

TRABAJO ESPECIAL DE GRADO

AMPLIACIÓN Y ACTUALIZACIÓN DEL SISTEMA SCADA DEL LABORATORIO DE REDES DE DISTRIBUCIÓN DE LA ESCUELA DE INGENIERÍA ELÉCTRICA DE LA UCV

Presentado ante la Ilustre
Universidad Central de Venezuela
Por el Br. Di Yorio O., Carlos A.
para optar al título de
Ingeniero Electricista

Caracas, 2008

TRABAJO ESPECIAL DE GRADO

AMPLIACIÓN Y ACTUALIZACIÓN DEL SISTEMA SCADA DEL LABORATORIO DE REDES DE DISTRIBUCIÓN DE LA ESCUELA DE INGENIERÍA ELÉCTRICA DE LA UCV

Prof. Guía: Ing. Tamara Pérez
Tutor Industrial: Ing. Dickson Mora

Presentado ante la Ilustre
Universidad Central de Venezuela
Por el Br. Di Yorio O., Carlos A.
para optar al título de
Ingeniero Electricista

Caracas, 2008

CONSTANCIA DE APROBACIÓN

Caracas, 29 de mayo de 2008

Los abajo firmantes, miembros del Jurado designado por el Consejo de Escuela de Ingeniería Eléctrica, para evaluar el Trabajo Especial de Grado presentado por el Bachiller Carlos A. Di Yorio O., titulado:

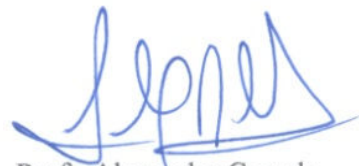
“AMPLIACIÓN Y ACTUALIZACIÓN DEL SISTEMA SCADA DEL LABORATORIO DE REDES DE DISTRIBUCIÓN DE LA ESCUELA DE INGENIERÍA ELÉCTRICA DE LA UCV”

Consideran que el mismo cumple con los requisitos exigidos por el plan de estudios conducente al Título de Ingeniero Electricista en la mención de Electrónica, y sin que ello signifique que se hacen solidarios con las ideas expuestas por el autor, lo declaran APROBADO.



Prof. Simón Morales.

Jurado



Prof. Alexander Cepeda

Jurado



Prof. Tamara Pérez A.

Prof. Guía

DEDICATORIA

A mis Padres Carlo y Ana.

A mi Novia Mónica.

A mi Hermana Maricarmen.

A mi Familia.

RECONOCIMIENTOS Y AGRADECIMIENTOS

A mis padres por todo el apoyo y soporte a lo largo de mi vida.

A la Profesora Tamara Pérez por su ayuda en general y asesoría.

Al Ingeniero Dickson Mora por su asesoría y contribución.

A mi novia por el apoyo y ayuda prestada en la elaboración de este trabajo.

Al Profesor Simón Morales por todos los favores y ayuda prestada.

A los compañeros Gustavo Gómez, Andrés Avendaño y todos aquellos que estuvieron directa o indirectamente relacionados con el desarrollo del Trabajo.

A todos en la empresa Wonderware, en especial a Arnoldo Rodríguez y Hender Molina por los favores, cursos dictados y ayuda en general.

Al Profesor Froilán Lozada por las licencias de productos Microsoft.

Al Profesor Luis Fernández por la ayuda en la resolución de conflictos con la red de la escuela.

Al Profesor A. Cepeda, J. Molina y W. Malpica por la colaboración prestada en el laboratorio de Máquinas.

A todos los compañeros y amigos, como Joao Farinha, Luis León, Alejandro González, Alejandro Ramírez, Gemma Torres, Servando Álvarez, Dany Abreu, Jorge Herrera y Cesar Roldán, que estuvieron involucrados con mi formación y me apoyaron con su amistad desinteresada.

A todos los profesores de la Facultad y de la Escuela de Ingeniería Eléctrica, en especial a Marta Serpa, Mary Power, Robustiano Gorgal y Manuel Marichal.

Di Yorio O., Carlos A.

**AMPLIACIÓN Y ACTUALIZACIÓN DEL SISTEMA SCADA DEL
LABORATORIO DE REDES DE DISTRIBUCIÓN DE LA EIE-UCV**

Profesor Guía: Ing. Tamara Pérez. Tutor Industrial: Ing. Dickson Mora. Tesis. Caracas. U.C.V. Facultad de Ingeniería. Escuela de Ingeniería Eléctrica. Ingeniero Electricista. Opción: Electrónica. Institución: Wonderware of Venezuela, Inc. Trabajo de Grado. 2007. 122 páginas.

Palabras Claves: Automatización; Control Distribuido; SCADA; HMI; Wonderware.

Resumen. Se configuró un Sistema SCADA y un HMI basado en la tecnología ArchestrA de Wonderware. La estructura de la Arquitectura Distribuida realizada contiene un nodo configurado como Servidor de Objetos de Aplicaciones y Base de Datos de Configuración, otro Redundante del primero, un nodo Historiador, tres nodos HMI y dos I/O Servers para la comunicación con, una Maqueta de Procesos controlada por un PLC DirectLogic, unas RTUs Ducati que adquieren los parámetros eléctricos de la EIE-UCV y el SCADA realizado bajo iFix. Los Servidores de Objetos de Aplicación son configurados empleando el IAS, el HMI es realizado bajo InTouch y el Historiador con InSQL, todos productos de Wonderware. Para la comunicación con la Maqueta, las Remotas y el iFix, se emplea un Servidor OPC llamado TOPServer, el Gateway Modbus TCP y el FactorySuite Gateway respectivamente. La implementación del SCADA se realiza con éxito, dejando una arquitectura base para futuras actualizaciones y ampliaciones.

ÍNDICE GENERAL

	Pág.
DEDICATORIA	ii
RECONOCIMIENTOS Y AGRADECIMIENTOS	iii
RESUMEN	iv
ÍNDICE GENERAL	v
LISTA DE TABLAS	viii
LISTA DE FIGURAS, ILUSTRACIONES O GRÁFICAS	ix
SIGLAS	xii
ACRÓNIMOS	xiii
INTRODUCCIÓN	1
PLANTEAMIENTO DEL PROBLEMA	4
OBJETIVOS	5
CAPITULO I	
FUNDAMENTOS TEÓRICOS	6
1.1. AUTOMATIZACIÓN INDUSTRIAL	6
1.2. REDES DE CAMPO	9
1.2.1. Generalidades sobre las Redes de Campo	9
1.2.2. Clasificación de las Redes de Campo	12
1.2.2.1. Pirámide de Áreas de Aplicaciones en Automatización	12
1.2.2.2. Redes Field and Host	14
1.2.2.3. Tipos de Buses	17
1.3. MODELO DE REFERENCIA OSI	19
1.3.1. Modelo de Referencia OSI	19
1.3.1.1. Capa Física (Capa 1)	20
1.3.1.2. Capa de Enlace (Capa 2)	24
1.3.1.3. Capa de Red (Capa 3)	25
1.3.1.4. Capa de Transporte (Capa 4)	25
1.3.1.5. Capa de Sesión (Capa 5)	26
1.3.1.6. Capa de Presentación (Capa 6)	26

1.3.1.7. Capa de Aplicación (Capa 7)	27
1.3.1.8. Funcionamiento entre Capas	27
1.3.2. Descripción Básica de algunos Protocolos y/o Redes de Campo	29
1.3.2.1. Modbus	29
1.3.2.2. OLE for Process Control (OPC)	31
1.3.2.3. DirectNET	32
1.3.2.4. K-Sequence	34
1.3.2.5. ECOM	34
1.3.2.6. Ducbus	35
1.4. DISPOSITIVOS Y PROGRAMAS PARA AUTOMATIZACIÓN INDUSTRIAL Y/O SCADA	35
1.4.1. Dispositivos para Control Industrial	36
1.4.1.1. Controladores Lógicos Programables (PLC)	36
1.4.1.2. Controladores de Automatización Programable (PAC)	38
1.4.1.3. Computadores Industriales	40
1.4.2. Sistemas de Transmisión y Recolección de Datos	40
1.4.2.1. Ducati MACH 30	41
1.4.3. Programas Wonderware para SCADA y HMI	42
CAPITULO II	
DESCRIPCIÓN DEL SCADA	45
2.1. IMPLEMENTACIÓN ANTERIOR DEL SCADA	45
2.2. LIMITACIONES ENCONTRADAS EN EL SCADA ACTUAL Y DISPOSITIVOS A AÑADIR	47
2.2.1. Adquisición de Parámetros Eléctricos	48
2.2.2. Actualización anterior del SCADA bajo iFix	49
2.2.3. Maqueta de Procesos	52
2.2.4. Computadores para la actualización del SCADA	54
2.3. DESCRIPCIÓN DEL SOFTWARE EMPLEADO	56
2.3.1. Computadores Restantes del SCADA	56
2.3.2. Programación del PLC DirectLogic de la Maqueta de Procesos	57
2.3.3. Componentes de Configuración para una Arquitectura con Wonderware	60
2.3.3.1. Application Object Server (Servidor de Objetos de Aplicaciones)	60
2.3.3.2. Configuration Database (Base de datos de Configuración)	61
2.3.3.3. Engineering Station (Estación de Ingeniería)	61

2.3.3.4. Historian Node (Nodo Historiador)	62
2.3.3.5. Visualization Node (Nodo Visualizador)	62
2.3.3.6. I/O Server Node (Nodo de Servidores I/O)	62
2.3.3.7. Web Server (Servidor Web)	62
2.3.4. Pruebas de Comunicación	63
2.3.4.1. Comunicación entre el PLC y el SCADA	63
2.3.4.2. Comunicación entre las RTU y el SCADA	65
2.3.4.3. Comunicación entre el iFix y el SCADA	66
2.4. ESQUEMA FINAL Y DESCRIPCIÓN DE LOS EQUIPOS	68
CAPITULO III	
IMPLEMENTACIÓN DEL SCADA CON WONDERWARE	73
3.1. CONFIGURACIÓN DEL SOFTWARE PARA LA ADQUISICIÓN DE DATOS	75
3.1.1. TOPServer	75
3.1.2. Gateway Modbus TCP	78
3.1.3. Servidor OPC del iFix	78
3.1.4. FactorySuite Gateway	80
3.1.5. Programa Cerrar	81
3.2. CONFIGURACIÓN DEL SOFTWARE PARA LA ARQUITECTURA DEL SCADA	82
3.2.1. Industrial Application Server 2.1 Patch 2	82
3.2.2. Industrial SQL Server 9.0 y Alarm DBLogger Manager	89
3.3. CONFIGURACIÓN DEL SOFTWARE PARA EL HMI	92
3.3.1. ActiveFactory 9.2	92
3.3.2. InTouch 9.5 SP1	93
CAPITULO IV	
VALIDACIÓN Y PRUEBAS DEL SCADA	98
CONCLUSIONES	107
RECOMENDACIONES	109
REFERENCIAS BIBLIOGRÁFICAS	112
BIBLIOGRAFÍAS	116
GLOSARIO	120
ANEXOS	123

LISTA DE TABLAS

	Pág.
Tabla 1. Características principales de las redes Field and Host.	16
Tabla 2. Resumen de las Capas OSI.	27
Tabla 3. Características de los computadores del LRD.	49
Tabla 4. Características de los computadores y su función para el SCADA.	55
Tabla 5. Convención de nombres del SCADA.	A4-1
Tabla 6. Contraseñas de los Computadores del SCADA.	A6-1

LISTA DE FIGURAS, ILUSTRACIONES O GRÁFICAS

	Pág.
Figura 1. Telar de Jacquard.	7
Figura 2. Modelo de la Hiladora “Spinning Jenny”.	7
Figura 3. Diferentes tipos de Interfaces empleadas a lo largo de la historia.	9
Figura 4. Evolución de las Redes de Campo.	11
Figura 5. Pirámide de Áreas de Aplicaciones de Redes de Automatización.	12
Figura 6. Esquema de Conexión Típico para redes Field-Host.	14
Figura 7. Esquema de Conexión Distribuido Típico para redes Field-Host.	16
Figura 8. Comparación según Funcionalidad y Tipo de datos de los Buses.	17
Figura 9. Comparación de las Redes comerciales más comunes.	17
Figura 10. Capas del Modelo de Referencia OSI.	20
Figura 11. Topología de Redes más Comunes.	22
Figura 12. Topología de Redes más Comunes.	28
Figura 13. Descripción de una Solicitud Modbus.	30
Figura 14. Descripción de Solicitudes de lectura y escritura en DirectNET.	33
Figura 15. Algunas configuraciones con los 4 tipos de bases del PLC DL205.	38
Figura 16. Comparación de PAC con PC y PLC.	39
Figura 17. Hardware PAC.	40
Figura 18. Algunos Transmisores y Recolectores de Datos.	41
Figura 19. Ducati Mach30.	41
Figura 20. Áreas de una automatización industrial involucradas por ArchestrA.	42
Figura 21. Imágenes de los paquetes InTouch (WindowMaker y WindowViewer), Industrial Application Server (IDE) e IndustrialSQL Server (SMC Console).	44
Figura 22. Esquema del SCADA actual.	45
Figura 23. Algunas Plantillas del HMI.	47
Figura 24. Esquema inicial propuesto para la actualización del SCADA.	47

Figura 25. Banco de Pruebas y Canalización realizada.	49
Figura 26. Esquemático del Convertidor 232/485 diseñado.	50
Figura 27. Board en el programa Eagle y Ambas caras del Circuito Impreso.	51
Figura 28. Instalación del Switch.	52
Figura 29. Maqueta, Bomba de Vaciado (Azul claro) y Tanque de depósito.	53
Figura 30. Vista Externa e interna del cajetín y Variador de Frecuencia en detalle.	54
Figura 31. Esquema posterior a las modificaciones iniciales del SCADA.	56
Figura 32. Esquema general de la Maqueta de Procesos y su relación con el PLC.	58
Figura 33. Graficet para el Control de la Maqueta.	59
Figura 34. PLC de la Maqueta y Prueba del TOPServer.	64
Figura 35. Pantalla de Inicio Original y Modificada del iFix.	67
Figura 36. Esquema Final de la distribución física del SCADA.	68
Figura 37. Esquema Final del SCADA según niveles.	69
Figura 38. Modelo de Planta.	73
Figura 39. Configuración de la comunicación con el PLC de la Maqueta.	75
Figura 40. Configuración de uno de los Tags de la Maqueta.	76
Figura 41. Configuración de la comunicación con el Gateway de las Remotas.	77
Figura 42. Tags para la comunicación con el Gateway y la configuración de uno.	77
Figura 43. Dos vistas del Gateway Modbus TCP.	78
Figura 44. SCU del iFix y Power Tool con el servidor OPC.	79
Figura 45. Configuración del DataBlock Prueba2 y de uno del grupo iA8.	79
Figura 46. Configuración del Nodo OPC en el FSGateway.	80
Figura 47. Device Items añadidos y un diagnóstico de la comunicación.	81
Figura 48. Programa Cerrar configurado para el TOPServer y el Gateway.	81
Figura 49. Inicio del IDE y vista final de las Plataformas de la Galaxia Neutron.	83
Figura 50. Model, Deployment y Derivation View de la Galaxia Neutron.	84
Figura 51. Pestaña “General” y “Engine” de la Plataforma Scada0.	85
Figura 52. Pestañas “General” y “Redundancy” del Engine_0.	86
Figura 53. Pestañas “General”, “Scripts” y “Scan Group” del Cliente OPC.	86
Figura 54. Pestañas “General”, “Scripts” y “Topic” del FactorySuite Gateway.	87

Figura 55. Pestañas “Field Attributes” y “Scripts” del Objeto Remota13.	88
Figura 56. Objeto “Maqueta” y algunas pestañas de “Temperatura_RTD”.	88
Figura 57. Una lista de observación de la Galaxia definida en el Object Viewer.	89
Figura 58. Parámetros de Configuración y Estado de los módulos del InSQL.	90
Figura 59. Grupo privado creado y algunos parámetros de la Consola.	91
Figura 60. Algunas vistas del AlarmDBLogger Manager y el Purge/Archive.	91
Figura 61. Conexión con el InSQL y una vista del Trend en funcionamiento.	92
Figura 62. Plantilla Base y Plantilla de Inicio en ejecución.	94
Figura 63. Plantilla de la Maqueta de Procesos en ejecución.	94
Figura 64. Algunos elementos básicos de InTouch y Symbol Factory.	95
Figura 65. Plantilla “Remotas” en ejecución.	96
Figura 66. Plantilla Historiador y Alarma en ejecución.	97
Figura 67. Plantilla de Comunicación con iFix y menú de un cuadro de Texto.	97
Figura 68. Tanque, Cajetín y Plantilla Maqueta durante la fase de llenado.	99
Figura 69. Cajetín de la Maqueta y Control manual por el HMI.	99
Figura 70. Imágenes de la Plantilla Maqueta en dos casos de Advertencia.	100
Figura 71. Gráfica de una semana de la data almacenada.	101
Figura 72. Plantilla iFix y Remotas al ocurrir fallas en la comunicación.	105
Figura 73. Plantilla Inicio al ocurrir algunas fallas de comunicación.	106
Figura 74. Programa Correcto e Incorrecto en el PLC.	106
Figura 75. Flujo de Trabajo sugerido por Wonderware.	A1-1
Figura 76. Imagen y Board del Convertidor 232/485.	A2-1
Figura 77. Esquema de la Maqueta de Procesos.	A3-1
Figura 78. Configuración del enlace de comunicación.	A3-2
Figura 79. Primera parte del Diagrama de Escaleras.	A3-3
Figura 80. Segunda parte del Diagrama de Escaleras.	A3-4
Figura 81. Configuración del EDRUM.	A3-5
Figura 82. Última parte del Diagrama de Escaleras.	A3-6
Figura 83. Esquema general del SCADA.	A4-1
Figura 84. Ventana de Delphi para la programación del programa Cerrar.	A5-1

SIGLAS

ARC: Automation Research Corporation (Corporación de Investigación de Automatización).

EIE-UCV: Escuela de Ingeniería Eléctrica de la Universidad Central de Venezuela.

ISO: International Organization for Standardization (Organización Internacional para la Estandarización).

LCID: Laboratorio de Control, Instrumentación y Digitales.

LRD: Laboratorio de Redes de Distribución.

NEMA: National Electrical Manufacturers Association (Asociación Nacional de Fabricantes Eléctricos).

UCV: Universidad Central de Venezuela.

ACRÓNIMOS

ADU: Application Data Unit (Unidad de Datos de la Aplicación).

AOS: Application Object Server (Servidor de Objetos de Aplicaciones).

COM: Component Object Model (Modelo de Objetos de Componentes).

CPU: Central Process Unit (Unidad Central de Proceso).

DA: Data Access (Acceso de Datos).

DCOM: Distributed COM (COM distribuido).

DI: Device Integration (Integración de Dispositivos).

DCS: Distribute Control System (Sistema de Control Distribuido).

FF: Fieldbus Foundation.

GPRS: General Packet Radio Service (Servicio de Radio de Paquetes Generales).

HMI: Human Machine Interface (Interfaz Hombre Máquina).

IAS: Industrial Application Server (Servidor de Aplicaciones Industriales).

IDE: Integrated Development Enviroment (Ambiente de Desarrollo Integrado).

IO: In/Out (Entrada/Salida).

IPX: Internetwork Packet Exchange (Intercambio de Paquetes InterRed)

MODICON: Modular Digital Controller (Controlador Digital Modular).

OPC: Ole for Process Control (Ole para Control de Procesos).

OSI: Open System Interconnection (Interconexión de Sistemas Abiertos).

PAC: Programmable Automation Controller (Controlador de Automatización Programable).

PDU: Protocol Data Unit (Unidad de Datos del Protocolo).

PID: Proporcional Integral Derivativo.

PLC: Programmable Logic Controller (Controlador Lógico Programable).

RLL: Relay Ladder Logic (Lógica de Relés en Escalera)

RTU: Remote Terminal Unit (Unidad Terminal Remota).

SCADA: Supervisory Control and Data Acquisition (Control Supervisor y Adquisición de Datos).

SDS: Smart Distributed System (Sistema Distribuido Inteligente).

SMC: System Managment Console (Consola de Administración del Sistema).

SMS: Short Message Service (Servicio de Mensajes Cortos).

SO: Sistema Operativo.

TCP/IP: Transmission Control Protocol/Internet Protocol (Protocolo de Control de Transmisión/Protocolo de Internet).

XML: Extensible Markup Language (Lenguaje de Marcas Extensibles).

INTRODUCCIÓN

La industria se ha caracterizado por su continuo desarrollo y crecimiento. Desde su primer cambio radical durante la revolución industrial a finales del siglo XVIII, se buscó la forma de sustituir la lenta producción manual por una más eficiente y automatizada. La industria actual no escapa de este pensamiento; mientras más procesos manuales o semi-automatizados se realicen, habrá más retrasos e ineficiencias en la producción. Por esta causa muchas empresas se han visto en la necesidad de automatizarse, requiriendo la mano de obra humana para la supervisión, control y tareas no automatizables.

La necesidad de la industria de operar continuamente sin interrupciones a veces se ha visto mermada por la ausencia de un servicio eléctrico confiable, y aunque estén completamente automatizadas, algunas han optado por contar con plantas privadas de electricidad en sus procesos críticos. Es por ello que para prevenir y minimizar los efectos de los cortes y fallas del servicio, requieren de un continuo monitoreo de los parámetros eléctricos. Para realizar dicha supervisión normalmente se emplean RTU especializados en esta área, mientras que para la automatización son utilizados en su mayoría PLC y PAC (también llamados DCS) y, en ocasiones no críticas, los computadores industriales.

El incremento de las etapas automatizadas en las industrias ha impulsado la gran cantidad de instrumentos de medición y control de toda índole que se instalan cada día en las fábricas. Esto ha generado la necesidad de centralizar la supervisión y el control crítico de las plantas, logrando facilitar la labor a los operadores y supervisores, disminuyendo el costo asociado de operación e incrementando la eficiencia y producción de las mismas. Esta necesidad ha impulsado el desarrollo de sistemas SCADA cada día más robustos y confiables, y de HMI más amigables con el operador y que requieran de menos tiempo de aprendizaje para su operación.

La EIE-UCV tiene entre sus proyectos la actualización y mantenimiento de un SCADA para monitorear los parámetros eléctricos de las subestaciones de la UCV, así como también, para ampliar sus potencialidades prestando servicios a terceros en análisis de calidad de servicio, ahorro energético, entre otros. Este proyecto se ha venido desarrollando desde el año 2000, cuando se propuso y especificó un sistema SCADA teórico para la red de distribución eléctrica de la UCV [1].

Luego de varios trabajos intermedios, en el año 2003 se lleva a cabo un primer desarrollo práctico, cuando es realizado un Gateway para convertir el protocolo y medio de transmisión, de propietario sobre RS-485 a Modbus sobre TCP/IP, de las RTUs Ducati que adquieren los parámetros eléctricos (Tensiones, Corrientes, Factores de Potencia, Potencias, Energías y otros) de la red de distribución de la UCV [2]. En el año 2006 se realiza un sistema SCADA para el LRD sobre el paquete iFix de Intellution, utilizando los RTU antes mencionados como únicos sistemas a supervisar y monitorear por parte de la interfaz Hombre Máquina [3].

La Escuela de Ingeniería Eléctrica cuenta con una maqueta de procesos controlada por PLC, equipos Opto22 para control distribuido y otros. La idea básica del trabajo consiste en realizar una automatización de un ambiente industrial a pequeña escala dentro de la universidad. Para ello, se integrarán algunos de los dispositivos antes mencionados y el SCADA actual del laboratorio de Redes de distribución sobre iFix.

A diferencia de los trabajos de grado anteriores, este se realizará empleando la plataforma Wonderware para SCADA y HMI, a fin de analizar las diferencias entre iFix y este, e implementar un sistema basado en el Industrial Application Server por primera vez en una Universidad con sede en Caracas y por segunda vez en una Universidad Venezolana.

En el Capítulo I, se expondrán algunos de los conceptos básicos de automatización y redes industriales, así como de los dispositivos y programas empleados en este trabajo. En el Capítulo II, se describirá y explicará la arquitectura distribuida realizada y el esquema final de esta basada en los equipos y dispositivos utilizados. En el Capítulo III, se describirá y analizará, según el caso, los pasos seguidos para la implementación del SCADA con los Productos Wonderware. El Capítulo IV contiene la descripción de las validaciones y pruebas realizadas al Sistema SCADA implementado. Por último, se muestran las Recomendaciones y Conclusiones del Trabajo.

PLANTEAMIENTO DEL PROBLEMA

Este trabajo viene a representar la continuación de otros que se vienen realizando desde el año 2000 en la escuela. La propuesta nace ante la necesidad de ampliar, mejorar y actualizar el SCADA actual del laboratorio de redes de Distribución de la Escuela de Ingeniería Eléctrica utilizando los productos Wonderware para SCADA y HMI. La nueva versión del SCADA debe obtener los datos directamente de los dispositivos de campo y de forma indirecta a través del SCADA anterior realizado bajo iFix.

OBJETIVOS

OBJETIVO GENERAL

Ampliar y Actualizar el sistema SCADA actual del laboratorio de redes de Distribución de la Escuela de Ingeniería Eléctrica empleando la plataforma Wonderware, tal que permita visualizar la información existente añadiendo nuevos sistemas de Supervisión y Control, unificando las diferentes fuentes de información en una plataforma de software respetando la arquitectura actual y cumpliendo con la estructura básica de un sistema de control distribuido.

OBJETIVOS ESPECIFICOS

Identificar los PLC y RTU que formarán parte del SCADA.

Diseñar la estructura y organización del SCADA empleando la plataforma Wonderware.

Establecer la comunicación del SCADA con cada uno de los PLC y RTU.

Realizar la comunicación entre el SCADA y el sistema actual de supervisión de las remotas realizado con el paquete iFix de Intellution.

Implementar el historiador de datos y alarmas para las variables que lo requieran.

Configurar las alarmas y eventos.

Implementar la Interfaz Hombre-Máquina del SCADA basada en los productos Wonderware.

Verificar el completo funcionamiento del SCADA.

CAPITULO I

FUNDAMENTOS TEÓRICOS

1.1. AUTOMATIZACIÓN INDUSTRIAL

En esta sección se tratarán generalidades de la automatización industrial y una breve reseña cronológica de esta, enfatizando en la evolución de los Controladores Lógicos Programables y las Interfaces Hombre-Máquina, por ser los dispositivos más importantes asociados a la automatización industrial actual.

La automatización industrial se entiende como el uso de sistemas, técnicas y/o dispositivos para el control de un proceso de índole industrial de forma tal que este opere con ninguna o poca intervención humana. Esta ha disminuido el esfuerzo físico, necesidad sensorial y mental del hombre, discriminándolo generalmente a la supervisión y operación asistida de la maquinaria. Al ser independiente de las decisiones de las personas, la automatización ha generado incrementos y mejoras en la calidad, presión, repetitividad y eficiencia de los productos fabricados, además de una reducción del trabajo humano (sobre todo en ambientes de alto riesgo). [4]

Aunque la historia de la automatización se remonta a la era Pre-Cristiana, no fue hasta el año 1745 que se dio origen a la automatización industrial, cuando el francés Joseph Marie Jacquard (Lyon, Francia 1752 - Oullins, Francia 1834) inventó el primer telar controlado por tarjetas perforadas. Este autómatas posteriormente sería conocido como “Métier Jacquard” o en español como “Telar Jacquard”, que se muestra en la Figura 1. [5]



Figura 1. Telar de Jacquard. [5]

El primer auge importante de la automatización industrial ocurre en Inglaterra a partir de 1750, cuando surge la revolución industrial impulsada en sus inicios por la automatización de la industria textil y el desarrollo de los procesos del hierro. Durante esta era de gran desarrollo económico y tecnológico se realizan reformas al sistema agrícola, impulsado por la creación de la máquina de vapor y la hiladora multi-bobina denominada “Spinning Jenny” y el ferrocarril. La hiladora antes mencionada es mostrada en la Figura 2. [6]



Figura 2. Modelo de la Hiladora “Spinning Jenny”. [6]

Posterior a ese período se llevan a cabo una gran cantidad de descubrimientos e inventos en el área de la automatización, pero no es hasta en los años 50', con el surgimiento de los semiconductores cuando ocurre otra de las transformaciones importantes de la industria. A finales de los 60', la compañía automotriz Estadounidense General Motors oferta un concurso para reemplazar sus instalaciones de sistemas cableados por uno electrónico. Este concurso concluye en la creación del PLC por parte del Ingeniero Estadounidense Dick Morley (Massachusetts, USA) en 1968, quién funda la empresa Bedford Associates y resulta ganador del concurso. Desde los 70' con la invención del Microprocesador y su incorporación en la fabricación de los PLC, se hizo posible la programación sin necesidad de recablear, la comunicación con computadores centrales y efectuar cálculos aritméticos. [7] [8] [9]

En la actualidad, se cuenta con la cuarta generación de PLC que posee mejoras en las comunicaciones, mayor capacidad de almacenamiento y procesamiento respecto a las generaciones anteriores. No sólo el adelanto tecnológico ha sido respecto al PLC. También las Interfaces Hombre-Máquina (HMI), que son parte fundamental de un ambiente industrial (incluso más antiguas que los PLC), han pasado por tres etapas: (a) Batch (1945-1968), cuya característica principal fue el gran costo, gran cantidad de problemas, además del hecho de que la solicitud se hacía mediante tarjetas perforadas y la respuesta al usuario no se generaba en tiempo real. (b) Línea de Comandos (1969-1983), período que se caracterizó por la evolución a un sistema de transacciones pedido-respuesta en un vocabulario avanzado y en forma de línea de comandos, con mejoras en los tiempos de respuestas de días y horas a segundos. (c) Gráfica (1984 a la actualidad), cuya característica principal es interacción con el usuario por medio de una interfaz gráfica. Algunos tipos de interfaces empleadas a lo largo de la historia son mostradas en la Figura 3. [9] [10]



Figura 3. Diferentes tipos de Interfaces empleadas a lo largo de la historia. [10]

Hoy en día, la automatización abarca diferentes ramas de la ingeniería que van desde lo más básico como el diseño de sensores y actuadores, objetivos de la instrumentación industrial; pasando por los sistemas de control, los sistemas de transmisión y recolección de datos hasta llegar a los diferentes programas para SCADA e interfaces Hombre-Máquina.

1.2. REDES DE CAMPO

En esta sección se explicarán conceptos generales de las redes de campo, una breve reseña cronológica de éstas y algunas de las clasificaciones más comunes.

1.2.1. Generalidades sobre las Redes de Campo

En un principio, las redes de campo se conocían como un sistema de comunicación que interconecta al menos 2 dispositivos de forma analógica o digital con el fin de transmitir información para ser analizada y/o almacenada. Pero con el avance de la tecnología, las comunicaciones y los estándares de seguridad industrial, el concepto de redes de campos ha cambiado hasta el punto que en la actualidad una red de campo se conoce como un sistema de comunicación digital bidireccional que

permite la interconexión de múltiples instrumentos, computadores y otros, con el fin de realizar acciones de control, supervisión y administración de procesos.

Se ha impulsado el desarrollo y construcción de las redes de campo, ya que traen como ventajas la reducción de cableado, minimización de errores de comunicación por falla de cableado, reducción de costos de mano de obra, flexibilidad para realizar control distribuido, simplificación de cableado para nuevas instalaciones, reducción en los tiempos de reparación e instalación de nuevos dispositivos, posibilidad de conectar una amplia gama de HMI y software para SCADA comerciales, incremento de la confiabilidad de los sistemas de producción, entre otras.

Hoy en día suena imposible imaginar una automatización industrial sin una red que interconecte todos los dispositivos y permita realizar acciones de control y supervisión. Pero estas redes no siempre fueron así, en sus inicios el control de procesos se realizaba de forma manual y la indicación de los instrumentos era local, requiriendo gran cantidad de operadores y supervisores por zona para asegurar un buen funcionamiento.

Posteriormente surgen los controladores y registradores, trayendo consigo la aparición de los cuartos de control o “Control Room”, donde llegaba toda la información de forma neumática (y posteriormente de forma eléctrica), y que tenía como función primordial la centralización de la supervisión y monitoreo de los procesos. Esto trajo como consecuencia que en plantas muy grandes, se requiriera de una gran cantidad de cableado para lograr llevar todas las señales a los cuartos de control, lo que se traducía en un gran costo.

Desde los años 80', con la aparición de los dispositivos inteligentes de campo, se logra reducir el cableado, ya que la gran cantidad de dispositivos de comunicación se conectan a una misma red, lo que abarata significativamente los costos de

automatización. Esta tendencia es mucho mayor hoy en día, debido a que la gran cantidad de fabricantes de controladores y dispositivos de comunicaciones existentes en el mercado han generado una fuerte competencia, lo que se ha traducido en reducción de costos y mejoras en los servicios. Una evolución de las redes de campo desde el punto de vista gráfico es mostrada en la Figura 4.

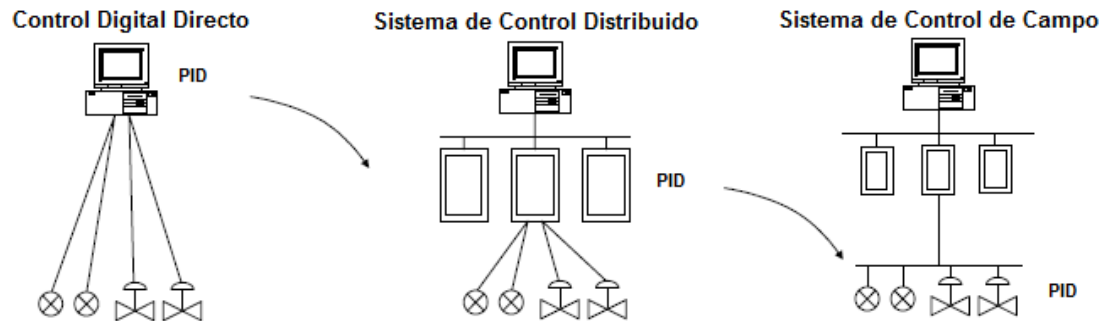


Figura 4. Evolución de las Redes de Campo. [11]

Para definir completamente una red hace falta seleccionar además del mensaje y el tipo de cable, la topología de la instalación y el protocolo de transmisión, que a su vez dependen de muchos factores tales como, la longitud de la instalación, el nivel de ruido presente, la alimentación de los dispositivos, el tipo de dispositivo a comunicar, las condiciones ambientales, la velocidad de la transmisión, la cantidad de dispositivos y muchos otros factores.

Para simplificar el trabajo de los instaladores, muchos de los fabricantes de dispositivos y organizaciones relacionadas han realizado múltiples pruebas a los diferentes tipos de redes industriales, logrando definir los valores máximos y estándares de estas. Así, cuando se escucha hablar de una red Modbus RTU, FF o de cualquier otro protocolo, no sólo se refiere al protocolo de comunicación, sino también a una cantidad de características previamente definidas por los diseñadores de estas redes que se deben cumplir para que funcionen adecuadamente. [11] [12]

1.2.2. Clasificación de las Redes de Campo

Antes de caracterizar una red de campo, es importante conocer su uso y clasificación desde el punto de vista Industrial. Existen diferentes tipos de clasificaciones, a continuación se detallan 3 de estas: La Pirámide de Áreas de Aplicaciones (Se enfoca en el Tiempo de respuesta y la Aplicación), Las redes Field-Host (Orientada exclusivamente en las aplicaciones) y por último, la clasificación según el Tipo de Bus (Enfocada a la Capacidades de las redes).

1.2.2.1. Pirámide de Áreas de Aplicaciones en Automatización

Una de las más comunes y que describe bien el ambiente Industrial es la denominada Pirámide de Áreas de Aplicaciones para Redes de Automatización. En esta, se clasifican las redes considerando dos características principales: los tiempos de respuesta, que van desde menores a 1 s hasta menores a 10 ms; y los niveles de automatización de planta, que van desde las redes gerenciales hasta los dispositivos de campos. La pirámide de Áreas de Aplicaciones de Redes de Automatización es mostrada en la Figura 5.

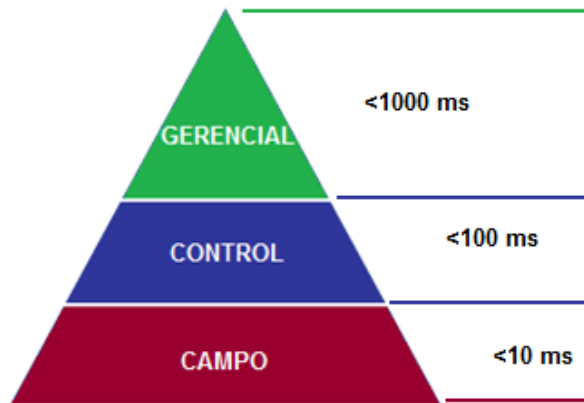


Figura 5. Pirámide de Áreas de Aplicaciones de Redes de Automatización.

Redes de Campo (fieldBus Networks): Aunque puede traer confusión este nombre, no debe confundirse con las Redes de campo (FieldBus en inglés, con “F” mayúscula al inicio) que engloban todas las redes de automatización. Estas se ubican

en la base de la pirámide. Su tiempo de respuesta o ciclo de bus es menor a 10 ms. Tienen como función interconectar todos los dispositivos de campo, desde los que requieren alimentación por bus para ser usados en ambientes peligrosos, tales como algunos posicionadores y transmisores; hasta dispositivos complejos, tales como Analizadores, Unidades Remotas I/O, entre otros. Las redes de campo y los protocolos más conocidos incluyen, FOUNDATION fieldbus (H1 y HSE), PROFIBUS (PA y DP), DeviceNet, Hart, Modbus RTU Seriplex, Interbus-S, LonWorks y AS-Interface.

Redes de control y LAN (Control System and Local Area Networks): Se ubican en el medio de la pirámide. Su tiempo de respuesta o ciclo de bus es menor a 100 ms. Tienen como función interconectar los dispositivos para control y monitoreo de planta, tales como PLC, Computadores Industriales, HMI, entre otros. Las redes de control y los protocolos utilizados más conocidos incluyen, Profibus FMS, ControlNet, FOUNDATION fieldbus (HSE) y Modbus TCP.

Redes Gerenciales (Plant-wide Networks): Se ubican en el tope de la pirámide. Su tiempo de respuesta o ciclo de bus es menor a 1000 s. Tienen como función primordial obtener la data de los dispositivos de control y de campo a los niveles gerenciales de la planta para el análisis y toma de decisiones. También pueden ejercer acciones de control directas sobre la planta, aunque en teoría no son empleadas para tal fin. La red Gerencial más conocida por excelencia es EtherNet (se refiere a la capa Física/Enlace, ya que existen gran cantidad de protocolos sobre esta).

Con la consolidación del Ethernet como estándar industrial, muchos de los fabricantes han optado por modificar o crear protocolos que implementan en sus equipos para comunicarse sobre esta red, esto ha ocasionado que en la mayoría de las empresas se cuenten con equipos de control, monitoreo, historiadores y otros que se comunican a través de Ethernet. Es por ello, que ha venido tomando cada vez más

importancia la clasificación de las redes industriales en lo que algunos autores denominan Redes Field and Host. [13] [14]

1.2.2.2. Redes Field and Host

Las Redes Field and Host, expresan una forma más simplificada y sencilla de clasificar las redes industriales. En esta, se clasifican las redes según las necesidades o aplicaciones de cada uno de los dispositivos que intervienen en el control, supervisión y, en general, en la automatización de una planta. Una imagen que contiene un esquema típico de conexión que permite diferenciar las redes de nivel Host (Host-Level) con las de nivel Field (Field-Level) es mostrada en la Figura 6.

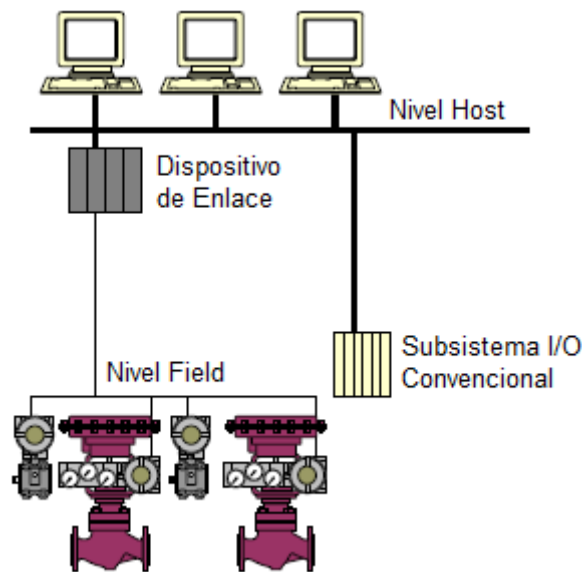


Figura 6. Esquema de Conexión Típico para redes Field-Host.

Nivel Field (Field Level o Nivel de Campo): En este nivel se incluyen todos aquellos dispositivos que poseen aplicaciones específicas, tales como, válvulas, posicionadores, algunos transmisores y en general dispositivos finales de control. Los protocolos dominantes para instrumentos de procesos en el Nivel de Campo incluyen HART, FOUNDATION Fieldbus H1 y PROFIBUS PA.

Estas redes fueron diseñadas teniendo en mente que la longitud de los cables de transmisión debían ser largos para abarcar la mayor cantidad de dispositivos posibles y poder llevar la información desde este nivel hasta los cuartos de control, controladores o dispositivos de enlace que se encuentran distantes; y para que los dispositivos de campo pudieran tomar la potencia que requieren desde la red. Es por ello, que este nivel los instrumentos y dispositivos aparecen en grandes cantidades, y a menudo se encuentran cientos o miles de estos.

Los diseñadores de estas redes tuvieron en cuenta muchos aspectos para mantener bajos los costos de modernización e instalación de nuevas redes. Es por ello, que estas no necesitan de un cableado especial y su velocidad es moderada.

Nivel Host (Host Level o Nivel de Anfitrión): En este nivel se incluyen el resto de los dispositivos que no pertenecen al nivel de campo. Estos poseen aplicaciones variadas que van desde Estaciones de trabajo (Workstation), HMI y dispositivos de supervisión, hasta Controladores, PLC y dispositivos de Enlace (Gateway). En este nivel, las redes basadas en cableado Ethernet son el estándar dominante de la industria. Existen gran cantidad de protocolos basados en Ethernet, tal como FOUNDATION Fieldbus HSE, PROFINet, Modbus TCP y muchos otros.

Con la mejora de las redes de campo, se produjo una explosión de información por la posibilidad de obtener gran cantidad de data de los dispositivos de campo. Esto originó que las redes de control propietarias quedaran relegadas a un segundo plano por la poca capacidad que poseían de manejar rápidamente grandes cantidades de información. A diferencia de las redes propietarias, las basadas en Ethernet se abrieron paso en el mundo industrial por su capacidad de transferir grandes cantidades de información a una velocidad muy alta. Esto mejoró significativamente el control de planta e históricos generados, añadiendo nuevas capacidades tales como diagnóstico, mantenimiento y configuración remota. [11]

Un ejemplo de esquema de conexión distribuido típico para redes Field-Host es mostrado en la Figura 7.

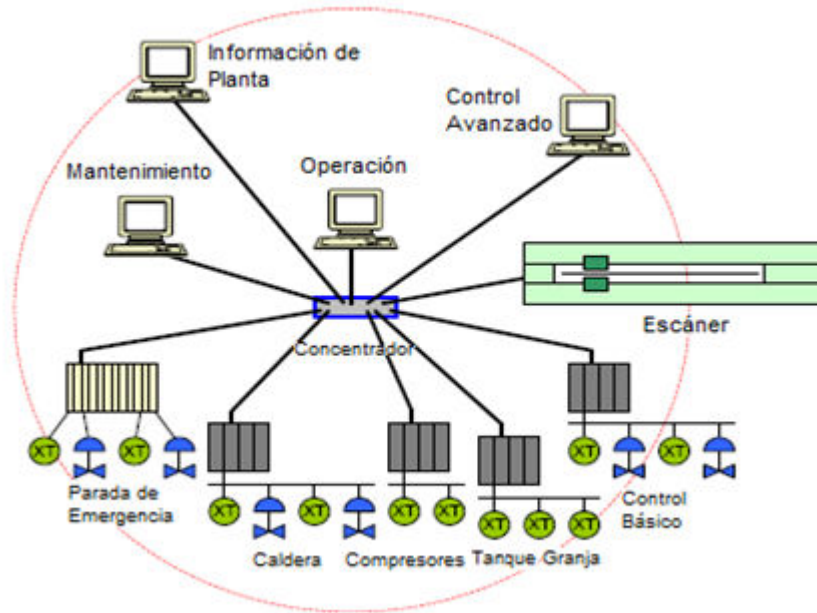


Figura 7. Esquema de Conexión Distribuido Típico para redes Field-Host.

En la Tabla 1, se realiza un comparación de las características principales de las redes Field and Host. [11]

Característica	Nivel Field	Nivel Host
Velocidad	Baja	Alta
Distancia	Larga	Corta**
Redundancia de medio	No	Si
Comunicación Multipunto	Si	No
Alimentación por Bus	Si	No
Seguridad Intrínseca	Si	No
** Se puede aumentar si se emplea Fibra Óptica.		

Tabla 1. Características principales de las redes Field and Host.

1.2.2.3. Tipos de Buses

Quizás la forma más común de clasificar las redes de campo sea dependiendo del tipo de bus. En la actualidad existen tres tipos de buses orientados a campo: SensorBus, DeviceBus y FieldBus; aunque algunos autores añaden otro tipo de bus adicional llamado ControlBus, que en realidad no vendría siendo parte de los Buses orientados a campo, pero es incluido debido a que posee una parte de la funcionalidad de estos. Una comparación entre los tipos de Buses es mostrada en la Figura 8.

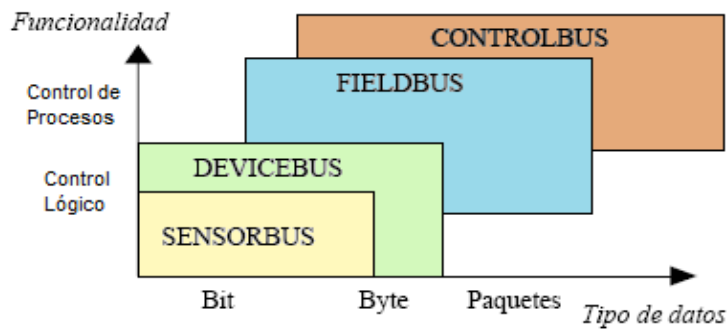


Figura 8. Comparación según Funcionalidad y Tipo de datos de los Buses.

A continuación se señalan las características principales de los distintos tipos de buses existentes y se muestra una imagen con una comparación de las redes comerciales más comunes en la Figura 9.

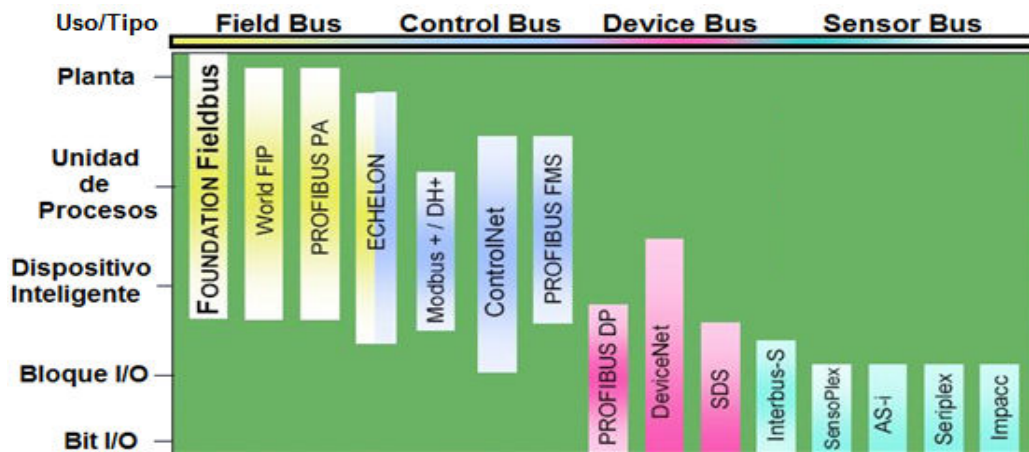


Figura 9. Comparación de las Redes comerciales más comunes. [15]

SensorBus (Bus de Sensores):

- La Información es transmitida en bits.
- Aplicaciones Discretas: Todas las Variables son digitales.
- El control típico es PLC/PC.
- El tiempo de respuesta es menor a 5 ms y no presenta diagnóstico.
- Se conectan captadores, sensores de proximidad, actuadores, botoneras, interruptores y otros con un controlador central.
- Su función principal es la de distribuir las Entradas/Salidas digitales.
- Los SensorBus más conocidos incluyen ASi, FlexIO e Interbus-S.

DeviceBus (Bus de Dispositivo):

- La Información es transmitida en bytes.
- Aplicaciones Discretas: La mayoría de las Variables son digitales, aunque algunas son analógicas.
- El control típico es PLC/PC.
- El tiempo de respuesta es menor a 5 ms.
- El diagnóstico es simple.
- Se conectan dispositivos como sensores fotoeléctricos con diagnóstico y controladores básicos.
- Su función principal es la de compartir dispositivos de campo y comandos entre varios equipos de control.
- Los DeviceBus más conocidos incluyen Device-Net, Profibus DP y CAN.

FieldBus (Bus de Campo):

- La Información es transmitida en palabras o tablas.
- Aplicaciones dirigidas a Procesos: La mayoría de las Variables son analógicas, aunque soportan las digitales.
- El control típico es Distribuido.

- El tiempo de respuesta es menor a 100 ms.
- El diagnóstico es sofisticado.
- Se conectan dispositivos más complejos como válvulas inteligentes con PID y diagnóstico, controladores básicos y computadores.
- Su función principal es la de repartir la aplicación.
- Los Field Bus más conocidos incluyen FF, Profibus PA y LonWorks.

ControlBus (Bus de Control):

- Es similar al Field Bus, pero en general tiene mayor capacidad para manejar paquetes y realizar control.
- Los Control Bus más conocidos incluyen Modbus+, Profibus FMS, Data Highway + (DH+) y ControlNet. [15]

1.3. MODELO DE REFERENCIA OSI

En esta sección se explicará el Modelo OSI y se describirán algunas de las Redes de Campo.

1.3.1. Modelo de Referencia OSI

Este modelo nace a raíz de la investigación de la ISO a finales de los 70' para enfrentar el problema de incompatibilidad de redes y con el fin de encontrar un conjunto de reglas aplicables de forma general a todas las redes. Con base en esta investigación, la ISO desarrolla un modelo de referencia que sirve como guía a los fabricantes para desarrollar redes compatibles con otras. A continuación se explica el modelo de referencia de capas OSI. [16]

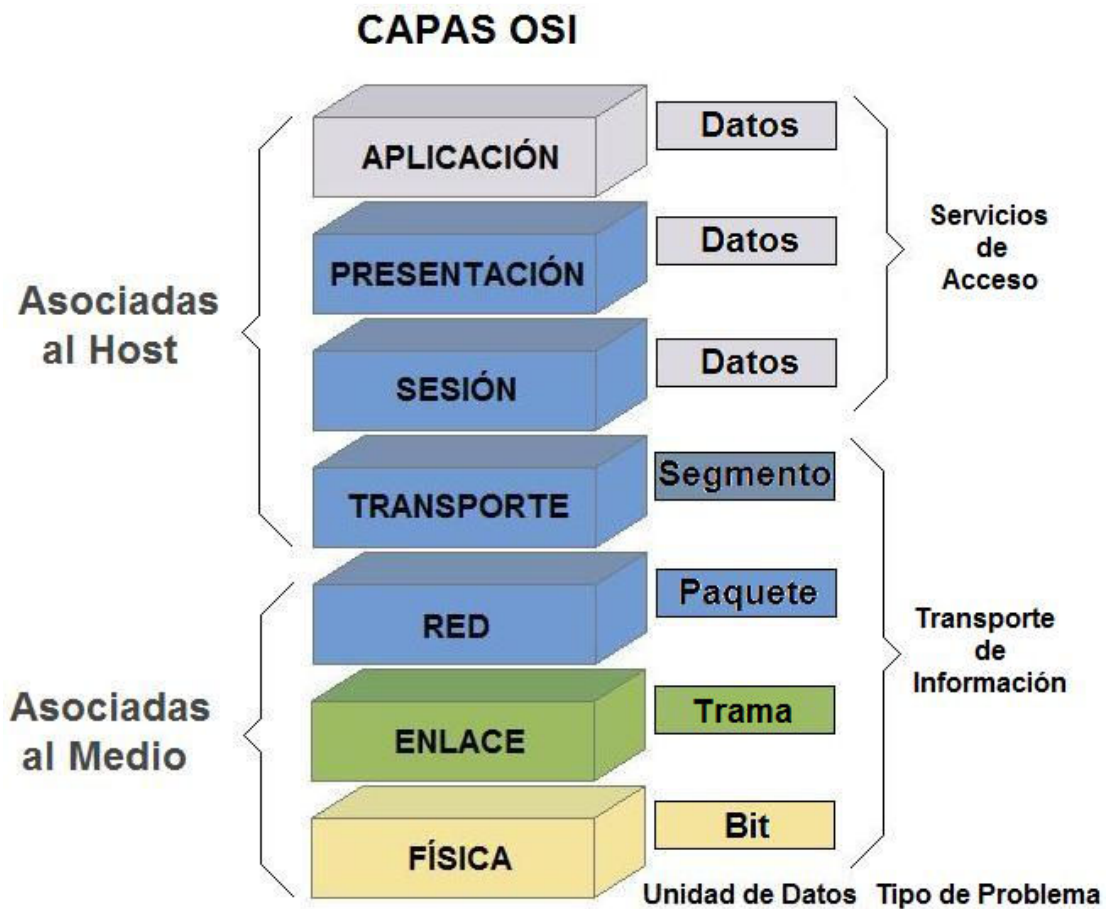


Figura 10. Capas del Modelo de Referencia OSI.

1.3.1.1. Capa Física (Capa 1)

Es la capa inferior del modelo de Referencia OSI. En esta se definen todas las especificaciones mecánicas, eléctricas, funcionales y los procedimientos necesarios para crear, mantener y descartar conexiones físicas entre dispositivos. En otras palabras, define la relación entre un dispositivo y un medio físico de comunicación (indica a los dispositivos como transmitir y recibir del medio). Las funciones más importantes realizadas por la Capa Física incluyen:

- Definir el medio de comunicación (Fibra óptica, Par trenzado, ondas de radio, microondas, entre otros).
- Establecer y terminar una conexión a un medio de comunicaciones.

- Transformar las tramas de datos que provienen de la capa Enlace en una señal adecuada al medio físico.
- Transmitir y recibir los bits.
- Participar en el proceso donde los recursos de comunicación son compartidos entre varios dispositivos (o usuarios). Por ejemplo: En la resolución de Conflictos y Control de Flujo.
- Modular y/o convertir la información a transmitir o recibir (de los dispositivos o usuarios) por el canal de comunicaciones.
- Garantizar la conexión (aunque no la fiabilidad de esta).

Topología de los medios de comunicación:

El tipo de conexión que se realiza en la capa física puede influir en el diseño de la capa Enlace. Una clasificación que se puede realizar es según la cantidad de dispositivos que comparten el mismo medio de comunicación, según esto existen dos opciones: *Conexiones punto a punto*, donde el medio se comparte entre dos dispositivos (RS232, Infrarrojo, otros), y *Conexiones multipunto*, donde más de dos dispositivos se comunican por el mismo medio (Wi-Fi, RS485, Ethernet, otros).

La clasificación anterior resulta poco específica a la hora de detallar los distintos tipos de conexiones que se pueden realizar con las redes de comunicaciones, además, no sólo existen algunos tipos de conexiones multipunto y punto a punto, sino también combinaciones de ambas.

Al momento de seleccionar una Red de Comunicaciones, es importante conocer la topología o disposición física de conexión entre los nodos (cada uno de los dispositivos que se comunican por la red), debido a que permitirá descartar y/o seleccionar las redes que mejor se adapten. Las topologías de redes más comunes son mostradas en la Figura 11.

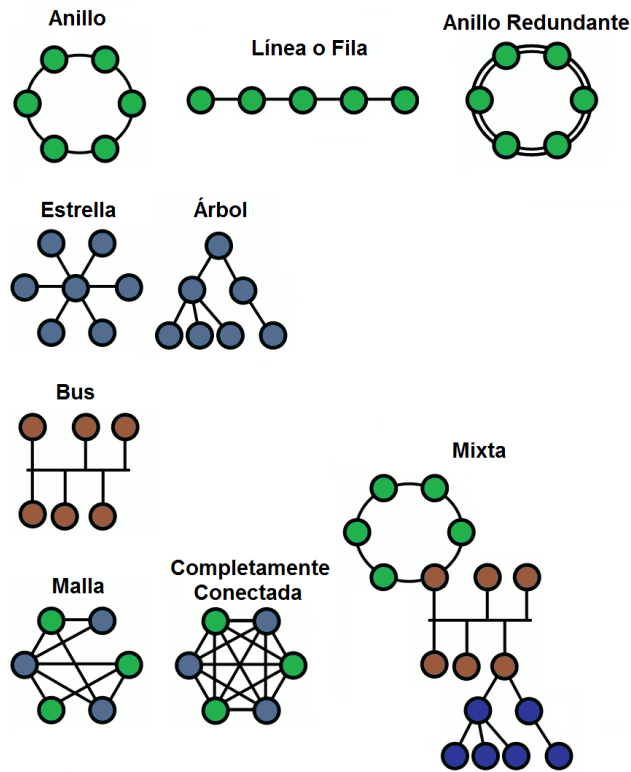


Figura 11. Topología de Redes más Comunes.

De todas estas, muchos autores concuerdan en que sólo existen tres (3) tipos de topologías fundamentales y el resto son derivadas o combinaciones de estas.

La primera de estas, la topología tipo Anillo, se trata de un tipo de conexión en donde cada uno de los nodos se conecta con otros dos y el último se conecta con el primero para formar un anillo. En este tipo de conexiones, la información viaja en un solo sentido, de forma circular y pasa de nodo a nodo hasta llegar al destino. El problema más grave que presenta este tipo de topología es que si algún nodo falla o se rompe una de las conexiones, es muy probable que no se puedan comunicar el resto de los nodos que forman parte del anillo. Existen variantes tales como la topología de Anillo Redundante, en donde se emplea una conexión redundante e independiente y que

generalmente se utiliza cuando falla el anillo principal (es de destacar que la información fluye en sentido opuesto al del anillo principal); o la tipo Línea, que no es más que una topología tipo anillo en donde el primer y el último nodo no se conectan entre sí (o sea, se mantiene abierta la conexión).

La segunda de las topologías fundamentales se trata de la tipo Estrella, en esta todos los nodos se conectan a uno central para comunicarse con el resto de los nodos. En este tipo de topología si un nodo falla o una conexión se cae, el resto de los nodos siempre podrán comunicarse por el central; pero si este último presenta problemas, ningún nodo se podrá comunicar. Una de las variantes de la topología tipo estrella es la tipo Árbol, esta se puede ver como varias topologías estrellas conectadas entre sí y que tiene un nodo central superior por cada estrella conectada.

La tercera y última de las topologías fundamentales es la tipo Bus, en esta todos los nodos se conectan entre sí a través de un mismo medio. Quizás la ventaja más importante que presenta respecto a las otras topologías, es que si un nodo falla, el resto se podrá comunicar sin ningún problema a través del medio, ahora, si la conexión se cae, dependiendo del punto es posible que algunos dispositivos no se puedan comunicar. A pesar de esto, resulta ser la topología más segura a nivel de campo, ya que la conexión entre los dispositivos sólo dependen de medio de comunicación y no de los otros dispositivos como en el caso de la anillo y estrella.

También es posible encontrar topologías híbridas o mixtas de las fundamentales. En estas se buscan combinar diferentes fortalezas y debilidades que presentan las topologías fundamentales y sus variantes, para lograr una que resulte mejor (en cuanto a redundancia, centralización u otros) en fines específicos. Ejemplos de estos tipos de topologías incluyen la tipo malla y la completamente conectada por nombrar algunas.

Dispositivos adicionales que actúan a Nivel Físico:

Existen fundamentalmente dos (2) tipos de dispositivos que forman parte de las redes y actúan a este nivel. *Los repetidores*, que funcionan amplificando y/o reconstruyendo la señal para retransmitirla y lograr que esta llegue a su nodo de destino; y los *Concentradores* o *Hubs*, que funcionan repitiendo la señal de entrada a todos los puertos conectados a este (menos el de la señal de entrada) y son base fundamental para realizar conexiones o convertir la topología en tipo estrella. [16] [17]

1.3.1.2. Capa de Enlace (Capa 2)

También llamada Capa de Vinculación de datos, proporciona organización y controla la transmisión de la data. Además detecta y corrige errores (en la mayoría de los casos) que pudieran ocurrir en la capa física.

Originalmente esta capa fue concebida para realizar conexiones punto-punto y punto-multipunto, pero con el desarrollo de las redes de área local por parte de la IEEE (Proyecto 802) surgió la necesidad de crear subcapas para adaptar el modelo OSI a las LAN. Las subcapas creadas incluyen la LLC (Logical Link Control o Control de Enlace Lógico) y la MAC (Medium Access Control o Control de Acceso al medio), la primera se encarga de emitir y recibir las tramas, establecer y cerrar conexiones lógicas, detectar errores de secuencias en tramas y controlar el flujo; mientras que la segunda reglamenta el acceso al soporte de comunicación (Acceso por consulta (Token), por tiempo (TDMA) o aleatorio (CSMA)).

Dispositivos adicionales que actúan a Nivel de Enlace:

Existen fundamentalmente dos (2) tipos de dispositivos que forman parte de las redes y actúan a este nivel, estos son los *Puentes* o *Bridges* y los *Conmutadores* o *Switches*. Ambos organizan el paso de datos de una red hacia otra con base en la dirección física de destino de cada paquete, la diferencia entre ambos dispositivos radica en que el primero es utilizado para conectar

dos (2) segmentos de red (o dividir una red en dos segmentos), mientras que el segundo se emplea para conectar más de dos redes.

La diferencia entre estos dispositivos que actúan a nivel de Enlace y un concentrador que actúa a nivel físico, es que este último pasa todas las tramas con cualquier destino a todos los nodos conectados, mientras que los primeros únicamente pasan las tramas correspondientes a cada segmento, evitando así tráfico innecesario en la red.

1.3.1.3. Capa de Red (Capa 3)

Esta capa proporciona una transferencia de paquetes (de datos) desde un origen a un destino vía una o más redes, manteniendo la calidad de servicio requerida por la capa de transporte. Esta capa realiza funciones de enrutamiento, reporta errores de entrega y, dependiendo del caso, efectúa fragmentación y reensamblado.

Dispositivos adicionales que actúan a Nivel de Red:

Los Enrutadores, encaminadores o Routers son los dispositivos más conocidos que actúan a este nivel. Tradicionalmente se solía trabajar con Routers para redes fijas (Ethernet, ADSL y otras), pero con el auge de las tecnologías inalámbricas en los últimos tiempos, han aparecido los que realizan la interfaz entre redes fijas y móviles (Wi-Fi, GPRS, Edge, otros). Las VLAN o Redes de Área Local Virtuales son conmutadores que actúan a este nivel, por tanto, permiten crear múltiples redes en un mismo dispositivo y encaminar los paquetes entre estas, realizando por tanto funciones de enrutamiento o Routing.

1.3.1.4. Capa de Transporte (Capa 4)

Esta capa realiza transferencias de datos de forma transparente entre usuarios finales, proporcionando un servicio confiable de transferencia de datos a las capas

superiores. Dentro de las capas asociadas al transporte de información, está ubicada en el tope de estas. Entre sus funciones y características principales, destacan:

- Aceptar los datos de las capas superiores, dividirlos en pequeños trozos si es necesario, y enviarlos a la capa de Red.
- Controlar la confiabilidad de un enlace dado a realizando control de flujo, segmentación/desegmentación y control de errores.
- Aislar las capas superiores de las distintas implementaciones de redes en las capas inferiores.
- Supervisar los segmentos y retransmitir los que fallan en el caso de protocolos orientados a estado y conexión.
- Proveer servicios de conexión para la capa de sesión.

1.3.1.5. Capa de Sesión (Capa 5)

Establece, gestiona y finaliza las conexiones entre procesos y/o aplicaciones finales (locales y remotos). Esta capa proporciona los mecanismos para administrar el diálogo (sesiones) entre los usuarios, sin importar que la operación sea Simplex, Half-Duplex o Full-Duplex, además establece los procedimientos de Iniciación, Puntos de Verificación, Finalización y Reinicio. Entre los servicios más importantes realizados por esta capa, destacan: Controlar la sesión, Controlar la concurrencia y Mantener puntos de Verificación para retransmitir en caso de fallas.

1.3.1.6. Capa de Presentación (Capa 6)

Su función principal es la representación de los distintos tipos de información procedentes de la capa de aplicación. Se encarga de manejar las estructuras de datos abstractas y realizar las conversiones necesarias para la correcta interpretación de las mismas. En esta se tratan aspectos tales como la semántica y la sintaxis de los datos transmitidos. Además, permite cifrar y comprimir los datos. Esta es la primera capa que trabaja en sí el contenido de la comunicación más que el establecimiento de esta.

1.3.1.7. Capa de Aplicación (Capa 7)

Ofrece servicios a aplicaciones y procesos finales de usuario. La aplicación más común de esta capa es proveer la conversión semántica entre procesos de aplicaciones asociadas. Es de destacar que el usuario no solicita el intercambio de datos directamente a través de este nivel, sino que suele interactuar con programas o aplicaciones que a su vez invocan, de manera transparente para el usuario, los servicios de transferencia de data. Esto es debido a que el modelo OSI no incluye Interfaces Máquina-Usuario o HMI. [16] [17] [18]

La Tabla 2 resume las aplicaciones o características más resaltantes de cada una de las capas, así como algunos ejemplos característicos de estas:

Capa	Funciones/Característica	Ejemplos
Aplicación	Servicios de Redes a aplicaciones	Modbus, HTTP, FTP
Presentación	Representación de los datos y Encriptamiento	ASCII, MIDI, AFP
Sesión	Administrar el diálogo entre Hosts	NetBIOS, SIP
Transporte	Conexiones entre usuarios finales y Fiabilidad	TCP, UDP, NBF
Red	Enrutamiento y Direccionamiento Lógico	IP, IPX
Enlace	Direccionamiento Físico (MAC y LLC)	Ethernet, WLAN
Física	Señal, Medio y Transmisión Binaria	RS232, 10Base-T, SONET

Tabla 2. Resumen de las Capas OSI.

1.3.1.8. Funcionamiento entre Capas

El proceso de encapsulamiento que ocurre entre las capas es mostrado en la Figura 12. Inicialmente la Capa de Aplicación recibe la data procedente de la aplicación externa, añadiéndole un Encabezado o Header a la data para luego ser transferida a la capa de presentación. Este Encabezado contiene 2 tipos de mensajes:

- La información de control para la Capa de Aplicación del usuario final o proceso de destino.
- La información de control para la Capa de Presentación o la capa inmediata inferior, en caso de que la red no haya sido diseñada con esta.

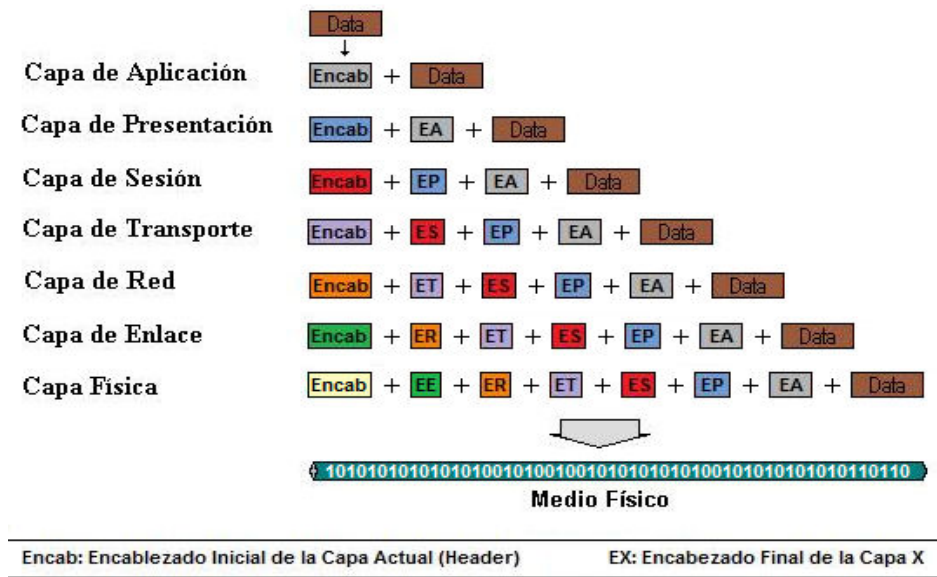


Figura 12. Topología de Redes más Comunes.

Al recibir el mensaje procedente de la Capa de Aplicación, la Capa de Presentación (o la inmediata inferior) analiza la información de control que le ha sido enviada de la capa superior, realiza las acciones correspondientes y de igual forma como se realizó en la capa de Aplicación, le añade un encabezado con información para la capa inferior y la capa similar del usuario destino. El proceso anterior se repite para cada una de las capas, hasta llegar a la Capa de Física, en donde se realiza la adecuación al medio físico para ser transmitida por el canal de comunicación.

La capa física del usuario destino recibe la información y realiza lo siguiente:

- Analiza el Encabezado correspondiente a la capa análoga del origen.
- Realiza las conversiones y acciones de control requeridas.

- Elimina el Header de la Capa análoga de origen
- Dependiendo del caso, añade un encabezado de control para la capa inmediatamente superior.
- Transfiere el mensaje al nivel inmediato superior.

Esto se repite para cada una de las capas superiores, hasta llegar a la Capa de Aplicación, en donde se obtiene la data que será empleada por el usuario destino. [16]

1.3.2. Descripción Básica de algunos Protocolos y/o Redes de Campo

1.3.2.1. Modbus

Es un protocolo de comunicaciones posicionado en la Capa de Aplicación (Capa 7) del Modelo OSI. Este permite la comunicación Cliente/Servidor entre dispositivos conectados por diferentes tipos de buses o redes. Modbus es el Estándar Industrial para Redes Seriales desde 1979 cuando fue publicado por Modicon para usarse con su familia de PLC. Su popularidad en el ambiente industrial actual es muy amplia debido a cuatro razones principales:

- El protocolo es abierto y gratuito.
- La Red Industrial es relativamente fácil de instalar.
- El protocolo es fácil de implementar.
- Existen gran cantidad de dispositivos industriales y fabricantes que emplean Modbus como estándar para comunicaciones.

Modbus es un protocolo de Solicitud/Respuesta o Request/Reply y ofrece servicios especificados por códigos de funciones preestablecidas. Estos códigos forman parte de las Unidades de Datos de Protocolo (PDU) y son prácticamente la base del protocolo. Además de la PDU, se introducen campos que dependen del tipo de red por el cual se transmite Modbus, para formar lo que se conoce como Unidad de Aplicaciones de Datos (ADU). Una solicitud Modbus, independientemente del tipo de red, contiene el formato que se muestra en la Figura 13.

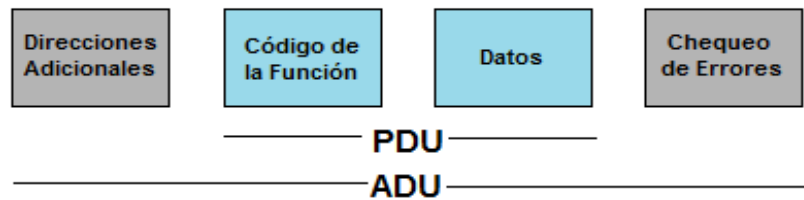


Figura 13. Descripción de una Solicitud Modbus.

Los códigos de las Funciones válidas Modbus son de 1 byte y van desde la 001 hasta la 255, aunque el rango desde 128 hasta 255 está reservado para las respuestas de excepción. La mayoría de los códigos de funciones se encuentran definidos y totalmente especificados por la Organización Modbus. Algunos códigos son abiertos a los usuarios (User-defined) para que estos puedan especificar y definir cualquier función no incluida que se adapte a sus necesidades, estos códigos incluyen el rango 65-72 y 100-110. El resto de los códigos son Públicos, definidos en su totalidad y realizan gran cantidad de funciones comunes de comunicación, tales como la lectura y escritura de Coils, lectura y escritura de registros y otros.

Modbus presenta algunas limitaciones por ser un protocolo diseñado para realizar la comunicación entre los primeros PLC (en los años 70). A continuación se nombran algunas de las más importantes:

- Los tipos de datos soportados por el protocolo no son tan amplios, por lo que algunos tipos de datos más recientes no son soportados.
- Los objetos Binarios de gran longitud no son soportados.
- Se pueden conectar hasta un máximo de 254 remotas a una maestra.
- Como el protocolo es Maestro/Esclavo, los esclavos no pueden generar reportes por excepción, esto obliga a la interrogación constante de los dispositivos remotos por parte del maestro, lo que se traduce en un consumo amplio de ancho de banda.

Modbus no sólo se utiliza como protocolo para comunicación serial. Algunas variaciones de este han sido realizadas debido su gran popularidad en la industria y al

avance de las redes de comunicaciones industriales, por lo que en la actualidad es muy común encontrar dispositivos que emplean Modbus sobre TCP/IP para comunicarse. También se ha implementado Modbus sobre SMS y GPRS, aunque no son tan comunes como el empaquetado sobre TCP/IP. [13] [19] [20]

1.3.2.2. OLE for Process Control (OPC)

Es el nombre original dado a una especificación estándar abierta para la automatización industrial desarrollada en 1996 producto de la colaboración de múltiples proveedores de equipos para automatización industrial, conjuntamente en cooperación con Microsoft. El estándar OPC original fue llamado “OPC Specification”, pero con el surgimiento de los nuevos estándares, su nombre fue cambiado a lo que hoy en día se conoce como OPC Data Access o OPC-DA. Este OPC original fue basado en las tecnologías OLE, COM y DCOM desarrolladas por Microsoft para su sistema operativo Windows. La especificación define un juego de objetos, interfaces y métodos estándares para usar en el control de procesos y facilitar la interoperabilidad en aplicaciones de automatización industrial. [21] [22]

Inmediatamente después del desarrollo del OPC-DA, se crea la Fundación OPC para mantener el estándar, desarrollar mejoras e impulsarlo a nivel industrial. Hoy en día, la fundación OPC cuenta con 10 especificaciones (7 desarrolladas en su totalidad y 3 emergentes), las cuales se nombran a continuación:

- **OPC Data Access:** Empleada para transferir información en tiempo real desde un PLC, los RTU y otros sistemas de control a HMI y otros clientes de monitoreo. Actualmente se cuenta con la 3ra generación de OPC-DA.
- **OPC Alarms & Events:** Proporciona notificaciones de alarmas y eventos por demanda (on demand).
- **OPC Batch:** Proporciona una interfaz para el intercambio de capacidades de equipamiento y las condiciones de operación actuales.
- **OPC Data eXchange:** Esta especificación permite la comunicación servidor a servidor a través de redes de campo basadas en Ethernet,

proporcionando interoperabilidad entre distintos tipos de redes de campo. Además, añade la posibilidad de configuración remota, diagnósticos y monitoreo/administración de servicios.

- **OPC Historical Data Access:** Proporciona acceso a información almacenada.
- **OPC Security:** Este estándar especifica cómo controlar el acceso de los clientes a los servidores, con la función de mantenerlos protegidos contra la modificación no autorizada de parámetros de procesos.
- **OPC XML-DA:** Proporciona flexibilidad, reglas consistentes y formatos para presentar la información de planta empleando XML.
- **OPC Complex Data:** Una especificación que le permite a servidores presentar y describir tipos de datos más complejos, tales como, estructuras binarias y documentos XML.
- **OPC Commands:** Un grupo creado recientemente para desarrollar un nuevo juego de interfaces que le permitirán, a los clientes y servidores OPC, identificar, enviar y monitorear comandos de control que se ejecutan en un dispositivo.
- **OPC Unified Architecture:** Un nuevo juego de especificaciones no basadas en el COM de Microsoft, que proporcionará estándares para la comunicación inter-plataforma. Este elimina la necesidad de Windows.

1.3.2.3. DirectNET

Probablemente el más conocido y emblemático de la empresa AutomationDirect. Este protocolo es abierto y fue diseñado para ser usado en comunicaciones seriales RS232-C y RS422 con los productos de la familia DirectLOGIC. Este se emplea para comunicaciones entre PLC o entre computadores y uno o varios PLC. Es un protocolo maestro-esclavo, por lo que un PLC debe ser configurado como maestro en una red de varios PLC. La velocidad máxima de comunicación de este protocolo es de 38,4 kbps.

La ventaja principal de este protocolo radica en que un PLC puede ser la estación maestra de un conjunto de varios PLC y computadores. Además, la gran mayoría de los productos de DirectLOGIC cuentan con este protocolo de comunicaciones por ser un protocolo relativamente viejo. Entre las desventajas, se encuentra que no puede manipular data a nivel de bits y que no es un protocolo industrial estándar.

El protocolo realiza lecturas y escrituras desde una estación maestra a un esclavo, empleando los siguientes comandos: Solicitud (Enquiry - ENQ), que inicia una solicitud con una estación esclava; Encabezado (Header - HDR), que define si la operación es de lectura o escritura, la dirección del esclavo y el tipo y cantidad de datos a transmitir; Datos (Data - DATA), que contiene la información que se transmite; Reconocimiento (Acknowledge - ACK), que verifica si la comunicación funciona correctamente; y el Final de Transmisión (End of Transmission - EOT), que indica que la comunicación ha finalizado. El proceso de lectura y escritura de DirectNET es mostrado en la Figura 14. [23] [24]

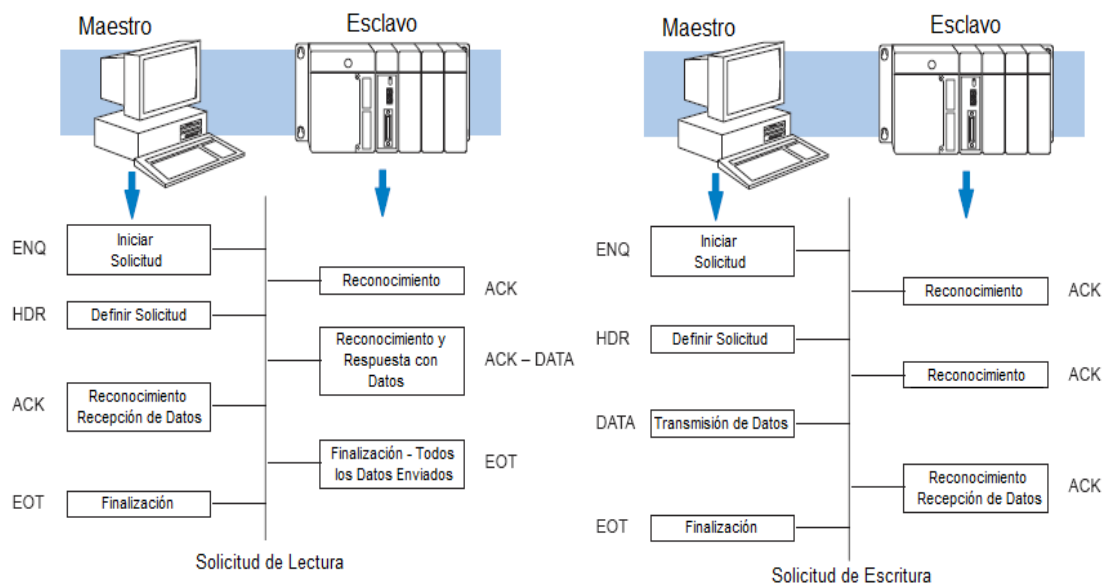


Figura 14. Descripción de Solicitudes de lectura y escritura en DirectNET. [23]

1.3.2.4. K-Sequence

Es un protocolo de comunicaciones propietario de AutomationDirect diseñado para ser usado en redes seriales RS232-C y RS422. La ventaja respecto al DirectNET, es que este puede realizar operaciones de escritura sobre bits individuales, tales como puntos I/O y relés de control. Este protocolo es empleado comúnmente para comunicarse entre un computador con el software DirectSoft y un PLC DirectLOGIC. Su velocidad máxima es de 38,4 kbps al igual que el DirectNET. También se tiene conocimiento que es empleado para realizar la comunicación interna entre los módulos de un PLC DirectLOGIC. A pesar de ser un protocolo propietario, es posible obtener la información de este (luego de firmar documentos de confidencialidad y no divulgación) para realizar aplicaciones sin pagar regalías. [25]

1.3.2.5. ECOM

Es un protocolo de comunicaciones basado en una red peer-to-peer. Este se transmite sobre TCP/IP o IPX (en desuso, pero soportado de igual forma para redes más antiguas). Al igual que el K-Sequence, este protocolo es propietario, pero la compañía AutomationDirect proporciona las especificaciones luego de firmar contratos de no divulgación y confidencialidad para aquellos desarrolladores que deseen realizar aplicaciones con este.

Este protocolo es empleado para realizar operaciones de lectura y escritura entre varios PLC o entre computadores y PLC. ECOM es primordialmente un protocolo empleado para la programación de los PLC desde el software de programación DirectSoft, pero en casos que no se cuente con los módulos de comunicación más recientes (que poseen Modbus sobre TCP), este es empleado de igual forma para realizar las comunicaciones entre PLC, HMI y cualquier otro dispositivo compatible. [26]

1.3.2.6. Ducbus

Es un protocolo propietario público del tipo maestro/esclavo de la empresa Italiana DUCATI Energía, comúnmente empleado como protocolo de comunicaciones para redes seriales RS485 en sus dispositivos de medición y análisis de energía. Este permite un direccionamiento máximo de hasta 98 dispositivos sobre una misma red. Una solicitud de datos o de cambio de configuración y una respuesta a una solicitud de datos, están compuestas de la siguiente forma: [2] [27]

<STX> <DISP> <DIR> <BLANK> <DATA> <BLANK> <CHKSUM> <ETX>

<STX>: Inicio de Transmisión.

<DISP>: Código Indicador del dispositivo que inicia o responde la solicitud (Si es Maestro o Esclavo).

<DIR>: Dirección del esclavo.

<BLANK>: Código Blanco o separador.

<DATA>: Código referente a una función previamente definida (en el caso de una solicitud) o el valor del parámetro solicitado (en el caso de una respuesta).

<CHKSUM>: Suma de verificación de la trama.

<ETX>: Fin de Transmisión.

Una respuesta a un cambio de configuración por parte de un esclavo está compuesta únicamente por los comandos <ACK> o <NAK> referentes a los códigos de Reconocimiento (ACK) y No Reconocimiento (NAK) de la solicitud realizada.

1.4. DISPOSITIVOS Y PROGRAMAS PARA AUTOMATIZACIÓN INDUSTRIAL Y/O SCADA

En esta sección se describirán brevemente algunos de los dispositivos más comúnmente empleados para realizar control en un ambiente industrial actual,

algunos sistemas de transmisión y recolección de datos y el paquete de programas Wonderware para SCADA y HMI a nivel industrial.

1.4.1. Dispositivos para Control Industrial

Los sistemas empleados comúnmente para realizar control en la actualidad se pueden resumir básicamente en 3 dispositivos, estos son, los Controladores Lógicos Programables (PLC), los Controladores de Automatización Programable (PAC) y los Computadores y/o Computadores Industriales.

1.4.1.1. Controladores Lógicos Programables (PLC)

En primer lugar, y gracias al auge del control distribuido, se encuentran los PLC. Estos dispositivos son diseñados para controlar procesos secuenciales en tiempo real y para un ambiente industrial. La NEMA los define como un aparato electrónico que opera digitalmente, el cual emplea memoria programable para el almacenamiento interno de instrucciones que se utilizan en la implementación de funciones específicas (que pueden ser lógicas, secuenciales, temporizadoras, contadoras y aritméticas) y en la realización de control a varios tipos de maquinarias y procesos (a través de entradas y salidas analógicas o digitales).

La primera generación de PLC nace en 1968 después de que Dick Morley, un ingeniero estadounidense, presentara su invención en un concurso de General Motors para sustituir los sistemas de relés. Desde esa época ha cambiado significativamente el concepto y las prestaciones de los PLC. En la actualidad la mayoría de estos pueden comunicarse en distintos protocolos y por las redes más comunes empleadas a nivel industrial, además de ser muy robustos y permitir distintas posibilidades de programación (diagrama de escalera, diagrama de bloques y otros). [7] [8]

DirectLogic205

Es una de las series de PLC modulares de la Empresa AutomationDirect. Por ser una serie modular, es posible configurar un PLC de esta serie de diferentes formas. A continuación se mencionan algunos de los módulos actuales existentes en el mercado para la configuración de un PLC DirectLogic205:

Bases: Las bases pueden ser para 3, 4, 6 y 9 módulos (uno de los módulos es empleado para colocar el CPU).

CPU: Existen 4 posibilidades de CPU dependiendo del caso. La primera corresponde al DL230, este modulo contiene 2,4K totales de memoria de palabras (2048 correspondientes para Escalera y 384 para memoria virtual), acepta la programación RLL y contiene un puerto RS232C para la programación y comunicación serial. El DL240 es la segunda posibilidad de configuración, este contiene mayor cantidad de memoria de palabra, y a diferencia del DL230, contiene dos puertos RS232C. También posee más instrucciones de programación y presenta mejores prestaciones que su predecesor.

La tercera y cuarta posibilidad corresponden al DL250-1 y al DL260. Ambos presentan gran cantidad de mejoras en cuanto a la compatibilidad con otros protocolos, cantidad de memoria, operaciones con variables de punto flotante, operaciones aritméticas (DL260) y contienen un puerto de comunicaciones RS232C y otro RS232C, RS422 o RS485 (DL260). En resumen, a medida que aumenta el dígito de las decenas, presentan mejores prestaciones.

Módulos de Entrada/Salida y especiales: Se pueden instalar módulos de entradas digitales, entradas analógicas, salidas digitales a relé, módulos de alimentación, módulos de comunicaciones (para serial e Ethernet en distintos protocolos), módulos para termopares y otros especiales. No todos los módulos son

compatibles con cada uno de los CPU y las tensiones de alimentación (para los módulos de entrada y/o salida) pueden variar dependiendo de cual este instalado.

La mayoría de los módulos de entrada poseen bornes de alimentación externos (aunque las bases poseen una fuente interna) con la finalidad de darle la posibilidad al usuario de reducir el ruido originado por la fuente de alimentación conmutada de la base o de alimentar con una fuente que entregue mayor potencia en caso de tener variables que así lo requieran. Desde el punto de vista de la programación, el PLC contienen gran cantidad de funciones especiales, tales como, distintos Timers y Contadores, PID Loop, EDRUM, bloques de operaciones aritméticas y otros. Algunas configuraciones de PLCs DL205 son mostradas en la Figura 15. [28]



Figura 15. Algunas configuraciones con los 4 tipos de bases del PLC DL205.

1.4.1.2. Controladores de Automatización Programable (PAC)

Los PAC o Controladores de Automatización Programable, se muestran actualmente como tecnología emergente y se proyectan como la siguiente generación de PLC. Los PAC surgen por la necesidad de realizar un control avanzado que no pueden lograr los PLC tradicionales, pero manteniendo la dureza y confiabilidad que no poseen los computadores (ver Figura 16). El nombre PAC fue apodado a estos dispositivos en el año 2002 por el grupo consultor estadounidense ARC. [29]

Estos dispositivos son similares a un PC en cuanto a su constitución interna, al procesamiento y al almacenamiento de datos, aunque el sistema de tratamiento de los puertos y la adquisición de datos funcionan de forma similar a la de un PLC (mantienen la filosofía del escaneo y escritura comunes de los PLC).

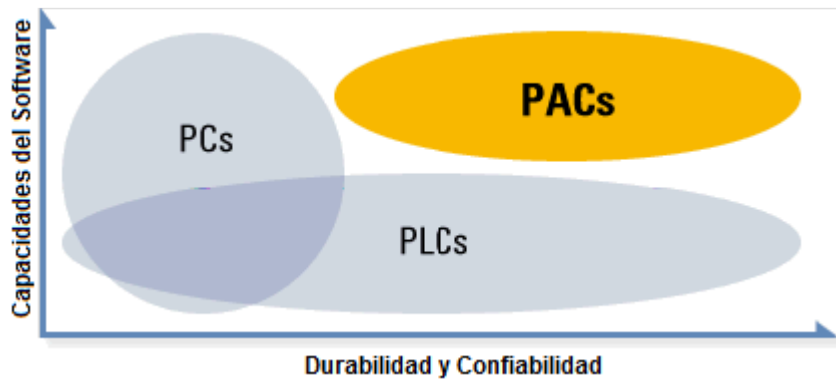


Figura 16. Comparación de PAC con PC y PLC. [31]

En un estudio realizado por la National Instruments, aproximadamente el 80% de los procesos industriales se pueden resolver con herramientas tradicionales de control. Este 80% es comúnmente resuelto con pequeños PLC, también llamados microPLC, en un conjunto máximo de 20 instrucciones tipo escalera. Para el 20% restante, se necesitan herramientas de control más avanzadas. Estos procesos generalmente se resuelven con PLC más elaborados y de mayor capacidad, pero la programación en escalera o en algún equivalente se hace muy complicada y a veces se requiere un procesamiento o de operaciones que un PLC no puede realizar (y en donde la solución natural sería emplear un computador). Los PAC apuntan a resolver estos casos, manteniendo la confiabilidad, empaque y durabilidad de un PLC pero con la flexibilidad y funcionalidad de un software para PC.

Muchos de los fabricantes actuales de PLC han empezado a desarrollar PAC y a introducirlos al mercado. Algunos de estos han tomado como base uno de sus PLC y han mejorado el software para adaptarse a las definiciones de PAC dadas por el grupo consultor ARC. Entre los productores de PAC destacan Opto22, Rockwell Automation, National Instruments y ABB. Algunos ejemplos de Hardware PAC comerciales son mostrados en la Figura 17. [30] [31] [32]



Figura 17. Hardware PAC.

1.4.1.3. Computadores Industriales

Tal cual como su nombre lo indica, es un computador al que se le instala un software para realizar un proceso de control industrial. Aunque su uso está en decadencia con la aparición de los PAC, algunas empresas aún los emplean. La poca confiabilidad y durabilidad son los grandes problemas de estos dispositivos, propiedades básicas que se buscan en una automatización industrial. [31]

1.4.2. Sistemas de Transmisión y Recolección de Datos

Para adquirir y transmitir la data desde el sensor a cualquier otro dispositivo se emplean transmisores, y según el caso, recolectores de datos. Un transmisor es un equipo electrónico que toma las mediciones de los sensores y las transmite en tiempo real a un dispositivo de control o registrador. Con el avance de la tecnología los sistemas de transmisión han avanzado para permitir almacenar temporalmente la data adquirida e incluso realizar control crítico.

La definición clásica de un Recolector de Datos indica que son dispositivos orientados a la adquisición, almacenamiento y el registro de múltiples variables. Estos han evolucionado permitiéndoles transmitir la data adquirida a cualquier otro dispositivo. Hoy en día, la definición de un recolector de datos y de un transmisor se tiende a solapar.

Los RTU o Remote Terminal Unit son dispositivos basados en microcontroladores que se encargan de adquirir la data directamente de los sensores y transmitirla a otros dispositivos para su procesamiento. Si se ve desde este punto de vista, las RTU se pueden clasificar como transmisores de campo, pero las más modernas tienen la capacidad de almacenar datos temporalmente, realizar acciones de control remoto (Solicitadas desde un dispositivo de Control o PC), e incluso ejecutar una lógica de control local, es por eso que cada vez más clasificar a un dispositivo con los términos RTU, PLC y PAC (o DCS) tiende a ser más complicado. Algunos transmisores y recolectores comerciales son mostrados en la Figura 18. [33]



Figura 18. Algunos Transmisores y Recolectores de Datos.

1.4.2.1. *Ducati MACH 30*

Es un panel analizador de energía trifásica (220 V) de la división de energía de la empresa italiana Ducati. Esta RTU es capaz de medir múltiples variables eléctricas utilizadas para el análisis de calidad de energía. Entre estas se encuentran, las tensiones de línea, tensiones de fase, corrientes de línea, factor de potencia, potencia entregada, energía entregada, frecuencia de la red, THDF de tensión, THDF de corriente y otras. Además, cuenta con una salida de comunicación serial 485 (en protocolo propietario Ducbus), 4 salidas programables, 2 entradas de 4-20 mA y almacenamiento temporal de algunas variables. Una imagen de la Ducati Mach 30 es mostrada en la Figura 19. [27]



Figura 19. Ducati Mach30. [27]

1.4.3. Programas Wonderware para SCADA y HMI

Un programa para SCADA no contiene únicamente las herramientas para lograr el monitoreo de variables en tiempo real, la gran mayoría brinda la posibilidad de almacenar datos, generar alarmas y enviar comandos de control a los dispositivos de campo. La programación y definición de las variables comúnmente se realiza por Tags, o sea, variables globales a nivel de programa que pueden ser asignadas a una variable física o definidas como cualquiera de los tipos de datos usuales de la informática (dependiendo del programa) para realizar alguna operación interna.

Wonderware, una empresa de origen estadounidense fundada en 1987 y actualmente una de las unidades de negocios de la empresa británica Invensys, es uno de los grandes a nivel mundial en programas para SCADA y HMI. La empresa ha enfocado su desarrollo a la tecnología ArchestrA, una infraestructura de software diseñada por Invensys para unificar todas las aplicaciones actuales y emergentes de esta empresa, sus asociados y las aplicaciones internas de los clientes, en un modelo de planta sincronizado que permita la adaptación y aliente el mejoramiento sucesivo. Un esquema de las áreas involucradas por ArchestrA es mostrado en la Figura 20.

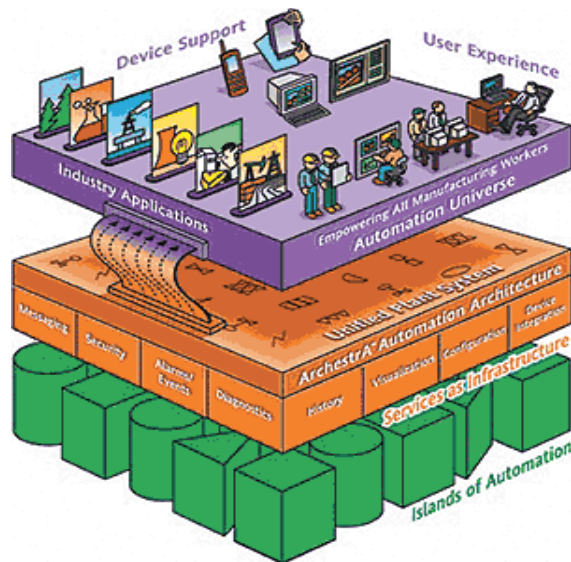


Figura 20. Áreas de una automatización industrial involucradas por ArchestrA. [34]

ArchestrA es definida por Invensys como una arquitectura amplia de automatización de planta e información, diseñada desde sus inicios para extender la vida de los sistemas heredados o sistemas antiguos (legacy). Invensys afirma que el acercamiento completo de ArchestrA a la arquitectura industrial reduce significativamente el costo total de propiedad de la planta, gracias a que las aplicaciones de automatización son de fácil instalación, operación, modificación, mantenimiento, reproducción y Escalabilidad.

Los programas Wonderware explicados brevemente a continuación, toman como base la tecnología ArchestrA para generar un paquete de productos que incrementan la funcionalidad, flexibilidad y conectividad industrial:

- ***Industrial Application Server:*** El corazón de la Tecnología. Una plataforma que abarca todo el sistema y que le da la posibilidad a los ingenieros de planta de realizar ingeniería colaborativa o distribuida. Entre sus funciones se encuentran la adquisición de data en tiempo real, la administración de las alarmas y eventos, la centralización de la seguridad, la manipulación de los datos e información y la realización de despliegues o deployments remotos. El IDE es un programa incluido en este paquete, en el cual se realizan las configuraciones, despliegues y programación de objetos.
- ***InTouch:*** El software para realizar las HMI y control básico según el caso. Permite la visualización de los procesos de una planta.
- ***ActiveFactory:*** Un complemento para el análisis de datos que permite la aceleración y mejoramiento de toma de decisiones en todos los niveles de planta. Este paquete ofrece una serie de clientes que adquieren la data del IndustrialSQL Server para el análisis y toma de decisiones.
- ***IndustrialSQL Server:*** El histórico de planta.

- **Device Integration:** Un paquete de productos que ofrece una librería de cientos de objetos de comunicaciones (I/O Servers y DA Servers).
- **Information Server:** El software para portales industriales que permite la administración y visualización de la planta vía Internet o Intranet.
- Otros productos como InTrack, InBatch, InControl, DT Analyst, SCADAAlarm y otros. [35]

Algunos ejemplos de los productos Wonderware se muestran en la Figura 21.

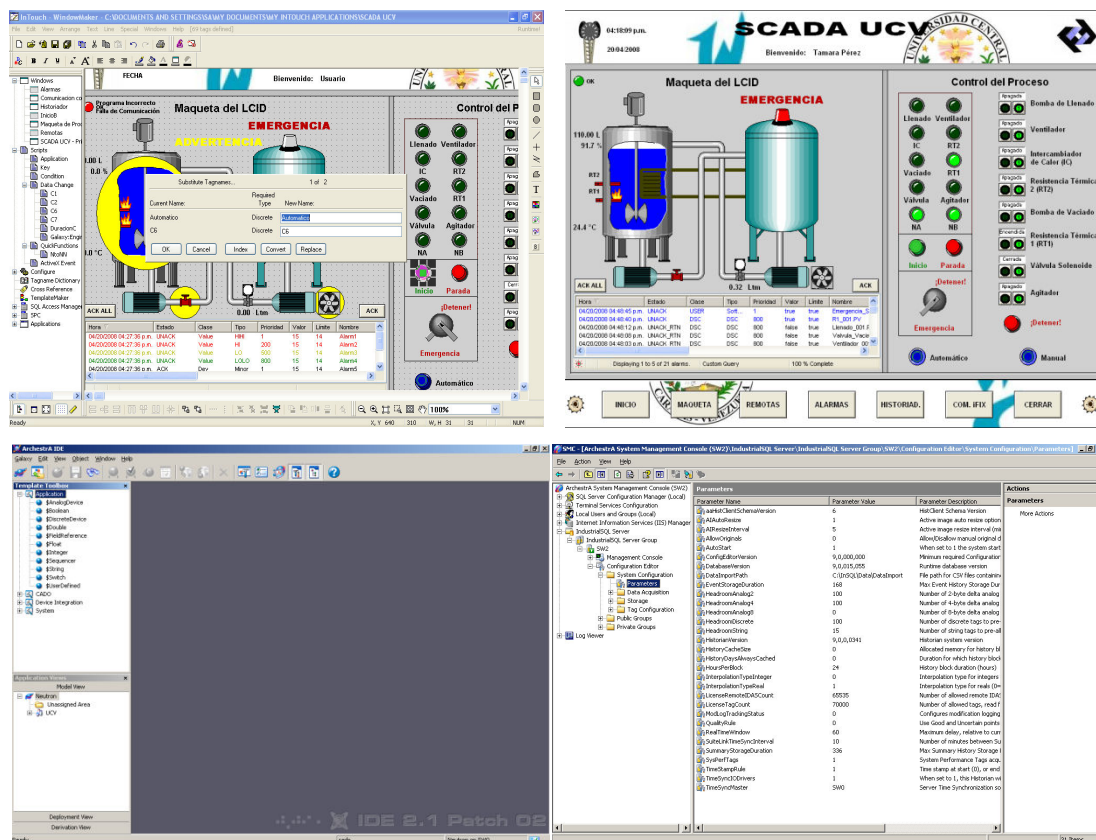


Figura 21. Imágenes de los paquetes InTouch (WindowMaker y WindowViewer), Industrial Application Server (IDE) e IndustrialSQL Server (SMC Console).

CAPITULO II

DESCRIPCIÓN DEL SCADA

2.1. IMPLEMENTACIÓN ANTERIOR DEL SCADA

El Laboratorio de Redes de Distribución (LRD) de la EIE-UCV contiene un sistema que adquiere datos de unas remotas (Ducati y Lifasa) de un banco de prueba ubicado en el Laboratorio de Máquinas Eléctricas, y los muestra en un HMI desarrollado bajo la plataforma iFix de Intellution.

Los datos de las remotas son solicitados por un servidor Modbus integrado en el paquete iFix, éste interroga un Gateway, desarrollado en una tesis de grado anterior [2], para convertir los protocolos seriales de las remotas en Modbus sobre TCP. El iFix y el resto de los programas se encuentran instalados y configurados en computadores ubicados en el LRD. Una imagen que describe este sistema se muestra en la Figura 22.

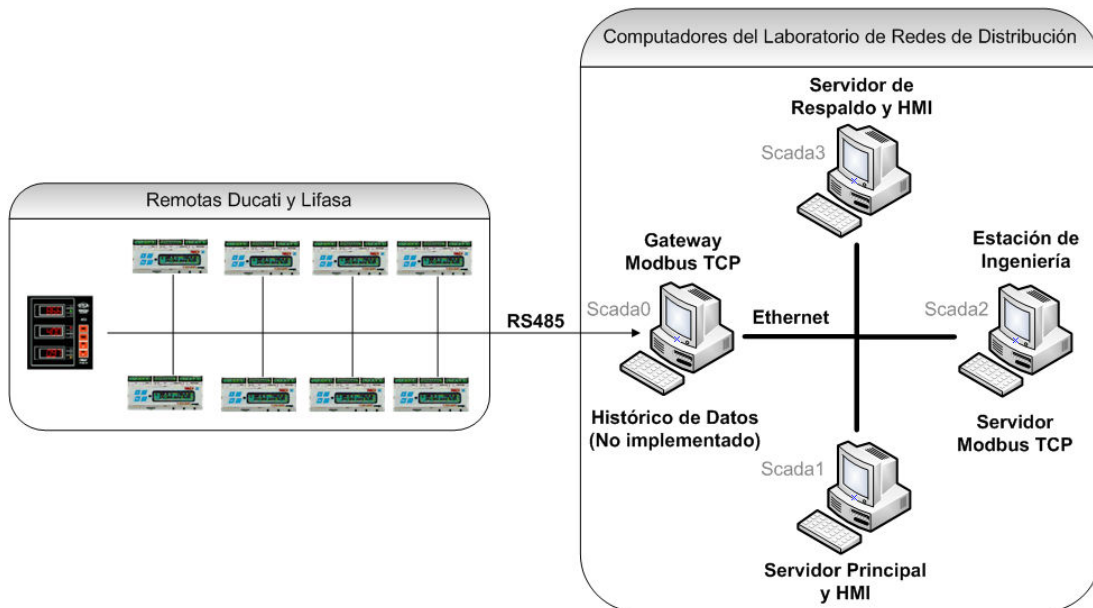


Figura 22. Esquema del SCADA actual.

El esquema anterior, basado en los resultados del trabajo del Ing. Nelson Salcedo [3], contiene dos áreas físicas, la primera correspondiente a la adquisición de los datos por parte de las remotas instaladas en el Laboratorio de Máquinas y la segunda correspondiente a los computadores y el sistema SCADA en sí del Laboratorio de Redes de Distribución.

Las Remotas Ducati y Lifasa miden los parámetros eléctricos del tablero principal de la EIE-UCV. Un total de ocho remotas Ducati y una remota Lifasa son instaladas para realizar las mismas mediciones de prueba, además de ser conectadas en paralelo por medio de una red RS485. Esta red interconecta las dos áreas físicas por medio de un par trenzado blindado, que es convertido de RS485 a RS232 para poder realizar la adquisición con una tarjeta serial estándar de cualquier computador.

Como se observa en la Figura 22, un computador del LRD contiene el Gateway [2] que convierte los protocolos seriales de las remotas Ducati (Ducbus) y Lifasa (Modbus) a Modbus sobre TCP. Todos los computadores se interconectan entre sí por medio de una red local Ethernet. Los tres computadores restantes ubicados en el Laboratorio, contienen el Servidor Principal del iFix, el servidor de respaldo y la estación de ingeniería; además, es posible visualizar la interfaz Hombre-Máquina en todas estas máquinas. El Historiador de datos se encuentra en el mismo computador donde fue instalado el Gateway (aunque no se encuentra operativo).

El HMI muestra algunos parámetros eléctricos de la remota Lifasa y de sólo tres remotas Ducati en tiempo real. También contiene una tabla para el reconocimiento de alarmas, una pantalla para observar el estado de las computadoras que conforman el SCADA y una plantilla para configurar la seguridad. Algunas vistas de las plantillas del HMI son mostradas en la Figura 23.

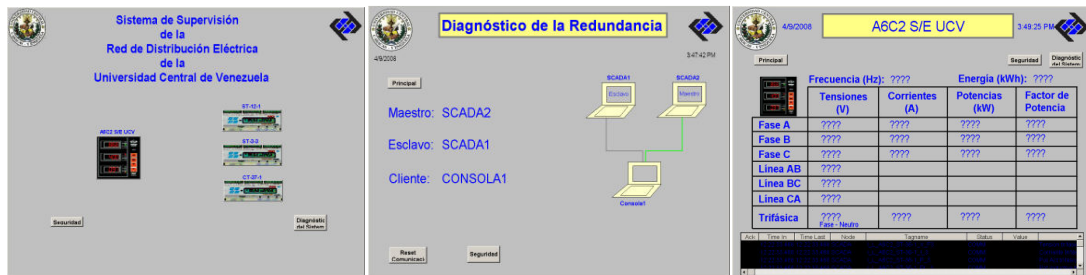


Figura 23. Algunas Plantillas del HMI.

2.2. LIMITACIONES ENCONTRADAS EN EL SCADA ACTUAL Y DISPOSITIVOS A AÑADIR

La actualización del SCADA se formula bajo la idea principal de implementar la tecnología ArchestrA haciendo uso de las licencias donadas por la Empresa Wonderware of Venezuela, inc a la Coordinación de Postgrado de la EIE-UCV. Este es planteado de forma tal que cumpla las mismas funciones que el actual (realizado con iFix), pero basado en una tecnología orientada a objetos que adicionalmente incluya un histórico de datos funcional, al menos una conexión con un PLC y que se comunique con el SCADA existente. De esta forma se propone el esquema inicial para la actualización del SCADA del LRD en la Figura 24.

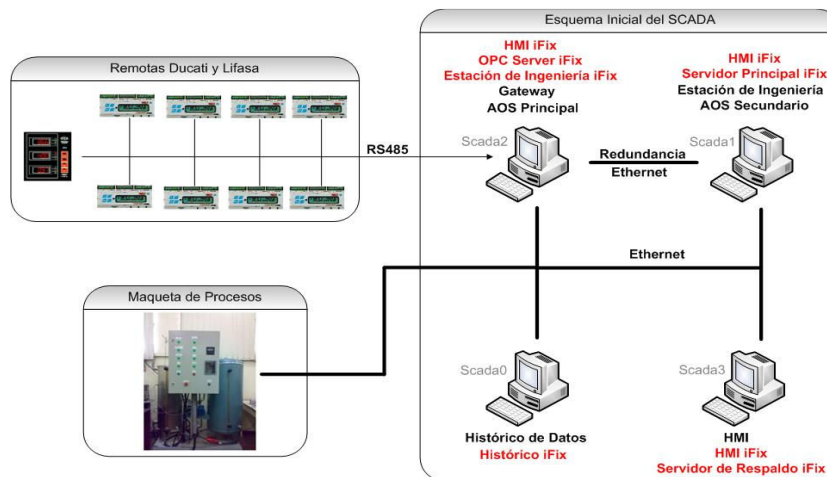


Figura 24. Esquema inicial propuesto para la actualización del SCADA.

En el esquema anterior se detallan 3 etapas: la primera correspondiente a las remotas Ducati y Lifasa para la adquisición de parámetros eléctricos, localizadas en un banco de prueba del laboratorio de máquinas; la segunda etapa referente a la maqueta de procesos automatizada con un PLC DirectLogic, ubicada en el laboratorio de Control, Instrumentación y Digitales (LCID); y por último, el esquema básico de los computadores con el paquete Wonderware para la actualización del SCADA y su función actual con el SCADA bajo iFix.

En base al esquema inicial y al Flujo de Trabajo sugerido por Wonderware mostrado en el Anexo 1, se propone una metodología para la implementación del SCADA. En esta se define como primer paso, la identificación del estado de los dispositivos que forman parte del SCADA actual y del resto de los equipos incluidos en la actualización, además de la definición de la topología del sistema cumpliendo con la estructura básica de un Sistema de Control Distribuido. A continuación se explica el estado inicial de cada una de las etapas definidas en el esquema anterior y las decisiones y acciones tomadas por esta causa:

2.2.1. Adquisición de Parámetros Eléctricos

Al analizar el estado de las RTU, se nota la ausencia de la remota Lifasa en el banco de pruebas y de la existencia de un cable de comunicaciones no canalizado ni suficientemente largo como para conectarse al PC que adquiere la data de las RTU. La ausencia de la remota es debida a la asignación de la misma a otro trabajo de grado, por lo que se tuvo que excluir de esta actualización del SCADA. Dicha falta no ocasiona modificación alguna de los objetivos del trabajo de grado por estar en funcionamiento el resto de las remotas Ducati. Respecto al cable de comunicaciones, se decide la sustitución completa del mismo por uno de longitud suficiente para ser canalizado por las canaletas del Laboratorio de Máquinas, y llegar al computador seleccionado para la adquisición de los datos de las remotas. En la Figura 25, se muestran imágenes del banco de pruebas y de la canalización realizada.



Figura 25. Banco de Pruebas y Canalización realizada.

2.2.2. Actualización anterior del SCADA bajo iFix

Al analizar el estado de la actualización anterior del SCADA, se consigue con que se encuentra configurado en cuatro computadores con hardware tipo servidor ubicados en el Laboratorio de Redes de Distribución. Las características de dichos PCs y una imagen de estos, son mostradas en la Tabla 3.




				
Nombre	Scada0	Scada1	Scada2	Scada3
Procesador	Pentium III 1,3 GHz	Pentium III 1,3 GHz	Pentium III 1,3 GHz	Pentium III 1,3 GHz
RAM	256 Mb	256 Mb	256 Mb	256 Mb
Disco Duro	40 Gb	40 Gb	40 Gb	40 Gb
S.O.	Win2000 Server	Windows 2000	Windows 2000	Windows 2000
Red	2 Tarjetas 10/100	2 Tarjetas 10/100	2 Tarjetas 10/100	2 Tarjetas 10/100
Función	<ul style="list-style-type: none"> • Adquisición de data de las RTU. • Histórico (No Funcional). 	<ul style="list-style-type: none"> • Servidor Principal. • HMI. 	<ul style="list-style-type: none"> • Estación de Ingeniería. • Comunicación con Gateway. 	<ul style="list-style-type: none"> • Servidor Secundario. • HMI.

Tabla 3. Características de los computadores del LRD.

Entre los detalles que se hallaron respecto a la configuración física y el estado de los dispositivos del SCADA, destacan dos: El primero referente a la falta de un convertidor 232/485 o en su defecto de una tarjeta de red 485; y el segundo respecto a la necesidad de un punto de red adicional para la conexión a Ethernet de un computador faltante. Ante la ausencia de un convertidor 232/485 o de una tarjeta 485 en alguno de los computadores pertenecientes al SCADA, se genera la necesidad de diseñarlo. Luego de distintas pruebas y varias modificaciones en el diseño se llega al esquemático mostrado en la Figura 26.

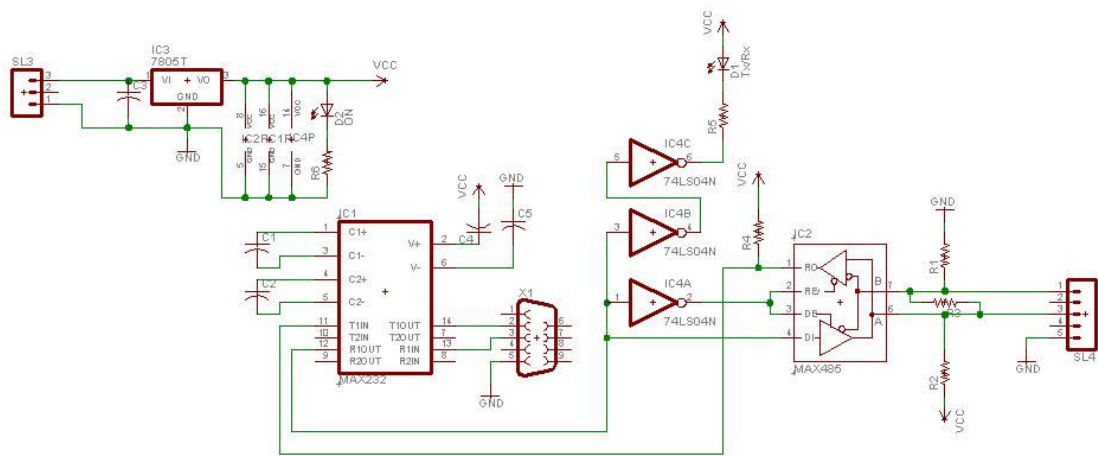


Figura 26. Esquemático del Convertidor 232/485 diseñado.

El circuito se basa en los integrados MAX232 y MAX485 para su funcionamiento. El MAX232 convierte la lógica y los niveles de tensión de las señales en estándar RS232 a niveles TTL. Luego, empleando un método sencillo el MAX485 convierte las señales a modo diferencial cumpliendo con el estándar RS485, de igual forma, se realiza el proceso contrario. Para poder entender el funcionamiento del circuito, es necesario tener un conocimiento básico de la comunicación serial. Lo importante radica en conocer que un 0 lógico inicia una transmisión, un 1 lógico la termina y mientras el canal está inactivo debe verse siempre un 1 lógico en el canal. Con esto en mente, el circuito es diseñado para que los canales se queden en un 1 lógico mientras no sea transmitido un 0 desde cualquiera de los dos lados.

En caso de ser enviado un cero lógico por parte del dispositivo conectado al terminal 232, el MAX232 convierte el nivel lógico a un 0 lógico TTL, lo que obliga al buffer inferior del MAX485 a pasar de alta impedancia a un estado, tal que, la diferencia de tensión entre A y B sea menor a 0 (es decir, $A < B$). Si el cero lógico es enviado por parte del dispositivo conectado al terminal 485, el MAX485 mide una diferencia de tensión entre A y B menor a 200 mV (o sea, $B - A > 200 \text{ mV}$), lo que coloca el pin 1 de dicho dispositivo en un estado lógico bajo, que es convertido posteriormente a los niveles de tensión adecuados por el MAX232.

Es importante destacar que el circuito es diseñado con la función de escuchar al dispositivo conectado del lado del terminal 485 siempre que no haya una transmisión activa, aunque en caso de existir una colisión en la transmisión, el del terminal 232 tendrá la prioridad en la comunicación.

Por motivos de estética, funcionalidad y trascendencia en el tiempo, el montaje final es realizado sobre circuito impreso. El diseño del Board es realizado utilizando la versión de prueba del programa Eagle de CadSoft, mientras que el proceso de impresión sobre la baquelita es realizado empleando la técnica de papel transfer. En la Figura 27 se muestran imágenes del board y del circuito impreso final, mientras que en el Anexo 2 se expone la lista de componentes utilizados.

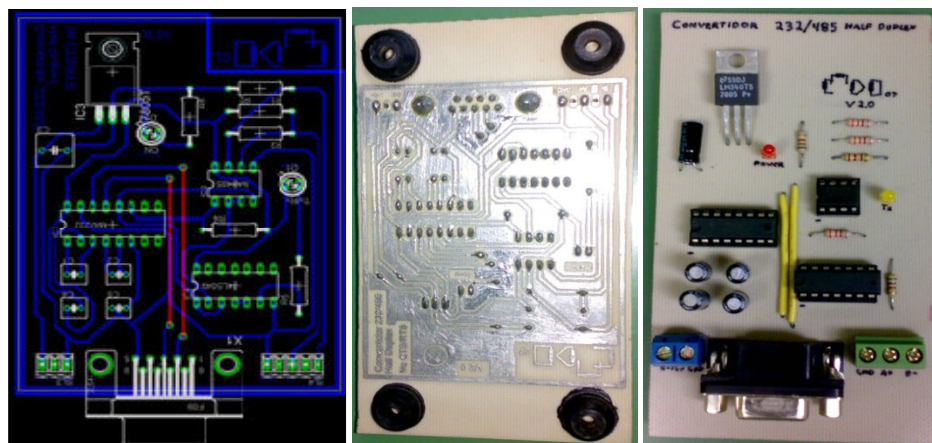


Figura 27. Board en el programa Eagle y Ambas caras del Circuito Impreso.

El otro detalle que se nota respecto al estado del SCADA actual, es la falta de un punto adicional de red en el Laboratorio de Redes de Distribución. Aunque esta conexión a Ethernet es totalmente necesaria para la operación del SCADA, no se encuentra el cable adicional canalizado en el laboratorio. Para solventar este problema, se opta por adquirir un Switch de 5 puertos marca TP-Link y utilizarlo para crear el punto de red adicional necesitado. El Cable utilizado como Uplink del Switch es el que inicialmente se encontraba conectado a la máquina Scada1. Las conexiones para las PCs identificadas como Scada0 y Scada1 son derivadas del Switch. Una imagen que muestra la instalación de este dispositivo se muestra en la Figura 28.



Figura 28. Instalación del Switch.

2.2.3. Maqueta de Procesos

Al igual que en el resto de las etapas del SCADA, en la maqueta de procesos ubicada en el Laboratorio de Control se presentan problemas iniciales. El primero de ellos, se debe al no bombeo de agua por parte de la bomba de vaciado. Mientras tanto, el segundo problema presentado, es ocasionado por la desconfiguración de algunos parámetros del variador de frecuencia de la bomba de llenado, impidiendo el bombeo rápido del agua del tanque de depósito al tanque de procesos.

El problema presentado con la bomba de vaciado se debe a la tranca de su sistema interno producto de la gran cantidad de restos de metal oxidado presente en el agua. Esto obstaculiza la descarga de agua del tanque izquierdo al derecho, lo que impide completar el ciclo del proceso. A pesar de que se intenta reparar la bomba por todos los medios posibles, al final se sustituyó por una completamente nueva, logrando así el funcionamiento del sistema de vaciado. El óxido presente en el agua se cree que es debida a la corrosión interna del tanque de depósito de la maqueta. En la Figura 29 se muestra una imagen de la maqueta, de la nueva bomba de vaciado y del tanque de depósito.



Figura 29. Maqueta, Bomba de Vaciado (Azul claro) y Tanque de depósito.

Una de las etapas del sistema de llenado lo constituye el variador de frecuencia marca Mitsubishi. Dicho dispositivo permite configurar parámetros importantes para el control de un motor, tales como, el tiempo de aceleración, el tiempo de frenado, las frecuencias de salida mínima y máxima, la curva Tensión/Frecuencia (V/F) y otros. Adicionalmente, el variador ofrece una protección electrónica al motor midiendo los parámetros eléctricos de este, y en caso de sobrecarga, evita que entre en un estado no deseado y se dañe. Este fue reconfigurado siguiendo las instrucciones del manual de uso. El variador de frecuencia Mitsubishi instalado en la Maqueta de procesos es mostrado en la Figura 30.



Figura 30. Vista Externa e interna del cajetín y Variador de Frecuencia en detalle.

2.2.4. Computadores para la actualización del SCADA

En un principio, se pensaba utilizar únicamente los cuatro computadores del LRD para la implementación de la actualización del SCADA, pero debido a que el iFix no puede coexistir en un mismo computador con cualquier programa del paquete Wonderware no diseñado para esa función, se ve en la necesidad de emplear otros.

A pesar de que ambos programas pueden estar instalados en un mismo computador, los especialistas de Wonderware recomiendan no realizarlo de esta forma, ya que, se generan problemas de comunicación si los dos intentan intercambiar información con otros dispositivos externos, debido al uso del DCOM y otros componentes. Esto condujo a la necesidad de añadir nuevos computadores para poder realizar la instalación de los productos Wonderware, y por ende, la actualización del SCADA. Los PCs seleccionados se encuentran ubicados en el Laboratorio de Control, Instrumentación y Digitales (LCID), y sus características y funciones para el SCADA se muestran en la Tabla 4.




			
Nombre	SW0	SW2	Control8
Procesador	Pentium IV HT 3,0 GHz	Intel Celeron 1,8 GHz	Pentium IV 3,2 GHz
RAM	512 Mb	256 Mb	512 Mb
Disco Duro	80 Gb	40 Gb	80 Gb
S.O.	Windows XP	Windows 2000	Windows 2000
Red	1 Tarjeta 10/100	1 Tarjeta 10/100	1 Tarjeta 10/100
Función	<ul style="list-style-type: none"> • Estación de Ingeniería. • AOS Principal. 	<ul style="list-style-type: none"> • Histórico de datos. 	<ul style="list-style-type: none"> • HMI.

Tabla 4. Características de los computadores y su función para el SCADA.

Aunque estos computadores resuelven en parte el problema de los programas Wonderware, aún se debe emplear al menos 1 PC del Laboratorio de Redes de Distribución para realizar la adquisición de datos de las remotas y la comunicación con el iFix. El computador seleccionado para esta función resulta ser el que tiene bajo nombre Scada2, debido a que contiene el objeto OPC del iFix y se encuentra suficientemente cerca de la canaleta con el cable de la red 485.

Además, se requiere de un PC adicional para la implementación de la redundancia del Servidor de Objetos de Aplicaciones (AOS). En este caso, se selecciona un computador portátil de forma temporal, esperando la adquisición o asignación de un computador de mesa para ser sustituida. El esquema físico de conexiones implementado queda tal como se muestra en la Figura 31.

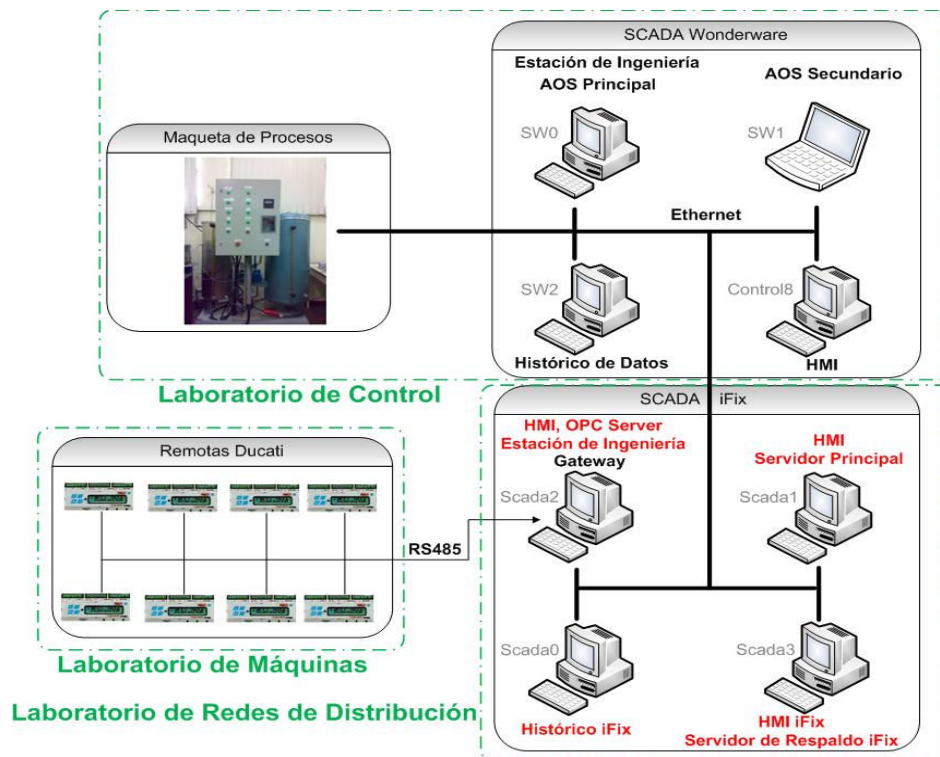


Figura 31. Esquema posterior a las modificaciones iniciales del SCADA.

2.3. DESCRIPCIÓN DEL SOFTWARE EMPLEADO

En los siguientes puntos se detallan los requisitos de instalación de los programas para el SCADA y su configuración, se explica la programación del PLC DirectLogic de la Maqueta de procesos y se especifican las pruebas de comunicación realizadas y modificaciones debidas a esto.

2.3.1. Computadores Restantes del SCADA

Debido a que el computador Scada0 del Laboratorio de Redes de Distribución no poseía un uso funcional como histórico del iFix, se solicita formatear dicho PC para de ser empleado como AOS Redundante y eliminarle completamente cualquier programa instalado relacionado con el anterior SCADA.

Como el programa Industrial Application Server requiere un Sistema Operativo tipo Servidor, al computador Scada0 se le coloca el Windows 2000 Server. La razón por la cual se instala un sistema operativo relativamente viejo, en vez de uno más reciente, se debe a que el computador no posee los requerimientos mínimos necesarios para ser instalado un Windows 2003 Server o uno posterior. Con el objeto de comprobar las potencialidades en la comunicación de Wonderware, se solicitan dos computadores adicionales del LCID para ser utilizados como HMI, uno de ellos (Control1) se formatea completamente instalándose WinXP Profesional (para conexión con el histórico), mientras que el otro es dejado con su Win2000 inicial (Control8). La razón de emplear Windows XP en Control1 se debe a que el componente que grafica la data del histórico no es compatible con Windows 2000.

Los computadores identificados como SW0 y SW2 se pusieron a punto, instalándose como nuevo Sistema Operativo el Windows 2003 Server. En el caso de SW0, debido a que es requerido para la instalación del Industrial Application Server; de igual forma, representa un requisito en SW2 para la instalación del Historian. Un punto importante respecto a los Sistemas Operativos Windows, es que se deben deshabilitarse las actualizaciones automáticas para evitar que se instale o actualice algún componente que pueda generar el mal funcionamiento de los programas instalados en las computadoras. De igual forma, es recomendable la instalación de los productos Wonderware sobre sistemas operativos Windows en inglés, y únicamente instalar los programas mínimos necesarios en cada uno de los PCs.

2.3.2. Programación del PLC DirectLogic de la Maqueta de Procesos

Debido a que no existe una aplicación exclusiva de dicha maqueta, se realiza la programación de un proceso tipo batch genérico de un llenado y vaciado de tanques, dejando abierta la posibilidad de efectuar un control manual de las variables de la maqueta por software.

La maqueta de procesos realizada en el año 2003 por Rafael Giordano [36], consiste en un sistema de dos tanques con bombas de agua, resistencias térmicas, agitador y sensores de distintos tipos controlados con un PLC, para realizar prácticas de laboratorio en la escuela de Ingeniería Eléctrica. En la Figura 32, se muestra un esquema general de la maqueta de procesos donde se relaciona las entradas y salidas del PLC con los procesos.

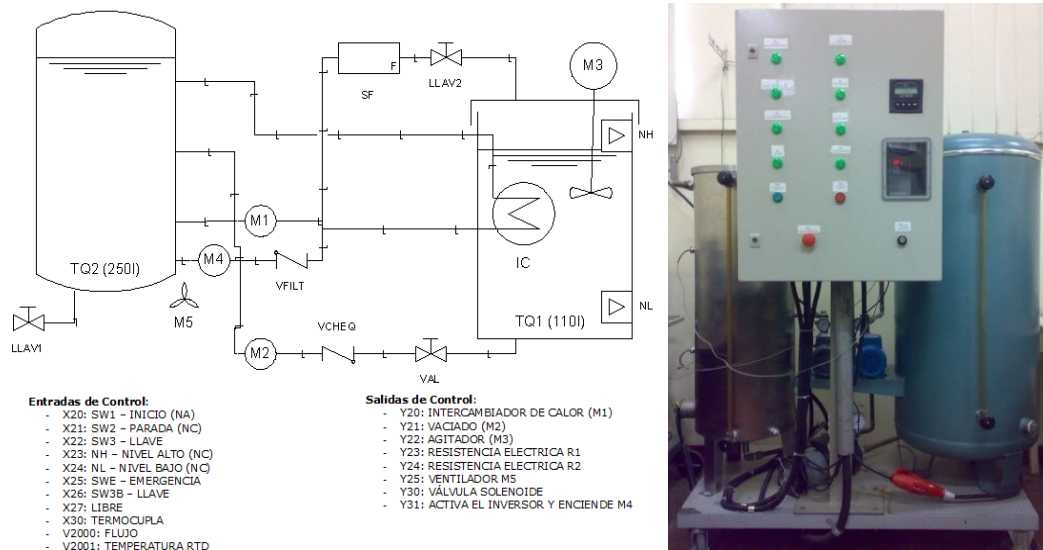


Figura 32. Esquema general de la Maqueta de Procesos y su relación con el PLC.

Debido a que se quiere realizar un control automático de la maqueta pero también tener la posibilidad de efectuar un control manual, la programación del PLC se realiza en Diagrama de Escaleras con la idea de utilizar el bloque EDRUM para efectuar el control automático, y la lógica de contactos para el control manual de los dispositivos. El bloque EDRUM es una de las bondades que presenta el programa DirectSoft, y permite la configuración por tiempo y eventos de un proceso tipo batch.

El Graficet del programa para el PLC de la Maqueta de Procesos es mostrado en la Figura 33. La programación en detalle del Diagrama de Escaleras y la explicación de cada una de las líneas se realiza en el Anexo 3.

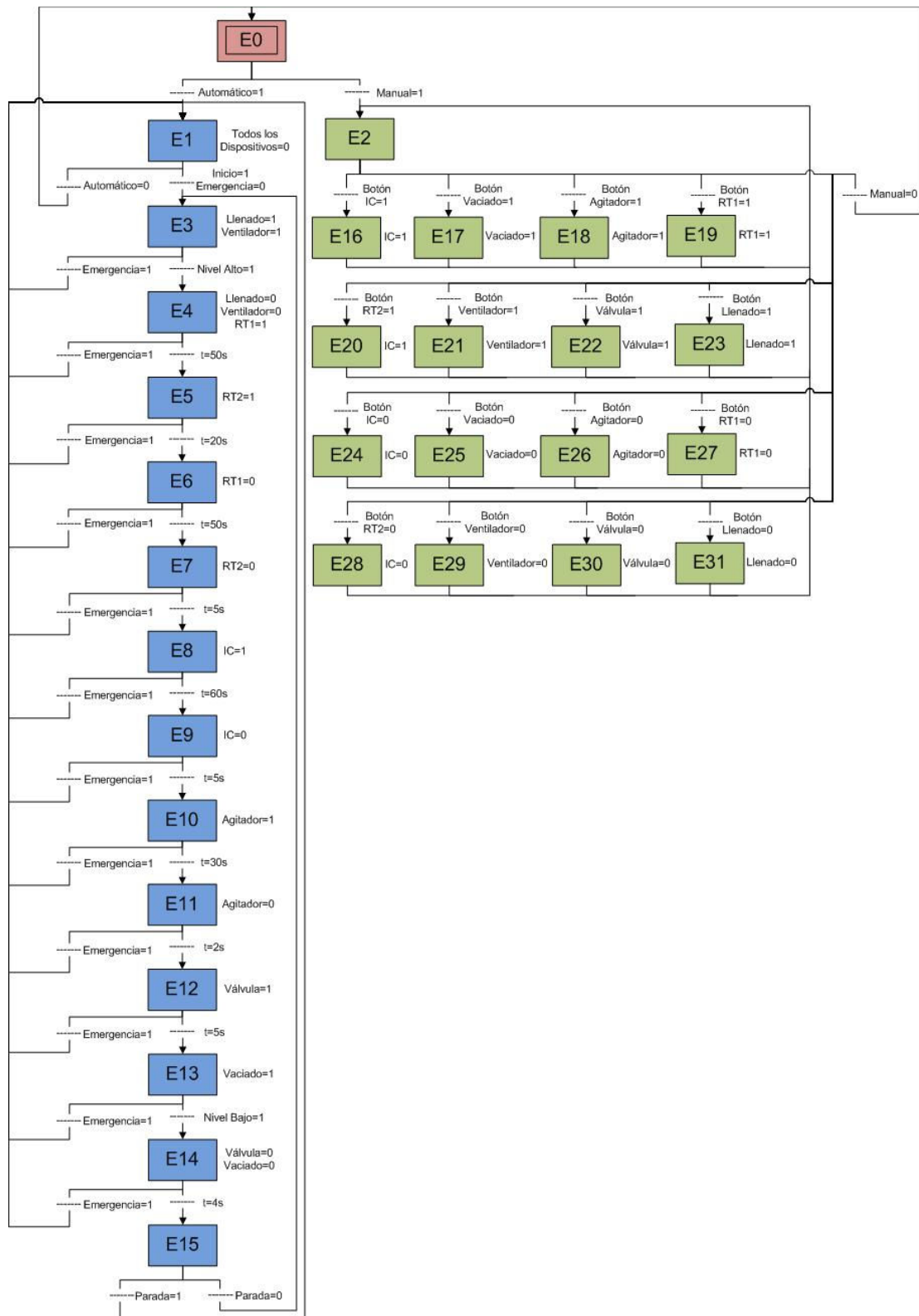


Figura 33. Grafcet para el Control de la Maqueta.

2.3.3. Componentes de Configuración para una Arquitectura con Wonderware

Una vez realizada la programación del PLC, la atención se vuelca a la configuración de los computadores bajo la plataforma Wonderware. A continuación se detallan los pasos para la instalación de los diferentes componentes de configuración para una arquitectura basada en Wonderware:

2.3.3.1. Application Object Server (Servidor de Objetos de Aplicaciones)

Para poder realizar la configuración de un computador como Servidor de Objetos de Aplicaciones (AOS), es necesaria la instalación del Industrial Application Server (IAS) 2.1. Como la idea es que el computador bajo esta función ejecute todos los objetos de Wonderware, únicamente hace falta la instalación del Bootstrap, que es el objeto base para las comunicaciones entre los productos Wonderware [35].

Debido a que se suelen hacer modificaciones de configuración desde algunos nodos remotos que presentan los objetos en ejecución, resulta conveniente la instalación del programa Integrated Development Environment (IDE) incluido en el IAS. A través de este programa se realizan las configuraciones de todos los objetos locales y remotos de una Galaxia. La Galaxia es el nombre dado al sistema que engloba todas las aplicaciones y equipos de la automatización [35].

Una vez instalados los programas anteriores, se deben desplegar las Plataformas y los Engines en cada una de las computadoras que cumplan esta función. Una Plataforma es un objeto que describe un PC dentro de la Galaxia. Un Engine por el contrario, es un objeto necesario para ejecutar todos los elementos que corren en una Galaxia. El Engine debe estar contenido en una Plataforma para poder realizar su función.

2.3.3.2. Configuration Database (Base de datos de Configuración)

Para la configuración de este nodo es necesaria la previa instalación del Microsoft SQL Server en un SO tipo Servidor. La versión del producto seleccionada para tal fin es el SQL Server 2000, actualizándose posteriormente a SQL Server 2005. Una vez realizado esto, se procede a la instalación del Industrial Application Server (IAS) 2.1, seleccionando los componentes Bootstrap y Galaxy Repository. El Galaxy Repository (Repositorio de Datos) es un programa incluido en el paquete del IAS que administra los datos de configuración asociados con una o más Galaxias [35]. Este nodo debe contener una Plataforma desplegada para funcionar adecuadamente.

Al igual que con el AOS, resulta conveniente instalar el IDE en este tipo de componentes de configuración, con la finalidad de realizar modificaciones desde el mismo nodo a objetos de la Galaxia. Desde el punto de vista práctico, este componente se suele implementar junto con la estación de ingeniería en un mismo PC. El computador empleado en la mayoría de los casos resulta ser una portátil por cuestiones de movilidad y protección del Repositorio de Datos (Galaxy Repository).

2.3.3.3. Engineering Station (Estación de Ingeniería)

Todos los tipos de topologías recomendadas incluyen un computador configurado como estación de ingeniería. Para la configuración de este componente es necesaria la instalación del Bootstrap y del IDE. Debido a que la estación de ingeniería contiene todos los programas para realizar las configuraciones y programaciones de los objetos, resulta necesario instalar el programa para la realización de Interfaces Hombre-Máquinas de Wonderware llamado InTouch WindowMaker. El System Management Console (SMC) es el último producto requerido por este componente de configuración. Este programa no es más que una consola que permite realizar varias funciones tales como, apertura de archivos log locales y remotos, Administración de DA Servers y Administrar las plataformas a través de una Interfaz simple.

2.3.3.4. Historian Node (Nodo Historiador)

Este nodo requiere del IndustrialSQL Server, que es un programa para almacenar toda la data histórica y proporcionar datos en tiempo real a clientes tales como ActiveFactory e Information Server. Este componente de configuración no requiere del Bootstrap debido a que emplea el DCOM para el envío de datos, pero si necesita la instalación previa del MS SQL Server 2000 o posteriores. Una vez instalado este último, es actualizado a su versión del 2005 y posteriormente se instala el InSQL Server. Es común combinar el Histórico de datos con el de Alarmas, pero como el programa del Histórico de Alarmas está contenido en la instalación del InTouch, se requiere la instalación de este producto una vez realizada la del InSQL.

2.3.3.5. Visualization Node (Nodo Visualizador)

Este no es más que un computador ejecutando un HMI bajo InTouch. Este nodo requiere de la instalación del Bootstrap y del despliegue de una plataforma para poder realizar la comunicación entre la galaxia y el InTouch.

2.3.3.6. I/O Server Node (Nodo de Servidores I/O)

Existen varias formas de configurar un computador como Nodo de Servidores I/O. En todas se recomienda la instalación del SMC para poder realizar la configuración local del Servidor y depurar errores de este. Dependiendo del objeto que se esté instalando, es requerida la instalación del Bootstrap, del despliegue de una Plataforma y del despliegue de un Engine, además del despliegue del Servidor instalado. Un ejemplo donde no es requerido el Bootstrap, la Plataforma y el Engine, lo constituye la instalación del Factory Suite Gateway. En este únicamente es requerido el SMC y el objeto para la implementación del nodo.

2.3.3.7. Web Server (Servidor Web)

Es un nodo que presenta el Information Server. Este permite acceder a datos en tiempo real de una Galaxia. Solo necesita del Bootstrap, una plataforma desplegada y del Information Server instalado en un computador.

2.3.4. Pruebas de Comunicación

Debido a los problemas presentados con el estado inicial del SCADA y de los dispositivos en general, se decide realizar la inversión de dos puntos de la metodología. Es por ello que se adelantan las pruebas de comunicación y se efectúan antes de la definición del modelo de planta y de la convención de nombres. Las pruebas realizadas incluyen 3 tipos de dispositivos, a continuación son explicadas:

2.3.4.1. Comunicación entre el PLC y el SCADA

El PLC DirectLogic instalado en la maqueta de procesos posee un modulo de comunicaciones Ethernet que no acepta Modbus TCP como protocolo. Aunque otro de los módulos comerciales si admite Modbus TCP, el instalado sólo acepta los protocolos propietarios de AutomationDirect, y Wonderware no posee los objetos para la comunicación en dichos protocolos. Esto obliga la búsqueda de aplicaciones y/o programas de otros vendedores (Third-Party) para la resolución del problema.

En la búsqueda realizada se consiguieron dos programas que permiten la conversión a OPC de los protocolos propietarios de AutomationDirect con que se comunica el PLC. El primero es un programa de la misma empresa fabricante de los PLC llamado KEP Direct, y el otro es un programa de un Third Party certificado de Wonderware bajo nombre de TOPServer. Después de realizar distintas pruebas, ambos funcionaron de forma adecuada, pero se decidió emplear el TOPServer debido a que posee mayor variedad de objetos de comunicación con otros productos y la empresa que lo realiza está certificada por Wonderware.

En un principio, se seleccionó el computador SW0 para instalar el TOPServer, pero analizando un poco la función de AOS principal de este computador, si se produjera una falla en este equipo, se perdería totalmente la comunicación con la maqueta porque el redundante sería incapaz de comunicarse con el TOPServer. Esto obliga a utilizar otro computador para instalar el programa.

Luego de buscar entre los equipos disponibles, se selecciona el computador Scada2 para realizar la instalación de este producto. La razón de la elección, se debe a que este producto posee el servidor OPC del iFix, y debido a que este computador ya cumple una función inherente de I/O Server, es seleccionado para amoldarse lo más posible a la topología recomendada por Wonderware. Una vez instalado en su destino final se realiza la prueba de comunicación correspondiente, la cual consta de dos partes. La primera correspondiente a la comunicación del TOPServer con la Maqueta y la segunda referente a la adquisición de datos del TOPServer por parte del IAS. En la Figura 34 se muestra el PLC de la Maqueta y el TOPServer durante la prueba.

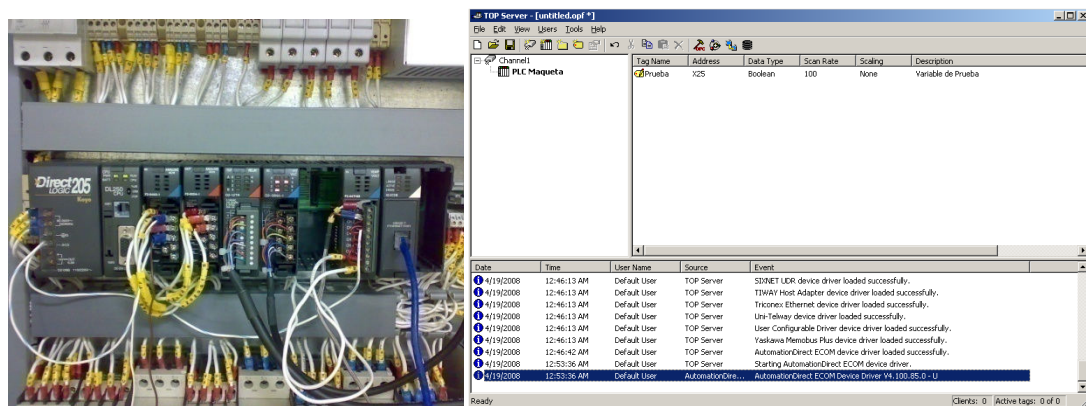


Figura 34. PLC de la Maqueta y Prueba del TOPServer.

La primera parte de la prueba se realiza configurando un Canal en el TOPServer para el protocolo ECOM y un Dispositivo DL250 correspondientes al PLC de la Maqueta. Una vez configurado adecuadamente, se crea un Tag que apunte a alguna dirección conocida del PLC y se ejecuta el QuickClient (un cliente de prueba) para verificar el valor adquirido. La segunda parte de la prueba consiste en la configuración de un objeto tipo OPC en el IDE para ser ejecutado en un Engine sobre la Plataforma del computador SW0. La idea de esta es adquirir la data del Tag previamente creado en el TOPServer y mostrarlo en el Object Viewer para verificar la comunicación. Ambas pruebas resultaron exitosas, lográndose la comunicación entre el PLC y Wonderware.

2.3.4.2. Comunicación entre las RTU y el SCADA

Se intentó establecer la comunicación entre el Gateway Ducati-Modbus TCP y el IAS, para ello se configuró un objeto de comunicación Modbus TCP y se desplegó en conjunto con el Engine y la plataforma del Computador SW0. Por más que se variaron los parámetros más comunes del objeto Modbus TCP en el IDE, no se logró establecer la comunicación con el Gateway, esto motivó a la búsqueda de un método alternativo para realizarlo.

Debido a que el TOPServer posee Modbus TCP entre sus objetos de comunicaciones, se realiza una prueba por ensayo y error (primero mediante el uso del QuickClient, y luego comunicándose IAS – TOPServer - Gateway), lográndose adquirir los parámetros de las Remotas.

Debido a que los objetos de comunicación Wonderware siguen estándares internacionales, hace pensar que el protocolo Modbus TCP programado en el Gateway es una variación del estándar, que es reconocida por el TOPServer más no por Wonderware. La configuración del TOPServer, así como del resto de los programas es detallada más adelante en este trabajo.

Una vez logrado el intercambio de información con el Gateway, se realiza otra prueba de comunicación. En este caso se instala el Gateway en el mismo PC que contiene el TOPServer, no lográndose establecer la comunicación debido a que el Gateway sólo acepta conexiones entrantes desde un PC remoto. Esto obliga a la necesidad de buscar un computador adicional que posea el TOPServer y que cumpla la función de un segundo I/O Server. Luego de analizar nuevamente los equipos disponibles, el PC más idóneo para este trabajo resulta ser Scada1, esto debido a que no requiere la instalación del Bootstrap y no genera conflicto con el iFix. En definitiva, para no tener dos TOPServer en funcionamiento, la comunicación con la Maqueta de Procesos y con las Remotas es realizada en un mismo Servidor (con sólo un TOPServer instalado) que se ubica en la máquina Scada1.

Aunque el TOPServer admite la comunicación con Wonderware empleando SuiteLink, un protocolo propietario de Wonderware diseñado para realizar su comunicación interna de manera eficiente, se establece OPC como forma de comunicación con Wonderware. Esto con la finalidad de tener variedad en los objetos de comunicación y para emplear uno de los protocolos para redes industriales más conocidos a nivel mundial, a pesar de ser menos eficiente y confiable que el SuiteLink de Wonderware.

2.3.4.3. Comunicación entre el iFix y el SCADA

Se programa un Objeto de comunicaciones OPC en el IDE y se realiza la búsqueda del servidor OPC del iFix ubicado en Scada2. El resultado de esta búsqueda es negativo, por lo que se procede a revisar la configuración del Servidor OPC de iFix. Este a pesar de estar activo, no contiene ningún elemento interno añadido a su base de datos. Debido a que la cantidad de Tags a agregar eran excesivos, se procedió a añadir sólo los de una remota definida en la base de datos de iFix (iA8), y la de dos variables de prueba creadas para realizar las pruebas de comunicación. Para mayor información sobre la convención de nombres ver el Anexo 4.

Al añadir dichos Tags el servidor se mantiene inaccesible, y debido a que no se puede instalar el Bootstrap en Scada2 (por la incompatibilidad entre este objeto y el iFix), se realiza la instalación del FactorySuite Gateway para intentar comunicarse con el OPC del iFix desde el mismo nodo. Efectivamente, este Servidor logra acceder al OPC de forma local, por lo que se termina empleando como Gateway entre el iFix y Wonderware. Una vez logrado acceder al OPC del iFix, se crea un objeto de comunicación SuiteLink en el IDE y se despliega en el Engine y Plataforma del PC SW0. La función de este objeto es la de comunicarse con el FactorySuite Gateway utilizando el protocolo propietario SuiteLink de Wonderware. La prueba de comunicación realizada incluye la adquisición de los Tags de una de las remotas y los valores de las variables de prueba creadas en la base de datos de iFix.

La prueba finaliza con resultados variados, ya que, la adquisición de datos de las variables de prueba se realiza de forma correcta, pero no se logra obtener los valores de los Tags de la Remota. Al indagar un poco más en la configuración y el funcionamiento del iFix, se logra notar que los Data Access apuntaban a una dirección IP que no correspondía con la ubicación del Gateway. Al realizar el cambio de dicha dirección y dejar funcionando el sistema por un tiempo (medio día), no se observó ningún valor adquirido correspondiente a las RTU Ducati.

Como el iFix no logra adquirir la data de las remotas, se opta por modificar la pantalla de inicio del HMI y agregar una zona de prueba para la comunicación con Wonderware. Esta zona de pruebas contiene una variable entera de 8 bits, una cadena de caracteres y un botón. Ambas variables pueden ser modificadas con un valor cualquiera dentro de los rangos permitidos. Al presionar el botón, se incrementa en uno el valor de la variable entera, y se coloca en la cadena de caracteres si este es par o impar. Estas dos variables son adquiridas correctamente por el IAS. En la Figura 35 se muestra una imagen de la pantalla de inicio original y modificada del iFix:



Figura 35. Pantalla de Inicio Original y Modificada del iFix.

2.4. ESQUEMA FINAL Y DESCRIPCIÓN DE LOS EQUIPOS

Luego de resolver cada uno de los problemas y siguiendo la topología recomendada por Wonderware, se realiza el Esquema Final de la distribución física del SCADA tal como se muestra en la Figura 36.

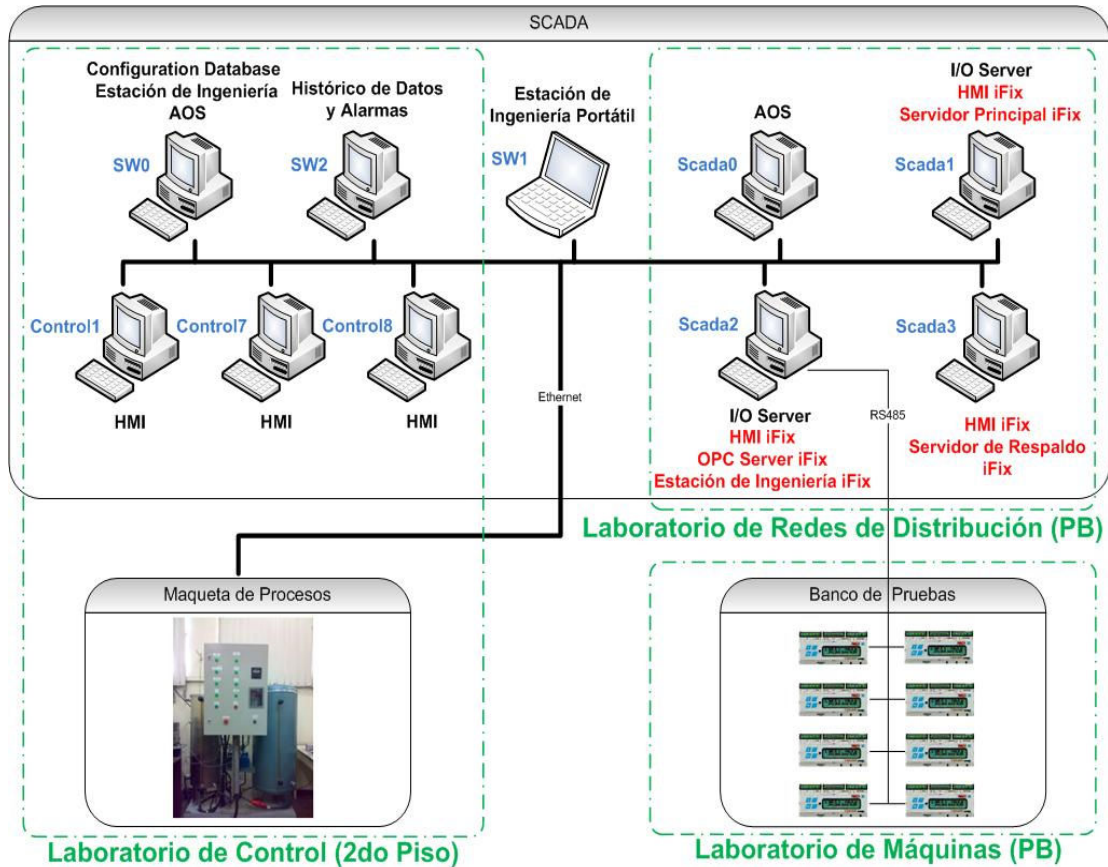


Figura 36. Esquema Final de la distribución física del SCADA.

Una Imagen que contiene el esquema final del SCADA según los niveles y el tipo de dispositivo, es mostrada en la Figura 37. Un resumen de los equipos y dispositivos que forman parte del SCADA son detallados después de la figura anteriormente mencionada. Los inicios de sesión de Windows y las contraseñas de los computadores empleados en el presente trabajo son mostrados en el Anexo 6.

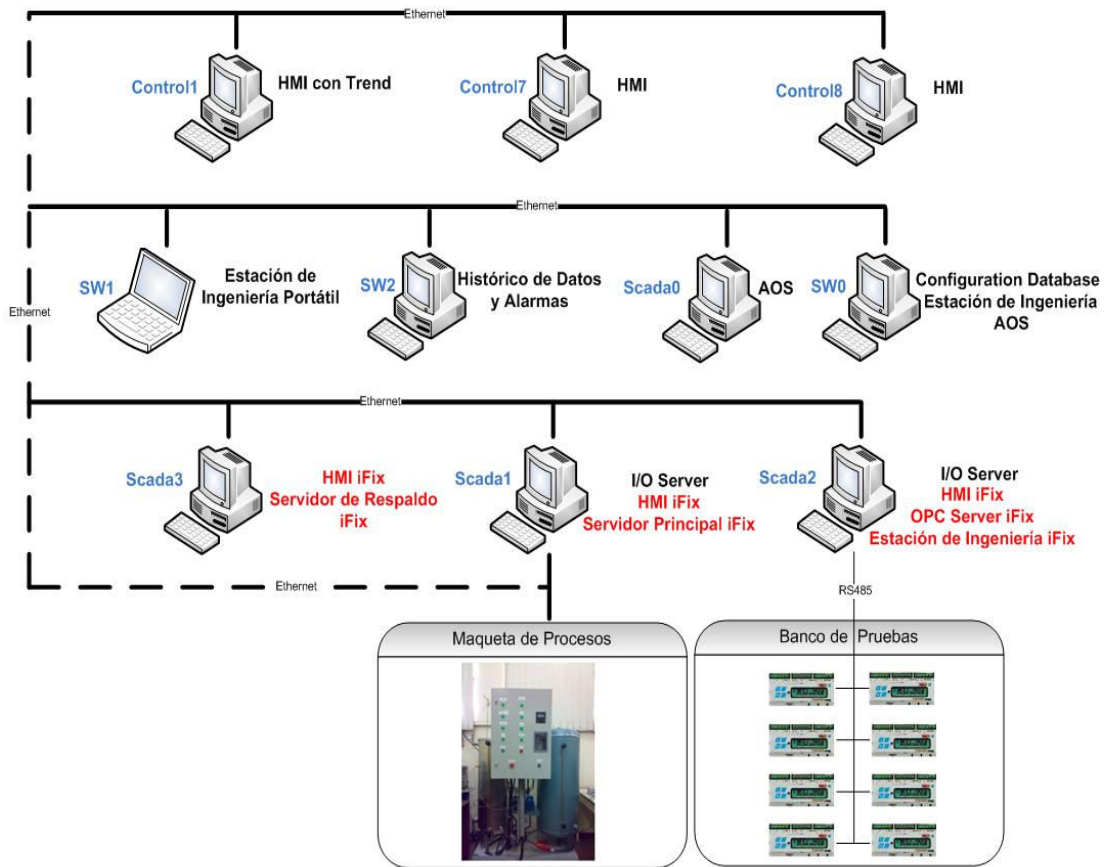


Figura 37. Esquema Final del SCADA según niveles.

Banco de Pruebas

- **Función:** Adquirir los parámetros Eléctricos del tablero principal de la EIE-UCV.
- **Dispositivos:** 8 Remotas Ducati Mach 30 conectadas en una red RS485. Cada remota adquiere 210 Tags, 100 de los cuales fueron identificados como parámetros eléctricos (1680 Tags en total).

Control1

- **Función:** HMI con Acceso a Histórico de datos en tiempo real a través de ActiveFactory.
- **Hardware:** Pentium IV 3,2 GHz, 512 Mb de RAM, Disco Duro de 67,6 Gb y 1 Tarjeta de Red 10/100.
- **Sistema Operativo:** Windows XP Professional (Inglés).

- **Software:** Bootstrap, InTouch, ActiveFactory y Plataforma (Desplegada).

Control7

- **Función:** HMI sin Acceso a Histórico de datos en tiempo real.
- **Hardware:** Pentium IV 3,2 GHz, 512 Mb de RAM, Disco Duro de 67,6 Gb y 1 Tarjeta de Red 10/100.
- **Sistema Operativo:** Windows 2000 Profesional (Español).
- **Software:** Bootstrap, InTouch y Plataforma (Desplegada).

Control8

- **Función:** HMI sin Acceso a Histórico de datos en tiempo real.
- **Hardware:** Pentium IV 3,2 GHz, 512 Mb de RAM, Disco Duro de 67,6 Gb y 1 Tarjeta de Red 10/100.
- **Sistema Operativo:** Windows 2000 Profesional (Español).
- **Software:** Bootstrap, InTouch y Plataforma (Desplegada).

Maqueta de Procesos

- **Función:** Realizar procesos tipo batch. Utilizada en prácticas de varias materias como equipo a controlar. Se muestra en la Figura 32.
- **Dispositivos:** 1 PLC DirectLogic con módulos de entrada analógica, salida digital a relé, termopar y comunicaciones sobre Ethernet; 1 tanque de acero inoxidable para la realización de procesos; 1 tanque para almacenaje de agua; 3 bombas de agua, una para el llenado del tanque de procesos, otra para el vaciado del mismo y una última para bombear agua a través del intercambiador de calor; 1 intercambiador de Calor ubicado en el tanque de procesos con entrada y salida de agua del, y hacia el otro tanque; 1 agitador, 2 Resistencias Térmicas utilizadas para calentar la mezcla y 2 Sensores de Nivel ubicados en el tanque de procesos; Botones e indicadores ubicados en la parte posterior de un cajetín; Relés, sistemas de protección, y una Fuente Conmutada de 24V Telemecanique ubicados dentro del Cajetín; otros.

En resumen, se manejan 8 Entradas digitales, 2 posiciones de memoria correspondientes a la misma cantidad de Entradas analógicas de 8 bits y 8 Salidas digitales.

Scada0

- **Función:** AOS.
- **Hardware:** Pentium III 1,3 GHz, 256 Mb de RAM, Disco Duro de 36,4 Gb y 2 Tarjetas de Red 10/100.
- **Sistema Operativo:** Windows 2000 Server (Inglés).
- **Software:** MS SQL Server 2000, Bootstrap, IDE, Galaxy Repository, SMC, Plataforma (Desplegada) y Engine (Desplegado).

Scada1

- **Función:** I/O Server (contiene el TOPServer para la comunicación con el DirectLogic de la Maqueta de Procesos y el Gateway Ducbus-Modbus TCP), Servidor Principal y HMI del iFix.
- **Hardware:** Pentium III 1,3 GHz, 256 Mb de RAM, Disco Duro de 36,4 Gb y 2 Tarjetas de Red 10/100.
- **Sistema Operativo:** Windows 2000 Profesional (Español).
- **Software:** TOPServer y iFix.

Scada2

- **Función:** I/O Server (contiene el Gateway Ducbus-Modbus TCP para realizar la adquisición de datos de las remotas y el FactorySuite Gateway para realizar la comunicación con el Servidor OPC del iFix ubicado en esta misma PC), Estación de Ingeniería, Servidor OPC y HMI del iFix.
- **Hardware:** Pentium III 1,3 GHz, 256 Mb de RAM, Disco Duro de 36,4 Gb y 2 Tarjetas de Red 10/100.
- **Sistema Operativo:** Windows 2000 Profesional (Español).
- **Software:** Gateway Ducbus-Modbus TCP, FactorySuite Gateway, SMC y iFix.

Scada3

- ***Función:*** Servidor de Respaldo y HMI del iFix.
- ***Hardware:*** Pentium III 1,3 GHz, 256 Mb de RAM, Disco Duro de 36,4 Gb y 2 Tarjetas de Red 10/100.
- ***Sistema Operativo:*** Windows 2000 Profesional (Español).
- ***Software:*** iFix.

SW0

- ***Función:*** AOS, Configuration Database y Estación de Ingeniería.
- ***Hardware:*** Pentium IV 3,0 GHz, 512 Mb de RAM, Disco Duro de 76,3 Gb y 1 Tarjeta de Red 10/100.
- ***Sistema Operativo:*** Windows 2003 Server (Inglés).
- ***Software:*** MS SQL Server 2005, Bootstrap, IDE, Galaxy Repository, SMC, Plataforma (Desplegada), Engine (Desplegado) y DIRECTSoft.

SW1

- ***Función:*** Estación de Ingeniería portátil.
- ***Hardware:*** AMD Turion64 X2 1,67 GHz, 2 Gb de RAM, Disco Duro de 120 Gb y 1 Tarjeta de Red 10/100.
- ***Sistema Operativo:*** Windows Vista (Inglés) con Windows 2003 Server (Inglés) y Windows 2000 Server (Inglés) sobre una Máquina Virtual VMWare.
- ***Software:*** MS SQL Server 2005, Bootstrap, IDE, SMC, InTouch, DIRECTSoft, TOPServer y ActiveFactory.

SW2

- ***Función:*** Histórico de datos y Alarmas.
- ***Hardware:*** Intel Celeron 1,8 GHz, 256 Mb de RAM, Disco Duro de 38,2 Gb y 1 Tarjeta de Red 10/100.
- ***Sistema Operativo:*** Windows Server 2003 (Inglés).
- ***Software:*** MS SQL Server 2005, INSQL Server, ActiveFactory, SMC e InTouch (para el Histórico de Alarmas AlarmDB Logger).

CAPITULO III

IMPLEMENTACIÓN DEL SCADA CON WONDERWARE

Ya establecida la comunicación, se procedió a la definición de la convención de nombres y el modelo de las áreas en el Industrial Application Server. El modelo de las áreas o modelo de planta, identificado como Model View bajo el ambiente del IDE, define la relación entre los objetos y su representación física en el esquema de automatización. Para su definición, resulta necesario conocer el esquema físico de lo que se está automatizando.

Como el objetivo de este trabajo consiste en controlar una maqueta de procesos y adquirir los datos de unas remotas que se ubicarían en un futuro a lo largo de toda la UCV, se parten de estas premisas para la definición del Model View (ver Anexo 4). El Modelo de Planta del SCADA se muestra en la Figura 38.

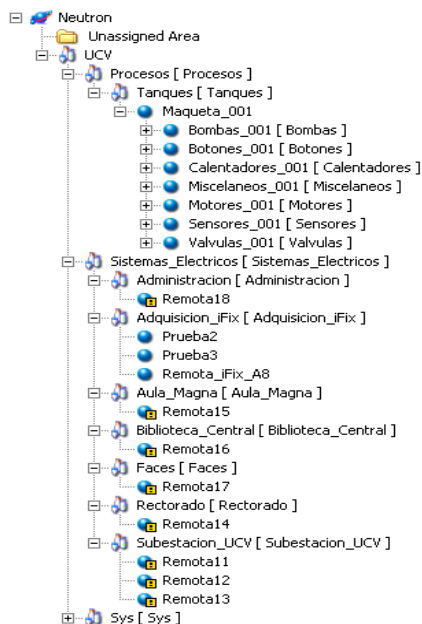


Figura 38. Modelo de Planta.

Lo importante de este modelo, es que facilita en gran medida la programación y permite olvidarse de la convención de nombres de los tags, por trabajar con objetos organizados en un esquema de planta dentro del mismo ambiente de programación.

Analizando más a fondo el Modelo de Planta, se observa que dentro de la Galaxia llamada “Neutron”, se define un área bajo el nombre “UCV” que engloba todos los objetos y viene a representar el lugar físico que contiene la automatización. Esta se subdivide en 3 categorías, la primera correspondiente a los Procesos, la segunda referente a los Sistemas Eléctricos y la última, a los componentes del Sistema que no pertenecen a ningún área física.

En la actualidad existe únicamente un proceso automatizado de tanques, por lo que en “Procesos” sólo se crea un área llamada “Tanques” que incluye el objeto “Maqueta” correspondiente a la Maqueta del LCID. De igual forma se especifican las áreas de los Sistemas Eléctricos según se tenía previsto en el SCADA original (ver Anexo 4) y se asocian las remotas correspondientes a estas.

Como se puede notar, no hace falta hacer una convención de nombres para declarar los Tags del sistema, sino más bien, de la especificación completa de los procesos a automatizar y del esquema global. Esto representa una de las ventajas de esta plataforma respecto a los sistemas tradicionales basados en Tags.

A partir de este punto, la programación y configuración de cada uno de los componentes del SCADA se realiza de forma paralela. A pesar de esto, se explicarán los procedimientos seguidos para la configuración de cada uno de los componentes de forma separada, comenzando por los programas y objetos de adquisición de datos, seguido de los programas para la configuración y programación del SCADA y por último el software empleado para la realización del HMI.

3.1. CONFIGURACIÓN DEL SOFTWARE PARA LA ADQUISICIÓN DE DATOS

3.1.1. TOPServer

Es un programa de la empresa Software Toolbox, un Third party de Wonderware, que posee una gran cantidad de objetos para la comunicación con dispositivos industriales convencionales y poco convencionales. En este caso se instaló el demo del programa descargado de la página web de la empresa [37], este tiene una duración de 2 horas a partir de su momento de ejecución. Una vez instalado el programa, se procedió a la configuración de un canal y un dispositivo para la comunicación con el PLC DL205 empleando el protocolo DirectLogic ECOM. En este caso sólo se tuvo que seleccionar el modelo del PLC y la dirección IP del mismo, dejando el resto de los parámetros por defecto. La configuración de la comunicación del TOPServer con el PLC de la Maqueta se muestra en la Figura 39.

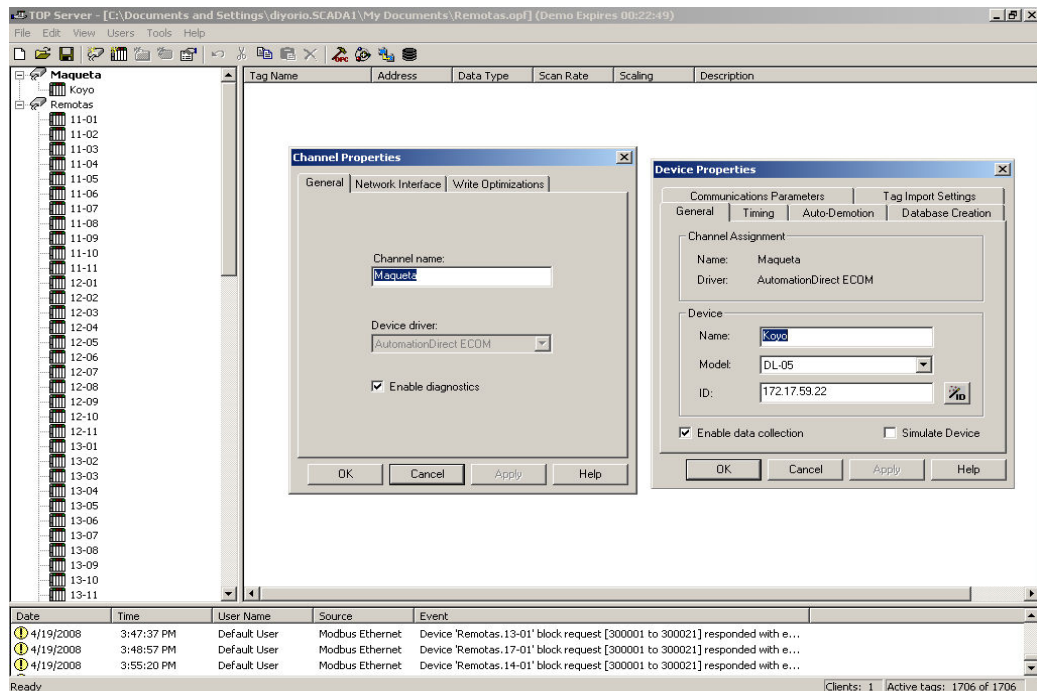


Figura 39. Configuración de la comunicación con el PLC de la Maqueta.

Ya configurado el enlace, se procedió a añadir cada uno de los Tags para las variables de entrada, salida, posiciones de memoria y variables especiales. En la Figura 40 se muestra el menú de configuración del Tag Y25 de la Maqueta.

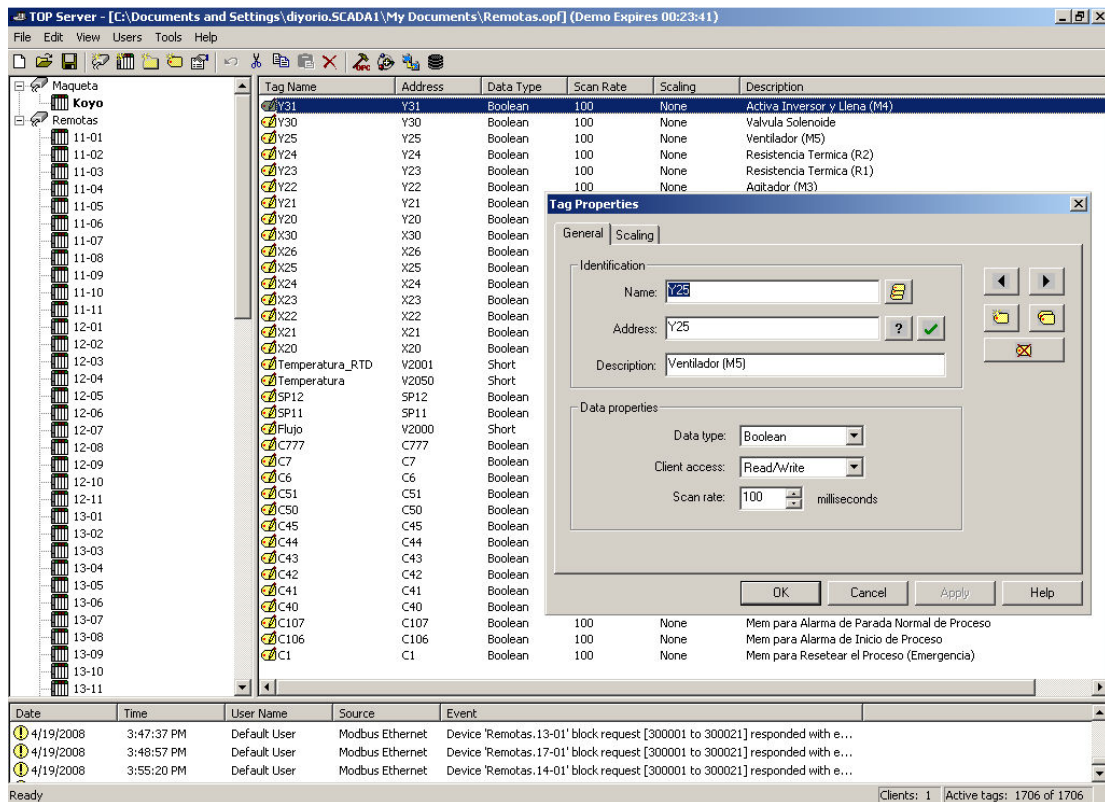


Figura 40. Configuración de uno de los Tags de la Maqueta.

Al igual que en el caso anterior, el enlace con el Gateway de las remotas requiere de un canal. A diferencia de la Maqueta, este es seleccionado del tipo Modbus Ethernet y se añaden tantos dispositivos como sea necesario para realizar la adquisición de todos los parámetros de las remotas. La cantidad máxima de Tags por dispositivo permitidos por el programa es de 30. La configuración del canal y de uno de los dispositivos es mostrada en la Figura 41, mientras que el resto de los parámetros no incluidos son dejados por defecto. Para la configuración de los otros dispositivos sólo es necesario cambiar el ID para direccionar al IP del Gateway y el número correspondiente de la remota.

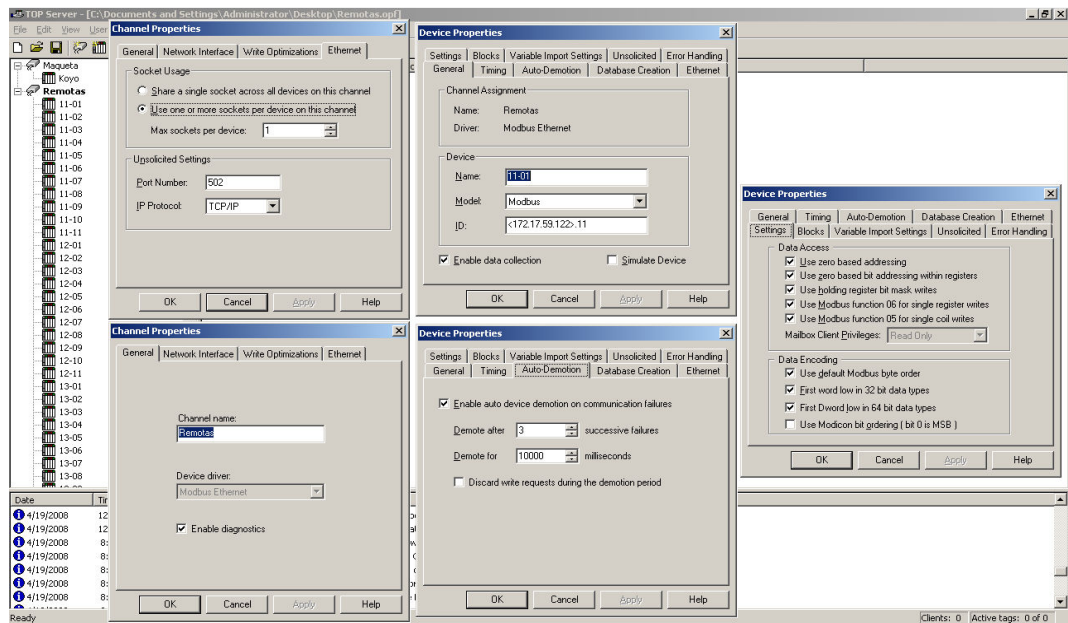


Figura 41. Configuración de la comunicación con el Gateway de las Remotas.

Una vez configurado el enlace, se procedió a añadir los Tags para realizar la adquisición de todas las variables de cada una de las remotas. En la Figura 42 es mostrada la configuración de uno de estos.

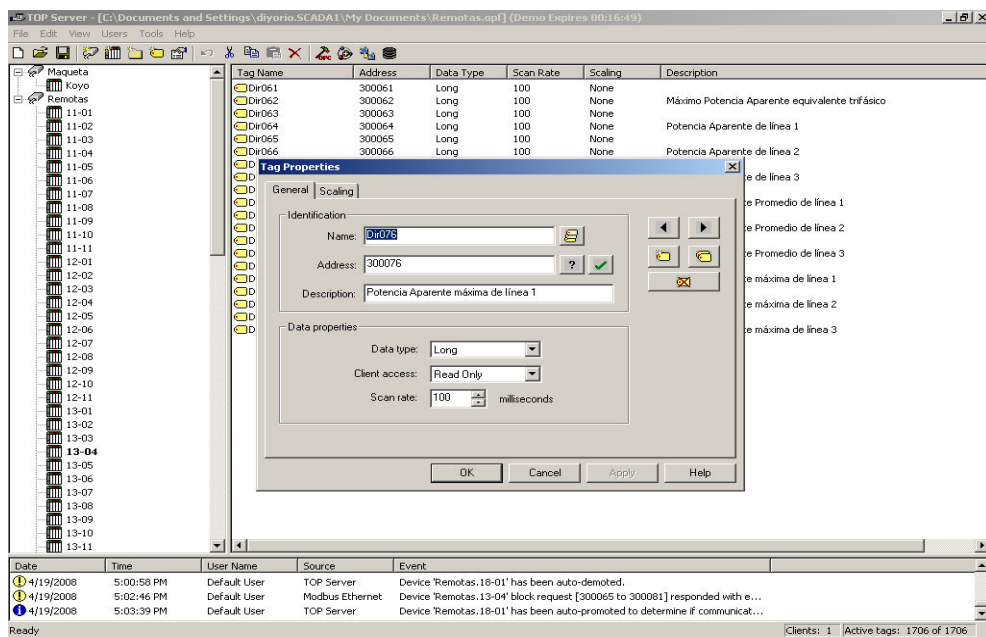


Figura 42. Tags para la comunicación con el Gateway y la configuración de uno.

3.1.2. Gateway Modbus TCP

Es el nombre dado al programa realizado por Luis Ortiz en su trabajo de grado para convertir los protocolos DUCBUS y Modbus RTU a Modbus sobre TCP [2]. Para la configuración de la comunicación con las remotas en este programa, sólo se tuvo que pulsar el botón de “Explorar Ahora” luego de instalar el programa, y esperar a que realizara la identificación de las remotas conectadas. Una vez realizado este procedimiento, se presionó el botón de operar para observar el estado de la comunicación. Algunas imágenes de este programa son mostradas en la Figura 43.

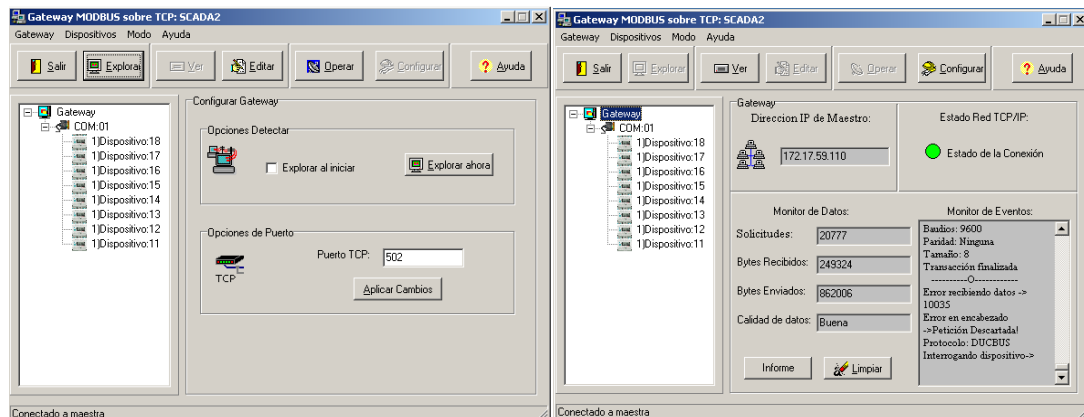


Figura 43. Dos vistas del Gateway Modbus TCP.

3.1.3. Servidor OPC del iFix

Este forma parte fundamental del software iFix y es utilizado para enviar el valor de las variables declaradas en la base de datos de iFix a cualquier otro programa o dispositivo comunicándose por OPC. En teoría este Servidor debía estar configurado inicialmente, pero resultó no ser así. El primer paso realizado fue abrir el SCU dentro del entorno del iFix y conectarse al cliente OPC local para la apertura del Power Tool. Una vez allí, se crea un servidor OPC con nombre iFix y tantos grupos como dispositivos configurados existiesen. Los datablocks se crean únicamente para el grupo Pruebas y el grupo de la Remota iA8.

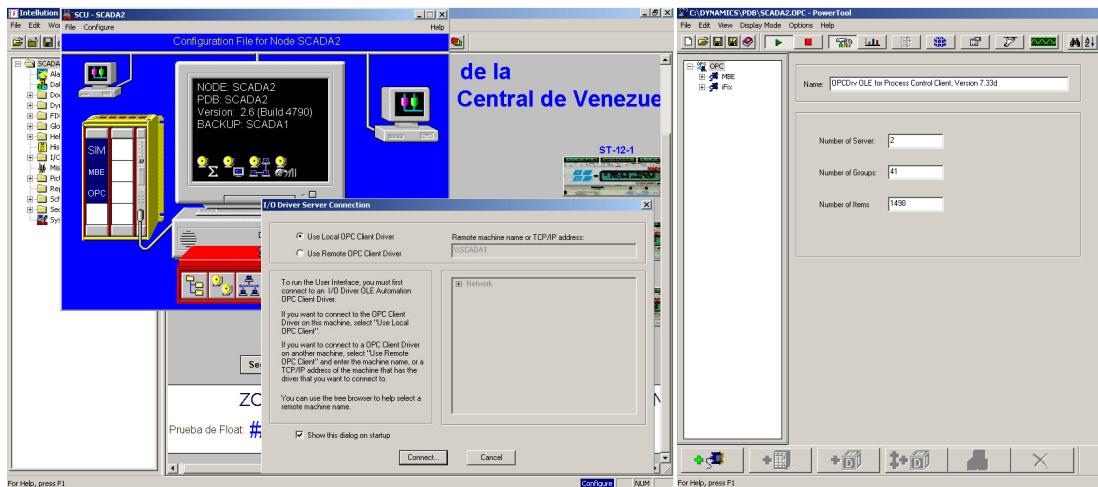


Figura 44. SCU del iFix y Power Tool con el servidor OPC.

Las imágenes de la Figura 44 muestran el SCU y el Power Tool con el Servidor OPC. Hasta este punto la configuración es la creada por defecto. Una vez que se ha llegado a la configuración de los Data Blocks en el Power Tool, es necesario ubicar el nombre del ítem en la base de datos del iFix. Para esto se presiona el botón “Browse Server” y se identifica el ítem a añadir. Esto se realiza para las dos variables de prueba creadas y para todos los elementos que componen el grupo iA8. Las configuraciones realizadas al DataBlock “Prueba2” y un DataBlock elegido al azar del grupo iA8 en el Power Tool, son mostradas en la Figura 45.

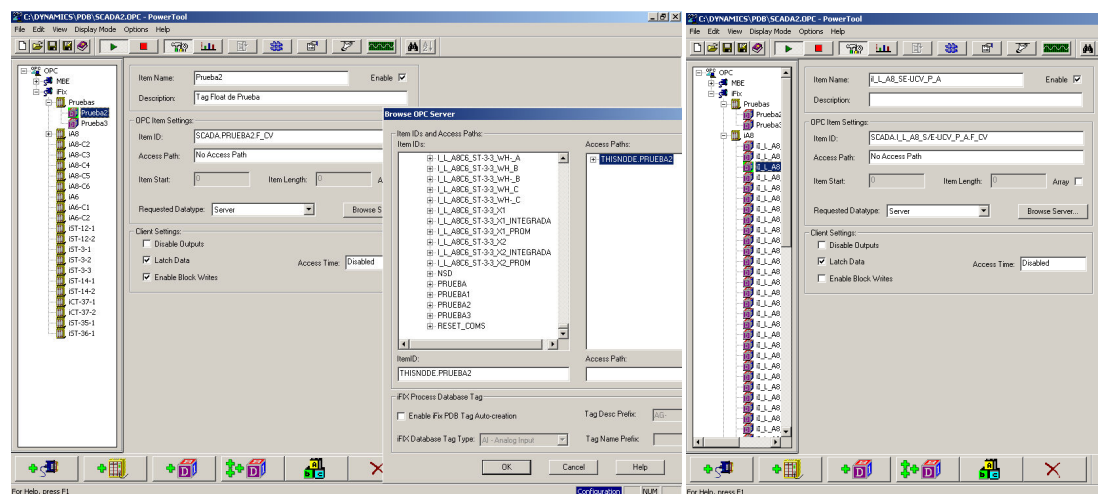


Figura 45. Configuración del DataBlock Prueba2 y de uno del grupo iA8.

3.1.4. FactorySuite Gateway

El FactorySuite Gateway es un DAServer perteneciente a los Device Integration Objects de Wonderware. La función primordial de este es convertir OPC a SuiteLink sin la necesidad de tener una plataforma desplegada localmente. Este se emplea en el proyecto como Gateway entre el OPC del iFix y el IAS de Wonderware. Su instalación se realiza con las opciones por defecto. La forma de acceder a este producto se efectúa a través del SMC, en su sección DAServer Manager, y una vez en el programa, se realiza la configuración del ArchestrA.FSGateway.1. Al darle click derecho sobre este nombre, se selecciona añadir un nodo OPC y se nombra iFix. En la Figura 46 se muestra la configuración del nodo OPC en el FSGateway.

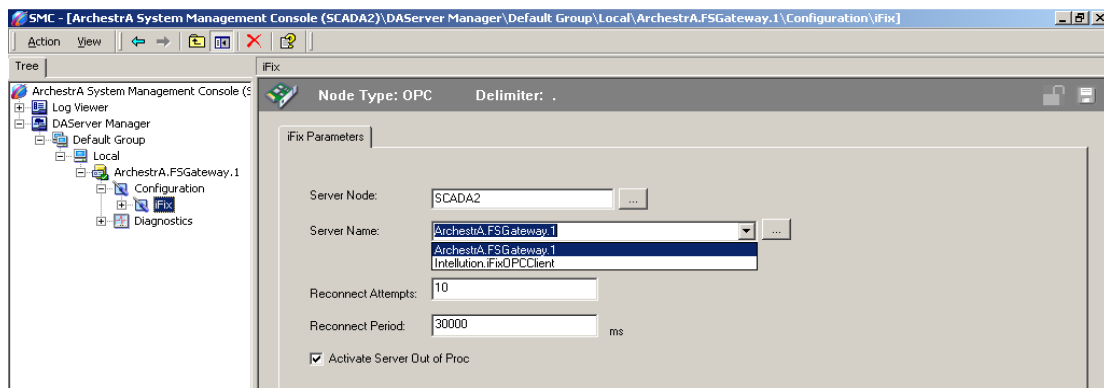


Figura 46. Configuración del Nodo OPC en el FSGateway.

Una vez que ha aparecido la ventana de configuración del nodo, se especifica como nodo del Servidor a “Scada2” y como Nombre del Servidor al OPC del iFix, dejando el resto de las opciones por defecto. Una vez finalizada la configuración del Nodo, se realiza la programación del grupo, para esto, se amplía la pestaña “iFix”, se selecciona la “Default” y se exploran los ítems del servidor OPC, añadiendo los deseados. Seguidamente, se inicia el FSGateway haciendo click sobre el botón verde en el menú superior del programa una vez seleccionado el servidor “ArchestrA.FSGateway.1”. En la Figura 47 se muestra la ventana de la pestaña “Default” con los Device Items añadidos y un diagnóstico de estos.

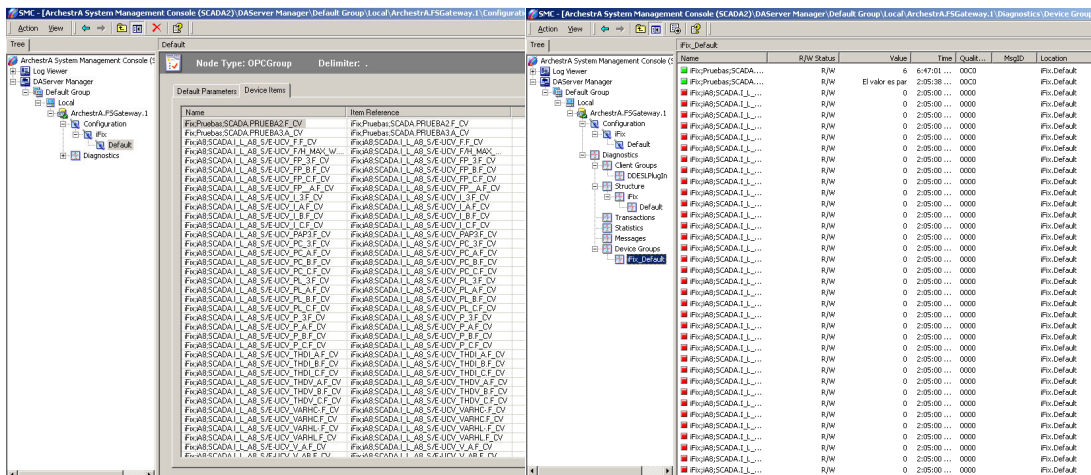


Figura 47. Device Items añadidos y un diagnóstico de la comunicación.

3.1.5. Programa Cerrar

Este programa fue realizado empleando Delphi 6, con la finalidad de cerrar un programa luego de un tiempo definido y abrirlo cierto tiempo después. Este programa se crea para cerrar y abrir el TOPServer una vez cumplido el tiempo de expiración del demo. También es empleado para reiniciar una vez al día el Gateway Modbus TCP, con la finalidad de evitar la congelación del mismo debido a un posible bug en ese programa. Para la configuración del programa cerrar, únicamente hace falta escribir la dirección del ejecutable y seleccionar los tiempos. Algunas imágenes de ejecución del programa son mostradas en la Figura 48 y el código de programación en el Anexo 5.

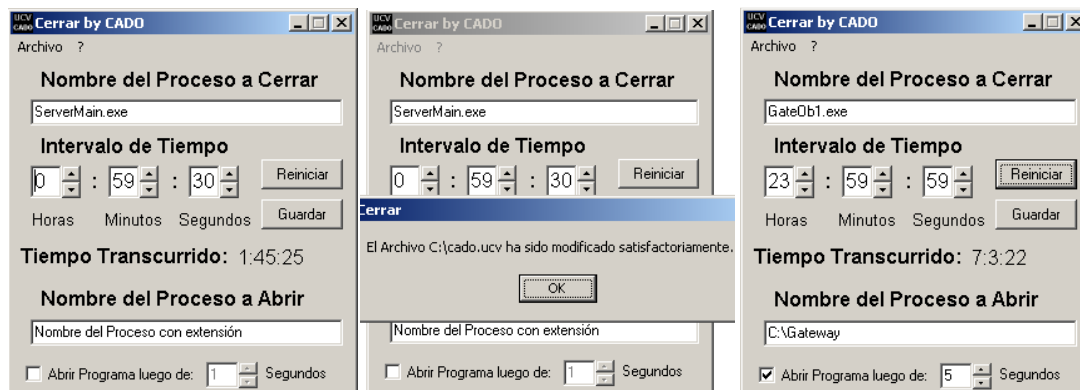


Figura 48. Programa Cerrar configurado para el TOPServer y el Gateway.

3.2. CONFIGURACIÓN DEL SOFTWARE PARA LA ARQUITECTURA DEL SCADA

3.2.1. Industrial Application Server 2.1 Patch 2

Es el programa base para la automatización basada en Galaxias. El paquete incluye la instalación del Bootstrap, el IDE y el Galaxy Repository, además del SMC y otros elementos adicionales. El Bootstrap, por ser el objeto fundamental de comunicación, es instalado en prácticamente todas las PCs del SCADA. El Galaxy Repository por el contrario, se instala en el computador que aloja el nodo de configuración de la base de datos. El SMC es instalado en todas las computadoras que poseen algún producto de Wonderware localmente, menos en las que cumplen la función de HMI. El SMC no solo se encuentra en el IAS, sino que es incluido en casi todos los paquetes de productos Wonderware.

El Integrated Development Environment (IDE) es el programa empleado en la configuración de todos los objetos de la Galaxia. Para su ejecución es requerido el Bootstrap y el MS SQL Server localmente. La instalación del MS SQL Server 2000 se realiza seleccionando la opción de instalar las herramientas de servidor y cliente, dejando las opciones por defecto y seleccionando el uso de una cuenta local en modo mixto. Posterior a ingresar la contraseña solicitada para el inicio de sesión del usuario sa (en todos los casos se utilizó “wwucv”), se selecciona el tipo de licencia “per seat” para un máximo de 10. Al finalizar, se actualiza a SQL 2005 con las opciones por defecto y utilizando las configuraciones anteriores en caso de ser solicitadas.

Una vez instalado el SQL 2005, se procede a la instalación del IAS. Para ello se selecciona el Bootstrap, el GR y/o el IDE dependiendo de la función del computador dentro del SCADA. Las opciones de instalación del IAS son las de por defecto, una vez seleccionados los componentes a añadir.

Al iniciar el IDE, es requerida la creación de una Galaxia. Para ello es necesario seleccionar como nodo GR al computador que posee la base de datos de configuración. En este proyecto la Galaxia es creada bajo el nombre de “Neutron” en SW0. Ya en el ambiente del IDE, se realiza la configuración y programación del SCADA en sí. El primer paso es crear tantas plataformas como sean necesarias para abarcar todas las computadoras donde sea requerida una. En este caso se añaden 6 plataformas, 2 (SW0 y Scada0) para alojar los Engines y a su vez todos los objetos de comunicación y configuración, 3 (Control1, Control7 y Control8) para realizar la comunicación con el HMI y una última (SW1) para la estación de ingeniería portátil. En la Figura 49 se muestra el inicio del IDE donde se conecta a la Galaxia y las plataformas definidas en la Galaxia Neutron.

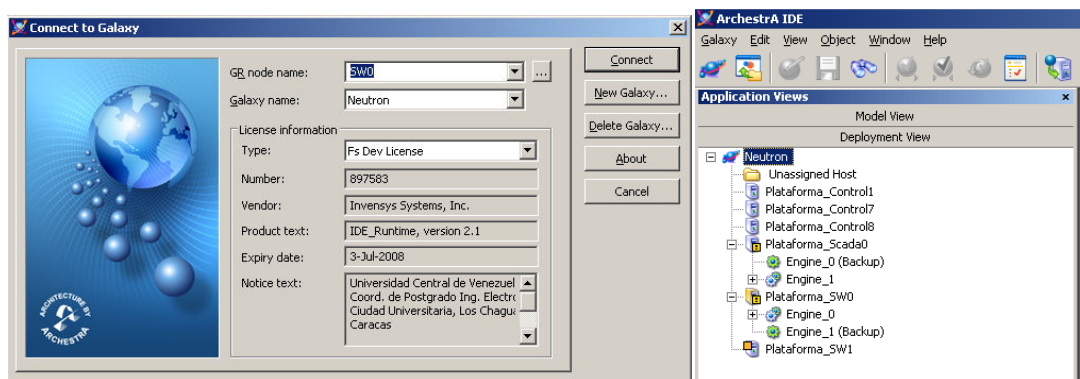


Figura 49. Inicio del IDE y vista final de las Plataformas de la Galaxia Neutron.

Antes de seguir con la descripción de las configuraciones realizadas, es necesario explicar los tres tipos de vista del entorno IDE. La primera llamada Model View, permite definir el modelo de planta del sistema dentro del ambiente. La segunda o Deployment View, permite seleccionar las Plataformas y Engines donde serán ejecutados los objetos de comunicación y otros, una vez que han sido desplegados. La última con nombre Derivation View, permite visualizar los objetos fuentes y sus derivados de una forma genealógica, con la idea de poder reconocer rápidamente cualquiera de los hijos y el tipo de objeto padre. Una imagen con los 3 tipos de vista para la Galaxia Neutron es mostrada en la Figura 50.

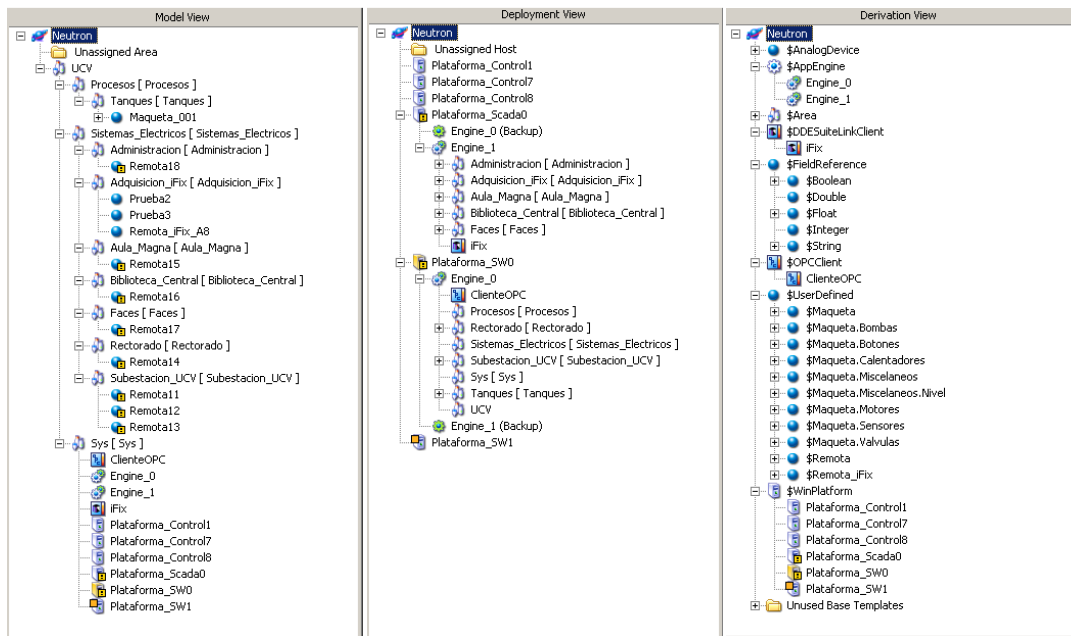


Figura 50. Model, Deployment y Derivation View de la Galaxia Neutron.

Entre los parámetros configurados de las Plataformas SW0 y Scada0 mostrados en la Figura 51 destacan, la configuración como proveedor de alarmas de InTouch, la dirección de red de la plataforma, la dirección IP de la tarjeta de red local para la configuración de la redundancia, la habilitación y configuración del nodo Historiador y la dirección del archivo para el almacenamiento temporal de los datos históricos. Aunque en la Figura son mostradas sólo las pestañas “General” y “Engine” de la Plataforma Scada0, las de la SW0 son configuradas de forma similar con la única diferencia de que la dirección IP de la tarjeta de red cambia por tratarse de un computador diferente.

Otros parámetros configurados incluyen, algunos elementos para habilitar alarmas de variables internas de las plataformas (Pestaña “Alarms”), por ejemplo, la pérdida de conexión con los Engines. También, es posible almacenar datos de variables de ejecución de las Plataformas para su análisis posterior (Pestaña “Platform History”), como por ejemplo, el porcentaje de RAM promedio utilizada de la PC.

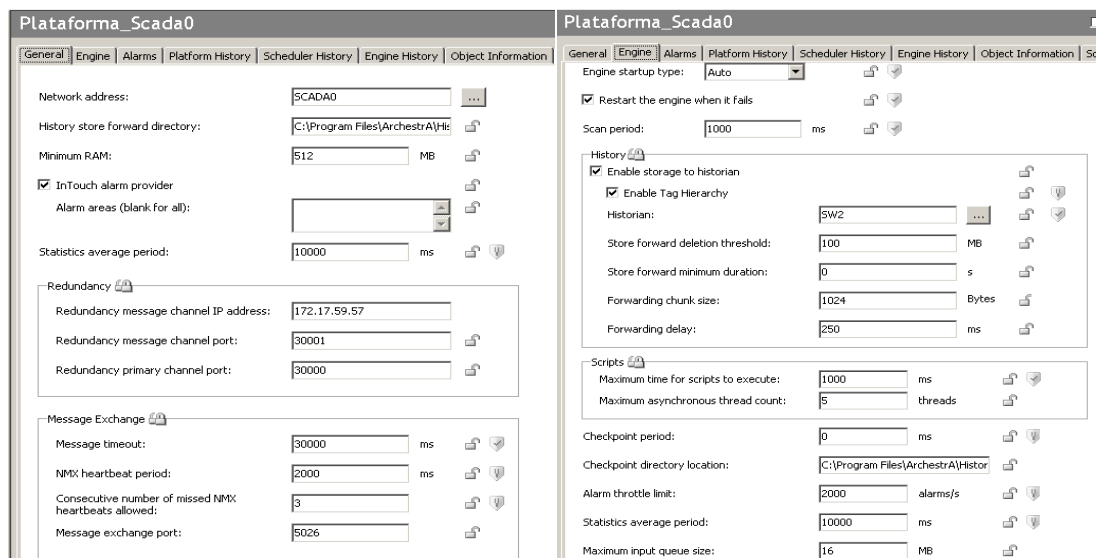


Figura 51. Pestaña “General” y “Engine” de la Plataforma Scada0.

El otro tipo de Plataforma empleada en el presente trabajo no contiene ninguna configuración adicional, siendo el único parámetro modificado la dirección del equipo. Las Plataformas Control1, Control7, Control8 y SW1 son de este tipo.

En la Galaxia se definen dos Engines similares, el primero se coloca sobre SW0 y el segundo en Scada0. Como ambos Engines son configurados redundantes, el IDE automáticamente crea un objeto para cada Engine que debe ser colocado sobre la Plataforma redundante. Entre las configuraciones importantes realizadas destacan, la habilitación de la redundancia, el reinicio automático del Engine al fallar y la conexión con el Histórico. Una imagen del Engine_0 es mostrada en la Figura 52.

Una de las ventajas que presenta Wonderware, es que las aplicaciones pueden ser distribuidas entre las dos plataformas que contienen los Engines y reducir el impacto del SCADA sobre el sistema. De esta forma no se habla que se tiene un sistema principal y otro redundante, sino más bien, que ambos son principales y redundantes a la vez. Así, si una plataforma presenta un problema, la otra acoge los objetos que se encontraban ejecutándose en la que presenta la falla y los coloca en ejecución conjuntamente con los que presentaba anteriormente.

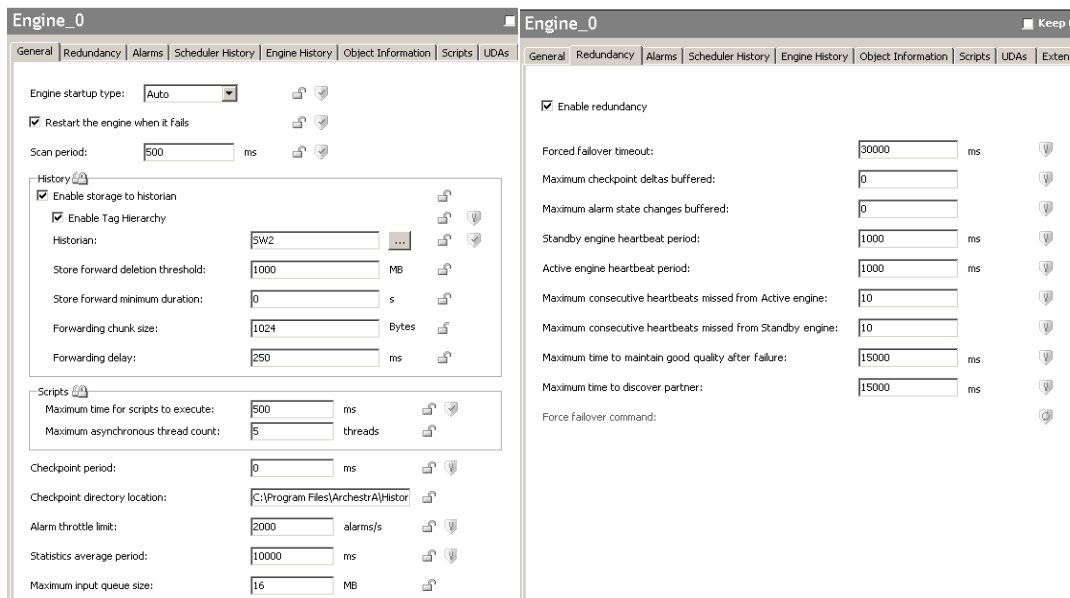


Figura 52. Pestañas “General” y “Redundancy” del Engine_0.

Los siguientes objetos definidos en el IDE son los de Comunicaciones. El primero de ellos es un Cliente OPC para la comunicación con el TOPServer, configurado como se muestra en la Figura 53. Los parámetros importantes incluyen el nodo donde se encuentra ubicado el cliente, el nombre de éste, los grupos de escaneo, las variables a adquirir y un script añadido para la reconexión automática.

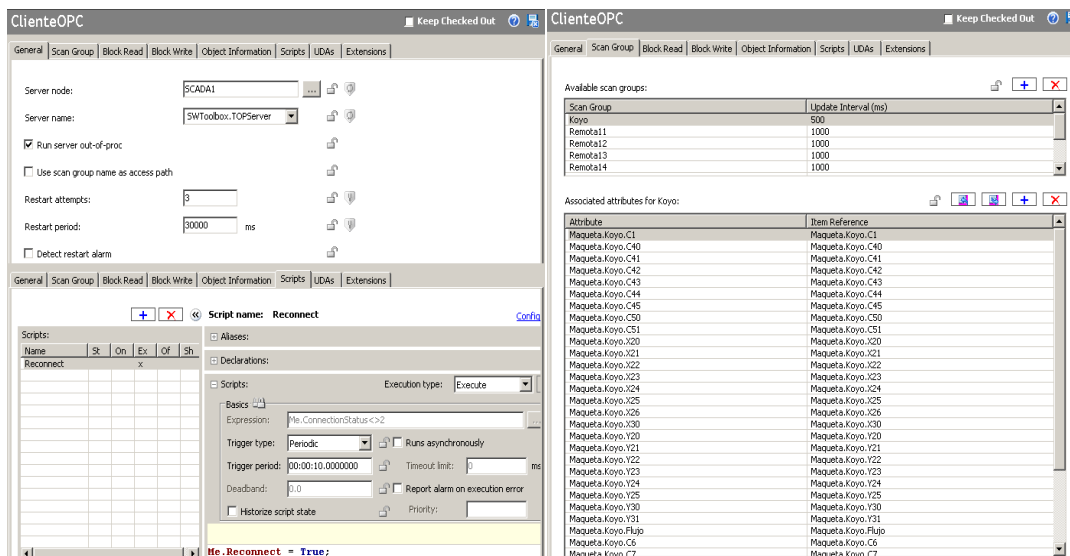


Figura 53. Pestañas “General”, “Scripts” y “Scan Group” del Cliente OPC.

El otro objeto se trata de un cliente FactorySuite Gateway para la comunicación con el iFix. Los parámetros modificados se muestran en la Figura 54, e incluyen el nodo del equipo con el Servidor, el grupo y las variables a adquirir, y un script de reconexión automática similar al del cliente OPC.

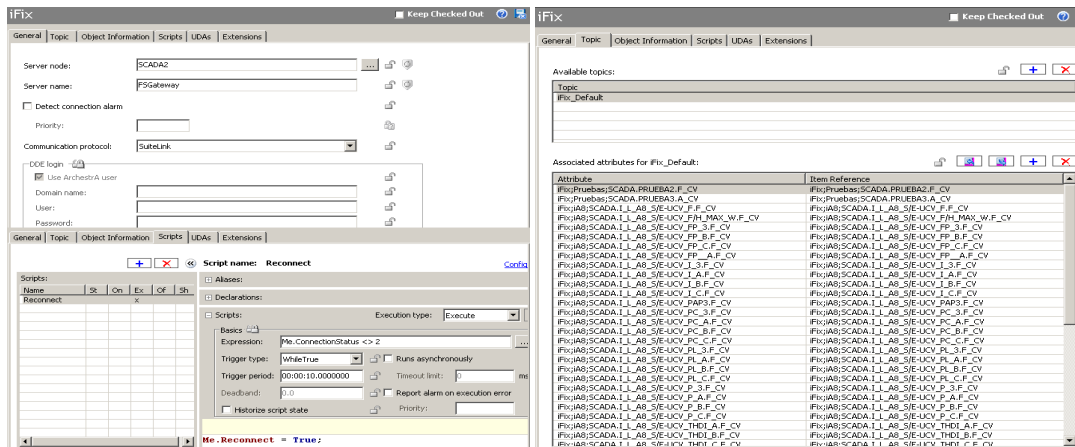


Figura 54. Pestañas “General”, “Scripts” y “Topic” del FactorySuite Gateway.

Una vez creados los clientes de comunicación, se procede a añadir los objetos para el procesamiento individual de cada uno de los datos. Esos objetos pueden ser derivados de los ya definidos incluidos en el producto, o pueden ser creados a través del User Define. Para mostrar la configuración de los dos tipos, se implementó en la Galaxia un objeto Remota tipo User Define, para la adquisición de todos los parámetros relacionados con cada una de las RTU Ducati, y un objeto basado en los ya incluidos, para la manipulación de las variables de la Maqueta de Procesos.

El objeto Remota fue el que requirió de mayor configuración. En este se tuvo que añadir cada una de las variables y, añadirle la dirección para apuntar al Cliente OPC, además de escalarlo y almacenarlo dependiendo del caso (ver Figura 55). Las alarmas no fueron configuradas en este caso, debido al desconocimiento de los valores límites para cada una de las variables, la inexistencia de un operador constante del SCADA y la imposibilidad de realizar un control asociado a la medición. De todas formas, es posible la configuración de estas en un futuro.

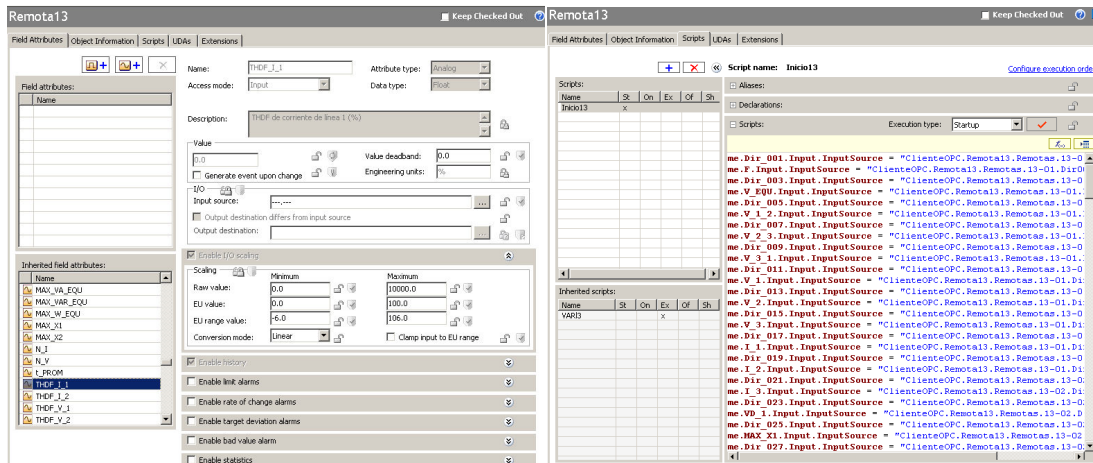


Figura 55. Pestañas “Field Attributes” y “Scripts” del Objeto Remota13.

El objeto Maqueta contiene la configuración de las variables y áreas de la Maqueta de Procesos. Como este está formado por instancias y plantillas derivadas de los objetos incluidos en el programa, sólo es necesario configurar la dirección de la variable y seleccionar si se desea escalar, almacenar y generar eventos o alarmas dependiendo del caso. En la Figura 56 se muestran algunas pestañas para la configuración del objeto “Temperatura_RTD” y su ubicación dentro de “Maqueta”.

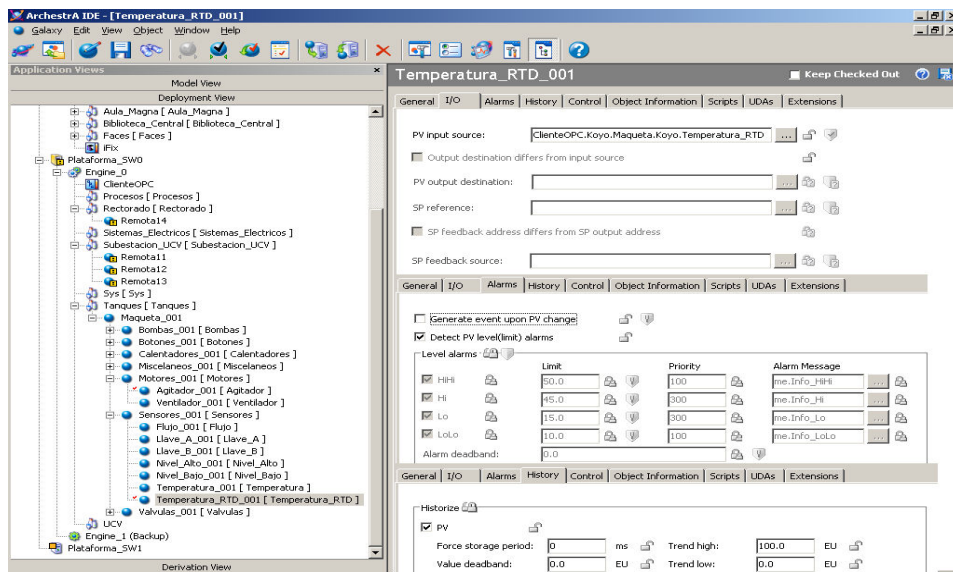


Figura 56. Objeto “Maqueta” y algunas pestañas de “Temperatura_RTD”.

Es importante destacar que la seguridad instalada y la que mejor se adapta al SCADA es la del tipo “Galaxy” [35], debido a que centraliza todo el sistema, y los usuarios y grupos son modificados desde el IDE. Por último, es posible analizar el estado de cada uno de los objetos definidos en la Galaxia utilizando el Object Viewer (ver Figura 57). Este es un programa incluido en el IAS, se emplea para verificar el estado de los dispositivos, modificar y observar el valor de cada una de las variables.

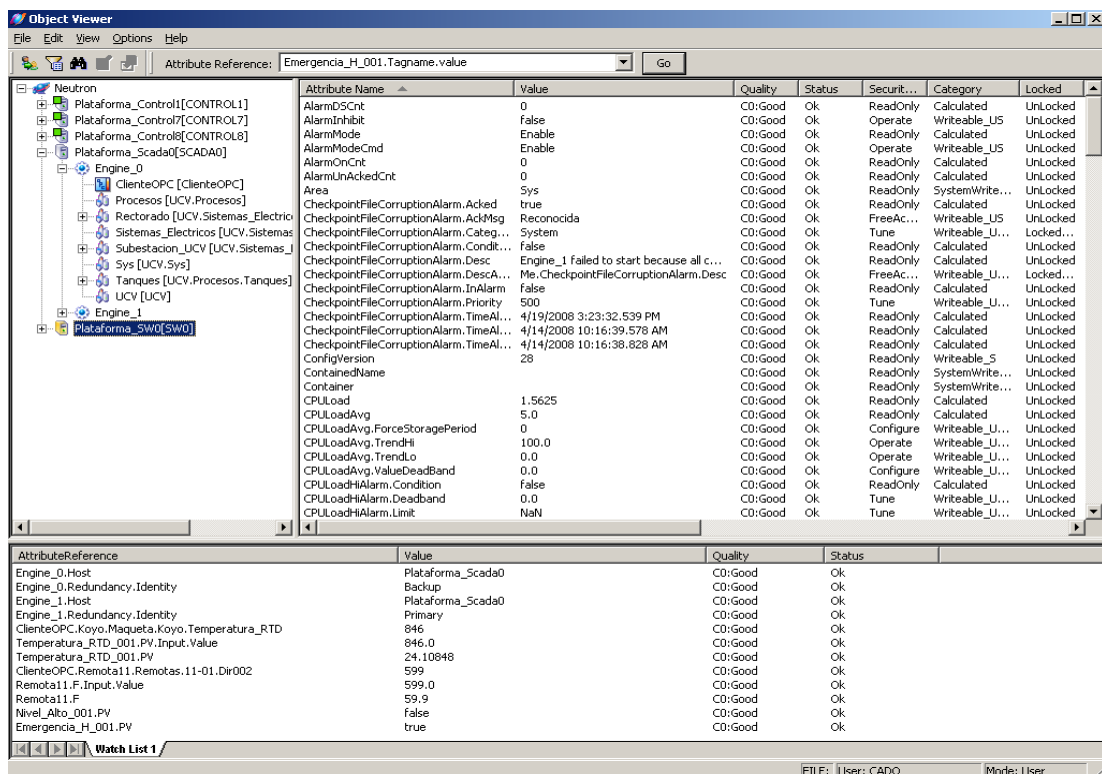


Figura 57. Una lista de observación de la Galaxia definida en el Object Viewer.

3.2.2. Industrial SQL Server 9.0 y Alarm DBLogger Manager

El Industrial SQL Server es el Historiador de datos de Wonderware. La instalación de este producto tiene como requisito el MS SQL Server. El computador seleccionado para desempeñar la función de Historiador fue el SW2. En este, fue instalado el MS SQL Server 2000 y actualizado a la versión 2005 siguiendo los mismos pasos explicados en la instalación del IAS de la sección anterior.

Una vez actualizado el MS SQL Server, se procedió a la Instalación del InSQL Server 9.0 seleccionando todas las opciones por defecto. La instalación de este producto incluye el SMC para la configuración y modificación del InSQL. Al abrir la consola, es de importancia observar que todos los módulos se encuentren en ejecución. En caso de que alguno de estos no haya iniciado, se deben revisar los parámetros de configuración mostrados en la Figura 58 (en especial el de AutoStart) y en el peor de los casos, reinstalar el producto.

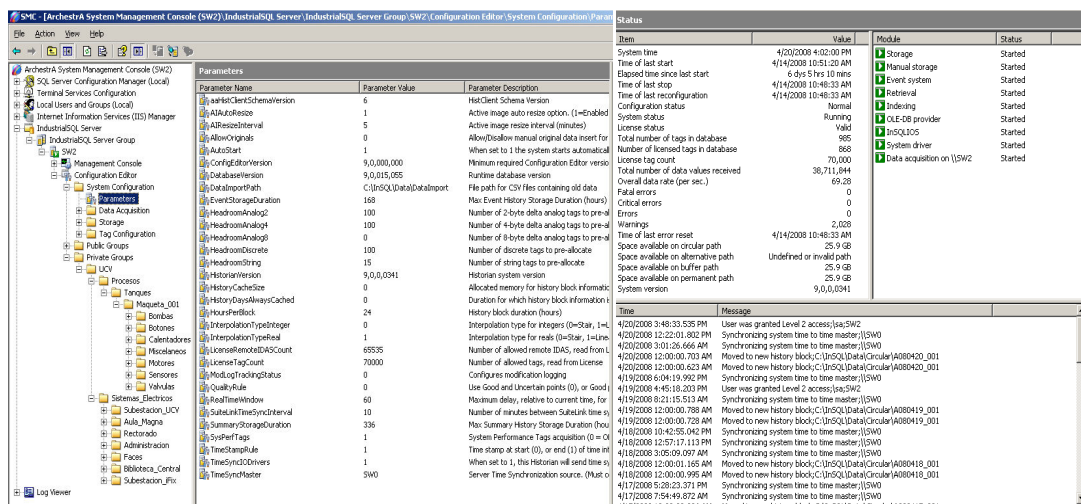


Figura 58. Parámetros de Configuración y Estado de los módulos del InSQL.

Este producto no requiere de mucha configuración, de hecho, el único parámetro de importancia que se debe modificar es el AutoStart, para que el InSQL ejecute sus servicios de forma automática al iniciar Windows. Como el nodo SW2 se encuentra configurado desde el IDE, este carga automáticamente la configuración de todos los objetos del programa, siendo el único requisito el inicio previo del nodo Historiador antes que los AOS. Una de las características de este producto es que acepta la creación de Grupos Privados, lo que permite al usuario configurar el modelo de planta y cualquier otra clasificación de interés como un grupo, para que pueda ser cargado y observado más fácilmente desde el entorno Trend del ActiveFactory. En la Figura 59 se muestra un grupo privado creado con todas las variables del área “UCV” y algunos parámetros de la consola.

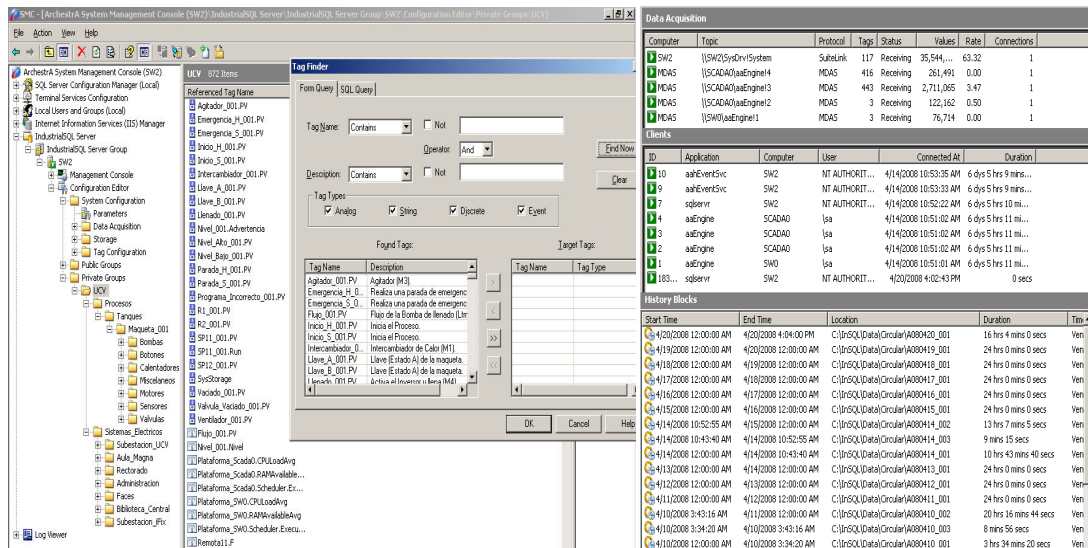


Figura 59. Grupo privado creado y algunos parámetros de la Consola.

El AlarmDBLogger Manager es un complemento de InTouch que permite el almacenamiento de las alarmas, el cual es recomendado instalarlo en el mismo computador que aloja al histórico de datos. Para su configuración, se crea la base de datos por defecto seleccionando modo consolidado y colocando los Query de ambas plataformas que alojan los objetos del SCADA, tal como se muestra en la Figura 60. Para evitar el agotamiento de la memoria en un futuro, se configura el Alarm Purge/Archive para eliminar los datos almacenados anteriores a 15 días.

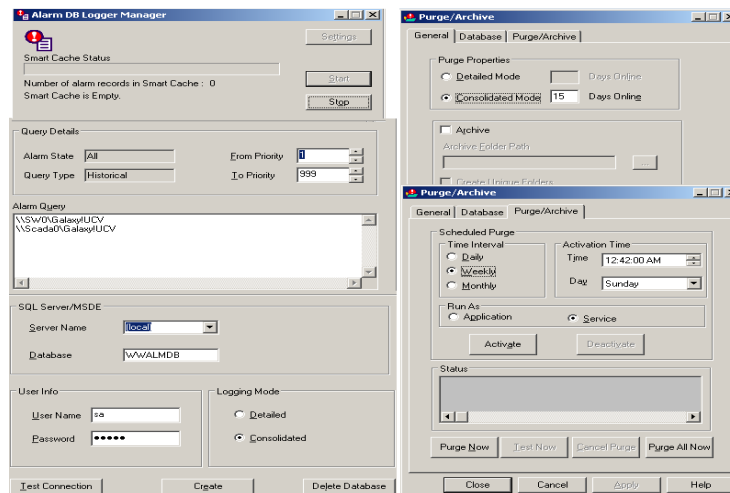


Figura 60. Algunas vistas del AlarmDBLogger Manager y el Purge/Archive.

3.3. CONFIGURACIÓN DEL SOFTWARE PARA EL HMI

3.3.1. ActiveFactory 9.2

Este es un programa que permite la visualización y el acceso a la data almacenada por el InSQL. Su instalación se realiza utilizando las opciones por defecto en los nodos del HMI y del Historiador. Es empleado por InTouch como consultor en tiempo real de los datos almacenados, mientras que es utilizado en el nodo del Historiador para realizar la configuración local del mismo.

Se ejecuta el Trend realizando la configuración de conexión con el Servidor InSQL de SW2. Para ello se ingresó como nombre de usuario “sa” y contraseña “wwucv”, no requiriéndose de ningún otro tipo de configuración en el ambiente del programa. Si se desea graficar los datos almacenados de alguna variable, sólo hace falta arrastrarla desde el “Tag Picker” (ubicado a la izquierda del Trend) hasta el grid (posicionado en la parte inferior del programa) y seleccionar el Checkbox. En caso de querer realizar un cambio de escala, mover la base de tiempo, añadir anotaciones, ver la data adquirida en tiempo real u otros, el programa ofrece algunos instrumentos en su barra de Herramientas para lograr dichas acciones. En la Figura 61 se muestra la configuración de la conexión con el InSQL y una vista del Trend en funcionamiento.

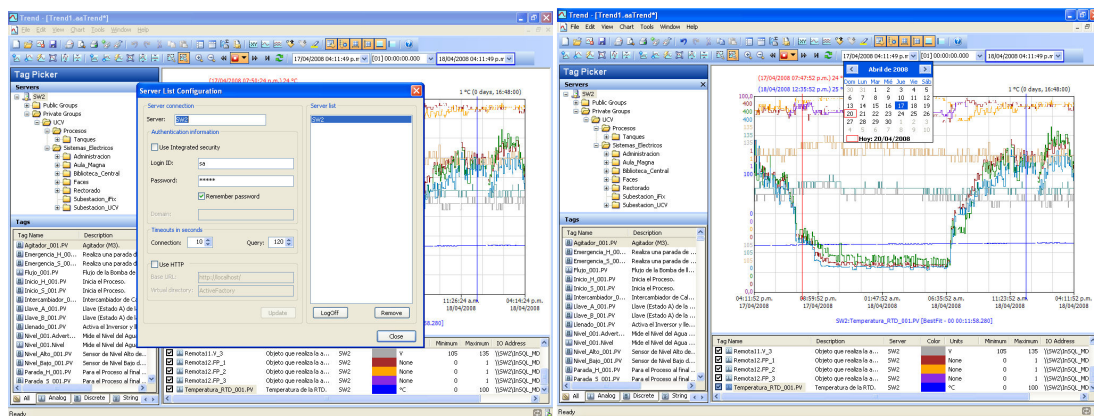


Figura 61. Conexión con el InSQL y una vista del Trend en funcionamiento.

3.3.2. InTouch 9.5 SP1

InTouch es el programa empleado para realizar la programación y ejecución del HMI de Wonderware. Para su Instalación, sólo hace falta seguir las opciones por defecto del producto. El paquete de InTouch 9.5 trae consigo el WindowMaker, utilizado para realizar la programación y configuración de las pantallas del HMI, el WindowViewer, empleado para ejecutar el HMI, y otros programas que realizan funciones asociadas o complementan el producto.

El WindowMaker contiene una gran cantidad de gráficos e imágenes de dispositivos industriales y misceláneos, empleados para mejorar la interfaz visual de los HMI. Además, ofrece una gran cantidad de instrucciones de programación que facilitan la creación de la aplicación y la programación de las animaciones. Aunque resulta posible la comunicación directa con Servidores para la adquisición de data desde InTouch, en este trabajo no se utiliza por tratarse de un SCADA distribuido orientado al IAS. Este programa es instalado en todas las PCs identificadas como HMI, Estación de Ingeniería y en el Nodo Historiador. El HMI configurado consta de una plantilla base que contiene 7 botones, 6 de los cuales abren otras plantillas y uno para cerrar el programa, además de animaciones e imágenes que mejoran la estética de este. Los botones permiten la apertura de las plantillas “Inicio”, “Maqueta”, “Remotas”, “Alarmas”, “Historiad.” y “Com. iFix”.

En la plantilla “Base”, se encuentran botones para el inicio y cierre de sesión, además de mostrar el estado de cada uno de los equipos que conforman el SCADA. El Cierre e Inicio de Sesión fue programado basándose en las instrucciones “LogOff” y “AttempInvisibleLogon”. Por tratarse de una seguridad del tipo “Galaxy”, no fue necesaria la configuración de usuarios ni grupos desde InTouch. También existe la posibilidad de configurar un cierre automático de sesión luego de algún tiempo de inactividad, pero no fue considerado.

La sección de la Plantilla “Inicio” que muestra el estado de los equipos del SCADA, fue programada a través de la revisión (realizada cada minuto) del estado de las plataformas y los Servidores de comunicación. En la Figura 62 se muestra la plantilla base y la de inicio en el InTouch WindowViewer.



Figura 62. Plantilla Base y Plantilla de Inicio en ejecución.

La plantilla “Maqueta” mostrada en la Figura 63, contiene principalmente tres secciones. La primera muestra un esquema animado en tiempo real de la Maqueta, la segunda contiene botones para realizar el control remoto de esta y la tercera permite el reconocimiento y visualización de las alarmas del sistema en tiempo real.

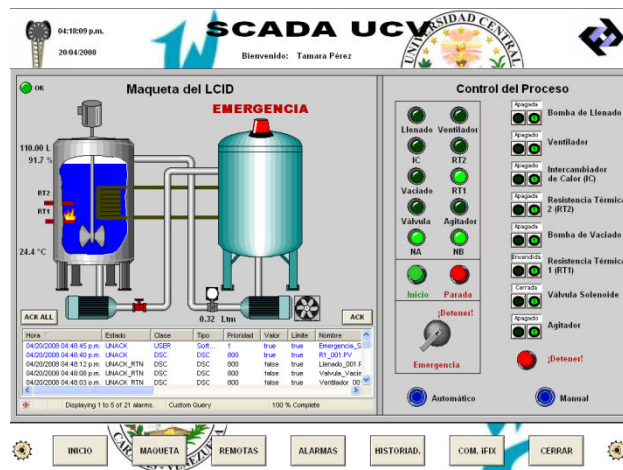


Figura 63. Plantilla de la Maqueta de Procesos en ejecución.

El esquema animado fue realizado en su gran mayoría con elementos del Symbol Factory y animados según su estado, por el contrario, en la sección correspondiente al control de la maqueta fueron utilizados los componentes básicos de animación del InTouch. La visualización de las alarmas se realiza con el objeto “AlarmViewerCtrl”, mientras tanto, el reconocimiento se puede efectuar haciendo click derecho dentro del objeto y seleccionando la opción “Ack” y “AckAll”, o a través de dos botones programados con los controles ActiveX de este. La Figura 64 muestra algunos elementos básicos del InTouch y el Symbol Factory.

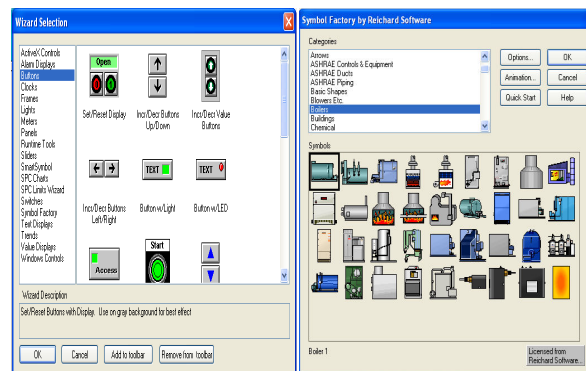


Figura 64. Algunos elementos básicos de InTouch y Symbol Factory.

La plantilla “Remotas” mostrada en la Figura 65, contiene la visualización de los parámetros eléctricos adquiridos por las remotas en tiempo real. Aunque la forma de presentación empleada en esta plantilla se encuentra prácticamente en desuso, fue realizada de esta forma para mostrar todo el abanico de posibilidades del InTouch. La forma actual de realizar la visualización en tiempo real es con el ActiveFactory o con el Real-Time Trend. Los controles de escala y movimiento en el tiempo se obtienen separando el objeto “16-Pen Trend Scooter” y posicionándolo adecuadamente. Para seleccionar las variables a graficar en cada uno de los Trends son empleados unos “listbox”. Al ser seleccionado algún ítem de estos y ser presionado el botón de aceptar, se realiza la búsqueda de la variable en el “listbox” a través de la instrucción “wcFindItem” para luego convertirla a un string que será empleado para apuntar la variable indirecta (graficada por el trend) al objeto correspondiente de la galaxia.

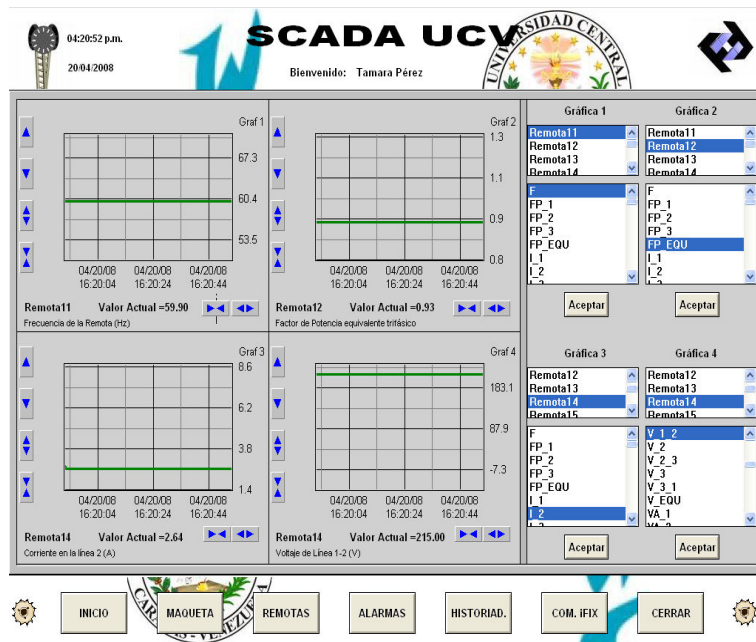


Figura 65. Plantilla “Remotas” en ejecución.

Las plantillas “Historiad.” y “Alarmas” contienen elementos relacionados con el histórico de datos y el de alarmas. Los objetos utilizados en la plantilla “Alarmas”, realizan el acceso a la base de datos de las alarmas con la función de graficar las 20 alarmas de mayor ocurrencia y mostrar todas las ocurridas en un intervalo de tiempo determinado. El objeto que muestra de forma gráfica las 20 alarmas de mayor ocurrencia se llama “AlarmPareto”, mientras que, el “AlarmDBViewCtrl” expone la lista de las alarmas. Para configurar ambos objetos es necesario ingresar el nodo del servidor que aloja el AlarmDBLogger, el usuario y la contraseña de acceso.

La plantilla “Historiad.” no es más que un Trend añadido como control ActiveX para ser ejecutado dentro del ambiente del InTouch. Este no necesita configuración alguna más que el nodo donde se encuentra el historiador de datos, el nombre de inicio de sesión y la contraseña de acceso al mismo. En la Figura 66 se muestran las plantillas “Historiad.” y “Alarmas” en ejecución:

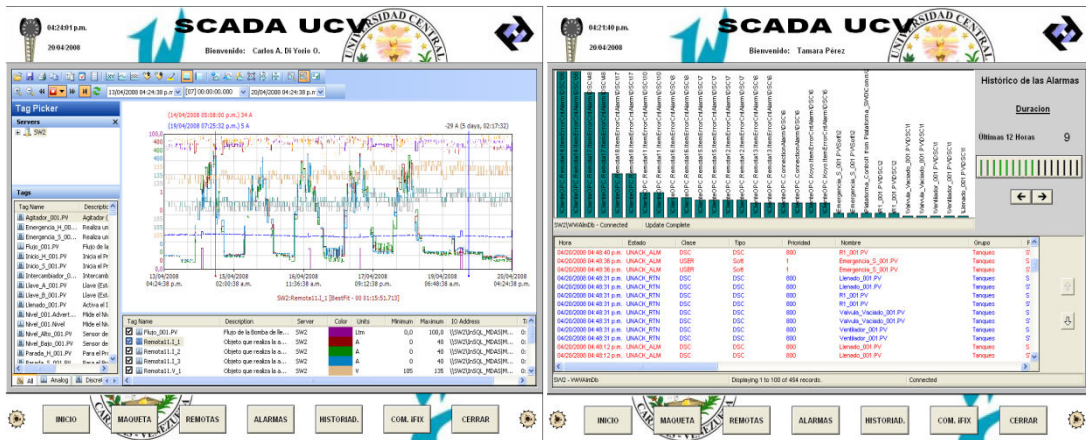


Figura 66. Plantilla Historiador y Alarma en ejecución.

La plantilla “iFix” de la Figura 67, muestra los datos de los parámetros eléctricos adquiridos en iFix. Esta se encuentra compuesta de una tabla que varía dinámicamente con el cambio de los valores para asemejarse al SCADA anterior, y de una zona de comunicaciones en donde se adquiere el valor de las variables de prueba declaradas en la base de datos de iFix. El valor de los elementos de la tabla nunca cambia de cero debido a que el iFix no logra adquirir los datos de las remotas. Para la programación de esta plantilla sólo fue necesario añadir una gran cantidad de cuadros de textos y configurarlos adecuadamente para mostrar los datos.



Figura 67. Plantilla de Comunicación con iFix y menú de un cuadro de Texto.

CAPITULO IV

VALIDACIÓN Y PRUEBAS DEL SCADA

Para verificar el funcionamiento del SCADA distribuido, comprobar el control implementado y estudiar los posibles casos de fallas del sistema, se realizaron una serie de pruebas, las cuales, las más resaltantes son detalladas a continuación:

a) Inicio y Parada por Hardware de la Maqueta de Procesos: Esta prueba consistió en la manipulación de los botones de Inicio (verde), Parada (rojo) y de Emergencia (rojo grueso) ubicados en la parte frontal del cajetín de la maqueta.

Se pulsó el botón de inicio y se dejó funcionando la maqueta de procesos durante varios ciclos, luego fue presionado el botón de parada notándose que el proceso finalizaba al realizar el vaciado del tanque hasta el límite inferior, tal cual como fue configurado. Posteriormente, se volvió a iniciar la maqueta y se presionó el botón de emergencia en cada una de las fases del proceso, lográndose detener instantáneamente todos los equipos activados y reiniciando el proceso a su inicio.

Además, fue comprobada la reacción del HMI ante cada uno de los eventos por hardware de la maqueta. De esta forma, al ser presionado el botón de inicio, parada y/o emergencia, se nota la aparición de una línea indicando esta acción en la tabla de alarmas de la plantilla Maqueta, y se comprobó el almacenamiento de esta en la tabla de histórico de alarmas de la Plantilla Alarmas. De igual forma, fueron comprobadas que cada una de las animaciones realizadas al ocurrir la activación de un proceso o una alarma, correspondían a las programadas en el HMI. Una imagen del Tanque de la Maqueta de Procesos, del Cajetín y de la plantilla Maqueta del HMI durante la fase de llenado es mostrada en la Figura 68.

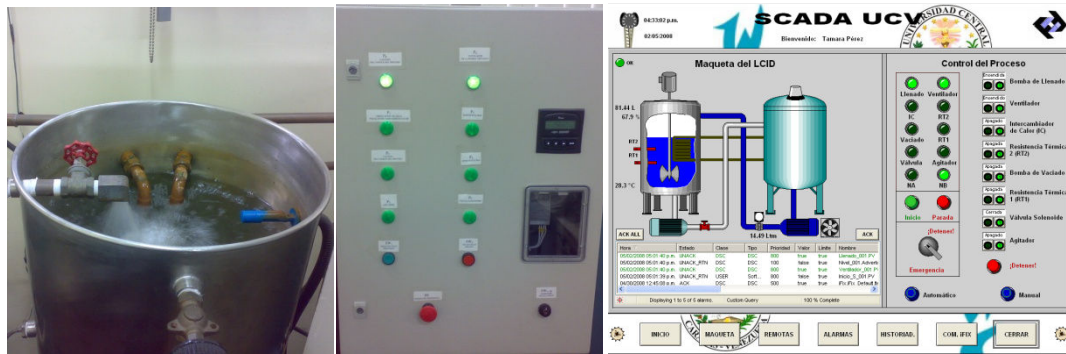


Figura 68. Tanque, Cajetín y Plantilla Maqueta durante la fase de llenado.

b) Manipulación por Software de la Maqueta de Procesos: Esta prueba consistió en la activación de los botones para la manipulación manual y automática de la maqueta de procesos a través del HMI.

En primer lugar, se realizaron las mismas pruebas del caso anterior, pero para los botones de inicio, parada y emergencia por software, resultando en el funcionamiento correcto de los mismos. Luego, se realizaron verificaciones del funcionamiento de los botones de manipulación manual de la maqueta, para esto, se activaron y desactivaron desde el HMI cada uno de los dispositivos controlados, observando su funcionamiento en la maqueta y la representación en el HMI. Un caso de control manual de la maqueta por el HMI es mostrado en la Figura 69.

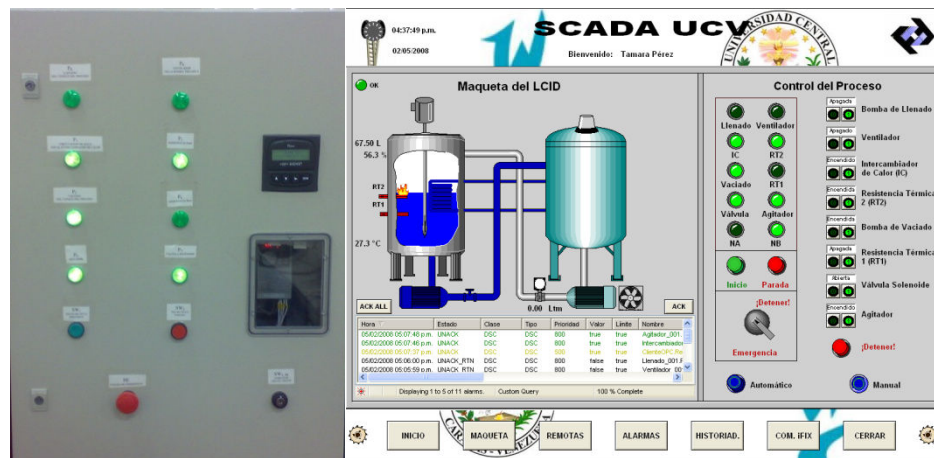


Figura 69. Cajetín de la Maqueta y Control manual por el HMI.

Como no todos los dispositivos se pueden activar de forma independiente, se realizaron todas las posibles pruebas de generación de Advertencias y alarmas al operador a través del HMI. Las pruebas realizadas fueron:

- La activación de la bomba de vaciado sin haber ocurrido la apertura de la válvula solenoide.
- La activación de la bomba de llenado sin encenderse el ventilador de esta.
- La activación de la bomba de vaciado por debajo del nivel inferior del tanque.
- La activación de la bomba de llenado por encima del nivel alto del tanque
- La activación de las resistencias térmicas cuando la temperatura es superior a 45 °C.
- La falla de los sensores de lleno y vacío.

Todas las pruebas generaron la visualización de la palabra “Advertencia” en el HMI y la respectiva activación de la alarma correspondiente en la Tabla de Alarmas de la Plantilla Maqueta. Una imagen de la plantilla Maqueta al ser activada la bomba de vaciado sin abrirse la válvula solenoide y otra de la activación de la bomba de llenado por encima del nivel superior del tanque son mostradas en la Figura 70.

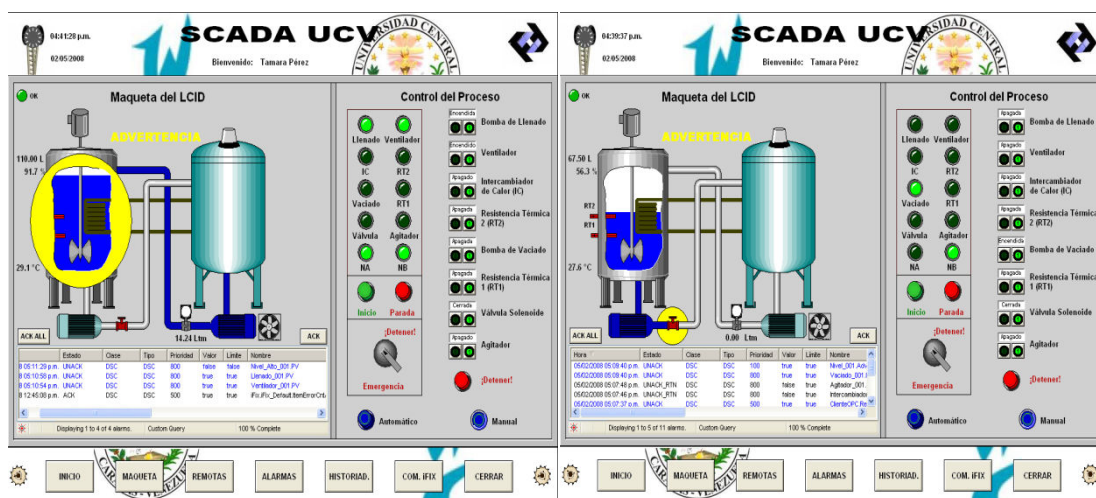


Figura 70. Imágenes de la Plantilla Maqueta en dos casos de Advertencia.

c) **Almacenamiento de la data en el InSQL:** Otra de las validaciones del SCADA consistió en acceder a la data almacenada de una semana por el InSQL desde el HMI. En este caso se realizó la conexión con el nodo SW2 desde el objeto Trend del ActiveFactory, incluido en la Plantilla Historiador del InTouch. El resultado fue la adquisición correcta de la data almacenada, la cual fue graficada y mostrada al usuario. Dicha gráfica es mostrada en la Figura 71.

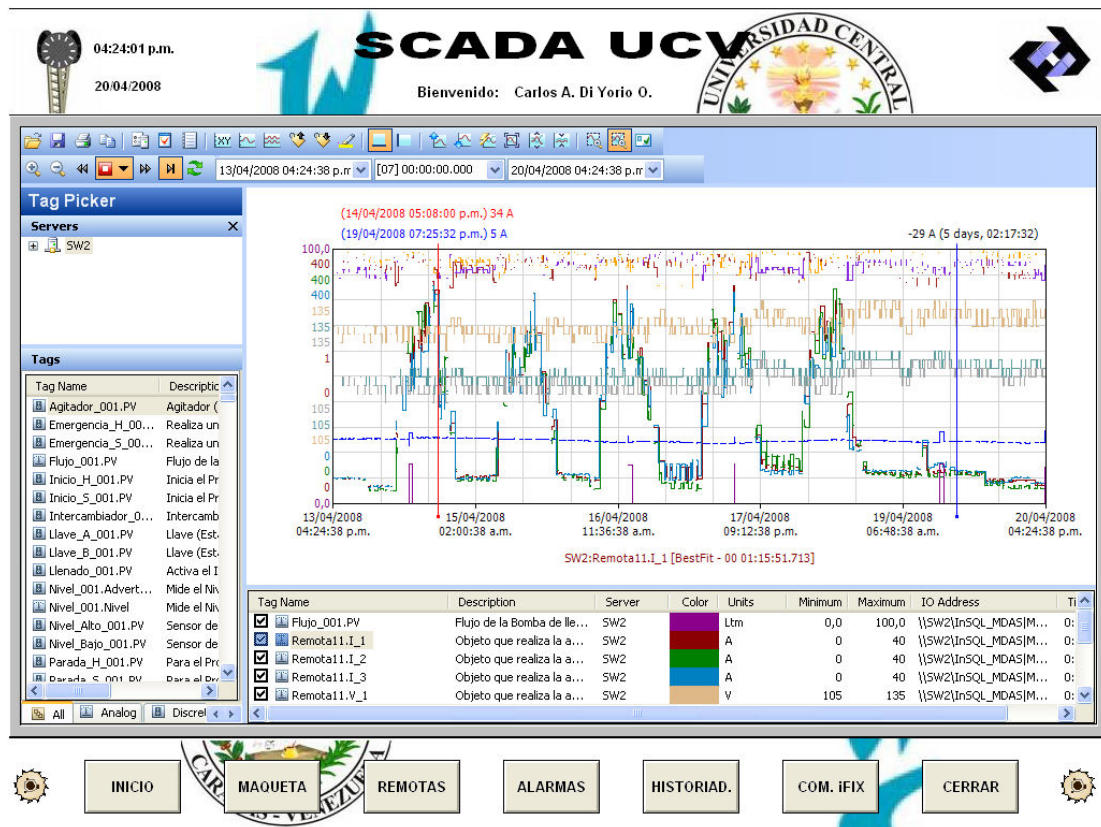


Figura 71. Gráfica de una semana de la data almacenada.

d) **Pérdidas de Comunicación:** Las pérdidas de comunicación pueden ser ocasionadas por múltiples causas, entre estas destacan los fallos de energía, el colgado de las PCs, la falla de los dispositivos, la desconexión del cable de comunicaciones, las fallas en la red de la escuela, entre otras.

Todos los posibles casos fueron probados, algunos de estos generados de forma intencional y otros no intencionales, con resultados variados en cuanto a la forma y procedimientos de recuperación de la comunicación del SCADA. A continuación son descritas las posibles causas de pérdidas en la comunicación, la reacción del SCADA y los métodos de recuperación manual en caso de ser necesario:

- ***Falla General de Comunicación del Sistema y Casos Generales.***

Reconocimiento: Se muestra el recuadro que bordea las imágenes de la mayoría de los computadores y dispositivos de la plantilla Inicio en color Rojo, y las plantillas muestran valores desactualizados, incoherentes o no muestran algún valor.

Causas: Falla general de energía, concentradores del LCID se encuentran desconectados, concentrador del Laboratorio de Máquinas no está energizado, problemas con el concentrador del Laboratorio de Desarrollo de la Escuela de Ingeniería Eléctrica o con otros relacionados con la red interna de la EIE.

Recuperación: En caso de una falla eléctrica, como los equipos no poseen inicio automático de sesión de Windows, se deben iniciar manualmente. Una vez iniciados, la mayoría de los recuadros que bordean los computadores en la plantilla de Inicio deben cambiar de color rojo a verde, en caso de no ser así, se debe verificar que el inicio de los PCs se haya realizado correctamente, y por último, reiniciar el computador.

Un caso comúnmente presentado es que al ocurrir una falla eléctrica, los Engines no inician correctamente. En caso de que se presente este tipo de falla, se debe reiniciar los computadores Scada0 y SW0 (en ese orden). Si esto no resuelve el problema, se debe abrir el Plattform Manager desde el SMC de Scada0 o SW0, y verificar que las Plataformas y los Engines se hayan iniciado correctamente (ejecutándolos en caso de que esto no haya ocurrido). En última instancia, se debe realizar el redesplicue de la Galaxia para la resolución de este problema.

- ***Pérdida de Comunicación con la Maqueta.***

Reconocimiento: Al intentar realizar acciones de control sobre la maqueta esta no responde, la plantilla Maqueta muestra valores no adecuados o no actualizados, se muestra el recuadro que bordea la imagen de la maqueta en la plantilla Inicio de color rojo o se observa una línea en la tabla de alarmas indicando la pérdida de comunicación con la Maqueta.

Causas: El TOPServer se encuentra en su período de reinicio, ha sido desconectado el Cable Ethernet del PLC, la alimentación de la maqueta se encuentra desconectada, el interruptor principal de la maqueta está apagado, el PC Scada1 se encuentra apagado o no ha iniciado sesión, alguno de los Engines no se encuentran en ejecución o existe un mal funcionamiento de alguna de las PCs.

Recuperación: En caso de que el TOPServer se encuentre en su período de reinicio, lo único que se debe hacer es esperar unos segundos para el restablecimiento de la comunicación. Si esto no resuelve el problema, se debe revisar la conexión de red al PLC, energizar la maqueta o realizar el proceso de Recuperación de Falla general de comunicación del Sistema descrito anteriormente.

- ***Pérdida de Comunicación con las Remotas.***

Reconocimiento: La plantilla Remotas muestra valores no adecuados o no actualizados (ver Figura 72), se muestra el recuadro que bordea la imagen de la Remota Ducati en la plantilla Inicio de color Rojo o se observa una línea en la tabla de alarmas indicando la pérdida de comunicación con las Remotas.

Causas: El TOPServer se encuentra en su período de reinicio, el PC Scada1 y/o Scada2 se encuentra apagado o no ha iniciado sesión, el convertidor RS232/485 se encuentra desconectado, alguno de los Engines no se encuentran en ejecución o existe un mal funcionamiento de alguna de las PCs.

Recuperación: En caso de que el TOPServer se encuentre en su período de reinicio, lo único que se debe hacer es esperar unos segundos para el restablecimiento de la comunicación. Se debe revisar el estado del Gateway en Scada2, y reiniciarlo conjuntamente con el programa Cerrar, en caso de que este se encuentre inactivo.

Si el recuadro que bordea la imagen de la remota Ducati en la plantilla Inicio se encuentra en Amarillo, no se debe de realizar acción alguna, ya que esto significa que existen algunos fallos en la comunicación con el OPC o con las remotas que se solucionan automáticamente. Si ninguno de estos casos resuelve el problema, se debe realizar en última instancia el proceso de Recuperación por Falla general de comunicación del Sistema descrito anteriormente.

- ***Pérdida de Comunicación con el Historiador.***

Reconocimiento: Al realizar el inicio de sesión desde el Trend es imposible conectarse con SW2, se muestra el recuadro que bordea la imagen del computador del Histórico en la plantilla Inicio de color Rojo o se observa una línea en la tabla de alarmas indicando la pérdida de comunicación con el Historiador.

Causas: El computador del Historiador se encuentra apagado o no ha iniciado sesión en Windows, no han iniciado los servicios del PC correctamente, el PC no responde o existen problemas en la red.

Recuperación: La solución en casi todos los casos incluye el reinicio inicial del computador para intentar resolver el problema. Si esto no resuelve el inconveniente, se debe revisar que todos los servicios del InSQL hayan iniciado su operación (que su estado se encuentre en verde) y que no existan fallas en la red.

- ***Pérdida de Comunicación con alguno de los computadores restantes.***

Reconocimiento: La plantilla de Comunicación con iFix no muestra valor alguno (ver Figura 72), se muestra el recuadro que bordea la imagen de alguno de los

computadores restantes en la plantilla Inicio de color Rojo o se observa una línea en la tabla de alarmas indicando la pérdida de comunicación con alguna plataforma.

Causas: El computador Scada2 o alguno de los computadores restantes se encuentran apagados o no han iniciado sesión en Windows, el iFix del computador Scada2 no se ha abierto, el FactorySuite Gateway no ha iniciado correctamente en Scada2, el PC no responde o existen problemas en la red.

Recuperación: La solución para la mayoría de los casos incluye el reinicio inicial del computador que presenta la falla. Si esto no resuelve el inconveniente, se debe revisar que los programas asociados a la función de cada computador se encuentren abiertos y en ejecución. En caso de falla con la comunicación con el iFix, se debe revisar que el programa iFix se encuentre abierto en Scada2. Para el resto de los computadores, si no es posible resolver los inconvenientes, se debe recurrir a la solución expuesta en caso de Falla general de las comunicaciones.



Figura 72. Plantilla iFix y Remotas al ocurrir fallas en la comunicación.

Una imagen de la Plantilla Inicio al ocurrir una Falla de Comunicación General y otra de la pérdida de comunicación con la Maqueta y las Remotas producto del reinicio del TOPServer son mostradas en la Figura 73.



Figura 73. Plantilla Inicio al ocurrir algunas fallas de comunicación.

e) **Programa Incorrecto en el PLC de la Maqueta:** Otra de las pruebas realizadas consistió en reescribir el programa del PLC de la Maqueta con uno completamente diferente. En este caso se logró observar el cambio de color, de verde a rojo, del indicador ubicado en la parte superior izquierda de la plantilla Maqueta, y la aparición de la frase “Programa Incorrecto” al lado del indicador. Este problema es resuelto al reescribir el programa del PLC de la Maqueta y el realizado para este trabajo. Cuando se están ejecutando otras tareas sobre la maqueta, que ameriten la modificación del programa raíz del PLC, el SCADA arrojará el mismo resultado descrito anteriormente. Una imagen que describe el caso anterior es mostrada en la Figura 74.

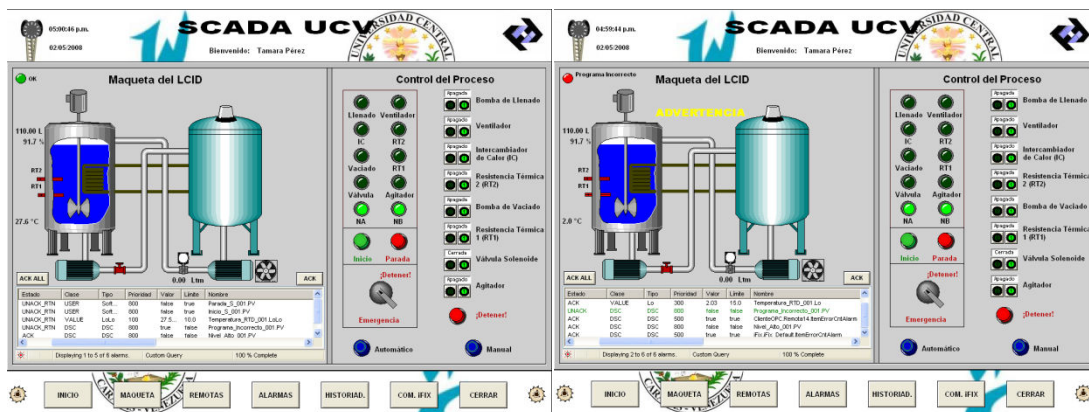


Figura 74. Programa Correcto e Incorrecto en el PLC.

CONCLUSIONES

El sistema SCADA implementado en el presente trabajo establece adecuadamente la comunicación con el PLC de la maqueta de control, las remotas Ducati y el SCADA anterior bajo iFix, almacena los datos y alarmas de las variables previamente seleccionadas y muestra a través de un HMI los datos, las alarmas y eventos de manera sencilla y amigable al usuario.

El sistema SCADA implementado representa una mejora significativa respecto al SCADA anterior, ya que, además de realizar la adquisición de los parámetros eléctricos de las remotas, incluye una maqueta de procesos que contiene instrumentos para la medición de variables no eléctricas, tales como el nivel, el caudal y la temperatura; y permite un control remoto sobre el PLC de dicha maqueta, abriendo la posibilidad de apertura de prácticas de laboratorio a distancia.

El número de computadores empleados en la implementación del SCADA, a pesar de que superan el máximo recomendado inicialmente, ninguno de estos fueron subutilizados y se adecuaron eficientemente a su funcionamiento dentro del sistema.

El sistema SCADA basado en el IAS mejora significativamente el uso de los computadores que funcionan como AOS mientras no se ha generado el Failover, ya que, la configuración de la redundancia de Wonderware permite la repartición de los objetos en ejecución entre un computador y su par redundante (Balance de cargas).

El tipo de seguridad utilizado en el Sistema SCADA implementado, constituye un modelo de seguridad a nivel de Galaxia que permite la adición y eliminación de usuarios, roles y grupos del sistema, de forma transparente y segura para el resto de los computadores que lo conforman.

La data histórica almacenada por el SCADA del presente trabajo permite realizar un análisis de consumo de energía y de los parámetros eléctricos de la escuela de Ingeniería Eléctrica. Dicha información se podría utilizar para la optimización y mejoramiento de la calidad de servicio, y el ahorro de energía.

Los productos para SCADA y HMI orientado a objetos resultan ventajosos desde todo punto de vista respecto a los programas tradicionales basados en Tags. Estos permiten la organización de las variables en forma jerárquica, hacen prácticamente independiente la definición de los objetos respecto de los gráficos y animaciones del HMI, facilitan la programación en términos generales y reducen considerablemente el tiempo total de implementación del SCADA.

El modelo de planta constituye uno de los primeros pasos a seguir para la realización de un SCADA y proporciona una vista general de la estructura y objetos que conforman una automatización, que se traducen en una progresiva disminución de los tiempos de programación, configuración e implementación total del SCADA.

Finalmente, el trabajo desarrollado constituye el primer paso en la implementación física de un Sistema de Control Distribuido en la Escuela de Ingeniería Eléctrica, así como también, el primer paso en el desarrollo de tesis con este producto (Wonderware).

RECOMENDACIONES

Migrar todos los PCs que fueron definidos y empleados en este trabajo a un sistema compuesto de cuatro computadores, entre los cuales, dos de estos sean utilizados como AOS, I/O Server y HMI, uno sea empleado como historiador y Servidor Web, y el último posea la Estación de Ingeniería y el Configuration Database (Galaxy Repository). De igual forma, es recomendable la puesta en marcha del sistema actual sobre iFix si se desea la conservación del mismo, y la migración y reconfiguración de todos los elementos Wonderware sobre los PCs antes descritos.

Los objetos tipo User Define deben utilizarse para definir objetos de pocas variables, debido a que se hacen muy pesados para su apertura y modificación posterior, y se dificulta el entendimiento de su modelo. Para definir sistemas u objetos de mayor tamaño, es recomendable agrupar objetos separados, tal como se realizó con el objeto Maqueta de este trabajo.

Los programas de Wonderware deben estar instalados en computadores que presenten sólo el Sistema Operativo, para evitar las posibles fallas que se puedan generar por incompatibilidad con otros componentes.

A pesar de que los Engines son ejecutados en dos plataformas distintas, la identificada bajo el nombre SW0 siempre presenta inconvenientes de comunicación ocasionando por fallas en la Red, carga excesiva del computador, pérdidas de comunicación momentáneas con el OPC y otros, que generan el mal funcionamiento del Engine y a su vez del SCADA. Para evitar esto es recomendable colocar en ejecución ambos Engines sobre la plataforma Scada0, y dejar SW0 como redundante en caso de falla de este último.

Igualmente, por limitaciones en las prestaciones de los computadores Control1, Control7 y Control8 y al congestionamiento de la red de la escuela, no se realizó la carga de todos los objetos al inicio de la ejecución del Window Viewer, aunque en un futuro si se migra el SCADA, debería modificarse el HMI realizado bajo InTouch para que lo realice, y así, evitar el tiempo de espera ocasionado por la apertura de estos una vez solicitados por el operador. De esta forma la aplicación correría más rápidamente mientras es utilizada, aunque tardaría una gran cantidad de tiempo en su apertura.

Los computadores que conforman los nodos del AOS deberían poseer dos tarjetas de red con un cable cruzado entre ellas y estar físicamente cerca para lograr un funcionamiento adecuado de la redundancia. En este caso, esto no se realizó debido a la imposibilidad de poseer dos computadores en el LRD para ser configurados con esta función.

Como el inicio de sesión de las PCs del SCADA no se realiza automáticamente, es imprescindible estar en constante revisión del estado de los computadores debido a que pueden haberse reiniciado o apagado por presentar un corte de energía, lo que ocasionaría el estancamiento del mismo en la pantalla de bienvenida del Windows y evitaría el arranque de los programas del SCADA. La recomendación en este caso vendría siendo el cambio de inicio de sesión de forma manual a automática.

Debido a que el laboratorio de redes de distribución sólo posee 3 puntos para la conexión a la red, es necesario la modificación y recanalización de este. La propuesta en este caso, vendría siendo la de traer un solo cable Ethernet desde el concentrador del Laboratorio de Máquinas para conectar el Switch recientemente adquirido y recanalizar a partir de este. Esto mejoraría la cantidad de errores de comunicación y reduciría el ancho de banda utilizado.

Debido a que Wonderware posee entre sus herramientas un programa para la realización y modelaje de protocolos llamado RPM Toolkit, resulta conveniente la implementación de un Gateway para el protocolo DUCBUS utilizando esta herramienta. Esto mejoraría significativamente la comunicación con las remotas y eliminaría la necesidad del TOPServer debida a la variante Modbus utilizada en el Gateway Modbus TCP.

Resulta imprescindible la instalación de un aire acondicionado en el laboratorio de redes de distribución para la ventilación de las máquinas. De igual forma, es recomendable la adquisición de un par trenzado blindado que se adapte a las especificaciones de una red 485, y de un convertidor 232/485 industrial.

Aunque el objeto Remota se encuentra funcional y se ejecuta correctamente, este presenta aproximadamente 200 variables internas que dificultan su modificación y comprensión. Éste se encuentra definido de esta forma con propósitos demostrativos, pero en una actualización posterior del SCADA, este objeto debería ser sustituido por una plantilla similar a como se hizo con la maqueta.

REFERENCIAS BIBLIOGRÁFICAS

[1] Gudiño, Jorge. Especificación para el Sistema SCADA del sistema de distribución Eléctrico de la Universidad Central de Venezuela./ Gudiño Jorge (Tesis).-- Caracas: Universidad Central de Venezuela, 2000.

[2] Ortiz, Luis. Gateway para la Integración de RTU con protocolos propietarios a una red de adquisición de datos sobre iFix./ Ortiz Luis (Tesis).-- Caracas: Universidad Central de Venezuela, 2003.

[3] Salcedo, Nelson. Integración del Sistema de Supervisión del Laboratorio de SCADA de la Escuela de Ingeniería Eléctrica de la UCV./ Salcedo Nelson (Tesis).-- Caracas: Universidad Central de Venezuela, 2006.

[4] Automatización Industrial – Wikipedia, la enciclopedia libre [En línea]. <http://es.wikipedia.org/wiki/Automatizaci%C3%B3n_industrial#Historia_de_la_Automatizaci.C3.B3n> [Consulta: Septiembre 2007].

[5] Joseph Marie Jacquard – Wikipédia [En línea]. <http://fr.wikipedia.org/wiki/Joseph_Marie_Jacquard> [Consulta: Septiembre 2007].

[6] Spinning Jenny – Wikipedia, la enciclopedia libre [En línea]. <http://es.wikipedia.org/wiki/Spinning_Jenny> [Consulta: Septiembre 2007].

[7] PLC [En línea]. <<http://www.sc.ehu.es/sbweb/webcentro/automatica/WebCQMHI/PAGINA%20PRINCIPAL/PLC/plc.htm>> [Consulta: Marzo 2008].

[8] PLCTutor.com – What is a PLC? [En línea]. <<http://www.plctutor.com/index.php?page=whatisapl>> [Consulta: Marzo 2008].

[9] Controlador lógico programable – Wikipedia [En línea]. <http://es.wikipedia.org/wiki/Controlador_l%C3%B3gico_programable> [Consulta: Septiembre 2007].

[10] Chapter 2. History: A brief History of User Interfaces [En línea]. <<http://www.ca-tb.org/~esr/writings/taouu/html/ch02.html>> [Consulta: Septiembre 2007].

[11] Georgiev04d.pdf [En línea]. <<http://www.kar.elf.stuba.sk/~georgiev/data/georgiev04d.pdf>> [Consulta: Noviembre 2007].

[12] Jonas, Berge. Introduction to Fieldbuses for Process Control (Libro Web), USA: ISA, 2002.

[13] Modbus_Application_Protocol_V1_1b.pdf [En línea]. <http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf> [Consulta: Enero 2008].

[14] PB_FFIELDBUS Issue 2.pdf [En línea]. <[http://library.abb.com/GLOBAL/SCOT/SCOT203.nsf/VerityDisplay/54154F7F109CEB8C80256B590039651D/\\$File/PB_FFIELDBUS%20Issue%202.pdf](http://library.abb.com/GLOBAL/SCOT/SCOT203.nsf/VerityDisplay/54154F7F109CEB8C80256B590039651D/$File/PB_FFIELDBUS%20Issue%202.pdf)> [Consulta: Septiembre 2007].

[15] Redes1c.pdf [En línea]. <<http://www.fi.uba.ar/materias/6629/redes1c.pdf>> [Consulta: Noviembre 2007].

[16] Modelo OSI – Wikipedia, la enciclopedia libre [En línea]. <http://es.wikipedia.org/wiki/Modelo_OSI> [Consulta: Octubre 2007].

[17] OSI REFERENCE MODEL [En línea]. <http://www.comsoc.org/livepubs/50_journals/pdf/RightsManagement_eid=136833.pdf> [Consulta: Octubre 2007].

[18] Redes2byn.pdf [En línea]. <<http://www.fi.uba.ar/materias/6629/redes2byn.pdf>> [Consulta: Noviembre 2007].

[19] Modbus [En línea]. <<http://en.wikipedia.org/wiki/Modbus>> [Consulta: Enero 2008].

[20] Modbus-IDA Specifications and Implementation Guides [En línea]. <<http://www.modbus.org/specs.php>> [Consulta: Enero 2008].

[21] OPC Data Access – Wikipedia, the free encyclopedia [En línea]. <http://en.wikipedia.org/wiki/OPC_Data_Access> [Consulta: Marzo 2008].

[22] Opc_tutorial_printable.pdf [En línea]. <http://www.matrikonopc.com/training/opc-multimedia-tutorial/opc_tutorial_printable_version.pdf> [Consulta: Marzo 2008].

[23] DirectNET User Manual.pdf [En línea]. <http://www.hosteng.com/Ethernet_SDK/DirectNET%20User%20Manual.pdf> [Consulta: Marzo 2008].

[24] Hxecomm.pdf [En línea]. <<http://automationdirect.com/static/manuals/hxecomm/hxecomm.pdf>> [Consulta: Marzo 2008].

[25] Koyo PLC's K-Sequence Protocol – Forums.MrPLC.com [En línea]. <<http://forums.mrplc.com/index.php?showtopic=4750&mode=linearplus>> [Consulta: Marzo 2008].

[26] Ethernet Interface Specification v3.pdf [En línea]. <http://www.hosteng.com/Ethernet_SDK/Ethernet%20Interface%20Specification%20v3.pdf> [Consulta: Marzo 2008].

[27] Manual de uso de la Ducati Mach 30 / Mach 20.

[28] D2uservol1.pdf [En línea]. <<http://www.automationdirect.com/static/manuals/d2user/d2uservol1.pdf>> [Consulta: Marzo 2008].

[29] Programmable Automation Controller [En línea]. <http://en.wikipedia.org/wiki/Programmable_automation_controller> [Consulta: Marzo 2008].

[30] Programmable Automation Controllers (PACs) – National Instruments [En línea]. <<http://www.ni.com/pac/esa/>> [Consulta: Marzo 2008].

[31] PAC Combina la Robustez del PLC y la Funcionalidad de la PC – OfficeInfo – Américas – National Instruments [En línea]. <<http://digital.ni.com/worldwide/latam.nsf/web/all/DBCC2AA73D23AABB86256F49006EFD2B>> [Consulta: Marzo 2008].

[32] PACs para Control Industrial, el Futuro del Control – Developer Zone – National Instruments [En línea]. <<http://zone.ni.com/devzone/cda/tut/p/id/5948>> [Consulta: Marzo 2008].

[33] Remote Terminal Unit – Wikipedia, the free encyclopedia [En línea]. <http://en.wikipedia.org/wiki/Remote_Terminal_Unit> [Consulta: Octubre 2007].

[34] The Industrial Ethernet Book – Articles: Staying in control part 1: Look behind ArcestrA [En línea]. <<http://ethernet.industrial-networking.com/articles/articledisplay.asp?id=176>> [Consulta: Marzo 2008].

[35] Wonderware. Training Manual – Wonderware System Platform Part 1 Course. 2007.

[36] Giordano, Rafael. Diseño e Implementación de un Sistema de Control Didáctico./ Giordano Rafael (Tesis).-- Caracas: Universidad Central de Venezuela, 2003.

[37] Wonderware_Downloads [En línea]. <http://www.toolboxopc.com/Support/More_Downloads/Wonderware_Downloads/wonderware_downloads.html> [Consulta: Octubre 2007].

BIBLIOGRAFÍAS

About OPC – What is OPC? [En línea]. <http://www.opcfoundation.org/Default.aspx/01_about/01_whatis.asp?MID=AboutOPC> [Consulta: Marzo 2008].

Al final.com: Introducción al Ethernet [En línea]. <<http://www.alfinal.com/Temas/macc.shtml>> [Consulta: Octubre 2007].

Annoyances.org – Getting Internet Connection Sharing (ICS) to work with PPPoE [En línea]. <<http://www.annoyances.org/exec/show/article04-105>> [Consulta: Febrero 2008].

Ch6.pdf [En línea]. <<http://www.automatiomdirect.com/static/manuals/dadnet/ch6.pdf>> [Consulta: Marzo 2008].

Conectar dos PC en red – instalar una tarjeta de red. [En línea]. <http://www.pasarlascanutas.com/tarjeta_de_red/tarjeta_de_red0002.htm> [Consulta: Febrero 2008].

Create your own Database using Delphi's "File Of" Typed files [En línea]. <http://delphi.about.com/od/fileio/a/fileof_delphi.htm> [Consulta: Febrero 2008].

Charte Ojeda, Francisco. Delphi 6, Madrid: ANAYA Multimedia, 2001.

Delphi Pages [En línea]. <<http://www.delhipages.com/threads/thread.cfm?ID=15143&G=15137>> [Consulta: Febrero 2008].

Delphi: Ejecutando una aplicación externa [En línea]. <<http://www.latiumsoftware.com/es/delphi/00002.php>> [Consulta: Febrero 2008].

Essential Pascal [En línea]. <<http://www.marcocantu.com/epascal/English/ch07str.htm>> [Consulta: Febrero 2008].

Ethernet SDK for EBCs, ECOMs and EDRVs [En línea]. <http://www.hosteng.com/Ethernet_SDK.htm> [Consulta: Marzo 2008].

Fabricar cable de red normal y cruzado.. [En línea]. <<http://www.configurarequipo.com/doc383.html>> [Consulta: Febrero 2008].

FileExists function [En línea]. <<http://delphi.about.com/library/rtl/blrtlFileExists.htm>> [Consulta: Febrero 2008].

Internet Packet Exchange [En línea]. <http://es.wikipedia.org/wiki/Internet_Packet_Exchange> [Consulta: Marzo 2008].

Introducción a las Comunicaciones Industriales.pdf [En línea]. <http://www.disa.bi.edu.es/spanish/ftp/material_asignaturas/Fundamentos%20de%20Automatizaci%F3n%20Industrial/Comunicaciones%20y%20Supervisi%F3n/Introducci%F3n%20a%20las%20Comunicaciones%20Industriales.pdf> [Consulta: Noviembre 2007].

Keplcm.pdf [En línea]. <http://www.automationdirect.com/static/manuals/dak_eplcm/keplcm.pdf> [Consulta: Marzo 2008].

MAX1487-MAX491.pdf [En línea]. <<http://datasheets.maxim-ic.com/en/ds/MAX1487-MAX491.pdf>> [Consulta: Septiembre 2007].

MAX220-MAX249.pdf [En línea]. <<http://datasheets.maxim-ic.com/en/ds/MAX220-MAX249.pdf>> [Consulta: Septiembre 2007].

Mitsubishi Electric – Variadores de Frecuencia – FR-S500 [En línea]. <http://www.mitsubishi-automation.es/products/inverters_fr-s500.html> [Consulta: Enero 2008].

Modbus_Messaging_Implementation_Guide_V1_0b.pdf [En línea]. <http://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf> [Consulta: Enero 2008].

Object Linking and Embedding – Wikipedia, the free encyclopedia [En línea]. <http://en.wikipedia.org/wiki/Object_Linking_and_Embedding> [Consulta: Marzo 2008].

OLE for process control – Wikipedia, the free encyclopedia [En línea]. <http://en.wikipedia.org/wiki/OLE_for_process_control> [Consulta: Marzo 2008].

Opto 22 – Controllers, PAC Project, On-the-rack [En línea]. <http://www.opto22.com/site/pr_cat_sc.aspx?qs=100210241055> [Consulta: Marzo 2008].

PLC Timeline | PLCdev [En línea]. <http://www.plcdev.com/plc_timeline> [Consulta: Marzo 2008].

Presentación de la Prof. Tamara Pérez de la cátedra PLC Semestre 03-2006. Escuela de Ingeniería Eléctrica. UCV, Noviembre 2006.

Presentación del Prof. Panayotis Tremante de la cátedra Sistemas de Control Distribuido Semestre 01-2006. Escuela de Ingeniería Eléctrica. UCV, Abril 2004.

Problemas al guardar datos en un archivo [En línea]. <<http://www.clubdelphi.com/forums/showthread.php?t=19778>> [Consulta: Febrero 2008].

Re: Stop another program from Delphi app [En línea]. <<http://www.festra.com/wwwboard/messages/12710.html>> [Consulta: Febrero 2008].

Redes – Monografías.com [En línea]. <<http://www.monografias.com/trabajos11/reco/reco.shtml>> [Consulta: Septiembre 2007].

Revolución Industrial – Wikipedia, la enciclopedia libre [En línea]. <http://es.wikipedia.org/wiki/Revolucion_industrial> [Consulta: Septiembre 2007].

SCIP.be – Articles: Delphi – Starting external applications [En línea]. <<http://www.scip.be/index.php?Page=ArticlesDelphi10&Lang=EN>> [Consulta: Febrero 2008].

Switch – Wikipedia, la enciclopedia libre [En línea]. <<http://es.wikipedia.org/wiki/Switch>> [Consulta: Enero 2008].

Tech Thread – A User’s Guide to Configuring Serial Ports for DirectLOGIC PLC’s: A Two Part Series [En línea]. < http://www.automationnotebook.com/2005_Issue_3/rechthread_January2005.html> [Consulta: Marzo 2008].

Tema3-notas.pdf [En línea]. < http://www.isa.uniovi.es/~chema/ri_archivos/Tema3-notas.pdf> [Consulta: Diciembre 2007].

Torry’s Delphi Pages [En línea]. <<http://www.swissdelphicenter.ch/torry/showcode.php?id=266>> [Consulta: Febrero 2008].

VLAN – Wikipedia, la enciclopedia libre [En línea]. < <http://es.wikipedia.org/wiki/VLAN>> [Consulta: Enero 2008].

Wonderware. Training Manual – Fundamentals of IndustrialSQL Server™ Historial 9.0 and ActiveFactory™ Software 9.1 Course. 2006.

Wonderware. Training Manual – InTouch™ HMI 9.5 Fundamentals of Application Development Course. 2006.

Wonderware. Training Manual – InTouch™ HMI 9.5 Advanced Application Development Course. 2006.

GLOSARIO

ActiveX: Son objetos de programación reutilizables que encapsulan propiedades y ejecutan tareas específicas.

Bug: Es el resultado de un fallo o deficiencia de un programa de computadoras.

COM: Una plataforma introducida por Microsoft que permite la comunicación entre procesos y la creación dinámica de objetos en cualquier lenguaje de programación que lo soporte.

Data Block: Es un objeto para la adquisición de datos de iFix.

DAServer: Es un servidor Stand-Alone de Wonderware para la comunicación con dispositivos industriales y otros.

DCOM: Del inglés, Distribute Component Objetc Model; es un estándar desarrollado por Microsoft que permite la comunicación entre componentes sin importar el lugar donde estén ubicados físicamente.

Diagrama de Escaleras: Es un lenguaje de programación básico basado en los esquemas eléctricos de control clásico.

Drum: Dispositivo de programación para realizar procesos por etapas que involucran el control de varias variables simultáneamente. La palabra se origina en el tambor utilizado antiguamente para la reproducción automática de notas en los pianos.

Deployment: Su traducción al español es Despliegue. Se utiliza para hacer referencia a la instalación y ejecución de un objeto en un sistema.

ECOM: Protocolo propietario de AutomationDirect para la comunicación entre dispositivos de control por Ethernet.

Engine: Se refiere al Motor donde son ejecutados los objetos de una Galaxia.

Failover: Expresión empleada comúnmente para hacer referencia al intercambio de un enlace de comunicación cuando se ha producido una falla.

Galaxia: Es el nombre dado por Wonderware al espacio de trabajo que engloba todos los objetos de aplicaciones y equipos de la automatización.

Gateway: Interface entre dos sistemas de comunicación, protocolos o arquitecturas.

GPRS: Es un servicio de datos móvil orientado a objetos comúnmente empleada en telefonía celular.

Half Duplex: Método de comunicación bidireccional donde la comunicación es realizada a través de una misma línea mediante el uso intermitente esta.

PID: Es un mecanismo genérico para la realimentación en lazos de control. Ampliamente usado en Sistemas de Control.

Plataforma: El objeto que representa un PC físico en el ambiente IDE de Wonderware.

Power Tool: Es un programa incluido en el iFix empleado para la modificación y configuración de los Servidores, grupos y DataBlocks.

Proceso Batch: Proceso que se ejecuta por etapas o pasos. Proceso Secuencial.

RS232: Estándar que define la interface para la comunicación serial entre dos elementos.

RS485: Estándar que define la interface para la comunicación serial multipunto.

SCU: Programa que forma parte del iFix que contiene múltiples opciones para la configuración de objetos de comunicación.

SMS: Un protocolo de comunicaciones que permite el intercambio de mensajes cortos de texto entre teléfonos celulares.

SuiteLink: Es un protocolo Industrial propietario de Wonderware diseñado para realizar la comunicación entre sus productos de forma eficiente.

Script: Conjunto de instrucciones que permiten la automatización de tareas.

ANEXOS

[ANEXO 1]

[Flujo de Trabajo sugerido por Wonderware]

Debido a que existen una gran cantidad de formas para plantear e implementar los proyectos basados en la tecnología ArchestrA, Wonderware sugiere en sus manuales de training [35] una metodología base para la ejecución efectiva de estos. En la Figura 75, se muestra el Flujo de Trabajo sugerido para la realización de un proyecto basado en ArchestrA.

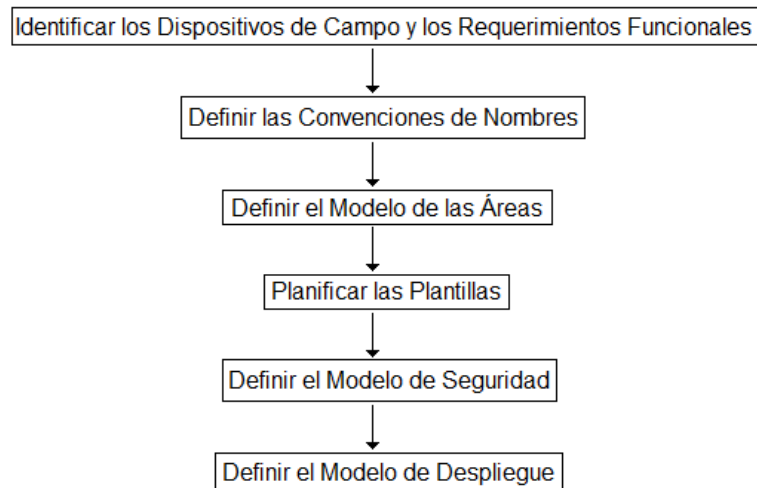


Figura 75. Flujo de Trabajo sugerido por Wonderware.

El primer paso trata de la identificación de todos los dispositivos que forman parte del proyecto, esto significa, buscar el lugar físico donde se encuentran ubicados y realizar un modelo que contenga la relación entre todos los dispositivos. Los Requerimientos Funcionales se refieren a detallar cada uno de los dispositivos en cuanto a la cantidad y tipo de entradas y salidas, cálculos para escalamientos y otros requeridos, almacenamiento de los valores para análisis futuros, alarmas, eventos y la seguridad de cada uno de los dispositivos.

En el segundo paso se sugiere la definición de las convenciones de nombres para cada uno de los dispositivos y variables que así lo requieran. Es importante recalcar que como se está trabajando con un software orientado a objetos, no es necesario incluir en el nombre de las variables la identificación de la herencia, tal como si se estuviese trabajando con un programa para SCADA tradicional.

El tercer paso del Flujo de Trabajo contiene la definición del Modelo de las áreas, en esta se definen cada una de las áreas funcionales y las sub-áreas donde estarán ubicados los objetos referentes a los dispositivos. En una planta típica se definiría un área de recepción, un área de procesos, un área de empaquetamiento y un área de despacho.

El cuarto paso, referente a la definición de las Plantillas, constituye uno de los más importantes en cuanto al ahorro de tiempo y reutilización de objetos se refiere. Una plantilla es un elemento que contiene parámetros comunes de configuración para objetos que son usados varias veces en un proyecto, esto permite la utilización en múltiples ocasiones de un objeto que sólo ha sido definido una sola vez.

Una vez que ha sido creada una plantilla, es posible crear plantillas hijas e instancias a partir de esta. Una plantilla hija, no es más que otra plantilla que posee las configuraciones de los parámetros bases y las restricciones puestas en la plantilla madre, mientras que una Instancia, es un objeto derivado de una plantilla que representa un dispositivo físico o una variable de la planta. Si se llegase a modificar la plantilla madre, los cambios de estas son propagadas a todas sus Instancias, a las plantillas hijas y las Instancias de estas últimas.

El quinto paso trata de la definición del modelo de seguridad, en este se clasifican los objetos en distintos grupos de seguridad que sólo pueden ser accedidos por usuarios identificados con distintos Roles. También, se escoge el tipo de Autenticación de Usuario (también llamado Tipo de Seguridad), siendo las posibles

opciones “None” para ningún tipo de Seguridad, “Galaxy” para una seguridad a nivel de Galaxia, “OS User based” para una seguridad basada en la autorización de usuarios desde el SO de cada PC y “OS Group based” para una seguridad basada en la autorización de usuarios desde el SO de cada PC, pero validados de forma grupal (esto es, si un usuario identificado con un nombre en un PC, tendría el mismo Rol si inicia sesión bajo ese mismo identificador en otro computador).

El sexto paso recomendado por Wonderware es la definición del Modelo de Despliegue, esto es, asignar las áreas y objetos a cada uno de los Engines y Plataformas donde van a ser ejecutados.

Una vez que se ha cumplido todo el Flujo de Trabajo sugerido, sólo hace falta crear los objetos gráficos y las animaciones en el HMI para completar la implementación de un Sistema SCADA bajo la tecnología ArcestrA.

[ANEXO 2]

[Convertidor 232/485]

Lista de componentes empleados:

- 1 74LS04 (Compuerta NOT TTL).
- 1 7805 (Regulador Monolítico de 5 V).
- 4 Condensadores de 1 μ F y 1 Condensador de 100 μ F.
- 1 Conector DB9, 1 Conector de 2 entradas (azul) y otro de 3 entradas (verde).
- 1 Led Rojo y otro Amarillo de 3 mm.
- 1 MAX232.
- 1 MAX485.
- 1 Resistencia de 120 Ω , 3 Resistencias de 3.3k Ω y 2 Resistencias de 510 Ω .

Imagen y Board ampliados del convertidor 232/485.

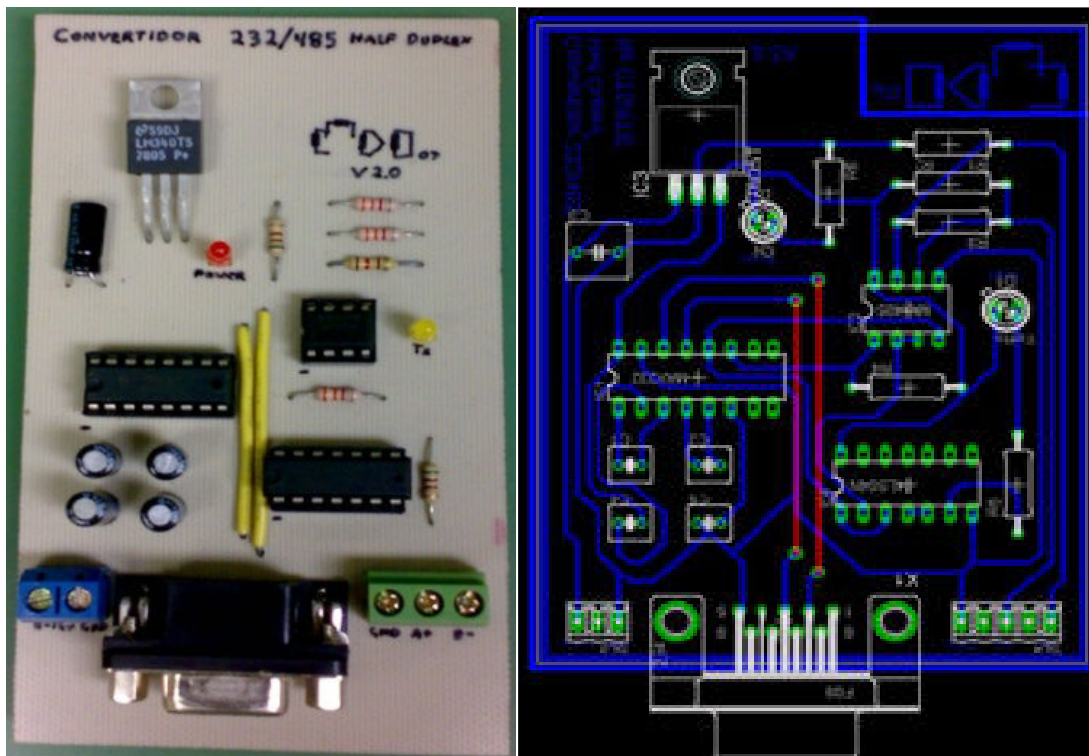


Figura 76. Imagen y Board del Convertidor 232/485.

[ANEXO 3]

[Programación realizada al PLC de la Maqueta]

Un esquema que posee todos los dispositivos de la Maqueta de Procesos del LCID y su relación con las variables de Entrada, Salida y Memorias del PLC DirectLogic, se muestra en la Figura 77.

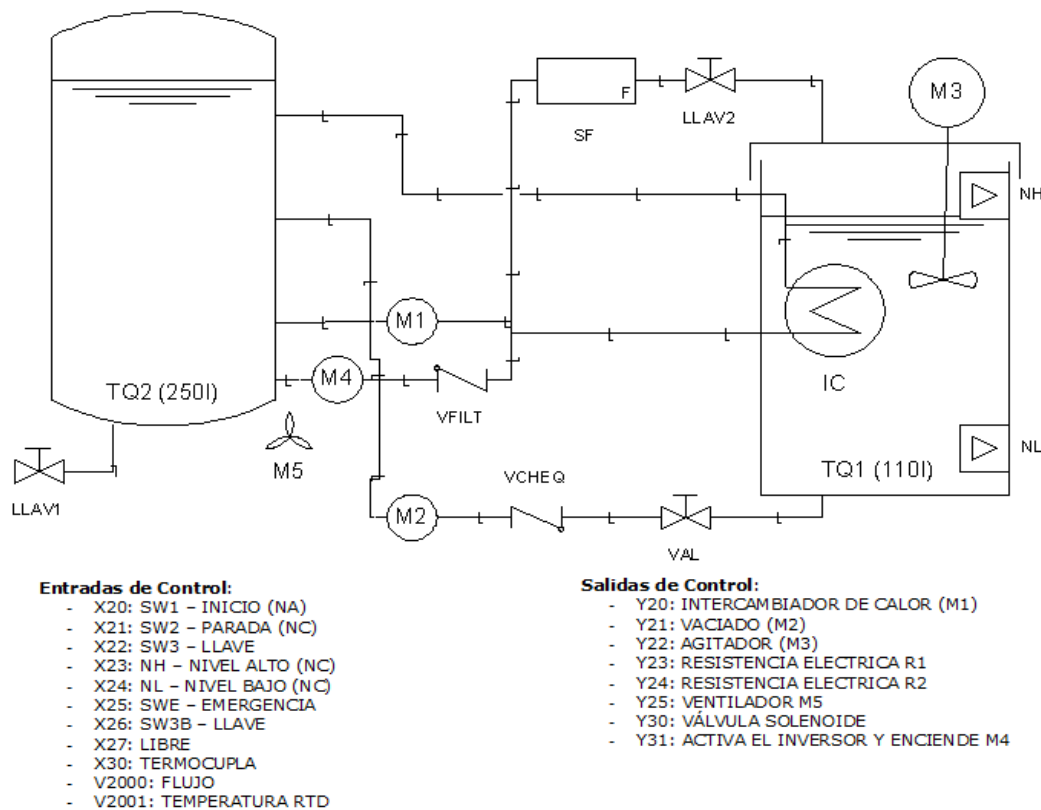


Figura 77. Esquema de la Maqueta de Procesos.

La programación del PLC que controla la maqueta se realiza empleando el programa DirectSoft. Inicialmente es configurado el Enlace de comunicación, para esto, se seleccionó la Familia de la Serie 205, el PLC del tipo DL250, el puerto de comunicaciones Ethernet, el protocolo de transporte UDP/IP, la identificación del módulo como ID 3 y el protocolo ECOM. Estas configuraciones antes mencionadas realizadas en el ambiente del programa DirectSoft son mostradas en la Figura 78.

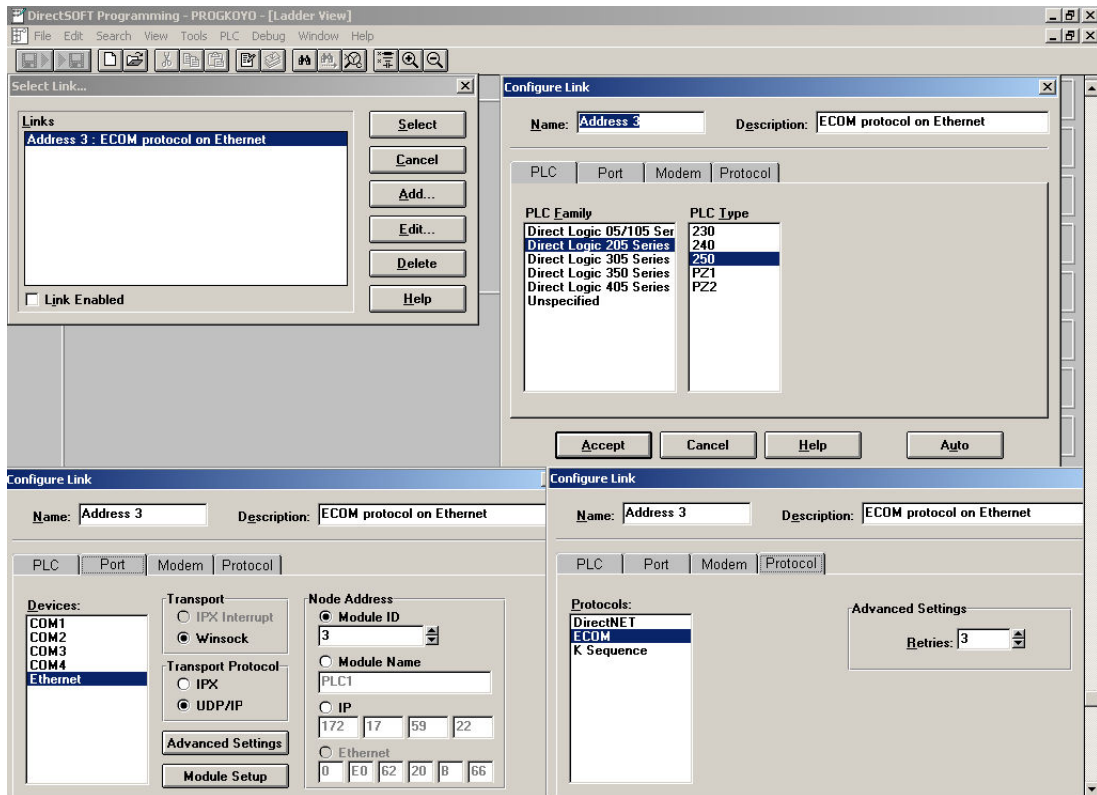


Figura 78. Configuración del enlace de comunicación.

Una vez establecida la comunicación con el PLC, se procede a la programación del mismo. A continuación se muestran y describen cada una de las etapas de la programación en diagrama de escalera:

La primera línea contiene el registro especial SP1 que se coloca en 1 siempre y cuando este el PLC en el estado de ejecución. Este se emplea para indicar que la programación en el PLC es la correcta. La segunda línea contiene la instrucción para realizar el apagado de todas las memorias de los dispositivos de la maqueta cuando es presionado el botón de Emergencia. Las líneas de código 3 hasta la 6, se colocan para generar un retardo corto entre la activación de alguno de los botones de inicio o parada y su reconocimiento por parte del SCADA.

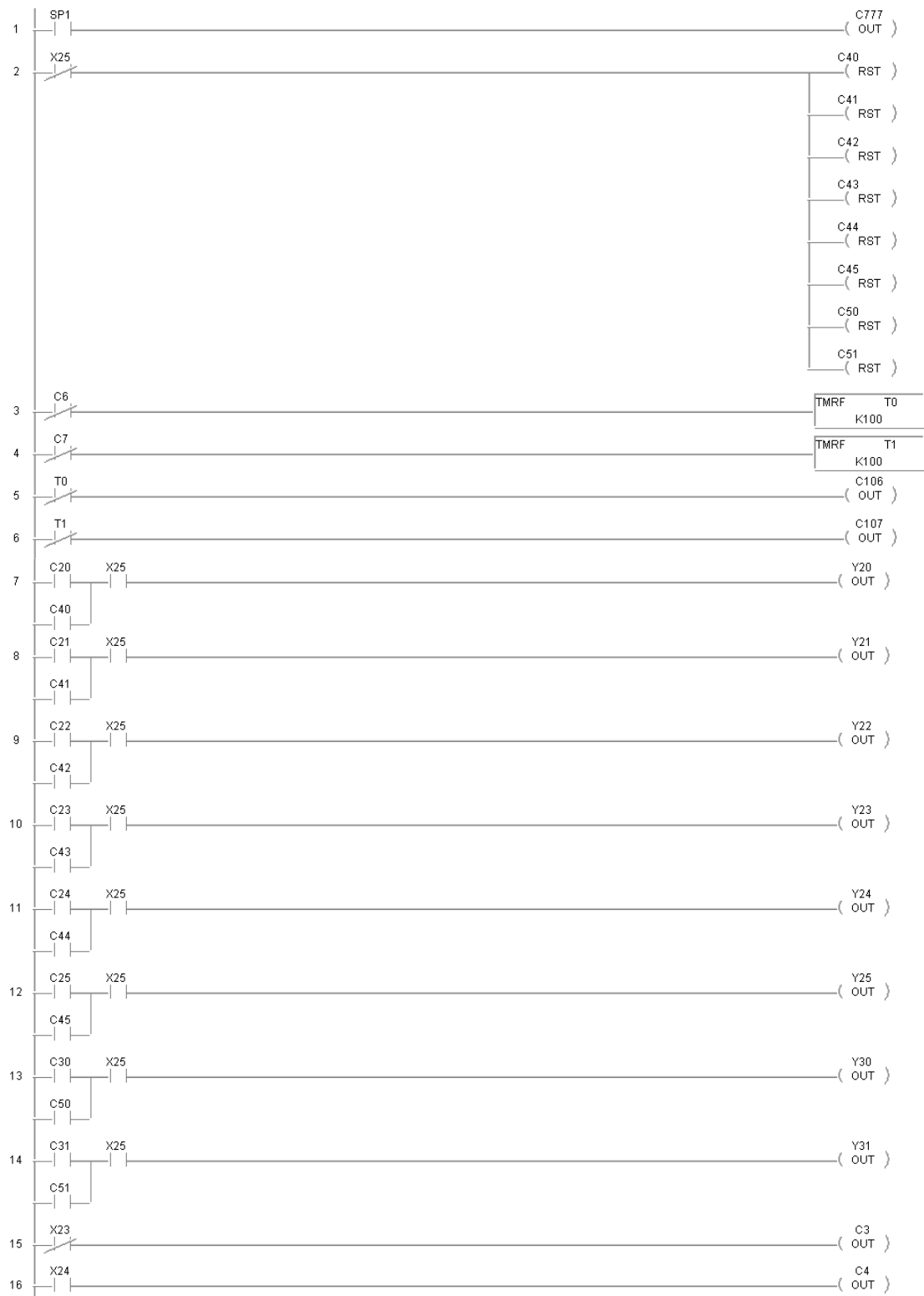


Figura 79. Primera parte del Diagrama de Escaleras.

Desde la línea 7 hasta la 14 se realiza la programación de las salidas. En estas, se asocian 2 banderas a cada una de las salidas, las que comienzan por 20 y 30 para la asignación al Drum y las que empiezan por 40 y 50 para la modificación manual por software. El botón de emergencia también es colocado en todas las salidas para la interrupción de estas una vez presionado. Las líneas 15 y 16 son empleadas para relacionar la activación del límite de nivel alto y bajo del tanque con dos banderas para ser usadas en el Drum.

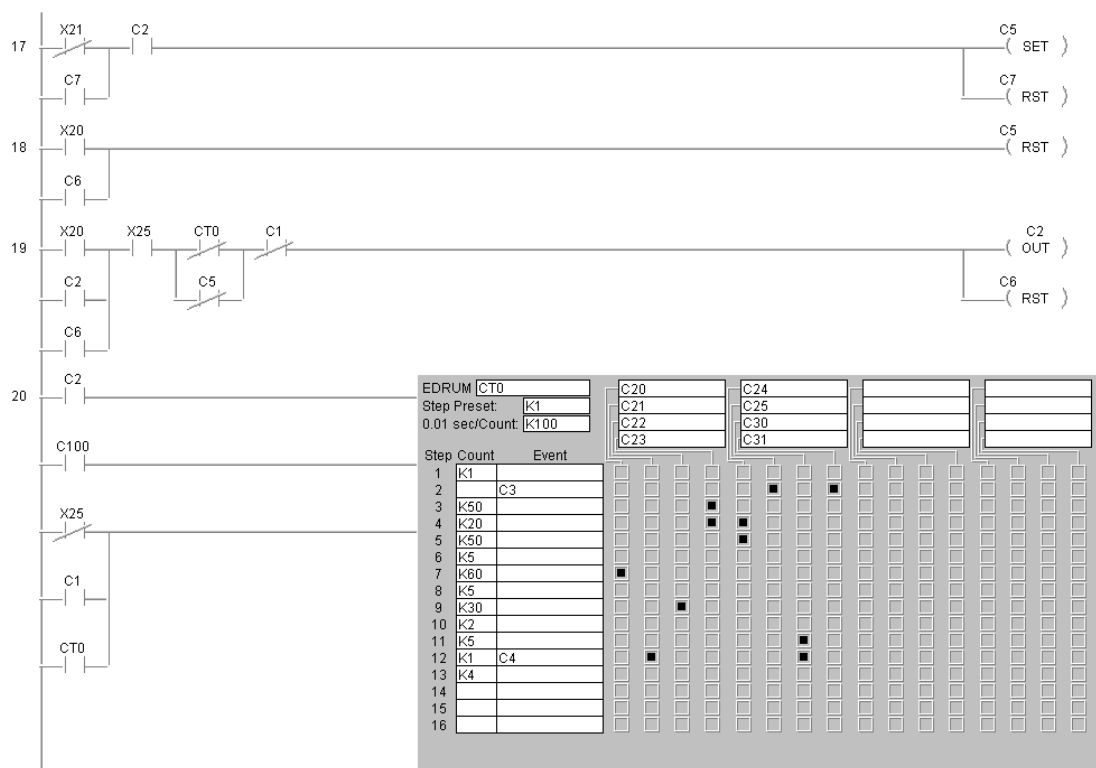


Figura 80. Segunda parte del Diagrama de Escaleras.

En las líneas 17,18 y 19, se realiza la programación de banderas para el control del inicio y la parada del Drum. La línea 20 contiene el Drum que viene a ser la parte fundamental de la programación del proceso tipo Batch en el PLC. En este, se programa un proceso que cumple las siguientes instrucciones:

- Se llena el Tanque izquierdo hasta el límite de nivel alto.
- Se enciende la Resistencia Térmica 1 durante 50 segundos.

- Se encienden ambas Resistencias Térmicas por 20 segundos.
- Se apaga la RT1 y se mantiene encendida la RT2 durante 50 segundos.
- Se apagan ambas resistencias durante 5 segundos.
- Se enciende el Intercambiador de Calor por un minuto.
- Se apaga el IC y se deja reposar durante 5 segundos.
- Se enciende el agitador por medio minuto.
- Se apaga el agitador y se esperan 2 segundos.
- Se abre la válvula de vaciado y 5 segundos después se vacía el tanque hasta el nivel bajo.
- Se repite el proceso si no se ha presionado el botón de Parada.

Una imagen donde se detalla de mejor forma la configuración del EDRUM empleado en la programación del PLC, se muestra en la Figura 81.

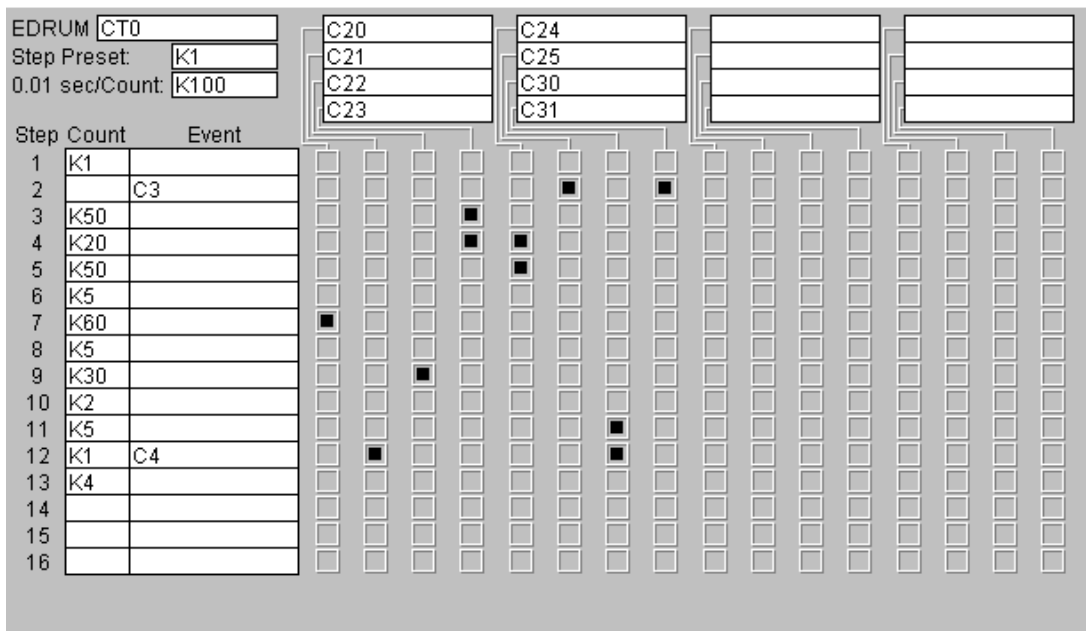


Figura 81. Configuración del EDRUM.

La última parte de la programación se utiliza para realizar la adquisición de todas las entradas y salidas no booleanas y guardarlas en posiciones de memorias accesibles desde el SCADA.

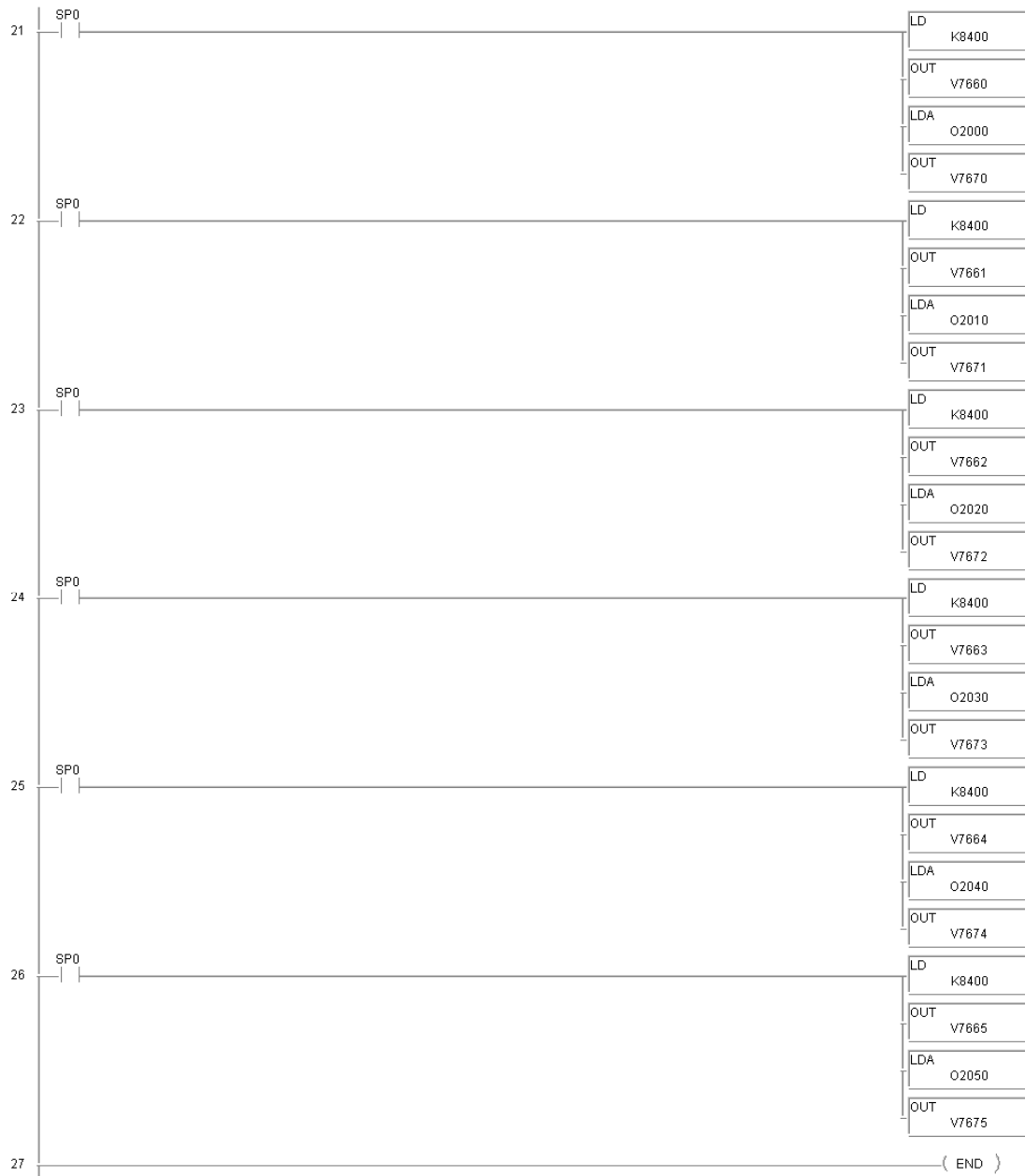


Figura 82. Última parte del Diagrama de Escaleras.

[ANEXO 4]

[Modelo General y Convención de Nombres del SCADA]

La versión más reciente del modelo general del SCADA es planteada por Luis Ortiz en el 2003 para su trabajo grado, en este se definen los tipos de enlaces y la ubicación de las remotas en la Universidad. El esquema se muestra en la Figura 83.

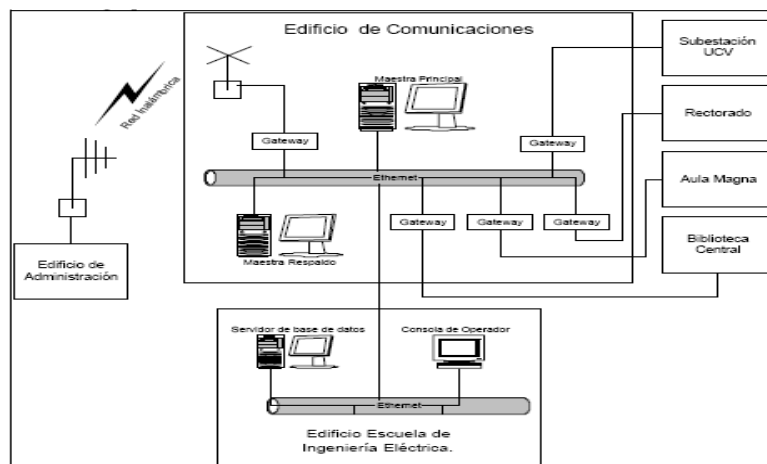


Figura 83. Esquema general del SCADA.

La convención de nombres más reciente para la definición de Tags es realizada por N. Salcedo en su trabajo de Especialización y se muestra en la Tabla 5.

Canal	Remota	Descripción	Cantidad de Dígitos para direccionamiento	Bit Base	Dirección IP del Gateway	Dirección Modbus
Canal0	A8	Alimentador A8 S/E UCV	4	0	192.168.1.10	1
Canal0	A8-C2	Alimentador A8 Circuito 2 - S/E UCV	4	0	192.168.1.10	3
Canal0	A8-C3	Alimentador A8 Circuito 3 - S/E UCV	4	0	192.168.1.10	4
Canal0	A8-C4	Alimentador A8 - Circuito 4 - S/E UCV	4	0	192.168.1.10	5
Canal0	A8-C5	Alimentador A8 - Circuito 5 - S/E UCV	4	0	192.168.1.10	6
Canal0	A8-C6	Alimentador A8 - Circuito 6 - S/E UCV	4	0	192.168.1.10	7
Canal0	A6	Alimentador A6 - S/E UCV	4	0	192.168.1.10	8
Canal0	A6-C1	Alimentador A6 - Circuito 1 - S/E UCV	4	0	192.168.1.10	9
Canal0	A6-C2	Alimentador A6 - Circuito 2 - S/E UCV	4	0	192.168.1.10	10
Canal0	ST-12-1	Rectorado - Acometida 1 - A8C6	4	0	192.168.1.10	11
Canal0	ST-12-2	Rectorado - Acometida 2 - A8C6	4	0	192.168.1.10	12
Canal0	ST-3-1	Aula Magna - Acometida 1 - A8C4	4	0	192.168.1.10	13
Canal0	ST-3-2	Aula Magna - Acometida 2 - A8C4	4	0	192.168.1.10	14
Canal0	ST-3-3	Aula Magna - Acometida 3 - A8C6	4	0	192.168.1.10	15
Canal0	ST-14-1	Biblioteca Central - Acometida 1 - A8C6	4	0	192.168.1.10	16
Canal0	ST-14-2	Biblioteca Central - Acometida 2 - A8C6	4	0	192.168.1.10	17
Canal0	CT-37-1	Admin y Educación - Acometida 1 - A6C1	4	0	192.168.1.10	18
Canal0	CT-37-2	Admin y Educación - Acometida 2 - A6C2	4	0	192.168.1.10	19
Canal0	ST-35-1	FACES - Acometida 1 - A6C2	4	0	192.168.1.10	20
Canal0	ST-36-1	FACES - Acometida 2 - A6C2	4	0	192.168.1.10	21

Tabla 5. Convención de nombres del SCADA.

[ANEXO 5]

[Programa Cerrar]

En la Figura 84 es mostrada la ventana desde el ambiente Delphi 6 para la programación del programa Cerrar y seguidamente el Código del programa.

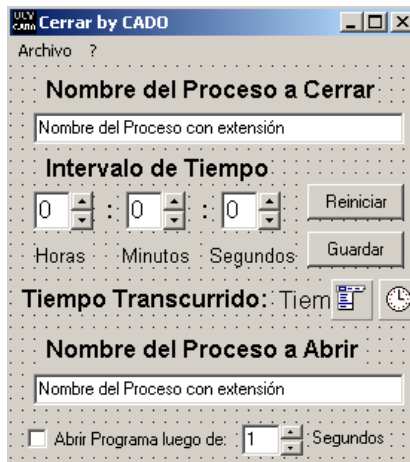


Figura 84. Ventana de Delphi para la programación del programa Cerrar.

Código de Programación

Unit CerrarTOP;

Interface

Uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Tlhelp32, ComCtrls, ExtCtrls, ToolWin, ActnMan,
ActnCtrls, ActnMenus, Menus, ShellApi;

Type

TForm1 = class(TForm)
 Button1: TButton;
 Timer1: TTimer;
 Label1: TLabel;
 Edit1: TEdit;
 UpDown1: TUpDown;
 Edit2: TEdit;
 UpDown2: TUpDown;
 Label2: TLabel;
 Label3: TLabel;
 Label4: TLabel;
 Edit3: TEdit;
 UpDown3: TUpDown;
 Label5: TLabel;
 Label6: TLabel;
 MainMenu1: TMainMenu;
 Archivo1: TMenuItem;
 Ayuda1: TMenuItem;
 Salir1: TMenuItem;

```

Ayuda2: TMenuItem;
Acerca1: TMenuItem;
Predeterminar1: TMenuItem;
Label8: TLabel;
Button2: TButton;
Label7: TLabel;
Edit4: TEdit;
Label9: TLabel;
CheckBox1: TCheckBox;
Edit5: TEdit;
Label10: TLabel;
UpDown4: TUpDown;
Label11: TLabel;
Edit6: TEdit;
procedure Button1Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Ayuda2Click(Sender: TObject);
procedure Acerca1Click(Sender: TObject);
procedure Salir1Click(Sender: TObject);
procedure Predeterminar1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure CheckBox1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

```

Var

```

Form1: TForm1;
Segundos, Minutos, Horas, SSegundos, SMinutos, SHoras, AbrirSeg, SAbrirSeg: Integer;
Nombre, SNombre, SNombre2: String;
SNombre1:PChar;
cargo, Abrir: Boolean;

```

Implementation

```
{SR *.dfm}
```

Procedure Escribe(Seg,Minu,Hor:Integer; Nomb:String; Ab:Boolean; AbSeg:Integer; Nomb1:String);

```

var
  Archivo: textfile;
  Aux:String;
begin
  AssignFile(Archivo,'C:\cado.ucv');
  Rewrite(Archivo);
  Writeln(Archivo,'8486');
  Aux:=inttostr(Seg);
  Writeln(Archivo,Aux);
  Aux:=inttostr(Minu);
  Writeln(Archivo,Aux);
  Aux:=inttostr(Hor);
  Writeln(Archivo,Aux);
  Writeln(Archivo,Nomb);
  If Ab Then
  begin
    Writeln(Archivo,'1');
    Aux:=inttostr(AbSeg);
    Writeln(Archivo,Aux);
    Writeln(Archivo,Nomb1);
  end
  Else
  begin
    Writeln(Archivo,'0');
  end;
  Writeln(Archivo,'*****');
  Writeln(Archivo,'*****');
  Writeln(Archivo,'*****');
  Writeln(Archivo,'NO BORRAR. Archivo empleado en el Programa "Cerrar by CADO" ');
  Writeln(Archivo,' Carlos Di Yorio, UCV Caracas-Venezuela 2008 ');
  Writeln(Archivo,'*****');
  Writeln(Archivo,'*****');
  Writeln(Archivo,'*****');
  CloseFile(Archivo);
  Showmessage('El Archivo C:\cado.ucv ha sido modificado satisfactoriamente.');
```

```
end;
```



```

Procedure Lee(Var Seg,Minu,Hor: Integer; Var Nomb:String; Edit1,Edit2,Edit3,Edit4,Edit5,Edit6:TEdit; Var Ab:Boolean; Var
AbSeg:Integer; Var CheckBox:TCheckBox; UpDown:TUpDown; Var Nomb1:String);
var
  Archivo: textfile;
  Aux:String;
begin
  AssignFile(Archivo,'C:\cado.ucv');
  If not FileExists('C:\cado.ucv') Then
    begin
    Application.MessageBox('El Archivo no fue encontrado. Se creará uno con los valores predeterminados.','Advertencia',
    MB_ICONINFORMATION);
    Seg:=0;
    Minu:=0;
    Hor:=2;
    Nomb:='ServerMain.exe';
    Escribe(Seg,Minu,Hor,Nomb,Abriu,SAbrirSeg,Nomb1);
    end;
  Reset(Archivo);
  If not EOF(Archivo) then
    begin
    Readln(Archivo,Aux);
    If Aux='8486' then
      begin
      Readln(Archivo,Aux);
      Seg:=strtoint(Aux);
      Readln(Archivo,Aux);
      Minu:=strtoint(Aux);
      Readln(Archivo,Aux);
      Hor:=strtoint(Aux);
      Readln(Archivo,Nomb);
      Readln(Archivo,Aux);
      If Aux='1' Then
        begin
        Ab:=true;
        CheckBox.Checked:=true;
        Edit5.Enabled:=true;
        UpDown.Enabled:=true;
        Readln(Archivo,Aux);
        AbSeg:=strtoint(Aux);
        Readln(Archivo,Nomb1);
        Edit6.Enabled:=true;
        Edit6.Text:=Nomb1;
        end
      else
        begin
        Ab:=false;
        AbSeg:=1;
        end;
      end
    else
      begin
      CloseFile(Archivo);
      Seg:=0;
      Minu:=0;
      Hor:=2;
      Nomb:='ServerMain.exe';
      Escribe(Seg,Minu,Hor,Nomb,Ab,AbSeg,Nomb1);
      Application.MessageBox('El Archivo esta corrupto. Se ha creado uno con los valores predeterminados.','Advertencia', MB_ICONWARNING);
      end;
    end
  else
    begin
    CloseFile(Archivo);
    Seg:=0;
    Minu:=0;
    Hor:=2;
    Nomb:='ServerMain.exe';
    Escribe(Seg,Minu,Hor,Nomb,Ab,AbSeg,Nomb1);
    Application.MessageBox('El Archivo está en blanco. Se ha creado uno con los valores predeterminados.','Advertencia', MB_ICONWARNING);
    end;
  CloseFile(Archivo);
  Edit1.Text:=inttostr(Hor);
  Edit2.Text:=inttostr(Minu);
  Edit3.Text:=inttostr(Seg);
  Edit4.Text:=Nomb;
  Edit5.Text:=inttostr(AbSeg);

```

```

end;

Function KillTask(ExeFileName: string): Integer;
const
  PROCESS_TERMINATE = $0001;
var
  ContinueLoop: BOOL;
  FSnapshotHandle: THandle;
  FProcessEntry32: TProcessEntry32;
begin
  Result := 0;
  FSnapshotHandle := CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS, 0);
  FProcessEntry32.dwSize := SizeOf(FProcessEntry32);
  ContinueLoop := Process32First(FSnapshotHandle, FProcessEntry32);

  while Integer(ContinueLoop) <> 0 do
  begin
    if ((UpperCase(ExtractFileName(FProcessEntry32.szExeFile)) =
      UpperCase(ExeFileName)) or (UpperCase(FProcessEntry32.szExeFile) =
      UpperCase(ExeFileName))) then
      Result := Integer(TerminateProcess(
        OpenProcess(PROCESS_TERMINATE,
          BOOL(0),
          FProcessEntry32.th32ProcessID),
          0));
      ContinueLoop := Process32Next(FSnapshotHandle, FProcessEntry32);
    end;
  CloseHandle(FSnapshotHandle);
  end;

  procedure TForm1.Button1Click(Sender: TObject);
  begin
    KillTask(SNombre);
    Segundos:=0;
    Minutos:=0;
    Horas:=0;
  end;

Procedure TForm1.Timer1Timer(Sender: TObject);
begin
  If not cargo Then
  begin
    cargo:=true;
    Lee(SSegundos,SMinutos,SHoras,SNombre,Edit1,Edit2,Edit3,Edit4,Edit5,Edit6,Abrir,SAbrirSeg,CheckBox1,UpDown1,SNombre2);
  end;
  Segundos:=Segundos+1;
  If Segundos=60 Then
  begin
    Segundos:=0;
    Minutos:=Minutos+1;
    If Minutos=60 Then
    begin
      Minutos:=0;
      Horas:=Horas+1;
    end;
  end;
  If (Horas=SHoras) and (Minutos=SMinutos) and (Segundos=SSegundos) Then
  begin
    KillTask(SNombre);
    Segundos:=0;
    Minutos:=0;
    Horas:=0;
  end;
  If Abrir and (Horas=0) and (Minutos=0) and (Segundos=SAbrirSeg) Then
  begin
    If ShellExecute(Form1.Handle, 'open',PChar (SNombre2), nil, nil, SW_MINIMIZE)<=32 Then
      Application.MessageBox('No se pudo ejecutar la aplicación debido a que el programa a iniciar no fue encontrado o es incorrecto.!', 'Advertencia',
      MB_ICONEXCLAMATION);
    end;
    Label7.Caption:=inttostr(Horas)+ ':'+inttostr(Minutos)+ ':'+inttostr(Segundos);
  end;
end;

Procedure TForm1.FormCreate(Sender: TObject);
begin
  Segundos:=0;
  Minutos:=0;
  Horas:=0;

```

```

cargo:=false;
Abrir:=false;
SAbrirSeg:=1;
end;

```

Procedure TForm1.Ayuda2Click(Sender: TObject);

```

begin
  Application.MessageBox('Este programa cierra y abre un proceso cada cierto tiempo. El Nombre del proceso con la extensión debe ser
introducido en el campo correspondiente. Al modificar algún dato, se debe presionar el botón de guardar para que sean debidamente
cargados.','AYUDA', MB_ICONASTERISK);
end;

```

Procedure TForm1.Acerca1Click(Sender: TObject);

```

begin
  Application.MessageBox('Programa realizado por Carlos Di Yorio. EIE-UCV, Caracas - Venezuela 2008. Versión 3.2.','ACERCA',
MB_ICONASTERISK);
end;

```

Procedure TForm1.Salir1Click(Sender: TObject);

```

begin
  Close();
end;

```

Procedure TForm1.Predeterminar1Click(Sender: TObject);

```

begin
  SHoras:=strtoint(Edit1.Text);
  SMinutos:=strtoint(Edit2.Text);
  Ssegundos:=strtoint(Edit3.Text);
  SNombre:=Edit4.Text;
  SAbrirSeg:=strtoint(Edit5.Text);
  SNombre2:=Edit6.Text;
  If (((SHoras>23) or (SHoras<0) or (SMinutos>60) or (SMinutos<0) or (Ssegundos>60) or (Ssegundos<0) or (SAbrirSeg<1) or (SAbrirSeg>59))
or ((SHoras=0) and (SMinutos=0) and (Ssegundos=0)) or (SNombre="") or (SNombre2=")) Then
  Application.MessageBox('No se guardaron los datos porque uno o todos son incorrectos o están fuera del límite.','ERROR', MB_ICONERROR)
else
  Escribe(Ssegundos,SMinutos,SHoras,SNombre,Abrir,SAbrirSeg,SNombre2)
end;

```

Procedure TForm1.Button2Click(Sender: TObject);

```

begin
  SHoras:=strtoint(Edit1.Text);
  SMinutos:=strtoint(Edit2.Text);
  Ssegundos:=strtoint(Edit3.Text);
  SNombre:=Edit4.Text;
  SAbrirSeg:=strtoint(Edit5.Text);
  SNombre2:=Edit6.Text;
  If (((SHoras>23) or (SHoras<0) or (SMinutos>60) or (SMinutos<0) or (Ssegundos>60) or (Ssegundos<0) or (SAbrirSeg<1) or (SAbrirSeg>59))
or ((SHoras=0) and (SMinutos=0) and (Ssegundos=0)) or (SNombre="") or (SNombre2=")) Then
  Application.MessageBox('No se guardaron los datos porque uno o todos son incorrectos o están fuera del límite.','ERROR', MB_ICONERROR)
else
  Escribe(Ssegundos,SMinutos,SHoras,SNombre,Abrir,SAbrirSeg,SNombre2)
end;

```

Procedure TForm1.CheckBox1Click(Sender: TObject);

```

begin
  IF CheckBox1.Checked Then
  begin
    Edit5.Enabled:=True;
    UpDown4.Enabled:=True;
    Abrir:=True;
    Edit6.Enabled:=True;
  end
  Else
  begin
    Edit5.Enabled:=false;
    UpDown4.Enabled:=false;
    Abrir:=false;
    Edit6.Enabled:=false;
  end;
end;
end.

```

[ANEXO 6]

[Contraseñas de los Computadores del SCADA]

Todas las PCs del Scada requieren de un inicio manual una vez reiniciadas o apagadas, esto ciertamente esta no es la mejor opción. Todos los computadores deberían ser configurados para realizar el inicio de sesión automático, para en caso de que hubiese un corte de energía o cualquier otra causa que reinicie el sistema, este no deba esperar el inicio de sesión por parte de un operador para poder realizar su función. Esto fue dejado así por recomendaciones de los administradores de los laboratorios. En la Tabla 6, se enumeran los inicios de sesión para cada uno de los computadores utilizados en el SCADA.

Nombre	Ubicación	Sesión Recomendada	Comentarios
SW0	Lab. Control <i>Llave:</i> 5018461	<u>Usuario:</u> Administrator <u>Contraseña:</u> wwucv	También es posible iniciar sesión con la cuenta “sa” utilizando la misma contraseña.
SW2	Lab. Control <i>Llave:</i> 5018463	<u>Usuario:</u> Administrator <u>Contraseña:</u> wwucv	
Scada0	Lab. Máquinas <i>Llave:</i> 5018450	<u>Usuario:</u> Administrator <u>Contraseña:</u> wwucv	
Scada1	Lab. Máquinas	<u>Usuario:</u> diyorio <u>Contraseña:</u>	La contraseña es vacía.
Scada2	Lab. Máquinas <i>Llave:</i> 5018455	<u>Usuario:</u> diyorio <u>Contraseña:</u>	
Scada3	Lab. Máquinas	<u>Usuario:</u> diyorio <u>Contraseña:</u>	
Control1	Lab. Control <i>Llave:</i> 5018464	<u>Usuario:</u> sa <u>Contraseña:</u> wwucv	La cuenta tipo Administrador o Administrator es restringida por el encargado del Laboratorio.
Control7	Lab. Control <i>Llave:</i> 5018470	<u>Usuario:</u> sa <u>Contraseña:</u> wwucv	
Control8	Lab. Control <i>Llave:</i> 5018454	<u>Usuario:</u> sa <u>Contraseña:</u> wwucv	

Tabla 6. Contraseñas de los Computadores del SCADA.