



UNIVERSIDAD  
DE LOS ANDES  
MERIDA VENEZUELA

UNIVERSIDAD DE LOS ANDES  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA ELÉCTRICA

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)  
DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA SCADA PARA  
PRÁCTICAS DEMOSTRATIVAS DEL LABORATORIO DE  
CONTROL UTILIZANDO EL PLC SIEMENS S7-300 Y  
SOFTWARE LIBRE

Br. Mario David Uzcátegui Contreras

Mérida, junio, 2019

Reconocimiento-No comercial- Compartir igual



UNIVERSIDAD  
DE LOS ANDES  
MERIDA VENEZUELA

UNIVERSIDAD DE LOS ANDES  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA ELÉCTRICA

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA SCADA PARA  
PRÁCTICAS DEMOSTRATIVAS DEL LABORATORIO DE  
CONTROL UTILIZANDO EL PLC S7-300 Y SOFTWARE LIBRE**

Trabajo de Grado presentado como requisito parcial para optar al título de Ingeniero  
Electricista

Br. Mario David Uzcátegui Contreras  
Tutor(es): M.Sc. Oscar Enrique Blanco Ortiz

Mérida, junio, 2019

UNIVERSIDAD DE LOS ANDES  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA ELÉCTRICA

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA SCADA PARA  
PRÁCTICAS DEMOSTRATIVAS DEL LABORATORIO DE CONTROL  
UTILIZANDO EL PLC S7-300 Y SOFTWARE LIBRE**

Br. Mario David Uzcátegui Contreras

Trabajo de Grado, presentado en cumplimiento parcial de los requisitos exigidos para optar al título de Ingeniero Electricista, aprobado en nombre de la Universidad de Los Andes por el siguiente Jurado.

www.bdigital.ula.ve

---

Prof. (Ing. Junior Marquina)

---

Prof. (Ing. David Quintero)

---

Prof. (Ing. MSc. Oscar E. Blanco O.)

## **AGRADECIMIENTOS**

A mi mamá María, a mi papá Mario, a mis hermanos, familiares, amigos, a María Virginia, compañeros de estudios, profesores, y a mi tutor Oscar. Gracias a todos por su apoyo, colaboración y motivación.

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

**Mario David Uzcátegui Contreras. Diseño e implementación de un sistema SCADA para prácticas demostrativas del laboratorio de control utilizando el PLC S7-300 y software libre.** Universidad de Los Andes. Tutor(es): Mario David Uzcátegui Contreras. Mayo 2019

## Resumen

Las industrias están invirtiendo cada día más en automatizar sus procesos con el objetivo de cumplir con la demanda global de bienes y servicios. Gracias a la evolución de las tecnologías, se ha logrado desarrollar programas y equipos que se adaptan a las necesidades actuales. Los sistemas SCADA han traído grandes beneficios, entre los que se destacan: producción, confiabilidad, mayor eficiencia, flexibilidad, seguridad y calidad. El software libre se está haciendo una tendencia debido a que es de código abierto, lo que implica poder modificarlo sin mayores restricciones, además es gratis en la mayoría de los casos. Este trabajo de grado propone la realización de un sistema SCADA utilizando software libre y un PLC SIEMENS S7-300. Como primer paso se eligió un programa que cumpliera con los requerimientos de los equipos disponibles en el laboratorio de control de la escuela de ingeniería eléctrica de la Universidad de Los Andes. Después, se diseñaron tres prácticas demostrativas para implementarlas con el programa SCADA. Por último, se acondicionó la maqueta de semáforos y se construyó otra para la práctica de llenado de botellas.

**Descriptor:** PLC, SCADA.

## ÍNDICE GENERAL

APROBACIÓN.....	ii
AGRADECIMIENTOS.....	iv
RESUMEN.....	v
INTRODUCCIÓN.....	1
<b>Capítulos</b>	<b>pp.</b>
<b>1. PLANTEAMIENTO DEL PROBLEMA.....</b>	<b>2</b>
1.1 El problema.....	2
1.2 Justificación.....	3
1.3 Objetivos.....	3
1.3.1 Objetivos generales.....	3
1.3.2 Objetivos específicos.....	3
1.4 Metodología.....	4
1.5 Alcance.....	4
1.6 Limitaciones.....	4
<b>2. MARCO TEÓRICO.....</b>	<b>5</b>
2.1 Antecedentes.....	5
2.2 Marco teórico.....	6
2.2.1 Automatización industrial.....	6
2.2.2 Definición de PLC.....	7
2.2.3 Ventajas del PLC.....	8
2.2.4 Estructura de un PLC.....	9
2.2.5 Protocolo de comunicación.....	10
2.2.6 Multi point interfaz.....	12
2.2.7 Lenguaje de programación.....	12
2.2.8 Diagrama en escalera.....	14
2.2.9 Componentes más comunes del diagrama en escalera.....	15
2.2.9.1 Temporizadores.....	15
2.2.9.2 Contadores.....	16
2.2.9.3 Comparadores.....	16
2.2.10 SCADA.....	16
2.2.11 Human machine interfaz.....	19
2.2.12 Software libre.....	19
2.2.13 Servidor OPC.....	21
<b>3. CONFIGURACION DEL SISTEMA SCADA.....</b>	<b>24</b>

3.1	Selección del PLC a utilizar en el sistema SCADA.....	24
3.2	Selección del programa SCADA.....	25
3.3	Configuración del servidor OPC.....	26
3.4	Descripción de IndigoSCADA.....	31
3.5	Configuración de IndigoSCADA.....	33
	3.5.1 IndigoSCADA como cliente OPC.....	33
	3.5.2 Creación de puntos SCADA.....	35
3.6	Procedimiento para crear la HMI.....	42
<b>4.</b>	<b>PRÁCTICAS A IMPLEMENTAR CON EL SISTEMA SCADA.....</b>	<b>44</b>
4.1	Práctica 1: Semáforo de 4 intercepciones.....	44
4.2	Práctica 2: Sistema de llenado automático de tanques.....	49
4.3	Práctica 3: Sistema de llenado y tapado automático de botellas.....	52
<b>5.</b>	<b>RESULTADOS.....</b>	<b>55</b>
5.1	Práctica 1.....	55
5.2	Práctica 2.....	59
5.3	Práctica 3.....	65
	CONCLUSIONES.....	69
	RECOMENDACIONES.....	70
	REFERENCIAS.....	71
	ANEXOS.....	74

## ÍNDICE DE FIGURAS

<b>Figura</b>	<b>pp.</b>
Automatización.....	7
Representación escalera.....	14
Representación escalera.....	15
Temporizador con retardo a la conexión.....	15
Contador UP.....	16
Comparador de igualdad entre enteros.....	17
Representación de un sistema SCADA.....	17
Hardware en un sistema SCADA.....	19
Ejemplo de HMI.....	20
Representación OPC.....	23
PLC Siemens S7-300.....	24
Adaptador MPI/USB.....	25
Programa de prueba.....	26
Elección de protocolo.....	26
Configuración de protocolo.....	27
Configuración de protocolo.....	27
Conexión USB.....	28
Configuración de nueva variable.....	28
Configuración de variable.....	29
Comprobación de valor de variable.....	29
Transfiriendo a servidor OPC.....	30
Transfiriendo a servidor OPC.....	30
Transfiriendo a servidor IBH.....	31
Símbolo del sistema.....	33
Configuración de protocolo.....	34
Importando base de datos.....	34
Creando nuevo archivo de base de datos.....	35
Creando nuevo archivo de base de datos.....	35
Importación completada.....	36
Iniciando IndigoSCADA.....	36
Iniciando IndigoSCADA.....	36
Configurando como cliente OPC, paso 1.....	37
Configurando como cliente OPC, paso 2.....	37
Configurando como cliente OPC, paso 3.....	38
Configurando como cliente OPC, paso 4.....	38



Configurando como cliente OPC, paso 5.....	39
Configurando como cliente OPC, paso 6.....	39
OPCPoints sin comunicación.....	40
Licencias del servidor IBH OPC.....	40
Licencia Demo del servidor IBH OPC.....	41
Estado de las variables en OK.....	41
Creación de HMI.....	42
Creación de HMI.....	43
Programa de prueba para envío de comandos.....	43
Maqueta usada en la práctica de semáforos .....	44
Diseño preliminar de la práctica de semáforos en IndigoSCADA.....	45
Diagrama de flujo semáforo .....	47
Diseño preliminar de llenado de tanques en IndigoSCADA.....	49
Diagrama de flujo de llenado de tanques .....	50
Diseño preliminar de llenado de botellas en IndigoSCADA .....	52
Diagrama de flujo llenado de botellas.....	53
Semáforos desactivados.....	55
Envío de Comandos.....	56
Semáforos en Rojo y semáforo A con luz de cruce.....	56
Ambos semáforos en verde.....	57
Semáforos en Amarillo.....	57
Gráfica en tiempo real.....	58
Base de datos históricos.....	58
Tanques vacíos y gandola desconectada.....	59
Llenando tanque principal.....	60
Tanques 1 y 2 llenos, tanque principal vacío hasta la mitad.....	60
Vaciando tanque 1.....	61
Llenado tanque 2.....	61
Gráfica en tiempo real del nivel del tanque principal.....	62
Creando reporte para una de las variables.....	62
Carpeta de archivos con los reportes.....	63
Reporte con tres variables.....	63
Tabla de valores de la variable OPCPoint20.....	64
Tanque vacío.....	65
Proceso de llenado y tapado detenido.....	65
Llenado tanque.....	66
Tanque lleno.....	66
Llenando y tapando.....	67
Motor de cinta encendido.....	67
Botellas en la cinta.....	68

## INDICE DE TABLAS

<b>Tabla</b>		<b>pp.</b>
1	Descripción de entradas del PLC para semáforos.....	48
2	Descripción de salidas del PLC para semáforos.....	48
3	Descripción de entradas del PLC para llenado de tanques.....	51
4	Descripción de salidas del PLC para llenado de tanques.....	51
5	Descripción de entradas del PLC para llenado de botellas .....	54
6	Descripción de salidas del PLC para llenado de botellas .....	54

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

# INTRODUCCIÓN

Los sistemas de supervisión, control y adquisición de datos (SCADA) ha hecho mucho más eficiente los procesos de manufactura en las industrias, incrementando la producción, reduciendo costes y disminuyendo los accidentes laborales. Con el objetivo de aprovechar los controladores lógicos programables (PLCs) disponibles en el laboratorio de control de la escuela de ingeniería eléctrica de la Universidad de Los Andes, se decidió implementar un sistema SCADA de prueba. Se utilizó un PLC Siemens S7-300 conectado a una computadora a través de un adaptador MPI/USB. El software libre está marcando tendencia debido a que permite al usuario ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Por lo tanto, se utilizó un programa con estas características para implementar el sistema SCADA.

El presente trabajo está dividido en 5 capítulos. En el primero se analiza el problema a resolver y se justifica las razones para la realización del mismo. Se mencionan antecedentes donde se implementan sistemas SCADA con software propietario y por último se establecen los objetivos. En el capítulo 2 se dan definiciones de los conceptos utilizados. El capítulo 3 describe los equipos disponibles para implementar sistema SCADA, también los programas a utilizar y cómo configurarlos. Para el capítulo 4 se desarrollaron los diagramas de flujo de cada práctica a implementar. En el último capítulo 5 se muestran capturas de pantalla de las prácticas en funcionamiento en cada uno sus casos. Como anexo se da un manual de como cargar el programa al PLC y el SCADA para cada una de las prácticas.

# CAPÍTULO 1

## PLANTEAMIENTO DEL PROBLEMA

El presente capítulo describe las bases necesarias para ejecutar este proyecto de grado, estableciendo las distintas razones que permitan crear las condiciones del proyecto.

### 1.1 EL PROBLEMA

El desarrollo de las tecnologías ha permitido implementar nuevos equipos en la producción de bienes y servicios. Las compañías que fabrican estos equipos también ofrecen programas para utilizarlos en sistemas SCADA. Este tipo de *software* requiere del pago de una licencia de alto costo, por lo que algunas pequeñas y medianas industrias no disponen del presupuesto necesario para utilizarlo. El alto costo de la licencia no es el único inconveniente, también se imponen restricciones al uso de este tipo de paquetes computacionales. No es posible ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Es por ello que se han constituido desde hace algunos años, personas, comunidades y/o empresas para desarrollar programas SCADA de uso libre.

El Laboratorio de control de la Escuela de Ingeniería Eléctrica de la Universidad de Los Andes dispone de PLCs los cuales tienen un gran potencial didáctico y académico. La materia del laboratorio de control se ve en la necesidad de incorporar nuevas tecnologías para el desarrollo de la cátedra, lo que les brindará a los estudiantes mayor cantidad de conocimientos que son de suma importancia para el ámbito profesional.

## 1.2 JUSTIFICACIÓN

Las empresas invierten cada día más en tecnologías en busca de mayor eficiencia, calidad, producción en menor tiempo, mayores ganancias y menores costos. Es por ello que se han desarrollado nuevos dispositivos de automatización con mayores capacidades que permiten interconectarse en una red, incrementando las prestaciones de todo el sistema. Por lo que se hace necesario diseñar programas que permitan *supervisar, controlar, adquirir datos* de planta en tiempo real. Los sistemas SCADA hacen esto posible por lo que resulta imprescindible adaptar las materias en el área de automatización de la Escuela de Ingeniería Eléctrica a las nuevas tendencias. Implementar un sistema SCADA de prueba para el Laboratorio de Control servirá como una herramienta didáctica importante para la captación de conocimientos de los alumnos de esta materia. El uso de programa SCADA de *software libre* se ha incrementado debido a dos razones fundamentales: Uso gratuito (en la mayoría de los casos) y de código abierto. Al implementar el SCADA con un programa de este tipo permitiría explorar y descubrir en detalle sus ventajas y posibilidades.

## 1.3 OBJETIVOS

### 1.3.1 Objetivo General

Diseñar e implementar un sistema SCADA para prácticas demostrativas del Laboratorio de Control, utilizando el PLC Siemens S7-300 y software libre.

### 1.3.2 Objetivos Específicos

- Estudiar el funcionamiento del PLC Siemens S7-300.
- Analizar el funcionamiento de los sistemas SCADA.
- Seleccionar un software libre para la implementación del sistema SCADA.
- Realizar pruebas de comunicación entre el PLC y el SCADA.
- Diseñar tres (3) prácticas demostrativas.
- Implementar el SCADA para cada una de las prácticas.

## 1.4 METODOLOGÍA

Según el propósito de la investigación, existen diferentes estudios que la clasifican. Según [1], definen los tipos de investigación en básica y aplicada. Este trabajo se puede ubicar en el tipo de investigación aplicada, ya que sirve de alternativa al uso de SCADAs propietarios en la industria.

Barrios [2] los clasifica en: a. Investigación de Campo; b. Investigación Documental; c. Proyectos Factibles y d. Proyectos Especiales. La presente investigación también se puede establecer en la clasificación de Investigación de Campo experimental ya que consiste en la búsqueda de un software que cumpla con las especificaciones de los equipos disponibles. [2] establece que “se entiende por investigación de Campo, el análisis sistemático de problemas en la realidad, con el propósito bien sea de describirlos, interpretarlos, entender su naturaleza y factores constituyentes, explicar sus causas y efectos, o predecir su ocurrencia”.

## 1.5 ALCANCE

Se obtiene un sistema SCADA con un PLC S7-300 conectado a una computadora con sistema operativo Windows XP. Los datos del PLC se obtienen por medio de un servidor OPC y se utiliza un software SCADA como cliente OPC. Se desarrollaron 3 prácticas demostrativas. Para simular un proceso real se conectan maquetas al PLC para 2 de las 3 prácticas.

## 1.6 LIMITACIONES

Debido a los requisitos del programa SCADA al sistema operativo, solo se puede conectar a las computadoras del laboratorio con Windows XP de 32 bits. Se dispone de una licencia DEMO para el uso del servidor OPC, es por ello que el sistema SCADA opera solo treinta minutos seguidos por lo que se necesita reiniciar el programa después de transcurrido este tiempo.

# CAPÍTULO 2

## MARCO TEÓRICO

### 2.1 ANTECEDENTES

Angarita, en su tesis de grado [3] “Programación del PLC modelo S7-300 con STEP7 LITE” propone, una guía de usuario del PLC siemens S7-300 utilizando el software STEP 7 LITE como herramienta de programación. El autor describe su trabajo como “una herramienta pedagógica en cuya concepción, basada en el espíritu práctico y la sencillez, con el fin de que todos aquellos que se inicien al fascinante mundo de los automatismos adquieran unos conocimientos elementales, aprendan a interpretar los esquemas básicos y conozcan los procedimientos para montar equipos respetando las normas internacionales y europeas”. En definitiva, el autor contribuye como guía a la instalación y puesta en marcha del software. En él se explican los procedimientos necesarios para crear y estructurar los programas de usuario y se incluyen descripciones de los elementos del lenguaje de programación.

Calderón [4], en su trabajo de grado titulado “Control y monitoreo SCADA de un proceso experimental, utilizando PLC Siemens S7-300 y software LABVIEW” realiza un sistema SCADA para un proceso experimental de una planta hidráulica. Para el desarrollo del trabajo utiliza el PLC siemens S7-300 programado con el programa STEP 7 LITE. Para el sistema SCADA hace uso del software LABVIEW, el cual requiere una licencia. Para la comunicación entre el PLC y LABVIEW, el autor aplicó el protocolo de comunicación OLE for Process Control (OPC) el cual en principio es exclusivo del sistema operativo WINDOWS.

Hernández [5] en su trabajo de grado que lleva por título: “Desarrollo de un sistema SCADA para la medición de voltajes con sistemas embebidos para el Laboratorio de Mecatrónica de la Facultad de Mecánica” utiliza una tarjeta de adquisición de datos de modelo NI cRIO-9074 con su respectivo software el cual se utilizó para establecer la comunicación con la computadora. El objetivo del autor es obtener datos de las mediciones de voltajes con Sistemas Embebidos para el Laboratorio de Mecatrónica de la Facultad de Mecánica usando la tarjeta de adquisición de datos y el software LABVIEW para la comunicación entre dispositivos y la realización de la interfaz hombre máquina. El autor concluye que el sistema “permite monitorear, controlar y registrar de una manera rápida, en tiempo real e histórico los parámetros principales de los ejemplos de aplicación, con lo que se puede automatizar procesos industriales aplicando nuevas tecnologías”.

Malavé, en su tesis [6] “Diseño de un sistema de supervisión, adquisición y control para las estaciones de radares meteorológicos del Instituto Nacional de Meteorología e Hidrología (INAMEH)” desarrolla un sistema SCADA para realizar un monitoreo de los periféricos externos que forman parte de las estaciones de radares meteorológicos del INAMEH. El software que usa el autor para el SCADA, programar el PLC y la HMI es el WinCC Advanced V11. La comunicación entre el PLC y el SCADA es a través del protocolo TCP/IP. “La mencionada interfaz está compuesta por una visualización gráfica de todas las variables que participan en el proceso, contando con un historiador de eventos donde se observan el estado de las variables en un determinado momento, gráficas del nivel y altura de combustible dentro de los tanques y alarmas analógicas; como indicador al operador sobre los diferentes eventos que están ocurriendo en el proceso”. Como conclusión, el autor destaca que el diseño del sistema SCADA se realizó satisfactoriamente cumpliendo con todos los requisitos y sirviendo como base para futuras expansiones en las estaciones de radares meteorológicos.

## **2.2 MARCO TEÓRICO**

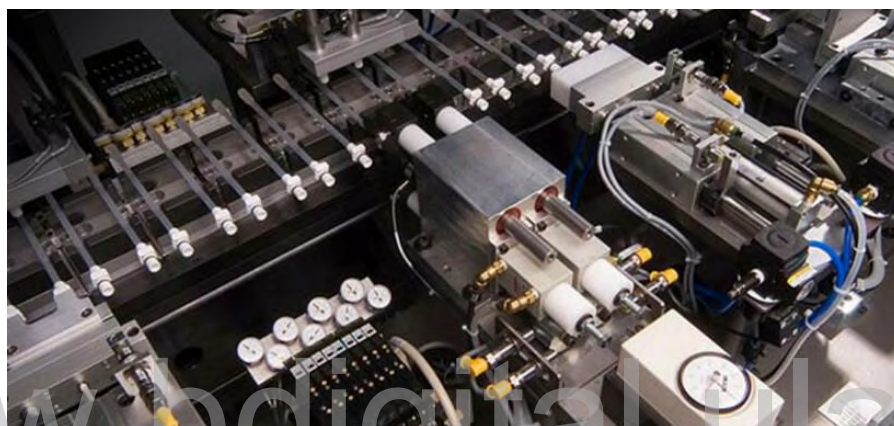
### **2.2.1 Automatización industrial**

La automatización industrial es el uso de equipos eléctricos, electrónicos, electromecánicos (hardware) y software para controlar maquinarias que a su vez controlan procesos industriales.



Es la rama de la ingeniería que abarca áreas como los sistemas de control, instrumentación industrial; que incluye los sensores, transmisores de campo, sistemas de transmisión, recolección de datos y las aplicaciones de software.

La automatización puede definirse también como la utilización de tecnologías y sistemas de control automático que resultan en la operación y control de procesos industriales en forma automática, sin intervención humana significativa, logrando un rendimiento superior al control manual [7]. Estos dispositivos de automación incluyen a los PLCs.



**Figura 1. Automatización Fuente: EXSOL.**

### **2.2.2 Definición de PLC**

Por sus siglas en inglés, Programmable Logic Controller (Controlador Lógico Programable), es un dispositivo usado para controlar procesos. El control se realiza sobre la base de una lógica definida a través de un programa.

Se define como un computador especializado utilizado para controlar máquinas y procesos [3]. Utiliza una memoria programable para almacenar instrucciones y funciones específicas que incluyen control de encendido y apagado, temporizadores, conteo, secuenciador, manipulación aritmética y de información.

El PLC es un equipo electrónico que activa salidas digitales (ceros y unos) y/o analógicas en función de las entradas que reciba, que pueden ser analógicas o digitales. Funciona como una interconexión de compuertas lógicas. Generalmente las entradas del PLC se activan

con sensores e interruptores. En la industria se utilizan generalmente sensores de nivel, temperatura, flujo y presión, interruptores de carrera, interruptores manuales. Las salidas del PLC generalmente se conectan a bobinas de contactores que a su vez controlan motores, calefactores, luminarias etc.

Surgió en los años 60 gracias a la aparición y posterior desarrollo del microcontrolador. Sustituyeron los tableros de relés que son de gran tamaño y consumen mucha potencia. A medida que se aumentó la complejidad de los mismos, los tableros de relés se volvieron más grandes y costosos. Solo se podía utilizar una única lógica o trabajo específico, por lo que algún cambio de proceso implicaba cambio en el cableado o uso de más relés. El PLC se volvió una mejor opción ya que es compacto, ocupa mucho menos espacio que un tablero de relés, es flexible y fácil de programar y no hace falta volver a cablear.

### 2.2.3 Ventajas del PLC

- Flexible: Posibilidad de reemplazar la lógica cableada de un tablero o de un circuito impreso de un sistema electrónico, mediante un programa que corre en un PLC.
- Tiempos de respuesta más rápidos, ahorro de tiempo de trabajo, tanto en las conexiones a realizar, como en la puesta en marcha y en el ajuste del sistema.
- Menos cableados.

El programa reemplaza mucho del cableado externo que se requeriría para controlar un proceso lo cual permite:

- Reducción de Espacio.
- Estandarización.
- Diseño modular: Fácil de reparar y expandir.
- Manejo de sistemas mucho más complicados.
- Conjuntos de instrucciones sofisticadas disponibles.
- Permite realizar diagnósticos (Fácil de reparar).

Ya que toda la lógica es contenida en la memoria del PLC, no existe la posibilidad de cometer un error en la lógica de cableado, permitiendo así:

- Incremento de la confiabilidad: Una vez que el programa ha sido escrito y probado puede ser descargado a otros PLCs.
- Mayor flexibilidad: Los fabricantes de equipos originales pueden proveer actualizaciones de sistema a un proceso, simplemente enviando un nuevo programa.
- Es más fácil el crear y cambiar un programa en un PLC que cablear un circuito entero. Los usuarios finales pueden modificar el programa en el campo.
- Menores costos: Originalmente los PLC fueron diseñados para reemplazar los controles lógicos por relé. El ahorro en los costos gracias al uso de PLC ha sido tan significativo que el control por relés se está volviendo obsoleto, excepto para aplicaciones de alta potencia. Generalmente, si una aplicación requiere más de 6 relés de control, será usualmente menos costoso el instalar un PLC.
- Capacidad de comunicaciones: Un PLC puede comunicarse con otros controladores o equipos de computación. Pueden ser interconectados para realizar funciones como: control de supervisión, recopilación de información, monitoreo de dispositivos y parámetros de procesos, carga y descarga de programas.
- Tiempos de respuesta más rápidos: los PLC operan en tiempo real, lo cual significa que ante un evento que se dé en el campo tendrá como resultado una operación o que una salida tome efecto: Las máquinas pueden procesar miles de objetos por segundo y estos gastan sólo una fracción de segundo en frente de un sensor, lo cual requiere que el PLC tenga una rápida capacidad de respuesta.
- Mucho más fácil de reparar: los PLC tienen diagnósticos residentes y funciones de sobre posición que permiten a los usuarios de forma sencilla rastrear y corregir problemas de hardware y software.

#### **2.2.4 Estructura de un PLC**

Para explicar el funcionamiento de un PLC, deben distinguirse las siguientes partes:

- Interfaces de entradas y salidas.
- Central Processing Unit (CPU) o Unidad Central de Procesamiento.
- Memoria.
- Dispositivos de Programación.

Explicación de su funcionamiento:

El usuario ingresa el programa a través del dispositivo adecuado (un dispositivo de programación o PC) y éste es almacenado en la memoria. El CPU procesa la información que recibe del exterior a través de la interfaz de entrada y de acuerdo con el programa, activa una salida a través de la correspondiente interfaz de salida. Evidentemente, las interfaces de entrada y salida se encargan de adaptar las señales internas a niveles del CPU. Por ejemplo, cuando el CPU ordena la activación de una salida, la interfaz adapta la señal y acciona un componente (transistor, relé, etc.).

Al comenzar el proceso, el CPU lee el estado de las entradas. A continuación, ejecuta la aplicación empleando el último estado leído. Una vez completado el programa, el CPU ejecuta tareas internas de diagnóstico y comunicación. Al final del ciclo se actualizan las salidas.

El tiempo de ciclo depende del tamaño del programa, del número de E/S (entradas y salidas) y de la cantidad de comunicación requerida: leer entradas, ejecutar programa, ciclo PLC, actualizar salidas, diagnósticos y comunicación.

### **2.2.5 Protocolo de comunicación**

En informática y telecomunicación, un protocolo de comunicaciones es un sistema de reglas que permiten que dos o más entidades de un sistema de comunicación se comuniquen entre ellas para transmitir información por medio de cualquier tipo de variación de una magnitud física. Se trata de las reglas o el estándar que define la sintaxis, semántica y sincronización de la comunicación, así como también los posibles métodos de recuperación de errores. Los protocolos pueden ser implementados por hardware, por software, o por una combinación de

ambos. También se define como un conjunto de normas que permite la comunicación entre ordenadores, estableciendo la forma de identificación de estos en la red, la forma de transmisión de los datos y la forma en que la información debe procesarse.

Carolina Lagos establece que: “Los buses de datos que permiten la integración de equipos para la medición y control de variables de proceso, reciben la denominación genérica de buses de campo. Un bus de campo es un sistema de transmisión de información (datos) que simplifica enormemente la instalación y operación de máquinas y equipamientos industriales utilizados en procesos de producción” [8].

Los autómatas programables suelen estar conectados a toda una red de dispositivos que sirven en su conjunto para el control, manejo y supervisión del proceso automatizado. Estos dispositivos pueden ser sensores, actuadores, PC, otros PLCs, etc.

El desarrollo de las redes industriales ha permitido conectar todos estos dispositivos, aumentando el rendimiento de los sistemas y proporcionando nuevas posibilidades como lo

son:

- Intercambio de datos entre sectores del proceso y entre departamentos.
- Visualización y supervisión de todo el proceso productivo.
- Toma de datos del proceso más rápido.
- Programación a distancia, sin necesidad de estar a pie de fábrica

Los sistemas de comunicación utilizan formatos bien definidos (protocolo) para intercambiar mensajes. Cada mensaje tiene un significado exacto destinado a obtener una respuesta de un rango de posibles respuestas predeterminadas para esa situación en particular. Normalmente, el comportamiento especificado es independiente de cómo se va a implementar. Los protocolos de comunicación tienen que estar acordados por las partes involucradas. Para llegar a dicho acuerdo, un protocolo puede ser desarrollado dentro de estándar técnico. Un protocolo de comunicación, también llamado en este caso protocolo de red, define la forma en la que los distintos mensajes o tramas de bit circulan en una red de computadoras.

Los protocolos de comunicación industrial más utilizados son: *Process Field Bus* (PROFIBUS), FIELDBUS, MODBUS, *High way Addressable Remote Transducer* (HART).

### 2.2.6 Multi point interfaz

Por sus siglas en inglés MPI, es una interfaz desarrollada por la empresa Siemens y su objetivo es la de comunicar múltiples PLC S7-300/400 y equipos de programación en una red de tipo *multi punto* donde puede haber más de dos equipos conectados. Para programar estos modelos de PLC se conectan a computadoras por medio de adaptadores MPI/USB.

M. Angarita establece: “El SIMATIC S7-300 tiene varios mecanismos de comunicación. La comunicación de resultados transmitidos por las redes utilizando bloques de comunicación se hace mediante el servicio de comunicación global de datos, las redes de CPU pueden intercambiar datos con cada una de las otras unidades centrales de procesamiento, esto permite a un CPU acceder a la memoria de datos de otra CPU. La comunicación global de datos solo puede ser enviada vía interfaces multipunto (MPI). La interface multipunto (MPI) está integrada al CPU; para conexión simultánea de los mandos de programación, PC y sistemas de automatización SIMATIC S7, M7 o C7” [3].

Con este puerto se puede comunicar fácilmente a distancias reducidas sin requerir módulos adicionales. Se pueden enviar datos a 4 distintos aparatos al mismo tiempo y utilizando siempre el mismo puerto a una velocidad de 187,5 Kbits cada segundo. Para pequeñas redes de comunicación o pequeños volúmenes de datos la CPU ofrece el servicio de datos globales, que permite intercambiar cíclicamente cantidades de datos en paquetes de hasta 22 bytes como máximo.

### 2.2.7 Lenguaje de programación

Es un conjunto de símbolos y textos inteligibles por la unidad de programación, que le sirven al usuario para codificar sobre un cierto autómatas las instrucciones deseadas.

Los lenguajes de programación usados en automatización son:

- Basados en texto: lista de instrucciones; texto estructurado.
- Gráficos: diagrama en escalera; diagrama de bloques funcionales.

Un lenguaje de programación es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como puede ser un ordenador. En el caso de los PLCs, los lenguajes de programación surgieron al mismo tiempo que la aparición del primer PLC, en 1968. Así se explica porque no se utilizaron para este fin lenguajes de programación de alto nivel como Pascal y C y, en su lugar, se emplearon otros lenguajes más simples y fáciles de entender.

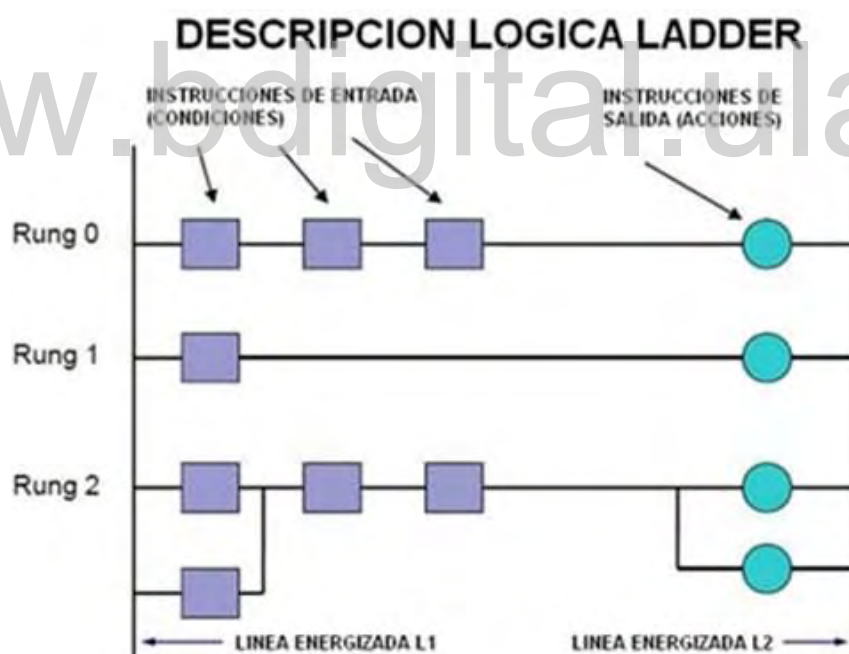
En el estándar IEC 61131 se consideró estos lenguajes para la programación de los PLCs. Los lenguajes de programación en PLCs fueron diseñados para establecer una comunicación directa entre el usuario y la máquina. Gracias a este vínculo, se es capaz de crear un programa con las instrucciones necesarias para controlar el comportamiento de cualquier proceso o máquina [9]. En el estándar se definieron los siguientes 5 lenguajes:

- Diagrama de Funciones Secuenciales (SFC) – un lenguaje de bloques de funciones secuenciales.
- Diagrama de Bloques de Funciones (FBD) – un lenguaje de diagramas de bloques secuenciales.
- Diagramas de Tipo Escalera (LAD) – un lenguaje de diagramas de relés (denominado de tipo escalera).
- Texto Estructurado (ST) – un lenguaje de alto nivel como el del tipo de texto estructurado (similar a C y, sobre todo a Pascal).
- Lista de instrucciones (IL o STL) – lenguaje de tipo ensamblador con uso de acumuladores.

En resumen, los lenguajes de programación para PLC son de dos tipos, visuales y escritos. Los visuales (SFC, FBD y LAD) admiten estructurar el programa por medio de símbolos gráficos. Los escritos (ST e IL o STL) son listados de sentencias que describen las funciones a ejecutar.


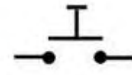

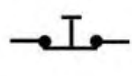
### 2.2.8 Diagrama de programación en escalera

Es el lenguaje gráfico más utilizado en la programación de PLCs. Consiste básicamente en la conexión gráfica entre variables de tipo Booleano, comparable a los antiguos controladores de tipo relé, donde se representa el flujo de energía en diagramas de circuitos eléctricos. Se le denomina de tipo escalera debido a la similitud de cada segmento del programa al escalón de una escalera. “Este lenguaje permite representar gráficamente el circuito de control de un proceso, con ayuda de símbolos de contactos normalmente cerrados (N.C.) y normalmente abiertos (N.A.), relés, temporizadores, contadores, registros de desplazamiento, etc. Cada uno de estos símbolos representa una variable lógica cuyo estado puede ser verdadero o falso” [10]. Para STEP 7, este lenguaje se conoce como LAD (Ladder Logic). La figura 2 y 3 muestra un ejemplo de un programa de este tipo.



**Figura 2. Representación escalera. Fuente: Rocatek.**



- Contacto normalmente abierto:  ≡ 
- Contacto normalmente cerrado:  ≡ 

### Ejemplo:

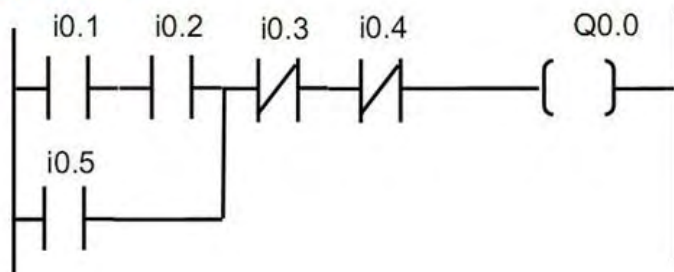


Figura 3. Representación escalera. Fuente: profesorpaul.blospot.

## 2.2.9 Componentes más comunes del lenguaje en escalera

### 2.2.9.1 Temporizadores

Es un elemento que permite poner cuentas de tiempo con el fin de activar o desactivar bobinas pasado un cierto tiempo desde la activación. Existen varios tipos de temporizadores entre los que se encuentran: Con retardo a la conexión, con retardo a la conexión con memoria, con retardo a la desconexión, como impulso y arrancar. El temporizador mostrado en la figura 4 se utiliza en el programa STEP 7 LITE y es de tipo retardo a la conexión. Cuando se produce un flanco de subida en la entrada S, se activa el temporizador con el tiempo establecido, en este caso 0.5 segundos. Al culminar el tiempo, la salida Q se establece en nivel alto siempre y cuando el nivel en la entrada S se mantenga en alto.

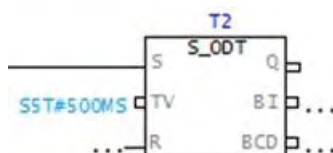


Figura 4. Temporizador con retardo a la conexión. Fuente: Elaboración propia.

### 2.2.9.2 Contadores

Es capaz de llevar el cómputo de las activaciones de sus entradas, por lo que resulta adecuado para memorizar sucesos que no tengan que ver con el tiempo pero que se necesiten realizar un determinado número de veces. La figura 5 muestra un contador hacia arriba usado en STEP 7 LITE. Para realizar la cuenta debe haber flancos de subida en la entrada CU. Cada flanco aumenta en 1 la cuenta del contador. El contador C0 aumenta su cuenta en 1 cada vez que en la entrada I124.0 hay un flanco de subida. El valor de cuenta se almacena en la variable MW10. Si el valor de cuenta cambia de 0 a 1, la salida Q cambia de nivel bajo a alto. Para reiniciar la cuenta es necesario establecer la entrada R controlada por I124.5 en nivel alto.

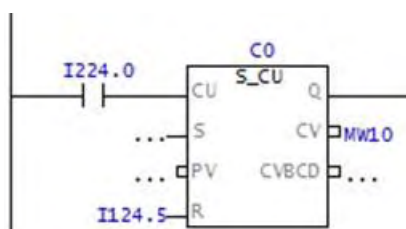


Figura 5. Contador UP. Fuente: Elaboración propia.

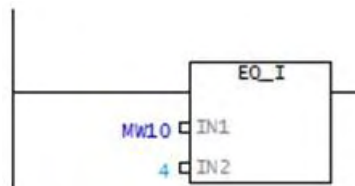
### 2.2.9.3 Comparadores

Permite comparar el valor de dos variables y entrega un nivel alto dependiendo de la condición que se establezcan previamente. Por ejemplo, si se establece un comparador “mayor que” entre dos variables, este determina si el primer valor dado es mayor que el segundo, si se cumple esta condición el elemento entrega un nivel alto. Se pueden comparar valores enteros o de punto flotante. Las condiciones pueden ser “mayor que”, “mayor o igual que”, “menor”, “menor o igual que”, “igual” o “desiguales”. La figura 6 muestra un comparador de condición “igual que”, compara el valor de dos enteros IN1 y IN2.

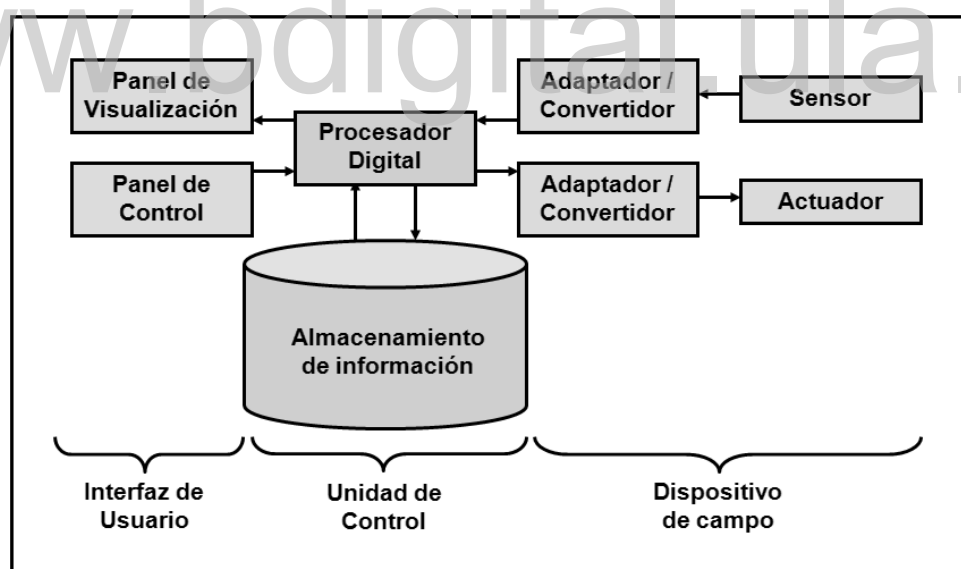
### 2.2.10 SCADA

Los sistemas SCADA son el conjunto de equipos y programas que permiten (de forma remota) la supervisión, control y registro de los datos provenientes de los equipos situados en una planta de producción. Los datos se obtienen de sensores que envían las señales eléctricas a

procesadores digitales. Desde este punto, la información se envía a almacenamiento, paneles de visualización y control. Si se da alguna orden a través del panel de control, el procesador digital envía la señal eléctrica hasta el actuador. La figura 7 muestra una representación generalizada de un sistema SCADA.



**Figura 6. Comparador de igualdad entre enteros. Fuente: Elaboración propia.**



**Figura 7: Representación de un sistema SCADA. Fuente: Blanco O. y otros, 2009.**

Según las iniciales SCADA se puede definir cada una de sus funciones. Hernández [5] las define de la siguiente manera:

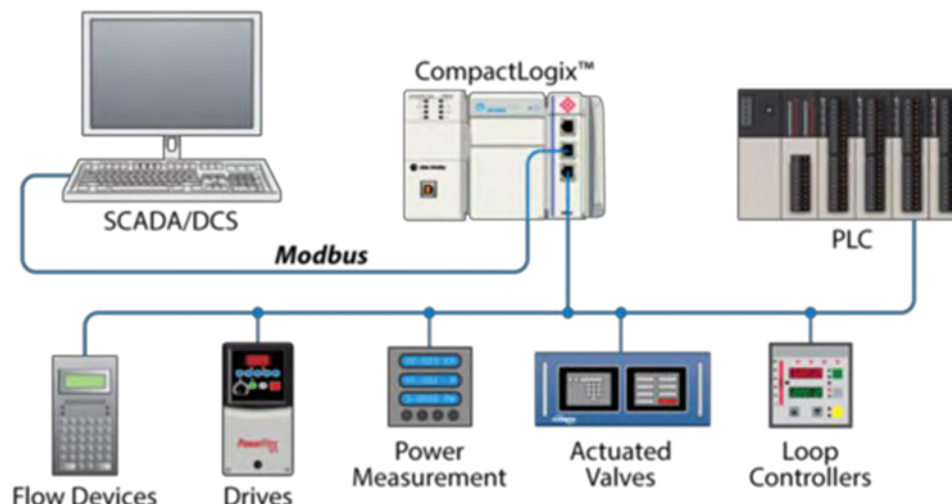
- Supervisión. -Por medio de un monitor es posible que un operador observe la evolución de las variables del proceso, lo que permite las tareas de mantenimiento y estadísticas de fallas.
- Control. -El sistema puede automáticamente activar o desactivar equipos remotamente (por ejemplo, abrir válvulas y activar motores). El operador puede ejecutar acciones de control manual para modificar la evolución del proceso en situaciones irregulares.
- Adquisición de datos. -Recolectar, procesar, almacenar y mostrar la información recibida desde los equipos de campo.

Además de estas características, los sistemas SCADA permiten:

- Generación de reportes. -Con los datos adquiridos se pueden generar representaciones gráficas, predicciones, control estadístico, gestión de la producción, gestión administrativa y financiera, etc.
- Representación de señales de alarma. -A través de las señales de alarma se logra alertar al operador frente a una falla o la presencia de una condición perjudicial o fuera de lo aceptable. Estas alarmas pueden ser tanto visuales como sonoras.

Los sistemas SCADA se pueden constituir en dos grandes grupos que son *Hardware* y *Software*.

- Hardware. -Son todos los equipos necesarios para recolección, transmisión y procesamiento de los datos de campo. Estos equipos son: PLCs, RTUs, computadoras, unidad terminal maestra (MTU), actuadores y sensores (figura 8). Al respecto, Blanco [11] establece que los SCADA utilizan diversas arquitecturas de comunicación, una de las más básicas utiliza un servidor centralizado y uno o varios clientes (PLC, Unidad Terminal Remota (RTU), Computadores Automatas Programables (PAC)).
- Software. -Es el programa donde se construye el interfaz humano máquina, permite restringir el acceso de personas no autorizadas al sistema, generar señales de alarmas en caso de falla, almacena los registros de las variables de campo en una base de datos.



**Figura 8. Hardware en un sistema SCADA. Fuente: IndiaMART.**

### 2.2.11 Human machine interfaz

La Interfaz Hombre-Máquina (HMI) es la interfaz entre el proceso y los operarios en los sistemas SCADA; se trata básicamente de un panel de instrumentos del operario. Es la principal herramienta utilizada por operarios y supervisores de línea para coordinar y controlar procesos industriales y de fabricación. El HMI traduce variables de procesos complejos en información útil y procesable.

La función de los HMI consiste en mostrar y controlar las variables en tiempo real. Proporcionan gráficos de procesos visuales que aportan significado y contexto al estado del motor y de la válvula, niveles de depósitos y otros parámetros del proceso. Suministran información operativa al proceso, permiten controlar y optimizar al regular los objetivos de producción y de proceso. En la figura 9 se muestra un ejemplo de HMI.

### 2.2.12 Software libre

En 1986 Richard Stallman [12], fundador de la “Free Software Foundation (FSF)”, define el término software libre, en la cual establece que todo programa informático que cumpla con 4 leyes fundamentales, puede considerarse software libre.

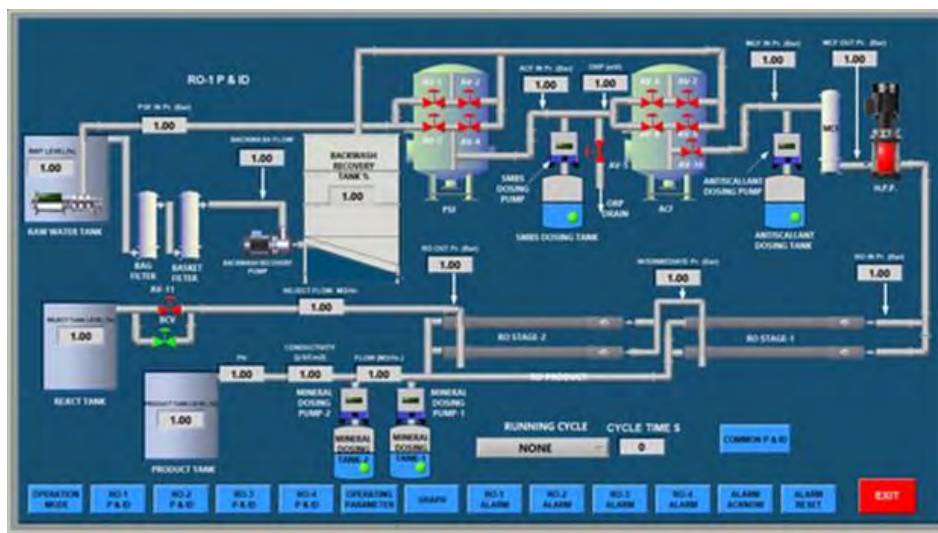


Figura 9. Ejemplo de HMI. Fuente: IndiaMART.

Libertad 0. La libertad de usar el programa, con cualquier propósito (uso).

Libertad 1. La libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a las propias necesidades (estudio).

Libertad 2. La libertad de distribuir copias del programa, con lo cual se puede ayudar a otros usuarios (distribución).

Libertad 3. La libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie (mejora).

Las cuatro libertades implican tener acceso al código fuente, porque estudiar y modificar software sin su código fuente no tiene sentido.

En la definición de software libre no se establece un precio. Solo contempla el uso libre del mismo, por lo que no se prohíbe la comercialización. El término “Free software” en inglés es ambiguo debido a que se interpreta generalmente como gratis. Esta confusión ha hecho que otros movimientos difundan el término *open source*, el cual establece únicamente aspectos técnicos. Los motivos dados por Richard Stallman y la fundación FSF para hablar de software libres utilizan los aspectos éticos y filosóficos de la libertad.

“«Software libre» es el software que respeta la libertad de los usuarios y la comunidad. A grandes rasgos, significa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Es decir, el «software libre» es una cuestión de libertad, no de precio. Para entender el concepto, piense en «libre» como en «libre expresión», no como en «barra libre». En inglés, a veces en lugar de «free software» se dice «libre software», empleando ese adjetivo del francés o español, derivado de «libertad», para mostrar que se quiere decir que el software es gratuito.

Promovemos estas libertades porque todos merecen tenerlas. Con estas libertades, los usuarios (tanto individualmente como en forma colectiva) controlan el programa y lo que este hace. Cuando los usuarios no controlan el programa, decimos que dicho programa «no es libre», o que es «privativo». Un programa que no es libre controla a los usuarios, y el programador controla el programa, con lo cual el programa resulta ser un instrumento de poder injusto” [12].

A pesar de las 4 leyes básicas, estas no impiden que se puedan aplicar ciertas reglas de uso del software. En la mayoría de los casos se establecen condiciones de uso por medio de una licencia pública general (GPL). En este tipo de licencia, el autor conserva los derechos de autor (copyright). Esta permite la modificación y redistribución del software, solo bajo las condiciones estipuladas en la misma.

### **2.2.13 Servidor OPC**

Es un software que sirve de interfaz entre una o varias fuentes de datos y una o varias aplicaciones clientes. Las fuentes de datos pueden ser PLCs, módulos I/O, controladores etc. Las aplicaciones clientes en general son SCADAs, HMIs, generadores de informes, generadores de gráficos etc. El servidor OPC es una aplicación de software que cumple una o más especificaciones definidas por la fundación OPC [13].

OPC es una aplicación de Object Linking and Embedding (OLE) creada por Microsoft para el control de procesos. Los Servidores OPC clásicos utilizan la infraestructura COM/DCOM de Microsoft Windows para el intercambio de datos. Lo que significa que esos Servidores OPC

deben instalarse bajo el Sistema Operativo de Microsoft Windows. Un Servidor OPC puede soportar comunicaciones con múltiples Clientes OPC simultáneamente.

“En una arquitectura Cliente OPC/ Servidor OPC, el Servidor OPC es el esclavo mientras que el Cliente OPC es el maestro. Las comunicaciones entre el Cliente OPC y el Servidor OPC son bidireccionales, lo que significa que los Clientes pueden leer y escribir en los dispositivos a través del Servidor OPC” [13].

Existen cuatro tipos de servidores OPC definidos por la OPC Foundation, y son los siguientes:

- Servidor OPC DA – OPC Data Access - especialmente diseñado para la transmisión de datos en tiempo real.
- Servidor OPC HDA– Basado en la especificación de Acceso a Datos Historiados que provee al Cliente OPC HDA de datos históricos.
- Servidor OPC A&E Server– Basado en la especificación de Alarmas y Eventos – transfiere Alarmas y Eventos desde el dispositivo hacia el Cliente OPC A&E.
- Servidor OPC UA – Basado en la especificación de Arquitectura Unificada – basado en el set más nuevo y avanzado de la OPC Foundation, permite a los Servidores OPC trabajar con cualquier tipo de datos.

El objetivo fundamental del servidor OPC es traducir los datos obtenidos de la fuente de datos, a un protocolo que sea compatible con una a varias aplicaciones clientes. En el caso del cliente y el servidor OPC, el tipo de datos deben ser el mismo. Por ejemplo, una aplicación cliente compatible con OPC DA solo podrá escribir o leer a un servidor OPC DA. En la figura 10 se muestra una representación de la comunicación OPC.



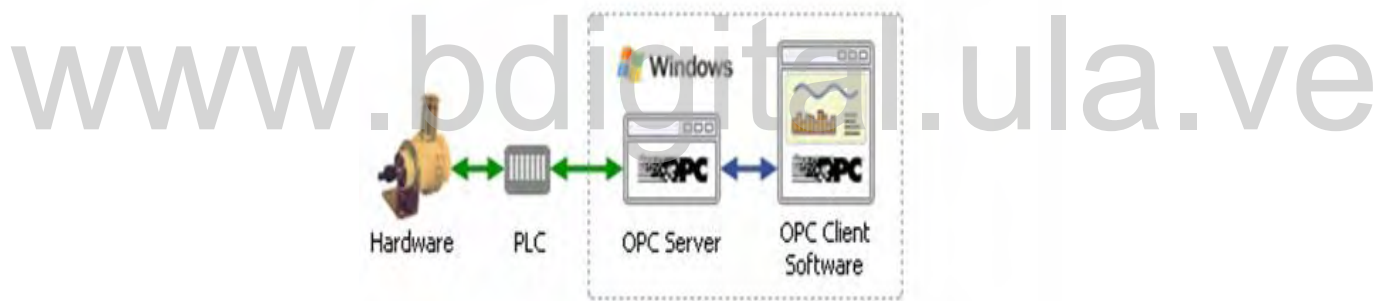


Figura 10. Representación OPC, Fuente: MatrikonOPC.

# CAPÍTULO 3

## CONFIGURACIÓN DEL SISTEMA SCADA

Este capítulo se inicia detallando los equipos utilizados para implementar el sistema SCADA. Después se explica el procedimiento que se utilizó para seleccionar los programas en base a las características de los equipos disponibles y como se configuran cada uno de ellos.

### 3.1 Selección del PLC a utilizar en el sistema SCADA

Para la selección del PLC a utilizar se determinaron los puertos de comunicación de cada equipo disponible en el laboratorio de control de la escuela de ingeniería eléctrica. Se concluyó que todas las computadoras del laboratorio disponen de comunicación USB, por lo que la selección del PLC se redujo a todos aquellos que tengan adaptadores a USB. Para implementar el sistema SCADA se decidió utilizar el modelo S7-300 (figura 11) de la marca Siemens, ya que es el equipo con más reciente fecha de fabricación y cuenta con adaptador MPI/USB (figura 12).



**Figura 11. PLC Siemens S7-300. Fuente: Siemens.**



**Figura 12. Adaptador MPI/USB. Fuente: Siemens.**

Este PLC tiene las siguientes especificaciones: Fuente de alimentación de 24V modelo: PS307 5A 307-1E00-0AA0; CPU 312C 312-5BE03-0AB0, DI10/DO6xDC24V (10 entradas, 6 salidas digitales); Módulo adicional modelo: SM323 DI8/DO8xDC24V 323-1BH01-0AA0 (8 entradas y 8 salidas digitales).

### **3.2 Selección del programa SCADA**

Para la selección del programa SCADA se realizó la búsqueda de software *libre* compatible con el protocolo de comunicación MPI/USB. Como resultado no se logró conseguir un programa SCADA compatible directamente con este protocolo. Se pudo determinar que algunos de estos programas pueden configurarse como cliente OPC. Como consecuencia, se dispuso a hallar un servidor OPC compatible con el protocolo antes mencionado. El servidor que cumple con estos requerimientos es IBH OPC.

A continuación se describen una serie de pasos para configurar el servidor

### 3.3 Configuración del Servidor IBH OPC

Para realizar pruebas con el servidor OPC se necesitó cargar al PLC un programa sencillo utilizando una entrada y una salida. Activando la salida Q124.0 con la entrada I124.0. La carga de secuencias lógicas al PLC se realizó con el programa STEP 7 Lite. El programa representando en lenguaje en escalera se muestra en la figura 13.

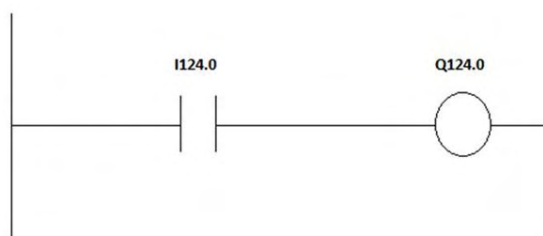


Figura 13. Programa de prueba, Fuente: Elaboración propia.

Después de cargar el programa en el PLC se procede a configurar el servidor OPC. Como primer paso se elige el protocolo “S7 Simatic Net” (figura 14).

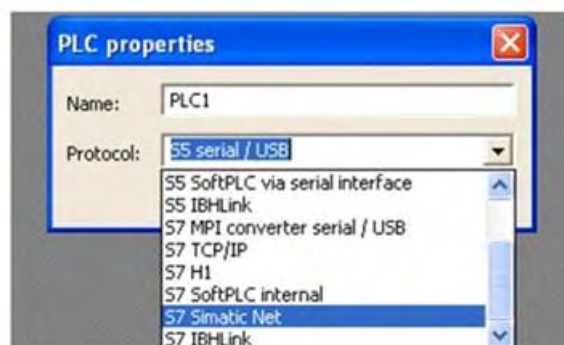
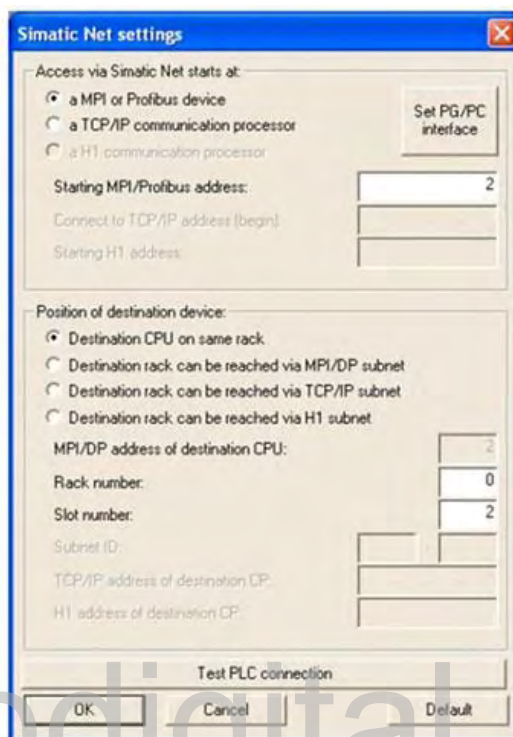
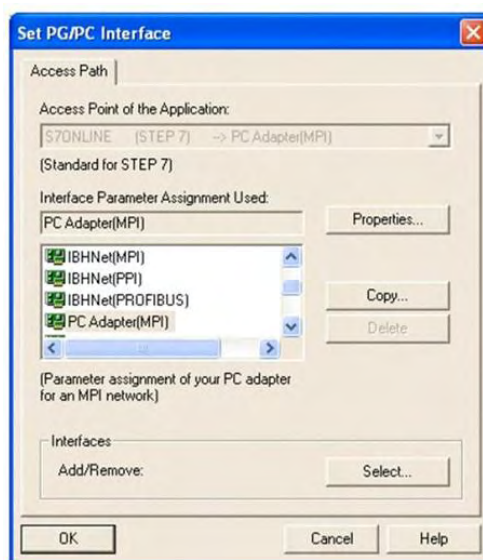


Figura 14. Elección de protocolo. Fuente: Elaboración propia.

Se escoge “a MPI or Profibus device”. Haciendo click en “Set PG/PC interface” (figura 15 y 16).

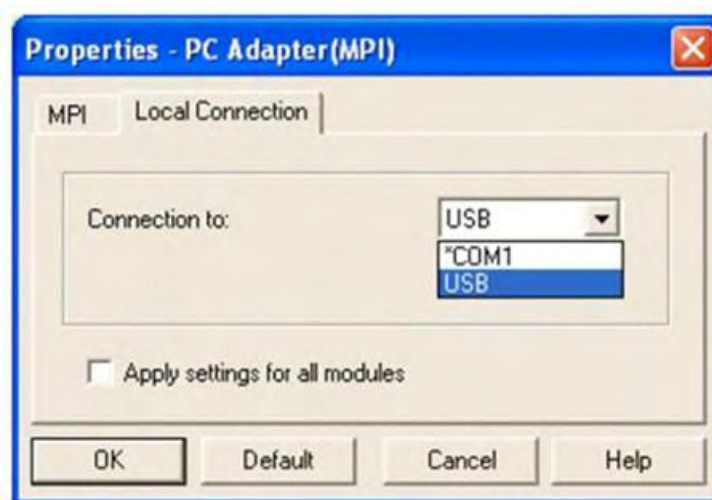


**Figura 15. Configuración de protocolo.**



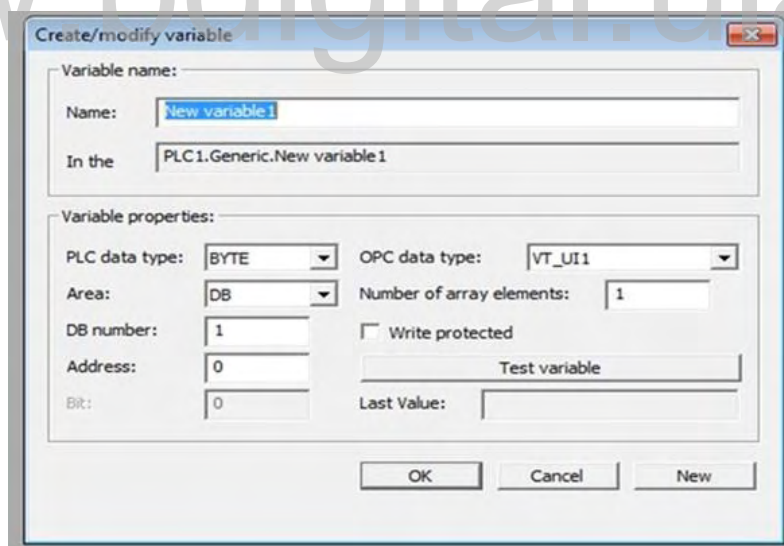
**Figura 16. Configuración de protocolo.**

Haciendo click en “Properties”, seguidamente en “local connection”. Eligiendo USB como se muestra en la figura 17.



**Figura 17. Conexión USB.**

Para probar la conexión con el PLC se presiona el icono “Test PLC conexión” (figura 16).



**Figura 18. Configuración de nueva variable.**

Se debe crear una variable para la entrada y otra para salida. Se le da un nombre a cada variable, el tipo de dato BOOL y la dirección de cada una (figuras 18 y 19).

Variable name:

Name: entrada1

In the PLC1.Generic.entrada1

Variable properties:

PLC data type: BOOL OPC data type: VT\_BOOL

Area: I Number of array elements: 1

DB number: 1 Write protected

Address: 124 Test variable

Bit: 0 Last Value:

OK Cancel New

Figura 19. Configuración de variable.

www.bdigital.ula.ve

Dando click en “Test variable” se muestra en “Last Value” se muestra el valor actual de la variable entrada1 (figura 20).

Test variable

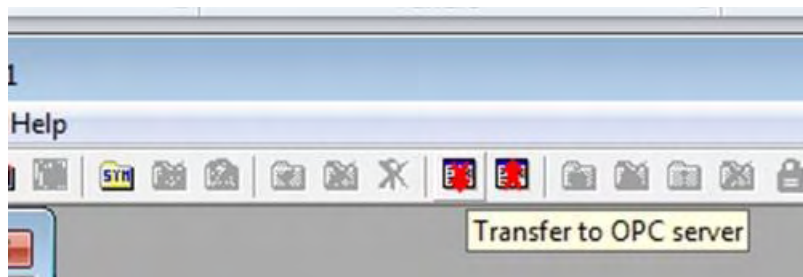
Last Value: TRUE

OK Cancel New

Figura 20. Comprobación de valor de variable.

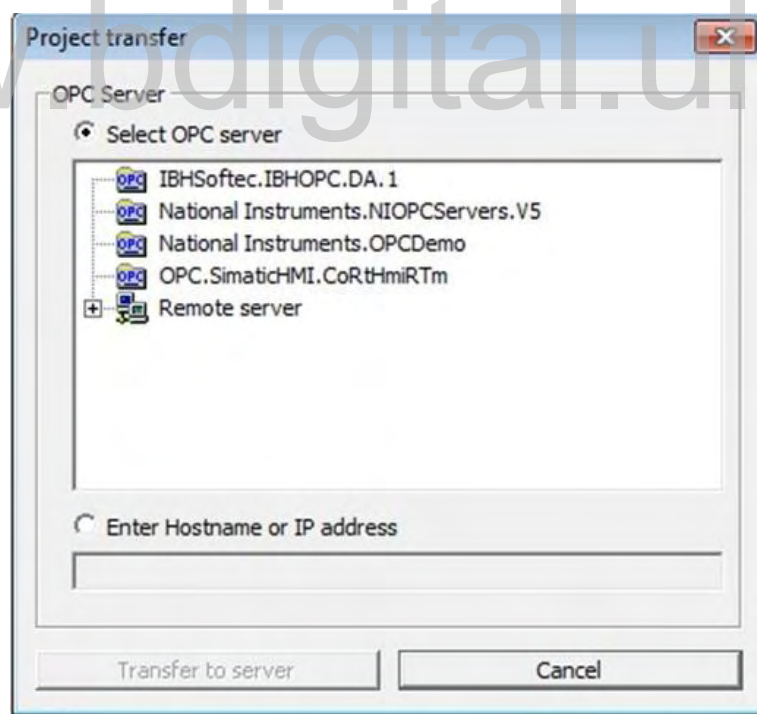


Para que cualquier cliente OPC reconozca al servidor OPC, es necesario transferir los datos al servidor OPC. En la barra de herramientas se elige “Transfer to OPC server” (figura 21).



**Figura 21. Transfiriendo a servidor OPC.**

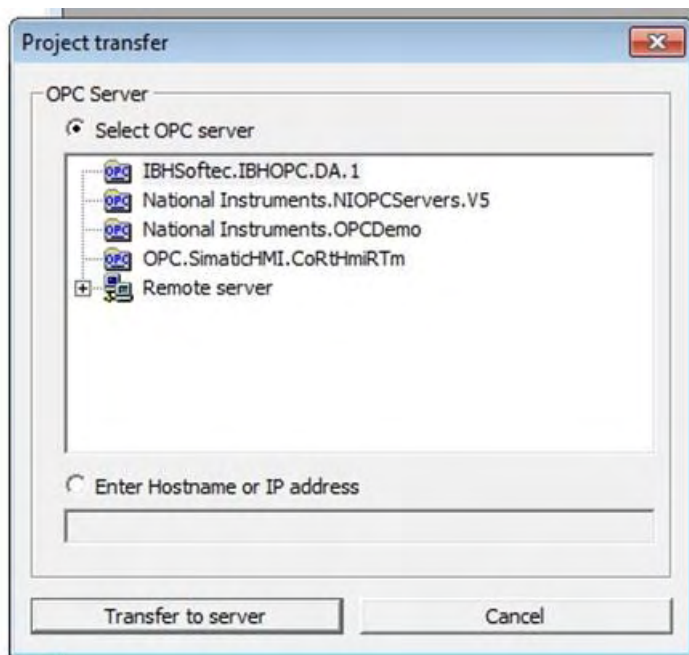
Para transferir los datos se necesita guardar el proyecto con un nombre y ubicación (figura 22).



**Figura 22. Transfiriendo a servidor OPC.**

Elegir IBHSoftec.IBHOPC.DA.1 y presionar “Transfer to server” (figura 23).





**Figura 23. Transfiriendo a servidor IBH.**

Al finalizar este paso, cualquier aplicación cliente puede reconocer al servidor IBH OPC server. Se comprobó que el estado que se muestra en el servidor OPC coincide con el estado de la variable en el PLC, confirmando el uso de este servidor para el proyecto.

### **3.4 Descripción de IndigoSCADA**

Como introducción a la descripción del programa se dan los motivos para la elección del mismo. En el proceso de investigación en la web se realizó una lista de los programas SCADA de software libre más utilizados. En ella destacan SCADABR y RapidSCADA. Se realizaron pruebas de comunicación con los programas anteriores configurándolos como cliente OPC. Con SCADABR no se logró establecer ningún tipo de comunicación con el servidor OPC. Utilizando RapidSCADA solo fué posible la recepción de datos provenientes del servidor OPC. Por último, IndigoSCADA permitió la comunicación bidireccional, razones por las cuales se decidió utilizar este programa para implementar el sistema SCADA.

IndigoSCADA es un software libre que sirve para implementar sistemas SCADA que utilicen los siguientes protocolos: MODBUS, DNP3, OPC DA 2.05, OPC A&E 1.1, IEC 60870-5-101, IEC 60870-5-103, IEC 60870-5-104, MQTT 3.1.1. Consta de 3 subprogramas: *manager*, *HMI designer*, *protocolo configuration*.

IndigoSCADA tiene las siguientes características:

- Uso de datos para generar reportes diarios, semanales y mensuales.
- Históricos y representaciones gráficas en tiempo real de los datos.
- Notificación de alarmas y eventos en tiempo real.
- Base de datos en tiempo real e históricos.
- Múltiples usuarios con diferentes derechos de acceso.
- Desarrollado bajo el estándar IEC 60870-5-101.
- Escrito en lenguaje C.
- Soporte de múltiples HMI.

Es una aplicación de 32 bits y pensada para Windows. Está demostrado su correcto funcionamiento en los siguientes sistemas operativos [14]:

- Microsoft Windows 2000 Professional 32-bit.
- Microsoft Windows XP Professional 32-bit with Service Pack 3.
- Microsoft Windows Vista Home Premium 32-bit with Service Pack 2.
- Microsoft Windows Vista Business 32-bit with Service Pack 2.
- Microsoft Windows Vista Ultimate 32-bit with Service Pack 2.
- Microsoft Windows Server 2003 Standard Edition 32-bit with Service Pack 2.
- Microsoft Windows 7 Ultimate/Enterprise 32-bit.
- Microsoft Windows 7 Professional 32-bit.
- Microsoft Windows 10 64-bit.

El manual de usuario [14] establece que el programa requiere una computadora con al menos 1GHz de memoria procesador y memoria RAM de 1GB. Cumpliendo con estos requerimientos se utilizó una computadora con Procesador Pentium(R) D 2.80GHz, memoria

RAM de 1GB y sistema operativo Microsoft Windows XP Professional 32-bit con Service Pack 3.

### 3.5 Configuración de IndigoSCADA

Primero se detalla el proceso para la Instalación de IndigoSCADA y después como configura.

Para instalar se ejecuta el archivo “indigo-scada-binaries-17-11-2018-13-00-rev308”, estar de acuerdo con las condiciones de uso y elegir la ubicación de instalación. Por defecto el programa se instala en el disco C. También se piden hacer excepciones de firewall para los siguientes procesos: dspserver, rtsqlserver, sqlserver, rtps\_mngr.

#### 3.5.1 IndigoSCADA como cliente OPC.

La configuración del protocolo OPC se hace por medio del símbolo del sistema (figura 24), donde se especifica el servidor OPC, la dirección IP donde se aloja y el identificador PROGID. Se ingresan las siguientes líneas de código:

```
cd C:\scada\bin
```

```
opc_client_da.exe -a 127.0.0.1 -p IBHSoftec.IBHOPC.DA -e configuration_file.sql -q {005F4CFE-830E-4961-A40C-407924572F98}
```

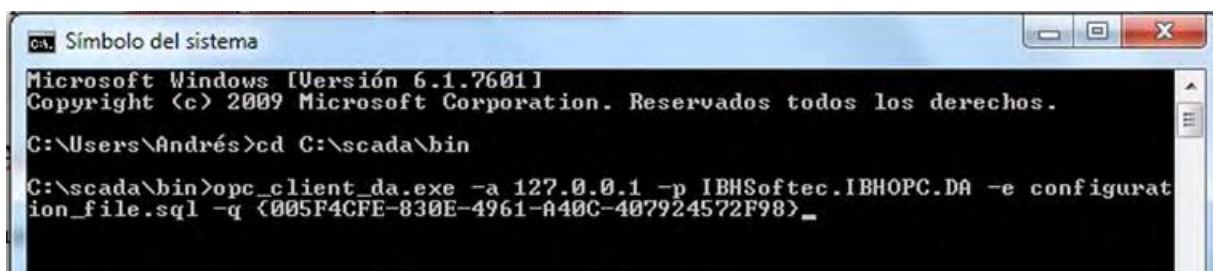
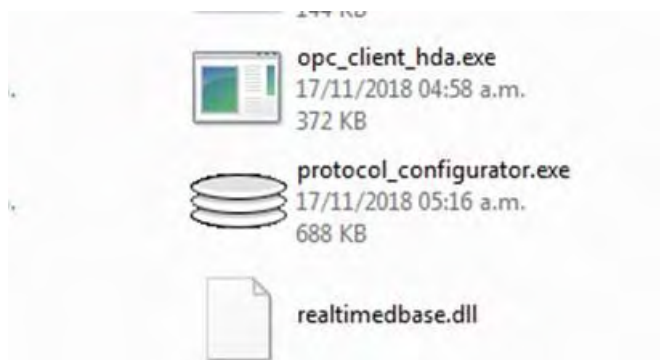


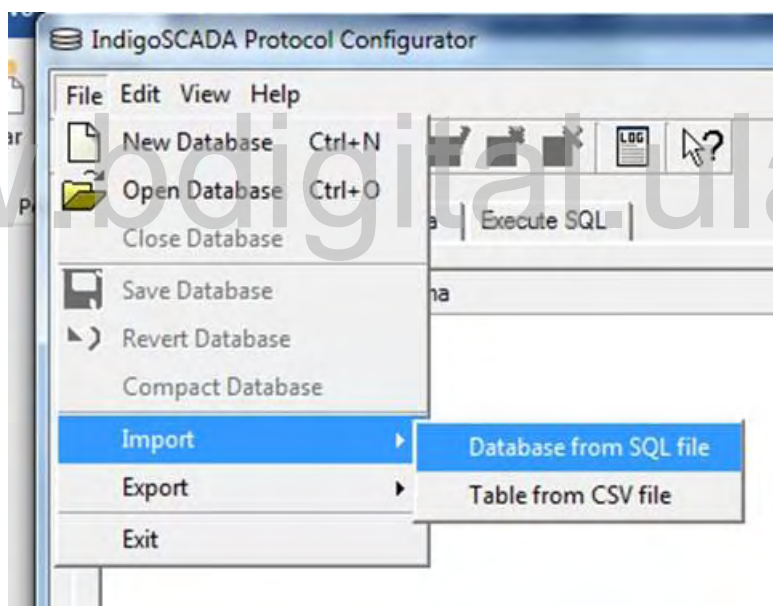
Figura 24. Símbolo del sistema. Fuente: Elaboración propia.

Esto guarda en el archivo “configuration\_file.sql” ubicado en la carpeta C:\scada\project, la lista de las etiquetas (ítems) disponibles en el servidor OPC. Para la creación de la base de

datos de la configuración del protocolo se ejecuta protocol\_configurator.exe (figura 25) ubicado en C:\scada\bin. Se selecciona File->Import->Database from SQL file (figura 26).

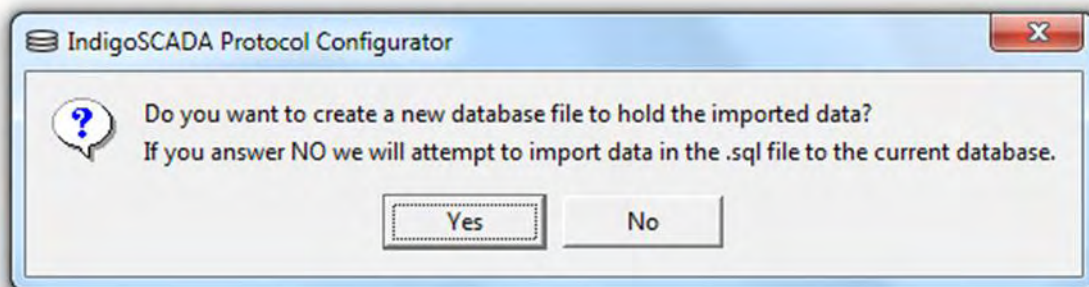


**Figura 25. Protocol configuration.**



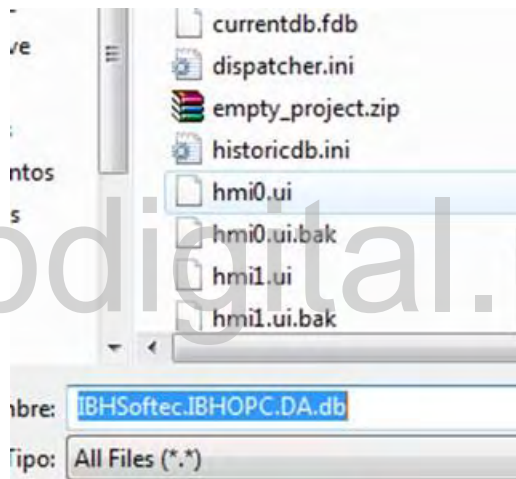
**Figura 26. Importando base de datos.**

A la pregunta mostrada en el recuadro de la figura 27 se selecciona “Yes”.



**Figura 27. Creando nuevo archivo de base de datos.**

El nombre de la base de dato debe ser el OPCServerProgID con extensión .db. En este caso es: IBHSoftec.IBHOPC.DA (figura 28).



**Figura 28. Creando nuevo archivo de base de datos.**

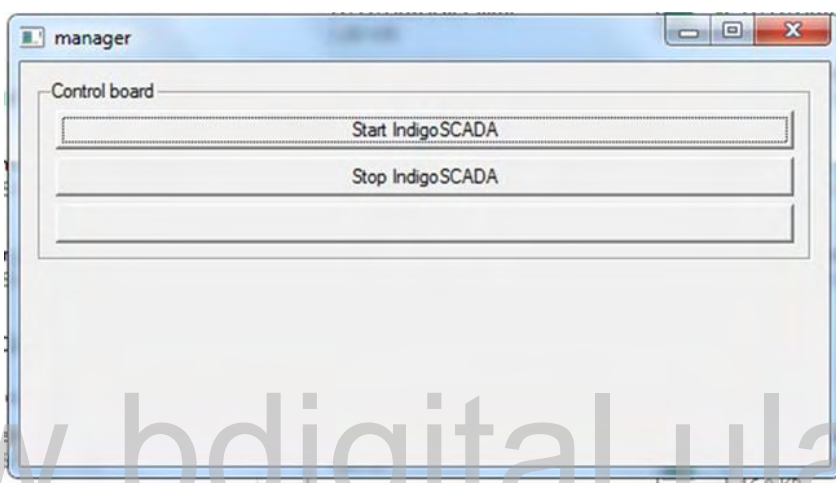
Esperar algunos segundos hasta que aparezca la ventana de confirmación de importación completada como se observa en la figura 29.

### 3.5.2 Creación de puntos SCADA.

Los puntos SCADA son el enlace entre las variables creadas en el servidor OPC e IndigoSCADA. Es decir, cada punto SCADA corresponde a una variable del servidor OPC. Para la creación de los nuevos puntos SCADA o “SCADA Points” se ejecuta manager.exe (figura 30).

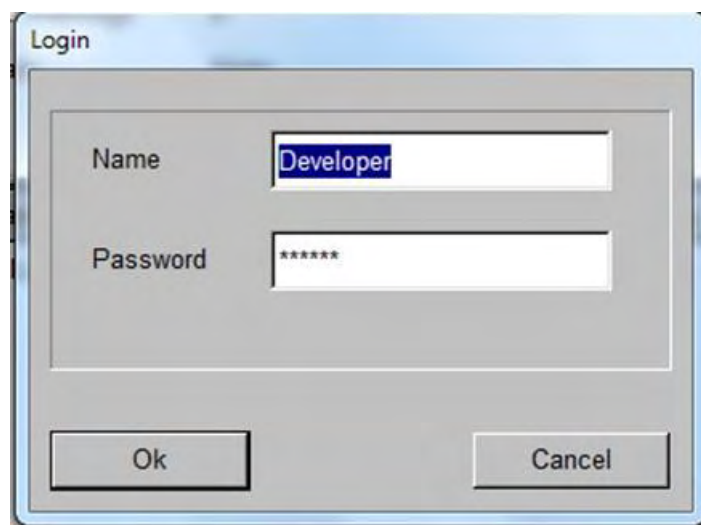


**Figura 29. Importación completada.**



**Figura 30. Iniciando IndigoSCADA.**

Se selecciona Start y aparece la ventana como se muestra en la figura 31.



**Figura 31. Iniciando IndigoSCADA.**

Seleccionar el menú: Configure\Configure Units (figura 32).

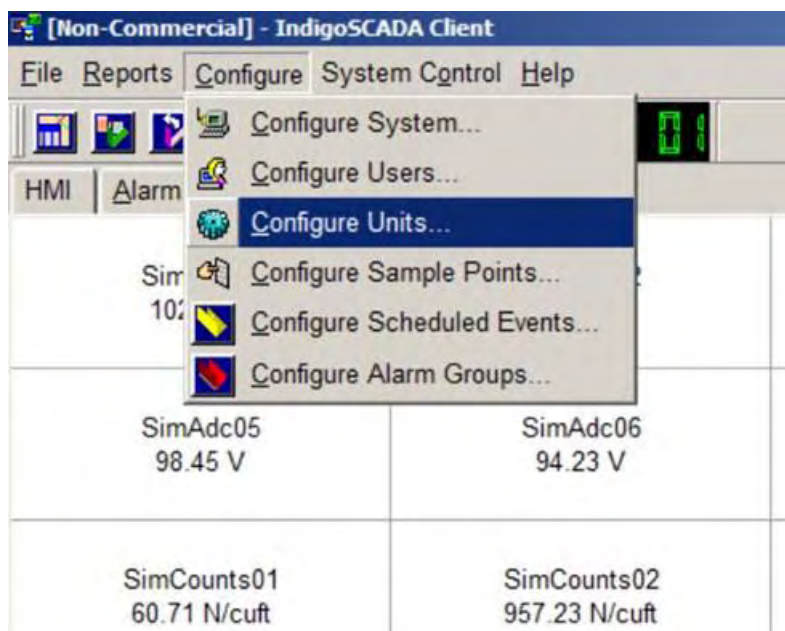


Figura 32. Configurando como cliente OPC, paso 1.

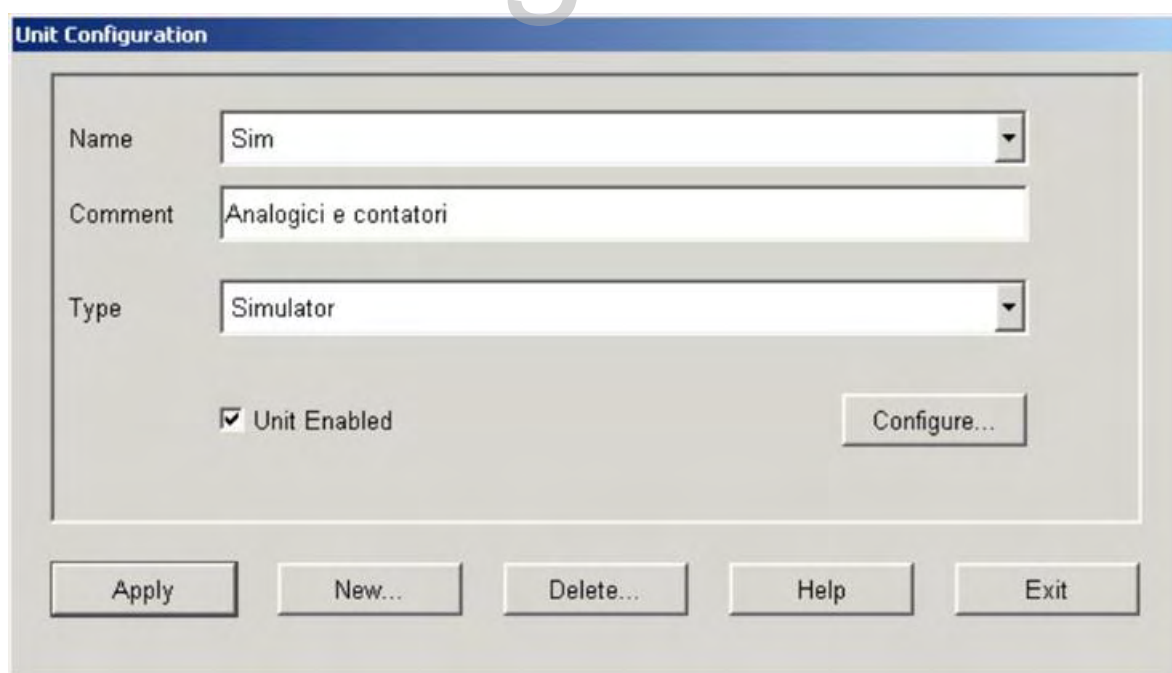


Figura 33. Configurando como cliente OPC, paso 2.



Presionando el botón New. Se escribe en “Name” OPC y se presiona OK (figuras 33 y 34).

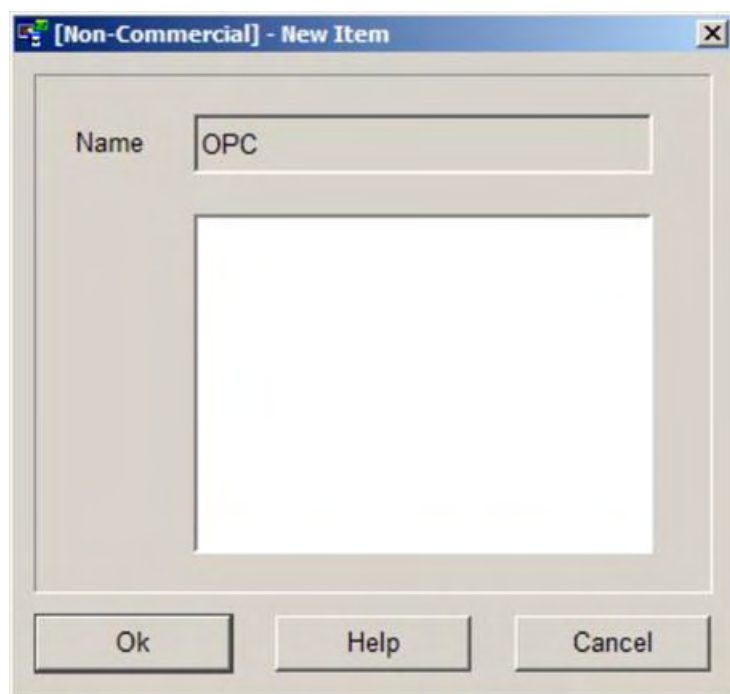


Figura 34. Configurando como cliente OPC, paso 3.

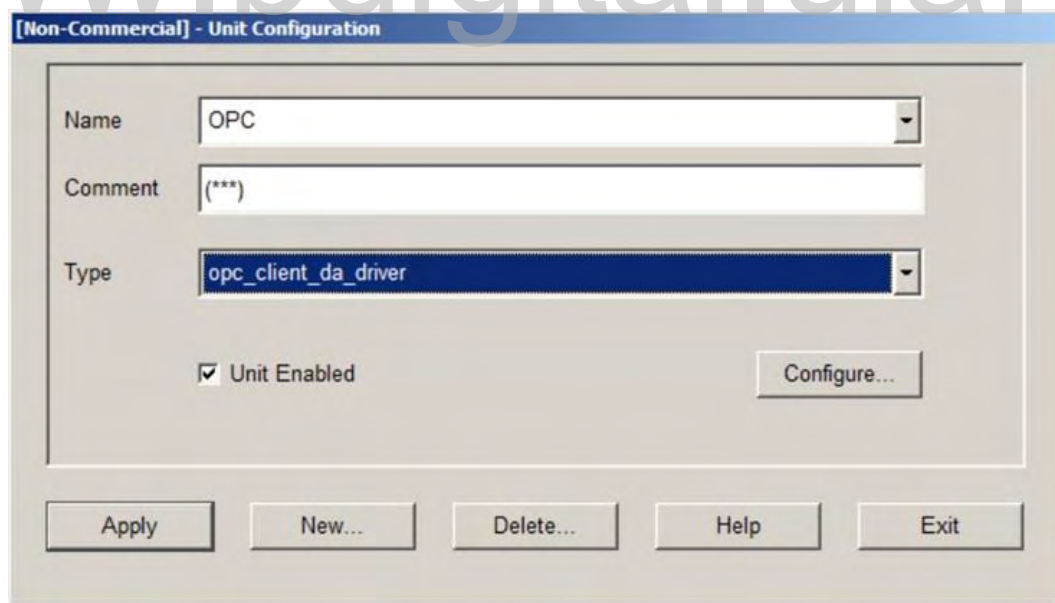
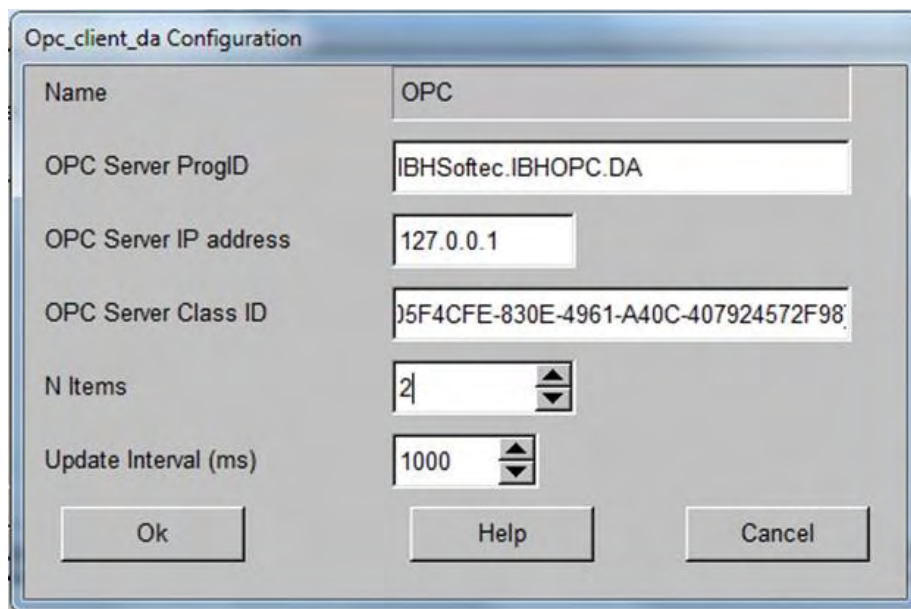


Figura 35. Configurando como cliente OPC, paso 4.

En la figura 35 se muestra la elección de “opc\_client\_da\_driver” en Type.

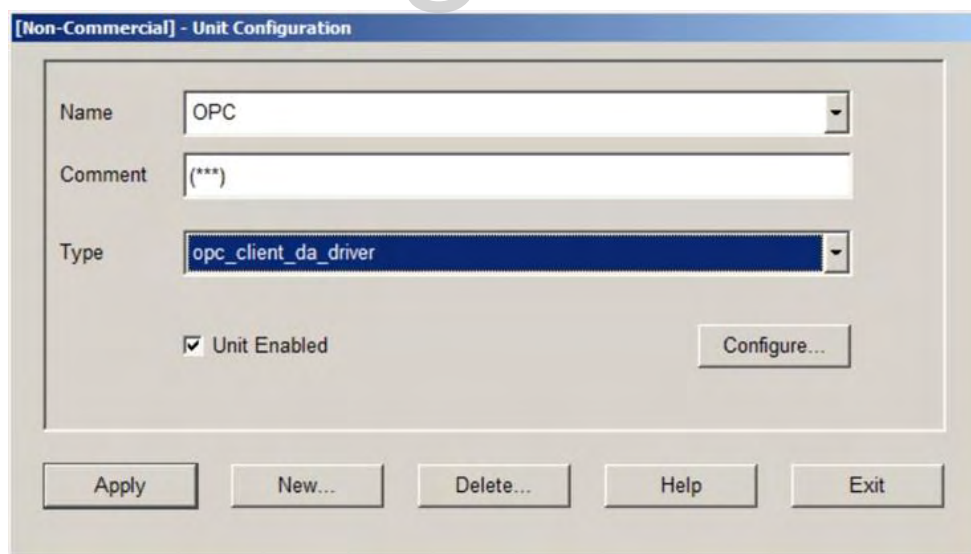


Eligiendo “Configure” (figura 35), aparece el recuadro “Opc\_client\_da Configuration” el cual se llena con los datos mostrados en la figura 36. El número de Items (N Items) y el tiempo de actualización de datos (Update Interval) varían en función de la práctica que se vaya a utilizar.



Name	OPC
OPC Server ProgID	IBHSoftec.IBHOPC.DA
OPC Server IP address	127.0.0.1
OPC Server Class ID	05F4CFE-830E-4961-A40C-407924572F98
N Items	2
Update Interval (ms)	1000

Figura 36. Configurando como cliente OPC, paso 5.



Name	OPC
Comment	(***)
Type	opc_client_da_driver
Unit Enabled	<input checked="" type="checkbox"/>

Figura 37. Configurando como cliente OPC, paso 6.

Presionando “Apply” (figura 37), se aplican los cambios realizados. En el menú “System Control” (figura 38) se elige “Restart Monitor”. Presionando Ok, aparecen rectángulos blancos que representan los OPCPoints.

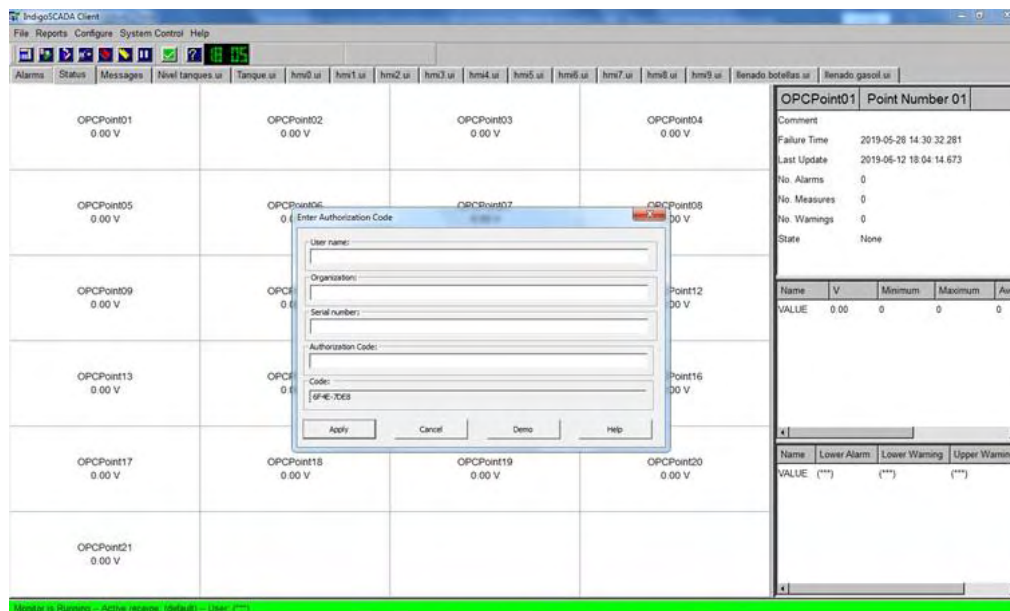


Figura 38. OPCPoints sin comunicación.

La figura 39 muestra el tipo de licencia del servidor IBH OPC a utilizar. Se elige la licencia “Demo”.

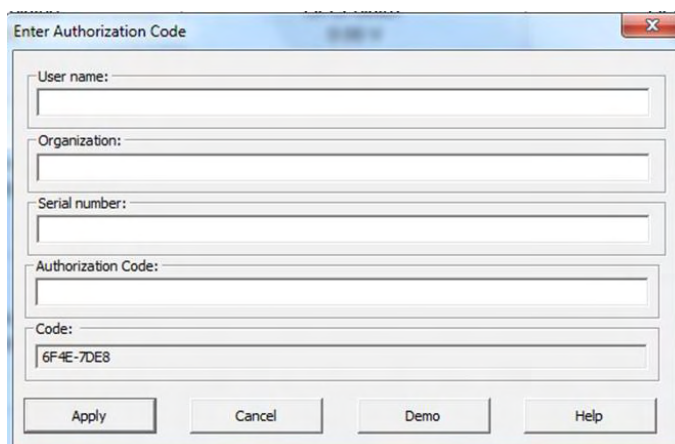


Figura 39. Licencias del servidor IBH OPC.

La figura 40 indica que la versión demo terminará después de media hora. Al cumplirse este tiempo aparece en pantalla el rectángulo de la figura 39, por lo que es posible reiniciar la comunicación.

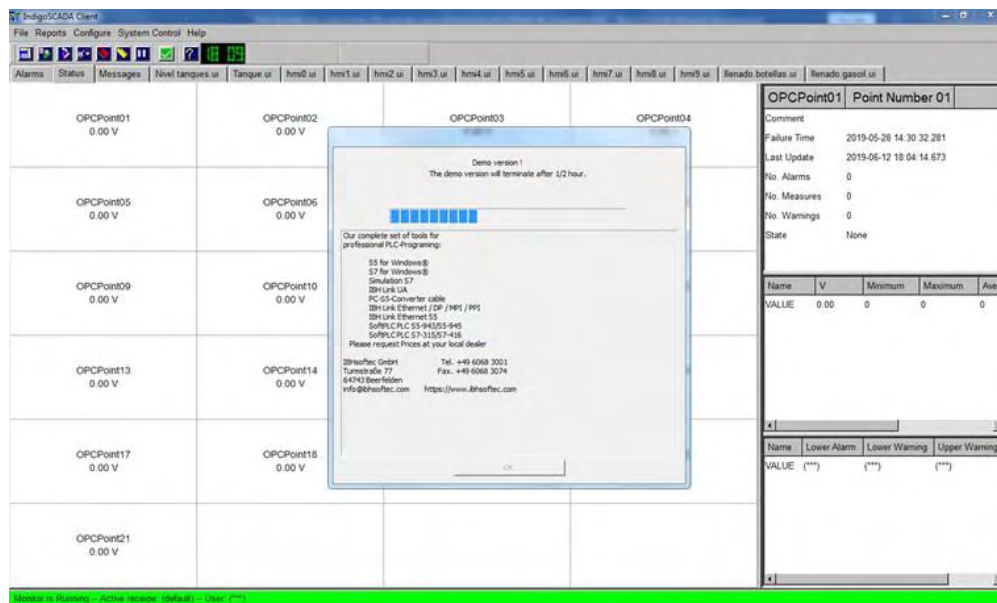


Figura 40. Licencias Demo del servidor IBH OPC.

Los rectángulos que representan los OPCPoints se tornan de color verde cuando se establece una correcta comunicación entre IndigoSCADA y el servidor OPC. En la parte inferior de cada rectángulo se muestra el valor de cada OPCPoint. Como ejemplo, en la figura 41, el OPCPoint01 tiene un valor de “0.00 V”. A partir de este punto se puede crear las HMIs.

Alarms	Status	Messages	llenado.gasoi.ui			
OPCPoint01	OPCPoint02	OPCPoint03	OPCPoint04	OPCPoint05	OPCPoint06	OPCPoint07
0.00 V	0.00 V	6.00 V	1.00 V	0.00 V	1.00 V	0.00 V
OPCPoint09	OPCPoint10	OPCPoint11	OPCPoint12	OPCPoint13	OPCPoint14	OPCPoint15
0.00 V	0.00 V	0.00 V	0.00 V	0.00 V	0.00 V	0.00 V
OPCPoint17	OPCPoint18	OPCPoint19	OPCPoint20	OPCPoint21	OPCPoint22	OPCPoint23
0.00 V	0.00 V	0.00 V	1.00 V	0.00 V	0.00 V	0.00 V

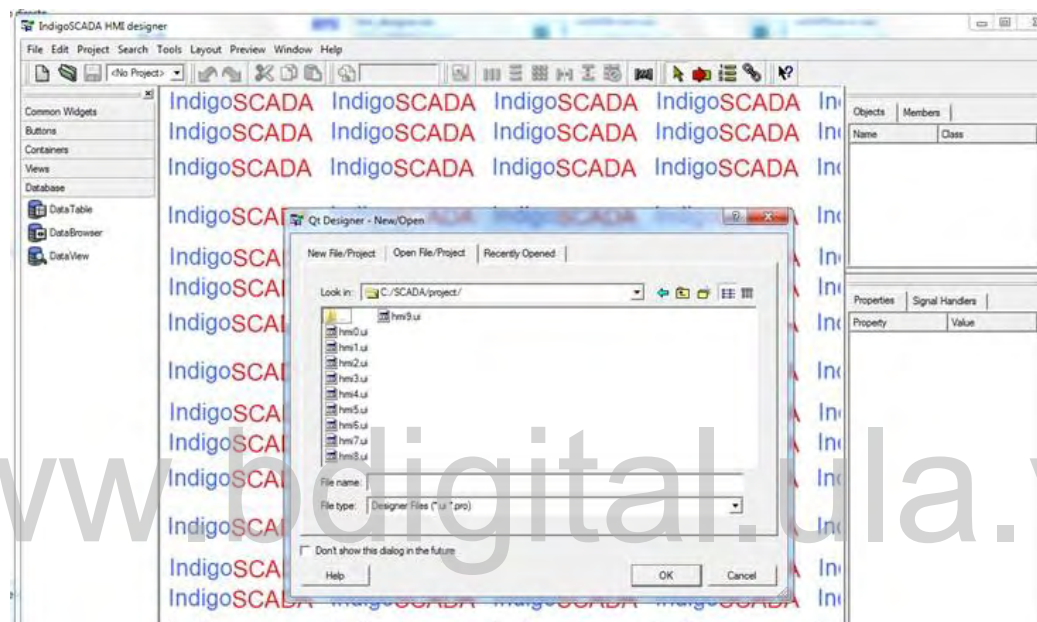
  

OPCPoint01	Point Number 01
Comment	
Failure Time	2019-05-28 14:30:32.281
Last Update	2019-06-12 18:04:14.673
No. Alarms	0
No. Measures	0
No. Warnings	0
State	None
Name	V
Minimum	Maximum
VALUE	0.00 0 0 0 0
Name	Lower Alarm
VALUE	(**)
Name	Lower Warning
VALUE	(**)
Name	Upper Warning
VALUE	(**)

Figura 41. Estado de las variables en OK.

### 3.6 Procedimiento para crear la HMI

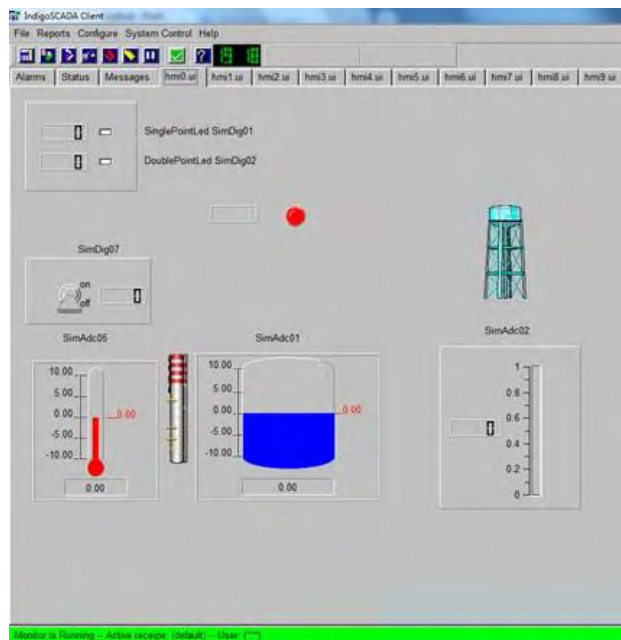
Para crear una hmi se ejecuta “hmi\_designer.exe”. Como primera prueba se abrió el archivo hmi0.ui que tiene por defecto WIDGETS para simulación (figura 42). Agregando un Led y una pantalla LCD. Asociando cada objeto a la misma variable.



**Figura 42. Creación de HMI.**

Se guarda el archivo modificado. En IndigoSCADA client, se presiona File y la opción Exit. El programa se cierra por unos segundos y se vuelve a abrir automáticamente con los cambios hechos en hmi0 (figura 43).

Cuando la variable asociada está en nivel alto, el led se torna rojo y en la pantalla LCD se muestra el número 1. El color del LED es verde si el nivel de la variable es bajo, en este caso en la pantalla LCD se muestra el número 0.



**Figura 43. Creación de HMI.**

Para comprobar la comunicación bidireccional entre IndigoSCADA y el servidor OPC, se realizó otra prueba con envíos de comandos. Por este motivo, se modificó el programa del PLC para activar la salida Q124.0 con una bandera independiente M0.0 (figura 44). Utilizando IndigoSCADA para cambiar el estado de la bandera de bajo a alto, se pudo comprobar que la salida Q124.0 cambió de nivel bajo a alto.



**Figura 44. Programa de prueba para envío de comandos.**



# CAPÍTULO 4

## PRÁCTICAS A IMPLEMENTAR CON EL SISTEMA SCADA

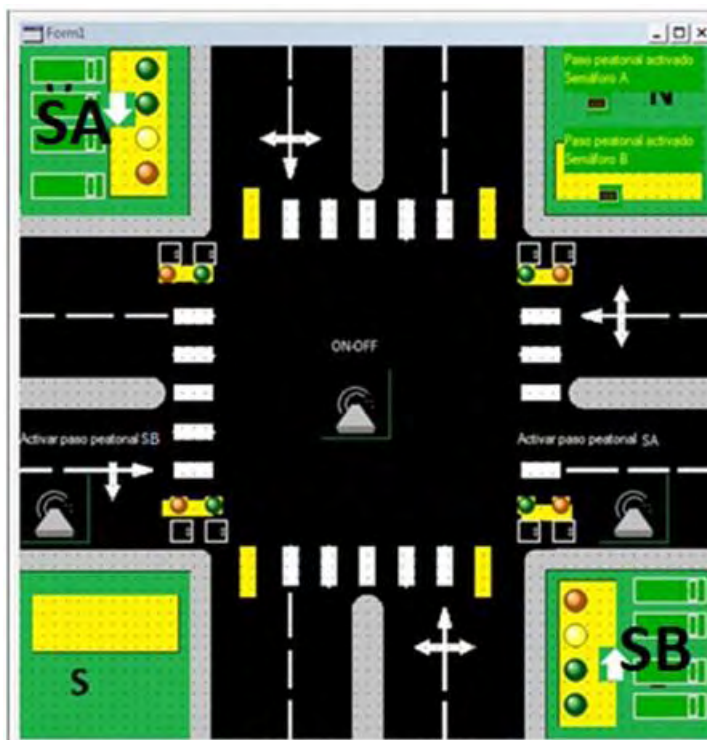
En este capítulo se describe el diseño de cada una de las tres prácticas a implementar. Para explicar el funcionamiento de las mismas se utilizan diagramas de flujo. Se muestran tablas de entradas y salidas (del PLC) que se proponen a utilizar.

### 4.1 SEMÁFORO DE CUATRO INTERCEPCIONES

Esta práctica se basa en la maqueta que se muestra en la figura 45, utilizando los semáforos SM4(A), SM2 (B), las luces peatonales SP2, SP4 y los pulsadores P2, P4.



Figura 45. Maqueta usada en la práctica de semáforos.



**Figura 46. Diseño preliminar de la práctica de semáforos en IndigoSCADA.**

El PLC Siemens S7-300 utilizado para implementar las prácticas dispone de 14 salidas, por lo que no es posible utilizar todos los LEDs mostrados en la figura 45. Por tal motivo se decidió utilizar 4 de las 5 luces LED para cada uno de los semáforos SA y SB (figura 46), además de los 4 LEDs para paso peatonal. En total se utilizan 12 salidas del PLC, cumpliendo con la disponibilidad establecida anteriormente.

Se desea implementar un programa en el PLC Siemens S7-300 el cual permita controlar los LEDs mencionados en el párrafo anterior. Cada semáforo tiene luz roja, amarilla, verde y verde para cruce. El semáforo B simula el control del flujo de vehículos desde el oeste hacia el este y norte (de izquierda a derecha y hacía arriba, desde la perspectiva de la figura 45). El semáforo A controla el flujo desde el este al oeste y sur (de derecha a izquierda y hacia abajo, desde la perspectiva de la figura 45).

Se decidió activar o desactivar todo el sistema de dos formas:

- Utilizando la entrada I124.0 del PLC (tabla 1).

- Por medio del interruptor “ON-OFF” de la interfaz gráfica (46), para efectos de control remoto.

El mismo principio se aplica al paso peatonal (pulsador p1 y p4; interruptores de la interfaz gráfica).

El funcionamiento de los semáforos activados se plantea de la siguiente manera:

- Ambos semáforos están en rojo (RA, RB), durante 10 segundos. Luz de cruce semáforo B (VCB) durante 5 segundos.
- Cuando el semáforo A tenga 5 segundos en rojo (de los 10 en total del punto anterior), A encenderá su luz de cruce (VCA) durante 5 segundos. El semáforo B apaga su luz de cruce.
- Ambos semáforos están en verde (VA, VB), durante 10 segundos.
- Ambos semáforos están en amarillo (AA, AB) durante 3 segundos.
- Las luces rojas peatonales (RPA, RPB) permanecerán activas en caso de que no se active el paso peatonal.
- De activarse el paso peatonal (PA, PB), se encienden las luces verdes peatonales (VPA, VPB) si los semáforos están en rojo.
- La activación del paso peatonal deshabilita la luz de cruce del semáforo correspondiente. Es decir; si se activa el paso peatonal del semáforo A y el sistema esté en rojo, no se enciende la luz de cruce del semáforo A.
- No es posible activar el paso peatonal si el semáforo correspondiente está en luz de cruce.

El diagrama de flujo de la figura 47 explica el proceso.





**Tabla 1. Descripción de entradas del PLC para semáforos.**

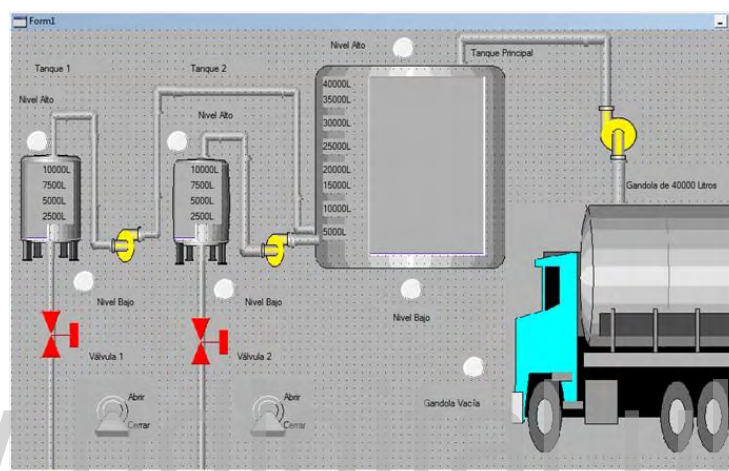
Dirección	Descripción
I124.0	Activar y desactivar los semáforos (ON-OFF)
I124.1	Activar paso peatonal semáforo A (P2)
I124.2	Activar paso peatonal semáforo A (P4)

**Tabla 2. Descripción de salidas del PLC para semáforos.**

Dirección	Descripción
Q124.0	Luz roja semáforo A
Q124.1	Luz roja semáforo B
Q124.2	Luz verde semáforo A
Q124.3	Luz verde semáforo B
Q124.4	Luz amarilla semáforo A
Q124.5	Luz amarilla semáforo B
Q0.0	Luz de cruce semáforo A
Q0.1	Luz de cruce semáforo B
Q0.2	Luz roja peatonal semáforo A
Q0.3	Luz roja peatonal semáforo B
Q0.4	Luz verde peatonal semáforo A
Q0.5	Luz verde peatonal semáforo B

## 4.2 SISTEMA DE LLENADO AUTOMÁTICO DE TANQUES

Consta de una gandola de 40.000 litros que llena de Gasoil a un tanque principal con la misma capacidad a través de una bomba. Este tanque suministra el líquido por medio de bombas a dos tanques de 10.000 litros. Cada uno de los contenedores dispone de indicadores de nivel bajo y alto. Los tanques secundarios tienen válvulas de vaciado.



**Figura 48. Diseño preliminar de llenado de tanques en IndigoSCADA.**

Se desea implementar un sistema que simule el llenado automático de los tanques mostrados en la figura 48. Para la activación del proceso de llenado se decidió usar 3 condiciones. La primera emplea la entrada I124.0 del PLC (tabla 3) para activar y desactivar todo el proceso. Como segunda condición, un sensor que determina si la gandola está conectada al tanque principal (entrada I124.1). Por último, se emplea una bandera en el programa del PLC que establece si la gandola está vacía. Para iniciar el proceso con los tanques vacíos, es necesario resetear los contadores utilizados para indicar el nivel de cada tanque. Para esto se utiliza la entrada I124.4.

El funcionamiento del sistema de llenado es el siguiente:

- Si el tanque principal está vacío y la gandola cisterna no está vacía, se activa la bomba de llenado del tanque principal (tabla 4). Se asume que el tiempo de llenado del tanque es de 16 segundos.

- Si se llena completamente el tanque principal y los tanques secundarios están vacíos, se activan las bombas de llenado de los tanques secundarios. Se asume que cada tanque secundario tarda 8 segundos en llenarse.
- Si se vacía alguno de los tanques secundarios y el tanque principal no está vacío, se activa la bomba correspondiente.
- Por razones de simplicidad se decidió que, al activar alguna válvula de vaciado, el tanque correspondiente se vacía completamente.

Las válvulas de vaciado de cada tanque secundario se activan manualmente desde campo (entradas I124.1 y I124.2), o desde la interfaz gráfica mostrada en la figura 48. El diagrama de flujo de la figura 49 describe el proceso.

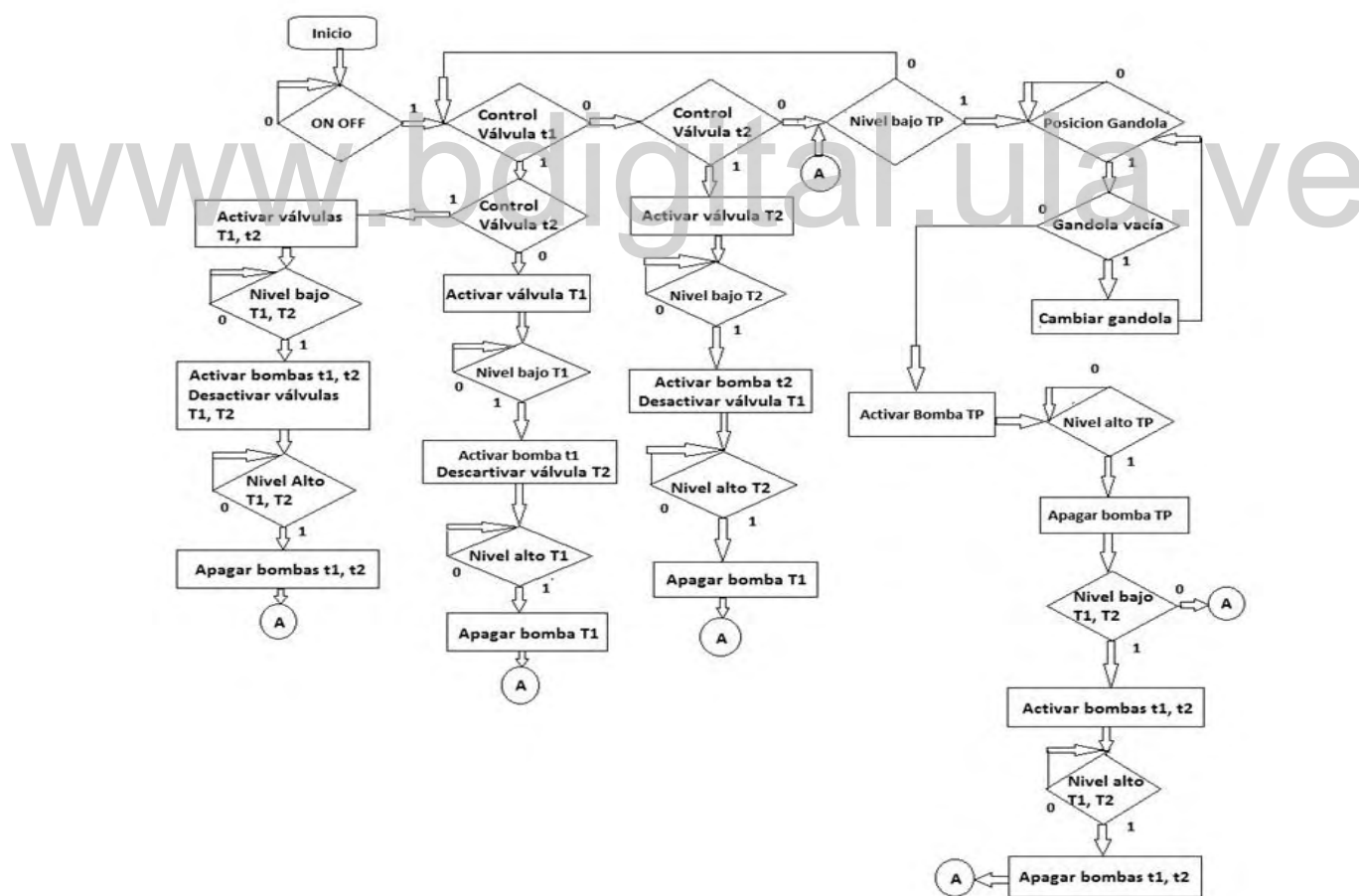


Figura 49. Diagrama de flujo de llenado de tanques. Fuente: Elaboración Propia.

**Tabla 3. Descripción de entradas del PLC para llenado de tanques.**

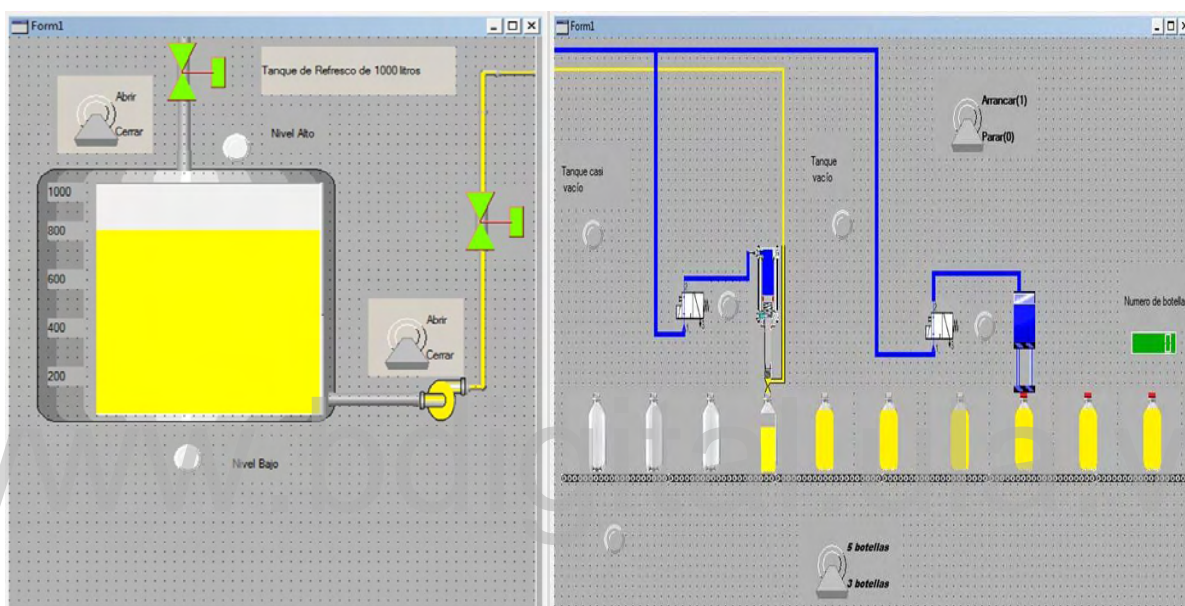
Dirección	Descripción
I124.0	Activar y desactivar llenado
I124.1	Posición de gandola
I124.2	Válvula T1
I124.3	Válvula T2
I124.4	Resetear contadores (tanques vacíos)

**Tabla 4. Descripción de salidas del PLC para llenado de tanques.**

Dirección	Descripción
Q124.0	Bomba tanque principal
Q124.1	Bomba tanque 1
Q124.2	Bomba tanque 2
Q124.3	Válvula tanque 1
Q124.4	Válvula tanque 2
Q124.5	Nivel bajo tanque principal
Q0.0	Nivel alto tanque principal
Q0.1	Nivel bajo tanque 1
Q0.2	Nivel alto tanque 1
Q0.3	Nivel bajo tanque 2
Q0.4	Nivel alto tanque 2

### 4.3 SISTEMA AUTOMÁTICO DE LLENADO Y TAPADO DE BOTELLAS

Consiste de una cinta transportadora donde se llenan y tapan botellas de refresco. La cinta transportadora es impulsada por un motor eléctrico. El refresco se almacena en un tanque y es impulsado por una bomba hasta el pistón de llenado de botellas. El tanque cuenta con indicadores de nivel alto y bajo.



**Figura 50. Diseño preliminar de llenado de botellas en IndigoSCADA**

Se desea implementar un sistema que sincronice el llenado y tapado de botellas en una misma cinta transportador y que además controle la frecuencia con la que se llenan y tapan las botellas. El sistema debe contar con la opción de activarse desde la HMI (figura 50), o activando la entrada I124.0 (tabla 5) del PLC. Es posible llevar la cuenta de botellas tapadas, por lo que se plantea utilizar la entrada 124.3 para resetear la cuenta.

El funcionamiento del sistema se asume de la siguiente manera:

- Si el nivel del tanque no es bajo, se puede activar el llenado y tapado de botellas.
- Al activarse el proceso, se activa la válvula de tres vías que extiende el pistón de llenado. El llenado de botellas dura 4 segundos.
- El tapado de botellas se inicia al mismo tiempo del llenado.

- Si el número de botellas máximo seleccionado en la cinta es de 3, el motor de la cinta (salida Q124.0 de la tabla 6) se activa durante 4 segundos y se coloca una botella vacía al comienzo de la cinta
- Si el número de botellas máximo seleccionado es de 5, el motor de la cinta transportadora se activará durante 2 segundos y se coloca una botella vacía al comienzo.
- Si el tanque se está vaciando, se puede llenar manualmente activando la válvula de llenado (desde la HMI, o activando I0.0).
- Si el tanque se vacía, se detiene todo el proceso y sólo se puede llenar activando la válvula de llenado manualmente.

En la figura 51 se observa un diagrama de flujo donde se explica el procedimiento anterior.

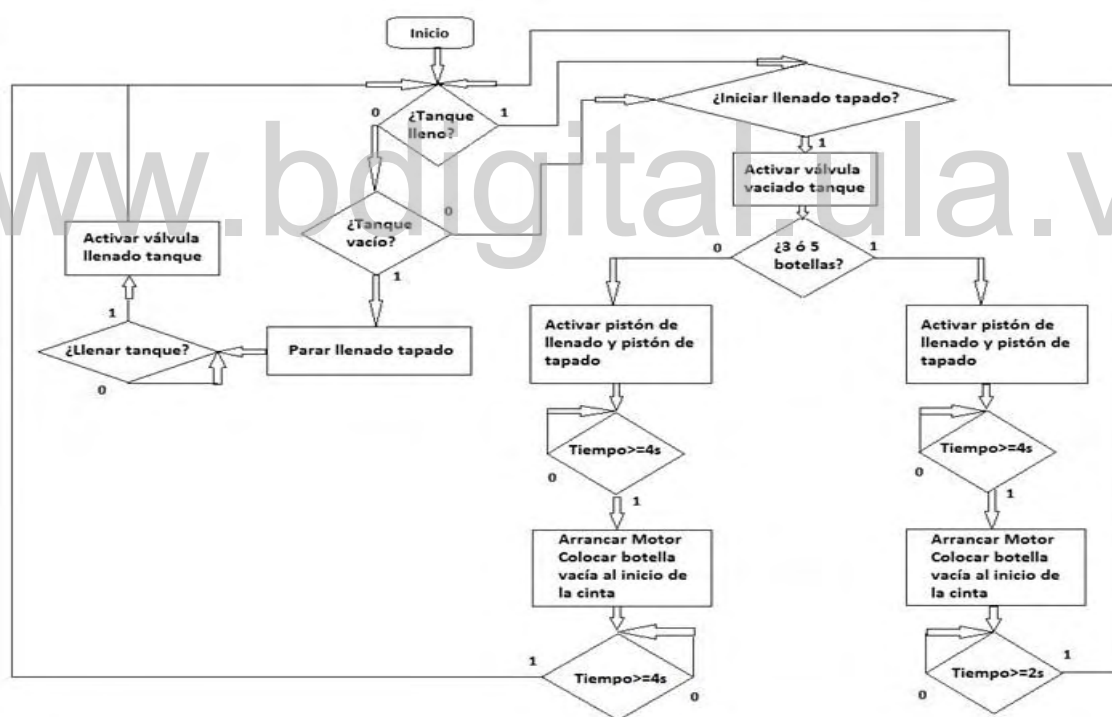


Figura 51. Diagrama de flujo llenado de botellas.

**Tabla 5. Descripción de entradas del PLC para llenado de botellas.**

Dirección	Descripción
I124.0	Iniciar proceso
I124.3	Reseteo de cuenta de botellas tapadas
I0.0	Llenado manual de tanque

**Tabla 6. Descripción de salidas del PLC para llenado de botellas.**

Dirección	Descripción
Q124.0	Motor de cinta transportadora
Q124.1	Nivel bajo tanque
Q124.2	Nivel alto tanque
Q124.3	Válvula de pistón de llenado
Q124.4	Válvula de pistón tapado
Q124.5	Válvula de llenado tanque
Q0.0	Válvula de vaciado tanque



# CAPÍTULO 5

## RESULTADOS

Para cada una de las prácticas se muestran capturas de pantallas ejecutándose en IndigoSCADA. Al final del capítulo se expondrán las conclusiones y recomendaciones finales del trabajo.

### 5.1 PRÁCTICA 1

La figura 52 muestra los semáforos desactivados.

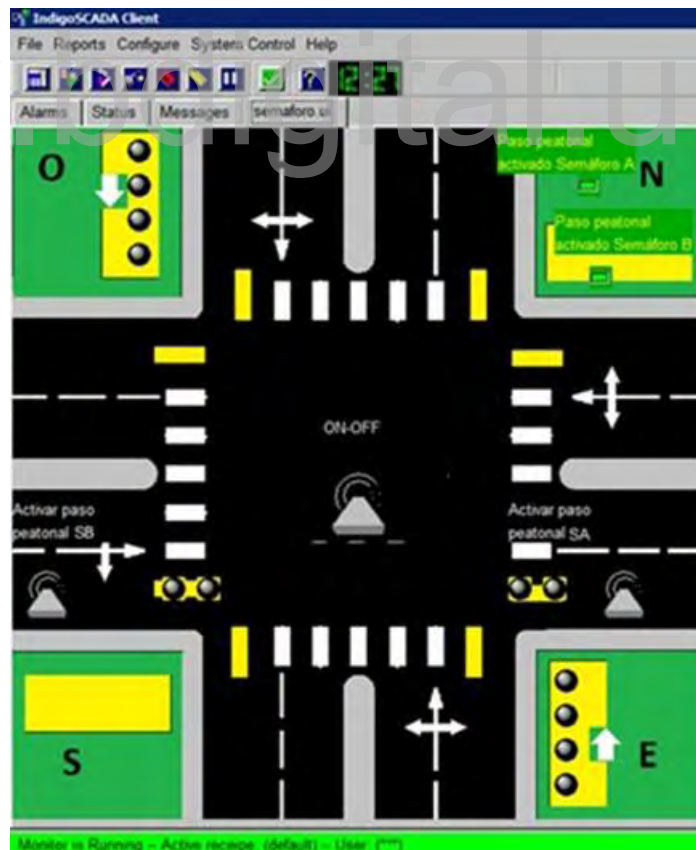


Figura 52. Semáforos desactivados. Fuente: Propia

Enviando el comando a la bandera asociada al “switch” ON-OFF (figuras 53).

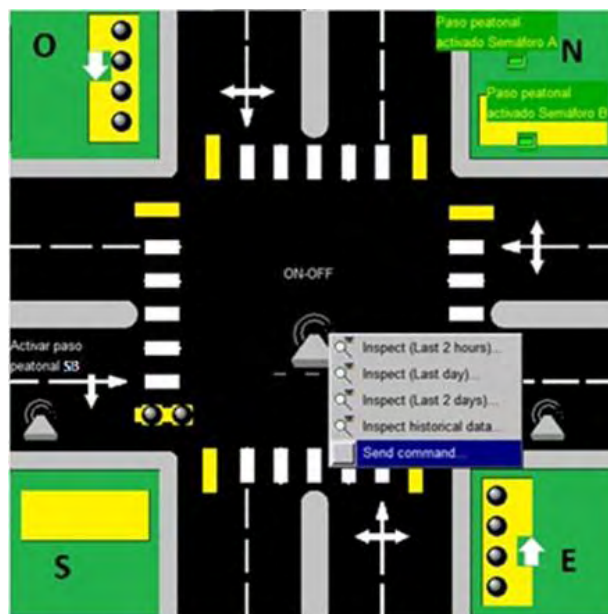


Figura 53. Envío de Comando.

En las figuras 54, 55 y 56 se muestran los semáforos en rojo, verde y amarillo respectivamente.

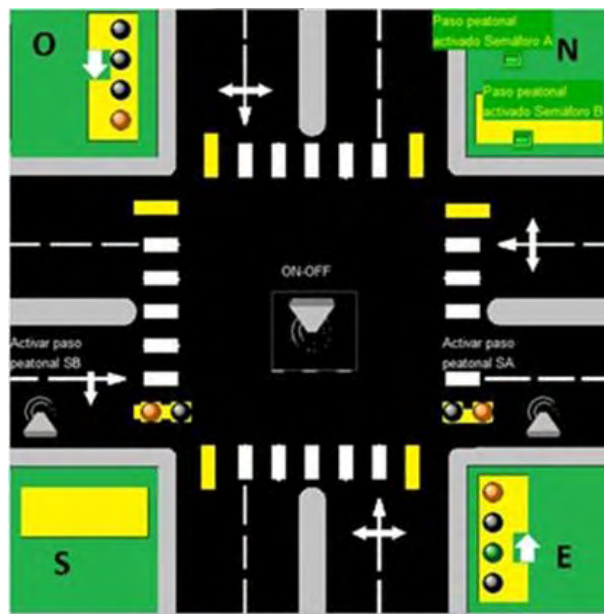


Figura 54. Semáforos en Rojo y semáforo B con luz de cruce.

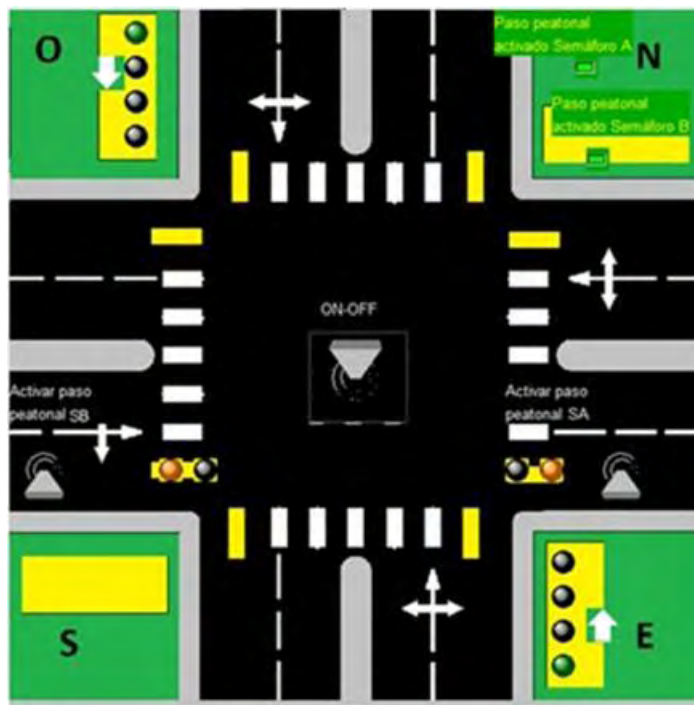


Figura 55. Ambos semáforos en verde.

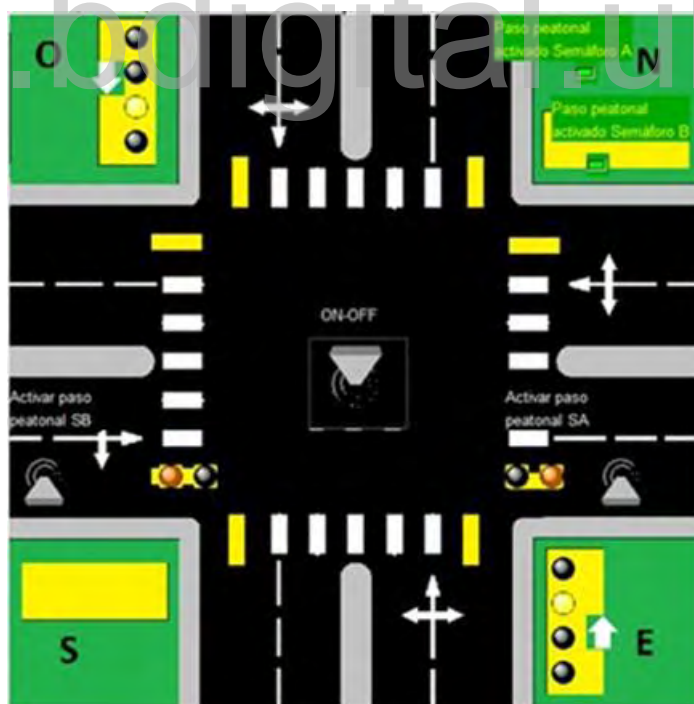


Figura 56. Semáforos en Amarillo.

IndigoSCADA permite la generación de gráficas en tiempo real con Pen trace (figura 57).

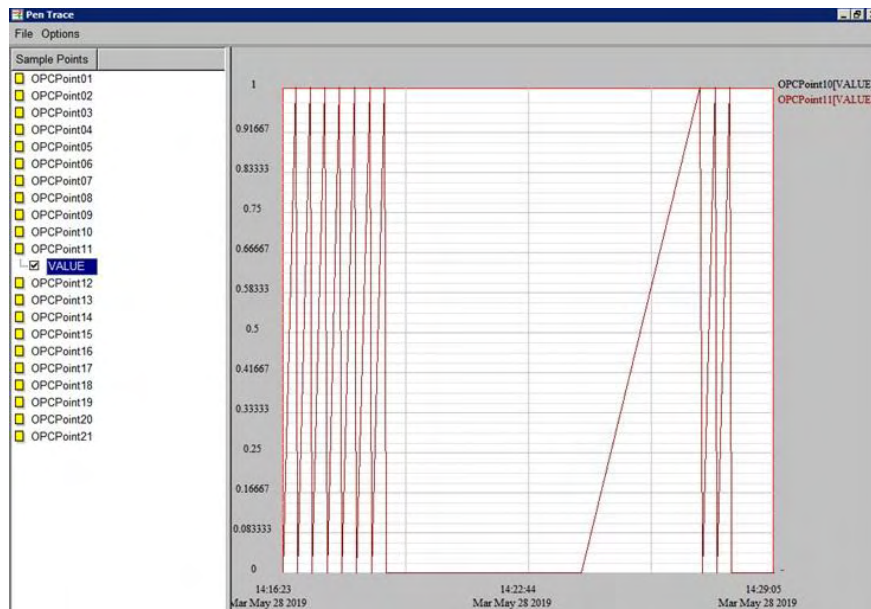


Figura 57. Gráfica en tiempo real.

Es posible acceder a la base de datos de cada una de las variables directamente de la interfaz principal del programa (figura 58).

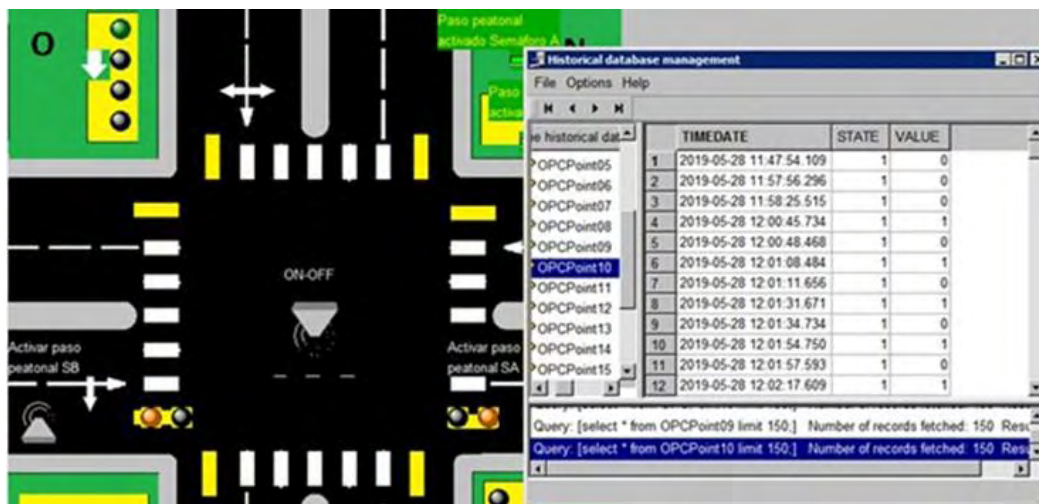
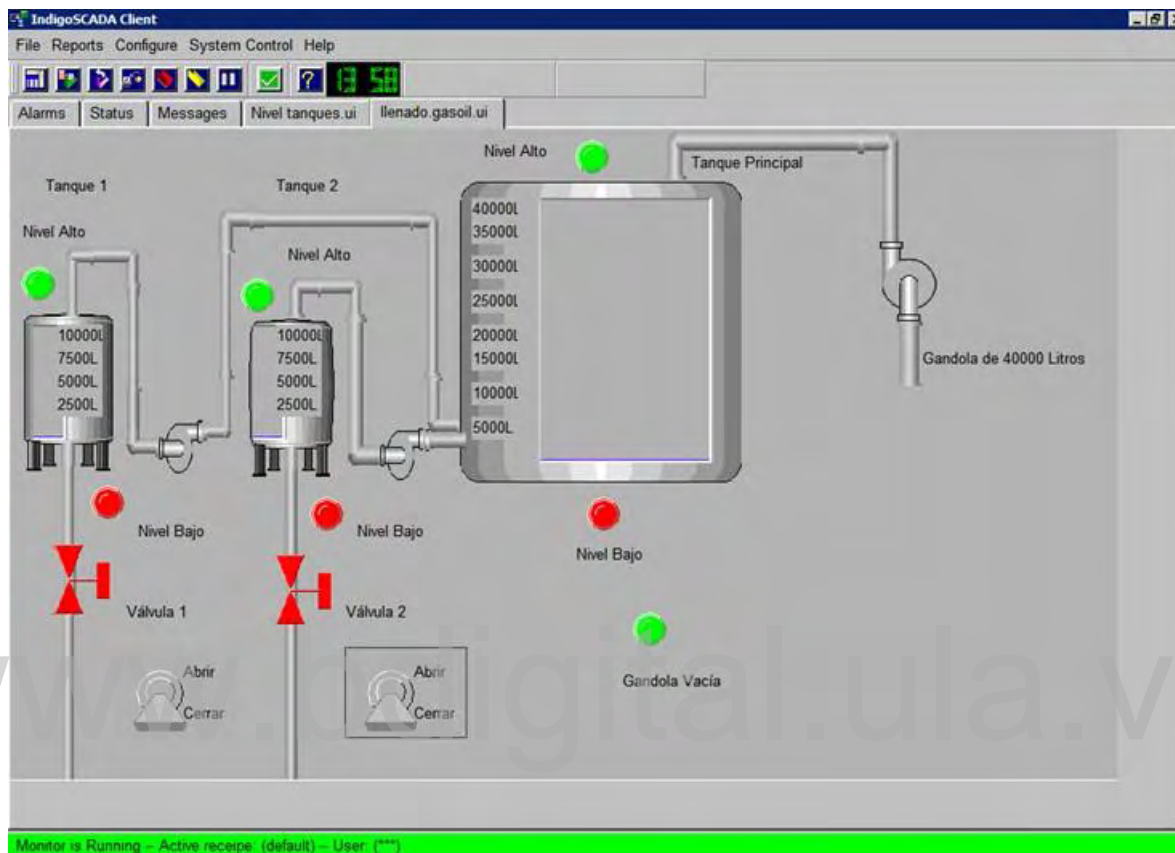


Figura 58. Base de datos históricos.

## 5.2 PRÁCTICA 2

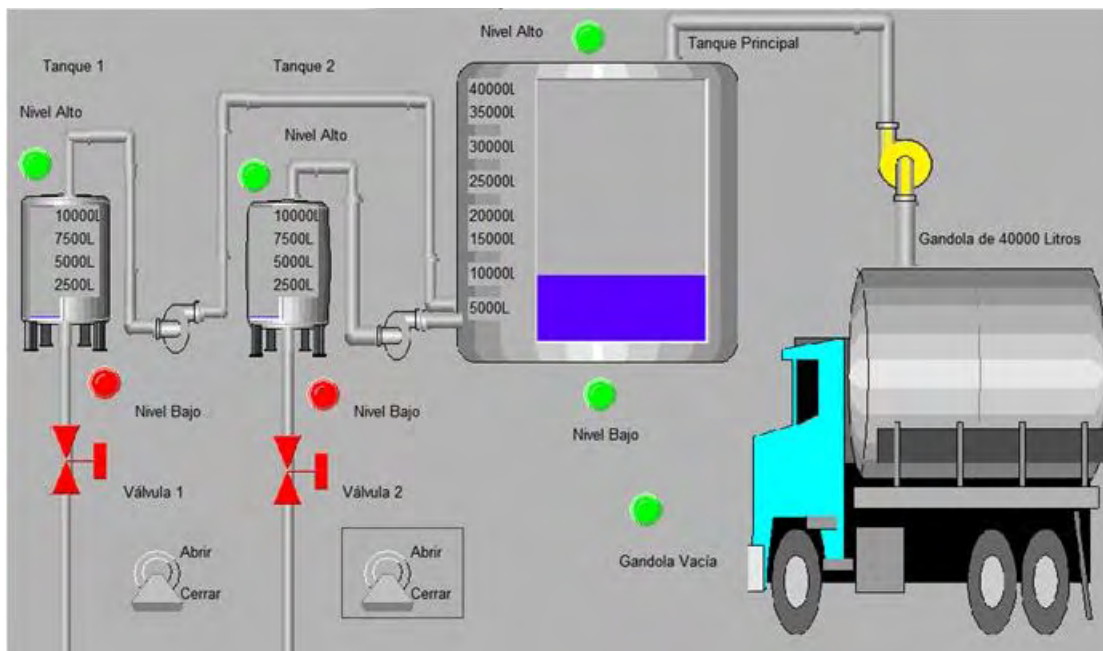
En la figura 59 se muestra los tanques vacíos y la gandola desconectada.



**Figura 59. Tanques vacíos y gandola desconectada.**

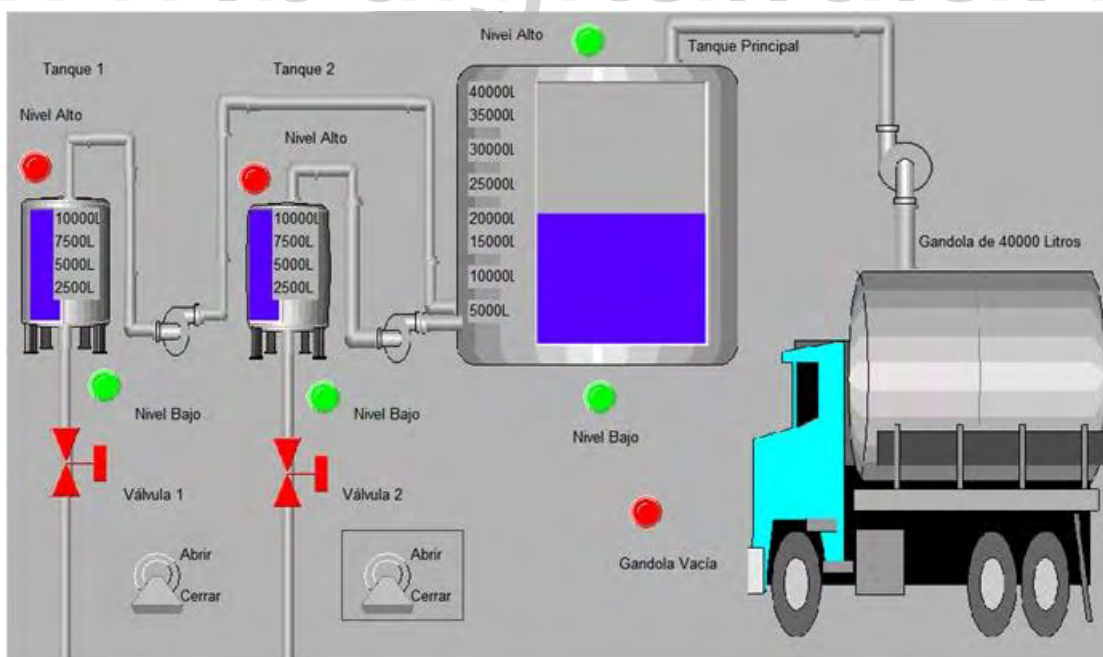
Conectando la entrada “ON-OFF” con dirección I124.0 y el sensor de conexión de la gandola (dirección I124.1), se activa la bomba que llena el tanque principal.





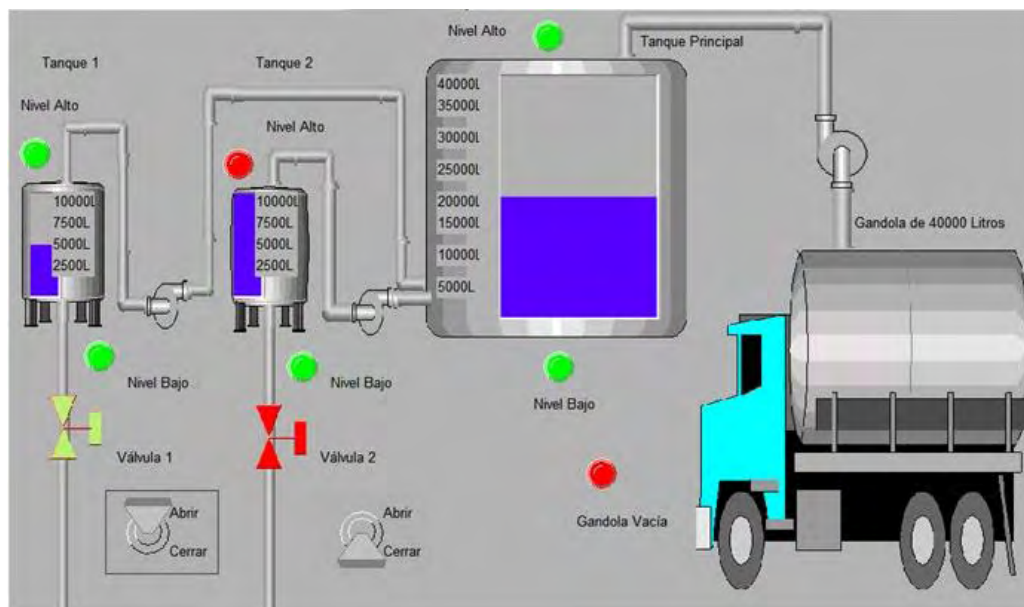
**Figura 60. Llenando tanque principal.**

Al llenarse completamente el tanque principal, se activan las bombas a los tanques 1 y 2. La figura 61 muestra los tanques llenos y el tanque principal vacío hasta la mitad.



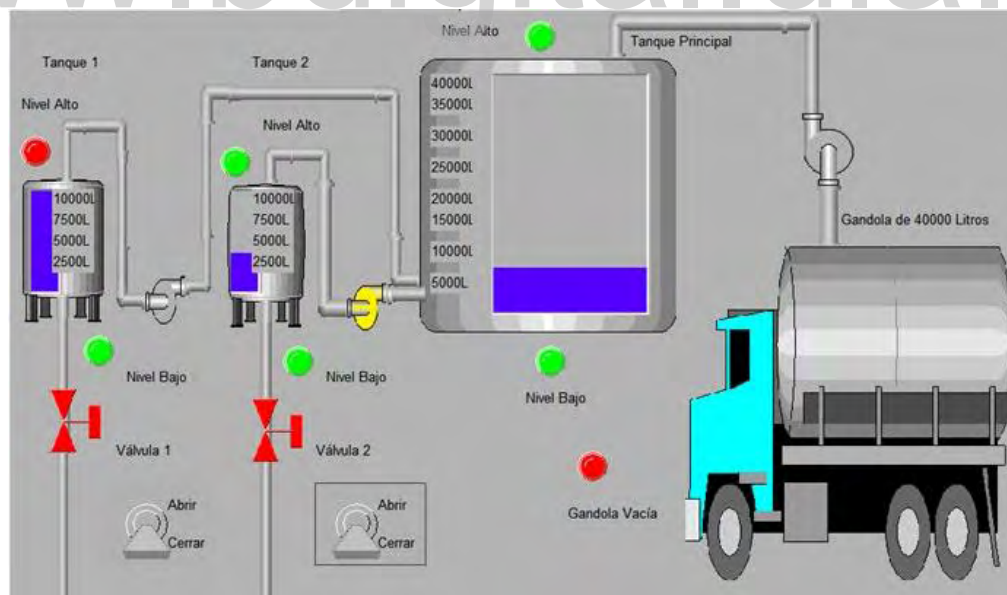
**Figura 61. Tanques 1 y 2 llenos, tanque principal vacío hasta la mitad.**

Vaciando el Tanque 1 después de enviar el comando a la válvula 1 (figura 62).



**Figura 62. Vaciando tanque 1.**

www.bdigital.ula.ve



**Figura 63. Llenado tanque 2.**

La figura 64 muestra una gráfica en tiempo real del nivel del tanque principal.

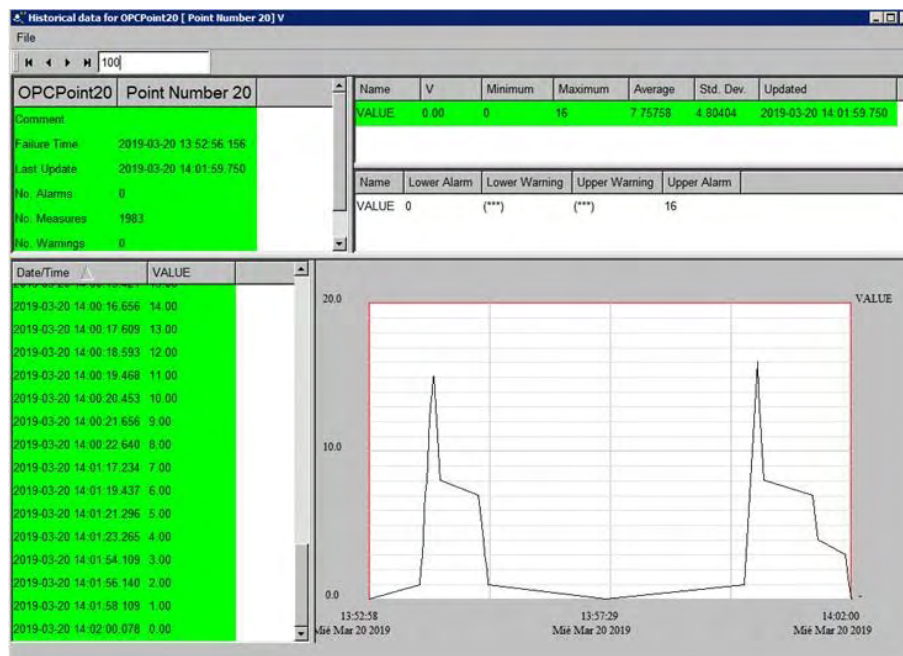


Figura 64. Gráfica en tiempo real del nivel del tanque principal.

Creando reporte con los valores de todos los tanques (figura 65).

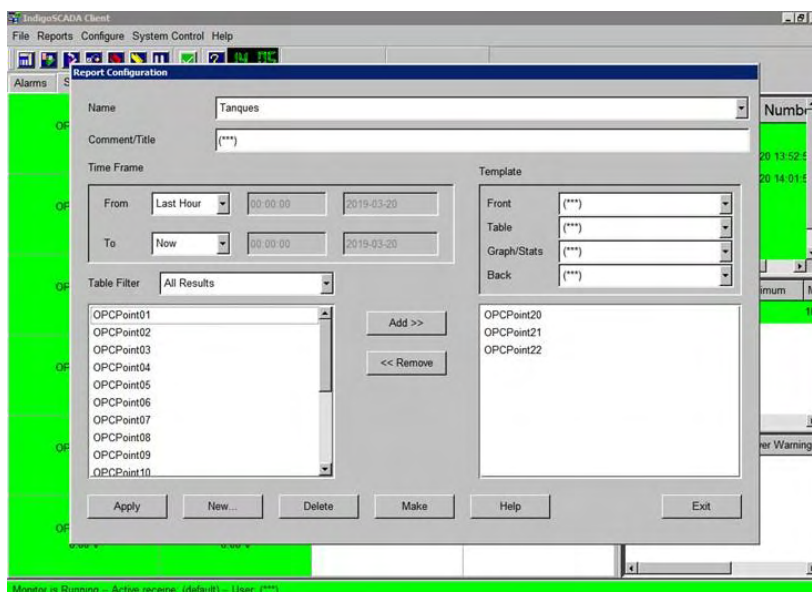


Figura 65. Creando reporte para una de las variables.



Los reportes se guardan en la carpeta “report” en la carpeta “scada” en el disco local (figura 66).

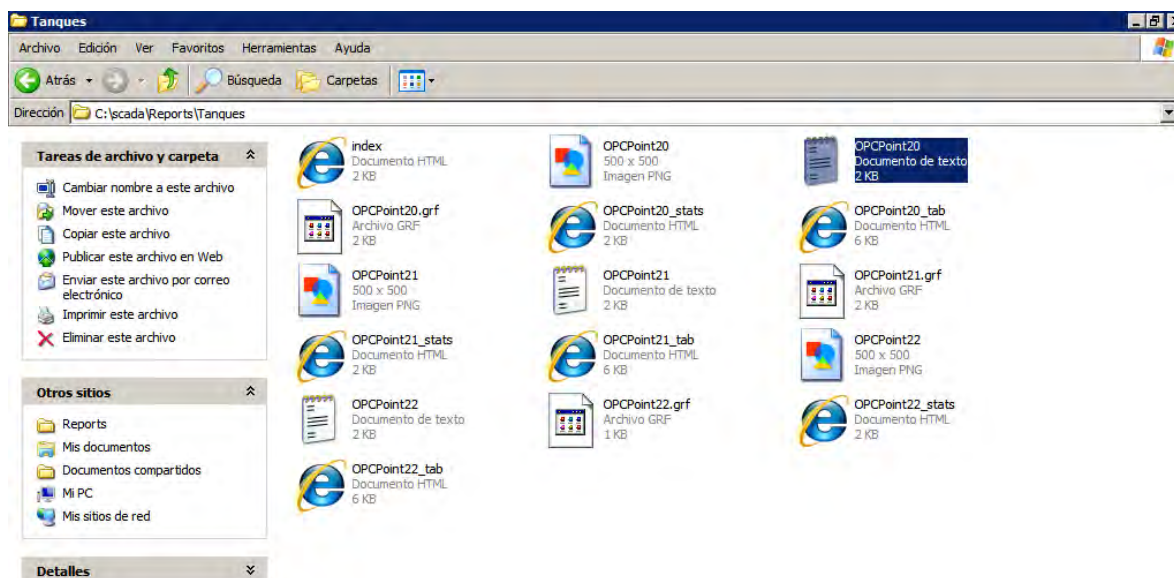


Figura 66. Carpeta de archivos con los reportes.

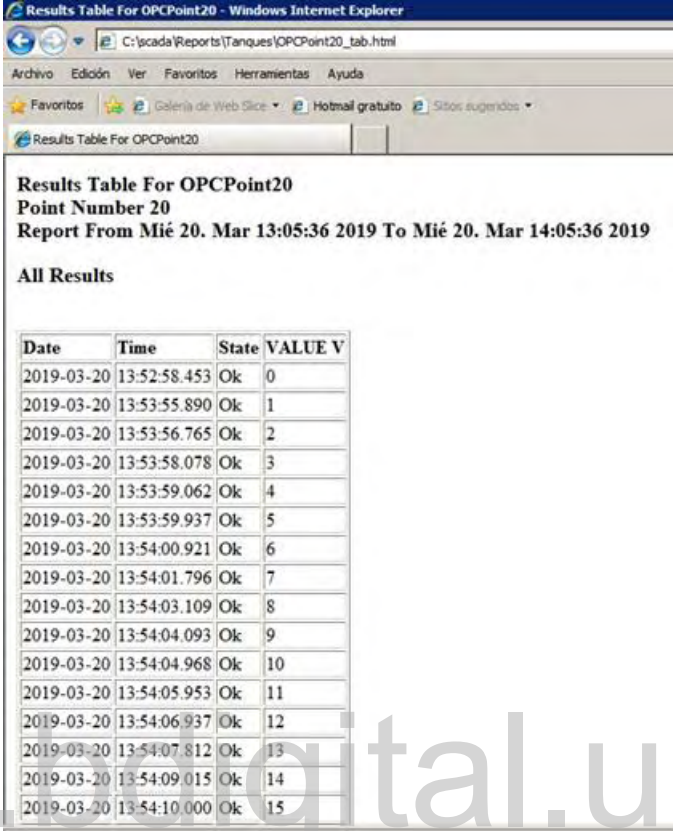
www.bdigital.ula.ve

En la figura 67 se observa los reportes de 3 variables en formato HTML.



Figura 67. Reporte con tres variables.

La tabla de valores del tanque principal se muestra en la figura 68.



The screenshot shows a web browser window titled "Results Table For OPCPoint20 - Windows Internet Explorer". The address bar displays the file path "C:\pccada\Reports\Tanques\OPCPoint20\_tab.html". The browser interface includes a menu bar with "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayuda". Below the menu bar, there are search and utility icons. The main content area displays the following text:

**Results Table For OPCPoint20**  
**Point Number 20**  
**Report From Mié 20. Mar 13:05:36 2019 To Mié 20. Mar 14:05:36 2019**

**All Results**

Date	Time	State	VALUE V
2019-03-20	13:52:58.453	Ok	0
2019-03-20	13:53:55.890	Ok	1
2019-03-20	13:53:56.765	Ok	2
2019-03-20	13:53:58.078	Ok	3
2019-03-20	13:53:59.062	Ok	4
2019-03-20	13:53:59.937	Ok	5
2019-03-20	13:54:00.921	Ok	6
2019-03-20	13:54:01.796	Ok	7
2019-03-20	13:54:03.109	Ok	8
2019-03-20	13:54:04.093	Ok	9
2019-03-20	13:54:04.968	Ok	10
2019-03-20	13:54:05.953	Ok	11
2019-03-20	13:54:06.937	Ok	12
2019-03-20	13:54:07.812	Ok	13
2019-03-20	13:54:09.015	Ok	14
2019-03-20	13:54:10.000	Ok	15

Figura 68. Tabla de valores de la variable OPCPoint20.

### 5.3 PRÁCTICA 3

Al estar vacío el tanque de refresco, se detiene todo el proceso de llenado y tapado (figura 69 y 70).

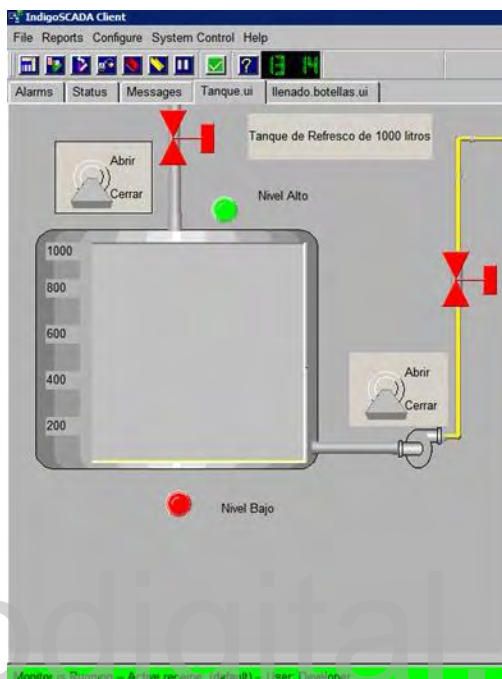


Figura 69. Tanque vacío.

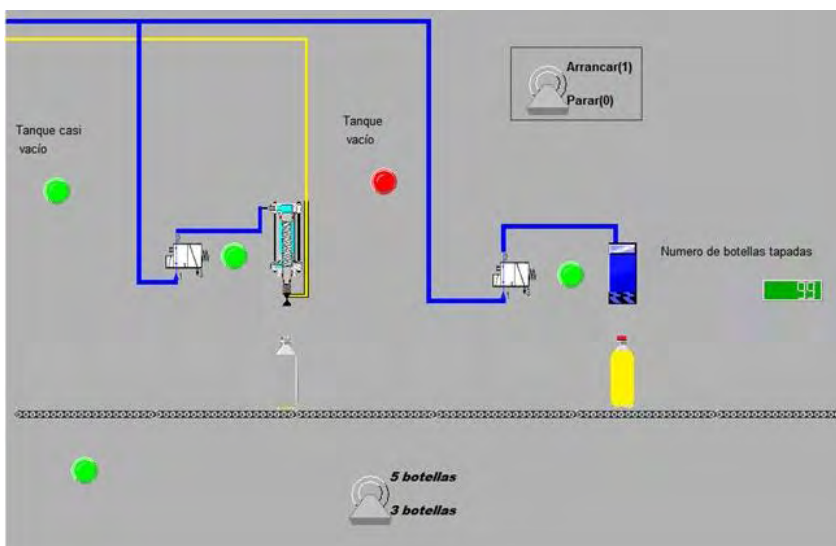


Figura 70. Proceso de llenado y tapado detenido.

Activando la válvula, llenando el tanque (figura 71). Tanque lleno (figura 72).

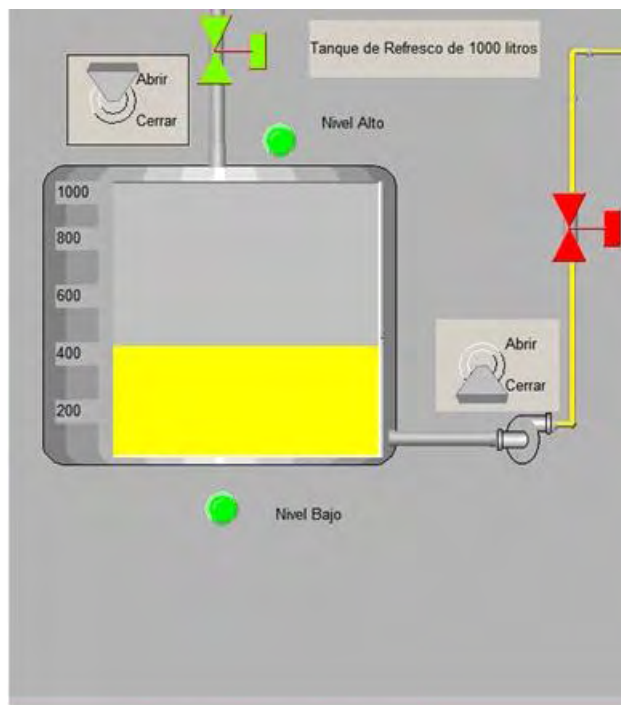


Figura 71. Llenado tanque.

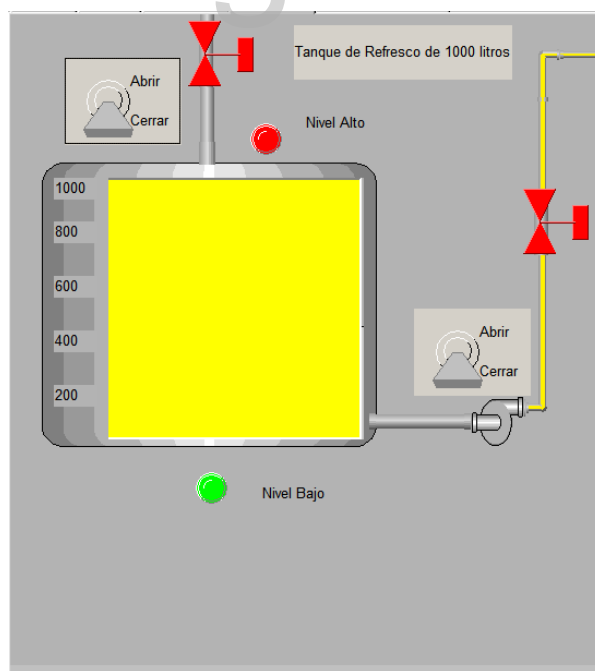


Figura 72. Tanque lleno.

Llenado botella, escogido 3 botellas máximo en la cinta transportadora (figura 73 y 74).

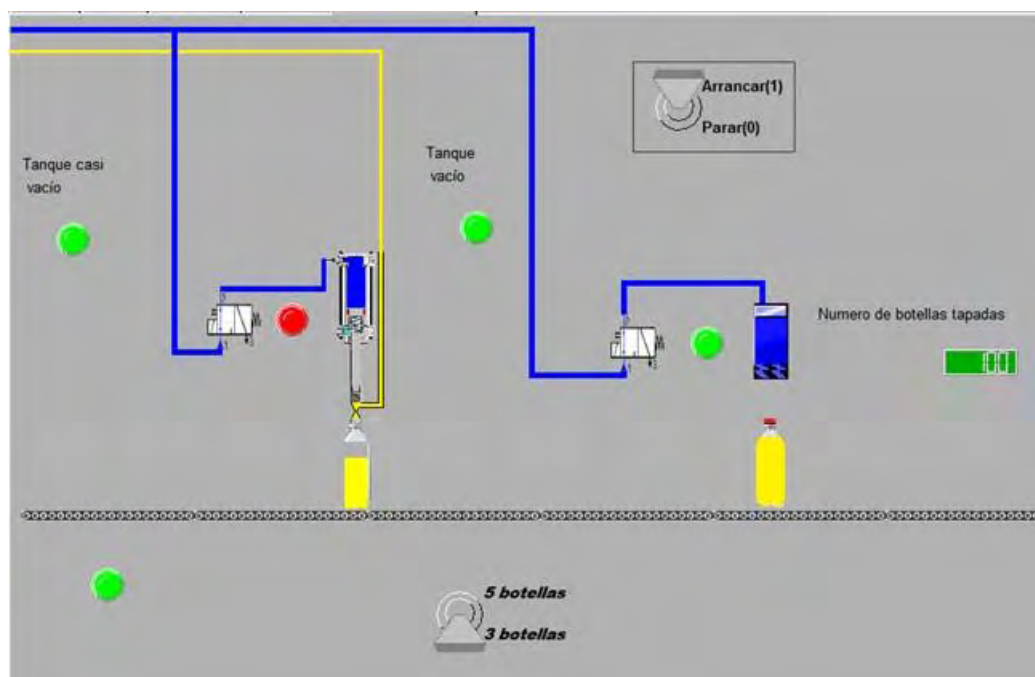


Figura 73. Llenando y tapando.

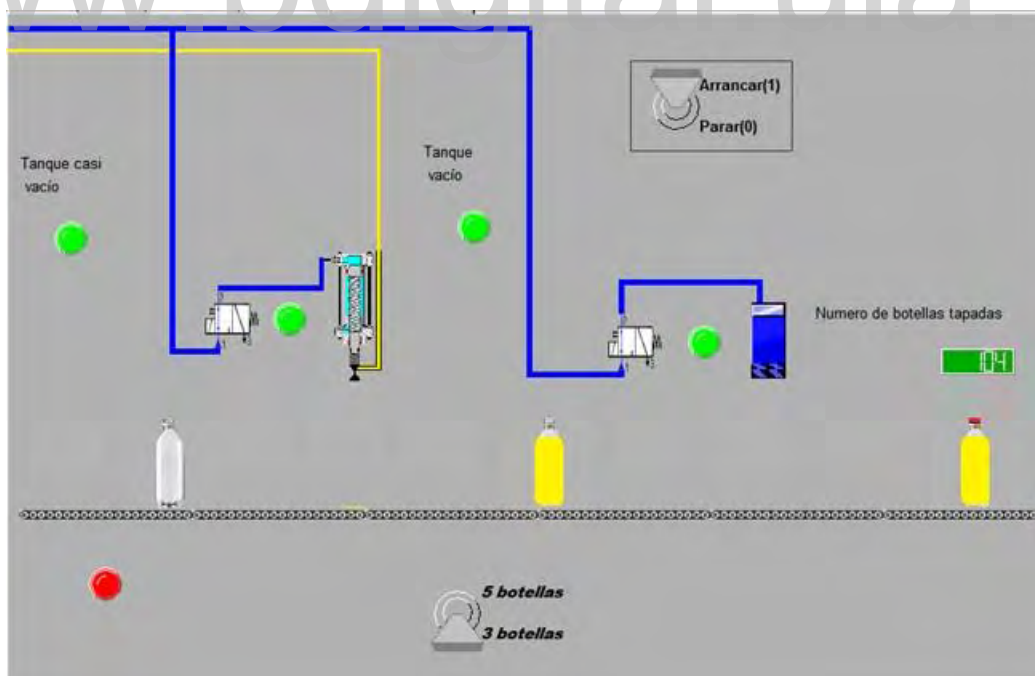


Figura 74. Motor de cinta encendido.

Enviando el comando 1 a la bandera asociada al swicht se cambia el número de botellas a 5 (figura 75).

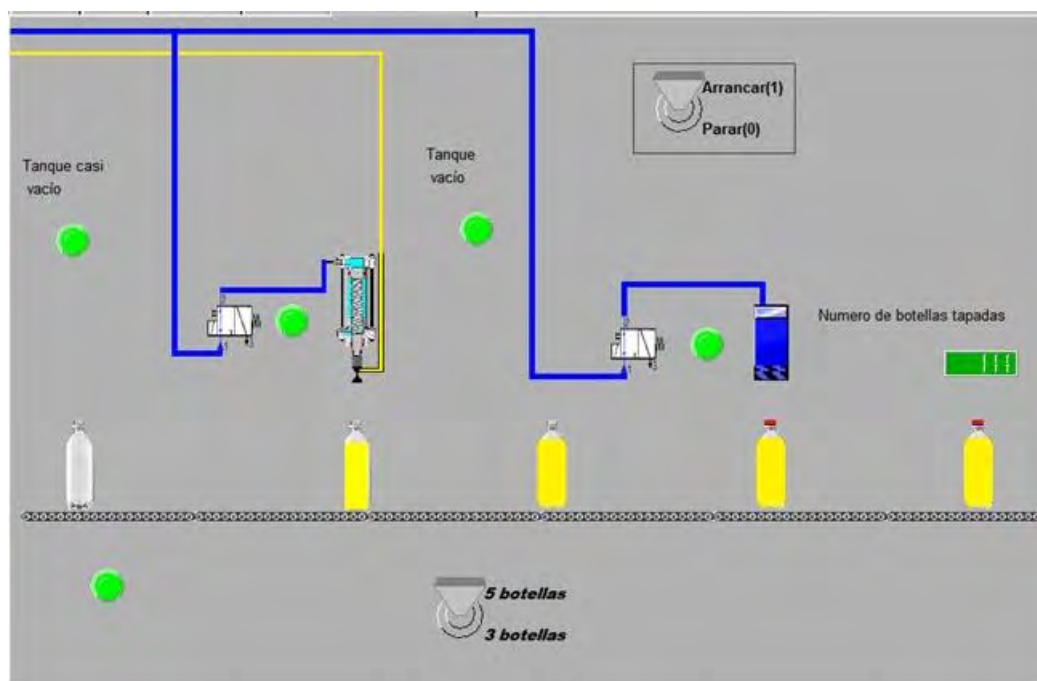


Figura 75. 5 botellas en la cinta.

## CONCLUSIONES

- Se hizo un estudio del PLC Siemens S7-300 disponible en el laboratorio de control de la escuela de ingeniería eléctrica el cual viene solo con módulos de entradas y salidas digitales y un adaptador MPI/USB.
- Del análisis de los sistemas SCADA y de los equipos disponibles en el laboratorio no se logró encontrar algún programa de software libre para implementar el sistema SCADA, que fuera compatible directamente con el puerto MPI/USB.
- Debido al tipo de protocolo del PLC y a los programas SCADA disponibles, se decidió utilizar un servidor y cliente OPC para la comunicación entre el PLC y el sistema SCADA.
- Se realizaron pruebas de comunicación entre el PLC, servidor OPC e IndigoSCADA, logrando satisfactoriamente el envío y recepción de datos entre todos los programas.
- Se realizó el diseño de tres (3) prácticas demostrativas para su uso en el Laboratorio de control de la escuela de Ingeniería Eléctrica. Estas se diseñaron para funcionar con el PLC Siemens S7-300 utilizando el programa STEP 7 Lite.
- Se implementaron exitosamente las prácticas en el sistema SCADA realizando la HMI, registro de alarmas y base de datos en IndigoSCADA para cada una de ellas. Durante el desarrollo del trabajo de grado se constató que las prácticas diseñadas funcionan correctamente.

## RECOMENDACIONES

- Para la práctica del semáforo, se puede completar el número de salidas requeridas para la maqueta, utilizando dos PLC Siemens S7-300, o añadiendo un módulo de salidas y entradas digitales a uno de los PLC. También es posible simular las salidas faltantes utilizando banderas en el programa del PLC.
- Utilizar módulos de entradas y salida analógicas del PLC S7-300 en las prácticas 2 y 3 para la supervisión, control y adquisición de datos, de los niveles de tanques.
- Configurar el sistema SCADA para utilizar el servidor y cliente OPC en diferentes computadoras.

www.bdigital.ula.ve



## REFERENCIAS

- [1] I. Hurtado, J. Toro *“Paradigmas y métodos de investigación en tiempos de cambio”*. Caracas: Colección Minerva. Los libros del nacional. Venezuela. 2007.
- [2] M. Barrios, *“Trabajos de Grado de Especialización y Maestría y Tesis Doctorales”*. Caracas: FEDUPEL. Venezuela. 2011.
- [3] M. Angarita, *“Programación del PLC modelo S7-300 con STEP 7 Lite”*. Trabajo fin de carrera. Ingeniería Eléctrica. Universidad de Los Andes. Mérida. 2009.
- [4] J. Calderón, *“Control y monitoreo SCADA de un proceso experimental, utilizando PLC Siemens S7-300 y software LABVIEW”*. Trabajo fin de carrera en ingeniería Eléctrica. Universidad Nacional Autónoma de México. México DF. 2009. [En línea] Disponible en: <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/1529/Tesis%20Completa.pdf?sequence=1>
- [5] M. Hernández, *“Desarrollo de un Sistema SCADA para la medición de voltajes con sistemas embebidos para el laboratorio de mecatrónica de la facultad de mecánica”*. Tesis de pregrado en ingeniería de mantenimiento. Escuela Superior Politécnica de Chimborazo. Riobamba Ecuador. 2010. [En línea] Disponible en: <http://dspace.espoch.edu.ec/bitstream/123456789/1137/1/25T00140.pdf>
- [6] L. Malavé, *“Diseño de un sistema de supervisión adquisición y control para las estaciones de radares meteorológicos del Instituto Nacional de Meteorología e Hidrología (INAMEH)”*. Tesis de pregrado en ingeniería Eléctrica. Universidad Central de Venezuela. Caracas. 2012. [En línea] Disponible en: <http://saber.ucv.ve/bitstream/123456789/4281/1/TRABAJO%20ESPECIAL%20DE%20GRADO%20LUIS%20D%20MALAVE%20R%20final%20pdf.pdf>
- [7] ¿Qué es la automatización industrial? (sf). *EXSOL* (sf) [En línea]. Disponible en: <http://www.exsol.com.ar/automatizacion-industrial/>
- [8] C. Lagos, “Protocolo de comunicación industrial”. *Electroindustrias*. Septiembre 2006. [En línea] Disponible en: <http://www.emb.cl/electroindustria/articulo.mvc?xid=562> [Accedido: 27-Ene-2019]
- [9] Lenguaje de programación. (sf). *ingmecafenix*(sf) [En línea]. Disponible en: <https://www.ingmecafenix.com/automatizacion/lenguajes-programacion-plc/> [Accedido: 27-Ene-2019]
- [10] diagrama de escalera plc. (sf). *unicrom*(sf) [En línea]. Disponible en: <https://unicrom.com/diagrama-de-escalera-plc/> [Accedido: 27-Ene-2019]
- [11] O. Blanco, *“Propuesta de un Esquema de Automatización integrado para procesos de producción por lotes”*. Tesis M.Sc. en Automatización e Instrumentación. Universidad de Los Andes. Mérida. 2013.

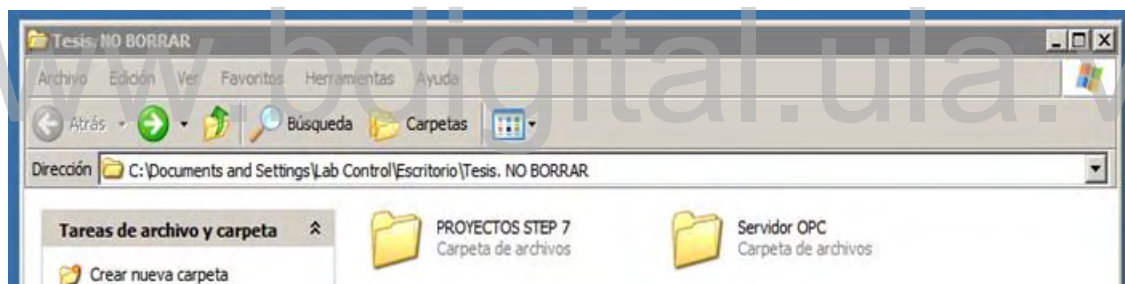
- [12] Filosofía. (sf). *GNU* (sf) [En línea]. Disponible en:  
<https://www.gnu.org/philosophy/free-sw.es.html>  
[Accedido: 27-Ene-2019]
- [13] OPC servidor. (sf). *MatrikonOPC*(sf) [En línea]. Disponible:  
<https://www.matrikonopc.es/opc-servidor/index.aspx>  
[Accedido: 28-Ene-2019]
- [14] User manual. (2018). *IndigoSCADA*(revision 308) [En línea]. Disponible en:  
[http://www.enscada.com/a7khg9/IndigoSCADA\\_user\\_manual.pdf](http://www.enscada.com/a7khg9/IndigoSCADA_user_manual.pdf)  
[Accedido: 29-Ene-2019]

www.bdigital.ula.ve

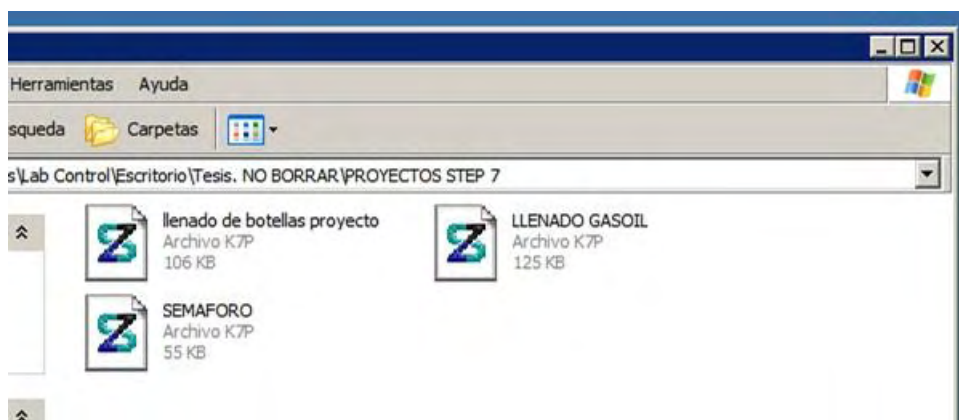
## ANEXOS

A continuación, se describen una serie de pasos para realizar cualquiera de las 3 prácticas

1. Paso. Conectar el adaptador MPI/USB del PLC Siemens S7-300 al puerto USB de la computadora. Encender el PLC y colocarlo en modo STOP. Comprobar que los 3 LEDs del adaptador MPI/USB se enciendan. Cargar el programa al PLC; Dirigirse a la carpeta “Tesis. NO BORRAR” ubicada en el escritorio de Windows. Abrir la carpeta “Proyectos STEP 7” y dar click en cualquiera de los 3 archivos con el nombre de la práctica que se desee probar. Cargar el programa en STEP 7 Lite.



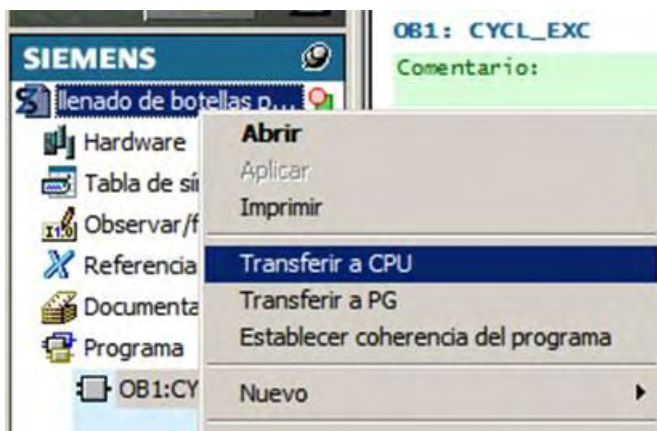
**Ilustración 1.**



**Ilustración 2.**



**Ilustración 3.**

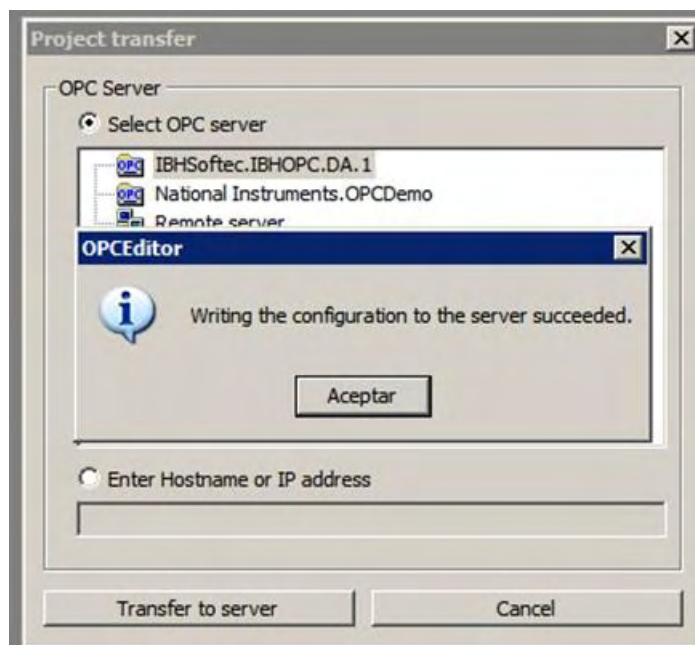


**Ilustración 4.**



**Ilustración 5.**

2. Paso. Transferir datos al servidor OPC; en la carpeta “Tesis. NO BORRAR” abrir el archivo con el nombre de la práctica que se cargó al PLC. Dar click en “Transfer to OPC server”. Las figuras 21, 22 y 23 del capítulo 3 muestran este paso.



**Ilustración 6.**

En caso de mostrarse un mensaje diferente al de la ilustración 6, es necesario reiniciar la computadora y transferir al servidor OPC hasta que aparezca el mensaje mostrado en la ilustración 6.

3. Paso. Ir a C:\scada\project, abrir la carpeta con el nombre de la práctica que se cargó al PLC. Copiar la carpeta "Project" y pegarla en C:\scada\project. Dar "copiar y reemplazar todo".



**Ilustración 7.**

4. En la dirección C:\scada\bin, ejecutar "manager.exe" y elegir "Start IndigoSCADA". Las figuras 30 y 31 del capítulo 3 explican este paso.

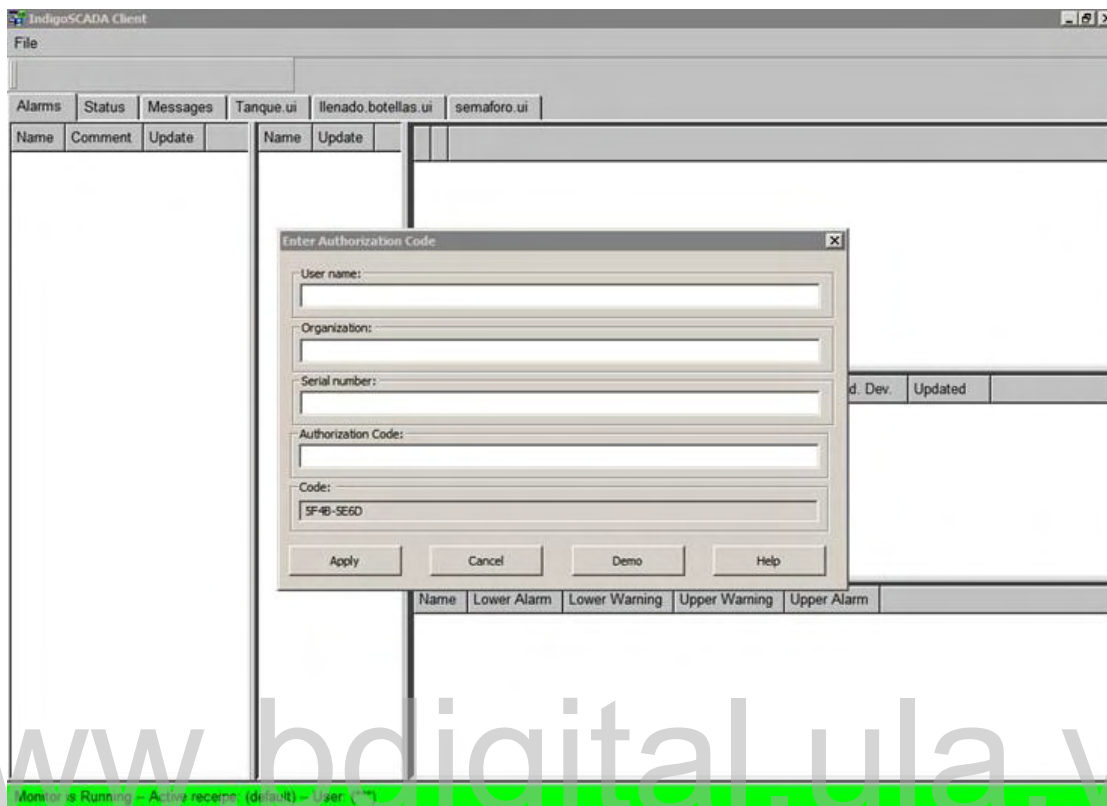


Ilustración 8.

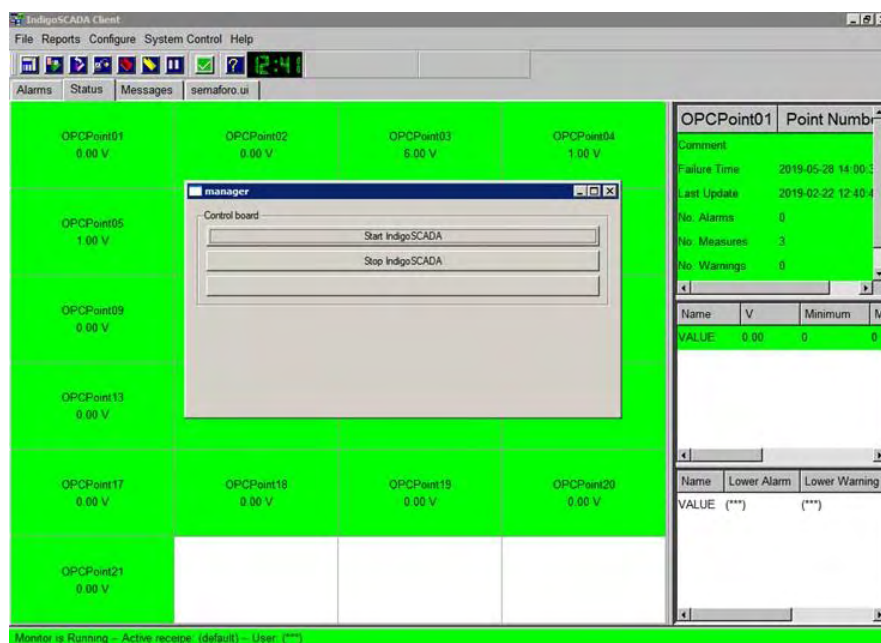


Ilustración 9.

Dar click en OK, seleccionar la pestaña Status. Esperar que salga la pantalla del servidor OPC IBH y escoger licencia DEMO (ilustración 8). Esperar que los recuadros estén de color verde (ilustración 9). Escoger la pestaña con el nombre de la hmi a utilizar. Establecer el PLC en modo RUN. La práctica esta lista para utilizarse. En el capítulo 4 se describe el funcionamiento de cada práctica.

En caso de escoger la práctica del semáforo es posible conectar el PLC a la maqueta. En la siguiente tabla se describe la conexión de los terminales de la maqueta con las correspondientes entradas y salidas del PLC. Es necesario conectar el terminal positivo (terminal rojo) y negativo (terminal negro) del PLC, con los terminales designados V+ y GND respectivamente en la maqueta.

**Tabla 1. Entradas y Salidas del PLC.**

Entrada PLC	Salida PLC	Símbolo en la maqueta
I124.1		P2
	Q124.0	R semáforo A
I124.2		P4
	Q124.1	R semáforo B
	Q124.2	V semáforo A
	Q124.3	V semáforo B
	Q124.4	A semáforo A
	Q124.5	A semáforo B
	Q0.0	LC semáforo A
	Q0.1	LC semáforo B
	Q0.2	PRSA
	Q0.3	PRSB
	Q0.4	PVSA
	Q0.5	PVSB