ESCUELA POLITECNICA NACIONAL

FACULTAD DE INGENIERIA ELECTRICA

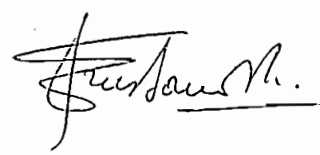
IMPLEMENTACION DE UN SISTEMA SCADA PARA LA OPERACION DE PROTOTIPOS DE LABORATORIO

EDGAR JAVIER PROAÑO CEVALLOS

Tesis previa a la obtención del Título de
Ingeniero en Electrónica y Control

QUITO, NOVIEMBRE 1995

Certifico que la presente Tesis ha sido elaborada en su totalidad por el señor Edgar Javier Proaño Cevallos.



Ing. Patricio Burbano
Director de Tesis

AGRADECIMIENTO:

A mis padres y hermanos, por su constante amor y permanente entrega;

Al Ing. Patricio Burbano, por su inestimable ayuda para desarrollar este trabajo;

A todos los profesores que contribuyeron no solo a mi instrucción profesional, sino a mi formación personal;

A mis compañeros y amigos con los que he compartido estos años de estudio.

DEDICATORIA:

A la mayor gloria de Dios, Uno y Trino; y al honor de la Santísima Virgen María.

CONTENIDO

Página

Capítulo 1: INTRODUCCION 1

1.1. ASPECTOS GENERALES 2

1.2. EVOLUCION DE LOS SISTEMAS DE CONTROL DIGITAL 4

1.2.1. Control digital directo 5

1.2.2. Control secuencial 8

1.2.3. Control distribuido 10

1.2.4. Control con instrumentación inteligente 13

1.2.5. Control supervisor 15

Capítulo 2: SISTEMAS *SCADA* 17

2.1. DESCRIPCION GENERAL DE UN SISTEMA *SCADA* 18

2.1.1. Definición 18

2.1.2. Ventajas de la aplicación de sistemas de control *SCADA* 20

2.1.3. Implementación de sistemas *SCADA* en el país 21

2.2. CONFIGURACION DE UN SISTEMA *SCADA* 22

2.2.1. Sistemas de control distribuido, origen de los sistemas *SCADA* 23

2.2.2. Módulos operacionales en los sistemas *SCADA* 23

2.2.2.1. Automatización básica 23

2.2.2.2. Plataforma de supervisión 23

2.2.2.3. Medios de comunicación 25

2.2.3. Alternativas configurativas para la implementación de un sistema *SCADA* 26

2.2.3.1. Interconexión multidrop 27

2.2.3.2. Interconexión radial y ramificada 28

2.5.2. Alternativas de control digital en tiempo real 60
2.5.2.1. Controladores dedicados 60
2.5.2.2. Controladores lógicos programables 61
2.5.2.3. Controles computarizados 62
2.5.3. Algoritmos digitales para control en tiempo real 64
2.5.3.1. Algoritmos convencionales 64
2.5.3.2. Algoritmos no lineales 66
2.5.3.3. Algoritmos óptimos 66
2.5.3.4. Algoritmos adaptivos 67
2.5.3.5. Algoritmos inteligentes 68

Capítulo 3: DISEÑO DE UN SISTEMA *SCADA* PARA PROTOTIPOS DE LABORATORIO 70

3.1. ESTRUCTURACION DEL SISTEMA 71
3.1.1. Fines didácticos del sistema *SCADA* de laboratorio 71
3.1.2. Condiciones adversas a la implementación del sistema *SCADA* de laboratorio 72
3.1.3. Caraterísticas del sistema *SCADA* de laboratorio 72
3.1.3.1. Configuración general 72
3.1.3.2. Control digital directo 73
3.1.3.3. Plataforma de supervisión 74
3.1.4. Posibilidades de expansión del sistema *SCADA* de laboratorio 75
3.2. INTEGRACION DE PROTOTIPOS 76
3.2.1. Sistema de control de nivel de líquidos 76
3.2.2. Módulo electrónico de simulación de funciones de transferencia 78
3.3. MODELACION Y SIMULACION DE PLANTAS INDUSTRIALES 80
3.3.1. Simulador de un horno batch 80

3.3.2. Simulador universal de plantas lineales de segundo orden 88
3.4. ADQUISICION Y TRANSMISION DE INFORMACION 88
3.4.1. Adquisición y salida de datos con prototipos reales 89
3.4.1.1. Descripción general de la tarjeta DAS-128 89
3.4.1.2. Acceso a los pórticos 90
3.4.2. Transmisión de datos con prototipos virtuales 91
3.4.2.1. Intercambio dinámico de datos 92
3.4.2.2. Consideraciones especiales en la elaboración de prototipos virtuales para su utilización con el sistema *SCADA* de laboratorio 93
3.4.3. Consolidación de operaciones de adquisición y transmisión de datos 94

Capítulo 4: IMPLEMENTACION DE SOFTWARE PARA SUPERVISION Y CONTROL 96
4.1. REQUERIMIENTOS DEL SISTEMA 97
4.1.1. Requerimientos de hardware 97
4.1.2. Requerimientos de software 98
4.2. DIRECTRICES PARA LA PROGRAMACION 98
4.3. MODULOS DEL PROGRAMA 100
4.3.1. Archivo de proyecto SCADA.MAK 101
4.3.2. Módulo general DEFINE.BAS 101
4.3.3. Módulo general GENERAL.BAS 101
4.3.4. Forma SCDMAIN.FRM 101
4.3.5. Forma SUMMARY.FRM 104
4.3.6. Cuadro de diálogo DLGCONFIG.FRM 104
4.3.7. Cuadro de diálogo DLGLOOPS.FRM 105
4.3.8. Cuadros de diálogo DLGVAR.FRM, INPDF.FRM y OPTDF.FRM 107

4.3.9. Cuadro de diálogo DLGCFN.FRM 109
4.3.10. Cuadro de diálogo DLGEVENT.FRM 110
4.3.11. Cuadro de diálogo DLGALARM.FRM 113
4.3.12. Cuadro de diálogo DLGSTAT.FRM 114
4.3.13. Cuadro de diálogo DLGREGIST.FRM 114
4.3.14. Cuadro de diálogo DLGSELECT.FRM 114
4.3.15. Forma FRMONLINE.FRM 117
4.3.16. Ventanas flotantes GFGRAPH.FRM, GFSTAT.FRM GFALARM.FRM, GFCONT.FRM y GDMSG.FRM 117
4.3.17. Temporizador TIMEX 120

Capítulo 5: ENSAYOS, RESULTADOS Y CONCLUSIONES 124

5.1. ENSAYOS SOBRE EL SISTEMA *SCADA* DE LABORATORIO 125
5.1.1. Pruebas de comunicación 125
5.1.2. Pruebas sobre las funciones de control y supervisión 126
5.1.3. Pruebas con prototipos complejos 126
5.1.3.1. Pruebas en el sistema de control de nivel de líquidos 127
5.1.3.2. Pruebas en el horno batch 127
5.2. VALIDACION DE RESULTADOS 128
5.2.1. Resultados de la operación del sistema de control de nivel de líquidos bajo el dominio del sistema *SCADA* 128
5.2.2. Resultados de la operación del horno batch bajo el dominio del sistema *SCADA* 132
5.3. CONCLUSIONES Y RECOMENDACIONES 135
5.3.1. Respecto a los sistemas *SCADA* en general 135
5.3.2. Respecto al sistema *SCADA* de laboratorio 136
5.3.3. Respecto a programación en Windows de aplicaciones de tiempo real 138

5.3.4. Respecto a la preparación profesional en sistemas *SCADA* 140

5.3.5. Respecto a afinamientos en la operación del sistema *SCADA* de laboratorio 141

REFERENCIAS BIBLIOGRAFICAS 143

ANEXOS 149

A. MANUAL DE USUARIO DEL PAQUETE COMPUTACIONAL *SCADA SYSTEM* 150

B. ARCHIVOS EN VISUAL BASIC 3.0 DEL PAQUETE COMPUTACIONAL *SCADA SYSTEM* 184

C. APLICACION *SCADA SYSTEM* 184

CAPITULO 1

INTRODUCCION

1.1. ASPECTOS GENERALES

1.2. EVOLUCION DE LOS SISTEMAS DE CONTROL DIGITAL

1.1. ASPECTOS GENERALES.

El objetivo de la presente tesis es la implementación de un sistema de control supervisor con adquisición inteligente de datos, para la operación y control de múltiples prototipos de plantas industriales a pequeña escala.

El proyecto abarca un estudio teórico concreto y suficientemente profundo de los sistemas *SCADA*, en lo referente a configuración topológica, adquisición y transmisión inteligente de datos, supervisión de eventos, tratamiento de información y aplicación de algoritmos control digital en estructuras distribuidas.

Cuando haya sido establecida la base teórica apropiada sobre la cual se pueda elaborar un diseño coherente, se procederá a estructurar un modelo a escala de laboratorio, de un sistema *SCADA*, de acuerdo a las facilidades y necesidades actuales, considerando también, las posibilidades de expansión del equipo de laboratorio en un futuro.

El sistema a diseñarse y desarrollarse involucra la programación de un paquete computacional apto para el seguimiento y operación de los diferentes lazos de control, que han de ser supervisados a través de una estación central. La aplicación será escrita en lenguaje Visual BASIC, versión 3.0 Professional, y orientada para su utilización bajo las normas del entorno Microsoft WINDOWS 3.1 o superiores. Estos programas se escribirán en forma modular y universal, de modo que sea factible cualquier mejora y afinamiento, de presentarse la oportunidad para realizar nuevas versiones de los mismos.

Por otra parte, el proyecto incluye la integración de diversos bloques o sistemas existentes en laboratorio, que hasta la presente fecha tienen escasa o ninguna interrelación entre sí, desprovechando de esta manera, la mayor parte de sus potencialidades educativas. Entre estos, se hallan prototipos de sistemas reales, equipo digital de adquisición y transmisión de datos, controladores digitales, y computadores personales. Gran parte de este equipo es

fruto de anteriores trabajos de investigación, pero con cierta adecuación y acondicionamiento, se obtendrá un gran rédito académico.

Adicionalmente, han de desarrollarse modelos computacionales de plantas industriales de elevada complejidad, los cuales, si bien están desviados de la temática del control supervisor como tal, son obligadamente necesarios para la comprobación del funcionamiento del sistema *SCADA* en el verdadero límite de sus capacidades.

Naturalmente, tras la conformación e implementación en laboratorio del sistema en cuestión, se procederá a la experimentación y consiguiente validación de resultados, dentro de las limitaciones establecidas. De este modo, será factible extrapolar los conceptos y procedimientos utilizados en el desarrollo del presente trabajo hacia la construcción de sistemas de escala industrial.

Así, con la unificación del equipo de laboratorio y la programación de un paquete computacional inteligente se pretende introducir las más innovadoras técnicas de control moderno en los programas de estudio.

El vertiginoso crecimiento de las diferentes disciplinas tecnológicas, entre ellas las concernientes al control de sistemas, ha propiciado la incorporación de pequeños lazos de control a estructuras automatizadas cada vez mayores; y, por supuesto, más complicadas. Este proceso de expansión llega a su máxima dimensión en los sistemas *SCADA* (*Supervisory Control And Data Acquisition*), en los cuales diversos procesos, distribuidos en una extensa zona geográfica, son interconectados, programados y supervisados por computadores pertenecientes a una amplia red de comunicaciones. Estos computadores están a su vez subordinados a estaciones matrices conformando una estructura jerárquica. La conformación de un sistema de control con este tipo de arquitectura permite el acceso inmediato desde cualquier terminal de operación a cualquier proceso en localización remota. Esta ha sido precisamente la clave para el desarrollo de industrias, compañías y

consorcios multinacionales de trascendental importancia. La introducción de este tipo de tecnología en nuestro país es ya una realidad, y se considera oportuno que la Escuela Politécnica Nacional preparare profesionales con sólidos conocimientos en sistemas *SCADA*, que puedan ofrecer sus servicios a compañías interesadas y contribuir de forma directa al desarrollo del país.

1.2. EVOLUCION DE LOS SISTEMAS DE CONTROL DIGITAL.

En las últimas décadas el asombroso desarrollo de la tecnología del estado sólido ha hecho inevitable que ésta sea tomada en cuenta en el diseño e implementación de sistemas de control, los cuales afrontan cada vez, problemáticas más complicadas debido a la creciente demanda de dispositivos pseudo inteligentes que participen en procesos industriales.

El primer obstáculo constituyó la elaboración de una teoría, congruente con la teoría de control desarrollada hasta ese entonces, que permitiera el diseño y evaluación de sistemas que contengan elementos de operación digital. La participación de la matemática fue en este punto algo más que decisiva. Fueron retomados y analizados varios conceptos y teoremas que desembocaron en la formulación de dos métodos para el tratamiento efectivo de variables discretas, estos son: la transformada Z y las variables de estado.

A la par fueron implementados controladores digitales, que conforme a la consistencia de la teoría y al afinamiento de la tecnología, fueron sustituyendo poco a poco a los sistemas clásicos de control, salvando los casos en los que los méritos de los elementos analógicos no pudieron ser superados.

El corazón de esta nueva generación de controladores es el microprocesador, cuya aparición en los ámbitos tecnológicos, a fines de los años 60, proyectó el control digital a una nueva dimensión; dimensión en la cual la versatilidad, confiabilidad y velocidad del

elemento, multiplican las posibilidades de éxito cuando se pretende ejercer acciones de seguimiento y regulación en variables de proceso; todo esto, con costos de montaje, operación y mantenimiento sorprendentemente bajos. A partir de aquí, la meta fue adaptar el problema de control a la filosofía de los microprocesadores, y gracias a la sólida base matemática existente, este objetivo fue alcanzado poco tiempo después del lanzamiento del microprocesador al mercado. [13]

Camino a la automatización industrial, fueron concebidas diversas estrategias que han permitido afrontar con suficiente solvencia la manipulación de variables de proceso bajo la óptica del control digital. Así fueron desarrollados los siguientes esquemas:

- Control digital directo
- Control secuencial
- Control distribuido
- Control con instrumentación inteligente
- Control supervisor

Las constantes innovaciones en el campo científico y tecnológico han perfeccionado el control computarizado a tal punto, que hoy en día, su aplicación en la mayoría de procesos industriales es prácticamente una norma.

1.2.1. Control digital directo.

Al extenderse la participación de los controladores digitales en procesos industriales se hizo general el diagrama de control indicado en la figura 1.1. Se puede observar que la variable de salida del proceso, es decir, la señal a controlarse, pasa por una fase de acondicionamiento y conversión anlógica-digital. El producto de esta acción es comparado con una señal patrón, fijada directamente por un operador, o indirectamente a través de algún proceso diferente. La magnitud de esta diferencia será tomada en cuenta

por el computador para realizar la acción de control pertinente, y de acuerdo a ésta, emitir una señal digital de salida adecuada para la corrección del error. La respuesta del computador deberá revertirse al proceso, pasando primeramente por un conversor digital-analógico, un amplificador de potencia y un actuador apropiado. [13]

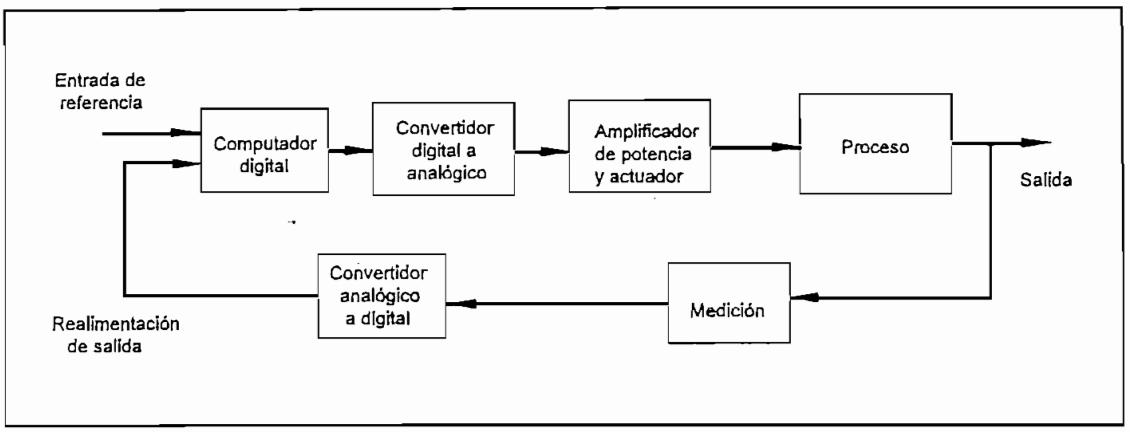


Figura 1.1. Diagrama de bloques de un sistema de control por computador.

Cabe advertir que el principio de *realimentación negativa*, usado en sistemas analógicos de control, permanece invariante ante la incorporación de un computador digital, como parte fundamental del mecanismo de control, es decir, visualizados en forma global, los resultados de la realimentación son igualmente beneficiosos tanto en un caso como en el otro. La diferencia se da sustancialmente en el hecho de que la comparación entre el valor de referencia y el valor de salida no es contínua, como en los sistemas analógicos, sino que es discreta en el tiempo. Es decir, aún cuando la salida es permanentemente sensada, la conversión analógica-digital y el procesamiento de datos por parte del computador digital, son actividades desarrolladas en forma periódica. El tiempo que transcurre entre una medición *digital* de la variable de salida y la siguiente, se conoce como período de muestreo. Este es un parámetro de vital importancia, puesto que incide directamente en la estabilidad del sistema y en la calidad de respuesta que se puede obtener a la salida del proceso. Otra diferencia, pero sin duda de menor importancia que la primera, es que los controladores digitales tienen una exactitud limitada por la resolución de los convertidores analógico-digitales; es decir, el computador solo podrá trabajar con

las cifras significativas que le permita el número de bits por palabra establecidos en el sistema digital.

El control analógico convencional usa generalmente un algoritmo PID (*proporcional, integral y derivativo*) para compensar eficazmente la diferencia entre variable de salida y entrada referencial de un determinado proceso. La siguiente ecuación hace explícita esta ley de control:

$$u(t) = K_p \cdot \left(e(t) + \frac{1}{T_i} \cdot \int e(t)dt + T_d \cdot \frac{de(t)}{dt} \right)$$

Notar que $e(t) = r(t)-y(t)$ es el error, o diferencia entre magnitud de salida $y(t)$ y el nivel de consigna $r(t)$, K_P ganacia del compensador, T_i tiempo integral y T_d tiempo derivativo. Esta ecuación o algunas variaciones ligeras de la misma, son universalmente utilizadas con controles analógicos; y, su equivalente discreto ofrece muy buenos resultados al ser implementado en un control digital. Sin entrar en demostraciones, una adecuada aproximación en tiempo discreto de la ecuación de control indicada anteriormente puede ser:

$$u(k) = u(k-1) + K_p\left(1 + \frac{T}{2T_i} + \frac{T_d}{T}\right)e(k) + K_p\left(-1 + \frac{T}{2T_i} + \frac{2T_d}{T}\right)e(k-1) + K_p\frac{T_d}{T}e(k-2)$$

Donde T es el período de muestreo. [24]

Programar un microprocesador o microcontrolador para que calcule una señal digital de actuación en base a esta función es extremadamente simple, lo que quiere decir que el control digital directo de un lazo puede implementarse sin mayor complicación utilizando un micro elemento. La amplicación de este esquema a sistemas multilazo, o la

interconección de sistemas haciendo que las salidas de unos sean el *set point* de otros, puede realizarse con dispositivos electrónicos de mayor capacidad, programados con lenguajes de alto nivel, como es el caso de los computadores personales. Más adelante se hará referencia a este tipo de estructuras al hablar del control distribuido.

1.2.2. Control secuencial.

Sin duda alguna, en un proceso industrial, el control contínuo de las magnitudes físicas que puedan ser de interés, es una acción de importancia vital, pero igualmente lo es, la ejecución secuencial de un conjunto de tareas, que deberán activarse cuando el proceso haya llegado a un estado determinado y/o cumpla con ciertas condiciones, ya sean éstas establecidas por el operador o por la naturaleza misma del proceso. La constante comparación de la situación de un sistema, planta o proceso con un grupo de condiciones referenciales y la consiguiente toma de decisiones a partir de la validación de las mismas, son labores que entran en el dominio del control sequencial.

Son realmente numerosas las actividades industriales que requieren ser desarrolladas como un conjunto sucesivo de etapas o sub procesos, y es muy probable que, para responder adecuadamente a las exigencias de los complicados procesos industriales de hoy, la transición entre estas distintas instancias se dé en forma suave, automática y sincronizada. Nuevamente aquí, el elemento ideal para cumplir con estas operaciones es el microprocesador. Así, la norma que permitiría a los fabricantes de circuitos integrados concebir y desarrollar la arquitectura de un elemento semejante, o dicho de otra manera, el concepto original bajo el cual se habrían de diseñar los microprocesadores fue, y aún lo es, conseguir un dispositivo electrónico miniaturizado que sea capaz de seguir fielmente un conjunto de instrucciones, tomar decisiones booleanas y realizar cálculos matemáticos. De esta premisa se sigue que la razón de ser de los microprocesadores es el control secuencial, el cual, al ser enfocado exclusivamente hacia el cálculo de una ley de control que actúe sobre un sistema en lazo cerrado, toma la forma de control digital directo.

Bajo la perspectiva del control sequencial, un proceso real debe dividirse en tantas fases como sean necesarias, las cuales deben llevarse a cabo, unas en forma serial, otras en forma simultánea, de acuerdo a una programación en tiempo de las mismas. La figura 1.2. ilustra el desarrollo temporal de un proceso hipotético, explicando así la situación de secuencialidad.

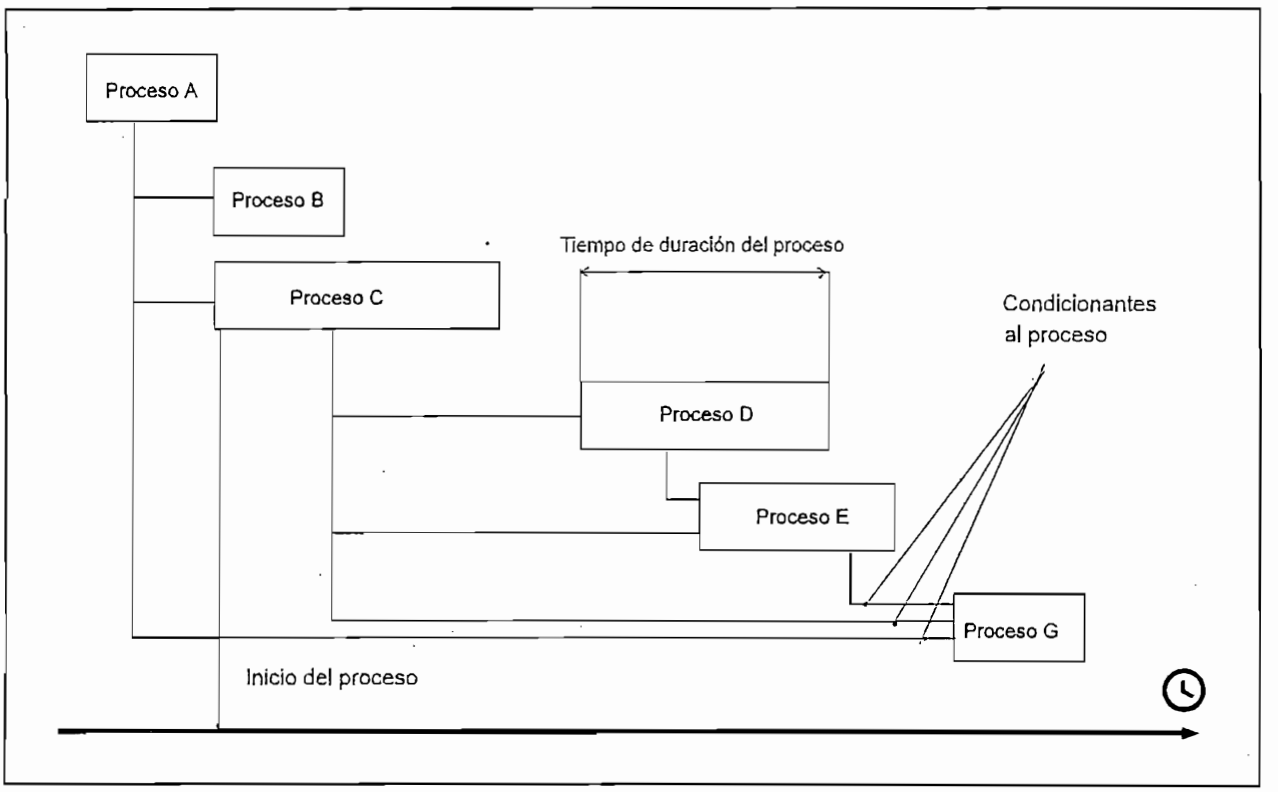


Figura 1.2. Diagrama de secuencialidad de un proceso.

Para dar por concluida una etapa y pasar a la siguiente, se debe verificar que en el proceso se hayan cumplido las condiciones necesarias y suficientes que permitan dicha transición. Estos filtros condicionantes pueden ser de todo tipo, pero es usual encontrar, de acuerdo a las necesidades generalizadas de la industria, que éstos sean intervalos de tiempo preestablecidos, invariancia de las variables de salida alrededor de un punto de equilibrio, órdenes directas del operador, cumplimiento de secuencias de menor categoría, etc. Con los adecuados métodos de sensado y las apropiadas interfaces para el microcontrolador,

todo esto y más, puede ser resuelto con un programa computacional descomplicado.

La forma más difundida de ejercer control sobre la secuencialidad de un proceso recibe el nombre de *"check before operate"* (verificar antes de actuar), y consta de tres partes. Primeramente se debe realizar una inspección de las variables involucradas, tanto las de entrada como las de salida del sistema. Este examen debe, en lo posible, ser de muy corta duración, con el fin de que los datos adquiridos reflejen la situación *instantánea* del proceso, evitando así la aparición de errores por concepto de retardo. Acto seguido debe procederse a la valoración y comparación de las variables ingresadas con los patrones referenciales establecidos por el ingeniero diseñador y conocedor del proceso. Para esta etapa es indispensable la existencia de un programa, preferiblemente llamado instructivo, el cual debe indicar claramente las acciones a tomarse cuando el proceso haya llegado a un punto decisivo. Finalmente, y de acuerdo a lo que estipule el programa anteriormente citado, se han de alterar intencionalmente las variables de salida, con el fin de llevar al proceso a las etapas subsiguientes. Estas acciones deberán ejecutarse periódicamente conforme el proceso evolucione, y se lo hará con una frecuencia razonablemente alta, respecto a las constantes de tiempo involucradas, de modo que el sistema no entre en regiones de inestabilidad o de impredictibilidad. [48]

En la actualidad, los procedimientos de medición, comparación, confirmación y actuación, son realizados por los *controladores lógicos programables*, sistemas de control, conocidos más frecuentemente con el nombre de PLC's. Estos equipos se han vuelto increiblemente populares en el ámbito industrial, debido a su versatilidad, a su gran capacidad de maniobrar sobre complicados procesos automáticos, e indiscutiblemente, debido a los costos relativamente bajos de instalación, operación y mantenimiento.

1.2.3. Control distribuido.

Con frecuencia, al realizar el análisis teórico de un sistema a gran escala, se suele suponer,

por simplicidad, que dicho ente se encuentra formado por una combinación de pequeños subsistemas, interconectados entre sí, operando simultáneamente en tareas específicas, pero con un fin común. Este artificio de la lógica, inicialmente de validez intuitiva, pero hoy justificado plenamente por los axiomas bajo los cuales se sustenta la teoría de control, no solo es aplicable a estudios teóricos, sino que puede hacerse extensible al *acoplamiento real* de subsistemas que consoliden una estructura más grande. ¿Cómo es posible esto? Por una vez más, gracias a los microprocesadores.

Se puede obtener una idea más clara de lo que significa el control distribuido si se visualiza una planta y sus componenetes, en las figura 1.3.a y 1.3.b:

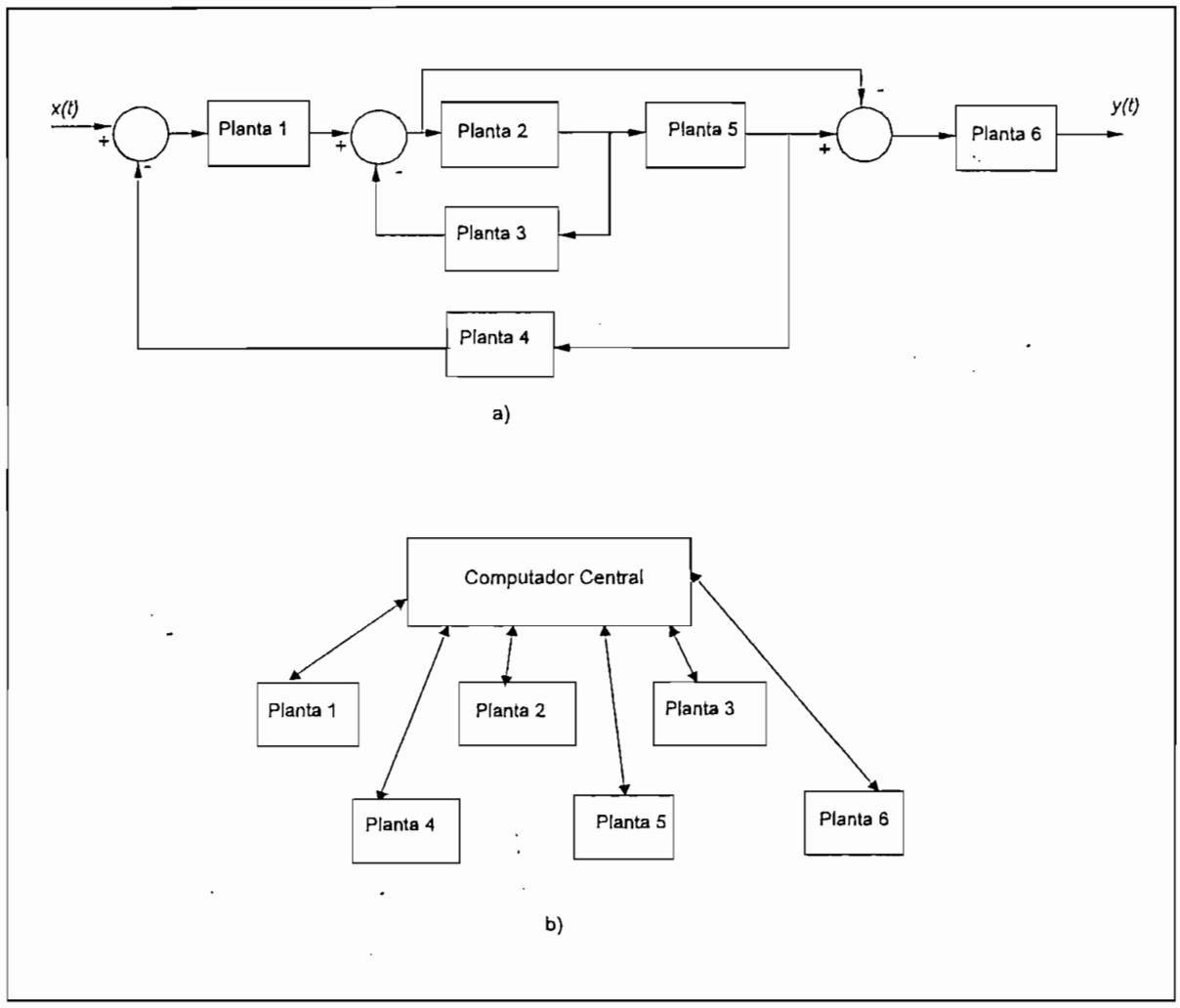


Figura 1.3. a) Interconexión normal de un sistema. b) Interconexión en una estructura distribuida

La hipotética planta, a la que se le atribuye esta topología , bien podría tener sus componentes interconectados directamente, utilizando para ello, los sensores y actuadores apropiados (Figura 1.3.a.); pero la concepción actual intentaría convertir a este sistema en una estructura de control distribuido (Figura 1.3.b). Esto significa, que los subsistemas serían enlazados a través de un computador central, como si este fuera una interface universal que permitiera la interacción de todos los lazos del proceso, a través del *acoplamiento digital* de los módulos que constituyen la planta original [30]. Por supuesto, para que esto sea posible, se requiere que cada uno de los subsistemas posea un microprocesador, o a lo menos, un circuito integrado dedicado, que facilite la interacción del dispositivo particular con el computador, y por medio de él, con el resto del sistema.

Nótese también, que al configurar una planta como un conjunto de sistemas distribuidos, se crea necesariamente una estructura jerárquica de dos niveles de control: primero, el que propiamente, y que de forma individual, se ejerce en el lugar de cada proceso por medio de un microprocesador; y, en segundo término, el que actúa coordinando las operaciones entre ellos, generalmente llevado a cabo por un computador de potencia superior. Si el proceso o la planta son de mayor complejidad, se podría incrementar el número de niveles en la estructura jerárquica del sistema, en cuyo caso, conforme se ascienda de categoría, se crecerá en capacidad y autoridad, pero se disminuirá en especificidad. Más detalles sobre la organización jerárquica de un sistema se discutirán en el apartado correspondiente a los sistemas de control supervisor, apartado 1.2.5.

Por todo lo mencionado hasta este punto, se puede concluir que la innovadora idea del control distribuido conlleva las siguientes consecuencias: aumento de la confiabilidad global del sistema, incremento de rendimiento y productividad del proceso, facilidad en el mantenimiento y la actualización de equipos, modularidad en la arquitectura de la planta, y, lo que generalmente es de mayor ponderación, la reducción de costos al suponerse un ahorro en los antiguos elementos, usualmente de tipo analógico, para la interconección de sistemas.

1.2.4. Control con instrumentación inteligente.

Conforme la pendiente del desarrollo tecnológico aumenta, las necesidades de la sociedad moderna se extralimitan, y consecuentemente, los proyectos científicos y los procesos industriales orientados a satisfacer las mismas, se vuelven cada vez más complicados. El momento en el que los expertos en ingeniería de control propusieron el esquema de sistemas distribuidos como la más sensata solución a la complejidad de los sistemas reales, se presentó el inconveniente de que los elementos convensionales de medición y transmisión de magnitudes físicas, no eran lo suficientemente compatibles, ni flexibles con la estrategia planteada. Por esta razón, se dió énfasis a la *instrumentación inteligente*, entendida ésta, no como una nueva modalidad de control, sino como un poderoso complemento para el monitoreo de un exigente proceso.

En lo que concierne a esta nueva generación de dispositivos de instrumentación, se pueden mencionar los siguientes aspectos:

En primer lugar, permanece intacto el elemento primario del sensor, y con probabilidad también, el transductor *variable de proceso - voltaje.* No sucede lo mismo con el acondicionador de señal, que de acuerdo a este tipo de tecnología, deberá incluir un conversor analógico-digital de resolución suficientemente fina y filtros digitales con características apropiadas para la variable a examinarse. Por otro lado, es menester implementar un procedimiento de compensación, linearización y limitación de la señal proveniente de la etapa primaria del instrumento, usando para ello, una circuitería digital exclusiva, o bien, una subrutina escrita en lenguaje para microprocesador. Finalmente, el equipo no se consideraría completo si no incluyere dos etapas adicionales, una de ellas asignada a la modulación de señales, y la otra dedicada a la transmisión de las mismas. Se sobreentiende, que estas dos últimas instancias encontrarán su dual en el lado de recepción de información, configurando así, un sistema de medición y transmisión completo, como el expuesto en la figura 1.4: [12]

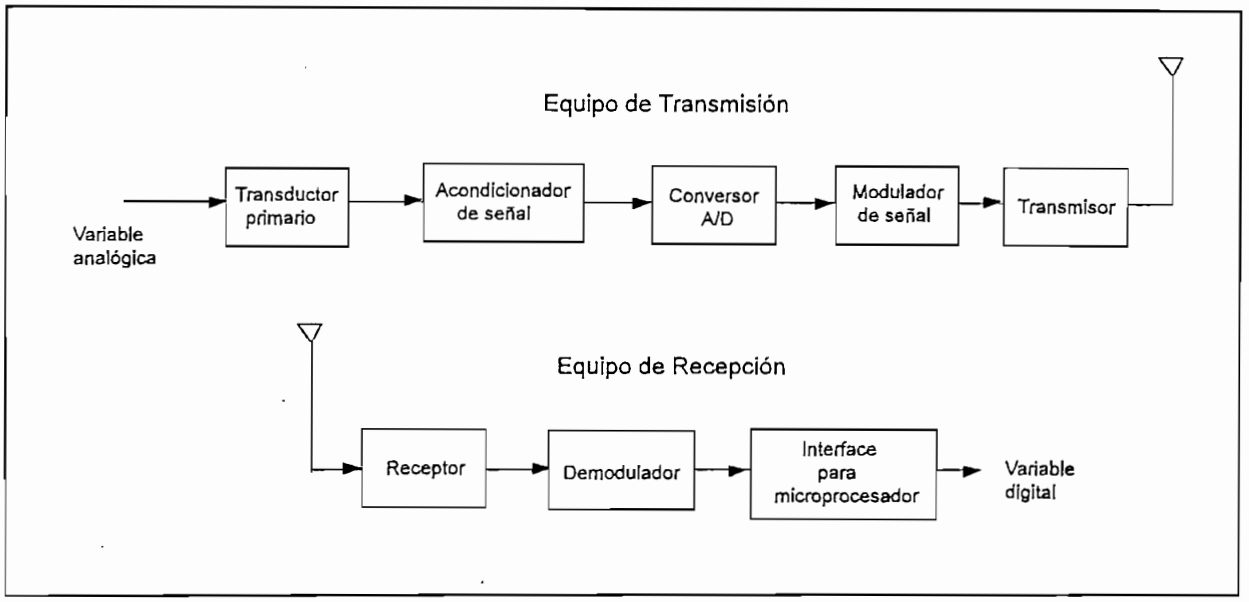


Figura 1.4. Diagrama de bloques de un equipo de telemetría.

En cierto tipo de sensores, especialmente en aquellos de localización remota, además de las características generales señaladas, se requiere incluir funciones como: auto diagnóstico y auto calibración del instrumento, detección por aproximación estadística el valor medido verdadero, señalización y/o registro *in situ* de las variables supervisadas, y muchas otras. Igualmente, los circuitos de modulación y transmisión, podrán ser tan simples como los que sugieren las normas de comunicación serial y paralela entre computadores, o bien, tan complicados como los que se llevan a cabo para establecer comunicación vía micro onda; todo esto sujeto a las necesidades del proceso y a las posibilidades económicas.

Dada la complejidad de un sistema de medición y transmisión como el descrito, es altamente recomendable, mas no indispensable, la incorporación de un microcontrolador a su circuitería básica, bajo cuyo régimen se realicen todas las operaciones mencionadas. Nuevamente aquí, es el microprocesador o microcontrolador quien marca el paso en esta impresionante revolución tecnológica.

1.2.5. Control supervisor.

Consecuencia lógica de tener un sistema complejo dividido en subsistemas simples, donde cada uno de ellos ejecuta labores específicas y pertenece a un particular nivel de injerencia ante el funcionamiento global del sistema, es la formulación de una nueva táctica de control, conocida con el nombre de control supervisor. De acuerdo a lo discutido en numerales anteriores, no es posible hablar de sistemas de control distribuido sin definir por lo menos un ente coordinador y decisorio al cual queden subordinados todos los lazos del proceso, y por tal razón, se propicia una concatenación inmediata y evidente entre los sistemas de estructura jerárquica y distribuida con los sistemas de mando supervisor.

No existe hasta el momento, una definición estricta y única de control supervisor, ésta varía de autor en autor, de acuerdo a la magnitud y alcance de la aplicación, a la cual van dirigido sus particulares estudios; sin embargo, todos coincidirán en la noción fundamental de disponer de un equipo y de un procedimiento que permita la *vigilancia* de unidades remotas desde un punto de observación cualquiera, perteneciente a una categoría jerárquica lo suficientemente decisiva. [31] [3]

En palabras más sencillas, los sistemas supervisores se diseñan con el fin de ejecutar todas las funciones que convencionalmente las realiza el individuo operador, como son: verificación de la estabilidad de los procesos y alerta ante su funcionamiento fuera de los límites normalmente permisibles, ajuste del punto de operación ante perturbaciones aleatorias imprevistas, diagnóstico de las unidades en operación, recopilación permanente o periódica de la información relativa a las variables de interés, con la consiguiente organización y presentación de la misma, desplegadas eficientemente en cada nivel jerárquico, de acuerdo a las necesidades y funciones del mismo.

Por tales razones, los sistemas supervisores deben: estar dotados de una inteligencia virtual que les brinde una elevada capacidad de decisión, conformar una red de comunicaciones lo

suficientemente amplia que les facilite establecer contacto con todos los subsistemas; y por su puesto, permitírseles el acceso a toda la información requerida para llevar adelante su tarea.

Los detalles de cómo un sistema de mando supervisor realiza todas estas actividades, serán expuestos en el capítulo siguiente, *Sistemas SCADA*, pero se puede anticipar un breve diagrama (Figura 1.5) que ilustre la configuración jerárquica de los mismos e identifique brevemente su límite de actividades.

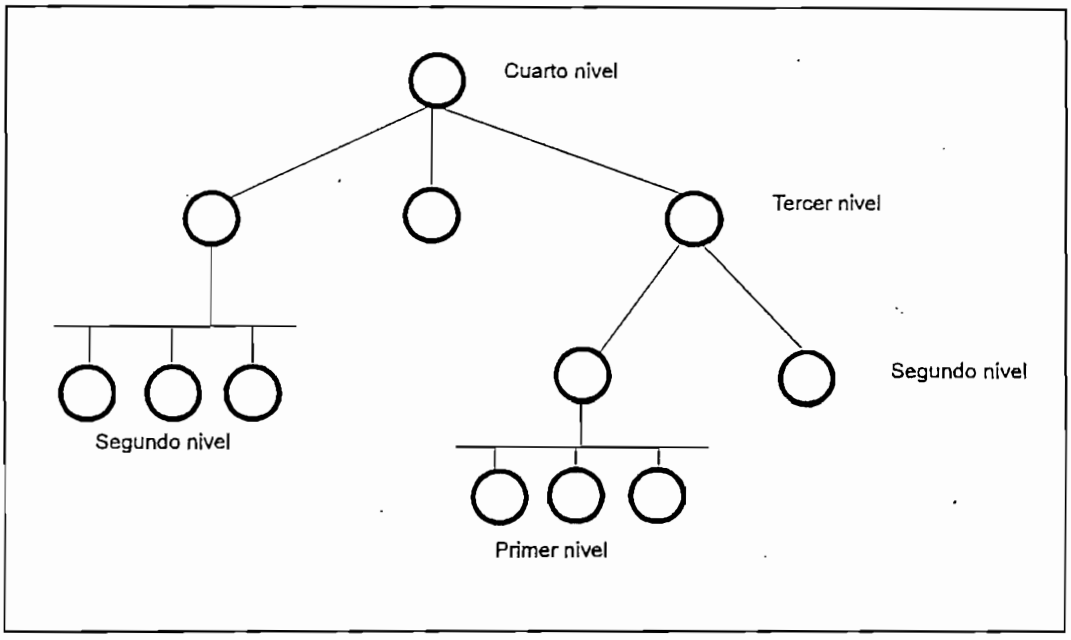


Figura 1.5. Estructura jerarquizada de un Sistema Supervisor de cuatro niveles.

Con esta síntesis acerca de la evolución tecnológica de los sistemas digitales de control, la cual, puede considerarse como una de las más connotadas de este siglo, ha sido factible concentrar paulatinamente la atención hacia el tema medular del presente trabajo de investigación: los sistemas de control supervisor con adquisición inteligente de datos, o sistemas *SCADA*. Aunque se considera necesario conocer y entender las instancias previas a la aparición de sistemas *SCADA* en ambientes industriales, no es pertinente al presente estudio, el análisis detallado y el tratamiento de dichos temas.

CAPITULO 2

SISTEMAS *SCADA*

2.1. DESCRIPCION GENERAL DE UN SISTEMA *SCADA*.

2.2. CONFIGURACION DE UN SISTEMA *SCADA*.

2.3. ADQUISICION, TRANSMISION Y ACTUACION INTELIGENTE.

2.4. SUPERVISION DE EVENTOS.

2.5. CONTROL DIGITAL BAJO EL ESQUEMA *SCADA*.

2.1. DESCRIPCION GENERAL DE UN SISTEMA *SCADA*.

En la sección previa se hizo una rápida introducción a los sistemas *SCADA* mencionando brevemente cómo los sistemas de control digital se transformaron en las complicadas plataformas supervisoras que hoy en día inundan la industria. Sin embargo, no se hizo más que una ligera reseña de la historia tecnológica de la segunda mitad de este siglo, pues debe quedar en claro que el desarrollo ha sido vertiginoso no solo en la dirección de las técnicas de control digital, sino en todas las disciplinas que con ella se relacionan, como son, sistemas de comunicación, redes de información, computación y programación orientada a objetivos, aplicaciones de inteligencia artificial, robótica y cibernética, por mencionar las principales. Es un hecho universalmente admitido, que más del cincuenta por ciento del conocimiento humano ha sido adquirido en el transcurso de estas últimas décadas.

Conocidas pues, las razones tecnológicas y económicas que han propiciado el surgimiento de los sistemas *SCADA*, se puede proseguir con una definición y descripción genérica de los mismos.

2.1.1. Definición.

SCADA System es la abreviación de ***Supervisory Control And Data Acquisition System,*** que en castellano significa Sistema de Control Supervisor y de Adquisición de Datos. Esta frase brinda ya una clara noción de lo que es y de lo que realiza un sistema de este tipo. Sin embargo, para el propósito de este trabajo, se considera conveniente profundizar esta definición y enunciar formalmente que una estructura *SCADA* es una forma de control, básicamente de tipo supervisor, orientada a plantas multiproceso, cuyos lazos de control, operantes bajo un contexto jerárquico, tienen localización remota y/o local. Adicionalmente, se debe recalcar que la acción supervisora está estrechamente ligada a un mecanismo de adquisición inteligente de datos, que permite no solo el fiel

seguimiento de las magnitudes de interés del sistema, sino que facilita la organización de dicha información, con fines de registro, control maestro y proyección estadística. [26] [57]

Para entender de mejor forma la anterior sentencia, ha de examinarse por separado cada una de sus proposiciones:

"Forma de control de tipo supervisor". Considerando el tipo de plantas a las cuales van destinadas las acciones de control de un sistema *SCADA*, es decir, plantas de extensa y complicada topología, puede asumirse que éste debe tener fundamentalmente un caracter supervisor. Razón por la cual, ha de compartir todas las características y modalidades de operación, propias de un sistema de control supervisor, como las descritas en el capítulo introductorio y detalladas más adelante en este mismo capítulo.

"Orientada a plantas multiproceso, con estructura jerárquica y distribuida". De esta sencilla explicación son perfectamente identificables tres bloques característicos de los sistemas *SCADA*: primeramente, los equipos de control digital que interactúan directamente cón los procesos en cuestión, también llamados Unidades Terminales Remotas, en segundo lugar, los equipos de control que conforman la estructura supervisora en sus diferentes niveles, conocidos más frecuentemente como Estaciones Maestras y Subestaciones; y por último, los equipos de comunicación que unen los dos bloques anteriores. El detalle de estos grupos funcionales será dado a conocer en las secciones siguientes, por el momento, lo que debe quedar establecido, es que la utilización de estos componentes en una estructura de diseño coherente, es lo que hace a un sistema *SCADA* tan idóneo y apetecible para la gobernación de procesos industriales complicados.

"Adquisición inteligente de datos". Normalmente, las acciones de monitoreo y vigilancia, intrínsecas a los sistemas supervisores, no pueden considerarse eficientes si éstas no se asocian a un procedimiento de adquisición de datos lo suficientemente refinado que permita no solo la extracción de datos en forma bruta, sino también operaciones como la

corrección de errores de medición, obtención de valores medios, índices de repetibilidad, factores de correlación, evaluación de tendencias; y en lo posible, predicción del estado futuro de un sistema. Cabe señalar, que en los sistemas *SCADA*, algunas de estas funciones se realizan en los mismos dispositivos de instrumentación, los cuales, con la tecnología de nuestros días, se comportan como verdaderos componentes pseudo inteligentes; sin embargo, el resto de funciones, en especial aquellas que incluyen difíciles rutinas matemáticas para la manipulación y filtración de información, son implementadas a nivel de software, logrando así ser tan exactas y precisas como se requieran, con solo un ingenioso programa.

"Con fines de registro, control maestro y proyección estadística". He aquí un resumen de las funciones globales de un sistema *SCADA*. Es precisamente por esta versatilidad, que dichos sistemas de control son ahora tan populares, ya que de una manera unificada y simultánea afrontan todos los aspectos de la problemática del control.

2.1.2. Ventajas de la aplicación de sistemas de control *SCADA*.

Por todo lo mencionado anteriormente, un sistema *SCADA* se reconoce por su capacidad de seguir y regular, desde el principio hasta el fin, el desarrollo de un proceso industrial por complicado que sea. Las bondades de esta estrategia de control ayudan a los individuos operadores a observar de cerca todos los aspectos de la planta desde cualquier punto de la misma. Para ello solo es necesario tener acceso a un computador personal que esté enlazado al sistema. Nótese, que tan solo esta característica es suficiente para producir una optimización de los recursos en la planta, con su consiguiente reducción de costos en el rubro de gastos de operación.

Adicionalmente, bajo una modalidad de control como esta, se pueden incluir programas para la planificación de producción, mantenimiento preventivo, control de calidad, seguridad industrial, por mencionar los más usuales. Estas directrices harán encaminar al

proceso en cuestión a la siempre ambicionada meta de la calidad total.

Finalmente, puesto que un sistema *SCADA* es diseñado con arquitectura abierta, de forma modular y flexible, no se restringe la posibilidad de expansión de la planta o del proceso, sino por el contrario, las factibiliza. Esta expansión puede ser entendida, ya sea como una incursión en nuevas líneas de producción que soliciten la atención de un control completo, o bien, como la necesidad de reorganizar y actualizar los equipos de una rama de producción ya existente, que pretende satisfacer las necesidades cambiantes de una sociedad.

2.1.3. Implementación de sistemas *SCADA* en el país.

Actualmente existen compañías de categoría internacional diseñadoras e implementadoras de sistemas *SCADA*, las cuales, con su indiscutible experiencia y reconocido éxito, han ido imponiendo poco a poco sus estándares ingenieriles en el ámbito industrial, entre ellas, ABB Kent-Taylor, Ferranti International Controls, Landys & Gyr, SR Telecom, etc. La participación de empresas como estas, en proyectos industriales nacionales de gran envergadura, ha ido incrementandose paulatinamente conforme el Ecuador se ha visto irremediablemente forzado a afrontar las necesidades de una industria cambiante, cada vez más sofisticada y contagiada del modernismo tecnológico.

De hecho, en la actualidad, se han incorporado sistemas de control de tipo *SCADA* en varios proyectos nacionales como: CENACE (Centro Nacional de Control de Energía), desarrollado para el control de la operación del Sistema Nacional Interconectado [26] [1]; Sistema SEDIS (System-Europe DISpatch), orientado al control de distribución de carga eléctrica en Qutio [61][14]; Proyecto Mica-Quito Sur, implementado para la distribución de agua potable; entre los más sobresalientes [57]. Proyectos de automatización de menor calibre, como el control de hornos batch en la empresa EDESA, o el control inteligente del edificio matriz del Banco del Pichincha y Diners Club, también han recurrido a la

estructura *SCADA* para su implementación [36].

Con esto se resalta la importancia de este trabajo de tesis, puesto que con un sistema *SCADA* de carácter experimental, los conocimientos mínimos de esta revolucionaria técnica de control estarían al alcance de los futuros profesionales en la rama de Ingeniería de Control, a cargo de quienes estará el éxito de diversos proyectos de automatización industrial en el país.

2.2. CONFIGURACION DE UN SISTEMA *SCADA*.

2.2.1. Sistemas de control distribuido, origen de los sistemas *SCADA*.

El increible avance que han experimentado la teoría y la técnica del control digital en las últimas décadas, ha permitido intervenir y mejorar el desenvolvimiento de procesos industriales de gran magnitud; sin embargo, la obsolecencia y las restricciones de su primitiva infraestructura originaron serios conflictos en la implementación de un sistema de control satisfactorio. Estos inconvenientes fueron superándose conforme los expertos en la rama de control de procesos incursionaron en el campo de los sistemas de control distribuido. Finalmente, al reconocerse el éxito de esta novedosa forma de control, fueron aceptándose como norma industrial los diferentes conceptos, condiciones y criterios que dicha modalidad de control exigía. Así, en un tradicional sistema de control distribuido podían diferenciarse tres distintos entes operativos: los lazos de control, el organismo supervisor y los medios de intercomunicacion.

Los sistemas *SCADA* evolucionaron de los sistemas de control distribuido y ahora constituyen una herramienta mucho más poderosa y sofisticada que ellos, pero no por eso se sustentan en otros principios o realizan funciones radicalmente diferentes. Necesariamente deben contemplar los mismos aspectos y afrontar la misma problemática

en lo que se refiere a automatización, supervisión y comunicación. [26] [57]

2.2.2. Módulos operacionales en los sistemas *SCADA*.

2.2.2.1. Automatización básica. La automatización de base comprende el conjunto de sensores, transductores, controladores, registradores y actuadores que dialogan directamente con los distintos subprocesos que se pretenden gobernar. Normalmente, los equipos asociados a los lazos de control de un determinado subproceso se encuentran localizados en extrema cercanía al mismo y conforman lo que se conoce como Unidad Terminal Remota, o bien *RTU* (del inglés Remote Terminal Unit).

Estas unidades forman el primer nivel de control en toda estructura jerárquica, y su actividad es administrada por uno o más microprocesadores, de acuerdo con la necesidad. Cada *RTU* es totalmente independiente, no es frecuente ni recomendable que comparta tareas con otro *RTU*, pues esto implicaría una degradación en las características de autonomía y compactación del sistema de control. [63]

2.2.2.2. Plataforma de supervisión. La plataforma supervisora, por su parte, agrupa a los equipos que comandan la actividad de las Unidades Terminales Remotas. Así, se consolidan las Estaciones Maestras (*CSS*, Central Supervision Station) y Subestaciones de Observación (*SCS*, Surrounding Control Stations). En antagonismo con su nombre, la función de éstas pocas veces se orienta al control; usualmente solo se dedican a seguir de cerca el desempeño de las *RTU*, interrelacionar digitalmente dichas unidades de modo que las salidas de unas sean las entradas de otras, y mantener actualizados todos los registros del sistema. Sin embargo, de ser requerido, las estaciones y subestaciones pueden intervenir en el proceso con una acción concreta de control, como cambiar los valores de referencia (set points), ajustar parámetros en controladores, iniciar o interrumpir procesos secuenciales, entre muchas otras. Por tales razones, estas entidades, que no son más que computadores calificados que ejecutan un programa apropiado, constituyen los niveles

segundo, tercero y superiores, de una arquitectura de mando jerárquico. [63]

Con el fin de que el sistema global tenga un alto grado de confiabilidad y disponibilidad, suele considerarse, como una medida de precaución ante fallas impredecibles, la redundancia de equipos en puntos clave del mismo.

Si se trata de subestaciones, el dispositivo redundante es generalmente la Unidad Central de Procesamiento, o *CPU*, perteneciente al computador supervisor. El equipo auxiliar no entrará en funcionamiento hasta que un dispositivo de detección de errores solicite un reemplazo ante una falla de carácter *no restaurable*. [26]

En algunos diseños se busca que los *RTU* asociados a una subestación de observación sean tan autosuficientes como lo permita el sistema, de manera que, al producirse un desperfecto a nivel de subestación, dichas unidades puedan desacoplarse de la estructura supervisora sin incidir mayormente en la estabilidad de los lazos que controlan. Una vez superada la falla, se reincorporarían al organismo supervisor. Cabe indicar, que esta alternativa no es muy utilizada debido al costo que implica implementar unidades terminales remotas completamente auto abastecidas.

En el caso de estaciones maestras se prevee respaldo para la unidad de computación central, considerando un duplicado de su *CPU* y de todos sus periféricos. En estado normal de operación, la única función del computador auxiliar es mantenerse actualizado en la información proveniente del proceso mediante una línea de datos de alta velocidad que interconecta el par de computadores. De producirse una avería, el mando de las subestaciones se transfiere del computador en servicio al computador en espera, el cual, sin conflicto ha mantenido al día el material informativo. Esta transferencia se da mediante un dispositivo de conmutación al cual van conectadas las líneas de datos provenientes de las subestaciones, y que responde a una señal emitida por algún mecanismo de actuación ante fallas.

Finalmente, dependiendo de qué tan delicado sea el proceso que se controla, puede resultar indispensable incluir Fuentes de Alimentación Ininterrumpida (*UPS*, Uninterrupted Power Source) y/o grupos auxiliares para la generación de corriente alterna, y mantener así, los signos vitales del sistema.

2.2.2.3. Medios de comunicación. Este tercer bloque involucra el conjunto de vías que permiten el flujo de información entre unidades terminales remotas, subestaciones de observación y estaciones maestras. Evidentemente, este problema debe ser afrontado tanto desde el punto de vista de hardware como desde el punto de vista de software.

El sistema de comunicación que se utilice como soporte para un control *SCADA*, depende necesariamente de la magnitud del proyecto y de la configuración especifica que tenga la planta en cuestión.

Por ejemplo, si el sistema es relativamente simple, podría optarse por la comunicación serial o paralela entre computadores. Estas alternativas son suficientemente efectivas, confiables, y ante todo económicas, pues los equipos requeridos para hacerlas viables han sido increiblemente difundidos en el mercado y probados hasta el extremo.

Para sistemas de tamaño moderado, lo usual es configurar una red local, conocida también como red *LAN* (Local Area Network), la cual normalemente utiliza cable telefónico, cable coaxial o fibra óptica para la transmisión de información. Muchas compañías en la actualidad se dedican a la fabricación de equipos y programas que permiten la manipulación este tipo de redes, resulta entonces, muy propicio considerarlas en un sistema *SCADA*.

Para proyectos de vasta extensión, es preferida la incorporacion de redes *WAN* (Wide Area Network). Estos sistemas transmiten información vía microonda, que es el medio más apropiado en la actualidad. El avance tecnológico en esta rama de las

telecomunicaciones ha hecho que estos sistemas sean extremadamente confiables, pero en igual medida, extremadamente costosos, siendo solo recomendables cuando las distancias entre unidades remotas, subestaciones y estaciones superan los cientos de kilómetros.

En resumen, el sistema de comunicaciones a emplearse debe ser proyectado junto con la planta y el proceso en sí, no es técnico limitarse ni excederse en su dimensionamiento.

La temática de las comunicaciones, aún cuando sea solo enfocada para la interconexión de los distintos puntos de observación de un sistema *SCADA*, es exageradamente grande, razón por la cual, en este trabajo de investigación no se profundizará más en el tema, y se relegará dicho estudio a otros proyectos de tesis.

2.2.3. Alternativas configurativas para la implementación de un sistema *SCADA*.

Como ya se ha recalcado, lo más frecuente es que un control *SCADA* se diseñe con la intención de atender las operaciones de una planta industrial formada por un conjunto de subplantas que se hallan dispersas en una extensión geográfica considerable. Por tal razón, para garantizar la adecuada operación de los medios de control, es fundamental partir de una apropiada distribución y estructuración de los sistemas y subsistemas involucrados, y de las vías de comunicación que los enlazan.

Esta discusión debe darse en dos aspectos, primeramente en lo relativo a la interconexión física de los subsistemas y a la disposición de los canales de comunicación, y en segundo lugar, en lo referente a la dependencia jerárquica y lógica que debe existir entre los diferentes puntos de observación. No es lo mismo que una subestación supervisora utilice a otra solo como un nodo de tránsito de información, desde y hacia otros lugares, a que pueda tener acceso y dominio sobre los datos que manipula esta última. [5] [7]

En las décadas anteriores se han propuesto y desarrollado diversas configuraciones

topológicas que fácilmente se adaptan a cualquier necesidad en un sistema distribuido. Estas son:

- Interconexión Multidrop
- Interconexión radial y ramificada
- Interconexión en lazo unidireccional y bidireccional
- Interconexión en lazo parcial
- Interconexión total

2.2.3.1. Interconexión Multidrop. Es una de las configuraciones más simples, pues todas las estaciones comparten una única vía de transmisión, conocida como Data Highway, la cual puede ser un par telefónico, un cable coaxial, una fibra óptica, o algún otro medio, tal cual lo ilustra la figura 2.1. Esta singular característica implica una gran limitación: solo una estación puede enviar información a la vez, y consecuentemente se debe desarrollar un algoritmo que decida en favor de alguna estación cuando se presente un caso de congestion.

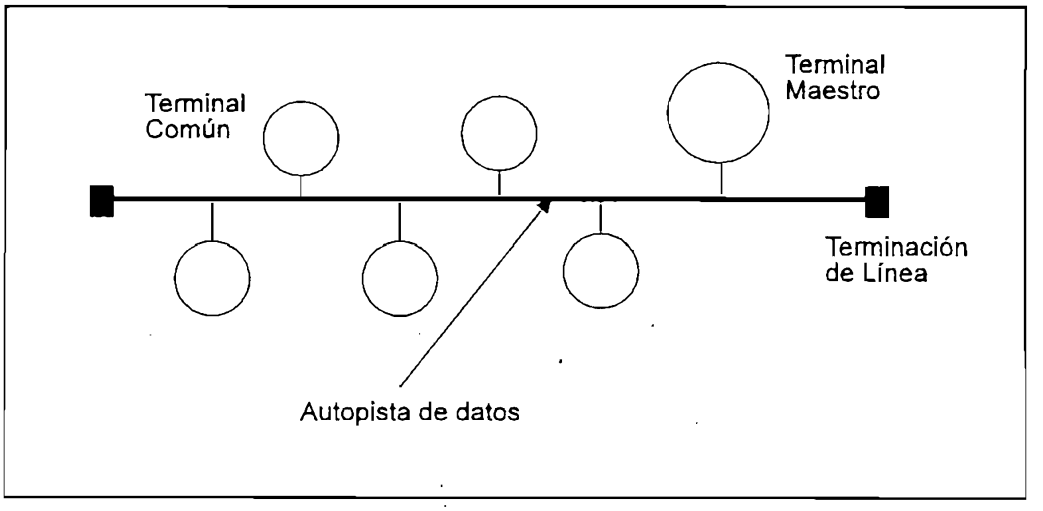


Figura 2.1. Interconexión Multidrop.

Ventajas:

- Al reducir el número de canales de comunicación, se incurrirá en una reducción de costos por conceptos de montaje o expansión.

- El medio de transmisión es pasivo, y por tanto, altamente confiable.
- Es un sistema fácilmente reconfigurable, tanto en lo que significa añadir o eliminar estaciones, como en lo que implica establecer cuál de ellas es la estación matriz.
- Los protocolos de comunicación son simples y eficaces.

Desventajas:

- Todas las comunicaciones fallan si el medio de transmisión se rompe en cualquier punto, y por tanto, es necesario duplicar el número de líneas.
- Las técnicas para evitar congestión son bastante complicadas.
- No es fácil implementar una transmisión Full Duplex con una sola línea de comunicación.
- Se requieren líneas de gran capacidad para poder sobrellevar toda la actividad comunicativa si el sistema es muy grande.
- Se requiere módulos terminales para mantener una baja impedancia en la línea.

2.2.3.2. Interconexión radial y ramificada. Esta es una estructura permite la conexión de cualquier estación perférica con otra, pero posiblemente solo con la participación de otras estaciones; situación que se encuentra explicada en la figura 2.2. A pesar de sus limitaciones es comúnmente utilizada en sistemas de control *SCADA*, pues pone de manifiesto la naturaleza jerárquica del mismo.

Ventajas:

- Relativamente económica, cada estación requiere una sola línea de comunicación para ser integrada al sistema.
- La subordinación de una estación a otra fija directamente el nivel de autoridad que le corresponde.
- La falla en una estación periférica no afecta en modo alguno a las comunicaciones en el resto del sistema.

Desventajas:

- La falla en un enlace aisla complentamente a la estación involucrada y todas las estaciones vinculadas con ella. Nótese que si la falla se da en el equipo de comunicaciones de la estación principal, todo el sistema se ve afectado, por ello, se requiere reforzar la confiabilidad del sistema incorporando equipo redundante.
- Suele darse problemas de congestionamiento en los nodos de tránsito.
- Las comunicaciones entre una estación de bajo nivel con una estación de muy alto nivel suelen tener retardos, por lo menos de un nodo de por medio.

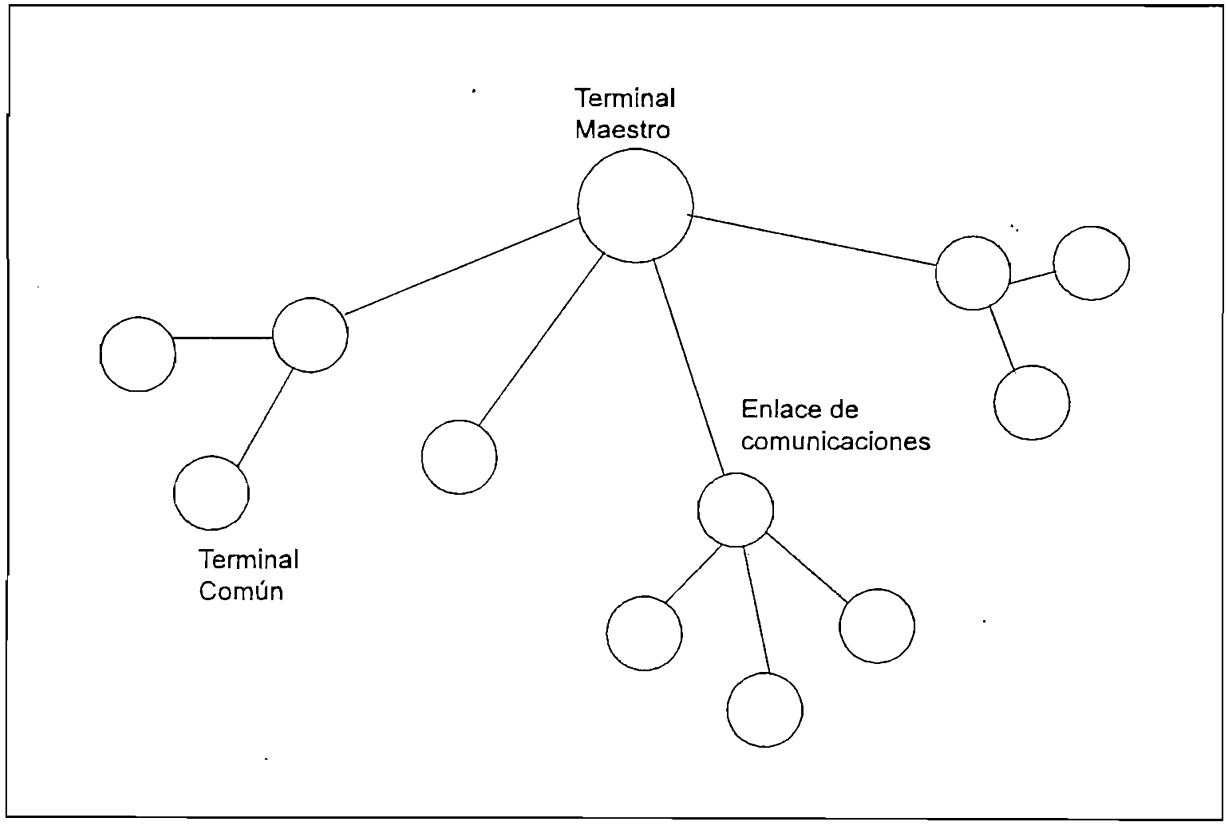


Figura 2.2. Interconexión radial y ramificada.

2.2.3.3. Interconexión en lazo unidireccional. Cada estación se encuentra enlazada con la estación vecina más cercana a través de una conexión simplex, de manera que se forma una red de estaciones en circuito cerrado, en la cual el flujo de información

2.2.3.4. Interconexión en lazo bidireccional. Topológicamente esta configuración es igual a la anterior, con la salvedad de que el enlace entre dos estaciones vecinas es de tipo full duplex. Ver la figura 2.3.b.

Ventajas:

- Puede implementarse en forma de dos lazos unidireccionales.
- De producirse una falla en una estación, no necesariamente se pierde el contacto con otras, ya que la información tiene una vía alternativa de acceso.

Desventajas:

- Se debe desarrollar un algoritmo que gobierne los enlaces, optimice el tráfico de información y prevenga al sistema de situaciones de congestionamiento.

2.2.3.5. Interconexión parcial. Esta es una configuración en la que se procura que cada estación esté enlazada directamente a otras dos por lo menos, garantizando que cualquier estación del sistema tenga acceso a otra a través de más de una ruta de enlace. Referirse a la figura 2.4.

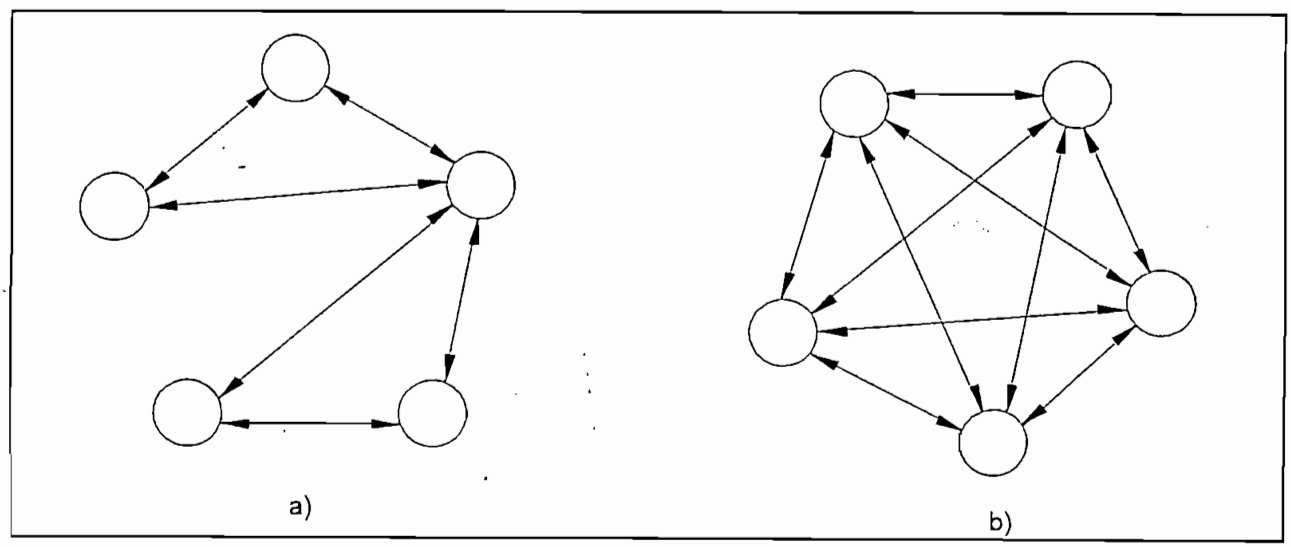


Figura 2.4 a) Interconexión Parcial, b) Interconexión Total.

Ventajas:

- Proporciona al sistema una alta integridad, ya una falla de comunicación en una estación cualquiera no significa el aislamiento de una sección o secciones del sistema.
- El paralelismo de vías de comunicación entre dos puntos del sistema incide positivamente en la velocidad de transmisión de información.
- Sistemas de comunicación de este tipo son muy flexibles y eficientes, pues permiten optimizar el tráfico y reducir el congestionamiento.
- Los costos de expansión son bajos, se requieren solo una o dos vías de comunicación para integrar una nueva estación al sistema.

Desventajas:

- Dependiendo de cuántos puntos intermedios hay entre dos estaciones que desean comunicarse, el retardo introducido puede ser grande.
- El software requerido para manejar con eficiencia las rutas de enlace suele ser extremadamente complicado.

2.2.3.6. Interconexión total. Bajo este esquema, cada estación del sistema está interconectada directamente a todas las demás estaciones, por medio de líneas dedicadas.

Ventajas:

- Tiene un altísimo grado de integración, no existe forma de que una estación quede aislada del resto aún cuando simultáneamente se produzcan varias obstrucciones en el sistema de comunicación.
- No hay problemas de retardo en las comunicaciones, puesto que el enlace entre las distintas estaciones es directo.

Desventajas:

- Extremadamente costoso, ya que para ***n*** estaciones se requieren $\frac{n(n+1)}{2}$ líneas full duplex, y ***n-1*** puertos en cada estación.

- Los costos de expansión también son elevados, ya que para integrar la estación número *m* deben instalarse *m-1* nuevas líneas de comunicación e incorporar un nuevo puerto en cada estación existente.

2.2.4. Selección de una configuración para un proyecto *SCADA*.

El análisis expuesto anteriormente sucita una gran controversia cuando se requiere elegir una configuración adecuada para un sistema distribuido. Entidades como *IEC* (International Electrotechnical Comission), *ANSI* (American Normalization and Standarization Institute) o *IEEE* (Institute of Electronic and Electrical Engineers) han propuesto guías y recomendado alternativas que permiten optar correctamente por una configuración apropiada para desarrollar sobre ella una aplicación *SCADA*. [5] [63] En resumen se tiene lo siguiente:

En primer lugar se deberá determinar cuán extenso es el proyecto, en magnitud territorial, y cuán dispersos están los posibles puntos de observación. A continuación se deberá verificar la posibilidad de que la información se difunda globalmente a todas las estaciones antes que de manera restringida a solo un grupo de ellas. Se exigirá también un análisis que determine qué puntos del sistema requieren duplicar el equipo de comunicación para casos de emergencia. Se buscará que cada estación cuente con más de un medio de acceso, y se desarrollarán diversas técnicas para el enrutamiento de información. Se considerará el medio apropiado de transmisión dependiendo, entre varias razones, de la frecuencia y el tráfico en las comunicaciones. A partir de esto, se podrá optar por el equipo que más se ajuste a las necesidades, y finalmente se diseñarán los programas computacionales que permitan la operación de dicho equipo.

Con estas breves directrices habrá de seleccionarse una de las configuraciones descritas anteriormente, o bien, una configuración particular que conjugue las características de dos o más de ellas.

2.3. ADQUISICION, TRANSMISION Y ACTUACION INTELIGENTE.

Una vez establecidas la topología del sistema y la modalidad para la transmisión de datos entre los diferentes nodos del mismo, ha de enfocarse la atención a los niveles de control primarios existentes en el sistema, es decir, ha de analizarse por separado la situación de cada unidad terminal remota sobre la cual comandará el sistema *SCADA*.

Como se ha dicho anteriormente, cada *RTU* es responsable de una sección del proceso global, y sobre ella reacáen las funciones de adquisición, transmisión, registro, control y actuación de cada lazo de control que gobierna. Consecuentemente, siendo la instrumentacion industrial la protagonista de estos sucesos, se considera apropiado dedicarle un espacio exclusivo en esta tesis. Sin embargo, no se hablará sobre la instrumentación tradicional, ya que no es considerada como la mejor opción para un proyecto *SCADA*, ésta sería escasamente eficiente dada la capacidad del entorno de operación bajo la cual va a funcionar, más bien, se abordará la temática de la instrumentación de alto nivel, conocida hoy en día también como instrumentación inteligente.

2.3.1. Adquisición inteligente de datos.

Como parte del instrumental inteligente se encuentran los módulos de adquisición de datos, los cuales centran su actividad en un microprocesador o microcontrolador. Estos elementos no alteran en modo alguno el estado del proceso en cuestión, es decir, no realizan sobre él ninguna maniobra de control, pero sí manipulan, con fines perfectamente establecidos, la información que de él es recogida. Con la intervención de un micro elemento, la labor de adquisición de datos pasa de la cruda extracción de información a la cerebral administración de la misma.

En un dispositivo de adquisición de datos se pueden identificar varias etapas, cada una de

2.3.1.1. Acondicionamiento digital de señal. Tras el proceso de conversión analógico/digital frecuentemente suelen incluirse procedimientos que permiten la linearización de la respuesta de los sensores primarios y secundarios, pues es conocido que todos los transductores de variables de proceso no tienen un comportamiento lineal, por lo menos no en todo su rango. Así mismo, se procura una compensación digital que minimice los efectos térmicos y vibratorios a los cuales están sujetos dichos dispositivos. [20]

2.3.1.2. Filtros digitales. Después de este acondicionamiento digital preliminar, es común la implementación de filtros digitales de diferente tipo. Estos no son más que algoritmos computacionales basados en expresiones matemáticas orientadas a mejorar la calidad de los datos y a decifrar cualquier característica predominante de un lote de información. [20]

Los filtros digitales son clasificados en dos categorías: filtros no-recursivos y filtros recursivos. Los primeros calculan las señales de salida actual en base a una ponderación, tanto de las señales de entrada actuales, como de las señales de entrada anteriores. Esta ponderacion se entiende como la suma de productos matriciales entre matrices de coeficientes constantes, que son los coeficientes del filtro, con vectores de entrada.

$$\begin{pmatrix} y_1(k) \\ \vdots \\ y_n(k) \end{pmatrix} = \mathbf{C}_0 \begin{pmatrix} x_1(k) \\ \vdots \\ x_m(k) \end{pmatrix} + \mathbf{C}_1 \begin{pmatrix} x_1(k-1) \\ \vdots \\ x_m(k-1) \end{pmatrix} + \cdots + \mathbf{C}_r \begin{pmatrix} x_1(k-r) \\ \vdots \\ x_m(k-r) \end{pmatrix}$$

En cambio, para estimar las señales de salida en filtros recursivos, además de ponderar el conjunto de entradas actuales y anteriores, se deben considerar las señales de salida del filtro producidas instantes atrás. Así, al resultado de las primeras operaciones debe sumársele el resultado de la suma de los productos de matrices de coeficientes de

recursividad por vectores de salidas anteriores.

$$\begin{pmatrix} y_1(k) \\ \vdots \\ y_n(k) \end{pmatrix} = \mathbf{C}_0 \begin{pmatrix} x_1(k) \\ \vdots \\ x_m(k) \end{pmatrix} + \cdots + \mathbf{C}_r \begin{pmatrix} x_1(k-r) \\ \vdots \\ x_m(k-r) \end{pmatrix} + \mathbf{D}_1 \begin{pmatrix} y_1(k-1) \\ \vdots \\ y_n(k-1) \end{pmatrix} + \cdots + \mathbf{D}_s \begin{pmatrix} y_1(k-s) \\ \vdots \\ y_n(k-s) \end{pmatrix}$$

El diseño de filtros digitales se centra en la estimación de los coeficientes de las matrices anteriormente señaladas ($C_0,\ldots,C_s$, $D_1,\ldots,D_s$), y para el efecto se recurre a las respuestas de frecuencia que se procura tengan dichos filtros. Por ser increiblemente complicado el proceso de sístesis de estas matrices, es indispensable la asistencia de un computador que ejecute algún programa especializado.

Una importante característica de los filtros digitales es su capacidad de seleccionar o discriminar con finura una determinada banda de frecuencias. Por esta razón son muy utilizados para reducir el ruido electrónico o ruido térmico, no eliminado en etapas anteriores, y más que nada, para atenuar el ruido de cuantización, introducido inevitablemente por el proceso de conversión analógico/digital. Sin embargo, esta no es la única, ni la mejor facultad de un filto digital, pues más interesante aún resulta el hecho de que un algoritmo de esta naturaleza, con coeficientes adecuados, puede proporcionar señales de salida que sean una confiable *predicción* de futuros valores en las señales de entrada. [2]

Los procedimientos inteligentes de adquisición de datos hacen uso intensivo de filtros digitales, dados los beneficios que su implementación trae consigo; sin embargo, no son los únicos recursos utilizables. En instrumental más avanzado, como el caso de analizadores, y más aun, en reconocedores se emplean métodos de inteligencia artificial para procesar la información recibida.

2.3.1.3. Analizadores. Los analizadores son dispositivos que extraen una muestra representativa del medio a estudiar e interaccionan con ella para determinar si en dicho medio están presentes ciertas características que son de interés. Ejemplos de estos dispositivos son los analizadores industriales para acometidas eléctricas, analizadores espectrales de señales electromagnéticas, analizadores espectrales de radiación solar, analizadores de composición química, analizadores de concentración, analizadores de contaminación ambiental, etc. Nótese que los analizadores no se limitan a obtener una representación *cuantitativa*, por sofisticada que sea, de alguna variable de proceso, sino por el contrario, lo que se busca con un analizador es una descripción *cualitativa* de la muestra recogida. [55]

Para entender mejor esta situación, se hará referencia a un analizador industrial de variables eléctricas. Este equipo es utilizado para monitorar el comportamiento las magnitudes corriente I, voltaje V, potencia activa P, poterncia reactiva Q, potencia aparente S, frecuencia f, factor de potencia fp, y distorsión armónica total THD, presentes en la acometida de una instalación industrial. Este analizador tiene normalmente bobinas de corriente y voltaje similares a las de un vatímetro trifásico. De estas bobinas se extrae la información básica del sistema, es decir, las magnitudes de corrientes de línea, voltajes de línea y potencia trifásica. Algunas representaciones de ellas como sus valores DC y RMS se obtienen directamente con un transductor ordinario; otras, como la forma de onda y el espectro de frecuencias requieren que las señales de voltaje y corriente sean muestreadas periódicamente a un ritmo de 10000 mediciones por segundo, o más, pues se procurará siempre tener la mejor resolución posible. A partir de estos datos, y con ayuda de un microcontrolador, se sintetizan los demás. El siguiente razonamiento demostrará cómo esto es factible.

Por definición, la potencia aparente en un sistema trifásico es $S = \sqrt{3} \cdot V_s \cdot I_s$, y el factor de potencia $fp = \frac{P}{S} = fp_{dist} \cdot fp_{desp}$. Con la ayuda de filtros digitales se pueden determinar las magnitudes de cada una de las componentes armónicas de las señales de corriente y

calcular así los valores $fp_{dist} = \frac{I_1}{I_s}$ y $THD = \frac{\sqrt{I_s^2 - I_1^2}}{I_1} \times 100$, que respectivamente son el factor de potencia de distorsión y el porcentaje de distorsión armónica total. Posteriormente se puede calcular el factor de potencia de desplazamiento fp_{desp} y la fase de la componente fundamental $\Phi_1 = \arccos(fp_{desp})$. Conocidos estos valores, y sujetándose a las definiciones, es posible determinar finalmente la potencia trifásica reactiva $Q = \sqrt{3} \cdot V_s \cdot I_1 \cdot \sin(\Phi_1)$ y la potencia trifásica de distorsión $D = \sqrt{S^2 - P^2 - Q^2}$. De esta manera llegan a ser conocidos todos los parámetros importantes de un sistema de alimentación de corriente alterna. [47] [21]

Con este ejemplo, lo que debe quedar claro es que en un analizador algunos datos son medidos, otros son calculados, y otros son conjeturados, siendo imprescindible para ello, la intervención de un microcontrolador. Esto es lo que busca la instrumentación inteligente, decifrar la información que permanece oculta en las primeras instancias del proceso de adquisión de datos.

2.3.1.4. Reconocedores. Cierto tipo de información escapa del dominio de los analizadores, es decir, no basta un mero cálculo matemático para poder acceder a ella, sino una evaluación mucho más profunda, casi *humanoide* se podría decir. Este es el caso de los reconocedores. Estos son sofisticadísimos equipos que recopilan información, la analizan, y la comparan con patrones preestablecidos, a través de un proceso artificial de reconocimiento. El desarrollo de la ciencia a este respecto ha sido inigualable.

Sustentándose en modernos métodos de inteligencia artificial, se pretende dar a los equipos electrónicos ciertos rasgos humanos como son la capacidad de identificación y asociación, la potestad de razonamiento y decisión, así como las complicadas facultades de adaptación y aprendizaje. Sin embargo, en lo que se refiere a instrumentación, conciernen solo los aspectos de identificación y asociación de imágenes y sonido, puesto que los demás son más bien aplicables a estrategias de control. [49] [50]

Actualmente, se realizan intensos estudios sobre de las diferentes técnicas que permitan la identificación y asociación de imágenes y sonido. Entre ellas se cuenta con Redes Neuronales, Sistemas Expertos, o una ingeniosa combinación de ambos. En base a estas técnicas se busca poner fin a las difíciles tareas de reconocimiento de figuras bidimensionales y tridimensionales, reconocimiento de habla, y reconocimiento de texto manuscrito.

Incursionar en estos temas parece inecesario dado el alcance y el objetivo de este trabajo de investigación. De hecho lo es, pues existe suficiente campo para poner en marcha varios proyectos de investigación y desarrollo, pero se ha considerado prudente al menos mencionar estas innovaciones debido a que en la actualidad los sistemas *SCADA* han comenzado a invadir nuevos territorios, tal es el caso de sistemas de alta seguridad, aplicados a edificios y planteles con acceso restringido de personal. Estos sistemas de seguridad funcionan bajo una modalidad *SCADA* y hacen uso exhaustivo de equipo similar al descrito en este apartado.

2.3.2. Transmisión inteligente de datos.

Esta rama híbrida entre la instrumentación y las telecomunicaciones surge de la necesidad de transmitir idóneamente la información que con tanta laboriosidad ha sido captada en los equipos de adquisición digital de datos. Es el esqueleto que mantiene unidas todas las instancias de un lazo de control.

Los datos extraidos de un proceso pasan por diferentes tramos: de los niveles de transducción a los niveles de análisis digital, de éstos a los niveles de control, y de estos últimos ya sea a los niveles de supervisión o de regreso a la planta a través de los elementos de actuación. La comunicación a lo largo de cada uno de estos debe estar garantizada.

Para llevar la información de las etapas de transducción a las etapas de análisis, las

entidades de normalización de procesos industriales recomiendan el uso de señales de corriente, en el rango de 4 a 20 mA, o de señales de presión, en el rango de 3 a 15 psi, para los casos en los que los convertidores de corriente sean difíciles de implementar o amenacen peligrosamente la seguridad de la planta [20]. Actualmente, los transmisores de estos tipos de señales son construidos con características inteligentes, entendiéndose como tales, la capacidad de verificar si la información fue correctamente recibida y las habilidades de auto calibración del equipo y de auto inspección en busca de desperfectos.

De acuerdo al esquema de transmisión de datos establecido anteriormente, una vez convertida la señal a su expresión digital y sometida a un procesamiento primario, es enviada a las unidades de control más cercanas, que en el caso de sistemas distribuidos son los *RTU*. En prácticamente todos los casos esta transmisión se da a través de líneas estándar de comunicación serial o paralela, o bien, a través de una tarjeta de expansión dedicada exclusivamente a la recolección de datos.

El aspecto *inteligente* en este nivel de transmisión se da en el protocolo de comunicación utilizado para el efecto y en las facilidades que éste presta para el reconocimiento y la corrección de errores. Esta situación es afrontada tanto a nivel de hardware como de software, y son varias las compañías de equipos de computación que proponen soluciones rentables a esta problemática. De ahí que no constituye mayor conflicto acoplar los equipos de instrumentación a los equipos de control. [15] [16]

Avanzando en la trayectoria, el tramo siguiente es el que comunica las unidades remotas con estaciones de mayor jerarquía. Este punto ya ha sido discutido con suficiente profundidad en la sección 2.2.3, referirse a ella para mayor información.

Finalmente, queda el tramo que enlaza el controlador con los dispositivos de actuación. Si los actuadores son sofisticados, la interconexión de estos elementos será directamente a través de líneas de transmisión digital, si no lo son, las señales de control deberán enviarse

a etapas previas de decodificación y de amplificación de potencia.

En este segundo caso, las señales digitales son transformadas a sus versiones analógicas, ya sea de corriente o de presión, y una vez así, son enviadas a los actuadores. Como es de esperarse, los equipos que realizan estas tareas tienen las mismas funciones inteligentes que sus duales en la parte de adquisición de datos.

2.3.3. Actuación inteligente.

De los equipos que intervienen directamente en un lazo de control quedan solo por mencionar los actuadores, aquellos dispositivos que convierten en señal reguladora del proceso, una instrucción concreta proveniente del controlador.

Nuevamente, aquí se dejará de lado el instrumental convencional y se tratará sobre los progresos del modernismo tecnológico. Este es el caso de los actuadores que tienen como parte medular de su construcción un sistema microprocesado.

Las ventajas de la tecnología digital a este respecto son muy grandes, entre éstas: la interpretación directa de una señal digital emitida por el organismo de control, aumento de la exactitud y presición de la señal acutante que se envía al proceso, ahorro de energía al emplear mayoritariamente señales de baja potencia, fácil acoplamiento de los actuadores a un computador digital, costos cada vez menores de elementos semiconductores, y la existencia de procesos que, por maniobrar con señales de pulso, son intrínsecamente digitales. [13] [62]

Además de estas virtudes generales, son también apreciadas las capacidades de auto diagnóstico y auto calibración discutidas anteriormente, las cuales solo son viables por la presencia de un microcontrolador. Cabe mencionar también, que a actuadores de alto nivel, con probabilidad no se les envía una única señal de control, sino un conjunto de

comandos que a través de ellos se traduzcan en una secuencia de acciones.

La reconocida eficiencia de los circuitos microprocesados en el campo de los actuadores ha permitido que la operación automática de equipo sea difundida en hogares, en centros de aprendizaje e investigación, e igualmente en parques industriales.

Particularmente en la industria han sido abreviadas de las actividades humanas todas las tareas peligrosas, repetitivas, tediosas o simples. La prueba de esto es que hoy se emplean máquinas que cargan, sueldan, cortan, ajustan o moldean, con el objeto de conseguir presición, seguridad, economía y productividad. Inclusive se ha intentado ir más allá, y se ha buscado la manera de dotar a las máquinas de ciertas características antropomorfas. Así, los robots, cyborgs y demás entes humanoides han pasado del delirio ficcional a una auténtica realidad, y son ahora aplicados a procesos de ensamblaje, exploración marina y espacial. [23]

Nótese que es ambiguo el límite existente entre el instrumental de adquisición y el de transmisión de datos a nivel de lazo de control, igualmente lo es el que existe entre este último y el instrumental de actuación. Muchos autores funden las tres clasificaciones en dos: la de precontrol y la de poscontrol, pues no siempre es fácil categorizar a un equipo exclusivamente dentro de las tres clases. Sin embargo, aquí han sido discutidas por separado por la simple razón que así son fácilmente identificables sus características *inteligentes*, las cuales constituyen el tema central de esta sección.

2.4. SUPERVISION DE EVENTOS.

Superada la discusión acerca de la instrumentación que intervendrá en los lazo de control del proceso, se procede a analizar la forma en la que cada unidad terminal remota, subestación de observación, o estación maestra hará uso de esta instrumentación para

diseño del sistema *SCADA*. El sitio en el que se consolidan estas actividades era, hasta hace poco, una consola de maniobras adecuada específicamente para dichos propósitos, sin embargo, la tendencia actual ha desplazado este equipo y lo ha reemplazado por un computador digital con una poderosa interface gráfica y un amigable entorno de trabajo.

Resulta evidente entonces que el núcleo del sistema supervisor es el programa de computación desarrollado con dicho fin, y consecuentemente, el éxito de las operaciones implicadas depende directamente de la funcionalidad y eficiencia de dicho programa.

Para la elaboración del paquete computacional en cuestión, han de responderse a las siguientes interrogantes:

- ¿Quiénes serán los usuarios?
- ¿Qué información es necesaria?
- ¿Qué información está disponible?
- ¿Qué requerimientos tiene la planta?, y
- ¿Qué requerimientos tiene el sistema de control?.

Bajo estas premisas y con la intención de satisfacer a plenitud las necesidades del particular proyecto *SCADA* involucrado, se procede a diseñar el programa de supervisión y control [3]. A continuación se detallan los principales aspectos que deben contemplarse en el mismo.

2.4.1. Configuración lógica del sistema.

En la sección 2.2.3. (Alternativas configurativas para la implementación de sistemas *SCADA*) se mencionó la necesidad de configurar a un sistema, tanto en su aspecto topológico, como en su aspecto jerárquico, y se recalcó la sustancial diferencia que hay entre ambos conceptos.

La estructuración topológica del sistema es el primer paso en el diseño de un proyecto *SCADA*, y su resultado es único y prácticamente definitivo. Mientras tanto, es función del organismo supervisor permitir, de forma dinámica, la configuración jerárquica y logica del sistema *SCADA*, en otras palabras, debe prestar la facilidad para definir los niveles de injerencia de un sistema distribuido y someter una unidad de observación cualquiera a un nivel jerárquico preestablecido. Igualmente, debe permitir el acoplamiento y desacoplamiento de una unidad del sistema para realizar en ella cualquier acción de mantenimiento preventivo o restaurativo.

Dentro de este campo se encuentra también la potestad de asociar distintos lazos de control a un *RTU* particular, y así direccionar toda la información proveniente de los mismos a una cadena jerárquica determinada. [37]

2.4.2. Acoplamiento digital de subsistemas.

Ha sido mencionado en secciones anteriores, que en sistemas dispersos existe la posibilidad de acoplar las señales de salida provenientes de algunos subprocesos con las entradas de otros, o bien, relacionar el estado de unos como factores condicionantes en otros. Existen muchas maneras de concatenar procesos distribuidos. Nuevamente, es obligación del supervisor viabilizar y garantizar estas relaciones de interdependencia.

Naturalmente, culaquier acoplamiento deberá hacerse a través de una unidad de observación de categoría superior a los procesos involucrados, de modo que se tenga predominio sobre la información proveniente de unos y de otros.

Aquí se hace evidente la necesidad de establecer previamente una estructura de depencencia lógica entre procesos, unidades remotas, subestaciones y estaciones, para poder llevar a cabo un acoplamiento digital entre subsistemas.

2.4.3. Monitoreo de hardware.

También es propio del supervisor monitorar el comportamiento del equipo de control y comunicaciónes. Para esta tarea se han de incorporar dispositivos dedicados y disponer de rutinas de inspección apropiadas, en base a las cuales se rastree el funcionamiento del equipo y se prevea con anticipación la sucitación de cualquier falla grave.

2.4.3. Monitoreo de hardware.

También es propio del supervisor monitorar el comportamiento del equipo de control y comunicaciónes. Para esta tarea se han de incorporar dispositivos dedicados y disponer de rutinas de inspección apropiadas, en base a las cuales se rastree el funcionamiento del equipo y se prevea con anticipación la sucitación de cualquier falla grave.

En sistemas *SCADA* es norma tener una disponibilidad de 99.9% o más. Entiéndase como disponibilidad la relación: $\frac{t_{\text{total}} - t_{\text{fuera de servicio}}}{t_{\text{total}}}$.

Para garantizar esta cifra no es suficiente la inclusión de equipo redundante con su respectivo procedimiento de actualización; pues adquiere elevada importancia también, la actividad de monitoreo de hardware por parte del órgano supervisor. [26]

2.4.4. Recopilación y evaluación de información.

Existe una gran diferencia entre la adquisición de datos con fines de control y la recolección de información con propósitos de supervisión. Esta diferencia radica tanto en la modalidad de la recopilación de datos, como en la naturaleza de la acción que ha de ejecutarse al ser evaluada dicha información.

En el primer caso, es absolutamente necesario muestrear las variables de proceso involucradas con una frecuencia constante y de valor suficientemente elevado, de manera que las representaciones discretas de dichas variables reflejen fielmente las características dinámicas de la planta. Cada muestra adquirida es comparada con un patrón referencial, y a partir de esta comparación es cuantificada una magnitud de error. Como se sabe, los sistemas de control hacen uso de este error para computar una magnitud de corrección, o ley de control, la cual al ser realimentada a la planta tenderá a acercar el valor real de la variable muestreada con el valor referencial esperado para ella. En otras palabras, cada

dato adquirido es inmediatamente utilizado para controlar el proceso.

En cambio, en un sistema supervisor la información puede ser recibida de dos formas: a través de un reporte periódico representativo o por medio de un reporte por excepción. Además, debe recalcarse el hecho de que dicha información no es evaluada numéricamente sino lógicamente, y como resultado de dicha evaluación no se devuelve una magnitud de control sino una decisión supervisora plasmada en maniobras de operación. [26]

Bajo un esquema de supervisión, tras recibirse del equipo de adquisición de datos un número significativo de muestras de un conjunto determinado de variables, suelen determinarse valores como la media, mediana, desviación estándar, índices de repetitibilidad, factores de correlación, tasas de cambio, entre otras magnitudes estadísticas. Estos valores son enviados al organismo de supervisión en forma de *reporte periódico representativo*, es decir, de manera cíclica, una vez realizadas tantas mediciones como fueren necesarias.

Por otro lado, se entiende por *reporte por excepción* el envio de datos cuando en el proceso se registra la ocurrencia de un evento determinado. Estos eventos pueden ser: el cumplimiento de un intervalo de tiempo preestablecido, la estabilización de una variable entorno a un valor determinado, la superación de los límites de funcionamiento normal de las magnitudes del proceso, en fin, cualquier condición fijada con anterioridad por el personal de operación de la planta.

Lo que debe quedar claro es que para la vigilancia o supervisión de un proceso no se requiere el conjunto total de datos, sino por el contrario, un número reducido de ellos, pero que resuman el estado del sistema e indiquen qué factores son de verdadera incidencia en el desarrollo normal del proceso.

La evaluación de la información extraida por el supervisor sigue un procedimiento

semejante al siguiente:

En primer lugar se identifica cuál es el nombre de la variable que se reporta, y a qué lazo de control y estación remota está asociada dicha variable. A continuación se determina el tipo de dato que se recibe; este puede ser una magnitud contínua, como el caso de un valor promedio, o una magnitud booleana, como el caso de la generación de cierto evento en el proceso. En este punto, se verifica si dicho valor cumple o no con alguna condición predefinida por el operador que desencadene algún tipo de acción supervisora. De ser así, y de ser varias las instrucciones a ejecutarse, se las ordena según su prioridad y se debate cuál de ellas tiene supremacía sobre las demás en caso de existir conflicto. Finalmente se procede a ejecutar, según el orden, el conjunto de comandos listados.

Este modo de proceder suele confundirse con el Control Secuencial; de hecho, comparten sus mismos principios, pero la aclaración siguiente mostrará porqué un sistema supervisor es más versátil y dominante que un sistema de control secuencial.

En controladores de este último tipo las acciones a seguirse, tras la mencionada evaluación de información, normalmente se limitan a la producción de señales booleanas que determinen el arranque, detenimiento, o reinicialización de distintas etapas de un proceso. Si el control es un poco más completo, se responderá también con señales analógicas y/o digitales con las cuales se pueda corregir el comportamiento de una o varias magnitudes involucradas. Este es el caso de las versiones más modernas de controladores lógicos programables (*PLC*). Pero con un sistema supervisor, se logra influir en el proceso de una forma mucho más dramática y decisiva, puesto que se permite el ajuste intempestivo del punto de referencia, la recalibración automática de los parámetros de las leyes de control, o bien, el cambio instantáneo de los propios algoritmos de control digital, conforme a las necesidades de un proceso.

2.4.5. Definición de alarmas y mensajes.

Se ha indicado que una de las funciones supervisoras es observar si alguna variable del proceso sobrepasa ciertos límites de seguridad. Esto implica necesariamente que el ente supervisor debe permitir la definición de los rangos de operación de cada variable y facilitar la formulación de alarmas asociadas con el incumplimiento de dichas condiciones de operación.

Respecto al rango de operación de una variable particular han de fijarse cotas superiores, inferiores, e incluso en torno a la zona de cero. Naturalmente, para cada uno de estas, pueden considerarse regiones de operación en estado normal, estado de alerta, estado de emergencia y estado de falla. Así, lo común es una escala de límites como la siguiente:

- Límite superior de falla
- Límite superior de emergencia
- Límite superior de alerta
- Límite superior de zona de cero
- Línea de cero
- Límite inferior de zona de cero
- Límite inferior de alerta
- Límite inferior de emergencia
- Límite inferior de falla

La figura 2.5 explica bastante bien estos criterios. El infringimiento de cualquier condición crítica, como las indicadas aquí, debe ser notificado al personal de operación través de una alarma apropiada.

Se entiende por alarma una petición o solicitud de atención enviada al operador, la cual ha sido causada por la ocurrencia de algún evento de carácter extraordinario. Para definir

una alarma, además de determinar la condición causal, es necesario establecer su prioridad ante otras alarmas y no permitir conflictos entre ellas. Igualmente, es conveniente dimensionar un ligero tiempo de retardo en su activación, de modo que sea diferenciable el efecto transitorio de una perturbación o de una acción de control determinada, con la evolución del proceso a un verdadero estado de emergencia.

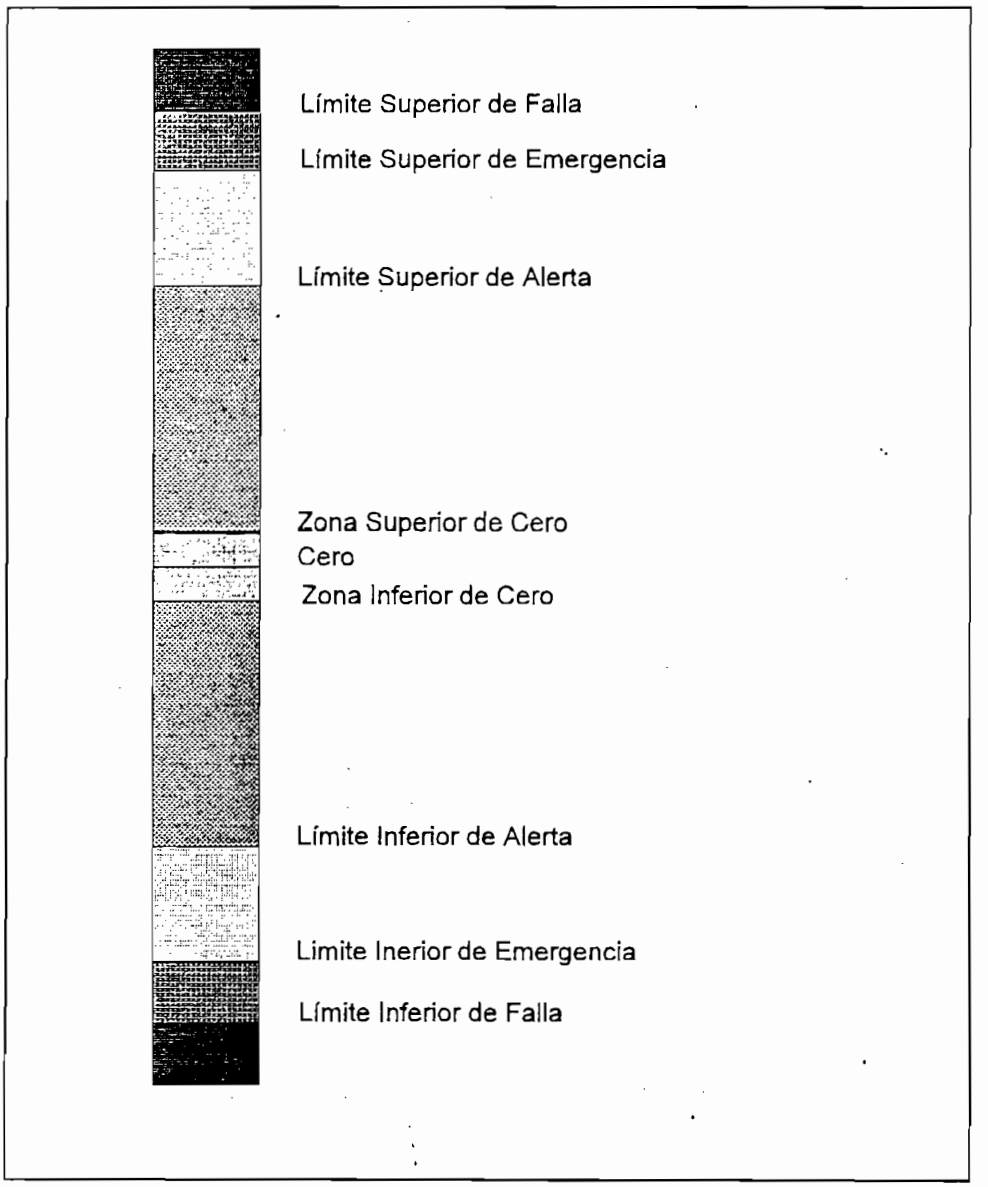


Figura 2.5. Escala de Límites.

Los avisos al operador deben acompañarse de señales audibles y visuales apropiadas, de manera que resulte evidente el requerimiento de atención al proceso por parte del personal.

Para esto, es recomendable atenerse a ciertos normativos industriales relacionados con señalización y seguridad. Y, aunque resulte algo obvio mencionar, los mensajes enviados deben contener el nombre de la estación y del lazo de control que emitieron la señal de alarma, y conjuntamente con estos, la razón de ser de la misma.

No es función del sistema supervisor hacer un análisis *post-mortem* que determine las causas que provocaron las fallas, esto es tarea neta de ingenieros y especialistas. Sin embargo, con el desarrollo de Sistemas Expertos, estas áreas de trabajo van relegándose a sofisticados equipos de computación, los cuales han sido acondicionados para ejercer tales funciones en modalidad ON-LINE. [9]

2.4.6. Inicialización y actualización de registros.

Al hacer referencia a los análisis *post-mortem* se sobreentiende la existencia de registros permanentes, en los cuales se almacena toda información que pasa por las manos del sistema supervisor.

Los registros son normalmente dispositivos fuera de línea que van actualizando sus datos de manera periódica, ya sea, cada hora, día, semana, mes, etc. El almacenamiento en término corto es utilizado para evaluar el desarrollo del proceso y el desempeño del sistema de control. Mientras tanto, el almacenamiento en término largo se orienta a la planificación futura en términos de expansión y a la reformulación de estrategias de producción.

Y así, según el fin que tengan, la informacion podrá ser guardada en diferentes medios: cartillas de registro, papel contínuo, cintas magnéticas, diskettes, discos rígidos, e inclusive memorias NVRAM (memorias no volátiles de lectura y escritura).

2.4.7. Emisión de reportes.

Los reportes son formas más elaboradas que los registros. Estos últimos solo recopilan información, mientras los primeros la resumen de una manera eficiente y la presentan al operador para que la juzgue.

En el aspecto práctico esto significa que no solo se informa la sucitación de alguna alarma, sino cuántas veces y a qué tiempo ésta se ha presentado desde el último reporte, igualmente, no solo se indicará el nuevo promedio de una variable, sino con qué velocidad ha cambiado y hacia a qué valor posiblemente se dirige.

Estas rutinas se denominan Tagged Time Data Routines (*TTD*), que como se observa, no son más que funciones estadísticas correctamente aplicadas al historial de un proceso. [26] Naturalmente, estos valores deben ser inicializados automáticamente cada que arranca un proceso y cada que se emite un informe nuevo.

En cuanto a los reportes solo queda por mencionar, que éstos pueden ser periódicos al igual que la actualización de registros, o bien, a diferencia de ellos, pueden emitirse por una orden expresa del operador.

2.4.8. Despliegue e introducción de información.

La interface hombre - máquina es uno de los aspectos más cuidadosamente diseñados en todo sistema de supervisión, pues siendo el enlace entre el proceso y el operador, no es posible concebir un sistema de control de esta naturaleza con falencias a este respecto.

De acuerdo a lo discutido y afirmado en ocaciones anteriores, la tendencia actual impone que el diálogo entre el hombre y la planta se dé a través de un computador digital. Los lenguajes de programación de extremado alto nivel existentes hoy en día, permiten la

fabricación de paquetes computacionales que contengan poderosas ayudas visuales orientadas tanto a simplificar la labor ordinaria del operador, como a su vez, desentenderle de todo conocimiento riguroso de los sistemas de computación actuales. No es extraño entonces, encontrar sofisticadísimos programas que carezcan de dificultades en su utilización, pero que a la vez, realizcen trabajos de alta calidad y profesionalismo.

En lo refernte a la presentación de información, puede argumentarse que todo conjunto de datos categorizados como trascendentes en el desenvolvimiento de un proceso deben estar al alcance inmediato del personal técnico, y deben ser expuestos ante él de manera clara y coherente. Entre la información presentada deben considerarse los siguientes puntos:

- Diagramas ON-LINE del Sistema, en los cuales se detalle la estructuración del mismo y la interdependencia de unos lazos de control con otros.
- Monitoreo del estado del Sistema, es decir, visualización del status de cada lazo de control, pudiendo ser éste: normal, alerta, emergencia, o falla.
- Listas de eventos sucitados, los cuales han de incluir los instantes de tiempo en los que se dieron, las condiciones que los promovieron, y las acciones tomadas a partir de ellos.
- Graficación de curvas, tanto de variables en tiempo real, como de registros anteriores.
- Reportes de resumen, ceñidos a los criterios discutidos en el apartado anterior.
- Libros de registro del personal técnico encargado de la vigilancia y operación del Sistema.

Tan importante como la presentación de la información vital de un proceso, es la introducción de parámetros al mismo por parte del operador. Así, la interface gráfica de un sistema supervisor correctamente diseñado debe permitir:

- Configuración y reconfiguración de la estructura del sistema.
- Arranque y detenimiento de las actividades de control y supervisión.
- Cambio arbitrario de puntos de ajuste o niveles de referencia.
- Calibración de parámetros en los algoritmos de control digital.

- Selección de diferentes técnicas de control predefinidas, aplicables a un mismo lazo.
- Formulación de alarmas con todos sus atributos.
- Establecimiento de condiciones y consecuencias en la generación de eventos.
- Anotaciones explícitas del operador.

Con estas aclaraciones se completa la descripción de un sistema supervisor de características convencionales. El paso siguiente es el tratamiento de las estrategias de control digital aplicables en un contexto supervisor.

2.5. CONTROL DIGITAL BAJO EL ESQUEMA *SCADA*.

Para completar el marco teórico sobre los sistemas *SCADA* queda solo por mencionar las técnicas de control de lazo comúnmente utilizadas en un entorno de esta naturaleza. Intencionalmente, esta sección ha sido reservada para el final del capítulo por la sencilla razón de que un método de control digital, cualquiera que éste sea, presenta distintos caracteres dependiendo si opera autónomamente o si se desenvuelve bajo una estructura supervisora. Ha sido necesario por tanto, definir y describir primero los aspectos fundamentales de un sistema supervisor, y a partir de ellos, plantear y dimensionar correctamente las alternativas de control aplicables a la situación.

2.5.1. Requerimientos para el control digital en tiempo real.

Enfóquese ahora la atención sobre los equipos controladores encargados del desenvolvimiento de cada uno de los lazos de interés. Las condiciones consideradas como indispensables para calificar a un controlador como un dispositivo en tiempo real, necesariamente deben ser abordadas en primera instancia. Al enunciarlas más adelante, se buscará no perder generalidad, sino por el contrario asociarlas y describirlas como cualidades comunes de un amplio conjunto de equipos comercialmente difundidos. En el

apartado 2.5.2., sin embargo, se discutirán precisamente las diferencias entre dichos equipos y se discernirá sobre su opción de formar o no parte de un sistema *SCADA*.

2.5.1.1. Configuración de un controlador. Un controlador de lazo consiste esencialmente de un procesador central rodeado de circuitería periférica, tanto universal como especializada, que le permiten interactuar directamente con el proceso, con el personal y con otros dispositivos de control pertenecientes al sistema. Normalmente los controladores son construidos sobre una base modular que permite fácilmente la adaptación de un mismo dispositivo a las diversas situaciones y requerimientos del proceso. Este esqueleto de soporte incluye el procesador central, entradas y salidas, tanto analógicas como digitales, canales de telemetría, líneas de interrupción, líneas de enlace con otros controladores y consola de operacíon [5]. Obsérvese el diagrama de la figura 2.6:

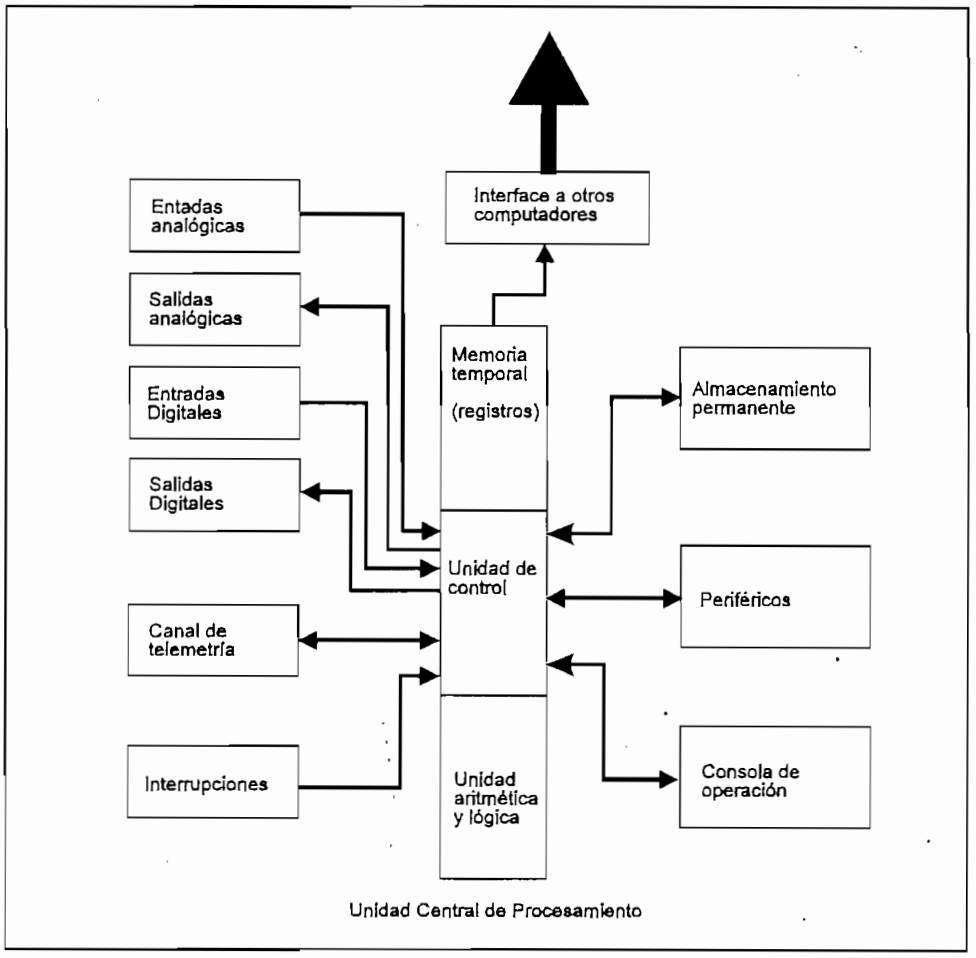


Figura 2.6 Diagrama de un sistema de control computarizado.

2.5.1.2. Procesador central. El procesador central consiste básicamente en un microcontrolador con circuitería auxillar para el direccionamiento de memoria y manejo de periférficos. Puede ser considerado como el corazón del equipo de control, pues tiene a su cargo las actividades aritméticas y lógicas necesarias para el cómputo de las leyes de actuación, y la coordinación de los demás elementos del controlador. [19]

En la etapa de diseño del procesador central deben tenerse en cuenta los aspectos siguientes, pues todos ellos insiden directamente en la facilidad de su implementación y en la efectividad de su funcionamiento:

- Tamaño de palabras binarias
- Set de instrucciones
- Modos de direccionamiento
- Transferencia de información
- Manejo de interrupciones
- Velocidad de procesamiento
- Memoria interna

No se detallará ninguno de estos por cuanto se extienden más allá de los objetivos primordiales de este trabajo.

2.5.1.3. Almacenamiento de información. En un controlador de lazo se exigen dos formas de almacenamiento de información: en elementos de memoria de acceso rápido y en medios electrónicos o magnéticos para registros permanentes. [19]

La primera se refiere a los bancos de memoria del procesador central donde se encuentra el programa de control y la información inmediata, la cual es recibida, procesada y enviada de vuelta al proceso. Este tipo de almacenamiento permite acceso aleatorio a la información, manipulación paralela bit a bit, facilidad de transmisión hacia periféricos

Memoria (*DMA*, del inglés Direct Memory Access), puesto que utiliza procesadores auxiliares con ciclos independientes de lectura y escritura, que no demandan la vigilancia del procesador central. Así, esta técnica permite llevar a cabo, de forma paralela, funciones de adquisición, procesamiento y envío de datos; razón por la cual es la técnica más apetecida para el control en tiempo real, aún cuando ella implique la adaptación de circuitería externa y una elevación del costo del control.

2.5.1.5. Interrupciones. Entiéndase por interrupción, una señal originada por el proceso, por el operador, o por el mismo controlador que provoca el detenimiento de la actividad ordinaria del procesador y la inmediata atención a una rutina específica. Son usuales causas de interrupción los siguientes factores:

- Verificación de alarmas y de eventos por parte del sistema supervisor
- Maniobras explícitas del personal de operación
- Reloj interno en tiempo real
- Control del acceso directo a memoria
- Coordinación de la actividad de varios microprocesadores en paralelo
- Fallas en el funcionamiento de equipo
- Interrupción intempestiva de alimentación (Power Fail Warning)

El manejo de interrupciones es uno de los problemas más difíciles de implementar en un dispositivo de tiempo real, pues su naturaleza aleatoria hace que la programación pierda mucho de su encantable estructuración y de su recomendada verificación. Sin embargo, esta es una situación a la que es menester adaptarse, puesto que en el carácter aleatorio de dichos sucesos se encuentra presisamente la clave del éxito del control en tiempo real. [4]

Así, para diseñar correctamente un programa administrador de interrupciones deberá cuestionarse sobre los siguientes aspectos:

- ¿Cuándo deben examinarse las líneas de interrupción?
- ¿Qué características deben tener las señales de interrupción?
- ¿Cómo se transfiere el control de la rutina principal a la rutina de servicio?
- ¿Cómo debe el procesador central almacenar la información vital antes de atender una interrupción?
- ¿Cómo debe determinarse el origen de la interrupción?
- ¿Cómo puede diferenciarse entre interrupciones importantes y de orden secundario?
- ¿Cómo puede debatirse entre interrupciones de igual prioridad que se han producido simultáneamente?
- ¿Cómo puede desactivarse la atención a una o varias fuentes de interrupción?

Con esto termina la descripción de las características universales de los equipos de control en tiempo real. Se procederá ahora con la descripción de las particularidades de cada tipo de controlador.

2.5.2. Alternativas de control digital en tiempo real.

En el proceso de evolución hacia la automatización total, las necesidades de los sectores productivo, industrial y de consumo, han forzado la inventiva y capacidad ingenieril de los fabricantes de equipos controladores, demandando dispositivos cada vez más versátiles, precisos y confiables, que pongan fin a sus deficiencias tecnológicas. Muchos han sido los prototipos desarrollados, pero conforme se han refinado y uniformizado las normas industriales se han delineado tres tendencias fundamentales: dispositivos de control dedicados, controladores lógicos programables y controles computarizados.

2.5.2.1. Controladores dedicados. Estos son dispositivos que se diseñan y construyen para satisfacer una específica necesidad de un particular proceso. Normalmente se implementan en base a un microcontrolador de programación única, y por esta razón, no

permiten la adaptación del programa a distintas situaciones, sino únicamente una limitada calibración de los parametros de control existentes. [4]

Sin embargo, a pesar de esta inconveniencia, son controladores de tiempo de respuesta muy corto y por ello son extensamente utilizados en la regulación de procesos rápidos. El lenguaje de programación comúnmente utilizado es código Ensamblador, y a esto se debe la elevada velocidad de actuación de dichos dispositivos. También se acostumbra a programar en versiones simplificadas de C, BASIC o FORTRAN, aún cuando su utilización incida desfavorablemente en el tiempo de respuesta del control, pues en cambio, permiten una programación estructurada, y más importante todavía, facilitan la utilización de funciones aritméticas con números enteros o de reales de simple precisión. [19][25]

La mayoría de estos controladores pueden ser incorporados a un sistema de control distribuido como es un sistema *SCADA*, ya que se fabrican respetando ciertas normas de intercomunicación. Pero, aún cuando esto es factible, más popular es su utilización lazos de control autónomos.

2.5.2.2. Controladores lógicos programables. Los controladores lógicos programables, conocidos también como *PLC's*, son equipos que se desarrollaron para sustituir paneles de control en base a relés; es decir, fueron originalmente concebidos para recibir entradas binarias, operar sobre ellas con funciones booleanas, y devolver señales del mismo tipo en respuesta. Con el acelerado ritmo de tecnificación, los *PLC's* se volvieron cada vez más flexibles y fáciles de utilizar, y han llegado a tener éxito más allá de lo esperado. Hoy en día, estos equipos pueden manejar tanto señales analógicas como digitales, y comandar secuencias lógicas, temporizaciones, conteos, operaciones aritméticas e inclusive leyes de control como algoritmos PID. Sin embargo, la verdadera ventaja de estos dispositivos es que permiten programar el conjunto de funciones y rutinas que se desea se cumplan en un determinado proceso; consiguiendo con esto, que un mismo controlador pueda ser

utilizado para la gobernación de distintos procesos, sin necesidad de adaptaciones complicadas sino con una simple reprogramación de su memoria estática.

La programación de los *PLC's* se realiza a través de un computador personal o por medio de un terminal portátil expresamente diseñado para configurar controladores de instalación remota. El lenguaje que se utiliza es un código, gráfico o textual, exclusivo para la programación de *PLC's*, que lo proporciona cada fabricante. Usualmente se implementa la programación tipo Ladder, que es una forma gráfica de ingresar funciones y parámetros al controlador, tal como si se diagramara un circuito eléctrico industrial en norma DIN. Otras modalidades de programación, aunque menos utilizadas, son: los bloques digitales y las sentencias textuales. [48]

Los fabricantes de estos equipos previendo la necesidad de enlazar varios *PLC's* a un red de comunicaciones y formar una estructura de control más grande, han construido algunos modelos con canales de comunicación de distinta capacidad y alcance. Esta es una razón más para que sean sean tan codiciados en proyectos de automatización.

2.5.2.3. Controles computarizados. El tercer grupo de controladores en tiempo real lo conforman los computadores, ya sean de operación individual o de operación coordinada bajo una configuración reticular, es decir, en red. Dichos equipos integran las más sofisticadas técnicas de control y los más amigables entornos de trabajo, pues las posibilidades que presenta la programación actual prácticamente no tiene límites. Brindan además la posibilidad de registrar, almacenar y transmitir información hacia otras estaciones de control, razones de mucho peso cuando se pretende extender su utilización más allá de un simple control, y abarcar rutinas de supervisión para sistemas de magnitud considerable. [62]

Sucede sin embargo, que tanta versatilidad es producto de una dificultosa implementación,

Para hacer una elección adecuada debe definirse en base a la teoría, intuición o experiencia, una función de costo, conocida también como función de criterio, que es una cuantificación del efecto que produce la trayectoria de control escogida sobre el sistema a controlarse. En segundo lugar, deben establecerse las condiciones de respuesta, es decir, todos los factores que definan el comportamiento deseado del sistema.

Así, el problema de control óptimo, que dicho de paso, es extremadamente difícil, se centra en hallar una trayectoria de control que estabilice al sistema en torno a una señal de set point, minimice la función de costo y cumpla con las condiciones de respuesta establecidas. [59]

2.5.3.4. Algoritmos adaptivos. Los procesos reales siempre presentan efectos no lineales y señales ruidosas intrínsecas que son despreciadas al momento de diseñar los sistemas de control correspondientes. Por tal razón dichos controles responden adecuadamente solo bajo ciertas condiciones y en un rango lineal de funcionamiento. Si se pretende operar en regiones ligeramente diferentes, o si eventualmente el nivel de ruido aumenta, los sistemas de control dejarán de ejercer su acción reguladora, e incluso podrán llevar al proceso a regiones de severa inestabilidad.

Para solucionar este inconveniente se desarrollaron los sistemas de control adaptivo, los cuales tienen la facultad de modificar automáticamente los parámetros de control cuando cambian impredeciblemente las condiciones de trabajo. Son varios hasta el momento, los algoritmos diseñados para este propósito, sin embargo, todos ejecutan cíclicamente las siguientes funciones:

- Identificación en línea de las características dinámicas de la planta
- Comparación de la respuesta real con la respuesta deseada
- Reajuste de los parámetros del control

3.1.2. Condiciones adversas a la implementación del sistema *SCADA* de laboratiorio.

En todo proyecto de automatización el sistema de control se *diseña, implementa y condiciona* de acuerdo al proceso que se pretende monitorar, buscando siempre el funcionamiento no solo satisfactorio, sino óptimo de toda la planta. Dicho de manera más clara, es el equipo de control el que se adapta a las particulares necesidades de la planta, y no éstas últimas a las posibilidades que ofrece el equipo de control.

Sin embargo, los recursos disponibles de la Escuela Politécnica Nacional, en particular, de la Facultad de Ingeniería Eléctrica, y por consiguiente del Departamento de Electrónica y Control, pondrá serias limitaciones en el equipo requerido para el desarrollo de este proyecto, teniendo que proceder de modo totalmente inverso al señalado, adaptando con conformidad las facilidades que pueda brindar el sistema de control a implementarse a las características de los equipos disponibles en los laboratorios.

3.1.3. Características del sistema *SCADA* de laboratorio.

Concéntrese ahora la atención sobre la potencialidad del nombrado ente de control, respetando las limitaciones que impone la ligera infraestructura del Laboratorio de Sistemas de Control al cual quedará sujeto este proyecto.

Se indicarán las especificaciones técnicas del Sistema *SCADA* de Laboratorio considerando los siguientes aspectos: configuración general, control digital directo y plataforma de supervisión.

3.1.3.1. Configuración general. A este respecto, propuesta la necesidad de maniobrar solo sobre prototipos pequeños que actúen de manera similar a un determinado proceso industrial, se ve *necesario y suficiente* la conformación de una estructura de control

[redacted] a detallar la operación de este dispositivo, se puede anticipar el hecho de que éste presenta gran facilidad para manejar tanto señales analógicas como señales digitales, y en virtud de esta característica, el sistema *SCADA* debe ser diseñado para permitir la programación de los algoritmos de control digital más comunes para variables en representación de 8 bits, entre estos: controladores PID discretos y redes discretas de adelanto y atraso de fase y

$$G(s) = \frac{0.00281}{z - 0.9778}$$

Este modelo servirá más adelante para estimar los coeficientes de las leyes de control.

Cabe recalcar que si el Sistema *SCADA* de Laboratorio es sobrecargado con varios prototipos simultáneamente se debe utilizar a un período de muestreo mayor, de modo que se garantice el tiempo suficiente para que el computador efectúe las operaciones que reqiera.

El período de muestreo ha sido escogido haciendo una evaluación de la rapidez del sistema en lazo abierto y comprometiendo al máximo la velocidad de muestreo del sistema de control *SCADA*, el cual, por las razones detalladas más adelante en este mismo capítulo, no puede tener un barrido más rápido que 10 Hz.

3.2.2. Módulo electrónico de simulación de funciones de transferencia.

Este es un equipo desarrollado para estudiar las técnicas de control digital aplicadas sobre plantas contínuas sencillas. Al igual que el caso anterior, es también el producto de una tesis de grado, y consta esencialmente de un módulo electrónico que permite la construcción de funciones de transferencia en base a la interconexión de circuitos lineales con amplificadores operacionales. Las salidas de las distintas plantas modeladas se conectan a las entradas analógicas de un sistema de adquisición y salida de datos, como la tarjeta DAS-128, mientras que las señales de control que salen de la misma realimentan la planta a través de sus puntos de entrada. [24]

El dispositivo en cuestión está representado en la figura 3.2.

Los dos circuitos superiores simulan dos sistemas integradores asociados a la función de

3.3. MODELACION Y SIMULACION DE PLANTAS INDUSTRIALES.

A pesar de que los prototipos descritos en la sección anterior servirán para ilustrar el funcionamiento del sistema *SCADA* de laboratorio, no son ni los más apropiados, ni los más calificados ejemplos que se pueden utilizar para probar el rendimiento de dicho sistema en el extremo máximo de sus posibilidades. Por tal razón, se ha dedicado una sección entera del presente capítulo para la descripción de modelos de plantas industriales relativamente complicadas, señalando que, por ser temas que se alejan del sentido fundamental de esta tesis de grado, no se confrontarán discusiones profundas sobre las teorías que sostienen la validez de dichos modelos, dejando más bien dicho trabajo a los expertos en la materia.

3.3.1. Simulador de un horno batch.

Los hornos batch son equipos industriales de alta tecnología que permiten el cocimiento de piezas de cerámica de diferente tipo: sanitarios, aislantes eléctricos para alta tensión, piezas mecánicas de precisión y de extremada dureza, etc.

Tal objetivo se consigue llevando la temperatura de todos los puntos del horno a lo largo de un patrón de temperatura en función del tiempo, similar a los de la figura 3.3. [35]

Las curvas indicadas variarán ligeramente entre un caso y otro dependiendo del tipo de elementos que se cuecen, su número, su tamaño, y las relaciones en su composición química.

El funcionamiento de estos equipos resulta bastante complicado debido a que un fiel seguimiento de la curva de Set Point no se consigue sino con un instrumental preciso, un control confiable y auto ajustable; y, un conjunto de actuadores que permitan tanto el calentamiento como el enfriamiento del recinto, es decir, no puede llevarse a cabo sin la

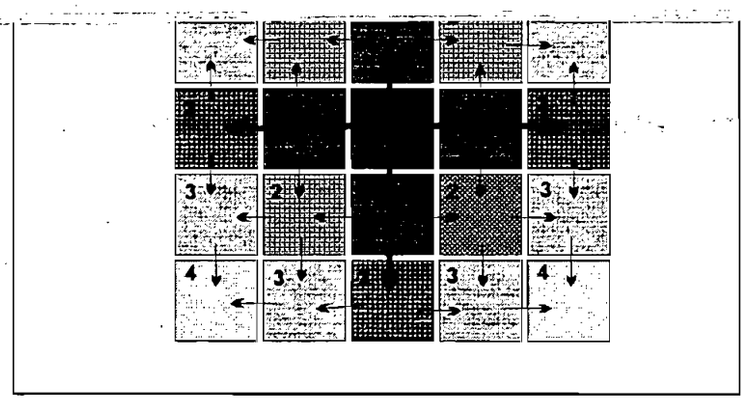


Figura 3.5. Propagación del calor en dos dimensiones.

El modelo exige que cada sector sea definido con diversos parámetros. Primeramente, debe

3.3. MODELACION Y SIMULACION DE PLANTAS INDUSTRIALES.

A pesar de que los prototipos descritos en la sección anterior servirán para ilustrar el funcionamiento del sistema *SCADA* de laboratorio, no son ni los más apropiados, ni los más calificados ejemplos que se pueden utilizar para probar el rendimiento de dicho sistema en el extremo máximo de sus posibilidades. Por tal razón, se ha dedicado una sección entera del presente capítulo para la descripción de modelos de plantas industriales relativamente complicadas, señalando que, por ser temas que se alejan del sentido fundamental de esta tesis de grado, no se confrontarán discusiones profundas sobre las teorías que sostienen la validez de dichos modelos, dejando más bien dicho trabajo a los expertos en la materia.

3.3.1. Simulador de un horno batch.

Los hornos batch son equipos industriales de alta tecnología que permiten el cocimiento de piezas de cerámica de diferente tipo: sanitarios, aislantes eléctricos para alta tensión, piezas mecánicas de precisión y de extremada dureza, etc.

Tal objetivo se consigue llevando la temperatura de todos los puntos del horno a lo largo de un patrón de temperatura en función del tiempo, similar a los de la figura 3.3. [35]

Las curvas indicadas variarán ligeramente entre un caso y otro dependiendo del tipo de elementos que se cuecen, su número, su tamaño, y las relaciones en su composición química.

El funcionamiento de estos equipos resulta bastante complicado debido a que un fiel seguimiento de la curva de Set Point no se consigue sino con un instrumental preciso, un control confiable y auto ajustable; y, un conjunto de actuadores que permitan tanto el calentamiento como el enfriamiento del recinto, es decir, no puede llevarse a cabo sin la

Para entender mejor esta técnica de simulación es necesario visualizar el proceso en solo dos dimensiones. Obsérvese la figura 3.5. Considérese que en el sector 0 existe una fuente que irradia calor en sus cuatro posibles direcciones, afectando directamente a los sectores denominados con 1. Estos a su vez afectarán a los marcados con 2, pero la propagación ya no será en las cuatro direcciones, sino solo en aquellas en las que la propagación del calor tenga sentido físico, es decir, para el sector 2 superior la propagación vendrá desde abajo solo por medio del sector 1 superior; para el sector 2 derecho la propagación vendrá solo de la izquierda a través del sector 1 derecho; mientras que para el sector 2 superior derecho la propagación vendrá tanto desde abajo como desde la izquierda por la intervención del sector 1 derecho y del sector 1 superior respectivamente. Los sectores 2 propagarán el calor a los sectores 3, y así sucesivamente.

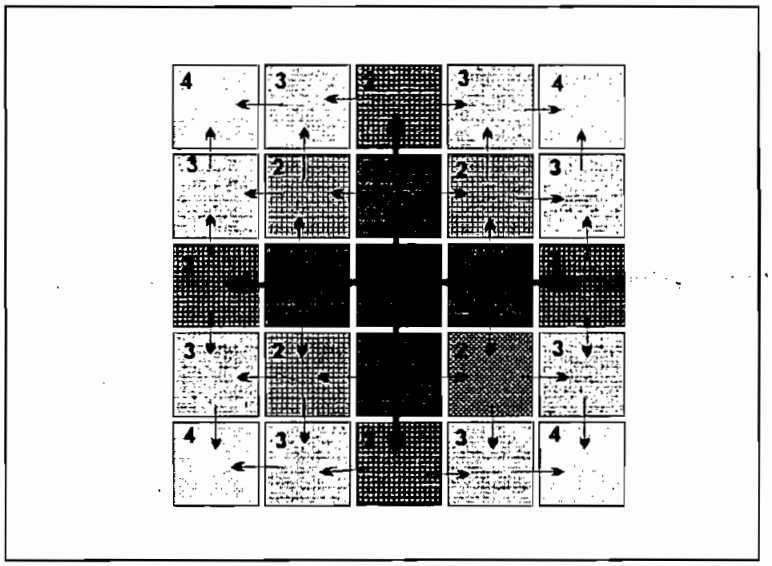


Figura 3.5. Propagación del calor en dos dimensiones.

El modelo exige que cada sector sea definido con diversos parámetros. Primeramente, debe contener referencias a los sectores de los cuales recibe el calor; en segundo lugar, debe incluir coeficientes que indiquen qué fracción de calor recibe de dichos sectores, resumiendo en ellos los efectos de conducción, convexión y radiación; adicionalmente habrá de considerarse un ligero porcentage de ruido que simule las turbulencias de aire debido al gradiente de temperatura existente entre sectores contiguos; y, por último,

tendrá que indicar qúe tiempo de retardo existe entre una alteración en los sectores previos y su incidencia en el sector actual.

Cuando se extrapola esta técnica a 3 dimensiones, deberán considerarse coeficientes de propagación y retardos en esta nueva dimensión, haciendo el problema considerablemente más complejo. El cálculo de tales coeficientes es una tarea complicada que no trasciende a esta tesis, razón por la cual serán omitidos. [29] [35]

El sector que contiene al quemador recibe un tratamiento especial, puesto que no se calienta instantáneamente a la temperatura deseada sino que se comporta como un sistema de primer orden con constante de tiempo relativamente larga y error en estado estable casi nulo [18]. Para este trabajo se considerará la siguiente función de transferencia:

$$G(s)=\frac{0.9998}{87.5s+1}$$

Más aún, un horno industrial como el que se quiere simular contendrá más de un quemador, y para el presente caso habrán de considerarse dos. Esta nueva complicación se resuelve aplicando el criterio de superposición de fuentes, lo que implica duplicar el número de parámetros definidos para cada sector, utilizando un primer grupo de ellos cuando solo opera el primer quemador mientras el otro tiene valor nulo, y el grupo restante, en el caso inverso.

Un nuevo problema surgue cuando se pretende evaluar la temperatura media del horno, pues es necesario informar al sistema de control con un valor representativo de la temperatura, la cual, como puede preverse no es uniforme en su interior, por lo menos no lo es sino hasta llegar completamente al estado estable. Este obstáculo ha sido sorteado simulando la presencia de un instrumental inteligente que sensa la temperatura del horno en diferentes puntos, evalúa su media aritmética y envía dicha señal al control. En el

horno existirán dos sensores de este tipo, uno que monitorea el comportamiento de la región derecha y otro el de la izquierda, cada uno de estos cierra un lazo de control con el quemador perteneciente a la misma región.

Complicado como parece, el problema no termina allí. Obsérvese la curva de Set Point anterior (Figura 3.3). El tramo en el que se busca decrementar la temperatura introduce un nuevo problema en el ámbito de los actuadores: ¿cómo disminuir controladamente la temperatura con una pendiente mayor que la definida por el enfriamiento natural? Felizmente existen cierto tipo de quemadores tipo ISO-JET que permiten la regulación contínua de la temperatura de la llama en un rango de 50 °C a 2500 °C.

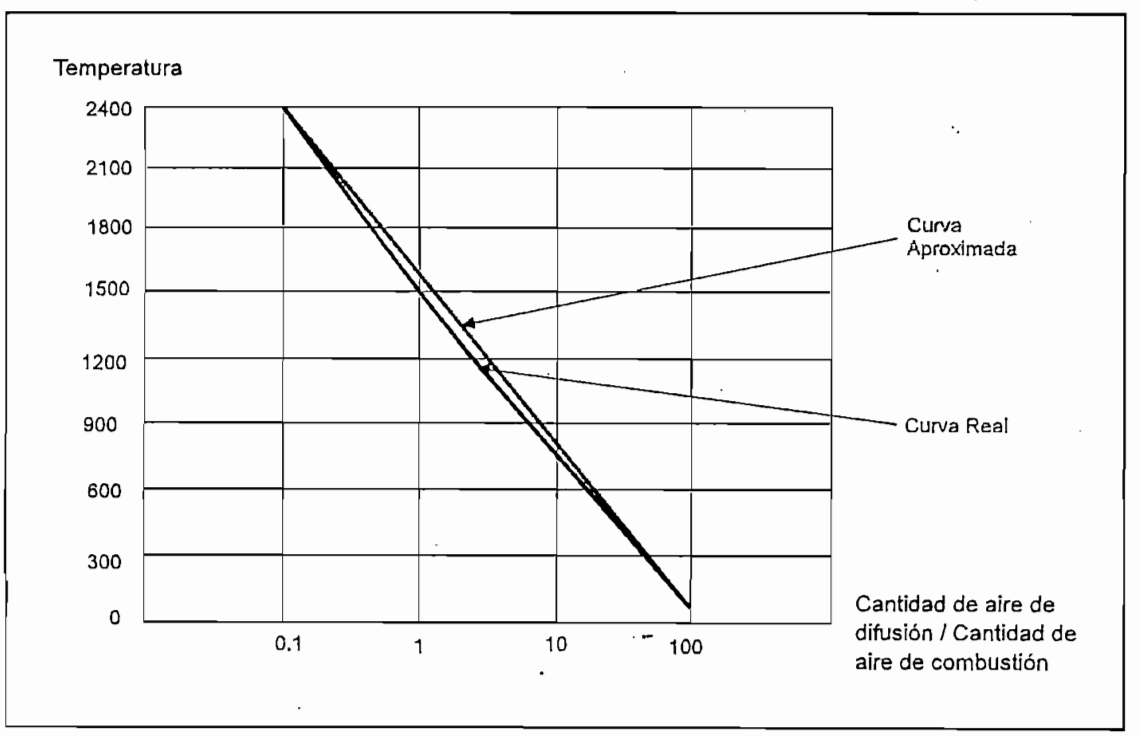


Figura 3.6. Curva de respuesta de un actuador ISO-JET.

El mecanismo de estos quemadores es el siguiente:

Cada quemador tiene tres servoválulas de fina precisión. Una de ellas controla el flujo de combustible, otra el flujo de aire de combustión, y la última, el flujo de aire de difusión.

Las dos primeras se fijan con anticipación para definir la temperatura máxima de la llama en el quemador, y adicionalmente lograr una mezcla estequiométrica que produzca una combustión sin humo y con el mínimo de monóxido de carbono. La tercera válvula inyecta aire a presión al quemador, pero no para servir de comburente sino para dar a la llama un *recubrimiento* de aire frío que neutralice su excesivo calor, logrando estabilizar la temperatura de la misma en algún punto intermedio. [20]

Así, la temperatura de la llama puede adquirir cualquier valor entre 50 °C y 2500 °C, dependiendo de la relación entre la concentración de aire de difusión y la concentración de aire de combustión en el quemador. La figura 3.6. explica dicha dependencia.

Otra considerable ventaja de este tipo de quemadores es que mantiene su temperatura a lo largo de todo el chorro, el cual, dependiendo del modelo del quemador podrá tener un largo desde 1m a 15m. Para el caso tratado se ha utilizado un quemador con chorro de 1m de largo.

El modelo computaciónal simula el comportamiento de dichos quemadores, fijando las válvulas de combustible y comburente para una temperatura máxima de 2400 °C, mientras que el aire de difusión se regula con otra válvula.

Finalmente, se simulará también la presencia de analizadores de combustión, que periódicamente emitirán los valores de la concentración de CO_2 y CO en la supuesta cámara de eliminación de gases.

La complejidad de todo este instrumental de medición y actuación, en conjunto con la delicadeza del proceso en sí, hacen necesaria la presencia de un organismo de control y supervisión de muy alto nivel para garantizar el éxito en cada ciclo de funcionamiento. Este es, por tanto, un prototipo idóneo para ser gobernado por un sistema *SCADA*. La figura 3.7 muestra un esquema completo del horno expuesto.

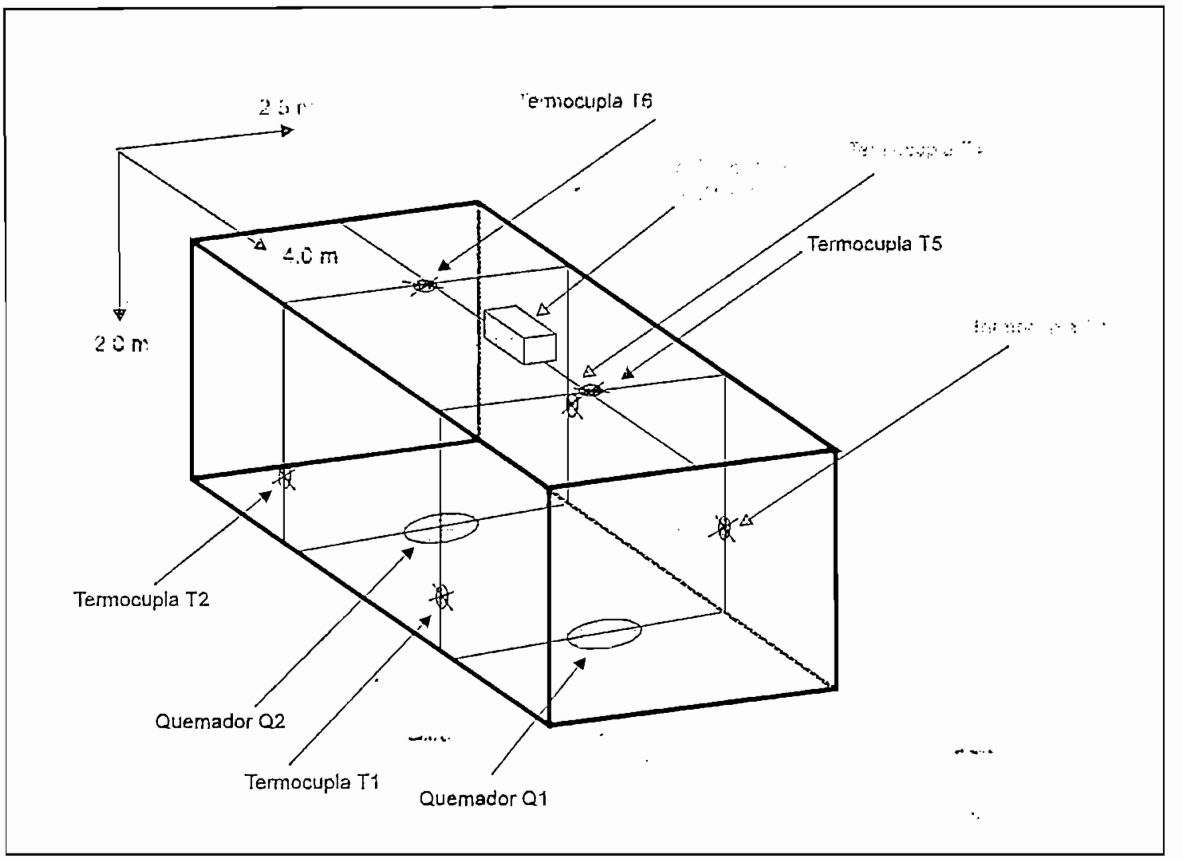


Figura 3.7. Esquema del Horno Batch.

Al realizar distintas pruebas en lazo abierto con el menciondado modelo, se encuentra que en su globalidad el horno se comporta como un sistema de primer orden con un pequeño retardo. El modelo está dado por:

$$G(s) = \frac{0.9975e^{-2s}}{141.3s+1}$$

Con esta ecuación se podrá obtener el modelo discreto de la planta con período de muestreo de 5 segundos, y a partir de él, calcular las leyes de compensación adecuadas.

El programa que contiene el modelo numérico del mencionado Horno Batch se muestra en la figura 3.8.

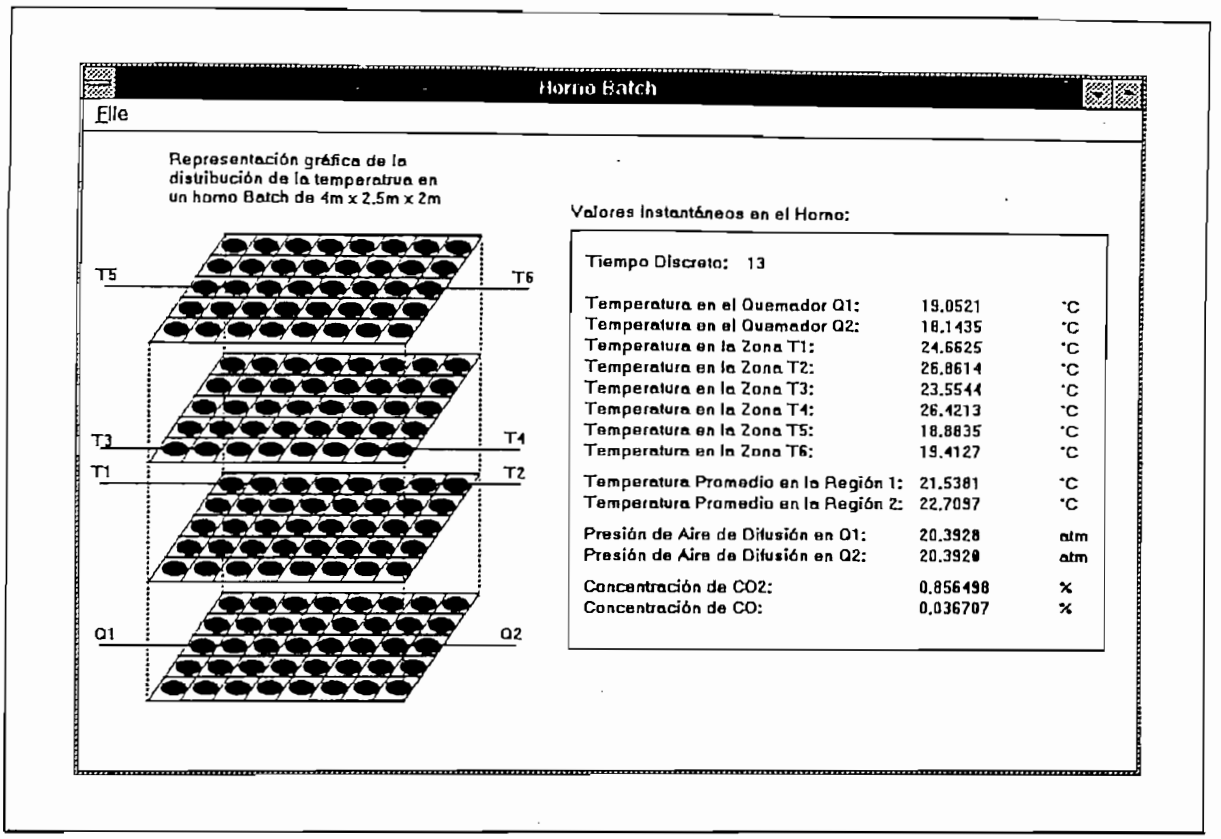


Figura 3.8. Pantalla Principal del programa de simulación de un Horno Batch: BATCH.EXE.

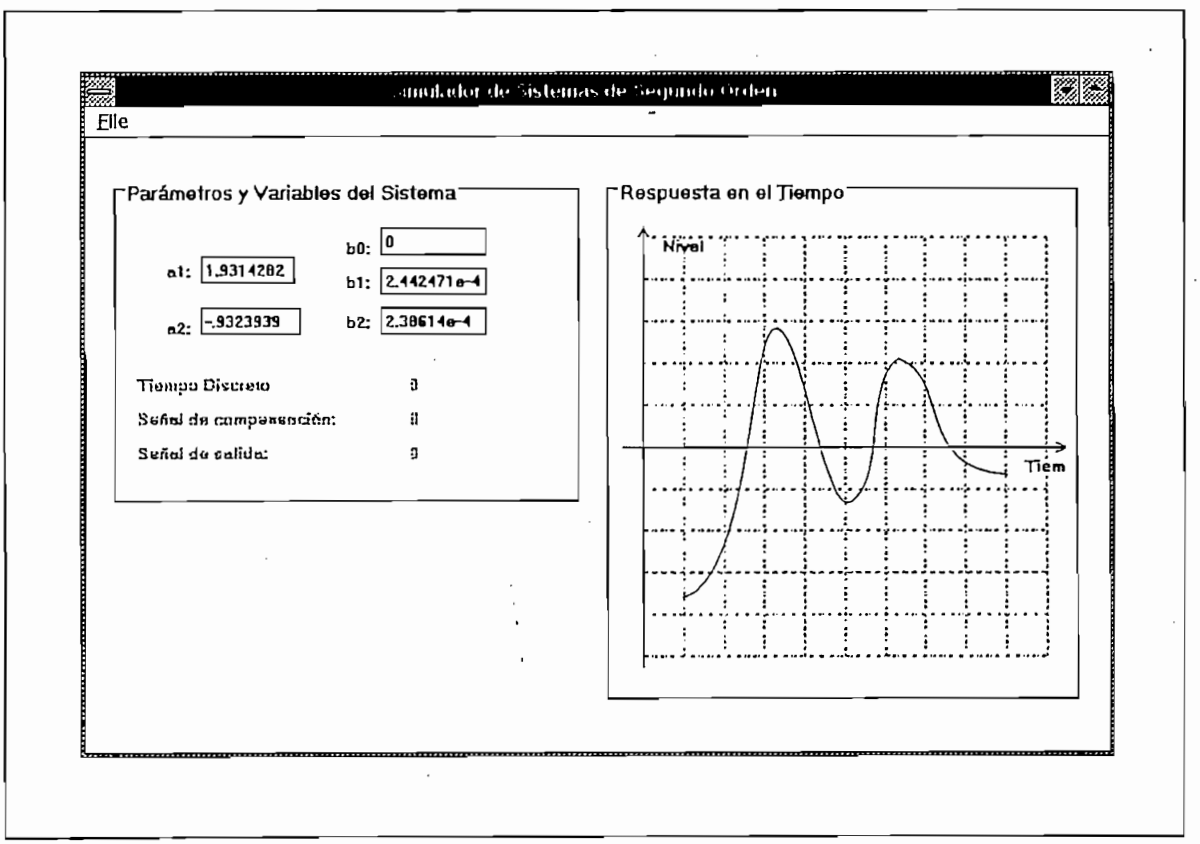


Figura 3.9. Pantalla Principal del programa de simulación SIMULATE.EXE.

3.3.2. Simulador universal de plantas lineales de segundo orden.

Este es un pequeño modelo computacional, que a pesar de no tener la complejidad del anterior, presenta en cambio la versatilidad de simular cualquier sistema lineal, tanto de primer orden, como de segundo, utilizando para ello su correspondiente equivalente en tiempo discreto.

El programa está hecho de manera que el usuario, quien ingresa los coeficientes de la planta, pueda cambiarlos en cualquier instante, inclusive cuando el sistema se encuentre En-Línea. Esto permitirá no solo estudiar el comportamiento de plantas variantes en el tiempo, sino que en combinación con el programa principal del sistema *SCADA* de laboratorio, también se podrán desarrollar controles que se ajusten satisfactoriamente a este tipo de plantas.

El programa desarrollado para este fin se visualiza en la figura 3.9.

3.4. ADQUISICION Y TRANSMISION DE INFORMACION.

Volviendo al tema del Sistema *SCADA* de Laboratorio solo queda por tratar el aspecto que concierne al intercambio de información entre los prototipos al computador; y, después de su tratamiento matemático, del computador de regreso a los prototipos; cerrando así el lazo de control.

Existe un ligero inconveniente a este respecto, puesto que la misma entidad controladora y supervisora debe manejar tanto modelos reales como modelos computacionales, o en mejores términos, debe por un lado, tener la capacidad de acceder a direcciones de puertos de entrada y salida, mientras que por otro, la facultad de acceder a variables externas pertenecientes a los programas que contienen los modelos computacionales. Estas dos

opciones serán tratadas por separado y luego conciliadas bajo un mismo procedimiento.

3.4.1. Adquisición y salida de datos con prototipos reales.

Se ha anticipado que el intercambio de información entre los diferentes prototipos reales y el computador será por medio de la tarjeta de adquisición de datos DAS-128. Al igual que muchos de los equipos que conforman el sistema *SCADA* de Laboratorio, este fue desarrollado en una anterior tesis de grado, razón por la cual no se expondrán sino solo los aspectos fundamentales de la misma. [10][11]

3.4.1.1. Descripción general de la tarjeta DAS-128. La mencionada tarjeta se instala en cualquiera de las ranuras de expansión de 16 bits del computador. No requiere alimentación adicional, pues los 5Vdc que necesita lo suministra el computador a través del slot de expansión. La dirección de la tarjeta se asigna por medio de un conjunto de jumpers localizados en la misma. Normalmente se usa la dirección base 0300H, la cual nunca es utilizada por otros dispositivos omputacionales, eliminando así cualquier posibilidad de conflicto. Sin embargo, pueden asignarse las direcciones base en 0200H, 0210H, 0220H, 0230H, 0310H, 0320H ó en 0330H, pero de hacerlo así habrá que asegurarse que ninguna otra tarjeta utilice tales direcciones.

La tarjeta DAS-128 puede manejar 8 entradas analógicas/digitales de 8 bits, 8 salidas digitales/anlógicas de 8 bits, 8 entradas digitales de 8 bits y 8 salidas digitales de 8 bits. Los pórticos analógicos tienen un rango de operación entre 0 y 10Vdc, correspondientes al intervalo 00H a FFH, es decir, una resolución de 0.039V/bit; mientras tanto los pórticos digitales tienen un valor verdadero en 10Vdc, y falso en 0 Vdc . Las direcciones de dichos pórticos están dadas por:

Desde xx0H hasta xx7H se encuentran las entradas A/D y salidas D/A de 8 bits.

Desde xx8H hasta xxFH se encuentran las entradas y salidas digitales de 8 bits.

Notar que las mismas direcciones se comparten para las labores de entrada y salida de datos, esto es factible puesto que la tarjeta internamente reconoce la operación que se pretende realizar, y automáticamente habilita uno u otro pórtico.

La particular opción que presenta la tarjeta DAS-128 para conformar un puerto digital bidirecional a partir de un puerto digital de entrada y uno de salida no será utilizada por el sistema *SCADA* de Laboratorio.

El pórtico digital en la dirección xxFH será utilizado bit a bit, permitiendo al sistema de control disponer de 8 entradas y 8 salidas digitales de 1 bit. Cabe señalar que la tarjeta en sí no permite acceder bit a bit; por consiguiente, deberá ejecutarse un artificio a nivel de software que permita realizar esta operación.

3.4.1.2. Acceso a los pórticos. La mayoría de lenguages de bajo y alto nivel incluyen funciones que permiten acceder a una dirección particular de memoria y así, a un determinado puerto de la tarjeta. En BASIC se disponen de la función **INP(Port)** y de la sentencia **OUT Port, Value** respectivamente para la lectura y escritura de un byte de información en una dirección específica de memoria. El lenguage C también permite tales efectos con las funciones **inp(Port)** y **outp(Port, Value)** respectivamente. Por su parte, el lenguage ensamblador de los procesadores 80X86 con las instrucciones **in ac,DX** y **out ac,DX** permite ya sea, ingresar al acumulador el valor contenido en la dirección DX, o enviar a dicha dirección el contenido del acumulador.

Sin embargo, los lenguajes de extremo alto nivel como el Visual BASIC 3.0, en el cual será desarrollado el programa de adquisición de datos, carecen de funciones que operen directamente con direcciones específicas de memoria. Por tanto, será necesario escribir en C o código ensamblador, una *librería de enlace dinámico*, o DLL (del inglés Dynamic Link

Library), la cual incluya las funciones requeridas [28]. Esta librería será compilada y enlazada independientemente del módulo en Visual BASIC, pero desde éste, aquella podrá ser invocada.

Las librerías de enlace dinámico .DLL son programas desarrollados específicamente para el entornto Windows. Tienen características muy similares a los programas ejecutables ordinarios .EXE para Windows, pero a diferencia de estos últimos, que son eliminados de la memoria al concluirse su ejecución, éstas permanecerán, pudiendo ser invocadas tantas veces como se requiera. [41] [42] [43]

Cabe resaltar de todos modos, que esta forma indirecta de acceder a las direcciones de memoria toma un tiempo considerablemente mayor respecto a la modalidad ordinaria, incidiendo negativamente en la velocidad máxima de adquisición de datos. Normalmente, con un computador 486 con velocidad de procesamiento de 66 MHz, no es posible realizar más de 100 llamadas por segundo. Dato verificado en laboratorio.

3.4.2. Transmisión de datos con prototipos virtuales.

Los prototipos virtuales, o modelos computacionales, requieren un tratamiento totalmente diferente. Ahora, en vez de acceder a una dirección de memoria fija, deberá rastrearse una variable relocalizable que contenga el valor de una magnitud de proceso producida por el programa simulador.

En un principio esto puede parecer difícil puesto que el entorno Microsoft Windows asigna y reasigna la localización de las variables en memoria dependiendo de factores que escapan al control del usuario. Pero afortunadamente los diseñadores de este famoso sistema operativo han propuesto una sencilla y confiable solución: el *intercambio dinámico de datos* o DDE (Dynamic Data Exchange), alternativa que es soportada y aprovechada, extraordinariamente bien, por las aplicaciones escritas en Visual BASIC 3.0.

3.4.2.1. **Intercambio dinámico de datos.** El intercambio dinámico de datos es un mecanismo de Windows por medio del cual dos aplicaciones pueden establecer una *conversasión* e intercambiar información contínua y automáticamente. Quien inicia la conversación se conoce como aplicación *destino*, mientras que la aplicación que responde se conoce como *fuente*. Para nuestro caso, la aplicación destino es el programa principal del Sistema *SCADA* para Laboratorio, mientras que la aplicación fuente es el programa simulador. [41]

Para establecer el enlace entre los programas de interés, la aplicación destino debe llamar a la aplicación fuente y esperar su aceptación al diálogo. Si este ha sido confirmado, se procede a adquirir información de la aplicación fuente o a enviarla. Existen algunas formas de llevar a cabo esto; una primera, que es manual, requiere la expresa solicitud de la aplicación destino; una segunda, que es automática, inmediatamente actualiza la información en la aplicación destino si la variable en la aplicación fuente cambia repentinamente su valor; y una última, que es semi automática, en la cual la aplicación fuente envía una notificación a la aplicación destino esperando que ésta actualice su contenido. La idoneidad de cada método es relativa a cada tipo de aplicación, mas para los fines requeridos por el sistema *SCADA* de Laboratorio habrá de considerarse el método manual, leyendo y escribiendo datos periódicamente tal como si se invocara una función de lectura o escritura para la tarjeta DAS-128.

Para que la aplicación destino pueda tener acceso a la información de la aplicación fuente es necesario que se refiera al *ámbito* y al *item* que contienen la información deseada, éstos constituyen una pseudo dirección de la variable. El *ambito* está conformado por el nombre registrado de la aplicación fuente y por el nombre de la ventana que contiene los datos. El *item* por otro lado, es el nombre de la variable objeto que en sí contiene la información. La forma de especificar esto en la aplicación destino depende del lenguaje en el que ha sido desarrollada. Para mayores detalles de estos procedimientos referirse a los

Manuales Técnicos o a los Archivos de Ayuda en Línea de los respectivos lenguajes de programación.

El mecanismo de intercambio dinámico de datos presenta de todos modos una falencia, y es que tarda desde algunas décimas de segundos, para las transacciones más sencillas, hasta varios segundos, en aquellos casos de actualización de imágenes. No es novedad que esta lentitud puede poner en riesgo la calidad del control. Sin embargo, dado que el material de intercambio son puros números de simple precisión, se puede garantizar una tasa de dos intercambios perfectos por segundo, uno en cada sentido de la conversación. [41]

Este limitante es parcialmente ignorado debido a que la mayoría de procesos industriales reales tienen constantes de tiempo de varias decenas de segundos, de modo que la frecuencia de muestreo de 1Hz se vuelve relativamente cómoda.

3.4.2.2. Consideraciones especiales en la elaboración de prototipos virtuales para su utilización con el sistema *SCADA* de laboratorio. Los programas de simulación elaborados en esta tesis de grado, así como cualquier otro programa futuro diseñado para operar con el sistema ***SCADA*** de laboratorio requieren un mínimo de infraestructura en el tema de los enlaces dinámicos para garantizar una compatibilidad satisfactoria. Referirse al anexo A, apartado 8.1.

En primer lugar, los programas de simulación deberán ser desarrollados en cualquier lenguaje que soporte la programación en Microsoft Windows. De manera especial se recomiendan las plataformas integradas de desarrollo de software Microsoft Visual BASIC 3.0, Microsoft Visual C++ 1.0, Borland C++ 4.0, o cualquier versión mejorada de estos programas.

Por otro lado, se obliga que todas las variables objeto involucradas en la conversasión estén agrupadas bajo *una sola* ventana madre, pues el sistema de control ***SCADA*** permite

solamente establecer diálogo con una ventana por subsistema virtual. Es importante señalar que dicha ventana debe ser diseñada para aceptar enlaces dinámicos.

Las variables objeto que contengan la información deseada deben ser declaradas como tipo *Label* (Etiquetas) o *TextBox* (Cuadros de Texto). No se reconocerá ningún otro tipo de estructura. Estas variables simularán los puertos de entrada y salida de la tarjeta de adquisición de datos DAS-128.

El Sistema ***SCADA*** de Laboratorio exige la presencia de una variable objeto del tipo *Label* en la aplicación fuente, la cual, pueda ser utilizada como temporizador de dicha aplicación. En cada instante de muestreo el sistema de control aumentará en uno el contenido de esta variable, y ante la sucitación de dicho evento, el programa de simulación deberá realizar los cálculos necesarios para estimar el nuevo estado de la planta. Resulta notoria la obligatoriedad de esta variable objeto, y no es redundancia recalcar la advertencia de que ésta no debe ser alterada intencionalmente por la misma aplicación fuente.

Por último, las aplicaciones fuente deben contener el objeto *Control Box* (Cuadro de Control), y en él deben incluir la opción *Close* (Cerrar), pues el programa central de supervisión y control recurre a este elemento para cerrar la aplicación cuando la planta esté fuera de línea.

En los demás aspectos, los programas de simulación tienen completa libertad. No existen límites al número de ventanas, objetos, o variables en general, sino los que de por sí restrinja el lenguaje de programación o el entorno Windows.

3.4.3. Consolidación de operaciones de adquisición y transmisión de datos.

Aunque la modalidad en la recepción y transmisión de datos es completamente distinta en los casos de prototipos reales y prototipos computacionales, el sistema ***SCADA*** deberá

funcionar sin alteraciones perceptibles por el usuario frente a cualquiera de las dos opciones.

Al iniciar una sesión con el programa de aplicación del sistema *SCADA* de laboratorio deberá especificarse si se utilizarán prototipos reales o prototipos virtuales, y en el caso de estos últimos, deberán estipularse los más indispensables atributos. A excepción de este inevitable hecho, el sistema *SCADA* de laboratorio no hará distinción entre ambas naturalezas, todas las facultades del sistema de control y supervisión serán asequibles por unos y otros.

CAPITULO 4

IMPLEMENTACION DE SOFTWARE PARA SUPERVISION Y CONTROL

4.1. REQUERIMIENTOS DEL SISTEMA.

4.2. DIRECTRICES PARA LA PROGRAMACION.

4.3. MODULOS DEL PROGRAMA.

4.1. REQUERIMIENTOS DEL SISTEMA.

Se ha enunciado con anterioridad que el aparato motriz de todo sistema *SCADA* es el programa computacional que coordina las actividades de supervisión y control de los distintos lazos que lo conforman. El presente capítulo tiene por objeto diseñar, analizar y detallar un paquete de software cuyo exclusivo fin es la gobernación del citado sistema *SCADA* de laboratorio.

Existe un conjunto de condicionantes fundamentales que especifican las características mínimas que debe presentar el computador personal en el cual se pretende instalar el programa en cuestión. Por conveniencia, a éstos se los ha dividido en dos grupos: requerimientos de hardware y requerimientos de software.

4.1.1. Requerimientos de hardware.

Se consideran indispensables las siguientes especificaciones:

- Procesador 80386 de 33 MHz.
- 2 MB de memoria RAM.
- 2 MB libres en disco duro.
- Monitor VGA, SVGA, o cualquier otro compatible con Microsoft Windows.
- Mouse compatible con Microsoft Windows.
- Tarjeta de adquisición de datos DAS-128.

Aunque ésta es la configuración mínima, se recomienda instalar el programa en un computador 80486 de 66 MHz con 4 MB de memoria, pues el desempeño será notablemente mejor, especialmente cuando se comanden varios subsistemas simultáneamente.

La tarjeta de adquisición no es necesaria si solo van a ejecutarse programas de simulación de plantas industriales.

4.1.2. Requerimientos de software.

El programa no podrá correr si no cuenta con la siguiente infraestructura:

- Microsoft Windows versión 3.1 o superior, operante en Modo Enhanced.
- Microsoft Visual BASIC versión 3.0 Profesional, o superior.
- Librería IOPORT.DLL en el directorio \windows\system\ utilizada para entrada y salida de datos [28].

La instalación de la plataforma de desarrollo de software Microsoft Visual BASIC, no es extrictamente indispensable, pero si recomendable en alto grado. La razón para esto es que Microsoft Visual BASIC no permite crear archivos ejecutables autosuficientes (*Stand Alone*), pues además del archivo .EXE que contiene el módulo principal de un determinado programa, siempre es necesaria la presencia de la librería VBRUN300.DLL, y en el particular caso de la aplicación técnica del sistema ***SCADA*** de laboratorio es obligatoria la presencia de los siguientes archivos de Visual BASIC en el directorio \windows\system\: SPIN.VBX, THREED.VBX, CMDIALOG.VBX GRAPH.VBX, GSWDLL.DLL, y GSW.EXE. Así, para evitar conflictos innecesarios se sugiere instalar en forma completa el lenguaje Microsoft Visual BASIC versión 3.0 Profesional.

4.2. DIRECTRICES PARA LA PROGRAMACION.

Antes de entrar a la etapa de programación propiamente dicha, es necesario declarar un conjunto de guías que orienten el desarrollo del sistema ***SCADA*** de laboratorio hacia los intereses que motivan a esta tesis de grado, convirtiendo a dicho programa en un producto

idóneo para el efecto.

En primer lugar, se ve conveniente que el programa y su documentación sean elaborados tanto en idioma inglés como en español; el uno por ser el lenguaje técnico por excelencia y el otro por ser el idioma común en nuestro medio.

En segundo término, el programa deberá ser implementado en completa conformidad a las normas que exige la compañía Microsoft para cualquier programa desarrollado bajo el entorno Microsoft Windows. Es decir, quien utilice el programa deberá beneficiarse con la capacidad gráfica, la compartición de recursos, y la operación multitarea, características intrínsecas de Windows.

Respecto a la interface gráfica desarrollada para el efecto, ésta deberá ser concebida de tal manera, que un usuario corriente de aplicaciones para Windows, pueda navegar en el programa sin dificultad alguna. Mas será necesario que dicho usuario tenga claros los conceptos de los Sistemas de Control Automático para que pueda obtener provecho del programa en cuestión.

Es reconocido el hecho de que una interfaz gráfica correctamente elaborada acelera el trabajo del usuario, minimiza la ocurrencia de errores por utilización inapropiada, y crea en el usuario una confortable sensación de dominio del computador. Por tales razones, todas las ventanas, cuadros de dialogo, y otros objetos de interfaz gráfica contenidos el programa, deberán reflejar amabilidad frente al usuario, sin necesariamente comprometer su imagen de profesionalismo.

Deberá enfatizarse también en la corrección de errores introducidos intencional o descuidadamente por el usuario; pues todo programa en Windows tiene un apropiado sistema de seguimiento de operaciones, y éste no puede ser la excepción.

En cuanto a la documentación, es necesario que sea corta, clara y consistente, pues nunca faltará quien no sepa cómo proceder llegado a un punto determinado del programa, o cómo conseguir un particular efecto con las alternativas que el programa presente. Dicha documentación será principalmente detallada en un Manual de Usuario, y de manera parcial y diluída en los archivos propios del programa.

En lo concerniente a las opciones de menú, deberán incluirse las referentes a: manejo de proyectos, configuración del sistema en aspectos indispensables y optativos, monitoreo de subsistemas en línea y organización de pantallas. Más detalles a este respecto se dan en la sección siguiente del presente capítulo.

Y finalmente, en el orden técnico, el programa deberá manejar un máximo de 16 lazos de control con períodos de muestreo, múltiplos de décimas de segundo, arbitrariamente establecidos. Además, deberá tener la capacidad de manejar dos tarjetas DAS-128, las cuales establecen un total de 16 entradas Analógicas/Digitales de 8 bits, 16 salidas Digitales/Analógicas de 8 bits, 14 entradas Digitales de 8 bits, 14 salidas Digitales de 8 bits, 16 entradas Digitales de 1 bit y 16 salidas Digitales de 1 bit.

4.3. MODULOS DEL PROGRAMA.

Se cuenta ya con todos los criterios teóricos y prácticos que permitirán plasmar en realidad concreta el tan discutido sistema *SCADA* de laboratorio.

A continuación, se describen con mayor o menor detalle, dependiendo de la necesidad, las características de diseño y operación de todos los archivos que conforman la mencionada aplicación.

4.3.1. Archivo de proyecto SCADA.MAK.

Este archivo contiene la declaración de todos los módulos, ventanas y controles que toman parte en el programa ejecutable SCADA.EXE.

4.3.2. Módulo general DEFINE.BAS.

Es el archivo en el cual se hallan las declaraciones de todas las variables globales del programa, la definición de las estructuras de datos, la declaración de funciones externas prototipadas en librerías de enlace dinámico (.DLL) , e igualmente, la definición de constantes del programa.

4.3.3. Módulo general GENERAL.BAS.

Es un módulo que incluye funciones y subrutinas de alcance global necesarias para el funcionamiento del programa; entre éstas, es de trascendental importancia la referente a la inicialización de variables generales.

4.3.4. Forma SCDMAIN.FRM.

Este archivo es la espina dorsal del programa SCADA.EXE, pues contiene la definición de la ventana principal, barra de menú, con sus respectivos procedimientos, y por sobre todo, la rutina de adquisición de datos, control y supervisión.

La ventana principal es del tipo Multi Document Interface Window (Ventana de Interface para Documentos Múltiples), o MDI. Esta es la clase de ventana que es menester definir cuando se requiere agrupar dentro de sí varias ventanas *hijas*.

A esta ventana queda subordinada la barra de menú que está estructurada de la manera

indicada en la figura 4.1:

File	Define	Options	Command	View	Window	Help
New	System Loops ...	Alarms ...	Go ...	Real Time Monitoring	Tile	About
Open ...	Real Time I/O Variables ...	Statistics ...	Go All	Statistical Information	Cascade	
Save	Control Functions ...	Register ...	Halt ...	Alarm Reports	Arrange Icons	
Save As ...	Event Programming ...		Halt All	Control Panel		
Configure ...			Reset All			
Exit						

Figura 4.1. Barra de menú.

El menú File (*Archivo*) contiene todas las opciones referentes al manejo de proyectos SCADA. Así, la opción New (*Nuevo*) permite al usuario crear un nuevo proyecto, definir en él, nuevos lazos de control, nuevas variables de tiempo real, entre otras cosas. La opción Open (*Abrir*) permite acceder a archivos de formato .SCD, que contienen proyectos SCADA creados por el usuario en sesiones anteriores; y como es lógico, también se cuenta con las opciones Save (*Guardar*) y Save As (*Guardar Como*), para almacenar dichos archivos a su tiempo. Existe también la opción Configure (*Configurar*), la cual despliega un cuadro de diálogo en el cual se definen determinados parámetros de hardware necesarios para el funcionamiento normal del programa. Por último, se tiene la opción Exit (*Salir*), la cual permite al usuario abandonar el programa, no sin antes ejecutar determinados procedimientos de seguridad.

El menú Define (*Definir*) invoca a distintos cuadros de diálogo en los cuales se definen las *características indispensables* de un proyecto SCADA. Resulta obvio, que dicho menú debe permitir la definición de los diferentes lazos de control pertenecientes al proyecto, sus respectivas variables en tiempo real, las leyes de control que interactúan con ellas y la programación lineal de los eventos que se espera sucedan durante la operación en línea.

El siguiente menú, Options (*Opciones*), invoca a determinados cuadros de diálogo que permiten al usuario establecer ciertos detalles respecto a la operación en línea. A diferencia del caso anterior, las definiciones hechas en estos cuadros de diálogo no son de naturaleza obligatoria, mas pueden tener mayor o menor grado de trascendencia en el proyecto, dependiendo de la delicadeza de las magnitudes físicas involucradas. De acuerdo a este criterio, en el mencionado menú se han agrupado las opciones: Alarms (*Alarmas*), Statistics (*Estadísticas*) y Register (*Registros*), que permiten declarar para cada variable del sistema, niveles de alarma, evaluaciones estadísticas y medios de registro, respectivamente.

El menú Command (*Comando*) se utiliza para conectar o desconectar, uno o varios subsistemas a la línea de control. Con este propósito se han establecido las opciones Go (*Arrancar*), Go All (*Arrancar todos*), Halt (*Detener*), y Halt All (*Detener todos*). Cabe mencionar que no es posible acceder a cualquier opción de los menúes anteriores, una vez que el sistema se encuentre en línea, pues no es dable una reconfiguración del sistema mientras éste se encuentra en pleno funcionamiento. Por tal razón se ha incluido la opción Reset All (*Restablecer todo*), la cual no solo detiene las actividades de control y supervisión de todos los lazos, sino que saca al sistema fuera de línea para realizar cualquier ajuste por parte del usuario.

Cuando el sistema se encuentre operando en línea, se habilita el menú View (*Visualizar*), el cual, desplegando ventanas especiales permite al usuario observar el comportamiento en tiempo real, el estado de alarmas, y los datos estadísticos de una particular variable. Adicionalmente, si la naturaleza de la variable lo permite, se podrá fijar manualmente el valor de referencia o Set Point a través de una ventana que emula un tablero de control.

En ventanas MDI no puede faltar el menú Window (*Ventana*), que como todo usuario de Microsoft Windows sabe, permite ordenar el contenido de la ventana *Madre* a través de las opciones Tile (*Mosaico*), Cascade (*Cascada*) y Arrange Icons (*Arreglar Iconos*).

Además de la barra de menú, se halla definida una barra de utilitarios, la cual contiene básicamente las mismas opciones que los menúes File (*Archivo*) y View (*Visualizar*). Dicha ayuda gráfica ha sido incluida solo para mejorar la presentación del programa.

La forma SCDMAIN también incluye la declaración de un temporizador, denominado TIMEX, utilizado para producir interrupciones periódicas al procesador, solicitando así su atención cuando se requiera ejecutar tareas de adquisición, control y supervisión. El funcionamiento detallado de este ente de programación se explicará más adelante en el presente capítulo, cuando hayan sido analizados ciertos aspectos concernientes a la definición de parámetros obligatorios y optativos de un proyecto SCADA.

4.3.5. Forma SUMMARY.FRM.

Esta es una ventana del tipo *Hija* subordinada a la ventana SCDMAIN. Cada proyecto SCADA utilizará esta ventana para desplegar un resumen de la información vital en dicho proyecto. En esta ventana se visualizarán los nombres de los distintos lazos de control con sus respectivos períodos de muestreo; los nombres de las variables de tiempo real asociadas a ellos, la dirección de su pórtico de entrada o salida, su rango de operacion y sus debidas unidades; y, adicional a todo esto, los nombres de todas las funciones de control definidas en cada subsistema.

4.3.6. Cuadro de diálogo DLGCONFG.FRM.

Esta es una pequeña ventana que permite al usuario establecer cierta información relacionada con el computador en el cual se opera. Así, se solicita en primer lugar, un nombre para la actual estación de trabajo; a continuación, el número de tarjetas de adquisición DAS-128 y sus respectivas direcciones base; luego, el pórtico de comunicaciones seriales utilizable por el programa; el directorio de disco en el cual se registrarán los historiales de las variables que lo requieran; y en última instancia, el tipo

de prototipos utilizados en la actual sesión de trabajo, pudiendo ser éstos, modelos reales o modelos computacionales, de acuerdo a lo advertido en el capítulo anterior. Ver el cuadro de diálogo mostrado en la figura 4.2.

Cabe mencionar que no es posible detectar la presencia o definir la dirección de las tarjetas DAS-128 vía software, pues éstas no fueron diseñadas con esa opción. Por tal motivo, es completa responsabilidad del usuario verificar que las direcciones verdaderas de las tarjetas coincidan con los valores declarados en este cuadro de diálogo. Si por equivocación o descuido esto no es así, el programa funcionará de manera impredecible cuando entre en línea, pudiendo inclusive afectar seriamente al computador.

4.3.7. Cuadro de diálogo DLGLOOPS.FRM.

Es el cuadro de diálogo invocado con la opción de menú Define/System Loops (Definir/Lazos de Control). Esta ventana permite declarar todos los subsistemas o lazos de control que quedarán bajo el dominio del computador. Ver figura 4.3.

Cada lazo será identificado con un nombre apropiado. No existe restricción alguna en cuanto a los caracteres utilizados, pero sí en su extensión máxima; ésta no puede superar los 64 caracteres.

Además del nombre, es indispensable fijar un período de muestreo característico del subsistema. Este puede variar desde 0.1 segundos en subsistemas reales o 1 segundo en subsistemas virtuales, hasta 600 segundos (10 minutos).

Si el lazo corresponde a una planta virtual, es menester indicar la ruta completa y el nombre del archivo que ejecutará la simulación, y junto con éste, el nombre registrado de la ventana que contiene las variables que serán objeto de intercambio. Adicionalmente, de acuerdo a lo discutido en el capítulo anterior, en este cuadro de diálogo deberá

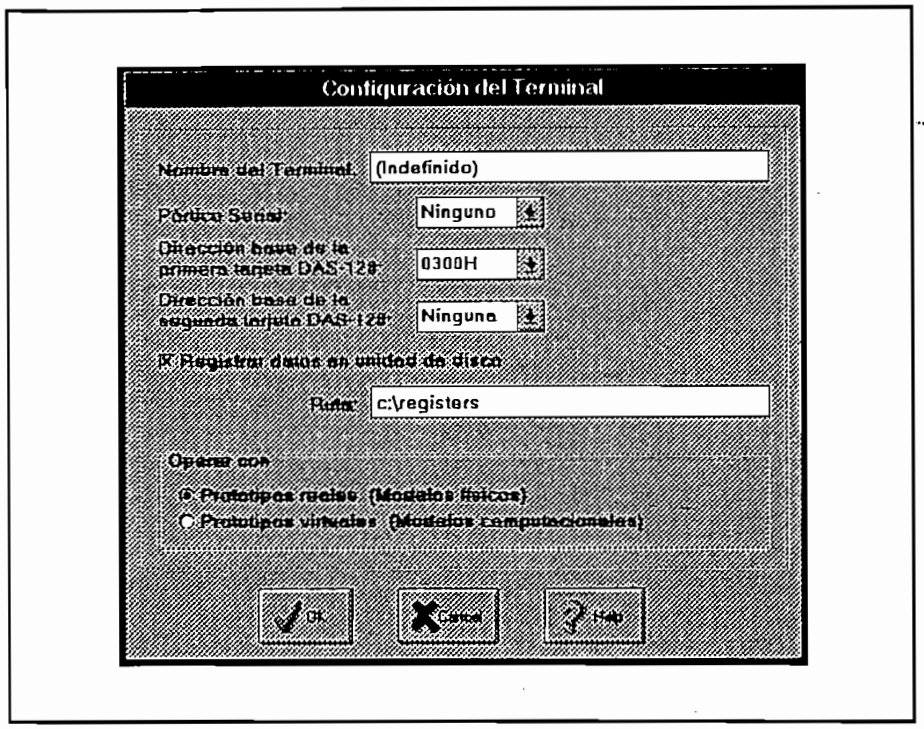


Figura 4.2. Cuadro de diálogo para la configuración del Sistema

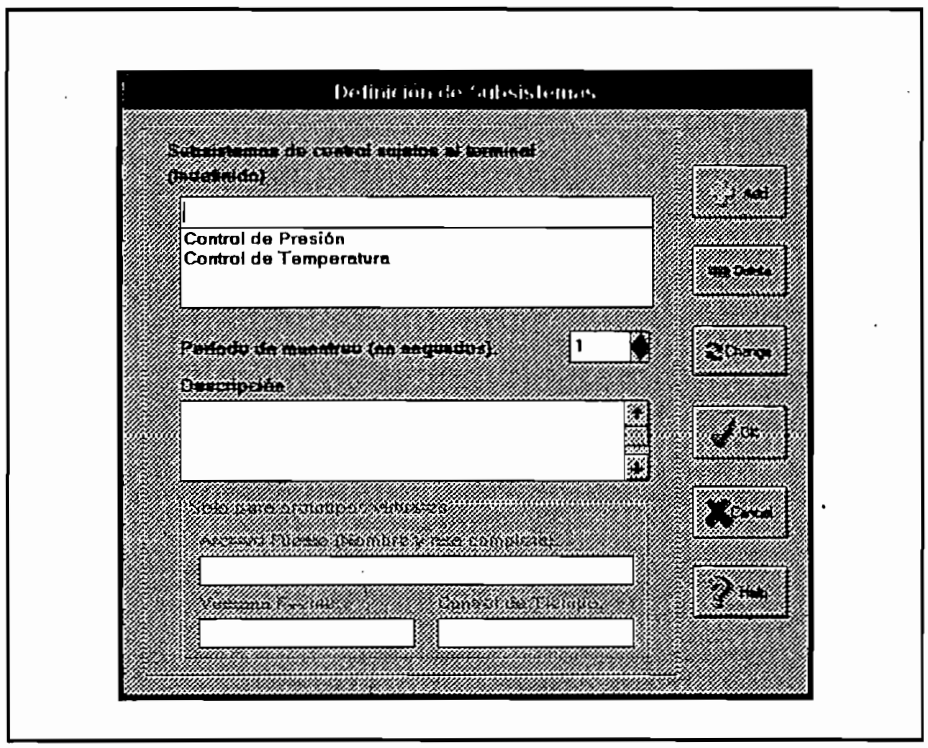


Figura 4.3. Cuadro de diálogo para definición de Subsistemas de Control.

estipularse el nombre de la variable de temporización contenida en el programa de simulación, a la cual el programa SCADA deberá referirse para enviar el valor de tiempo discreto.

Por último, este cuadro de diálogó presenta la opción para introducir un pequeño texto, de no más de 256 caracteres de extensión, que describa brevemente la función del subsistema definido. Este parámetro no es indispensable, pero es útil para mantener informado al usuario acera de los fines y características del subsistema en cuestión.

4.3.8. Cuadros de diálogo DLGVAR.FRM, INPDF.FRM y OPTDF.FRM.

Son ventanas que permiten la definición de las variables de tiempo real sujetas a cada uno de los subsistemas caracterizados con anterioridad en el cuadro de dialogo correspondiente. El archivo **DLGVAR.FRM** contiene el cuadro de diálogo principal, en el cual, cada variable es definida al especificar los siguientes puntos:

- Nombre de la variable, sujeto a las mismas reglas que en el caso anterior.
- Descripción breve de la variable, no indispensable.
- Tipo de variable, seleccionable entre Entrada A/D de 8 bits, Entrada D de 8 bits, Entrada D de 1 bit, Salida D/A de 8 bits, Salida D de 8 bits y Salida D de 1 bit; entendiéndose que el calificativo de *Entrada* o *Salida* viene dado en relación al sistema de control y no a la planta.
- Dirección del pórtico de entrada o salida relativa a la dirección base del dispositivo de adquisición de datos.
- Nombre de la variable del archivo de simulación, especificable solo si la variable pertenece a un subsistema virtual.

El archivo **INPDF.FRM** contiene un cuadro de diálogo para opciones avanzadas de variables de entrada. En éste puede solicitarse la generación de una señal de Set Point y su

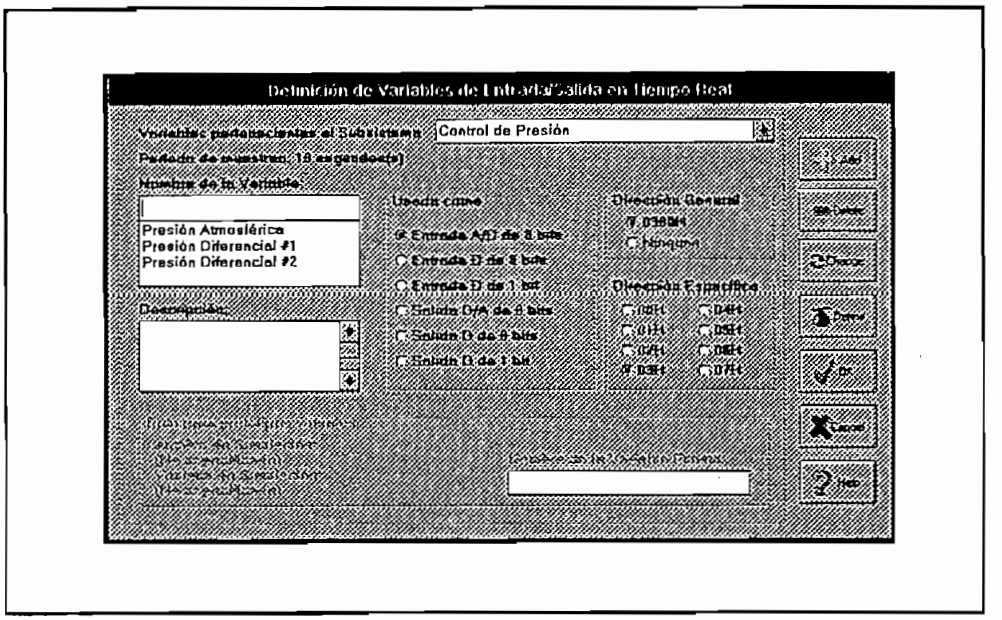


Figura 4.4. Cuadro de diálogo para la definición de Variables de Tiempo Real.

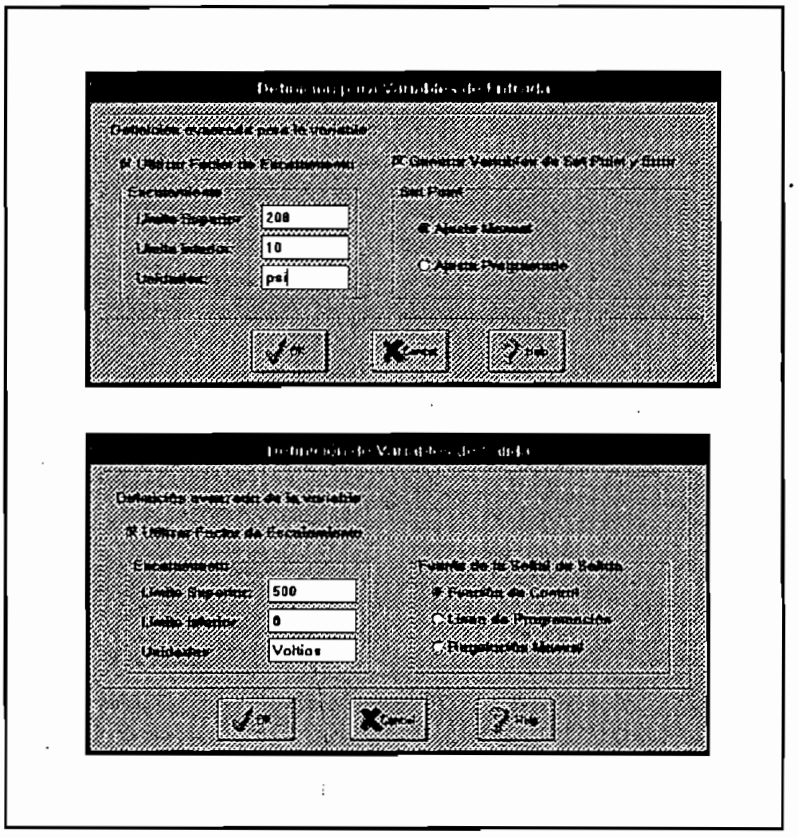


Figura 4.5. Cuadros de diálogo para definiciones avanzadas de Variables de Tiempo Real.

consecuente señal de Error, relativas a una determinada variable de entrada; y de ser así, puede especificarse el origen de tal señal de referencia: Ajuste Manual o Ajuste Programado. Adicionalmente, el cuadro de diálogo citado permite fijar el *rango analógico de operación* de la variable, junto a sus respectivas unidades, es decir, permite establecer el escalamiento correspondiente a la resolución de 8 bits.

Por su parte, el archivo **OPTDF.FRM** contiene una ventana de diálogo similar a la anterior pero con opciones aplicables solo a variables de salida.

Las figura 4.4. y 4.5. ilustran estos cuadros de diálogo.

4.3.9. Cuadro de diálogo DLGCFN.FRM.

Esta es una ventana que permite relacionar una variable de entrada con una variable de salida, a través de una ley o función de control discreto. Para el efecto, las variables deben pertenecer a un mismo subsistema y estar apropiadamente configuradas para este fin. Obsérvese la figura 4.6.

La definición de dicho algoritmo requiere: declaración de un nombre o etiqueta (que se ajuste a las reglas citadas en los casos anteriores), especificación de un tipo básico de control, y de acuerdo a éste, la introducción apropiada de sus respectivos coeficientes.

El tipo de control puede ser uno de los siguientes:

- PID discreto, en cuyo caso se deberán introducir las constantes proporcional, integral y derivativa de un modelo contínuo.
- Red discreta de adelanto o retraso de fase, la cual requerirá la especificación de un cero y un polo en el plano S, y de un valor de ganancia en lazo abierto.
- Algoritmo genérico de tercer orden, en el cual se han de especificar los coeficientes de una

ecuación de diferencias que relaciona la señal de error con la señal de control.

- Control ON/OFF simple, en cuyo caso se exigirá un valor de umbral y una orientación (creciente o decreciente) de la no-linealidad.

- Control ON/OFF con histéresis, que requerirá los valores del umbral superior e inferior, y al igual que el caso anterior, una orientación de la no-linealidad.

Cabe señalar que un mismo par de variables pueden intervenir en la definición de varias leyes de control. No existe verdadero confilicto en esto, puesto que los algoritmos de control no se activan automáticamente cuando el sistema entra en línea, sino que requieren de una señal explícita proveniente del ente supervisor para comenzar a ejercer su función. La siguiente sección aclara este detalle.

4.3.10. Cuadro de diálogo DLGEVENT.FRM.

Esta es una complicada pantalla que permite al usuario programar el desarrollo progresivo de un proceso, en base a simples cláusulas condicionales del conocido tipo *IF (proposición)... ...THEN (acción).*

El cuadro de diálogo se encuentra dividido en tres regiones importantes: la primera de ellas permite definir qúe condición debe ser evaluada por el organismo de supervisión ; la segunda, qué acción se debe llevar a cabo como consecuencia de lo anterior; y la última, que conjuga ambas declaraciones en una sola sentencia y define para ella una adecuada prioridad. Referirse a la figura 4.7.

Con la intención de no caer en innecesarias complicaciones de programación, y bajo la premisa de facilitar al usuario la utilización del programa, el módulo de supervisión incluirá un número restringido de facilidades.

Así, las proposiciones condicionales solo podrán evaluar el cumplimiento de los siguientes

eventos:

- Tiempo de Proceso igual a un valor definido por el usuario.
- Tiempo de Proceso mayor que un valor definido por el usuario.
- Tiempo de Proceso menor que un valor definido por el usuario.
- Tiempo de Proceso comprendido entre dos valores definidos por el usuario.
- Variable de Proceso igual a un valor definido por el usuario.
- Variable de Proceso mayor que un valor definido por el usuario.
- Variable de Proceso menor que un valor definido por el usuario.
- Variable de Proceso comprendida entre dos valores definidos por el usuario.

Se debe recalcar que la variable independiente *Tiempo de Proceso* se refiere al número de décimas de segundo transcurridos desde que el subsistema ingresó a la línea de control. Notar también, que *Variable de Proceso* es cualquier señal de entrada, salida o error, producida por lazos de control subordinados al programa SCADA. Y por supuesto, si se hiciera referencia a una magnitud de 1 bit de resolución, solo será aplicable la cuarta cláusula.

Mientras que en la contraparte, al verificarse alguna proposición de las mencionados arriba, podrá ordenarse la ejecución de cualquiera de las acciones siguientes:

- Fijar una variable de Set Point a un valor definido por el usuario.
- Fijar una variable de Salida a un valor definido por el usuario.
- Habilitar un algoritmo de control definido con anterioridad.
- Deshabilitar un algoritmo de control definido con anterioridad.

Cabe realizar un pequeño comentario respecto a las dos últimas opciones. Puesto que existe la posibilidad de implementar varias leyes de control para un mismo par de variables, al realizar una programación lineal de un proceso, el usuario debe tener la

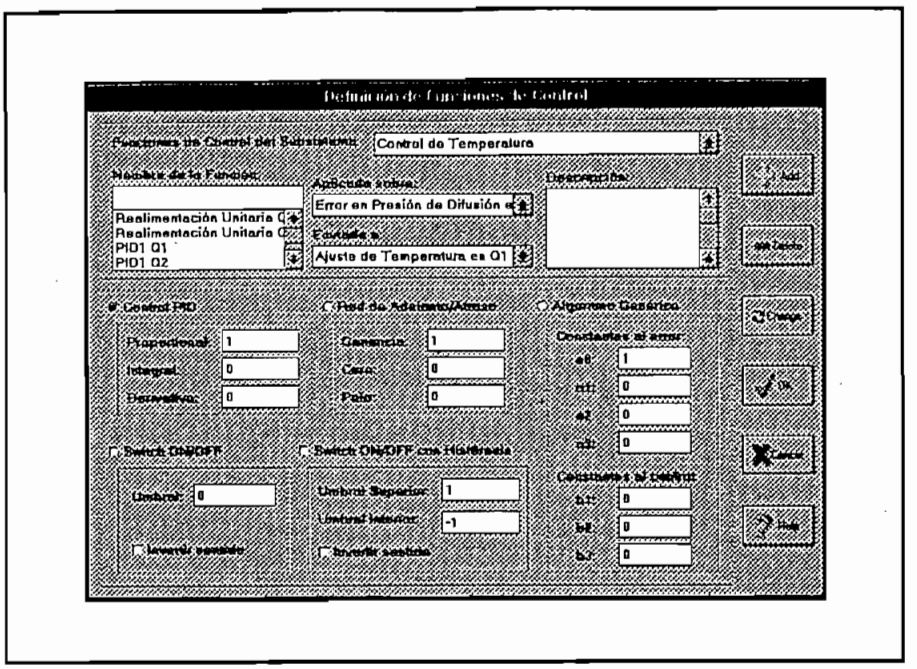


Figura 4.6. Cuadro de diálogo para la definición de Funciones de Control.

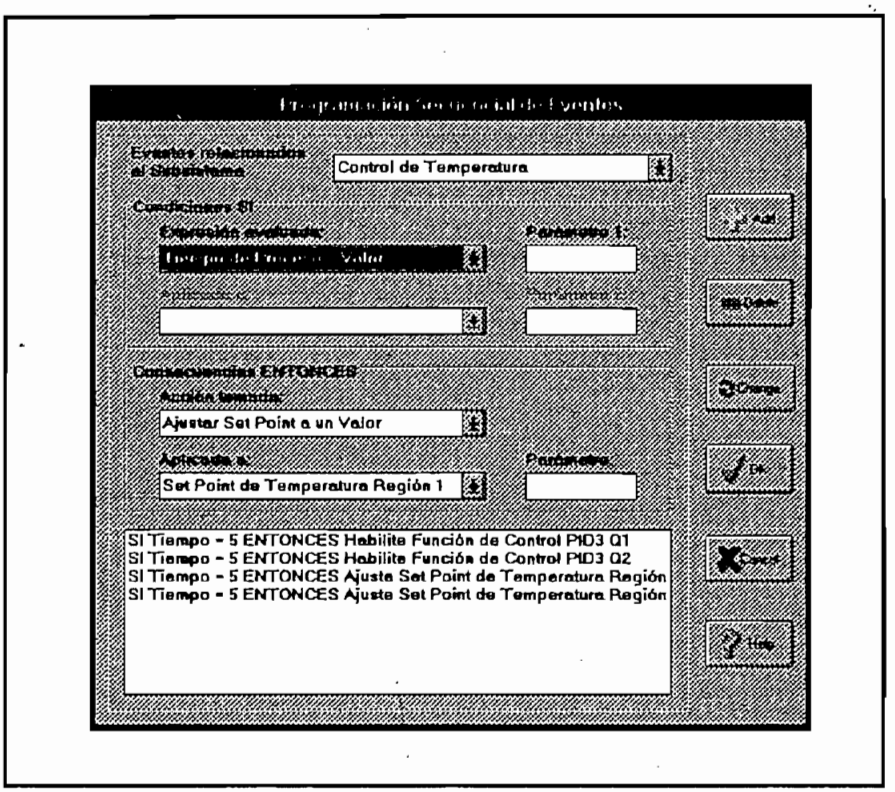


Figura 4.7. Cuadro de diálogo para la definición de Eventos.

precaución de deshabilitar primero cualquier algoritmo de control que se encuentre en funcionamiento antes de habilitar algún otro; de no hacerlo así, aún cuando el sistema de control no corre riesgo alguno, el proceso en cuestión no evolucionará de acuerdo a las perspectivas supuestas.

Un último aspecto a tratar apunta hacia la manipulación de eventos que se sucitan simultáneamente, y que de manera disyuntiva buscan ejecutar una acción sobre una misma variable. Estos casos, cabe mencionar, son producto de una descuidada e incoherente programación lineal, pero no por ello pueden ser pasados por alto por el organismo supervisor; mas aún, se incurre en la obligación de diseñar el mecanismo idóneo que permita sobrellevar contingencias de esta naturaleza.

El programa SCADA permite realizar esto de manera considerablemente simple. Las proposiciones *IF ... THEN* construidas con la indicada interfaz gráfica son enviadas a una lista de acceso dinámico (parte inferior del anterior cuadro de diálogo), la cual permite al usuario ordenar dichas proposiciones según su prioridad de ejecución. Así, cuando por conciente o inconciente error se verifica el cumplimiento simultáneo de dos o más eventos que desencadenarán acciones antagónicas sobre una misma variable de proceso, se recurre a esta lista y se ejecuta *únicamente* la sentencia con mayor prioridad.

4.3.11. Cuadro de diálogo DLGALARM.FRM.

Esta ventana está dedicada a la definición de alarmas para variábles analógicas o digitales que tengan 8 bits de resolución. Naturalmente, no es obligatorio fijar niveles de alarma para que al sistema le sea permitido entrar en línea, pero de hecho, es una práctica altamente recomendable. Ver figura 4.8.

El programa SCADA permite establecer cuatro niveles de alarma a lo largo del *rango analógico de operación* de la variable, éstos son:

- Alarma en el nivel superior de emergencia,
- Alarma en el nivel superior de alerta,
- Alarma en el nivel inferior de alerta, y
- Alarma en el nivel inferior de emergencia.

4.3.12. Cuadro de diálogo DLGSTAT.FRM.

Se ha mencionado en capítulos anteriores que un sistema supervisor debe realizar evaluaciones estadísticas sobre los datos históricos de una variable. Precisamente este cuadro de diálogo permite al usuario declarar qué funciones estadísticas deben ejecutarse sobre el historial de cada variable de interés. Las opciones incluyen: Valor Medio, Valor Máximo, Valor Minimo, Desviación Estándar y/o Distribución de frecuencias, y son solo utilizables en variables de 8 bits de resolución [46]. Ver figura 4.9

4.3.13. Cuadro de diálogo DLGREGIS.FRM.

El programa SCADA permite enviar el valor instantáneo de las variables de proceso, tanto de entrada, como de salida, de 1 u 8 bits, a tres dispositivos formales: pantalla gráfica, archivo de formato serial en disco duro o disco flexible, y/o pórtico serial de comunicaciones. Obsérvese la figura 4.10.

La escritura en disco flexible o diskette, aunque es una alternativa factible, no es en absoluto conveniente, pues las operaciones de lectura o escritura toman tiempos exageradamente largos, que incidirán desfavorablemente en la rapidez del control, y por ende, en su estabilidad.

4.3.14. Cuadro de diálogo DLGSELEC.FRM.

Esta es una pequeña pantalla que aparece ante las órdenes de menú Command/Go o

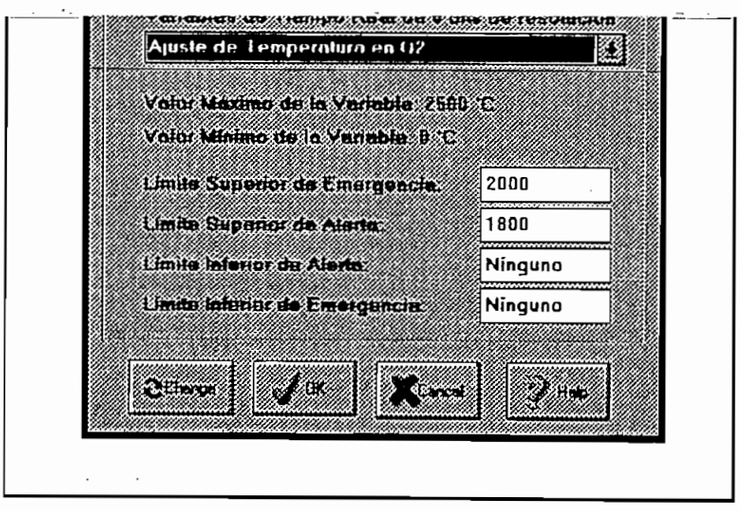


Figura 4.8. Cuadro de diálogo para la definición de Alarmas.

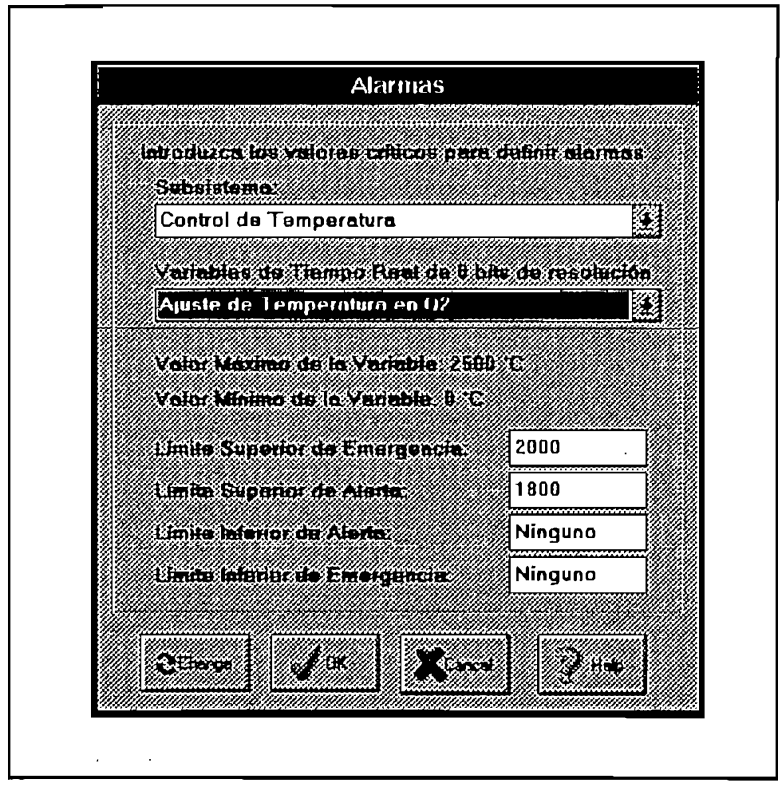


Figura 4.8. Cuadro de diálogo para la definición de Alarmas.

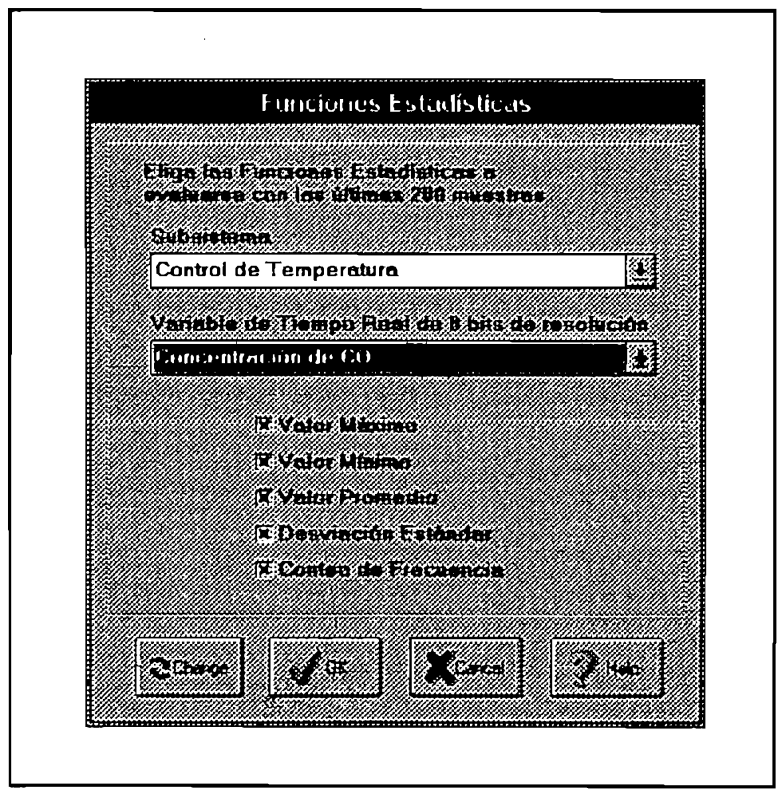


Figura 4.9. Cuadro de diálogo para la selección de Funciones Estadísticas.

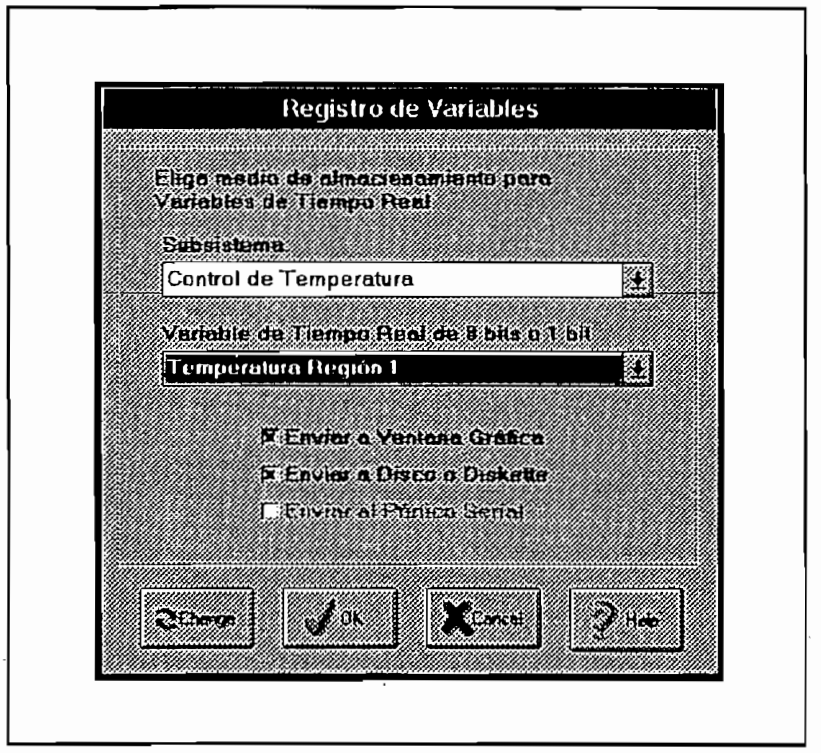


Figura 4.10. Cuadro de diálogo para la selección de Medios de Registro.

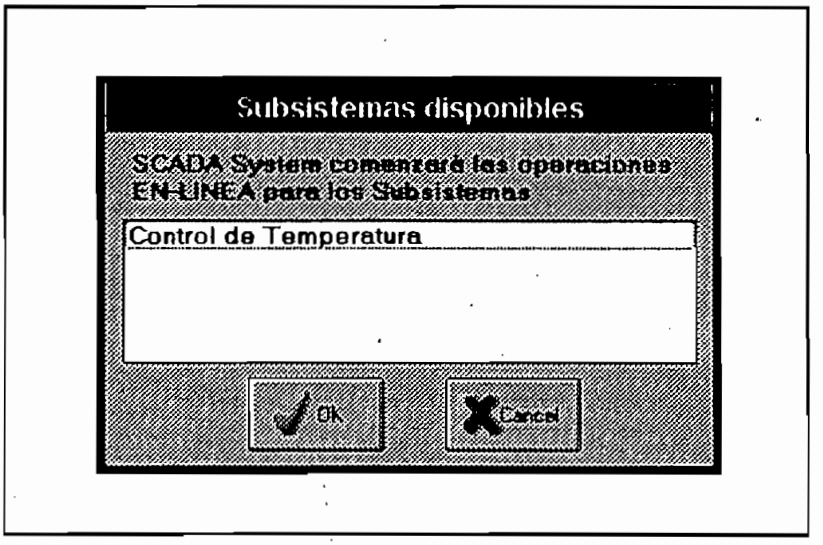


Figura 4.11. Cuadro de diálogo para la selección de Subsistemas.

Command/Halt, que tiene como singular propósito, permitir al usuario escoger de entre los subsistemas definidos, cuáles entran o salen de la línea de control, según sea el caso. La figura 4.11 ilustra este cuadro de diálogo.

4.3.15. Forma FRMONLIN.FRM.

Cuando el sistema entra en línea, el programa crea y asigna una ventana de este tipo a cada lazo de control definido. En ella se visualizarán los valores de tiempo real de todas las variables que le pertenezcan; la información incluye el estado de saturación que tienen las variables, es decir, una indicación respecto a si éstas intentan sobrepasar los límites de 0 y 10 Vdc entre los cuales opera la tarjeta de adquisición de datos.

Una lista de variables es expuesta en cada ventana, de la cual, el usuario puede elegir alguna en particular y solicitar información más detallada. Dicha información incluye: gráfica de su evolución con el transcurso del tiempo, valor de sus coeficientes estadísticos, lista de alarmas ocurridas, y de permitirlo su naturaleza, modificación manual por parte del operador a través de un pseudo panel de control.

4.3.16. Ventanas flotantes GFGRAPH.FRM, GFSTAT.FRM, GFALARM.FRM, GFCONT.FRM y GDMSG.FRM.

Se consideran ventanas flotantes aquellas que han sido definidas con la opción ALWAYS_ON_TOP, exclusiva de la verisión 3.1 de Microsoft Windows [44] [45]. Como de hecho lo confirma el nombre de este particular atributo, las ventanas que lo poseen tienen la posibilidad, de presentarse *siempre por encima* de todas las otras ventanas visibles en un determinado momento, mientras dure su existencia en pantalla. Sin importar qué acción realice el usuario, estas ventanas no pasarán a planos secundarios, exigiendo así una atención permanente de su parte.

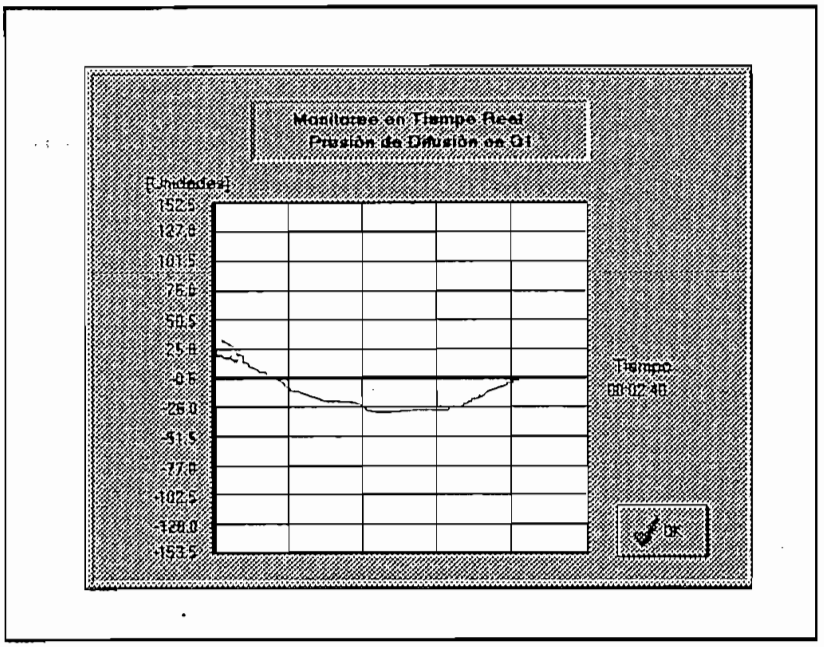


Figura 4.12. Ventana para la graficación en tiempo real.

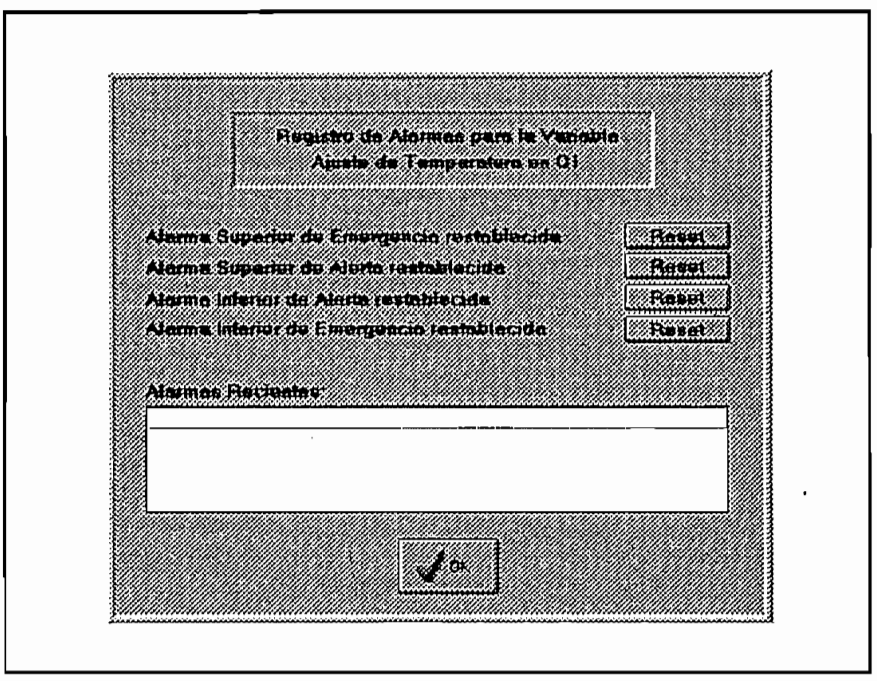


Figura 4.13. Pantalla para reporte de alarmas.

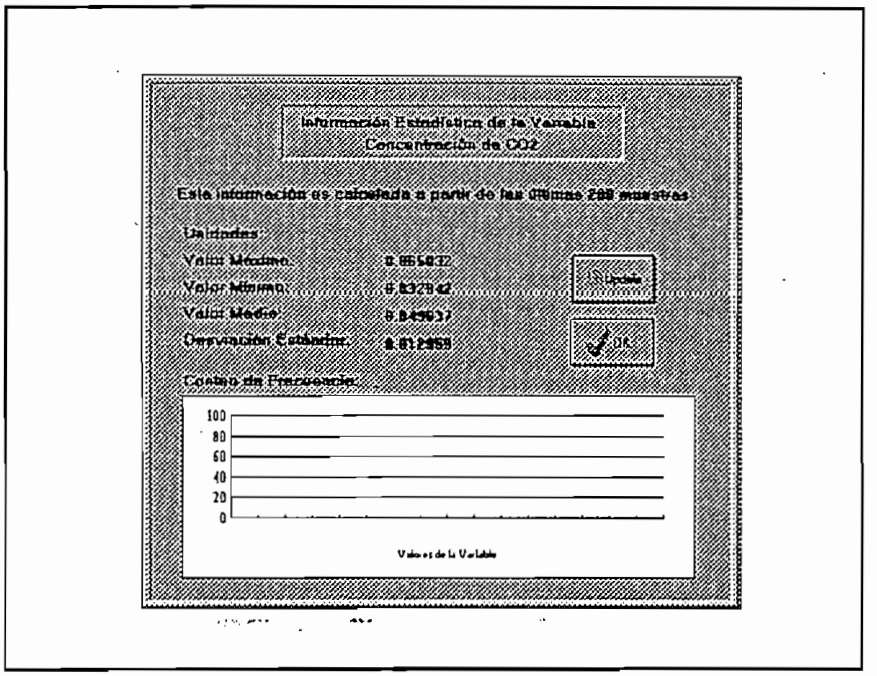


Figura 4.14. Ventana para la visualizacion de parámetros estadísticos.

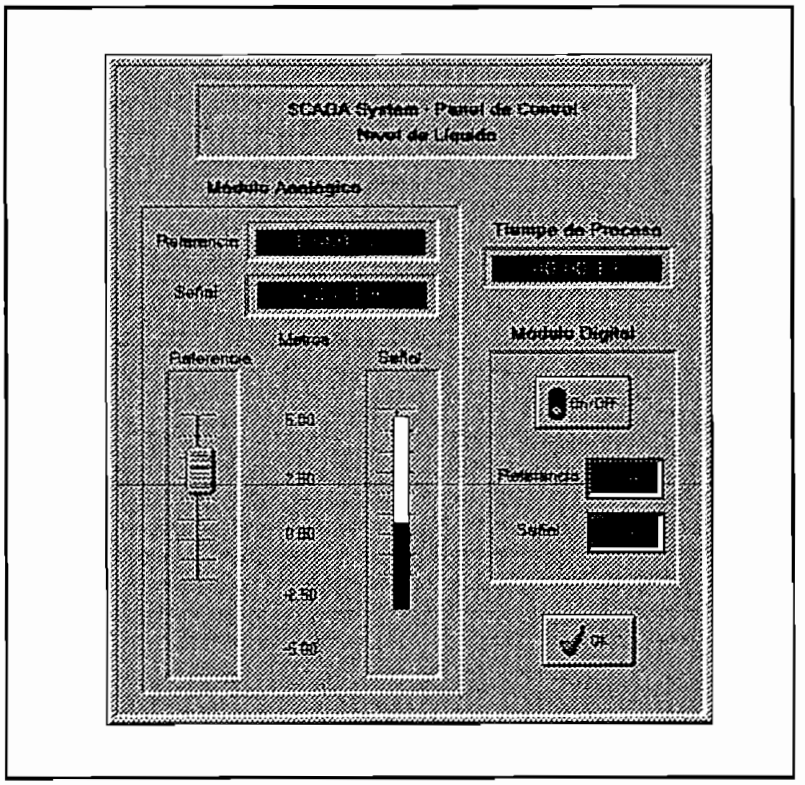


Figura 4.15. Pantalla que emular un Panel de Control.

Estas ventanas han sido diseñadas con extremo cuidado, de manera que la maniobra sobre variables de tiempo real sometidas al programa SCADA, resulte extraordinariamente sencilla. En el respectivo orden señalado, las mencionadas ventanas sirven para: graficación en tiempo real de una variable, cálculo de parámetros estadísticos, registro de la ocurrencia de alarmas, ajuste manual de los niveles de consigna y generación de mensajes por parte del ente supervisor, según se indica en las figuras 4.12, 4.13, 4.14 y 4.15.

4.3.17. Temporizador TIMEX.

Conforme a lo indicado en el apartado 4.3.4., TIMEX es un objeto de Visual BASIC que ha sido condicionado para solicitar periódicamente, al sistema operativo Windows, la dedicación de ciclos de procesamiento que permitan llevar a cabo las pesadas tareas de cálculo implicadas en la adquisición de datos, control digital y supervisión de subsistemas.

La *rutina de interrupción* procesada en tales condiciones es por demás complicada, pues la versatilidad que el programa ofrece al usuario en la definición de distintos lazos de control, cada uno de ellos, con su particular período de muestreo, número indeterminado de variables de tiempo real, y variedad de opciones estipuladas en los cuadros de diálogo previamente discutidos, incide irremediablemente en dificultad de su implementación.

Para comprender perfectamente la manera en que Windows maneja las interrupciones por temporización dadas en una aplicación, se requiere dominar el modo en que dicho sistema operativo administra los recursos del computador, o por lo menos, conocer las sutilezas a las que han recurrido los diseñadores Windows para lograr un funcionamiento armónico de diferentes paquetes de software bajo un hardware siempre variante. Cualquier discusión detallada sobre este tema queda fuera del alcance de esta tesis, mas aquello considerado como necesario para la comprensión cabal del presente programa será mencionado a continuación.

Las aplicaciones de Windows funcionan bajo una óptica distinta a la de los tradicionales programas de DOS. En DOS, cada programa ejecutado se apropia inevitablemente de todos recursos del sistema, volviendo casi imposible la operación multitarea. En Windows, por el contrario, los programas comparten todas las potencialidades del computador, y entre ellas, el tiempo de procesamiento. Esto es posible ya que es el sistema operativo en sí quien administra los recursos, y no cada programa en particular.

Esta concepción conlleva inevitablemente a un nuevo tipo de programación, conocida como programación para el tratamiento de eventos, en la cual, cada aplicación debe hacer una solicitud a Windows para utilizar un particular recurso del computador en un instante determinado. Es entonces el mecanismo administrativo de Windows quien acredita o desacredita tal solicitud, y de aceptarla, envía de retorno a la aplicación un mensaje apropiado que dará acceso al solicitado recurso. Bajo este esquema, el tiempo de procesamiento dedicado a una aplicación no se comporta como una excepción. Así, cuando una aplicación produce una interrupción de tiempo, se envía primeramente a Windows una solicitud de atención, si ésta es consentida, Windows replica a la debida aplicación con el mensaje WM_TIMER.

De regreso a la aplicación, el mensaje no es atendido inmediatamente, sino que se posiciona al final de una cola de mensajes pendientes (si los hubiera), los cuales, después de haber sido tramitados uno a uno, darán paso a la verdadera interrupción. [8] [52] [53] [54]

El problema aún no termina allí, pues las aplicaciones desarrolladas en Visual BASIC presentan un comportamiento singular: Si eventualmente a la cola de mensajes pendientes llegan dos o más mensajes WM_TIMER, producidos por un mismo objeto temporizador o por diferentes, Windows *consolida* dichos mensajes y se procede a ejecutar el primero de ellos. En otros términos, una interrupción no puede interrumpir a otra. [41] [42]

Esta inevitable *burocracia* del sistema Windows pone un límite a la frecuencia máxima en las interrupciones, y para el particular caso de los temporizadores de Visual BASIC, no se tendrá acceso a más de mil interrupciones por segundo. En principio esta cifra es tentadora e ilusionante, pero la complejidad de las operaciones realizadas por el programa SCADA en cada ciclo de interrupción, disminuirá dramáticamente esta posibilidad a 20 interrupciones por segundo, si se utilizan plantas reales; y, a 2 interrupciones por segundo, en el caso de plantas simuladas. Esta información ha sido extraida de pruebas de laboratorio, tomando como objeto de experimentación un computador 80386 de 33 Mhz de velocidad y 2 Mbytes de memoria RAM.

En base a estos resultados, el programa se ha diseñado para que el temporizador **TIMEX** haga a Windows dos solicitudes de interrupción por ciclo de control. Durante la primera, no se realizará cálculo alguno, pero su existencia es crucial, puesto que así se obliga a Windows a procesar cualquier evento pendiente en el programa SCADA, fruto de alguna acción del usuario sobre la interface gráfica. Esta maniobra normalmente no es necesaria, pero hay que prevenir cualquier monopolización del tiempo del procesamiento por parte de otras aplicaciones, especialmente de los programas de simulación, que al procesar sus no ligeros cálculos ingenieriles pueden retener la atención del procesador central y del coprocesador matemático por más tiempo del disponible.

En la segunda interrupción, por el contrario, se realizan todas las actividades de adquisición, control, y supervisión relacionadas con los subsistemas en línea. El procedimiento para ello es el siguiente:

El primer paso consiste en verificar el cumplimiento del período de muestreo de todos los lazos de control definidos. Luego, dando acceso a aquellos que lo soliciten, se procede a recaudar el estado de las variables de entrada, ya sea por medio de la tarjeta DAS-128 o a través de los enlaces dinámicos; y acto seguido, se calculan las señales de error correspondientes.

A continuación, se evalúan las condiciones necesarias de todas las sentencias de la programación secuencial, ejecutando la acción pertinente en aquellas que sean calificadas como verdaderas. La lista de prioridades señalada en el apartado 4.3.10., orientada a resolver casos conflictivos, debe ser tomada en cuenta en este preciso instante.

El paso siguiente consiste en calcular y realimentar la señal de control, utilizando para esto una lista de las funciones de transferencia hábiles al momento.

Después de estas acciones se evalúa el estado de alarma de todas las variables y en todos sus niveles, y de ser necesario, se informará al usuario sobre la aparición de cualquier síntoma irregular, a través de la ventana de mensajes GDMSG.FRM.

Finalmente, se actualizará el contenido de todas las ventanas FRMONLIN.FRM activas, y de requerirlo, también el contenido de las ventanas flotantes GFGRAPH.FRM, GFALARMS, GFSTAT.FRM, y GFCONT.FRM.

Hasta aquí, lo referente al programa SCADA.

La información contenida en este capítulo está orientada a la descripción de criterios de diseño, implementación y funcionamiento del programa de supervisión y control SCADA. Sin embargo, existe información pormenorizada referente a la óptima utilización de la mencionada aplicación, que no tiene lugar en esta tesis de grado, sino que se encuentra registrada en el respectivo Manual de Usuario. Se sugiere entonces, recurrir a dicho manual para satisfacer cualquier otra inquietud.

CAPITULO 5

ENSAYOS, RESULTADOS Y CONCLUSIONES

5.1. ENSAYOS SOBRE EL SISTEMA *SCADA* DE LABORATORIO

5.2. VALIDACION DE RESULTADOS

5.3. CONCLUSIONES Y RECOMENDACIONES

5.1. ENSAYOS SOBRE EL SISTEMA *SCADA* DE LABORATORIO.

Para considerar exitoso el diseño y la implementación del paquete computacional SCADA System es menester someterlo a evaluaciones de distinto tipo y verificar así las características que en el plano teórico se han supuesto. Las pruebas a realizarse comprenden tres aspectos: verificación de las comunicaciones entre el sistema de control y los prototipos, comprobación de las funciones de supervisión y control, evaluación de la operación con prototipos reales y virtuales complicados. La presente sección tiene por objeto solo enunciar las pruebas realizadas, mas los resultados de las mismas serán resumidos en la sección 5.2.

5.1.1. Pruebas de comunicación.

La naturaleza radicalmente diferente de los prototipos reales y virtuales, implica que las pruebas de transmisión de datos, desde y hacia el sistema *SCADA*, deben ser realizadas por separado, utilizando mecanismos apropiados para cada una de ellas.

Así, el acceso a modelos físicos a través de la tarjeta DAS-128 puede ser comprobado utilizando una sencilla red eléctrica RC de segundo orden. Los resultados obtenidos pueden ser generalizados a cualquier otra planta que requiera la interfase de la mencionada tarjeta.

Por su parte, para verificar los enlaces dinámicos requeridos con cualquier modelo computacional se utilizará el programa de simulación de plantas de segundo orden expuesto en la sección 3.3.

5.1.2. Pruebas sobre las funciones de control y supervisión.

Los mismos prototipos utilizados para comprobar las comunicaciones pueden ser considerados para realizar este nueva secuencia de pruebas. Los parámetros a evaluarse son los siguientes:

- Configurabilidad del sistema *SCADA*, es decir, facilidad en la definición de subsistemas de control y de toda característica esencial para la operación del sistema.
- Calidad de los algoritmos de control, lo cual incluye a todas las alternativas propuestas en la aplicación: reguladores PID, redes de adelanto y atraso de fase, algoritmos genéricos y controles no-lineales.
- Programabilidad y funcionalidad de las tareas de supervisión, entre las cuales se considera: monitoreo de eventos, seguimiento de alarmas, evaluaciones estadísticas y registro permanente de información.

Es fundamental que cada una de estas pruebas se realice por separado, de este modo se evitará el efecto que pueda tener una característica particular del sistema *SCADA* sobre el desempeño de otra.

5.1.3. Pruebas con prototipos complejos:

La intención de estas pruebas es evaluar el comportamiento del sistema *SCADA* bajo circunstancias difíciles y delicadas. De este modo, ya no solo se califica el rendimiento del sistema bajo una u otra perspeciva exclusivamente, sino que se logra evaluar el desempeño del sistema *SCADA* como una entidad completa. Para el efecto se utilizarán dos

prototipos: el Sistema de Control de Nivel de Líquidos y el Horno Batch; ambos descritos en la sección 3.3 de la presente tesis.

5.1.3.1. Pruebas en el sistema de control de nivel de líquidos. El problema fundamental de este sistema consiste en seguir una señal de set point, que el usuario establezca arbitrariamente, de la manera más rápida y suave que permita el control *SCADA*. Se busca además, advertir al operador la superación intempestiva de niveles críticos en la cota de agua, de modo que éste pueda ejecutar las acciones correctivas apropiadas. Se requiere también, registrar la información de las variables de tiempo real de forma permanente en un archivo para su análisis fuera de línea.

5.1.3.2. Pruebas en el horno batch. Conforme a lo mencionado en la sección 3.3, los hornos de cerámica requieren elevar, mantener y disminuir la temperatura de su interior, de acuerdo a una curva de set point previamente definida. Para este ensayo se requiere entonces definir una trayectoria referencial de temperatura en base a una programación secuencial. El valor real de la temperatura del horno debe ser una curva suave pero congruente con la trayectoria de set point establecida; y para el efecto, se deben diseñar los algoritmos de control necesarios. Por otro lado, dada la complejidad del prototipo y la delicadeza del proceso, se requiere el sensado permanente de todas las variables de trasendencia; y cuando amerite, la notificación sobre la evolución del proceso hacia un estado de alerta o emergencia. Al igual que el caso anterior, se exige el almacenamiento en disco de todas las variables involucradas, con el fin de analizar la evolución de dichas magnitudes, a lo largo de la secuencia de operaciones establecida.

Resulta evidente que éstos son precisamente aquellos realizados sobre el Sistema de Control de Nivel de Líquidos y el Horno Batch.

5.2.1. Resultados de la operacíon del sistema de control de nivel de líquidos bajo el dominio del sistema *SCADA*.

La configuración del sistema es sencilla. Solo tiene una variable de entrada, con su respectiva señal de referencia, y una variable de salida, cuyo valor se calcula en base a una

5.2. VALIDACIÓN DE RESULTADOS.

En laboratorio se realizaron todas las pruebas indicadas en la sección 5.1, y los resultados fueron satisfactorios. Se hicieron numerosas pruebas en plantas reales y virtuales, de primer y segundo orden. No se considera necesario exponer los resultados de cada uno de los experimentos, sino únicamente de aquellos que se califiquen como representativos. Resulta evidente que éstos son precisamente aquellos realizados sobre el Sistema de Control de Nivel de Líquidos y el Horno Batch.

5.2.1. Resultados de la operacíon del sistema de control de nivel de líquidos bajo el dominio del sistema *SCADA*.

La configuración del sistema es sencilla. Solo tiene una variable de entrada, con su respectiva señal de referencia, y una variable de salida, cuyo valor se calcula en base a una ley de control. El período de muestreo del sistema se fijó en 0.1 segundos.

En un primer intento se utilizó un algoritmo PID sin tener mayor éxito, pues aunque la señal de nivel de líquido presentaba una característica muy buena, ésta solo se conseguía con una señal de control oscilante e incluso saturada. Ver figuras 5.1 y 5.2.

Se intentó también con un algoritmo genérico parametrizado como control incremental, pero no se logró conseguir estabilidad en la señal de control realimentada a la planta. Se realizaron varios intentos con éstos y otros controladores, pero ninguno dió el resultado esperado.

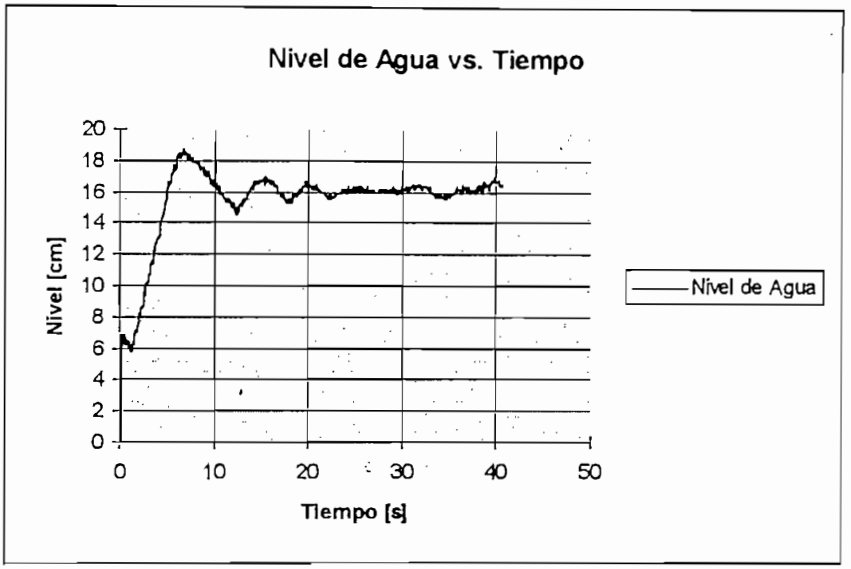


Figura 5.1. Nivel de agua en el primer tanque utilizando un algoritmo PID.

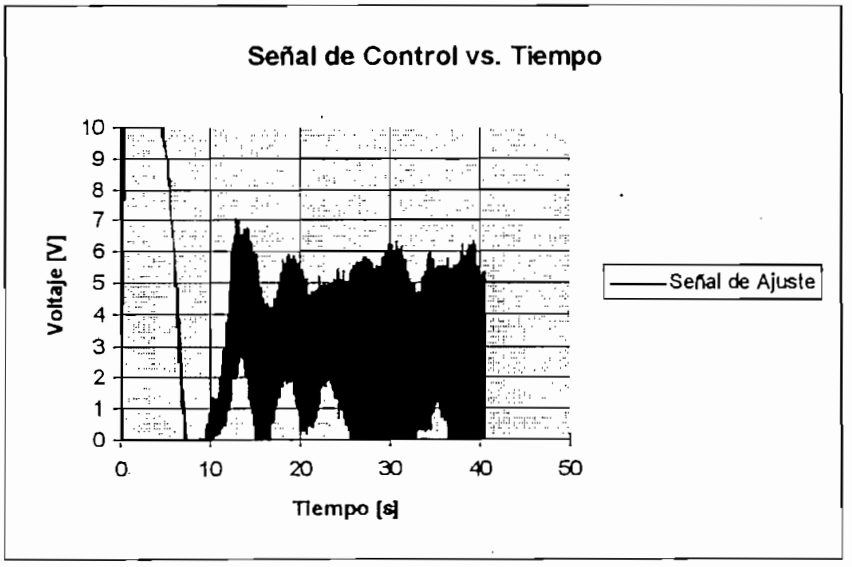


Figura 5.2. Señal de actuación realimentada a la bomba de agua utilizando un algoritmo PID.

Se llegó a la conclusión que no es factible mejorar la señal de actuación debido a que ésta es calculada en base a un error que nunca se estabiliza. La razón de esta inestabilidad es el ruido de cuantización introducido por la poca resolución de la tarjeta de adquisición de datos. El valor de tal señal de ruido es 0.039 V/bit, o en este caso, 0.16 cm/bit. Si a este valor de incertidumbre se suma el efecto del ruido ambiental y la imprecisión del dispositivo de medida, se consigue un límite de inestabilidad equivalente a 0.4 cm, valor determinado en laboratorio.

Finalmente, se consideró la posibilidad de acondicionar un control no-lineal adaptable a la magnitud del error. Utilizando el módulo para programación de eventos de la aplicación **SCADA System** se codificó un algoritmo de alto nivel establecido por las sentencias:

- SI Error de Nivel de Agua > 3 ENTONCES Ajustar Señal de Actuación a 10V
- SI 3 > Error de Nivel de Agua > 2 ENTONCES Ajustar Señal de Actuación a 5V
- SI 2 > Error de Nivel de Agua > 1 ENTONCES Ajustar Señal de Actuación a 4V
- SI 1 > Error de Nivel de Agua > 0.5 ENTONCES Ajustar Señal de Actuación a 3.5V
- SI 0.5 > Error de Nivel de Agua > -0.5 ENTONCES Ajustar Señal de Actuación a 3V
- SI -0.5 > Error de Nivel de Agua > -1 ENTONCES Ajustar Señal de Actuación a 2V
- SI Error de Nivel de Agua < -1 ENTONCES Ajustar Señal de Actuación a 0V

El resultado fue un control en extremo rápido y suave; y, su estabilidad se garantiza ya que para una señal de actuación de 3V, la misma cantidad de agua que inyecta la bomba, se evacúa por la tubería de salida.

Los resultados de este eperimento se ven en las figuras 5.3 y 5.4.

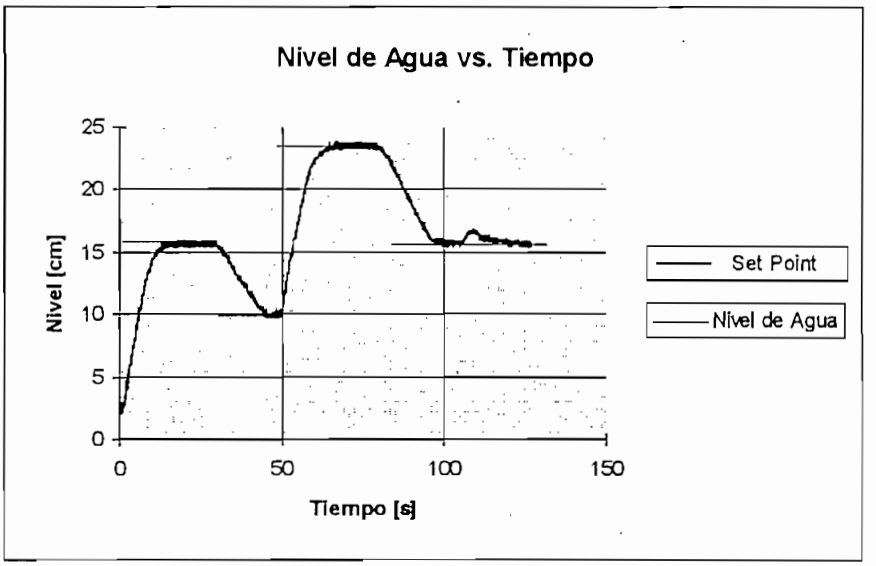


Figura 5.3. Nivel de agua en el primer tanque utilizando un algoritmo no-lineal adaptivo.

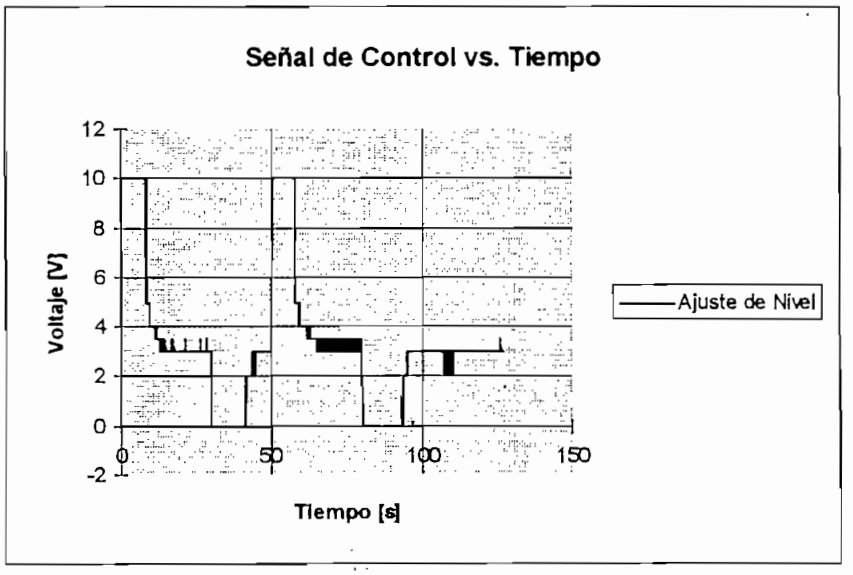


Figura 5.4. Señal de actuación realimentada a la bomba de agua utilizando un algoritmo no-lineal adaptivo

5.2.2. Resultados de la operación del horno batch bajo el dominio del sistema SCADA.

A diferencia del caso anterior, este es una planta multivariable, en la cual, son de interés las siguientes magnitudes: Temperatura de los quemadores Q1 y Q2; Temperatura de las termocuplas T1, T2, T3, T4, T5 y T6; Temperatura promedio de las regiones R1 y R2; Presión de aire de difusión P1 y P2; y, por último, concentración de CO y CO_2 en la tubería de escape. El sistema fue configurado para adquirir y enviar datos con un período de 5 segundos.

La programación de la curva de set point fue hecha en base a 30 sentencias cronológicas similares a: *SI Tiempo de Proceso = 1000 ENTONCES Ajustar Set Point de Temperatura a 400°C*. Un grupo de ellas se encarga de elevar progresivamente la temperatura del horno hasta 1600°C en pasos de 200°C; operación que toma 8 minutos aproximadamente. Las restantes se encargan de ejecutar el proceso inverso. En su globalidad, el proceso dura aproximadamente 30 minutos.

Se hicieron pruebas del sistema en lazo abierto para sintetizar un modelo dinámico de la planta. Ver sección 3.3.1. En base a este modelo se estimaron los coeficientes de los controles PID que se intervendrían en la estabilización de la señal de temperatura en las diferentes etapas del proceso.

Las figuras 5.5, 5.6, 5.7 y 5.8 muestran el comportamiento de las variables anteriormente citadas, con el transcurso del tiempo. De esta manera se pone en evidencia el éxito del control *SCADA* en este experimento.

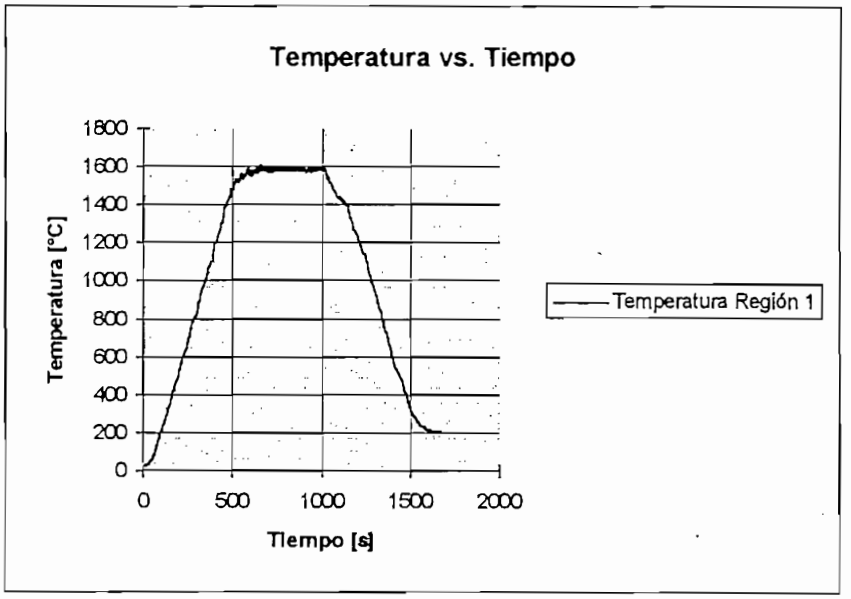


Figura 5.5. Curvas de Temperatura en el Horno Batch.

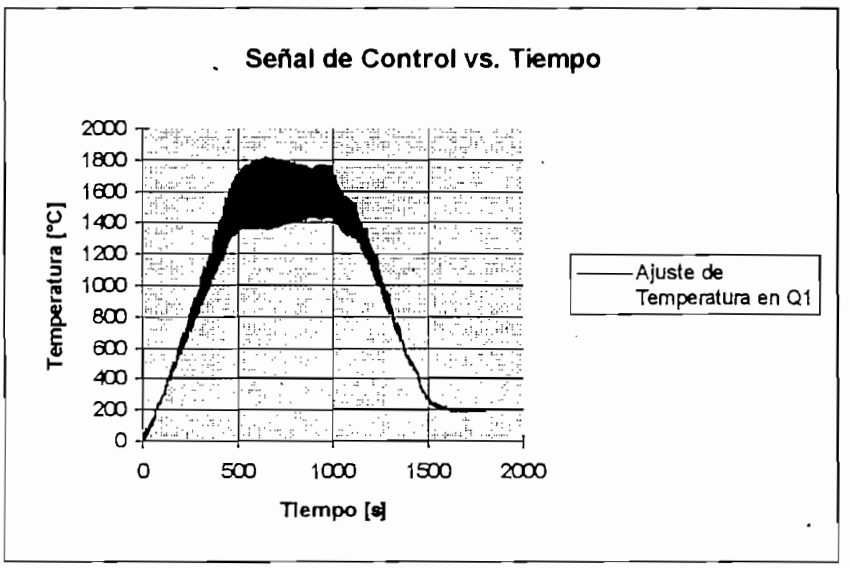


Figura 5.6. Señal de Control realimentada al Horno Batch.

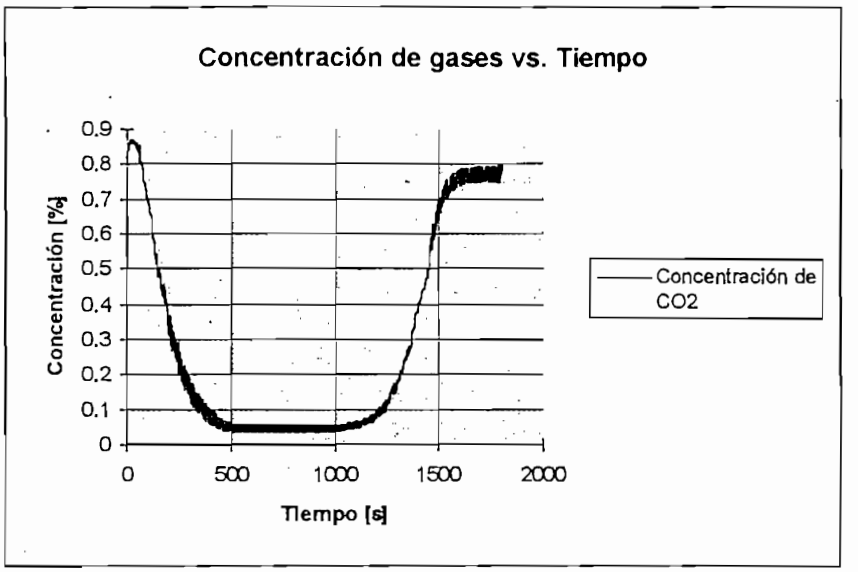


Figura 5.7. Actuación de las válvulas de difusión del Horno Batch.

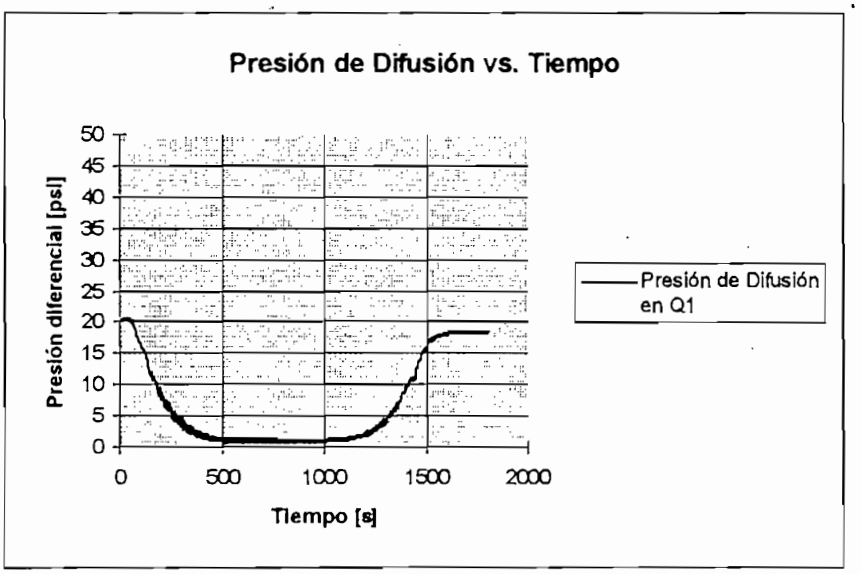


Figura 5.7. Actuación de las válvulas de difusión del Horno Batch.

Resulta exageradamente complicada la descripción de todas las sesiones de trabajo ejecutadas en el programa SCADA System para la realización de cada una de las pruebas. Pues para llegar a los resultados expuestos en esta sección ha sido necesario primero configurar la estación de trabajo, definir los lazos de control, variables de tiempo real, funciones de control, secuencia de operaciones y ciertas características adicionales para el correcto funcionamiento del sistema; a continuación, registrar los resultados, analizarlos fuera de línea, corregir los parámetros necesarios y reiniciar el proceso. Este ciclo de operaciones se repitió tantas veces como fueron necesarias para optimizar el funcionamiento de las plantas. El apartado A.7 de los anexos describe una sesión de trabajo típica con la aplicación SCADA System; ésta proporcionará una clara noción del extenso trabajo realizado para llevar a buen término estos experimentos.

5.3. CONCLUSIONES Y RECOMENDACIONES.

Tras las etapas de diseño, implementación, aplicación y evaluación del sistema de *SCADA* laboratorio se puede concluir y recomendar lo siguiente:

5.3.1. Respecto a los sistemas *SCADA* en general.

- Los sistemas *SCADA* son los entes de control más generalizados y amplios que pueden implementarse para la operación de cualquier planta industrial. Son muy pocas las tareas que un sistema *SCADA* perfectamente diseñado es incapaz de ejecutar. No importa qué tan grande o difícil sea el proceso que se comanda, un sistema *SCADA* siempre presentará alternativas versatiles, flexibles y confiables para la gobernación de plantas industriales.

- En relación a las formas clásicas de control, los sistemas *SCADA* permiten afrontar situaciones no usuales, ante las cuales, las primeras irremediablemente fallarían. Esto se debe a que la estructura *SCADA* es una integración inteligente de todas las modalidades de control existentes bajo una sola premisa: el óptimo desarrollo del proceso.

- Para conformar un sistema *SCADA* se requieren dispositivos de adquisición, transmisión y control. Pero el aspecto central del sistema *SCADA*, alrededor del cual funcionan armónica y coherentemente todos los equipos involucrados, es el programa computacional desarrollado para las funciones de supervisión y control. Por tanto, no es extraño el hecho de que en un proyecto *SCADA* el rubro más costoso sea precisamente el software técnico de aplicación.

5.3.2. Respecto al sistema *SCADA* de laboratorio.

- Enfocando la atención al caso particular del sistema de control supervisor y adquisición de datos desarrollado para la gobernación de prototipos de laboratorio, es necesario afirmar que ha desempeñado idóneamente todas las funciones para las cuales fue diseñado.

- El Laboratorio de Sistemas de Control cuenta ahora con una herramienta apropiada para la operación de plantas industriales de pequeña escala, que han sido (o serán) desarrolladas sin el adecuado soporte computacional.

- Sin embargo, la elaboración de un programa tan universal y genérico, como es el paquete **SCADA System**, necesariamente implica la pérdida de aptitud para afrontar circunstancias específicas de cada planta subordinada. Aún cuando ésta sea una limitación

que puede sortearse con una programación de nivel profesional, queda establecido el precio que se paga por tener un sistema de control tan solvente: cantidad o calidad. Cuál de los dos parámetros preferir, depende exclusivamente de las necesidades del proyecto. En este caso se ha considerado la operación *moderadamente satisfactoria* de prototipos de *diferente* clase, prioritaria al *sobresaliente* desempeño de *un solo* prototipo.

- De entre los aspectos más exitosos del sistema *SCADA* de Laboratorio se encuentra la factibilidad de realizar simples programaciones secuenciales que permitan gobernar el desarrollo de un proceso. Esta característica particular permite afrontar situaciones complicadas ante las cuales, los controles convencionales simplemente fallan. Tal es el caso de la programación de controles adaptivos no-lineales desarrollados para la regulación del nivel de agua en uno de los prototipos probados.

- Otra notable ventaja de la mencionada aplicación es la amigable y elegante interface gráfica implementada para llevar a cabo una sesión de trabajo *SCADA*. No es necesario ser un experto para manejar el programa con destreza razonable; basta conocer el ambiente Windows y tener claros los conceptos el control digital.

- En resumen, la aplicación técnica SCADA **System** cumple con las características más importantes de un organismo *SCADA*, que son: configurabilidad del sistema, estabilidad y calidad de los controladores, programabilidad de eventos y previsión de fallas, maniobrabilidad sobre las plantas involucradas, manipulación y presentación adecuada de información; y por supuesto, facilidad de uso.

5.3.3. Respecto a la programación en Windows de aplicaciones de tiempo real.

- Con relación a la calidad del sistema operativo Windows para los fines propuestos, vale la pena mencionar que favorece al software *SCADA* solo de forma parcial. La excelente interfaz gráfica que puede ser desarrollada en la aplicación es lo más apropiado para rutinas de configuración y supervisión, pero para tareas de control digital directo, ésta incide desfavorablemente. Las vistosas ventanas de todo programa en Windows son elementos que consumen ingentes recursos del computador, haciendo de la adquisición, control, y salida de datos, un proceso de importancia secundaria.

- El mayor inconveniente de Windows para el control digital directo es la manera en que éste atiende las interrupciones de temporización. La conocida *cola de mensajes* de una aplicación de Windows, tan útil en otros aspectos, es un serio tropiezo cuando se pretende desarrollar rutinas de control digital de respuesta casi instantánea.

- Por supuesto, ésta es una desventaja del sistema Windows 3.1 exclusivamente, ya que el novedoso Windows 95 promete mejorar el funcionamiento de las aplicaciones al instituir la operación multitarea entendida ésta en el más riguroso sentido.

- Por su parte, el entorno de desarrollo de software Visual BASIC 3.0 Profesional es un excelente herramienta para la fabricación de elaboradas pantallas gráficas, pero es un verdadero obstáculo para la programación en tiempo real. Primeramente, no existen funciones propias del lenguaje que permitan acceder a localidades arbitrarias de memoria, y en segundo lugar, los temporizadores de Visual BASIC son elementos de operación restringida.

- Sin embargo, para la implementación del programa SCADA System se contó con la ayuda de profesionales en la rama de la computación que aportaron valiosamente con distintas alternativas para superar los inconvenientes mencionados. Así fueron desarrolladas las librerías de enlace dinámico (.DLL) requeridas para el acceso al bus de direcciones.

- De la investigación realizada se presume también que la relativa deficiencia de Windows y Visual BASIC para la implementación de rutinas de control en tiempo real puede ser superada hasta cierto punto si se desarrollan en Visual C++ controles exclusivos de Visual BASIC (librerías .VBX) para la adquisición y salida de datos en tiempo real.

- Otra alternativa consiste en desarrollar hardware y software DMA (Acceso Directo a Memoria) para adquirir y enviar datos hacia los prototipos. Pero ésta es una opción que se aleja de uno de los objetivos de la presente tesis, que es la integración del equipo *actualmente* existente en el laboratorio.

- El desarrollo, aplicación y *contrastación* de los diferentes mecanismos de control en tiempo real operantes bajo los beneficios y restricciones de Windows, es en sí un tema de investigación completo, por lo cual, en esta tesis no han sido tratados sino solo con la profundidad requerida para desarrollar un sistema ***SCADA*** a escala de laboratorio.

- Las razones hasta aquí señaladas son suficientes para considerar que una aplicación de tiempo real desarrollada en Windows no puede tener una frecuencia de barrido extremadamente alta. Teóricamente el límite superior para ésta es 1000 Hz; mas dependiendo de la complejidad de las rutinas de adquisición, control y supervisión, y de la

potencia del equipo utilizado, dicho límite puede reducirse considerablemente. En el caso de la aplicación desarrollada en esta tesis se ha establecido el conservador valor de 10Hz como la frecuencia máxima de barrido para sistemas reales, y 1Hz para sistemas virtuales.

- Por tanto, ante sistemas con constantes de tiempo críticamente cortas, no hay otra solución sino recurrir a la ayuda de controles autónomos dedicados, que de manera directa actúen sobre la planta. Esta opción implica la creación de un nivel más en la estructura jerárquica de control, en la cual el organismo supervisor del sistema *SCADA* ocupa el rango más elevado.

- Felizmente, la mayoría de variables industriales son de evolución lenta, lo cual posibilita la implementación de un sistema *SCADA* sin controladores auxiliares para casi cualquier tipo de proceso reproducido en laboratorio.

5.3.4. Respecto a la preparación profesional en sistemas *SCADA*.

- Por la importancia que han adquirido los sistemas *SCADA* en el país en los últimos años, se considera oportuno profundizar el contenido de la materia Control Computarizado en el ámbito de los sistemas *SCADA*.

- Se sugiere que los estudiantes de la especialidad de Electrónica y Control realicen prácticas de laboratorio en las materias afines con el software de aplicación desarrollado en esta tesis.

- Como parte de la instrucción que reciben los estudiantes de la mencionada especialidad

se puede proponer adicionalmente el desarrollo de modelos computacionales, compatibles con la aplicación SCADA System, que representen plantas industriales de elevada complejidad. El beneficio de esto será evidente en los temas de diseño, modelación, simulación y compensación de sistemas.

- Además de los prototipos descritos en esta tesis, pueden formar parte del sistema *SCADA* de laboratorio, cualquier otro prototipo que utilice la tarjeta DAS-128. Particular referencia se hace al módulo experimental Motomátic, el cual, con la intervención de una entidad de control autónoma como la estación Microlab 51, puede fácilmente incorporarse a la estructura *SCADA*.

- Se considera de vital importancia para la Escuela Politécnica Nacional desarrollar varios programas de post grado, entre los cuales, tengan prioridad aquellos orientados a la automatización industrial; pues la coyuntural situación del país a este respecto obliga a que las instituciones educativas de nivel superior aporten con profesionales sobresalientes que sepan afrontar los retos de una industria en desarrollo.

5.3.5. Respecto a afinamientos en la operación del sistema *SCADA* de laboratorio.

- Se recomienda instalar la aplicación SCADA System en computadores presonales 486 de 66Mhz (o superiores), que tengan un mínimo de 4 MB de memoria RAM. Aún cuando ésta no es la infraestructura mínima para correr el citado programa, se considera la configuración óptima para sacar verdadero provecho del sistema *SCADA* de laboratorio.

- Se exorta también a la utilización de tarjetas de adquisición de datos con resolución

mayor a 8 bits, ya que así, se reduce el ruido de cuantización y se mejora la calidad del control. El programa SCADA System, en términos generales, puede soportar cualquier otro dispositivo de adquisición y salida de datos que tenga un modo de direccionamiento similar al de la tarjeta DAS-128, es decir, direccionamiento de los pórticos de entrada y salida mediante mapeo de memoria.

- Finalmente, para aplicaciones de complejidad superior a la de aquellas descritas en esta tesis, posiblemente requieran de un computador con características específicas y una versión mejorada de la aplicación SCADA System.

REFERENCIAS BIBLIOGRAFICAS.

1. ABB, Product Information Manual: "S.P.I.D.E.R. MicroSCADA", revision eigth, ABB Stromberg Power Oy, London, 1986.

2. Banda H., "Sistemas Expertos y Redes Neuronales", Escuela Politécnica Nacional, Facultad de Ingeniería de Sistemas, Quito, 1994.

3. Bayas J., Tesis de Grado: "Desarrollo del Control Supervisor para el Laboratorio de Instrumentación", Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica, Quito, 1992.

4. Bennett S., Linkens D., "Real Time Computer Control", IEE Control Engineering Series 24 - Comprehensively revised edition, Peter Peregrinus Ltd., London, 1985.

5. Bennett S., Linkens D., "Industrial Digital Control Systems", IEE Control Engineering Series 37 - Comprehensively revised edition, Peter Peregrinus Ltd., London, 1988.

6. Benitez D., Tesis de Grado: "Diseño y Construcción de un Sistema de Control de Nivel de Líquidos", Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica, Quito, 1994.

7. Black U., "Data Communication and Distributed Networks", Prentice Hall, New York, 1993.

8. Borland, "Borland C++ for Windows Reference", Borland International Inc., California, 1994.

9. Brito R., "Sistemas Expertos aplicados a la Industria", Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica, Quito, 1992.

10. Cevallos F., Tesis de Grado: "Diseño y Construcción de una Tarjeta de

Adquisición de Datos para Computadores Personales", Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica, Quito, 1993.

11. Cevallos F., "Manual de Usuario de la Tarjeta DAS-128", Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica, Quito, 1993.

12. Creuss A., "Instrumentación Industrial", segunda edición, Editorial Marcombo, México, 1981.

13. Dorf R., "Sistemas Modernos de Control. Teoría y Práctica", Addison-Wesley Iberoamericana, segunda edición, California, 1989.

14. EEQ, Memorias Proyecto: EEQ - SYSTEMS EUROPE - EPN, "Diseño de Unidades Terminales Remotas", Qutio, 1990.

15. Eggebrecht L., "Interfacing to the IBM Personal Computer", Mac Millan Computer Publishing, California, 1990.

16. EPN, Seminario: "Hardware de Computadores y Redes", Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica, Qutio, 1995.

17. EPN, Seminario: "Redes de Computadores", Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica, Qutio, 1995.

18. EPN, Curso: "Control Automático", Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica, Qutio, 1993.

19. EPN, Curso: "Control con Microprocesadores", Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica, Qutio, 1993.

20. EPN, Curso: "Instrumentación Industrial", Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica, Qutio, 1993.

21. Fisher M., "Power Electronics", Thompson Information Publishing Group, Boston, 1991.

22. Franklin G., Powell D., "Digital Control", Addison - Wesley Publishing

Company, California, 1980.

23. Fu K., González R., Lee C., "Robótica: Control, Detección, Visión e Inteligencia", McGraw - Hill, Madrid, 1994.

24. Fustillos R., Tesis de Grado: "Algoritmos de Control Digital Directo", Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica, Quito, 1994.

25. Godfrey T., "Lenguaje Ensamblador para Microcomputadoras IBM", Prentice Hall Hispanoamericana, México, 1991.

26. Guzman R., Tesis de Grado: "Sistemas SCADA en Distribución de Energía Eléctrica", Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica, Quito, 1993.

27. Hernández J., Burbano P., Jornadas de Ingeniería Eléctrica y Electrónica: "Programa de entrenamiento para ajuste del regulador PID en procesos industriales de control de nivel de líquidos", Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica, Quito, 1994.

28. Hinojosa M., "Control supervisor de temperatura de un caldero asociado a un intercambiador de calor", Escuela Politécnica Nacional - Universidad Andina Simón Bolivar, Quito, 1995.

29. Holman J., "Heat Transfer", fourth edition, McGraw - Hill Book Company, New York, 1976.

30. IEE, "International conference on Distributed Computer Control Systems", IEE Conference Publication No. 153, 1977.

31. IEEE, "Fundamentals of Supervisory Control Systems - IEEE Tutorial Course", IEEE Service Center, Piscataway.

32. Isermann R., "Digital Control Systems", Sringer - Verlog Berlin Heidelberg, New York, 1981.

33. Kalani G., "Microprocesor Based Distributed Control Systems", Prentice Hall, New York, 1988.

34. Katsuhiko O., "Ingeniería de Control Moderna", Prentice Hall Hispanoamericana, México, 1993.

35. Kern D., "Process Heat Transfer", International Student Edition, McGraw - Hill Book Company, New York, 1960.

36. Ledesma B., Conferencia: "Proyecto de Inteligenciación del edificio matriz del Banco del Pichincha y Diners Club", Escuela Politécnica Nacional - I2E, Qutio, 1995.

37. Lister A., Magee J., Sloman M., "Distributed Process Control Systems: Programming and Configurantion", Imperial College Research Report, No. 80/12, 1980.

38. Madron T., "Redes de Area Local", colección Megabyte, Grupo Noriega Editores, México, 1993.

39. Maloney T., "Electrónica Industrial. Dispositivos y Sistemas", Prentice Hall Hispanoamericana, México, 1983.

40. Mannesmann H., "The Process Contol System for the Process Industry - A system description", Contronic "P", Hartman & Braun.

41. Microsoft, "Visual BASIC 3.0 PRO. User's Manual", Microsoft Corporation, New York, 1993.

42. Microsoft, "Visual BASIC 3.0 PRO. Technical Reference", Microsoft Corporation, New York, 1993.

43. Microsoft, "Visual BASIC 3.0 PRO. Knowledge Base Articles from Microsoft Product Support", Microsoft Corporation, New York, 1993.

44. Microsoft, "Software Development Kit Reference", Microsoft Corporation, New

York, 1992.

45. Microsoft, "Aplication Programming Interface Reference", Microsoft Corporation, New York, 1992.

46. Miller Y., Freund J., Johnson R., "Probabilidad y Estadística par Ingenieros", cuarta edición, Prentice Hall Hispanoamericana, México, 1992.

47. Mohan N., Undeland T., Robbins W., "Power Electronics: Converters, Applications and Design", John Wiley and Sons, New York, 1989.

48. Molina J., Seminario: "Controladores Lógicos Programables", Escuela Politécnica Nacional, Centro de Educación Contínua, Quito, 1993.

49. Nebendahl D., "Sistemas Expertos - Introducción a la técnica y aplicación", Siemens - Editorial Marcombo, México, 1995.

50. Nebendahl D., "Sistemas Expertos - Experiencia de la práctica", Siemens - Editorial Marcombo, México, 1995.

51. Ortiz F., Tesis de Grado: "Optimización jerárquica de sistemas lineales con función de costo cuadrática", Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica, Quito, 1992.

52. Pappas C., Murray W., "Manual de Borland C++ 4.0", Osborne - McGraw - Hill, Madrid, 1994.

53. Pappas C., Murray W., "Programación en Windows", Osborne - McGraw - Hill, Madrid, 1993.

54. Porter A., "Programación en C++ para Windows", Osborne - McGraw - Hill, Madrid, 1994.

55. Quintana J., Banda H., Jornadas de Ingeniería Eléctrica y Electrónica: "Identificación de parámetros utilizando un algoritmo genético de optimización. Aplicación al circuito equivalente de una máquina de

inducción", Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica, Quito, 1994.

56. Roczinsky S., "Simulación por Computadora", colección Megabyte, Grupo Noriega Editores, México, 1993.

57. Santamaría G., Tesis de Grado: "Estudio del Sistema de Transmisión para el Sistema SCADA del Proyecto de Agua Potable: Mica - Quito Sur, de la EMAAP-Q", Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica, Quito, 1994.

58. Shcildt H., "Turbo C/C++. Manual de Referencia", Osborne - McGraw - Hill, Madrid, 1990.

59. Singh M., Titli A., "Decomposition, Optimization and Control", Pergamon Press, London, 1978.

60. Tokheim R., "Fundamento de los Microprocesadores", Colección SHAUM, segunda edición, McGraw - Hill Book Company, Mexico, 1992.

61. Vankelecom J., International Symposium: "Mini Systems for Control and Management of Distribution of Electric Loads in Developing Countries", Quito, 1990.

62. Warwick K., Rees D., "Computer Control of Industrial Processes", IEE Control Engineering Series 21 - Comprehensively revised edition, Peter Peregrinus Ltd., London, 1984.

63. Weitzman C., "Distributed Micro/Mini Computer Systems", Prentice Hall, New York, 1980.

ANEXOS

Anexo A. Manual de Usuario del paquete computacional SCADA System.

Anexo B. Archivos en Visual BASIC 3.0 del paquete computacional SCADA System.

Anexo C. Aplicación SCADA System.

ANEXO A. MANUAL DE USUARIO DEL PAQUETE COMPUTACIONAL SCADA SYSTEM.

A.1. Descripción general del programa.

SCADA es una aplicación técnica, de carácter exclusivamente didáctico, diseñada para la gobernación de pequeños prototipos industriales, ya sean éstos, modelos físicos (prototipos reales) o modelos computacionales (prototipos virtuales). SCADA es un programa que utiliza algoritmos de control digital directo y rutinas elementales de supervisión, con las cuales es posible monitorar el desenvolvimiento de un proceso y ejecutar sobre él, las más comunes acciones de control.

A.2. Requerimientos del sistema.

Para instalar y utilizar el programa SCADA correctamente se requiere:

- Procesador 80386 de 33 MHz.
- 2 MB de memoria RAM (4 MB si se pretende utilizar prototipos virtuales).
- 2 MB libres en disco duro (1 MB adicional para el proceso de instalación).
- Microsoft Windows 3.1 operando en modo *Enhanced*.
- Monitor VGA compatible con Microsoft Windows 3.1.
- Mouse compatible con Microsoft Windows 3.1.
- Tarjeta de adquisición de datos DAS-128 o compatibles. (No indispensable si se pretende solo utilizar prototipos virtuales).

A.3. Instalación del programa SCADA.

Antes de proceder con la instalación del programa SCADA asegúrese que el computador cumpla los requerimientos indicados en **A.2.** Si se desea operar con modelos físicos de plantas industriales lea cuidadosamente el *Manual de Usuario de la Tarjeta DAS-128*, en lo referente a Instalación. Revise y tome nota de la dirección base asignada a la tarjeta.

PRECAUCION: SI LA TARJETA DAS-128 TIENE UNA DIRECCIÓN QUE INTERFIERA CON OTRO DISPOSITIVO DEL COMPUTADOR, O SI SE UTILIZA EL PROGRAMA SCADA DIRECCIONANDO EQUIVOCADAMENTE A ESTA TARJETA, EL SISTEMA PUEDE VOLVERSE INESTABLE Y PRODUCIR DAÑOS SEVEROS EN EL COMPUTADOR.

Para instalar el programa SCADA:

- Arranque el computador y entre al sistema operativo Windows
- Introduzca el diskette de instalación No. 1 en la unidad de disco apropiada; por ejemplo: A:\
- Elija la opción de menú File(*Archivo*) / Run (*Correr*)
- Escriba A:\SETUP
- Siga las instrucciones indicadas en pantalla.

Para arrancar el programa haga un doble *click* en el ícono etiquetado con SCADA

System. Si la instalación fue exitosa en breves instantes aparecerá la ventana de presentación y el área de trabajo de la aplicación.

A.4. Programa SCADA System.

A.4.1. Administración de un Proyecto.

A.4.1.1. Iniciar un Proyecto. Para iniciar un nuevo proyecto SCADA elija la opción de menú File(*Archivo*) / New(*Nuevo*), o haga *click* en el botón apropiado de la barra de utilidades. Aparecerá una ventana titulada con System Definitions Summary (*Resumen de Definiciones del Sistema*). Esta ventana contiene una síntesis de todos las definiciones y parámetros importantes que caracterizan al proyecto.

A.4.1.2. Guardar la información de un proyecto. Si no ha guardado la información de su proyecto, elija la opción de menú File(*Archivo*) / Save As (*Guardar* Como). Aparecerá un cuadro de diálogo en el cual se debe especificar el nombre del archivo, con extensión .SCD, que contendrá la información del actual proyecto. Si ha hecho cambios importantes a su trabajo, puede actualizar el contenido del archivo de proyecto. Para esto elija la opción de menú File(*Archivo*) / Save(*Guardar*) , o haga *click* en el botón apropiado de la barra de utilidades.

A.4.1.3. Recuperar la información de un proyecto. Para abrir un archivo que contenga la información de algún proyecto anteriormente guardado, elija la opción de menú File(*Archivo*) / Open(*Abrir*), o haga *click* en el botón apropiado de la barra de utilidades. Aparecerá un cuadro de diálogo en el cual se debe especificar el nombre del

archivo, con extensión .SCD, que contiene la información del proyecto que desea recuperar.

A.4.1.4. Salir del programa SCADA. Para salir de la aplicación elija la opción de menú File(*Archivo*) / Exit(*Salir*). Si su trabajo no ha sido guardado o si ha sido modificado desde la última vez que lo guardó, aparecerá un cuadro de diálogo preguntándole si desea guardar la información del actual proyecto.

A.4.2. Configuración del sistema.

Cada proyecto SCADA debe ser configurado para operar bajo condiciones específicas de hardware. Para configurar o reconfigurar un proyecto SCADA elija la opción de menú File(*Archivo*) / Configure(*Configurar*). Aparecerá un cuadro de diálogo titulado System Configuration (Configuración del Sistema), en el cual se deben especificar los siguientes parámetros:

- Terminal Name (*Nombre del Terminal*): parámetro opcional que identifica la estación de trabajo con un nombre apropiado definido por el usuario.
- Serial Port (*Pórtico Serial*): indica cuál de los pórticos seriales del computador será utilizado por el programa SCADA para transmitir la información del sistema EN-LINEA. Por defecto este parámetro es Ninguno. Ver apartado **A.4.5.1.**
- First DAS-128 High Address (*Dirección Base de la primera tarjeta DAS-128*): especifica la dirección base para un dispositivo de adquisición de datos DAS-128. Por defecto el valor de esta dirección es 0200H.
- Second DAS-128 High Address (*Dirección base de la segunda tarjeta DAS-128*):

especifica la dirección base de un segundo dispositivo de adquisición de datos DAS-128 (si existiese). Por defecto el valor de este parámetro es Ninguno.

- Allow data registering on hard disk or diskette (*Permitir almacenamiento de información en disco o diskette*): determina si la información del sistema EN-LINEA puede o no ser almacenada en en disco o diskette, utilizando archivos de formato secuencial.
- Register path (*Ruta de registro*): indica la ruta de DOS bajo la cual se almacenará la información del sistema EN-LINEA, si la opción anterior así lo permite. Ver apartado **A.4.5.1.**
- Operate with (*Operar con*): permite determinar el tipo de prototipos a ser manipulados por el programa **SCADA**, pudiendo ser éstos: Real Prototypes (*Prototipos Reales*) o Virtual Prototypes (*Prototipos Virtuales*). La primera opción es la utilizada por defecto, en cuyo caso, es indispensable la presencia de la(s) tarjeta(s) DAS-128.

NOTA: Se deben especificar las direcciones base de los dispositivos de adquisición de datos aún cuando se trabaje solo con prototipos virtuales.

A.4.3. Definición de parámetros indispensables de un proyecto SCADA.

Una vez configurado el sistema se procede a la definición de Subsistemas de Control, Variables de Tiempo Real, Funciones de Control y Eventos Programados.

A.4.3.1. Definición de Subsistemas de Control. Para definir o redefinir los Subsistemas de Control pertenecientes al actual proyecto **SCADA** se elije la opción de menú

Define(*Definir*) / System Loops (*Subsistemas de Control*). Aparecerá un cuadro de diálogo titulado System Loops Definition (*Definición de Subsistemas de Control*), en el cual se deben especificar los siguientes parámetros:

- Name (*Nombre*): parámetro que identifica de manera única a cada Subsistema de Control perteneciente al actual proyecto **SCADA**. No existe restricción alguna para su representación, salvo que ésta no puede exceder los 64 caracteres.
- Description (*Descripción*): texto opcional que permite incluir comentarios específicos referentes a cada Subsistema de Control. No debe exceder los 256 caracteres.
- Sampling Period (*Período de Muestreo*): parámetro que determina la velocidad de adquisición y envío de datos relativa a cada Subsistema de Control. Este parámetro se especifica en décimas de segundo, y su valor no puede ser inferior a 1 (0.1 segundos) en subsistemas reales, o 10 (1 segundo) en el caso de subsistemas virtuales; y, en ningún caso dicho valor puede ser mayor que 6000. (10 minutos).
- Simulation Program (*Programa de Simulación*): especifica nombre del archivo ejecutable (con su ruta completa de DOS) que realizará la simulación en tiempo real del Subsistema definido. El programa de simulación referido debe ser compatible con **SCADA System,** y solo necesita ser especificado si se ha configurado al sistema para operar con prototipos virtuales. Ver apartados **A.4.2.** y **A.6.1-3.**
- Simulation Window (*Ventana de Simulación*): especifica el nombre de la ventana, perteneciente al programa de simulación, que contiene los pórticos virtuales para adquisición y envío de datos. Este nombre solo necesita ser especificado si se ha configurado al sistema para operar con prototipos virtuales. Ver apartados **A.4.2.** y **A.6.1-3.**

- Timing Variable (*Variable de Temporización*): especifica el nombre de una variable, perteneciente al programa de simulación, que es utilizada como temporizador de la simulación. Este nombre solo necesita ser especificado si se ha configurado al sistema para operar con prototipos virtuales. Ver apartados **A.4.2.** y **A.6.1-3.**

Para añadir nuevos Subsistemas de Control, se especifican primero todas sus características y se presiona el botón Add (*Añadir*). El nuevo Subsistema de Control formará parte de una lista de definiciones que aparece en el presente cuadro de diálogo.

Para cambiar las características de un Subsistema de Control definido anteriormente, se selecciona de la lista de definiciones del presente cuadro de diálogo, el nombre del Subsistema de interés. Se desplegará entonces toda la información correspondiente a dicho Subsistema, a partir de la cual, se pueden realizar las correcciones necesarias. Para efectivizar los cambios presione el botón Change (*Cambiar*).

Para eliminar un Subsistema de Control definido anteriormente, se selecciona de la lista de definiciones del presente cuadro de diálogo, el nombre del Subsistema de interés, y luego se presiona el botón Delete (*Eliminar*).

Para validar las definiciones y regresar a la ventana principal de **SCADA System** presione el botón OK (*Aceptar*). Para regresar a la ventana principal de **SCADA System** sin validar las definiciones realizadas, presione el botón Cancel (*Cancelar*).

A.4.3.2. Definición de Variables de Entrada y Salida en Tiempo Real. Para definir o

redefinir las Variables de Entrada y Salida asociadas a un Subsistema de Control del actual proyecto SCADA se elije la opción de menú Define(*Definir*) / Real Time I/O Variables (*Variables de E/S de Tiempo Real*). Aparecerá un cuadro de diálogo titulado Real Time I/O Variables Definition (*Definición de Variables de E/S de Tiempo Real*), en el cual se deben especificar los siguientes parámetros:

- System Loop (*Subsistema*): nombre del Subsistema de Control al cual está subordinada la variable definida.
- Name (*Nombre*): parámetro que identifica de manera única a cada Variable de Tiempo Real. Se pueden definir dos o más variables con el mismo nombre siempre que pertenezcan a Subsistemas diferentes. No existe restricción alguna para su representación, salvo que ésta no puede exceder los 64 caracteres.
- Description (*Descripción*): texto opcional que permite incluir comentarios específicos referentes a cada Variable de Tiempo Real. No debe exceder los 256 caracteres.
- Used as (*Utilizada como*): parámetro que indica la naturaleza de la variable definida, pudiendo ser esta: Entrada Analógica/Digital de 8 bits, Entrada Digital de 8 bits, Entrada Digital de 1 Bit, Salida Digital/Analógica de 8 bits, Salida Digital de 8 bits, o Salida Digital de 1 Bit.
- High Address (*Dirección Base*): parámetro que especifica la dirección en 16 bits del pórtico de E/S utilizado por variable.
- Low Address (*Dirección Específica*): parámetro que especifica la dirección en 8 bits del pórtico de E/S utilizado por la variable. El resultado de la suma de este parámetro con el anterior define completamente la dirección del pórtico de E/S utilizado por la variable. Por ejemplo: si High Address = 0300H y Low Address = 04H, la dirección resultante será 0304H.

- Simulation Variable (*Variable de Simulación*): especifica el nombre de la variable, perteneciente al programa de simulación, que es utilizada como pórtico virtual para la adquisición o envío de datos. Este nombre solo necesita ser especificado si se ha configurado al sistema para operar con prototipos virtuales. Ver apartados **A.4.2.** y **A.6.1-3.**

Si se presiona el botón Define (*Definir*) aparecerá un nuevo cuadro de diálogo, en el cual se pueden especificar los siguientes parámetros:

Para variables de entrada (de cualquier tipo):

- Use Scaling Factor (*Usar Factor de Escalamiento*): permite habilitar o deshabilitar el escalamiento de 8 bits a un rango definido por el usuario, de acuerdo con los valores especificados en Upper Limit (*Límite Superior*), Lower Limit (*Límite Inferior*) y Units (*Unidades*).
- Generate Set Point and Error Variables (*Generar Variables de Referencia y Error*): permite habilitar o deshabilitar la generación de una señal de referencia y su consiguiente señal de error relativas a la variable definida. Si se habilita esta opción, se debe especificar el origen de la señal de referencia, pudiendo ser éste: Manual Adjustment (*Ajuste Manual*) o Programmed Adjustment (*Ajuste Programado*). En el primer caso, el valor de la señal de referencia podrá ser calibrado cuando el sistema esté En-Línea a través del Panel de Control; en el segundo caso, el valor de la señal de referencia podrá ser ajustado cuando el sistema esté Fuera-de-Línea a través de una Programación de Eventos. Ver apartado **A.4.3.4.**

Para variables de salida (de cualquier tipo):

- Use Scaling Factor (*Usar Factor de Escalamiento*): permite habilitar o deshabilitar el escalamiento de 8 bits a un rango definido por el usuario, de acuerdo con los valores especificados en Upper Limit (*Límite Superior*), Lower Limit (*Límite Inferior*) y Units (*Unidades*).
- Output Variable Source (*Fuente de la Variable de Salida*): permite especificar la procedencia de la señal de salida, pudiendo ser ésta: Control Function (*Función de Control*), Programmed Entry (*Línea de Programa*), o Manual Regulation (*Regulación Manual*). En el primer caso, el valor de la señal de salida podrá ser ajustado automáticamente cuando el sistema esté En-Línea, de acuerdo a los resultados de una ley de control específica; en el segundo caso, el valor de la señal de salida podrá ser ajustado cuando el sistema esté Fuera-de-Línea a través de una Programación de Eventos; en el tercer caso, el valor de la señal de salida podra ser ajustado cuando el sistema esté En-Línea a través del Panel de Control. Ver apartado **A.4.3.4**.

Para añadir nuevas Variables de Tiempo Real, se especifican primero todas sus características y se presiona el botón Add (*Añadir*). La nueva variable formará parte de una lista de definiciones que aparece en el presente cuadro de diálogo.

Para cambiar las características de una Variable de Tiempo Real definida anteriormente, se selecciona de la lista de definiciones del presente cuadro de diálogo, el nombre de la variable de interés. Se desplegará entonces toda la información correspondiente a dicha variable, a partir de la cual, se pueden realizar las correcciones necesarias. Para efectivizar los cambios presione el botón Change (*Cambiar*).

Para eliminar una Variable de Tiempo Real definida anteriormente, se selecciona de la lista de definiciones del presente cuadro de diálogo, el nombre de la variable de interés, y luego se presiona el botón Delete (*Eliminar*).

Para validar las definiciones y regresar a la ventana principal de SCADA System presione el botón OK (*Aceptar*). Para regresar a la ventana principal de SCADA System sin validar las definiciones realizadas, presione el botón Cancel (*Cancelar*).

A.4.3.3. Definición de Funciones de Control. Para definir o redefinir las Funciones de Control asociadas a un Subsistema de Control del actual proyecto SCADA se elije la opción de menú Define(*Definir*) / Control Functions (*Funciones de Control*). Aparecerá un cuadro de diálogo titulado Control Functions Definition (*Definición de Funciones de Control*), en el cual se deben especificar los siguientes parámetros:

- System Loop (*Subsistema*): nombre del Subsistema de Control al cual está subordinada la ley de control definida.
- Name (*Nombre*): parámetro que identifica de manera única a cada Función de Control. Se pueden definir dos o más funciones con el mismo nombre siempre que pertenezcan a Subsistemas diferentes. No existe restricción alguna para su representación, salvo que ésta no puede exceder los 64 caracteres.
- Description (*Descripción*): texto opcional que permite incluir comentarios específicos referentes a cada Función de Control. No debe exceder los 256 caracteres.
- Applied to (*Aplicada sobre*): especifica la variable de entrada a cuyo error se aplicará la Función de Contol definida. Las Funciones de Control solo pueden definirse para aquellas variables de entrada que tengan habilitada la opción Generate Set

Point and Error Variables (*Generar Variables de Referencia y Error*). Ver apartado A.4.3.3.

- Send to (*Enviar a*): especifica la variable de salida que recibirá el resultado de la Función de Control. Las Funciones de Control solo pueden definirse para aquellas variables de salida que tengan el atributo Output Variable Source (*Fuente de la Variable de Salida*) ajustado a la opción Control Function (*Función de Control*).
- Control Type (*Tipo de Control*): permite escojer la ley de control que relacionará las variables de entrada y salida estipuladas anteriormente. Las Funciones de Control pueden ser de uno de los siguientes tipos:

  - PID discreto, en cuyo caso se deberán introducir las constantes proporcional, integral y derivativa de un modelo contínuo.
  - Red discreta de adelanto o retraso de fase, en cuyo caso se deberá introducir un cero y un polo en el plano S, y un valor de ganancia en lazo abierto.
  - Algoritmo genérico de tercer orden, en cuyo caso se deberán introducir los coeficientes de una ecuación de diferencias.
  - Control ON/OFF simple, en cuyo caso se deberá introducir un valor de umbral y una orientación (creciente o decreciente) de la no-linealidad.
  - Control ON/OFF con histéresis, en cuyo caso se deberán introducir los valores de umbral superior e inferior, y una orientación (creciente o decreciente) de la no-linealidad.

La variable de entrada, de salida, o ambas, especificadas en una Función de Control,

pueden intervenir en otras definiciones. No existe verdadero conflicto en esto, puesto que los algoritmos de control definidos no se activan automáticamente cuando el sistema entra En- Línea, sino que requieren de una orden explícita de SCADA System. Ver apartado A.4.3.4.

Para añadir nuevas Funciones de Control, se especifican primero todas sus características y se presiona el botón Add (*Añadir*). La nueva función formará parte de una lista de definiciones que aparece en el presente cuadro de diálogo.

Para cambiar las características de una Función de Control definida anteriormente, se selecciona de la lista de definiciones del presente cuadro de diálogo, el nombre de la función de interés. Se desplegará entonces toda la información correspondiente a dicha función, a partir de la cual, se pueden realizar las correcciones necesarias. Para efectivizar los cambios presione el botón Change (*Cambiar*).

Para eliminar una Función de Control definida anteriormente, se selecciona de la lista de definiciones del presente cuadro de diálogo, el nombre de la función de interés, y luego se presiona el botón Delete (*Eliminar*).

Para validar las definiciones y regresar a la ventana principal de SCADA System presione el botón OK (*Aceptar*). Para regresar a la ventana principal de SCADA System sin validar las definiciones realizadas, presione el botón Cancel (*Cancelar*).

A.4.3.4. Programación de Eventos. SCADA permite al usuario editar un conjunto de proposiciones, a manera de programación lineal, que permiten a cada Subsistema ejecutar

ordenadamente una secuencia de operaciones. Esta programación se realiza por medio de sentencias condicionales del tipo *SI (condición) ... ENTONCES (consecuencia).*

Para programar o reprogramar la secuencia de operaciones relacionadas con un Subsistema de Control del actual proyecto **SCADA** se elije la opción de menú Define(*Definir*) / Programmed Events (*Programación de Eventos*). Aparecerá un cuadro de diálogo titulado Programmed Events Definition (*Definición de Eventos Programados*), en el cual se deben especificar los siguientes parámetros:

- System Loop (*Subsistema*): nombre del Subsistema de Control al cual va dirigida la programación lineal de eventos.
- Evaluated expression (*Expresión a evaluarse*): especifica la condición que debe ser evaluada por **SCADA System** previa a la ejecución de una orden. Esta puede ser:

    - Tiempo de Proceso igual a un valor definido por el usuario.
    - Tiempo de Proceso mayor que un valor definido por el usuario.
    - Tiempo de Proceso menor que un valor definido por el usuario.
    - Tiempo de Proceso comprendido entre dos valores definidos por el usuario.
    - Variable de Proceso igual a un valor definido por el usuario.
    - Variable de Proceso mayor que un valor definido por el usuario.
    - Variable de Proceso menor que un valor definido por el usuario.
    - Variable de Proceso comprendida entre dos valores definidos por el usuario.

NOTA: *Tiempo de Proceso* se refiere al número de *décimas de segundos* transcurridos desde que el Subsistema ingresó a la línea de control. *Variable de Proceso* es cualquier señal de entrada, salida o error, producida por el Subsistema.

- Applied to (*Aplicada sobre*): especifica la variable de entrada, salida o error, perteneciente al Subsistema de Control señalado, que es objeto de la evaluación anterior. Es requerida solo por las cuatro últimas condiciones.

NOTA: Variables definidas con magnitud de 1 Bit solo pueden ser evaluadas con la quinta condición.

- Parameter 1 (*Parámetro 1*) y Parameter 2 (*Parámetro 2*): especifican los valores definidos por el usuario que intervienen en la evaluación anterior.
- Action executed (*Acción ejecutada*): especifica la orden que ha de producirse cuando se haya cumplido la respectiva condición. Esta puede ser:

    - Fijar una variable de Set Point a un valor definido por el usuario.
    - Fijar una variable de Salida a un valor definido por el usuario.
    - Habilitar un algoritmo de control definido con anterioridad.
    - Deshabilitar un algoritmo de control definido con anterioridad.

- Applied to (*Aplicada sobre*): especifica la variable de salida, set point, o función de control, perteneciente al Subsistema de Control señalado, que es objeto de la acción anterior.
- Parámetro (Parámeter): especifica el valor definido por el usuario que interviene en la

orden anterior.

Para añadir nuevas sentencias a una Programación de Eventos, se especifican primero todas sus características y se presiona el botón Add (*Añadir*). Las sentencias definidas se listan en un recuadro localizado en la parte inferior del cuadro de diálogo. El orden en el que están listadas representa el *orden según su prioridad:* aquellas más importantes se encuentran al principio de la lista, las menos importantes al final. Una sentencia definida en algún punto de la lista tiene prioridad respecto a todas las sentencias listadas abajo de ella, pero no prioridad respecto a las sentencias listadas arriba de ella.

Conforme se definen, las sentencias se van añadiendo al final de la lista. Si se reqiere dar una prioridad específica a una sentencia formulada, primero debe señalarse el punto de la lista antes del cual se desee colocar la sentencia; se edita entonces la sentencia conforme a lo indicado anteriormente; y luego, se presiona el botón Add (*Añadir*).

Para cambiar las características de una sentencia definida anteriormente en una Programación de Eventos, se selecciona de la lista respectiva, la sentencia de interés. Se desplegará entonces toda la información correspondiente a dicha sentencia, a partir de la cual, se pueden realizar las correcciones necesarias. Para efectivizar los cambios presione el botón Change (*Cambiar*).

Para eliminar una sentencia definida anteriormente en una Programación de Eventos, se selecciona de la lista respectiva, la sentencia de interés; luego se presiona el botón Delete (*Eliminar*).

Una vez definidos los parámetros indispensables para la operación de **SCADA System** se procede a la definición de Alarmas, Funciones Estadísticas y Medios de Registro, válidos solo para variables de entrada o salida con resolución de 8 bits.

A.4.4.1. Definición de Alarmas. Para definir o redefinir las Alarmas asociadas a una Variable de Tiempo Real del actual proyecto **SCADA** se elije la opción de menú Options(*Opciones*) / Alarms (*Alarmas*). Aparecerá un cuadro de diálogo titulado Alarms

- Variable Name (*Variable*): nombre de la Variable de Tiempo Real cuyos niveles de alarma se quiere definir.
- Upper Emergency Limit (*Nivel Superior de Emergencia*): especifica el límite máximo, cuyo exceso producirá una alarma de EMERGENCIA.
- Upper Warning Limit (*Nivel Superior de Alerta*): especifica el límite máximo, cuyo exceso producirá una alarma de ALERTA.
- Lower Warning Limit (*Nivel Inferior de Alerta*): especifica el límite mínimo, cuyo exceso producirá una alarma de ALERTA.
- Lower Emergency Limit (*Nivel Inferior de Emergencia*): especifica el límite mínimo, cuyo exceso producirá una alarma de EMERGENCIA.

Ninguno de estos parámetros es indispensable, pero de especificarse, deben estar dentro del rango (con las debidas unidades) definido para la Variable de Tiempo Real en cuestión. Ver apartado **A.4.3.2.** Tampoco es lícito sobreponer entre sí sus dominios de operación, es decir, p. ej: no es válido definir Upper Emergency Limit = 300 °C y Lower Warning Limit = 360 °C, porque se sobreentiende que Upper Emergency Limit tiene un valor numéricamente superior a Lower Warning Limit.

Para añadir o cambiar los niveles de alarma de una Variable de Tiempo Real, edite primero los valores apropiados, y presione el botón Change (*Cambiar*) del presente cuadro de diálogo.

Para validar las definiciones y regresar a la ventana principal de **SCADA System** presione el botón OK (*Aceptar*). Para regresar a la ventana principal de **SCADA System** sin validar las definiciones realizadas, presione el botón Cancel (*Cancelar*).

A.4.4.2. **Definición de Funciones Estadísticas.** Para definir o redefinir las Funciones Estadistícas que han de evaluadarse sobre una Variable de Tiempo Real del actual proyecto SCADA se elije la opción de menú Options(*Opciones*) / Statistics (*Funciones Estadísticas*). Aparecerá un cuadro de diálogo titulado Statistical Functions Definition (*Definición de Funciones Estadísticas*), en el cual se deben especificar los siguientes parámetros:

- System Loop (*Subsistema*): nombre del Subsistema de Control al cual está subordinada la variable cuyas opciones se quiere especificar.
- Variable Name (*Variable*): nombre de la Variable de Tiempo Real cuyas opciones se quiere especificar.
- Maximum value (*Valor máximo*): devuelve el valor máximo de la variable evaluado con las últimas 200 muestras.
- Minimum value (*Valor mínimo*): devuelve el valor mínimo de la variable evaluado con las últimas 200 muestras.
- Average value (*Valor medio*): devuelve la media aritmética de la variable evaluada con las ultimas 200 muestras.
- Standard Deviation (*Desviación Estándar*): devuelve la raiz cuadrada de la variancia de la variable evaluada con las últimas 200 muestras.
- Frequency Count (*Conteo de Frecuencia*): devuelve un gráfico que representa la distribución de frecuencia de la variable considerando las últimas 200 muestras.

Para añadir o eliminar Funciones Estadísticas asociadas una Variable de Tiempo Real, señale primero las opciones deseadas, y presione el botón Change (*Cambiar*) del presente cuadro de diálogo.

Para validar las definiciones y regresar a la ventana principal de SCADA System presione el botón OK (*Aceptar*). Para regresar a la ventana principal de SCADA System sin validar las definiciones realizadas, presione el botón Cancel (*Cancelar*).

A.4.4.3. Definición de Medios de Registro. Para establecer los Medios de Registro para una Variable de Tiempo Real del actual proyecto SCADA se elije la opción de menú Options(*Opciones*) / Registers (*Medios de Registro*). Aparecerá un cuadro de diálogo titulado Registers Definition (*Definición de Medios de Registro*), en el cual se deben especificar los siguientes parámetros:

- System Loop (*Subsistema*): nombre del Subsistema de Control al cual está subordinada la variable cuyas opciones se quiere especificar.
- Variable Name (*Variable*): nombre de la Variable de Tiempo Real cuyas opciones se quiere especificar.
- Send to Window (*Enviar a ventana gráfica*): permite visualizar el comportamiento en tiempo real de la variable en cuestión, a través de una ventana gráfica. Esta opción es siempre escogida por defecto.
- Send to Disk (*Enviar a disco*): permite registrar la información del Subsistema en un archivo de formato serial. Los datos registrados no conservan la característica de tiempo real, puesto que son enviados asincronicamente en lotes de 16K. Por convención, todos los datos del Subsistema válidos para un instante dado están separados por el caracter ASCII número 9 (TAB), mientras que cada conjunto de datos, perteneciente a un instante de tiempo, está separado del siguiente por el caracter ACII número 13 (CR). Ver apartado **A.4.2.** La información contenida en este archivo puede ser importada por otros programas como Microsoft EXCEL

para su análisis Fuera-de-Línea.

- Send to Serial (*Enviar al pórtico serial*): permite enviar la información al pórtico serial del computador, conservando el mismo formato que el caso anterior. Ver apartado **A.4.2.** No existen aplicaciones especificas que puedan recibir y analizar la información, éstas deben ser desarrolladas por el usuario. El protocolo utilizado es UART de 8 bits de información, sin bits de paridad, y un bit de parada; la velocidad de transmisión es 9600 bps.

Para especificar los Medios de Registro apropiados para una Variable de Tiempo Real, señale primero las opciones deseadas, y presione el botón Change (*Cambiar*) del presente cuadro de diálogo.

Para validar las definiciones y regresar a la ventana principal de **SCADA System** presione el botón OK (*Aceptar*). Para regresar a la ventana principal de **SCADA System** sin validar las definiciones realizadas, presione el botón Cancel (*Cancelar*).

A.4.5. Operación EN-LINEA.

Pueden ingresar a la línea de control algunos o todos los Subsistemas de Control definidos en un proyecto **SCADA**. Para hacer esto, se elige la opción de menú Command(*Comando*) / Start(*Arrancar*), o la opción Command(*Comando*) / Start All (*Arrancar Todos*). En el primer caso, aparecerá un cuadro de diálogo titulado System Loops Available (*Subsistemas Disponibles*), en el cual se deben seleccionar los Subsistemas que ingresarán a la línea de control. En el segundo caso, todos los Subsistemas definidos en el actual proyecto

SCADA entrarán automáticamente a la línea de control. Durante la operación En-Línea, no es posible editar ningún parámetro del proyecto SCADA.

NOTA: SCADA System tomará algunos segundos antes de entrar a la línea de control para verificar rutinas de comunicación, control y supervisión. Si en este punto se presentan mensajes de error, automáticamente todos los Subsistemas abandonarán la línea de control, y el programa retornará al modo de edición.

Pueden abandonar la línea de control algunos o todos los Subsistemas de Control que se hallen operando En-Línea. Para hacer esto, se elige la opción de menú Command(*Comando*) / Halt(*Detener*), o la opción Command(*Comando*) / Halt All (*Detener Todos*). En el primer caso, aparecerá un cuadro de diálogo titulado System Loops Available (*Subsistemas Disponibles*), en el cual se deben seleccionar los Subsistemas que abandonarán la línea de control. En el segundo caso, todos los Subsistemas definidos en el actual proyecto SCADA abandonarán automáticamente la línea de control. Con ninguna de estas opciones el programa retorna al modo de edición.

Para retornar al modo de edición se elige la opción de menú Command(*Comando*) / Reset(*Restaurar todo*).

IMPORTANTE: Es altamente recomendable guardar la información del proyecto SCADA antes de ingresar a la línea de control.

Al iniciar la operación En-Línea aparecerán ventanas de texto con la información en tiempo real de cada uno de los Subsistemas En-Línea. Cada ventana contiene una lista de las variables asociadas al Subsistema en cuestión, con su respectiva dirección en memoria,

su valor instantáneo y sus unidades.

En cualquier momento de la operación En-Línea se puede acceder a información especifica de cualquiera de las variables listadas. Para hacer esto, haga *click* en el nombre de la variable de interés, y escoja una de las siguientes opciones:

- Opción de menú View(*Visualizar*) / Real Time Monitoring (*Comportamiento en Tiempo Real*), para visualizar en una ventana gráfica la magnitud de la variable en función del tiempo. Esta alternativa es válida solo para variables de resolución de 8 bits que tienen habilitada la opción Send to Window (*Visualizar en Ventana Gráfica*). Ver apartado **A.4.4.3.**
- Opción de menú View(*Visualizar*) / Statistical Information (*Información Estadística*), para visualizar en una ventana gráfica los resultados estadísticos de la variable considerando las últimas 200 muestras. Esta alternativa es válida solo para variables de resolución de 8 bits que permitan hacer evaluaciones estadisticas . Ver apartado **A.4.4.2.**
- Opción de menú View(*Visualizar*) / Alarms Register (*Registro de Alarmas*), para visualizar en una ventana la lista de las últimas 10 alarmas relacionadas con la variable. Esta alternativa es válida solo para variables de resolución de 8 bits cuyos niveles de alarma hayan sido explícitamente definidos. Ver apartado **A.4.4.1.** Las alarmas producidas se restablecen manualmente ingresando a esta ventana y presionando el respectivo botón de Reset (*Restablecer*).
- Opción de menú View(*Visualizar*) / Control Panel (*Panel de Control*), que emula un panel de control y permite operar manualmente sobre la variable. Esta alternativa es válida solo para variables de entrada que generen señal de referencia de ajuste

manual, o para variables de salida de regulación manual. Ver apartado A.4.3.2.

A.5. Limitaciones del programa SCADA System.

El programa **SCADA System** tiene ciertos límites en cuanto a sus capacidades:

- No se puede trabajar con prototipos reales y prototipos virtuales simultáneamente.
- No se pueden definir más de 16 subsistemas de operación simultánea.
- Ningún subsistema puede tener período de muestreo inferior a 1 segundo.
- No se pueden utilizar más de 2 tarjetas de adquisición de datos DAS-128, por tanto, no pueden definirse más de:
    - 16 entradas analógico/digitales con resolución de 8 bits.
    - 16 salidas digital/analógicas con resolución de 8 bits.
    - 14 entradas digitales con resolución de 8 bits.
    - 14 salidas digitales con resolución de 8 bits.
    - 16 entradas digitales con resolución de 1 bit.
    - 16 salidas digitales con resolución de 1 bit.
- No se pueden definir más de 256 funciones de control diferentes.
- No se pueden definir más de 1024 líneas en una programación de eventos.

A.6. Prototipos virtuales.

A.6.1. Comunicación. Para que una planta (real o virtual) pueda operar bajo el comando

de SCADA System es indispensable la presencia de un medio de comunicación bidireccional que enlace la planta con el sistema de control/supervisión. Para prototipos reales dicho medio es la tarjeta de adquisición y salida de datos DAS-128, mientras que para prototipos virtuales, dada su naturaleza, se utiliza el Intercambio Dinámico de Datos (Dynamic Data Exchange o DDE).

A.6.2. Intercambio dinámico de datos. Es un mecanismo de Windows por medio del cual dos aplicaciones pueden establecer una *conversasión* e intercambiar información contínua y automáticamente. Quien inicia la conversación se conoce como aplicación *destino*, mientras que la aplicación que responde se conoce como *fuente*. SCADA System siempre será la aplicación destino, y el programa de simulación desarrollado, la aplicacíon fuente.

A.6.3. Normas para el desarrollo de prototipos virtuales. Para que un programa de simulación pueda ser utilizado por SCADA System, en calidad de prototipo virtual, debe sujetarse a las siguientes restricciones:

- Ser desarrollado en un lenguaje que soporte la programación en Microsoft Windows. Se recomiendan las plataformas integradas de desarrollo de software Microsoft Visual BASIC 3.0, Microsoft Visual C++ 1.0, Borland C++ 4.0, o versiones mejoradas de los mismos.
- Acondicionar al programa para aceptar conversasiones DDE.
- Definir al programa como aplicación fuente.- Declarar a las variables objeto involucradas en la conversasión DDE como tipo *Label* (Etiquetas) o *TextBox* (Cuadros de Texto). Estas variables *simularán* los puertos de entrada y salida de la

tarjeta de adquisición de datos DAS-128.

- Agrupar todas las variables objeto involucradas en la conversasión DDE bajo *una sola* ventana.
- Incluir una variable objeto del tipo *Label* que pueda ser utilizada por **SCADA System** como temporizador de la aplicación. Esta variable NO DEBE ser alterada por la misma aplicación fuente.

IMPORTANTE: En cada instante de muestreo el programa **SCADA System** aumentará en uno el contenido de esta variable. Ante la sucitación de este evento, el programa de simulación deberá realizar los cálculos necesarios para calcular el nuevo estado de la planta.

- Contener el objeto *Control Box* (Cuadro de Control), e incluir la opción *Close* (Cerrar), ya que **SCADA System** recurre a este elemento para cerrar la aplicación cuando la planta esté fuera de línea.

En los demás aspectos, los programas de simulación tienen completa libertad. No existen límites al número de ventanas, objetos, o variables en general, sino los que de por sí restrinja el lenguaje de programación o el entorno Windows.

Para mayor información, referirse a los Manuales Técnicos o a los Archivos de Ayuda en Línea de los respectivos lenguajes de programación.

A. 7. Sesión de trabajo con el programa SCADA System.

En este apartado se explica mediante un ejemplo, el desarrollo de una sesión de trabajo con el programa **SCADA System**. El ejemplo utilizado hace uso de un prototipo virtual, es decir, un programa de simulación para emular el comportamiento de una planta real. Ver apartados **A.6.1-3.** El diseño, implementación y utilización de dicho prototipo está detallado en el apartado **A.8.1.**

- Entre al programa **SCADA System** y elija la opción de menú File(*Archivo*)/New(*Nuevo*).
- Elija la opción de menú File(*Archivo*)/Configure(*Configurar*). Aparecerá el cuadro de diálogo System Configuration (*Configuración del Sistema*). Defina sus propiedades de acuerdo a lo siguiente:

Propiedad	Especificación
Nombre del Terminal	Unidad Local de Procesamiento
Pórtico serial:	Ninguno
Dirección base de la primera tarjeta DAS-128	0200H (o la que corresponda)
Dirección base de la segunda tarjeta DAS-128	Ninguna
Permitir almacenamiento en disco	Si
Ruta de registro	c:\scada\register
Operar con	Prototipos Virtuales

Presione el botón OK para efectivizar los cambios y volver a la ventana principal.

- Elija la opción de menú Define(Definir)/System Loops (Subsistemas de Control). Aparecerá el cuadro de diálogo System Loops Definition (Definición de Subsistemas de Control). Defina sus propiedades de acuerdo a lo siguiente:

Parámetro	Especificación
Nombre del Subsistema	Control de Temperatura
Descripción	Ensayo con SCADA System
Período de muestreo	20 (décimas de segundo)
Programa de simulación	c:\scada\simfiles\temperat.exe
Venata de simulación	frmSistema
Variable de temporización	iTiempoDiscreto

Presione el botón Add para añadir el actual lazo de control a la lista de subsistemas subordinados al terminal "Unidad Local de Procesamiento". Presione el botón OK para efectivizar los cambios y volver a la ventana principal.

- Elija la opción de menú Define(Definir)/Real Time Variables (Variables de Tiempo Real). Aparecerá el cuadro de diálogo Real Time Variables Definition (*Definición de Variables de Tiempo Real*). Defina sus propiedades de acuerdo a lo siguiente:

Parámetro	Especificación
Subsistema	Control de Temperatura
Nombre de la variable	Temperatura
Descripción	Variable de entrada al sistema SCADA
Usada como	Entrada A/D de 8 bits
Dirección base	0200H (o la que corresponda)
Dirección específica	00H
Variable de simulación	fTemperatura

Presione el botón Define para establecer características adicionales sobre la actual variable. Aparecerá el cuadro de diálogo Advanced Input Variable Definition (*Definición Avanzada de Variables de Entrada*). Defina sus propiedades de acuerdo a lo siguiente:

Parámetro	Especificación
Usar factor de escalamiento	Si
Límite superior	100
Límite inferior	0
Unidades	°C
Generar variables de Set Point y Error	Si
Señal de Set Point	Ajuste Manual

Presione el botón OK para efectivizar los cambios y volver al cuadro de diálogo anterior. En dicho cuadro presione el botón Add para añadir la actual variable a la lista de variables de tiempo real asociadas al subsistema "Control de Temperatura".

Defina una segunda variable con los siguientes parámetros:

Parámetro	Especificación
Subsistema	Control de Temperatura
Nombre de la variable	Señal de Control
Descripción	Salida del sistema SCADA
Usada como	Salida D/A de 8 bits
Dirección base	0200H (o la que corresponda)
Dirección específica	00H
Variable de simulación	fControl

Presione el botón Define para establecer características adicionales sobre la actual variable. Aparecerá el cuadro de diálogo Advanced Output Variable Definition (*Definición Avanzada de Variables de Salida*). Defina sus propiedades de acuerdo a lo siguiente:

Parámetro	Especificación
Usar factor de escalamiento	Si
Límite superior	10
Límite inferior	0
Unidades	Voltios
Origen de la señal de salida	Función de Control

Presione el botón OK para efectivizar los cambios y volver al cuadro de diálogo anterior. En dicho cuadro presione el botón Add para añadir la actual variable a la lista de variables de tiempo real asociadas al subsistema "Control de Temperatura". Presione finalmente el botón OK para efectivizar los cambios y volver a la ventana principal.

- Elija la opción de menú Define(*Definir*)/Control Functions (*Funciones de Control*). Aparecerá el cuadro de diálogo Control Functions *Definition (Definición de Funciones de Control*). Defina sus propiedades de acuerdo a lo siguiente:

Parémetro	Especificación
Subsistema	Control de Temperatura
Nombre de la Función	Realimentación Unitaria
Descripción	Lazo cerrado simple
Aplicada a	Error de Temperatura
Enviada a	Señal de Control
Tipo de Control	PID
Constante proporcional	1
Constante integral	0
Constante derivativa	0

Presione el botón Add para añadir la actual función a la lista de contoladores pertenecientes al subsistema "Control de Temperatura". Presione el botón OK para

efectivizar los cambios y volver a la ventana principal.

- Elija la opción de menú Define(*Definir*)/Programmed Events (*Programación de Eventos*). Aparecerá el cuadro de diálogo Programmed Events *Definition (Definición de Eventos programados*). Defina sus propiedades de acuerdo a lo siguiente:

Parémetro	Especificación
Subsistema	Control de Temperatura
Expresión evaluada	Tiempo de Proceso = Valor
Parámetro 1	20 (décimas de segundo)
Acción ejecutada	Habilitar función de control
Aplicada a	Realimentación Unitaria

Presione el botón Add para añadir la actual sentencia a la lista de eventos pertenecientes al subsistema "Control de Temperatura". Presione el botón OK para efectivizar los cambios y volver a la ventana principal.

- Elija la opción de menú Options(Opciones)/Alarms (Alarmas). Aparecerá el cuadro de diálogo Alarms Definition (Definición de Alarmas). Defina sus propiedades de acuerdo a lo siguiente:

Parámetro	Especificación
Subsistema	Control de Temperatura
Variable	Temperatura
Nivel superior de emergencia	90
Nivel superior de alerta	75

Presione el botón Change para fijar los cambios en la actual variable. Presione el botón OK para efectivizar los cambios y retornar a la ventana principal.

- Elija la opción de menú Options(Opciones)/Register (Medios Registros). Aparecerá el cuadro de diálogo Registers Definition (Definición de Medios de Registro). Defina sus propiedades de acuerdo a lo siguiente:

Parámetro	Especificación
Subsistema	Control de Temperatura
Variable	Temperatura
Enviar a ventana gráfica	Si
Enviar a disco	Si

Presione el botón Change para fijar los cambios en la actual variable. Defina los siguientes parámetros para una segunda variable.

Parámetro	Especificación
Subsistema	Control de Temperatura
Variable	Señal de Control
Enviar a ventana gráfica	Si
Enviar a disco	Si

Presione el botón Change para fijar los cambios en la actual variable. Presione el botón OK para efectivizar los cambios y retornar a la ventana principal.

- Elija la opción de menú File(*Archivo*)/Save As (*Guardar Como*). Aparecerá un cuadro de

diálogo en el cual se debe especificar el nombre del archivo que contendrá la actual sesión de trabajo. Especifique un nombre como: c:\scada\samples\temperat.scd

- Elija la opción de menú Command(*Comando*)/Start All (*Arrancar Todos*) para ingresar a la línea de control. Aparecerá un cuadro de diálogo en el que se debe especificar el nombre del archivo en el cual se registrará la información de tiempo real. El archivo aparecerá en el directorio c:\scada\register (especificado anteriormente). Especifique un nombre como: temperat.txt . En breves instantes aparecerá una ventana con la información del subsistema "Control de Temperatura".

- De la lista de variables de tiempo real selecciónе "Temperatura". Elija ahora la opción de menú View(*Visualizar*)/Control Panel (*Panel de Control*). Aparecerá una ventana que simula un panel de control para la variable "Temperatura". Fije el nivel de referencia de esta variable al valor de 100 °C. Presione OK para salir.

- Elija la opción de menú View(*Visualizar*)/ Real Time *Monitoring* (*Monitoreo en Tiempo Real*) para visualizar el comportamiento de la variable "Temperatura".

- Realice las pruebas necesarias y cuando haya terminado elija la opción de menú Command(*Comando*)/Reset All (*Restablecer Todo*) para volver al modo de edición.

NOTA: Para que esta sesión de trabajo funcione correctamente implemente primero el programa de simulación descrito en el apartado **A.8.1.**

A.8. Elaboración de un prototipo virtual utilizando Visual BASIC 3.0.

Entre al entorno de programación Visual BASIC 3.0 y desarrolle un nuevo proyecto.

Aparecerá una forma en blanco titulada Form1. Especifique en ella las siguientes propiedades:

Parámetro	Especificación
Nombre	frmSistema
LinkTopic	frmSistema
LinkMode	1-Source
ControlBox	True

Dentro de esta ventana declare 6 etiquetas con los siguientes nombres: lblTemperatura, lblControl, lblTemporización, iTiempoDiscreto, fTemperatura y fControl. Especifique los siguientes parámetros:

Parámetro	Especificación
lblTemperatura.Caption	Valor de Temperatura:
lblControl.Caption	Valor de la Señal de Control:
lblTemporización.Caption	Tiempo Discreto:
iTiempoDiscreto	0
fTemperatura	0
fControl	0

Declare el procedimiento **Sub iTiempoDiscreto_Change()**, y escriba el siguiente código:

```
Static T0,T1,C0,C1 as single
'Algoritmo de control:  y(k)=0.5u(k-1)+0.8u(k-1)
C1=C0
C0=Val(fControl)
T0=0.05*C1+0.8*T1
fTemperatura=T0
T1=T0
```

Guarde el proyecto y compilelo bajo el nombre c:\scada\simfiles\temperat.exe. Este es el archivo utilizado por SCADA System.

ANEXO B. ARCHIVOS EN VISUAL BASIC 3.0 DEL PAQUETE COMPUTACIONAL SCADA SYSTEM.

La aplicación completa es demasiado extensa para ser listada en esta tesis, por tanto, se ha optado por incluir todos los archivos pertenecientes al programa SCADA System en un diskette adjunto. Cualquier modificación a estos archivos debe hacerse dentro del entorno de programación Visual BASIC 3.0.

ANEXO C. APLICACION SCADA SYSTEM.

Se ha incluido también una copia autorizada del diskette instalador de la aplicación ejecutable **SCADA System versión 1.0 en Español**. Referirse al anexo A para su instalación.