



**UNIVERSIDAD DE CARABOBO**  
**FACULTAD DE INGENIERÍA**  
**ESCUELA DE INGENIERÍA MECÁNICA**



**IMPLEMENTACIÓN DE LA CONEXIÓN ENTRE UN CONTROLADOR  
LÓGICO PROGRAMABLE (PLC) Y UNA APLICACIÓN BASADA EN UN  
COMPUTADOR**

**MOSQUERA A., ROMER M.  
TURKINGTON W., TIMOTHY S.**

**VALENCIA, 2008**



**UNIVERSIDAD DE CARABOBO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA MECÁNICA**



**IMPLEMENTACIÓN DE LA CONEXIÓN ENTRE UN CONTROLADOR  
LÓGICO PROGRAMABLE (PLC) Y UNA APLICACIÓN BASADA EN UN  
COMPUTADOR**

**TRABAJO ESPECIAL DE GRADO PRESENTADO ANTE LA ILUSTRE  
UNIVERSIDAD DE CARABOBO PARA OPTAR AL TÍTULO DE  
INGENIERO MECÁNICO**

**MOSQUERA A., ROMER M.  
TURKINGTON W., TIMOTHY S.**

**VALENCIA, JULIO DE 2008**



**UNIVERSIDAD DE CARABOBO**  
**FACULTAD DE INGENIERÍA**  
**ESCUELA DE INGENIERÍA MECÁNICA**



Los abajo firmantes miembros del jurado designado para evaluar el Trabajo Especial de Grado titulado **“IMPLEMENTACIÓN DE LA CONEXIÓN ENTRE UN CONTROLADOR LÓGICO PROGRAMABLE (PLC) Y UNA APLICACIÓN BASADA EN UN COMPUTADOR”** realizado por los bachilleres **Mosquera A., Romer M. C.I. V-18.087.626** y **Turkington W., Timothy S. C.I. V-16.994.708**, hacemos constar que hemos revisado y aprobado dicho trabajo.

---

**Adriana Herrera**  
Tutor

---

**Napoleón González**  
Jurado

---

**Luis Escalona**  
Jurado



**UNIVERSIDAD DE CARABOBO**  
**FACULTAD DE INGENIERÍA**  
**ESCUELA DE INGENIERÍA MECÁNICA**



El presente trabajo especial de grado consiste en la implementación de la conexión entre un controlador lógico programable (PLC) y una aplicación basada en un computador. Esta conexión se realizó mediante el método de OLE for Process Control (OPC). Se utilizó como servidor OPC el S7-200 PC Access que es desarrollado por el fabricante Siemens y está certificado por la OPC Foundation. Como cliente OPC se utilizó OPC Soft que es una herramienta desarrollada por el Prof. Ángel Villegas de la Escuela de Ingeniería Eléctrica y se pudo comprobar la eficacia de la comunicación entre el S7-200 PC Access y OPC Soft, mediante dos aplicaciones las cuales son: (1) el control de tráfico en una intersección de vías y (2) el control de llenado y vaciado de un tanque. Ambas aplicaciones fueron controladas por un PLC Siemens S7-200 disponible en el Laboratorio de Instrumentación de la Escuela de Ingeniería Mecánica.



**UNIVERSIDAD DE CARABOBO**  
**FACULTAD DE INGENIERÍA**  
**ESCUELA DE INGENIERÍA MECÁNICA**



El presente trabajo especial de grado fue realizado gracias al financiamiento del Consejo de Desarrollo Científico y Humanístico (CDCH) el cual nos facilitó los recursos para la adquisición del software OPC Siemens S7-200 PC Access con el que se realizó la implementación de la conexión entre un controlador lógico programable (PLC) y una aplicación basada en un computador.



## DEDICATORIA

“Pero por la gracia de Dios soy lo que soy; y su gracia no ha sido en vano para conmigo” 1 Corintios 15:10

Realmente son muchas las personas a las cuales tengo que agradecerles, no solamente por estar conmigo en el desarrollo de la tesis sino también por haberme dado su apoyo a lo largo de toda la carrera.

“Doy gracias a mi Dios siempre...”, por el don inefable de la salvación en Cristo y por su gracia en sostenerme cada día y colmarme de toda bendición en los lugares celestiales en Cristo Jesús. Ha sido por su gracia que he podido culminar mis estudios, por lo tanto la gloria es y será siempre para Él.

Le doy gracias también a mis padres por haberse esforzado en darme una buena educación y por apoyarme en toda la carrera. Por haberme provisto de todo lo necesario en mi vida y en mis estudios.

A Raquel y Eleonor porque aunque estuvieron lejos al final de mi carrera siempre estaban pendientes de mí, aconsejándome y dándome apoyo.

Le doy gracias también a Ruth por ser tan especial conmigo, por su compañía y amor. Porque se preocupaba conmigo y siempre estuvo a mi lado ayudándome y animándome hasta ahora.

A Timothy por haberme aguantado con lo fastidioso que era con la tesis ☺ Fue una buena experiencia de aprendizaje haber compartido con él en la universidad no solamente por ser familia sino por ser un hermano en la fe que comparte los mismos intereses en el Señor que yo.

“El que se gloría, gloriése en el Señor” 1 Corintios 1:31

Romer M Mosquera



## DEDICATORIA

Este logro en la vida no hubiese sido posible sin la ayuda y el apoyo incondicional del Dios Creador y Todopoderoso y de su Hijo mi Salvador.

Por eso lo dedico a:

Mis Padres: Sidney y Ruth

Por su esfuerzo en instruirme en el Camino de la Vida

Por su ejemplo para mí en la vida

Y por su esfuerzo en educarme, mantenerme, y cuidarme en las primeras etapas de la vida.

A mis Hermanos Evelyn, Carol y Ryan

Por su paciencia para conmigo, su respaldo y ayuda

Por lo que significan para mí

Y por todo lo que han hecho por mí

A mis demás familiares y amigos en la vida que me han animado a seguir adelante

No importando las situaciones, conflictos o problemas

Y en especial a mi primo, amigo, hermano en Cristo y compañero de tesis

Romer Miguel por su mansedumbre, ejemplo y conocimiento

Sin la cual no hubiese cumplido esta etapa en la vida.

Y estas palabras que fueron de ayuda, fortaleza y guía en estos años de estudio

*La senda de los justos es como la luz de la aurora,*

*Que va en aumento hasta que el día es perfecto.*

*Proverbios 4:18*

*Guarda la ley y el consejo*

*Y serán vida a tu alma,*

*Y gracia a tu cuello.*

*Entonces andarás por tu camino confiadamente,*

*Y tu pie no tropezará*

*Porque el Señor será tu confianza.*

*Proverbios 3:21-26*

*Deléitate asimismo en Dios,*

*Y el te concederá las peticiones de tu corazón*

*Encomienda al Señor tu camino,*

*Y confía en él; y él hará*

*Salmo 3:4-5*



## AGRADECIMIENTOS

Agradecemos a la ilustre Universidad de Carabobo por habernos permitido realizar y culminar nuestros estudios universitarios.

A nuestra tutora la Prof. Adriana Herrera por habernos orientado a lo largo del desarrollo de la tesis y por haber estado disponible para atendernos.

Al Prof. Luis Escalona que nos ayudó cuando se nos presentaban dificultades con la tesis y también por habernos permitido utilizar el laboratorio cuando lo necesitábamos para realizar las pruebas con el OPC.

Agradecemos también al Prof. Ángel Villegas por su asesoría en el tema de OPC y por facilitarnos su programa OPCSoft con el cual desarrollamos las aplicaciones en el computador.

Romer Mosquera

Timothy Turkington



## RESUMEN

Este trabajo especial de grado tiene como finalidad plantear una solución al problema que existe en el Laboratorio de Automatización Industrial de la Escuela de Mecánica de la Facultad de Ingeniería, donde no se cuenta con el espacio ni los recursos económicos disponibles para implementar procesos industriales controlados por PLC, por lo cual se plantea la implementación de una aplicación basada en el computador la cual será controlada por el PLC.

Es necesario que esta interconexión sea a través de un estándar de comunicación que permita integrar diversos fabricantes de PLC, por lo cual se estableció la conexión mediante el estándar OPC (OLE for Process control). Para establecer el estándar OPC es necesario contar con un Servidor OPC, que permite la comunicación entre el PLC y la aplicación, y un Cliente OPC, donde es diseñada la aplicación que muestra gráficamente el proceso simulado. El Servidor OPC seleccionado fue el S7-200 PC Access mientras que el Cliente OPC seleccionado fue el OPCSoft.

La primera aplicación desarrollada consiste en el control de tráfico de una intersección de vías con cuatro semáforos cuya secuencia de luces es controlada por el PLC. La segunda aplicación desarrollada consiste en el control del llenado y vaciado de un tanque.

Estas aplicaciones permiten verificar la interconexión entre el computador y el PLC así como la factibilidad de implementar ésta alternativa como solución a la situación planteada.



## ÍNDICE GENERAL

INTRODUCCION .....	1
CAPÍTULO I: El Problema	
1.1.- Planteamiento del Problema .....	3
1.2.- Objetivos .....	4
1.2.1.- General .....	4
1.2.2.- Específicos .....	4
1.3.- Justificación de la investigación .....	5
1.4.- Limitaciones .....	6
1.3.- Delimitaciones .....	6
CAPÍTULO II: Marco Teórico	
2.1.- Antecedentes de la investigación .....	8
2.2.- Controlador Lógico Programable (PLC) .....	9
2.2.1.- Definición .....	9
2.2.2.- Campos de aplicación .....	9
2.2.3.- Ventajas e inconvenientes .....	12
2.2.4.- Funciones básicas de un PLC .....	13
2.2.5.- Estructura del PLC .....	16
2.2.5.1.- Estructura externa .....	16
2.2.5.2.- Estructura interna .....	18
2.2.6.- Funcionamiento .....	32
2.2.6.1.- Modo de funcionamiento .....	33
2.2.6.2.- Ciclo de funcionamiento .....	34
2.2.7.- Tiempo de ejecución y control en tiempo real .....	36
2.3.- OLE for Process Control (OPC) .....	39
2.3.1.- Introducción .....	39
2.3.1.1.- Definición .....	39
2.3.1.2.- Historia .....	39



2.3.1.3.- Aplicación de OPC .....	43
2.3.1.4.- Arquitectura OPC .....	45
2.3.1.5.- Adaptación a la Aplicación del Cliente .....	45
2.3.1.6.- Métodos de Aplicación .....	46
2.3.1.7.- Esquema General .....	47
2.3.1.8.- Funcionalidad OPC .....	47
2.3.2.- Bases de OPC .....	49
2.3.2.1.- Objetos e interfases de OPC .....	49
2.3.2.2.- Perspectiva General del Acceso de Datos OPC .....	50
2.3.2.3.- Perspectiva General de la Gestión de Alarmas y Eventos OPC .....	52
2.3.2.4.- Perspectiva General del Acceso de Datos Históricos OPC .....	53
2.3.2.5.- Integración OPC .....	54
2.3.2.6.- Arquitectura OPC general y componentes .....	54
2.3.2.7.- Servidores remotos vs. Locales .....	55
2.3.3.- Especificaciones OPC .....	56
2.3.3.1.- OPC Common .....	56
2.3.3.2.- OPC Data Access (OPC DA) .....	59
2.3.3.3.- OPC Alarm and Events (OPC A&E) .....	66
2.3.3.4.- OPC HDA 1.X .....	73
2.3.3.5.- OPC Batch .....	76
2.3.3.6.- OPC DX .....	78
2.3.3.7.- OPC XML .....	79
2.3.4.- Ejemplo de la conexión en una red industrial utilizando OPC .....	80
2.3.5.- Cliente OPC .....	82
2.4.- Definición de términos básicos .....	83
<b>CAPÍTULO III: Marco Metodológico</b>	
3.1.- Nivel de la investigación .....	86
3.2.- Diseño de la investigación .....	86
<b>CAPÍTULO IV: Desarrollo de la investigación</b>	
4.1.- Formas de comunicación entre un computador y un PLC .....	89



4.2.- Selección del método de interconexión entre el PLC y el computador .....	93
4.3.-Arquitectura de conexión .....	98
4.3.1.- PLC Siemens S7 200 .....	99
4.3.2.- Cable Convertidor RS-485/USB .....	105
4.3.3.- Computador .....	105
4.3.3.1.- Servidor OPC .....	105
4.3.3.2.- Cliente OPC .....	105
4.4.- Desarrollo de la aplicación en el computador .....	106
4.4.1.- Selección del software que soportará la aplicación en el computador (cliente OPC) .....	106
4.4.2.- Diseño de aplicaciones en el computador con base en el software seleccionado que serán controladas por el PLC .....	110
4.4.2.1.- Control de tráfico en una intersección de vías .....	110
4.4.2.2.- Control del llenado y vaciado de un tanque .....	131
4.5.- Implementación del sistema de comunicación entre el controlador lógico programable y el computador .....	134
4.5.1.- Control de tráfico en una intersección de vías .....	135
4.5.2.- Control del llenado y vaciado de un tanque .....	142
<b>CAPÍTULO V: Conclusiones y Recomendaciones</b>	
5.1.- Conclusiones .....	146
5.2.- Recomendaciones .....	147
Referencias Bibliográficas .....	148
Apéndices .....	149



## INDICE DE TABLAS

Tabla 2.1.- Historia de OPC .....	39
Tabla 4.1.- Comparación entre Métodos de Interconexión entre el PLC y el PC .....	93
Tabla 4.2.- Características del CPU 224 .....	101
Tabla 4.3.- Especificaciones de las Entradas y Salidas .....	101
Tabla 4.4.- Especificaciones de la Memoria .....	102
Tabla 4.6.- Especificaciones dimensionales del PLC S7 .....	104
Tabla 4.7.- Variables de entrada .....	117
Tabla 4.8.- Variables de salida .....	118
Tabla 4.9.- Variables para la aplicación del Tanque .....	132



## INDICE DE FIGURAS

Figura 2.1.- Ejemplo de un PLC: Micrologix 1500 .....	9
Figura 2.2.- Sistema de control de los procesos de tratamiento en cámaras de naranja .....	10
Figura 2.3.- Sistema de control para soplado de plástico .....	11
Figura 2.4.- Maquinaria de embalajes, aplicación Logo .....	11
Figura 2.5.- Control del sistema de aire acondicionado .....	11
Figura 2.6.- Control de señales de tráfico .....	12
Figura 2.7.- Funciones del PLC .....	15
Figura 2.8.- Autómatas compactos .....	17
Figura 2.9.- Autómatas semimodulares .....	17
Figura 2.10.- Autómatas Modulares .....	18
Figura 2.11.- Arquitectura básica de un autómata programable .....	19
Figura 2.12.- Distribución de la memoria en el autómata .....	29
Figura 2.13.- Resumen de la partes del PLC .....	32
Figura 2.14.- Secuencia básica de operación del autómata .....	33
Figura 2.15.- Ciclo de funcionamiento del autómata .....	34
Figura 2.16.- Esquema general .....	44
Figura 2.17.- Arquitectura OPC .....	45
Figura 2.18.- Esquema de funcionalidad de OPC .....	48
Figura 2.19.- Cliente OPC .....	49
Figura 2.20.- Relación Cliente/Servidor OPC .....	50
Figura 2.21.- Relación grupo/item .....	51
Figura 2.22.- Relación Cliente/Servidor OPC .....	54
Figure 2.23.- Interfaces OPC .....	54
Figura 2.24.- Arquitectura típica de OPC .....	55
Figura 2.25.- Estructura de los objetos Group e Item en el servidor .....	60
Figura 2.26.- Modelo de objetos lógico .....	61



Figura 2.27.- Diseño de un servidor OPC DA .....	63
Figura 2.28.- Modelo lógico de un objeto servidor A&E .....	68
Figura 2.29.- Diseño de un servidor .....	69
Figura 2.30.- OPCEventServer .....	71
Figura 2.31.- OPC EventSubscription .....	72
Figura 2.32.- OPC EventAreaBrowser .....	73
Figura 2.33.- Interfaz del cliente OPC HDA .....	75
Figura 2.34.- Interfaz del servidor OPC HDA .....	76
Figura 2.35.- Interfaces del servidor OPC Batch .....	78
Figura 2.36.- OPC DX .....	79
Figura 2.37.- Esquema de instalación para el caso estudiado .....	81
Figura 2.38.- Esquema con la solución OPC .....	82
Figura 4.1.- Controladores conectados a través de E/S digitales .....	89
Figura 4.2.- Interfases CANopen, una forma de comunicación industrial .....	91
Figura 4.3.- Método de conectividad por medio de un OPC .....	92
Figura 4.4.- Ambiente de trabajo de PC Access .....	96
Figura 4.5.- Arquitectura de conexión .....	99
Figura 4.6.- Identificación de las partes del Siemens S7-CPU224 .....	100
Figura 4.7.- Identificación de las partes del Siemens S7-CPU224 .....	100
Figura 4.8.- Especificaciones para el montaje .....	103
Figura 4.9.- Diagramas de Cableado para corriente Directa .....	104
Figura 4.10.- Diagramas de Cableado para corriente Continua .....	105
Figura 4.11.- Control de Procesos por LabVIEW .....	107
Figura 4.12.- Diagrama de bloques con VisSim .....	109
Figura 4.13.- Excel como cliente OPC .....	110
Figura 4.14.- Fase Inicial de la aplicación desarrollada en el Computador .....	111
Figura 4.15.- Sincronización de los Semáforos .....	112
Figura 4.16. Diagrama funcional del semáforo .....	113
Figura 4.17.- Agregar un nuevo PLC .....	114
Figura 4.18.- Propiedades del PLC .....	114



Figura 4.19.- Carpetas creadas en el servidor OPC .....	115
Figura 4.20.- Propiedades del ítem (entradas) .....	116
Figura 4.21.- Propiedades del ítem (salidas) .....	117
Figura 4.22.- Variables de entrada .....	119
Figura 4.23.- Variables de salida (Luces Verdes) .....	119
Figura 4.24.- Ventana principal de OPC Soft .....	120
Figura 4.25.- Interruptor con Led en OPC Soft .....	121
Figura 4.26.- Cambiar el nombre del pulsador .....	122
Figura 4.27.- Ventana de propiedades del pulsador .....	123
Figura 4.28.- Seleccionar la propiedad del pulsador .....	124
Figura 4.29.- Seleccionando el OPC .....	124
Figura 4.30.- Seleccionar el Servidor OPC S7200.OPCServer .....	125
Figura 4.31.- Seleccionar el Ítem .....	125
Figura 4.32.- Identidad del Ítem .....	126
Figura 4.33.- Interruptor con Led en OPC Soft .....	126
Figura 4.34.- Ventana de propiedades del Led en OPC Soft .....	127
Figura 4.35.- Seleccionar la propiedad del led .....	128
Figura 4.36.- Seleccionar el OPC .....	128
Figura 4.37.- Servidor OPC S7200.OPCServer .....	129
Figura 4.38.- Seleccionando el Ítem .....	129
Figura 4.39.- Identidad del Ítem .....	129
Figura 4.40.- Área de proceso con los semáforos .....	130
Figura 4.48.- Ejemplo de llenado de un tanque .....	131
Figura 4.41.- Etapa 0 .....	135
Figura 4.42.- Etapa 1 .....	136
Figura 4.43.- Etapa 2 .....	137
Figura 4.44.- Etapa 3 .....	138
Figura 4.45.- Etapa 4 .....	139
Figura 4.46.- Etapa 5 .....	140
Figura 4.47.- Etapa 6 .....	141



---

Figura 4.48.- Tanque vacío .....	142
Figura 4.49.- Tanque parcialmente lleno .....	143
Figura 4.50.- Tanque lleno .....	144



## INTRODUCCIÓN

En la actualidad los procesos industriales son cada vez más complejos y exigentes lo cual obliga a los controladores y redes industriales a manejar un mayor número de variables y a enviar y recibir datos con una mayor rapidez. La complejidad en las redes industriales ha aumentado los costos de adquisición, instalación y mantenimiento de dichos procesos automatizados. Para reducir estos costos se han desarrollado herramientas para simular la automatización de estos procesos industriales que permiten visualizar el desempeño de los diferentes elementos que componen dicho proceso.

Estas mismas herramientas pueden ser utilizadas de manera didáctica para interconectar un controlador y una computadora y ser implementados en laboratorios donde no se cuenta con el espacio y recursos económicos suficientes para construir y controlar dichos procesos en físico.

En el presente trabajo de grado se plantea una alternativa de solución donde son utilizadas estas herramientas para combatir los altos costos de implementación de estos procesos y la falta de espacio físico que se tiene actualmente en el laboratorio.

El trabajo está constituido de la siguiente manera, en el capítulo I se plantea la situación problemática, los objetivos que se quieren alcanzar, además de las limitaciones, delimitaciones y alcance. En el capítulo II se presenta una documentación teórica sobre el Controlador Lógico Programable (PLC), el estándar de comunicación OPC, algunas definiciones básicas y antecedentes de la investigación. En el capítulo III se plantea la metodología que se empleó así como las diferentes etapas de la investigación. En el capítulo IV se presentan las formas de comunicación entre un PLC y un computador, la selección del Servidor y el Cliente OPC, se especifica la arquitectura de comunicación, el desarrollo de la aplicación en el computador y su respectiva implementación. En el capítulo V se plantean las conclusiones y recomendaciones. Por último, se anexan algunos apéndices donde se detalla el procedimiento seguido para la selección del software y el código de programación del PLC utilizado para controlar las aplicaciones desarrolladas.



# **CAPÍTULO I**

## **EL PROBLEMA**



## 1.1.- PLANTEAMIENTO DEL PROBLEMA.

El rango de posibles aplicaciones en las que pueden emplearse los recursos disponibles en los Controladores Lógicos Programables (PLC), adquiridos por el laboratorio de Automatización e Instrumentación Industrial, están restringidas por algunos factores.

La necesidad de optimizar la utilización de los recursos disponibles del PLC, pueden solventarse de dos formas distintas:

- De una Forma Física: a través del diseño y construcción de aplicaciones y/o procesos donde puedan emplearse dichos recursos de control.
- De una Forma Virtual: a través de la simulación de aplicaciones y/o procesos basados en PC.

Al considerar y evaluar la primera opción mencionada, se concluye que ésta conlleva a limitaciones económicas y de espacio físico. La limitación económica, quizás sea la más preponderante, ya que se necesitaría diseñar, construir e instalar algún proceso industrial representativo u otra aplicación parecida, lo que acarrearía la necesidad de disponer de significativos recursos económicos o grandes fondos de los que no se disponen en la actualidad. La limitación de espacio físico, se refiere a la disponibilidad inexistente de ubicar una aplicación representativa en las instalaciones del laboratorio. Y por último, a lo sumo se tendrían dos aplicaciones representativas que restringirían el aprovechamiento total de los recursos disponibles en los PLC, es decir, siempre se trabajaría con la misma aplicación y por ende se presenta el problema de que sólo se puede desarrollar una única aplicación que ocasiona limitación en la enseñanza.

La segunda opción permitiría, la simulación de una aplicación desarrollada en un computador personal gobernada por el Controlador Lógico Programable, cuyas ventajas son las siguientes: permite desarrollar un amplio rango de procesos distintos, es decir, permite una mayor flexibilidad favorable para el aprovechamiento de los recursos del PLC y para el proceso de enseñanza didáctica. Además, no requiere de un amplio espacio físico ya que la



aplicación a desarrollar estará simulada en el computador. Y la diferencia económica con respecto al caso anterior, es sustancial.

Para solventar el problema del mejor aprovechamiento de los recursos del laboratorio basados en la forma virtual, se requerirá de una herramienta que permita desarrollar aplicaciones en un computador personal y se necesitaría, entonces, de la: “IMPLEMENTACIÓN DE LA CONEXIÓN ENTRE UN CONTROLADOR LÓGICO PROGRAMABLE (PLC) Y DICHA APLICACIÓN BASADA EN UN COMPUTADOR” para su posterior manipulación, modificación y control. Dicha interfaz de comunicación debe ser estándar, porque se requiere que la conexión sea compatible con la mayor cantidad de fabricantes de PLC posibles.

## **1.2.- OBJETIVOS.**

### **1.2.1.- General**

- Implementar la conexión entre un Controlador Lógico Programable (PLC) y una aplicación basada en un computador.

### **1.2.2.- Específicos**

- Indagar acerca de los diversos métodos de interconexión entre un controlador lógico programable y un computador.
- Seleccionar el método de interconexión entre el controlador lógico programable y el computador con el que se puedan integrar diversos fabricantes de PLC.
- Establecer la arquitectura necesaria para el método de interconexión seleccionado.
- Seleccionar el software en el que se soportará el desarrollo de la aplicación en el PC y que muestre gráficamente la interacción entre el PLC y el PC.
- Diseñar la aplicación en el computador con base en el software seleccionado que será controlada por el PLC.



- Implementar el sistema de comunicación entre el controlador lógico programable y el computador.

### **1.3.- JUSTIFICACIÓN DE LA INVESTIGACIÓN.**

Actualmente, es prácticamente imposible visualizar un proceso industrial que esté completamente ajeno a la automatización. Se sabe que el PLC es una de las herramientas más utilizadas y más poderosas en el campo de la automatización de procesos por lo que es importante que todo ingeniero conozca como trabajar con estos dispositivos. Parte del perfil del Ingeniero Mecánico es desarrollar y aplicar nuevas tecnologías para la producción de bienes y servicios y aunque el PLC tiene muchos años en el mercado, sigue siendo una tecnología de vanguardia para la automatización de procesos.

La interconexión entre un computador y un PLC permite desarrollar aplicaciones que puedan ser visualizadas gráficamente en la computadora mientras que son controladas mediante el PLC, tiene bajo costo de mantenimiento (o mejor dicho, el costo es nulo), permite ser utilizada muchas veces sin tener desgaste, la particularización de cada proceso (o aplicación que se vaya a desarrollar), la rápida incorporación de nuevas tecnologías, el bajo costo por función, etc.

Se espera que con este trabajo de grado los ingenieros mecánicos egresados de la Universidad de Carabobo puedan aprender de manera efectiva las múltiples funciones y ventajas que ofrecen los PLC, además que se logra una mejor dotación del laboratorio de automatización industrial.



## **1.4.- LIMITACIONES**

- El PLC disponible para desarrollar este trabajo será el SIEMENS S7-200.
- Recursos económicos para la adquisición de los equipos necesarios, limitados al monto de las ayudas menores del CDCH.

## **1.5.- DELIMITACIONES**

Este trabajo especial de grado tiene como alcance poder establecer la comunicación entre un PLC con un computador de forma tal que el PC aloje una aplicación a ser controlada, en la cual pueda apreciarse la ejecución de órdenes provenientes del PC y la emisión de señales hacia el PLC. Las señales con las que se trabajarán en este trabajo especial de grado, son señales digitales.



# **CAPÍTULO II**

## **MARCO TEÓRICO**



## 2.1.- ANTECEDENTES DE LA INVESTIGACIÓN.

Los siguientes antecedentes son trabajos especiales de grado que se han desarrollado utilizando controladores lógicos programables en diversas aplicaciones industriales, así como han implementado software que controlen los mismos y han desarrollado prácticas para algunos laboratorios que involucran controladores lógicos programables.

- En Noviembre del 2001 Fabiola Vitoria y Oscar Kalnins desarrollaron un trabajo especial de grado titulado **“Laboratorio para la enseñanza de la automatización industrial usando la técnica de SOFT-PLC”**, donde diseñaron un conjunto de prácticas para la materia de Automatización Industrial 2 (de la escuela de Ingeniería Eléctrica), las cuales están basadas en las técnicas de SOFT-PLC usando el software InControl de Wonderware, el cual está diseñado para ejecutarse desde un computador personal.
- En Diciembre de 2004 Christian Perdomo y Luís Rosales desarrollaron un trabajo especial de grado titulado **“Diseño e implementación de un Laboratorio y manual de prácticas de PLC para los alumnos del Ciclo técnico profesional de la Escuela Técnica Industrial Mariano Fernández Fortique de Cagua en el Estado Aragua”** donde elaboraron un manual de prácticas que permite a los estudiantes de dicha institución adquirir conocimientos en el área de Automatización utilizando PLC. Además seleccionaron los PLC adecuados para dicho laboratorio.
- En Julio del 2005 Jesús Márquez y José Pérez desarrollaron un trabajo especial de grado titulado **“Diseño, análisis y simulación del control automático de columnas de destilación bajo ambiente LABView”**, donde diseñaron un Software para simular el comportamiento de las columnas de destilación y sus respectivos procesos controlados por un PLC

## 2.2.- CONTROLADOR LÓGICO PROGRAMABLE (PLC) ([1] y [2])

### 2.2.1.- Definición

Un autómatas programable industrial (API) o Programmable Logic Controller (PLC), es un equipo electrónico, programable en lenguaje no informático, diseñado para controlar en tiempo real y en ambiente de tipo industrial, procesos secuenciales (figura 2.1).

Los procesos secuenciales son aquellos donde una misma combinación de señales o informaciones de entrada pueden dar una salida diferente según el orden en que se hayan sucedido los acontecimientos.

Un PLC trabaja en base a la información recibida por los captadores y el programa lógico interno, actuando sobre los accionadores de la instalación.



Figura 2.1 Ejemplo de un PLC: Micrologix 1500. Fuente: [www.ab.com/plclogic/](http://www.ab.com/plclogic/)

### 2.2.2.- Campos de aplicación

El PLC por sus especiales características de diseño tiene un campo de aplicación muy extenso. La continua evolución del hardware y software amplía constantemente este campo para poder satisfacer las necesidades que se detectan.

Su utilización se da fundamentalmente en aquellas instalaciones en donde es necesario un proceso de maniobra, control, señalización, etc. Por tanto, su aplicación abarca desde procesos de fabricación industriales de cualquier tipo a transformaciones industriales, control de instalaciones, etc.

Sus reducidas dimensiones, la facilidad de su montaje, la posibilidad de almacenar los programas para su posterior y rápida utilización, la modificación o alteración de los mismos, etc., hace que su eficacia se aprecie fundamentalmente en procesos donde existen necesidades tales como:

- Espacio reducido
- Procesos de producción periódicamente cambiantes
- Procesos secuenciales
- Maquinaria de procesos variables
- Instalaciones de procesos complejos y amplios
- Chequeo de programación centralizada de las partes del proceso

Ejemplos de aplicaciones generales:

- Maniobra de máquinas



**Figura 2.2 Sistema de control de los procesos de tratamiento en cámaras de naranja.**

Fuente: [www.soluciones.omron.es/soluciones/index.html](http://www.soluciones.omron.es/soluciones/index.html)

- Maquinaria industrial de plástico



Figura 2.3 Sistema de control para soplado de plástico. Fuente: [www.soluciones.omron.es/](http://www.soluciones.omron.es/)

- Maquinaria de embalajes



Figura 2.4 Maquinaria de embalajes, aplicación Logo. Fuente: [www.grupo-maser.com/](http://www.grupo-maser.com/)

- Instalación de aire acondicionado y/o calefacción.



Figura 2.5 Control del sistema de aire acondicionado. Fuente: <http://www.automation.siemens.co.uk>

- Señalización y control



Figura 2.6 Control de señales de tráfico. Fuente: <http://www.automation.siemens.co.uk>

- Chequeo de programas
- Instalaciones de seguridad
- Maniobra de instalaciones

### 2.2.3.- Ventajas e inconvenientes

No todos los autómatas ofrecen las mismas ventajas sobre la lógica cableada, ello es debido, principalmente, a la variedad de modelos existentes en el mercado y las innovaciones técnicas que surgen constantemente. Para un autómata de tipo medio:

#### **Ventajas:**

- Menor tiempo empleado en la elaboración de proyectos debido a que:
  - No es necesario dibujar el esquema de contactos
  - No es necesario simplificar las ecuaciones lógicas, ya que, por lo general la capacidad de almacenamiento del módulo de memoria es lo suficientemente grande.



- La lista de materiales queda sensiblemente reducida, y al elaborar el presupuesto correspondiente se elimina parte del problema que supone el contar con diferentes proveedores, distintos plazos de entrega.
- Posibilidad de introducir modificaciones sin cambiar el cableado ni añadir aparatos.
- Mínimo espacio ocupado.
- Menor coste de mano de obra de la instalación.
- Economía de mantenimiento. Además de aumentar la fiabilidad del sistema, al eliminar contactos móviles, los mismos autómatas pueden indicar y detectar averías.
- Posibilidad de gobernar varias máquinas con un mismo autómata.
- Menor tiempo para la puesta en funcionamiento del proceso al quedar reducido el tiempo de cableado.
- Si por alguna razón la máquina queda fuera de servicio, el autómata sigue siendo útil para otra máquina o sistema de producción.

#### **Inconvenientes:**

- Es necesario un programador, lo que obliga a adiestrar a uno de los técnicos en tal sentido.
- El coste inicial no es trivial.

#### **2.2.4.- Funciones básicas de un PLC**

El autómata programable debe realizar multitud de funciones y muchas de ellas simultáneamente, las funciones más clásicas son:



- **Detección:** Lectura de la señal de los captadores distribuidos por el sistema automatizado.
- **Mando:** Elaborar y enviar las acciones al sistema mediante los accionadores y preaccionadores.
- **Interfaz hombre-máquina (Human Machine Interface):** Mantener interacción con los operarios de producción, obedeciendo sus consignas e informando del estado del proceso.
- **Programación:** Para introducir, elaborar y cambiar el programa de aplicación del autómeta. El diálogo de programación debe permitir modificar el programa incluso con el autómeta controlando la máquina (online).

### **Nuevas Funciones:**

En los últimos años, en el campo de la automatización industrial se ha incorporado toda una gama de nuevas funcionalidades.

- **Redes de comunicación:** Permiten establecer comunicaciones con otras partes de control. Las redes industriales permiten la comunicación y el intercambio de datos entre autómetas a tiempo real.
- **Sistemas de supervisión:** Permiten comunicarse con ordenadores provistos de programas de supervisión industrial. Esta comunicación se realiza por una red industrial o por medio de una simple conexión por el puerto serie del ordenador.
- **Control de procesos continuos:** Además de dedicarse al control de sistemas de eventos discretos los autómetas llevan incorporadas funciones que permiten el control de procesos continuos. Disponen de módulos de entrada y salida analógicas y la posibilidad de ejecutar reguladores PID que están programados en el autómeta.

- Entradas/salidas distribuidas: Los módulos de entrada/salida no tienen porqué estar en el armario del autómeta, pueden estar distribuidos por la instalación. Se comunican con la unidad central del autómeta mediante un cable de red.
- Buses de campo: Mediante un solo cable de comunicación se pueden conectar al bus captadores y accionadores, reemplazando al cableado tradicional. El autómeta consulta cíclicamente el estado de los captadores y actualiza el estado de los accionadores.

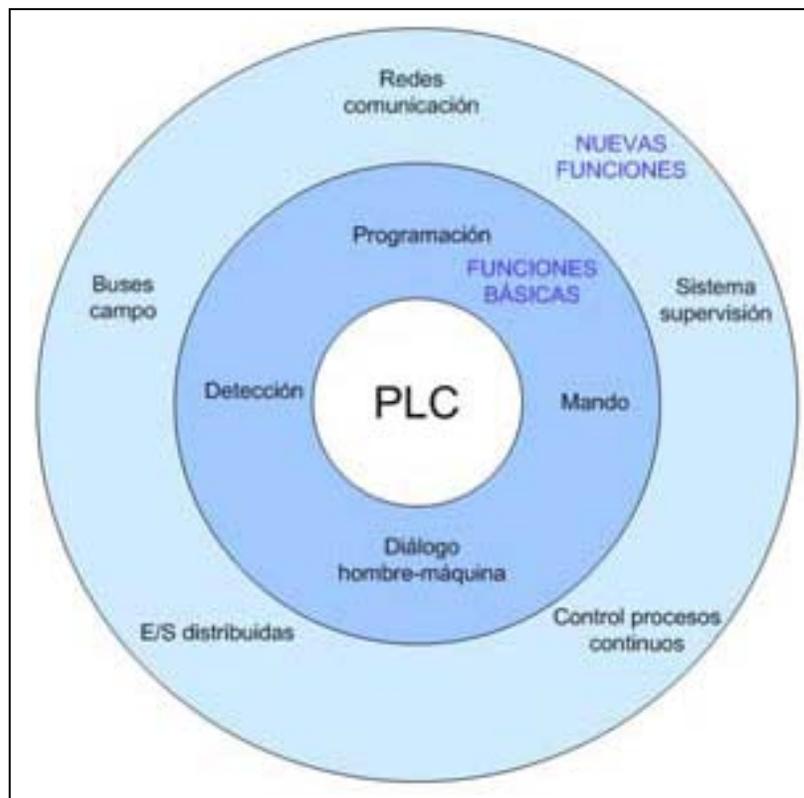


Figura 2.7 Funciones del PLC. Fuente: <http://automaindus.googlepages.com>



## 2.2.5.- Estructura del PLC

Este apartado está dedicado a conocer al autómatas en su parte física o hardware, no sólo en su configuración externa, sino también y fundamentalmente en la parte interna.

El autómatas está compuesto de diferentes elementos como CPU, fuente de alimentación, memoria, módulos de entradas y salidas (E/S), etc. que están colocados de diferente forma y modo según la estructura externa del autómatas.

### 2.2.5.1.- Estructura externa

El término estructura externa o configuración externa de un autómatas programable industrial se refiere al aspecto físico exterior del mismo, es decir, bloques o elementos en que está dividido.

Actualmente son tres las estructuras más significativas que existen en el mercado:

- Estructura compacta.
- Estructura semimodular (Estructura Americana).
- Estructura modular (Estructura Europea).

#### **Estructura compacta**

Este tipo de autómatas se distingue por presentar en un solo bloque todos sus elementos: fuente de alimentación, CPU, memorias, módulos de entradas y salidas (E/S), etc.

Son los autómatas de gama baja o nanoautómatas los que suelen tener una estructura compacta. Su potencia de proceso suele ser muy limitada, dedicándose a controlar máquinas muy pequeñas o cuadros de mando (**figura 2.8**).



Figura 2.8 Autómatas compactos.

### Estructura semimodular

Se caracteriza por separar las E/S del resto del autómata, de tal forma que en un bloque compacto están reunidas las CPU, memoria de usuario o de programa y fuente de alimentación y separadamente las unidades de E/S (figura 2.9). Son los autómatas de gama media los que suelen tener una estructura semimodular (Americana).



Figura 2.9 Autómatas semimodulares.

### Estructura modular

Su característica principal es que existe un módulo para cada uno de los diferentes elementos que componen el autómata: fuente de alimentación, CPU, Módulos de entradas y salidas (E/S), etc. La sujeción de los mismos se hace por carril DIN, placa perforada o sobre RACK.



Son los autómatas de gama alta los que suelen tener una estructura modular, que permite una gran flexibilidad en su constitución. También facilita su mantenimiento debido a que si algún elemento falla, puede ser rápidamente sustituido (figura 2.10).



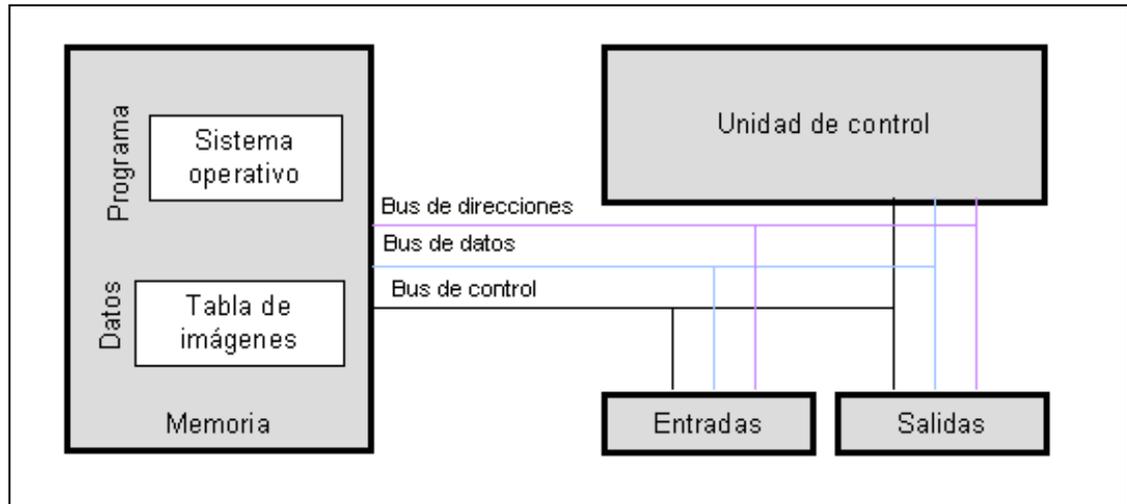
Figura 2.10 Autómatas Modulares.

### 2.2.5.2.- Estructura interna

En este apartado se profundizará en la estructura interna de cada uno de los diferentes elementos que componen el autómata, sus funciones y su funcionamiento.

El autómata está constituido por diferentes elementos, pero tres son los básicos (figura 2.11):

- Unidad de control o CPU
- Unidades de entradas y salidas
- Unidad de memoria



**Figura 2.11 Arquitectura básica de un autómata programable.**

Fuente: Boix, O., [et al.] Tecnología eléctrica. Barcelona, Ediciones Ceysa 2002, p. 399-405

El medio físico a través del cual el procesador se comunica con el resto de elementos del sistema son los buses de comunicación. Normalmente, existen tres tipos:

- Bus de direcciones: el procesador envía la dirección del elemento al cual quiere enviar o que le envíe información.
- Bus de datos: es por donde todos los elementos enviarán los datos.
- Bus de control: es aquel mediante el cual el procesador explica que operación se está efectuando.

Con las partes mencionadas puede decirse que se tiene un autómata pero para que sea operativo son necesarios otros elementos:

- Fuente de alimentación
- Interfaces
- La unidad o consola de programación
- Los dispositivos periféricos



## **CPU (Central Processing Unit)**

La CPU (Central Processing Unit) es la parte inteligente del autómata. Interpreta las instrucciones del programa de usuario de forma secuencial y consulta el estado de las entradas. Dependiendo de dichos estados y del programa, ordena la activación de las salidas deseadas.

La capacidad de cálculo y la velocidad de procesamiento dependen del número y los tipos de procesadores que tenga. La mayor parte de los autómatas tienen una CPU con un solo procesador, pero cada vez hay más, que tienen las funciones descentralizadas entre varios procesadores normalmente diferentes.

La CPU está constituida por los siguientes elementos:

- Procesador
- Memoria monitor del sistema
- Circuitos auxiliares

### Procesador:

Está constituido por el microprocesador, el reloj (generador de onda cuadrada) y algún chip auxiliar. El microprocesador es un circuito integrado (chip), que realiza una gran cantidad de operaciones, que pueden agruparse en:

- Operaciones de tipo lógico.
- Operaciones de tipo aritmético.
- Operaciones de control de la transferencia de la información dentro del autómata.

Para que el microprocesador pueda realizar todas estas operaciones está dotado de unos circuitos internos:



- Circuitos de la unidad aritmética y lógica o ALU: Es la parte del microprocesador donde se realizan los cálculos y las decisiones lógicas para controlar el autómata.
- Circuitos de la unidad de control (UC) o Decodificador de instrucciones: Decodifica las instrucciones leídas en memoria y genera las señales de control.
- Acumulador: Es el encargado de almacenar el resultado de la última operación realizada por el ALU.
- Flags: también llamados indicadores de resultado, que pueden ser consultados por el programa.
- Contador de programa: Encargado de la lectura de las instrucciones de usuario.
- Bus (interno): No son circuitos en si, sino zonas conductoras en paralelo que transmiten datos, direcciones, instrucciones y señales de control entre las diferentes partes del microprocesador.

### Memoria monitor del sistema

En la memoria ROM del sistema (memoria de solo lectura), el fabricante ha grabado una serie de programas ejecutivos, software del sistema y es a estos programas a los que accederá el microprocesador para realizar las funciones. Estas funciones se realizan en determinados tiempos de cada ciclo y son:

- Vigilar que el tiempo de ejecución del programa de usuario no exceda de un determinado tiempo máximo. A esta función se le denomina Watchdog.
- Ejecutar el programa usuario de forma secuencial.



- Crear una imagen de las entradas, ya que el programa de usuario no debe acceder directamente a dichas entradas.
- Renovar el estado de las salidas en función de la imagen de las mismas, obtenida al final del ciclo de ejecución del programa usuario.
- Chequeo del sistema: inicialización tras puesta en tensión o reset, rutinas de test y de respuesta a error de funcionamiento.

## **Entradas y salidas**

Los módulos o secciones de entrada y salida son los que permiten comunicar el autómatas con el proceso que está controlando y con el operador.

La sección de entradas mediante la interfaz, adapta y codifica de forma comprensible para la CPU las señales procedentes de los dispositivos de entrada o captadores.

La sección de salida también mediante interfaz trabaja de forma inversa a las entradas, es decir, decodifica las señales procedentes de la CPU, las amplifica y las envía a los dispositivos de salida: preactuadores o actuadores (lámparas, relés, etc.).

Hay dos tipos de entradas o salidas según el tipo de señal que gestionan:

- Digitales o binarias, las normales del autómatas
- Analógicas

Las entradas o salidas pueden clasificarse también según su conexión:

- Locales: Dentro del mismo armario del autómatas, sin utilizar el procesador de comunicaciones. Así pueden ser compactas (conexión por bus interno) o modulares (conexión por bus de expansión).



- Remotas: En armarios remotos, utilizando procesadores de comunicaciones. En este caso también pueden ser compactas (conexión a CPU por bus local) o modulares (bus local + bus de expansión).

### Entradas digitales

Los módulos de entrada digitales permiten conectar el autómatas a captadores de tipo todo o nada (finales de carrera, pulsadores, etc.), en los que se da una información cualitativa y no cuantitativa.

Los módulos de entrada digitales trabajan con señales de tensión, por ejemplo cuando por una vía llegan 24 VCC. se interpreta como un "1" y cuando llegan 0 V. se interpreta como un "0". Los niveles de tensión estándar para los autómatas son: 24 VCC (el más común), 110 VCA o 220 VCA. El hecho que las tensiones sean tan elevadas se debe a la gran longitud que suelen tener los cables hasta el captador en las instalaciones industriales, que puede provocar ruido. Con este voltaje se asegura que la señal llegue al módulo de entrada. De esta manera los captadores tampoco necesitan una fuente de alimentación extra.

El proceso de adquisición de la señal digital consta de varias etapas.

- Protección contra sobretensiones, si la corriente es alterna se rectifica
- Filtrado de posibles ruidos que entran por el cable del captador. El tiempo de filtrado se puede programar en algunos módulos. Esto puede llevar problemas a la hora de utilizar captadores de alta frecuencia, para contajes más rápidos existen entradas especiales que pueden contar pulsos a altas frecuencias.
- Puesta en forma de la onda
- Aislamiento galvánico o por optoacoplador.



Una vez terminado este proceso la señal se deposita como un "0" o "1" en la memoria del módulo.

### Entradas analógicas

Los módulos de entrada analógicas permiten que los autómatas programables trabajen con introductores de señal analógicos y lean señales de este tipo como pueden ser la temperatura, la presión o el caudal, es decir, reciben una tensión (o una corriente) variable y que este valor se convierta en una variable numérica del autómata.

Lo que realiza es una conversión A/D (analógico/digital), puesto que el autómata solo trabaja con señales digitales. Esta conversión se realiza con una precisión o resolución determinada (número de bits: 8 -10 bits) y cada cierto intervalo de tiempo (período de muestreo). Es decir, la magnitud analógica se convierte en un número que se deposita en una variable interna del tipo palabra del autómata.

Los módulos de entrada analógica pueden manejar tensión o intensidad en unos rangos comprendidos de 0 a 10 V, -10 a 10 V, -20 a 20 mA, 0 a 20 mA o 4 a 20 mA.

El proceso de adquisición de la señal analógica consta de varias etapas:

- Filtrado
- Conversión A/D
- Memoria interna

Conviene utilizar cable trenzado y apantallado (con la pantalla conectada a tierra en uno de los extremos) para las entradas analógicas y cortocircuitar o unir a masa aquellas entradas por tensión que no se hayan de utilizar ya que si se dejan sin conectar pueden entrar perturbaciones electromagnéticas que pueden afectar a las otras entradas.



### Salidas digitales

Un módulo de salida digital permite al autómeta programable actuar sobre los preaccionadores y accionadores que admitan órdenes del tipo todo o nada.

Existen dos tipos de salidas digitales:

- Módulos electromecánicos: los elementos que conmutan son contactos de relés internos al módulo. Pueden usarse tanto para corriente alterna como continua. Las salidas a relé son libres de tensión y ésta debe provenir de una fuente de alimentación (interna o externa al autómeta) conectada al común del contacto.
- Módulos de salida estáticos (bornero): los elementos que conmutan son componentes electrónicos como transistores (corriente continua, 24 V.CC) o triacs (corriente alterna, 110V.CA).

Los módulos de salidas estáticos, al emitir señales de tensión eléctrica, solo pueden actuar sobre elementos que trabajan todos a la misma tensión, en cambio los módulos de salida electromecánicos (relés), como constan de salidas a contacto seco, pueden actuar sobre elementos que trabajen a tensiones distintas. Tienen el inconveniente de que la velocidad de respuesta es pequeña y, por tanto, no pueden emplearse en aplicaciones que requieran cambios rápidos en las salidas.

Los módulos de salida estáticos a 24 V.CC son menos costosos que los electromecánicos, pero muchas veces hay que efectuar una activación en cascada mediante un relé para poder actuar sobre el accionador.

El proceso de envío de la señal digital consta de varias etapas:

- Puesta en forma
- Aislamiento



- Circuito de mando (relé interno)
- Protección electrónica
- Tratamiento cortocircuitos

### Salidas analógicas

Los módulos de salida analógica permiten que el valor de una variable numérica interna del autómatas se convierta en tensión o intensidad.

Los rangos típicos de estas salidas están comprendidos de 0 a 10 V, -10 a 10 V, 0 a 20 mA, -20 a 20 mA y 4 a 20 mA. Conviene usar cable trenzado y apantallado.

Lo que realiza es una conversión D/A (digital/analógica), puesto que el autómatas solo trabaja con señales digitales. Esta conversión se realiza con una precisión o resolución determinada (número de bits) y cada cierto intervalo de tiempo (período de muestreo). La precisión suele ir desde los 12 a los 14 bits.

Esta tensión o intensidad puede servir de referencia de mando para actuadores que admitan mando analógico como pueden ser los variadores de velocidad, las etapas de los tiristores de los hornos, reguladores de temperatura, etc. permitiendo al autómatas realizar funciones de regulación y control de procesos continuos.

La opción de una señal de salida analógica consta de varias etapas:

- Aislamiento galvánico
- Conversión D/A
- Circuitos de amplificación y adaptación
- Protección electrónica de la salida

Las señales analógicas sufren un gran proceso de adaptación tanto en los módulos de entrada como en los módulos de salida. Las funciones de



conversión A/D y D/A que realiza son esenciales. Por ello a los módulos de E/S analógicos se les considera módulos de E/S especiales.

#### Comentarios sobre las señales de entrada y salida analógicas:

- Las entradas con polaridad (-10 a 10 V., -20 a 20 mA.) permiten la conexión de sensores para variables de magnitud de polaridad variable.
- Las entradas y salidas por tensión (0 a 10V., -10 a 10V.) tienen el inconveniente de que, si el cable es un poco largo, la caída de tensión hace que la tensión al final de los hilos sea diferente a la de origen con el correspondiente error de medida o de actuación.
- Las entradas y salidas 4 a 20 mA. presentan la ventaja de que el rango normal de medida no comienza en cero sino en 4 mA.; de esta manera se puede detectar una rotura de hilos (u otras averías).

### **Memoria**

La memoria es el elemento donde el autómata guarda todo cuanto necesita para ejecutar la tarea de control, tiene dos partes diferenciadas: la correspondiente a los datos del proceso y la que corresponde al programa. En la mayoría de los casos existe una configuración base de memoria a la que se puede añadir, hasta ciertos límites, capacidad en función de las necesidades.

#### Datos del proceso:

- Parte fija: Señales de planta, tabla de imágenes de entradas y salidas.
- Parte variable o memoria interna: en función de las necesidades de cada programa. En un autómata programable, la memoria interna es aquella que almacena el estado de las variables que maneja el autómata: entradas, salidas, contadores, relés internos, señales de estado, etc.



La clasificación de la memoria interna no se realiza atendiendo a sus características de lectura y escritura, sino por el tipo de variables que almacena y el número de bits que ocupa la variable.

Las variables contenidas en la memoria interna, pueden ser consultadas y modificadas continuamente por el programa, cualquier número de veces. Esta actualización continua de los datos obliga a construir la memoria con dispositivos RAM. Y son:

- Variables internas, de bit y de palabra.
- Datos alfanuméricos y constantes.

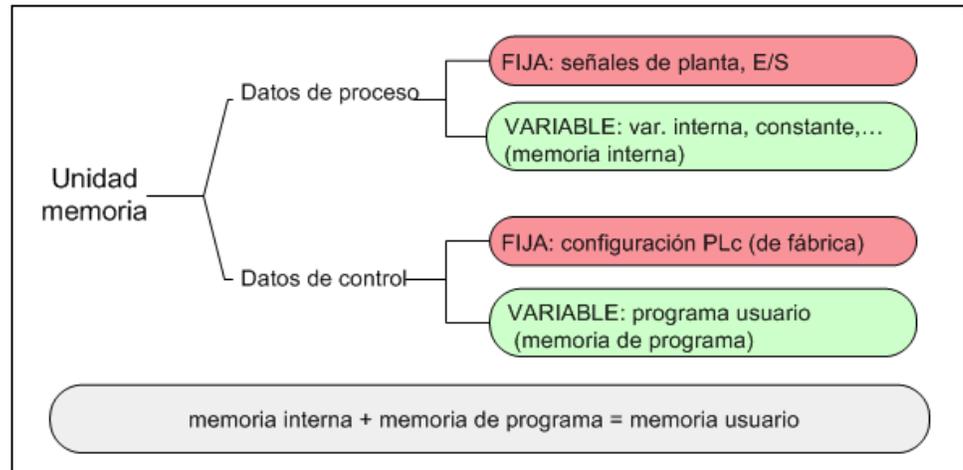
#### Datos de control:

- Parte fija: configuración del autómata (modo de funcionamiento, número de E/S conectadas). Esta parte viene programada de fábrica y es la que se encarga de la lectura de entradas /salidas.
- Parte variable o memoria de programa: instrucciones de usuario (programa). En la figura 2.1.3 se muestra la distribución de la memoria en el autómata.

Los diferentes tipos de memoria que existen según su comportamiento respecto a lectura y escritura son:

- Memorias volátiles: RAM (Random Acces Memory). Memoria de lectura y escritura. Se pueden leer, escribir y borrar fácilmente por el propio programa todas las veces que se quiera a través de los buses internos. Tienen el inconveniente que pierden la información grabada cuando se desconecta la alimentación. En muchos casos se pone una pequeña batería o un condensar para prevenir la pérdida. La memoria RAM se utiliza principalmente como memoria interna (datos de

proceso), y únicamente como memoria de programa en el caso de que pueda asegurarse el mantenimiento de los datos con una batería exterior.



**Figura 2.12 Distribución de la memoria en el autómata.**

Fuente: Boix, O., [et al.] Tecnología eléctrica. Barcelona, Ediciones Ceysa 2002, pag. 399-405

- Memorias no volátiles: Pueden ser leídas a voluntad pero difieren en la forma en que se pueden escribir. Todas ellas mantienen la información aunque se pierda la alimentación eléctrica
  - \* ROM (Read Only Memory). Memoria de solo lectura, no reprogramable. Se programan en el momento de su fabricación y se utilizan para almacenar el programa monitor del sistema.
  - \* PROM (Programmable ROM). Se pueden programar eléctricamente en cualquier momento pero no se pueden borrar ( solo se pueden programar una vez).
  - \* EPROM (Erasable ROM). Memoria de solo lectura, reprogramable con borrado por ultravioletas. Se utilizan para almacenar el programa de usuario, una vez que ha sido convenientemente depurado.



- \* EEPROM (Electrically Erasable PROM). Memoria de solo lectura, alterable por medios eléctricos. Se emplean principalmente para almacenar programas, aunque en la actualidad es cada vez más frecuente el uso de combinaciones RAM + EEPROM, utilizando estas últimas como memorias de seguridad que salvan el contenido de las RAM. Una vez reanudada la alimentación, el contenido de la EEPROM se vuelca sobre la RAM. Las soluciones de este tipo están sustituyendo a las clásicas RAM + batería puesto que presentan muchos menos problemas.
- \* EAROM (Electrically Alterable ROM). Se comporta como una memoria RAM pero los datos no se pierden al cortar la alimentación. Es la única memoria no volátil en la que se puede modificar una parte del contenido sin borrarla completamente.

### **Fuente de alimentación**

La fuente de alimentación proporciona las tensiones necesarias para el funcionamiento de los distintos circuitos del sistema.

La alimentación a la CPU puede ser de continua a 24 VCC, tensión muy frecuente en cuadros de distribución, o en alterna a 110/220 VCA. En cualquier caso es la propia CPU la que alimenta las interfaces conectadas a través del bus interno.

La fuente de alimentación del autómatas puede incorporar una batería tampón, que se utiliza para el mantenimiento de algunas posiciones internas y del programa usuario en memoria RAM, cuando falla la alimentación o se apaga el autómatas.



## Interfaces

En el control de un proceso automatizado, es imprescindible un diálogo entre operador-máquina junto con una comunicación entre la máquina y el autómatas, estas comunicaciones se establecerán por medio del conjunto de entradas y salidas del citado elemento.

De entre todos los tipos de interfaces que existen, las interfaces específicas permiten la conexión con elementos muy concretos del proceso de automatización. Se pueden distinguir entre ellas tres grupos bien diferenciados:

- Entradas/salidas especiales: se caracterizan por no influir en las variables de estado del proceso de automatización. Únicamente se encargan de adecuar las E/S, para que puedan ser inteligibles por la CPU, si son entradas, o para que puedan ser interpretadas correctamente por los actuadores (motores, cilindros, etc.), en el caso de las salidas.
- Entradas/salidas inteligentes: admiten múltiples modos de configuración, por medio de unas combinaciones binarias situadas en la misma tarjeta. De esta forma se descarga de trabajo a la unidad central, con las ventajas que conlleva.
- Procesadores periféricos inteligentes: son módulos que incluyen su propio procesador, memorias y puntos auxiliares de entrada/salida. Estos procesadores contienen en origen un programa especializado en la ejecución de una tarea concreta, a la que le basta conocer los puntos de consigna y los parámetros de aplicación para ejecutar, de forma autónoma e independiente de la CPU principal, el programa de control. Un ejemplo podrían ser los módulos de control de ejes en las que se mantiene el control de posición y, al mismo tiempo, envían consignas a un variador de velocidad.

## Unidad de programación

Es el conjunto de hardware y software mediante los cuales el programador introduce y depura las secuencias de instrucciones (en uno u otro lenguaje) que constituyen el programa a ejecutar. Normalmente se usan los ordenadores personales con el software adecuado en cada caso.

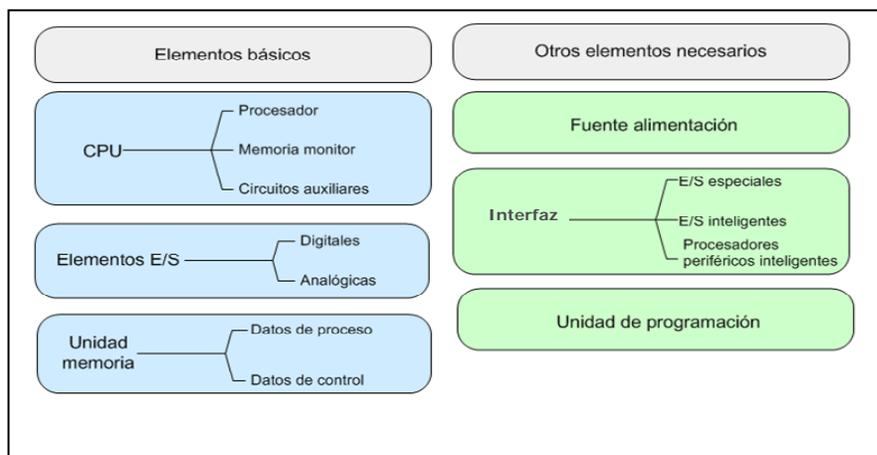


Figura 2.13 Resumen de la partes del PLC.

Fuente: [http://www.grupo-maser.com/PAG\\_Cursos/Auto/auto2/auto2/PAGINA%20PRINCIPAL/index.htm](http://www.grupo-maser.com/PAG_Cursos/Auto/auto2/auto2/PAGINA%20PRINCIPAL/index.htm)

### 2.2.6.- Funcionamiento

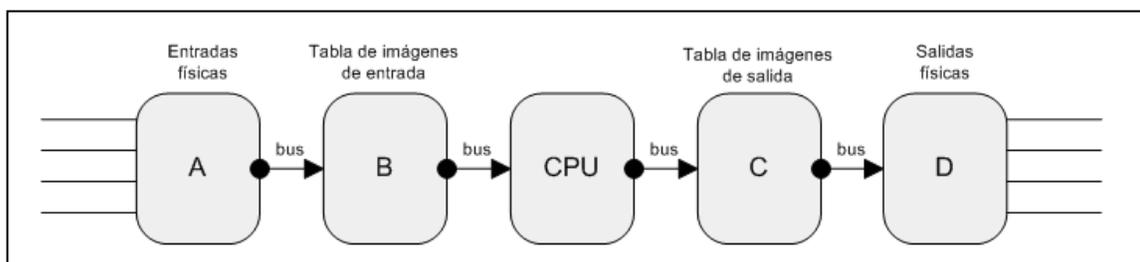
Los autómatas programables son máquinas secuenciales que ejecutan correlativamente las instrucciones indicadas en el programa de usuario almacenado en su memoria, generando unas órdenes o señales de mando a partir de las señales de entrada leídas de la planta (aplicación). Al detectarse cambios en las señales, el autómata reacciona según el programa hasta obtener las órdenes de salida necesarias. Esta secuencia se ejecuta continuamente para conseguir el control actualizado del proceso.

La secuencia básica de operación del autómata se puede dividir en tres fases:

- Lectura de señales desde la interfaz de entradas.
- Procesado del programa para obtención de las señales de control.

- Escritura de señales en la interfaz de salidas.

A fin de optimizar el tiempo, la lectura y escritura de las señales se realiza a la vez para todas las entradas y salidas. Entonces, las entradas leídas de los módulos de entrada se guardan en una memoria temporal (imagen de entradas). A ésta acude la CPU en la ejecución del programa, y según se va obteniendo las salidas, se guardan en otra memoria temporal (imagen de salidas). Una vez ejecutado el programa completo, estas imágenes de salida se transfieren todas a la vez al módulo de salida (figura 2.14).



**Figura 2.14** Secuencia básica de operación del autómata.

Fuente: [http://www.grupo-maser.com/PAG\\_Cursos/Auto/auto2/auto2/PAGINA%20PRINCIPAL/index.htm](http://www.grupo-maser.com/PAG_Cursos/Auto/auto2/auto2/PAGINA%20PRINCIPAL/index.htm)

El autómata realiza también otra serie de acciones que se van repitiendo periódicamente, definiendo un ciclo de operación. Dichas acciones se pueden observar en el diagrama de bloques de la figura 2.15.

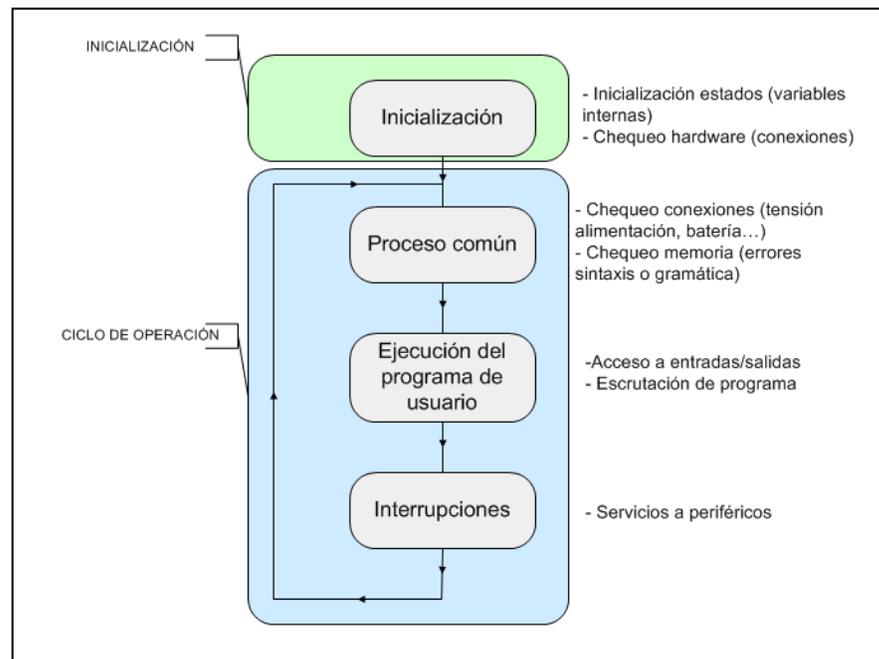
### 2.2.6.1.- Modo de funcionamiento

Existen dos modos básicos de trabajo del autómata:

- On-line: el terminal sólo puede trabajar cuando está conectado al autómata, visualizando y modificando el programa directamente en su memoria.
- Off-line: el terminal trabaja sobre el programa de la memoria del terminal y, si se desea, al acabar se copia sobre la memoria del autómata.

Es razonable pensar que tanto la programación on-line como la transferencia de un programa preparado off-line se han de hacer con el procesador en modo STOP (o PROG). Hay algunos autómatas que permiten hacerlo en modo RUN; en este caso

la modificación entra en funcionamiento al comenzar el siguiente scan de programa. Aunque sea posible modificar el programa en modo RUN, es aconsejable no hacerlo ya que cualquier error en la pulsación de una tecla puede tener consecuencias graves, al estar funcionando el autómatas.



**Figura 2.15 Ciclo de funcionamiento del autómatas.**

Fuente: [http://www.grupo-maser.com/PAG\\_Cursos/Auto/auto2/auto2/PAGINA%20PRINCIPAL/index.htm](http://www.grupo-maser.com/PAG_Cursos/Auto/auto2/auto2/PAGINA%20PRINCIPAL/index.htm)

### 2.2.6.2.- Ciclo de funcionamiento

El funcionamiento del autómatas es, salvo el proceso inicial que sigue a un Reset, de tipo secuencial y cíclico, es decir, las operaciones tienen lugar una tras otra, y se van repitiendo continuamente mientras el autómatas esté alimentado.

La figura 2.15 muestra esquemáticamente la secuencia de operaciones que ejecuta el autómatas, siendo las operaciones del ciclo de operación las que se repiten indefinidamente. El ciclo de funcionamiento se divide en dos partes llamadas Inicialización y Ciclo de Operación.



## Inicialización

Antes de entrar en el ciclo de operación el autómata realiza una serie de acciones comunes, que tratan fundamentalmente de inicializar los estados del mismo y chequear el hardware. Estas rutinas de chequeo, incluidas en el programa monitor ROM, comprueban:

- El bus de conexiones de las unidades de E/S.
- El nivel de la batería, si esta existe.
- La conexión de las memorias internas del sistema.
- El módulo de memoria exterior conectado, si existe.

Si se encontrara algún error en el chequeo, se activaría el LED de error y quedaría registrado el código del mismo.

Comprobadas las conexiones, se inicializan las variables internas:

- Se ponen a OFF (0) las posiciones de memoria interna (excepto las mantenidas o protegidas contra pérdidas de tensión).
- Se borran todas las posiciones de memoria de imagen de E/S.
- Se borran todos los contadores y temporizadores (excepto los mantenidos o protegidos contra pérdidas de tensión).

Transcurrida la Inicialización y si no han aparecido errores, el autómata entra en el Ciclo de Operación.

## Ciclo de operación

Este ciclo puede considerarse dividido en tres bloques que corresponden a:

- Proceso Común



- Ejecución del programa
- Interrupciones: servicio a periféricos

### Proceso común

En este primer bloque se realizan los chequeos cíclicos de conexiones y de memoria de programa, protegiendo el sistema contra:

- Errores de hardware (conexiones E/S, ausencia de memoria de programa, etc.).
- Errores de sintaxis (programa imposible de ejecutar).

El chequeo cíclico de conexiones comprueba los siguientes puntos:

- Niveles de tensión de alimentación.
- Estado de la batería si existe.
- Buses de conexión con las interfaces.

El chequeo de la memoria de programa comprueba la integridad de la misma y los posibles errores de sintaxis y gramática:

- Mantenimiento de los datos, comprobados en el "checksum".
- Existencia de la instrucción END de fin de programa.
- Estructura de saltos y anidamiento de bloque correctas.
- Códigos de instrucciones correctas.

### Ejecución del programa:

En este segundo bloque se consultan los estados de las entradas y de las salidas y se elaboran las órdenes de mando o de salida a partir de ellos.



### Servicio a periféricos:

Este tercer y último bloque es únicamente atendido si hay pendiente algún intercambio con el exterior. En caso de haberlo, la CPU dedica un tiempo limitado, de 1 a 2 ms., para atender el intercambio de datos. Si ese tiempo no fuera suficiente, el servicio queda interrumpido hasta el siguiente ciclo.

### **2.2.7.- Tiempo de ejecución y control en tiempo real**

El tiempo total que el autómata emplea para realizar un ciclo de operación se llama tiempo de ejecución de ciclo de operación o más sencillamente tiempo de ciclo (Scan time).

Dicho tiempo depende de:

- El número de E/S involucradas y tiempo de acceso a las mismas.
- La longitud del programa usuario (tiempo de escrutación).
- El número y tipo de periféricos conectados al autómata.
- Tipo de CPU que procesa.

Los tiempos totales de ciclos son entonces la suma de tiempos empleados en realizar las distintas operaciones del ciclo:

- Autodiagnóstico. (T1)
- Actualización de E/S. (T2)
- Ejecución de programa. (T3)
- Servicio a periféricos. (T4)

(Los tiempos de ejecución de instrucciones se miden en unidades de microsegundos, resultando un tiempo de escrutación del programa variable en función del número e



instrucciones contenidas. Precisamente el tiempo de escrutación es uno de los parámetros que caracterizan a un autómata expresado normalmente en milisegundos por cada mil instrucciones ms/k).

$$\text{Tiempo total SCAN} = T1 + T2 + T3 + T4$$

El uso de las tablas de imágenes de entradas y salidas es muy importante. Si las entradas se leyeren cada vez que se necesitan y las salidas se escribiesen también cada vez, resultaría que una misma entrada que se utiliza más de una vez dentro de un scan de programa tendría estados diferentes con lo que el automatismo no funcionaría correctamente.

El tiempo de respuesta real del PLC incluye otros factores que provocan retardos y no son constantes. Este está controlado permanentemente por el Watchdog. Son retardos introducidos por:

- El tiempo de respuesta de los sensores
- La tasa de muestreo de las entradas
- El tiempo requerido por el cómputo de las respuestas
- La tasa de muestreo de las salidas
- El tiempo de respuesta de los actuadores



## 2.3.- OLE FOR PROCESS CONTROL (OPC) [3]

### 2.3.1.- Introducción

#### 2.3.1.1.- Definición

OPC corresponde a un conjunto de especificaciones basadas en los estándares de Microsoft (COM, DCOM, OLE Automation y Active X) que cubren los requerimientos de comunicación industrial entre aplicaciones y dispositivos, especialmente en lo que se refiere a comunicaciones en tiempo real.

#### 2.3.1.2.- Historia

Tabla 2.1 Historia de OPC. Fuente: OPC Foundation

<b>1990</b>	<b><i>Prehistoria – Windows 3.0</i></b>
<b>1992</b>	2.0 OLE y WinSEM
<b>1995</b>	Los principios de OPC
<b>1996</b>	Versión 1.0 de OPC Fundación de OPC
<b>1998</b>	Un alcance más amplio para OPC
<b>2001</b>	Prueba de la conformidad

#### La prehistoria – Windows 3.0

Con la introducción de Windows 3.0 en 1990 se hizo posible OPC, sobre una plataforma barata, para ejecutar aplicaciones múltiples simultáneamente. Aún mejor, Windows proporcionó un mecanismo estándar para aquellas aplicaciones que intercambian datos en tiempo de ejecución. Este mecanismo era el Intercambio de datos Dinámico, o DDE, y no mucho antes, los usuarios vieron los beneficios de tener sus procesos o datos de planta dentro de aplicaciones de propósito general como Microsoft Excel. Pronto sin embargo, las limitaciones de DDE se hicieron



claras. No era muy robusto, no había ningún apoyo para DDE a través de una red, y, por si fuera poco, su ancho de banda era muy limitado. Se intentó rectificar estas limitaciones y se obtuvieron distintas opciones de Software produciendo el mayor impacto el Wonderware's InTouch™ SCADA, que introdujo un medio de conectar a una red el tráfico DDE (NetDDE™, que era posteriormente ocupado por Microsoft), y también aumentó el ancho de banda eficaz de DDE empaquetando objetos de datos múltiples en cada paquete o mensaje (FastDDE™). La desventaja principal de este software, y otros como AdvanceDDE™ del Software de Rockwell, consistía en que estaban sujetos a derechos privados, requiriendo pagos a sus inventores, y así nunca se lograría un verdadero estándar de industria.

### **OLE 2.0 y WinSEM**

Cuando OLE 2.0 se lanzó en 1992, era evidente que tarde o temprano reemplazaría los usos de DDE, porque era más flexible, más robusto, y usaba mecanismos más eficientes de transporte de datos.

Alrededor del mismo tiempo, un grupo que se llama WinSEM (Windows en Ciencia, Ingeniería y Fabricación) comenzó a reunirse en la oficina central Redmond de Microsoft.

Los miembros de este grupo eran en gran parte de las áreas de control industrial y adquisición de datos, con Microsoft actuando como catalizador. Por 1994, había interés firme, enfocado por WinSEM, en el uso de técnicas OLE para mover datos de proceso entre aplicaciones en (casi) tiempo real. En particular, un número de vendedores SCADA vio la posibilidad de estandarizar la interfaz entre el núcleo SCADA y los drivers de dispositivo que eran en realidad responsables de adquirir los datos.

Potencialmente, esto podía beneficiar a vendedores SCADA y fabricantes de equipo: el vendedor SCADA no tendría que invertir en la escritura de drivers,



mientras el fabricante de equipo tiene que proporcionar sólo un driver que trabajase con todo el software de Windows. La oferta más interesante fue propuesta por US Data en marzo de 1995. Comparado con la especificación OPC, este documento ahora parece muy simple. Sin embargo, se establecen la mayor parte de los conceptos clave de OPC.

Después de la publicación de este prometedor documento, progresó hacia un estándar muy lento. Esto era el punto de vista de los implicados en WinSEM (incluyendo Microsoft). Esto supone el origen de OPC Task Force.

### **Los Principios de OPC**

OPC Task Force publica en 1995 en el transcurso del ISA Show desarrollado en Nueva Orleans con una nueva versión. Sus miembros eran Fisher-Rosemount (ahora Emerson Process Management), Intellution, Intuitive Technology, OPTO 22, y Rockwell Software. Microsoft actúa como apoyo.

La primera versión preliminar de la especificación OPC fue lanzada en diciembre de 1995, y era presentado a WinSEM en Redmond en enero de 1996. A pesar del resentimiento de que un grupo de élite asumiera el esfuerzo de estandarización, la respuesta era favorable y muy constructiva.

Una segunda especificación se publica en marzo 1996, y se realizan seminarios 'JumpStart' en Dallas, Texas (abril 1996), Londres, Inglaterra (julio 1996), y Japón (agosto 1996) para ofrecer una introducción a los interesados en el estándar propuesto. En cada ocasión la respuesta era positiva. Las diapositivas de Powerpoint usadas en el primer OPC JumpStart están todavía disponibles en [ftp: // zilker.net/pub/opc/JumpStart](ftp://zilker.net/pub/opc/JumpStart).



## **OPC la Versión 1.0**

La versión 1.0 de la especificación OPC fue lanzada el 29 de agosto 1996. En 1997 aparece una versión 1.0A en la cual se corrige la Especificación de Acceso de Datos OPC.

## **Fundación OPC**

Buscando la opinión en la industria, se tomó la decisión de que la especificación OPC debía ser manejada por una organización independiente, no lucrativa llamada la Fundación OPC. La Fundación OPC se presentó por primera vez en el ISA Show en 1996 realizado en Chicago, con demostraciones de servidor OPC de varias empresas en la cabina Microsoft, y realizando la primera reunión de Asamblea general de miembros.

Comenzaron a aparecer a finales de 1996 productos comerciales que usan OPC. A mediados de 1998, se confirma OPC como el estándar de industria. La Fundación tiene el apoyo de la industria (mas de 150 miembros en total), sin incluir industrias de Japón y Europa. La fundación tiene su propia dirección web, <http://www.opcfoundation.org>.

## **Un Ámbito Más amplio para OPC**

OPC Data Access 2.0 se publica a finales de 1998 y contiene documentos de especificación separados para clientes e interfaces de automatización. La especificación de cliente incluyó un mecanismo más robusto. Las interfaces de automatización, mientras tanto, fueron rediseñados para mejorar su uso, y aprovechar mejor la tecnología. También a finales de 1998, el alcance de OPC fue ampliado con la publicación de las Alarmas y la especificación de interfaz de cliente de Acontecimientos, bajo la versión 1.0. En el año 2000 se publican nuevas



especificaciones de interfaz de cliente que cubren el Acceso de Datos Históricos, Batch y la Seguridad.

### **Prueba De Conformidad**

Posiblemente el desarrollo más significativo de 2001 era la versión OPC Foundation's Compliance Testing y el programa de certificación para servidores de Acceso de Datos OPC, que pronto serían ampliados para cubrir Servidores de acontecimientos y Alarmas. En el 2001 también se publicó la especificación de interfaz de automatización para el Acceso de Datos Histórico, y la versión 2.0 de especificación de interfaz Batch custom. La especificación de OPC es una especificación técnica no-propietaria que define un sistema de interfaces estándares basados sobre la tecnología de OLE/COM.

#### **2.3.1.3.- Aplicación de OPC**

OPC es un mecanismo uniforme para comunicar a numerosas fuentes de datos, a dispositivos en el nivel planta de la fábrica, o a una base de datos en un cuarto de control. La arquitectura de la información para la Industria del Proceso mostrada en la Figura 2.16, implica los niveles siguientes:

- Administración de Campo. Con la llegada de dispositivos inteligentes, se puede proporcionar una riqueza de información sobre dispositivos de campo que no estaban previamente disponibles. Esta información proporciona los datos de un dispositivo, sus parámetros de configuración, las materias de la construcción, etc. Toda esta información debe ser presentada al usuario, y a cualquier aplicación que la use.
- Administración de Procesos. La instalación de Sistemas Distribuidos de Control (DCS) y sistemas SCADA para supervisar y controlar datos del

proceso de fabricación disponibles electrónicamente que habían sido recopilados manualmente.

- Administración del negocio. Los beneficios pueden ser obtenidos instalando sistemas del control. Esto se consigue integrando la información recogida del proceso en los sistemas de negocio que maneja aspectos financieros de la fabricación. Proporcionar esta información eficazmente a aplicaciones de cliente aminora el esfuerzo requerido para proporcionar esta integración.

Para conseguir esto, los fabricantes necesitan conseguir acceso a los datos del nivel planta de la fábrica e integrarlos en su sistema de negocio. Los fabricantes deben ser capaces de utilizar las herramientas (Paquetes de SCADA, las Bases de datos, etc.) para satisfacer sus necesidades. La clave es una arquitectura abierta y eficaz de comunicación en el acceso de datos, y no los tipos de datos.

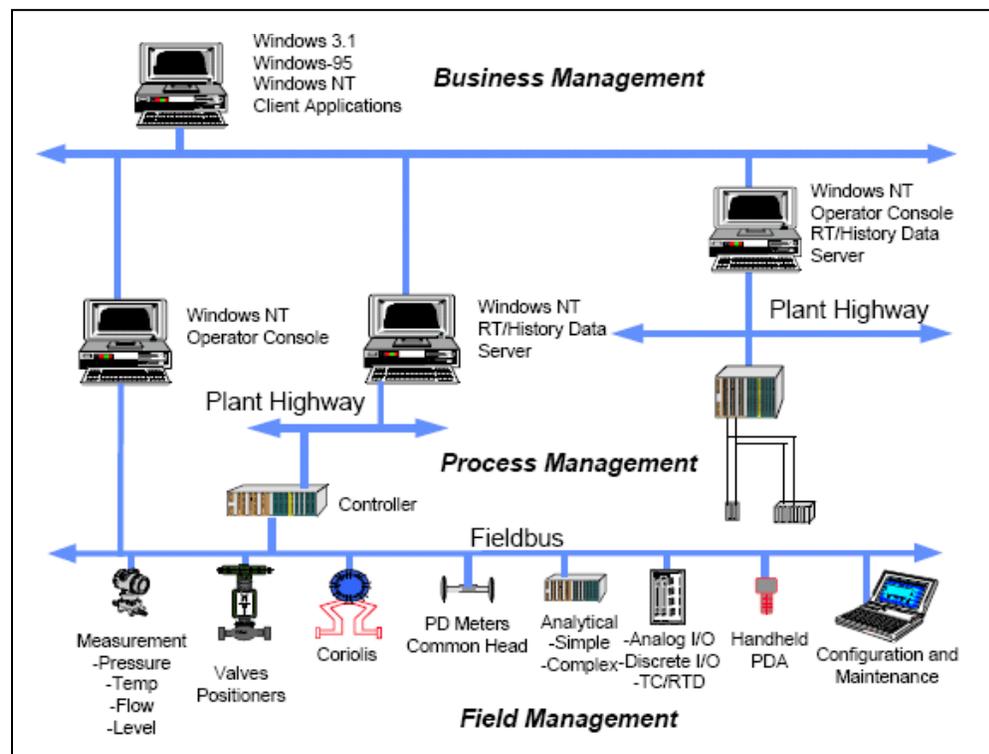


Figura 2.16 Esquema general. Fuente: OPC Foundation

### 2.3.1.4.- Arquitectura OPC

Lo que se necesita para las aplicaciones es una manera común de acceder a los datos de cualquier fuente, como un dispositivo o una base de datos. El servidor OPC en esta figura y en secciones posteriores es sinónimo de cualquier servidor que proporciona la interfaz OPC.

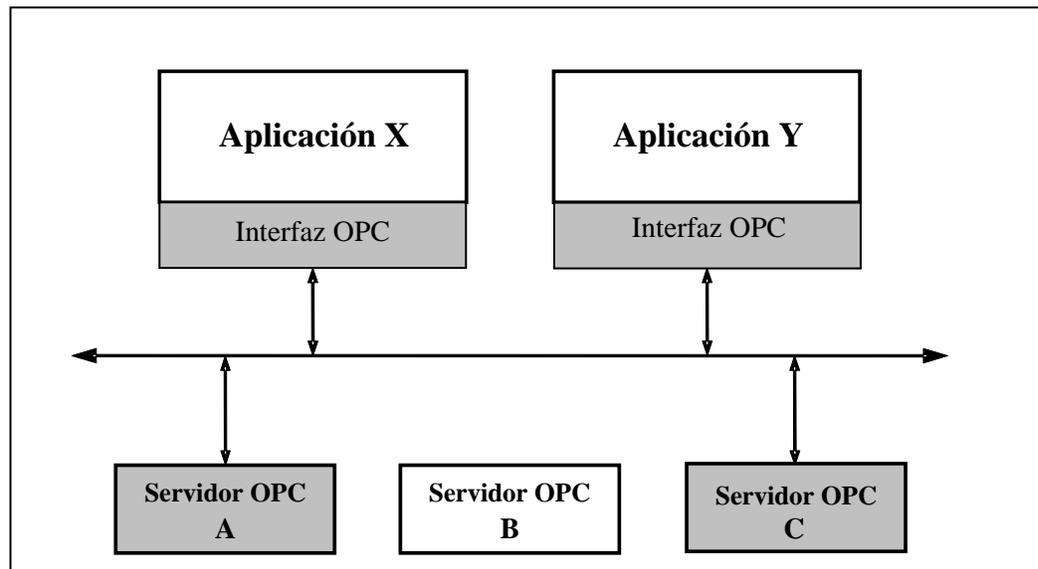


Figura 2.17 Arquitectura OPC. Fuente: OPC Foundation

### 2.3.1.5.- Adaptación a la Aplicación del Cliente

Hay muchas aplicaciones de cliente que requieren datos de una fuente, y acceden a esos datos desarrollando Drivers para sus propios paquetes. Esto ocasiona los siguientes inconvenientes:

- Para cada fabricante de hardware se necesita un driver particular.
- Las características del hardware no son soportadas por todos los drivers.



- Un cambio en las características del hardware afecta a algunos drivers
- Generalmente, dos paquetes de hardware no pueden acceder simultáneamente al mismo dispositivo, ya que cada uno tiene drivers independientes.

Los fabricantes del hardware procuran resolver estos problemas desarrollando drivers, pero son entorpecidos por diferencias en los protocolos de cada cliente. No han logrado todavía desarrollar un driver eficiente que pueda ser usado por todos los clientes. OLE para el Control de Proceso (OPC) establece una línea entre proveedores de hardware y software. Lo cual Facilita un mecanismo para proporcionar los datos de una fuente y comunicar esos datos a cualquier aplicación del cliente. Un fabricante puede desarrollar un servidor optimizado para comunicarse con la fuente de datos, y mantener el mecanismo de acceso a los datos de la fuente o dispositivo. Incorporando al servidor una interfaz OPC, que permite a cualquier cliente OPC el acceso a sus dispositivos.

#### **2.3.1.6.- Métodos de Aplicación**

Un número creciente de aplicaciones se desarrolla en ambientes como Visual Basic (VB), Delphi, Power Builder, etc. OPC debe tener en cuenta esta tendencia. Así, Microsoft diseña OLE/COM para permitir que componentes escritos en C y C ++, sean utilizados por programa de cliente (escritos en VB o Delphi para un dominio totalmente diferente). Los componentes software se escribirán en C y C ++ para encapsular la complejidad de acceso a datos de un dispositivo, de forma que permita a los promotores de aplicación de gestión escribir en VB y tener acceso a los datos de piso de planta. El objetivo de todas las especificaciones es el desarrollo de servidores OPC en C y C++, y así, facilitar el desarrollo de aplicaciones de cliente de OPC en el lenguaje escogido. La arquitectura y el diseño de las interfaces son pensados para apoyar el desarrollo de servidores OPC también en otros lenguajes.



### 2.3.1.7.- Esquema General

Como se muestra en la figura 2.18, OLE para el Control de Proceso (OPC™) está diseñado para permitir a las aplicaciones de cliente el acceso a los datos de planta de una manera consistente. OPC proporcionará muchos beneficios:

- Los fabricantes de hardware sólo tienen que hacer un conjunto de componentes de programa para que los clientes los utilicen en sus aplicaciones.
- Los fabricantes de software no tienen que adaptar los drivers ante cambios de hardware.
- Con OPC, la integración de sistemas en un entorno heterogéneo será simple.

### 2.3.1.8.- Funcionalidad OPC

El objetivo principal de OPC es entregar las especificaciones a la industria tan rápido como sea posible. Con esto en la mente, el alcance de las primeras versiones del documento se limita a áreas comunes de todos los vendedores. La funcionalidad adicional se definirá en versiones futuras. Por lo tanto, las primeras versiones se enfocan en:

- Acceso de Datos en línea, en otras palabras, la lectura y escritura eficiente de datos entre una aplicación y un dispositivo de control de proceso
- El Manejo de Alarma y el Acontecimiento, es decir, los mecanismos para que Clientes de OPC sean notificados de las condiciones especificadas y alarma.
- Acceso a Datos Históricos, se refiere a, la lectura, procesamiento y redacción de datos históricos

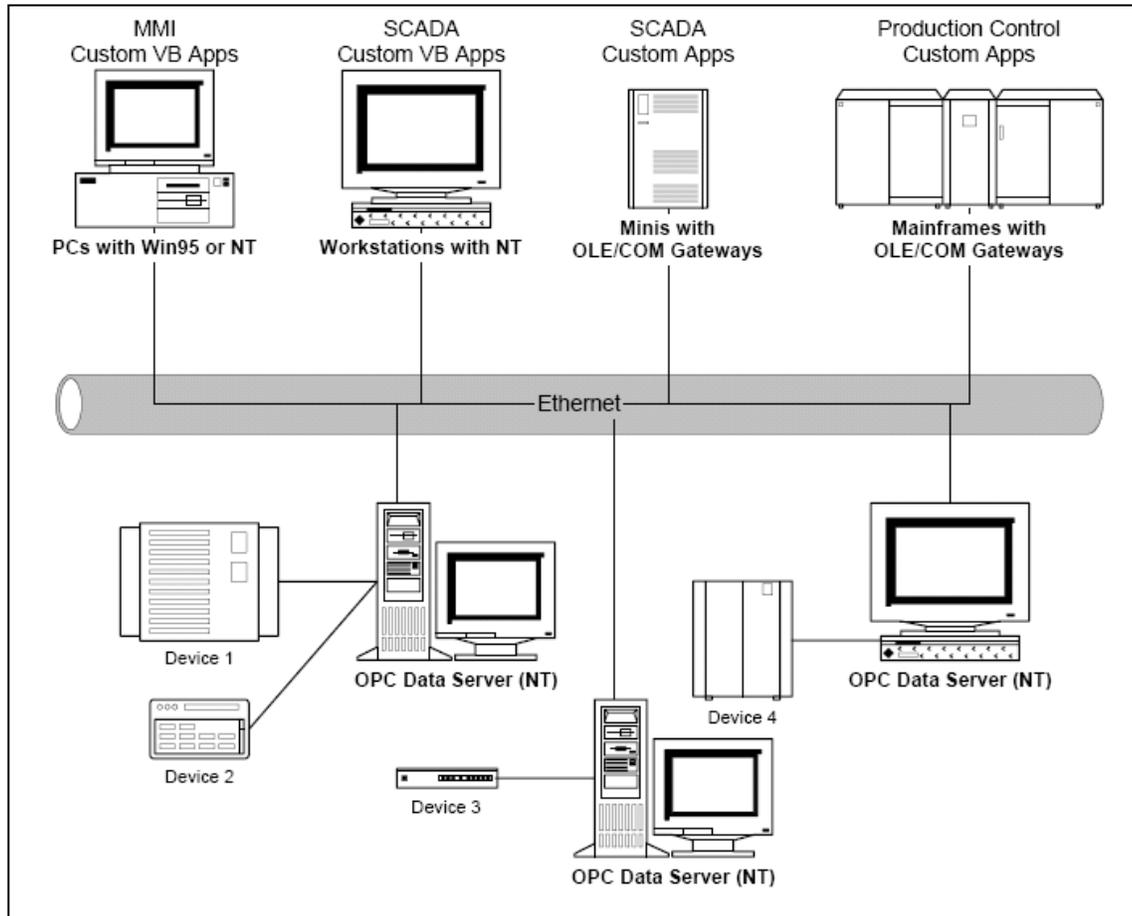


Figura 2.18 Esquema de funcionalidad de OPC. Fuente: OPC Foundation

Las funcionalidades tales como la seguridad, datos históricos y acontecimientos son características que pertenecen a versiones siguientes.

Otras metas para el diseño de OPC son:

- Sencillo de aplicar.
- Flexibilidad para acomodar múltiples necesidades
- Proporcionar un alto nivel de funcionalidad
- Ser eficiente.

Las especificaciones incluyen lo siguiente:



- Un conjunto de interfaces COM para el uso de cliente y servidor.
- Referencias a un conjunto de interfaces de Automatización OLE para apoyar a clientes desarrollados con las aplicaciones de gestión de niveles más altos como Excel, Visual Basic, etc.

Microsoft distribuyó la tecnología OLE (DCOM) para facilitar a clientes la comunicación con servidores remotos.

### 2.3.2.- Bases de OPC

#### 2.3.2.1.- Objetos e interfaces de OPC

OPC está basado en la tecnología OLE/COM de Microsoft.

Esta especificación describe los objetos de OPC COM y sus interfaces implementadas por servidores OPC. Un cliente OPC se puede conectar a servidores OPC proporcionados por más de un fabricante.

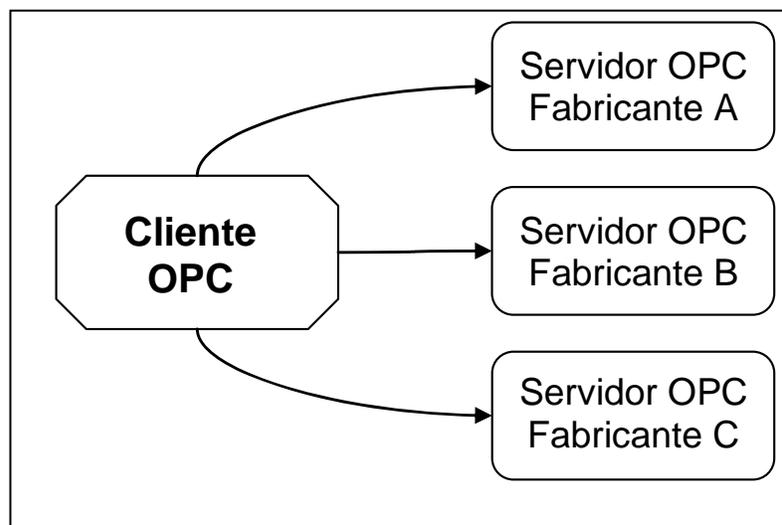


Figura 2.19 Cliente OPC. Fuente: OPC Foundation

Los servidores OPC pueden ser proporcionados por diferentes fabricantes. El código que da el fabricante determina los equipos y los datos a los que el servidor va a tener acceso y también determina los detalles de cómo el servidor accede físicamente a los datos.

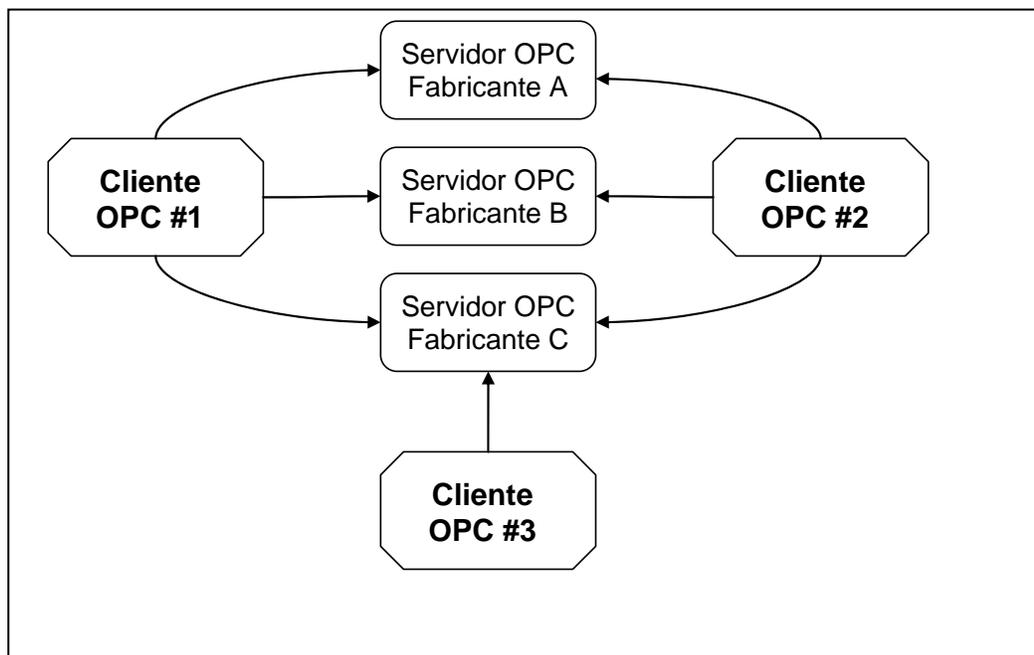


Figura 2.20 Relación Cliente/Servidor OPC. Fuente: OPC Foundation

### 2.3.2.2.- Perspectiva General del Acceso de Datos OPC

A un nivel alto, un Servidor de Datos OPC está compuesto por varios elementos: el servidor, el grupo y el ítem. El servidor OPC mantiene información sobre el servidor y sirve como almacén para objetos del grupo OPC. El objeto del grupo OPC mantiene información sobre sí mismo y provee de mecanismos para contener y organizar lógicamente ítems de OPC. Los grupos OPC proveen un modo para que los clientes organicen los datos. Por ejemplo, el grupo puede representar los ítems en un display del operador o informe especial. También se pueden crear conexiones basadas en excepciones entre el cliente y los ítems en el grupo y se pueden habilitar o deshabilitar

según sean necesarias. Un cliente OPC puede configurar la velocidad a la que un servidor OPC debería de proveer los cambios de datos al cliente OPC.

Hay dos tipos de grupos, los públicos y los locales (o privados). Los públicos son para compartir entre muchos clientes y los locales son privados para cada cliente. Dentro de cada grupo el cliente puede definir más de un ítem OPC.

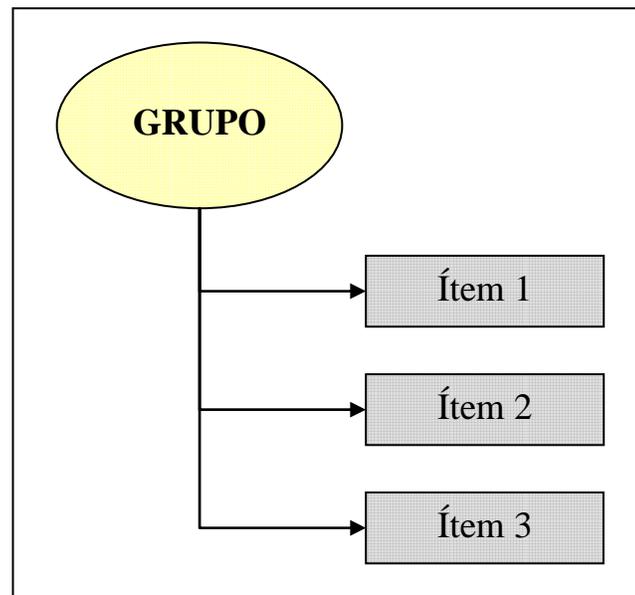


Figura 2.21 Relación grupo/ítem. Fuente: OPC Foundation

Los Items OPC representan conexiones a las fuentes de datos dentro de un servidor. Desde el punto de vista de una interfaz, un Item OPC, no es accesible como un objeto por parte de un Cliente OPC. Por lo tanto, no hay ninguna interfaz externa definida para un Item OPC. Todo los accesos a Items OPC se hacen a través de un objeto del Grupo OPC que contiene al Item OPC. A cada Item se le asocian sellos de Valor, Calidad y Tiempo. El Valor se expresa en la forma VARIANT y la Calidad es parecido a lo especificado por Fieldbus. Es importante entender que los items no son fuentes de datos, son únicamente conexiones a ellas. Por ejemplo, los tags en un sistema DCS existen independientemente si en ese momento hay un Cliente OPC accediendo a ellos.



### 2.3.2.3.- Perspectiva General de la Gestión de Alarmas y Eventos OPC

Estas interfaces ofrecen mecanismos para que los Clientes OPC reciban notificaciones de ciertos eventos especificados y de condiciones de alarma. También ofrecen servicios que permiten a Clientes OPC determinar los eventos y condiciones soportados por el Servidor OPC y obtener su estado.

Se hace uso de las entidades denominadas como alarmas y eventos. En lenguaje informal estos dos conceptos se usan de forma indiferente. En OPC, alarma es una condición anormal y por lo tanto es un caso especial de condición. Una condición es un estado concreto del Servidor de Eventos OPC o de uno de los objetos contenidos por dicho servidor, que puede resultar de interés para sus clientes. Por ejemplo, el tag FC101 puede tener las siguientes condiciones asociadas a él: HighAlarm, HighHighAlarm, Normal, LowAlarm, y LowLowAlarm.

Por otro lado, un evento es un suceso detectable que es significativo para un Servidor OPC, para el equipo al que representa y para sus Clientes OPC. El evento puede estar o no asociado a una condición. Por ejemplo, las transiciones a las condiciones HighAlarm y Normal son eventos asociados a condiciones. Pero sin embargo, las acciones del operador, cambios de la configuración y errores del sistema son ejemplos de eventos que no están asociados a ninguna condición específica. Los Clientes OPC pueden solicitar la notificación de ciertos eventos específicos.

La interfaz del Servidor de Eventos IOPC ofrece métodos para permitir a un Cliente OPC:

- Determinar los tipos de eventos que puede soportar un Servidor OPC.
- Suscribirse a algunos eventos, de tal forma que los Clientes OPC puedan recibir notificaciones sobre ellos. Los filtros se pueden usar para definir un subconjunto de eventos deseados.



- Acceder y manipular condiciones implementados por un Servidor OPC.

Además de la interfaz IOPC un Servidor de eventos puede soportar otro tipo de interfaces para mostrar las condiciones implementadas por un servidor.

#### **2.3.2.4.- Perspectiva General del Acceso de Datos Históricos OPC**

Los motores históricos producen una información que ha de ser distribuida a usuarios y a clientes software que estén interesados en dicha información. Normalmente la mayoría de los sistemas históricos utilizan una interfaz propia para diseminar los datos. No hay capacidad para utilizar las soluciones históricas que ya existen con otras capacidades en un entorno plug-n-play. Esto hace que sea necesario que el desarrollador cree otra vez su infraestructura para que sea la misma para todos sus productos.

Hay distintos tipos de servidores Históricos. Algunos importantes que son soportados por esta especificación son los siguientes:

- Servidores de datos simples. Estos servidores ofrecen la capacidad de almacenar datos simples.
- Servidores de análisis y compresión de datos complejos. Estos servidores ofrecen capacidad de compresión de datos además de la capacidad de almacenaje de datos. Son capaces de ofrecer funciones de análisis de datos, tales como cálculo de la media, del mínimo, del máximo... Pueden actualizar datos y tener un resumen de actualizaciones.

### 2.3.2.5.- Integración OPC

A pesar de que OPC está diseñado principalmente para acceder a datos de un servidor en red, las interfaces OPC se puede usar para distintas aplicaciones. En el nivel más bajo puede recoger datos de aparatos físicos y llevarlo a SCADA o DCS, o de un servidor SCADA o DCS a una aplicación. La arquitectura y el diseño permiten que se pueda configurar un Servidor OPC que permita a una aplicación cliente acceder a datos de distintos Servidores OPC.

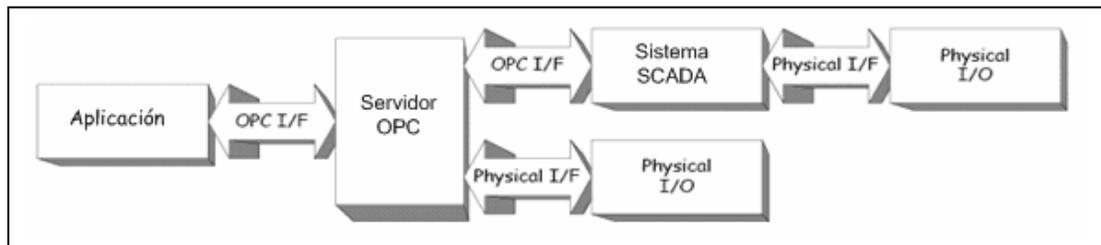


Figura 2.22 Relación Cliente/Servidor OPC. Fuente: OPC Foundation

### 2.3.2.6.- Arquitectura OPC general y componentes

Las especificaciones OPC contienen siempre dos tipos de interfaces; Interfaces Custom e interfaces de Automatización. Esto se puede ver en la figura 2.23.

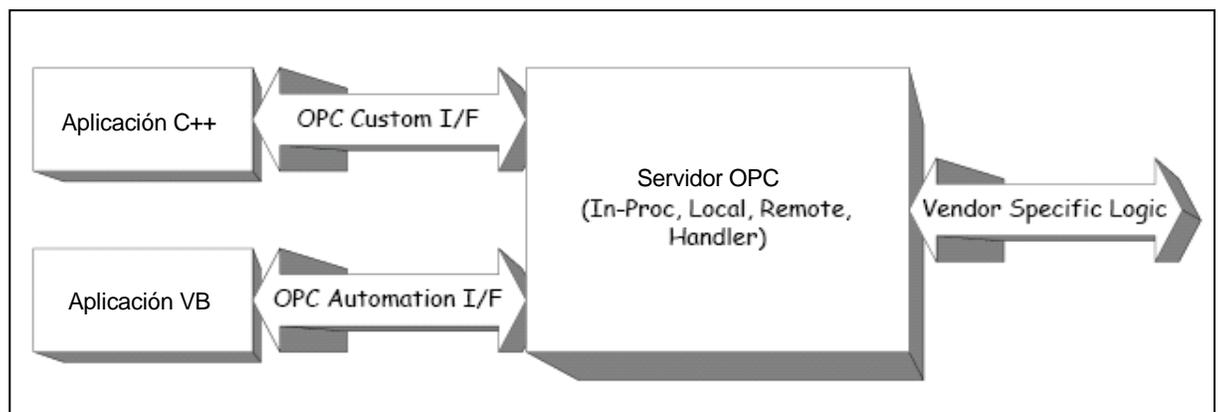


Figure 2.23 Interfaces OPC. Fuente: OPC Foundation

Hay varias consideraciones cuando se implementa un servidor OPC. La más importante es la frecuencia de transferencia de datos de rutas (paths) de comunicación que no se pueden compartir a aparatos físicos. Por lo tanto se espera que el Servidor OPC sea un EXE local o remoto que incluya código que se encargue de recoger datos de aparatos físicos o de bases de datos.

Una aplicación Cliente OPC se comunica con un Servidor OPC mediante las interfaces Custom y Automation previamente mencionadas. Los servidores OPC han de implementar la interfaz de Custom y opcionalmente pueden implementar la de Automation.

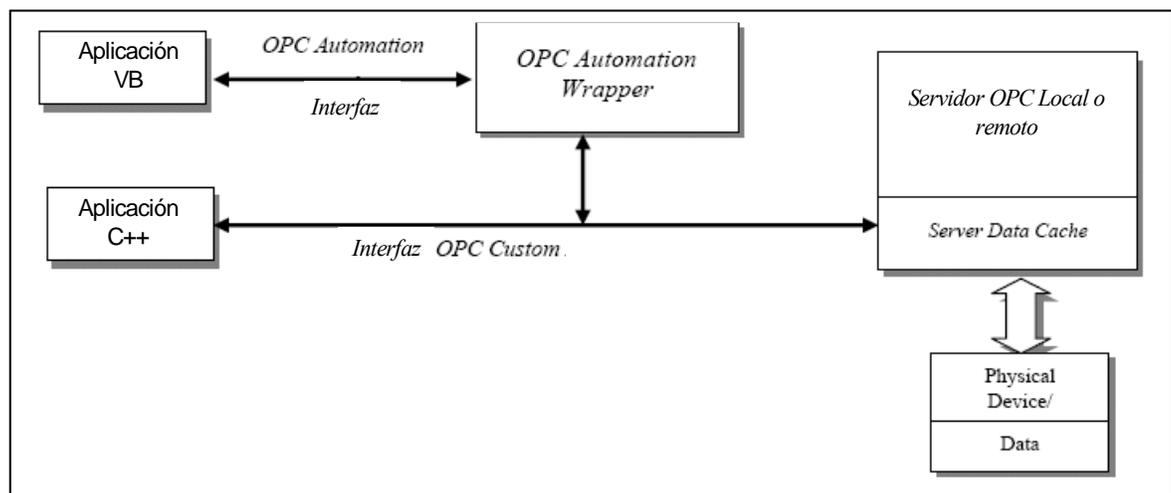


Figura 2.24 Arquitectura típica de OPC. Fuente: OPC Foundation

### 2.3.2.7.- Servidores remotos vs. locales

A la hora de montar las redes OPC hay dos alternativas:

- Los clientes deben conectarse siempre a un servidor local que hará uso de un esquema de red existente.



- El cliente se puede conectar al servidor que desee.

### **2.3.3.- Especificaciones OPC**

El estándar OPC está dividido en diferentes especificaciones. Todas ellas tienen como base común la especificación OPC Common y dependiendo de la aplicación para la que estén definidas las especificaciones, implementarán unas interfaces u otras aparte de las implementadas en OPC Common. Las especificaciones que se pueden encontrar dentro del estándar OPC son:

- OPC Common
- OPC Data Access
- OPC Alarm & Events
- OPC Historical Data Access
- OPC Batch
- OPC DX
- OPC XML

A parte de estas especificaciones se siguen preparando nuevas especificaciones.

#### **2.3.3.1.- OPC Common**

Como su nombre indica es la especificación común a todos los servidores y clientes que implementen OPC en cualquiera de las especificaciones arriba mencionadas. Esta especificación define tres funciones que deben ser implementadas siempre:

- Desconexión de los clientes a petición de los servidores. Esta función es implementada a través de la interfaz IOPCShutdown.



- Registro de servidores dentro del sistema.
- Capacidad de dar a conocer a las aplicaciones clientes los servidores instalados en una máquina. Esto se lleva a cabo gracias a la interfaz IOPCServerList.

### **Tipos de Interfaces**

La especificación OPC hace una clasificación de las interfaces atendiendo a varios criterios:

- El lenguaje de programación utilizado a la hora de diseñar los clientes y servidores.
- La necesidad de su implementación en los servidores como en los clientes.

Según el lenguaje de programación se pueden distinguir dos tipos de interfaz:

- Interfaz Automation
- Interfaz Custom

Una aplicación cliente OPC se comunica con un servidor OPC a través de las interfaces Custom y Automation. Los programas clientes que están creados con un lenguaje script, como por ejemplo Visual Basic utilizarán la interfaz “Automation”. En cambio aquellos que están creados en C++ utilizarán de forma más sencilla la interfaz “Custom”. Para establecer una comunicación entre un servidor OPC y un cliente OPC que están programados en diferentes lenguajes se puede usar “OPC Automation Wrapper”. El “*OPC Automation Wrapper*” representa un enlace de conexión entre por ejemplo un servidor OPC programado en C/C++ y una aplicación programada en Visual Basic.

Los servidores OPC tienen que implementar la interfaz “Custom”, y opcionalmente pueden implementar la interfaz “Automation”. Atendiendo a la



necesidad o no de su implementación se pueden distinguir también dos tipos de interfaces:

- Interfaces obligatorias: Las interfaces obligatorias como su nombre indica tienen que estar siempre implementadas en el diseño tanto de un cliente como de un servidor OPC. Las interfaces obligatorias en un servidor tienen que tener todas las funcionalidades que están definidas dentro de las interfaces implementadas. Por otra parte un cliente se comunicará con un servidor llamando a las funciones de la interfaz obligatoria.
- Interfaces opcionales: Se trata de las interfaces que pueden estar implementadas o no en los servidores. Cuando un servidor soporta una interfaz opcional es muy importante que este implemente todas las funciones que estén definidas dentro de esa interfaz opcional, incluso si la función devuelve como resultado un error de función no implementada (E\_NOTIMPL). Un cliente que desee utilizar las funcionalidades de una interfaz opcional debe solicitar al servidor sus interfaces opcionales. El cliente debe estar desarrollado de manera que no utilice las interfaces opcionales que no existan.

### **Definición de Interfaces**

Dentro de la especificación OPC Common están definidas las siguientes interfaces:

#### IOPCShutdown

Como ya se comentó anteriormente, los servidores OPC poseen la capacidad de pedir la desconexión de los clientes. Esta capacidad la poseen todo tipo de servidores OPC. Esta funcionalidad está disponible a través de un punto de conexión en el servidor y en el cliente a través de la interfaz IOPCShutdown.



### IConnectionPointContainer

Esta interfaz proporciona acceso al punto de conexión para la interfaz IOPShutdown.

### IOPCCCommon

Esta interfaz es usada por todos los servidores OPC independientemente de que pertenezcan a una especificación u otra. Proporciona la capacidad de establecer y demandar identificadores locales (LocaleID) que serán específicos para cada sesión cliente/servidor. Un cliente que mantenga conexiones con un servidor OPC Data Access y un servidor OPC Alarms and Events usará esta interfaz de forma independiente con cada servidor.

### IOPCServerList

Esta interfaz puede trabajar de forma remota. Determina el tipo de servidores que están disponibles en una máquina.

## **2.3.3.2.- OPC Data Access (OPC DA)**

En un servidor OPC Data Acces se pueden diferenciar tres tipos de objetos:

- Objeto OPC Server.
- Objeto OPC Group
- Objeto OPC Item

El objeto OPC Server es el objeto COM al cual se conectan las aplicaciones. Debajo de él se encuentran una colección de objetos OPC Group. Estos objetos son creados por la aplicación de forma dinámica para mantener listas de tags y atributos que en OPC son llamados ítems. Por ejemplo una interfaz HMI debería crear un grupo por cada imagen que tenga abierta. El contenido de los ítems y grupos puede variar en el tiempo dependiendo de las necesidades de las aplicaciones.

El servidor OPC Data Access tiene dos formas de estructurar los objetos Group e Item:

- Forma jerárquica
- Forma plana

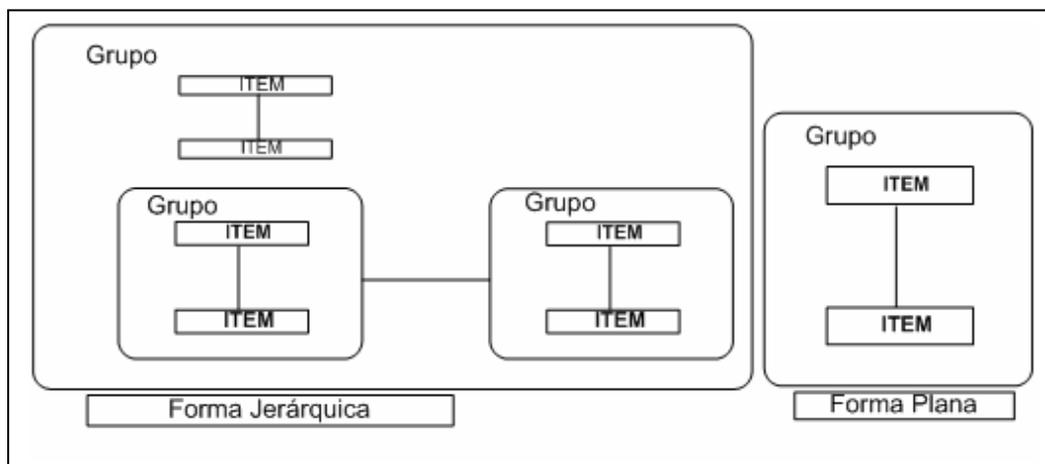


Figura 2.25: Estructura de los objetos Group e Item en el servidor. Fuente: OPC Foundation

### El Modelo Lógico de OPC DA

En la figura 2.26 se muestra un esquema en la que se representa el modelo lógico de los objetos donde se puede apreciar cada objeto y la relación que mantienen entre ellos

#### Objeto OPC Server

El servidor OPC DA establece un objeto del tipo OPC Server para cada cliente y crea un canal de comunicación para cliente por separado. De esta forma se evita que el flujo de información entre cliente y servidor se vea disminuido a causa de otros clientes. El objeto OPC Server mantiene información del actual servidor y se usa como depósito de objetos OPC Group.

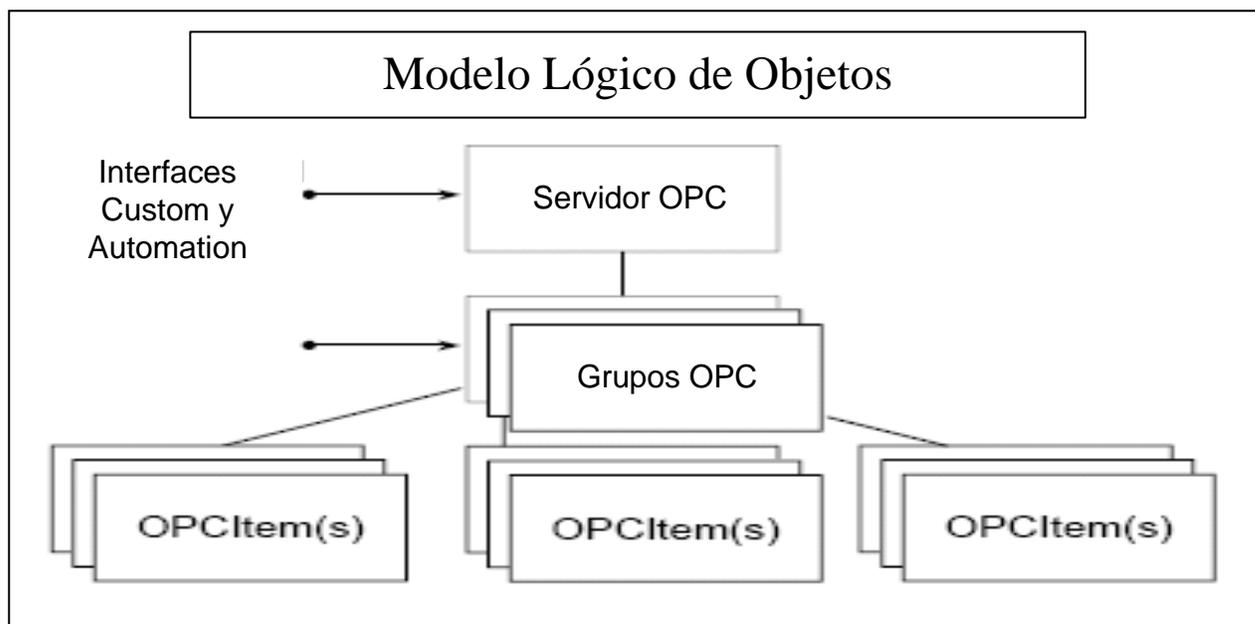


Figura 2.26: Modelo de objetos lógico. Fuente: OPC Foundation

### Objeto OPC Group

El objeto OPC Group tiene como finalidad el proporcionar a las aplicaciones un mecanismo para organizar los datos que necesitan. Diferentes grupos pueden ser usados por diferentes partes de la aplicación. El objeto OPC Group además de mantener información sobre sí mismo se encarga del mantenimiento y la organización lógica de los objetos OPC Item. La transmisión de datos del servidor OPC DA actúa a nivel del OPC Group. Cada Grupo tiene que tener un nombre único relacionado con el cliente OPC. El cliente OPC puede cambiar más tarde este nombre, pudiendo especificar incluso el activo o inactivo del OPC Group. El servidor OPC DA es el encargado de generar grupos OPC. Además de crear grupos es posible suprimir un grupo, obtener el nombre de un grupo y enumerar los grupos. Los grupos contienen items, que corresponden a datos en el servidor. Si el servidor es un supervisor, los items se corresponden con tags. Si el servidor es un autómatas los items se corresponden con registros (DB) Los servidores OPC pueden leer sus datos de bases de datos, mapas ADD-IN, o desde cualquier fuente de datos.



### OPC Item

El objeto OPC Item representa un punto de conexión entre el servidor y el valor real en el dispositivo físico. Esto significa que los Items OPC no representan las fuentes actuales de datos pero solo contienen la dirección al tag apropiado en la configuración del servidor. Se trata de un objeto transitorio que existe junto con el servidor OPC. Al contrario que el OPC Group y el OPC Server, el Item OPC no soporta interfaces OPC y por eso no es un objeto COM. Es un objeto interno del servidor OPC que mantiene información importante de las necesidades solicitadas por el cliente OPC (por ejemplo los datos usados para actualizar valores, el estado activo o inactivo de los valores requeridos, etc.)

Desde la perspectiva del cliente OPC, el Item OPC no representa la fuente de datos actual sino únicamente la conexión lógica a la fuente de datos. Usando el identificador del item (Item ID), el OPC Item está asociado unívocamente al item definido en la configuración física del servidor OPC.

Los Items se identifican por su nombre. El servidor debe utilizar una técnica para controlar la correspondencia de los nombres de los Items y los datos fuente que él maneja. Estos nombres son accesibles a través de la interfaz “Browse”. Los items se encuentran dentro de los objetos OPC Group y se caracterizan por atributos y propiedades entre las cuales se puede destacar:

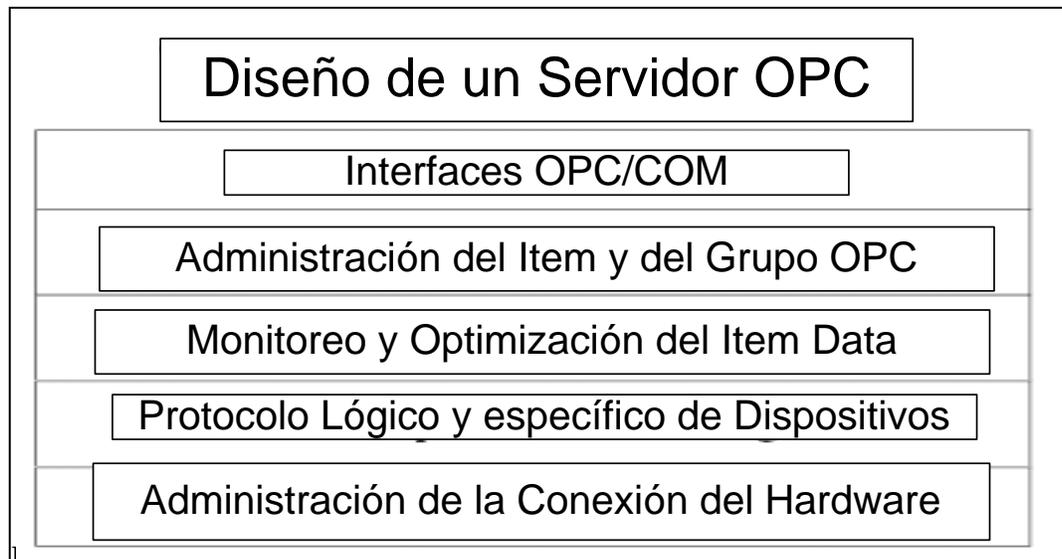
- Nombre
- Valor
- Calidad
- Marca de tiempo

### ItemID

ItemID es un único identificador del tag y es usado por el cliente OPC para establecer la conexión con el servidor OPC.

### **Diseño de un Servidor OPC DA**

En su nivel más bajo, el objeto OPC Server es un driver I/O que puede transmitir datos desde un dispositivo físico conectado, normalmente un PLC. Otra parte integral del OPC Server es un bloque que optimiza la recolección de datos y mantiene la más alta actuación en la comunicación contra el PLC. Usando las interfaces apropiadas de la especificación OPC Data Acces, los datos transmitidos son ofrecidos a los clientes OPC conectados.



**Figura 2.27** Diseño de un servidor OPC DA. Fuente: OPC Foundation

En algunos casos los servidores OPC no están diseñados para la comunicación con dispositivos específicos. Estos servidores OPC no incluyen algunas capas las cuales se sustituyen en este caso por un protocolo de comunicación con la interfaz.



## Definición de Interfaces OPC DA

### Interfaces del objeto OPC Server

En este objeto están definidas las siguientes interfaces:

- *IOPCServer*: Esta interfaz permite al cliente manejar los grupos.
- *IOPCBrowseServerAddressSpace* (*opcional*): Esta interfaz permite que la aplicación explore los nombres de los tags y atributos disponibles en el servidor.

Estas interfaces permiten a las aplicaciones:

- Crear y borrar grupos
- Explorar los tags disponibles
- Traducir los códigos de error en texto inteligible
- Obtener información sobre el estado del servidor

### Interfaces para el manejo del objeto OPC Group

Para el manejo del objeto OPC Group existen las siguientes interfaces:

- *IOPCGroupStateMgt*: Después de que el cliente haya creado un grupo esta interfaz permite el control del comportamiento de ese grupo. Da la posibilidad de cambiar el nombre del grupo, activarlo, desactivarlo, crear una copia...
- *IOPCItemMgt*: Permite al cliente añadir ítems a un grupo

Estas interfaces permiten a las aplicaciones:

- Añadir y quitar ítems de los grupos
- Leer o escribir valores de uno o más ítems en un grupo



### Interfaces para leer y escribir en el objeto OPC Group

Existen tres maneras para que un cliente pueda leer datos de un OPC Group:

- Lectura síncrona
- Lectura asíncrona
- Excepción (suscripción)

La lectura síncrona y asíncrona trabaja con listas específicas de ítems del OPC Group que son proporcionadas por el llamante. La suscripción devuelve cualquier ítem en el grupo que cambia.

Las interfaces que permiten este tipo de operaciones son los siguientes:

- *IOPCAsyncIO*
- *IOPCSyncIO*
- *IDataObject*: Esta interfaz es usada para descolgar callbacks que se usan en los modos de acceso a datos asíncrono y suscripción.

### Interfaces en el cliente

En el cliente solo hay una interfaz *IAdviseSink* que es una interfaz del estándar de Microsoft. Se trata de una interfaz callback.



### 2.3.3.3.- OPC Alarm and Events (OPC A&E)

La especificación OPC A&E describe las interfaces y los objetos que se implementan en los servidores OPC A&E. Estos servidores proporcionan mecanismos que permiten a los clientes estar informados de ciertas condiciones de alarmas. Por otra parte gracias a las interfaces que se implementan en los servidores A&E se da la posibilidad de que los clientes conozcan las alarmas que son soportadas por los servidores y el estado actual de las mismas.

Un servidor OPC A&E genera:

- La conexión de clientes
- La suscripción de condiciones y subcondiciones
- La notificación de alarmas

Los servidores OPC Alarm & Event son necesarios para señalar:

- Alarmas sobre datos de sensores: temperatura, presión, etc.
- Alarmas sobre parámetros de control: stop, open, close.
- Actualizaciones sobre estado de la información.
- Estado de la conexión hardware:
  - Estado del software local.
  - Completar las secuencias de sistemas como “batch”.
- Otro tipo de eventos que no se ajustan dentro de servidores.

OPC DA OPC no está limitado a datos de sensores. Puede exponer cualquier tipo de datos. Basta con crear un servidor software OPC para exponer los datos y su espacio de nombre. Hay que remarcar que la interfaz OPC es una interfaz software, no se necesita ningún hardware específico o infraestructura de red.



## Conceptos

- Alarmas: Según OPC una alarma es una condición anormal sobre el proceso y corresponde a una OPC Condition específica. Una alarma puede estar asociada o no a alguna condición y en caso de no estarlo simplemente representar un estado interno, que expresa algo significativo en el contexto del servidor.
- Condiciones: Una condición es un estado nombrado en el servidor o en uno de sus objetos susceptibles de interesar a un cliente. Por ejemplo el tag FIC 101 puede tener las condiciones “LevelAlarma” asociada a él. Una condición puede tener asociada a ella subcondiciones. Las condiciones pueden ser de estado único o multiestado (HighAlarm, HighHighAlarm LowAlarm, LowLowAlarm). Además pueden tener asociados tres estados variables: enabled, active y acknowledge. Las condiciones se pueden asociar a áreas. Esto afecta a dos áreas de la interfaz: filtering y browsing.
- Eventos: Un evento es un acontecimiento detectable el cual es significativo para el servidor OPC Event, el dispositivo al que represente y sus clientes OPC. El evento puede estar asociado o no a una condición. Por ejemplo las transiciones dentro de la condición “LevelAlarm” y la vuelta a la normalidad de la condición son eventos asociados a condiciones, sin embargo los cambios en la configuración del sistema y los errores del sistema son ejemplos de eventos que no están relacionados con condiciones específicas. Los eventos no tiene representación directa en el modelo OPC.

OPC distingue tres tipos de eventos:

- Condition Related: Transiciones en estados o fuera de ellos sobre eventos asociados a una condición
- Tracking Related: Eventos que no están asociados con alarmas pero que implican la interacción entre un cliente y objeto “Target” dentro del servidor OPC A&E

- Simple: Los otros eventos. Utilizados para codificar los errores materiales.
- Fuente: Una fuente está asociado a una condición. Puede ser un tag, un equipo o cualquier sub-sistema.
- Zonas: Los eventos y condiciones disponibles en un servidor están organizados en una o más áreas o zonas. Una zona es el agrupamiento del equipo de planta configurado por el usuario. Si las zonas están disponibles, el cliente deberá crear un objeto `OPCEventAreaBrowser` para explorar la organización de las zonas de proceso. El cliente podrá entonces filtrar las suscripciones especificando las zonas de proceso y limitando así los eventos que le envía el servidor.

### El Modelo Lógico OPC A&E

La fuente de los datos puede ser un servidor OPC A&E, un servidor DA, o un equipo.

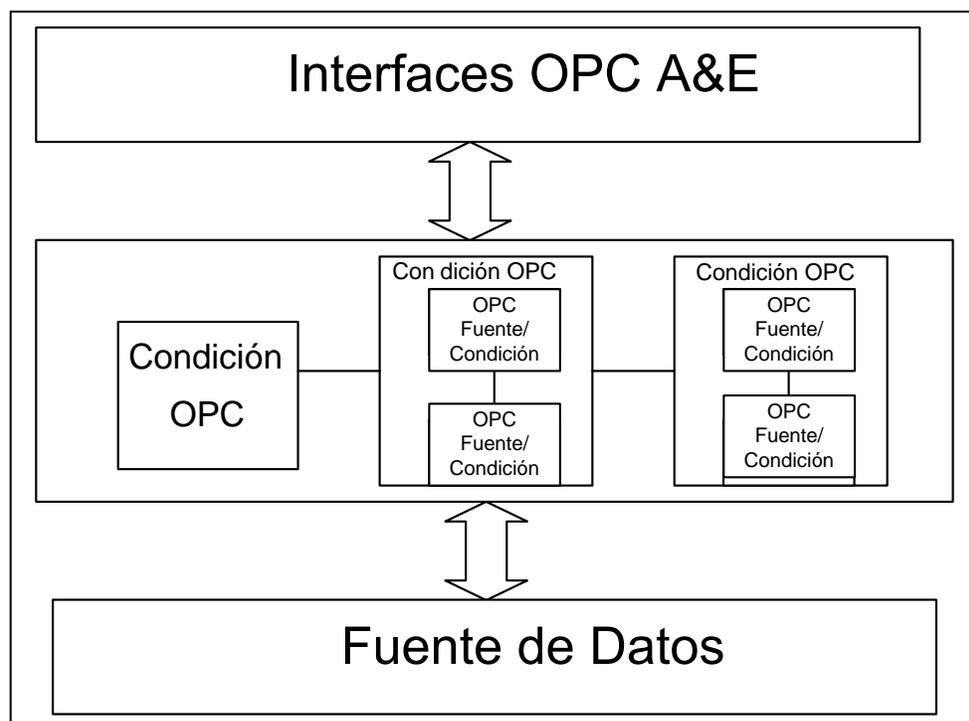


Figura 2.28 Modelo lógico de un objeto servidor A&E. Fuente: OPC Foundation



El objeto servidor OPC es el objeto COM al cual las aplicaciones se conectarán. Cada cliente verá un objeto COM lógico independiente. debajo hay un colección de objetos subscriptores. Estos son creados de forma dinámica por la aplicación cliente. Se supone que la mayoría de las aplicaciones serán capaces de usar un único subscriptor con un único filtro.

### Diseño de un Servidor

El servidor proporciona el manejo de los subscriptores OPC y también controla, media y optimiza en el acceso por múltiples clientes El servidor es básicamente un driver I/O que entiende como hablar con algunos vendedores de alarmas y eventos (hardware o software). El servidor reporta los eventos generados por este proveedor vía las interfaces OPC.

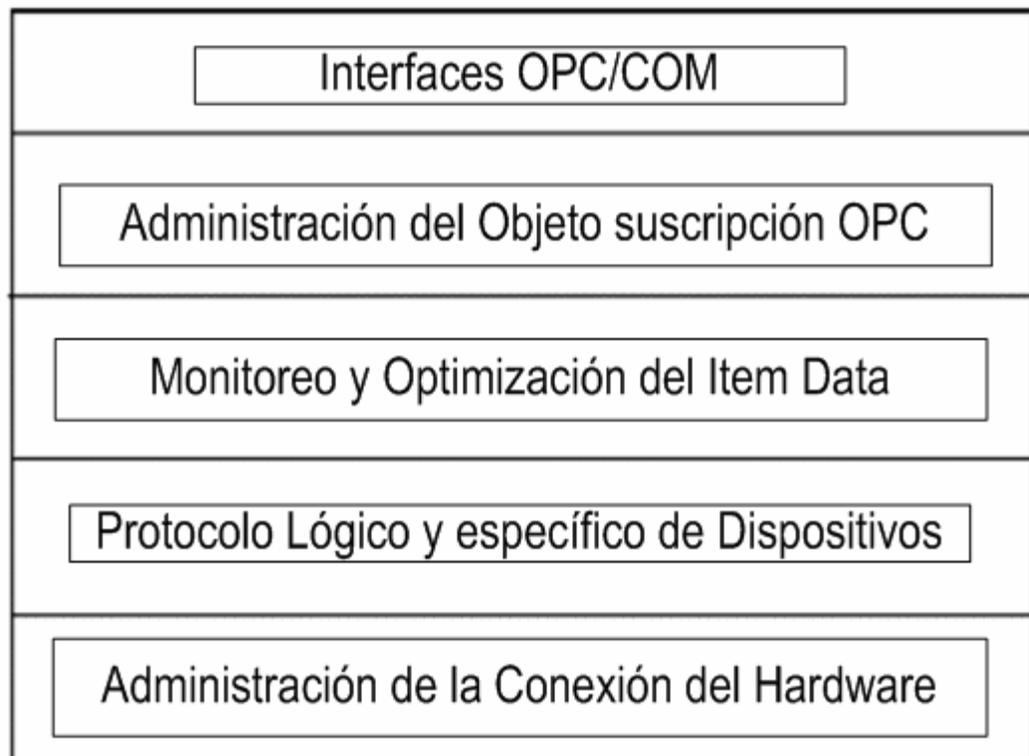


Figura 2.29 Diseño de un servidor. Fuente: OPC Foundation



## **Problemas no Resueltos por OPC A&E**

Hay una serie de problemas que OPC Alarms & Events no intenta resolver:

- Global naming: subscripción a uno o más servidores. La manera en que múltiples servidores resuelven el conflicto de nombres no se especifica en OPC. Por lo tanto cuando se conecta a un servidor, se tiene que saber el 'PROGID' del servidor y a que nodo físico conectarse.
- Data Acces: Esto se lleva a cabo en otra especificación OPC.
- Security: Pertenece a otra especificación OPC.
- Configuración de las alarmas: Igual que el caso de acceso a datos esto se deja al fabricante.

## **Definición de Interfaces OPC A&E**

Cualquier objeto COM que soporte la interfaz IOPCEventServer es un servidor OPC de eventos.

En muchos casos un servidor OPC Data Access que tenga un objeto OPCEventServer cumplirá los dos papeles de servidor de datos y de servidor de eventos.

Sin embargo habrá ocasiones en las que sea más ventajoso tener un servidor dedicado exclusivamente a eventos, es decir un servidor que no sea también un servidor OPC Data Access.

En la especificación se definen los siguientes objetos COM:

## OPC EventServer

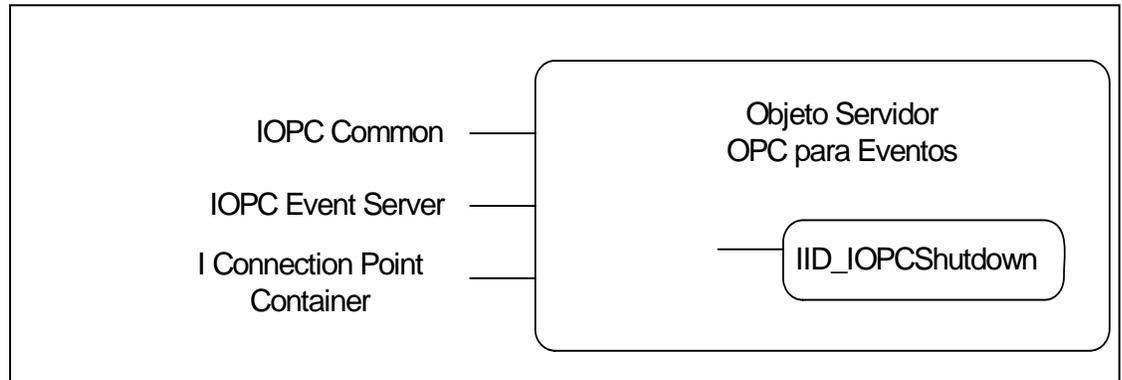


Figura 2.30 OPCEventServer. Fuente: OPC Foundation

Este objeto proporciona las siguientes interfaces:

- *IOPCommon*: Hace posible la implementación de funciones que son comunes a otros servidores OPC.
- *IOPCEventServer*: Se usa para generar los siguientes objetos: OPC Event Subscription y OPC Event Area Browser.
- *IconnectionPointContainer*: Son interfaces estándar DCOM para objetos conectables.

Estas interfaces permiten a la aplicación:

- Traducir código de errores en texto inteligible
- Obtener información del estado del servidor
- Crear y manejar suscripciones
- Determinar el criterio de filtro disponible para el servidor
- Explorar la áreas de alarma y condiciones disponibles

## OPCEventSubscription

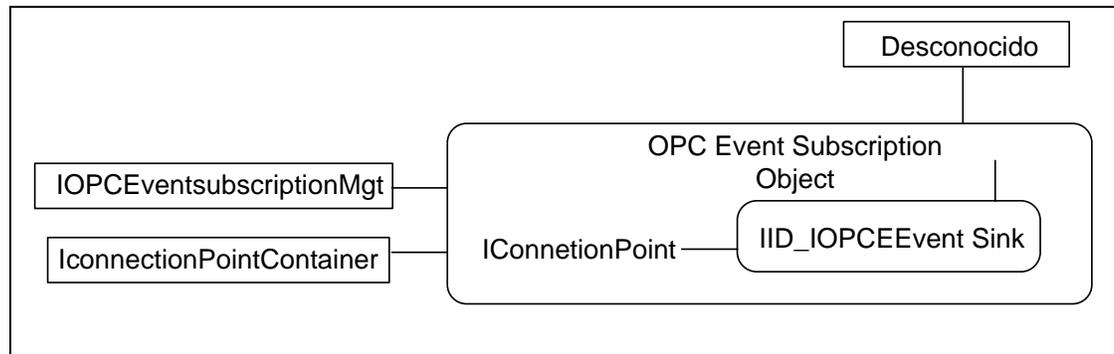


Figura 2.31 OPC EventSubscription. Fuente: OPC Foundation

Las interfaces de este objeto son:

- *IOPCEventSubscriptionMgt*: Se usa para configurar filtros y otros atributos.
- *IConnectionPointContainer*: Son interfaces estándar DCOM para objetos conectables.

Estas interfaces permiten al cliente:

- Establecer una conexión callback.
- Parar y resumir la conexión.
- Establecer y modificar los criterios del filtro.
- Indicar que parámetros adicionales tienen que ser devueltos con la información básica cuando una condición cambie de estado.
- Obtener una actualización del estado de todas las condiciones monitorizadas.

## OPCEventAreaBrowser

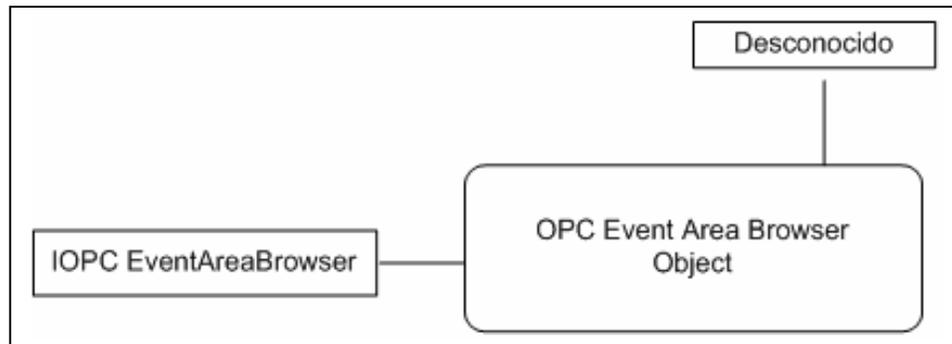


Figura 2.32 OPC EventAreaBrowser. Fuente: OPC Foundation

Este objeto es opcional, su interfaz es:

- *IOPCEventAreaBrowser*: proporciona a los clientes una manera de explorar la organización del área de proceso implementada en el servidor.

### 2.3.3.4.- OPC HDA 1.X

La especificación OPC HDA define la manera en que son tratados los archivos OPC. Por otra parte en esta especificación se definen también las interfaces que permiten a los clientes OPC HDA acceder a los datos archivados en los servidores OPC HDA. Un servidor OPC HDA puede guardar los datos de un servidor DA o directamente en un formato propietario. Un servidor DA puede ser también un servidor HDA. Un cliente HDA puede ser un módulo de curva de tendencia, un módulo estadístico o cualquier módulo que utilice archivos.

La especificación OPC HDA es complementaria pero independiente de las especificaciones OPC DA y OPC A&E. Los puntos comunes se reagrupan en la



especificación común. Hace referencia a servidores A&E cuando estos soportan las interfaces específicos de OPC HDA.

### Conceptos

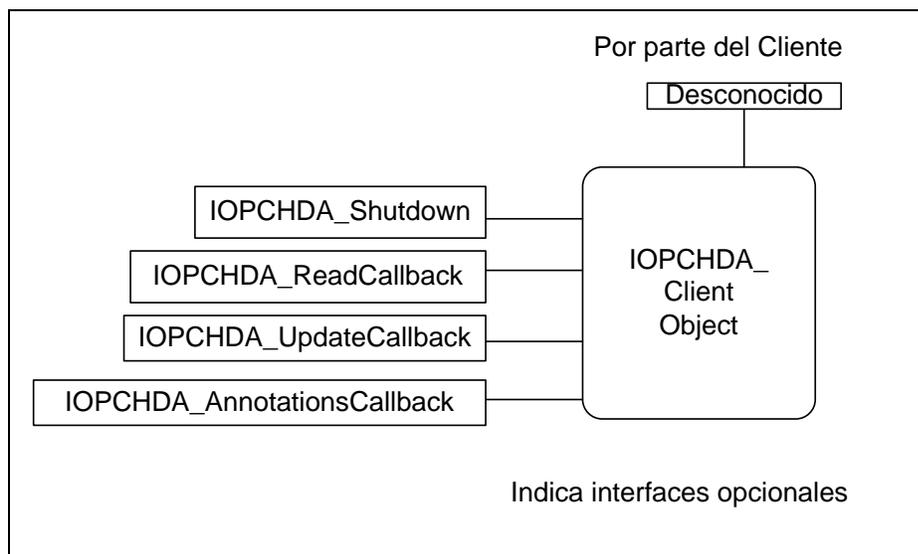
- *Atributos*: Describen los parámetros propios de los Item como el nombre, tipo de dato, unidad de conversión, fuente de dato. Los atributos específicos pueden estar definidos para cada servidor.
- *Agregados*: Permiten especificar el tratamiento a los datos. El significado de cada agregado está ligado a cada servidor. Los agregados no soportados por un servidor deben asociarse a <OPC\_E\_INVALIDARG> en el código.

Los agregados suplementarios a los agregados definidos por la fundación OPC pueden estar añadidos en el servidor a favor de la implementación, se reserva los atributos de 0 a 0x7.

- *Items*: La estructura de Items OPC HDA comprende el identificador del Item, su valor, su marca de tiempo, su calidad, el agregado al que pertenecen y sus atributos.

### Definición de Interfaces OPC HDA

Los objetos del servidor OPC HDA proporcionan la capacidad de leer datos del servidor histórico y de escribir datos en el servidor. El tipo de datos históricos son dependientes del servidor. Se accede a todos los objetos COM a través de interfaces. El cliente solo ve las interfaces. Los objetos que se describen a continuación son representaciones “lógicas” que no tienen nada que ver con la implementación interna del servidor. Las figuras que se muestran a continuación son una suma de los objetos OPC y sus interfaces. Las interfaces que son opcionales se indican entre corchetes ([ ])



**Figura 2.33 Interfaz del cliente OPC HDA.** Fuente: OPC Foundation

La petición Shutdown es necesaria para permitir al servidor OPC HDA apagarse de forma limpia y sin alteraciones. El cliente debe liberar toda la memoria que le ha sido proporcionada por el servidor y finalizar todas las conexiones.

La interfaz Browser proporciona al cliente un método para revisar el espacio de direcciones de los históricos. Este espacio de direcciones puede ser jerárquico o plano en los servidores. Esta interfaz está diseñada para soportar una estructura jerárquica, de esta forma un espacio de direcciones plano es representado como un único nivel jerárquico. La interfaz browser es indispensable en servidores HDA de gran tamaño ya que permite a los clientes revisar el espacio de direcciones de una manera gráfica simple.

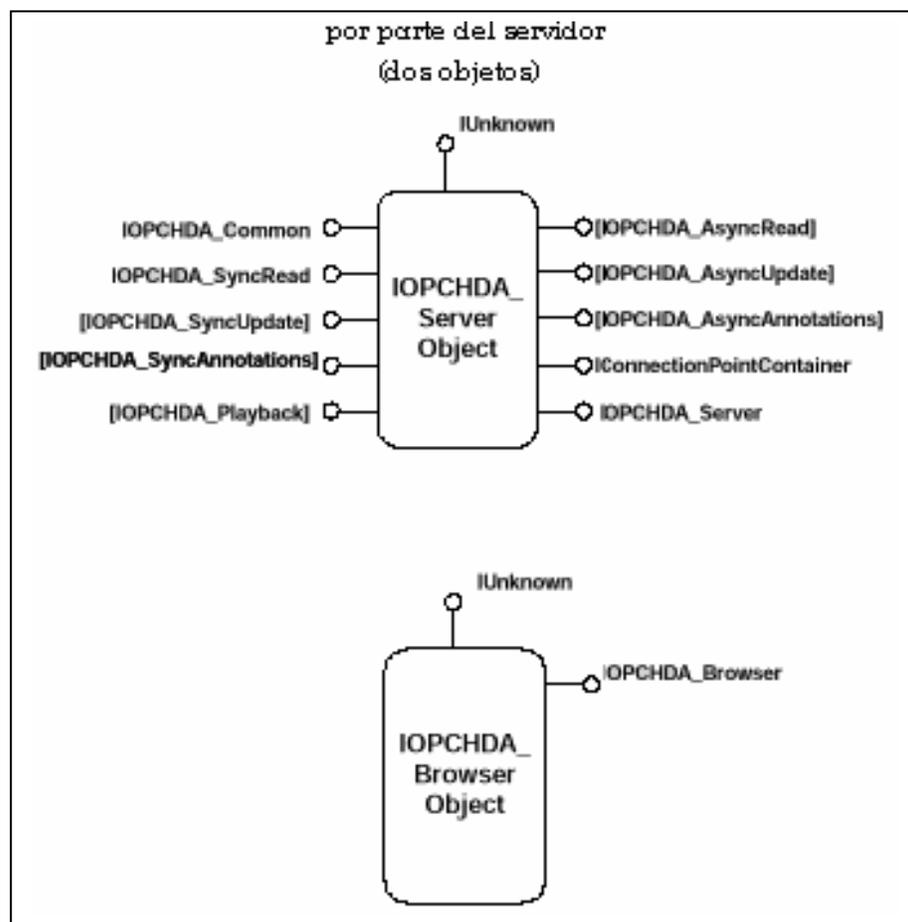


Figura 2.34 Interfaz del servidor OPC HDA. Fuente: OPC Foundation

### 2.3.3.5.- OPC Batch

Esta especificación está construida sobre otros estándares que son OPC DA y la especificación IEC 61512-1. En la especificación OPC Batch, el nombre de los ítems respeta una sintaxis definida en la especificación IEC 61512-1. La especificación IEC aporta una terminología que está integrada en OPC Batch.

La especificación OPC Batch define:

- Los datos relativos a lotes en curso de ejecución.
- La información relativa al equipamiento indispensable para comprender el contexto de ejecución Batch.



- El registro de la ejecución del lote.
- El contenido de los modelos del Batch.

### **El Modelo Funcional OPC Batch**

OPC Batch se usa para enviar modelos de lotes en un proceso de manufactur y monitorear su ejecución(sistemas de laboratorio, sistemas de control por lotes, etc.). Es importante el hecho de que cada servidor (cliente) OPC Batches es al mismo tiempo un servidor (cliente) OPC DA, en otras palabras un servidor (cliente) OPC Batch incluye aparte de la especificación OPC Batch, la especificación OPC DA incluyendo algunas interfaces opcionales.

Los servidores OPC Batch permiten acceder a datos llamados “Data Source”. El tipo de estos “Data-Sources” dependen de la implementación del servidor.

La especificación OPC Batch define un espacio de nombres fijo usando tres modelos. Los tres modelos definidos en la especificación son:

- Modelo físico: Es una colección jerárquica de los tags de los equipos incluidos en zonas, células de proceso, unidades, módulos de control.
- Modelo Batch: Es una colección de procedimientos.
- Lista Batch ID: Proporciona un método sencillo de obtener los identificadores Batch.

## Definición de Interfaces OPC Batch

Por parte del servidor:

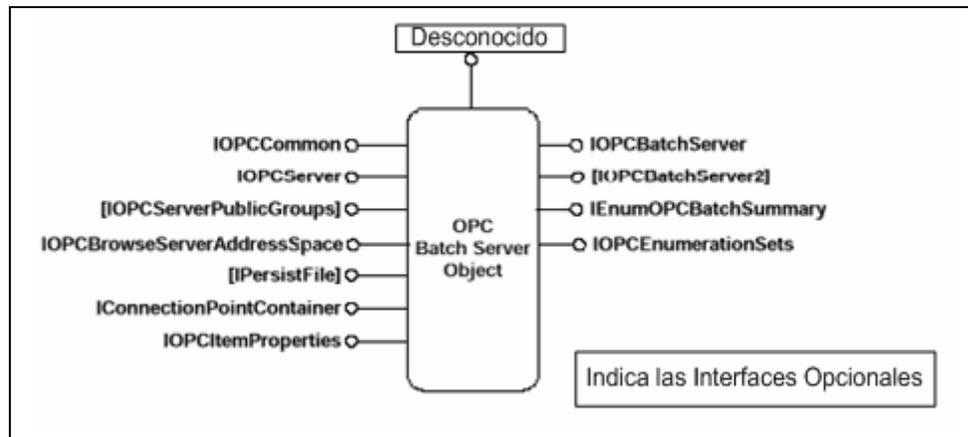


Figura 2.35 Interfaces del servidor OPC Batch. Fuente: OPC Foundation

Por parte del cliente:

Las interfaces son las mismas que para el servidor OPC DA: No hay específicos para los clientes Batch.

### 2.3.3.6.- OPC DX

La especificación OPC DX (Data Exchange) regula las comunicaciones servidor a servidor a través de redes de comunicación Ethernet. Esta especificación permite servicios de gestión y configuración remotos. Se trata de extensiones del estándar OPC Data Access.

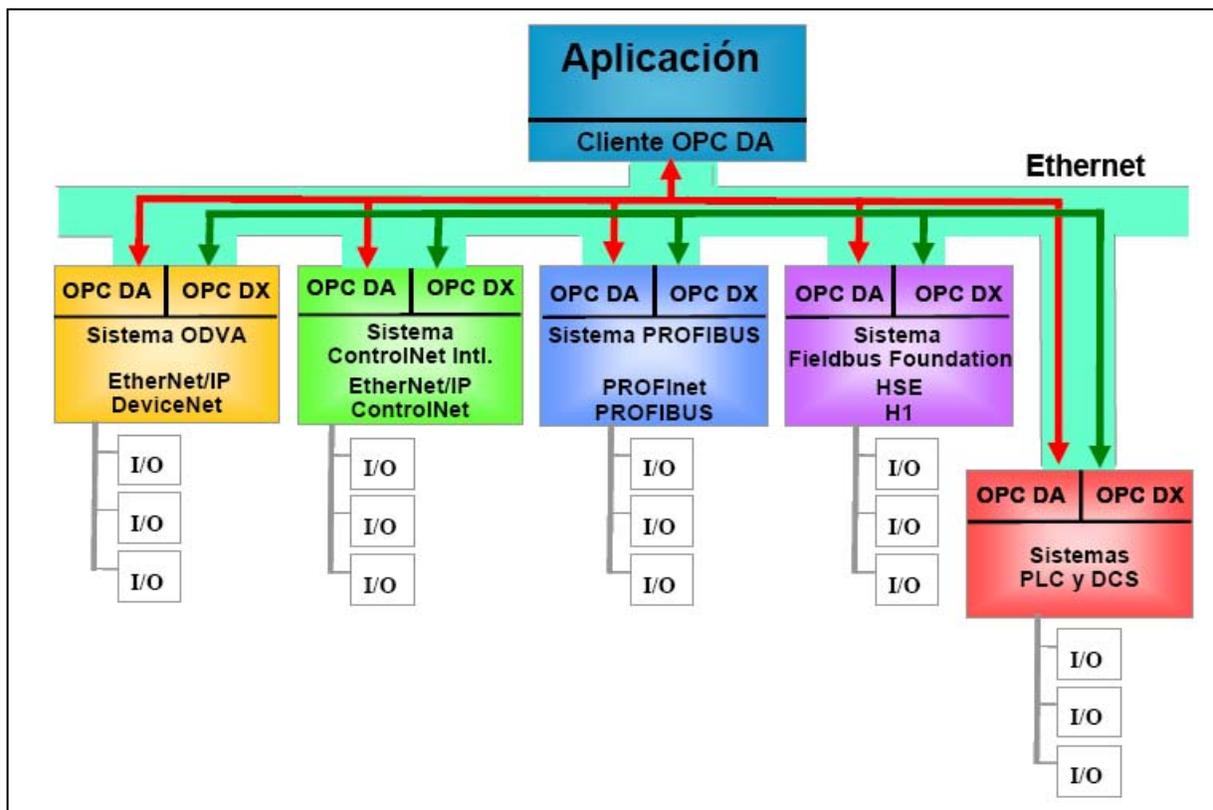


Figura 2.36 OPC DX. Fuente: OPC Foundation

### 2.3.3.7.- OPC XML

La Fundación OPC ha desarrollado la especificación OPC XML cuyo objetivo es desarrollar flexibles y consistentes regulaciones para hacer disponibles datos tecnológicos vía OPC usando XML (Extensible Markup Language) en la Internet/Intranet.

Las características de XML permiten la fácil escritura de cualquier tipo de estructura de datos y al mismo tiempo, una transmisión vía Internet de archivos con formato XML, justo igual que con el formato HTML.



### 2.3.4.- Ejemplo de la conexión en una red industrial utilizando OPC

Una forma sencilla de cómo mostrar el por qué se necesita utilizar OPC en el control de procesos en lugar de los circuitos electrónicos convencionales como relés, relés temporizados, etc, se ilustrará a través de un ejemplo de un caso estudiado desarrollado por una organización del negocio de las petroquímicas. En la figura 2.37, se muestra los dispositivos y las aplicaciones de software que estaban funcionando en esta planta petroquímica en particular.

El PLC, el sensor de vibración, y el motor de cálculos trabajan bajo diferentes protocolos de comunicación – TSAA, ModBus y DDE. Así que las aplicaciones de software que necesitan data de estos dispositivos también necesitan estar ambientados con los protocolos TSAA, ModBus y DDE. Adicionalmente cada dispositivo es servidor de las tres aplicaciones. El historial de proceso, la aplicación GUI proveyendo una interfaz hombre-máquina y el monitor de la Condición de la Máquina, todos comunicándose con todos los dispositivos resulta en una complejidad. Este despliegue le tomó a la compañía cerca de 10 días y un costo aproximado de \$50.000.

Ahora, este mismo esquema puede ser desarrollado de una manera más inteligente utilizando servidores OPC. En la figura 2.38 se muestra el nuevo esquema y se puede observar cómo se reduce la complejidad. Ahora todas las aplicaciones de software hablan únicamente al servidor OPC y responden también solamente al OPC. El servidor OPC provee la Caja Negra para las aplicaciones de software alrededor del hardware.

La nueva implementación tomó solo 2 días para desplegarse y incurrió en un costo de aproximadamente \$10.000. Pudieron haber eliminado dos servidores OPC e implementado dispositivos driver justo dentro de un único servidor OPC.

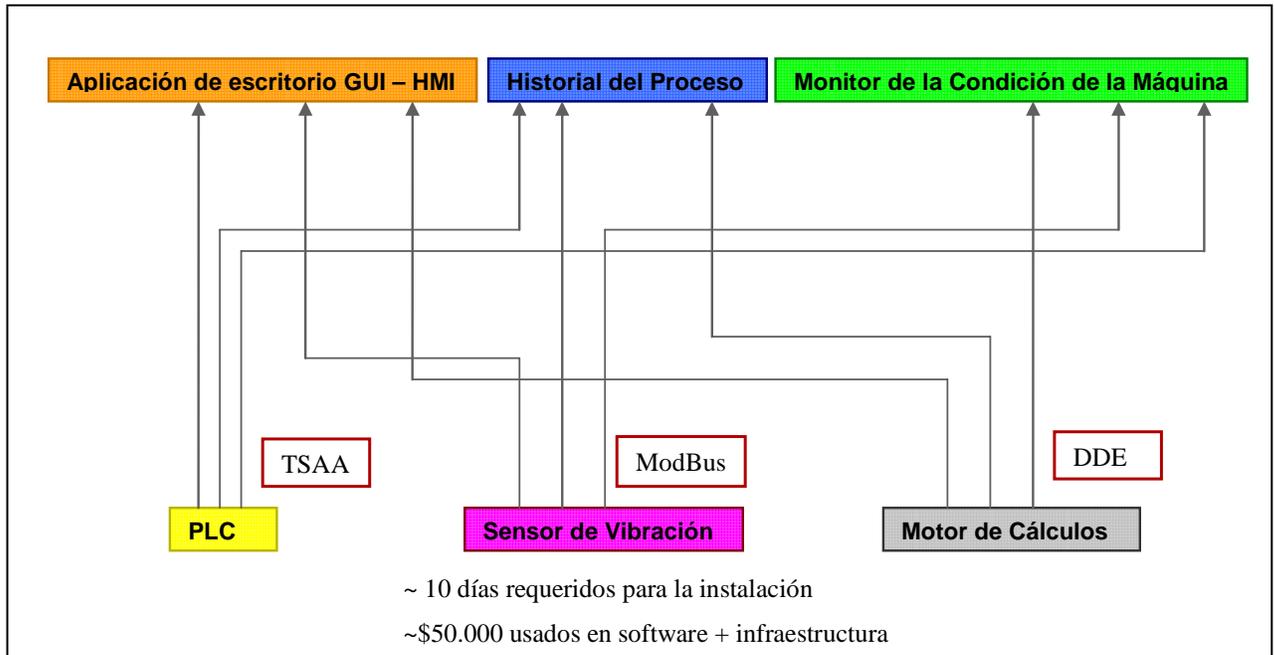


Figura 2.37 Esquema de instalación para el caso estudiado. Fuente: OPC Foundation

Como se observó, los beneficios son:

- Reducción de carga en el dispositivo.
- La capacidad de actualización ha aumentado.
- Los servidores OPC proveen captura de data.
- Las aplicaciones del cliente no necesitan de saber los detalles del protocolo del hardware.
- Incremento de la vida del dispositivo (ya no necesita servir más a múltiples servidores).
- Interoperabilidad (Unix/Linux y Windows – ambas plataformas son soportadas por los OPC).
- Estandarización.

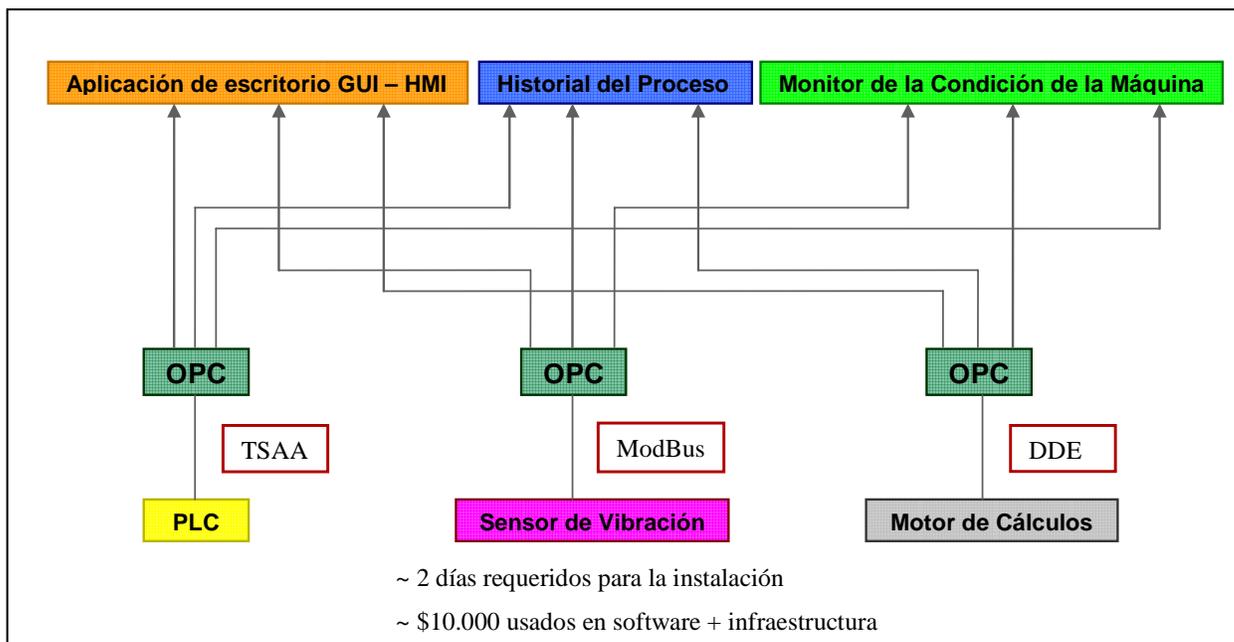


Figura 2.38 Esquema con la solución OPC. Fuente: OPC Foundation

### 2.3.5.- Cliente OPC

El Cliente OPC es una herramienta de supervisión, adquisición y análisis de datos tomados directamente del servidor OPC.

Una vez que el servidor OPC está escrito para un dispositivo en particular, puede ser utilizado nuevamente por una aplicación capaz de actuar como un cliente OPC.

El principal objetivo del cliente OPC es disponer de una manera fiable de los datos de proceso que el servidor OPC adquiere de los autómatas para poder mejorar y optimizar el funcionamiento de la instalación. Para ello se basará en los datos adquiridos por el servidor OPC. En el pasado, el cliente disponía solamente de ciertos datos que no eran fiables y que dependían demasiado de los errores humanos al realizar las lecturas de las diferentes variables que podían adquirir a través de hojas de cálculo y apuntes manuscritos de los operarios. Esta modalidad de trabajo le llevaba en numerosas



ocasiones a realizar cambios en el funcionamiento del proceso sin saber realmente si dichos cambios iban o no a representar una ventaja en producción.

La tecnología OPC proporciona la posibilidad al cliente OPC de no errar por carencia de datos o por datos mal adquiridos.

Cada cliente interroga al servidor que contiene los datos que necesita. Los datos estarán disponibles para todas las aplicaciones que acceden vía OPC, lo que provoca que los clientes OPC no sólo puedan utilizar estos datos sino también compartirlos con terceros.

El Cliente OPC puede ser desde una tabla Excel que muestre los datos del proceso suministrados por el PLC a través del servidor OPC hasta una simulación gráfica que visualmente represente la misma información.

El Cliente OPC extiende el ambiente de desarrollo gráfico con funcionalidad adicional para un rápido desarrollo de aplicaciones de medición distribuida, control y monitoreo de alta cuenta de canales. El Cliente OPC hace fácil graficar los históricos de las tendencias en tiempo real, mejorar la seguridad en pantallas principales, tener acceso a datos automáticamente, así como agregar un sistema de alarma, escala y seguridad a la variable compartida.

## 2.4.- DEFINICIÓN DE TÉRMINOS BÁSICOS.

- **Redes de trabajo industrial:** las redes de comunicación industrial permiten conocer todo lo referente a un proceso industrial a través de las variables fundamentales medidas por instrumentos instalados en campo, permitiendo a la gerencia saber como está funcionando su empresa. Con esto se busca que estén intercomunicados dispositivos tales como: sensores, actuadores, PLCs, microcontroladores, máquinas, computadores, controladores, y en general todos los dispositivos involucrados en un



sistema de automatización industrial; con el fin de sincronizar todo el proceso de producción de la planta.

- **Protocolo de comunicación:** es un lenguaje para la comunicación de información. Son las reglas y procedimientos que se utilizan en una red para comunicarse entre los nodos que tienen acceso al sistema de cable. En su forma más simple, un protocolo se puede definir como las reglas que gobiernan la semántica (significado de lo que se comunica), la sintaxis (forma en que se expresa) y la sincronización (quién y cuándo transmite) de la comunicación.
- **OLE:** Object Linking and Embedding (OLE) es un entorno unificado de servicios basados en objetos modificables y cuya arquitectura puede ser extendida a través de servicios a medida. OLE fue desarrollado por Microsoft con el objetivo general de permitir un alto grado de integración entre sus componentes.
- **Software:** se denomina software, programática, equipamiento lógico o soporte lógico a todos los componentes intangibles de una computadora, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica.
- **Hardware:** el hardware se refiere a todos los componentes físicos (que se pueden tocar), en el caso de una computadora personal serían los discos, unidades de disco, monitor, teclado, la placa base, el microprocesador.
- **Driver:** un controlador de dispositivo (llamado normalmente controlador o, en inglés, driver) es un programa informático que permite al sistema operativo interactuar con un periférico, haciendo una abstracción del hardware y proporcionando una interfaz, posiblemente estandarizada, para usarlo. Se puede esquematizar como un manual de instrucciones que le indica cómo debe controlar y comunicarse con un dispositivo en particular. Por tanto, es una pieza esencial, sin la cual no se podría usar el hardware.



# **CAPÍTULO III**

## **MARCO METODOLÓGICO**



### **3.1.- NIVEL DE LA INVESTIGACIÓN**

Este trabajo sigue el método de investigación de proyecto factible ya que se pretende resolver un problema o satisfacer una necesidad ampliada en una organización determinada previamente. También, por el propósito de este trabajo, se puede definir como una investigación de carácter aplicable, ya que será puesto en marcha en el momento en que la posibilidad de financiarlo lo permita.

### **3.2.- DISEÑO DE LA INVESTIGACIÓN**

Para llevar a cabo este trabajo especial de grado es necesario cumplir con una serie de actividades hasta llegar a cumplir el objetivo planteado anteriormente. Por lo tanto, se seguirán las siguientes actividades:

#### **1. Búsqueda de información**

En esta fase, se investigará todo lo relacionado con los PLC, su forma de programación, su estructura y lenguaje de comunicación. Asimismo, se buscará la forma de relacionarlos con los computadores para lograr una comunicación bidireccional.

#### **2. Determinación de las formas de comunicación entre un PLC y un Computador**

Habrá que determinar primero cuáles son las formas más comunes que se utilizan en la actualidad para conectar los PLC con los computadores, de manera que en la fase siguiente escoger el sistema de comunicación que va a utilizar.

#### **3. Selección del sistema de interconexión**

Una vez que se haya investigado sobre las maneras posibles en que se puede conectar un PLC con un computador hay que seleccionar el sistema de interconexión que se utilizará. Cabe destacar que este sistema de interconexión debe ser universal, es decir, que sea un sistema de comunicación estándar donde, sin importar el fabricante de PLC o el tipo de red de comunicación industrial, se pueda comunicar con el computador.



#### **4. Definición de la arquitectura de comunicación**

En esta fase, se establecerá la estructura o la forma que tendrá la conexión del PLC y el computador. Se definirá el hardware de conexión para el método seleccionado.

#### **5. Selección del software**

Una vez establecido el sistema de interconexión y la arquitectura de comunicación, se seleccionará el programa con el que se mostrará la interacción entre el PLC y el computador. El programa tiene que trabajar en ambiente Windows que es el más conocido y empleado por los usuarios de computadoras. Por supuesto, el programa seleccionado debe ser compatible con los equipos que se van a utilizar. El programa también debe percibir los cambios en los valores de entrada y salida y mostrarlos en tiempo real.

#### **6. Diseño de la aplicación en el computador**

Con base en el programa seleccionado se diseñará la aplicación o la forma en que se va a visualizar las entradas y salidas del PLC. Esto se representará gráficamente lo que hará posible que el usuario pueda interactuar con el sistema de control, de manera que pueda cambiar valores en la entrada y ver cómo el PLC responde a esas señales.

#### **7. Implementación del sistema de interconexión entre el PLC y el computador**

En esta fase final se implementará la interconexión entre el PLC y el computador de forma que se pueda mostrar que el computador está interactuando con el PLC.

#### **8. Pruebas y ajustes finales**

Será necesario que una vez implementado el sistema de interconexión se hagan las pruebas de su funcionamiento a fin de hacer los ajustes necesarios que garanticen el funcionamiento óptimo de la aplicación.



# **CAPÍTULO IV**

## **DESARROLLO DEL PROYECTO**

## 4.1.- FORMAS DE COMUNICACIÓN ENTRE UN COMPUTADOR Y UN PLC

Con el fin de que el computador aloje alguna aplicación y el PLC controle la misma, existen, por lo menos, tres formas de interconectar un computador a un PLC, a través de:

- Puertos Digitales
- Redes de comunicación industrial, o
- Servidores OPC (OLE for Process Control)

### Puertos Digitales

Es la forma más sencilla y directa de comunicar un PLC con una computadora personal. Un puerto de entrada o salida digital se comunica a una tarjeta instalada en el computador, lo que permite una comunicación efectiva y en tiempo real, de manera que cuando el PLC dé una señal inmediatamente se detecta el cambio y se observa dicha variación en el computador.

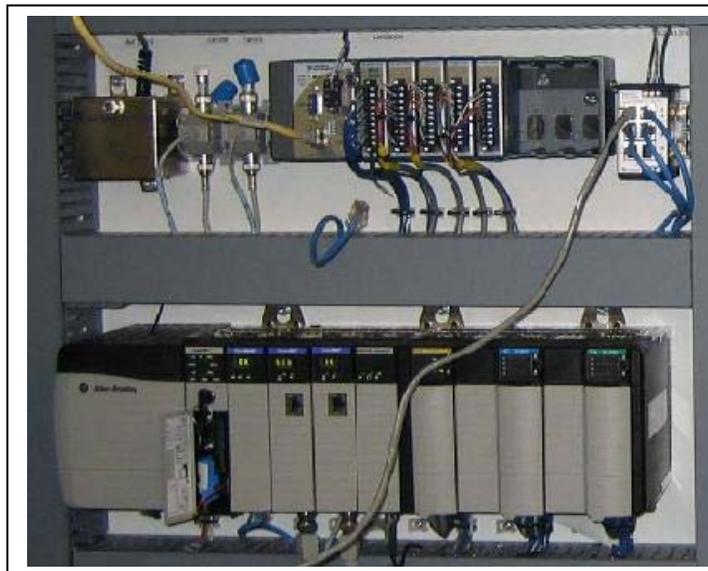


Figura 4.1 Controladores conectados a través de E/S digitales. Fuente: [www.ni.com](http://www.ni.com)



Este método permite a los usuarios de PLC integrar fácilmente Entradas y Salidas (E/S) digitales dentro de su programación (Ej.: Formato diagrama escalera), además que permite velocidades de respuesta muy altas. Las altas velocidades de respuesta se logran debido a la sencilla y directa estructura de comunicación, la cual se establece, por lo general, a través de un cable decodificador suministrado por el fabricante. Para efectuar la comunicación el método más común sería alternar una sola línea digital, permitiéndole enviar un bit de datos para aspectos como estatus, o pasa/falla. Si requiere sacar más información, tal como códigos de errores, números grandes, o implementar un protocolo, pueden utilizarse múltiples puertos o líneas de E/S digital. Por ejemplo, con 8 líneas de E/S Digital, se pueden leer y escribir hasta 256 valores distintos. Finalmente, pueden generarse pulsos a través de una línea digital. La generación de pulsos puede utilizarse para contar con tiempos y disparos precisos en los dispositivos de automatización, como en actuadores y PLCs. Pero también tiene sus desventajas ya que en ocasiones hay la necesidad de utilizar los mismos puertos digitales (por los cuales se efectúa la comunicación) para otras aplicaciones.

### **Redes de Comunicación Industrial**

La comunicación entre el PC y el PLC esta basada en diferentes protocolos que existen hoy en día y que permiten a la computadora establecer (intercambio de datos o información) con sensores o dispositivos inteligentes, PLC, redes industriales, es decir, está basado en un protocolo industrial. Este método tiene como ventaja que es directo, es decir, se puede lograr una comunicación efectiva con el procesador. Sin embargo, algunos protocolos y buses de comunicación industrial no son soportados.

Entre los protocolos y buses de comunicación más conocidos en la industria están Profibus, Modbus y DeviceNet, entre otros. En la figura 4.2 se muestra una interfaz CANopen que es otra de las formas de comunicación industrial. Estos

protocolos no son compatibles entre sí, es decir, no pueden compartir datos, información, etc, una vez que se adoptan cualquiera de estos dispositivos como protocolo de comunicación el usuario esta confinado a trabajar solo con dicho protocolo y tiene restricciones a la hora de realizar ampliaciones o introducir nuevos dispositivos a su red industrial.



**Figura 4.2 Interfaces CANopen, una forma de Comunicación industrial.** Fuente: [www.ni.com](http://www.ni.com)

### **Servidores OPC**

OPC es un estándar de comunicación que fue desarrollado específicamente para el campo de control y para la supervisión de procesos. Se originó cuando era necesario un mismo lenguaje de comunicación entre diferentes tipos de PLC con diferentes tipos de comunicación (Ethernet, Profinet o CCLink), como se muestra en la figura 4.3. El OPC, entonces, es un servidor que funciona como un estándar o conjunto de reglas que deben cumplir los diferentes fabricantes de PLC para comunicar diferentes tipos de PLC con un mismo lenguaje. Basado en estas cualidades OPC se ha convertido en los últimos años en el método de conectividad preferido por el 68% de las aplicaciones que involucran PLC.

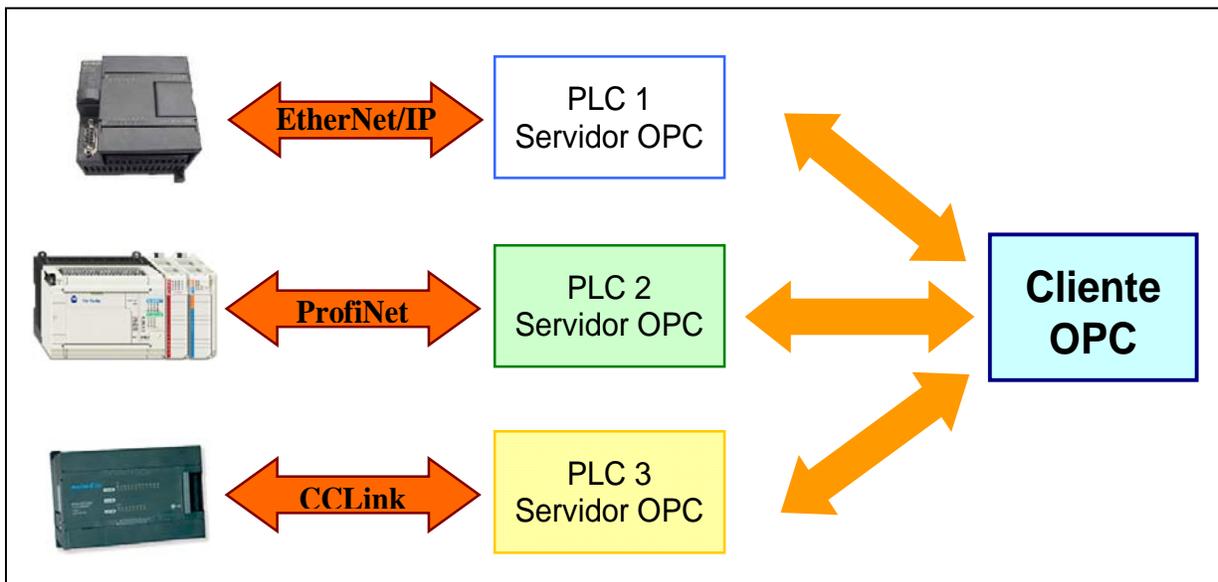


Figura 4.3 Método de conectividad por medio de un OPC. Fuente: OPC Foundation

Se han mencionado ya tres formas (al menos), de conectividad entre un PLC y un computador. Cada una de ellas tiene sus ventajas y desventajas, dependiendo de su aplicación. Hay aplicaciones que requieren velocidad de respuesta inmediata, existen otras en lo que es importante es la capacidad de transmisión de datos, etc..

Para este caso la necesidad de alcanzar costos reducidos en la implementación y tener un estándar de comunicación establecido, son vitales para la selección del método de conectividad.

En la Tabla 4.1 se muestra un resumen de algunas de las ventajas y desventajas de cada uno de los métodos de interconexión entre un controlador lógico programable y un computador. Solo se mencionan las ventajas y desventajas más notables, no se procede a un análisis detallado de cada una de las formas de conectividad.



Tabla 4.1 Comparación entre Métodos de Interconexión entre el PLC y el PC.

<i>Método de Interconexión</i>	<i>Ventajas</i>	<i>Desventajas</i>
<b>Puertos Digitales</b>	<ul style="list-style-type: none"><li>- Comunicación en tiempo real.</li><li>- Suministrada por el fabricante</li><li>- Conexión Sencilla</li></ul>	<ul style="list-style-type: none"><li>- Transmisión de datos limitada.</li><li>- Incompatibilidad con otras marcas</li></ul>
<b>Redes de Comunicación Industrial</b>	<ul style="list-style-type: none"><li>- Amplia capacidad y velocidad de transmisión de datos</li></ul>	<ul style="list-style-type: none"><li>- Incompatibilidad entre protocolos</li><li>- Costos adquisición</li></ul>
<b>Ole for Process Control OPC</b>	<ul style="list-style-type: none"><li>- Compatibilidad entre distintas marcas.</li><li>- Costos de implementación reducidos.</li><li>- Estándar de comunicación.</li></ul>	<ul style="list-style-type: none"><li>- Mediciones repetitivas e instantáneas</li></ul>

#### **4.2.- SELECCIÓN DEL MÉTODO DE INTERCONEXIÓN ENTRE EL PLC Y EL COMPUTADOR.**

Habiendo evaluado las diferentes ventajas y desventajas de cada uno de los métodos expuestos anteriormente, se observa que el método de interconexión que cumple con los requerimientos, necesidades y expectativas de este proyecto, tales como: el establecimiento de un estándar de comunicación, bajos costos de implementación y compatibilidad entre distintos fabricantes de PLC; es el método de Servidores OPC. A continuación se exponen con más detalles las razones por las cuales se seleccionó este método:



- Establece un estándar de comunicación. El OLE para control de procesos (OPC) es un estándar abierto para compartir datos entre dispositivos de campo y aplicaciones de ordenador basado en OLE de Microsoft. Permite a las aplicaciones leer y escribir valores de proceso y que los datos sean compartidos fácilmente en una red de ordenadores. El estándar, gobernado por la Fundación OPC, es de dominio público y se encuentra disponible para cualquier usuario.
- Genera menores costos de implementación y de operación en comparación con los otros métodos de interconexión. Tradicionalmente, los fabricantes de software para acceso de datos de proceso tenían que desarrollar drivers específicos para cada tipo de hardware al que querían acceder. Cada software requería un driver distinto para cada hardware, implicando un esfuerzo enorme, al que hay que añadir el de las actualizaciones continuas. Debido a que solamente hay que adquirir un servidor OPC en lugar de diversos drivers para cada dispositivo de control que exista en la instalación, se disminuyen así los costos de inversión.
- Facilita la integración entre distintos fabricantes de PLC. Hay mayor flexibilidad en los sistemas de control debido que no hay la restricción de utilizar un solo fabricante y su protocolo de comunicación, ya que con el uso de la tecnología OPC se pueden tener dispositivos de distintos fabricantes conectados a una misma red, interactuando entre sí y manejando la misma información.
- Establece un menor grado de complejidad en las redes industriales ya que se reducen la cantidad de drivers requeridos para las interfaces de comunicación.

Parece evidente que el futuro inmediato de los sistemas de control debe pasar por la flexibilidad tanto del hardware como del software. Se podrán conectar entre sí aquellos componentes que más se adapten a las necesidades específicas de cada punto determinado



del proceso. La información será gestionada de manera que a cada usuario se le presenten los datos que necesite y de la forma en que los necesite.

Todo ello implica sistemas con gran capacidad de “apertura”. El hardware estará basado en instrumentos inteligentes, con capacidad de control PID incluido, buses estándar y módulos de Entradas y Salidas, como interfaz de campo.

## **Selección del Software OPC**

Entre todas las ventajas, opciones y bondades que ofrece la tecnología OPC. Se requiere seleccionar un software OPC que sea compatible con el PLC utilizar, que permita interconexión entre el servidor y el cliente, además de permitir innumerables accesos al cliente OPC, por lo que se recurre al procedimiento mostrado en el Apéndice I para seleccionar la mejor solución.

A continuación se presenta un breve resumen de algunos software disponibles en el mercado y que son posibles alternativas de solución.

### **S7-200 PC Access**

S7-200 PC Access es el servidor OPC para sistemas de destino S7-200. Permite la edición y visualización de datos del sistema de automatización SIMATIC S7-200 con aplicaciones Windows estándar, p.ej. Visual Basic, Visual C++ o Excel. S7-200 PC Access está certificado por la OPC-Foundation para el uso con todos los clientes OPC estándar.

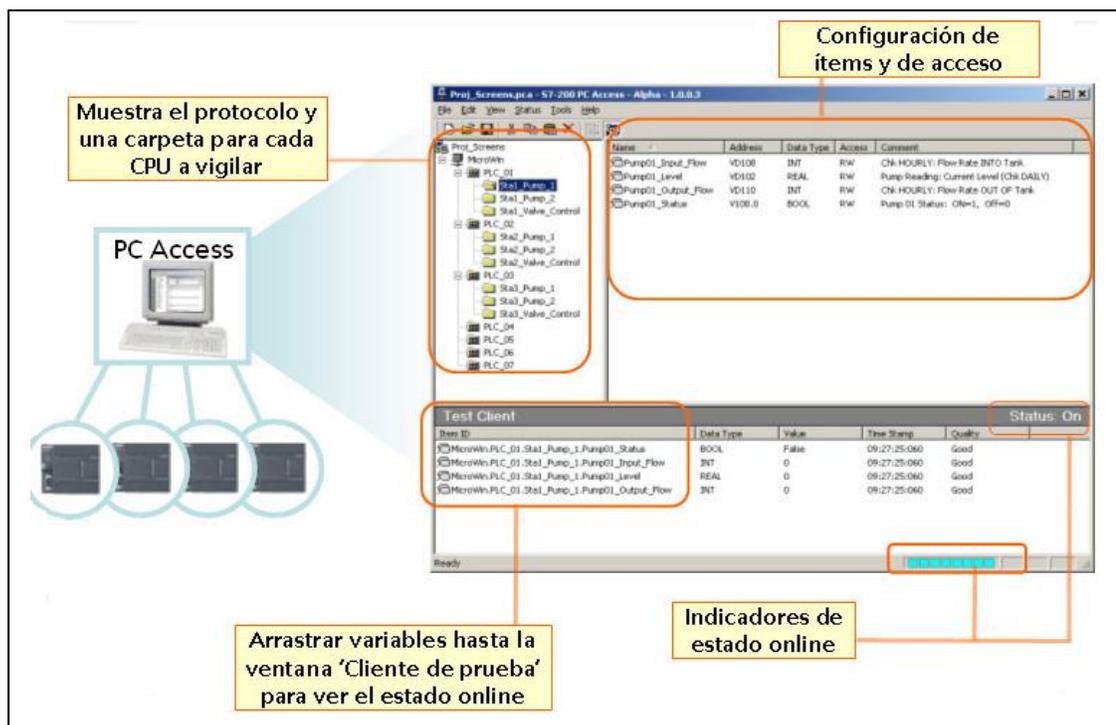


Figura 4.4 Ambiente de trabajo de PC Access. Fuente: PC Access

Se pueden crear, por ejemplo, aplicaciones de base de datos, interfaces de manejo y visualización (HMI), herramientas para evaluaciones estadísticas, p.ej. con Excel o módulos de cálculo para requisitos complejos S7-200 PC Access se ejecuta en PG/PC bajo Windows 2000/XP. Es posible conectar simultáneamente máximo 8 sistemas de destino S7-200 a través de conexiones online y observar mediante S7-200 PC Access. Todas las conexiones online pueden ser utilizadas en el PG/PC al mismo tiempo por STEP 7-Micro/Win. Para el intercambio de datos se pueden utilizar todos los protocolos S7-200.

### Ventajas de S7-200 PC Access

- \* Instalación rápida.
- \* Soporta OPC Data Access hasta versión 2.05.
- \* Cumple con los estándares OPC.
- \* Interopera con cualquier cliente OPC estándar.



- \* Completa ayuda en pantalla HTML.
  - \* Soporte de varios idiomas.
  - \* Soporta módems y Ethernet.
  - \* Incluye proyectos de ejemplo.
  - \* Soporta CPUs de Siemens.
  - \* Soporta el cable Smart.
  - \* Bajo costo.
  - \* Varios enlaces con CPUs.
  - \* Sin protección contra copia.
  - \* Cliente de prueba OPC integrado.
  - \* Importa símbolos de Micro/WIN.
  - \* Soporta los tipos de datos S7-200.
  - \* Permite a los autómatas pequeños:
    - Optimizar el rendimiento.
    - Mejorar el funcionamiento.
    - Mantener aplicaciones, máquinas e instalaciones.
- Estas mejoras se logran gracias al:
- Acceso fácil a los datos.
  - Sencillo proceso de control y observación.

### **Características del S7-200 PC Access**

- \* Cliente OPC integrado.
- \* Complemento de Excel para supervisar en hojas de cálculo.
- \* Marca de hora cada vez que se actualizan las variables (si se utiliza el cliente de prueba).
- \* Opciones avanzadas (p. ej. Límites máximo y mínimo).



### **ICONICS OPC Servers**

ICONICS OPC Servers es capaz de proveer comunicación directa con una larga lista de PLCs compatibles, dispositivo de Entradas y Salidas o Controlador, y está basado en la tecnología COM/DCOM. ICONICS OPC Servers viene completo con Explorador de Configuración OPC para su rápida configuración, monitoreo integrado en tiempo real de ítems OPC y una aplicación de cliente de prueba OPC.

### **KEPServerEX OPC Server**

El último servidor de Kepware, KEPServerEX, toma la tecnología OPC a nuevos niveles de desempeño mientras mantiene su facilidad de uso y calidad con la cual ya es conocido. Ofrece un amplio rango de conectividad y de protocolos de comunicación. Está certificado por la OPC Foundation para el estándar de comunicación Data Access hasta la versión 3.0.

Como se demuestra en el Apéndice I, el Servidor OPC que se adapta mejor a las necesidades del proyecto es el Servidor S7-200 PC Access, con el cual se realizará la interconexión entre el PLC y el computador.

## **4.3.- ARQUITECTURA DE CONEXIÓN**

Para establecer la conexión entre el PLC y la aplicación desarrollada en el computador se requiere de los siguientes componentes:

- PLC Siemens S7 200.
- Cable Convertidor PC/PPI, RS-485/RS-232.

- Computador:
  - Servidor OPC: es el software que estará comunicado directamente con el PLC para la adquisición y envío de datos.
  - Cliente OPC: es el software donde será desarrollada la aplicación gráfica.

La Figura 4.5 muestra un esquema de la arquitectura de conexión implementada.

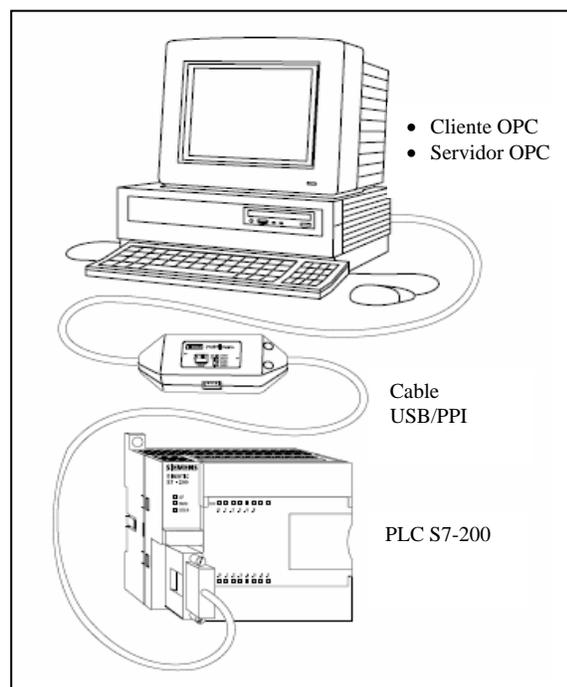


Figura 4.5 Arquitectura de conexión. Fuente: Manual Siemens S7 200

#### 4.3.1.- PLC Siemens S7 200.

A continuación se desarrollan las características del PLC y de los otros elementos que componen la arquitectura de conexión:

**Descripción del PLC:** El controlador lógico programable disponible en el Laboratorio de Automatización Industrial de la Escuela de Ingeniería Mecánica de la UC es un Siemens S-7 CPU 224. Este PLC incorpora en una carcasa compacta: un microprocesador, una fuente de alimentación integrada, así como circuitos de entrada

y de salida que conforman un potente Micro-PLC (ver Figura 4.5). A través del puerto de comunicación se carga el programa en el S7- CPU 224, hecho esto el mismo contendrá la lógica necesaria para observar y controlar los dispositivos, de acuerdo a las entradas y salidas respectivas, de la aplicación.

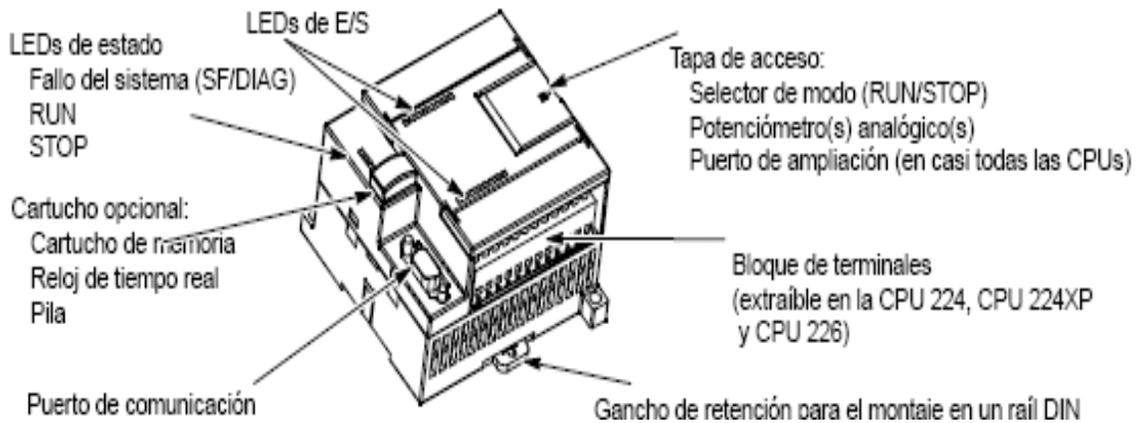


Figura 4.6 Identificación de las partes del Siemens S7-CPU224. Fuente: Manual Siemens S7 200

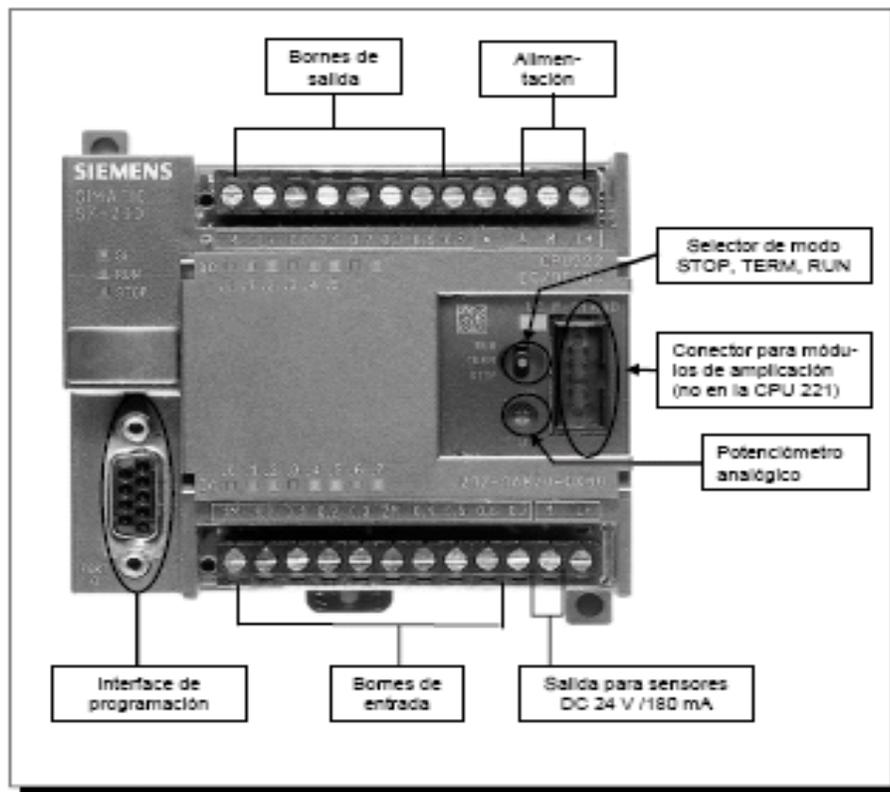


Figura 4.7 Identificación de las partes del Siemens S7-CPU224. Fuente: Manual Siemens



Desde la Tabla 4.2 a la Tabla 4.5 se presentan las especificaciones técnicas y características relevantes del PLC Siemens S7-CPU 224

**Tabla 4.2 Características del CPU 224.** Fuente: Manual Siemens S7 200

<i>Características del CPU</i>	
<b>Modelo de CPU</b>	<i>CPU 224</i>
<b>Dimensiones en mm (l x a x p)</b>	<i>120,5 x 80 x 62</i>
<b>Peso</b>	<i>360 g</i>
<b>Disipación</b>	<i>7 W</i>
<b>Tensión c.c. disponible +5 V c.c.</b>	<i>660 mA</i>
<b>Tensión c.c. disponible +24 V c.c.1</b>	<i>280 mA</i>
<b>Alimentación Nominal</b>	<i>120 a 240 V c.a</i>

**Tabla 4.3 Especificaciones de las Entradas y Salidas.** Fuente: Manual Siemens S7 200

<i>Entradas y Salidas E/S</i>	
<b>Entradas Digitales</b>	<i>14 x 24 V c.c</i>
<b>Salidas Digitales</b>	<i>10 Salidas de Relé</i>
<b>Puertos COM</b>	<i>1</i>
<b>Bloque de terminales extraíble</b>	<i>Sí</i>
<b>Salidas Analógicas</b>	<i>No</i>
<b>Entradas Analógicas</b>	<i>No</i>
<b>Nº máx. de módulos de ampliación</b>	<i>7</i>
<b>Entradas de captura de impulsos</b>	<i>14</i>
<b>Contadores rápidos</b>	<i>6 Contadores (total)</i>
<b>Fase simple</b>	<i>6 a 30 kHz</i>
<b>Dos fases</b>	<i>4 a 20 kHz</i>



Tabla 4.4 Especificaciones de la Memoria. Fuente: Manual Siemens S7 200

<i>Memoria</i>	
<b>Tamaño del programa de usuario (EEPROM) con edición en modo RUN</b>	<i>8192 bytes</i>
<b>sin edición en modo RUN</b>	<i>12288 bytes</i>
<b>Datos de usuario (EEPROM)</b>	<i>8192 bytes</i>
<b>Respaldo (condensador de alto rendimiento)</b>	<i>Típ. 100 h (mín. 70 h a 40° C)</i>
<b>(pila opcional)</b>	<i>Típ. 200 días</i>

Tabla 4.5 Especificaciones y características Generales. Fuente: Manual Siemens S7 200

<i>Datos Generales</i>	
<b>Temporizadores</b>	<i>256 temporizadores en total: 4 temporizadores de 1 ms, 16 temporizadores de 10 ms, y 236 temporizadores de 100 ms</i>
<b>Contadores</b>	<i>256 (respaldo por condensador de alto rendimiento o pila)</i>
<b>Marcas internas almacenadas al desconectar la CPU</b>	<i>256 (respaldo por condensador de alto rendimiento o pila) 112 (almacenamiento en EEPROM)</i>
<b>Velocidad de ejecución booleana</b>	<i>0,22 μs por operación</i>
<b>Puertos (potencia limitada)</b>	<i>1 puerto RS-485</i>
<b>Velocidades de transferencia</b>	<i>9,6 Kbit/s</i>
<b>PPI</b>	<i>19,2 Kbit/s</i>
<b>DP/T</b>	<i>187,5 Kbit/s</i>

Dimensiones para el montaje del PLC Siemens S7 sobre el riel DIN 50 022

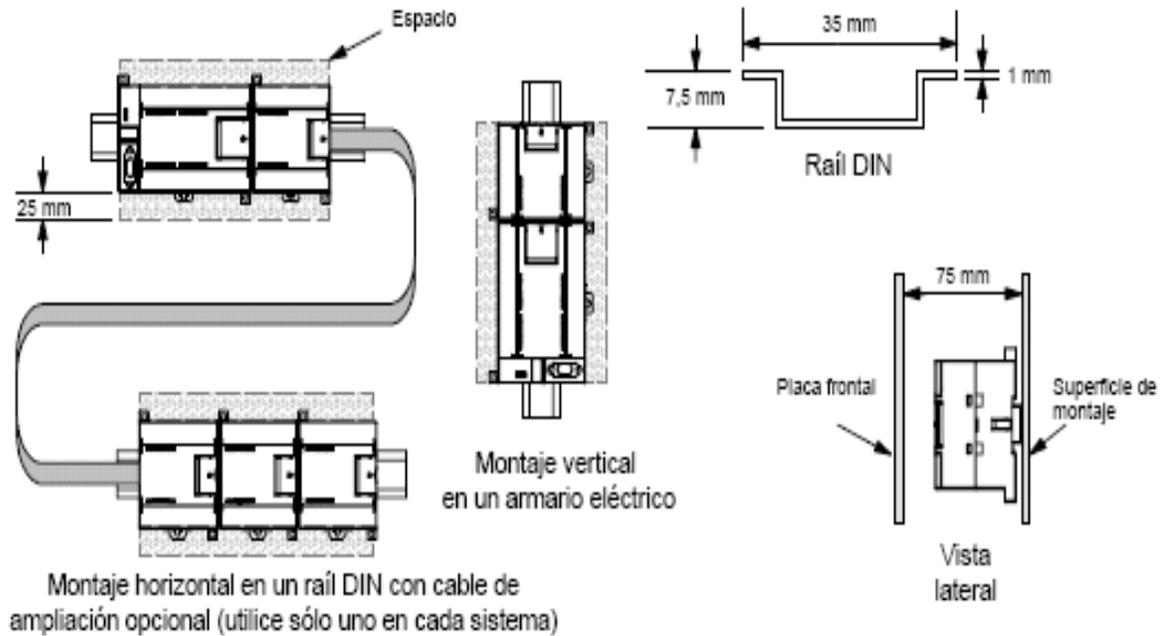


Figura 4.8 Especificaciones para el montaje. Fuente: Manual Siemens S7 200

Tabla 4.6 Especificaciones dimensionales del PLC S7. Fuente: Manual Siemens S7 200

Módulo S7-200	Ancho A	Ancho B
CPU 221 y CPU 222	90 mm	82 mm
CPU 224	120,5 mm	112,5 mm
CPU 224XP	140 mm	132 mm
CPU 226	196 mm	188 mm
Módulos de ampliación: 4 y 8 E/S DC y E/S de relé (8I, 4Q, 8Q, 4I/4Q) y salidas analógicas (2 AQ)	46 mm	38 mm
Módulos de ampliación: 16 E/S digitales (16I, 8I/8Q), E/S analógicas (4AI, 4AI/1AQ), RTD, Termopar, PROFIBUS, Ethernet, Internet, AS-Interface, 8 E/S AC (8I y 8Q), Posición y Módem	71,2 mm	63,2 mm
Módulos de ampliación: 32 E/S digitales (16I/16Q)	137,3 mm	129,3 mm

En las Figuras 4.9 y 4.10 se presentan los diagramas de cableado para el Siemens S7-CPU 224, tanto para corriente directa como para corriente continúa.

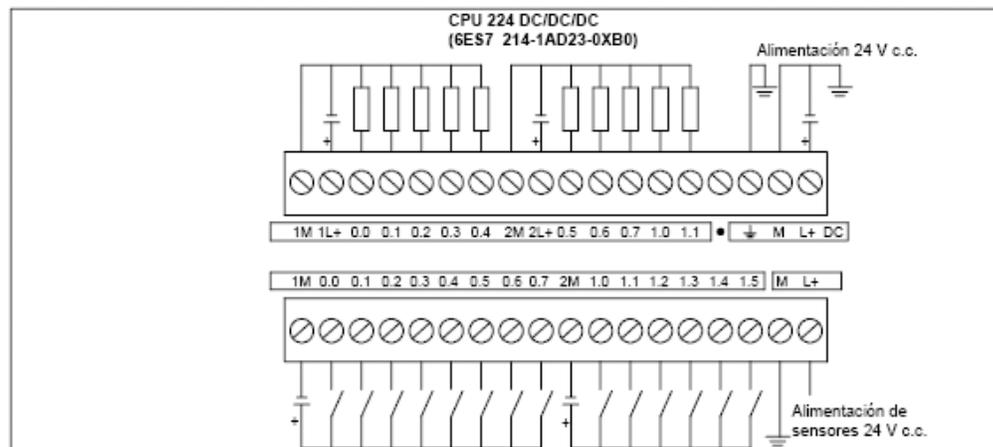


Figura 4.9 Diagramas de Cableado para corriente Directa. Fuente: Manual Siemens S7 200

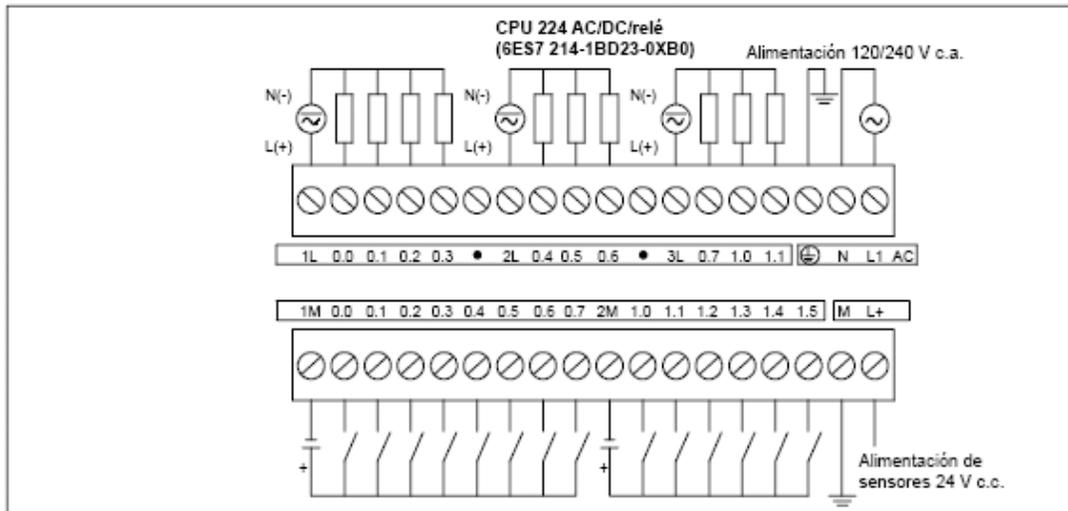


Figura 4.10 Diagramas de Cableado para corriente Continua. Fuente: Manual Siemens S7 200

#### 4.3.2.- Cable Convertidor RS-485/USB

Es el cable suministrado por el fabricante para la programación e intercambio de información entre el PLC y el computador. El puerto de comunicación del PLC es un RS-485 y el puerto de comunicación del computador es un puerto USB.

#### 4.3.3.- Computador

En el computador estará alojado tanto el servidor OPC como el cliente OPC.

**4.3.3.1.- Servidor OPC:** Es el software que estará comunicado directamente con el PLC para la adquisición y envío de datos. Se utilizará PC Access como servidor OPC.

**4.3.3.2.- Cliente OPC:** Es el software donde será desarrollada la aplicación gráfica. Se utilizará OPCSoft como cliente OPC.



## **4.4.- Desarrollo de la aplicación en el Computador**

### **4.4.1.- Selección del software que soportará la aplicación en el computador (cliente OPC)**

Primeramente debe seleccionarse el software que soportará la aplicación a ser desarrollada en el computador, que será controlada por el PLC. Esto se realizó bajo un procedimiento especificado en el Apéndice II.

Se requiere un software que permita desarrollar la aplicación en el PC y que muestre gráficamente la interacción entre el PLC y el PC. Además, debe facilitar la posibilidad de interconexión entre el PLC y el computador a través del Cliente OPC.

#### **Alternativas de Solución**

A continuación se presenta una breve descripción de algunos software disponibles en el mercado nacional y que son posibles alternativas de solución.

### **LabVIEW 8.2**

LabVIEW es un ambiente de desarrollo gráfico para control avanzado, monitoreo, análisis en tiempo real y mantenimiento predictivo. Este software soporta OLE para Control de Procesos (OPC) para intercambio de información entre dispositivos de automatización. Incluye compatibilidad con el OPC Data Access 3.0, adición reciente a la especificación OPC incrementando desempeño y confiabilidad para tener acceso de datos en tiempo real a partir del hardware y software del control de procesos.

LabVIEW permite de una forma sencilla la creación de aplicaciones de interfaz humano-máquina (HMI) para monitoreo remoto y control. Además,

proporciona cientos de objetos para desarrollar interfaces profesionales para el usuario: gráficas, tablas, perillas, termómetros, y más. Esto se logra arrastrando los objetos de la paleta de controles para su posterior configuración.

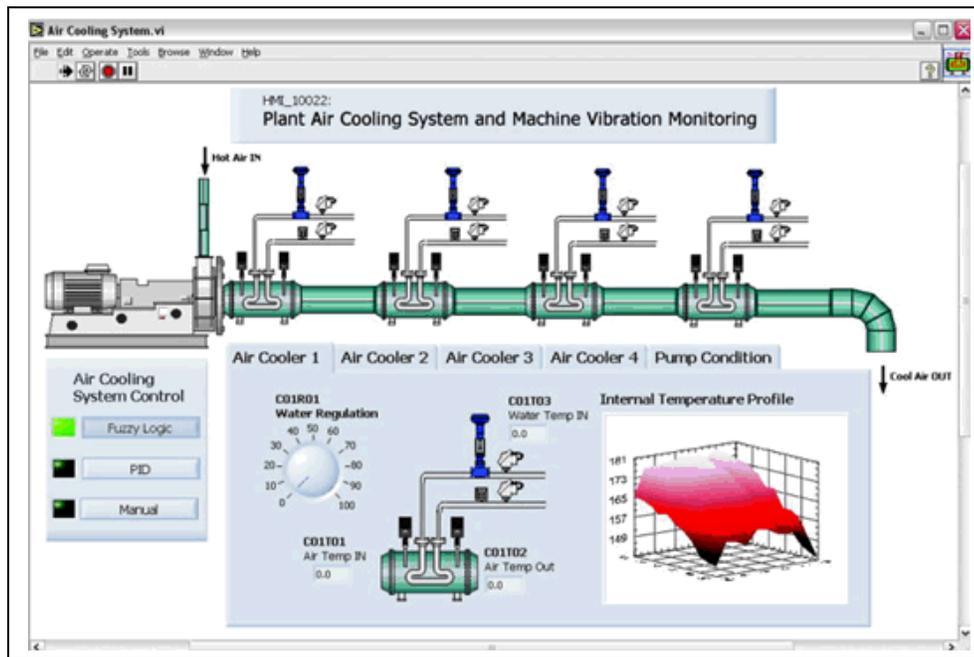


Figura 4.11 Control de Procesos por LabVIEW. Fuente: [www.ni.com](http://www.ni.com)

## OPC Soft

OPCSoft es una aplicación de software que aprovecha las características del estándar OPC para realizar el monitoreo y control de procesos industriales por computadora con interfaces amigables y una arquitectura abierta, capaz de crecer o adaptarse según las necesidades cambiantes de la instalación industrial. El usuario cuenta con un entorno visual donde puede insertar y manipular los instrumentos asociados a las variables proporcionadas por cualquier servidor OPC.

Adicionalmente, existe la posibilidad de modificar las diferentes características visuales y el comportamiento de los instrumentos para que se



adecuen al proceso e incluso se pueden implementar algoritmos de control personalizados por medio de guiones (scripts) ejecutados en intervalos de tiempo definidos por el usuario/operador. El software genera además, un registro histórico de las variables elegidas que puede ser recuperado en cualquier momento. Al fundamentarse en el estándar OPC, es posible agregar dispositivos de diferentes fabricantes a medida que estos son adquiridos e incorporados al proceso.

## **VisSim**

VisSim es una aplicación especialmente orientada a la simulación y el modelado de sistemas dinámicos complejos. El programa combina un entorno de diseño de modelos orientado a bloques para la definición del sistema del proceso a simular con un potente motor de simulación.

VisSim no es sólo un producto sino una familia de herramientas para el modelado y la simulación de sistemas de comunicación, sistemas de control y diseño de sistemas automatizados.

La interfaz visual de VisSim ofrece al usuario un método muy simple para desarrollar y construir, modificar y mantener modelos complejos; el motor de simulación proporciona rápidas y precisas soluciones para diseños de sistemas lineales, no lineales, sistemas continuos y discretos e híbridos. Si los requerimientos del diseño se extienden más allá de lo contemplado en la librería de bloques de VisSim, se pueden incluir bloques de usuario escritos en C, Fortran o Pascal. Para el diseño de sistemas de control, VisSim provee de un entorno totalmente integrado que permitirá realizar todas las tareas de simulación y modelado sin escribir una sola línea de código.

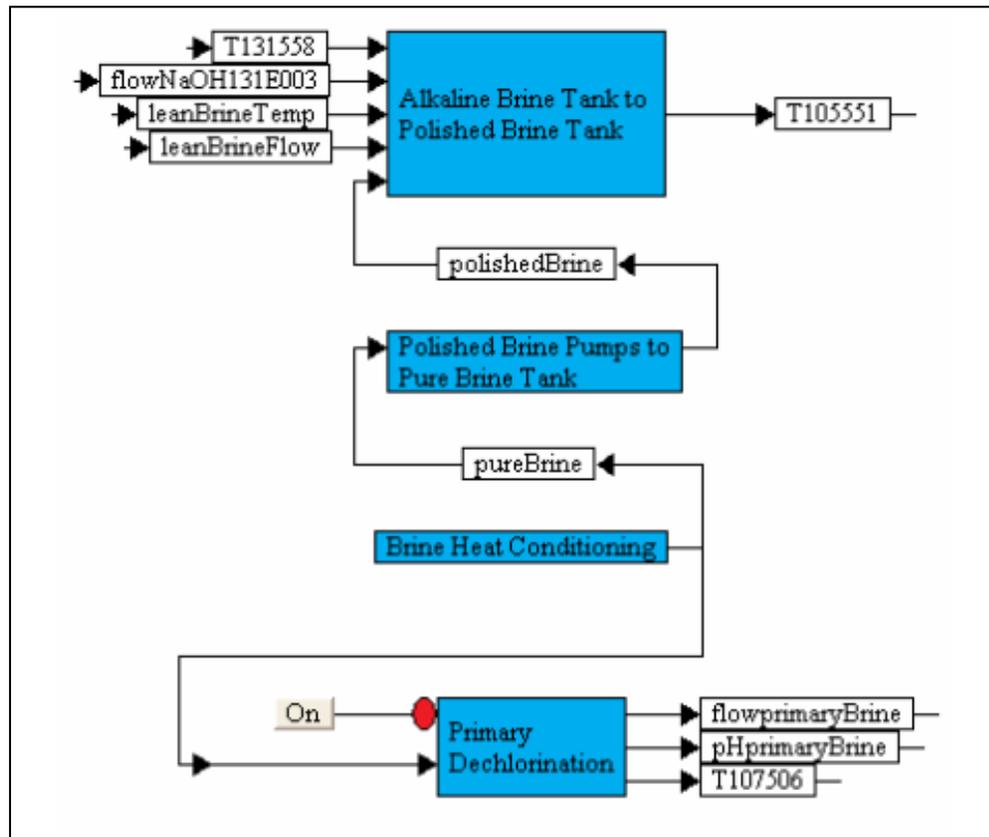


Figura 4.12 Diagrama de bloques con VisSim. Fuente: VisSim

## Excel

Es un programa de Microsoft para manejar hojas de cálculos. La hoja de cálculo está organizada en columnas y filas.

Para utilizar Excel como cliente OPC de Simatic OPC Server hay que emplear el mismo control *DatCon*, controlador de comunicaciones, que se usa en Visual Basic.

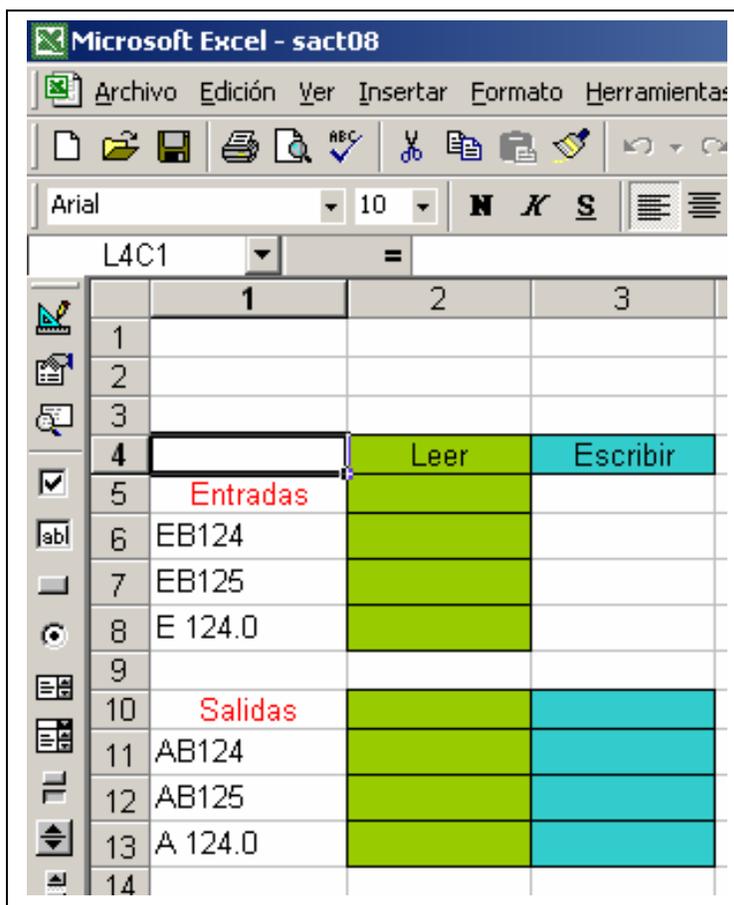


Figura 4.13 Excel como cliente OPC. Fuente: Microsoft Excel

Como se muestra en el Apéndice II el software de cliente OPC seleccionado bajo el cual será desarrollada la aplicación en el computador fue el OPCSoft.

#### 4.4.2.- Diseño de aplicaciones en el computador con base en el software seleccionado que serán controladas por el PLC.

##### 4.4.2.1.- Control de tráfico en una intersección de vías

La primera aplicación a diseñar tiene que ver con el control del tráfico de vehículos en una intersección de vías, el cual consta de 4 semáforos (ver Figura 4.14), que trabajarán de acuerdo a la secuencia mostrada en la Figura 4.15.

La Figura 4.14 muestra la intersección de vías y donde se observa que todos tienen sus tres luces de rojo, amarillo y verde y sólo dos de ellos tienen luz para cruzar.

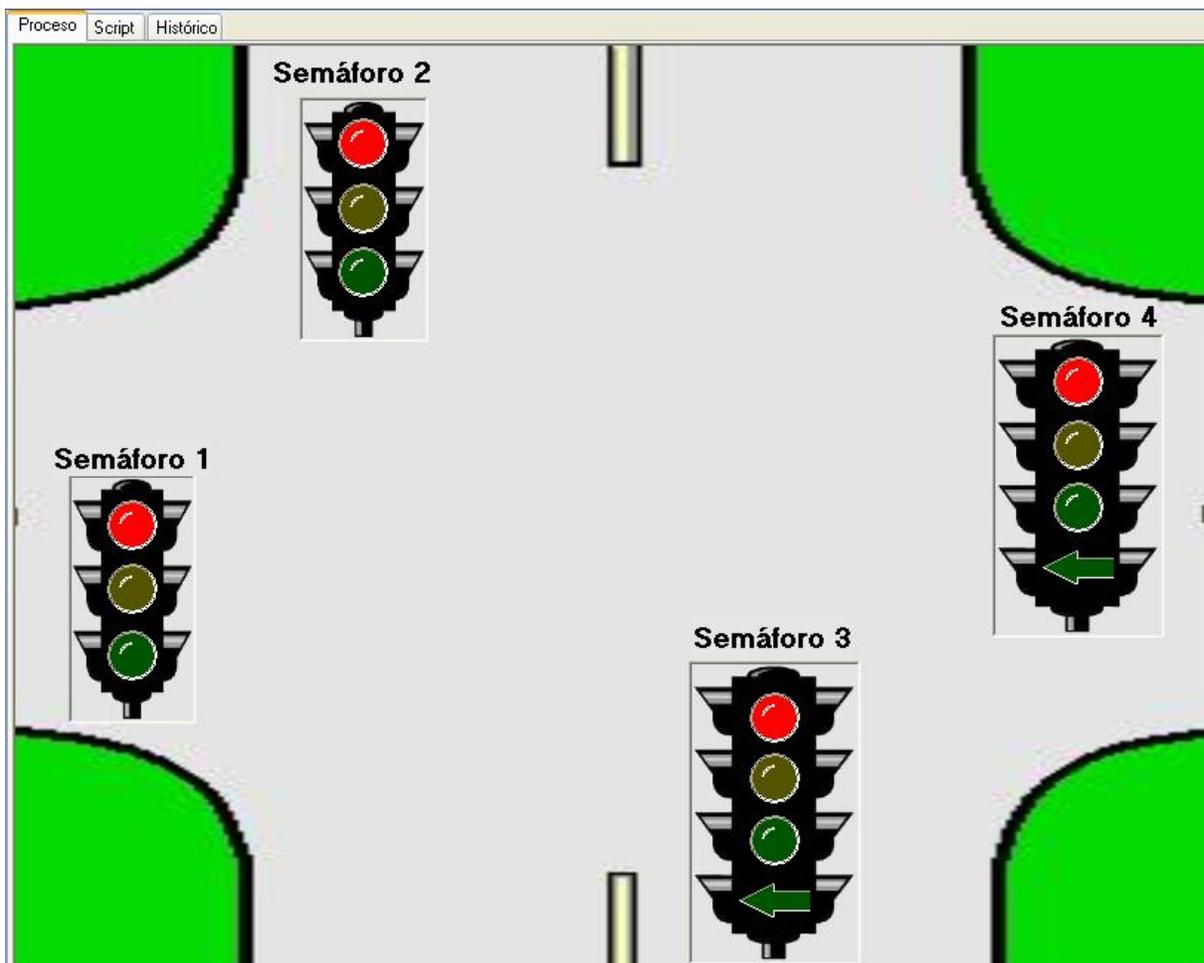


Figura 4.14. Fase Inicial de la aplicación desarrollada en el Computador



Sincronización entre los semáforos, pertenecientes a la intersección representada en la aplicación:

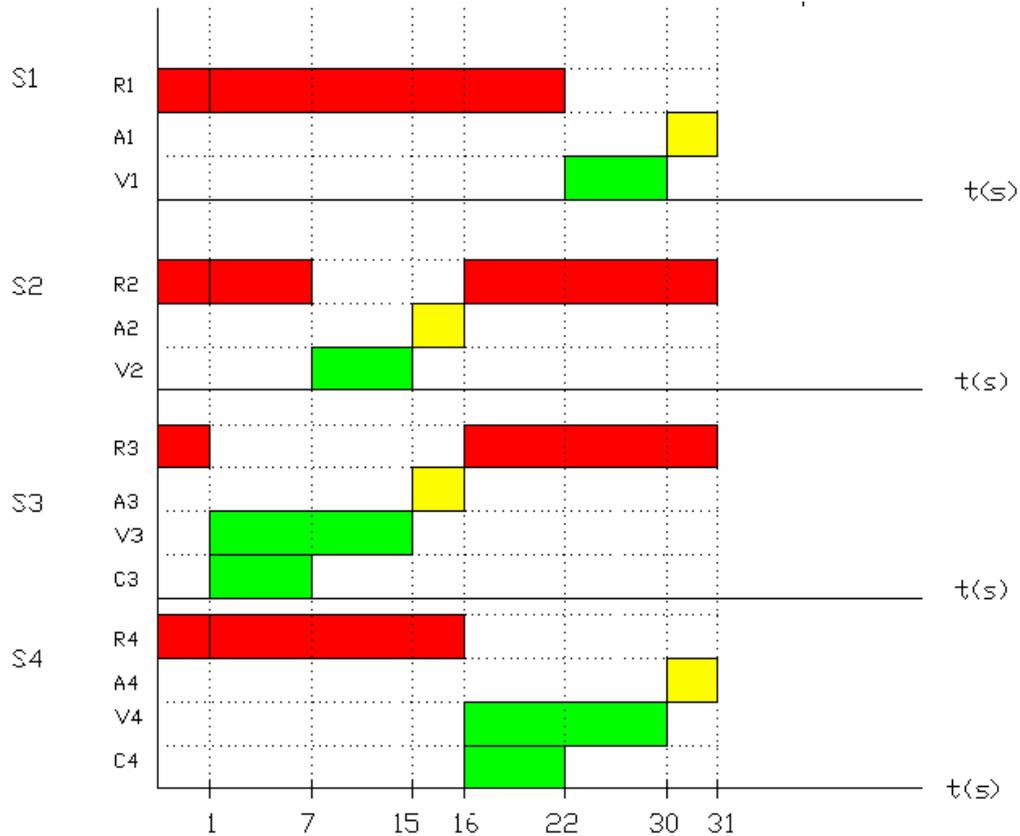


Figura 4.15. Sincronización de los Semáforos. Fuente: Elaboración Propia

Leyenda:

<b>S1:</b> Semáforo 1	<b>R1:</b> Rojo 1	<b>A1:</b> Amarillo 1	<b>V1:</b> Verde 1
<b>S2:</b> Semáforo 2	<b>R2:</b> Rojo 2	<b>A2:</b> Amarillo 2	<b>V2:</b> Verde 2
<b>S3:</b> Semáforo 3	<b>R3:</b> Rojo 3	<b>A3:</b> Amarillo 3	<b>V3:</b> Verde 3
<b>S4:</b> Semáforo 4	<b>R4:</b> Rojo 4	<b>A4:</b> Amarillo 4	<b>V4:</b> Verde 4

**C3:** Cruce 3

**C4:** Cruce 4

### Diagrama funcional

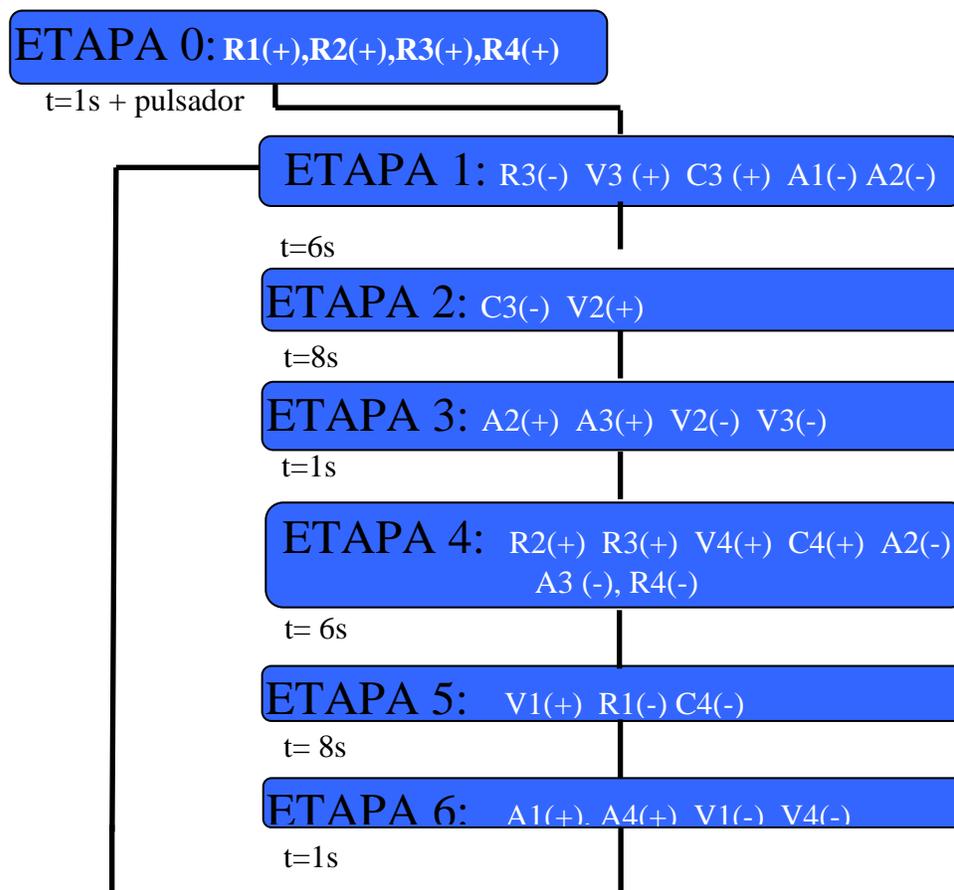


Figura 4.16. Diagrama funcional del semáforo. Fuente: Elaboración Propia

En cada una de las etapas se representan las acciones que realiza el indicador visual o lámpara en la etapa: (+) representa lámpara encendida (-) representa que la lámpara se apaga. Por ejemplo: al iniciar la etapa 1: se encienden las luces verde 3, la luz de cruce 3, y se apagan las luces amarillas 1 y 2. Además, se apaga la luz roja 3. El tiempo que tarda la luz de cruce 3, encendida, es de 6 segundos. Luego, se ejecuta la etapa 2, con las consideraciones correspondientes. Y de la misma forma transcurren el resto de las etapas del diagrama funcional.

Este diagrama funcional se implementa como un programa en diagrama escalera en el PLC, para que este dispositivo interactúe con la aplicación descrita más adelante. El diagrama escalera con el que se controla el semáforo se presenta en el anexo 1.

**Configuración del Servidor OPC:** Como ya se ha expuesto anteriormente fue utilizado el Servidor OPC S7-200 PC Access. A continuación se describen los pasos que se deben seguir dentro del programa para crear las variables:

1. Hacer clic en Nuevo, en la esquina superior izquierda, para crear un nuevo proyecto.
2. Hacer clic derecho sobre la opción MicroWin(COM1) y seleccionar la opción Nuevo PLC. Esto permitirá abrir la ventana de Propiedades del PLC donde se debe colocar el nombre que se le va a asignar al PLC y la dirección de la estación bajo la cual se encuentra configurado el PLC. Colocar el nombre “Siemens S7-200” asociado a la dirección de estación 2.

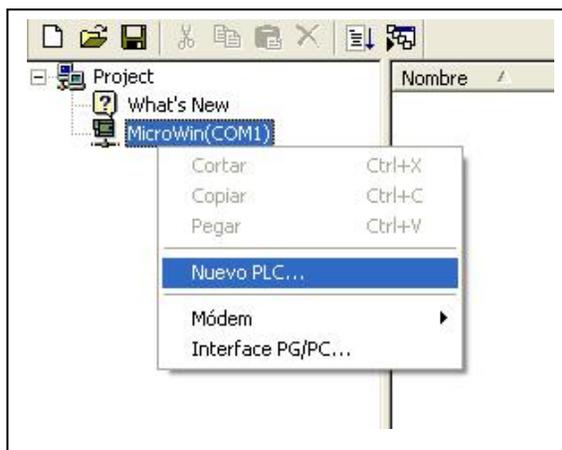


Figura 4.17 Agregar un nuevo PLC. Fuente: PC Access

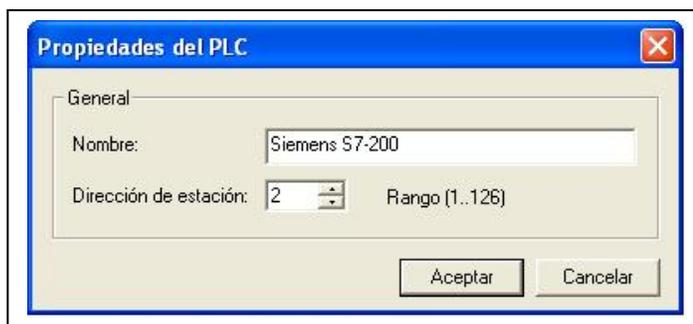


Figura 4.18 Propiedades del PLC. Fuente: PC Access

3. Hacer clic derecho sobre el PLC que se acaba de crear: S7-200 y seleccionar la opción Nuevo>Carpeta. Se crearán dos carpetas; una para las variables de entradas y otra para las variables de salida. Las variables de entrada serán los botones que darán inicio y fin a la aplicación mientras que las variables de salida serán las luces de cada uno de los semáforos.
4. Dentro de la carpeta de Salidas crear tres carpetas para identificar las luces rojas, amarillas y verdes. Las carpetas creadas se muestran en la Figura 4.19.



**Figura 4.19** Carpetas creadas en el servidor OPC. Fuente: PC Access

5. Crear las variables con las cuales va a trabajar el servidor OPC. Hacer clic derecho en la carpeta Entradas y seleccionar la opción Nuevo>Ítem. Cada ítem estará asociado a una variable de entrada del PLC. Aparecerá la ventana de Propiedades del ítem donde se permite colocar el nombre simbólico, dirección en la memoria, unidades de ingeniería y la descripción. En el nombre simbólico se le asigna el nombre a la variable con la cual estará identificada y aparecerá también la identidad (ID) con la cual estará asociada la variable dentro de cualquier cliente OPC. En la dirección en la memoria se asigna la dirección del PLC a la cual estará asociada la variable; si la variable es de Lectura/escritura, Lectura solamente o Escritura solamente y el tipo de datos de la variable. También se le debe asignar el tipo de datos que corresponde a dicha variable. La primera variable de entrada tendrá por nombre “Comenzar” y será la variable que estará asociada al inicio del ciclo del semáforo. Esta variable estará asociada a la dirección M1.4 dentro del

PLC. Todas las variables de entrada y salida serán de Lectura/Escritura. Asignarle el tipo de datos booleano (BOOL). En la Figura 4.20 se muestra la ventana de propiedades de la variable.

Propiedades del ítem

Nombre simbólico:

Nombre: Comenzar

ID: MicroWin.Siemens S7-200.Entradas.Comenzar

Dirección en la memoria

Dirección: M1.4 Read/Write

Tipo de datos: BOOL

Unidades de ingeniería

Máxima: 0.0000000

Mínima: 0.0000000

Descripción

Comentario:

Aceptar Cancelar

Figura 4.20 Propiedades del ítem (entradas). Fuente: PC Access

6. Se repite el procedimiento para las variables de salida. Hacer clic derecho en la carpeta Luces Rojas y seleccionar la opción Nuevo>Ítem. Cada ítem estará asociado a una variable de salida del PLC. Aparecerá la ventana de Propiedades del ítem donde se le dará nombre simbólico a la variable y la dirección en la memoria a la cual estará asociada. A esta variable colocarle por nombre R1, estará asociada a la dirección Q0.0 en el PLC y será una variable de tipo booleana (BOOL), como se muestra en la Figura 4.21.

**Figura 4.21** Propiedades del ítem (salidas). Fuente: PC Access

7. Se deben repetir los pasos 6 al 8 para todas las variables de entrada que se van a utilizar, asignándole un nombre y una dirección en el PLC diferente a cada de las variables. Todas las variables de entrada serán de tipo booleana. En la Tabla 4.1 se muestran todas las variables de entrada que se emplearán para el proyecto del semáforo.

**Tabla 4.7** Variables de entrada. Fuente: Elaboración propia

<i>Nombre</i>	<i>Dirección</i>	<i>Tipo de datos</i>
<b>Comenzar</b>	M1.4	<i>Booleana</i>
<b>Finalizar</b>	M1.3	<i>Booleana</i>



8. Se deben repetir los pasos 7 y 8 para todas las variables de entrada que se van a utilizar haciendo clic derecho en cada una de las carpetas (Luces Rojas, Luces Amarillas y Luces Verdes) dependiendo de la luz con la que se estará trabajando, asignándole un nombre y una dirección en el PLC diferente a cada variable. Todas las variables de salida serán de tipo booleana. En la Tabla 4.2 se muestran todas las variables de salida que se emplearán para el proyecto del semáforo. En la Figura 4.22 se muestran todas las variables de entrada como aparecen en el servidor OPC y en la Figura 4.23 se muestran todas las variables de salida como aparecen en el servidor OPC.

**Tabla 4.8 Variables de salida.** Fuente: Elaboración propia

<i>Carpeta</i>	<i>Nombre</i>	<i>Dirección</i>	<i>Tipo de datos</i>
Luces Rojas	R1	Q0.0	Booleana
	R2	Q0.1	Booleana
	R2	Q0.2	Booleana
	R4	Q0.3	Booleana
Luces Amarillas	A1	Q0.4	Booleana
	A2	Q0.5	Booleana
	A3	Q0.6	Booleana
	A4	Q0.7	Booleana
Luces Verdes	V1	Q1.0	Booleana
	V2	Q1.1	Booleana
	V3	M0.7	Booleana
	V4	M1.0	Booleana
	C3	M1.1	Booleana
	C4	M1.2	Booleana



Figura 4.22 Variables de entrada. Fuente: PC Access



Figura 4.23 Variables de salida (Luces Verdes). Fuente: PC Access

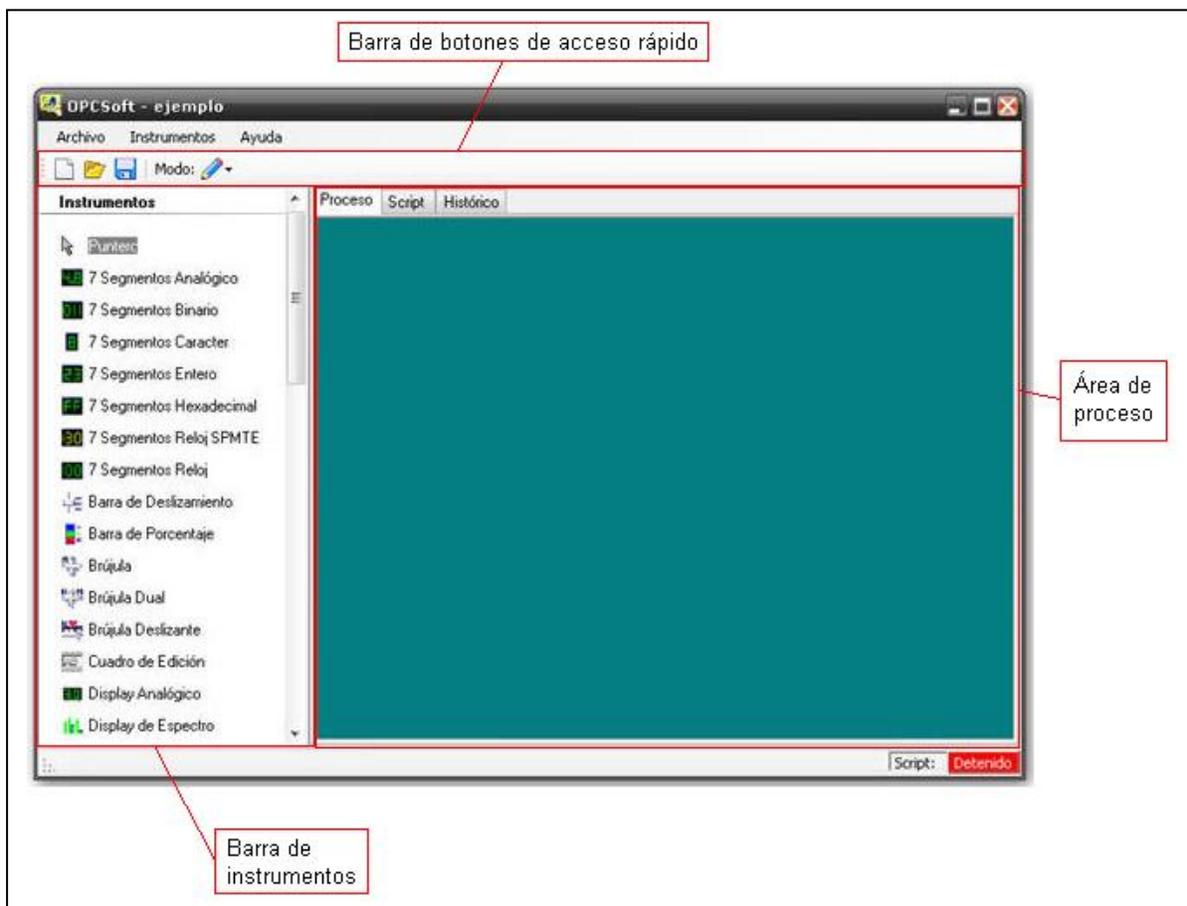
### Trabajando con OPC Soft

Como ya se expuso en el punto 4.4 el software seleccionado con el cual será desarrollada la aplicación en el computador fue OPC Soft.

### Ventana Principal

En la Figura 4.24 se muestra la ventana principal de OPC Soft. Los menús que se encuentran disponibles son: Archivo y Ayuda y, como botones de acceso rápido

están disponibles: Nuevo y Abrir. Una vez que el usuario ha seleccionado entre las opciones de crear un proyecto Nuevo o Abrir uno existente, la ventana principal se muestra como la siguiente imagen:



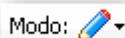
**Figura 4.24** Ventana principal de OPC Soft. Fuente: OPCSoft

La aplicación puede estar en dos (2) modos: Edición y Ejecución. Para cambiar entre ellas bastará con ubicar el cursor del ratón sobre el botón identificado con el texto “Modo:” en la barra de botones de acceso rápido y presionar el botón izquierdo del dispositivo manual, lo que desplegará una lista desde la cual el usuario podrá establecer el modo de la aplicación.

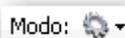
La imagen asociada al botón variará de acuerdo al modo en que se encuentre.



Si está en “Modo: Edición”:



y si está en “Modo: Ejecución”:



### Aplicación por medio de la cual será comprobada la interconexión entre el PLC y el PC

Para crear la simulación del semáforo en OPC Soft se seguirán los siguientes pasos.

1. Hacer clic en Nuevo, en la esquina superior izquierda, para crear un nuevo proyecto. Inmediatamente se le debe asignar un nombre al proyecto, colocarle Simulación del Semáforo.
2. En la barra de instrumentos que aparece en el lado izquierdo de la pantalla hacer clic en el Interruptor con Led y arrastrarlo hasta el área de proceso. Una vez que se ha seleccionado un elemento, el cursor del ratón dentro del área de proceso cambia indicando el instrumento que se encuentra latente a ser agregado. Luego aparecerá el pulsador con led, como se muestra en la Figura 4.25. Se trabajará con interruptores para las variables de entrada y luces circulares para las variables de salida, excepto las luces para cruzar donde se trabajará con luces tipo flecha.

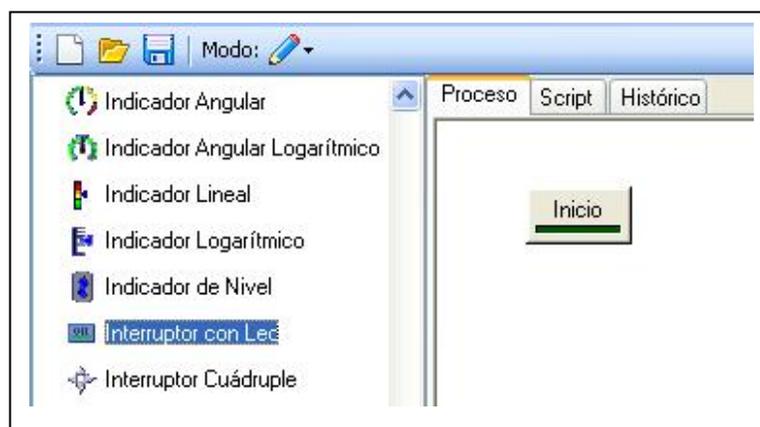


Figura 4.25 Interruptor con Led en OPC Soft. Fuente: OPCSoft

3. OPC Soft le asigna un nombre por defecto al instrumento. Este nombre se puede cambiar haciendo clic derecho en el instrumento y seleccionando la opción Cambiar Nombre. Al primer pulsador colocarle el nombre Comenzar.



**Figura 4.26** Cambiar el nombre del pulsador. Fuente: OPCSoft

4. Configurar el pulsador para asociarlo al Servidor OPC. Hacer clic derecho nuevamente en el pulsador y seleccionar la opción Propiedades. Aparecerá una ventana similar a la de la Figura 4.27. Cambiar el texto para colocarle Inicio.

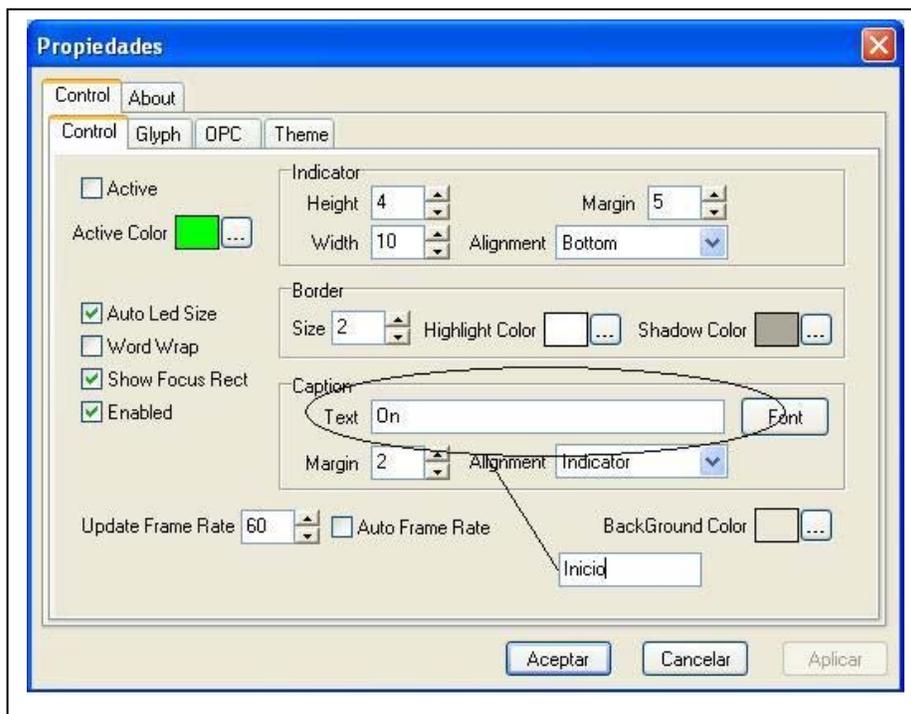


Figura 4.27 Ventana de propiedades del pulsador. Fuente: OPCSoft

5. Para asociar el pulsador a la variable Comenzar del Servidor OPC hay que seleccionar la pestaña OPC y hacer clic en el botón Add. Aparecerá una ventana con la lista de propiedades del pulsador. Seleccionar la propiedad que el Servidor OPC deberá modificar, en este caso será la propiedad Active para que el Servidor OPC actúe directamente sobre la activación y desactivación del pulsador.

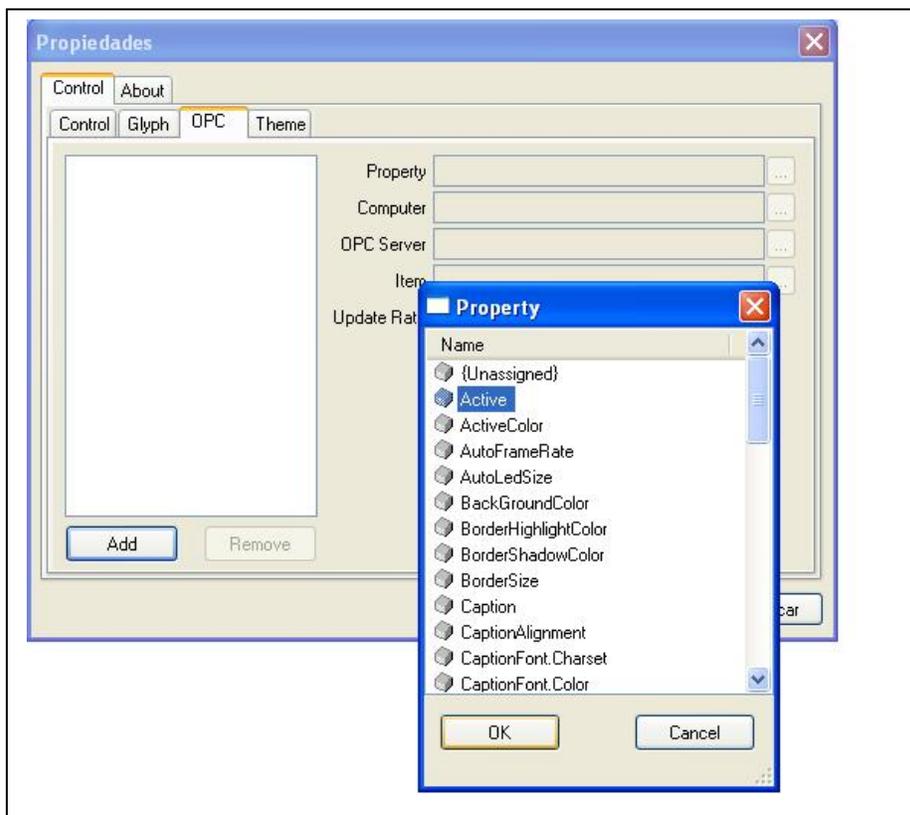


Figura 4.28 Selección de la propiedad del pulsador. Fuente: OPCSoft

6. Hacer clic en el botón de OPC Server para seleccionar el OPC. Aparecerá la ventana con todos los Servidores OPC instalados en el computador.

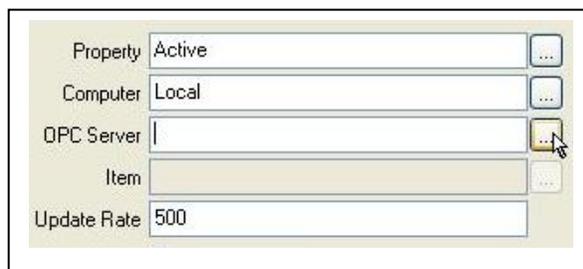


Figura 4.29 Selección del OPC. Fuente: OPCSoft



Figura 4.30 Selección del Servidor OPC S7200.OPCServer. Fuente: OPCSoft

7. Luego hacer clic en el botón de Ítem y seleccionar en el PLC S7-200, dentro de la carpeta de Entradas, la variable Comenzar. En la parte inferior de la ventana (como se muestra en la Figura 4.31) aparecerá la identidad (ID) de la variable que se está seleccionando. Una vez seleccionada la variable aparecerá nuevamente la identidad de la variable en la ventana de propiedades, como se muestra en la Figura 4.32

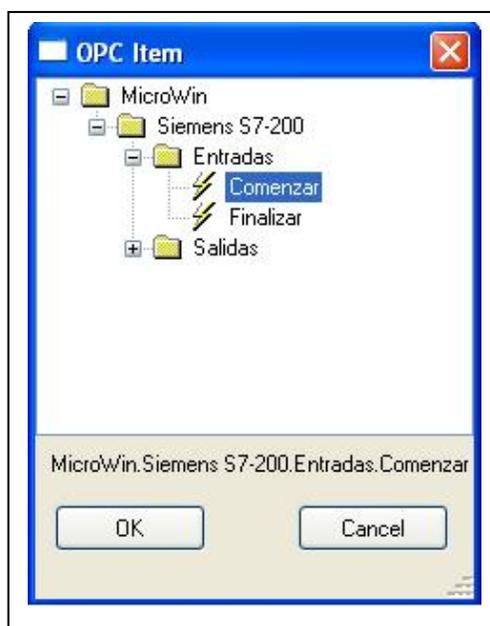


Figura 4.31 Selección del Ítem. Fuente: OPCSoft

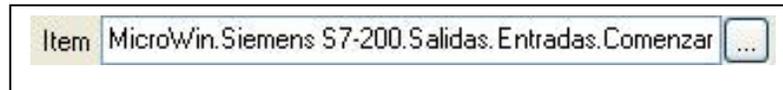


Figura 4.32 Identidad del Ítem. Fuente: OPCSoft

8.- En la ventana Script se colocará una línea de programación en lenguaje Visual Basic para hacer que el interruptor sea monoestable. La línea de programación se repetirá cada 400 milisegundos, y será como sigue a continuación:

```
If Comenzar.Active = True Then  
    Comenzar.Active = False  
End If
```

9. Se repiten los pasos 2 al 7 para el pulsador de Finalizar.

10. En la barra de instrumentos que aparece en el lado izquierdo de la pantalla hacer clic en el Led Tipo Círculo y arrastrarlo hasta el área de proceso. Utilizar Led Tipo Círculo para las variables de salida que son las luces rojas, amarillas y verdes. Para las luces de cruce se utilizarán Led Tipo Flecha.

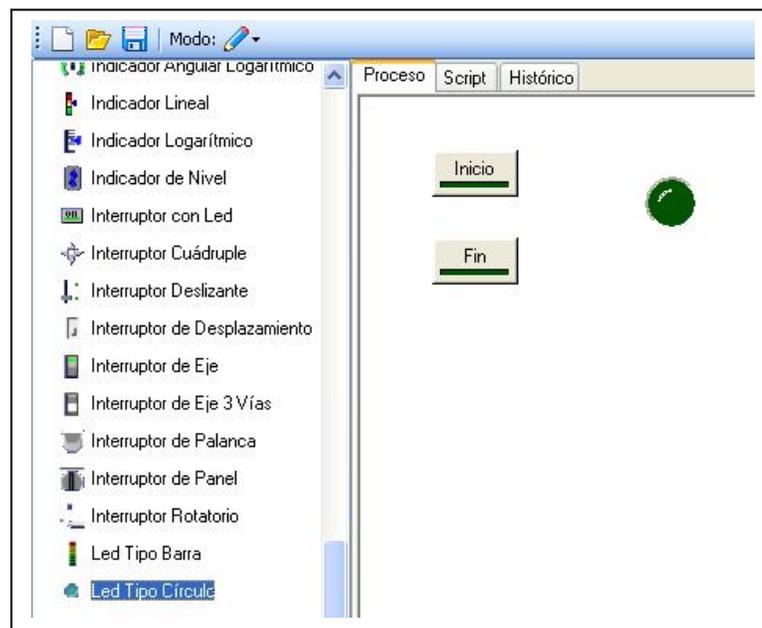


Figura 4.33 Interruptor con Led en OPC Soft. Fuente: OPCSoft

11. OPC Soft le asigna un nombre por defecto al LED. Este nombre se puede cambiar haciendo clic derecho en el instrumento y seleccionando la opción Cambiar Nombre. Colocarle al primer pulsador el nombre SemáforoRojo1.
12. Para configurar el pulsador para asociarlo al Servidor OPC hay que hacer clic derecho nuevamente en el pulsador y seleccionar la opción Propiedades. Aparecerá una ventana similar a la de la Figura 4.34. El color predeterminado del led es verde. Cambiar el color de activado (Active Color) y desactivado (Inactive Color) a rojo brillante y rojo opaco, respectivamente. De igual forma habrá que cambiar el color de las luces amarillas cuando vayan a ser creadas.

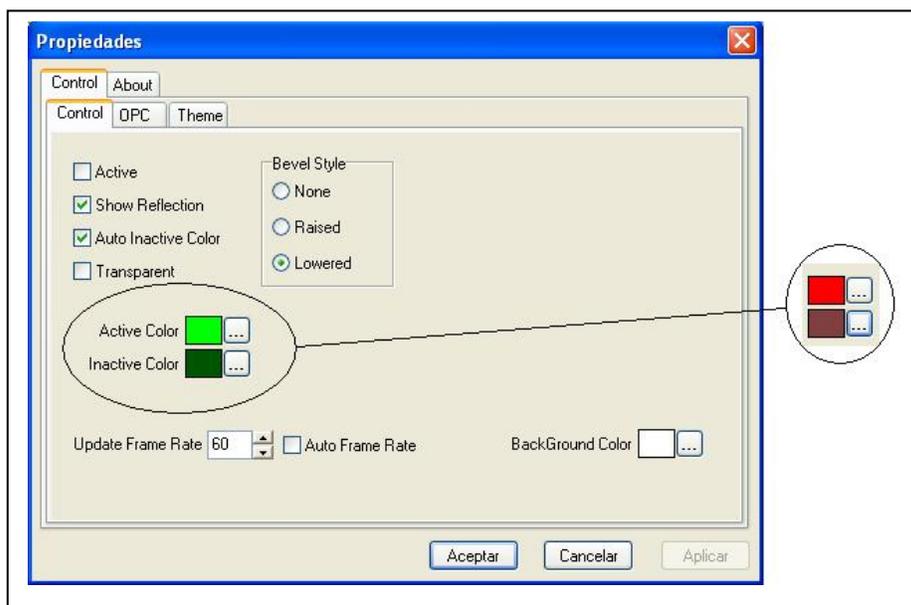


Figura 4.34 Ventana de propiedades del Led en OPC Soft. Fuente: OPCSoft

13. Para asociar el pulsador a la variable Comenzar del Servidor OPC hay que seleccionar la pestaña OPC y hacer clic en el botón Add. Aparecerá una ventana

con la lista de propiedades del pulsador. Seleccionar la propiedad Active para que el Servidor OPC actúe directamente sobre el encendido y apagado de la luz.

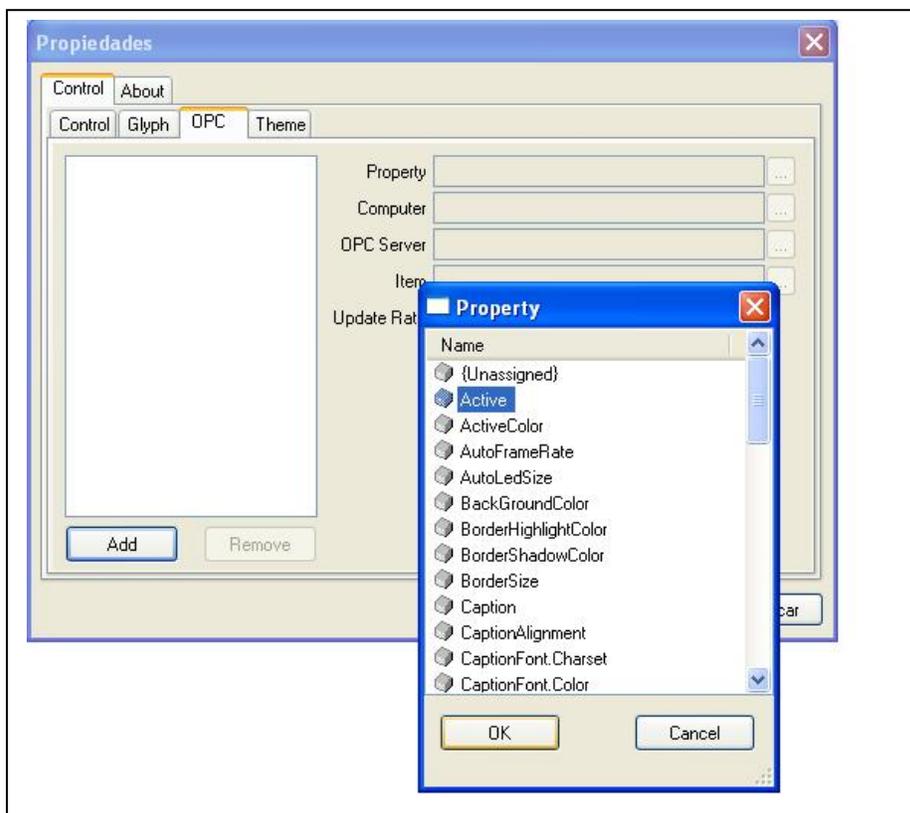


Figura 4.35 Selección de la propiedad del led. Fuente: OPCSoft

14. Luego hacer clic en el botón de OPC Server para seleccionar el OPC. Aparecerá la ventana con todos los Servidores OPC instalados en el computador.

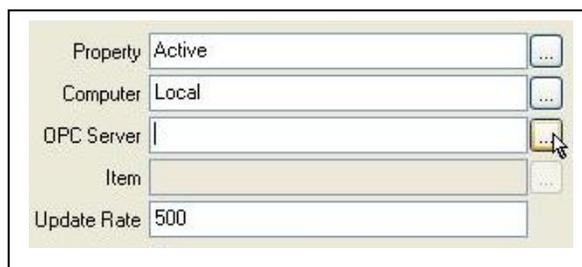


Figura 4.36 Selección del OPC. Fuente: OPCSoft



Figura 4.37 Servidor OPC S7200.OPCServer. Fuente: OPCSoft

15. Luego hacer clic en el botón de Ítem y seleccionar en el PLC S7-200, dentro de la carpeta de Salidas, la variable R1. En el cuadro de texto de Ítem aparecerá la identidad (ID) de la variable.

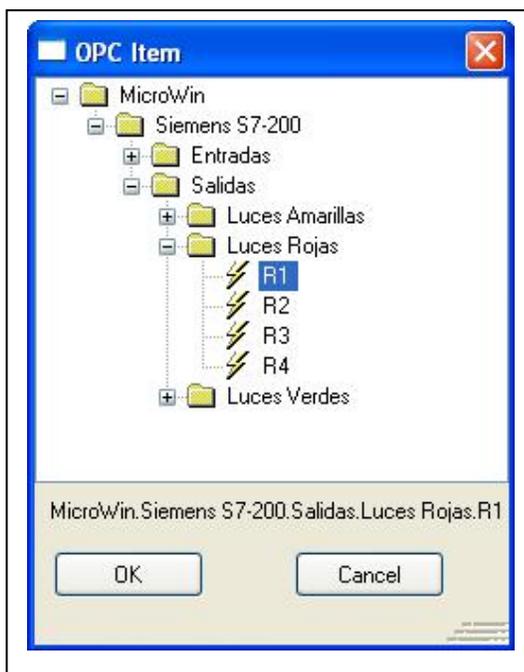


Figura 4.38 Selección del Ítem. Fuente: OPCSoft

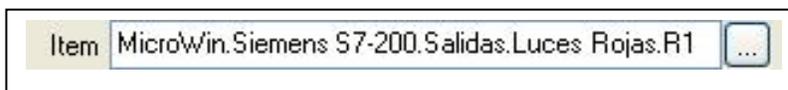


Figura 4.39 Identidad del Ítem. Fuente: OPCSoft

16. Se repiten los pasos 10 al 15 para las demás luces.
17. Al terminar de colocar todas las luces el área de proceso debe quedar como se muestra en la Figura 4.40

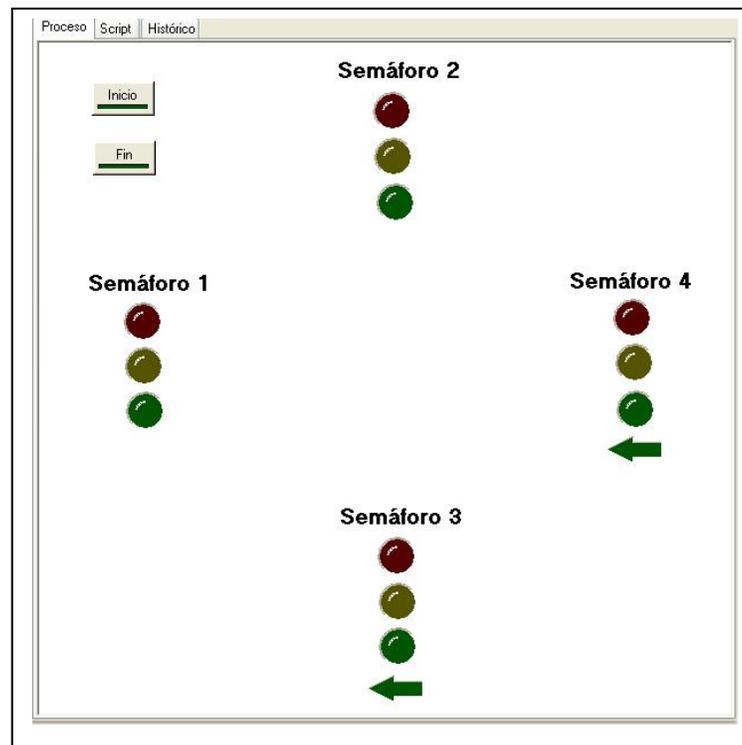


Figura 4.40 Área de proceso con los semáforos. Fuente: OPCSoft

#### 4.4.2.2.- Control del llenado y vaciado de un tanque

A continuación se presenta una segunda aplicación para demostrar la factibilidad del OPC en el control de procesos. La simulación consiste en el llenado de un tanque el cual será realizado mediante una bomba que se enciende y se apaga con el interruptor “Bomba” (ver Figura 4.48).

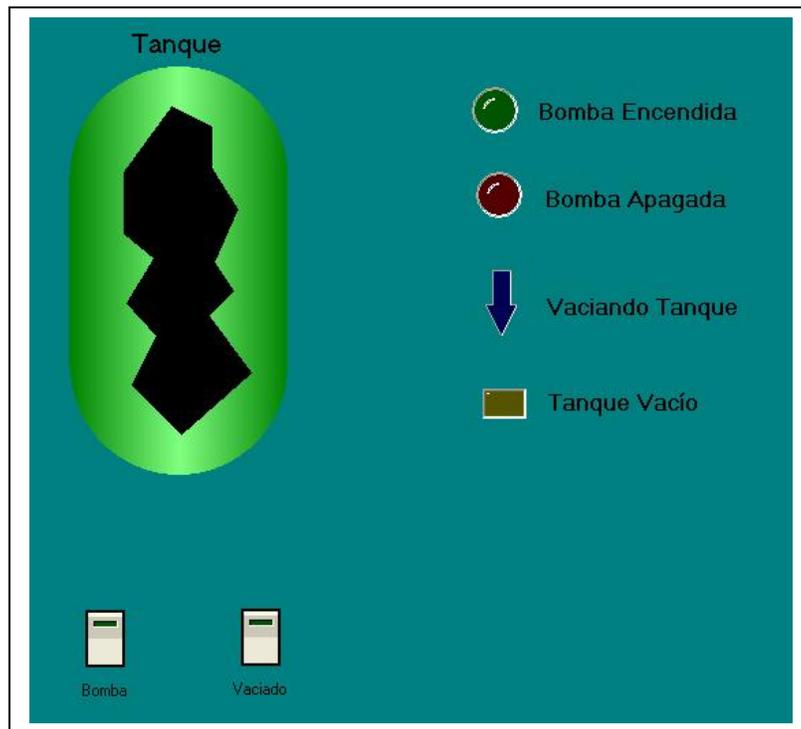


Figura 4.48 Ejemplo de llenado de un tanque. Fuente: Elaboración Propia

Tanto el interruptor “Bomba” como el interruptor “Vaciado” serán biestables y son para encender la bomba que llenará el tanque y para vaciar el tanque, respectivamente.

Todos los elementos de este ejemplo se configuran mediante el mismo procedimiento de los interruptores y luces expuestos en el ejemplo del semáforo. La variante en este ejemplo está en el Tanque. En la tabla 4.9 se presentan las variables necesarias para incluir en este ejemplo.



Tabla 4.9 Variables para la aplicación del Tanque. Fuente: Elaboración propia

<i>Carpeta</i>	<i>Nombre</i>	<i>Dirección</i>	<i>Tipo de datos</i>
Interruptores	Bomba	M0.0	Booleana
	Vaciado	M0.1	Booleana
Luces	Bomba Encendida	Q0.0	Booleana
	Bomba Apagada	Q0.1	Booleana
	Vaciado de Tanque	Q0.2	Booleana
	Tanque Lleno	M0.2	Booleana

### Configuración del Tanque en el cliente OPCSoft:

Para simular el llenado del tanque se utilizará una propiedad del objeto que se denomina Position y refleja el nivel en el cual se encuentra el tanque en un determinado momento.

Se agregará una línea de programación en la ventana Script, utilizando el ciclo If y que se repetirá cada 500 milisegundos, para que comience a llenar el tanque, de manera constante, una vez que detecte que la luz de Bomba Encendida esté activa. La línea de programación en VBScript será como sigue:

```
If Led_BombaE.Active Then  
    TANQUE.Position = TANQUE.Position + 1  
End If
```

Lo mismo se hará cuando se quiera vaciar el tanque. En este caso se tienen que cumplir dos condiciones: (1) Que la luz de Bomba apagada esté activa, (2) que la luz de vaciando tanque esté activa. La línea de programación en VBScript será como sigue:

```
If Led_VaciandoTanque.Active And Led_BombaA.Active Then
```



TANQUE.Position = TANQUE.Position - 1

End If

A continuación se muestran las demás líneas de programación que se colocaron adicionalmente con su función: Todas las condiciones que estén dentro de la ventana Script se repetirán cada 500 milisegundos.

- \* Cuando detecta que el tanque está lleno, desactiva el pulsador que enciende la bomba.

If TANQUE.Position = TANQUE.PositionMax Then

Bomba.Active = False

End If

- \* Cuando detecta que el tanque está vacío, desactiva el pulsador de vaciado de tanque.

If TANQUE.Position = TANQUE.PositionMin Then

Vaciado.Active = False

End If

- \* Cuando se detecta que el tanque está lleno, se enciende una luz que indica esta condición y que además da una señal al PLC para que impida que se encienda la bomba nuevamente. De no estar lleno el tanque completamente la luz permanecerá apagada.

If TANQUE.Position = TANQUE.PositionMax Then

Led\_TanqueLleno.Active = True

Else Led\_TanqueLleno.Active = False

End If



- \* Una etiqueta indicará “Tanque Lleno” cuando el tanque llegue a su máximo nivel, de lo contrario indicará “Tanque Parcialmente Lleno”

```
If TANQUE.Position = TANQUE.PositionMax Then  
    ETIQUETA6.Caption = "Tanque Lleno"  
Else ETIQUETA6.Caption = "Tanque Parcialmente Lleno"  
End If
```

- \* La misma etiqueta expuesta anteriormente indicará “Tanque Vacío” cuando el tanque llegue a su nivel mínimo.

```
If TANQUE.Position = TANQUE.PositionMin Then  
    ETIQUETA6.Caption = "Tanque Vacío"  
End If
```

Las acciones de control del llenado y vaciado del tanque, se efectuarán a través de un programa en diagrama escalera que estará alojado en el PLC, para que este dispositivo interactúe con la aplicación descrita anteriormente. El diagrama escalera con el que se controla el vaciado y llenado del tanque se presenta en el anexo 2.

#### **4.5.- Implementación del sistema de comunicación entre el controlador lógico programable y el computador.**

Una vez desarrollada la aplicación en el software OPC SOFT y habiendo configurado el servidor OPC con el software S7-200 PC Access. Se implementó la simulación en el laboratorio de automatización en la escuela de Ingeniería Mecánica de la Universidad de Carabobo.

### 4.5.1.- Control de tráfico en una intersección de vías

Las imágenes que se presentan a continuación muestran cada una de las etapas descritas en el punto 4.5

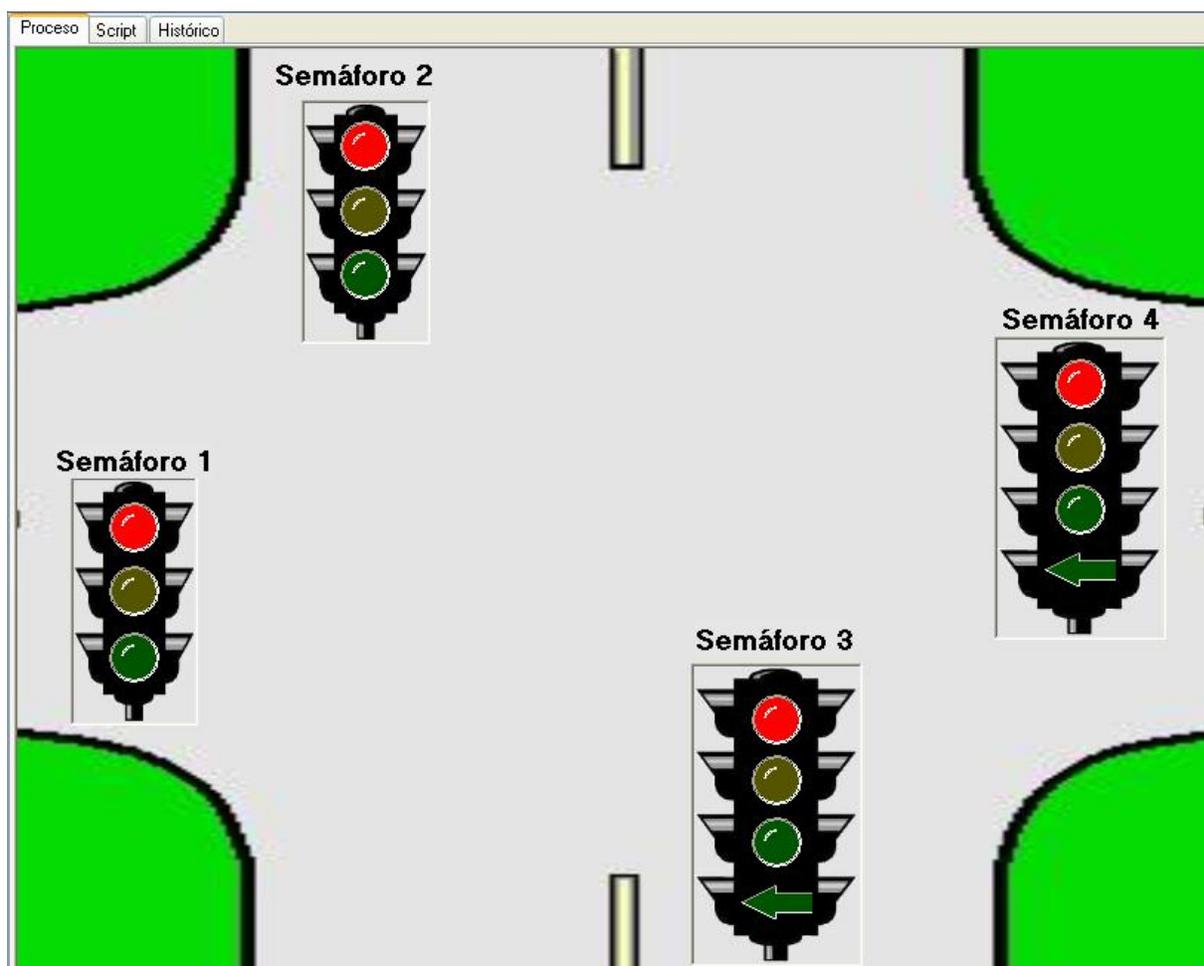


Figura 4.41 Etapa 0. Fuente: OPCSoft -Elaboración Propia

**Etapa 0:** Es la etapa inicial de este proceso, en donde todos los semáforos se encuentran por un período de un segundo en rojo. Al accionar el pulsador comienza el desarrollo de las etapas subsiguientes.

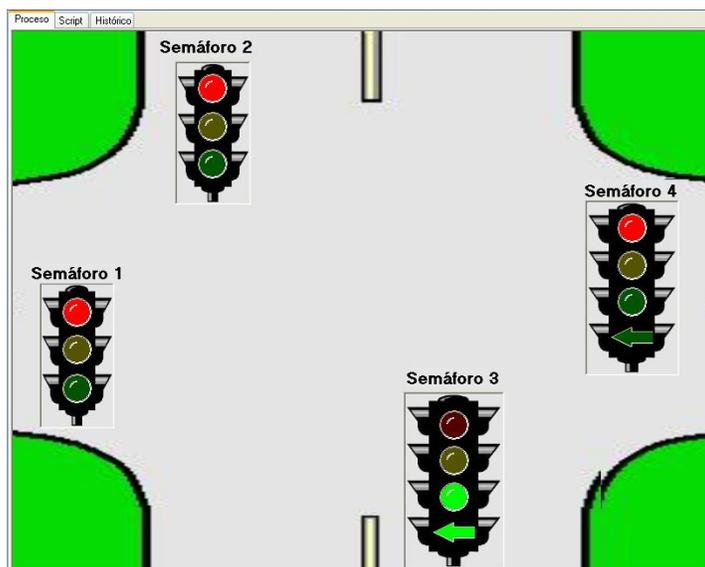


Figura 4.42 Etapa 1. Fuente: OPCSoft -Elaboración Propia

**Etapa 1:** Es la etapa donde cambia el semáforo # 03. La luz verde (V3) se enciende simultáneamente con la luz de cruce (C3), por un período de 6 segundos. Los demás semáforos permanecen en rojo.

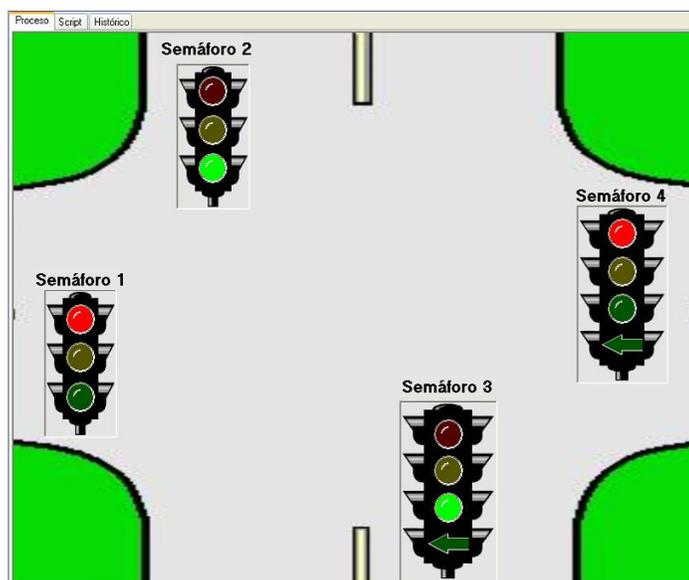


Figura 4.43 Etapa 2. Fuente: OPCSoft -Elaboración Propia

**Etapa 2:** Es la etapa donde se enciende la luz verde (V2) por un período de 8 segundos. Y por el mismo tiempo se mantiene encendida la luz verde (V3). Los demás semáforos permanecen en rojo.

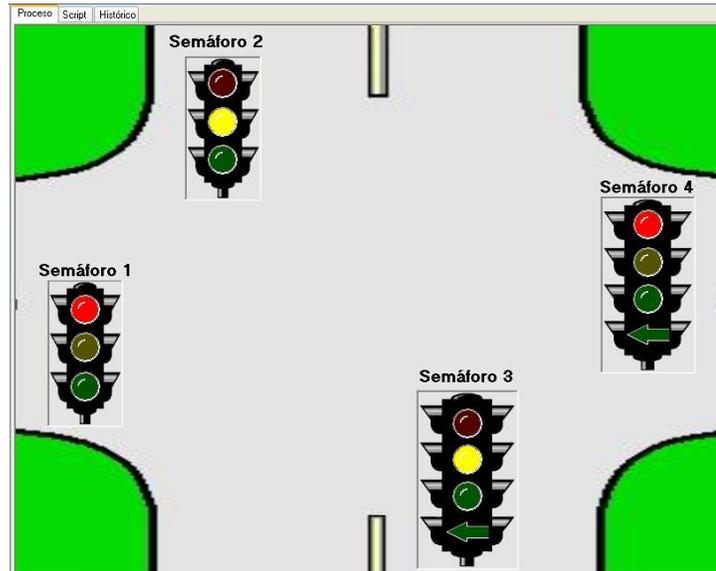


Figura 4.44 Etapa 3. Fuente: OPCSoft -Elaboración Propia

**Etapa 3:** Es la etapa donde se enciende la luz amarilla (A2) y (A3) por un período de 1 segundo. Los demás semáforos permanecen en rojo.

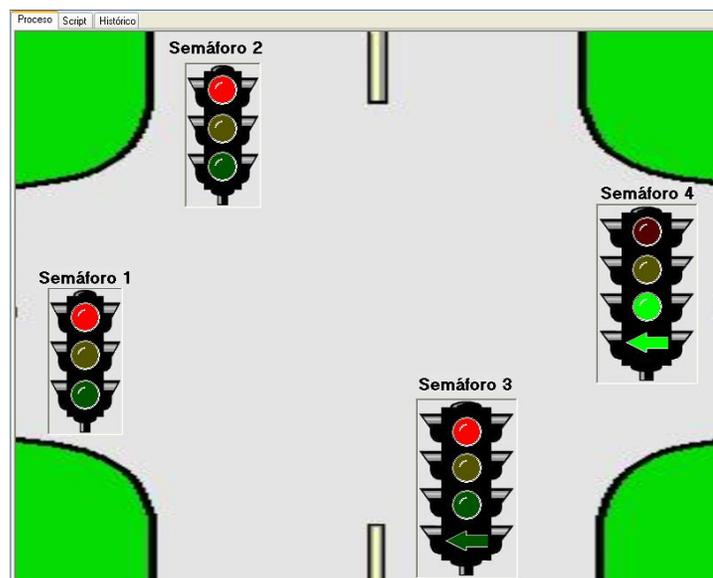


Figura 4.45 Etapa 4. Fuente: OPCSoft -Elaboración Propia

**Etapa 4:** Es la etapa donde cambia el semáforo # 04. La luz verde (V4) se enciende simultáneamente con la luz de cruce (C4), por un período 6 segundos. Los demás semáforos permanecen en rojo.

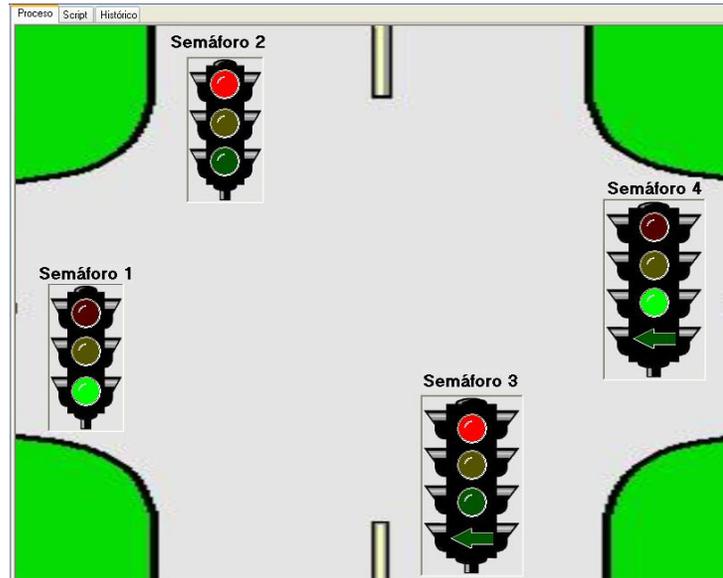


Figura 4.46 Etapa 5. Fuente: OPCSoft -Elaboración Propia

**Etapa 5:** Es la etapa donde se enciende la luz verde (V1) por un período de 8 segundos. Y por el mismo tiempo se mantiene encendida la luz verde (V4). Los demás semáforos permanecen en rojo

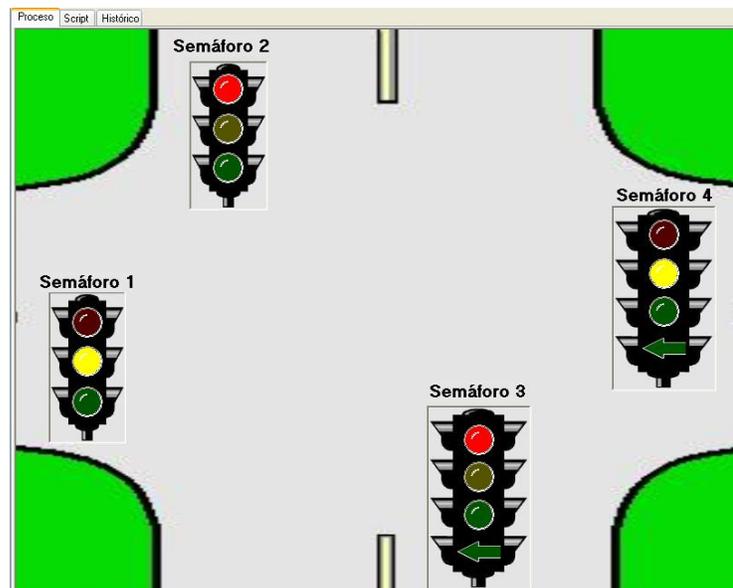


Figura 4.47 Etapa 6. Fuente: OPCSoft -Elaboración Propia

**Etapa 6:** Es la etapa donde se enciende la luz amarilla (A1) y (A4) por un período de 1 segundo. Los demás semáforos permanecen en rojo. Luego de esta etapa el ciclo

sigue su secuencia volviendo a la etapa 1. El botón finalizar interrumpe el ciclo y regresa la aplicación a la etapa 0.

#### 4.5.2.- Control del llenado y vaciado de un tanque

A continuación se presentan imágenes de las etapas del llenado del tanque.

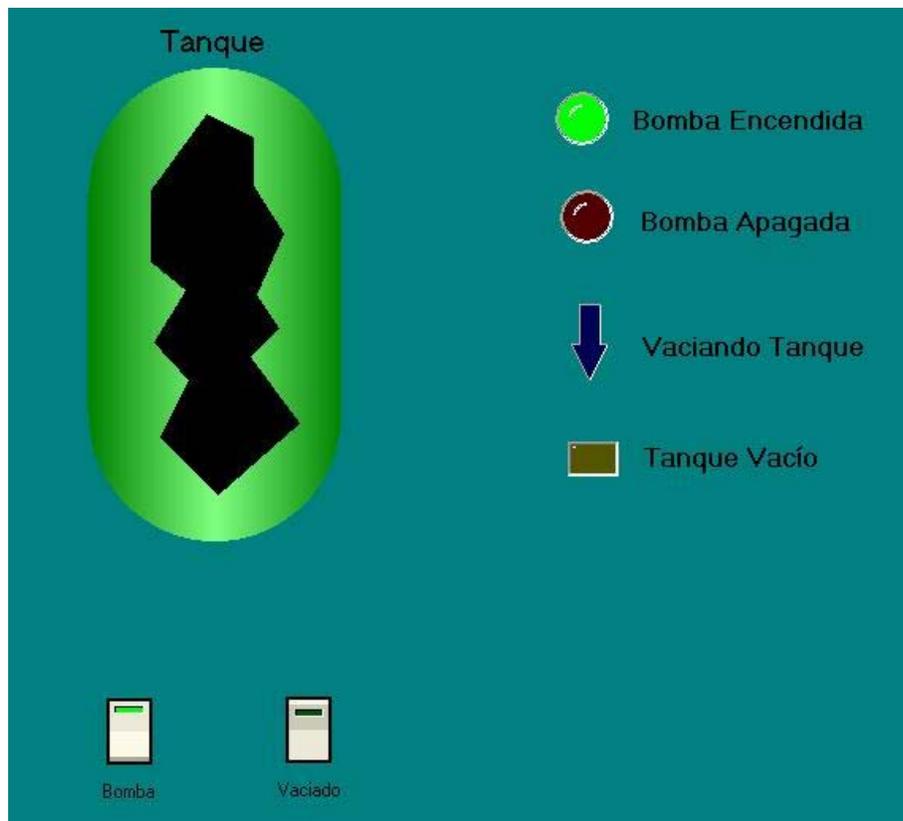


Figura 4.48 Tanque vacío. Fuente: OPCSoft -Elaboración Propia

La figura 4.48 representa el comienzo del proceso de control de llenado y vaciado de un tanque. Se muestra el tanque vacío. El pulsador de encendido de la bomba y el led correspondiente se encuentran encendidos.

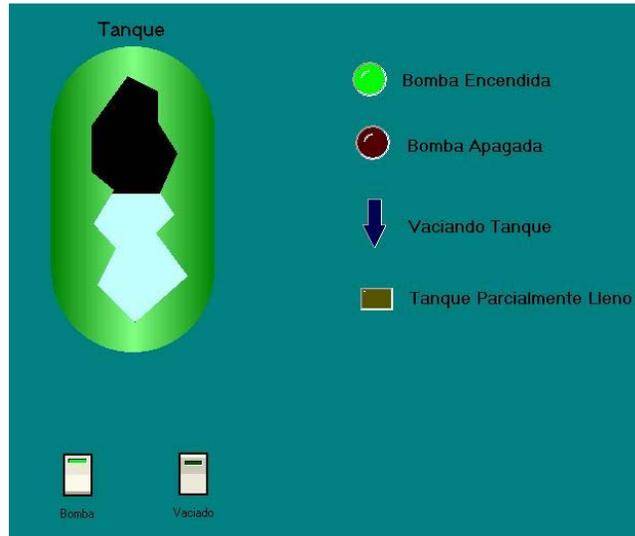


Figura 4.49 Tanque parcialmente lleno. Fuente: OPCSoft -Elaboración Propia

La figura 4.49 representa el tanque parcialmente lleno. El pulsador de encendido de la bomba y el led correspondiente se encuentran encendidos.

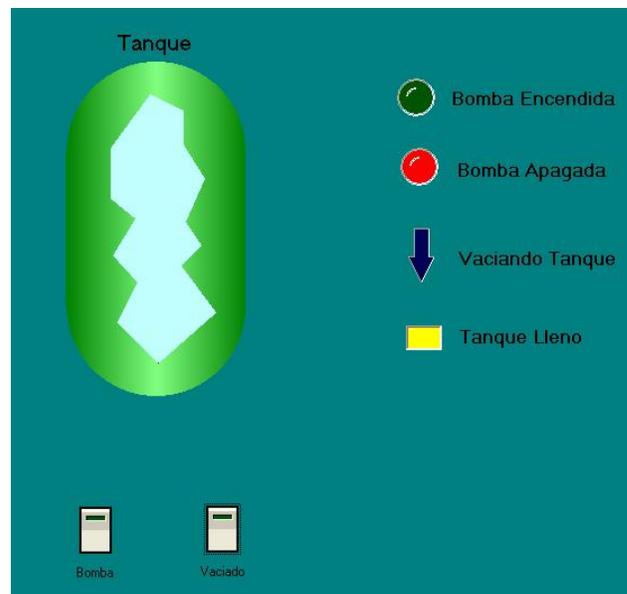


Figura 4.50 Tanque lleno. Fuente: OPCSoft -Elaboración Propia

La figura 4.50 representa el tanque lleno. Se encuentra encendido el led que indica que el tanque se encuentra lleno. Y el led que indica que la bomba se encuentra apagada.



## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**



## 5.1.- CONCLUSIONES.

- La eficacia de la comunicación implementada quedó establecida al comprobarse la interconexión entre el cliente OPC y el Servidor OPC. Y con dicha comunicación implementada, se controlaron los cambios ocurridos durante el transcurso de los procesos diseñados.
- El servidor OPC permite mantener una comunicación bidireccional durante el transcurso de la simulación del proceso, es decir, el PLC controla la aplicación y al mismo tiempo desde el cliente OPC se envían instrucciones que modifican las variables de entrada del PLC.
- El Cliente OPC es una herramienta que puede ser utilizada con fines didácticos y que permite simular procesos industriales, mostrados de manera gráfica y por medio del cual se puede interactuar, los cuales pueden ser controlados por un PLC de manera que no se necesita la construcción de un proceso industrial en físico.
- El estándar OPC agrega un paso en la comunicación entre el PLC y el computador, lo cual supone un retraso en el tiempo de respuesta y puede afectar el proceso de automatización cuando se necesita realizar mediciones instantáneas y repetitivas.



## 5.2.- RECOMENDACIONES.

- Implementar las aplicaciones en prácticas del laboratorio de automatización industrial de la escuela de Ingeniería Mecánica de la Universidad de Carabobo.
- Si se quiere realizar otra aplicación diferente a las desarrolladas en el presente trabajo de grado, asegurarse de vincular los ítems en el Servidor OPC con la dirección correcta en el PLC juntamente con su tipo de variable correspondiente.
- Cuando se quiera vincular los objetos en el cliente OPC con los ítems del servidor OPC, es recomendable abrir primero el programa del servidor OPC y cargar las variables correspondientes a la aplicación que ejecutará en el cliente OPC y luego trabajar con los objetos en el cliente OPC.



## REFERENCIAS BIBLIOGRAFICAS

- [1] Teoría básica sobre Controladores Lógicos Programables. (2006). [On-Line]. Disponible en <http://automaindus.googlepages.com>
- [2] Empresa especializada en la fabricación, asesoría y consulta sobre Controladores Lógicos Programables. (2008). [On-line]. Disponible en <http://www.automation.siemens.com>
- SIEMENS SERIE 7. Manual de programación de autómatas de la serie S7. 2004
- Empresa especializada en la fabricación, asesoría y consulta sobre Controladores Lógicos Programables. (2008). [On-line]. Disponible en <http://www.ab.com/plclogic>.
- Teoría sobre Automatización Industrial. Disponible [On-Line] en: [http://www.grupomaser.com/PAG\\_Cursos/Auto/auto2/auto2/PAGINA%20PRINCIPAL/index.htm](http://www.grupomaser.com/PAG_Cursos/Auto/auto2/auto2/PAGINA%20PRINCIPAL/index.htm)
- [3] Información, Noticias, certificaciones y Teoría sobre OPC. [On-Line] Disponible en <http://www.OPCFoundation.org>



# APÉNDICE I

## SELECCIÓN DEL SERVIDOR OPC



En este apéndice se utilizará un procedimiento que permitirá la selección del Servidor OPC a ser implementado en la conexión entre un Controlador Lógico Programable (PLC) y un computador. En el procedimiento se evalúan varias alternativas de solución, se les asigna una ponderación a cada uno de los criterios y por último se selecciona la mejor alternativa como solución.

A continuación se procede a ejecutar el procedimiento:

### I.- Probables Soluciones

En la tabla I se especifican las alternativas de Solución en cuanto a la selección del software.

**Tabla I. Alternativas de Solución.** Fuente: Elaboración Propia

<i>Psx</i>	<i>Alternativas de Solución</i>
<b>Ps1</b>	KEPServerEX OPC Server
<b>Ps2</b>	ICONICS OPC Server
<b>Ps3</b>	S7-200 PC Access

### II.- Restricciones

En la tabla se especifican las restricciones.

**Tabla II. Restricciones.** Fuente: Elaboración Propia

<i>Restricciones</i>	
<b>R1</b>	Costo del Software menor a 1000 \$
<b>R2</b>	Compatibilidad con S7-200 CPU 224
<b>R3</b>	Certificación de OPC Foundation



### III.- Aplicación de restricciones a las alternativas de solución

En la tabla III se evalúan las soluciones respecto de cada restricción existente donde posteriormente se obtendrá la solución final con la cual se va a trabajar.

Tabla III. Aplicación de Restricciones. Fuente: Elaboración Propia

<i>Psx</i>	<i>R1</i>	<i>R2</i>	<i>R3</i>
<b>Ps1</b>	No	Si	Si
<b>Ps2</b>	No	No	Si
<b>Ps3</b>	Si	Si	Si

### IV.- Soluciones Obtenidas

Se tabulan las soluciones obtenidas en la tabla IV

Tabla IV. Soluciones Obtenidas. Fuente: Elaboración Propia

<i>Psx</i>	<i>Sx</i>	<i>Solución</i>
<b>Ps1</b>	S1	KEPServerEX OPC Server
<b>Ps4</b>	S2	S7-200 PC Access

### V.- Criterios

En la tabla V se describen y se codifican los criterios para realizar la evaluación

Tabla V. Criterios para la evaluación. Fuente: Elaboración Propia

<i>Cx</i>	<i>Criterio</i>
<b>C1</b>	Disponibilidad Inmediata
<b>C2</b>	Facilidad de Operación
<b>C3</b>	Menor Costo del Software

### VI.- Ponderación de Criterios

En esta sección se ponderaran los criterios de acuerdo a la importancia que tiene uno respecto del otro, utilizando una escala del 1 al 4.



Tabla VI. Criterios para la evaluación. Fuente: Elaboración Propia

<i>Pcx</i>	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>N</i>	<i>Pond</i>
<b>C1</b>				1	3
<b>C2</b>	C1			0	2
<b>C3</b>	C3	C3		2	4

## VII.- Ponderación de soluciones respecto a Criterios

Aquí se evalúan las soluciones obtenidas en pasos anteriores y tabulados en la tabla V con respecto a cada criterio, utilizando una escala del 1 al 2.

### Criterio 1: Disponibilidad Inmediata

Tabla VII. Ponderación - Criterio 1. Fuente: Elaboración Propia

<i>Sx</i>	<i>S1</i>	<i>S2</i>	<i>N</i>	<i>Pond</i>
<b>S1</b>			0	1
<b>S2</b>	S2		1	2
<b>Importancia</b>	2	1		

Solución Indicada: S2

### Criterio 2: Facilidad de Operación

Tabla VIII. Ponderación - Criterio 2. Fuente: Elaboración Propia

<i>Sx</i>	<i>S1</i>	<i>S2</i>	<i>N</i>	<i>Pond</i>
<b>S1</b>			1	2
<b>S2</b>	S2		0	1
<b>Importancia</b>	1	2		

Solución Indicada: S1



### Criterio 3: Menor Costo del Software

Tabla IX Ponderación - Criterio 3. Fuente: Elaboración Propia

<i>Sx</i>	<i>S1</i>	<i>S2</i>	<i>N</i>	<i>Pond</i>
S1			1	1
S2	S1		0	2
<b>Importancia</b>	2	1		

Solución Indicada: S2

### VIII.- Ponderación Final

Al multiplicar el valor de los criterios por el valor que obtuvo la solución con respecto a cada criterio, se obtiene la sumatoria total de cada solución respecto a todos los criterios. Una vez realizado esto, se comparan los valores obtenidos y se selecciona el mayor de todos, resultando así la mejor solución del problema descrito con anterioridad.

#### Solucion 1:

Tabla X Ponderación – Solución 1. Fuente: Elaboración Propia

<i>Criterio</i>	<i>Pcx</i>	<i>Ps1</i>	<i>Pcx*Ps1</i>
1	4	1	4
2	3	2	6
3	5	1	5
			$\Sigma = 15$



## Solucion 2:

Tabla XI Ponderación – Solución 2. Fuente: Elaboración Propia

<i>Criterio</i>	<i>Pcx</i>	<i>Ps2</i>	<i>Pcx*Ps1</i>
1	4	2	8
2	3	1	3
3	5	2	10
			$\Sigma = 21$

Solución Apropiada: S2

Luego de evaluar todas las posibles soluciones formuladas respecto a las restricciones planteadas y los criterios de trabajo establecidos en el trabajo especial de grado, se puede concluir que la solución 2 (**S7-200 PC Access**) es la más indicada para satisfacer los requerimientos del proyecto.



# APÉNDICE II

## SELECCIÓN DEL CLIENTE OPC



En este apéndice se utilizará un procedimiento que permitirá la selección del Cliente OPC en el cuál se desarrollará la aplicación en el computador y que será controlada por el PLC. En el procedimiento se evalúan varias alternativas de solución, se les asigna una ponderación a cada uno de los criterios y por último se selecciona la mejor alternativa como solución.

A continuación se procede a ejecutar el procedimiento:

### I.- Probables Soluciones

En la tabla XII se especifican las alternativas de Solución en cuanto a la selección del software.

**Tabla XII. Alternativas de Solución.**

<i>Psx</i>	<i>Alternativas de Solución</i>
<b>Ps1</b>	LabVIEW 8.2 NI
<b>Ps2</b>	VisSim
<b>Ps3</b>	Microsoft Excel
<b>Ps4</b>	OPC SOFT

### II.- Restricciones

En la tabla XIII se especifican las restricciones expuestas anteriormente.

**Tabla XIII. Restricciones.**

<i>Psx</i>	<i>R1</i>	<i>R2</i>	<i>R3</i>
<b>Ps1</b>	Si	Si	Si
<b>Ps2</b>	No	No	Si
<b>Ps3</b>	No	Si	Si
<b>Ps4</b>	Si	Si	Si



### III.- Aplicación de restricciones a las alternativas de solución

En la tabla XIV se evalúan las soluciones respecto de cada restricción existente donde posteriormente se obtendrá la solución final con la cual se va a trabajar.

Tabla XIV. Aplicación de Restricciones.

<i>Restricciones</i>	
<b>R1</b>	Costo del Software menor a 1200 \$
<b>R2</b>	Interacción grafica
<b>R3</b>	Comunicación por medio de OPC

### IV.- Soluciones Obtenidas

Se tabulan las soluciones obtenidas en la tabla XV

Tabla XV. Soluciones Obtenidas.

<i>Psx</i>	<i>S</i>	<i>Solución</i>
<b>Ps1</b>	S1	LabVIEW 8.2 NI
<b>Ps4</b>	S2	OPC SOFT

### V.- Criterios

En la tabla XVI se describen y se codifican los criterios para realizar la evaluación

Tabla XVI. Criterios para la evaluación

<i>Cx</i>	<i>Criterio</i>
<b>C1</b>	Ambiente didáctico
<b>C2</b>	Disponibilidad Inmediata
<b>C3</b>	Capacidad de aplicaciones
<b>C4</b>	Menor Costo del Software



## VI.- Ponderación de Criterios

En esta sección se ponderaran los criterios de acuerdo a la importancia que tiene uno respecto del otro, utilizando una escala del 1 al 5.

Tabla XVII. Criterios para la evaluación

<i>Pcx</i>	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>N</i>	<i>Pond</i>
<b>C1</b>					2	4
<b>C2</b>	C1				0	2
<b>C3</b>	C1	C3			1	3
<b>C4</b>	C4	C4	C4		3	5

## VII.- Ponderación de soluciones respecto a Criterios

Aquí se evalúan las soluciones obtenidas en pasos anteriores y tabulados en la tabla XVIII con respecto a cada criterio, utilizando una escala del 1 al 2.

### Criterio 1: Ambiente didáctico

Tabla XVIII. Ponderación - Criterio 1

<i>Sx</i>	<i>S1</i>	<i>S2</i>	<i>N</i>	<i>Pond</i>
<b>S1</b>			0	1
<b>S2</b>	S2		1	2
<b>Importancia</b>	2	1		

Solución Indicada: S2



**Criterio 2: Disponibilidad Inmediata**

**Tabla XIX. Ponderación - Criterio 2**

<i>Sx</i>	<i>S1</i>	<i>S2</i>	<i>N</i>	<i>Pond</i>
S1			0	1
S2	S2		1	2
<b>Importancia</b>	2	1		

Solución Indicada: S2

**Criterio 3: Capacidad de aplicaciones**

**Tabla XX. Ponderación - Criterio 3**

<i>Sx</i>	<i>S1</i>	<i>S2</i>	<i>N</i>	<i>Pond</i>
S1			1	2
S2	S1		0	1
<b>Importancia</b>	1	2		

Solución Indicada: S1

**Criterio 4: Menor Costo del Software**

**Tabla XXI. Ponderación - Criterio 4**

<i>Sx</i>	<i>S1</i>	<i>S2</i>	<i>N</i>	<i>Pond</i>
S1			0	1
S2	S2		1	2
<b>Importancia</b>	2	1		

Solución Indicada: S2

**4.4.9 Ponderación Final**

Al multiplicar el valor de los criterios por el valor que obtuvo la solución con respecto a cada criterio, se obtiene la sumatoria total de cada solución respecto a todos los criterios. Una vez realizado esto, se comparan los valores obtenidos y se selecciona el mayor de todos, resultando así la mejor solución del problema descrito con anterioridad.



Solucion 1:

Tabla XXII. Ponderación – Solución 1

<i>Criterio</i>	<i>Pcx</i>	<i>Ps1</i>	<i>Pcx*Ps1</i>
1	4	2	8
2	2	2	4
3	3	1	3
4	5	2	10
			$\Sigma = 25$

Solucion 2:

Tabla XXIII. Ponderación – Solución 2

<i>Criterio</i>	<i>Pcx</i>	<i>Ps2</i>	<i>Pcx*Ps1</i>
1	4	1	4
2	2	1	2
3	3	2	6
4	5	1	5
			$\Sigma = 17$

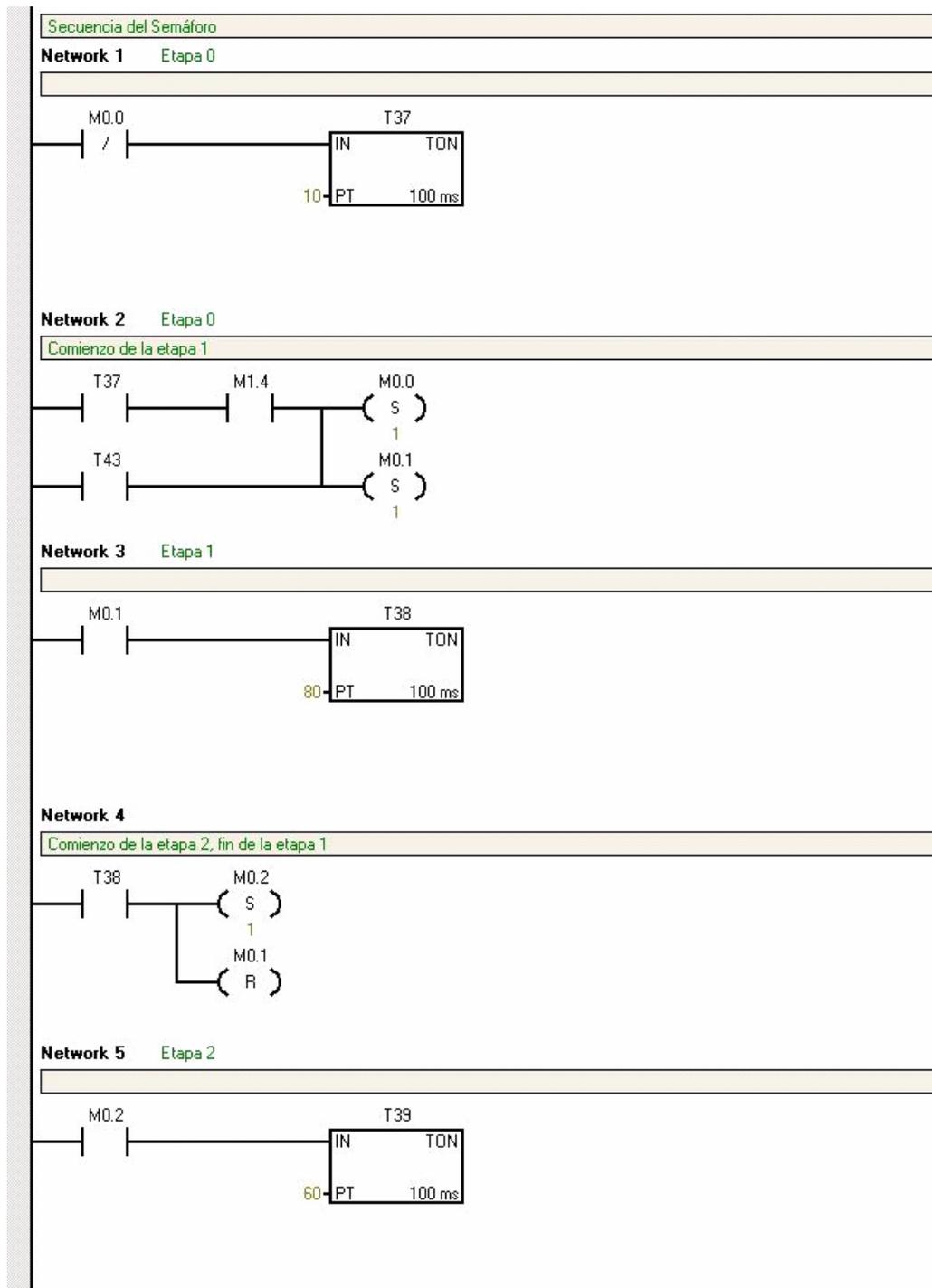
**Solución Apropriada: S2**

Luego de evaluar todas las posibles soluciones formuladas respecto a las restricciones planteadas y los criterios de trabajo establecidos en el trabajo especial de grado, se puede concluir que la solución 2 (**OPC SOFT**) es la más indicada para satisfacer los requerimientos del proyecto.



# ANEXO 1

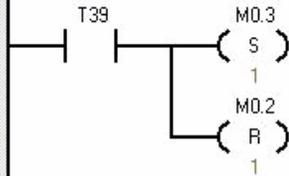
## PROGRAMACIÓN DEL CONTROL DE TRÁFICO EN UNA INTERSECCIÓN DE VÍAS



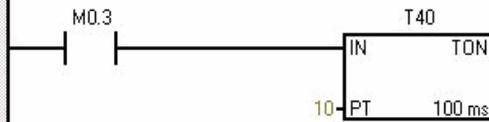


**Network 6**

Comienzo de la etapa 3, fin de la etapa 2

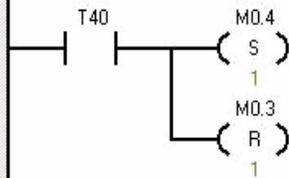


**Network 7** Etapa 3

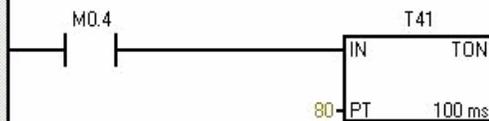


**Network 8**

Comienzo de la etapa 4, fin de la etapa 3

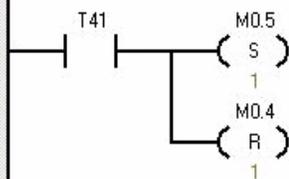


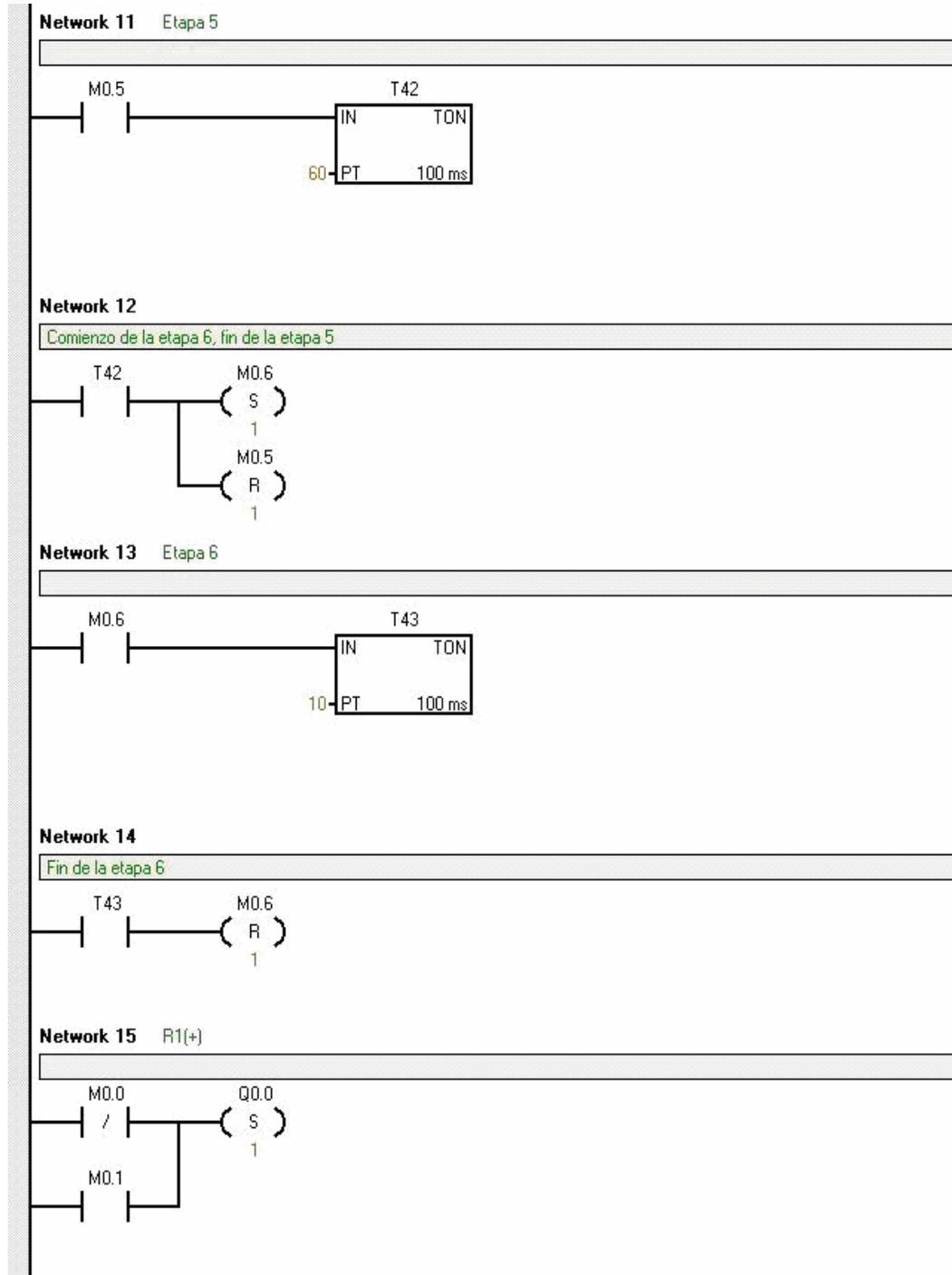
**Network 9** Etapa 4

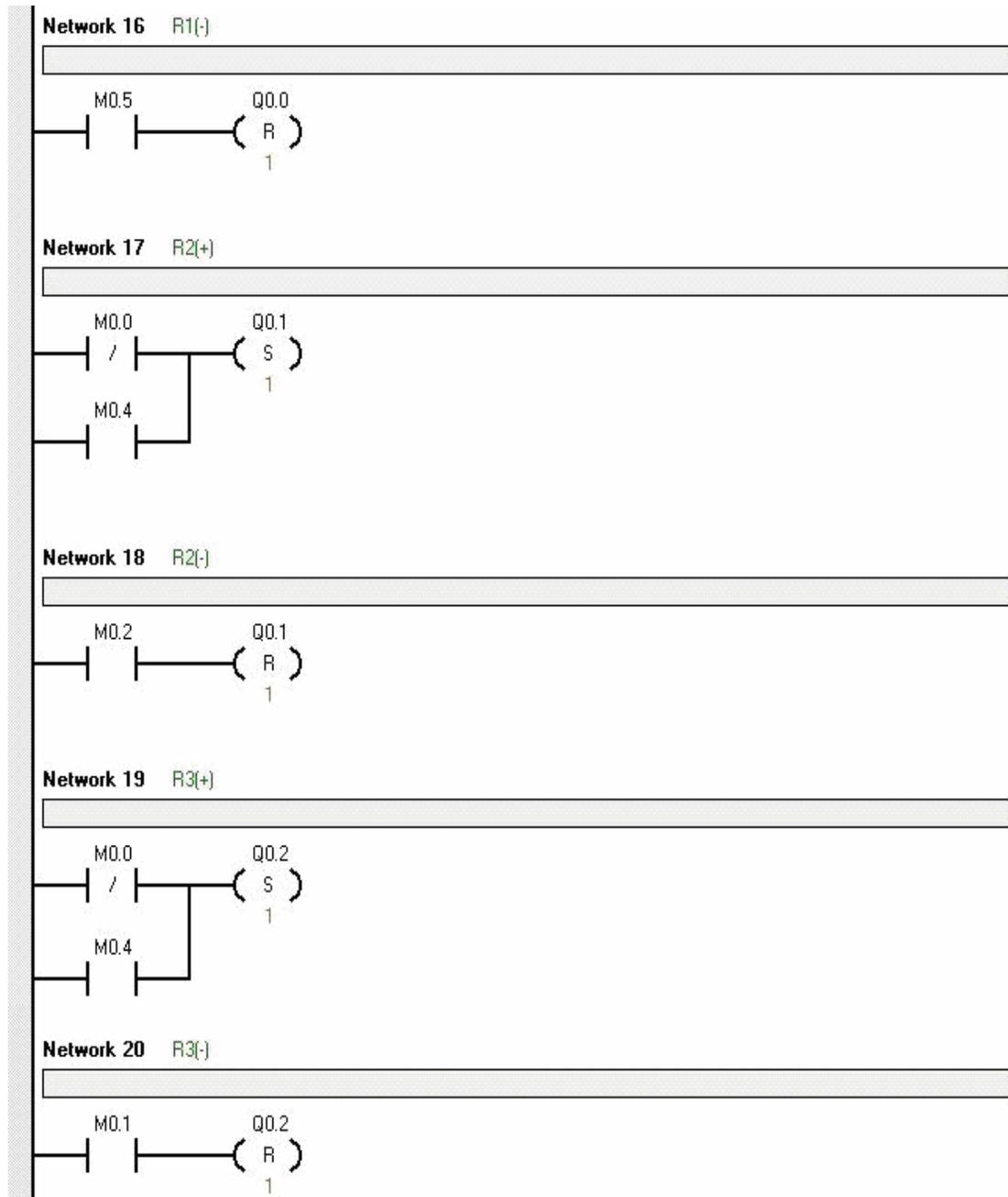


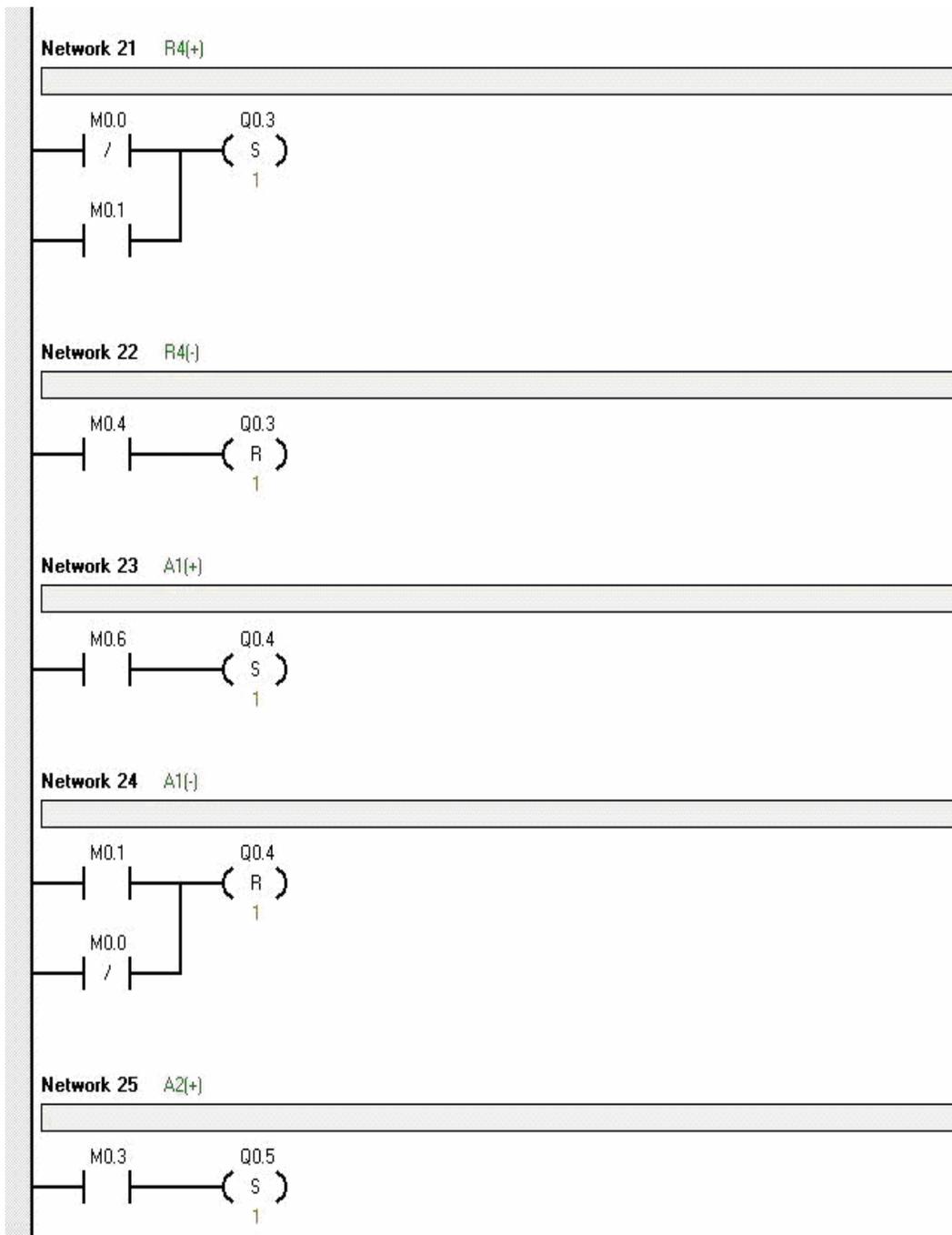
**Network 10**

Comienzo de la etapa 5, fin de la etapa 4



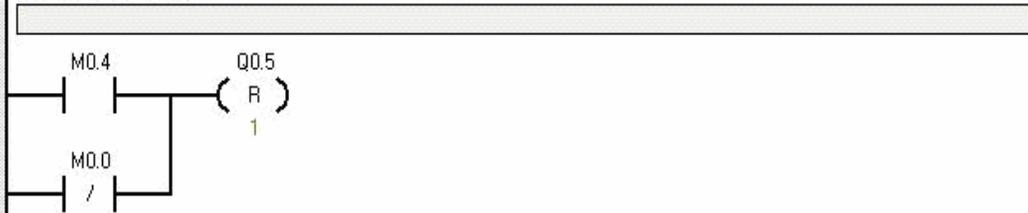








Network 26 A2(-)



Network 27 A3(+)



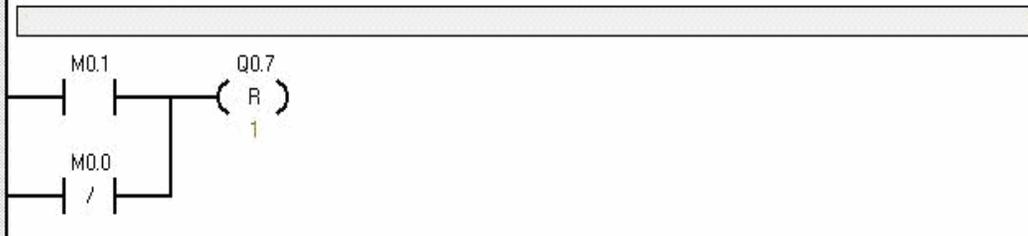
Network 28 A3(-)

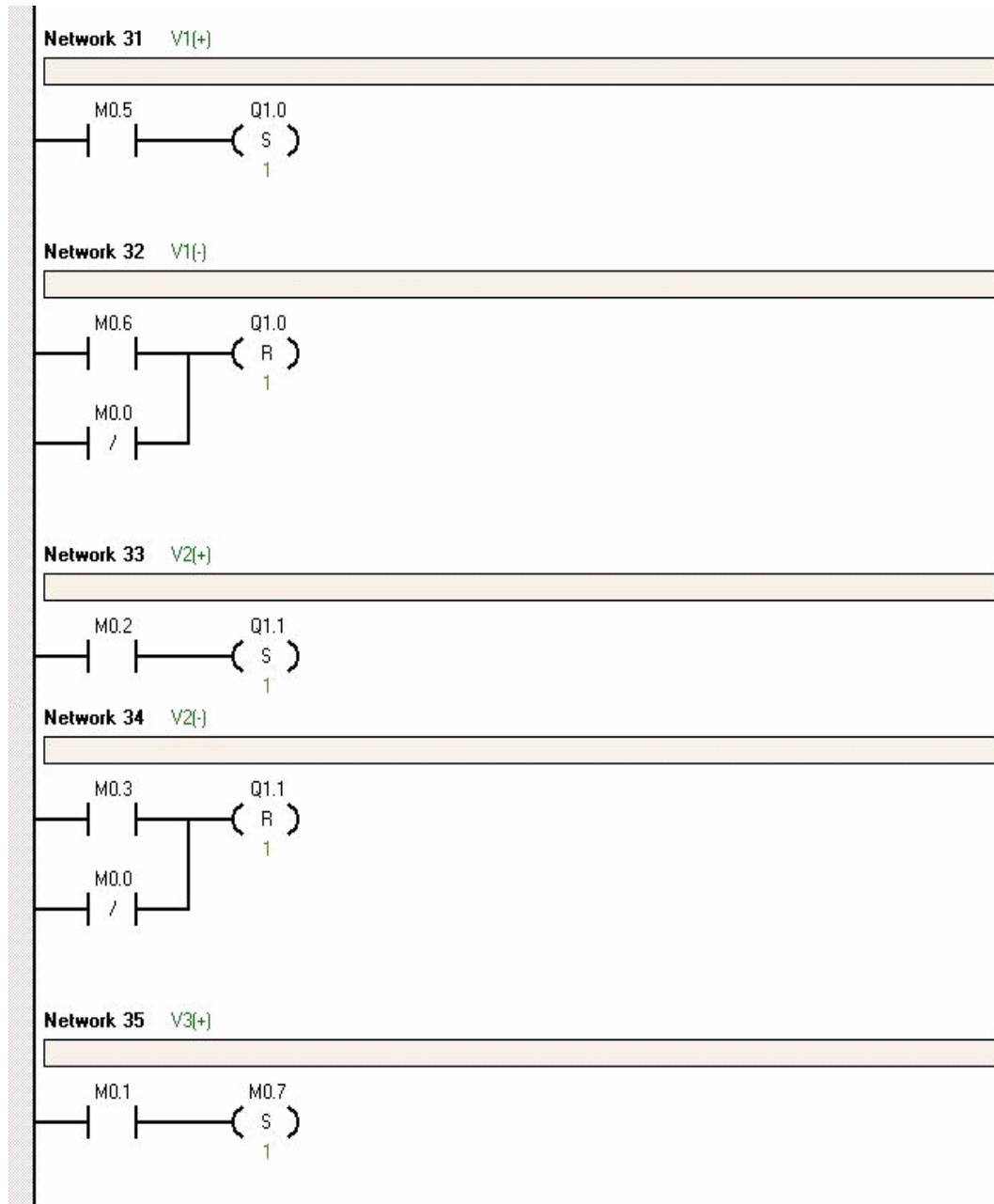


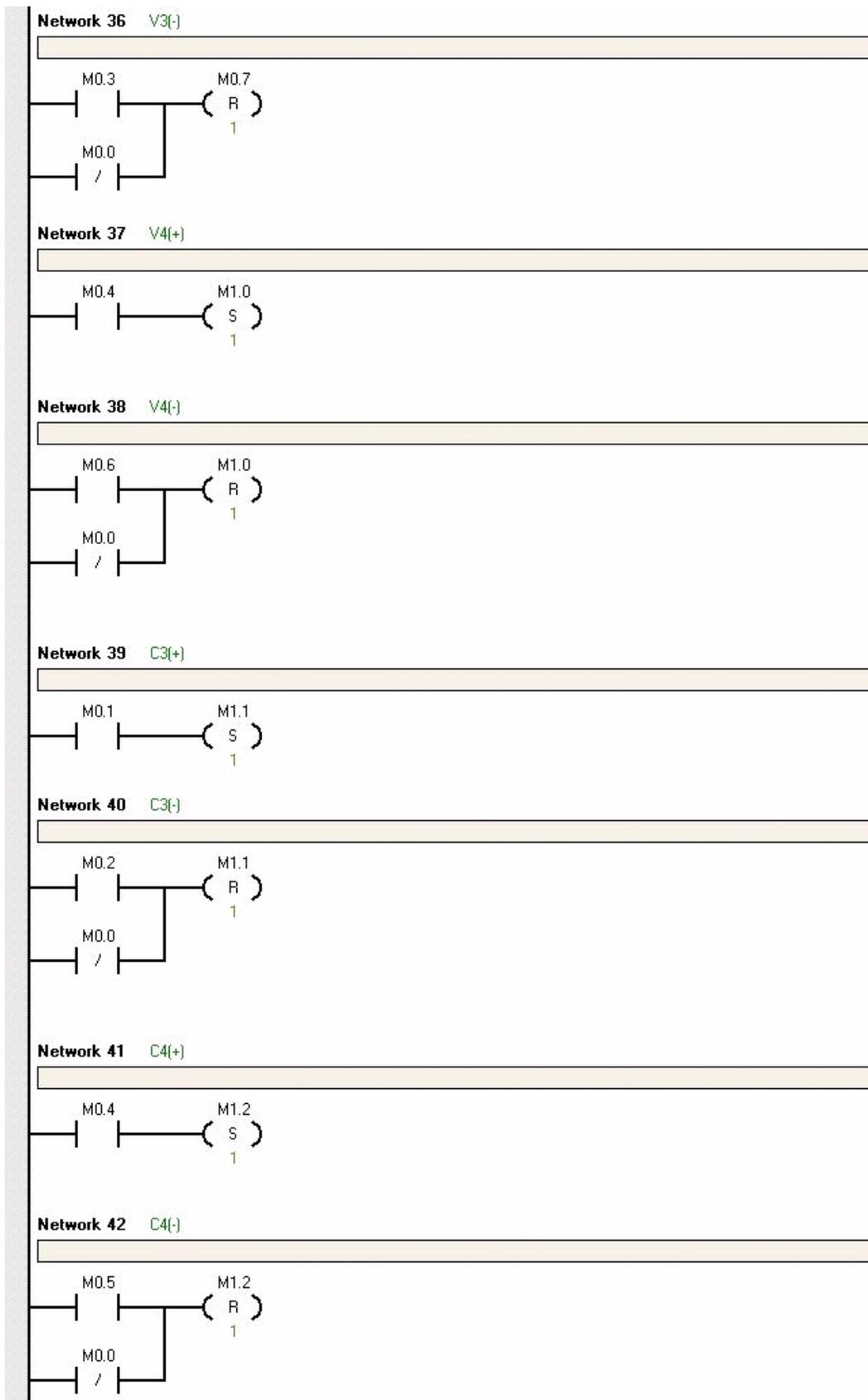
Network 29 A4(+)



Network 30 A(-)









# ANEXO 2

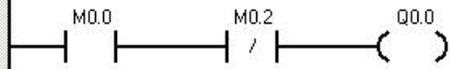
## PROGRAMACIÓN DEL CONTROL DE LLENADO Y VACIADO DE UN TANQUE



Programación para el control de llenado y vaciado del tanque

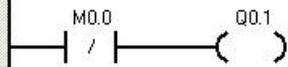
**Network 1** Título de segmento

Encender Bomba



**Network 2**

Apagar Bomba



**Network 3**

Vaciar Tanque

