



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Apuntes para el uso básico
del PLC S7-1200 para las
asignaturas del área de
Automatización Industrial**

MATERIAL DIDÁCTICO

Que para obtener el título de
Ingeniero Mecatrónico

P R E S E N T A

Alejandro Espino Romero

ASESOR DE MATERIAL DIDÁCTICO

M.F. Gabriel Hurtado Chong



Ciudad Universitaria, Cd. Mx., 2019

Apuntes para el uso básico del PLC S7-1200 para las asignaturas del área de
Automatización Industrial

Alejandro Espino Romero

Facultad de Ingeniería, UNAM

Agradecimientos

Trabajo realizado con el apoyo del Programa UNAM-DGAPA-PAPIME Proyecto PE107719

"En verdad hay sentimientos que es mejor que se queden en lo platónico; y es mejor recordarlos así, irreales, inacabados, porque eso es lo que los hace perfectos".

- Gabriel García Márquez

Contenido

Objetivo.....	6
Antecedentes	7
1. Introducción al Controlador de Lógica Programable (PLC) S7-1200 de Siemens	9
1.1 Componentes.....	9
1.2 Características y especificaciones técnicas	10
2. Introducción al entorno de programación TIA Portal versión 15.....	14
2.1 Componentes principales del entorno y de la interfaz.....	14
2.2 Configuración de hardware	16
2.3 Configuración de software	25
2.4 Comunicación entre el controlador y la computadora.....	28
2.5 Estructura general de un programa para el S7-1200	34
2.6 Orden de procesamiento y ejecución de un programa en el S7-1200	49
3. Programación del PLC S7-1200 mediante TIA Portal V15	53
3.1 Entradas y salidas discretas	54
3.2 Memorias o marcas	57
3.3 Tipos de temporizadores	59
3.4 Tipos de contadores.....	66
3.5 Registros de corrimiento	72
3.6 Constantes numéricas.....	77
3.7 Funciones de comparación	78

3.8	Funciones aritméticas	81
4.	Ejemplos de aplicación.....	91
	Resultados	99
	Discusión.....	100
	Índice de figuras.....	103
	Índice de tablas.....	108

Objetivo

El presente documento tiene como objetivo crear apuntes sobre los componentes básicos, la conexión y la programación de instrucciones básicas del nuevo equipo de PLC con que cuenta el laboratorio de Automatización Industrial, mismo que estará en servicio a partir del semestre 2020-2, de igual forma servirá como documentación para la impartición de cursos de capacitación para profesores que vayan a impartir las asignaturas Automatización Industrial y Automatización Avanzada. Por lo que este trabajo busca lograr que tanto alumnos como docentes conozcan las características de un PLC S7-1200 y del entorno de programación TIA Portal V15, así como aprender a implementar funciones básicas de programación y ser capaces de aplicarlas en la solución de problemas de ingeniería.

Antecedentes

El Laboratorio de Automatización Industrial (anteriormente llamado Laboratorio de Neumática y PLC) fue creado en febrero de 1994 e inició actividades docentes en enero de 1995 impartiendo clases de laboratorio para la asignatura de Electrónica Básica, la asignatura optativa de Temas Selectos de Mecatrónica para Ingeniería Mecánica y la asignatura de Automatización y Robótica para Ingeniería Industrial. Desde entonces, el laboratorio ha prestado servicio a la comunidad de manera ininterrumpida.

En el año 2002, el laboratorio cambió su nombre a como se le conoce en la actualidad y, en el año 2004, con la creación de la carrera de Ingeniería Mecatrónica, la instalación pasó a impartir la materia que lleva el mismo nombre, Automatización Industrial, de carácter optativo para las carreras de Ingeniería Mecatrónica, Ingeniería Mecánica e Ingeniería Industrial. Así mismo, se comenzó la impartición de la asignatura de Temas Selectos de Mecatrónica enfocada a Automatización Avanzada.

En el año 2016 se aprobó una nueva modificación de los planes de estudios de las carreras impartidas en la Facultad de Ingeniería, con lo que la asignatura de Automatización Industrial se convirtió en obligatoria para las carreras de Ingeniería Mecatrónica e Ingeniería Industrial, y obligatoria de elección para el campo de profundización de Automatización de la carrera de Ingeniería Mecánica. Debido a esto, la oferta de grupos de la asignatura ha aumentado a más del doble, por lo que el uso del equipo y las instalaciones también se ha visto incrementado.

El equipo con el que cuenta el laboratorio en la actualidad, particularmente los controladores lógicos programables (PLC, por sus siglas en inglés), ha estado en servicio desde la creación del laboratorio. Los equipos de PLC con los que se ha impartido la asignatura, desde hace casi 25 años, son de la marca *Square-D*, modelo Micro-1. Debido a todo el tiempo de uso que se les ha dado semestre tras semestre, estos PLC ya se encuentran deteriorados. Por ejemplo, los botones de algunos programadores manuales están ya borrosos y no se distinguen bien; además, algunos programadores tienen los soportes rotos por lo que ya no pueden asegurarse correctamente al CPU y quedan “colgando”. Aún más importante, estos equipos son ya bastante obsoletos al grado de no contar con refacciones en el mercado actualmente.

Para poder dar atención no sólo a las necesidades académicas de los alumnos sino a las necesidades planteadas por la industria de contar con profesionales con conocimiento en el manejo de equipo de vanguardia, el laboratorio de Automatización Industrial se hizo, en el año 2019, de 12 PLC de la familia S7-1200 de la marca Siemens, los cuales, entre otras cosas, cuentan con una mayor cantidad de entradas y salidas discretas, además de integrar en el mismo PLC entradas y salidas analógicas, y la posibilidad de comunicarse vía inalámbrica con los equipos de cómputo.

La brecha que existe entre el equipo adquirido y el que se tiene en el laboratorio es enorme. Si bien los fundamentos teóricos de programación son los mismos (programación en escalera), el equipo de Siemens cuenta con una gama mucho más amplia de instrucciones, además de tener un software de programación propio a través de equipo de cómputo. Esto genera que hacer uso del nuevo equipo sea muy diferente a lo que se tiene hoy en día.

A partir de lo anterior, se planteó la necesidad de generar documentación que permitiera, tener conocimientos del manejo básico de los PLC S7-1200 que vayan desde la conexión y comunicación de los equipos, hasta la programación, simulación de puesta en marcha de programas completos. Así mismo, se planteó que la población objetivo fuera, por una parte, los estudiantes que cursen alguna de las materias impartidas por el laboratorio para que de esta manera tuvieran un recurso sobre el que apoyarse a lo largo del curso para obtener información concreta para el manejo básico del equipo, sobre todo para aquellos que no han tenido ningún acercamiento a equipos de este tipo. Por otra parte, se pensó dirigir también el documento a los profesores, tanto los que actualmente impartan la asignatura como los que se puedan integrar en el futuro, para que cuenten con una referencia del manejo básico del equipo para impartir el curso de manera más fluida.

Como resultado se propuso este material como un trabajo de apoyo a la docencia que tenga utilidad para alumnos y profesores en aras de mejorar el proceso de transición del equipo usado en la impartición de la asignatura, además de servir como preámbulo para generar apuntes de estudio que apoyen el aprendizaje de los usuarios del laboratorio.

1. Introducción al Controlador de Lógica Programable (PLC) S7-1200 de Siemens

1.1 Componentes

Como parte de una donación realizada por la Sociedad de Exalumnos de la Facultad de Ingeniería (SEFI), el laboratorio de Automatización Industrial de la Facultad de Ingeniería recibió diverso equipo industrial de la marca Siemens, de entre lo que destaca un lote de Controladores de Lógica Programable (PLC) S7-1200. El CPU de este equipo cuenta con un microprocesador, una fuente de alimentación integrada, circuitos de entrada y salida (E/S), comunicación PROFINET integrada, E/S de control de movimiento de alta velocidad y entradas analógicas incorporadas. En cuanto a la comunicación, es posible añadir al PLC módulos adicionales para trabajar en redes PROFIBUS, GPRS, RS485 o RS232.

1

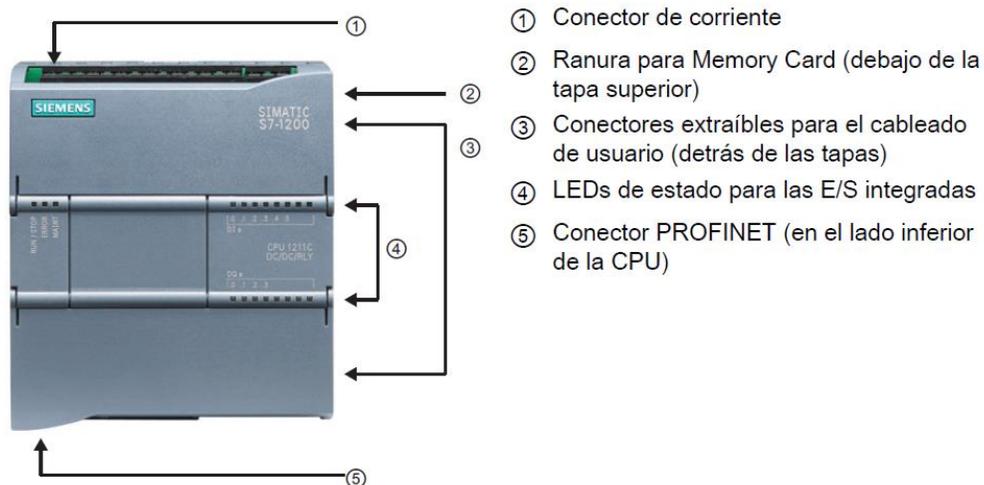


Figura 1.1 - Vista frontal de un PLC S7-1200 con un CPU 1211C

El equipo se recibió distribuido dentro de 6 gabinetes industriales previamente armados, e incluye los siguientes elementos, para la impartición de las asignaturas de Automatización Industrial (AI) y de Automatización Avanzada (AA):

- 12 CPU 1215C AC/DC/RLY
- 12 licencias de software SIMATIC STEP 7 TIA PORTAL BÁSICO
- 12 módulos de 16 entradas digitales y 16 salidas digitales SIMATIC S7-1200
- 4 switches compactos con 4 puertos RJ45 SIMATIC S7-1200 CSM 1277

¹ [Imagen de un PCL S7-1200] Siemens. SIMATIC Controlador programable S7-1200. Manual de sistema (2015).

- 2 interfaces HMI KTP 1200 Básico
- 2 fuentes de alimentación Power Module PM1207
- 6 puntos de acceso para red inalámbrica SCALANCE W761-1 RJ45
- 6 antenas omnidireccionales IWLAN ANT795-6MP
- 6 cables R-SMA confeccionados para la conexión de ANT795-6MP.

1.2 Características y especificaciones técnicas

A continuación, se presentan las características técnicas principales para algunos de los equipos recibidos.

Tabla 1.1 - Características del CPU 1215C

Característica	Unidades
Memoria de usuario	100 [kB] de trabajo
	4 [MB] de carga
	10 [kB] remanente
E/S integradas locales	14E/10S (Discretas) 2E/2S (Analógicas)
Memoria imagen de proceso	1024 [bytes]
Área de marcas	8192 [bytes]
Slots de ampliación con módulo de señales	8
Slots de ampliación con módulo de comunicación	3
Contadores rápidos (HSC)	6
Generadores de impulsos	4
Puertos PROFINET	2 (Ethernet)

En los CPU de gama S7-1200 la memoria de usuario se encuentra dividida en tres tipos:

De carga: Permite almacenar de forma no volátil el programa del usuario, los datos del sistema y la configuración del equipo.

De trabajo: Almacena las partes del programa de usuario que son relevantes para la ejecución del programa, el CPU copia elementos del proyecto desde la memoria de carga a la memoria de trabajo. La ejecución del programa tiene lugar exclusivamente en el área correspondiente a la memoria de trabajo.

Remanente: Se emplea para almacenar datos de manera no volátil de la memoria de trabajo. Si se produce algún corte en la alimentación, el CPU puede retomar nuevamente ciertos valores a partir de la memoria remanente.

2



Figura 1.2 - Vista frontal de un PLC con CPU 1215C

² [Imagen de CPU 1215C]. Recuperado de <https://www.amazon.com/Siemens-215-1AG40-0XB0-SIMATIC-S7-1200-PROFINET/dp/B076CXPZBP> el 7 de octubre de 2019 a las 15:35

Tabla 1.2 - Características de los módulos E/S SM 1223

Característica	Unidades
Entradas discretas	16
Salidas discretas	16 a relevador
Voltaje a las entradas (configurable)	[20.4, 28.8] [V]
Corriente de entrada permitida	[1.0, 4.0] [mA]
Voltaje a las salidas permitido	[5, 250] [V _{CA}] [5, 30] [V _{CD}]
Corriente máxima a la salida	2 [A]
Voltaje en las bobinas	24 [V _{DC}]

3



Figura 1.3 - Vista del módulo adicional de E/S

³ [Imagen de módulo SM1223]. Recuperado de <https://www.conradelektronik.dk/p/sps-digitalt-in-outputmodul-siemens-sm-1223-6es7223-1ph32-0xb0-197577> el 7 de octubre de 2019 a las 15:40

Tabla 1.3 - Características de la HMI KTP1200

Característica	Unidades
Modo de operación	Táctil + Teclas
Tamaño	12.1 [in]
Resolución	1200 x 800 [píxeles]
Memoria de usuario	10 [MB]
Número de variables máximo	800

4



Figura 1.4 - Interfaz HMI 1200 Basic

⁴ [Imagen de módulo HMI 1200 Basic]. Recuperado de <https://www.fluitronic.es/simatic-hmi-ktp1200-basic-pn-6av2123--2mb03--0ax0> el 7 de octubre de 2019 a las 15:53

2. Introducción al entorno de programación TIA Portal versión 15

El entorno de programación TIA Portal brinda una alternativa sencilla para la edición de los programas de control de los PLC, además de poder hacer conexiones de redes entre múltiples dispositivos o equipos adicionales como HMI o módulos de control de motores, por ejemplo. El software que se tiene en el laboratorio es SIMATIC STEP 7 Basic, el cual cuenta con herramientas de control para los PLC S7-1200 y los paneles de la Gama HMI Basic Panels. Así mismo, STEP 7 (S7) ofrece la posibilidad de trabajar con los siguientes lenguajes de programación:

- KOP (LD o LAD): Diagrama de contactos o lógica de escalera.
- FBS (FUP o FBD): Diagrama de funciones, programación por bloques.
- SLC: Lenguaje estructurado, programación basada en texto.

2.1 Componentes principales del entorno y de la interfaz

El software TIA Portal ofrece la posibilidad de trabajar con dos vistas diferentes del proyecto, con lo que se puede usar la vista que mejor se adapte a las necesidades de trabajo del usuario. La **vista de portal** (Figura 2.1) muestra diferentes bloques (llamados portales) que están orientados a tareas específicas y se encuentran organizados según las funciones de las herramientas que contienen. La **vista del proyecto** (Figura 2.2) está orientada a los elementos de programación del proyecto actual, la creación de programas y la comunicación de equipos. En la Figura 2.1 se muestran los siguientes elementos de la vista de portal:

1. **Área de portales:** Aquí se muestran los diferentes bloques de funciones con los que se organizan las tareas del software.
2. **Tareas de portal:** En esta parte se muestran las tareas específicas con las que cuenta el portal seleccionado.
3. **Panel de selección:** Las diferentes opciones para la tarea seleccionada se muestran aquí.
4. **Cambio de vista:** Permite cambiar el entorno a la vista de proyecto.

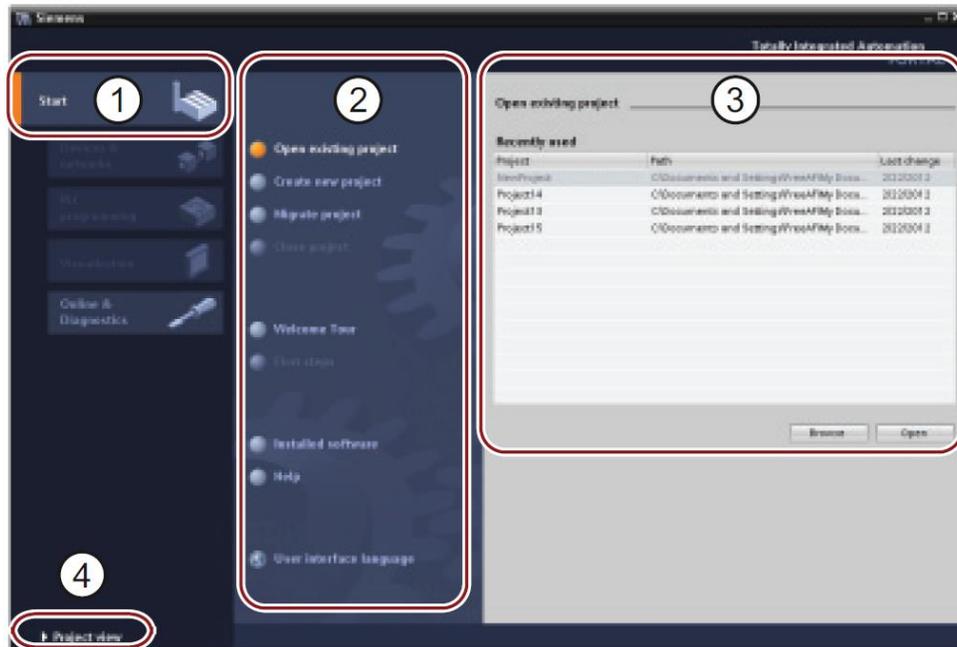


Figura 2.1 - Visualización de la vista de portal en S7

En la Figura 2.2 se muestran los siguientes elementos de la vista de proyecto:

1. **Menús y barra de herramientas:** Se muestran los diferentes menús de edición y herramientas del proyecto.
2. **Árbol del proyecto:** Aquí se despliegan los dispositivos con los que se está trabajando, así como los elementos de programación creados.
3. **Área de trabajo:** En este espacio se muestran las redes de dispositivos, los elementos de hardware de cada componente, además de ser el lugar donde se crean y se editan los programas.
4. **Task cards:** Aquí se encuentran las instrucciones del programa. Las instrucciones se agrupan por carpetas (funciones).
5. **Ventana de inspección:** Muestra las propiedades e información acerca del objeto seleccionado en el área de trabajo.
6. **Cambio de vista:** Permite cambiar el entorno a la vista de portal.
7. **Barra de editor:** Muestra todos los editores de programa que están abiertos.

⁵ [Vista de portal en TIA Portal v15] Siemens. SIMATIC Controlador programable S7-1200. Manual de sistema (2015).

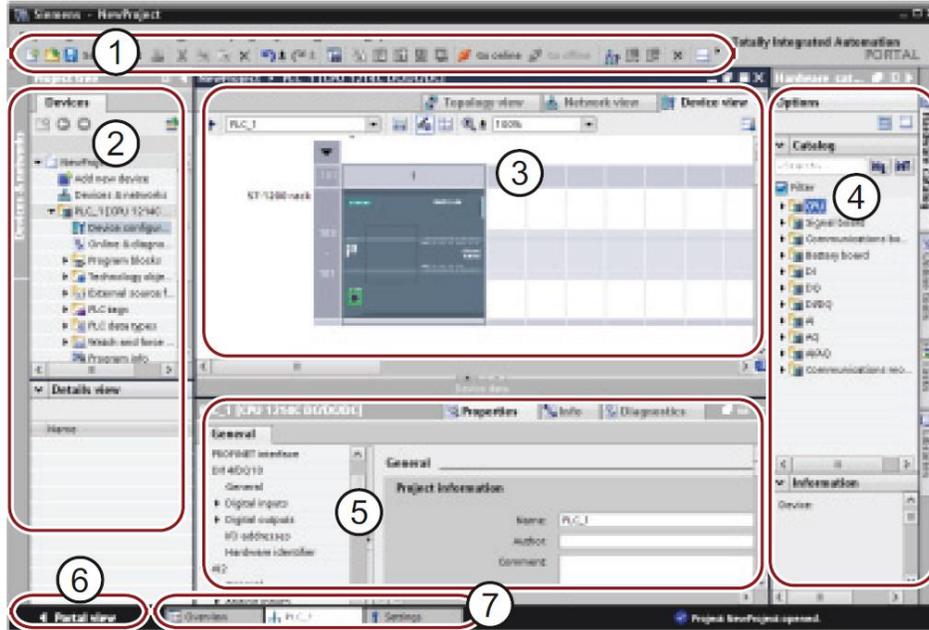


Figura 2.2 - Visualización de la vista de proyecto en S7

2.2 Configuración de hardware

Para empezar a trabajar es necesario crear un nuevo proyecto. La forma más sencilla de hacerlo es desde la **vista de portal**. Al abrir el TIA Portal, la vista de portal es la que aparece por defecto. En esta ventana, en el portal *Iniciar* se selecciona la tarea *Crear proyecto*. En el panel de selección se pedirán los datos del nuevo proyecto como son el nombre, la ruta donde se guardará, la versión en que se configurará el proyecto, el autor y algún comentario del proyecto en cuestión.

⁶ [Vista de proyecto en TIA Portal v15] Siemens. SIMATIC Controlador programable S7-1200. Manual de sistema (2015).

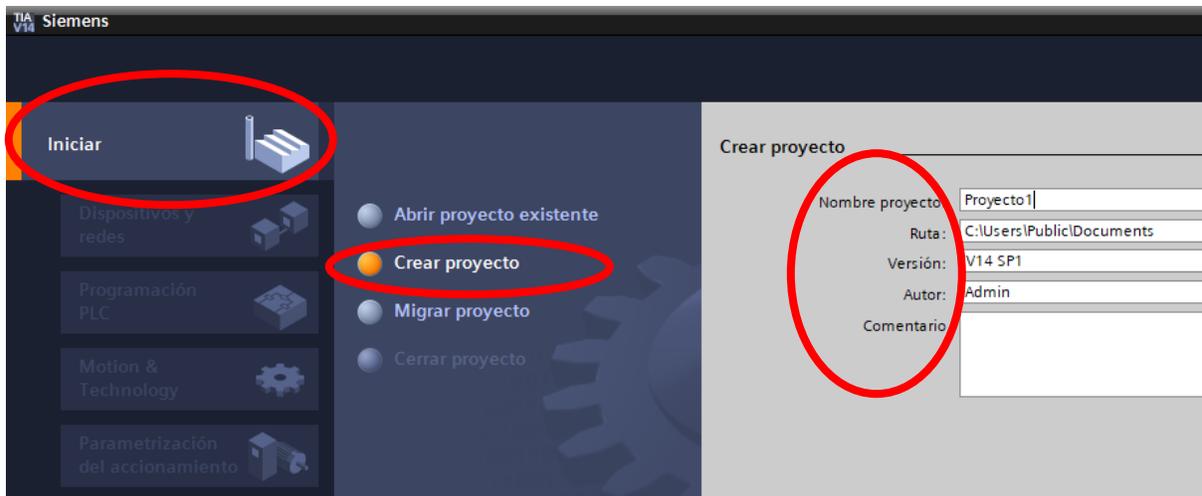


Figura 2.3 - Panel de creación de nuevo proyecto

Para crear la configuración de dispositivos del PLC es necesario agregar un CPU y los módulos adicionales al proyecto.

- En la **vista de portal**, se selecciona el portal *Dispositivos y redes*, y en el área de tareas se selecciona *Agregar dispositivo*.



Figura 2.4 - Tarea del portal *Dispositivos y redes* para añadir un CPU

- En la **vista de proyecto**, bajo el nombre del proyecto, hacer doble clic en *Agregar dispositivo*.



Figura 2.5 - Tarea del portal Dispositivos y redes para añadir un CPU

De cualquiera de las vistas de las que se haya seleccionado la opción de agregar dispositivo, aparecerá una ventana como la que se muestra en la Figura 2.5, a través de la cual se pueden añadir los diferentes tipos de dispositivos con los que se puede trabajar: controladores, HMI, sistemas PC o sistemas de accionamiento.

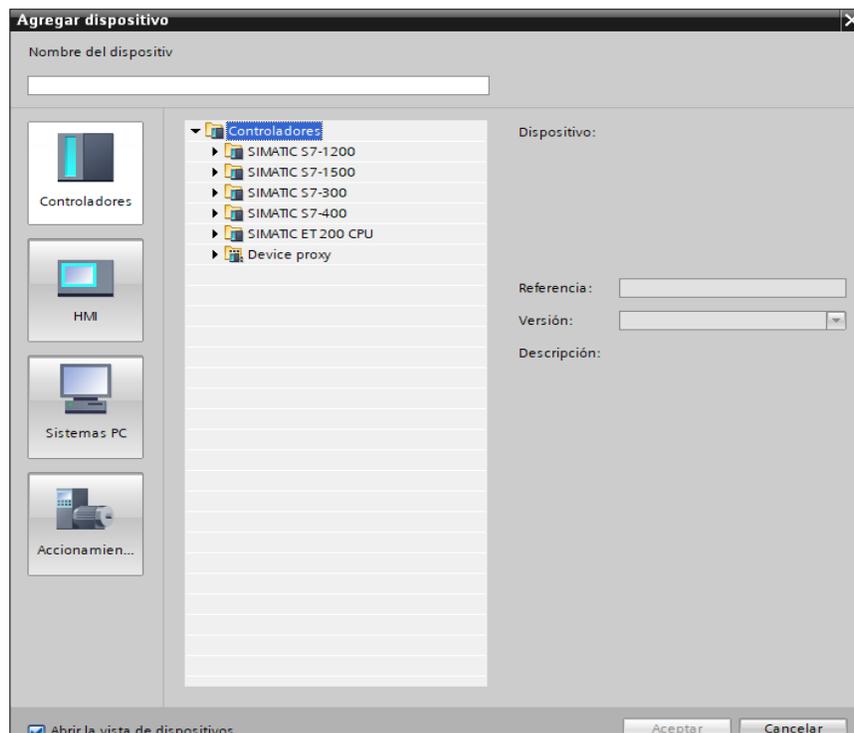


Figura 2.6 - Ventana de selección de dispositivos

Para insertar el CPU con el que se va a trabajar es posible hacerlo de dos formas diferentes:

- Si se conoce el modelo y la versión de firmware se puede añadir el CPU directamente del catálogo, buscando en el listado que aparece en la ventana de dispositivo para los controladores. Al dar doble clic en el dispositivo deseado, en el área de trabajo aparecerán el rack y el CPU listo para ser configurado.

En el caso del CPU que se tiene en el laboratorio, la versión de firmware correspondiente es 6ES7 215-1BG31-0XB0.

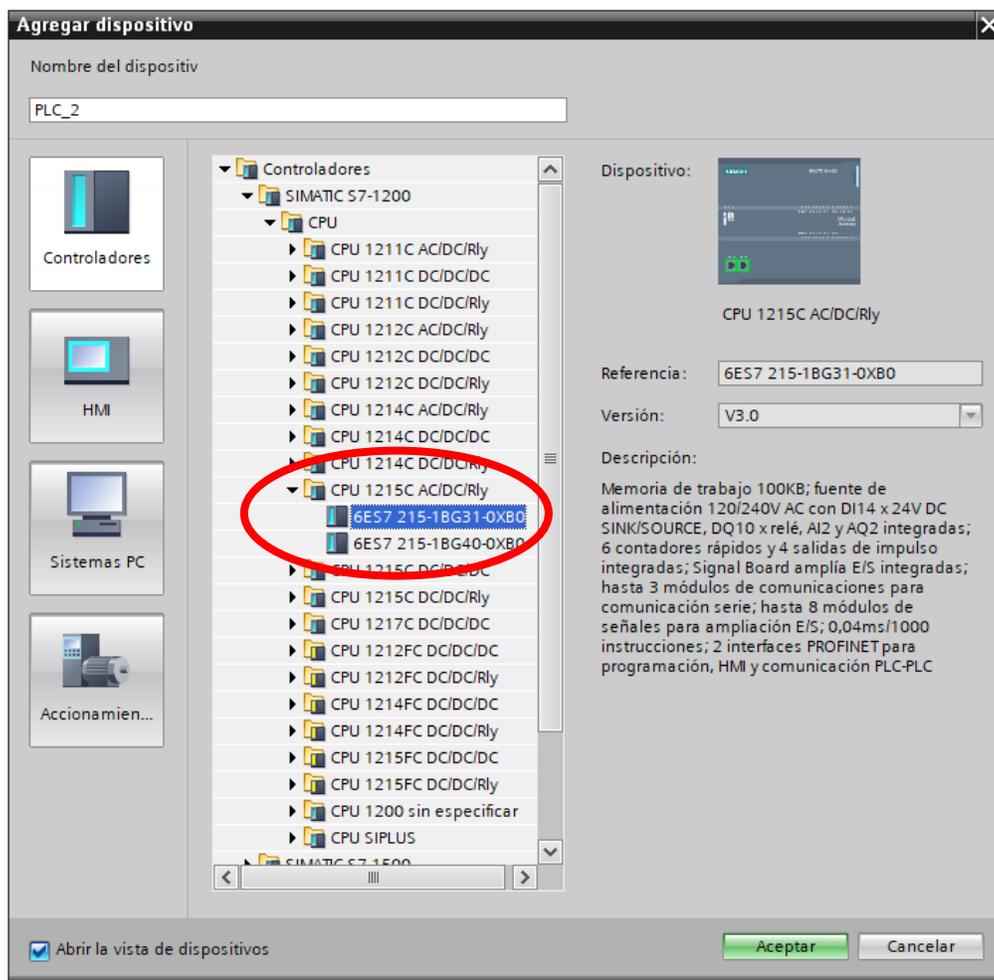


Figura 2.7 - Selección del CPU correspondiente al que se tiene en el laboratorio.

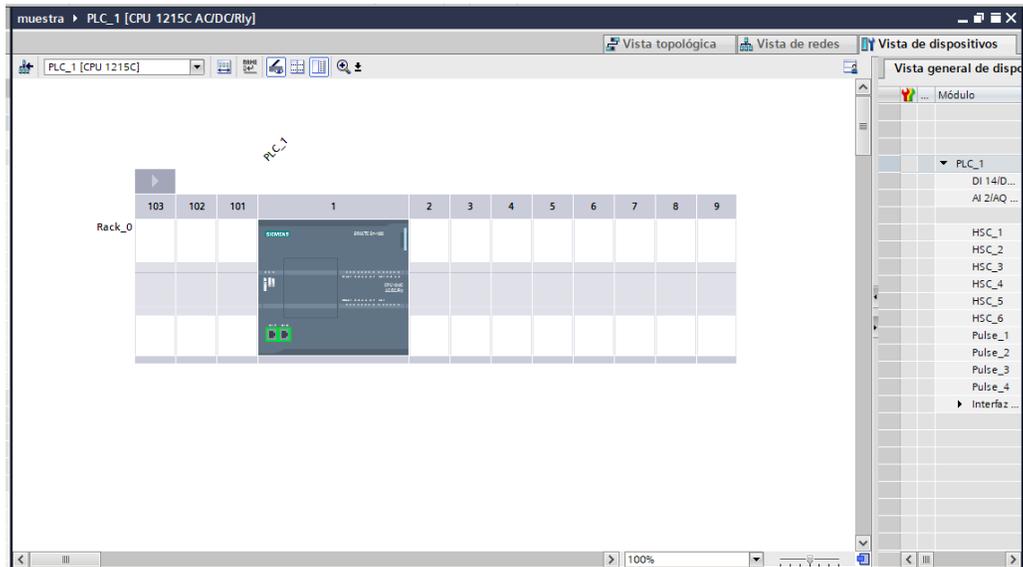


Figura 2.8 - Rack y CPU en el espacio de trabajo tras seleccionar el dispositivo deseado

- Si existe ya una conexión con un CPU es posible añadirlo al proyecto sin conocer previamente el firmware ni el modelo específico (sólo la serie a la que pertenece). Para esto se requiere tener conectado el dispositivo a la computadora y seleccionar, en la ventana de dispositivo, un CPU sin especificar; éste se encuentra al final del catálogo de la serie de CPU.

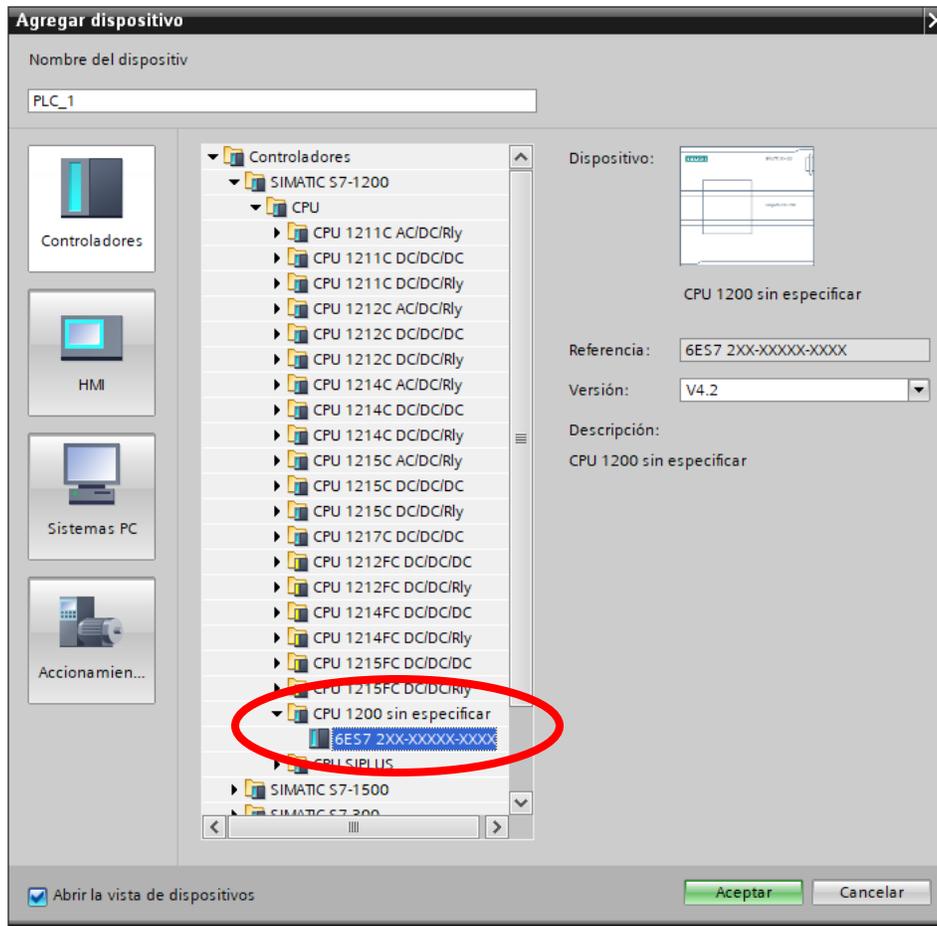


Figura 2.9 - CPU sin especificar en el catálogo de series 1200

En el área de trabajo aparecerá un CPU vacío y sin especificar. En la parte inferior aparecerá un recuadro amarillo (ver Figura 2.10) con el que se puede hacer la configuración del dispositivo. En dicho recuadro se da clic a la opción *determinar la configuración del dispositivo conectado*.



Figura 2.10 - CPU sin especificar en el área de trabajo

Se abrirá una ventana donde se muestran todos los equipos conectados actualmente a la red. Entonces se selecciona aquel que se quiera agregar. Tras seleccionar el CPU en el cuadro de diálogo se pulsa el botón de *Detección* y automáticamente el software carga la configuración de hardware del CPU seleccionado, incluidos todos los módulos que puedan estar conectados. Ya pueden configurarse los parámetros de los dispositivos.

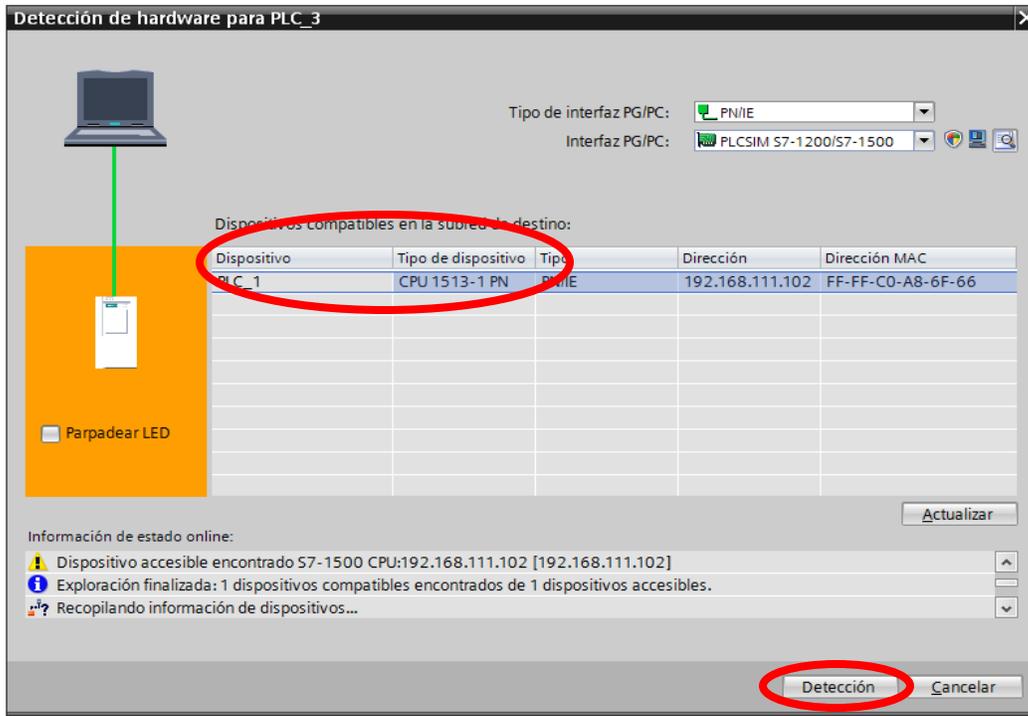


Figura 2.11 - Ventana de selección de los CPU detectados en la red

En la parte derecha del espacio de trabajo se encuentra la ventana de *Catálogo de hardware*, dentro de la cual se encuentra la lista de los diferentes módulos que pueden ser añadidos al PLC. En el caso del laboratorio se cuenta con el módulo adicional DI 16xVCD/DQ 16xRelay, el cual aparece en la carpeta DI/DQ del catálogo. Se puede, ya sea dar doble clic al nombre del dispositivo, o bien, arrastrarlo directamente a algún *slot* del rack en el área de trabajo. Con esto, el módulo está listo para ser empleado junto con el PLC.

Es importante tener en consideración que los módulos de comunicación se insertan en los *slots* del lado izquierdo del PLC; mientras que los módulos de señales se conectan a la derecha.

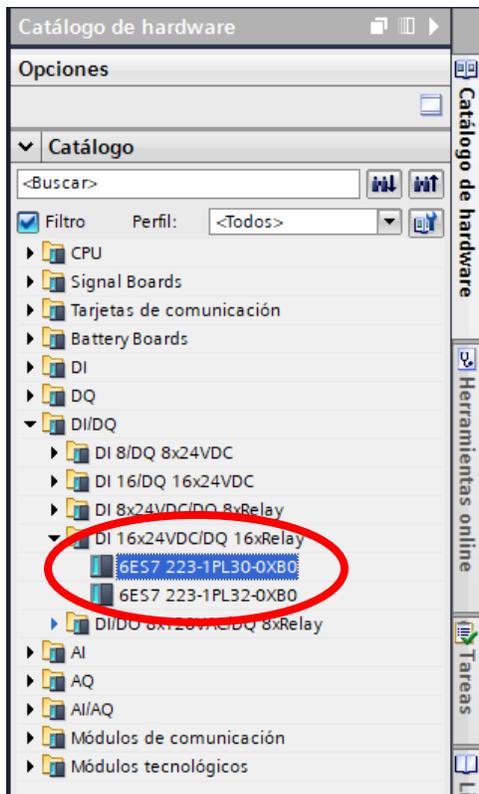


Figura 2.12 - Ventana de catálogo de hardware

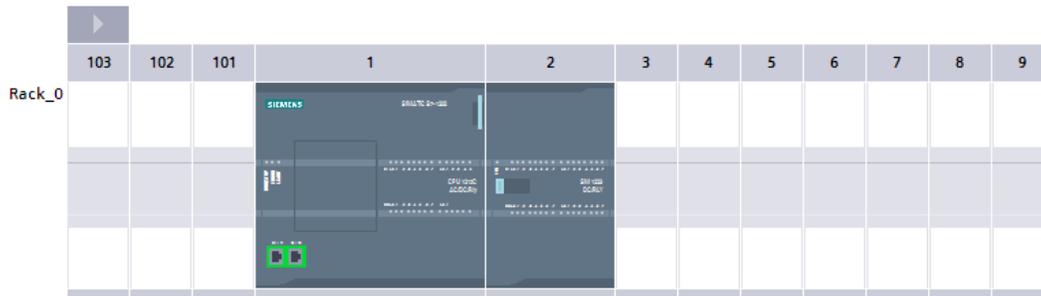


Figura 2.13 - Módulo E/S conectado al rack del CPU

2.3 Configuración de software

Una vez que se han añadido los dispositivos con los que se va a trabajar, es necesario configurar algunos parámetros operativos del CPU. Para visualizar las opciones de configuración es necesario seleccionar el CPU correspondiente y, en la ficha de trabajo, se debe seleccionar la pestaña *Propiedades*, con lo que se desplegará la ventana de configuración como se muestra en la Figura 2.14

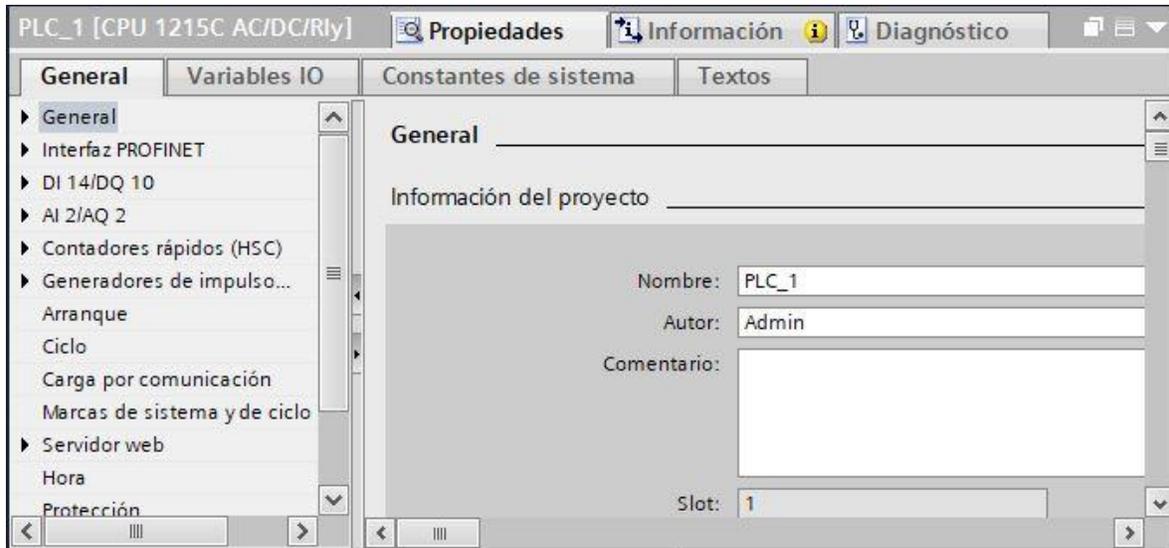


Figura 2.14 - Ventana de configuración de parámetros del CPU

Los parámetros de operación que se pueden configurar desde esta vista son:

- **Interfaz PROFINET:** Permite configurar la dirección IP del CPU para usarse en una red PROFINET; así mismo, se puede configurar la sincronización horaria ya sea con una computadora o con una red previamente formada.
- **DI, DO, AI:** Aquí se configura la reacción de las E/S locales (integradas al CPU); es decir, se pueden modificar tiempos de filtro para las entradas discretas, o bien, la respuesta de las salidas ante una parada del CPU.

- **Contadores rápidos (HSC) y generadores de señales:** Desde esta opción se puede habilitar la configuración de los contadores rápidos (HSC por sus siglas en inglés). También se pueden configurar las salidas del CPU para su utilización como generadores de señales, los cuales son empleados en las operaciones de trenes de pulso (PTO) y de modulación de ancho de pulso (PWM).
- **Arranque:** El CPU tiene tres estados operativos: STOP, ARRANQUE y RUN; éstos se discutirán a mayor detalle en la sección 2.6. En esta pestaña de configuración se pueden definir tres eventos del arranque:
 - o *Arranque tras POWER ON:* Permite configurar la reacción del CPU ante una transición de OFF a ON.
 - o *Compatibilidad de hardware soportada:* Configura la estrategia de sustitución para todos los componentes del sistema.
 - o *Tiempo de asignación de los parámetros para E/S distribuidas:* Configura un tiempo máximo para que las E/S distribuidas pasen a conmutarse online, es decir, queden disponibles. Esto debido a que los módulos de comunicación reciben la alimentación y los parámetros de comunicación del CPU durante el arranque. El tiempo de asignación permite que las E/S conectadas a dichos módulos cambien a online.
- **Ciclo:** Permite establecer un tiempo de ciclo máximo y/o mínimo para la ejecución del programa principal.
- **Marcas de sistema y de ciclo:** Aquí se puede habilitar un byte para funciones de "memoria de sistema" y un byte para funciones de "marca de ciclo". El tema de marcas se discutirá a profundidad en la [sección 3.2](#)
- **Hora:** Selecciona la zona horaria y configura el horario de verano/invierno.
- **Protección:** El CPU ofrece tres niveles de seguridad para restringir el acceso a determinadas funciones. Al configurar el nivel de protección y la contraseña del CPU, se limitan las funciones y áreas de memoria accesibles sin introducir una contraseña. La configuración permite ajustar la protección de lectura/escritura y la contraseña para acceder al CPU.
- **Recursos de conexión:** El CPU soporta un cierto número máximo de conexiones asíncronas simultáneas para PROFINET y PROFIBUS. En esta pestaña se ofrece un

resumen de las conexiones de comunicación que están disponibles para la CPU y del número de conexiones que se han configurado.

El número de conexiones de comunicación que se pueden tener en un mismo PLC son las siguientes:

- 8 conexiones para OPEN User Communication (protocolo basado en TCP/IP que permite la conexión del PLC con un dispositivo de internet).
- 3 conexiones S7 CPU a CPU como servidor.
- 8 conexiones S7 CPU a CPU como cliente.
- 3 conexiones a HMI.
- 1 conexión a un dispositivo de programación externo.

Es también necesario configurar los módulos adicionales que se vayan a usar con el CPU. Para esto, de la misma manera en que se accedió a la configuración del CPU, basta con seleccionar el módulo en la vista de dispositivos y usar la ficha *Propiedades* de la ventana de inspección.

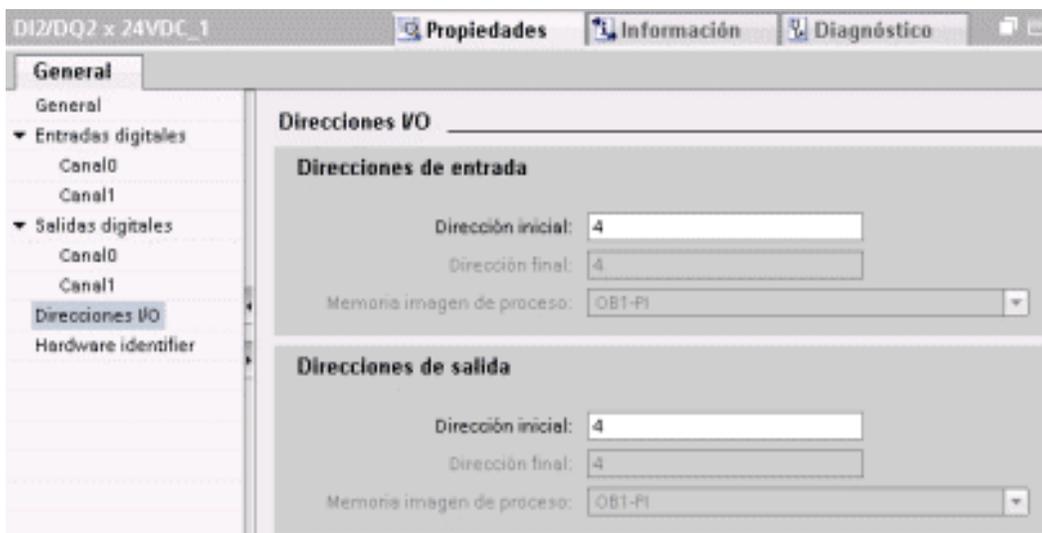


Figura 2.15 - Ventana de configuración de los parámetros de un módulo E/S

Las características que se pueden configurar en los módulos son las siguientes:

- **E/S digitales:** Las entradas pueden configurarse para detectar flancos ascendentes o descendentes; así como para la "captura de pulsos" (esto es, para que permanezcan activadas tras un pulso momentáneo).
- **E/S analógicas:** Es posible configurar los parámetros de las distintas entradas, tales como el tipo de medición (voltaje o corriente) o el rango; así como habilitar el diagnóstico de rebase por defecto o por exceso. Las salidas analógicas ponen a disposición parámetros tales como el tipo de salida (voltaje o corriente) y de diagnóstico, como cortocircuito (para salidas de tensión) o diagnóstico de límite superior/inferior.

2.4 Comunicación entre el controlador y la computadora

⁷Para establecer comunicación entre la computadora y el CPU se requiere de una conexión Ethernet. Para ello es necesario tomar en cuenta que una conexión Ethernet requiere conocer la dirección MAC, la dirección IP y la máscara de subred para cada dispositivo. Debido a ello es necesario asignarle tanto a la computadora como al PLC una dirección IP y una máscara de subred.

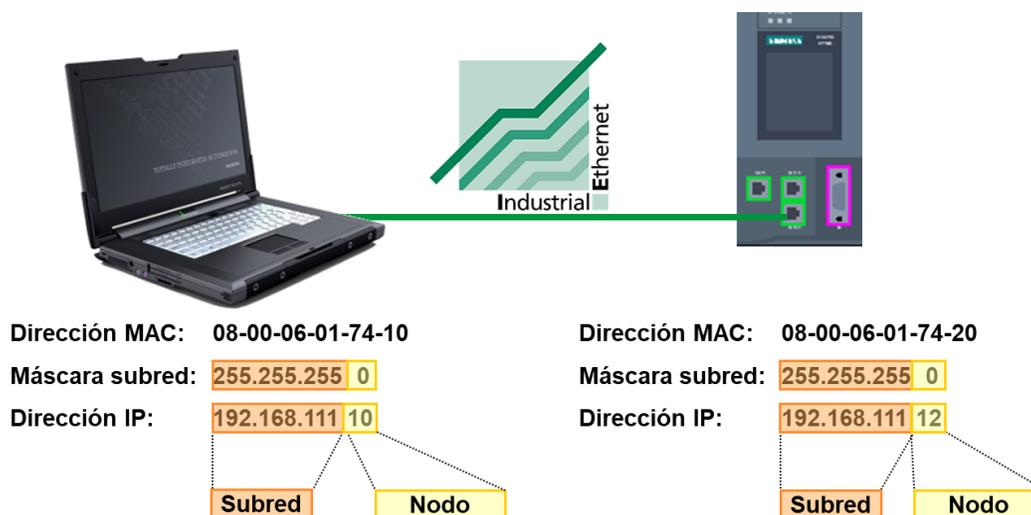


Figura 2.16 - Esquema de una conexión Ethernet (Elaboración propia)

⁷ Siemens. (2018). *TIA Portal V14 SP1. HANDS ON* [Diapositivas de PowerPoint].

Para hacer esta asignación es necesario ir a la ventana de *Propiedades*, misma donde se hace la configuración del CPU. En esta parte se selecciona la opción *Interfaz PROFINET*, luego se da clic en la opción *Direcciones Ethernet* y se colocan los datos de la dirección IP y de la máscara de subred. Mientras el CPU sólo tenga una dirección IP y no se haya cargado ninguna configuración de hardware, siempre se muestra el aviso "No compatible" durante la conexión online.

- **Dirección IP:** La ID de red del CPU y la computadora deben ser exactamente iguales. La ID de red es la primera parte de la dirección IP (los tres primeros octetos) y determina la red IP empleada. Para el caso de los PLC empleados en el laboratorio, las direcciones IP permitidas son 192.168.111.XXX, de donde XXX puede tomar valores entre 120 y 131.
- **Máscara de subred:** La máscara de subred del CPU y de la computadora deben ser exactamente iguales. Normalmente, la máscara de subred tiene el valor 255.255.255.0.

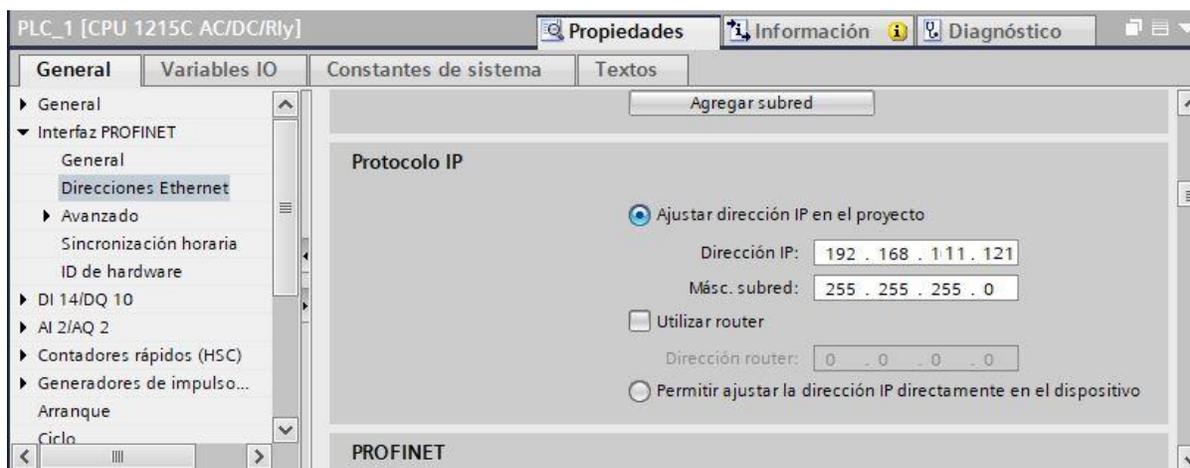


Figura 2.17 - Ventana de configuración de los parámetros de red

Para cargar en el CPU los datos del proyecto, se necesita un enlace *online* entre la computadora y el CPU. Primero, se selecciona en el árbol del proyecto la carpeta (por ejemplo, PLC_1) cuyos datos del proyecto se quieren cargar en el equipo y, del submenú *Online*, se selecciona *Cargar en dispositivo*. También es posible hacer lo mismo haciendo clic en la barra de funciones sobre el icono *Cargar en equipo*. Los datos del proyecto (configuración del equipo y programa de STEP 7) se cargarán entonces al CPU.

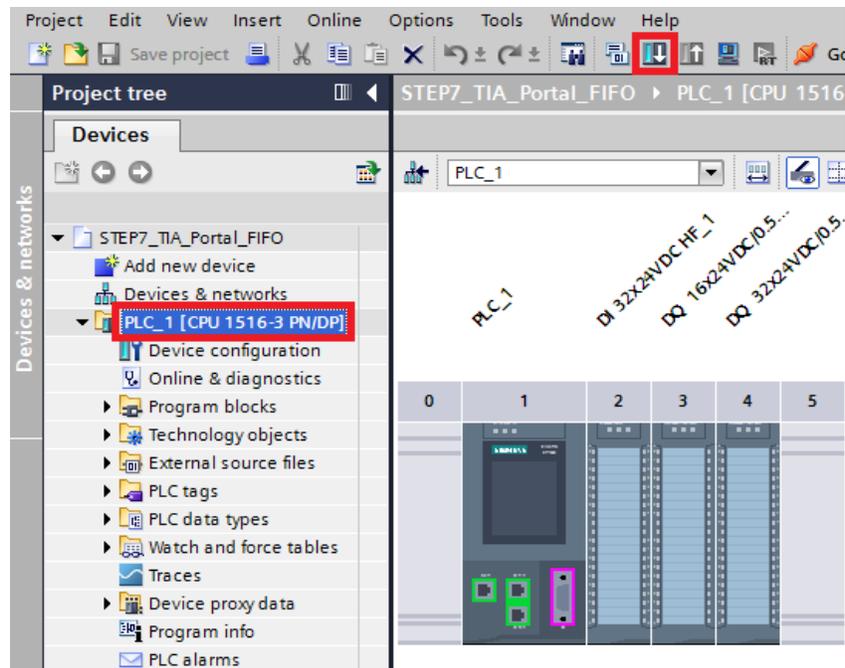


Figura 2.18 - Opciones para cargar un programa al PLC

Así mismo, se tiene la posibilidad de cargar componentes individuales como la configuración de hardware o bloques de funciones, y no necesariamente el proyecto completo. Si se hace clic con el botón derecho del ratón sobre la carpeta "PLC_1" en la navegación del proyecto, en el menú emergente y en la opción *Cargar en dispositivo* se tienen las siguientes opciones:

- *Hardware y software (sólo cambios)*: La configuración del equipo y los bloques software modificados se cargan en el CPU.
- *Configuración hardware*: Sólo se carga en el CPU la configuración de hardware.
- *Software (sólo cambios)*: Se cargan en la CPU todos los bloques de software modificados.

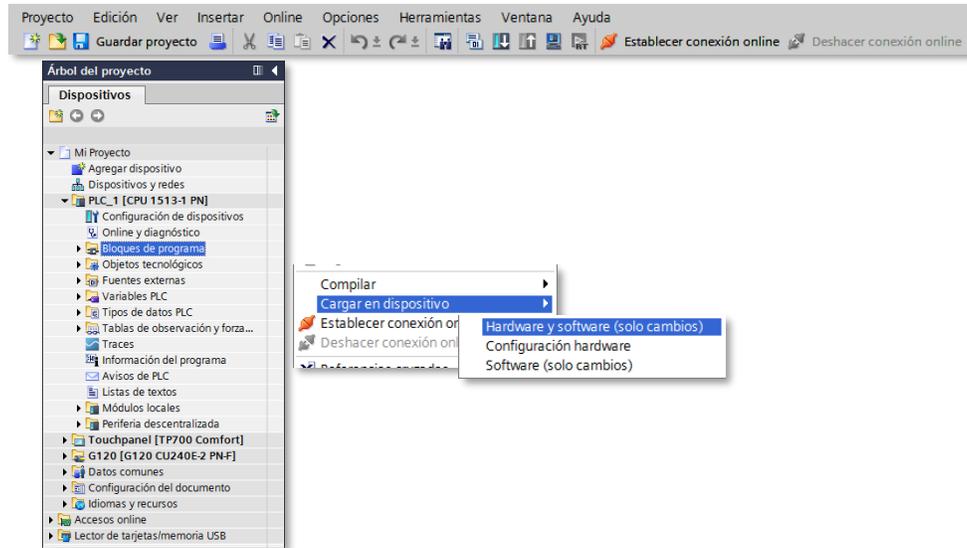


Figura 2.19 - Opciones de carga del programa al PLC

A continuación, se abre la venta de *Carga avanzada*. Aquí se tiene que definir la interfaz PG/IP. Tal como ahí aparece, se debe seleccionar en *Tipo de Interfaz PG/PC* la opción *PN/IE*, con lo que la *Interfaz PG/PC* se identificará automáticamente. Se debe esperar mientras se intenta establecer la comunicación y luego se da clic en la casilla *Mostrar dispositivos compatibles*. Se debe entonces haber reconocido el dispositivo que se ha conectado. Se puede hacer una comprobación seleccionando el dispositivo y activando la casilla *Parpadear LED*, con lo que los 3 LED de estado del PLC en cuestión parpadearán.

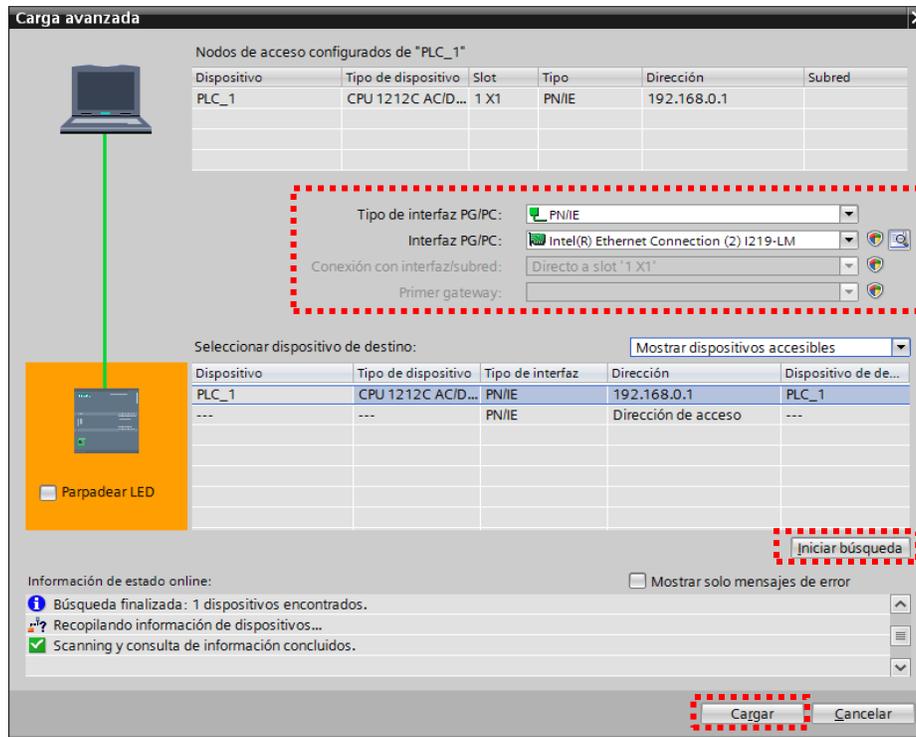


Figura 2.20 - Cuadro de diálogo Carga avanzada

Ahora, se da clic en el botón *Cargar* y se abrirá una ventana de *Vista preliminar de carga*. En el caso de indicarse que existen diferencias entre la configuración del proyecto y la del PLC, debido a que se le está colocando una IP diferente a la que tenía anteriormente, se debe activar la casilla de *Sobreescribir todos* y luego dar clic en *Cargar*. Otro inconveniente puede darse si el PLC se encuentra ejecutando algún programa, en cuyo caso se selecciona como acción la opción de *Parar todos* y finalmente se da clic en *Cargar*.

Si todo aparece correctamente en la *Vista preliminar de carga*, esto es, que todas las casillas tengan un ícono de palomita azul, se puede dar clic en *Cargar*.

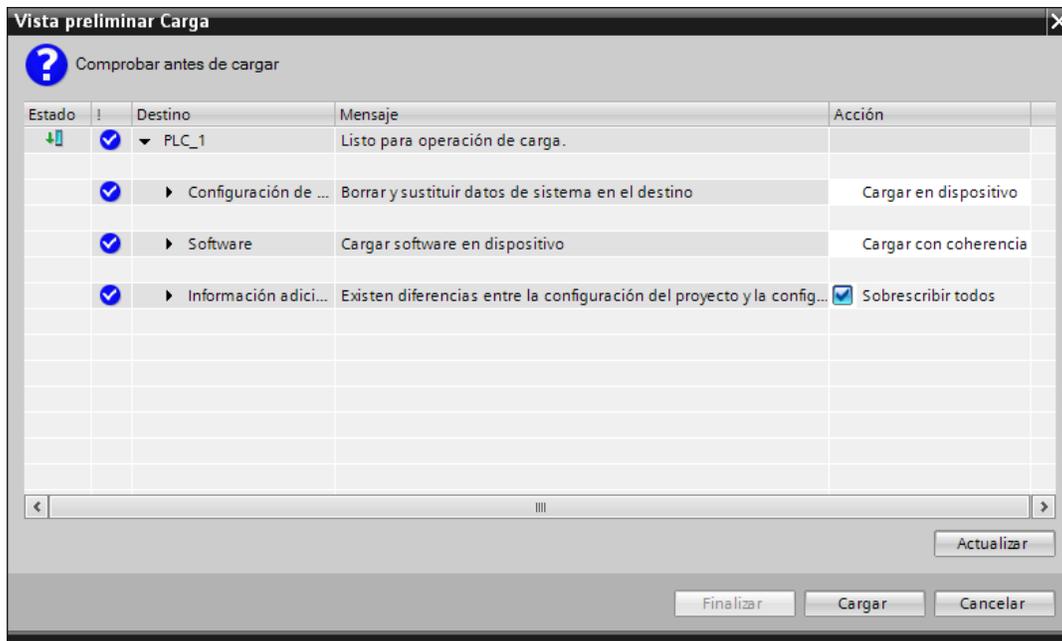


Figura 2.21 - Vista preliminar de carga sin advertencias

Al finalizar de cargar deberá aparecer la ventana de *Resultados de la operación de carga*. Se selecciona entonces la acción *Arrancar todos* para que el PLC se coloque en modo RUN, y para concluir se da clic en *Finalizar*. Si la carga al PLC fue exitosa, en la *Ventana de inspección*, en la pestaña de *Información* y dentro de la pestaña *General* debe aparecer el mensaje de *Operación de carga finalizada (errores: 0; advertencias: 0)*, como se muestra en la Figura 2.23. Además, en el PLC el LED de RUN/STOP debe haber pasado de ser color ámbar a ser color verde.

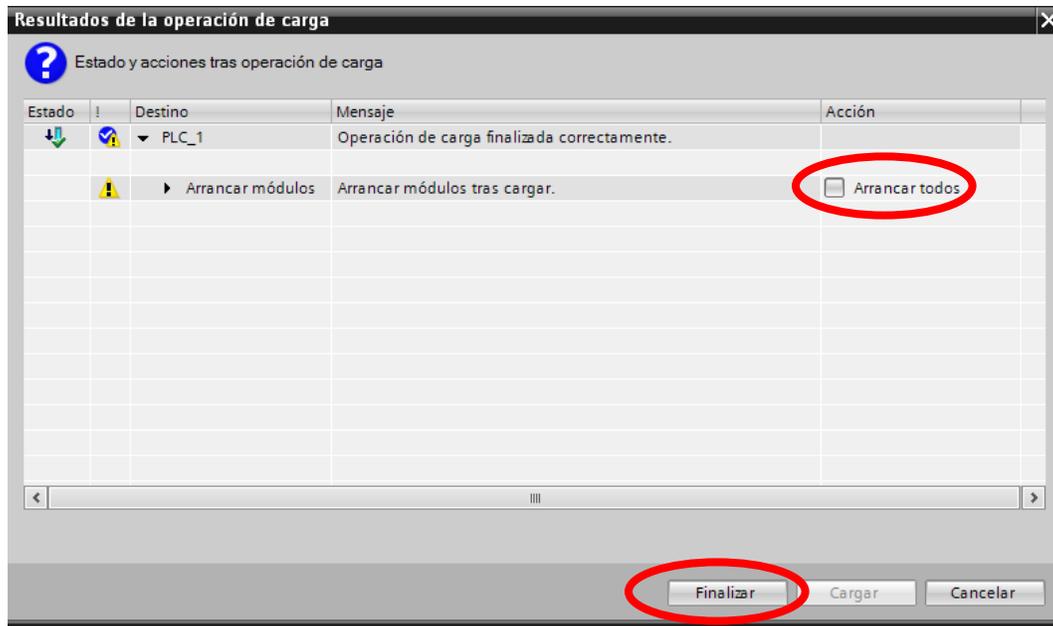


Figura 2.22 - Ventana de Resultados de operación de carga

✓	PLC_1	24/09/2019 12:14:18
✓	Configuración hardware	24/09/2019 12:14:25
✓	La configuración hardware se ha cargado correctamente.	24/09/2019 12:14:44
✓	PLC_1 arrancado.	24/09/2019 12:14:50
✓	'Main' se ha cargado correctamente.	24/09/2019 12:14:42
i	Ha finalizado la búsqueda de dispositivos en la interfaz Intel(R) Dual Band Wireless-AC 8260. .	24/09/2019 12:13:58
✓	Operación de carga finalizada (errores: 0; advertencias: 0).	24/09/2019 12:14:51

Figura 2.23 - Operación de carga exitosa sin errores ni advertencias

2.5 Estructura general de un programa para el S7-1200

La programación del CPU se realiza a través de distintos tipos de bloques lógicos, los cuales permiten estructurar el programa; y, en función de cómo son empleados, dan lugar a diferentes topologías de programación. Cada tipo de bloque tiene un rango de números de bloque que puede admitir, además de una prioridad de ejecución.

Se tienen los siguientes bloques lógicos para el S7-1200:

- **Bloques de organización (OB):** Son los bloques más elementales y son los que definen la estructura del programa. Los OB no pueden ser llamados por ningún otro tipo de bloque, incluyendo otros OB; éstos sólo pueden ser ejecutados por un evento de arranque particular. El tipo fundamental de OB se denomina *De ciclo*. Éstos se

ejecutan de manera cíclica cuando el CPU se encuentra en estado operativo RUN. El bloque OB1 es considerado el bloque principal del programa y contiene el programa principal del usuario. Este bloque tiene la posibilidad de llamar a otros bloques (a excepción de otros OB). Es posible utilizar varios OB de ciclo. Éstos se ejecutan en orden numérico. Dado que el bloque OB1 es el bloque predeterminado, los demás OB de ciclo deben identificarse como OB 200 o superior.

- **Bloques de función (FB) y funciones (FC):** Cada uno de estos bloques tiene parámetros de entrada y de salida que le permiten compartir información con el bloque que los haya llamado. La diferencia entre un FB y un FC reside en que los FB tienen asociado un bloque de datos (DB) de instancia para almacenar los estados durante la ejecución y poder usarlos en otro bloque del programa. Los números de bloque válidos para FB y FC van de 1 a 65535.
- **Bloques de datos (DB):** Almacenan datos que pueden ser empleados por otros bloques del mismo programa. Los números de bloque válidos van de 1 a 65535

Según la forma en que se organice el programa en los distintos bloques lógicos, se tienen los siguientes tipos de programación:

- **Programación lineal:** Todas las instrucciones se encuentran en un bloque de organización (normalmente en el OB1, denominado comúnmente *Main*).
- **Programación dividida por áreas:** En este caso todas las instrucciones y partes del programa se agrupan en funciones individuales, las cuales se encuentran en bloques aislados. Entonces el OB1 se encarga de llamar a los bloques uno detrás del otro.
- **Programación estructurada:** En esta tipología las funciones reutilizables son programadas en bloques individuales. El OB de ciclo u otros bloques pueden llamar a estos bloques para ejecutar tareas determinadas.

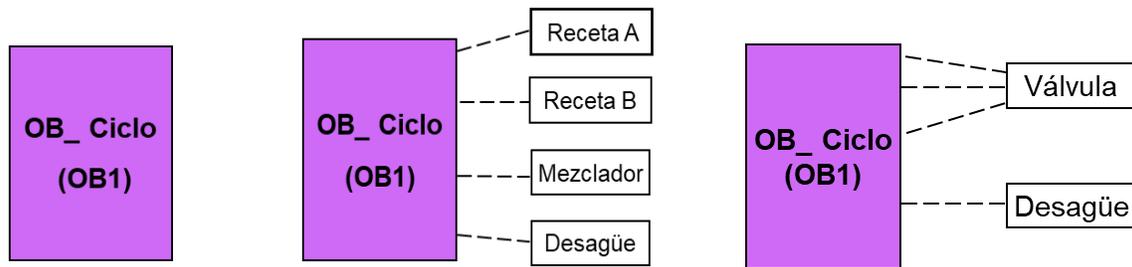


Figura 2.24 - Representación de las diferentes tipologías de programación. Lineal (izquierda), por áreas (centro) y estructurada (derecha)

Para insertar un nuevo bloque es necesario estar en la vista de proyecto. En el *Árbol de proyecto*, debajo de la carpeta del PLC donde se está trabajando, se da doble clic en la flecha izquierda de la carpeta *Bloques de programa*. Si se desea trabajar en el bloque principal se da doble clic en el bloque *Main [OB1]*. Por el contrario, si se desea crear un bloque distinto se da doble clic en la opción *agregar nuevo bloque*.

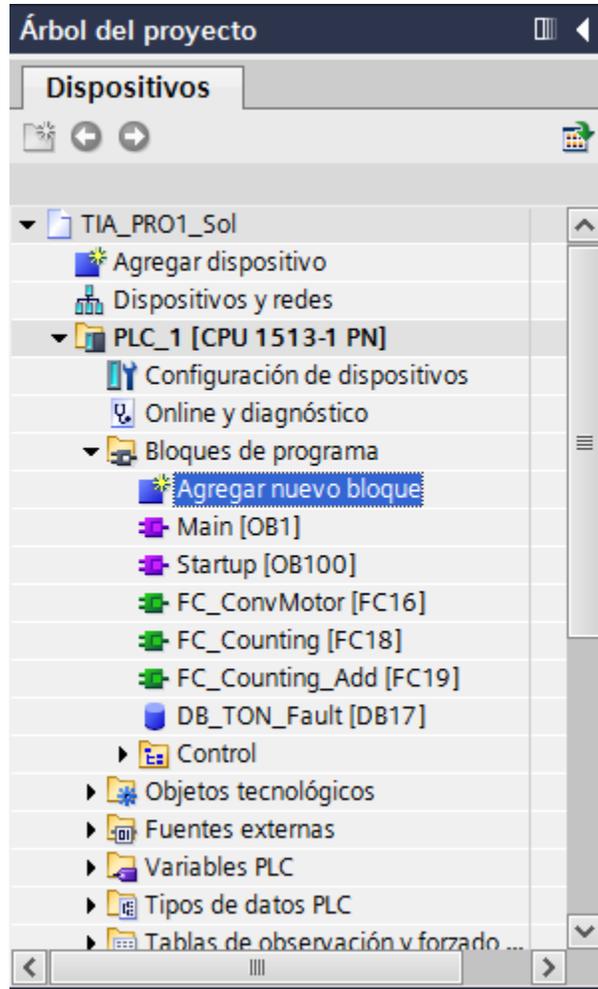


Figura 2.25 - Árbol de proyecto con la opción para añadir nuevos bloques

A continuación, aparecerá la ventana *Agregar nuevo bloque*, mostrada en Figura 2.25. En este recuadro se selecciona el tipo de bloque que se desea crear (OB, FB, FC o DB); así mismo, se le da un nombre para poder identificarlo a través del programa, además del tipo de lenguaje de programación (independientemente de la selección en este momento, es posible modificar el lenguaje de programación del bloque en cualquier momento). También se puede numerar el bloque de función ya sea siguiendo un orden ascendente o alguno personalizado.

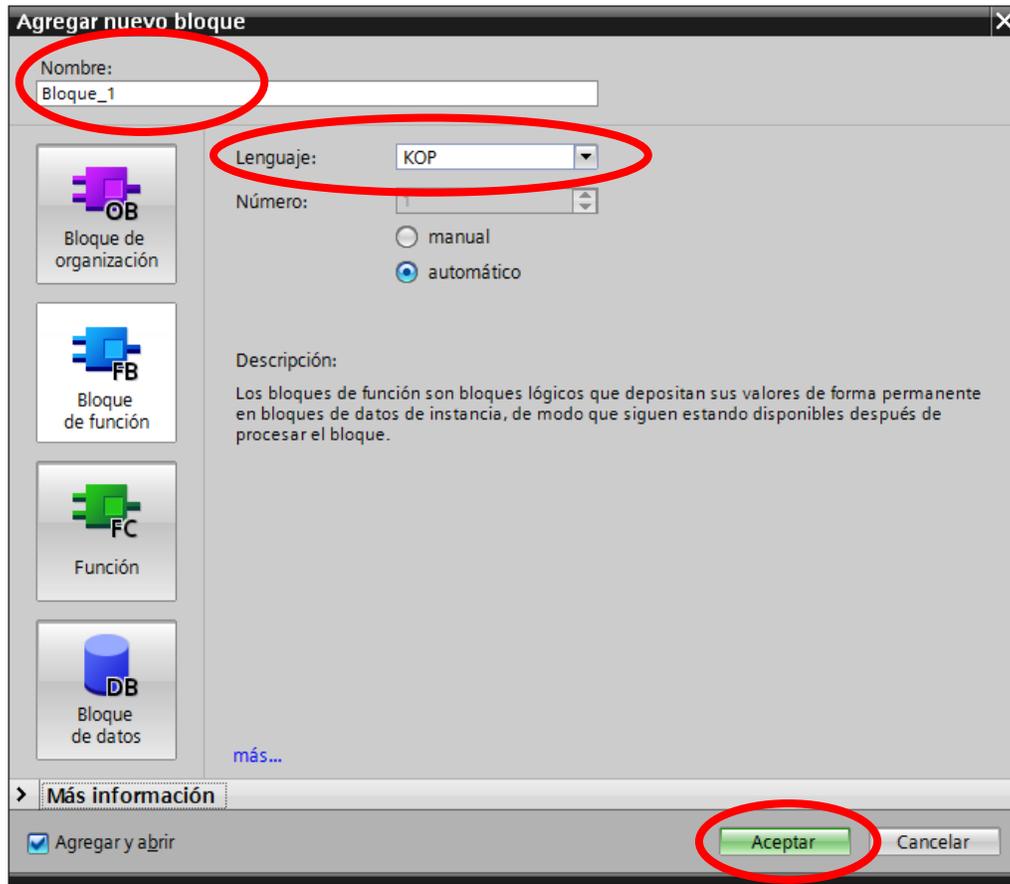


Figura 2.26 - Ventana Agregar nuevo bloque con la respectiva configuración

Una vez que se está ya en el bloque en el que se desea trabajar, aparecerá una ventana en el área de trabajo como la que se muestra en la Figura 2.27. Desde aquí se tienen varias opciones de edición y visualización de la programación del PLC. TIA Portal ofrece la opción de darle títulos a los bloques, además de la posibilidad de comentar cada segmento o escalón de programación, con lo que es posible entonces indicar qué hace cada área del código y se tiene entonces una mejor manera de verificación del contenido, corrección de errores o futuras adaptaciones que se deban hacer a un programa ya hecho. También en esta ventana se tienen las instrucciones más usadas (contactos NA o NC, instrucciones en paralelo, nuevos escalones, etc.), con la posibilidad de expandir el menú añadiendo las opciones favoritas del usuario.

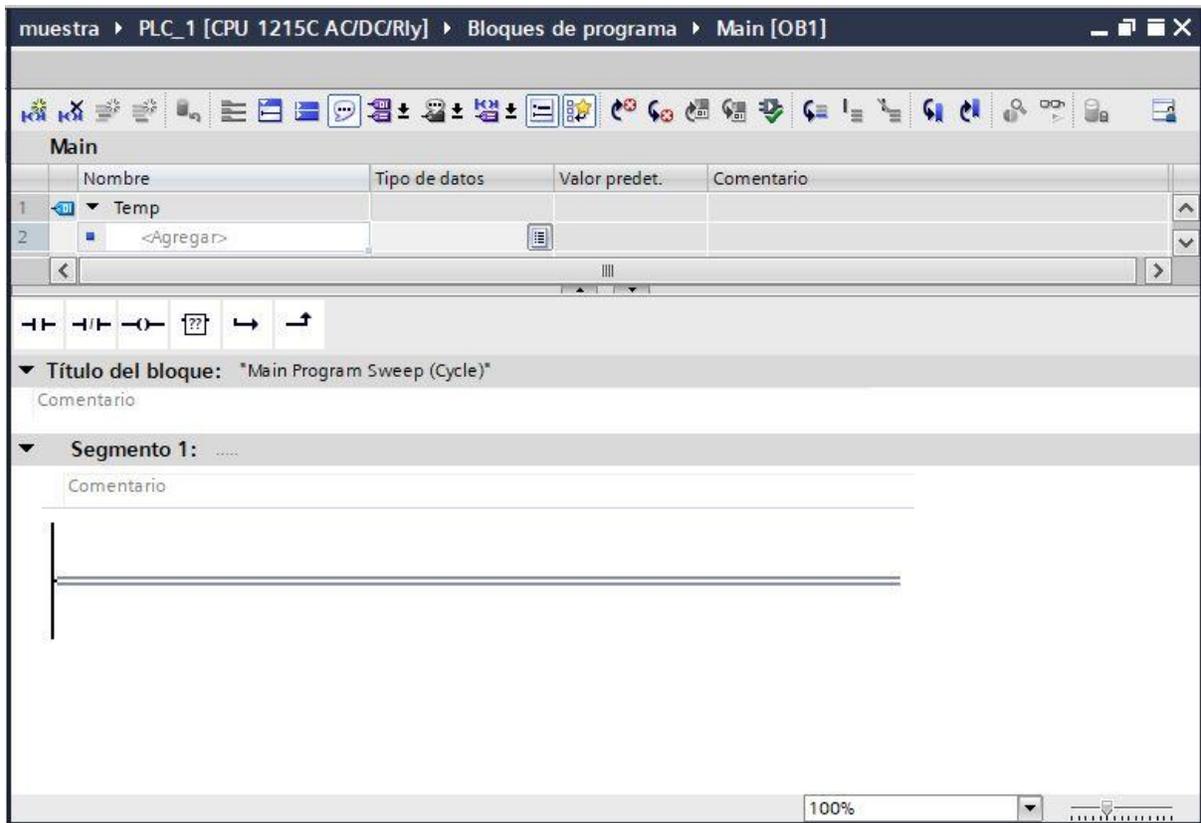


Figura 2.27 - Área de trabajo programación de bloques en TIA Portal

Del lado derecho del área de trabajo se tiene el panel de *Instrucciones*, en el que se encuentran organizadas por carpetas todas las instrucciones con las que se puede programar el PLC S7-1200. Para añadir una instrucción al área de trabajo basta con arrastrarla y, en automático, el programa nos mostrará las posibles posiciones que el bloque de instrucciones puede tomar.



Figura 2.28 - Panel de Instrucciones con la organización de los diferentes bloques de programación

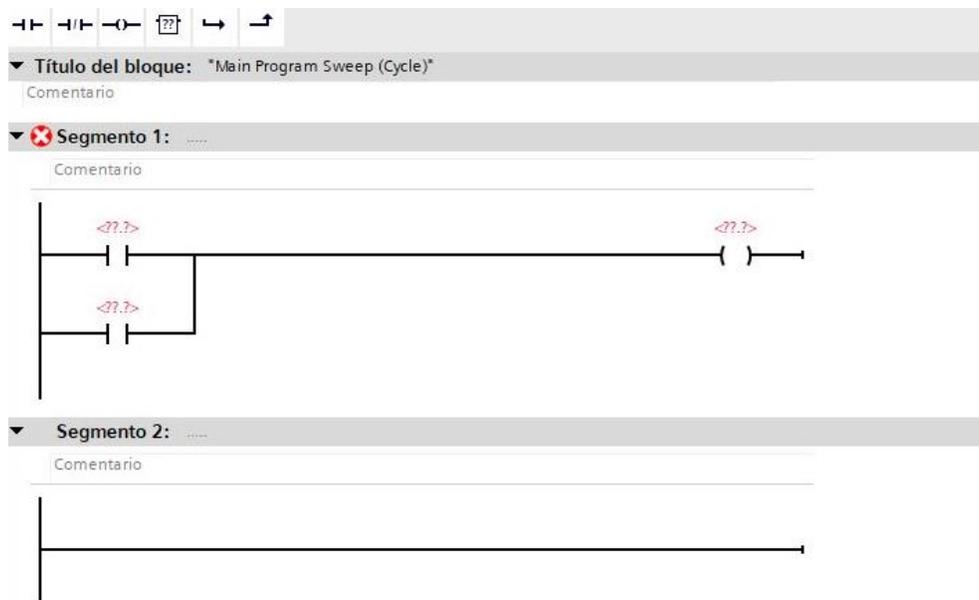


Figura 2.29 - Ejemplo del área de trabajo con algunos contactos en el primer segmento

El siguiente paso es la asignación de las variables de entrada y de salida del PLC a cada contacto. Para esto es necesario dar clic en la flecha junto a la carpeta *Variables PLC* en el *Árbol de proyecto*. A continuación, se hace doble clic en la opción *Agregar tabla de variables*, con esto aparecerá una nueva opción con el nombre por defecto *Tabla de variables_1 [0]*. Al dar doble clic sobre ella se abrirá una ventana (Figura 2.31) donde se puede hacer el registro y configuración de las variables a manejar en el programa.

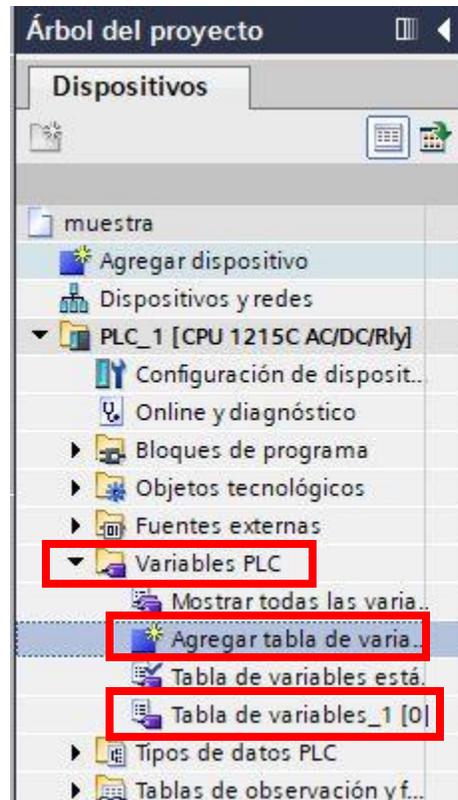


Figura 2.30 - *Árbol de proyecto con la opción de añadir tabla de variables*

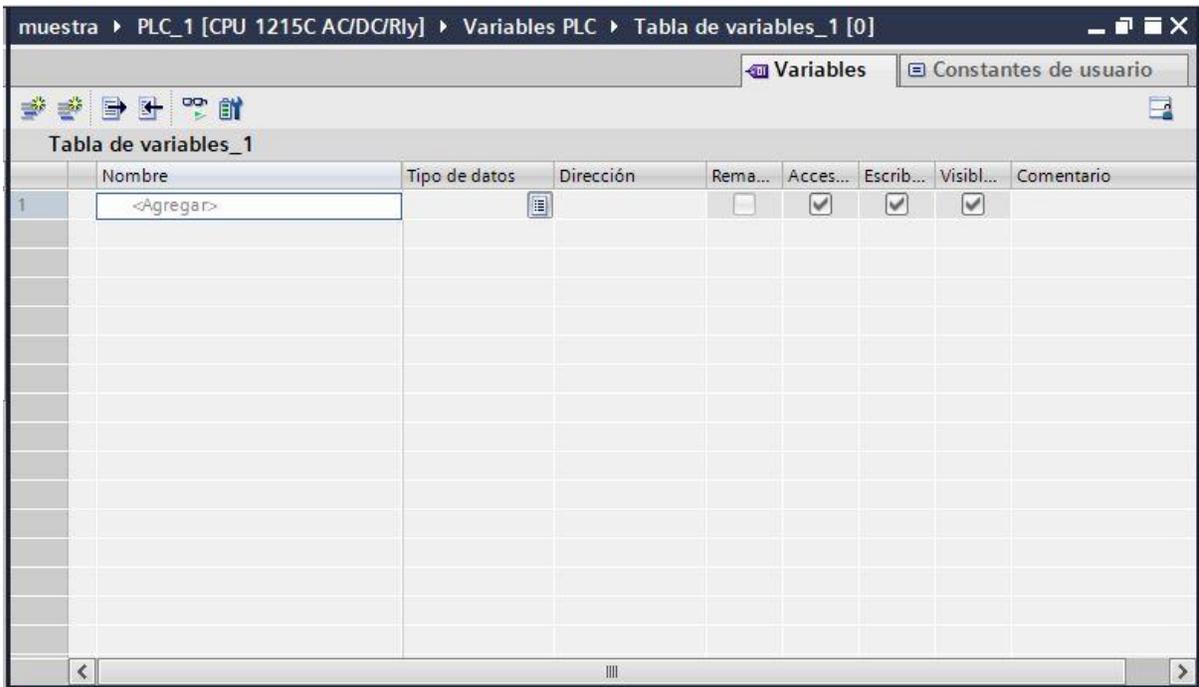


Figura 2.31 - Tabla de variables (en blanco)

Para la definición de variables se deben completar los siguientes campos en la tabla:

- *Nombre*: Etiqueta con la que se reconocerá a la variable. En esencia son nombres simbólicos de las direcciones de memoria de los datos, tanto del PLC como de los módulos I/O adicionales. Los nombres de las variables pueden contener caracteres alfanuméricos, y algunos caracteres especiales como espacios, guiones medios o bajos.
- *Tipo de datos*: Se refiere al tipo y tamaño de información que contendrá la variable. Este tema se discutirá más a fondo más adelante en esta misma sección. *Dirección*: Es el área de memoria global que en específico ocupará la variable, y que se asocia unívocamente con una entrada o salida del PLC o de un módulo adicional. La organización de esta memoria está dada en elementos de entrada (I), salida (Q) y marcas (M). Éstas últimas serán discutidas en la [sección 3.2](#)

	Nombre	Tipo de datos	Dirección	Rema...	Acces...	Escrib...	Visibl...	Comentario
1	Arranque	Bool	%I1.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Botón normalmente abierto
2	Motor	Bool	%Q0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Motor trifásico
3	<Agregar>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figura 2.33 - Ejemplo de variables declaradas en la tabla de variables

El siguiente paso es asociar las variables ya creadas a los contactos correspondientes. Para esto, la manera más sencilla de hacerlo es dar clic en el nombre de la tabla de variables, y en la parte inferior del árbol del proyecto aparecerá una ventana de *Vista detallada* de las variables contenidas en esa tabla. Lo único que resta es arrastrar la variable deseada al contacto correspondiente sobre las marcas <??.?> que se encuentran sobre él. También es posible dar clic en las marcas <??.?> y escribir el nombre de la variable a asignar, o bien, seleccionarla directamente del menú emergente que aparece al dar clic en las marcas.

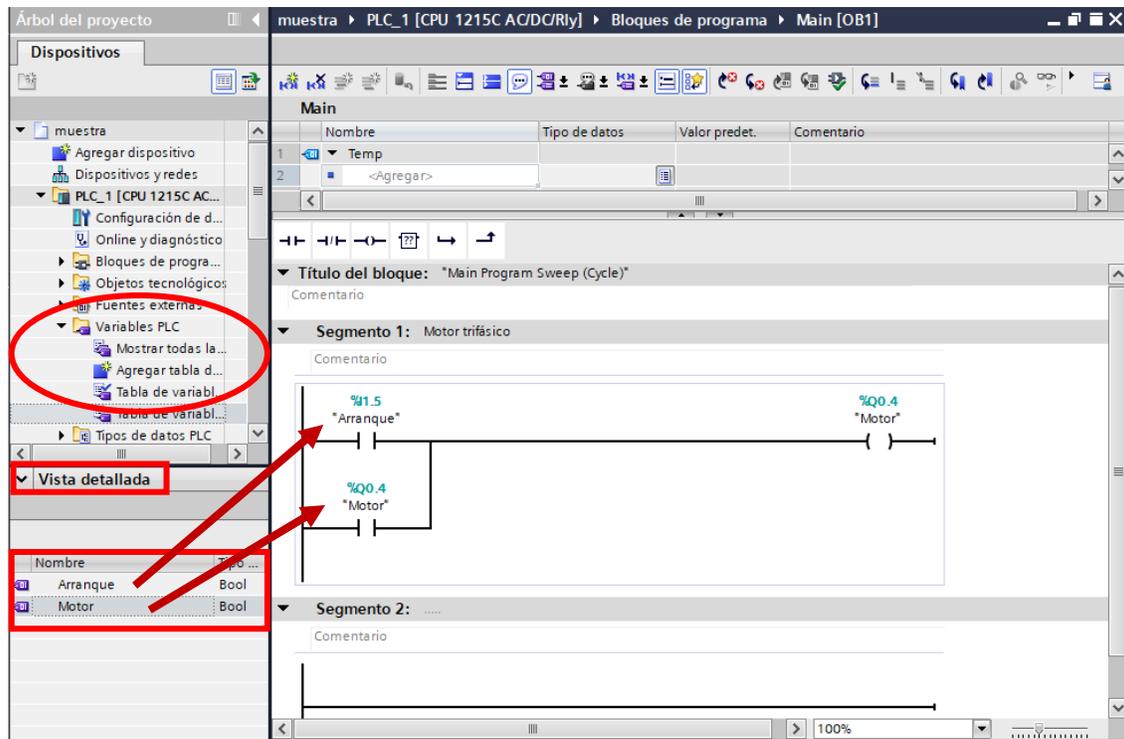


Figura 2.32 - Esquema de contactos con variables asociadas

Como en muchos lenguajes de programación, en el PLC S7-1200 se tienen dos tipos de variables con las que se organizan los datos que se manejan en el programa:

- *Variables globales*: Son datos que son accesibles desde cualquier parte del programa y manejan la información de las entradas, las salidas y las marcas de memoria.
- *Variables locales*: Se refieren a datos que son declarados para su uso dentro de un bloque específico, por lo que no se puede acceder a ellos desde ninguna otra parte del programa; estas variables almacenan información temporal que los bloques emplean para procesar los datos.

En la siguiente tabla se muestran características generales de ambos tipos de variables.

Tabla 2.1 - Características de las variables globales y locales del PLC S7-1200

Característica	Variables globales	Variables locales
Rango de validez	Accesibles desde cualquier parte del PLC El nombre de la variable debe ser único para todo el PLC	Accesible sólo dentro el bloque donde fue declarada El nombre de la variable debe ser único dentro del bloque
Operandos	Entradas Salidas Marcas Variables en bloque de datos Temporizadores/contadores	Variables temporales (en todos los bloques lógicos) Variables estáticas (sólo en FB)
Área de declaración (definición)	Tabla de variables del PLC Bloque de datos globales	Área de declaración del bloque
Representación simbólica	Las variables globales se representan entre comillas (“Variable”)	Las variables locales van precedidas por # (#Variable)

El CPU es capaz de manejar, dentro de las variables tanto locales como globales diferentes tipos de datos, lo que le permite conocer su longitud (tamaño de memoria que ocupa) así como distinguir la forma en que debe ser interpretada. Cada instrucción de programación recibe una cierta cantidad de parámetros de entrada, y si bien algunos parámetros pueden recibir diferentes tipos de dato, si éste no coincide con el que el bloque tiene determinado, la instrucción no se ejecutará correctamente o no se ejecutará en absoluto y se tendrá un error.

Un **parámetro formal** es el identificador en una instrucción que indica la ubicación de los datos que deben utilizarse (ejemplo: la entrada IN1 de una instrucción ADD). Un **parámetro actual** es la posición de memoria o constante que contiene los datos que debe utilizar la instrucción, y se expresan precedidas por el carácter "%" (ejemplo: %MD400). El tipo de datos del parámetro actual definido por el usuario debe concordar con uno de los tipos de datos que soporta el parámetro formal especificado por la instrucción.

En las siguientes tablas se describen los tipos de datos con los que el CPU puede trabajar:

Tabla 2.2 - Tipos de datos bit y secuencia de bits

Tipo de dato	Tamaño en bits	Tipo de número	Rango numérico
Bool	1	Booleano	FALSE/TRUE
		Binario	0 o 1
		Octal	8#0 o 8#1
		Hexadecimal	16#0 o 16#1
Byte	8	Binario	2#0 a 2#11111111
		Entero sin signo	0 a 255
		Octal	8#0 a 8#377
		Hexadecimal	B#16#0 a B#16#FF
Word	16	Binario	2#0 a 2#1111111111111111
		Entero sin signo	0 a 65535
		Octal	8#0 a 8#177777
		Hexadecimal	W#16#0 a W#16#FFFF, 16#0 a 16#FFFF,
DWord	32	Binario	2#0 a 2#11111111111111111111111111111111
		Entero sin signo	0 a 4294967295
		Octal	8#0 a 8#3777777777
		Hexadecimal	DW#16#0000_0000 a DW#16#FFFF_FFFF, 16#0000_0000 a 16#FFFF_FFFF

Tabla 2.3 - Tipos de datos enteros

Tipo de datos	Tamaño en bits	Rango numérico	Ejemplo de constante
USInt	8	0 a 255	78, 2#01001110
SInt	8	-128 a 127	+50, 16#50
UInt	16	0 a 65535	65295, 0
Int	16	-32 768 a 32 767	30000, -128
UDInt	32	0 a 4 294 967 295	4 042 322 160
DInt	32	-2 147 483 648 a 2 147 483 467	-2 131 754 992

De la tabla anterior, conviene señalar la siguiente nomenclatura:

U = sin signo, S = simple, D = doble

En el caso del manejo de números reales, el PLC emplea un método de representación llamado *punto flotante*, el cual es una forma de notación científica que los sistemas computacionales emplean para representar números reales extremadamente grandes y pequeños de forma eficiente y compacta. El S7-1200 representa los números enteros en dos tipos diferentes: números de precisión simple de hasta 6 cifras significativas (Real) de 32 bits, y números de precisión doble de hasta 15 cifras significativas (LReal) de 64 bits.

Tabla 2.4 - Tipos de datos reales

Tipo de datos	Tamaño en bits	Rango numérico	Ejemplo de constante
Real	32	-3.402823e+38 a -1.175 495e-38, ±0, +1.175 495e-38 a +3.402823e+38	123.456, -3.4, 1.0e-5
LReal	64	-1,7976931348623158e+308 a -2,2250738585072014e-308, ±0, +2,2250738585072014e-308 a +1,7976931348623158e+308	12345,123456789e40, 1.2E+40

Para las variables de tiempo que maneja el CPU se tienen las siguientes:

- El tipo *Time* se guarda como entero doble con signo y se interpreta como milisegundos.
- El tipo *Date* se guarda como valor entero sin signo y se interpreta como número de días agregados en formato dd/mm/aaaa.
- El tipo *TOD* se guarda como entero doble sin signo y se interpreta como el número de milisegundos desde media noche para obtener la hora específica del día.
- El tipo *DTL* usa una estructura de 12 bytes para guardar información sobre la fecha y hora.

Tabla 2.5 - Tipos de datos de fecha y hora

Tipo de datos	Tamaño en bits	Rango	Ejemplo de constante
Time	32	T#-24d_20h_31m_23s_648ms a T#24d_20h_31m_23s_647ms Almacenado como: 2.147.483.648 ms hasta +2.147.483.647 ms	T#1d_2h_15m_30s_45ms TIME#10d20h30m20s630ms 500h10000ms
Date	16	D#1990-1-1 a D#2168-12-31	D#2009-12-31 DATE#2009-12-31 2009-12-31
TOD (Hora)	32	TOD#0:0:0.0 a TOD#23:59:59.999	TOD#10:20:30.400 TIME_OF_DAY#10:20:30.400 23:10:1
DTL (Fecha y hora largo)	96 (12 bytes)	Mín.: DTL#1970-01-01- 00:00:00.0 Máx.: DTL#2554-12-31- 23:59:59.999 999 999	DTL#2008-12-16- 20:30:20.250

Las variables de carácter y cadena que maneja el CPU son las siguientes:

- El tipo *Char* ocupa un byte de memoria y guarda un único carácter codificado en formato ASCII.
- El tipo *String* almacena una secuencia de caracteres, cada uno de un byte. Contiene el número de caracteres total y el número de caracteres actual. Este tipo de dato ofrece una capacidad de almacenamiento de hasta 256 bytes, con los que se almacena el número total de caracteres (1 byte), el número de caracteres actual (1 byte) y una cadena de hasta 254 caracteres; de lo anterior se deduce que cada carácter ocupa, como lo indica el tipo *Char*, un byte.

Tabla 2.6 - Tipos de carácter y cadena

Tipo de datos	Tamaño en bits	Rango	Ejemplo de constante
Char	8	Códigos de caracteres ASCII: 16#00 a 16#FF	'A', 't', '@'
String	N + 2 bytes	N = (0 a 254 bytes de caracteres)	'ABC'

Existe también un tipo de dato denominado *Array*, con el cual se pueden manejar arreglos de hasta seis dimensiones que contengan varios elementos del mismo tipo de datos. Las matrices deben seguir las siguientes directrices:

Nombre [ind1_min.. ind1_max , ind2_min.. ind2_max, ...] of <tipo de dato>

- Todos los parámetros de la matriz deben tener el mismo tipo de datos.
- El índice puede ser negativo, pero el límite superior debe ser mayor o igual que el límite inferior.
- Las matrices pueden tener entre una y seis dimensiones.
- Las declaraciones multidimensionales mín..máx están separadas por caracteres coma.
- No es posible crear matrices anidadas ni matrices de matrices.
- El tamaño de memoria de una matriz es igual al tamaño de un elemento multiplicado por el número total de elementos dentro de la matriz.

Tabla 2.7 - Tipo de datos de arreglos y matrices

Tipo de datos	Índice de matriz	Tipo de datos válidos	Reglas para el índice de una matriz
ARRAY	Constante o variable	USInt, SInt, UInt, Int, UDInt, Dint	Límites de valores: -32768 a +32767

2.6 Orden de procesamiento y ejecución de un programa en el S7-1200

El CPU tiene tres estados de operación y según el estado en el que se encuentre serán las tareas que el PLC puede realizar.

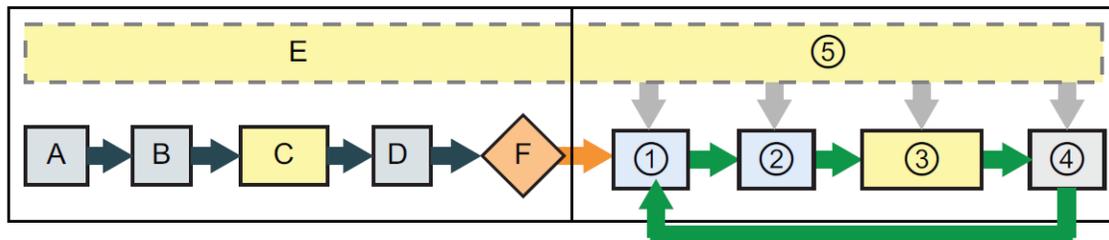
- *STOP*: En este modo el CPU no ejecuta ninguna instrucción. Es cuando se pueden hacer modificaciones al programa cargado, así como cargar y descargar proyectos al dispositivo.
- *ARRANQUE*: Es un estado de transición entre el paro y la ejecución, ocurre cuando, por ejemplo, se energiza el PLC después de estar desconectado de la alimentación. Si existe algún evento de alarma, éste no es procesado.
- *RUN*: Es el modo de ejecución del programa, en el cual se procesan los OB cíclicos del programa. Es en este estado en el que se pueden tomar en cuenta interrupciones, así como el procesamiento de eventos de alarma.

El PLC tiene tres tipos de arranque, los cuales se pueden configurar para cada programa según sean las necesidades:

- Sin arranque: En este caso, el PLC mantendrá el modo STOP en cuanto se conecte a la alimentación. No se ejecutará ningún OB de arranque, en caso de que existan.
- Arranque en caliente – RUN: En esta modalidad, el PLC ejecutará los OB de arranque una vez y pasará automáticamente a modo RUN, con lo que ejecutará el programa de usuario que tenga en memoria.
- Arranque en caliente – Modo de operación antes de desconexión: En este caso, una vez que la alimentación sea conectada el PLC mantendrá el mismo estado (RUN o STOP) que tenía antes de quitarse la alimentación.

Cada vez que el modo operativo cambia de STOP a RUN, el PLC borra las entradas de la memoria imagen de proceso, inicializa las salidas de la memoria imagen de proceso y ejecuta los OB de arranque, así como los FB y FC asociados. El orden de ejecución y transición entre el modo ARRANQUE y el modo RUN se describe en la Figura 2.34

9



ARRANQUE

- A Borra el área de memoria I (imagen)
- B Inicializa las salidas con el último valor o el valor sustitutivo
- C Ejecuta los OBs de arranque
- D Copia el estado de las entradas físicas en la memoria I
- E Almacena los eventos de alarma de la cola de espera que deben procesarse una vez que se haya pasado al estado operativo RUN
- F Habilita la escritura de la memoria Q en las salidas físicas

RUN

- ① Escribe la memoria Q en las salidas físicas
- ② Copia el estado de las entradas físicas en la memoria I
- ③ Ejecuta los OBs de ciclo
- ④ Realiza autodiagnóstico
- ⑤ Procesa alarmas y comunicaciones en cualquier parte del ciclo

Figura 2.34 - Esquema de ejecución de los estados de ARRANQUE y RUN

Una vez que se haya entrado al modo RUN, el PLC ejecuta de manera cíclica el programa de usuario. En cada ciclo, el CPU actualiza las salidas, lee las entradas, actualiza los módulos de comunicación y reacciona a los eventos de alarma de usuario y peticiones de comunicación. Las peticiones de comunicación se procesan periódicamente durante el ciclo. Estas acciones (excepto los eventos de alarma de usuario) se procesan con regularidad y en orden secuencial. Los eventos de alarma habilitados se procesan según su prioridad en el orden en que aparecen. El sistema garantiza que el ciclo se procese dentro de un periodo

⁹ Siemens. (2018). *TIA Portal V14 SP1. HANDS ON* [Diapositivas de PowerPoint].

denominado tiempo de ciclo máximo. De lo contrario, se generará un evento de error de tiempo.

El orden de ejecución de programa para el PLC, conocido comúnmente como *scan*, se describe a continuación:

1. El ciclo inicia con la lectura del estado actual de las salidas discretas y analógicas de la memoria de imagen de proceso de salidas (PIQ, del inglés *Process Image* y Q que representa las salidas). Estos valores se escriben luego en las salidas físicas del PLC. Cuando una instrucción accede a una salida física, se actualizan tanto la memoria imagen de proceso de las salidas como la salida física.
2. Posteriormente se realiza la lectura del estado actual de las entradas discretas y analógicas del PLC. Estos valores se escriben luego en la memoria de imagen de proceso de entradas (PII, del inglés *Process Image* e I que representa las entradas). Cuando una instrucción accede a una entrada física, se modifica el valor de ésta, pero no se actualiza la memoria imagen de proceso de las entradas.
3. Finalmente, el programa de usuario se ejecuta desde la primera hasta la última instrucción. Esto incluye todos los OB de ciclo, así como sus FC y FB asociados. Los OB de ciclo se ejecutan en el orden correspondiente al número de OB, comenzando con el número de OB más bajo. Posteriormente, el ciclo se repite desde el paso 1.

Es importante tener en cuenta las siguientes consideraciones durante la ejecución del ciclo en modo RUN:

- Las comunicaciones se procesan periódicamente durante todo el ciclo, siendo posible que se interrumpa la ejecución del programa de usuario debido a alguna petición externa.
- El autodiagnóstico de errores incluye comprobaciones periódicas del sistema y de estado de los módulos de E/S.
- Los eventos de alarmas pueden ser ejecutados en cualquier momento del ciclo. Cuando ocurre un evento, el PLC interrumpe el ciclo y llama el OB configurado para procesar ese evento. Una vez que el OB haya finalizado el procesamiento del evento, el PLC reanuda la ejecución del programa de usuario en el punto de interrupción.

El tiempo en el que el PLC procesa y ejecuta el ciclo del estado operativo del modo RUN se denomina *tiempo de ciclo*. El CPU ofrece la posibilidad de configurar un tiempo de ciclo máximo y/o mínimo según las necesidades del programador. El CPU monitorea este tiempo de ciclo y reacciona en caso de ser excedido. Si el programa de usuario no incluye ningún OB80, el PLC genera un error y pasa a modo STOP; de existir el OB80, éste es ejecutado.

Tabla 2.8 - Rangos para el tiempo de ciclo

Tiempo de ciclo	Rango [ms]	Valor predeterminado
Máximo	1 a 6000	150 [ms]
Mínimo	1 hasta tiempo de ciclo máximo	No habilitado

Para configurar el tiempo de ciclo se debe estar en la pestaña *General* de las propiedades del CPU, en la opción *Ciclo*. Es en este panel donde se puede configurar el tiempo de ciclo máximo y habilitar el tiempo de ciclo mínimo de ser necesario.

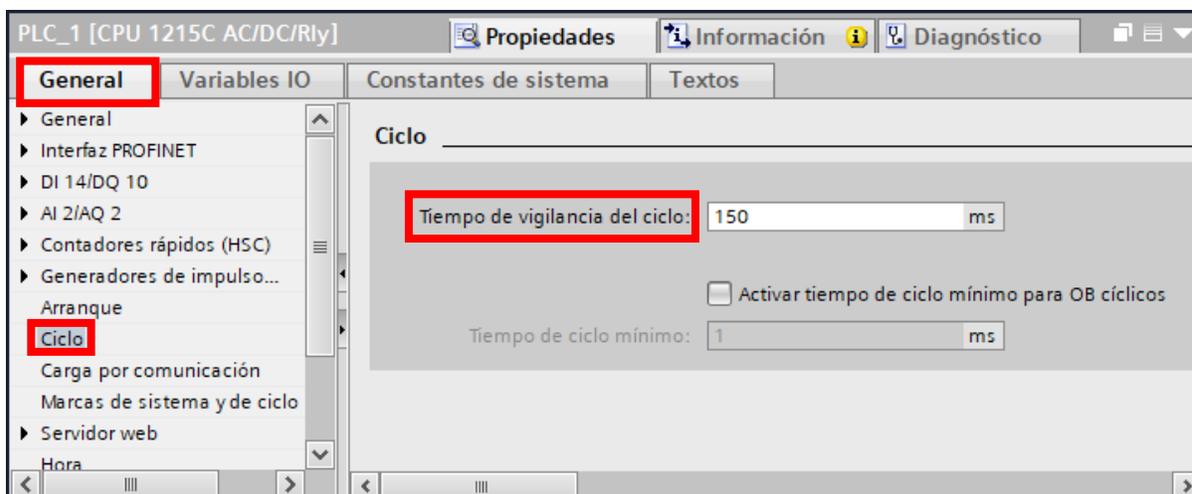


Figura 2.35 - Panel de configuración del tiempo de ciclo

Finalmente, se puede entonces resumir la ejecución del programa de usuario con el esquema presentado en la Figura 2.36

10

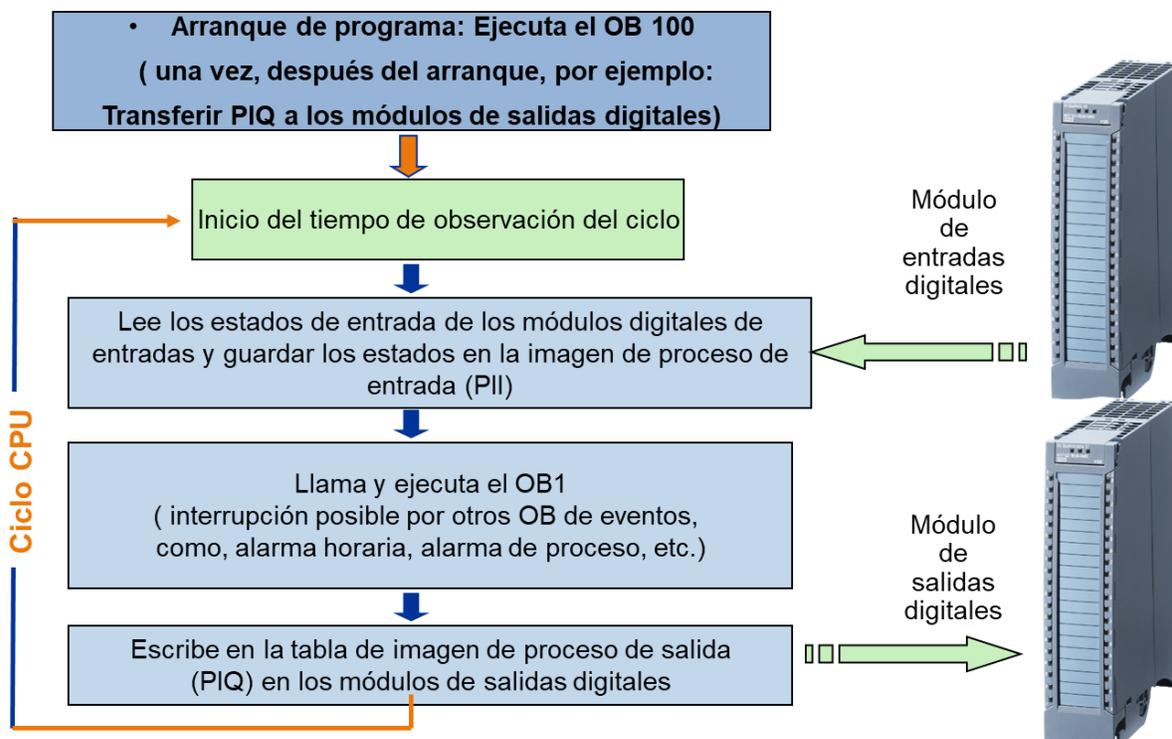


Figura 2.36 - Resumen del ciclo de ejecución del programa de usuario

3. Programación del PLC S7-1200 mediante TIA Portal V15

El software TIA portal ofrece la posibilidad de programar en distintos lenguajes de programación según sean las necesidades inmediatas; o bien, en función del tipo de instrucciones que se desean implementar.

Para comenzar a programar es necesario estar en la ventana del bloque en el que se desea trabajar. Una vez ahí, del lado derecho aparecerá la ventana de *Instrucciones*. Aquí se encuentran organizadas por categorías todas las funciones con las que cuenta el PLC. A su vez, en cada categoría se encuentran agrupadas por carpetas funciones de un mismo tipo.

¹⁰ Siemens. (2018). *TIA Portal V14 SP1. HANDS ON* [Diapositivas de PowerPoint].

Los elementos más básicos para la programación en escalera se encuentran en la pestaña de *Instrucciones básicas*, en la carpeta *General*. Aquí se encuentran las opciones para añadir un nuevo segmento (peldaño), abrir y cerrar ramas para introducir instrucciones en paralelo; o bien, añadir entradas para los bloques que admiten múltiples parámetros de ingreso.

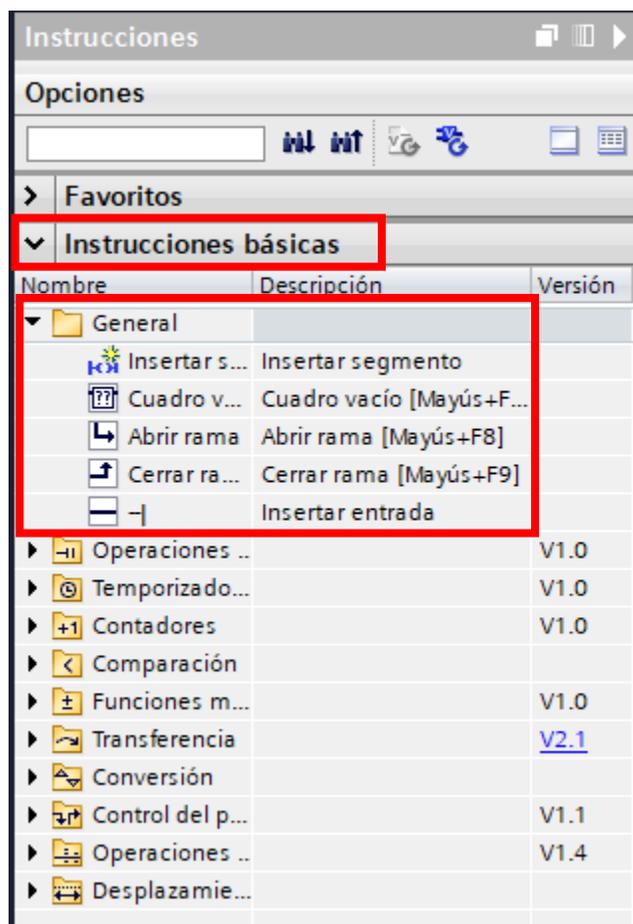


Figura 3.1 - Instrucciones generales para la construcción de un diagrama de escalera

3.1 Entradas y salidas discretas

Los **contactos normalmente abiertos (NA)** y **normalmente cerrados (NC)** son los elementos fundamentales para la programación con lógica de escalera. Los contactos se pueden conectar a otros contratos en serie para implementar una condición *and*; o bien, en paralelo para tener condiciones *or*. Un contacto NA se cierra cuando el valor del bit asociado es igual a 1. De manera análoga, un contacto NC se abre cuando el valor del bit asociado es 1. Ambos tipos de contactos se encuentran en la carpeta *Operaciones lógicas con bits*.

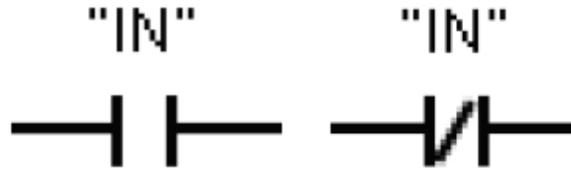


Figura 3.2 - Contacto NA (izquierda) y NC (derecha) en TIA Portal

Las **bobinas de salida**, llamadas también *asignaciones*, permiten escribir valores de bit de 0 o 1 en las salidas correspondientes en la memoria de imagen de proceso, con lo que se pueden activar o desactivar los actuadores que se encuentren conectados a las salidas físicas del PLC. En el caso de los PLC S7-1200 se tienen dos tipos de salidas: las bobinas *directas*, las cuales se ponen en 1 cuando son activadas, y en 0 cuando no; y las bobinas *invertidas*, que funcionan de manera contraria asignando un 0 cuando son activadas y un 1 cuando no. Ambos tipos de bobinas se encuentran en la carpeta *Operaciones lógicas con bits*.

12

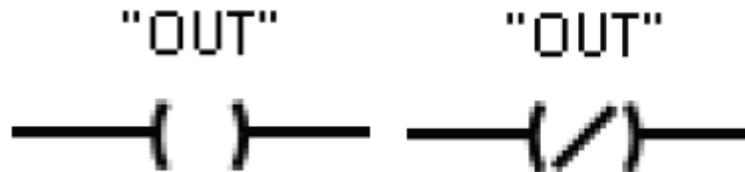


Figura 3.3 - Bobina directa (izquierda) y bobina invertida (derecha)

En TIA Portal se tiene la opción de usar instrucciones para mantener activa una salida sin necesidad de usar enclavamientos, estas instrucciones son exactamente equivalentes a las instrucciones *Set* y *Reset* en algunos PLC, o a las instrucciones *Latch* y *Unlatch* en otros PLC. La instrucción **Activar salida** asigna un 1 a la dirección de salida asociada, y ésta mantiene dicho estado aún si la instrucción ya no está activa. Para desactivar entonces la salida asociada es necesario usar la instrucción **Desactivar salida** para asignar un 0 en la dirección. Ambas instrucciones se encuentran en la carpeta *Operaciones lógicas con bits*.

¹¹ [Imagen de un contacto NA y NC] Siemens. SIMATIC Controlador programable S7-1200. Manual de sistema (2015).

¹² [Imagen de una bobina NA y NC] Siemens. SIMATIC Controlador programable S7-1200. Manual de sistema (2015).



Figura 3.4 - Activar salida (izquierda) y desactivar salida (derecha) en TIA Portal

En el PLC S7-1200 se tiene también la posibilidad de tener entradas y salidas que se ejecuten durante un único *scan*. Estas instrucciones son llamadas genéricamente *flancos*. Un **flanco positivo** o **ascendente** es aquel que se activa cuando una señal pasa de un nivel bajo (0, OFF) a un nivel alto (1, ON); mientras que un **flanco negativo** o **descendente** se activa cuando la señal cambia de un nivel alto a un nivel bajo. Se tienen diferentes formas de manejar los flancos:

- *Detección de flanco ascendente/descendente de un operando*: El estado de este contacto cambia a 1 cuando se detecta un cambio de flanco ascendente o descendente (dependiendo del tipo de consulta) en el bit de **entrada** asociado. Se trata entonces de una lectura de un *scan*. En esencia, su funcionamiento es similar al de los contactos NA y NC con la diferencia que la señal dura sólo un *scan*.

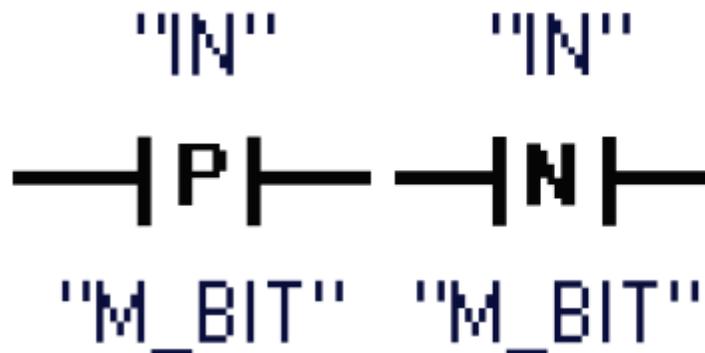


Figura 3.5 - Consulta de flanco ascendente (izquierda) y flanco descendente (derecha)

¹³ [Imagen de bobina S y bobina R] Siemens. SIMATIC Controlador programable S7-1200. Manual de sistema (2015).

- *Activar operando con flanco de señal ascendente/descendente*: El estado de la salida de esta bobina cambia a 1 cuando se detecta un flanco ascendente o descendente (dependiendo del tipo de consulta) en el bit de **salida** asociado, esto es, en la continuidad del peldaño donde se encuentra. Se trata entonces una salida de un *scan*.

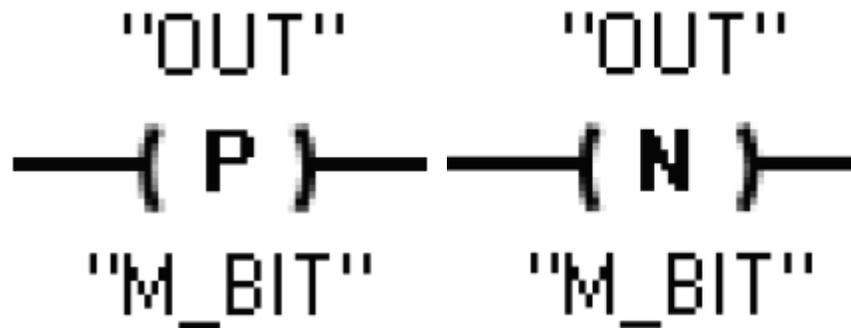


Figura 3.6 - Activación por flanco ascendente (izquierda) y activación por flanco descendente (derecha)

3.2 Memorias o marcas

Las **memorias** o **marcas** son espacios libres en la memoria interna del PLC. El área de marcas (abreviada como M) se emplea para el manejo de relevadores internos, éstos almacenan y manejan el estado intermedio de una operación y demás información de control. Es importante remarcar entonces el hecho de que son entradas y salidas “virtuales”, no reciben ni escriben información de las entradas físicas ni sobre los actuadores. Entre sus aplicaciones están el almacenamiento de resultados de combinaciones de entradas o salidas; o bien, el uso como banderas de proceso.

Las marcas, al ser espacios de áreas de memoria, únicamente pueden tomar valores booleanos, es decir, 0 o 1. Si se requiere guardar una mayor cantidad de información es posible almacenar conjuntos de bits; esto es, se pueden tener marcas del tamaño de un bit, un byte (8 bits), una palabra (16 bits) o una palabra doble (32 bits).

Cada posición de memoria tiene una dirección unívoca. El programa de usuario emplea dichas direcciones para acceder a la información contenida en dicha posición. El direccionamiento de las marcas consta de los siguientes elementos:

- Identificador de memoria (M).
- Tamaño del dato que se desea leer o escribir ('B' para byte, 'W' para palabra, 'DW' para palabra doble; en el caso de guardar un único bit se omite el tamaño).

- Dirección inicial del dato. Se escribe el número del byte (entre 0 y 8191) seguido de un punto (.) y el bit de la dirección deseada (entre 0 y 7).

Lo anterior se ilustra en la Figura 3.7. En dicho esquema está representada de manera gráfica la organización del área de marcas. En la cuadrícula se muestra el mapa de memoria. Los renglones representan bytes de memoria, que están numerados del 0 al 8191 de arriba a abajo (teniendo un total de 8192 bytes). Cada byte está dividido en 8 bits, representados en la imagen por los cuadros pequeños. Los bits se numeran del 0 al 7 de derecha a izquierda. Para el ejemplo mostrado en Figura 3.7, la marca mostrada corresponde a la dirección M3.4

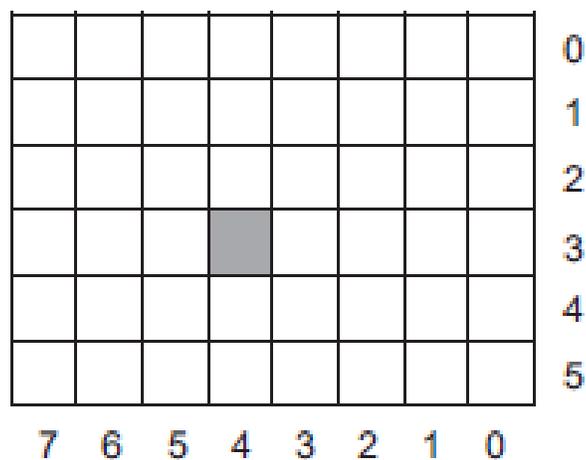


Figura 3.7 - Esquema del área de memoria de marcas

La forma en que se manejan las marcas es muy similar a como se usan las entradas y salidas de campo. En la tabla de variables del PLC (Figura 3.8) se hace el registro de la nueva variable con su respectiva etiqueta y tipo de dato. En el campo de *dirección* se ingresa la

	Nombre	Tipo de datos	Dirección	Rema...	Acces...	Escrib...	Visibl...	Comentario
1	Arranque	Bool	%I1.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Botón normalmente abierto
2	Motor	Bool	%Q0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Motor trifásico
3	Salida1	Bool	%M3.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	Regenera			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figura 3.8 - Tabla de variables. Existe una variable con una marca asignada

dirección del área de memoria de inicio de la marca con el formato especificado anteriormente.

De manera análoga, en el diagrama de contactos se asigna la variable con marca al contacto o salida deseado ya sea arrastrando el nombre de la variable desde el panel de variables, o escribiendo el nombre en las marcas <??.?>

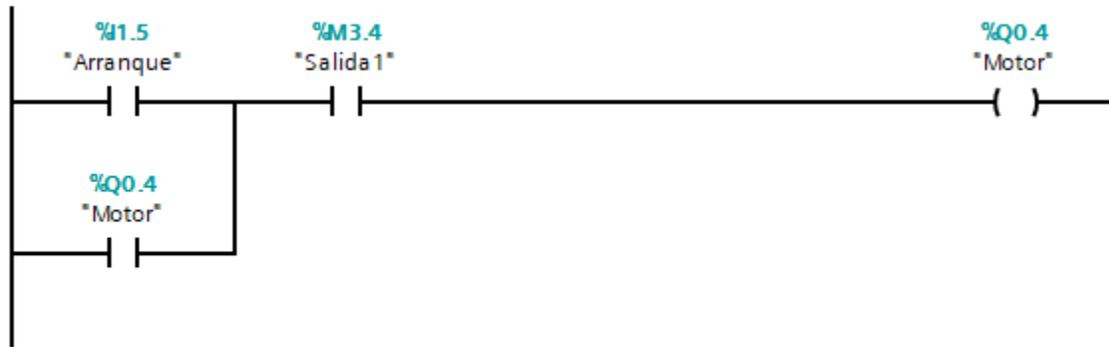


Figura 3.9 - Contacto con la variable asociada a la marca anterior

Es importante tomar en cuenta que, tanto para las entradas, salidas y marcas, la etiqueta que nombra a cada variable debe ser unívoca, por lo que no puede repetirse el mismo nombre de variables en una misma tabla de variables. Así mismo, la dirección de memoria debe ser unívoca, por lo que se debe cuidar no repetir direccionamiento ni a la memoria de proceso ni al área de marcas.

3.3 Tipos de temporizadores

Las instrucciones con temporizadores se utilizan para crear retardos programados. El número de temporizadores que pueden utilizarse en un programa de usuario está limitado sólo por la memoria disponible en el PLC. Cada temporizador utiliza un bloque de datos DB de 16 bytes para guardar la información del temporizador. TIA Portal crea automáticamente el DB al introducir la instrucción.

Es importante mencionar que tanto para los temporizadores como para los contadores (ver [sección 3.4](#)), no es riguroso colocar el bloque de instrucción al final del peldaño. Esto implica que la salida del temporizador (Q) puede conectarse directamente a una salida discreta para

activarla, o bien puede usarse en conjunto con otros contactos para formar bloques, por lo que no es totalmente necesario crear un contacto que se asocie con la salida del temporizador. A continuación, se describen los cuatro tipos de temporizadores con los que se puede trabajar con el PLC S7-1200.

- **TP (Pulse timer):** Comúnmente llamada **Impulso**. Esta instrucción activa la salida Q por un tiempo programado. La instrucción se ejecuta cuando la entrada IN cambia de 0 a 1 (flanco ascendente). Cuando la instrucción se ejecuta, se empieza a contar el tiempo programado PT y se activa la salida Q durante este lapso, independientemente de cómo evolucione la señal de entrada (ver Figura 3.11). La detección de un nuevo flanco de señal ascendente tampoco influye en el estado lógico de la salida Q mientras transcurra el tiempo PT.

El valor de tiempo actual se puede consultar en la salida ET. Este valor de tiempo empieza a contar a partir de T#0s y termina al alcanzarse el valor del tiempo PT. Una vez alcanzado este tiempo PT y si el estado lógico de la entrada IN es "0", se desactiva la salida ET.

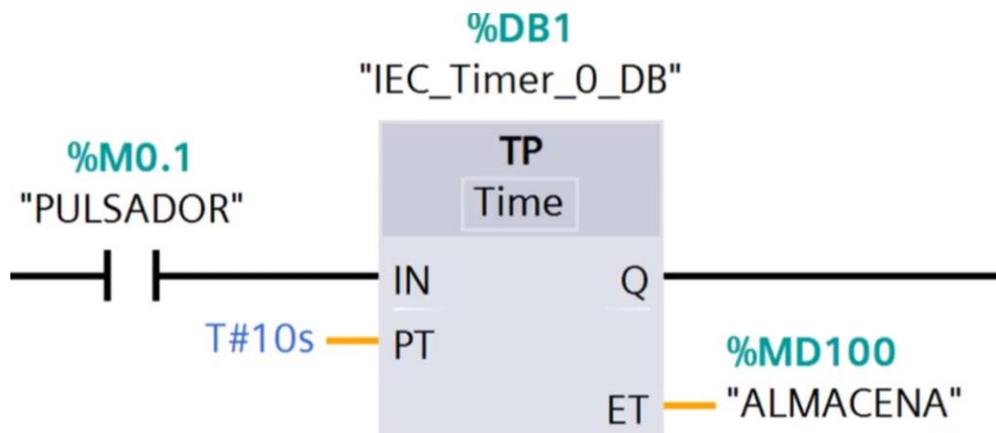


Figura 3.10 - Bloque del temporizador TP

Tabla 3.1 - Parámetros para el temporizador TP

Parámetro	Declaración	Tipo de dato	Descripción
IN	Input	BOOL	Entrada de arranque, inicia la instrucción
PT	Input	TIME	Duración del impulso. Debe ser positivo
Q	Output	BOOL	Salida del impulso
ET	Output	TIME	Valor de tiempo actual

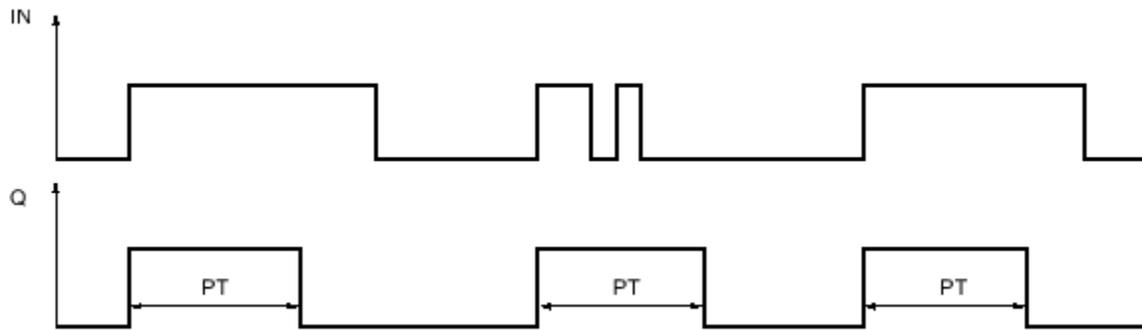


Figura 3.11 - Gráfica de tiempos de la entrada IN y la salida Q para la instrucción TP

Se debe tomar en cuenta que los valores PT (tiempo predeterminado) y ET (tiempo transcurrido) se almacenan en el bloque DB como milisegundos. Los datos TIME utilizan el identificador T# y pueden introducirse como unidad de tiempo simple ("T#200ms o 200) o como unidades de tiempo compuestas "T#2s_200ms" con una escala de 1 [ms].

- **TON (ON-delay timer):** Es un temporizador de retardo a la conexión. Éste retarda la activación de la salida Q por el tiempo programado PT. La instrucción se inicia cuando la entrada IN cambia de 0 a 1 (señal ascendente). Cuando se inicia la instrucción, se empieza a contar el tiempo programado PT. Una vez transcurrido el tiempo PT, la salida Q devuelve el estado lógico 1. La salida Q permanecerá activada mientras la entrada de arranque esté activa (Figura 3.13) Cuando el estado lógico de la entrada de arranque se desactiva, se desactiva también la salida Q. La función de temporización se reinicia al detectarse un nuevo flanco de señal ascendente en la entrada de arranque. Si se interrumpe la señal IN antes de que termine de transcurrir el tiempo PT, el tiempo ET vuelve a T#0 y se reinicia el temporizador.

Al igual que en la instrucción TP, el valor de tiempo actual se puede consultar en la salida ET. Este valor de tiempo empieza a contar a partir de T#0s y termina al alcanzarse el valor del tiempo PT. La salida ET se desactiva en cuanto la entrada IN se desactiva.

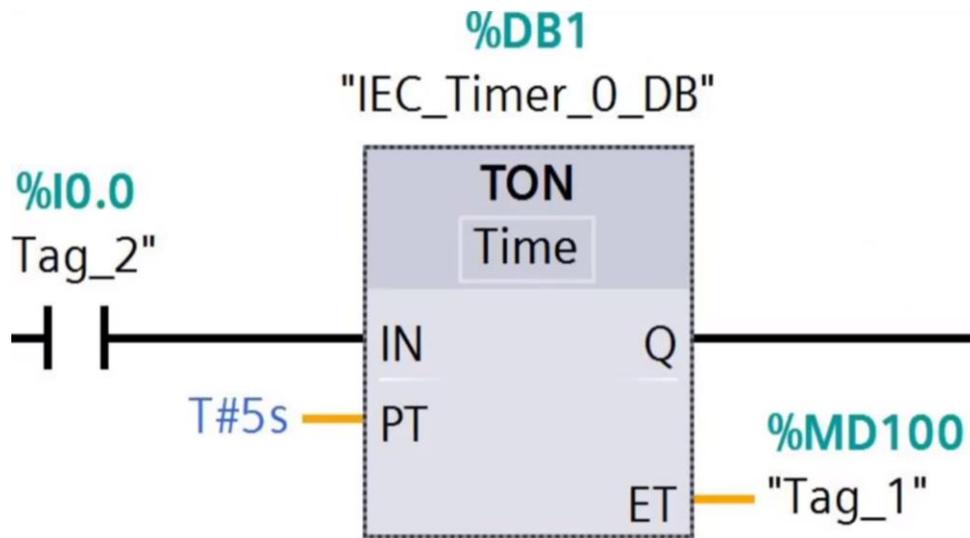


Figura 3.12 - Bloque del temporizador TON

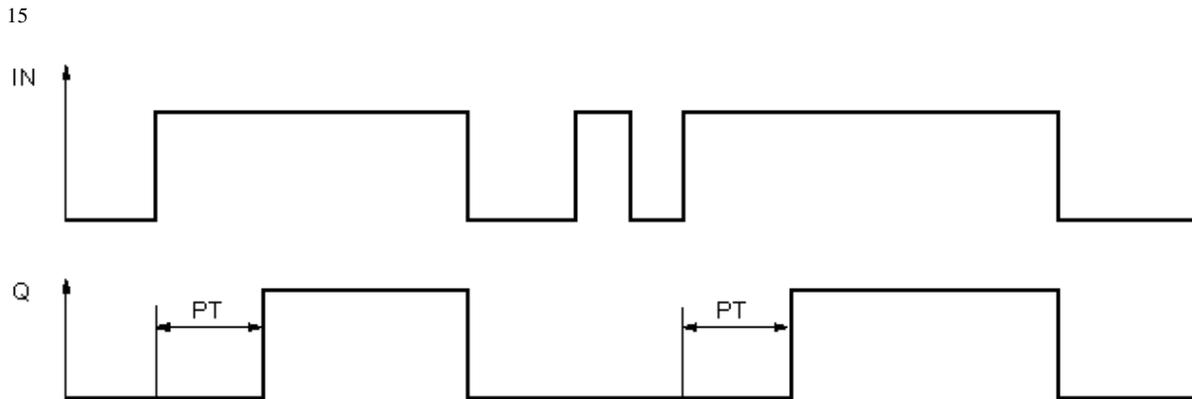


Figura 3.13 - Gráfica de tiempos de la entrada IN y la salida Q para la instrucción TON

Tabla 3.2 - Parámetros para el temporizador TON

Parámetro	Declaración	Tipo de dato	Descripción
IN	Input	BOOL	Entrada de arranque, inicia la instrucción
PT	Input	TIME	Tiempo de retardo a la conexión. Debe ser positivo
Q	Output	BOOL	Salida que se activa una vez transcurrido el tiempo PT
ET	Output	TIME	Valor de tiempo actual

¹⁵ SIMATIC STEP 7 Basic (15.1) [Software] (2018).

- **TOF (OFF-delay timer):** Es una instrucción de retardo a la desconexión. Ésta retarda la desactivación de la salida Q por el tiempo programado PT. Es el inverso a la instrucción TON, ya que la salida Q se activa cuando la entrada IN cambia de 0 a 1 (señal ascendente). Cuando el estado de la entrada IN cambia nuevamente a 0, el tiempo programado PT comienza a contar. La salida Q permanece activada mientras transcurre el tiempo PT. Una vez transcurrido el tiempo PT se desactiva la salida Q (Figura 3.15). Si el estado lógico de la entrada IN cambia a 1 antes de que transcurra el tiempo PT, se inicializa el temporizador con lo que el estado lógico de la salida Q permanece en 1. Las características de la salida ET son las mismas que para el temporizador TON.

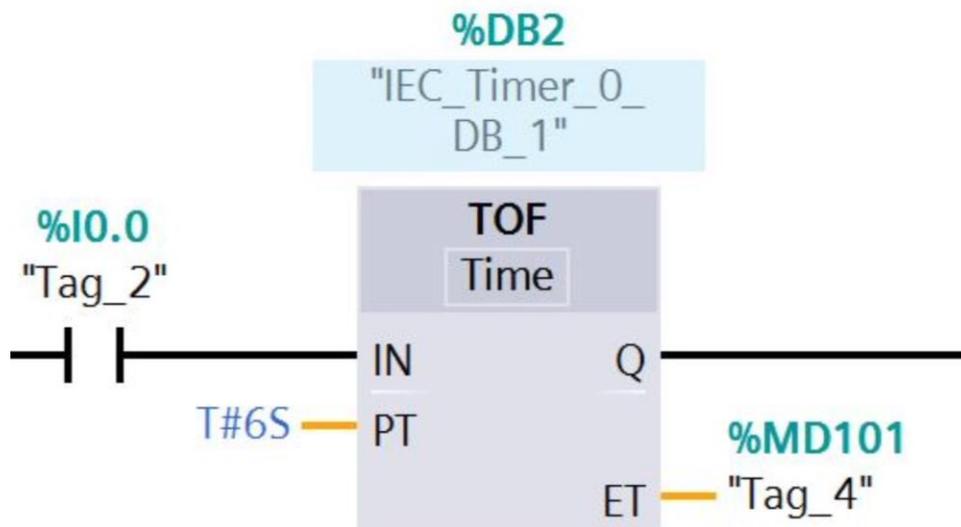


Figura 3.14 - Bloque del temporizador TOF

16

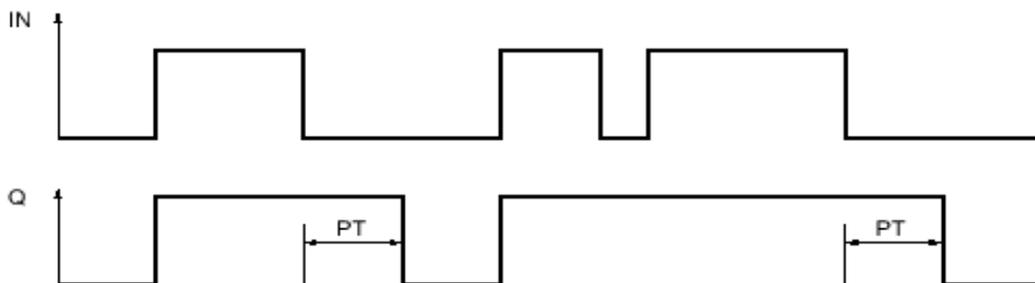


Figura 3.15 - Gráfica de tiempos de la entrada IN y la salida Q para la instrucción TOF

¹⁶ SIMATIC STEP 7 Basic (15.1) [Software] (2018).

Tabla 3.3 - Parámetros para el temporizador TOF

Parámetro	Declaración	Tipo de dato	Descripción
IN	Input	BOOL	Entrada de arranque, inicia la instrucción
PT	Input	TIME	Tiempo de retardo a la desconexión. Debe ser positivo
Q	Output	BOOL	Salida que se desactiva una vez transcurrido el tiempo PT
ET	Output	TIME	Valor de tiempo actual

- **TONR (ON-delay Retentive timer):** Es una instrucción de acumulación de tiempo. Ésta acumula valores de tiempo dentro de un periodo definido por el parámetro PT. Cuando el estado lógico de la entrada IN cambia de "0" a "1" (señal ascendente), se ejecuta la instrucción y se empieza a contar el tiempo PT. Mientras transcurre el tiempo PT se van acumulando los valores de tiempo que se leen cuando la entrada IN está activa (Figura 3.17). El tiempo acumulado se deposita en la salida ET y se puede consultar allí. Una vez se ha alcanzado el tiempo PT, la salida Q devuelve el estado lógico 1. El parámetro Q permanece a "1" aunque el estado lógico del parámetro IN cambie de "1" a "0" (flanco de señal descendente). La entrada R desactiva las salidas ET y Q independientemente del estado lógico de la entrada de arranque, por lo que tiene prioridad.

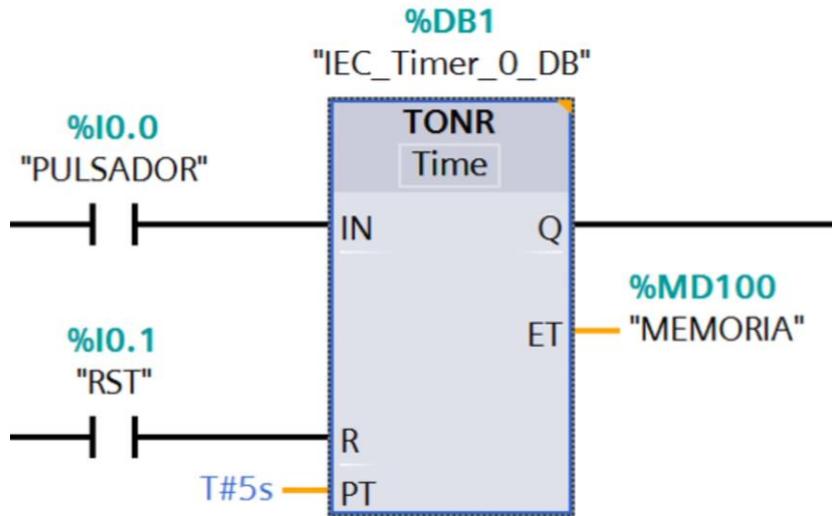


Figura 3.16 - Bloque del temporizador TONR

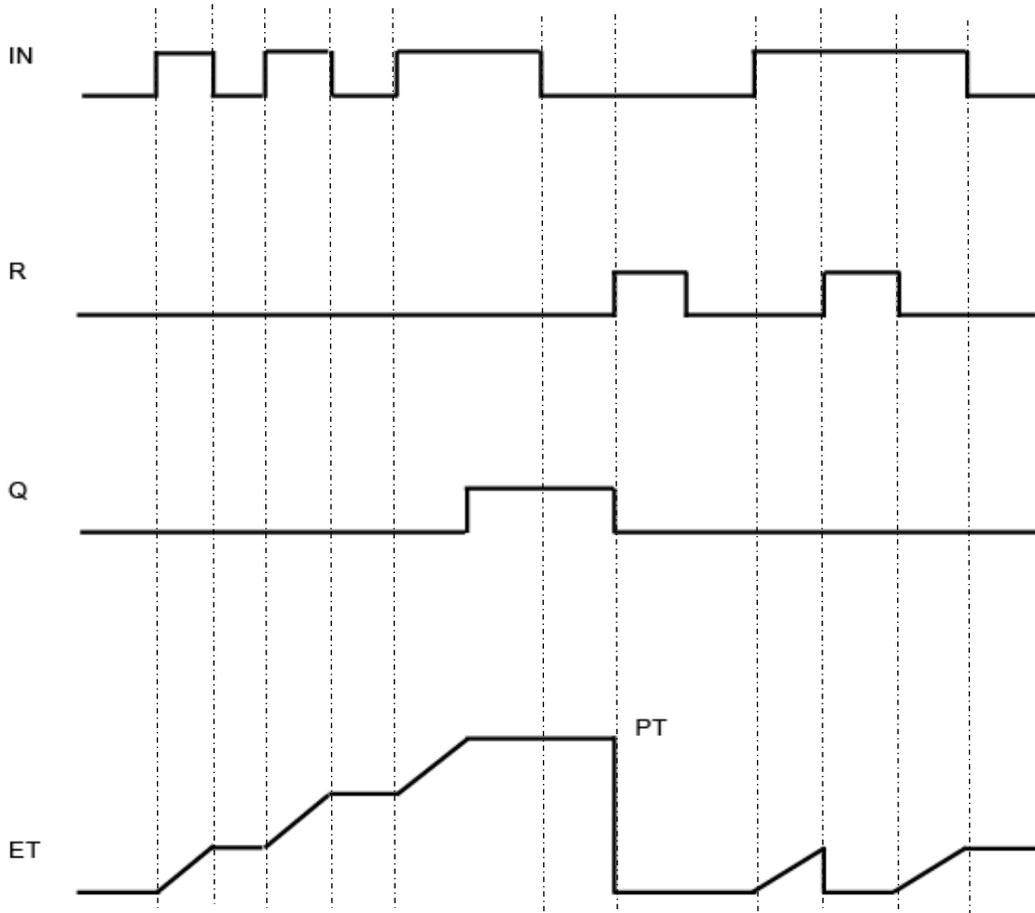


Figura 3.17 - Gráfica de tiempos de las señales IN, R, Q, ET y PT para la instrucción TONR

¹⁷ SIMATIC STEP 7 Basic (15.1) [Software] (2018).

Tabla 3.4 - Parámetros para el temporizador TONR

Parámetro	Declaración	Tipo de dato	Descripción
IN	Input	BOOL	Entrada de arranque, inicia la instrucción
PT	Input	TIME	Tiempo a acumular. Debe ser positivo
Q	Output	BOOL	Salida que se desactiva una vez transcurrido el tiempo PT
ET	Output	TIME	Tiempo acumulado respecto al tiempo PT
R	Input	BOOL	Regresa a 0 las variables Q y ET

3.4 Tipos de contadores

Los contadores son instrucciones que ayudan a registrar el número de veces que ha sucedido un evento. El número de contadores que puede usarse en un programa está limitado únicamente por la cantidad de memoria disponible en el PLC dado el programa del usuario. Es importante remarcar que estas instrucciones están limitadas por el tiempo de ejecución del OB en que están contenidas. Esto quiere decir que el periodo de ejecución del OB debe ser suficiente para detectar todas las transiciones o cambios en las entradas de los contadores. Si la frecuencia de conteo es mayor que la ejecución del OB, deben usarse instrucciones avanzadas de conteo rápido.

Al igual que en el caso de los temporizadores, los bloques de instrucciones de conteo no necesariamente se colocan al final de los peldaños, sino que es posible conectar la salida del contador (Q) a salidas discretas directamente sin la necesidad de emplear un contacto asociado a la salida del contador.

A continuación, se describen los tres tipos de contadores con los que cuenta el PLC S7-1200:

- **CTU (Count Up)**: Es una instrucción de conteo ascendente. Cuando el estado lógico de la entrada CU cambia de 0 a 1 (flanco ascendente), se ejecuta la instrucción y el valor actual de la salida CV se incrementa en uno. El valor de CV se incrementa cada vez que se detecta un flanco de señal ascendente, hasta alcanzar el valor límite superior indicado en la salida CV. Cuando se alcanza el valor límite superior, el

estado lógico de la entrada CU deja de tener efecto en la instrucción. El estado del contador se puede consultar en la salida Q (Figura 3.19). El estado lógico de la salida Q es determinado por el parámetro PV. Si el valor de conteo actual es mayor o igual que el valor del parámetro PV, la salida Q adquiere un estado lógico de 1. En todos los demás casos, el estado lógico de la salida Q es 0. El valor de la salida CV se pone a cero cuando el estado lógico de la entrada R cambia a 1. Mientras la entrada R tenga el estado lógico 1, el estado lógico de la entrada CU no tendrá efecto alguno en la instrucción, por lo que la señal de reinicio tiene prioridad.

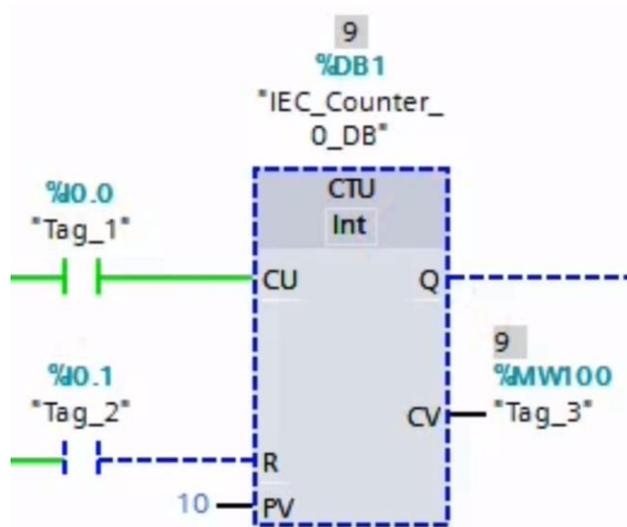


Figura 3.18 - Bloque de la instrucción CTU

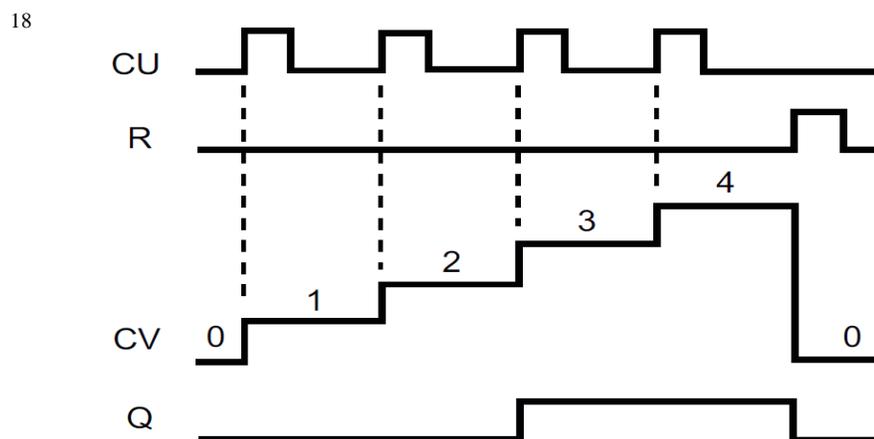


Figura 3.19 - Esquema de tiempo de las señales CU, R, CV y Q

¹⁸ SIMATIC STEP 7 Basic (15.1) [Software] (2018).

Tabla 3.5 - Parámetros para contador CTU

Parámetro	Declaración	Tipo de dato	Descripción
CU	Input	BOOL	Señal de conteo
R	Input	BOOL	Señal de reinicio
PV	Input	Enteros positivos	Límite de conteo
Q	Output	BOOL	Estado actual del contador
CV	Output	Enteros positivos	Valor actual del contador

- **CTD (Count Down):** Es una instrucción de conteo descendente. Es el contrario del contador CTU. Cuando el estado lógico de la entrada CD cambia de 0 a 1 (flanco ascendente), se ejecuta la instrucción y el valor actual de conteo de la salida CV se decrementa en uno. Cada vez que se detecta un flanco de señal ascendente, el valor del contador decrementa hasta alcanzar el valor límite inferior indicado. Cuando se alcanza el valor límite inferior, el estado lógico de la entrada CD deja de tener efecto en la instrucción. El estado del contador se puede consultar en la salida Q. Si el valor actual del contador es menor o igual a cero, la salida Q adopta el estado lógico 1. En todos los demás casos, el estado lógico de la salida Q es 0 (Figura 3.21). El valor de la salida CV se carga con el valor del parámetro PV cuando el estado lógico de la entrada LD cambia a 1. Mientras la entrada LD tenga el estado lógico 1, el estado lógico de la entrada CD no tendrá efecto alguno en la instrucción, esto quiere decir que la señal de carga tiene prioridad (de manera análoga a la señal de reinicio en la instrucción CTU).

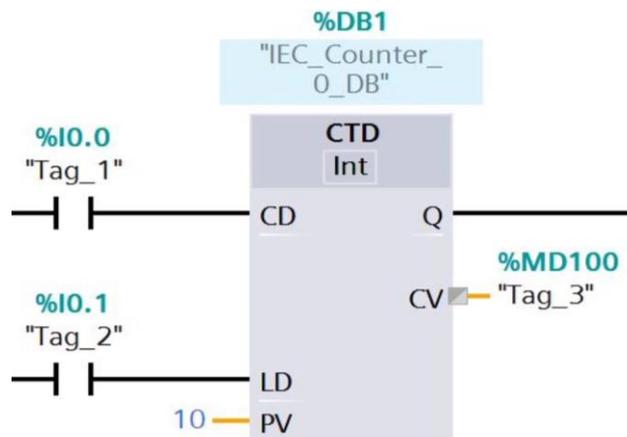


Figura 3.20 - Bloque de la instrucción CTD

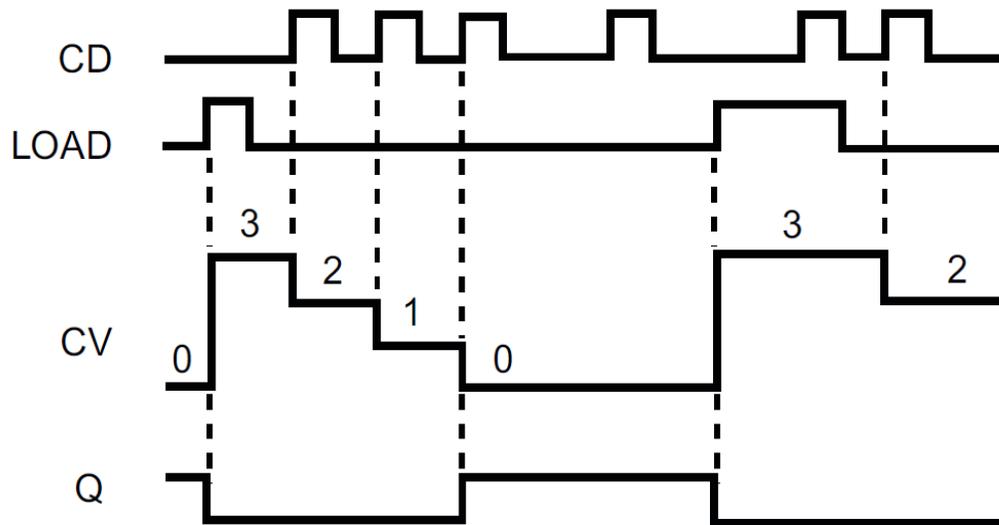


Figura 3.21 - Esquema de tiempos de las señales CD, LD, CV y Q

Tabla 3.6 - Parámetros para contador CTD

Parámetro	Declaración	Tipo de dato	Descripción
CD	Input	BOOL	Señal de conteo
LD	Input	BOOL	Señal de carga
PV	Input	Enteros	Valor inicial a cargar del contador
Q	Output	BOOL	Estado actual del contador
CV	Output	Enteros	Valor actual del contador

- **CTUD (Count Up and Down):** Es una instrucción de conteo ascendente y descendente. Cuando el estado lógico de la entrada CU cambia de 0 a 1 (flanco ascendente), el valor actual del contador se incrementa en uno y se muestra en la salida CV. Cuando el estado lógico de la entrada CD cambia de 0 a 1 (flanco ascendente), el valor de conteo de la salida CV se decrementa en uno. Si en un ciclo del programa se detecta un flanco de señal ascendente en las entradas CU y CD, el valor actual del contador no se modifica en la salida CV. El valor del contador se puede seguir incrementando hasta alcanzar el valor límite superior del tipo de datos indicado en la salida CV (revisar las tablas de rangos de valores para los tipos de

datos en la sección 2.5 para mayor información). Una vez alcanzado el valor límite superior, el valor de conteo no se incrementa más, aunque se detecte un flanco de señal ascendente. Cuando se alcanza el valor límite inferior indicado, ya no se decrementa el valor de conteo. Si el estado lógico de la entrada LD cambia a 1, el valor actual del contador CV adopta el valor del parámetro PV. Mientras la entrada LD tenga el estado lógico 1, el estado lógico de las entradas CU y CD no tendrá efecto alguno en la instrucción; la señal de carga tiene prioridad parcial. El valor de conteo se pone a cero si el estado lógico de la entrada R cambia a "1". Mientras la entrada R tenga el estado lógico "1", un cambio del estado lógico de las entradas CU, CD y LD no tendrá efecto alguno en la instrucción, por lo que la señal R tiene prioridad total sobre la instrucción (Figura 3.23). El estado del contador ascendente se puede consultar en la salida QU. Si el valor de conteo actual es mayor o igual al valor del parámetro PV, la salida QU devuelve un 1 lógico. En todos los demás casos, el estado lógico de la salida QU es 0. El estado del contador descendente se puede consultar en la salida QD. Si el valor de conteo actual es menor o igual a cero, la salida QD adopta el estado lógico "1". En todos los demás casos, el estado lógico de la salida QD es 0.

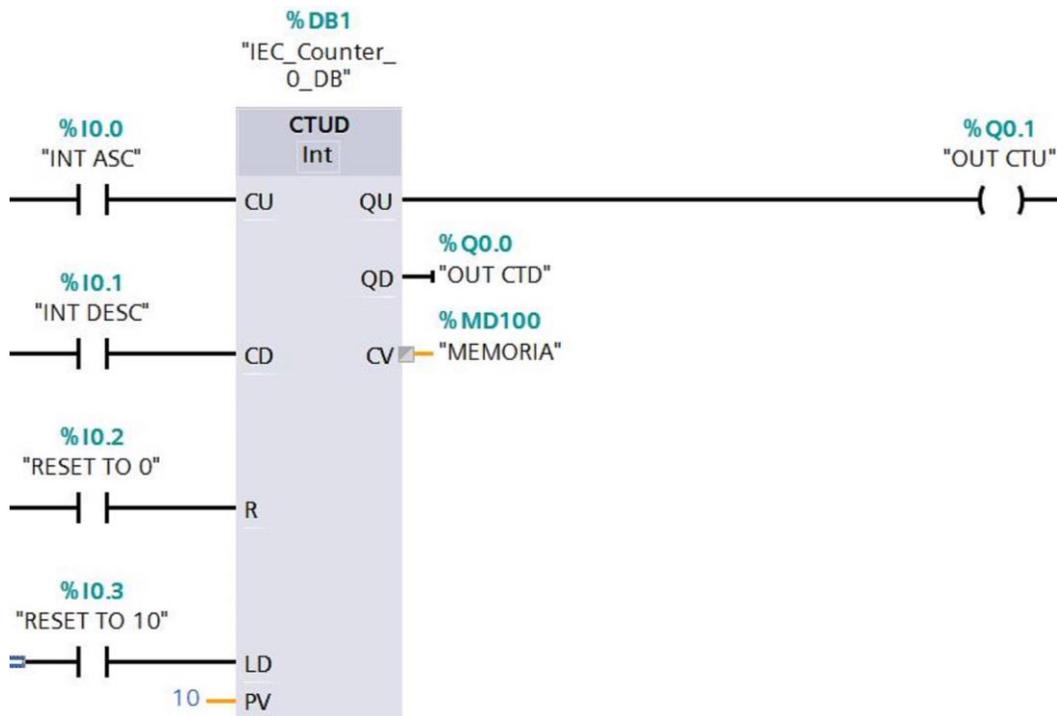


Figura 3.22 - Bloque de la instrucción CTUD

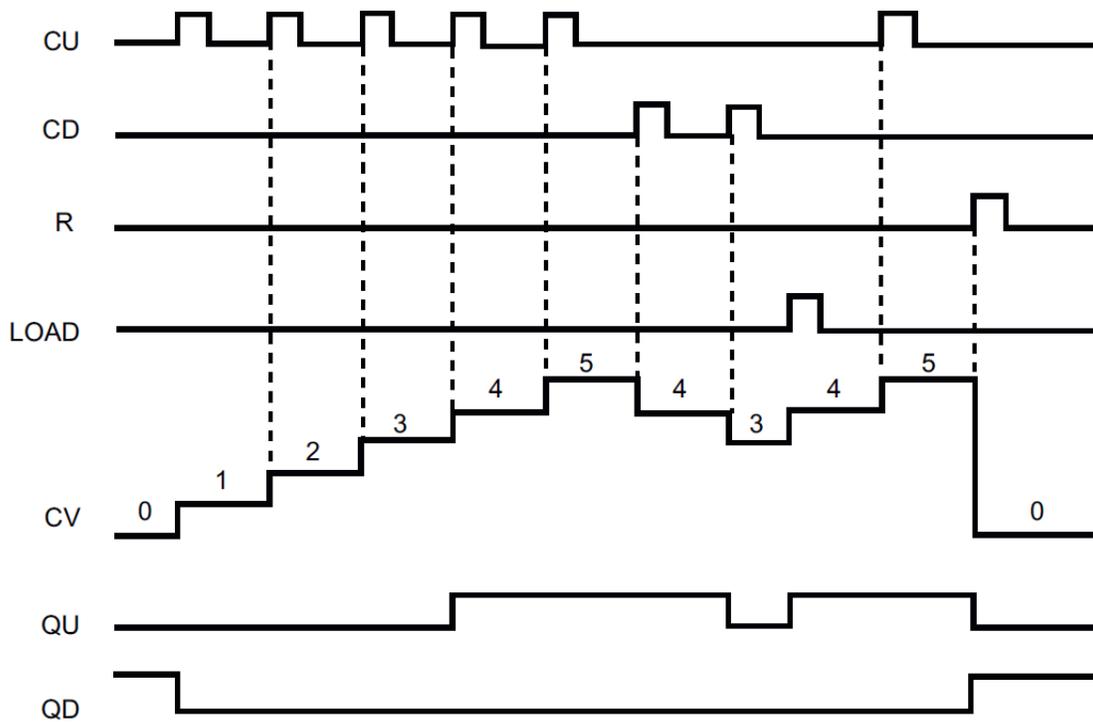


Figura 3.23 - Esquema de tiempos para las señales CU, CD, R, LD; CV, QU, y QD

Tabla 3.7 - Parámetros para contador CTUD

Parámetro	Declaración	Tipo de dato	Descripción
CU	Input	BOOL	Señal de conteo ascendente
CD	Input	BOOL	Señal de carga descendente
R	Input	BOOL	Señal de reinicio
LD	Input	BOOL	Señal de carga
PV	Input	Enteros	Valor con el que se activa QU. Valor con el que se carga CV cuando LD = 1
QU	Output	BOOL	Estado del contador ascendente
QD	Output	BOOL	Estado del contador descendente
CV	Output	Enteros	Valor actual del contador

3.5 Registros de corrimiento

Las funciones de desplazamiento y rotación sirven para manipular el contenido de la memoria y desplazarlo a direcciones adyacentes. A continuación, se describen las cuatro funciones de desplazamiento y rotación disponibles en el PLC S7-1200:

- **SHR (Shift Right):** Es una instrucción de registro de corrimiento hacia la derecha. El bloque permite desplazar el contenido del operando de la entrada IN de bit en bit hacia la derecha y consultar el resultado en la salida OUT. El parámetro N determina el número de bits que debe desplazarse el valor indicado. Si el valor del parámetro N es 0, el valor de la entrada IN se copia en el operando de la salida OUT. Si el valor del parámetro N es mayor que el número de bits disponibles, el valor del operando de la entrada IN se desplaza hacia la derecha las posiciones de bit disponibles. En los valores sin signo, se rellenan con ceros los bits que quedan libres en el área izquierda del operando al realizar el desplazamiento. Si el valor indicado lleva signo, las posiciones libres se ocupan con el estado lógico del bit de signo.

21

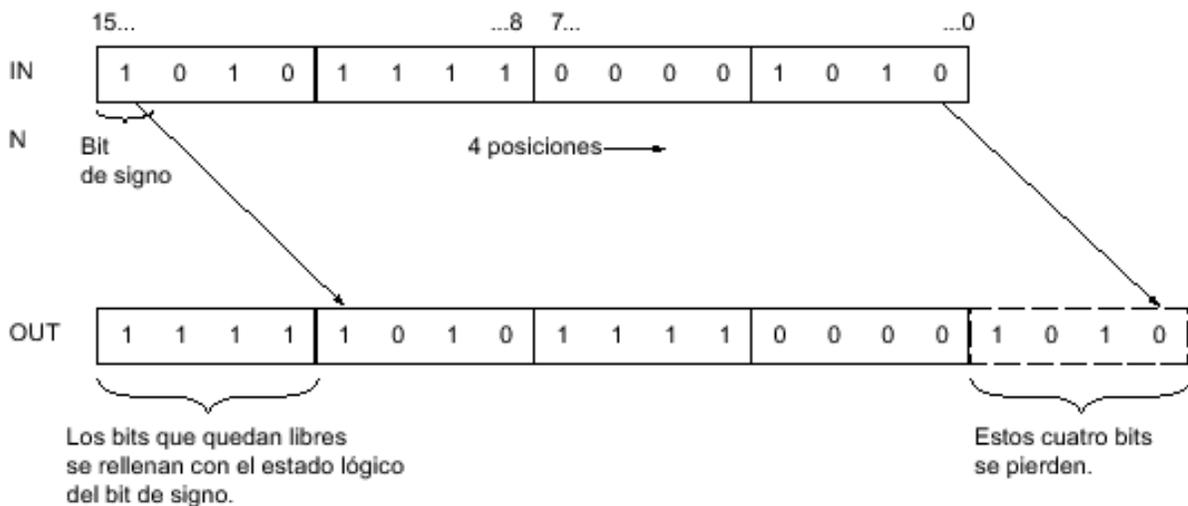


Figura 3.24 - Se muestra cómo el contenido de un operando del tipo Word se desplaza cuatro posiciones de bit hacia la derecha

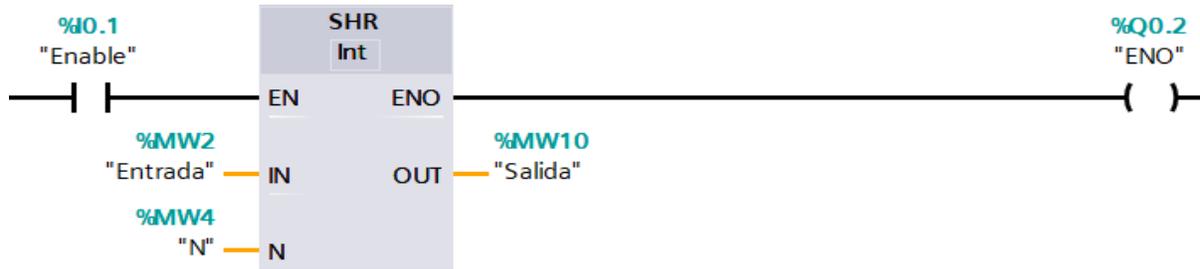


Figura 3.25 - Bloque de la instrucción SHR

Tabla 3.8 - Parámetros para el registro de corrimiento SHR

Parámetro	Declaración	Tipo de dato	Descripción
EN	Input	BOOL	Señal de habilitación
ENO	Output	BOOL	Salida de habilitación
IN	Input	Secuencias de bits	Secuencia a desplazar
N	Input	USInt, UInt, UDInt	Número de posiciones a desplazar
OUT	Output	Secuencias de bits	Resultado del desplazamiento

- **SHL (Shift Left):** Es una instrucción de registro de corrimiento hacia la izquierda, ésta permite desplazar el contenido del operando de la entrada IN de bit en bit hacia la izquierda y consultar el resultado en la salida OUT, por lo que es la instrucción complementaria al registro de corrimiento a la derecha SHR. Igualmente, el parámetro N determina el número de bits que debe desplazarse el valor indicado. Si el valor del parámetro N es "0", el valor de la entrada IN se copia en el operando de la salida OUT. Si el valor del parámetro N es mayor que el número de bits disponibles, el valor del operando de la entrada IN se desplaza hacia la izquierda las posiciones de bit disponibles. Los bits que quedan libres en el área derecha del operando al realizar el desplazamiento se rellenan con ceros.

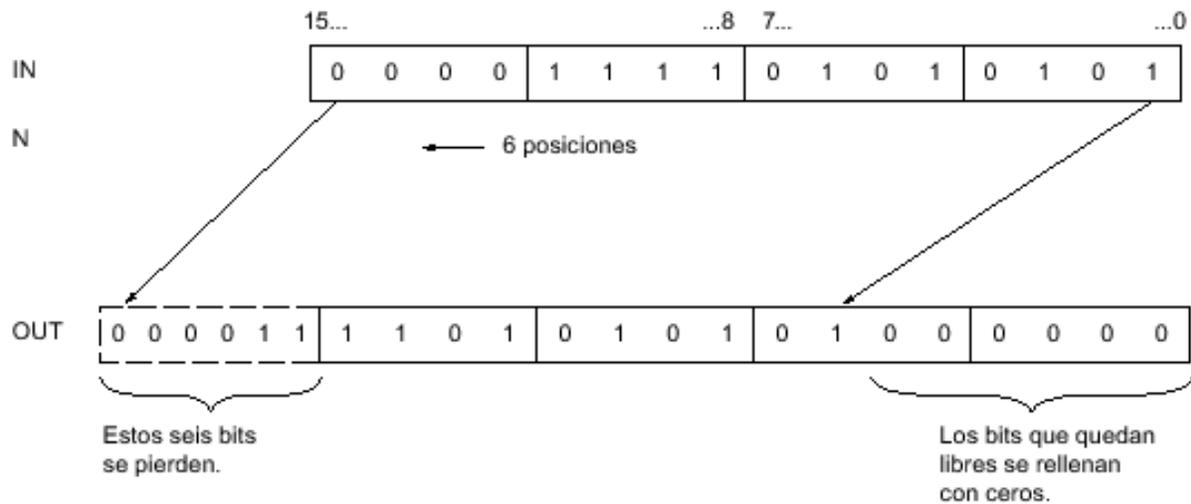


Figura 3.27 - Se muestra cómo el contenido de un operando del tipo Word se desplaza seis posiciones de bit hacia la izquierda

22

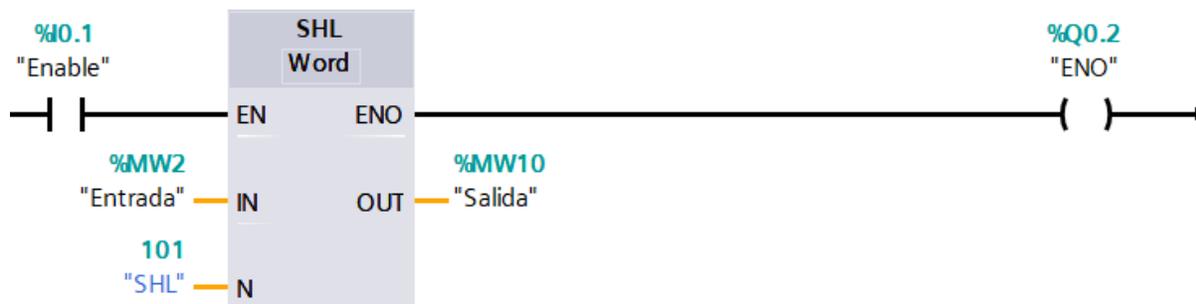


Figura 3.26 - Bloque de la instrucción SHL

Tabla 3.9 - Parámetros para el registro de corrimiento SHL

Parámetro	Declaración	Tipo de dato	Descripción
EN	Input	BOOL	Señal de habilitación
ENO	Output	BOOL	Salida de habilitación
IN	Input	Secuencias de bits	Secuencia a desplazar
N	Input	USInt, UInt, UDIInt	Número de posiciones a desplazar
OUT	Output	Secuencias de bits	Resultado del desplazamiento

- **ROR (Rotate Right):** Es una instrucción de rotación hacia la derecha, permite rotar el contenido del operando de la entrada IN de bit en bit hacia la derecha y consultar el resultado en la salida OUT. El parámetro N determina el número de bits que debe rotarse el valor indicado. Los bits que quedan libres al realizar la rotación se rellenan con los bits desplazados hacia afuera. Si el valor del parámetro N es 0, el valor de la entrada IN se copia en el operando de la salida OUT. Si el valor del parámetro N es mayor que el número de bits disponibles, el valor del operando en la entrada IN rota de todos modos el número indicado de posiciones de bit.

23

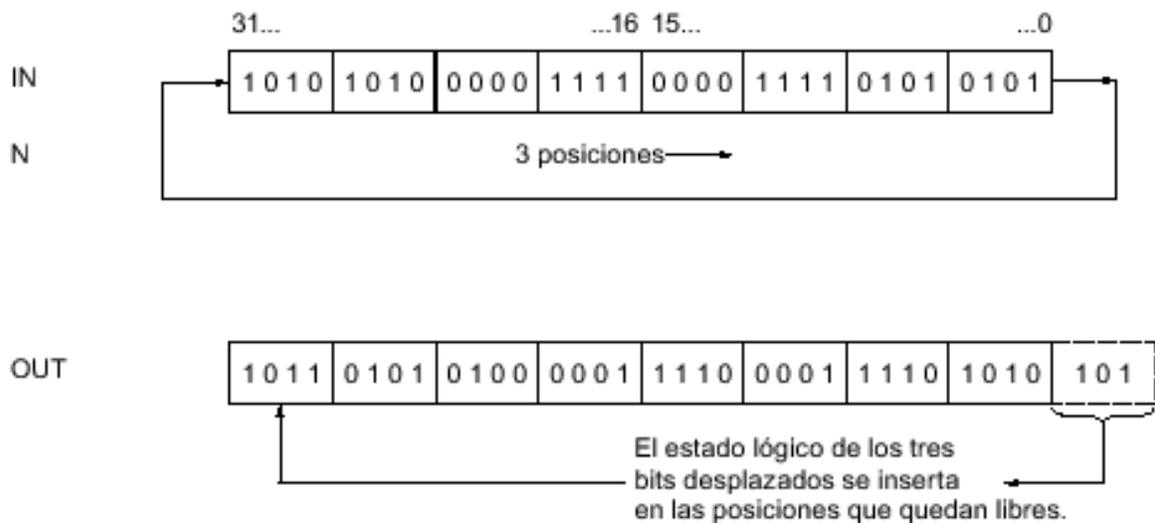


Figura 3.28 - Se muestra cómo el contenido de un operando del tipo DWord rota tres posiciones hacia la derecha



Figura 3.29 - Bloque de la instrucción ROR

Tabla 3.10 - Parámetros para la rotación ROR

Parámetro	Declaración	Tipo de dato	Descripción
EN	Input	BOOL	Señal de habilitación
ENO	Output	BOOL	Salida de habilitación
IN	Input	Secuencias de bits	Secuencia a rotar
N	Input	USInt, UInt, UDInt	Número de posiciones a rotar
OUT	Output	Secuencias de bits	Resultado de la rotación

- **²⁴ROR (Rotate Right):** Es una instrucción de rotación a la izquierda, la cual permite rotar el contenido del operando de la entrada IN de bit en bit hacia la izquierda y consultar el resultado en la salida OUT; es entonces la instrucción complementaria a la instrucción ROR. De la misma manera a su instrucción análoga, el parámetro N determina el número de bits que debe rotarse el valor indicado. Los bits que quedan libres al realizar la rotación se rellenan con los bits desplazados hacia fuera. Si el valor del parámetro N es "0", el valor de la entrada IN se copia en el operando de la salida OUT. Si el valor del parámetro N es mayor que el número de bits disponibles, el valor del operando en la entrada IN rota de todos modos el número indicado de posiciones de bit.

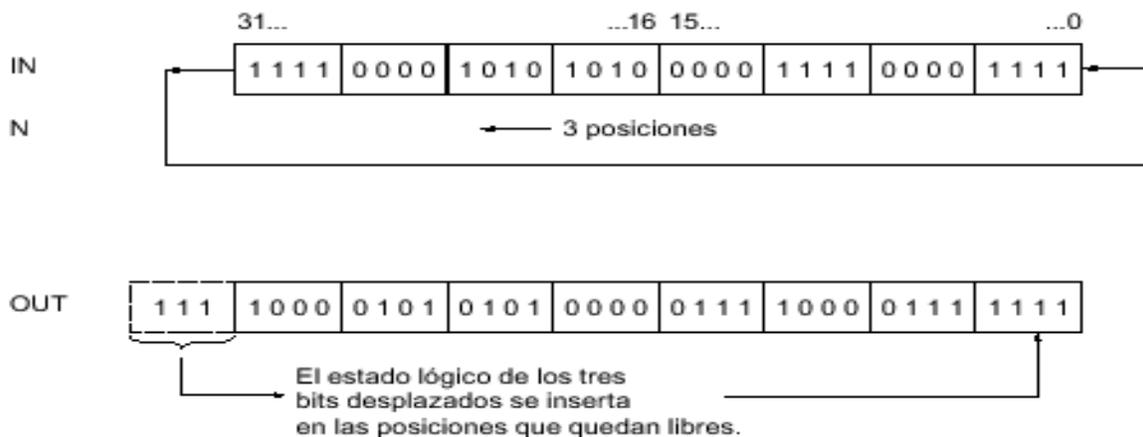


Figura 3.30 - Se muestra cómo el contenido de un operando del tipo DWord rota tres posiciones hacia la izquierda

²⁴ SIMATIC STEP 7 Basic (15.1) [Software] (2018).



Figura 3.31 - Bloque de la instrucción ROL

Tabla 3.11 - Parámetros para la rotación ROL

Parámetro	Declaración	Tipo de dato	Descripción
EN	Input	BOOL	Señal de habilitación
ENO	Output	BOOL	Salida de habilitación
IN	Input	Secuencias de bits	Secuencia a rotar
N	Input	USInt, UInt, UDInt	Número de posiciones a rotar
OUT	Output	Secuencias de bits	Resultado de la rotación

3.6 Constantes numéricas

El PLC S7-1200 ofrece la posibilidad de declarar constantes globales que pueden ser usadas en cualquier parte del programa de usuario, tanto en los bloques de organización OB, como en los bloques de función FB. Estas constantes, denominadas *constantes de usuario*, pueden ser de cualquier tipo de dato, de los cuales ya se habló en la sección 2.5 previamente. Las constantes de usuario no están asociadas a un área de memoria en particular y no pueden ser modificadas en la ejecución del programa; sin embargo, pueden ser usadas como parámetros de entrada a bloques funcionales.

Para hacer la definición de las constantes es necesario ir a la tabla de variables del PLC. En dicha ventana, en la parte superior derecha, hay una pestaña nombrada *Constantes de usuario*; es ahí donde se hace la declaración de las constantes globales. La ventana de constantes es idéntica a la ventana de asignación de variables. Cada renglón representa una constante, a la cual se le asigna un nombre, un tipo de datos y el valor deseado; es posible también añadir un comentario con el que se dé una breve explicación de dicha constante.

	Nombre	Tipo de datos	Valor	Comentario
1	Entrada 1	Int	5	
2	Entrada 2	DWord	520	
3	Entrada 3	String	'Hola mundo'	
4	Salida 1	Bool	False	
5	Salida 2	Byte	2#10011101	
6	Agregar			

Figura 3.32 - Tabla de constantes de usuario

3.7 Funciones de comparación

Las instrucciones de comparación sirven para comprobar condiciones entre dos variables. El PLC S7-1200 ofrece la posibilidad de hacer tres tipos de comparaciones: matemáticas, de rango y de validez. A continuación, se describe cada una de ellas.

- **Comparaciones aritméticas:** Comparan dos elementos del mismo tipo de dato. Si la comparación resulta ser verdadera, el contacto de la instrucción de comparación se cierra. Se trata de comparaciones binarias, por lo que sólo pueden recibir dos argumentos de entrada. Los tipos de datos que se pueden comparar son los siguientes: SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, String, Char, Time, y constantes. Es necesario especificar en el contacto el tipo de datos que se van a comparar. Se tienen en total seis posibles comparaciones.

Tabla 3.12 - Descripción de comparaciones aritméticas

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



Figura 3.33 - Contacto de comparación (igualdad) en TIA Portal. Se especifica que se van a comparar datos del tipo Byte. El parámetro superior es el primer argumento; y el inferior, es el segundo argumento

- **Comparaciones de rango:** Estas funciones comprueban si un valor de entrada está dentro (instrucción `IN_Range`, Valor dentro de rango) o fuera (instrucción `OUT_Range`, Valor fuera de rango) de un intervalo de valores específico. Los límites del rango se determinan mediante las entradas `MIN` y `MAX`. Las instrucciones comparan el valor de la entrada `VAL` con los valores de las entradas `MIN` y `MAX` y transfieren el resultado de la comparación a la salida del bloque.

Para la función **IN_Range**, si el valor de la entrada `VAL` cumple la comparación $MIN \leq VAL \leq MAX$, la salida del bloque devuelve el estado lógico "1".

Para la función **OUT_Range**, si el valor de la entrada `VAL` cumple la comparación $VAL \leq MIN$ o $MAX \leq VAL$, la salida del bloque devuelve el estado lógico "1".

Si no se cumple la comparación, la salida del cuadro devuelve el estado lógico "0".

En cualquier caso, si el estado lógico de la entrada del bloque es "0", no se ejecuta la instrucción. Para ejecutar la función de comparación es necesario que los valores que desean compararse sean del mismo tipo de datos. Así mismo, se debe especificar en el bloque el tipo de dato que se va a comparar.

²⁵ SIMATIC STEP 7 Basic (15.1) [Software] (2018).

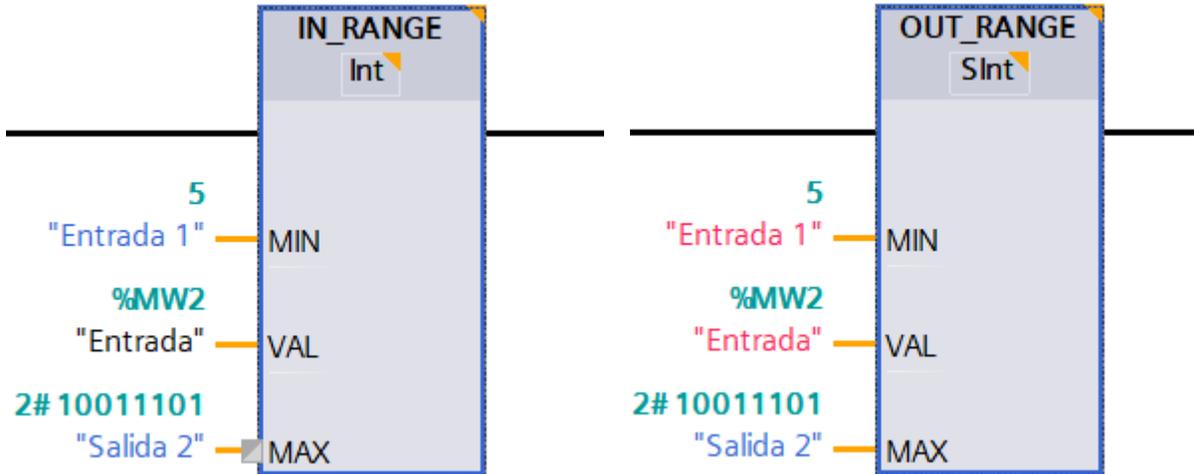


Figura 3.34 - - Bloque de comparación dentro de rango de valores enteros (izquierda) y fuera de rango de valores enteros signados (derecha)

Tabla 3.13 - Parámetros para las comparaciones de rango

Parámetro	Declaración	Tipo de dato	Descripción
Entrada al bloque	Input	BOOL	Resultado lógico anterior
MIN	Input	Enteros, reales en punto flotante	Límite inferior del rango de comparación
VAR	Input	Enteros, reales en punto flotante	Valor a comparar
MAX	Input	Enteros, reales en punto flotante	Límite superior del rango de comparación
Salida del bloque	Output	BOOL	Resultado de la comparación

- **Comparaciones de validez:** Estos contactos comprueban si una referencia de datos de entrada es un número real válido según la especificación IEEE 754, es decir, verifica si es un número en punto flotante válido. La instrucción *OK* (*Comprobar validez*) devuelve el estado lógico 1 cuando el valor del operando es un número en punto flotante válido cuando se realiza la consulta.; en todos los demás casos, la salida

de la instrucción *Comprobar validez* devuelve el estado lógico 0. Por el contrario, la instrucción *NOT_OK (Comprobar invalidez)* devuelve el estado lógico 1 cuando el valor del operando es un número en punto flotante inválido cuando se realiza la consulta.

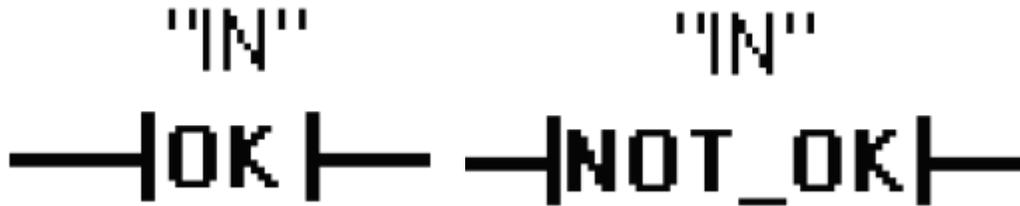


Figura 3.35 - Contacto de comprobación de validez (izquierda) y de invalidez (derecha).
El único parámetro de entrada es el valor a comparar

3.8 Funciones aritméticas

- **CALCULATE:** Esta instrucción permite definir y ejecutar una expresión para calcular operaciones matemáticas o combinaciones lógicas complejas en función del tipo de datos seleccionado. El tipo de datos de la instrucción se selecciona en la lista desplegable "???" del cuadro de la instrucción. Dependiendo del tipo de datos seleccionado se puede combinar la funcionalidad de determinadas instrucciones para ejecutar un cálculo complejo. La expresión puede contener los nombres de los parámetros de entrada y la sintaxis de las instrucciones. No está permitido indicar nombres o direcciones de operandos. El cuadro de la instrucción contiene en estado básico al menos 2 entradas (IN1 e IN2). El número de entradas es ampliable con la instrucción *Insertar entrada* que se encuentra en la carpeta *General*. Las entradas insertadas se numeran en orden ascendente en el cuadro. Los valores de las entradas se utilizan para ejecutar la expresión indicada. No obstante, no es necesario utilizar en la expresión todas las entradas definidas. El resultado de la instrucción se transfiere a la salida OUT.

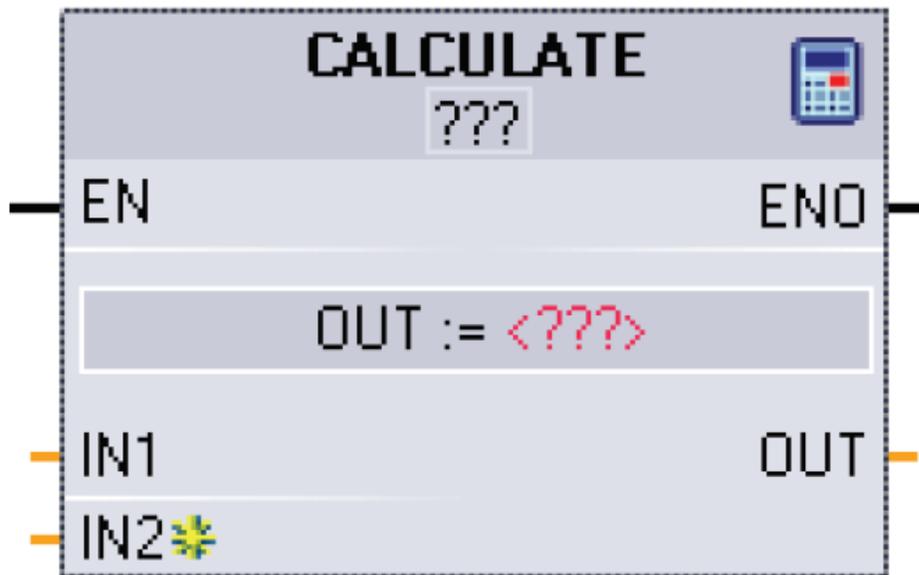


Figura 3.36 - Bloque de instrucción CALCULATE

Al hacer clic en el icono de la calculadora se abre el cuadro de diálogo para definir la función matemática. La ecuación se introduce como entradas (IN1, IN2, etc.) y operaciones. Cuando se hace clic en *Aceptar* para guardar la función, el cuadro de diálogo crea automáticamente las entradas de la instrucción CALCULATE. En la parte inferior del editor, se muestra un ejemplo y una lista de las operaciones matemáticas que se pueden incluir.

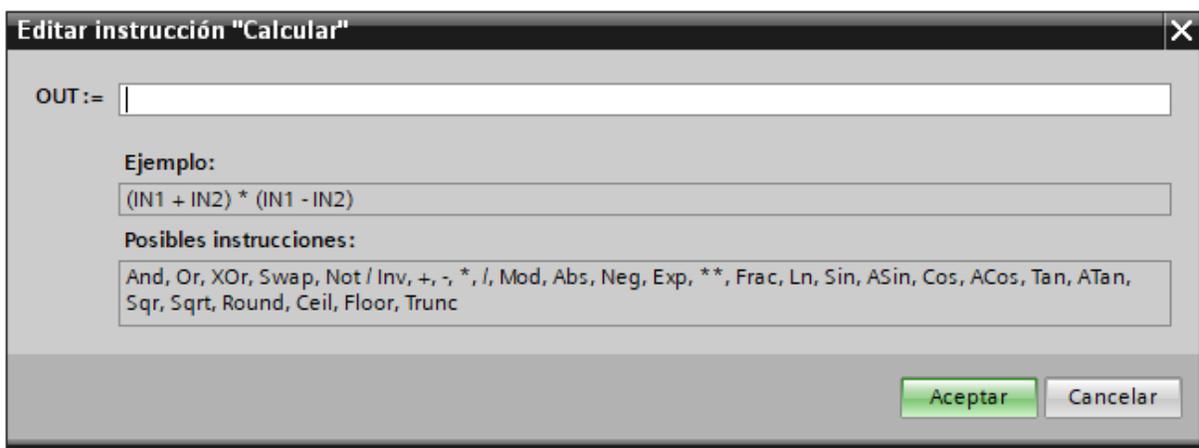


Figura 3.37 - Cuadro de diálogo de edición de la función CALCULATE

Cuando se ejecuta CALCULATE y todas las operaciones individuales del cálculo se realizan de forma correcta, entonces ENO devuelve 1. Si no es así, ENO devuelve 0.

Tabla 3.14 - Parámetros de la instrucción CALCULATE

Parámetro	Declaración	Tipo de dato	Descripción
EN	Input	BOOL	Entrada de habilitación
ENO	Output	BOOL	Salida de habilitación
INn	Input	Secuencias de bits, enteros, números en punto flotante	Entradas disponibles
OUT	Output	Secuencias de bits, enteros, números en punto flotante	Salida a la que se mueve el resultado final

- **Aritmética básica (ADD, SUB, MUL, DIV, MOD):** Son las operaciones básicas de adición, sustracción, multiplicación, división y módulo. En el caso de las instrucciones ADD y MUL, es posible añadir más de dos entradas.

Tabla 3.15 - Descripción de las operaciones aritméticas básicas

Tipo de operación	Resultado devuelto
ADD	$OUT = IN1 + IN2$
SUB	$OUT = IN1 - IN2$
MUL	$OUT = IN1 * IN2$
DIV	$OUT = IN1 / IN2$
MOD	$OUT = IN1 \text{ mod } IN2$

El tipo de datos de la instrucción se selecciona en la lista desplegable "???". Si el bloque está habilitado (EN = 1), la instrucción matemática realiza la operación indicada en los valores de entrada IN1 e IN2 y almacena el resultado en la dirección de memoria que indica el parámetro de salida OUT. Una vez finalizada correctamente la operación, la instrucción pone ENO a 1. Si ocurre algún error en la operación, ya sea por la naturaleza de la operación o por incompatibilidad en los tipos de datos, la salida ENO devuelve 0.

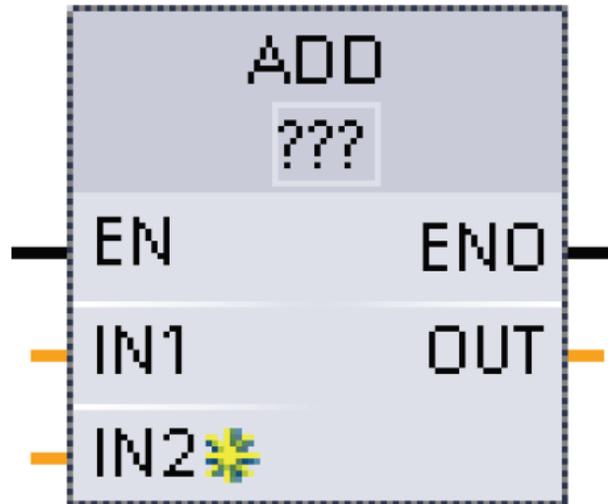


Figura 3.38 - Bloque de instrucción ADD

Tabla 3.16 - Parámetros de las instrucciones aritméticas

Parámetro	Declaración	Tipo de dato	Descripción
EN	Input	BOOL	Entrada de habilitación
ENO	Output	BOOL	Salida de habilitación
INn	Input	Enteros, números en punto flotante	Entradas disponibles
OUT	Output	Enteros, números en punto flotante	Resultado de la operación

- **NEG (Negative):** Es una instrucción de generación de complemento a dos. Permite cambiar el signo del valor de la entrada IN y consultar el resultado en la salida OUT. Esto es, si la entrada IN tiene un valor positivo, se deposita el equivalente negativo de este valor en la salida OUT y viceversa. Si ocurre un error en la instrucción, la salida ENO devuelve un 0 lógico; en caso contrario, toma un valor de 1 lógico.

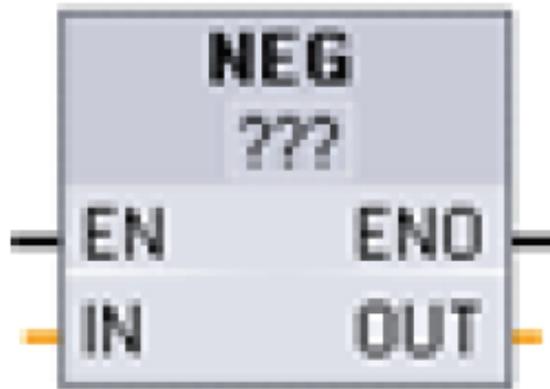


Figura 3.39 - Bloque de la instrucción NEG

Tabla 3.17 - Parámetros de la instrucción NEG

Parámetro	Declaración	Tipo de dato	Descripción
EN	Input	BOOL	Entrada de habilitación
ENO	Output	BOOL	Salida de habilitación
IN	Input	SINT, INT, DINT, números en punto flotante	Valor de entrada
OUT	Output	SINT, INT, DINT, números en punto flotante	Complemento a dos del valor de entrada

- **INC, DEC (Increment, Decrement):** La instrucción INC permite cambiar el valor del operando del parámetro IN/OUT al siguiente valor superior y consultar el resultado; mientras que la instrucción DEC cambia el operando al siguiente valor inferior. La ejecución de la instrucción se inicia sólo si el estado lógico de la entrada de habilitación EN es 1 lógico. Si no ocurre ningún rebase por exceso durante la ejecución, la salida de habilitación ENO también devuelve el estado lógico 1.

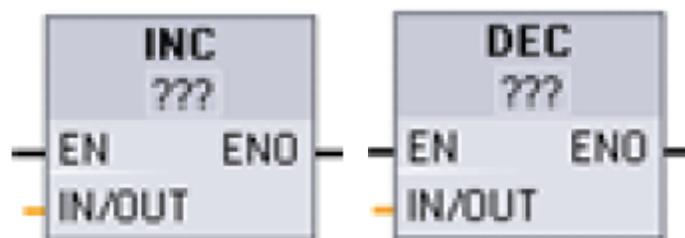


Figura 3.40 - Bloque de la instrucción INC (izquierda) y de la instrucción DEC (derecha)

Tabla 3.18 - Parámetros de las instrucciones INC y DEC

Parámetro	Declaración	Tipo de dato	Descripción
EN	Input	BOOL	Entrada de habilitación
ENO	Output	BOOL	Salida de habilitación
IN/OUT	InOut	Enteros	Valor a incrementar/decrementar

- **ABS (Absolute):** La instrucción permite calcular el valor absoluto del valor indicado en la entrada IN. El resultado de la instrucción se deposita en la salida OUT y se puede consultar allí.



Figura 3.41 - Bloque de la instrucción ABS

Tabla 3.19 - Parámetros de la instrucción ABS

Parámetro	Declaración	Tipo de dato	Descripción
EN	Input	BOOL	Entrada de habilitación
ENO	Output	BOOL	Salida de habilitación
IN	Input	SINT, INT, DINT, números en coma flotante	Valor de entrada
OUT	Output	SINT, INT, DINT, números en coma flotante	Valor absoluto de la entrada

- **MIN, MAX (Minimum, Maximum):** Estas instrucciones comparan los valores de las entradas disponibles y escriben el valor ya sea menor o mayor en la salida OUT. El número de entradas se puede ampliar en el cuadro de la instrucción mediante entradas adicionales. Las entradas se numeran de forma ascendente en el cuadro. Para la ejecución de la instrucción se deben indicar como mínimo dos valores de entrada y como máximo 100. Si existe un error en la ejecución de la operación, la salida ENO devuelve un 0 lógico; de lo contrario, devuelve un 1 lógico.

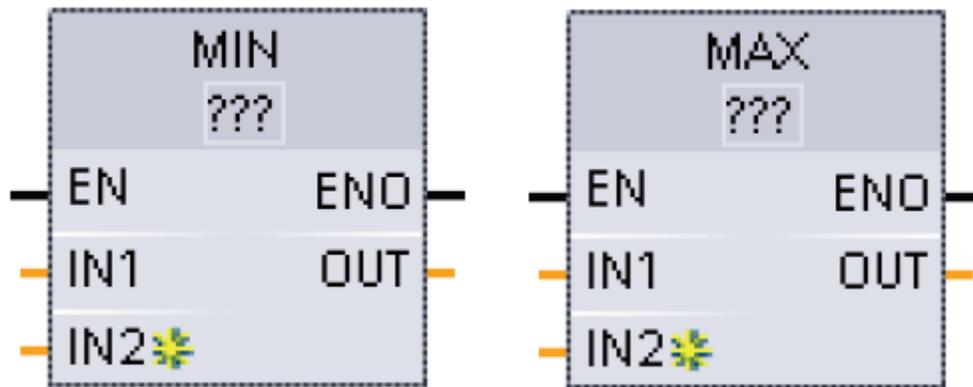


Figura 3.42 - Bloque de la instrucción MIN (izquierda) y de la instrucción MAX (derecha)

Tabla 3.20 - Parámetros de las instrucciones MIN y MAX

Parámetro	Declaración	Tipo de dato	Descripción
EN	Input	BOOL	Entrada de habilitación
ENO	Output	BOOL	Salida de habilitación
INn	Input	Enteros, números en punto flotante, DTL, DT	Valores de entrada a comparación
OUT	Output	Enteros, números en punto flotante, DTL, DT	Resultado, valor máximo/mínimo

- **LIMIT:** Es una instrucción de ajuste de valor entre límites. La instrucción limita el valor de la entrada IN a los valores de las entradas MN y MX. Si el valor de la entrada IN cumple la condición $MN \leq IN \leq MX$, se copia en la salida OUT (cumpliendo previamente que $MN \leq MX$). Si no se cumple la condición y el valor de entrada IN es menor que el límite inferior MN, la salida OUT adopta el valor límite MN. Si el

límite superior MX se rebasa por exceso, la salida OUT adopta el valor de la entrada MX. Si las variables indicadas no son del mismo tipo de datos, un operando no tiene un valor válido o el valor del límite inferior es mayor que el del límite superior, entonces la salida ENO devuelve un 0 lógico. Si la operación se realiza con éxito, el valor lógico de la salida ENO será de 1.



Figura 3.43 - Bloque de la instrucción LIMIT

Tabla 3.21 - Parámetros de la instrucción LIMIT

Parámetro	Declaración	Tipo de dato	Descripción
EN	Input	BOOL	Entrada de habilitación
ENO	Output	BOOL	Salida de habilitación
MN	Input	Enteros, números en punto flotante, TIME, TOD, DATE, DTL, DT	Límite inferior
IN	Input	Enteros, números en punto flotante, TIME, TOD, DATE, DTL, DT	Valor de entrada a comparar
MX	Input	Enteros, números en punto flotante, TIME, TOD, DATE, DTL, DT	Límite superior
OUT	Output	Enteros, números en punto flotante, TIME, TOD, DATE, DTL, DT	Resultado

- **Instrucciones matemáticas en punto flotante:** Las instrucciones en punto flotante sirven para programar operaciones matemáticas utilizando los tipos de datos Real o LReal únicamente. Con estas instrucciones es posible llevar a cabo operaciones matemáticas mucho más complejas y avanzadas.

Tabla 3.22 - Operaciones matemáticas en punto flotante

Operación	Resultado dado
SQR	Calcula el cuadrado de la entrada, $OUT = IN^2$
SQRT	Calcula la raíz cuadrada de la entrada, $OUT = \sqrt{IN}$
LN	Calcula el logaritmo natural de la entrada, $OUT = \ln(IN)$
EXP	Calcula el valor exponencial, $OUT = e^{IN}$
EXPT	(*) Elevar a una potencia $IN2$, $OUT = IN1^{IN2}$
FRAC	Determinar decimales (parte fraccionaria del número en punto flotante)
SIN	Calcula el valor del seno de IN , $OUT = \text{sen}(IN)$ en radianes
ASIN	Calcula el valor del arcoseno de IN , $OUT = \text{arcsen}(IN)$
COS	Calcula el valor OUT del coseno de IN , $OUT = \text{cos}(IN)$ en radianes
ACOS	Calcula el valor del arcocoseno de IN , $OUT = \text{arccos}(IN)$
TAN	Calcula el valor de la tangente de IN , $OUT = \text{tan}(IN)$ en radianes
ATAN	Calcula el valor de la arcotangente de IN , $OUT = \text{arctan}(IN)$

(*) Es la única instrucción con dos parámetros de entrada. Los parámetros de EXPT $IN1$ y OUT siempre son del mismo tipo de datos, que puede ser Real o LReal. Para el parámetro del exponente $IN2$ se puede elegir entre varios tipos de datos.

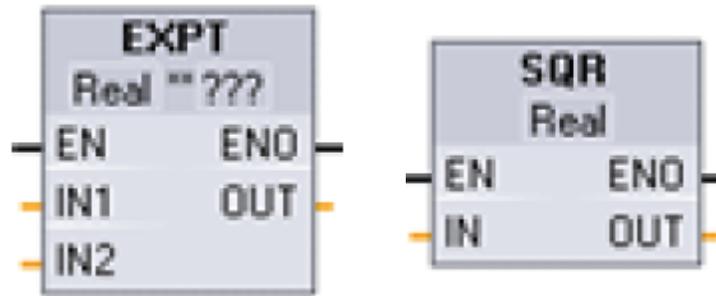


Figura 3.44 - Ejemplos de bloques de instrucciones matemáticas en punto flotante. Elevar a la potencia (izquierda) y o elevar al cuadrado (derecha)

Tabla 3.23 - Parámetros de las instrucciones matemáticas en coma punto flotante

Parámetro	Declaración	Tipo de dato	Descripción
EN	Input	BOOL	Entrada de habilitación
ENO	Output	BOOL	Salida de habilitación
IN	Input	Números en punto flotante	Valor de entrada
OUT	Output	Números en punto flotante	Resultado de la operación

4. Ejemplos de aplicación

A continuación, se presentarán breves ejemplos de aplicación de algunas de las funciones elementales en los que se describe y explica su funcionamiento, además de querer ilustrar aplicaciones que, si bien son bastante simples, como ejemplos didácticos son muy útiles para aquellos que nunca han usado un PLC.

a) Accionamiento de un motor con paro de emergencia.

Este representa una de las aplicaciones más básicas en cuanto manejo de entradas y salidas discretas se refiere. Se tiene un motor trifásico de 220 [V_{CA}], el cual se desea accionar mediante un botón pulsador que servirá como arranque. Cuando el botón es pulsado, el motor debe permanecer encendido con independencia de si el arranque se mantiene presionado o de si éste es presionado en repetidas ocasiones. Para detener el motor se requiere presionar un botón que funcionará como paro; si este se mantiene presionado, el motor no debe arrancar aún si se presiona el botón de arranque (paro prioritario).

27

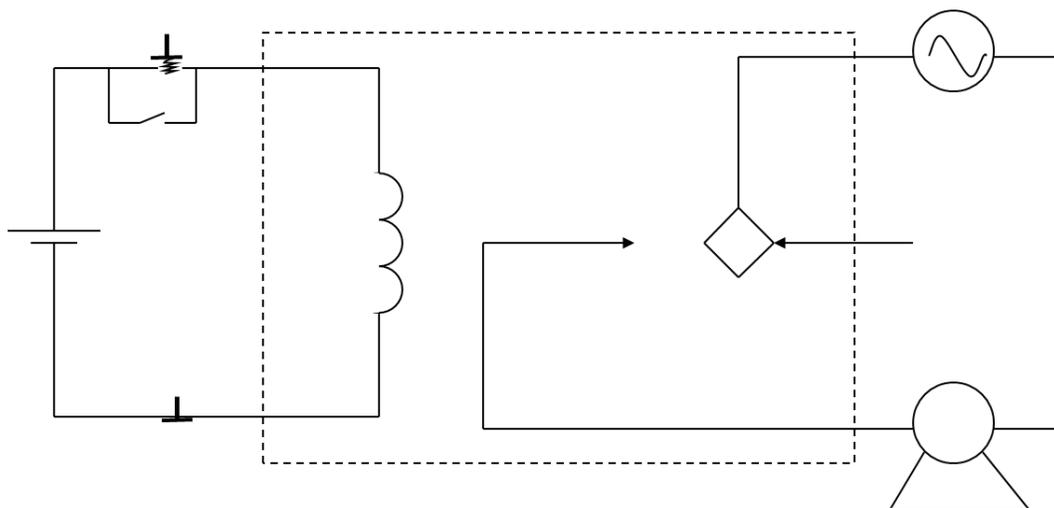


Figura 4.1 - Esquema del problema de arranque de un motor con paro prioritario.

En la Figura 4.1 se presenta el esquema de la situación mediante un circuito de relevador. Del lado izquierdo se tiene el circuito de control conformado por un botón pulsador normalmente abierto (parte superior); y del lado derecho, el circuito del motor con su respectiva alimentación. Cuando el inicio es presionado, la bobina del relevador se energiza

lo que hace que éste conmute y cierre el circuito entre el motor y la fuente de poder. Al presionar una vez el inicio, se cierra también el contacto paralelo al botón, por lo que, aunque se deje de pulsar el botón, la bobina queda continuamente energizada y el motor puede seguir encendido. Cuando el botón de paro del circuito de control (parte inferior) se presiona, el circuito se abre, por lo que la bobina se desenergiza, el relevador conmuta y el motor se apaga. Si el paro sigue presionado, por más que se presione el arranque, el circuito permanecerá abierto y la bobina no podrá ser energizada para accionar el motor.

Solución: Para comenzar, se plantea una tabla de entradas y salidas en la que se describen sus componentes y direcciones de programación en el PLC

Tabla 4.1 - Tabla de entradas y salidas para el arranque de un motor

Entrada/Salida	Etiqueta	Descripción	Dirección
Botón de inicio	Arranque	Botón pulsador NA	I1.0
Botón de paro	Paro	Botón enclavado NA	I1.1
Motor	Motor	Motor trifásico de 220 [V _{CA}]	Q1.0

El programa con lógica de escalera implementado con el software TIA Portal se muestra en la Figura 4.2

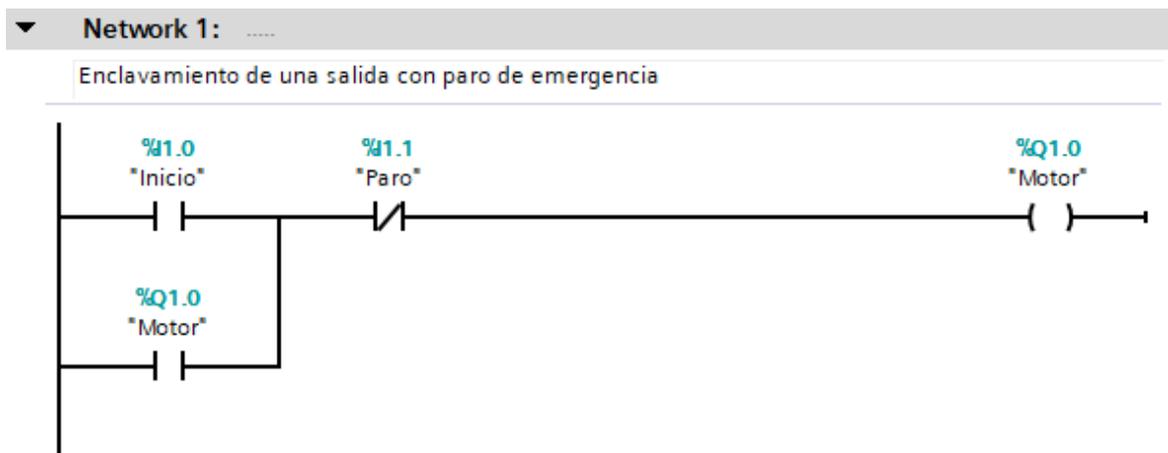


Figura 4.2 - Programa de arranque de un motor con TIA Portal

Cuando se activa la entrada I1.0 se cierra el contacto asociado, por lo que existe continuidad en el peldaño y la salida Q1.0 se activa. Simultáneamente, el contacto asociado a la salida Q1.0 se cierra, por lo que, aunque se abra el contacto de la entrada I1.0, la salida del peldaño puede seguir encendida. Si se activa la entrada I1.1, el contacto asociado se abre y no permite cerrar el peldaño aún si se activa la entrada I1.0 y la salida nunca se activará.

b) Semáforo con temporización de retardo a la conexión

En este ejemplo se desea programar un semáforo vehicular simple. Se tiene un botón enclavado que dará inicio al ciclo del semáforo, el cual comenzará con el encendido de la luz verde durante 5 [s]; posteriormente, la luz verde se apagará y se encenderá la luz ámbar durante 3 [s]. Transcurrido el tiempo de encendido de la luz ámbar, ésta se apagará e inmediatamente después se encenderá la luz roja durante 5 [s]. Si el botón de inicio se mantiene activado, el ciclo se repetirá indefinidamente. Si el botón de inicio se desactiva en cualquier momento del ciclo, éste terminará de completar el encendido de las luces restantes y no iniciará la secuencia de nuevo.

Solución: De la misma manera que en el ejemplo anterior, se plantea una tabla de entradas y salidas para la organización de la información dada.

Tabla 4.2 - Tabla de entradas y salidas para el semáforo vehicular

Entrada/Salida	Etiqueta	Descripción	Dirección
Botón de inicio	Inicio	Botón enclavado NA	I1.0
Luz verde (Siga)	LuzVerde	Lámpara de 120 [V _{CA}]	Q2.0
Luz ámbar (Precaución)	LuzAmbar	Lámpara de 120 [V _{CA}]	Q2.1
Luz verde (Alto)	LuzRoja	Lámpara de 120 [V _{CA}]	Q2.2

En las siguientes figuras se muestra la programación del ejercicio en TIA Portal haciendo uso de la instrucción TON (para más información, consultar la sección 3.3).

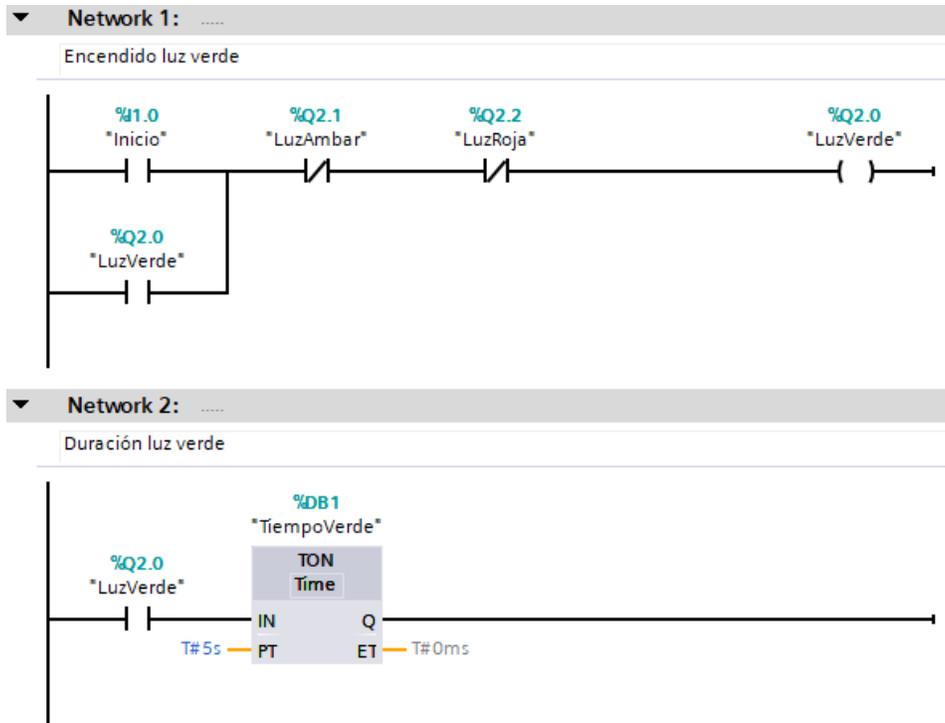


Figura 4.3 - Encendido y tiempo de duración de la luz verde

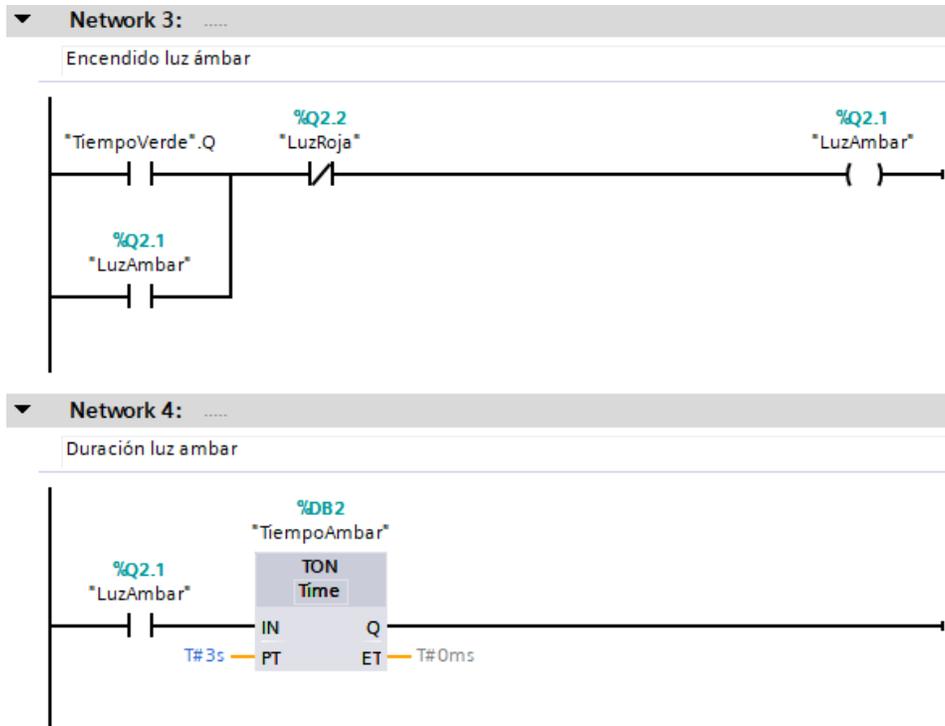


Figura 4.4 - Encendido y tiempo de duración de la luz ámbar

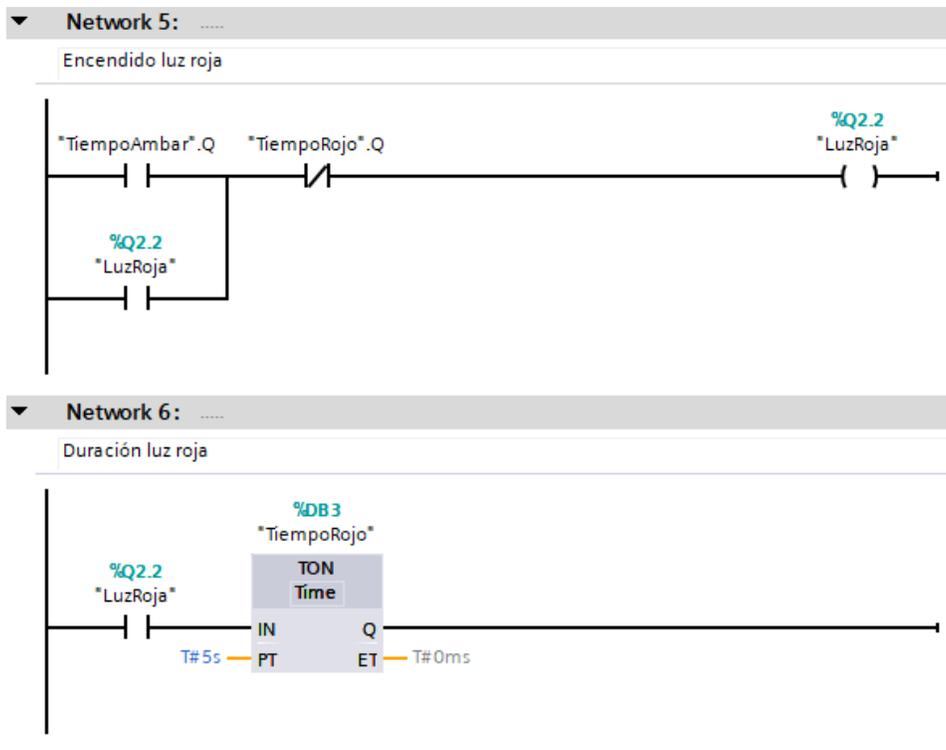


Figura 4.5 - Encendido y tiempo de duración de la luz roja

En la Figura 4.3 se muestra el funcionamiento de la luz verde. Cuando el botón de inicio es accionado se activa la salida correspondiente a la luz verde y ésta se auto enclava. La misma salida de la luz verde se asocia con un contacto que servirá como señal de entrada a un primer temporizador de tipo TON llamado “TiempoVerde”, el cual se accionará a los 5 [s] según lo planteado en el ejercicio. La señal de salida de dicho temporizador se usa para activar la salida asociada a la luz ámbar (Figura 4.4), la cual se auto enclava para poder permanecer encendida. Cuando la luz ámbar se enciende, se abre el contacto asociado a ésta en el peldaño de accionamiento de la luz verde, por lo que mientras esté prendida la luz ámbar, la luz verde no podrá ser accionada aún si el botón de inicio está accionado.

Adicionalmente, un contacto asociado a la luz ámbar se usa como señal de inicio de un segundo temporizador TON llamado “TiempoAmbar”, que tiene un tiempo programado de 3 [s]. Cuando ha pasado el tiempo solicitado, el contacto asociado a la salida del temporizador se activa por lo que enciende la salida de la luz roja (Figura 4.5). Cuando la luz roja se enciende, de manera similar a lo que ocurre con la luz ámbar, se abre el contacto asociado a ésta en el peldaño de accionamiento de la luz ámbar, por lo que ésta se apaga. Paralelamente,

se abre también el contacto asociado a la luz roja en el peldaño de accionamiento de la luz verde, por lo que mientras esté prendida la luz roja, la luz verde no podrá ser accionada aún si el botón de inicio está accionado.

Finalmente, otro contacto asociado a la luz roja funciona como señal de inicio de un tercer temporizador TON llamado “TiempoRojo”, que tiene un tiempo programado de 5 [s]. Cuando ha pasado el tiempo programado, el contacto asociado a la salida, que se encuentra en el peldaño de activación de la luz roja, se abre, por lo que esta luz se apaga. Debido a que tanto la luz roja como la luz ámbar están apagadas, la luz verde puede volver a encenderse si el botón de inicio sigue accionado, con lo que entonces se repite el ciclo.

c) Semáforo con temporización de impulso

Para este ejemplo se plantea exactamente la misma situación que en el ejercicio anterior, la diferencia radica en que la solución que se propone usar otro tipo de temporizador, un impulso (sección 3.3).

Solución: La tabla de entradas y salidas es exactamente la misma que la propuesta en el ejercicio anterior. Ahora bien, la programación en TIA Portal se modifica y se muestra en las figuras siguientes.

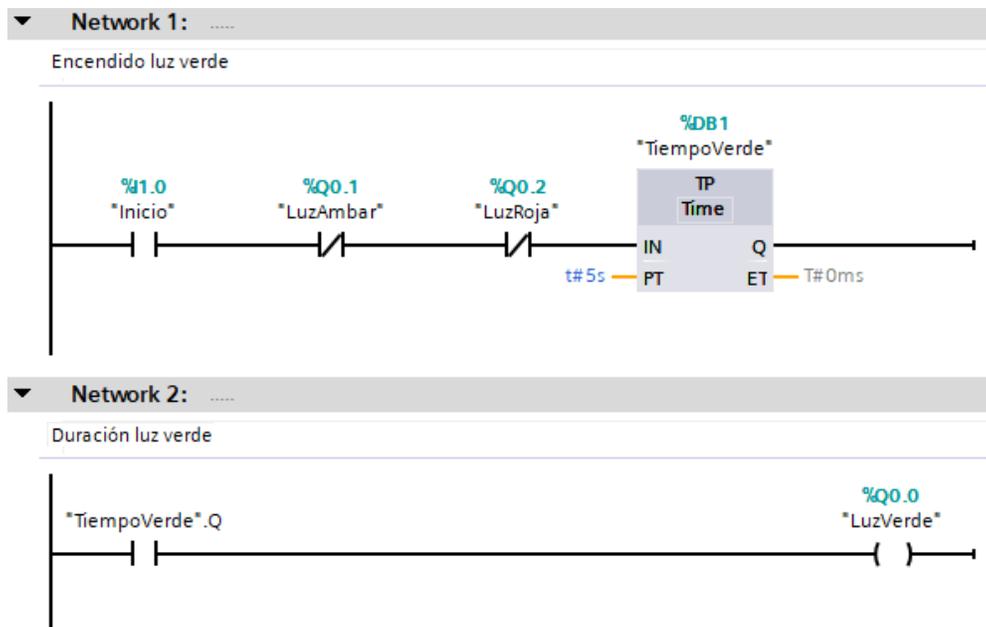


Figura 4.6 - Encendido y duración de la luz verde

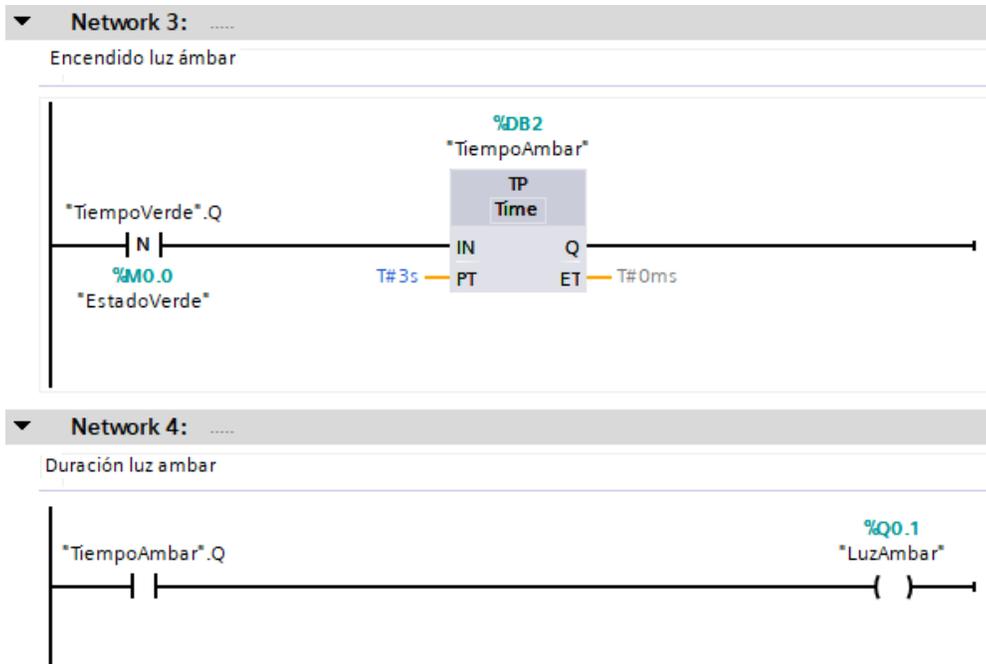


Figura 4.7 - Encendido y duración de la luz ámbar

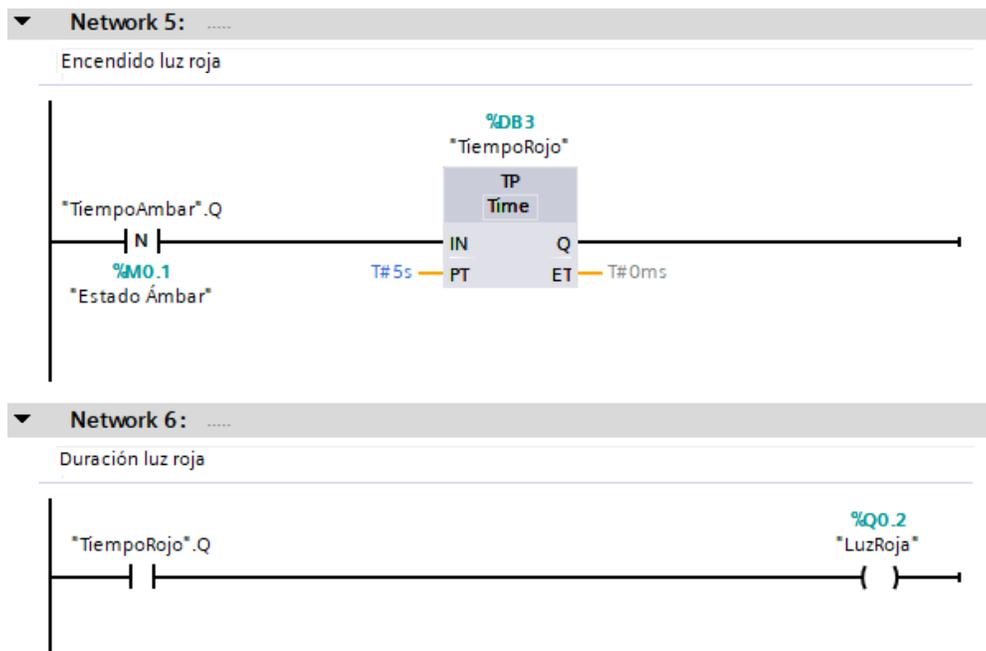


Figura 4.8 - Encendido y duración de la luz roja

La interpretación del diagrama de escalera difiere a la del ejercicio anterior. Cuando se presiona el botón pulsador, la instrucción de impulso TP “TiempoVerde” se activa, por lo que el contacto asociado a su salida permanecerá cerrado el tiempo programado en el temporizador, es decir, 5 [s], con lo que se encenderá la luz verde durante ese tiempo (Figura 4.6).

Ahora bien, inicialmente la salida del TP “TiempoVerde” estaba en estado 0 lógico y al activarse pasa a estar en estado 1 lógico, por lo que se tiene un flanco ascendente (o positivo). Cuando el tiempo del impulso termina, la salida del TP pasa de 1 lógico a 0 lógico, con lo que se tiene ahora un flanco descendente (o negativo). Es este último cambio de flanco con el que se da la señal de inicio para un segundo temporizador TP llamado “TiempoAmbar” con una duración de 3 [s] (Figura 4.7). Dado que los temporizadores TP no requieren una señal de entrada constante, resulta irrelevante que la activación del contacto asociado a la salida del temporizador “TiempoVerde” dure un único *scan*. En el contacto detección de flanco descendente de la entrada del TP “TiempoAmbar” se visualiza que se requieren dos entradas: la señal de la que se quiere detectar el cambio de flanco y una variable auxiliar que funcione como área de memoria para guardar el estado anterior de la señal y poder hacer la comparación.

Cuando el TP “TiempoAmbar” recibe la señal de detección de flanco descendente por parte de la salida del TP anterior, se activa la señal de salida asociada a la luz roja. A continuación, de manera análoga a lo que ocurrió con la luz ámbar, se hace la detección de flanco descendente de la salida del TP “TiempoAmbar” para poder activar un tercer TP llamado “TiempoRojo” (Figura 4.8). Una vez activado este TP, su salida se mantendrá activa durante 5 [s] con lo que se encenderá la luz roja. Cuando haya terminado el tiempo establecido del último TP, la luz roja se apagará, con lo que se cerrará el contacto asociado a esta salida en el bloque de activación de la señal de inicio del TP “TiempoVerde” y, si el botón de inicio está accionado, el ciclo puede comenzar nuevamente.

Resultados

Cuando se revisan ofertas laborales que competen al área de automatización industrial, no resulta raro encontrar que las empresas buscan profesionistas que estén capacitados no solo en la programación de PLC, sino que muchas veces requieren personal capacitado en la programación de equipos de las marcas Siemens y Allen-Bradley principalmente. Debido a ello resultaba urgente que el laboratorio de Automatización Industrial modernizara sus instalaciones y equipos con el fin de que el alumnado pueda adquirir las nociones y la experiencia de programación con equipos mucho más recientes y que fueran de marcas líderes en la industria nacional e internacional. De no ser así, existiría un rezago en las competencias de los egresados lo que dificultaría su inserción en el campo laboral y perderían competitividad.

Así mismo, el laboratorio de Automatización Industrial, junto con más de 20 laboratorios de la Facultad de Ingeniería, se encuentra certificado por parte de Certificación Mexicana S.C. con la norma NMX-CC-900-IMNC-2015 ISO 9001:2015. Esto implica un gran compromiso para mantener altos estándares de calidad no sólo en las instalaciones de los laboratorios, sino también en el personal docente que trabaja en ellos y que participa directamente en el proceso de enseñanza-aprendizaje. Resulta entonces más evidente la utilidad de contar con documentación como la presente dado que funciona al mismo tiempo como un material de apoyo a la docencia y como una evidencia del constante esfuerzo que hace la institución por continuar brindando una enseñanza de calidad.

Un primer borrador de este trabajo fue empleado para servir como base en la impartición de un curso DGAPA en el intersemestral 2019-2 bajo el nombre de *Actualización para impartir la asignatura Automatización Industrial*, el cual tenía como objetivo la capacitación y actualización del personal docente en el uso y programación del nuevo equipo adquirido por el laboratorio para la impartición de la asignatura de Automatización Industrial. A partir de este curso, y tomando en cuenta el propio contenido temático de la asignatura, se hicieron las modificaciones pertinentes para continuar con este esfuerzo de documentación.

Discusión

El presente documento tenía planteado como objetivo la creación de apuntes sobre los componentes, la programación y el uso básico de los equipos Siemens S7-1200 recientemente adquiridos por el laboratorio. Este texto ha cubierto la descripción elemental de las características del nuevo equipo y del software de su programación, haciendo especial énfasis en la conexión y programación de instrucciones básicas que cubran las necesidades planteadas a su vez por el objetivo de la asignatura de Automatización Industrial.

Debido a que el nuevo equipo quedará en funcionamiento y en servicio para la impartición de la asignatura a partir del semestre 2020-2, no ha sido posible tener una retroalimentación por parte de los alumnos respecto a la utilidad del presente material. Sin embargo, como se comentó brevemente en la sección anterior, esta documentación sirvió como base para organizar e impartir un curso DGAPA de actualización de profesores, en el cual se tuvo una muy buena respuesta por parte de los asistentes. Dicho curso pudo ser impartido en tiempo y forma de acuerdo con los contenidos planeados e inclusive se tuvo la oportunidad de brindar nociones, aunque muy elementales, sobre el funcionamiento y manejo no solo del equipo de PLC, sino de las interfaces humano-máquina (HMI) también adquiridas por el laboratorio. Esto puede considerarse como un primer acierto en torno a la pertinencia de los contenidos aquí desarrollados.

De manera complementaria, este trabajo tiene la intención de servir como apoyo a los profesores del laboratorio en la creación de nuevo material didáctico para la impartición de sus clases debido a la transición al nuevo equipo. Con esto, los docentes tienen una guía de los contenidos y formas de trabajo del equipo para la creación de, por ejemplo, dispositivas, ejercicios más complejos que hagan uso de las diferentes instrucciones que el software ofrece, o bien modificación de las prácticas concernientes al tema de PLC en el contenido de la asignatura.

Este trabajo puede verse como una primera iteración de una documentación que debe estar en constante cambio y actualización para responder a las necesidades de los alumnos y de los usuarios del laboratorio en general. A lo largo del tiempo en que se ha trabajado en este material, se han encontrado diversas áreas de oportunidad con las que se puede enriquecer el texto. En primer lugar, se ha considerado la posibilidad de incluir un apartado de “resolución de problemas” (*troubleshooting*, por el término en inglés) en el que se

mencionen la solución de los problemas más comunes a los que se puede enfrentar un usuario al estar haciendo uso del equipo como pueden ser errores al establecer la comunicación, duplicidad de direcciones IP o direcciones incompatibles, manejo incorrecto de instrucciones, desconexión de la comunicación debido a fallos energéticos, tan solo por mencionar algunos. Así, los usuarios tendrían una guía rápida que les permita hacer las correcciones necesarias de manera independiente sin tener que recurrir inmediatamente a buscar ayuda ya sea del personal del laboratorio o de otros medios como lo es el internet.

Junto con el software TIA Portal, entre la paquetería adquirida, se tiene el programa PLCSIM de Siemens, el cual es un software que permite la simulación de los programas en un entorno virtual. Esta herramienta puede resultar muy útil sobre todo en los primeros momentos de la enseñanza del manejo del equipo para poder adaptarse a su uso y reducir la probabilidad de que se comentan errores al momento de utilizar el equipo físico. Por lo que, en segundo lugar, se ha planteado la opción de incluir un apartado sobre el manejo del simulador que cumpla con los mismos objetivos de ofrecer las bases elementales para el uso de dicha herramienta.

Este texto ha sido posible gracias al apoyo del proyecto PAPIME PE107719 que lleva por título *Diseño e implementación de prácticas de laboratorio para la asignatura Automatización Avanzada*, el cual busca el desarrollo de un manual de prácticas que atienda a las necesidades de la asignatura de nueva creación, Automatización Avanzada, la cual también se busca incorporar al Sistema de Gestión de Calidad del cual el laboratorio forma parte. En dicha asignatura, de carácter optativo, se abordan los temas de programación avanzada de PLC, manejo de HMI y nociones básicas de redes industriales. Debido a ello se plantea como tercera y última área de oportunidad, el abordaje de los temas concernientes a la asignatura de Automatización Avanzada, con lo que entonces este trabajo se extendería a las instrucciones más avanzadas con las que se cuenta en el software como son manejo de cadenas y caracteres, implementación de ciclos, o bien el uso y manejo de subrutinas para la estructuración de programas. Adicionalmente, se buscaría añadir también un apartado sobre el uso y manejo de las HMI con las que se cuentan para que, de la misma manera que con los PLC, alumnos y profesores tengan una guía rápida del funcionamiento, puesta en marcha y uso de dichos recursos para el mejor aprovechamiento de su paso por el laboratorio.

En suma, tomando en cuenta los puntos mencionados y los que se puedan considerar a lo largo del tiempo, este documento eventualmente dejaría de ser una serie de apuntes básicos, sino pasarían a ser un manual integral del uso de los equipos de PLC del laboratorio, el cual estaría conformado con una descripción más amplia de las diversas capacidades con las que cuenta el equipo tanto en software como en aplicaciones.

En un futuro cercano, se tiene ya programada la impartición de un nuevo curso DGAPA que tiene como fin formar a los asistentes en el uso básico tanto de los PLC como de las HMI, abordando desde temas elementales como son la conexión del PLC con la computadora y la programación de instrucciones básicas, a tópicos más complejos como son las redes y la comunicación entre varios PLC. Para un futuro un poco más distante, se desea ofrecer cursos de distintos niveles tanto a alumnos como a profesores interesados en el manejo de los PLC, además de inclusive ofrecer cursos de capacitación para la industria, con los cuales se busca generar ingresos extraordinarios para así continuar adquiriendo nuevos equipos y mejorar las instalaciones del laboratorio. Con esto se busca que no solo el laboratorio de Automatización Industrial, sino la Facultad y la misma Universidad, como instituciones, continúen manteniéndose a la vanguardia en la formación de profesionales que sean capaces de dar respuesta a los retos y demandas que se tienen en la sociedad contemporánea.

Índice de figuras

Figura 1.1 - Vista frontal de un PLC S7-1200 con un CPU 1211C	9
Figura 1.2 - Vista frontal de un PLC con CPU 1215C	11
Figura 1.3 - Vista del módulo adicional de E/S.....	12
Figura 1.4 - Interfaz HMI 1200 Basic	13
Figura 2.1 - Visualización de la vista de portal en S7	15
Figura 2.2 - Visualización de la vista de proyecto en S7	16
Figura 2.3 - Panel de creación de nuevo proyecto*.....	17
Figura 2.4 - Tarea del portal Dispositivos y redes para añadir un CPU*	17
Figura 2.5 - Tarea del portal Dispositivos y redes para añadir un CPU*	18
Figura 2.6 - Ventana de selección de dispositivo*	18
Figura 2.7 - Selección del CPU correspondiente al que se tiene en el laboratorio*.....	19
Figura 2.8 - Rack y CPU en el espacio de trabajo al seleccionar el dispositivo deseado*..	20
Figura 2.9 - CPU sin especificar en el catálogo de series 1200*.....	21
Figura 2.10 - CPU sin especificar en el área de trabajo*	22
Figura 2.11 - Ventana de selección de los CPU detectados en la red*.....	23
Figura 2.12 - Ventana de catálogo de hardwar*	24
Figura 2.13 - Módulo E/S conectado al rack del CPU*	24
Figura 2.14 - Ventana de configuración de parámetros del CPU*	25
Figura 2.15 - Ventana de configuración de los parámetros de un módulo E/S*	27
Figura 2.16 - Esquema de una conexión Ethernet*	28
Figura 2.17 - Ventana de configuración de los parámetros de red*	29
Figura 2.18 - Opciones para cargar un programa al PLC*	30
Figura 2.19 - Opciones de carga del programa al PLC*	31

Figura 2.20 - Cuadro de diálogo Carga avanzada*	32
Figura 2.21 - Vista preliminar de carga sin advertencias*	33
Figura 2.22 - Ventana de Resultados de operación de carga*	34
Figura 2.23 - Operación de carga exitosa sin errores ni advertencias*	34
Figura 2.24 - Representación de las diferentes tipologías de programación. Lineal (izquierda), por áreas (centro) y estructurada (derecha)	36
Figura 2.25 - Árbol de proyecto con la opción para añadir nuevos bloques*	37
Figura 2.26 - Ventana Agregar nuevo bloque con la respectiva configuración*	38
Figura 2.27 - Área de trabajo programación de bloques en TIA Portal*	39
Figura 2.28 - Panel de Instrucciones con la organización de los diferentes bloques de programación*	40
Figura 2.29 - Ejemplo del área de trabajo con contactos en el primer segmento*	40
Figura 2.30 - Árbol de proyecto con la opción de añadir tabla de variables *	41
Figura 2.31 - Tabla de variables (en blanco)*	42
Figura 2.32 - Esquema de contactos con variables asociadas*	43
Figura 2.33 - Ejemplo de variables declaradas en la tabla de variables*	43
Figura 2.34 - Esquema de ejecución de los estados de ARRANQUE y RUN	50
Figura 2.35 - Panel de configuración del tiempo de ciclo*	52
Figura 2.36 - Resumen del ciclo de ejecución del programa de usuario	53
Figura 3.1 - Instrucciones generales para la construcción de un diagrama de escalera*	54
Figura 3.2 - Contacto NA (izquierda) y NC (derecha) en TIA Portal	55
Figura 3.3 - Bobina directa (izquierda) y bobina invertida (derecha)	55
Figura 3.4 - Activar salida (izquierda) y desactivar salida (derecha) en TIA Portal	56
Figura 3.5 - Consulta de flanco ascendente (izquierda) y flanco descendente (derecha)	56

Figura 3.6 - Activación por flanco ascendente (izquierda) y activación por flanco descendente (derecha).....	57
Figura 3.7 - Esquema del área de memoria de marcas	58
Figura 3.8 - Tabla de variables. Existe una variable con una marca asignada*	58
Figura 3.9 - Contacto con la variable asociada a la marca anterior*	59
Figura 3.10 - Bloque del temporizador TP*	60
Figura 3.11 - Gráfica de tiempos de la entrada IN y la salida Q para la instrucción TP	61
Figura 3.12 - Bloque del temporizador TON*	62
Figura 3.13 - Gráfica de tiempos de la entrada IN y la salida Q para la instrucción TON ..	62
Figura 3.14 - Bloque del temporizador TOF*	63
Figura 3.15 - Gráfica de tiempos de la entrada IN y la salida Q para la instrucción TOF ...	63
Figura 3.16 - Bloque del temporizador TONR*	65
Figura 3.17 - Gráfica de tiempos de las señales IN, R, Q, ET y PT para la instrucción TONR	65
Figura 3.18 - Bloque de la instrucción CTU*	67
Figura 3.19 - Esquema de tiempo de las señales CU, R, CV y Q	67
Figura 3.20 - Bloque de la instrucción CTD*	68
Figura 3.21 - Esquema de tiempos de las señales CD, LD, CV y Q	69
Figura 3.22 - Bloque de la instrucción CTUD*	70
Figura 3.23 - Esquema de tiempos para las señales CU, CD, R, LD; CV, QU, y QD	71
Figura 3.24 - Se muestra cómo el contenido de un operando del tipo Word se desplaza cuatro posiciones de bit hacia la derecha.....	72
Figura 3.25 - Bloque de la instrucción SHR*	73
Figura 3.26 - Bloque de la instrucción SHL*	74

Figura 3.27 - Se muestra cómo el contenido de un operando del tipo Word se desplaza seis posiciones de bit hacia la izquierda	74
Figura 3.28 - Se muestra cómo el contenido de un operando del tipo DWord rota tres posiciones hacia la derecha.....	75
Figura 3.29 - Bloque de la instrucción ROR*	75
Figura 3.30 - Se muestra cómo el contenido de un operando del tipo DWord rota tres posiciones hacia la izquierda	76
Figura 3.31 - Bloque de la instrucción ROL*	77
Figura 3.32 - Tabla de constantes de usuario*	78
Figura 3.33 - Contacto de comparación (igualdad) en TIA Portal. Se especifica que se van a comparar datos del tipo Byte. El parámetro superior es el primer argumento; y el inferior, es el segundo argumento.....	79
Figura 3.34 - - Bloque de comparación dentro de rango de valores enteros (izquierda) y fuera de rango de valores enteros signados (derecha)*	80
Figura 3.35 - Contacto de comprobación de validez (izquierda) y de invalidez (derecha). El único parámetro de entrada es el valor a comparar	81
Figura 3.36 - Bloque de instrucción CALCULATE*.....	82
Figura 3.37 - Cuadro de diálogo de edición de la función CALCULATE*.....	82
Figura 3.38 - Bloque de instrucción ADD*	84
Figura 3.39 - Bloque de la instrucción NEG*	85
Figura 3.40 - Bloque de instrucción INC (izquierda) y de instrucción DEC (derecha) *	85
Figura 3.41 - Bloque de la instrucción ABS*.....	86
Figura 3.42 - Bloque de instrucción MIN (izquierda) y de instrucción MAX (derecha) * ..	87
Figura 3.43 - Bloque de la instrucción LIMIT*	88

Figura 3.44 - Ejemplos de bloques de instrucciones matemáticas en punto flotante. Elevar a la potencia (izquierda) y o elevar al cuadrado (derecha)*	90
Figura 4.1 - Esquema del problema de arranque de un motor con paro prioritario.....	91
Figura 4.2 - Programa de arranque de un motor con TIA Portal.....	92
Figura 4.3 - Encendido y tiempo de duración de la luz verde	94
Figura 4.4 - Encendido y tiempo de duración de la luz ámbar.....	94
Figura 4.5 - Encendido y tiempo de duración de la luz roja.....	95
Figura 4.6 - Encendido y duración de la luz verde.....	96
Figura 4.7 - Encendido y duración de la luz ámbar.....	97
Figura 4.8 - Encendido y duración de la luz roja.....	97

NOTA: Las imágenes marcadas con asterisco (*) son de elaboración propia

Índice de tablas

Tabla 1.1 - Características del CPU 1215C.....	10
Tabla 1.2 - Características de los módulos E/S SM 1223	12
Tabla 1.3 - Características de la HMI KTP1200	13
Tabla 2.1 - Características de las variables globales y locales del PLC S7-1200	44
Tabla 2.2 - Tipos de datos bit y secuencia de bits	45
Tabla 2.3 - Tipos de datos enteros	46
Tabla 2.4 - Tipos de datos reales	46
Tabla 2.5 - Tipos de datos de fecha y hora.....	47
Tabla 2.6 - Tipos de carácter y cadena	48
Tabla 2.7 - Tipo de datos de arreglos y matrices.....	49
Tabla 2.8 - Rangos para el tiempo de ciclo	52
Tabla 3.1 - Parámetros para el temporizador TP	60
Tabla 3.2 - Parámetros para el temporizador TON	62
Tabla 3.3 - Parámetros para el temporizador TOF.....	64
Tabla 3.4 - Parámetros para el temporizador TONR.....	66
Tabla 3.5 - Parámetros para contador CTU	68
Tabla 3.6 - Parámetros para contador CTD.....	69
Tabla 3.7 - Parámetros para contador CTUD.....	71
Tabla 3.8 - Parámetros para el registro de corrimiento SHR	73
Tabla 3.9 - Parámetros para el registro de corrimiento SHL.....	74
Tabla 3.10 - Parámetros para la rotación ROR.....	76
Tabla 3.11 - Parámetros para la rotación ROL.....	77
Tabla 3.12 - Descripción de comparaciones aritméticas	78

Tabla 3.13 - Parámetros para las comparaciones de rango.....	80
Tabla 3.14 - Parámetros de la instrucción CALCULATE	83
Tabla 3.15 - Descripción de las operaciones aritméticas básicas	83
Tabla 3.16 - Parámetros de las instrucciones aritméticas	84
Tabla 3.17 - Parámetros de la instrucción NEG	85
Tabla 3.18 - Parámetros de las instrucciones INC y DEC.....	86
Tabla 3.19 - Parámetros de la instrucción ABS.....	86
Tabla 3.20 - Parámetros de las instrucciones MIN y MAX	87
Tabla 3.21 - Parámetros de la instrucción LIMIT	88
Tabla 3.22 - Operaciones matemáticas en punto flotante.....	89
Tabla 3.23 - Parámetros de las instrucciones matemáticas en coma punto flotante.....	90
Tabla 4.1 - Tabla de entradas y salidas para el arranque de un motor.....	92
Tabla 4.2 - Tabla de entradas y salidas para el semáforo vehicular	93