



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA MECÁNICA - ELÉCTRICA

**DISEÑO DE AUTOMATIZACIÓN DE ROTATIVA DE IMPRESIÓN  
GOSS UTILIZANDO LA PLATAFORMA CONTROL LOGIX**

**German Contreras Jacinto**

**Asesorado por ing. Enrique Edmundo Ruiz Carballo**

**Guatemala, agosto de 2005**

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**AUTOMATIZACIÓN DE ROTATIVA DE IMPRESIÓN GOSS UTILIZANDO  
LA PLATAFORMA CONTROL LOGIX**

TRABAJO DE GRADUACIÓN

PRESENTADO A JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**GERMAN CONTRERAS JACINTO**

ASESORADO POR ING. ENRIQUE EDMUNDO RUIZ CARBALLO

AL CONFERÍRSELE EL TÍTULO DE  
**INGENIERO EN ELECTRÓNICA**

GUATEMALA, AGOSTO DE 2005

**UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**  
**FACULTAD DE INGENIERÍA**



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	
VOCAL II	Lic. Amahán Sánchez Álvarez
VOCAL III	Ing. Julio David Galicia Celada
VOCAL IV	Br. Kenneth Issur Estrada Ruiz
VOCAL V	Br. Elisa Yazminda Vides Leiva
SECRETARIA	Inga. Marcia Ivonne Véliz Vargas

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Sydney Alexander Samuels Milson
EXAMINADOR	Ing. Otto Andrino
EXAMINADOR	Inga. Ingrid de Loukota
EXAMINADOR	Ing. Julio César Solares Peñate
SECRETARIO	Ing. Pedro Antonio Aguilar Polanco

## **HONORABLE TRIBUNAL EXAMINADOR**

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

### **AUTOMATIZACIÓN DE ROTATIVA DE IMPRESIÓN GOSS UTILIZANDO LA PLATAFORMA CONTROL LOGIX,**

tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica – Eléctrica con fecha 24 de Enero de 2004.

**German Contreras Jacinto**

## **AGRADECIMIENTOS A:**

- MI DIOS:** por haberme dado la oportunidad de conocerlo, por enseñarme el propósito de mi vida y que mis virtudes y cualidades son un regalo que él me concedió para que mis sueños convergieran con los suyos y a través de mis debilidades, él se glorifique. Gracias, infinitamente gracias;
- MI PADRE:** por cuidarme, guiarme, enseñarme principios que hoy son parte de mi vida. Por que sé que mi vida es el reflejo de su enseñanza y sus virtudes;
- MI MADRE:** por su amor, tolerancia, comprensión, gracias por apoyarme en todo momento, porque en mis frustraciones tu me has enseñado a levantarme y seguir mi camino;
- MARTA:** por que Dios te ha puesto en mi camino como ángel, porque me has ayudado, lo que soy y seré son el fruto de tu esfuerzo;
- MIS HERMANOS:** Marta, Rolando, Eduardo, Daniel, Ana gracias a cada uno porque he aprendido de ustedes y han contribuido en la formación de mi vida;
- MIS AMIGOS:** los de las chamuscas, laboratorio, academia de inglés, grupo de amistad de Casa de Dios, gracias por enseñarme el valor de la amistad;
- MI ASESOR:** gracias por brindarme su valioso tiempo y experiencia académica, por hablarme con sus actitudes ejemplares.

## **DEDICATORIA A:**

- MI DIOS:** porque he aprendido que nunca he estado solo, has caminado a mi lado en los malos y buenos momentos de mi vida y a través de tu palabra guías y formas mi camino;
- MI ALMA:** porque has aprendido a escuchar y obedecer la voz de Dios y sabes que es mejor terminar algo que empezarlo y que la humildad, sabiduría y esfuerzo te llevarán a donde nunca pensaste que llegarías, como hoy;
- MI FAMILIA Y AMIGOS:** porque los desafíos que se presentan en la vida son los que marcan etapas de cambio en la vida, y cuando se presenta uno existe la oportunidad de caminar un escalón más;
- MI NACIÓN** porque desde el primer día de clase en la Universidad en que llegué tarde y no encontraba el salón de clases se me confió y se me dio la responsabilidad de terminar la carrera.

## ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	VII
LISTA DE ABREVIACIONES Y SÍMBOLOS	XI
GLOSARIO	XIII
RESUMEN	XVI
OBJETIVOS	XVII
INTRODUCCIÓN	XVIII
FIGURAS	VII
TABLAS	IX
<b>1 PLATAFORMA CONVENCIONAL</b>	<b>1</b>
1.1 Introducción a los controladores lógicos programables	1
1.1.1 Definición del controlador lógico programable	1
1.2 Principios de operación	2
1.3 Arquitectura del controlador lógico programable	5
1.3.1 Procesadores	5
1.3.2 Tiempo de exploración	7
1.4 Sistema de memoria	9
1.4.1 Organización de memoria e interacción con E/S	11
1.4.1.1 Memoria del sistema	11
1.4.1.2 Memoria de aplicación	13
1.4.1.2.1 Área de tabla de datos	13
1.4.1.2.2 Área de programa de usuario	16
1.4.2 Configuración de memoria – Direccionamiento de E/S	16
1.4.2.1 Organización de tabla de datos	16

1.4.2.1.1	Direccionamiento de E/S	16
1.5	Implementación y diseño	17
1.5.1	Definición de tarea de control	17
1.5.2	Estrategia de control	17
1.5.3	Guía de implementación	18
1.5.4	Organización del programa e implementación	20
1.5.4.1	Creación del diagrama de flujo y secuencia de salidas	20
1.5.4.2	Configuración del controlador	22
1.5.4.3	Direccionamiento de E/S	22
1.5.4.4	Asignación en la tabla de registros	24
1.5.4.5	Documentación del programa	25
1.5.4.5.1	Resumen del sistema	26
1.5.4.5.2	Configuración del sistema	27
1.5.4.5.3	Diagrama de conexión de E/S	27
1.5.4.5.4	Asignación de direcciones de E/S	27
1.5.4.5.5	Asignación de direcciones de registros internos	28
1.5.4.5.6	Tabla de asignación de registros	29
1.5.4.5.7	Declaración de variables	29
1.5.4.5.8	Programa de control impreso	30
1.5.4.5.9	Almacenamiento del programa de control	30
<b>2</b>	<b>PLATAFORMA CONTROL LOGIX</b>	<b>31</b>
2.1	Arquitectura controlador Control Logix	31
2.2	Organización de tags	32
2.2.1	Estructura de tags	32
2.2.2	Tipo de tag	33
2.2.3	Tipo de dato	33
2.2.4	Alcance	34
2.3	Características de la plataforma Control Logix	35



2.3.1	Programación basada en direccionamiento con tags	35
2.3.2	Estructuras de datos	36
2.3.3	Diseño e implementación	37
2.3.4	Modularidad de código de programa	39
2.4	Componentes de un proyecto de software	40
2.4.1	Tareas	40
2.4.1.1	Tareas continuas	41
2.4.1.2	Tareas periódicas	41
2.4.1.3	Tareas de eventos	41
2.4.2	Programas	42
2.4.3	Rutinas	42
2.4.4	Rutina principal	44
2.4.5	Subrutina	44
2.5	Configuración de la ejecución de una tarea	44
2.6	Ejecución de múltiples programas	46
2.7	Redundancia de equipo	46
2.7.1	Falla de corte	48
2.7.2	Sincronización	48
2.8	Comunicación con E/S	49
2.8.1	Intervalo de solicitud de paquete RPI	49
2.8.2	Formato de comunicación	50
2.8.2.1	Conexión directa	50
2.8.2.2	Conexión de chasis optimizado	51
2.8.2.3	Pertenencia	52
2.9	Modelo productor consumidor	53
2.9.1	Tags producidos	54
2.9.2	Tags consumidos	54
<b>3</b>	<b>CONSIDERACIÓN DE LOS PARÁMETROS DE DISEÑO</b>	<b>57</b>

3.1	Criterios de uso de tareas, programas y rutinas	57
3.1.1	Prioridad de tareas	57
3.1.2	Factores que afectan la ejecución de tareas	57
3.1.2.1	Sistema de encabezado de información	57
3.1.2.2	Procesamiento de salidas	58
3.1.2.3	Número de tareas	58
3.1.3	Configuración de una tarea continua	59
3.1.4	Configuración de una tarea periódica	59
3.1.5	Configuración de una tarea de eventos	60
3.2	Desarrollo de aplicaciones en rutinas	60
3.2.1	Lógica en escalera	60
3.2.2	Diagramas de bloques de funciones	61
3.2.3	Diagrama de secuencia de funciones	61
3.2.4	Texto estructurado	61
3.3	Métodos de programación	61
3.3.1	Programación inline	61
3.3.2	Programación indexada	62
3.3.3	Programación buffered	63
3.4	Direccionamiento de datos	64
3.4.1	Tipo de datos atómicos	65
3.4.2	Tipo de datos compuestos	65
3.5	Comunicación en una red Ethernet	66
3.5.1	Protocolo TCP/IP	66
3.5.2	Parámetros en una red Ethernet	66
3.5.2.1	Dirección IP	66
3.5.2.2	Pasarela	67
3.5.2.3	Máscara de subred	67
3.6	Comunicación en una red RIO	68
3.6.1	Parámetros en una red RIO	69

3.6.1.1	Velocidad de transmisión	69
3.6.1.2	Direccionamiento	70
3.7	Variador de velocidad de C.D. 1395	70
3.7.1	Descripción del variador de velocidad de C.D. 1395	70
3.7.2	Programación de parámetros	72
3.7.2.1	Estructura de la tabla de parámetros	72
3.7.2.2	Almacenamiento de la tabla de parámetros	73
3.7.3	Configuración de parámetros	74
3.7.3.1	Parámetros sumideros	74
3.7.3.2	Parámetros fuente	76
3.8	Interfaz humano máquina	76
3.8.1	Plataforma dedicada	76
3.8.2	Panel View 1500	77

## **4 DISEÑO DE LA INGENIERIA DE CONTROL**

### **DE LA ROTATIVA DE IMPRESIÓN 79**

4.1	Antecedentes	79
4.2	Planteamiento del problema	80
4.3	Diseño de la ingeniería de control de la rotativa de impresión	80
4.3.1	Descripción del sistema	80
4.3.2	Componentes de hardware	82
4.3.2.1	Arquitectura del sistema	82
4.3.2.2	Selección de módulos discretos de E/S	83
4.3.2.3	Selección de la red de comunicación	84
4.3.2.3.1	Red de comunicación Ethernet/IP	84
4.3.2.3.2	Red universal de comunicación de E/S remotas	86
4.3.2.4	Selección de controlador	86
4.3.2.4.1	Requerimientos de memoria	87
4.3.2.5	Selección de chasis	88

4.3.2.6	Selección de fuente de alimentación	89
4.3.3	Componentes de Software	89
4.3.3.1	RSLogix 5000	89
4.3.3.2	RSView Studio	91
4.3.3.3	RSNetworx	92
4.3.3.4	Software RSLinx	92
4.3.4	Programación del autómeta	93
4.3.4.1	Rutina principal	94
4.3.4.2	Rutina control de variadores de velocidad	96
4.3.4.3	Rutina de control de unidades	98
4.3.4.4	Rutina control doblador	98
4.3.4.5	Rutina control de enkels	100
4.3.5	Programación de la interfaz humano máquina	100
4.3.5.1	Pantalla rotativa de impresión GOSS	101
4.3.5.2	Pantalla arquitectura de rotativa de impresión GOSS	102
4.3.5.3	Pantalla torre de transmisión	103
4.3.6	Puesta en marcha	104

# ÍNDICE DE ILUSTRACIONES

## FIGURAS

1	Diagrama conceptual de aplicación	1
2	Diagrama de bloques de un controlador lógico programable	2
3	Diagrama de bloques de los principales componentes de la CPU	2
4	Ilustración de un ciclo de exploración	3
5	Interfase de E/S	4
6	Diagrama de bloques de la CPU	5
7	Una configuración multiprocesador	6
8	Representación de un ciclo de exploración total	7
9	Ilustración de una señal que no será detectada durante un de ciclo de exploración normal	9
10	Diagrama de aplicación de bloques simplificado de la memoria total de un PLC	10
11	Diagrama simplificado del mapa de memoria de un PLC	11
12	Mapa de la memoria de aplicación	13
13	Área de almacenamiento de la tabla de datos	15
14	Diagrama de flujo del desarrollo de una aplicación de automatización convencional	14
15	Símbolos de diagrama de flujo	15
16	Arquitectura del controlador Control Logix	31
17	Estructura de un proyecto y alcance de los tags	17
18	Diferencia entre la plataforma Control Logix y la convencional en el diseño e implementación de un proyecto de automatización	38
19	Ejecución de una tarea continua	44

20	Ejecución de una tarea periódica	45
21	Organización de dos programas sin conflictos de ejecución	46
22	Sistema redundante Control Logix	47
23	Transferencia de tags en una conexión directa	51
24	Transferencia de tags en una conexión de chasis optimizado	52
25	Compartimiento de tags en un sistema Control Logix	55
26	Ejemplo de programación inline	62
27	Ejemplo de programación indexada	63
28	Ejemplo de programación buffered	64
29	Pasarela que conecta dos redes	67
30	Máscaras de subredes por defecto	68
31	Ejemplo de una red RIO	68
32	Arquitectura interna del drive 1395	71
33	Estructura de la tabla de parámetros	72
34	Parámetros fuente y sumidero de un drive 1395	75
35	Rotativa de impresión GOSS	81
36	Arquitectura de red de comunicación de rotativa de impresión GOSS	82
37	Modelo productor consumidor para E/S	83
38	Módulo digital 1756 de E/S	84
39	Ubicación del controlador, módulos de comunicación y módulos de E/S	88
40	Software de programación RSLogix 5000	90
41	Estructuras: Unidad, drive, torre, y transmisión	93
42	Palabra Logic Status	96
43	Palabra Logic Command	96
44	Doblador de periódicos	99
45	Pantalla rotativa de impresión GOSS	101
46	Pantalla de arquitectura de rotativa de impresión GOSS	102
47	Pantalla de torre de rotativa de impresión GOSS	103

## TABLAS

I.	Guía de desarrollo de aplicación para PLCs convencionales	19
II.	Asignación de direcciones de E/S	23
III.	Asignación de direcciones para registros internos	23
IV.	Asignación en la tabla de registros	24
V.	Componentes de una buena documentación	26
VI.	Asignación de direcciones de E/S	28
VII.	Asignación de direcciones de registros internos	29
VIII.	Tipos de datos de tags	33
IX.	Bits utilizados en diversas estructuras definidas	34
X.	Ventajas de la plataforma Control Logix sobre la convencional en el direccionamiento de datos	36
XI.	Ventajas de la plataforma Control Logix sobre la convencional al usar estructuras definidas por el usuario	37
XII.	Ventajas de la plataforma Control Logix sobre la convencional en el diseño e implementación de proyectos	38
XIII.	Ventajas de la plataforma Control Logix sobre la convencional en la modularidad de programación	40
XIV.	Consideración de uso de tareas, programas y rutinas	43
XV.	Opciones de transferencia de datos en la plataforma Control Logix	54
XVI.	Beneficios y consideraciones de datos atómicos	65
XVII.	Beneficios y consideraciones de datos compuestos	65
XVIII.	Distancia máxima en una red RIO	69
XIX.	Parámetros utilizados para el control del drive	74
XX.	Parámetros del Panel View 1500	77
XXI.	Especificaciones eléctricas de módulos discretos de E/S	84
XXII.	Especificaciones módulo de comunicación Ethernet,1756-ENBT	85

XXIII.	Especificaciones de módulo de comunicación 1756-DHRIO	86
XXIV.	Especificaciones del controlador Control Logix 1756-L61	87
XXV.	Estimación de uso de memoria	87
XXVI.	Ubicación el controlador, módulos de comunicación y módulos de E/S	88
XXVII.	Especificaciones eléctricas de la fuente de alimentación	89
XXVIII.	Parámetros de red Ethernet de los nodos de la rotativa de impresión	92



## LISTA DE ABREVIACIONES Y SÍMBOLOS

<b>AC</b>	<i>Altern Current</i> : corriente alterna.
<b>ASCII</b>	<i>American Standard Code of Information Interchange</i> : código estándar americano para el intercambio de información.
<b>COS</b>	<i>Change of State</i> : cambio de estado.
<b>CPU</b>	<i>Central Process Unit</i> : unidad central de proceso del controlador lógico programable.
<b>DC</b>	<i>Direct Current</i> : corriente directa.
<b>DH+</b>	<i>Data Highway Plus</i> : red de comunicación propiedad de Allen Bradley.
<b>EEPROM</b>	<i>Electrically Erasable Programmable Read Only Memory</i> : memoria de lectura, programable y borrrable eléctricamente.
<b>E/S</b>	Entradas salidas.
<b>EOS</b>	<i>Exploration Output Scan</i> : tiempo de exploración de salidas.
<b>HMI</b>	<i>Human – Machine Interfase</i> : interface humano – máquina.
<b>HP</b>	<i>Horse Power</i> : caballos de potencia.
<b>IEC</b>	<i>International Electrotechnical Comisión</i> : Comisión Eléctrica Internacional.
<b>IP</b>	<i>Internet Protocol</i> : protocolo de internet.
<b>JSR</b>	<i>Jump to subrutina</i> : instrucción de salto a una subrutina.
<b>MCR</b>	<i>Master Clear Reset</i> : instrucción de restablecimiento de ciclo de exploración.
<b>PC</b>	<i>Personal Computer</i> : computadora personal
<b>PLC</b>	<i>Programmable Logic Controller</i> : controlador lógico programable.
<b>TCP</b>	<i>Transmission Control Protocol</i> : protocolo de control de transmisión.
<b>RAM</b>	<i>Random Aleatory Memory</i> : memoria de sólo lectura.
<b>RIO</b>	<i>Remote Input Output</i> : red de comunicación remota de entradas salidas.

**RPI**

*Request Packet Interval:* intervalo de tiempo de actualización de información de estado de módulos de E/S.

## GLOSARIO

<b>Aplicación:</b>	combinación de rutinas, programas, tareas, configuración de E/S utilizada para definir la operación de un controlador;
<b>Conexión:</b>	enlace de comunicación entre dos dispositivos, por ejemplo, un controlador y un módulo de E/S, Panel View u otro controlador;
<b>Conexión directa:</b>	enlace de transferencia de datos en tiempo real entre un controlador y un módulo de E/S;
<b>Estructura:</b>	una estructura almacena un grupo de datos, cada uno de los datos puede ser de diferente tipo de dato;
<b>Ethernet:</b>	red abierta de comunicación a nivel de información;
<b>Hub:</b>	concentrador de conexiones, comúnmente usado en Ethernet;
<b>Maestro – esclavo:</b>	modelo de comunicación donde el maestro habla y los esclavos responden;
<b>Nodo:</b>	dispositivo identificado en una red, capaz de comunicarse;
<b>NetLinx:</b>	redes abiertas de comunicación en la automatización industrial: Ethernet/IP, Controlnet y Devicenet;

<b>Offline:</b>	edición de un proyecto que está almacenado en una computadora;
<b>Productor – consumidor:</b>	modelo reciente de comunicación, donde los mensajes que se producen en una estación pueden ser consumidos por cualquier otra;
<b>Proyecto:</b>	archivo almacenado en una computadora personal que almacena la lógica, configuración, datos y documentación de un controlador;
<b>RSLogix 5000:</b>	software de programación en escalera, diagrama de bloques de funciones, texto estructurado para controladores de la plataforma Logix, de Rockwell Software Inc.;
<b>RSNetwork:</b>	software de configuración de redes, de Rockwell Software Inc.;
<b>RSView Studio:</b>	software de edición de representaciones visuales de un proceso o máquina, de Rockwell Software Inc.;
<b>Tag:</b>	área de memoria del controlador donde se almacena un dato;
<b>Tag base:</b>	tag que define la localidad de memoria donde es almacenado un dato;
<b>Tag consumido:</b>	tag que recibe los datos transmitidos por un tag producido;
<b>Tag global:</b>	tag accesible en cualquier parte del proyecto;

<b>Tag consumido:</b>	tag que un controlador permite compartir con otros controladores;
<b>Tag seudónimo:</b>	un tag que representa otro tag;
<b>Tarea:</b>	mecanismo calendarizado para la ejecución de un programa;
<b>Tarea continua:</b>	tarea que se ejecuta continuamente;
<b>Tarea de eventos:</b>	tarea que se ejecuta cuando ocurre un evento específico;
<b>Tarea periódica:</b>	tarea que se ejecuta por el sistema operativo del controlador a un periodo de tiempo repetitivo.



## **RESUMEN**

El presente trabajo de graduación trata acerca de la introducción de la Tecnología de automatización Control Logix en la industria guatemalteca, como una opción respecto de la plataforma convencional que se ha usado por décadas en Guatemala.

La plataforma convencional se ha utilizado en variedad de procesos y máquinas; y, ha sido funcional, eficaz y eficiente, sin embargo, la plataforma Control Logix tiene incorporado una variedad de ventajas y principalmente en el proceso de diseño que ha empezado a ser mucho más atractiva que la convencional.

Las características de la plataforma convencional se explican en el capítulo 1, y, de la Control Logix en el capítulo 2, siendo éstos dos capítulos los que hacen énfasis en sus diferencias. En el capítulo 3 se detallan los parámetros a considerar en el diseño y en el capítulo 4 el diseño de ingeniería de control de la rotativa de impresión Goss.





## **OBJETIVOS**

### **Generales**

Dar a conocer en Guatemala la plataforma Control Logix respecto de la convencional, a través del diseño de la automatización de la rotativa de impresión Goss.

### **Específicos**

1. Dar a conocer las ventajas de la de la plataforma Control Logix respecto de la convencional.
2. Considerar los parámetros para el diseño de la automatización de la rotativa de impresión Goss al utilizar la plataforma Control Logix.
3. Diseñar la ingeniería de control de la rotativa de impresión Goss utilizando la plataforma Control Logix.



## INTRODUCCIÓN

En Guatemala, se han implementado proyectos de automatización con equipos que durante años han sido confiables y han suplido las necesidades de las industrias guatemaltecas. Sin embargo, considerando la demanda y la creciente competitividad global, Guatemala se ve en la necesidad de utilizar e investigar nuevas tecnologías que estén a la vanguardia, para competir en el ámbito internacional en la calidad, eficiencia, productividad, etc, de productos y servicios.

La plataforma Control Logix provee estas características de desempeño, pero existe cierta barrera ante nuevas tecnologías en la industria guatemalteca, ya que, se requiere de inversión inicial de recursos materiales, económicos, humanos y tiempo en la introducción de esta tecnología en Guatemala.

El autor de este trabajo trata de mostrar la justificación del uso de esta tecnología, a través del diseño de la automatización de la rotativa de impresión Goss utilizando la plataforma Control Logix, en donde se pueden apreciar las ventajas a través de la comparación con la plataforma convencional, los parámetros a considerar en el diseño, y, el diseño para el desarrollo de proyectos de automatización utilizando esta tecnología.



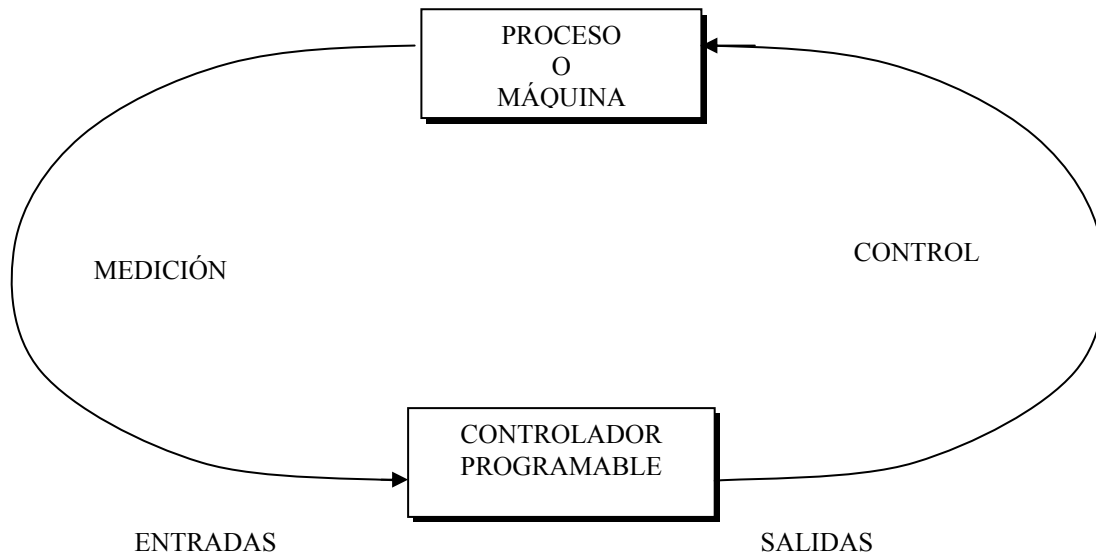
# 1 PLATAFORMA CONVENCIONAL

## 1.1 Introducción a los controladores lógicos programables

### 1.1.1 Definición del controlador lógico programable

Los controladores lógicos programables, también, llamados PLCs, son dispositivos de estado sólido de la familia de computadoras, usan circuitos integrados en lugar de dispositivos electromecánicos para implementar control de funciones. Estos son capaces de almacenar instrucciones, tales como: secuencias, temporizadores, contadores, aritmética, manipulación de datos, y, comunicación, para controlar máquinas industriales y procesos. La siguiente figura ilustra un diagrama conceptual de una aplicación.

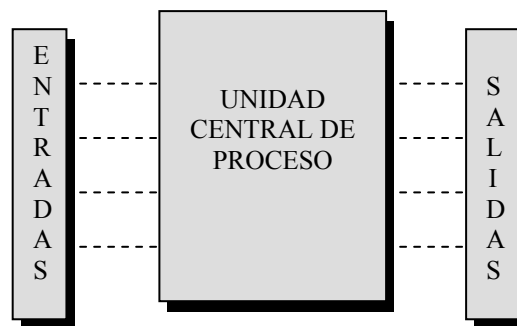
**Figura 1. Diagrama conceptual de aplicación**



## 1.2 Principios de operación

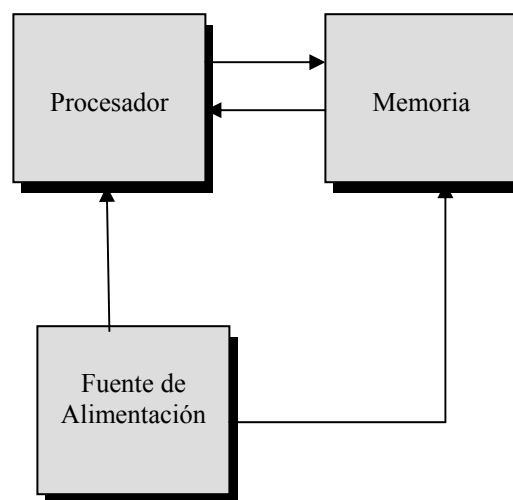
Un controlador programable, como se ilustra en la figura 2, consiste de dos secciones básicas: Unidad central de proceso e interfase de E/S.

**Figura 2. Diagrama de bloques de un controlador lógico programable**



La unidad central de proceso (CPU) gobierna todas las actividades del PLC. Los siguientes tres componentes, mostrados en la siguiente figura 3, forman la CPU.

**Figura 3. Diagrama de bloques de los principales componentes de la CPU**

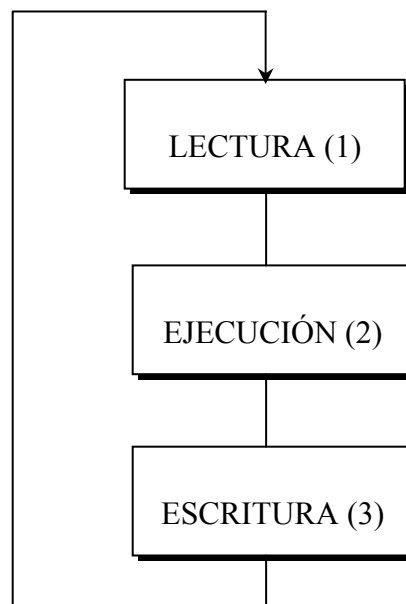


La operación de un controlador lógico programable es relativamente sencilla. El sistema de E/S está, físicamente, conectado a los dispositivos de campo, los cuales se encuentran en la máquina o son usados en el control del proceso. Estos dispositivos de campo pueden ser E/S discretas o analógicos, tales como: interruptores de límite, transductores de presión, pulsadores, arrancadores de motor, solenoides, etc.

Durante su operación, la CPU completa tres procesos: (1) lee, o acepta, los datos de entrada de los dispositivos de campo vía la interfase de entrada, (2) ejecuta, o realiza, el programa de control almacenado en la memoria del sistema y (3) escribe, o actualiza, los dispositivos de salida vía la interfase de salida.

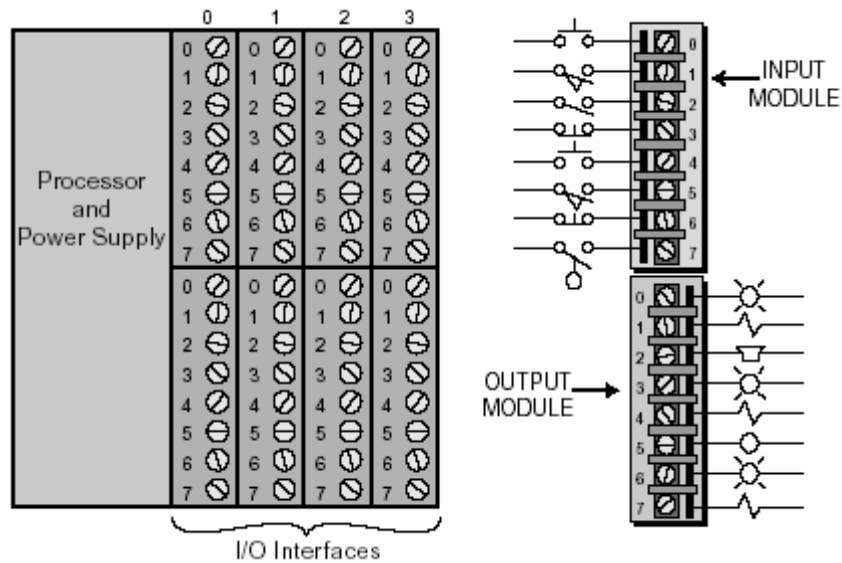
Este proceso de lectura de datos, ejecución del programa en memoria y actualización de las salidas es conocido como exploración. La figura 4 ilustra una representación gráfica de un ciclo.

**Figura 4. Ilustración de un ciclo de exploración**



El sistema de E/S forma la interfase, a través de la cual dispositivos son conectados al controlador, como se muestra en la figura 5. El principal propósito de la interfase es de evaluar las varias señales recibidas de o enviadas a dispositivos externos en el campo. Las señales provenientes de sensores, pulsadores, interruptores de límite, sensores análogos, etc, son cableadas a las terminales en la interfase de entrada. Los dispositivos que serán controlados, tales como: arrancadores de motores, solenoides de válvulas, luces piloto y válvulas de posición son conectados a las terminales de la interfase de salida. El sistema de alimentación provee todos los voltajes requeridos para la apropiada operación de las secciones de unidad de procesamiento.

**Figura 5. Interfase de E/S**



Fuente: L.A. Bryan, “Controladores lógicos programables”, 2da edición, pág. 12



### 1.3 Arquitectura del controlador lógico programable

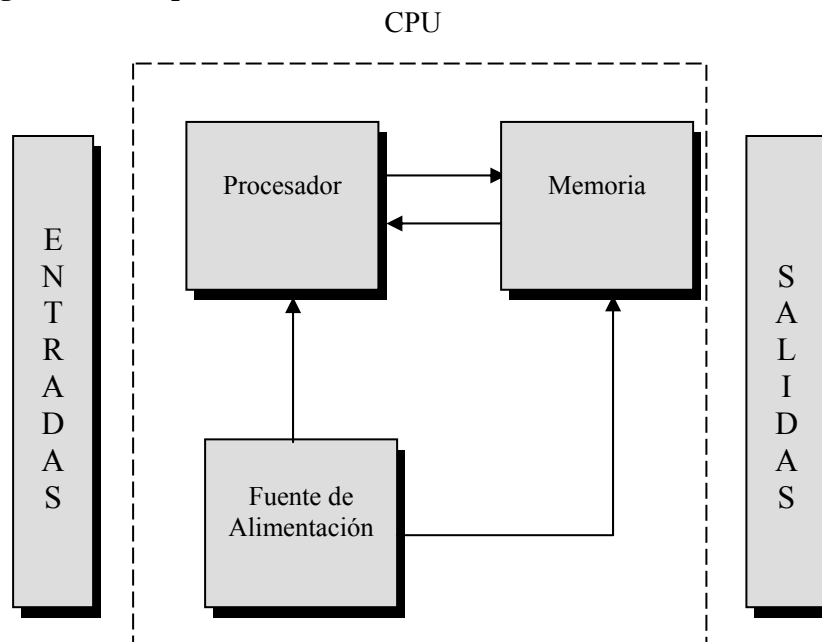
#### 1.3.1 Procesadores

La unidad central de procesamiento o CPU, es el elemento más importante de un PLC. La CPU puede ser considerada el cerebro del sistema. Los tres componentes de la CPU son:

- procesador;
- sistema de memoria;
- fuente de alimentación.

En la figura 6 se ilustra un diagrama de bloques simplificado. La arquitectura puede diferir de un fabricante a otro, pero, en general, la mayoría de CPUs siguen esta organización de tres componentes.

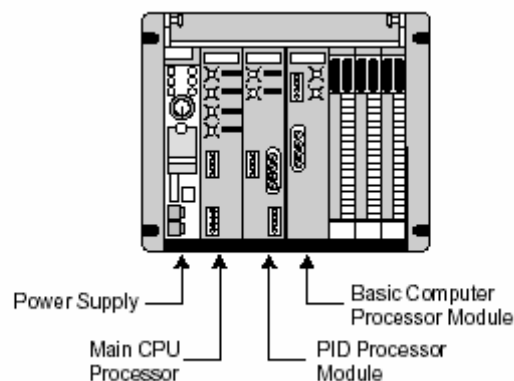
**Figura 6. Diagrama de bloques de la CPU**



La principal función del procesador es comandar y gobernar las actividades del sistema completo. Este desempeña esta función interpretando y ejecutando una colección de programas de sistema conocido como ejecutable. El ejecutable es un grupo de programas supervisores que está, permanentemente, almacenado en el procesador y es considerado parte del procesador. Ejecutando este programa, el procesador puede ejecutar todo el control, procesamiento, comunicación, etc.

La CPU de un PLC puede contener más de un procesador para ejecutar las operaciones del procesador y / o comunicaciones, los extra procesadores incrementan la velocidad de estas operaciones. Esta ventaja de utilizar diferentes microprocesadores para dividir tareas de control y comunicación es conocida como **multiprocesamiento**. La figura 7 ilustra una configuración multiprocesador.

**Figura 7. Una configuración multiprocesador**



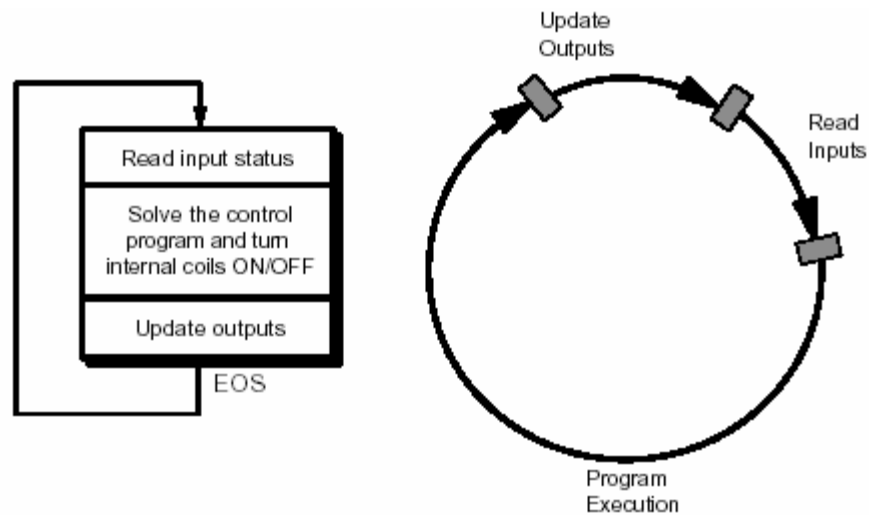
Fuente: L.A. Bryan, “Controladores lógicos programables”, 2da edición, pág. 85

Los microprocesadores usados en PLCs son caracterizados, de acuerdo al tamaño de palabra o el número de bits que usan para ejecutar operaciones, simultáneamente. Palabras estándar son 8, 16 y 32 bits. Este tamaño de palabra afecta la velocidad a la que el procesador ejecuta la mayoría de operaciones.

### 1.3.2 Tiempo de exploración

La función básica de un controlador programable es leer todas las entradas y entonces ejecutar el programa de control, el cual acorde a la lógica programada, activará o desactivará los dispositivos de salida. En realidad, este último proceso de activar o desactivar los dispositivos de salida ocurre en dos pasos. Primero, cuando el procesador ejecuta la lógica interna programada, activará o desactivará las bobinas internas programadas. Energizar o desenergizar estas salidas internas, no cambiará el estado de los dispositivos de salida. Entonces, cuando el procesador ha finalizado de evaluar toda la lógica de control del programa activará o desactivará las bobinas internas, ejecutará una actualización del módulo de interfase de salida, por lo tanto, cambiará el estado de los dispositivos conectados a la interfase de salida. Este proceso de lectura de las entradas, ejecución del programa y actualización de las salidas es conocido como exploración.

**Figura 8. Representación de un ciclo de exploración total**



Fuente: L.A. Bryan, “Controladores lógicos programables”, 2da edición, pág. 86.

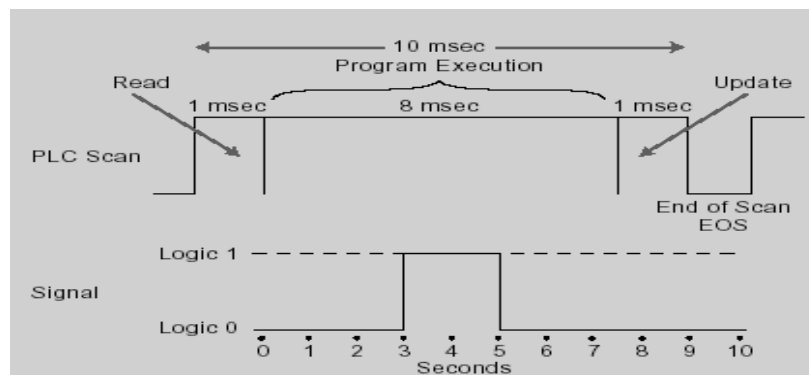
En la figura anterior se puede observar una representación gráfica de un ciclo de exploración. El proceso de exploración es repetido de la misma manera, cíclicamente, haciendo secuencial la operación desde el inicio hasta el final. Algunas veces, a manera de simplicidad, los fabricantes de PLCs llaman a la ejecución del programa de control **exploración del programa** y la lectura y actualización de salidas **exploración de actualización de E/S**. Sin embargo, el ciclo de exploración total del sistema incluye ambos. La señal interna del procesador, la cual indica el final del ciclo de exploración del programa, es llamada señal de final de la exploración EOS, siglas en inglés.

El tiempo que toma para implementar un ciclo de exploración es llamado tiempo de exploración. El tiempo de exploración es el tiempo total que el PLC toma para completar el programa y ejecutar la exploración de actualización de E/S. La exploración del programa de control, generalmente, depende de dos factores: (1) la cantidad de memoria utilizada por el programa de control y (2) el tipo de instrucciones usadas en el programa, el cual afecta el tiempo necesario para ejecutar las instrucciones. El tiempo para realizar un ciclo de exploración puede variar de unas pocas décimas hasta 50 milisegundos.

Manufacturadores de PLCs especifican el tiempo de exploración, basado en la cantidad de memoria de aplicación usada, 1 msec/1K de memoria programada. Sin embargo, otros factores, también, afectan este parámetro. El uso de subsistemas remotos de E/S puede incrementar el tiempo de exploración, desde que el PLC transmite y recibe la actualización de E/S de sistemas remotos. El monitoreo del programa de control, también, incrementa el tiempo de exploración porque el microprocesador debe enviar datos acerca del estado de las bobinas y contactos para dispositivos de monitoreo (PC).

El ciclo de exploración es, normalmente, continuo, un proceso secuencial de lectura del estado de entradas, evaluando la lógica de control y actualización de salidas. Un procesador es capaz de leer una entrada hasta el punto en que esta no sea más rápida que el tiempo de exploración. Por ejemplo, si un controlador tiene un tiempo total de 10 mseg, como se aprecia en la figura 9, y debe monitorear una señal de entrada que cambia de estado dos veces durante un período de 8 mseg, menos del tiempo de exploración, el controlador programable no será capaz de “ver” la señal, resultando en un posible mal funcionamiento del proceso o máquina. Esta característica debe ser considerada siempre cuando lee señales de entrada y caracteres ASCII.

**Figura 9. Ilustración de una señal que no será detectada durante un ciclo de exploración normal**



Fuente: L.A. Bryan, “Controladores lógicos programables”, 2da edición, pág. 87

#### 1.4 Sistema de memoria

La característica más importante de un controlador programable es la capacidad de cambiar el programa de control rápida y fácilmente. La memoria de sistema es el área en la CPU donde todas las secuencias de instrucciones o programas son almacenadas y ejecutadas por el procesador, para proveer el control deseado de los dispositivos de campo.

Las secciones de memoria que contienen el programa de control puede ser cambiado, o, reprogramado, para adaptar el procedimiento de la línea de manufactura o requerimientos del nuevo sistema. El sistema total de memoria en un PLC esta compuesto de dos diferentes memorias, como se observa en la figura 10.

- Memoria de ejecución.
- Memoria de aplicación.

**Figura 10. Diagrama de bloques simplificado de la memoria total de un PLC**



La **memoria de ejecución** es una colección de programas, permanentemente, almacenados, considerados parte propia del PLC. Estos programas supervisores de la actividad del sistema, tal como la ejecución del programa de control y comunicación con dispositivos periféricos. La sección de ejecución es la parte de la memoria del PLC donde están almacenadas las instrucciones de software, instrucciones básicas, funciones de transferencia de bloques, instrucciones matemáticas, etc. Esta área de memoria es no accesible por el usuario.

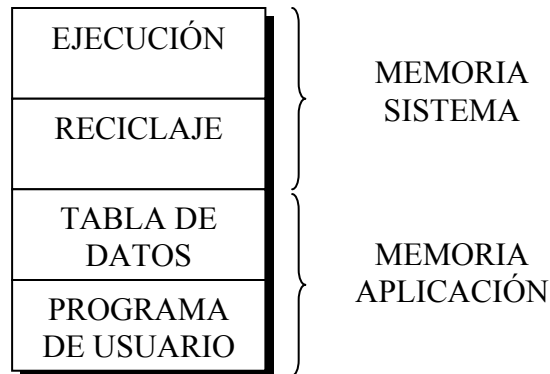
La **memoria de aplicación** provee un área de almacenamiento para las instrucciones programadas, por el usuario, que forman parte del programa de aplicación. El área de la memoria de aplicación esta compuesta de varias áreas, cada una con una función y uso específico.

## 1.4.1 Organización de memoria e interacción con E/S

### 1.4.1.1 Memoria del sistema

La memoria del sistema está compuesta de 2 secciones principales, las cuales están compuestas en otras áreas. En la figura 11 se observa la organización de memoria, conocida como mapa de memoria. Aunque las dos secciones principales, memoria del sistema y memoria de aplicación están juntas en la figura, no, necesariamente, son adyacentes, físicamente o por dirección. El mapa de memoria no sólo muestra que está almacenado en memoria, sino, también, donde están almacenados los datos, de acuerdo a las localidades específicas llamadas direcciones de memoria. Entender el mapa de memoria es muy útil cuando creamos el programa del PLC y definimos la tabla de datos.

**Figura 11. Diagrama simplificado del mapa de memoria de un PLC**



Aunque dos controladores diferentes, raramente, tienen idénticos mapas de memoria, estos deben tener cuatro áreas básicas de memoria, las cuales son las siguientes.

- **Área de programa de usuario:** esta área provee almacenamiento para instrucciones programadas por el usuario. En esta también se almacena el programa de control.

- **Área de ejecución:** esta área es una colección de programas permanentemente almacenados, considerados parte del propio sistema. Estos programas supervisan las actividades del sistema, como la ejecución del programa de control, comunicación con dispositivos periféricos, y otro tipo de actividades del sistema.
- **Área de reciclaje:** esta es una área de almacenamiento temporal usada por la CPU para almacenar una cantidad relativamente pequeña de datos para cálculos internos y control. La CPU almacena datos en esta área de memoria por corto tiempo para evitar prolongar el tiempo de acceso al manipular datos de la memoria principal.
- **Área de tabla de datos:** esta área almacena todos los datos asociados con el programa de control, como datos de temporizadores, contadores y otras constantes y variables usadas por el programa de control o la CPU. La tabla de de datos también almacena información del estado de los sistemas de entradas, una vez hayan sido leídas, y salidas, una vez hayan sido activadas por el programa de control.

El área de ejecución y reciclaje son áreas ocultas al usuario y pueden ser consideradas una única área de memoria, para nuestros propósitos, es llamada **memoria del sistema**. Por otro lado, la tabla de datos y programa de usuario son accesibles y son requeridas por el usuario para aplicaciones de control. Estos son llamados memoria de aplicación.

El total de memoria, especificada para un controlador, podría incluir memoria del sistema y de aplicación. Por ejemplo, un controlador con un máximo de 64K podría tener rutinas de ejecución que usan 32K y un área de trabajo, reciclaje, de 1/4K. Este arreglo deja un total de 31 3/4K para la de aplicación.

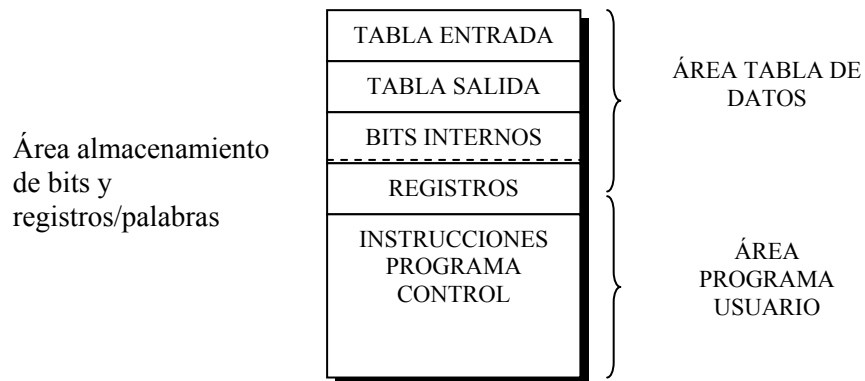


Aunque este no es siempre el caso, el máximo de memoria especificada para un controlador, normalmente, incluye sólo la memoria de aplicación disponible. Otros controladores podrían especificar sólo la memoria de usuario disponible para el programa de control, asumiendo un área de tabla de datos fija definida por el fabricante.

### 1.4.1.2 Memoria de aplicación

La memoria de aplicación almacena instrucciones programadas y datos que el procesador usará cuando ejecute sus funciones de control. En la figura 12 se ilustra el mapa de los elementos de esta área. Cada controlador tiene un máximo de área de memoria de aplicación, la cual varía dependiendo del tamaño del controlador. El controlador almacena todos los datos en la tabla de datos de la memoria de aplicación, mientras que ésta almacena instrucciones programadas en el área de programa de usuario.

**Figura 12. Mapa de la memoria de aplicación**



#### 1.4.1.2.1 Área de tabla de datos

Esta sección de la memoria de aplicación está compuesta de varias áreas, como se observa en la figura 12. Estas son:

- tabla datos de entrada;
- tabla datos de salida;
- área de almacenamiento.

Estas áreas contienen información en forma binaria representando el estado de E/S, números, y códigos.

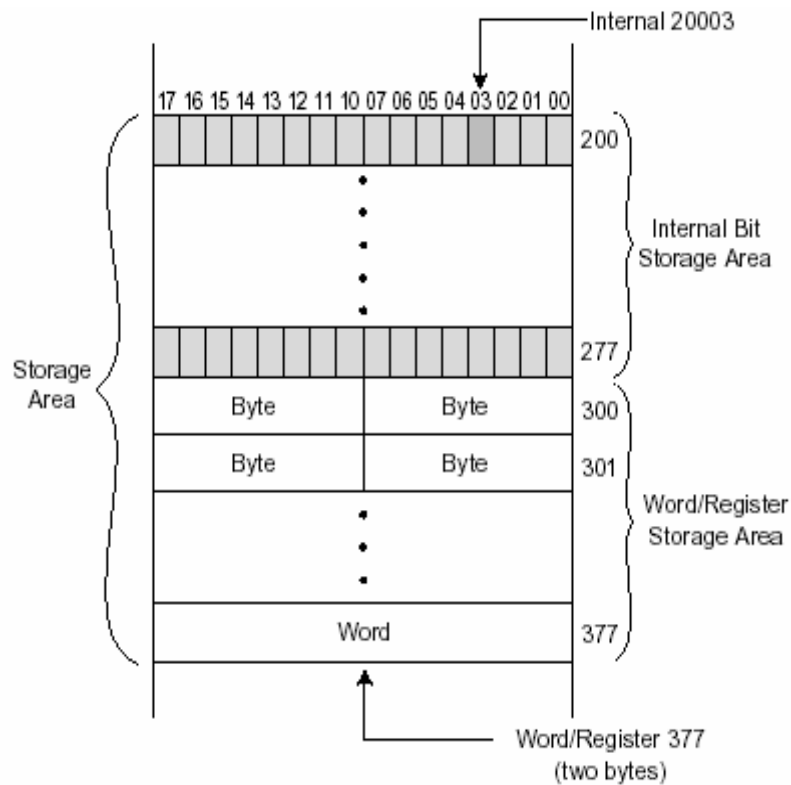
- **Tabla de entrada:** la tabla de entrada es un arreglo de bits que almacena el estado de las entradas digitales conectadas a la interfase de entrada del PLC. El número máximo de bits de entrada de la tabla de entrada es igual al máximo número de campos de entrada que pueden ser conectadas al PLC. De este modo, cada entrada conectada tiene un bit análogo en la tabla de entrada, correspondiente a la terminal a la cual está conectada la entrada.

Durante la operación del PLC, el procesador leerá el estado de cada entrada en el módulo de entrada y colocará el valor en la dirección correspondiente en la tabla de entrada de datos. Esta tabla es, constantemente, cambiada para reflejar los cambios del módulo de entrada. Estos cambios de la tabla de entrada se hacen durante la parte de lectura de la actualización de E/S.

- **Tabla de salida:** la tabla de salida es un arreglo de bits que controlan el estado de los dispositivos de salida, conectados a las salidas digitales. Como en la tabla de datos de entrada, cada salida conectada tiene un bit análogo, el cual se representa en el programa de control durante la exploración del programa, cambiando el estado de las salidas durante la exploración de actualización de salidas.

- Área de almacenamiento:** el propósito de esta área de la tabla de datos es almacenar datos cambiantes. El área de almacenamiento consiste de dos partes: área interna de almacenamiento de bits y área de almacenamiento de registros / palabras, como se aprecia en la figura 13. El área interna de almacenamiento de bits contiene bits que corresponden a salidas internas, bobinas internas, relés internos. Las salidas internas no controlan, directamente, los dispositivos de salida porque están almacenadas en direcciones que no se encuentran en el mapa de la tabla de salida de datos, y, por ende, cualquier dispositivo de salida.

**Figura 13. Área de almacenamiento de la tabla de datos**



Fuente: L.A. Bryan, “Controladores lógicos programables”, 2da edición, pág. 124

#### **1.4.1.2.2 Área de programa de usuario**

La sección de programa de usuario de la memoria de aplicación es reservada para el almacenamiento de la lógica de control. Todas las instrucciones del PLC que controlan la máquina o proceso son almacenadas en esta área. El lenguaje del software de ejecución del procesador, el cual representa cada instrucción del PLC, almacena sus instrucciones en la memoria de programa de usuario.

Cuando un PLC ejecuta su programa, el procesador interpreta la información en la memoria de programa de usuario y controla los bits utilizados en la tabla de datos que corresponden a las E/S reales o internas.

### **1.4.2 Configuración de memoria – Direccionamiento de E/S**

#### **1.4.2.1 Organización de tabla de datos**

La tabla de organización de datos o configuración, como es llamada en ocasiones, es muy importante. La configuración no sólo define la dirección discreta del dispositivo, sino, también, los registros que serán usados para control numérico y análogo, además de las bases para operaciones de tiempo y conteo del PLC.

##### **1.4.2.1.1 Direccionamiento de E/S**

La estructura de E/S de un controlador es diseñado con algo en mente. Los dispositivos de campo son conectados a los módulos de E/S del PLC, los cuales están localizados en el chasis. La localización de cada dispositivo de E/S en el chasis es mapeado a la tabla de E/S, donde la ubicación del módulo E/S define la dirección de los dispositivos conectados al módulo.

## **1.5 Implementación y diseño**

### **1.5.1 Definición de tarea de control**

El usuario debe iniciar resolviendo la solución del proceso definiendo la tarea de control, la cual, es determinada por lo que necesita hacer. Esta información provee el fundamento para el programa de control. Para minimizar errores, la tarea de control debe ser definida por aquellos que están familiarizados con la operación de la máquina o proceso. La apropiada definición de la tarea de control esta directamente relacionada con el buen funcionamiento del programa de control.

La tarea de control define eventos a varios niveles. Todos los departamentos involucrados deben trabajar juntos para determinar que entradas serán requeridas.

Si la tarea de control es realizada manualmente o por lógica de relés, el usuario debe revisar los pasos del procedimiento manual para determinar mejoras, si alguna, puede o desea hacerse. Aunque la lógica de relés puede implementarse, directamente, en un PLC, el procedimiento debe ser rediseñado, cuando sea posible, para alcanzar las necesidades del proyecto y capitalizar en las características de un controlador.

### **1.5.2 Estrategia de control**

Después de haber definido la tarea de control, la planeación y solución pueden iniciarse. Este procedimiento, comúnmente, involucra a determinar una estrategia de control, la secuencia de los pasos que deben ocurrir dentro del programa para producir el efecto deseado. Esta parte del desarrollo del programa es conocida como desarrollo de un algoritmo.

La estrategia de implementación para una tarea de control sigue el desarrollo de un algoritmo. El usuario debe implementar el control, a partir de un conjunto de instrucciones dado y dar la solución en un número finito de pasos. Si desarrollar el algoritmo para resolver el problema se dificulta, el diseñador debe retornar a la definición de la tarea de control para redefinir el problema. La regla fundamental para definir la estrategia del programa es *pensar primero, luego programar*.

### **1.5.3 Guía de implementación**

Un controlador recibe comandos del programa de control, el conjunto de instrucciones o algoritmos solución creados por el programador. Por lo tanto, el desempeño del programa de control depende de que tan organizado sea el usuario. Hay muchas maneras de resolver un problema; pero si la aplicación es desarrollada de manera sistemática, la probabilidad de errores es menor.

Las técnicas usadas para implementar el programa de control varían de acuerdo al programador. Sin embargo, el programador debería seguir ciertos procedimientos. En la tabla 1 se muestra una lista de procedimientos para programación de nuevas aplicaciones y modernizaciones.

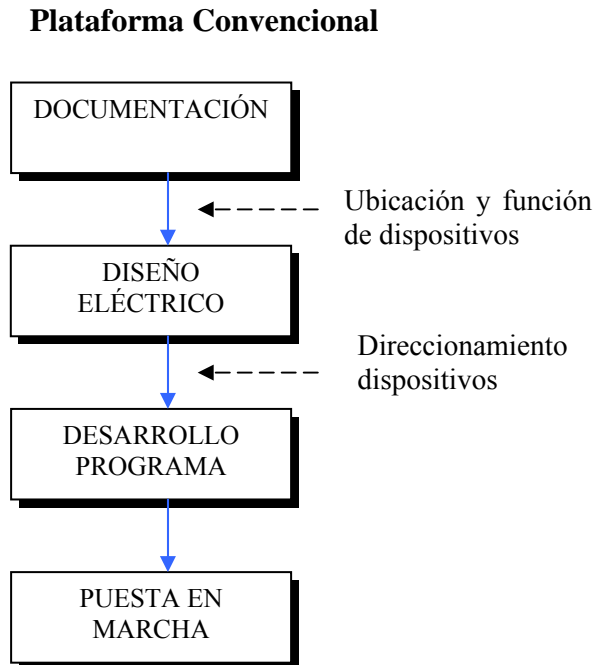
**Tabla I. Guía de desarrollo de aplicación para PLCs convencionales**

<b>Nuevas Aplicaciones</b>	<b>Modernizaciones</b>
Entender el funcionamiento deseado del sistema	Entender el actual funcionamiento del proceso o máquina
Revisar posibles métodos de control para optimizar la operación del proceso o máquina	Revisar la operación y optimizar cuando sea posible
Diagrama del flujo del proceso	Asignar direcciones E/S y direcciones internas a entradas y salidas
Implementación del diagrama de flujo usando simbología de diagramas lógicos o lógica de relés	Traducir el diagrama de lógica de escalera en código del PLC
Asignación de direcciones E/S y direcciones internas para entradas y salidas	
Traducir la lógica a implementar en código del PLC	

---

En el desarrollo de sistemas con PLCs convencionales, la documentación define dónde estarán todas las E/S y, entonces, el diseño eléctrico define el cableado entre dispositivos y PLC. Normalmente, no se puede empezar a desarrollar el programa hasta que las direcciones de las E/S sean asignadas porque se necesitan las direcciones para escribir el programa. En la figura 14 se muestran las etapas de desarrollo de una aplicación.

**Figura 14. Diagrama de flujo del desarrollo de una aplicación de automatización convencional**



#### **1.5.4 Organización del programa e implementación**

Organización es una palabra clave cuando se está programando e implementando una solución de control. En proyectos grandes, un nivel alto de organización es necesario, especialmente cuando un grupo de personas esta involucrado.

##### **1.5.4.1 Creación del diagrama de flujo y secuencia de salidas**

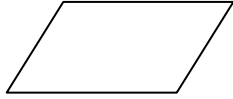
El diagrama de flujo es una técnica, frecuentemente, cuando se planifica un programa después de una descripción escrita. Un diagrama de flujo es una representación pictográfica que graba, analiza, y comunica información, además de describir el proceso en una manera secuencial. Cada paso en un diagrama representa una operación, si una E/S, decisión o dato de proceso es cumplida.



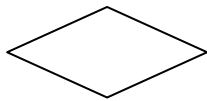
**Figura 15. Símbolos de diagrama de flujo**



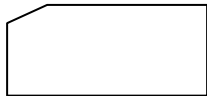
**Proceso:** un grupo de una o más instrucciones que realizan una función;



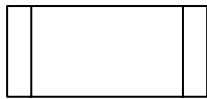
**E/S:** cualquier función involucrada con un dispositivo de E/S;



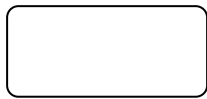
**Decisión:** un punto en el programa donde una trayectoria alterna es posible;



**Preparación:** un grupo de una o más instrucciones que determinan el estado de un subsiguiente proceso;



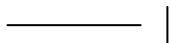
**Proceso predefinido:** un grupo de operaciones no detalladas en el diagrama de flujo;



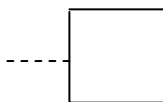
**Terminal:** inicio, fin, o punto de interrupción en un programa;



**Conector:** entrada a, salida a, otra parte del diagrama de flujo;



**Línea de flujo:** dirección de un proceso o flujo de datos;



**Anotación:** describe comentarios o explicaciones para aclaraciones.

### **1.5.4.2 Configuración del controlador**

La configuración del PLC debe considerarse durante el desarrollo del diagrama de flujo y la lógica de secuencia. La configuración define que módulos de E/S serán usados, además donde estarán localizados en el chasis local y chasis remoto. La ubicación de los módulos determina las direcciones de E/S que serán usadas en el programa de control.

Durante la configuración, el usuario debe considerar lo siguiente: posibles expansiones futuras; módulos especiales de E/S; y la ubicación de los módulos en el chasis, todos los módulos AC de E/S juntos, y todos los de DC y analógicos juntos, etc. Al considerar estos detalles, a lo largo de la documentación del sistema, resultará en un buen diseño del sistema.

### **1.5.4.3 Direccionamiento de E/S**

La asignación de entradas y salidas es uno de los más importantes procedimientos durante la etapa de organización de programación e implementación. La asignación de E/S indica cuáles son las entradas del PLC que serán conectadas a los dispositivos de entrada, y, cuáles son las salidas que manejarán los dispositivos de salida. La asignación de registros internos, incluye temporizadores, contadores, MCRs, etc, y también se realizan en este paso.

En la tabla II se muestra la asignación de direcciones de E/S, mientras que en la tabla III se muestra la asignación de direcciones para registros internos. Estas asignaciones pueden ser extraídas de la lógica de compuertas o escalera de símbolos que fueron usadas para describir la lógica de secuencia.

**Tabla II. Asignación de direcciones de E/S**

Tipo Módulo	Direcciones E/S			Descripción
	Chasis	Grupo	Terminal	
Entrada	0	0	0	LS1 – Posición
	0	0	1	LS2 – Detector
	0	0	2	Interruptor
	0	0	3	PB1 –
Salida	0	0	4	SOL1
	0	0	5	PL1
	0	0	6	PL2
	0	0	7	Motor M1
Salida	0	0	0	SOL2
	0	1	1	PL3

**Tabla III. Asignación de direcciones para registros internos**

Dispositivo	Dirección interna	Descripción
CR7	1010	CR7 reemplazo
TDR10	T200	ON – temporizador de
CR10	1011	CR10 reemplazo
CR14	1012	CR14 reemplazo
-	1013	Enclavamiento

Los números usados para direcciones de E/S dependen del PLC utilizado. Estas direcciones pueden representarse en octal, decimal o hexadecimal. La sección de descripción de la tabla especifica los dispositivos a su correspondiente dirección.

Durante la asignación de E/S, el usuario debe agrupar entradas y salidas asociadas. Esta agrupación permitirá monitoreo y manipulación de un grupo de E/S simultáneamente.

#### 1.5.4.4 Asignación en la tabla de registros

La asignación de direcciones a los registros usados en el programa de control es otro aspecto importante de la organización del PLC. El camino más fácil para asignar registros es una lista de todos los registros disponibles del PLC. Entonces, conforme son usados, se describe el contenido de cada registro. En la tabla IV se muestra una tabla de asignación de registros para los primeros 15 registros en un sistema PLC, desde la dirección 2000<sub>8</sub> hasta la dirección 2016<sub>8</sub>.

**Tabla IV. Asignación tabla de registros**

<b>Registro</b>	<b>Contenido</b>	<b>Descripción</b>
2000	Entrada analógica	Temperatura entrada
2001	Entrada analógica	Temperatura entrada
2002	Disponible	-
2003	Disponible	-
2004	Entrada TSW	Ajuste entrada de TSW
2005	Entrada TSW	Ajuste volumen (V1)
2006	Constante 2350	Constante temporizador
2007	Acumulado	Valor acumulado contador
2010	Disponible	-
2011	Disponible	-
2012	Constante 1000	Inicio de tabla
2013	Constante 1010	
2014	Constante 1023	
2015	Constante 1089	

#### **1.5.4.5 Documentación del programa**

Mientras la apropiada programación del PLC es importante para un buen desempeño en tiempo real de la aplicación, todo el trabajo es perdido sin la adecuada documentación del sistema. Sin la documentación, las actividades del sistema, tales como cambios, instalación y mantenimiento son difíciles de realizar. La documentación es una colección ordenada de información acerca de la operación de la máquina o proceso y los componentes de hardware y software del sistema de control. La documentación es una referencia muy valiosa durante el diseño, instalación, arranque, depuración, y mantenimiento del sistema. Al diseñador del sistema, documentación debe ser una herramienta de trabajo utilizada durante la fase de diseño. Si los componentes de la documentación son creados y mantenidos durante el diseño del sistema, estos darán los siguientes beneficios.

- Proveerán una fácil y precisa comunicación a todos los componentes involucrados con el sistema.
- Servirán como una referencia al diseñador durante y después de la fase de diseño.
- Ayudarán al diseñador, o alguien más, responder preguntas, diagnóstico de problemas, y modificar el programa si se requieren cambios.
- Servirán como material de entrenamiento para el operador quién será la interfase con el sistema y para el personal de mantenimiento.
- Permitirá reproducir o alterar el sistema para otros propósitos.

La apropiada documentación del sistema proviene de la recopilación de hardware, además de software, información. El equipo de diseño eléctrico del sistema, usualmente, provee esta información al usuario final. En la tabla V está el listado de los componentes de la documentación.

**Tabla V. Componentes de una buena documentación**

<b>Componentes de documentación</b>
Resumen del sistema
Configuración del sistema
Diagrama de conexión de E/S
Asignación de direcciones de E/S
Asignación de direcciones registros internos
Tabla de asignación de registros
Declaración de variables
Programa de control impreso
Almacenamiento del programa de control

#### **1.5.4.5.1 Resumen del sistema**

Un buen diseño del sistema inicia con una completa comprensión del problema y una buena descripción del proceso a ser controlado. Esto es seguido por un esquema aproximado de lo que se manejará para implementar el sistema de control. Una vez finalizado el sistema, el personal involucrado en el diseño debe de proveer una descripción global, o resumen, el esquema y procedimiento usado para controlar el proceso. Un resumen debe proveer lo siguiente:

- una clara declaración del problema de control o tarea;
- descripción de la estrategia de diseño o filosofía utilizada para implementar la solución del problema, la cual define las funciones de los principales componentes de hardware y software del sistema, además de porque fueron seleccionados;
- declaración de los objetivos a ser alcanzados.

#### **1.5.4.5.2 Configuración del sistema**

La configuración del sistema es un diagrama del arreglo del sistema. En realidad, es un diagrama pictórico de los elementos de hardware definidos en el resumen del sistema. Esta muestra la ubicación, conexiones y detalle mínimos de los principales componentes del hardware, CPU, subsistemas, periféricos, interfaces gráficas, etc. La configuración del sistema no sólo indica la ubicación física de los subsistemas, sino también la asignación de las direcciones de las de E/S en los chasis.

Si el sistema del controlador involucra una red de comunicación con otros componentes, la configuración del sistema debe mostrar un diagrama de bloques general de la red completa (todos los nodos) y los principales componentes conectados a ella.

#### **1.5.4.5.3 Diagrama de conexión de E/S**

Un diagrama de conexión de E/S muestra las conexiones de los dispositivos de campo de entrada salida a los módulos del PLC. Este diagrama incluye normalmente fuentes de alimentación y subsistemas conectados a la CPU.

#### **1.5.4.5.4 Asignación de direcciones de E/S**

Un documento de asignación de direcciones de E/S identifica cada dispositivo de campo por dirección, chasis, grupo, y terminal, el tipo de módulo de entrada o salida (115VAC, 24VDC) el tipo de dispositivo de campo, interruptor de limite, solenoide, y la función que el dispositivo ejecuta en el campo. La tabla VI muestra un documento de asignación de direcciones de E/S. Este documento es similar a la tabla de asignación de E/S que será completada antes de desarrollar el programa de control.

**Tabla VI. Asignación de direcciones de E/S**

Dirección	Tipo de E/S	Dispositivo	Función
0120	115VAC entrada	PB	Pulsador de arranque
0121	115VAC entrada	LS	Limite arriba
0122	115VAC entrada	PS (NC)	Presión OK
0123	115VAC entrada	PB (NC)	Reset PB 2
•	•	•	•
•	•	•	•
0230	24VDC salida	Solenoide	Desvío
0231	24VDC salida	PL	En posición
0232	24VDC salida	PL	Arrancado

#### **1.5.4.5.5 Asignación de direcciones de registros internos**

Esta es una importante parte del paquete de documentación. Porque los registros internos son usados para programación de temporizadores, contadores, programadores tienden a usar estos libremente, sin considerar su uso. Sin embargo, al igual que con E/S reales, el mal uso de registros internos puede resultar en mal operación del sistema.

La buena documentación de registros internos simplifica modificaciones durante la puesta en marcha. Por ejemplo, imagine que en una puesta en marcha se necesita modificar una o más líneas de programación para añadir un enclavamiento extra. Para esta modificación, el usuario debe utilizar bobinas internas que no han sido asignadas. Si el documento de las direcciones de los registros internos esta actualizada y precisa, mostrando direcciones usadas y no usadas, el usuario puede, rápidamente, localizar direcciones de registros internos disponibles. Esto ahorra tiempo y evita confusión. El documento de asignación de direcciones de registros internos indica la dirección, tipo y función de cada registro en el programa. La tabla VII ilustra un documento de este tipo.



**Tabla VII. Asignación de direcciones de registros internos**

<b>Registro</b>	<b>Tipo</b>	<b>Descripción</b>
1000	Bobina	Usada para enclavamiento
1001	Bobina	Maneja contacto contador
1002	Comparación	Usado para comparación
•	•	•
T100	Temporizador	Temporizador retardo

#### **1.5.4.5.6 Tabla de asignación de registros**

La mayoría de aplicaciones utilizan registros para almacenar o mantener información de temporizadores, contadores o comparaciones. Guardar una copia precisa del uso y cambios de estos registros es muy crítico. Al igual que con un documento de asignación de E/S, la tabla de asignación de registros muestra si o no un registro es usado.

#### **1.5.4.5.7 Declaración de variables**

En un entorno de programación IEC 1131-3, la documentación de las direcciones físicas de E/S, direcciones de registros internos, asignación de registros requiere que los dispositivos conectados al PLC vía E/S sean declarados o definidos, como variables. Una apropiada declaración de variables, incluye el nombre de la entrada, salida o registro interno, debe incluirse en cada documento.

#### **1.5.4.5.8 Programa de control impreso**

En este documento se muestra cada instrucción programada con las direcciones asociadas a cada entrada y salida. La impresión, sin embargo, no da una información real acerca de la función de cada instrucción o cual dispositivo es evaluado o controlado. Por esta razón, el código del programa de control sólo, sin la documentación mencionada, no es adecuado para interpretación del sistema de control. La mayoría de fabricantes provee un paquete de documentación que permite la programación del dispositivo, generalmente una PC, computadora personal, para ingresar etiquetas o nomenclatura de mnemónicos para los elementos del programa de control.

La mayoría de software IEC 1131 incluye un paquete de documentación de las variables definidas usadas así como las etiquetas de los elementos programados. Estos sistemas también incluyen un resumen de la declaración y tipos de variables declaradas.

#### **1.5.4.5.9 Almacenamiento del programa de control**

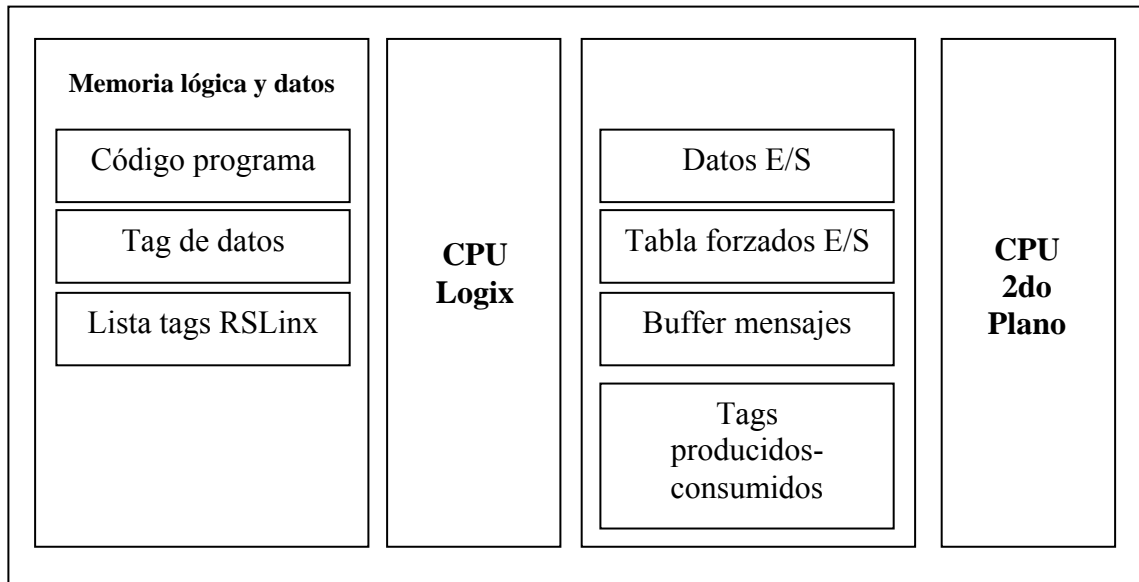
En la mayor parte de situaciones, la programación del PLC ocurre en un lugar diferente de donde el controlador, finalmente, será instalado. Por esta razón, el usuario debe de guardar el programa de control en un dispositivo de almacenamiento. Esta práctica permite al usuario enviar o transportar el programa almacenado al lugar de instalación y descargar este en la memoria del controlador, rápidamente. Esto se hace, usualmente cuando el sistema utiliza una memoria volátil, pero esto es, también, con memorias no volátiles con el propósito de tener una copia de seguridad.

## 2 PLATAFORMA CONTROL LOGIX

### 2.1 Arquitectura controlador Control Logix

La arquitectura de un controlador Control Logix se muestra en figura 16.

**Figura 16. Arquitectura controlador Control Logix**



Las características del controlador son las siguientes:

- la CPU Logix ejecuta el código de aplicación y mensajes;
- la CPU de 2do plano se comunica con E/S y envía y recibe datos del 2do plano;
- la CPU de 2do plano opera independientemente de la CPU Logix, de esta manera envía y recibe información de E/S.

## 2.2 Organización de tags

### 2.2.1 Estructura de tags

Tag: un tag es un nombre asignado a un área de la memoria del controlador donde son almacenados datos. Los tags son el mecanismo básico para asignar memoria, referenciar datos, y monitorear datos. Algunas características de los tags son:

- el tamaño mínimo para un tag es de 4 bytes;
- cuando se crea un tag que almacena datos que requieren menos de 4 bytes, el controlador utiliza 4 bytes, pero el dato sólo utiliza los bytes que necesita.

El controlador utiliza el nombre del tag internamente y no necesita referencia – cruzada para una dirección física.

- En PLCs convencionales, una dirección física identifica cada ítem de dato.
- En PLCs convencionales, direccionamiento sigue un formato numérico fijo que depende del tipo de dato, tal como N7:8, F8:3.
- Símbolos son requeridos para hacer lógica fácil de entender.
- En controladores Control Logix, no hay formato numérico fijo. El propio nombre del tag identifica el dato. Esto nos permite.
  - Organizar datos que reflejan la máquina o proceso.
  - Documentar, por medio de nombres de tags, aplicaciones de usuario en paralelo con el desarrollo de la aplicación.

Cuando se crea un tag, se asignan las siguientes propiedades: tipo de tag, tipo de dato y alcance.

### 2.2.2 Tipo de tag

El tipo de tag define como opera dentro del proyecto. En la tabla VIII, se muestra los tipos de datos de acuerdo a la función.

**Tabla VIII. Tipos de datos de tags**

<b>Función</b>	<b>Tipo</b>
Almacenar un valor o valores para uso en la lógica del programa	Base
Representar un tag	Seudónimo
Enviar datos a otro controlador	Tag producido
Recibir datos de otro controlador	Tag consumido

### 2.2.3 Tipo de dato

Define el tipo de dato que el tag almacena, como bit, entero, valores de punto flotante, cadenas de caracteres.

Una *estructura* es un tipo de dato resultado de la combinación de otros tipos de datos, las características son:

- una estructura es construida para crear un único tipo de dato para emparejar una necesidad específica;
- dentro de una estructura, cada tipo de dato individual es llamado miembro.
  - Como los tags, los miembros tienen un nombre y tipo de dato.
  - Un controlador Control Logix contiene un juego de estructuras predefinidas, tipos de datos, para uso con instrucciones específicas tales como temporizadores, contadores, bloques de funciones, etc.

- Se pueden crear estructuras de usuario.

En la tabla IX se muestran algunos ejemplos de estructuras y la cantidad de bits que estas utilizan de los 4 bytes posibles (32bits).

**Tabla IX. Bits utilizados en diversas estructuras definidas**

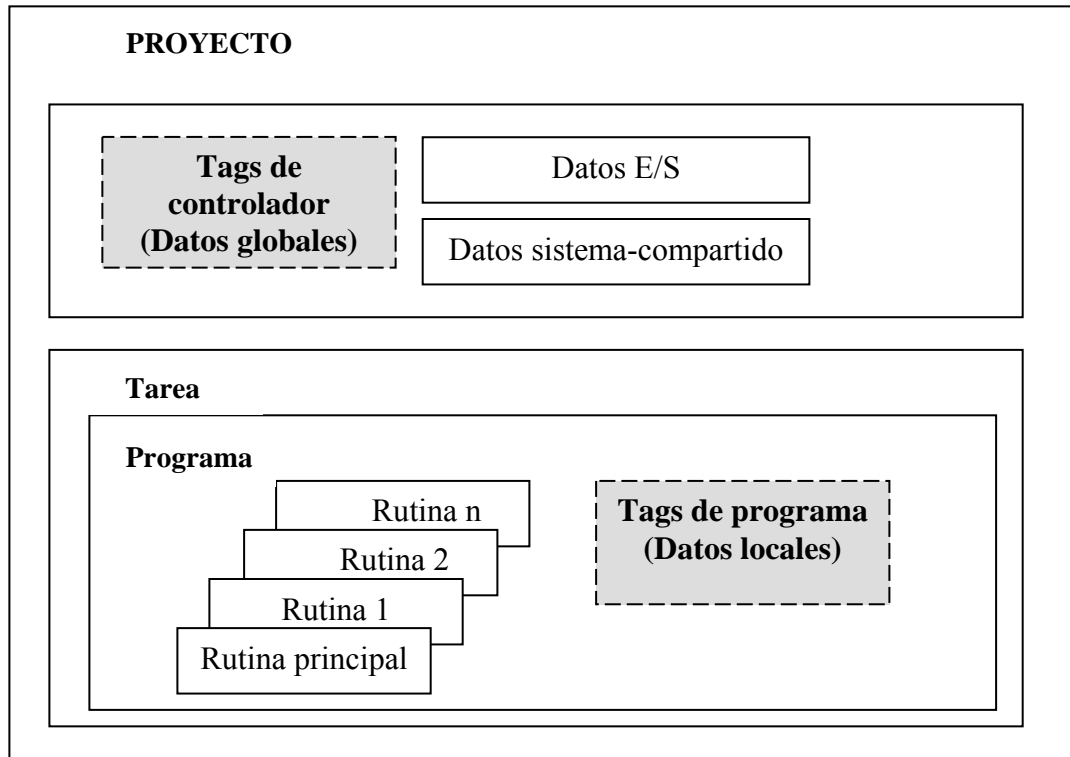
Tipo de dato	Bits					
	31	16	15	8	7	1 0
Bool-Booleano	No utilizado					0 o 1
Sint-Entero sin signo	No utilizado				-128 a +127	
Int - Entero	No utilizado			-32,7678 a +32767		
Dint - Doble entero	-2,147,483,648 a +2,147,483,647					
Real - Real	-3.40282347E <sup>38</sup> a -1.17549435E <sup>-38</sup> (valores negativos)					
	0					
	1.17549435E <sup>-38</sup> a 3.40282347E <sup>38</sup> (valores positivos)					

#### 2.2.4 Alcance

Un controlador Control Logix permite dividir la aplicación en múltiples programas, cada uno con sus propios datos. No hay conflictos entre tags con el mismo nombre en diferentes programas.

Cuando se crea un tag, se puede definir, ya sea como tag controlador, dato global, o tag de programa para un programa específico, dato local, como se observa en la figura 17.

**Figura 17. Estructura de un proyecto y alcance de los tags**



## **2.3 Características de la plataforma Control Logix**

### **2.3.1 Programación basada en direccionamiento con tags**

En PLCs convencionales se utiliza direccionamiento físico para cada elemento. Cada dirección, usualmente, depende del tipo de elemento y el esquema de direccionamiento del PLC. El software de programación almacena una tabla mostrando la relación entre las direcciones físicas y el nombre del elemento al cual se refiere.

Si el ingeniero no tiene una copia de usuario del programa con la documentación cuando se conecta al PLC, las direcciones son sin sentido, por lo que necesita conocer las direcciones físicas de los datos. Esto ocurre en situaciones de mantenimiento de la máquina o proceso.

Con Control Logix, la CPU entiende nombres reales, como “velocidad transmisión” y “válvula impresión”. No es necesario mantener una referencia cruzada entre los nombres de los tags y las direcciones físicas. Sólo hay una dirección: el nombre del tag, así los programas son más fáciles de leer, incluso si no se tiene documentación. Esto da como resultado las ventajas que se muestran en la tabla X.

**Tabla X. Ventajas de la plataforma Control Logix sobre la convencional en el direccionamiento de datos**

<b>Ventajas:</b>
• Reducción del tiempo de ingeniería y costos.
• Menos código de programación.
• Fácil mantenimiento, especialmente si no se tiene el programa.
• Fácil de entender la documentación.
• Fácil de entender la estructura de los datos del PLC cuando se lee de otros PLCs o paquetes de software.

### **2.3.2 Estructuras de datos**

En los PLCs convencionales todos los datos deben de encajar en la estructura definida por el PLC. Por ejemplo, datos enteros son colocados en el archivo de enteros, datos de punto flotante en archivos de punto flotantes, etc.

En la plataforma Control Logix, estructuras de memorias pueden ser desarrolladas para simular el proceso en la rotativa de impresión. Primero, una estructura es creada y contiene todos los parámetros de la máquina o proceso. Entonces, se usa para almacenar los parámetros, por ejemplo, para cada torre de la rotativa de impresión hay una estructura de usuario. Las ventajas obtenidas se ilustran en la tabla XI.



**Tabla XI. Ventajas de la plataforma Control Logix sobre la convencional al usar estructuras definidas por el usuario**

<b>Ventajas:</b>
• Reducción del tiempo de ingeniería de programación al utilizar estructuras.
• Programas fáciles de entender y leer durante el desarrollo y mantenimiento.
• No es necesario manejar complejas estructuras de memoria.

### **2.3.3 Diseño e implementación**

En el desarrollo de sistemas de control convencionales, un diagrama del sistema define donde todos los dispositivos de E/S serán ubicados y, entonces, el diseño eléctrico define donde los elementos serán cableados. Normalmente, no se puede iniciar a desarrollar el programa hasta que las direcciones de las E/S sean asignadas, porque son necesarias para escribir el código del programa.

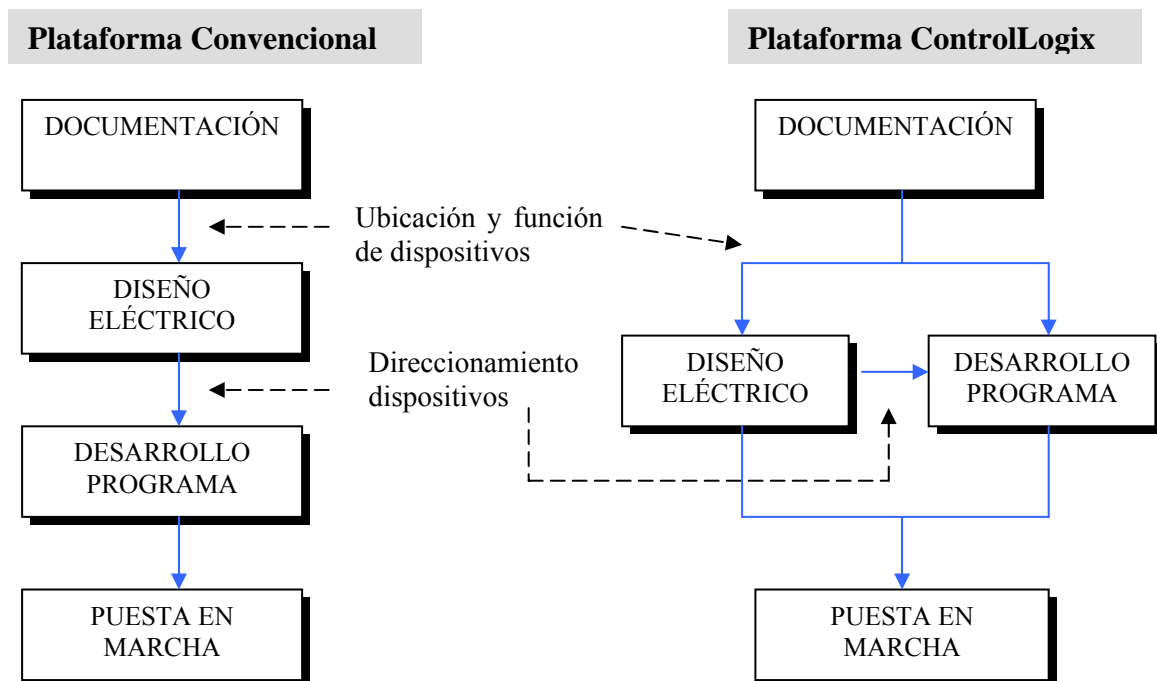
Con los controladores Control Logix, el código del programa puede ser escrito en paralelo con el diseño eléctrico, utilizando los tags. Después de escrito el código, los tags pueden ser mapeados a los puntos de E/S reales, si necesidad de re-escribir el código. Esta ingeniería concurrente ayuda a reducir el tiempo que toma en el diseño e implementación del sistema, porque los elementos que son hechos en serie en la plataforma convencional, en Control Logix se hacen, simultáneamente.

En la tabla XII se observan las ventajas de la plataforma Control Logix sobre la convencional en el diseño e implementación de proyectos, mientras que en la figura 18 se ilustra un diagrama esquemático que muestra la diferencia del proceso de diseño e implementación.

**Tabla XII. Ventajas de la plataforma Control Logix sobre la convencional en el diseño e implementación de proyectos**

<b>Ventajas:</b>
• Permite el desarrollo del código del programa sin el diseño eléctrico completo.
• Reducción del tiempo de ingeniería de diseño y puesta en marcha.
• Se puede crear una librería de programas reutilizables que pueden ser usados en múltiples proyectos.
• Tecnología de programación de seudónimo de tags que permite separar los nombres de tags de las direcciones físicas de E/S.

**Figura 18. Diferencia entre la plataforma Control Logix y la convencional en el diseño e implementación de un proyecto de automatización**



#### **2.3.4 Modularidad de código de programa**

Para nuestra aplicación de la rotativa de impresión Goss, se tienen dos torres para la transmisión. Consideremos que el proyecto se está desarrollando con la plataforma convencional y ya se tiene creado las tablas de datos y el código de programación para una de las torres y se necesita crear los datos para la otra torre.

En la plataforma convencional el programa del PLC consiste de un programa principal y las subrutinas. Todos los datos del programa están almacenados en una tabla de datos. Con esta estructura de datos los pasos para crear la tabla de datos y el código de programación para la segunda torre es como sigue:

1. se copia el programa de la primera torre para crear el de la segunda torre y se asignan las direcciones de E/S;
2. se revisa el código de programación para asegurarse que las direcciones no están repetidas o erróneas.

En la plataforma Control Logix se permite el uso de múltiples programas, cada uno con sus propios datos. No habrá conflictos en la tabla de direcciones de datos entre los dos diferentes programas pues no requiere la reasignación de direcciones internas usadas en cada programa o revisión de conflicto de direcciones entre programas. Los pasos para el diseño de la segunda torre es:

1. copiar el programa de la primera torre a la segunda torre;
2. asignar direcciones de E/S.

En la tabla XIII se muestran las ventajas al programar el código en la plataforma Control Logix sobre la plataforma convencional.

**Tabla XIII. Ventajas de la plataforma Control Logix sobre la convencional en la modularidad de programación**

<b>Ventajas:</b>
• Estructura de programación que permite crear código modular.
• Cada programa tiene su propia tabla de datos.
• No hay conflictos entre direcciones de programas diferentes.
• Direcciones de E/S pueden ser asignadas en una tabla.
• Fácil de reutilizar librerías de código de programa.

## **2.4 Componentes de un proyecto de software**

El sistema operativo de controlador es un sistema preventivo conforme a la norma IEC 61131-3. Los componentes se listan enseguida.

### **2.4.1 Tareas**

Se pueden configurar tareas continuas, periódicas, o eventos. Un controlador Control Logix permite usar múltiples tareas para programar y priorizar la ejecución de los programas basados en un criterio específico. Esto da un balance entre el tiempo de procesamiento del controlador y las diferentes operaciones de la aplicación. Algunas de las características de las tareas son:

- el controlador ejecuta sólo una tarea a la vez;
- una tarea diferente puede interrumpir una tarea que se está ejecutando y tomar el control;
- en una tarea dada, sólo un programa se ejecuta a la vez.

### **2.4.1.1 Tareas continuas**

Las tareas continuas se ejecutan en segundo plano todo el tiempo. Cualquier tiempo de CPU no asignado para otras operaciones, aplicaciones de movimiento, comunicaciones, y tareas periódicas y de eventos, es utilizado para ejecutar los programas dentro la tarea continua. Cuando la tarea se ejecuta, se reinicia inmediatamente, además un proyecto no requiere una tarea continua, si se usa, puede existir sólo una tarea continua.

### **2.4.1.2 Tareas periódicas**

Una tarea periódica ejecuta una función a un periodo específico. Cuando el tiempo de la tarea periódica expira ocurre lo siguiente.

- Interrumpe cualquier tarea de prioridad inferior.
- Se ejecuta una vez.
- Retorna al control donde la tarea previa se interrumpió.

### **2.4.1.3 Tareas de eventos**

Una tarea de eventos ejecuta una función sólo cuando un evento específico ocurre. Cuando el evento ocurre, la tarea realiza lo siguiente.

- Interrumpe cualquier tarea de prioridad inferior.
- Se ejecuta una vez.
- Retorna al control donde la tarea previa se interrumpió.

El evento puede ser:

- cambio en una entrada digital;
- nuevo muestreo de un dato analógico;
- ciertas operaciones de movimiento;
- tags consumidos.
- Instrucciones de eventos

### **2.4.2 Programas**

Una tarea puede tener hasta 32 programas, cada uno con sus propias rutinas y tags locales. Una vez una tarea es activada, todos los programas asignados a la tarea se ejecutan en el orden que están listados en el organizador del controlador. Los programas son útiles para desarrollar proyectos por múltiples programadores. Durante el desarrollo, el código en un programa que hace uso de los tags locales puede ser duplicado en un segundo programa y minimizar la posibilidad de colisión o confusión con los nombres de tags.

### **2.4.3 Rutinas**

Las rutinas contienen el código ejecutable. Cada programa tiene una rutina principal, la cual es la primera en ejecutarse en la ejecución del programa. Utiliza lógica, como saltos a subrutinas (JSR), para llamar otras rutinas. Se puede especificar, opcionalmente, una rutina de falla. Se deben utilizar las consideraciones que se observan en la tabla XIV para determinar cuando usar una tarea, programa o rutina.

**Tabla XIV. Consideraciones de uso de tareas, programas y rutinas**

Comparación	Tarea	Programa	Rutina
Cantidad:	Varía de acuerdo al controlador (4, 8 o 32).	32 programas por tarea.	Ilimitado número de rutinas por programa.
Función:	Determina como y cuando será ejecutado el código.	Organiza grupos de rutinas necesarias para compartir un área de datos común.	Contiene código ejecutable que controla la máquina o proceso.
Uso:	<ul style="list-style-type: none"> <li>• La mayoría de código debe estar en una tarea continua.</li> <li>• Se puede usar una tarea periódica para procesos lentos o tiempos de operación críticos.</li> <li>• Se usan tareas de eventos para eventos que requieren sincronización.</li> </ul>	<ul style="list-style-type: none"> <li>• Colocar la mayoría de piezas o celdas de planta en programas aislados.</li> <li>• Usar programas para aislar diferentes programadores .</li> <li>• Ejecución de orden configurable en una tarea..</li> </ul>	<ul style="list-style-type: none"> <li>• Aislar máquinas.</li> <li>• Usar el lenguaje apropiado para procesos.</li> <li>• Modularizar código en subrutinas que pueden llamarse múltiples veces.</li> </ul>
Consideración	<ul style="list-style-type: none"> <li>• Un alto número de tareas puede ser difícil de depurar.</li> <li>• Podría necesitarse deshabilitar las salidas de procesos en algunas tareas para mejorar el desempeño.</li> <li>• Tareas se pueden inhibir para prevenir ejecución.</li> </ul>	<ul style="list-style-type: none"> <li>• Datos de múltiples programas deben ir en áreas de datos globales.</li> <li>• Los programas se listan en el organizador en el orden de ejecución.</li> </ul>	<ul style="list-style-type: none"> <li>• Subrutinas con múltiples llamadas pueden dificultar la depuración.</li> <li>• Datos pueden ser referenciados en área de datos locales y globales.</li> <li>• La llamada de un gran número de rutinas puede afectar el tiempo de exploración.</li> </ul>

#### 2.4.4 Rutina principal

Cuando un programa se ejecuta, su rutina principal se ejecuta. La rutina principal se utiliza para llamar, ejecutar, otras rutinas, subrutinas.

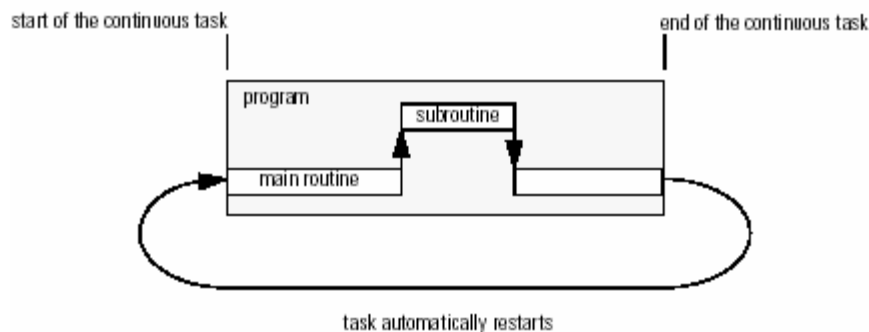
#### 2.4.5 Subrutina

Cualquier otra rutina distinta de la rutina principal o de falla es una subrutina. Para ejecutar una subrutina, se utiliza un salto en otra rutina, así como en la rutina principal.

### 2.5 Configuración de la ejecución de una tarea

Cuando se crea un nuevo proyecto, el software de programación crea automáticamente una tarea inicial que es configurada para que se ejecute siempre, tarea continua. Cuando la tarea ejecuta un ciclo de exploración completo, que se reinicia inmediatamente al finalizar. En la figura 19 se ilustra la ejecución de una tarea continua.

**Figura 19. Ejecución de una tarea continua**



Fuente: “Logix 5000 Controllers Common Procedures”, **Programming Manual**, Pub. 1756-PM001G-EN-P, 2004, cap. 1, pág. 18.

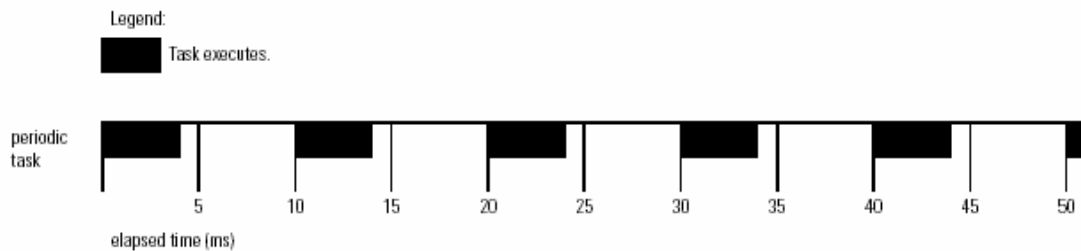


Si se está familiarizado con aplicaciones DCS o se planea un sistema usando un diagrama de bloques funcionales, se puede configurar una tarea para que se ejecute cada cierto tiempo, tarea periódica. Esto permite actualizar el diagrama de bloques cada cierto intervalo de tiempo.

- Siempre que el intervalo de tiempo para la tarea expira, la tarea se ejecuta una vez más.
- Se puede configurar el periodo desde 1ms hasta 2000s.
- Si se utiliza una tarea periódica además de una continua, la tarea periódica interrumpe la ejecución de la tarea continua. Cuando la tarea periódica es ejecutada, el control retorna a la tarea continua.

En la figura 20 se ilustra la ejecución de una tarea periódica que se repite cada 10ms

**Figura 20. Ejecución de una tarea periódica**

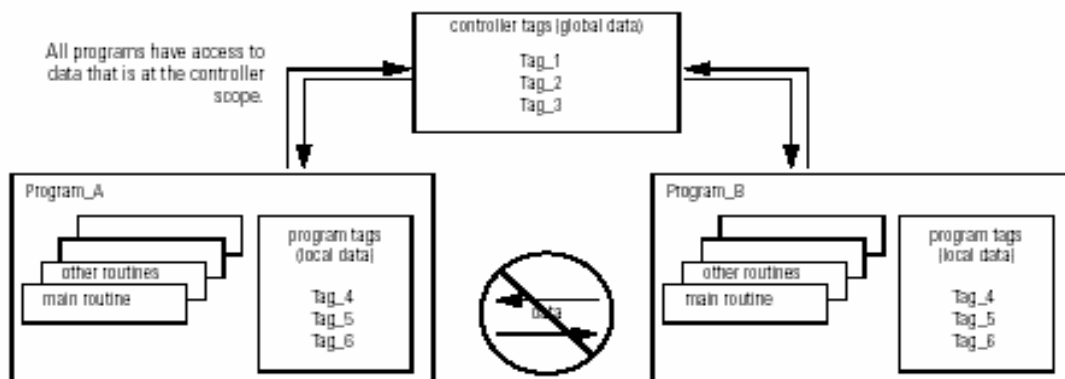


Fuente: “Logix 5000 Controllers Common Procedures”, **Programming Manual**, Pub. 1756-PM001G-EN-P, 2004, cap. 1,pág. 18.

## 2.6 Ejecución de múltiples programas

Un controlador Control Logix permite dividir las aplicaciones en múltiples programas, cada uno con sus propios datos, no hay conflictos entre nombres de tags de distintos programas. Esto hace fácil el uso repetido de código de programa y nombres de tags en múltiples programas. En la figura 21 se observa como están aislados dos programas.

**Figura 21. Organización de dos programas sin conflictos de ejecución**



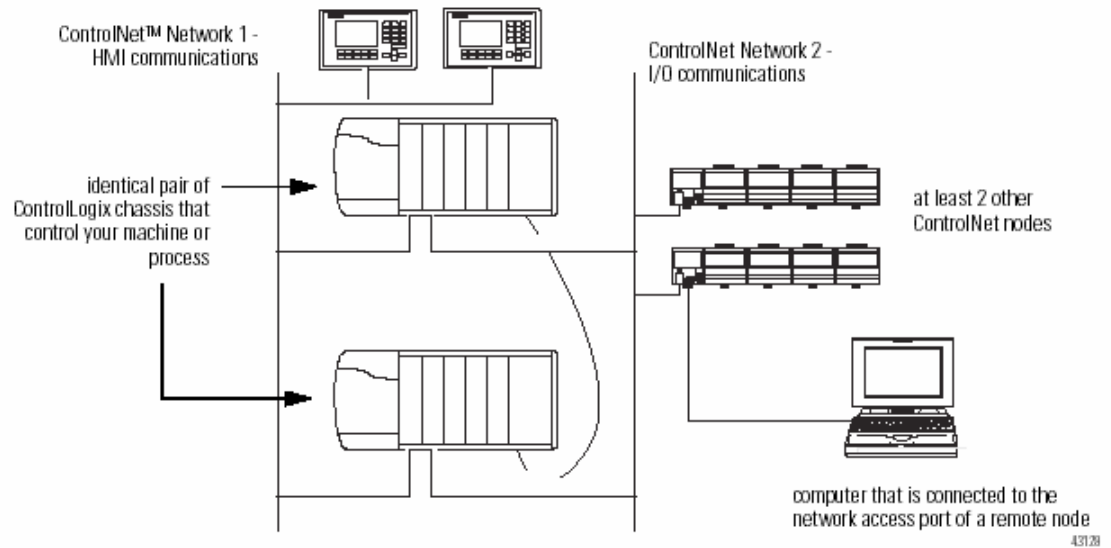
Fuente: "Logix 5000 Controllers Common Procedures", **Programming Manual**, Pub. 1756-PM001G-EN-P, 2004, cap. 1, pág. 20.

No se pueden acceder a datos locales de rutinas que no pertenezcan al propio programa. Se debe tomar en cuenta que cada tarea debe de tener al menos una tarea.

## 2.7 Redundancia de equipo

Un sistema de redundancia Control Logix usa un par idéntico al chasis original Control Logix para mantener la máquina o proceso trabajando si algún problema ocurre con un controlador. La figura 22 muestra el esquema de un sistema redundante.

**Figura 22. Sistema redundante Control Logix**



Fuente: “Control Logix Redundancy System”, **User Manual**, Pub. 1756-UM523D-EN-P, 2003, cap. 1, pág. 1.

Un sistema redundante no requiere programación adicional y es transparente a cualquier dispositivo conectado a la red de comunicación. Se utiliza un módulo especial para mantener la comunicación entre el par de chasis redundantes. En un sistema redundante el chasis primario contiene el controlador primario, el cual controla la máquina o proceso.

El controlador secundario está en espera de tomar control de la máquina o proceso. Un controlador secundario siempre reside en un chasis diferente del controlador primario.

### **2.7.1 Falla de corte**

Cuando una falla ocurre en cualquiera de los elementos del chasis primario el control cambia al controlador secundario. Una falla de corte ocurre por cualquiera de las siguientes razones:

- cualquiera de las siguientes situaciones en el chasis primario:
  - pérdida de alimentación;
  - falla mayor del controlador;
  - quitar, inserción, o falla de cualquier módulo en el chasis primario;
  - interrupción o desconexión del cable Ethernet;
- comando del controlador primario;
- comando del software RSLinx.

### **2.7.2 Sincronización**

Para tomar el control completo, el controlador secundario requiere el mismo proyecto del controlador primario. Esto, también, requiere la actualización de los valores de los tags. Durante la sincronización el módulo especial revisa que los módulos pareja en el chasis redundante sean compatibles con el chasis original. Este módulo, también, provee la trayectoria de transferencia del contenido del controlador primario al controlador secundario. La sincronización ocurre cuando se enciende el chasis secundario

Cuando no hay emparejamiento entre el chasis primario y el secundario, el chasis secundario falla en la sincronización con el controlador primario. Cuando esto ocurre, el controlador secundario es incapaz de tomar el control de la máquina o proceso.

En un par de chasis redundantes, el primer chasis que se enciende es el primario. Cuando se enciende el chasis secundario se sincroniza con el chasis primario.

## **2.8 Comunicación con E/S**

Para que exista comunicación entre los módulos de E/S y el controlador se deben de configurar ciertas opciones primero. Las opciones de configuración varían entre módulos, pero hay ciertas opciones comunes que se pueden configurar:

- intervalo de solicitud de paquete RPI;
- formato de comunicación;
- clave electrónica.

### **2.8.1 Intervalo de solicitud de paquete RPI**

Una conexión entre un enlace de comunicación entre dos dispositivos, como por ejemplo, entre un controlador y un módulo E/S, interfaz humano máquina u otro controlador. Las conexiones son asignaciones de medios que proveen más fiabilidad de comunicación entre dispositivos que mensajes desconectados. El RPI especifica el período de tiempo, al cual se actualizan los datos en una conexión. El número de conexiones que un sólo controlador puede tener es limitado. Existen los siguientes tipos de comunicación utilizando conexiones:

- módulos de E/S;
- tags consumidos y producidos;
- ciertas instrucciones de mensajes.

Los valores se actualizan asincrónicamente a la ejecución de la lógica. Al intervalo de tiempo especificado, el controlador actualiza un valor independientemente de la lógica de ejecución.

## **2.8.2 Formato de comunicación**

El formato de comunicación determina la estructura de datos para los tags que están asociados con el módulo. Muchos módulos de E/S soportan diferentes formatos. Cada formato utiliza una estructura de datos diferente. El formato de comunicación determina:

- conexión directa o de chasis optimizado;
- pertenencia.

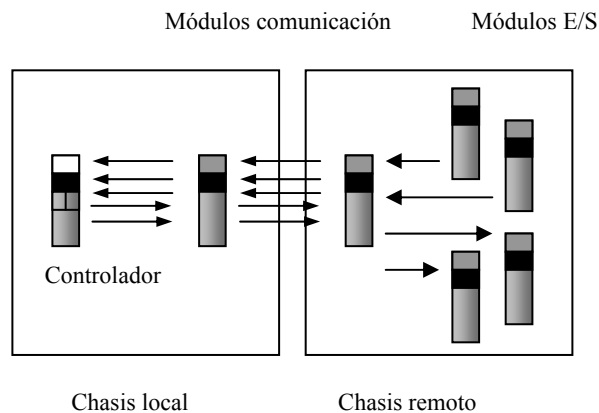
Los controladores Control Logix utilizan conexiones para transmitir datos de E/S, las cuales pueden ser directas o de chasis optimizado.

### **2.8.2.1 Conexión directa**

Es una conexión en tiempo real de transferencia de datos entre el controlador y un módulo de E/S. El controlador mantiene y monitorea la conexión con el módulo de E/S. Cualquier pérdida de comunicación, como falla de módulo o remoción de un módulo mientras está energizado, activa los bits de falla en el área de datos asociados con el módulo. Cada módulo transfiere sus datos de / hacia el controlador individualmente, cada módulo puede determinar su propia razón (RPI). Se pueden enviar más datos por módulo, como datos de diagnóstico y analógicos. En la figura 23 se observa la transferencia de tags en una conexión directa. Algunas de las consideraciones en este tipo de conexión son:

- requiere conexiones adicionales y recursos de red;
- es un método soportado únicamente en el chasis local;
- datos de E/S son presentados como tags individuales.

**Figura 23. Transferencia de tags en una conexión directa**

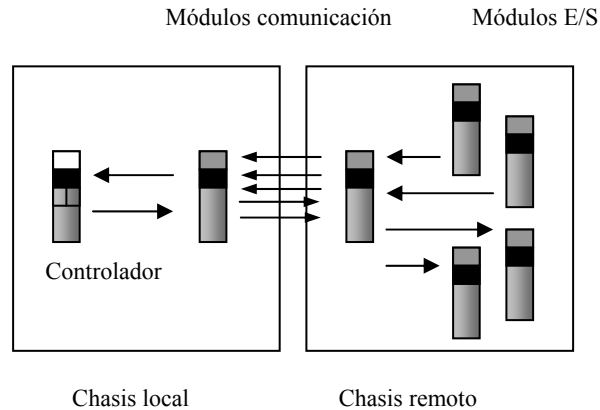


### 2.8.2.2 Conexión de chasis optimizado

Para módulos digitales de E/S se puede utilizar este tipo de comunicación. Este tipo de conexión consolida comunicación entre el controlador y todos los módulos digitales en el chasis. Una sola conexión puede utilizarse para un chasis completo de módulos digitales, por lo que reduce recursos de red y recargamiento de la misma. En la figura 24 se muestra una transferencia de datos en una conexión de chasis optimizado. Algunas de las consideraciones en este tipo de conexión son:

- todos los módulos envían datos a la misma tasa;
- las ranuras no usadas son comunicadas;
- se necesita una conexión directa para datos analógicos y de diagnóstico;
- limitado a usarse en chasis remoto;
- datos E/S son presentados como arreglos con tags seudónimo para cada módulo.

**Figura 24. Transferencia de tags en una conexión de chasis optimizado**



### 2.8.2.3 Pertenencia

En la plataforma Control Logix los módulos de E/S transmiten información de comunicación a través de la red de comunicación. Esto significa que múltiples dispositivos pueden recibir los mismos datos al mismo tiempo de un sólo dispositivo. Al seleccionar el formato de comunicación, se debe seleccionar si se va a establecer una relación propietario o sólo escuchar el módulo.

- **Controlador propietario:** es el controlador que crea la configuración primaria y conexión de comunicación al módulo. El controlador propietario escribe los datos de configuración y puede establecer una conexión con el módulo.
- **Conexión sólo – lectura:** este es un tipo de conexión de E/S donde otro controlador provee la configuración de datos para el módulo E/S. Un controlador que usa este tipo de conexión sólo monitorea el módulo. El controlador no puede escribir datos de configuración y sólo puede mantener una conexión con el módulo de E/S cuando el controlador propietario está controlando el módulo de E/S.



## 2.9 Modelo productor consumidor

El modelo productor - consumidor viene a cubrir las deficiencias del antiguo modelo fuente - destino, ya que, los mensajes se identifican por su contenido y ofrece las siguientes ventajas:

- múltiples nodos pueden consumir los mismos datos simultáneamente desde un sólo productor;
- los nodos se pueden sincronizar fácilmente para obtener un rendimiento más preciso del sistema;
- los dispositivos se pueden comunicar autónomamente, sin la necesidad de un maestro de sistema.

Este modelo permite los dos métodos de acceso descritos en el modelo fuente – destino: maestro – esclavo y compañero – compañero, pero además ofrece otras importantes capacidades.

- Multi – lanzamiento, *multicast*: esta propiedad se refiere a poder transmitir un mismo mensaje a varios usuarios al mismo tiempo.
- Cambio de estado COS, *Change Of State*: los dispositivos producen datos solamente cuando existen cambios en la información o cuando se generan eventos, lo que reduce significativamente el tráfico en la red.
- Cíclico, *Cyclic*: los dispositivos producen datos basados en un tiempo base cíclico, el cual es muy útil para datos que son muy cambiantes, señales analógicas, ya que, preserva que ellos dominen el ancho de banda, si se usara el modo “cambio de estado”; además, es usado para funciones periódicas donde la precisión es importante. (Galindo Morán, 2002.26).

En los controladores Control Logix se permite producir, transmitir, y consumir, recibir, tags para compartir en el sistema. En la tabla XV se muestra las opciones para transferencias de datos entre controladores en la plataforma Control Logix:

**Tabla XV. Opciones de transferencia de datos en la plataforma Control Logix**

<b>Transmisión de datos</b>	<b>Utilizar</b>
Se necesita entregar o recibir datos a un intervalo de tiempo específico	Tag producido o consumido
Enviar datos cuando una condición ocurre en la aplicación	Ejecutar una instrucción de mensaje MSG

---

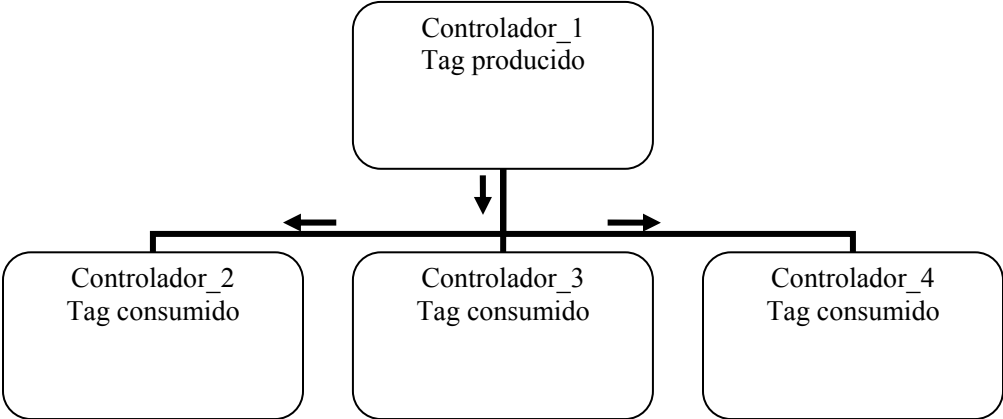
### **2.9.1 Tags producidos**

Un tag producido es aquel que está disponible para uso de otros controladores en el sistema. Múltiples controladores pueden, simultáneamente, consumir, recibir, los datos. Un tag producido envía datos a uno o más tags consumidores sin uso de lógica. El tag producido envía sus datos a una tasa igual al intervalo de solicitud de datos RPI de los datos consumidos.

### **2.9.2 Tags consumidos**

Un tag consumido es aquel que recibe datos de un tag producido. El tipo de dato del tag consumido deber ser igual al tipo de dato, incluyendo cualquier arreglo de cualquier dimensión, del tag producido. El RPI del tag consumido determina el período al cual se actualizan los datos. En la figura 25 se muestra un diagrama de cómo se distribuye la información en un sistema que utiliza tags producidos y consumidos.

**Figura 25. Compartimiento de tags en un sistema Control Logix**





## **3 CONSIDERACIÓN DE LOS PARÁMETROS DE DISEÑO**

### **3.1 Criterios de uso de tareas, programas y rutinas**

#### **3.1.1 Prioridad de tareas**

Cada tarea en el controlador tiene un nivel de prioridad que determina cual tarea se ejecuta primero cuando múltiples tareas son activadas. Un nivel alto de prioridad (1) interrumpe cualquier tarea de prioridad inferior (15). La tarea continua tiene el nivel más bajo de prioridad y es siempre interrumpida por una tarea periódica o de evento. Los controladores Control Logix soportan hasta 32 tareas de usuario y 15 niveles de prioridad.

#### **3.1.2 Factores que afectan la ejecución de tareas**

##### **3.1.2.1 Sistema de encabezado de información**

El sistema de encabezado es el tiempo que el controlador utiliza en mensajes de comunicación y tareas de segundo plano.

- Un mensaje de comunicación es cualquier comunicación que no se configura a través del fólder de configuración de comunicación de E/S del proyecto, tal como instrucciones de mensajería.
- Un mensaje de comunicación ocurre sólo cuando una tarea periódica o de evento no se está ejecutando. Si se utilizan múltiples tareas, se debe de asegurar que el tiempo de exploración e intervalos de ejecución dejan suficiente tiempo para mensajes de comunicación.

- El sistema de encabezado de información interrumpe sólo la tarea continua.
- La porción del tiempo del sistema de encabezado de información especifica el porcentaje de tiempo, excluyendo el tiempo para tareas periódicas y de eventos, que el controlador dedica a mensajes de comunicación.

### **3.1.2.2 Procesamiento de salidas**

Al final de una tarea, el controlador ejecuta operaciones de encabezado, procesamiento de salidas, para los módulos de salida del sistema. Esto podría afectar la actualización de los módulos de E/S del sistema. Se pueden desactivar el procesamiento de salidas para una tarea específica, el cual reduce el tiempo transcurrido de la tarea.

### **3.1.2.3 Número de tareas**

Si se tienen muchas tareas, puede dar lugar a lo siguiente.

- La tarea continua podría tomar demasiado tiempo para completarse.
- Tareas podrían traslaparse. Si una tarea es interrumpida muy frecuentemente, podría no completarse la ejecución antes de que sea activada nuevamente.
- La comunicación del controlador podría ser lenta.
- Si la aplicación es diseñada para colección de datos, se debe evitar el uso de múltiples tareas. El cambio entre múltiples tareas limita el ancho de banda de comunicación.

### **3.1.3 Configuración de una tarea continua**

La tarea continua es creada, automáticamente, cuando se abre un nuevo proyecto en el software de programación RSLogix 5000. La ejecución de una tarea continua es similar a la de un procesador convencional. Un controlador Control Logix soporta una tarea continua, pero no es requerida dentro del proyecto. Se puede configurar si la tarea actualiza los módulos de salida al final de la tarea continua. La tarea continua se puede cambiar a una periódica o de evento si así se requiere.

La CPU deja una porción de tiempo entre la tarea continua y el sistema de encabezado de información. Cada transición entre tareas de usuario y sistema de encabezamiento de información toma tiempo adicional a la CPU para cargar y restaurar información de tareas.

### **3.1.4 Configuración de una tarea periódica**

Una tarea periódica se ejecuta automáticamente basada en un intervalo de tiempo preconfigurado. Esta tarea es similar a una interrupción de tiempo ajustable en un procesador convencional. Se puede configurar si la tarea actualiza los módulos de salida al final de la ejecución de la tarea. Después que la tarea se ejecuta, no se ejecuta, nuevamente, hasta el intervalo de tiempo configurado ha expirado.

Si la aplicación tiene mucho tráfico de comunicación, tal como instrucciones de mensajes o comunicaciones con el software RSLinx, se debe utilizar una tarea periódica en lugar de una tarea continua. Esto evita la información de encabezado asociada con cambio de tareas, lo cual mejora el desempeño de la aplicación.

### **3.1.5 Configuración de una tarea de eventos**

Una tarea de eventos se ejecuta automáticamente basada en un evento preconfigurado. Se puede configurar si la tarea actualiza los módulos de salida al final de la ejecución de la tarea. Después que la tarea se ejecuta, no se ejecuta nuevamente hasta que el evento configurado ocurre otra vez. Cada tarea de eventos requiere un evento específico que define cuando la tarea se ejecuta. En la activación por cambio de estado de un dato de entrada COS: *change of state*, el módulo de entrada, digital o analógico, activa la tarea de eventos basado en el cambio de estado de la configuración del módulo. Si se habilita el COS para múltiples puntos, un traslape de tareas de eventos puede ocurrir.

## **3.2 Desarrollo de aplicaciones en rutinas**

Cada rutina contiene lógica en un lenguaje de programación que el usuario selecciona, el lenguaje se escoge basado en la aplicación.

### **3.2.1 Lógica en escalera**

- Ejecución paralela o continua de múltiples operaciones, no secuencias.
- Operaciones booleanas o basadas en bits.
- Operaciones lógicas complejas.
- Mensajes y procesamiento de comunicación de datos.
- Enclavamiento de máquinas.
- Operaciones para personal de servicio y mantenimiento podrían tener interpretación en orden para resolver fallas de máquinas o procesos.
- Aplicaciones de servosistemas.



### **3.2.2 Diagramas de bloques de funciones**

- Procesos continuos y control de variadores de velocidad.
- Lazos de control.
- Cálculos en circuitos de flujo.

### **3.2.3 Diagrama de secuencia de funciones**

- Manejo de alto nivel de múltiples operaciones.
- Operaciones de secuencias repetitivas.
- Procesos de batch.
- Secuencias de control de movimiento, vía diagrama de funciones de secuencia con estructura de datos integrado.
- Operaciones de estado de máquina.

### **3.2.4 Texto estructurado**

- Complejas operaciones matemáticas.
- Arreglos especializados o tablas de procesos de lazos.
- Manejo de datos alfanuméricos ASCII o protocolo de procesamiento.

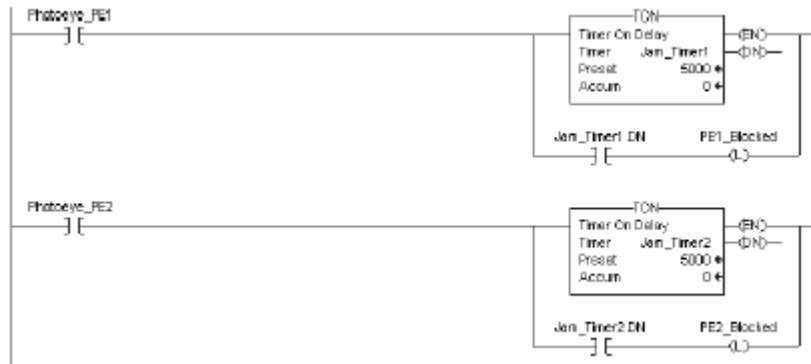
## **3.3 Métodos de programación**

### **3.3.1 Programación inline**

En este tipo de programación se escribe múltiples códigos con diferentes referencias de tags. Este tipo de programación tiene las siguientes características:

- usa más memoria;
- el más rápido tiempo de ejecución porque todos los tags referenciados son definidos antes de la ejecución del programa;
- requiere más tiempo para crearse y modificarse.

**Figura 26. Ejemplo de programación inline**



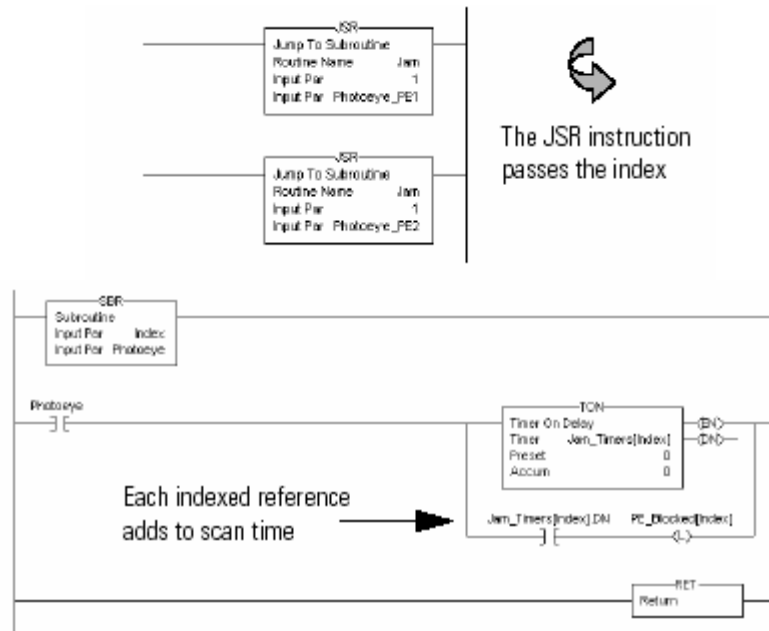
Fuente: “Logix 5000 Controllers Consideration Design”, **Reference Manual**, Pub. 1756-RM094A-EN-P, 2004, cap. 2, pág. 13.

### 3.3.2 Programación indexada

En esta programación se copia el código y se utiliza referencias indexadas a los datos almacenados en arreglos. Las características de esta programación son:

- una copia del código es rápida de desarrollar;
- tiempo de ejecución lenta porque todos los tags referenciados son calculados durante la ejecución del programa;
- puede ser difícil de mantenimiento porque los datos monitoreados no están sincronizados a la ejecución.

**Figura 27. Ejemplo de programación indexada**



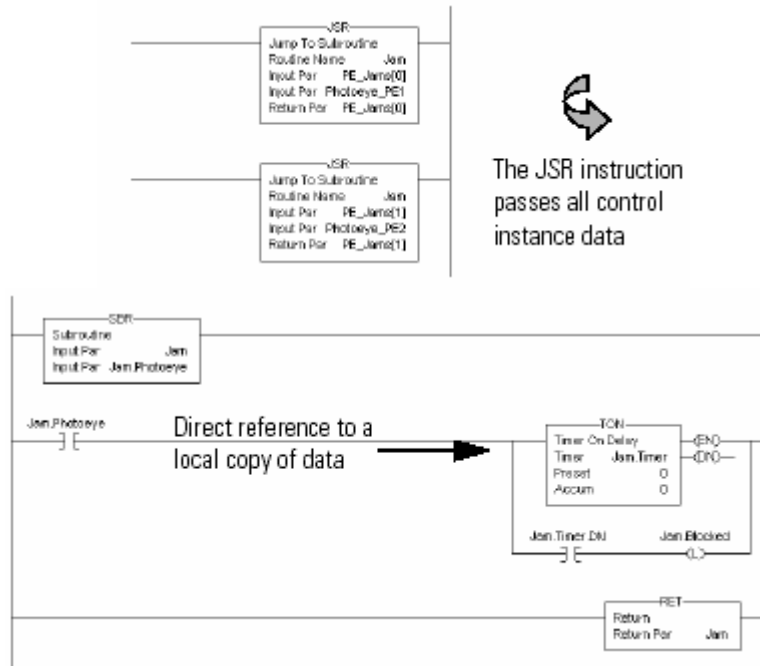
Fuente: “Logix 5000 Controllers Consideration Design”, **Reference Manual**, Pub. 1756-RM094A-EN-P, 2004, cap. 2, pág. 13.

### 3.3.3 Programación buffered

En este tipo de programación se copia los valores de un arreglo en tags y referencia estos tags buffer directamente. Las características de este tipo de programación son:

- una operación de copia puede ocurrir más rápida que desplazamientos múltiples indexados;
- elimina la necesidad de calcular compensaciones de arreglos durante la ejecución del programa;
- puede ser difícil de mantenimiento porque los datos monitoreados no están sincronizados a la ejecución.

**Figura 28. Ejemplo de programación buffered**



Fuente: “Logix 5000 Controllers Consideration Design”, **Reference Manual**, Pub. 1756-RM094A-EN-P, 2004, cap. 2, pág. 13.

### 3.4 Direccionamiento de datos

Los controladores Control Logix soportan tipos de datos atómicos IEC 61131-3, tales como: booleanos, enteros sin signo, enteros, dobles enteros, y reales. Los controladores, también, soportan tipos de datos compuestos, tales como arreglos, estructuras predefinidas, por ejemplo, contadores, temporizadores, y estructuras de datos definidas por el usuario.

### 3.4.1 Tipo de datos atómicos

**Tabla XVI. Beneficios y consideraciones de datos atómicos**

Beneficios:	Consideraciones:
<ul style="list-style-type: none"> <li>• Nombres individuales</li> <li>• Número de tags no limitado</li>   <li>• Editor de tags y monitoreo de datos puede filtrar tags individuales y despliega cualquier referencia</li> <li>• Se listan alfabéticamente en el editor de tags y monitor de datos</li>   <li>• Seudónimo para tags (el tag base y sus bits)</li> <li>• Pueden ser agregados en programación en línea</li> </ul>	<ul style="list-style-type: none"> <li>• Cada tag usa 32 bits de memoria</li> <li>• Requiere más encabezado de información, más memoria que datos compuestos</li> <li>• Se puede cambiar tipo de tag en programación offline</li>   <li>• Los miembros de estructura son listados en el orden que fueron definidos</li> </ul>

### 3.4.2 Tipo de datos compuestos

**Tabla XVII. Beneficios y consideraciones de datos compuestos**

Beneficios:	Consideraciones:
<ul style="list-style-type: none"> <li>• Permitido para nombres específicos y organización definida por el usuario</li> <li>• Consolida información en memoria del controlador</li>   <li>• Optimiza tiempo de comunicación y memoria</li> <li>• Arreglos pueden ser indexados dinámicamente</li> <li>• Se pueden crear arreglos en línea</li>   <li>• Seudónimo para estructuras definidas por el usuario</li> </ul>	<ul style="list-style-type: none"> <li>• Limite de 2Mbytes de datos para estructuras definidas por el usuario y arreglos</li> <li>• Seudónimo de tags no puede hacerse a la raíz de un tag de arreglo</li> <li>• Filtrado de editor de tags y monitor de datos limitado</li> <li>• Pueden crearse o cambiarse estructuras en programación offline</li> <li>• Arreglos sólo pueden cambiarse en programación offline</li> </ul>

La CPU Logix lee y manipula datos de 32 bits. Todos los datos inician con 32 bits de desplazamiento, así el mínimo de memoria para un tag es de 4 bytes. Cuando se crea un tag que almacena menos de 4 bytes, el controlador usa 4 bytes, pero los datos sólo usan los bits necesarios.

Para manipular datos enteros sin signo o, enteros, el controlador los convierte a dobles enteros, ejecuta el programa y, entonces, retorna el resultado a un entero sin signo o entero. Esto requiere memoria y tiempo de ejecución adicional cuando se compara al usar dobles enteros para la misma operación.

### **3.5 Comunicación en una red Ethernet**

#### **3.5.1 Protocolo TCP/IP**

El protocolo TCP/IP es un protocolo de la capa de transporte (TCP) y un protocolo de red (IP) comúnmente usado en ambientes de negocios para comunicación dentro redes y a través de interredes. El módulo Ethernet 1756-ENET/B utilizado para la comunicación entre el controlador y el Panel View, utiliza TCP/IP para mensajería “explícita”; esto es, mensajes en el cual el tiempo no es un factor crítico, tal como: descarga y carga de programas.

#### **3.5.2 Parámetros en una red Ethernet**

##### **3.5.2.1 Dirección IP**

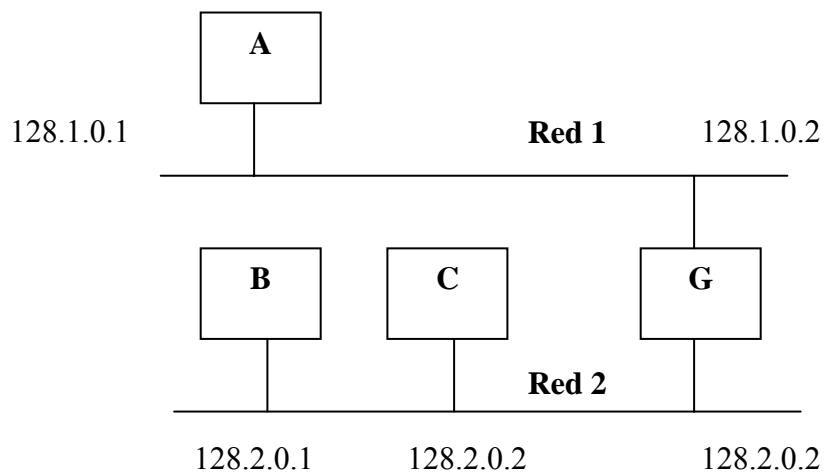
La dirección IP identifica cada nodo en la red IP, o sistema de redes conectadas. Cada nodo TCP/IP en una red, incluye módulo Ethernet, debe tener una única dirección IP.

La dirección IP es un doble palabra (32 bits) y tiene una parte de identificación de red ID y host ID. Cada red puede ser clase A, clase B, o clase C. La clase de red determina el formato de la dirección IP. Cada nodo en la misma red física debe tener una dirección IP de la misma clase e identificación de red. Cada nodo en la misma red debe tener un host ID diferente, de este modo, una única dirección IP.

### 3.5.2.2 Pasarela

Una pasarela conecta redes individuales físicas en un sistema de redes. Cuando un nodo necesita comunicarse con un nodo en otra red, una pasarela transfiere los datos entre las dos redes. En la figura 29 se ilustra como la pasarela G conecta las redes 1 y 2.

**Figura 29. Pasarela que conecta dos redes**



### 3.5.2.3 Máscara de subred

La máscara de subred permite distinguir la identificación de la red de la identificación de la estación y es usada, para determinar si la estación destino está en el segmento local o en algún otro segmento remoto de la red.

Existen máscaras por defecto, para cada clase de dirección, tal como se observa en la figura siguiente.

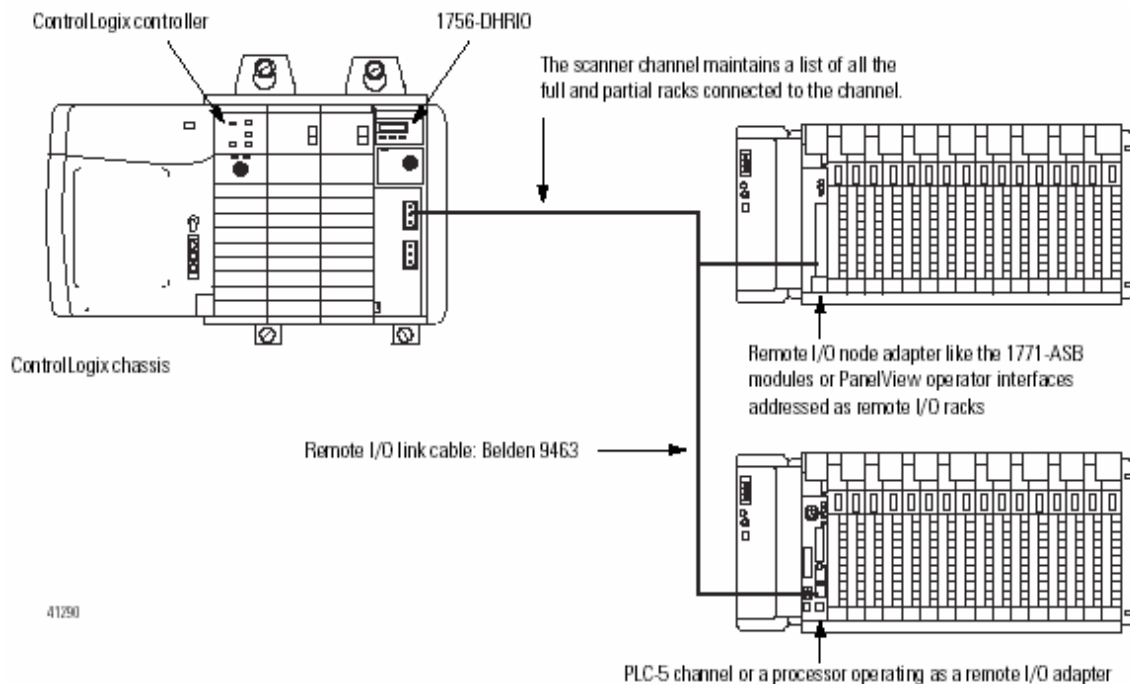
**Figura 28. Máscaras de subredes por defecto**

Clase de red	Bits usados para máscara de subred				Notación decimal
Clase A	11111111	00000000	00000000	00000000	255.0.0.0
Clase B	11111111	11111111	00000000	00000000	255.255.0.0
Clase C	11111111	11111111	11111111	00000000	255.255.255.0

La dirección IP y la máscara son operadas lógicamente ‘AND’, tanto en la fuente como en el destino, con lo que si ambos resultados son iguales, ambas estaciones son locales.

### 3.6 Comunicación en una red RIO

**Figura 30. Ejemplo de una red RIO.**



Fuente: “Control Logix Data Highway Plus-Remote I/O Communication Interface Module”, **User Manual**, Pub. 1756-UM514B-EN-P, 2003, cap. 8, pág. 2.



La red de comunicación RIO permite controlar módulos discretos o dispositivos que no están conectados en el chasis primario. En la plataforma Control Logix hay módulos de comunicación que se pueden configurar para trabajar en la red RIO, para establecer transferencias discretas y bloques de datos entre el controlador y dispositivos remotos de E/S. En la figura anterior se muestra un ejemplo de una red RIO.

### 3.6.1 Parámetros en una red RIO

En la plataforma Control Logix hay módulos de comunicación que se pueden configurar para operar sobre una red DH+ o RIO. Para los propósitos de este proyecto de automatización el módulo de comunicación se configuró para operar sobre una red RIO por cuestión de diseño.

#### 3.6.1.1 Velocidad de transmisión

Todos los dispositivos conectados a la red remota de E/S deben de comunicarse con la misma velocidad de transmisión de datos. Las velocidades disponibles son: 57.6kbps, 115.2kbps y 130.4kbps. La distancia máxima de alcance de la red depende de la velocidad de transmisión de datos.

**Tabla XVIII. Distancia máxima en una red RIO**

<b>Velocidad de transmisión</b>	<b>Distancia máxima</b>
57.6kbps	3,048.00m
115.2kbps	1,524.00m
230.4kbps	762.00m

### **3.6.1.2 Direccionamiento**

Se debe asignar direccionamiento parcial o chasis total usando el módulo de comunicación en el modo de red RIO. Ambos canales del módulo de comunicación deben tener el mismo direccionamiento, y estos se pueden comunicar con 0-37o o 40-47o nodos. Un canal tiene un máximo de 32 chasises para comunicarse y un máximo de 32 dispositivos conectados a él.

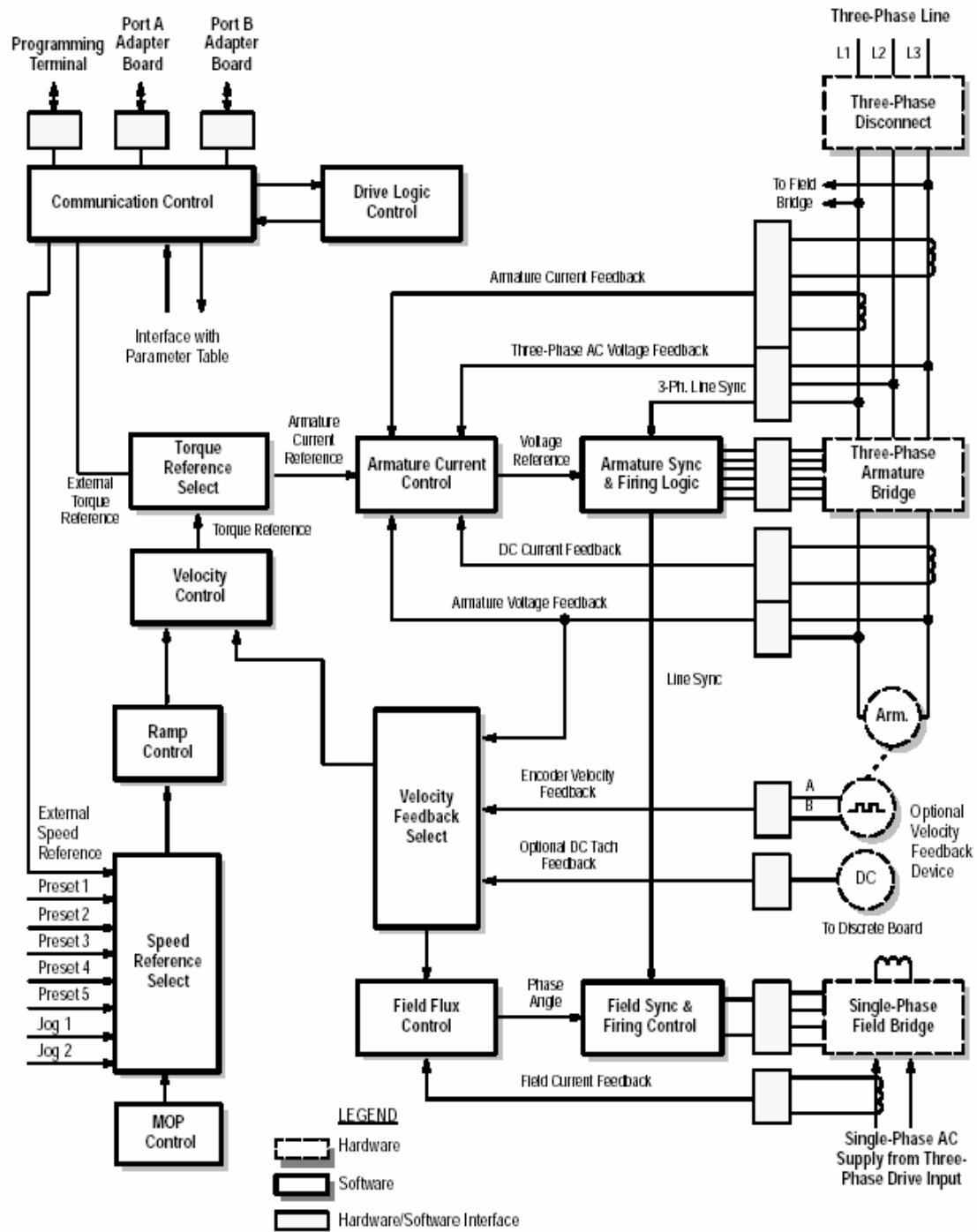
## **3.7 Variador de velocidad de C.D. 1395**

### **3.7.1 Descripción del variador de velocidad de C.D. 1395**

En la figura 31 se muestra la arquitectura interna y los bloques principales asociados con las funciones de control del variador de velocidad 1395. Todas las funciones de control del 1395 son editadas a través del uso de parámetros, los cuales pueden ser cambiados con una terminal de programación.

La información de retroalimentación es derivada de los dispositivos de hardware como se muestra en lado derecho de la figura 31. Las señales analógicas son convertidas en señales digitales para uso del drive. Las señales de control pueden provenir de dos tarjetas adaptadoras opcionales: una tarjeta adaptadora con dos puertos PORT A y PORT B o, a través de la tarjeta adaptadora de multi-comunicaciones. Toda la información de configuración y operación usada por el drive es almacenada en un sistema de tabla de parámetros. Cada parámetro incluye parámetros de configuración, fuente y sumidero de datos. La tarjeta adaptadora de multi-comunicaciones opcional permite que el drive sea controlado por medio de un controlador, a través de la red remota de E/S. La información en esta red es enviada por bloques, cada drive posee asignado un bloque de memoria y en cada bloque el controlador envía la información a cada drive.

Figura 31. Arquitectura interna del drive 1395



Fuente: "Bulletin 1395 Digital DC Drive", User Manual, Pub. 1795-5.4, 2003, 5.3.

### 3.7.2 Programación de parámetros

#### 3.7.2.1 Estructura de la tabla de parámetros

El variador de velocidad posee ciertos parámetros que son de importancia para su programación, todos los datos usados por el 1395 para controlar el motor son almacenados en la tabla de parámetros. Cada parámetro en la tabla contiene la información ilustrada en la figura 32.

Figura 32. Estructura de la tabla de parámetros

Parámetro Drive 1395
Número de Parámetro
Hex
Unidades
Nombre
Valor inicial
Min
Max
EE
Función/clasificación

- **Número de parámetro:** el número de parámetro en decimal.
- **Hex:** número de parámetro en hexadecimal.
- **Unidades:** indica las unidades desplegadas usando la terminal de programación y el valor del parámetro en unidades de ingeniería.
- **Nombre:** indica el nombre del parámetro como aparece en la terminal de programación.
- **Valor inicial:** valor del parámetro como aparecerá después que el comando Inicializar drive ha sido enviado a la terminal de programación.
- **Min:** valor mínimo permitido para el parámetro. Si no es dado el valor mínimo, el parámetro no ha sido asignado al límite mínimo.

- **Max:** valor máximo permitido para el parámetro. Si no es dado el valor máximo, el parámetro no ha sido asignado al límite máximo.
- **EE:** indica si el parámetro puede ser grabado en la EEPROM.
- **Función/clasificación:** indica la función de control al cual el parámetro esta asociado, y su clasificación.

### 3.7.2.2 Almacenamiento de la tabla de parámetros

Cuando la alimentación es aplicada a la tarjeta de control del drive, toda la tabla de parámetros es copiada de la EEPROM a la RAM. Toda la información almacenada en la RAM es pérdida cuando la alimentación es desconectada. Todos los parámetros en el drive requeridos para la calibración básica y escalamiento de las funciones de control deben ser almacenados permanentemente incluso cuando la alimentación es desconectada, para no ingresar nuevamente la información cada vez que el drive es energizado, para esto debe de guardarse los parámetros en la EEPROM del drive después de calibrado.

Cuando un parámetro es cambiado desde la terminal de programación o a través de un dispositivo externo conectado al PORT A o B, en nuestro caso a través de la tarjeta de comunicación RIO conectada al PORT A, la nueva información es almacenada en la RAM del drive. Si los datos serán almacenados en la EEPROM, se deben grabar los datos de la RAM a la EEPROM, y, esto se logra usando un comando de escritura a la EEPROM. En la tabla XIX, se muestra la tabla de parámetros de los parámetros utilizados para el control del drive.

**Tabla XIX. Parámetros utilizados para el control del drive**

<b>PARAMETRO</b>	<b>HEX</b>	<b>NOMBRE</b>	<b>UNIDADES</b>	<b>FUNCION</b>
100	64	LogicStatus		Lógica de control
110	6E	Torque Command	%	Selección torque
150	96	Logic Command		Lógica de control
606	25E	Base Motor Speed	RPM	Control velocidad
610	262	Rated Motor Volt	VOLTS	Realimentación
611	263	Motor Arm FLA	AMPS	Realimentación
625	271	Torque Mode		Selección torque
651	28B	Accel Time	SEC	Control de rampa
652	28C	Decel Time	SEC	Control de rampa

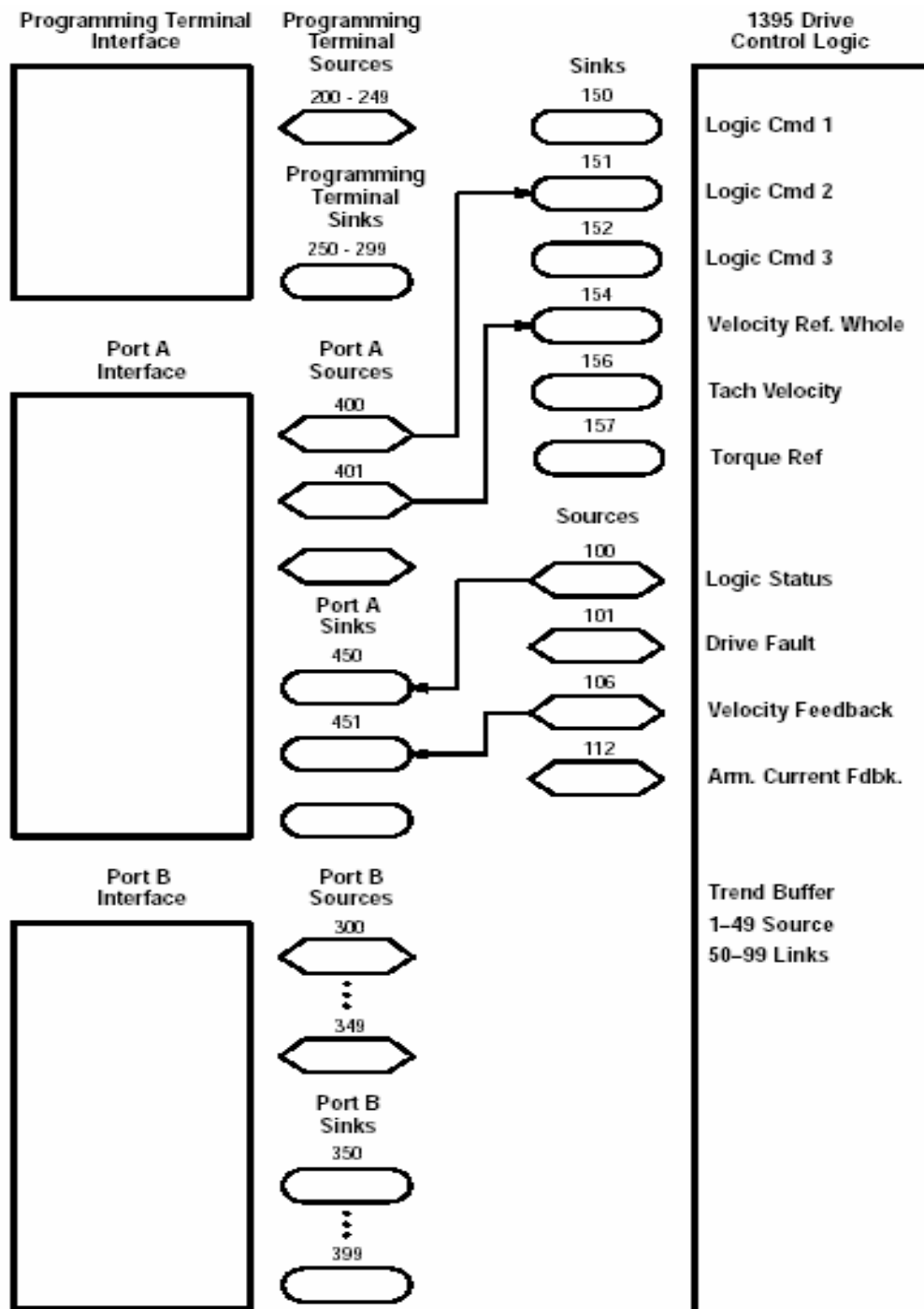
### **3.7.3 Configuración de parámetros**

La figura 31 muestra una apreciación de los parámetros asociados con la configuración del drive. El drive 1395 ha sido diseñado para aceptar entradas de control a través de tarjetas adaptadoras. Una porción del control del drive ha sido diseñado para actuar como una caja negra desde el punto de vista externo. Para ejecutar las funciones de control requeridas para aplicaciones específicas es necesario configurar varia información de control y referencia tal como comandos lógicos, referencia de velocidad y referencia de torque.

#### **3.7.3.1 Parámetros sumideros**

Varios parámetros asociados con la lógica de control han sido set aside específicamente para las tareas de recepción de información de entrada de dispositivos de control externos. Estos parámetros son llamados “sumideros”. En la figura 33 ilustra algunos de los parámetros sumideros usados para las entradas de la lógica de control. La función de cada parámetro sumidero ha sido predefina y no puede ser cambiada, cada parámetro ha sido definido para un uso específico, y, tendrá un tipo de dato específico y unidades de medida.

Figura 33. Parámetros fuente y sumidero de un drive 1395



Fuente: "Bulletin 1395 Digital DC Drive", User Manual, Pub. 1795-5.4, 2003, 5.11.

### **3.7.3.2 Parámetros fuente**

La información de entrada de los parámetros sink debe ser originaria de un parámetro fuente el cual transmite la información a través puerto Microbus. Como en la figura 31, hay 10 parámetros fuente asociados con cada puerto. Los dispositivos de hardware asociados con los parámetros fuente son determinados por la tarjeta adaptadora conectada físicamente al puerto.

La lógica de control también provee los parámetros fuente los cuales podrán ser usados para enviar información a los parámetros sumidero asociados con los puertos del Microbus. Algunos de estos parámetros fuente asociados con la lógica de control se muestran en la figura 31.

## **3.8 Interfaz humano máquina**

### **3.8.1 Plataforma dedicada**

La plataforma dedicada provee operaciones con ventanas donde se observan los procesos, usando gráficas, mensajes de textos, o ambos. Estos dispositivos son robustos y completamente equipados, hardware, software, y comunicaciones,. La operación y programación es sencilla: descarga de aplicaciones, configuración parámetros de comunicación, conexión cables de comunicación.

La plataforma dedicada esta optimizada para nivel de interfase con máquinas, proveendo una segura integración con el sistema de control. Esta reemplaza paneles tradicionales cableados y es el mecanismo de entrada/salida para la interacción del operador con una o más máquinas. La interfaz humano máquina Panel View 1500 es una pantalla para operador que es parte de la plataforma dedicada.



### 3.8.2 Panel View 1500

El Panel View 1500 es adecuado para aplicaciones que requieren monitorear, y desplegar información gráfica, permitiendo a los operadores entender rápidamente el estado del proceso o máquina. Algunas de las características del Panel View 1500 son las siguientes:

- Provee máxima flexibilidad, y fácil actualización.
- Comunicación a través de múltiples puertos.
- Integrado con el software RSView Studio para funciones avanzadas que incluyen: tendencias, expresiones, registro de datos, gráficas avanzadas.
- Manejo del panel a través de teclado, pantalla táctil, o teclado/pantalla táctil.
- Tarjeta de memoria para transferencia de archivos, datos, o actualización de sistema operativo.
- Fácil migración de aplicaciones existentes de Panel View Standard.
- Puerto de comunicación Ethernet y RS-232.
- Dos puertos USB para conexión de Mouse y teclado.

**Tabla XX. Parámetros del Panel View 1500**

<b>Especificación</b>	
Área de despliegue	228 x 304mm
Resolución	1024 x 768, gráficas de color de 18 bits
Memoria de aplicación	64MB
Teclas de función	40
Puertos de comunicación	Ethernet, RS-232
Voltaje de entrada	18...32VDC (24VDC nominal)
Potencia de consumo	70W (2.9A a 24VDC)
Software de aplicación	RSView Studio
Dimensiones	330 x 416 x 65mm



## **4 DISEÑO DE LA INGENIERIA DE CONTROL DE LA ROTATIVA DE IMPRESIÓN**

### **4.1 Antecedentes**

Gran cantidad de proyectos de automatización de máquinas y procesos complejos en Guatemala, se han integrado utilizando la plataforma convencional: familia SLC 500 y PLC5 la cual es bastante popular.

La plataforma Control Logix emergió aproximadamente hace unos años en los Estados Unidos, siendo una tecnología que ha sido introducida en países, altamente, industrializados o en vías. En Guatemala, la implementación de proyectos con esta plataforma ha sido escasa, pues, requiere de inversión en capacitación, material de entrenamiento, personal capacitado, etc. Las compañías guatemaltecas en la industria de impresión han sido automatizadas con la plataforma convencional, siendo esta la primera ha implementar con la plataforma Control Logix.

Previo a la propuesta de automatizar la rotativa de impresión Goss utilizando la plataforma Control Logix, el control de válvulas, ajustes de impresión, debían de realizarse con anticipación al funcionamiento normal de la máquina. Cada uno de los drives de CD era gobernado por un autómata independiente SLC 500, complicando la integración de los mismos. Para el monitoreo y estado de válvulas, motores, sensores de campo, velocidad de los drives, etc, no se tenía una interfaz humano máquina capaz de suministrar información apropiada del estado del sistema. Para el mantenimiento y solución de fallas del sistema resultaba complicado pues la comunicación desde una computadora personal con uno de los autómatas proveía información aislada del autómata.

## **4.2 Planteamiento del problema**

Las limitaciones del sistema existente son las siguientes.

- La operación inicial de la máquina debe de realizarse con tiempo de anticipación previo a la producción normal pues deben de hacerse ajustes manuales para la correcta operación del sistema.
- El operario no tiene ninguna interfaz humano máquina a través del cual observar el estado de los dispositivos del sistema: Válvulas, motores, sensores de campo.
- La transmisión de la rotativa de impresión posee dos drives y cada uno posee un autómeta SLC 500. El operario puede observar el estado de los drives, dispositivos críticos del sistema, a través de las calculadoras las cuales son limitadas y difícil uso.
- La integración del control de las transmisiones se ve complicada pues hay una gran cantidad de señales discretas a manejar y la programación para la comunicación entre drives se extiende.

## **4.3 Diseño de la ingeniería de control de la rotativa de impresión**

### **4.3.1 Descripción del sistema**

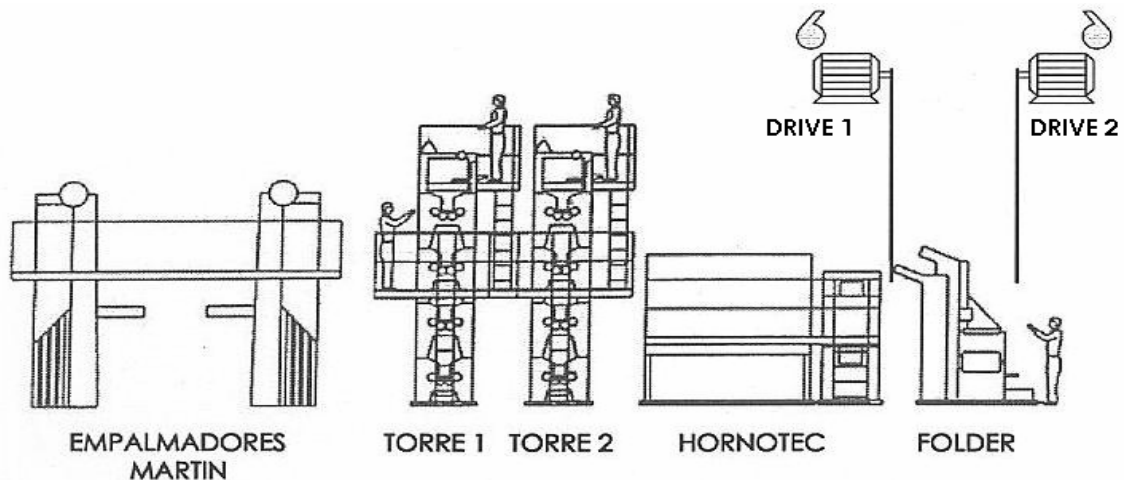
La rotativa de impresión Goss es una máquina que se utiliza para la impresión de periódicos. Esta rotativa consta de dos torres de impresión cada una con cuatro unidades que imprime un color diferente, en este caso, es una rotativa de impresión de ocho colores. Cada unidad de impresión tiene 4 válvulas que controlan el contraste, brillo y tono de cada color:

- válvula de impresión;
- válvula de micrométrico;
- válvula de tinta;
- válvula de agua.

Los colores utilizados para la impresión son los primarios: rojo, amarillo, azul y además, negro. La combinación de los colores primarios y el negro dan los restantes colores.

El eje de la rotativa es movido por medio de dos motores de C.D. cada uno de 75HP, que trabajan en paralelo para compartir la carga total del sistema a una velocidad máxima de 1900rpm. La rotativa está diseñada para trabajar a una velocidad máxima de impresión de 30,000 ejemplares por hora. En la figura 34 se ilustra un esquema de la rotativa de impresión.

**Figura 34. Rotativa de impresión Goss**



Fuente: “Rotativa de impresión GOSS”, SYTEK S.A., 2004.

Los empalmadores MARTIN son unos aparatos que se utilizan para alimentar con papel periódico la transmisión, un empalmador para cada torre. HORNOTEC es un aparato que se utiliza para impresiones de páginas satinadas, utilizado para impresión de anuncios especiales. El FOLDER se utiliza para la etapa final de impresión, se encarga de cortar y acomodar las hojas del periódico.

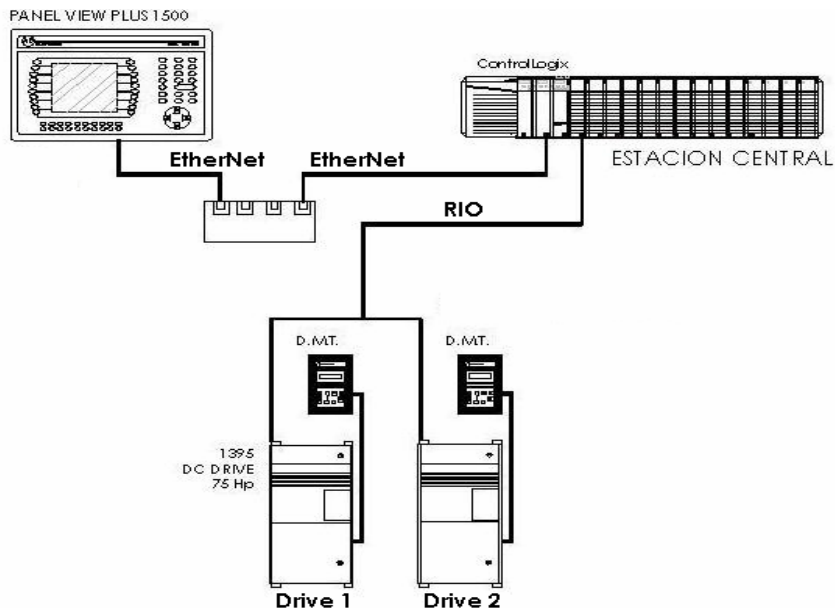
La interfaz humano máquina se comunica con el chasis del controlador a través de una red Ethernet. Los variadores de velocidad de C.D. se comunican a través de una red remota de E/S con el chasis del controlador.

### 4.3.2 Componentes de hardware

#### 4.3.2.1 Arquitectura del sistema

La arquitectura del sistema permite determinar la configuración de la red de comunicación y la ubicación de los componentes, como se ilustra en la figura 35.

Figura 35. Arquitectura red de comunicación de rotativa de impresión Goss

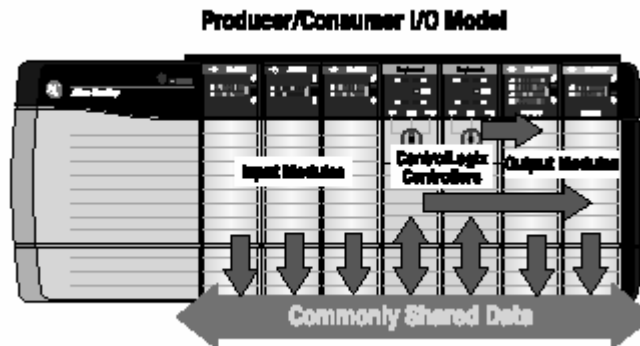


Existe un chasis de 17 espacios, en el cual está conectado los módulos de E/S, el módulo de comunicación Ethernet, módulo de comunicación de E/S remotas y el controlador. El chasis se comunica con la interfaz humano máquina a través de la red Ethernet por medio de un HUB y los variadores de velocidad de C.D. de 75HP a través de la red RIO.

#### 4.3.2.2 Selección de módulos discretos de E/S

La arquitectura Control Logix provee un amplio rango de módulos de entrada y salida y utiliza la tecnología productor consumidor, la cual permite compartir la información de entrada y estado de salida entre múltiples controladores, como se muestra en la figura 36.

Figura 36. Modelo productor consumidor para E/S



Fuente: "Control Logix Selection Guide", **Reference Manual**, Pub. 1756-SG001, 2004, pág. 7.

Cada módulo Control Logix montado en el chasis requiere un bloque Terminal removible para conectar todos los dispositivos de campo. En la figura 37 se muestra un módulo de E/S.

**Figura 37. Módulo digital 1756 de E/S**



Fuente: “Control Logix Selection Guide”, **User Manual**, Pub. 1756-SG001, 2004, pág. 7,.

Debido a la gran cantidad de entradas manejadas para este proyecto, se seleccionó módulos de entrada de 32 puntos en 24VDC. Para los módulos de salidas se utilizaron módulos de 16 puntos a relé aisladas. Las especificaciones de los módulos se observan en la tabla XXI.

**Tabla XXI. Especificaciones eléctricas de módulos discretos de E/S**

<b>Módulo de entrada digital</b>						
<b>Categoría</b>	<b>Tipo</b>	<b>Puntos</b>	<b>Voltaje</b>	<b>Voltaje Operación</b>	<b>Bloque terminal</b>	<b>Potencia disipación</b>
1756-IB32	Entrada	32	12/24VDC	10-31.2VDC	1756-TBCH	6..2W a 60°C
<b>Módulo de salida digital</b>						
<b>Categoría</b>	<b>Tipo</b>	<b>Puntos</b>	<b>Tiempo de retardo ON-OFF</b>	<b>Voltaje Operación</b>	<b>Bloque terminal</b>	<b>Potencia disipación</b>
1756-OW16I	Salida	16 N.A.	10 ms	10-256VAC	1756-TBCH	4.5W a 60°C

### **4.3.2.3 Selección de la red de comunicación**

#### **4.3.2.3.1 Red de comunicación Ethernet/IP**

Ethernet/IP protocolo industrial es una red industrial estándar abierta que soporta mensajería en tiempo real e intercambio de mensajes.



Esta emergió debido a la alta demanda de utilizar Ethernet para aplicaciones de control. Para el diseño de este proyecto se selecciono la red Ethernet/IP para la comunicación entre el controlador y la interfaz humano máquina por ser sencilla la configuración, instalación y de bajo costo. Algunas de las características de esta red son:

- manejo de datos de planta y envío de datos regular;
- configuración, recolección de datos, y control en una red de alta velocidad;
- útil en aplicaciones de tiempo crítico;
- conexión a Internet/intranet.

El módulo de comunicación 1756-ENBT es el dispositivo de comunicación entre el controlador y la interfaz humano máquina. Las características de este módulo son:

- controla E/S sobre una red Ethernet;
- actúa como un adaptador para E/S distribuidas en enlaces remotos;
- actúa como pasarela para dirigir mensajes a dispositivos sobre la red.

**Tabla XXII. Especificaciones módulo de comunicación Ethernet/IP, 1756-ENBT**

<b>Módulo de comunicación Ethernet/IP</b>			
<b>Categoría</b>	<b>Velocidad</b>	<b>Conexiones</b>	<b>Potencia disipación</b>
1756-ENBT	10/100Mbps	Cada módulo soporta: <ul style="list-style-type: none"> <li>• 64 conexiones TCP/IP.</li> <li>• 128 conexiones Logix (E/S e información.</li> <li>• 5000 mensajes/s.</li> </ul>	6..2W a 60°C

#### 4.3.2.3.2 Red universal de comunicación de E/S remotas

La red universal de comunicación de E/S remotas posee las siguientes características:

- conexiones entre adaptadores de E/S y controladores;
- envío de datos regular;
- control distribuido, cada controlador tiene sus propias E/S y se comunica con un controlador supervisor.

La red de comunicación de E/S remotas se seleccionó para la comunicación entre el controlador y los variadores de velocidad de C.D. 1395. Las características para este el módulo de comunicación 1756-DHRIO, se observan en la tabla XXIII.

**Tabla XXIII. Especificaciones de módulo de comunicación 1756-DHRIO**

Módulo de comunicación DHRIO			
Categoría	Velocidad	Consideración	Potencia disipación
1756-DHRIO	<ul style="list-style-type: none"><li>• 57.6Kbps</li><li>• 115.2Kbps</li><li>• 230.4Kbps</li></ul>	Resistores terminales de 150Ω y 82 Ω se conectan en el módulo y el dispositivo final, el valor depende de la distancia.	4.5W

#### 4.3.2.4 Selección de controlador

El controlador Control Logix proporciona una solución escalable capaz de direccionar una gran cantidad de puntos de E/S, 128,000 E/S digitales y un máximo de 4000 señales analógicas.

**Tabla XXIV. Especificaciones del controlador Control Logix 1756-L61**

<b>Controlador Control Logix 1756-L61</b>				
	<b>Memoria</b>			
<b>Categoría</b>	<b>Datos y lógica</b>	<b>E/S</b>	<b>Memoria no volátil</b>	<b>Potencia disipación</b>
1756-L61	2Mbytes	478Kbytes	Tarjeta CompactFlash	3.5W

El controlador puede ser colocado en cualquier ubicación del chasis y múltiples controladores pueden ser instalados en el mismo chasis. El sistema operativo multitarea soporta 32 tareas configurables que pueden ser priorizadas. Una tarea puede ser continua. Las otras tareas deben ser periódicas o de eventos. Cada tarea puede tener un máximo de 32 programas, cada uno con sus propios datos locales, permitiendo construir máquinas virtuales que operan, independientemente, dentro del mismo controlador. El controlador seleccionado para esta aplicación es el 1756-L1, el cual manejará aproximadamente 300 E/S digitales, tendrá el control de los parámetros de los variadores de velocidad de C.D. y la comunicación con la interfaz gráfica. En la tabla XXIV se ilustran las especificaciones para este controlador.

#### **4.3.2.4.1 Requerimientos de memoria**

En la siguiente ecuación se determina un estimado de la memoria necesaria para un controlador. Para esta aplicación se tiene una tarea continua, una tarea de eventos, 300 E/S y dos módulos de comunicación.

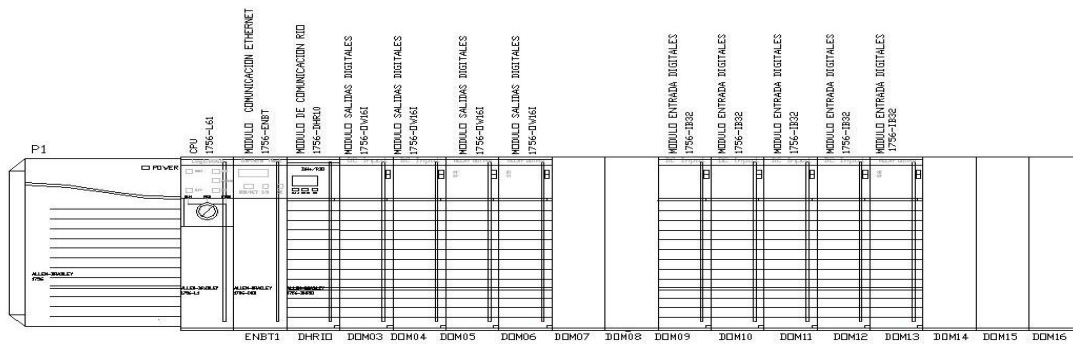
**Tabla XXV. Estimación de uso de memoria**

	<b>Cantidad</b>	<b>Factor</b>	<b>SubTotal (bytes)</b>
Tareas de controlador	2	4,000	8,000
Puntos digitales de E/S	300	400	120,000
Puntos analógicos de E/S	0	2,600	0
Módulos de comunicación	2	2,000	4,000
	<b>Total</b>		<b>132Kbytes</b>

### 4.3.2.5 Selección de chasis

El sistema Control Logix es un sistema modular que requiere un chasis para alojar los módulos. Se puede colocar cualquier módulo en cualquier ranura. El chasis utilizado para esta aplicación es de 17 espacios. La ubicación de los módulos de E/S, comunicación y el controlador se muestra en la figura 38.

**Figura 38. Ubicación de módulos de E/S, comunicación y controlador**



**Tabla XXVI. Ubicación del controlador, módulos de comunicación y módulos de E/S**

Ubicación de módulos en chasis de 17 espacios			
Ranura	Tipo	Categoría	Descripción
0	Controlador	1756-L61	CPU
1	Comunicación	1756-ENBT	Módulo comunicación Ethernet
2	Comunicación	1756-DHRIO	Módulo comunicación RIO
3	Salida	1756-OW16I	Módulo de salida digital a relé
4	Salida	1756-OW16I	Módulo de salida digital a relé
5	Salida	1756-OW16I	Módulo de salida digital a relé
6	Salida	1756-OW16I	Módulo de salida digital a relé
7	Disponible	-----	-----
8	Disponible	-----	-----
9	Entrada	1756-IB32	Módulo de entrada digital 24VDC
10	Entrada	1756-IB32	Módulo de entrada digital 24VDC
11	Entrada	1756-IB32	Módulo de entrada digital 24VDC
12	Entrada	1756-IB32	Módulo de entrada digital 24VDC
13	Entrada	1756-IB32	Módulo de entrada digital 24VDC
14	Disponible	-----	-----
15	Disponible	-----	-----
16	Disponible	-----	-----

#### 4.3.2.6 Selección de fuente de alimentación

Las fuentes de alimentación Control Logix son usadas junto con el chasis para proveer los voltajes necesarios: 1.2VDC, 3.3VDC, 5VDC y 24VDC. La fuente de alimentación provee la alimentación para el controlador, módulos de comunicación y los módulos de E/S. Las especificaciones eléctricas de la fuente de alimentación 1756-PA75 se observan en la tabla XXVII.

**Tabla XXVII. Especificaciones eléctricas de la fuente de alimentación**

Fuente de alimentación 1756-PA75	
Especificación	Valores
Voltaje de entrada nominal	120VAC o 220VAC
Rango de voltaje de entrada	85...265VAC
Máxima potencia real de entrada	95W
Máxima potencia aparente de entrada	240VA
Máxima carga del transformador	238VA
Frecuencia de entrada	47...63Hz
Máxima corriente de salida de segundo plano	1.5A – 1.2VDC 4A – 3.3VDC 10A – 5VDC 2.8A – 24VDC 75W total

### 4.3.3 Componentes de Software

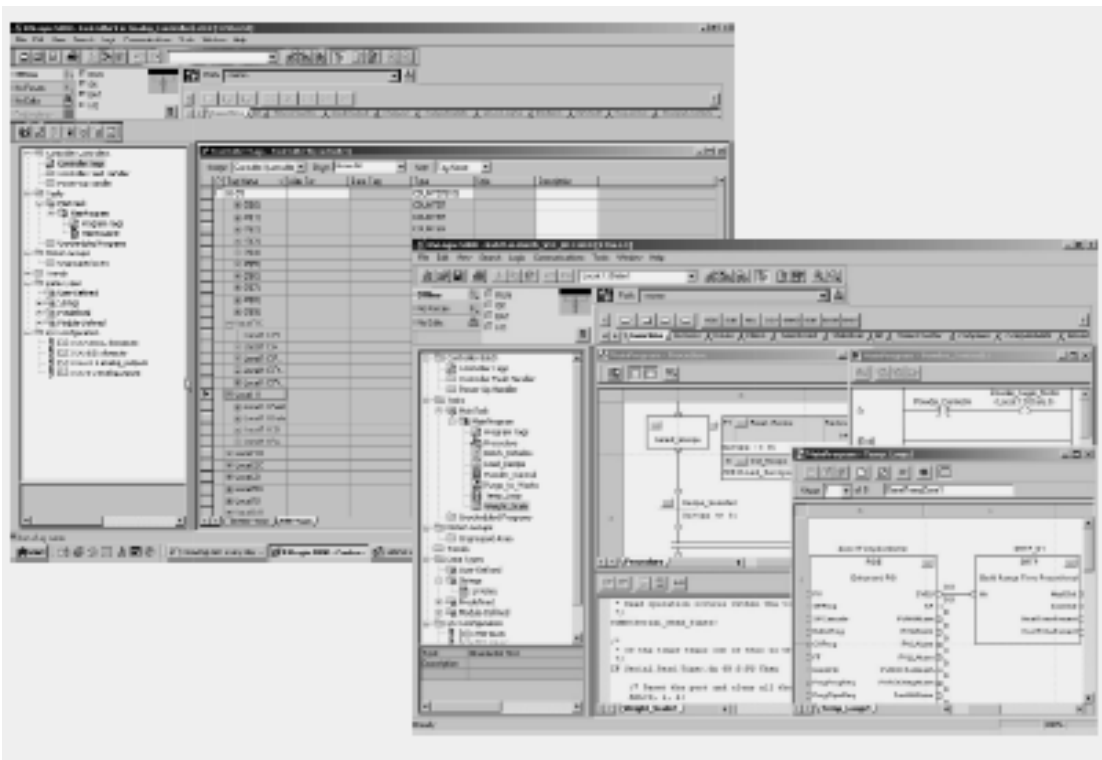
#### 4.3.3.1 RSLogix 5000

El software RSLogix 5000 se utiliza para la programación del controlador lógico programable. Mediante este software se realiza la configuración de los módulos digitales de E/S y de comunicación.

Una de las mayores ventajas al programar utilizando la plataforma Control Logix es que se puede desarrollar la programación del controlador en paralelo con el diseño eléctrico del sistema, lo cual no era posible con la plataforma convencional. Esta característica reduce el tiempo de ingeniería de programación.

El uso de tags y estructuras hace posible que se asignen direcciones con el nombre de la máquina o dispositivo, con esto, se pueden crear máquinas virtuales en el controlador. Esto ayuda para solución de fallas y mantenimiento del sistema.

**Figura 39. Software de programación RSLogix 5000**



RSLogix 5000 es un software diseñado de acuerdo a la norma IEC 61131-3 el cual ofrece programación en lógica de escalera, texto estructurado, diagrama de bloques de funciones y funciones de secuencias. Para la programación de este proyecto se seleccionó la programación en lógica en escalera ya que es fácil de utilizar y además, el control se limitaba a señales discretas, apagado/encendido de válvulas, señales de entrada de manijas, sensores inductivos, etc.

#### **4.3.3.2 RSView Studio**

RSView Studio es un software que se utiliza para crear las pantallas de aplicación que controlan, monitorean el proceso o máquina. Para nuestro caso, en este software se desarrollaron las pantallas de interfaz entre el operador y la rotativa para observar el estado de las válvulas, sensores, motores, fallas del sistema, etc. Este software está diseñado para trabajar en Windows 2000/XP. Algunas de las características de este software son:

- se pueden simular las aplicaciones durante el desarrollo de estas;
- se dispone de un editor completo de gráficas para crear pantallas;
- poderosas herramientas de edición para dibujar, rotar, agrupar, alinear, copiar y pegar objetos;
- incluye animación de objetos: relleno, posición, visibilidad, rotación, tamaño, color y desplazamiento;
- se puede importar gráficos de otros paquetes de software.

#### 4.3.3.3 RSNetworx

El software RSNetworx es la herramienta de configuración para el control de la red de comunicación. Con este software se puede crear una representación gráfica de la configuración de la red y se puede configurar los parámetros definidos de la red. A través de este software se configuraron los parámetros de la red Ethernet: dirección IP y máscara de subred. En la tabla XXVIII se ilustran los parámetros de red configurados a través del software RSNetworx para la red Ethernet del sistema. La dirección IP asignada al módulo Ethernet no sólo le da identidad a ese dispositivo, sino que el chasis completo de 17 espacios es identificado en la red con esa misma dirección IP.

**Tabla XXVIII. Parámetros de red Ethernet de los nodos de la rotativa de impresión**

Dispositivo	Dirección IP	Máscara de Subred
Módulo Ethernet, chasis de 17 espacios	198.162.0.125	255.255.255.0
PanelView Plus 1500	198.162.0.126	255.255.255.0

#### 4.3.3.4 Software RSLinx

RSLinx es un software de comunicación que se utiliza para configuración de parámetros de red de los dispositivos del sistema. RSLinx puede soportar múltiples aplicaciones de software comunicándose, simultáneamente, a una variedad de dispositivos en varias redes diferentes. Para la carga, descarga o edición en línea del programa editado en el software RSLogix 5000, RSLinx sirve como medio para la comunicación entre el controlador y la computadora.

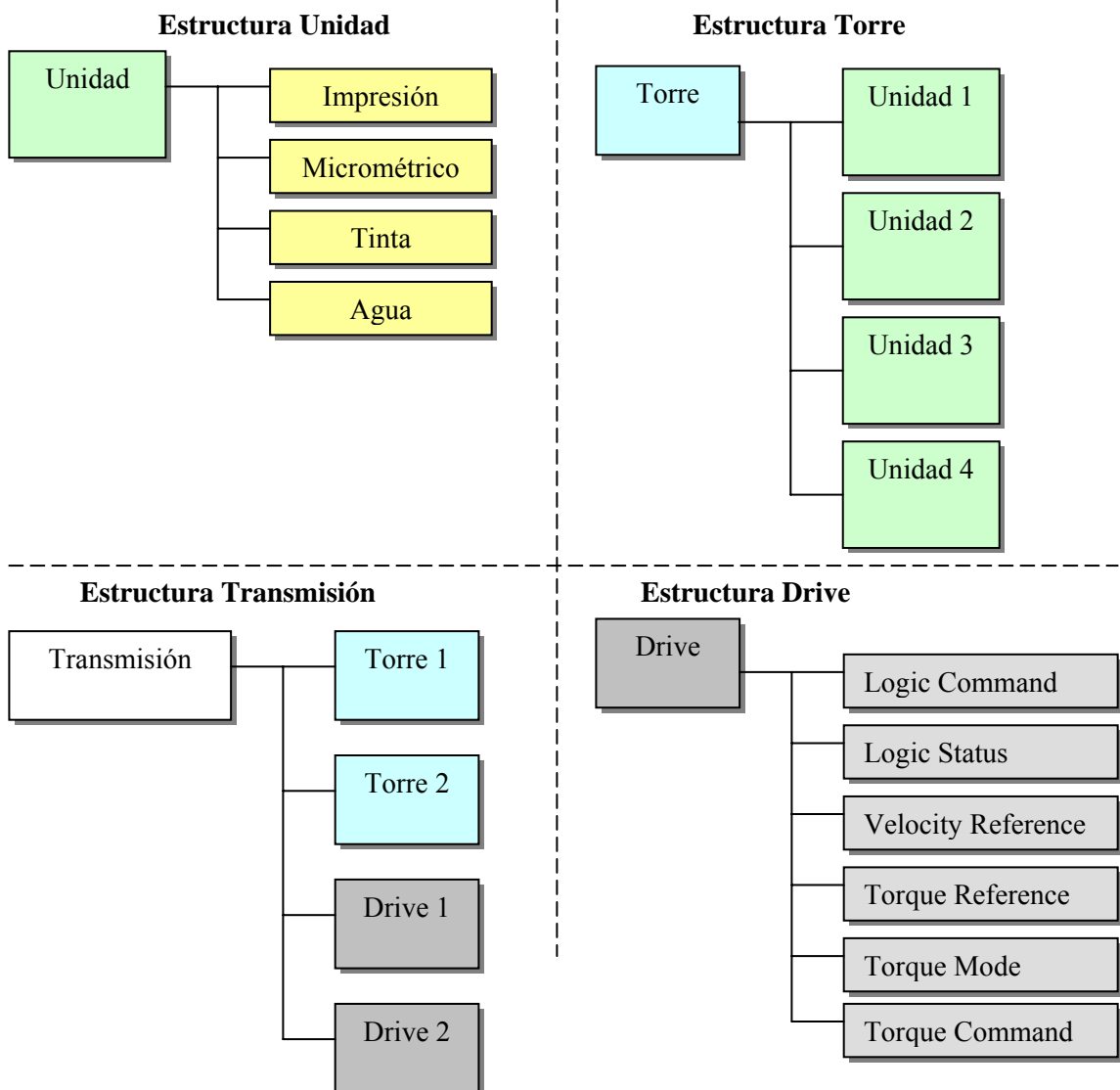
Al igual que RSNetworx, la configuración del módulo Ethernet se puede realizar en RSLinx pero la configuración del PanelView PLUS 1500 no se puede hacer, el cual requiere del software RSLinx Enterprise que viene incorporado con RSView Studio.



#### 4.3.4 Programación del autómata

Para la programación del autómata se utilizó el software RSLogix 5000 a través de la lógica en escalera. Para trabajar en forma ordenada se organizó el programa en 5 subrutinas. Aprovechando las ventajas que ofrece la plataforma logix, se crearon estructuras de acuerdo a los componentes físicos de la máquina. En la figura 40 se observa la estructura unidad, torre, drive y transmisión.

Figura 40. Estructuras: Unidad, drive, torre y transmisión



- **Unidad:** esta estructura que esta dividida en: Impresión, micrométrico, tinta y agua; las cuales son las cuatro válvulas de cada unidad de impresión.
- **Torre:** esta estructura consta de cuatro unidades. Físicamente, cada torre está formada de cuatro unidades de impresión, un color para cada una.
- **Drive:** esta estructura contiene cuatro palabras de control de los drives de C.D. de 75HP, que manipulan el paro, arranque, torque y velocidad de los motores.
- **Transmisión:** esta estructura consta de dos torres y de dos drives.

#### 4.3.4.1 Rutina principal

Dentro de la rutina principal se encuentra el control general de la transmisión. Comando de arranque, comando de trotar, comando de seguridad, comando de paro, aceleración y desaceleración de los motores.

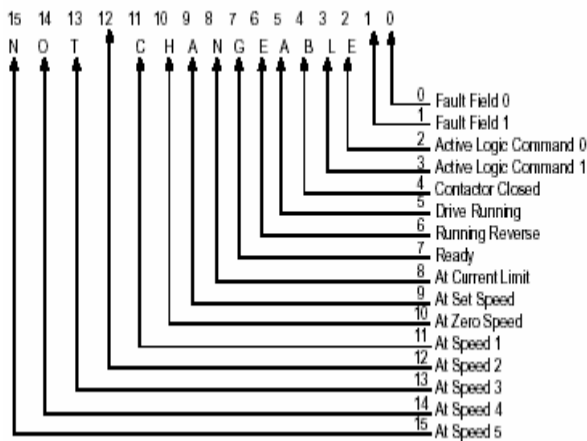
- **Comando de arranque:** el comando de arranque da la autorización para que la transmisión arranque, esto ocurre si no está activado el comando de seguridad o comando de paro. La transmisión arranca con un 10% de la velocidad nominal, dando un aviso a través de una bocina durante 3 segundos antes del arranque, esto se realiza con el propósito de dar un aviso sonoro a los operarios que la transmisión va a iniciar a girar y eviten poner las manos o cualquier parte del cuerpo en partes móviles de la máquina que puedan causarles accidentes.
- **Comando de paro:** en el tablero de control principal existe un botón de paro. El comando de paro se activa cuando el operario que está maniobrando la rotativa de impresión ve alguna anomalía en la máquina y, entonces, requiera parar inmediatamente la máquina presionando el botón de paro.

- **Comando de trotar:** en cada unidad de la transmisión hay un pulsador normalmente abierto llamado trotar. Este pulsador se utiliza para propósitos de mantenimiento y arranque inicial de la rotativa de impresión. Al pulsar este botón la rotativa arranca a una velocidad del 1% de la velocidad nominal, velocidad, suficientemente, lenta, para poder limpiar partes móviles de la máquina o insertar papel para iniciar el proceso de impresión. Este comando, también, está acompañado de un aviso sonoro previo el arranque de la transmisión para seguridad de los operarios.
- **Comando de seguridad:** el comando de seguridad se utiliza para parar, completamente, la rotativa de impresión al ocurrir una falla o alarma. Estas pueden ser: falla de uno de los drives, detección de papel roto, la rotativa ha sido deshabilitada, o, uno de los botones de seguridad esta activado. Cada unidad de impresión tiene un botón de seguridad que se utiliza para casos de emergencia, por ejemplo, cuando algún operario presiona este botón por alguna circunstancia de emergencia, un operario metió la mano accidentalmente en alguna parte móvil, para la rotativa instantáneamente.
- **Aceleración y desaceleración:** los motores de C.D. aceleran desde 0 hasta un 100% de la velocidad nominal. La aceleración/desaceleración de los motores se programo con un potenciómetro digital. En el tablero principal, hay un botón llamado FASTER y otro SLOWER, los cuales aceleran y desaceleran la transmisión, respectivamente. Al presionar y mantener pulsado el pulsador FASTER, la transmisión acelera a una razón de 0.5%/segundo, lo mismo se puede decir para la desaceleración con el pulsador SLOWER.

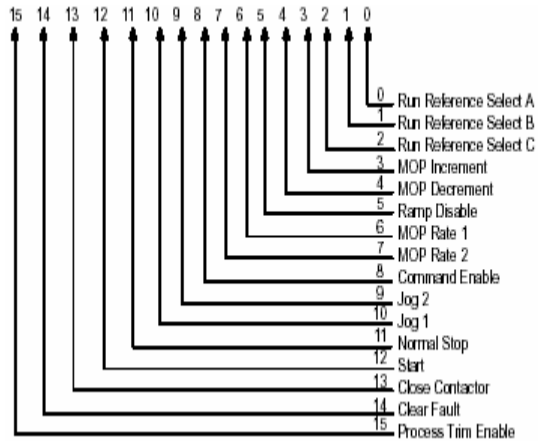
#### 4.3.4.2 Rutina control de variadores de velocidad

Los motores de C.D. de 75HP trabajan en conjunto para compartir la carga mecánica total. Los drives se encargan de regular la velocidad y el torque de los motores cuando trabajan acoplados e individualmente. Los variadores de velocidad son completamente programables a través de la red comunicación RIO, lo cual permite controlar el compartimiento de carga. Cuando trabajan acoplados los motores, ambos giran en sincronía cada uno con cierto porcentaje de la carga dependiendo de cual es el maestro y cual esclavo, seguidor. La operación en paralelo de lo motores depende de varios factores: el primer factor es el color de impresión lo cual determina las secciones a utilizar, y el segundo factor las unidades que serán acopladas a la transmisión.

**Figura 41. Palabra Logic Status**



**Figura 42. Palabra Logic Command**



En esta rutina está el control de los drives de velocidad: el arranque/paro, habilitación/deshabilitación, selección de maestro/esclavo, control de torque. Los drives de velocidad poseen cinco palabras de control a través de las cuales se realiza el control: Logic Command, Logic Status, Velocity Reference, Torque Mode y Torque Command .

Estas palabras de control son manipuladas a través de programación en el controlador, la transferencia de datos entre el controlador y los drives se hace por medio de la red remota de E/S (RIO). En las figuras 41 y 42 se muestran las palabras Logic Status y Logic Command con sus respectivos bits. Los bits sombreados con gris son los bits utilizados para control de los drives.

- **Arranque/paro:** el arranque/paro de los drives está relacionado con el comando de arranque que se encuentra en subrutina principal. Los motores de C.D. tienen un retardo de 3 segundos para el arranque. En la palabra de control Logic Command existe un bit para el arranque/paro, y es por medio de este bit que se controla.
- **Habilitación/deshabilitación:** los drives poseen un bit de habilitación que permite el arranque de estos. Este bit es parte de palabra Logic Command, Command Enable. La habilitación/deshabilitación de los drives la realiza el operario desde la interfaz humano máquina, y solamente es permitida cuando los motores están parados, si la rotativa esta en marcha no es permitida la deshabilitación de cualquiera de los drives.
- **Selección maestro/esclavo:** la selección de maestro/esclavo de los drives se hace por medio de la palabra de control Toque Mode. Al escribir 1d el drive actúa como maestro, al escribir 2d como esclavo. En la combinación maestro/esclavo ambos drives comparten la carga total del sistema, y giran en sincronía (a la misma velocidad). El drive maestro le “dice” al esclavo a la velocidad que debe girar y el torque que debe de aportar.

#### **4.3.4.3 Rutina de control de unidades**

Las unidades, como se menciono al inicio del capitulo, cada una imprime un color diferente en placas de aluminio que han sido calcadas con anterioridad. Los operarios insertan éstas láminas en las unidades y realizan los ajustes necesarios para la tonalidad, color y brillo de la impresión. Los colores primarios más el negro en conjunto hacen posible la creación de todos los colores.

En la rutina de control de unidades se encuentra la programación para el apagado y encendido de las válvulas de impresión, micrométrico, tinta y agua. En cada unidad hay un selector de manual/automático para el encendido de cada una de las válvulas que el operador selecciona.

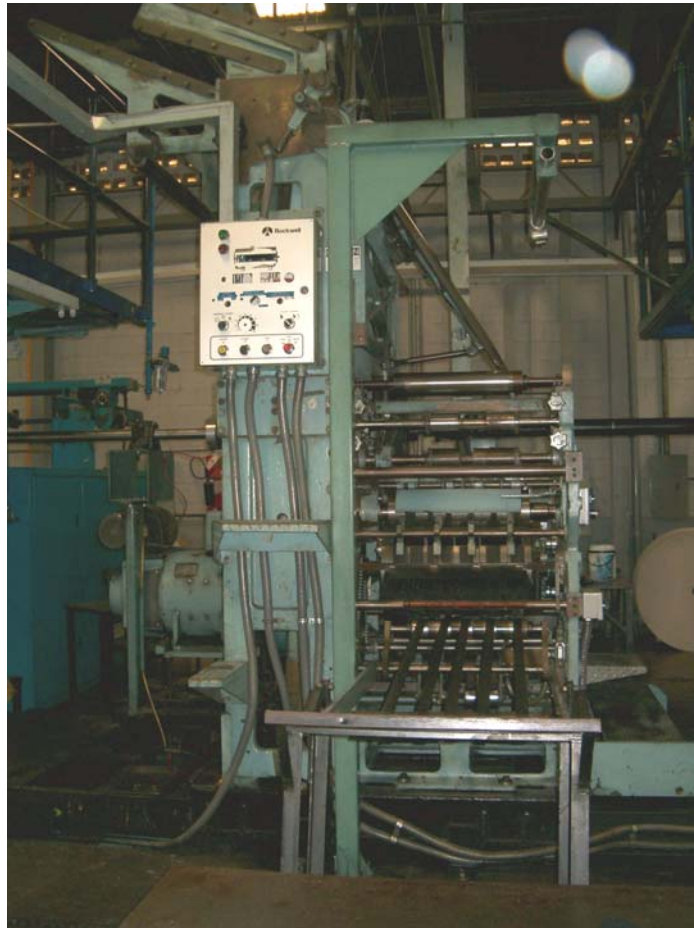
En el funcionamiento manual, las válvulas se encienden al seleccionar el selector en este modo; en el automático, las válvulas se activan de acuerdo a una velocidad de ajuste que se edita en la interfaz humano máquina, por ejemplo: si para micrométrico se tiene un 10%, cuando la transmisión alcance esa velocidad se activará esa válvula automáticamente.

#### **4.3.4.4 Rutina control doblador**

El doblador es la máquina encargada del doblado y corte del papel impreso para la presentación en periódico. Esta máquina es acoplada, mecánicamente, a la rotativa de impresión. Existe un sensor inductivo acoplado a un eje del doblador; la función de este sensor es de contar la cantidad de impresiones por segundo. Esta información es importante, pues de ella se obtiene las impresiones buenas, malas, y la cantidad de impresiones por hora de la rotativa. A pleno funcionamiento, se requiere que la rotativa imprima 30,000 ejemplares/hora.

Esta rutina se programó utilizando una tarea de eventos, activada por el sensor inductivo. Cada vez que el sensor inductivo detecta una revolución del eje, se cuenta como una impresión. Este sensor trabaja en 24VDC conectado a un módulo de entrada digital. La frecuencia de giro del eje es de 48Hz, mientras que la respuesta máxima del módulo digital es de 72Hz, lo cual es suficiente para contar los pulsos. En la figura 43 se observa el doblador de periódicos.

**Figura 43. Doblador de periódicos**



#### **4.3.4.5 Rutina control de enkels**

Los enkels son los alimentadores de papel hacia la transmisión, cada torre tiene incorporado uno. Estas máquinas tienen incorporado un sistema de control que regula la tensión del papel para evitar que este se rompa al alimentar las torres. La subrutina para el control de los enkels se encarga de controlar las electroválvulas que controlan la alineación del rollo de papel. El proceso de alineación del papel se hace al inicio de operación de la rotativa, en la que se hacen los ajustes de color de las unidades.

#### **4.3.5 Programación de la interfaz humano máquina**

Para la programación del Panel View 1500 se utilizó el software RSView Studio en donde se crearon las pantallas de: monitoreo, control, revisión e historial de alarmas de la rotativa de impresión. Esto permite un fácil manejo y supervisión de la impresión de periódicos de la rotativa. La animación es parte importante que ofrece la pantalla, pues permite ver los elementos activos de la rotativa, ayudando a determinar fallas en los drives, sensores, de la rotativa. Una herramienta importante que se incorporó a las pantallas es mostrar la cantidad de periódicos impresos por hora y la cantidad de periódicos impresos malos y buenos en un turno de trabajo. Esta valiosa información permite al departamento de producción cuantificar los periódicos que se distribuyen diariamente.

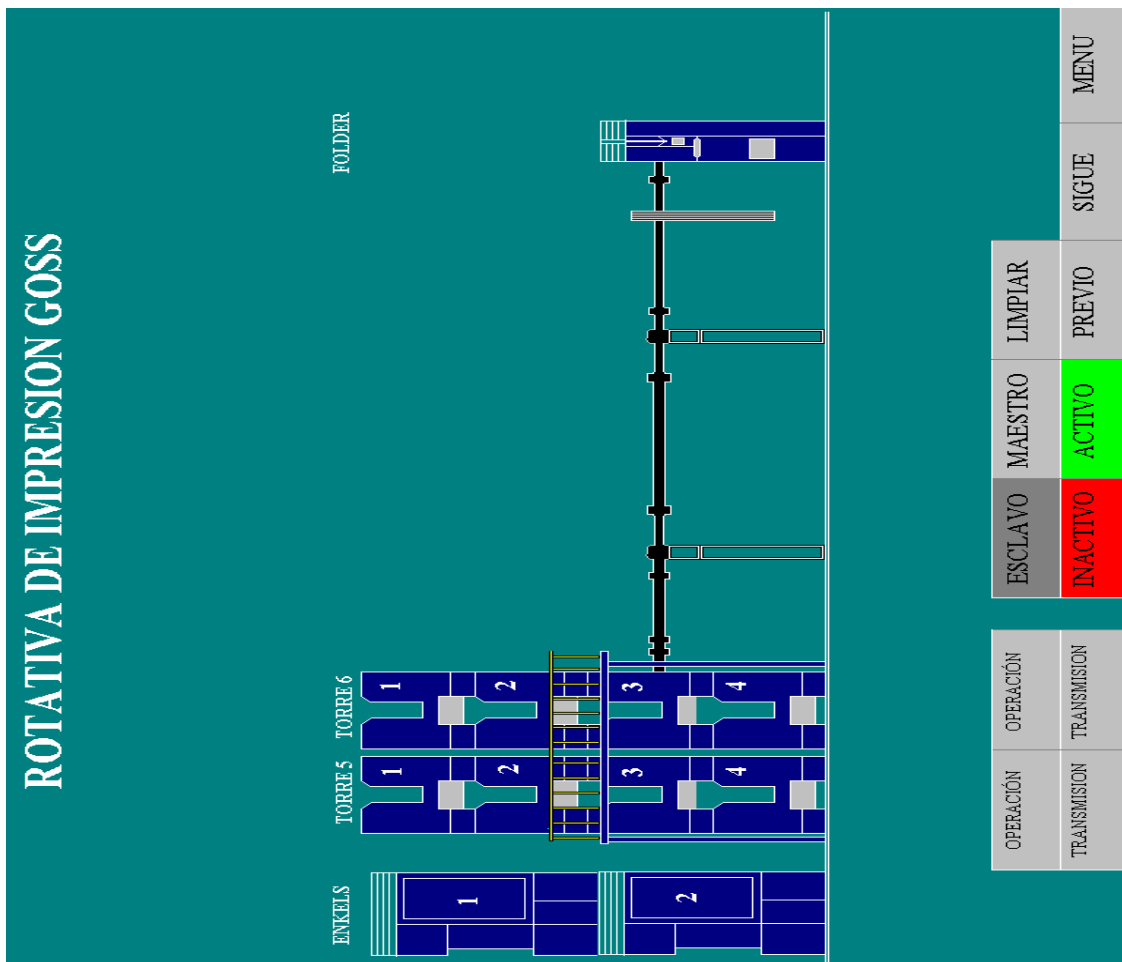
Las pantallas se diseñaron para que el operario trabaje en un ambiente agradable, amigable y sencillo. Aprovechando que la pantalla cuenta con un teclado y pantalla táctil, todas las funciones pueden manejarse desde ambos. Todas las imágenes estáticas, de fondo de pantalla, son a color 16 bits, sacando provecho de la memoria de 64MB del Panel View. Enseguida se explican las pantallas de aplicación para el manejo de la rotativa.



### 4.3.5.1 Pantalla rotativa de impresión GOSS

En esta pantalla se tiene el control y monitoreo general de la rotativa. Desde aquí, se controla y selecciona el drive maestro/esclavo, activación/desactivación, monitoreo de papel roto, limpieza de fallas de la rotativa. El control de los drives de velocidad, lo tiene únicamente el supervisor autorizado de planta, esto para evitar que no se seleccione un inadecuado acople de lo motores y no sobrecalentarlos.

Figura 40. Pantalla rotativa de impresión GOSS

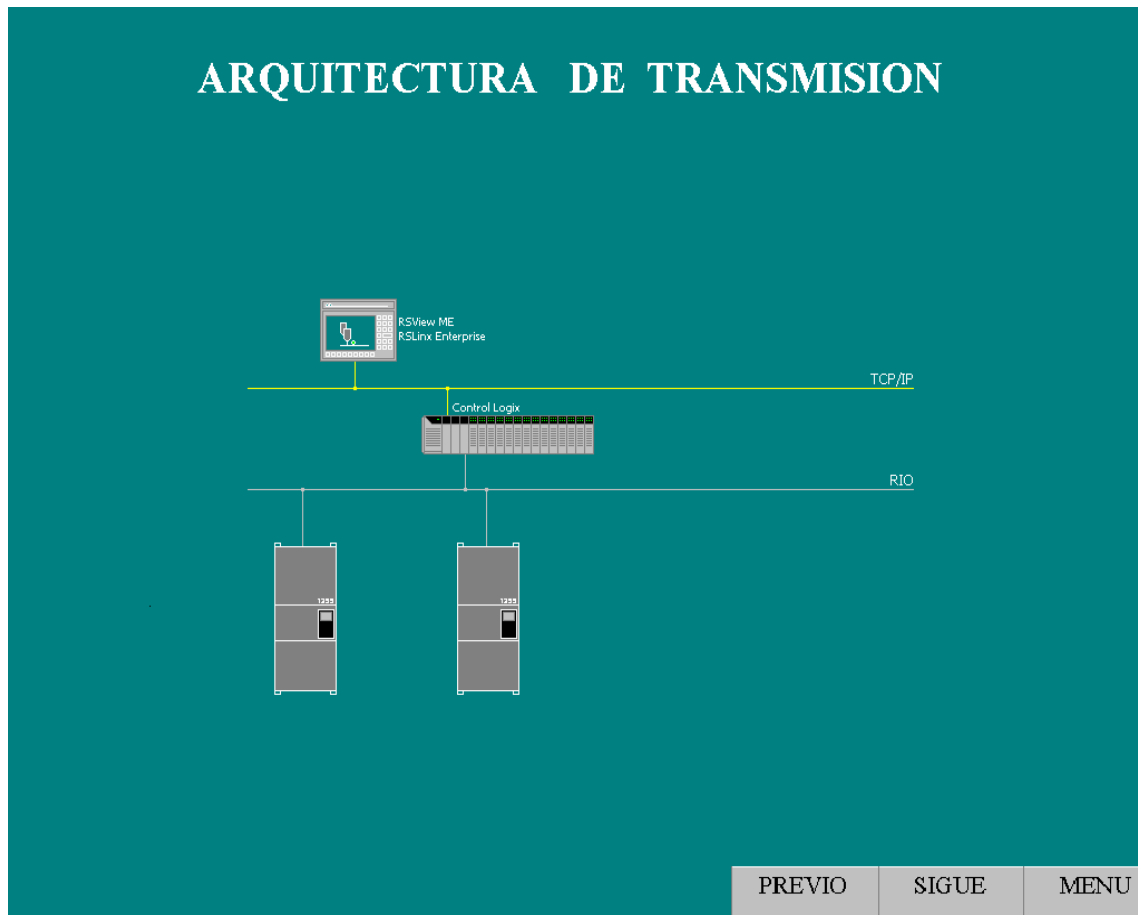


Fuente: “Pantallas rotativa de impresión GOSS”, Software RSView Studio, 2004.

#### 4.3.5.2 Pantalla arquitectura de rotativa de impresión GOSS

En esta pantalla se muestra la arquitectura de la rotativa: redes de comunicación de interconexión de dispositivos, chasis del controlador Control Logix, drives de velocidad 1395. Esta es una pantalla estática, el operador no puede realizar ninguna operación solamente esta para que tenga una idea de cómo esta interconectado el sistema desde una perspectiva general.

**Figura 41. Pantalla arquitectura rotativa de impresión GOSS**

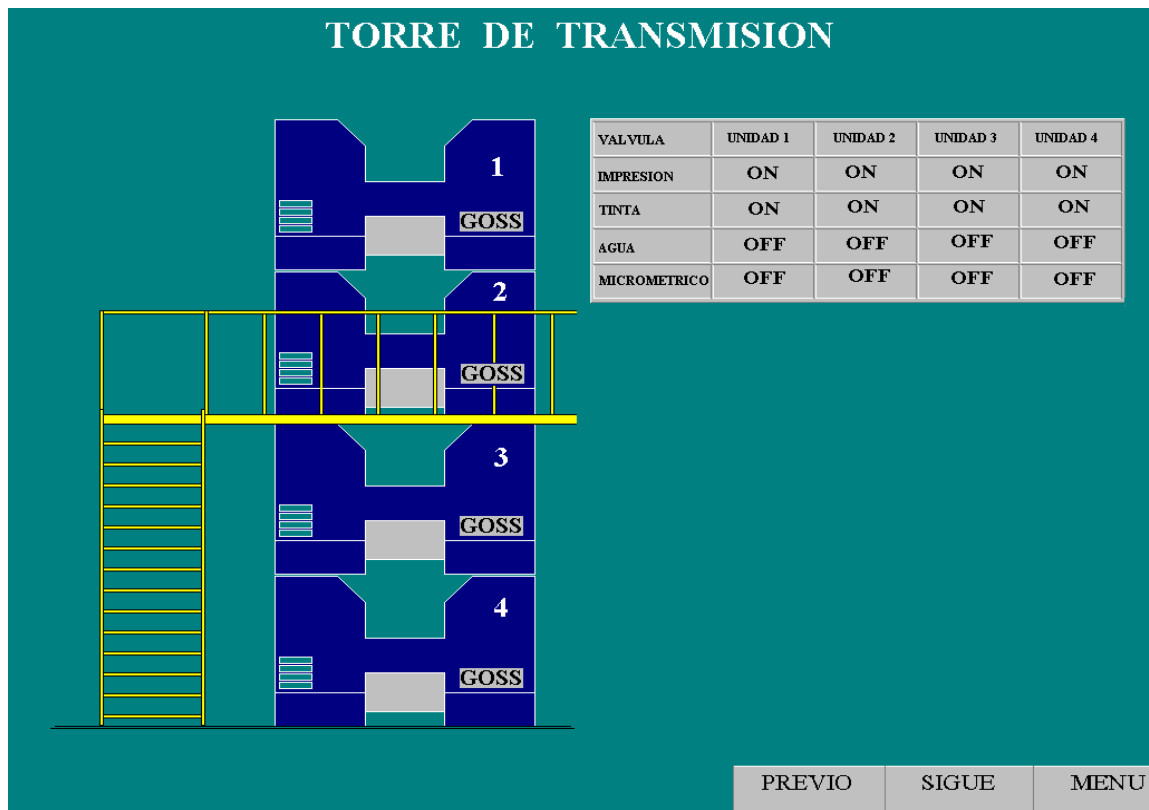


Fuente: “**Pantallas rotativa de impresión GOSS**”, Software RSView Studio, 2004.

### 4.3.5.3 Pantalla torre de transmisión

En esta pantalla se muestra el estado de las válvulas de impresión, agua, tinta y micrométrico de cada unidad. El operador enciende/apaga las válvulas desde las unidades en cada una de las torres y además, desde el pupitre central. Las válvulas pueden activarse en modo manual o automático. Por ejemplo, si el operador desea activar la válvula de impresión en modo manual debe de seleccionar en manual la manija que está en la unidad deseada y en el pupitre. Para esta transmisión hay dos torres de impresión, por lo tanto, existe una pantalla para la torre 5 y otra para torre 6.

Figura 42. Pantalla de torre rotativa de impresión GOSS



Fuente: “Pantallas rotativa de impresión GOSS”, Software RSView Studio, 2004.

#### 4.3.6 Puesta en marcha

La puesta en marcha del sistema comprende hacer funcionar el sistema completo de impresión. Para esto hay varios procedimientos ordenados que se realizan para evitar daño al equipo y accidentes al personal.

La puesta en marcha es un proceso que lleva varias semanas finalizarlo, pues deben de hacerse varias pruebas previas a la operación normal del sistema, además, se lleva a cabo en conjunto con los operarios, electricistas y mecánicos encargados de la operación y mantenimiento de la rotativa. Durante la puesta en marcha, se aprovecha para dar una breve inducción al personal para que aprendan a usar y revolver fallas del sistema.

- **Revisión de conexiones:** la revisión de las conexiones de los cables de potencia y control se realiza de acuerdo a los planos eléctricos. Los tableros son revisados eléctricamente antes de instalarlos en sitio con el cliente, para realizar las correcciones necesarias en taller. Para esto se revisa el calibre, color, tipo de cable, borne de conexión, simbología, ubicación de los componentes.
- **Revisión de voltajes:** una vez instalados eléctrica y físicamente los equipos, se revisa los voltajes de potencia y control. Esta revisión debe de realizarse con cuidado, pues en caso de existir una mala conexión o cortocircuito podría dañar el equipo. Para este procedimiento, se inicia revisando las tensiones del lado primario de los transformadores de control, y luego el secundario. Al energizar el equipo se revisa que el equipo encienda y funcione adecuadamente. Para el equipo de control se mide el voltaje de C.D. para el manejo de las E/S digitales. Las tensiones deben de AC deben de estar dentro de un rango del 2%.

- **Parametrización de los drives de C.D:** una vez revisados las conexiones y las tensiones se procede a la parametrización de los drives. Para esto, se ingresan los datos nominales de los motores: voltaje de campo, corriente de armadura máxima, velocidad nominal, tiempo de aceleración, tiempo de desaceleración, potencia. Estos se ingresan desde la terminal de programación. Los drives tienen internamente un software que permite la sintonización automática de los lazos de control: corriente, velocidad y de flujo de campo. Este procedimiento es muy importante pues los drives se encargan de calcular la inercia del sistema, ganancia de los lazos, etc.
- **Carga y prueba de programa del controlador:** al finalizar la parametrización ya se puede hacer las pruebas de comunicación entre el drive y el controlador. Para esto se carga el programa al controlador y se configura el módulo de comunicación de entradas/salidas remotas. Al existir comunicación entre el drive y el controlador se realizan pruebas forzando bits de control desde el controlador. Por ejemplo, para arrancar uno de los motores, se activa el bit Start de la palabra Logic Command. En este procedimiento, también se verifican todas las entradas y salidas digitales de campo; en el caso de las entradas se verifica que cada dispositivo de entrada active la entrada asignada, y en las salidas se fuerza cada una y se verifica que la salida se active.
- **Carga y prueba de programa de Panel View 1500:** en este procedimiento se verifica que exista comunicación entre controlador y el Panel View 1500. Para esto se configura el módulo de comunicación Ethernet y se asignan los parámetros de red a cada dispositivo. Para probar la comunicación se observa el estado de los objetos de las pantallas que se utilizan para monitoreo. Para los botones que se utilizan para escritura se revisa que escriban los datos a los tags correspondientes.



## CONCLUSIONES

- 1 La introducción de nuevas tecnologías en Guatemala como la plataforma de automatización Control Logix, tiende a provocar oposición, ya que, se requiere de inversión en capacitación, apertura a cierto riesgo, hardware y software, etc. Sin embargo, son estas nuevas tecnologías las que nos permite competir en el ámbito internacional en calidad, productividad, eficiencia, eficacia, etc.
- 2 El proceso de diseño de la rotativa de impresión Goss, utilizando la plataforma convencional, hubiera prolongado el tiempo de desarrollo del mismo, pues, la ingeniería concurrente de la plataforma Control Logix permite reducir el tiempo de diseño.
- 3 Las características de la plataforma Control Logix: rápido procesamiento de instrucciones, sistema multitarea, flexibilidad, modularidad de programación, alto rendimiento, etc. se debe en parte al backplane el cual funciona como una red NetLinx de alta velocidad.
- 4 En la plataforma Control Logix los controladores, módulos de E/S, y módulos de comunicación están dotados con inteligencia para funcionar como nodos independientes en la red de comunicación, como resultado se pueden combinar múltiples controladores, módulos de comunicación y E/S sin restricciones en el mismo chasis.





## RECOMENDACIONES

- 1 En Guatemala, existe gran variedad de proyectos funcionales automatizados con la plataforma convencional, por lo que previo a la decisión de diseñar una máquina o proceso con la plataforma Control Logix se recomienda evaluar si es necesario la implementación del proyecto con una plataforma de alto rendimiento como lo es Control Logix o con la plataforma convencional que podría satisfacer las demandas del proyecto.
- 2 La apertura al riesgo en la introducción de una nueva tecnología es necesaria para la aplicación de sistemas, procesos, etc. de alta tecnología. Inicialmente, el costo puede ser alto en cuanto a recursos materiales, económicos y financieros, pero las ventajas que ofrece una nueva tecnología recompensa ese esfuerzo, por lo tanto, es recomendable afrontar y desafiar el ingreso de nuevas tecnologías en la industria nacional.
- 3 La plataforma Control Logix provee una gran cantidad de ventajas que se le puede sacar provecho. Sin embargo, se recomienda que para obtener el máximo de las ventajas, los ingenieros de diseño sean capacitados en esta tecnología al grado de especialización, para poder conocer y explotar esta plataforma de automatización.



## BIBLIOGRAFÍA

- 1 Rockwell Automation, **Control Logix Ethernet Communication Interface Module**, Pub. 1756-UM051B-EN-P, Nov/2000
- 2 Rockwell Automation, **Logix 5000 Controllers Common Procedures**, Pub. 1756-PM001-EN-P, March/2004.
- 3 Rockwell Automation, **Logix5000 Controllers Design Considerations**, Pub. 1756-RM-EN-P, May/2004.
- 4 Rockwell Automation, **For an Integrated Architecture Take Control with Control Logix**, Pub. 1756-BR001D-EN-P, May/2003.
- 5 Rockwell Automation, **A Controller that Handles It All**, Pub. 1756-1.3, Nov/1999.
- 6 L.A. Bryan, E.A. Bryan, **Programmable Controllers, Theory and Implementation**, Second Edition, Industrial Text, 1997.
- 7 Galindo Morán, Sergio Iván. **Aplicación de Redes Abiertas de Comunicación en la Automatización Industrial Nacional** Tesis de Ing. Electrónica. USAC, Facultad de Ingeniería. 2002. 181pp.
- 8 Mendoza Bautista, Israel Ebenezer. **Diseño de Automatización del Proceso de Cocimiento de Productos alimenticios en la industria utilizando Equipos de Cocción (Marmitas)**. Tesis de Ing. Electrónica. USAC, Facultad de Ingeniería. 2004.

## INTERNET

10. Allen Bradley, [www.ab.com/networks](http://www.ab.com/networks), 2004.
11. Allen Bradley, [www.ab.com/drives](http://www.ab.com/drives), 2004.
12. Allen Bradley, [www.ab.com/Control Logix](http://www.ab.com/Control Logix), 2004.