

**Universidad Central “Marta Abreu” de Las Villas**

**Facultad de Ingeniería Eléctrica**

**Departamento de Automática y Sistemas Computacionales**



## **TRABAJO DE DIPLOMA**

**Prácticas de laboratorios reales con entrenador de  
autómatas de la firma Schneider**

**Autor: Joel Díaz García**

**Tutor: MSc. Samy Brito Barroso**

**Santa Clara**

**2017**

**"Año 59 de la Revolución"**

**Universidad Central “Marta Abreu” de Las Villas**

**Facultad de Ingeniería Eléctrica**

**Departamento de Automática y Sistemas Computacionales**



## **TRABAJO DE DIPLOMA**

### **Prácticas de laboratorios reales con entrenador de autómatas de la firma Schneider**

**Autor: Joel Díaz García**

E-mail: [jdgarcia@uclv.cu](mailto:jdgarcia@uclv.cu)

**Tutor: MSc. Samy Brito Barroso**

Dpto. de Automática, Facultad de Ing. Eléctrica, UCLV.

E-mail: [samyb@uclv.edu.cu](mailto:samyb@uclv.edu.cu)

**Santa Clara**

**2017**

**"Año 59 de la Revolución"**



Hago constar que el presente trabajo de diploma fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de estudios de la especialidad de Ingeniería en Automática, autorizando a que el mismo sea utilizado por la Institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos, ni publicados sin autorización de la Universidad.

---

Firma del Autor

Los abajo firmantes certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

---

Firma del Autor

---

Firma del Jefe de Departamento  
donde se defiende el trabajo

---

Firma del Responsable de  
Información Científico-Técnica

## **PENSAMIENTO**

***"El genio se hace con un 1% de talento y un 99% de trabajo"***

***Albert Einstein***

## DEDICATORIA

*A mis padres,  
por la confianza, el apoyo y la formación que me han brindado.*

*A mi novia por el amor que me brindas.*

*Dedico este trabajo principalmente a mi hermano Jhoan.*

## **AGRADECIMIENTOS**

Este proyecto significa mucho para mí, pues representa la culminación de una importante etapa de mi vida. Atrás quedan muchos años de duro esfuerzo y dedicación, que sin duda continuarán en el futuro. Durante todos estos años, muchas son las personas que me han ayudado en el día a día para poder continuar adelante. Aunque se hace imposible nombrar a todas ellas, me gustaría agradecer profundamente a todas las personas que, de alguna manera, son parte de la culminación de este proyecto.

En primer lugar, debo agradecer a mi tutor Samy Brito Barroso por la gran ayuda que me dio en este proyecto y el haber estado siempre disponible cuando lo necesitaba. Además, le agradezco a la profesora María de Carmen que nunca me dijo que no y al profesor Eduardo Izaguirre que también me ayudó.

También se me hace inevitable agradecer a los profesores que han dejado su huella a lo largo de la carrera.

## **RESUMEN**

El presente trabajo tiene la finalidad de instalar la instrumentación e implementar el software en el entrenador de autómatas de la firma Schneider Electric que se encuentra en el Laboratorio de Instrumentación y Control; y ofrecer unas guías metodológicas mediante las cuales los estudiantes pueden realizar sus prácticas de laboratorios reales, en la asignatura de Autómatas Programables. En general, se proponen prácticas de medición, adquisición de variables analógicas y digitales, comunicación, interfaz gráfica y monitoreo de las variables temperatura, presión y nivel. La temperatura se medirá en un calentador de agua, la presión en un calibrador de peso muerto y el nivel en un tanque de agua.

El entrenador cuenta con un PLC M241 tipo TM241CE40R de la firma Schneider Electric encargado de la adquisición de las variables analógicas y digitales, comunicación, etc; para la medición un sensor tipo Pt100 con transmisor SEM1503P, y dos transductores de presión tipo Cerabar T PMC131 de la firma Endress+Hauser.

El monitoreo se hace a través de una HMI tipo Magelis S5T que se encuentra en el entrenador y dicha interfaz se configura con el software Vijeo Designer.

## TABLA DE CONTENIDOS

PENSAMIENTO .....	i
DEDICATORIA .....	ii
AGRADECIMIENTOS .....	iii
RESUMEN .....	iv
INTRODUCCIÓN .....	1
CAPÍTULO 1.    REVISIÓN BIBLIOGRÁFICA.....	5
1.1    Clasificación de las maquetas de procesos.....	5
1.2    Los autómatas programables .....	6
1.2.1    Evolución de los PLC .....	6
1.2.2    Clasificación de los PLC.....	7
1.2.3    Productores principales de PLC.....	8
1.2.4    Principales softwares para la programación de PLC. ....	9
1.3    La interfaz hombre-máquina. ....	9
1.3.1    Funciones principales. ....	10
1.3.2    Software para la creación de HMI. ....	10
1.4    Relación teoría-práctica en el proceso docente.....	11
1.4.1    La práctica de laboratorio en el sistema de educación superior. ....	11
1.4.2    Clasificación de los laboratorios en la ciencia.....	13
1.5    Consideraciones Finales del Capítulo.....	15
CAPÍTULO 2.    ENTRENADOR DE LA FIRMA SCHNEIDER ELECTRIC .....	16
2.1    Elementos que componen el entrenador de autómatas de la firma Schneider.....	16
2.2    Autómata M241 tipo TM241CE40R. ....	17



2.2.1	Módulo de expansión TM4ES4.....	18
2.2.2	Módulo analógico TM3AM6G.....	19
2.3	Sensores y transductores.....	20
2.3.1	Transductor Cerabar PMC131.....	21
2.3.2	Sensor Pt100 y trasmisor SEM1503P.....	23
2.4	Conexión a la computadora.....	25
2.5	Software de programación SoMachine V4.1.....	26
2.6	El HMI Magelis STU tipo HMIS5T.....	26
2.6.1	Software Vijeo Designer V6.2 para el diseño y la configuración de HMIS5T. .....	28
2.7	Dispositivos de apoyo a las prácticas de laboratorio.....	28
2.8	Consideraciones finales del Capítulo.....	30
<b>CAPÍTULO 3. PRÁCTICAS DE LABORATORIOS REALES DE LA ASIGNATURA AUTÓMTAS PROGRAMABLES.....</b>		<b>31</b>
3.1	Procedimiento para la elaboración de las prácticas.....	31
3.2	Prácticas de laboratorios reales de Autómatas Programables.....	32
3.2.1	Práctica de laboratorio No.1.....	32
3.2.2	Práctica de laboratorio No.2.....	33
3.2.3	Práctica de laboratorio No.3.....	34
3.2.4	Práctica de laboratorio No.4.....	35
3.2.5	Práctica de laboratorio No.5.....	36
3.4	Análisis económico.....	37
3.5	Consideraciones finales del Capítulo.....	38
<b>CONCLUSIONES Y RECOMENDACIONES .....</b>		<b>39</b>
Conclusiones.....		39

Recomendaciones .....	39
REFERENCIAS BIBLIOGRÁFICAS .....	40
ANEXOS .....	43
Anexo I Componentes del M241 y descripción de ellos. ....	43
Anexo II Componentes del módulo Ethernet TM4ES4 y descripción de ellos. ....	45
Anexo III Instalación del transductor Cerabar T PMC 131 en el “Calibrador de peso muerto”. ....	46
Anexo IV Práctica No.1 Introducción al entorno del autómeta Modicon. ....	47
Anexo V Práctica No.2 Introducción al SoMachine.....	61
Anexo VI Práctica No.3 Instrucciones de aplicación. ....	76
Anexo VII Práctica No.4 Adquisición de Variable analógica. ....	87
Anexo VIII Práctica No.5 Interfaz Gráfica.....	98

## INTRODUCCIÓN

A partir del siglo XX la revolución industrial alcanza desenfrenado avance tecnológico y científico. En esta etapa comienzan a aparecer las grandes industrias productoras tales como la textil, química, alimenticia, armamentista, por citar algunos ejemplos.

Todo este desarrollo es consecuencia de una premisa fundamental: la existencia de fuentes de energía inagotable y barata. Pero su veracidad se ha puesto en entredicho con la crisis del petróleo iniciada en las últimas décadas del siglo XX.

Los sistemas productivos no son ajenos a todos estos avances. La empresa, motor del desarrollo del sector privado e incluso del sector público, se ve obligada casi siempre a incorporar las últimas tecnologías en sus procesos o de lo contrario corre el peligro de quedar obsoleta (Richmond 2009).

Con motivo del inevitable desarrollo científico-tecnológico es necesario formar profesionales capaces de desempeñarse en el campo de la producción industrial, donde los procesos son automatizados debido a la incorporación de modernas tecnologías basadas en la electrónica y la computación. La operación de estos modernos procesos de producción industrial, requiere de profesionales que los conozcan en profundidad y que sepan aplicar las competencias adquiridas para su instalación, operación y mantenimiento. Durante la formación de estos profesionales es indispensable realizar una vinculación real entre la teoría y la práctica, es por esta razón que las instituciones encargadas de la formación del personal calificado cuentan con entrenadores de autómatas para la realización de prácticas reales (Richmond 2009).

El entrenador de autómatas programables es un sistema didáctico para el estudio teórico-práctico de los PLC y sus aplicaciones. Con tal medio se posibilitan los siguientes temas de estudio (Vallduan 2014):

- Arquitectura del PLC.
- Definición de proyectos.
- Programación del PLC.
- Simulación de entradas y salidas analógicas y digitales.
- Visualización de las salidas en forma de led para las digitales y voltímetro para las analógicas.
- Conexión del PLC a módulos periféricos.

Un Controlador Lógico Programable, más conocido por sus siglas en inglés como PLC (Programmable Logic Controller), es utilizado para automatizar procesos electromecánicos (López Morocho and Navarrete Machado, 2015).

Un PLC es un ejemplo de un sistema de tiempo real, donde los resultados de salida deben ser producidos en respuesta a las condiciones de entrada dentro de un tiempo limitado, de lo contrario no se obtiene el resultado deseado. Están diseñados para múltiples señales de entrada y de salida, rangos de temperatura ampliados, inmunidad al ruido eléctrico y resistencia a la vibración y al impacto; los PLC son el elemento más importante de la automatización industrial y también de los entrenadores de autómatas dedicados a la impartición de prácticas reales (Vallejo 2014).

Son muchos los trabajos donde se usan los PLC para impartir las prácticas docentes, ejemplo es el “Empleo de Maquetas con Interface Hardware con Equipos PLC para Docencia en Automatización Naval” realizado por Julio Terrón, Carlos Corrales y Manuel J. López en el Departamento de Ingeniería de Sistemas y Automática de la Universidad de Cádiz (UCA), donde se realizaron diseños de automatismos basados en PLC con un grado de realismo que les permite probar los diseños en los sistemas reales o en maquetas realizadas para la Ingeniería Naval y Ciencias Náuticas (Terrón 2016).

Otro ejemplo es la experiencia “Plataforma para Prácticas Simuladas con el uso de Autómata como Controlador” de la autora Thuong Tran Thi en el año 2012, donde se utiliza un PLC MasterK 120S como controlador de un proceso industrial simulado en Matlab y se visualiza con Movicon (Tran Thi, 2012).

También el trabajo “Modernización del lazo de control de nivel en la industrial de tres variables”, del autor Yasiel Morera Martínez en el año 2013, donde se elabora una práctica real para el control de nivel de agua en un tanque utilizando un PLC MasterK 120S como controlador y se visualiza en LabView (Morera Martínez, 2013).

### **Situación del Problema:**

En la impartición de la asignatura Autómatas Programables de la carrera Ingeniería Automática se utilizan los autómatas como medios esenciales de enseñanza. Estos medios presentan más de diez años de explotación y una tecnología obsoleta. Por lo que a partir de la donación del entrenador de autómatas de la firma Schneider Electric que cuenta, entre otros medios, con un PLC M241 de gama media y tecnología actual, se hace imprescindible adecuarlo a las necesidades de la asignatura.

En consecuencia, con esta problemática, el objetivo de esta investigación es el siguiente:

### **Objetivo General:**

- Implementar en el entrenador de autómatas el hardware y el software necesarios para la confección de las prácticas reales de la asignatura Autómatas Programables.

### **Objetivos Específicos:**

- Identificar en la literatura especializada las principales tendencias en el diseño y modernización de los entrenadores de autómatas.
- Instalar la instrumentación adecuada para la adquisición de los datos.
- Programar el PLC M241 para la adquisición y transmisión de los datos.
- Realizar la HMI (*Human Machine Interface*) para el monitoreo de las variables en el HMISTU tipo Magelis S5T.
- Confeccionar las prácticas de laboratorios reales para la asignatura de Autómatas Programables.

Como resultados se obtienen las cinco guías metodológicas para las prácticas reales de la asignatura Autómatas Programables.

**Estructura y contenido de la tesis:**

La tesis, posterior a la introducción, incluye tres capítulos, conclusiones, recomendaciones, referencias bibliográficas y anexos. A continuación, se muestra un resumen del contenido de cada capítulo:

**CAPÍTULO 1:**

En este capítulo, titulado “Revisión Bibliográfica”, se aborda lo referido a las características de las maquetas de procesos industriales y a los entrenadores de autómatas; el empleo del PLC en la industria y en estos entrenadores para uso docente. También se brindan generalidades sobre temas relacionados con: la práctica de laboratorio en el sistema de educación superior y los sistemas HMI.

**CAPÍTULO 2:**

Titulado “Entrenador de la firma Schneider Electric”, se refiere a las características técnicas de los sensores y transductores a utilizar. A su vez, expone las particularidades del software Somachine, para la programación de PLC; y las del software Vijeo Designer, para la configuración del HMI tipo Magelis S5T. Al mismo tiempo, trata temas importantes como la instalación y la comunicación de los distintos instrumentos.

**CAPÍTULO 3:**

En el capítulo tres, titulado “Prácticas de laboratorio reales de la asignatura Autómatas Programables” se da a conocer el procedimiento para la elaboración de una práctica de laboratorio, y se proponen cinco prácticas de laboratorio reales de la asignatura Autómatas Programables y un estudio independiente que se elabora en un informe a entregar. Además se aborda la valoración económica del diseño realizado.

## **CAPÍTULO 1. REVISIÓN BIBLIOGRÁFICA**

En este capítulo se abordan las características técnicas de los sensores y transductores a utilizar. Al mismo tiempo, se exponen las particularidades del software Somachine, para la programación de PLC; y las del software Vijeo Designer, para la configuración del HMI. También se tratan la instalación y la comunicación de los distintos instrumentos.

### **1.1 Clasificación de las maquetas de procesos**

Las maquetas industriales para impartir docencia y entrenamiento pueden clasificarse en dos grupos: Plantas pilotos y Modelos educacionales (Granado 2010).

- Plantas piloto: son plantas de proceso a escala reducida. Estas plantas cuentan con instrumentación inteligente con capacidad de comunicación (sensores, actuadores) y pueden ser controladas por PLC, Sistemas de Control Distribuido (SCD) o por Tarjetas de Adquisición de Datos y PC. Las mismas permiten la realización de prácticas de gran envergadura, como proyectos de automatización con buses de campo, SCADA, etc.
- Modelo educacional: es un equipo el cual se conecta a una PC, para ser manejado desde algún software como Matlab, Simulink, Labview; y se utiliza para la medición y/o control, y para la visualización de variables fundamentales en el proceso. Con estos equipos es posible la realización de numerosos experimentos como, por ejemplo: la identificación y obtención de modelos (lineales o no-lineales), diseño de controladores, entre otros.

El entrenador de autómatas de la firma Schneider Electric no pertenece a ninguno de los dos grupos antes mencionados, ya que no incluye ningún proceso.

## 1.2 Los autómatas programables

El PLC es un elemento de estado sólido que controla elementos de salida basados en el estado de las entradas, mediante un programa desarrollado por el usuario. Se desarrollaron para reemplazar a los relés usados para control discreto.

El PLC comienza a usarse como controlador clásico por la década de los 80's, permitiendo una gran velocidad de respuesta. En la actualidad el PLC tiene gran implementación como controlador por sus abundantes prestaciones como es su capacidad de comunicación, fácil programación y resistencia a ambientes hostiles. También permite implementar controladores clásicos como el on-off, proporcional (P), proporcional derivativo (PD), proporcional integral (PI) o proporcional integral derivativo (PID) (Romero Rugel, 2014).

Los mismos poseen además capacidades adicionales inigualables y muy importantes para desempeñar mejor que ningún otro medio técnico moderno, las funciones de control automático industrial. En sus inicios fueron muy empleados para control secuencial de procesos, añadiéndosele posteriormente mayor capacidad de procesamiento y prestaciones (Izaguirre 2012).

### 1.2.1 Evolución de los PLC

Los PLC's se introducen por primera vez en la industria en la década de los 60's. La razón principal fue la necesidad de eliminar el gran costo que se producía al reemplazar el complejo sistema de control basado en relé y contactores. Entonces fue cuando *Bedford Associates* propuso algo denominado Controlador Digital Modular (Modicon, *Modular Digital Controller*) a un gran fabricante de coches. Otras compañías propusieron a la vez esquemas basados en ordenador, uno de los cuales estaba basado en el PDP-8. Entonces el MODICON 084 resulta ser el primer PLC del mundo en ser producido de manera comercial.

A mediados de los 70's las tecnologías dominantes de los PLC son las máquinas de estado secuencial y las CPU basadas en desplazamiento de bit. Los microprocesadores convencionales cedieron la potencia necesaria para resolver de forma rápida y completa la



lógica de los pequeños PLC's. Por cada modelo de microprocesador había un modelo de PLC basado en el mismo; el AMD 2903 fue el más utilizado.

Las habilidades de comunicación comenzaron a aparecer en 1973. El primer sistema fue el bus Modicon (Modbus). El PLC es capaz de dialogar con otros PLC's y en conjunto estar aislados de las máquinas que controlaban. También enviar y recibir señales de tensión variables, entrando en el mundo analógico.

En los 80's se produce un intento de estandarización de las comunicaciones con el protocolo MAP (*Manufacturing Automation Protocol*). También en este tiempo se reducen las dimensiones del PLC y se pasa a la programación simbólica a través de ordenadores personales en vez de los clásicos terminales de programación. Hoy el PLC más pequeño es del tamaño de un simple relé.

Los 90's son una década de gradual reducción en el número de nuevos protocolos, y en la modernización de las capas físicas de los protocolos más populares que sobrevivieron a los 80's. El último estándar (IEC 1131-3) intenta unificar el sistema de programación de todos los PLC en un único estándar internacional. Ahora disponemos de PLC's que pueden ser programados en diagramas de bloques, lista de instrucciones y texto estructurado al mismo tiempo (Antonio Valverde, 2014).

### **1.2.2 Clasificación de los PLC**

Desde el punto de vista de su estructura externa básica los PLC muestran dos configuraciones fundamentales (Izaguirre 2012):

- Estructura Compacta.
- Estructura Modular.

Ambas configuraciones no están determinadas por condiciones específicas, sino que dependen de varios aspectos, por ejemplo: costo, tipo de fabricante, prestaciones, dimensiones y peso, área de aplicación, etc.

#### **➤ Autómatas de Estructura Compacta.**

Se distinguen por presentar en una sola unidad todos sus elementos integrantes, es decir, la fuente de alimentación, CPU, memorias, salida para red de comunicación y un conjunto de

entradas y salidas que en general suelen ser digitales y no pasan de 20 en su totalidad. Un conector puerto serie permite enchufar el terminal de programación a una PC que se usa para realizar la programación y descargar el programa de aplicación a la memoria del autómata. Debido a sus pequeñas dimensiones y peso, el montaje mediante carril DIN o tornillos suele ser muy sencillo y rápido.

➤ **Autómatas de Estructura Modular.**

Como su nombre lo indica su estructura está compuesta por módulos que realizan funciones específicas. Entre ellos podemos mencionar los módulos de CPU's con memorias de usuario o de programa, módulo fuente de alimentación, módulos de E/S digitales y/o analógicas, módulos PID, módulos para la comunicación (Ethernet, por ejemplo). La unidad de programación se une al módulo CPU mediante cable y conector.

Los módulos para PLC se comercializan en una amplia gama y diversidad de opciones, las ofertas varían respecto a características, función, cantidad y tipo de E/S.

Los módulos analógicos más comunes poseen desde 2 hasta 8 canales para entradas que cubren los rangos desde 0 a 10 volts, 4 – 20 mA, 0 – 20 mA, así como los destinados a funciones específicas como por ejemplo, para la conexión directa de Pt-100, termopares, encoders, etc.

### 1.2.3 Productores principales de PLC

Son varias las compañías que alcanzan un alto desarrollo en la producción y comercialización de PLC en el mundo a lo largo de estas cinco décadas de historia. En nuestro país se trabaja con autómatas de varias de estas compañías. En la tabla 1.1 se relacionan las principales compañías fabricantes de PLC y su producto líder (Rosero et al., 2016).

Tabla 1.1 Compañías principales en desarrollo de autómatas

Compañía	PLC	País productor
Schneider Electric	TSX, Modicon	Francia
Siemens	S7	Alemania
OMRON	CPL	Japón
LG /LS	MK	Corea del Sur

ABB	AC	Estados Unidos
Mitsubishi	FX	Japón
Allen Bradley	SLC	Estados Unidos
Semaphore	TBox-MS	Malasia

#### 1.2.4 Principales softwares para la programación de PLC

El software para programar los PLC's se compone de un sistema operativo y un entorno de programación, capaz de soportar uno o más lenguajes de programación. Los componentes adicionales del software pueden ser: programas para diagnosticar y encontrar fallos, programar de visualización, comunicación, etc. Muy a menudo, estos programas son parte del sistema operativo. El sistema operativo es de tipo convencional como Windows, Linux, Mac OS etc. Algunos de estos software son (Schneider Electric, 2014; Vallejo 2014):

- TwidoSuite: Software de programación diseñado para asistir el desarrollo de proyectos que utilizan PLC Twido.
- XPSMFWIN: Software de programación para autómatas de seguridad XPSMF.
- Unity Pro: Software de programación depuración y operación para PLCs Modicon Quantum, Premium y M340.
- STEP 7-Micro/WIN: Software para programación de PLC Simatic S7-XX de la empresa Siemens.
- ProWORX 32: Software de programación para gama antigua de los PLCs Modicon
- PL7: Software de programación para los PLC TSX Micro y Premium.
- SoMachine: permite programar entre otras cosas la gama de PLC Modicon y poner en funcionamiento una amplia gama de elementos de *Flexible Machine Control* de Schneider Electric.

### 1.3 La interfaz hombre-máquina

Una HMI consiste en un software de aplicación diseñado de forma especial para ejecutarse sobre ordenadores destinados a la supervisión remota de instalaciones y permitir al operador conocer el estado del desempeño de la variable de la planta (Fuentes Revilla, 2015). Estos

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

### **1.3.1 Funciones principales**

La mejor interacción humano-máquina a través de una adecuada interfaz de usuario es aquella que brinde tanto comodidad, como eficiencia. Para lograr estos factores las principales funciones que brinda la HMI son (Campaña et al., 2016)

- Puesta en marcha y apagado.
- Control de las funciones manipulables del equipo.
- Manipulación de archivos y directorios.
- Herramientas de desarrollo de aplicaciones.
- Comunicación con otros sistemas.
- Información de estado.
- Configuración de la propia interfaz y entorno.
- Intercambio de datos entre aplicaciones.
- Control de acceso.
- Sistema de ayuda interactivo.

### **1.3.2 Software para la creación de HMI**

Lenguajes de programación como Visual C++ o Visual Basic, son utilizados para desarrollar HMI a medida del usuario. Una vez generado el software, el usuario no tiene posibilidad de reprogramarlo, sólo podrá hacerlo si tiene la llave del software.

Este software debe estar en constante evolución para poder satisfacer las necesidades que se detectan en el campo industrial. Su utilización se da en aquellas instalaciones donde es necesario realizar control, supervisión, adquisición de datos, generación de alarmas y reportes, representación de variables en la interfaz gráfica de forma clara y concisa.

Para la supervisión a nivel mundial se utilizan software tales como (Portilla et al., 2015):

- Monitor Pro, de Schneider Electric.
- Movicon X2, de Progea
- SCADA InTouch, de Logitek
- SYSMAC SCS, de Omron.
- WinCC, de Siemens.
- Vijeo Designer, de Schneider Electric.

#### **1.4 Relación teoría-práctica en el proceso docente**

Con frecuencia se habla de la relación teoría-práctica, de la coherencia pedagógica entre el pensar y el hacer. La teoría constituye un conjunto de leyes, enunciados e hipótesis que conforman una serie de conocimientos científicos, sistematizados y organizados, que permiten derivar a partir de estos fundamentos reglas de actuación. Del mismo modo la práctica se entiende como una praxis que implica conocimientos para fines determinados; durante la misma se aplica lo aprendido en la teoría, con el saber hacer (Álvarez 2012).

La teoría ilumina, fecunda, explica y condiciona la práctica; y esta a su vez, proyecta, refleja y desarrolla la teoría; por lo que es necesario saber que una va ligada a la otra; siendo la teoría el principal motor de las acciones y del sentido de la práctica profesional, de esta depende la coherencia educativa, la mejora escolar y el desarrollo docente. La relación teoría-práctica es la que ayuda a desarrollar las capacidades buscadas en los estudiantes. La práctica sirve a la teoría científica, por lo que se centra en actividades verificativas, experimentos a prueba de errores y manipulación de equipos (Clemente 2007).

##### **1.4.1 La práctica de laboratorio en el sistema de educación superior**

Según (Anon., 2007), las formas organizativas fundamentales del proceso docente educativo, son:

- La clase
- La práctica de estudio
- La práctica laboral
- El trabajo investigativo de los estudiantes
- La auto preparación de los estudiantes

- La consulta
- La tutoría

La clase como forma organizativa del proceso docente educativo, tiene como objetivos la adquisición de conocimientos, el desarrollo de habilidades y la formación de valores e intereses cognoscitivos y profesionales en los estudiantes, mediante la realización de actividades de carácter esencialmente académico. Esta se clasifica sobre la base de los objetivos que se deben alcanzar, en cada modalidad de estudio el profesor debe utilizar adecuadamente las posibilidades que brinda cada una, para contribuir al logro de los objetivos formulados en el programa de la asignatura, siendo sus principales tipos: la conferencia, la clase práctica, el seminario, la clase encuentro, la práctica de laboratorio y el taller (Abreu 2013).

La práctica de laboratorio en particular se define como el tipo de clase que tiene como objetivos que los estudiantes adquieran las habilidades propias de los métodos y técnicas de trabajo de la investigación científica, que amplíen, profundicen, consoliden, generalicen y comprueben los fundamentos teóricos de la disciplina mediante la experimentación, empleando para ello los medios necesarios. Como norma, en este tipo de clase, es imprescindible garantizar el trabajo individual de los estudiantes para la ejecución de las tareas previstas. En la práctica de laboratorio se profundizan los principios teóricos de la asignatura y la asimilación de los métodos y hábitos de trabajo experimental, con lo que se logra vincular la teoría con la práctica. El estudiante se enfrenta directamente relacionado con la solución de problemas de una ciencia particular. La misma está estructurada por las siguientes etapas, las que se describen en (Cañedo 2009).

- Preparación previa a la práctica
- Ejecución de la práctica
- Conclusiones de la práctica

La preparación previa se desarrolla fundamentalmente sobre la base del estudio teórico orientado por el profesor como fundamento de la práctica, así como el estudio de las técnicas y procedimientos de los experimentos correspondientes, teniendo en cuenta las etapas del proceso enseñanza-aprendizaje (Cañedo 2009):

- Motivación

- Orientación
- Ejecución
- Evaluación

Además, resulta necesario, establecer una secuencia de acciones que faciliten la orientación por parte del profesor, para la ejecución de la práctica de laboratorio, entre los que se encuentran (Cañedo 2009):

- Orientación de los objetivos y las tareas fundamentales a desarrollar y las técnicas operatorias básicas que se utilizarán.
- Distribución de materiales.
- Trabajo independiente de los estudiantes.
- Discusión colectiva de los resultados obtenidos.

Durante el desarrollo o ejecución de la práctica, es característico el trabajo de los estudiantes con el material de laboratorio (utensilios, instrumentos, aparatos, y reactivos), el empleo de las técnicas y métodos propios de cada actividad. Al concluir, el estudiante deberá analizar los resultados de las observaciones y experimentos realizados, llegando a conclusiones y generalizaciones obtenidas durante el desarrollo de las prácticas (González 2012).

#### **1.4.2 Clasificación de los laboratorios en la ciencia**

Toda actividad sistemática y creadora encaminada a aumentar el caudal de los conocimientos científicos y a aplicarlos en función de lograr un elevado desarrollo tecnológico se denomina actividad de investigación científica. La ciencia es una actividad eminentemente práctica, además de teórica, lo cual hace que en su enseñanza el laboratorio sea un elemento indispensable.

La práctica de laboratorio es una de las actividades más usadas e importantes dentro de la enseñanza aprendizaje de las ciencias, convirtiéndose en un medio práctico que permite al estudiante relacionar la realidad con la teoría desde un manejo conceptual y procedimental de la temática; desarrollar habilidades y destrezas para razonar, concretar pensamiento crítico y creativo (Stefania 2011).

Los laboratorios en la ciencia se clasifican en función de dos criterios, según la forma de acceder a los recursos para los propósitos de experimentación, puede ser local o remota y teniendo en cuenta la naturaleza del sistema físico, real o virtual. De la combinación de estos

dos criterios se obtienen cuatro clases de entornos muy diferentes, pero que abarcan todas las formas de experimentación posible, de esta manera quedan clasificados según la tabla 1.2 (Calvo 2008).

Tabla 1.2 Clasificación de los laboratorios en la ciencia

<b>Acceso</b>	<b>Real</b>	<b>Virtual</b>
<b>Local</b>	Representa el laboratorio como lo conocemos, en el que el estudiante se sitúa frente a los medios técnicos con una guía para proceder a la realización de las prácticas correspondientes.	Todo el entorno de trabajo es software y la interfaz de experimentación opera sobre un sistema simulado, virtual e inexistente físicamente que reside en el mismo ordenador que la interfaz.
<b>Remoto</b>	Constituye el acceso al equipamiento de un laboratorio real a través de una red. El usuario opera y controla de forma remota sistemas reales mediante una interfaz de experimentación que se ejecuta en un ordenador conectado a una red.	Esta forma de experimentación es similar al entorno Remoto-Virtual, en cuanto al acceso pero el sistema real se sustituye por un modelo, por lo que el estudiante trabaja con una interfaz de experimentación sobre un sistema virtual accesible a través de Internet.

Esta clasificación proporciona una guía clara de los distintos tipos de entornos posibles. Sin embargo, un sistema de experimentación completo no debería estar restringido a la utilización de uno de estos tipos solamente, debería atender a los cuatro tipos descritos anteriormente ya que un usuario podría experimentar sobre un sistema real o simulado de forma local o remota (Calvo 2008).

La utilización de los laboratorios reales en la carrera Ingeniería en Automática, específicamente los relacionados con la asignatura Automatas Programables, contribuyen al desarrollo de habilidades, proporcionando los conceptos y conocimientos básicos para la configuración de los PLC, la solución de problemas y la supervisión.

Disponer de laboratorios reales, con las características apropiadas se trata de plantear ejercicios que utilicen directamente equipamiento e instalaciones reales. Este tipo de



laboratorio resulta más motivador, permitiendo hacer uso de herramientas reales, como los componentes del entrenador de autómatas de la firma Schneider Electric.

No se puede negar que el trabajo práctico en el laboratorio proporciona la experimentación y el descubrimiento lo que evita el concepto de resultado correcto, que se tiene cuando se aprende de manera teórica. Pero el uso del laboratorio, requiere de tiempo adicional al de una clase convencional, para descubrir y aprender de los propios errores (Djelil 2006).

El laboratorio es el elemento y complemento más distintivo de la educación científica, tiene gran relevancia en el proceso de formación y en éste se puede conocer al estudiante con las habilidades que posee para desarrollar las actividades con la finalidad de consolidar y ganar conocimientos y actitudes adquiridas en el aula de clase (Rosario 2013).

### **1.5 Consideraciones Finales del Capítulo**

La creciente aplicabilidad de los entrenadores de autómatas está determinada por su utilidad en la preparación de los estudiantes de las carreras con perfil técnico. Los entrenadores de autómatas están compuestos en su mayoría por una CPU de prestaciones medias, módulos de comunicación, módulos analógicos y una interfaz hombre-máquina.

La práctica de laboratorio profundiza en los principios teóricos de la asignatura con lo que se logra vincular la teoría con la práctica. La realización de prácticas reales con el entrenador de autómatas programables ofrece ventajas por su semejanza con los procesos industriales y el alto grado de dificultad que se puede desarrollar en las mismas.

## **CAPÍTULO 2. ENTRENADOR DE LA FIRMA SCHNEIDER ELECTRIC**

En este capítulo se aborda lo referente a los sensores y transductores que forman parte del entrenador la firma Schneider Electric. Al mismo tiempo, se exponen las características del software SoMachine para la programación del autómatas M241. Además, se tratan las características del software Vijeo Designer para la programación del HMI tipo Magelis S5T como sistema de supervisión, así como los procesos donde se utiliza el entrenador.

### **2.1 Elementos que componen el entrenador de autómatas de la firma Schneider**

El entrenador de autómatas de la firma Schneider Electric se muestra en la figura 2.1 y está constituida por un PLC M241 tipo TM241CE40R, un sensor tipo Pt100 con transmisor SEM1503P, y dos transductores tipo Cerabar T PMC131. Esto posibilita la medición de tres variables, las cuales son: temperatura, presión y nivel; esta última se obtiene de la medición de presión al conocer que 100mbar corresponde a un metro de columna de agua. El entrenador también cuenta con un HMI tipo Magelis S5T para la supervisión de variables.



Figura 2.1 Entrenador de autómatas de la firma Schneider Electric.

## 2.2 Autómata M241 tipo TM241CE40R.

En la figura 2.2 se muestra el M241 tipo TM241CE40R de la firma Schneider Electric.



Figura 2.2 Autómata M241 tipo TM241CE40R.

### Características técnicas (Schneider Electric, 2014):

El autómata M241 tipo TM241CE40R presenta:

- ✓ 24 entradas digitales
  - 8 entradas rápidas
  - 16 entradas normales
- ✓ 16 salidas digitales
  - 4 salidas rápidas
  - 12 salidas de relé (2A)
- ✓ Puerto de comunicación
  - puertos de línea serie
  - 1 puerto Ethernet con protocolos Modbus TCP (cliente/servidor), Ethernet IP (adaptador), UDP, TCP.
  - 1 puerto de programación con un conector USB mini-B incorporado en los controladores M241, está dedicado a la comunicación con un PC equipado con SoMachine.
- ✓ Puertos serie incorporados.
- ✓ Fuente de alimentación es de 24V CC o de 100-240V CA.
- ✓ Memoria RAM de 64 Mbytes.

- ✓ Memoria Flash de 128 MB para guardar el programa y los datos en caso de corte de electricidad.
- ✓ Velocidad de ejecución: 22 ns para instrucciones booleanas.
- ✓ Consumo máximo 45 W.

En el Anexo I se muestran los componentes del M241 y una descripción de ellos.

### 2.2.1 Módulo de expansión TM4ES4

En la figura 2.3 se muestra el Módulo de Switch Ethernet TM4ES4.



Figura 2.3 Módulo de Switch Ethernet TM4ES4.

El módulo TM4ES4 es un switch de 4 puertos Ethernet con los siguientes protocolos: Modbus TCP (cliente/servidor), Ethernet IP (adaptador), UDP, TCP y SNMP. Está preparado para ser utilizado conectándolo al bus de comunicación de los controladores M241. Este módulo se utiliza para añadir la función Ethernet a los controladores TM241C24... y TM241C40... sin puerto Ethernet incorporado mientras que ofrece la funcionalidad adicional de un switch Ethernet.

Reglas de asociación:

Hasta tres módulos de comunicación se pueden añadir en la parte izquierda de los controladores lógicos M241 para aumentar sus posibilidades de conexión a redes Ethernet y Profibus.

Cuando está conectado a controladores lógicos con puerto Ethernet incorporado tipo TM241CE24... y TM241CE40... es un switch de cuatro puertos independiente: la

comunicación entre el módulo TM4ES4 y el controlador Modicon M241 no se realiza de forma automática por el conector de bus.

En la Anexo II se muestra los componentes del Módulo Ethernet TM4ES4 y una descripción de ellos.

### 2.2.2 Módulo analógico TM3AM6G.

En la figura 2.4 se puede observar el módulo de E/S analógico TM3AM6/ TM3AM6G.



Figura 2.4 Módulo de E/S analógico TM3AM6/ TM3AM6G

#### Características técnicas:

- ✓ 4 entradas analógicas: I0, I1, I2 e I3.
- ✓ 2 salidas analógicas: Q0 y Q1.
- ✓ Compatible con Modicon M221, Modicon M241, Modicon M251.
- ✓ Tipo de entrada analógica
  - corriente, rango de entrada: 0...20 mA
  - corriente, rango de entrada: 4...20 mA
  - tensión, rango de entrada: - 10...10 V
  - tensión, rango de entrada: 0...10 V
- ✓ Tipo de salida analógica.
  - Tensión, rango de entrada: - 10...10 V
  - Tensión, rango de entrada: 0...10 V
  - Corriente, rango de entrada: 0...20 mA
  - Corriente, rango de entrada: 4...20 mA
- ✓ Voltaje de la fuente: 24 VDC
- ✓ Número de canales: 6

- ✓ Familia de la marca de fábrica: Modicon
- ✓ Tiempo de estabilización: 1ms.

### Diagrama de cableado.

En la figura 2.5 se representan las conexiones entre las entradas y salidas, los sensores y los actuadores, así como sus líneas comunes:

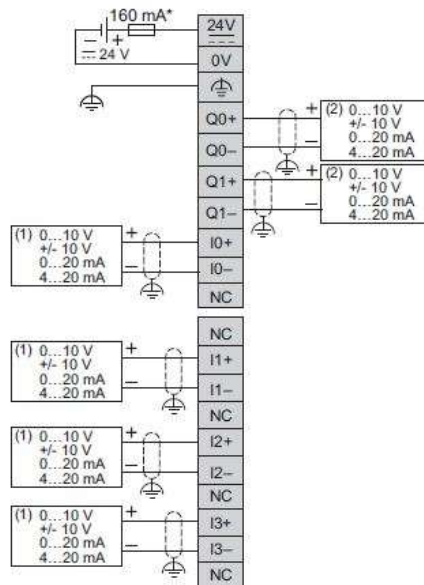


Figura 2.5 Diagrama de cableado para dispositivos de E/S analógicas, donde (1) indica los dispositivos de salida analógica de tensión y corriente y (2) indica los dispositivos de entrada analógica de tensión y corriente.

### **2.3 Sensores y transductores.**

Un transductor es un dispositivo que proporciona una salida, por ejemplo eléctrica, en respuesta a una magnitud física, propiedad o condición específica que se desea medir. El nombre del transductor ya nos indica cual es la transformación que realiza (por ejemplo: electromecánica, transforma una señal eléctrica en mecánica o viceversa)

Un sensor, propiamente hablando, es la parte del elemento primario que reacciona de alguna forma sensible a una propiedad física relacionada con la variable que se quiere medir y cuya respuesta se encuentra en una forma más conveniente, de modo que se utiliza como la entrada al resto del sistema. En la mayoría de los casos esto conlleva a la conversión de la cantidad física que se desea medir en una variable eléctrica, como resistencia, capacitancia,

inductancia, voltaje o corriente, que puede medirse con mucha precisión y en una relación predecible. El transductor, por lo tanto, es el conjunto formado por un sensor y un transmisor que acondiciona la señal a alguna forma más adecuada para el sistema de control.

La selección de un sensor para un proceso en particular requiere considerar la naturaleza de la medición. Las razones para seleccionar un equipo u otro varían.

Para la operación del controlador la linealidad de la medición es muy importante, especial en los sistemas de control analógico. Por simplicidad de los modelos, los sistemas de control se diseñan suponiendo que la variación dentro de este ámbito se representa lineal a la variable medida. Sin embargo, uno de los mayores problemas en la operación de medición se debe al hecho de que la salida de un sensor a menudo varía de forma no-lineal con la variable que se mide y esta no-linealidad puede producir efectos indeseados en el sistema bajo control. En el control analógico es muy difícil compensar el problema, aunque existen circuitos especiales para linealizar la respuesta del sensor.

En los sistemas de control digitales dicha linealización es más fácil de realizar, mediante un programa después de que los datos se ingresan (Creus 2010).

### **2.3.1 Transductor Cerabar PMC131.**

El transductor Cerabar PMC131, que forma parte del entrenador y se muestra en la figura 2.6, es el encargado de medir la presión en el calibrador de peso muerto y en el tanque de agua. Conociendo que 100mBar corresponde a 1 metro de columna de agua se puede medir el nivel del tanque (Creus 2010). El transductor de presión está dotado de un sensor capacitivo de cerámica que es muy preciso y además muy resistente a la corrosión y abrasión. (Endress+Hauser, 2011).



Figura 2.6 Sensor Cerabar T PMC131

#### Características técnicas del transductor Cerabar PMC131 para la variable presión:

El transductor Cerabar T PMC131 para la medición de la presión presenta las siguientes características (Endress+Hauser, 2011):

- Señal de salida: 4 a 20mA
- Alimentación de: 12 a 30V
- Rango de presión: 0 a 2Bar
- Potencia máxima: 6W
- Temperatura de trabajo: -20 a 100 °C

#### Comunicación Cerabar T PMC131 a PLC.

En la figura 2.7 se puede observar como se instala el transductor Cerabar al módulo de E/S analógicos TM3AM6.

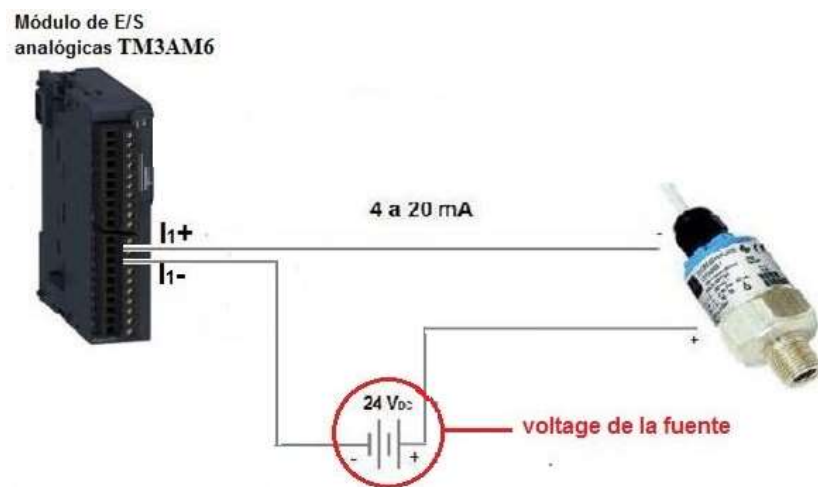


Figura 2.7 Conexión del sensor Cerabar T PMC131 al módulo del PLC.

#### Características técnicas del transductor Cerabar T para la medición de nivel:

Este transductor y el anterior tienen las mismas características, solo se diferencian en su rango de medida.

- Rango de presión: 0 a 100mBar

#### Comunicación Cerabar T PMC131 a PLC.

En el caso de la comunicación del sensor con el módulo analógico del PLC es igual al que se explica en la figura 2.7, la diferencia es que los cables de la señal de entrada al módulo se conectan por la entrada analógica I2.



### 2.3.2 Sensor Pt100 y trasmisor SEM1503P.

Los Pt100 pueden entregar precisiones de una décima de grado con la ventaja de que no se descompone entregando lecturas erróneas, sino que en la mayoría de los casos se abre, con lo cual el dispositivo medidor detecta inmediatamente la falla del sensor y da un aviso. Este comportamiento es una gran ventaja en muchas aplicaciones como en cámaras frigoríficas donde una desviación no detectada de la temperatura puede producir algún daño grave.

El entrenador de la firma Schneider Electric usa la Pt100 para la medición de la variable temperatura.

#### Características técnicas del sensor Pt100 (Endress+Hauser, 2011):

En la figura 2.8 se puede observar el sensor Pt100 que consiste en un alambre de platino que a 0°C tiene 100 ohms y que al aumentar la temperatura aumenta su resistencia eléctrica. Otras de sus principales características son:

- Elemento cilíndrico robusto
- Coeficiente: 0.00385
- Rango de -50 a 400°C
- Permite conexión de 2 ó 3 hilos.



Figura 2.8 Sensor Pt100 en el entrenador de la firma Schneider.

#### Características técnicas del transmisor SEM1503P:

En la figura 2.9 se observa el transmisor de temperatura SEM1503P, que es el encargado de proporcionar una salida de 4-20 mA. Además posee otras características técnicas como son:

- Alimentación 10 a 30V.
- Corriente de excitación 2 mA máximo.
- Tiempo de respuesta 100 ms.

- Temperatura de trabajo 50°C.



Figura 2.9 Transmisor de temperatura SEM1503P.

En la figura 2.10 se observa el tipo de conexión de tres hilos de la Pt100 al transmisor de temperatura SEM1503P.



Figura 2.10 Conexión de tres hilos de la Pt100 al transmisor de temperatura SEM1503P

La figura 2.11 muestra la conexión del transmisor de temperatura SEM1503P al módulo analógico para obtener en el PLC M241 la señal de 4 a 20mA del transmisor.



Figura 2.11 Conexión del transmisor SEM1503P al módulo analógico TM3AM6/G.

## 2.4 Conexión a la computadora.

Para transferir, ejecutar y monitorizar las aplicaciones, se conecta el controlador a un equipo que tenga instalado SoMachine mediante un cable USB o una conexión Ethernet.

### ✓ Conexión con puerto USB mini-B

En la figura 2.12 se puede observar una de las vías de conexión del M241 a la computadora mediante puerto USB mini-B. Este puerto de programación se utiliza para comunicar el software SoMachine con el autómatas. Esta conexión es adecuada para las actualizaciones rápidas del programa o las conexiones de corta duración para realizar el mantenimiento e inspección de los valores de los datos. No es adecuada para las conexiones a largo plazo, como la puesta en marcha o la supervisión.

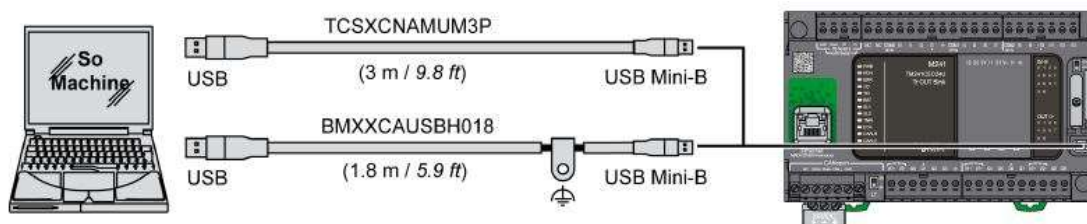


Figura 2.12 Conexión USB entre el autómatas M241 y la computadora.

✓ Conexión al puerto Ethernet

También puede conectar el controlador a un PC mediante un cable Ethernet, como se muestra en la figura 2.13.

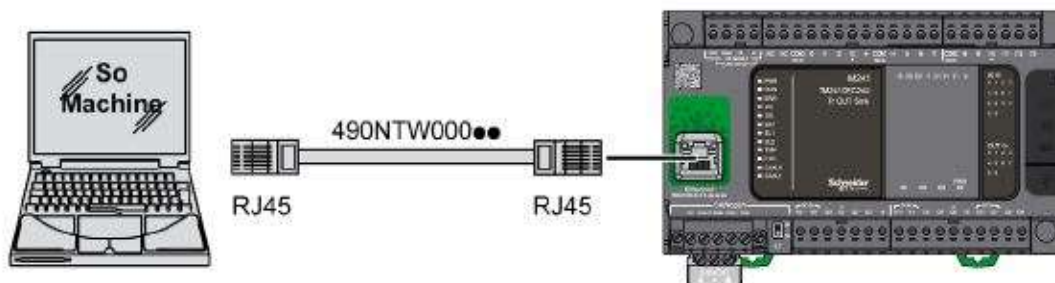


Figura 2.13 Conexión Ethernet entre el automático M241 y la computadora.

## 2.5 Software de programación SoMachine V4.1.

Los paquetes de software para la programación de autómatas convierten una computadora personal en un equipo de programación específico, aprovechando sus potentes recursos de interfaz con otros sistemas. Esta opción (computadora personal + software) constituye, junto con las consolas, prácticamente la totalidad de los equipos de programación utilizados por los programadores de autómatas (Ferreira 2005).

El software SoMachine V4.1 es una solución de software profesional, eficiente y abierto para desarrollar, configurar y poner en funcionamiento las funciones lógicas, control de motores, HMI y trabajo con redes relacionadas. SoMachine permite programar entre otras cosas la gama de PLC Modicon y poner en funcionamiento una amplia gama de elementos de la firma Schneider Electric.

El procedimiento básico a seguir para el uso del software SoMachine V4.1 es el siguiente:

- Creación de un nuevo proyecto.
- Selección y configuración del hardware a emplear.
- Desarrollo del programa. Creación del nuevo POU (Unidad Organizativa del programa).
- Simulación y/o descarga del programa al automático y/o al HMI.

## 2.6 El HMI Magelis STU tipo HMIS5T.

En la figura 2.14 se muestra la Magelis Serie HMISTU tipo HMIS5T.



Figura 2.14 Magelis Serie HMISTU tipo HMIS5T.

Características técnicas:

- ✓ Software de configuración fácil y potente, Vijeo Designer.
- ✓ Alimentación de 24 VDC.
- ✓ Puerto serie RJ45, RS485/RS232 multiprotocolo.
- ✓ Puerto mini-USB, lista para usar memoria USB, teclado.
- ✓ Pantalla a color de alta definición TFT – QVGA.
- ✓ Tamaño de pantalla 3.5”
- ✓ Acceso remoto vía el explorador y envío de correo electrónico.

En la figura 2.15 se muestra la conexión media Ethernet del HMI con el M241.



Figura 2.15 Conexión Ethernet del HMIS5T con el Autómata M241.

### **2.6.1 Software Vijeo Designer V6.2 para el diseño y la configuración de HMIS5T.**

Vijeo Designer es una aplicación de software de última generación con la que el usuario puede crear paneles de operadores y configurar parámetros operativos para dispositivos de la interfaz hombre-máquina. Este programa proporciona todas las herramientas necesarias para el diseño de un proyecto HMI, desde la adquisición de datos hasta la creación y la visualización de sinopsis animadas.

➤ **Funciones del Vijeo Designer.**

**1. Conectividad con varios autómatas.**

El usuario puede configurar un panel HMI con vistas a comunicarse de forma simultánea con varios dispositivos de otros fabricantes.

**2. Creación de pantallas HMI.**

Permite crear pantallas dinámicas para el panel HMI. Combina diferentes funciones en una aplicación sencilla, como objetos en movimiento, indicadores de nivel, indicadores de inicio/parada y conmutadores. La utilización de símbolos animados puede emplearse para crear y editar una pantalla gráfica de forma sencilla.

**3. Secuencias de comandos.**

Proporciona la función de secuencias de comandos, de manera que el usuario puede reutilizar componentes o frases completas en otros proyectos a partir de una aplicación.

**4. Mensajes en varios idiomas.**

Puede almacenar, en una misma aplicación, mensajes de alarma u objetos de texto en más de diez idiomas diferentes. Basta realizar una selección desde la pantalla de animación para cambiar la visualización al idioma seleccionado.

**5. Edición de variables a partir de otras aplicaciones.**

Puede importar y exportar variables y ajustes como archivos CSV. Asimismo, las variables que se crean en Vijeo Designer pueden exportarse a otras aplicaciones.

## **2.7 Dispositivos de apoyo a las prácticas de laboratorio**

La temperatura, como se muestra en la figura 2.16, se mide en el dispositivo “Baño de María”. En este dispositivo se vierte agua y se comienza conecta la resistencia. El agua comienza a

calentarse hasta alcanzar una temperatura determinada por el usuario mediante un selector, siempre igual o menor a los 100°C. La medición de la temperatura del agua se hace colocando en su interior la Pt100.



Figura 2.16 Dispositivo “Baño de María”.

La presión se mide en el “Calibrador de peso muerto” que se muestra en la figura 2.17. Este dispositivo está compuesto por un recipiente lleno de aceite al cual se le ejerce una fuerza que hace circular este líquido por una tubería que tiene dos extremos y en uno de ellos se coloca el transductor de presión Cerabar T PMC 131 con rango de 0 a 2Bar; la presión en ambos extremos es la misma. Un extremo se toma como referencia o patrón y el otro para calibrar y se realiza con la finalidad de calibrar cualquier dispositivo manométrico (siempre que la entrada sea del diámetro indicado).

La medición de la presión desde el entrenador de procesos se muestra en el ANEXO III.



Figura 2.17 “Calibrador de peso muerto” donde se mide la presión.



El nivel se mide en el tanque de agua que se muestra en la figura 2.18 con el otro transductor de presión Cerabar T PMC 131 con rango de 0 a 100mBar.



Figura 2.18 Tanque de agua donde se mide el nivel.

El sensor se instala en el lado inferior a 6 cm del fondo, para evitar que se acumule suciedad en el mismo y provoque falsas mediciones.

## 2.8 Consideraciones finales del Capítulo.

El entrenador de autómatas programables de la firma Scheneider Electric está formado por un autómata M241 y sus módulos correspondientes para la comunicación, la obtención de variables analógicas, la protección de motores y una interfaz hombre máquina. La conexión de sensores al módulo analógico permite la medición de magnitudes como la temperatura, la presión y el nivel.



## **CAPÍTULO 3. PRÁCTICAS DE LABORATORIOS REALES DE LA ASIGNATURA AUTÓMTAS PROGRAMABLES**

En el presente capítulo se exponen las bases teórico-metodológicas del proceso docente educativo en la asignatura de Autómatas Programables, haciendo énfasis en la importancia de las prácticas de laboratorio reales como una de las formas fundamentales de aprendizaje significativo y se exponen los laboratorios diseñados.

### **3.1 Procedimiento para la elaboración de las prácticas**

El procedimiento que se utiliza para elaborar las prácticas de laboratorio reales de Autómatas programables cuenta los siguientes pasos:

- **Tema:** Es una frase relativamente pequeña que da a conocer el tema a tratar, este se puede desarrollar en diferentes actividades.
- **Título:** En este apartado deberá expresarse el nombre de la práctica. El título deberá ser sugerente atractivo y relacionado con el tema o problema en estudio.
- **Objetivo:** Se plantea el objetivo a alcanzar con la realización de la práctica. Identifican la finalidad a la cual deben dirigirse los recursos y esfuerzos en aras de dar cumplimiento a los propósitos.
- **Materiales y equipos:** Se listan y describen los materiales y equipos necesarios para el desarrollo de la práctica de laboratorio. Especifica todo lo requerido en cuanto al tipo de equipos, materiales, tecnologías, herramientas y software, tanto para la etapa de experimentación como para la reproducción futura del problema en estudio.

- Conocimientos previos: Se refiere a los conocimientos teóricos que el estudiante debe poseer, en aras de contribuir al cumplimiento de los objetivos trazados en la práctica.
- Técnica operatoria: En este paso se describe de forma ordenada, los pasos a seguir durante el laboratorio. Este procedimiento debe conducir a que el estudiante adquiera la habilidad de seguir una secuencia de pasos a la hora de realizar la práctica.
- Conclusiones: Este paso se desarrollará de manera participativa donde el profesor guiará, mediante preguntas, a los estudiantes para que concluyan los principales aspectos tratados durante la práctica de laboratorio.
- Estudio independiente: Se debe orientar al estudiante, la realización de un ejercicio de laboratorio que permita sistematizar el conocimiento obtenido durante la práctica.
- Bibliografías: En este se indica la bibliografía básica y complementaria con la que fueron elaborados los contenidos de la práctica. Se validan el desarrollo de las prácticas de laboratorios.

### **3.2 Prácticas de laboratorios reales de Autómatas Programables.**

A continuación se describen las prácticas de laboratorio de Autómatas Programables.

La asignatura cuenta con 32 horas, distribuidas de la siguiente forma: 16 horas dedicadas a laboratorios reales, 10 horas para conferencias y 6 horas distribuidas en igual cantidad para un seminario, un trabajo de control parcial y un taller. Tiene dos temas, el tema uno abarca desde la práctica de laboratorio 1 de la asignatura hasta la práctica tres; el tema dos corresponde a la práctica cuatro y cinco.

#### **3.2.1 Práctica de laboratorio No.1**

**Tema 1:** Formación básica en la programación de Autómatas Modicon.

**Título:** Introducción al entorno del autómata Modicon.

**Objetivo:**

- Identificar la arquitectura externa e interna de los M241.
- Crear habilidades en el empleo del software SoMachine.

**Materiales y equipos:**

- Computadora con sistema operativo Windows, software Adobe Reader y SoMachine.
- Entrenador de autómatas Schneider Electric.

**Conocimientos previos:**

- Características esenciales de los autómatas programables y su software de programación. (Medios Técnicos de Automatización)

**Desarrollo del laboratorio:**

Para profundizar el entorno de trabajo se les indica a los estudiantes que identifiquen en el entrenador de procesos los componentes de hardware. Se comenta sobre las características de cada componente que conforma el entrenador y se muestran sus funcionalidades. Se repasa todo el hardware con el fin de que el estudiante confirme los conocimientos adquiridos en la conferencia. Además, se crean habilidades en la conexión y desconexión del entrenador.

En el caso del software se comienza a crear habilidades en su uso mediante la familiarización del estudiante con sus principales componentes.

El laboratorio número 1 se encuentra en su totalidad reflejado en el **anexo IV**.

**3.2.2 Práctica de laboratorio No.2**

**Tema 1:** Formación básica en la programación de Autómatas Modicon.

**Título:** Introducción al SoMachine.

**Objetivo:**

- Implementar las diferentes alternativas de lenguajes de programación en el Software SoMachine para el Modicon.
- Ejecutar los pasos para la programación de aplicaciones sencillas.

**Materiales y equipos:**

- Computadora con sistema operativo Windows y Software SoMachine.
- Entrenador de autómatas Schneider Electric.

**Conocimientos previos:**

- Características esenciales del M241 y el software de programación. (Conferencias 1 y 2 de la asignatura)
- Operaciones lógicas. (Electrónica Digital)
- Lenguajes de programación de Autómatas. (Medios Técnicos de Automatización)

**Desarrollo del laboratorio:**

Mediante la realización de dos ejercicios sencillos se pretende lograr que el estudiante programe en los dos principales lenguajes de programación que brinda el software. El esquema de contactos y el diagrama de bloques funcionales. Se aprovecha el ejercicio para introducir como configurar las entradas y salidas del autómata.

El laboratorio número 2 se encuentra en su totalidad reflejado en el anexo V.

Los ejercicios de estudio independiente se instruyen para reafirmar los contenidos estudiados en la conferencia y profundizar en las habilidades desarrolladas en el laboratorio.

El laboratorio número 2 se encuentra en su totalidad reflejado en el **anexo V**.

**3.2.3 Práctica de laboratorio No.3**

**Tema 1:** Formación básica en la programación de Autómatas Modicon.

**Título:** Instrucciones de aplicación.

**Objetivo:**

- Utilizar las instrucciones matemáticas y de comparación en el lenguaje de contactos (LD).
- Utilizar instrucciones de temporización y conteo en el lenguaje de diagrama de bloques funcionales.

**Materiales y equipos:**

- Computadora con sistema operativo Windows y Software SoMachine.
- Entrenador de autómatas Schneider Electric.

**Conocimientos previos:**

- Conferencias 3 de la asignatura.
- Operaciones lógicas. (Electrónica Digital)
- Lenguajes de programación de Autómatas. (Medios Técnicos de Automatización)

**Desarrollo del laboratorio:**

Se toman dos variables enteras y se realizan de operaciones matemáticas de suma, resta y multiplicación. Además, se comparan variables y números.

Se realizan ejercicios de conteo básico y de temporización. Este laboratorio tiene la particularidad que las instrucciones utilizadas en él son muy semejantes a las usadas en la asignatura MTA en la parte de autómatas por lo que el estudiante avanza muy rápido en la elaboración de los ejercicios.

Los ejercicios de estudio independiente de este laboratorio se asemejan más a problemas reales de la industria o los servicios.

El laboratorio número 3 se encuentra en su totalidad reflejado en el **anexo VI**.

### **3.2.4 Práctica de laboratorio No.4**

**Tema 2:** Formación avanzada en la programación de Autómatas Modicon.

**Título:** Adquisición de Variable analógica.

**Objetivo:**

- Instalar sensores pasivos a los autómatas Modicon M241.
- Configurar y programar los módulos analógicos TM3XX.

**Materiales y equipos:**

- Computadora con sistema operativo Windows y Software SoMachine.
- Sensores y transductores de temperatura, presión y nivel.
- Entrenador de autómatas Schneider Electric.

**Conocimientos previos:**

- Características técnicas y principio de funcionamiento de sensores-transductores: Pt100 y Cerabar. (Mediciones)
- Operaciones lógicas. (Electrónica Digital)
- Lenguajes de programación de Autómatas. (Medios Técnicos de Automatización)

**Desarrollo del laboratorio:**

Se comienza por la instalación de los sensores pasivos al entrenador de procesos. Luego se configuran en el software las entradas del módulo analógico. Se procede a trabajar con los valores obtenidos, se comparan y se activan salidas digitales; por último se simula y carga el programa al PLC.

El ejercicio de estudio independiente de este laboratorio reafirma los conocimientos desarrollados en la práctica real, realizando procedimientos muy parecidos en una máquina de llenado de botellas.

El laboratorio número 4 se encuentra en su totalidad reflejado en el **anexo VII**.

### **3.2.5 Práctica de laboratorio No.5**

**Tema:** Formación básica en la programación de Autómatas Modicon.

**Título:** Interfaz Gráfica.

**Objetivo:**

- Configurar paneles bases en la Magelis HMISTU tipo HMIS5T.
- Visualizar en el Vijeo Designer las variables del SoMachine.

**Materiales y equipos:**

- Computadora con sistema operativo Windows, Software SoMachine y Vijeo Designer.
- Entrenador de autómatas Schneider Electric.

**Conocimientos previos:**

- Características esenciales del HMIS5T y del software de programación Vijeo Designer. (Conferencia 6 de Autómatas Programables)
- Comunicación entre HMI y el M241(Conferencia 6 de Autómatas Programables)

**Desarrollo del laboratorio:**

Se inicia por abrir el software SoMachine en la PC, y mediante él abrir el software Vijeo Designer. Luego crear los paneles principales que permitan al usuario la navegación. Crear Visualizadores Numéricos en los paneles para mostrar el valor de las variables. Después se configuraran teclados emergentes que posibilitan el intercambio de información entre el operador y el HMI. También se utilizan luces pilotos que muestran el estado de la comparación entre un valor establecido por el usuario y el valor actual de la variable en su respectivo proceso.

El ejercicio de estudio independiente de este laboratorio consiste desarrollar conocimientos adquiridos en el laboratorio real pero para las variables presión y nivel. Además se orienta una actividad de investigación extra curricular.

El laboratorio número 5 se encuentra en su totalidad reflejado en el **anexo VIII**.

### 3.4 Análisis económico

Los laboratorios reales requieren de equipos electrónicos altamente costosos, en Cuba es muy difícil contar con un equipamiento así. La compra de un entrenador prediseñado para prácticas reales como los mostrados en las figuras 3.15 (SIEMENS, 2017) y figura 3.16 (Schneider Electric, 2017) puede costar entre 3000 y 3 500 euros.



Figura 3.15 Entrenador S7-1516 que permite el estudio de la programación, puesta en marcha y mantenimiento de controladores de gama media / alta.



Figura 3.16 Entrenador con PLC de la firma Schneider Electric y HMI para la supervisión.

Mientras que comprando los componentes por separado a la firma Schneider Electric y los sensores-transductores en la Endress+Hauser; y luego instalando la instrumentación e implementando el software en el PLC y la HMI, puede llegar a costar aproximadamente 1800 euros, ver tabla 3.2 (Schneider Electric, 2014).

Luego para elaborar una guía de prácticas reales habría que pagarle a un personal calificado, lo cual puede costar unos 500 euros, ver tabla 3.2.

Tabla 3.2 Relación de precio por componentes del entrenador de la firma Schneider Electric

Componente	Cantidad	Precio(euros)
HMI Magelis tipo S5T	1	207

PLC M241	1	342
Módulo TM3AM6/G	1	180
Módulo Switch Ethernet TM4ES4	1	100
Sensor Cerabar T PMC 131	2	400
Pt100	1	25
Transmisor SEM1503P	1	136
Fuente de 24V	1	145
Breaker	1	149
Cables y accesorios	-	100
Elaborar las prácticas de laboratorio reales	5	500
Total	-	2284

Esto posibilita que en vez de comprar un entrenador prediseñado por la firmas Simens o Schneider Electric, se pueden comprar los medios técnicos para implementar dos entrenadores. Por lo tanto se puede adquirir más equipamiento con una misma inversión, pero para su uso se tiene la limitante de la cantidad y prestaciones de los ordenadores presentes en el laboratorio.

### 3.5 Consideraciones finales del Capítulo.

Las prácticas de laboratorio de la asignatura Autómatas programables tiene una gran importancia en la misma ya que constituye más del 40 por ciento del total de horas. Además, los laboratorios permiten crear habilidades en la manipulación, programación y ejecución de los autómatas.

En las prácticas de laboratorio se definen paso a paso como solucionar la técnica operatoria y se orienta un estudio independiente para consolidar las habilidades desarrolladas en él.



## CONCLUSIONES Y RECOMENDACIONES

### Conclusiones

- 1 Los entrenadores de autómatas programables tienen un amplio uso a nivel mundial en la formación de personal calificado en la programación de autómatas. Su mayor utilidad radica en las facilidades que brinda en la ejecución de prácticas de laboratorios reales.
- 2 El entrenador de autómatas programables de la firma Schneider Electric tiene todos los componentes de hardware necesarios para la realización con él de prácticas de laboratorio reales en la asignatura autómatas programables.
- 3 Las guías de las prácticas de laboratorios reales de la asignatura autómatas programables permite el desarrollo de habilidades en la programación de autómatas y su interfaz gráfica.

### Recomendaciones

- 1 Implementar otras maquetas o entrenadores similares que permitan aumentar el número de estudiantes por laboratorios.
- 2 Continuar el desarrollo de las prácticas reales de comunicación entre Autómatas y mando a motores en la otra maqueta que se encuentra en el laboratorio.
- 3 Mejorar las prestaciones de software y hardware en las computadoras del laboratorio 231 al fin que los laboratorios reales se ejecuten con la velocidad adecuada.

## REFERENCIAS BIBLIOGRÁFICAS

- Abreu González, Ana Ma., (2013). Las formas de docencia y el trabajo independiente en la educación superior.
- Álvarez Álvarez, C., (2012). La relación de la teoría y la práctica en los procesos de enseñanza- aprendizaje,” vol. 30 no 2, pp. 383–402.
- Anon., 2007. Reglamento para el Trabajo Docente y Metodológico en la educación superior.. En: s.l.:s.n.
- Calvo Gordillo, I., (2008). Laboratorios remotos y virtuales en enseñanzas técnicas y científica.
- Campaña, P., Marcelo, C., Campaña, R., Luis, J., (2016). Análisis del sistema actual de riego y productividad para la implementación de un sistema automático de riego controlando los parámetros de temperatura y humedad mediante sistema HMI en la finca “San Nicolas” sector de Illuchi ubicado en el barrio Juan Pablo Segundo en el cantón Latacunga, provincia de Cotopaxi en el periodo 2015-2016.
- Cañedo Iglesias, C., (2009). Fundamentos teóricos para la implementación de la didáctica en el proceso enseñanza-aprendizaje.
- Clemente L, M., (2007). La complejidad de las relaciones teoría-práctica en educación.
- Creus, A. (2010). Instrumentación Industrial. Ciudad de México Alfaomega Group Editor.
- Chacón García, A., (2013). Plataforma didáctica para el estudio de procesos térmicos en laboratorio de instrumentación industrial.
- Djelil Belarb, A., (2006). La importancia de los laboratorios.
- Endress+Hauser, (2011). Catálogo general para máquinas de automatización e industriales.

- Fuentes Revilla, E., (2015). Diseño aplicaciones HMI con pantallas Magelis. Disseny
- González Nieves, Y., (2012). Prácticas de laboratorio con el software Altera MAX+Plus II y el Kit de desarrollo KHF-E5.
- Izaguirre, E., (2012). Sistemas de automatización U. C. M. A. d. L. Villas. Santa Clara, Editorial Feijóo. **1**: 136.aplicacions HMI amb pantalles Magelis.
- León, U., (2010). Introducción a la maqueta industrial de 4 variables. Laboratorio Remoto de Automática. León, España, Grupo SUPPRES: 12.
- López Morocho, H.V., Navarrete Machado, D.F., (2015). Diseño e implementación de una estación de posicionamiento en tres ejes con probetas de distintos materiales y tamaños con PLC para el laboratorio de control y automatización de procesos industriales de la Facultad de Mecánica de la ESPOCH (B.S. thesis). Escuela Superior Politécnica de Chimborazo.
- Morera Martínez, Y., (2013). Modernización del lazo de control de nivel en la maqueta industrial de tres variables.
- Oriol Boix, Aragonés., (2014). Construcción de la maqueta de un ascensor para la realización de prácticas con autómatas programables y diseño de las prácticas correspondientes. Escuela de Ingeniería en Electrónica Industrial, Barcelona, España, Universidad de Barcelona.
- Portilla, P., Andres, G., Yanez, G., Darly, I., 2015. Control de nivel de líquido implementado en la planta de entrenamiento gunt rt-450 con tecnología siemens (plc s7-1200 y hmi ktp 400 mono basic panel).
- Richmond, E., (2009). Diseño y construcción de una interfaz de control de nivel, temperatura y flujo de agua en un tanque para uso en prácticas de laboratorio Facultad de Ingeniería. Escuela de Ingeniería Química. San José, Costa Rica, Universidad de Costa Rica. Ingeniero: 169.
- Ríos Sabogal, G.A., (2016). Diseño de una estación para la medición de temperatura de los gases de un motor de combustión interna (B.S. thesis). Pereira: Universidad Tecnológica de Pereira.

- Romero Rugel, L.A., (2014). Diseño e implementación de un entrenador de Instrumentación Industrial, con aplicaciones en los procesos de temperatura, presión y nivel. Facultad de Ingenierías. Carrera Ingeniería Electrónica. Guayaquil, Ecuador Universidad Politécnica Salesiana
- Rosario, J., (2013). Las TIC para el proceso enseñanza-aprendizaje en los laboratorios. Vol. 2, no. 2174–8985, 2013.
- Rosero, T., Wladimir, W., Palomino, Z., Guillermo, E., (2016). Estudio y análisis de diseño de una red de acceso con tecnología plc (Power Line Communications) para las unidades del Ejército Ecuatoriano.
- Schneider Electric., (2014). Automatización y control industrial: lista precios Enero 2014 - ESMKT02023A14\_Industry\_PDF\_completa.pdf [WWW Document]. URL [https://www.schneiderelectric.es/documents/local/soporte/tarifas/2014/enero/pdf/ESMKT02023A14\\_Industry\\_PDF\\_completa.pdf](https://www.schneiderelectric.es/documents/local/soporte/tarifas/2014/enero/pdf/ESMKT02023A14_Industry_PDF_completa.pdf) (accessed 6.9.17).
- Schneider Electric., (2017). Material didáctico de centro de Formación.
- SIEMENS, 2017. 314 SCE - CF - S71500 [WWW Document]. URL <https://es.scribd.com/document/341963589/314-SCE-CF-S71500> (accessed 6.9.17).
- Stefania Cuellar, A., (2011). Enseñanza de Química y su articulación con el nivel medio.
- Terrón, J., (2016). Empleo de Maquetas con Interface Hardware con Equipos PLC para Docencia en Automatización Naval
- Tran Thi, T., (2012). Plataforma para Prácticas Simuladas con el uso de Autómata como Controlador.
- Vallejo, H.D., (2014). Los controladores lógicos programables. Fuente Httpwww Todopic Com ArutilesPLC Pdf Fecha Consulta 6, 6.
- Vallduan, P.I., (2014). Los entrenadores de Autómatas en la Automatización Industrial.
- Valverde Endaramarco, A., (2014). Implementación del PLC Simatic S7-1200 y pantalla táctil al módulo de transporte en 3 ejes para el laboratorio de control y manipulación automática de la escuela de Ingeniería de Mantenimiento de la ESPOCH. Facultad de Mecánica de la ESPOCH (B.S. thesis). Escuela Superior Politécnica de Chimboraz

## ANEXOS

### Anexo I Componentes del M241 y descripción de ellos.

En la figura 1 se muestran los componentes de M241 y en la tabla 1 la descripción de ellos

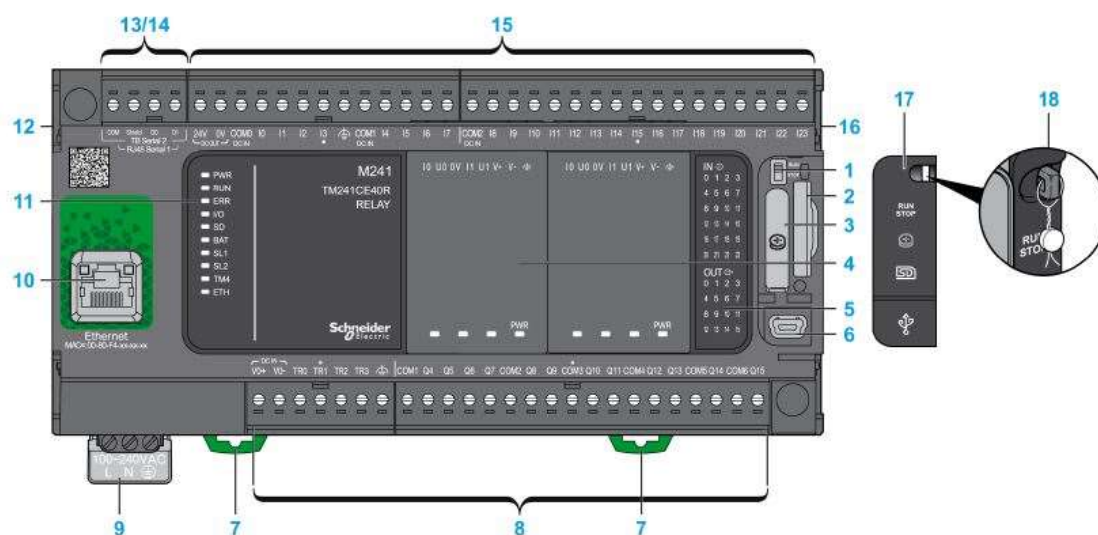


Figura 1 Componentes del Autómata M241 tipo TM241CE40R.

Tabla 1 Descripción de los componentes del Autómata M241 tipo TM241CE40R.

Numero	Descripción
1	Interruptor Run/Stop
2	Slot para tarjeta SD
3	Soporte de la batería
4	Slot para cartucho

5	Indicadores LED para indicar los estados de E/S
6	Puerto de programación USB mini-B / para la conexión de terminales a un PC de programación (SoMachine)
7	Carril DIN (segmento DIN) de cierre de clip para 35 mm (1,38 in)
8	Salidas de relé incrustadas
	Salidas de transistor rápidas incrustadas
	Bloque de terminales extraíble de la salida
9	Fuente de alimentación de 50/60 Hz de 100 a 240 V CA
10	Puerto Ethernet / Tipo RJ45 (RS-232 o RS-485)
11	Indicadores LED de estado
12	Conector de bus de TM4
13	Puerto de la línea serie 1 / Tipo RJ45 (RS-232 o RS-485)
14	Puerto de la línea serie 2 / Tipo del bloque de terminales de tornillo (RS-485)
15	Entradas digitales incrustadas
	Bloque de terminales extraíble de la entrada
16	Conector de bus TM3/TM2
17	Cubierta de protección (slot para tarjeta SD, interruptor Run/Stop y puerto de programación USB mini-B)
18	Gancho de sujeción (gancho no incluido)

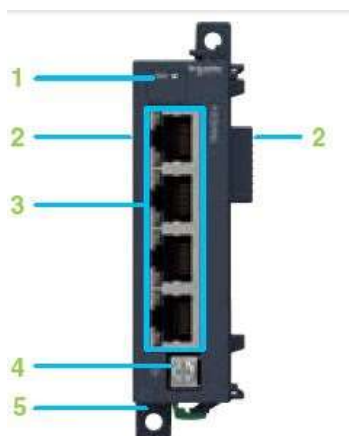
**Anexo II Componentes del módulo Ethernet TM4ES4 y descripción de ellos.**

Figura 1 Componentes del Módulo Ethernet TM4ES4

Tabla 1 Descripción de los componentes del Módulo Ethernet TM4ES4

Componente	Descripción
1	Indicador LED de alimentación
2	Conector de bus (uno en cada lateral)
3	Conectores RJ45 para red Ethernet con LED de actividad
4	Borna de tornillo para la conexión a tierra (FE)
5	Pestaña para bloqueo en carril simétrico

### Anexo III Instalación del transductor Cerabar T PMC 131 en el “Calibrador de peso muerto”.





**Anexo IV Práctica No.1 Introducción al entorno del autómeta Modicon.****Tema # 1 Formación básica en la programación de Autómatas Modicon.****Sumario:**

Alimentación, conexión y módulos del autómeta M241.

Ambiente de trabajo del software SoMachine.

**Objetivos:**

Identificar la arquitectura externa e interna de los M241.

Crear habilidades en el empleo del software SoMachine y el entrenador de la firma Schneider Electric.

**Materiales y equipos:**

Computadora con sistema operativo Windows, software Adobe Reader y SoMachine.

Entrenador de autómatas Schneider Electric.

**Conocimientos previos:**

Características esenciales de los autómatas programables y su software de programación.  
(Medios Técnicos de Automatización)

**Técnica Operatoria:****1-Para conocer el entorno de trabajo.**

a) Observe el entrenador de autómatas de Schneider Electric.

Una imagen de ella se muestra en la figura 1, la cual se usará como principal material para la realización de los laboratorios reales en las asignaturas de Autómatas Programables.



Figura 1. Entrenador de la firma Schneider Electric

b) Identifique cada componente del entrenador de autómatas.

b.1) El Controlador Lógico M241. Refiera sus principales características.

b.2) Módulo Ethernet TM4ES4. Refiera sus principales características. Explique las reglas de asociación del mismo.

b.3) Módulo de E/S analógicas TM3AM6/G. Identifique la forma de cableado del mismo.

b.4) Módulo TM3XTYS4 de arrancador de motores. Identifique sus partes y componentes.

b.5) HMI Magelis STU tipo HMIS5T. Identifique sus partes y componentes.

**Conclusiones parciales.** En la figura 2 se muestra una imagen del entrenador de la firma Schneider Electric resaltando todos componentes.

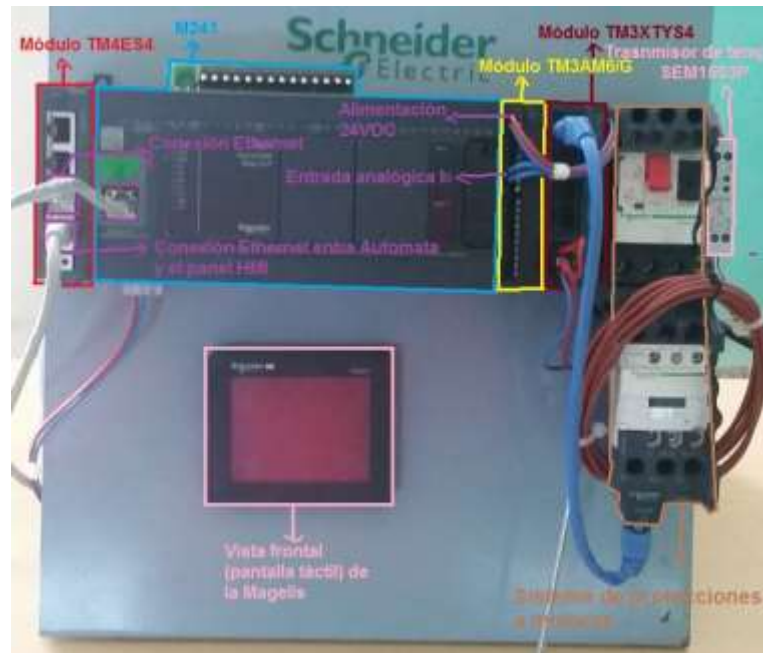


Figura 2. Componentes del entrenador de la firma Schneider Electric.

## 2- El software SoMachine.

Se ofrece una guía básica para iniciar un proyecto en el software SoMachine.

### Ejecutar los pasos siguientes:

#### Paso 1- Abrir SoMachine.

Como se muestra en la figura 3 para abrir SoMachine seleccione el icono SoMachine en el menú de inicio de Windows ó:

Inicio » Programas » Schneider Electric » SoMachine Software » V4.1 » SoMachine V4.1

También puede hacer doble clic en el icono de SoMachine en el escritorio.

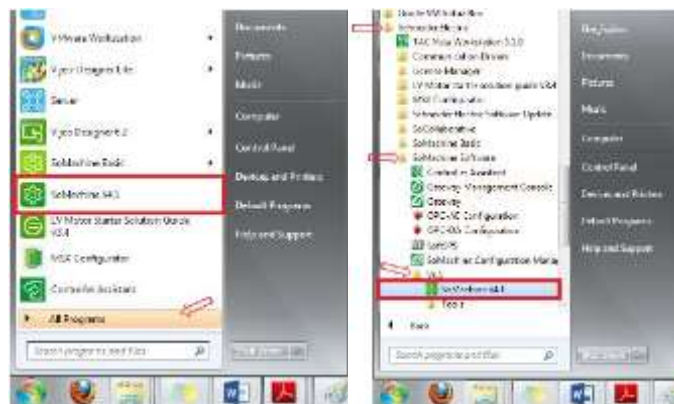


Figura 3. Vías para abrir el software SoMachine

**Paso 2-** Crear un nuevo proyecto.

Para crear un nuevo proyecto presione el botón ‘Nuevo proyecto’ en la pantalla de inicio, como se muestra en la figura 4.

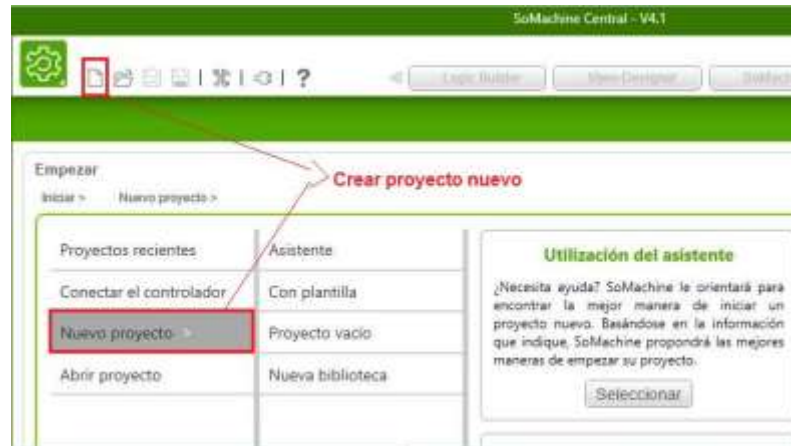


Figura 4. Crear proyecto

**Paso 3-** Crear proyecto vacío.

Cuando se crea un ‘Nuevo Proyecto vacío’ se crea un proyecto simple, sin pre configuración de dispositivos, ni lógica. Sobre la base de su nivel de experiencia, esta opción le ofrece total flexibilidad.

Como se muestra en la figura 5, haga clic en ‘Proyecto vacío’ y aparecen dos pestañas en el área de trabajo, en la pestaña ‘General’ escriba el nombre de nuestro proyecto ‘Mi\_1er\_proyecto’. Posteriormente de clic en el botón ‘Crear proyecto’ para continuar.



Figura 5. Proyecto vacío.

**Paso 4-** Bloque de configuración.

Como se muestra en la figura 6 para añadir dispositivos al proyecto desde la ventana general de SoMachine Central, seleccione la pestaña de ‘Flujo de trabajo’. Dentro de ella debe dar clic en el bloque de ‘Configuración’ para añadir y/o eliminar cualquier dispositivo.

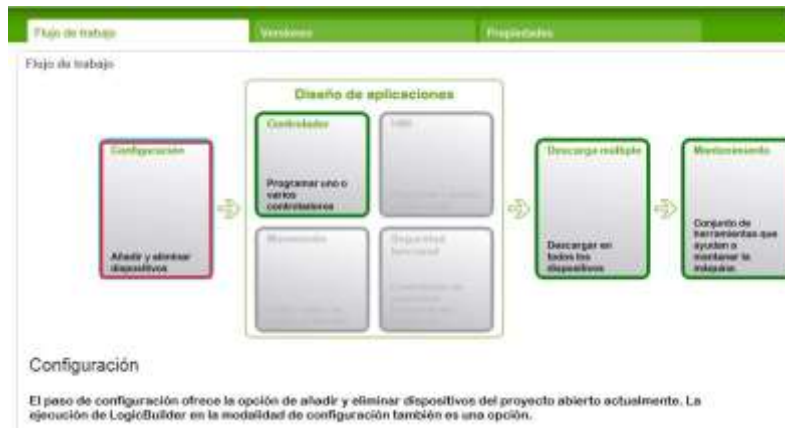


Figura 6. Bloque de configuración

**Paso 5-** Añadir dispositivos.

Como se muestra en la figura 7 en la ventana de configuración aparecen dos áreas, a la izquierda tenemos el catálogo de dispositivos que se pueden añadir al proyecto. En el área de la derecha aparecen todos los dispositivos que ya se han añadido al proyecto. Presione las teclas verdes para adicionar o eliminar dispositivos a la aplicación.



Figura 7. Añadir dispositivos al nuevo proyecto

**Paso 6-** Acceder a la ventana de programación.

Como se muestra en la figura 8 para abrir la ventana de programación ‘Logic Builder’ desde la ventana general de SoMachine Central, seleccione el bloque ‘Controlador’ dentro de ‘Diseño de aplicaciones’. O en la barra de herramientas de acceso, que se muestra en la parte superior seleccionar ‘Logic Builder’.



Figura 8. Acceder a la ventana de programación

En caso de aparecer una ventana flotante titulada “DTMRepositorio” cuando abrimos el Logic Builder, que nos solicita agregar nuevos DTM para este proyecto simplemente ciérrala.

**Paso 7-** Interfaz de programación.

En la figura 9 se muestra las diferentes áreas de la interface de programación.



Figura 9. Áreas de la interface de programación.

- Barra de herramientas de programación: barra de acceso rápido a los elementos principales de programación, como se muestra en la figura 10 y 11 los iconos de la barra de programación, variarán en función del lenguaje de programación elegido para ese POU (Unidad Principal Organizativa).
- Barra de herramientas del diagrama de contactos (LD).

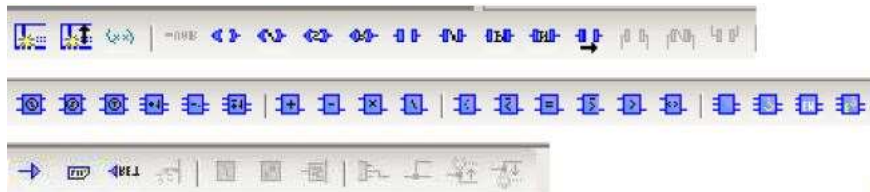


Figura 10. Barra de herramientas (LD)

- Barra de herramientas de diagrama de funciones (FBD).



Figura 11. Barra de herramientas (FBD)

- Navegador de proyecto: En esta área se muestra los diferentes elementos que componen el programa, estructurado para una mejor comprensión de que se está haciendo en el proyecto.

El navegador consta de tres pestañas.

- Pestaña de Dispositivos: Como se muestra en la figura 12, se utiliza para configurar el hardware del controlador elegido.



Figura 12. Pestaña de Dispositivos

- Pestaña de Aplicación: Como se muestra en la figura 13 esta pestaña se utiliza para realizar la programación de la aplicación y todos los elementos relacionados (lista de variables globales, tareas, estructura de datos, visualizaciones, POU's).



Figura 13. Pestaña de aplicaciones

- Pestaña de Herramientas: En la figura 14 se muestra esta pestaña en la cual se encuentra el administrador de bibliotecas y las visualizaciones que se creen el proyecto.





Figura 14. Pestaña de herramientas

- Área de trabajo: En la figura 15 se muestra la zona de trabajo que es donde se realizarán todas las parametrizaciones, programación y llamadas de cada uno de los elementos seleccionados en las diferentes pestañas de navegación. Las llamadas realizadas a los diferentes objetos en el navegador quedarán como pestañas de acceso rápido en la parte superior del área de trabajo.

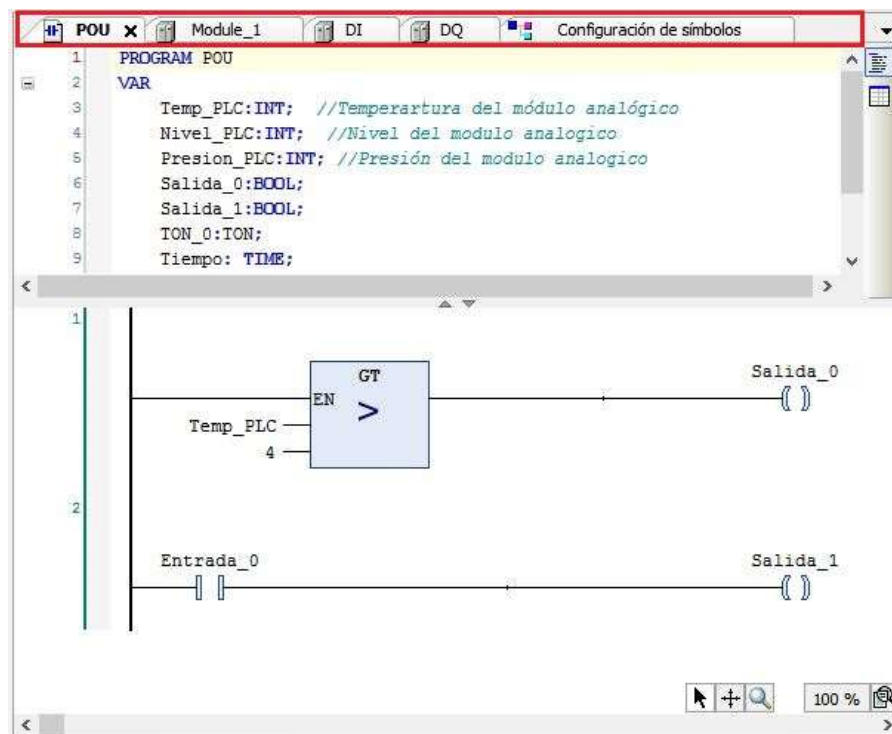


Figura 15. Área de trabajo de software SoMachine

- Declaración de variable: Como se muestra en la figura 16 cuando se abre un POU, en el área de trabajo aparece un área privada de declaración de variables de este POU (estas variables serán privadas de este POU).

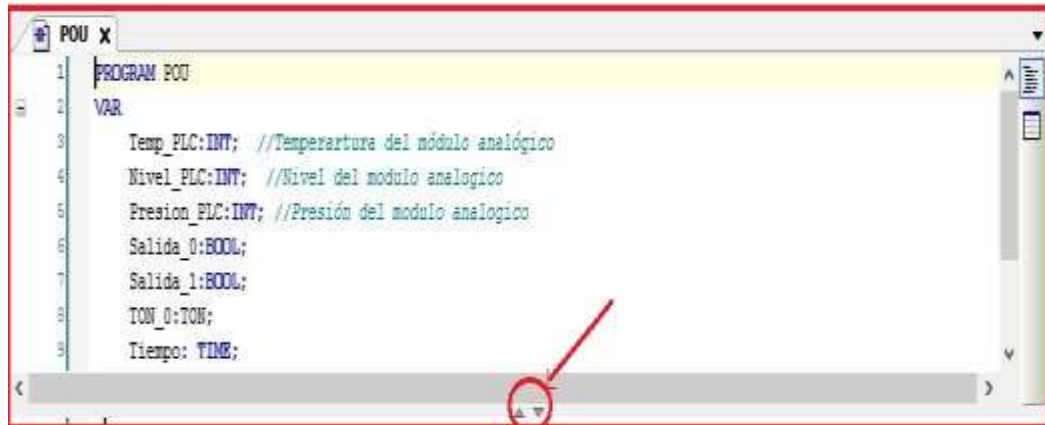


Figura 16. Área de declaración de variables.

- Área de programación: es el área donde se realizarán la programación del POU seleccionado, el formato de visualización dependerá del lenguaje de programación seleccionado:
  - Diagrama de contactos (LD).
  - Diagrama de bloques funcionales (FBD).
  - Diagrama de funciones continuas (CFC).
  - Diagrama funcional secuencial (SFC).
  - Lista de instrucciones (IL).
  - Texto estructurado (ST).
- Herramientas: Esta área se muestra en la figura 17 y contiene las funciones y herramientas que se pueden utilizar para la programación del POU, o del objeto que esté en el área de trabajo en ese momento.

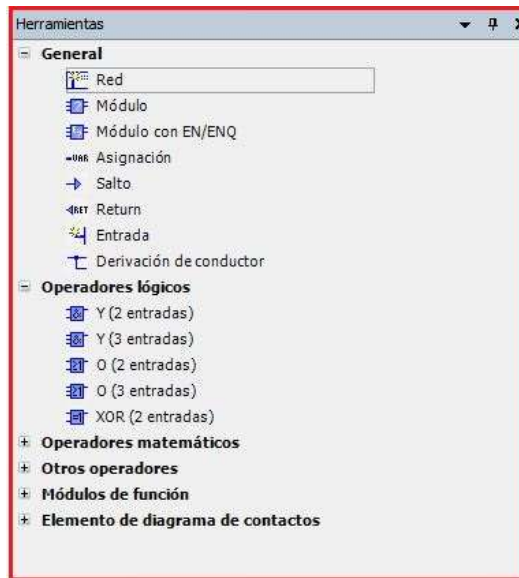


Figura 17. Área que muestra las herramientas a utilizar en el POU.

- Pestaña de mensajes de compilación: Esta pestaña se muestra en la figura 18, normalmente se encuentra en la parte inferior de la ventana de Logic Builder, nos muestra los mensajes de compilación (errores, advertencias y mensajes).

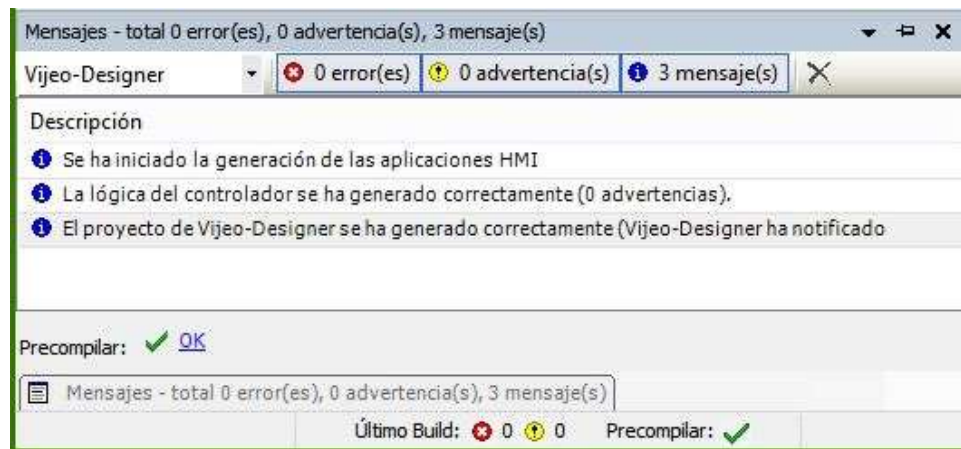


Figura 18. Área de compilación

### Paso 8- Agregar los módulos de ampliación.

Hay tener en cuenta las limitaciones de hardware de cada uno de los controladores a la hora de añadir módulos de expansión. Por ejemplo, el M241 tiene una limitación de 7 módulos de ampliación siempre que no se ponga módulos de expansión, también hay que tener en cuenta que: en este controlador, los módulos TM2 no se pueden intercalar con módulos TM3, sino hay que ponerlos al final del bus del PLC.

En la figura 19 se muestra que Debajo del ‘IO\_Bus (IO bus – TM3)’ en la pestaña ‘Dispositivos’ en el ‘Navegador’, aparecerán representados todos los módulos de expansión añadidos.

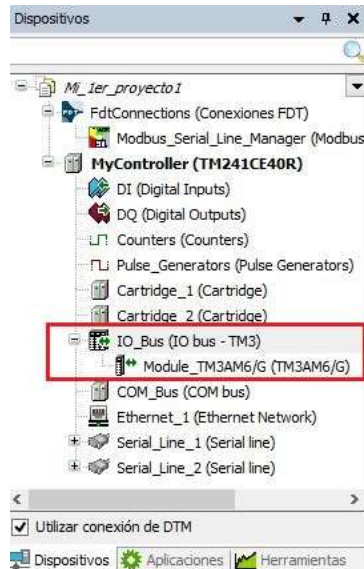


Figura 19. Representación de los módulos añadido

Como se muestra en la figura 20 para agregar los módulos al Controlador Lógico en el ‘Navegador’ seleccionamos la pestaña de ‘Dispositivos’ expandimos el controlador y al seleccionar ‘IO\_BUS (IO bus – TM3)’ aparece en lado de la derecha un símbolo de ‘+’ para añadir módulos de expansión, cuando salga a la derecha la ventana ‘Agregar el dispositivo’ seleccionamos el fabricante y el módulo y luego para agregarlo al programa hacemos doble clic encima del módulo o damos clic en la pestaña ‘Agregar dispositivo’.

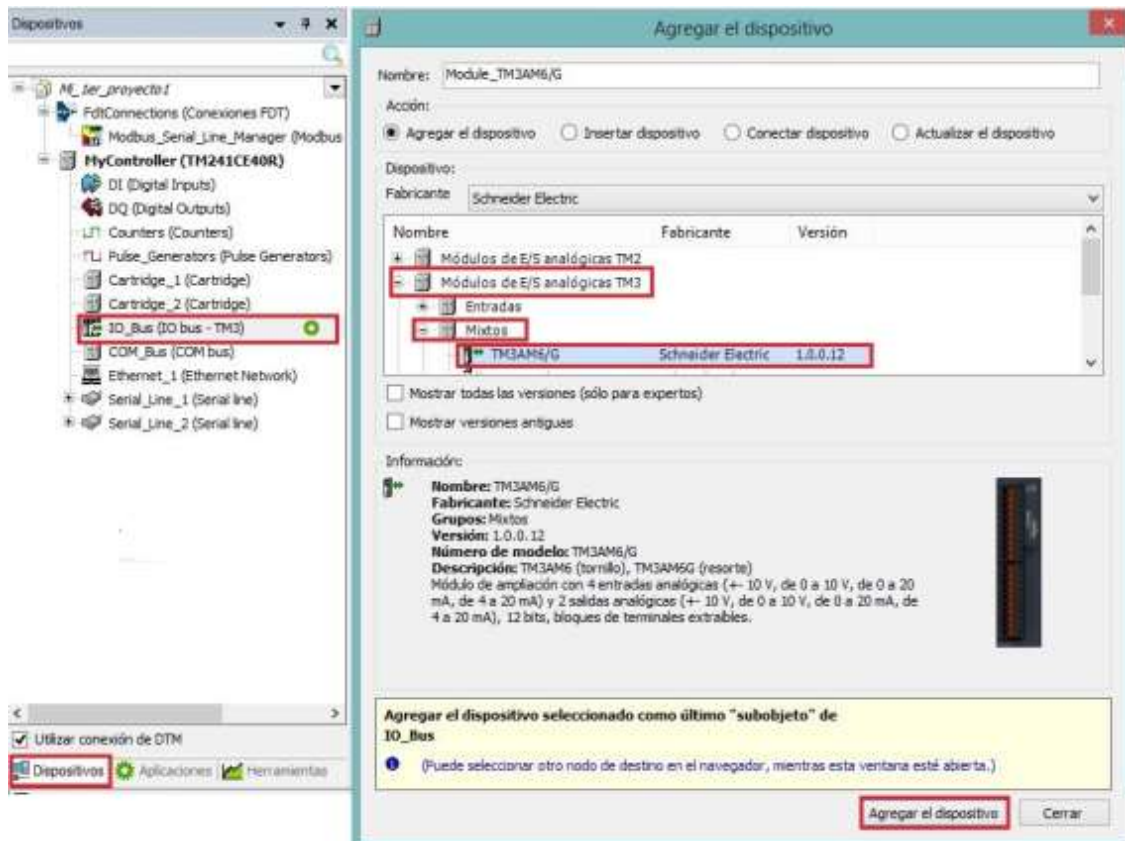


Figura 20. Agregar módulo TM3AM6/G

### Conclusiones:

¿Qué características tiene los módulos de expansión del entrenador de autómatas de la firma Schneider Electric?

¿Cuáles son los pasos esenciales para crear un proyecto con el software SoMachine y como se configura el hardware del entrenador?

### Bibliografía:

1-Manual de Ejercicios. SoMachine v4.1. Curso de Certificación. Disponible en: <https://moodle.uclv.edu.cu>

2-SoMachine. Manual de formación. Disponible en: <https://moodle.uclv.edu.cu>

3-Modicon TM3. Módulos de E/S digitales. Guía de hardware, abril del 2014.

Disponible en: <https://moodle.uclv.edu.cu>

4-Modicon TM4. Módulos de ampliación. Guía de programación, abril del 2014.

Disponible en: <https://moodle.uclv.edu.cu>

5-InfoPLC 2017. Manuales y ejemplos programación autómatas, PLC, HMI. Disponible en:

<http://www.infopl.net/descargas/240-schneider-electric/automatas/m241-m251>

**Anexo V Práctica No.2 Introducción al SoMachine.****Tema # 1 Formación básica en la programación de Autómatas Modicon.****Sumario:**

Programación del autómata M241 en el software SoMachine.

**Objetivos:**

Implementar las diferentes alternativas de lenguajes de programación en el Software SoMachine para el Modicon.

Ejecutar los pasos para la programación de aplicaciones sencillas.

**Materiales y equipos:**

Computadora con sistema operativo Windows y Software SoMachine.

Entrenador de autómatas Schneider Electric.

**Conocimientos previos:**

Características esenciales del M241 y el software de programación. (Conferencias 1 y 2 de la asignatura)

Operaciones lógicas. (Electrónica Digital)

Lenguajes de programación de Autómatas. (Medios Técnicos de Automatización)

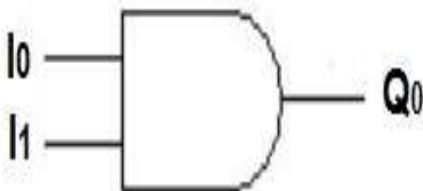
**Técnica Operatoria:****1- Realice las operaciones lógicas que se muestran en la figura 1 y 2.**

Figura 1. AND con salida Q0



Figura 2. OR con salida Q1

Para llevar a cabo esta tarea se debe seguir los siguientes pasos: (paso 1 al 10) son comunes para ambas operaciones, el paso 11 y 12 solo para la operación 2; el paso 13 es común para ambas.

**Paso 1-** Abrir el SoMachine y seguir los pasos del 2 al paso 6 del laboratorio#1.

Una vez abierta la interfaz de programación, realizamos los primeros pasos que son los más importantes, ellos son los siguientes:

añadir un POU (Unidad Organizativa del Programa).

agregar la llamada a una nueva tarea.

**Paso 2-** Agregar un POU. Un POU es una sección donde se escribirá el código del programa, para ello se va a la pestaña ‘Aplicaciones’ y dentro de ella como se muestra en la figura 3 hay un elemento titulado ‘Application(tipo del controlador)’ y a su lado un símbolo de ‘+’ el cual te permite añadir el POU.

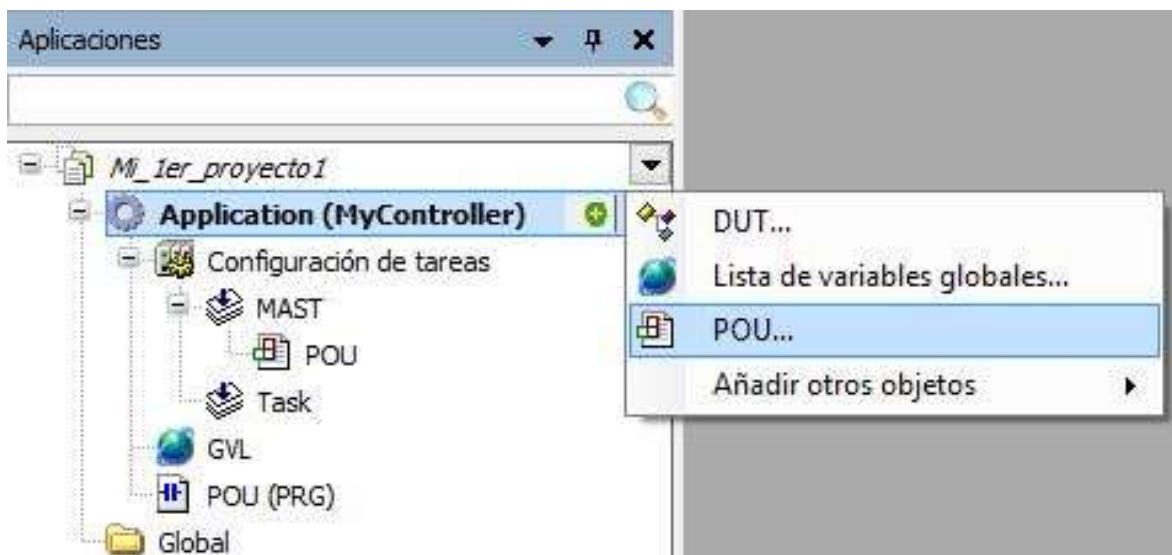


Figura 3. Agregar un POU

Luego como se muestra en la figura 4, hay 3 tipos de POU, seleccionamos el tipo Programa y el lenguaje de implementación diagrama de contactos (LD) y damos ‘Agregar’.



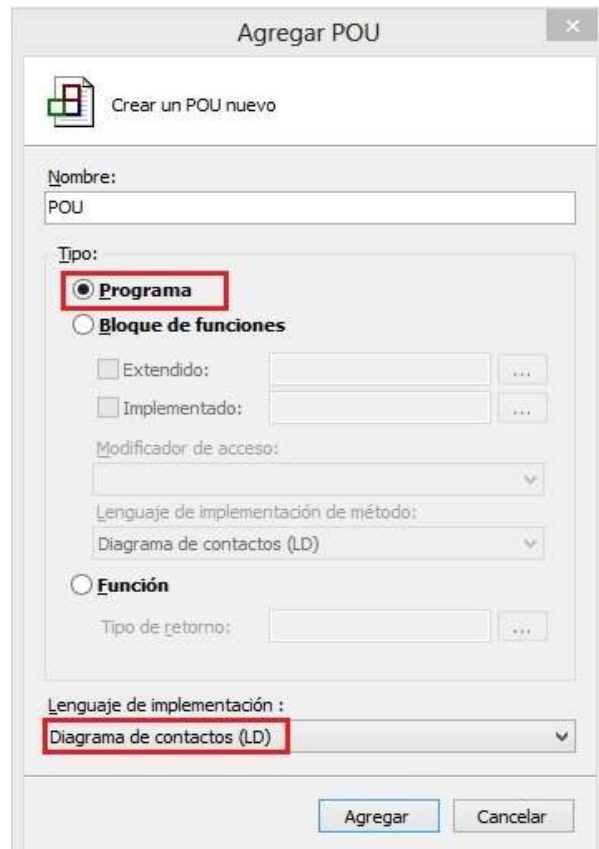


Figura 4. Selección del lenguaje de implementación

Luego de agregado el POU se muestra la interfaz que se puede observar en la figura 5.

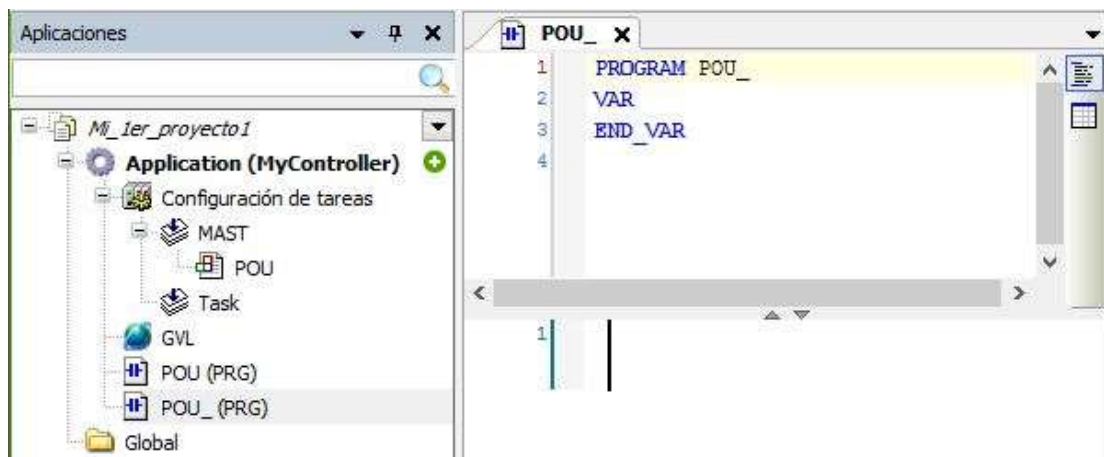


Figura 5. Interfaz de programación

**Paso 3-** Agregarle la llamada a una tarea a ese nuevo POU es otro paso de gran importancia, una tarea es un grupo de secciones de programa y subrutinas ejecutadas de una manera cíclica o periódica por la tarea principal MAST, esta tarea controla la ejecución de la Aplicación

(POU) en el controlador. Para agregar la llamada a la tarea se va a la pestaña ‘Aplicaciones’, dentro hay un elemento titulado ‘Configurador de tareas’ y a su lado hay un símbolo de ‘+’ que te permite añadir la tarea, como se muestra en la figura 6.

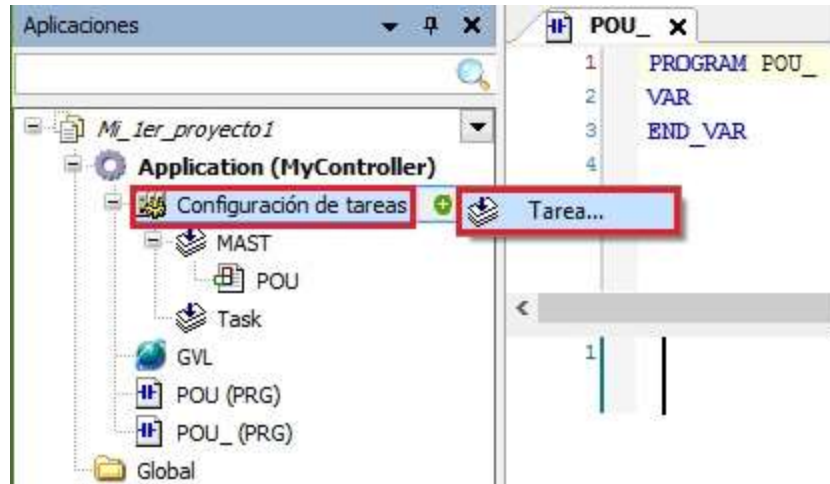


Figura 6. Añadir una tarea

Luego como se muestra en la figura 7 introducimos el nombre deseado para la tarea y damos clic en ‘Agregar’

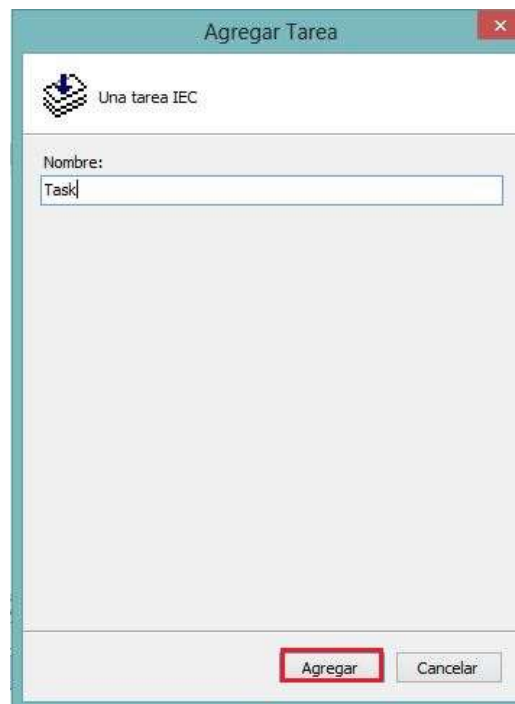


Figura 7. Poner nombre de la tarea

Como se muestra en la figura 8 seleccionamos el elemento ‘Agregar llamada’ y nos aparece la pestaña ‘Accesibilidad’ la cual puede observarse en la figura 9.

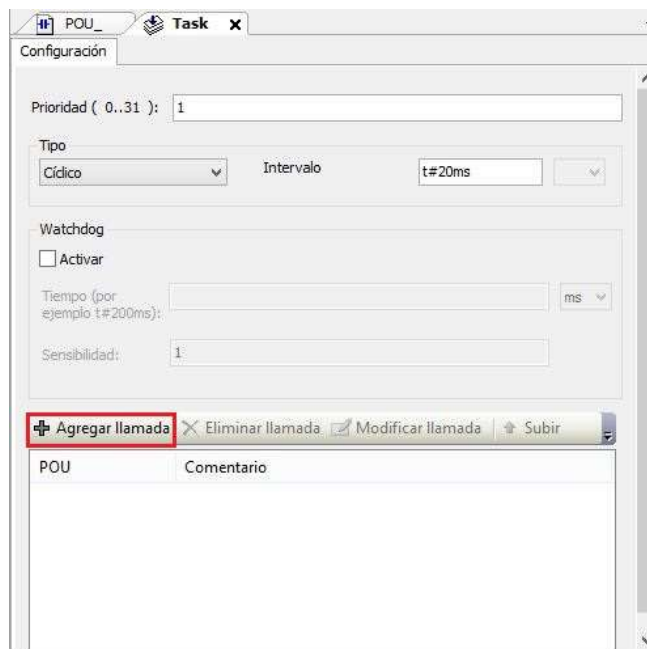


Figura 8. Agregar llamada a la tarea

En la figura 9 se muestra la pestaña ‘Accesibilidad’ en la cual seleccionamos el nombre de nuestra POU y presionamos aceptar. De esta forma la tarea se ejecuta cíclicamente controlando la ejecución de nuestra aplicación POU en nuestro PLC M241.

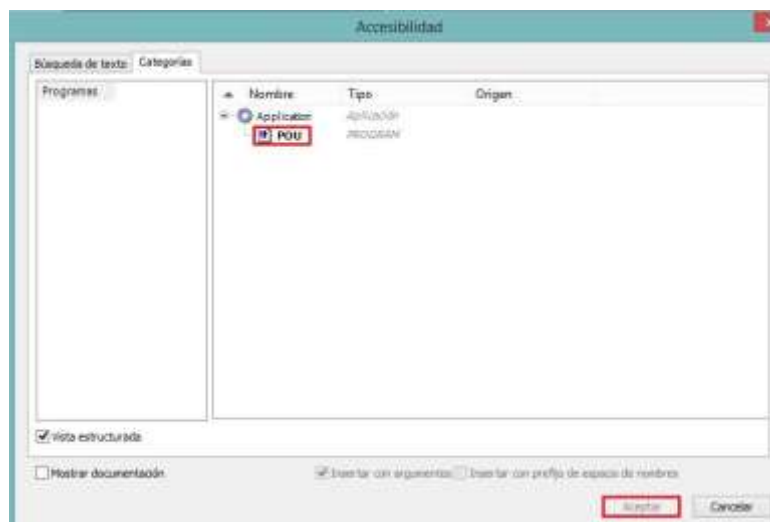


Figura 9. Selección del POU

**Paso 4-** Asignar una variable a cada canal de entrada digital que se va a utilizar. Como es necesario utilizar entradas en la técnica operatoria usaremos 4 entradas digitales (I0, I1, I2 e I3) de las 24 entradas digitales con las que cuenta el PLC M241 tipo TM241CE40R añadido al proyecto en el paso 5 del laboratorio#1. En la figura 10, se muestra la pestaña 'Dispositivos' dentro de la pestaña de 'Navegación', ahí seleccionamos el elemento 'DI (Digital Inputs)' lo cual coloca como se muestra en la figura 11, una pestaña de acceso rápido en el 'Área de trabajo'.

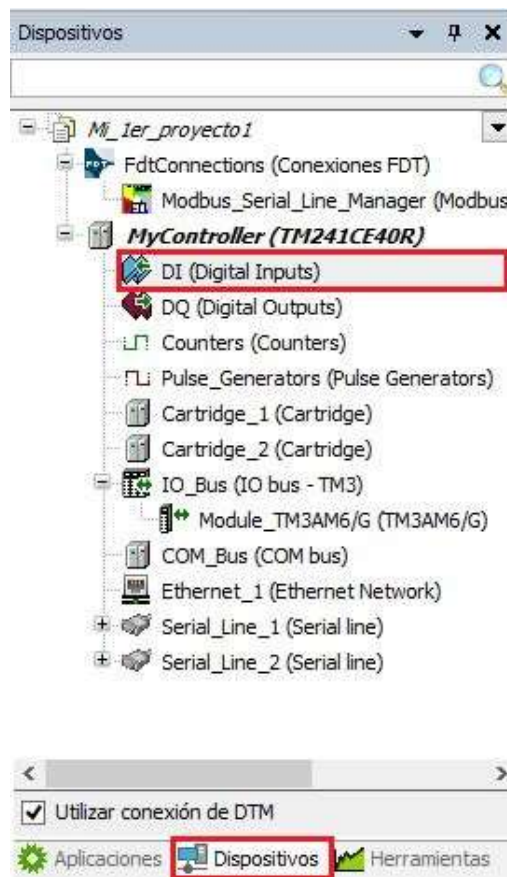


Figura 10. Selección de las Entradas Digitales del M241

En la figura 11 se muestra:

el formato en el que se debe asignar el nombre de la variable al canal de entrada digital seleccionado: 'Application.nombredelPOU.nombredela variable'.

y luego debe dar clic en el símbolo que esta circulado de azul, para que el mismo cambie al estado de asignación a un canal de entrada, como puede observarse en los símbolos de asignación que están dentro del rectángulo rojo de la figura.



Figura 11. Asignación de la variable con un canal de entrada digital.

**Paso 5-** Declarar las variables asignadas. Ya en este momento hay cuatro canales de entrada digitales los cuales dependen del estado de las variables que se les asignaron, ahora vaya al 'Área de declaración de variables' y declare las cuatro variables utilizadas (I0,I1,I2 e I3), para ello consulte el paso 7 del laboratorio#1, debe quedar como muestra la figura 12.

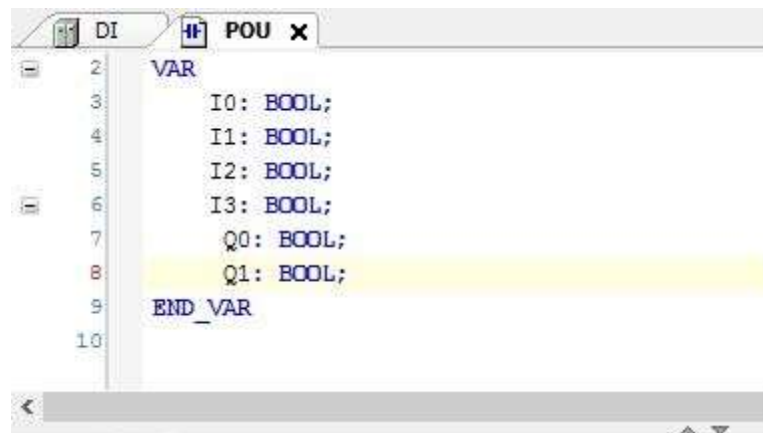


Figura 12. Declaración de las variables asociadas a los canales digitales de entradas y salidas

**Paso 6-** Asignar una variable a cada canal de salida digital. También como es necesario en la técnica operatoria usaremos dos salidas digitales (Q0 y Q1) de las 16 salidas digitales con que cuenta nuestro PLC; para asociar esas dos salidas digitales a dos variables, vamos a seguir el paso 4, debe quedar como se muestra en la figura 13.

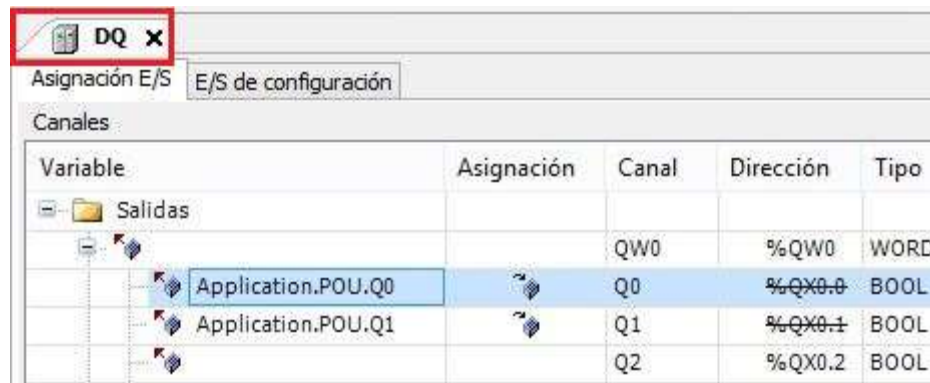


Figura 13. Asignación de la variable con un canal de salida digital

**Paso 7-** Declarar las variables que se le asignaron al canal de salida digital. Declarar en el 'Área de declaración de variables', las dos variables utilizadas (Q0 y Q1), para ello consulte el paso 7 del laboratorio#1, debe quedar como muestra la figura 14.



Figura 14. Área de declaración de variables y programación.

Ya tenemos los dos canales de salidas digitales dependiendo del estado de las variables que se les asignaron. Ahora vamos a realizar las instrucciones de código en el lenguaje de contactos (LD), para ellos vamos a seguir los siguientes pasos:

**Paso 8-** Seleccionar la red 1 (línea de código 1). En la interfaz de programación mostrada en el paso 7 del laboratorio#1 ir al 'Área de programación' y seleccionar la red 1 que se muestra en la figura 14, hay que seleccionarla para que se activen los operadores de códigos en LD que se encuentran situados en la 'Barra de herramientas' y así poder utilizarlos.

**Paso 9-** Insertar contactos (entradas) y bobinas (salidas). Como se muestra en la figura 15, dar clic en insertar contacto (dos contactos uno para la entrada I0 y otro para la entrada I1), e insertar bobina (para la salida Q0).

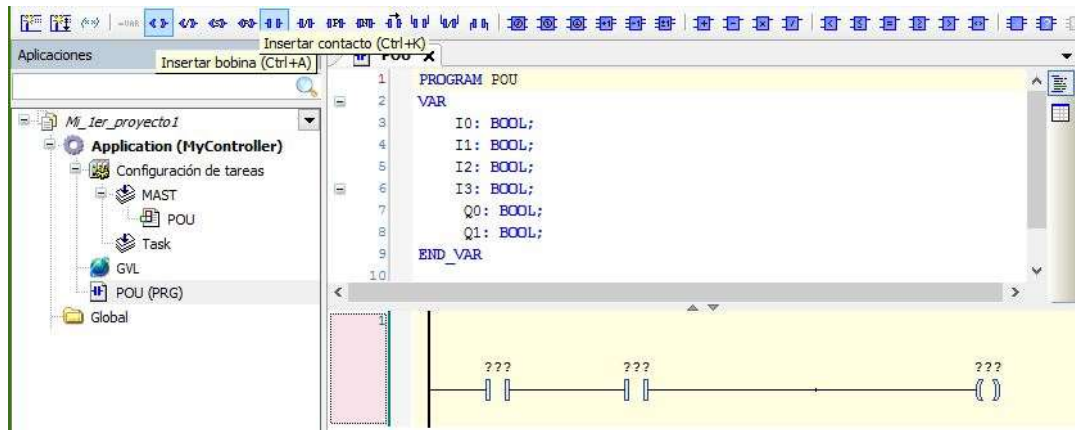


Figura 15. Insertar contactos y bobinas

**Paso 10-** Establecer variable. El paso que sigue se muestra en la figura 16 y consiste en establecer la variable que se pondrá en los contactos insertados (variables de entrada) y en la bobina insertada (variable de salida), que como se explicó anteriormente estas variables que se usan en el POU están asignadas a entradas y salidas digitales. Para escoger dicha variable, se da clic en el símbolo '...' que aparece a lado del contacto o de la bobina, esto abre la pestaña 'Accesibilidad' en donde se da clic en el elemento 'Variables' que está a la izquierda de la pestaña y a la derecha aparecen todas las variables declaradas en el POU, y ahí escogemos la de nuestro interés y damos 'Aceptar'.

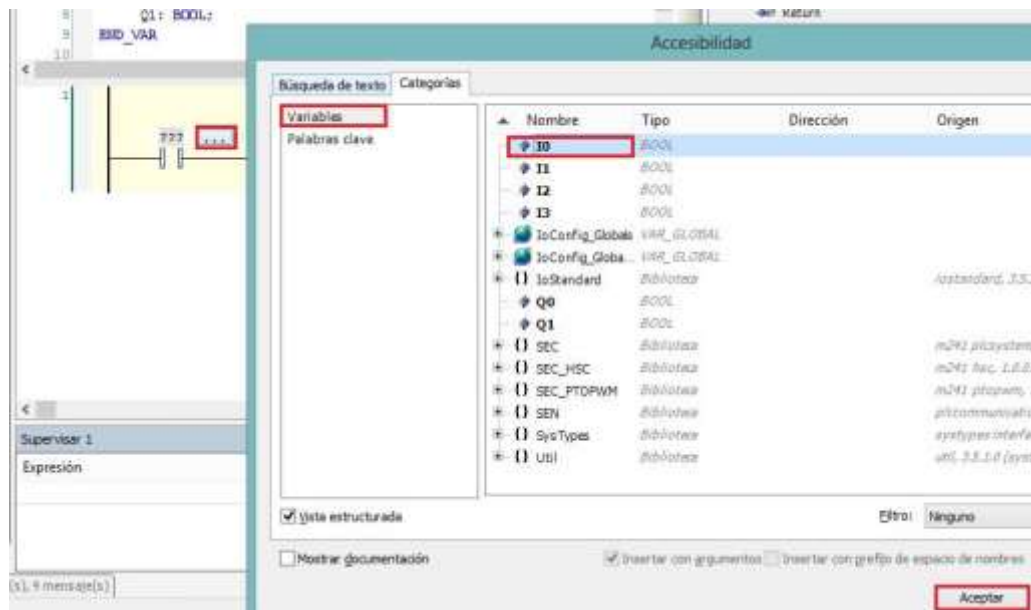


Figura 16. Establecer la variable



Como se muestra en la figura 17, nos queda implementada nuestra primera línea de código que corresponde al AND de la figura 1.

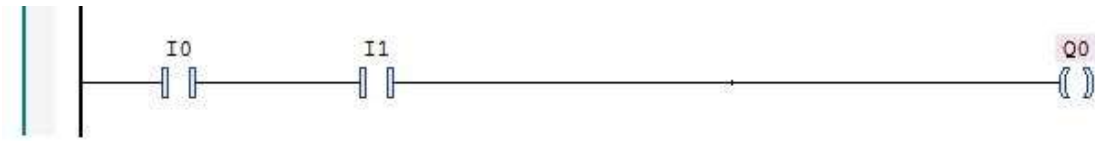


Figura 17. AND en lenguaje de contactos

**Paso 11-** Para la realización de la segunda línea de código es necesario seleccionar la red 1 y dar clic derecho, entonces seleccionar 'Insertar red (por debajo)', como se observa en la figura 18.

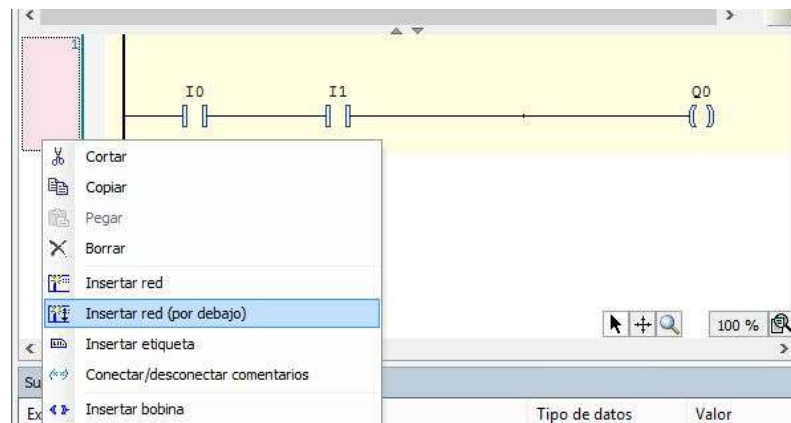


Figura 18. Insertar red

**Paso 12-** Luego como los dos contactos (para la entrada I2 e I3) que se quieren insertar están en paralelos se inserta uno y se hace clic derecho sobre él, como se muestra en la figura 18, y se selecciona 'Insertar contacto paralelo (por debajo)' y después se inserta la bobina (para la salida Q1), el resultado es el OR que se muestra en la figura 19.

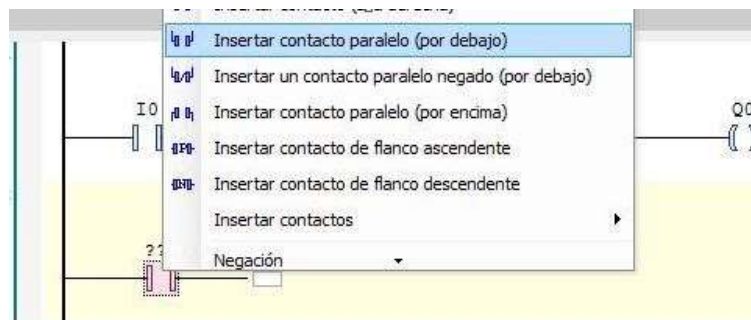


Figura 18. Insertar contacto paralelo



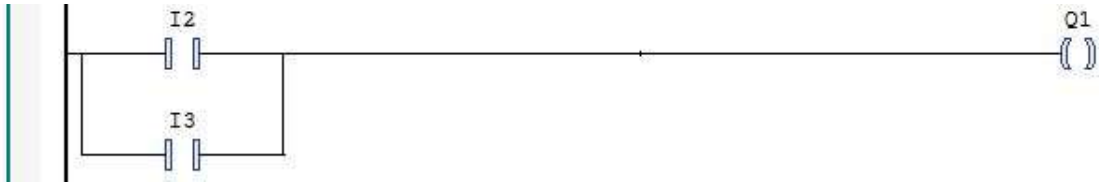


Figura 19. OR en lenguaje de contactos

**Paso 13-** Salvar y Compilar para ver los errores, mensajes y advertencias. En la figura 20 se muestra como salvar el proyecto.



Figura 20. Guardar proyecto

Para compilar vaya a la barra de trabajo que se muestra en la figura 21 y realice lo indicado en ella.

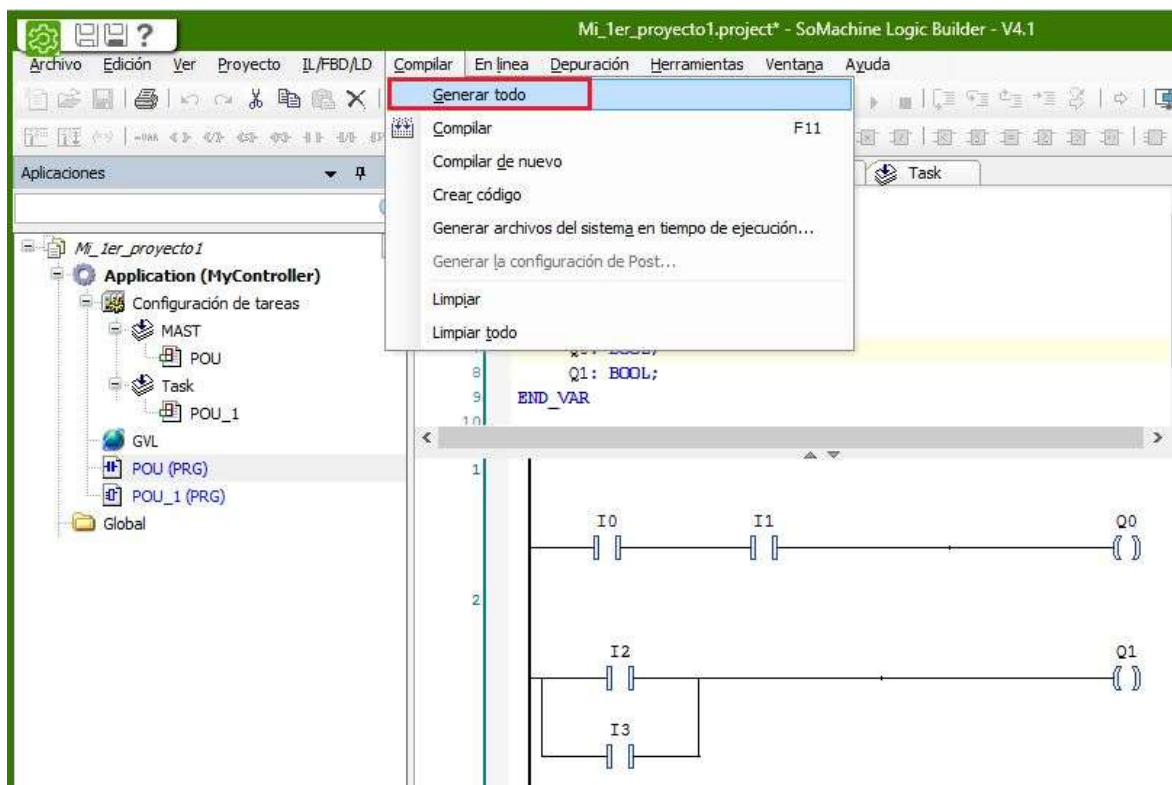


Figura 21. Compilar proyecto

## 2- Realice la operación lógica AND mostrada en la figura 1 en el lenguaje de diagramas de bloques funcionales (FBD).

**Paso 1-** Agregamos un nuevo POU, como se realiza en el paso 2 de la técnica operatoria 1, y seleccionamos función de diagrama de bloques (FBD) como su lenguaje de implementación.

**Paso 2-** Agregar la llamada a una nueva tarea, como se realiza en el paso 3 de la técnica operatoria 1.

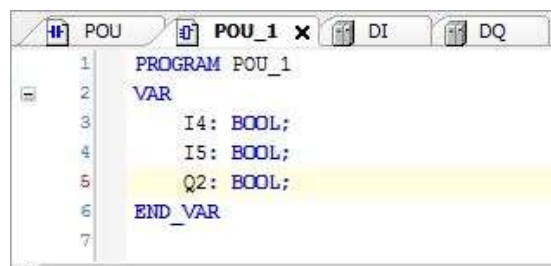
**Paso 3-** Asignar una variable a cada canal de E/S digital. Como es necesario utilizar dos entradas y una salida usaremos las entradas digitales (I4 e I5) y la salida digital (Q2) procedimiento se muestra en la paso 4 y el paso 6 de la técnica operatoria 1; tener en cuenta que el formato para nombrar la variable es 'Application.nombredelPOU.nombredelavariablen', el resultado se muestra en la figura 22.



Variable	Asignación	Canal
idwDI_IDW0		IDW0
Application.POU.I0		I0
Application.POU.I1		I1
Application.POU.I2		I2
Application.POU.I3		I3
Application.POU_1.I4		I4
Application.POU_1.I5		I5

Figura 22. Asignación de entradas digitales

**Paso 4-** Declarar las variables asignadas a cada canal de E/S digital, el procedimiento se muestra en el paso 5 y paso 7 de la técnica operatoria 1, el resultado debe quedar como se muestra en la figura 23.



```

PROGRAM POU_1
VAR
  I4: BOOL;
  I5: BOOL;
  Q2: BOOL;
END_VAR
  
```

Figura 23. Declaración de las variables

**Paso 5-** Seleccionar la red 1 como se realiza en el paso 8 de la técnica operatoria 1.

**Paso 6-** Ir a la barra de herramientas que se muestra en la figura 24, donde se muestra los operadores lógicos del lenguaje de bloques funcionales y seleccionar el AND, luego establecer la variable en cada una de las dos entradas y en la salida del AND.

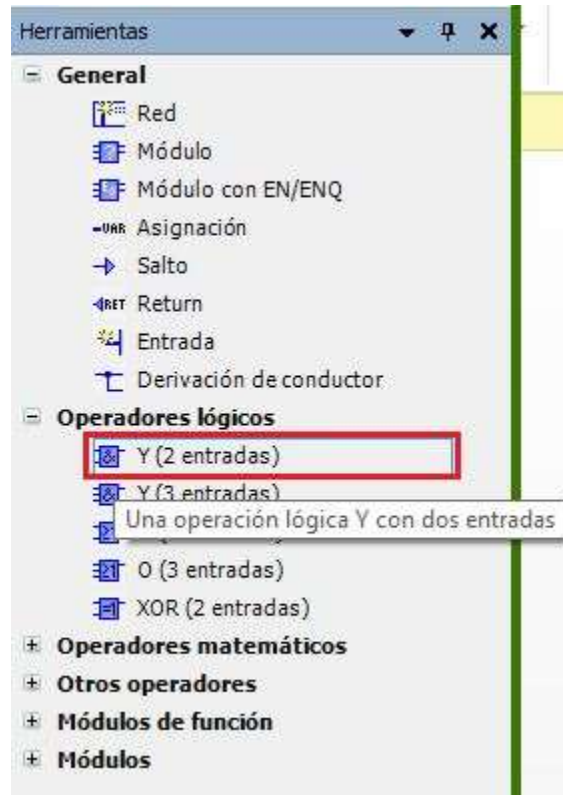


Figura 24. Selección del AND

En la figura 25 se puede observar cómo queda implementada el AND en FBD.

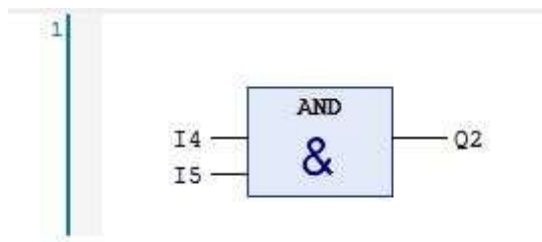


Figura 25. AND en diagrama de bloques funcionales

**Paso 7-** Salvar y Compilar. Para ello Repita el paso 13.

**Conclusiones:**

¿Cuáles son los lenguajes de implementación más utilizados en el soMachine?

¿Qué diferencias encuentra entre el lenguaje de contactos (LD) y el diagrama de bloques funcionales (FBD)?

¿Cuántas entradas y salidas digitales tiene el autómatas M241 y qué nos permite dichas E/S digitales?

**Bibliografía:**

1-SoMachine. Guía de programación, abril del 2014. Disponible en:

<https://moodle.uclv.edu.cu>

2-Manual de Ejercicios. SoMachine v4.1. Curso de Certificación.

Disponible en: <https://moodle.uclv.edu.cu>

3-SoMachine. Manual de formación. Disponible en: <https://moodle.uclv.edu.cu>

4-InfoPLC 2017. Manuales y ejemplos programación autómatas, PLC, HMI.

Disponible en:

<http://www.infoplc.net/descargas/240-schneider-electric/automatas/m241-m251>

**Comprobación de los conocimientos adquiridos:**

Realice los ejercicios del 1 al 4 de la comprobación de los conocimientos adquiridos en el laboratorio, elabore un informe y súbalo a la carpeta de la asignatura en el Moodle.

**Ejercicio 1**

Realizar  $I0 = I2 \text{ XOR } I3$ .

(La operación lógica XOR aplicada a dos variables booleanas da un resultado cierto sólo cuando una de las dos variables es cierta)

**Ejercicio 2**

Para la elevación de una carga en un taller de maquinado se utiliza una grúa viajera cuya estación de mando consta de tres botones. Utilizar un autómatas para el control del motor donde la entrada I0 es usada para arrancar el motor en el sentido de subida, la entrada I1 se

utiliza para arrancar el motor en sentido inverso y I2 es la entrada de emergencia y se usa para detener al motor. Los magnéticos del motor son manejados por las salidas Q1 para la subida y Q2 en la operación de bajada.

### **Ejercicio 3**

Cuando un push\_Button es presionado (entrada I0) la salida Q0 es activada, esta será desactivada al volver a presionar el push\_Button.

### **Ejercicio 4**

Activar la salida Q5 si las entradas I1 y I2 se encuentran activas, activar la salida Q4 para la combinación de las entradas I1 y I3 activas y activar la salida Q3 si se cumple que I1 y I4 se encuentran activas.

## Anexo VI Práctica No.3 Instrucciones de aplicación.

### Tema # 1 Formación básica en la programación de Autómatas Modicon.

#### Sumario:

Instrucciones matemáticas y de comparación.

Instrucciones de conteo y temporización.

#### Objetivos:

Utilizar las instrucciones matemáticas y de comparación en el lenguaje de contactos (LD).

Utilizar instrucciones de temporización y conteo en el lenguaje de diagrama de bloques funcionales.

#### Materiales y equipos:

Computadora con sistema operativo Windows y Software SoMachine.

Entrenador de autómatas Schneider Electric.

#### Conocimientos previos:

Características esenciales del M241 y el software de programación. (Conferencias 1 y 2 de la asignatura)

Operaciones lógicas. (Electrónica Digital)

Lenguajes de programación de Autómatas. (Medios Técnicos de Automatización)

#### Técnica Operatoria:

- 1- Realice las operaciones lógicas que se muestran en la figura 3, 4 y 5, haga una simulación offline.

var1=5 var2=10

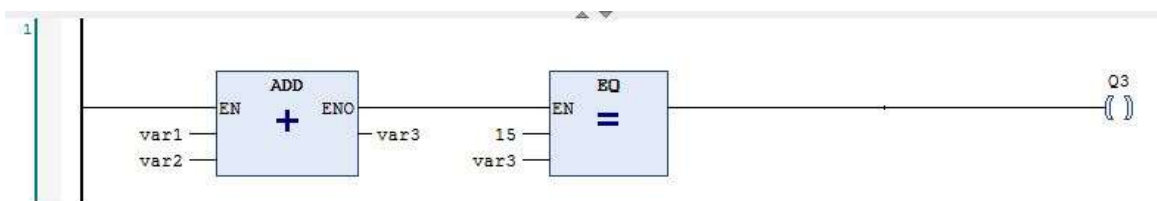


Figura 3. Línea de código 1.

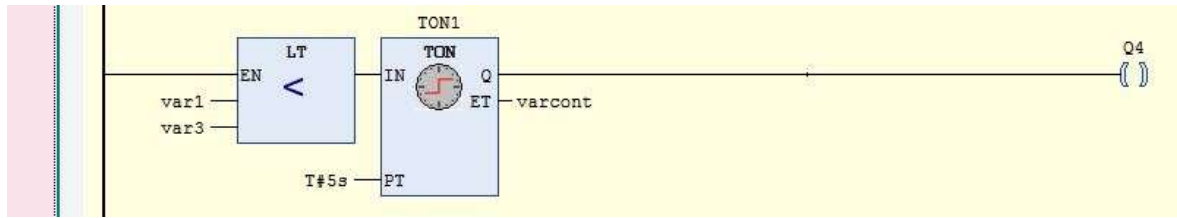


Figura 4. Línea de código 2.

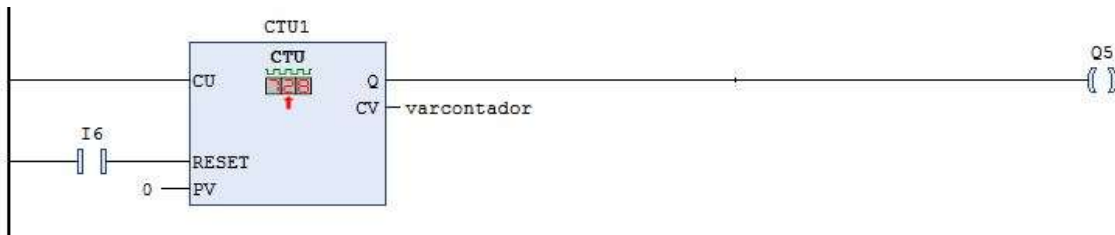


Figura 5. Línea de código 3.

Para llevar a cabo esta tarea se debe seguir los siguientes pasos: (paso 1 al 8) son para la línea de código 1 que se muestra en la figura 3, del paso 9 al 12 para la línea de código 2 que se muestra en la figura 4 y el resto para la línea de código 3 que se muestra en la figura 5.

**Paso 1-** Añadir un nuevo POU al proyecto, añadir llamada a una tarea, asignar una variable a cada canal de E/S digital (se usan 1 variable de entrada y 3 variables de salida), se muestra en el paso 2, 3, 4 y 6 de la técnica operatoria 1 del laboratorio#2.

**Paso 2-** Declarar las variables, en la figura 6 se observa otra manera de declarar una variable, después de tener el bloque de código en el área de trabajo se escribe el nombre encima del signo de '???' o del signo '...' que se muestra en la figura 7, luego se presiona *Enter* y nos sale una pestaña titulada 'Declarar variable', la ventaja de esta pestaña respecto a declarar la variable de forma manual estudiada en el paso 7 de la técnica operatoria 1 del laboratorio#2 es que da facilidades como la de escribir un valor inicial a la variable, además de que elimina la posibilidad de cometer un error en la escritura del código, las variables creadas se muestran en el área de declaración de variables como se observa en la figura 8.

Figura 6. Declarar variable

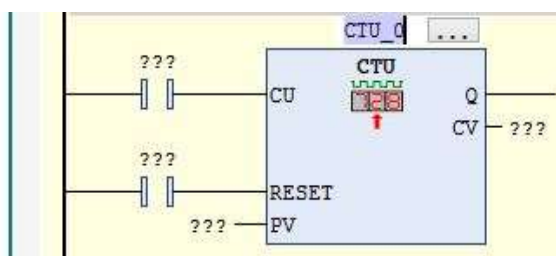


Figura 7. Establecer las variables

```

1  PROGRAM POU_2
2  VAR
3      Q3: BOOL;
4      Q4: BOOL;
5      Q5: BOOL;
6      var1: INT := 5;
7      var2: INT := 10;
8      var3: INT;
9      I6: BOOL;
10     TON1: TON;
11     varcont: TIME;
12     varcontador: WORD;
13     CTU1: CTU;
14 END_VAR

```

Figura 8. Declarar variables

**Paso 3-** Seleccionar la red 1, se muestra en el paso 7 de la técnica operatoria 1 del laboratorio#2.



**Paso 4-** Agregar el bloque ADD, para ello ir al área de ‘Herramientas’ que se muestra en la figura 9 y buscar el bloque ADD, el cual es el primer elemento de la lista de operadores matemáticos, entonces lo arrastramos a la red 1(línea de código 1).



Figura 9. Seleccionar bloque ADD.

**Paso 5-** Agregar el bloque EQ que es un comparador de igualdad, lo haremos de la misma forma que en el paso anterior, el resultado de este paso y el anterior se muestra en la figura 10.

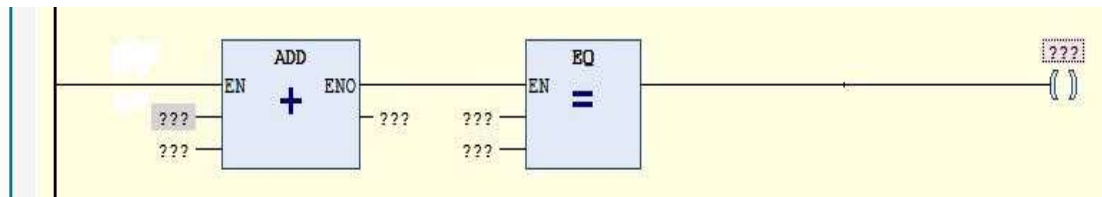


Figura 10. Bloque de igualdad

**Paso 6-** Colocarle el nombre e inicialización a cada variable, para ello consulte el paso 2, el resultado se muestra en la figura 3.

**Paso 7-** Salvar y Compilar, para ello consulte el paso 13 de la técnica operatoria 1 del laboratorio#2.

**Paso 8-** Simulación. Luego de haber salvado y compilado nuestro programa, podemos hacer una simulación offline. En la barra de menú general de la ventana de programación, seleccionar En línea » Simulación, una vez seleccionado, se está en modo simulación hasta que se deshabilite esta opción. Seleccionar 'En línea » Iniciar la sesión' para conectarse al simulador y poder transferir la aplicación en caso de estar conectado al autómatas o simplemente para poder simular el proyecto, esta ventana se observa en la figura 11.

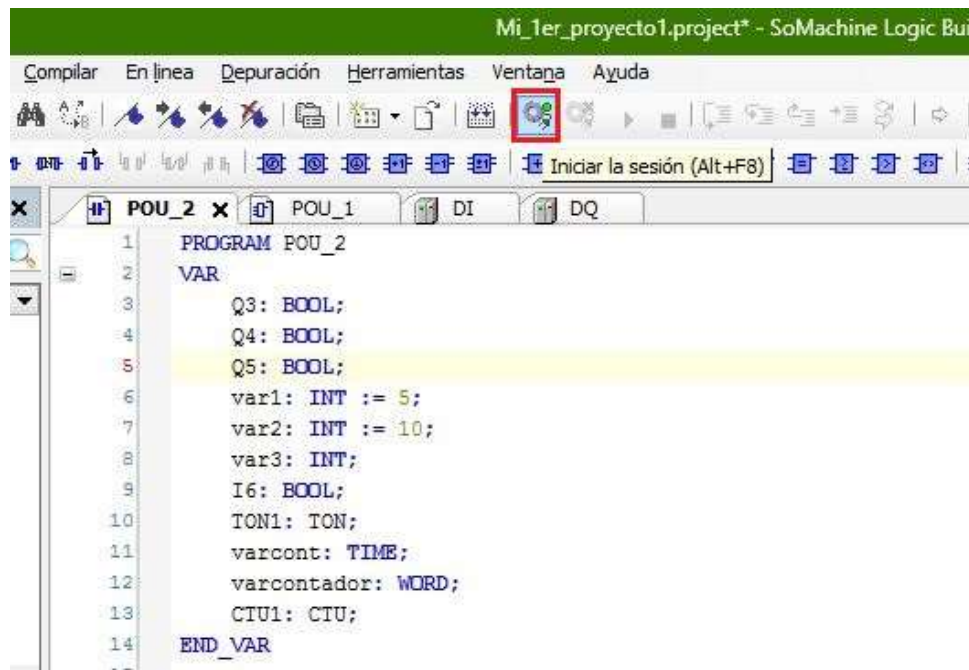


Figura 11. Iniciar sesión para simular

Como se observa en la figura 12, el modo de simulación se indica en el software con un rectángulo rojo con la palabra 'SIMULACIÓN' situada en la línea de información en la parte inferior de la ventana. Además, en la ventana de dispositivos el nombre de controlador se pone en letras cursivas.

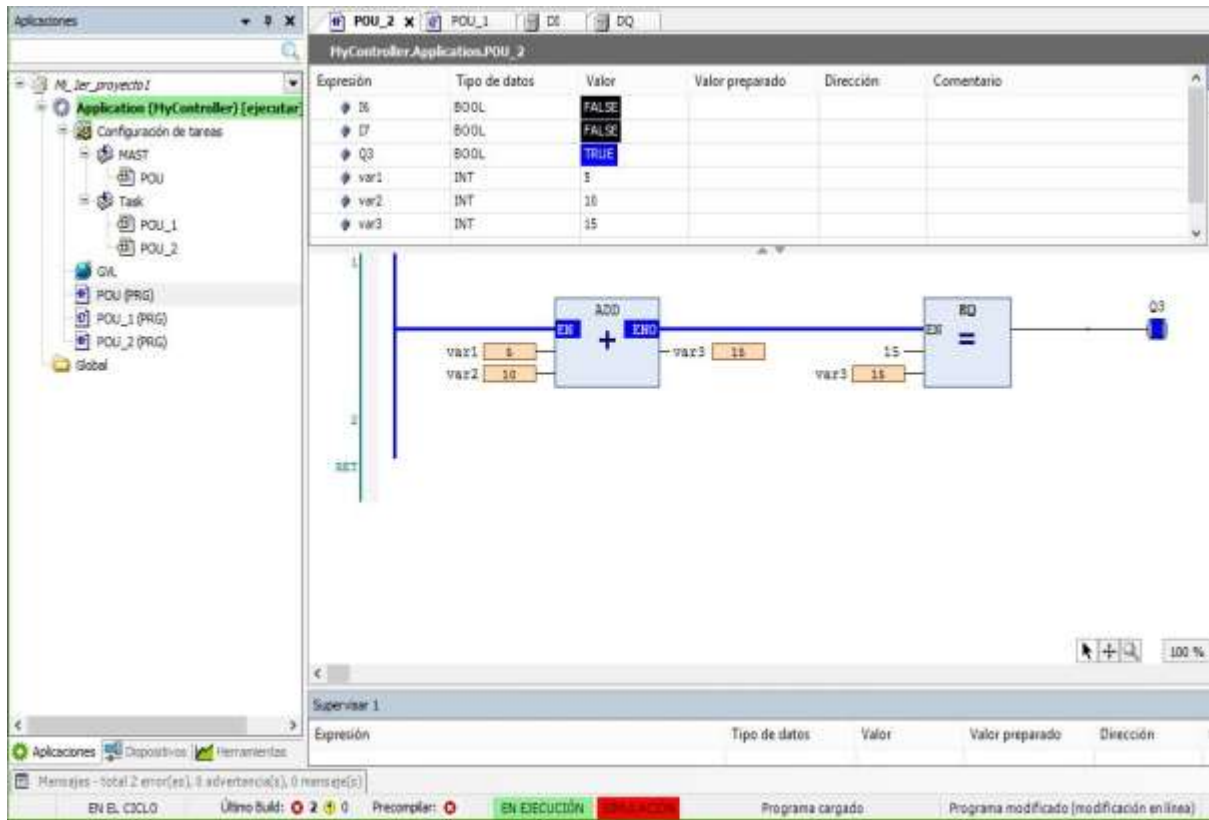


Figura 12. Simulación de la línea de código 1

En la figura 12 se puede observar como el bloque ADD, suma los dos números inicializados en las variables var1 y var2, el resultado lo escribe en la var3 y como el bloque EQ compara la igualdad de numero 15 con el valor de la var3 resultante de la suma del bloque ADD, y como se cumple la igualdad se activa la salida digital Q3.

**Paso 9-** Para la realización de la segunda línea de código es necesario seleccionar la red 1 y dar clic derecho, entonces seleccionar 'Insertar red (por debajo)', como se observa en la figura 18 del laboratorio#2.

**Paso 10-** Agregar el bloque LT (menor que), para ello ir al área de 'Herramientas' que se muestra en la figura 9 y buscar el bloque en la lista de operadores matemáticos, entonces lo arrastramos a la red 2.

**Paso 11-** Agregar el bloque TON y la salida Q4, para ello ir al área de 'Herramientas' que se muestra en la figura 9 y buscar en la lista de 'Elementos de diagramas de contactos' el bloque y luego arrastrarlo a línea de código, el resultado se muestra en la figura 13.

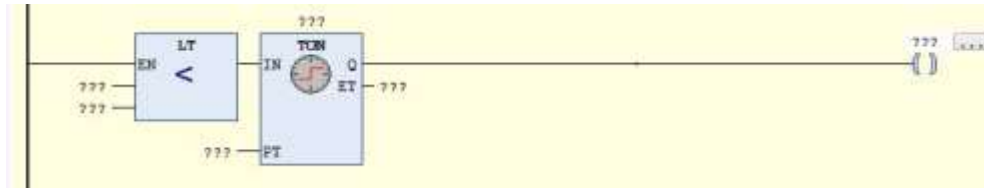


Figura 13. Bloque de comparación

**Paso 12-** Colocarle el nombre e inicialización a cada variable, se muestra en el paso 6. El resultado de la segunda línea de código se muestra en la figura 4. Luego seguir los pasos 7 y 8, la simulación de esta línea se observa en la figura 14.

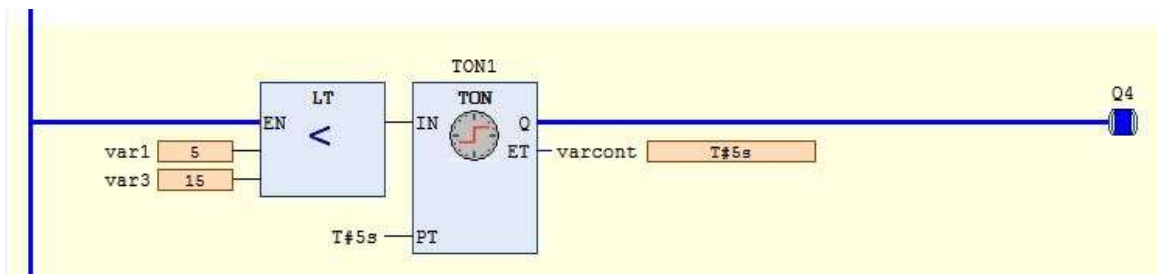


Figura 14. Simulación de la segunda línea de código

En la figura 14 puede observarse que el bloque comparador LT 'menor que', compara los números inicializados en las variables var1 y var2, y siempre que  $var1 < var2$  se activa su salida y habilita el bloque temporizador TON el cual como su señal de entrada IN está activa en 1 empieza a contar en la variable ET el tiempo transcurrido hasta llegar a los 5 segundos establecidos en la variable PT y sólo en ese instante el activa su salida Q, la cual activa la salida Q4.

**Paso 13-** Para la realización de la tercera línea es necesario 'Insertar red (por debajo)' de la red 2, como se observa en la figura 18 del laboratorio#2.

**Paso 14-** Agregar el bloque CTU (contador ascendente), que se muestra en la figura 7, para ello ir al área de trabajo que se muestra en la figura 14 y buscar el bloque en la lista de operadores matemáticos, entonces lo arrastramos a la red 3.

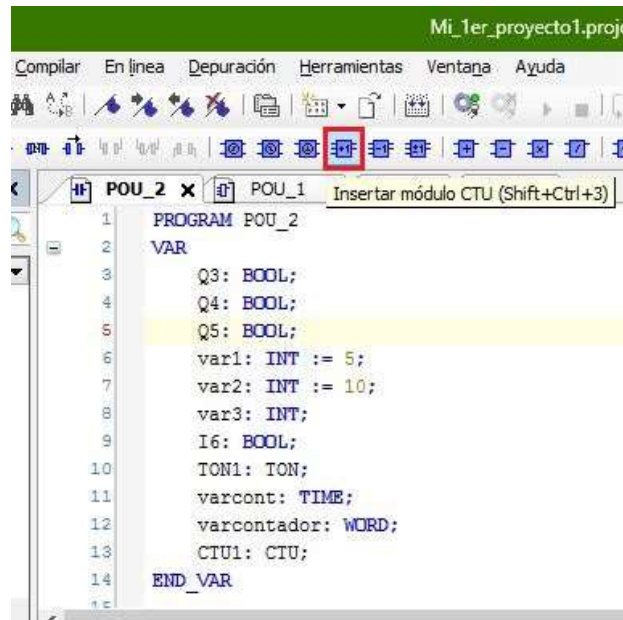


Figura 15. Agregar contador CTU

**Paso 15-** Agregar la salida Q5 y colocarle el nombre e inicialización a cada variable, como se muestra en el paso 6. El resultado de la tercera línea de código se muestra en la figura 5. Luego seguir los pasos 7 y 8, la simulación se observa en la figura 16.

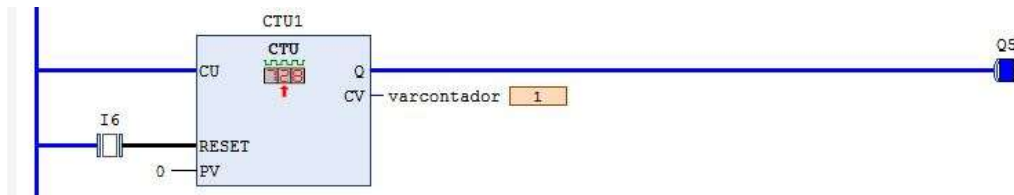


Figura 16. Agregar salida Q5

En la simulación mostrada en la figura 16 se puede observar como al activarse el borde de subida de la señal CU el contador empieza a contar incrementándose el valor del CV, la salida Q es verdadera siempre que el  $CV \geq PV$ , y siempre que Q sea verdadera, se activa la salida Q5.

### Conclusiones:

¿Qué facilidades ofrecen las instrucciones matemáticas y de comparación en la programación de autómatas?

¿Cuáles son los diferentes tipos de temporizadores y contadores, y en qué se diferencian?

**Bibliografía:**

1-SoMachine. Guía de programación, abril del 2014. Disponible en:

<https://moodle.uclv.edu.cu>

2-Manual de Ejercicios. SoMachine v4.1. Curso de Certificación.

Disponible en: <https://moodle.uclv.edu.cu>

3-SoMachine. Manual de formación. Disponible en:

<https://moodle.uclv.edu.cu>

4-InfoPLC 2017. Manuales y ejemplos programación autómatas, PLC, HMI.

Disponible en:

<http://www.infoplac.net/descargas/240-schneider-electric/automatas/m241-m251>

**Comprobación de los conocimientos adquiridos:**

Realice los ejercicios del 1 al 3 de la comprobación de los conocimientos adquiridos en el laboratorio, elabore un informe y súbalo a la carpeta de la asignatura en el Moodle.

**Ejercicio 1**

Un garaje dispone de 25 plazas libres. Se desea que:

- Mientras el número de plazas libres sea mayor o igual a 5 deberá lucir en la entrada del garaje una luz verde (Q0).
- Cuando el número de plazas libres sea inferior a 5 se deberá activar una señal de aviso (Q1).
- Por último, una vez que el garaje este totalmente ocupado se deberá iluminar una luz roja y apagar la señal de aviso.

Observe la tabla 1 y la figura 17.

Tabla 1. Entradas/Salidas

Símbolo	Descripción
I0	Sensor en la entrada del garaje

I1	Sensor en la salida del garaje
I2	Pulsador para iniciar el estado del garaje
Q0	Luz verde
Q1	Señal de aviso
Q2	Luz roja
CTUD	Contador ascendente-descendente

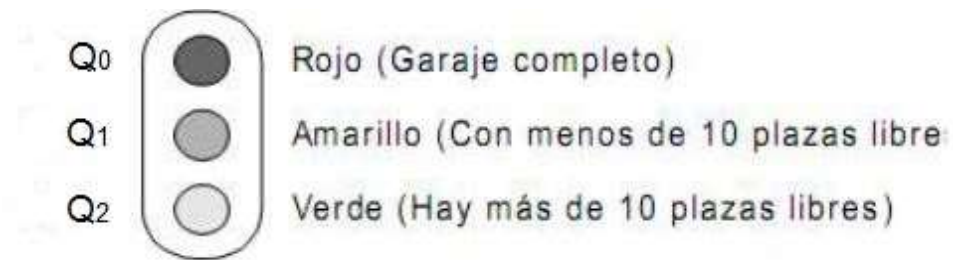


Figura 17. Señalización de luces

## Ejercicio 2

Control de la acción de un brazo motorizado.

La figura 18 representa una máquina que tiene un brazo motorizado. Cuando se pulsa el botón de arranque con el brazo en la posición 1, el brazo gira en sentido horario y detiene su rotación una vez que ha llegado a la posición 2. Transcurridos 5 segundos, el brazo gira en sentido anti-horario hasta la posición 1 y se para. El ciclo se puede repetir de nuevo, cuando se pulse el botón de arranque.

I0 Botón de arranque

I1 Posición 1

I2 Posición 2

Q0 Giro en sentido horario

### Q1 Giro en sentido anti-horario

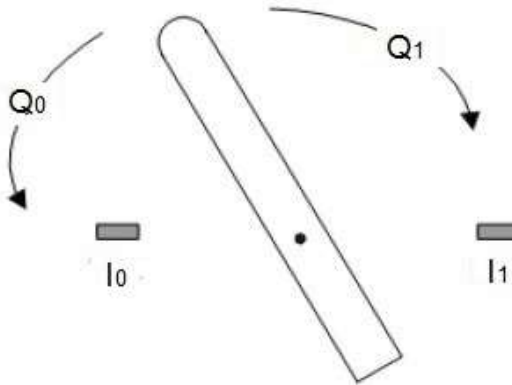


Figura 18. Máquina que tiene un brazo motorizado

### Ejercicio 3

Alumbrado de una escalera temporizada.

Este ejemplo de programa sirve para el alumbrado de una escalera. El pulsador que se muestra en la figura 19 está conectado sobre la entrada I0 del control. Tras accionar un pulsador de conexión, se conecta la luz durante un tiempo de 30 segundos, y en ese momento se activa la salida Q0 prevista. Si durante este tiempo se acciona de nuevo un pulsador de conexión, el intervalo de tiempo comienza de nuevo desde el principio. De esta forma se asegura que la luz se apaga transcurridos 30 segundos después de la última pulsación del pulsador.

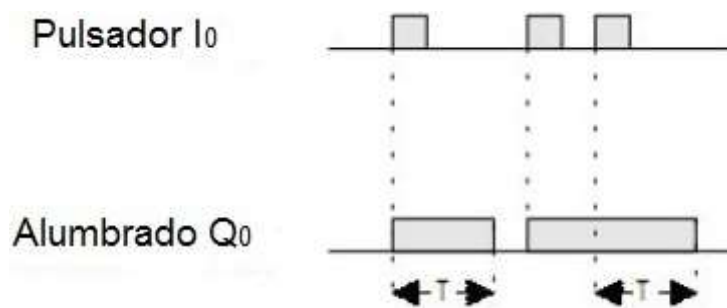


Figura 19. Pulsador



**Anexo VII Práctica No.4 Adquisición de Variable analógica.****Tema # 2 Formación avanzada en la programación de Autómatas Modicon.****Sumario:**

Conexión de sensores pasivos al módulo analógico del autómata.

Configuración de la comunicación sensor-autómata.

Programación del módulo analógico TM3AM6.

**Objetivos:**

Conocer las entradas y salidas analógicas de los módulos TM3XXX los autómatas Modicon M2XX.

Instalar sensores pasivos a los autómatas Modicon M2XX.

Configurar y programar los módulos analógicos TM3XX.

Medir valores de temperatura, presión y nivel con sensores pasivos.

**Materiales y equipos:**

Computadora personal con sistema operativo Windows y software SoMachine.

Sensores y transductores de temperatura, presión y nivel.

Entrenador de la firma Schneider Electric.

**Conocimientos previos:**

Características técnicas y principio de funcionamiento de sensores-transductores: Pt100 y Cerabar. (Mediciones)

Operaciones lógicas. (Electrónica Digital)

Lenguajes de programación de Autómatas. (Medios Técnicos de Automatización)

**Técnica Operatoria:**

**1- Instalar el sensor Pt100, transmisor SEM1503P y los transductores Cerabar T PMC131 en el entrenador de la firma Schneider Electric.**

Para la realización de esta técnica operatoria vaya a la Conferencia 5. El resultado del cableado en el módulo de E/S analógicas debe quedar como muestra la figura 1, para ello observe la figura 5 del laboratorio#1 donde se muestra el diagrama de cableado de este módulo para conocer cuáles son los pines correspondientes a las E/S analógicas de los dispositivos.

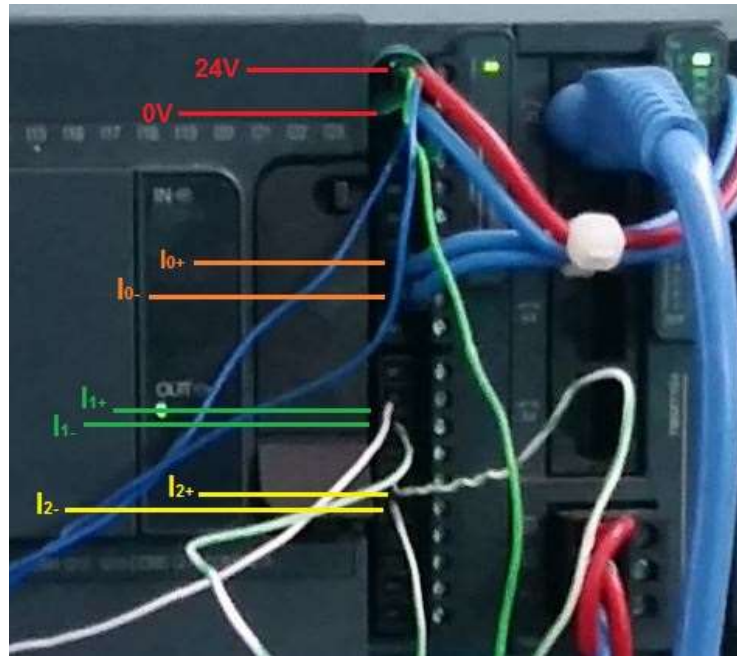


Figura 1. Cableado del módulo TM3AM6/G

**2- Instalar los sensores y transductores en los procesos correspondientes: “Tanque de agua”, “Calibrador de peso muerto” y “Baño de María”.**

Instale los sensores y transductores en los procesos correspondientes, en caso de los transductores Cerabar T PMC131 fijarse bien en el rango de medición, en el “Calibrador de peso muerto” va el Cerabar T PMC131 que tiene campo de medida de 0 a 2Bar, y en “Tanque de agua” va el que tiene campo de medida de 0 a 100mBar. El resultado de la instalación puede observarse en los anexos III, IV y V.

**3- Configurar el módulo de E/S analógicas TM3AM6/G para la adquisición de las variables analógicas, temperatura, presión y nivel.**

**Paso 1-** Llevar a cabo los pasos del 1 al 3 del laboratorio#2.

**Paso 2-** Agregar al POU el módulo de E/S analógicas TM3AM6/G como se muestra en el paso 8 del laboratorio#1 “Agregar Módulos de Ampliación”, en la figura 2 se observa como buscar y agregar este módulo.

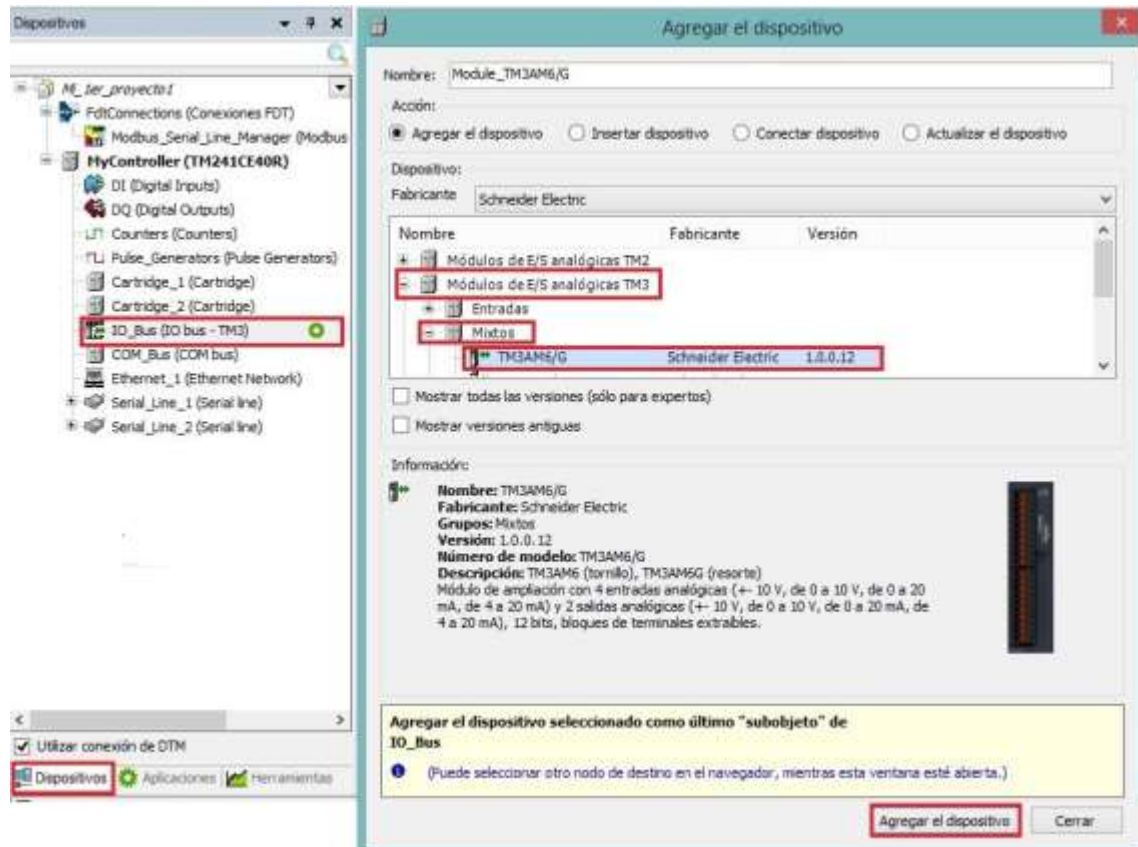


Figura 2. Agregar módulo TM3AM6/G

En la figura 2 también puede observarse como al seleccionar el módulo se muestra una descripción de el mismo, conociéndose así que este cuenta con 4 entradas analógicas (IW0 a IW3) y con 2 salidas analógicas (QW0 y QW1).

En la figura 3 se muestra que Debajo del ‘IO\_Bus (IO bus – TM3)’ en la pestaña ‘Dispositivos’ en el ‘Navegador’, aparecerán representados todos los módulos de expansión añadidos.



Figura 3. Representación de los módulos añadidos

### Paso 3- Configurar el módulo de E/S analógicas TM3AM6/G.

Si dentro del 'Navegador' hacemos doble clic en uno de los módulos de ampliación añadidos, en el área de trabajo aparece la ventana de configuración del módulo, como se observa en la figura 4.

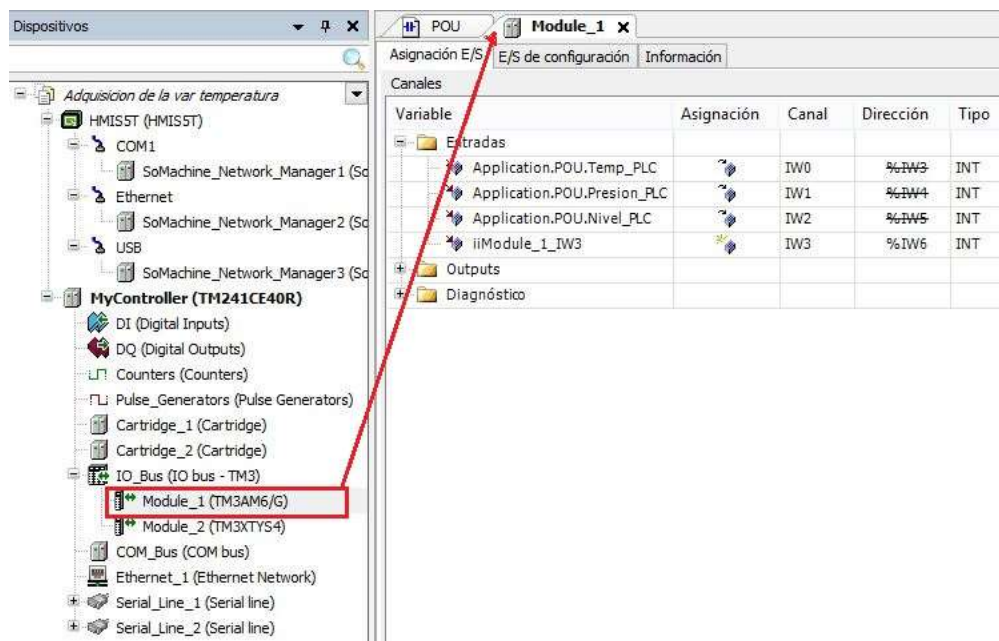


Figura 4. Configuración del módulo analógico

En la pestaña ‘Asignación de E/S’ podemos asignarle una variable a cada una de las entradas ó salidas físicas del módulo, escribiéndolo en el campo ‘Variable’ de cada una de ellas, para nombrar la variable seguir la siguiente estructura ‘Application.nombredelPOU.nombredelavariabla’, como se muestra en la figura 5.

En la pestaña ‘E/S de Configuración’ activaremos aquellas entradas/ salidas analógicas que vamos a utilizar y la configuraremos, como se muestra en la figura 5.

POU		Module_1 X				
Asignación E/S		E/S de configuración				
Parámetro		Tipo	Valor	Valor predeterminado	Unidad	Descripción
Entradas						
IW0						
Type	Enumeration of BYTE	4 - 20 mA	No se utiliza			Modalidad derango
Minimum	INT(-32768...149)	0	-32768			Valor mínimo
Maximum	INT(1...32767)	150	32767			Valor máximo
InputFilter	INT(0..1000)	0	0 * 10 ms			Filtro de entrada
Sampling	Enumeration of BYTE	1	1 ms/Channel			Selección de muestreo de entrada
IW1						
Type	Enumeration of BYTE	4 - 20 mA	No se utiliza			Modalidad derango
Minimum	INT(-32768...1999)	0	-32768			Valor mínimo
Maximum	INT(1...32767)	2000	32767			Valor máximo
InputFilter	INT(0..1000)	0	0 * 10 ms			Filtro de entrada
Sampling	Enumeration of BYTE	1	1 ms/Channel			Selección de muestreo de entrada
IW2						
Type	Enumeration of BYTE	4 - 20 mA	No se utiliza			Modalidad derango
Minimum	INT(-32768...999)	0	-32768			Valor mínimo
Maximum	INT(1...32767)	1000	32767			Valor máximo
InputFilter	INT(0..1000)	0	0 * 10 ms			Filtro de entrada
Sampling	Enumeration of BYTE	1	1 ms/Channel			Selección de muestreo de entrada
IW3						

Figura 5. Configuración de entradas/salidas

Para configurar una entrada analógica, primero tenemos que determinar que tipo de señal analógica nos llega (eso dependerá del tipo de sensor), en el campo ‘Tipo’ en la columna ‘Valor’, seleccionaremos entre (No Usada, 0 - 10V, -10 - +10 V, 0 - 20mA, 4 - 20mA). Cuando seleccionamos un tipo de señal se habilita el resto de campos de la configuración.

Mínimo: Valor mínimo de la señal (quiere decir que si hemos configurado una señal 4 - 20mA cuando a la señal física tenga los 4mA que valor numérico queremos ver en el programa, por defecto 0).

Máximo: Valor máximo de la señal (quiere decir que si hemos configurado una señal 4 - 20 mA cuando a la señal física tenga los 20A que valor numérico queremos ver en el programa, por defecto 10000).

Filtro de entrada: Valor de la ventana de tiempo de filtrado (quiere decir que para señales que lleguen con ruido y que pueden dar valores no reales debido a las interferencias, se utiliza este filtro para minimizarlo).

Muestreo: Valor de muestreo (Es el tiempo de muestreo de cada una de las señales analógicas).

Por tanto, para establecer el valor “Máximo” hay que guiarse por el rango de los sensores y transductores:

Para la variable temperatura: es 150°C, que corresponde al máximo valor del rango del sensor Pt100.

Para la variable presión es 2000mBar=2Bar que es el máximo valor del rango del transductor de presión Cerabar T PMC131 que usa en el “Calibrador de peso muerto”

Para la variable nivel es 1000mm=1metro (ya que el rango del transductor de presión Cerabar T PMC131 es 100mBar que corresponde a 1metro de columna de agua).

**4- Realizar una comparación entre un valor de la temperatura real del proceso y un valor patrón establecido (20°C), que active la salida digital “Salida\_0” (Q0) como se muestra en la figura 6.**

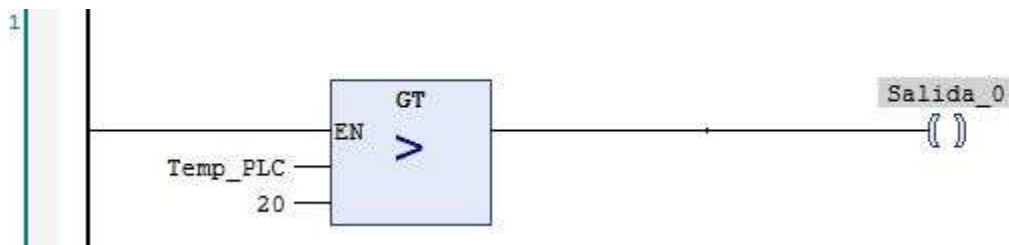


Figura 6. Código de comparación entre las temperaturas.

Para realizar la comparación entre el valor actual del proceso y la temperatura patrón (20°C), código que se muestra en la figura 6, se debe continuar a partir del paso 3 de la técnica operatoria 3 y realizar los pasos siguientes:

**Paso 1-** Declarar las variables que se le asignaron a los canales de entradas analógicas (IW0, IW1 e IW2) del módulo TM3AM6/G, en el 'Área de declaración de variables', para ello consulte el paso 7 del laboratorio #1, el resultado se muestra en la figura 7.

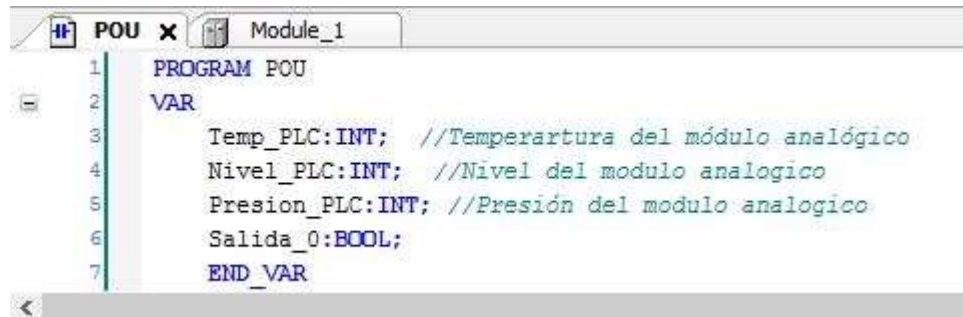


Figura 7. Declaración de las variables

**Paso 2-** Asignar una variable (Salida\_0) a un canal de salida digital (Q0), para cuando la temperatura del proceso sea mayor que el patrón de comparación que se puso en el programa (20°C) se active tal salida digital; para asignación puede auxiliarse del paso 6 del laboratorio#2, la asignación se muestra en la figura 8.

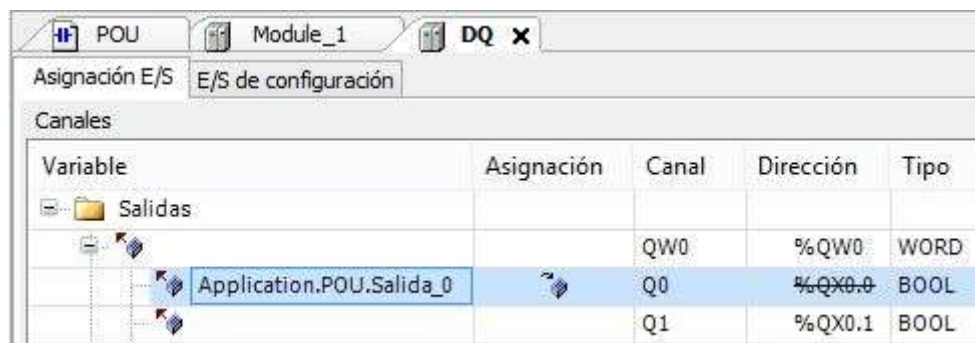


Figura 8. Asignación de entradas/salidas

**Paso 3-** Insertar la línea de código arrastrando hacia el “Área de Programación” el bloque de comparación “GT” y la bobina “Salida\_0” asignada al canal de salida digital (Q0), para ello auxiliarse del paso 9 de la técnica operatoria 1 del laboratorio#2, el resultado de este paso se muestra en la figura 9.

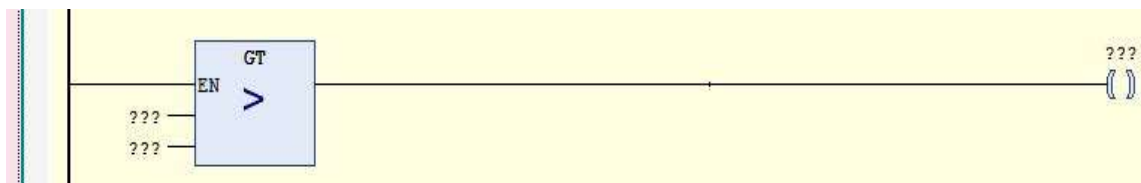


Figura 9. Código en el área de programación



**Paso 4-** Colocarle el nombre de cada variable, para ello consulte el paso 2 de la técnica operatoria 1 del laboratorio#3, y colocar el patrón de comparación (20°C), el resultado se muestra en la figura 6.

**Paso 5-** Salvar, Compilar y Simular el programa para ver posibles errores y mensajes de sugerencias, para ello consulte el paso 7 y paso 8 de la técnica operatoria 1 del laboratorio#3. El resultado de la Simulación se observa en la figura 10, donde puede observarse que el valor actual de la temperatura del proceso “Baño de María” está almacenado en la variable Temp\_PLC debido q que es la variable que asignamos al canal de entada analógico I0 del módulo de E/S analógicas TM3AM6/G. Como la temperatura del proceso es 27°C y es mayor que la temperatura patrón que establecimos (20°C) pues se activa en verdadero la Salida\_0 asignada a la salida digital Q0. También puede observarse los valores de las demás variables en los demás procesos; la presión en el “Calibrador de peso muerto” es 1000mBar=1Bar y el nivel en el “Tanque de agua” es 300mm=30cm.

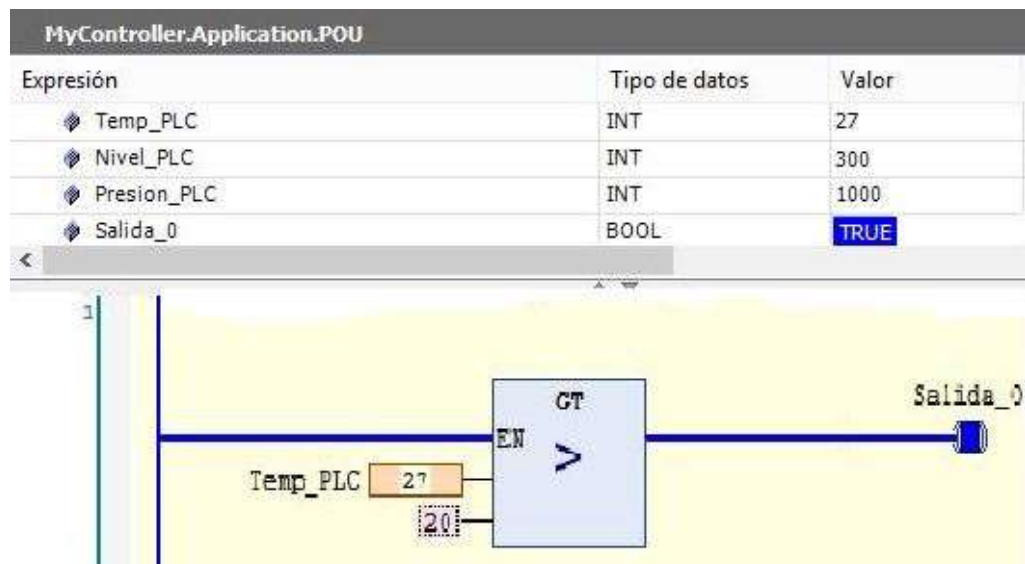


Figura 10. Simulación

## 2- Cargar el programa en el PLC y realizar la Simulación Online.

**Paso 1-** Para cargar el programa al PLC y luego realizar una simulación Online es necesario antes que todo “Conectarse al Autómata”.



La opción ‘Conectar al controlador’ nos permite escanear por Ethernet y Modbus los controladores conectados y poder conectarnos directamente a ellos y realizar tareas de mantenimiento como (Cargarle un proyecto, Descargarle un proyecto, firmware, etc.).

Cuando se inicia la página de inicio de SoMachine Central seleccionaremos la función ‘Conectar al controlador’ y seguiremos las acciones que se muestran en la tabla 1.

Tabla 1. Pasos para conectar al controlador

Paso	Acción	Comentario
1	Haga clic en el botón Conectar al controlador	Aparecerá el cuadro de dialogo seleccionar Controlador
2	La red Ethernet y los puertos USB están escaneando los controladores disponibles	La lista visualizada de controladores contiene los controladores de la red que han enviado una respuesta a la solicitud de SoMachine
3	Seleccione un controlador de la lista y haga clic en el botón “Seleccionar”	Aparecerá el cuadro de dialogo Opciones
4	Seleccione una de las siguientes opciones: Abrir archivo de proyecto (examinar) Crear un proyecto nuevo Crear u proyecto con la plantilla Cargar el proyecto del controlador Descargar el proyecto actual al controlador Utilizar herramientas de mantenimiento	
5	Haga clic en el botón Continuar	La opción seleccionada se ejecuta

En el paso 4 que se muestra en la tabla 1, seleccionamos la opción descargar el proyecto actual al controlador y luego realizamos la Simulación Online, como se puede observar en la figura 11 la salida digital Q0 está activa debido a que la temperatura del proceso es mayor que la temperatura patrón (20°C).



Figura 11. Simulación Online

### Conclusiones:

¿Qué es un módulo de E/S analógico y qué posibilidad nos brinda?

¿Qué aspectos se deben tener en cuenta para instalar sensores al autómatas?

¿Qué configuración se le debe hacer al módulo analógico del autómatas para leer la señal proveniente de un sensor pasivo?

Realice el ejercicio 1 de la comprobación de los conocimientos adquiridos en el laboratorio, elabore un informe y súbalo a la carpeta de la asignatura en el Moodle.

### Bibliografía:

1-SoMachine. Guía de programación, abril del 2014. Disponible en:

<https://moodle.uclv.edu.cu>

2-Manual de Ejercicios. SoMachine v4.1. Curso de Certificación.

Disponible en: <https://moodle.uclv.edu.cu>

3-InfoPLC 2017. Manuales y ejemplos programación autómatas, PLC, HMI.

Disponible en:

<http://www.infopl.net/descargas/240-schneider-electric/automatas/m241-m251>

4-Creus, Antonio 1995. Instrumentación Industrial. Capítulo 5, epígrafe 5.1.2. páginas 190-197. Poligráfico Evelio Rodríguez Curbelo.

### Comprobación de los conocimientos adquiridos:

1-Se desea automatizar la máquina de llenado de botellas de ron que se muestra en la figura 12 en una destilería. La misma que se muestra en la figura 3.10, funciona mediante una estera transportadora que lleva la botella hasta el lugar donde será llenada, cuando el proceso de llenado culmina la estera se mueve hasta que otra botella vacía ocupe su lugar. La estera es controlada por un motor que funciona acoplado a un magnético ('1' marcha motor '0' detiene motor). Se instalan dos sensores de presencia en la pared, SP1 para detectar si la botella está en posición ('1' botella en posición '0' no hay ninguna botella) y SP2 para saber si la botella está vacía o no ('1' botella llena '0' botella vacía o no hay botella). El resto de la máquina de llenado está compuesta por un depósito que tiene dos lazos de nivel acoplados, esta funciona de la siguiente manera. Si el depósito está vacío se cierra la válvula 1 y se abre la válvula 2 para introducir en el mismo 750 ml de ron que es la cantidad que lleva la botella. Cuando se alcanza esta cantidad se cierra la válvula 2 y se espera a que haya una botella vacía en posición para entonces abrir la válvula 1 hasta que se vacíe el depósito. Una vez llena la botella se cierra la válvula 1 y se continúa el proceso. Se tiene una botonera con dos botones y sus leds correspondientes, un botón y led verde es de encendido y el otro con led rojo es de apagado. El sensor de nivel tiene un rango de 0 a 2 litros y una salida de 4 a 20 mA. Las válvulas 1 y 2 son electrónicas y poseen un servomotor con entrada 24 volt y acción directa. Como controlador usamos un PLC S7 200 con los módulos necesarios.

1.1-Selección la CPU a utilizar y los módulos correspondientes.

1.2-Tabla de variables.

1.3-Programa del PLC.

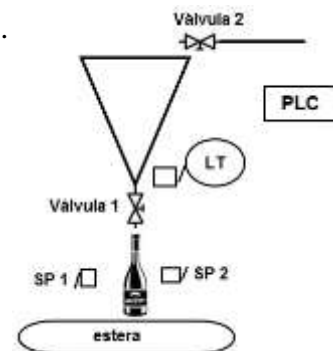


Figura 12. Máquina de llenado

**Anexo VIII Práctica No.5 Interfaz Gráfica.****Tema # 2 Formación avanzada en la programación de Autómatas Modicon.****Sumario:**

Introducción al ambiente de trabajo del software Vijeo Designer.

Desarrollo de un proyecto en Vijeo Designer.

Paneles bases en la Magelis HMISTU tipo HMIS5T y las ventanas emergentes.

Vincular las variables del Vijeo Designer con variables externas.

Descargas y comprobación de proyectos en Runtime.

**Objetivos:**

Conocer el ambiente de trabajo del software Vijeo Designer.

Configurar paneles bases en la Magelis HMISTU tipo HMIS5T.

Visualizar en el Vijeo Designer las variables del SoMachine.

Comprobar la vinculación entre el autómata y su interface gráfica.

**Materiales y equipos:**

Computadora personal con sistema operativo Windows, software SoMachine y Vijeo Designer.

Entrenador de la firma Schneider Electric.

**Conocimientos previos:**

Características esenciales del HMIS5T y del software de programación Vijeo Designer.

(Conferencias 6 de Autómatas Programables)

Comunicación entre HMI y el M241 (Conferencia 6 de Autómatas Programables)

**Técnica Operatoria:**

**1- Crear el panel base principal que se muestra en la figura 1, que permita al usuario la navegación según la variable de interés (presión, nivel y temperatura).**



Figura 1. Panel Principal

**Paso 1-** Ejecutar el programa.

Para iniciar Vijeo Designer, en el Menú de Inicio de Windows, seleccione la ruta Programas » Schneider Electric » Vijeo Designer » Vijeo Designer.

También se puede hacer doble clic en el icono de acceso rápido a Vijeo Designer que hay en el escritorio, el cual puede observarse en la figura 2.



Figura 2. Icono del Vijeo Designer

**Paso 2-** Creación de un nuevo proyecto.

La creación de un nuevo proyecto de Vijeo Designer se realiza desde el SoMachine en el momento en que añadimos un dispositivo HMI a nuestro proyecto. Por tanto, para crear un nuevo proyecto realice los pasos 2, 3, 4, 5 y 6 del laboratorio#1.

**Paso 3-** Exportar las variables desde el SoMachine al Vijeo Designer.

La posibilidad de vincular los nombres y las direcciones, símbolos y variables configuradas en el software Somachine a las variables de Vijeo-Designer. Nos permite ahorrar tiempo y la posibilidad de actualizarlas.

Para usar las variables que nos interesa supervisar en el panel HMI debemos seguir los siguientes pasos:

**Paso 3.1-** Declarar las variables a usar en la HMI como variables globales de nuestro proyecto en el software SoMachine, como se muestra en la figura 3.

Para ello vaya al “Área de navegación del proyecto” a la pestaña “Dispositivos” y seleccione el elemento “GVL”, el cual crea un acceso directo en el “Área de trabajo” donde declararemos nuestras variables globales.

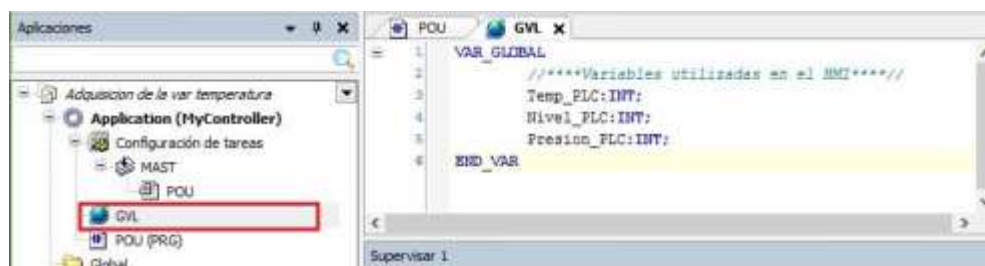


Figura 3. Declaración de variables globales

**Paso 3.2-** Seleccionar el elemento “Variables” que se encuentre en el área de las “Herramientas” para seleccionar las variables que queremos compartir con el Vijeo Designer, se muestra en la figura 4.

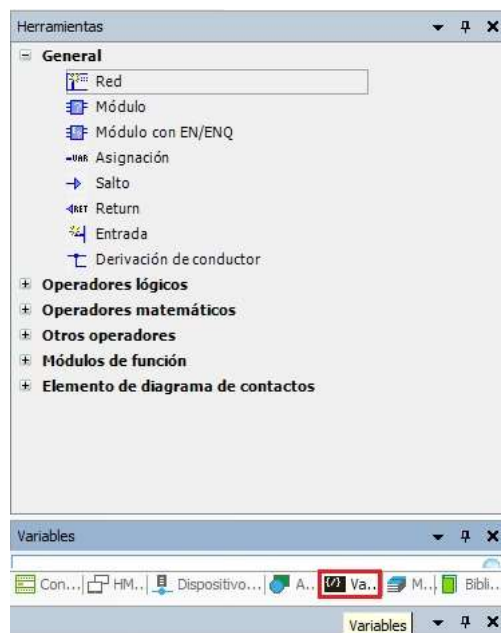


Figura 4. Seleccionar elemento “Variables” en el área de Herramientas.

En la figura 5, se observan las variables que se publicaran en el Vijeo Designer cuando se realice la nueva descarga en línea.

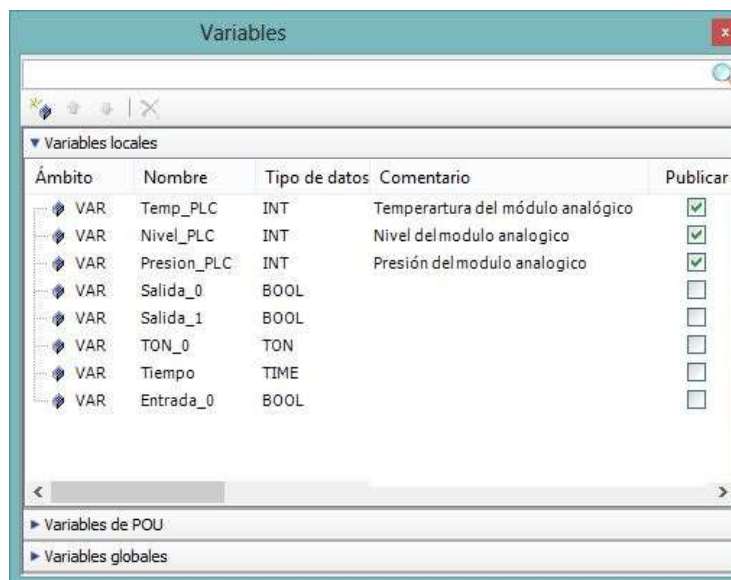


Figura 5. Variables para publicar al HMI

**Paso 3.3-** En el área “Herramientas” seleccionar la pestaña “Dispositivos” e ir al elemento “Configurador de Símbolos” y dar clic para que salga en el “Área de trabajo” como un acceso directo. Luego verificar que las variables que queremos publicar estén marcadas y ejecutar el comando “Crear”, ver figura 6.

Las variables se transmitirán en la próxima descarga o cambio en línea.

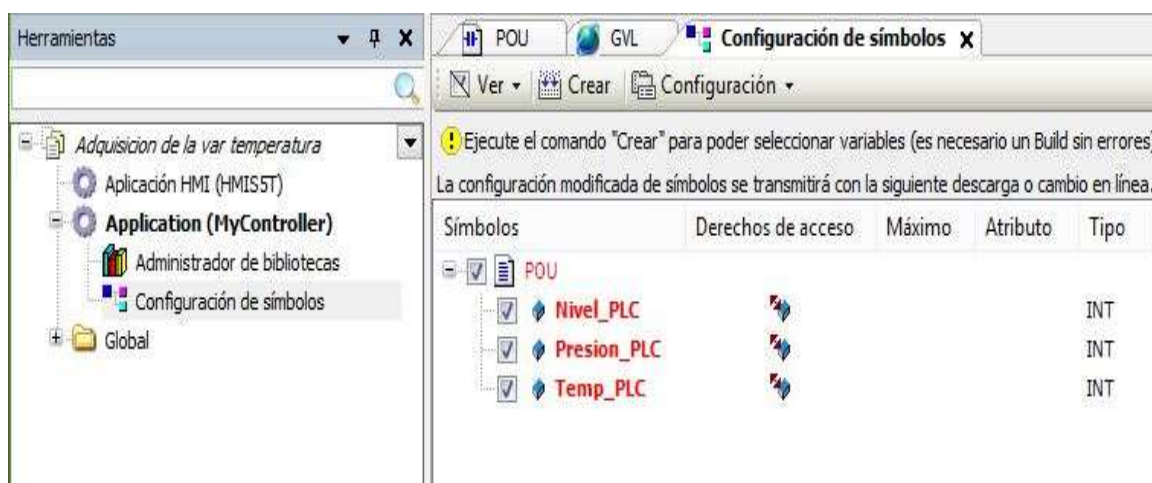


Figura 6. Variables en el Configurador de símbolos

**Paso 4-** Abrir el nuevo proyecto.

Para abrir el nuevo proyecto de Vijeo Designer primero tenemos que abrir el software, que se debe hacer desde el mismo SoMachine. En la barra de herramientas del software SoMachine presione el elemento “?” que se muestran en la figura 7, para activar el “Conmutador de herramientas rápido” que te permite ir al Vijeo Designer, esta opción ya configura el “Administrador de E/S” del Vijeo Designer estableciendo una red llamada “SomachineRed01” la cual enlaza el dispositivo HMI Magelis y el autómata M241, permitiendo su comunicación, ver figura 8.

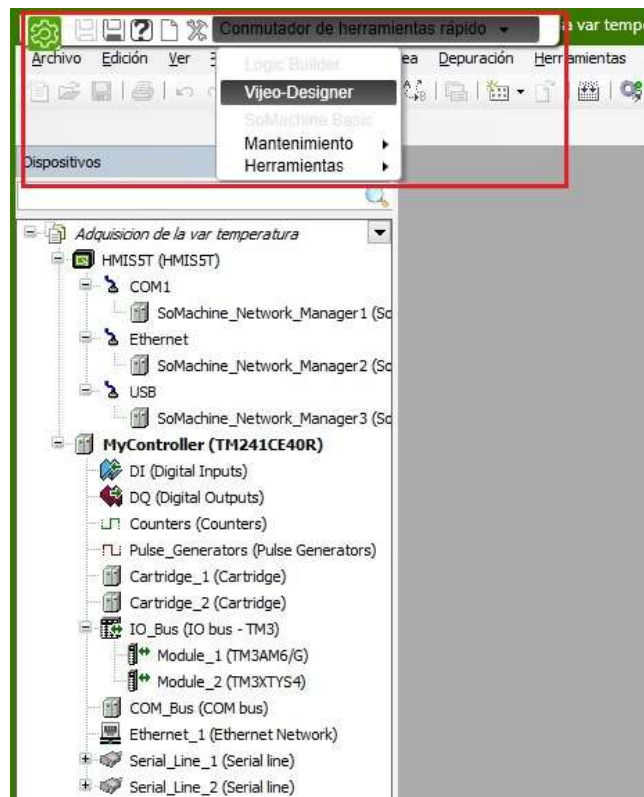


Figura 7. Conmutador de herramientas rápido

En la figura 8 puede observarse el nuevo proyecto, el cual contará con 7 paneles bases (1 panel principal y 2 paneles para cada variable: presión, nivel y temperatura)



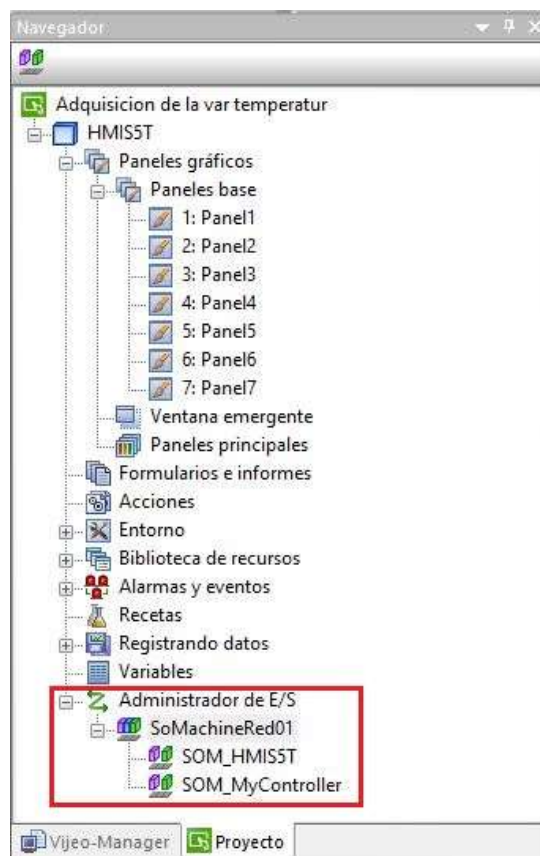


Figura 8. Administrador de E/S del Vijeo Designer

**Paso 5-** Configuración del destino.

Las máquinas de destino representan un interfaz hombre-máquina (HMI) que ejecuta la aplicación de usuario descargada del Vijeo-Designer, como se representa en la figura 9.



Figura 9. Máquina de destino

En la figura 10 se muestra el campo “General” donde se establece la dirección IP del destino, para ello ir al área del Navegador » HMIS5T »General.

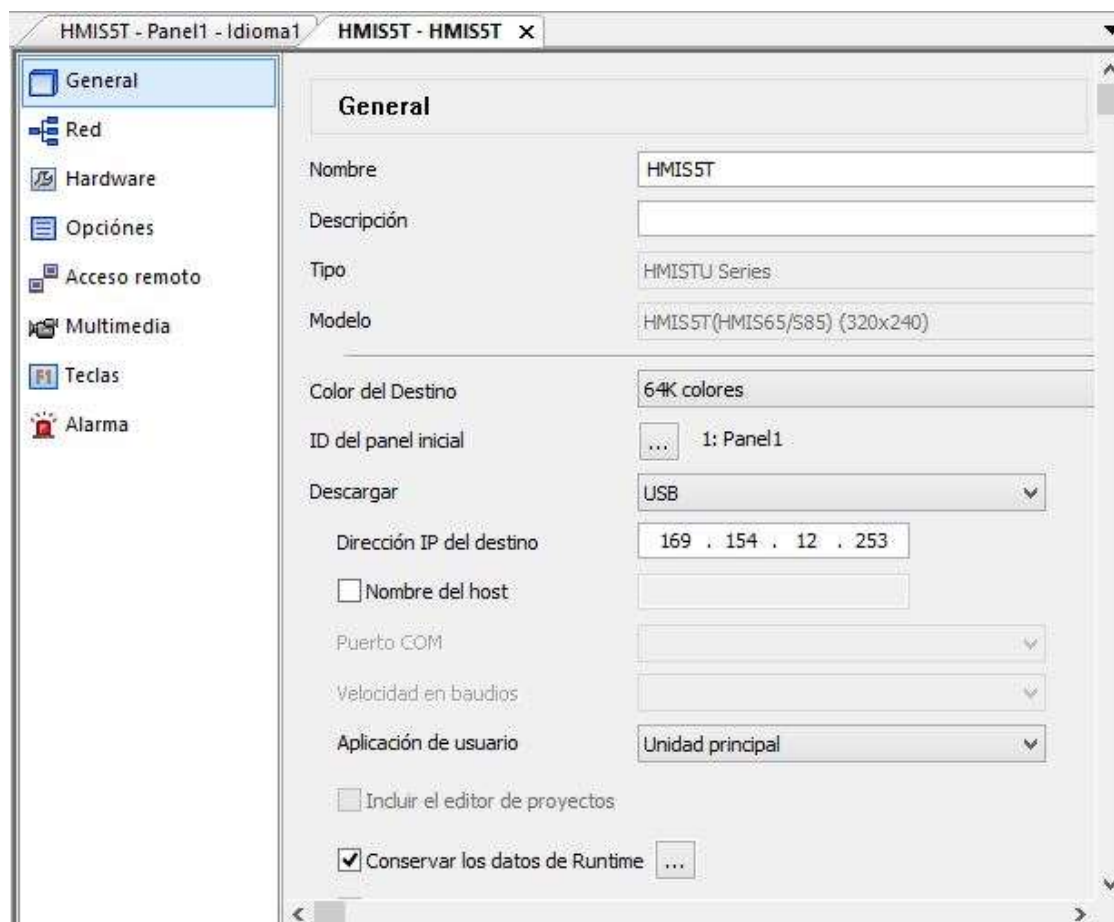


Figura 10. Dirección IP de la máquina de destino

En la figura 11 se muestra el campo “Red” donde se observa la dirección IP establecida en la figura anterior como Dirección IP por defecto para la conexión vía Ethernet.

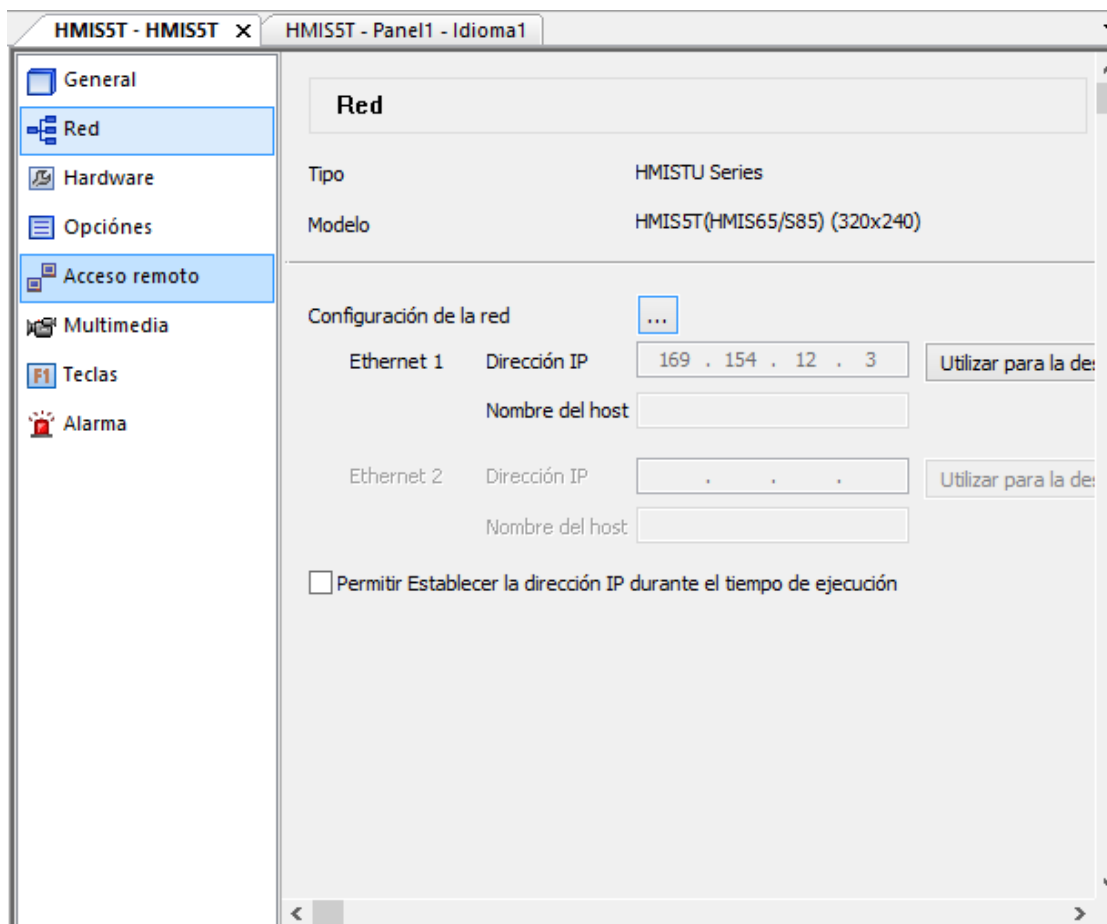


Figura 11. Configuración de red

**Paso 6-** Creación de las variables.

Para crear variables se tiene que hacer doble clic en el icono de 'Variables' que hay en el navegador, esto abrirá en el área de trabajo el 'Editor de variables'.

En el editor de variables hacer clic sobre el icono de 'Crear Variable' que hay en la barra del editor, para abrir la ventana flotante de 'Nueva Variable'. En la ventana de 'Nueva Variable' configuraremos las propiedades de la variable, el tipo de variable, si es interna o externa y si es externa la dirección de memoria del dispositivo configurado, ver figura 12.

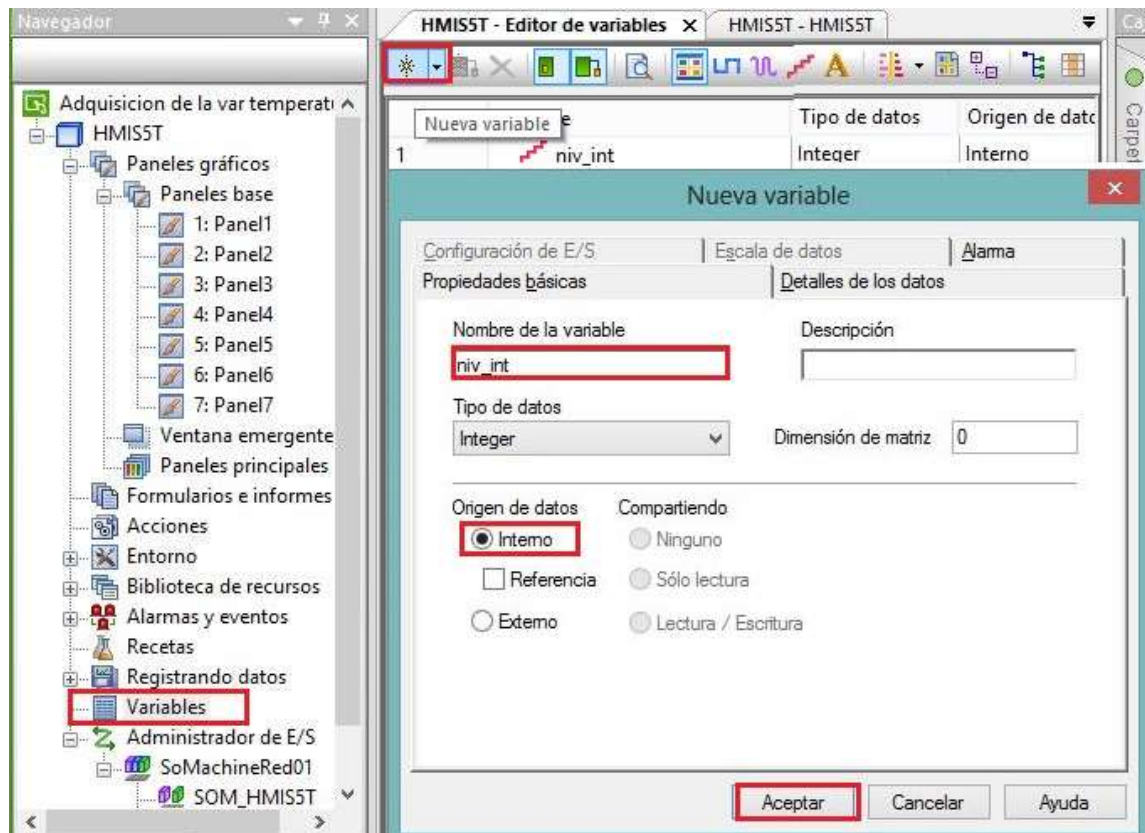


Figura 12. Crear nueva variable

En la figura 13 se observan todas las variables que se crearon en el proyecto:

Internas:

niv\_int: variable que almacena el patrón de nivel introducido por el usuario para comparar con el nivel del proceso.

pre\_int: variable que almacena el patrón de presión introducido por el usuario para comparar con la presión del proceso.

temp\_int: variable que almacena el patrón de temperatura introducido por el usuario para comparar con la temperatura del proceso.

var\_activ\_niv\_XXXX: variable booleana que se activa según sea XXXX(mayor, menor o igual) el patrón de nivel introducido que el nivel actual del proceso.

var\_activ\_pre\_XXXX: variable booleana que se activa según sea XXXX(mayor, menor o igual) el patrón de presión introducido que la presión actual del proceso.

var\_activ\_temp\_XXXX: variable booleana que se activa según sea XXXX(mayor, menor o igual) el patrón de temperatura introducido que la temperatura actual del proceso.

Externas:

Nivel\_PLC: variable analógica, que se encuentra almacenada en el PLC y contiene el valor del nivel del proceso.

Temp\_PLC: variable analógica, que se encuentra almacenada en el PLC y contiene el valor de la temperatura del proceso.

Presion\_PLC: variable analógica, que se encuentra almacenada en el PLC y contiene el valor de la presión del proceso.

	Nombre	Tipo de datos	Origen de datos	Grupo de escan...
1	niv_int	Integer	Interno	
2	pre_int	Integer	Interno	
3	temp_int	Integer	Interno	
4	var_activ_niv_igual	BOOL	Interno	
5	var_activ_niv_mayor	BOOL	Interno	
6	var_activ_niv_menor	BOOL	Interno	
7	var_activ_pre_igual	BOOL	Interno	
8	var_activ_pre_mayor	BOOL	Interno	
9	var_activ_pre_menor	BOOL	Interno	
10	var_activ_temp_igual	BOOL	Interno	
11	var_activ_temp_mayor	BOOL	Interno	
12	var_activ_temp_menor	BOOL	Interno	
13	_SoM			
	MyController			
	Application			
	POU			
	Nivel_PLC	INT	Externo	SOM_MyContr...
	Presion_PLC	INT	Externo	SOM_MyContr...
	Temp_PLC	INT	Externo	SOM_MyContr...

Figura 13. Variables del proyecto

**Paso 7-** Crear el panel base.

Paneles base: Un panel es el lienzo en el cual dibuja objetos y pone objetos en la Caja de herramientas.

En la figura 14 se observa como crear el panel base seleccionando en el “Navegador” el elemento “Panel base” y dar clic derecho, nos sale una ventana donde seleccionaremos “Nuevo panel”.

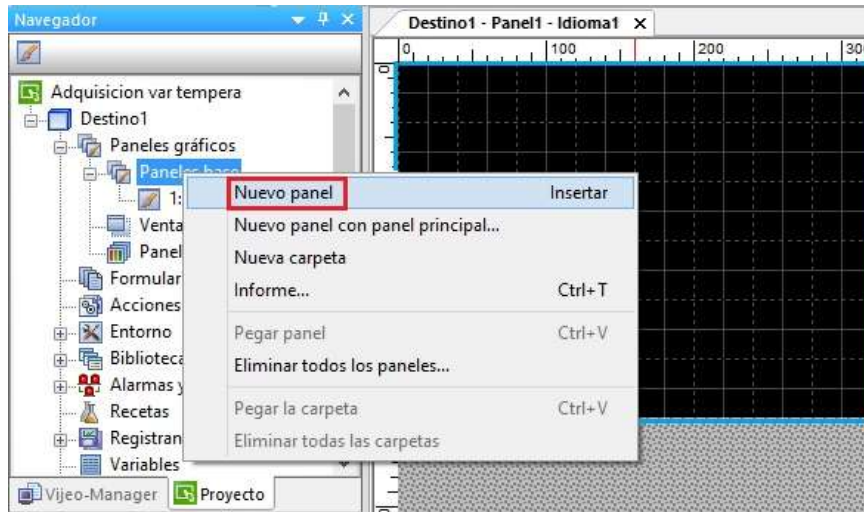


Figura 14. Creación del panel base

En la figura 15 se observa el elemento “Color de fondo”, donde podemos seleccionar el color de nuestro panel base.

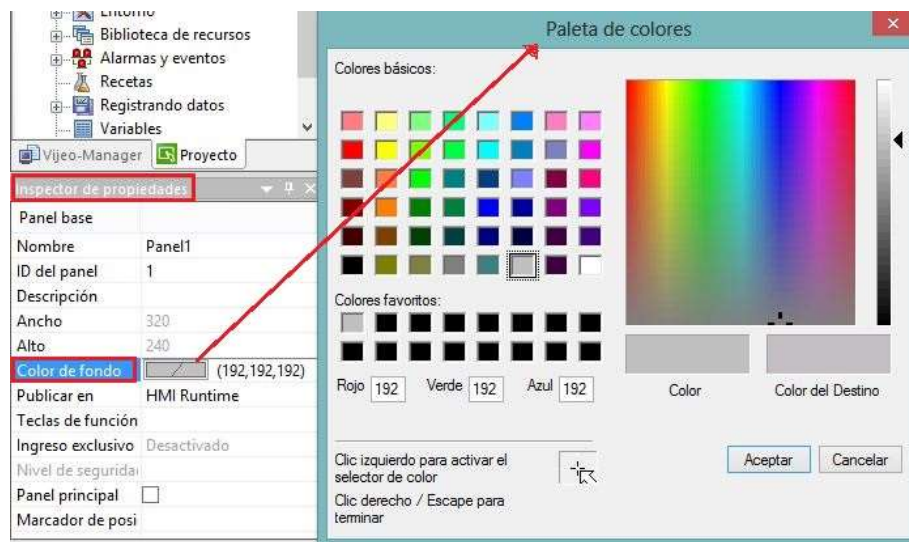


Figura 16. Seleccionar el color de fondo del panel base



**Paso 8-** Colocar una imagen de fondo en el panel base.

Vaya a su PC y copie una imagen y dentro del software Vijeo Designer seleccione el panel base y pegue (Ctrl+V) la imagen, luego ajustarla al panel según se desea, observe la figura 17.

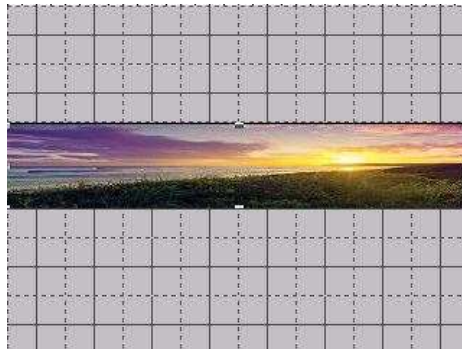


Figura 17. Imagen de fondo en el panel base

**Paso 9-** Escribir textos en el panel base.

En la “Barra de herramientas” se encuentra circulado el elemento representado por una “A”, seleccionando este elemento podemos escribir textos en el panel, para utilizarlos como títulos y etiquetas. En la figuras 18 se observa el tamaño, estilo de fuente y el título con que debe quedar nuestro panel.

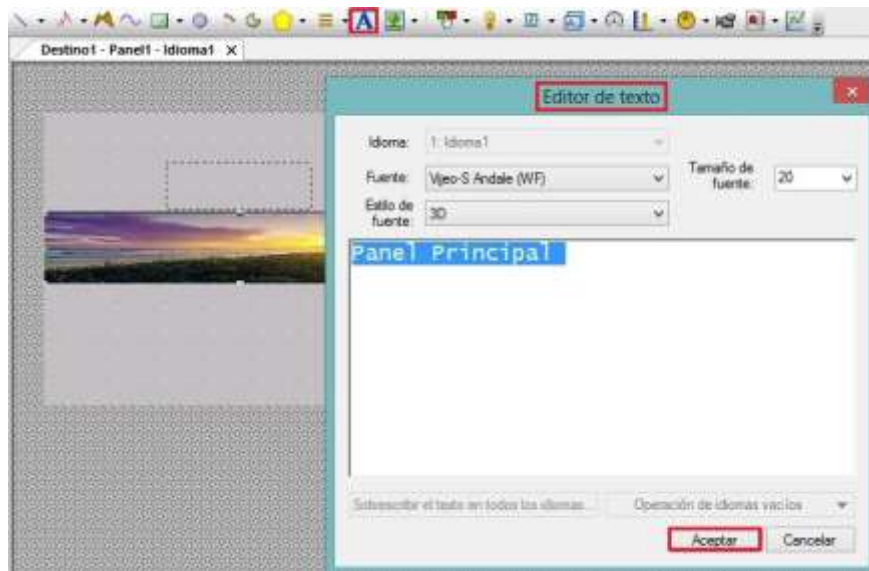


Figura 18. Añadir textos en el panel

En la figura 19 se muestra el resultado de la implementación de los pasos del 1 al paso 9.



Figura 19. Panel principal con el interruptor de cambio de panel

**Paso 10-** Repetir el paso 8 hasta tener los 7 paneles necesarios en el proyecto, que se muestran en la figura 12.

**Paso 11-** Agregar y configurar interruptor de cambio de panel.

En la figura 20 se observa el elemento que hay que seleccionar en la “Barra de herramientas” para añadir un interruptor al panel que permita el cambio hacia otro panel base, también se observa como configurar el interruptor de manera que crealiza la acción “Cambio al panel 2”.

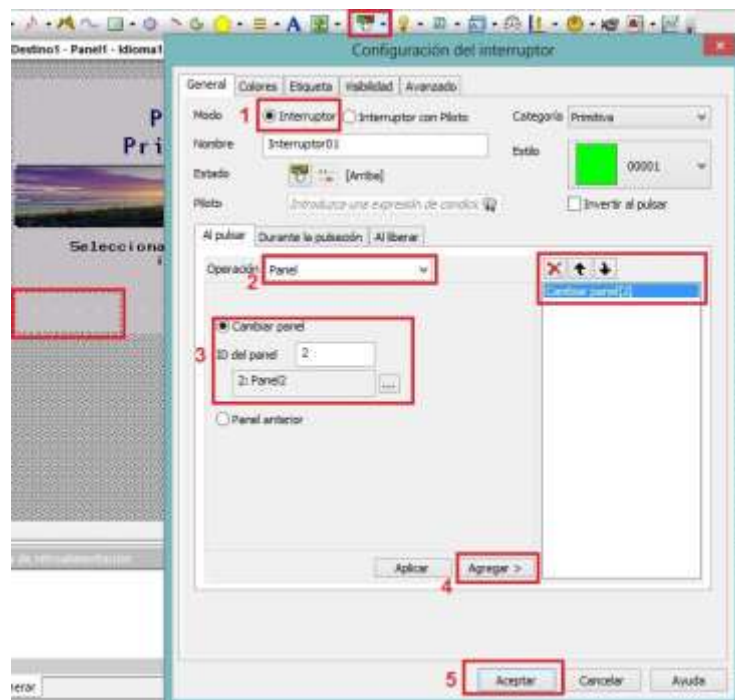


Figura 20. Añadir interruptor para el cambio de panel.



En la figura 21 se observa como configurar la etiqueta del panel base, debe quedar como muestra la figura 19.



Figura 21. Configuración de la etiqueta del interruptor.

Realizar el paso 9 hasta tener los tres interruptores: uno para cada variable de interés.

**2- Crear el panel que se muestra en la figura 22, el cual contenga un Visualizador Numérico que le permita al usuario visualizar la variable temperatura del proceso y regresar al panel principal o ir al panel siguiente.**



Figura 22. Visualización de la variable temperatura.

**Paso 1-** Configurar el visualizador numérico

En la figura 23 y figura 24 se observa el elemento “Visualizador Numérico” y como configurarlo para poder mostrar el valor de la temperatura el cual esta almacenado en la variable Temp\_PLC; por lo que en el campo “Variable” se le da clic al simbolo de los ”dos bombillos” el cual te permite escoger la variable de interés, ver figura 23.

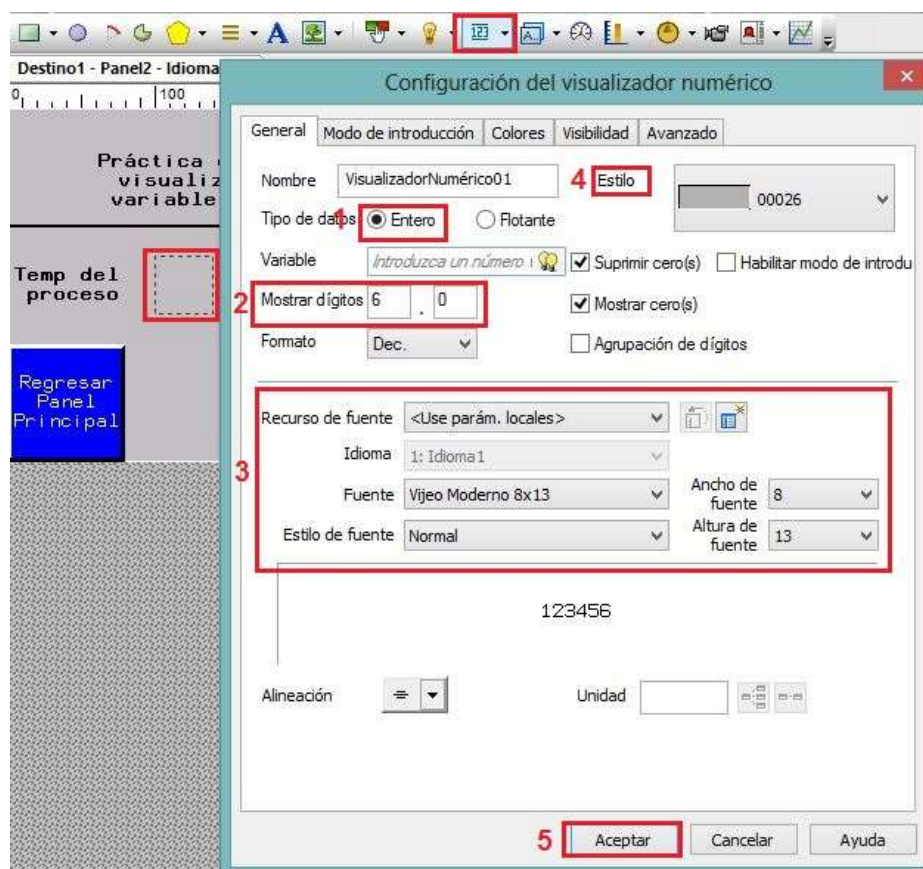


Figura 23. Configuración del Visualizador Numérico

### **Paso 2-** Añadir rectángulo de animación de relleno vertical.

En la figura 24 se muestra el elemento rectángulo y como hay que configurarlo para lograr la animación de relleno vertical de la temperatura. En la figura en el campo indicado por un “2” se tiene que dar clic en el símbolo de “dos bombillos” para seleccionar la variable de interés (Temp\_PLC). En el primer campo “Intervalo de valores” poner el rango de medición del sensor y en el segundo el intervalo de visualización (en %).

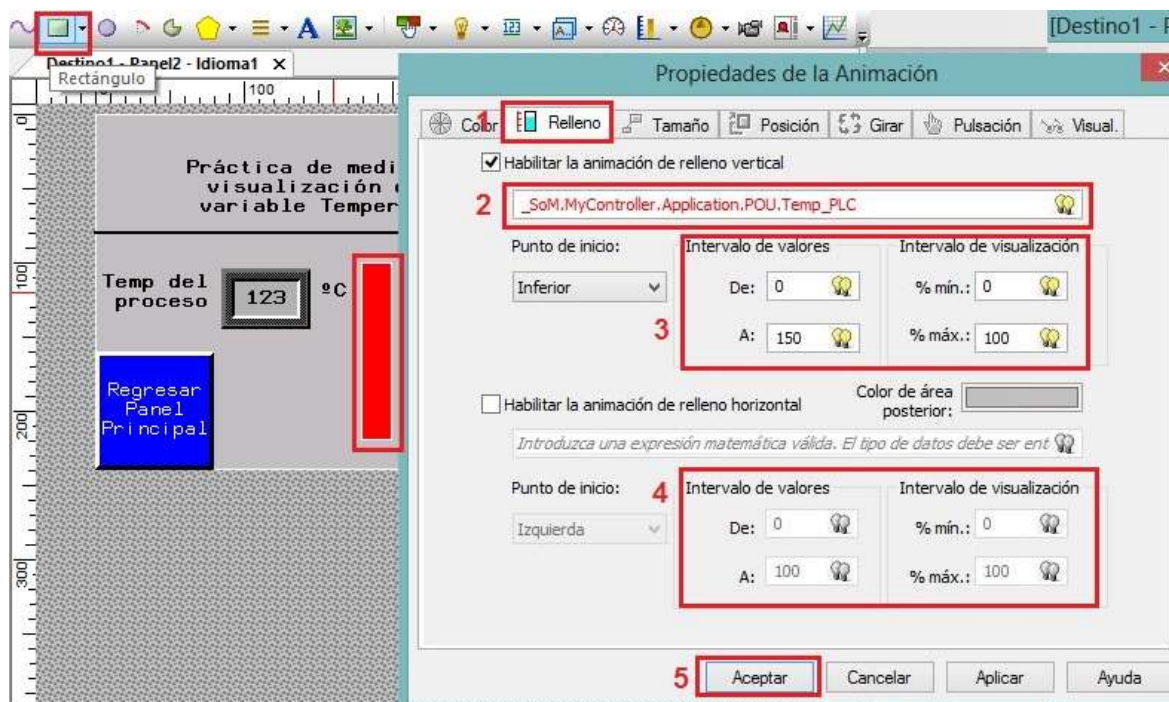


Figura 24. Animación de relleno vetical.

**Paso 3-** Selección de una figura para unirla con el rectángulo de animación vertical.

En la figura 25 se observa como se seleccionó el termómetro ya que es la figura más indicada para unirla con el rectángulo de animación vertical.

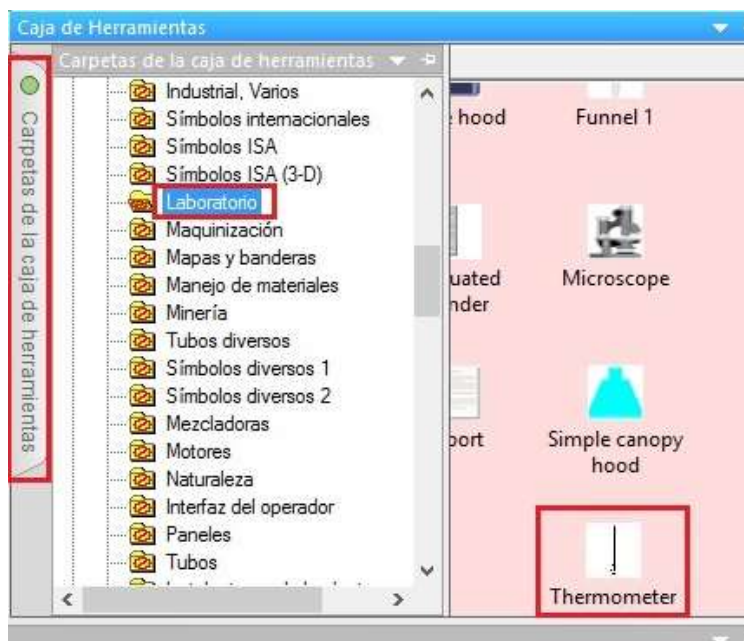


Figura 25. Selección del termómetro

Ya una vez seleccionado el termómetro y configurado su animación vertical, se puede observar el panel base resultante en la figura 22.

**3- Crear el panel que se muestra en la figura 26, el cual contenga un visualizador numérico para mostrar la temperatura actual del proceso y compararla con un valor patrón de temperatura introducido por el usuario mediante un teclado emergente; el panel debe dar la opción de regresar al panel anterior o al principal**



Figura 26. Comparación de las temperaturas

**Paso 1-** Para realizar el visualizador numérico para mostrar la temperatura actual del proceso y el relleno vertical de la temperatura en el termómetro, así como los interruptores de cambio de panel seguir los pasos anteriores.

**Paso 2-** Activar el método de introducción de teclado.

En la figura 27 se muestra como activar en el visulizador numérico el teclado emergente para que el usuario pueda escribir el valor patrón para comparar. Tmbien se puede observar donde colocar el limite inferior y superior que queremos que se introduzca por el teclado emergente, para un valor superior o inferir a estos límites no se introduce y el progrmaa da una alerta de error.



Figura 27. Activar método de introducción por teclado

En la figura 26, puede observarse el panel resultante de los pasos 14 y 15, y en la figura 28, el teclado emergente que le sale al usuario para que introduzca el patrón de comparación.

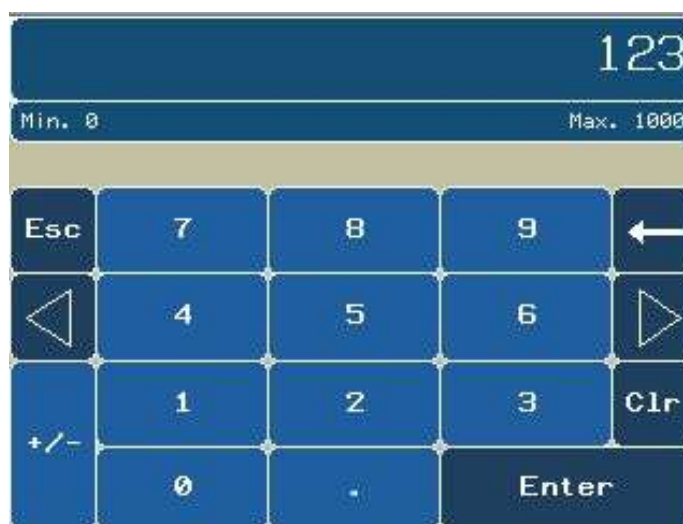


Figura 28. Teclado emergente.

4- Añadir al panel de la técnica operatoria 3, tres luces pilotos y que se active la que indique el resultado de la comparación (mayor, igual o menor) establecida, observe la figura 29.



Figura 29. Resultado de la comparación

**Paso 1-** Crear las luces pilotos.

En la figura 30 se muestra el elemento “Luz piloto” el cual se selecciona y se configura para añadir al panel de la figura 29 las tres luces pilotos para que se active la que se corresponda con el resultado de la comparación entre el valor de la temperatura del proceso y la temperatura patrón introducida por el usuario mediante el teclado emergente. En la figura se puede observar como en el campo indicado por un “2” se escribe la variable booleana a la que responde la activación o no de la luz piloto.

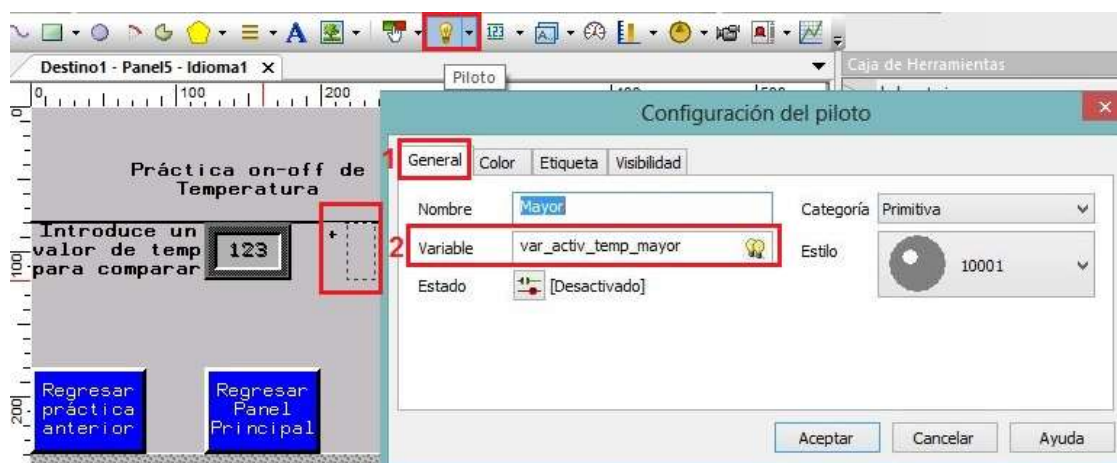


Figura 30. Luces pilotos



En la figura 31 se puede observar como escoger el color que mostrara la luz piloto cuando este activada o desactivada.

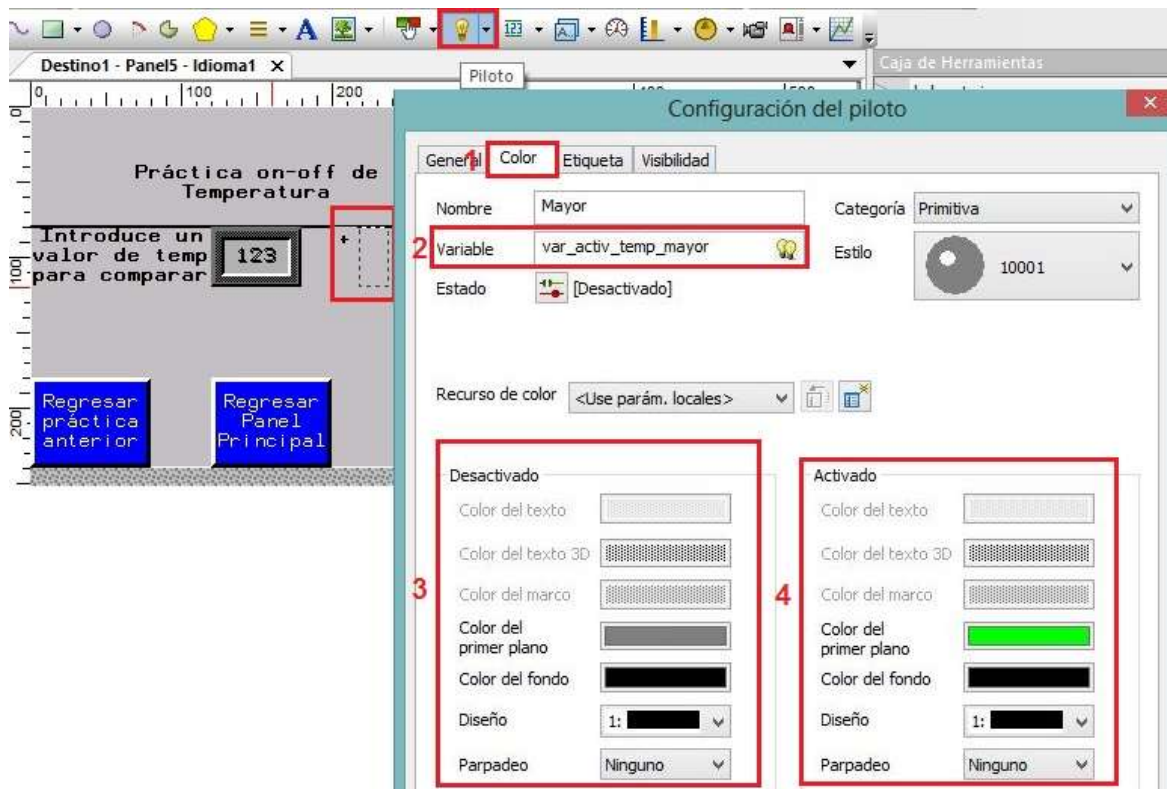


Figura 31. Escoger color de activación-desactivación de la luz piloto.

En la figura 32 se puede observar la luz piloto que se activara cuando el valor de temperatura introducido por el usuario sea mayor que la temperatura del proceso



Figura 32. Luz piloto mayor

Repetir el paso 1 hasta tener las tres luces pilotos, debe quedar como muestra en la figura 29.

**Paso 2-** Crear una acción para darle la funcionalidad a las luces pilotos de activarse-desactivarse.

En la figura 33 se puede observar como seleccionar el elemento “Acciones dentro del área del “Navegador”, esto nos creara una acceso directo en la “Barra de herramientas” en donde vamos a crear una nueva acción y configurarla. Seleccionar como tipo del disparador “Periódica” y dar clic en siguiente. El resto de la configuración de la acción se muestra en la figura 34 y 35.

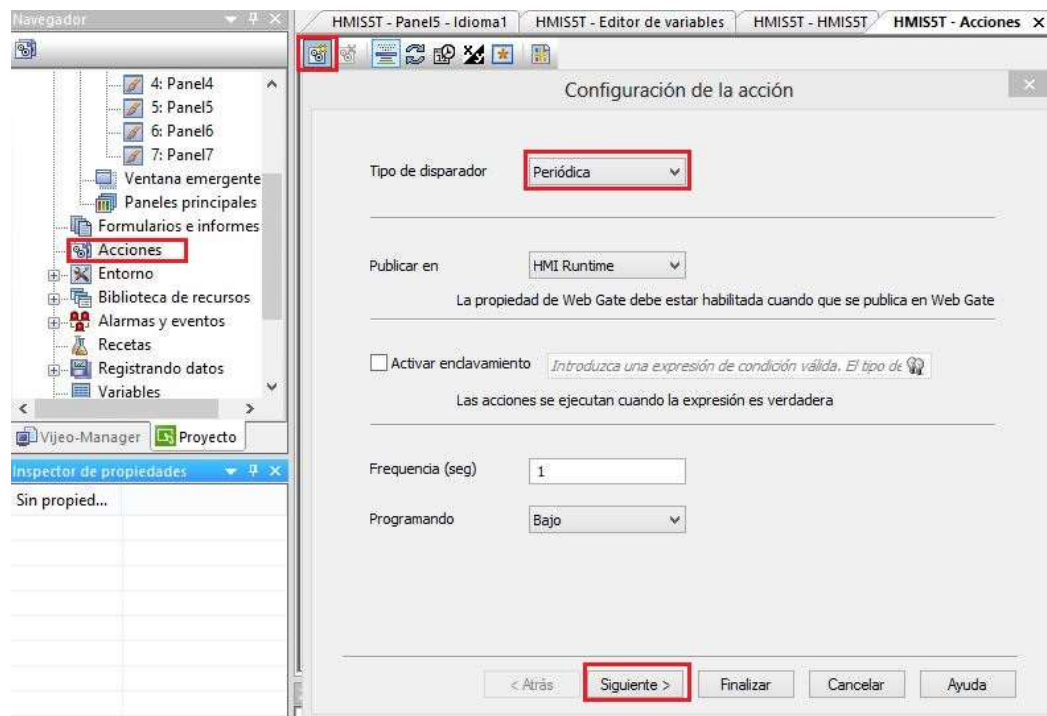


Figura 33. Configurar una acción.

En la figura 34 se puede observar como se escoge la operación “Script” ya que significa que la acción que queremos hacer no está predeterminada por el software, sino que será una acción que vamos a crear nosotros en el lenguaje C++.



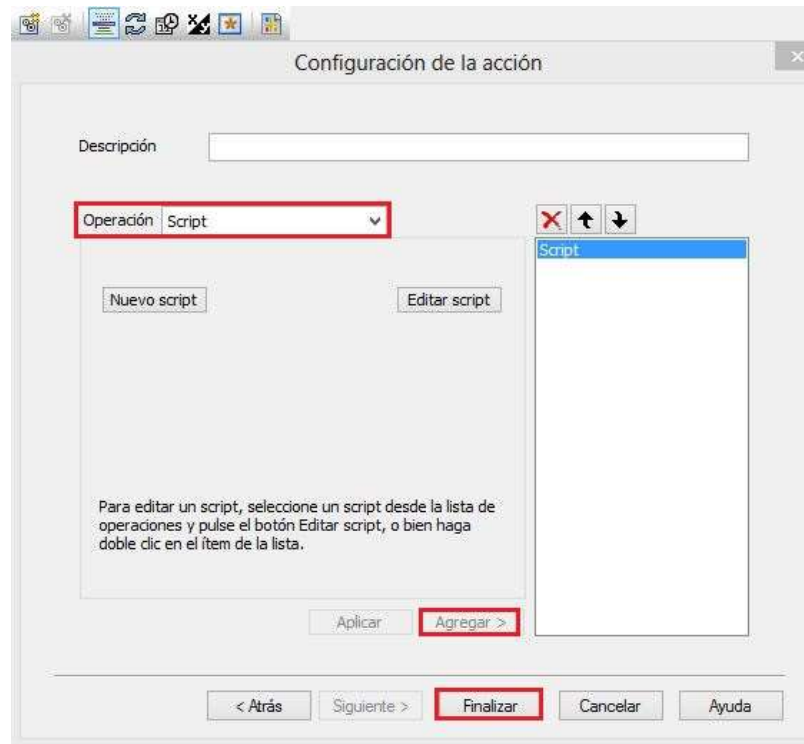


Figura 34. Seleccionar que tipo de acción.

En la figura 35 podemos observar la programación del “Script” básicamente lo que hace es cargar en dos variables enteras la temperatura del proceso y la temperatura patrón introducida por el usuario mediante el teclado emergente, entonces establecer una comparación entre ellas y según sea el resultado activar una variable booleana que es la variable que activa la luz piloto.



Figura 35. Programación en C++ de Script

En la figura 36 se puede observar los tres Script creados, uno para cada luz piloto.

Acciones				
	Disparador	Propiedad	Enclavamiento	Acciones
1	Periódica	Repetir cada 1 seg.		Script
2	Periódica	Repetir cada 1 seg.		Script
3	Periódica	Repetir cada 1 seg.		Script

Figura 36. Scripts creados

Para comprobar la funcionalidad de los Script sobre las luces pilotos, puede observarse la figura 39 donde se muestra como el usuario al introducir por el teclado emergente que se

observa en la figura 38 el valor de 27°C y al ser igual a la temperatura del proceso se activa la luz piloto de igualdad.



Figura 38. Teclado emergente



Figura 39. Comparación entre los valores de temperaturas.

## 5- Realizar la simulación del proyecto y descarga hacia el HMIS5T.

### Paso 1- Simulación del proyecto.

Permite simular la máquina de destino y controlar los diferentes datos que provienen de los dispositivos de E/S.

En la figura 40 se observa que para simular hay que ir al área del “Navegador” luego dar clic derecho y seleccionar “Iniciar simulación de dispositivo” y luego sale la barra de

Compilación que se muestra en la figura 41, al terminar de cargar esta barra entramos en el modo de simulación

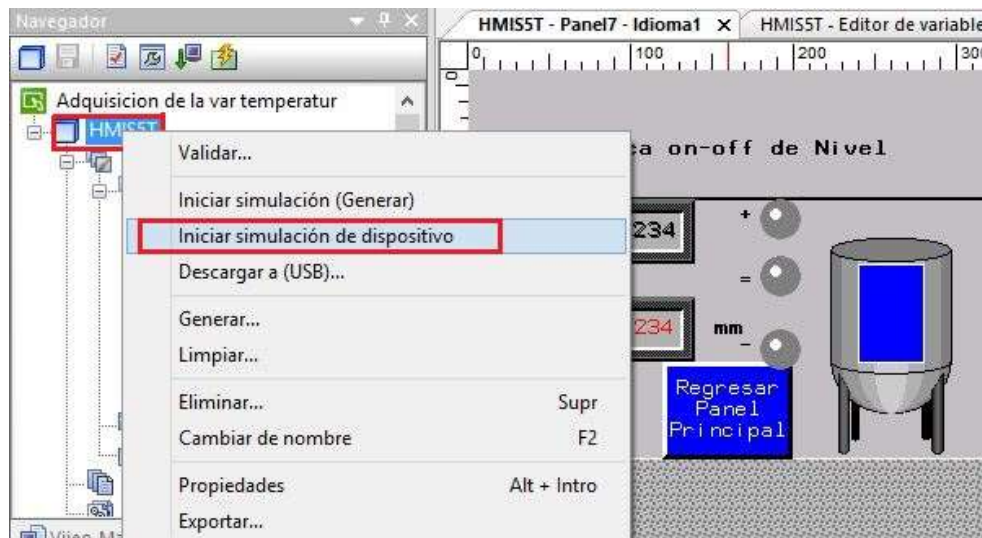


Figura 40. Simulación del proyecto

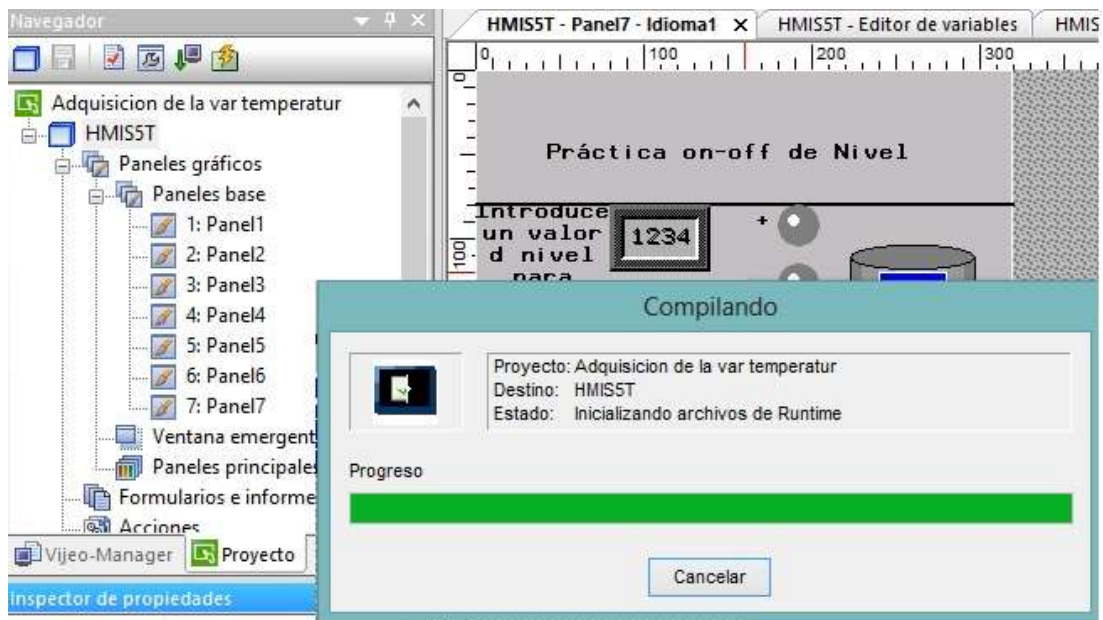


Figura 41. Barra de Compilación

**Paso 2-** Descargar del proyecto a través de Ethernet.

Se tiene que poner la IP de la pantalla HMI. Para ello se ha de seleccionar en destino la opción 'Descargar' seleccionar 'Ethernet'. Una vez seleccionado que la descarga se realizará a través

de Ethernet desplegar la opción ‘Descargar’ y poner la IP que se desea de la máquina de destino, como se observa en la figura 10.

Como se observa en la figura 42, cuando se haya configurado la IP del destino, haciendo clic en el botón derecho, se seleccionará ‘Descarga a (Ethernet “IP de destino”)’ y se realizará la descarga.

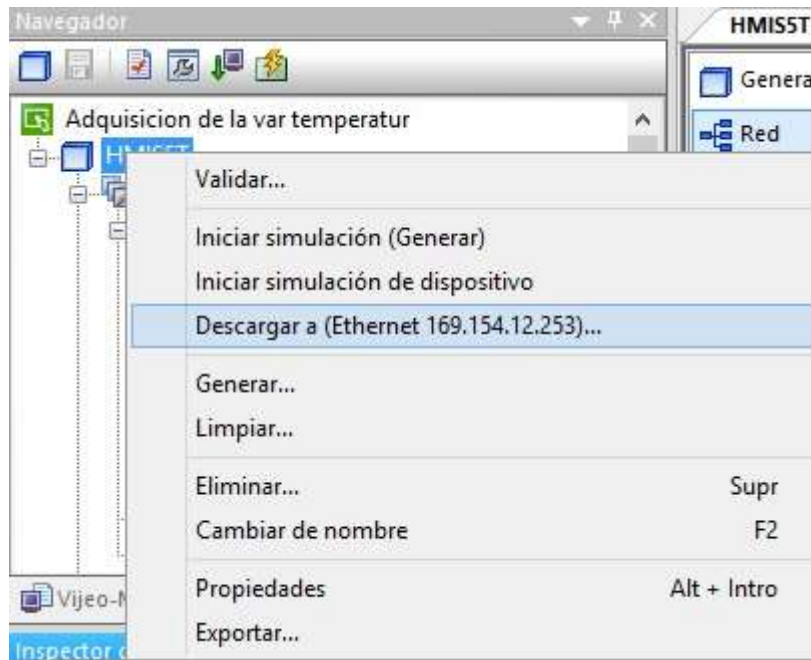


Figura 42. Descargar vía Ethernet

### Conclusiones:

¿Cuáles son las áreas de trabajo del software Vijeo Designer y descríbalas?

¿Qué pasos debe seguir para la correcta configuración de los paneles base Magelis HMISTU tipo HMIS5T que permite la visualización de las variables del autómata?

Realice los ejercicios 1 y 2 de la comprobación de los conocimientos adquiridos en el laboratorio, elabore un informe y súbalo a la carpeta de la asignatura en el Moodle.

### Bibliografía:

1- SoMachine. Guía de programación, abril del 2014. Disponible en:

<https://moodle.uclv.edu.cu>

2- Vijeo Designer Tutorial. enero 2005. Curso de Certificación.

Disponible en: <https://moodle.uclv.edu.cu>

3-Vijeo Designer. Manual de formación. Disponible en: <https://moodle.uclv.edu.cu>

5-Video tutoriales del Vijeo Designer. Disponible en: <https://moodle.uclv.edu.cu>

6-InfoPLC 2017. Manuales y ejemplos programación autómatas, PLC, HMI.

Disponible en:

<http://www.infoplcn.net/descargas/240-schneider-electric/automatas/m241-m251>

### **Comprobación de los conocimientos adquiridos:**

1- Realizar la técnica operatoria 2, 3, 4 y 5 para las variables presión y nivel.

\* como actividad de investigación extra curricular, realice la animación de la estera cuando se cambia de botella. En los videos tutoriales del Vijeo Designer encontrara como animar figuras.