

**IMPLEMENTACIÓN DE UN PROTOTIPO DE ADQUISICIÓN DE DATOS Y
SUPERVISIÓN DE CONTROL (SCADA) EN EL ENVASADO DE LÍQUIDOS
PARA LA INDUSTRIA DE PRODUCTOS QUIMICOS Y FARMACÉUTICOS**

EDISSON GARCÍA QUINTERO

LUIS FELIPE RAMÍREZ ARIAS

INSTITUCIÓN UNIVERSITARIA DE ENVIGADO

FACULTAD DE INGENIERÍAS

PROGRAMA INGENIERÍA ELECTRÓNICA

ENVIGADO

2011

Este trabajo es dedicado a nuestras familias que con esfuerzo y esmero hicieron posible que cumpliéramos nuestros sueños profesionales, destacando el amor de nuestras madres que siempre estuvieron a nuestro lado apoyándonos para lograr el cumplimiento de esta importante meta.

AGRADECIMIENTOS

Mis más sinceros agradecimientos al señor Andrés Tobón Carmona Gerente de Laboratorios Athos, quien nos abrió las puertas de su empresa para que buscáramos una solución a los problemas de operaciones manuales que afectan el rendimiento en el proceso de envasado de producto.

Al profesor Carlos Andrés González, Ingeniero Eléctrico e Instructor del SENA (PLC), quien nos guio con sus conocimientos, brindándonos las bases para la elaboración del software para los sistemas SCADA.

A nuestra asesora Paula Cristina Zapata Cortes, quien nos acompañó en la elaboración del trabajo de grado.

CONTENIDO

	Pág.
1. PRESENTACIÓN DEL TRABAJO DE GRADO	1
1.1 PROBLEMA	1
1.1.1. Planteamiento del problema	1
1.1.2. Formulación del problema	2
1.2. OBJETIVOS	2
1.2.1. Objetivos General	2
1.2.2. Objetivos Especificos	2
1.3. JUSTIFICACIÓN	2
1.4. DISEÑO METODOLÓGICO	3
1.5. PRESUPUESTO (FICHA DE PRESUPUESTO)	4
1.6. DIAGRAMA DE GANTT	7
2. APROXIMACIÓN A LOS SISTEMAS SCADA	8
2.1. GRUPOS DE INVESTIGACIÓN SCADA	8
2.2. GENERALIDADES DE LOS GRUPOS SCADA	9
2.2.1. Esquema Básico	10
2.2.2. Módulos de un sistema SCADA	11
3. ASPECTOS TÉCNICOS DE LABORATORIOS ATHOS	14
3.1. RESEÑA HISTÓRICA DE LA EMPRESA ATHOS	14

3.2. DESCRIPCIÓN DEL PROCESO DE OPERACIÓN DE LA MÁQUINA LLENADORA DE ENVASES	16
3.3. ETAPAS DE LLENADO DE ENVASES EN LA INDUSTRIA DE PRODUCTOS QUÍMICOS Y FARMACEUTICOS	16
3.3.1. ETAPA DE CONTROL MANUAL DE ENCENDIDO Y PROGRAMACIÓN DE LA ENVASADORA	17
3.3.2. ETAPA DE SUCCIÓN DE LÍQUIDO	17
3.3.3. ETAPA DE ALMACENAMIENTO DE MATERIA PRIMA	18
3.3.4. ETAPA DE ENVASADO	19
3.3.5. ETAPA DE SENSADO	20
3.3.6. ETAPA DE RETENCIÓN DE LÍQUIDO	21
3.4. GENERALIDADES DE LOS SENSORES	23
3.4.1. SENSORES INDUCTIVOS	23
3.5. PROCESO DE GENERACIÓN DE REPORTES UTILIZADOS ACTUALMENTE EN EL LLENADO	24
3.6. PLC	26
3.6.1. HISTORIA DE LOS PLC	27
3.6.2. VENTAJAS – DESVENTAJAS	28
3.6.3. CAMPOS DE APLICACIÓN	29
3.6.4. FUNCIONES BÁSICAS DE UN PLC	31
3.6.5. ESTRUCTURA GENERAL DEL PLC	32
3.6.6. CPU	33
3.6.7. PROGRAMADOR MONITOR (PM)	35
3.6.8. MÓDULOS DE ENTRADA SALIDA	35

3.69.4. FUENTE DE ALIMENTACIÓN	39
3.7. UNIDAD DE PROGRAMACIÓN	41
3.7.1. UNIDAD DE PROGRAMACIÓN GENERALIDADES CONTROLADOR LÓGICO PROGRAMABLE (PLC)	41
3.7.2. UNIDAD DE PROGRAMACIÓN PUERTOS DE COMUNICACIÓN	42
4. DISEÑO E IMPLEMENTACION DEL SISTEMA SCADA	44
4.1. SOFTWARE PARA LA PROGRAMCIÓN DEL PLC SIMATICS7300	44
4.1.2. REQUISITOS PARA LA INSTALACION	44
4.1.3. LENGUAJE DE PROGRAMCIÓN	45
4.1.4. LENGUAJE DE PROGRAMCIÓN FUP	45
4.1.5. LENGUAJE DE PROGRAMCIÓN AWL	46
4.1.4 LENGUAJE DE PROGRAMCIÓN KOP	46
4.2. LISTA DE OPERACIONES DE CONTAJE	49
5. CREACIÓN DEL PROYECTO EN SIMATIC	55
5.1. SIMULACIÓN DEL PROYECTOS EN SIMATIC	58
6. DESARROLLO DEL PROYECTO EN SIMATIC	59
6. 1 DESARROLLO DEL PROGRAMA EN SIMATIC	60
7. PROGRAMCIÓN ENTORNO GRÁFICO	63
7.1. DESCARGA E INSTALACIÓN SOFTWARE DE ENTORNO GRÁFICO “WINLOG”	63
7.2. MANEJO DE PROGRAMA “WINLOG”	66
8. ELABORACION DEL SOFTWARE DE VISUALIZACIÓN	74
8.1. DESCRIPCION DEL PROCESO	74

8.2. REGISTRO DE OPERADOR Y REFERENCIA DEL PRODUCTO A ENVASAR	74
8.3. ELABORACIÓN DEL CÓDIGO DE VISUALIZACIÓN	77
8.3.1. CÓDIGO	77
8.3.2. TABLA DE VARIABLES GLOBALES	82
8.3.3. LISTA DE INSTRUCCIONES WINLOG	83
9. CONCLUSIONES	88
10. RECOMENDACIONES	89
11. BIBLIOGRAFIA	90
12. ANEXOS	

LISTA DE FIGURAS

	Pág.
Figura 1. Esquema SCADA	10
Figura 2. Ejemplo de interfaz usuario	11
Figura 3. Configuración interfaz gráfica	12
Figura 4. Ejemplo interfaz gráfica operador	12
Figura 5. Módulo del proceso	13
Figura 6. Gestión y archivo de datos	13
Figura 7. Llenadora de envase	16
Figura 8. Control encendido y programador manual llenadora	17
Figura 9. Pistón neumático	17
Figura 10. Bomba manual	18
Figura 11. Tolva de distribución	18
Figura 12. Tolva de distribución con materia prima	19
Figura 13. Inyectora de líquido	19
Figura 14. Pedal de accionamiento	20
Figura 15. Sensores inductivos	20
Figura 16. Émbolo de pistón neumático	21
Figura 17. Cilindros de retención de líquido	21
Figura 18. Tapones de retención de líquido	22
Figura 19. Máquina reemplazada	22
Figura 20. Circulación corriente	23
Figura 21. Campo magnético	23

Figura 22. Formato de reportes	25
Figura 23. PLC siemens S7 200 y 300	26
Figura 24. PLC MODICON	27
Figura 25. Diagramas de bloques de la estructura de un PLC	33
Figura 26. Unidad de procesamiento central del PLC	34
Figura 27. Resumen de las partes del PLC	41
Figura 28. PLC Simatic S7-300 CPU 313	42
Figura 29. Ejemplo de un segmento en FUP	45
Figura 30. Ejemplo de un segmento en AWL	46
Figura 31. Ejemplo de un segmento KOP	46
Figura 32. Lista de operaciones de temporizador	51
Figura 33. Ventana de asistente de un proyecto	55
Figura 34. Ventana de asistente de un proyecto escogencia de la CPU	55
Figura 35. Ventana de asistente de un proyecto escogencia del lenguaje de programación	56
Figura 36. Ventana de asistente de un proyecto – nombre del proyecto	56
Figura 37. Ventana de configuración PLC	57
Figura 38. Plantilla de programación de PLC	57
Figura 39. Activar simulador del PLC	58
Figura 40. Ventana de simulación del PLC	58
Figura 41. Tabla de datos	59
Figura 42. Ventana de inicio instalación del WINLOG	63
Figura 43. Ventana de licencia del WINLOG	64
Figura 44. Ventana instalación del WINLOG	64

Figura 45. Ventana de instalación del WINLOG	65
Figura 46. Ventana de finalización de la instalación WINLOG	65
Figura 47. Proyecto WINLOG	66
Figura 48. Configuración parámetros WINLOG	67
Figura 49. Configuraciones parámetros adaptador WINLOG	68
Figura 50. Configuración parámetros conexión WINLOG	68
Figura 51. Configuración parámetros divice WINLOG	69
Figura 52. Variables digitales	69
Figura 53. Pestañas de muestreo	70
Figura 54. Variables numéricas	71
Figura 55. Pestaña general	72
Figura 56. Eventos	72
Figura 57. Variables numéricas	74
Figura 58. Configuración	75
Figura 59. Configuración usuarios	75
Figura 60. Procesos	77
Figura 61. Tabla de Variables globales	82

LISTA DE TABLAS

	Pag.
Tabla 1. Características técnicas PLC Simatic S7-300 CPU 313C	42
Tabla 2. Instrucciones de operaciones lógicas	47
Tabla 3. Instrucciones operaciones de comparación	48
Tabla 4. Características instrucciones ZAEHLER	49
Tabla 5. Características instrucción Z_RUECK	49
Tabla 6. Instrucción (SZ) (poner contador inicial)	50
Tabla 7. Instrucción (ZV) (incrementar contador)	50
Tabla 8. Instrucción (ZR) (Decrementar contador)	51
Tabla 9. Temporizadores	52
Tabla 10. Instrucción de temporizador S_IMPULS	52
Tabla 11. Instrucción de temporizador S_VIMP	53
Tabla 12. Instrucción de temporizador S_EVERZ	53
Tabla 13. Instrucción de temporizador S_EVERZ	54
Tabla 14. Instrucción de temporizador S_EVERZ	54

GLOSARIO

Automatización: La automatización es un sistema donde se transfieren tareas de producción, realizadas habitualmente por operadores humanos a un conjunto de elementos tecnológicos.

Antibióticos: es una sustancia química producida por un ser vivo o derivada sintética de ella que a bajas concentraciones mata —por su acción bactericida— o impide el crecimiento —por su acción bacteriostática— de ciertas clases de microorganismos sensibles, y que por su efecto, se utiliza en medicina humana, animal u horticultura para tratar una infección provocada por dichos gérmenes.

Autómata programable: Puede definirse como un equipo electrónico programable en lenguaje no informático y diseñado para controlar, en tiempo real y en ambiente industrial, procesos secuenciales.

Bobina: es un componente pasivo de un circuito eléctrico que, debido al fenómeno de la autoinducción, almacena energía en forma de campo magnético.

Cosmético: se utilizan para realzar la belleza del cuerpo humano. Su uso está extendido entre las mujeres y ocasionalmente entre los hombres, según los distintos tipos de culturas. La industria cosmética actual está dominada por una serie de multinacionales que surgieron a principios del siglo XX.

Cpu: unidad central de procesamiento. Es el componente en una computadora digital que interpreta las instrucciones y procesa los datos contenidos en los programas de la computadora

Eslabones: es cada uno de los anillos o elementos que forman una cadena. También se suele aplicar el término a cadenas de otro tipo, tales como alimentarias, humanas.

Farmacéuticos: son los profesionales de la salud expertos en el uso de los fármacos y sus consecuencias en el cuerpo humano

Hardware: corresponde a todas las partes físicas y tangibles de una computadora: sus componentes eléctricos, electrónicos, electromecánicos y mecánicos.

Interfaz: es conjunto de métodos para lograr interactividad entre un usuario y una computadora

Productividad: la relación entre la producción obtenida por un sistema de producción o servicios y los recursos utilizados para obtenerla.

Inyector: es un dispositivo utilizado para bombear fluidos

Microprocesador: es un circuito integrado que contiene algunos o todos los elementos necesarios para conformar una (o más) "unidad central de procesamiento" UCP, también conocido como CPU (por sus siglas en inglés: Central Process Unit). En la actualidad este componente electrónico está compuesto por millones de transistores, integrados en una misma placa de silicio.

Materia prima: son los materiales extraídos de la naturaleza que nos sirven para construir los bienes de consumo

Prototipo: Un prototipo es una representación limitada del diseño de un producto que permite a las partes responsables de su creación experimentar, probarlo en situaciones reales y explorar su uso.

Pic: es un dispositivo electrónico, programable por el usuario y destinado a gobernar máquinas o procesos lógicos y secuenciales.

Pedal: es un dispositivo que se utiliza para accionar dispositivos, electrónicos, mecánicos etc.

Pistón: Se trata de un émbolo que se ajusta al interior de las paredes de un cilindro

Relé: es un dispositivo electromecánico, que funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes.

Simulación: es la experimentación con un modelo de una hipótesis o un conjunto de hipótesis de trabajo.

Scada: es el acrónimo de Supervisory Control And Data Acquisition (Supervisión, Control y Adquisición de Datos). SCADA es un sistema basado en computadores que permite supervisar y controlar a distancia una instalación de cualquier tipo.

Software: se refiere al equipamiento lógico o soporte lógico de un computador digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica

Sensor: es un dispositivo capaz de transformar magnitudes físicas o químicas, llamadas variables de instrumentación, en magnitudes eléctricas.

Tolva: un dispositivo destinado a depósito y canalización de materiales granulares o pulverizados.

Vitaminas: son compuestos químicos que el cuerpo del ser humano necesita para llevar a cabo reacciones químicas, que permiten al organismo nutrirse en forma adecuada.

Winlog: es un paquete de software de SCADA/HMI en tiempo real, poderoso y fácil de usar, para la supervisión de plantas industriales y civiles. Un entorno de desarrollo integrado provee diferentes herramientas (Gate Builder, Template Builder, Code Builder) para la creación fácil e intuitiva de aplicaciones multilinguaje

RESUMEN

La automatización de procesos a nivel industrial tiene sentido desde el punto de vista de mejora de la productividad y de la calidad del producto final y de la cadena de procesos productivos, además que evita re-procesos que representan costos para la empresa. Con los Sistemas SCADA, se pretende aumentar la productividad, y consecuentemente aumentar los ingresos y el margen de utilidad, además de que al evitar re-procesos, se disminuye el costo de producción, teniendo así la posibilidad de vender a un precio más competitivo.

De acuerdo a lo anterior, las ventajas de diseño, simulación e implementación de un Sistema SCADA en la industria, representa para la empresa ganancias en tiempos de producción, disminución de fallas en la calidad, disminución de tiempos de re-procesos, disminución en la pérdida de materia prima, aumento de la competitividad y, por ende, se refleja un incremento notorio en el margen de utilidad.

El proyecto "IMPLEMENTACIÓN de un prototipo DE ADQUISICIÓN DE DATOS Y SUPERVISION DE CONTROL (scada) en el envasado de líquidos para la industria PRODUCTOS QUIMICOS Y FARMACÉUTICOS "; pretende evaluar el mejoramiento en los procesos productivos de las Empresa ATHOS, a partir de la simulación de un sistema SCADA.

Como primera instancia, se realizará un diagnóstico de los procesos actuales de reportes de tiempos improductivos en el llenado de envases, que se maneja en las mencionadas empresas. Ésta se hace porque es a partir de estos datos que se determinará al final, la eficiencia, eficacia y pertinencia del proyecto.

Después de esta toma de información, se comenzará con el desarrollo de la simulación a través de un algoritmo de control comunicado con PLC, por medio de PLC siemens S7 300, que manejará todo el proceso de llenado de los envases.

Simultáneamente, se irá generando la herramienta visual que utilizarían los empleados de las empresas para interactuar y controlar el proceso, a través de la interfaz WINLOG.

Finalmente, se evaluará el funcionamiento físico de los procesos de llenado y se hará la comparación de los resultados en mejoras de tiempo y efectividad, con

respecto a los datos tomados inicialmente, para demostrar, la pertinencia o no del sistema SCADA en la empresa ATHOS.

ABSTRACT

The process automation to industrial level makes sense from the point of view of improvement of the productivity and of the quality of the final product and of the chain of productive processes, besides the fact that it avoids re-processes that represent costs for the company. With the Systems SCADA, one tries to increase the productivity, and consistently to increase the income and the margin of utility, besides which on having avoided re-processes, the cost of production is diminished, having like that the possibility of selling to a more competitive price.

In agreement to the previous thing, the advantages of design, simulation and implementation of a System SCADA in the industry, it represents for the company earnings in times of production, decrease of faults in the quality, decrease of re-process times, decrease in the loss of raw material, increase of the competitiveness and, for ende, a well-known increase is reflected in the margin of utility.

The project "IMPLEMENTACIÓN DE UN PROTOTIPO DE ADQUISICION DE DATOS Y SUPERVISION DE CONTROL (SCADA) EN EL ENVASADO DE LÍQUIDOS PARA LA INDUSTRIA PRODUCTOS QUIMICOS Y FARMACEUTICOS", it tries to evaluate the improvement in the productive processes of the Empresa ATHOS, from the simulation of a system SCADA.

As the first instance, there was realized a diagnosis of the current processes of reports of unproductive times in the filling of packings. which one handles in the mentioned companies. This is done because it is from this information that one was determining ultimately, the efficiency and relevancy of the project.

After this capture of information, it was begun by the development of the simulation across an algorithm of control communicated with PLC, by means of PLC SIEMES S7 300, who were handling the whole process of filling of the packings.

Simultaneously, there will be generated the visual tool that the personnel of the companies would use to interact and to control the process, across the Interface WINLOG.

Finally, there was evaluated the physical functioning of the processes of filling and the comparison of the results would be done in improvements of time and efficiency, with regard to the information taken initially, to demonstrate, in conclusion, the relevancy of the system SCADA in the company ATHOS.

1. PRESENTACIÓN DEL TRABAJO DE GRADO

1.1. PROBLEMA

1.1.1. Planteamiento del problema Actualmente en Colombia existen aproximadamente cuatrocientas empresas productoras de cosméticos y farmacéuticos, de las cuales diez concentran el 65% de la producción.

Este sector es considerado un sector importador y exportador de medicamentos, e importador neto de principios activos, excepto de antibióticos y vitaminas.

“En 2003 el sector participó con el 3% del PIB industrial y con el 4% del empleo industrial. Los eslabones de mayor contribución en la producción fueron los detergentes y productos de aseo (23%), perfumes y lociones (16%) y jabones (14%)”¹

En Medellín encontramos el 5% de las empresas productoras de perfumes y lociones, entre ellas “Medicamentos Athos” que será el objeto de análisis, estudio e implementación, puesto que presenta baja productividad debido a su método de control de envasado de líquidos y almacenamiento de registros.

Se busca realizar el análisis y simulación del efecto y el aporte que brindaría la implementación de un sistema de este tipo en una empresa que carece de control alguno.

En la empresa Medicamentos Athos el proceso se maneja con una máquina que posee una válvula y un tanque que es llenado con el producto en proceso, la cual posteriormente será depositada en los envases establecidos, abriendo y cerrando la válvula de forma manual.

La cantidad de líquido depositado en los envases y el registro del número de elementos envasados queda en control de los operarios quienes realizan el proceso en forma manual, lo que impide garantizar la homogeneidad en la cantidad de líquido envasado.

Es por ello que se sugiere estudiar la viabilidad de un sistema SCADA, el cual brindara mejor control y la posibilidad de suplir la acción del operador tanto en la cantidad del líquido envasado como la obtención del registro de los elementos envasados, brindando así mejor rendimiento del proceso.

¹Proexport .Guía inversión extranjera [en línea] [citado en 13 de Agosto 2007], Disponible en Internet <<http://www.proexport.com.co/invest/GuiaInversionExtranjera2006/>. >

1.1.2. Formulación del problema

¿Cómo lograr mayor eficiencia en el proceso de llenado de envases de la empresa Athos?

1.2 OBJETIVOS

1.2.1 **Objetivo General:** Diseñar un sistema de Adquisición de datos y supervisión de control de la línea del producto en proceso de envasado de líquidos en máquinas llenadoras de envases, para la empresa farmacéutica y cosmética Athos.

1.2.2 Objetivos Específicos:

- Evaluar las etapas funcionales que hacen parte del proceso de llenado de envases para la industria farmacéutica y cosmética.
- Analizar el proceso actual de reportes de tiempos improductivos utilizados en el llenado de envases para la industria farmacéutica y cosmética.
- Simular el sistema Scada de llenado de envases para empresa Athos.Itda

1.3. JUSTIFICACIÓN.

En los ambientes industriales se tuvo información continuamente sobre el comportamiento de los dispositivos que integran la tarea de producción, es un elemento de gran importancia para lograr un verdadero control sobre una planta de producción, que se ve reflejado en la mejora de la productividad de los dispositivos que se van a controlar, las máquinas llenadoras facilitarán la automatización de los procesos debido a que permiten la flexibilidad para trabajar con hardware y software en tiempo real .

Por esto los beneficios de incorporar tecnologías de automatización en el campo farmacéutico y de cosméticos se verán reflejados en:

- Eliminación de contaminación por derrames de producto en el proceso.
- Flexibilización de los procesos de producción para la elaboración de diferentes productos que den respuesta a las demandas actuales y futuras de la organización.
- Aumento en la velocidad y eficiencia para la transferencia entre las etapas del producto en proceso, incrementando su productividad.
- Reducción en los tiempos de limpieza y optimización en el empleo de sistemas de apoyo crítico.
- Mejoramiento del nivel de calificación del personal que opera las tecnologías.

Este tipo de sistema SCADA posibilita integrar labores de adquisición, control, procesamiento y almacenamiento de los datos de la máquina y del proceso, brindando así la posibilidad de implementar acertadamente acciones correctivas, facilitando mejorar la eficiencia de la línea de producción.

Todo sistema que se utilice a niveles empresariales y de producción, estará sujeto al manejo humano por lo tanto, hacer de este sistema una componente eficaz, fácil de usar y amigable, indudablemente traerá beneficios directos e indirectos a la hora de su utilización.

Los sistemas SCADA no escapan a esta realidad, especialmente si se refiere a una interfaz de usuario, componente con la cual el usuario interactúa directamente con el proceso que está supervisando y controlando.

1.4. DISEÑO METODOLÓGICO.

En el desarrollo de este proyecto se empleara una metodología descriptiva y cuantitativa, usando las reglas operacionales para el manejo del software (winlog) y hardware (plc), y se basa en distintas etapas como son:

RECOPILACIÓN DE INFORMACIÓN: En esta etapa se indagan las técnicas y métodos relacionados con el llenado de envases en la industria de cosméticos y farmacéutica.

ESTUDIO DEL SISTEMA DE FUNCIONAMIENTO: Se estudiará el proceso de recopilación de tiempos improductivos del sistema.

DESARROLLO DE SOFTWARE PARA LA VISUALIZACIÓN DEL PROCESO: En esta etapa se desarrollará un entorno gráfico SCADA donde se visualizará el proceso de llenado y la productividad de tiempos del proceso.

DOCUMENTACIÓN: En esta fase se elabora la documentación del proceso y del proyecto.

1.5. PRESUPUESTO (FICHA DE PRESUPUESTO)

Las tablas siguientes muestran el presupuesto utilizado en prototipo del proyecto con sus respectivos rubros.

PRESUPUESTO GLOBAL DEL TRABAJOS DE GRADO				
RUBROS	FUENTES			TOTAL
	Estudiantes	IUE	Externa	
Personal	\$ 5.500.000	\$	\$	\$ 5.500.000
Material y suministros	\$ 280.000	\$	\$ 5.680.000	\$ 5.960.000
Salidas de campo	\$ 60.000	\$	\$	\$ 60.000
Bibliografía	\$ 60.000	\$ 120.000	\$	\$ 180.000
Equipos	\$	\$ 1.680.000	\$	\$ 1.680.000

Otros	\$ 200.000	\$	\$	\$ 200.000
TOTAL	\$ 6.100.000	\$ 1.800.000	\$ 5.680.000	\$ 13.580.000

DESCRIPCIÓN DE LOS GASTOS DE PERSONAL						
Nombre De Estudiantes	FUNCIÓN EN EL trabajo	DEDICACIÓN horas/semana	FUENTES			TOTAL
			Estudiantes	IUE	Externa	
Edisson García	Investigador	8	\$ 2.400000	\$	\$	\$2.400.000
Felipe Ramírez	Investigador	8	\$ 2.400000	\$	\$	\$2.400.000
Paula Zapata	Asesor	2	\$	\$700.000	\$	\$ 700.000
TOTAL			\$ 4.800.000	\$700.000	\$	\$5.500.000

DESCRIPCIÓN DE LAS SALIDAS DE CAMPO				
DESCRIPCIÓN DE LAS SALIDAS DE CAMPO	FUENTES			TOTAL
	Estudiantes	IUE	Externa	
Desplazamiento a empresas	\$ 60.000	\$	\$	\$ 60.000
TOTAL	\$ 60.000	\$	\$	\$ 60.000

DESCRIPCIÓN BIBLIOGRAFÍA				
DESCRIPCIÓN BIBLIOGRAFÍA	FUENTES			TOTAL
	Estudiantes	IUE	Externa	
Libros	\$	\$ 120.000	\$	\$ 120.000
Internet	\$ 60.000	\$	\$	\$ 60.000
TOTAL	\$ 60.000	\$ 120.000	\$	\$ 180.000

DESCRIPCIÓN EQUIPOS				
DESCRIPCIÓN EQUIPOS	FUENTES			TOTAL
	Estudiantes	IUE	Externa	
Computador	\$	\$1.500.000	\$	\$ 1.500.000
Multímetro	\$	\$ 180.000	\$	\$ 180.000
TOTAL	\$	\$	\$	\$ 1.680.000

DESCRIPCIÓN DEL MATERIALES Y SUMINISTROS				
Material	FUENTES			TOTAL
	Estudiantes	IUE	Externa	
Materiales de impresión	\$ 80.000	\$	\$	\$ 80.000
Elementos electrónicos	\$ 200.000	\$	\$	\$ 200.000
PLC	\$	\$	\$ 2.500.000	\$ 2.500.000
TOTAL	\$ 280.000	\$	\$ 5.500.000	\$ 5.780.000

1.6. DIAGRAMA DE GANTT FORMACION

NOMBRE DE TAREA							
* RECOPIACION DE INFORMACION				29 dias	mar 20/11/08	vier 28/12/08	Edisson Garcia-Felipe Ramirez
Vistas a las empresas de la industria farmaceutica y de cosmeticos				15 dias	mar 20/11/08	lun10/12/08	Edisson Garcia-Felipe Ramirez
identificar las etapas de llenado den envases en la industria farmaceutica y de cosmeticos				3 dias	mie26/12/08	vier 28/12/08	2 Edisson Garcia-Felipe Ramirez
* ESTUDIO DEL SITEMA DE FUNCIONAMIENTO				10 dias	lun17/1/09	jue 27/1/09	1 Edisson Garcia-Felipe Ramirez
Conocer el proceso actual de reportes inproductivos utilizados en el llenado de envases				10 dias	lun17/1/09	jue 27/1/09	Edisson Garcia-Felipe Ramirez
* DESARROLLO DE SOFTWARE PARA LA VISUALIZACION DEL PROCESO				120 dias	jue 14/1/09	vier 27/06/09	4 Edisson Garcia-Felipe Ramirez
Indagar sobre los PLC MICROLOGIIX				8 dias	mie14/1/09	mie24/01/09	1,5 Edisson Garcia-Felipe Ramirez
Indagar sobre la interfaz WINLOG				8 dias	jue 24/1/09	mie4/02/09	1,7 Edisson Garcia-Felipe Ramirez
Elaboracion del software de visualizacion				24 dias	jue5/02/09	vie6/03/09	8 Edisson Garcia-Felipe Ramirez
pruebas del software de visualizacion				5 dias	mar10/03/09	lun16/03/09	9 Edisson Garcia-Felipe Ramirez
elaboracion del software de control				20 dias	mar17/03/09	vier10/04/09	10 Edisson Garcia-Felipe Ramirez
pruebas del software de control				10 dias	lun13/04/09	vier24/04/09	11 Edisson Garcia-Felipe Ramirez
enlace entre software de visualizacion y de control				24 dias	mar27/04/09	vier29/05/09	9,12 Edisson Garcia-Felipe Ramirez
prueba del sistema SCADA				20 dias	vier29/05/09	vier26/06/09	13 Edisson Garcia-Felipe Ramirez
* DOCUMENTACION				18 dias	vier26/06/09	lun22/07/09	14,6 Edisson Garcia-Felipe Ramirez



2. APROXIMACIÓN A LOS SISTEMAS SCADA

2.1. GRUPOS DE INVESTIGACIÓN SCADA.

A continuación se nombran las empresas, grupos de investigación y personas que han hecho estudios sobre automatización mediante un sistema SCADA.

ANDIDET S.A. BOGOTA

Suministra soluciones SCADA, permitiendo la recolección, mando a distancia, administración y visualización de señales análogas, digitales. Permitiendo de esta forma la toma de decisiones en los sitios de control central.²

TELEMETRICK S.A., por su parte, fue ganador del clúster de energía y servicios públicos en la segunda convocatoria de 2005 con un proyecto que brinda soluciones para servicios de transmisión digital de datos sobre redes de telefonía móvil para controlar y supervisar remotamente procesos industriales y comerciales. Teniendo en cuenta las grandes necesidades del sector eléctrico del país y con base en los estudios realizados por el CIDET -Centro de Investigación y Desarrollo Tecnológico del Sector Eléctrico Colombiano-, y en entrevistas con ingenieros encargados del estudio, se comenzó a darle cuerpo e importancia a un proyecto que estaba siendo demandado por casi todas las empresas de distribución eléctrica del país: la tele medición de contadores de energía eléctrica. Paralelamente, Telemetrik ha estado trabajando en la iniciativa de la automatización de invernaderos para la producción de flores. "Inicialmente, la empresa concentrará sus esfuerzos en la solución integral en la automatización, control, monitoreo y supervisión remota de las variables del microclima - temperatura, humedad relativa, luminosidad y ventilación-, para las empresas del sector floricultor colombiano"³

²ANDIDET S.A. [en línea] [citado en 17 de Agosto 2007], Disponible en Internet <<http://anditel.com.co/mapa.htm>. >[

³Parque Manizales [en línea], [citado en 17 de Agosto 2007] ,Disponible en Internet <<<http://parque.manizales.unal.edu.co/parque/portal/index.php?option=content&task=view&id=55&Itemid><http://parque.manizales.unal.edu.co/parque/portal/index.php?option=content&task=view&id=55&Itemid>>>

QUIMBAYA TELEMETRIC (Quimbaya Risaralda)

Sector de petróleo y gas transporte (gasoductos y oleoductos), producción (pozos de extracción), almacenamiento y terminales de distribución. Utiliza los sistemas SCADA para recolectar y transmitir datos a lo largo de los poliductos, monitorear los patios de tanques de almacenamiento, los puntos de despacho y recolección y monitorear y controlar los campos de producción.

Sector farmacéutico Aplicaciones típicas para este segmento incluyen monitoreo de procesos, control de plantas envasadoras de líquidos.

Sector eléctrico Las aplicaciones típicas incluyen monitoreo de las centrales de generación y líneas de transmisión, control de cargas y redes de distribución. Una gran preocupación para este sector es la interrupción del suministro eléctrico y lo que puede significar para aquellas personas y empresas que dependan directamente de este servicio. Esta situación requiere de soluciones que garanticen un servicio eficiente con mínimos de tiempo de falla.⁴

2.2. GENERALIDADES DE LOS SISTEMA SCADA.

Adquisición de datos y supervisión de control SCADA proviene de las siglas de Supervisor y Control And Data Adquisición (Adquisición de datos y supervisión de control). Es una aplicación software de control de producción, que se comunica con los dispositivos de campo y controla el proceso de forma automática desde la pantalla del ordenador, a su vez proporciona información del proceso a diversos usuarios: operadores, supervisores de control de calidad, supervisión, mantenimiento, etc.

Los sistemas de interfaz entre usuario y planta basados en paneles de control repletos de indicadores luminosos, instrumentos de medida y pulsadores, están siendo sustituidos por sistemas digitales que implementan el panel sobre la pantalla de un ordenador.

El control directo lo realizan los controladores autónomos digitales y/o autómatas programables y están conectados a un ordenador que realiza las funciones de diálogo con el operador, tratamiento de la información y control de la producción, utilizando sistema SCADA.⁴

⁴ QUIMBAYA TELEMTRIC. [en línea] [citado en 3 Marzo 2007], Disponible en Internet [.<www.gutele.com/index.htm>](http://www.gutele.com/index.htm)

Funciones principales:

- Adquisición de datos para recoger, procesar y almacenar la información recibida.
- Supervisión, para observar desde un monitor la evolución de las variables de control.
- Control, para modificar la evolución del proceso, actuando bien sobre los reguladores autónomos básicos (consignas, alarmas, menús, etc.) o bien directamente sobre el proceso mediante las salidas conectadas.

2.2.1. Esquema básico

Esquema básico de un sistema de Adquisición, supervisión y control figura (1).

Figura 1 Esquema SCADA, (diagrama de bloques)

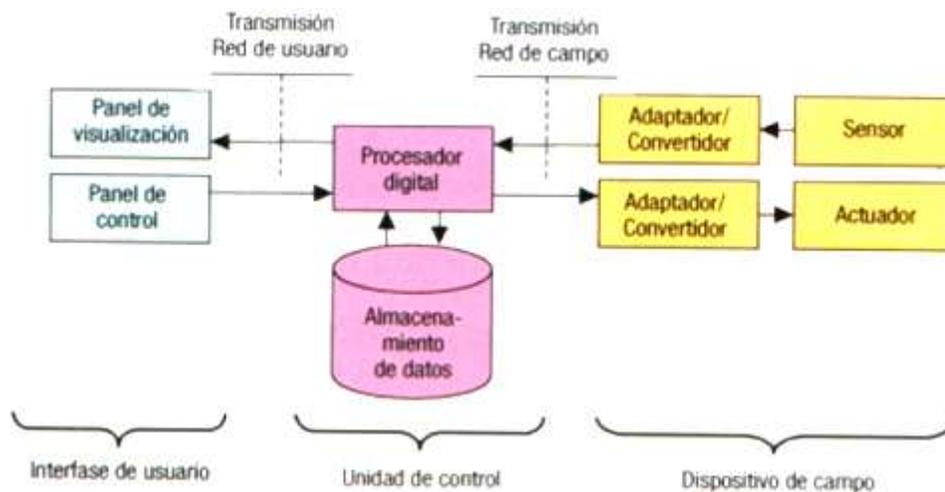
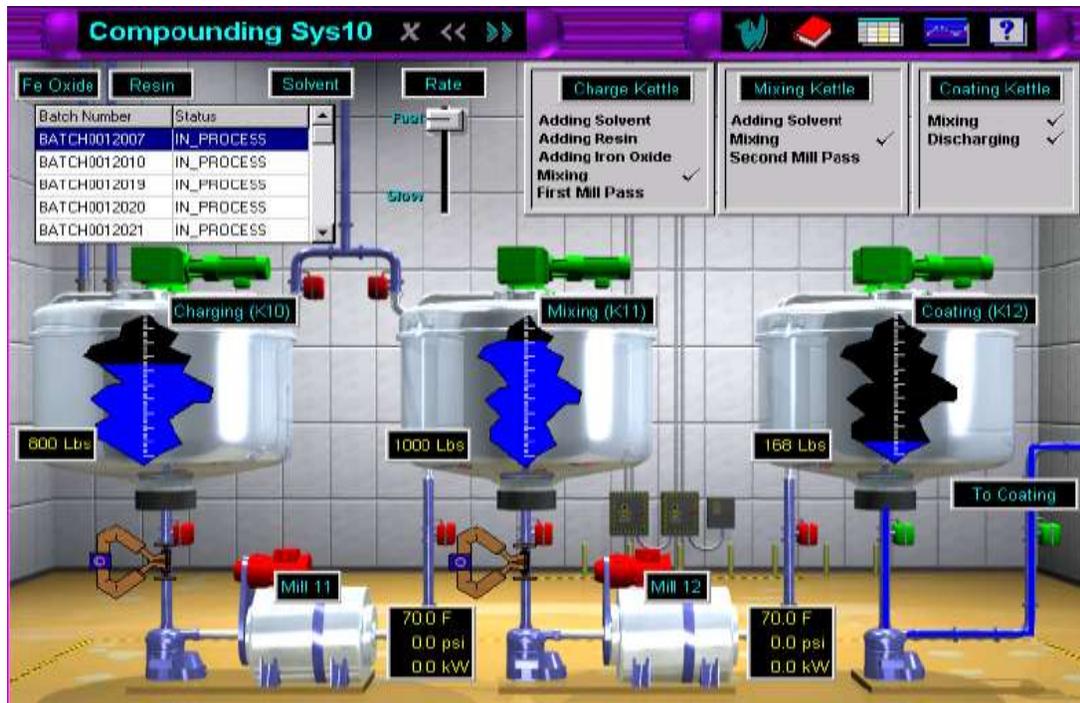


Figura 2 Ejemplo interfaz de usuario



2.2.2. MÓDULOS DE UN SISTEMA SCADA

Los módulos o bloques software son los siguientes:

- Configuración.
- Interfaz Gráfico del Operador.
- Módulo de Proceso.
- Gestión de Archivo de Datos.
- Comunicación.

Configuración: permite al usuario definir el entorno de trabajo de su SCADA, adaptándolo a la aplicación particular que se desea desarrollar.

Figura 3 configuración interfaz grafica



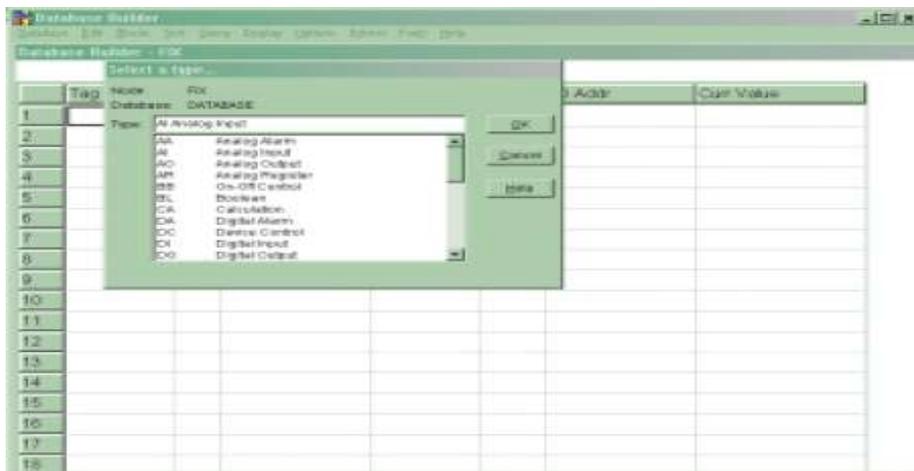
Interfaz gráfico del operador: proporciona al operador las funciones de control y supervisión de la planta. El proceso se representa mediante sinópticos gráficos.

Figura 4 ejemplo interfaz gráfica operador



Módulo de proceso: ejecuta las acciones de mando pres programados a partir de los valores actuales de variables leídas. La programación se realiza por medio de un lenguaje de bloques programación.

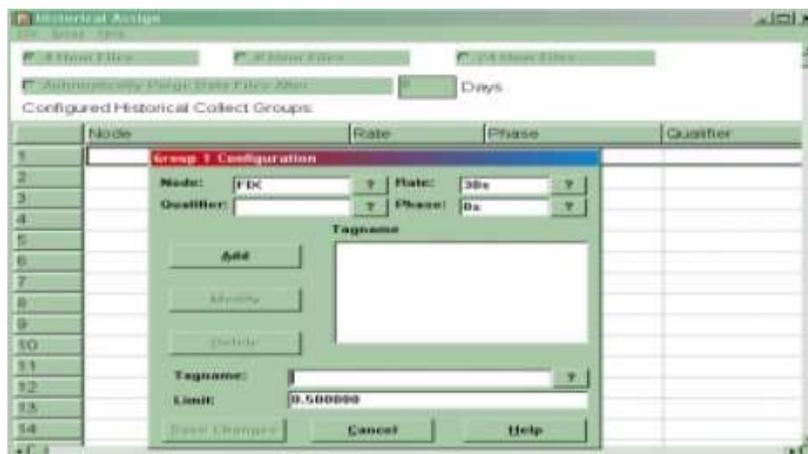
Figura 5 modelo del proceso



Gestión y archivo de datos:

Se encarga del almacenamiento y procesado ordenado de los datos, de forma que otra aplicación o dispositivo pueda tener acceso a ellos.

Figura 6 gestión y archivo de datos



Comunicaciones: se encarga de la transferencia de información entre la planta y la arquitectura hardware que soporta el SCADA, y entre ésta y el resto de elementos informáticos de gestión.

3 ASPECTOS TÉCNICOS DE LABORATORIOS ATHOS

3.1. RESEÑA HISTÓRICA DE LA EMPRESA ATHOS (PRODUCTOS QUÍMICOS Y FARMACEUTICOS)

Fue fundado el 31 de Agosto de 1.961 por el Químico Farmaceuta Alfonso Tobón Ospina, con el nombre “LABORATORIO ATHOS”, dicho nombre Honor a las iniciales de su fundador.

El Doctor Alfonso Tobón era un prestigioso Farmaceuta, en su época poseía unas importantes farmacias en el Valle de Aburra, las cuales eran La Santander # 1, #2 y #3.

Gracias al éxito que presentaron sus farmacias en el transcurso de los años vieron la posibilidad de crear un Laboratorio, el cual produce y distribuye productos Químicos y Farmacéuticos en el Departamento de Antioquia.

El Laboratorio desde su comienzo ha sido una empresa rentable, pero la filosofía de su fundador, era que solo le otorgara utilidades para cubrir sus gastos personales, sin una mentalidad de crecimiento para la empresa.

El Laboratorio desde sus inicios, fue manejado de una manera informal y sin una proyección de crecimiento y posicionamiento en el mercado, hasta el mes de Junio de 1986, cuando murió su fundador de un infarto cardiaco.

Desde ese momento, el Laboratorio pasó a propiedad del Dr. Winston Tobón Ochoa, hijo único de su fundador Alfonso Tobón.

Su nuevo apoderado no se dedicó al Laboratorio por su escaso tiempo y falta de conocimientos administrativos, debido a que en ese momento y en la actualidad, ejerce su profesión de Médico Ortopedista, entonces él vio la necesidad de

contratar un administrador para que tomara las riendas del Laboratorio y continuara con el desempeño que traía desde su comienzo.

La nueva persona que se dedicó de tiempo completo, fue el señor Efraín Carmona, que realizó una labor Administrativa pasiva y fue muy desorganizado en la planeación, dirección y control del Laboratorio, siempre presentó un balance de equilibrio, sin presentar utilidad, lo cual preocupó a su propietario.

El actual administrador, solo se limitó a continuar con los planes anteriores y no proyectó el futuro del laboratorio, el cual manejó sin criterio hasta el año de 1.994, cuando fue suspendida su labor por los pocos resultados obtenidos.

Desde entonces fue manejado por la esposa del Dr. Winston Tobón, la señora Gloria L. Carmona de Tobón, quien realizo un buen trabajo en el área comercial, pero a nivel administrativo aportó muy poco por su falta de experiencia y conocimientos administrativos.

En el año de 1.996 la dirección del Laboratorio pasó a manos del señor Carlos A. Tobón C., hijo del actual dueño. Desde ese momento al Laboratorio se le hizo una reestructuración administrativa de todos los procesos; se inicio desde cero, sin embargo se tuvo en cuenta la experiencia adquirida por la empresa en el transcurso de los años. Esta reorganización se lleva a cabo porque su actual encargado es Administración de Negocios en EAFIT y pudo aplicar los conocimientos adquiridos. El actual modo de dirección, planeación, dotación y control de todos los procesos ha arrojado muy buenos resultados en cuanto a la proyección del Laboratorio y la implementación de políticas, objetivos y valores. En los últimos meses ha presentado un crecimiento notorio en cuanto a la participación en el mercado; por este motivo se ve en la necesidad de conocer cómo está el mercado y que está esperando del Laboratorio, en cuanto a la satisfacción de necesidades en el corto, mediano y largo plazo.

ENTREVISTA con Carlos A. Tobón C, Gerente empresa ATHOS Ltda. 8 septiembre 2007, Medellin

3.2. DESCRIPCIÓN DEL PROCESO DE OPERACIÓN DE LA MÁQUINA LLENADORA DE ENVASES

En primer lugar se selecciona el tapón de succión fig. (18) que corresponde al tipo de envase a utilizar y líquido a envasar, se instala en el cilindro, se bombea la materia prima fig. (7) desde el tanque hasta la tolva fig. (12), se encienden el compresor de aire y los sensores.

Luego se procede a colocar los envases debajo del inyector fig. (13), Finalizando se presiona un pedal fig. (14), este proceso se repite continuamente en forma manual.

3.3. ETAPAS DE LLENADO DE ENVASES EN LA INDUSTRIA DE PRODUCTOS QUÍMICOS Y FARMACEÚTICOS

El proceso de llenado de envases se contituye de las siguien etapas :

Figura. 7 Llenadora de envases.

maquina utilizada actualmente para el proceso de llenado de los envases



3.3.1. Etapa de control manual de encendido y programación de la envasadora. Esta etapa se conforma por suiches mecanicos on /off, que sirven como panel de control manual. Allí encontramos el encendido de la válvula de llenado, la dosificadora , compresor de aire Figura (8)

Figura.8 Controles de donde se enciende y programa la llenadora manual mente



3.3.2. Etapa de succión de líquido. Está conformada por un pistón neumático, su funcionalidad consiste en succionar el líquido de la tolva y expulsar la materia prima hacia un cilindro de control de paso de líquido.figura (9).

Figura. 9 Pistón neumático el cual es accionado por un pedal para que se llene de la materia prima y luego espulsarlo



3.3.3. Etapa de almacenamiento de materia prima. Consiste en almacenar la materia prima ubicada en las canecas de distribución figura (10) y hacia la tolva de distribución figura (11)

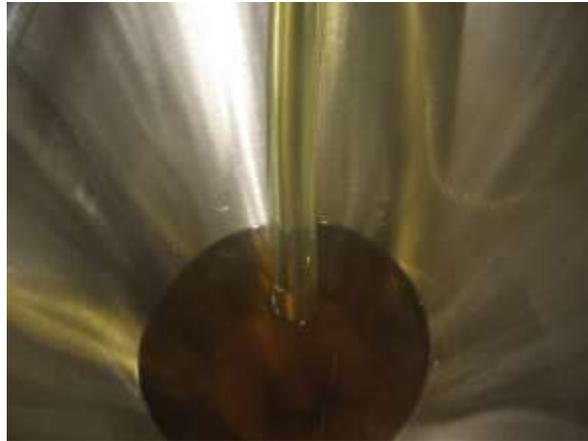
Figura. 10 bomba manual, la cual es accionada manualmente por el operador para llenar la tolva



Figura. 11 tolva de distribución, en la cual se introduce la materia prima



Figuras. 12 tolba de distribución, en la cual se introduce la materia prima



3.3.4. Etapa de envasado. En esta etapa el operador toma el envase, lo lleva a la parte donde se encuentra la inyectora y posteriormente acciona el pedal que llena el envase.

Figura. 13 Inyectora de líquido, en esta se introduce la boca del frasco para realizar el proceso de llenado



Figura. 14 pedal de accionamiento, este dispositivo es accionado por el operario cada vez que quiere llenar un envase

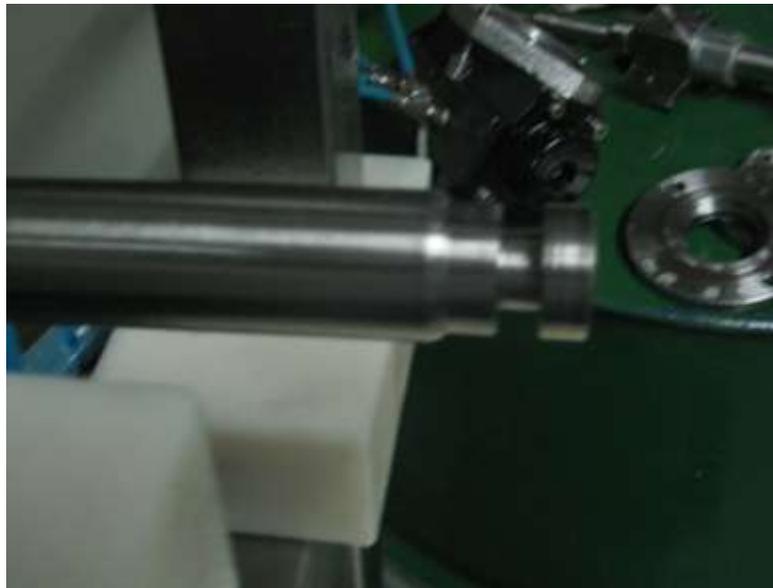


3.3.5. Etapa de sensado. Esta etapa consta de dos sensores inductivos los cuales posicionan y sensan la salida o entrada del émbolo del pistón neumático

Figura. 15 sensores inductivos, estos dispositivos son los encargados de sensar dentro del pisto la ubicación exacta del émbolo.



Figura. 16 émbolo del pistón neumático, encargado de subcionar la materia prima



3.3.6. Etapa de retención de líquido. Esta etapa se compone de dos elementos fundamentales que sirven para diferenciar el tipo de líquido que se desea envasar, teniendo en cuenta su viscosidad.

Figura.17 Cilindros de retención de líquido, en el cual se almacena temporalmente la materia prima para luego ser espulsada.



Figura. 18 Tapones de retención de líquido, estos dos tapones se utiliza de acuerdo a la viscosidad de la materia prima a envasar



Todavía se realiza el proceso de llenado de envases con una llenadora muy vieja, que consta de mangueras conectadas a un compresor de aire y otros mecanismos

para su funcionamiento los cuales resultan engorrosos a la hora de requerir alta productividad y agilidad en el producto terminado.

Figura. 19 máquina reemplazada, esta es la antigua llenadora de envases



3.4. GENERALIDADES DE LOS SENSORES

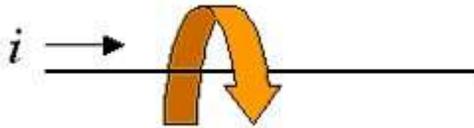
3.4.1. SENSORES INDUCTIVOS

Los sensores inductivos son una clase especial de sensores que sirven para detectar materiales metálicos ferrosos. Son de gran utilización en la industria, tanto para aplicaciones de posicionamiento como para detectar la presencia de objetos metálicos en un determinado entorno.⁶

Conceptos teóricos

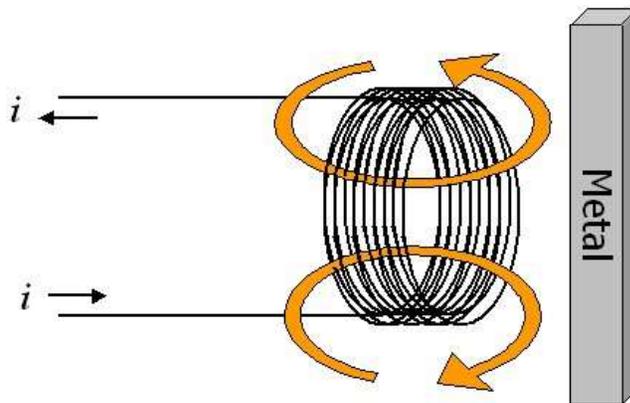
Una corriente (i) que circula a través de un hilo conductor, genera un campo magnético que está asociado a ella. (figura 20)

Figura. 20 circulación corriente



Los sensores de proximidad inductivos contienen un devanado interno. Cuando una corriente circula por el mismo, un campo magnético es generado, el cual tiene la dirección de las flechas naranjas (figura 21). Cuando un metal es acercado al campo magnético generado por el sensor de proximidad, éste es detectado.

Figura. 21 campo magnético



⁶wikipedia .sensor inductivo [en línea], [citado en 13 de Agosto 2007], Disponible en Internet <http://es.wikipedia.org/wiki/Sensor_inductivo>

La bobina del sensor inductivo induce corrientes en el material a detectar. Éstas, a su vez, generan un campo magnético que se opone al de la bobina del sensor, causando una reducción en la inductancia de la misma.

Esta reducción en la inductancia de la bobina interna del sensor, trae consigo una disminución en la impedancia de ésta.

Cuando el metal a detectar es removido de la zona de detección, el oscilador podrá generar nuevamente el campo magnético con su amplitud normal. Es en este momento en que el circuito detector nuevamente detecta este cambio de impedancia y envía una señal al amplificador de salida para que sea éste quien, nuevamente, restituya el estado de la salida del sensor.

Si el sensor tiene una configuración "Normal Abierta", éste activará la salida cuando el metal a detectar ingrese a la zona de detección. Lo opuesto ocurre cuando el sensor tiene una configuración "Normal Cerrada" Estos cambios de estado son evaluados por unidades externas tales como: PLC, Relés, PC, etc.

3.5. PROCESO DE GENERACIÓN DE REPORTE UTILIZADOS ACTUALMENTE EN EL LLENADO

Este proceso es llevado a cabo por un operario quien es el encargado del llenado de los envases, este, cuando el proceso de vaciado ha sido culminado, haciendo uso de un formato llamado orden de producto fig. (22) comienza con la descripción del producto que se está envasando en ese momento, la cantidad, lote, fecha de fabricación y demás aspectos, aspectos que generan retardos en el proceso y en general a la producción, puesto que no se puede reiniciar el proceso de llenado hasta que este reporte se encuentre listo.

Figura. 22 formatos de reportes, actualmente son llenados los reportes por el operario indicando cual fue el proceso de envasado terminado



ATHOS
Productos Químicos Farmacéuticos

LIMPIEZA Y DESINFECCION DE AREAS Y EQUIPOS

AREA: _____

EQUIPO: _____

PRODUCTO ANTERIOR: _____

FECHA: _____

DESINFECTANTE: _____

DESINFECTADO POR: _____

ORDEN DE PRODUCCION

PRODUCTO: _____

N° DE LOTE: _____

CANTIDAD: _____

REF/PRESENT: _____

CONTENIDO: _____

FECHA: _____

MATERIA PRIMA

COD	NOMBRE	N° LOTE	CANTIDAD	PESADO POR

MATERIALES

COD	NOMBRE	N° LOTE	CANTIDAD	CONTADO POR

PROCESO DE PRODUCCION

NOMBRE OPERARIOS: _____

DESPEJE DE LINEA: (VERIFIQUE)

AREA LIMPIA _____ EQUIPOS LIMPIOS _____ DOCUMENTOS DEL PROCESO ANTERIOR _____
 PRODUCTOS EN EL AREA DEL PROCESO ANTERIOR _____ MATERIALES Y/O UTENSILIOS DEL PROCESO ANTERIOR _____

HORA DE INICIO: _____ HORA DE TERMINACION _____ TIEMPO UTILIZADO: _____

SE CUMPLEN TODOS LOS PASOS DEL PROCEDIMIENTO? _____

OBSERVACIONES: _____

CONTROL DEL PESO O CONTENIDO:

--	--	--	--	--	--	--	--	--

PRODUCTO TERMINADO

CANTIDAD PROGRAMADA: _____ CANTIDAD OBTENIDA: _____ %RENDIMIENTO: _____

DEVOLUCIONES

BUENO		MALO	
DESCRIPCION	CANTIDAD	DESCRIPCION	CANTIDAD

V°B° _____ DIRECCION TECNICA FECHA: _____

3.6. PLC (CONTROLADORES LÓGICOS PROGRAMABLES)

Un PLC es una computadora especializada basada en un microprocesador que se encarga de diferentes funciones de control de muchos tipos y niveles de complejidad.

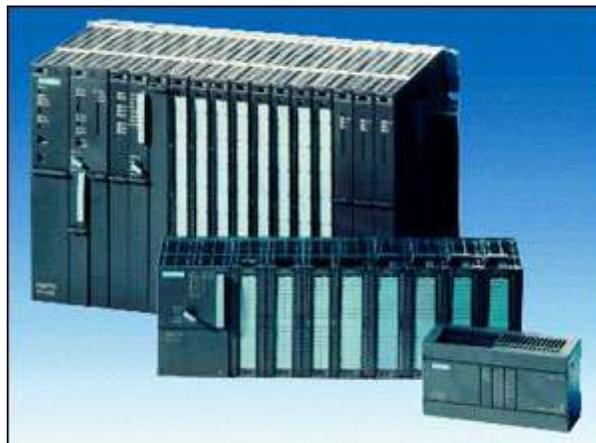
Su propósito general es monitorear parámetros cruciales de un proceso y ajustar las condiciones de operación del mismo de acuerdo a las necesidades de la empresa.

El PLC tiene la ventaja de que puede ser programado, controlado y operado por una persona aunque esta no tenga mayores conocimientos en operación de computadoras, en esencia el operador del PLC dibuja las líneas y los dispositivos en un diagrama de escalera en una pantalla, y es dibujo resultante se convierte a lenguaje de máquina posteriormente se ejecuta como un programa de usuario.

Anteriormente al PLC se le denominaba PC (Controlador Programado), posteriormente con la aparición de la computadora personal (PC) se le empezó a denominar PLC (controlador lógico programado) para evitar confusiones (figura 23)

Un PLC puede operar cualquier sistema que posea dispositivos de salida digitales o analógicos.

Figura. 23 PLC: SIEMENS S7 200 Y 300



3.6.1. HISTORIA DE LOS PLC. Los primeros sistemas de PLC aparecieron a finales de los años 60 y principios de los años 70. Estos se crearon inicialmente para la industria automotriz. Tradicionalmente, las plantas de automóviles debían apagarse por completo al momento de cambiar el modelo en producción. Los primeros modelos de PLC, que se implementaron al mismo tiempo que otras técnicas de automatización, lograron reducir el tiempo que la planta debía estar detenida. La reprogramación del PLC por medio de un teclado reemplazó a un panel lleno de cables en donde se podían observar relés, temporizadores y otros componentes, la primera de estas formas resulta mucho más fácil y rápida.

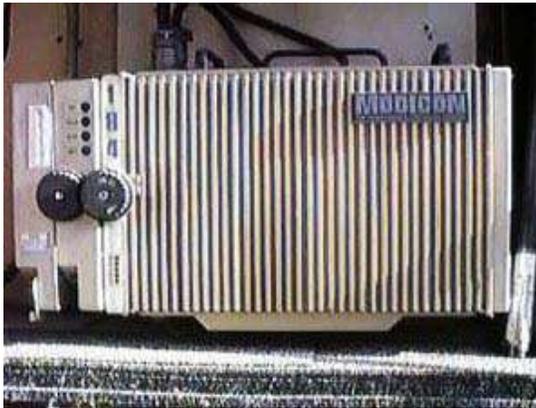
El primer controlador lógico programable fue el MODICON (Modular Digital Controller); el modelo 084 fue desarrollado por Dick Morley en 1969. (Figura 24) podemos ver una ilustración de uno de estos controladores.⁷

Estos primeros autómatas presentaban problemas con los procedimientos de reprogramación, debido a que los programas eran complicados y para realizar los cambios se requería un programador con experiencia. A través de la década de 1970, se realizaron mejoras a estos dispositivos para hacerlos más amigables con el operador. En 1978 se le introdujo un chip microprocesador que incrementó la capacidad de automatización del sistema y bajó los costos; además se desarrollaron programas más accesibles para las personas en general, escritos en lenguaje de alto nivel.

En la década de 1980 se incrementó considerablemente el uso de los PLC debido a sus mejores características y se extendieron sus aplicaciones a diversas áreas como generación de energía y sistemas de seguridad.

En los años 1990 también se empezó a explotar su uso en hogares, equipo médico, y sus aplicaciones siguen aumentando.

Figura. 24 PLC MODICON



⁷Electronica Unicrom. Historia del plc [en línea],[citado en 11 de Agosto 2007], Disponible en Internet <http://www.unicrom.com/art_historia_PLC.asp>

3.6.2. Ventajas – Desventajas

Algunas de las ventajas de usar un PLC son las siguientes:

Flexibilidad: un modelo de PLC puede controlar varias máquinas diferentes, cada una con su programa propio; a su vez no se requiere un controlador para cada una como ocurría anteriormente.

Implementación de cambios y corrección de errores: cuando se decide cambiar un programa o parte de él, se puede realizar desde un dispositivo de entrada (teclado) en solo unos minutos; anteriormente se requería realizar de nuevo el alambrado del panel de control, simplemente no le permitía al usuario hacer alguna modificación en el programa.

Bajo costo: el incremento de la tecnología ha posibilitado introducir mayor cantidad de funciones en espacios más reducidos y más baratos.

Prueba piloto: el PLC programado puede ser probado para evaluarlo en condiciones de laboratorio, de esta forma se realizan las modificaciones necesarias para que trabaje correctamente en el campo.

Velocidad de operación: la velocidad de operación de un programa de PLC es bastante rápida, en general está determinada por el tiempo de escaneo de los datos, que ronda los milisegundos.

Seguridad: un cambio en el programa del PLC no puede hacerse a menos que éste sea debidamente intervenido; en los sistemas de relés se podían realizar cambios fácilmente sin documentación presentando inconvenientes como olvidarlos fácilmente.

Estas son sólo algunas de las ventajas que presentan los PLC, aunque pueden derivarse otras de las aplicaciones en un determinado campo específico.

Algunas desventajas en el uso de estas herramientas son:

Tecnología nueva: puede resultar difícil cambiar el modo de pensar de algunas personas de los relés a un concepto más computarizado como el PLC. Sin embargo, actualmente el uso de las computadoras está bastante extendido en hogares y oficinas, además de en la industria, sin embargo la gente se apodera de este campo.

Aplicaciones de programas fijos: algunas aplicaciones son de función simple, es decir, su operación no se cambia nunca o casi nunca. No es una buena inversión un PLC con su capacidad de manejo de múltiples programas y facilidad de variar los mismos, si no se va a utilizar sus ventajas y/o características.

Consideraciones ambientales: las condiciones ambientales en que se desarrollan ciertos procesos, como alta temperatura y vibraciones, producen interferencias en los dispositivos electrónicos del PLC.

Operación segura en fallas: en los sistemas de relés, una falla en la alimentación detiene el sistema, y el relé no se restaura automáticamente cuando este acaba. Esto se puede evitar programando en un PLC, sin embargo, algunos programas requieren alimentación para detener el dispositivo. Esta desventaja se puede resolver agregando relés de seguridad al sistema del PLC.

3.6.3. CAMPOS DE APLICACIÓN. El PLC por sus especiales características de diseño tiene un campo de aplicación muy extenso. La constante evolución del hardware y software amplía constantemente este campo para poder satisfacer las necesidades que se detectan en el espectro de sus posibilidades reales.

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.

Sus reducidas dimensiones, la extremada 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 en que se producen necesidades tales como:

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

Ejemplos de aplicaciones generales:

- Maniobra de máquinas
- Maquinaria industrial de plástico
- Máquinas transfer
- Maquinaria de embalajes
- Maniobra de instalaciones
- Maniobra de aire acondicionado, calefacción etc.
- Instalaciones de seguridad

- Señalización y control
- Chequeo de programas
- Señalización del estado de procesos

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. Tales consideraciones permiten hablar de las ventajas que proporciona 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 suficientemente grande.
- Posibilidad de introducir modificaciones sin cambiar el cableado ni añadir periféricos.
- Mínimo espacio de ocupación.
- 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 cableado.
- Si por alguna motivo la máquina queda fuera de servicio, el autómata sigue siendo útil para otra máquina o sistema de producción.

Inconvenientes

Como inconvenientes podríamos hablar:

- Hace falta un programador lo que obliga a adiestrar a uno de los técnicos.
- El costo inicial también puede ser un inconveniente.

3.6.4. FUNCIONES BÁSICAS DE UN PLC.

- Detección: lectura de los capacitores distribuidos por el sistema de fabricación.
- Mando: elaborar y enviar las acciones al sistema mediante los accionadores y preaccionadores.
- Diálogo hombre máquina: mantener un diálogo con los operarios de producción, obedeciendo sus consignas e informando del estado del proceso.
- Programación: para introducir, elaborar el programa de aplicación del autómeta, el diálogo de programación debe modificar el programa incluso con el autómeta controlando la máquina.
- 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.
- Sistema 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 serial 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 controladores PID que están programados en el autómeta.
- Entradas/salidas distribuidas: los módulos de entrada/ salida no tienen porque estar en el armario de autómeta, puede estar distribuidos por la instalación. Se comunican con la unidad central del autómeta mediante un cable de red. ⁸

3.6.5. ESTRUCTURA GENERAL DEL PLC

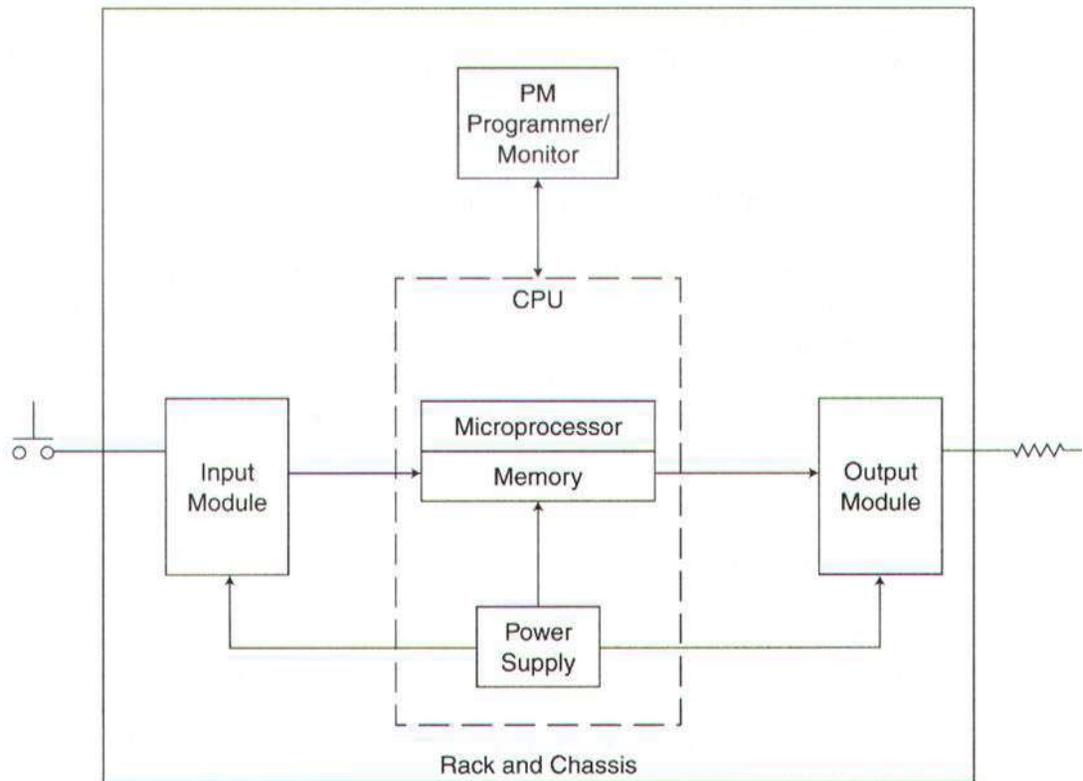
Las unidades principales de un PLC y la forma como están interconectadas figuras (26) son:

1. Unidad de procesamiento central (CPU): es el cerebro del sistema y tiene tres sub-partes:
 - Microprocesador: se encarga de las operaciones lógicas y matemáticas.

- Memoria: es el área de la CPU en la cual se almacenan los datos y la Información; también contiene el software del sistema y el programa de Usuario.
 - Fuente de alimentación: fuente eléctrica que transforma la alimentación AC a los diversos valores DC que el sistema requiere para operar.
2. Programador/monitor: es el dispositivo usado para comunicarse con los circuitos del PLC; puede ser una terminal de mano, terminales industriales o computadoras personales.
 3. Módulos I/O: el módulo de entradas tiene terminales para conectar las señales eléctricas de dispositivos exteriores, como sensores o transductores.
 4. Anaquel y bastidor: el anaquel en el cual las partes del PLC son ensambladas y la “caja” en la cual se montan los diferentes módulos del PLC. Algunos componentes opcionales del sistema del PLC son:
 - Impresora
 - Grabador/reproductor de programa: algunos sistemas antiguos utilizaban cintas como dispositivos de almacenamiento secundario. Actualmente se utilizan discos flexibles y otros medios más modernos de almacenamiento secundario.

⁸Grupo-master. Funciones básicas de un plc [en línea], [citado en 11 de agosto 2007], Disponible en Internet < http://www.grupo-maser.com/PAG_Cursos/Auto/auto2/auto2/PAGINA_PRINCIPAL/PLC/plc.htm >

Figura. 25 Diagramas en bloques de la estructura de un PLC.



3.6.6. CPU

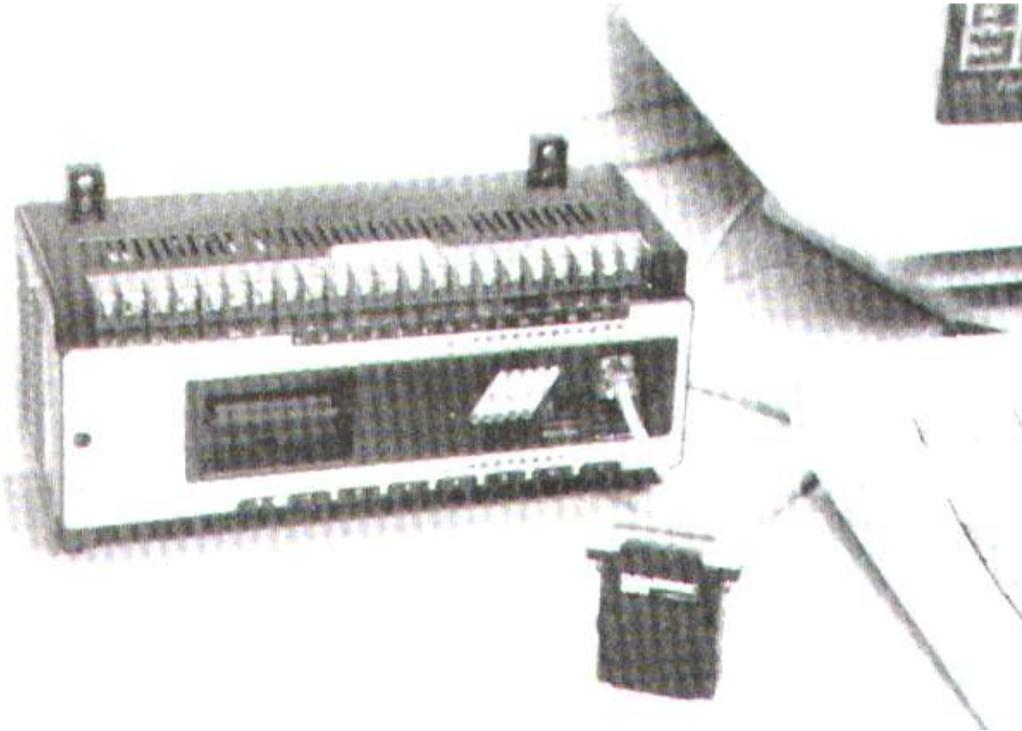
La CPU (Central Processing Unit) es la parte inteligente del autómat. Interpreta las instrucciones del programa de usuario de forma secuencial y consulta el estado de las entradas. Dependiendo de dichos estados y del programa, ordena la activación de las salidas deseadas. La capacidad de cálculo y la velocidad de procesamiento dependen del número y 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.⁹

La CPU está constituida por los siguientes elementos:

- Procesador
- Memoria monitor del sistema
- Circuitos auxiliares

⁹Electronica Unicrom, CPU plc [en línea],[citado en 20 de agosto 2007], Disponible en Internet http://www.unicrom.com/tut_PIC2.asp >

Figura. 26 Unidad de procesamiento central del PLC



Algunas unidades tienen facilidades para agregar o expandir la memoria; otras no las tienen. Las CPU también cuentan con varios puertos para conectar los cables que van a los otros módulos, si es un PLC modular; aunque también existen PLC integrados, que traen todas las partes en un único módulo. El sistema operativo básico se almacena en una memoria de sólo lectura (ROM), que no se borra cuando ocurre una falla en la alimentación. Los CPU tienen interruptores de operación, algunos de los cuales requieren una llave para prevenir los cambios que intente el personal no autorizado. Algunas de las posiciones de estos interruptores son:

Apagado: el sistema no puede trabajar ni ser programado

Trabajando: permite al sistema trabajar, pero no se pueden alterar los programas

Programa o apagado/programa: el sistema no puede trabajar, pero puede ser programado o reprogramado

Trabajando/programa: el sistema puede trabajar y se pueden realizar modificaciones mientras está trabajando

Deshabilitado: todas las salidas están apagadas o en estado de no-operación.

3.6.7. PROGRAMADOR/MONITOR (PM). También se le conoce como dispositivos de interfaz humano/máquina (HMI). Este se conecta al CPU mediante un cable; después de la programación del CPU, el PM por lo general no es tan necesario para la operación del sistema y puede ser removido; se puede utilizar el mismo PM para diversos dispositivos, más si se toma en cuenta que algunos fabricantes actuales tienen dispositivos especialmente hechos para transportarlos y utilizarlos en el lugar donde se encuentra el PLC, o HMI portátiles ¹⁰.

3.6.8. MÓDULOS DE ENTRADA SALIDA. Los elementos de entrada y salida son los que permiten comunicar el autómata con el proceso que está controlando y con el operador.

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

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: pre actuadores o actuadores (lámparas, relés, etc.).

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

- Digitales o binarias.

- Analógicas.

¹⁰Grupo tecnológico, módulos entrada salida [en línea],[citado en 23 de agosto 2007] Disponible en Internet <http://www.grupo-maser.com/PAG_Cursos/Auto/auto2/auto2/PAGINA%20PRINCIPAL/PLC/ESTRUCTURAS/ESTRUCTURA%20INTERNA/INTERFACES/interfaces.htm>

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 V. 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 V.CC (el más común), 110 V.CA o 220 V. CA. 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 opto acoplador.
- 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 accionadores de mando analógico y lean señales de este tipo como pueden ser la (temperatura, la presión o el caudal), es decir, admiten 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), ya 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 (periodo 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 leer los niveles de voltaje entre 0a 10v,- 10 a 10v o de corriente entre -20 a 20 mA, 0 a 20 mA o 4 a 20 mA.

El proceso de adquisición de una 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 autómata 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 autómata) 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 suministrar tensión, solo pueden actuar sobre elementos que trabajan todos a la misma tensión, en cambio los módulos de salida electromecánicos (relés), al ser libres de tensión, pueden actuar sobre elementos que trabajen a tensiones distintas. Estas salidas estaticas carecen de velocidad de respuesta ya que tiene un periodo muerto de conmutación 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 autómata se convierta en voltaje o en corriente. Los rangos típicos de esta salida 0 a 10 V., -10 a 10 V., 0 a 20 mA., -20 a 20 mA.y 4 a 20 mA.

Lo que realiza es una conversión D/A (digital/analógica), puesto que el autómata solo trabaja con señales digitales. Esta conversión se realiza con una precisión o

resolución determinada (número de bits) y cada cierto intervalo de tiempo (periodo muestreo). La precisión suele ir desde los 12 a los 14 bits.

Este voltaje o corriente 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 autómeta realizar funciones de regulación y control de procesos continuos.

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

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

3.6.9. FUENTE DE ALIMENTACIÓN

La fuente de alimentación proporciona los voltajes necesarios para el funcionamiento de los distintos circuitos del sistema.

La alimentación a la CPU puede ser continua a 24 V.CC (nuestro caso), o alterna a 110/220 V.CA. En cualquier caso es la propia CPU la que alimenta las interfaces conectadas a través del bus interno.

La fuente de alimentación del autómeta puede incorporar una batería de reserva, 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 autómeta.

Interfaces

En el control de un proceso automatizado, es imprescindible un diálogo entre operador-máquina junto con una comunicación entre la máquina y el autómeta, 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:

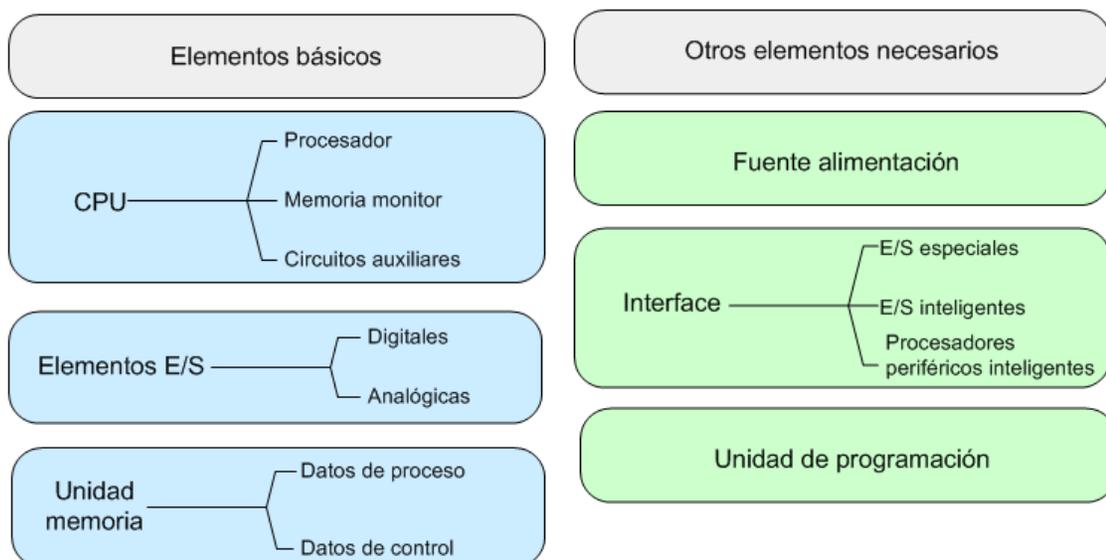
- **Entradas/salidas especiales:** se caracterizan por no influir en las variables de estado del proceso de automatización. Únicamente se encargan de adecuar las E/S, para que puedan ser inteligibles por la CPU, para que puedan ser interpretadas correctamente por los sensores o actuadores (motores, cilindros, etc.), que entregan o reciben señales ya sea por su forma o por su aplicación.
- **Entradas/salidas inteligentes:** admiten múltiples modos de configuración, por medio de unas combinaciones binarias situadas en la misma tarjeta. Lo que permite implementar lazos de control on/off.
- **Procesadores periféricos inteligentes:** son módulos que incluyen su propio procesador, memorias y puntos auxiliares de entrada/salida. Es decir con programas dedicados en su memoria, parametrizado por el usuario, 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. Ejemplo control PID ¹¹

¹¹ Josep Balcells ,Automatas programables [en línea],[citado en 22 de septiembre 2007]Disponible en Internet <http://books.google.com/books?id=xfSjADqe70C&printsec=frontcover&hl=es#v=onepage&q&f=false>

3.7. UNIDAD DE PROGRAMACIÓN

Es el conjunto de medio, tanto 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.

Figura. 27 Resumen de la partes del PLC



3.7.1 UNIDAD DE PROGRAMACIÓN GENERALIDADES CONTROLADOR LÓGICO PROGRAMABLE (PLC) SIMATIC S7300

El controlador lógico programable (PLC) utilizado para la realización de este proyecto es un PLC **SIMATIC S7 -300**.

SIMATIC S7 es un producto de la empresa SIEMENS y responde a la serie actual de los controladores lógicos programables.

Figura. 28 PLC SIMATIC S7 -300. CPU 313C



Tabla. 1 características técnicas PLC SIMATIC S7 -300. CPU 313C

CARACTERÍSTICAS TÉCNICAS	CPU 313C
Entradas/Salidas integradas	24 DI / 16 DO
Máx. nº E/S mediante EM´s	1024
Máx. nº canales analógicos	69 (7 integradas)
Memoria de programa/datos	32 KB / 10 KB
Tiempo de ejecución/instrucción.	0,1 µs
Marcas/Contad./Temporizadores.	2.048/256/256
Reloj en tiempo real	Integrado
Salidas de impulsos	1 x RS 485
Puerto de comunicaciones	MPI, AS-I – Profibus –Ethernet

3.7.2. UNIDAD DE PROGRAMACIÓN PUERTOS DE COMUNICACIONES

A continuación se observa la lista de los módulos de comunicación S7 que se pueden conectar a las distintas subredes. Además se indican los servicios que soporta cada uno de dichos módulos.

SUBREDES: para satisfacer las distintas exigencias de los niveles de automatización (control central, célula, campo, actuador/sensor) SIMATIC ofrece las subredes siguientes:

- AS- INTERFAS: El AS- Interface o Actuador –sensor interface es un sistema de conexión para el nivel inferior de la pirámide de automatización, el nivel de campo. Sirve especialmente para interconectar sensores y actuadores binarios. La cantidad de datos está limitada a un máximo de 4 bits por esclavo.
- MPI : la sub red MPI se utiliza para los niveles de campo y de célula de poca extensión. La red MPI está constituida por un interface multipunto integrado en los SIMATIC S7 y está concebida como puerto de programación de PG, así como para interconectar un número reducido de CPUs que intercambian pequeñas cantidades de datos (hasta 70 bytes).
- PROFIBUS: PROFIBUS constituye la red para el nivel de campo y de célula en el sistema de comunicación abierto y no propietario de SIMATIC. PROFIBUS es simplemente apropiado para una transferencia rápida de cantidades medianas de datos (aproximadamente 200 bytes).
- INDUSTRIA ETHERNET: la industria ETHERNET es la red para el nivel de control central y el de la célula en el sistema de comunicación abierto y no propietario de SIMATIC. Industria Ethernet es especialmente apropiado para una transferencia rápida de grandes cantidades de datos.
- CONEXIÓN PUNTO A PUNTO: Una conexión punto a punto no es una subred en el sentido tradicional de la palabra .Dicha conexión se realiza en SIMATIC a través de procesadores de comunicación punto a punto, conectándose (PLC, scanner, PC, etc.)

4. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA SCADA

4.1 SOFTWARE PARA LA PROGRAMACIÓN DEL PLC SIMATIC S7 -300

Para la programación del PLC utilizado en nuestro caso utilizamos el software STEP 7.

4.1.2 Requisitos para la instalación

Sistema operativo - exigencias mínimas:

	Procesador	RAM	Gráficos
MS Windows 2000 Professional	600MHz	256 MB*)	XGA 1024x768 16 Bit profundidad de color
MS Windows XP Professional	600MHz	256 MB*)	XGA 1024x768 16 Bit profundidad de color

*) Se recomienda una memoria principal de al menos 512 MB

Si trabaja con configuraciones hardware (HW Config) o proyectos STEP7 que sean amplios que contengan por ejemplo varios cientos de módulos (CPU, esclavos, periferia.), será necesario utilizar una unidad de programación (PG) o un PC de características estándar.

En caso de utilizar un PC, necesitará adicionalmente un programador de PROM externo para las tarjetas de memoria (Memory Cards / MMC) SIMATIC, si desea guardar sus programas de usuario en EPROMs, así como una interfaz SIMATIC NET, si desea ejecutar en su PC las funciones online que acceden al sistema de automatización.

4.1.3 Lenguaje de programación

El software de programación step7 de Siemens permite, mediante las instrucciones del autómeta, desarrollar el programa de usuario que posteriormente, se transfiere a la memoria de programa de usuario en este caso al Plc.

La programación del autómeta consiste en el desarrollo de una sucesión de instrucciones ordenadas, escritas en un lenguaje de programación concreto. Estas instrucciones están disponibles en el sistema de programación y resuelven el control de un proceso determinado.

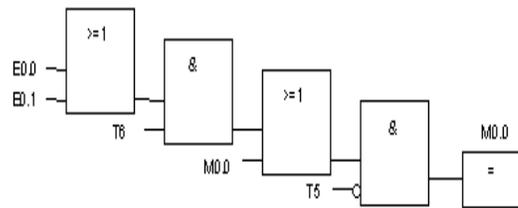
Cuando hablamos de los lenguajes de programación nos referimos a las diferentes formas en que se puede escribir el programa usuario. Para nuestro caso el software básico STEP7 contiene los lenguajes de programación KOP, FUP, AWL. Los demás lenguajes de programación pueden adquirirse opcionalmente; en este caso se trabajará sobre el lenguaje KOP.

4.1.4 Lenguaje de programación Fup (diagrama de funciones)

El lenguaje de programación FUP (diagrama de funciones) utiliza los símbolos gráficos del álgebra booleana para representar la lógica. También es posible representar en conexión directa con los cuadros lógicos funciones complejas, como por ejemplo funciones matemáticas.

Figura. 29 Ejemplo de un segmento en FUP

Segmento 1: fase verde para peatones



4.1.5 Lenguaje de programación AWL (lista de instrucciones)

El lenguaje de programación AWL (lista de instrucciones) es un lenguaje textual orientado a la máquina. Las diversas instrucciones equivalen a los pasos de trabajo con los que la CPU ejecuta el programa. Las instrucciones pueden ser agrupadas en segmentos.

Figura. 30 Ejemplo de un segmento en AWL

```
Segmento 1: control de la válvula de descarga
U(
O
O #Bobina
)
UN #Cerrar
= #Bobina

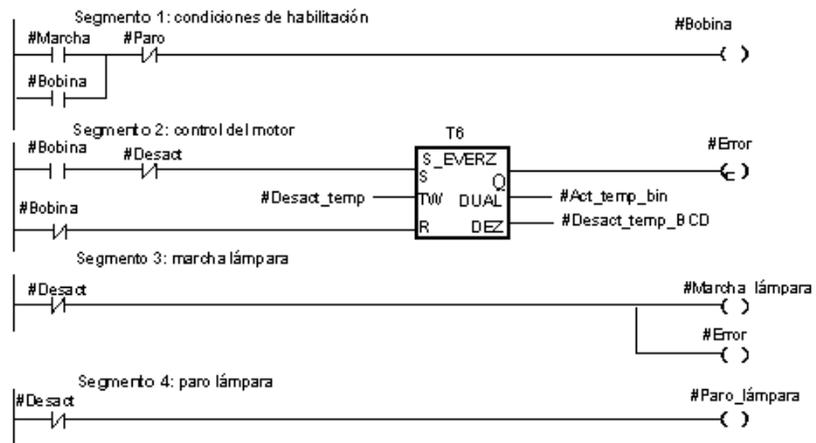
Segmento 2: indicación "válvula abierta"
U #Bobina
= #Indic_abierta

Segmento 3: indicación "válvula cerrada"
UN #Bobina
= #Indic_cerrada
```

4.1.6 Lenguaje de programación KOP (esquema de contactos)

La representación del lenguaje de programación gráfico KOP (esquema de contactos) es similar a la de los esquemas de circuitos. Los elementos de un esquema de circuitos, tales como los contactos normalmente cerrados y normalmente abiertos, se agrupan en segmentos. Uno o varios segmentos constituyen el área de instrucciones de un bloque lógico.

Figura. 31 Ejemplo de un segmento en kop.

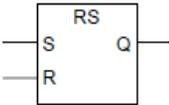


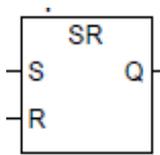
En este trabajo nos centraremos en el lenguaje de programación gráfico KOP.

A continuación enunciaremos instrucciones que intervienen en el lenguaje gráfico KOP.

Se dispone de las operaciones lógicas:

Tabla. 2 instrucciones de operaciones lógicas

SÍMBOLO	NOMBRE	DESCRIPCIÓN
--- ---	Contacto normalmente abierto	--- --- (Contacto normalmente abierto) se cierra si el valor del bit consultado, que se almacena en el <operando> indicado, es "1". Si el contacto está cerrado, la corriente fluye a través del contacto y el resultado lógico (RLO) es "1".
--- / ---	Contacto normalmente cerrado	--- / --- (Contacto normalmente cerrado) se abre si el valor del bit consultado, que se almacena en el <operando> indicado, es "0". Si el contacto está cerrado, la corriente fluye a través del contacto y el resultado lógico (RLO) es "1".
XOR	O-exclusiva	XOR (O-exclusiva) genera un RLO de "1" si el estado de señal de los dos bits indicados es distinto.
--- NOT ---	Invertir resultado lógico (RLO)	--- NOT --- (invertir resultado lógico) invierte el bit RLO.
---()	Bobina de relé, salida	---() (Bobina de relé (salida)) opera como una bobina en un esquema de circuitos. Si la corriente fluye hasta la bobina (RLO = 1), el bit en el <operando> se pone a "1". Si no fluye corriente hasta la bobina (RLO = 0), el bit en el <operando> se pone a "0".
---(R)	Desactivar salida	---(R) (Desactivar salida) sólo se ejecuta si el RLO de las operaciones anteriores es "1" (flujo de corriente en la bobina). Si fluye corriente a la bobina (RLO es "1"), el <operando> indicado del elemento se pone a "0".
---(S)	Activar salida	---(S) (Activar bobina) sólo se ejecuta si el RLO de las operaciones anteriores es "1" (flujo de corriente en la bobina). Si el RLO es "1", el <operando> indicado del elemento se pone a "1".
	RS Activar flip-flop de desactivación	RS (Activar flip-flop de desactivación) se desactiva si el estado en la entrada R es "1" y si el estado en la entrada S es "0". De no ser así, cuando el estado en la entrada R es "0" y el estado en la entrada S es "1", se activa el flip-flop. Si el RLO es "1" en ambas entradas, la operación Desactivar flip-flop de activación ejecuta en el <operando> indicado primero la operación Desactivar y seguidamente la operación Activar, de modo que la dirección permanece activada para el resto del ciclo de programa.
	SR Desactivar flip-flop de activación	SR (Desactivar flip-flop de activación) se activa si el estado en la entrada S es "1" y si el estado de la entrada R es "0". De no ser así, cuando el estado en la entrada S es "0" y el estado de la entrada R es "1", se desactiva el flip-flop. Si el RLO es "1" en ambas entradas, la operación Desactivar flip-flop de activación ejecuta en el <operando> indicado primero la operación Activar y seguidamente la operación Desactivar, de modo que la dirección permanece desactivada para

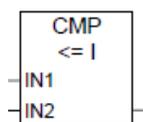


el resto del ciclo de programa.

Se dispone de operaciones de comparación: Las operaciones comparan las entradas IN1 e IN2 según los tipos de comparación

Tabla. 3 instrucciones operaciones de comparación

SÍMBOLO	NOMBRE	DESCRIPCIÓN
	Igual	
	Diferente a	
	Mayor que	
	Menor que	
	Mayor o Igual	



Menor o Igual

4.2 Lista de operaciones de contaje

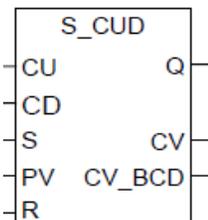
Para poner el contador a un valor determinado hay que introducir un número de 0 a 999, por ejemplo 127, en el siguiente formato: C# 127. C# sirve para indicar el formato decimal codificado en binario.

- **ZAEHLER Parametrizar e incrementar/decrementar contador**

Tabla. 4 característica instrucción ZAEHLER

PÁRAMETROS	TIPO DE DATOS	DESCRIPCIÓN
N. ° de C	COUNTER	Número de identificación del contador; el área varía según CPU utilizada
CU	BOOL	Entrada de contaje adelante
CD	BOOL	Entrada de contaje atrás
S	BOOL	Entrada para predeterminar el contador
PV	WORD	Valor numérico introducido en forma de C#<valor> en el margen comprendido entre 0 y 999
PV	WORD	Valor para inicializar el contador
R	BOOL	Entrada de puesta a 0
CV	WORD	Valor actual del contador, número Hexadecimal
CV_BCD	WORD	Valor actual del contador, número BCD
Q	BOOL	Estado del contador

SÍMBOLO

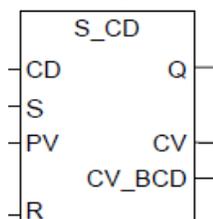


- **Z_RUECK Parametrizar y decrementar contador**

Tabla.5 característica instrucción Z_RUECK

PARÁMETROS	TIPO DE DATOS	DESCRIPCIÓN
N. ° de C	COUNTER	Número de identificación del contador; el área varía según CPU utilizada
CD	BOOL	Entrada de contaje atrás
S	BOOL	Entrada para predeterminar el contador
PV	WORD	Valor numérico introducido en forma de C#<valor> en el margen comprendido entre 0 y 999
PV	WORD	Valor para inicializar el contador
R	BOOL	Entrada de puesta a 0
CV	WORD	Valor actual del contador, número Hexadecimal
CV_BCD	WORD	Valor actual del contador, número BCD
Q	BOOL	Estado del contador

SÍMBOLO



- **Poner contador al valor inicial**

Tabla. 6 instrucción (SZ) (poner contador inicial)

PARÁMETROS	TIPO DE DATOS	DESCRIPCIÓN
N. ° de C	COUNTER	Número del contador a predeterminar
<Valor predeterminado>	WORD	Valor para la preselección BCD (0-999)
SÍMBOLO	---(SZ)	(SZ) (Inicializar el contador) se ejecuta solamente en caso de que haya un flanco ascendente en el RLO. En este caso se transmite el valor predeterminado al contador indicado.

- **Incrementar contador**

Tabla. 7 instrucción (ZV) (incrementar contador)

PARÁMETROS	TIPO DE DATOS	DESCRIPCIÓN
N. ° de C	COUNTER	Número específico del contador; el área varía según la CPU utilizada
<Valor predeterminado>	WORD	Valor para la preselección BCD (0-999)
SÍMBOLO	---(ZV)	(ZV) (Contar adelante) incrementa en "1" el valor del contador indicado si hay un

flanco ascendente en el RLO y el valor del contador es menor que "999". El valor del contador no se altera si no hay ningún flanco ascendente, ni tampoco en caso de que el contador tenga ya el valor "999".

- **Decrementar contador**

Tabla. 8 instrucciones (ZR) (Decrementar contador)

PARÁMETROS	TIPO DE DATOS	DESCRIPCIÓN
N. ° de C	COUNTER	Número específico del contador; el área varía según la CPU utilizada
SÍMBOLO	---(ZR)	(ZR) (Contar atrás) decreenta en "1" el valor del contador indicado si hay un flanco ascendente en el RLO y el valor del contador es mayor que "0". El valor del contador no se altera si no hay ningún flanco ascendente, ni tampoco en caso de que el contador tenga ya el valor "0".

Lista de operaciones de temporización

En STEP 7 podemos encontrar tipos de instrucciones de temporizadores las cuales son:

Figura. 32 lista de operaciones de temporización

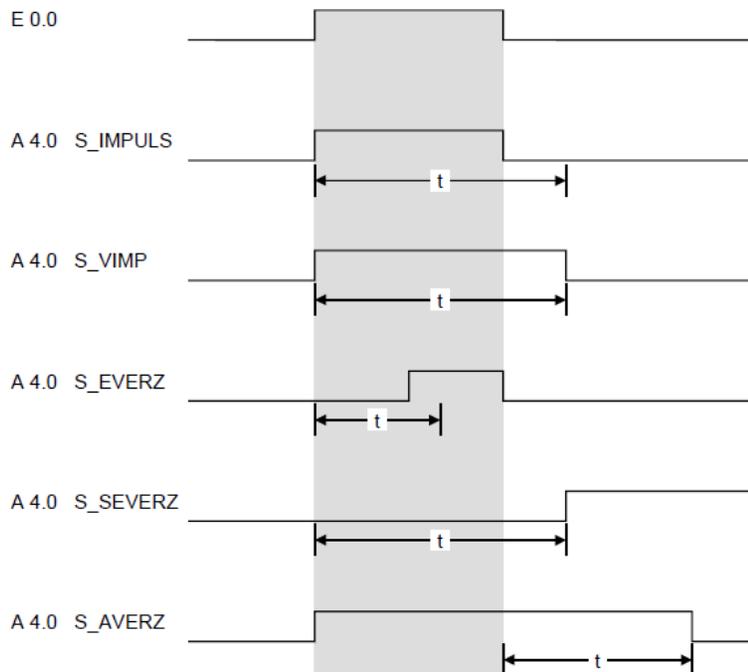


Tabla. 9 temporizadores

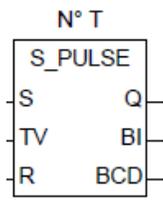
TEMPORIZADORES	DESCRIPCIÓN
S_IMPULS Temporizador de impulso	El tiempo máximo que la señal de salida permanece a 1 corresponde al valor de temporización t programado. La señal de salida permanece a 1 durante un tiempo inferior si la señal de entrada cambia a 0.
S_VIMP Temporizador de impulso Prolongado	La señal de salida permanece a 1 durante el tiempo programado, independientemente del tiempo en que la señal de entrada esté a 1.
S_EVERZ Temporizador de retardo a la conexión	La señal de salida es 1 solamente si ha finalizado el tiempo programado y la señal de entrada sigue siendo 1.
S_SEVERZ Temporizador de retardo a la conexión con memoria	La señal de salida cambia de 0 a 1 solamente si ha finalizado el tiempo programado, independientemente del tiempo en que la señal de salida esté a 1.
S_AVERZ Temporizador de retardo a la desconexión	La señal de salida es 1 cuando la señal de entrada es 1 o cuando el temporizador está en marcha. El temporizador arranca cuando la señal de entrada cambia de 1 a 0.

- S_IMPULS

Tabla.10 instrucción de temporizador S_IMPULS

ARÁMETROS	TIPO DE DATOS	DESCRIPCIÓN
N. ° de t	TIMER	Número de identificación del temporizador, el área varía según la CPU que se utilice
S	BOOL	Entrada de arranque
TV	S5TIME	Valor de temporización predeterminado
R	BOOL	Valor de temporización predeterminado
BI	WORD	Valor de temporización actual, codificado en binario
BCD	WORD	Tiempo restante, formato BCD
Q	BOOL	Estado del temporizador

SÍMBOLO

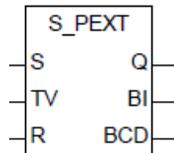


- S_VIMP

Tabla. 11 instrucción de temporizador S_VIMP

PARÁMETROS	TIPO DE DATOS	DESCRIPCIÓN
N. ° de t	TIMER	Número de identificación del temporizador, el área varía según la CPU que se utilice
S	BOOL	Entrada de arranque
TV	S5TIME	Valor de temporización predeterminado
R	BOOL	Entrada de desactivación
BI	WORD	Valor de temporización actual, codificado en binario
BCD	WORD	Tiempo restante, formato BCD
Q	BOOL	Estado del temporizador

SÍMBOLO

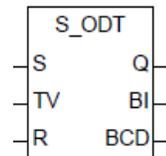


- S_EVERZ

Tabla. 12 instrucción de temporizador S_EVEREZ

PARÁMETROS	TIPO DE DATOS	DESCRIPCIÓN
N. ° de t	TIMER	Número de identificación del temporizador, el área varía según la CPU que se utilice
S	BOOL	Entrada de arranque
TV	S5TIME	Valor de temporización predeterminado
R	BOOL	Entrada de desactivación
BI	WORD	Valor de temporización actual, codificado en binario
BCD	WORD	Valor de temporización actual, formato BCD
Q	BOOL	Estado del temporizador

SÍMBOLO



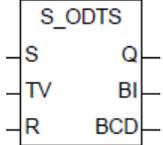
- S_SEVERZ

Tabla. 13 instrucciones de temporizador S_EVERZ

PARÁMETROS	TIPO DE DATOS	DESCRIPCIÓN
N. ° de t	TIMER	Número de identificación del temporizador, el área varía según la CPU que se utilice
S	BOOL	Entrada de arranque
TV	S5TIME	Valor de temporización predeterminado
R	BOOL	Entrada de desactivación
BI	WORD	Valor de temporización actual, codificado en binario
BCD	WORD	Valor de temporización actual, formato BCD
Q	BOOL	Estado del temporizador

N° T

SÍMBOLO



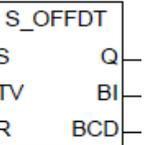
- S_AVERZ

Tabla. 14 instrucción de temporizador S_AVERZ

PARÁMETROS	TIPO DE DATOS	DESCRIPCIÓN
N. ° de t	TIMER	Número de identificación del temporizador, el área varía según la CPU que se utilice
S	BOOL	Entrada de arranque
TV	S5TIME	Valor de temporización predeterminado
R	BOOL	Entrada de desactivación
BI	WORD	Valor de temporización actual, codificado en binario
BCD	WORD	Valor de temporización actual, formato BCD
Q	BOOL	Estado del temporizador

N° T

SÍMBOLO



5. CREACIÓN DEL PROYECTO EN SIMATIC

Haga doble clic en el ícono **Administrador SIMATIC** en el escritorio de Windows. Elija el comando de menú **Archivo > Asistente "Nuevo proyecto"**, si no se ha abierto automáticamente.

Figura. 33 ventana de asistente de un proyecto

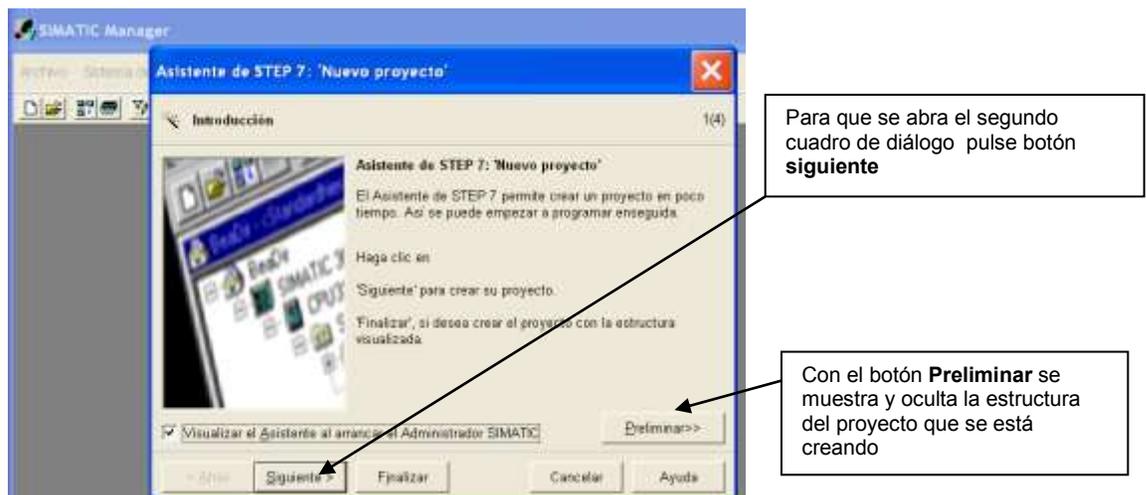


Figura. 34 ventana de asistente de un proyecto escogencia de la CPU

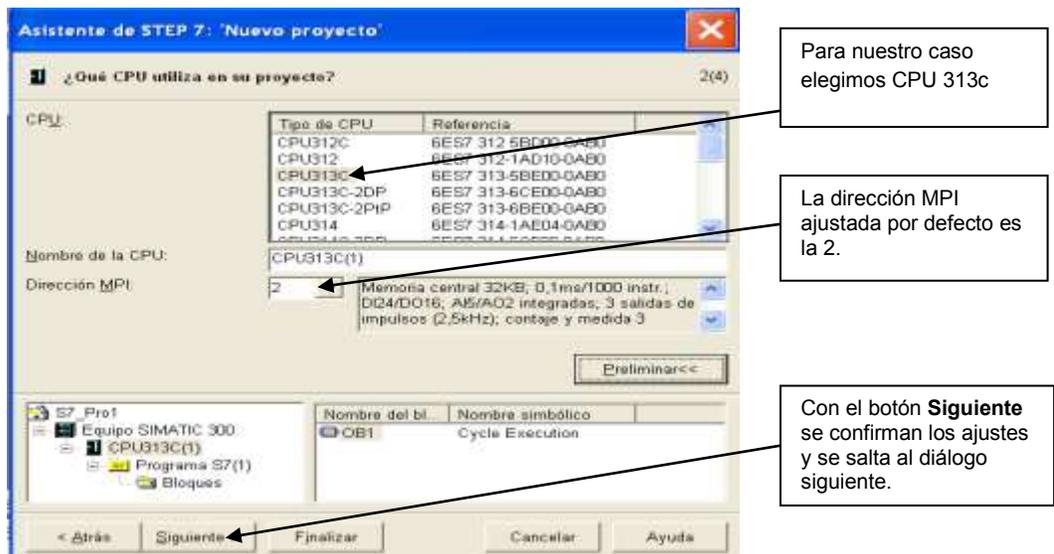


Figura. 35 Ventana de asistente de un proyecto escogencia del lenguaje de programación

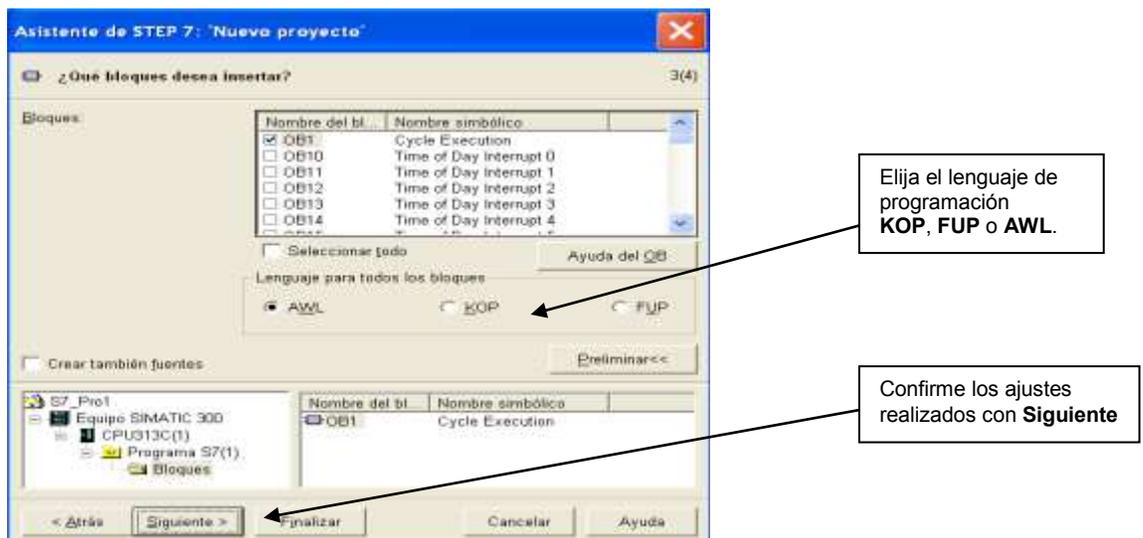
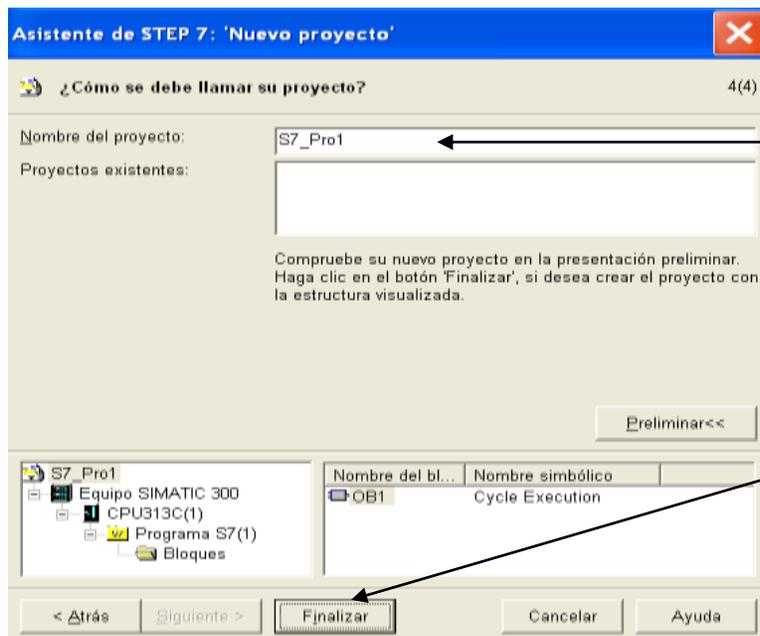


Figura. 36 Ventana de asistente de un proyecto – nombre del proyecto



Seleccione con doble clic el nombre que aparece en la casilla "Nombre del proyecto"

El botón **Finalizar** permite generar el nuevo proyecto conforme a la presentación preliminar.

Al pulsar el botón de comando **Finalizar** se abre el Administrador SIMATIC mostrando la ventana del proyecto creado, "S7_pro1". De forma estándar, el asistente de STEP7 se activa cada vez que se arranca el programa.

Figura. 37 ventana de configuración PLC

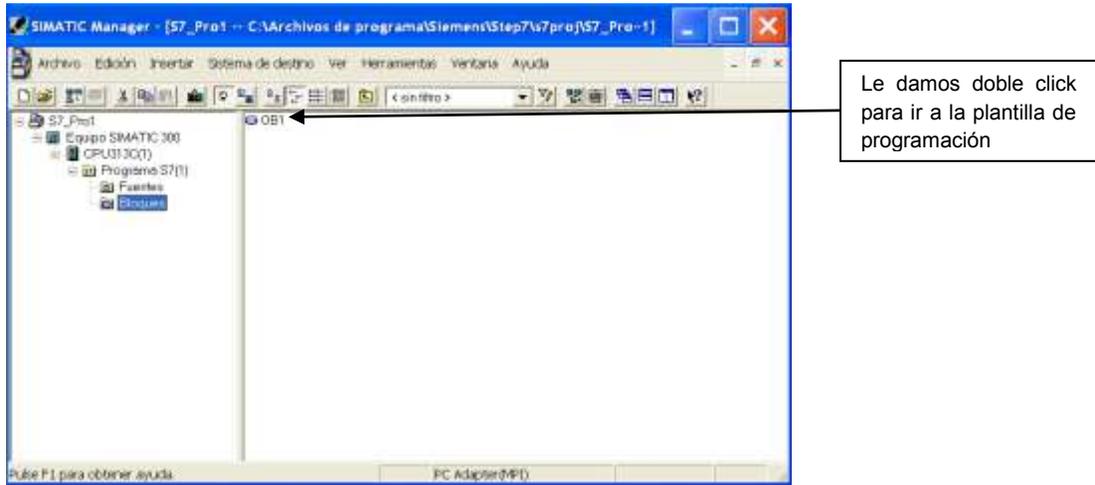
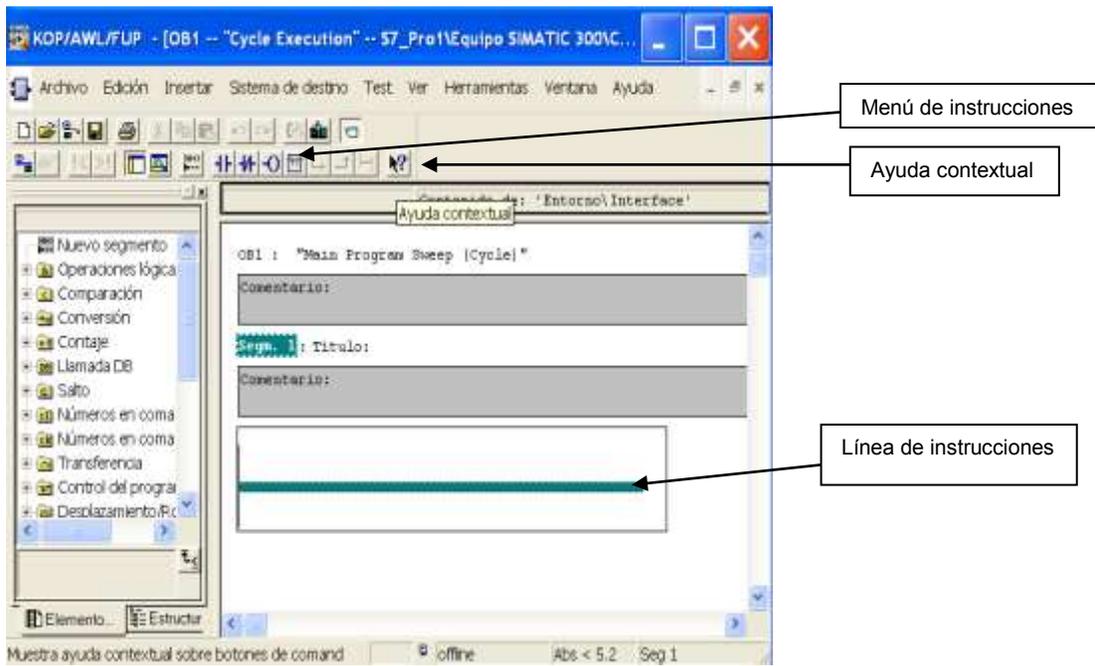
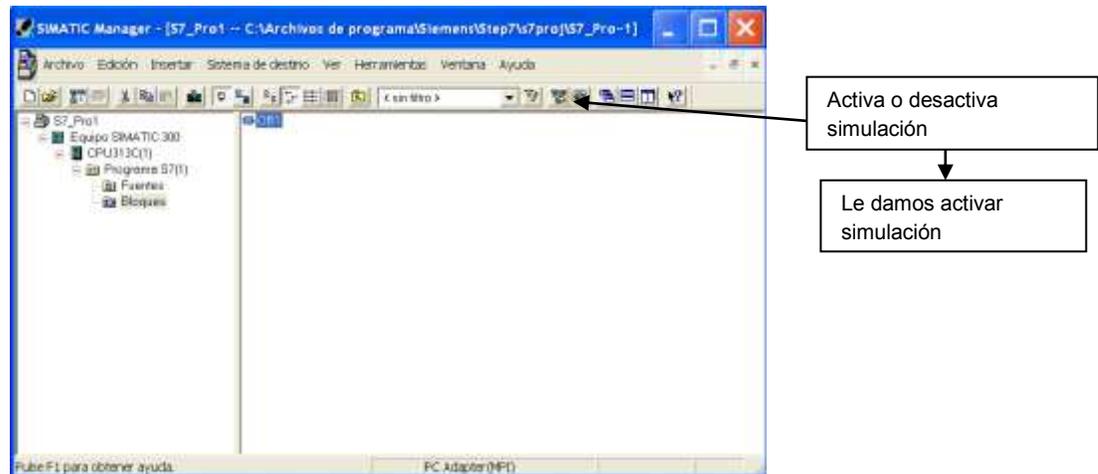


Figura. 38 plantilla de programación de PLC



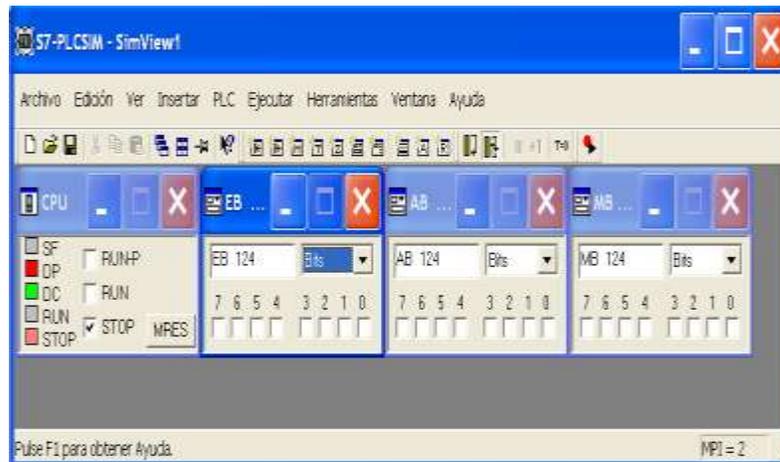
5.1 SIMULACIÓN DEL PROYECTO EN SIMATIC

Figura. 39 activar simulador del PLC



Luego de pulsar sobre el botón de simulación nos aparece una nueva pantalla donde podremos visualizar las entradas, salidas, memorias, contadores y temporizadores del programa cargado virtualmente al PLC.

Figura. 40 Ventana de simulación del PLC



Para poder simular el programa deseado pulsamos en RUN

EB: entradas
 AB: salidas
 MB: memorias
 T: temporizadores
 C: contadores

6. DESARROLLO DEL PROYECTO EN SIMATIC

Como ya hemos expresado nos centraremos en el uso del lenguaje KOP.

La programación está compuesta por dos partes principales. TABLA DE DATOS (DB1) es donde asignaremos a cada entrada y salida el tipo de estructura, comentarios además le asociaremos un nombre; este último nos será de gran utilidad para al momento de programar en la parte del Winlog encargada de la edición de las interfaces que se diseñarán en este SCADA.

Creamos una tabla de datos que nos será de mucha ayuda, con el fin de darle un nombre a cada variable (cada entrada o salida la identificaremos con un seudónimo), figura (41).

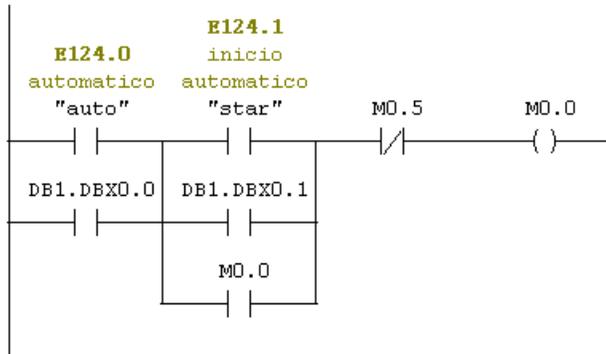
Figura. 41 Tabla de datos

Dirección	Nombre	Tipo	Valor inicial	Comentario
0.0		STRUCT		
+0.0	auto	BOOL	FALSE	Variable provisional
+0.1	star	BOOL	FALSE	inicia modo automatico
+0.2	stop	BOOL	FALSE	detiene modo automatico
+0.3	sensornivelbajol	BOOL	FALSE	sensor de nivel bajo de tolva
+0.4	sensorposicion	BOOL	FALSE	sensor de posicion banda
+0.5	s1	BOOL	FALSE	sensor de piston dentro
+0.6	s2	BOOL	FALSE	sensor de piston fuera
+0.7	sensornivelaltol	BOOL	FALSE	sensor de nivel alto
+1.0	manwal	BOOL	FALSE	
+1.1	starbanda	BOOL	FALSE	
+1.2	starmotor	BOOL	FALSE	
+1.3	bomba	BOOL	FALSE	bomba
+1.4	piston	BOOL	FALSE	piston
=2.0		END_STRUCT		

6.1. DESARROLLO DEL PROGRAMA EN SIMATIC

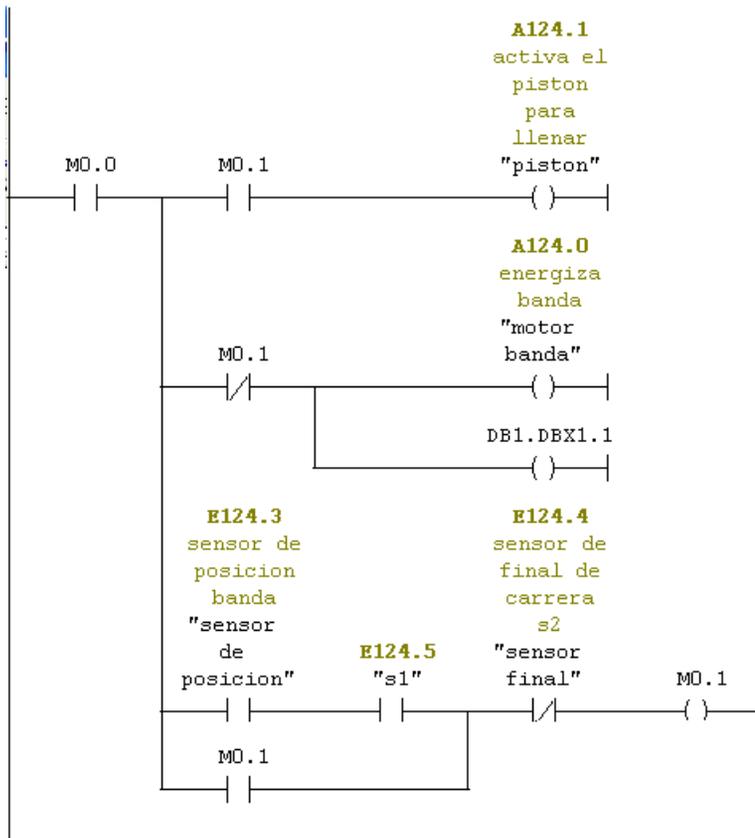
Segm. 1: STAR O STOP

PERMITE INICIAR O PARAR EL PROCESO DE LLENADO



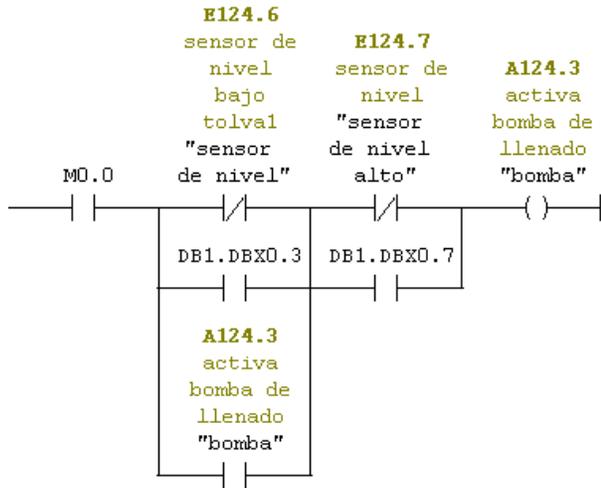
Segm. 2: LLENADO AUTOMATICO

INICIA EL PROCESO DE LLENADO EN MODO AUTOMATICO



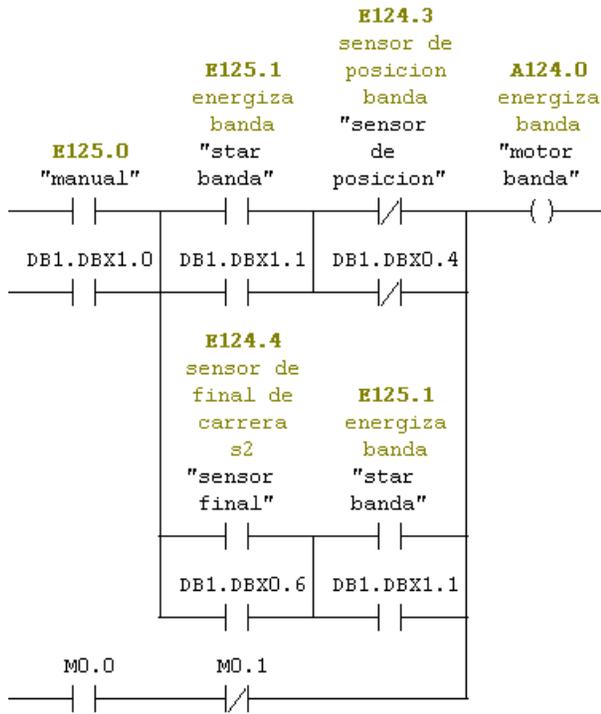
Segm. 3 : ACTIVA LA BOMBA DE LLENADO

Comentario:

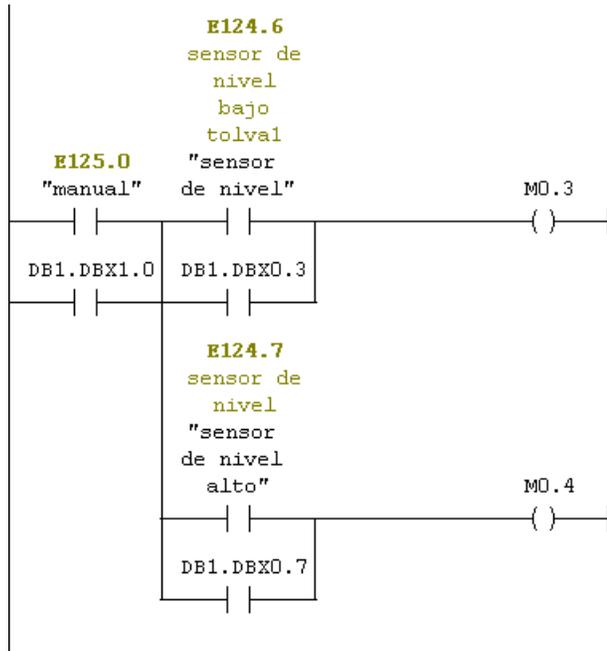
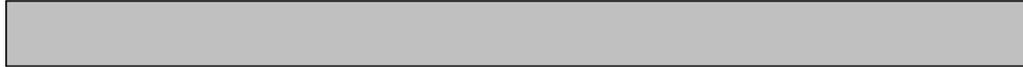


Segm. 4 : MODO MANUAL

INICIA EL PROCESO EN MODO MANUAL



Segm. 5 : LLENADO MANUAL



7. PROGRAMACIÓN ENTORNO GRÁFICO

Para el desarrollo del entorno gráfico nos basamos en un software para sistemas SCADAS llamado WINLOG versión (2.06.18) de la empresa italiana “SIELCO SISTEM”; donde la misión es ofrecer soluciones SCADA para la industria y la construcción de la automatización.

7.1. DESCARGA E INSTALACIÓN SOFTWARE DE ENTORNO GRÁFICO “WINLOG”

Esta herramienta de desarrollo de sistemas SCADA la podemos descarga como demo desde la página de SIELCO SISTEM. (www.sielcosistemi.com).

1. Una vez descargado el instalador, iniciamos haciendo doble clic sobre el archivo ejecutable. El archivo empezará a ser analizado y luego mostrará una opción para el idioma. Seleccionamos el idioma de nuestra preferencia entre INGLÉS e ITALIANO.

Figura. 42 ventana de inicio instalación del WINLOG



2. Haciendo clic en “next” de este proceso nos llevará a la etapa del contrato de licencia. Después de haber comprendido lo que éste menciona seleccionar la

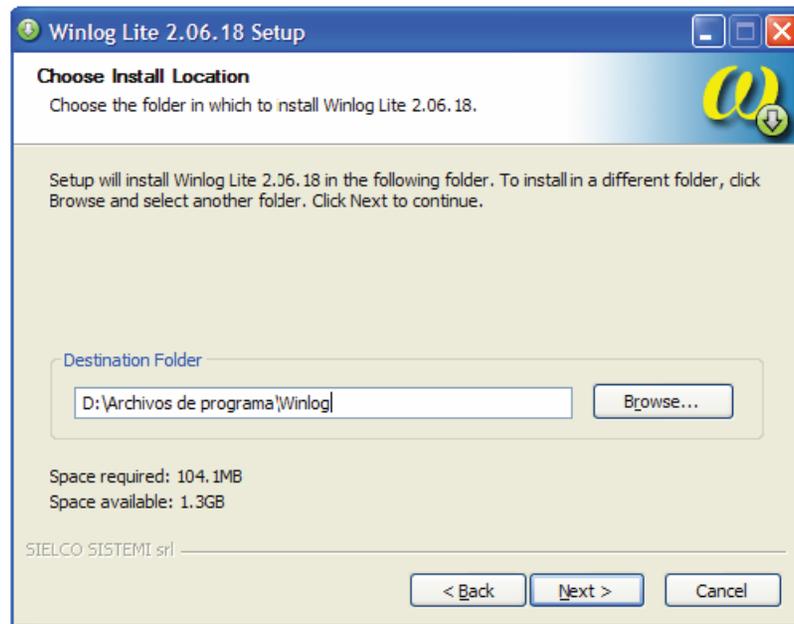
opción "I accept. the terms in the License Agreement". Luego damos clic en "next" nuevamente

Figura. 43 ventana de licencia del WINLOG



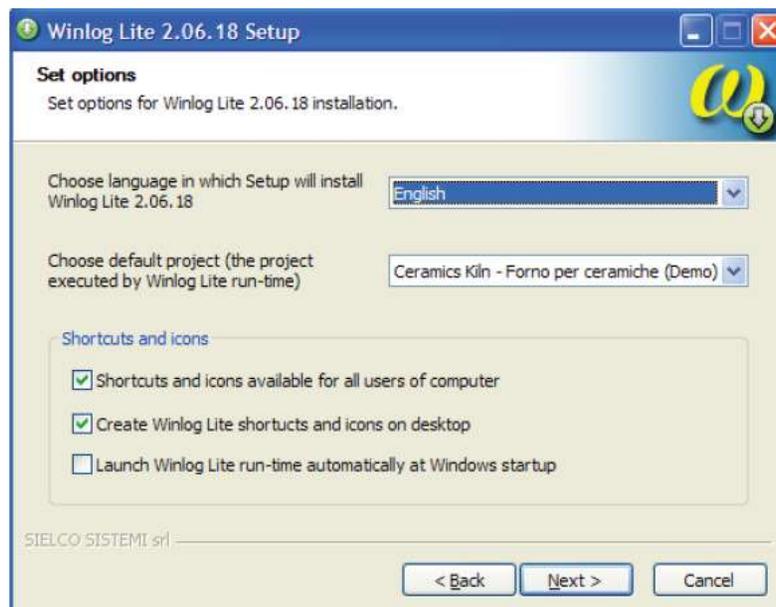
3. elegimos el directorio en el cual se instalará el programa. Por defecto, el programa elegirá Archivos de programa\Winlog\ en el disco duro principal. Para continuar con el proceso de instalación damos clic en "next".

Figura. 44 Ventana instalación del WINLOG



4. Seleccionamos las opciones tal y como se muestran en la siguiente figura y damos clic en "next" para continuar.

Figura. 45 Ventana de instalación del WINLOG



5. Finalmente aparecerá la opción “Install” sobre la cual hacemos clic para inicial la instalación del programa.
6. Una vez finalizada la instalación, se pueden elegir dos opciones antes de cerrar el diálogo de instalación presionando el botón “Finish”.

Figura. 46 ventana de finalización de la instalación WINLOG



7.2. MANEJO DEL PROGRAMA (Software Winlog 2.06.18)

El Winlog está compuesto por dos partes principales. El RUNTIME es el encargado de simular y ejecutar todo el programa configurado.

El PROJECT MANAGER es la parte del Winlog encargada de la edición de las interfaces que se diseñarán en este SCADA.

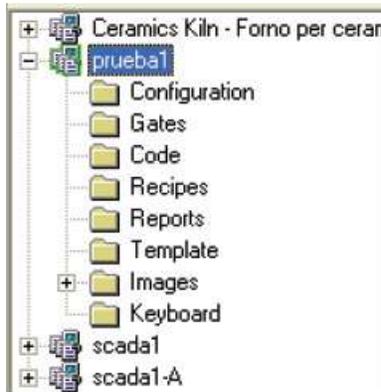
Para abrirlo y ejecutarlo existen dos principales maneras:

- Desde el escritorio de su PC buscando un icono con el símbolo y el nombre “Project Manager”.
- Desde el menú Inicio – Programas- Winlog lite y aparecerá el mismo símbolo con el nombre “Project Manager”.

Lo primero que haremos es crear un nuevo proyecto. Para ello se debe hacer clic en el icono con la imagen.

El programa preguntará el nombre para el nuevo archivo y se le debe asignar uno, por ejemplo “prueba”.
Una vez creado el proyecto el programa ha diseñado un árbol completo, así:

Figura. 47 proyectos winlog



La explicación de algunas partes importantes de este árbol se puede apreciar a continuación:

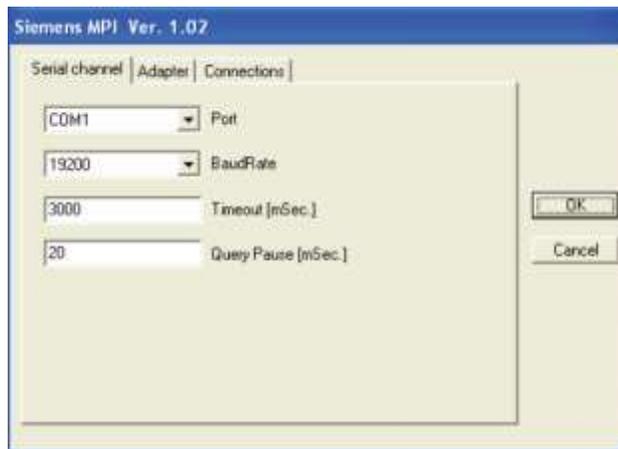
CONFIGURACIÓN:

Dentro de la rama “configuración” se pueden encontrar otras ramas aún más específicas como lo son:

- **Options:** Dentro de las opciones posibles a modificar son la capacidad del disco duro, dirección IP del computador y algunos otros aspectos generales.
- **Channels:** Esta rama es muy importante puesto que se configuran los protocolos de comunicación entre el PC y el PLC. Posee varios canales para guardar las diferentes configuraciones. Dentro de las posibles configuraciones de los canales se encuentra una llamada “Siemens MPI”, la cual sirve para comunicar el SCADA con el PLC Siemens s7-300 específicamente. El número del canal asignado a la configuración se debe tener en cuenta para la creación de variables.

Los parámetros que se deben configurar dentro de este driver MPI para Siemens 300 son:

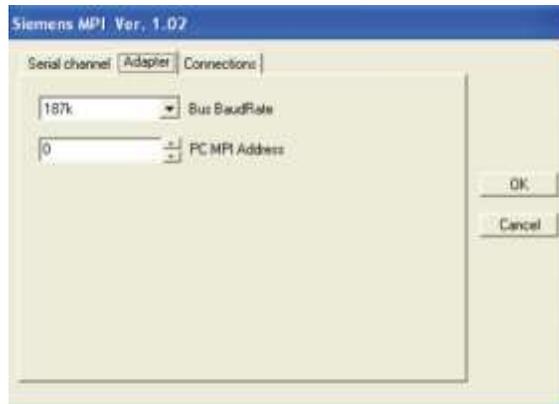
Figura. 48 configuración parámetros winlog



Dentro de los cuales se encuentra el puerto COM1 (serial), velocidad de 19200 baudios/seg generalmente, timeout y Query pause.

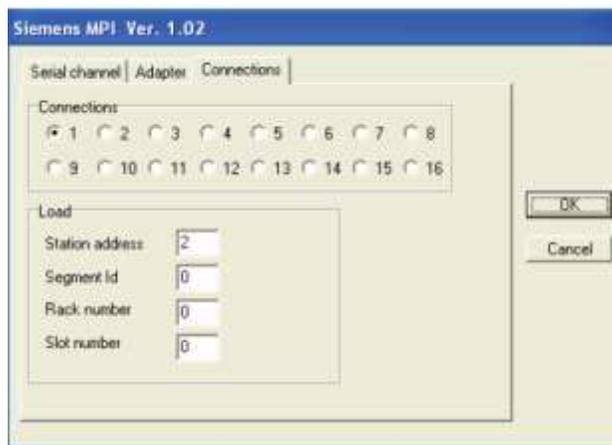
En la pestaña para el adaptador se debe configurar principalmente la dirección MPI del PC.

Figura. 49 Configuración parámetros adaptador winlog



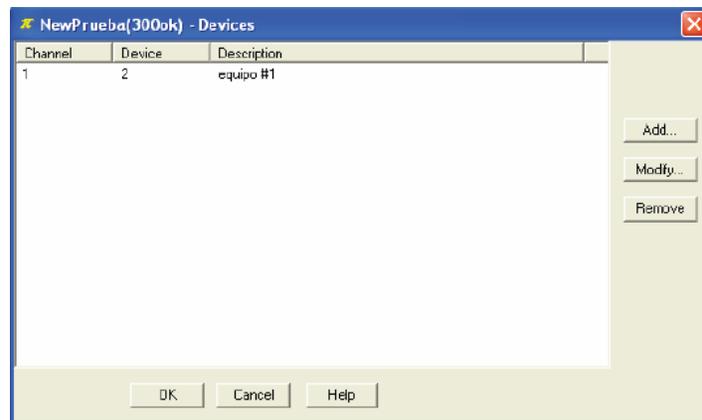
En la pestaña para conexiones se debe configurar “*station address*” como 2, debido a que esta es la dirección del PLC dentro de la interfaz MPI.

Figura. 50 Configuración parámetros conexiones winlog



- **Devices:** Aquí se pueden adicionar varios elementos o “aparatos” al mismo canal de comunicación. A cada uno se le asigna un atributo numérico que lo diferencia de los otros. Para el caso del Siemens 300 se debe configurar un nuevo *device* con dirección 2, así:

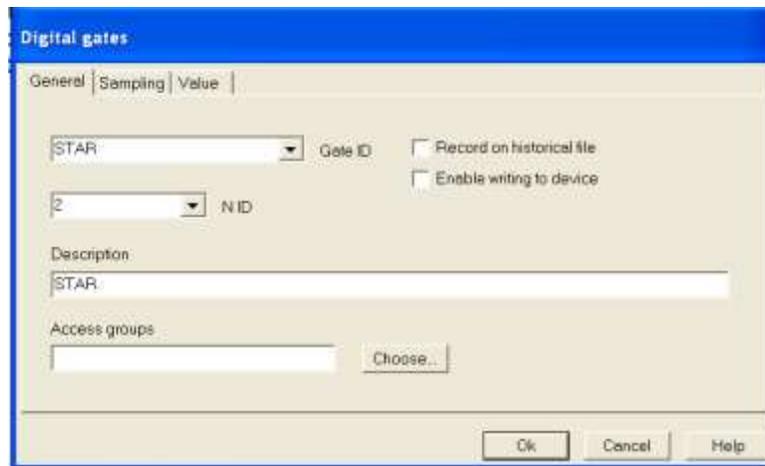
Figura. 51 Configuración parámetros device winlog



GATES: Se pueden seleccionar diferentes tipos de variables, como lo son numéricas, digitales, alarmas y eventos.

- **VARIABLES TIPO DIGITAL:** Un ejemplo de una variable digital se muestra a continuación.

Figura. 52 variables digitales



Dentro de esta pestaña se encuentran varias casillas, se describe a continuación:

GATE ID: Se usa para darle un nombre a la variable que se está creando. El nombre no debe contener espacios. Se deben activar ambas opciones al lado derecha de esta casilla para habilitar la escritura en el PLC

N ID: Utilizada para dar un número de identificación a la variable. No pueden existir dos variables del mismo tipo con el mismo N ID, incluso si tienen diferente nombre

DESCRIPTION: Espacio utilizado para documentar la variable.

ACCESS GROUPS: Utilizado para agrupar las variables en diferentes bloques.

Figura. 53 Pestañas de muestreo



CHANNEL -> PROTOCOL: Se debe seleccionar un número de canal. Este número debe estar previamente configurado en la opción CONFIGURATION – CHANNELS.

DEVICE: Se le debe asignar el número del PLC dentro de la MPI para que el sistema scada pueda acceder a los datos

ADDRESS: Es utilizada para que el Winlog busque la variable específica dentro del device especificado. Haciendo clic en el símbolo se puede acceder a ayudas específicas de direccionamiento para cada protocolo.

Por ejemplo, en el gráfico, se muestra el direccionamiento para una variable que ocupa el bit 2.0 del bloque de datos 1 del PLC. Si se desea acceder a una variable tipo entrada el direccionamiento sería EBX.Y y para una tipo salida sería ABX.Y;

donde X es el byte donde se encuentra la variable y la Y significa el bit dentro del byte X .

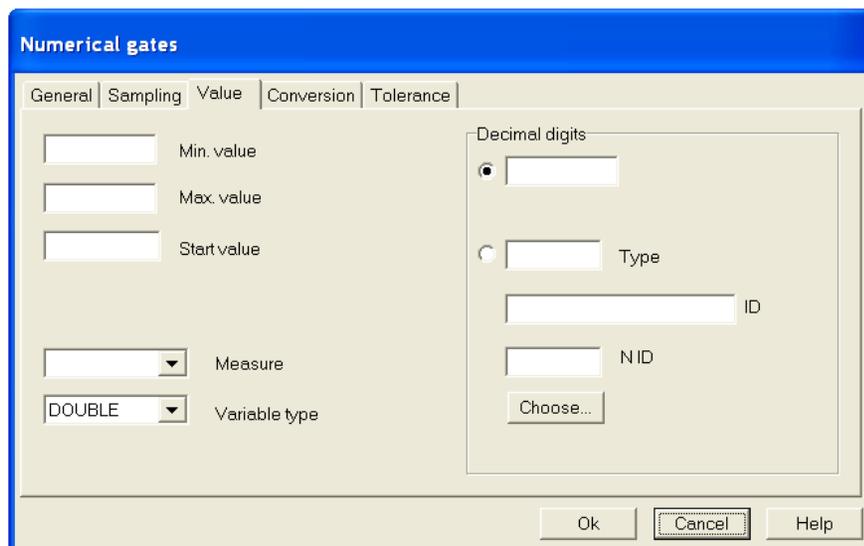
SAMPLE: Sirve para indicar el muestreo que se le hace a la variable actual. Si se elige la opción “*always*” se hará el muestreo todo el tiempo. Si se elige la opción “*never*” nunca se muestreará y con la opción “*if in monitor*” sólo se muestrea cuando aparezca en pantalla.

SAMPLE FREQ: Significa que se debe ingresar un valor numérico para elegir los intervalos de tiempo entre muestras. Si el número que se ingresa es un “0”, entonces el muestreo se realizará siempre.

VARIABLES TIPO NUMÉRICO: Las pestañas general y muestreo de este tipo de variables son idénticas a las de variables tipo digital. La pestaña de muestreo tiene una pequeña diferencia en la dirección, puesto que en este caso no se quiere leer un bit sino un byte, por lo que la parte de bit de la dirección debe ser removida.

La diferencia principal entre la creación de estas variables y las digitales es la pestaña *value*

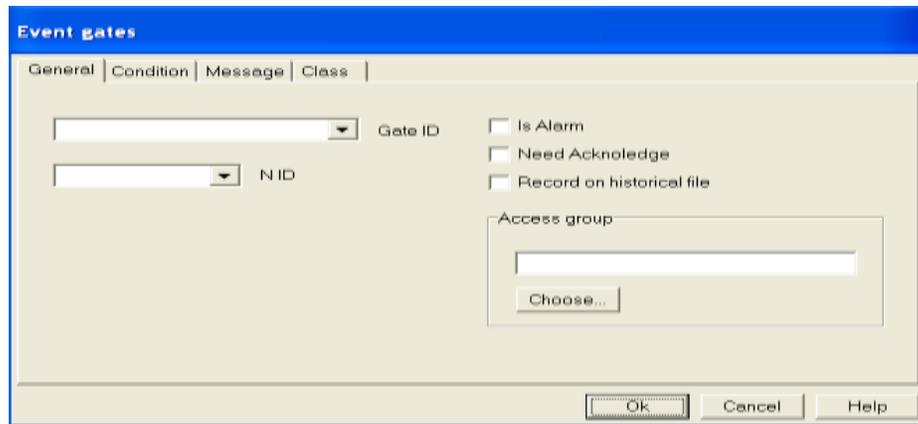
Figura. 54 variables numericas



Dentro de esta pestaña lo más importante es configurar la casilla *Variable type*, donde se le asigna un tipo a la variable, este debe corresponder con el tipo de dato asignado a esta variable cuando se declaró en el PLC.

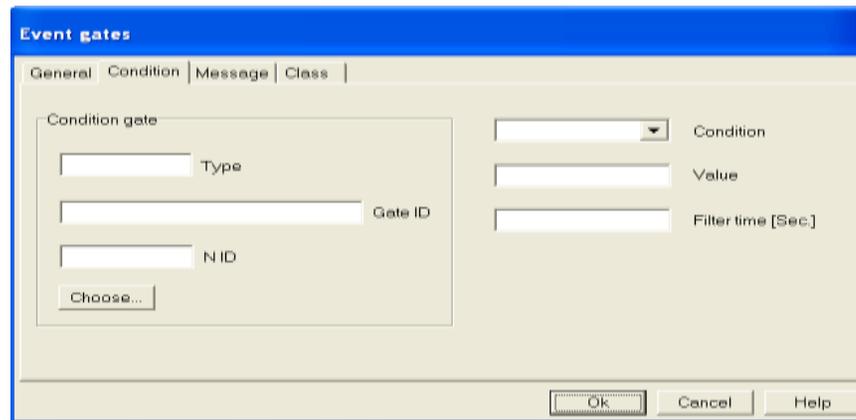
VARIABLES TIPO ALARMA: Estas variables dependen directamente de otra variable de cualquier tipo declarada previamente. La pestaña general se muestra a continuación, donde las casillas *Gate ID* y *N ID* son exactamente iguales a las descritas anteriormente. Debido a que es una alarma, se debe seleccionar la opción “*Is Alarm*”.

Figura. 55 Pestaña General



Las condiciones para la activación o no activación de la alarma se configuran en la pestaña condición

Figura. 56 eventos



En el recuadro a la izquierda llamado Condición Gate no se debe escribir nada. Simplemente se debe buscar la variable haciendo clic en el botón Choose.

Al lado derecho de la pestaña se puede seleccionar la condición. En este caso, la alarma se activará en el momento que la variable numérica "análoga" supere el valor de 60.

Dentro de la pestaña de mensaje se puede configurar un mensaje que aparece en caso de que la alarma esté activa. Este mensaje solo aparecerá en el modo de funcionamiento runtime en la esquina inferior izquierda del mismo

CODE

Este programa, ofrece la opción de codificar una serie de instrucciones que pueden ayudar al programa a trabajar mejor. Dentro de estas instrucciones se pueden crear funciones específicas, de manera que los atributos de los objetos y muchas otras posibilidades sean modificados.

Por ejemplo, para crear animaciones es necesario llamar una función encargada de cambiar los atributos de posición del elemento a animar. También se pueden modificar atributos como tamaño y texto.

Además, existe la posibilidad de crear reportes, e incluso ejecutarlos desde el código; Dicha programación se hace parecida al lenguaje C++.

Una función muy importante es utilizada para ejecutar un "template" en el momento que se corra el *runtime*, es decir, se abrirá esta plantilla por defecto. Si se desea hacer esto se debe añadir el siguiente segmento al código.

Ejemplo:

```
Function void main ()
#startup
TPageOpen ("template#1");
End
```

El comando “function void” creará una función y la palabra “main” es el nombre de la misma.

La instrucción “#startup” indica que esta función se ejecutará al iniciar el *runtime*.

El comando TPageOpen se encarga de abrir el *template* que posea el mismo nombre que el texto entre comillas.

Otros ejemplos de funciones se pueden encontrar en las ayudas del paquete.

TEMPLATE

Aquí se crean las interfaces hombre-máquina (HMI). Esta parte es vital dentro del diseño gráfico.

Se deben incluir los gráficos y relacionarlos con las variables que van en el proceso.

IMAGES

En este lugar se almacenan todas la gráficas que conforman el entorno gráfico.

8. ELABORACIÓN DEL SOFTWARE DE VISUALIZACIÓN

8.1. DESCRIPCIÓN DEL PROCESO

El proceso inicia seleccionando el modo de ejecución, ya sea automático o manual. En modo automático interviene dos switches que serán nombrados “STAR”, “STOP” cuya finalidad es dar inicio al proceso oprimiendo (“STAR”) y detener el proceso (“STOP”).

En modo manual intervienen 3 switches que serán nombrados como “MOTOR”, “LLENADO”, “BOMBA”; el switch llamado motor da inicio a la banda transportadora, el switch “LLENADO” da la señal al cilindro neumático el cual cumple la función de inyectar la materia prima al envase de presentación, y por último la “BOMBA” esta cumple la función de pasar la materia prima que en este caso es líquido desde el tanque de almacenamiento de este, hasta la tolva de distribución; cabe señalar que tanto en el tanque de almacenamiento y la tolva de distribución tiene sensores de nivel donde nos indicará con señales luminosas los niveles mínimos y máximos de éste.

8.2. REGISTRO DE OPERADOR Y REFERENCIA DEL PRODUCTO A ENVASAR

Figura. 57 inicio entorno, es donde el administrador ingresa con clave y registra los operarios, también ingresan los operarios a controlar el proceso de llenado

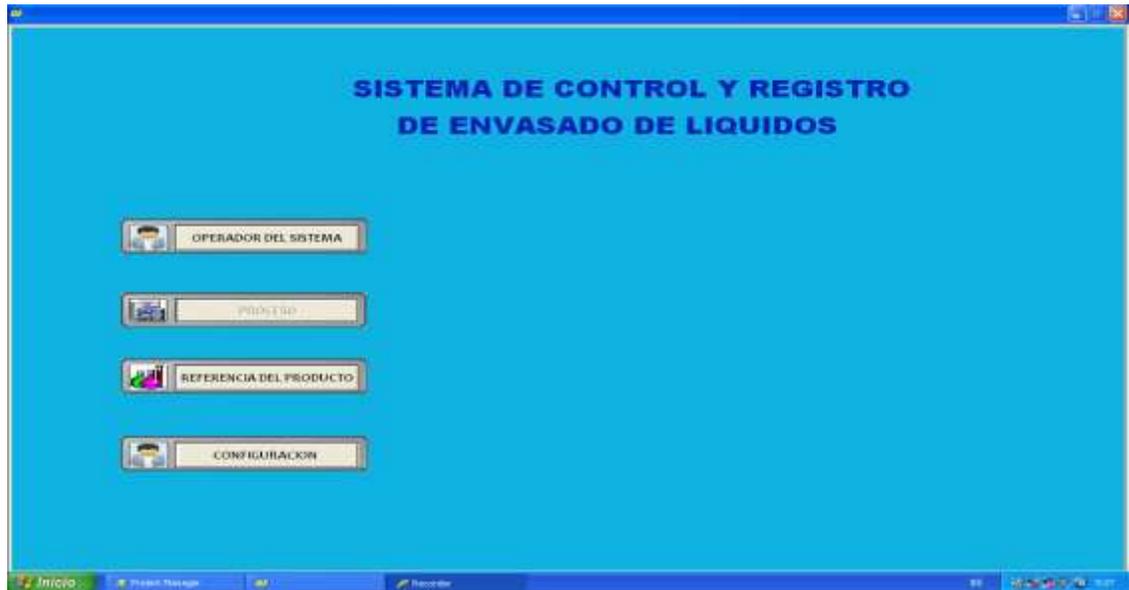


Figura 58 configuración, es donde el administrador ingresa los operarios



Figura 59 configuración usuarios



➤ **TABLERO DE CONTROL DE MANDO**

El tablero de control de mando nos permitirá elegir el modo de ejecución del proceso ya sea manual o automático. Figura (61)

➤ **BANDA TRANSPORTADORA**

Cabe señalar que el proceso al cual estamos enfocando este proyecto carece de este mismo. Figura (61)

➤ **TOLVA DE DISTRIBUCIÓN**

La tolva de distribución permite la facilidad de almacenamiento de líquido en pocas cantidades. Figura (61)

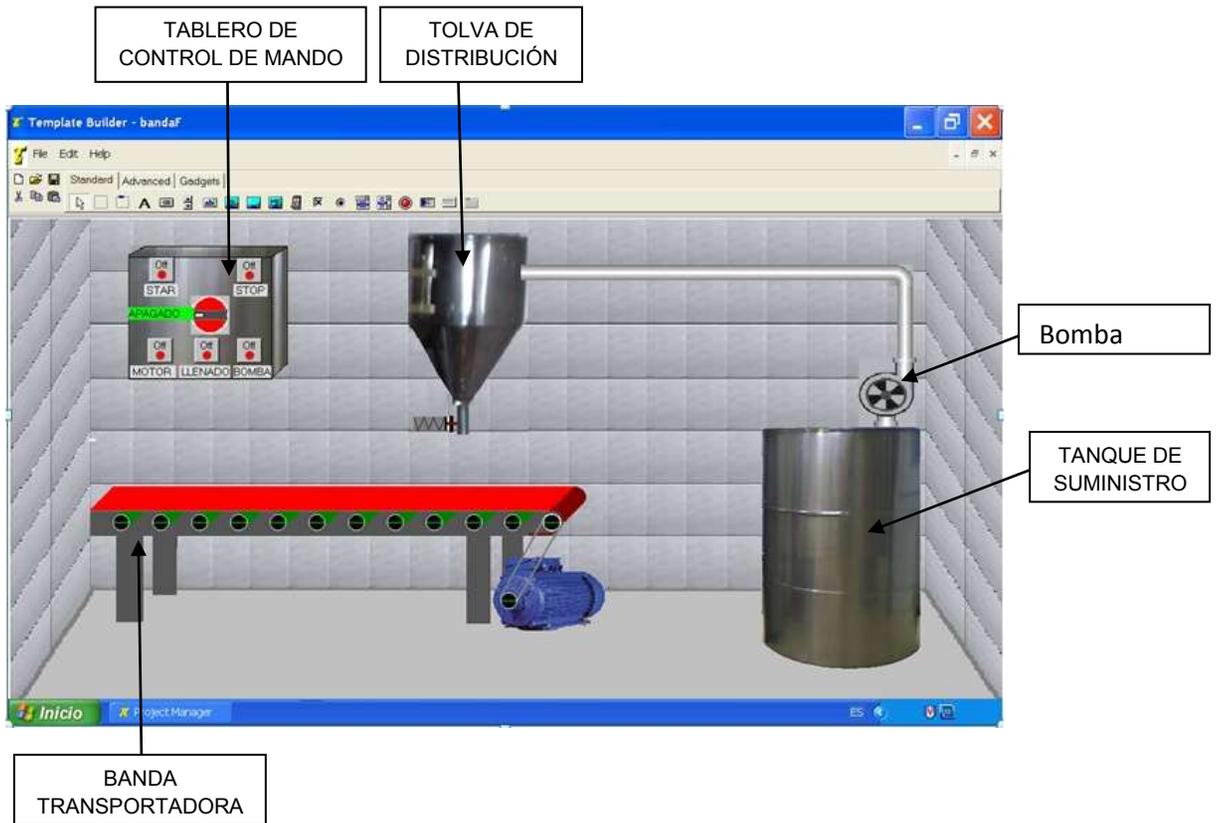
➤ **TANQUE DE SUMINISTRO**

El tanque de suministro nos permite el almacenamiento del producto en gran cantidad para posteriormente ser embasado a la tolva de distribución. Figura (61)

➤ **BOMBA**

La bomba se encarga de succionar líquido desde el tanque de almacenamiento hasta la tolva de distribución. Figura (61)

Figura. 60 proceso simulado del llenado de envases



8.3. ELABORACIÓN DEL CODIGO DE VISUALIZACIÓN

8.3.1. CÓDIGO

```

global bool x=true;    ///declaración de variables locales

global bool y=true;

global bool z=true;

function void main1()

SetDigGateValue("bandaon",14,0);    //transcurso de banda inicial///
SetDigGateValue("recorrido",15,0);    SetDigGateValue("recorrido1",16,0);
    
```

```

end

function void ini()

SetDigGateValue("aut",1,0);

Sleep(1000);

if (GetDigGateValue("aut",1)==1) then

TObjSetText(8,"AUTOMATICO");

end

if (GetDigGateValue("man",2)==1) then

TObjSetText(8,"MANUAL");

end

if (GetDigGateValue("aut",1)==0 && GetDigGateValue("man",2)==0) then

TObjSetText(8,"APAGADO");

end

end

function void selector()

```

Rutina de selector

Apagado a automático

```

if (GetDigGateValue("aut",1)==0 && GetDigGateValue("man",2)==0) then

SetDigGateValue("aut",1,1);

TObjSetText(8,"AUTOMATICO");

x =true;

y =true;

SetDigGateValue("varios19",19,1);

SetDigGateValue("recorrido1",16,0);

StopFunction("selector");

end

```

```

.....Pasar de automático a manual.....
if (GetDigGateValue("aut",1)==1 && GetDigGateValue("man",2)==0) then
SetDigGateValue("man",2,1);
SetDigGateValue("aut",1,0);
TObjSetText(8,"MANUAL");
StopFunction("selector");
End

.....Para pasar de manual a apagado.....
if (GetDigGateValue("man",2)==1 && GetDigGateValue("aut",1)==0) then
SetDigGateValue("aut",1,0);
SetDigGateValue("man",2,0);
TObjSetText(8,"APAGADO");
StopFunction("selector");
End

End

.....Para iniciar secuencia de posición inicial.....
Function void llenado1()      ...posición inicial de llenado....
While (WindowIsOpen())
.....Control automático.....
If ( GetDigGateValue("star",4)==1 && x==true) Then
SetDigGateValue("varios19",19,0);
SetDigGateValue("bandaon",14,1);
sleep(2000);
SetDigGateValue("bandaon",14,0);
x=false;
SetDigGateValue("recorrido",15,1);
sleep(4000);

```

```

SetDigGateValue("recorrido",15,0);
SetDigGateValue("recorrido1",16,1);
end
if (GetDigGateValue("senposicion",9)==1 &&
GetDigGateValue("piston",10)==1)then// &&
GetDigGateValue("varios18",18)==1) Then //ejecuta la primera rutina de llenado
SetDigGateValue("recorrido1",16,0);
If (GetDigGateValue("senposicion",9)==1) Then //
SetDigGateValue("varios18",18,1);
sleep(1000);
SetDigGateValue("varios18",18,0);
End
end
... control manual....
If ( GetDigGateValue("man",2)==1 && GetDigGateValue("motor",6)==1 && y==true ) Then
SetDigGateValue("varios19",19,0);
SetDigGateValue("bandaon",14,1); //transcurso de banda inicial///
sleep(2000);
SetDigGateValue("bandaon",14,0);
y=false;
SetDigGateValue("recorrido",15,1);
sleep(4000);
SetDigGateValue("recorrido",15,0);
SetDigGateValue("recorrido1",16,1);
end
if (GetDigGateValue("senposicion",9)==1 ///ejecuta la primera rutina de llenado
GetDigGateValue("piston",10)==1 &&

```

```

GetDigGateValue("motor",6)==1)then
SetDigGateValue("recorrido1",16,0);
If (GetDigGateValue("piston",10)==1 &&
GetDigGateValue("motor",6)==1 ) Then //
SetDigGateValue("varios18",18,1);
sleep(1000);
SetDigGateValue("varios18",18,0);
End
End
End
End
function void main()
#startup
BOOL guardar;
TPageOpen("prueba");
SetDigGateValue("operador",1,0);
SetDigGateValue("referencia",2,0);
end

function void operador()
SetDigGateValue("operador",1,1);
end

function void operador1()
bool guardar =QuestionBox("guardar cambios",);
If (guardar==1) Then
MessageBeep(2); //

```

```
SetDigGateValue("operador",1,0);
```

```
End
```

```
end
```

```
function void referencia()
```

```
SetDigGateValue("referencia",2,1);
```

```
end
```

```
function void referencia1()
```

```
bool guardar =MessageBox("guardar cambios",);
```

```
If (guardar==1) Then //
```

```
MessageBeep(2);
```

```
SetDigGateValue("referencia",2,0);
```

```
End
```

```
end
```

8.3.2. TABLA DE VARIABLES GLOBALES

Figura. 61 Tabla de variables globales

	Channel	Device	Gate ID	N ID	Address	Description	Start value	Sample	Sample freq. (Sec)	Read block	Record on historical file	Enable writing to device	Access groups
1	1	2	operador	1			0	Never			F	F	0
2	1	2	aut	1		automatico	0	Never	0		F	F	0
3	1	2	man	2		manual	0	Never			F	F	0
4	1	2	star	4		star	0	Never	0		F	F	0
5	1	2	stop	5		stop	0	Never	0		F	F	0
6	1	2	bomba	3		bomba de llenado	0	Never			F	F	0
7	1	2	motor	6		motor	0	Never	0		F	F	0
8	1	2	sensnivelbajo	7		sensor nivel bajo	0	Never	0		F	F	0
9	1	2	sensnivelalto	8		sensor nivel alto	0	Never	0		F	F	0
10	1	2	senposicion	9		sensor de posicion	0	Never	0		F	F	0
11	1	2	piston	10		piston	0	Never	0		F	F	0
12	1	2	bandaon	14		bandaon	0	Never	0		F	F	0
13	1	2	recomido	15		recomido	0	Never	0		F	F	0
14	1	2	recomido1	16		posicion inicial	0	Never	0		F	F	0
15	1	2	varios	17		variable	0	Never			F	F	0
16	1	2	varios18	18		variable	0	Never	0		F	F	0
17	1	2	varios19	19		variable	0	Never			F	F	0
18	1	2	referencia	2		variable	0	Never			F	F	0

8.3.3. LISTA DE INSTRUCCIONES WINLOG

#startup: Permiten a unirse más instrucciones y por tanto, crear un complejo de la instrucción, y, al mismo tiempo, evitar la redundancia de código.

Beep : Se envía una señal acústica.

- Syntax : Void Beep()

BitAnd: Hace una operación and entre valor1 y valor2.

- Syntax : int BitAnd(int Value1, int Value2)

BitMask: Hace and entre un número y una máscara.

- Syntax : int BitMask(int Num, int Mask)

BitNot: Hace negativo un bit.

- Syntax : int BitNot(int Value)

BitOr: Hace or entre valor1 y valor2.

- Syntax : int BitOr(int Value1, int Value2)

hartSetTimeRange: Establece la fecha de inicio del tiempo y la fecha de finalización del tiempo de los parámetros de una plantilla que figura en una plantilla.

ChartSetTimeRangeEndWidth: Establece la fecha de finalización de tiempo y rango de tiempo de los parámetros de una plantilla que figura en una plantilla.

Close: Se cierra la ventana de código actualmente en primer plano.

Cos: saca coseno de un valor real (ángulo en radianes).

- Syntax : Real Cos(Real Value)

Do While: ejecuta ciclo mientras cumpla la condición.

- Syntax : Do
 [Instructions]
 While (*Comparison*);

EnableShutdownActivar / desactivar ordenador apagado automático después de salir de un período de sesiones de la supervisión.

- Syntax :EnableShutdown(Bool Flag)

EvngateExists: Comprobar si el evento / alarma existe.

- Syntax : Bool EvngateExists(String Name, Int Id)

Exit: Cerrar CBuilder y actual sesión de trabajo.

Exp: Sacar el valor exponencial de un número real (e^x).

- Syntax : Real Exp(Real Value)

For cycle: ejecuta un ciclo donde el número de iteraciones están predefinidos, necesita un valor entero como contador.

- Syntax : For *VarName* = *Expression* To *Expression* Do
 [*Instructions*]
End

HexStrToInt: Convierte un número hexadecimal a un entero.

- Syntax : Int HexStrToInt(String HexValue)

HisViewEnablePrintOnCreation(): Imprime automáticamente las alarmas o eventos históricos vista histórico .

- Syntax : void HisViewEnablePrintOnCreation()

IconMessageBox: Muestra en la pantalla una ventana con texto y especifica el título, así como cualquier combinación de iconos predefinidos y los botones. Se espera la confirmación del usuario.

- Syntax : Int IconMessageBox(String Text, String Title, Int ButtonType, Int IconType, Int DefaultButton)

If Then Else : Se ejecuta un trozo de código sólo si la condición es cierta.

- Syntax : If (*Condition*) Then *Instructions*
 [*Else Instructions*]
End

String InputDialog: Muestra en la pantalla un cuadro de diálogo que contiene un texto, un título, una cadena de campo que pueden ser cambiados, y algunos botones para confirmar o deshacer, una cadena de campo es una cadena que el usuario puede cambiar a voluntad usando el teclado.

IntToHexStr: Convierte un valor en formato hexadecimal.

- Syntax : String IntToHexStr(Int Value, Int Digits)

IntToStr: Convierte un valor en su cadena de representación.

- Syntax : String IntToStr(Int Value)

MessageBeep: Se envía una señal acústica especificado.

- Syntax Void MessageBeep(int type)

MessageBox: Muestra en la pantalla una ventana con texto y el título especificado. Se espera la confirmación del usuario.

- Syntax : Void MessageBox(String Text, String Title)

Mod: ejecuta la división de dos enteros.

- Syntax : Int Mod(int A, int B)

NumGateExists: Comprueba si la gate numérico especificado existe.

- Syntax : Bool NumGateExists(String Name, Int Id)

Play: Reproduce un archivo de audio con la extensión WAV

- Syntax : Void Play(String AudioFileName)

PlaySound: Reproduce un archivo de audio con extensión WAV o MP3.

- Syntax : Void PlaySound(String AudioFileName, bool WaitMode, int RepeatCounter)

Pow: Elevar al exp (x^n)

- Syntax : Real Pow(Real Base, Real Exponent)

MessageBox: Muestra en la pantalla una ventana con texto y el título especificado. Se espera que el usuario elija Sí o No

- Syntax : Bool MessageBox(String Text, String Title)

Rand: saca un valor aleatorio generado de rango especificado.

- Syntax : Int Rand(Int Range)

RealToInt: Convierte un valor real en un entero (descartando decimales)

- Syntax : int RealToInt(Real Value)

RealToStr: Convierte un valor real a su representación de cadena.

- Syntax : String RealToStr(Real Value)

ReportAppendRecord: Este comando se utiliza para DAT informe configurado para "Guardar registro de comando". Cuando se ejecuta este comando, un nuevo registro se adjunta al presente informe el archivo, si el archivo no existe, será creado.

- Syntax :ReportAppendRecord(String Name)

ReportCreate: Se elabora un Informe de archivo.

- Syntax :Bool ReportCreate(String Name)

Sin: Saca seno de un valor real (ángulo en radianes).

- Syntax :Bool ReportCreate(String Name)

Sleep: Se espera un período de tiempo expresado en milisegundos.

- Syntax :Void Sleep(int MilliSeconds)

Sqrt: Saca la raíz cuadrada de un valor real.

- Syntax :Real Sqrt(Real Value)

StopFunction: Deja de correr una función.

- Syntax :Int StopFunction (String FunctionName)

StopSound: Detiene sonido que comenzó con la función PlaySound.

- Syntax :Void StopSound()

StrGateExists: Comprueba si la cadena especificada gate existe.

- Syntax: Bool StrGateExists(String Name, Int Id).

String: cadena de caracteres con diferente longitud.

Tan : Saca la tangente de un valor real (ángulos en radianes)

- Syntax: Real Tan(Real Value)

TemplateAlarmsStatus: Abre la plantilla de estado de alarmas.

- Syntax: void TemplateAlarmsStatus()

TObjGetH: Devuelve la altura del objeto especificado.

- Syntax: Int TObjGetH(Int Id).

TObjGetStatus: Devuelve el estado de el objeto especificado.

- Syntax: Int TObjGetStatus(Int Id).

TObjGetW: Devuelve la anchura del objeto especificado.

TObjGetX: Devuelve la abscisa del objeto especificado a través de su identificador.

- Syntax: String TObjGetText(Int Id)

TObjSetStatus: Cambia el estado de el objeto especificado.

- Syntax: Void TObjSetStatus(Int Id, Int status)

TPageClose: Se cierra la plantilla de página especificada.

- Syntax: void TPageClose(Integer Page).

TPageOpen: Se abre una página de la plantilla con un nombre específico a la misma.

- Syntax: int TPageOpen(String Name)

WindowIsOpen: Abre una nueva plantilla cuando esta es llamada

- Syntax: Bool WindowIsOpen()

9. Conclusiones

- Por tanto vemos que la simulación realizada con el prototipo, mostro la efectividad de manipular el proceso de llenado de envases con el sistema SCADA brindando mayor rentabilidad y control de su proceso para la empresa.
- Se implementó un prototipo SCADA en el envasado de líquidos, el cual sule algunas necesidades que se tenían en el envasado de materias primas en la industria de productos químicos y farmacéuticos, facilitando más control sobre los tiempos empleados para el embasado del producto.
- El aporte fundamental fue comparar la efectividad del proceso de llenado, entre los tiempos reales del proceso y los tiempos simulados, teniendo como resultado una eficiencia notable con el prototipo SCADA, dando viabilidad a la implementación de este prototipo.
- Los operarios encargados de la manipulación de la llenadora debían llenar un formato especial a mano de las cantidades realizadas, con este nuevo sistema se genero un archivo que se actualiza cada vez que se pone en marcha el proceso de llenado de cualquier producto, mostrando la fecha, la hora, el operario, el producto envasado y número de lote, desplazando a si el actual formato de reportes, el cual es obligatorio para presentarse como soporte al Invima.(instituto nacional de vigilancia de medicamentos y alimenticios) y a su vez servirá como base de datos de los materiales manipulados por cada empleado.
- Como actualmente Athos busca cada vez mas optimizar la producción de la materia prima en sus empresa, es una oportunidad posible mostrar con un sistema SCADA, la facilidad que se tiene de manipular los procesos de las materias primas a través de un entorno visual ubicado en una PC portátil o de escritorio; haciendo tanto para el operador como para el supervisión más ameno el control de los procesos.

-

10. RECOMENDACIONES

- Diseñar un dispositivo de envasado con características especiales que permitan el manejo seguro de sustancias altamente inflamables y corrosivas como son: el hipoclorito de sodio, ácido nítrico, ácido sulfúrico, formol, como alternativa para disminuir los riesgos ocupacionales tanto a los empleados como la planta física de laboratorios Athos.
- Implementar un sistema de refrigeración que permita un rápido choque térmico (temperaturas menores a -5°C) para mejorar los tiempos de preparación en el área de perfumería.
- Mejorar el sistema actual de envasado de los productos semisólidos, como son: vaselina, lanocrema, mentol, manteca de cacao, entre otros productos. Este se lograría con la implementación de una envasadora con sistema de control térmico el cual garantizará pasar de un estado semisólido a líquido para disminuir los tiempos de emvasado.

11. ANEXOS

REPORTE DEL SISTEMA DE PRODUCCION CON EL OPERADOR EN TIEMPO REAL						
dd/MM/yyyy H:mm:ss	operador0	materia_pri	cantidad_ml	cantidad_en	productos_e	
27/10/2009 08:23	08:25	JUAN GUILLE	AGUA DE RO:	240	1300	1
27/10/2009 08:23	08:25	JUAN GUILLE	AGUA DE RO:	240	1300	2
27/10/2009 08:23	08:25	JUAN GUILLE	AGUA DE RO:	240	1300	3
27/10/2009 08:23	08:25	JUAN GUILLE	AGUA DE RO:	240	1300	4
27/10/2009 08:23	08:25	JUAN GUILLE	AGUA DE RO:	240	1300	5
27/10/2009 08:23	08:25	JUAN GUILLE	AGUA DE RO:	240	1300	6
27/10/2009 08:24	08:26	JUAN GUILLE	AGUA DE RO:	240	1300	7
27/10/2009 08:24	08:26	JUAN GUILLE	AGUA DE RO:	240	1300	8
27/10/2009 08:24	08:26	JUAN GUILLE	AGUA DE RO:	240	1300	9
27/10/2009 08:24	08:26	JUAN GUILLE	AGUA DE RO:	240	1300	10
27/10/2009 08:24	08:26	JUAN GUILLE	AGUA DE RO:	240	1300	11
27/10/2009 08:24	08:26	JUAN GUILLE	AGUA DE RO:	240	1300	12
27/10/2009 08:25	08:27	JUAN GUILLE	AGUA DE RO:	240	1300	13
27/10/2009 08:25	08:27	JUAN GUILLE	AGUA DE RO:	240	1300	14
27/10/2009 08:25	08:27	JUAN GUILLE	AGUA DE RO:	240	1300	15
27/10/2009 08:25	08:27	JUAN GUILLE	AGUA DE RO:	240	1300	16
27/10/2009 08:25	08:27	JUAN GUILLE	AGUA DE RO:	240	1300	17
27/10/2009 08:25	08:27	JUAN GUILLE	AGUA DE RO:	240	1300	18
27/10/2009 08:26	08:28	JUAN GUILLE	AGUA DE RO:	240	1300	19
27/10/2009 08:26	08:28	JUAN GUILLE	AGUA DE RO:	240	1300	20
27/10/2009 08:26	08:28	JUAN GUILLE	AGUA DE RO:	240	1300	21
27/10/2009 08:26	08:28	JUAN GUILLE	AGUA DE RO:	240	1300	22
27/10/2009 08:26	08:28	JUAN GUILLE	AGUA DE RO:	240	1300	23
27/10/2009 08:26	08:28	JUAN GUILLE	AGUA DE RO:	240	1300	24
27/10/2009 08:27	08:29	JUAN GUILLE	AGUA DE RO:	240	1300	25
27/10/2009 08:27	08:29	JUAN GUILLE	AGUA DE RO:	240	1300	26
27/10/2009 08:27	08:29	JUAN GUILLE	AGUA DE RO:	240	1300	27
27/10/2009 08:23	08:29	JUAN GUILLE	AGUA DE RO:	240	1300	28
27/10/2009 08:23	08:29	JUAN GUILLE	AGUA DE RO:	240	1300	29
27/10/2009 08:23	08:29	JUAN GUILLE	AGUA DE RO:	240	1300	30
27/10/2009 08:23	08:30	JUAN GUILLE	AGUA DE RO:	240	1300	31
27/10/2009 08:23	08:30	JUAN GUILLE	AGUA DE RO:	240	1300	32
27/10/2009 08:23	08:30	JUAN GUILLE	AGUA DE RO:	240	1300	33
27/10/2009 08:23	08:30	JUAN GUILLE	AGUA DE RO:	240	1300	34
27/10/2009 08:23	08:30	JUAN GUILLE	AGUA DE RO:	240	1300	35
27/10/2009 08:23	08:30	JUAN GUILLE	AGUA DE RO:	240	1300	36
27/10/2009 08:23	08:31	JUAN GUILLE	AGUA DE RO:	240	1300	37
27/10/2009 08:23	08:31	JUAN GUILLE	AGUA DE RO:	240	1300	38

BIBLIOGRAFÍA

- ANDIDET S.A. [en línea] [citado en 17 de Agosto 2007], Disponible en Internet <<http://anditel.com.co/mapa.htm>. >
- Electrónica Unicrom. Historia del plc [en línea][citado en 11 de Agosto 2007],Disponible en Internet <http://www.unicrom.com/art_historia_PLC.asp>
- Grupo-máster, Funciones básicas de un plc [en línea],[citado en 11 de Agosto 2007],disponible en Internet <http://www.grupo-maser.com/PAG_Cursos/Auto/auto2/auto2/PAGINA_PRINCIPAL/PLC/plc.htm >
- Grupo tecnológico, módulos entrada salida [en línea] ,[citado en 23 de Agosto 2007],Disponible en Internet <http://www.grupo-maser.com/PAG_Cursos/Auto/auto2/auto2/PAGINA%20PRINCIPAL/PLC/ESTRUCTURAS/ESTRUCTURA%20INTERNA/INTERFACES/interfaces.htm >
- Josep Balcells ,Automatas programables [en línea],[citado en 22 de Septiembre 2007],Disponible en internet <<http://books.google.com/books?id=xfSjADge70C&printsec=frontcover&hl=es#v=onepage&q&f=false>>
- Parque Manizales [en línea] [citado en 17 de Agosto 2007], Disponible en Internet <<<http://parque.manizales.unal.edu.co/parque/portal/index.php?option=content&task=view&id=55&Itemid><http://parque.manizales.unal.edu.co/parque/portal/index.php?option=content&task=view&id=55&Itemid>=.>
- Proexport .Guía inversión extranjera [en línea] [citado en 13 de Agosto 2007],Disponible en Internet <<http://www.proexport.com.co/invest/GuiaInversionExtranjera2006/>. >
- QUIMBAYA TELEMTRIC. [en línea][citado en 3 Marzo 2007],Disponible en Internet .<www.qutele.com/index.htm>
- SENA ,memorias "Seminario de transferencia tecnologica sistemas scada" SENA agosto 2006, Medellin

- wiki pedía .sensor inductivo [en línea] [citado en 13 de Agosto 2007], Disponible en Internet
<http://es.wikipedia.org/wiki/Sensor_inductivo>