



# **UNIVERSIDAD NACIONAL DE LOJA**

## **ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES**

### **CARRERA DE TECNOLOGÍA EN ELECTRICIDAD Y CONTROL INDUSTRIAL**

#### **TEMA:**

**“AUTOMATIZACIÓN DE UN PROTOTIPO DE SISTEMA  
SONORO CON MINI-PLC PARA EL CAMBIO DE  
JORNADA EN EL AEIRNNR”**

**INFORME TÉCNICO PREVIO A OPTAR POR EL  
TÍTULO DE TECNÓLOGO EN ELECTRICIDAD Y  
CONTROL INDUSTRIAL**

#### **AUTOR**

*Stalin Javier Ortega Serrano*

#### **DIRECTOR**

*Ing. Julio César Cuenca Tinitana, Mg. Sc.*

**LOJA - ECUADOR**

**2015**

# CERTIFICACIÓN

Ing. Julio César Cuenca Tinitana, Mg. Sc.

**DOCENTE DEL ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES DE LA UNIVERSIDAD NACIONAL DE LOJA; Y DIRECTOR DEL INFORME TÉCNICO.**

## **CERTIFICA:**

Que el trabajo de investigación titulado *“AUTOMATIZACIÓN DE UN PROTOTIPO DE SISTEMA SONORO CON MINI-PLC PARA EL CAMBIO DE JORNADA EN EL AEIRNNR”*, desarrollado por el señor **Stalin Javier Ortega Serrano**, previo a optar el Título de *Tecnólogo en Electricidad y Control Industrial*, ha sido realizado bajo mi dirección, mismo que cumple con los requisitos exigidos en las normas de graduación, por lo que autorizo su presentación ante el tribunal.

Loja, Febrero del 2015



Ing. Julio César Cuenca Tinitana, Mg. Sc.  
**DIRECTOR DEL INFÓRME TÉCNICO**

# AUTORÍA

Yo **STALIN JAVIER ORTEGA SERRANO**, declaro ser autor del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales, por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio Institucional-Biblioteca Virtual.

**Autor:** STALIN JAVIER ORTEGA SERRANO

**Firma:**



**Cédula:** 1105001216

**Fecha:** 20/04/2015

**CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL, Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.**

Yo **STALIN JAVIER ORTEGA SERRANO**, declaro ser autor de la tesis titulada: **AUTOMATIZACIÓN DE UN PROTOTIPO DE SISTEMA SONORO CON MINI-PLC PARA EL CAMBIO DE JORNADA EN EL AEIRNNR**, como requisito para optar al grado de: **TECNÓLOGO EN ELECTRICIDAD Y CONTROL INDUSTRIAL**; autorizo al sistema bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI; en las redes de información del país y del exterior, con las cuales tenga convenido la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los veinte días del mes de abril del dos mil quince, firma el autor.



**Firma:**

**Autor:** Stalin Javier Ortega Serrano

**Cedula:** 1105001216

**Dirección:** Saraguro, Juan Antonio Montesinos y Reina de Quito **Teléfono:** 072200427

**Correo:** stalino\_sos@hotmail.es

**Celular:** 0997126146

**DATOS COMPLEMENTARIOS**

**Director de Tesis:** Ing. Julio César Cuenca Tinitana, Mg. Sc.

**Tribunal de grado:** Ing. Edwin Bladimir Paccha Herrera, Mg. Sc.

Ing. Byron Agustín Solórzano Castillo, Mg. Sc.

Ing. Juan Gabriel Ochoa Aldeán, Mg. Sc.

## **AGRADECIMIENTO**

A Dios, por estar conmigo en cada paso de mi vida, fortaleciendo mi corazón y espíritu, y por haber puesto en mi camino a aquellas personas que constantemente me acompañado y apoyado durante todo el periodo de estudio.

A mi familia que se constituyen en el pilar fundamental de mi vida personal y profesional, apoyándome en mis estudios de forma irrestricta y continua.

**STALIN JAVIER ORTEGA**

## **DEDICATORIA**

A Dios por haberme dado la salud, la fuerza mental, la fuerza espiritual y la capacidad intelectual para lograr mis objetivos y la culminación de este proyecto.

A mis Padres, Familiares y amigos que creyeron, me apoyaron incondicionalmente para hacer de esta meta una realidad.

**STALIN JAVIER ORTEGA**

## **RESUMEN**

Un controlador lógico programable (PLC), se emplea en procesos industriales que necesitan ser controlados a través de procesos secuenciales en tiempo real para automatizar una o varias necesidades, como: Espacio reducido, Procesos de producción periódicamente cambiantes, Procesos secuenciales, Maquinaria de procesos variables entre otros.

En el presente trabajo práctico se diseñó e implementó un módulo de prácticas con un PLC Siemens 230RC, que permite desarrollar y poner a prueba sus múltiples funcionalidades en procesos que son susceptibles de automatizar; además, se desarrolló con fines educativos y de prueba la automatización de un prototipo de sistema sonoro para indicar el cambio de hora en el AEIRNNR, detallando cada etapa que interviene.

El sistema sonoro se implementó físicamente en el módulo de prácticas, utilizando una programación manual asíncrona que utiliza 2 entradas (para manual y automático) y 1 salidas del PLC con una sirena de 110V-AC interconectadas entre sí, de tal forma que permite que el proceso sea cíclico. Permitiéndonos así cumplir cada objetivo planteado, conforme se detalla a continuación:

### **Objetivo General:**

- Diseñar un prototipo para un sistema sonoro con un Mini-PLC para simular el cambio de horario en cada jornada del AEIRNNR.

### **Objetivos Específicos:**

- Diseñar un esquema en Logo-Soft e implementar en un Mini-PLC para determinar el cambio de jornada en el AEIRNNR.
- Construir un prototipo en Hardware que permita simular un sistema sonoro con un Mini-PLC para el AEIRNNR.
- Implementar el prototipo de sistema sonoro en el laboratorio de electrónica del AEIRNNR, para que complementar los conocimientos en el funcionamiento de un PLC a través de la práctica y manipulación del mismo.



## SUMMARY

A logical programmable controller (PLC), uses in industrial processes that need to be controlled across sequential real time processes to automate one or several needs, as: limited Space, Processes of production from time to time sequential bankers, Process, Machinery of variable processes between others.

In the present practical work a module of practices was designed and implemented by a PLC Siemens 230RC, that allows to develop and to test his multiple functionalities in processes that are capable of automating; in addition, it developed with educational ends and of it proves the automation of a prototype of sonorous system to indicate the change of hour in the AEIRNNR, detailing every stage that intervenes.

The sonorous system was implemented physically in the module of practices, using a manual asynchronous programming that uses 2 income (for manual and automatic) and 1 gone out of the PLC with a siren of 110V-AC interconnected between yes, in such a way that it is allowed that the process should be cyclical. Permitting this way to fulfill every raised, similar aim it is detailed later:

### **General aim:**

- To design a prototype for a sonorous system with a Mini-PLC to simulate the change of schedule in every day of the AEIRNNR.

**Specific aims:**

- To design a scheme in Logo-Soft and to help in a Mini-PLC to determine the change of day in the AEIRNNR.
- To construct a prototype in Hardware that allows to simulate a sonorous system with a Mini-PLC for the AEIRNNR.
- To implement the prototype of sonorous system in the laboratory of electronics of the AEIRNNR, in order that to complement the knowledge in the functioning of a PLC across the practice and manipulation of the same one.

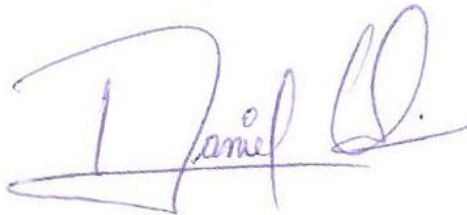
## CERTIFICACIÓN SUMMARY

Lic. Daniel Fernando Ocampo Vásquez  
**DOCENTE DE INGLÉS EN UTPL.**

### **CERTIFICA:**

Que la traducción al idioma inglés del resumen del trabajo de investigación titulado ***“AUTOMATIZACIÓN DE UN PROTOTIPO DE SISTEMA SONORO CON MINI-PLC PARA EL CAMBIO DE JORNADA EN EL AEIRNNR”***, desarrollado por el señor **Stalin Javier Ortega Serrano**, ha sido realizado y revisado bajo mi dirección, mismo que cumple con las reglas y normas gramaticales exigidas por el idioma.

Loja, Abril del 2015



Lic. Daniel Fernando Ocampo Vásquez  
**DOCENTE DE INGLÉS EN UTPL**

# ÍNDICE GENERAL

CERTIFICACIÓN .....	ii
AUTORÍA.....	iii
CARTA DE AUTORIZACIÓN.....	iv
AGRADECIMIENTO.....	v
DEDICATORIA .....	vi
RESUMEN.....	vii
SUMMARY .....	ix
CERTIFICACIÓN SUMMARY .....	xi
ÍNDICE GENERAL.....	xii
A. INTRODUCCIÓN .....	2
B. DESCRIPCIÓN TÉCNICA .....	4
B.1. SISTEMAS DE CONTROL AUTOMÁTICO.....	4
B.1.1. Componentes fundamentales de los sistemas de control .....	7
B.1.1.1. Captadores (sensores o transductores) .....	8
B.1.1.2. Actuadores.....	10
B.1.1.3. Unidades de control.....	11
B.1.1.4. Acondicionamiento de señales .....	12
B.1.2. Control de procesos por ordenador .....	13
B.2. CONTROLADOR LÓGICO PROGRAMABLE PLC .....	17

B.2.1.	Desarrollos de los PLC .....	19
B.2.2.	Definición y principios de operación.....	21
B.2.3.	Funcionamiento del PLC .....	22
B.2.3.1.	Otras funciones adicionales del PLC.....	23
B.2.4.	Arquitectura Básica de los PLC.....	25
B.2.4.1.	Unidad Central de Proceso – CPU .....	26
B.2.4.2.	Buses: .....	27
B.2.4.3.	Memoria .....	28
B.2.5.	Clasificación de PLC .....	29
B.2.5.1.	PLC tipo Nano:.....	29
B.2.5.2.	PLC tipo Compactos: .....	29
B.2.5.3.	PLC tipo Modular: .....	30
B.2.6.	Lenguajes de programación de PLC.....	30
B.2.6.1.	Programa y lenguaje de programación .....	30
B.2.6.2.	Programas de aplicación y del sistema.....	31
B.2.6.3.	Tipos de lenguajes de programación de PLC .....	32
B.2.6.4.	La norma IEC 1131-3.....	32
B.2.6.5.	Lenguaje ladder .....	33
B.2.6.6.	Lenguaje booleano (Lista de Instrucciones).....	36
B.2.6.7.	Diagrama de funciones (FBD) .....	37
B.2.6.8.	Lenguaje de texto estructurado (ST) .....	38

B.2.6.9.	Sequential function chart (SFC) .....	39
B.3.	LOGO SOFT .....	40
B.3.1.	Hardware y software necesarios .....	41
B.3.2.	Tipos de equipos .....	42
B.3.3.	Estructura de LOGO .....	43
B.3.4.	Identificación de LOGO .....	44
B.3.5.	Variantes de LOGO .....	45
B.3.6.	Montaje y cableado de LOGO .....	46
B.3.6.1.	Cablear LOGO .....	46
B.3.6.2.	Conectar la alimentación .....	47
B.4.	PROGRAMACIÓN DE LOGO SIEMENS .....	47
B.4.1.	Bornes .....	49
B.4.1.1.	LOGO reconoce los siguientes bornes .....	50
B.4.2.	Bloques y números de bloque .....	50
B.4.2.1.	Bloques .....	51
B.4.2.2.	Funciones lógicas .....	51
B.4.2.3.	Representación de un bloque en la pantalla de LOGO .....	51
B.4.2.4.	Asignación de número de bloque .....	52
B.4.3.	Las 4 reglas de oro para manejar LOGO .....	53
B.4.3.1.	Regla 1 - Cambio del modo de operación .....	53
B.4.3.2.	Regla 2 - Salidas y entradas .....	54

B.4.3.3.	Regla 3 - Cursor y posicionamiento del cursor .....	54
B.4.3.4.	Regla 4 - Planificación .....	54
B.4.4.	Vista de conjunto de los menús de LOGO .....	55
B.4.5.	Introducción del programa en Logo.....	56
B.4.5.1.	Pasar al modo de operación "Programación" .....	56
B.4.5.2.	Menú principal de LOGO.....	56
B.4.5.3.	Menú Programación de LOGO: .....	57
B.4.5.4.	Menú Edición de LOGO: .....	57
B.4.5.5.	Modo de programación de LOGO: .....	57
B.4.5.6.	Introducir el programa.....	58
B.5.	LOGO SOFT COMFORT.....	59
B.5.1.	Descripción de la pantalla de operación .....	60
B.5.1.1.	Plataforma de programación .....	61
B.5.1.2.	Elementos de la pantalla de operación .....	61
B.5.1.3.	Barra de menús.....	61
B.5.1.4.	Barra de Herramientas.....	61
B.5.1.5.	Barra de símbolos estándar.....	62
B.5.1.6.	Barra de herramientas Herramientas .....	62
B.1.1.1.	Línea de estado.....	63
B.1.2.	Elaboración de programa.....	63
B.1.2.1.	Introducción de un nuevo programa.....	63

B.1.2.2.	Elaboración mediante barra de íconos.....	64
B.1.2.3.	Selección de bloques funcionales.....	64
B.1.2.4.	Unión de bloques funcionales: Reglas .....	65
B.1.2.5.	Carga del programa desde LOGO al PC. ....	65
B.1.2.6.	Interfaz de usuario.....	66
B.1.2.7.	Seleccionar bloques.....	67
B.1.2.8.	Conectar bloques .....	70
B.1.2.9.	Modificar y optimizar la visualización.....	71
B.1.2.10.	Editar líneas de conexión seleccionadas.....	71
B.1.2.11.	Almacenamiento de esquema de conexiones terminado .....	72
B.1.2.12.	Simulación del circuito.....	72
B.1.2.13.	Test online .....	73
C.	MATERIALES PARA ELABORACIÓN DE PROYECTO.....	74
D.	PROCESO METODOLÓGICO.....	75
D.1.	IMPLEMENTACIÓN DE MÓDULO DE CAMBIO DE HORARIO DE CLASES .	75
D.1.1.	Construcción del Módulo de Prácticas con PLC .....	75
D.1.1.1.	Diseño.....	75
D.1.1.2.	Montaje y Armado .....	76
D.1.1.2.1.	Conectores,.....	78
D.1.1.2.2.	Sirena.....	78
D.1.1.2.3.	Dispositivos de Protección .....	78



D.1.2.	Programación.....	79
D.1.2.1.	Alarma para cambio de horario de clase con PLC .....	79
E.	CONCLUSIONES .....	83
F.	RECOMENDACIONES .....	84
G.	BIBLIOGRAFÍA.....	85
H.	ANEXOS.....	87
	PROYECTO.....	87

**“AUTOMATIZACIÓN DE UN  
PROTOTIPO DE SISTEMA SONORO  
CON MINI-PLC PARA EL CAMBIO DE  
JORNADA EN EL AEIRNNR”**

## A. INTRODUCCIÓN

Con la llegada de los autómatas programables, los llamados PLC, la industria sufrió un impulso importante, que ha facilitado de forma notable que los procesos de producción o control se hayan flexibilizado mucho.

Cuando se inventaron, comenzaron llamándose PC (Controlador programable), pero con la llegada de los ordenadores personales de IBM, cambió su nombre a PLC. En Europa se les conoce como autómatas programables. Sin embargo, la definición más apropiada sería: Sistema Industrial de Control Automático que trabaja bajo una secuencia almacenada en memoria, de instrucciones lógicas.

Un controlador de lógica programable (PLC), es una pequeña computadora usada en la automatización de procesos del mundo real, tales como líneas de producción, máquinas herramientas, manejo de partes, empaquetado, bandas transportadoras, estaciones de bombeo, semáforos, etc. El rango de complejidad de los sistemas controlados mediante PLC va desde aplicaciones pequeñas dedicadas hasta poderosas y extremadamente complejas líneas de ensamblado (por ejemplo, en la manufactura de vehículos). El PLC usualmente utiliza un microprocesador. A diferencia de la computadoras de propósito general, el PLC es empacado y diseñado para trabajar en amplios rangos de temperatura, suciedad, y son inmunes al ruido eléctrico. Mecánicamente son más fuertes y resistentes a la vibración e impacto.

Con este precedente en el presente trabajo se desarrollará un proyecto con la aplicación de PLC, el cual responderá a los objetivos planteados conforme se detallaron con anterioridad en el resumen.

## **B. DESCRIPCIÓN TÉCNICA**

### **B.1. SISTEMAS DE CONTROL AUTOMÁTICO**

El control automático de procesos constituye un caso particular de la "automatización", que se puede definir como: *“la supresión total o parcial de la intervención humana en la ejecución de tareas agrícolas, industriales, domésticas, administrativas o científicas”*. La Real Academia de Ciencias Exactas Físicas y Naturales define la “automática” como: *“el estudio de los métodos y procedimientos cuya finalidad es la sustitución del operador humano por un operador artificial en la generación de una tarea física o mental previamente programada”*.

Se considera como *proceso*, al conjunto de los diferentes momentos sucesivos de un fenómeno natural o artificial o también al de las diferentes acciones sucesivas realizadas para conseguir un determinado resultado. Un *sistema* es una combinación de componentes que actúan conjuntamente para cumplir un determinado objetivo.

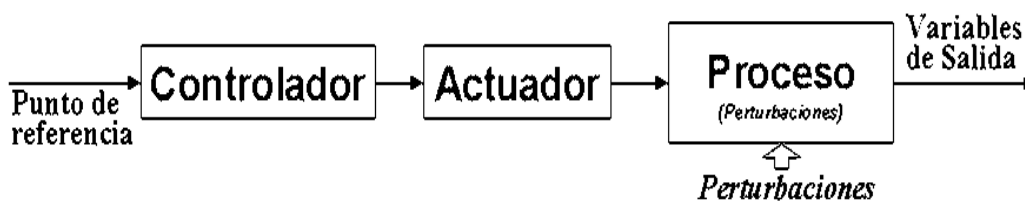
Se denomina *perturbaciones de un proceso* como aquellos hechos o situaciones que se producen en el sistema y que tienden a afectar adversamente al desarrollo del proceso. Si la perturbación se genera dentro del sistema, se denomina *interna*, mientras que una *externa* se genera fuera del mismo.

La palabra *control* se toma en el sentido de regular, dirigir o mandar y se puede considerar como el conjunto de acciones complementarias encaminadas al mando y dominio de otras actividades y por ello, un *sistema de control* es una interconexión de componentes que forman una configuración de tal manera que mandan, dirigen o regulan a un sistema según una respuesta deseada del mismo.

Se denomina *consigna* (punto de referencia) a las condiciones de referencia en las que esperamos que se desarrolle el proceso o dicho de otra forma, representa el comportamiento deseado para el mismo. Con la consigna se establece los posibles márgenes de variación de determinadas magnitudes que serán comparadas con otras de la misma naturaleza medidas en el proceso y que lo caracterizan.

Los sistemas de control, según que el desarrollo del proceso tenga efecto o no sobre la acción de control, se pueden clasificar en: sistemas de *bucle (o lazo) abierto* y sistemas de *bucle cerrado*.

Los *sistemas en bucle abierto* se caracterizan porque la información o variables que controlan el proceso circulan en una sola dirección, desde la unidad de control al proceso. El sistema de control no recibe ninguna información sobre el desarrollo (variables de salida) del mismo y, por tanto, éste no tiene ningún efecto sobre la acción de control. La *Figura 1* muestra el diagrama de bloques de un sistema de control de este tipo:

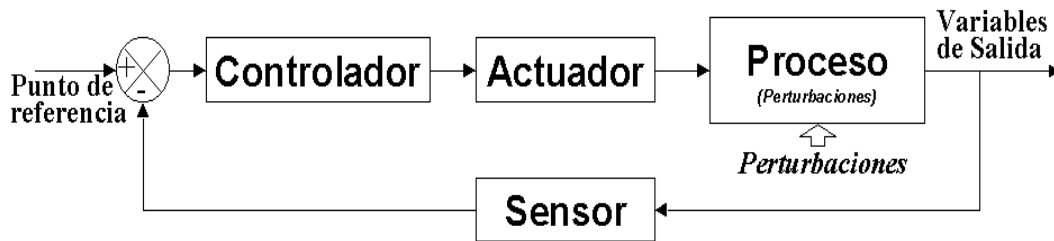


**Figura 1.** Diagrama de bloques de un sistema de control en bucle abierto

En estos sistemas, la consigna determina una condición de operación fija no adaptable a los posibles cambios. La precisión depende de la calibración y, cuando se producen perturbaciones (internas o externas), el proceso no se desarrolla de la forma esperada.

Los *sistemas en bucle cerrado*, se tiende a mantener una relación preestablecida entre el desarrollo del proceso (variables de salida) y las entradas de referencia (consigna),

comparándolas y utilizando su diferencia como señal de control. En la *Figura 2* se muestra el diagrama de bloques de uno de estos sistemas.



**Figura 2.** Diagrama de bloques de un sistema de control en bucle

Cualquier proceso que se quiera controlar, estará sometido a perturbaciones (internas y externas) que tenderán a desviarlo de su desarrollo deseado. Para conocer el desarrollo del proceso y poder compensar así sus desviaciones, se seleccionan aquellas magnitudes que lo caractericen, y que son medidas a través de los captadores o sensores, de tal forma que nos permiten conocer si el proceso está evolucionando dentro de los márgenes preestablecidos o no.

Las señales que producen los sensores se comparan con las señales de referencia o consigna (que establecen la situación deseada) de tal forma que la señal diferencia de ambas (señal error) determina las acciones a realizar, permitiendo acercar la respuesta del sistema a la deseada.

El término lazo o bucle cerrado implica el *uso de la acción de control realimentado para reducir el error en el sistema.*

En la mayoría de los procesos controlados se utiliza el de lazo cerrado, ya que posibilita tanto el control continuo de determinadas magnitudes y por tanto la posibilidad de reacción sobre su desarrollo, así como en aquellos en que la acción total se puede subdividir en acciones parciales de manera que para realizar una determinada es

necesario que se hayan realizado las precedentes y por ello hay que conocer esa circunstancia.

Los sistemas en bucle cerrado los procesos se desarrollan reaccionando y compensando las perturbaciones que puedan producirse, mientras que en los de bucle abierto las perturbaciones lo pueden modificar o alterar.

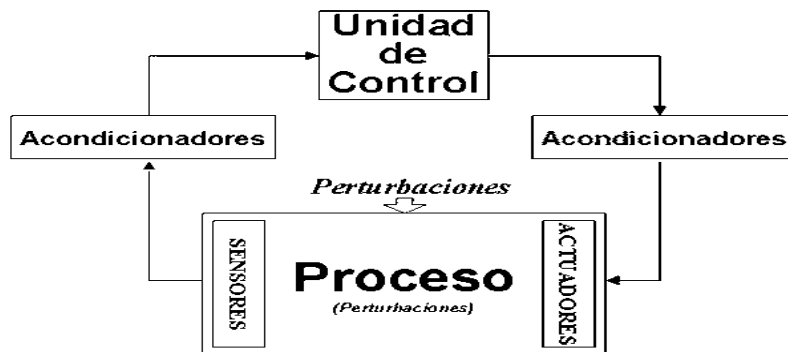
Los sistemas en bucle abierto no presentan problemas de estabilidad (al no existir realimentación), mientras que en los de bucle cerrado dichos problemas pueden ser importantes si se tiende a corregir en exceso las desviaciones producidas.

Se indicará, por último, que la automatización de un proceso no consiste únicamente en sustituir uno o varios operadores humanos por dispositivos que realicen el mismo trabajo, sino que implica la reconsideración más o menos profunda de la naturaleza y de los elementos del mismo y obliga a analizar las costumbres adquiridas y las soluciones tradicionales.

### **B.1.1. Componentes fundamentales de los sistemas de control**

En la Figura 3 se ha representado un diagrama de bloques simplificado de un sistema de control en bucle cerrado. En él aparecen los tres elementos básicos de este tipo de control: los *captadores* (sensores o transductores) que miden cómo se desarrolla el proceso, los *actuadores* que nos permiten intervenir sobre él y la *unidad de control* que, a partir de la información proveniente de los sensores y de su comparación con la consigna, genera las órdenes necesarias para los actuadores.





**Figura 3.** Elementos fundamentales de un sistema de control en bucle cerrado

Si el proceso se desarrolla correctamente se deberán mantener las actuaciones que se realizan sobre él, por el contrario si trata de salirse de los márgenes preestablecidos, el sistema deberá reaccionar mediante la actuación adecuada para volver a la evolución deseada. Los captadores y actuadores, por tanto, conectan el sistema de control con el proceso.

Por último, aparece un cuarto elemento que, con el nombre genérico de  *acondicionadores de señal*  engloban todas aquellas etapas necesarias para que las señales se transmitan adecuadamente entre los diferentes elementos del sistema. En los próximos apartados se profundiza un poco más sobre estos cuatro elementos básicos de un sistema de control.

#### ***B.1.1.1. Captadores (sensores o transductores)***

Las denominaciones de sensor y transductor se utilizan generalmente como sinónimos, aunque la primera tiene un significado más amplio. Se denomina *transductor* a todo dispositivo que convierte una señal de una forma física determinada, en otra de diferente forma física convirtiendo un tipo de energía en otro (por ejemplo un motor convierte energía eléctrica en energía mecánica).

Por otro lado, la palabra *sensor* es un sistema capaz de ser sensible (medir) determinadas magnitudes físicas (normalmente no percibidas por nuestros sentidos) y proporcionar una señal de salida utilizable (bien directamente o tras un procesamiento). Un sensor es, por tanto, un dispositivo que detecta la variación de una magnitud física y la convierte en una señal útil para el sistema que se servirá de ella y controlará un proceso.

Existen diferentes formas de clasificar a los sensores según distintos criterios, por ejemplo, según la señal de salida, los sensores pueden ser *analógicos* (si la salida varía de forma continua dentro del intervalo de medida), o *digitales* (si la salida varía en forma de saltos o pasos discretos, codificada mediante dos niveles eléctricos claramente diferenciados), siendo mayoría los primeros frente a los segundos. Según el origen de la energía de la señal de salida, se puede hablar de *generadores* y *moduladores*. En los primeros dicha energía se obtiene a partir de la energía de la señal de entrada no necesitando alimentación (energía) exterior, por el contrario en los segundos la energía de salida proviene principalmente de la fuente de alimentación y lo que hace el sensor es darle forma (modularla) según varíe la magnitud medida. Por otro lado, el tiempo de respuesta de los sensores (o su característica dinámica) nos permite clasificarlos en *estáticos* o *dinámicos*. Los primeros son adecuados para medida de magnitudes constantes (o de variación lenta en el tiempo) mientras que los segundos lo son para las de variación rápida, siendo esta característica muy importante ya que el sensor debe adaptarse a la velocidad de cambio de la magnitud medida. También se pueden clasificar por la magnitud que miden (temperatura, presión, humedad, etc.), campo de aplicación, rangos de medida, modo de operación, etc.

Aunque cada sensor debería producir una señal de salida proporcional únicamente a la magnitud para la que ha sido diseñado, en la práctica, los sensores se ven afectados por otras magnitudes, ello obliga a realizar una cuidadosa selección entre los diferentes tipos, ya que la adecuada selección de los sensores es fundamental para el buen funcionamiento de un sistema de control. Un ejemplo se puede ver en la medida de la humedad del suelo mediante sondas basadas en la conductividad eléctrica, ya que aunque esta característica eléctrica depende de la humedad del suelo, también puede verse afectada por otros factores tales como la temperatura, la existencia de sales disueltas (abonos), etc.

En base a lo anteriormente indicado, se ve que son numerosas las características que hay que tener en cuenta a la hora de seleccionar el sensor más adecuado para cada aplicación, pero las más importantes a considerar son: *naturaleza de las magnitudes a medir, tipo de fenómeno físico a medir (estático o dinámico), rangos de medida, sensibilidad, resolución, precisión requerida, linealidad, deriva* y muy especialmente *el tipo de señal eléctrica de salida (tensión o corriente)*.

Los sensores más comúnmente utilizados en los sistemas de riego son: caudalímetros, sondas de nivel, transductores de presión, presostátos, tensiómetros, termómetros, higrómetros, piranómetros, anemómetros, etc.

#### ***B.1.1.2. Actuadores***

La función de los actuadores es realizar, en respuesta a una señal de entrada (proveniente de la unidad de control), una acción eléctrica, mecánica, hidráulica, neumática o térmica sobre un determinado proceso, con el fin de que éste se desarrolle en la forma adecuada.

Los actuadores en general, se pueden agrupar en *electromecánicos, electrohidráulicos, electroneumáticos, refrigeradores y calefactores, etc.*

### ***B.1.1.3. Unidades de control***

En los últimos años se ha producido un importante cambio en la concepción y estructura de los sistemas de control, habiéndose pasado de los sistemas analógicos a los digitales.

La utilización del microprocesador (microcontrolador) y del ordenador en el control de procesos ha permitido, no sólo la obtención de mejores características y mayores prestaciones en el control que las que se obtendrían con un sistema analógico, sino que además posibilitar funciones de adquisición y tratamiento de los datos del proceso (comprobaciones, tendencias, promedios, etc.). Los controladores digitales están muy por encima de los analógicos en cuanto a precisión desde el punto de vista de ruidos internos y efectos de deriva.

Las unidades de control comerciales basadas en microprocesador o microcontrolador reciben el nombre de autómatas programables. Estos equipos suelen tener una configuración modular en cuanto al número de entradas, salidas, memoria de almacenamiento de datos, etc. (lo que permite adaptarlos a diferentes necesidades), se programan fácilmente mediante lenguajes especiales de programación y a un precio, aunque variable según características, no elevado.

La elección de la unidad de control depende del tipo de aplicación, estando condicionada por la complejidad del sistema a controlar, la velocidad de respuesta del sistema, las condiciones ambientales, presupuesto de ejecución, etc.

Un elemento fundamental en los sistemas que utilizan microprocesadores u ordenadores (hardware) como unidad de control lo constituyen los programas (software) que

determinaran su funcionamiento, de tal forma que ambos constituyan un sistema integrado. Los programas realizados para el control de un determinado proceso deben contemplar todas las situaciones que se puedan producir en él, y establecer las pautas de actuación para su adecuado tratamiento.

Los programas además de controlar el proceso pueden incluir otras rutinas para el tratamiento de situaciones fuera de margen (alarmas), de adquisición y almacenamiento de datos, de procesamiento en tiempo real, etc. Se indicará, por último, que los programas de control deben, en general, diseñarse para que su utilización sea fácil y, en algunos casos, incluso para personas sin conocimientos en informática.

#### ***B.1.1.4. Acondicionamiento de señales***

Las señales eléctricas producidas en los sensores que deben llegar a la unidad de control, así como las generadas en ésta última y que deben enviarse a los actuadores, precisan de un medio adecuado para su transmisión en el que no se pierda la información que contienen ya que si ésta se produjera, el sistema no podría funcionar correctamente (medidas erróneas y ordenes de actuación incorrectas).

Un factor fundamental en la transmisión de señales a distancia es precisamente esta última entre los elementos a comunicar y que puede variar desde unos pocos metros hasta cientos de kilómetros. La distancia es, por tanto, un factor primordial que condiciona el medio físico a utilizar para la comunicación.

Los principales medios utilizados con este tipo de señales son: los cables eléctricos de diferente tipo (principalmente coaxial), las líneas telefónicas disponibles, las ondas electromagnéticas (radio), fibras ópticas, e incluso vía satélite (estos dos últimos raramente utilizados en aplicaciones de control). La utilización de uno u otro medio

depende, además de las distancias entre los elementos, de las características del proceso a controlar y la infraestructura existente y del presupuesto disponible para este fin.

En la transmisión de las señales eléctricas por cualquiera de los medios indicados, suelen ser necesarias etapas de acondicionamiento eléctrico (amplificación, filtrado, conversión tensión-corriente, linealización, etc.) que las adapte a la naturaleza, margen de entrada y, en general, a las características de entrada tanto de la unidad de control como de los actuadores.

Para proteger los sistemas de control de cualquier contacto accidental con la red de distribución de energía eléctrica, así como de descargas por distintas tomas de neutros que estén a diferente potencial, o bien por fallo o accidente en la manipulación de las conexiones, se recurre al aislamiento galvánico entre las señales que provienen de los sensores y las que se conectan a la unidad de control.

Este tipo de aislamiento, principalmente de tipo óptico, permite la comunicación entre los diferentes elementos del sistema utilizando dispositivos fotoemisores y fotodetectores (acoplamiento por luz) pero sin que exista conexión eléctrica entre ellos.

El acondicionamiento de las señales digitales por la naturaleza de estas es, en general, más sencillo que el correspondiente de las de tipo analógico.

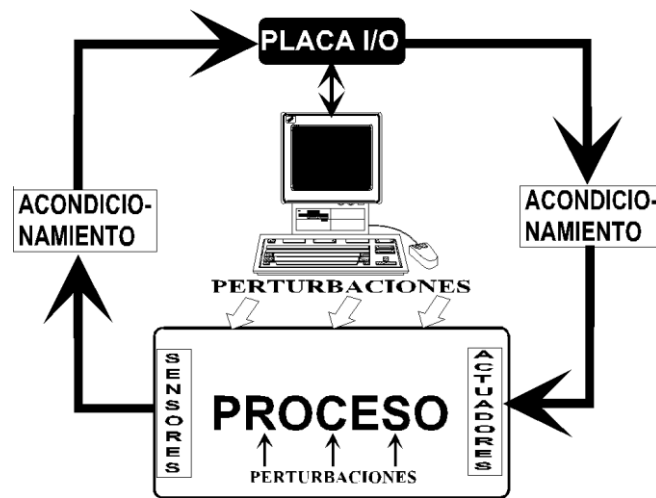
### **B.1.2. Control de procesos por ordenador**

Los grandes avances en el campo de los microordenadores (PC's), cada vez más rápidos y fiables y también más baratos, ha permitido su utilización en el control de sistemas a pequeña escala a precios competitivos.

La utilización del ordenador en sistemas de control posibilita además tareas de tratamiento de datos y supervisión.

En cuanto a la primera, destacar las amplias posibilidades que presenta el tratamiento digital de la información (almacenamiento, representación gráfica, etc.). Respecto a las tareas de supervisión, resaltar que permiten la comprobación de límites de tolerancia de las variables del proceso y, en su caso, activar las correspondientes alarmas.

En la *Figura 4* se muestra el diagrama de bloques de un sistema que utiliza el ordenador.



*Figura 4. Diagrama de bloques del control por ordenador*

Siguiendo el esquema propuesto, se encuentra en primer lugar los sensores, en contacto con las magnitudes del proceso a medir, estos convierten los parámetros físicos (temperatura, humedad, etc.) en señales eléctricas de tensión o corriente, con alta o baja impedancia de salida, pudiendo producir señales diferenciales de salida con ruido en modo común, etc.

La señal producida por el sensor debe adaptarse, en su caso, a la naturaleza y margen de entrada que precisa la tarjeta o placa de "Entrada/Salida" (I/O, Input/Output), además de poder ser enviada a ésta a grandes distancias. Esta función la realizan los  *acondicionadores* de señal. Como se dijo, en algunas aplicaciones es necesario que exista un aislamiento galvánico entre las señales que provienen de los sensores y las que se

conectan al ordenador (buses).

Debido al espectacular desarrollo y a la utilización generalizada de los ordenadores tipo PC en sistemas de control y de adquisición de datos, se han desarrollado placas I/O de altas prestaciones. El éxito de estas placas multifunción se debe a su potencialidad para la resolución de problemas tanto en aplicaciones generales como específicas, a un precio cada vez más reducido. Aunque existen en el mercado tarjetas con diferentes características y prestaciones, básicamente todas ellas disponen de entradas y salidas analógicas y digitales.

Como es sabido, los ordenadores únicamente procesan información de tipo digital, es decir, tanto las señales que puede recibir del exterior (de los sensores) como las que puede enviar hacia el exterior (a los actuadores) son de tipo digital.

Algunos sensores, como se indicó, producen señales de tipo digital, por lo que podrían aplicarse directamente a las entradas (buses) del ordenador, pero la mayoría producen señales de tipo analógico, necesitando una conversión (o acondicionamiento) de la información para que pueda ser tratada por el ordenador. Este tipo de transformación de la señal se denomina conversión Analógica/Digital (A/D) o "digitalización", que básicamente consiste en la transformación en un código binario discreto adecuado para su procesamiento digital.

La conversión A/D precisa de cuatro etapas u operaciones fundamentales: *muestreo*, *retención*, *cuantificación* y *codificación*.

Con el *muestreo* se obtienen los valores instantáneos de la señal a convertir. La señal analógica debe ser muestreada periódicamente para extraer la información que contiene. La elección del período de la toma de muestras es fundamental y viene determinado



principalmente por la frecuencia de la señal analógica. En efecto un período largo de muestreo (baja frecuencia de muestreo) puede producir una pérdida de información, por el contrario un período muy corto hace que se tomen más muestras de las necesarias lo que produce un "exceso" de la información a tratar y, en general, hará que los sistemas resulten más caros.

La frecuencia de muestreo depende, por tanto, de la variación en el tiempo de la señal a convertir y se determina por la aplicación del teorema de *Shannon* que se enuncia: " *Para que el muestreo de una señal con espectro de frecuencia limitado no implique pérdida de información, es preciso que la frecuencia de muestreo sea, al menos, el doble de la frecuencia máxima de la señal analógica*".

La etapa de *retención* es imprescindible para que la señal analógica no cambie durante el tiempo en que se está produciendo la conversión.

En la etapa de *cuantificación* se convierte la señal analógica, que puede tomar infinitos valores en su margen de variación, en un conjunto finito (discreto) de N valores que dependerá de la resolución del convertidor.

En la última etapa de *codificación* se asigna un determinado código (dependiendo de cada aplicación) a cada uno de los niveles obtenidos en la cuantificación.

Las entradas al sistema de control determinan, siguiendo los algoritmos (programas) de control, las actuaciones necesarias en el proceso. Dicha actuación se produce a través de las salidas (analógicas y digitales) de la placa I/O. Estas señales, igual que las producidas por los sensores, deben ser acondicionadas y transmitidas hacia los actuadores, siendo este tipo de acondicionamiento similar al indicado anteriormente.

Algunos actuadores se pueden controlar con señales de tipo digital, en cambio otros precisan la aplicación de señales analógicas, en cuyo caso es necesario realizar una conversión D/A (caso dual al de la A/D) entre las salidas del ordenador y la necesaria para el actuador.

## **B.2. CONTROLADOR LÓGICO PROGRAMABLE PLC**

La ingeniería de control se ha desarrollado a través del tiempo. En el pasado los humanos eran el método para controlar los sistemas. En tiempos recientes la electricidad se ha utilizado para el control, los primeros controles eléctricos fueron los relevadores. Los relevadores permiten encender o apagar un circuito eléctrico sin necesidad de un interruptor mecánico. Fue inventado por Joseph Henry en 1835.

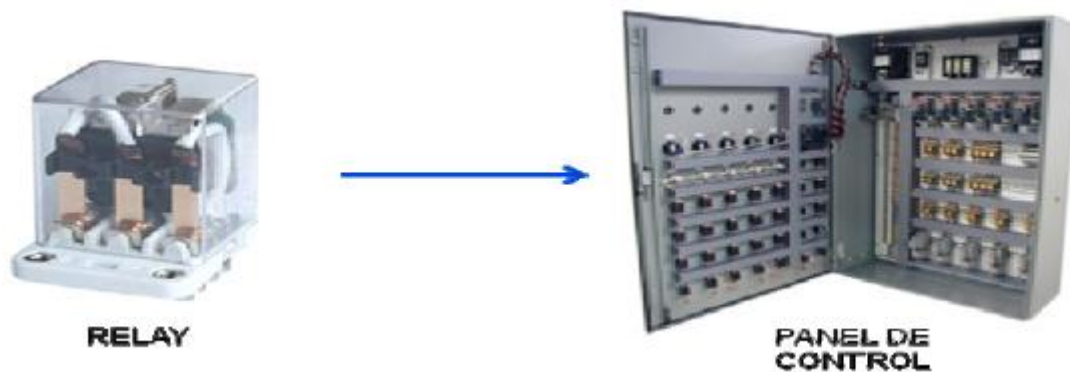
Un controlador de lógica programable es una pequeña computadora usada en la automatización de procesos del mundo real, tales como líneas de producción, máquinas herramientas, manejo de partes, empaquetado, bandas transportadoras, estaciones de bombeo, semáforos, etc., (Figura 5). El rango de complejidad de los sistemas controlados mediante PLC va desde aplicaciones pequeñas dedicadas hasta poderosas y extremadamente complejas líneas de ensamblado (por ejemplo, en la manufactura de vehículos). El PLC usualmente utiliza un microprocesador. A diferencia de la computadoras de propósito general, el PLC es empacado y diseñado para trabajar en amplios rangos de temperatura, suciedad, y son inmunes al ruido eléctrico. Mecánicamente son más fuertes y resistentes a la vibración e impacto.



*Figura 5. Procesos industriales que utilizan PLC*

El control automático permite la producción de un producto consistente a un costo razonable y el PLC es la tecnología de control prevalente en la manufactura.

Antes de los PLC, la única forma de implementar un circuito de control para una máquina era mediante el uso de relés (Figura 6). Una máquina que tenía que controlar muchos motores y solenoides requería de muchos relés, los cuales tenían que ser alambrados en el orden correcto para lograr el funcionamiento deseado.



*Figura 6. Panel de control con relés electromagnéticos*

Dado que los relés eran electromecánicos los mismos fallaban frecuentemente. Fallas en las bobinas y contactos eran difíciles de diagnosticar y reemplazar, además si un cambio en el circuito era necesario significaba el re-alambrado de todos los relés involucrados. Lo anterior sin mencionar que un panel con centenas de relés consumía grandes cantidades de energía. (Ferreira, 2005)

Los PLC fueron inventados como un reemplazo menos costoso para viejos sistemas automatizados que usaban centenas de relés y temporizadores. Un PLC puede ser programado para reemplazar miles de relés. (Ferreira, 2005)

### **B.2.1. Desarrollos de los PLC**

Los Controladores Lógico Programables (PLC) fueron inventados como respuesta a las necesidades de la industria automotriz. Inicialmente fueron adoptados por las empresas para sustituir la lógica cableada. En 1968 GM Hydramatic (la división de transmisiones automáticas de General Motors) emitió una solicitud para realizar una propuesta que sustituyera la lógica cableada. La propuesta ganadora fue realizada por Bedford Associates. El primer PLC fue el MODICON 084. (Morriss Brian, 2000)

En la Tabla 1 se muestra, de forma muy breve, eventos, considerados importantes en el desarrollo del PLC, y las fechas en que estos tuvieron lugar. Como se puede apreciar, la tecnología de los PLC no es nueva.

**Tabla 1. Desarrollo de los PLC**

<b>Año</b>	<b>Naturaleza del desarrollo</b>
1968	Concepto inicial desarrollado
1969	CPU del controlador (hardware), con instrucciones lógicas, 1K de memoria y 128 I/O
1971	Primeras aplicaciones fuera de la industria automotriz
1972	Instrucciones de temporización y conteo
1973	Instrucciones aritméticas y de movimiento de datos. Comunicaciones Computadora – PLC
1974	Varios procesadores en un PLC, 12K de memoria, 1024 I/O, terminal de programación CRT.
1975	Control PID
1976	Sistema remoto de I/O
1977	Pequeños PLC basados en microprocesadores
1978	Comunicaciones PLC – PLC
1980	Módulos de I/O inteligentes, Programación con computadoras personales Características de software ampliadas (e.g., documentación)
1983	PLC grandes: 4M de memoria, 8192 puntos de I/O
1985	Protocolo para la automatización de manufactura de la GM (MAP)
1986	Otros lenguajes de programación (además de Ladder)
1993	Conexión a ETHERNET Lenguajes estándar IEC 61131-3
1994	Protocolos estándar para I/O remotas
1996	Control basado en PC como alternativa a los PLC
2000	Redes de comunicaciones IEC 61158

*Fuente:* [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lmt/maza\\_c\\_ac/capitulo4.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lmt/maza_c_ac/capitulo4.pdf)

Paralelo al avance en la tecnología de los PLC, también lo han hecho los lenguajes de programación y las capacidades de comunicación, junto a otras importantes características.

Hoy en día, los PLC ofrecen “*scan time*” más rápidos, espacios eficientes y de alta densidad para sistemas de entradas y salidas, e interfaces especiales que permiten que dispositivos no tradicionales sean conectados directamente al PLC. No solamente pueden conectarse con otros sistemas de control, ellos pueden realizar funciones de reporte y diagnosticar sus propias fallas así como la falla de una máquina o proceso.

### **B.2.2. Definición y principios de operación**

Un controlador lógico programable (Programmable Logic Controller PLC) (Figura 7), es un dispositivo operado digitalmente, que usa una memoria para el almacenamiento interno de instrucciones con el fin de implementar funciones específicas, tales como lógica, secuenciación, registro y control de tiempos, conteo y operaciones aritméticas, para controlar a través de entradas/salidas digitales o analógicas, varios tipos de máquinas o procesos. (Ferreira, 2005)



*Figura 7. PLC Siemens*

Los PLC operan de manera secuencial y cíclica, es decir, una vez finalizado el recorrido completo de un programa, comienza a ejecutar su primera instrucción.

Los elementos que contiene un PLC son:

- **Unidad Central de proceso:** La unidad central es el “cerebro” del PLC. Este toma las decisiones relacionadas al control de la máquina o proceso. Durante su operación, el CPU recibe entradas de diferentes dispositivos de sensado, ejecuta decisiones lógicas, basadas en un programa almacenado en la memoria, y controla los dispositivos de salida de acuerdo al resultado de la lógica programada.
- **Módulos de entrada y salida:** son la sección del PLC en donde sensores y actuadores son conectados y a través de los cuales el PLC monitorea y controla el proceso.
- **Fuente de Alimentación:** La fuente de alimentación convierte altos voltajes de corriente de línea (115V 230V CA) a bajos voltajes (5V, 15V, 24V CD) requeridos por el CPU y los módulos de entradas y salidas.
- Dispositivos periféricos
- Interfaces

### **B.2.3. Funcionamiento del PLC**

El funcionamiento del PLC es un continuo ciclo cerrado y una vez que se pone en marcha, el procesador realiza una serie de tareas según el siguiente orden:

- a. Al encender el procesador ejecuta un auto-chequeo de encendido y bloquea las salidas. A continuación, si el chequeo ha resultado correcto, el PLC entra en el modo de operación normal.
- b. El siguiente paso lee el estado de las entradas y las almacena en una zona de la memoria que se llama tabla de imagen de entradas.

- c. En base a su programa de control, el PLC actualiza una zona de la memoria llamada tabla de imagen de salida.
- d. A continuación el procesador actualiza el estado de las salidas "copiando" hacia los módulos de salida el estado de la tabla de imagen de salidas (de este modo se controla el estado de los módulos de salida del PLC, relay, triacs, etc.).
- e. Vuelve a ejecutar el paso b)

Cada ciclo de ejecución se llama ciclo de barrido (scan), el cual normalmente se divide en:

- Verificación de las entradas y salidas
- Ejecución del programa

#### ***B.2.3.1. Otras funciones adicionales del PLC***

- a. En cada ciclo del programa, el PLC efectúa un chequeo del funcionamiento del sistema reportando el resultado en la memoria, que puede ser comprobada por el programa del usuario.
- b. El PLC puede controlar el estado de las Inicializaciones de los elementos del sistema: cada inicio de un microprocesador también se comunica a la memoria del PLC.
- c. Guarda los estados de las entradas y salidas en memoria: Le puedes indicar al PLC el estado que deseas que presenten las salidas o las variables internas, en el caso de que se produzca un fallo o una falta de energía en el equipo. Esta funcionalidad es esencial cuando se quieren proteger los datos de salida del proceso.



- d. Capacidad modular: Gracias a la utilización de Microprocesadores, puedes expandir los sistemas PLC usando módulos de expansión, en función de lo que te requiera el crecimiento de tu sistema. Puede expandirse a través de entradas y salidas digitales, análogas, etc., así como también con unidades remotas y de comunicación.

Los PLC han ganado popularidad en las industrias y probablemente continuarán predominando por algún tiempo, debido a las ventajas que ofrecen:

- Son un gasto efectivo para controlar sistemas complejos. [6]
- Son flexibles y pueden ser aplicados para controlar otros sistemas de manera rápida y fácil. [6]
- Su capacidad computacional permite diseñar controles más complejos.
- La ayuda para resolver problemas permite programar fácilmente y reduce el tiempo de inactividad del proceso. [6]
- Sus componentes confiables hacen posible que pueda operar varios años sin fallas. [6]
- Capacidad de entradas y salidas. [6]
- Monitoreo. [6]
- Velocidad de operación. [6]
- Están diseñados para trabajar en condiciones severas como: vibraciones, campos magnéticos, humedad, temperaturas extremas. [6]

### B.2.4. Arquitectura Básica de los PLC

La arquitectura básica de un PLC es mostrada en la Figura 8. Los principales componentes son *el módulo procesador, la fuente de alimentación y los módulos de entrada y salida.*

El módulo de procesador consiste de la unidad central de procesamiento (CPU) y la memoria. Además del microprocesador, la CPU también contiene al menos una interface para un dispositivo de programación y puede contener interfaces para I/O remotas y otras redes de comunicación. (Melore Phil, 2001)

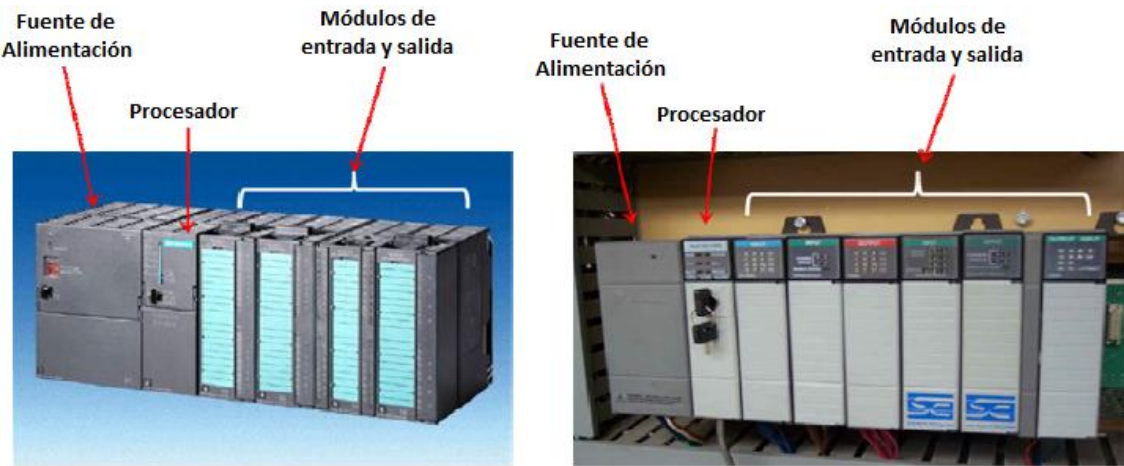
La fuente de alimentación es usualmente un módulo separado y los módulos de I/O están separados del procesador. Los tipos de módulos de entrada pueden ser *discretos* (on / off), *analógicos* (variables continuas), y *módulos especiales* tales como Control de movimiento y contadores de alta velocidad. Los dispositivos de campo son conectados a los módulos de I/O. (Melore Phil, 2001)



**Figura 8.** Arquitectura de un PLC

En dependencia de la cantidad de I/O y del procesador del PLC particular, los módulos de I/O pueden estar en el mismo chasis que el procesador y en uno o varios chasis.

Figura 9.



*Figura 9. Arreglo de PLC*

#### ***B.2.4.1. Unidad Central de Proceso – CPU***

Lleva a cabo la mayoría de los procesos del sistema, su desempeño depende de una lista de ejecución que se provee, denominada programa. Los programas que se ejecutan son dos, el primero de autoconfiguración cuando el sistema arranca y el segundo de usuario, diseñado para una aplicación específica. Por lo general la CPU viene integrada en un chip semiconductor caso en el cual recibe el nombre de micro-procesador. (Little, 2002)

Los principales componentes funcionales de la CPU son:

- ***La Unidad Lógica y Aritmética - ALU:*** Realiza operaciones aritméticas como: suma, resta, multiplicación, comparación, desplazamiento, entre otras, y operaciones lógicas como: AND, OR, EXOR, NOT, entre otras.

- **La Unidad de Control:** Se encarga de: a) Sincronizar las tareas de la CPU, b) Determinar todas las rutas por las cuales fluirá la información a través de los buses, y c) Interpreta el programa. La Unidad de Control, tiene como salidas, centenares de líneas de selección, invasión y conmutación requeridas por los diferentes elementos de la CPU. Su función es habilitar los niveles lógicos adecuados a tiempos y secuencias precisos, con el fin de ejecutar una instrucción completa. La entrada de la Unidad de Control cuenta con un decodificador de instrucciones que recibe el código binario correspondiente a una instrucción de programa. La principal tarea de la Unidad de Control es leer secuencialmente los códigos de instrucción de la memoria del programa y hacer que el resto de la CPU ejecute dichas instrucciones.
- **Lenguaje de Máquina:** El programa a ejecutar es recibido por la CPU en código binario o lenguaje de máquina.
- **Banco de Registros:** Es una pequeña memoria interna de la CPU que almacena los datos temporales necesarios para la ejecución del programa.

#### **B.2.4.2. Buses:**

Caminos por los cuales fluye la información hacia los distintos componentes de la CPU.

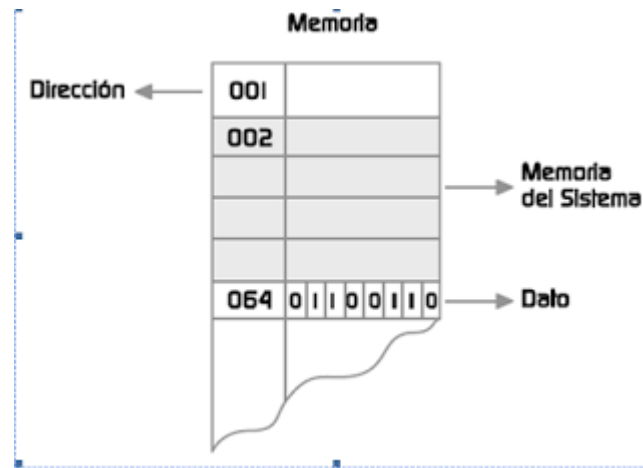
Los Buses se clasifican en:

- **Bus de Datos:** Canal que lleva y trae datos desde y hacia la memoria, los registros internos, la ALU y los puertos.
- **Bus de Direcciones:** Lleva valores de dirección hacia la memoria y el bus de direcciones externo.

- **Bus de Control:** Pone señales de control en los diversos bloques funcionales.

### B.2.4.3. Memoria

Almacén de información del sistema. Contiene datos numéricos en código binario y está dividida en posiciones de memoria, a cada una de las cuales le corresponde una dirección de memoria, cada posición de memoria es un arreglo de una determinada cantidad de bits (8 o 16 bits). Figura 10.



**Figura 10.** Estructura de la Memoria

Las posibilidades de acceso se clasifican en:

Memorias de Solo Lectura: Se utilizan para almacenar programas y datos constantes; La manera como están dispuestos los circuitos en la CPU determinan que solo se pueda leer su contenido.

Existen Memorias de Solo Lectura de los siguientes tipos:

- **Memoria de Solo Lectura - ROM (Read Only Memory):** Son grabadas por el fabricante del chip y su información no puede ser alterada.

- **Memoria de Solo Lectura Programable – PROM (Programmable Read Only Memory):** Salen de la fábrica en blanco para ser programadas por el usuario por una sola vez.
- **Memoria de Solo Lectura Programable y Borrable - EPROM (Erasable Programmable Read Only Memory):** Se diferencia de las PROM en que pueden ser borradas y reprogramadas; generalmente el borrado se efectúa con radiación ultravioleta.

### **B.2.5. Clasificación de PLC**

Debido a la gran variedad de tipos distintos de PLC, tanto en sus funciones, en su capacidad, en el número de I/O, en su tamaño de memoria, en su aspecto físico y otros, es que es posible clasificar los distintos tipos en varias categorías.

#### **B.2.5.1. PLC tipo Nano:**

Generalmente PLC de tipo compacto (Fuente, CPU e I/O integradas) que puede manejar un conjunto reducido de I/O, generalmente en un número inferior a 100. Permiten manejar entradas y salidas digitales y algunos módulos especiales. (Little, 2002)

#### **B.2.5.2. PLC tipo Compactos:**

Estos PLC tienen incorporado la Fuente de Alimentación, su CPU y módulos de I/O en un solo módulo principal y permiten manejar desde unas pocas I/O hasta varios cientos (alrededor de 500 I/O) , su tamaño es superior a los Nano PLC y soportan una gran variedad de módulos especiales, tales como:

- Entradas y salidas análogas
- Módulos contadores rápidos

- Módulos de comunicaciones
- Interfaces de operador
- Expansiones de i/o

#### ***B.2.5.3. PLC tipo Modular:***

Estos PLC se componen de un conjunto de elementos que conforman el controlador final, estos son:

- Rack
- Fuente de Alimentación
- CPU
- Módulos de I/O
- Comunicaciones.
- Contaje rápido.

### **B.2.6. Lenguajes de programación de PLC**

#### ***B.2.6.1. Programa y lenguaje de programación***

Se puede definir un programa como un conjunto de instrucciones, órdenes y símbolos reconocibles por el PLC, a través de su unidad de programación, que le permiten ejecutar una secuencia de control deseada. El Lenguaje de Programación en cambio, permite al usuario ingresar un programa de control en la memoria del PLC, usando una sintaxis establecida. (Melore Phil, 2001)

Al igual como los PLC se han desarrollado y expandido, los lenguajes de programación también se han desarrollado con ellos. Los lenguajes de hoy en día tienen nuevas y más

versátiles instrucciones y con mayor poder de computación. Por ejemplo, los PLC pueden transferir bloques de datos de una localización de memoria a otra, mientras al mismo tiempo llevan cabo operaciones lógicas y matemáticas en otro bloque. Como resultado de estas nuevas y expandidas instrucciones, los programas de control pueden ahora manejar datos más fácilmente.

Adicionalmente a las nuevas instrucciones de programación, el desarrollo de nuevos módulos de entradas y salidas también ha obligado a cambiar las instrucciones existentes.

#### ***B.2.6.2. Programas de aplicación y del sistema***

Los programas de aplicación que crean los usuarios están orientados a ejecutar, a través del controlador, tareas de automatización y control. Para ello, el usuario escribe el programa en el lenguaje de programación que mejor se adapte a su trabajo y con el que sienta poseer un mejor dominio. En este punto es importante señalar, que algunos fabricantes no ofrecen todas las formas de representación de lenguajes de programación, por lo que el usuario deberá adaptarse a la representación disponible.

Por otro lado, el conjunto de programas que realizan funciones operativas internas del controlador, incluyendo los traductores de lenguaje, reciben la denominación de programas del sistema o software del sistema. Un elemento importante de éste, es el sistema operativo, cuyos servicios incluyen el manejo de los dispositivos de entrada y salida del PLC, el almacenamiento de la información durante largos períodos, el procesamiento de los programas del usuario, etc. Estos programas ya vienen escritos y



están almacenados en una memoria No volátil dentro de la CPU, por lo tanto no se pierden ni alteran en caso de pérdida de alimentación al equipo.

#### ***B.2.6.3. Tipos de lenguajes de programación de PLC***

En la actualidad cada fabricante diseña su propio software de programación, lo que significa que existe una gran variedad comparable con la cantidad de PLC que hay en el mercado. No obstante, actualmente existen tres tipos de lenguajes de programación de PLC como los más difundidos a nivel mundial; estos son:

- Lenguaje de contactos o Ladder
- Lenguaje Booleano (Lista de instrucciones)
- Diagrama de funciones

Es obvio, que la gran diversidad de lenguajes de programación da lugar a que cada fabricante tenga su propia representación, originando cierta incomodidad al usuario cuando programa más de un PLC.

#### ***B.2.6.4. La norma IEC 1131-3***

La Comisión Electrotécnica Internacional (IEC) desarrolló el estándar IEC 1131, en un esfuerzo para estandarizar los Controladores Programables. Uno de los objetivos del Comité fue crear un conjunto común de instrucciones que podría ser usado en todos los PLC. Aunque el estándar 1131 alcanzó el estado de estándar internacional en agosto de 1992, el esfuerzo para crear un PLC estándar global ha sido una tarea muy difícil debido a la diversidad de fabricantes de PLC y a los problemas de incompatibilidad de programas entre marcas de PLC. (Melore Phil, 2001)

El estándar IEC 1131 para controladores programables consiste de cinco partes, una de las cuales hace referencia a los lenguajes de programación y es referida como la IEC 1131-3.

El estándar IEC 1131-3 define dos lenguajes gráficos y dos lenguajes basados en texto, para la programación de PLC. Los lenguajes gráficos utilizan símbolos para programar las instrucciones de control, mientras los lenguajes basados en texto, usan cadenas de caracteres para programar las instrucciones. (Little, 2002)

- Lenguajes Gráficos
  - Diagrama Ladder (LD)
  - Diagrama de Bloques de Funciones (FBD)
- Lenguajes Textuales
  - Lista de Instrucciones (IL)
  - Texto Estructurado (ST)

Adicionalmente, el estándar IEC 1131-3 incluye una forma de programación orientada a objetos llamada **Sequential Function Chart (SFC)**. SFC es a menudo categorizado como un lenguaje IEC 1131-3, pero éste es realmente una estructura organizacional que coordina los cuatro lenguajes estándares de programación (LD, FBD, IL y ST). La estructura del SFC tuvo sus raíces en el primer estándar francés de **Grafcet** (IEC 848).

#### ***B.2.6.5. Lenguaje ladder***

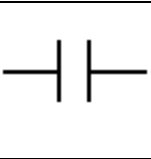
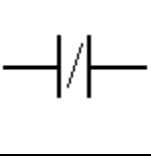
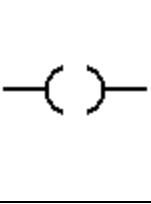
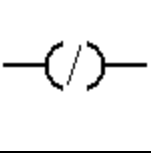
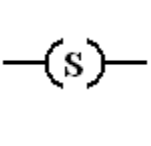
El LADDER, también denominado lenguaje de contactos o de escalera, es un lenguaje de programación gráfico muy popular dentro de los Controladores Lógicos Programables (PLC), debido a que está basado en los esquemas eléctricos de control

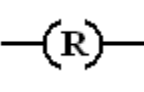
clásicos. De este modo, con los conocimientos que todo técnico eléctrico posee, es muy fácil adaptarse a la programación en este tipo de lenguaje. Su principal ventaja es que los símbolos básicos están normalizados según normas NEMA y son empleados por todos los fabricantes. (Little, 2002)

- **Elementos de programación**

Para programar un PLC con LADDER, además de estar familiarizado con las reglas de los circuitos de conmutación, es necesario conocer cada uno de los elementos de que consta este lenguaje. En la Tabla 2 se puede observar los símbolos de los elementos básicos junto a sus respectivas descripciones.

**Tabla 2.** Simbología y descripción de los elementos de programación

Símbolo	Nombre	Descripción
	Contacto NA	Se activa cuando hay un uno lógico en el elemento que representa, esto es, una entrada (para captar información del proceso a controlar), una variable interna o un bit de sistema.
	Contacto NC	Su función es similar al contacto NA anterior, pero en este caso se activa cuando hay un cero lógico, cosa que deberá de tenerse muy en cuenta a la hora de su utilización.
	Bobina NA	Se activa cuando la combinación que hay a su entrada (izquierda) da un uno lógico. Su activación equivale a decir que tiene un uno lógico. Suele representar elementos de salida, aunque a veces puede hacer el papel de variable interna.
	Bobina NC	Se activa cuando la combinación que hay a su entrada (izquierda) da un cero lógico. Su activación equivale a decir que tiene un cero lógico. Su comportamiento es complementario al de la bobina NA.
	Bobina SET	Una vez activa (puesta a 1) no se puede desactivar (puesta a 0) si no es por su correspondiente bobina en RESET. Sirve para memorizar bits y usada junto con la bina RESET dan una enorme potencia en la

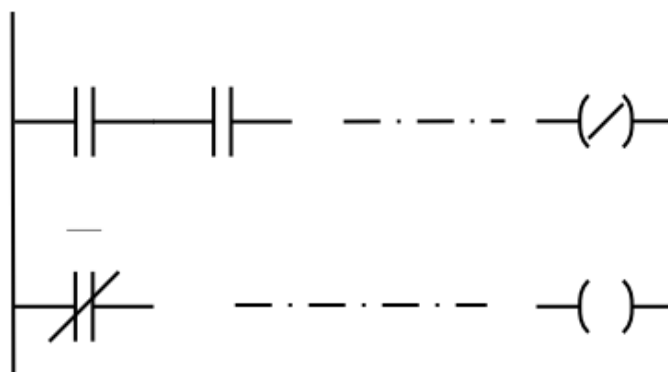
		programación.
	Bobina SET	Permite desactivar una bobina SET previamente activada.

Fuente: <http://www.abcinnova.com/articulos-e-informacion/18-ique-es-un-plc-y-que-beneficios-tiene.html>

- **Programación**

Una vez conocidos los elementos que LADDER proporciona para su programación, resulta importante resaltar cómo se estructura un programa y cuál es el orden de ejecución. (Little, 2002)

La Figura 11 representa la estructura general de la distribución de todo programa LADDER, contactos a la izquierda y bobinas y otros elementos a la derecha.



**Figura 11.** Estructura de distribución de todo programa LADDER

En cuanto a su equivalencia eléctrica, se puede imaginar que las líneas verticales representan las líneas de alimentación de un circuito de control eléctrico.

El orden de ejecución es generalmente de arriba hacia abajo y de izquierda a derecha, primero los contactos y luego las bobinas, de manera que al llegar a éstas ya se conoce

el valor de los contactos y se activan si procede. El orden de ejecución puede variar de un controlador a otro, pero siempre se respetará el orden de introducción del programa, de manera que se ejecuta primero lo que primero se introduce.

- **VARIABLES INTERNAS Y BITS DE SISTEMA**

Las variables internas son bits auxiliares que pueden ser usados según convenga, sin necesidad de que representen ningún elemento del autómeta. Se suele indicar mediante los caracteres B o M y tienen tanto bobinas como contactos asociados a las mismas. Su número de identificación suele oscilar, en general, entre 0 y 255. Su utilidad fundamental es la de almacenar información intermedia para simplificar esquemas y programación. (Little, 2002)

Los bits de sistema son contactos que el propio autómeta activa cuando conviene o cuando se dan unas circunstancias determinadas. Existe una gran variedad, siendo los más importantes los de arranque y los de reloj, que permiten que empiece la ejecución desde un sitio en concreto y formar una base de tiempos respectivamente. Su nomenclatura es muy diversa, dependiendo siempre del tipo de autómeta y fabricante.

#### ***B.2.6.6. Lenguaje booleano (Lista de Instrucciones)***

El lenguaje Booleano utiliza la sintaxis del Álgebra de Boole para ingresar y explicar la lógica de control. Consiste en elaborar una lista de instrucciones o nemónicos, haciendo uso de operadores Booleanos (AND, OR, NOT, etc.) y otras instrucciones nemónicas, para implementar el circuito de control. El lenguaje “Lista de Instrucciones” (IL) de la Norma IEC 1131-3, es una forma de lenguaje Booleano. (Little, 2002)

**Tabla 3.** Ejemplo de programación Booleana (entradas y salidas)

A	I	2.3
A	I	4.1
O	I	3.2
=	Q	1.6

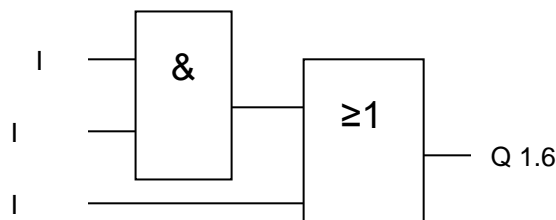
Fuente: <http://recursostic.educacion.es/observatorio/web/gl/component/content/article/502-monografico-lenguajes-de-programacion?start=2>

### **B.2.6.7. Diagrama de funciones (FBD)**

Es un lenguaje gráfico que permite al usuario programar elementos (bloque de funciones del PLC) en tal forma que ellos aparecen interconectados al igual que un circuito eléctrico. Generalmente utilizan símbolos lógicos para representar al bloque de función (Figura 12). Las salidas lógicas no requieren incorporar una bobina de salida, porque la salida es representada por una variable asignada a la salida del bloque.

El diagrama de funciones lógicas, resulta especialmente cómodo de utilizar, a técnicos habituados a trabajar con circuitos de puertas lógicas, ya que la simbología usada en ambos es equivalente.

Adicionalmente a las funciones lógicas estándares y específicas del vendedor, el lenguaje FBD de la Norma IEC 1131-3 permite al usuario construir sus propios bloques de funciones, de acuerdo a los requerimientos del programa de control.



**Figura 12.** Ejemplo de programación utilizando diagramas de funciones

#### ***B.2.6.8. Lenguaje de texto estructurado (ST)***

Texto estructurado (ST) es un lenguaje de alto nivel que permite la programación estructurada, lo que significa que muchas tareas complejas pueden ser divididas en unidades más pequeñas. ST se parece mucho a los lenguajes de computadoras BASIC o PASCAL, que usa subrutinas para llevar a cabo diferentes partes de las funciones de control y paso de parámetros y valores entre las diferentes secciones del programa.

Al igual que LD, FBD e IL, el lenguaje de texto estructurado utiliza la definición de variables para identificar entradas y salidas de dispositivos de campo y cualquier otra variable creada internamente. Incluye estructuras de cálculo repetitivo y condicional, tales como: FOR...TO; REPEAT..... UNTIL X; WHILE X... ; IF ... THEN ...ELSE. Además soporta operaciones Booleanas (AND, OR, etc.) y datos específicos como fecha y hora. (Morriss Brian, 2000)

La programación en Texto Estructurado es apropiada para aplicaciones que involucran manipulación de datos, ordenamiento computacional y aplicaciones matemáticas que utilizan valores de punto flotante. ST es el mejor lenguaje para la implementación de aplicaciones de inteligencia artificial, lógica difusa, toma de decisiones, etc. Figura 13.

```
IF Manual AND Alarm THEN
    Level = Manual_Level;
    Mixer = Start AND NOT Reset
ELSE IF Other_Mode THEN
    Level = Max_level;
ELSE
    Level = (Level_Indic X100)/Scale;
END IF;
```

***Figura 13. Ejemplo de programación con Lenguaje Estructurado***

### ***B.2.6.9. Sequential function chart (SFC)***

Es un “lenguaje” gráfico que provee una representación diagramática de secuencias de control en un programa. Básicamente, SFC es similar a un diagrama de flujo, en el que se puede organizar los subprogramas o subrutinas (programadas en LD, FBD, IL y/o ST) que forman el programa de control. (Morriss Brian, 2000)

SFC es particularmente útil para operaciones de control secuencial, donde un programa fluye de un punto a otro una vez que una condición ha sido satisfecha (cierta o falsa).

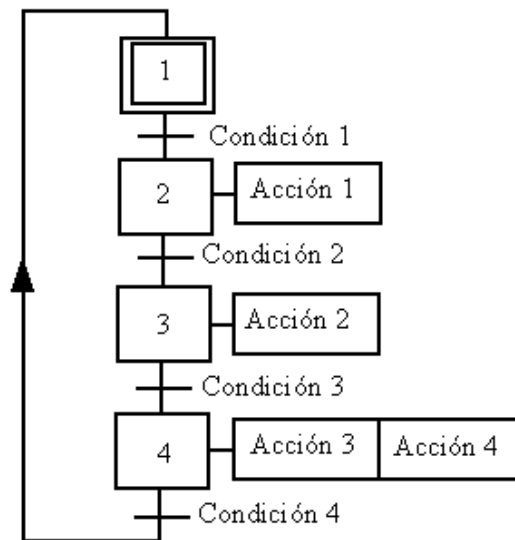
El marco de programación de SFC contiene tres principales elementos que organizan el programa de control:

- Pasos (etapas)
- Transiciones (condiciones)
- Acciones

El programa irá activando cada una de las etapas y desactivando la anterior conforme se vayan cumpliendo cada una de las condiciones. Las acciones se realizarán en función de la etapa activa a la que están asociadas.

Por ejemplo, la etapa 1 activa tras arrancar el programa, al cumplirse la "Condición 1", se activará la etapa 2, se desactivará la 1, y se realizará la "Acción 1". Figura 14.





*Figura 14. Ejemplo de programación con diagramas de flujo SFC*

Como se mencionó anteriormente, el lenguaje SFC tiene su origen en el estándar francés GRAFCET (Grafica de Control de Etapas de Transición). El grafcet también utiliza etapas, transiciones y acciones, que operan de la misma manera como en SFC.

### **B.3. LOGO SOFT**

LOGO es el módulo lógico universal de los PLC Siemens y lleva integrados:

- Control
- Unidad de operación y visualización
- Fuente de alimentación
- Interface para módulos de programa y cable de PC
- Ciertas funciones básicas usuales en la práctica, por ejemplo para activación/desactivación retardada y relé de impulsos
- Reloj temporizador

- Marcas binarias
- Determinadas entradas y salidas según el tipo del equipo

Con LOGO se solucionan cometidos en la técnica de instalaciones en edificios (por ejemplo alumbrado de escaleras, luz exterior, toldos, persianas, alumbrado de escaparates, etc.), así como en la construcción de armarios de distribución, de máquinas y de aparatos (por ejemplo controles de puertas, instalaciones de ventilación, bombas de aguas residuales, etc.).

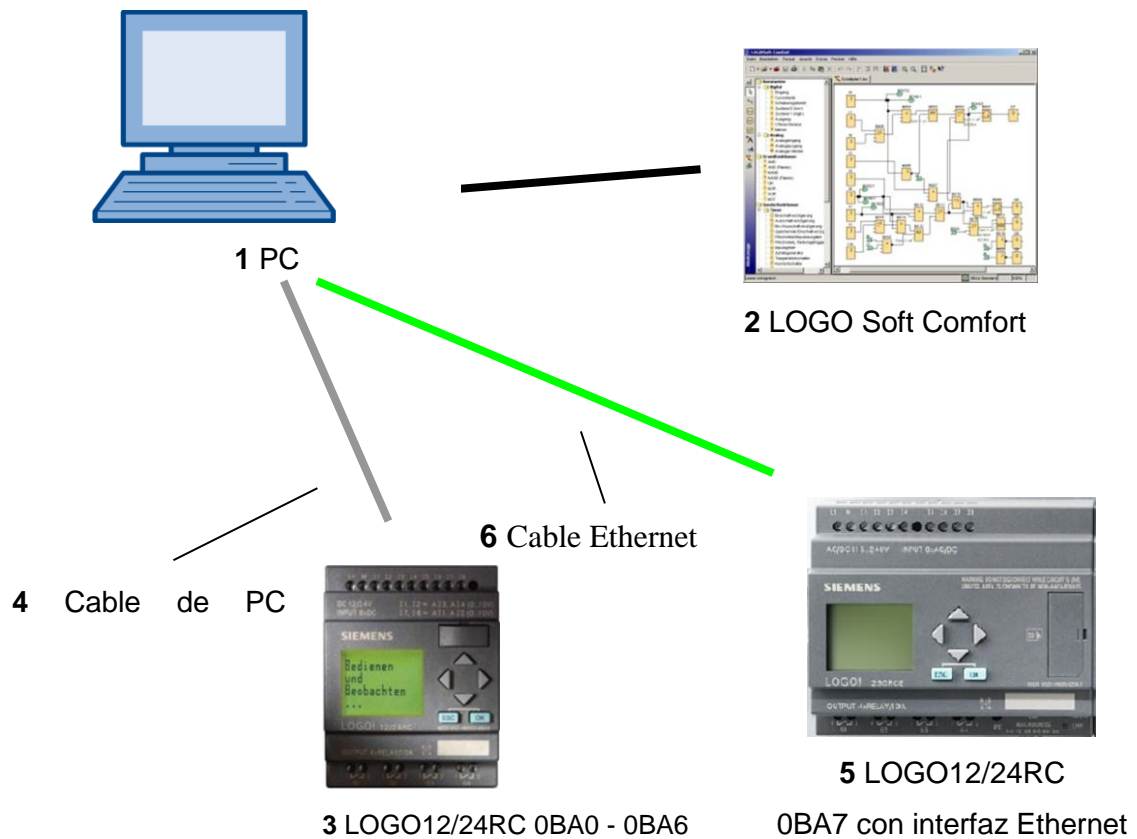
LOGO puede utilizarse asimismo para los controles especiales de invernaderos o invernáculos, para procesar previamente señales en controles y mediante la variante ASI- para el control descentralizado "in situ" de máquinas y procesos.

Para las aplicaciones en serie en la construcción de máquinas pequeñas, aparatos y armarios de distribución, así como en el sector de instalaciones, se prevén variantes especiales sin unidad de operación.

Para trabajar adecuadamente con este módulo se requieren que el usuario tenga conocimientos medios en el manejo de Windows.

### **B.3.1. Hardware y software necesarios**

Los artefactos que intervienen en el proceso de programación de una tarea en un PLC se aprecian en la Figura 15. En la que se aprecia la forma en la que interactúa un PLC con una PC.



**Figura 15.** Estructura de conexión de un PLC y Logo Soft

- 1 PC a partir de Pentium III, sistema operativo a partir de Windows 7, tarjeta de red
- 2 Software LOGO Soft Comfort Versión 7.1
- 3 Módulo lógico LOGO 0BA0 – 0BA6 (optativo)
- 4 Cable de PC a LOGO (optativo)
- 5 Módulo lógico LOGO 0BA7 con interfaz Ethernet (optativo)
- 6 Cable Ethernet (optativo)

### B.3.2. Tipos de equipos

Se prevén las siguientes variantes de LOGO tanto para 12 V y 24 V como para 230 V:

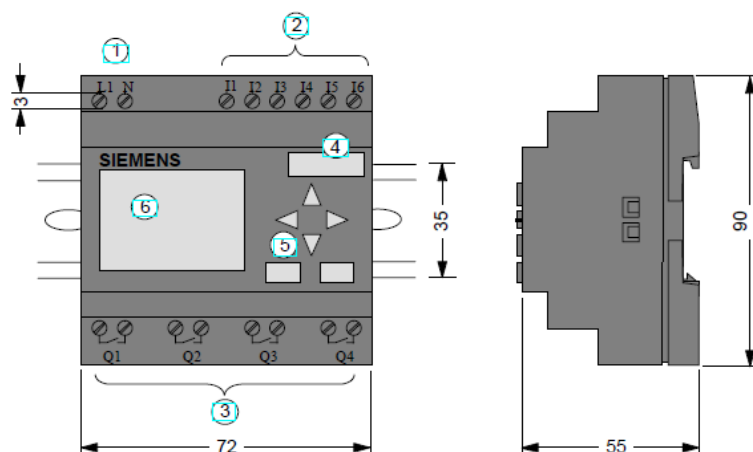
- Variante estándar con 6 entradas y 4 salidas, integrada en 72 x 90 x 55 mm
- Variante sin display con 6 entradas y 4 salidas, integrada en 72 x 90 x 55 mm
- Variante larga con 12 entradas y 8 salidas, integrada en 126 x 90 x 55 mm
- Variante de bus con 12 entradas y 8 salidas, así como conexión de bus adicional de interface AS, a través de la que hay disponibles en el sistema bus otras 4 entradas y otras 4 salidas. Todo ello integrado en 126 x 90 x 55 mm.

Todas las variantes incluyen 19 funciones básicas y especiales listas para la redacción de programas.

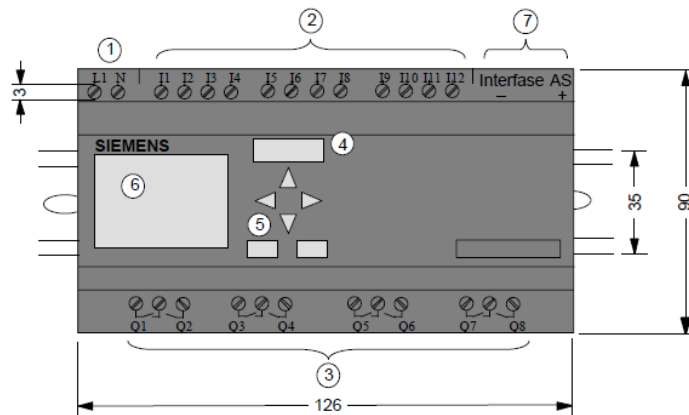
Las distintas variantes permiten la adaptación sumamente flexible a su aplicación especial.

LOGO ofrece soluciones que abarcan desde la pequeña instalación doméstica, pasando por cometidos de automatización menores, hasta las aplicaciones de gran envergadura con implementación del sistema bus de interface AS.

### B.3.3. Estructura de LOGO



*Figura 16. Versión estándar LOGO*



**Figura 17. LOGO...L/LOGO...LB11**

En base a la estructura de los 2 modelos de PLC siemens que se aprecia en la Figura 16 y Figura 17, a continuación se describe la nomenclatura establecida:

1. Alimentación
2. Entradas y Salidas
3. Receptáculo de módulo con revestimiento
4. Panel de manejo (no en RCo)
5. Display LCD (no en RCo)
6. Conexión de interface AS (sólo en LB11)

#### **B.3.4. Identificación de LOGO**

De la identificación de LOGO se deducen diferentes características del mismo:

- 12: Versión de 12 V, 24: Versión de 24 V, y 230: Versión de 115/230 V
- R: Salidas de relé (sin R: salidas de transistor)
- C: Reloj de temporización semanal integrado
- o: Variante sin display
- L: Cantidad doble de salidas y entradas

- B11: Esclavo con conexión de bus de interface AS





En el transcurso restante de esta descripción se utilizan pequeños pictogramas para identificar los tipos con funciones diferentes. Estos pictogramas aparecen cuando las informaciones se refieren sólo a una parte de las variantes de LOGO:

- Variante estándar con 6 entradas y 4 salidas, integrada en 72 x 90 x 55 mm
- Variante estándar sin display con 6 entradas y 4 salidas, integrada en 72 x 90 x 55 mm
- Variante ".L" con 12 entradas y 8 salidas, integrada en 126 x 90 x 55 mm
- Variante "B11" con 12 entradas y 8 salidas, así como conexión de bus adicional de interface AS, con 4 entradas virtuales y 4 salidas virtuales, integrada en 126 x 90 x 55 mm.

### B.3.5. Variantes de LOGO

En la Tabla 4 se describen las variantes previstas de LOGO.

*Tabla 4. Modelos de PLC Logo*

Símbolo	Designación	Salidas	Tipo
	LOGO! 12RC	4 * 230 V; 10 A	relé
	LOGO! 24	4 * 24 V; 0,3 A	transistor
	LOGO! 24RC	4 * 230 V; 10 A	relé
	LOGO! 230RC	4 * 230 V; 10 A	relé
	LOGO! 24RCo	4 * 230 V; 10 A	relé
	LOGO! 230RCo	4 * 230 V; 10 A	relé
	LOGO! 12RCL	8 * 230 V; 10 A	relé
	LOGO! 24L	8 * 24 V; 0,3 A	transistor
	LOGO! 24RCL	8 * 230 V; 10 A	relé
	LOGO! 230RCL	8 * 230 V; 10 A	relé
	LOGO! 24RCLB11	8 * 230 V; 10 A	relé
	LOGO! 230RCLB11	8 * 230 V; 10 A	relé

Fuente: <http://ocw.upc.edu/sites/default/files/materials/15012628/40197-3452.pdf>

### **B.3.6. Montaje y cableado de LOGO**

El montaje y el desmontaje se muestran a base de un gráfico de LOGO 230RC. Las medidas que deben adoptarse rigen análogamente también para los demás módulos LOGO.

LOGO debe montarse en una caja o un armario de distribución. Tras el montaje, los bornes deben quedar cubiertos para impedir con certeza que se toquen por descuido piezas de LOGO bajo tensión.

LOGO sólo puede ser montado y cableado por un especialista cualificado que conozca y observe las reglas generales de la técnica, así como las prescripciones y normas vigentes en cada caso.

LOGO tiene las dimensiones para equipos de instalación estipuladas en DIN 43880.

LOGO debe encajarse en un perfil soporte de 35 mm de ancho según DIN EN 50022.

Anchura de LOGO:

- LOGO: 72 mm de ancho, equivalente a 4 unidades de división (versión estándar).
- LOGO...RCO: 72 mm de ancho, equivalente a 4 unidades de división (versión estándar).
- LOGO...L: 126 mm de ancho, equivalente a 7 unidades de división.
- LOGO...B11: 126 mm de ancho, equivalente a 7 unidades de división.

#### ***B.3.6.1. Cablear LOGO***

Para cablear LOGO, utilice un destornillador con ancho de hoja de 3 mm.

Para los bornes no se requieren casquillos terminales, pudiendo utilizarse conductores con secciones de hasta:

- 1 x 2,5 mm<sup>2</sup>
- 2 x 1,5 mm<sup>2</sup> por cada segundo

Pares de giro para la conexión: 0,4...0,5 Nm ó 3...4 LBin

### ***B.3.6.2. Conectar la alimentación***

Las variantes de LOGO 230 son adecuadas para tensiones de red con valor nominal de 115 V AC ó 230 V AC y las variantes de LOGO 24 y 12 para tensiones de alimentación de 24 V c.c. ó 12 V c.c. LOGO 24 RC/RCo es apropiado además para la conexión a 24 V AC. Las tolerancias de tensión, frecuencias de red y consumos admisibles se especifican en los datos técnicos.

## **B.4. PROGRAMACIÓN DE LOGO SIEMENS**

Por programación se entiende aquí la introducción de un circuito. Un programa LOGO equivale sencillamente a un esquema de circuitos, pero representado de manera algo diferente.

En LOGO el controlador está integrado con la unidad de manejo y visualización. Con la unidad de manejo y visualización de LOGO se pueden crear y editar programas, así como ejecutar funciones de sistema.

A través de una interfaz pueden leerse programas externos de un módulo de programación o bien a través de un cable de PC, desde el software de programación LOGO-SOFT. Con LOGO-SOFT también es posible realizar una simulación del



circuito en el ordenador, además de crear programas, y también imprimir esquemas generales.

En los módulos lógicos LOGO ya están integradas las funciones básicas habituales en la práctica, listas para su uso, como por ejemplo, la conexión retardada, la desconexión retardada, relés de corriente, programadores horarios, marcas binarias, así como entradas y salidas dependiendo del tipo de dispositivo.

LOGO permite resolver tareas:

- En las instalaciones en viviendas y edificios (por ejemplo, iluminación de escaleras, luces exteriores, marquesinas, persianas, iluminación de escaparates, entre otros).
- En la construcción de armarios eléctricos y el montaje de máquinas y aparatos (por ejemplo, controles de portón, instalaciones de ventilación, bombas de aguas industriales, entre otros).

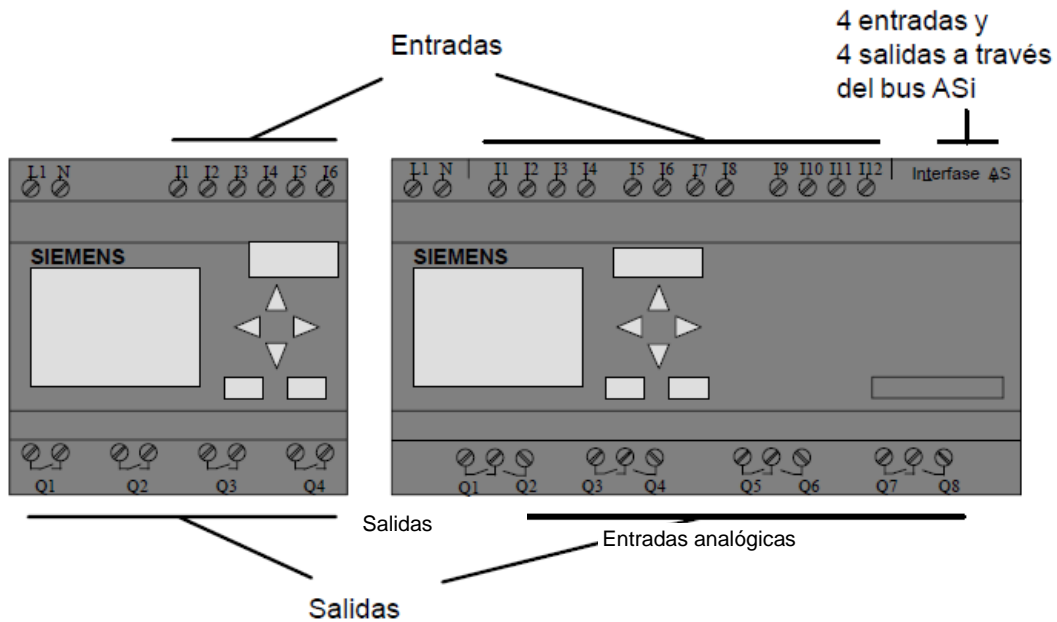
Además, LOGO se puede utilizar para controladores especiales para el procesamiento de señales. Gracias a la conexión a Así-Interface, es posible el uso como periferia descentralizada con inteligencia propia local para el control de máquinas y procesos. De este modo se pueden ejecutar tareas de control en el módulo lógico LOGO y descargar así al controlador maestro.

Para las aplicaciones en serie en la construcción de máquinas pequeñas, aparatos y armarios eléctricos, así como en el sector de instalaciones, existen variantes especiales sin unidad de mando. Estas deben cargarse seguidamente a través de un módulo de programación o a través del software de PC LOGO-SOFT.

### B.4.1. Bornes

LOGO dispone de entradas y salidas:

Las entradas se designan con la letra I y una cifra. Si mira hacia LOGO desde el frente, verá en la parte superior los bornes para las entradas. Las entradas analógicas solamente se encuentran en la parte inferior en los módulos analógicos LOGO AM2 y AM2 PT100. Las salidas se han identificado con una Q y una cifra. Los bornes de las salidas se ven en la figurab.18.








**Figura 18.** Descripción de partes de un LOGO

Las entradas y salidas pueden tener el estado '0' o el estado '1'. El estado '0' significa que en la entrada no hay tensión aplicada, y el estado '1' significa que hay tensión aplicada. Los bornes hi, lo y x se han previsto para facilitarle la introducción de programas. 'hi' (high) tiene asignado el estado '1' de forma fija. 'lo' (low) tiene asignado el estado '0' de

forma fija. Si no desea conectar una entrada de un bloque, utilice a continuación el borne 'x'.

#### B.4.1.1. LOGO reconoce los siguientes bornes

Tabla 5. Descripción de bornes de LOGO

Bornes	 	MD 	MA 	AM2AQ 	
Entradas	LOGO 230RCE LOGO 230RC/RCo LOGO 24RC/RCo	Dos grupos: I1 hasta I4 e I5 hasta I8	I9 hasta I24	AI1 hasta AI8	Ninguna
	LOGO 12/24RCE LOGO 12/24RC/RCo LOGO 24/24o LOGO 24C/24Co	I1, I2, I3-I6, I7, I8 AI3, AI4 ... AI1, AI2	I9 hasta I24	AI5 hasta AI8	
Salidas	Q1 hasta Q4		Q5 a Q16	Ninguna	AQ1, AQ2
Lo	Señal con nivel '0' (desc.)				
Hi	Señal con nivel '1' (con.)				
X	Terminal existente no utilizado				
Marca	Marcas digitales: M1 hasta M27 Marcas analógicas: AM1 hasta AM6 (0BA6) AM1 hasta AM16 (0BA7)				
Bits de registro de desplazamiento	S1 hasta S8 (0BA6) S1.1 hasta S4.8 (0BA7)				
Entradas de red <sup>1)</sup>	NI1 hasta NI64 (solo 0BA7)				
Entradas de red analógicas <sup>1)</sup>	NAI1 hasta NAI32 (solo 0BA7)				
Salidas de red <sup>1)</sup>	NQ1 hasta NQ64 (solo 0BA7)				
Salidas de red analógicas <sup>1)</sup>	NAQ1 hasta NAQ16 (solo 0BA7)				

Fuente: <http://ocw.upc.edu/sites/default/files/materials/15012628/40197-3452.pdf>

#### B.4.2. Bloques y números de bloque

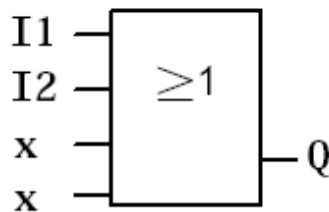
Se presentará cómo crear circuitos con los elementos de LOGO y cómo interconectar los bloques entre sí y con las entradas y salidas.

#### ***B.4.2.1. Bloques***

Un bloque en LOGO es una función que convierte información de entrada en información de salida. Antes tenía Ud. que cablear los distintos elementos en el armario de distribución o en la caja de conexiones. Durante la programación, los bornes se conectan con bloques. A tal efecto, basta con elegir la conexión deseada en el menú **Co**. El menú **Co** debe su nombre al término inglés "Connector" (borne).

#### ***B.4.2.2. Funciones lógicas***

Los bloques más sencillos son funciones lógicas (por ejemplo: AND, OR).

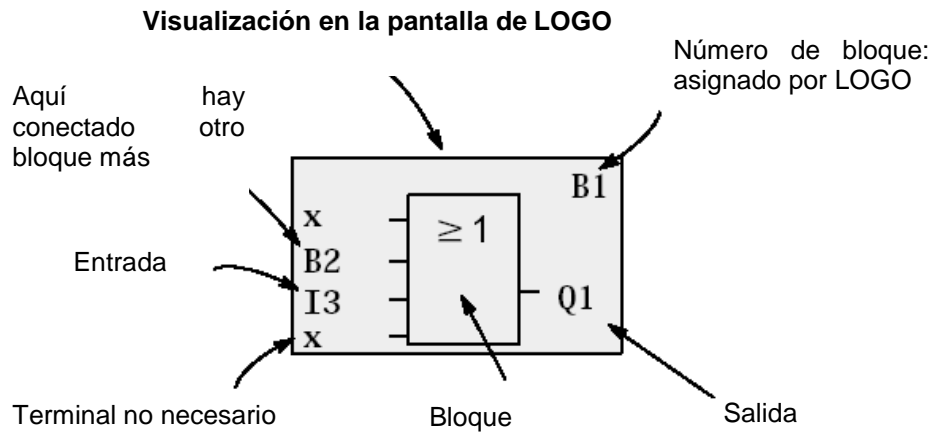


***Figura 19. Bloque de conexión OR.***

En la Figura 19, están conectadas las entradas I1 e I2, al bloque OR. Las dos últimas entradas del bloque no se utilizan y por tanto se les asigna x.

#### ***B.4.2.3. Representación de un bloque en la pantalla de LOGO***

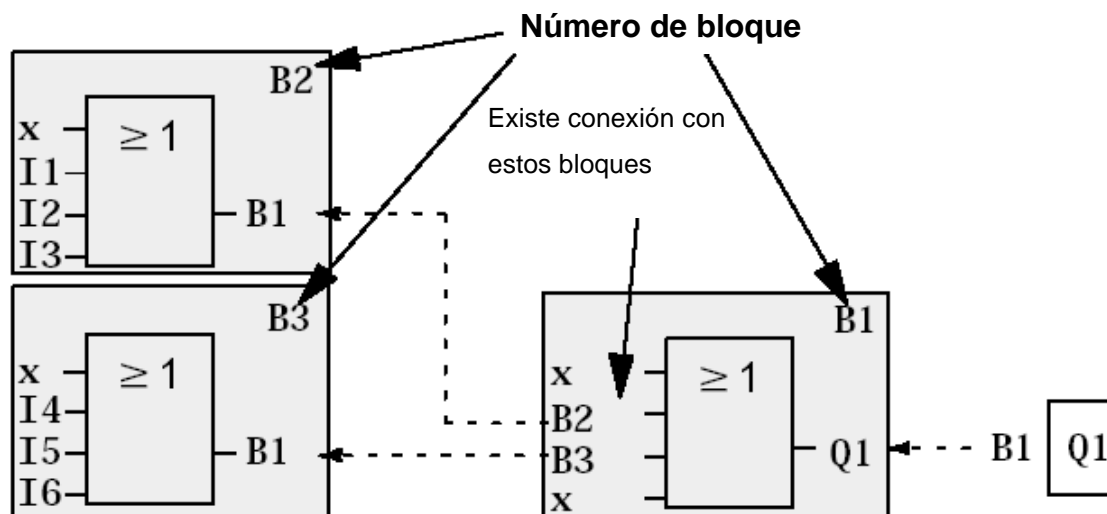
En la Figura 20 se muestra una pantalla típica de LOGO. Sólo puede representarse un bloque en cada caso. Debido a ello, hemos previsto números de bloque para ayudarle a controlar un circuito en conjunto.



*Figura 20. Descripción de un bloque de LOGO*

#### **B.4.2.4. Asignación de número de bloque**

Siempre que inserte un bloque en un programa, LOGO asigna a este bloque un número. Mediante el número de bloque, LOGO le muestra la conexión entre los bloques. Los números de bloque sólo pretenden facilitar su orientación en el programa, éste hecho se describe en la Figura 21.



*Figura 21. Partes de un bloque*

En el esquema general del circuito puede ver tres pantallas de LOGO que componen el programa. Como puede ver, LOGO conecta los bloques entre sí a partir de los números de bloque.

### **B.4.3. Las 4 reglas de oro para manejar LOGO**

#### ***B.4.3.1. Regla 1 - Cambio del modo de operación***

El programa se elabora en el modo de operación **Programación**.

Tras conectar la alimentación y aparecer “**No Program / Press ESC (No hay programa/Presione ESC)**” en la pantalla, debe pulsar la tecla **ESC** para acceder al modo de operación **Programación**.

La modificación de los valores de tiempo y de parámetros en un programa ya existente puede realizarse en los modos de operación **Parametrización** y **Programación**.

Durante la parametrización, LOGO se encuentra en el **modo RUN**, es decir, el programa continúa procesándose. Para programar deberá finalizar la ejecución del programa con el comando "**Stop**". Para acceder al **modo RUN**, elija el comando de menú '**Start (Inicio)**' del menú principal.

En el **modo RUN**, para regresar al modo de operación **Parametrización**, deberá pulsar la tecla **ESC**. Si está en el modo de operación **Parametrización** y desea regresar al modo de operación **Programación**, elija el comando "**Stop**" del menú **Parametrización** y responda con "**Yes (Sí)**" a "**Stop Prg (Parar programa)**", colocando el cursor sobre “**Yes (Sí)**” y pulsando la tecla **OK (Aceptar)**.

#### ***B.4.3.2. Regla 2 - Salidas y entradas***

**Introduzca los circuitos siempre desde la salida hasta la entrada.**

Es posible enlazar una salida con varias entradas, pero no conectar varias salidas a una entrada. Dentro de un circuito del programa no se puede enlazar una salida con una entrada precedente. Para tales retroacciones internas es necesario intercalar marcas o salidas.

#### ***B.4.3.3. Regla 3 - Cursor y posicionamiento del cursor***

Para la introducción de un circuito es válido lo siguiente:

Si el cursor aparece subrayado, se puede mover.

- con las teclas  $\Leftarrow$ ,  $\Rightarrow$ ,  $\Uparrow$  o  $\Downarrow$  se mueve el cursor en el circuito
- con **OK (Aceptar)** se cambia a "Seleccionar borne/bloque"
- con **ESC** se sale del modo de introducción del circuito

Si el cursor aparece enmarcado, deberá elegir un borne/bloque.

- con las teclas  $\Uparrow$  o  $\Downarrow$  se selecciona un borne/un bloque
- con **OK (Aceptar)** se acepta la selección
- con **ESC** se retrocede un paso

#### ***B.4.3.4. Regla 4 - Planificación***

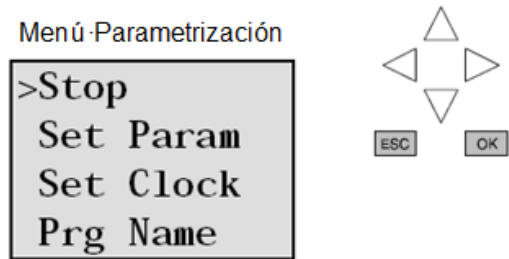
Antes de introducir un circuito, diseñelo completamente sobre el papel o prográmelo en LOGO directamente con LOGO Soft o LOGO Soft Comfort.

LOGO solamente puede guardar programas completos.

Si se ha introducido un circuito de forma incompleta, LOGO no podrá salir del modo de operación **Programación**.

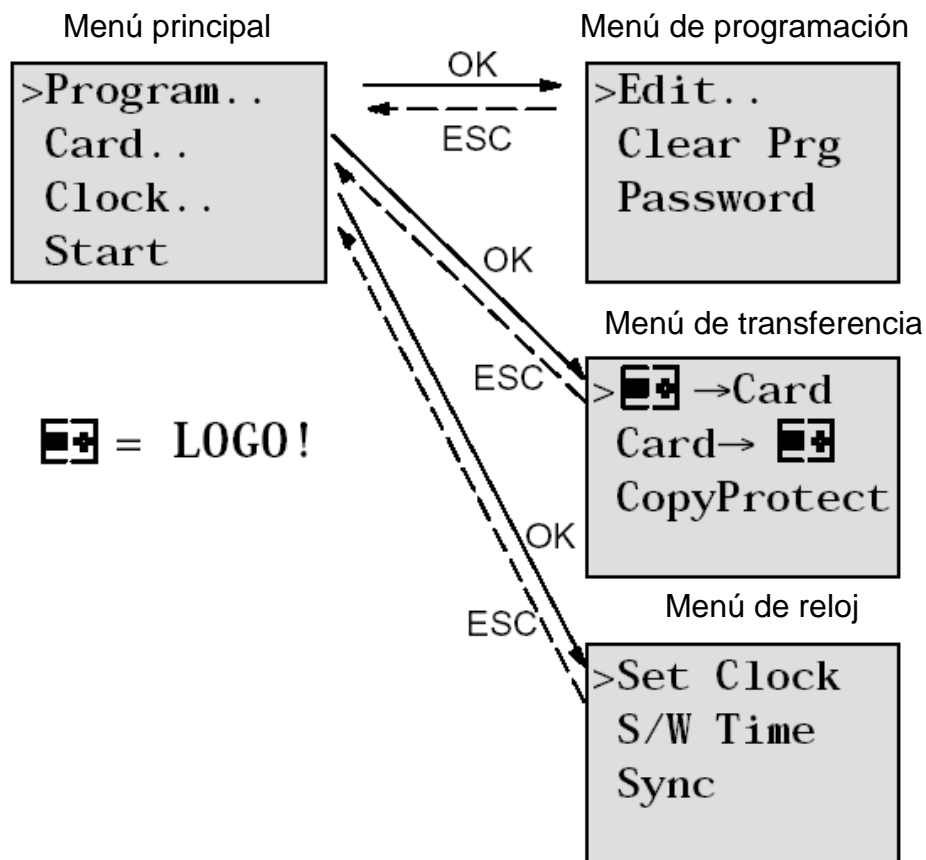
#### B.4.4. Vista de conjunto de los menús de LOGO

**Modo de operación Parametrización.** Figura 22.



*Figura 22. Menú para parametrizar un LOGO*

**Modo de operación Programación.** Figura 23.



*Figura 23. Menú de opciones para el menu de parametrización*

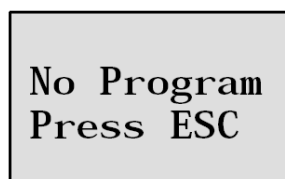


### B.4.5. Introducción del programa en Logo

A continuación se cómo hacerlo con un ejemplo de introducción de programa.

#### B.4.5.1. Pasar al modo de operación "Programación"

Cuando se ha conectado LOGO a la red eléctrica y ha aplicado la tensión. En la pantalla se muestra la indicación que muestra la Figura 24.



*Figura 24. Mensaje de inicio de un PLC sin programar*

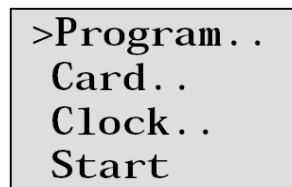
- Pase LOGO al modo de operación Programación.
  - Para ello, pulse la tecla ESC. A continuación pasará al menú principal de LOGO.

#### B.4.5.2. Menú principal de LOGO

En el primer lugar de la primera línea se muestra el carácter ">". Figura 25.

Con las teclas de cursor (↑, ↓) se mueve el carácter ">" arriba y abajo.

Mueva el signo ">" hasta "Program... (Programa...)" y pulse la tecla **OK (Aceptar)**.

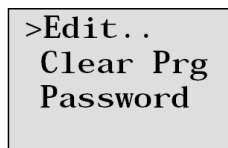


*Figura 25. Pantalla del PLC de configuración e inicio*

#### **B.4.5.3. Menú Programación de LOGO:**

Aquí también se puede mover el carácter ">" con las teclas de cursor (↑, ↓). Coloque el signo ">" en "**Edit... (Editar)**" (para introducir texto) y pulse la tecla **OK (Aceptar)**.

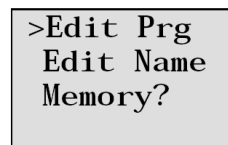
Figura 26.



*Figura 26. Pantalla del PLC para ingresar programación*

#### **B.4.5.4. Menú Edición de LOGO:**

Coloque el signo ">" en "**Edit Prg (Editar programa)**" (para editar el programa) y pulse la tecla **OK (Aceptar)**. Figura 27.



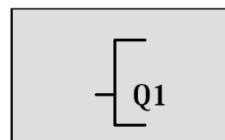
*Figura 27. Pantalla del PLC para editar y nombrar un programa*

#### **B.4.5.5. Modo de programación de LOGO:**

LOGO le mostrará la primera salida (Figura 28.):

Con las teclas de cursor (↑, ↓) se pueden seleccionar las demás salidas.

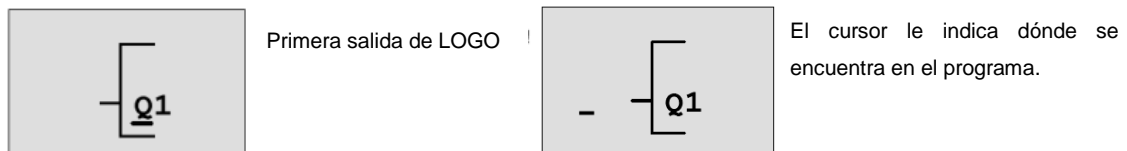
Ahora puede introducir su circuito.



*Figura 28. Pantalla del PLC para inicio de programación por bloques*

#### B.4.5.6. Introducir el programa

Se introduce ahora el programa (desde la salida hasta la entrada). Al principio LOGO muestra la salida Q1:

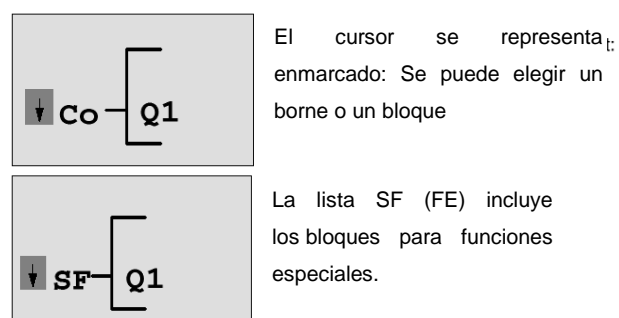


*Figura 29. Pantalla para programar salida Q1*

La letra Q de Q1 está subrayada (Figura 29). El guión bajo se denomina **cursor**. En el programa, el cursor indica el lugar en el que se encuentra en ese momento. El cursor se puede desplazar mediante las teclas de cursor. Ahora pulse la tecla **Cursor izquierda**.

Figura 30.

Introduzca aquí solo el primer bloque. Pase al modo de introducción pulsando la tecla **OK (Aceptar)**.

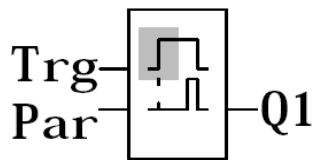


*Figura 30. Opciones de Funciones a programar que interactúan con Q1*

El cursor ya no es del tipo subrayado, sino que está enmarcado y parpadea. Al mismo tiempo, LOGO ofrece diferentes posibilidades de selección. Seleccione SF (pulsar la

tecla **Cursor abajo** hasta que aparezca SF (FE)), y después la tecla **OK (Aceptar)**.

Ahora LOGO le muestra el primer bloque de la lista de funciones especiales (SF (FE)):



Si selecciona un bloque para una función básica o una función especial, LOGO muestra el bloque de la función. El cursor se halla en el bloque y está enmarcado. Con las teclas ▼ y ▲ puede seleccionar el bloque que desee.

*Figura 31. Retardo a la conexión (On-Line) para salida Q1*

Confirme su elección con **OK (Aceptar)**.

El bloque para el retardo a la conexión posee 2 entradas. La entrada superior es la entrada de disparo (Trg). A través de dicha entrada se inicia el retardo a la conexión. En nuestro ejemplo, el retardo a la conexión se inicia desde el bloque OR B2. A través de la entrada de parametrización (Par) se ajusta el tiempo para el retardo a la conexión.

Figura 31.

## **B.5. LOGO SOFT COMFORT**

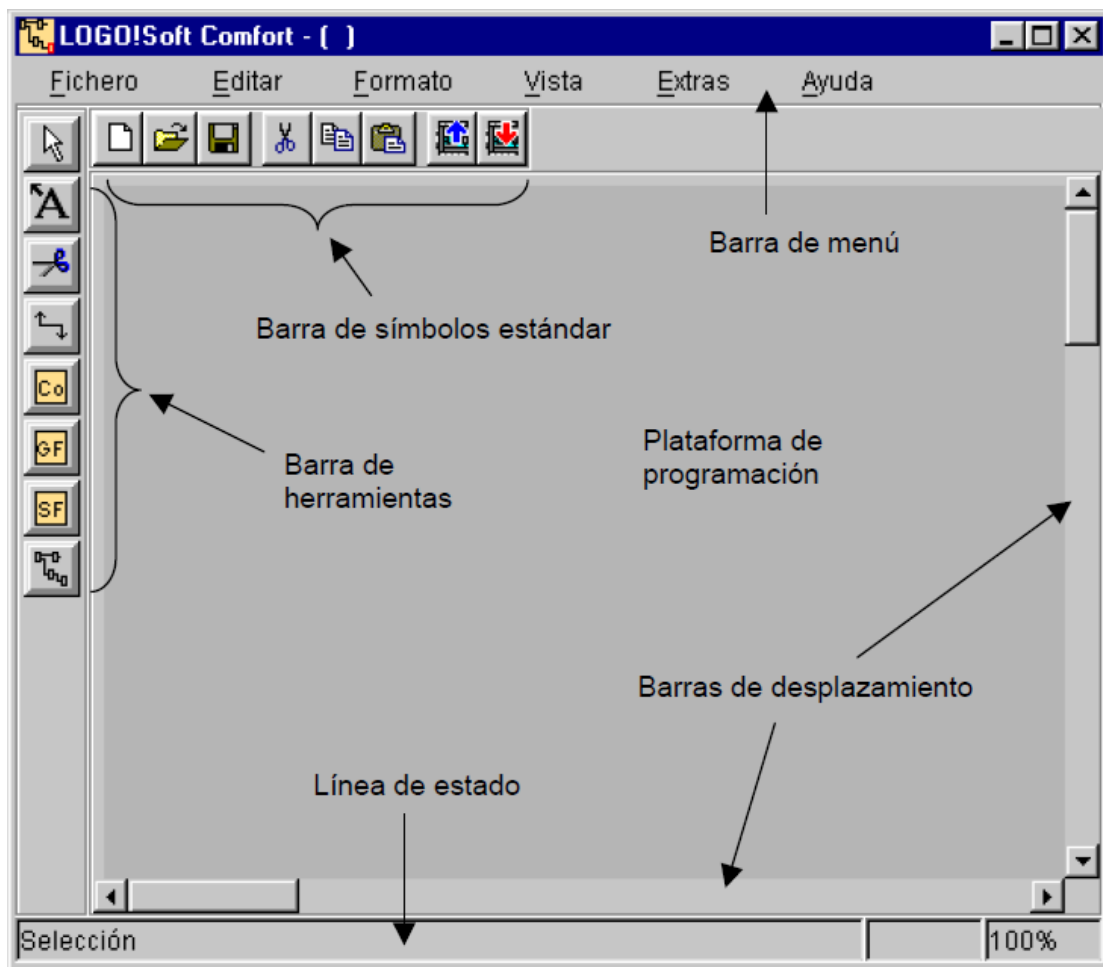
Logo Soft Comfort proporciona al usuario de Windows una herramienta que le permite elaborar con más sencillez y confort en su PC programas de conmutación de forma más eficiente, confortable y clara que hasta ahora; así como imprimir con detalle y claridad la documentación para los elementos de los circuitos. La elaboración del programa tiene lugar colocando los elementos de programación libremente en una plataforma y uniéndolos entre sí.

En particular facilitan la labor del usuario, entre otras cosas, la simulación off line del programa, que posibilita la indicación simultánea del estado de varias funciones

especiales, así como la documentación con calidad profesional de los programas de conmutación elaborados. Entre las peculiaridades se tiene:

- Elaboración muy sencilla y confortable de programas de conexiones
- Extensa documentación de programas de conexiones
- Amplias funciones de impresión
- Mínima tasa de errores en programas de conexiones gracias a su simulación en el PC.

### B.5.1. Descripción de la pantalla de operación



*Figura 32. Plataforma de programación de Logo Soft Comfort*

#### ***B.5.1.1. Plataforma de programación***

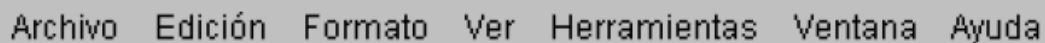
Una vez arrancado el programa verán la plataforma de operación de LOGO Soft Comfort. La mayor parte de la pantalla la ocupa entonces el área dedicada a la elaboración de esquemas de conexiones. En esta plataforma de programación se disponen los símbolos y enlaces del programa de conexiones.

#### ***B.5.1.2. Elementos de la pantalla de operación***

Para que tampoco con circuitos extensos se pierda una visión general, en la parte inferior y la derecha de la plataforma de programación están dispuestas unas reglas deslizables, que permiten desplazar la imagen de conexiones en dirección horizontal y vertical.

#### ***B.5.1.3. Barra de menús***

Encima está situada la barra de menús. En la barra de menús encontrarán los más diversos comandos para la elaboración y la administración de programas de conexiones. Esto incluye también configuraciones y funciones de transferencia de programas.



Archivo Edición Formato Ver Herramientas Ventana Ayuda

***Figura 33. Barra de menús de Logo Soft Comfort***

#### ***B.5.1.4. Barra de Herramientas***

A la izquierda se encuentra la barra de herramientas. Mediante los botones de mando dispuestos en ésta se puede cambiar a diferentes modos de procesado, para elaborar o procesar un programa con rapidez y sencillez, entre las que se tiene:

- La barra de herramientas estándar ,

- La barra de herramientas Herramientas.
- La barra de herramientas de simulación.

#### ***B.5.1.5. Barra de símbolos estándar***

Encima del entorno de programación se encuentra la barra de herramientas Estándar. Al iniciar Logo Soft Comfort por primera vez, aparece una barra de herramientas estándar reducida a las funciones esenciales.

La barra de herramientas *Estándar* le proporciona acceso directo a las principales funciones de Logo Soft Comfort.

La barra de herramientas Estándar completa aparece cuando se accede a un programa para su edición en el entorno de programación.



***Figura 34. Barra de herramientas estándar***

Mediante los botones de esta barra se pueden realizar, entre otras, las siguientes operaciones: crear un programa; cargar, guardar o imprimir un programa ya existente; cortar, copiar y pegar objetos de un circuito o transferir datos desde y hacia Logo. La barra de herramientas Estándar se puede seleccionar y mover con el ratón. Si cierra la barra de herramientas Estándar, ésta permanecerá siempre acoplada a la barra de menús.

#### ***B.5.1.6. Barra de herramientas Herramientas***

Debajo se encuentra la barra de herramientas *Herramientas*. Mediante los botones dispuestos en ésta se puede cambiar a diferentes modos de operación, para crear o procesar un programa rápida y fácilmente.



*Figura 35. Barra de herramientas Herramientas*

La barra de herramientas Herramientas se puede seleccionar y mover con el ratón. Si cierra esta barra de herramientas, ésta permanecerá siempre acoplada sobre la barra de menús.

#### ***B.1.1.1. Línea de estado***

En el borde inferior de la ventana de programación se halla una línea de estado. En ésta se dan algunas indicaciones sobre la herramienta en actividad, la situación de programa y el factor zoom actual.

### **B.1.2. Elaboración de programa**

#### ***B.1.2.1. Introducción de un nuevo programa***

Justo cuando hayan iniciado una sesión de Logo Soft Comfort, pueden comenzar inmediatamente con la elaboración de un nuevo programa.

Si ya han trabajado con Logo Soft Comfort y todavía se encuentran partes antiguas de circuitos sobre su plataforma de programación, hagan en la barra de símbolos estándar clic sobre el botón de mando “Fichero nuevo” para lograr un nuevo programa de conexiones.



*Figura 36. Fichero nuevo*



Desaparece el programa existente en ese momento y pueden comenzar sobre una plataforma de programación libre con la elaboración de un nuevo programa de conexiones.

#### ***B.1.2.2. Elaboración mediante barra de íconos***

Para iniciarse en la elaboración de esquemas de conexiones con Logo Soft Comfort, sólo se necesitan los botones de mando de las dos barras de íconos.

Para elegir los botones de mando se pasa el puntero del ratón sobre el botón de mando y se pulsa la tecla izquierda de aquél. El botón de mando seleccionado aparece entonces como confirmado.



***Figura 37. Botón de mando***

#### ***B.1.2.3. Selección de bloques funcionales***

El primer paso para la elaboración de esquemas de conexiones consiste en la selección de los bloques funcionales requeridos para el circuito a crear. Queda a su criterio si en primer lugar colocan las entradas y salidas, las funciones básicas o las funciones especiales. En la barra de herramientas encontrarán bajo **C<sub>o</sub>** las constantes y las bornes de conexión, es decir, diferentes entradas y salidas y nivel de señal fijos. Bajo **GF** encontrarán las funciones básicas del Álgebra de Boole, es decir, los más simples elementos digitales de enlace. Bajo **SF** encontrarán bloques con funciones especiales.



*Figura 38. Bloques funcionales*

#### ***B.1.2.4. Unión de bloques funcionales: Reglas***

Para la unión de bloques funcionales son de aplicación las siguientes reglas:

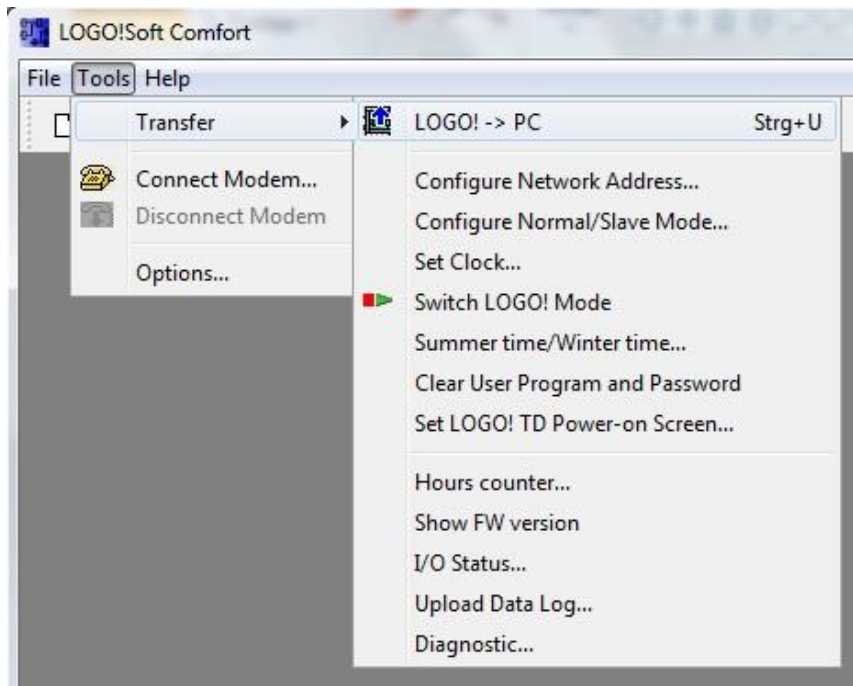
- Una unión siempre sólo puede crearse entre una entrada de bloque y una salida de bloque.
- Una salida puede estar enlazada a varias entradas, pero no una entrada a más de una salida.
- Una entrada y una salida no pueden unirse entre sí en la misma ruta de programa (sin recursión). De necesitarse tal conexión hay que intercalar un marcador o una salida.
- En las funciones especiales existen también "pines de conexión" verdes. Éstos no constituyen pines de acoplamiento, sino sirven para la coordinación de los ajustes de parámetros adyacentes.

#### ***B.1.2.5. Carga del programa desde LOGO al PC.***

Una vez que se haya establecido la conexión por cable y se hayan realizado los ajustes en la interfaz, se puede cargar el programa desde LOGO en el PC. Haga clic en el botón

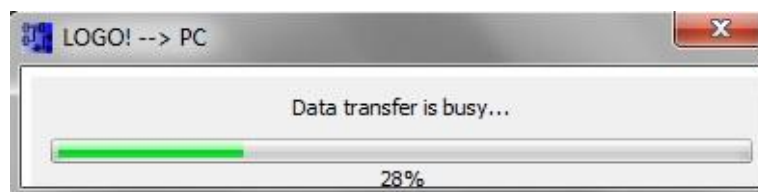


para LOGO -> PC o transfiera el programa a través del menú Tools (Herramientas).



**Figura 39.** Ventana para cargar un programa desde LOGO al PC.

La transferencia de datos comienza tras confirmar la interfaz.

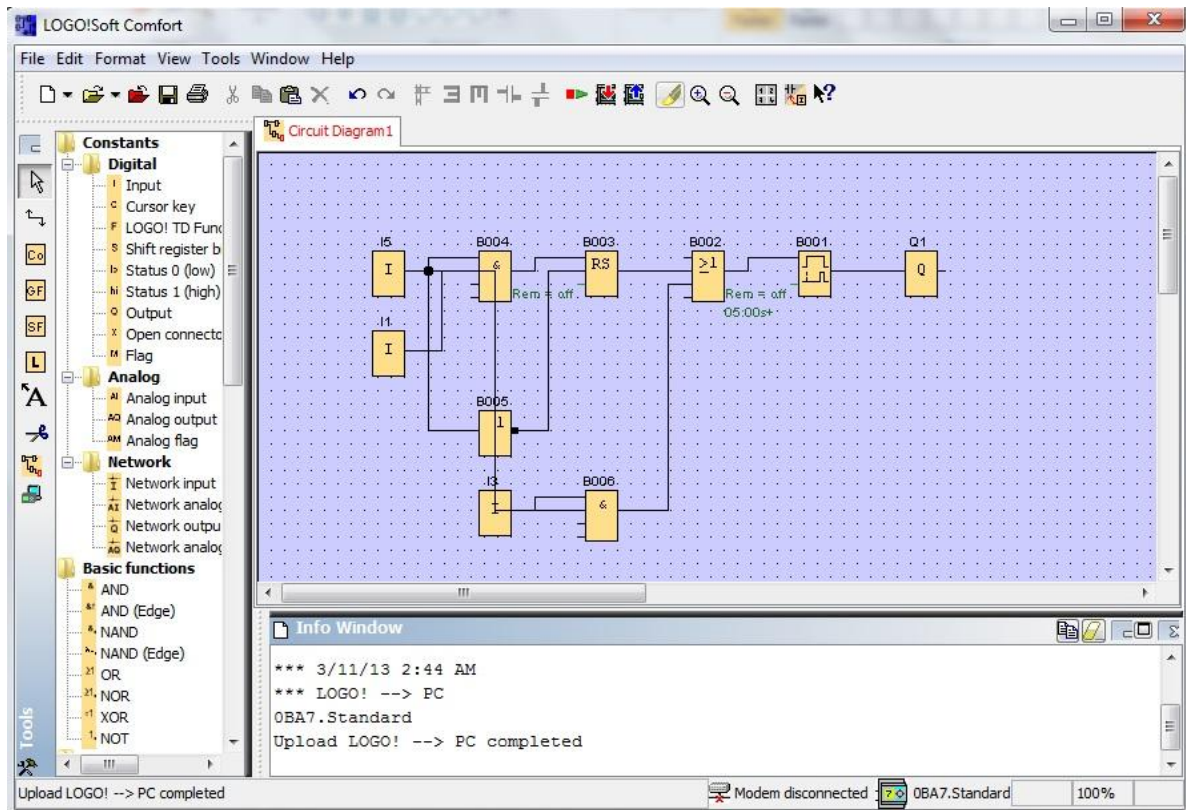


**Figura 40.** Ventana de avance de carga de programa

Una vez cargado el programa, se muestran la interfaz de usuario y el esquema de conexiones

#### **B.1.2.6. Interfaz de usuario**

Ahora puede ver la interfaz de usuario completa de Logo Soft Comfort. La mayor parte de la pantalla la ocupa entonces el área dedicada a la creación de esquemas de conexiones. En la interfaz de programación se disponen los símbolos y enlaces del programa. La lupa permite aumentar la representación.



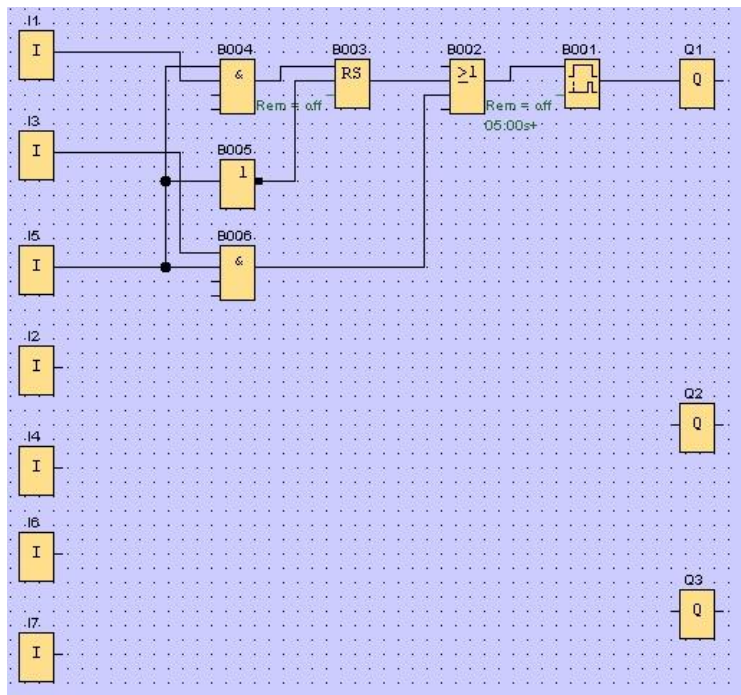
*Figura 41. Interfaz de trabajo de Logo para usuario*

En primer lugar, los objetos del esquema de conexiones deben disponerse ordenadamente.

Los bloques y los cables de conexión se desplazan haciendo clic sobre ellos con el ratón.

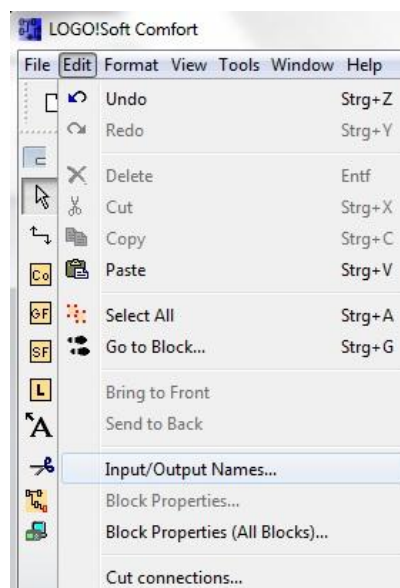
#### **B.1.2.7. Seleccionar bloques**

Los primeros pasos para la creación de esquemas de conexiones son la elección y la asignación de las entradas y salidas necesarias para el circuito que se desea implementar. En primer lugar se insertan los bloques de las entradas I2, I4, I6 e I7 y las salidas Q2 y Q3.



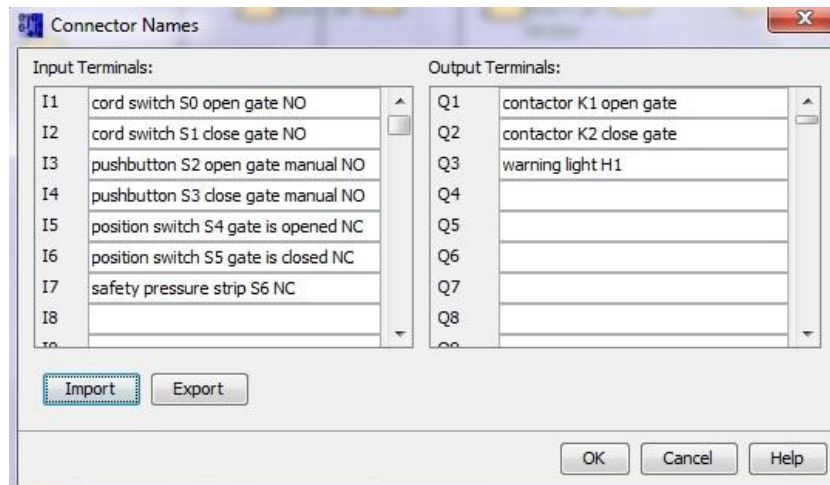
**Figura 42.** Selección de bloques para programa

A continuación se asignan los nombres de las conexiones. Seleccione en el menú Edit (Editar) los Input/output names (Nombres de las conexiones).



**Figura 43.** Cambiar nombres de conexiones

Para mayor claridad, aquí se pueden identificar los Input Terminals (Bornes de entrada) y Output Terminals (Bornes de salida) con los nombres de las conexiones (asignación de símbolos).

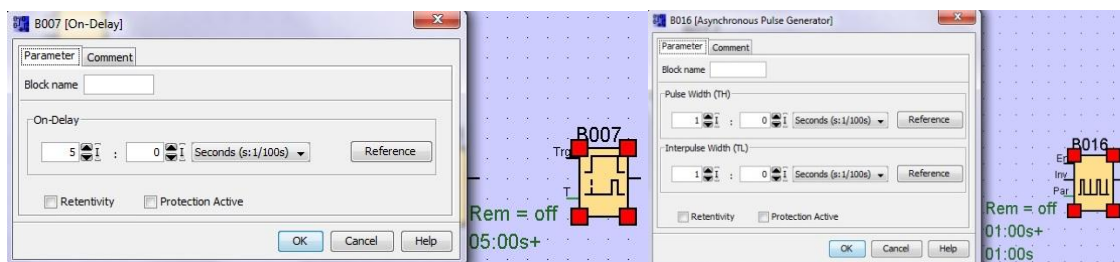


**Figura 44.** Nombre de los bornes de entrada y salida del programa

A continuación se insertan todos los bloques ordenados desde la salida a la entrada. De este modo se disponen los números de bloque como en la programación con LOGO.

En primer lugar, hacer clic con el ratón en el símbolo de la lista de selección y después colocar el bloque en el esquema de conexiones.


Haciendo doble clic en un bloque en el esquema de contactos, podrá ajustar las propiedades y los parámetros.



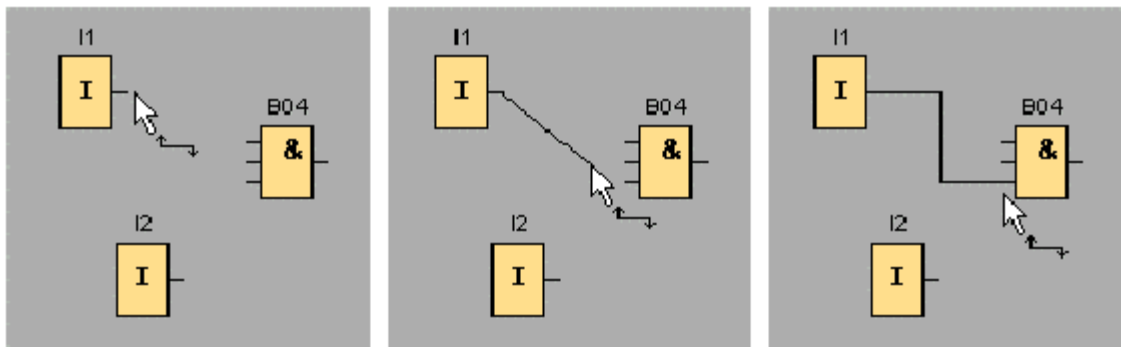
**Figura 45.** Propiedades y parámetros de un esquema de contactos

### B.1.2.8. Conectar bloques

Para completar el circuito, es preciso interconectar además los distintos bloques entre sí.

A este efecto, en la barra de herramientas "Herramientas", haga clic en el botón  que permite conectar bloques.

Mueva el cursor del ratón sobre la conexión de un bloque y pulse el botón izquierdo del ratón. Manteniendo pulsado el botón, mueva el cursor del ratón a una conexión que desee conectar a la primera conexión, y suelte el botón del ratón. Logo Soft Comfort interconectará así ambas conexiones.



*Figura 46. Conexión entre bloques*

Logo Soft Comfort ofrece otra posibilidad de conectar bloques: haciendo clic con el botón derecho del ratón en la entrada o salida de un bloque. Si elige el comando Connect with Block (Conectar con bloque...) del menú contextual, se abrirá una lista de selección que incluye todos los bloques con los que puede establecerse una conexión. Haga clic en el bloque con el que desea establecer una conexión, y Logo Soft Comfort dibujará la

conexión. Este tipo de conexión es idóneo para interconectar bloques que se encuentran muy alejados unos de otros en la interfaz de programación.


#### ***B.1.2.9. Modificar y optimizar la visualización***

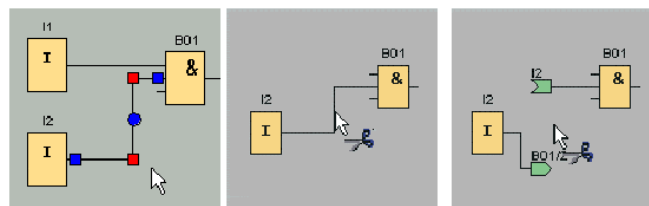
El programa queda listo una vez insertados y conectados los bloques. Para obtener una vista clara y fácilmente comprensible del circuito creado es necesario realizar algunos retoques.

Los objetos posicionados (por ejemplo bloques y líneas) se pueden desplazar si fuese necesario.

#### ***B.1.2.10. Editar líneas de conexión seleccionadas***

Existe una posibilidad especial para editar las líneas de conexión. Las líneas de conexión seleccionadas tienen marcas azules tanto redondas como cuadradas. Las marcas redondas permiten desplazar las líneas de conexión perpendicularmente a su recorrido. Las marcas cuadradas pueden utilizarse para reasignar el comienzo y final de una línea de conexión. Las líneas se desplazan arrastrando las marcas redondas.

Tras activar la herramienta  tijera/conector, haga clic en una conexión. La línea de conexión seleccionada se interrumpe gráficamente. Sin embargo, la conexión entre ambos bloques permanece activa.




***Figura 47. Interrupción de una conexión temporal***

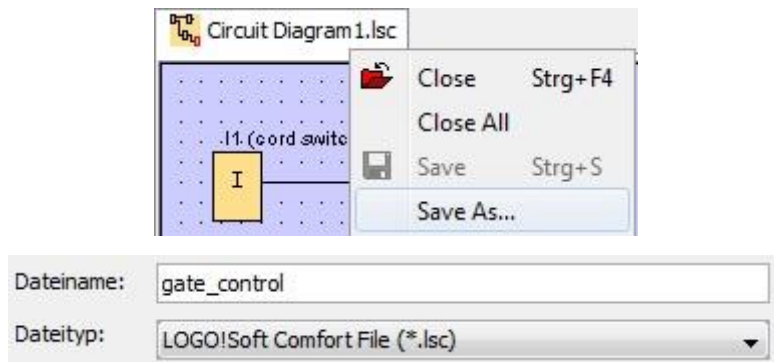


### ***B.1.2.11. Almacenamiento de esquema de conexiones terminado***

Una vez que nuestro circuito se ha introducido y optimizado, podrá guardarse.

Haga clic en el símbolo de disquete  para guardar.


En Save as (Guardar como) se puede asignar el nombre de proyecto "Control de portón de fábrica".



***Figura 48. Como guardar el programa realizado***

### ***B.1.2.12. Simulación del circuito***


Con la simulación del programa se puede comprobar un programa y modificar su parametrización. Ello garantiza que se transfiera al dispositivo Logo un programa optimizado que funcione correctamente. Las señales de entrada deben preajustarse para la simulación.

Para iniciar la simulación, haga clic con el ratón en el botón  Simulación de la barra de herramientas "Herramientas". De este modo se conmuta al modo de simulación.

Una vez que haya probado su programa con la simulación de Logo Soft Comfort, podrá

transferirlo del PC a LOGO con el botón .

### B.1.2.13. Test online

Con el botón para el test online  se puede probar el programa online utilizando LOGO. Se mostrarán los estados de las entradas y salidas y las conexiones lógicas.

Haga clic en el botón  Iniciar observación.

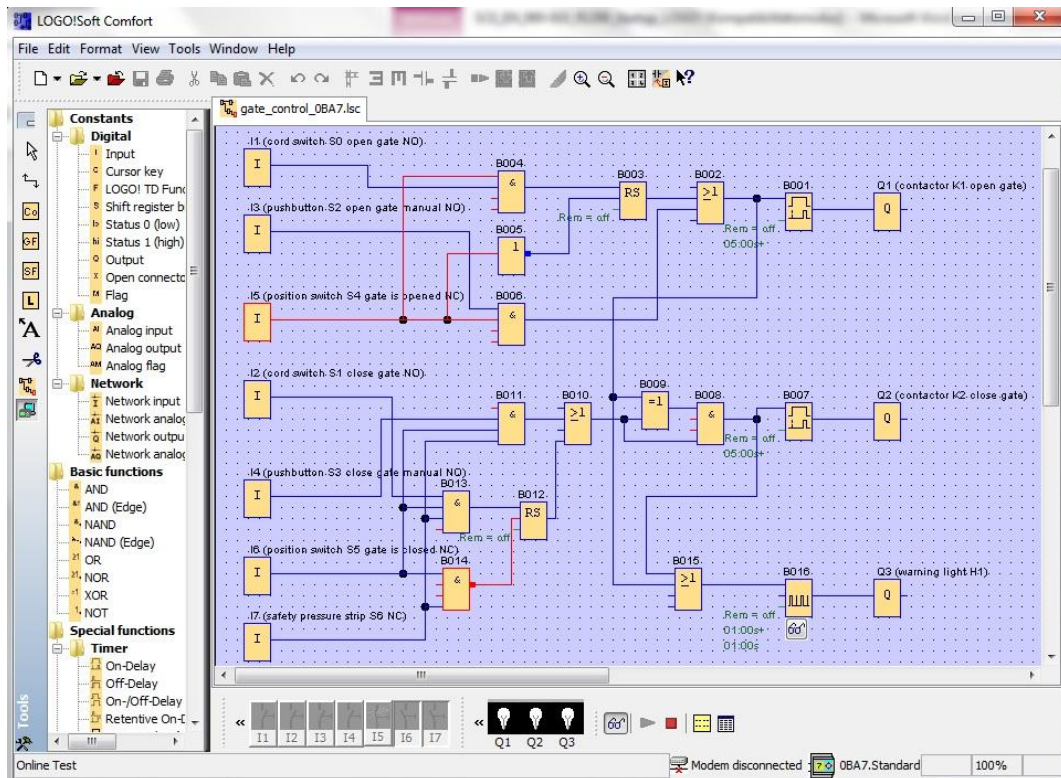


Figura 49. Funcionamiento de un programa

## C. MATERIALES PARA ELABORACIÓN DE PROYECTO

A continuación se describen los principales materiales necesarios para la elaboración de un prototipo para simular la sirena controlada para el cambio de hora en el AEIRNNA de la Universidad Nacional de Loja.

*Tabla 6. Materiales para el prototipo del sistema sonoro.*

CANTIDAD	MATERIAL
1	PLC Siemens 230RC
1	Sirena de 110V – AC
8	Jacks de Banana (macho)
20	Jacks de Banana (hembra)
1	Cable de audio (10 metros)
1	Maqueta de Madera
1	Patch Cord (2 metros)
1	Switch ON/OFF
1	Portafusibles
1	Fusible de 2.5A
1	Led indicador de 110V-AC
1	Switch 3 estados (ON-OFF-ON)

*Fuente: Autor*

## D. PROCESO METODOLÓGICO

### D.1. IMPLEMENTACIÓN DE MÓDULO DE CAMBIO DE HORARIO DE CLASES

Tras el análisis del funcionamiento, tipos, utilidad y programación de un PLC, se procedió a diseñar un módulo que permite implementar un sistema automatizado para simular el cambio de horario de clases en el AEIRNNR, el cual consta de un diseño, armado, programación e implementación.

#### D.1.1. Construcción del Módulo de Prácticas con PLC

En la etapa de construcción se consideró las funcionalidades que posee un PLC, para integrar y desarrollar un diseño compacto, flexible y eficiente que permita interconectar de forma sencillas cada uno de sus pines en la implementación de prácticas para automatizar un proceso.

##### D.1.1.1. Diseño

Para el diseño es importante tener presente los requerimientos, funciones y acciones debe cumplir, además de que tiene que ser ajustable y adaptable para otras automatizaciones o prácticas; de ahí que se tuvo en mente lo siguiente:



*Figura 50. Esquema de funcionamiento de sistema de cambio de horario*

Para ser adaptable y fácil de usar para otras automatizaciones, es importante cambiar sencillo diseño principal a uno un tanto más complejo, que permite manipular todas las entradas y salidas del PLC con el fin de cumplir el cometido.

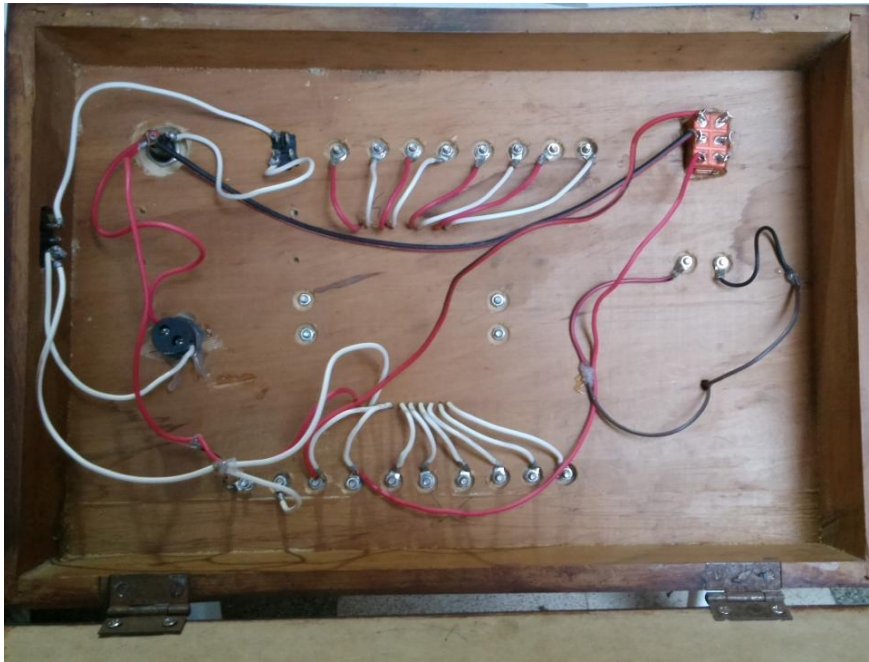
#### *D.1.1.2. Montaje y Armado*

Para utilizar todas las entradas y salidas de un PLC e implementar un módulo de prácticas, hay que tener presente que se debe utilizar: conectores, transformador, sirena, sistemas de protección.



*Figura 51. Estructura del diseño del módulo con PLC*

Teniendo en mente la estructura diseñada (figura 51) se procedió a armar el módulo con un PLC y materiales reales, haciéndolo amigable, portátil y fácil de manipular para la realización de múltiples automatizaciones. (Figura 52, 53 y 54).



*Figura 52. Vista de la distribución interna del módulo.*



*Figura 53. Vista de la distribución externa del módulo.*



*Figura 54. Vista del módulo terminado e implementado.*

Una vez implementado el módulo es importante describir para que sirve cada una de las instalaciones realizadas.

#### ***D.1.1.2.1. Conectores,***

Permiten de forma sencilla conectar las entradas y salidas del PLC, sin dañar los bornes del dispositivo y sin distinción entre DC y AC.

#### ***D.1.1.2.2. Sirena***

El indicador permite al usuario tener una señal del tipo sonora, que en este caso servirá para informar al cuerpo docente y estudiantil acerca del cambio de hora de clase.

#### ***D.1.1.2.3. Dispositivos de Protección***

Procuran la protección de todo el módulo especialmente del PLC, en el caso de que se realice una mala conexión, así para el efecto se instalaron:

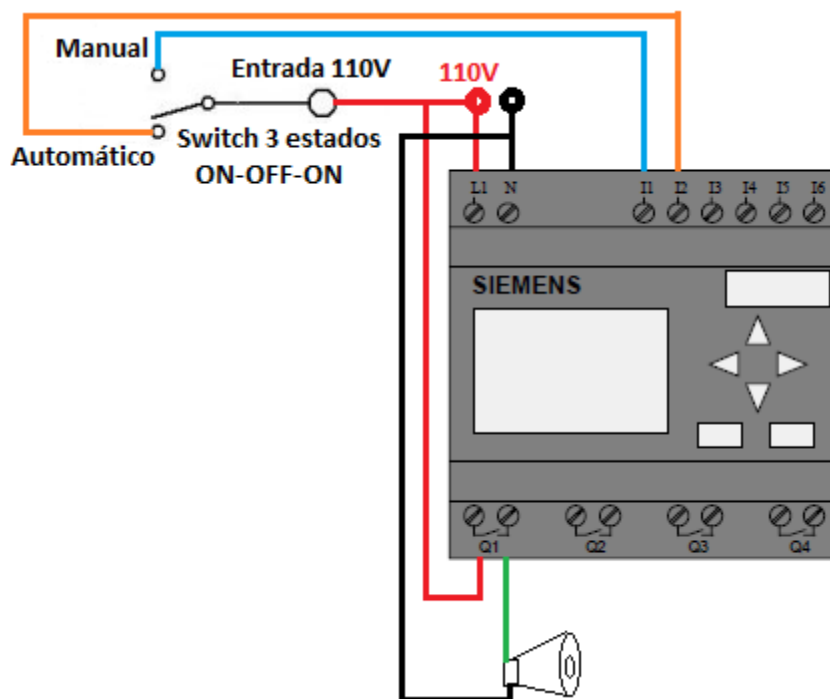
- Fusible 2.5A
- Switch de 110 V a 240 V AC-DC.
- Luz indicadora de 110V- AC

### D.1.2. Programación

El proceso de programación del PLC, está compuesto por el diagrama de conexión física en el módulo y los bloques de programación de LOGO,

#### D.1.2.1. Alarma para cambio de horario de clase con PLC

Con el módulo implementado, se desarrolló una práctica en la que se automatizó el cambio de horario con un sistema sonoro de aviso para el AEIRNNR. El diagrama de conexión entre las entradas y salidas del PLC con las señales respectivas, se establece conforme a la figura 55.

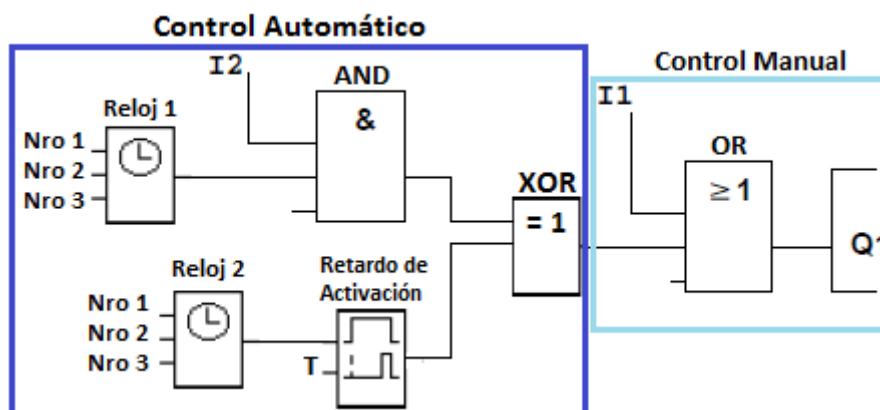


*Figura 55. Esquema de conexión del PLC en el módulo*



- La conexión de los puertos  $L_1$  y  $N$  se realiza a 110V-AC.
- En virtud de que la sirena funciona con 110V-AC, la entrada de  $Q_1$  está conectada al suministro controlado de energía y la salida al pin de la sirena.
- El switch de 3 estados (On-Off-On) se conecta internamente, a la entrada  $I_1$  que permite un control manual de la sirena y a la entrada  $I_2$  que activa el control automático con el tiempo pre-programado.
- La salida de  $Q_1$  sirve para activar la sirena que da el aviso de cambio de horario.

La programación manual que se introdujo en el PLC se realiza a través de los diagramas de bloque de LOGO conforme al gráfico de la figura 56.



*Figura 56. Esquema de programación en Logo.*

Básicamente, el programa que permite cumplir las metas propuestas se encuentra diseñado utilizando los bloques: Funciones constantes ( $I_1$  e  $I_2$ ), funciones especiales (ON-Line) y funciones generales (Compuertas: Or, And y Xor), que se encuentran ubicados de forma estratégica para que el programa permita una ejecución manual y automática:

- $I_1$  → Entrada que permite la activación manual de la sirena.
- $I_2$  → Entrada que activa la opción automática.
- T → Es el tiempo que se espera para activar la sirena.
- Reloj 1 → Son los días y horas a las que se activa la sirena.
- Reloj 2 → Tiempo de activación de sirena

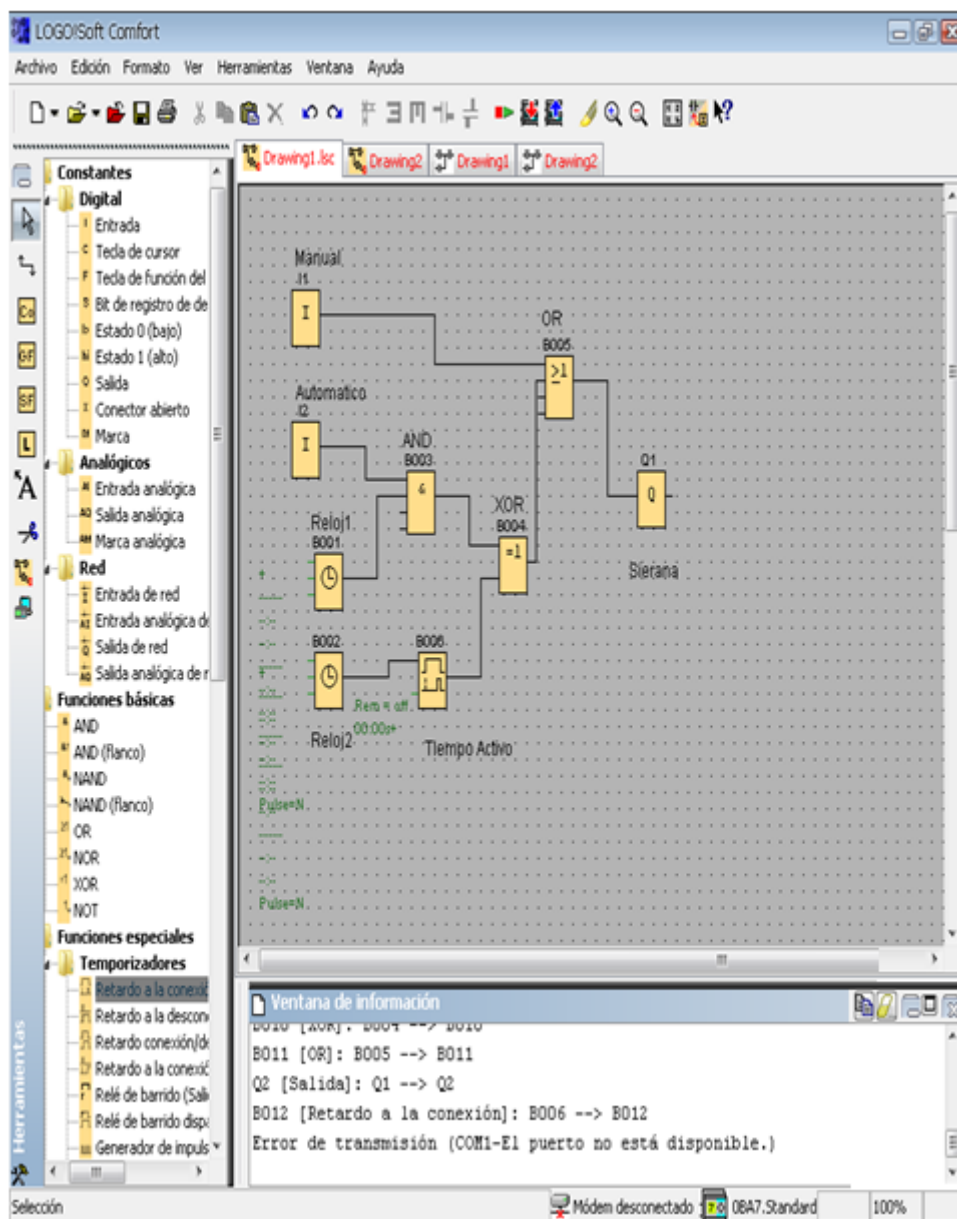


Figura 57. Esquema de programación simulado en Logo Soft Comfort.

En la figura 57 se simuló la programación en Logo Soft Confort, para poder corregir los errores de una forma más sencilla que permite observar los parámetros de programación y los datos que cada bloque arroja.

Finalmente, una vez terminada la programación y corregidos los errores se armó el esquema en el módulo de prácticas (Figura d.8), en la que se comprobó el éxito de la automatización.



*Figura 58. Esquema de conexión armado en el módulo*

## E. CONCLUSIONES

- ✓ Se implementó un módulo de prácticas que integra un PLC Siemens 230RC, con un diseño que permite que sea flexible, portable y admita la implementación de un sin número de prácticas utilizando todas las entradas y salidas, con voltajes alternos y continuos entre 110V y 240V.
- ✓ Se automatizó un prototipo de sistema sonoro que permite simular el cambio de horario de clase, con un control manual y automático que utiliza 2 entradas, 1 salidas, 1 switch de 3 estados y una sirena de 110V-AC del módulo.
- ✓ Se diseñó e ingresó un programa de forma manual en el PLC utilizando los diagramas de LOGO, que se encuentra compuesto por funciones especiales (On Line), funciones generales (Compuertas: Or, And y Xor) y funciones constantes (entradas y salidas), con un control manual que permite al usuario activar al sirena en cualquier instante y un control automático que en los horarios y tiempos pre-programados.
- ✓ La implementación del módulo con PLC adaptable para múltiples automatizaciones, permite mejorar la calidad del aprendizaje en el funcionamiento y manejo de los PLC.
- ✓ El módulo consta con un sistema de protección compuesto por un switch, un fusible y una luz indicadora de 110V, para en el caso de existir un cortocircuito por alguna mala conexión proteger el PLC y los elementos que constituyen el módulo.

## F. RECOMENDACIONES

- ✓ Para utilizar las entradas  $I_1$  e  $I_2$  del módulo de prácticas es importante que el switch de 3 estados se encuentre en la posición OFF.
- ✓ Es muy importante verificar todo el cableado del módulo antes de conectarlo a la red eléctrica, para así evitar daños en el PLC y los artefactos que se están controlando.
- ✓ Se debe evitar manipular un PLC cuando se está ejecutando su programación, para que no ocurran daños en los artefactos que intervienen el proceso de control.
- ✓ Se debe tener en cuenta que antes de automatizar una tarea en un PLC, es necesario tener una buena base de conocimiento sobre el funcionamiento de un PLC para realizar un diagrama en el que consten todos los detalles de lo que se debe hacer para que nada salga mal.
- ✓ Es indispensable revisar periódicamente los dispositivos de protección del módulo, para poder garantizar su funcionamiento a largo tiempo.

## **G. BIBLIOGRAFÍA**

### **LIBROS CONSULTADOS**

- [1] ABB, "40 & 50 Series, Technical Manual"
- [2] ABB, "AC31GRAF Programming Software, Software Manual", 1997
- [3] Andy Swales, Schneider Electric, "Open MODBUS/TCP Specification", Release 1.0, 29 March 1999
- [4] FERREIRA, M. V. A. PLC - Power Line Communication. UFF, 2005.
- [5] J. Webb y R. Reis, "Programmable Logic Controllers: Principles and Applications", 4th. edition, Prentice Hall, ISBN 0-13-679408-4, 1999
- [6] LITTLE, A. D. White Paper On Power Line Communications (PLC) and its Impact on the Development of Broadband in Europe. PUA, 2002.
- [7] Phil Melore, "PLC Tutor", sitio web [www.PLC.net](http://www.PLC.net), 2001
- [8] S. Brian Morriss, "Programmable Logic Controllers", Prentice Hall, ISBN 0-13-095565-5, 2000
- [9] TEIXEIRA, E. R. D. PLC – Power Line Communications. 2005.

### **PÁGINAS WEB DE INTERÉS:**

- [10] <http://www.abcinnova.com/articulos-e-informacion/18-ique-es-un-plc-y-que-beneficios-tiene.html>
- [11] <http://bricos.com/2012/08/que-es-un-plc-control-logico-programable/>
- [12] [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lmt/maza\\_c\\_ac/capitulo4.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lmt/maza_c_ac/capitulo4.pdf)

- [13] <http://ctinmx.com/que-es-un-plc/>
- [14] <http://es.wikipedia.org/wiki/PLC>
- [15] <http://www.esgerencia.com/pag/152/plc---objetivos>
- [16] [http://www.fio.unicen.edu.ar/usuario/ggacosta/TranspaSSCC/06capitulo\(PLC\).pdf](http://www.fio.unicen.edu.ar/usuario/ggacosta/TranspaSSCC/06capitulo(PLC).pdf)
- [17] [http://www.plc4ever.com/media/secciones/soft/manual\\_AUNA\\_PLC.pdf](http://www.plc4ever.com/media/secciones/soft/manual_AUNA_PLC.pdf)
- [18] <http://ocw.upc.edu/sites/default/files/materials/15012628/40197-3452.pdf>
- [19] <http://recursostic.educacion.es/observatorio/web/gl/component/content/article/502-monografico-lenguajes-de-programacion?start=2>
- [20] <http://www.sc.ehu.es/sbweb/webcentro/automatica/WebCQMH1/PAGINA%20PRINCIPAL/PLC/plc.htm>
- [21] [https://www.uclm.es/profesorado/rcarcelen\\_plc/prog.htm](https://www.uclm.es/profesorado/rcarcelen_plc/prog.htm)
- [22] [http://www.unicrom.com/Tut\\_ProgramarPLC.asp](http://www.unicrom.com/Tut_ProgramarPLC.asp)
- [23] <http://www.xatakahome.com/la-red-local/redes-plc-ii-como-sacarles-el-maximo-partido>

## **H. ANEXOS**

## **PROYECTO**