



SUBSECRETARÍA DE EDUCACIÓN SUPERIOR  
DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR TECNOLÓGICA  
INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ

SECRETARÍA DE  
EDUCACIÓN PÚBLICA

SEP

# **INFORME TÉCNICO DE RESIDENCIA PROFESIONAL**

**INGENIERÍA MECÁNICA**

**PRESENTA:**

**CARLOS RENÉ ESCANDÓN AMORES**

**NOMBRE DEL PROYECTO:**

**Estandarización de rutinas de control en PLC's para  
aplicaciones con inversores, arrancadores suaves,  
servomotores y arranques a tensión plena.**

**PERIODO DE REALIZACIÓN:**

**AGOSTO-DICIEMBRE 2012**

## Índice

<b>1</b>	<b>Introducción.....</b>	<b>5</b>
<b>2</b>	<b>Justificación.....</b>	<b>6</b>
<b>3</b>	<b>Objetivos del proyecto. ....</b>	<b>6</b>
<b>4</b>	<b>Caracterización del área de residencia. ....</b>	<b>7</b>
<b>5</b>	<b>Problemas a resolver.....</b>	<b>8</b>
<b>6</b>	<b>Alcances y limitaciones. ....</b>	<b>8</b>
<b>6.1</b>	<b>Alcances.....</b>	<b>8</b>
<b>6.2</b>	<b>Limitaciones.....</b>	<b>8</b>
<b>7</b>	<b>Fundamento teórico. ....</b>	<b>9</b>
<b>7.1</b>	<b>Un poco de historia del PLC y definición.....</b>	<b>9</b>
<b>7.2</b>	<b>¿Qué es la Lógica Programable? .....</b>	<b>9</b>
7.2.1	Matrices Lógicas Programables, PLA. ....	10
7.2.2	Dispositivos Lógicos Programables Complejos.....	11
<b>7.3</b>	<b>Aplicación de los PLC´s. ....</b>	<b>11</b>
<b>7.4</b>	<b>Estructura de un PLC. ....</b>	<b>11</b>
7.4.1	La memoria.....	12
7.4.2	CPU.....	13
7.4.3	Unidades de E/S.....	14
7.4.4	Interfaces.....	15
7.4.5	Unidades de Programación. ....	15
7.4.6	Dispositivos Periféricos.....	15
<b>7.5</b>	<b>Programación de PLC´s. ....</b>	<b>16</b>
<b>8</b>	<b>Procedimiento y descripción de las actividades realizadas.....</b>	<b>17</b>

<b>9</b>	<b>Resultados, planos, gráficas, prototipos y programas.....</b>	<b>18</b>
<b>10</b>	<b>Estándar de programación para las rutinas de control de Nivel 1 con ControlLogix 5000 .....</b>	<b>18</b>
<b>10.1</b>	<b>Niveles de automatización y control. ....</b>	<b>19</b>
10.1.1	Nivel 0. ....	19
10.1.2	Nivel 1. ....	20
10.1.3	HMI Nivel 1. ....	21
10.1.4	Nivel 2. ....	21
<b>10.2</b>	<b>Operación de la planta sin Nivel 2.....</b>	<b>21</b>
<b>10.3</b>	<b>Definición de Hardware Familia ControlLogix.....</b>	<b>22</b>
<b>10.4</b>	<b>Configuración: Nombres de dispositivos .....</b>	<b>24</b>
10.4.1	Nombre del proyecto .....	24
10.4.2	Nombre del archivo.....	24
10.4.3	Nombre del Controlador. ....	25
10.4.4	Nombre de chasis.....	25
10.4.5	Nombre de los DROPS.....	25
10.4.6	Numero de nodo en DROPS donde se encuentran las CPU´s. ....	26
10.4.7	Nombre de los Taps. ....	26
10.4.8	Nombre de los Coaxiales de conexión entre Taps.....	26
10.4.9	Nombre de módulos de comunicación.....	26
10.4.10	Nombre de módulos de entrada / salida.....	28
10.4.11	Identificación del cableado a módulos de entrada / salida.....	28
10.4.12	Archivos de configuración de ControlNet y DeviceNet. ....	29
<b>10.5</b>	<b>Tags: nomenclatura de instrumentos y señales .....</b>	<b>29</b>
10.5.1	Conformación de Tags .....	29
10.5.2	Conformación de Tags programación PLC .....	30
<b>10.6</b>	<b>Programación. ....</b>	<b>33</b>
10.6.1	Tags I/O.....	34
10.6.2	Tags y tipo de datos para comunicación con nivel 2.....	34
10.6.3	Lectura y escritura de tags desde nivel 2.....	35
10.6.4	Definición del nombre de variables de Estados .....	36
10.6.5	Definición del nombre de variables de Alarmas .....	37

10.6.6	Definición de nombres de Comandos desde el N2 o N1 al PLC .....	38
10.6.7	Definición de variables de Lectura desde N2 o N1 .....	39
10.6.8	Definición de variables de Escritura desde N2 o N1 .....	41
<b>10.7</b>	<b>Fallas. ....</b>	<b>42</b>
10.7.1	Falla de módulo. ....	42
<b>10.8</b>	<b>Organización de programas y rutinas. ....</b>	<b>43</b>
<b>11</b>	<b>Conclusiones y recomendaciones. ....</b>	<b>44</b>
<b>12</b>	<b>Glosario. ....</b>	<b>45</b>
<b>13</b>	<b>Referencias bibliográficas y virtuales. ....</b>	<b>47</b>

## 1 Introducción.

---

Los Autómatas Programables Industriales, mejor conocidos como Controladores Lógicos Programables (PLC´s), tienen, por sus especiales características de diseño, un campo de aplicación muy extenso. La constante evolución del hardware y software amplía constantemente este campo para poder satisfacer las necesidades que se detectan en el espectro de sus posibilidades reales.

Por eso, debido a los cambios constantes en este ramo, se podrá especificar una metodología de desarrollo y organización de software (estándar) válido en la empresa TERNIUM para ser aplicado a sistemas de automatización y control basados en Controladores Lógicos Programables, así como el hardware utilizado para el desarrollo de los procesos de automatización en dicha empresa y poder marcar un estándar para las rutinas de control.

En TERNIUM, como empresa internacional, se manejan tres tipos principales de PLC´s con sus respectivos programas de configuración, pero durante la residencia profesional únicamente se trabajarán con el PLC marca Allen Bardley, modelo Control Logix serie 5000 es por eso que en este reporte se verán las características de dicho modelo.

## **2 Justificación.**

---

El advenimiento de una cantidad importante de desarrollo de lógica de programación para los PLC's, hace necesario definir un lenguaje común a las áreas de ingeniería, mantenimiento y procesos. Al mismo tiempo se necesita especificar el proceso de automatización para permitir que proveedores externos desarrollen las aplicaciones bajo normas claras y concretas

De esta manera, el software de Nivel 1 (ver marco teórico) de automatización de toda la planta tendrá los mismos lineamientos, a fin de que sea interpretado y mantenido por los diferentes operadores.

Este reporte será útil para toda persona que tenga conocimientos acerca de los PLC's, así como de las definiciones que rodean el campo de la automatización, redes, electricidad básica, motores eléctricos de diversos tipos, mapeo y programación.

## **3 Objetivos del proyecto.**

---

- Conocer las diferentes formas de comunicación entre el PLC y el control de motores y sus aplicaciones industriales.
- Conocer las diferentes formas de comunicación y protocolos más utilizadas y sus ventajas tecnológicas en TERNIUM.
- Entender cada una de las formas de programación correspondiente a cada tipo de equipo.
- Conocimiento de las nomenclaturas de documentos oficiales para la organización TERNIUM.
- Conocer los métodos de difusión de los estándares dentro del departamento de Automatización.

## **4 Caracterización del área de residencia.**

---

Este sector trata de asegurar la ejecución de la implementación eficaz y eficiente de las inversiones y de los proyectos de automatización, control y modelado de procesos productivos del nivel 1 considerado en la empresa, con la finalidad de aumentar la capacidad de producción, disminuir los costos, mejorar la calidad de los productos y servicios y mejorar la gestión industrial en general.

Apoya en la asesoría técnica y en la negociación de los contratos suministros y servicios principales, apoyando técnicamente el proceso de evaluación y seguimiento de los proveedores y contratistas.

Garantizar la implementación en tiempo y forma de las inversiones y de los proyectos de automatización, control y modelado de procesos productivos para todas las áreas de área Manager Norte, con la finalidad de aumentar la capacidad de producción, disminuir los costos, mejorar la calidad de los productos y servicios, así como mejorar la gestión industrial en general.

También tiene que colaborar en el diseño, ejecución y seguimiento de proyectos de ingeniería mediante el desarrollo de estudios de factibilidad, ingeniería básica y de detalle, estudio de especificaciones e inserción de equipos, estandarización de componentes y sistemas, mejora de procesos, y el control de costos, para los proyectos de las áreas de su responsabilidad.

Apoyar en el diseño de los planes estratégicos de las inversiones, en el campo de automatización y control, para las áreas de su responsabilidad. Colaborar en las discusiones técnicas, brindando soporte técnicamente el proceso de evaluación y seguimiento de los procesos.

Colaborar en la recopilación de información para la estandarización de procesos, de tal manera que los nuevos proyectos del departamento puedan seguir un lineamiento único para facilitar la realización, el desempeño y, en caso de que exista alguna falla, poder encontrarla de manera rápida y óptima para posteriormente corregirla.

## **5 Problemas a resolver.**

---

- Investigar las rutinas utilizadas en el último año en los diferentes proyectos y que estén relacionados con temas de control de motores.
- Elaborar las rutinas base estándar que den la mejor solución a los proyectos de TERNIUM.
- Generación de los documentos base para la aprobación del Comité de Estandarización de N1.
- Presentación de la documentación al Comité de Estándares de Nivel 1 y al Departamento de Automatización.

## **6 Alcances y limitaciones.**

---

### **6.1 Alcances.**

Todas las áreas de automatización en la empresa TERNIUM de nivel 1 y 2 (ver marco teórico para las definiciones), que requieran información sobre estándares para el manejo y programación de los PLC´s Allen Bradley.

### **6.2 Limitaciones.**

Se limitará únicamente a PLC´s Allen Bradley con programación en escalera y bloques para el Área Manager Norte de TERNIUM, abarcando características básicas para la forma de programación que se debe manejar en esta área.



## 7 Fundamento teórico.

---

### 7.1 Un poco de historia del PLC y definición.

Los Controladores Lógicos Programables, (PLC's, Programmable Logic Controller) nacieron a finales de la década de los 60s y principios de los 70s. Las industrias que propiciaron este desarrollo fueron las automotrices. En ellas se usaban sistemas industriales basadas en relevadores (relés), en sus sistemas de manufactura.

En estos últimos años los sistemas informáticos, mecánicos, electrónicos y de comunicaciones (redes y protocolos) se integran entre ellos en un todo armónico y funcional como un único complejo automático. Así nació la automatización que ya se ha convertido en el fundamento de todos los procesos industriales avanzados y, en consecuencia, una disciplina de base común a todas las direcciones de especialización profesional. La automatización en su concepto más amplio es el control y la gestión de sistemas automáticos accionados, mediante un conjunto de técnicas y dispositivos particulares.

Un Controlador Lógico Programable (PLC) o Autómata Programable, es un equipo electrónico programable, basado en microprocesadores, diseñado para controlar procesos secuenciales en tiempo real y en ambientes industriales. Su manejo y programación puede ser realizada por personal eléctrico o electrónico sin conocimientos informáticos. Realiza funciones lógicas: series, paralelos, temporizaciones, cuentas y otras más potentes como cálculos, regulaciones, etc. Los PLC han evolucionado rápidamente desde su concepción en 1968, por un grupo de ingenieros de General Motors.

### 7.2 ¿Qué es la Lógica Programable?

La lógica programable, es una familia de componentes que contienen conjuntos de elementos lógicos (AND, OR, NOT, LATCH, FLIP-FLOP, etc.) que pueden configurarse para que cumplan cualquier función lógica que el usuario desee y que el componente soporte. Podríamos decir que los Dispositivos Lógicos Programables por el usuario (DLP) son la "antesala" de Controladores Lógicos programables.

### 7.2.1 Matrices Lógicas Programables, PLA.

Las PLA son matrices lógicas programables. Estos dispositivos contienen compuertas AND y OR programables lo que permite a cualquier término AND alimentar cualquier término OR. Las PLA probablemente tienen la mayor flexibilidad frente a otros dispositivos con respecto a la lógica funcional. Normalmente poseen realimentación desde la matriz OR hacia la matriz AND que puede usarse para implementar máquinas de estado asíncronas (ver figura 1).

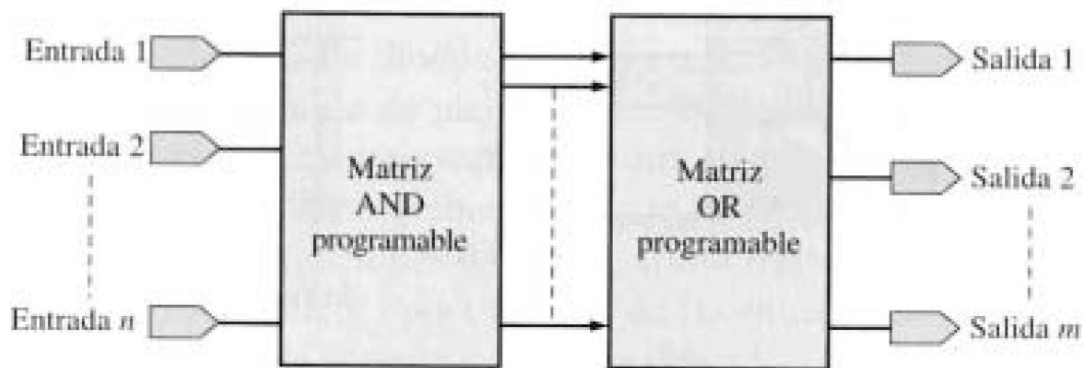


Figura 1: Ejemplo de una Matriz Lógica Programable, también se conoce como FPLA (Field Programmable Logic Array), debido a que es el usuario y no el fabricante el que la programa.

Se denomina máquina de estados a un modelo de comportamiento de un sistema con entradas y salidas, en donde las salidas dependen no sólo de las señales de entradas actuales sino también de las anteriores. Las máquinas de estados se definen como un conjunto de estados que sirve de intermediario en esta relación de entradas y salidas, haciendo que el historial de señales de entrada determine, para cada instante, un estado para la máquina, de forma tal que la salida depende únicamente del estado y las entradas actuales.

La mayoría de las máquinas de estado, sin embargo, se implementan como máquinas síncronas. Con esta perspectiva, los fabricantes crearon un tipo de PLA denominado Secuencial que posee registros de realimentación desde la salida de la matriz OR hacia la matriz AND.

### 7.2.2 Dispositivos Lógicos Programables Complejos.

Los DLP's complejos son lo que el nombre implica, Dispositivos Complejos de Lógica Programable. Se consideran PAL muy grandes que tienen algunas características de las PLA. La arquitectura básica es muy parecida a la PAL con la capacidad para aumentar la cantidad de compuertas AND para cualquier compuerta OR fija. Esto se puede realizar quitando compuertas AND adyacentes o empleando compuertas AND desde una matriz. Esto permite que cualquier diseño pueda ser implementado dentro de estos dispositivos.

### 7.3 Aplicación de los PLC's.

El PLC por sus especiales características de diseño tiene un campo de aplicación muy extenso. La constante evolución del hardware y software amplía continuamente este campo para poder satisfacer las necesidades que se detectan en el aspecto de sus posibilidades reales. Su utilización se da fundamentalmente en aquellas instalaciones en donde es necesario realizar procesos de maniobra, control, señalización, etc. Por tanto, su aplicación abarca desde procesos de fabricación industrial de cualquier tipo al de transformaciones industriales, control de instalaciones, etc.

### 7.4 Estructura de un PLC.

Todos los PLC's comerciales poseen una estructura externa compacta en la que están todos los elementos. Sin embargo, podemos decir que existen básicamente dos formas externas de presentación de los PLC's, una modular y la otra compacta.

En cuanto a la estructura modular existen:

- Estructura americana: separa las E/S del resto del autómata.
- Estructura europea: cada módulo es una función (fuente de alimentación, CPU, E/S, etc.).

Los micro-PLC's suelen venir sin caja, en formato kit, ya que su empleo no es determinado y se suele incluir dentro de un conjunto más grande de control o dentro de la misma maquinaria que se debe controlar.

**Sección de entradas:** se trata de líneas de entrada, las cuales pueden ser de tipo digital o analógico. En ambos casos se tienen rangos de tensión característicos, los cuales se encuentran en las hojas de características dadas por el fabricante. A estas líneas conectaremos los sensores, y las líneas de transmisión.

**Sección de salidas:** son una serie de líneas, que también pueden ser de carácter digital o analógico. A estas líneas conectaremos los actuadores. Tanto las entradas como las salidas están aisladas de la CPU según el tipo de autómatas que utilizemos. Normalmente se suelen emplear optoacopladores en las entradas y relés/optoacopladores en las salidas.

Un elemento importante es el microprocesador que forma parte del “corazón” de la CPU. La unidad central de proceso (CPU) se encarga de procesar el programa de usuario que le introduciremos. Para ello disponemos de diversas zonas de memoria, registros, e instrucciones de programa (parte superior del diagrama en bloques). Adicionalmente, en determinados modelos más avanzados, podemos disponer de funciones ya integradas en la CPU; como reguladores Proporcionales Integrales Derivativos (PID), control de posición, etc. Muchos equipos poseen una unidad de alimentación (algunas CPU la llevan incluida). También se dispone de una unidad o consola de programación que nos permitirá introducir, modificar y supervisar el programa de usuario. Los dispositivos periféricos, como nuevas unidades de E/S, más memoria, unidades de comunicación en red, etc., y las interfaces facilitan la comunicación del autómatas mediante enlace serie con otros dispositivos (como un PC).

#### 7.4.1 La memoria.

Dentro de la CPU disponemos de un área de memoria, la cual posee “varias secciones” encargadas de distintas funciones. Así tenemos:

- Memoria del programa de usuario: aquí introduciremos el programa que el PLC va a ejecutar cíclicamente.

- Memoria de la tabla de datos: es la zona encargada de atribuir las funciones específicas del programa.

Se suele subdividir en zonas según el tipo de datos (como marcas de memoria, temporizadores, contadores, etc.).

- Memoria del sistema: aquí se encuentra el programa en código de máquina que monitoriza el sistema (programa del sistema o firmware). Este programa es ejecutado directamente por el microprocesador/microcontrolador que posea el PLC.
- Memoria de almacenamiento: se trata de una memoria externa que empleamos para almacenar el programa de usuario, y en ciertos casos parte de la memoria de la tabla de datos. Suele ser de uno de los siguientes tipos: EPROM, EEPROM ó FLASH.

Cada PLC divide su memoria de esta forma genérica, haciendo subdivisiones específicas según el modelo y fabricante.

#### **7.4.2 CPU.**

La Unidad de proceso Central (CPU) es el corazón del PLC. Es la encargada de ejecutar el programa de usuario mediante el programa del sistema (es decir, el programa de usuario es interpretado por el programa del sistema). Sus funciones son vigilar que el tiempo de ejecución del programa de usuario no exceda un determinado tiempo máximo (Tiempo de ciclo máximo). A esta función se la suele denominar Watchdog (perro guardián). También se encarga de ejecutar el programa de usuario, crear una imagen de las entradas, ya que el programa de usuario no debe acceder directamente a dichas entradas.

Otra función es la de renovar el estado de las salidas en función de la imagen de las mismas obtenida al final del ciclo de ejecución del programa de usuario.

Por último, también se encarga de realizar el chequeo del sistema. Para ello el PLC posee un ciclo de trabajo, que ejecutará de forma continua el diagrama de flujo mostrado en la figura 2.

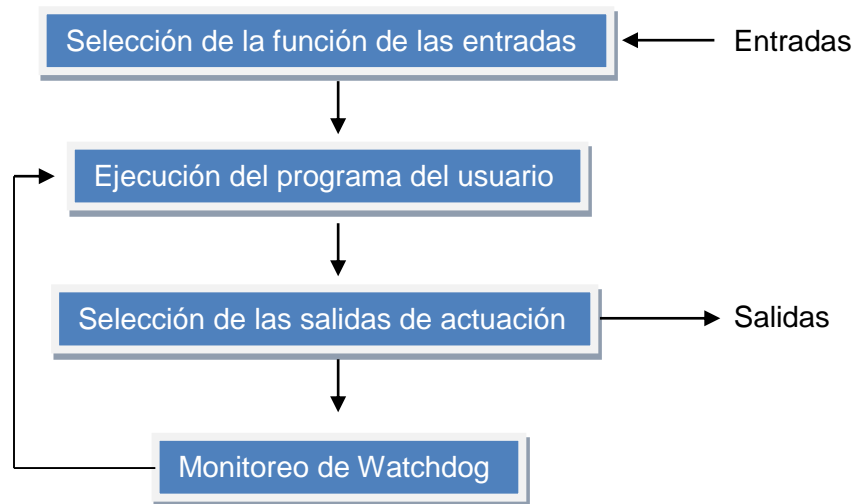


Figura 2:Ciclo de trabajo del CPU de un PLC

#### 7.4.3 Unidades de E/S.

Generalmente se dispone de dos tipos de E/S:

- **Digital.**
- **Analógica.**

Las E/S digitales se basan en el principio de todo o nada, es decir no conducen señal alguna o poseen un nivel mínimo de tensión. Estas E/S se manejan nivel de bit dentro del programa de usuario. Las E/S analógicas pueden poseer cualquier valor dentro de un rango determinado especificado por el fabricante. Se basan en convertidores A/D y D/A aislados de la CPU (ópticamente o por etapa de potencia). Estas señales se manejan a nivel de byte o palabra (8/16 bits) dentro del programa de usuario. Las E/S son leídas y escritas dependiendo del modelo y del fabricante, es decir pueden estar incluidas sus imágenes dentro del área de memoria o ser manejadas a través de instrucciones específicas de E/S.

#### **7.4.4 Interfaces.**

Todo PLC, salvo casos excepcionales, posee la virtud de poder comunicarse con otros dispositivos (como un PC). Lo normal es que posea una interface serie del tipo RS-232/RS-422. A través de esta línea se pueden manejar todas las características internas del controlador, incluida la programación del mismo, y suele emplearse para monitorización del proceso en otro lugar separado.

#### **7.4.5 Unidades de Programación.**

La programación del PLC puede ser hecha por una unidad de programación que suele ser en forma de calculadora. Es la forma más simple de programar el equipo, y se suele reservar para pequeñas modificaciones del programa o la lectura de datos en el lugar de colocación del equipo. También se puede usar una consola de programación. Es un terminal a modo de ordenador que proporciona una forma más cómoda de realizar el programa de usuario y observar parámetros internos del PLC. Desfasado actualmente.

El modo más empleado para programar un PLC es mediante una computadora tipo PC. Permite programar desde un ordenador personal estándar, con todo lo que ello supone: "herramientas más potentes, posibilidad de almacenamiento en soporte magnético, impresión, transferencia de datos, monitorización mediante software SCADA, etc."

Para cada caso el fabricante proporciona lo necesario, bien el equipo o el software/cables adecuados. Cada equipo, dependiendo del modelo y fabricante, puede poseer una conexión a uno o varios de los elementos anteriores. En el caso de los micro-plc se escoge la programación por PC o por unidad de programación integrada en la propia CPU.

#### **7.4.6 Dispositivos Periféricos**

El PLC, en la mayoría de los casos, puede ser ampliable. Las ampliaciones abarcan un gran abanico de posibilidades, que van desde las redes internas (LAN, etc.), módulos auxiliares de E/S, memoria adicional... hasta la conexión con otros autómatas del mismo modelo. Cada fabricante facilita las posibilidades de ampliación de sus modelos, los cuales pueden variar incluso entre modelos de la misma serie.

## 7.5 Programación de PLC's.

Existen diversos tipos de programación para PLC's. En la actualidad, podemos encontrar la programación tipo escalera, la programación en bloques, secuencial, texto estructurado, etc. siendo los más comunes los primeros dos tipos de procedimientos. El detalle existente para la programación de un PLC es que cada una de las empresas diseñadoras presenta un software determinado para la configuración de los procesos de su PLC propio.

Algunas de las instrucciones básicas en los lenguajes de programación mencionados están los siguientes:

- Instrucciones lógicas: AND, OR, NOT, XOR, SET, RESET.
- Instrucciones de módulos de programación: Temporizadores, contadores, registros de desplazamientos, comparadores.
- Instrucciones de control: Control de marcha, condiciones de restart, saltos condicionales.
- Instrucciones matemáticas: Suma, resta, multiplicación y división.
- Instrucciones de comparación: Comparación de bits, bytes, igualdad, mayor que, menor que.
- Instrucciones de traslación: Traslación de datos en memoria.
- Instrucciones de conversión: decimal, binario, BCD.

Se recomienda consultar las siguientes ligas (Documento en electrónico, dar click sobre el nombre de cada manual) o los manuales incluidos en el disco para mayor información sobre el lenguaje de programación en escalera y en bloques usado en los PLC's Allen Bradley modelo 5000 con el programa ControlLogix así como para el manejo de controladores Logix.

1. Diagrama de Lógica de Escalera para ControlLogix 5000. Anexo 1.
2. Logix5000 Controllers General Instructions Reference Manual. Anexo 2.
3. Logix5000 Controllers Function Block Diagram. Anexo 3.
4. Control Logix System User Manual. Anexo 4.
5. Introduction to Logix Controllers. Anexo 5.



## **8 Procedimiento y descripción de las actividades realizadas.**

---

Este proceso se realizó en función a los problemas a resolver que se presentaron en el capítulo 5 de este reporte, se llevó a cabo en forma lineal debido a que cada tarea realizada servía como base para la siguiente actividad.

Primeramente se realizó la investigación de rutinas utilizados en el último año en los diferentes proyectos que se hicieron en TERNIUM y que estuvieran relacionados con temas de control por medio de PLC. En este punto, el asesor interno sugirió una entrevista por correo con cada uno de los ingenieros del área de automatización de nivel 1 para poder conocer su punto de vista acerca del tema y que ellos, con su experiencia, aportaran cierta información que serviría para el desarrollo de las prácticas.

Con la información recabada en las entrevistas, se obtuvieron ciertos estándares que se manejaban al momento de la práctica de control de diversas maquinarias, principalmente al momento de programar dichos PLC's y así se pudo conjuntar variada información para su análisis.

Después de haber recabado la información se procedió a verificar los documentos que TERNIUM proporciona a los ingenieros como estándar de automatización para cotejar dichos archivos con los obtenidos de la entrevista y así poder generar un reporte de utilización y una comparativa, conteniendo las distintas rutinas más utilizadas por TERNIUM.

Al término del análisis, se procedió a las visitas a campo para poder verificar que la información que había sido proporcionada fuera verídica y durante ese tiempo, con ayuda del asesor, se pudieron conocer los procesos de instalación de los PLC's. También se pudo comprobar el uso de los estándares en la programación de dichos controladores, ya que se observó cómo se modificaban estos durante los días de paro de actividades de la planta y en ocasiones que se requerían como por ejemplo, días en que el PLC, marcaba un error de funcionamiento o cuando se averiaba algún componente.

Otra de las actividades extras a los objetivos del proyecto fue la puesta en línea de los PLC's conforme a la ingeniería básica (ver anexo 6 en el disco para ejemplo de

ingeniería básica manejada) que se tenía que elaborar previamente para proceder con la instalación, en dicha actividad se pudo conocer conexiones diversas como DeviceNet, Ethernet y Puerto Serial.

## **9 Resultados, planos, gráficas, prototipos y programas.**

---

Como resultado del proceso de *Residencia Profesional* y como marcaba el objetivo general de la misma, se logró elaborar el siguiente documento, que es un estándar de programación para las rutinas de control más usadas en aplicaciones diversas de la empresa TERNIUM en el Área Manager Norte correspondiente a nivel 1 y se muestra a continuación.

### **10 Estándar de programación para las rutinas de control de Nivel 1 con ControlLogix 5000**

---

El desarrollo de un estándar en el área de automatización permite generar en la práctica de la ingeniería ciertas características que hacen de dicho proceso una herramienta útil para el ingeniero que está en contacto en el desarrollo de un nuevo sistema automatizado, es este caso específico, un sistema mediante PLC's. Estas características presentan ciertas ventajas como las siguientes.

- Facilitar la interpretación de los distintos programas
- Unificar criterios de programación para simplificar las tareas de mantenimiento
- Simplificar las puestas en marcha
- Permitir reutilizar partes de los programas para otras aplicaciones
- Establecer las bases para desarrollos en sistemas de diagnóstico y alarmas de secuencia

## 10.1 Niveles de automatización y control.

El propósito de este tema es aclarar los conceptos de niveles de automatización y control adoptados por TERNIUM. Un sistema integral de automatización comprende tres niveles (0, 1 y 2) los cuales deben ser diseñados por el proveedor bajo los lineamientos abajo descritos y deben ser puestos a consideración de TERNIUM para su aprobación

### 10.1.1 Nivel 0.

Es el nivel encargado de interactuar con el proceso o planta, y comprende todos los elementos de campo. Estos elementos son los siguientes:

- Cables
- Borneras
- Sensores
- Electroválvulas
- Instrumentos / Trasmisores
- Convertidores de CC.
- Convertidores de CA.
- Elementos de maniobra (seccionadores, guardamotors, térmicos, contactores, relés, etc.)
- Motores

Es muy importante el análisis para la selección del elemento de sensado para cada punto de la máquina o proceso. Tener en cuenta, la temperatura, humedad, grado de protección, polución, grado de seguridad, método de medición, precio, recomendaciones del fabricante, experiencia de su utilización en aplicaciones iguales o similares.

**IMPORTANTE:** Todo sensor o instrumento debe tener su ingeniería de montaje y conexión de piping, eléctrica y/o red de comunicación correspondiente. (Manuales de instalación y procedimiento de desarrollo)

### 10.1.2 Nivel 1.

Este nivel es el comprendido por los PLC. Es el sistema que controla directamente todos los accionamientos de la planta y es el responsable de todas las decisiones en la operación de la planta. La planta no puede operar sin este nivel, por lo cual debe ser diseñado con criterios de máxima confiabilidad.

La programación debe desarrollarse siguiendo los estándares de TERNIUM, para facilitar la comprensión del software por parte del personal de planta, así como facilitar también las posibles mejoras que se incorporen en un futuro. Todo el software desarrollado debe ser simulado y probado antes de instalarse en planta SIN EXCEPCIÓN.

Las principales tareas que cumple son:

- Control directo de todos los sensores de campo, motores, electro válvulas, etc.
- Lazos de control de proceso
- Master de referencias (de Velocidades, Tensiones, Fuerzas, variables críticas de proceso)
- Generación de alarmas
- Seguimiento de datos tecnológicos y del proceso. (Tracking)

La cantidad de PLC a instalar dependerá de la complejidad de la instalación. Se deberá evaluar la mantenibilidad del sistema, el tiempo de revisión asociado al proceso que se deba controlar y la facilidad de diferentes equipos de trabajo en la puesta en marcha.

Se debe analizar la funcionalidad y criticidad del proceso para determinar si se utilizan PLC en hot stand by.

Si la línea o planta a instalar tiene muchos servicios auxiliares es posible que se considere un PLC auxiliar para servicios. Si la línea o planta a instalar tiene equipamiento de regulación de velocidad de motores que necesitan un sincronismo de velocidades es normal que se instale un PLC Master de Referencia para sincronizar a todos los equipos.

Para aplicaciones de control rápido se podrán colocar controladores dedicados de otras marcas para tal fin. Estos controladores deben comunicarse con el PLC y sólo deben ejecutar las funciones rápidas del control.

### **10.1.3 HMI Nivel 1.**

Para plantas o sistemas pequeños, y con el consentimiento de TERNIUM, se podrán implementar pantallas operativas con terminales HMI Magelis-Schneider.

Estos equipos estarán destinados a la operación local (pie de máquina), reemplazando tableros de mandos locales de pulsadores, por ejemplo, en centrales hidráulicas, sistemas de bombas de refrigeración, etc.

En este caso la programación de la terminal estará a cargo del proveedor de la automatización de la planta y las pantallas operativas de la terminal estarán supervisadas y deberán ser aprobadas por TERNIUM.

La conexión a los PLCs será mediante la red Ethernet ModBus.

### **10.1.4 Nivel 2.**

TERNIUM cuenta con un SCADA propio desarrollado sobre el Sistema Operativo QNX, en sus versiones 4 o 6, para realizar el diálogo entre el operador y la máquina.

Es el responsable de todas las decisiones de supervisión, pantallas de operación de proceso, alarmas, históricos, bases de datos tecnológicos, recetas (setup) de operación según producto, realimentación al operador de las variables, etc.

## **10.2 Operación de la planta sin Nivel 2**

La automatización del Nivel 1 deberá estar desarrollada de tal manera de poder operar la planta, aunque sea parcialmente, si el Nivel 2 queda fuera de operación, pero con las siguientes consignas:

- Si la operación no es posible, se debe poder llevar el proceso a una condición segura.

- Se operará si es que no se requiere un registro formal o legal de variables del producto o proceso.
- Si la operación sin el Nivel 2 no afecta a la calidad del producto y al registro de datos o producción.
- Si no se requieren efectuar cambios de *presets* del producto.

Es decir que en el caso de una interrupción en el Nivel 2, la planta o proceso deberá poder tener algún tipo de operación alternativa desde pupitres o HMI de Nivel 1.

### 10.3 Definición de Hardware Familia ControlLogix.

La siguiente es la lista de elementos de hardware que se recomienda utilizar en proyectos basados en tecnología ControlLogix.

Se pueden utilizar dispositivos diferentes de la misma familia para aquellos casos en que la aplicación así lo requiera.

a) Chasis. - ([Ver lista completa.](#))

Cat. N°	Peso Aprox.	Dimensiones Aproximadas
<b>1756-A10</b>	1.45 kg (3.2 lb)	<b>137 x 483 x 145 mm (5.4 x 19.0 x 5.8 in.)</b>
<b>1756-A13</b>	1.90 kg (4.2 lb)	<b>137 x 588 x 145 mm (5.4 x 23.2 x 5.8 in.)</b>
<b>1756-A17</b>	<b>2.20 kg (4.8 lb)</b>	<b>137 x 738 x 145 mm (5.4 x 29.1 x 5.8 in.)</b>

b) Fuentes - ([Ver lista completa](#))

Cat. N°	Tensión Entrada	Rango de operación	Potencia Máxima
<b>1756-PB72/C</b>	<b>24V dc</b>	<b>18...32V dc</b>	<b>95 W</b>

c) Controladores - ([Ver lista completa](#))

Cat. N°	Memoria usuario	Memoria I/O	Memoria No Volátil
<b>1756-L63</b>	<b>8196 K bytes</b>	<b>478 K bytes</b>	<b>64 MbytesCompactFlash</b>

d) Módulos de entradas digitales. ([Ver lista completa](#))

Cat. N°	N° Entradas	Voltaje Nominal	Voltaje operativo
<b>1756-IA16I</b>	16 Aisladas Individualmente	120V ac	<b>79...132V ac</b>
<b>1756-IA32</b>	32	120V ac	<b>74...132V ac</b>
<b>1756-IB16I</b>	16 Aisladas Individualmente	12/24V dc	<b>10...30V dc</b>
<b>1756-IB32</b>	<b>32</b>	<b>12/24V dc</b>	<b>10...31.2V dc</b>

e) Módulos de Salidas digitales. - ([Ver lista completa](#))

Cat. N°	N° Salidas	Volt Nominal	Voltaje operativo
<b>1756-OA16I</b>	16 Aisladas Individualmente	120/240V ac	<b>74...265V ac</b>
<b>1756-OB32</b>	<b>32</b>	<b>12/24V dc</b>	<b>10...31.2V dc</b>

f) Módulos de entradas Analógicas - ([Ver lista completa](#))

Cat. N°	N° Entradas	Resolución	<u>Sensores Soportados</u>
<b>1756-IF16</b>	8 Dif., 4 alta vel, 16 single-ended.	16 bits	<b>±10.25V/0...5.125V/0...10.25V</b>
<b>1756-IR6I</b>	6 RTD Aisladas	16 bits	<b>Ver el Link</b>
<b>1756-IT6I</b>	<b>6 entradas termocupla Aisladas</b>	<b>16 bits</b>	<b>Ver el Link</b>

g) Módulos de salidas Analógicas - ([Ver lista completa](#))

Cat. N°	N° Salidas	Resolución	<u>Sensores Soportados</u>
<b>1756-OF8</b>	<b>8 tensión o Corriente</b>	<b>15 bits</b>	

h) Módulos de Comunicación - ([Ver lista completa](#))

Cat. N°	Tasa	Características
1756-ENBT	10/100 Mbps	<b>Cada módulo soporta máximo de 128 conexiones TCP/IP</b>
1756-CNBR	5 Mbps	<b>Módulo de comunicación Control Net Redundante</b>
1756-DNB	125/250/500Kbps	<b>Módulo de comunicación Devicenet</b>

i) Módulos varios - ([Buscar otros dispositivos](#))

Cat. N°	Características
1757-SRM	<b>Modulo para sistema redundante. Se necesita 1 por Chasis</b>
1788-CN2DN	<b>Dispositivo vínculo ControlNet a Devicenet</b>

## 10.4 Configuración: Nombres de dispositivos

### 10.4.1 Nombre del proyecto

El nombre del proyecto hará referencia al sistema que le corresponde controlar. No hay restricciones referido a la cantidad de caracteres que debe poseer, pero es aconsejable que no sea demasiado grande, no más de diez caracteres. Cuando la envergadura del proyecto lo requiera, es aconsejable agregar un prefijo que indique a que proyecto está referido.

Por ejemplo para el caso de la Colada Continua 2, todos los proyectos deberán empezar con "CCD2\_".

### 10.4.2 Nombre del archivo.

Este corresponde al nombre del archivo que se guardará en la PC y en general será igual al nombre del proyecto.



### 10.4.3 Nombre del Controlador.

Este corresponde al nombre del controlador y deberá ser denominado con una de letra.

Por ejemplo: A, B, C.....M

Ejemplos de nombres de un controlador:

CCD2\_A\_Linea1, CCD2\_B\_Linea2, CCD2\_C\_Marcadora1, CCD2\_D\_Marcadora2

### 10.4.4 Nombre de chasis.

Antes de describir como es el nombre de cada módulo, se define que cada chasis deberá tener un nombre o identificador.

El Chasis donde se encuentra la CPU se lo identificará como 00. En caso de que exista una redundancia el chasis primario será 0A y el secundario se lo identificara como 0B.

Los chasis remotos, como los de entradas y salidas, se identificarán con números que van, para el primer chasis remoto, desde 01 en adelante.

### 10.4.5 Nombre de los DROPS.

Para el DROP donde está la CPU, el número será el 00, identificado también con la letra del PLC.

Recordar, que el ID de la CNBR del DROP donde se encuentra la CPU, es el valor más alto de los nodos scheduled de la red.

La nomenclatura del DROP se forma de acuerdo a la siguiente configuración.

A000A

A	00	0A	Dígitos
		En Drops Locales – 00 o 0A y 0B para sistemas redundantes	2
		En Drops Remotos -Número de ID de la CNBR del DROP.	2
		Posición de cabeceras de red a la que esta asociado el DROP. (00 si es LOCAL)	2
		Nombre de PLC	1

Ejemplo 1: D000A → DROP Local Primario de PLC D

Ejemplo 2: A000B → DROP Local redundante de PLC A

Ejemplo 3: F0304 → DROP Remoto 4 conectado a CNBR en Slot 3 del DROP local del PLC F

#### 10.4.6 Numero de nodo en DROPS donde se encuentran las CPU's.

El número de nodo para el DROP XXXXA será el número ID de la CNBR

El número de nodo para el DROP XXXXB será el número ID de la CNBR + 1

#### 10.4.7 Nombre de los Taps.

La nomenclatura de los Taps se forma según la siguiente configuración:

TAP A0315B

TAP A0315B						Dígitos
A	03	15	B	A o B dependiendo del canal		1
			Número de Nodo ControlNet del Drop			2
			Número del slot de la CNBR del drop 0			2
Nombre de PLC						1

#### 10.4.8 Nombre de los Coaxiales de conexión entre Taps.

La nomenclatura de los coaxiales se forma según la siguiente configuración:

Cx A02 07 06B

Cx A02 07 06B						Dígitos	
Cx	A	02	07	06	B	A o B dependiendo del canal	2
			Número de Nodo al que entra			2	
			Numero de Nodo del que Sale			2	
			Posición de cabeceras de red en el rack de la CPU			2	
Nombre de PLC						1	
Abreviatura de Coaxial						N	

#### 10.4.9 Nombre de módulos de comunicación.

Los módulos de comunicación que se encuentran en el chasis cero, donde se encuentra el controlador, serán llamados de la siguiente forma:

CNBR\_A000A02

CNBR_	A	00	0A	0 2	Dígitos
Número de slot donde se encuentra el módulo en el Chasis					2
Nombre del chasis al que está asociada esta tarjeta.					2
Posición de cabeceras de red en el rack de la CPU					2
Nombre de PLC					1
Tipo de Módulo (Ej. del caso Módulo ControlNet Redundante 1756-CNBR)					N

Y si existe chasis redundancia, el módulo en éste será entonces:

CNBR\_A000B02

De esta manera si se tiene una tarjeta de Ethernet ésta se llamara:

ENBT\_A000A04

Al definir una red *ControlNet* para remote I/O, debajo de ésta habrá que agregar las tarjetas de ControlNet de cada chasis. Estas se denominarán como sigue:

CNBR\_A020100

CNBR_	A	02	01	0 0	Dígitos
Número de slot donde se encuentra el módulo en el Chasis					2
Nombre del chasis al que está asociada esta tarjeta.					2
Posición de cabeceras de red en el rack de la CPU.					2
Nombre de PLC					1
Tipo de Módulo (Ej. del caso Módulo ControlNet Redundante 1756-CNBR)					N

Nota1: En este tipo de red, el módulo que se encuentra en el chasis cero se denomina cabecera. Su número de nodo será el último de la red. Dejando reservas, este nodo puede ser el número 20. En caso de existir redundancia, se tiene que dejar reservado el número de nodo inmediato superior a éste, pues éste número será asignado a la cabecera ControlNet del secundario.

Nota 2: el número que identifica a los chasis remotos se asigna según el número de nodo de la tarjeta ControlNet que se coloque en él. De esta forma el primer número de nodo es el 1 por lo que el chasis será el 1.

#### 10.4.10 Nombre de módulos de entrada / salida.

El nombre que llevarán los módulos de entrada salida será dado por:

OB16D-A020305

OB16D_	A	02	03	05		Dígitos
					Número de slot donde se encuentra el módulo en el Chasis	2
					Nombre del chasis al que está asociada esta tarjeta.	2
					Posición de cabeceras de red en el rack de la CPU.	2
					Nombre de PLC	1
					Tipo de Módulo (Ej. del caso Módulo 1756-OB16D)	N

Esto se interpreta como que el módulo OB16D se encuentra en el slot 5 del chasis remoto llamado 03 perteneciente a una red I/O cuya cabecera se encuentra en el slot 2 del chasis donde se encuentra el controlador cuyo nombre es A.

En caso de tener módulos dentro del chasis donde está el controlador, (solamente en un sistema no redundante) será por ejemplo: OB16D\_ A00A06, que indica que la tarjeta está colocada en el slot 06 del rack 00.

#### 10.4.11 Identificación del cableado a módulos de entrada / salida.

Para facilitar la lectura y el entendimiento de planos, la nomenclatura asignada al cableado a módulos de IO sigue la misma línea que la nomenclatura utilizada para identificar los módulos agregando solamente el número de canal al cual corresponde el cableado.

A02030512

A	02	03	05	12	Dígitos
Número de Canal del Módulo IO					2
Número de slot donde se encuentra el módulo en el Chasis					2
Nombre del chasis al que está asociada esta tarjeta.					2
Posición de cabeceras de red.					2
Nombre de PLC					1

#### 10.4.12 Archivos de configuración de ControlNet y DeviceNet.

Al configurar redes ControlNet o DeviceNet, es necesario guardar el archivo de configuración de la red utilizado en el software RSNetWorx, para luego ingresarlos dentro de la configuración de los módulos correspondientes.

El nombre de estos archivos se define de la siguiente manera:

AH2\_CNT\_A0001.XC

AH2	CNT	A0001	.XC	Dígitos
Extensión				2
Nombre del Módulo Cabecera de Red / CCM / Red / etc.				5
Tipo de Red: CNT= CONTOLNET; DNT= DEVICENET				3
Nombre del proyecto				N

### 10.5 Tags: nomenclatura de instrumentos y señales

#### 10.5.1 Conformación de Tags

Los tags son los elementos de identificación que se representa en la ingeniería básica de todo una instalación. Los tags a utilizar en el listado de motores y sensores, se deberán formar utilizando el siguiente esquema:

2	0	1	-	P	D	S	H	-	3	7	0	B
Planta	Sistema			Instrumento					Lazo		Sufijo	

Los números de Planta y sistema, así como la nomenclatura de los instrumentos se obtienen de las tablas estándar de nomenclaturas de procesos e instrumentos para cada sector.

Ej.: 2456-R-DP-251071 – ACERIA/COLADA CONTÍNUA Y DESBASTES – WATER SYSTEM

### 10.5.2 Conformación de Tags programación PLC

Las variables o tag base asociadas a las señales de IO del PLC se armarán de la siguiente forma

Ej.: Tag PID - 103LT050 – Nivel de cisterna de circuito cerrado (Ver tabla 1 y 2)

#### **sistema\_equipo\_variable**

sistema: Circuito cerrado, por ejemplo: CCerrado

equipo: Cisterna - EjCist

Variable: es el texto que identifica la variable, instrumento proceso o acción que llega o sale al PLC: Nivel de cisterna. EJ: Nivel

Ejemplo de tag PLC:

Nombre de variable

**Ccerrado\_ Cist\_Nivel**

En el campo “comentario” de la variable se debe completar con el texto que va aparecer en la aplicación de infotag, y describe la función de la variable que sirve para su interpretación, por lo tanto tendrá el mismo formato.

En la descripción se deberá agregar entre paréntesis el tag de PID o Plano de proceso vinculado al tag de PLC.

Esto se hace para poder tener en PLC un Tag con un nombre descriptivo que facilite la interpretación del programa y en la descripción una referencia a Tag de PID que permita relacionar el tag de PLC con el Tag en Planos.

De esta forma exportando la tabla de variables de plc y filtrando se obtiene fácilmente una tabla de referencias cruzadas entre Tag-PLC y Tag-PID

Del ejemplo anterior:

Comentario

**Nivel de Cisterna de Circuito cerrado (103-LT-050)**

Tag Completo

**Ccerrado\_ Cist\_Nivel**

Nivel de Cisterna de Circuito cerrado (103-LT-050)

	Primera Letra		Letras Sucesivas		
	Variable	Modificador	Función de lectura pasiva	Función de salida	Modificador
<b>A</b>	Análisis		Alarma		Accionado / Abierto
<b>B</b>	Llama / Quemador				Bajar
<b>C</b>	Conductividad			Control/Regulación	No Accionado / Cerrado
<b>D</b>	Densidad	Diferencial			
<b>E</b>	Tensión/Electro	Emergencia	Elemento primario		
<b>F</b>	Caudal	Frecuencia			Falla
<b>G</b>	Calibre		Vidrio / Visualización		
<b>H</b>	Manual				Alto
<b>I</b>	Corriente		Indicador o indicación		Entrada
<b>J</b>	Potencia				Jogues
<b>K</b>	Tiempo			Estación de Control	
<b>L</b>	Nivel/Límite		Luz / Lámpara		Bajo
<b>M</b>	Humedad		Motor		Medio
<b>N</b>					
<b>O</b>		Nulo/Cero	Orificio / Restricción		Salida
<b>P</b>	Presión / Vacío		Punto de Prueba		
<b>Q</b>	Cantidad				
<b>R</b>	Radiación	Referencia	Registro		Reserva/Reversa
<b>S</b>	Velocidad / Frecuencia	Seguridad		Switch/Estado	
<b>T</b>	Temperatura/Torque			Trasmisor	
<b>U</b>	Multivariable		Multifunción	Multifunción	Multifunción
<b>V</b>	Vibración			Válvula	
<b>W</b>	Peso / Fuerza		Vaina		
<b>X</b>		Eje X	Genérico		Genérico
<b>Y</b>	Estado/Indicación	Eje Y		Relé/Salida Digital	
<b>Z</b>	Posición/Límite	Eje Z		Elemento final de control sin clasificar	

Tabla 1: Variables que identifican la acción o el instrumento de proceso que sale del PLC



Señales Eléctricas			
GM	Guardamotor	Q	Interruptor dispositivo general
JB	Caja de conexiones	PR	Pulsador de Reserva Local
LR	Selectora Local/Remoto	PX	Pulsador Genérico
MA	Orden de Marcha/Abrir/Avance	RA	Confirmación de marcha
MC	Orden de Parada / Cerrar	RC	Confirmación cierre general
OL	Sobrecarga Dispositivo	RE	Confirmación Relé
OV	Cero Velocidad Dispositivo	RR	Confirmación Contramarcha
PA	Pulsador Arranque	RS	Confirmación Listo
PE	Pulsador Parada Emergencia	VF	Vibro alimentador
PP	Pulsador Parada	VDS	Variador de velocidad para motores

Tabla 2: Variables compuestas de señales eléctricas para Tag's en el PLC.

## 10.6 Programación.

El software de desarrollo de Nivel 1 para programación de los PLC's es el control Logix Versión 17 o superior (Ver marco teórico y manuales anexos).

Los lenguajes de programación utilizados en TERNIUM son FBD (Function Block Diagram) y LadderLogic. No utilizar los lenguajes SFC (Sequential Function Chart), Lista de instrucciones y texto estructurado, salvo que la aplicación estrictamente lo requiera por razones técnicas (ej.: programación de funciones, rutinas especiales, etc.). En este caso debe tener la autorización de TERNIUM.

En ControlLogix el programa se organiza en secciones y módulos funcionales. Todo el programa debe estar ordenado por sistemas colocando para cada uno un módulo funcional incluyendo sus correspondientes secciones.

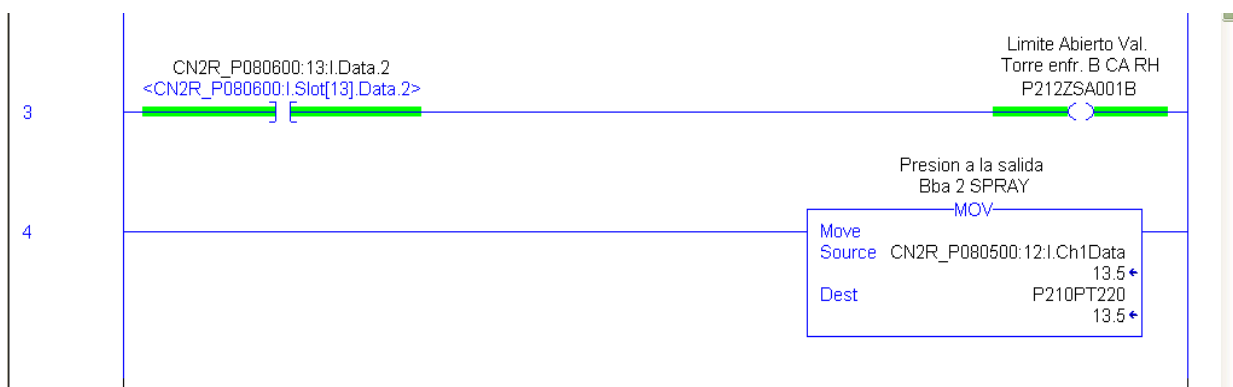
Existen secciones fijas que deben estar incluidas en el programa:

1. Constantes: En esta sección estarán todas las constantes usadas en el software. Tales como ancho de bobina, distancias fijas, velocidades de referencia fijas, etc.
2. Configuración de entradas: En esta sección se preparan todas las entradas analógicas que se usarán en el programa. Esta preparación se refiere a Escalamientos, filtros, operaciones matemáticas, bloque EHC, etc.
3. Alarmas: Generación de alarmas.

### 10.6.1 Tags I/O

Todas las entradas de campo, tanto digitales como analógicas, deberán estar mapeadas en el programa en un tag interno. De esta manera, en caso de una emergencia, se puede cambiar el origen del tag con una simple modificación en el programa.

En la figura 3 se muestra el mapeo a tags internos de una entrada digital y una analógica.



M577										
f = CONCATENAR(B577;I577;J577;K577;L577)										
	H	I	J	K	L	M	N	O	P	
1	CCM / RT	Cab. Red	Rack	Slot	Canal	Id.Canal	Tag	Descripción	I/O	
563	RIO42	08	06	13	01	P08061301	P212ZSC001A	Limite Cerrado Val. Torre enfr. A CA RH	DI	
564	RIO42	08	06	13	02	P08061302	P212ZSA001B	Limite Abierto Val. Torre enfr. B CA RH	DI	
565	RIO42	08	06	13	03	P08061303	P212ZSC001B	Limite Cerrado Val. Torre enfr. B CA RH	DI	
566	RIO42	08	06	13	04	P08061304	P212ZSA001A	Valv. Abierta Rensiccion. Sistema CA RH	DI	

N247										
f =										
	H	I	J	K	L	M	N	O	P	
1	CCM / RT	Cab. Red	Rack	Slot	Canal	Id.Canal	Tag	Descripción	I/O	
230										
231	RIO42	08	05	12	00	P08051200	P210PT120	Presion a la salida Bba 1 SPRAY	AI	
232	RIO42	08	05	12	01	P08051201	P210PT220	Presion a la salida Bba 2 SPRAY	AI	
233	RIO42	08	05	12	02	P08051202	P210PT320	Presion a la salida Bba 3 SPRAY	AI	
234	RIO42	08	05	12	03	P08051203	P210PT420	Presion a la salida Bba 1 CA	AI	
235	RIO42	08	05	12	04	P08051204	P210PT520	Presion a la salida Bba 2 CA	AI	

Figura 3: Ejemplo de mapeo de tags internos de una entrada digital y una analógica.

### 10.6.2 Tags y tipo de datos para comunicación con nivel 2

Nivel 2 puede leer y escribir datos booleanos, y reales. Para ello estas variables (tags) deben ser alias de tags base. Estos tags base deben ser arreglos de 120 elementos, doble enteros o reales.

Para ser reconocidos por el driver de nivel 2, los tags deben seguir una convención de nombres. Esto permite que el archivo generado por la exportación de tags, sea procesado en forma automática por un software de nivel 2 para generar los archivos de configuración que necesita el driver correspondiente.

El nombre del archivo de exportación debe estar formado por un prefijo indicativo del nombre de la línea, y el nombre del procesador: Linea\_NombreProcesador.csv

Por ejemplo: CCD2\_B.csv. El cual indica que es un archivo de tags de CCD2 correspondiente al PLC B.

### **10.6.3 Lectura y escritura de tags desde nivel 2**

Para poder leer o escribir variables de nivel 2, éstas deben estar asociadas (Alias) a tags base en forma de arreglos que deben cumplir el siguiente formato:

Tags de Lectura booleanos:

Arreglo "N2L\_Estados" de 120 posiciones tipo DINT

Tags de Escritura booleanos:

El nombre del tag debe comenzar con la cadena "N2E\_Comandos" de 120 posiciones tipo DINT

Tags de Lectura Analógicos:

Arreglo "N2L\_Registros" de 120 posiciones tipo REAL

Tags de Lectura / Escritura Analógicos:

Arreglo "N2LE\_Registros" de 120 Posiciones tipo REAL

Tags de Alarmas:

Arreglo "N2L\_Alarmas" de 120 Posiciones tipo DINT

Si los tags no son declarados como alias de los arreglos leídos por nivel 2, no serán leídos o escritos por nivel 2 y son solo de uso interno.

En caso de completarse un arreglo deberá crearse otro de 120 posiciones agregando un número consecutivo que determina la cantidad de agregados

EJ

Tags de Alarmas:

Arreglo "N2L\_Alarmas1" de 120 Posiciones tipo DINT

#### 10.6.4 Definición del nombre de variables de Estados

Las variables asociadas a estados que son generadas en el PLC o N1 y leídas desde N2 se denominarán de la siguiente manera:

##### **sistema\_equipo\_estado**

- Sistema: es el más general, por ejemplo: CONV1, por Convertidor 1
- Equipo: es el más particular y haciendo referencia al ejemplo anterior puede ser: MNorteSup por Motor Norte Superior
- Estado: es el texto que identifica el estado correspondiente, por ejemplo: en marcha o arriba.

Ejemplo:

Nombre de variable

**CONV1\_MNorteSup\_en\_marcha**

En el campo "comentario" de la variable se debe completar con el texto que va aparecer en la aplicación de infotag, y describe la función de la variable que sirve para su interpretación, por lo tanto tendrá el mismo formato.

Del ejemplo anterior:

Comentario

**Convertidor 1- Motor Norte Superior en marcha**

Para poder ser leído por Nivel 2 este tag debe hacerse alias de un bit de la matriz "N2L\_Estados". (Ver figura 4)

Ej: Alias de el bit 20 de N2L\_Estados[9]

Name	Alias For	Data Type	Description
M767_2A_OK	N2L_Estados[9].2	BOOL	Motor Vibro 1 Tolva2 OK
CONV1_MNorteSup_en_marcha	N2L_Estados[9].20	BOOL	Convertidor 1: Motor Norte Superior 1 en marcha
Vibros_Tolva1_Remoto	N2L_Estados[9].21	BOOL	Vibros Tolva 1 en modo Remoto

Figura 4: Ejemplo de Nombre, alias y descripción en el Software ControlLogix 5000.

### 10.6.5 Definición del nombre de variables de Alarmas

Las variables asociadas a alarmas se denominarán de la siguiente manera:

#### **ALM\_sistema\_equipo\_alarma**

- ALM: escribir esta palabra textual, sirve para identificar que la variable se trata de una alarma y sirve para filtrar en la tabla de declaración de variables.
- Sistema: es el más general, por ejemplo: CONV1, por Convertidor 1
- Equipo: es el más particular y haciendo referencia al ejemplo anterior puede ser: MNorteSup por Motor Norte Superior
- Alarma: es el texto que identifica la alarma, continuado con el ejemplo: alta\_Corr (alta corriente).

Ejemplo completo:

Nombre de variable

**ALM\_CONV1\_MNorteSup\_Alta\_Corr**

En el campo “comentario” de la variable se debe completar el texto que va aparecer en la pantalla al operador, por lo tanto tendrá el mismo formato.

Del ejemplo anterior:

Comentario

**Convertidor 1- Motor Norte Superior: Alta Corriente**

### 10.6.6 Definición de nombres de Comandos desde el N2 o N1 al PLC

Las variables asociadas a comandos que se realizan desde el N2 ó N1 al PLC se denominarán de la siguiente manera:

#### **DN2E\_sistema\_equipo\_comando**

DN2E: escribir esta palabra textual, sirve para identificar que la variable se trata de un comando Digital Escrito desde Nivel 2 al PLC.

#### **DN1E\_sistema\_equipo\_comando**

DN1E: escribir esta palabra textual, sirve para identificar que la variable se trata de un comando Digital Escrito desde Nivel 1 ó HMI (EjPanelView) al PLC.

- Sistema: es el que más abarca, por ejemplo: CONV1, por Convertidor 1
- Equipo: es el más particular y haciendo referencia al ejemplo anterior puede ser: MNorteSup por Motor Norte Superior.
- Comando: es el texto que identifica la acción u orden del comando, por ejemplo: pulsador marcha o pulsador subir rodillo.

Ejemplos:

**DN2E\_CONV1\_MNorteSup\_PB\_marcha**

**DN1E\_CONV1\_MNorteSup\_PB\_marcha**

En el campo “comentario” de la variable se debe completar el texto que va aparecer en la aplicación de login de comandos y en la de infotag, por lo tanto tendrá el mismo formato.

Del ejemplo anterior:

**Convertidor 1 – Motor Norte Superior: pulsador marcha**

Para completar con toda la información relacionada para la configuración de un comando que se escribe desde Nivel 2 al PLC, en el texto del comentario se debe agregar entre corchetes [ ] el siguiente argumento:

[T:P] Tipo pulso

[T:N] Tipo Nivel

[T:J] Tipo Jog

Ejemplo completo:

Nombre de variable:

**DN2E\_CONV1\_MNorteSup\_PB\_marcha**

Comentario de la variable:

**Convertidor 1 – Motor Norte Superior : pulsador marcha[T:P]**

Estas variables deberán ser asociadas mediante alias a un bit de la matriz: "N2E\_Comandos".

#### 10.6.7 Definición de variables de Lectura desde N2 o N1

Las variables asociadas a lectura de variables analógicas desde Nivel 2 a Nivel 1 o PLC se denominarán de la siguiente manera:

##### **AN2L\_sistema\_equipo\_nombre**

AN2L: escribir esta palabra textual, sirve para identificar que la variable es del tipo Analógica y el Nivel 2 la lee desde el Nivel 1 o PLC.

##### **AN1L\_sistema\_equipo\_nombre**

AN1L: escribir esta palabra textual, sirve para identificar que la variable es del tipo Analógica y el Nivel 1 o HMI (ej: PanelView) la lee del Nivel 1 o PLC.

- Sistema: es el que más abarca, por ejemplo: CONV1 , por Convertidor 1
- Equipo: es el más particular y haciendo referencia al ejemplo anterior puede ser: MNorteSup por Motor Norte Superior.
- Nombre: es el texto que identifica la variable de proceso, por ejemplo: corriente del motor.

Ejemplo:

Nombre de variable:

**AN2L\_CONV1\_MNorteSup\_Corriente**

**AN1L\_CONV1\_MNorteSup\_Corriente**

En el campo “comentario” de la variable se debe completar con el texto que va aparecer en la aplicación de trending y en la de infotag, por lo tanto tendrá el mismo formato.

Del ejemplo anterior:

Comentario

**Convertidor 1 – Motor Norte Superior : Corriente del Motor**

Para completar con toda la información relacionada para la configuración de una variable analógica de entrada, en el texto del comentario se debe agregar entre corchetes [ ] los siguientes argumentos:

**[U:unidad m:mínimo M:máximo e:escalado],**

- Unidad: unidad de ingeniería (Ej: bar, amp, etc.)
- Mínimo y máximo: equivalen al valor mínimo y máximo que registrará la variable en la unidad de ingeniería anterior. Este rango servirá para escalar cada variable en la aplicación de visualización de trending.
- Escalado: es el valor que multiplica a la variable en el nivel 2 para ser mostrada en pantalla o en el trending. Por ejemplo: 0.1, ó 1.

Ejemplos completos:

Nombre de variable:

**AN2L\_CONV1\_MNorteSup\_Corriente**

Comentarios:

**Convertidor 1 – Motor Norte Superior : Corriente del Motor [U:amp m:0 M:500 e:1]**

Estas variables deberán ser asociadas mediante alias a un registro de la matriz: “N2L\_Registros”.



### 10.6.8 Definición de variables de Escritura desde N2 o N1

Las variables asociadas a escrituras analógicas desde N2 al N1 o PLC se denominarán de la siguiente manera:

#### **AN2E\_sistema\_equipo\_nombre**

AN2E: escribir esta palabra textual, sirve para identificar que la variable es del tipo Analógica y el Nivel 2 la Escribe al Nivel 1 o PLC.

#### **AN1E\_sistema\_equipo\_nombre**

AN1E: escribir esta palabra textual, sirve para identificar que la variable es del tipo Analógica y el Nivel 1 o HMI (PanelView) la Escribe al Nivel 1 o PLC.

- Sistema: es el que más abarca, por ejemplo: CONV1 , por Convertidor 1
- Equipo: es el más particular y haciendo referencia al ejemplo anterior puede ser: MNorteSup por Motor Norte Superior.
- Nombre: es el texto que identifica la variable de proceso, por ejemplo: set point de corriente.

Ejemplos:

Nombre de variable:

**AN2E\_CONV1\_MNorteSup\_SP\_Corr**

**AN1E\_CONV1\_MNorteSup\_SP\_Corr**

Al igual que en la lectura de variables analógicas, para completar con toda la información relacionada para la configuración de una variable analógica de escritura al PLC, en el texto del comentario se debe agregar entre corchetes [ ] los mismo argumentos que para la lectura.

**[U:unidad m:mínimo M:máximo e:escalado],**

Ejemplo completo:

Nombre de variable:

**AN2E\_CONV1\_MNorteSup\_SP\_Corr**

Comentario:

### Convertidor 1 – Motor Norte Superior : Set Point Corriente del Motor [U:amp m:0 M:500 e:1]

Estas variables deberán ser asociadas mediante alias a un registro de la matriz: "N2LE\_Registros".

## 10.7 Fallas.

### 10.7.1 Falla de módulo.

A cada uno de los módulos que se encuentren en la configuración del controlador habrá que chequearlos de forma de sacar alguna alarma para verificar si este dispositivo está fallando.

Con la instrucción GSV -GetSystemValue (No disponible en Bloques) se puede obtener el código de falla de un módulo. Cuando el valor de este registro es cero indica que el módulo no posee falla alguna. Un valor distinto de cero indica que el módulo ha fallado. Por lo tanto para determinar la falla del módulo se verifica que el valor de este registro sea distinto de cero.

Dado que la Instrucción GSV abre un punto de comunicación ControlNet, deben encuestarse los módulos de a uno secuencialmente mediante una rutina Periódica para evitar saturar la Red.

Una Opción de programación sería la siguiente: (Ver figura 5)

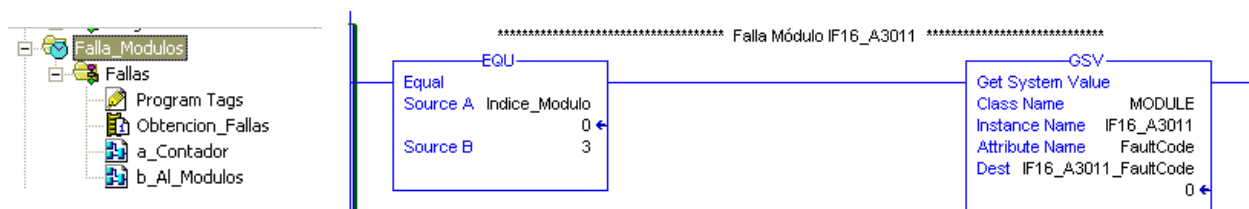


Figura 5: Ejemplo de programación de una alarma GSV.

Esta lógica se repetiría para cada módulo a encuestar. Con esta rutina temporizada, cada vez que se ejecuta, se incrementa un puntero (Indice\_Modulo) y éste indicará a que módulo se encuesta. Una vez encuestado, tiene que esperar su turno para volver a ser encuestado.

Luego con la palabra obtenida, comparándola con cero, se genera la alarma de falla de módulo.

### 10.8 Organización de programas y rutinas.

Dentro de la tarea continua se agregarán todos los programas que se necesiten. El nombre de cada programa deberá representar al equipo, o al grupo de equipos que se van a controlar, o la acción que se va a tomar, etc.

Dentro de cada programa están todas las rutinas que manejan al equipo. Siempre habrá una de ella que se define como principal y a partir de ésta se llaman a las demás para su ejecución.

El nombre de cada rutina empezará con un ante fijo, una letra desde la “A\_” en adelante o con una letra y número (A01\_). Esto se hace para que visualmente las rutinas queden ordenadas según como se van a ejecutar, pues el Logix5000, para mostrar las rutinas, las ordena en forma alfabética. El orden de ejecución de las rutinas se define en la rutina principal, según se las vaya llamando. La rutina principal (debe programarse en ladder) no llevará ningún ante fijo ya que ésta, por ser definida como principal, queda siempre en primer lugar. (Ver figura 6)

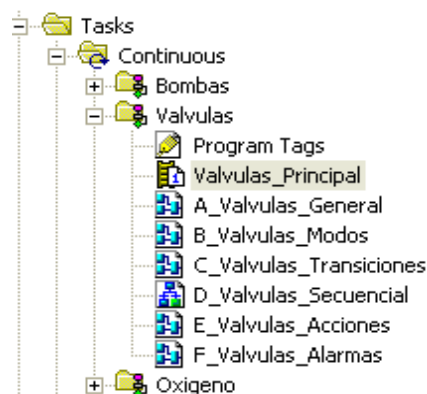


Figura 6: Forma de organización de los programas en ControlLogix.

Siguiendo con el nombre de las subrutinas, luego del ante fijo, se pondrá el nombre del programa y después se agregará un descriptivo.

A excepción de la rutina principal (ladder) y las rutinas de secuenciadores (SFC) se aconseja que las la mayoría de las rutinas sean programadas en lenguaje de Bloques de función. La utilización de otros lenguajes como ladder o texto estructurado queda determinada por los requerimientos de la programación (funciones especiales que solamente existen en lenguaje ladder, o lógica compleja o repetitiva adecuada para ser realizada en texto estructurado).

## 11 Conclusiones y recomendaciones.

---

En el transcurso de las prácticas profesionales se fueron llevando a cabo los objetivos que se plantearon al inicio del mismo proceso, para poder tener en cuenta lo que se quería aprender y hasta donde ese conocimiento se podía llegar a profundizar tomando en cuenta todas ventajas y limitaciones que englobaban. Sin duda alguna concluir satisfactoriamente este ciclo ha aportado grandes conocimientos a mi persona en el ámbito ingenieril y más específico en el área de automatización.

Se pudieron definir muchas cosas durante el proceso de desarrollo de mi residencia, ya que pude conocer diversas formas de comunicación entre un PLC y el control de diversas máquinas y llegué a comprender que a pesar de que existan determinados pasos para cada sistema, un PLC te da la flexibilidad de ir mejorando el proceso mientras este seguía en funcionamiento, ya que con técnicas, como el muy mencionado *sistema redundante*, se permitía realizar ese proceso, sin parar un proceso entero que representaría pérdidas millonarias para la empresa

Otro aspecto importante que pude entender trata acerca de la comunicación. Existen protocolos para poder comunicar todos los PLC's de la planta entre sí, sin importar que modelo sean e inclusive que sean de diferentes marcas ya que todo depende de que los protocolos sean aplicados de forma correcta. Lo primero que se conoció fueron las diversas formas de comunicación entre los PLC's, como son el Ethernet, DeviceNet y ControlNet con diferentes protocolos de comunicación, como Modbus y ProfiBus.

Ahora, con respecto al trabajo industrial en el área de automatización, pude comprender que todo se basa en un estándar para homogeneizar la ingeniería y así, en palabras coloquiales, hablar el mismo idioma, sin importar que un ingeniero venga de otra planta de la misma empresa, y en casos que no sea de nuevo ingreso, pueda comprenderlo con simples conocimientos en el área y leyendo los manuales de la empresa.

Todo esto que se mencionó anteriormente no puede ser llevado a cabo sin la correcta difusión del mismo en la empresa y a cada uno de los departamentos que tengan que ver con automatización en las diversas plantas de TERNIUM en México. Es por esto que cada final de mes existen actualizaciones de procesos de programación así como una retroalimentación de los usuarios de este estándar para notificarlos con el jefe de área y este a su vez reunirse con el comité de estandarización de nivel 1 y llevar a cabo las modificaciones y estos sean notificados nuevamente a los usuarios de dicho estándar.

Finiquitando todo lo mencionado anterior, puedo decir que durante el desarrollo del tema de mis prácticas profesionales comprendí que sin un estándar sería complicado comprender el trabajo que otros ingenieros de la misma planta y de la misma área estuvieran realizando, porque cada uno desarrollaría su trabajo con la nomenclatura que más le agradara y la programación que más fácil manejara, haciendo que trabajar con dichos PLC's fuera una tarea titánica para la mayoría y muchas veces se podría generar fallas pequeñas o graves, las cuáles podrías hacer que la empresa perdiera la base de su funcionamiento, es decir, el dinero.

## 12 Glosario.

---

- Estandarización: es la redacción y aprobación de normas que se establecen para garantizar el acoplamiento de elementos contruidos independientemente, así como garantizar el repuesto en caso de ser necesario, garantizar la calidad de los elementos fabricados, la seguridad de funcionamiento y trabajar con responsabilidad social.

- **Asíncrona:** hace referencia al suceso que no tiene lugar en total correspondencia temporal con otro suceso.
- **Optoacoplador:** es un dispositivo de emisión y recepción que funciona como un interruptor activado mediante la luz emitida por un LED que satura un componente optoelectrónico,
- **SCADA:** (acrónimo de **S**upervisory **C**ontrol **A**nd **D**ata **A**cquisition - Supervisión, Control y Adquisición de Datos) es un software para ordenadores que permite controlar y supervisar procesos industriales a distancia. Facilita retroalimentación en tiempo real con los dispositivos de campo (sensores y actuadores) y controlando el proceso automáticamente. Provee de toda la información que se genera en el proceso productivo (supervisión, control calidad, control de producción, almacenamiento de datos, etc.) y permite su gestión e intervención.
- **DeviceNet:** es un sistema de red que se utiliza en la industria de la automatización para interconectar dispositivos de control para el intercambio de datos.
- **QNX:** es un sistema operativo de tiempo real basado en Unix que cumple con la norma POSIX. Es desarrollado principalmente para su uso en dispositivos empujados. Desarrollado por QNX Software Systems empresa canadiense.
- **ControlNet:** es un protocolo de red abierto para aplicaciones de automatismos industriales, también es conocido como bus de campo. ControlNet fue mantenido en un principio por ControlNet Internacional, pero en 2008 el soporte y administración de ControlNet fue transferido a ODVA, que administra actualmente todos los protocolos de la familia Common Industrial Protocol.

## 13 Referencias bibliográficas y virtuales.

---

- Vallejo, Horacio. Los controladores lógicos programables. ed. Saber Electrónica, No. 166
- Molinari, Nolberto. No. 1.[Documento en línea]. Disponible desde Internet en:  
Formato pdf. en:  
<[http://www.edudevices.com.ar/download/articulos/PLC/CURSO\\_PLC\\_01.pdf](http://www.edudevices.com.ar/download/articulos/PLC/CURSO_PLC_01.pdf)>
- Librería de manuales de PLC´s Allen Bradley y programación en ControlLogix disponibles en  
<[http://literature.rockwellautomation.com/idc/groups/public/documents/webassets/browse\\_category.hcst](http://literature.rockwellautomation.com/idc/groups/public/documents/webassets/browse_category.hcst)>