARP.
- **Capa de enlace de datos:** maneja todos los aspectos que un paquete IP requiere para efectuar un enlace físico real con los medios de la red. Esta capa incluye los detalles de la tecnología LAN y WAN y todos los detalles de las capas físicas y de enlace de datos del modelo OSI.

En la **Figura 2** se puede ver gráficamente el modelo TCP/IP.



**Figura 2.** Modelo TCP/IP.

### 3.3. PROFINET

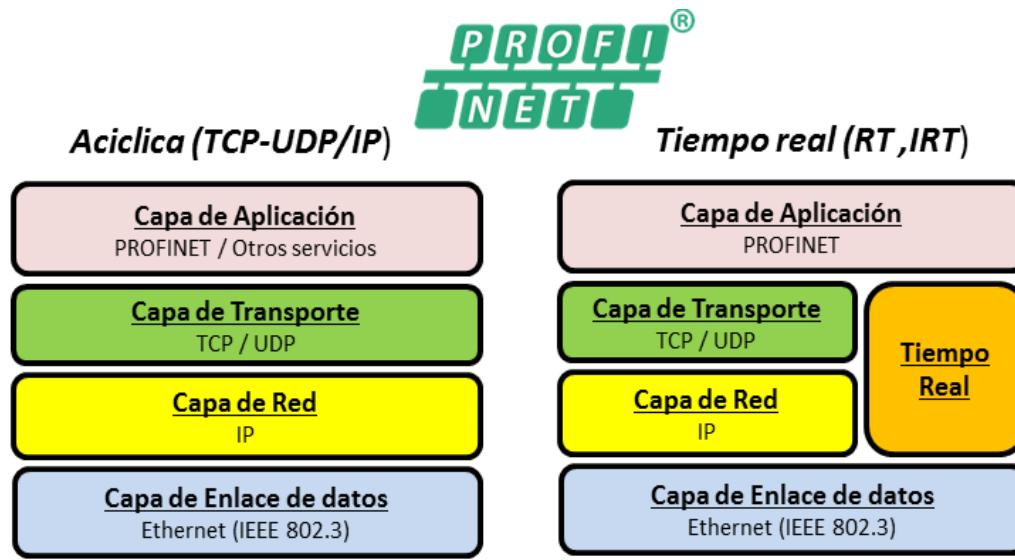
Es un concepto moderno para los estándares de automatización distribuidos, y se basa en protocolos Ethernet, UDP, TCP e IP. Integra los actuales sistemas de bus de campo (PROFIBUS en particular). Además, representa una aportación clave, consiste en que las partes existentes de un sistema de versión anterior se pueden incorporar, dando la seguridad al usuario de poder invertir en este nuevo sistema sin necesidad de cambiar todo el grupo de trabajo.

PROFINET se caracteriza por tener los siguientes modos de transmisión:

- Transmisión acíclica sin prioridad de tiempo (TCP/IP, NRT) alrededor de 100ms (Control y HMI).
- Transmisión cíclica con prioridad de tiempo (RT) de 5-10ms.

- Transmisión isócrona en tiempo real (IRT) < 1 ms.

En la **Figura 3** aparece descrito la pila en el modelo TCP/IP de PROFINET.



**Figura 3.** PROFINET en el modelo TCP/IP.

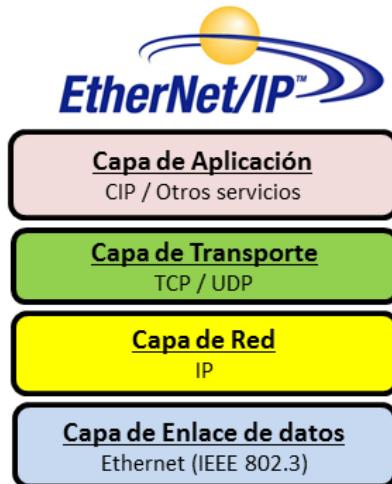
### 3.4. Ethernet/IP

Ethernet/IP es un protocolo de red abierta establecido y mantenido por ODVA (Empresa diseñadora del protocolo Ethernet/IP). Se introdujo en 2001 y hoy es la más desarrollada, probada y completa solución de red industrial Ethernet para la automatización de fabricación, con un rápido crecimiento ya que los usuarios tratan de aprovechar las ventajas de las tecnologías abiertas de internet. Ethernet/IP es un miembro de una familia de las redes que implementa el Protocolo Industrial Común, CIP (del inglés *Common Industrial Protocol*) en sus capas inferiores.

En la capa de aplicación, el CIP define un conjunto de objetos de aplicación, perfiles de dispositivos que definen las interfaces comunes y comportamientos. CIP, permite comunicación de extremo a extremo entre los dispositivos en las diferentes redes CIP.

Ethernet/IP brinda servicios de comunicación a Ethernet y TCP/IP, permitiendo a múltiples proveedores la interoperabilidad entre los dispositivos de Ethernet, así como con las otras redes de CIP. Este protocolo diseñado para aplicaciones de automatización abarca un amplio conjunto de servicios de comunicaciones para aplicaciones de automatización, control, seguridad, sincronización, movimiento, configuración e información.

En la **Figura 4** aparece descrito la pila en el modelo TCP/IP de Ethernet/IP.



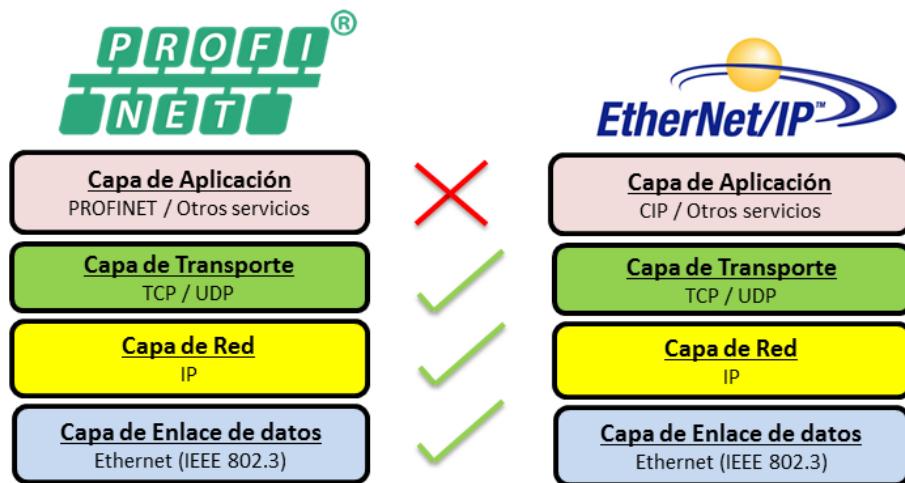
**Figura 4.** Ethernet/IP en el modelo TCP/IP.

### 3.5. Compatibilidad entre PROFINET y Ethernet/IP

Los protocolos de comunicación PROFINET (acíclica) y Ethernet/IP presentan compatibilidad en los tres primeros niveles o capas del modelo TCP/IP, sin embargo, en la cuarta capa (Aplicación) cada uno se encuentra definido para trabajar con servicios y aplicaciones distintas, ver **Figura 5**.

Esta discrepancia entre los protocolos en la capa de aplicación del modelo de referencia TCP/IP imposibilita la comunicación entre un dispositivo que trabaje con

PROFINET y otro con Ethernet/IP debido a: la incompatibilidad en la representación y codificación de los datos, en los modos de diálogos y en aplicaciones que permitan al usuario interactuar con el nivel de aplicación, para de este modo acceder a los niveles inferiores con el objetivo de transmitir y recibir exitosamente la información.



**Figura 5.** Comparación entre PROFINET y EtherNet/IP.

Ya que el objetivo principal de esta práctica es comunicar un PLC Siemens SIMATIC S7-1200 (PROFINET) con un Allen Bradley ControlLogix (Ethernet/IP) se implementará un ordenador como *Gateway* (Puerta de enlace) para transmitir la data de un PLC a otro de forma bidireccional.

El *Gateway* será un ordenador que recolectará los datos (representación y decodificación) desde cada PLC mediante la implementación de un servidor OPC (NI OPC Server) que actuará como protocolo en la capa de aplicación.

Recolectados los datos, serán usados por una aplicación que se encargará de establecer la lógica necesaria para la transmisión de los datos, esta aplicación será LabVIEW, previamente usado para el diseño de HMIs, por lo cual el estudiante se encuentra familiarizado con él.

En la **Figura 6** es mostrado un esquema para la comunicación entre los controladores.

En la **Figura 7** aparece en detalle el funcionamiento de la capa de aplicación del *Gateway* (ordenador).

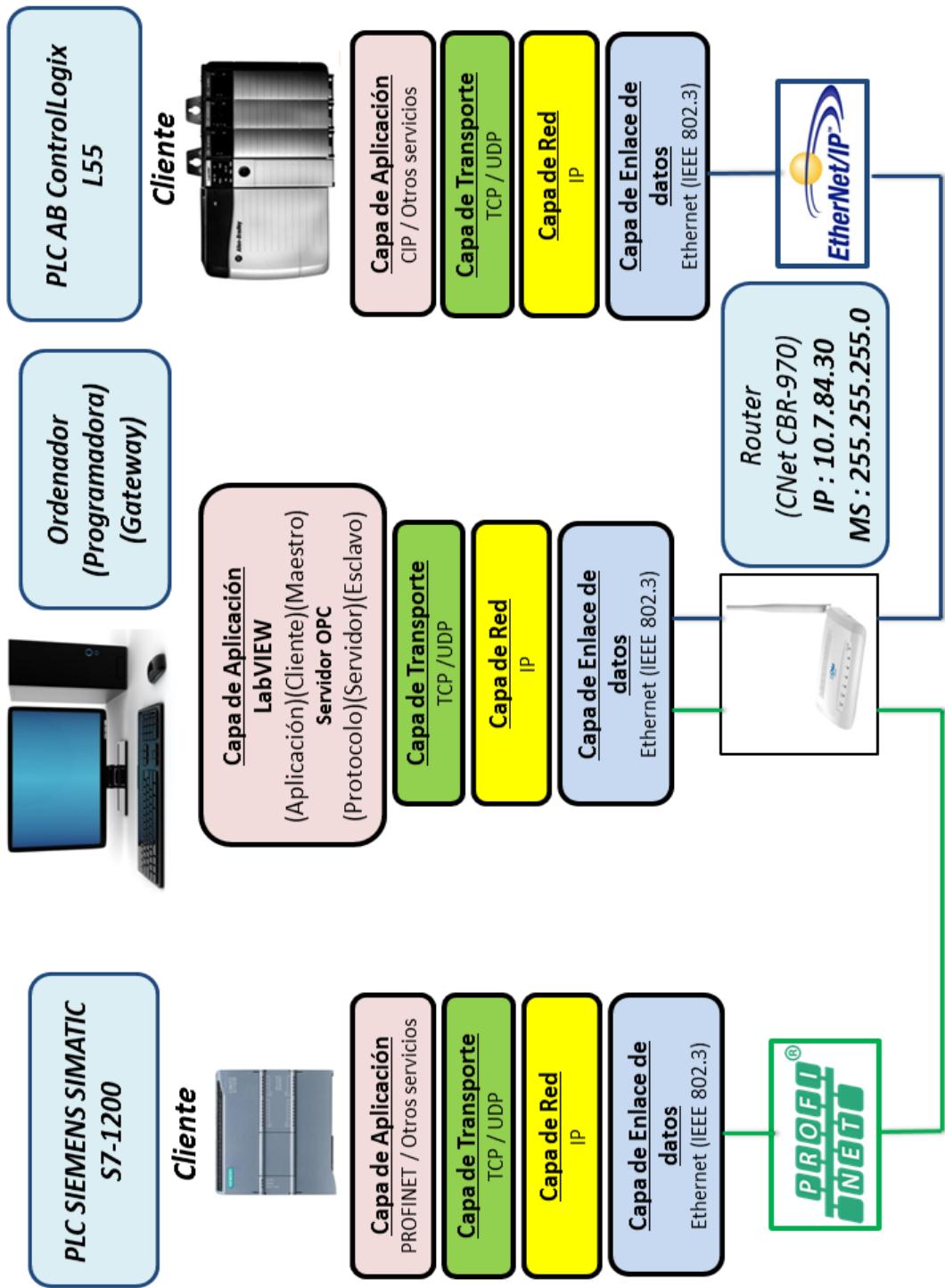


Figura 6. Modelo para la comunicación entre un PLC Siemens SIMATIC S7-1200 y Allen Bradley ControlLogix.

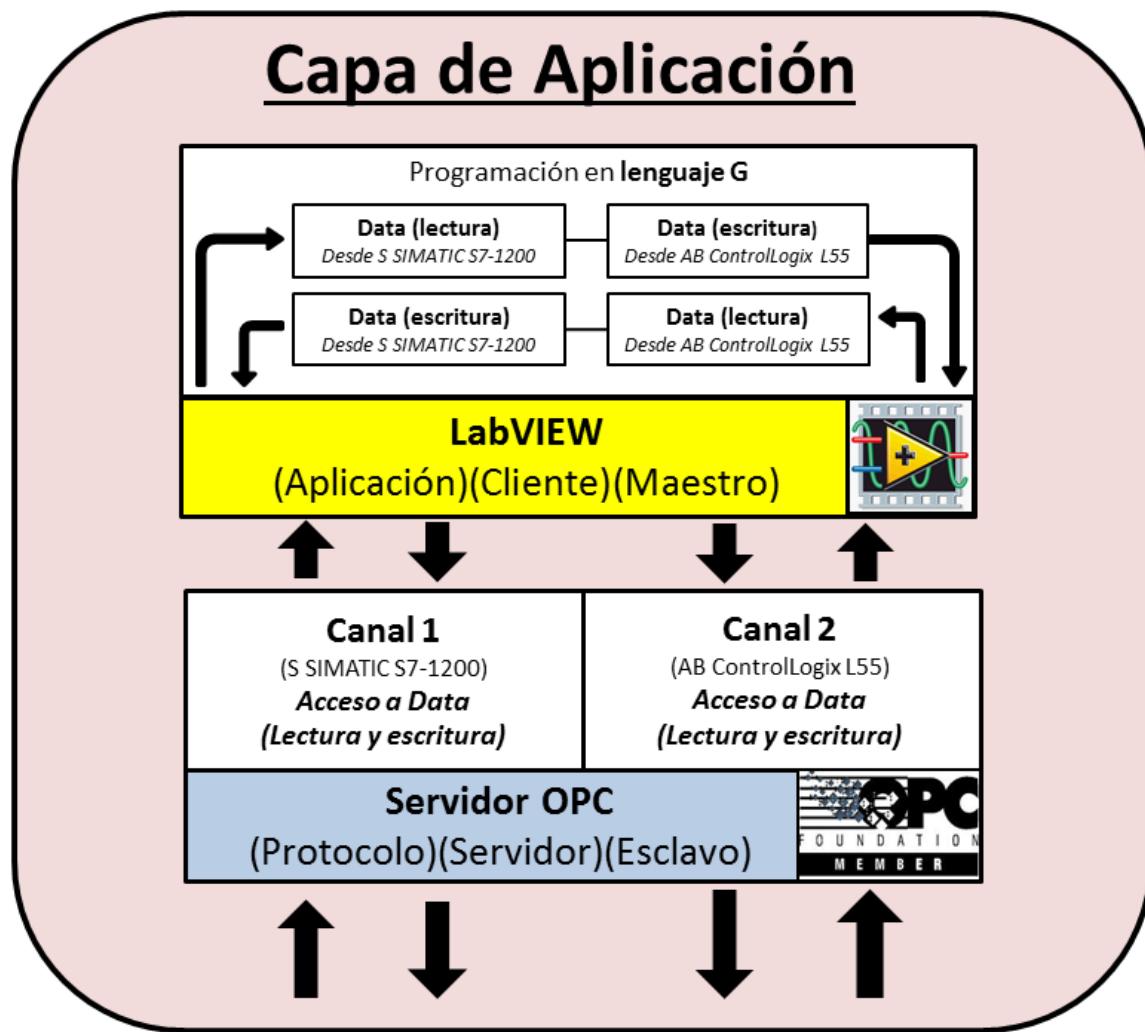
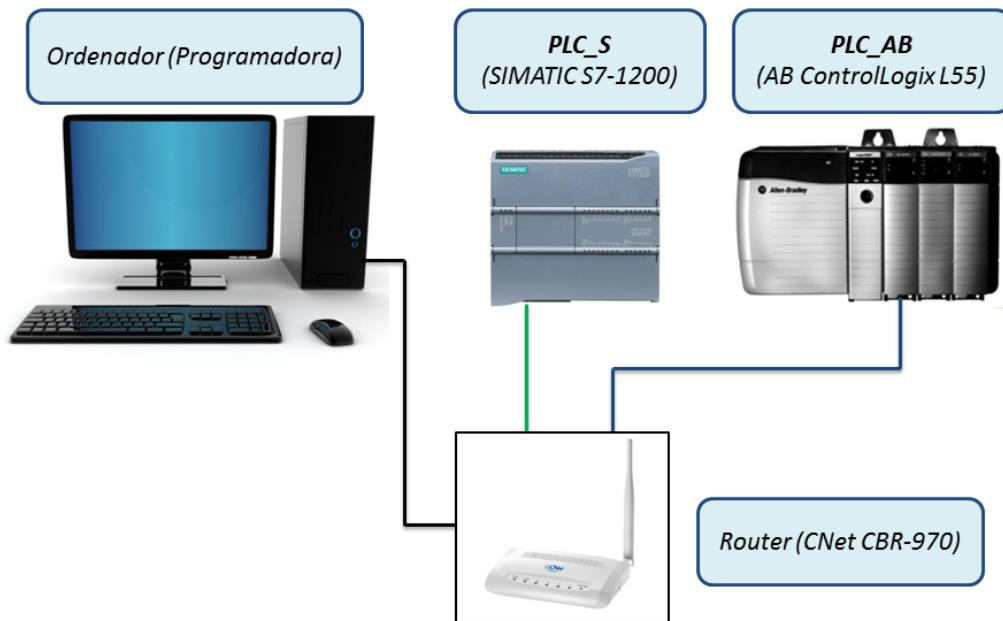


Figura 7. Capa de aplicación del Ordenador (Programadora) (Gateway).

## 4. PROCEDIMIENTO

### 4.1. Configuración de la red

La red sobre la que se trabajará en el desarrollo de esta práctica estará compuesta de cuatro (4) elementos: un (1) PLC SIMATIC S7-1200, un (1) PLC Allen Bradley ControlLogix CPU L55, un (1) ordenador con tarjeta de red integrada y un enrutador o encaminador (*router* CNet CBR-970). La red será definida por una *topología en estrella* debido al uso del enrutador, en la **Figura 8** se observa el conexionado de los distintos elementos de la red.

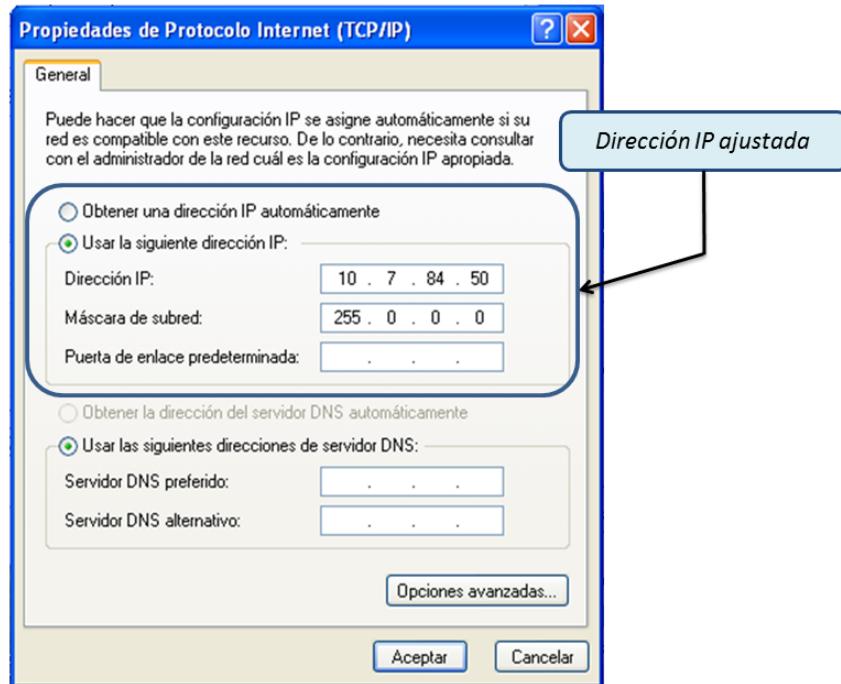


**Figura 8.** Elementos conectados en red (topología en estrella).

Conocida la topología y el conexionado de los distintos elementos, se configurará uno por uno para el correcto funcionamiento de la red.

- **Configuración del ordenador (programadora):** acceda a las configuraciones de su tarjeta de red y ajuste en *Protocolo de Internet (TCP/IP)* la dirección IP en

**10.7.84.50**, en la **Figura 9** aparece este procedimiento en entorno Microsoft Windows XP.



**Figura 9.** Configuración de la dirección IP del ordenador.

- **Configuración del router CNet CBR-970:** en este caso particular, la red será unificada mediante un router de la marca CNet modelo CBR-970, el procedimiento de asignación de la dirección IP en este router es similar al de cualquier otro (marca y modelo).

Remítase al panel de configuraciones del router insertando la dirección IP por defecto del router en algún navegador web (vía http), posteriormente acceda a las configuraciones de IP de área local (LAN), en este caso *IP Config / LAN* y ajuste la dirección en **10.7.84.30**, en la **Figura 10** puede evidenciarse parte de este procedimiento.

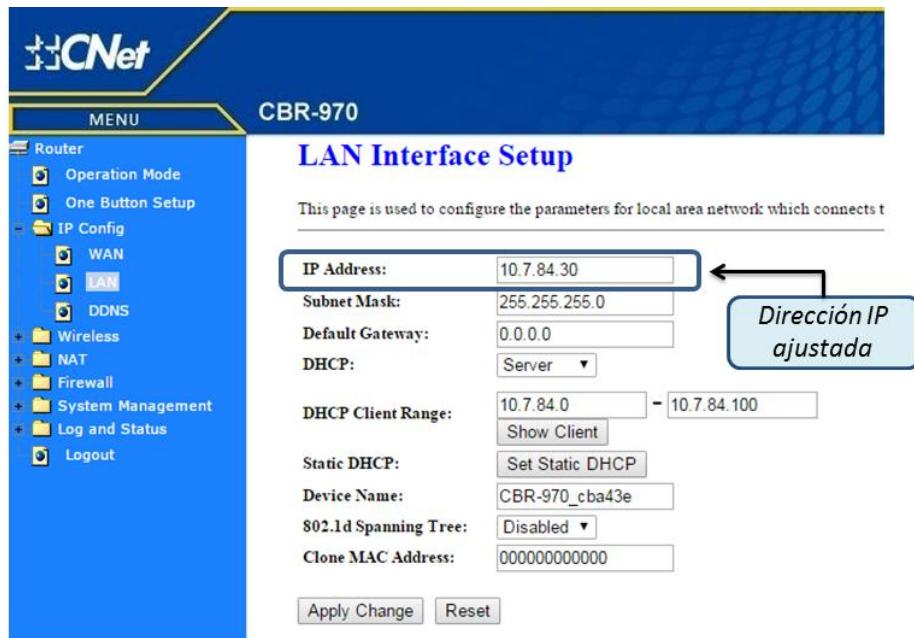


Figura 10. Configuración de la dirección IP del router.

- Configuración del PLC SIMATIC S7-1200:** desde el TIA PORTAL cree un proyecto con un nuevo dispositivo, una vez detectado el PLC\_S (PLC\_1) independientemente de la dirección IP que posea (pero dentro de la red), acceda a las *Propiedades / General / Interfaz PROFINET / Protocolo IP*, en ella ajuste la dirección IP en 10.7.84.40, ver Figura 11.

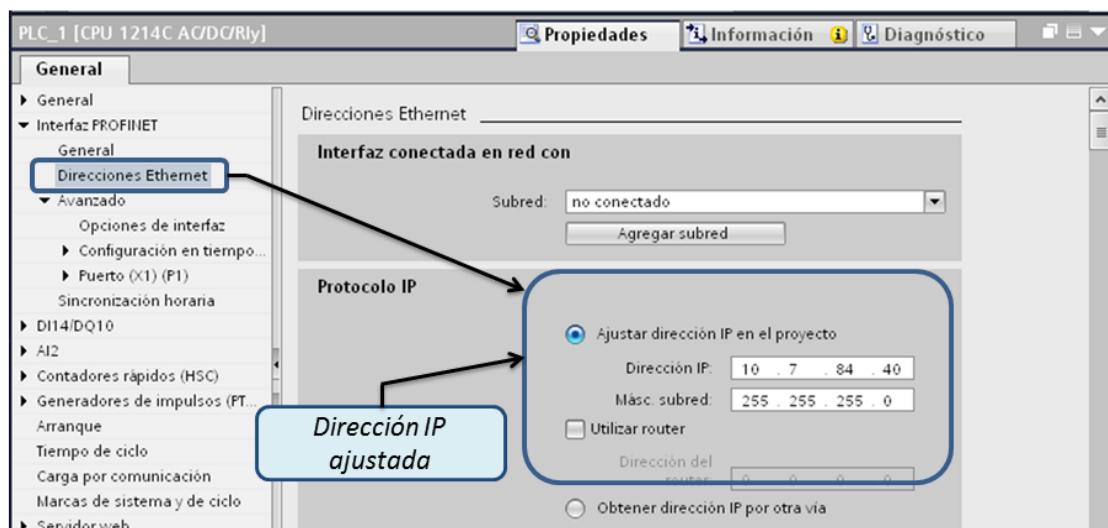
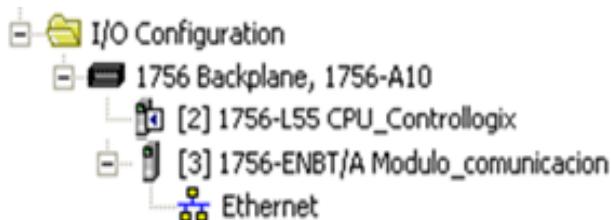


Figura 11. Configuración de la dirección IP del PLC\_S.

- **Configuración del AB ControlLogix L55:** para realizar algún cambio en el direccionamiento IP del módulo de comunicaciones Ethernet/IP 1756-ENBT/A, acceda a RSLogix con el módulo y CPU previamente detectadas en *Controller Organizer / I/O Configuration*, ver **Figura 12**.



**Figura 12.** CPU y módulo ENBT/A dispuestos en *Controller Organizer / I/O Configuration*.

Posteriormente, acceda a las propiedades del módulo ENBT/A y ajuste la dirección IP en **10.7.84.25** tanto para offline como online:

**Offline** desde *Module Properties / General / Address / Host Name*.

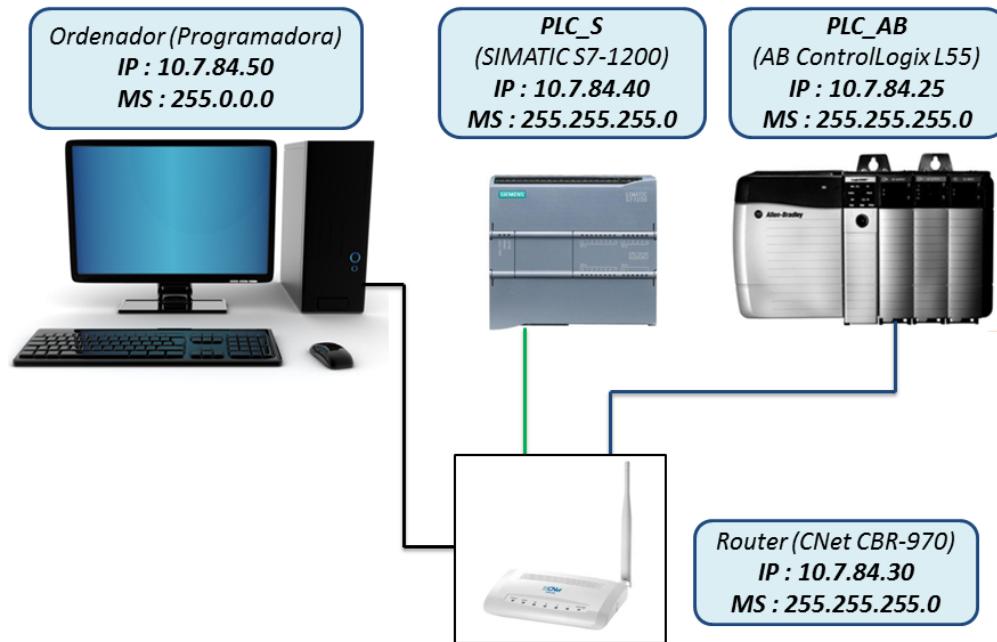
**Online** desde *Port Configuration / IP Address*.

Luego diríjase a *Who Active* y pulse *Clear Project Path*, espere unos segundos y se detectará el módulo con la nueva dirección IP, por ultimo pulse *Set Project Path*.

Pulse *Go online* para verificar el correcto funcionamiento de los cambios.

Finalizada la configuración agregue los respectivos módulos de entradas y salida digitales.

Terminada la configuración del PLC AB ControlLogix, ha concluido con los ajustes necesarios para el correcto funcionamiento de la red, en la **Figura 13** se visualiza la red con todos sus equipos integrados.



**Figura 13.** Elementos conectados y configurados en red.

#### 4.2. Comunicación bidireccional entre los PLCs controlada y supervisada desde una HMI

Se desea transmitir y recibir “un dato” desde cada PLC, ver **Figura 14**.

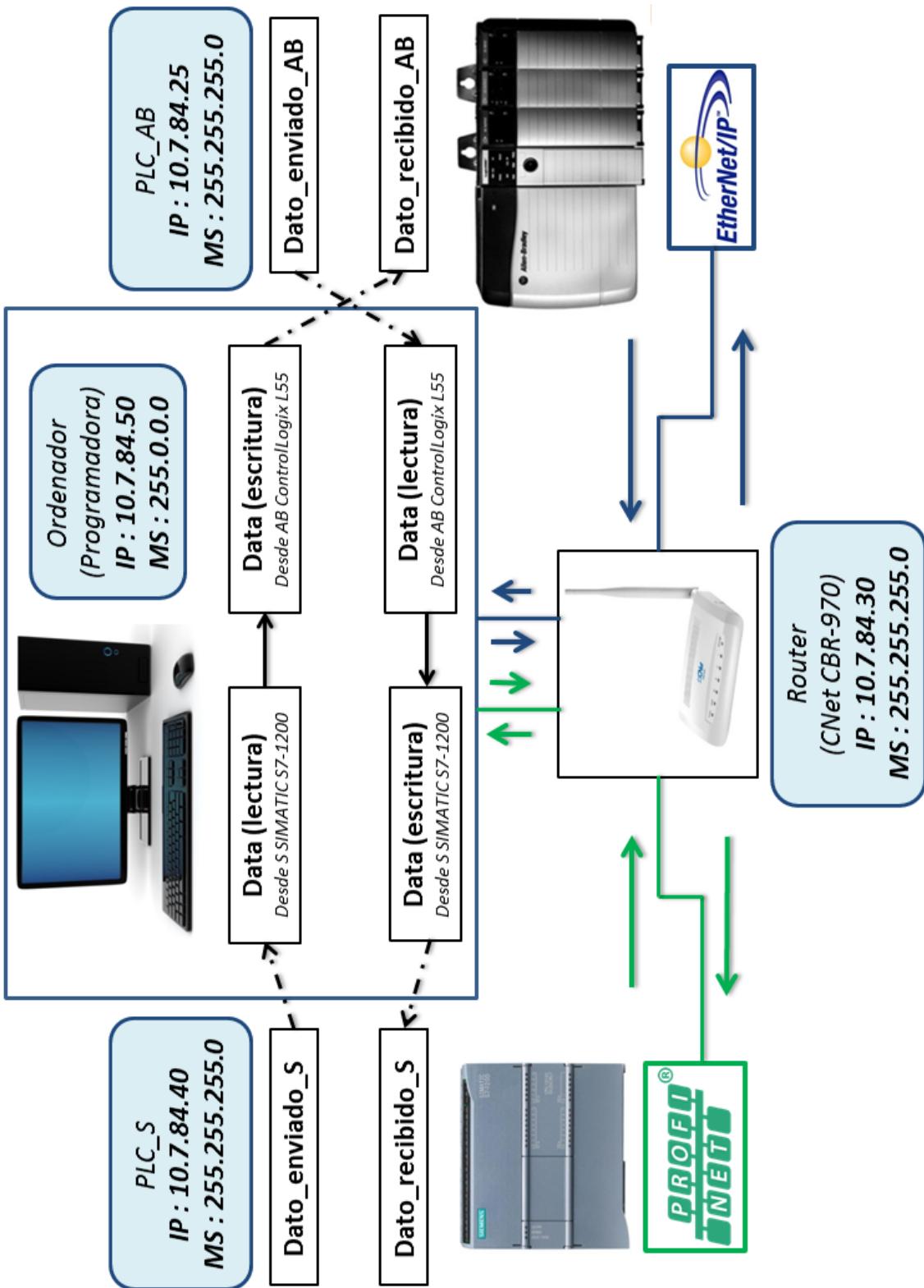


Figura 14. Comunicación bidireccional entre los PLCs.

En primer lugar, desde TIA PORTAL acceda a *Árbol del proyecto / Dispositivos / PLC\_1 / Variables PLC / Mostrar todas las variables*, dentro de la tabla cree dos (2) variables enteras con un tamaño de memoria de una palabra (*Word*, 16 bits) y llámelas como *Dato\_enviado\_S* y *Dato\_recibido\_S*, véase la Figura 15.

Práctica 5 > PLC_1 [CPU 1214C AC/DC/Rly] > Variables PLC								
	Nombre	Tabla de variables	Tipo de datos	Dirección	Rema...	Visibl...	Acces...	Co...
1	Dato_enviado_S	Tabla de variables ...	Int	%MW10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	Dato_recibido_S	Tabla de variables ...	Int	%MW12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	<Agregar>				<input type="button" value=""/>			

Figura 15. Variables creadas desde *Variables PLC*.

Luego, acceda a *Árbol del proyecto / Dispositivos / PLC\_1 / Tablas de observación / Agregar nueva tabla de observación*, hecho esto se agregará una nueva tabla de observación, dentro de ella agregue las variables creadas previamente en el procedimiento anterior (*Dato\_enviado\_S* y *Dato\_recibido\_S*), ver Figura 16.

Práctica 5 > PLC_1 [CPU 1214C AC/DC/Rly] > Tablas de observación > Tabla de observación_1							
i	Nombre	Dirección	Formato visualiz...	Valor de observa...	Valor de forzado	...	
1	"Dato_enviado_S"	%MW10	DEC	0	<input type="checkbox"/>		
2	"Dato_recibido_S"	%MW12	DEC	0	<input type="checkbox"/>		
3	<Agregar>				<input type="button" value=""/>		

Figura 16. Variables agregadas a *Tabla de observación\_1*.

Ahora vaya al **RSLogix 5000** y desde *Controller Organizer / MainProgram / Edit Tags*, agregue dos nuevos *tags* del tipo DINT con los nombres de *Dato\_enviado\_AB* y *Dato\_recibido\_AB*, ver Figura 17.

Scope: MainProgram					
Name	Value	Force Mask	Style	Data Type	
+ Dato_enviado_AB	0		Decimal	DINT	
+ Dato_recibido_AB	0		Decimal	DINT	

Figura 17. Variables agregadas a MainProgram.

Finalizada la creación de las variables en cada PLC, ejecute el NI OPC server, en el cree y configure dos canales cada uno con su respectivo equipo y variables asociadas, ver **Figuras 18 y 19**.

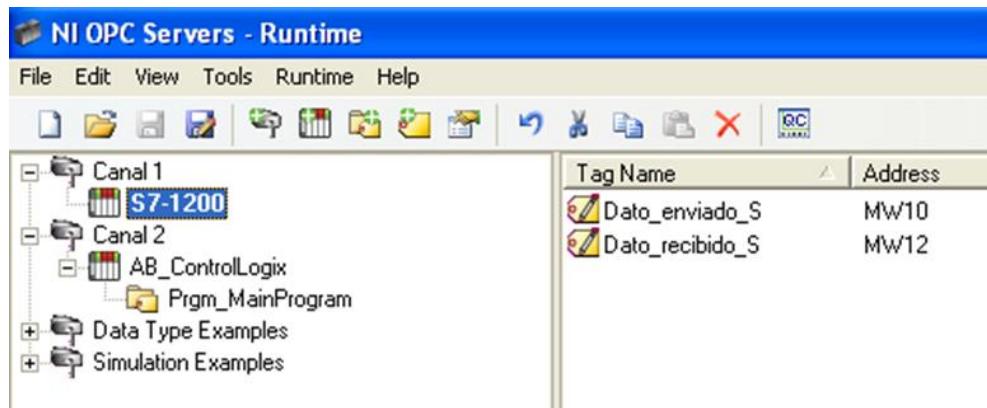


Figura 18. Canal 1 con el PLC SIMATIC S7-1200 configurado y variables agregadas.

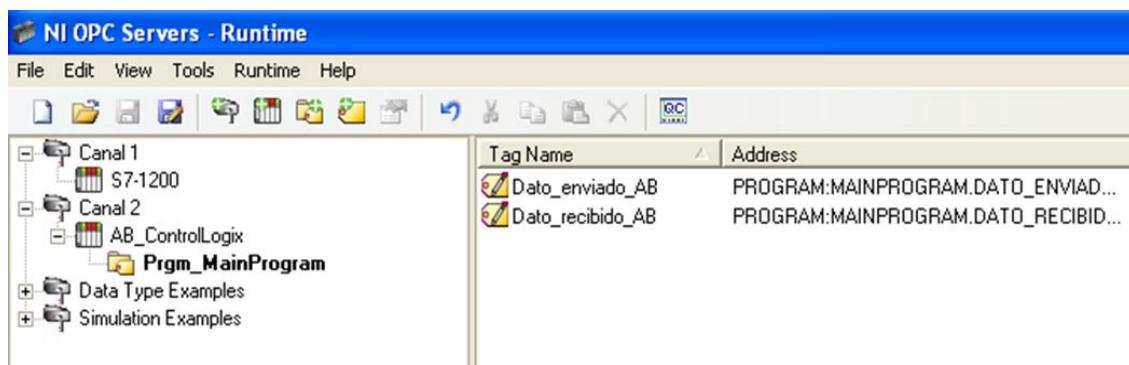
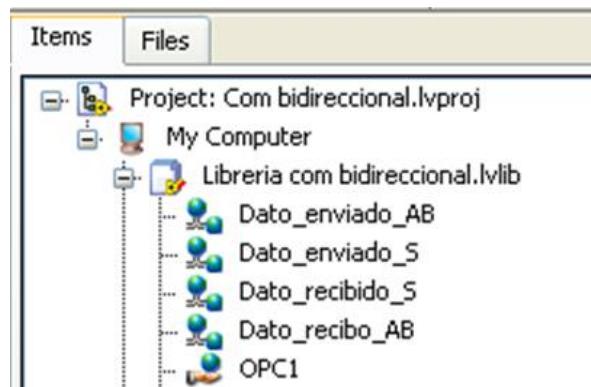


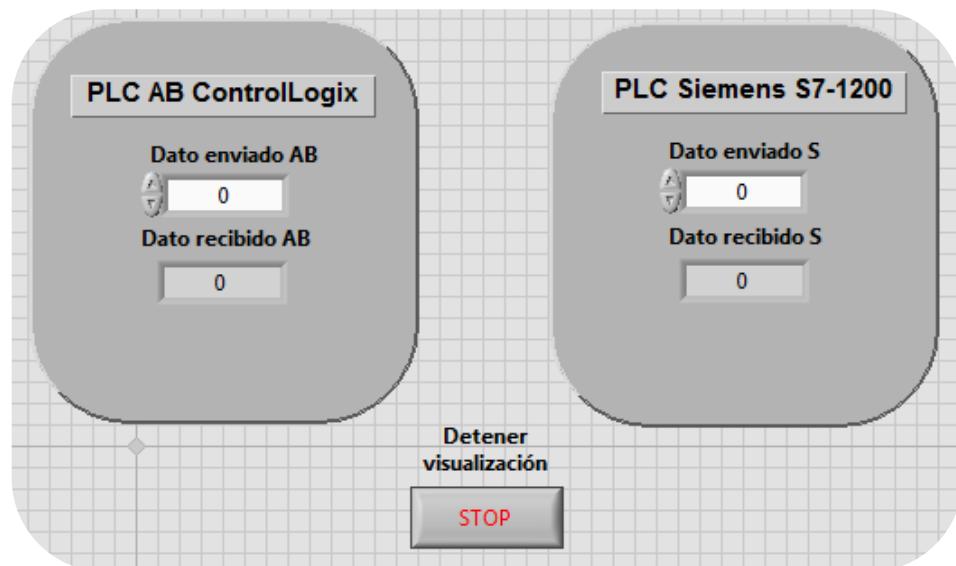
Figura 19. Canal 2 con el PLC AB ControlLogix configurado y variables agregadas.

Configurado el servidor OPC ejecute LabVIEW, en este último cree un *Blank Project*, luego configure el *OPC Client*, de tal manera se creará la librería contenedora de las variables, luego agregue las variables en función de los *tags* creados previamente en el servidor OPC, ver **Figura 20**.



**Figura 20.** Variables agregadas a la librería.

Posteriormente, deberá crear un nuevo VI. Agregue controles e indicadores numéricos para manipular y visualizar los datos de envío y recepción, en la **Figura 21** puede verse el diseño por el que puede optar para el *Front Panel* de la interfaz.



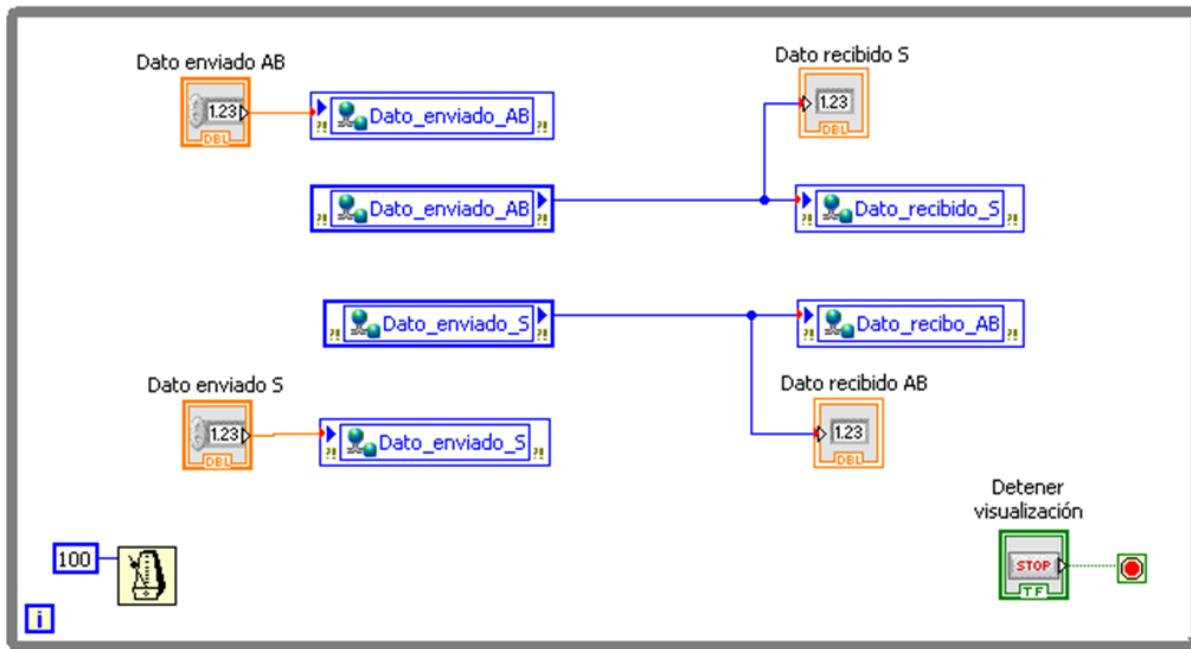
**Figura 21.** Interfaz (VI) para el control y supervisión de la transferencia de datos.

Luego, la programación para lograr establecer la comunicación deberá hacerse de la siguiente manera desde el *Block Diagram*:

*Dato\_enviado\_AB* en modo de acceso *Read* conectado a *Dato\_recibido\_S* en modo de acceso *Write*.

*Dato\_enviado\_S* en modo de acceso *Read* conectado a *Dato\_recibido\_AB* en modo de acceso *Write*.

En la **Figura 22** puede verse el diagrama de bloques con las variables conectadas para establecer comunicación y controles e indicadores asociados al control y visualización de variables del proceso en la interfaz.

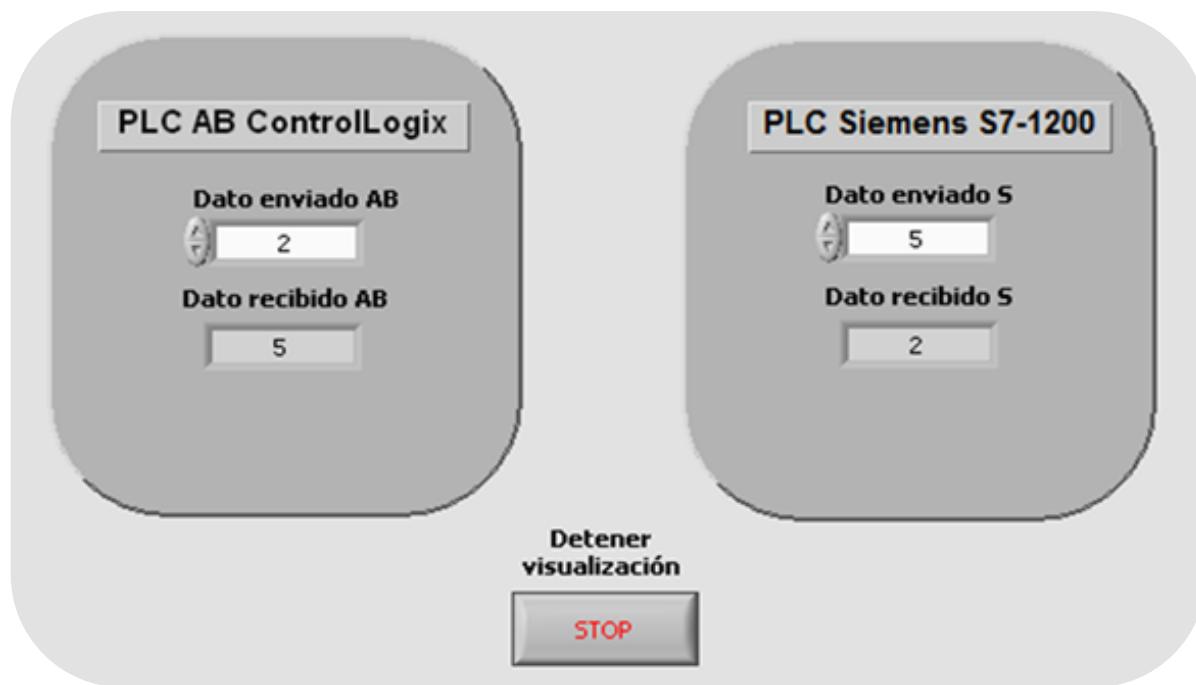


**Figura 22.** *Block Diagram* de la interfaz para la transferencia de datos.

Terminado el programa para establecer comunicación y la interfaz gráfica descargue y ejecute las configuraciones previas (creación de variables) en cada PLC, después

ejecute la interfaz (*Run*) para realizar la prueba de comunicación entre los controladores.

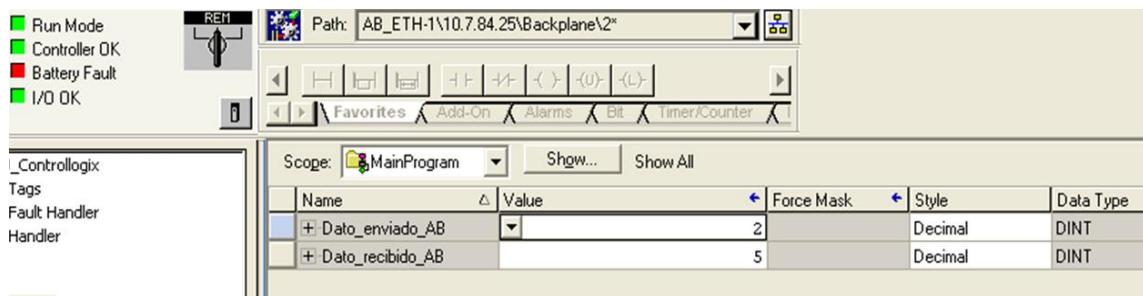
Ajuste los parámetros *Dato\_enviado\_AB* y *Dato\_enviado\_S* dentro de la interfaz en cualquier valor permisible para el tipo de dato con el cual se está trabajando (entero sin signo) y note como es transmitido al otro PLC, para ello visualice las **Figura 23** (Interfaz gráfica), **Figura 24** (*Tabla de observación\_1* en PLC\_S) y **Figura 25** (*MainProgram* en PLC\_AB).



**Figura 23.** Interfaz (VI) para el control y supervisión de la transferencia de datos ejecutándose.

Práctica 5 > PLC_1 [CPU 1214C AC/DC/Rly] > Tablas de observación > Tabla de observación_1						
	i	Nombre	Dirección	Formato visualiz...	Valor de observa...	Valor de forzado
1		"Dato_enviado_S"	%MW10	DEC	5	<input type="checkbox"/>
2		"Dato_recibido_S"	%MW12	DEC	2	<input type="checkbox"/>
3		<Agregar>				<input type="checkbox"/>

**Figura 24.** *Tabla de observación\_1* del PLC SIMATIC S7-1200 en ejecución.



**Figura 25.** MainProgram del PLC AB ControlLogix en ejecución.

Es importante destacar que en este ejercicio se transmitieron datos del tipo entero, sin embargo puede transmitir una gran variedad de tipos de datos siempre y cuando se mantenga la congruencia de los tipos y tamaños de los datos en lo que implica el proceso de comunicación, es decir, de extremo a extremo (PLC / Gateway [OPC-LabVIEW] / PLC).

**Nota:** Es posible manipular la data transmitida prescindiendo de los controles de la interfaz (manteniendo solo el conexionado de las variables de los PLCs) y haciendo uso de los entornos de programación (TIA PORTAL y RSLogix 5000).

#### 4.3. Diseño de un secuenciador doble controlado y supervisado mediante una HMI.

Con el fin de poner en práctica la comunicación entre los dispositivos en una aplicación de mayor alcance ilustrativo y didáctico, se diseñará un secuenciador doble con las siguientes características a considerar:

- Ambos PLCs ejecutarán un algoritmo que permita la activación y desactivación secuencial de tres (3) salidas (Primero el PLC\_S y luego el PLC\_AB) de forma cíclica, además, cada PLC deberá ser capaz de contar el número de veces que se ha repetido la secuencia, y finalizado el conteo (establecido por el usuario), se activará una cuarta salida (en cada PLC).

- Se hará uso de una comunicación bidireccional entre los PLCs para establecer el funcionamiento cíclico.
- El PLC\_S será el encargado de iniciar y detener el proceso (Maestro).
- En el PLC\_S deberán asociarse variables a PT (tiempo preseleccionado) en los temporizadores y CV (valor de conteo actual) y PV (valor de conteo predeterminado) en los contadores para su posterior uso y control en la HMI
- En el PLC\_AB las variables PRE (preajuste) y ACC (acumulado) de los temporizadores y contadores se utilizaran para el monitoreo y control del proceso secuencial desde la HMI.

En cuanto a la HMI, considere lo siguiente:

Cada PLC tendrá que contar con sus respectivos controles e indicadores de proceso: INICIO, PARADA (únicamente PLC\_S), Ajuste de tiempo (velocidad del secuenciador), Ajuste de cuenta (conteo de repeticiones de secuencia), Reiniciar conteo, Conteo actual, Indicador 1, Indicador 2, Indicador 3, Indicador de Finalización de cuenta.

En la **Figura 26** aparece un esquema básico de funcionamiento del secuenciador.

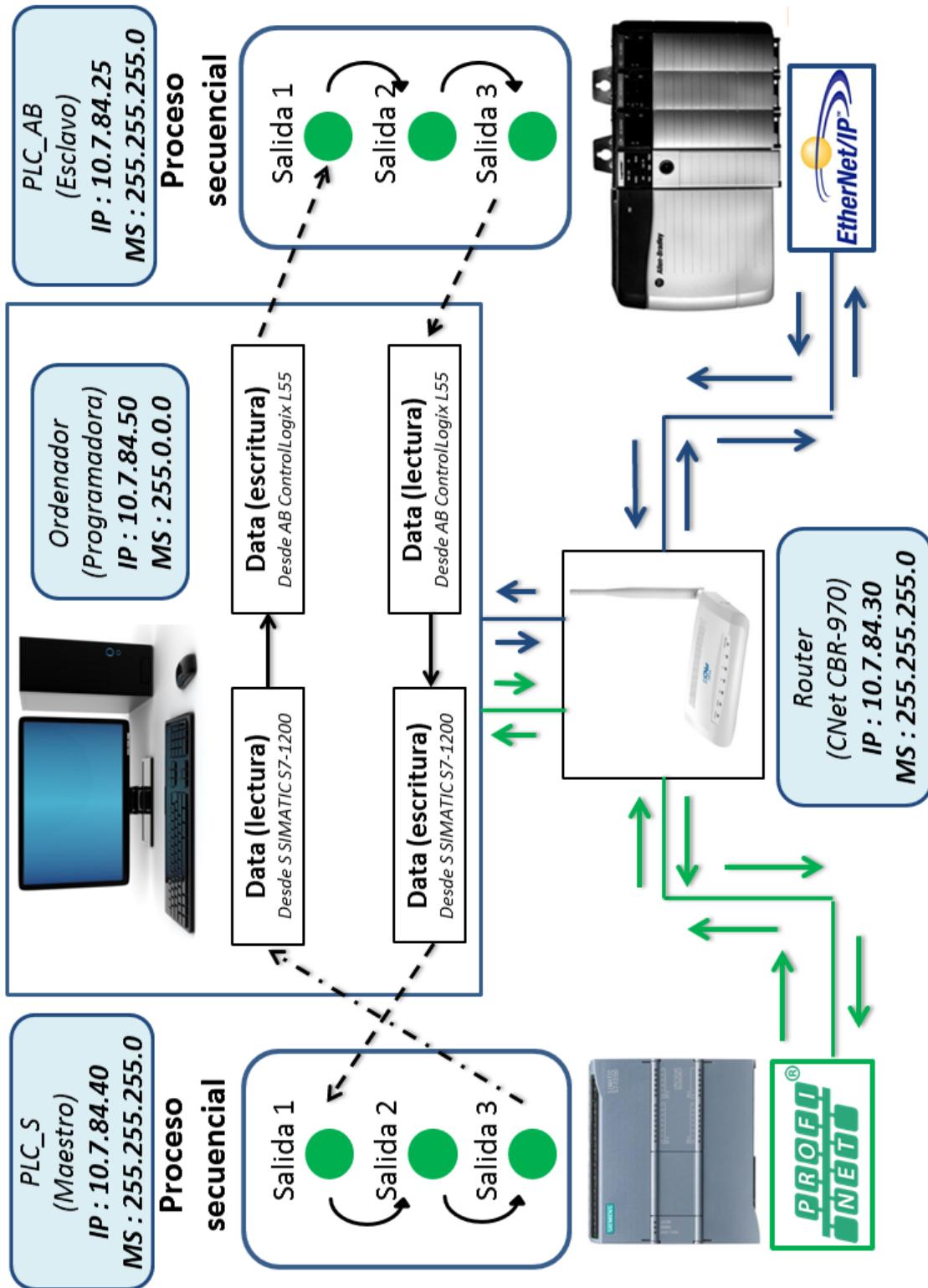


Figura 26. Esquema de funcionamiento del secuenciado doble (transmisión bidireccional).

Las lógicas que deberá programar en cada PLC son similares a las desarrolladas en el punto **5.5 de la práctica 4**.

Para empezar, diríjase al *Main [OB1]* del **PLC\_S** y programe en el la siguiente lógica:

**Segmento 1:** lógica de inicio y detención del proceso, ver **Figura 27**.

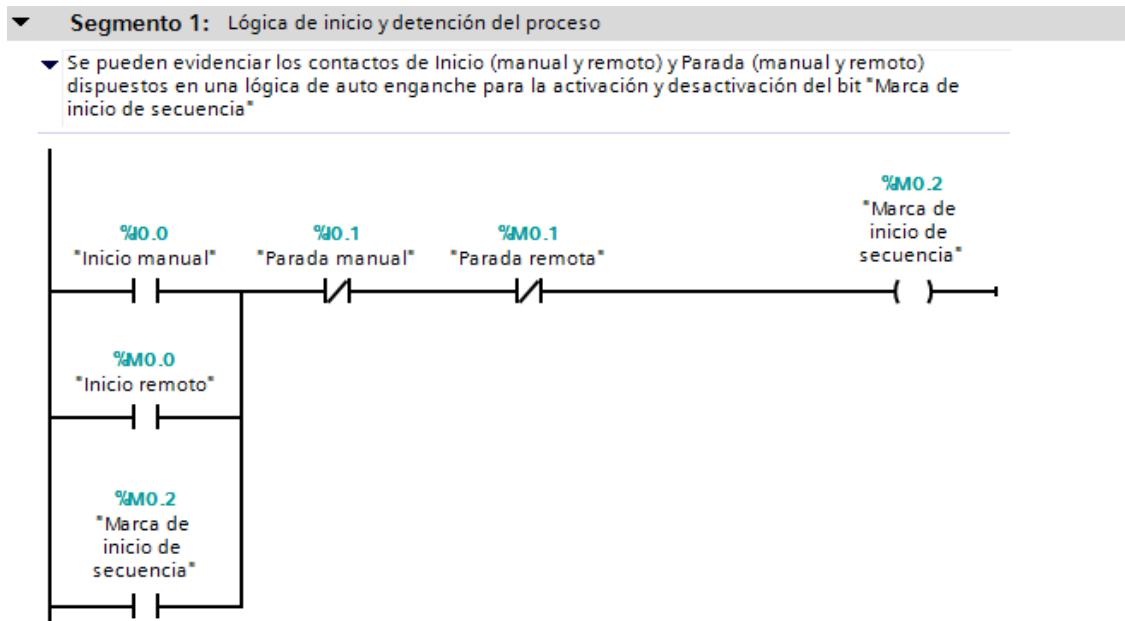


Figura 27. Segmento 1 (PLC\_S).

**Segmento 2:** lógica de secuencia, ver **Figura 28**.

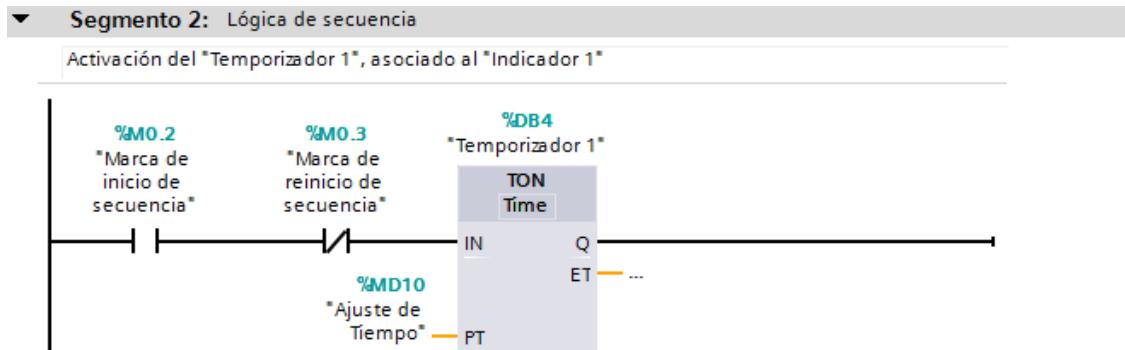
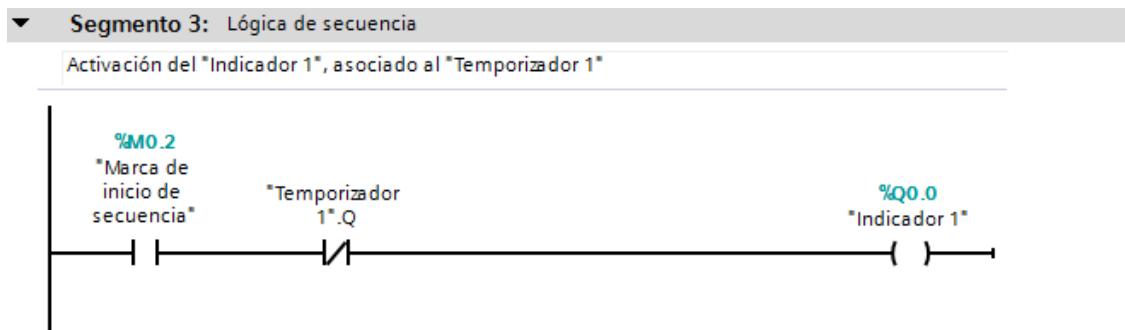


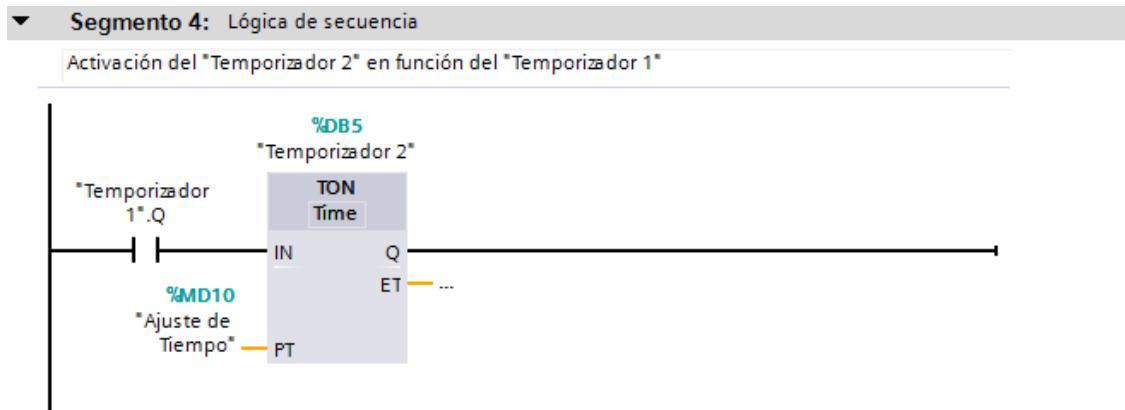
Figura 28. Segmento 2 (PLC\_S).

**Segmento 3:** lógica de secuencia, ver **Figura 29.**



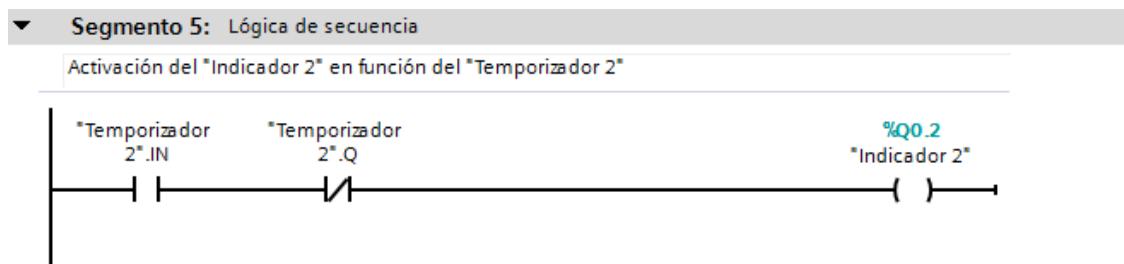
**Figura 29.** Segmento 3 (PLC\_S).

**Segmento 4:** lógica de secuencia, ver **Figura 30.**



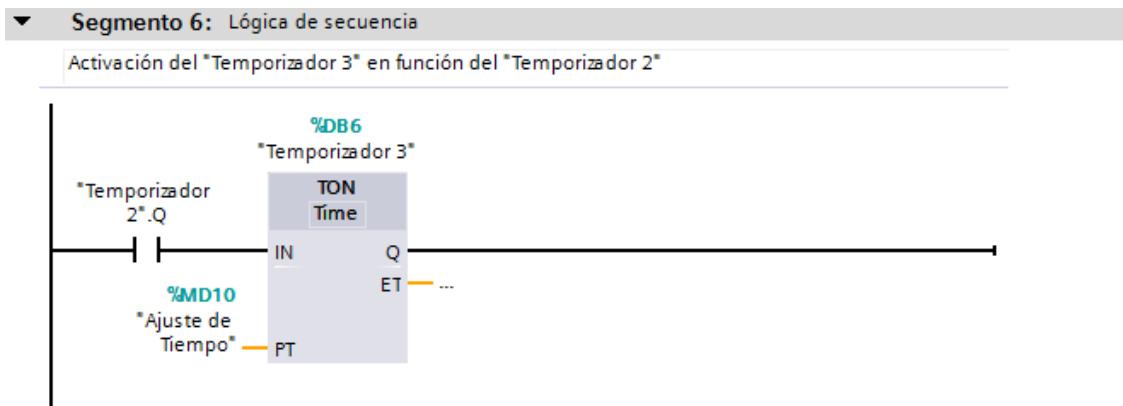
**Figura 30.** Segmento 3 (PLC\_S).

**Segmento 5:** lógica de secuencia, ver **Figura 31.**



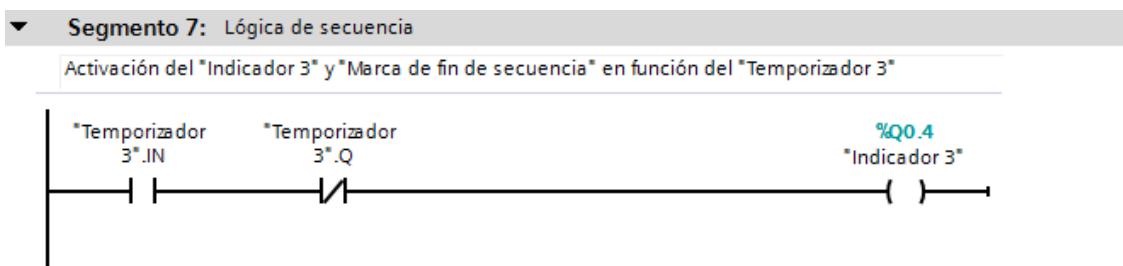
**Figura 31.** Segmento 5 (PLC\_S).

**Segmento 6:** lógica de secuencia, ver **Figura 32.**



**Figura 32.** Segmento 6 (PLC\_S).

**Segmento 7:** lógica de secuencia, ver **Figura 33.**

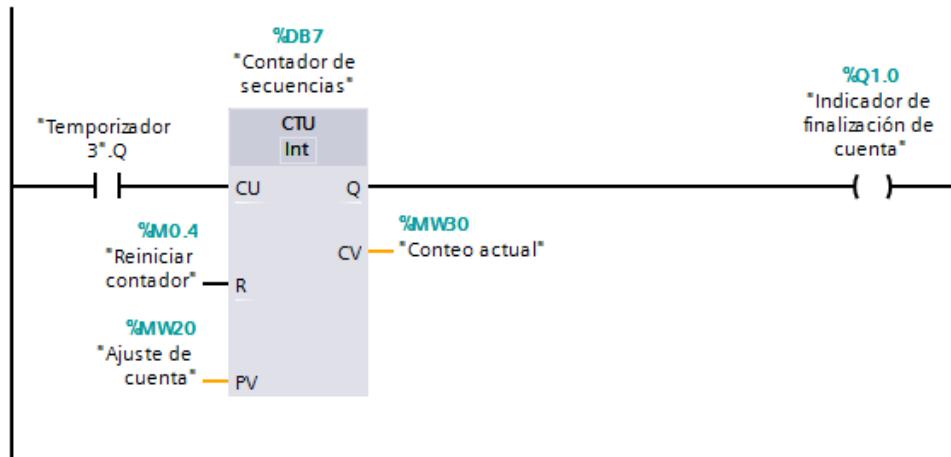


**Figura 33.** Segmento 7 (PLC\_S).

**Segmento 8:** lógica de conteo de secuencias, ver **Figura 34**.

▼ Segmento 8: Lógica de Conteo de secuencias

Activación del "Contador de secuencias" en función del "Temporizador 3"

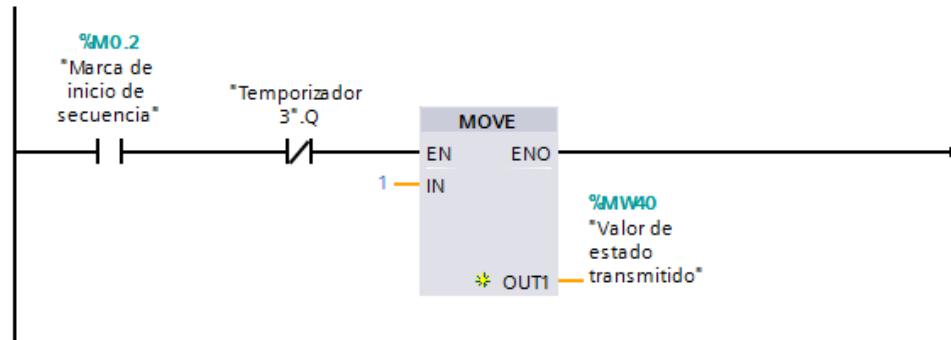


**Figura 34.** Segmento 8 (PLC\_S).

**Segmento 9:** lógica de estado de operatividad de la etapa de secuencia del PLC\_2, ver **Figura 35.**

▼ Segmento 9: Lógica de estado de operatividad de la etapa de secuencia del PLC\_1

Transferencia del valor 1 (Activo) hacia MW10 en "función del Temporizador 3"



**Figura 35.** Segmento 9 (PLC\_S).

**Segmento 10:** lógica de estado de operatividad de la etapa de secuencia del PLC\_1 y PLC\_2, ver **Figura 36.**

▼ Segmento 10: Lógica de estado de operatividad de la etapa de secuencia del PLC\_1 y PLC\_2

Transferencia del valor 0 (Inactivo) hacia MW10 en función del "Temporizador 3"

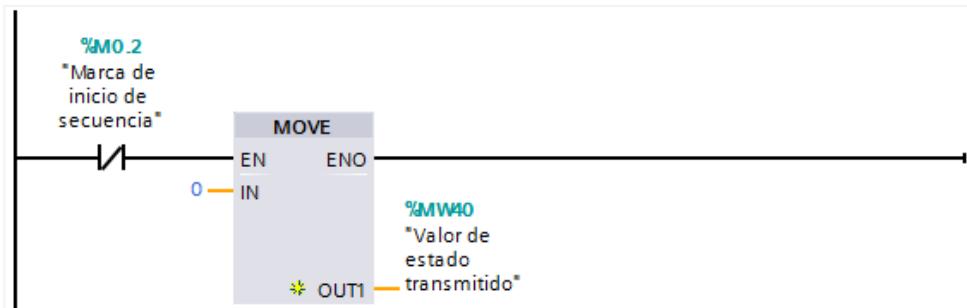


Figura 36. Segmento 10 (PLC\_S).

**Segmento 11:** lógica de estado de operatividad de la etapa de secuencia del PLC\_2, ver **Figura 37**.

▼ Segmento 11: Lógica de estado de operatividad de la etapa de secuencia del PLC\_2

Transferencia del valor 2 (Activo) hacia MW10 en "función del Temporizador 3"

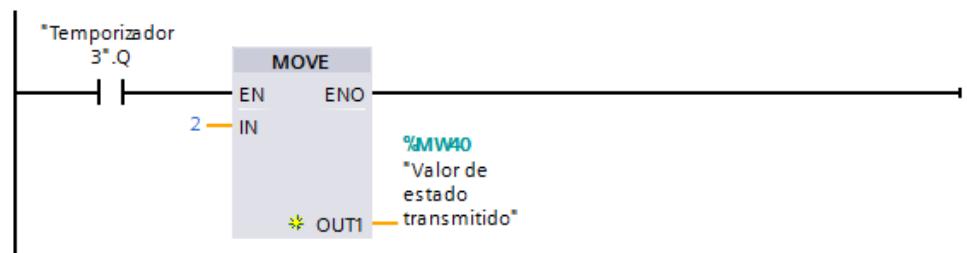


Figura 37. Segmento 11 (PLC\_S).

**Segmento 12:** lógica de reinicio de secuencia en función del PLC\_2, ver **Figura 38**.

▼ Segmento 12: Lógica de reinicio de secuencia en función del PLC\_2

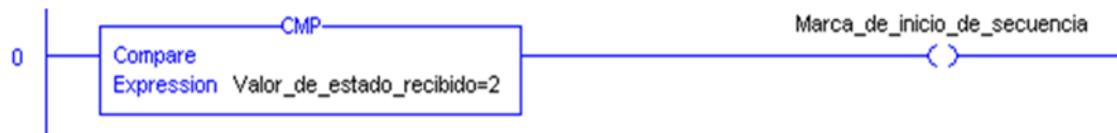
Comentario



Figura 38. Segmento 11 (PLC\_S).

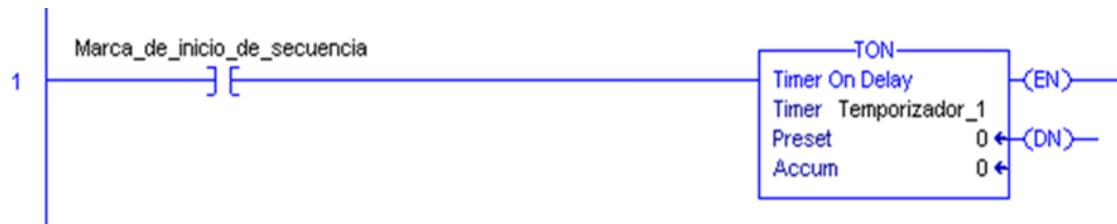
Finalizado el programa en el **PLC\_S**, vaya al *MainProgram / MainRoutine* del **PLC\_AB**, donde deberá desarrollar el programa como se muestra a continuación:

**Segmento 0:** lógica de inicio de secuencia en función del PLC\_S, ver **Figura 39**.



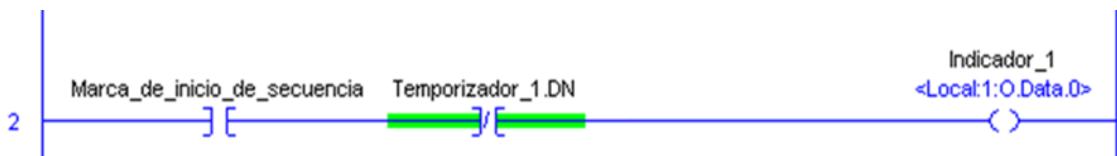
**Figura 39.** Segmento 0 (PLC\_AB).

**Segmento 1:** lógica de secuencia, ver **Figura 40**.



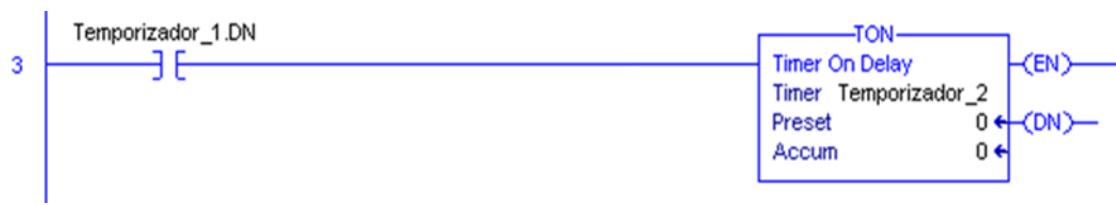
**Figura 40.** Segmento 1 (PLC\_AB).

**Segmento 2:** lógica de secuencia, ver **Figura 41**.



**Figura 41.** Segmento 2 (PLC\_AB).

**Segmento 3:** lógica de secuencia, ver **Figura 42**.



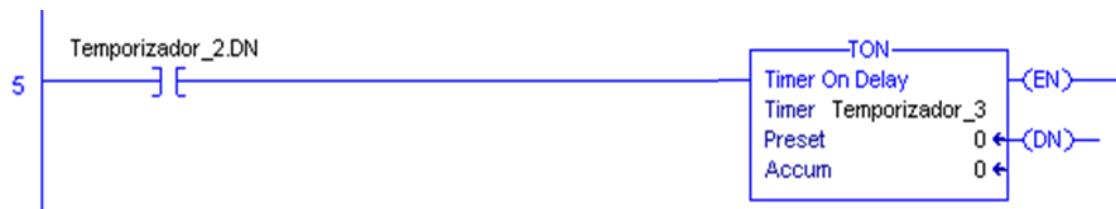
**Figura 42.** Segmento 3 (PLC\_AB).

**Segmento 4:** lógica de secuencia, ver **Figura 43.**



**Figura 43.** Segmento 4 (PLC\_AB).

**Segmento 5:** lógica de secuencia, ver **Figura 44.**



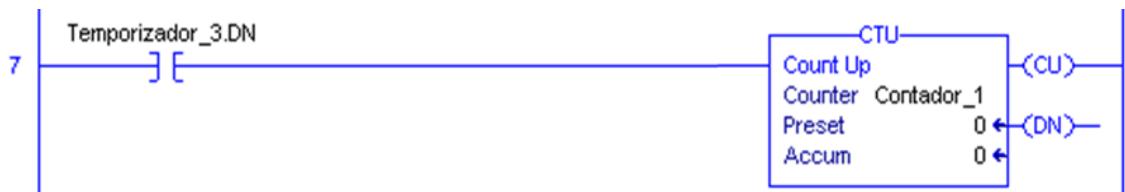
**Figura 44.** Segmento 5 (PLC\_AB).

**Segmento 6:** lógica de secuencia, ver **Figura 45.**



**Figura 45.** Segmento 6 (PLC\_AB).

**Segmento 7:** lógica de conteo de secuencias, ver **Figura 46.**



**Figura 46.** Segmento 7 (PLC\_AB).

**Segmento 8:** lógica de conteo de secuencias, ver **Figura 47**.



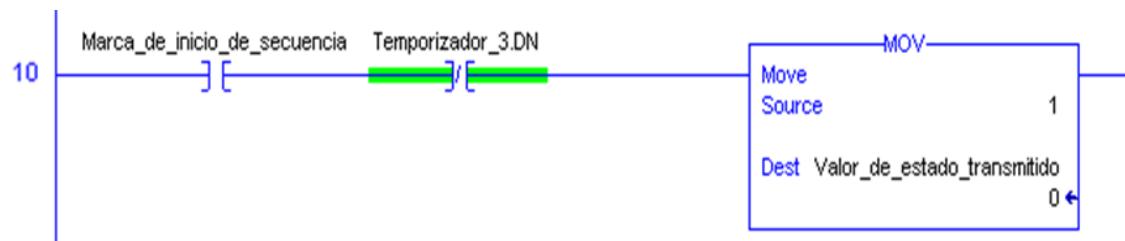
**Figura 47.** Segmento 8 (PLC\_AB).

**Segmento 9:** lógica de conteo de secuencias (reiniciar contador), ver **Figura 48**.



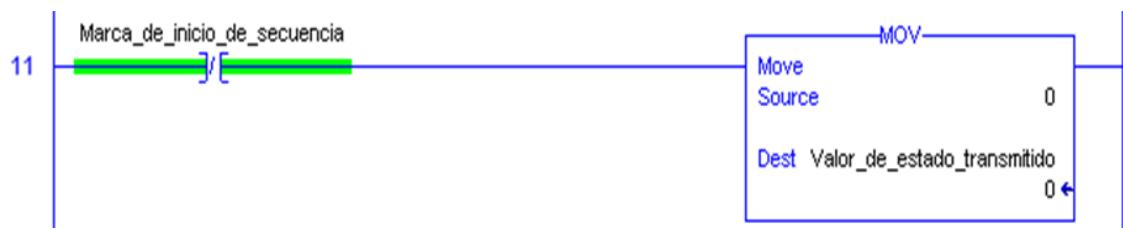
**Figura 48.** Segmento 9 (PLC\_AB).

**Segmento 10:** lógica estado de operatividad de la etapa de secuencia del PLC\_AB, ver **Figura 49**.



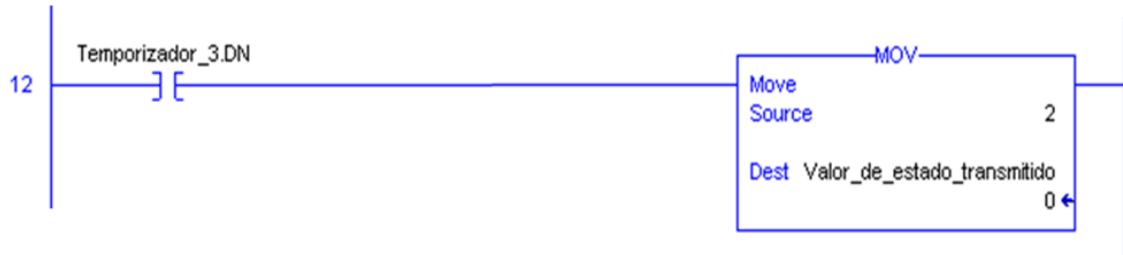
**Figura 49.** Segmento 10 (PLC\_AB).

**Segmento 11:** lógica estado de operatividad de la etapa de secuencia del PLC\_AB, ver **Figura 50**.



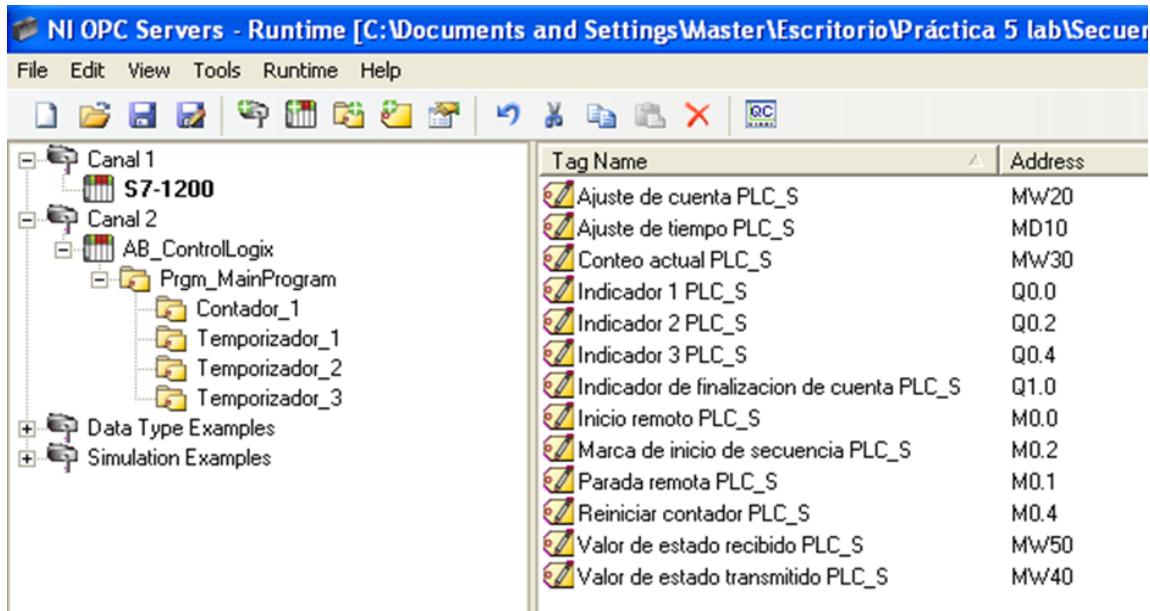
**Figura 50.** Segmento 11 (PLC\_AB).

**Segmento 12:** lógica estado de operatividad de la etapa de secuencia del PLC\_AB, ver **Figura 51.**

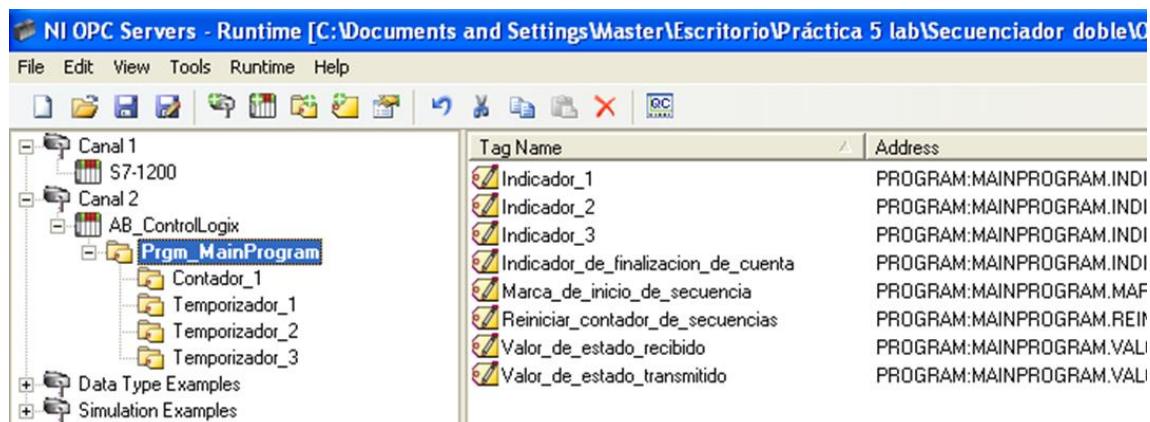


**Figura 51.** Segmento 11 (PLC\_AB).

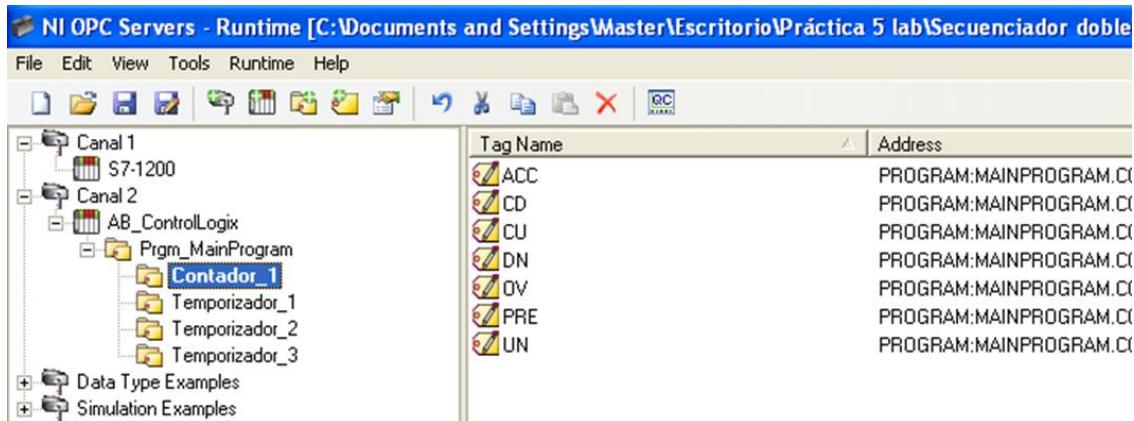
Finalizada la creación de las lógicas en cada PLC, ejecute el NI OPC server, en el cree y configure dos canales cada uno con su respectivo equipo y variables asociadas, ver **Figura 52** (Canal 1 S7-1200), **Figura 53** (Canal 2 AB ControlLogix *Prgm\_MainProgram*), **Figura 54** (Canal 2 AB ControlLogix *Prgm\_MainProgram / Contador\_1*), **Figura 55** (Canal 2 AB ControlLogix *Prgm\_MainProgram / Temporizador\_1/2/3*).



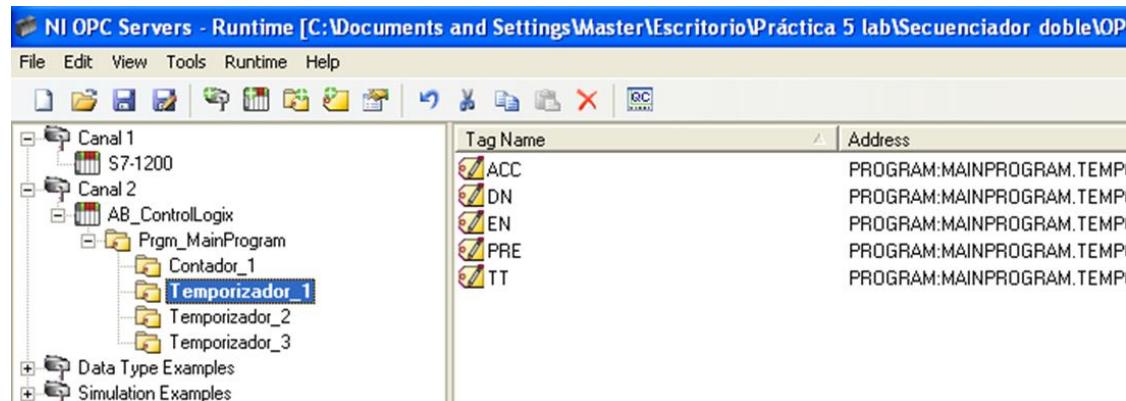
**Figura 52.** Canal 1 con el PLC SIMATIC S7-1200 configurado y variables agregadas.



**Figura 53.** Canal 2 con el PLC AB ControlLogix configurado y variables agregadas en *Prgm\_MainProgram*.

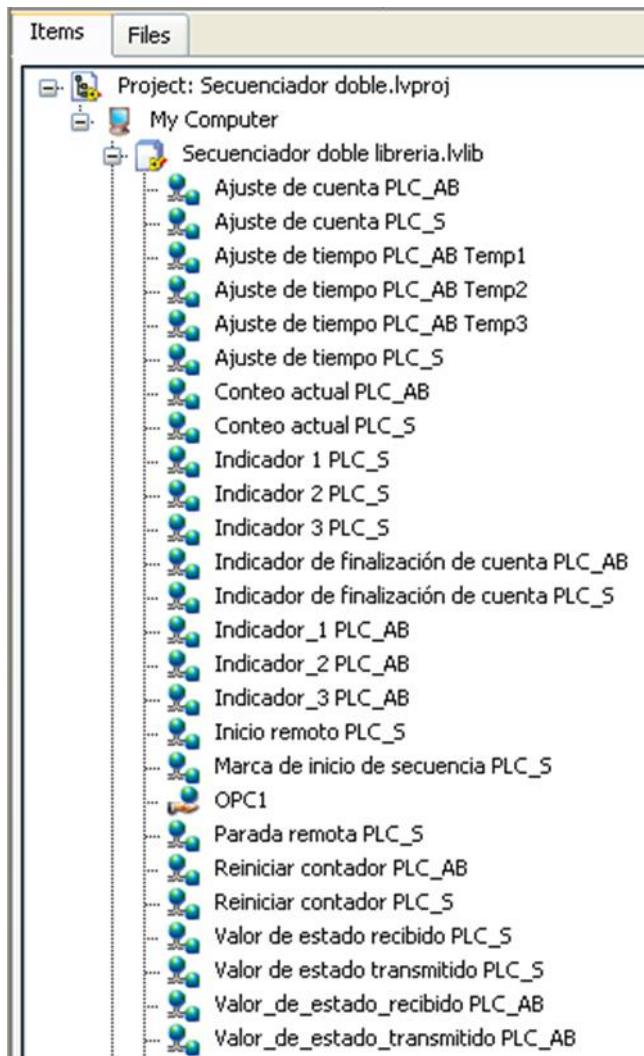


**Figura 54.** Canal 2 con el PLC AB ControlLogix configurado y variables agregadas en *Prgm\_MainProgram / Contador\_1*.



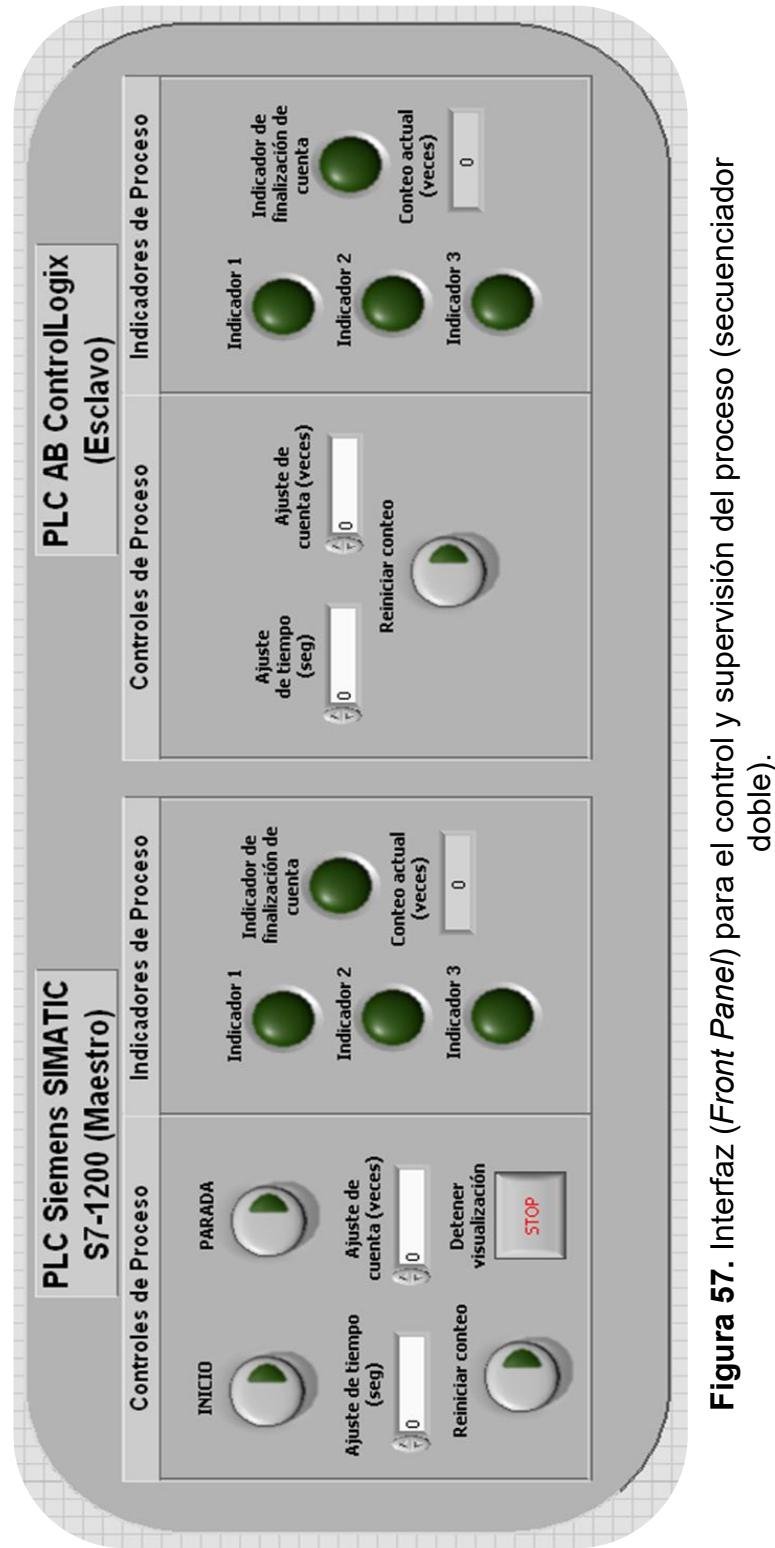
**Figura 55.** Canal 2 con el PLC AB ControlLogix configurado y variables agregadas en *Prgm\_MainProgram / Temporizador\_1*.

Configurado el servidor OPC ejecute LabVIEW, en el cree un *Blank Project* y configure el *OPC Client*, de tal modo se creará la librería contenedora de las variables, luego agregue las variables en función de los *tags* creados previamente en el servidor OPC, ver **Figura 56**.



**Figura 56.** Variables agregadas a la librería.

Después cree un nuevo VI. Agregue controles e indicadores numéricos para manipular y visualizar los datos de envío y recepción, en la **Figura 57** se visualiza el diseño por el que puede optar para el *Front Panel* de la interfaz.



**Figura 57.** Interfaz (Front Panel) para el control y supervisión del proceso (secuenciador doble).

Luego, la programación para lograr establecer la comunicación deberá hacerse de la siguiente manera desde el *Block Diagram*:

*Valor de estado transmitido PLC\_S* en modo de acceso *Read* conectado a *Valor\_de\_estado\_recibido PLC\_AB* en modo de acceso *Write*.

*Valor\_de\_estado\_transmitido PLC\_AB* en modo de acceso *Read* conectado a *Valor de estado recibido PLC\_S* en modo de acceso *Write*.

En la **Figura 58** se muestra el diagrama de bloques con las variables conectadas para establecer comunicación y controles e indicadores asociados al control y visualización de variables del proceso en la interfaz.

Finalizado el programa para establecer comunicación y la interfaz gráfica, descargue los programas a cada PLC y colóquelos en modo *Run*, después ejecute la interfaz (*Run*) para realizar la prueba de comunicación y funcionamiento del proceso entre los controladores.

Obsérvese:

En la **Figura 59** la interfaz en ejecución.

En la **Figura 60** (PLC\_S) y **Figura 61** (PLC\_AB) el monitoreo de los *tags* de cada PLC mientras se ejecuta el proceso mediante la herramienta *Quick Client*.

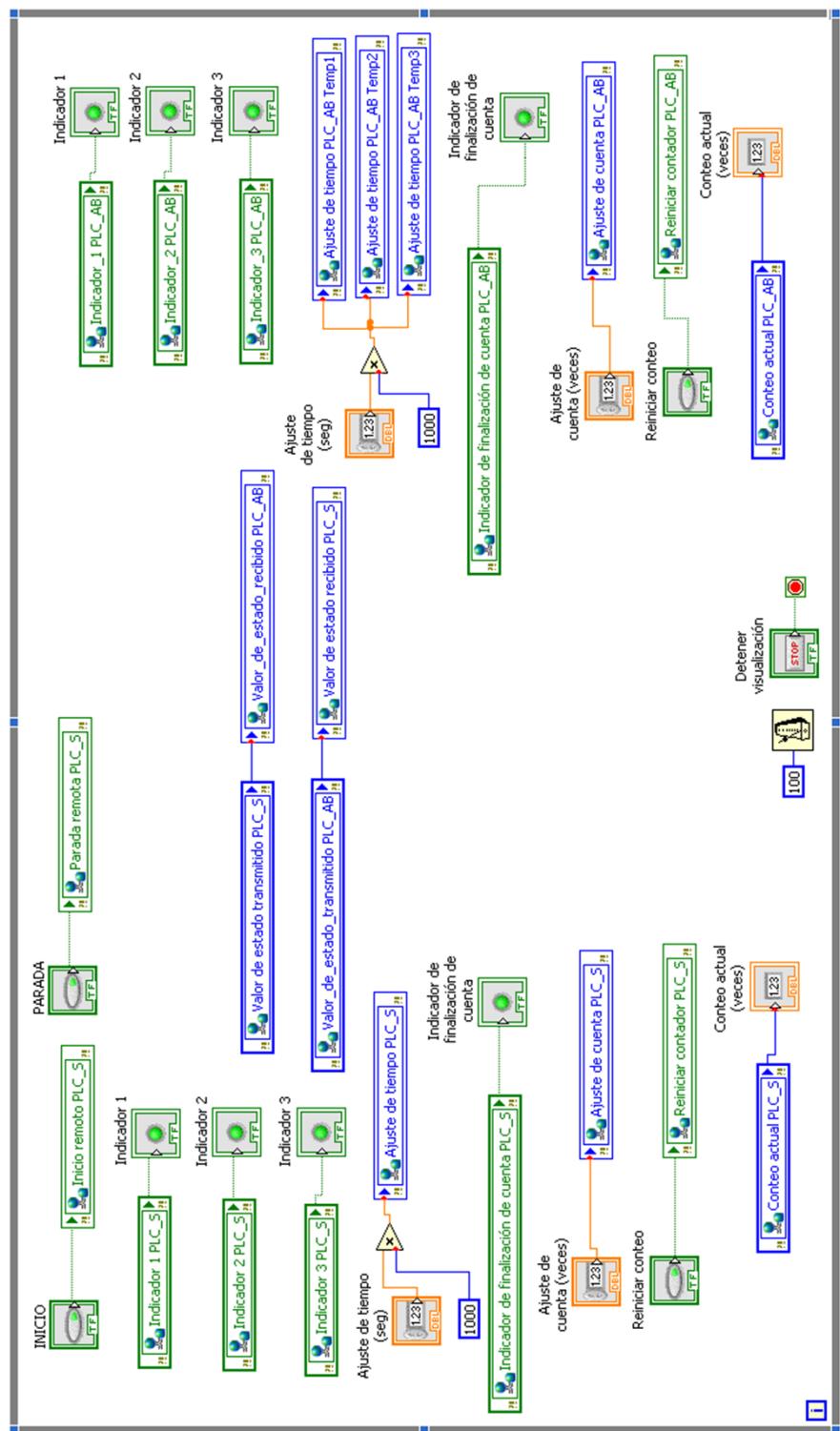
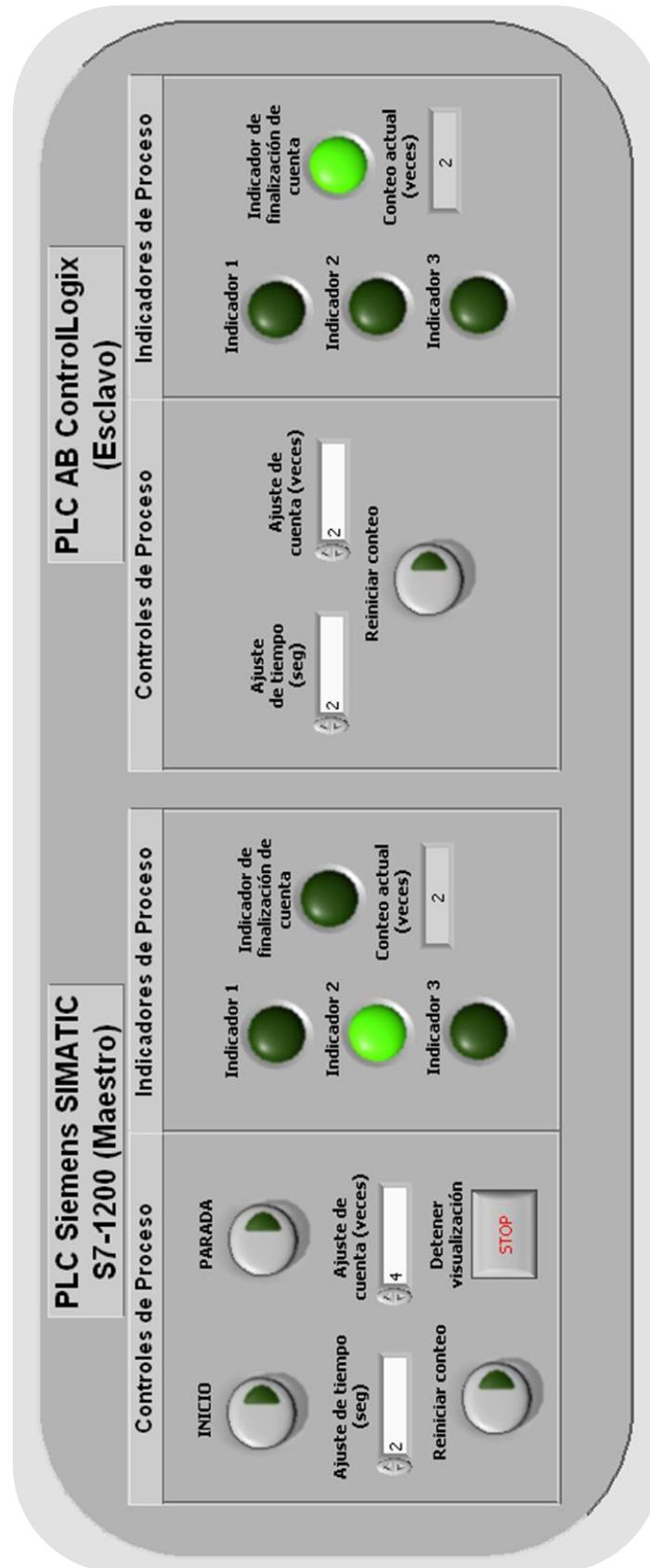


Figura 58. Block Diagram de la interfaz para la comunicación, control y supervisión del proceso entre los PLCs (secuenciador doble).



**Figura 59.** Interfaz (Front Panel) para el control y supervisión del proceso (secuenciador doble) en ejecución.

Item ID		Data Type	Value	Timestamp	Quality
Canal 1.S7-1200.Rack		Byte	0	02:16:10.484	Good
Canal 1.S7-1200.Slot		Byte	1	02:16:10.484	Good
Canal 1.S7-1200.Ajuste de cuenta PLC_S		Word	4	02:16:11.468	Good
Canal 1.S7-1200.Ajuste de tiempo PLC_S		DWord	2000	02:16:11.468	Good
Canal 1.S7-1200.Conteo actual PLC_S		Word	2	02:16:11.468	Good
Canal 1.S7-1200.Indicador 1 PLC_S		Boolean	0	02:16:11.468	Good
Canal 1.S7-1200.Indicador 2 PLC_S		Boolean	0	02:16:11.468	Good
Canal 1.S7-1200.Indicador 3 PLC_S		Boolean	0	02:16:11.468	Good
Canal 1.S7-1200.Indicador de finalización de cuenta PLC_S		Boolean	0	02:16:11.468	Good
Canal 1.S7-1200.Inicio remoto PLC_S		Boolean	0	02:16:11.468	Good
Canal 1.S7-1200.Marca de inicio de secuencia PLC_S		Boolean	0	02:16:11.468	Good
Canal 1.S7-1200.Parada remota PLC_S		Boolean	0	02:16:11.468	Good
Canal 1.S7-1200.Reiniciar contador PLC_S		Boolean	0	02:16:11.468	Good
Canal 1.S7-1200.Valor de estado recibido PLC_S		Word	0	02:16:11.468	Good
Canal 1.S7-1200.Valor de estado transmitido PLC_S		Word	0	02:16:11.468	Good

**Figura 60.** Verificación del correcto monitoreo de los *tags* del PLC\_S.

Item ID		Data Type	Value	Timestamp	Quality
Canal 2.AB_ControlLogix.Prgm_MainProgram.Contador_1.ACC		Long	2	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Contador_1.CD		Boolean	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Contador_1.CU		Boolean	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Contador_1.DN		Boolean	1	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Contador_1.0V		Boolean	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Contador_1.PRE		Long	2	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Contador_1.UN		Boolean	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Indicador_1		Boolean	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Indicador_2		Boolean	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Indicador_3		Boolean	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Indicador_de_finalizacion_de_cuenta		Boolean	1	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Marca_de_inicio_de_secuencia		Boolean	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Reiniciar_contador_de_secuencias		Boolean	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Temporizador_1.ACC		Long	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Temporizador_1.DN		Boolean	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Temporizador_1.EN		Boolean	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Temporizador_1.PRE		Long	2000	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Temporizador_1.TT		Boolean	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Temporizador_2.ACC		Long	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Temporizador_2.DN		Boolean	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Temporizador_2.EN		Boolean	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Temporizador_2.PRE		Long	2000	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Temporizador_2.TT		Boolean	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Temporizador_3.ACC		Long	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Temporizador_3.DN		Boolean	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Temporizador_3.EN		Boolean	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Temporizador_3.PRE		Long	2000	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Temporizador_3.TT		Boolean	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Valor_de_estado_recibido		Long	0	02:16:10.515	Good
Canal 2.AB_ControlLogix.Prgm_MainProgram.Valor_de_estado_transmitido		Long	0	02:16:10.515	Good

**Figura 61.** Verificación del correcto monitoreo de los *tags* del PLC\_AB