

# UNIVERSIDAD TECNOLÓGICA ISRAEL



## CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES

### TEMA:

Estudio, Diseño e Implementación de un Laboratorio y Guía de Prácticas con PLCs para la Carrera de Electrónica y Telecomunicaciones de la Universidad Israel

Trabajo de Graduación previo a la obtención del título de Ingeniero en Electrónica y Telecomunicaciones

### AUTOR

Karla Pamela Hidalgo Suárez

### TUTOR

Ing. Fabrizio Villasís

Quito Ecuador

Noviembre 2012

# GUÍA DE PRÁCTICAS

## ÍNDICE DE CONTENIDO

<b>PRÁCTICA No. 1</b> .....	1
Conociendo los PLCs Siemens S7-1200.....	1
<b>PRÁCTICA No. 2</b> .....	5
Conociendo el entorno de Programación de STEP 7.....	5
<b>PRÁCTICA No. 3</b> .....	11
Aprendiendo el funcionamiento del PLC Siemens S7-1200.....	11
<b>PRÁCTICA No. 4</b> .....	19
Programando al PLC Siemens S7-1200.....	19
<b>PRÁCTICA No. 5</b> .....	39
Temporizadores.....	39
<b>PRÁCTICA No. 6</b> .....	52
Flip-flops. ....	52

## PRÁCTICA No. 1

**NOMBRE (S):**

**FECHA DEL LABORATORIO:**

### 1. TÍTULO

Conociendo los PLCs Siemens S7-1200.

### 2. OBJETIVOS

#### 2.1 GENERAL

Identificar las partes del PLC Siemens S7-1200.

#### 2.2 ESPECÍFICOS

Ubicar físicamente todos los puertos que constituyen al PLC Siemens S7-1200.

### 3. LISTA DE MATERIALES Y EQUIPOS

- Modulo PLC S7-1200.

### 4. INDICACIONES GENERALES

Utilice un archivo electrónico para escribir sus respuestas y envíelo por correo a la siguiente dirección: *e-mail\_del\_profesor*.

Este símbolo ♣ indica que debe responder el correspondiente numeral y entregar el resultado en el archivo electrónico.

## 5. PROCEDIMIENTO

**5.1** Tome y analice el PLC. Considere que el controlador S7-1200 ofrece la flexibilidad y potencia necesarias para controlar una gran variedad de dispositivos para las distintas necesidades de automatización. Gracias a su diseño compacto, configuración flexible y amplio juego de instrucciones, el S7-1200 es idóneo para controlar una gran variedad de aplicaciones.

La CPU incorpora un microprocesador, una fuente de alimentación integrada, circuitos de entrada y salida, PROFINET (Red LAN) integrado, E/S de control de movimiento de alta velocidad y entradas analógicas incorporadas, todo ello en una carcasa compacta, conformando así un potente controlador. La CPU contiene la lógica necesaria para vigilar y controlar los dispositivos de la aplicación. La CPU vigila las entradas y cambia el estado de las salidas según la lógica del programa de usuario, que puede incluir lógica booleana, instrucciones de contaje y temporización, funciones matemáticas complejas, así como comunicación con otros dispositivos inteligentes.

Para comunicarse con una programadora, la CPU incorpora un puerto PROFINET integrado. La CPU puede comunicarse con paneles HMI (Interacción humano-máquina) o una CPU diferente en la red PROFINET.

Para garantizar seguridad en la aplicación, todas las CPUs S7-1200 disponen de protección por contraseña, que permite configurar el acceso a sus funciones.

## Tipos de CPUs

El PLC Siemens S7-1200 tiene una CPU 1214C.

Función	CPU 1211C	CPU 1212C	CPU 1214C
Dimensiones físicas (mm)	90 x 100 x 75	90 x 100 x 75	110 x 100 x 75
Memoria de usuario <ul style="list-style-type: none"> <li>Memoria de trabajo</li> <li>Memoria de carga</li> <li>Memoria remanente</li> </ul>	<ul style="list-style-type: none"> <li>25 KB</li> <li>1 MB</li> <li>2 KB</li> </ul>	<ul style="list-style-type: none"> <li>25 KB</li> <li>1 MB</li> <li>2 KB</li> </ul>	<ul style="list-style-type: none"> <li>50 KB</li> <li>2 MB</li> <li>2 KB</li> </ul>
E/S integradas locales <ul style="list-style-type: none"> <li>Digital</li> <li>Analógico</li> </ul>	<ul style="list-style-type: none"> <li>6 entradas 4 salidas</li> <li>2 entradas</li> </ul>	<ul style="list-style-type: none"> <li>8 entradas 6 salidas</li> <li>2 entradas</li> </ul>	<ul style="list-style-type: none"> <li>14 entradas 10 salidas</li> <li>2 entradas</li> </ul>
Tamaño de la memoria imagen de proceso <ul style="list-style-type: none"> <li>Entradas</li> <li>Salidas</li> </ul>	<ul style="list-style-type: none"> <li>1024 bytes</li> <li>1024 bytes</li> </ul>	<ul style="list-style-type: none"> <li>1024 bytes</li> <li>1024 bytes</li> </ul>	<ul style="list-style-type: none"> <li>1024 bytes</li> <li>1024 bytes</li> </ul>
Área de marcas (M)	4096 bytes	4096 bytes	8192 bytes
Ampliación con módulos de señales	Ninguno	2	8
Signal Board	1	1	1
Módulos de comunicación	3	3	3
Contadores rápidos <ul style="list-style-type: none"> <li>Fase simple</li> <li>Fase en cuadratura</li> </ul>	<ul style="list-style-type: none"> <li>3</li> <li>3 a 100 kHz</li> <li>3 a 80 kHz</li> </ul>	<ul style="list-style-type: none"> <li>4</li> <li>3 a 100 kHz 1 a 30 kHz</li> <li>3 a 80 kHz 1 a 20 kHz</li> </ul>	<ul style="list-style-type: none"> <li>6</li> <li>3 a 100 kHz 3 a 30 kHz</li> <li>3 a 80 kHz 3 a 20 kHz</li> </ul>
Salidas de impulsos <sup>1</sup>	2	2	2
Memory Card (opcional)	Sí	Sí	Sí
Tiempo de respaldo del reloj en tiempo real	Típico: 10 días / Mínimo: 6 días a 40 °C		
Velocidad de ejecución de funciones matemáticas con números reales	18 µs/instrucción		
Velocidad de ejecución booleana	0,1 µs/instrucción		

<sup>1</sup>Solo las CPUs con salida DC (sin relé) soportan las salidas de impulsos.

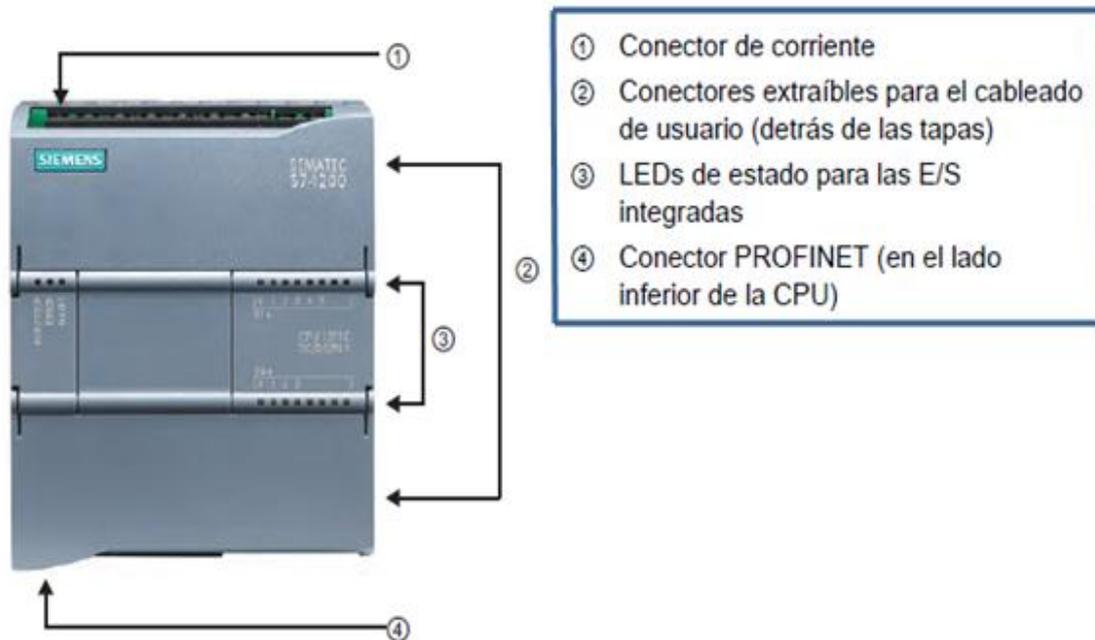
Los diferentes modelos de CPUs ofrecen una gran variedad de funciones y prestaciones que permiten crear soluciones efectivas destinadas a numerosas aplicaciones.

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

♣ ¿Cuántas entradas y salidas analógicas tiene el PLC?

**5.2** Abrir las tapas frontales del PLC e identifique los nombres de los conectores hembras con conexión por tornillo, conocidos como conectores extraíbles.

En la siguiente figura ubique todas las partes que constituyen el PLC.



*Figura 1. Partes de un PLC*

**5.3** ♣ ¿Cuántas entradas y salidas digitales tiene el PLC S7-1200?

**5.4** ♣ ¿Qué tipo de CPU utiliza? Y ¿es el más potenciado de su familia?

## 6. CONCLUSIONES Y RECOMENDACIONES

**6.1.** Exponga sus conclusiones respecto a la presente práctica.

**6.2.** Exponga sus recomendaciones respecto a la presente práctica.

## PRÁCTICA No. 2

**NOMBRE (S):**

**FECHA DEL LABORATORIO:**

### 1. TÍTULO

Conociendo el entorno de Programación de STEP 7.

### 2. OBJETIVOS

#### 2.1. GENERAL

Familiarizar al estudiante con Step 7 Basic.

#### 2.2. ESPECÍFICOS

Conocer cada una de las áreas del entorno gráfico que posee Step 7 Basic.

### 3. LISTA DE MATERIALES Y EQUIPOS

- PC, con Step 7 Basic.

### 4. INDICACIONES GENERALES

Utilice un archivo electrónico para escribir sus respuestas y envíelo por correo a la siguiente dirección: *e-mail\_del\_profesor*.

Este símbolo ♣ indica que debe responder el correspondiente numeral y entregar el resultado en el archivo electrónico.

## 5. PROCEDIMIENTO

5.1. Ejecutar la aplicación TIA Portal V11 cuyo acceso directo se encuentra en el escritorio.

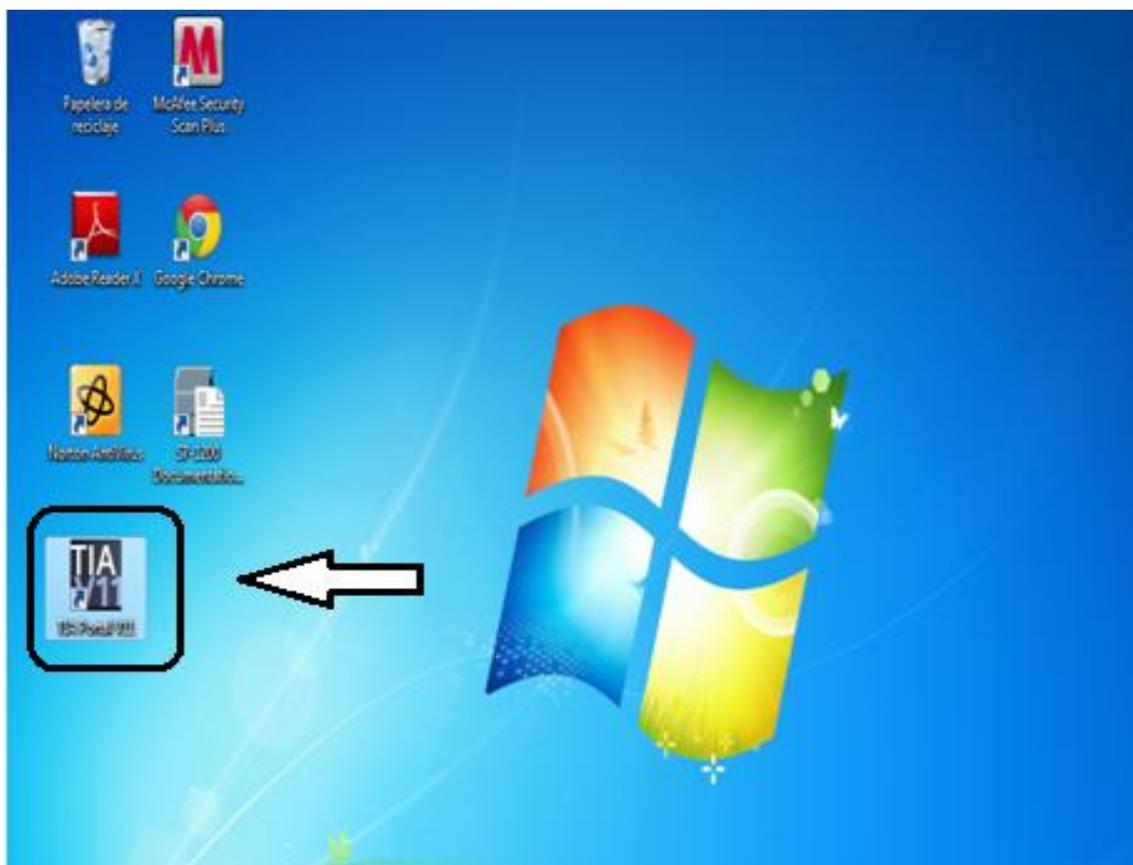


Figura 1. Ejecutar Aplicación.

5.2 “Vistas del Proyecto”. La Vista del Proyecto proporciona una vista funcional de las tareas del proyecto y organiza las herramientas de acuerdo con la tarea que se va a realizar. Es posible determinar fácilmente el procedimiento y la tarea que debe seleccionarse.

Preste atención a la figura y sus áreas:

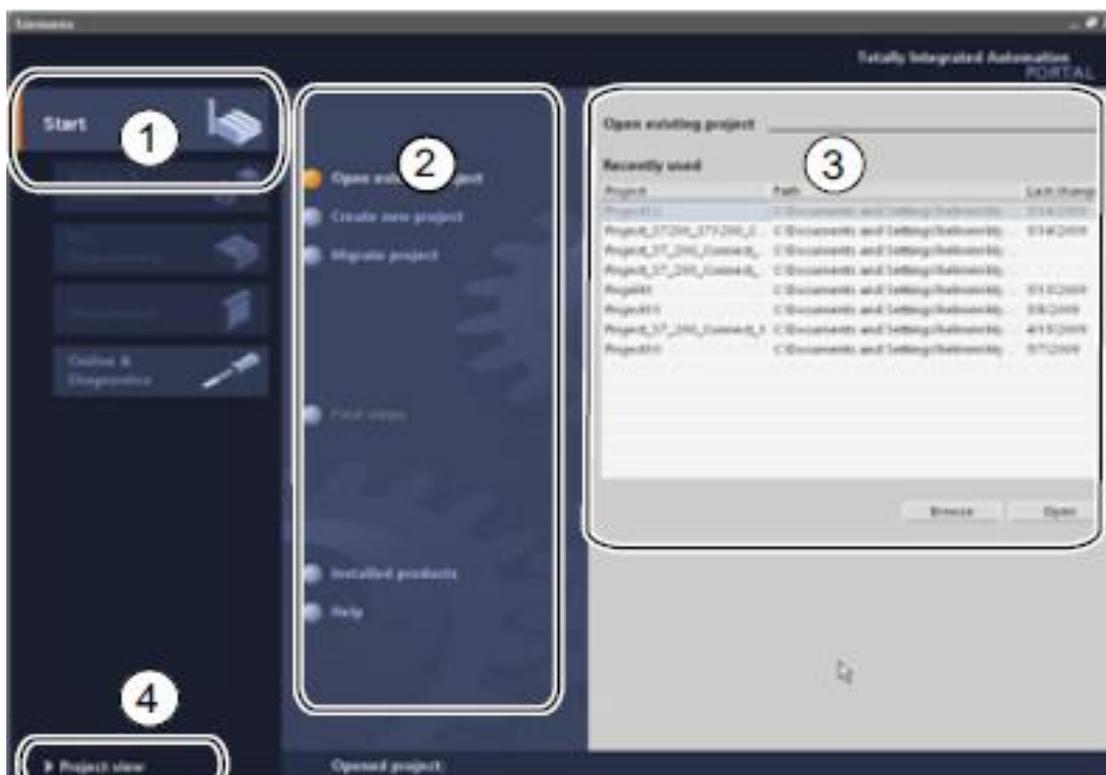


Figura 2. Vistas del Proyecto

- ① Portales para las diferentes tareas.
- ② Tareas del portal seleccionado.
- ③ Panel de selección para la acción seleccionada.
- ④ Cambia a la vista del proyecto.

**5.3** Dar clic en el numeral 4 “Vista del Proyecto”. La vista del proyecto proporciona acceso a todos los componentes del proyecto. Observe cada una de las partes.

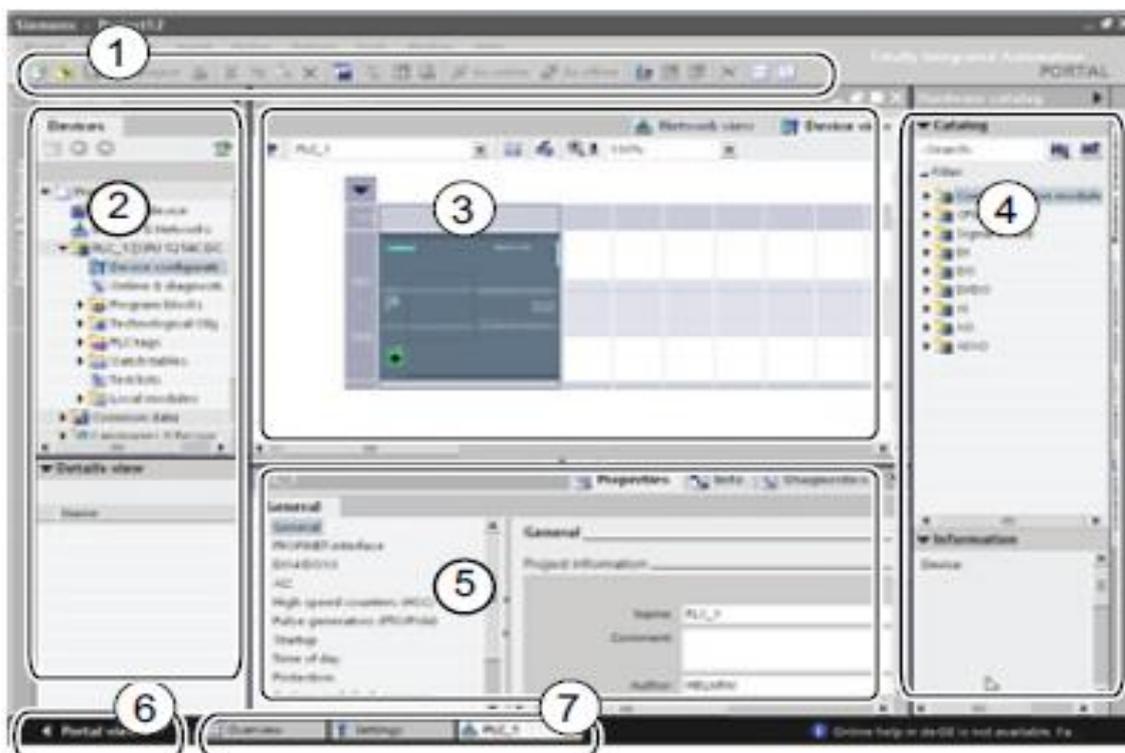
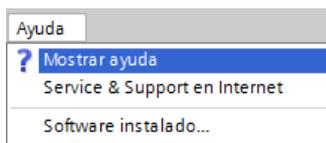


Figura 3. Vista de un proyecto.

- ① Menús y barra de herramientas.
- ② Árbol del proyecto.
- ③ Área de trabajo.
- ④ Task Cards.
- ⑤ Ventana de inspección.
- ⑥ Cambia a la vista del portal.
- ⑦ Barra del editor.

**5.4** “Vista del panel de Ayuda”. STEP 7 Basic provee un completo sistema de información y ayuda en pantalla, en el que se describen todos los productos TIA SIMATIC que se han instalado. El sistema de información se abre en una ventana que no oculta las áreas de trabajo.

En *Menús y barra de herramientas (1)*, haga clic en “Ayuda” y se desplegará la



ventana del “Sistema de Información”.

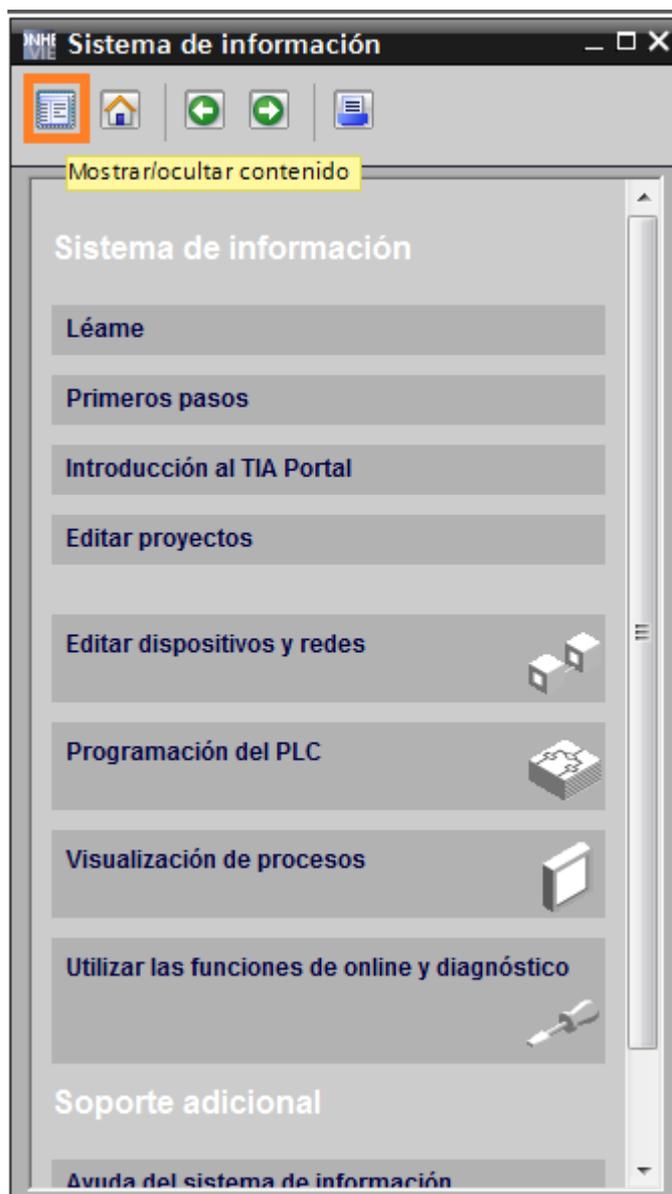


Figura 4. Sistema de Información.

En la barra de herramientas, dar un clic en el botón "Mostrar/ocultar contenido"



del Sistema de Información para ver el contenido de la ventana de ayuda. Al ser una ventana se puede maximizar, minimizar y cambiar su tamaño.

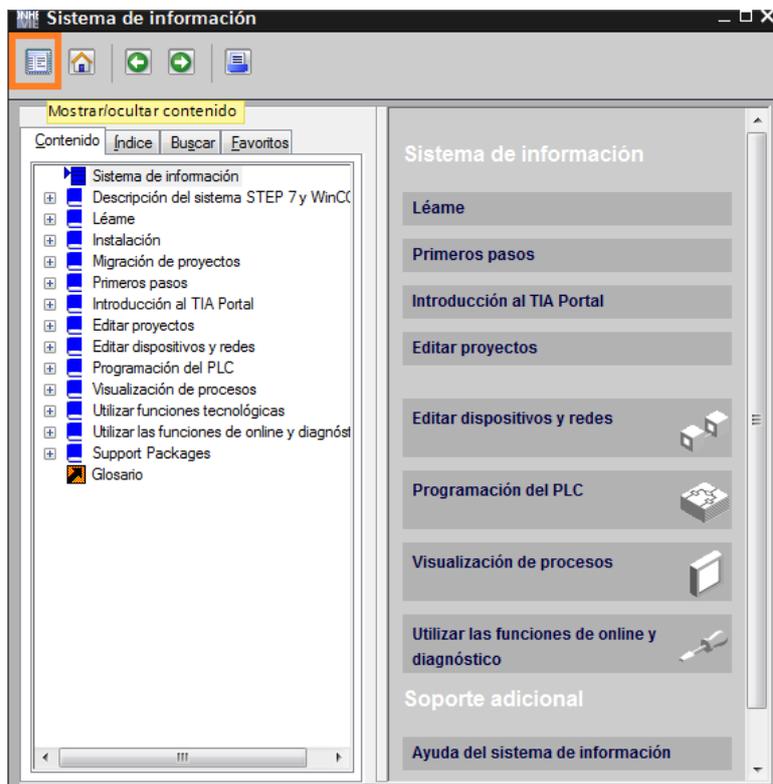


Figura 5. Contenido de la ventana de ayuda.

5.5 ♣ Regrese a la ventana "Vista de un Proyecto". ¿Qué muestra la ventana de inspección?

5.6 ♣ ¿Para qué sirve la barra de editores?

## 6. CONCLUSIONES Y RECOMENDACIONES

6.1. Exponga sus conclusiones respecto a la presente práctica.

6.2. Exponga sus recomendaciones respecto a la presente práctica.

## **PRÁCTICA No. 3**

**NOMBRE (S):**

**FECHA DEL LABORATORIO:**

### **1. TÍTULO**

Aprendiendo el funcionamiento del PLC Siemens S7-1200.

### **2. OBJETIVOS**

#### **2.1. GENERAL**

Verificar el correcto funcionamiento del PLC.

#### **2.2. ESPECÍFICOS**

Aprender las instrucciones para verificar el correcto funcionamiento del PLC.

### **3. LISTA DE MATERIALES Y EQUIPOS**

- Modulo PLC S7-1200.
- Cable gemelo AWG14, un extremo con toma corriente simple y el otro extremo pelado.
- Destornillador plano de 2.4mm.
- PC, con Step 7 Basic.
- Cable Ethernet punto a punto.
- Simulador Siemens de entradas y salidas de 8 posiciones.
- Cámara fotográfica.

#### 4. INDICACIONES GENERALES

Utilice un archivo electrónico para escribir sus respuestas y envíelo por correo a la siguiente dirección: *e-mail\_del\_profesor*.

Este símbolo ♣ indica que debe responder el correspondiente numeral y entregar el resultado en el archivo electrónico.

#### 5. PROCEDIMIENTO

5.1. Introduzca la parte pelada del cable gemelo AWG14 al conector hembra con conexión por tornillo (L1 - N) de corriente alterna (120-240VAC) del PLC.

5.2. Conecte el otro extremo del cable gemelo al toma corrientes.

5.3. Conecte el cable de red punto a punto en los conectores RJ45 del PLC y de la PC.

5.4. En la PC ejecutar la aplicación TIA Portal V11.

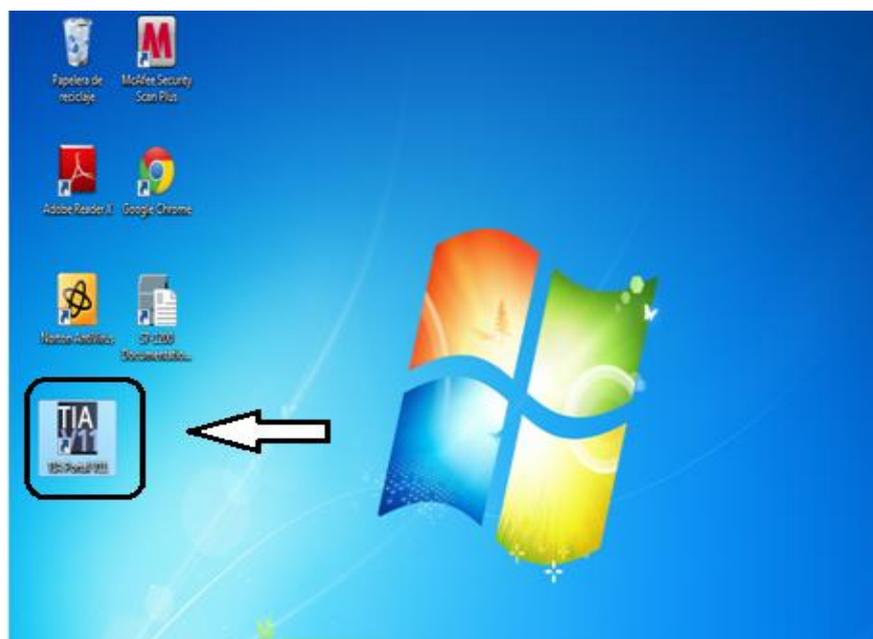
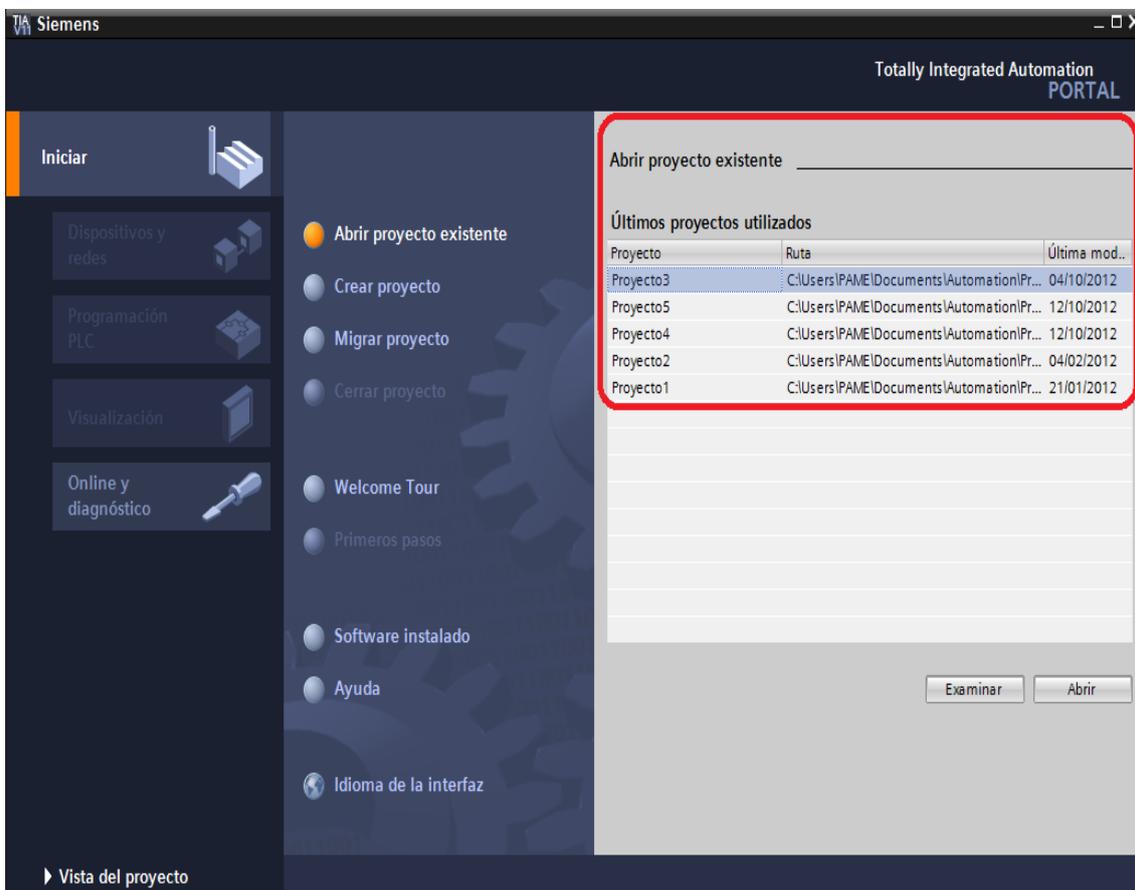


Figura 1. Ejecutar Aplicación TIA Portal.

**5.5.** Dentro de la pantalla de inicio aparece seleccionado por defecto la opción de "Abrir proyecto existente". En el lado derecho de la ventana se encuentra el listado de los proyectos guardados anteriormente.



*Figura 2. Pantalla inicial.*

**5.6.** En la ventana inicial seleccionar la opción "Crear proyecto", en donde se visualizará los campos para identificar el nuevo proyecto (Figura 3). Por ejemplo: quien es el autor, comentario, dirección donde quiere que se guarde el proyecto, etc. A continuación dar clic en "Crear".

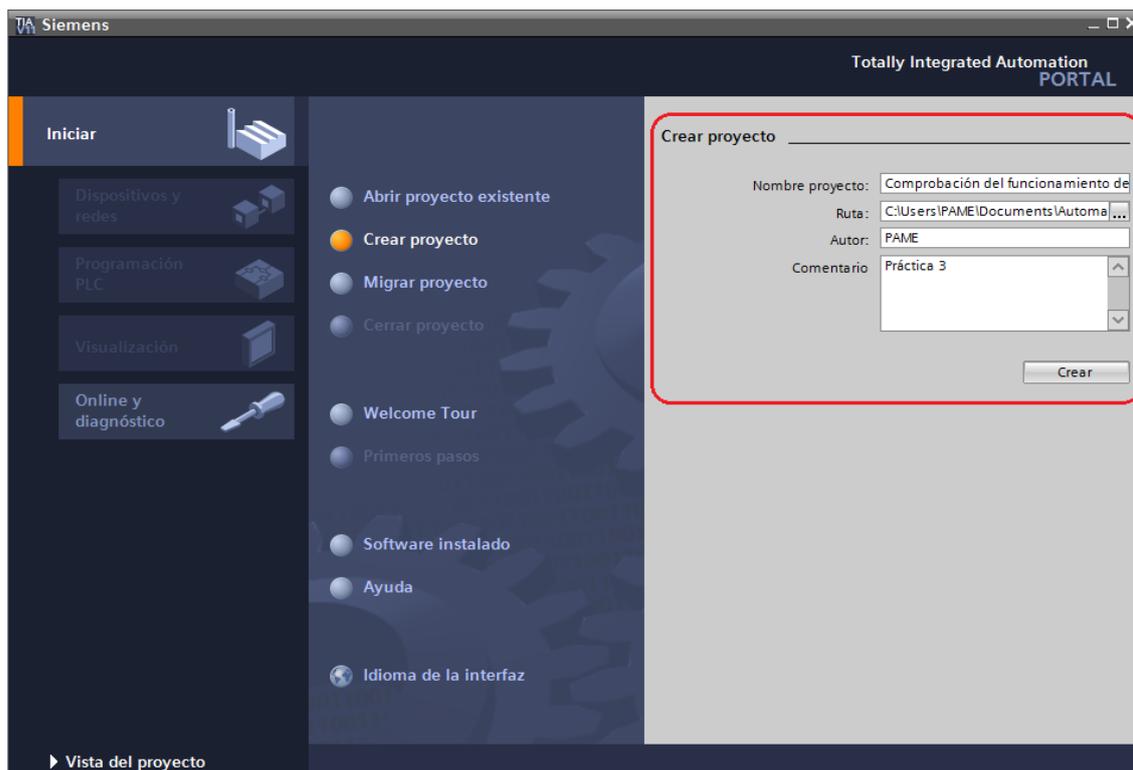


Figura 3. Campos de identificación de proyecto.

**5.7.** Al dar clic en “Crear” aparece la “Vista Portal” en donde por defecto seleccione la opción “Primeros pasos”, en el cual hay tres opciones para poder seguir con el proyecto:

- “Configurar un dispositivo”.
- “Crear un programa PLC”.
- “Configurar una imagen HMI”.

Dar clic en la opción “Configurar dispositivo” para configurar el tipo de PLC con el cual se va a trabajar y cuáles van a ser sus características.

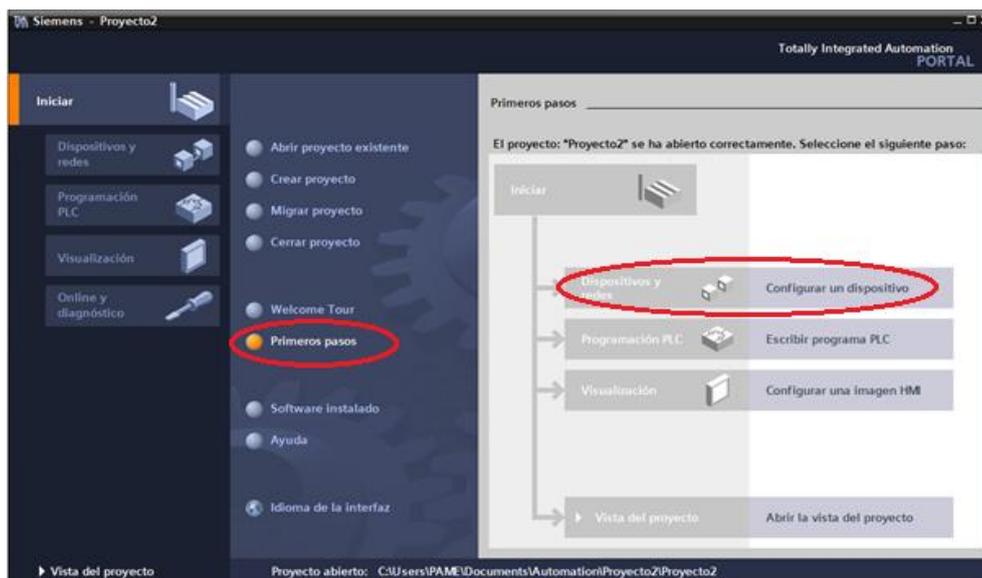


Figura 4. Primeros Pasos.

**5.8.** Después de haber dado clic en “Configuración de dispositivo” se visualiza una nueva ventana en donde está seleccionada por defecto la opción “Mostrar todos los dispositivos” (Figura 5). Ir al ícono donde dice “Agregar dispositivo”, aquí se visualizan todas las CPU con las que puede trabajar el PLC dependiendo de su configuración.

Seleccionar la “CPU 1200 sin especificar” para que al realizar la conexión con el PC, el programa reconozca las características del PLC y si fuera el caso a los módulos adicionales que estén conectados.

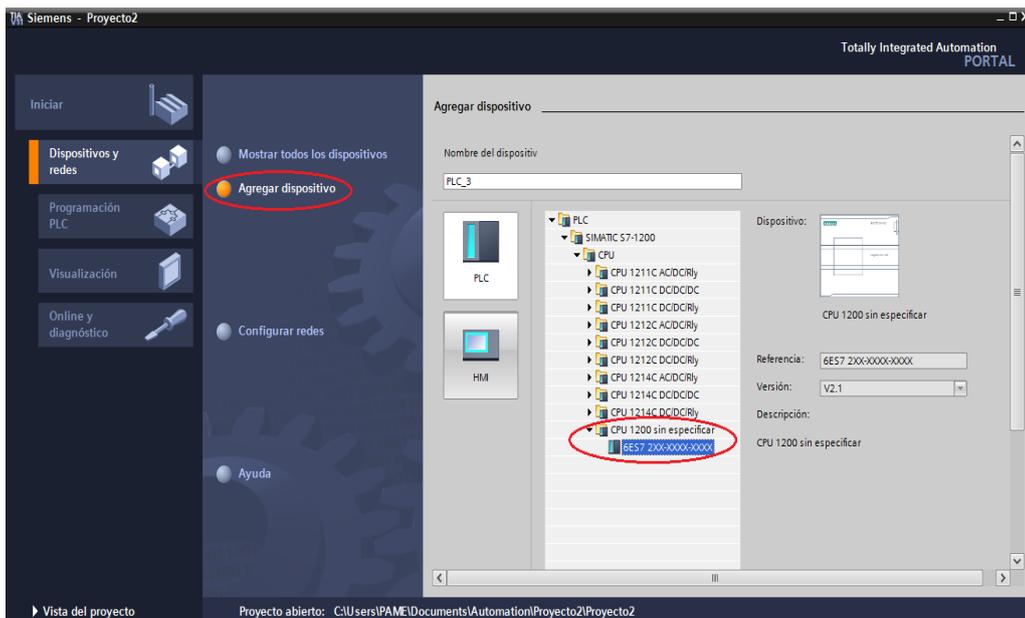


Figura 5. CPU 1200 sin especificar.

### 5.9. Una vez seleccionada la CPU dar clic en “Agregar”

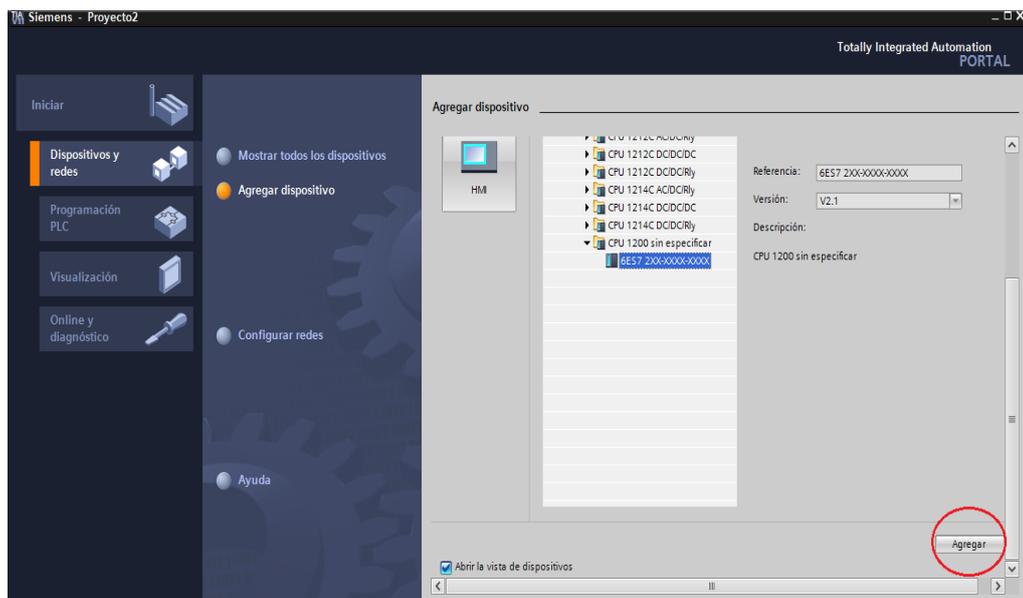


Figura 6. Agregar la CPU seleccionada.

5.10. Una vez agregada la CPU se despliega la “Vista del Proyecto”. Al estar en este entorno se puede ver la información en color naranja a un lado de la “CPU del PLC sin especificar” la cual nos ayudará con la detección del PLC.

Dar clic en “determinar”.

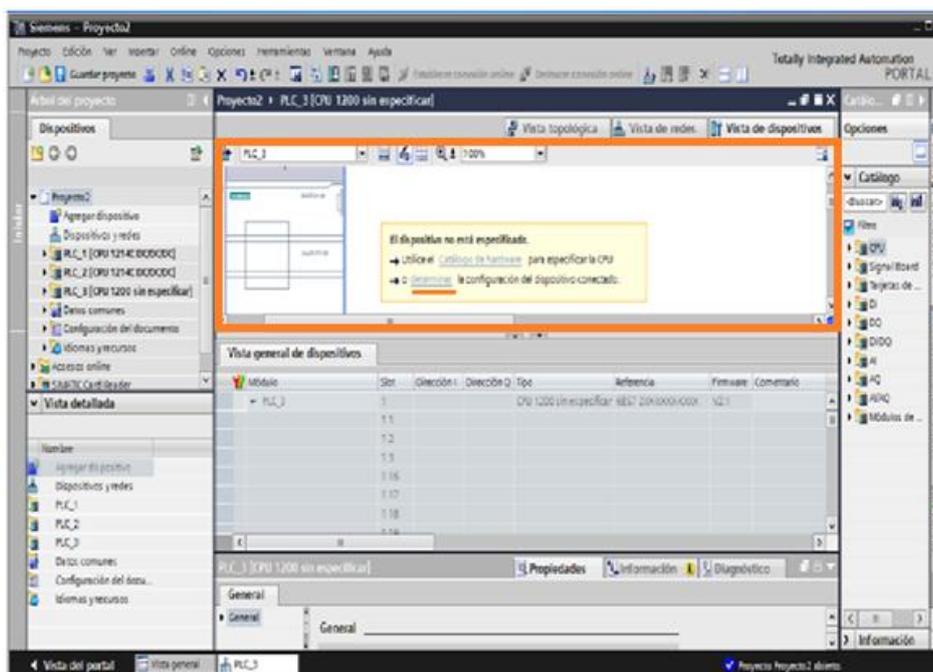


Figura 7. Vista del proyecto.

**5.11.** Después de dar clic en “determine” se despliega la pantalla de la Figura 8. En la cual se va a seleccionar la tarjeta de red de la PC. En la parte superior derecha seleccionar la tarjeta de conexión Ethernet, luego de esto empieza a detectar los dispositivos que estén conectados automáticamente.

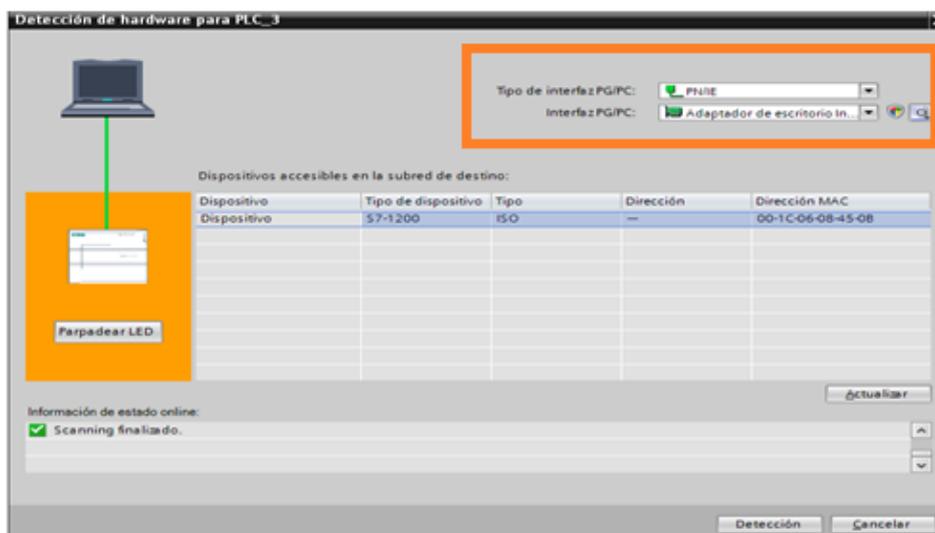


Figura 8. Detección de Hardware.

5.12. Aquí el dispositivo ya está conectado. Seleccionar el dispositivo en la lista (Dispositivo S7-1200). Para verificar su conexión existe la opción “Parpadear LED”.

Al dar clic en esta opción se observa en el PLC el parpadeo de los LEDs de estado, para finalizar dar clic en “detención” en donde se genera la imagen de los dispositivos conectados en Step 7 (Figura 9).

♣ ¿Parpadearon los LEDs? Tomar foto y adjuntarlo en el punto anexos.

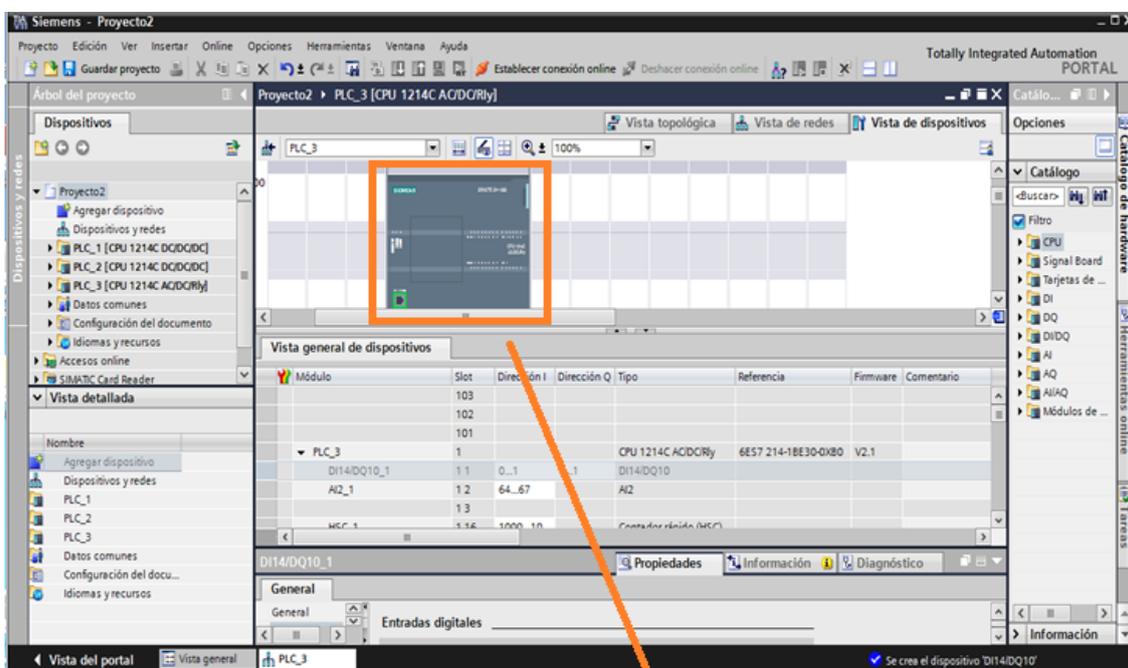


Figura 9. Imagen del dispositivo conectado listo para su programación.

5.13. ♣ ¿Qué tipo de CPU detectó Step 7?

## 6. CONCLUSIONES Y RECOMENDACIONES

6.1. Exponga sus conclusiones respecto a la presente práctica.

6.2. Exponga sus recomendaciones respecto a la presente práctica

## 7. ANEXOS

## PRÁCTICA No. 4

**NOMBRE (S):**

**FECHA DEL LABORATORIO:**

### 1. TÍTULO

Programando al PLC Siemens S7-1200.

### 2. OBJETIVOS

#### 2.1. GENERAL

Realizar un sencillo programa utilizando el Step 7.

#### 2.2. ESPECÍFICOS

Aprender los pasos necesarios para desarrollar un en el PLC S7-1200.

### 3. LISTA DE MATERIALES Y EQUIPOS

- Modulo PLC S7-1200.
- Cable gemelo AWG 14, un extremo con toma corriente simple y el otro extremo pelado.
- Destornillador plano de 2.4mm.
- PC, con Step 7 Basic.
- Cable Ethernet punto a punto.
- Simulador Siemens de entradas y salidas de 8 posiciones.
- Cámara fotográfica.

#### 4. INDICACIONES GENERALES

Utilice un archivo electrónico para escribir sus respuestas y envíelo por correo a la siguiente dirección: *e-mail\_del\_profesor*.

Este símbolo ♣ indica que debe responder el correspondiente numeral y entregar el resultado en el archivo electrónico.

#### 5. PROCEDIMIENTO

**5.1** Introduzca la parte pelada del cable gemelo AWG14 al conector hembra con conexión por tornillo (L1 - N) de corriente alterna (120-240VAC) del PLC.

**5.2** Conecte el otro extremo del cable gemelo al toma corrientes.

**5.3** Conecte el cable de red punto a punto en los conectores RJ45 del PLC y de la PC.

**5.4** En el ambiente del laboratorio es necesario simular dispositivos de entrada que indiquen eventos o cambios de estados al PLC. Para ello no se necesita más que de interruptores que den estados lógicos (ceros y unos) de acuerdo a su posición. Siemens facilita una tarjeta simuladora de entradas de 8 interruptores de palanca, que de acuerdo a su posición, indicarán cada uno de ellos un estado lógico al PLC, de acuerdo a la entrada que esté conectada a cada una de ellas. Esta tarjeta toma el nombre de "Tarjeta Simuladora Siemens de Entradas de 8 Posiciones".

Introduzca la tarjeta Simuladora Siemens de entradas de 8 posiciones en el conector hembra con conexión por tornillo de corriente continua (24VDC) del PLC, de la siguiente manera: viendo de frente a los interruptores de la tarjeta simuladora inserte el pin de la izquierda en el conector de extensión que tiene la etiqueta "L+", luego de esto ajuste los tornillos.

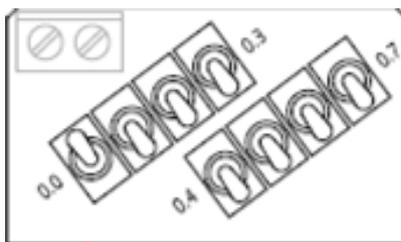


Figura 1. Tarjeta simuladora Siemens de entradas y de 8 posiciones.



Figura 2. Tarjeta simuladora Siemens colocada en el PLC.

**5.5** La programación de un PLC se la realiza mediante STEP 7, el cual ofrece los siguientes lenguajes de programación estándar:

- KOP (esquema de contactos): es un lenguaje de programación gráfico. Su representación se basa en esquemas de circuitos.

Los elementos de un esquema de circuitos, tales como los contactos normalmente cerrados y normalmente abiertos, así como las bobinas, se combinan para formar segmentos.

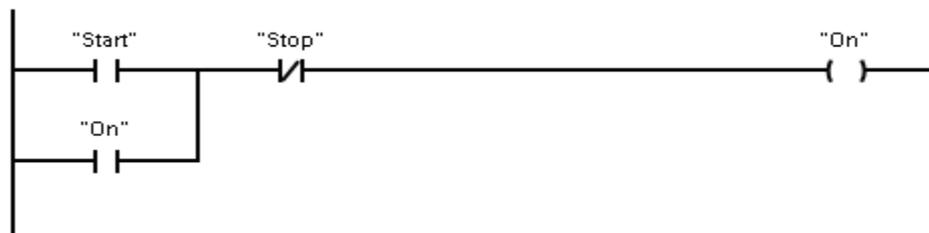


Figura 3. Esquema de contactos KOP

Para crear la lógica de operaciones complejas, es posible insertar ramas para los circuitos paralelos. Las ramas paralelas se abren hacia abajo o se conectan directamente a la barra de alimentación. Las ramas se terminan hacia arriba. KOP ofrece instrucciones con cuadros para numerosas funciones, por ejemplo: matemáticas, temporizadores, contadores y transferencia.

STEP 7 no limita el número de instrucciones (filas y columnas) de un segmento KOP. Todo segmento KOP debe terminar con una bobina o cuadro.

- FUP (diagrama de funciones): es un lenguaje de programación que se basa en los símbolos lógicos gráficos empleados en el álgebra booleana.

Al igual que KOP, FUP es un lenguaje de programación gráfico. La representación de la lógica se basa en los símbolos lógicos gráficos del álgebra booleana.

Para crear la lógica de operaciones complejas, inserte ramas paralelas entre los cuadros.



Figura 4. Esquema de contactos FUP.

Las funciones matemáticas y otras operaciones complejas pueden representarse directamente en combinación con los cuadros lógicos.

STEP 7 no limita el número de instrucciones (filas y columnas) de un segmento FUP.

Al crear un bloque lógico, se debe seleccionar el lenguaje de programación que se va a emplear en dicho bloque. El programa de usuario puede emplear bloques lógicos creados con cualquiera de los lenguajes de programación.

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

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

Tras finalizar el procesamiento de los OBs de ciclo, la CPU vuelve a ejecutarlos inmediatamente. Esta ejecución cíclica es el tipo de procesamiento "normal" que se utiliza para los controladores lógicos programables. En numerosas

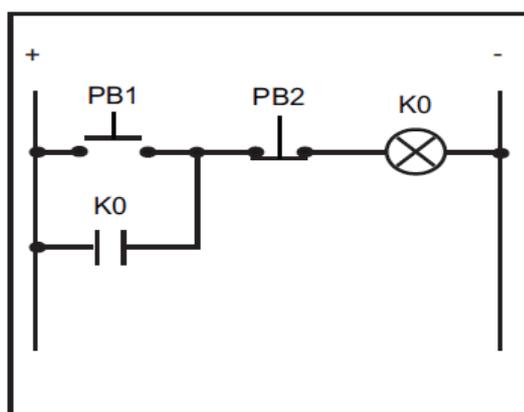
aplicaciones, el programa de usuario entero está contenido en un solo OB de ciclo.

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

**5.6** Se va a realizar un programa en el cual una salida K0 se activa cuando se cierra la entrada de un pulsador PB1, normalmente abierto. Puesto que es un circuito de autorretención utiliza el estado de K0.

K0 permanece activo (ON) tras haberse abierto o soltado PB1. Si se acciona la entrada del pulsador normalmente cerrado PB2 se desactiva K0 (OFF). K0 permanece en OFF hasta que la entrada del pulsador PB1 se cierre o se pulse nuevamente.

Si K0 se utiliza como entrada paralela a PB1, el circuito permanece "enclavado" hasta que se desactiva K0 por el pulsador PB2.



*Figura 5. Circuito de autorretención.*

En la figura 5 se utilizan pulsadores para que se entienda el objetivo del programa. En Step 7 no existen pulsadores, únicamente existen los contactores.

Contactador: es un aparato mecánico de conexión y desconexión eléctrica, accionado por cualquier forma de energía, menos manual, capaz de establecer, soportar e interrumpir corrientes en condiciones normales del circuito, incluso las de sobrecarga. Para realizar el programa se utilizarán los siguientes contactores:

**---| |---: Contacto normalmente abierto:**

La activación de un contacto normalmente abierto depende del estado lógico del operando correspondiente. Si el estado lógico del operando es "1", se cierra el contacto normalmente abierto y el estado lógico de la salida de la instrucción se pone a "1". Si el estado lógico del operando es "0", el contacto normalmente abierto no se activa y el estado lógico de la salida de la instrucción se pone a "0".

**---| / |---: Contacto normalmente cerrado.**

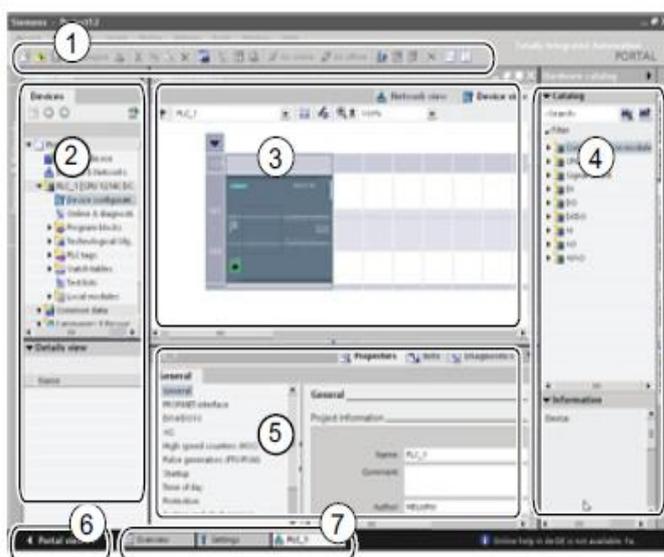
La activación de un contacto normalmente cerrado depende del estado lógico del operando correspondiente. Si el estado lógico del operando es "1", se "abre" el contacto y se interrumpe el flujo de corriente hacia la barra de alimentación derecha. En este caso, la salida de la operación devuelve el estado lógico "0".

**---( )---: Asignación.**

La instrucción "Asignación" permite activar el bit de un operando indicado. Si el resultado lógico en la entrada de la bobina es "1", el operando indicado adopta el estado lógico "1". Si el estado lógico en la entrada de la bobina es "0", el bit del operando indicado se pone a "0".

**5.7** Realice el mismo procedimiento que el desarrollado en la práctica 3.

**5.8** Una vez detectado el PLC ir al "Árbol del Proyecto" en donde se visualiza los dispositivos conectados. Utilice el diálogo "Agregar nuevo bloque" para crear el programa de usuario.



### Vista del proyecto

- ① Menús y barra de herramientas
- ② Árbol del proyecto
- ③ Área de trabajo
- ④ Task Cards
- ⑤ Ventana de inspección
- ⑥ Cambia a la vista del portal
- ⑦ Barra del editor

Figura 6. Vista del proyecto.

**5.9** Para realizar el circuito de autorretención, utilice un contacto normalmente abierto. El contacto normalmente abierto hace que la corriente fluya al activar el interruptor.

Haga clic en el “contacto normalmente abierto” en la barra de "Favoritos" que se encuentra en el “Área de trabajo” para insertarlo en el segmento.



Figura 7. Barra de Favoritos, “contacto normalmente abierto”.

**5.10** El contacto normalmente cerrado hace que la corriente fluya hasta que se active el interruptor. Activando un contacto normalmente cerrado se interrumpe el flujo de corriente.

Haga clic en el “contacto normalmente cerrado” en la barra de "Favoritos" que se encuentra en el “Área de trabajo” para insertarlo en el segmento.

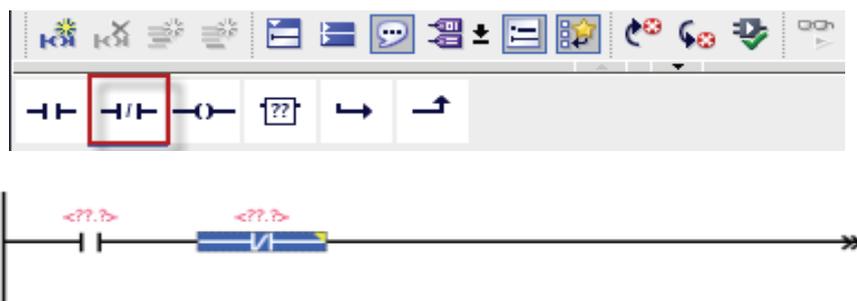


Figura 8. Barra de Favoritos, “contacto normalmente cerrado”.

**5.11** La corriente fluye entre los dos contactos para excitar la bobina. La instrucción “asignación” asigna el resultado lógico del operando indicado. Haga clic en la instrucción “Asignación” en la barra de "Favoritos" que se encuentra en el “Área de trabajo” para insertarlo en el segmento.

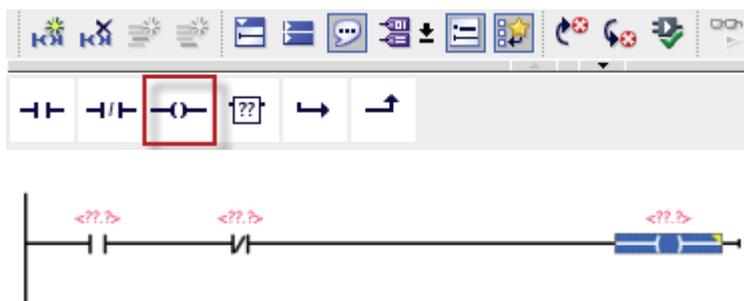


Figura 9. Barra de Favoritos, instrucción “asignación”.

**5.12** Para que la bobina permanezca activada tras desactivar el interruptor "On" (interruptor I0.0), se crea una rama paralela.

1. Seleccione la barra de alimentación del segmento.



Figura 10. Barra de alimentación del segmento.

2. Haga clic en "Abrir rama" en la barra de "Favoritos" para abrir una rama desde la barra de alimentación.



Figura 11. Insertando una rama al segmento.

- 5.13** Al conectar la nueva rama entre los dos contactos se asegura que la corriente pueda fluir hasta la bobina, tras desactivar el primer interruptor (I0.0).



Figura 12. Barra de alimentación del segmento con una nueva rama.

- 5.14** El contacto normalmente cerrado puede interrumpir el circuito y desactivar la bobina.

Haga clic en "contacto normalmente abierto" en la barra de "Favoritos" para insertarlo en la nueva rama. Cierre la rama arrastrando el final hasta el segmento.

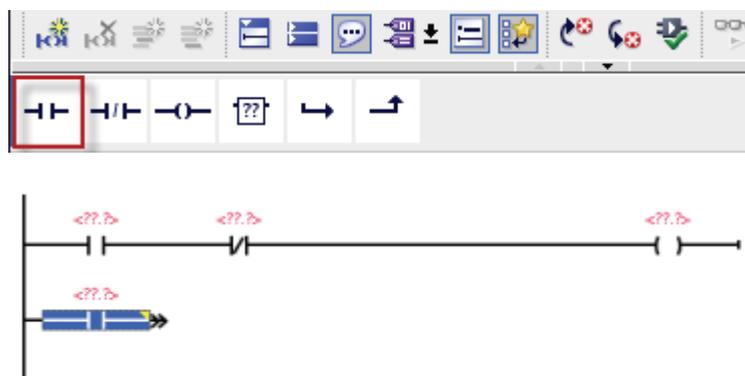


Figura 13. Insertando un "contacto normalmente cerrado" en la nueva rama.

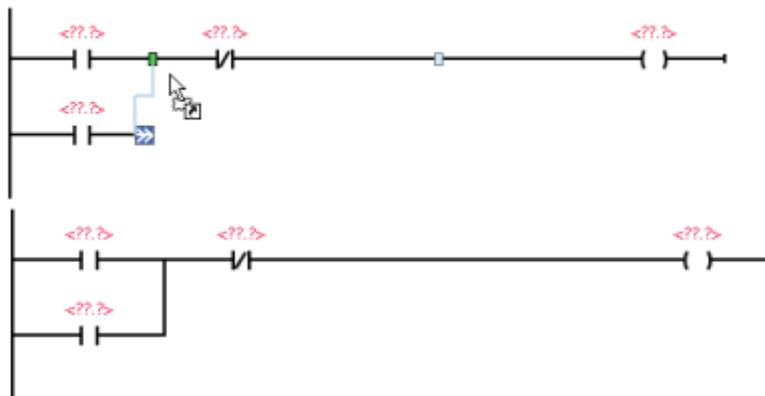


Figura 14. Cerrando la nueva rama.

### 5.15 Guardar Proyecto.

Haga clic en el botón "Guardar proyecto" de la barra de herramientas para guardar los ajustes realizados.

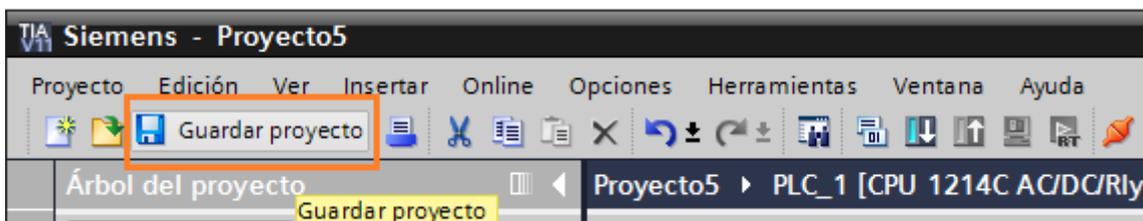


Figura 15. Guardar Proyecto

**5.16** A continuación se crearan las variables para asignar las instrucciones del programa de usuario a las entradas y salidas del programa.

Para introducir las variables y direcciones se utiliza el "Área de Memoria" en condiciones estándar. Todas las E/S digitales y analógicas se actualizan de forma síncrona con el ciclo, utilizando un área de memoria interna denominada memoria imagen de proceso. La memoria imagen de proceso contiene una instantánea de las entradas y salidas físicas de la CPU, de la Signal Board y de los módulos de señales.

La CPU provee las áreas de memoria siguientes para almacenar el programa de usuario, los datos y la configuración:

- La memoria de carga permite almacenar de forma no volátil el programa de usuario, los datos y la configuración. Cuando un proyecto se carga en la CPU, se almacena primero en el área de memoria de carga. Esta área se encuentra bien sea en una Memory Card (si está disponible) o en la CPU. Esta área de memoria no volátil se conserva incluso tras un corte de alimentación. Es posible aumentar la cantidad de memoria de carga disponible para registros con la instalación de una Memory Card.
- La memoria de trabajo ofrece almacenamiento volátil para algunos elementos del proyecto mientras se ejecuta el programa de usuario. La CPU copia algunos elementos del proyecto desde la memoria de carga en la memoria de trabajo. Esta área volátil se pierde si se desconecta la alimentación. La CPU la restablece al retornar la alimentación.
- La memoria remanente permite almacenar de forma no volátil un número limitado de valores de la memoria de trabajo. El área de memoria remanente se utiliza para almacenar los valores de algunas posiciones de memoria durante un corte de alimentación. Si ocurre un corte de alimentación, la CPU dispone de suficiente tiempo de retención para respaldar los valores de un número limitado de posiciones de memoria definidas. Estos valores remanentes se restablecen al retornar la alimentación.

### **Direccionamiento de áreas de memoria**

STEP 7 facilita la programación simbólica. Se crean nombres simbólicos o "variables" para las direcciones de los datos, ya sea como variables PLC asignadas a direcciones de memoria y E/S o como variables locales utilizadas

dentro de un bloque lógico. Para utilizar estas variables en el programa de usuario basta con introducir el nombre de variable para el parámetro de instrucción. Para una mejor comprensión de cómo la CPU estructura y direcciona las áreas de memoria, los siguientes párrafos explican el direccionamiento "absoluto" al que se refieren las variables PLC. La CPU ofrece varias opciones para almacenar datos durante la ejecución del programa de usuario:

- Memoria global: La CPU ofrece distintas áreas de memoria, incluyendo entradas (I), salidas (Q) y marcas (M). Todos los bloques lógicos pueden acceder sin restricción alguna a esta memoria.
- Bloque de datos (DB): Es posible incluir DBs en el programa de usuario para almacenar los datos de los bloques lógicos. Los datos almacenados se conservan cuando finaliza la ejecución del bloque lógico asociado. Un DB "global" almacena datos que pueden ser utilizados por todos los bloques lógicos, mientras que un DB de instancia almacena datos para un bloque de función (FB) específico y está estructurado según los parámetros del FB.
- Memoria temporal: Cada vez que se llama un bloque lógico, el sistema operativo de la CPU asigna la memoria temporal o local (L) que debe utilizarse durante la ejecución del bloque. Cuando finaliza la ejecución del bloque lógico, la CPU reasigna la memoria local para la ejecución de otros bloques lógicos.

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

Las referencias a las áreas de memoria de entrada (I) o salida (Q), como I0.3 o Q1.7, acceden a la memoria imagen del proceso. Para acceder inmediatamente a la entrada o salida física es preciso añadir ":P" a la dirección (p. ej. I0.3:P, Q1.7:P

o "Stop:P"). El forzado permanente sólo escribe un valor en una entrada física (Ix.y:P) o en una salida física (Qx.y:P). Para forzar permanentemente una entrada o salida, agregue una ":P" a la variable PLC o dirección.

Área de memoria	Descripción	Forzado permanente	Remanente
I Memoria imagen de proceso de las entradas	Se copia de las entradas físicas al inicio del ciclo	No	No
I_:P <sup>1</sup> (entrada física)	Lectura inmediata de las entradas físicas de la CPU, SB y SM	Sí	No
Q Memoria imagen de proceso de las salidas	Se copia en las salidas físicas al inicio del ciclo	No	No
Q_:P <sup>1</sup> (salida física)	Escritura inmediata en las salidas físicas de la CPU, SB y SM	Sí	No
M Área de marcas	Control y memoria de datos	No	Sí (opcional)
L Memoria temporal	Datos locales temporales de un bloque	No	No
DB Bloque de datos	Memoria de datos y de parámetros de FBs	No	Sí (opcional)

<sup>1</sup> Para acceder inmediatamente (o forzar permanentemente) las entradas o salidas físicas es preciso añadir ":P" a la dirección o variable (p. ej. I0.3:P, Q1.7:P o "Stop:P").

Tabla 1. Área de memoria

Toda posición de memoria diferente tiene una dirección unívoca. El programa de usuario utiliza estas direcciones para acceder a la información de la posición de memoria. La figura 16 muestra cómo acceder a un bit (lo que también se conoce como direccionamiento "byte.bit"). En este ejemplo, el área de memoria y la dirección del byte (M = marca y 3 = byte 3) van seguidas de un punto (".") que separa la dirección del bit (bit 4).

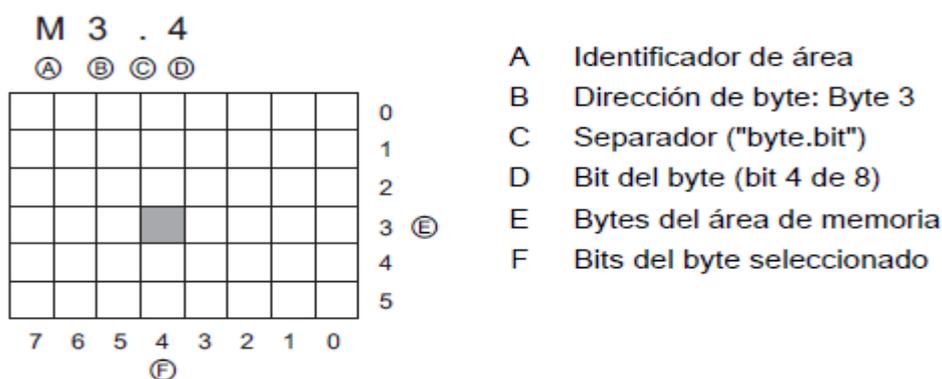


Figura 16. Direccionamiento de memoria.

**Tipo de Datos:** Los tipos de datos se utilizan para determinar el tamaño de un elemento de datos y cómo deben interpretarse los datos. Todo parámetro de instrucción soporta como mínimo un tipo de datos. Algunos parámetros soportan varios tipos de datos. Sitúe el cursor sobre el campo de parámetro de una instrucción para ver qué tipos de datos soporta el parámetro en cuestión.

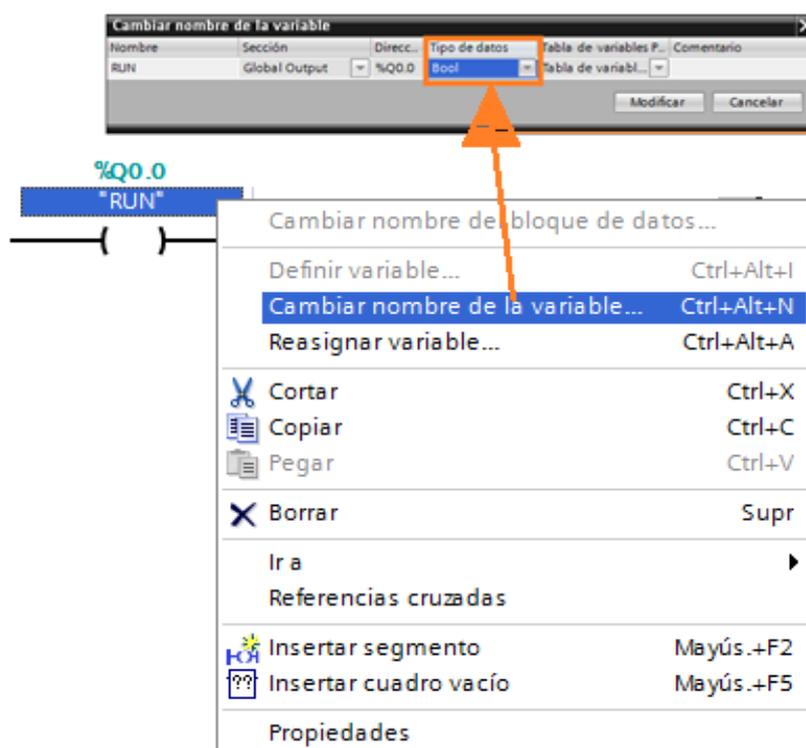


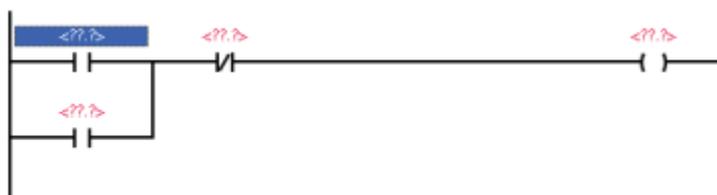
Figura 17. Tipo de datos que soporta una instrucción.

Tipos de datos	Descripción
Tipos de datos de bits y secuencias de bits	<ul style="list-style-type: none"> <li>• Bool es un valor de bit o booleano.</li> <li>• Byte es un valor de byte (8 bits).</li> <li>• Word es un valor de 16 bits.</li> <li>• DWord es un valor de palabra doble (32 bits).</li> </ul>
Tipos de datos enteros	<ul style="list-style-type: none"> <li>• USInt (entero sin signo de 8 bits) y Sint (entero con signo de 8 bits) son enteros "cortos" (8 bits o 1 byte de memoria) que pueden tener o no signo.</li> <li>• UInt (entero sin signo de 16 bits) e Int (entero con signo de 16 bits) son enteros (16 bits o 1 palabra de memoria) que pueden tener o no signo.</li> <li>• UDInt (entero de 32 bits sin signo) y DInt (entero de 32 bits con signo) son enteros dobles (32 bits o 1 palabra doble de memoria) que pueden tener o no signo.</li> </ul>
Tipos de datos de números reales	<ul style="list-style-type: none"> <li>• Real es un valor de número real de 32 bits o en coma flotante.</li> <li>• LReal es un valor de número real de 64 bits o en coma flotante.</li> </ul>
Tipos de datos de fecha y hora	<ul style="list-style-type: none"> <li>• Date es un valor de fecha de 16 bits (similar a un valor UInt) que contiene el número de días desde el 1 de enero de 1990. El valor de fecha máximo es 65535 (16#FFFF), que corresponde al 6 de junio de 2169. Todos los posibles valores de Date son válidos.</li> <li>• DTL (Date and Time Long) es una estructura de 12 bytes que almacena información de fecha y hora en una estructura predefinida. <ul style="list-style-type: none"> <li>- Año (UInt): de 1970 a 2554</li> <li>- Mes (USInt): de 1 a 12</li> <li>- Día de la semana (USInt): de 1 (domingo) a 7 (sábado)</li> <li>- Horas (USInt): de 0 a 23</li> <li>- Minutos (USInt): de 0 a 59</li> <li>- Segundos (USInt): de 0 a 59</li> <li>- Nanosegundos (UDInt): de 0 a 999999999</li> </ul> </li> <li>• Time es un valor de tiempo CEI de 32 bits (parecido al valor Dint) que almacena el número de milisegundos (de 0 a 24 días, 20 horas, 31 minutos, 23 segundos y 647 ms). Todos los posibles valores de Time son válidos. Los valores de Time se pueden usar para cálculos, y se pueden obtener tiempos negativos.</li> <li>• TOD (Time of Day) es un valor de hora de 32 bits (parecido al valor Dint) que contiene el número de milisegundos desde medianoche (de 0 a 86399999).</li> </ul>
Tipos de datos de caracteres y cadenas	<ul style="list-style-type: none"> <li>• Char es un carácter simple de 8 bits.</li> <li>• String es una cadena de longitud variable de hasta 254 caracteres.</li> </ul>

Tabla 2. Tipo de datos.

**5.17** El siguiente paso consiste en asignar los contactos y bobinas a las entradas y salidas de la CPU. Para estas direcciones se crean variables.

1. Seleccione el primer contacto y haga doble clic en el operando ("`<??.?>`").



2. Introduzca la dirección "I0.0" para crear una variable predeterminada para esta entrada.



3. Introduzca la dirección "I0.1" para el contacto normalmente cerrado. Introduzca una dirección de una salida ("Q0.0") para la bobina.



**5.18** El nombre predeterminado de la variable creado por STEP 7 Basic se puede cambiar fácilmente.

Haga clic con el botón derecho del ratón en la instrucción (contacto o bobina) y elija el comando "Cambiar nombre de la variable" del menú contextual.

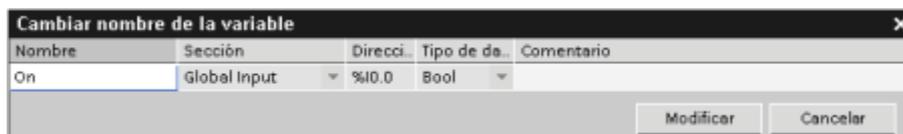
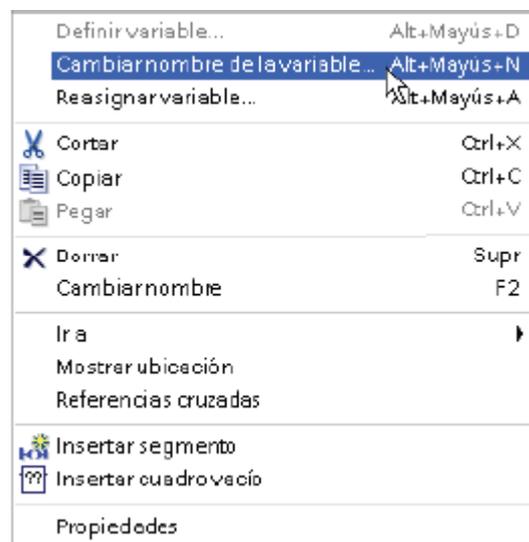


Figura 18. Cambiando el nombre a una variable

### 5.19 Cambie de nombre a las siguientes variables:

- Cambie "Tag\_1" (I0.0) a "On".
  - Cambie "Tag\_2" (I0.1) a "Off".
  - Cambie "Tag\_3" (Q0.0) a "Run".
- 
- Seleccione el “contacto normalmente cerrado” que esta insertado en la segunda rama y de el nombre de “Run” a la variable.

El circuito de autorretención está terminado.



Figura 19. Circuito de autorretención terminado.

♣ ¿Qué tipo de datos se utiliza en el programa?

### 5.20 Guardar proyecto

Haga clic en el botón "Guardar proyecto" de la barra de herramientas para guardar los ajustes realizados.

### 5.21 Cargar el programa de usuario en la CPU.



Abra el editor de programas y haga clic en el botón "Cargar en dispositivo".

**5.22** Tras establecer la conexión con la CPU, STEP 7 Basic muestra el diálogo "Cargar vista preliminar". Haga clic en "Cargar" para cargar el programa de usuario en la CPU. Antes de hacer clic en "Finalizar", seleccione "Arrancar todos" para asegurar que la CPU pase a estado operativo RUN.

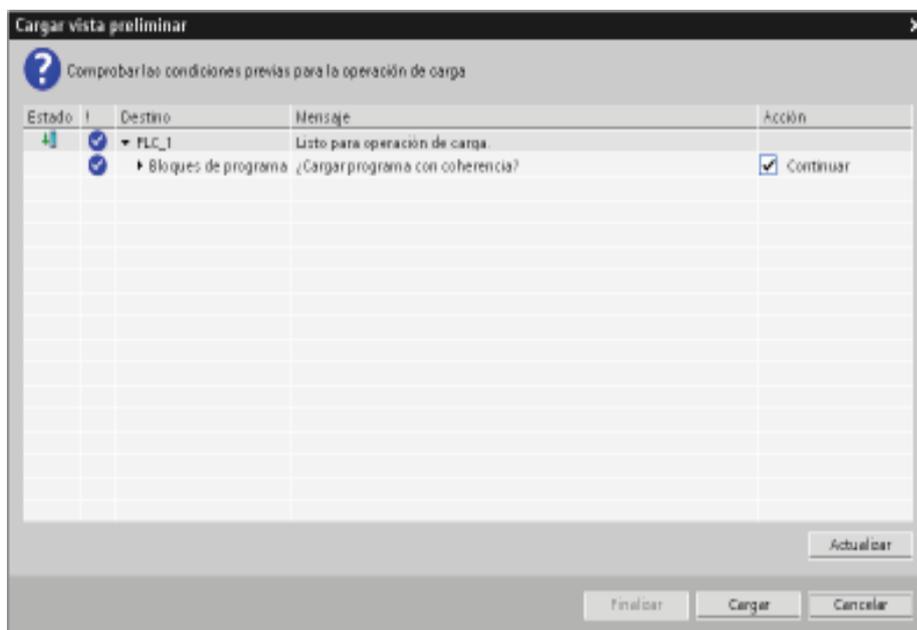


Figura 21. Cargando programa a la CPU.

**5.23** Prueba de operación del programa de usuario.

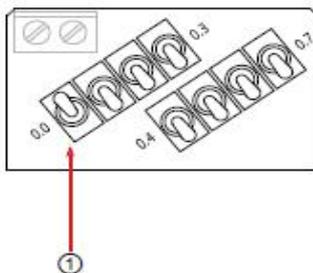
1. Active el interruptor

"On" (I0.0).

Los LEDs de estado de

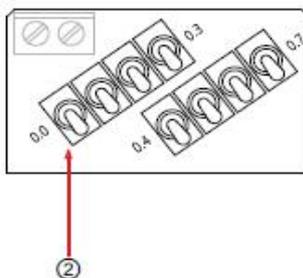
"On" (I0.0) y "Run"

(Q0.0) se encienden.



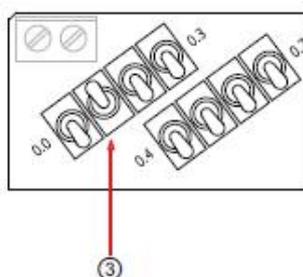
2. Desactive el interruptor "On" (I0.0).

El LED de estado de "On" (I0.0) se apaga, pero el de "Run" (Q0.0) permanece encendido.



3. Active el interruptor "Off" (I0.1).

El LED de estado de "Off" (I0.1) se enciende y el de "Run" (Q0.0) se apaga.



5.24 ♣ Tomar fotografías de la práctica para comprobar su funcionamiento.

## 6. CONCLUSIONES Y RECOMENDACIONES

6.1. Exponga sus conclusiones respecto a la presente práctica.

6.2. Exponga sus recomendaciones respecto a la presente práctica

## 7. ANEXOS

## PRÁCTICA No. 5

**NOMBRE (S):**

**FECHA DEL LABORATORIO:**

### 1. TÍTULO

Temporizadores.

### 2. OBJETIVOS

#### 2.1. GENERAL

Realizar un programa en Step 7 con temporizadores.

#### 2.2. ESPECÍFICOS

Aprender a programar al PLC Siemens S7-1200 con temporizadores.

### 3. LISTA DE MATERIALES Y EQUIPOS

- Modulo PLC S7-1200.
- Cable gemelo AWG 14, un extremo con toma corriente simple y el otro extremo pelado.
- Destornillador plano de 2.4mm.
- PC, con Step 7 Basic.
- Cable Ethernet punto a punto.
- Simulador Siemens de entradas y salidas de 8 posiciones.
- Cámara fotográfica.

#### 4. INDICACIONES GENERALES

Utilice un archivo electrónico para escribir sus respuestas y envíelo por correo a la siguiente dirección: *e-mail\_del\_profesor*.

Este símbolo ♣ indica que debe responder el correspondiente numeral y entregar el resultado en el archivo electrónico.

#### 5. PROCEDIMIENTO

5.1. Realice el mismo procedimiento que el de la práctica 3 y 4.

5.2. Realice un programa en el cual se utilice un circuito de autorretención y se usen temporizadores para obtener el siguiente resultado al simular la práctica:

Al activar el interruptor "ON" (I0.0) al cabo de 5 segundos se activará el LED de estado de la salida Q0.1. Desactive la entrada I0.0 y active la entrada I0.1.

Al cabo de 5 segundos se desactivará el LED de estado de la salida Q0.1.

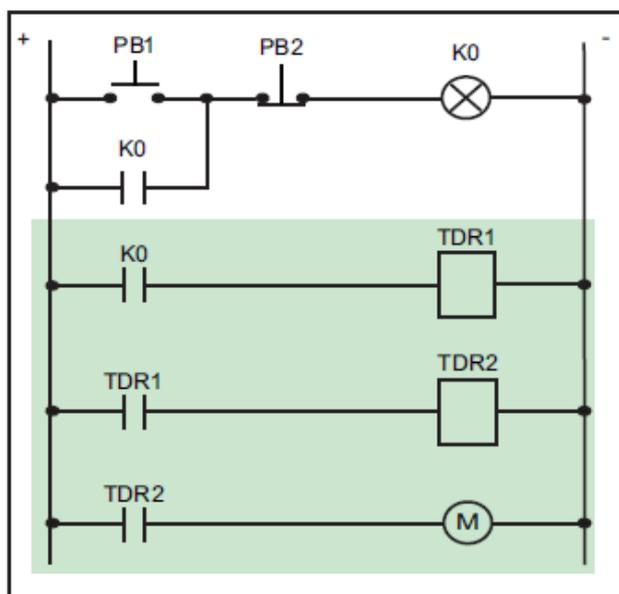
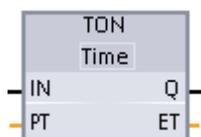


Figura 1. Circuito del programa.

En la figura 1 se utilizan pulsadores para que se entienda el objetivo del programa. En Step 7 no existen pulsadores, únicamente existen los contactores.

**5.3.** Después de haber realizado el mismo procedimiento que el de las prácticas 3 y 4 se a continuación se insertará un temporizador al programa. Los temporizadores utilizan un contador binario encargado de contar los pulsos suministrados por algún circuito oscilador, con una base de tiempo estable y conocida. En Step 7 se utiliza la instrucción TON y TOF para simular un temporizador ya sea de retardo al conectar o un temporizador de retardo al desconectar.

### TON: Retardo al conectar.



La instrucción TON se utiliza para activar una salida (Q) tras un retardo preseleccionado. El temporizador utiliza una estructura almacenada en un bloque de datos para conservar sus datos. Este bloque de datos se asigna al colocar la instrucción de temporización en el editor.

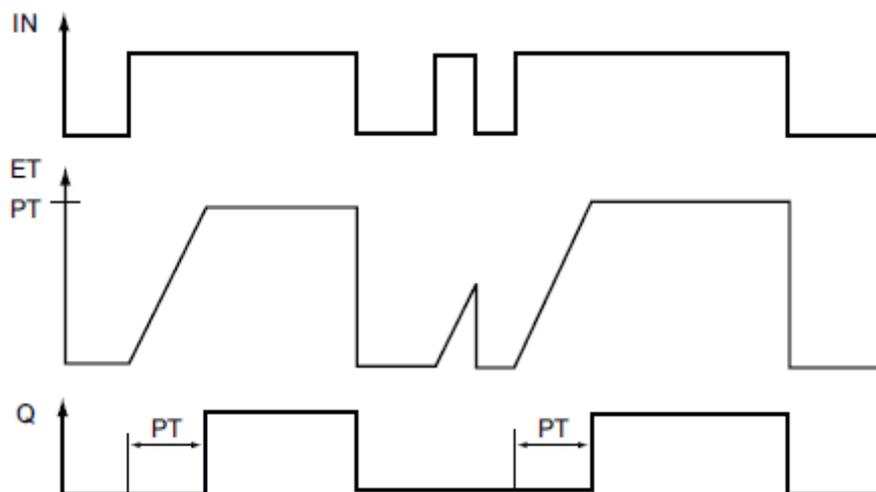


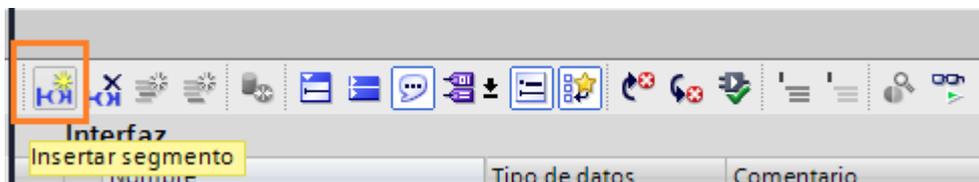
Figura 2. Diagrama de tiempo de un "temporizador de retardo al conectar".

Cuando se activa el temporizador ( $IN = 1$ ), el temporizador de retardo a la desconexión espera un tiempo preseleccionado (PT) hasta activar su salida

(Q = 1). La salida permanecerá activada (Q = 1) mientras la entrada esté activada (IN = 1). El temporizador de retardo a la conexión utiliza un bloque de datos (DB) para conservar sus datos. Este bloque de datos se asigna al insertar la instrucción TON en el segmento.

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

**5.4.** Inserte un nuevo segmento al bloque del programa. Dar clic en la barra de herramientas que se encuentra en el área de trabajo.



*Figura 3. Insertando un nuevo segmento.*

**5.5.** En el segundo segmento del programa de usuario utilice la instrucción TON que se activará 5 segundos después de haberse activado la salida "Run" del circuito de autorretención.

Haga clic en el contacto normalmente abierto en la barra de "Favoritos" para insertar la instrucción.

Para la dirección de la instrucción "contacto normalmente abierto" seleccione la variable "Run". Expanda la carpeta "Temporizadores" en la Task Card "Instrucciones" y arrastre el temporizador TON hasta el segmento.

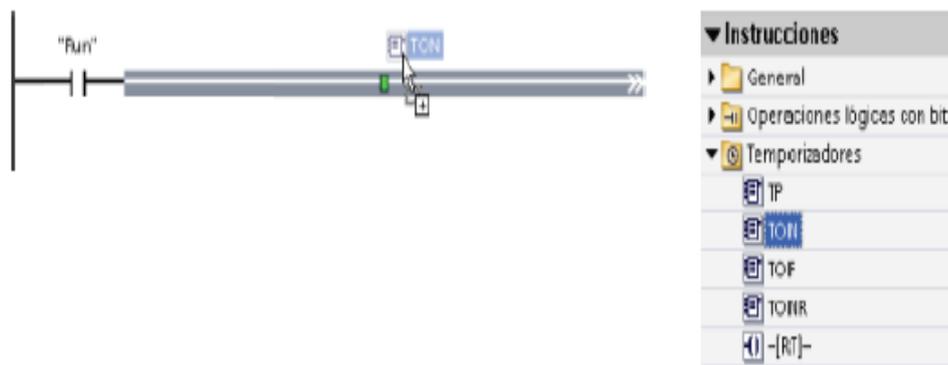


Figura 4. Insertando el temporizador en el segmento 2.

**5.6.** Al insertar la instrucción TON en el segmento se crea automáticamente un bloque de datos (DB) de instancia única para almacenar los datos del temporizador.

- Haga clic en "Aceptar" para crear el DB.

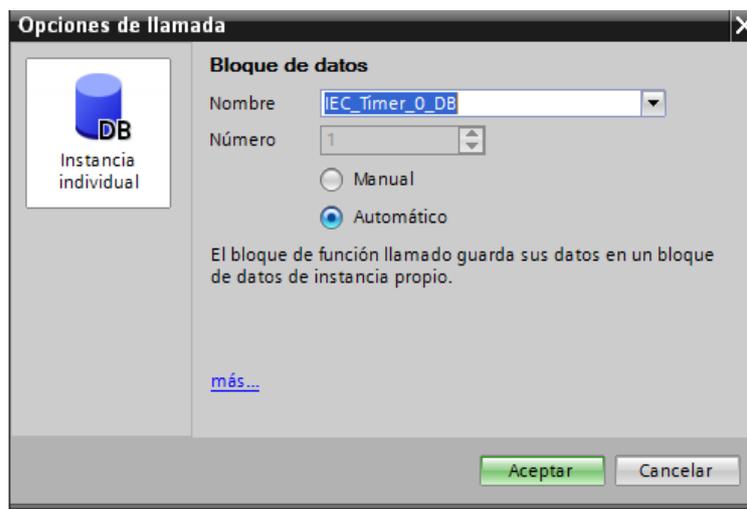
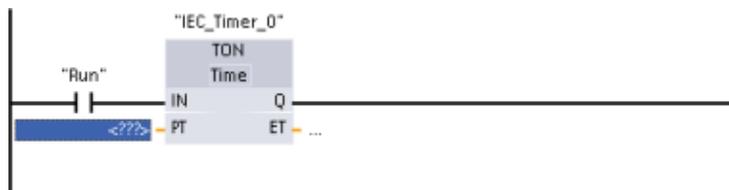


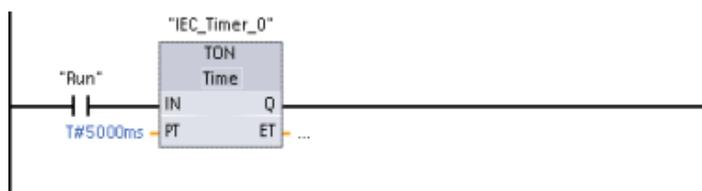
Figura 5. Bloque de un temporizador con instrucción TON

## 5.7. Cree ahora un retardo de 5 segundos.

1. Haga doble clic en el parámetro del tiempo preseleccionado (PT).



2. Introduzca el valor de constante "5000" (para 5000ms, es decir, segundos).



También es posible introducir

"5s", es decir, 5 segundos.

("5h" introduce 5 horas y

"5m" introduce 5 minutos.)

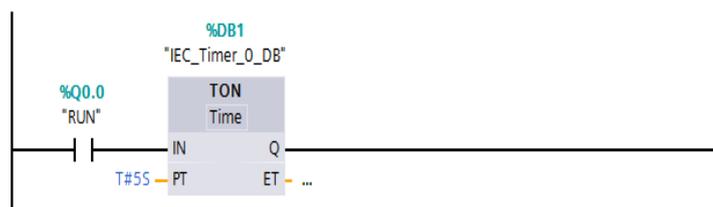


Figura 6. Creando un retardo de 5 segundos.

5.8. Inserte una bobina que se active al cabo de 5 segundos (el valor de preselección de la instrucción TON).

En este programa introduzca "M0.0" para la dirección, ésta almacena el valor en el área de marcas (M). Cambie el nombre de la variable a "Delay\_5s".

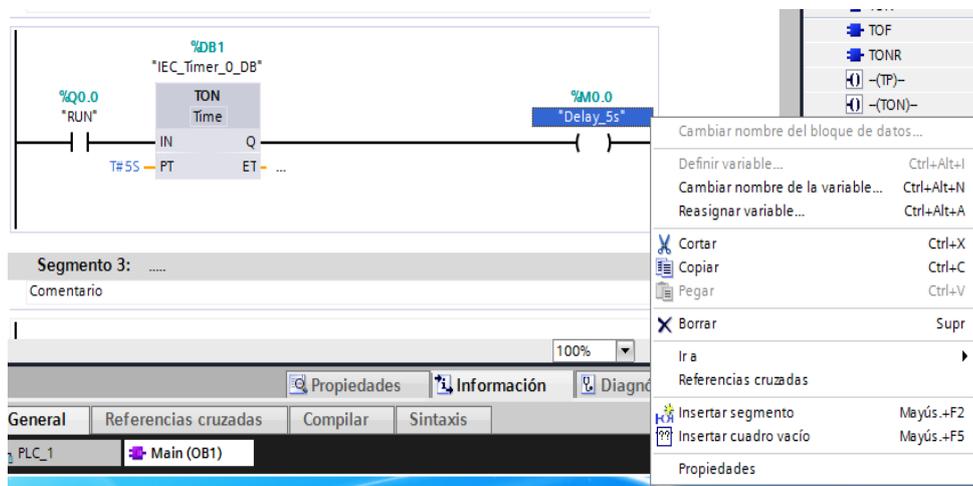


Figura 7. Insertando una bobina.

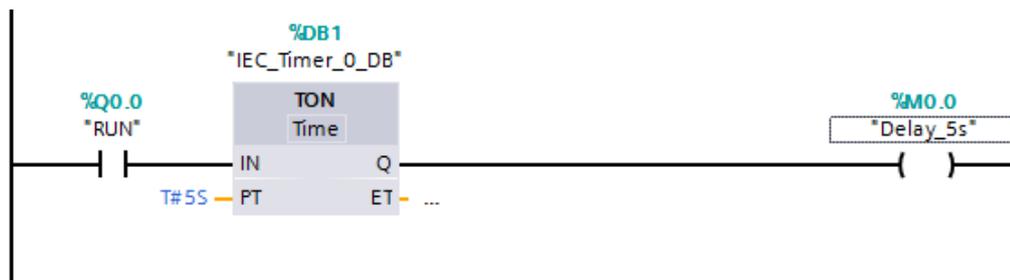


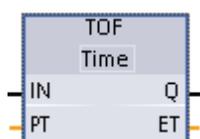
Figura 8. Segmento 2 terminado.

## 5.9. Guardar proyecto

Haga clic en el botón "Guardar proyecto" de la barra de herramientas para guardar los ajustes realizados.

**5.10.** En un tercer segmento se agregará un “temporizador de retardo a la desconexión” (TOF).

### TOF: Retardo al desconectar.



La instrucción TOF se utiliza para mantener activada una salida (Q) durante un tiempo preseleccionado tras haberse desactivado la entrada (IN). Cuando es habilitada por un flanco positivo en IN, la instrucción TOF activa Q. El temporizador TOF arranca en el flanco negativo de IN. Una vez transcurrido el tiempo preseleccionado (PT), la instrucción TOF desactiva Q.

Cuando se activa el temporizador (IN = 1), se activa la salida (Q = 1) del temporizador de retardo a la desconexión. Una vez transcurrido un tiempo preseleccionado (PT), se desactiva la salida del temporizador de retardo a la desconexión.

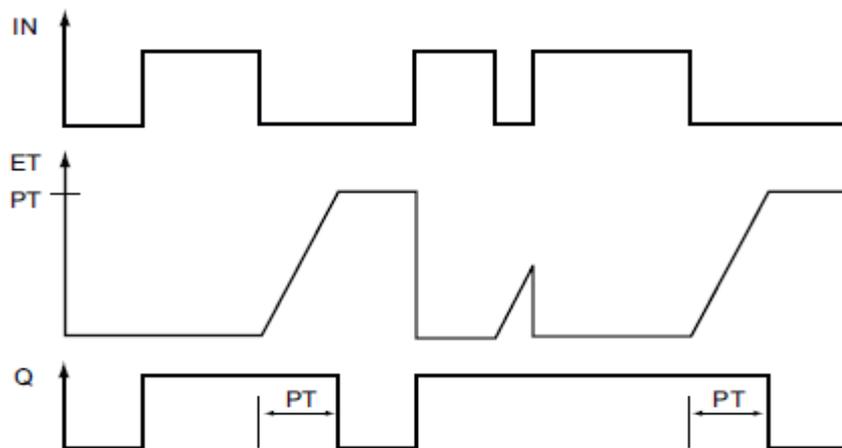


Figura 9. Diagrama de tiempo de un temporizador con retardo al desconectar.

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

También es posible introducir una dirección de memoria Time (valor de 4 bytes con signo) para almacenar el valor de preselección. Esto permite al programa de usuario cambiar el valor de preselección según las condiciones operativas, si es necesario. El tiempo transcurrido (ET) también se puede almacenar en una dirección de memoria Time (valor de 4 bytes con signo) a la que pueden acceder otros elementos del programa de usuario.

**5.11.** Utilizar un “temporizador de retardo al desconectar” para activar una salida durante 5 segundos en un tercer segmento.

El tercer segmento del programa de usuario utiliza una instrucción TOF que se activa y permanece activada durante 5 segundos tras activarse el temporizador TON. En el caso del “temporizador de retardo al desconectar” utilizará como nombre de variable "Delay\_5s" para activarlo.

- A continuación inserte un nuevo segmento al bloque del programa. Dar clic en la barra de herramientas que se encuentra en el área de trabajo.



Figura 10. Insertando nuevo segmento.

- 5.12.** Inserte en el segmento 3 un “contacto normalmente abierto”. Asigne el siguiente nombre a la variable: "Delay\_5sec".

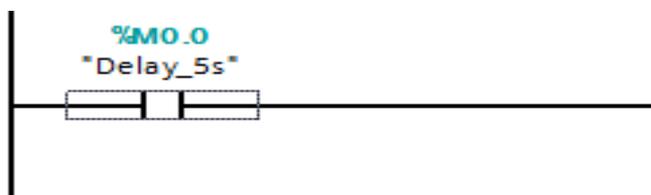


Figura 11. Insertando un tercer segmento y creando la variable “Delay\_5s”.

- 5.13.** El bit "Delay\_5s" se activa 5 segundos tras haberse activado la salida "Run".

Para insertar el temporizador TOF, haga doble clic en el temporizador TOF en la Task Card "Instrucciones". Al igual que en el caso del temporizador TON, STEP 7 Basic crea un DB para los datos del temporizador.

Haga clic en "Aceptar" para crear el temporizador e insertar la instrucción TOF.

Para el valor de preselección de la instrucción TOF, introduzca 5000 (para 5000 ms, es decir, 5 segundos) en el parámetro PT.

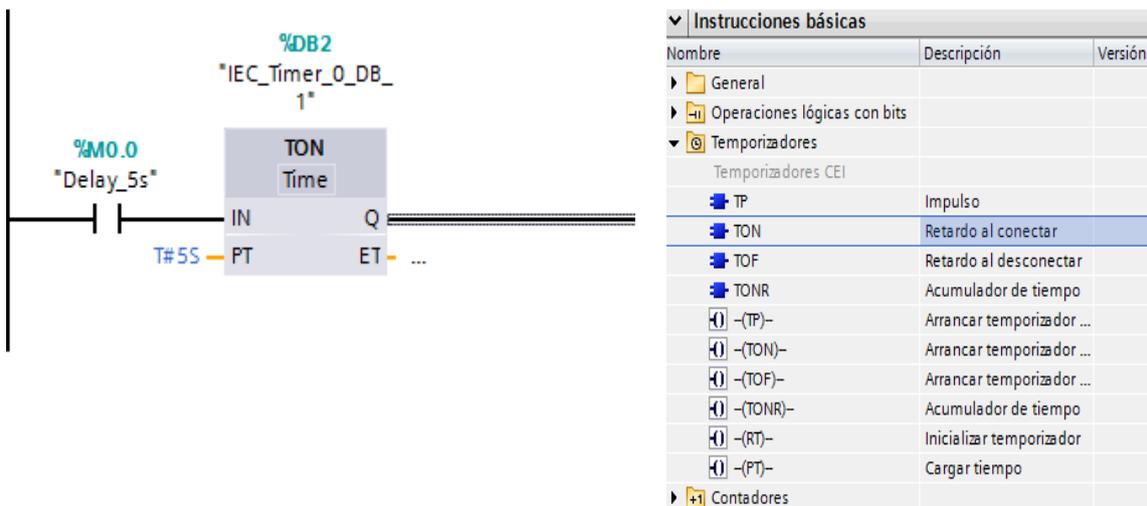


Figura 12. Insertando un "temporizador de retardo al desconectar".

**5.14.** Cree ahora una bobina que se active al activarse la instrucción TOF. Introduzca "M0.1" como dirección. Cambie el nombre de la variable a "On\_for\_5s".

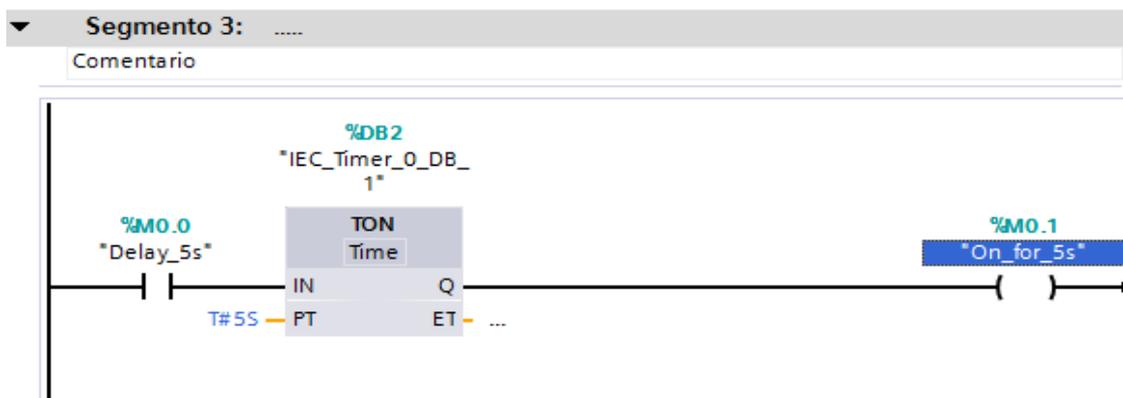


Figura 13. Segmento 3 terminado.

Cuando el temporizador TON active "Delay\_5s", el temporizador TOF activará el bit "On\_for\_5s" durante 5 segundos.

**5.15.** Para completar el programa de usuario, cree un cuarto segmento. Realice el mismo procedimiento del punto 5.4.

1. Inserte un contacto normalmente abierto. Para la dirección, seleccione la variable "On\_for\_5s".
2. Inserte una bobina. Introduzca "Q0.1" como dirección. Cambie el nombre de la variable a "MotorStart".



Figura 14. Segmento 4 terminado.

Cuando se active el temporizador TOF, "On\_for\_5s" activará la salida "MotorStart" durante 5 segundos.

#### 5.16. Guardar los ajustes realizados en el proyecto.

Ahora puede guardar los ajustes realizados. Haga clic en el botón "Guardar proyecto" en la barra de herramientas.

#### 5.17. Cargar el programa en la CPU.



Ahora puede cargar el programa de usuario en la CPU. Abra el editor de programas y haga clic en el botón "Cargar en dispositivo".

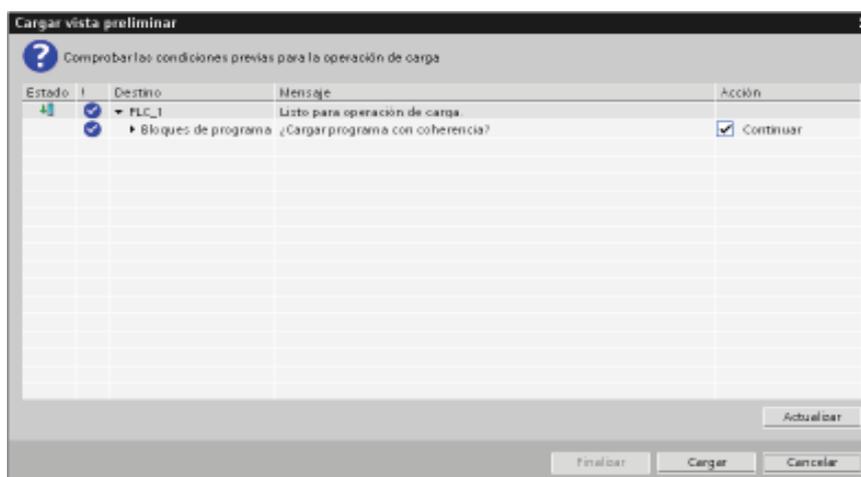


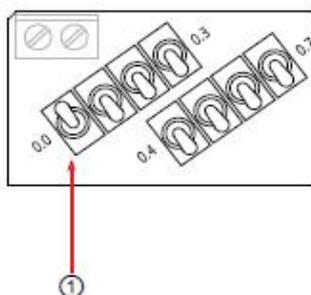
Figura 15. Cargando el programa a la CPU.

Tras establecer la conexión con la CPU, STEP 7 Basic muestra el diálogo "Cargar vista preliminar". Haga clic en "Cargar" para cargar el programa de usuario en la CPU. Antes de hacer clic en "Finalizar", seleccione "Arrancar todos" para asegurar que la CPU pase al estado operativo RUN.

### 5.18. Prueba de operación del programa de usuario.

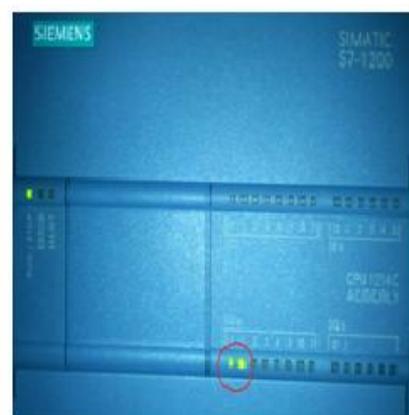
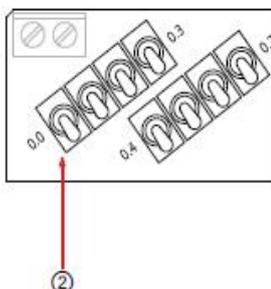
1. Active el interruptor I0.0

Los LEDs de estado de I0.0 y Q0.0 se encienden. Luego de 5 segundos se enciende el LED de estado Q0.1.



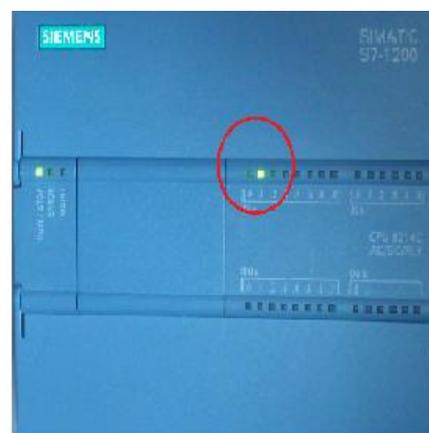
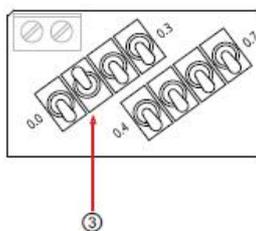
2. Desactive el interruptor I0.0.

El LED de estado de I0.0 se apaga, pero el LED de Q0.0 y Q0.1 permanece encendido.



3. Active el interruptor I0.1

El LED de estado de I0.1 se enciende. El de Q0.0 y Q0.1 se apagan. Desactive el Interruptor I0.1 y se apagan todo los LEDs de I y Q.



5.19. ♣ Tomar fotografías de la práctica para comprobar su funcionamiento.

## **6. CONCLUSIONES Y RECOMENDACIONES**

6.1. Exponga sus conclusiones respecto a la presente práctica.

6.2. Exponga sus recomendaciones respecto a la presente práctica

## **7. ANEXOS**

## PRÁCTICA No. 6

**NOMBRE (S):**

**FECHA DEL LABORATORIO:**

### 1. TÍTULO

Flip-flops.

### 2. OBJETIVOS

#### 2.1. GENERAL

Realizar un sencillo programa en Step 7 con flip-flops.

#### 2.2. ESPECÍFICOS

Aprender a programar al PLC Siemens S7-1200 con flip-flops.

### 3. LISTA DE MATERIALES Y EQUIPOS

- Modulo PLC S7-1200.
- Cable gemelo AWG 14, un extremo con toma corriente simple y el otro extremo pelado.
- Destornillador plano de 2.4mm.
- PC, con Step 7 Basic.
- Cable Ethernet punto a punto.
- Cámara fotográfica.
- Simulador Siemens de entradas y salidas de 8 posiciones.

#### 4. INDICACIONES GENERALES

Utilice un archivo electrónico para escribir sus respuestas y envíelo por correo a la siguiente dirección: *e-mail\_del\_profesor*.

Este símbolo ♣ indica que debe responder el correspondiente numeral y entregar el resultado en el archivo electrónico.

#### 5. PROCEDIMIENTO

5.1. Realice el mismo procedimiento que el de la práctica 3.

5.2. Realizar un programa utilizando flip-flops de manera que Interruptor I1 encienda el LED1. El interruptor I2 debe encender el LED2 y apagar el LED1. Al activar el interruptor I3 se enciende el LED3 y se apaga el LED2, si se activa el interruptor I4 se reiniciará el sistema.

5.3. En el “Árbol del proyecto” en “dispositivos” dar clic en el bloque Main [OB1] en donde se programa lo que se desea hacer.

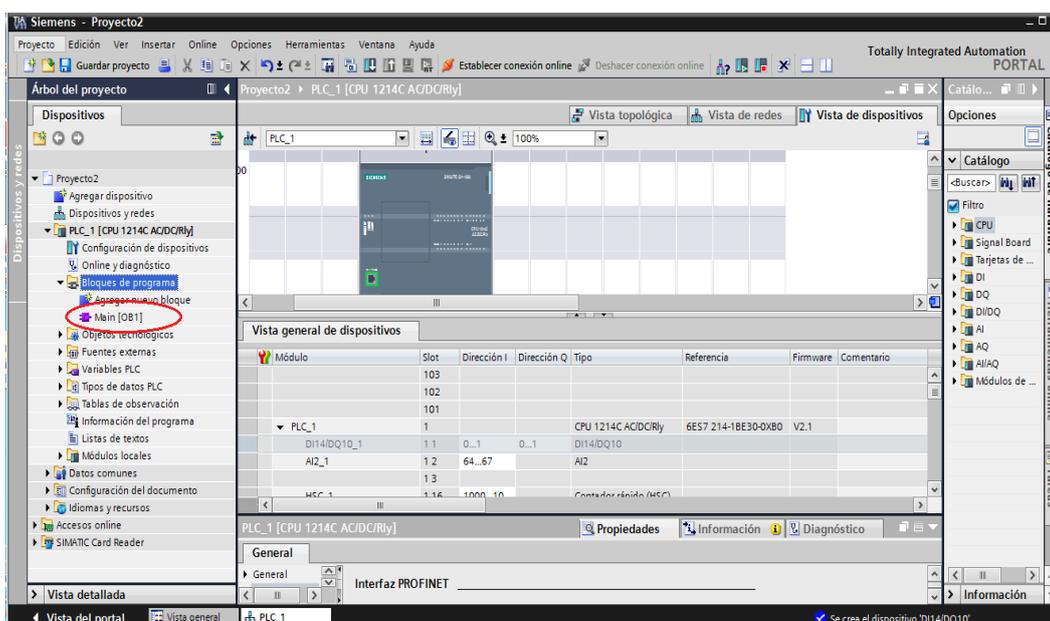


Figura1. Bloques de programación.

**5.4.** Haga clic en el “contacto normalmente abierto” en la barra de "Favoritos" que se encuentra en el “Área de trabajo” para insertarlo en el segmento.



Figura 2. Barra de Favoritos, “contacto normalmente abierto”.

**5.5.** Todos los circuitos digitales utilizan datos binarios para funcionar correctamente, los circuitos están diseñados para contar, sumar, separar, etc. Para esta práctica se utilizarán Flip-flops. En Step 7 se utiliza la instrucción “SR”



**SR: Flip flop de activación/desactivación.**

La operación "Flip flop de activación/desactivación" sirve para activar o desactivar el bit de un operando indicado en función del estado lógico de las entradas S y R1. Si el estado lógico de la entrada S es "1" y el de la entrada R1 es "0", se activa (se pone a "1") el operando indicado. Si el estado lógico de la entrada S es "0" y el de la entrada R1 es "1", se desactiva (se pone a "0") el operando indicado. La entrada R1 predomina sobre la entrada S. Si el estado lógico de las entradas S y R1 es "1", el estado lógico del operando indicado se pone a "0".

**5.6.** Expanda la carpeta "Operaciones lógicas con bits" en la Task Card "Instrucciones" y arrastre la instrucción “SR” hasta el segmento 1.

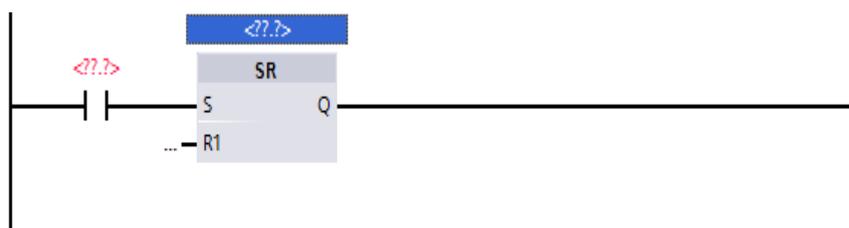


Figura 3. Insertando Flip-flop al segmento 1.

**5.7.** El “contacto normalmente cerrado” hace que la corriente fluya hasta que se active el interruptor. Activando un “contacto normalmente cerrado” se interrumpe el flujo de corriente.

Haga clic en el “contacto normalmente cerrado” en la barra de "Favoritos" que se encuentra en el “Área de trabajo” para insertarlo en el segmento.

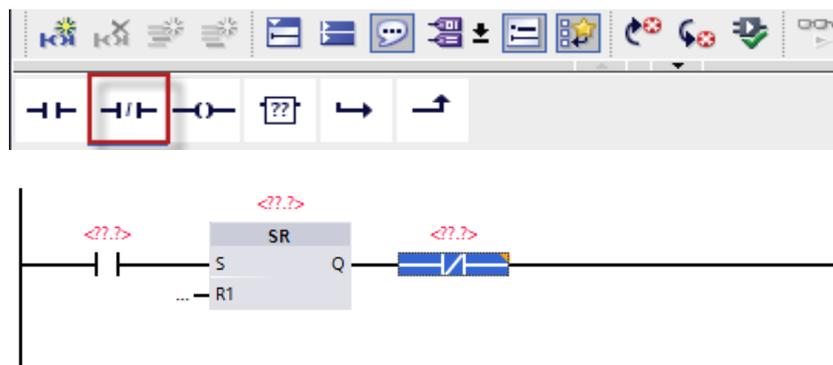


Figura 4. Barra de Favoritos, “contacto normalmente cerrado”.

**5.8.** La corriente fluye entre los dos contactos para excitar la bobina. La instrucción “asignación” asigna el resultado lógico del operando indicado.

Haga clic en la instrucción “Asignación” en la barra de "Favoritos" que se encuentra en el “Área de trabajo” para insertarlo en el segmento.

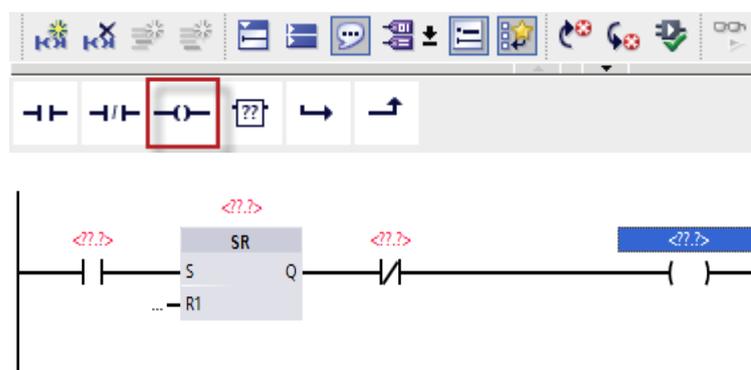


Figura 5. Barra de Favoritos, instrucción “asignación”

**5.9.** Introduzca las direcciones a las siguientes variables y cambie el nombre de las mismas:

- "I0.0" para el "contacto normalmente abierto" y dar el nombre de I1a la variable.
- "%M0.1" para la instrucción "SR" y dar el nombre de FLIP-FLOP1. A la entrada "R1" introduzca la dirección "%M0.0" y dar el nombre de "RESET" a la variable "R1".
- "Q0.1" para el "contacto normalmente cerrado" y dar el nombre de LED2 a la variable.
- "%Q0.0" para la instrucción "asignación" y dar el nombre de LED1.

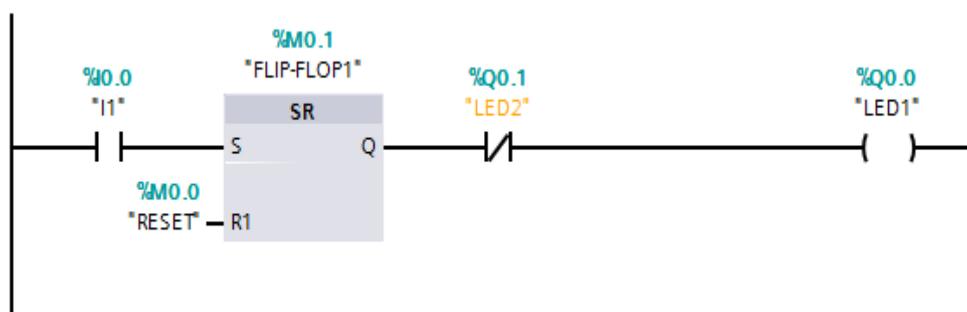


Figura 6. Segmento 1 terminado.

**5.10.** A continuación inserte un nuevo segmento al bloque del programa. Dar clic en la barra de herramientas que se encuentra en el área de trabajo.



Figura 7. Insertando nuevo segmento.

**5.11.** Haga clic en el “contacto normalmente abierto” en la barra de "Favoritos" que se encuentra en el “Área de trabajo” para insertarlo en el segmento.



Figura 8. Barra de Favoritos, “contacto normalmente abierto”.

**5.12.** Realice el mismo procedimiento del punto 5.11.

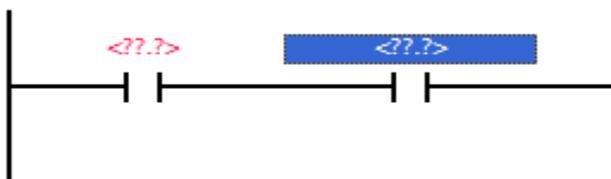


Figura 9. Agregando un “contacto normalmente abierto”.

**5.13.** Expanda la carpeta "Operaciones lógicas con bits" en la Task Card "Instrucciones" y arrastre la instrucción “SR” hasta el segmento 1.

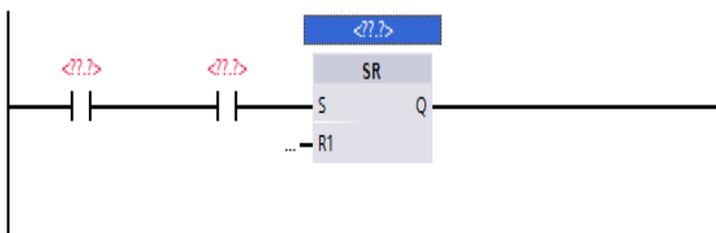


Figura 10. Insertando Flip-flop al segmento 1.

**5.14.** El “contacto normalmente cerrado” hace que la corriente fluya hasta que se active el interruptor. Activando un “contacto normalmente cerrado” se interrumpe el flujo de corriente.

Haga clic en el “contacto normalmente cerrado” en la barra de "Favoritos" que se encuentra en el “Área de trabajo” para insertarlo en el segmento.

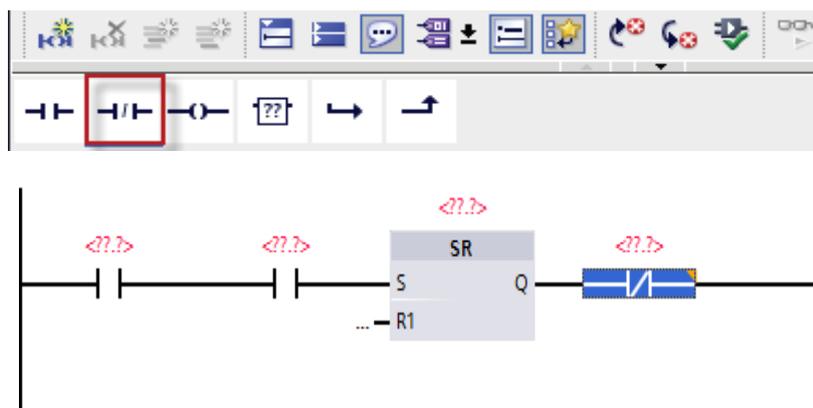


Figura 14. Barra de Favoritos, “contacto normalmente cerrado”.

**5.15.** La corriente fluye entre los dos contactos para excitar la bobina. La instrucción “asignación” asigna el resultado lógico del operando indicado.

Haga clic en la instrucción “Asignación” en la barra de "Favoritos" que se encuentra en el “Área de trabajo” para insertarlo en el segmento.

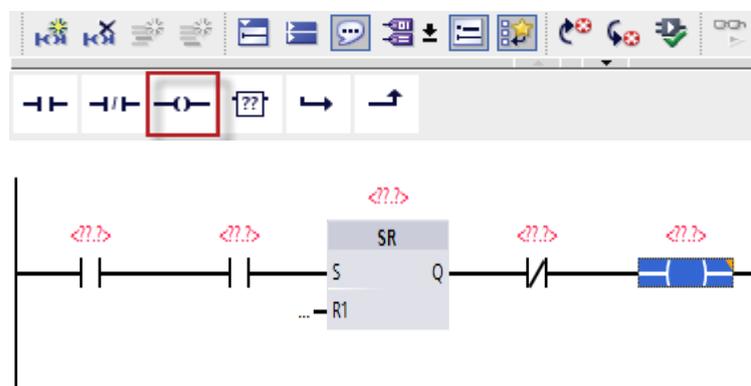


Figura 15. Barra de Favoritos, instrucción “asignación”

**5.16.** Introduzca las direcciones a las siguientes variables y cambie el nombre de las mismas:

- "I0.1" para el “contacto normalmente abierto” y dar el nombre de “I2” a la variable.

- "Q0.0" para el "contacto normalmente abierto" y dar el nombre de "LED1" a la variable.
- "%M0.2" para la instrucción "SR" y dar el nombre de "FLIP-FLOP2". A la entrada "R1" introduzca la dirección "%M0.0" y dar el nombre de "RESET" a la variable "R1".
- "Q0.2" para el "contacto normalmente cerrado" y dar el nombre de "LED3" a la variable.
- "%Q0.1" para la instrucción "asignación" y dar el nombre de LED2.

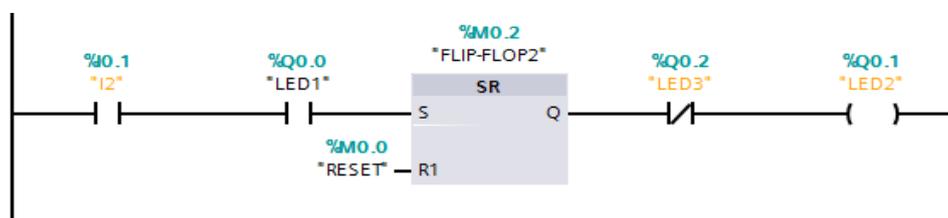


Figura 16. Segmento 2 terminado.

**5.17.** Realice el mismo procedimiento que el de los puntos 5.10, 5.11, 5.12, 5.13, 5.15.

**5.18.** Introduzca las direcciones a las siguientes variables y cambie el nombre de las mismas:

- "Q0.2" para el "contacto normalmente abierto" y dar el nombre de "I3" a la variable.
- "Q0.1" para el "contacto normalmente abierto" y dar el nombre de "LED2" a la variable.
- "%M0.3" para la instrucción "SR" y dar el nombre de "FLIP-FLOP3". A la entrada "R1" introduzca la dirección "%M0.0" y dar el nombre de "RESET" a la variable "R1".
- "%Q0.2" para la instrucción "asignación" y dar el nombre de LED3.

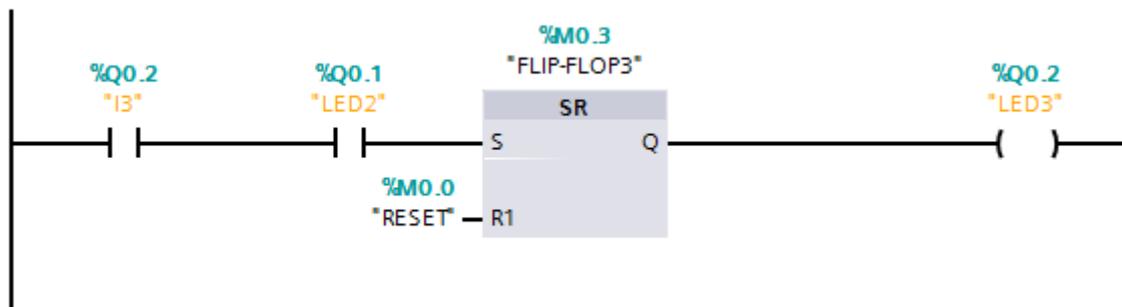


Figura 17. Segmento 3 terminado.

**5.19.** Para completar el programa de usuario, cree un cuarto segmento. Inserte un contacto normalmente abierto. Utilice “%I0.3” para la dirección y de el nombre de I4 a la variable. Luego inserte una bobina (instrucción “asignación”). Utilice “%M0.0” para la dirección.

♣ ¿Qué nombre adquirió la variable?

**5.20.** Prueba de operación del programa de usuario.



Figura 18. Modo inicial

1. Al activar el interruptor 1 (I0.0) se enciende el foco1 (Q0.0).



Al desactivarlo se queda encendido el foco1 (Q0.0).



2. Al activar el interruptor 2 (I0.1) se encienda el foco2 (Q0.1) y se apague el foco1 (Q0.0)



Al desactivarlo quede encendido el foco2 (Q0.1).



3. Por último al activar el interruptor 3 (I0. 2) y se reiniciará el sistema.



Al desactivarlo vuelve al estado inicial.



- 5.21. ♣ Tomar fotografías de la práctica para comprobar su funcionamiento.

## 6. CONCLUSIONES Y RECOMENDACIONES

6.1. Exponga sus conclusiones respecto a la presente práctica.

6.2. Exponga sus recomendaciones respecto a la presente práctica.

## 7. ANEXOS