



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería Mecánica Eléctrica

**IMPLEMENTACIÓN DEL ACCESO, COMUNICACIÓN, MONITOREO Y  
CONTROL DEL PLC DEL LABORATORIO DE ELECTRONEUMÁTICA  
DESDE UNA RED LOCAL**

**William Antonio Hernández Aguirre**

Asesorado por el Ing. Juan Carlos Sánchez Meyer

Guatemala, julio de 2010

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**IMPLEMENTACIÓN DEL ACCESO, COMUNICACIÓN, MONITOREO Y  
CONTROL DEL PLC DEL LABORATORIO DE ELECTRONEUMÁTICA  
DESDE UNA RED LOCAL**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA

POR

**WILLIAM ANTONIO HERNÁNDEZ AGUIRRE**

ASESORADO POR EL INGENIERO JUAN CARLOS SÁNCHEZ MEYER  
AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO ELECTRÓNICO**

GUATEMALA, JULIO DE 2010

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Inga. Glenda Patricia García Soria
VOCAL II	Inga. Alba Maritza Guerrero de López
VOCAL III	Ing. Miguel Ángel Dávila Calderón
VOCAL IV	Br. Luis Pedro Ortiz de León
VOCAL V	Agr. Jose Alfredo Ortiz Herincx
SECRETARIA	Inga. Mayra Grisela Corado

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Luis Manuel Ramírez Ramírez
EXAMINADOR	Ing. Armando Alonzo Rivera Carrillo
EXAMINADOR	Ing. Byron Odilio Arrivillaga Mendez
SECRETARIA	Inga. Marcia Ivonne Véliz Vargas

## **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:

### **IMPLEMENTACIÓN DEL ACCESO, COMUNICACIÓN, MONITOREO Y CONTROL DEL PLC DEL LABORATORIO DE ELECTRONEUMÁTICA DESDE UNA RED LOCAL,**

tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, el 15 de abril del 2009.



**WILLIAM ANTONIO HERNÁNDEZ AGUIRRE**

Guatemala 22 de octubre de 2009

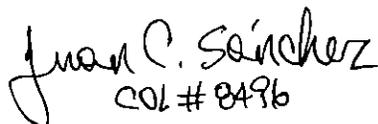
Ingeniero  
Julio Cesar Solares  
Coordinador Área Electrónica  
Escuela de Ingeniería Mecánica Eléctrica

Estimado Ingeniero

Me permito hacer de su conocimiento que he tenido a la vista y he revisado el trabajo de tesis titulado: IMPLEMENTACIÓN DEL ACCESO, COMUNICACIÓN, MONITOREO Y CONTROL DEL PLC DEL LABORATORIO DE ELECTRONEUMÁTICA DESDE UNA RED LOCAL, trabajo que ha realizado el estudiante William Antonio Hernández Aguirre carné 2000-10358, y le doy mi visto bueno y entera aprobación.

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente



Juan C. Sánchez  
COL # 8496

Ing. Juan Carlos Sánchez Meyer  
Catedrático Curso Robótica  
Escuela de Ingeniería Mecánica Eléctrica



Ref. EIME 45.2009  
Guatemala, 10 de NOVIEMBRE 2009.

FACULTAD DE INGENIERIA

Señor Director  
Ing. Mario Renato Escobedo Martínez  
Escuela de Ingeniería Mecánica Eléctrica  
Facultad de Ingeniería, USAC.

Señor Director:

**Me permito dar aprobación al trabajo de Graduación titulado:  
“IMPLEMENTACIÓN DEL ACCESO, COMUNICACIÓN,  
MONITOREO Y CONTROL DEL PLC DEL LABORATORIO DE  
ELECTRONEUMÁTICA DESDE UNA RED LOCAL”, del  
estudiante William Antonio Hernández Aguirre, que cumple con los  
requisitos establecidos para tal fin.**

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,

**ID Y ENSEÑAD A TODOS**

  
**Ing. Julio César Solares Peñate**  
**Coordinador Area de Electrónica**



JCSP/sro



REF. EIME 71. 2009.

**El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; William Antonio Hernández Aguirre titulado: “IMPLEMENTACIÓN DEL ACCESO, COMUNICACIÓN, MONITOREO Y CONTROL DEL PLC DEL LABORATORIO DE ELECTRONEUMÁTICA DESDE UNA RED LOCAL”, procede a la autorización del mismo.**

**Ing. Mario Renato Escobedo Martínez**



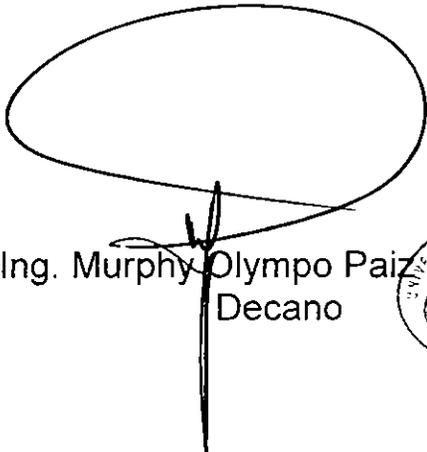
**GUATEMALA, 12 DE NOVIEMBRE 2009.**



Ref. DTG.248.2010

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al trabajo de graduación titulado: **IMPLEMENTACIÓN DEL ACCESO, COMUNICACIÓN, MONITOREO Y CONTROL DEL PLC DEL LABORATORIO DE ELECTRONEUMÁTICA DESDE UNA RED LOCAL**, presentado por el estudiante universitario **William Antonio Hernández Aguirre**, autoriza la impresión del mismo.

IMPRÍMASE

  
Ing. Murphy Olympo Paiz Recinos  
Decano



Guatemala, julio 2010

## **AGRADECIMIENTOS**

Te agradezco Señor mío, por concederme la misericordia de ser tu hijo y haberme concedido reconocer, conocer, ver y tener el regalo más grande de todos, la vida eterna que me has dado en Cristo Jesús.

Te agradezco también por la familia que me concediste a mí, una familia donde me enseñaste de ti y me concediste el privilegio y la bendición de tener unos padres ejemplares en todo, en esfuerzo, en dedicación, en sacrificio, en disciplina, en amor, en paciencia, en dulzura. Gracias especialmente por mis padres José Antonio y Emma Elvira.

Te agradezco también por cada circunstancia y situación que permites en mi vida, pues todo lo que me has permitido ha sido para mi bien, para que conforme a tu propósito yo este hoy aquí.

Te agradezco por los buenos amigos, las tan excelentes personas que me has permitido conocer para hacer de mí la persona que hoy soy, no digo que ya sea algo, pero sé que has hecho de mí una nueva criatura, y que los has usado para que yo aprenda algo de Ti cada día. Tú los conoces a cada uno por su nombre, te ruego bendigas en todo sus vidas.

## ÍNDICE GENERAL

<b>ÍNDICE DE ILUSTRACIONES.....</b>	<b>V</b>
<b>GLOSARIO.....</b>	<b>VII</b>
<b>RESUMEN.....</b>	<b>IX</b>
<b>OBJETIVOS.....</b>	<b>XI</b>
<b>INTRODUCCIÓN.....</b>	<b>XIII</b>
<b>1 CONCEPTOS BÁSICOS DE LOS SISTEMAS: PLC, REDES Y CÓDIGO HTML.....</b>	<b>1</b>
1.1 PLC.....	1
1.2 Redes.....	5
1.3 Código Html.....	8
<b>2 PROCESOS INDUSTRIALES Y PROGRAMACIÓN DE PLC.....</b>	<b>13</b>
2.1 Procesos Industriales.....	13
2.2 Programación de PLC.....	14
<b>3 REDES DE DATOS, TOPOLOGÍA DE REDES Y ETHERNET.....</b>	<b>37</b>
3.1 Redes de Datos o Networking.....	37
3.2 Topología de Redes.....	47
3.2.1 Topología estrella.....	48
3.2.2 Topología Bus.....	48
3.2.3 Topología Estrella/Bus .....	50
3.2.4 Topología Anillo .....	50
3.2.5 Topología Estrella /Anillo .....	51

3.3	Ethernet.....	52
3.4	Direcciones IP .....	55
3.4.1	Clases de redes.....	56
3.4.2	Asignación de direcciones IP.....	58
3.4.3	Máscara de Subred.....	58
<b>4</b>	<b>CÓDIGO HTML Y ADQUISICIÓN DE DATOS.....</b>	<b>61</b>
4.1	Código HTML .....	61
4.1.1	Fondos.....	62
4.1.2	Tablas.....	62
4.1.3	Encabezados.....	64
4.1.4	Párrafos.....	65
4.2	Adquisición de Datos: Controles ActiveX .....	66
4.3	Integración de ActiveX en la página Web.....	71
<b>5</b>	<b>DISEÑO DEL SISTEMA PARA EL ACCESO, MONITOREO Y CONTROL DEL PLC DEL LABORATORIO DE ELECTRONEUMÁTICA DESDE UNA PÁGINA WEB.....</b>	<b>79</b>
5.1	Página WEB.....	79
5.1.1	Página Principal o MAIN.HTML.....	81
5.1.2	Página INDEX.HTML .....	82
5.1.3	Página LINKS.HTML.....	83
5.1.4	Página LOGO.HTML.....	84
5.1.5	Página AYUDA.HTML.....	85
5.1.6	Página COMANDOS.HTML .....	86
5.1.7	Página CONTROLADORES.HTML.....	88
5.1.8	Página DATOS.HTML.....	90
5.1.9	Página DESCUBRIR.HTML.....	92

5.1.10	Página REMOTO.HTML.....	95
5.2	Programa del PLC .....	97
5.3	Comunicación Ethernet PLC - PC: .....	104
<b>CONCLUSIONES.....</b>		<b>107</b>
<b>RECOMENDACIONES.....</b>		<b>109</b>
<b>BIBLIOGRAFÍA.....</b>		<b>111</b>
<b>APÉNDICE.....</b>		<b>113</b>



## ÍNDICE DE ILUSTRACIONES

### FIGURAS

1. Ejemplos de PLC Industriales Festo y Allen Bradley. ....	4
2. Ejemplos redes industriales. ....	7
3. Ejemplo de página Web.....	10
4. Ejemplo de página Web en HTML. ....	11
5. Ejemplo de programación en escalera con contactos, temporizadores, funciones matemáticas. ....	16
6. Ejemplo de control retroalimentado o PID. ....	19
7. Ejemplo de servodrive y servomotor.....	20
8. Ejemplo de aplicación cortadora de precisión.....	21
9. Sinóptico de proceso industrial: Secador.....	22
10. Configuración de PID.....	29
11. Orden de los ciclos en PLC.....	33
12. Ilustración para el ordenamiento de memoria en un PLC.....	34
13. Ejemplo de Red LAN, WAN e Internet.....	38
14. Ejemplo de Internetworking.....	38
15. La información cambia de forma al pasar por un medio de red de aplicación a aplicación.....	40
16. Modelo de referencia OSI.....	41
17. La comunicación no se da de igual a igual directamente sino que pasa por todas las capas del modelo. ....	44
18. Ilustración para el encapsulado de datos.....	46
19. Topología estrella. ....	48
20. Topología bus. ....	49
21. Topología bus-estrella. ....	50
22. Topología anillo.....	51
23. Tarjeta de red.....	53

24. Interfaz de OPC.....	68
25. Página web de OPC Festo. ....	76
26. Topología de red PLC - PC. ....	79
27. Imagen de página principal.....	80
28. Características del PLC FEC FC440. ....	98
29. Conector RJ45 y su configuración de cableado. ....	104
30. Opción 1 conexión directa entre dispositivos.....	105
31. Opción 2 conexión por medio de concentrador o switch. ....	105
32. Ilustración de conexión por medio de concentrador o switch. ....	106

## TABLAS

I Funciones Básicas en Ladder .....	14
II Descripción de los términos de la ecuación PID.....	18
III Asignación de IO para Ejemplo de Programación. ....	24
IV Trama de Ethernet .....	53
V Formato de instrucciones de Fondo en HTML .....	62
VI Formato de instrucciones de Tablas en HTML.....	64
VII Formato de instrucciones de Encabezados en HTML.....	64
VIII Formato de atributos de instrucciones de Tablas en HTML.....	65
IX Formato de Párrafos .....	66
X Formato para los contenedores .....	66
XI Propiedades, Metodos y Eventos del comando CommandX .....	69
XII Propiedades, Métodos y Eventos del comando ExchangeX.....	70
XIII Propiedades, Métodos y Eventos del comando DiscoverX.....	71
XIV Características del PLC FEC FC440 .....	98
XV Asignación de IO para programa de PLC .....	99

## GLOSARIO

<b>ActiveX</b>	Trozo de código prediseñado que puede ser incrustado dentro de otro programa o trozo de código para ejecutar funciones específicas.
<b>HTML</b>	Tipo de programación no formal en la que se diseñan páginas web por su versatilidad, significa por sus siglas en inglés, lenguaje de marcas de hipertexto, y se basa en la programación por marcas.
<b>PLC</b>	Procesador de señales para uso industrial, es una computadora que procesa estados en sus dispositivos de entrada, realiza una operación y toma una decisión que se refleja en los dispositivos de salida. Controla procesos industriales como la fabricación de productos de consumo diario.
<b>Ethernet</b>	Estándar para redes que se basa en el sistema de capas OSI.
<b>Red</b>	Conjunto de entidades conectadas entre si, para compartir recursos de un tipo específico.
<b>Ladder</b>	Lenguaje de programación más utilizado en para programar PLC basado en circuitos eléctricos.
<b>Protocolo</b>	Especificación de reglas a seguir por determinado lenguaje o comunicación. Por ejemplo, se habla de protocolo de comunicación RS232.

**OSI** Modelo o esquema de red descriptivo para asegurar una mayor compatibilidad e interoperabilidad entre distintos tipos de tecnología de red.

**TCP/IP** Protocolo de comunicaciones utilizado en internet, significa por sus siglas en inglés Protocolo de Control de Transmisión/Protocolo de Internet.

**PID** Control Proporcional – Integral – Derivativo se utiliza esta ecuación para resolver problemas en automatización donde se tiene que controlar algún parámetro de operación variable.

**Networking** Término que se refiere a la interoperabilidad de entidades dentro de una red.

## RESUMEN

El presente trabajo de graduación presenta la información necesaria para la implementación del acceso, comunicación, monitoreo y control del PLC del Laboratorio de Electroneumática de la Escuela de Ingeniería Mecánica Eléctrica de la Universidad de San Carlos de Guatemala, desde la red local existente.

La información contenida en este trabajo se puede dividir en cuatro partes mayores. La primera, que se encuentra en el capítulo dos, es la teoría necesaria para entender los sistemas de automatización industrial y la información útil que estos pueden generar y que resulta interesante extraer del sistema. El capítulo uno, se dedica a una breve introducción de todos los sistemas involucrados.

La segunda parte se dedica a las redes de información. Las redes, en general, consisten en "compartir recursos", y uno de sus objetivos es hacer que todos los programas, datos y equipo estén disponibles para cualquiera dentro de la red que así lo solicite, sin importar la localización física del recurso y del usuario. Y es precisamente compartir la información contenida en el PLC, lo que interesa lograr, llevándola así hasta los usuarios finales. Esto es explicado en el capítulo tres.

La tercera parte del trabajo, descrita en el capítulo cuatro, es la interfaz gráfica para el usuario final de los datos extraídos desde el PLC. Para dicha interfaz utilizaremos la programación HTML, o para decirlo de una manera más sencilla, se mostrarán dichos datos en una página Web.

El quinto y último capítulo está dedicado enteramente a la teoría necesaria para la completa implementación del sistema de adquisición de datos en el PLC del laboratorio y la visualización de información en una página Web.

Los códigos de los programas necesarios para todo lo involucrado en el trabajo están incluidos a lo largo del trabajo. Se incluye también un estimado de los costos de elaborar este sistema en el Laboratorio de Electroneumática.

## **OBJETIVOS**

### **GENERAL**

Diseñar un sistema de control basado en un controlador lógico programable que permita la efectiva comunicación, control, monitoreo y acceso a la información del proceso dentro de una red, específicamente la red del Laboratorio de Electroneumática, ubicado en el edificio T1 de la USAC.

### **ESPECÍFICOS**

1. Presentar las características de un PLC, aplicaciones y su utilidad en la automatización de la industria guatemalteca.
2. Mostrar la versatilidad que posee una red de datos que soporta protocolo Ethernet para acceder a un PLC.
3. Demostrar que utilizando los controles ActiveX se puede monitorear un proceso industrial controlado por PLC.



## INTRODUCCIÓN

Tener la información necesaria para tomar decisiones en todo momento, así se puede resumir lo que presenta este trabajo de graduación. Tener la información necesaria de un proceso industrial para tomar las decisiones correctas en todo momento que permitan mejorar el proceso, mejorar la calidad del producto, mejorar las ganancias al determinar donde se da desperdicio, donde no se cumplen requerimientos de eficiencia y productividad, etc.

Todo proceso industrial es pensado con la finalidad de ser eficiente en todas sus facetas, como lo son producción, rapidez, costos, mantenimiento, etc., una forma de hacerlo es controlarlo con un autómata programable, y es este quien lleva el control de cada acción y etapa del proceso para convertir la materia prima en producto final. Otro aspecto tecnológico bastante difundido en nuestra sociedad es el acceso a información valiosa a través de redes de información, como Internet, LAN, etc., tanto así que casi todo creemos poder encontrarlo en la red. El punto central de este trabajo de graduación es afirmar que se puede establecer comunicación, controlar, monitorear y acceder a la información del PLC del laboratorio de electroneumática del Edificio T1, y dejar asentados los principios para acceder a un controlador programado para un proceso específico en una fábrica en particular, con solo entrar a una página Web dentro de una red de área local, de área ancha o también una red VPN, donde se muestren los datos que contiene el PLC acerca del proceso que está controlando.

Para lograr esto se debe contar con un proceso industrial automatizado, una plataforma de control que permita acceder a sus datos de operación de

alguna manera y luego poder visualizarlos de una forma amigable. Una línea de PLC, o Controlador Lógico Programable, del fabricante FESTO tiene un puerto RJ45 que soporta protocolo TCP/IP, esto hace posible la comunicación del controlador a una red, a un servidor o a cualquier equipo con puertos de red, por lo que la traslación de datos desde el PLC a cualquier PC es viable. El PLC sobre el que se basa esta investigación cuenta, además de lo anteriormente descrito, con un controlador de aplicaciones de software desarrollado por la misma casa fabricante para acceder a información interna del PLC, y no solo ello sino también, modificarla, esto es un control ActiveX, mediante el cual es posible acceder por medio de software externo a la información y controles del autómeta.

# **1 CONCEPTOS BÁSICOS DE LOS SISTEMAS: PLC, REDES Y CÓDIGO HTML**

## **1.1 PLC**

Inevitablemente al hablar de PLC se habla de automatización industrial o control industrial. El principio básico del control industrial se basa en la medición de una condición en algún proceso y luego una acción para corregir dependiendo de las condiciones actuales de dicho proceso.

Un PLC es entonces la unidad central de procesamiento de dicha información; con base a la información de la medición de las condiciones del proceso decide qué acción tomar. PLC significa por sus siglas en inglés, Controlador Lógico Programable, y aunque no es el único sistema de control industrial (también existen otros dispositivos como tarjetas electrónicas diseñadas para funciones específicas, como pesadoras por permutación, y los DCS que se utilizan en sistemas con muchas señales analógicas de campo, como en sistemas de alimentos), si es uno de los que más avances ha tenido en la tecnología de la automatización industrial. El PLC ha llegado a competir con la velocidad de procesamiento de señales analógicas de los DCS's y la capacidad de las tarjetas electrónicas de realizar muchos cálculos matemáticos en fracciones de segundo.

Un PLC es entonces el centro del concepto de control automático de procesos. El uso intensivo de la ciencia de control automático es producto de una evolución que es consecuencia del uso difundido de las técnicas de medición y control. Su estudio intensivo ha contribuido al reconocimiento casi universal de sus ventajas. El control automático de procesos se usa

fundamentalmente porque reduce el costo de los procesos industriales, lo que compensa con creces la inversión en equipo de control. Además hay muchas ganancias intangibles, como por ejemplo la eliminación de mano de obra pasiva, la cual provoca una demanda casi equivalente de trabajo especializado. La eliminación de errores es otra contribución positiva del uso del control automático. La repetibilidad del proceso es otro resultado que está íntimamente ligado con el control de calidad del producto final.

En esta misma línea entonces, se ha desarrollado juntamente con los PLC la instrumentación industrial, el control neumático, hidráulico, mecánico, y las comunicaciones que estos soportan, teniendo puertos que “hablan” desde el protocolo RS232 hasta Ethernet IP (IP por Protocolo Industrial, por sus siglas en inglés) que no es más que una de las formas de presentación de la capa 7 del protocolo TCP IP.

Estos autómatas o PLC (indistintamente nos referiremos de aquí en adelante al PLC de cualquiera de estas formas, incluyendo CPU, Controlador) siguen un patrón programado secuencialmente o instanciado. Esto quiere decir que los PLC no actúan por si solos, sino que deben ser programados para ejecutar una tarea específica. También deben ser configurados para interactuar con su entorno, es decir, deben poder recibir y enviar información a equipos externos, como sensores o pantallas de operador.

La programación de un PLC es un tipo de programación que algunos autores llaman no informática, pues originalmente no se podían programar mediante el uso de sintaxis usada en lenguajes como Pascal, C++, etc., sino que se programan comúnmente en un lenguaje llamado escalera (Ladder en inglés). Este tipo de programación fue desarrollado para que fuera de fácil asimilación por los electricistas de mantenimiento de planta, pues la

programación escalera consta básicamente de símbolos que representan una línea de positivo, una línea de negativo, contactos abiertos, contactos cerrados y bobinas que son términos con los que un electricista está muy habituado a trabajar. Los PLC modernos pueden ser programados de muchas formas, desde la lógica de escalera hasta lenguajes de programación tradicionales como BASIC o C. Otro método es usar la Lógica de Estados (State Logic), un lenguaje de programación de alto nivel diseñado para programas PLC basándose en los diagramas de transición de estados.

Recientemente, el estándar internacional IEC 61131-3 se está volviendo muy popular. IEC 61131-3 define cinco lenguajes de programación para los sistemas de control programables: FBD (Function block diagram), LD (Ladder diagram), ST (Structured text, similar al Lenguaje de programación Pascal), IL (Instruction list) y SFC (Sequential function chart).

Mientras que los conceptos fundamentales de la programación del PLC son comunes a todos los fabricantes, las diferencias en el direccionamiento E/S, la organización de la memoria y el conjunto de instrucciones hace que los programas de los PLC nunca se puedan usar entre diversos fabricantes. Incluso dentro de la línea de productos de un mismo fabricante, diversos modelos pueden no ser directamente compatibles.

La estructura básica de cualquier autómata programable es:

Fuente de alimentación: convierte la tensión de la red, 110 ó 220VAC a baja tensión de VDC (24VDC por ejemplo) que es la que se utiliza como tensión de trabajo en los circuitos electrónicos que forma el autómata.

CPU: la Unidad Central de Procesos es el auténtico cerebro del sistema. Es el encargado de recibir órdenes del operario a través de la consola de

programación y el módulo de entradas. Después las procesa para enviar respuestas al módulo de salidas.

Módulo de entradas: aquí se unen eléctricamente los captadores (interruptores, sensores, finales de carrera...). La información que recibe la envía al CPU para ser procesada según la programación. Hay dos tipos de captadores conectables al módulo de entradas: los pasivos y los activos. Los módulos de entrada pueden ser también de tipo digital o analógico.

Módulo de salida: es el encargado de activar y desactivar los actuadores (bobinas de contactores, motores pequeños, bobinas de electroválvulas). La información enviada por las entradas a la CPU, cuando está procesada se envía al módulo de salidas para que éstas sean activadas (también los actuadores que están conectados a ellas). Hay tres módulos de salidas de tipo digital según el proceso a controlar por el autómatas: relés, triac y transistores. Entre los analógicos están las señales 4-20mA o 0-10VDC.

Figura 1. Ejemplos de PLC Industriales Festo y Allen Bradley.



Terminal de programación: la terminal o consola de programación es el que permite comunicar al operario con el sistema. Sus funciones son la transferencia y modificación de programas, la verificación de la programación y la información del funcionamiento de los procesos. Puede ser una PC o un terminal de programación.

Periféricos: ellos no intervienen directamente en el funcionamiento del autómatas, pero sí que facilitan la labor del operario, a estos más comúnmente se les llama paneles de visualización o paneles de operador.

Dentro de los fabricantes más reconocidos en la industria de la automatización se pueden mencionar a Festo (el PLC del laboratorio de electroneumática pertenece a esta marca y que tiene un área dedicada a Educación y ha desarrollado mucho control en neumática), Allen Bradley (marca bajo la cual se fabrica el hardware y el software es adjudicado a Rockwell Software que es una misma empresa; pero esta parte se dedica a desarrollar software), Siemens de origen alemán, Omron, Mitsubishi, Yokogawa, por mencionar algunos que se pueden encontrar en la industria guatemalteca.

## **1.2 Redes**

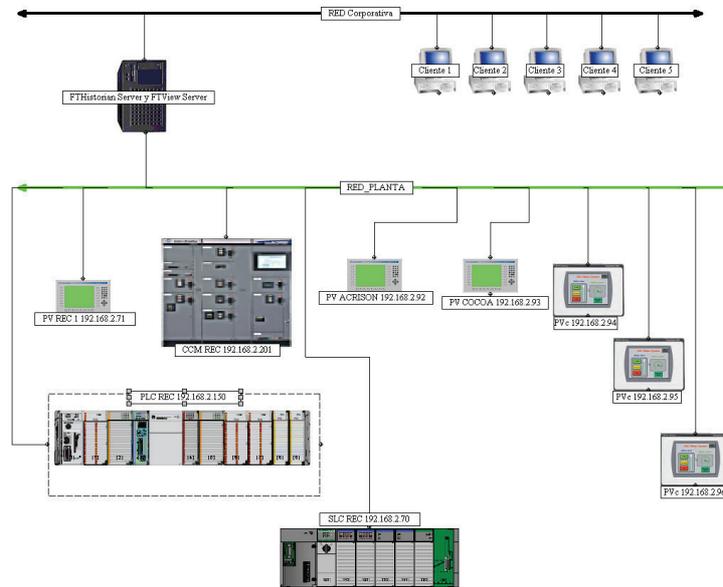
Dentro de ingeniería al nombrarse una red rápidamente se piensa en una red informática con una disposición servidor – cliente, Internet y recursos compartidos. Pero en términos más amplios, el término genérico "red" hace referencia a un conjunto de entidades (objetos, personas, etc.) conectadas entre sí. Por lo tanto, una red permite que circulen elementos materiales o inmateriales entre estas entidades, según reglas bien definidas.

Así, según el tipo de entidad involucrada, el término utilizado variará y se pueden tener por ejemplos: una red de transporte (conjunto de infraestructuras y vehículos usados para transportar personas y bienes entre diferentes áreas geográficas) una red telefónica (infraestructura usada para transportar señales de voz desde una estación telefónica a otra) o una red neural (conjunto de neuronas conectadas entre sí).

Por definición una red es un conjunto de dispositivos interconectados físicamente (ya sea vía alámbrica o vía inalámbrica) que comparten recursos y que se comunican entre sí, a través de reglas (protocolos) de comunicación. Dicho de otra forma, una red está formada por un medio de comunicación por el cual se transfiere la información, esto se le conoce como topología de red, esta puede ser alámbrico o inalámbrico; uno o más recursos por compartir, pueden ser recursos físicos como dispositivos de red monitores, cdroms, o información en forma de datos o archivos; y un lenguaje o reglas para la comunicación, que puede ser Ethernet, tcp/ip, ipx, etc.

Las redes pueden ser clasificadas por el tipo de recurso o información que se comparte. Se pueden encontrar redes de datos (compañías de beepers, de celulares SMS, proveedores de Internet, etc.), redes de video, redes de voz (compañías celulares), redes de audio (rockolas digitales, audio por Internet), o combinaciones de estas.

Figura 2. Ejemplos redes industriales.



La red que a nosotros nos concierne no es estrictamente una red informática, sino mas bien, una red de tipo industrial, es decir, una red en la cual los dispositivos que la conforman son controladores lógicos programables, computadoras personales, pantallas de operador, hubs, switches, cable utp (unshielded twisted pair). El protocolo a usar es Ethernet. La topología o forma física es una combinación entre tipo estrella y tipo bus. De las topologías, medios, direcciones IP, protocolo Ethernet y otros detalles hablaremos más a fondo en el capítulo tres.

Al final, las ventajas inherentes de una red son las capacidades de poder compartir información y recursos de una manera fácil y rápida. Al compartir información rápidamente o en tiempo real se pueden tomar decisiones importantes con la información precisa en el momento adecuado. Además, por lo general una red conlleva ventajas de ahorro económico al permitir un fácil

manejo y traslación de recursos entre diferentes lugares. Estas dos ventajas son las más sobresalientes para el tema que nos interesa.

### **1.3 Código Html**

HTML o Hyper Text Markup Language por sus siglas en inglés, es el lenguaje de programación por defecto para diseñar páginas de Internet y otras que no precisamente son de uso en la Web. No es estrictamente un lenguaje de programación, sino mas bien un formato que se caracteriza por poseer una serie de marcadores que designan las palabras reservadas del lenguaje, y entre cada marcador se explicitan los parámetros que se le darán a dicho objeto.

Este “lenguaje” surgió como una ampliación del SGML (Standard Generalized Markup Language), que consiste en un sistema para definir tipos de documentos estructurados y lenguajes de marcas para representar esos mismos documentos. Fue el físico nuclear Tim Berners-Lee. el primero en presentar este lenguaje como una solución a la necesidad que tenía de poder acceder a documentos específicos dentro de una muy amplia librería. Originalmente, él lo diseñó con 22 elementos, de los cuales 13 siguen siendo utilizados en el actual formato de HTML.

El HTML se ha consolidado como la más común herramienta para escribir páginas Web en la actualidad, aunque hay otros “lenguajes” que también pueden servir para esto o para añadir funcionalidad a páginas escritas en HTML, como PHP o Java. La versión actual de HTML es la 4.01 introducida en 1999.

De manera muy básica las instrucciones de HTML se componen de tres partes fundamentales:

A) El elemento al que se refiere la instrucción. Es la palabra que va después del signo <>.

Por ejemplo: <font> es un elemento. En este caso el elemento es el tipo de letra. La instrucción por tanto se refiere al tipo de letra.

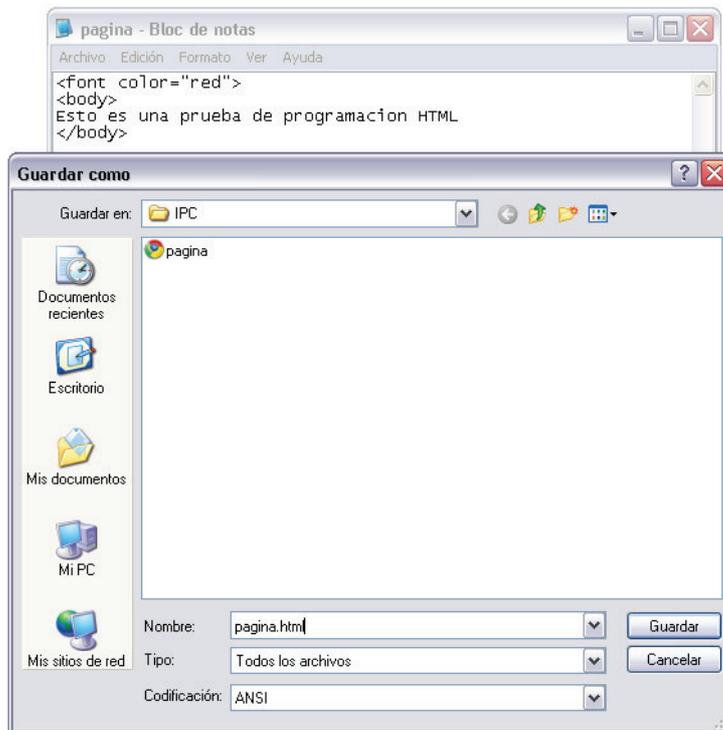
B) El atributo. El tamaño o el color son atributos posibles del tipo de letra.

C) El valor que toma el atributo. Por ejemplo, rojo o verde.

Por ejemplo, si se quiere al elemento "tipo de letra", darle el atributo color, con el valor rojo:

Elemento	Atributo	Valor	El tag de cierre
<FONT	COLOR	"RED	</FONT>
=	">		

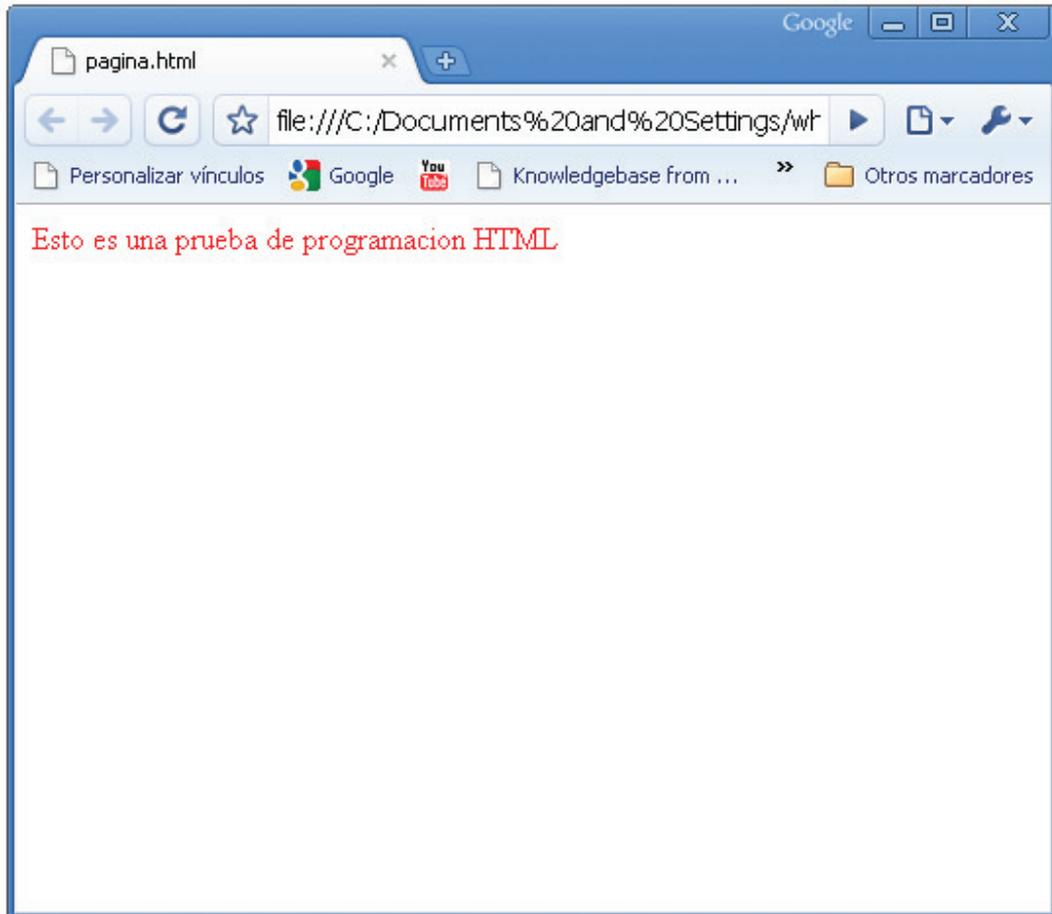
Figura 3. Ejemplo de página Web.



Para realizar una página Web existen varias herramientas en el mercado, como Microsoft FrontPage o Macromedia Dreamweaver, que son herramientas especializadas para hacer páginas Web con un interfaz de usuario muy amigable. Aunque lo único que se necesita es un editor de texto como Notepad, donde se puede escribir una instrucción como la del ejemplo anterior y al guardar el archivo se le da la extensión .HTML, y al abrirlo desde un explorador de Internet como MS Internet Explorer o Google Chrome, el texto se ve como una página Web. La siguiente figura lo ilustra mejor.

Luego, el archivo creado se puede abrir en un explorador Web. Nuestro ejemplo se verá como lo siguiente:

Figura 4. Ejemplo de página Web en HTML.





## **2 PROCESOS INDUSTRIALES Y PROGRAMACIÓN DE PLC**

### **2.1 Procesos Industriales**

Un proceso industrial o proceso de fabricación es el conjunto de operaciones necesarias para modificar las características de las materias primas y obtener un producto final. Dichas características pueden ser de naturaleza muy variada tales como la forma, la densidad, la resistencia, el tamaño, la estética, la composición química, el sabor, etc. En otras palabras un proceso industrial es el método por el cual obtenemos todos los productos que conocemos y utilizamos en la vida cotidiana, por ejemplo, del maíz obtener cereales extrusados o cocidos; de una determinada y cuidadosa combinación de polímeros, ácidos y sulfatos obtener jabones y detergentes, etc.

En la inmensa mayoría de los casos, para la obtención de un determinado producto serán necesarias multitud de operaciones individuales, de modo que, dependiendo de la escala de observación, puede denominarse proceso tanto al conjunto de operaciones desde la extracción de los recursos naturales necesarios hasta la venta del producto, como a las realizadas en un puesto de trabajo con una determinada máquina-herramienta. En el ámbito industrial se suelen considerar convencionalmente en dos grandes familias: Procesos de Tecnología Mecánica y Tecnología Química.

En la industria guatemalteca se puede encontrar procesos de alimentos, como cereales, pastas, bebidas carbonatadas y no carbonatadas, productos derivados lácteos, frituras, etc. Procesos agroindustriales como insecticidas, venenos, abonos químicos, abonos orgánicos y combinaciones de ambos.

Procesos de generación eléctrica, por hidroeléctricas o por quema de carbón o combustibles fósiles. Procesos de manufactura, como ropa, jeans, etc.

## 2.2 Programación de PLC

Normalmente un proceso industrial suele necesitar de supervisión y control. Para el control se dispone del PLC y para la supervisión se dispone de pantallas de operador y servidores de información, por donde pasa toda la información del proceso (datos, orden de activación de máquinas, datos de proceso, etc.). Dicha información pasa por una red (Internet, LAN, DH+ etc.), que une los distintos elementos o clientes. El servidor, además, debe estar asociado a una base de datos para poder consultar y/o tratar los diferentes datos de que dispondremos, acción que conocemos como recolección de datos.

Como se mencionó en el capítulo uno, la programación Escalera es la más común y de más fácil asimilación para programar PLC por derivarse directamente del concepto de Relevación Industrial. En ella nos centraremos para este trabajo. Empezaremos definiendo algunos estándares de programación Escalera.

Los términos más usados en la programación ladder son los contactos normalmente cerrados NC, normalmente abiertos NO, bobinas OTE o R1, etc.

Tabla I. Funciones Básicas en Ladder.

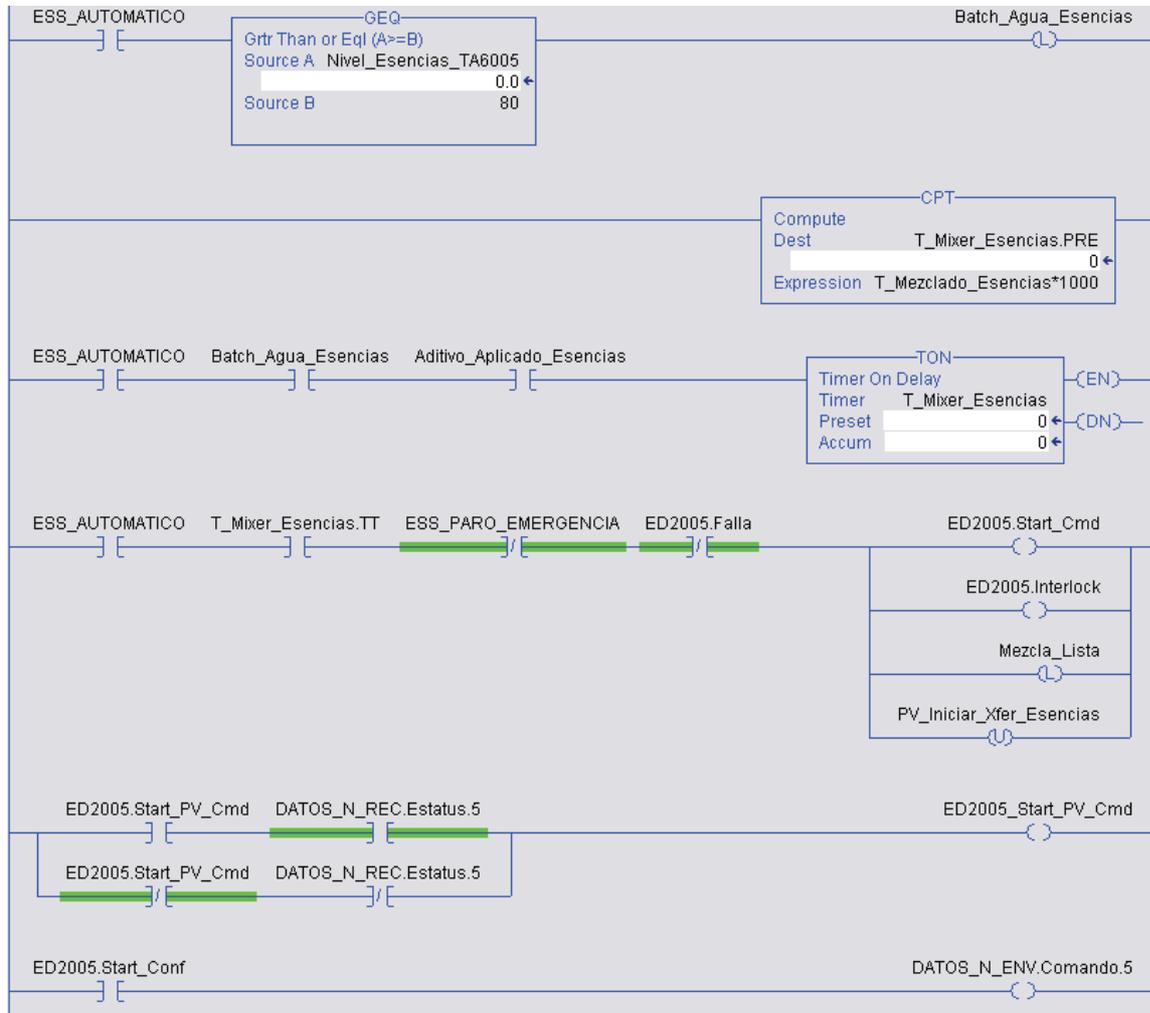
Tipo de Contacto	Visualización	Pasa corriente cuando...
Normalmente abierto		La referencia está ON
Normalmente cerrado		La referencia está OFF

Tipo de Bobina	Visualización	¿Corriente a la bobina?	Resultado
Normalmente abierta	—( )—	Si	Referencia ON
		No	Referencia OFF
Inversa	—( / )—	Si	Referencia OFF
		No	Referencia ON
Transición Positiva	—(↑)—	No → Si	Ref. ON durante un barrido
Transición Negativa	—(↓)—	Si → No	Ref. OFF durante un barrido
Set	—(s)—	Si	Ref. ON hasta que se apague con "R"
		No	Sigue igual
Reset	—(R)—	Si	Ref. OFF hasta que se active con "S"
		No	Sigue igual

Sumado a estos conceptos tenemos también, las funciones de temporizado (timers retentivos, timers acumulativos, timers), conteo (contadores hacia arriba, contadores hacia abajo), funciones matemáticas (suma, resta, valor absoluto, etc.) y funciones especiales que dependen de cada fabricante.

Al combinar este tipo de instrucciones se obtienen rutinas de control para prácticamente cualquier proceso industrial, desde los más sencillos hasta los controles más complicados. Se pueden lograr rutinas de control muy completas, desde encender un motor a la petición de un operador, hasta controlar la apertura o cierre de distribuidores de tipo hidráulico para permitir un flujo de agua hacia las paletas de una turbina en una generadora hidroeléctrica para mantener constante la entrega de potencia al sistema de energía eléctrica.

Figura 5. Ejemplo de programación en escalera con contactos, temporizadores, funciones matemáticas.



Cuando se habla de programar PLC existen diferentes tipos de control:

**Control todo-nada:** este tipo de control es el más simple, siendo programado como una orden directa desde una condición simple, es decir, se activa una salida al impulso de activación de una entrada, como por ejemplo, el encendido de un motor por decisión de operador.

**Control secuencial:** en este control se lleva una secuencia de operación, es decir, se permite el accionamiento del siguiente actuador o condición cuando se han dado ciertas condiciones anteriores o permisivos, como podrían ser el caso de encendido de un motor de banda hasta que se ha detectado que hay producto y además todas las partes de la banda están en su lugar y los sensores de seguridad están habilitados.

**Control retroalimentado:** es más comúnmente conocido como control PID (Proportional Integral Derivative), donde se tiene una señal para controlar o CV, una señal medida o de retroalimentación o PV, y un valor deseado o Setpoint SP. Básicamente la operación de un PID es la de controlar o llevar el valor de un parámetro hacia el deseado, controlando la salida del sistema. En otras palabras, si se quiere controlar y mantener la humedad relativa estable y fija dentro de un secador, por ejemplo, se tiene un medidor de RH% que entrega una señal proporcional a la humedad relativa en forma de 0 a 100%, luego un actuador que podría ser una compuerta para permitir o impedir el paso de vapor hacia el sistema, o un motor con velocidad variable para mover un ventilador que extraiga humedad del ambiente, y además un valor deseado de RH%. Así pues, el PID controla la salida hacia el actuador (digamos una compuerta) para abrir o cerrar según se va modificando la RH% dentro del sistema hasta llevarla al valor deseado. El control PID responde a la ecuación de control de lazo cerrado siguiente: (puede variar según el fabricante)

$$CV_n = CV_{n-1} + K_c \left( \Delta E + \frac{1}{60T_I} E \Delta t + 60T_D \frac{E_n - 2E_{n-1} + E_{n-2}}{\Delta t} \right)$$

Donde:

$$K_P = K_C$$

$$K_I = \frac{K_C}{T_I}$$

$$K_D = K_C T_D$$

Y

Tabla II. Descripción de los términos de la ecuación PID.

PIDE term:	Description:
CV	control variable
E	error in percent of span
$\Delta t$	update time in seconds used by the loop
$K_C$	controller gain
$T_I$	integral time constant in minutes per repeat a larger value of $T_I$ causes a slower integral response  It takes $T_I$ minutes for the integral term to repeat the action of the proportional term in response to a step change in error.
$T_D$	derivative time in constant in minutes

De manera que lo que se puede ajustar para tener una mejor respuesta son las constantes  $K_p$ ,  $K_i$  y  $K_d$ , que determinan el peso de cada componente en la ecuación.

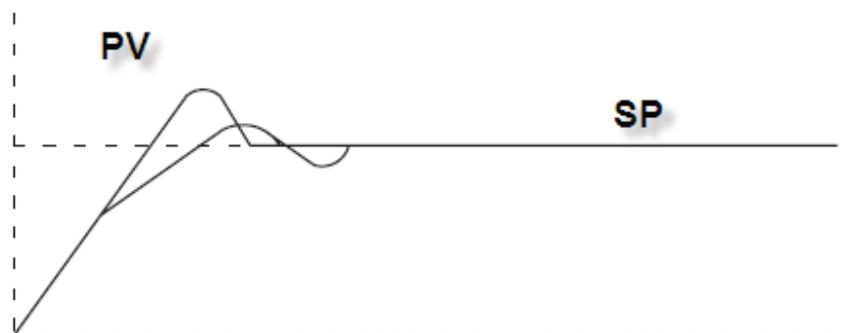
El peso que cada constante tiene dentro de la ecuación puede parecer a primera vista algo complejo, pues la forma de la ecuación solución PID es un cálculo integrodiferencial. A continuación explico la función de cada constante dentro de la ecuación.

Constante Proporcional  $K_p$ : La acción proporcional es una operación en la que la banda proporcional es establecida con respecto al setpoint SP, y

dentro de esa banda la variable de control CV se hace proporcional a la desviación. Esta constante es la de mayor peso cuanto más lejos se encuentra la variable de proceso PV del setpoint SP.

Constante Diferencial Kd: La acción proporcional y la acción integral ambas hacen correcciones con respecto al resultado del control, así que resulta un inevitable atraso en la respuesta. La acción derivativa compensa ese inconveniente. Como respuesta a un súbito disturbio esta entrega un gran valor en CV y rápidamente restaura el estado original. Una corrección es ejecutada con la CV hecha proporcional a la pendiente (coeficiente derivativo) causada por la desviación o disturbio. El peso de la acción derivativa es indicado por el tiempo de derivación, que es el tiempo requerido por la CV de la acción derivativa para alcanzar el mismo nivel que la CV de la acción proporcional con respecto al tamaño del error. Entre más grande sea el tiempo de derivación, mayor será la corrección de la acción derivativa.

Figura 6. Ejemplo de control retroalimentado o PID.



Constante Integral Ki: Combinando la acción integral con la acción proporcional se reduce el desvío de acuerdo al tiempo que ha transcurrido de tal manera que PV se ajuste al SP. El peso de la acción integral es indicado por el tiempo de integración, que es el tiempo requerido por la CV debido a la

acción integral para que esta tome el mismo nivel que la CV de la acción proporcional con respecto al tamaño del error. Entre más corto el tiempo de integración, más peso tiene en el comportamiento. Si el tiempo de integración es muy corto, mayor será la corrección hecha a CV en el tiempo y produce el efecto sierra en la curva.

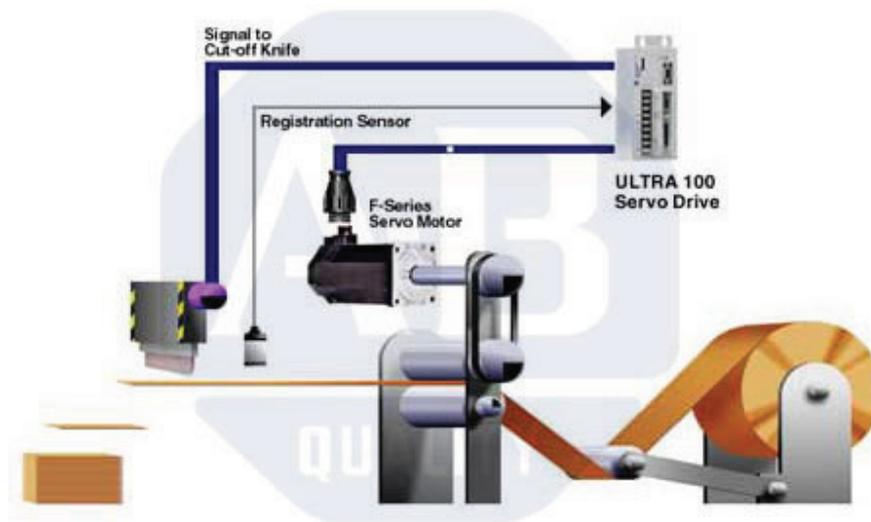
**Control de movimiento:** se conoce en el mercado industrial como Motion Control. Como su nombre indica, se basa en algoritmos de respuesta ultra rápidos para manipular un movimiento, su velocidad, su aceleración y su torque de una manera precisa y veloz. El más claro ejemplo de control de movimiento son los servomotores industriales. Estos servos tienen un motor inteligente, que tiene señal de potencia y una señal de retroalimentación de posición, en un mismo conector que va desde el servomotor hasta el servodrive o controlador de movimiento.

Figura 7. Ejemplo de servodrive y servomotor.



Las aplicaciones más comunes para servos son en empacadoras, brazos robóticos, cortadoras, formadoras, etc.

Figura 8. Ejemplo de aplicación cortadora de precisión.



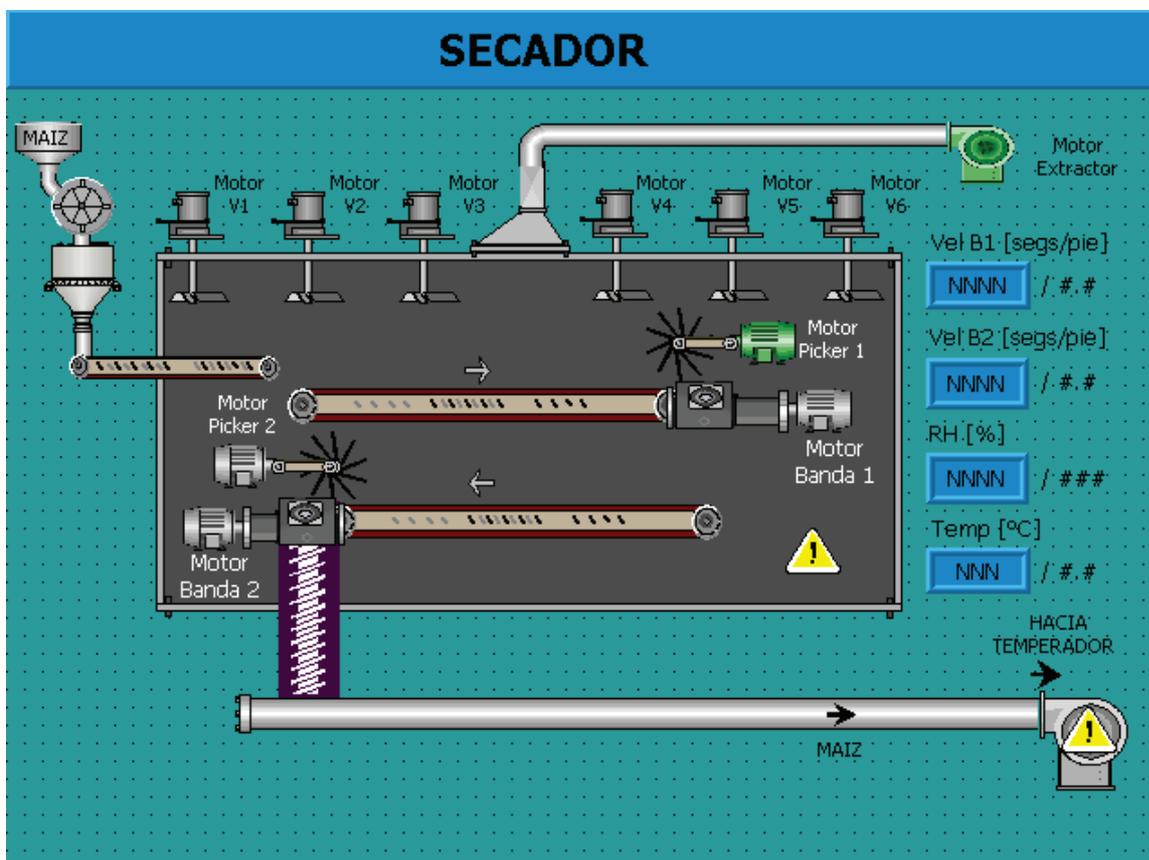
Podemos ilustrar mejor todo lo dicho y expuesto hasta este punto con un ejemplo. Supongamos un proceso de manufactura de cereales, donde después de cocido el cereal (en este caso maíz) este se pasa por un proceso de secado para llevarlo a una humedad deseada. El control tiene dos bandas motorizadas, un extractor, 6 ventiladores, un gusano para sacar el producto del secador, dos motores para picadores o desgrumadores de producto, una compuerta para ingreso de vapor al secador, un medidor de humedad relativa, un medidor de temperatura y sensores de seguridad en puertas del sistema.

Supongamos que la secuencia de encendido es la siguiente:

1. Primero, el operador decide cuándo encender la banda de entrada o banda 1.
2. Los motores de picadores arrancan solo si la banda uno está encendida.
3. La banda dos arranca si la banda uno y los picadores están encendidos.

4. El gusano arranca si la banda dos está encendida.
5. Los ventiladores y extractor encienden a decisión de operador.
6. Los medidores de RH% y temperatura están todo el tiempo activados.
7. La compuerta de vapor y su control están todo el tiempo activos.

Figura 9. Sinóptico de proceso industrial: Secador.



Suponemos además que sólo existe control automático para los equipos principales, y para los ventiladores y extractor sólo control manual.

Lo primero por hacer es definir la conexión eléctrica del sistema descrito. Esto es muy importante, pues se cuentan con entradas y salidas digitales para los motores e indicadores y sensores. Además, también se tienen entradas y salidas analógicas para los sensores de RH% y temperatura. Por regla general, el primer paso es definir las conexiones eléctricas, que son los planos eléctricos del sistema (no entraremos en detalle de los planos eléctricos sus estándares, simbología y software pues no son parte del presente trabajo). En este caso tendríamos la siguiente tabla de IO:

Empezamos por definir que cada motor contará con una entrada de confirmación de arranque (normalmente se cablea en un contacto auxiliar del contactor del respectivo motor), una salida para arranque del contactor, una entrada para encender la banda 1 y cada ventilador y extractor, dos entradas de seguridades de puertas (una puerta del secador y una puerta del gusano), una entrada analógica 4-20mA para cada sensor analógico, y una salida analógica 4-20mA también para la compuerta de vapor. Esto da un total de 22 entradas digitales DI, 12 salidas digitales, dos entradas analógicas y una salida analógica.

Tabla III. Asignación de IO para ejemplo de programación.

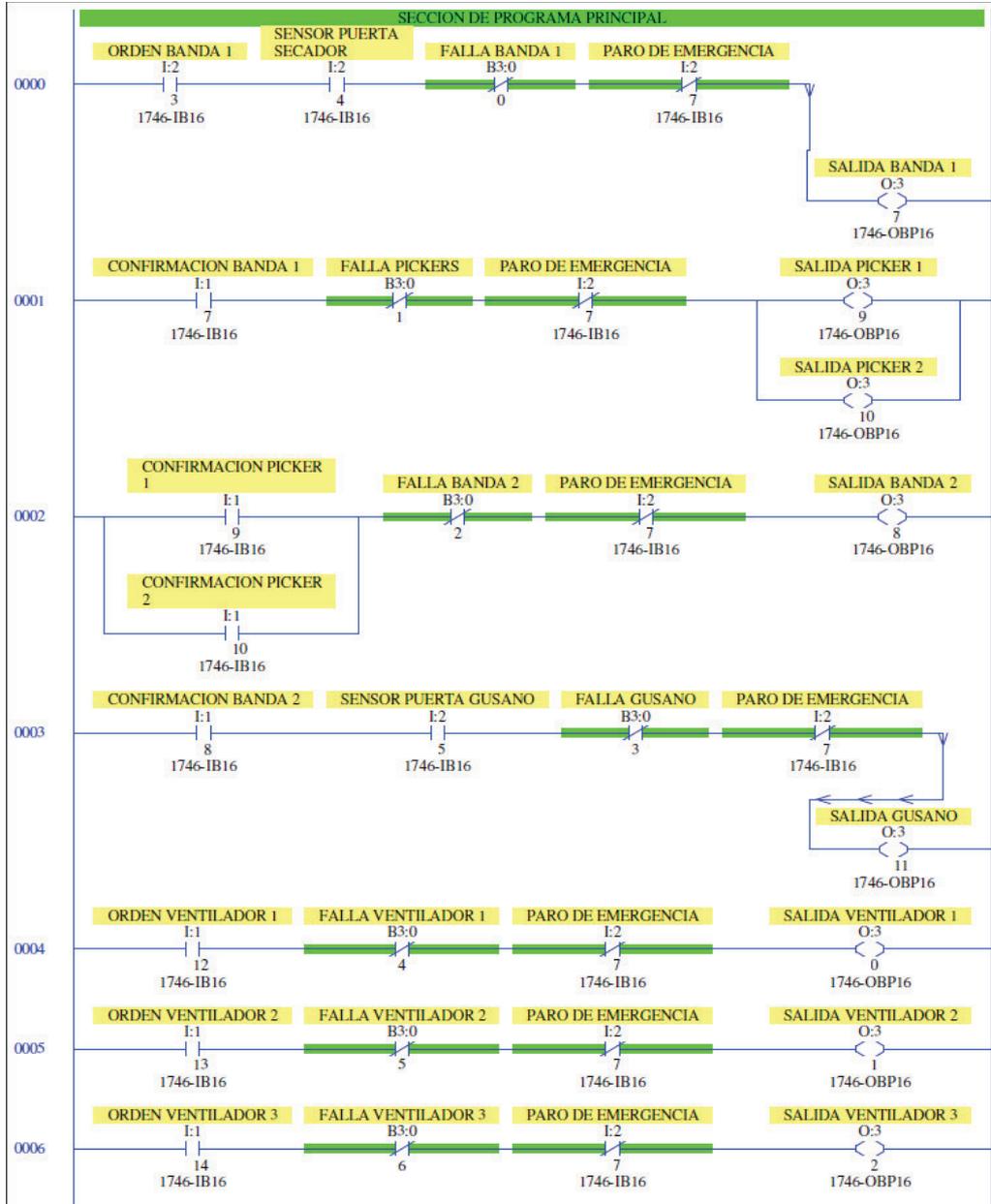
LABORATORIO ELECTRONEUMÁTICA		LISTADO DE SEÑALES				
			0		0	
No.	Descripción	I/O	Tipo	Vaux	No. ID 2	No. ID 3
1	Confirmación Arranque Motor Ventilador 1	DI		24VDC	I:1/0	BORNE X100 / 1
2	Confirmación Arranque Motor Ventilador 2	DI		24VDC	I:1/1	BORNE X100 / 2
3	Confirmación Arranque Motor Ventilador 3	DI		24VDC	I:1/2	BORNE X100 / 3
4	Confirmación Arranque Motor Ventilador 4	DI		24VDC	I:1/3	BORNE X100 / 4
5	Confirmación Arranque Motor Ventilador 5	DI		24VDC	I:1/4	BORNE X100 / 5
6	Confirmación Arranque Motor Ventilador 6	DI		24VDC	I:1/5	BORNE X100 / 6

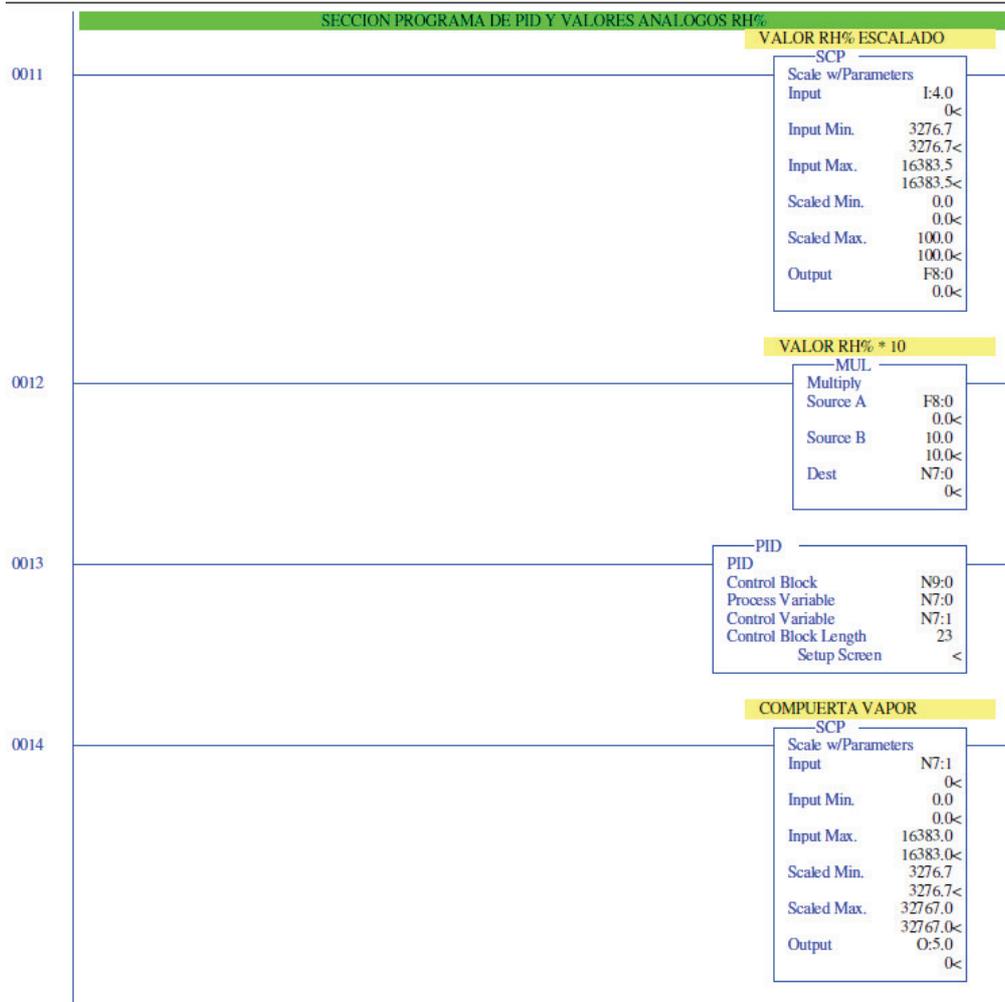
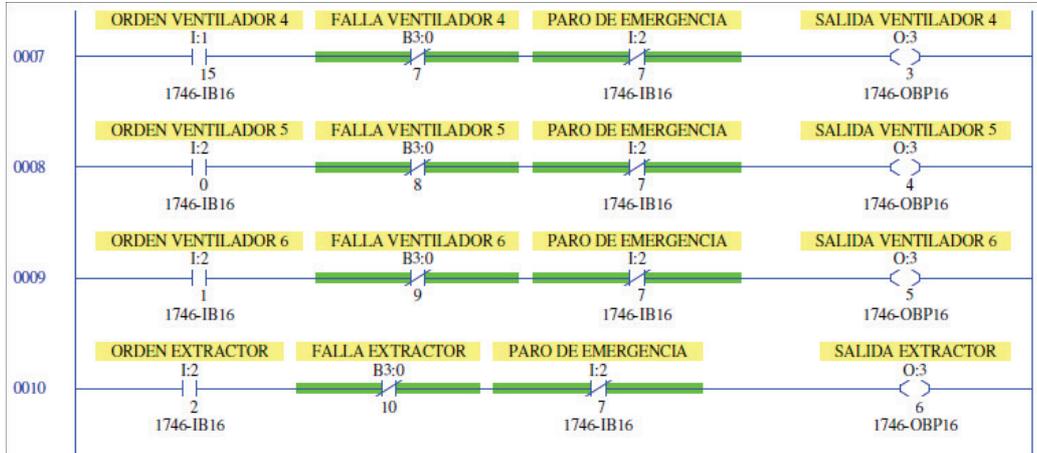
7	Confirmación Arranque Motor Extractor	DI	24VDC	I:1/6	BORNE X100 / 7
8	Confirmación Arranque Motor Banda 1	DI	24VDC	I:1/7	BORNE X100 / 8
9	Confirmación Arranque Motor Banda 2	DI	24VDC	I:1/8	BORNE X100 / 9
10	Confirmación Arranque Motor Picker 1	DI	24VDC	I:1/9	BORNE X100 / 10
11	Confirmación Arranque Motor Picker 2	DI	24VDC	I:1/10	BORNE X100 / 11
12	Confirmación Arranque Motor Gusano	DI	24VDC	I:1/11	BORNE X100 / 12
13	Orden Arranque Motor Ventilador 1	DI	24VDC	I:1/12	BORNE X100 / 13
14	Orden Arranque Motor Ventilador 2	DI	24VDC	I:1/13	BORNE X100 / 14
15	Orden Arranque Motor Ventilador 3	DI	24VDC	I:1/14	BORNE X100 / 15
16	Orden Arranque Motor Ventilador 4	DI	24VDC	I:1/15	BORNE X100 / 16
17	Orden Arranque Motor Ventilador 5	DI	24VDC	I:2/0	BORNE X100 / 17
18	Orden Arranque Motor Ventilador 6	DI	24VDC	I:2/1	BORNE X100 / 18
19	Orden Arranque Motor Extractor	DI	24VDC	I:2/2	BORNE X100 / 19
20	Orden Arranque Motor Banda 1	DI	24VDC	I:2/3	BORNE X100 / 20
21	Sensor Puerta Secador	DI	24VDC	I:2/4	BORNE X100 / 21
22	Sensor Puerta Gusano	DI	24VDC	I:2/5	BORNE X100 / 22
23	Reset de Fallas	DI	24VDC	I:2/6	BORNE X100 / 23
24	Paro Emergencia	DI	24VDC	I:2/7	BORNE X100 / 24
25	Salida Arranque Motor Ventilador 1	DO	24VDC	O:3/0	BORNE X100 / 25
26	Salida Arranque Motor Ventilador 2	DO	24VDC	O:3/1	BORNE X100 / 26
27	Salida Arranque Motor Ventilador 3	DO	24VDC	O:3/2	BORNE X100 / 27
28	Salida Arranque Motor Ventilador 4	DO	24VDC	O:3/3	BORNE X100 / 28
29	Salida Arranque Motor Ventilador 5	DO	24VDC	O:3/4	BORNE X100 / 29
30	Salida Arranque Motor Ventilador 6	DO	24VDC	O:3/5	BORNE X100 / 30
31	Salida Arranque Motor Extractor	DO	24VDC	O:3/6	BORNE X100 / 31
32	Salida Arranque Motor Banda 1	DO	24VDC	O:3/7	BORNE X100 / 32
33	Salida Arranque Motor Banda 2	DO	24VDC	O:3/8	BORNE X100 / 33
34	Salida Arranque Motor Picker 1	DO	24VDC	O:3/9	BORNE X100 / 34
35	Salida Arranque Motor Picker 2	DO	24VDC	O:3/10	BORNE X100 / 35
36	Salida Arranque Motor Gusano	DO	24VDC	O:3/11	BORNE X100 / 36
37	Sensor RH%	AI	20mA	O:4/0	BORNE X100 / 37
38	Sensor Temperatura	AI	20mA	O:4/1	BORNE X100 / 38
39	Salida Control Compuerta Vapor	AO	20mA	O:5/0	BORNE X100 / 39
40					

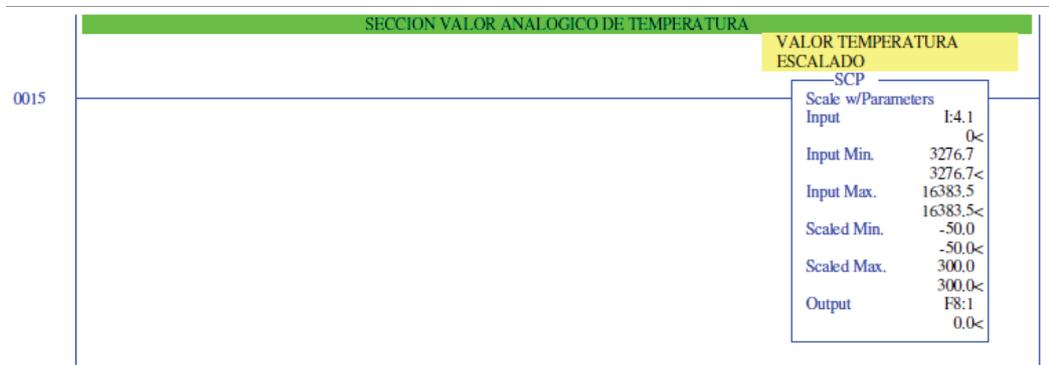
Cabe destacar de la lista de señales anterior, que se ha supuesto un PLC modular con dos módulos de 15 entradas digitales, un módulo de 15 salidas

digitales, un módulo de cuatro entradas analógicas, y un módulo de dos salidas analógicas. Además de lo anterior, he de decir que se ha asignado un número de borne a cada señal, pues es un estándar armar un tablero con lo anterior y llevar cada señal hasta bornes y de ahí salir con las señales a otros tableros de campo o potencia.

Para este tema de graduación, usaremos este mismo programa pero con algunas modificaciones al ejemplo de programación que estamos a punto de mostrar, pues el PLC del laboratorio no tiene módulos de expansión. Simularemos como medida alternativa, algunas de estas señales, las analógicas y algunas digitales, y eliminaremos la función de control. En el siguiente ejemplo de programación usaremos el software de programación de la casa matriz Allen Bradley, el Rslogix500.

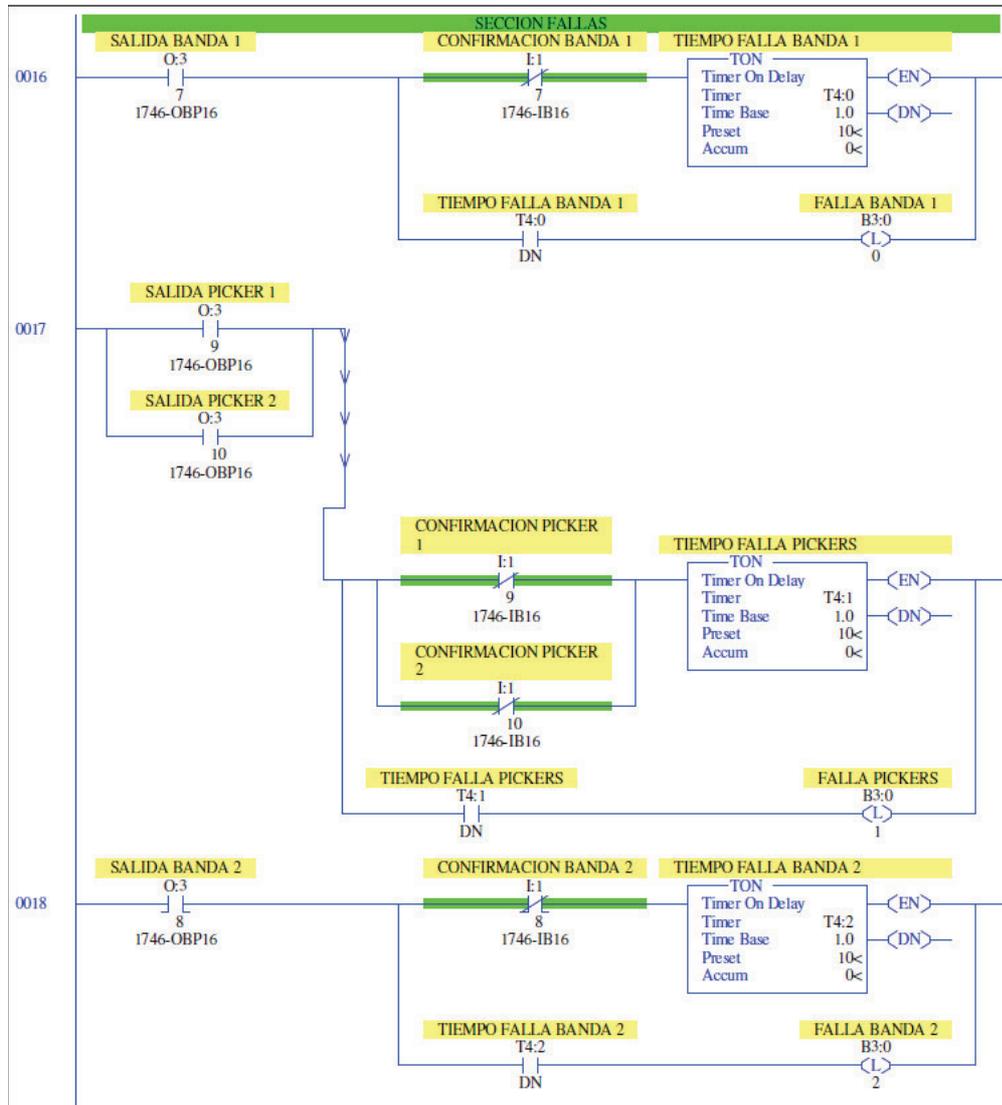
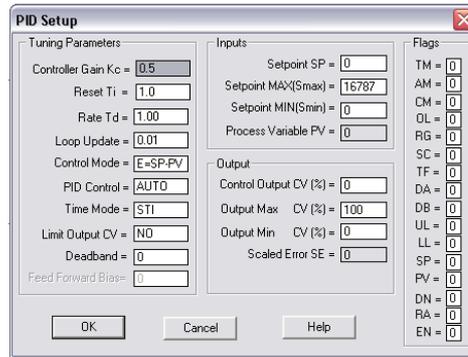


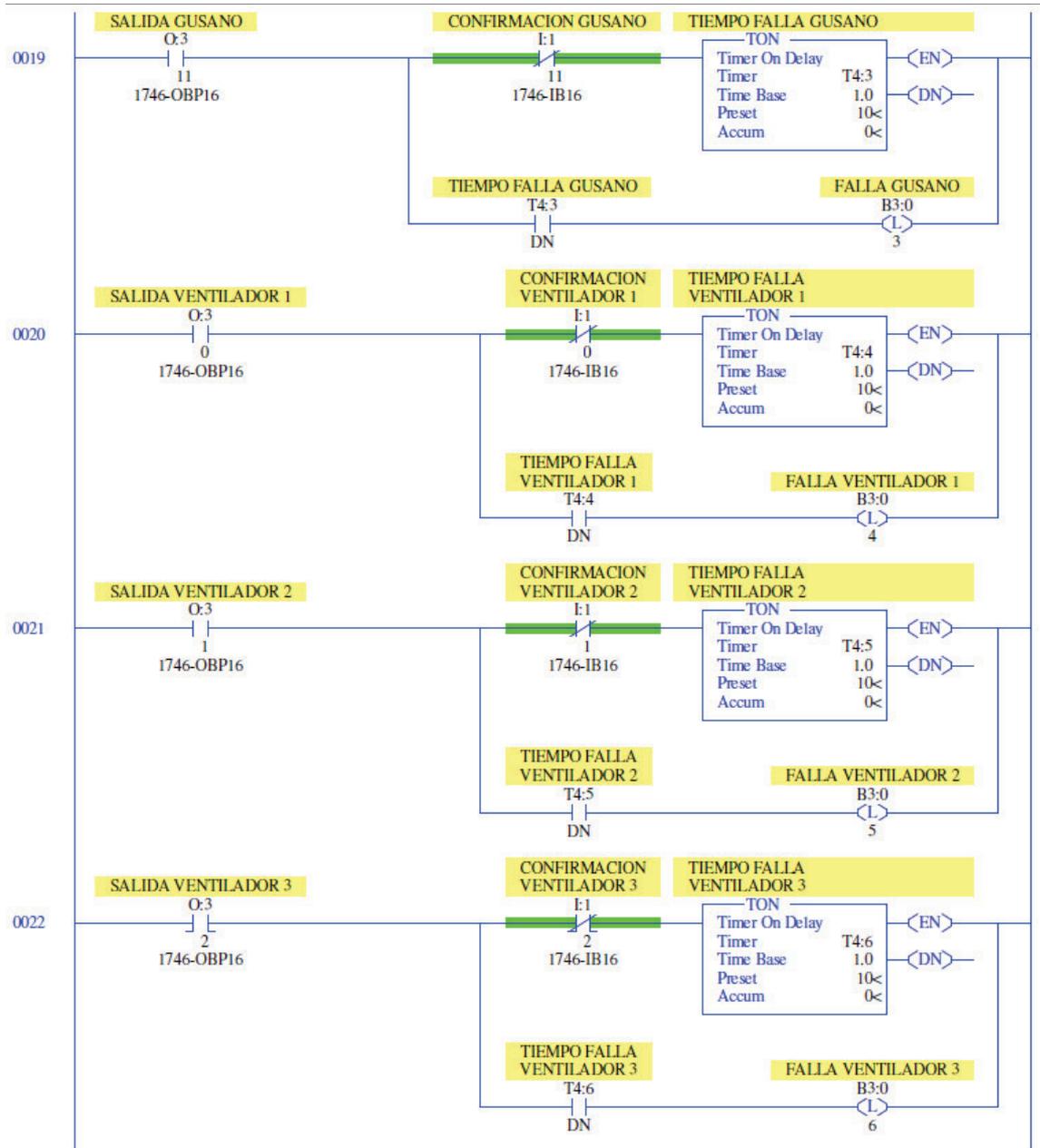


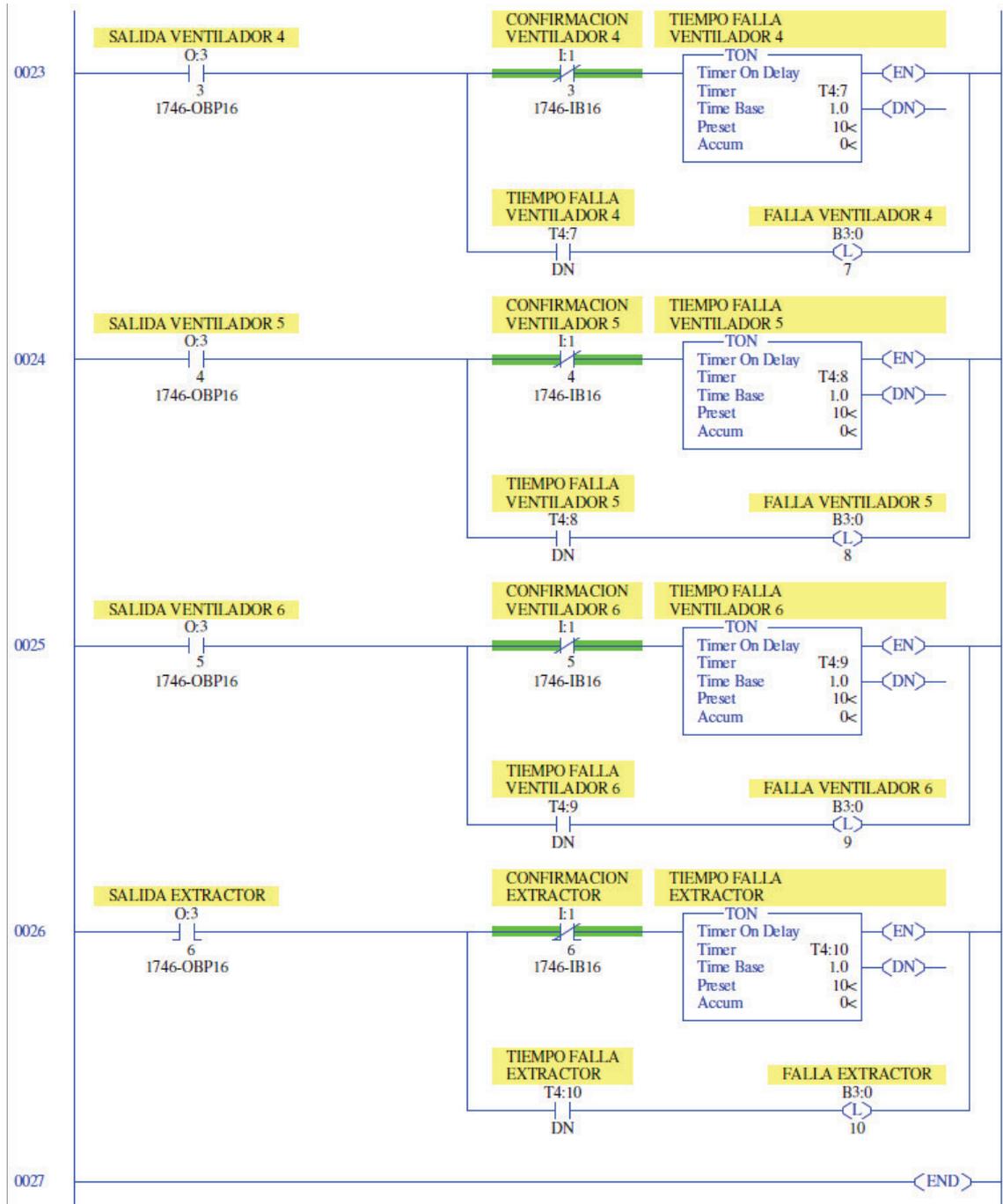


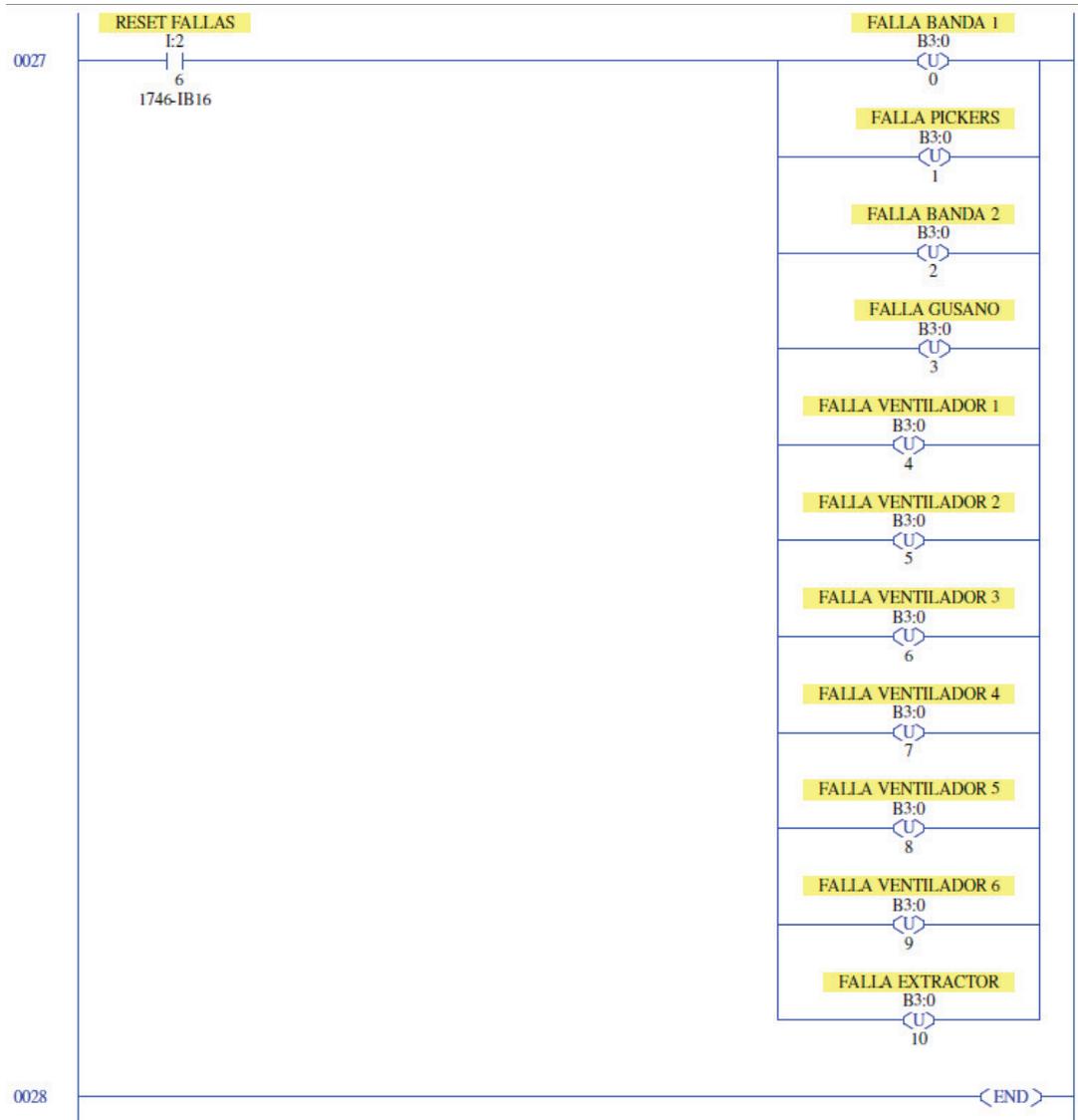
En esta parte del programa me gustaría detenerme a explicar algunas cosas. La primera, cuando se tienen señales analógicas estas llegan al PLC en forma de una señal eléctrica 4-20mA, el PLC la interpreta como un valor entero entre 0 a 32767, y dependiendo de la casa fabricante este valor puede variar. Para usarla como un valor entendible debe ser “escalada”, esto significa hacer coincidir el valor de entrada con un valor entendible de proceso. Por ejemplo, cuando un sensor detecta 0% de humedad relativa su salida por lo general será – dependiendo de la configuración del sensor – de 4mA. Esto, el PLC lo vería como un 3276. Así pues, es necesario “escalar” la señal para que un 3276 sea un 0% de RH%. Otra cosa que quiero explicar es la función PID. Arriba dimos la teoría de la función PID, aquí se utilizó como control para la compuerta de vapor. La salida de la función PID se aloja en la variable N7:1 que es un valor entero, este valor luego es escalado hacia la salida analógica para que corresponda la salida del PID con el valor de la salida analógica deseado. La tabla de constantes a modificar se muestra, aquí  $K_p = 0.5$ ,  $K_i = 0.5/1.0$ ,  $K_d = 0.5*1.00$  según la teoría vista.

Figura 10. Configuración de PID.





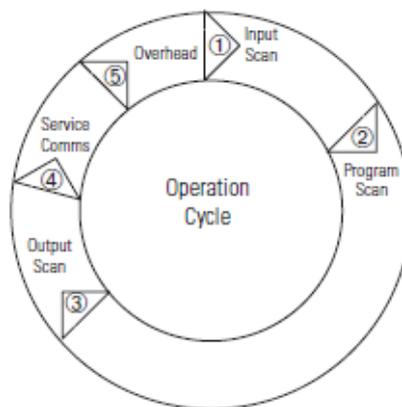




Vamos a explicar algunas cuestiones básicas acerca de los programas de PLC ya que tenemos el ejemplo anterior. Como todo programa, este tiene un inicio de secuencia y un fin. El programa se puede dividir en rutinas y tareas periódicas (dependiendo de la capacidad del PLC seleccionado). En el programa cada línea se llama “RUNG” y se identifica con un número. El número de identificación es el que precede a cada línea, como ejemplo, la línea del fin del programa es la RUNG 28.

El programa respeta un ciclo de operación, es decir, cuando lee las entradas, las salidas, cuando ejecuta el programa. Como en todo CPU, esto tiene un tiempo de ejecución:

Figura 11. Orden de los ciclos en PLC.



Se puede ver entonces que el primer paso del PLC es leer el estado de las entradas. Este estado pasa al archivo o espacio de memoria reservado para ser una imagen del estado físico de las entradas.

El segundo paso es la ejecución del programa. Aquí se llevan a cabo la interpretación y ejecución del programa, empezando siempre de izquierda a derecha y de arriba hacia abajo. Es decir, se ejecuta siempre primero el RUNG menor, pues la programación Ladder es secuencial. Dentro de este tiempo caben también las rutinas que son periódicas y las que son instanciadas desde el programa principal. Estas dos son útiles en programas muy grandes o para dividir el programa por áreas de procesos. El programa principal es obligatorio en los PLC, debe haber siempre una rutina principal desde la cual comenzar a ejecutar el programa.

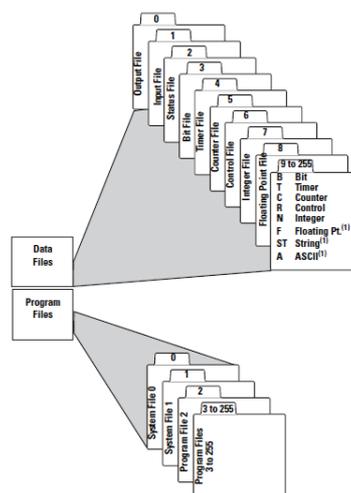
Luego de ejecutar el programa, todas las salidas toman el valor físico que les corresponde según el valor de su tabla imagen de datos de salidas.

El siguiente paso es ejecutar los servicios de comunicaciones en el caso de que el PLC posea algún puerto para comunicación. Se copia o se llena el buffer de comunicaciones según la tabla de datos respectiva.

Y luego, el último paso antes de volver a iniciar toda la secuencia, es un período llamado de gastos generales, que sirve para gestión de memoria del PLC, que es cuando las variables del área de datos en memoria toman el valor que les fue dado según el último SCAN del programa.

Este ciclo ha sufrido cambios y mejoras dependiendo del fabricante y de cuan nuevo es un equipo.

Figura 12. Ilustración para el ordenamiento de memoria en un PLC.



El área de memoria de un PLC (la descripción que se da es general y puede variar dependiendo del fabricante) está dividida en área de programa y área de datos. Ambas son de tipo RAM y guardan (como bien se entiende de su

nombre) el programa del PLC y la otra los datos de uso. Para esta última área es especialmente necesaria una batería instalada en el PLC.

El área de datos es el espacio de memoria donde se tienen los datos de uso, es decir, las variables de tipo bit, entero, flotante, etc., e incluso el espacio de memoria de la imagen del estado de las entradas y salidas.

Como se puede observar al evaluar de nuevo el programa ejemplo, además de programar el control para encendido de los motores, bandas y ventiladores y el control de RH%, se agregaron algunas funciones estándar en la programación industrial de este tipo de procesos. Dentro de esto me refiero a la parte de fallas de los sistemas. La falla de cada motor se levanta, cuando al respectivo motor se le da la orden de encender y la señal de confirmación de arranque no llega dentro de un determinado tiempo (10 segundos en este ejemplo). El bit asignado para dicha falla queda encendido permanentemente hasta que se restablezca la condición de falla y se presione el botón Reset de Fallas.



### **3 REDES DE DATOS, TOPOLOGÍA DE REDES Y ETHERNET**

#### **3.1 Redes de Datos o Networking**

Networking es la palabra técnica utilizada para describir mejor el concepto de red, esto es, la interconexión de estaciones de trabajo y dispositivos periféricos. De esta manera, es posible que distintos tipos de dispositivos se comuniquen entre si, sin importar el tipo, con el único requisito de “hablar” el mismo lenguaje, o sea, que tengan el mismo protocolo. La necesidad de Networking surgió de resolver 3 problemas principales: 1) Duplicidad de equipos y recursos. 2) Incapacidad de comunicación eficiente. 3) Falta de administración de red.

Fue así como surgieron los conceptos de LAN y WAN.

LAN: Una LAN conecta estaciones de trabajo, dispositivos periféricos, terminales y otros dispositivos en un área local (LAN = Local Area Network). Como resultado una empresa puede usar una LAN para unificar sus datos, sistemas de comunicaciones, equipos y servidores de archivos. Una LAN está diseñada para operar dentro de un área geográfica limitada, permitir que varios usuarios accedan a los mismos recursos con un ancho de banda elevado, proporcionar conectividad continua con los servicios locales, y para conectar dispositivos adyacentes.

WAN: Mientras una empresa crece, crece también así el número de equipos que utiliza y por supuesto su espacio geográfico. De esta manera el concepto de LAN se queda corto para poder comunicarse eficientemente, y cada LAN o departamento se convierte en una isla electrónica. Para eliminar este problema

se idearon las WAN que son redes de área ancha, o redes para un área geográfica extensa.

Figura 13. Ejemplo de Red LAN, WAN e Internet.

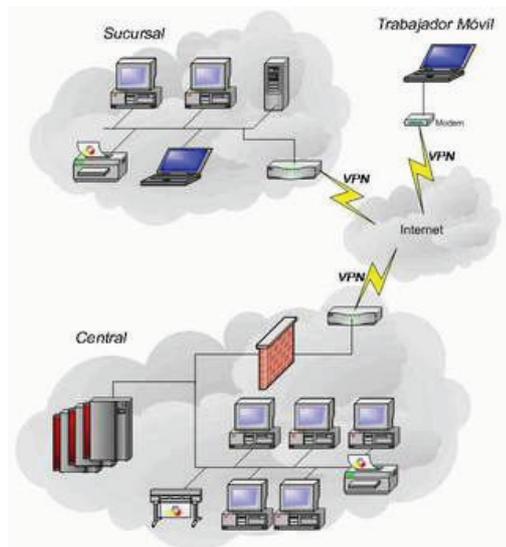
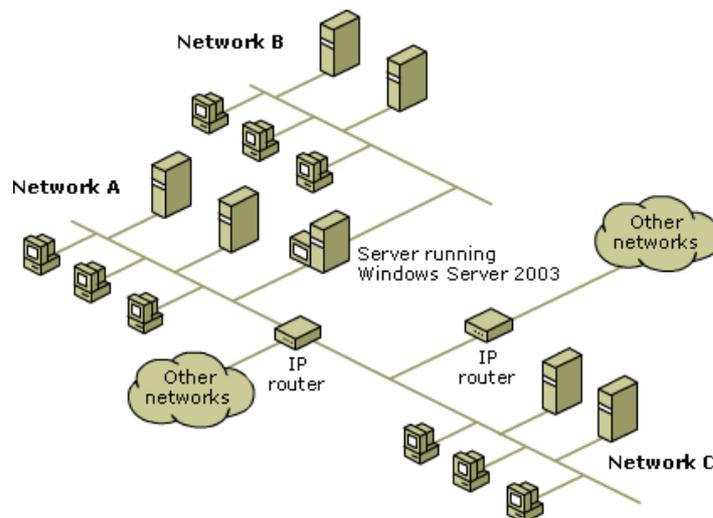


Figura 14. Ejemplo de Internetworking.

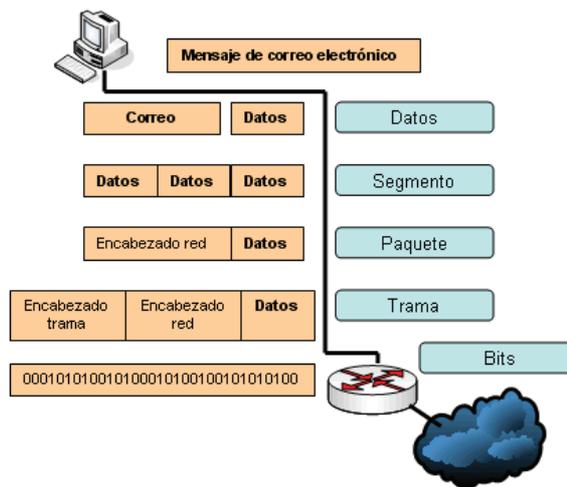


Así, mientras la necesidad de Networking crecía, también aumentaban los fabricantes de productos para redes. Y se hizo necesario el establecimiento de estándares. Para solucionar el problema de la incompatibilidad de las redes que no podían comunicarse entre si, la Organización Internacional para la normalización ISO analizó diversos diseños de redes. Así surgió en 1984 el modelo de referencia OSI.

El modelo de referencia OSI es un esquema de red descriptivo. Estos estándares aseguran una mayor compatibilidad e interoperabilidad entre distintos tipos de tecnología de red. Además, el modelo de referencia OSI describe la forma en que la información fluye a través de las redes. Es una estructura conceptual que especifica las funciones de red que se producen en cada capa. El modelo OSI describe la forma en que la información o los datos se trasladan desde programas de aplicación (como hojas de cálculo) a través de un medio de red (como cables) hasta otro programa de aplicación ubicado en otro equipo de una red. A medida que la información que se debe enviar desciende a través de las capas de un determinado sistema, el lenguaje utilizado se parece cada vez menos al lenguaje humano y cada vez más a los códigos uno y cero del lenguaje binario.

El modelo OSI divide el problema del traslado de información entre equipos a través de un medio de red en siete problemas más sencillos. Esta división de problemas se hizo debido a que estos siete problemas son relativamente independientes entre si, y por lo tanto más sencillos de resolver sin depender tanto de la información externa. Esto es lo que se conoce como capas. Así el modelo OSI se le conoce como el modelo de siete capas de OSI.

Figura 15. La información cambia de forma al pasar por un medio de red de aplicación a aplicación.



En la figura se indican los siete problemas distintos que se resuelven con el modelo OSI, es decir, las siete capas del modelo OSI:

Como se puede observar en la gráfica de capas, las primeras tres capas controlan la transmisión física de mensajes a través de la red, por esto se les conoce como capas de medio. Por otro lado, las capas superiores del modelo de referencia OSI se encargan de la transmisión precisa de datos entre equipos de la red, por lo cual se les llama capas de Host. La mayoría de los equipos de red implementan las siete capas aunque para organizar las operaciones algunas implementaciones de red incorporan funciones de varias capas a la vez.

Hay que entender una cosa, el modelo de referencia OSI no es una implementación de red, sino que especifica las funciones de cada capa. En el modelo OSI hay siete capas enumeradas que indican las distintas funciones de red. Si la red se divide en estas siete capas, se obtienen las siguientes ventajas:

Figura 16. Modelo de referencia OSI.



- ✓ Se dividen los aspectos interrelacionados del funcionamiento de la red en elementos menos complejos.
- ✓ Se definen las interfaces estándar para la compatibilidad plug and play y la integración de varios fabricantes.
- ✓ Permite que los ingenieros especialicen el diseño y promuevan la simetría en las distintas funciones modulares de Internetworking de redes de modo que interoperen entre sí.
- ✓ Impide que los cambios que se producen en un área afecten a las demás, para que cada área pueda evolucionar más rápidamente.
- ✓ Divide la complejidad de Internetworking en subconjuntos de operación separados, de aprendizaje más sencillo.

Ahora que hemos descrito las características básicas del modelo de división en capas OSI, se puede describir cada capa individual del OSI y sus funciones. Cada capa posee un conjunto predeterminado de funciones que debe ejecutar para que se produzca la comunicación. Las funciones se describen en las siguientes líneas.

Capa de aplicación: es la capa OSI más cercana al usuario; esta capa brinda servicios de red a las aplicaciones del usuario. Esta es distinta de las demás en el sentido de que no brinda servicios a ninguna otra capa, sino a procesos de aplicación que se ejecutan fuera del alcance del modelo OSI. Algunos ejemplos de esta capa serían los programas de hojas de cálculo, procesadores de texto, etc. Esta capa identifica y establece la disponibilidad de los diversos elementos que deben participar en la comunicación, sincroniza las aplicaciones que cooperan entre si y establece los procedimientos para la recuperación de errores y control de la integridad de los datos. También determina si existen suficientes recursos para la comunicación planificada.

Capa de presentación: asegura que la capa de aplicación de un sistema pueda leer la información enviada por la capa de aplicación de otro sistema. De ser necesario, la capa de presentación realiza una traducción entre varios formatos de representación de datos utilizando un formato de representación de datos común.

Capa de sesión: establece, administra y pone fin a las sesiones entre aplicaciones. Las sesiones son diálogos entre dos o más entidades de presentación. Brinda servicios a la capa de presentación. Además sincroniza el dialogo entre las entidades de la capa de presentación y administra el intercambio de datos. También tiene a su cargo la regulación básica de las

conversaciones o sesiones. La capa de sesión proporciona los recursos para la sincronización de unidades de diálogo, clase de servicio e informes de excepciones relacionados con problemas de las capas de sesión, de presentación y de aplicación.

Capa de transporte: segmenta y reensambla los datos en un flujo de datos. Mientras que las capas superiores están relacionadas con asuntos de aplicación, las cuatro inferiores se encargan del transporte de datos. Esta capa intenta suministrar un servicio de transporte de datos que proteja las capas superiores de los detalles de implementación de transporte. Específicamente, la capa de transporte se ocupa de temas como la confiabilidad del transporte a través de una Internetworking de redes. Al suministrar un servicio confiable, la capa de transporte proporciona mecanismos para el establecimiento, mantenimiento y finalización ordenada de los circuitos virtuales, la detección y recuperación de fallos de transporte y el control del flujo de información, para evitar que un sistema desborde con otros datos.

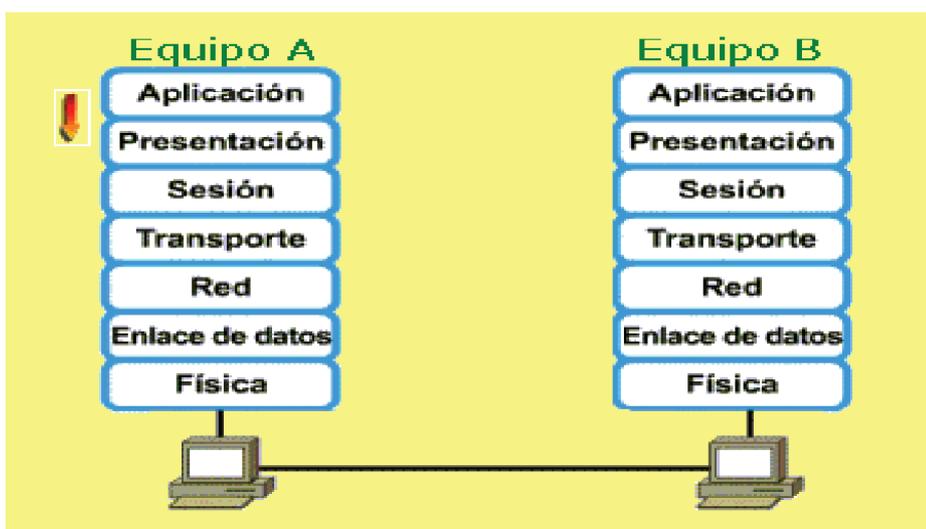
Capa de red: esta es una capa compleja que proporciona conectividad y selección de ruta entre dos sistemas finales, que pueden estar ubicados en redes geográficamente distintas. El mejor ejemplo de lo que hace la capa tres es un Router, que proporciona enrutamiento punto a punto, transmitiendo paquetes de datos y enrutando el tráfico entre distintas redes, basándose en el protocolo de red o en la información de esta capa. Estos tienen la capacidad de tomar decisiones con respecto a cuál es la mejor ruta para la entrega de datos en la red, se resuelve el problema del exceso de tráfico de broadcast o difusión. Aquí pues, se da el enrutamiento lógico. En esta capa se utilizan las direcciones de capa de red denominadas IP (Internet Protocol) o direcciones lógicas. Como las direcciones IP se implementan en software y se refieren a la red en la que

está ubicado un dispositivo, a veces, a estas direcciones de la capa 3 se les llama direcciones de protocolo o direcciones red.

Capa de enlace: ofrece un tráfico confiable de datos a través de un enlace físico. Para hacerlo se ocupa del direccionamiento físico (o direcciones MAC distinto al direccionamiento lógico o direcciones IP, que son direcciones de software), la topología de la red, la disciplina de la línea, la notificación de errores, la entrega ordenada de tramas y el control de flujo.

Capa física: define las especificaciones eléctricas, mecánicas, de procedimiento y funcionales para activar, mantener y desactivar el enlace físico entre sistemas finales. Las características tales como niveles de tensión, sincronización de cambios de tensión, velocidad de datos físicos, distancias de transmisión máximas, conectores físicos y otros atributos similares se definen a través de las especificaciones de la capa física.

Figura 17. La comunicación no se da de igual a igual directamente sino que pasa por todas las capas del modelo.

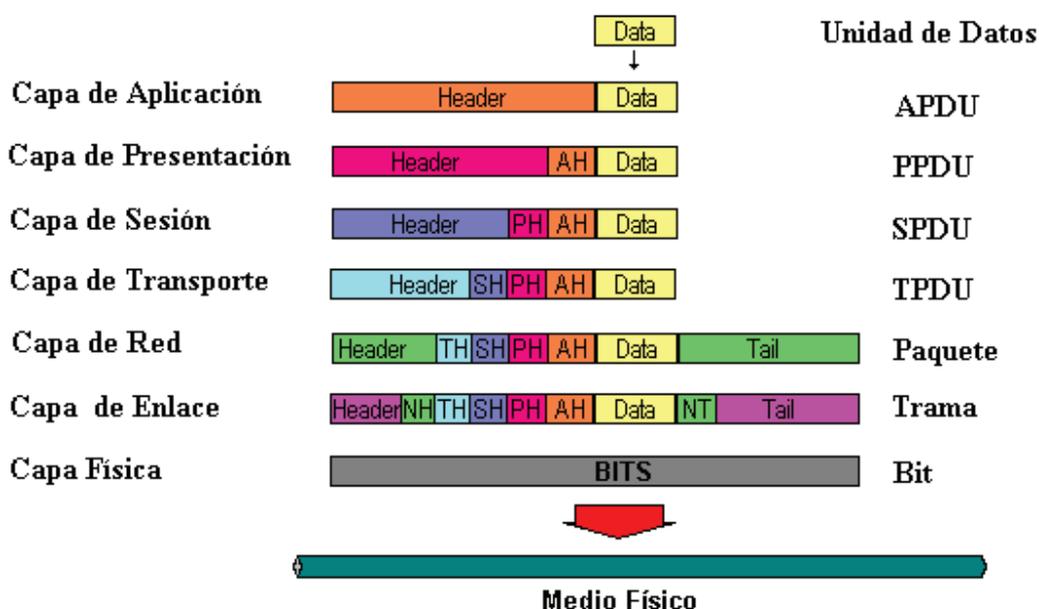


Tal como se ha venido diciendo, el modelo OSI no permite la posibilidad de una comunicación directa entre capas iguales de distintos sistemas. Para ejecutar estas tareas, una capa debe comunicarse con su capa igual del otro sistema. El protocolo de cada capa intercambia información, denominada PDU o unidades de datos de protocolo, entre capas iguales. Una determinada capa puede utilizar un nombre específico para su PDU.

Para comprender la estructura de las redes y su funcionamiento, se debe tener en cuenta que todas las comunicaciones en una red se originan en una fuente y se envían a un destino. La información que se envía a través de una red se denomina datos o paquetes de datos. Si un equipo desea enviar datos a otro equipo, en primer lugar los datos deben colocarse en paquetes mediante un proceso denominado encapsulación, que “envuelve” los datos en un encabezado de protocolo particular antes del tránsito en la red.

Cada capa depende de la función de servicio de la capa OSI inferior. Para brindar este servicio, la capa inferior utiliza la encapsulación para colocar la PDU de la capa superior en su campo de datos, luego le puede agregar cualquier encabezado e información final que utiliza la capa para ejecutar su función. Posteriormente, a medida que los datos se desplazan hacia abajo a través de las capas del modelo OSI, se agregan encabezados y la información al final.

Figura 18. Ilustración para el encapsulado de datos.



A medida que las redes ejecutan servicios para usuarios, el flujo y la organización de los paquetes de información varían. Básicamente, podemos enumerar los pasos del encapsulado de la siguiente forma:

Crear los datos: cuando un usuario envía un mensaje de correo electrónico, los caracteres alfanuméricos del mensaje se convierten en datos que pueden trasladarse a través de la internetwork.

Empaquetar los datos para transportarlos de punta a punta: los datos se empaquetan para transportarlos por Internetworking de redes. Al utilizar segmentos, la función de transporte asegura que los hosts del mensaje en ambos extremos del sistema de correo electrónico se puedan comunicar de forma confiable.

Adjuntar la dirección de red en el encabezado: los datos se colocan en un paquete o datagrama, que contiene un encabezado de red con direcciones

lógicas de fuente y destino. Estas direcciones ayudan a los dispositivos de red a enviar los paquetes a través de la red por una ruta seleccionada.

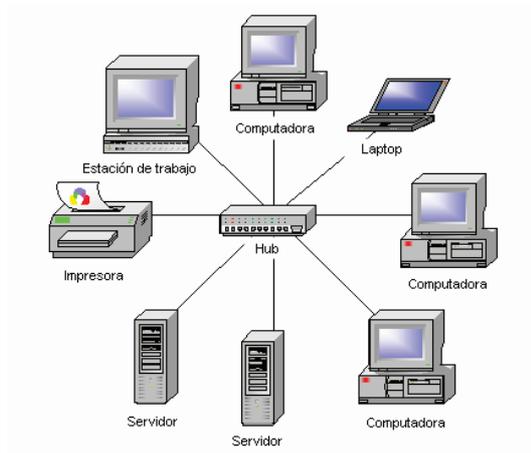
Adjuntar la dirección local en un encabezado de enlace de datos: cada dispositivo de red debe colocar el paquete en una trama. La trama le permite conectarse al próximo dispositivo de red conectado directamente en el enlace. Cada dispositivo en la ruta de red seleccionada requiere un entramado para conectarse al próximo dispositivo.

Convertir en bits para la transmisión: la trama debe convertirse en un patrón de unos y ceros para la transmisión en el medio. Una función de temporización permite que los dispositivos distinguan estos bits a medida que se trasladan por el medio. El medio en la Internetworking de redes físicas puede variar a lo largo de la ruta utilizada.

### **3.2 Topología de Redes**

Topología de red es la forma en que se distribuyen los cables de la red para conectarse con el servidor y con cada una de las estaciones de trabajo, es decir, la disposición de la red y los equipos. La topología de una red es similar a un plano de la red dibujado en un papel, ya que se pueden tender cables a cada estación de trabajo y servidor de la red. La topología determina donde pueden colocarse las estaciones de trabajo, la facilidad con que se tenderá el cable y el corte de todo el sistema de cableado. La flexibilidad de una red en cuanto a sus necesidades futuras se refiere, depende en gran parte de la topología establecida. Existen varias formas o tipos de topología:

Figura 19. Topología estrella.



### 3.2.1 Topología estrella

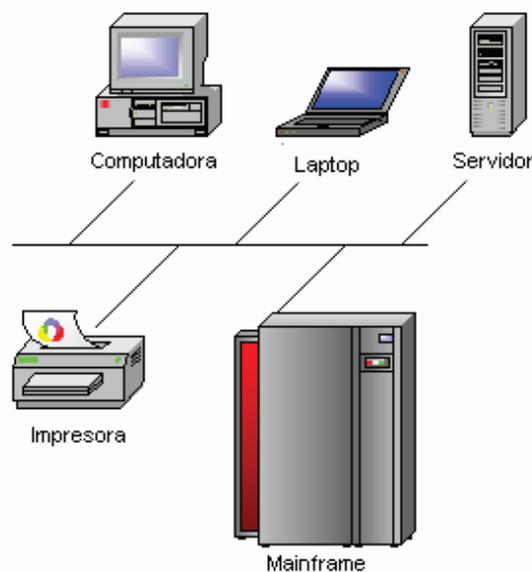
Se utiliza un dispositivo como punto de conexión de todos los cables que parten de las estaciones de trabajo. El dispositivo central puede ser el servidor de archivos en sí o un dispositivo especial de conexión. Ej: Starlan de AT&T. El diagnóstico de problemas es fácil, debido a que las estaciones de trabajo se comunican a través del equipo central. Los fallos en el nodo central son fáciles de detectar y es fácil cambiar los cables. La colisión entre datos es imposible, ya que cada estación tiene su propio cable, y resulta fácil ampliar el sistema. En algunas empresas tienden a agruparse los cables en la unidad central lo cual puede ocasionar errores de gestión.

### 3.2.2 Topología Bus

El servidor y todas las estaciones están conectados a un cable general central. Todos los nodos comparten este cable y éste necesita acopladores en ambos extremos. Las señales y los datos van y vienen por el cable, asociados a

una dirección destino. Cada nodo verifica las direcciones de los paquetes que circulan por la red para ver si alguna coincide con la suya propia. El cable puede extenderse de cualquier forma por las paredes y techos de la instalación. Ej: Ethernet y G-Net.

Figura 20. Topología bus.



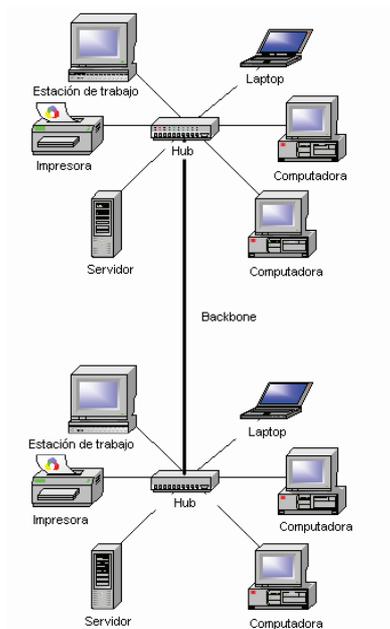
La topología bus usa una cantidad mínima de cable y el cable es muy fácil de instalar, ya que puede extenderse por un edificio en las mejores rutas posibles. Así el cable debe ir de equipo en equipo.

Las principales desventajas son: El cable central puede convertirse en un cuello de botella en entornos con un tráfico elevado, ya que todas las estaciones de trabajo comparten el mismo cable. Es difícil aislar los problemas de cableado en la red y determinar que estación o segmento de cable los origina, ya que todas las estaciones están en el mismo cable. Una rotura de cable hará caer el sistema.

### 3.2.3 Topología Estrella/Bus

Es una configuración combinada. Aquí un multiplexor de señal ocupa la posición del dispositivo central. El sistema de cableado de la red puede tomar la topología bus o anillo. Esto ofrece ventajas en el cableado de edificios que tienen grupos de trabajo separados por distancias considerables. Ej: ARCNET. Ofrece gran flexibilidad para configurar la distribución de los cables y adaptarla a cualquier edificio.

Figura 21. Topología bus-estrella.

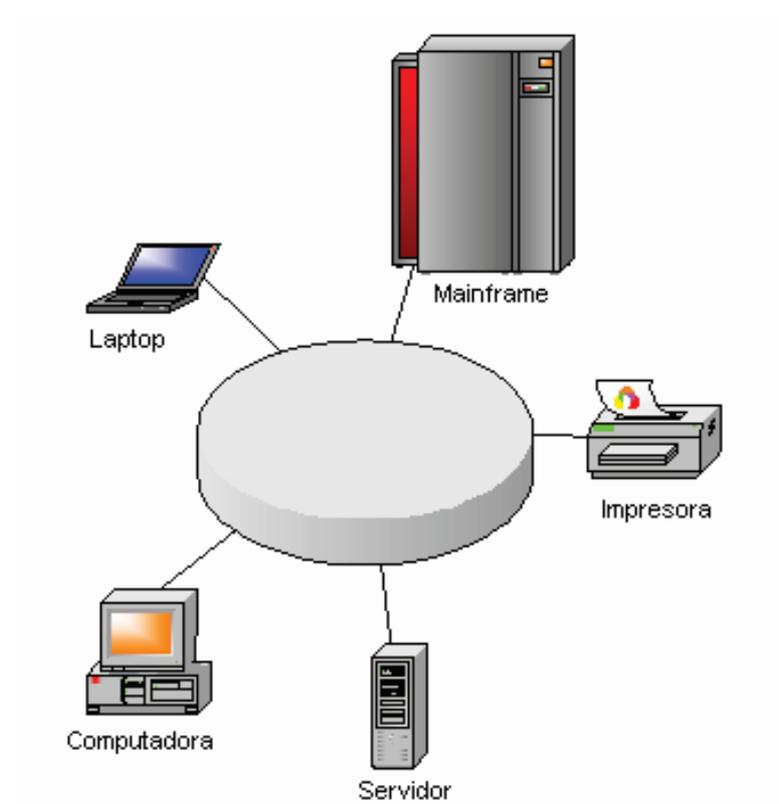


### 3.2.4 Topología Anillo

Las señales viajan en una única dirección a lo largo del cable en forma de un bucle cerrado. En cada momento, cada nodo pasa las señales a otro nodo.

Con la topología en anillo, las redes pueden extenderse a menudo a largas distancias, y el coste total del cableado será menor que en una configuración en estrella y casi igual a la bus. Una rotura del cable hará caer el sistema. Actualmente existen sistemas alternativos que evitan que esto ocurra.

Figura 22. Topología anillo.



### 3.2.5 Topología Estrella /Anillo

Existe un conector central. Las estaciones de trabajo se extienden a partir de este conector para incrementar las distancias permitidas. Ej: Token Ring de IBM

### 3.3 Ethernet

Es fácil confundir el concepto de Ethernet con muchos conceptos diferentes en cuanto a redes, pues se utiliza casi indistintamente para referirse a redes por su gran aceptación como el estándar en redes. Es Ethernet entonces un estándar de redes de computadoras de área local con acceso al medio por contienda CSMA/CD Acceso Múltiple con Detección de Portadora y Detección de Colisiones. Ethernet define las características de cableado y señalización de nivel físico y los formatos de tramas de datos del nivel de enlace de datos del modelo OSI. La Ethernet se tomó como base para la redacción del estándar internacional IEEE 802.3. Usualmente se toman Ethernet e IEEE 802.3 como sinónimos. Ambas se diferencian en uno de los campos de la trama de datos. Las tramas Ethernet y IEEE 802.3 pueden coexistir en la misma red.

Dentro de la especificación a nivel de capa 1, Ethernet es probablemente el esquema de red más usado, y el cable Ethernet de par trenzado es un sistema económico y fácil de instalar. Requiere de los siguientes componentes de hardware:

- Tarjeta de red con un conector hembra RJ-45
- Conector RJ-45
- Cable Ethernet de par trenzado conocido también como UTP y se divide en categorías dependiendo del nivel de blindaje.
- Concentrador

Figura 23. Tarjeta de red.



Una vez instalada la tarjeta de red y conectado el cableado al concentrador e instalado el software de red, el sistema quedará configurado.

A nivel de capa de enlace la trama de Ethernet tiene la siguiente forma:

Tabla IV. Trama de Ethernet.

<b>Trama DIX Ethernet</b>	<b>Preámbulo</b>	<b>Destino</b>	<b>Origen</b>	<b>Tipo</b>	<b>Datos</b>	<b>Relleno</b>	<b>FCS</b>
	8 bytes	6 bytes	6 bytes	2 bytes	0 a 1500 bytes	0 a 46 bytes	2 ó 4 bytes

Preámbulo: Un campo de siete bytes (56 bits) con una secuencia de bits usada para sincronizar y estabilizar el medio físico antes de iniciar la transmisión de datos. El patrón del preámbulo es:

10101010 10101010 10101010 10101010 10101010 10101010  
10101010

Estos bits se transmiten en orden, de izquierda a derecha y en la codificación Manchester representan una forma de onda periódica.

SOF (Start Of Frame) Inicio de Trama: Campo de un byte (8 bits) con un patrón de unos y ceros alternados y que termina con dos unos consecutivos. El patrón del SOF es: 10101011. Indica que el siguiente bit será el bit más significativo del campo de dirección MAC de destino. Aunque se detecte una colisión durante la emisión del preámbulo o del SOF, el emisor debe continuar enviando todos los bits de ambos hasta el fin del SOF.

Dirección de destino: Campo de seis bytes (48 bits) que especifica la dirección MAC de tipo EUI-48 hacia la que se envía la trama. Esta dirección de destino puede ser de una estación, de un grupo multicast o la dirección de broadcast de la red. Cada estación examina este campo para determinar si debe aceptar la trama (si es la estación destinataria).

Dirección de origen: Campo de seis bytes (48 bits) que especifica la dirección MAC de tipo EUI-48 desde la que se envía la trama. La estación que deba aceptar la trama conoce por este campo la dirección de la estación origen con la cual intercambiará datos.

Tipo: Campo de dos bytes (16 bits) que identifica el protocolo de red de alto nivel asociado con la trama o, en su defecto, la longitud del campo de datos. La capa de enlace de datos interpreta este campo. (En la IEEE 802.3 es el campo longitud y debe ser menor o igual a 1526 bytes.)

Datos: Campo de cero a 1500 Bytes de longitud. Cada Byte contiene una secuencia arbitraria de valores. El campo de datos es la información recibida del nivel de red (la carga útil). Este campo, también incluye los H3 y H4 (cabeceras de los niveles tres y cuatro), provenientes de niveles superiores.

Relleno: Campo de cero a 46 bytes que se utiliza cuando la trama Ethernet no alcanza los 64 bytes mínimos para que no se presenten problemas de detección de colisiones cuando la trama es muy corta.

FCS (Frame Check Sequence - Secuencia de Verificación de Trama): Campo de 32 bits (4 bytes) que contiene un valor de verificación CRC (Control de redundancia cíclica). El emisor calcula el CRC de toda la trama, desde el campo destino al campo CRC suponiendo que vale 0. El receptor lo recalcula, si el valor calculado es 0 la trama es válida.

### **3.4 Direcciones IP**

Como se mencionó anteriormente, el direccionamiento IP se da en la capa tres del modelo de referencia OSI y es necesario identificar a cada dispositivo de red para que pueda ser localizado y establecer una comunicación abierta y sin choques de información o ambigüedad de equipos en una misma red. Aunque existe el direccionamiento físico (MAC) el más común para utilizar dentro de una red por la versatilidad que otorga es el direccionamiento lógico o IP. Los equipos se identifican a través de la red en la capa tres mediante el protocolo IP (Protocolo de Internet). Este protocolo utiliza direcciones numéricas compuestas por cuatro números enteros (cuatro bytes) entre cero y 255, escritos en el formato xxx.xxx.xxx.xxx. Por ejemplo, 194.153.205.26 es una dirección IP.

Los equipos de una red utilizan estas direcciones para comunicarse, de manera que cada equipo de la red tiene una dirección IP exclusiva. El organismo a cargo de asignar direcciones públicas de IP, es decir, direcciones IP para los equipos conectados directamente a la red pública de Internet, es el

ICANN (Internet Corporation for Assigned Names and Numbers) que reemplaza el IANA desde 1998 (Internet Assigned Numbers Agency).

Una dirección IP es una dirección de 32 bits, escrita generalmente con el formato de cuatro números enteros separados por puntos. Una dirección IP tiene dos partes diferenciadas:

1. Los números de la izquierda indican la red y se les denomina netID (identificador de red).
2. Los números de la derecha indican los equipos dentro de esta red y se les denomina host-ID (identificador de host).

Tomemos una red escrita 58.0.0.0. Los equipos de esta red podrían tener direcciones IP que van desde 58.0.0.1 a 58.255.255.254. Por lo tanto, se trata de asignar los números de forma que haya una estructura en la jerarquía de los equipos y los servidores.

Cuanto menor sea el número de bits reservados en la red, mayor será el número de equipos que puede contener. De hecho, una red escrita 102.0.0.0 puede contener equipos cuyas direcciones IP varían entre 102.0.0.1 y 102.255.255.254 ( $256 \times 256 \times 256 - 2 = 16,777,214$  posibilidades), mientras que una red escrita 194.26.0.0 puede contener solamente equipos con direcciones IP entre 194.26.0.1 y 194.26.255.254 ( $256 \times 256 - 2 = 65,534$  posibilidades); ésta es el concepto de clases de direcciones IP.

### **3.4.1 Clases de redes**

Las direcciones de IP se dividen en clases, de acuerdo a la cantidad de bytes que representan a la red.

Clase A: En una dirección IP de clase A, el primer byte representa la red. El bit más importante (el primer bit a la izquierda) está en cero, lo que significa que hay  $2^7$  (00000000 a 01111111) posibilidades de red, que son 128 posibilidades. Sin embargo, la red 0 (bits con valores 00000000) no existe y el número 127 está reservado para indicar su equipo. Las redes disponibles de clase A son, por lo tanto, redes que van desde 1.0.0.0 a 126.0.0.0 (los últimos bytes son ceros que indican que se trata seguramente de una red y no de equipos). Los tres bytes de la derecha representan los equipos de la red. Por lo tanto, la red puede contener una cantidad de equipos igual a:  $2^{24}-2 = 16,777,214$  equipos. En binario, una dirección IP de clase A luce así:

```
0 xxxxxxx xxxxxxx xxxxxxx xxxxxxx
Red      Equipos
```

Clase B: En una dirección IP de clase B, los primeros dos bytes representan la red. Los primeros dos bits son uno y cero; esto significa que existen  $2^{14}$  (10 000000 00000000 a 10 111111 11111111) posibilidades de red, es decir, 16,384 redes posibles. Las redes disponibles de la clase B son, por lo tanto, redes que van de 128.0.0.0 a 191.255.0.0. Los dos bytes de la derecha representan los equipos de la red. La red puede entonces contener una cantidad de equipos equivalente a:  $2^{16}-2 = 65,534$  equipos. En binario, una dirección IP de clase B luce así:

```
10 xxxxxx xxxxxxx xxxxxxx xxxxxxx
Red          Equipos
```

Clase C: En una dirección IP de clase C, los primeros tres bytes representan la red. Los primeros tres bits son 1,1 y 0; esto significa que hay  $2^{21}$  posibilidades de red, es decir, 2,097,152. Las redes disponibles de la clases C

son, por lo tanto, redes que van desde 192.0.0.0 a 223.255.255.0. El byte de la derecha representa los equipos de la red, por lo que la red puede contener:  $2^8 - 2 = 254$  equipos. En binario, una dirección IP de clase C luce así:

```
110 Xxxxx XXXXXXXX XXXXXXXX XXXXXXXX
Red                Equipos
```

### 3.4.2 Asignación de direcciones IP

El objetivo de dividir las direcciones IP en tres clases A, B y C es facilitar la búsqueda de un equipo en la red. De hecho, con esta notación es posible buscar primero la red a la que uno desea tener acceso y luego buscar el equipo dentro de esta red. Por lo tanto, la asignación de una dirección de IP se realiza de acuerdo al tamaño de la red.

Clase	Cantidad de redes posibles	Cantidad de equipos en cada una
A	126	16777214
B	16384	65534
C	2097152	254

Las direcciones de clase A se utilizan en redes muy amplias, mientras que las direcciones de clase C se asignan, por ejemplo, a las pequeñas redes de empresas.

### 3.4.3 Máscara de Subred

Otro concepto profundamente ligado con las direcciones IP es el concepto de máscara de subred. El interés principal de una máscara de subred

reside en que permite la identificación de la red asociada con una dirección IP. Efectivamente, la red está determinada por un número de bytes en la dirección IP (un byte por las direcciones de clase A, dos por las de clase B y tres bytes para la clase C). Sin embargo, una red se escribe tomando el número de bytes que la caracterizan y completándolo después con ceros. Por ejemplo, la red vinculada con la dirección 34.56.123.12 es 34.0.0.0, porque es una dirección IP de clase A. Para averiguar la dirección de red vinculada con la dirección IP 34.56.123.12, simplemente se debe aplicar una máscara cuyo primer byte esté solamente compuesto por números uno (o sea 255 en decimal), y los siguientes bytes compuestos por ceros. La máscara es: 11111111.00000000.00000000.00000000

La máscara asociada con la dirección IP 34.208.123.12 es, por lo tanto, 255.0.0.0. El valor binario de 34.208.123.12 es: 00100010.11010000.01111011.00001100 De este modo, una operación lógica de AND entre la dirección IP y la máscara da el siguiente resultado:

```
00100010.11010000.01111011.00001100
AND
11111111.00000000.00000000.00000000
=
00100010.00000000.00000000.00000000
```

O sea 34.0.0.0 Esta es la red vinculada a la dirección 34.208.123.12

Generalizando, es posible obtener máscaras relacionadas con cada clase de dirección: Para una dirección de Clase A, se debe conservar sólo el primer byte. La máscara tiene el siguiente formato 11111111.00000000.00000000.00000000, es decir, 255.0.0.0 en decimales;

Para una dirección de Clase B, se deben retener los primeros dos bytes y esto da la siguiente máscara 11111111.11111111.00000000.00000000, que corresponde a 255.255.0.0 en decimales. Para una dirección de Clase C, siguiendo el mismo razonamiento, la máscara tendrá el siguiente formato 11111111.11111111.11111111.00000000, es decir, 255.255.255.0 en decimales.

## 4 CÓDIGO HTML Y ADQUISICIÓN DE DATOS

### 4.1 Código HTML

Como ya se mencionó en el capítulo 1, el código HTML es el más utilizado para la programación de páginas Web. A nosotros nos interesa más que la programación HTML en sí, la versatilidad y economía que tiene el presentar datos de esta manera. Pues un sistema de visualización industrial como un panel de operador puede llegar a tener precios que van de los USD\$300 hasta los USD\$15000 en algunos casos, sin contar el precio del software para programar dichos paneles o pantallas de operador ni la ingeniería para desarrollar la visualización. Al poder extraer los datos de proceso desde un PLC hacia una PC entonces puede hacerse un sistema de visualización de datos de bajo costo como es el caso de páginas de HTML, pues un explorador Web es un software freeware o de licencia de distribución gratuita y además sirve para crear páginas Web como vimos en el ejemplo sencillo del capítulo 1, para lo que tampoco tiene ningún costo la licencia.

No vamos a detenernos a ser un manual completo de programación HTML pero si vamos a presentar algunas herramientas que pueden ser útiles especialmente para nuestro fin. Podríamos decir entonces que una página web debe ser amigable a la vista y al uso, no tiene ningún sentido poner mucha información importante en una página si se escogieron mal la combinación de colores que hace imposible ver la página por más de cinco minutos, o hacer una página con todas las funciones que necesitas pero que sea imposible de entender el uso o necesario leer un manual de 500 páginas para poder usarla. Para entender esto vamos a presentar unas de las herramientas más comunes en programación Web.

Para nuestro propósito presentaremos como cambiar el fondo, tablas, y encabezados o textos.

#### 4.1.1 Fondos

Una imagen de fondo de una página Web puede aplicarse usando la etiqueta <BODY>:

Los colores se definen usando la notación #RRGGBB, donde RR, GG y BB representa cada uno un valor hexadecimal entre 00 y FF para rojo, verde y azul, respectivamente.

Tabla V. Formato de instrucciones de Fondo en HTML.

Atributo	Efecto visual
BACKGROUND="imagen"	Muestra la imagen como fondo
BGCOLOR="nombre_del_color o #XXXXXX"	Muestra el color especificado como fondo
VÍNCULO="color"	Establece el color de los hipervínculos
ALINK="color"	Establece el color del vínculo activo
VLINK="color"	Establece el color de los vínculos ya visitados
TEXTO="color"	Color del texto predeterminado

#### 4.1.2 Tablas

A menudo resulta útil presentar información de una manera más estructurada que en las listas. Las tablas permiten mostrar esta información en filas y columnas. Las tablas se definen como series de filas. Una tabla debe respetar las siguientes reglas:

- La tabla está enmarcada por las etiquetas <TABLE> y </TABLE>.
- El título de la tabla está enmarcado por <CAPTION> </CAPTION>
- Cada fila está enmarcada por <TR> </TR> (por Table Row).
- Las celdas de encabezamiento están enmarcadas por <TH> </TH> (por Table Header).

Los datos en cada celda están enmarcados por <TD> </TD> (Table Data).

Este es un ejemplo de tabla HTML:

```
<TABLE BORDER="1">
<CAPTION>Este es el título de la tabla</CAPTION>
<TR>
<TH> Título A1 </TH>
<TH> Título A2 </TH>
<TH> Título A3 </TH>
<TH> Título A4 </TH>
</TR>
<TR>
<TH> Título B1 </TH>
<TD> Valor B2 </TD>
<TD> Valor B3 </TD>
<TD> Valor B4 </TD>
</TR>
</TABLE>
```

Los resultados de este código:

Tabla VI. Formato de instrucciones de Tablas en HTML.

Este es el título de la tabla

<b>Título A1</b>	<b>Título A2</b>	<b>Título A3</b>	<b>Título A4</b>
<b>Título B1</b>	Valor B2	Valor B3	Valor B4

#### 4.1.3 Encabezados

HTML acepta seis encabezados para definir la estructura jerárquica de los párrafos en un documento:

Tabla VII. Formato de instrucciones de Encabezados en HTML.

Etiqueta	Efecto visual
H1	<b>Prueba</b>
H2	<b>Prueba</b>
H3	<b>Prueba</b>
H4	<b>Prueba</b>
H5	<b>Prueba</b>
H6	<b>Prueba</b>

Los siguientes atributos se ubican dentro de las etiquetas de estructura, de forma que los elementos pueden mostrarse en ubicaciones específicas:

Tabla VIII. Formato de atributos de instrucciones de Tablas en HTML.

Atributo	Valor	Efecto visual
ALIGN	LEFT	Texto alineado a la izquierda
	RIGHT	Texto alineado a la derecha
	CENTER	Texto centrado
	JUSTIFY	Texto justificado
NOWRAP		Evita que el navegador crea un salto de línea.
ID		Otorga una identificación a la etiqueta. Este atributo es especialmente útil para la programación con JavaScript
LANG		Especifica un idioma diferente
CLASS		Asigna una clase a los contenidos (para <a href="#">hojas de estilo</a> )
CLEAR		Se usa cuando el texto está rodeando una imagen. Ubica el texto tan abajo de la página como sea necesario para tener un margen claro.

Este es un ejemplo de encabezado:

```
<H1 ALIGN=JUSTIFY> Texto justificado </H1>
```

#### 4.1.4 Párrafos

HTML considera que los párrafos son bloques de texto. Los navegadores hacen lo posible para adaptar el contenido de los párrafos a la ventana, a menos que los atributos NOWRAP o NOBR estén explícitamente especificados. Dentro de un párrafo, todos los espacios, tabuladores y saltos de línea cuentan como un espacio simple. Para colocar textos en bloques, se utiliza el par de etiquetas <p> y </p>. Esta etiqueta puede incluir cualquiera de los atributos mencionados anteriormente. Los saltos de línea (saltar de una línea a la siguiente) se crean usando la etiqueta <br/>. La etiqueta <hr> se utiliza para insertar una línea horizontal.

Tabla IX. Formato de Párrafos.

Etiqueta	Atributo	Efecto visual
 		Salto de línea
<p> y </p>		Párrafo
<hr size=5 width=20% align=left >	SIZE WIDTH ALIGN	

Las etiquetas usadas para formatear texto se llaman contenedores.

Tabla X. Formato para los contenedores.

Contenedor	Efecto visual
<blockquote> y </blockquote>	Texto con indentación
<address> y </address>	<i>Para escribir información de contacto</i>
<note> y </note>	Para escribir una nota
<fn> y </fn>	Para crear una nota al pie
<banner> y </banner>	Hace que un bloque de texto adopte una posición fija con relación a la página
<pre> y </pre>	Para escribir texto preformateado; manteniendo espacios, saltos de líneas y tabuladores intactos

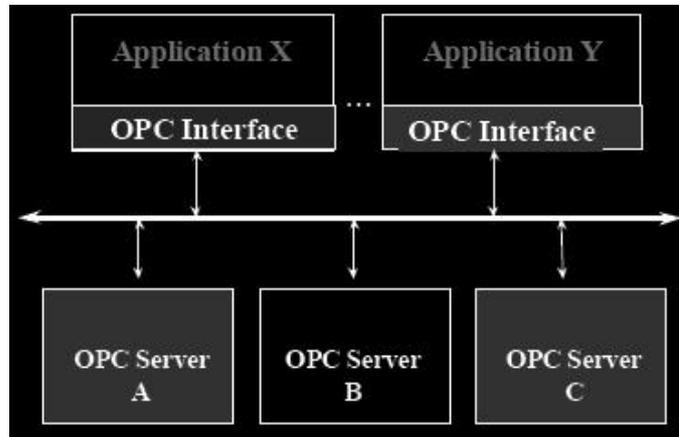
## 4.2 Adquisición de Datos: Controles ActiveX

La programación HTML no es complicada (fuera de cuidar la estética de la página para que sea siempre amigable a la vista y al uso) además que usando herramientas de programación Web se puede ahorrar todo esto y se convierte en una programación muy gráfica. Sin embargo, sí que es una programación muy versátil, por ejemplo pueden hacerse aplicaciones como operaciones bancarias, transacciones monetarias, compras en línea, juegos en línea, procesadores de palabras en línea, todo esto gracias a su versatilidad y

apertura a integrar trozos de código de terceros. Esta capacidad es precisamente la que aprovechamos para extraer datos de un PLC y llevarlos hasta una página Web. Vamos a ir directamente a explorar la forma en que vamos a llevar los datos desde el PLC y la forma en que los vamos a presentar en HTML. Para lograr extraer datos desde un PLC hay diferentes herramientas, como OPC o controles ActiveX. Como un ejemplo de OPC tenemos el software RSLinx que es la plataforma de comunicaciones de la marca Allen Bradley, un OPC es un servidor de datos de distintos servidores de datos hacia una o varias aplicaciones, de hecho OPC significa OLE for Process Control. Un OLE es un Object Linking and Embedding que fue concebido como una tecnología de Microsoft para compartir información entre distintas aplicaciones de Windows.

Un ActiveX en cambio es un trozo de código reutilizable u objeto que puede ser integrado en otros programas. No son programas ni aplicaciones completas en sí mismos, pues no corren por sí solos, pero sí pueden ser usados para armar bloques que pueden ser compartidos y reutilizados por diferentes paquetes de software. Un programador no necesita conocer la implementación interna de un control para poderlo integrar exitosamente en su ambiente de programación favorito. Un control correctamente diseñado ofrece una interface fácil de utilizar que consta de propiedades, métodos y eventos. Las propiedades son variables que pueden ser leídas o escritas por la aplicación y que frecuentemente se usan para especificar cómo debe comportarse el control. Los métodos son las funciones que ofrece el control y que pueden ser llamadas por la aplicación para controlar alguna operación. Los eventos son señales generadas por la aplicación para informar de eventos ocurridos dentro del control. Las propiedades, los métodos y eventos proporcionan comunicación de doble vía entre la aplicación y el control ActiveX.

Figura 24. Interfaz de OPC.



En este caso un servidor Web provee a un cliente que lo pida con tanto una página Web como cualquier cantidad de controles. La página Web no solo define la representación gráfica de la página sino también tiene código integrado en ella que controla al ActiveX para poder implementar funcionalidad específica.

Usar una página Web para la interacción con un PLC tiene varias ventajas:

El cliente sólo necesita estar conectado a la red y tener MS IE instalado para poder monitorear o controlar el PLC. El explorador de Internet IE es automática y gratuitamente instalado en los sistemas Windows.

Los PLC pueden proveer páginas Web y controles ActiveX a cualquier cliente usando el servidor Web integrado en ellos (en los que aplica, como el caso de los IPC de FESTO). Esto implica que un IPC puede proveer el software para controlar y monitorearse a sí mismo a un cliente, haciéndolo un dispositivo completamente autocontenido que no necesita de una PC en donde se instale un determinado y específico software.

Los controles ActiveX que son guardados en el IPC son comprimidos para minimizar el espacio en disco que ocupa, un control ActiveX comprimido generalmente usa 12Kb de espacio, mas las páginas web que se guardan en archivos de texto planos son fáciles y pequeñas para guardar en disco.

Es esta la herramienta que nosotros vamos a usar en este trabajo, los controles ActiveX necesarios a instalar son los siguientes (pueden conseguirse en la página de soporte de Festo):

**CommandX:** provee acceso remoto al intérprete de comandos del IPC, a través de esta interface la mayoría de funciones del IPC pueden ser controladas. Las respuestas a todos los comandos son enviadas de vuelta al control, lo que significa que este control puede ser usado para interrogar información contenida en el IPC

Tabla XI. Propiedades, Métodos y Eventos del comando CommandX.

Propiedades	
PacketCount	El número máximo de paquetes a enviar. El valor predeterminado es 3.
Interval	El tiempo a esperar antes de retransmitir un paquete en milisegundos.
Ip	La dirección IP del PLC al que se le envía el comando.
Métodos	
SendCommand()	Envía un paquete con comandos.
CancelCommand()	Detiene el envío de paquetes.
IsSending()	Consulta si se está enviando paquetes con comandos o no.
HasResult()	Consulta si se ha recibido respuesta al comando enviado.
GetResult()	Retorna el resultado recibido al comando enviado.
About()	Retorna información del control.
Eventos	
SendAttempt(cuenta)	Se ejecuta cuando se envía un paquete con comandos.
ResultReceived(resultado)	Se ejecuta cuando se ha recibido respuesta a un comando enviado.
Timeout()	Se ejecuta cuando no se ha recibido respuesta a un comando enviado después de cierto tiempo.





100 y 101), que luego son desplegadas en pantalla. El control está integrado en la página Web escrita en HTML. Los siguientes pasos son requeridos:

Primero debemos crear una nueva página Web, podemos usar la del ejemplo del capítulo uno y modificarla o crear una nueva de la misma manera presentada ahí.

Luego insertar el siguiente código:

```
<HTML>
<HEAD>
<TITLE>Pagina de demo del control CommandX ActiveX</TITLE>
<SCRIPT LANGUAGE="JavaScript">

function doUpdate()
{
    // enviar el comando
    CommandX.SendCommand();
}

</SCRIPT>

<SCRIPT LANGUAGE=javascript FOR=CommandX EVENT=ResultReceived>

// obtener la respuesta del ActiveX
var strResponse = CommandX.GetResult();

// la respuesta debe tener el formato =255=255=10=10
// dividir la respuesta en partes divididas por el signo = en arreglos
var arrParts = strResponse.split("=");

// escribir los resultados en pantalla
txtResult.style.color = "#000000";
txtResult.innerHTML = "Input 0: " + arrParts[1] + "<br>";
txtResult.innerHTML += "Output 0: " + arrParts[2] + "<br>";
txtResult.innerHTML += "Flag 100: " + arrParts[3] + "<br>";
txtResult.innerHTML += "Flag 101: " + arrParts[4];

// calendarizar la siguiente actualizacion en 500 ms
```

```

        setTimeout("doUpdate()", 500);

</SCRIPT>

<SCRIPT LANGUAGE=javascript FOR=CommandX EVENT=Timeout>

    // desplegar mensaje de tiempo de espera
    txtResult.style.color = "#ff0000";
    txtResult.innerHTML = "Timeout communicating with " + CommandX.Ip;

    // calendarizar la siguiente actualizacion en 500 ms
    setTimeout("doUpdate()", 500);

</SCRIPT>

</HEAD>
<BODY onload="doUpdate()">

<OBJECT id=CommandX classid=clsid:79D33D79-C967-44F4-BD75-970FC9C24A82>
  <PARAM NAME="Ip" VALUE="192.168.0.150">
  <PARAM NAME="Command" VALUE="dew0;daw0;dmw100;dmw101">
  <PARAM NAME="Interval" VALUE="250">
  <PARAM NAME="PacketCount" VALUE="4">
</OBJECT>

<SPAN id=txtResult style="font-family: Arial; font-size: 9pt"></SPAN>

</BODY>
</HTML>

```

Veamos el código con un poco más de detalle. La parte más importante puede ser hallada cerca del final del código, donde el control es integrado:

```

<OBJECT id=CommandX classid=clsid:79D33D79-C967-44F4-BD75-970FC9C24A82>
  <PARAM NAME="Ip" VALUE="192.168.0.150">
  <PARAM NAME="Command" VALUE="dew0;daw0;dmw100;dmw101">
  <PARAM NAME="Interval" VALUE="250">
  <PARAM NAME="PacketCount" VALUE="4">
</OBJECT>

```

Los controles ActiveX son identificados por su identificador global único, conocido como Class\_ID. En el código del ejemplo, el control de Class\_ID:

79D33D79-C967-44F4-BD75-970FC9C24A82

Que es el control CommandX. Las propiedades del control son especificadas de manera que envíe el comando dew0;daw0;dmw100;dmw101 a un IPC con la dirección 192.168.0.150. El control es también asignado al identificador CommandX, por el cual se le referirá dentro del código de la página.

Cerca del inicio de la página se encontrara un trozo de código integrado entre marcas <SCRIPT> y </SCRIPT>

```
<SCRIPT LANGUAGE="JavaScript">

function doUpdate()
{
    // envía el comando
    CommandX.SendCommand();
}

</SCRIPT>
```

El código define la función llamada doUpdate(). Cuando se invoca, esta función llama el método SendCommand() del control integrado CommandX.

La siguiente parte de código define un manejador para el evento ResultTeceived del control integrado:

```
</SCRIPT>

<SCRIPT LANGUAGE=javascript FOR=CommandX EVENT=ResultReceived>
```

```

// obtener la respuesta del ActiveX
var strResponse = CommandX.GetResult();

// la respuesta debe tener el formato =255=255=10=10
// dividir la respuesta en partes divididas por el signo = en arreglos
var arrParts = strResponse.split("=");

// escribir los resultados en pantalla
txtResult.style.color = "#000000";
txtResult.innerHTML = "Input 0: " + arrParts[1] + "<br>";
txtResult.innerHTML += "Output 0: " + arrParts[2] + "<br>";
txtResult.innerHTML += "Flag 100: " + arrParts[3] + "<br>";
txtResult.innerHTML += "Flag 101: " + arrParts[4];

// calendarizar la siguiente actualizacion en 500 ms
setTimeout("doUpdate()", 500);

</SCRIPT>

```

Quando el CommandX dispara un evento ResultReceived, este código regresa el resultado en forma de texto desde el control, lo divide en partes y despliega en resultado en la pantalla. Después de esto, una siguiente invocación de la función doUpdate() es calendarizada para ocurrir en 500ms. En caso de que se agote el tiempo de espera, el siguiente código es ejecutado, que despliega un mensaje de error y calendarizar otra actualización en 500 ms:

```

<SCRIPT LANGUAGE=javascript FOR=CommandX EVENT=Timeout>

// desplegar mensaje de tiempo de espera
txtResult.style.color = "#ff0000";
txtResult.innerHTML = "Timeout communicating with " + CommandX.Ip;

// calendarizar la siguiente actualizacion en 500 ms
setTimeout("doUpdate()", 500);

</SCRIPT>

```

El siguiente trozo de código llama al método doUpdate() cuando la página es cargada en el explorador, causando que el primer comando sea enviado:

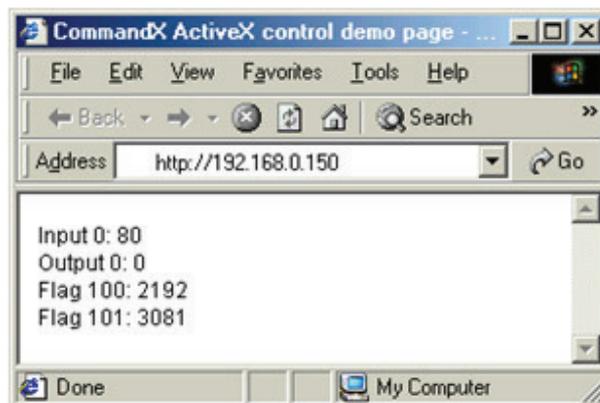
```
<BODY onload="doUpdate()">
```

Cuando el primer resultado es recibido o en un caso se agote el tiempo de espera, el método `doUpdate()` es recalendarizado por los manejadores de evento `ResultReceived` o por el tiempo de espera, causando que la página se actualice periódicamente. Finalmente el siguiente código HTML define un espacio en la página en el que los resultados del comando son escritos:

```
<SPAN id=txtResult style="font-family: Arial; font-size: 9pt"></SPAN>
```

El resultado final debería lucir como esto:

Figura 25. Página web de OPC Festo.



Hasta aquí se ha definido toda la teoría necesaria para la implementación del acceso, monitoreo y control del PLC del laboratorio de electroneumática. También se ha definido la teoría acerca de los controladores lógicos programables y un ejemplo de programación, así como la forma de la presentación de datos del PLC en una página Web valiéndonos de los controles ActiveX, y la forma de funcionamiento del medio de comunicaciones que se

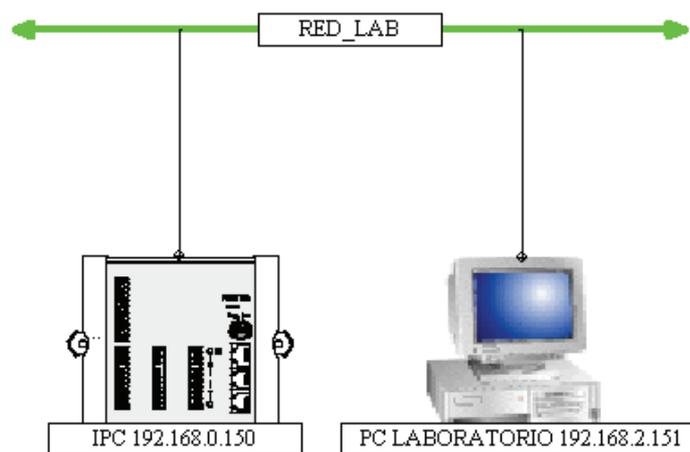
usará, en este caso una red Ethernet. Solo queda en el siguiente y último capítulo unificar toda la teoría y realizar la implementación del objetivo del presente trabajo de graduación.



## 5 DISEÑO DEL SISTEMA PARA EL ACCESO, MONITOREO Y CONTROL DEL PLC DEL LABORATORIO DE ELECTRONEUMÁTICA DESDE UNA PÁGINA WEB

Hasta este punto se ha expuesto toda la teoría necesaria para implementar el sistema en cuestión. Para explicar de manera básica este sistema consiste en un PLC, una computadora personal y la conexión de red entre ambos. El PLC ejecuta un programa simple que simula un proceso pequeño de secado, mientras que la computadora servirá para poder ver datos del PLC o proceso en tiempo real en forma de una página Web, y la comunicación entre ambos será por Ethernet.

Figura 26. Topología de red PLC - PC.



### 5.1 Página WEB

Para lograr el objetivo de mostrar los datos del PLC que interesan podría usarse un ejemplo HTML simple como el del capítulo anterior y funcionaría muy bien. Para este trabajo de graduación vamos a cuidar un poco más la estética y

presentaremos los datos no en una sola página de datos plana, sino más bien en un sitio donde se puedan tener todas las opciones de comunicación que el PLC permite. Una aclaración necesaria en cuanto al uso de HTML con el PLC es que como se utilizan controles ActiveX que es una tecnología de Microsoft, esta página para ser vista adecuadamente se recomienda sólo usar MS Internet Explorer.

El sitio en cuestión está compuesto entonces de un total de 10 páginas Web. Cada una ejecutará un trozo de código que permitirá separar las opciones de comunicación por páginas. A estas se podrá acceder como un enlace desde la página principal como se muestra en la imagen.

Figura 27. Imagen de página principal.



Desde este sitio puede controlar, monitorear y acceder a todos los procesos y datos del PLC seleccionado.

Este Sitio es un Portal de Acceso para controlar todos los procesos e información contenida en el PLC. Todos los datos mostrados son directamente accedidos desde y hacia el PLC.

Para navegar por todas las opciones de control del PLC puede acceder con los botones de la izquierda de su pantalla. Cada Sitio contiene las instrucciones para monitorear, controlar y acceder al PLC.

En "[Descubrir](#)" se hace una búsqueda de todos los PLC's conectados a la Red Local, para poder conocer a todos los IPC disponibles para controlarlos.

La opción "[Ver Datos](#)" le permite intercambiar datos con el PLC, sean estos estados de entradas, salidas, banderas, temporizadores, etc.

"[Controladores](#)" proporciona una lista de todos los controladores o drivers que contiene el PLC, útil para conocer que capacidades de comunicación tiene el IPC accedido.

La opción con mejores y mayores capacidades es "[Control Remoto](#)" pues desde ella se tiene acceso a controlar y manipular al PLC como si se estuviera frente al panel de control o de programación del PLC. Se puede ordenar al PLC que arranque, detenga, pause o reinicie cualquier proceso.

El acceso "[Enviar Comandos](#)" es una herramienta para programadores expertos pues permite el envío de comandos predefinidos por el PLC.

[Inicio](#)

[Descubrir](#)

[Ver Datos](#)

[Controladores](#)

[Control Remoto](#)

[Enviar Comandos](#)

[Ayuda](#)

Así se ve la página principal y desde ahí se tiene una explicación de cómo usar el sitio. Para editar el sitio puede utilizarse cualquier herramienta de edición de HTML. El código fuente de cada página se da a continuación, con solo poner el siguiente código fuente en un archivo HTML basta para que el sistema funcione.

### 5.1.1 Página Principal o MAIN.HTML

Esta página en combinación con las siguientes tres (Links.html, index.html y logo.html) conforman la página principal desde donde se puede navegar por el sitio del acceso, monitoreo y control del PLC.

```
<html>
<head>
<title>Comunicación PLC Internet</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script language="JavaScript" type="text/JavaScript">
<!--
function MM_reloadPage(init) { //reloads the window if Nav4 resized
  if (init==true) with (navigator) {if ((appName=="Netscape")&&(parseInt(appVersion)==4)) {
    document.MM_pgW=innerWidth; document.MM_pgH=innerHeight; onresize=MM_reloadPage; }}
  else if (innerWidth!=document.MM_pgW || innerHeight!=document.MM_pgH) location.reload();
}
MM_reloadPage(true);
//-->
</script>
</head>

<frameset rows="" cols="186,*" framespacing="0" frameborder="no" border="0" bordercolor="#FFFFFF">
  <frame src="Links.htm" name="leftFrame" scrolling="NO" noresize>
  <frameset rows="154,*" cols="" framespacing="0" frameborder="no" border="0" bordercolor="#000000">
    <frame src="Logo.htm" name="topFrame" scrolling="NO" noresize>
    <frame src="index.htm" name="mainFrame">
  </frameset>
</frameset>
</frameset>
<noframes><body>
```

```
</body></noframes>
</html>
```

### 5.1.2 Página INDEX.HTML

Aquí está el texto que se ve en la página de inicio. Está aparte por cuestiones de forma y no de función, es decir se usaron tablas para crear el sitio y cada tabla contiene un enlace a otra página y esta se actualiza al presionar sobre el botón respectivo. Así solo se actualiza el texto del centro de la página.

```
<html>
<head>
<title>Página de Inicio</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body>
<p><font face="Verdana, Arial, Helvetica, sans-serif">Este Sitio es un Portal
de Acceso para controlar todos los procesos e informaci&oacute;n contenida en
el PLC. Todos los datos mostrados son directamente accedidos desde y hacia el
PLC. </font></p>
<p><font face="Verdana, Arial, Helvetica, sans-serif">Para navegar por todas las
opciones de control del PLC puede acceder con los botones de la izquierda de
su pantalla. Cada Sitio contiene las instrucciones para monitorear, controlar
y acceder al PLC.</font></p>
<p><font face="Verdana, Arial, Helvetica, sans-serif">En &quot;<strong><a href="Descubrir.htm"
target="mainFrame">Descubrir</a></strong>&quot;
se hace una busqueda de todos los PLC's conectados a la Red Local, para poder
conocer a todos los IPC disponibles para controlarlos.</font></p>
<p><font face="Verdana, Arial, Helvetica, sans-serif">La opci&oacute;n &quot;<strong><a href="Datos.htm"
target="mainFrame">Ver
Datos</a></strong>&quot; le permite intercambiar datos con el PLC, sean estos
estados de entradas, salidas, banderas, temporizadores, etc.</font></p>
<p><font face="Verdana, Arial, Helvetica, sans-serif">&quot;<strong><a href="Controladores.htm"
target="mainFrame">Controladores</a></strong>&quot;
proporciona una lista de todos los controladores o drivers que contiene el PLC,
&uacute;til para conocer que capacidades de comunicaci&oacute;n tiene el IPC
```

```

accesado.</font></p>
<p><font face="Verdana, Arial, Helvetica, sans-serif">La opci&oacute;n con mejores
y mayores capacidades es &quot;<strong><a href="Remoto.htm" target="mainFrame">Control
Remoto</a></strong>&quot; pues desde ella se tiene acceso a controlar y manipular
al PLC como si se estuviera frente al panel de control o de programaci&oacute;n
del PLC. Se puede ordenar al PLC que arranque, detenga, pause o reinicie cualquier
proceso.</font></p>
<p><font face="Verdana, Arial, Helvetica, sans-serif">El acceso &quot;<strong><a href="comandos.htm"
target="mainFrame">Enviar
Comandos</a></strong>&quot; es una herramienta para programadores expertos pues
permite el envio de comandos predefinidos por el PLC.</font></p>
</body>
</html>

```

### 5.1.3 Página LINKS.HTML

Aquí están los botones para hacer los enlaces con el resto de las páginas.

```

<html>
<head>
<title>Accesos</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script language="JavaScript" type="text/JavaScript">
<!--
function MM_goToURL() { //v3.0
  var i, args=MM_goToURL.arguments; document.MM_returnValue = false;
  for (i=0; i<(args.length-1); i+=2) eval(args[i]+".location='"+args[i+1]+'");
}
//-->
</script>
</head>

<body>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>
  <input name="Button" type="button" onClick="MM_goToURL('parent.frames[\'mainFrame\'],'index.htm');return
document.MM_returnValue" value="Inicio">
</p>

```

```

<p>
  <input name="Button" type="button"
onClick="MM_goToURL('parent.frames[\mainFrame\'],'Descubrir.htm');return document.MM_returnValue"
value="Descubrir" >
</p>
<p>
  <input name="Button22" type="button"
onClick="MM_goToURL('parent.frames[\mainFrame\'],'Datos.htm');return document.MM_returnValue"
value="Ver Datos">
</p>
<p>
  <input name="Button2" type="button"
onClick="MM_goToURL('parent.frames[\mainFrame\'],'Controladores.htm');return document.MM_returnValue"
value="Controladores">
</p>
<p>
  <input name="Button2" type="button"
onClick="MM_goToURL('parent.frames[\mainFrame\'],'Remoto.htm');return document.MM_returnValue"
value="Control Remoto">
</p>
<p>
  <input name="Button2" type="button"
onClick="MM_goToURL('parent.frames[\mainFrame\'],'comandos.htm');return document.MM_returnValue"
value="Enviar Comandos">
</p>
<p>
  <input name="Button" type="button" onClick="MM_goToURL('parent.frames[\mainFrame\'],'Ayuda.htm');return
document.MM_returnValue" value="Ayuda">
  <font face="Verdana, Arial, Helvetica, sans-serif"> </font> </p>
</body>
</html>

```

#### 5.1.4 Página LOGO.HTML

Esta página es la que contiene la información del logo a mostrar, en nuestro caso el logo de la Universidad de San Carlos de Guatemala.

```

<html>
<head>
<title>Descripcion General</title>

```

```

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body>
<p align="center"> </p>
<p><font face="Verdana, Arial, Helvetica, sans-serif">Desde este sitio puede controlar,
monitorear y acceder a todos los procesos y datos del PLC seleccionado.</font></p>
</body>
</html>

```

### 5.1.5 Página AYUDA.HTML

Contiene una pequeña descripción de ayuda para el usuario.

```

<html>
<head>
<title>Ayuda y Preguntas Frecuentes</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body>

<p>&nbsp;</p>
<H2><font face="Verdana, Arial, Helvetica, sans-serif">Ayuda y Preguntas Frecuentes</font></H2>
<P><font face="Verdana, Arial, Helvetica, sans-serif"> </font><font size="3" face="Verdana, Arial, Helvetica,
sans-serif">El
primer paso para resolver cualquier problema es verificar la conectividad de
la Red de las PC con el PLC.</font>
<P><font face="Verdana, Arial, Helvetica, sans-serif"> La pagina consta basicamente de un menu que esta
desplegado en la parte izquierda de la pantalla, al presionar sobre cada boton se es llevado a la pagina donde
se da la funcion específica que se muestra. Si no aparece ningun PLC en la pantalla de DESCUBRIR verifique
que el PLC esté conectado y ambos en el mismo rango de direcciones IP, se recomienda usar las direcciones
siguientes: en el IPC 192.168.2.150 y el la PC 192.168.2.151. Esta página utiliza controles ActiveX asi que se
deben instalar y ademas permitir su uso. Debido a que los controles ActiveX son una tecnologia de Microsoft
esta página solo corre adecuadamente en MS Internet Explorer. </font>
<p><font face="Verdana, Arial, Helvetica, sans-serif">5703 8980 William Hern&acute;ndez</font></p>
</body>
</html>

```

## 5.1.6 Página COMANDOS.HTML

Desde esta página se envían comandos al PLC.

```
<HTML>
<HEAD>
<TITLE>Envío de Comandos</TITLE>

<SCRIPT>

function doSend()
{
    // Check user input
    if (txtIp.value == " || txtCommand.value == ")
    {
        alert("Error: no se especificó ninguna IP.");
        return;
    }

    // Set command and IP address of the commandX control
    commandX.ip = txtIp.value;
    commandX.Command = txtCommand.value;

    // Send the command
    if (commandX.SendCommand() == false)
        alert("Error: no se pudo enviar comando.");
    else
        btnSend.disabled = true;
}

</SCRIPT>

<SCRIPT for=commandX event="ResultReceived(result)">

    // Response received
    btnSend.disabled = false;
    if (result == "")
        txtResult.innerHTML = "Sin Respuesta.";
    else
        txtResult.innerHTML = result

</SCRIPT>
```

```

<SCRIPT for=commandX event="SendAttempt(count)">

    // Send attempt
    txtResult.innerText = "attempt " + count;

</SCRIPT>

<SCRIPT for=commandX event=Timeout>

    // Timeout !
    txtResult.innerText = "timeout";
    alert("Tiempo Agotado, verifique la dirección IP y la conexión de red");
    btnSend.disabled = false;

</SCRIPT>

<SCRIPT for=document event=onkeydown>
    // Handle enter key
    if (event.keyCode == 13)
    {
        event.returnValue = false;
        doSend();
    }

</SCRIPT>

</HEAD>
<BODY style="background-color: #FFFFFF; font-family: arial; font-size: 10pt">

<OBJECT id=commandX classid="clsid:79D33D79-C967-44F4-BD75-970FC9C24A82" VIEWASTEXT>
    <PARAM NAME="Ip" VALUE="">
    <PARAM NAME="Command" VALUE="">
    <PARAM NAME="Interval" VALUE="250">
    <PARAM NAME="PacketCount" VALUE="4">
</OBJECT>

<H2> <font size="+2" face="Verdana, Arial, Helvetica, sans-serif">Envío de Comandos </font></H2>
<P><font face="Verdana, Arial, Helvetica, sans-serif"> Esta página puede ser usada
para enviar comandos al interprete del IPC y se reciben los resultados de cada
comando enviado. Ingrese la dirección IP del IPC y el comando a enviar, luego
presione Enviar.</font>
<P>
<TABLE border=0 style="font-family: arial; font-size: 10pt">
    <TR>

```

```

        <TD width="62" align=right>IP address</TD>
        <TD width="229">
            <INPUT id=txtIp value="192.168.1.3" style="margin-left: 20px" maxLength=20>
        </TD>
    </TR>
    <TR>
        <TD align=right>Command</TD>
        <TD>
            <INPUT id=txtCommand value="!26" style="margin-left: 20px" maxLength=512>
            <INPUT id=btnSend type=button value="Enviar" onClick="doSend()">
        </TD>
    </TR>
    <TR>
        <TD align=right valign="top">Result</TD>
        <TD>
            <SPAN id=txtResult style="margin-left: 20px;"></SPAN>
        </TD>
    </TR>
</TABLE>

</BODY>
</HTML>

```

### 5.1.7 Página CONTROLADORES.HTML

Desde aquí se puede ver una lista de los IPC encontrados en la red.

```

<HTML>
<HEAD>
<TITLE>Lista de Controladores</TITLE>

<SCRIPT>
function doSend()
{
    // Check user input
    if (txtIp.value == "")
    {
        alert("Error: No se específico dirección IP.");
        return;
    }
}

```

```

        // Rewind the driver number counter
        driverNumber = 0;

        // Set IP address of the commandX control
        commandX.Ip = txtIp.value;

        // Update screen
        txtResult.innerHTML = "";

        // Begin querying for installed drivers
        doSendQuery();
    }

function doSendQuery()
{
    // Set the command to send
    commandX.Command = "!" + driverNumber;

    // Send the command
    if (commandX.SendCommand() == false)
        alert("Error: No se pudo enviar comando.");
    else
        btnSend.disabled = true;
}

</SCRIPT>

<SCRIPT for=commandX event="ResultReceived(result)">

    // Response received
    if (result.charAt(0) == '=')
    {
        // A driver was found, update display
        txtResult.innerHTML += "!" + driverNumber + result + "<br>";
    }

    // Check next driver
    if (++driverNumber > 100)
        btnSend.disabled = false;
    else
        doSendQuery();

</SCRIPT>

```

```

<SCRIPT for=commandX event=Timeout>

    // Timeout !
    alert("Tiempo agotado, verificar dirección IP conexión de red");
    btnSend.disabled = false;

</SCRIPT>

</HEAD>
<BODY style="background-color: #FFFFFF; font-family: arial; font-size: 10pt">

<OBJECT id=commandX classid=clsid:79D33D79-C967-44F4-BD75-970FC9C24A82 VIEWASTEXT>
    <PARAM NAME="Ip" VALUE="">
    <PARAM NAME="Command" VALUE="">
    <PARAM NAME="Interval" VALUE="250">
    <PARAM NAME="PacketCount" VALUE="4">
</OBJECT>

<H2> <font face="Verdana, Arial, Helvetica, sans-serif">Lista de Controladores
</font></H2>
<P><font face="Verdana, Arial, Helvetica, sans-serif"> Aquí se puede encontrar
una lista de controladores instalados en el IPC. Ingrese la dirección IP del
IPC que se quiere examinar y presione Inicio.<BR>
</font>
<P><font face="Verdana, Arial, Helvetica, sans-serif"> Dirección IP:</font>
<INPUT id=txtIp value="192.168.1.3" maxLength=20>
<INPUT id=btnSend type=button value="Inicio" onClick="doSend()">

<P>
<SPAN id=txtResult></SPAN>

</BODY>
</HTML>

```

### 5.1.8 Página DATOS.HTML

Aquí se tiene la página desde la cual se pueden leer datos del IPC.

```

<HTML>
<HEAD>
<TITLE>Intercambio de Datos</TITLE>

```



```

<H2> <font face="Verdana, Arial, Helvetica, sans-serif">Intercambio de Datos con
  PLC</font></H2>
<P><font face="Verdana, Arial, Helvetica, sans-serif"> Ingrese la dirección IP
  del PLC para intercambiar datos u operandos. Los primeros 10 operandos de cada
  tipo serán desplegados.</font>
<P>
<TABLE border=0 style="font-family: arial; font-size: 10pt">
  <TR>
    <TD align=right><font face="Verdana, Arial, Helvetica, sans-serif">Dirección
      IP</font></TD>
    <TD> <INPUT id=txtIp value="192.168.1.3" style="margin-left: 20px; width: 200px" maxLength=20>
    </TD>
  </TR>
  <TR>
    <TD align=right><font face="Verdana, Arial, Helvetica, sans-serif">Tipo de
      Operando</font></TD>
    <TD> <SELECT id=selectType onChange="doExchange()" style="margin-left: 20px; width: 200px">
      <OPTION selected>[Seleccione Tipo de Operando]</OPTION>
      <OPTION>Entradas</OPTION>
      <OPTION>Salidas</OPTION>
      <OPTION>Banderas</OPTION>
      <OPTION>Programas</OPTION>
      <OPTION>Registros</OPTION>
      <OPTION>Temporizadores</OPTION>
    </SELECT> </TD>
  </TR>
  <TR>
    <TD align=right valign=top><font face="Verdana, Arial, Helvetica, sans-serif">Resultados</font></TD>
    <TD> <SPAN id=txtResult style="margin-left: 20px;"></SPAN> </TD>
  </TR>
</TABLE>

</BODY>
</HTML>

```

### 5.1.9 Página DESCUBRIR.HTML

Esta página da la opción de mandar un comando de descubrir IPC conectados en la red.

```

<HTML>
<HEAD>

```

```
<TITLE>Buscar PLC en la Red</TITLE>
```

```
<SCRIPT>
```

```
function dolnit()
{
    // Define columns of the List view
    ListView1.ColumnHeaders.Add(1, "", "Dirección IP", 100);
    ListView1.ColumnHeaders.Add(2, "", "Máscara de Red", 100);
    ListView1.ColumnHeaders.Add(3, "", "Dirección MAC", 100);
    ListView1.ColumnHeaders.Add(4, "", "Proyecto", 100);
    ListView1.ColumnHeaders.Add(5, "", "Kernel", 100);
    ListView1.ColumnHeaders.Add(6, "", "Controlador IP", 100);
    ListView1.ColumnHeaders.Add(7, "", "CPU", 100);

    // Resize the view
    doSize();

    // Make view visible
    ListView1.style.visibility = "";

    // Start discovering IPCs
    doDiscovery();
}

function doSize()
{
    // Size the list view to the client area of the browser
    ListView1.style.width = Math.max(100, document.body.clientWidth -25);
    ListView1.style.height = Math.max(75, document.body.clientHeight -150 -txtHeader.style.bottom);
}

function doDiscovery()
{
    // Clear the List view
    ListView1.ListItems.Clear();

    // Start the discovery
    if (discoverX.Discover() == false)
    {
        txtStatus.innerText = "busqueda fallida.";
        alert("No se pudo enviar paquete de busqueda.");
        return;
    }
}
```

```

        // Update screen
        btnDiscover.disabled = true;
        txtStatus.innerText = "buscando...";
    }

</SCRIPT>
<SCRIPT for=discoverX event="Ready(count)">
    // Handle discovery ready event
    btnDiscover.disabled = false;

    // Display number of discovered IPCs
    if (count)
        txtStatus.innerText = count + " IPC's encontrados";
    else
        txtStatus.innerText = "No se encontraron IPC's";

</SCRIPT>
<SCRIPT for=discoverX event="Discovery(ip, mask, mac, project, kernel, driver, cpu)">
    // Add an entry to the List view
    var itemNew = ListView1.ListItems.Add(1, "", ip);
    // Write information about the discovered IPC to the new entry
    itemNew.SubItems(1) = mask;
    itemNew.SubItems(2) = mac;
    itemNew.SubItems(3) = project;
    itemNew.SubItems(4) = kernel;
    itemNew.SubItems(5) = driver;
    itemNew.SubItems(6) = cpu;

</SCRIPT>

<SCRIPT for=ListView1 EVENT=MouseUp>

    // Open home page of the clicked IPC
    if (ListView1.SelectedItem && confirm("Abrir página principal del PLC ?"))
        window.location.href = "http://" + ListView1.SelectedItem.Text;

</SCRIPT>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"></HEAD>
<BODY bgcolor="#009999" style="background-color: #FFFFFF; font-family: arial; font-size: 10pt"
onLoad="doInit()" onResize="doSize()">
<p>
<OBJECT id=discoverX classid=clsid:553E2D8D-48AD-4556-8205-86F2F9482565 VIEWASTEXT>
    <PARAM NAME="Interval" VALUE="350">
    <PARAM NAME="PacketCount" VALUE="3">

```

```

</OBJECT>
</p>
<p><strong><font size="+2" face="Verdana, Arial, Helvetica, sans-serif">Lista
de PLCs encontrados en la RED</font></strong></p>
<p><font face="Verdana, Arial, Helvetica, sans-serif">Desde aqu&iacute; se puede
hacer una busqueda de PLCs en la Red. Presione sobre Buscar para Encontrar los
PLC conectados a la RED.</font></p>
<H2 id=txtHeader>
</H2>
<P>
<INPUT name="Button" type=button id=btnDiscover style="width: 100px" accesskey=D
OnClick="doDiscovery()" value=Buscar>
<SPAN style="margin-left: 20px" id=txtStatus></SPAN>
<OBJECT id=ListView1 classid=clsid:58DA8D8A-9D6A-101B-AFC0-4210102A8DA7 VIEWASTEXT
style="font-family: arial; font-size: 10pt; left: 0px; width: 480px; top: 0px; height: 320px; visibility: hidden">
<PARAM NAME="SortKey" VALUE="0">
<PARAM NAME="View" VALUE="3">
<PARAM NAME="Arrange" VALUE="1">
<PARAM NAME="LabelEdit" VALUE="0">
<PARAM NAME="SortOrder" VALUE="0">
<PARAM NAME="Sorted" VALUE="-1">
<PARAM NAME="MultiSelect" VALUE="0">
<PARAM NAME="LabelWrap" VALUE="-1">
<PARAM NAME="HideSelection" VALUE="-1">
<PARAM NAME="HideColumnHeaders" VALUE="0">
<PARAM NAME="OLEDragMode" VALUE="0">
<PARAM NAME="OLEDropMode" VALUE="0">
<PARAM NAME="ForeColor" VALUE="-2147483640">
<PARAM NAME="BackColor" VALUE="-2147483643">
<PARAM NAME="BorderStyle" VALUE="1">
<PARAM NAME="Appearance" VALUE="1">
<PARAM NAME="MousePointer" VALUE="0">
<PARAM NAME="Enabled" VALUE="1">
<PARAM NAME="NumItems" VALUE="0">
</OBJECT>

</BODY>
</HTML>

```

### 5.1.10 Página REMOTO.HTML

Esta página permite el envío de comandos hacia el PLC.

```

<HTML>
<HEAD>
<TITLE>Control Remoto</TITLE>

<SCRIPT>

function doSendCommand(command)
{
    // Check user input
    if (txtIp.value == "")
    {
        alert("Error: no se especificó ninguna IP.");
        return;
    }

    // Write the status line
    txtStatus.style.color = "#000000";
    txtStatus.innerHTML = "Enviando Comando " + command + " al IPC " + txtIp.value;

    // Store the command and IP address
    commandX.Ip = txtIp.value;
    commandX.Command = command;

    // Send the command
    commandX.SendCommand();
}

</SCRIPT>

<SCRIPT for=commandX event="ResultReceived(result)">

    // Command was handled
    txtStatus.style.color = "#000000";
    txtStatus.innerHTML = "The command was handled by the remote IPC";

</SCRIPT>

<SCRIPT for=commandX event="Timeout()">

    // Timeout occurred
    txtStatus.style.color = "#ff0000";
    txtStatus.innerHTML = "There was a timeout communicating with the remote IPC";

</SCRIPT>

```

```

</HEAD>
<BODY style="background-color: #FFFFFF; font-family: arial; font-size: 10pt">

<OBJECT id=commandX classid=clsid:79D33D79-C967-44F4-BD75-970FC9C24A82 VIEWASTEXT>
  <PARAM NAME="Ip" VALUE="192.168.0.150">
  <PARAM NAME="Command" VALUE="">
  <PARAM NAME="Interval" VALUE="250">
  <PARAM NAME="PacketCount" VALUE="8">
</OBJECT>

<H2>
Control Remoto
</H2>

<P>
Desde esta página puede controlarse al PLC. Solo especifique la dirección IP del IPC a controlar y presione el
boton correspondiente a la orden que se quiere enviar.<br> <P>
IP address: <INPUT id=txtIp value="192.168.1.3" maxLength=20>

<P>
<BUTTON style="width: 100px" onclick="doSendCommand('r')">Iniciar</BUTTON>
<BUTTON style="width: 100px" onclick="doSendCommand('s')">Detener</BUTTON>
<BUTTON style="width: 100px" onclick="doSendCommand('b')">Pausar</BUTTON>
<BUTTON style="width: 100px" onclick="doSendCommand('l_boot')">Reiniciar</BUTTON>

<P>
<SPAN id=txtStatus style="font-family: Arial; font-size: 9pt"></SPAN>

</BODY>
</HTML>

```

## 5.2 Programa del PLC

Como mencionamos en el capítulo dos, usaremos la misma idea de proceso que en el ejemplo, pero dejaremos solamente las dos bandas, dos ventiladores, un extractor, y eliminaremos el control PID, los pickers, cuatro ventiladores, pero añadiremos un conteo de horas de trabajo y simularemos la señal de temperatura, esto debido a las capacidades del PLC del laboratorio en cuanto a IO. Las características del PLC a usar son las siguientes:

Tabla XIV. Características del PLC FEC FC440.

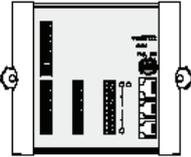
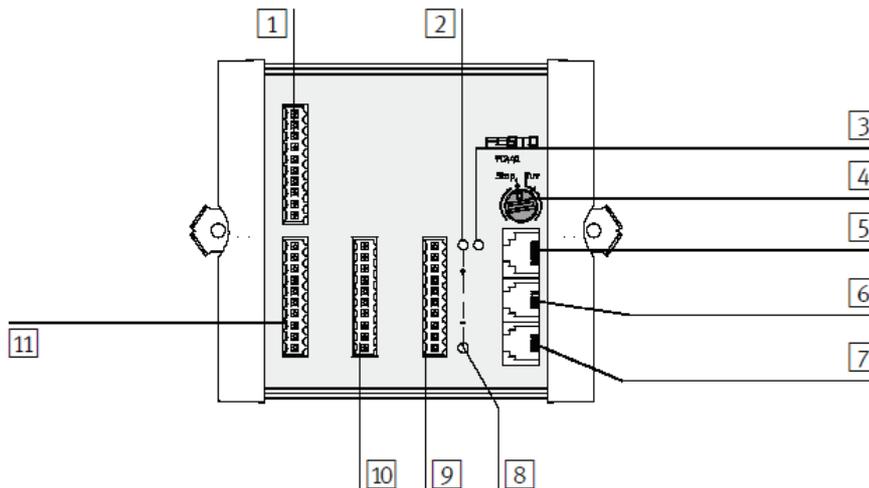
<p>IPC FEC FC440</p> 	<ul style="list-style-type: none"> <li>- 16 inputs, 8 transistor outputs, 24 V DC, Ethernet 10 BaseT.</li> <li>- Processor 80186</li> <li>- Working memory 512 Kbytes</li> <li>- Program memory 512 Kbytes flash</li> <li>- Interface RS232 via SM14/-15</li> <li>- Power consumption 200 mA nominal at 24 V</li> <li>- Operating system Festo FST, Multiprog</li> <li>- Dimensions 105 x 35 x 114,2</li> <li>- Overall dimensions 105 x 35 x 132,2</li> </ul>
--	--

Figura 28. Características del PLC FEC FC440.



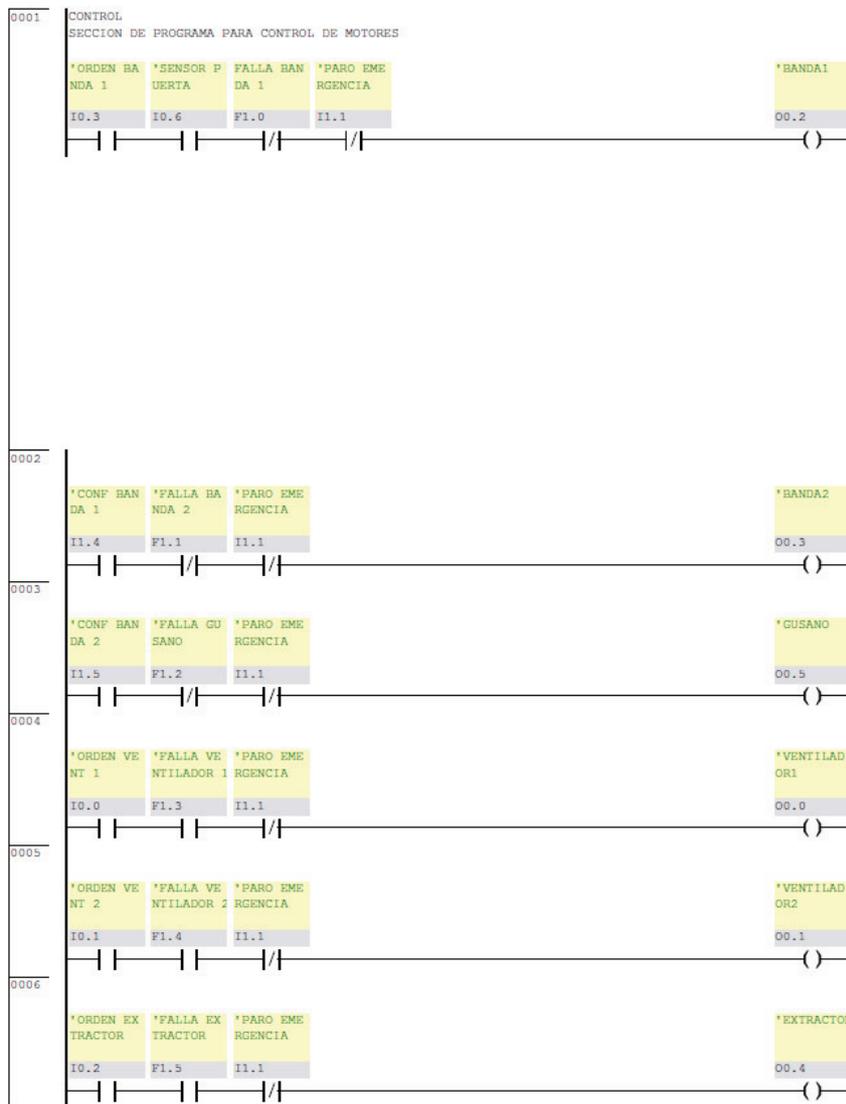
1. Entradas E0.0 a E0.7.
2. LED de encendido.
3. LED de comunicaciones de red.
4. Selector de RUN / STOP.
5. Conexión de red.
6. Interfaz de comunicaciones.
7. Interfaz de extensión.
8. LED de estado.
9. Fuente de alimentación.
10. Salidas A0.0 a A0.7.
11. Entradas E1.0 a E1.7.

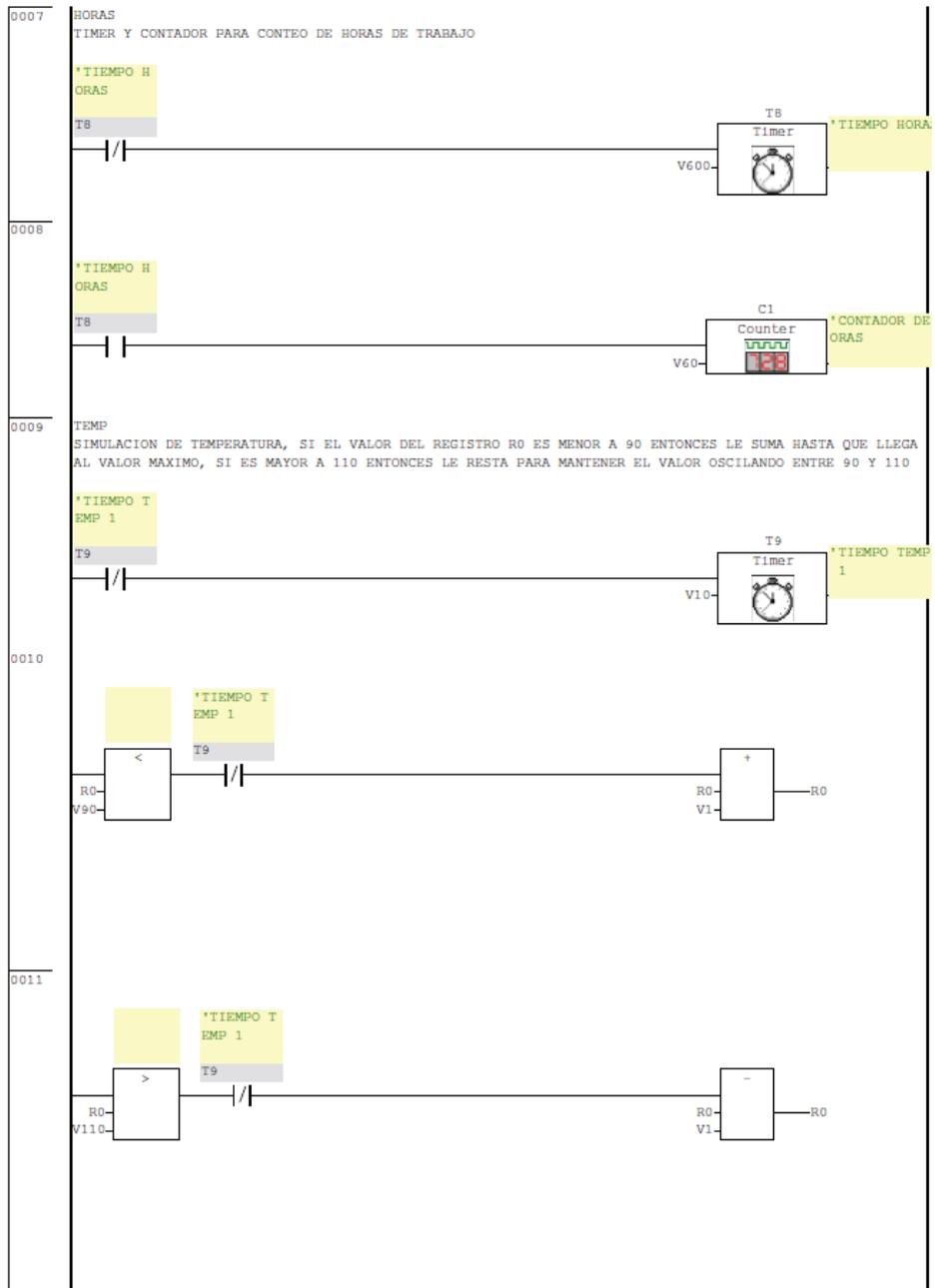
La lista de señales según las modificaciones mencionadas es la siguiente:

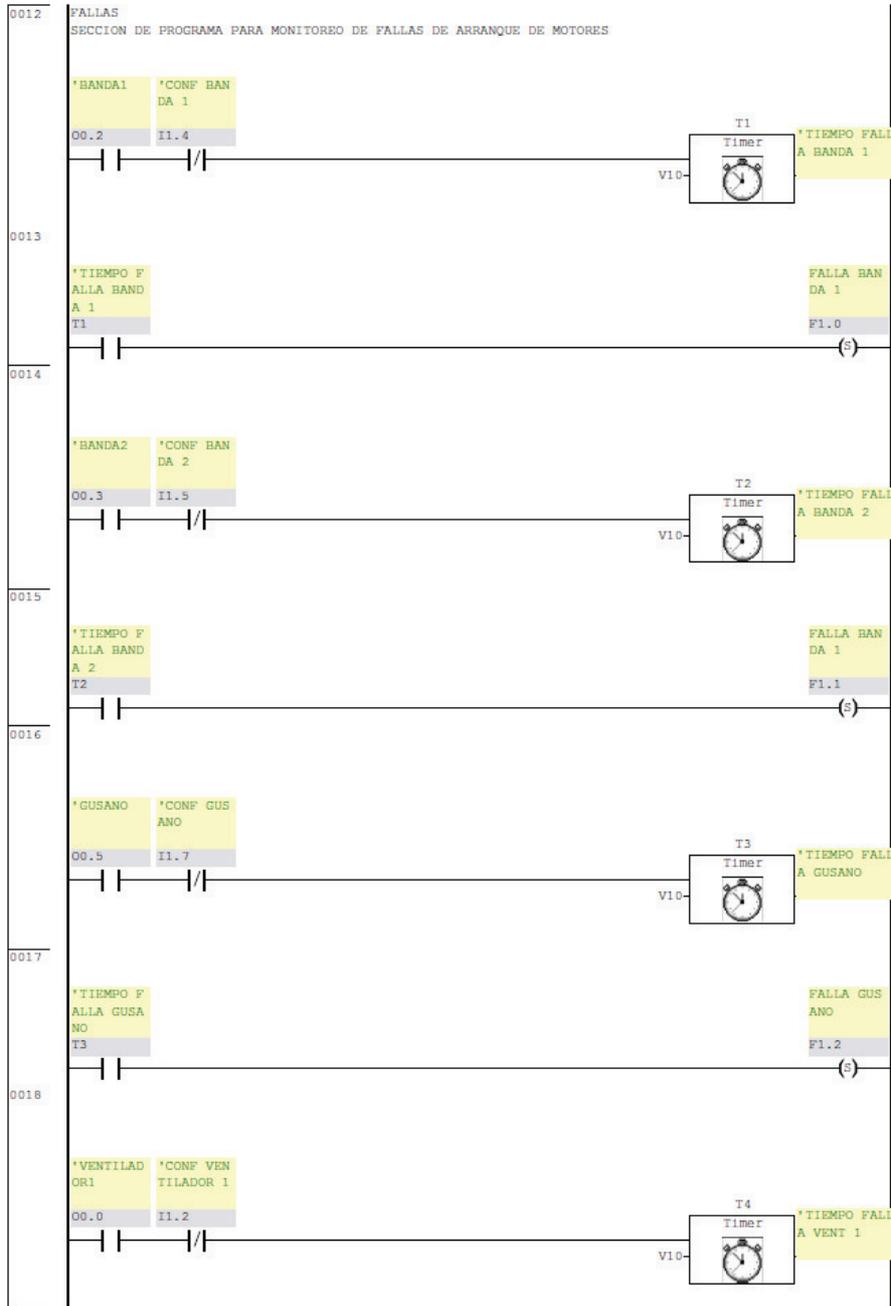
Tabla XV. Asignación de IO para programa de PLC.

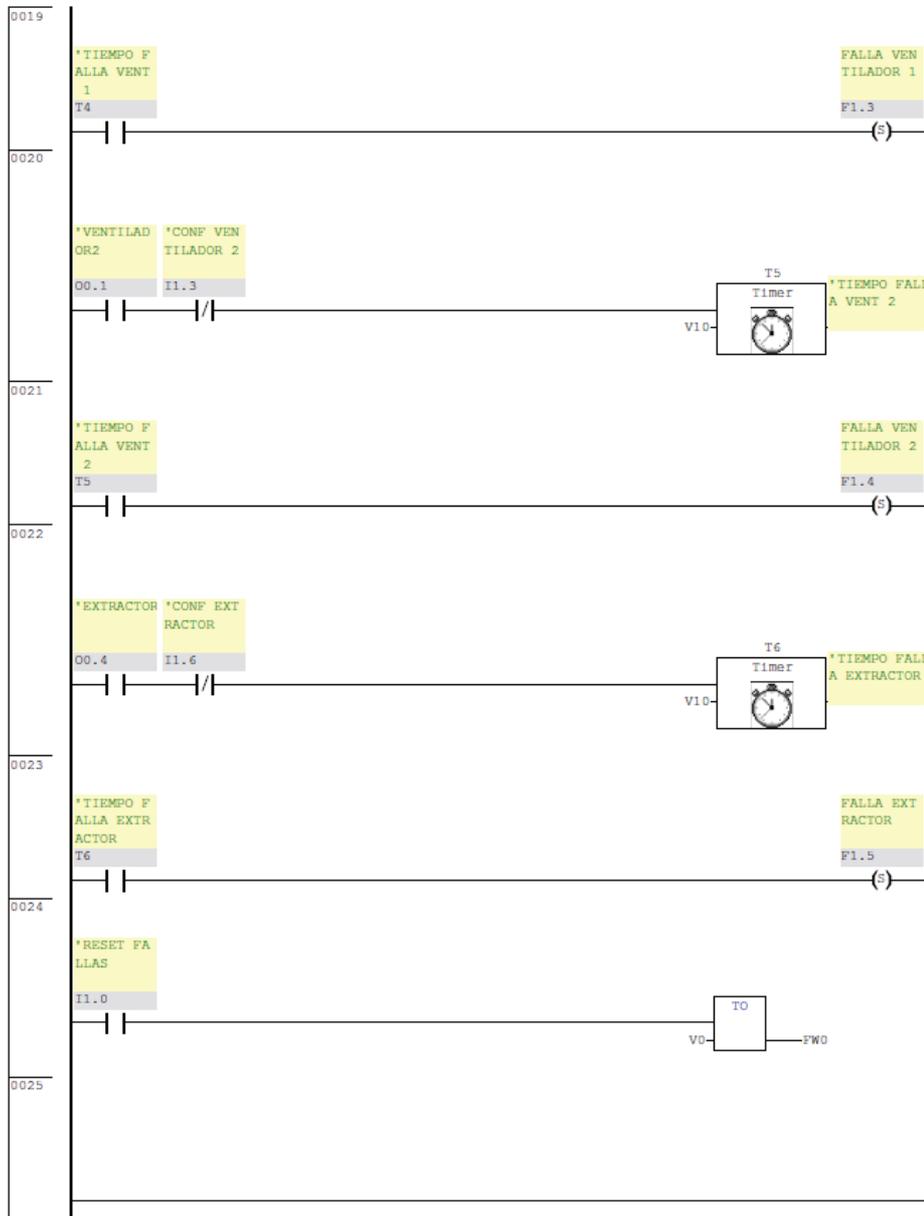
LABORATORIO ELECTRONEUMÁTICA		LISTADO DE SEÑALES				
				0	0	
No.	Descripción	I/O	Tipo	Vaux	No. ID 2	No. ID 3
1	Orden Arranque Motor Ventilador 1	DI		24VDC	E:0/0	BORNE X100 / 1
2	Orden Arranque Motor Ventilador 2	DI		24VDC	E:0/1	BORNE X100 / 2
3	Orden Arranque Motor Extractor	DI		24VDC	E:0/2	BORNE X100 / 3
4	Orden Arranque Motor Banda 1	DI		24VDC	E:0/3	BORNE X100 / 4
5	Orden Arranque Motor Banda 2	DI		24VDC	E:0/4	BORNE X100 / 5
6	Orden Arranque Motor Gusano	DI		24VDC	E:0/5	BORNE X100 / 6
7	Sensor Puerta Secador	DI		24VDC	E:0/6	BORNE X100 / 7
8	Orden Inicio Proceso	DI		24VDC	E:0/7	BORNE X100 / 8
9	Reset de Fallas	DI		24VDC	E:1/0	BORNE X100 / 9
10	Paro Emergencia	DI		24VDC	E:1/1	BORNE X100 / 10
11	Confirmación Arranque Ventilador 1	DI		24VDC	E:1/2	BORNE X100 / 11
12	Confirmación Arranque Ventilador 2	DI		24VDC	E:1/3	BORNE X100 / 12
13	Confirmación Arranque Banda 1	DI		24VDC	E:1/4	BORNE X100 / 13
14	Confirmación Arranque Banda 2	DI		24VDC	E:1/5	BORNE X100 / 14
15	Confirmación Arranque Extractor	DI		24VDC	E:1/6	BORNE X100 / 15
16	Confirmación Arranque Gusano	DI		24VDC	E:1/7	BORNE X100 / 16
17	Salida Arranque Motor Ventilador 1	DO		24VDC	A:0/0	BORNE X100 / 25
18	Salida Arranque Motor Ventilador 2	DO		24VDC	A:0/1	BORNE X100 / 26
19	Salida Arranque Motor Banda 1	DO		24VDC	A:0/2	BORNE X100 / 27
20	Salida Arranque Motor Banda 2	DO		24VDC	A:0/3	BORNE X100 / 28
21	Salida Arranque Motor Extractor	DO		24VDC	A:0/4	BORNE X100 / 29
22	Salida Arranque Motor Gusano	DO		24VDC	A:0/5	BORNE X100 / 30
23	Salida Sirena Fallas	DO		24VDC	A:0/6	BORNE X100 / 31
24	Reserva	DO		24VDC	A:0/7	BORNE X100 / 32

El código fuente se muestra a continuación:









El programa tiene comentarios donde se explica la secuencia de operación o algunas instrucciones para la simulación de temperatura o el conteo de horas.

### 5.3 Comunicación Ethernet PLC - PC:

Para asegurarnos que la comunicación entre IPC y PC sean exitosas debemos asegurarnos que ambos dispositivos estén en la misma red. Se aconseja usar las siguientes direcciones:

IPC 192.168.2.150

PC 192.168.2.151

Se pueden usar cualquier rango de direcciones pero siempre deben pertenecer ambos a la misma red. Además debe proveerse de una conexión física entre ambos dispositivos. Para esto puede ser tan simple como un cable UTP Categoría 5E con conectores RJ45 de tipo cruzado entre ambos dispositivos, o dos cables rectos y ambos yendo desde el dispositivo hasta un hub o switch.

Figura 29. Conector RJ45 y su configuración de cableado.

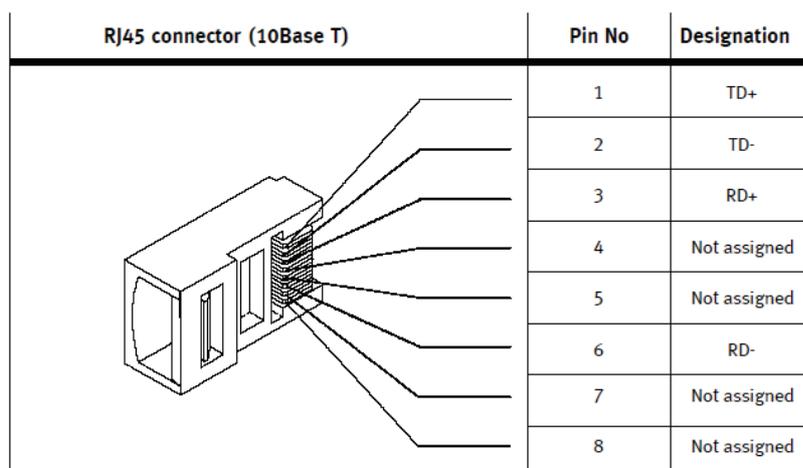


Figura 30. Opción 1 conexión directa entre dispositivos.

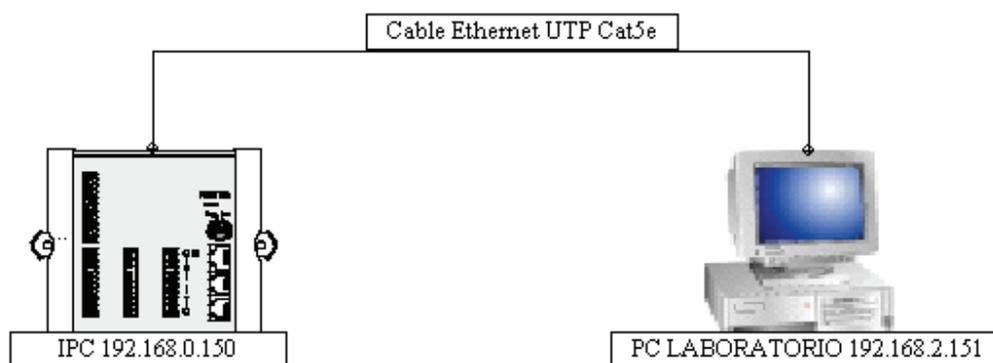


Figura 31. Opción 2 conexión por medio de concentrador o switch.

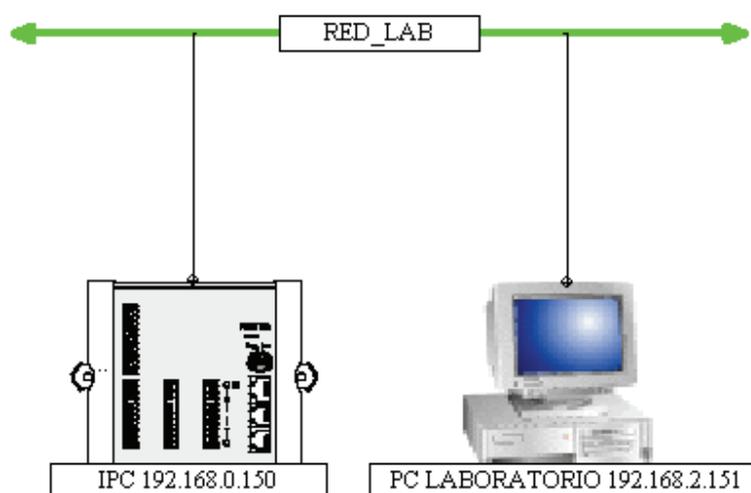
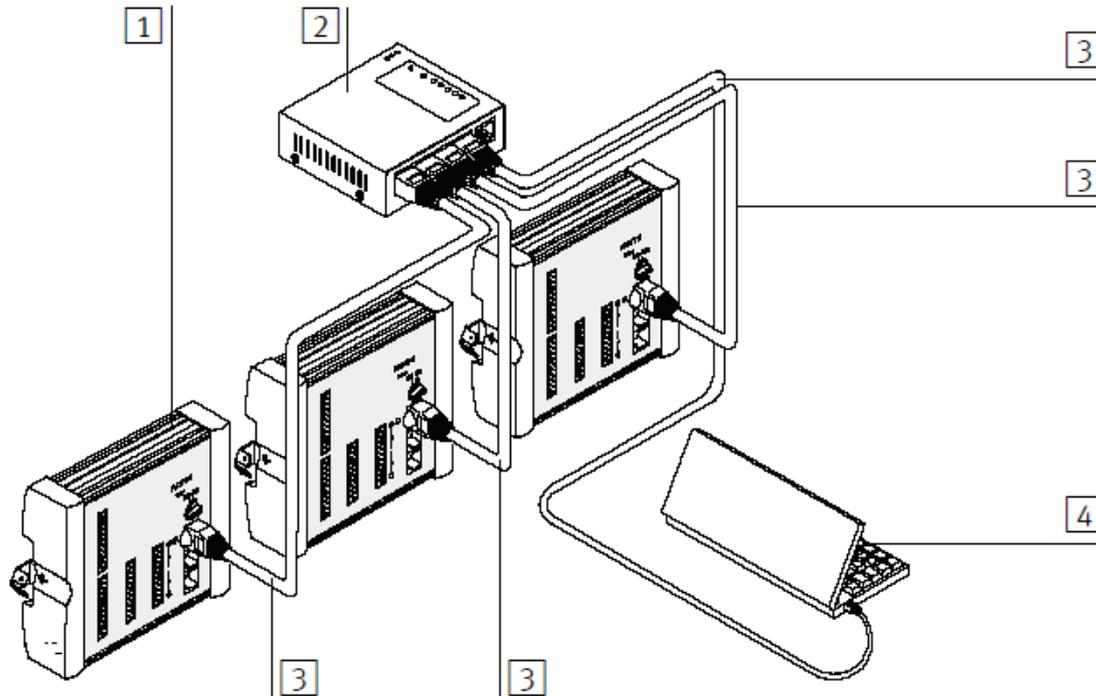


Figura 32. Ilustración de conexión por medio de concentrador o switch.



1. PLC FEC 440
2. HUB o Switch
3. Cable Ethernet recto
4. PC

Con todo lo expuesto hasta este punto se es completamente capaz de acceder, monitorear y controlar al PLC del laboratorio de Electroneumática por medio de una conexión Ethernet, ver cualquiera de las variables internas del PLC, iniciarlo, detenerlo, enviar comandos, etc.

## CONCLUSIONES

1. En cuanto al diseño de un sistema de control basado en un PLC que permita la efectiva comunicación y acceso a la información del proceso que controla, se ha mostrado que es totalmente factible desarrollar una herramienta de software que permita la comunicación entre una página web y un PLC, por medio de una red Ethernet y mostrar datos de proceso de la manera que mejor se requiera para las necesidades del usuario final, que en este caso son los estudiantes que utilizan el laboratorio de Electroneumática en la USAC.
2. Se ha podido constatar en el presente trabajo que la industria guatemalteca tiene avances significativos en cuanto a automatización, o dicho de manera más simple, la tendencia en Guatemala es darse cuenta que el uso de PLC puede mejorar un proceso industrial de muchas formas, técnica y financieramente, pues varios de los procesos comentados aquí están presentes en Guatemala y se encuentran totalmente automatizados.
3. Ni las redes Ethernet ni la programación web están sujetas solo a aplicaciones de internet, pues este estándar de redes y la programación HTML se han usado aquí para mostrar datos extraídos desde un PLC hacia una página web que hace de la interfaz gráfica de usuario amigable y una solución económica para las necesidades de la industria en Guatemala.
4. Los trozos de código previamente programados como los controles ActiveX pueden contenerse dentro de otro programa y/o trozo de código, para usarlos como interfaz y puerta de salida de datos desde un PLC hacia una computadora personal.



## RECOMENDACIONES

1. Una ampliación a esta herramienta podría ser la capacidad de configurar la red del acceso al PLC y poder hacer de esta una aplicación de acceso por internet por medio de una VPN. Para esto podrían utilizarse software especializados para crear redes de tipo VPN o conectar a la red un servidor con un sistema operativo tipo servidor como Windows server 2003.
2. La herramienta presentada aquí aunque está diseñada para ser usada en el laboratorio de Electroneumática y conectarse al PLC Festo instalado en dicho laboratorio, podría extenderse para mostrar datos de otros PLC de otros fabricantes, esto podría hacerse utilizando paquetes de software tipo OPC que cada fabricante provee, algunos con costo asociado, para la comunicación con estos PLC.



## BIBLIOGRAFÍA

1. **Academia de Networking de Cisco Systems:Guia del primer año** [Libro] / aut. Vito Amato Redactor de la Serie. - [s.l.] : Cisco Systems Ins, 2000.
2. **Automata Programable** [En línea]. - Febrero de 2006. - 21 de junio de 2009. - [http://es.wikipedia.org/wiki/Autómata\\_programable](http://es.wikipedia.org/wiki/Autómata_programable).
3. **Automatizacion Industrial** [En línea]. - Julio de 2006. - 21 de junio de 2009. - [http://es.wikipedia.org/wiki/Automatizaci%C3%B3n\\_industrial](http://es.wikipedia.org/wiki/Automatizaci%C3%B3n_industrial).
4. **Controlador Logico Programable** [En línea]. - 26 de noviembre de 2008. - 21 de junio de 2009. - [http://es.wikipedia.org/wiki/Controlador\\_l%C3%B3gico\\_programable](http://es.wikipedia.org/wiki/Controlador_l%C3%B3gico_programable).
5. **Diccionario Informatico: Control ActiveX** [En línea]. - 21 de junio de 2009. - <http://www.alegsa.com.ar/Dic/activex%20control.php>.
6. **El Concepto de Red** [En línea]. - 16 de octubre de 2008. - 21 de junio de 2009. - <http://es.kioskea.net/contents/initiation/concept.php3>.
7. **FST 4.0, Software Programacion IPC FESTO** / aut. FESTO. - [s.l.] : FESTO, 6 mayo 2004.
8. **Integrated Architecture, Software de Configuracion de PLC** / aut. Software Rockwell. - [s.l.] : Rockwell Software, 2009.

9. **Micro Programmable Controller IPC** [Libro] / aut. FESTO. - Esslingen : FESTO, 2001.
10. **PLC** [En línea]. - Diciembre de 2001. - 21 de junio de 2009. - <http://www.sc.ehu.es/sbweb/webcentro/automatica/WebCQMH1/PAGINA%20PRINCIPAL/PLC/plc.htm>.
11. **Programacion en Escalera** [Libro] / aut. Musalem Rodrigo A.. - [s.l.] : UTFSM, 2001.
12. **PROPUESTA PARA LA APLICACIÓN DE FUNCIONES DE COMUNICACIÓN EN EL PLC DE UNA MÁQUINA LLENADORA** / aut. Sanchez Juan Carlos. - Guatemala, Guatemala : Universidad de San Carlos de Guatemala, febrero de 2007.
13. **RSLogix500, Software Programacion PLC SLC Allen Bradley** / aut. Software Rockwell. - [s.l.] : Rockwell Software, 2009.
14. **RSLogix5000, Software Programacion PLC Familia Logix Allen Bradley** / aut. Software Rockwell. - [s.l.] : Rockwell Software, 2009.
15. **SLC 500 Modular Hardware, User Manual** [Libro] / aut. Bradley Allen. - [s.l.] : Allen Bradley, junio 2008. - Vols. 1747-UM011G-EN-P.

## APÉNDICE

### Costos

Una herramienta similar a la mostrada en los capítulos anteriores se puede encontrar como un adicional o software a parte en las casa matrices de PLC como Allen Bradley y su software RSBizware Historian u otros de similar índole como ION Enterprise de Schneider Electric. Estos paquetes de software poseen o tienen embebidas herramientas prediseñadas para adquirir datos y formular reportes, pero tienen un costo que puede empezar en los USD\$5000 para la licencias más sencillas pues tienen mucha ingeniería de desarrollo y estudio además de poseer licencias de terceros como de SQL.

La herramienta propuesta aquí si bien no puede compararse con las mencionadas en el párrafo anterior por todos los años de ingeniería empleados en el diseño y programación de ellas, tampoco puede compararse en precio, si bien no es totalmente gratis pues existe un costo ligado al tiempo empleado en programar la presentación en la página web, todas las herramientas de software son disponibles para el usuario final sin costo en este momento.

La parte de definición de costos siempre se vuelve complicada debido a la falta de experiencia en definir costos, a no contemplar todos los costos involucrados en la creación de una herramienta como esta o por no estar familiarizado con este tipo de funciones. Costo se define como la suma de esfuerzos y recursos que se han invertido para producir una cosa, de manera que vamos a definir un costo para esta herramienta basado en el precio por hora de ingeniería involucrada para desarrollarla. Así que tomando en cuenta

que el precio por hora que se cobra en el mercado por un ingeniero no especializado en proceso que se dedica a la integración de sistemas de automatización y energía ronda entre los USD\$25 y USD\$35, y que para desarrollar la herramienta de software como se presenta en el presente trabajo tomo unas 50 horas de desarrollo entre el diseño y la programación Web que es básicamente la que absorbe todo el tiempo, puesto que la programación del PLC depende del proceso a monitorear y es externo al precio de esta herramienta y la configuración de Red es casi nula y también depende del PLC que de nuevo es externo a la herramienta, estamos hablando de un costo aproximado que oscilaría entre los USD\$1250.00 y USD\$1750.00, que es alrededor de tres a cuatro veces más barata que una licencia de las más sencillas de recopilación de datos como las mencionadas arriba.