Optimización en el desarrollo de un programa para PLC

Zoraima Lucena, Miguel Indriago
Departamento de Ingeniería Electrónica. Universidad Nacional Experimental Politécnica, UNEXPO, Barquisimeto, Venezuela
Email: zlucena@unexpo.edu.ve, mindria@unexpo.edu.ve

Resumen

En los procesos industriales uno de los objetivos principales es obtener mayor producción que cumpla con estándares de calidad involucrando menores costos y tiempos de elaboración, la necesidad de satisfacer estos requerimientos ha llevado a que actualmente muchas de las empresas se encuentren automatizadas o en vías de la automatización. Una alternativa para automatizar procesos industriales son los controladores lógicos programables (PLC); Para los responsables del desarrollo de proyectos de automatización, el tiempo es una variable crítica, que incide directamente en el costo del proyecto. En este trabajo, se presenta una propuesta para implementar un software que permite optimizar este proceso, reduciendo los tiempos y los errores de programación, al permitir desarrollar el programa para el PLC de manera automática a partir de la lista de equipos de entrada/salida y la filosofía de control, esto además refleja una disminución en el costo del proyecto.

Palabras clave: Automatización industrial, PLC, norma ISA SP88.

Optimization in the development of a program for PLC

Abstract

One of the main objectives in the industrial processes is to obtain higher production that meets the standard of quality involving smaller costs and elaboration time. The needs to satisfy these requirements has driven many companies to be automated or in way to automation. An alternative to automate industrial processes is the programmable logical controllers (PLC). For those responsible for developing automation projects, time is a critical variable that impacts directly in the cost of the project. This work shows a proposal to implement software that allows us to optimize this process, by reducing the time and the programming errors, and to develop the program for the PLC in an automatic way from a list of input/output equipment and the control philosophy, this also produces a decrease in the cost of the project.

Keywords: Industrial automation, PLC, norm ISA SP88.

1. INTRODUCCIÓN

La automatización es una herramienta que permite mejorar el rendimiento de todo sistema donde es aplicada. En el caso de los procesos industriales, ésta involucra el manejo de gran cantidad de información asociada a los equipos, dispositivos y características del proceso. Los PLC surgen como una alternativa para la automatización de procesos industriales tomada muy en serio por toda la comunidad de automatización, incluso existe una norma de IEC (61131) que establece unos lineamientos para la programación del PLC y que usa Cartas de Funciones Secuenciales (SFC) o Grafcet [1], [2], [3] y [4]. No todos los PLC tienen la capacidad de ser programados en SFC, sólo los más costosos, por esta razón en este artículo se propone una alternativa que pueda ser aplicada a cualquier PLC pensando en la pequeña y mediana empresa cuyas inversiones de dinero en la automatización no son tan fuertes.

Para la automatización de los procesos industriales utilizando controladores lógicos programables (PLC), lo primero que se debe tener es una lista de los

dispositivos de entrada y salida involucrados en dicho proceso. También es necesaria la elaboración de la filosofía de control, en donde se describen paso a paso todas las secuencias de operación del sistema que se quiere automatizar y cómo están relacionados los dispositivos de entrada y salida con estos pasos. Luego, el programador a partir de ésta información elabora la aplicación para un PLC que permita manejar el proceso.

Este procedimiento puede resultar largo y tedioso, ya que a mayor complejidad en el proceso, mayor es el volumen de información que se genera, lo que se presta para cometer errores de lógica y de sintaxis en el momento de la programación, generándose pérdidas de tiempo que demoran la finalización del proyecto, lo que se traduce en incrementos de costos, este problema se ve acentuado cuando el programador es inexperto.

El objetivo principal de esta investigación es presentar una propuesta que permita generar de forma automática el programa de aplicación para el PLC a partir de la filosofía de control y la lista de entradas y salidas, permitiendo disminuir los tiempos y los errores (optimización) asociados a la automatización de procesos industriales. Para el desarrollo de la propuesta se tomó como base una estructura de programación basada en el modelo físico del estándar ISA SP88 [5].

2. FUNDAMENTOS TEÓRICOS

2.1. Modelo Físico del Estándar Industrial Internacional, SP88

ISA SP88 [5] es un estándar de control que busca facilitar la automatización de los procesos por lotes, que se caracterizan por ser inherentemente flexibles debido a la variedad de productos que pueden ser producidos y los equipos de control requeridos, razón por la cual muchas plantas de manufactura por lotes son todavía operadas manualmente.

El estándar posee dos modelos principales: físico y de procedimiento. En términos generales, el modelo físico es usado para describir equipos, y el modelo de procedimiento describe las recetas de los procesos secuenciales.

El modelo físico, puede ser usado para describir propiedades físicas de una compañía en términos de siete niveles: empresa, sitio, área, celda de proceso, unidad, módulo de equipo y módulo de control. En la Figura 1 se muestra el diagrama de los siete niveles que conforman el modelo físico.

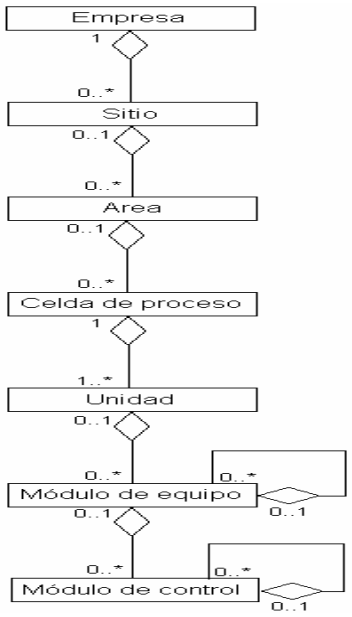

Figura 1. Modelo Físico en formato UML.

Los tres niveles superiores, empresa, sitio, y área, frecuentemente están definidos por consideraciones de negocio y son parte del modelo para identificar adecuadamente la relación de los equipos de los niveles inferiores de la fábrica [6].

Los cuatros niveles inferiores de este modelo se refieren a tipos de equipos específicos, son establecidos para definir técnicamente y delimitar grupos de equipos. Están definidos por actividades de ingeniería. Durante esas actividades de ingeniería, los equipos en un nivel son agrupados para formar equipos en los niveles superiores. Esto está hecho para simplificar la operación del equipo formado tratándolo como un solo equipo más grande. Una vez creado, el equipo no puede ser dividido excepto por una re-ingeniería del equipo en ese nivel.

Los módulos de control que son parte del modelo tienen las siguientes funciones [7]:
1. Cambio de modo manual - automático.
2. Parada de emergencia.

3. Verificación de respuesta de marcha.
4. Limpieza de las alarmas.

Los tipos de módulos de control que consideraremos en esta investigación son tres y se mencionan a continuación [7]:
Tipo 1. Una entrada y una salida.
Tipo 2. Dos entradas y una salida.
Tipo 3. Dos entradas y dos salidas.

Sólo se mencionan estos tres tipos de módulos de control por ser los de uso más frecuente, pero esto no quiere decir que sean los únicos. En las Figuras 2, 3 y 4 se muestran los diagramas lógicos que describen con detalles estos tres tipos de módulos de control.

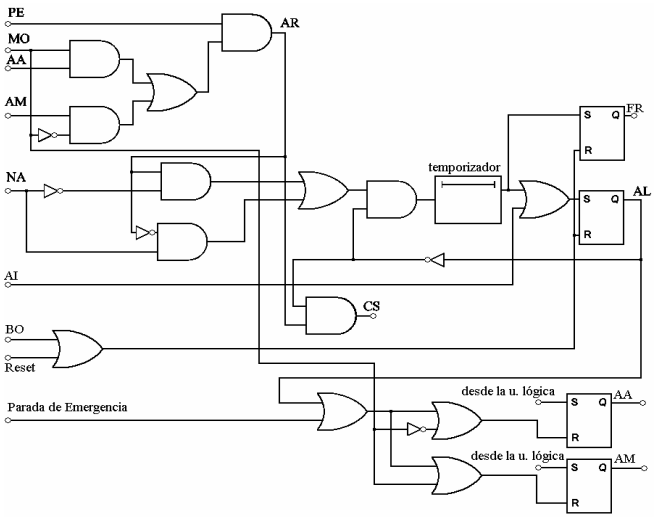

Figura 2. Módulo de control tipo 1, una entrada una salida.

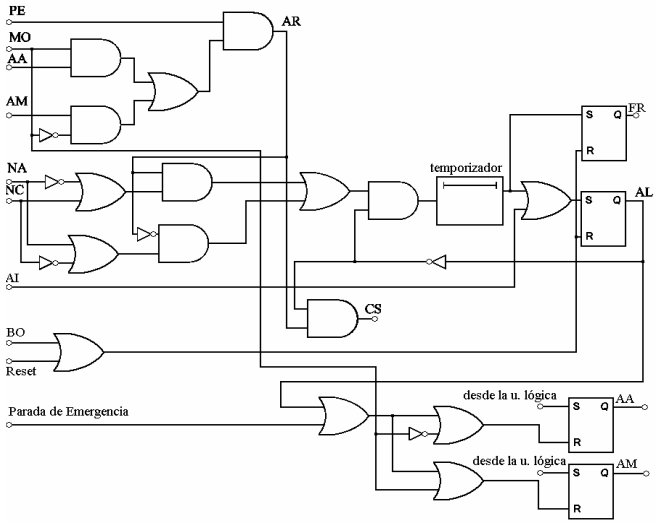

Figura 3. Módulo de control tipo 2, dos entradas una salida.

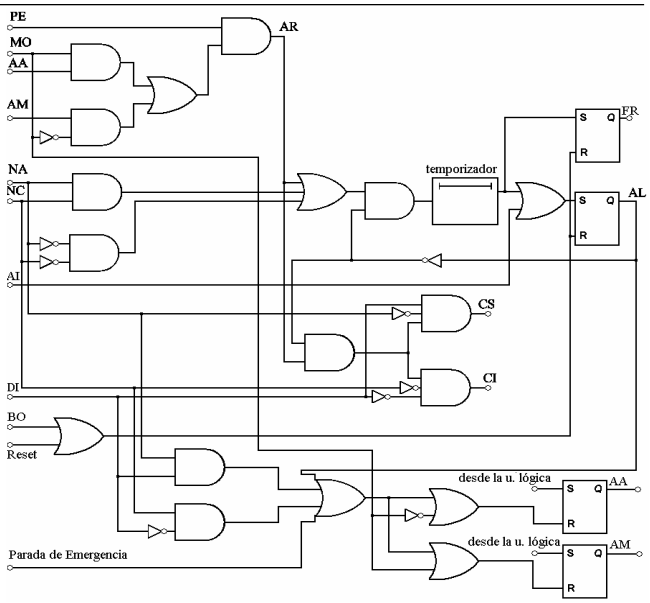

Figura 4. Módulo de control tipo 3, dos entradas dos salidas.

3. ESTRUCTURA PROPUESTA

El objetivo del trabajo es elaborar un programa que permita desarrollar de manera más rápida y confiable programas para la automatización de procesos industriales con PLC. Para ello se utilizó el modelo físico del estándar SP88 de la ISA [5] y [6] teniendo en especial consideración los cuatros niveles inferiores (celda de proceso, unidad, equipos de control y módulo de control) que son los que están más estrechamente ligados con los equipos que forman parte del proceso. En esta aplicación establecimos como norma que un PLC no va a gobernar unidades en más de una celda de proceso, esto permite definir de una forma ordenada una estructura tanto en la memoria de programa como en la memoria de datos del PLC de acuerdo con el modelo físico de la norma SP88.

3.1. Memoria de datos

3.1.1. Estructura del archivo de datos

Para el desarrollo de la aplicación se necesita que los datos del PLC sean al menos de 16 bits y se propone una estructura en los archivos de la memoria de datos que tengan, además de los archivos de datos de uso general, los archivos que se indican en la Tabla 1. Tal estructura va a permitir manejar con cada PLC hasta 10 unidades (de la 0 a la 9) de una celda de proceso.

Tabla 1. Distribución de la memoria de datos.

Cantidad	Tipo	Descripción del archivo
1	N o B	Datos generales del sistema
1	N o B	Entradas y salidas N13:64-N13:127 Salidas; N13:128 en adelante Analógicas
10	N o B	Módulos de control de la unidad 0-9
10	N o B	Permisivos y alarmas unidad 0-9
10	T	Temporizadores unidad 0-9

La tetras N, B y T dentro del tipo corresponden a entero, binario o temporizador, respectivamente. Es importante resaltar que el archivo de enteros debe ser de 16 bits y que cada uno de estos bits pueda ser accedido de forma individual y los bits del archivo binario deben poder ser asociados en grupos de al menos 16 bit. La principal finalidad de esta distribución de la memoria de datos es asignarle a cada dirección física (señal proveniente de una tarjeta de entrada o salida) una dirección lógica (dirección dentro de la memoria del procesador) fácilmente identificable.

Archivo de datos generales del sistema

En este archivo se colocan todos los datos que tienen que ver con las 10 unidades contenidas dentro del PLC. Hasta el momento se han definido 3 bits, estos son:

1. Parada de emergencia global
2. Reconocimiento general de alarmas
3. Simulación

Archivo de entradas y salidas

Este archivo va a contener todas las direcciones físicas de la lista de entrada y salida que no pertenezcan a ningún módulo de control. Tomando en cuenta que cada archivo está formado por elementos de 16 bits, el archivo está dividido en tres partes: del elemento 0 al 63 corresponden a las entradas digitales, del elemento 64 al 127 corresponden a las salidas digitales y del elemento 128 en adelante corresponden a las señales analógicas.

Módulos de control

Los elementos de cada archivo correspondiente a los módulos de control de las unidades 0 a la 9 están formados por un grupo de al menos 16 bits y cada bit tiene una función. En la Tabla 2 se resumen las funciones de cada uno y el archivo de programa en donde actúan.

Tabla 2. Función de los bits del módulo de control.

Bit	Descripción	Ext	E/S	Sc	Lg
N2U:m/0	Modo Manual = 0 Automático = 1	MO	x	x	x
N2U:m/1	Arranque en automático	AA			x
N2U:m/2	Arranque en manual	AM		x	x
N2U:m/3	Dirección	DI		x	x
N2U:m/4	Comando inverso	CI	x		
N2U:m/5	Permisivos	PE		x	x
N2U:m/6	Alarmas Interlock	AI			x
N2U:m/7	Comando de arranque	AR			
N2U:m/8	Entrada NA	NA	x	x	
N2U:m/9	Entrada NC	NC	x	x	
N2U:m/10	Comando de salida	CS	x		
N2U:m/11	Falla respuesta comando salida	FR		x	
N2U:m/12	ONS	ONS			x
N2U:m/13	ONS	ONS			x
N2U:m/14	Borrar alarmas	BO		x	
N2U:m/15	Alarma general	AL		x	

La U y la m en la columna de bit corresponden a la unidad y al módulo de control respectivamente. La columna Ext corresponde a la extensión con la que es abreviado el bit. La columna E/S indica con una x cuales bits pueden estar dentro del archivo de programa de E/S. La columna Sc indica cuales bits son afectados o leídos por un sistema de adquisición de datos o supervisorio (SCADA), en caso de que el controlador esté conectado a uno. La columna Lg indica con una x cuales bits pueden ser cambiados por la unidad lógica que será explicada más adelante.

La unidad estándar, que será explicada más adelante, no aparece en la tabla ya que independientemente de la aplicación estas unidades siempre son iguales y todos los bits del módulo de control están

presentes en ella con la excepción de los bits 12 y 13 que no tienen una función como tal definida.

En las Figuras 2, 3 y 4 se muestra la función de cada bit en los módulos de control considerados en este trabajo.

Permisivo y alarmas

Los elementos de cada archivo de enteros están formados por un grupo de al menos 16 bits. Los primeros 4 bits corresponden a los permisivos y los bits restantes corresponden a las alarmas. En la Tabla 3 se resume la función de cada bit. La U y la m corresponden a la unidad y al módulo de control respectivamente.

Tabla 3. Permisivos y alarmas.

Dirección	Descripción
N3U:m/0	Permisivo 0
N3U:m/1	Permisivo 1
N3U:m/2	Permisivo 2
N3U:m/3	Permisivo 3
N3U:m/4	Alarma 0
N3U:m/5	Alarma 1
N3U:m/6	Alarma 2
N3U:m/7	Alarma 3
N3U:m/8	Alarma 4
N3U:m/9	Alarma 5
N3U:m/10	Alarma 6
N3U:m/11	Alarma 7
N3U:m/12	Alarma 8
N3U:m/13	Alarma 9
N3U:m/14	Alarma 10
N3U:m/15	Alarma 11

Temporizadores para comando de falta de comprobación (archivos 40 al 49)

Estos archivos son de temporizadores. Cada temporizador tiene como consigna el tiempo máximo que puede pasar entre que se da un comando de salida y se reciben las entradas correspondientes. Si la consigna se alcanza se activa el bit FR. Estos temporizadores están en la unidad estándar.

3.2. Memoria de programa

En esta propuesta se establece que debe existir un archivo de programa para las entradas y salidas, y por cada unidad (de la 0 a la 9) existen dos archivos de programa, el archivo estándar que maneja todas las funciones que no dependen directamente de la aplicación, es decir aquellas funciones que independientemente de la aplicación siempre están presentes y el archivo lógico donde se programan aquellas funciones que son dependientes del proceso, por ejemplo un arranque secuencial. En la Tabla 4 se muestra una lista de los archivos que deben estar en la memoria de programa.

Tabla 4. Asignación de los archivos de programas.

Cantidad	DESCRIPCIÓN
1	Programa principal
1	Archivos de E/S
10	Unidad estándar 0-9
10	Unidad lógica 0-9

3.3. Estructura del archivo de entradas y salidas

El archivo de programa de entradas y salidas es donde se iguala la dirección física con la dirección lógica. Este archivo tiene una función similar a la de un espejo donde todo lo que ocurre en la dirección física se refleja en la dirección lógica.

3.4. Estructura de la unidad estándar

En los archivos de programas correspondientes a las unidades estándar de la 0 a la 9 están programados los comandos de salidas (CS y CI), el comando de arranque en manual o automático (AR), se activan los bits de falla por error en la realimentación (FR) y alarma (AL), se borran los bits de alarmas y ocurren las paradas predeterminadas por el sistema, como por ejemplo la parada de emergencia.

3.5. Formato de la lista de e/s

El software que se va a desarrollar para generar de manera automática el programa para el PLC va a tener como entrada un archivo separado por comas, con extensión CSV. Cada línea del archivo separado por coma consta de ocho campos. La primera línea

contiene el nombre de cada campo, los cuales se mencionan a continuación en el orden en que aparecen en el formato: Item, Descripción, Etiqueta, Dirección, Unidad, Módulo, Bit, y Tipo de módulo de control (MC).

Item: es la posición correspondiente de la entrada o salida dentro de la lista.

Descripción: describe brevemente al elemento o parte del elemento conectado a la entrada o salida. Este campo es muy útil para identificar a los elementos dentro del programa del PLC. Su contenido es opcional, aunque es muy recomendable que aparezca.

Etiqueta: es el nombre con el que se conoce al equipo o parte del equipo dentro del proceso. Su contenido es opcional.

Dirección: corresponde a la dirección de la entrada o salida en las tarjetas de entrada y salida del controlador. Su contenido es obligatorio.

Unidad*:* es la unidad a la cual pertenece el equipo según la división que se haya hecho de la celda de proceso perteneciente a la planta de acuerdo con el modelo físico del estándar SP88 de la ISA. Su contenido es obligatorio para las señales que forman parte de un módulo de control.

Módulo: es el módulo de control al cual pertenece el equipo según la división que se haya hecho de la unidad de acuerdo con el modelo físico del estándar SP88 de la ISA. Su contenido es obligatorio para las señales que forman parte de un módulo de control.

Bit: es el bit correspondiente de la señal dentro del módulo de control y puede ser uno de los siguientes cinco bits: modo (MO), normalmente abierto (NA), normalmente cerrado (NC), comando de salida (CS) y comando de salida en inverso (CI). Cada módulo de salida debe tener al menos el bit de comando de salida y el bit normalmente abierto o normalmente cerrado. Su contenido es obligatorio para las señales que forman parte de un módulo de control.

Tipo de MC: en este campo va un número del 1 al 3 dependiendo del tipo de equipo final de control que este asociado al módulo de control correspondiente. Su contenido es obligatorio para las señales que forman parte de un módulo de control. Las Figuras 2, 3 y 4 muestran el diagrama lógico de cada tipo de modulo de control.

3.6. Estructura del software para la aplicación

La estructura del software orientado a objetos que genera el programa para el PLC consta fundamentalmente de seis clases: La clase DatoES que es la clase de los datos contenidos en la lista de entrada/ salida. La clase ListaES contiene los métodos que le permiten a la clase E_S generar el archivo de programa de estrada/salida y a la clase Estándar generar los archivos de programas estándar de la unidades. La clase Lógica toma los datos de la clase ListaES, pero el arreglo para generar los archivos de programa lógico de las unidades debe hacerlo de forma manual de acuerdo a la filosofía de operación del proceso a controlar, hay muchas formas de hacerlo, se puede usar desde las instrucciones propias del PLC hasta usar técnicas como redes de petri [3], [4], [8] Y [9] o autómatas finitos [2] y [10] para desarrollar este componente del software que se puede extender a más de una clase. La clase Archivo de la cual se derivan las clases E_S, Estándar y Lógica contiene el método que finalmente generará el programa a ser leído por el PLC. La Figura 5 muestra un diagrama de clases del software propuesto.

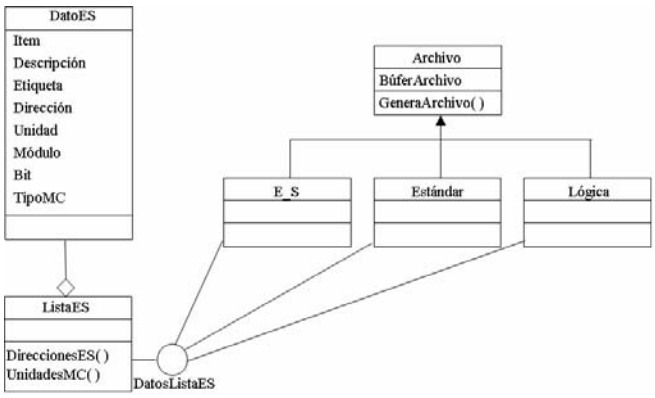

Figura 5. Diagrama de clases en UML

4. EL PROCESO UTILIZADO EN LA PRUEBA PROPUESTA

Este enfoque se implemento para procesadores SLC500 de Allen Bradley. La memoria de datos y de programa para dichos procesadores quedó estructurada como se muestra en las Tablas 5 y 6 respectivamente [11].

Tabla 5. Distribución de la memoria de datos en el SLC500.

No	Tipo	Descripción
0	O	Salidas
1	I	Entradas
2	S	Estado
10	N	Datos generales del sistema
13	N	Entradas y salidas N13:64-N13:127 Salidas; N13:128 en adelante Analógicas
20-29	N	Módulos de control unidades 0-9
30-39	N	Permisivos y alarmas unidad 0-9
40-49	T	Temporizadores unidad 0-9

Tabla 6. Distribución de la memoria de programa en el SLC500.

ARCHIVO	DESCRIPCIÓN
2	Programa principal
6	Archivos de E/S
10-19	Unidad estándar 0-9
20-29	Unidad lógica 0-9

Para medir los resultados del programa se probó el arranque secuencial de tres motores. La operación en automático es como sigue: Existe un botón de arranque y un botón de parada del sistema en automático. Cuando se presiona el botón de arranque se pone en marcha el primer motor, 5 segundos después el segundo motor y 5 segundos después el tercer motor. Cuando se presiona el botón de parada se detienen los tres motores. La operación en manual es como sigue: Cada motor tiene un botón de arranque y un botón de parada en manual. Cuando el motor está en modo manual arranca cuando se presiona el botón de arranque y se detiene cuando se presiona el botón de parada. Además cada motor tiene cableada la señal que indica que el térmico se disparó y ésta genera una alarma que debe detener al motor. En la Tabla 7 se muestra la lista de señales de entrada y salida tal cual como la contendría el archivo de entrada al software de la aplicación.

Para la programación de los archivos de programas lógicos se utilizaron un conjunto de seudo instrucciones que coinciden con el conjunto de instrucciones del PLC [11] mostradas en una caja de dialogo donde también están disponibles las direcciones

Tabla 7. Lista de señales de entrada y salida.

Item	Descripción	Etiqueta	Dirección	Unidad	Módulo	Bit	TipoMC
1	Arranque en automático	AA	I:1/0				
2	Parada en automático	PA	I:1/1				
3	Inicialización.	INI	I:1/2				
4	Modo manual automático motor 1	MOM1	I:1/3	0	0	MO	1
5	Arranque en manual motor 1	AMM1	I:1/4				
6	Parada en manual motor 1	PMM1	I:1/5				
7	Respuesta de marcha (confirmación de arranque) motor 1	NAM1	I:1/6	0	0	NA	1
8	Térmico disparado motor 1	TM1	I:1/7				
9	Bobina de arranque motor 1	CSM1	O:2/0	0	0	CS	1
10	Modo manual automático motor 2	MOM2	I:1/8	0	1	MO	1
11	Arranque en manual motor 2	AMM2	I:1/9				
12	Parada en manual motor 2	PMM2	I:1/10				
13	Respuesta de marcha (confirmación de arranque) motor 2	NAM2	I:1/11	0	1	NA	1
14	Térmico disparado motor 2	TM2	I:1/12				
15	Bobina de arranque motor 2	CSM2	O:2/1	0	1	CS	1
16	Modo manual automático motor 3	MOM3	I:1/13	0	2	MO	1
17	Arranque en manual motor 3	AMM3	I:1/14				
18	Parada en manual motor 3	PMM3	I:1/15				
19	Respuesta de marcha (confirmación de arranque) motor 3	NAM3	I:0.1/0	0	2	NA	1
20	Térmico disparado motor 3	TM3	I:0.1/1				
21	Bobina de arranque motor 3	CSM3	O:2/2	0	2	CS	1

Tabla 8. Tiempos asociados a la programación.

Para tres módulos de control tipo 1

Programación Tradicional

Líneas de programa		Tiempos promedios/Línea (min./línea)	Totales (min.)
E/S	12	1,5	18
Estándar	24	3	72
Lógica	18	2	36
Total tiempo de programación (min.)			126

Programación Propuesta

Líneas de programa		Tiempos promedios/Línea (min./línea)	Totales (min.)
E/S	12	0,014	0,17
Estándar	24	0	0
Lógica	12	0,333	3,996
Total tiempo de programación (min.)			4,166

lógicas con su respectiva descripción de tal manera de hacer más sencilla la programación para el PLC según indica la filosofía de operación. Estas seudo instrucciones se van a colocar en ocho campos por cada módulo de control de las unidades que maneja el PLC. Los ocho campos corresponden a: modo automático, modo manual, arranque automático, parada en automático, arranque en manual, parada en manual, permisivos y alarmas. Se midieron lo tiempos de dos programadores de aproximadamente la misma experiencia en programación y se obtuvieron los tiempos que se muestran en la Tabla 8.

Como se puede observar el tiempo que llevó programar el arranque secuencial de los tres motores es considerablemente menor con la propuesta hecha (4,16 min.) en comparación con la programación tradicional (2 hr. y 6 min.).

5. CONCLUSIONES

Se desarrolló un programa para generar de forma automática, el programa de aplicación para el PLC a partir de la filosofía de control y la lista de entradas y salidas. Para el programa se tomó como base una estructura de programación basada en el modelo físico del estándar ISA SP88.

Una comparación con el procedimiento de programación convencional, mostró que con el programa propuesto se logró una disminución sustancial de los tiempos de programación. Mientras los procesos más encajen en el modelo físico del estándar SP88 mayor eficiencia tendrá la propuesta presentada.

REFERENCIAS

[1] Brinksma, E., Mader, A. y Fehnker, A. (2001): "Verication and Optimization of a PLC Control Schedule". Faculty of Computer Science, University of Twente y Computer Science Department, University of Nijmegen.

[2] Fabian, M. y Hellgren, A. (1998): "PLC-based Implementation of Supervisory Control for Discrete Event Systems". Control Engineering, Department of Signals and Systems Chalmers University of Technology, SWEDEN.

[3] Park, E., Tilbury, D. y Khargonekar, P. (2000): "Modeling, Analysis, And Implementation Of Logic Controllers For Machining Systems Using Petri Nets And SFC". NSF Engineering Research Center for Reconfigurable Machining Systems.

[4] Buy, U. y Darabi, H. (2003): "Sidestepping verification complexity with supervisory control". University of Illinois at Chicago.

[5] Americam National Standard. (1995): ANSI/ISA–88.01–1995, "Batch Control Part 1:Models and Terminology".

[6] Indriago, Miguel. (2004): Automatización de procesos industriales. Reporte Técnico Universidad de los Andes. Mérida. Venezuela.

[7] Lucena, Z. (2004): "Software para la Conversión de una Filosofía de Control a un Esquema de Programación Escalera". Tesis de maestría. Unexpo. Barquisimeto. Venezuela.

[8] Zapata, G., Carrasco, E. (2002): "Estructuras Generalizadas Para Controladores Lógicos Modeladas Mediante Redes De Petri". Escuela de Ingeniería Eléctrica y Mecánica Universidad Nacional de Colombia sede Medellín.

[9] Hellgren, A., Lennartson, B. y Fabian, M. (2002): "Modelling and PLC-based Implementation of Modular Supervisory Control". Department of Signals and Systems Chalmers University of Technology , Sweden.

[10] Liu, J. y Darabi, H. 2002: "Ladder Logic Implementation of Ramadge-Wonham Supervisory Controller". Proceedings of the Sixth International Workshop on Discrete Event Systems IEEE.

[11] ROCKWELL AUTOMATION. (2001): SLC 500™ Instruction Set.