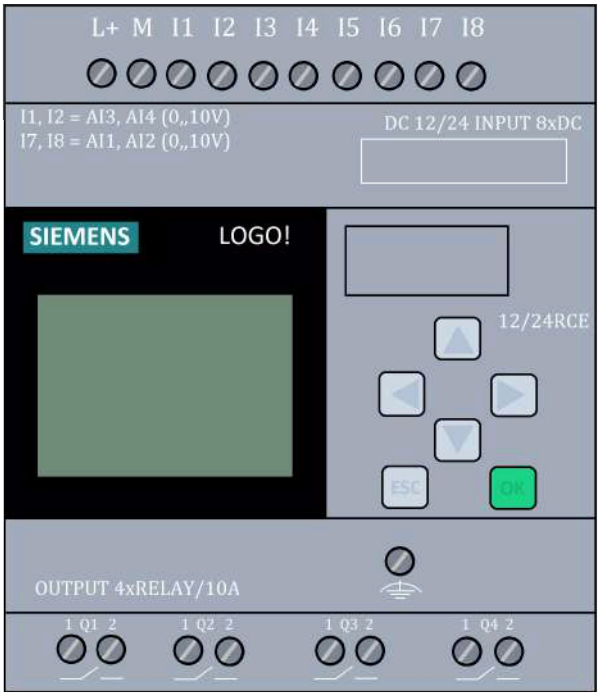
Cuadernillo del Taller de Automatización Industrial



por Tec. LARDET, Juan Pedro

2026

Autor: Juan Pedro Lardet
Año: 2026
https://doi.org/10.5281/zenodo.18674944 V3
https://doi.org/10.5281/zenodo.18666592 V2
https://doi.org/10.5281/zenodo.18666093 V1

Índice

Introducción	pág. 3
Programación	pág. 9
LOGO! Soft Comfort 8.4	pág. 13
Entorno de Simulación 3D	pág. 19
Ejercitación	pág. 25
CADe SIMU	pág. 28
TIA Portal V17 Step 7	pág. 38
S7 PLCSIM y Factory I/O	pág. 49
Más Ejercicios	pág. 58

Introducción

¿Qué es la automatización?
¿Qué es la programación?
¿Qué es un PLC?

En las ramas de la ingeniería actual priman conceptos como el de automatización, que sería algo así como *lograr que un sistema opere por su cuenta*. ¿Cómo es esto posible? ¿Cómo se logra? ¿Hay una sola forma?

Analicemos un caso: **se sube una pelota a la cima de una montaña. Luego se la suelta y esta cae por efecto y causa de la gravedad.**

Este fenómeno no es posible sin la acción previa del ser humano.

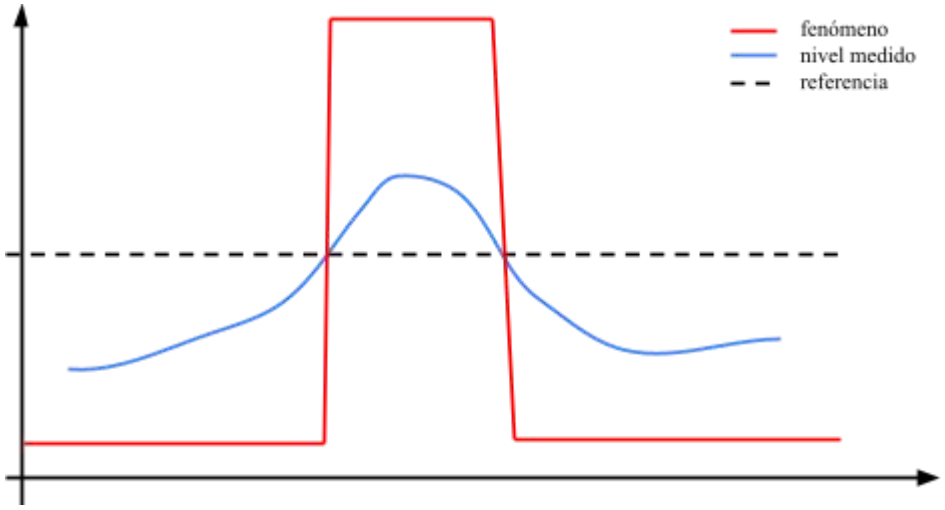
Todo automatismo requiere al ser humano. Antes, durante y después del fenómeno.

- Antes: planificación, preparación y configuración.
- Durante: control y mantenimiento.
- Después: reciclaje.

Existen diversos tipos de automatismos:

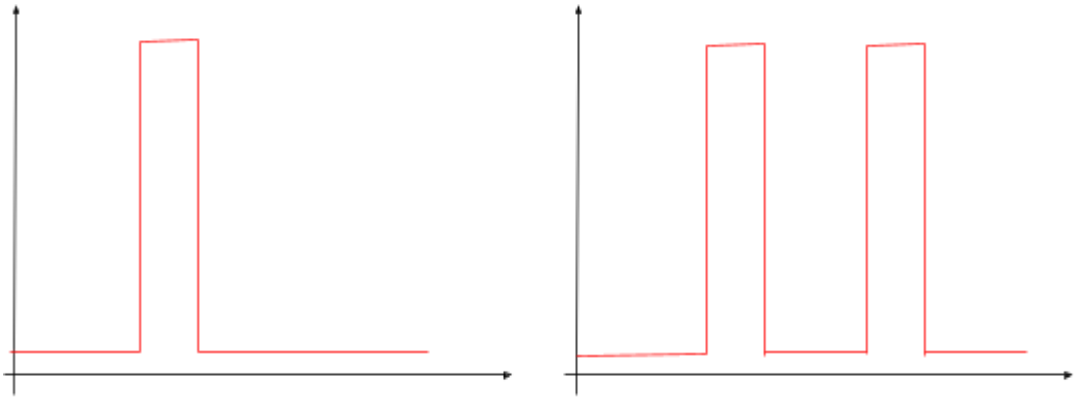
- Nivel cero (manual): control directo del operador.
  Ej. Sistema interruptor + lámpara.

- Nivel uno (comparador o por nivel): si el nivel medido es igual o superior a uno de referencia ocurrirá el fenómeno, a menos que que el nivel esté por debajo. En dicho caso, se interrumpirá el fenómeno.
  Ej. Flotante del tanque.



- Nivel dos (temporizado): es un caso aplicado del anterior, donde un fenómeno es activado solo cuando un fenómeno de control que dura un tiempo regular o que "siempre tarda lo mismo" alcanza un nivel de referencia configurado.

  Existen dos tipos a destacar:
  - Monoestable, el fenómeno ocurre una sola vez luego de aquello que lo dispara.
  - Astable, el fenómeno se repite en intervalos regulares.



Monoestable Astable

Todos estos ejemplos refieren a sistemas de comportamiento binario: encendido-apagado. Pero es posible complejizar esa mirada con información continua, como la posición, presión, temperatura, voltaje, caudal, etc.

Es así que muchos sistemas físicos en el presente son capaces de comportamientos binarios y/o pseudo continuos (casi continuos). Los Controladores Lógicos Programables (PLC) son ejemplos de esto.

Estos dispositivos pueden recibir instrucciones adecuadas, paso a paso de cómo resolver problemas determinados. Es a esto a lo que nos referimos con **programación**.

Existen diferentes tipos de problemas que pueden ser resueltos mediante programación. Estos dependen de las estrategias disponibles.

- Programación Cableada: el comportamiento de los sistemas dependen de las conexiones entre los diferentes tipos de elementos que los constituyen.

  Ejemplos de la programación cableada pueden ser sistemas tales como el automatismo básico de la cisterna de una casa, con relays, contactores, flotantes, bombas e interruptores.

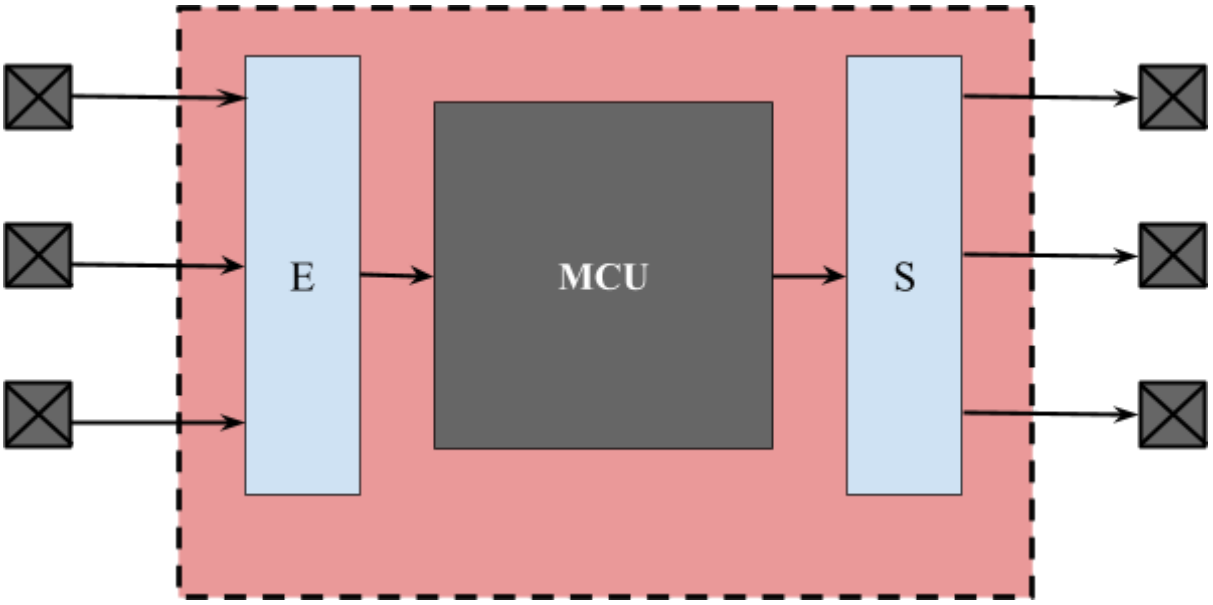
  También es posible interpretar que muchos circuitos integrados son de este tipo, como las compuertas lógicas y sistemas discretos compuestos por estas.

- Programación Embebida: se graba el comportamiento en el silicio, ya sea de forma permanente o semipermanente, particularmente en sus memorias. Esto es lo que se embebe.

  Ejemplos de la programación embebida son:
  - DSP, circuitos integrados dedicados al procesamiento de señales. Pueden procesar más de una señal simultáneamente, son altamente eficientes y veloces, costosos de diseñar.
  - FPGA, parecidos a los anteriores en el sentido que pueden procesar múltiples señales en simultáneo, sin embargo son más lentos. Es ventajoso que pueden ser reprogramados, para experimentar con ellos hasta llegar al diseño correcto.
  - MCU, son opciones muy populares y seguras, típicas en la industria y la computación. Son más económicos que los anteriores, pero son considerablemente más lentos, debido a que pueden procesar múltiples señales, pero solo de manera secuencial (de a una).

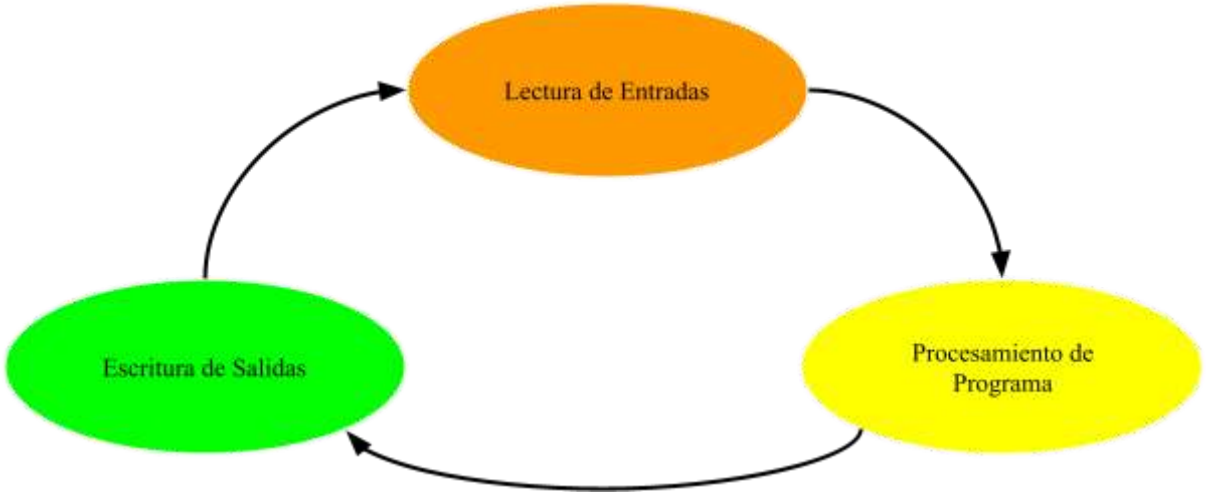
Los PLC poseen como cerebro uno o más MCU.

Es importante destacar que muchos de los dispositivos electrónicos, entre ellos los mencionados DSP, FPGA y MCU son muy sensibles a las condiciones de trabajo del ambiente industrial. Por lo tanto deben ser protegidos de manera acorde.

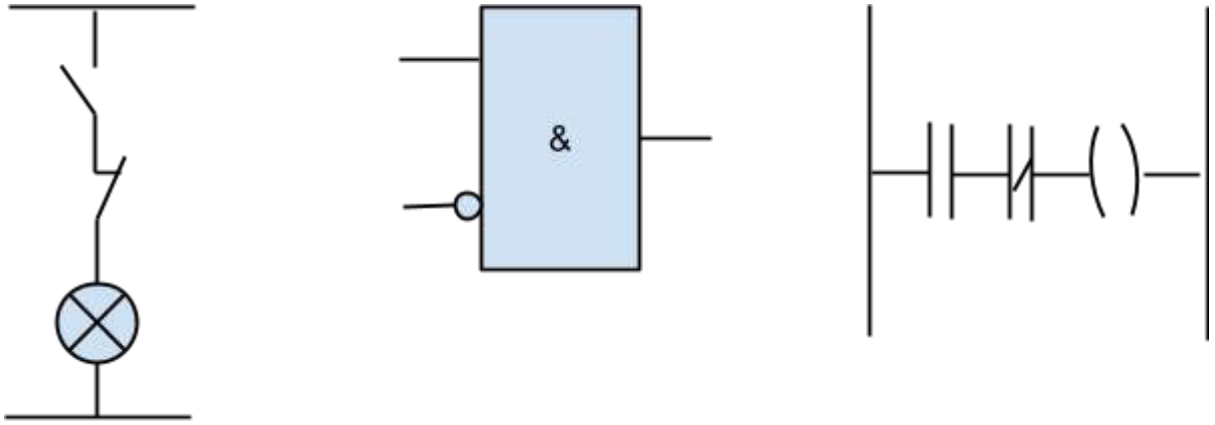


Se emplean circuitos específicos para Entrada y Salida, que aislan eléctricamente al MCU del exterior, y todo el circuito se protege mediante una Caja de Faraday, para prevenir los efectos electromagnéticos.

Los PLC son capaces de procesar múltiples datos de entrada y producir datos de salida. Pero solo de manera secuencial: **uno a uno**. Por esto es que se emplea algo denominado ciclo de escaneo. Que consiste en una secuencia que se repite para evaluar primero las entradas, almacenar su información en la memoria interna del MCU, luego procesar el código principal, y luego escribir los datos de salida.



A pesar de la resolución secuencial del algoritmo principal, los PLC pueden ser programados en más de una manera, inspiradas en aquellas de la programación cableada, que no es secuencial sino simultánea.



Prog. Cableada FBD/FUP LADDER/KOP

La “simultaneidad” que poseen los sistemas embebidos de los PLC, depende de un ciclo de escaneo de gran velocidad. Típicamente de 1 mS a 150 mS. Esto es considerablemente lento comparado a un MCU para comunicaciones. Pero para control de pulsadores, fines de carrera, flotantes, transductores industriales (sensores) y actuadores (motores, bombas, lámparas) es más que suficiente.

Para la comunicación con dispositivos extras como interfaces humano-máquina (HMI), variadores de frecuencia, switches industriales, se usan MCU especializados que deben ser incluidos en el interior del PLC. De no ser así, estas funcionalidades no podrán ser incluidas en el sistema.

La mayoría de los datos que emplearemos dentro de estos sistemas, incluso aquellos que se usan para HMI, switches, etc., son digitales o binarios. Es decir, que los datos representados son del tipo Encendido-Apagado, ON-OFF, o 0 y 1 lógicos.

¿Qué es esto?

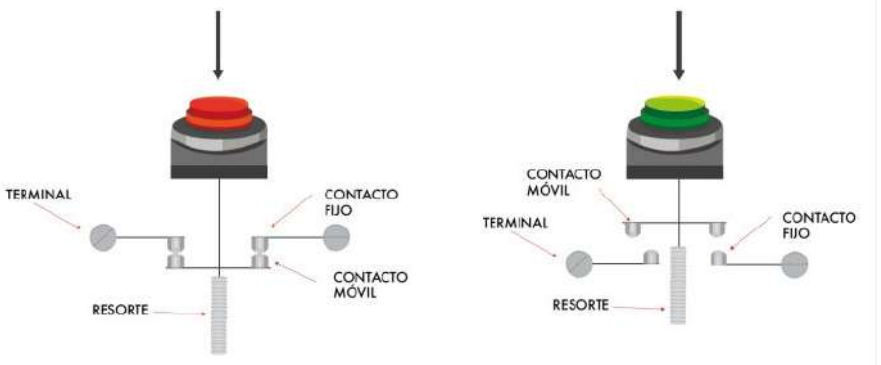
La representación de información binaria es una tradición en las ingenierías, en la matemática, e incluso en la conciencia y cómputo neurobiológico de muchos organismos. Se trata de la forma más básica de contar: hay o no hay. Mientras que estamos acostumbrados a sistemas como “en la casa de mi vecino hay 5 ovejas más que en la mía” (que requiere una cognición más avanzada), seres vivos y máquinas tenemos en común la computación (cuenta) de cantidades de una manera más fundamental: hay - no hay, encendido - apagado, lleno - vacío, caliente - frío, bueno - malo, etc.

Empleamos esta lógica con interruptores típicamente, mediante presionar una tecla de luz, podemos prender, y si presionamos el otro extremo de la misma, la luz se apaga.

Esta es una forma muy esencial de representar información. Llamamos a la unidad básica de información en la representación binaria cómo bit. El estado físico apagado suele ser representado con el 0 lógico, usando un bit de información. Para el estado físico encendido suele usarse el 1 lógico, nuevamente usando un bit de información.

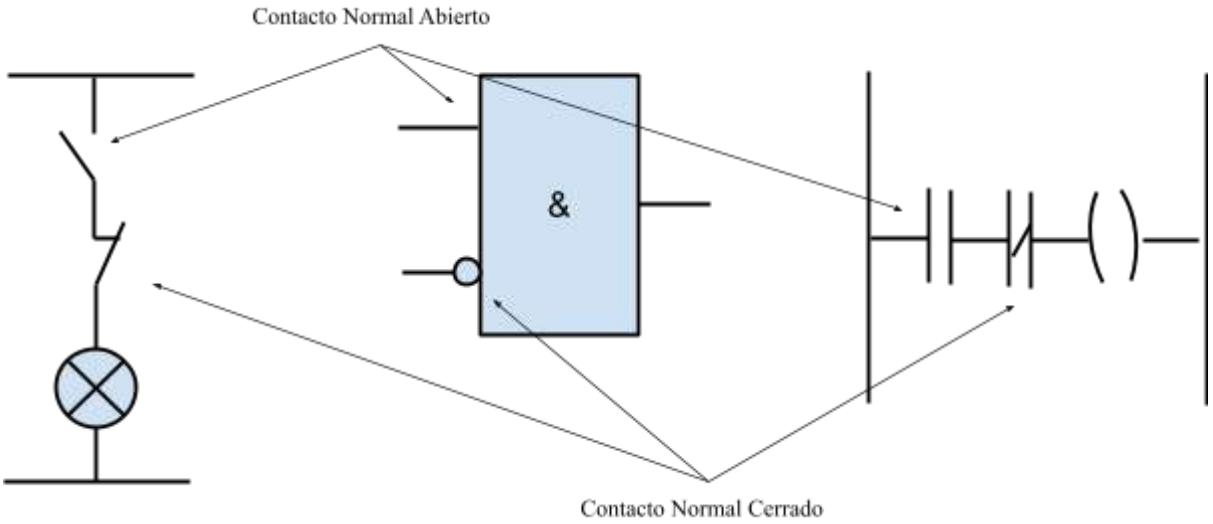
Podemos hacer sistemas que empleen operaciones entre múltiples bits de información provenientes de diferentes fuentes. Por ejemplo: NEGACIÓN, SUMA y MULTIPLICACIÓN binarias o booleanas.

El caso más simple: NEGACIÓN. Se requiere un pulsador del tipo opuesto al que se utilizará en el cotidiano. Para eso, es importante ponernos todos de acuerdo con qué tipo de pulsador usaremos para afirmar, y cual para negar.

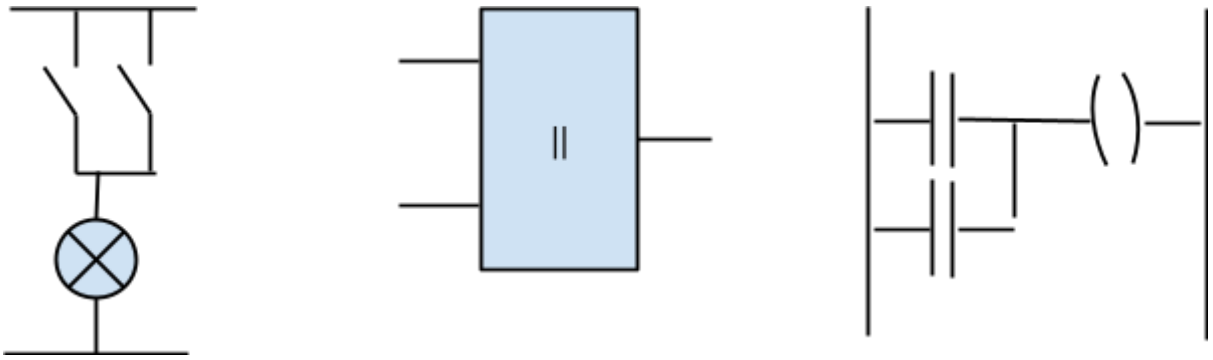


El de la izquierda, es un pulsador o contacto normalmente cerrado, porque el circuito eléctrico tiene continuidad entre terminales, la cual se interrumpe en caso de que se presione. Mientras que el de la derecha es un pulsador o contacto normalmente abierto, porque el circuito no tiene continuidad entre terminales a menos que se presione.

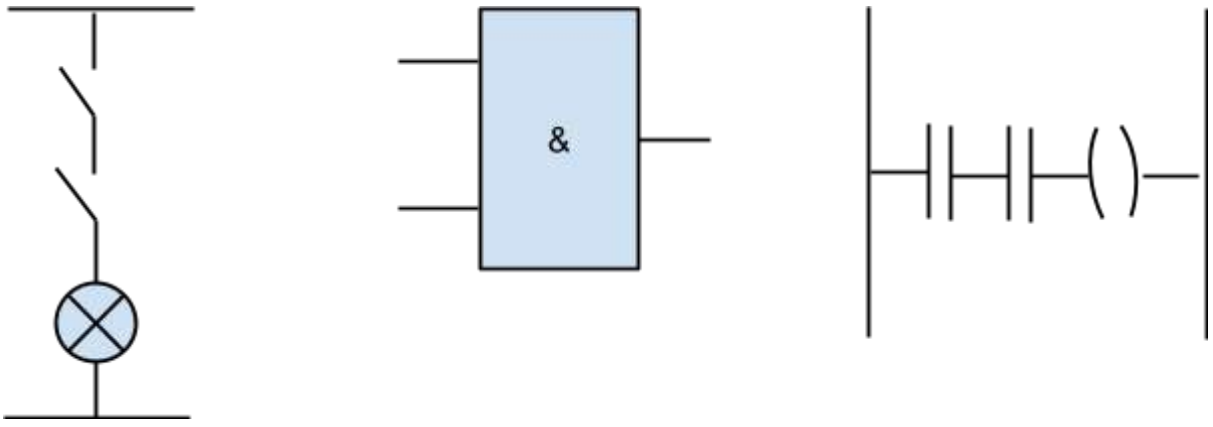
Decimos que para afirmar se utiliza el normal abierto. Mientras que para negar el normal cerrado. En el dibujo se representa uno de cada uno.



Veamos a continuación cómo se hace la operación SUMA: para esto se conectan dos pulsadores en paralelo.

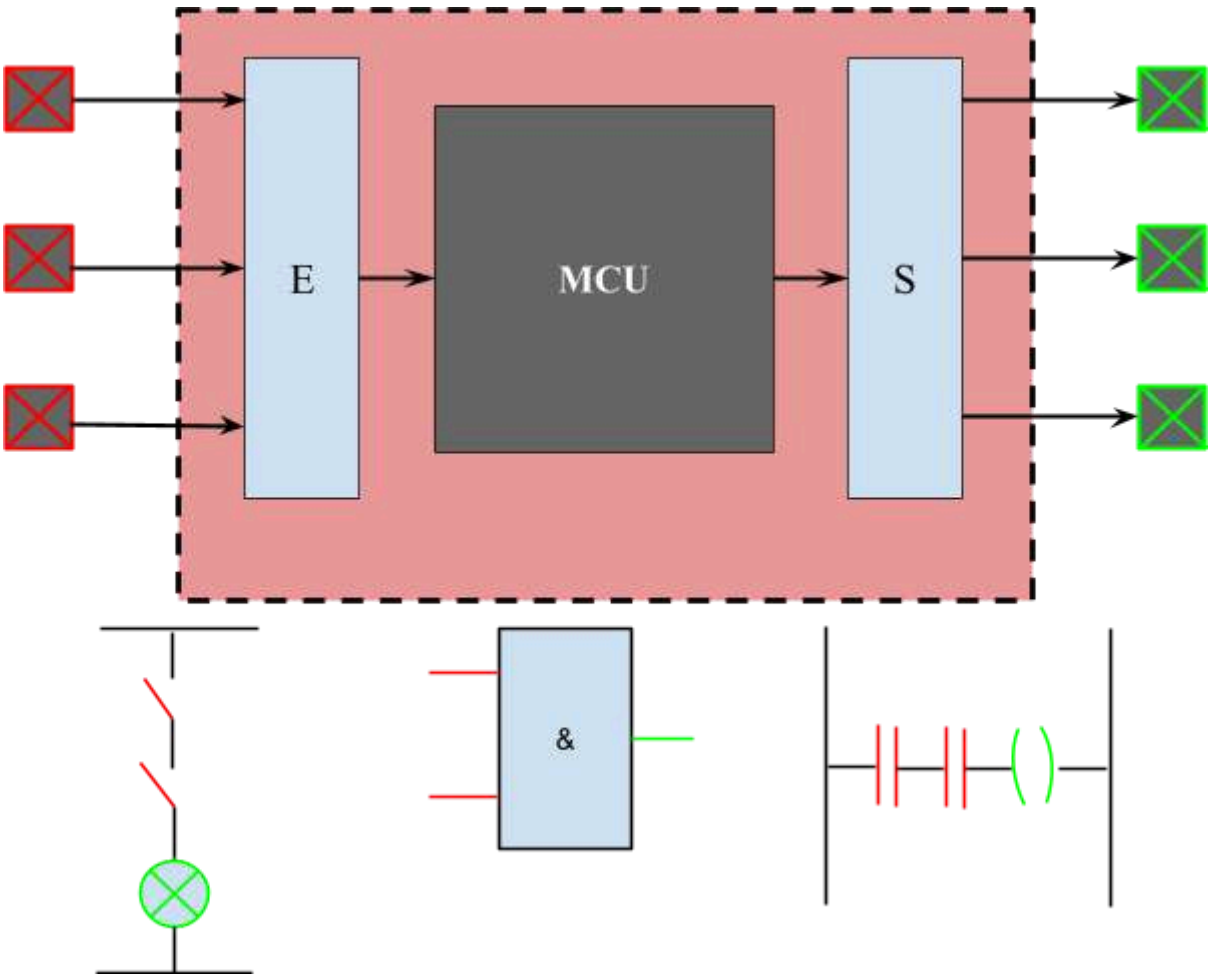


También es posible ponerlos en serie, logrando así la operación MULTIPLICACIÓN.



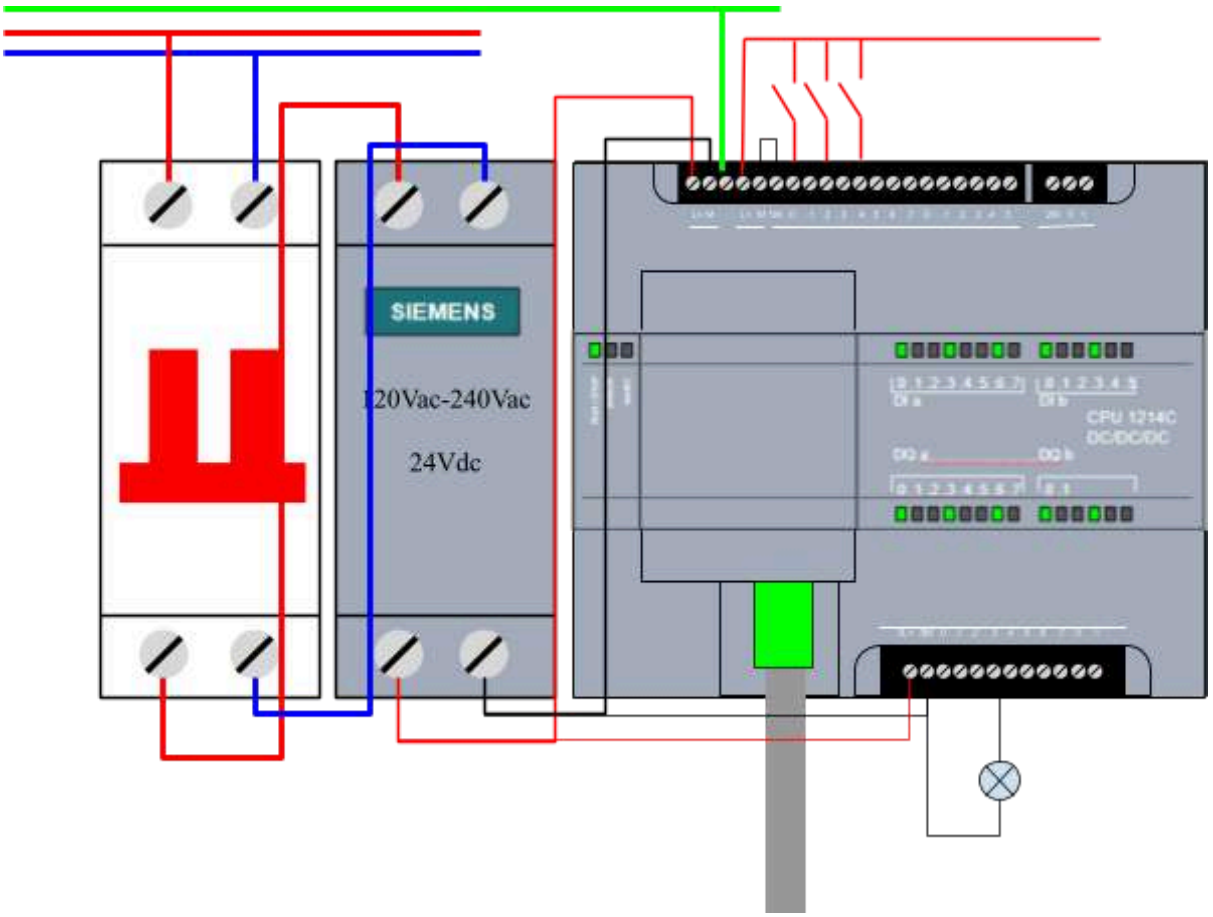
Programación

Analicemos con detenimiento lo que hemos hecho hasta ahora: hemos presentado a grandes rasgos el funcionamiento interno de los PLC y de sus programas. Ahora profundicemos particularmente uno de los lenguajes de programación más conocidos para la aplicación industrial: el lenguaje escalera (LADDER, o KOP en SIEMENS).



Existen Entradas (en Rojo) y Salidas (en Verde). Típicamente estas son contactos eléctricos (para entradas) y bobinas de relays (para salidas). En LADDER representamos las primeras con dos líneas paralelas, mientras que las segundas con dos curvas paralelas.

Identifiquemos cómo conectaríamos un PLC real:



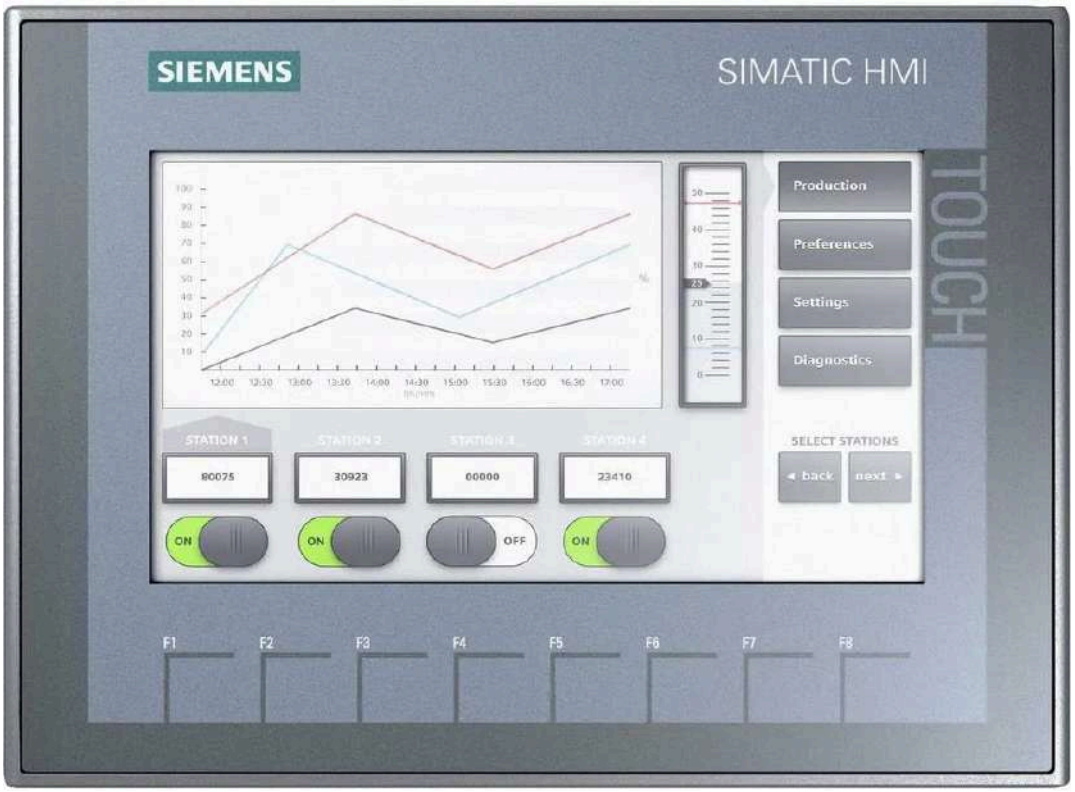
Esto depende mucho de los modelos disponibles, pues las empresas SIEMENS, Schneider Electrics, Delta, etc., producen dispositivos muy variados entre sí.

Durante el proceso de este taller emplearemos sobre todo dispositivos de SIEMENS.

SIEMENS llama al LADDER o lenguaje escalera KOP (lenguaje de contactos en alemán). Fabrica diversos aparatos diseñados para el automatismo. KOP es el estándar que emplean muchos de estos, sin embargo no es el único. Los PLC suelen ser de mayor tamaño y estar abocado a empresas, pero para ambientes educativos también hay disponibles LOGO! que son como los PLC pero más compactos.

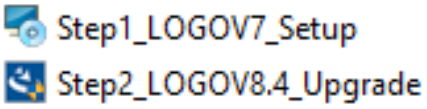
Tanto LOGO! como PLC requieren infraestructura adecuada para programarlos. Por ejemplo, una computadora, con el software específico para programar, simular o testear, así como también poder subir el código al MCU (llamada transferencia de PC a CPU) mediante un cable o conector adecuado. Hoy en el presente nos podemos encontrar con equipos viejos en funcionamiento que poseen una interfaz específica para esto, que se conecta al USB de la computadora, se adapta al RS-232 o RS-485, y de ahí al CPU. Sin embargo, en el presente nos encontramos con que muchos se pueden programar mediante ethernet. Para lo cual vienen Switches industriales de la empresa.

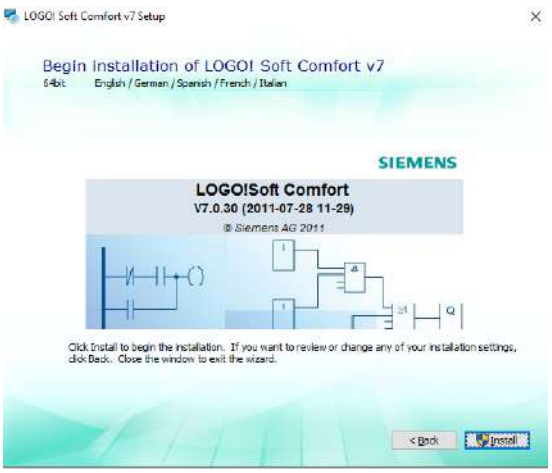
También vienen pantallas o displays comunes y táctiles para permitirle a los operadores interactuar con la maquinaria controlada por el o los PLC. Conocemos a estas pantallas como HMI.

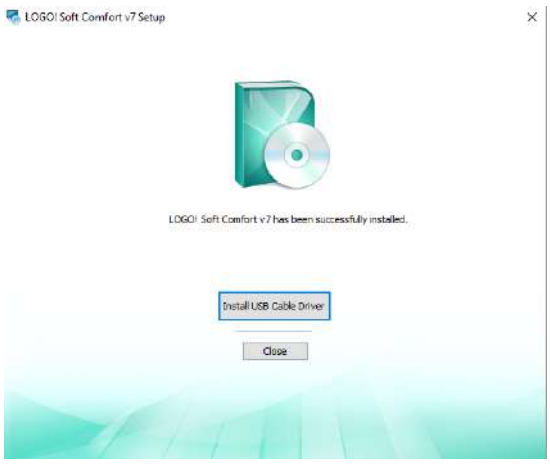


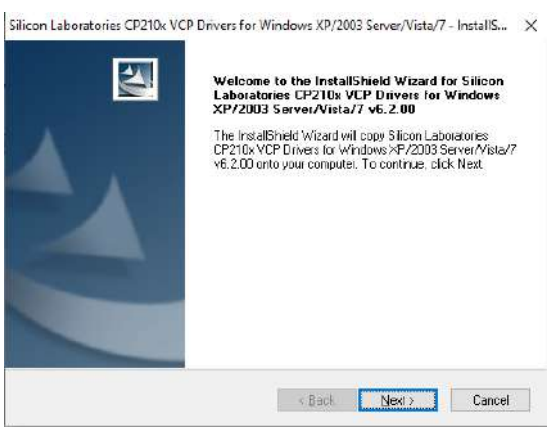
Iniciemos por programar un LOGO! 8.3, que contiene el conector ethernet. Por lo que es posible conectarlo a una computadora de manera directa o a un router hogareño. El software específico es el LOGO! Soft Comfort, y recomendamos emplear la versión 8.4.

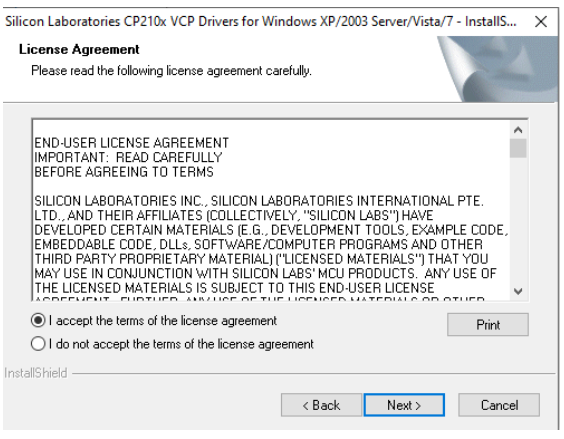
Para ello requerirán la documentación oficial.

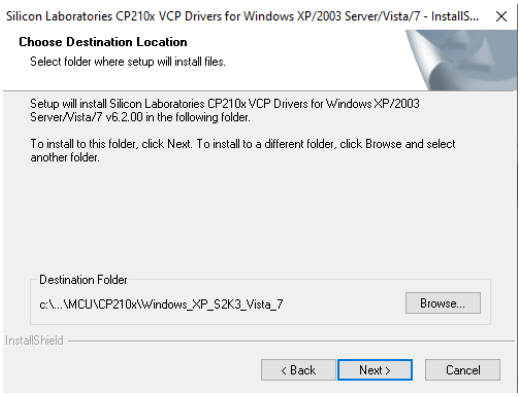


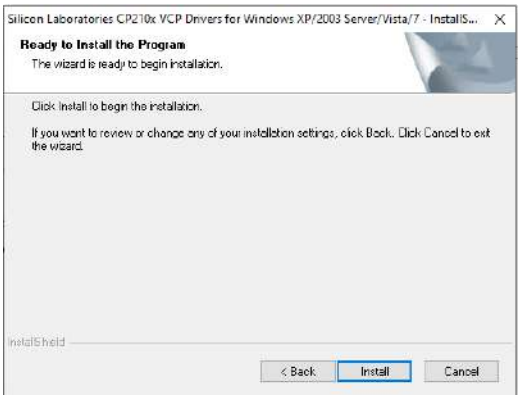


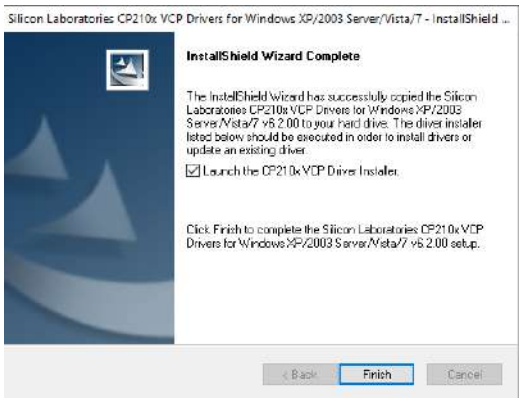


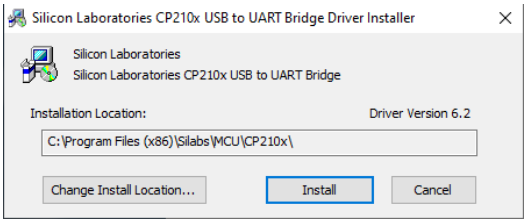


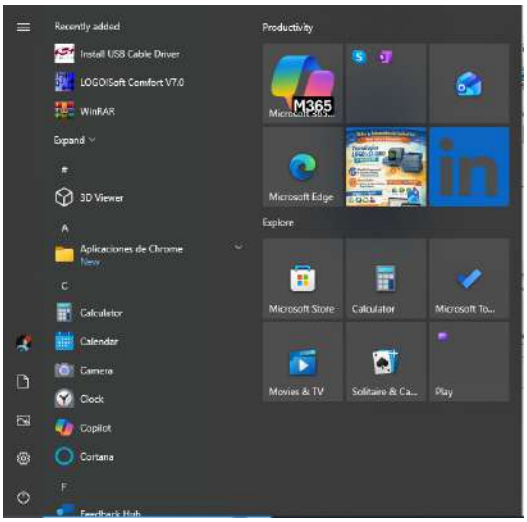


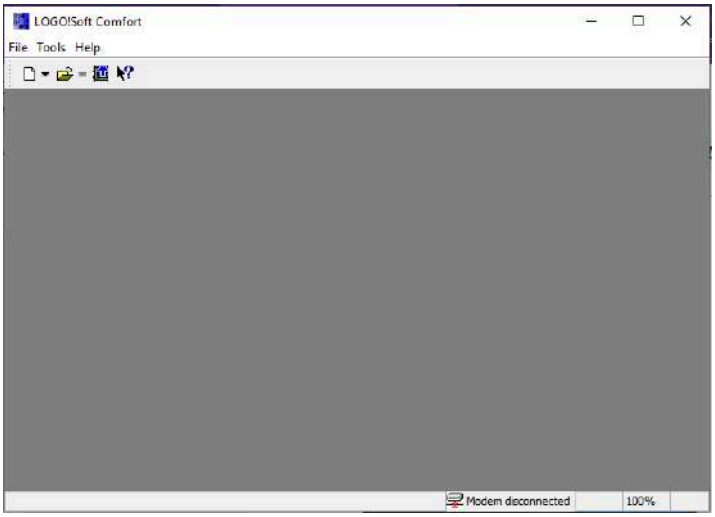




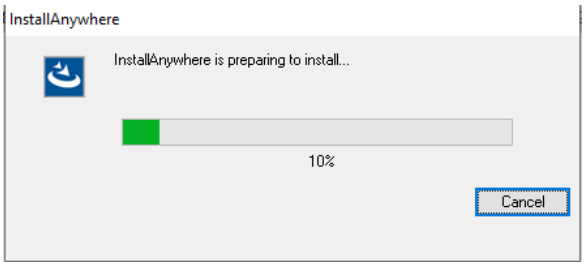






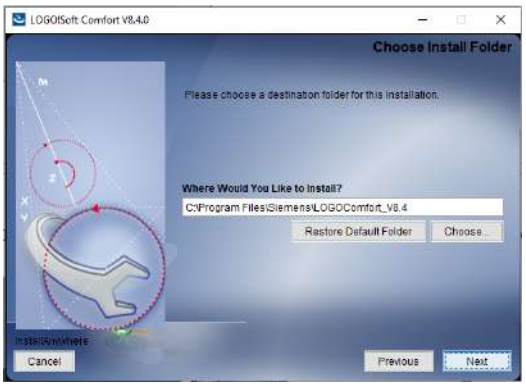


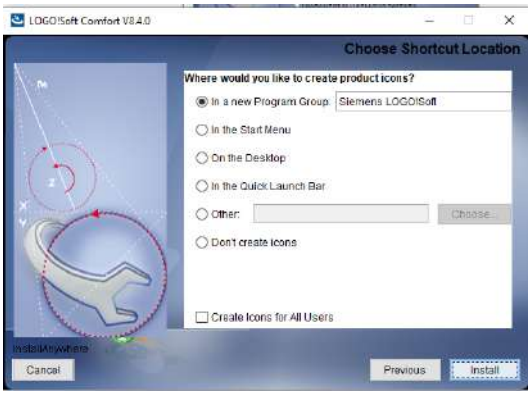
Continuamos con la actualización




Install LOGO!Soft Comfort 8.4.0
SIEMENS
LOGO! Soft Comfort
Version 8.4.0
LOGO!
© Siemens AG, 2023 All Rights Reserved.
Hello, welcome to V8.4.0 installer
English
OK

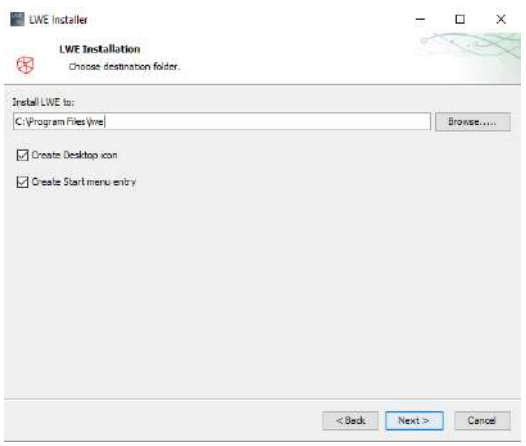

LOGO!Soft Comfort V8.4.0
License Agreement
Please read through the following license agreement carefully, including the security information it contains.
Please note:
This software is protected under German and/or US American Copyright Laws and provisions in international treaties. Unauthorized reproduction and distribution of this software or parts of it is liable to prosecution. It will be prosecuted according to criminal as well as civil law and may result in severe punishment and/or damage claims.
I accept all conditions of the listed license agreement(s). I hereby confirm that I have read and understood the security information on the safe operation of the products.
I accept the terms of the License Agreement
I do NOT accept the terms of the License Agreement
InstallAnywhere
Cancel
Previous
Next

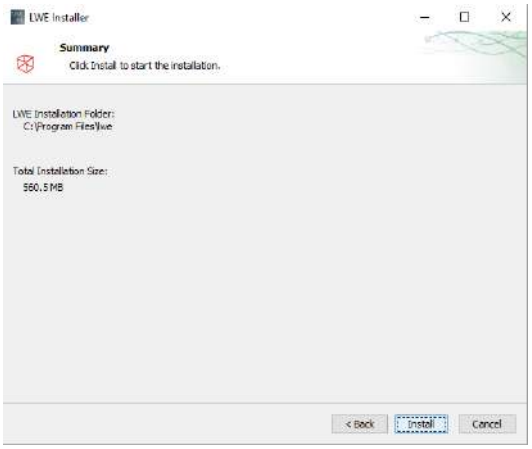

LOGO!Soft Comfort V8.4.0
Choose Install Folder
Please choose a destination folder for this installation.
Where Would You Like to Install?
C:\Program Files\Siemens\LOGOComfort_V8.4
Restore Default Folder
Choose...
InstallAnywhere
Cancel
Previous
Next


LOGO!Soft Comfort V8.4.0
Choose Shortcut Location
Where would you like to create product icons?
In a new Program Group
Siemens LOGO!Soft
In the Start Menu
On the Desktop
In the Quick Launch Bar
Other:
Choose...
Don't create icons
Create Icons for All Users
InstallAnywhere
Cancel
Previous
Install

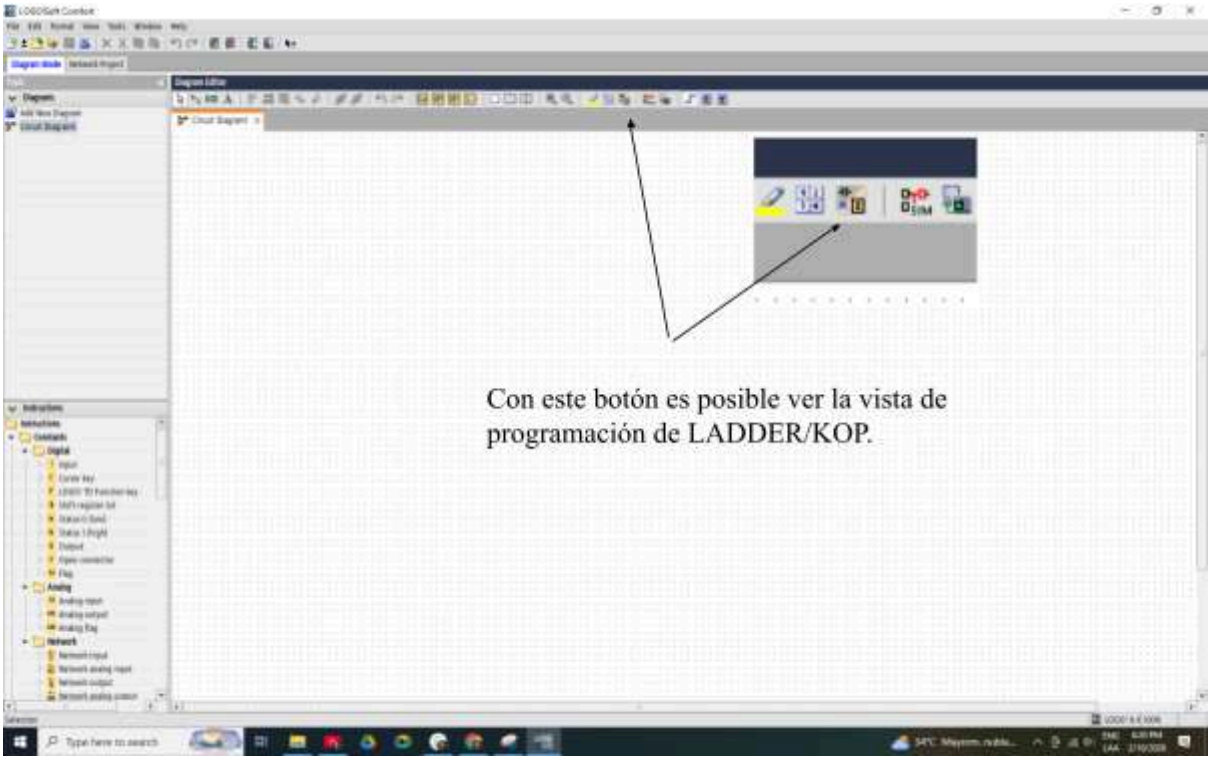

LOGO!Soft Comfort V8.4.0
Installing LOGO!Soft Comfort V8.4.0
New Features
– Support downloading one circuit diagram to a batch of connected BMs
– Support downloading a batch of circuit diagrams to all connected BMs
Installing... Uninstall.exe
InstallAnywhere
Cancel
52%

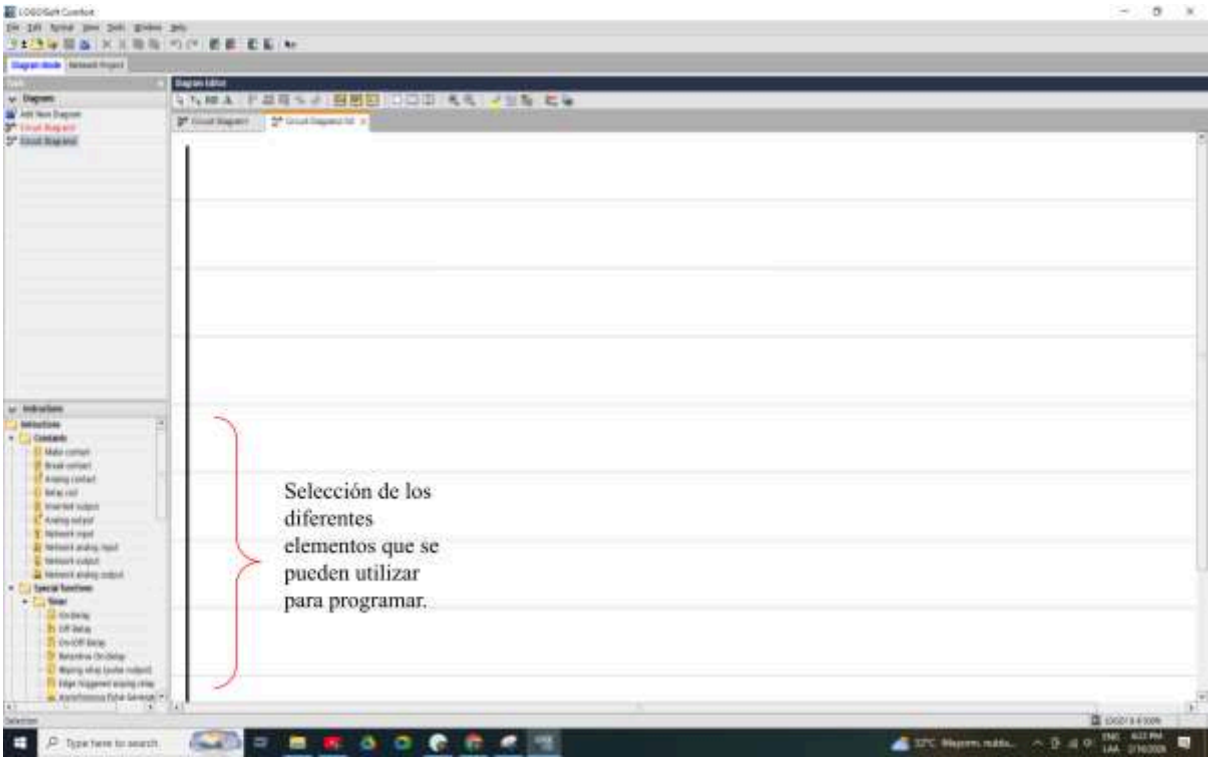

LWE Installer
Welcome to LWE Installer Wizard
This program will install LWE application on your computer.
Click Next to continue.
Next >
Cancel


LWE Installer
LWE Installation
Choose destination folder.
Install LWE to:
C:\Program Files\lwe
Browse.....
Create Desktop icon
Create Start menu entry
< Back
Next >
Cancel


LWE Installer
Summary
Click Install to start the installation.
LWE Installation Folder:
C:\Program Files\lwe
Total Installation Size:
560.5 MB
< Back
Install
Cancel

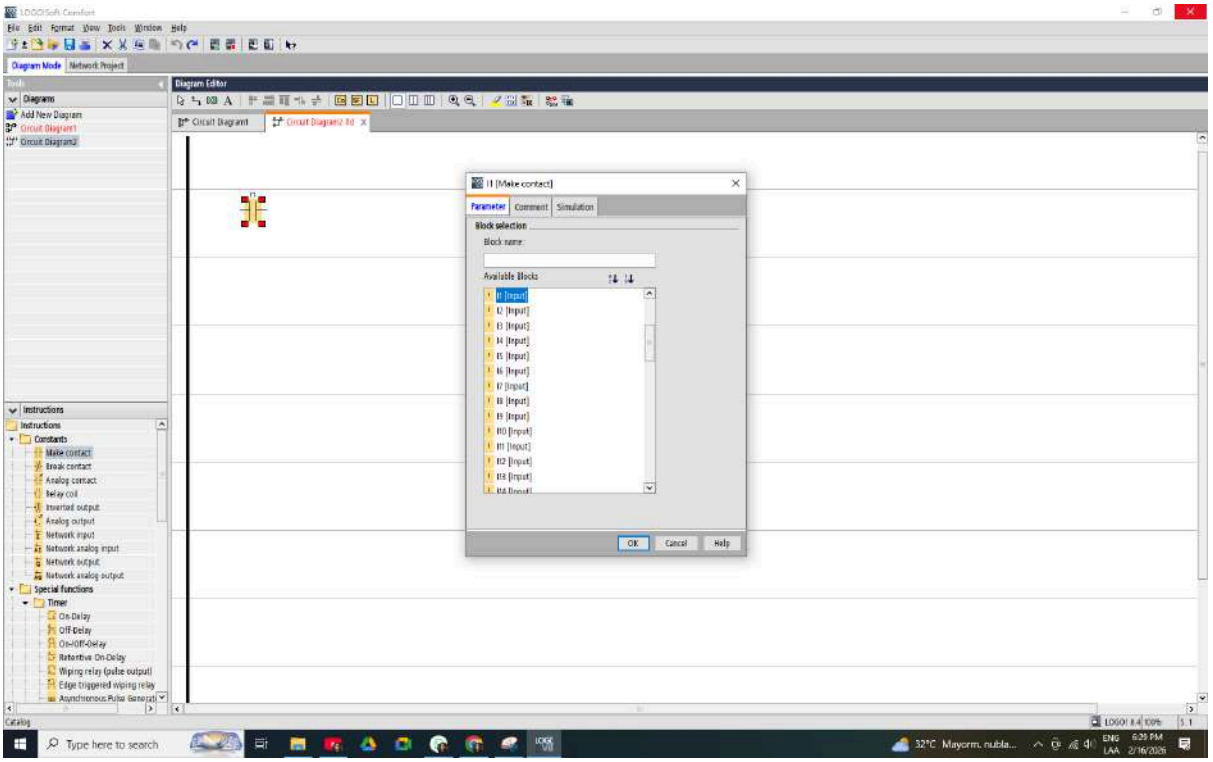
Una vez instalado, les invito a ir de a poco observando los diferentes elementos que vienen con la interfaz de LOGO! Soft Comfort 8.4.



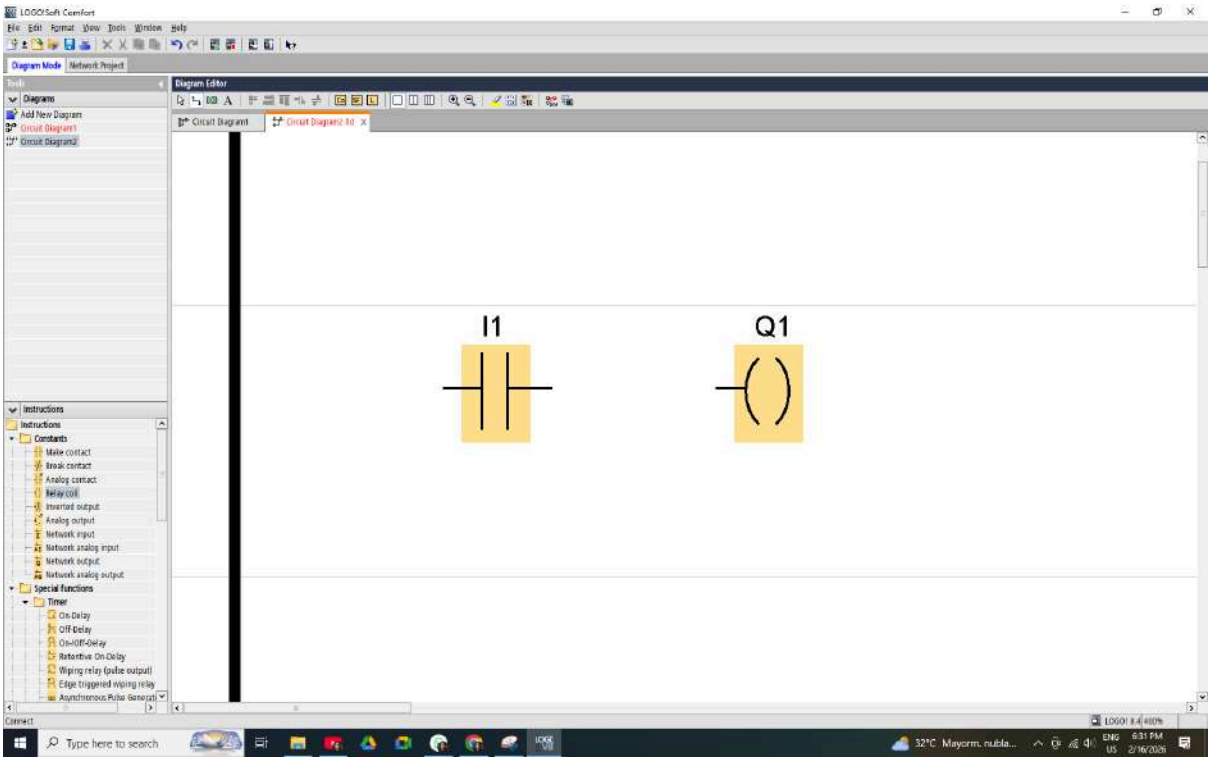


Si recorren la ventana de instrucciones encontrarán una gran variedad de elementos, diferentes de los vistos hasta ahora. Entre los primeros y más básicos se encuentran los contactos normal abierto y normal cerrado (afirmativo y negado lógico).

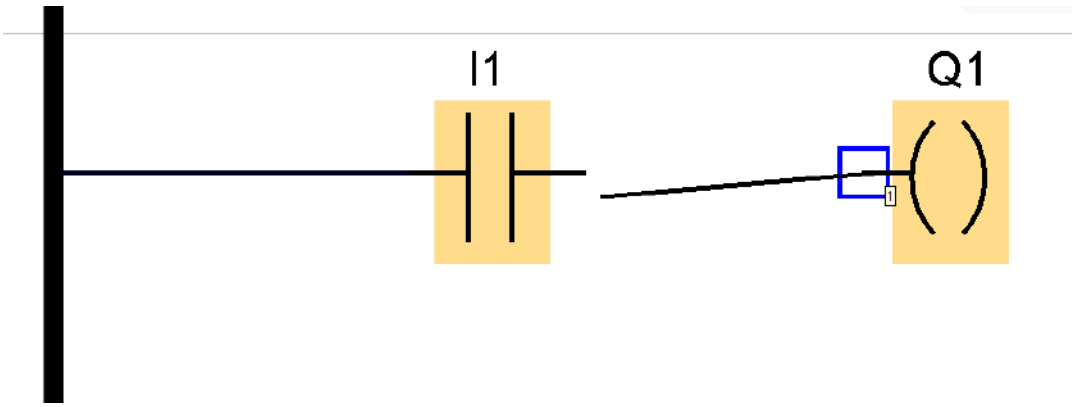
Para hacer uso de estos es necesario arrastrarlos hacia la superficie de trabajo.



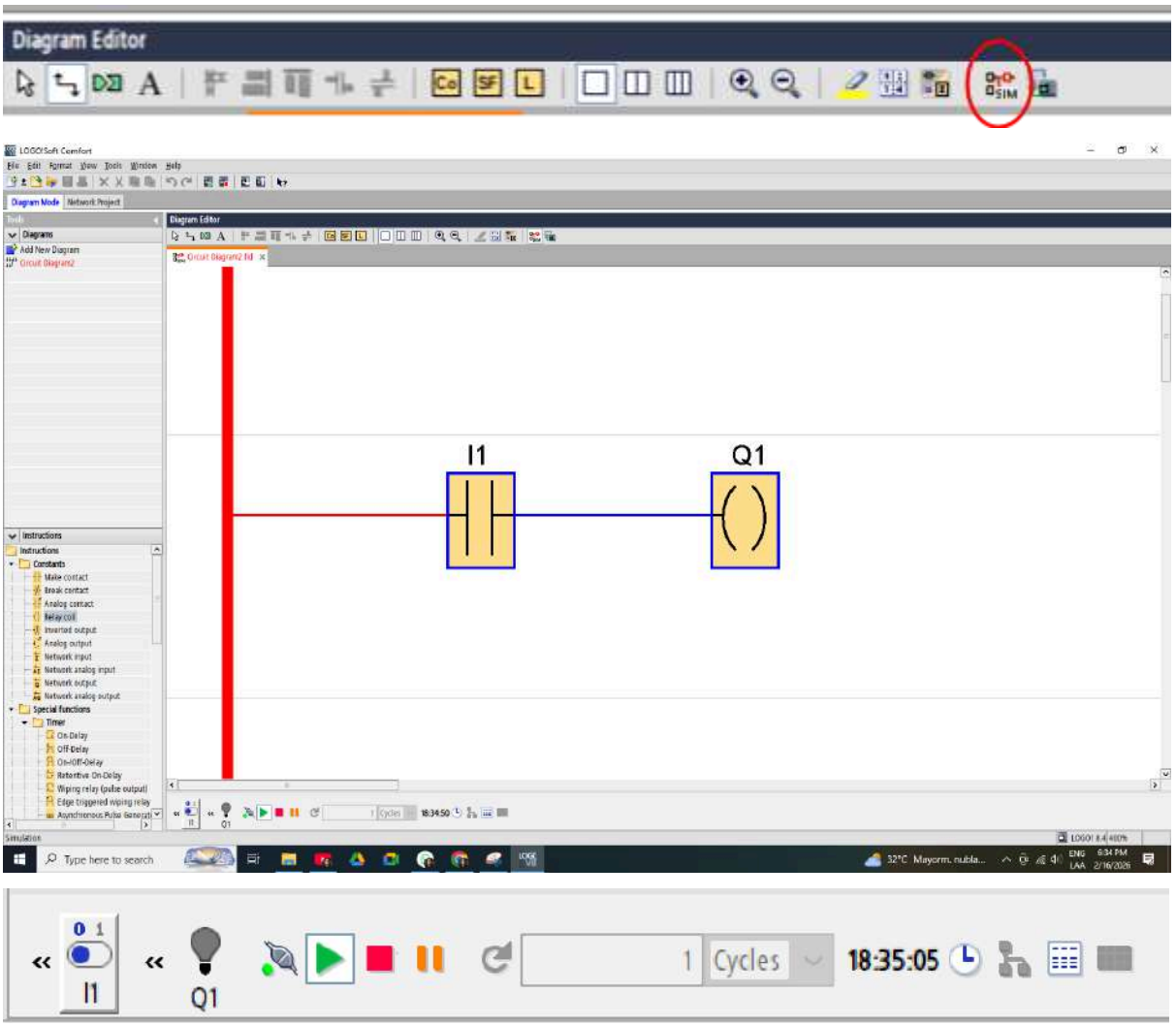
Un LOGO! 8.3 suele tener 8 entradas digitales. Numeradas I1, I2, I3,…, I8. Y unas 4 salidas numeradas como Q1, Q2, Q3 y Q4.



Con la herramienta de conexión podemos asociar los elementos de nuestro primer programa.



Ahora que hemos construido nuestro programa, podemos simular para corroborar su funcionamiento. Para eso usar el botón de simulación.

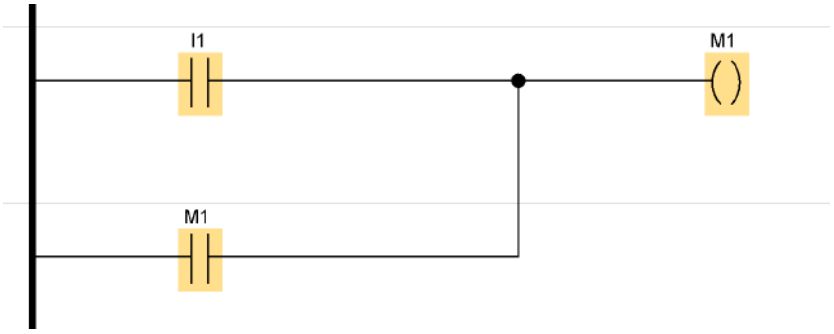


Al presionar el interruptor de la simulación con nombre I1, podremos observar cómo se enciende Q1.

Hemos visto los algoritmos esenciales, que constituyen los demás: NEGACIÓN, SUMA y MULTIPLICACIÓN. Por ejemplo el algoritmo de **enclavamiento**.

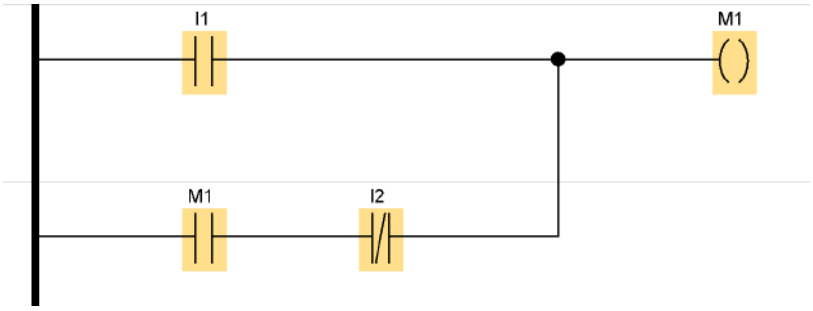
Este consiste en retroalimentar el dispositivo utilizando información de una salida. Una salida especial, fantasma. Conocida como **Marca**. Una salida que no tiene impacto sobre la física exterior eléctrica, sino que ocurre en el interior del PLC (o LOGO! en este caso).

Todo autómata programable posee un número finito de marcas, al igual que otros elementos que veremos en el futuro.

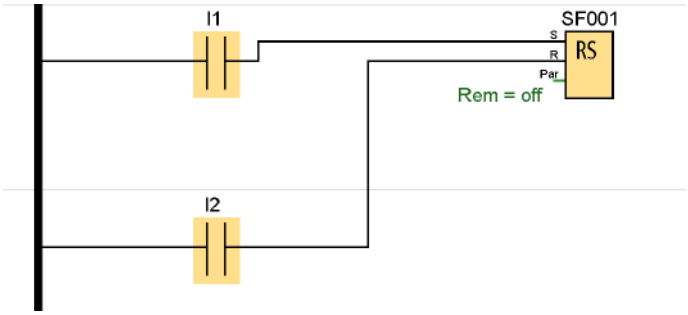


En este caso, lo que observaremos es que la marca será energizada cuando I1 sea oprimido. Luego, M1 se alimentará a sí mismo, independientemente del estado de I1. Esto generará que M1 quede encendida de manera permanente.

Para prevenir que sea de forma permanente, y que se pueda resetear el sistema, es posible introducir un segundo contacto, esta vez normal cerrado, para que solo cuando sea presionado, la retroalimentación de M1 se interrumpa.

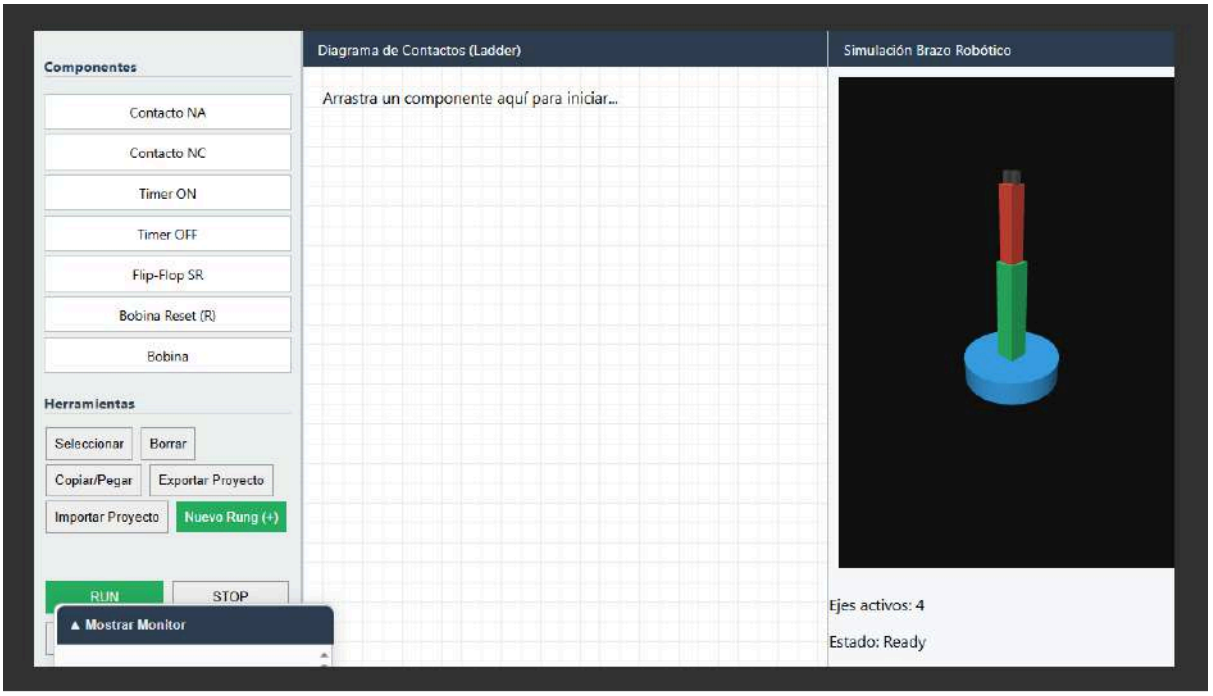


Como el funcionamiento de Marcas con el fin de establecer enclavamientos es común, se suele automatizar con partes más condensadas, en una cantidad finita nuevamente, como los bloques o bobinas SET-RESET. Lo que hacen es: al presionar I1, se enclavan; y luego al presionar I2 (normal abierto), se resetean.



Para practicar los conceptos trabajados hasta aquí les invito a moverse a una plataforma ligeramente distinta. Debido a las dificultades que puede presentar el LOGO! Soft Comfort en visualizar lo que estamos logrando, y simultáneamente abrir la discusión acerca de que el estándar de LADDER cambia según el dispositivo. Les invito a utilizar la siguiente plataforma.

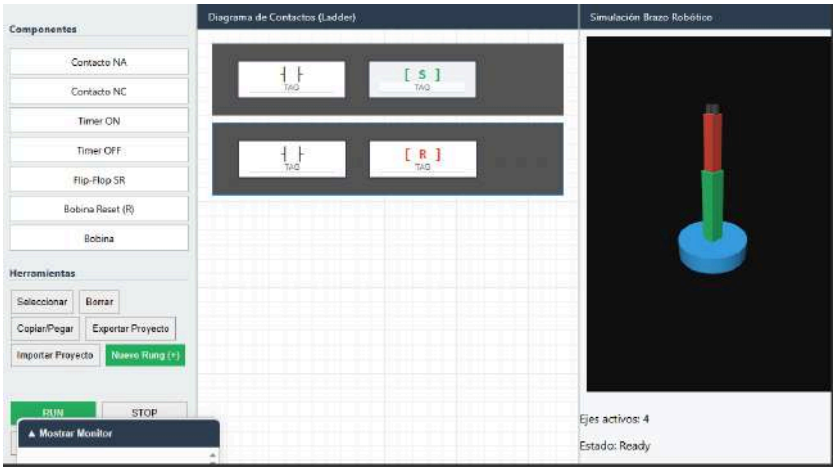
https://sites.google.com/view/lardetjuanpedro/simulador-de-plc



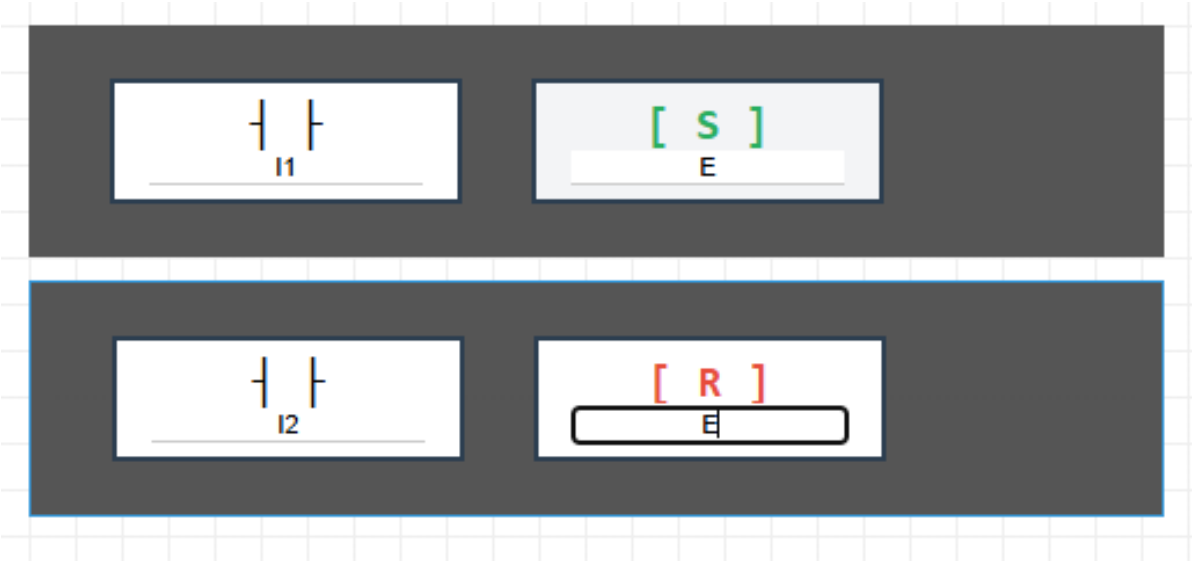
Aquí dejaremos de utilizar lámparas para pasar a controlar dispositivos más parecidos a los reales. Un brazo robótico de cuatro ejes de libertad. Cada eje está dominado por dos motores numerados M1_F (hacia adelante) y M1_B (hacia atrás), con su par de fines de carrera FC1_F y FC1_B.

Hagamos un enclavamiento con el Set (del tipo bobina, seleccionamos el Flip-Flop SR) para inicializar el programa. Para ello usamos un I1 y un I2 con el fin de encender cada bobina (set y reset). A la bobina hay que darle el mismo nombre para que funcione.

Para colocar más líneas donde colocar elementos, simplemente se aprieta el botón “nuevo rung”.



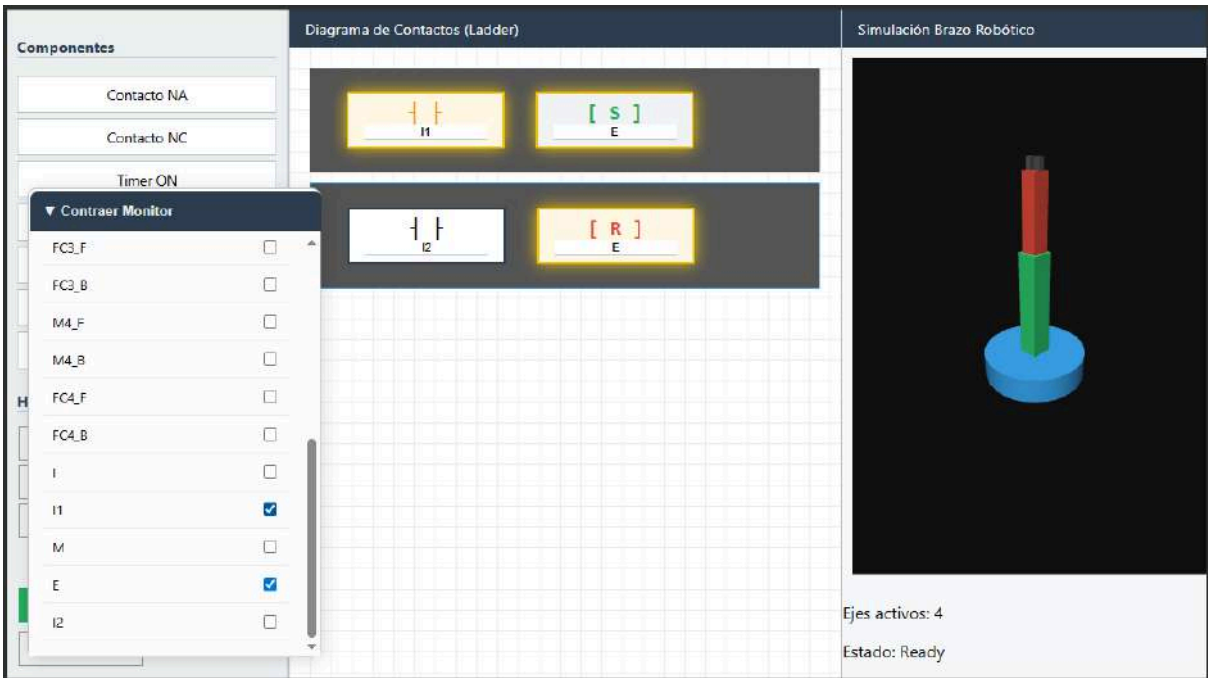
En muchos softwares de programación de PLC, como en el simulador ofrecido, veremos que existen mapas o menúes tipo panel para la determinación de los tipos de variables. Variables es el nombre que reciben los nombres de entradas y salidas, memorias y set-reset, que en este caso son del tipo booleano porque almacenan datos binarios. En el futuro esto se verá complejizado.



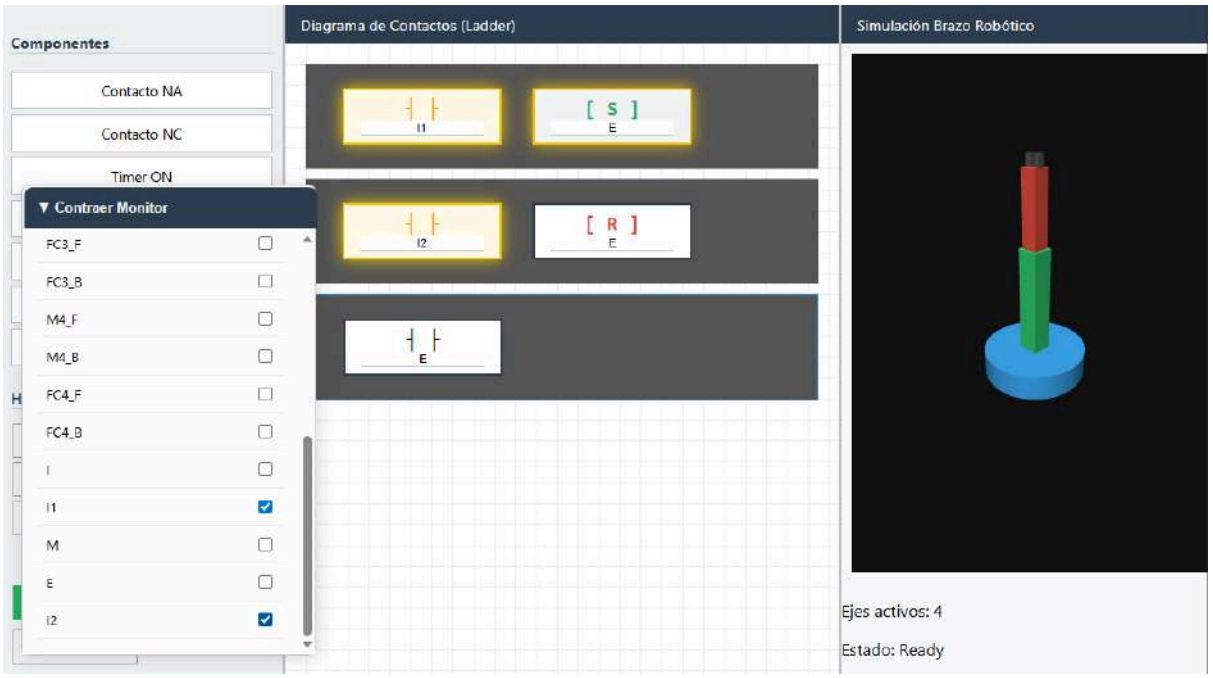
En la ventana desplegable con el nombre Monitor, podemos activar o desactivar al presionar sobre los bloques el estado lógico de cada variable.

Similar a lo que hicimos en LOGO! Soft Comfort.

Para esto, debemos darle a la opción play o simular dentro del entorno, para empezar a corroborar el funcionamiento.



Algo que este simulador nos permite observar es qué sucede si I1 e I2 están simultáneamente encendidas (ON, 1 lógico). Para ello, podemos nombrar un contacto normal abierto con la variable E, e identificar lo sucedido.

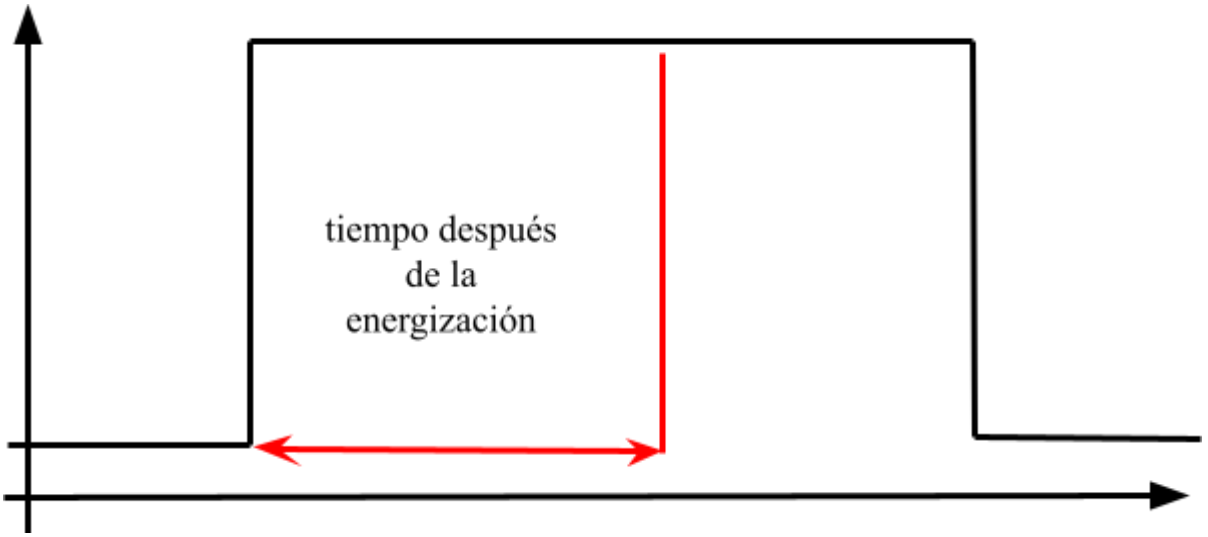


No brilla, por lo tanto, no hay efecto de I1 sobre el sistema a continuación de E.

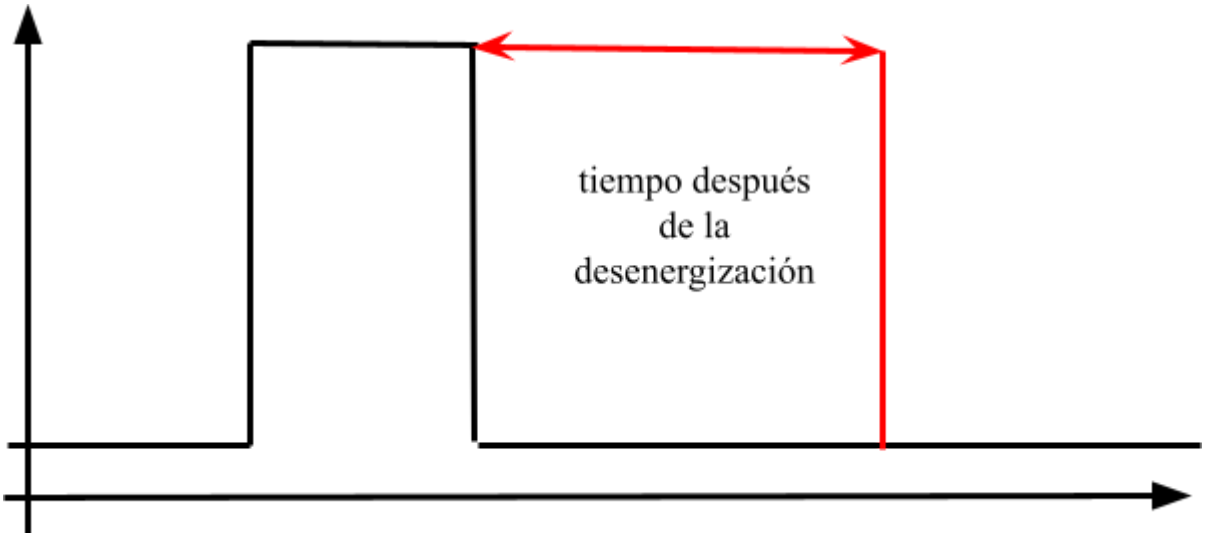
Ahora usemos los temporizadores para lograr movimientos a partir de E, que nos permitan secuenciar una coreografía del robot. Para eso, presentamos dos tipos no discutidos previamente de temporizadores, los TON y TOF.

Para el estudio de estos elementos, es común la introducción a diagramas de señales, porque permiten al estudiante una mayor autonomía al identificar el funcionamiento de cada bloque funcional o de contacto.

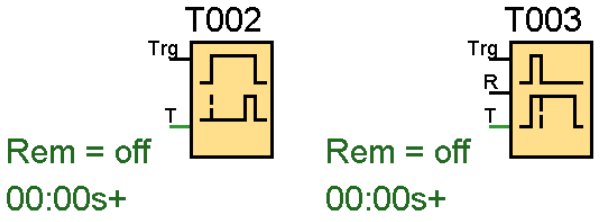
TON o timer on delay, es un temporizador del tipo monoestable, que se enciende tiempo después de que fué energizado. Hasta que se desenergice.



TOF o timer off delay, es un temporizador del tipo monoestable, que se apaga tiempo después de que se cortó el suministro energético.



Existen otros tipos, pero en este simulador contamos con estos clásicos. Mientras que LOGO! Soft Comfort posee algunas más: A continuación se adjuntan ejemplos de cómo se ven los del LOGO.



Este ejemplo lo que hace es resetear luego de 3 segundos, para que el robot gire en una dirección durante ese tiempo y luego frene.

Ejercitación

Ejercicio 1 — Identificación
Indicar si los siguientes sistemas son manuales o automáticos y justificar:
a) Interruptor de luz
b) Flotante de tanque
c) Puerta automática de supermercado
d) Semáforo

Ejercicio 2 — Clasificación
Clasificar cada sistema según tipo de automatismo:

- temporizado
- comparador
- manual

Sistemas:

- temporizador de horno
- termostato
- botón de arranque motor

Ejercicio 3 — Verdadero o falso
Justificar los falsos:

1. Un PLC ejecuta todas sus instrucciones simultáneamente.
2. Un contacto normalmente cerrado conduce corriente cuando no se presiona.
3. Un bit puede almacenar múltiples valores.

Ejercicio 4 — Tabla lógica
Completar la tabla de verdad para dos pulsadores en:
a) serie
b) paralelo

Ejercicio 5 — Traducción lógica
Dibujar el circuito eléctrico equivalente para:
Q1 = I1 AND I2
Q2 = I1 OR I3

Ejercicio 6 — Interpretación
Observá el circuito:

- I1 en serie con I2
- ambos en paralelo con I3

Preguntas:

1. ¿Cuándo se enciende la salida?
2. Escribir la ecuación lógica.

Ejercicio 7 — Primer programa

Diseñar un diagrama Ladder que:

- encienda Q1 al presionar I1
- se apague al soltarlo

Ejercicio 8 — OR lógico

Diseñar Ladder para:

Encender una lámpara si se presiona cualquiera de dos pulsadores.

Ejercicio 9 — AND lógico

Diseñar Ladder para:

Encender un motor solo si se presionan simultáneamente dos pulsadores de seguridad.

Ejercicio 10 — Auto-mantenimiento

Diseñar un circuito Ladder donde:

- I1 arranca motor
- I2 lo detiene
- el motor queda encendido aunque se suelte I1

Ejercicio 11 — Marca interna

Realizar un sistema que:

- active M1 al presionar I1
- apague M1 al presionar I2
- Q1 copie el estado de M1

Ejercicio 12 — Análisis

Responder:

¿Por qué es peligroso usar enclavamientos sin botón de parada?

Ejercicio 13 — TON

Diseñar programa donde:

- al presionar I1
- Q1 se encienda después de 3 segundos

Ejercicio 14 — TOF

Diseñar programa donde:

- Q1 se encienda inmediatamente
- se apague 5 segundos después de soltar I1
-

Ejercicio 15 — Secuencia

Diseñar sistema que:

1. encienda Q1
2. luego de 2 s encienda Q2
3. luego de 2 s apague Q1
4. luego de 2 s apague Q2

Ejercicio 16 — Sistema de tanque

Diseñar lógica para:

- bomba llena tanque si nivel bajo activo
- se detiene si nivel alto activo

Condición de seguridad:

Si ambos sensores fallan → bomba apagada.

Ejercicio 17 — Motor reversible

Motor con:

- avance (Q1)
- retroceso (Q2)

Reglas:

- nunca pueden activarse juntos
- botón avance: I1
- botón retroceso: I2
- paro general: I3

Ejercicio 18 — Diagnóstico

Un sistema presenta falla:

Al presionar start el motor arranca pero nunca se detiene.

Posibles causas lógicas:

- error en reset
- contacto mal tipo
- realimentación incorrecta

Explicar cada posibilidad.

🧠 EJERCICIO FINAL INTEGRADOR

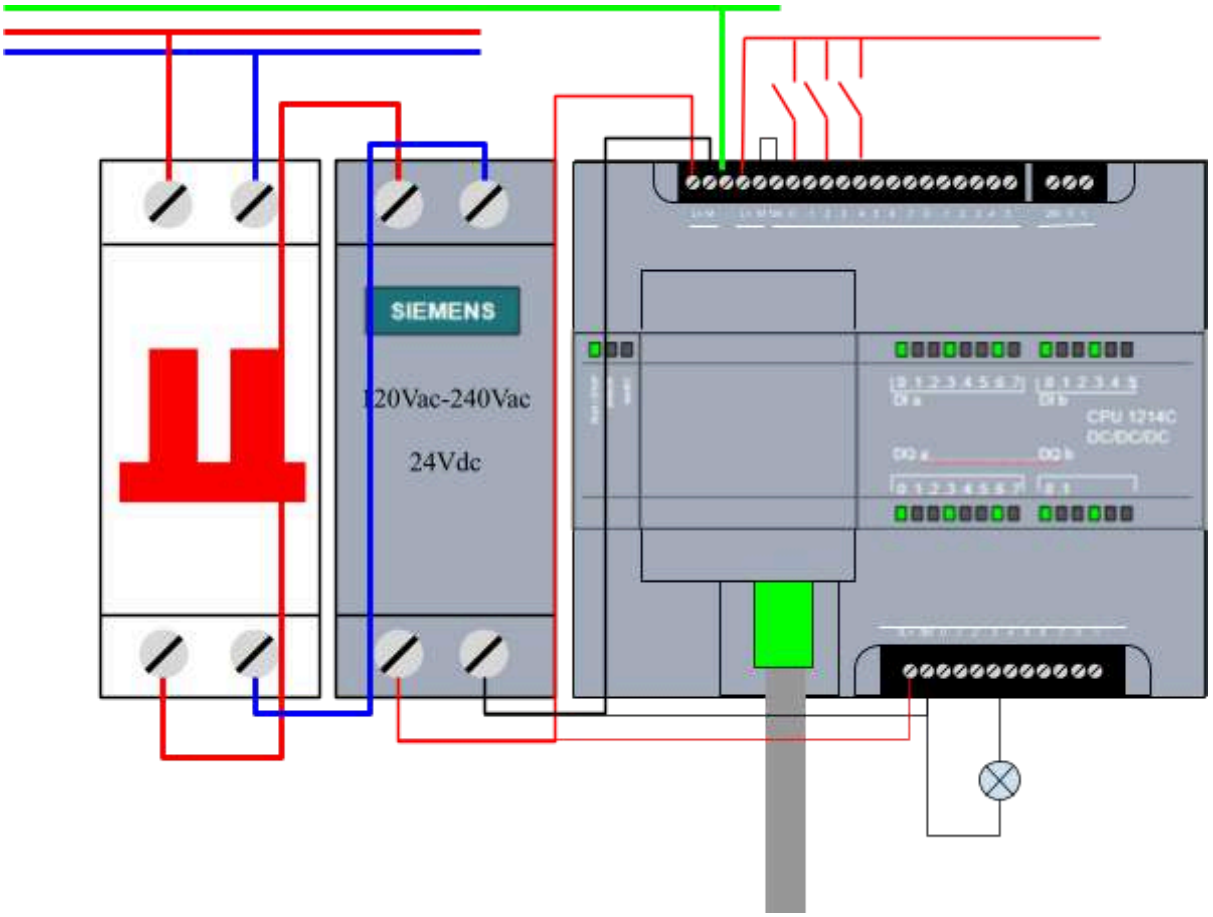
Diseñar un sistema completo para un brazo robótico que:

1. se active con START
2. ejecute secuencia de 3 movimientos
3. tenga parada de emergencia
4. tenga luz indicadora de sistema activo
5. vuelva a estado inicial al finalizar

CADe SIMU

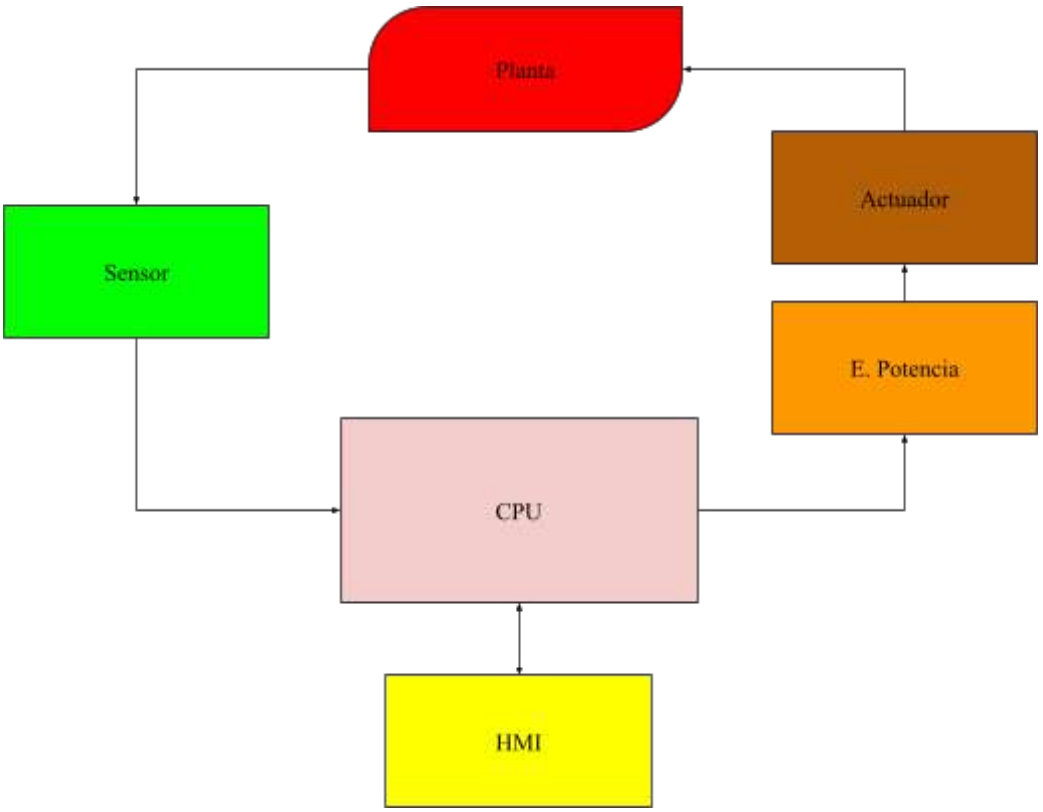
Ya nos hemos desafiado considerablemente. Hasta esta altura podríamos decir que conocemos cantidades considerables de conceptos y herramientas. Más aún, si hemos podido resolver hasta cierto punto el ejercicio integrador anterior poseemos buenas herramientas para profundizar en la práctica de la simulación eléctrica. Algo que hasta ahora quedaba medio flojo, el "¿Cómo vinculo mi programa con el entorno eléctrico?".

Retomemos para eso el diagrama eléctrico anterior.



Este sistema conecta un PLC del tipo S7-1200 de SIEMENS. Existen diversos estilos, y debemos ser cuidadosos de cada uno. Por ejemplo.

- Un CPU que dice S7-1200 1214c DC/DC/DC, es un PLC de la serie o familia 1200, diferente de la familia 1500 u otras. Se alimenta por un voltaje continuo de 24 voltios (la primer cifra del DC/DC/DC nos indica eso), tiene entradas de continua de hasta posiblemente 15 voltios (nos lo indica la segunda cifra) y finalmente tiene salidas transistorizadas de igual voltaje anterior, salida máxima de corriente de 0.2A cada una.
- Un CPU de la misma familia del tipo 1214c AC/DC/AC, es un PLC de la familia anterior, pero que se alimenta con 120 a 240 voltios de alterna (monofásico), tiene entradas digitales de continua y salida de alterna. Es recomendable para estos casos, de igual manera, aunque permita salida de alterna, utilizar un elemento de potencia intermedio de la salida hacia un actuador de potencia.

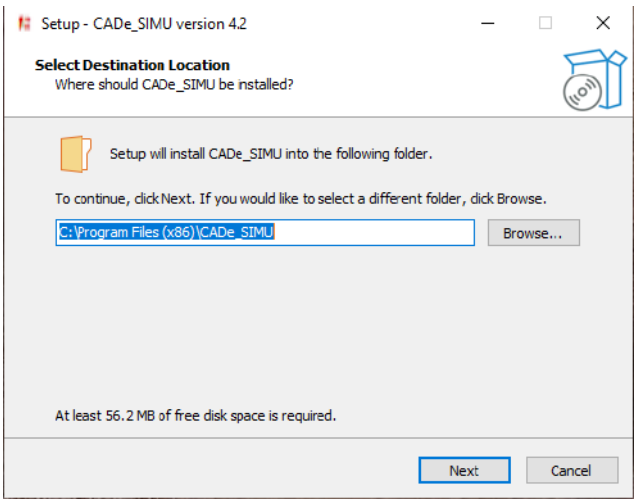


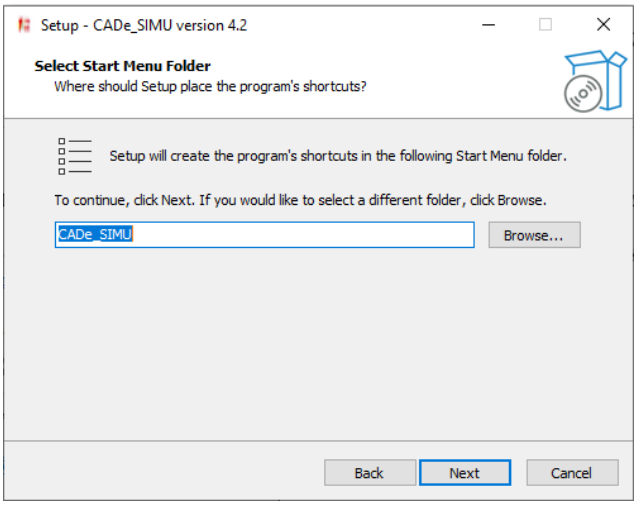
Este gráfico representa la ubicación relativa de cada elemento de un sistema de automatismo industrial, en relación con los demás. Vemos que se identifica el hecho de que la salida, ya sea de Transistor, AC (TRIAC) o Relay (Rly), debe ser conectada a una etapa de potencia a continuación. De manera tal que la E. de Potencia conduzca la energía adecuadamente hacia el actuador.

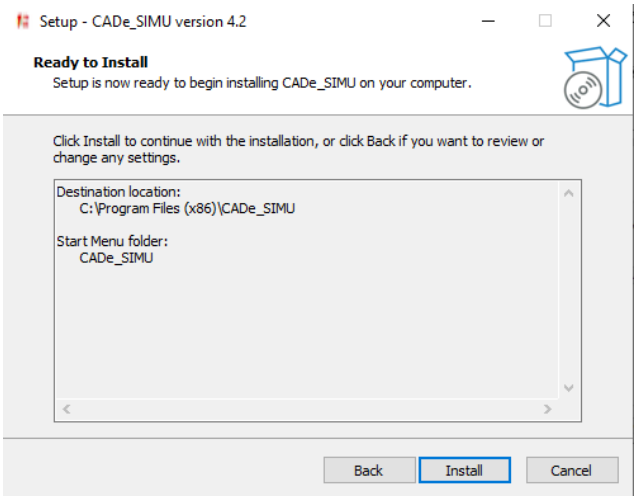
Es necesario agregar a la mayoría de los automatismos protecciones adecuadas para cada escenario. Por ejemplo, un caso, es la protección galvánica (transformador), que previene del ruido eléctrico considerablemente. Se pueden agregar diversas protecciones de sobrecorriente como fusibles y termomagnéticas. Rara vez veremos en estos dispositivos la necesidad de una protección diferencial.

Lo más seguro es la utilización de un PLC del tipo DC/DC/(cualquiera), de tal manera que se alimente con 24Vdc provistos por una fuente transformadora y reguladora, con filtros y protecciones internas. Preferiblemente de la misma empresa.

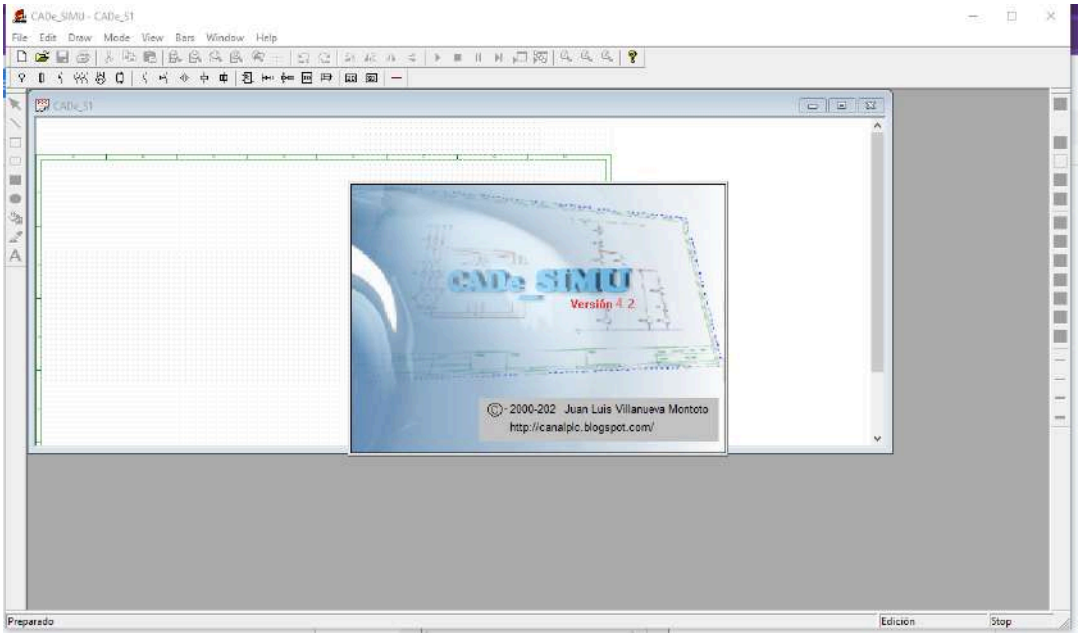
Veamos cómo es posible integrar estos conocimientos en un entorno de simulación y cad didáctico: https://cade-simu.com/#cade-simu-v42

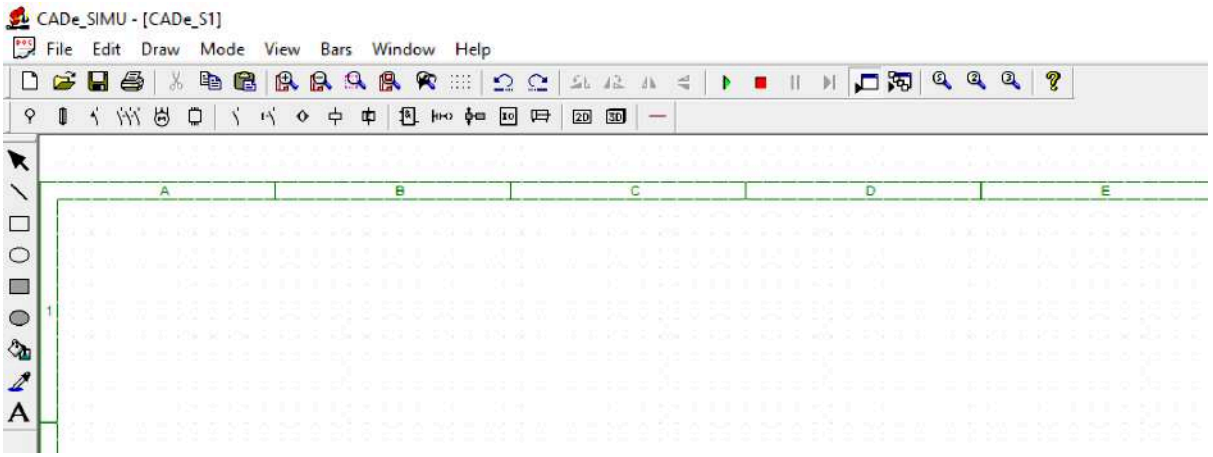




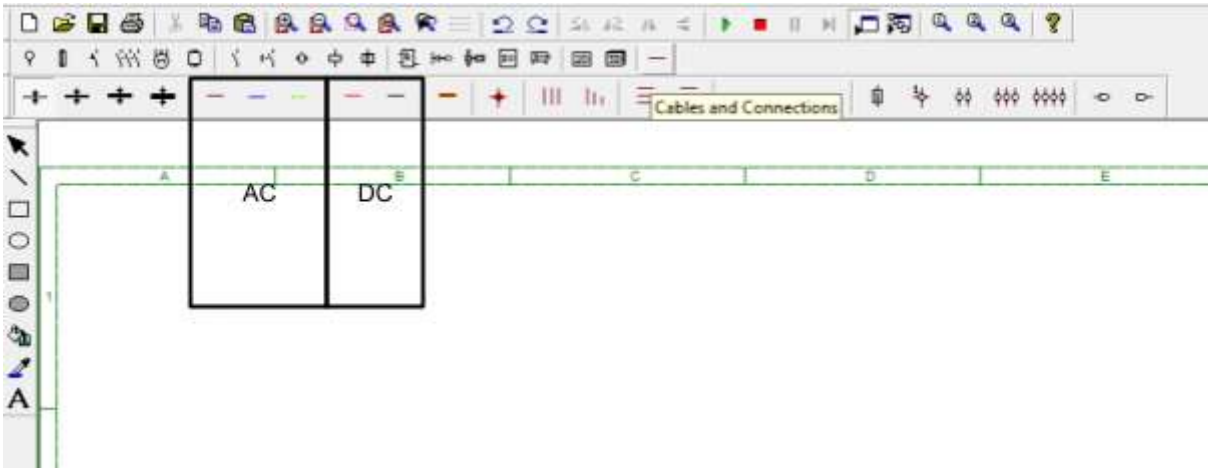


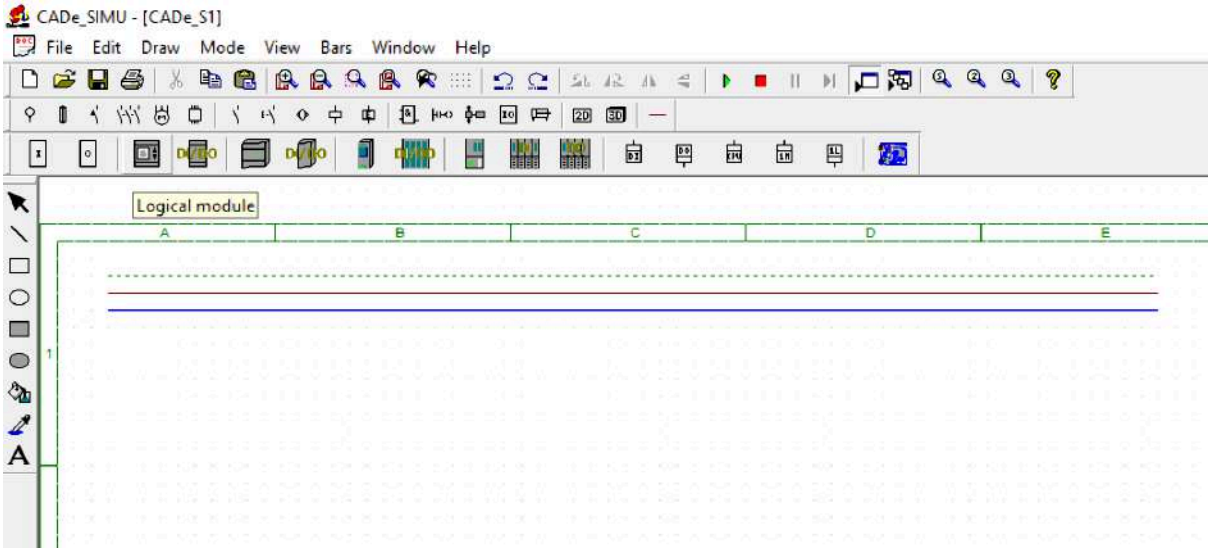
Al abrir el programa, nos solicitará la contraseña: 4962



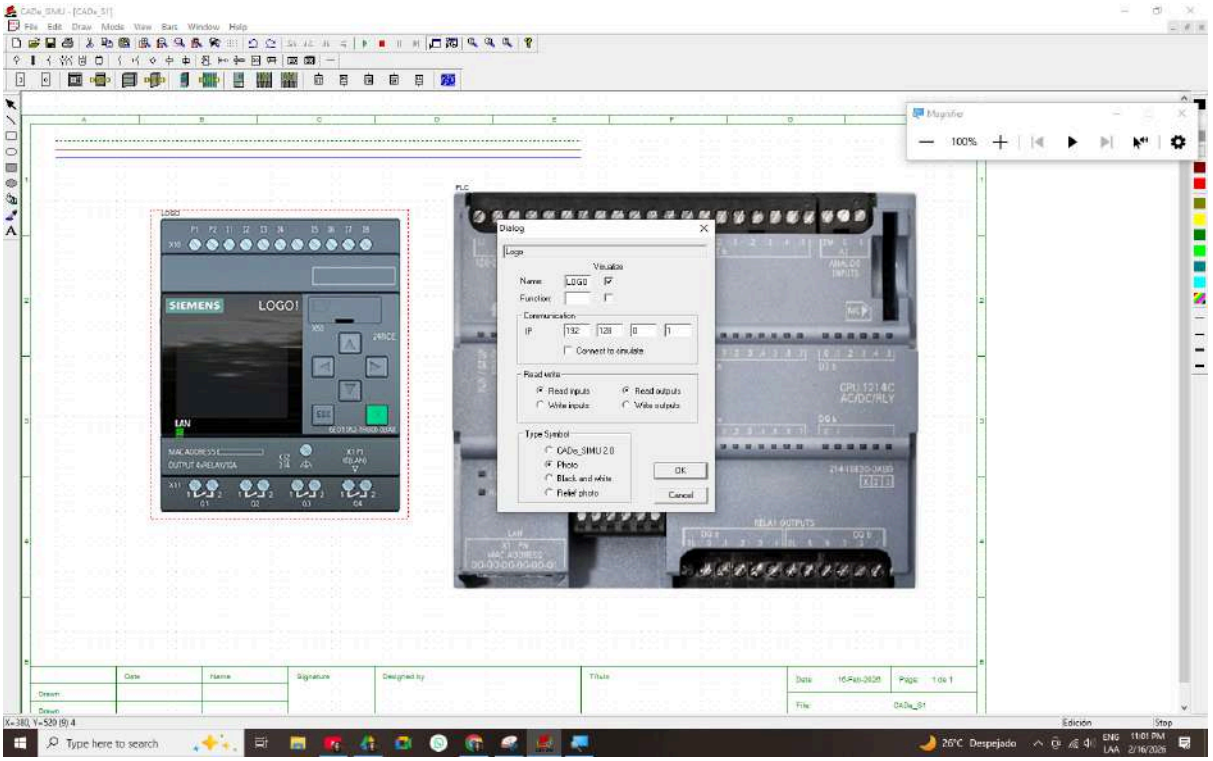


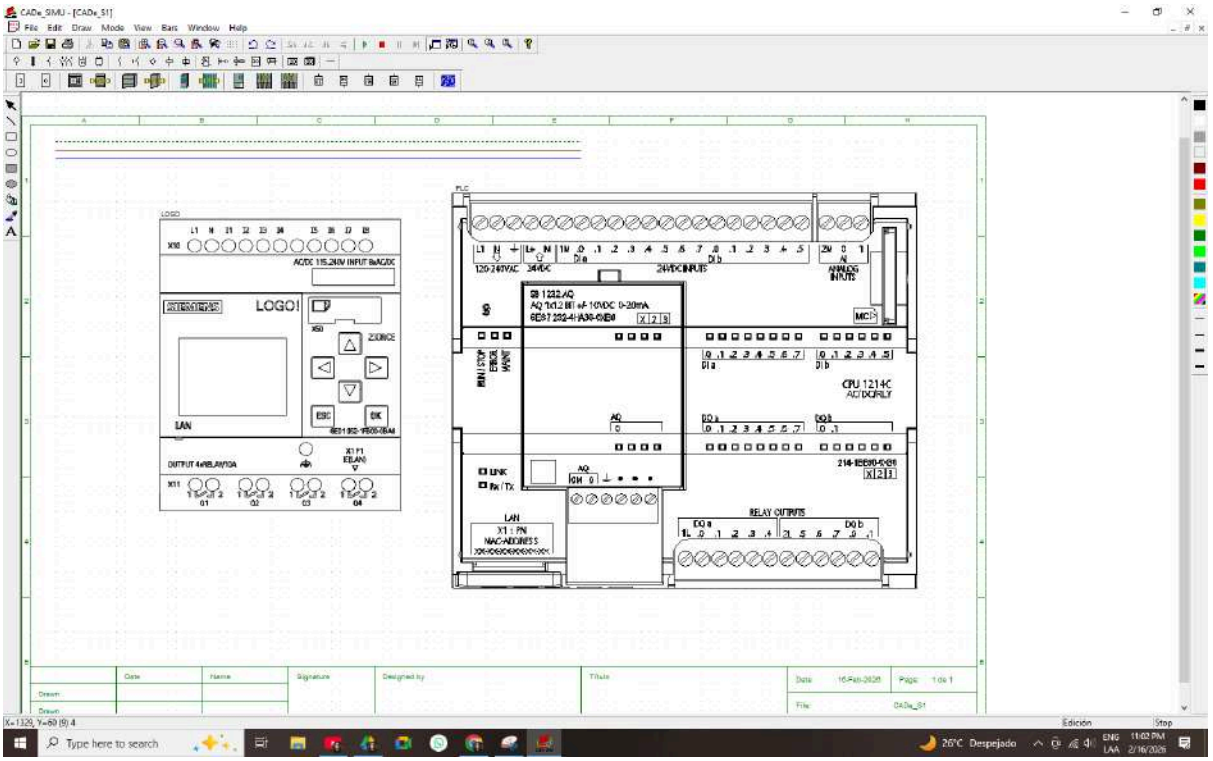
Explorando la interfaz encontrarán la opción de cables. Para este ejemplo, donde hablaremos de alimentación del PLC, usaremos cables de diverso tipo.



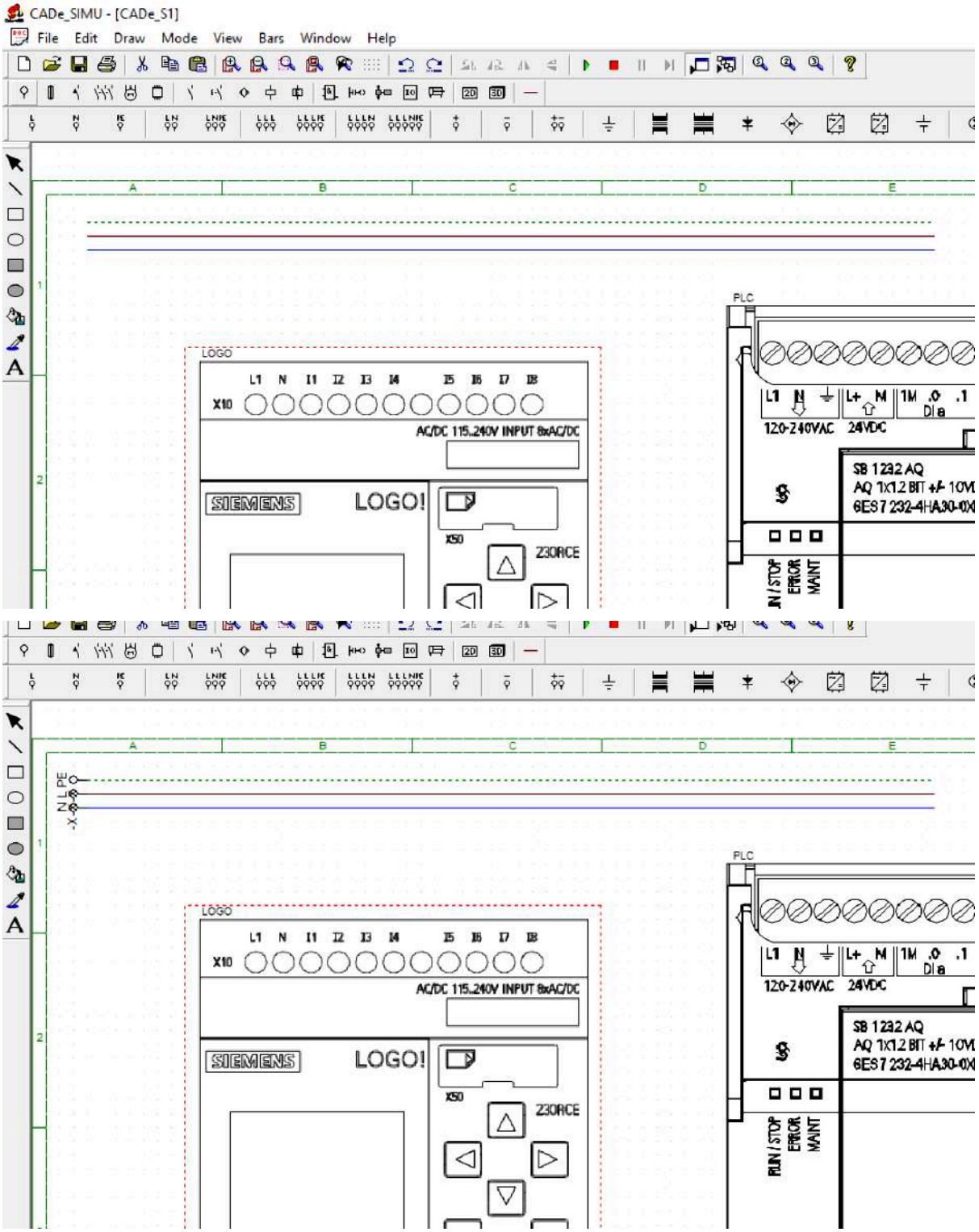


Agregamos (click sobre la opción y luego click sobre la pantalla para aplicar dicha selección de componente). Doble click sobre el controlador para editar su vista.





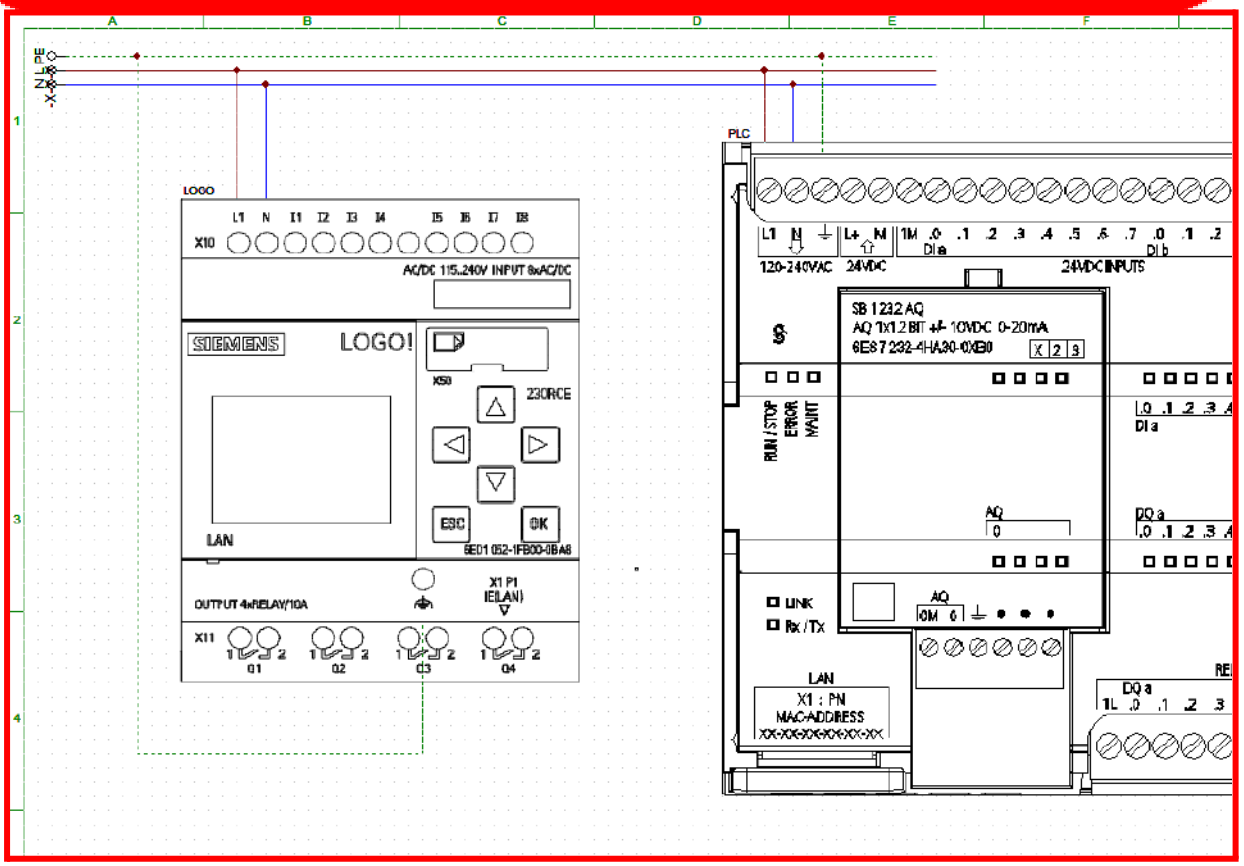
Con la primer opción podemos elegir la fuente deseada, y elementos relativos a la temática como transformadores (protección galvánica), fuentes AC/DC, etc.

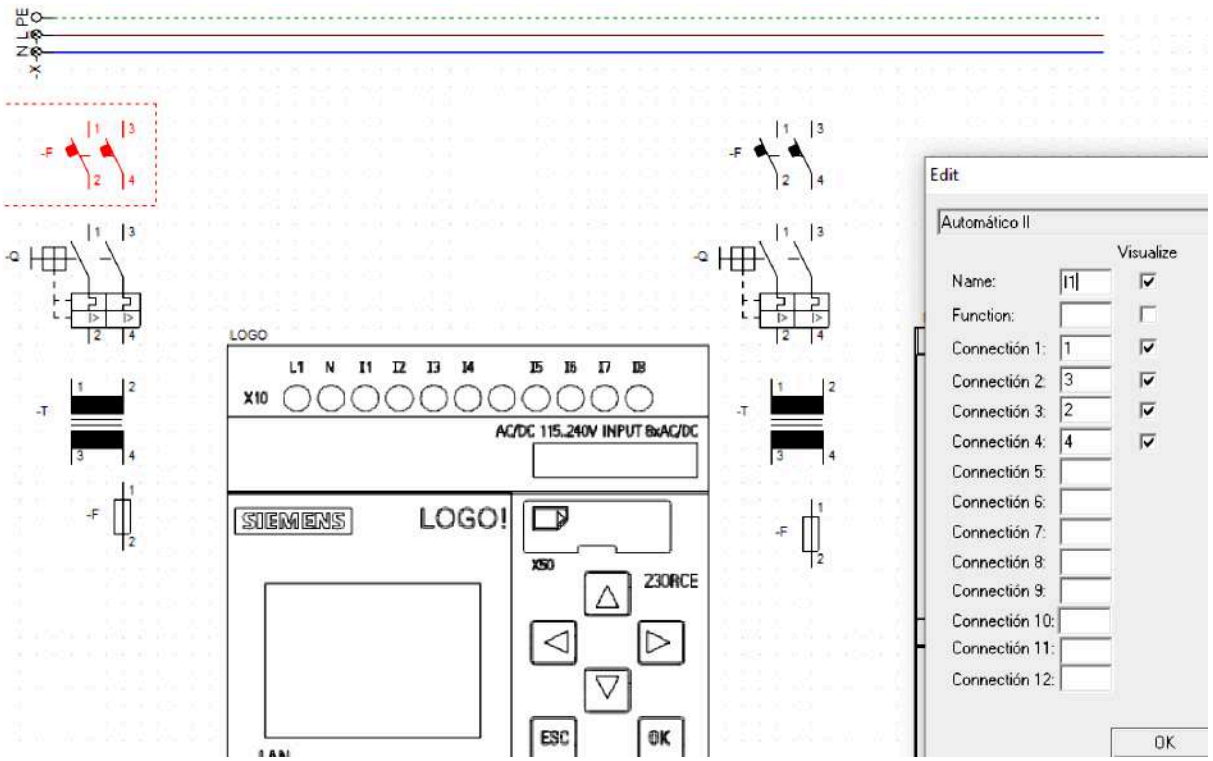


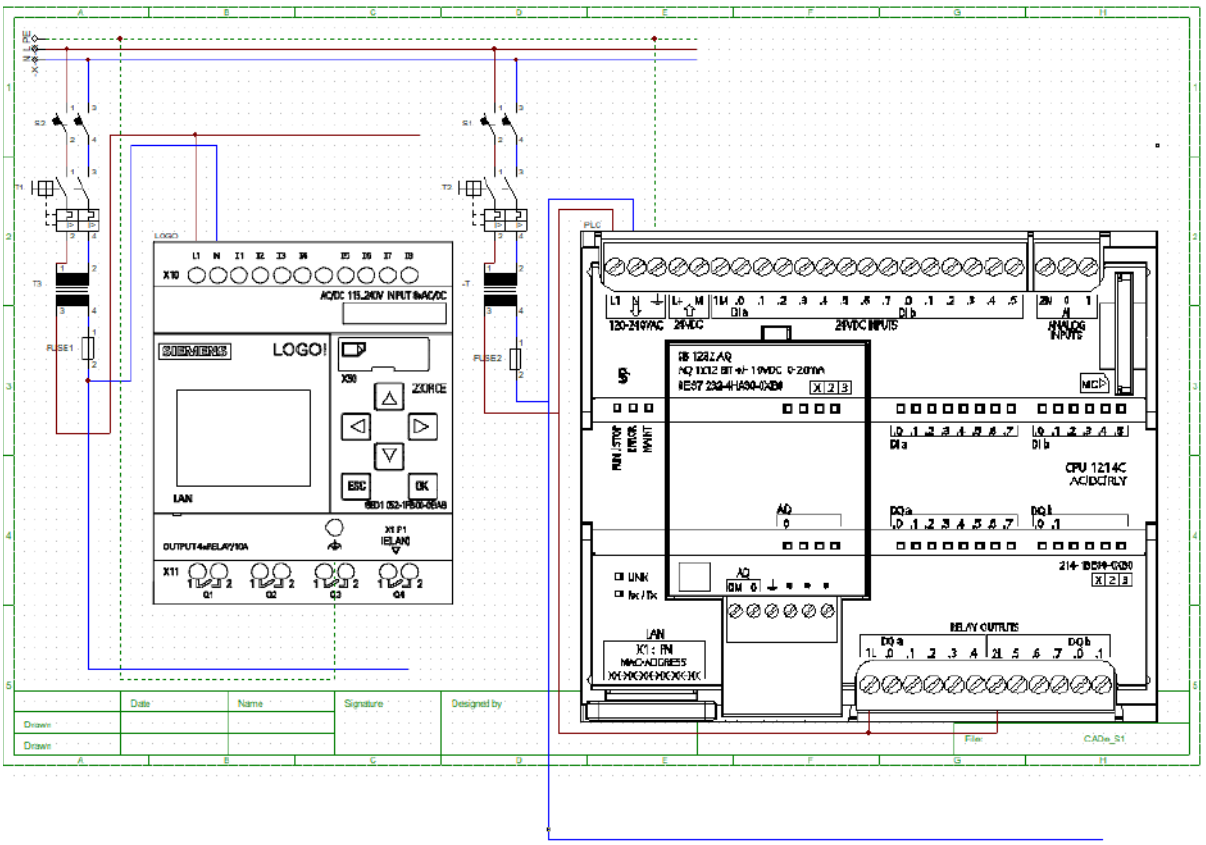
Veamos que el LOGO! tiene la posibilidad de alimentarse con 115 Vac a 240 Vac, incluídas sus entradas. Mientras que el PLC se alimenta con 120 Vac a 240 Vac, pero sus entradas son de 24 Vdc.

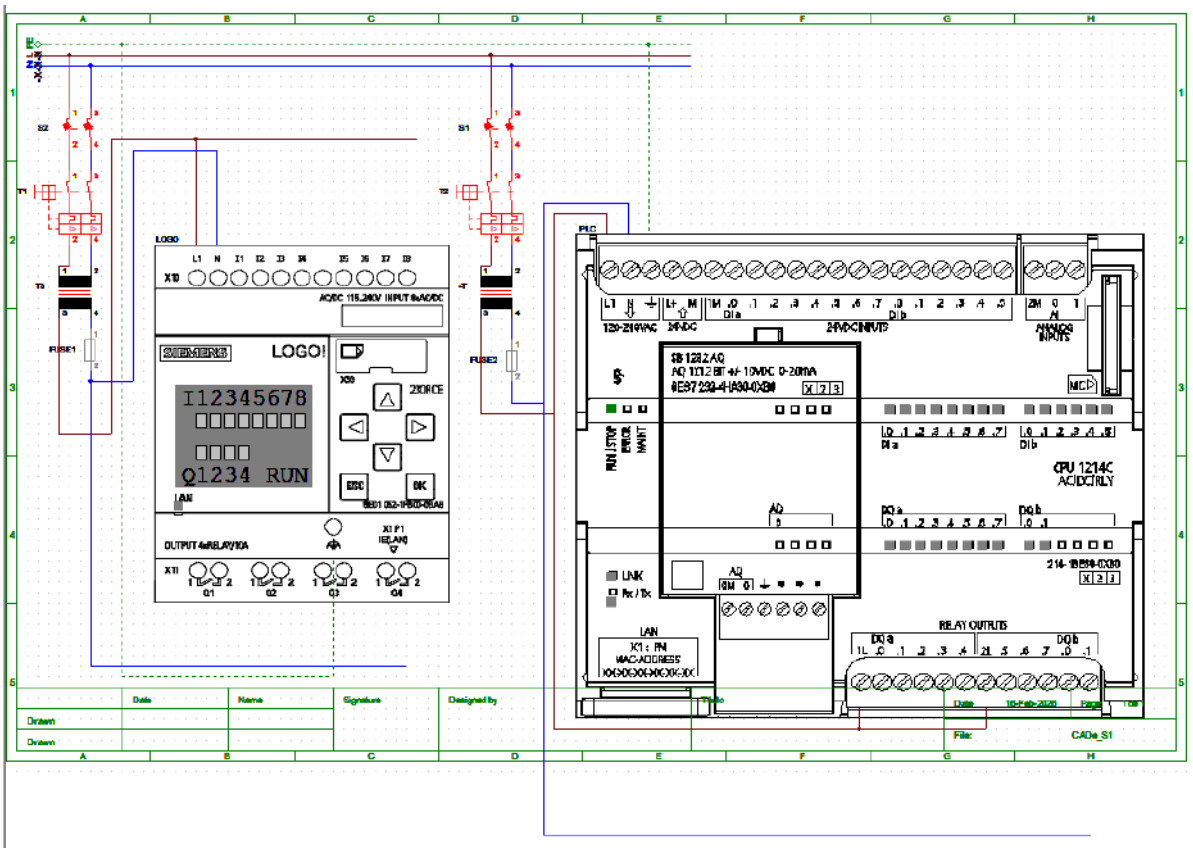
En ambos casos podemos conectar la puesta a tierra que actuará como salida para los campos electromagnéticos frenados por la caja faraday.

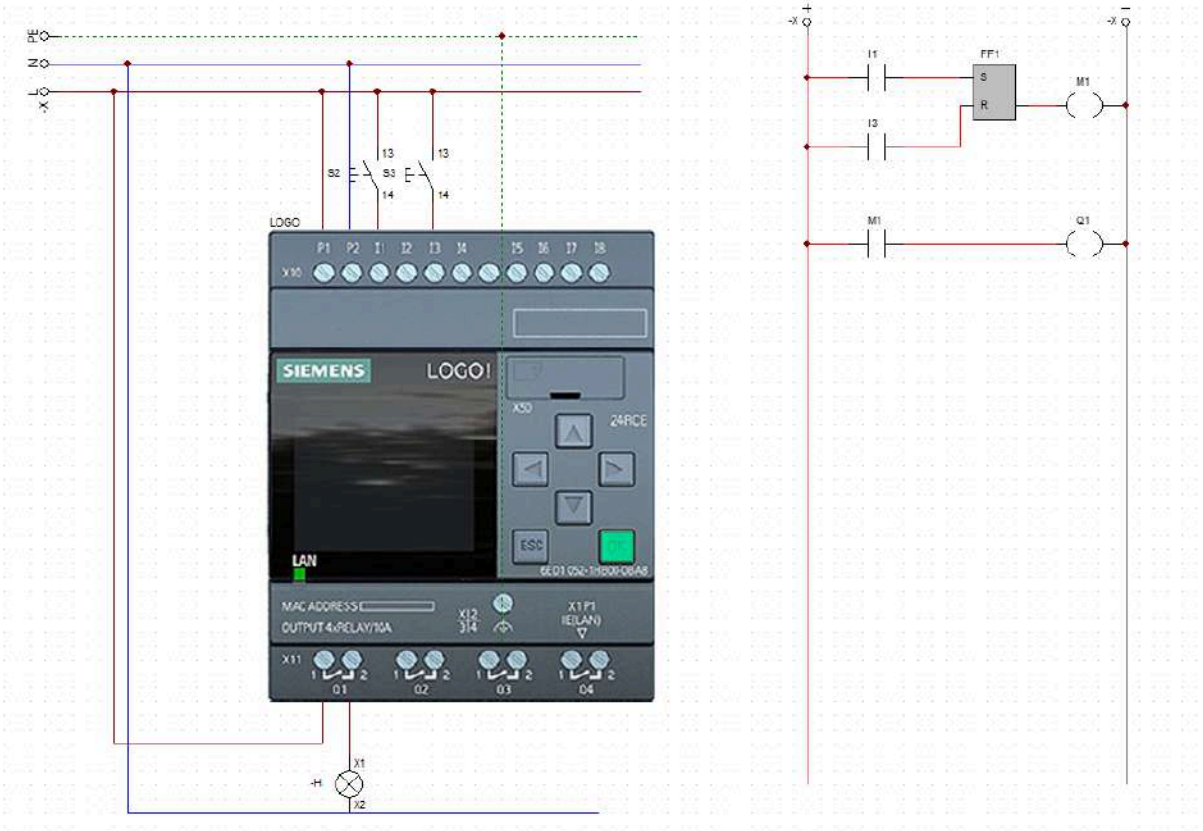
La conexión representada a continuación carece de interruptor, protección termomagnética y galvánica. Por lo tanto, es incorrecta.



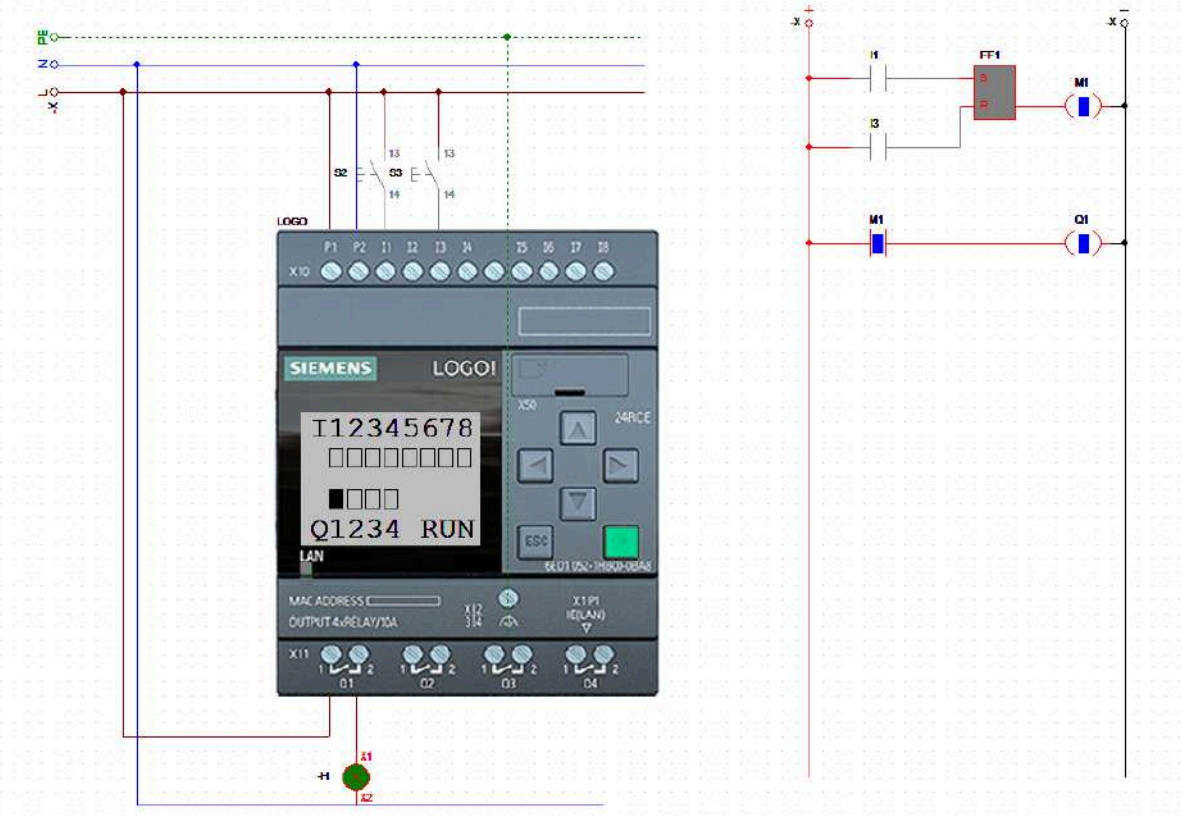


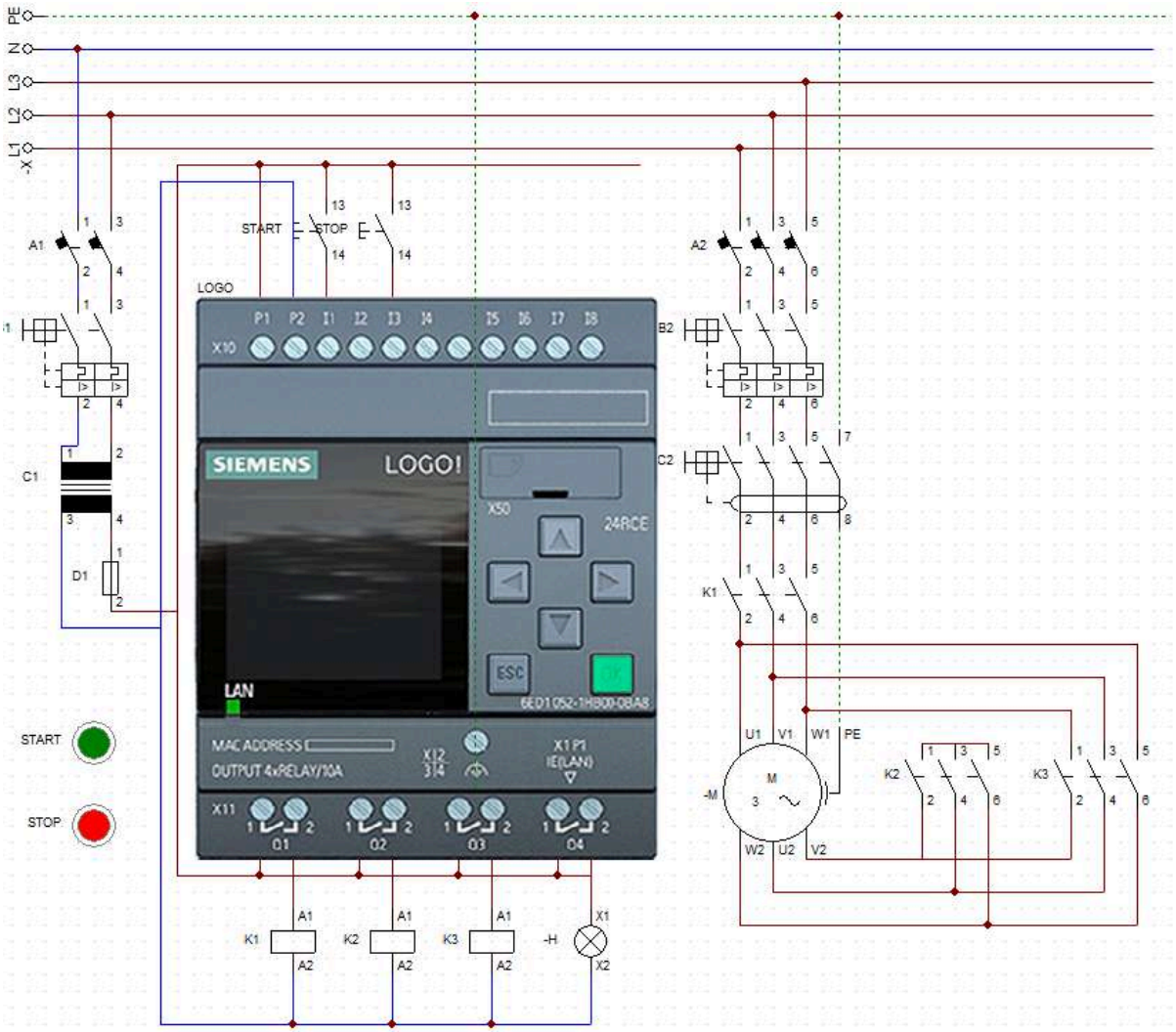

LOGO!
SIEMENS
CPU 1214C
AC/DC/RLY
RELAY OUTPUTS
CADe_S1


LOGO!
SIEMENS
I12345678
Q1234 RUN
CPU 1214C
AC/DC/RLY
RELAY OUTPUTS
CADe_S1



Luego de encender la simulación y pulsar S2.

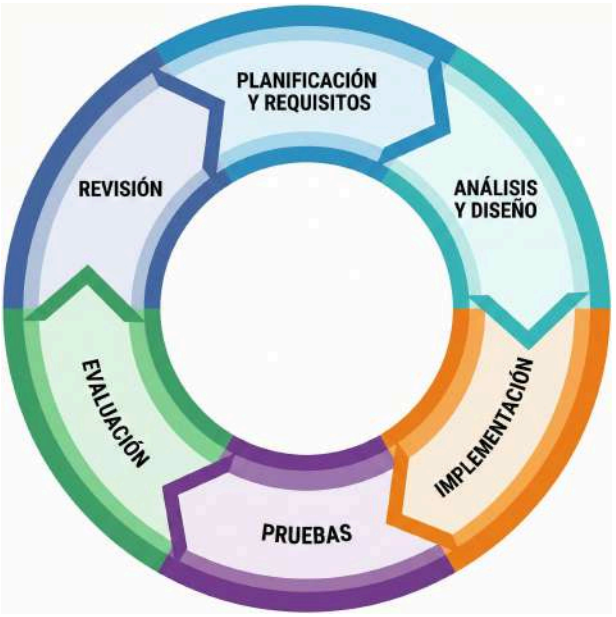




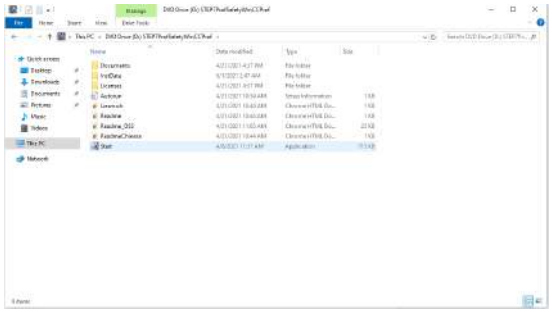
El ejercicio del arranque estrella-triángulo de un motor trifásico es un ejemplo clásico a simularse en CADe SIMU. Añadimos a continuación videos ejemplos del canal de youtube.

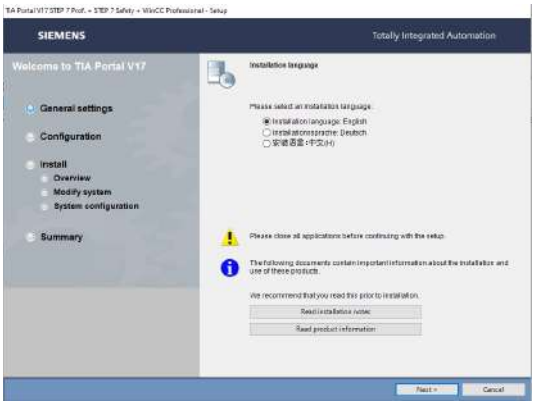
https://youtu.be/B7so3ZEL8Jo?si=Sy0TxQcQkRsYUERL
https://youtu.be/t8OwEEfukoI?si=S2eqIIM_EquKDYRh

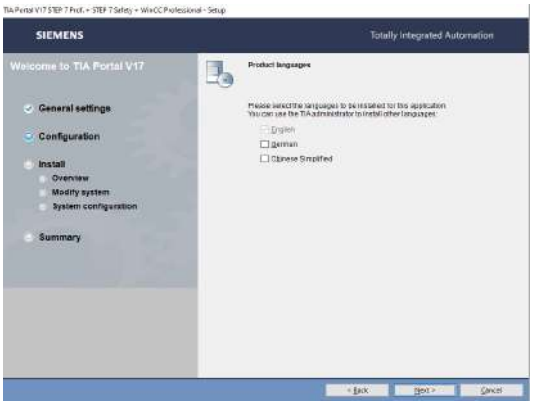
A partir de este momento es importante discutir la idea de posibles múltiples soluciones a cada problema o ejercicio. Incluso, el concepto de iteración es algo que es muy común: nunca se llega al escenario perfecto de programación e instalación. Ser capaces de adaptar un código o sistema, escalabilizarlo, proyectar a futuro nuevas mejoras o funcionalidades, es organizado.

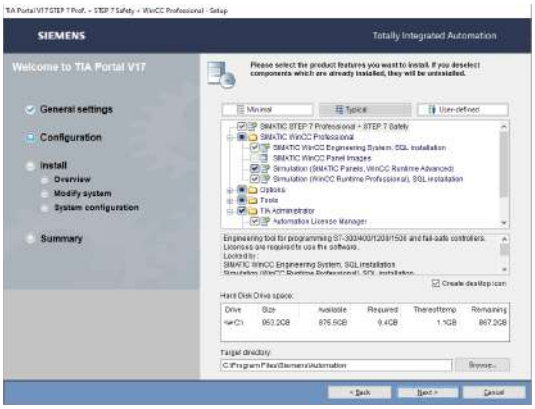


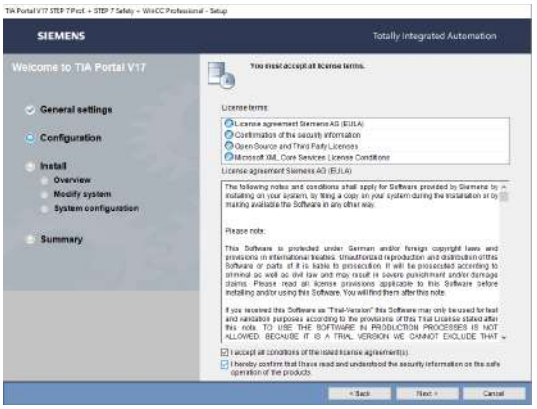
TIA Portal V17 Step 7

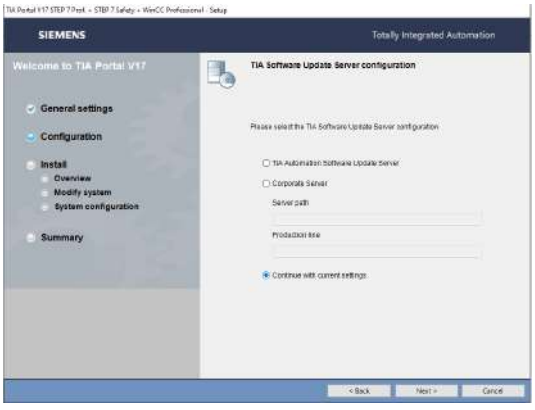


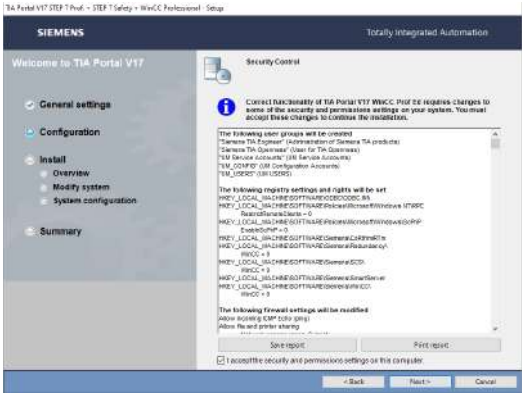


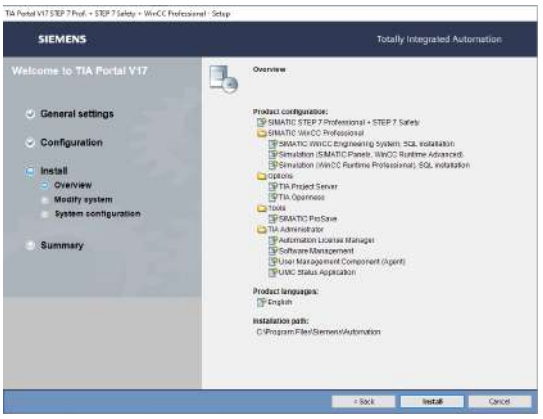




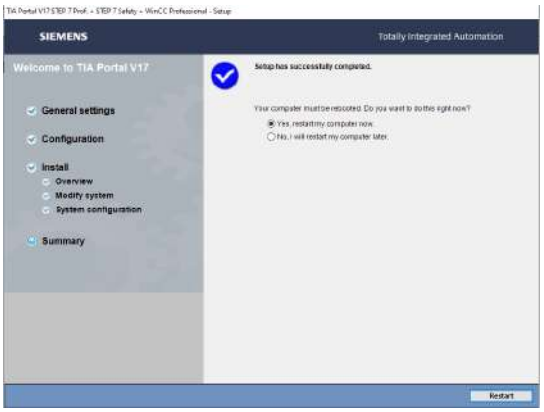


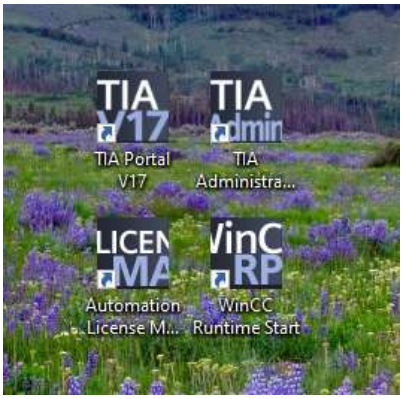




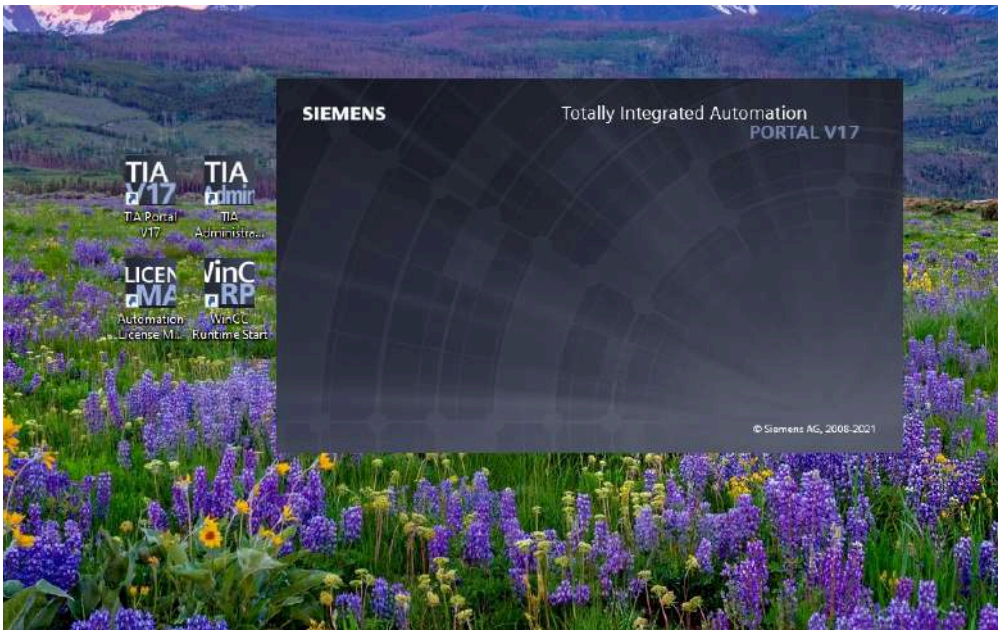

SIEMENS
Totally Integrated Automation
Welcome to TIA Portal V17
General settings
Configuration
Install
Overview
Modify system
System configuration
Summary
Install
Cancel


SIEMENS
Totally Integrated Automation
Welcome to TIA Portal V17
General settings
Configuration
Install
Overview
Modify system
System configuration
Summary
Integrated Engineering
Reduce your time to market
Cancel

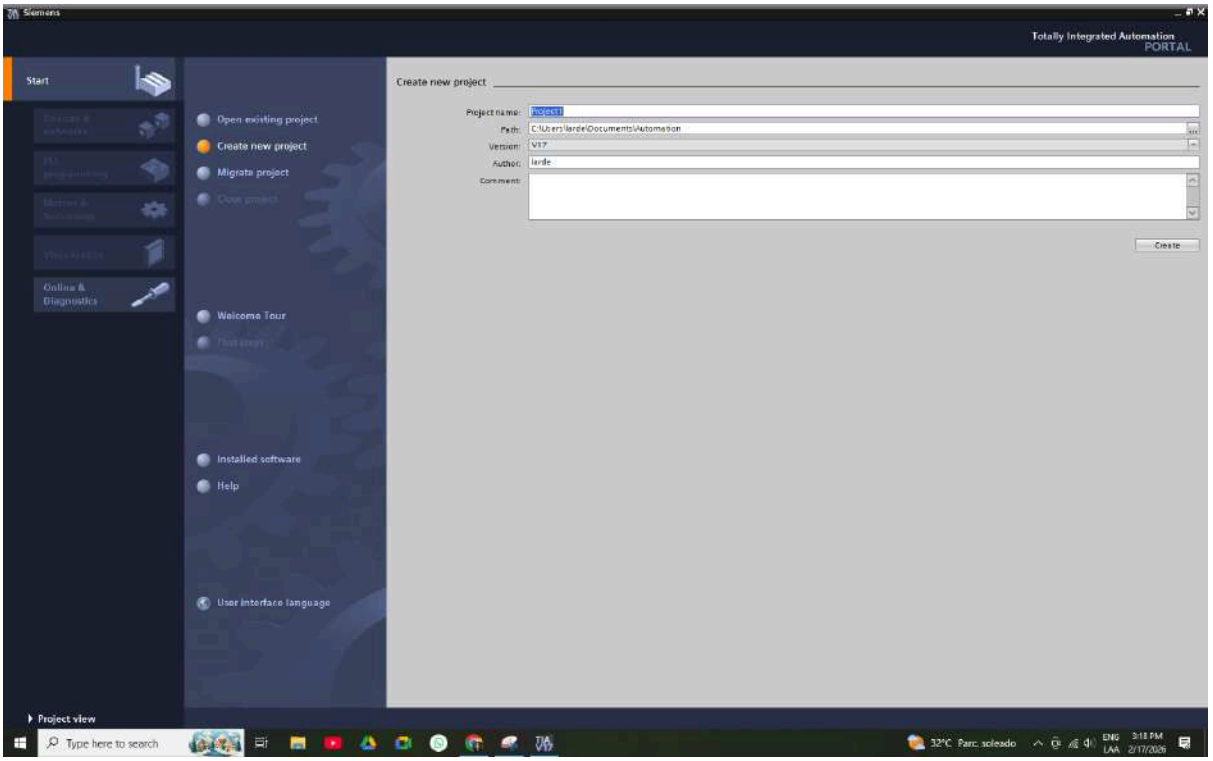

SIEMENS
Totally Integrated Automation
Welcome to TIA Portal V17
General settings
Configuration
Install
Overview
Modify system
System configuration
Summary
Restart

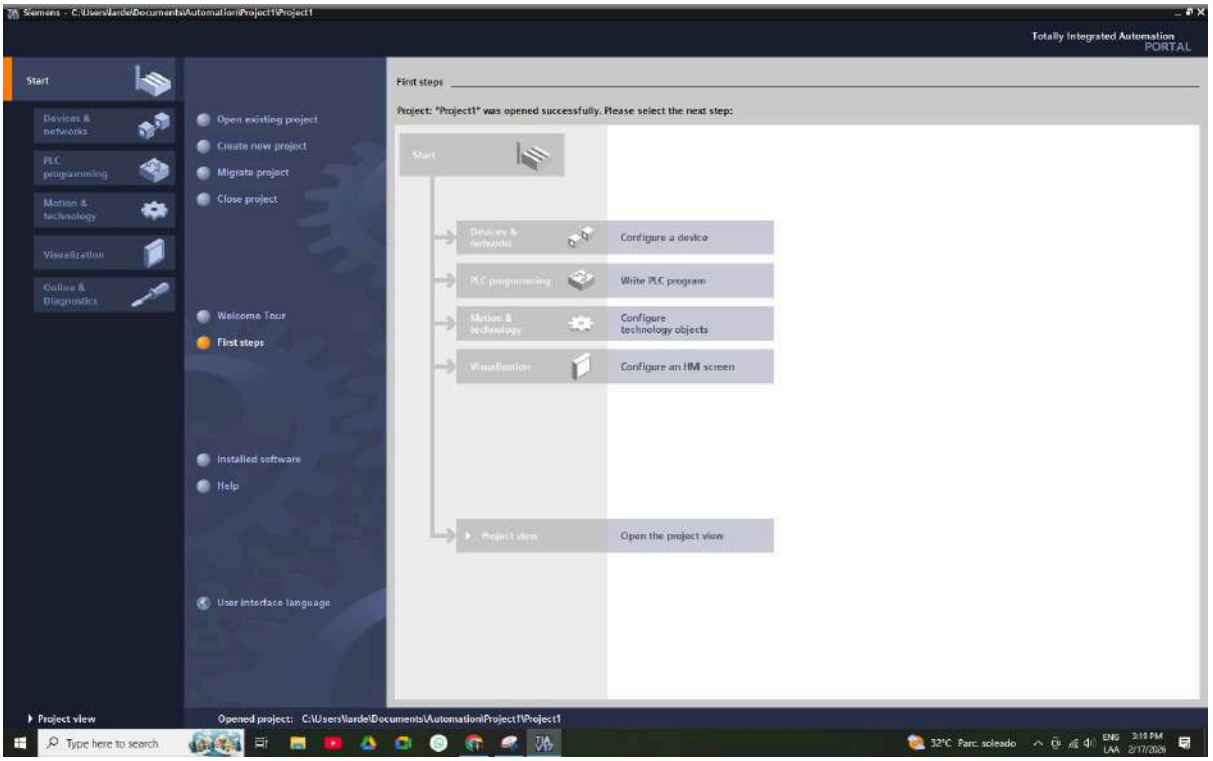

TIA Portal V17
TIA Administra...
Automation License M...
WinCC Runtime Start

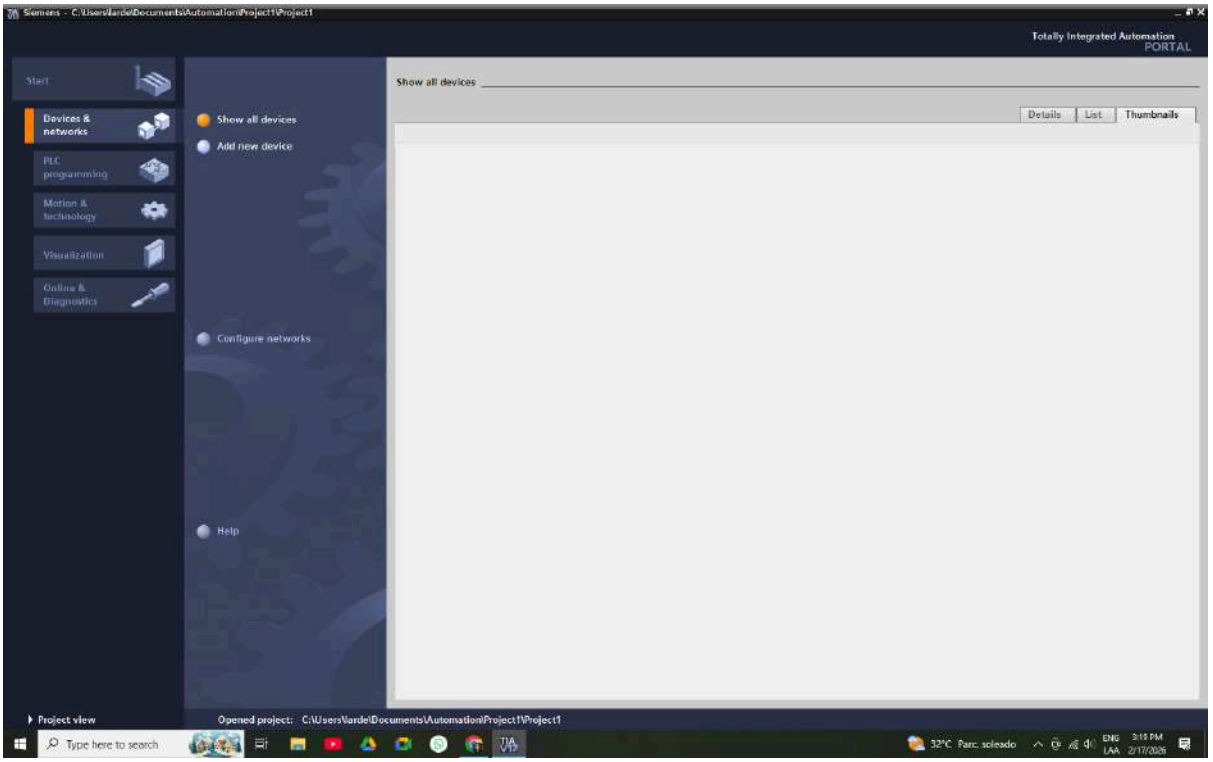
Iniciemos:

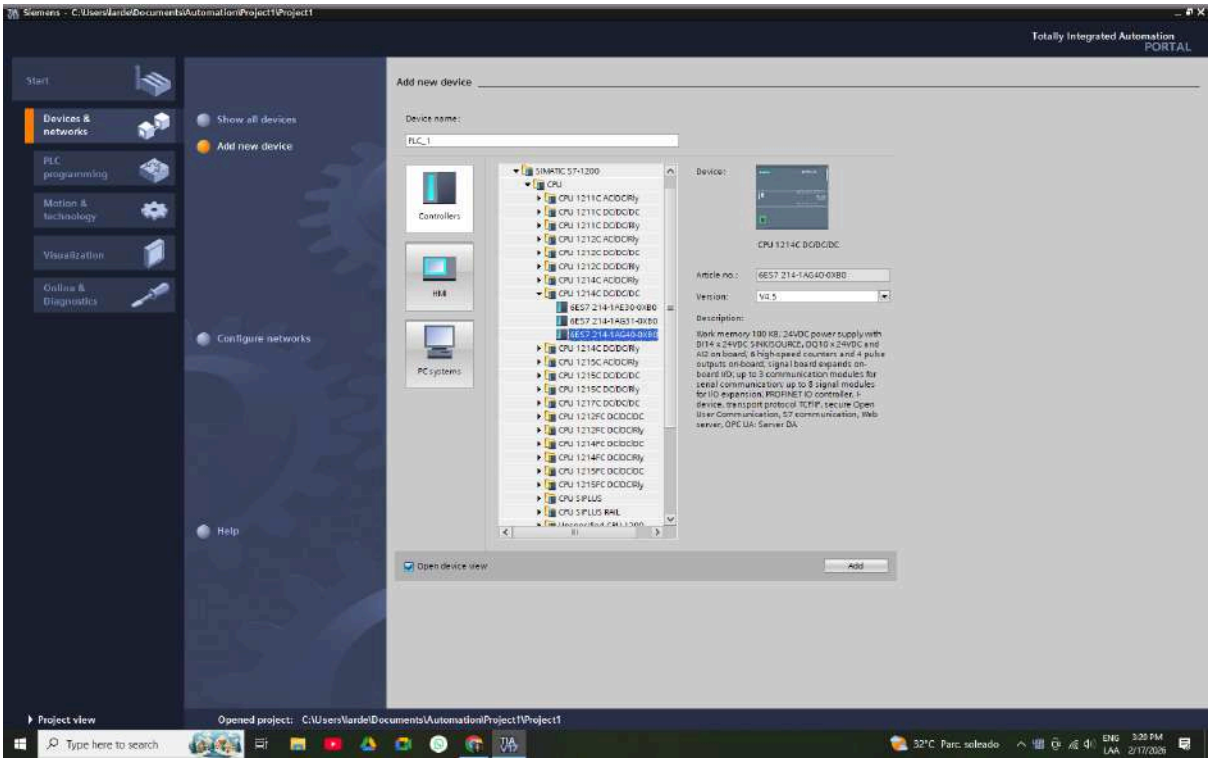

SIEMENS
Totally Integrated Automation
PORTAL V17
© Siemens AG, 2008-2021
TIA Portal V17
TIA Administra...
Automation License M...
WinCC Runtime Start

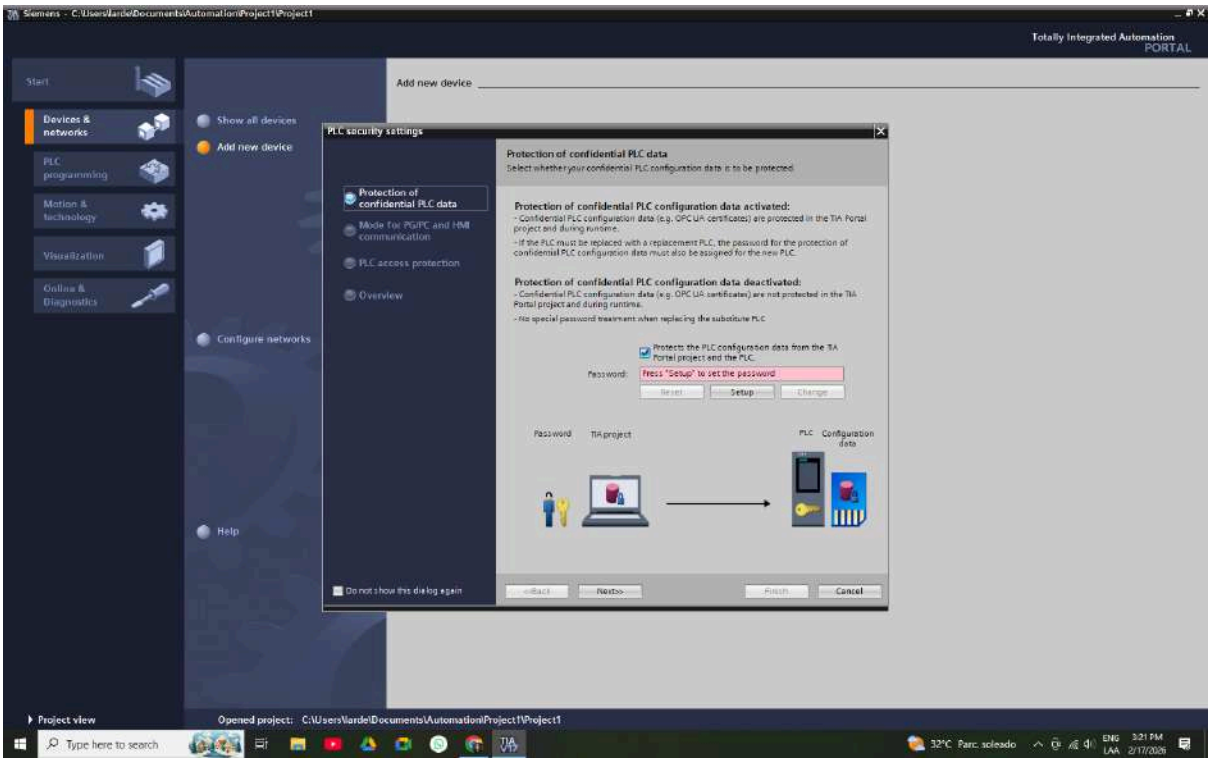
Primer proyecto:

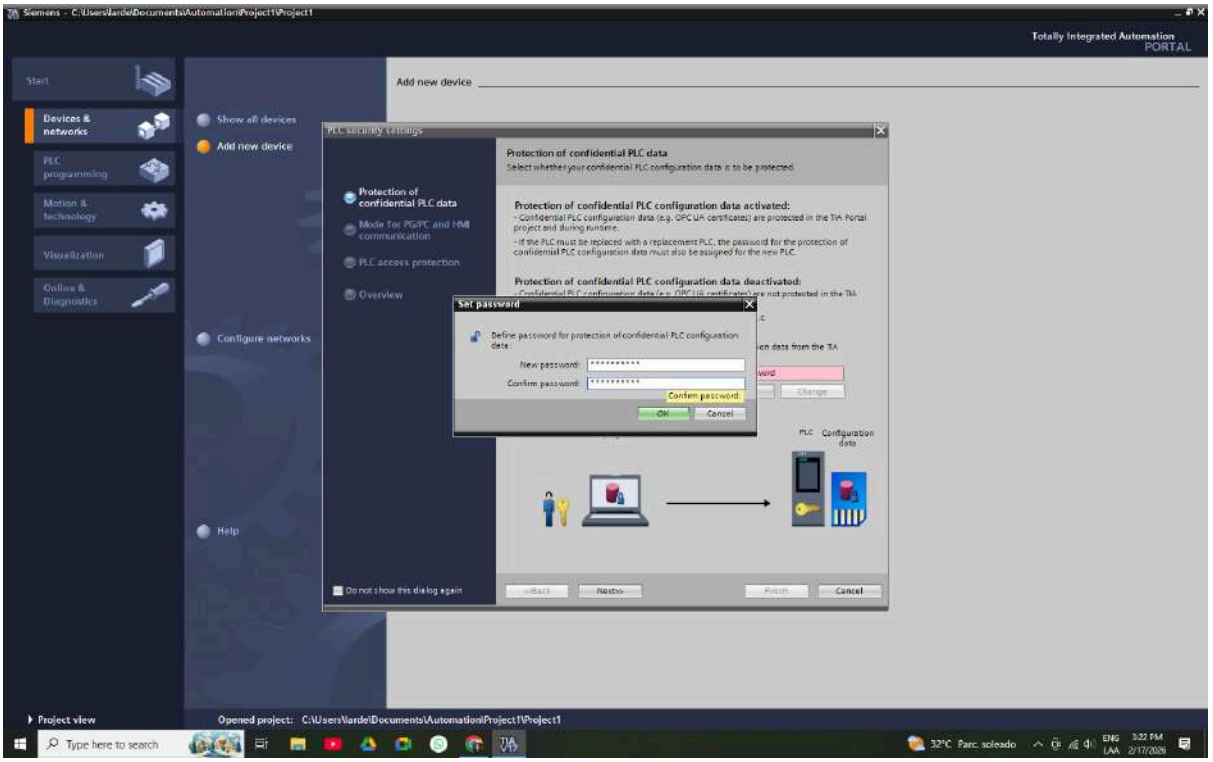


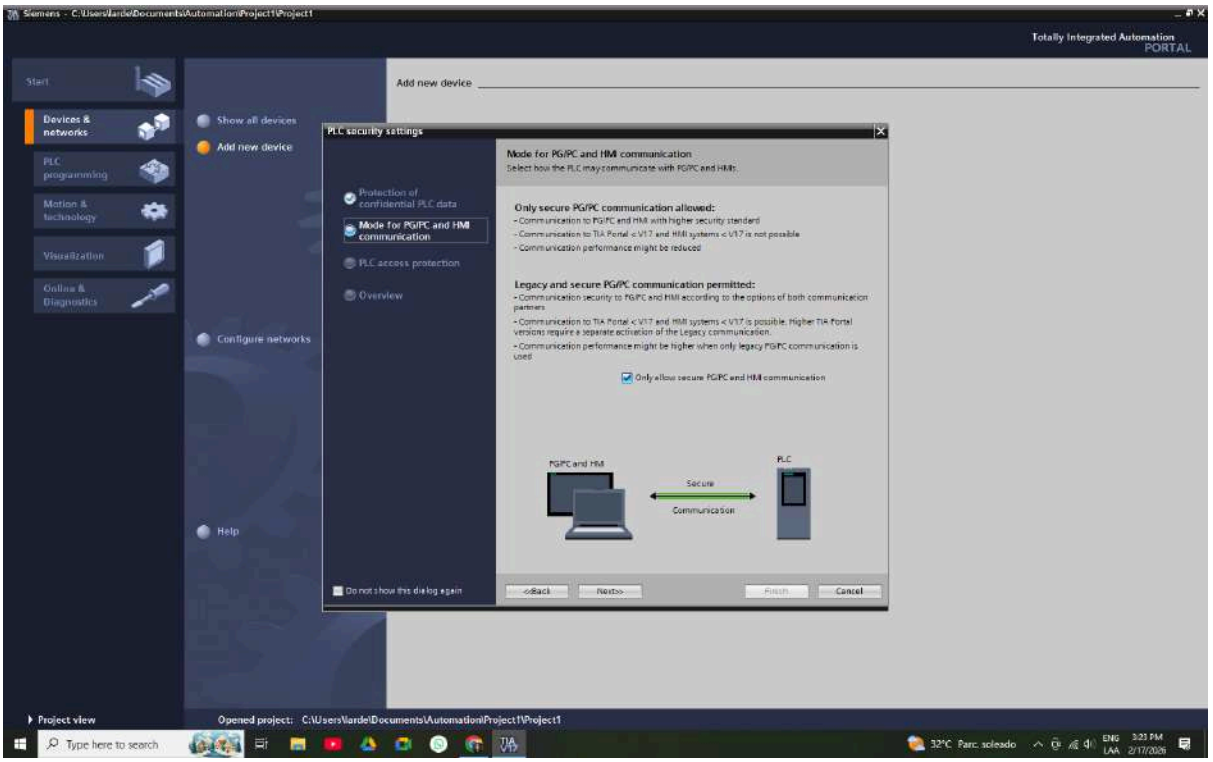


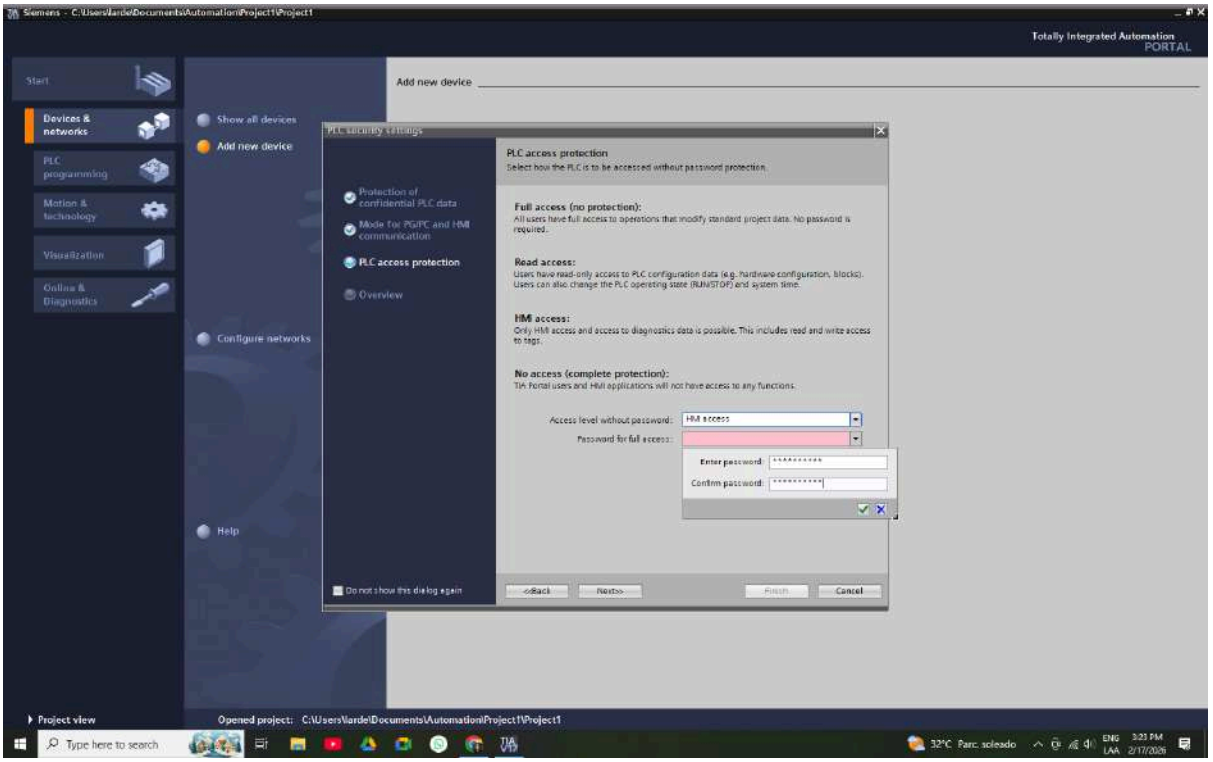

Totally Integrated Automation
PORTAL
Start
Devices & networks
PLC programming
Motion & technology
Visualization
Online & Diagnostics
Show all devices
Add new device
Configure networks
Help
Show all devices
Details
List
Thumbnails
Project view
Opened project: C:\Users\lardet\Documents\Automation\Project1\Project1
Type here to search

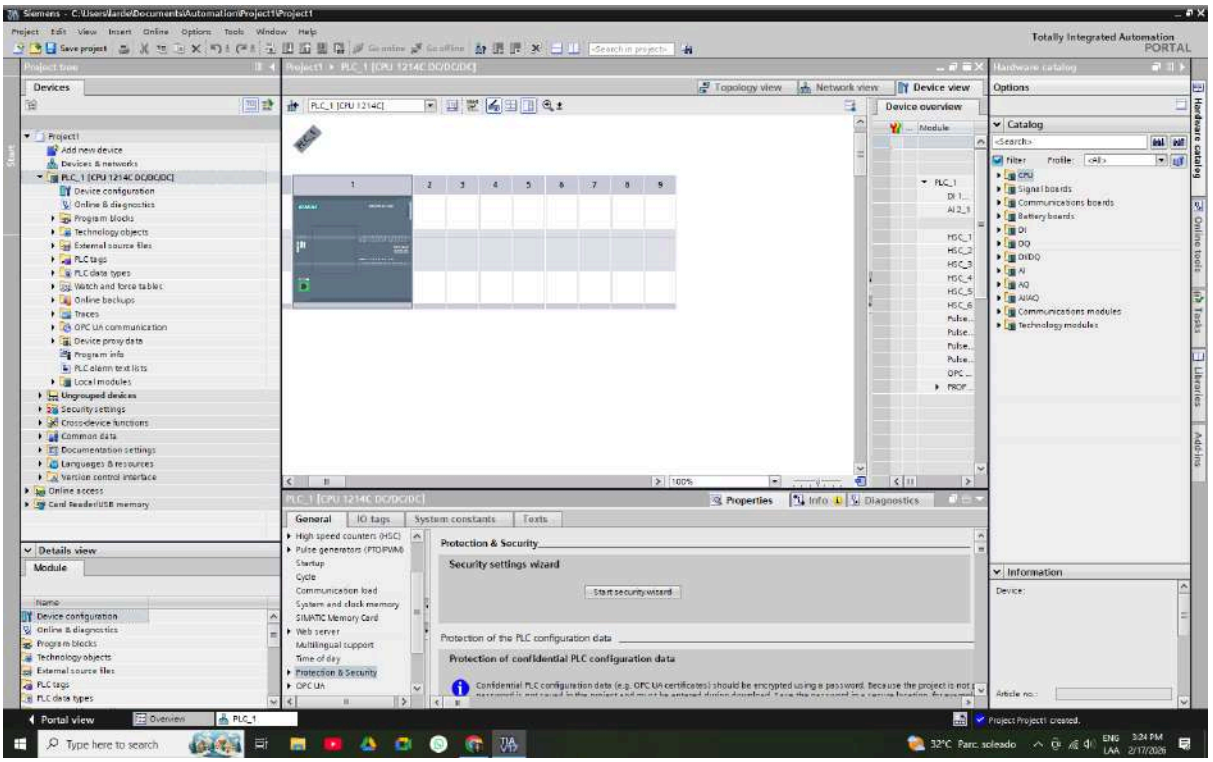

Totally Integrated Automation
PORTAL
Start
Devices & networks
PLC programming
Motion & technology
Visualization
Online & Diagnostics
Add new device
Show all devices
Add new device
Configure networks
Help
Device name:
PLC_1
Controllers
HMI
PC systems
SIMATIC S7-1200
CPU
CPU 1214C DC/DC/DC
Article no.:
6ES7 214-1AG40-0XB0
Version:
V4.5
Description:
Work memory 100 KB, 24VDC power supply with DI14 x 24VDC SINK/SOURCE, DQ10 x 24VDC and AI2 on board, 6 high-speed counters and 4 pulse outputs on-board, signal board expands on-board I/O, up to 3 communication modules for serial communication, up to 8 signal modules for I/O expansion, PROFINET IO controller, I-device, transport protocol TCP/IP, secure Open User Communication, S7 communication, Web server, OPC UA: Server DA
Open device view
Add
Project view
Opened project: C:\Users\lardet\Documents\Automation\Project1\Project1
Type here to search
32°C Parc. soleado
ENG
LAA
3:20 PM
2/17/2026

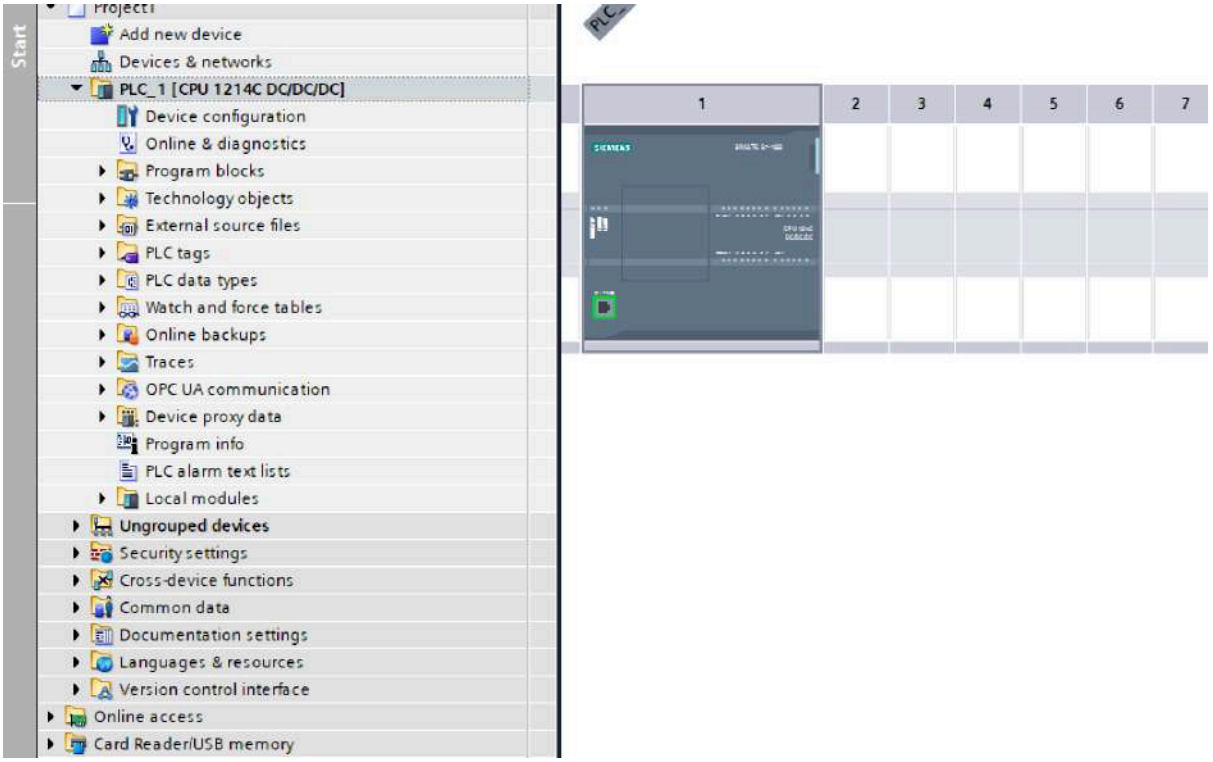

Totally Integrated Automation
PORTAL
Start
Devices & networks
PLC programming
Motion & technology
Visualization
Online & Diagnostics
Add new device
Show all devices
Add new device
Configure networks
Help
PLC security settings
Protection of confidential PLC data
Select whether your confidential PLC configuration data is to be protected.
Protection of confidential PLC data
Mode for PG/PC and HMI communication
PLC access protection
Overview
Protection of confidential PLC configuration data activated:
- Confidential PLC configuration data (e.g. OPC UA certificates) are protected in the TIA Portal project and during runtime.
- If the PLC must be replaced with a replacement PLC, the password for the protection of confidential PLC configuration data must also be assigned for the new PLC.
Protection of confidential PLC configuration data deactivated:
- Confidential PLC configuration data (e.g. OPC UA certificates) are not protected in the TIA Portal project and during runtime.
- No special password treatment when replacing the substitute PLC
Protects the PLC configuration data from the TIA Portal project and the PLC.
Password:
Press "Setup" to set the password
Reset
Setup
Change
Password
TIA project
PLC
Configuration data
Do not show this dialog again
<Back
Next>
Finish
Cancel
Project view
Opened project: C:\Users\lardet\Documents\Automation\Project1\Project1
Type here to search
32°C Parc. soleado
ENG
LAA
3:21 PM
2/17/2026

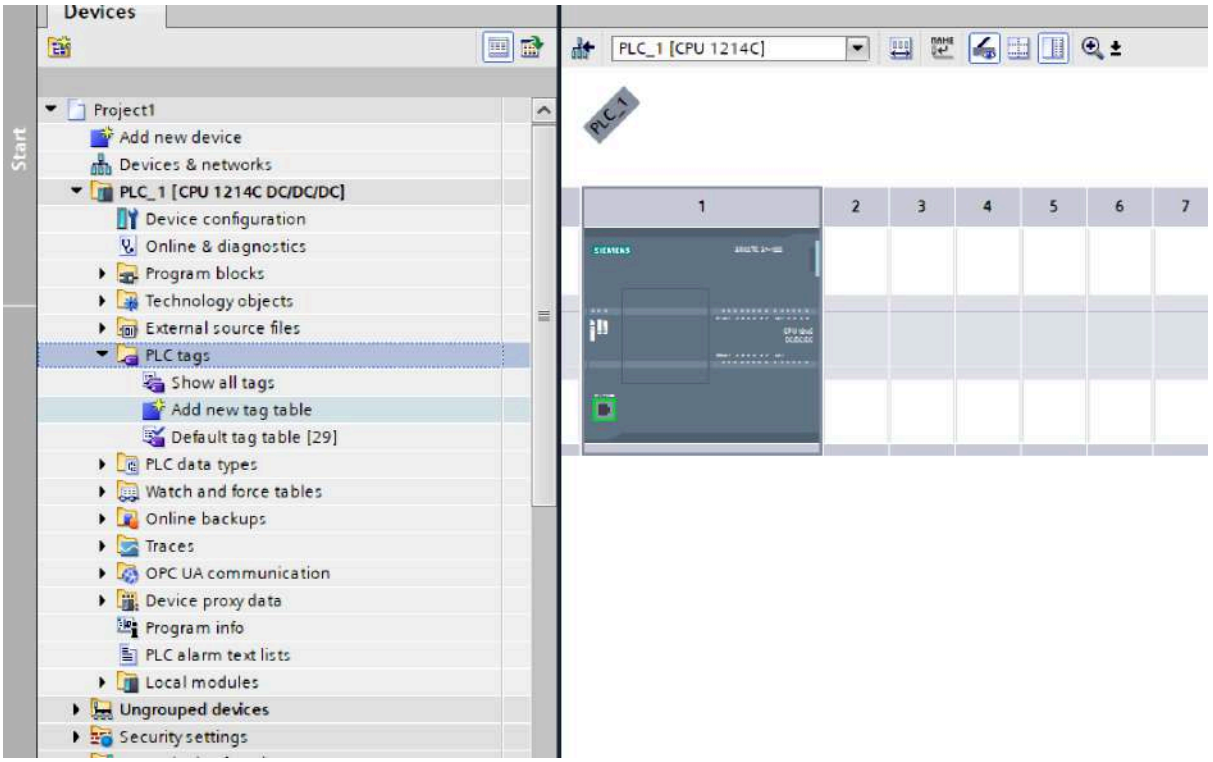

Totally Integrated Automation
PORTAL
Start
Devices & networks
PLC programming
Motion & technology
Visualization
Online & Diagnostics
Add new device
Show all devices
Add new device
Configure networks
Help
Project view
Protection of confidential PLC data
Mode for PG/PC and HMI communication
PLC access protection
Overview
Set password
Define password for protection of confidential PLC configuration data:
New password:
Confirm password:
Confirm password
OK
Cancel
Protection of confidential PLC data
Select whether your confidential PLC configuration data is to be protected.
Protection of confidential PLC configuration data activated:
Protection of confidential PLC configuration data deactivated:
PLC
Configuration data
Do not show this dialog again
<Back
Next>
Finish
Cancel
Opened project: C:\Users\lardet\Documents\Automation\Project1\Project1
Type here to search
32°C Parc. soleado
ENG
LAA
3:22 PM
2/17/2026


Totally Integrated Automation
PORTAL
Start
Devices & networks
PLC programming
Motion & technology
Visualization
Online & Diagnostics
Add new device
Show all devices
Add new device
Configure networks
Help
Project view
Mode for PG/PC and HMI communication
Select how the PLC may communicate with PG/PC and HMI.
Only secure PG/PC communication allowed:
Legacy and secure PG/PC communication permitted:
Only allow secure PG/PC and HMI communication
PG/PC and HMI
Secure
Communication
PLC
Protection of confidential PLC data
PLC access protection
Overview
Do not show this dialog again
<Back
Next>
Finish
Cancel
Opened project: C:\Users\lardet\Documents\Automation\Project1\Project1
Type here to search
32°C Parc. soleado
ENG
LAA
3:23 PM
2/17/2026

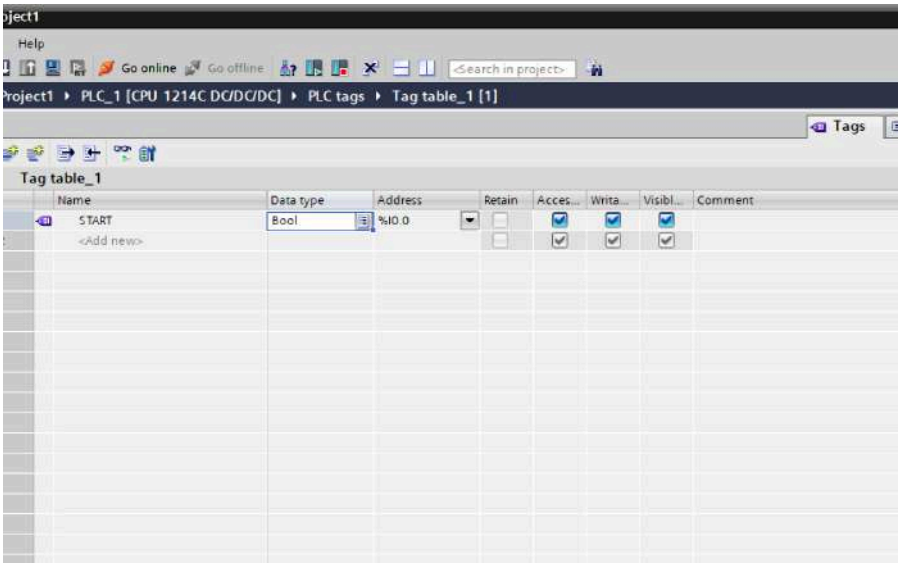

Totally Integrated Automation
PORTAL
Add new device
Devices & networks
PLC programming
Motion & technology
Visualization
Online & Diagnostics
Show all devices
Add new device
Configure networks
Help
PLC access protection
Select how the PLC is to be accessed without password protection.
Full access (no protection):
Read access:
HMI access:
No access (complete protection):
Access level without password:
Password for full access:
Enter password:
Confirm password:
Do not show this dialog again
<Back
Next>
Finish
Cancel
Project view
Opened project: C:\Users\lardet\Documents\Automation\Project1\Project1
Type here to search

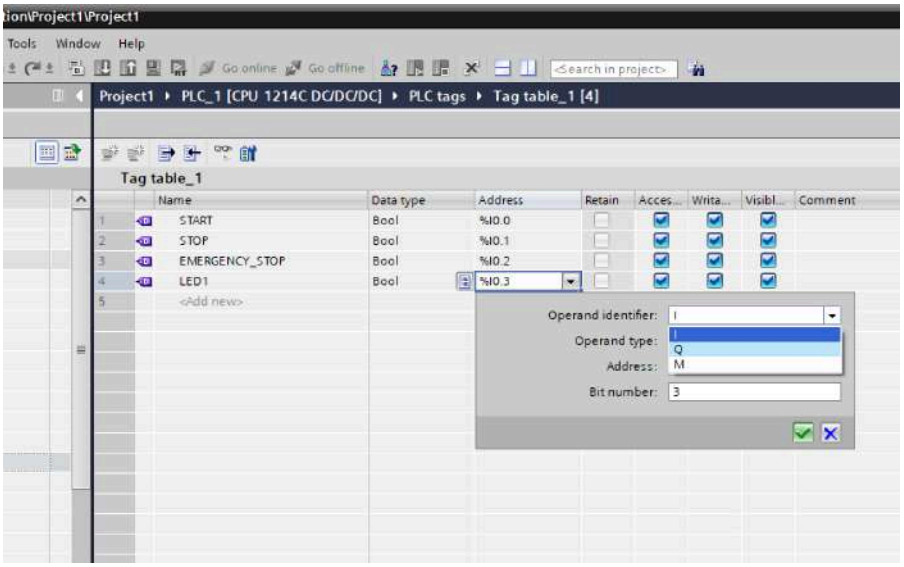

Totally Integrated Automation
PORTAL
Devices
Topology view
Network view
Device view
Device overview
Catalog
Options
Protection & Security
Security settings wizard
Start security wizard
Protection of the PLC configuration data
Protection of confidential PLC configuration data
Information
Portal view
Overview
Project Project1 created.
Type here to search

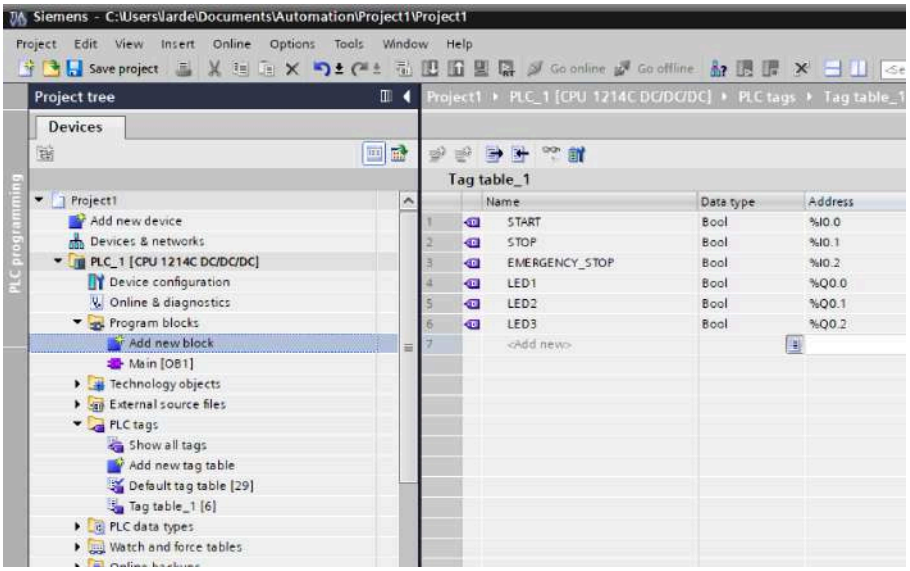

Start
Add new device
Devices & networks
PLC_1 [CPU 1214C DC/DC/DC]
Device configuration
Online & diagnostics
Program blocks
Technology objects
External source files
PLC tags
PLC data types
Watch and force tables
Online backups
Traces
OPC UA communication
Device proxy data
Program info
PLC alarm text lists
Local modules
Ungrouped devices
Security settings
Cross-device functions
Common data
Documentation settings
Languages & resources
Version control interface
Online access
Card Reader/USB memory
1
2
3
4
5
6
7

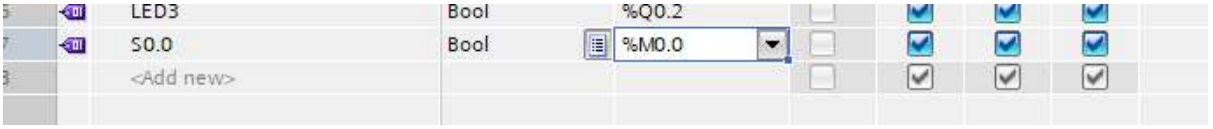

Devices
PLC_1 [CPU 1214C]
Start
Project1
Add new device
Devices & networks
PLC_1 [CPU 1214C DC/DC/DC]
Device configuration
Online & diagnostics
Program blocks
Technology objects
External source files
PLC tags
Show all tags
Add new tag table
Default tag table [29]
PLC data types
Watch and force tables
Online backups
Traces
OPC UA communication
Device proxy data
Program info
PLC alarm text lists
Local modules
Ungrouped devices
Security settings
1
2
3
4
5
6
7

Observemos una de las diferencias más importantes respecto de LOGO! Soft Comfort V8.4, que es una tabla para las variables y sus tipos, así como comentarios, nombres para que sea más fácil programar. Similar a la tabla monitor en el simulador didáctico.

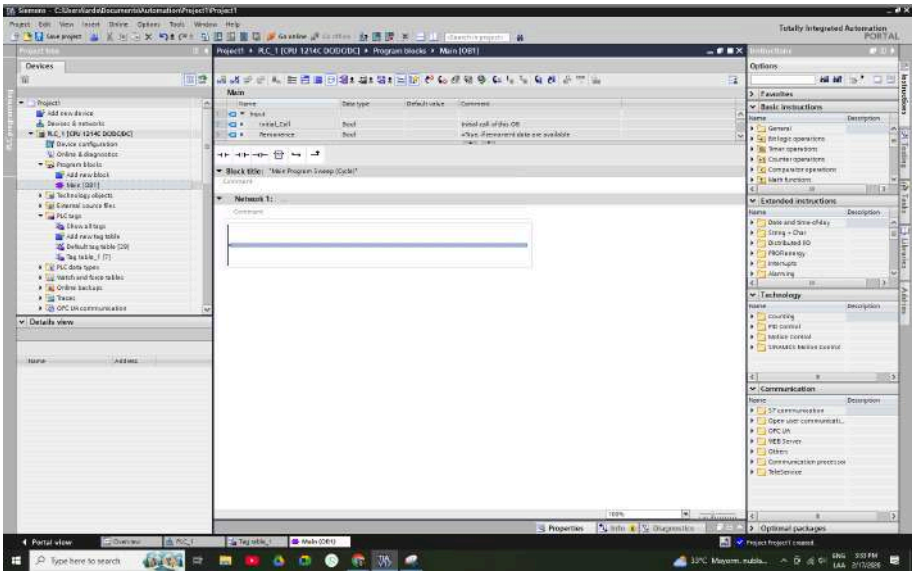


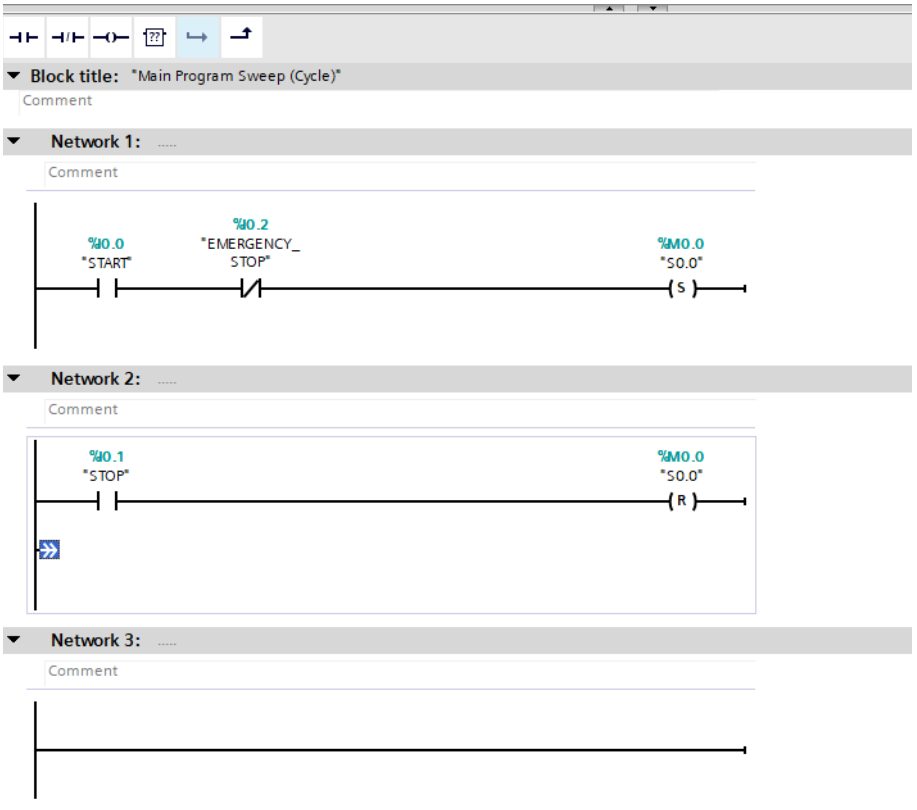


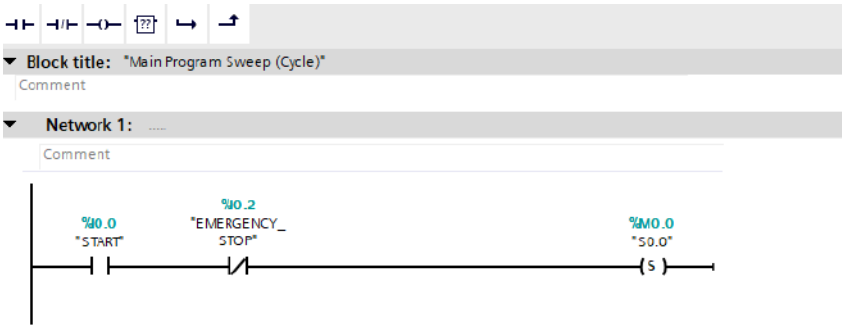


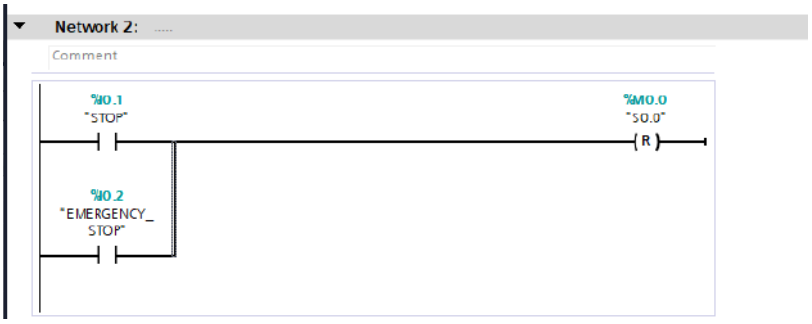


Luego en el segmento principal del programa, codifiquemos utilizando LADDER para hacer un sistema de retención (SET-RESET).

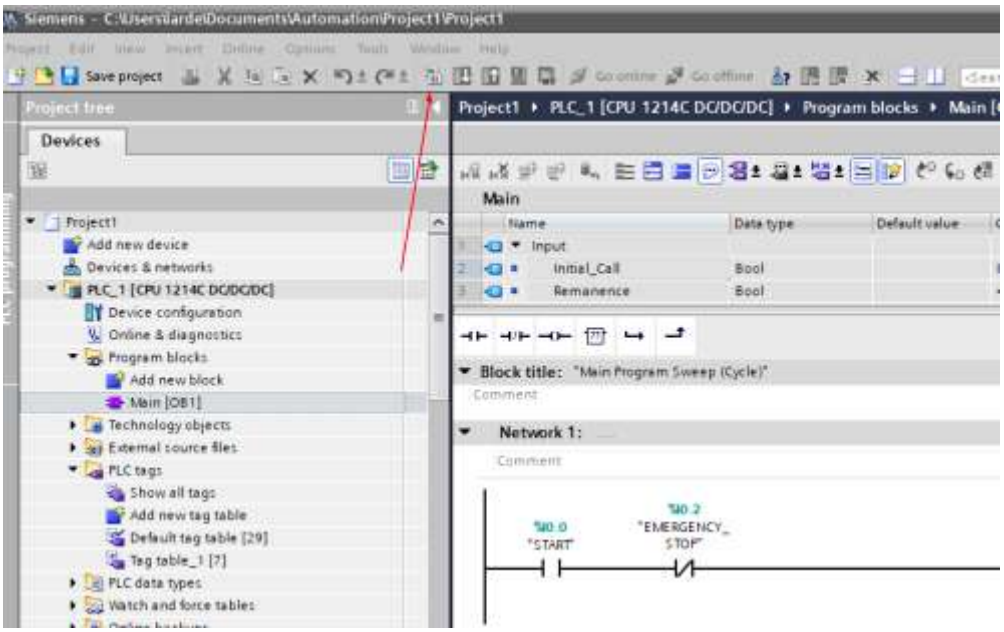


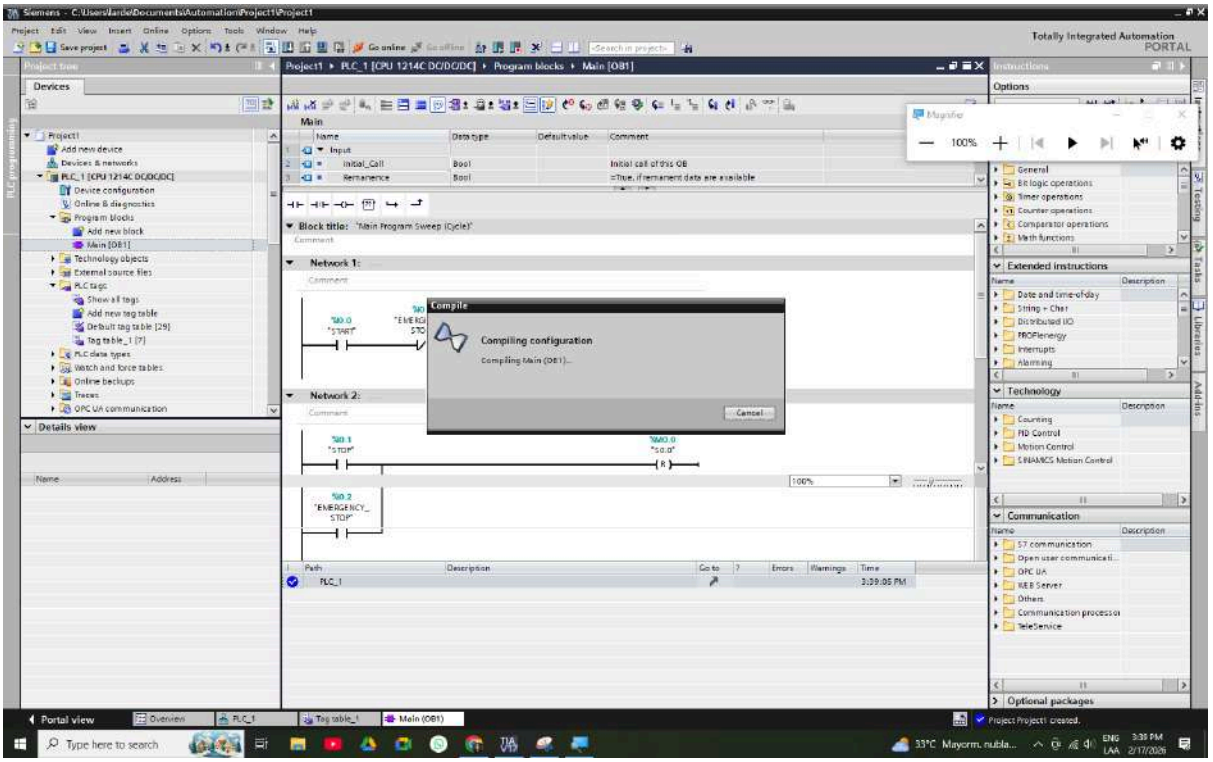




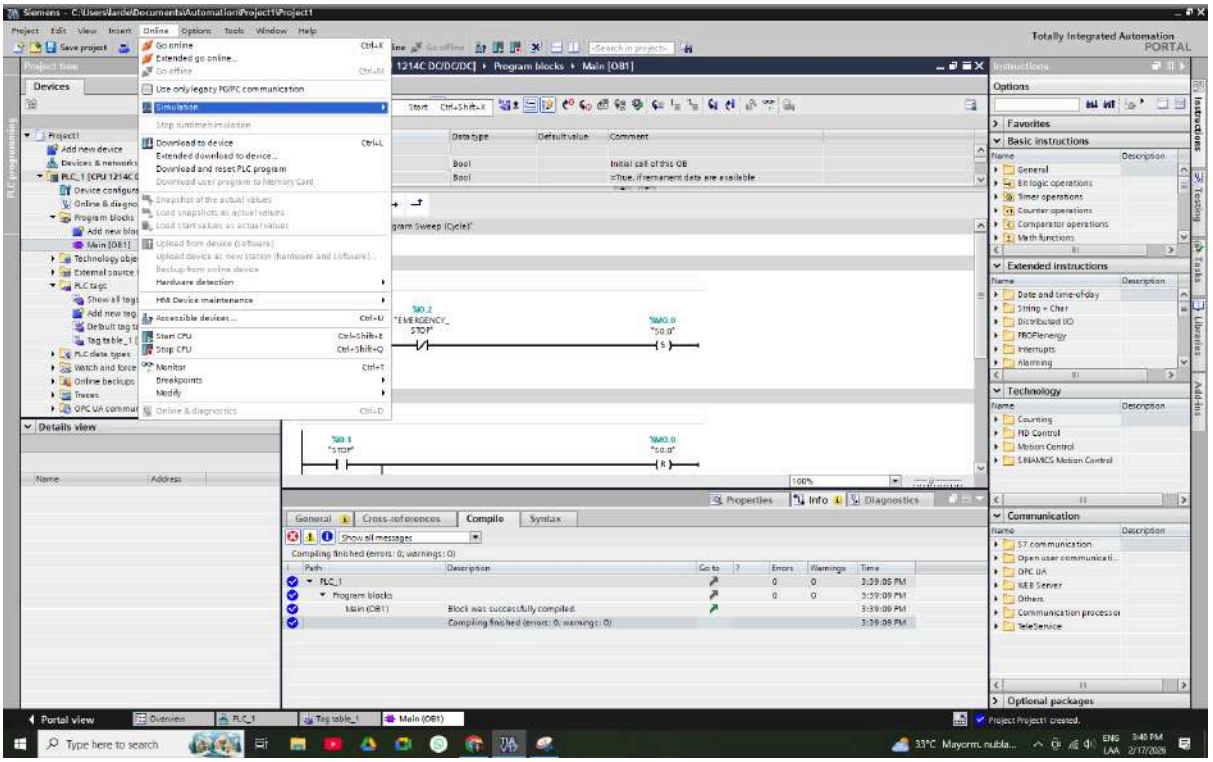


Para simular debemos compilar el proyecto.





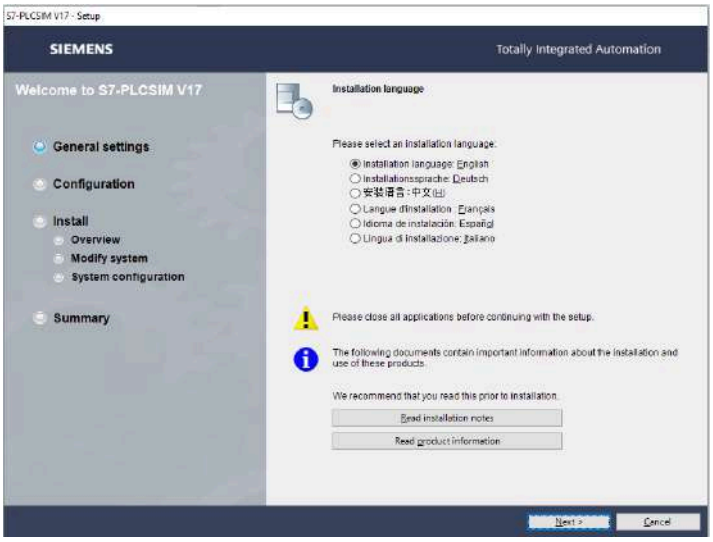
Y luego es posible simular.

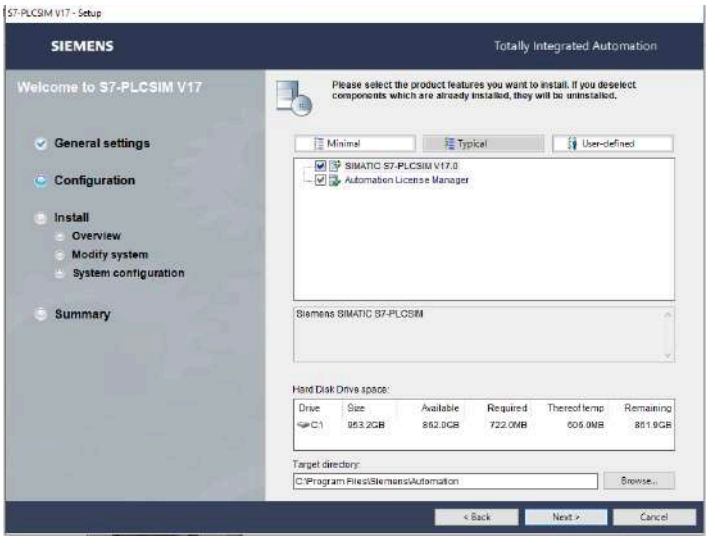


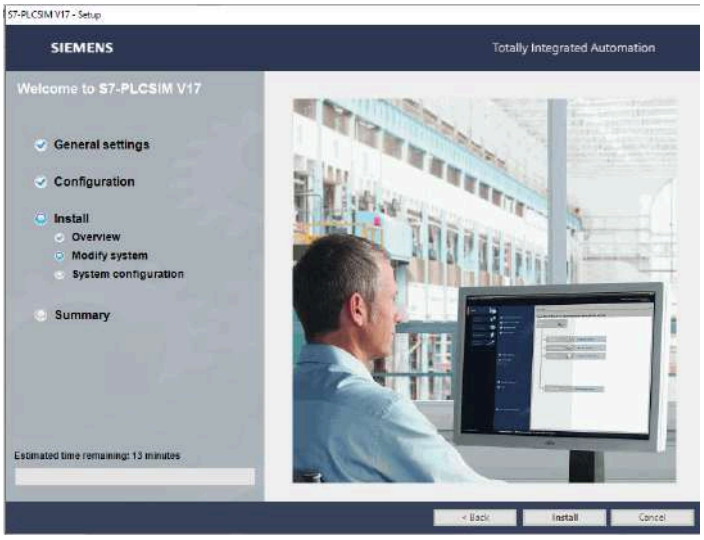
Para simular debemos tener instalado el SIMATIC PLC SIM V17.

S7 PLCSIM y Factory I/O

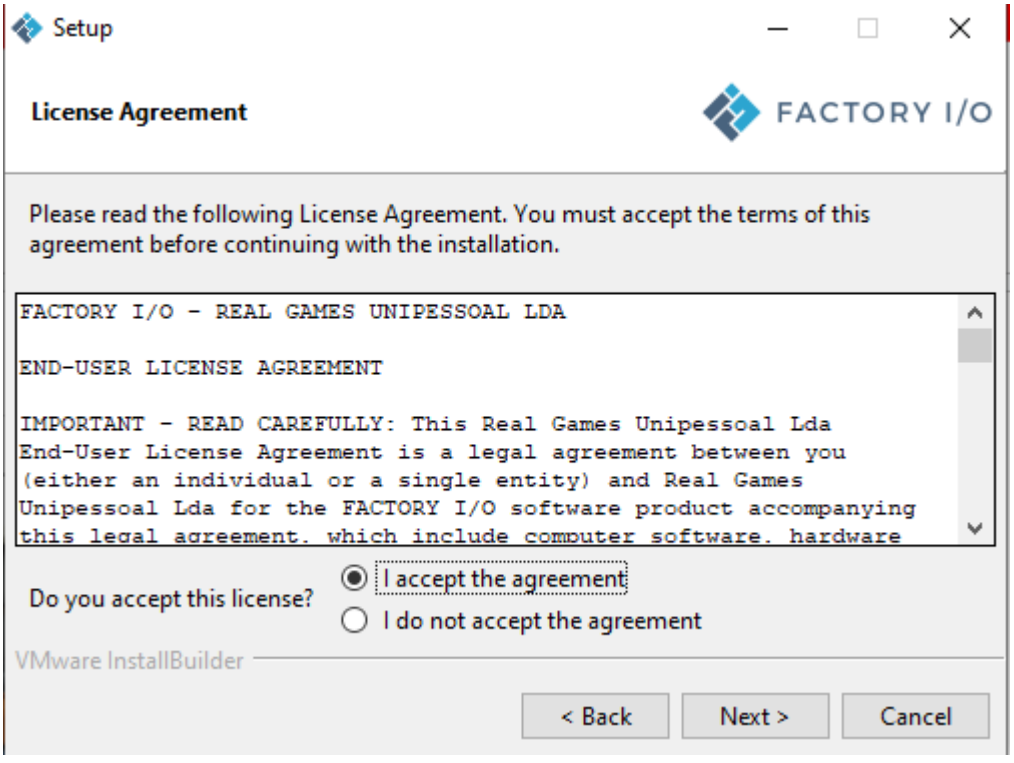
Para contar con una herramienta similar al simulador didáctico, podemos combinar (para lo cual se requiere gran poder de computo) TIA Portal V17 Step 7, con S7 PLCSIM V17 y Factory I/O V2.5.8.

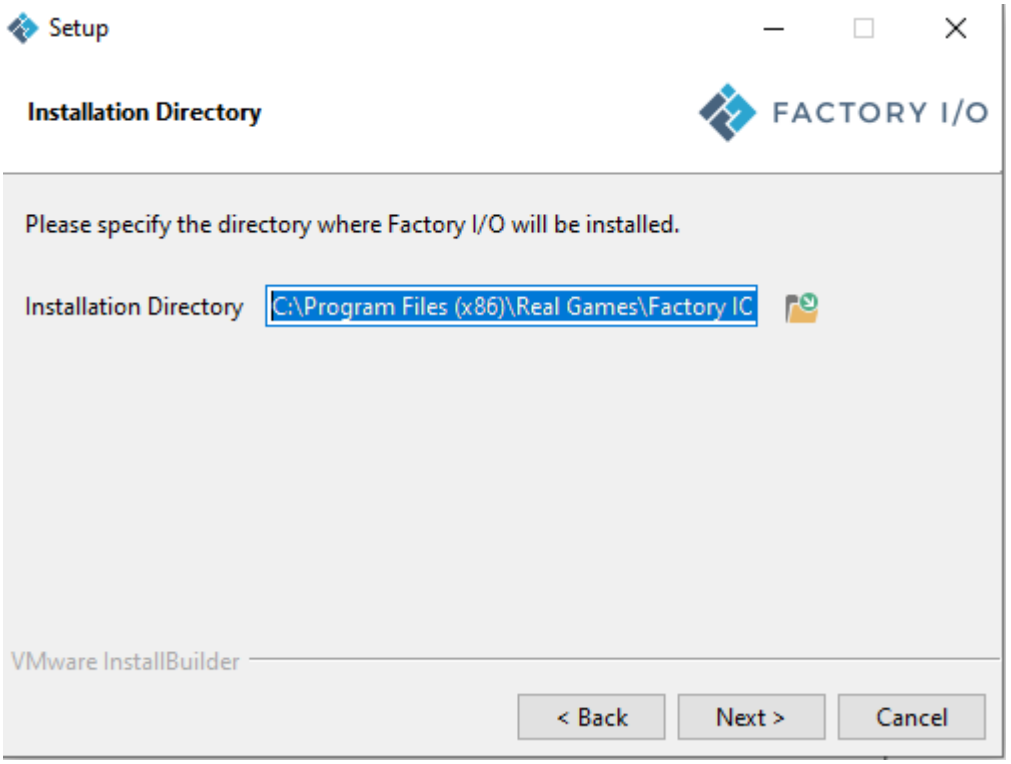


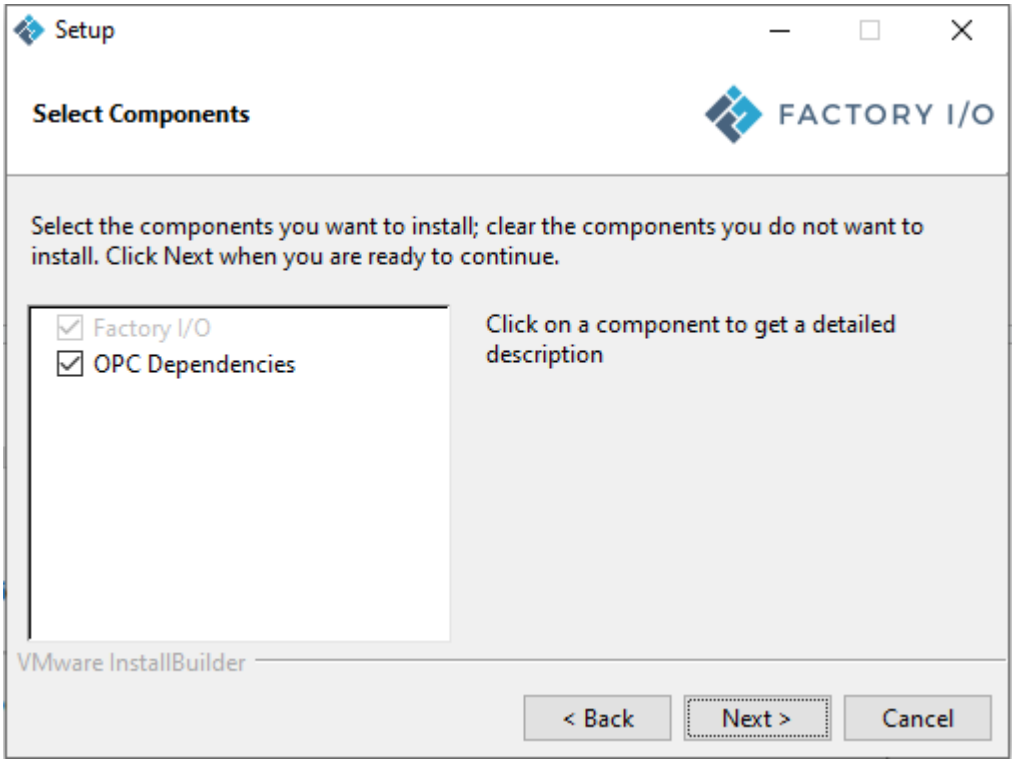


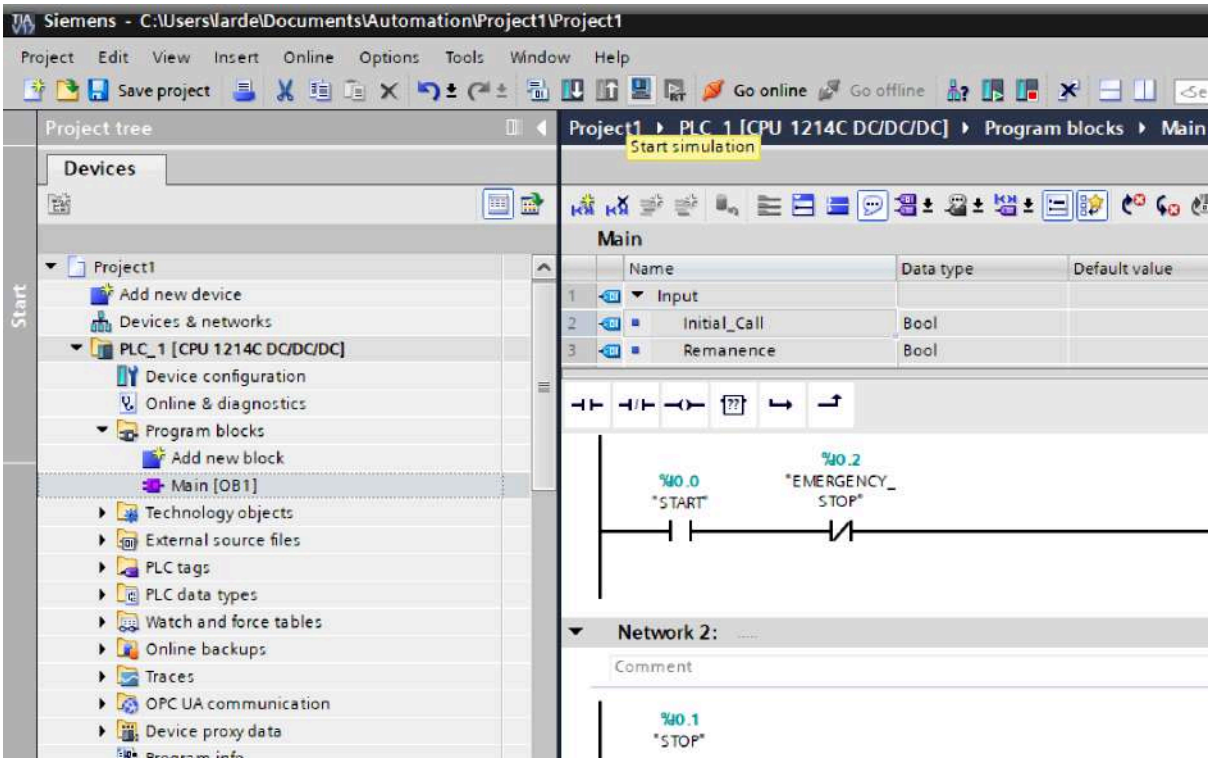


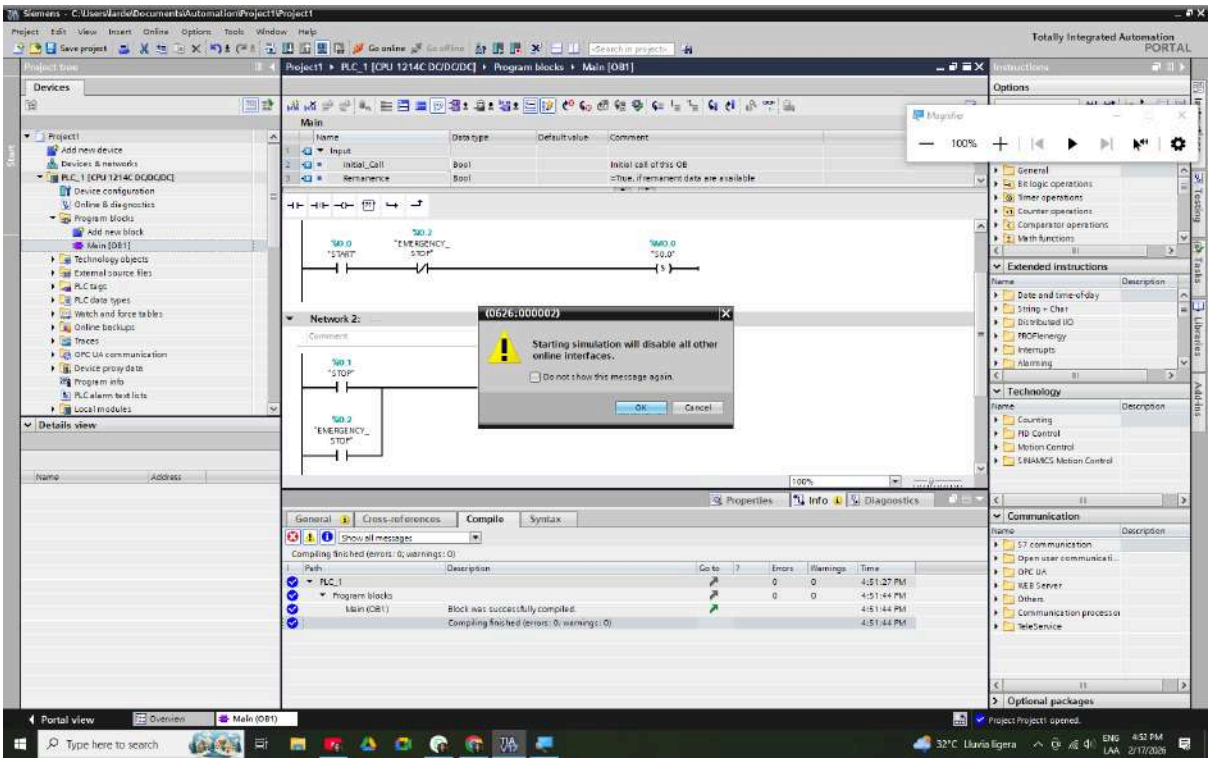

Setup
Setup - Factory I/O
Welcome to the Factory I/O Setup Wizard.
realgames
< Back
Next >
Cancel

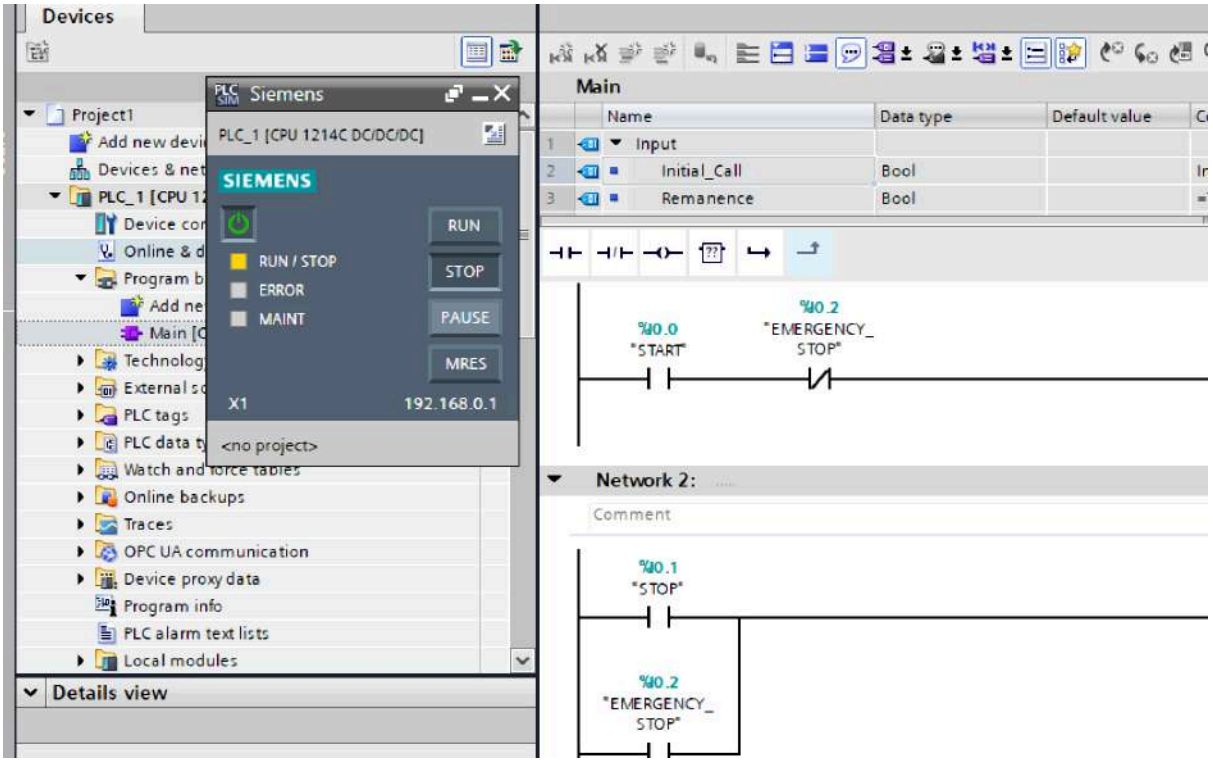

Setup
License Agreement
FACTORY I/O
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.
FACTORY I/O - REAL GAMES UNIPESSOAL LDA
END-USER LICENSE AGREEMENT
IMPORTANT - READ CAREFULLY: This Real Games Unipessoal Lda End-User License Agreement is a legal agreement between you (either an individual or a single entity) and Real Games Unipessoal Lda for the FACTORY I/O software product accompanying this legal agreement, which include computer software, hardware
Do you accept this license?
I accept the agreement
I do not accept the agreement
VMware InstallBuilder
< Back
Next >
Cancel

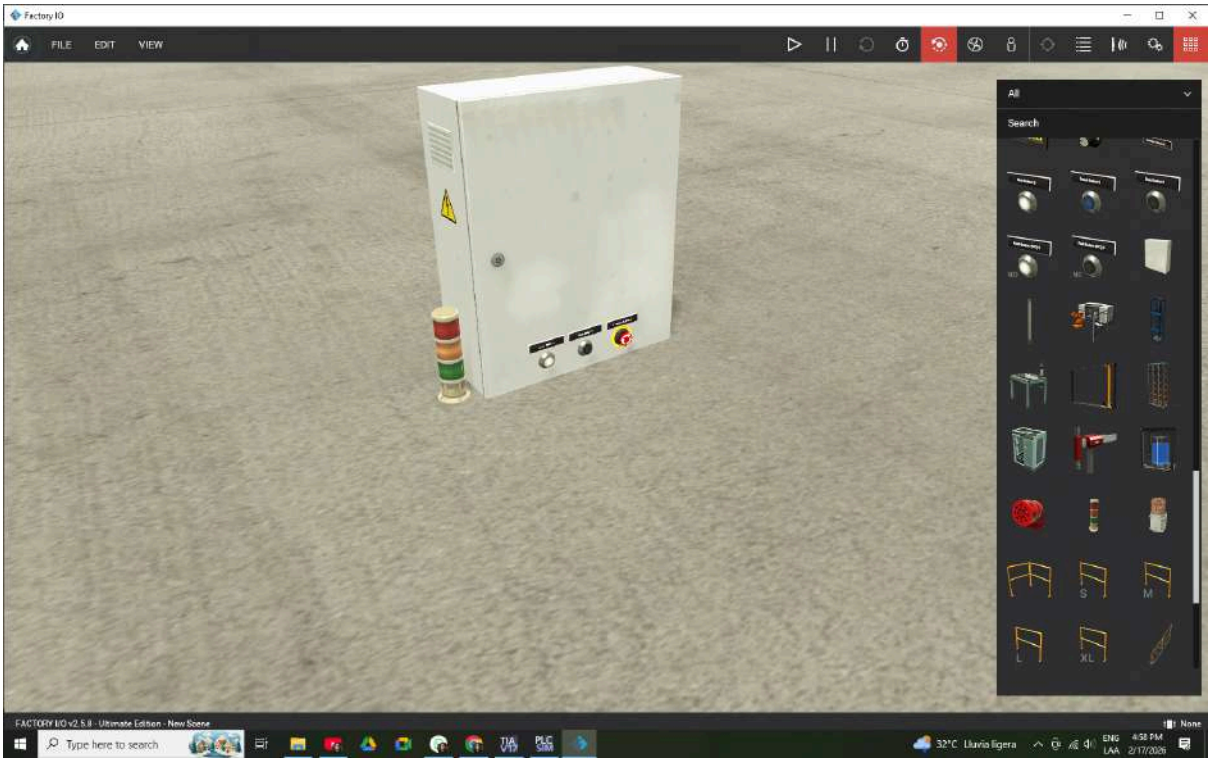

Setup
Installation Directory
FACTORY I/O
Please specify the directory where Factory I/O will be installed.
Installation Directory
C:\Program Files (x86)\Real Games\Factory IC
VMware InstallBuilder
< Back
Next >
Cancel

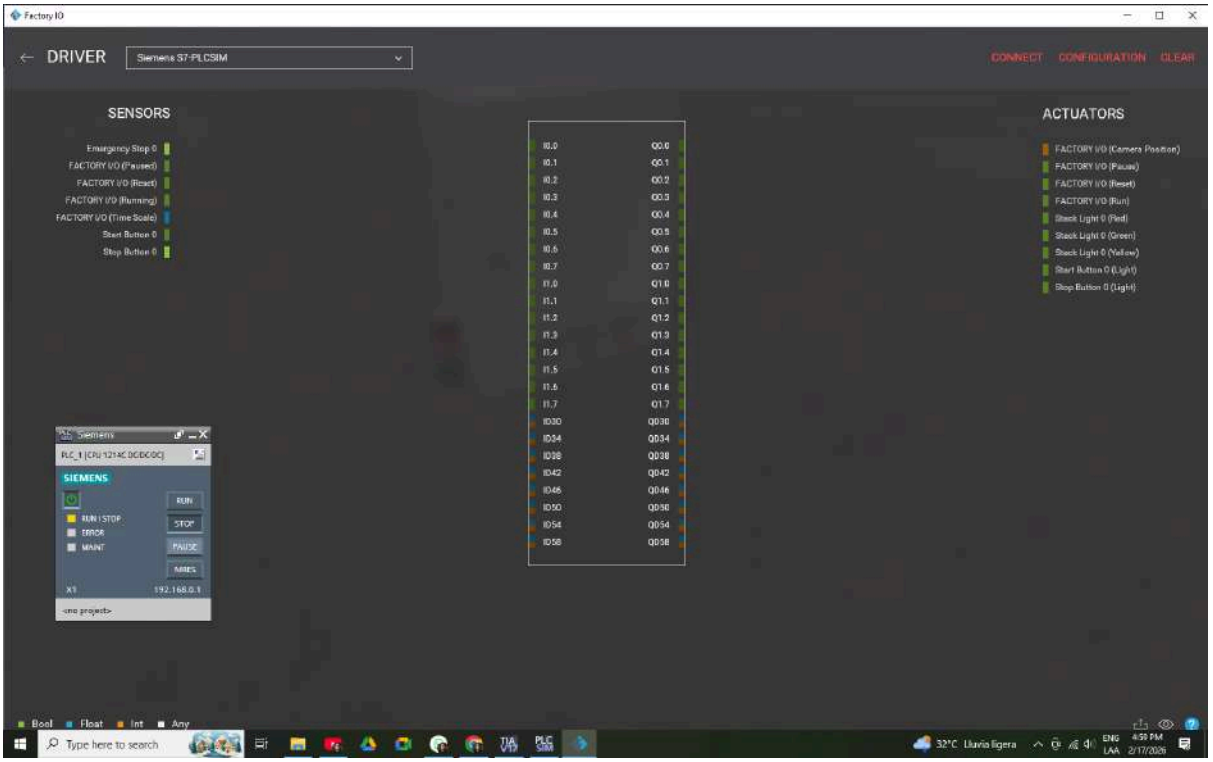

Setup
Select Components
FACTORY I/O
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
Factory I/O
OPC Dependencies
Click on a component to get a detailed description
VMware InstallBuilder
< Back
Next >
Cancel


Siemens - C:\Users\larde\Documents\Automation\Project1\Project1
Project Edit View Insert Online Options Tools Window Help
Save project
Go online
Go offline
Project tree
Devices
Project1
Add new device
Devices & networks
PLC_1 [CPU 1214C DC/DC/DC]
Device configuration
Online & diagnostics
Program blocks
Add new block
Main [OB1]
Technology objects
External source files
PLC tags
PLC data types
Watch and force tables
Online backups
Traces
OPC UA communication
Device proxy data
Project1 ▸ PLC_1 [CPU 1214C DC/DC/DC] ▸ Program blocks ▸ Main
Start simulation
Main
Name
Data type
Default value
Input
Initial_Call
Bool
Remanence
Bool
%I0.0
"START"
%I0.2
"EMERGENCY_ STOP"
Network 2:
Comment
%I0.1
"STOP"

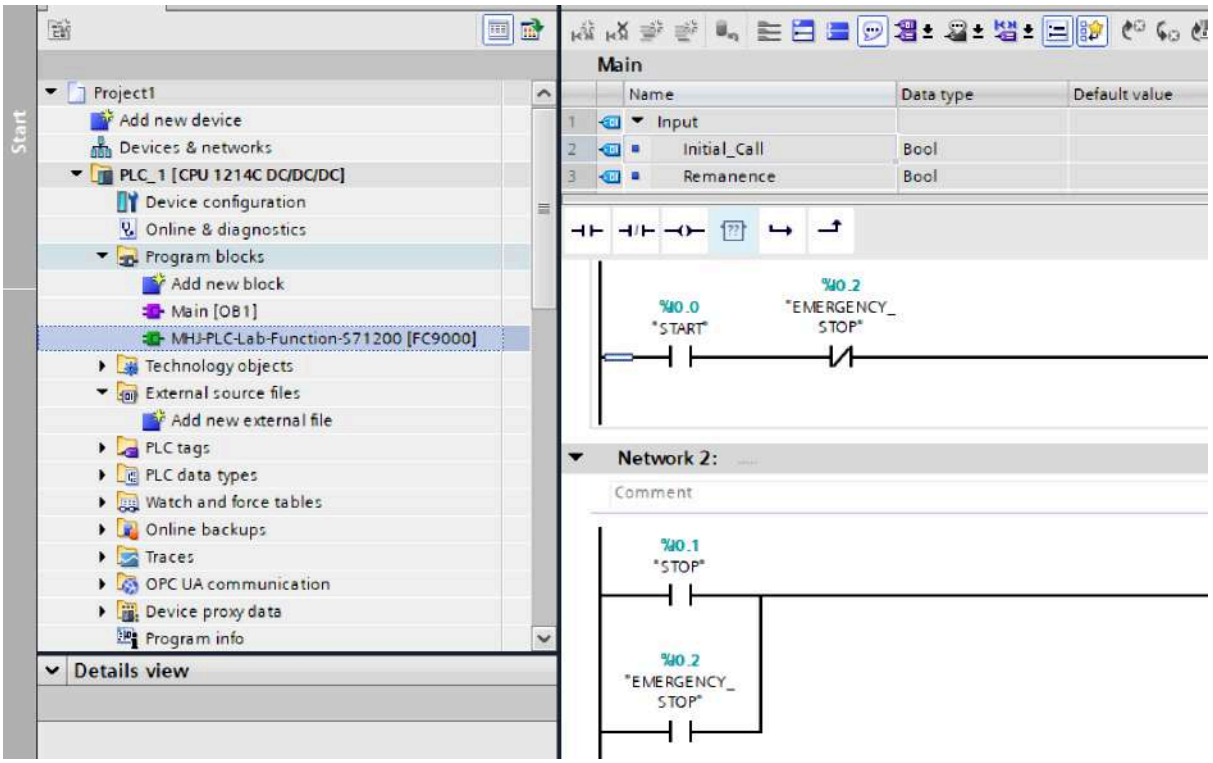

Totally Integrated Automation
PORTAL
Starting simulation will disable all other online interfaces.
Do not show this message again.
OK
Cancel
Network 2:
Network 3:
Network 4:

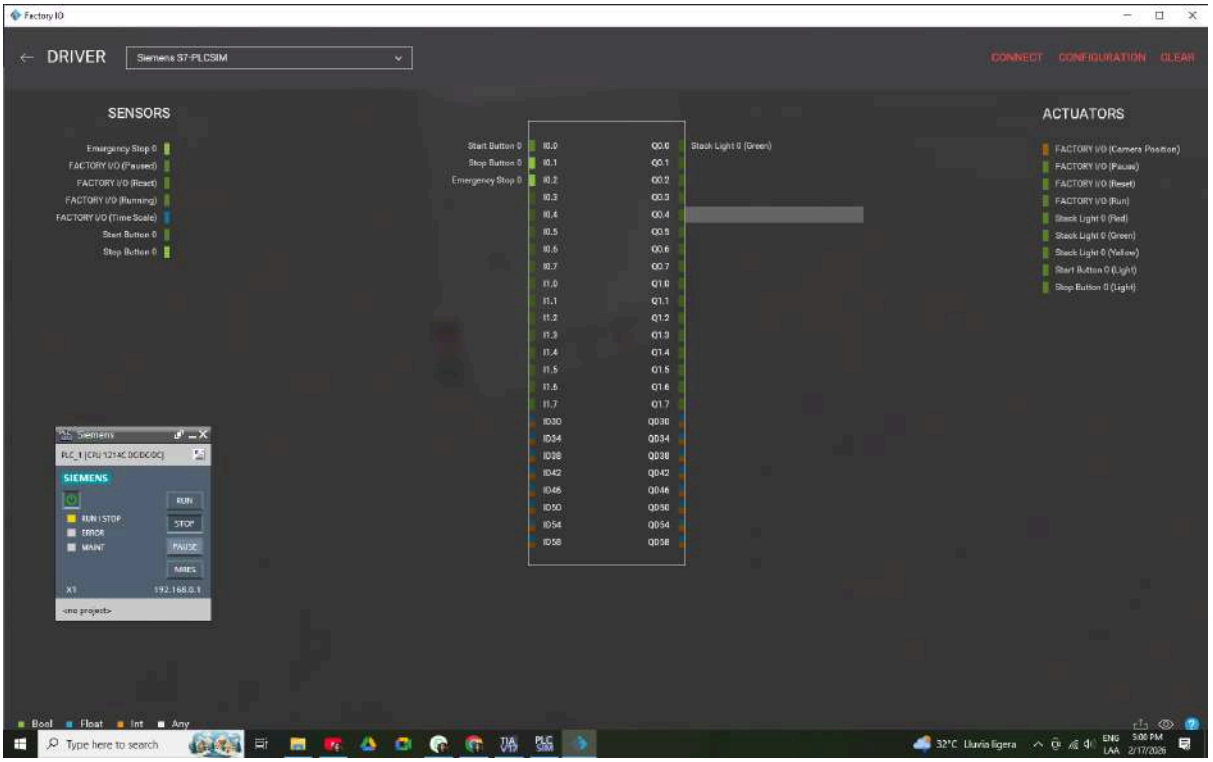

Devices
Project1
Add new device
Devices & networks
PLC_1 [CPU 1214C DC/DC/DC]
Device configuration
Online & diagnostics
Program blocks
Add new block
Main [OB1]
Technology objects
External source files
PLC tags
PLC data types
Watch and force tables
Online backups
Traces
OPC UA communication
Device proxy data
Program info
PLC alarm text lists
Local modules
Details view
Siemens
PLC_1 [CPU 1214C DC/DC/DC]
SIEMENS
RUN / STOP
ERROR
MAINT
RUN
STOP
PAUSE
MRES
X1
192.168.0.1
<no project>
Main
Name
Data type
Default value
Input
Initial_Call
Bool
Remanence
Bool
%I0.0
"START"
%I0.2
"EMERGENCY_STOP"
Network 2:
Comment
%I0.1
"STOP"
%I0.2
"EMERGENCY_STOP"

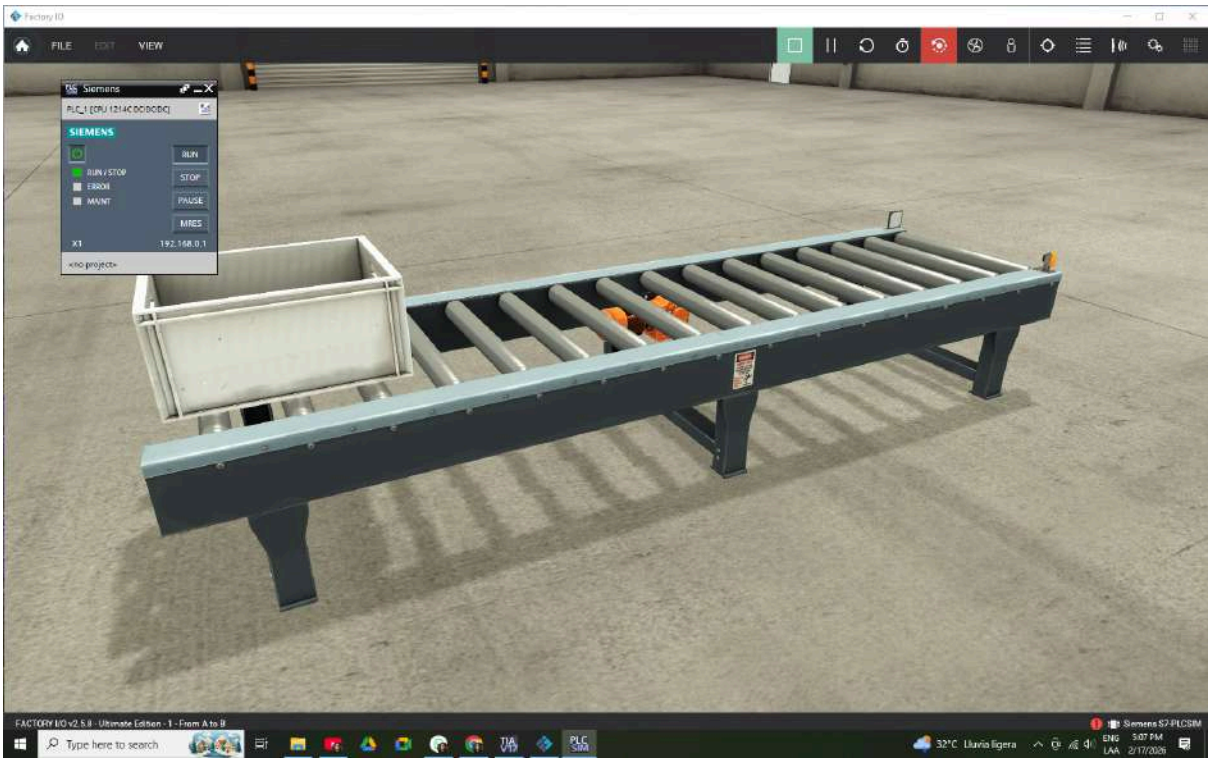

Factory IO
FILE
EDIT
VIEW
All
Search

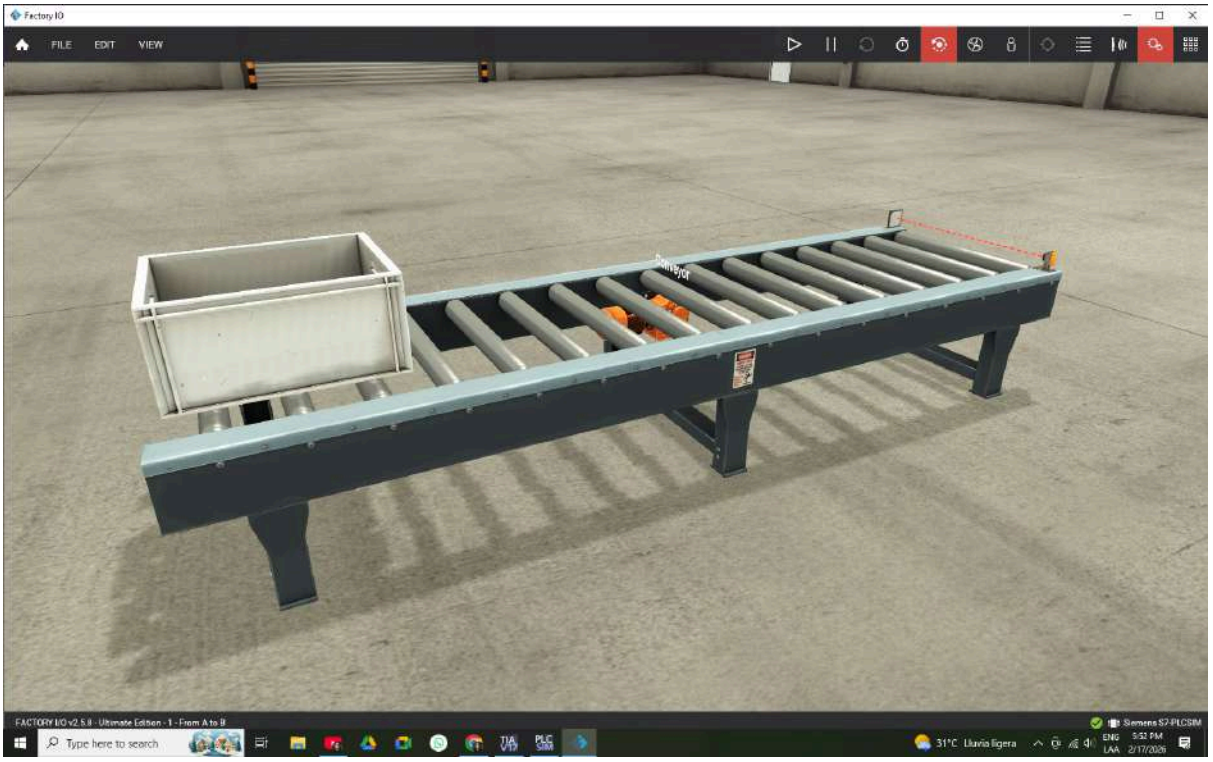


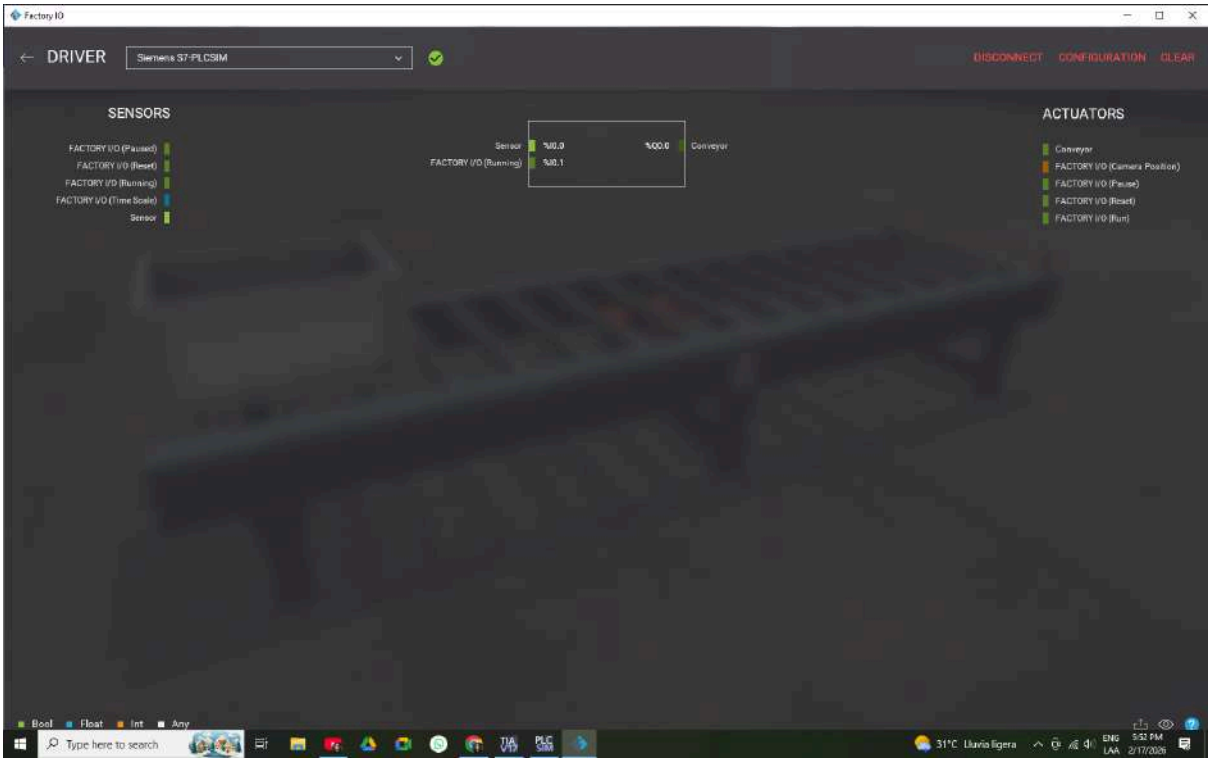
Para resolver problemas de compatibilidad con los softwares, es necesario emplear como primer medida, los templates del Factory I/O.
https://docs.factoryio.com/tutorials/siemens/setting-up-s7-plcsim-v13/#tia-portal-template-projects

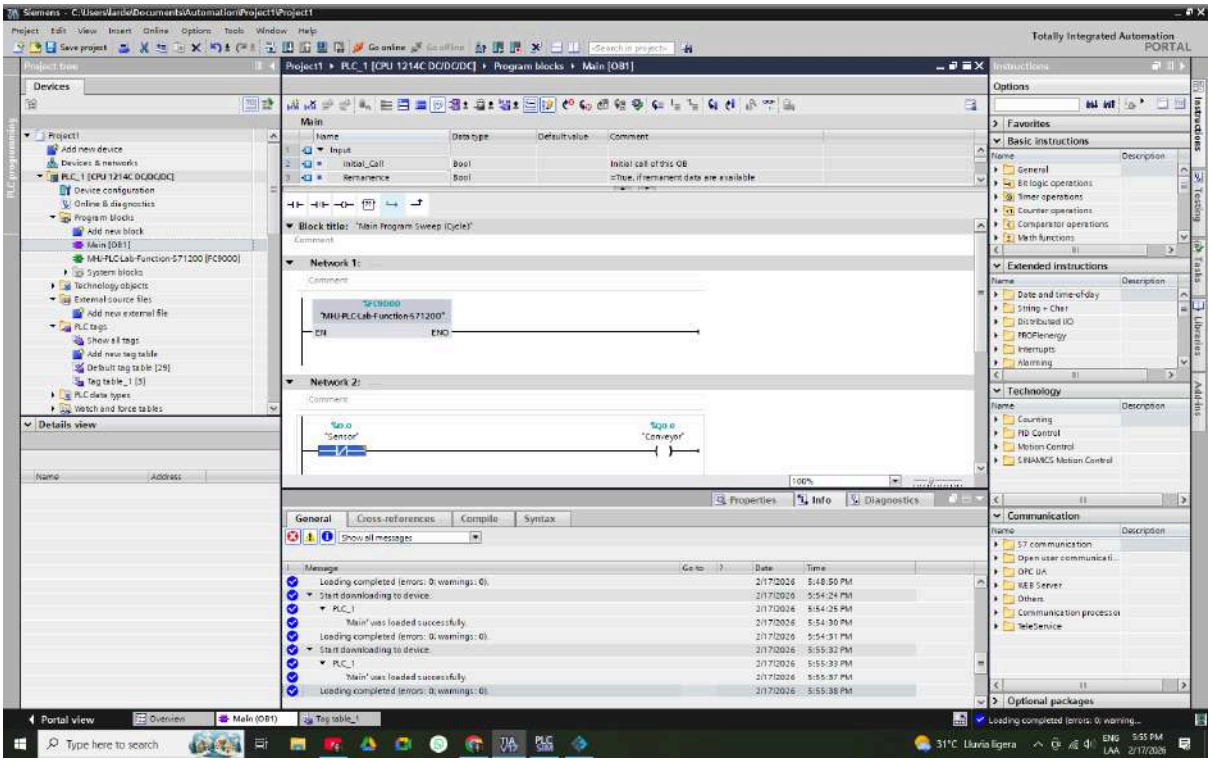


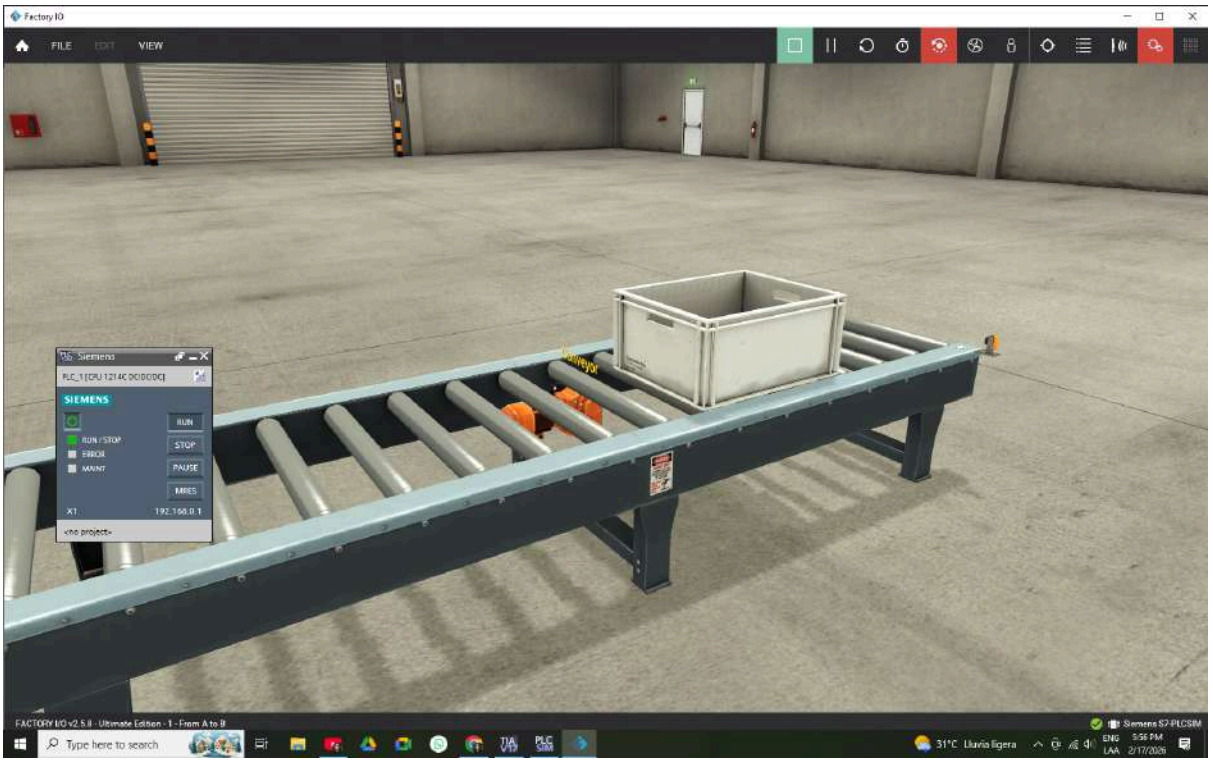

Factory IO
DRIVER
Siemens S7-PLCSIM
CONNECT
CONFIGURATION
CLEAR
SENSORS
ACTUATORS
Emergency Stop 0
FACTORY I/O (Paused)
FACTORY I/O (Reset)
FACTORY I/O (Running)
FACTORY I/O (Time Scale)
Start Button 0
Stop Button 0
Stack Light 0 (Green)
FACTORY I/O (Camera Position)
FACTORY I/O (Pause)
FACTORY I/O (Reset)
FACTORY I/O (Run)
Stack Light 0 (Red)
Stack Light 0 (Green)
Stack Light 0 (Yellow)
Start Button 0 (Light)
Stop Button 0 (Light)
Siemens
SIEMENS
RUN
STOP
PAUSE
MRES
RUN / STOP
ERROR
MAINT
192.168.0.1
<no project>
Bool
Float
Int
Any
Type here to search
32°C Lluvia ligera
ENG
LAA
5:06 PM
2/17/2026


Factory IO
FILE
EDIT
VIEW
Siemens
SIEMENS
RUN
STOP
PAUSE
MRES
RUN / STOP
ERROR
MAINT
192.168.0.1
<no project>
FACTORY I/O v2.5.8 - Ultimate Edition - 1 - From A to B
Siemens S7-PLCSIM
Type here to search
32°C Lluvia ligera
ENG
LAA
5:07 PM
2/17/2026


Factory IO
FILE
EDIT
VIEW
Conveyor
Type here to search


Factory IO
DRIVER
Siemens S7-PLCSIM
DISCONNECT
CONFIGURATION
CLEAR
SENSORS
FACTORY I/O (Paused)
FACTORY I/O (Reset)
FACTORY I/O (Running)
FACTORY I/O (Time Scale)
Sensor
Sensor
%I0.0
FACTORY I/O (Running)
%I0.1
%Q0.0
Conveyor
ACTUATORS
Conveyor
FACTORY I/O (Camera Position)
FACTORY I/O (Pause)
FACTORY I/O (Reset)
FACTORY I/O (Run)
Bool
Float
Int
Any
Type here to search


Totally Integrated Automation
PORTAL
Project1 ▸ PLC_1 [CPU 1214C DC/DC/DC] ▸ Program blocks ▸ Main [OB1]
Network 1:
Network 2:
"Sensor"
"Conveyor"


Factory IO
FILE
EDIT
VIEW
SIEMENS
RUN
STOP
PAUSE
MRES
Conveyor

Más Ejercicios

🚦 Nivel 1: Fundamentos y Lógica Digital

En este nivel, el objetivo es familiarizarte con el entorno, la conexión entre los tres softwares y la lógica de contactos más básica.

- Ejercicio 1: "Hola Mundo" - Control de una Lámpara

Objetivo: Establecer la comunicación bidireccional entre TIA Portal, PLCSIM y Factory IO.
Escenario en Factory IO: Escenario básico con un Botón (NO) y una Lámpara.
Tarea: Programa en el PLC que al presionar el botón, la lámpara se encienda. Al soltarlo, se apague.
Conceptos Clave: Lógica booleana básica (contacto NA y bobina), creación de tabla de tags y asignación de direcciones (E/S).

- Ejercicio 2: Marcha/Parada de un Motor con Retención

Objetivo: Implementar el clásico circuito "Set-Reset" o de autosostención.
Escenario en Factory IO: Un Motor o una Cinta Transportadora simple con dos pulsadores (Marcha/Parada).
Tarea: Programa el arranque y parada del motor. Asegúrate de que si se va la luz (simulación de fallo de red), el motor no vuelva a arrancar solo al recuperarse, sino que requiera un nuevo pulsador de marcha.
Conceptos Clave: Biestables (SR o RS), memoria intermedia y lógica de autosostención.

⏱️ Nivel 2: Introducción a Temporizadores y Contadores

Aquí comenzamos a gestionar eventos basados en el tiempo y en el conteo de piezas, algo esencial en automatización.

- Ejercicio 3: Encendido de una Lámpara con Retardo (Timer)

Objetivo: Usar un temporizador a la conexión (TON).
Escenario en Factory IO: Un Sensor y una Lámpara.
Tarea: Cuando el sensor detecte un objeto, la lámpara deberá encenderse, pero solo después de que el objeto haya permanecido frente al sensor durante 5 segundos continuos.
Conceptos Clave: Temporizador TON, lógica de flanco para evitar reinicios continuos.

- Ejercicio 4: Conteo de Piezas en una Banda

Objetivo: Utilizar un contador (CTU) y mostrarlo en un display virtual.
Escenario en Factory IO: Una Cinta Transportadora que arrastra piezas y un Sensor al final de la cinta.
Tarea: Cuenta el número de piezas que pasan por el sensor. Cuando se hayan contado 5 piezas, la cinta debe detenerse automáticamente. Incluye un botón externo de "Reinicio" para poner el contador a cero y reanudar la marcha.
Conceptos Clave: Contador CTU, comparadores y gestión de reset.

🏭 Nivel 3: Secuencias y Máquinas de Estados

Pasamos a controlar procesos con varios pasos secuenciales. Es la base de casi cualquier máquina automática.

- Ejercicio 5: Control de un Actuador Neumático (Doble Efecto)

Objetivo: Controlar un cilindro con finales de carrera.
Escenario en Factory IO: Un Cilindro de doble efecto con sensores de final de carrera (retraído y extendido) y una válvula 5/2 con dos bobinas (o una sola, según el modelo).
Tarea: Al presionar un botón de "Inicio", el cilindro debe avanzar. Al llegar al final de carrera de avanzado, debe retroceder automáticamente. El ciclo termina cuando el cilindro vuelve a su posición inicial.
Conceptos Clave: Secuencias, enclavamientos mecánicos (no activar avanzar y retroceder a la vez).

- Ejercicio 6: Montaje de una Tapa (Sistema Secuencial)

Objetivo: Implementar una secuencia de varios pasos coordinando actuadores.
Escenario en Factory IO: (Inspirado en proyectos educativos) Un sistema con un tope, un cilindro vertical y una ventosa .
Tarea:
Una pieza (base) llega a una estación y es detectada por un sensor.
Un tope se levanta para detenerla.
Un cilindro con una ventosa desciende, recoge una tapa y asciende.
El cilindro desciende de nuevo para colocar la tapa sobre la base.
La ventosa suelta y el cilindro asciende.
El tope baja y la pieza ensamblada sale de la estación.
Conceptos Clave: Programación estructurada, uso de pulsos (flanco) para avanzar de paso, creación de una máquina de estados sencilla.

🤖 Nivel 4: Proyectos Integrados (Aplicación Final)
Estos ejercicios integran todo lo aprendido y se asemejan a problemas reales de automatización.

- Ejercicio 7: Prensado con Control de Presión (Analógico)

Objetivo: Introducir el manejo de señales analógicas.
Escenario en Factory IO: Una prensa neumática o hidráulica con un sensor de presión analógico .
Tarea: El ciclo comienza con un botón. La prensa debe bajar y aplicar presión hasta alcanzar un valor de consigna (por ejemplo, 50 bares). Debe mantener esa presión durante 3 segundos y luego subir. Si la presión supera los 55 bares (sobrepresión), debe abortar el ciclo y subir inmediatamente por seguridad.
Conceptos Clave: Normalización y escalado de señales analógicas (instrucciones NORM_X y SCALE_X), comparadores analógicos y alarmas de seguridad.

- Ejercicio 8: Línea de Empaquetado Sencilla (Proyecto Final)

Objetivo: Abordar un proyecto multidisciplinar que simule un entorno real .
Escenario en Factory IO: Una pequeña línea con:
Una banda de alimentación (velocidad fija).

Un sensor de conteo.
Un actuador neumático de expulsión.
Una banda de salida para cajas.
Tarea: Automatiza el proceso: Las piezas entran por la banda principal. Por cada 10 piezas contadas, se debe activar el expulsor para que las empuje hacia una caja en la banda de salida. La línea debe poder pararse con un pulsador de emergencia que lo detenga todo inmediatamente.
Conceptos Clave: Integración de E/S digitales, contadores, temporizadores (para el tiempo de expulsión), gestión de emergencias y organización del código en bloques (FC para el conteo, FB para el expulsor, etc.).