

**Facultad de Ingeniería Eléctrica  
Departamento de Automática y  
Sistemas Computacionales**

## **TRABAJO DE DIPLOMA**

**Título: Desarrollo y optimización del software  
“SupClien 1.0”.**

**Autor: Marcel Hung Sáenz**

**Tutor: MSc. Iván Iglesias Navarro**

**Santa Clara  
“Año 49 de la Revolución”**



Hago constar que el presente trabajo de diploma fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de estudios de la especialidad de Ingeniería en Automática, autorizando a que el mismo sea utilizado por la Institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos, ni publicados sin autorización de la Universidad.

---

Los abajo firmantes certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

---

---

---

## **PENSAMIENTO**

*(...)Nuestro futuro es un futuro de hombres de ciencia.*

*Fidel Castro Ruz.*

## **DEDICATORIA**

A mi madre por su inmenso amor y entrega desinteresada,

A mi padre y abuela,

A toda mi familia y amigos por su ayuda en estos cinco años,

A mi tutor que ha sabido guiarme y enseñarme el valor del trabajo,

## **AGRADECIMIENTOS**

A todos los profesores que a lo largo de este tiempo han sabido educarme, a la Revolución que sin ella todo este sueño habría sido imposible de realizar, a todos los que de una forma u otra han tenido que ver con él.

## TAREA TÉCNICA

1. Análisis bibliográfico
    - Sistemas Supervisores (SCADA), enlace y comunicación.
    - SCADAs Clientes.
  2. Estudios de sistemas supervisores desarrollados previamente. (*SupEN*, *SiSEn*, *MedEn*).
  3. Estudio de herramientas de interoperatividad (*CliSerV*).
  4. Estudio y análisis del software Microsoft SQL Server.
  5. Actualización de la aplicación.
  6. Pruebas y análisis de los resultados (servidores de diferentes protocolos).
  7. Elaboración del informe final.
- 
-

## RESUMEN

En el presente trabajo hemos realizado un estudio minucioso del software (**SupClien 1.0**), hemos analizado sus características fundamentales, tanto estructurales como de funcionamiento, además de otras aplicaciones vinculadas a este como el software LabVIEW en el que fue creado.

Después de este análisis decidimos implementar una nueva versión de este software a la que llamaremos (**SupClien 1.1**). La cual tendrá aspectos diferentes a su predecesora los que harán que la misma alcance un funcionamiento casi óptimo. Empezaremos desde lo mas sencillo ya sea el cambio de color de fondo de las diferentes pantallas, acorde a patrones de diseño hasta la inclusión de una base de datos en Microsoft SQL Server la cual será el fondo para la parte de administración de la aplicación y además contiene las tablas donde estarán almacenadas todas las variables con sus características fundamentales. Otro de los aportes, es la creación del llamado Graficador de Estado. Esta versión mantiene algunos aspectos coincidentes con la anterior ya que consideramos estos de gran importancia. Las pruebas fueron realizadas con éxito y mostradas en el desarrollo del trabajo.

## TABLA DE CONTENIDOS

PENSAMIENTO .....	iv
DEDICATORIA .....	v
AGRADECIMIENTOS .....	vi
TAREA TÉCNICA.....	vii
RESUMEN .....	viii
INTRODUCCIÓN.....	1
Organización del informe .....	3
CAPÍTULO 1. Características de los Sistemas Supervisores Cliente-Servidor.....	5
1.1 Sistemas Supervisores.....	5
1.1.1 Supervisión. Conceptos .....	6
1.1.2 La supervisión en la industria actual.....	7
1.1.3 Prestaciones de los SCADAs.....	8
1.1.4 Requisitos.....	9
1.1.5 Módulos de un SCADA.....	9
1.2 Sistemas de comunicación Cliente Servidor.....	10
1.2.1 Servidor.....	10
1.2.2 Funciones generales de un servidor .....	11
1.2.3 Cliente.....	11
1.2.4 Funciones del Cliente.....	11
1.2.5 Modelo Cliente Servidor.....	12
1.2.6 Hardware Cliente Servidor .....	13
1.3 Conectividad entre aplicaciones (Interconectividad).....	14
1.4 Plataforma SQL Server para Base de Datos .....	15

1.4.1	SQL Server .....	15
1.4.2	Lenguaje de Consulta Estructurado. (SQL) .....	17
1.5	Sistemas Abiertos. Definiciones .....	17
1.5.1	Características generales.....	18
1.5.2	Beneficios .....	19
1.5.3	Interconexión de Sistemas Abiertos .....	20
1.5.4	Niveles del Modelo .....	21
1.6	Software LabVIEW .....	22
	Consideraciones finales del capítulo.....	24
	Referencias.....	25
CAPÍTULO 2. Desarrollo del Sistema Supervisor Cliente “SupClien 1.1” .....		26
2.1	Descripción estructural y funcional general del Sistema “SupClien 1.1” .....	26
2.2	Pantalla Principal .....	29
2.3	Pantallas Secundarias.....	30
2.4	Menú principal .....	30
2.4.1	Menú “Inicio” .....	31
2.4.2	Menú “Administración” .....	33
2.4.3	Menú “File” .....	34
2.4.4	Menú “Visualización”.....	35
2.4.5	Menú “Históricos” .....	36
2.4.6	Menú “Comunicación” .....	37
2.4.7	Menú “Pantallas” .....	39
2.4.8	Menú “Ayuda” .....	39
2.5	Base de Datos.....	40

Consideraciones finales del capítulo.....	41
CAPÍTULO 3. Pruebas, Análisis y Resultados .....	43
3.1 Pruebas y Resultados .....	43
Análisis economico.....	48
Consideraciones finales del capítulo.....	48
CONCLUSIONES .....	49
RECOMENDACIONES.....	50
BIBLIOGRAFÍA .....	51
Anexo 1: Estructura de conexión entre las pantallas del SupClien 1.1 .....	55
Anexo 2: Fragmento del documento .txt en el que se registran las variables y sus características.....	56
Anexo 3: Versión del SupClien 1.1 .....	57
Anexo 4: Carpeta que contiene los archivos .txt los cuales muestran en el nombre el día en que fueron creados. ....	58

## INTRODUCCIÓN

Al principio, en el surgimiento de la computación solo existían grandes ordenadores, los cuales, debido a sus limitadas capacidades de hardware solo realizaban algunas tareas de una manera muy precaria comparadas con las que se efectúan hoy en día. Además, eran de uso exclusivo de grandes y selectas organizaciones. Todo esto ha sufrido enormes cambios, en ocasiones de una manera vertiginosa, sobre todo en lo que a desarrollo del hardware se refiere, en la actualidad existen ordenadores con muchas más prestaciones y rapidez que aquellos pioneros de la informática y a su vez el acceso a los mismos está prácticamente generalizado.

El hombre ha aprovechado este desarrollo de los componentes electrónicos, sobre todo en la capacidad y velocidad de los microprocesadores y ha perfeccionado las aplicaciones que corren sobre estos. Partiendo de esto es que surgen los diferentes Sistemas de Supervisión y Control, dentro de los cuales se destacan los Sistemas Distribuidos y los Sistemas de Adquisición de Datos y Supervisión de Control (SCADA). Estos últimos son softwares de control, capaces de comunicarse con los dispositivos de campo (controladores autónomos, autómatas programables, etc.) y a partir de los datos adquiridos controlar el proceso de forma automática desde la pantalla del operador.

En nuestro país actualmente se lleva a cabo la Revolución Energética, por lo que están tomando una gran importancia los SCADAs que se encargan de la supervisión y control de la generación de energía eléctrica. En este sentido están orientados los sistemas desarrollados en los últimos años por el Departamento de Automática y Sistemas Computacionales de la Universidad Central “Marta Abreu” de Las Villas: Sistema de Supervisión Energética (*SupEN*) y Sistemas de Medición de Variables Eléctricas (*MedEN*). También se encuentra orientado en este sentido el (*Ion Enterprise*) instalado en Varadero. (Melton 2002)

No es solo la generación de electricidad la única industria que se ve beneficiada con la implementación de sistemas SCADA, puede decirse que son cada vez más las empresas que se plantean la automatización de una serie de procesos en su entorno industrial, con el objetivo de mejorar la productividad y aumentar la calidad del producto final y la seguridad en el trabajo.

## Introducción

---

La mayoría de los fabricantes diseñan sus sistemas SCADAs utilizando supervisores servidores y clientes, pero estos solo garantizan la interconexión de los mismos, usando un único protocolo que es el especificado por esa compañía, imposibilitando la posible conexión de supervisores de firmas diferentes.

El **problema científico** de nuestra investigación consiste en la implementación de algunas mejoras al software (**SupClien 1.0**), así como ser consecuentes con sus políticas de interconexión con otros sistemas similares (de diferentes protocolos) a nivel mundial, además, la posibilidad de insertar y mostrar dinámicamente sinópticos o mímicos de los procesos supervisados.

De aquí se deriva que el **objeto de estudio** de nuestro trabajo está relacionado con los mecanismos de comunicación abierta entre servidores y clientes, así como un estudio de las prestaciones y potencialidad del software Microsoft SQL Server, el cual nos será de gran utilidad para desarrollar el perfeccionamiento de nuestra aplicación.

Con el propósito de solucionar el problema científico planteado, se propone el **objetivo general** siguiente: desarrollar las actualizaciones al Sistema Supervisor Cliente (**SupClien 1.0**), mantener su arquitectura abierta, sus posibilidades de comunicación con otras aplicaciones y SCADAs, además de garantizar la inclusión dinámica de Mímicos, Sinópticos e Históricos.

Para lograr el cumplimiento de este objetivo general se trazaron las siguientes **tareas científicas**:

1. Análisis bibliográfico
  - Sistemas Supervisores (SCADA), enlace y comunicación.
  - SCADAs Clientes.
2. Estudios de sistemas supervisores desarrollados previamente. (*SupEN*, *SiSEn*, *MedEn*).
3. Estudio de herramientas de interoperatividad (*CliSerV*).
4. Estudio y análisis del software Microsoft SQL Server.
5. Actualización de la aplicación.

## Introducción

---

6. Pruebas y análisis de los resultados (servidores de diferentes protocolos).
7. Elaboración del informe final.

La **novedad científica** del trabajo consiste en la inclusión del software Microsoft SQL Server, el cual nos da la posibilidad, por ser un gestor de base de datos con grandes potencialidades, de perfeccionar lo relacionado con la administración del sistema y realizar una mejor supervisión y control de las variables medidas, almacenadas en sus tablas, ya que todo estaría contenido en una única base de datos, lo cual daría al traste con un funcionamiento óptimo de nuestro sistema.

Todo esto conlleva a un **aporte científico** que está dado por un **aporte teórico** relacionado con el análisis y sistematización de la bibliografía revisada acerca de los sistemas supervisores cliente-servidor (SCADAs), los diferentes protocolos de comunicación, los sistemas abiertos, la conectividad con las bases de datos y el Software LabVIEW; y por un **aporte práctico** relacionado con la obtención de un nuevo sistema que puede mejorar considerablemente con su implementación la esfera de los servicios y la producción; además de aportar en lo educacional una aplicación desarrollada en LabVIEW, que puede ser utilizada como objeto de estudio en la asignatura de Computación Aplicada.

### **Organización del informe**

El informe del trabajo se estructuró en las siguientes partes: Resumen, Introducción, tres Capítulos, Conclusiones y Recomendaciones, Bibliografía y Anexos.

El Capítulo I “**Características de los Sistemas Supervisores Cliente–Servidor**” es la revisión bibliográfica realizada acerca de los mismos, conceptos, su utilización en la industria actual, interconectividad entre aplicaciones, los sistemas abiertos, las bases de datos y el LabVIEW.

En el Capítulo II “**Sistema Supervisor Cliente “SupClien 1.1”**” está desarrollado lo referente a la estructura y funcionamiento de éste, sus partes principales y como fueron desarrolladas.

En el Capítulo III “**Pruebas y análisis de los resultados**” están expuestas las pruebas que se hicieron a medida que se iba desarrollando el sistema, las pruebas finales y el análisis de los resultados obtenidos en dichas comprobaciones.

## Introducción

---

Las conclusiones de nuestro trabajo están en consonancia con las tareas científicas planteadas, siendo consecuentes con los principales aportes, mejoras y resultados del trabajo.

Gran parte de la bibliografía consultada procede de Internet, además de revisar documentos en copia dura que forman parte de otros trabajos de defensa de maestrías y pregrado. La bibliografía y las referencias se elaboraron haciendo uso del Software Profesional de Gestión Bibliográfica *EndNote 7.0*.

En los anexos se muestran esquemas y gráficos complementarios al trabajo que facilitan en gran medida el entendimiento del mismo.

### **Referencias de la Introducción.**

Melton, J. (2002). Ion Enterprise SCADA. **20-Enero-2005**.

## **CAPÍTULO 1. Características de los Sistemas Supervisores Cliente-Servidor.**

Después de haber analizado la problemática y los antecedentes, pasaremos a concentrarnos en los objetivos de este trabajo y para eso analizaremos la base teórica del mismo. Por ello en este capítulo vamos a tratar lo referente a los Sistemas de Supervisión y Control o también llamados Sistemas SCADA<sup>1</sup>, la interconectividad que pueda existir entre clientes y servidores o entre diferentes aplicaciones de los mismos; hablaremos de los sistemas abiertos y el LabVIEW que será el software que nos facilitará la interconexión y la comunicación hombre máquina en nuestro trabajo.

Este capítulo está dividido por epígrafes, que se refieren a los temas mencionados anteriormente, ellos son:

- 1.1** Sistemas Supervisores.
- 1.2** Sistemas de Comunicación Cliente-Servidor.
- 1.3** Conectividad entre Aplicaciones (Interconectividad).
- 1.4** Plataforma SQL Server para base de datos.
- 1.5** Sistemas Abiertos.
- 1.6** Software LabVIEW.

### **1.1 Sistemas Supervisores**

La incesante demanda de calidad, reducción de gastos, mantenimiento y disminución de tiempo de producción, han provocado la necesidad de automatizar la vigilancia de los procesos. Se espera un alto rendimiento de estos y por tanto debemos eliminar paradas innecesarias, predecir la aparición de situaciones anómalas y actuar rápida y eficazmente en caso de darse estas. Para conseguir estos propósitos, la industria actual se ha dotado de guías de aseguramiento de la calidad y planes de mantenimiento preventivo con el fin de eliminar dichas situaciones.

---

<sup>1</sup>Supervisory Control And Data Acquisition ( Sistemas de Adquisición de datos y Supervisión de Control)

Paralelamente a este esfuerzo por ordenar la gestión de la producción mediante la implantación de estas nuevas metodologías de trabajo, ha habido la voluntad de automatizar e informatizar las industrias de producción con el doble objetivo de centralizar la vigilancia del proceso y de recopilar información relativa a la evolución del mismo. Con este propósito han aparecido los llamados sistemas SCADA.

Estos Sistemas SCADA han sustituido las salas de control por ordenadores o terminales y los bellos pero estáticos y voluminosos sinópticos por pantallas configurables y animadas. Se conservan los gráficos, sustituyendo el papel por representaciones en pantalla de datos almacenados en los discos duros. Hasta el momento se ha cambiado la presentación y modo de almacenamiento de la información, pero no su uso. Continúa siendo el operador de la sala de control quien decide sobre la evolución del proceso a la vista de los datos visualizados.

A todo esto se une que el número de mediciones registradas del proceso ha aumentado considerablemente a la vez que lo hacen en complejidad y los sistemas de control han tomado el relevo a los ajustes manuales. Frente a este incremento de información, cambiante y dinámica, el operador de control necesita de nuevas ayudas a su tarea de vigilancia y supervisión del proceso. Así, la supervisión se entiende como el soporte a la sistematización en el seguimiento de estos procesos, automatizando en la medida que sea posible las tareas necesarias para dicha actividad, llegando a ser sistemas completos con capacidad para la toma de decisiones.

Más específicamente en los siguientes epígrafes se hablará acerca de las características esenciales de los sistemas supervisores y de los sistemas SCADAs.

### **1.1.1 Supervisión. Conceptos**

Un sistema de supervisión no es más que una o varias aplicaciones que hará uso de toda la información adquirida por cualquier vía, relativa al proceso, para decidir sobre su correcto (o no) funcionamiento y proponiendo, a su vez, las acciones adecuadas para mantener los objetivos productivos fijados para el mismo.(Meléndez J. 2002)

El sistema de supervisión será el encargado de:

- 1. Registrar la evolución del proceso y detectar desviaciones indeseadas en las variables.**

2. Analizar estas desviaciones y deducir el motivo. Elaborar un diagnóstico de la situación.
3. Resolver la situación en línea, en caso de ser posible, o tomar las medidas adecuadas para que no vuelva a suceder.

Después de haber analizado el concepto de un sistema de supervisión y sus objetivos podemos ver que la idea es simple, aprovechar toda información para asegurar el funcionamiento correcto del proceso en todo instante. El modo de implementar esta idea es más oscuro y para ello se aprovechan todos los recursos existentes, resultando múltiples técnicas para abordar el problema de la supervisión de acuerdo con la información disponible. De igual forma las diferentes técnicas conllevan a diversos enfoques del problema, que dificulta el establecer una metodología común para la supervisión.

A pesar de ello, podemos considerar que para acometer la supervisión cabe recorrer tres etapas fundamentales: la detección de fallos, el diagnóstico de éstos y finalmente la reconfiguración del sistema, que le ha de permitir continuar operando de acuerdo con las especificaciones fijadas. En caso de no realizarse las tres etapas podemos considerar que el sistema de supervisión no es completo. En este caso, el entorno habitualmente desempeña tareas de vigilancia y asistencia al operario y se concibe como sistema de monitorización. Debe considerarse que los límites de uno y otros conceptos a menudo son difusos y se utilizan alternativamente para designar los mismos entornos.

### **1.1.2 La supervisión en la industria actual**

En la industria actual, tanto en Cuba como en el resto del mundo, han proliferado los llamados sistemas SCADA como sistemas de supervisión. Como su nombre indica, estos se atribuyen las funcionalidades de centralización o adquisición de datos, control y supervisión. Por tanto, se puede decir que representan el estado actual de esta disciplina en los entornos industriales.

Un SCADA es una aplicación o conjunto de aplicaciones de software con acceso a plantas, mediante comunicación digital con los instrumentos, actuadores e interfaz gráfica de alto nivel con el usuario. Las siglas SCADA se utilizan tanto para designar estas aplicaciones como los entornos utilizados para su desarrollo, asociando a ellos las marcas comerciales.

En este tipo de sistemas usualmente existe un ordenador que efectúa tareas de supervisión y gestión de alarmas, así como el tratamiento de datos y control de procesos. La comunicación se realiza mediante buses especiales o redes, todo lo cual se ejecuta normalmente en tiempo real y están diseñados para dar al operador de planta la posibilidad de supervisarlos y controlarlos.

### 1.1.3 Prestaciones de los SCADAs

En la bibliografía consultada encontramos varios autores que plantean de una forma u otra las diferentes prestaciones que nos brindan los sistemas SCADA's, tal es el caso de los profesores Carlos C. Lozano y Cristóbal Romero Morales en su artículo "*Introducción a SCADA*" (Lozano C. 2002), y de la Dra. Carmen D'Sousa con su monografía "*Sistemas SCADA*" (D'Sousa C. 2005). Lo planteado por esta última autora se pone a consideración por ser lo que más se ajusta a lo que se aplicará en el presente trabajo. D'Sousa expresa que un paquete SCADA debe estar en disposición de ofrecer las siguientes prestaciones:

1. Posibilidad de crear paneles de alarma que exigen la presencia del operador para reconocer una parada o situación de alarma con registro de incidencias.
2. Generación de históricos de señales de planta que pueden ser volcados para su proceso sobre una hoja de cálculo.
3. Ejecución de programas que cambian la ley de control o incluso anulan o modifican las tareas asociadas al autómata bajo ciertas condiciones.
4. Posibilidad de programación numérica que permite realizar cálculos aritméticos de elevada resolución sobre la CPU del ordenador.

Todas estas acciones se llevan a cabo mediante un paquete de funciones que incluye zonas de programación en un lenguaje de uso general (C, Pascal, o Basic), lo cual confiere una potencia muy elevada y una gran versatilidad. Algunos SCADAs ofrecen librerías de funciones para lenguajes de uso general que permiten personalizar de manera muy amplia la aplicación que desee realizarse con dicho SCADA.(D'Sousa C. 2005)

### 1.1.4 Requisitos

Un SCADA debe cumplir varios objetivos para que su instalación sea perfectamente aprovechada:

- Deben ser sistemas de arquitectura abierta, capaces de crecer o adaptarse según las necesidades cambiantes de las empresas o de los sistemas en sí.
- Deben comunicarse con total facilidad y de forma transparente al usuario con el equipo de planta y con las diferentes áreas dentro de la empresa (redes locales y de gestión).
- Deben ser programas sencillos de instalar, sin excesivas exigencias de hardware, fáciles de utilizar y con interfaces amigables con el usuario.

### 1.1.5 Módulos de un SCADA

Cuando se entra a analizar lo que serían los posibles módulos de un SCADA se pueden encontrar fundamentalmente dos tipos. Por un lado los módulos de un software SCADA, completamente orientado a una determinada aplicación y por otra parte los de los paquetes de software SCADA que permiten la configuración completa de aplicaciones de control y supervisión.

Analizando ambos casos podemos concluir que los módulos de software que hacen posible la adquisición, supervisión y control son los siguientes:

- **Configuración:** permite al usuario definir el entorno de trabajo de su aplicación, según la disposición de pantallas requerida y los niveles de acceso para los distintos usuarios.
- **Interfaz gráfico del operador:** proporciona al operador las funciones de control y supervisión de la planta.
- **Módulo de proceso:** ejecuta las acciones de mando pre-programadas a partir de los valores actuales de variables leídas. La programación se realiza por medio de bloques de programa en lenguaje de alto nivel (como C o Basic).
- **Gestión y archivo de datos:** se encarga del almacenamiento y procesado ordenado de los datos, de forma que otra aplicación o dispositivo pueda tener acceso a ellos.

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

El rol supervisor de los SCADAs actuales queda relegado básicamente a la activación y registro de “alarmas” asociadas al traspaso de límites por parte de variables analógicas. Una revisión detenida de la diversidad de productos de esta índole que se encuentran en el mercado permite afirmar de la mayoría de ellos, que aunque cumplan perfectamente con las tareas de adquisición de datos y control compartiendo responsabilidad con los dispositivos de campo tipo PLC, distan de ser entornos de supervisión expertos. Es decir, la integración de conocimiento especializado y su utilización en tareas de decisión queda reducida a su característica de sistema abierto.

### 1.2 Sistemas de comunicación Cliente Servidor

Antes de entrar a construir un sistema cliente-servidor, debemos formularnos una pregunta muy importante cuya respuesta influye notablemente en el costo del servidor y el cliente, además de afectar en gran medida la robustez y seguridad del sistema, dicha pregunta sería: ¿Qué parte de la aplicación debe ser hecha por el cliente y cuál por el servidor?

#### 1.2.1 Servidor

Un servidor en informática o computación es un programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes. Algunos servicios habituales son los de archivos, que permiten a los usuarios almacenar y acceder a los archivos de un ordenador y los servicios de aplicaciones, los cuales realizan tareas en beneficio directo del usuario final.

El servidor no es necesariamente una máquina de última generación grande y monstruosa, un servidor puede ser desde una computadora antigua (Pentium 2) hasta una máquina sumamente potente (servidores Web y bases de datos grandes, procesadores especiales de mayor cantidad de gigas de memoria).

### 1.2.2 Funciones generales de un servidor

Los servidores en una red cliente-servidor son los procesos que proporcionan información, recursos y servicios a los clientes de la red. Cuando un cliente pide un recurso, por ejemplo, un archivo, datos de una base de datos, acceso a aplicaciones remotas o impresión centralizada, el servidor proporciona estos recursos al cliente. Como se mencionó antes, los procesos inherentes a él pueden residir en una máquina que también actúa como cliente de otro servidor. Además de proporcionar este tipo de recursos, este puede dar acceso a otras redes, actuando como un servidor de comunicaciones que conecta a otros servidores, mainframes o minicomputadoras que actúan como hosts de la red. (Rosen J. 2000)

### 1.2.3 Cliente

El cliente recibe los servicios que ofrece un servidor. El término se usó inicialmente para dispositivos que no eran capaces de ejecutar programas por sí mismos, pero podían interactuar con ordenadores remotos por red. Estos terminales "tontos" eran clientes de los ordenadores centrales de tiempo compartido.

Pueden existir diferentes tipos de clientes como son los *livianos* y los *pesados*: los primeros no son más que aquellas computadoras en una arquitectura de red cliente-servidor que tienen muy poca o ninguna lógica de programa, por lo tanto dependen principalmente del servidor central para las tareas de procesamiento; los segundos (también "cliente rico" o "cliente grueso"), es un término de arquitectura cliente-servidor para aquel que realiza el grueso de las operaciones de procesamiento de datos. Los datos en sí mismos son almacenados en el servidor.

### 1.2.4 Funciones del Cliente

Los clientes en una red cliente-servidor son las máquinas o procesos que piden información, recursos y servicios a un servidor. Estas peticiones pueden ser diferentes: proporcionar datos de una base de datos, aplicaciones, partes de archivos o archivos completos a la máquina cliente. Esta disposición permite a la máquina cliente ser relativamente pequeña. Para cada tipo de entorno de cliente hay habitualmente software específico y a veces hardware, con algún software y hardware análogo en el servidor (Rosen J. 2000).

Los servidores pueden ser sistemas operativos diferentes como Windows NT, Windows 95, OS/2 y Unix. Este último es popular porque como sistema operativo de servidores puede ser utilizado en muchos tipos de configuraciones sobre máquinas servidor, además como servidores de archivos y de impresión.

Una vez expuesta las características de los clientes y lo servidores por separado, así como sus funciones, entonces a modo de resumen, se entrará ya en la concepción como sistema del modelo cliente-servidor.

### **1.2.5 Modelo Cliente Servidor**

La tecnología denominada Cliente-Servidor es utilizada por todas las aplicaciones de Internet/Intranet. Su funcionamiento se describirá en el siguiente párrafo:

Un cliente funciona en su ordenador local, se comunica con el servidor remoto y pide a éste información. El servidor envía la información solicitada. Un único servidor típicamente sirve a una multitud de clientes, ahorrando a cada uno de ellos el problema de tener la información instalada y almacenada localmente.

Los sistemas Cliente-Servidor pueden ser de muchos tipos, dependiendo de las aplicaciones que el servidor pone a disposición de los clientes. Entre otros, existen:

- Servidores de impresión, mediante el cual los usuarios comparten impresoras.
- Servidores de archivos, con los cuales los clientes comparten discos duros.
- Servidores de bases de datos, donde existe una única base de datos.
- Servidores de lotus notes, que permiten el trabajo simultáneo de distintos clientes con los mismos datos, documentos o modelos.

Los Servidores Web también utilizan la tecnología cliente-servidor, aunque añaden aspectos nuevos y propios a la misma.

El modelo Cliente-Servidor que predomina en la actualidad permite descentralizar el procesamiento y los recursos, sobre todo de cada uno de los servicios y de la visualización de la interfaz gráfica de usuario. Esto hace que ciertos servidores estén dedicados solo a una aplicación determinada y por lo tanto ejecutarla en forma eficiente. Entonces él o los clientes estarían en distintos puntos tan alejados como se quiera o como se pueda.

Es un elemento primordial la presencia de un medio físico de comunicación entre las máquinas. Dependerá de la naturaleza de este medio la viabilidad del sistema.

### 1.2.6 Hardware Cliente Servidor

La arquitectura Cliente-Servidor permite al usuario en una máquina, (la que ya se conoce como cliente), requerir algún tipo de servicio de otra a la que está unido (llamada servidor) mediante una red, por ejemplo, una LAN (Red de Área Local) o una WAN (Red de Área Mundial). Estos servicios pueden ser peticiones a una base de datos, de información contenida en archivos o los archivos en sí mismos o peticiones de imprimir datos en una impresora asociada. Aunque clientes y servidores suelen verse como máquinas separadas, pueden de hecho, ser dos áreas aparte en la misma máquina. Por tanto, una única máquina Unix puede ser al mismo tiempo cliente y servidor. Además, una máquina cliente unida a un servidor puede ser a su vez servidor de otro cliente y el servidor puede ser un cliente de otro servidor en la red. También es posible tener al cliente corriendo en un sistema operativo y el servidor en otro distinto.

Hay varios tipos comunes de máquinas clientes en entornos cliente-servidor. Uno de los clientes más populares es una computadora personal basada en Intel que ejecuta aplicaciones de DOS en un entorno Windows. Otro ejemplo de cliente, muy popular, es una terminal X; de hecho el sistema X Windows es un modelo cliente-servidor clásico. Hay también clientes Unix que ejecutan sistemas operativos como UnixWare. Un servidor que pide cosas a otro servidor es a su vez un cliente.

Aprovechando todas estas prestaciones que nos brinda el uso de un modelo cliente-servidor es que en este trabajo se decide realizar una aplicación *supervisor cliente* con comunicación abierta, el cual se conecta a cualquier *supervisor servidor* que proporcione la información necesaria para realizar una supervisión de un mismo proceso desde diferentes áreas de la misma industria o empresa, como son, la administración, mantenimiento o la propia gerencia. Es decir con este sistema se tendrá acceso a cualquier información del proceso en cuestión, sin necesidad de estar conectado físicamente al sistema que se supervisa y controla.

### 1.3 Conectividad entre aplicaciones (Interconectividad)

La necesidad de interoperabilidad entre los equipos, no sólo en el ambiente de las redes telemáticas sino también en las soluciones de protocolos de redes industriales, ha obligado a las diferentes empresas desarrolladoras de software a crear nuevos productos que permitan la comunicación entre diferentes sistemas que fueron diseñados para ser totalmente incompatibles.

Por ejemplo, en el ambiente industrial ha sido muy agresiva la batalla entre protocolos de redes de bus de campo (redes industriales especializadas para comunicar sensores, actuadores, controladores y otros dispositivos industriales); batalla que obligaba al usuario a ser fiel a una marca de dispositivos para toda su línea de producción, dada su incompatibilidad con los demás protocolos de comunicación. Sin embargo, en los últimos años se ha impulsado una unificación masiva de los sistemas industriales por parte de los principales desarrolladores de software del sector.

Esto ha traído consigo, que en los sistemas modernos exista una gran conectividad entre las aplicaciones en una misma PC y con el exterior, tanto por software como por hardware. El sistema operativo Windows ofrece varias posibilidades para la comunicación entre diferentes software.

Los sistemas de supervisión actuales explotan al máximo estas posibilidades de interconexión entre las que se pueden plantear:

- Enlace dinámico de bibliotecas (DLL).
- Intercambio dinámico de datos (DDE).
- Enlace con objetos (OLE).
- OLE para control de procesos (OPC).
- Atención a puertos de entrada/salida.
- Comunicación serie.
- Conexión a redes, uso del protocolo TCP/IP.
- Uso de drivers específicos para adquisición de datos (DAQ).

- Trabajo con bases de datos y lenguaje de consulta SQL.

A continuación una breve explicación acerca de algunos de estos mecanismos antes mencionados, usados para la interconexión entre aplicaciones.

### **1.4 Plataforma SQL Server para Base de Datos**

#### **1.4.1 SQL Server**

Microsoft SQL Server es una herramienta de gestión de bases de datos completamente relacional rápido y dinámico, que funciona sobre sistema operativo Windows 2000/2003 Server, ofreciendo un único punto de acceso a los datos del servidor por parte de las aplicaciones clientes, con la característica de que el procesamiento de la información es realizado por el servidor, independientemente de la presentación, que la efectúan las aplicaciones clientes; por lo que se puede decir que SQL Server es una apuesta certera de Microsoft en el mundo de los SGBD<sup>2</sup>.

SQL Server puede gestionar grandes bases de datos con millones de registros y permite a las aplicaciones clientes controlar la información recuperada del servidor, mediante el uso de varias herramientas y técnicas especializadas. Su funcionamiento es algo más complicado comparado con otros softwares de gestión de base de datos ya que para alcanzar un rendimiento óptimo, es necesaria la aportación técnica de un informático para configurarla correctamente. Por esta razón se decidió utilizar el Microsoft Access para la primera versión del sistema; aunque en este punto, la configuración no es tan costosa (ni en tiempo, ni económicamente) como por ejemplo pudiera ser el Oracle. (Stanek William R. 2000)

A continuación comentaremos algunas de las características principales que posee este gestor de base de datos:

- Esta herramienta presenta una gran escalabilidad, es decir, permite la existencia de miles de usuarios conectados a una base de datos.
- Presenta una gran potencia para Windows NT.

---

<sup>2</sup> Sistemas de Gestión de Bases de Datos.

- La complejidad de la gestión y de la administración de la base de datos se reduce gracias al interfaz gráfico que presenta.
- Es un gestor sencillo de instalar.
- Puede generar código HTML de forma automática.
- Está diseñada para entornos .NET (Intranet, Internet).
- Tienen un costo bastante bajo.
- Está orientada para entornos de desarrollo de Microsoft: Visual Basic o Visual C++.

Como podemos ver este motor de base de datos, basado en un modelo relacional, posee muy buenas características tanto para DBA<sup>3</sup>, como de interacción con la herramienta de construcción de aplicaciones.

SQL Server presenta muchas ventajas, para estandarizar todos los datos obtenidos, algunas de estas ventajas son presentadas a continuación:

- Con unos ciertos conocimientos de administración un usuario medio puede crear y gestionar bases de datos de un gran número de usuarios y clientes.
- Instalación en ordenadores en modo de uso local o bien en modo remoto, para aprovechar las características cliente/servidor.
- Configuración dinámica de la memoria y de la gestión de ocupación en disco
- Su implementación en redes es óptima lo que da una cierta confianza al entorno y posibilita un acceso rápido a todas las herramientas informáticas necesarias para sacar el trabajo adelante.
- Asistentes para la administración de las bases de datos, así como para diseñar su estructura.
- Una integración sofisticada, unida a la rapidez extrema a la hora de transformar los datos produce una sensación de credibilidad al programa, por lo que se puede confiar en este software de usuario para ser utilizado principalmente a nivel empresarial.

---

<sup>3</sup> Data Base Administrator (Administrador de base de Datos).

- No necesita de un DBA o consultor específico lo que favorece el mantenimiento por la propia empresa con menor gasto.
- Al poder manejar porciones de la base de datos específicas, cuya utilidad radica en que al no trabajar con el modelo relacional completo, que puede incluir un gran número de relaciones (tablas, archivos), el análisis y manipulación de los datos se facilita bastante.

En definitiva, se trata de una buena apuesta para aquellas empresas que precisen una base de datos corporativa para uso interno general, para comunicación por Intranet e Internet.(Eisenberg A., Kulkarni et al. 2004)

### **1.4.2 Lenguaje de Consulta Estructurado. (SQL)**

El lenguaje de consulta estructurado (SQL) es un lenguaje de base de datos normalizado, utilizado por los diferentes motores de bases de datos para realizar determinadas operaciones con ellos o sobre su estructura. Pero como sucede con cualquier sistema de normalización hay excepciones para casi todo; de hecho, cada motor de base de datos tiene sus peculiaridades y lo hace diferente de otro.

### **1.5 Sistemas Abiertos. Definiciones**

Actualmente existen en el mundo gran cantidad y diversidad de sistemas de medición y supervisión, cada cual adaptado a indicadores y requerimientos específicos para uno u otro dispositivo de medición o analizador de red.

La tecnología, por su parte, desarrolla sus productos cada vez con mayor velocidad, ofreciendo nuevas posibilidades y mejoras, pero que muchas veces no son compatibles con las tecnologías y sistemas ya instalados.

De aquí surge la necesidad de poder acceder a los nuevos sistemas y compartir información entre entornos o usuarios diferentes. Además, esto favorece la competencia dando lugar a mejores precios para los posibles usuarios de los mismos.

Para dar respuesta a todos estos problemas aparece el concepto de **Sistema Abierto**.

Según ISO<sup>4</sup> un sistema abierto es "todo sistema informático capaz de interconectarse con otros de acuerdo con normas establecidas".

Para X/Open<sup>5</sup> se trata de "entornos diseñados e implantados de acuerdo con normas ampliamente divulgadas y básicamente independientes de los fabricantes".

Otra definición es la proporcionada en el documento "**Directrices para una Arquitectura Informática**" redactado por la Comisión de las Comunidades Europeas donde se establece que:

"Sistemas Abiertos son aquellos sistemas y componentes que pueden ser especificados y adquiridos de fuentes distintas en un mercado competitivo. Las especificaciones de los Sistemas Abiertos deben ser controladas por organizaciones internacionales de normalización o, al menos, por un tipo de especificado tan independiente como sea posible, con un amplio grado de aceptación en el mercado. Una especificación de Sistema Abierto no debe ser propiedad de un único suministrador y debe estar disponible sin costo".

Finalmente para el **Gartner Group**, por citar un observador independiente, "los sistemas abiertos son aquellos cuyas especificaciones están aprobadas, publicadas y respaldadas por organismos independientes de normalización como por ejemplo ISO, POSIX y X/Open, entre otros".

### 1.5.1 Características generales

Todas las definiciones anteriores coinciden en que un Sistema Abierto debe cumplir las características de:

- **Interoperabilidad:** posibilidad de enlazar ordenadores y sistemas diferentes para trabajar conjuntamente, dando la sensación de ser un único sistema.
- **Portabilidad:** posibilidad de que una misma aplicación pueda funcionar en distintas plataformas físicas.

---

<sup>4</sup> International Standardization Organization (Organización Internacional de Normalización).

<sup>5</sup> Consorcio fundado en 1984, dedicado al desarrollo de estándares abiertos en el campo de las tecnologías de la información.

- **Crecimiento o Escalabilidad:** posibilidad de aumentar la potencia de la aplicación con sólo aumentar la de la plataforma física en la que se ejecuta.

Con el paso del tiempo, se afinó la definición de los sistemas abiertos. Estos deben utilizar componentes informáticos basados en estándares, independientemente del proveedor. Para diseñar un sistema abierto se deben contemplar estándares en el sistema operativo, sistema de comunicación, base de datos, interfaz de usuario, herramientas de desarrollo y los sistemas de administración.

### 1.5.2 Beneficios

Los beneficios que proporcionan los sistemas abiertos a distribuidores, desarrolladores, compradores y usuarios se centran en:

- **Elección de la plataforma física**

Los conceptos de portabilidad y escalabilidad de los sistemas abiertos permiten la elección de diferentes plataformas físicas en lugar de permanecer ligados a arquitecturas de un único fabricante. Esto permite personalizar las ofertas y servicios a precios competitivos. Las plataformas de sistemas abiertos pueden incluir ordenadores portátiles o personales de mesa, estaciones de trabajo, ordenadores con multiproceso simétrico y con proceso paralelo o tecnologías basadas en Internet.

- **Elección de los equipos físicos auxiliares y periféricos**

Los sistemas abiertos amplían la elección del usuario tanto en los tipos de periféricos disponibles (fax/módem, tarjetas de sonido, lectores CD-ROM e impresoras), como en el número y variedad de productos dentro de cada categoría. Esto se debe a que la interoperabilidad y las APIs<sup>6</sup> garantizan que los periféricos que soporten un sistema abierto sean compatibles con un gran conjunto de plataformas físicas. De esta forma, los usuarios pueden reducir su riesgo e incertidumbre cuando eligen productos que son compatibles.

---

<sup>6</sup> Application Program Interface (Interface de Aplicación de Programas).

- **Elección de equipos lógicos**

Un entorno que permite que las aplicaciones corran en distintas plataformas físicas proporciona las máximas ventajas. La utilización de interfaces de usuario comunes entre plataformas permite reducir el tiempo de aprendizaje y los costos, además de proporcionar una mejora de la productividad del usuario.

- **Elección en precio**

Los sistemas abiertos fomentan la competencia de precios al atraer mayor número de compradores y vendedores al mercado. Los usuarios pueden tomar su decisión seleccionando el precio y las características que desean.

- **Campos de Aplicación**

Los Sistemas Abiertos están presentes en:

- Ordenadores personales y estaciones de trabajo que presentan cada vez mayores prestaciones gráficas y de comunicación entre máquinas.
- Procesos y aplicaciones cooperativas en las que es posible la intervención de varios sistemas o equipos. En este sentido se relaciona con los conceptos de cliente/servidor y de programación orientada a objetos.
- Aplicaciones relacionadas con servicios de red y conectividad entre grandes entornos.

### **1.5.3 Interconexión de Sistemas Abiertos**

Para sistematizar las normas que deben seguir los equipos y las redes al comunicarse y establecer un marco de referencia independiente de los distintos fabricantes para garantizar la compatibilidad de los sistemas de comunicación, la Organización Internacional de Normalización, estableció en los años 80 un modelo al que denominó modelo para la Interconexión de Sistemas Abiertos, cuyas siglas, OSI<sup>7</sup>, hacen referencia a su nombre en inglés.

El modelo OSI describe una arquitectura jerárquica de 7 niveles y son las normas de comunicaciones en las que se fundamenta el protocolo TCP/IP. Para explicar cada uno de

---

<sup>7</sup> Open Systems Interconnection (Modelo de Interconexión de Sistemas Abiertos).

los niveles del modelo de referencia en relación a la transmisión de paquetes de datos se suele partir del superior que se llama nivel de Aplicación y es el más cercano al usuario. A este se va añadiendo y aclarando la información correspondiente a cada uno de los siguientes niveles hasta llegar al más básico, que da normas para la comunicación física.

### 1.5.4 Niveles del Modelo

**Aplicación:** Este nivel hace referencia a las normas que deben cumplir los protocolos con relación a las aplicaciones o servicios, como por ejemplo, un navegador, un cliente de correo electrónico, un procesador de textos, una hoja de cálculo y una base de datos. El protocolo obliga a las aplicaciones que van a manejarse dentro de una red de transmisión de paquetes a cumplir ciertas normas, que le permitan hacerse entender por las aplicaciones residentes en otras máquinas de la red.

**Presentación:** En este nivel se establecen las normas que se deben cumplir en relación a los códigos de caracteres. Esta información se añade a la del nivel anterior y da a conocer las características de los datos que maneja una aplicación concreta.

**Sesión:** Establece las normas que se deben cumplir en relación a la gestión de las conexiones de las aplicaciones en la red. Son como las normas de cortesía, señalan quién debe hablar y cuándo debe hacerlo, para pedir la palabra y que el interlocutor preste atención. Igualmente, la información relacionada con este nivel, se añade al paquete de datos.

**Transporte:** En este nivel se establecen las normas para identificar la procedencia y el destino de los datos que han de ir de un ordenador a otro. La información de este nivel se añade al paquete de datos, donde ya están las de los niveles anteriores.

**Red:** Este nivel establece las normas a cumplir con relación a la manera en que los paquetes transcurren por la red. La información que se añade al paquete de datos en este nivel, le permitirá moverse por todo el entramado que forma la red, buscando siempre el mejor camino para llegar a su destino final.

**Nivel de enlace:** Se establecen las normas para conseguir una transmisión fiable y libre de errores. La información añadida en este nivel se utiliza para gestionar la comunicación

entre dos máquinas de la red, enlazándolas entre las distintas máquinas por las que van pasando los paquetes de datos.

**Nivel físico:** En este se establecen las normas que deben cumplir los protocolos con relación a las características físicas de todos los componentes relacionados con la transmisión de datos a través de toda la red: estándares de módem, líneas de transmisión y controladores de comunicaciones.

### 1.6 Software LabVIEW

El LabVIEW es el software que se utilizará para el desarrollo de este supervisor cliente. El mismo posee diferentes mecanismos de conectividad o comunicación, lo que permite gran flexibilidad y a su vez realizar aplicaciones abiertas en la comunicación.

Es un revolucionario sistema de programación gráfica para aplicaciones que involucren adquisición, control, análisis y presentación de datos.

Los programas desarrollados mediante LabVIEW se denominan Instrumentos Virtuales (VI), porque su apariencia y funcionamiento imitan los de un instrumento real. Sin embargo son análogos a las funciones creadas en los lenguajes de programación convencionales. Los VI tienen una parte interactiva con el usuario y otra parte de código fuente, y aceptan parámetros procedentes de otros VI. Todos los VI tienen un panel frontal y un diagrama de bloques. Las paletas contienen las opciones que se emplean para crearlos y modificarlos. A continuación se procederá a realizar una somera descripción de estos conceptos.

#### A) Panel Frontal

Se trata de la interfaz gráfica del VI con el usuario. Esta interfaz recoge las entradas procedentes del usuario y representa las salidas proporcionadas por el programa. Un panel frontal está formado por una serie de botones, pulsadores, potenciómetros y gráficos. Cada uno de ellos puede estar definido como un control o un indicador. Los primeros sirven para introducir parámetros al VI, mientras que los indicadores se emplean para mostrar los resultados producidos, ya sean datos adquiridos o resultados de alguna operación.

#### B) Diagrama de bloques

El diagrama de bloques constituye el código fuente del VI. En él se realiza la implementación del programa o del VI para controlar o realizar cualquier procesamiento de las entradas y salidas que se crearon en el panel frontal. El diagrama de bloques incluye funciones y estructuras integradas en las librerías que incorpora el LabVIEW. El lenguaje de programación del mismo es gráfico (G) y las funciones y estructuras son nodos elementales. Son análogas a los operadores o librerías de funciones de los lenguajes convencionales. Los controles e indicadores que se colocaron previamente en el panel frontal, se materializan en el diagrama de bloques mediante los terminales.

El LabVIEW posee dos modos de trabajo, el de diseño y el de ejecución. En el primero se realiza toda la programación funcional de los controles e indicadores y en el segundo se visualiza el funcionamiento de los mismos. A continuación se mencionan algunas características de este software:

- Diseñado para instrumentación, control, procesamiento y presentación de la información.
- Usa la programación por flujo de datos.
- Velocidad de código compilado similar al “C”.
- Multitarea (se pueden crear caminos de flujos de datos independientes o paralelos. Se pueden ejecutar aplicaciones mientras se corren otras).
- Controles e indicadores para la presentación de la información (gráficos, interruptores, indicadores visuales, tanques, termómetros).
- Diferentes posibilidades para la comunicación con el exterior, uso de puertos, drivers en DDL, comunicación serie, enlace y comunicación con redes usando el protocolo TCP/IP.
- Programación modular jerárquica, cada VI confeccionado puede ser una función de otra aplicación.
- Posibilidad de uso de intercambio dinámico de datos DDE.
- Posibilidad de uso de otros objetos de otras aplicaciones a través de OLE.

Todas estas características a la hora de emplear este software dan al traste con una aplicación muy eficiente, rápida, segura y con muchas prestaciones. Es por ello que a continuación mencionamos algunas de las ventajas que proporciona el empleo de LabVIEW:

- Se reduce el tiempo de desarrollo de las aplicaciones al menos de 4 a 10 veces, ya que es muy intuitivo y fácil de aprender.
- Dota de gran flexibilidad al sistema, permitiendo cambios y actualizaciones tanto del hardware como del software.
- Da la posibilidad a los usuarios de crear soluciones completas y complejas.
- Con un único sistema de desarrollo se integran las funciones de adquisición, análisis y presentación de datos.
- El sistema está dotado de un compilador gráfico para lograr la máxima velocidad de ejecución posible.
- Tiene la posibilidad de incorporar aplicaciones escritas en otros lenguajes.

Para el empleo del LabVIEW no se requiere gran experiencia en programación, ya que se emplean íconos, términos e ideas familiares a científicos e ingenieros, y se apoya sobre símbolos gráficos en lugar de lenguaje escrito para construir las aplicaciones. Por ello resulta mucho más intuitivo que el resto de los lenguajes de programación convencionales.

### **Consideraciones finales del capítulo**

En el desarrollo de este capítulo se ha hecho un análisis de la bibliografía existente en el campo de los sistemas de supervisión cliente-servidor. Primeramente se trataron temas referentes al estado actual de los sistemas supervisores, la necesidad del empleo de los mismos y los beneficios de su utilización en la industria actual, analizándose además los sistemas de comunicación cliente-servidor; de estos se definió claramente los términos cliente y servidor al igual que las funciones fundamentales de cada uno de ellos.

Por otro lado se expusieron ampliamente los principales protocolos de comunicación y la necesidad de la estandarización de los mismos para lograr una mayor interoperatividad entre los sistemas, de aquí pasamos al estudio de la plataforma SQL Server, para

posteriormente analizar los sistemas de comunicación abiertos y finalizar con las características principales del LabVIEW.

De lo antes expuesto y valorando la importancia que tienen los sistema de supervisión y control en la industria actual, llegamos a la conclusión de que es necesario perfeccionar este sistema para usarlo en los niveles más altos de la pirámide de control, manteniendo sus propiedades de leer datos procedentes de equipos y/o aplicaciones de software y brinde la posibilidad de emplear determinados protocolos de comunicación para la interconexión de los mismos.

### **Referencias**

D'Sousa C. (2005). SISTEMAS SCADA. **22-Enero-2006.**

Eisenberg A., K. Kulkarni, et al. (2004). SQL 2003. 15-Marzo-2005.

Lozano C., R. M., Cristóbal. (2002). Introducción a SCADA. 10-Febrero-2004.

Meléndez J., C. J. (2002). Supervisión de Procesos Industriales. 15-Febrero-2004: 120p.

Rosen J. (2000). El Modelo Cliente-Servidor. 15-Marzo-2005.

Stanek William R. (2000). MICROSOFT SQL SERVER 2000. MANUAL DEL ADMINISTRADOR., Editorial McGraw-Hill. 15-febrero-2004: 646 p.

## **CAPÍTULO 2. Desarrollo del Sistema Supervisor Cliente “SupClien 1.1”**

En el capítulo anterior, se expuso toda la base teórica de la que parte este trabajo. Ahora pretendemos realizar un análisis de la estructura general, las partes principales, utilidad, importancia y novedad del sistema. Para esto el capítulo se ha dividido en los siguientes epígrafes:

**2.1** Descripción estructural y funcional general del Sistema “SupClien 1.1”.

**2.2** Pantalla Principal.

**2.3** Pantallas Secundarias.

**2.4** Menú Principal.

**2.5** Base de datos.

### **2.1 Descripción estructural y funcional general del Sistema “SupClien 1.1”**

En una industria existen áreas que no están vinculadas directamente al proceso, por ejemplo el departamento de mantenimiento, la dirección, administración o gerencia. Con el objetivo de que se pueda tener un pleno conocimiento del funcionamiento de un lazo de control o de un proceso completo sin necesidad de estar supervisando directamente a los mismos, se creó la versión anterior de este sistema. En el desarrollo de nuestro trabajo pretendemos mejorar y enriquecer lo ya desarrollado, así como mantener algunos aspectos que consideramos de trascendencia para nuestra investigación. Nuestro sistema “**SupClien 1.1**”, al igual que su versión anterior está conectado por medio de diferentes protocolos de comunicación a los Supervisores que están en el nivel más bajo de la pirámide de control, adquiriendo las señales o variables, procesándolas y controlando el proceso en sí, tomando de estos sólo las variables que le sean necesarias para desarrollar la supervisión en los niveles superiores.

Nuestra aplicación se desarrolla completamente sobre la plataforma Windows como Sistema Operativo. Comprobada la efectividad y eficiencia del lenguaje de programación gráfico LabVIEW, decidimos reutilizarlo para implementar las variaciones de nuestra versión. Esto nos permitió crear la aplicación central que va a dominar el funcionamiento total de nuestro sistema. Por otra parte y como actualización, utilizamos el software

Microsoft SQL Server sobre el cual desarrollamos lo que sería la base de datos del sistema (**BaseDatosSupClient.dbo**). Las características de esta base de datos será explicada en el epígrafe 2.5.

Luego de analizar diferentes posibilidades decidimos mantener el núcleo central de nuestra aplicación conformado por cuatro tareas fundamentales, las que nos permitirán llevar a cabo el correcto funcionamiento autónomo del sistema. Las tareas son las siguientes:

1. Reportes e Históricos.
2. Cálculos.
3. Comunicación.
4. Interfaz Hombre-Máquina.

En la primera de estas tareas (**Reportes e Históricos**) es conveniente recordar que existen tres tipos de reportes, que explicaremos a continuación:

El primero de ellos se relaciona con la entrada a la aplicación y que nos permite conocer qué usuario accedió, a qué hora y su nivel de acceso, así como la hora en que salió de la aplicación, configuración del ODBC para la conexión con la base de datos, inclusión de nuevos usuarios, eliminación de otros, conexión y desconexión con los Sistemas Supervisores Servidores. Esta parte de la tarea tiene carácter calendario, se ejecuta constantemente, incluyendo cambios de turnos, fines de semana o de mes. En este caso se va elaborando un reporte por cada día de jornal, pero una característica que incluye nuestro software y que además puede ser tomada como una ventaja, es que el reporte puede hacerse a solicitud del operador en cualquier momento del día. Todos estos reportes e históricos se guardan en un fichero “.txt” cuyo nombre esta conformado por el día, mes y año precedido por una S de supervisor, es decir, **Sddmmaa.txt**, este fichero se crea dentro de una carpeta ubicada en el mismo directorio donde se encuentra el software. Una vez guardados estos reportes, pueden ser impresos o almacenados en formato digital y ser utilizados para un posterior análisis estadístico del funcionamiento del Supervisor Cliente.

En el reporte que se hace desde el software Microsoft SQL Server se aprecia una de las mejoras que incluye esta aplicación, ya que mediante el mismo podemos tener

conocimiento del comportamiento, al finalizar la jornada de 8 horas, de todas las variables supervisadas. Además, este software nos da la oportunidad de poder imprimir este reporte al finalizar el trabajo en cada turno. Con el SQL Server se pueden realizar cambios en cuanto al tiempo de extensión del mismo.

La última parte de esta tarea consiste en la inclusión del Graficador de Estado, que nos permite conocer el comportamiento de todas las variables supervisadas de una en una. Esta es otra de las mejoras de nuestra aplicación, ya que el valor adquirido para cada variable es guardado en la base de datos, conjuntamente con la hora y fecha exactas de su adquisición, elementos que son mostrados ahora en nuestro graficador y que no estaba contemplado en la versión anterior. Esto nos da la posibilidad de que el usuario del sistema tenga conocimiento del valor de las variables desde cualquier momento anterior a su acceso.

Existen variables que no se obtienen directamente de los supervisores que están conectados a determinado proceso. Como la tarea de **Cálculo** resolvía esta problemática decidimos mantenerla en nuestra aplicación, de esta forma se realizarán los cálculos de eficiencia, nuevos períodos de muestreos, tiempo que dura una conexión, redefinición de límites para las alarmas y otros cálculos estadísticos que se implementan en función del área donde se emplee nuestra aplicación.

Continúa siendo la (**Comunicación**) una de las tareas más importante. Como antes, por medio de ésta se establecerá la comunicación a todos los niveles de la aplicación. Es debido a ella que nos encargaremos de configurar el ODBC para la conexión con la base de datos del sistema (**BaseDatosSupClient.dbo**), mantendremos la comunicación con dicha base de datos, transmitiremos la información de forma constante desde el sistema “**SupClien 1.1**” hasta la base de datos y viceversa; por último, y con la misma importancia, estableceremos la comunicación desde nuestro supervisor cliente con los supervisores servidores que estarán enviando los diferentes datos. Cada una de estas funciones la iremos dando a conocer en epígrafes posteriores.

En nuestro trabajo, mantiene relevante importancia la tarea de **Interfaz Hombre-Máquina** en lo que se refiere a su relación con el proceso de supervisión, por lo que decidimos realizar pocos cambios, por ejemplo, en el color que se presentan las ventanas al usuario, todo lo demás tiene un óptimo funcionamiento. En ella se conforman dos

funciones que pueden ser ejecutadas de forma simultánea: el refrescamiento de pantalla y la atención al operador. Con la primera el supervisor puede mantener constantemente actualizada la información que se muestra en las pantallas, para que llegue de la mejor forma y fiablemente al operador que lo maneja; en la segunda se lleva a cabo toda la atención del teclado y el mouse, se encarga de pasar la información introducida por el operador a las diferentes tareas y de mandar a ejecutar otras como la de **Cálculo**, la de **Reportes e Históricos** o la de **Comunicación**.

Al igual que antes todos los VI, tanto el principal como los secundarios, no tienen activo el botón (Cerrar), en su lugar cada ventana posee un botón (Cerrar o Cancelar), que se encarga de parar la ejecución, cerrar cada VI y liberar la memoria que estaba siendo usada por él.

## 2.2 Pantalla Principal

En una pantalla se muestran los elementos generales, de esta forma el operador puede ver toda la información, y accede a pantallas secundarias o particulares del sistema. En el **Anexo 1** hemos mostrado un gráfico donde se puede apreciar la estructura de conexión existente entre las mismas.

La pantalla o ventana Principal en modo de ejecución contiene una barra de Título donde se muestra el nombre de la aplicación, una barra de estado que indica quién está operando el supervisor, así como su categoría o nivel de privilegio, fecha y hora exactas, también una barra de menú que nos permite acceder a las diferentes pantallas secundarias donde se desarrolla y se muestra el gran peso operacional de este supervisor.

En nuestro estudio hemos considerado conveniente mantener un diagrama de bloques el que se compone por una estructura de secuencia conformada en cuatro pasos. El primero, “Inicio”, comprende la inicialización de la variable global “**Global SupClien**” donde se guarda la dirección en que podemos encontrar cada uno de los VI que contiene nuestra aplicación. “Desactivación del Menú” es el segundo paso, en él se consideran las opciones del menú que son desactivadas hasta tanto se registre un usuario. El Tercero, “Aplicación”, depende de las acciones que quiera desarrollar el operador por medio de las opciones del menú (mediante un “*Case*” que tiene programado en función de la opción solicitada, las llamadas a un VI u otro), además refresca la pantalla. Esta espera la realiza mediante un

ciclo “*While*” que solo puede ser interrumpido con la opción “Salir” del menú principal. En el cuarto paso de la secuencia, “Finalización”, es donde se pone en falso la opción “Salir” para que esté desactivada cuando se ejecute nuevamente la aplicación.

### 2.3 Pantallas Secundarias

Acorde a algunos estudios científicos realizados sobre la comunicación hombre máquina, las pantallas secundarias de este sistema, al igual que la principal, presentan una barra de título para identificar la ventana o VI que está activo; para hacer más fácil la programación el nombre de la pantalla no coincide con el nombre del VI, puesto que el de este último se tomó de forma que fuera más familiar al programador. Mantuvimos la política de diseño en lo que se refiere a que cada una de las ventanas tenga el mismo color de fondo que la ventana principal, de esta forma se evitan molestias al operador con el cambio permanente de colores de fondos en las diferentes pantallas. A continuación presentamos las diferentes opciones del Menú Principal que permiten acceder a cada una de las Pantallas Secundarias.

### 2.4 Menú principal

Como se había mencionado anteriormente, en la pantalla principal encontramos el menú que nos permite acceder a las diferentes funciones y ventanas de este sistema.

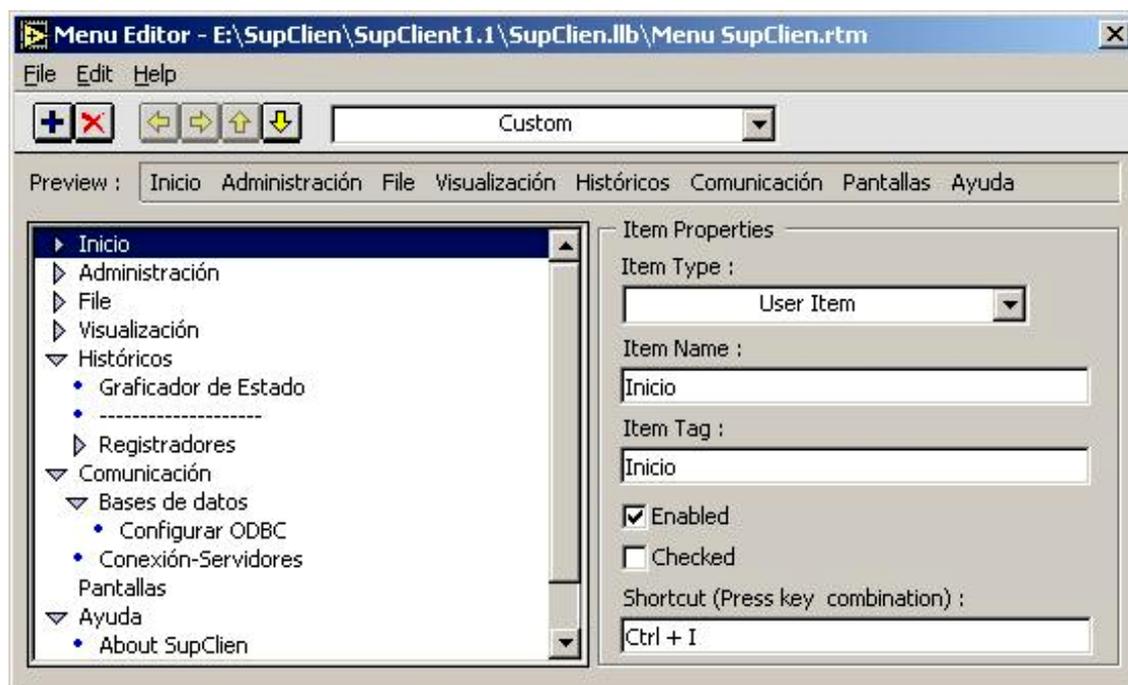
El proceso de confección del menú de nuestra aplicación se muestra en la **figura 2.1**, donde se pueden apreciar las opciones que para dicho menú se tuvieron en cuenta:

Cada una de estas opciones de menú llevan incluidos otros submenús o **User Item**<sup>8</sup>, que se desprenden de ellas y que son los que realmente dan paso a la ejecución de una u otra operación en el *Supervisor Cliente* “**SupClie n 1.1**”.

Decidimos mantener la política administrativa existente en la versión anterior de crear 4 niveles de privilegios (desde el 0 hasta el 3) para los usuarios del sistema, de esta forma tendrán acceso a una u otras de las opciones que brinda el menú en dependencia del nivel administrativo que este posea. De igual forma se consideran los mismos usuarios con sus correspondientes niveles de privilegios acordados y las opciones permitidas para cada uno.

---

<sup>8</sup> Denominación dada en el LabVIEW a las opciones de menú.



**Figura 2.1:** Confección del Menú Principal del Sistemas.

### 2.4.1 Menú “Inicio”

En cuanto se ejecuta el sistema aparece activado el menú Inicio conjuntamente con el de Ayuda, debido a que el mismo contiene la opción “Log In” que permite autenticar la entrada de un operador a éste. Con esta acción se da paso a su funcionamiento y se activan a su vez las restantes opciones del menú principal, de acuerdo al nivel de privilegios que posea el usuario que acceda.

En esta acción se despliegan dos opciones más: “Log off” y “Salir” que son las que permiten cerrar una sesión de trabajo y abandonar completamente el sistema.

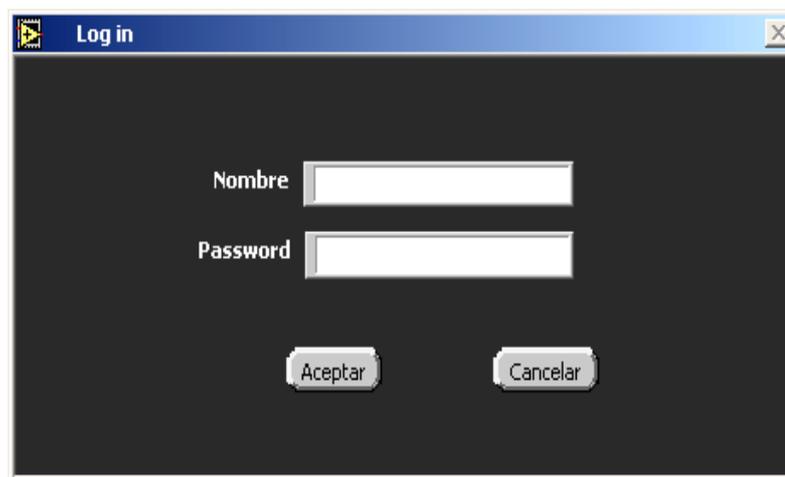
#### ➤ **Entrada al Sistema (Log in)**

En esta ventana se realiza la autenticación del usuario que vaya a entrar a manipular el sistema supervisor, la cual es comparada con un registro de usuarios que se encuentra en la base de datos del sistema, el VI **dialogpass1.1.vi** es el encargado de comparar el nombre y contraseña entrado con los que se encuentran registrados en la base de datos, si no hay ninguna coincidencia, es decir que no exista ese usuario, da la oportunidad de volverse a registrar, y si hay coincidencia se permite el acceso al sistema. Ahora en función del nivel

de acceso de ese usuario, el núcleo central de la aplicación le habilita ciertas y determinadas opciones del menú según le correspondan.

Esta acción constituye una de las mejoras que aporta nuestra versión, que a diferencia de la versión anterior, el registro de usuarios y sus contraseñas, se encontraban almacenados en una variable global. Con esto logramos acelerar y optimizar el funcionamiento del software en cuanto a lo que a administración se refiere, además de ganar en seguridad. Para ello, como se muestra en la **figura 2.2**, se colocó un **Digital Control**<sup>9</sup> y un **String Control**<sup>10</sup> donde se introduce el nombre y la contraseña del usuario; además contiene dos botones para aceptar o cancelar la acción.

Dentro del menú Inicio también tenemos las opciones para cerrar la sesión de trabajo: “Log off”, y para salir de la aplicación: “Salir”. La primera opción no llama específicamente a una ventana de la aplicación sino que hace uso de la herramienta **Two Button Dialog** del LabVIEW, la cual tiene tres entradas una para introducir el mensaje que se quiere mostrar, en este caso es: ¿Seguro que desea cerrar la sesión?; y las otras dos para definir el nombre que tendrán los botones de selección, en el caso que nos compete se usó: Si y No. Después de haber cerrado la sesión ya se está en condiciones de abandonar la aplicación por medio de la segunda opción mencionada.



**Figura 2.2:** Ventana de autenticación de usuario (dialogpass1.1.VI).

---

<sup>9</sup> Es un control en LabVIEW que permiten dar entradas a variables o datos que tengan valores numéricos.

<sup>10</sup> Es un control en LabVIEW que permiten dar entradas a variables o datos que sea cadenas de caracteres, este tipo de control admite números y letras.

### 2.4.2 Menú “Administración”

El menú Administración también se despliega en tres opciones: “Incluir Usuario”, “Eliminar Usuario” y “Lista de Usuarios” que ejecutan ciertas acciones.

#### ➤ “Incluir Usuario”

Registrar nuevos usuarios es un servicio que solo le está permitido a los operadores del sistema que estén en el grupo de los administradores o de la gerencia, ya que solamente para los niveles de privilegio 1 y 0 están activas estas opciones del menú de Administración. A diferencia de la aplicación anterior los usuarios son incluidos en la base de datos del sistema.

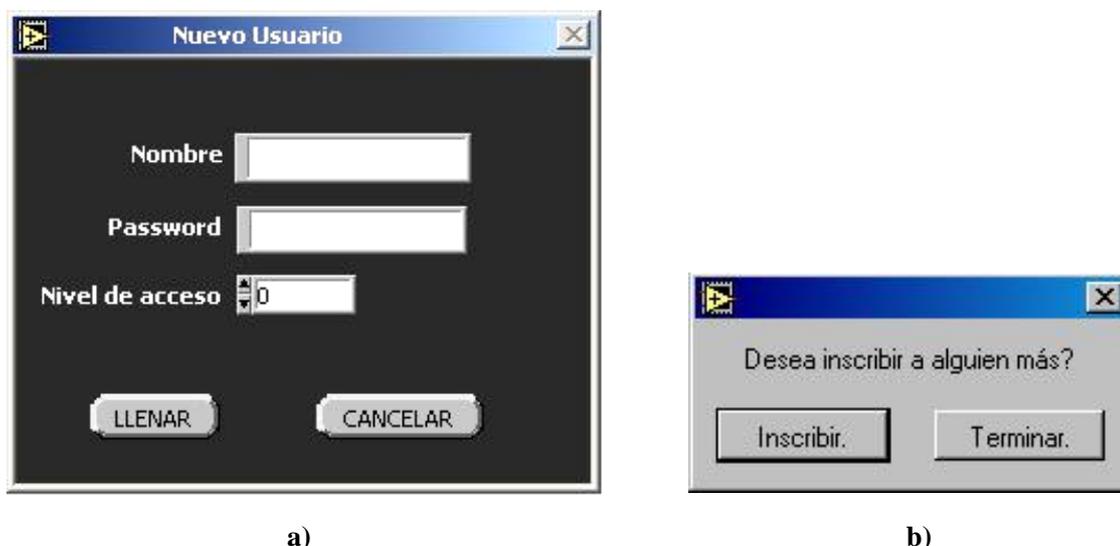
En la **figura 2.3 a)**, se observa la ventana para incluir un nuevo usuario y en la **figura 2.3 b)** la ventana que nos da la posibilidad de terminar o inscribir a otro usuario.

#### ➤ “Eliminar Usuario”

Entendemos conveniente mantener las opciones que contiene el menú “Administración” del anterior sistema sobre la opción de Eliminar usuario, con la diferencia de que ahora estos son eliminados, conjuntamente con sus contraseñas y niveles de privilegio de la base de datos existente y para acceder a ella hay que poseer los mismos niveles de privilegios mencionados anteriormente para la opción de “Incluir Usuario”. Este VI se llamó **borrarusuario1.1.vi** y se mantuvo su forma de operación utilizando el control del LabVIEW llamado Listbox, el cual consiste en una tabla tipo columna que se llena accediendo a la base de datos donde se encuentra el registro de usuarios y se muestra como una ventana de diálogo en la cual se selecciona el usuario a eliminar mediante un índice numérico que esta tabla asocia a cada una de sus filas, el usuario correspondiente a ese índice es borrado.

#### ➤ “Lista de Usuarios”

Esta es otra de las opciones del Menú “Administración” y nos permite tener conocimiento de todos los usuarios, así como sus niveles de privilegio. El VI que llama a esta aplicación es **passver1.1.vi**.



**Figura 2.3:** a) Ventana para introducir un nuevo Usuario.

b) Para Terminar o inscribir otro usuario.

### 2.4.3 Menú “File”

En este menú encontramos las opciones: “Ejecutar” y “Calculadora”. Incluimos el mismo para proporcionarles a los usuarios con niveles de privilegios del 0 al 2 una mayor facilidad a la hora de trabajar sobre el “SupClien 1.1”.

#### ➤ “Ejecutar”

Esta opción realiza una llamada al VI **Ejecutar.vi** el cual, después de haber seleccionado determinado fichero, lo ejecuta mediante la función **System exec.vi**. Toda esta funcionalidad es posible por medio del botón ‘Examinar’ que permite acceder a los directorios de la máquina o en la red y seleccionar el fichero a ejecutar.

#### ➤ “Calculadora”

Al igual que la versión anterior la segunda de las opciones que corresponde al menú “File” es la de la “Calculadora”; esta no hace más que ejecutar el **componente Active x**<sup>11</sup>

---

<sup>11</sup> Unidad de código ejecutable como un archivo .exe o .dll.

de Windows **Calc.exe**, haciendo uso de las potencialidades del **Active x**<sup>12</sup>, para esto también se emplea la función **System exec.vi**.

#### 2.4.4 Menú “Visualización”

El menú de Visualización contiene 3 opciones, las cuales son de gran importancia y le brinda una notable relevancia a cualquier sistema supervisor que las contenga; estas son: “Mímicos”, “Seguimiento por filas” y “Seguimiento por columnas”.

##### ➤ “Mímicos”

Esta opción permitirá al usuario acceder a los sinópticos de cada una de las áreas de la industria o a la empresa en general de acuerdo a sus intereses. Estos sinópticos son diagramas animados que reflejan los cambios en las variables claves de los procesos que se están supervisando. De acuerdo a su uso el operador tendrá conocimiento del comportamiento del mismo. Nuestra aplicación brinda la posibilidad de hacerle cambios a los mímicos existentes o incluir otros nuevos de forma dinámica. Mientras el “**SupClien 1.1**” esté en modo de ejecución, siempre que se presione la opción de “Mímicos”, la ventana se abrirá con la lista de estos, incluyendo los nuevos y las actualizaciones realizadas a los ya existentes. Al igual que en la versión anterior la nuestra permite la actualización dinámica de sus sinópticos.

Para lograr esto se creó una Carpeta en el mismo directorio de la aplicación en la cual están contenidos los esquemas de las diferentes áreas de la industria. Esto no origina ningún problema adicional, debido a que estos sinópticos ya deben estar creados para el Sistema Supervisor Servidor que está instalado en cada área en específico o en la planta en general. Además en la pantalla de “Selección de Mímicos” se colocó un **Listbox** en el cual se carga un listado con los sinópticos que se encuentran en dicha carpeta y se selecciona el esquema que se desea para posteriormente proceder a ejecutarlo, con solo presionar el botón ‘ABRIR’. La **figura 2.4** muestra la pantalla o VI en el que se lleva a cabo esta operación.

---

<sup>12</sup> Permite a los programadores ensamblar componentes de software reutilizable de una aplicación en otra.



**Figura 2.4:** Pantalla para la selección dinámica de sinópticos.

➤ **“Seguimiento por fila” y “Seguimiento por columna”**

Para tener una visualización tabulada del estado de las variables principales del proceso se crean el “Seguimiento por fila” y “Seguimiento por columna”, mecanismo que permitirá al usuario que opere el sistema detectar fácilmente el más mínimo cambio en cada una de las variables. En el seguimiento por fila se muestra la hora, los nombres de las variables, su descripción y su valor, mientras que en el seguimiento por columna solo se muestra la hora y el valor de la variable. Además se utilizó un **Menu Ring**<sup>13</sup> en cada fila o en cada columna dependiendo del tipo de seguimiento, con el cual se elaboró un pequeño menú local que le permite al operador seleccionar dentro de una lista de variables las que se quieren visualizar, esta, sin dudas, es otra de las ventajas que posee esta aplicación.

La actualización de la información a mostrar en cada variable es obtenida de la base de datos del sistema.

#### **2.4.5 Menú “Históricos”**

Este menú fue confeccionado específicamente para acceder a los diferentes registros de históricos. En él encontramos dos opciones: “Graficador de Estado” y “Registadores”; de

---

<sup>13</sup> Control del LabVIEW que permite hacer menú de despliegues para la selección de una variable dentro de una lista de ellas.

este último se derivan nuevas opciones de registradores en función de cómo quiera ser mostrada la información por el cliente.

Estas opciones realizan la llamada a dos ventanas (que también son pantallas secundarias de la aplicación), donde se muestra el comportamiento de cada variables en función del tiempo.

➤ **“Graficador de Estado”**

El VI que se encarga de los históricos del sistema muestra el estado de las variables una por una (o sea, la variable específica que desee el operador) durante la jornada de 8 horas, después de este tiempo la base de datos del sistema donde se encuentran las variables es refrescada, no sin antes crear un archivo **.txt** donde se encuentran los valores de las mismas durante ese turno, conjuntamente con la hora exacta en la que fue adquirido ese valor. Existe un fichero por día, que es lo mismo decir por jornada de trabajo del Sistema **“SupClien 1.1”**. En el **Anexo 2** se muestra un fragmento de uno de estos ficheros registrados en memoria. En este VI se hizo uso también de los **Menu Ring** para seleccionar la variable que se quiere visualizar. La visualización se realiza en un indicador tipo graficador **XYGraph**.

➤ **“Registrador Múltiple”**

El VI registrador múltiple tiene el mismo principio de funcionamiento que el VI anterior y la interfaz hombre máquina es bastante parecida, pero este opera con la diferencia de que en él se visualiza el estado de todas las variables. En el eje del tiempo del graficador **XYGraph** que se utiliza en esta pantalla se va refrescando automáticamente la fecha. Estos datos los adquiere mediante funciones de tiempo que posee el LabVIEW.

#### **2.4.6 Menú “Comunicación”**

Como se había planteado en el **epígrafe 2.1** la tarea de comunicación tiene a su cargo: realizar de forma eficiente el intercambio de información dentro del sistema, y la conexión del Sistema **“SupClien 1.1”** con los otros supervisores servidores. Para esto, el menú de

“Comunicación” engloba dos de las opciones más importantes de nuestro sistema, ellas son: “Configurar ODBC<sup>14</sup>” y “Conexión- Servidores”.

➤ **“Configurar ODBC”**

Cualquier sistema de Supervisión en Tiempo Real necesita de una herramienta que le permita realizar la conexión a las bases de datos de forma dinámica. También es conocido que desde cualquier aplicación, para poder acceder a una base de datos ya sea en Access, Excel, SQL Server, o cualquier base de datos profesional, primero es necesario configurar el ODBC debido a que el mismo es quien contiene los drivers que permiten el entendimiento de la aplicación con dichas bases de datos. En nuestro caso la comunicación es con el software SQL Server, el cual es un gestor de base de datos profesional de gran potencia que nos permite realizar algunas tareas que en la anterior versión no eran posibles o muy difíciles de ejecutar. Para realizar esta operación se programó el siguiente VI que se muestra en la **figura 2.5**. De esta forma, el operador solo tiene que definir el **data Source Name**, es decir, el nombre que se le quiera dar a la conexión, la **descripción** y buscar por medio del botón “Seleccionar” la **base de datos** a la cual se quiere acceder. En los tres casos se hizo uso de los **String Control**. Más adelante se explicará la estructura de dicha base de datos.



**Figura 2.5:** Pantalla para configurar el ODBC.

---

<sup>14</sup> Open DataBase Connectivity. Establece la conexión del software con las Bases de Datos.

### ➤ “Conexión Servidores”

Esta continúa siendo la opción principal en la comunicación de nuestro “**SupClien 1.1**” por medio de la cual se hace posible el enlace de este con los sistemas supervisores servidores que se encuentren en la red. Esta opción realiza un llamado a la plataforma “*CliSerV 1.0*” desarrollada por el MsC. Robby Gustabello Cogle en su trabajo de maestría, una vez conectada la aplicación a los servidores, se guarda en la base de datos el estado de las variables que se estén transmitiendo. De esta forma el sistema adquiere carácter de arquitectura abierta, debido a que permite la búsqueda de estas variables desde diferentes servidores a través de varios protocolos de comunicación: OPC, DDE, DataSocket y TCP/IP. Es decir, con el empleo de esta plataforma se obtuvo un resultado novedoso en la medida que se hizo posible conectar nuestra aplicación a servidores con diferentes protocolos de comunicación.

#### 2.4.7 Menú “Pantallas”

Esta opción fue creada para cargar dinámicamente el nombre de las pantallas que estén activas en la aplicación, primeramente aparece vacía, haciendo entonces al menú “Pantallas” un menú desplegable, donde aparecen estos nombres que sirven para maximizar las ventanas que están minimizadas.

Para esto el VI central del “**SupClien 1.1**” accede a la variable global “GlobalLista” donde se encuentra almacenada en un arreglo “Lista” el nombre de todas las ventanas activas, para posteriormente con la función “**Insert menú Items**” adicionarla al menú Pantallas del Menú principal.

#### 2.4.8 Menú “Ayuda”

En este menú se despliegan dos opciones a nuestro entender muy importantes para el correcto funcionamiento de “**SupClien 1.1**”, ellas son: “About SupClien, (Ctrl + ?)” y Versión (Ctrl + Shift + V).

### ➤ “About SupClien”

Al seleccionarlo se abre y ejecuta el programa del mismo nombre “**About SupClien.vi**”, que se carga en memoria desde el VI principal. Con ella se muestra en copia magnética un documento elaborado como manual de usuario, que permite a estos

familiarizarse rápidamente con la aplicación y servir de apoyo o herramienta de consulta para todos aquellos nuevos o antiguos usuarios que empleen el supervisor. En este manual se explica detalladamente la forma de operar cada una de las opciones del menú.

En el diagrama de bloques puede apreciarse que la apertura del documento “*Ayuda*” se realiza por medio de un objeto “**Active-x Automation**”, al cual se le configuró la propiedad “*Visible*” como verdadera y de la propiedad “*Documents*”, se invocan los métodos “*Open*”, “*FileName*” y “*Read Only*”.

#### ➤ “**Versión**”

Esta opción ejecuta el llamado a un pequeño VI informativo el cual muestra solamente los aspectos generales del *Sistema Supervisor Cliente “SupClien 1.1”*. Ver **Anexo 3**

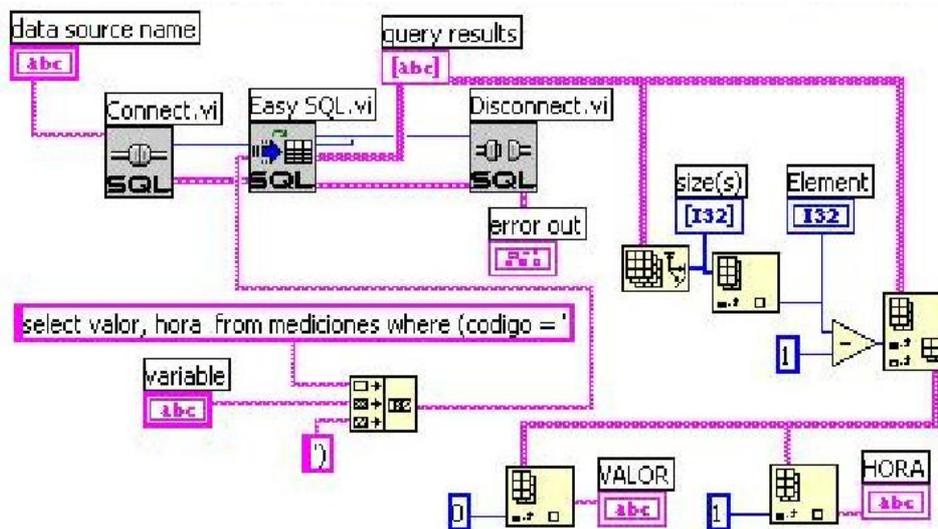
A continuación se realizará una descripción de la estructura de la base de datos **BaseDatosSupClient.dbo**.

## 2.5 Base de Datos

El almacenamiento de los datos obtenidos de las variables desde los servidores es fundamental, ya que es necesario mostrarlos en los históricos, registradores y para la futura toma de decisiones por parte de los directores de las áreas donde sea utilizado el “**SupClien 1.1**”. Para poder realizar estas acciones es primordial la conexión con el supervisor servidor, partiendo de la premisa que se mencionó en epígrafes anteriores de que antes de conectar cliente con servidor, debemos saber que existe tal servidor que suministra Xs datos; luego se hace uso de las sentencias de SQL que nos permiten escribir o leer datos en y desde una base de datos. La base de datos para esta versión se creó en Microsoft SQL Server, debido a la potencialidad y eficiencia de este programa y por ser una herramienta donde se crean las bases de datos con interfaz gráfica sencilla y fácil de manipular. Además de las opciones que nos brinda de programación de tareas, etc.

No en todos los casos son necesarios la totalidad de los datos adquiridos, por lo que una vez guardados estos datos se utilizan las funciones del Toolkit de SQL del LabVIEW (Connect.vi, Execute SQL.vi, Fetch Query Results.vi, Disconnect.vi), para seleccionar aquellos que sean de nuestro interés, lo mismo para mostrarlos o para elaborar los reportes

e históricos que serán utilizados en análisis estadísticos por los directivos de la industria o empresa.



**Figura 2.6:** Estructura SQL para la extracción de datos.

Con el objetivo de establecer el nexo entre la aplicación y la base de datos se hace necesario una secuencia específica **figura 2.6** que incluye: un elemento para la conexión con la base de datos, otro para introducir el comando de encuesta y un tercero para realizar la desconexión una vez finalizada la encuesta o introducción de los valores.

### Consideraciones finales del capítulo

Hasta aquí hemos expuesto todo lo referente a la estructura y funcionalidad del sistema “SupClien 1.1”, y a partir de nuestro estudio hemos llegado a conclusiones que reflejaremos a continuación:

- o La funcionalidad del sistema y su operatividad a la hora de ejecutar las llamadas a las diferentes “Pantallas Secundarias” se optimizan por el uso del Menú Principal en forma de barra.
- o La aplicación se hace más eficiente en lo que se refiere a tiempo de procesamiento por el uso de una sola pantalla principal, desde la cual accedemos a las restantes.
- o Se logra disminuir el tiempo de respuesta de la aplicación con la inicialización de los VI desde la primera secuencia del VI principal.

- o El uso de los reportes del funcionamiento del Sistema es un factor de suma importancia, y en esta versión se ven mejorados debido al empleo del gestor de base de datos SQL Server.
- o El uso del gestor de base de datos SQL Server proporciona a la aplicación gran flexibilidad en el manejo de los datos, ya que este nos brinda opciones de programación de tareas, lo que nos permite llevar fácilmente un Histórico del estado de las variables.

Después de analizar estos detalles estamos en condiciones de realizar las pruebas correspondientes para probar el correcto funcionamiento del “**SupClien 1.1**”, las cuales son plasmadas en el **Capítulo 3** del informe.

## CAPÍTULO 3. Pruebas, Análisis y Resultados

Después de haber analizado la estructura general, utilidad e importancia del Sistema, pasaremos a analizar los resultados obtenidos en las pruebas realizadas.

Debido a la magnitud de la aplicación nos vimos en la necesidad de realizar pruebas parciales, con el objetivo de comprobar el correcto funcionamiento de la misma, posteriormente, al finalizar su programación se pasó entonces a la ejecución de las pruebas finales. Precisamente a la descripción de todas estas pruebas y al análisis de sus resultados, es que está dedicado este capítulo.

### 3.1 Pruebas y Resultados

En lo referente a la parte de administración (formada por el menú “Inicio” y “Administración”) se creó la base de datos del sistema para almacenar los usuarios del mismo y se definieron un conjunto de estos con diferentes niveles de acceso y así verificar si se hacía la selección correcta del menú en función de este nivel, esto funcionó correctamente.

Se programó la aplicación para que, en caso de que los usuarios presionaran la opción “Salir” sin cerrar la sesión, se presentara una ventana de diálogo como la de la **figura 3.1**. Por tanto los usuarios deben cerrar la sesión antes de salir. Con esto se garantiza que en la opción “Log Off” se cierren todas las ventanas secundarias activas y además se regrese a la configuración del menú inicial que presenta el sistema, es decir, que quedaran activas solamente las opciones “Inicio” y “Ayuda”.



**Figura 3.1:** Ventana de alerta para cerrar una sesión.

Después de realizadas las funciones correspondientes a la selección dinámica de los mímicos, se procedió a su prueba. En la **figura 3.2** se muestra la ventana de selección de mímicos.



**Figura 3.2.** Selección de mímicos.

Se comprobó el funcionamiento de los reportes simplemente con el manejo de la aplicación y se observó la creación de los ficheros .txt correspondientes a cada día de trabajo, la tarea programada en el SQL también se ejecutó de forma óptima. En el **Anexo 4** se muestra un listado de estos ficheros.

Las opciones de “Seguimiento por Fila” y “Seguimiento por columna” se programaron con el objetivo de realizar un seguimiento tabulado del estado de las variables del sistema. Se creó en la base de datos una nueva tabla (**Variables**) con los siguientes campos: ‘Nombre de la Variable’, ‘Valor’, ‘Descripción’ y ‘Unidad de medida’. Esta base de datos sirvió para comprobar el acceso desde las diferentes pantallas del sistema y se actualiza con los datos que son adquiridos desde los servidores. En la **figura 3.3 a)** se muestra la ventana del “Seguimiento por Fila” y en **b)** la del “Seguimiento por Columna”.

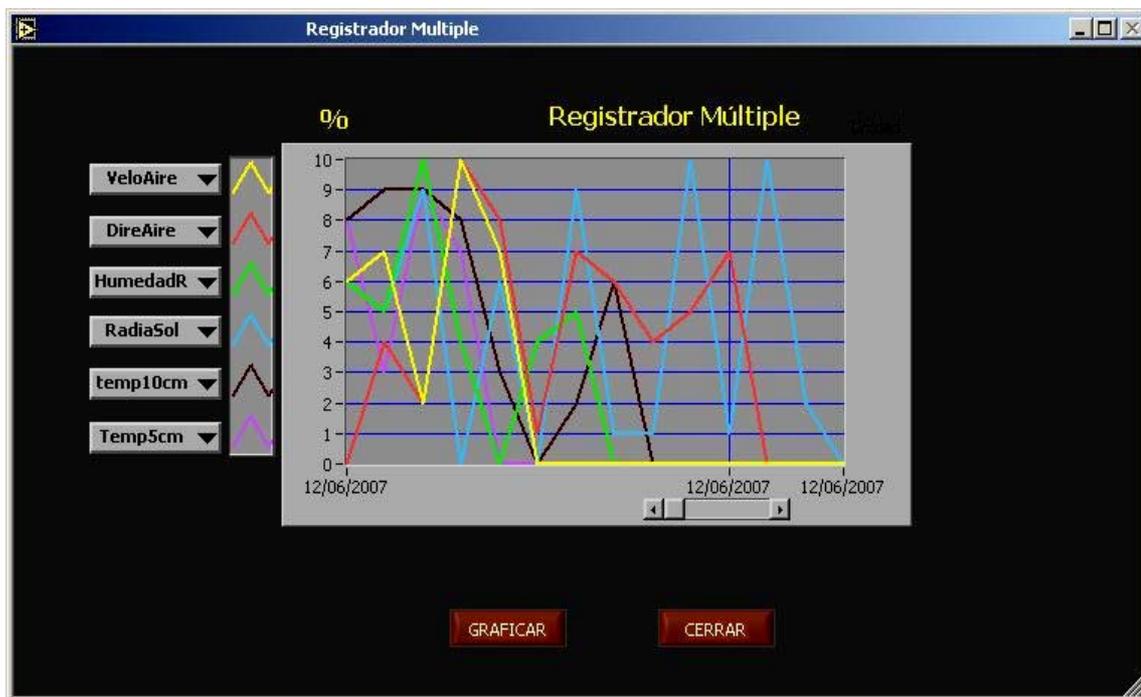
El “Registrador Múltiple” adquiere los valores desde la base de datos del sistema (**BaseDatosSupClient.dbo**) para que sean graficados. El resultado es mostrado en la **figura 3.4** donde podemos apreciar el funcionamiento de nuestro registrador, y la posibilidad que nos brinda de tener un conocimiento general del comportamiento de las variables durante el día de trabajo.

Variables	Hora	Descripción	Valor	Unidad
Precipit	16:14:05	Lluvia acumulada	1,00	mm/d
DireAire	16:12:55	Direccion del Aire	6,00	*
RadiaSol	16:13:55	Radiación solar	5,00	w/m2
VeloAire	16:07:17	Velocidad del aire	7,00	m/s
Precipit	16:14:05	Lluvia acumulada	1,00	mm/d
TempAire	16:13:15	Temperatura Ambiente	1,00	*C
TempSupf	16:13:45	Temperatura del suelo	3,00	*C
Tempa5cm	16:13:35	Temperatura a 5 cm	7,00	*C
Temp10cm	16:11:56	Temperatura a 10 cm	7,00	*C
PresionA	16:09:07	Presion Atmosférica	8,00	hPa
PresionA	16:09:07	Presion Atmosférica	8,00	hPa

Figura 3.3 a) Seguimiento por fila

Instante	VeloAire	DireAire	HumedadR	RadiaSol	Precipit	Precipit
11:26:19	7,00	6,00	2,00	5,00	1,00	1,00
11:26:20	7,00	6,00	2,00	5,00	1,00	1,00
11:26:20	7,00	6,00	2,00	5,00	1,00	1,00
11:26:21	7,00	6,00	2,00	5,00	1,00	1,00
11:26:22	7,00	6,00	2,00	5,00	1,00	1,00
11:26:23	7,00	6,00	2,00	5,00	1,00	1,00
11:26:23	7,00	6,00	2,00	5,00	1,00	1,00
11:26:24	7,00	6,00	2,00	5,00	1,00	1,00
11:26:25	7,00	6,00	2,00	5,00	1,00	1,00
11:26:25	7,00	6,00	2,00	5,00	1,00	1,00

Figura 3.3 b) Seguimiento por columna



**Figura 3.4** Registrador Múltiple

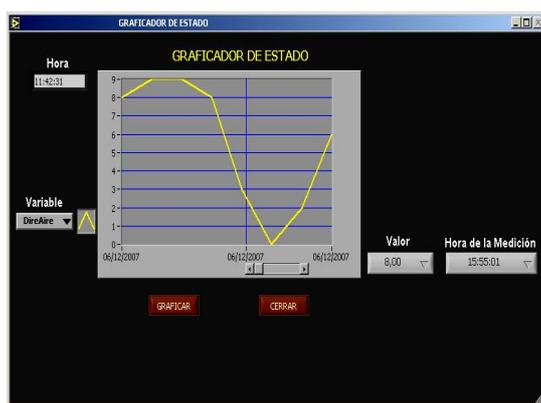
Al llegar a esta parte del trabajo ya nuestra versión de la aplicación estaba a punto y se había probado su funcionamiento por partes. Por tanto nos dispusimos a realizar las pruebas finales de la misma. Como se había explicado en el capítulo anterior se utilizó la plataforma “*CliSerV 1.0*” para la interconexión con los supervisores servidores.

Estas pruebas consistieron, en definir una serie de servidores desde los cuales se iban a adquirir los valores de las variables. Indistintamente estos servidores se ejecutaron y por medio de la plataforma “*CliSerV 1.0*” se estableció la conexión del Sistema Supervisor Cliente con dichos servidores. De esta forma los valores de las variables iban siendo guardados en la base de datos para posteriormente mostrarlos, ya sea en el “Graficador de Estado” o en el “Registrador Múltiple”. La **Tabla 3.2** contiene el listado de los servidores que se usaron así como las variables a las cuales se le actualizó el campo ‘Valor’.

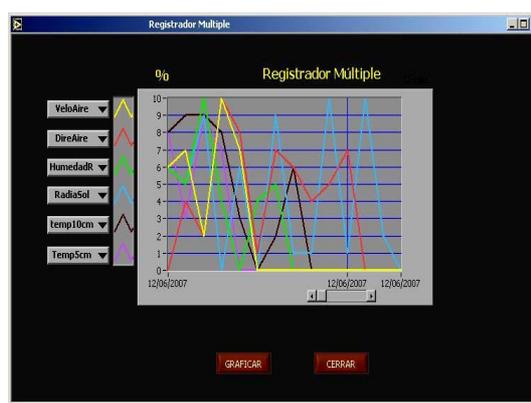
Número	Fuente del dato	Variable
1	Servidor DDE <i>KEPServerEx</i> , de una señal aleatoria.	VeloAire

2	Servidor DDE en LabVIEW, de una señal aleatoria.	DireAire
3	Servidor TCP/IP en LabVIEW, de una señal tipo aleatoria.	TempAire
4	Servidor DataSocket en LabVIEW, de una señal tipo aleatoria	HumedadR
5	Servidor <i>Matricon OPC Server for DDE</i> , que publica una señal aleatoria.	RadioSol
6	Servidor TCP/IP en LabVIEW, de una onda sinusoidal	Temp10cm

Se estableció la conexión con un servidor *DDE KEPServerEx* que publicaba una señal tipo aleatoria ('VeloAire'), la **figura 3.5 a)** describe el caso 1 de esta tabla y se observó como nuestro "Graficador de Estado" funcionó correctamente. Por otro lado en el **inciso b)** se puede ver el registro de cinco variables aleatorias 'DireAire', 'TempAire', 'HumedadR', 'RadioSol' y 'Temp10cm' procedentes de los servidores: *DDE en LabVIEW*, *TCP/IP en LabVIEW*, *DataSocket en LabVIEW*, *TCP/IP en LabVIEW* y *Matricon OPC Server*, correspondientes a los casos 2, 3, 4, 5 y 6 de la **Tabla 3.2**, respectivamente, al igual que en el primer caso, pero esta vez observando el "Registrador Múltiple" pudimos apreciar el funcionamiento óptimo del mismo.



**Figura 3.5 a)** "Graficador de Estado" mostrando la variable VeloAire.



**Figura 3.5 b)** "Registrador Múltiple" mostrando las variables obtenidas de los servidores.

**Análisis económico.**

Después de haber consultado toda la bibliografía referente a los sistemas supervisores y sus características, además de la estructura general y el funcionamiento de nuestra aplicación y haber hecho pruebas a nuestro software, podemos entonces hacer el análisis económico del mismo.

El SupClien 1.1 es una aplicación de muy bajo costo ya que para su implementación solo necesitamos de una computadora y enormes deseos de trabajar. Podemos decir también que su instalación en cualquier industria sería de gran beneficio para esta ya que sus requerimientos de máquina son muy viables, es decir, no se necesita de una súper computadora para su funcionamiento, sin dejar de mencionar lo que mejoraría dicha entidad en cuanto a producción y seguridad informática, debido a que los directivos tendrían un conocimiento total de lo que sucede en todas las áreas de su empresa.

**Consideraciones finales del capítulo**

El presente capítulo fue dedicado al análisis del correcto funcionamiento de nuestra versión: Sistema Supervisor Cliente “**SupClien 1.1**”. Se realizaron pruebas parciales durante su conformación con el objetivo de comprobar si la programación que se estaba realizando era correcta o no. Además se aplicaron las pruebas finales que sirvieron para constatar la aptitud del sistema.

Una vez comprobado que el sistema brinda un correcto funcionamiento, se está en condiciones de pasar a las conclusiones de la investigación.

## CONCLUSIONES

Con el presente trabajo se ha podido dar solución al problema científico que plantea la mejora de la versión anterior de nuestra aplicación, manteniendo sus características fundamentales de sistema de arquitectura abierta, con facilidad para la comunicación con otros sistemas (con diferentes protocolos).

Se revisaron temas afines con los sistemas supervisores, se acordaron los requisitos y objetivos fundamentales a llevar a cabo en la creación de esta versión.

Finalmente se logró una aplicación que cumple con los objetivos trazados inicialmente lo cual se probó a través de la realización de pruebas parciales y finales.

Después de un análisis exhaustivo se llegó a las siguientes conclusiones:

1. El **“SupClie 1.1”** permite a las empresas un desarrollo amplio gracias a sus características de arquitectura abierta.
2. Con la utilización de los mímicos el sistema adquiere una gran funcionalidad.
3. La utilización del Registrador Múltiple garantiza la supervisión del estado de todas las variables y un óptimo análisis a niveles directivos.
4. La creación en esta versión del **“Graficador de Estado”** es un aporte de gran importancia ya que este permite analizar cada variable por separado, conociéndose de esta su valor y la hora exacta a la que fue tomado el mismo, evitando cometer errores por lo engorroso de la visualización.
5. La utilización del Microsoft SQL Server es vital en el óptimo funcionamiento del software, ya que este le imprime mayor seguridad y velocidad a la hora de ejecutar las aplicaciones, además de sus facilidades para elaborar los reportes, los cuales pueden ser impresos y/o llevados a formato digital fácilmente para su posterior análisis.
6. Nuestro software y la bibliografía empleada pueden ser objeto de estudio en la carrera de Automática.

### RECOMENDACIONES

1. Comenzar a utilizar el sistema (en forma de prueba) en diferentes empresas de nuestra provincia y extender su uso a todo el país.
2. Incluir un servicio de Chat o de Correo para facilitar la interacción entre los directivos de las diferentes áreas de la empresa, en el proceso de toma de decisiones.
3. Utilizar nuestro software “**SupClien 1.1**” y toda la documentación utilizada en su elaboración como material de estudio de la asignatura Computación Aplicada, ya sea en el 5to año de la carrera de Automática o en asignaturas de Post-Grado.
4. Crear mímicos animados para el proceso de puesta a punto de la aplicación.
5. Se debería crear una nueva versión de esta aplicación en software libre (Linux) ya que la nuestra corre sobre Windows, lo cual impediría su posible comercialización hacia otros países.

**BIBLIOGRAFÍA**

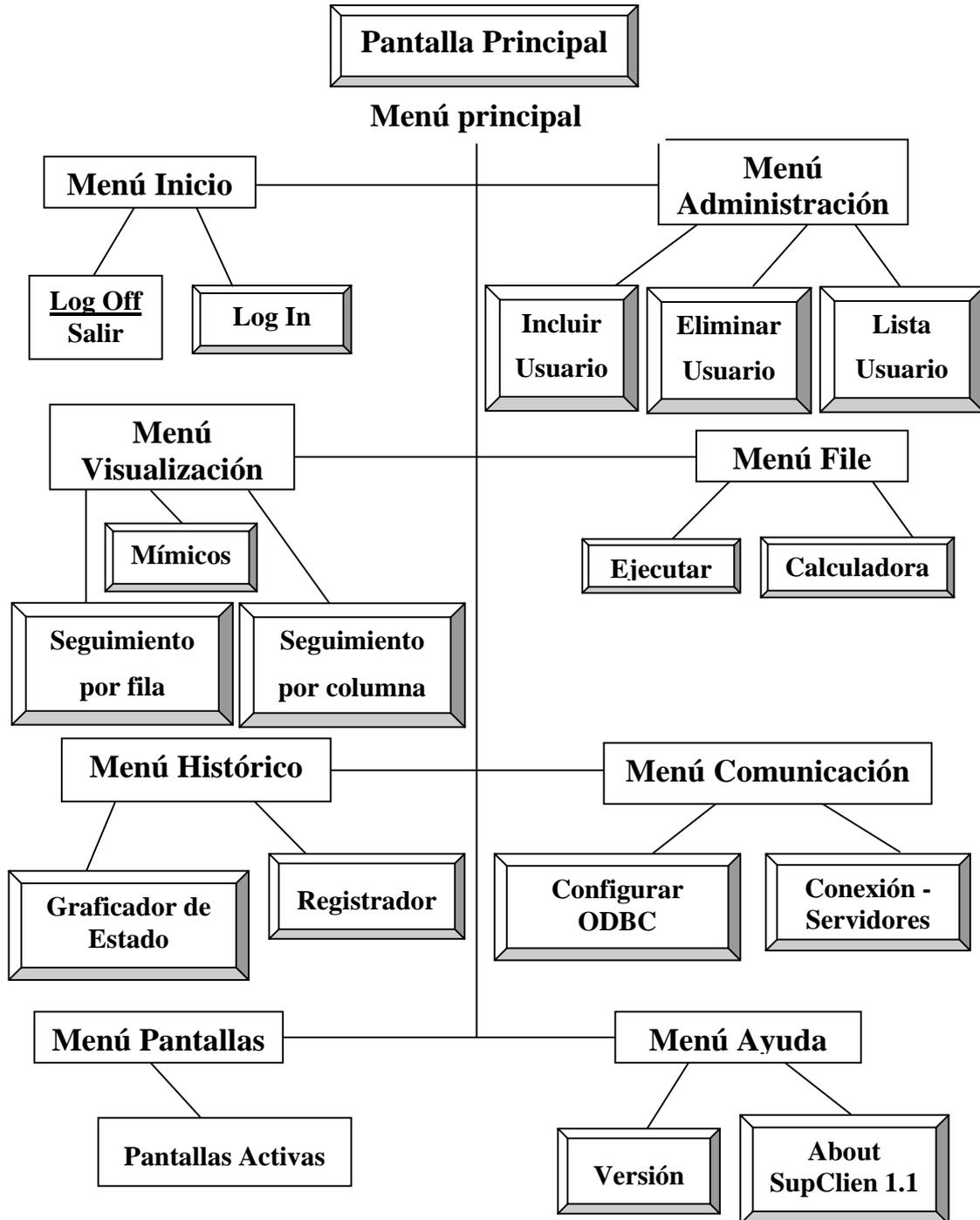
1. Adam K., 2005, Modeling of immersion lithography for OPC, From: <http://www.reed-electronics.com/semiconductor/article/CA501191> , Accedido en: 25-marzo-2007.
2. Balcells J., R.J., 1999, Introducción a SCADA, From: <http://eya.swin.net> , Accedido en: 2-febrero-2007.
3. Ballesteros R., Computación Aplicada. 1999.
4. Borges L., 2000, Sistemas de lectura remota del consumo, From:<http://neutron.ing.ucv.ve/revista/No7/Luis%20Borges%5CSISTEMAS%20DE%20LECTURA%20REMOTA%20DEL%20CONSUMO.htm> , Accedido en: 22-marzo-2007.
5. Caicedo O., 2004, Aplicaciones Web, Universidad del Cauca., From: [http://www.usb.edu.co/diplomado\\_aplicaciones\\_web.htm](http://www.usb.edu.co/diplomado_aplicaciones_web.htm), Accedido en: 15-marzo-2007.
6. Campos f., c.g., Rojas a., 2004, Sistema de control y supervisión industrial multiplataforma, From: <http://www.enlaceinformatico.unicauca.edu.co/docs/sistemacontrolsupervisionindustrialmultiplataforma.pdf> , Accedido en: 24-abril-2007.
7. Cervera J., 2002, Sistemas informáticos de control, From: <http://www.um.es/informatica/estudios/programas/ITIS/07BP.pdf>, Accedido en: 11-febrero-2007.
8. Colomer J., M.J., 2004, Introducción a la monitorización y supervisión experta de procesos. Métodos y herramientas., From: <http://eii.unex.es/academica/programas/ie/optativas%20microelectronica/Sistemas%20Supervision.pdf> , Accedido en: 18-marzo-2007.
9. Company, E., 2000, SCADA Ecofloat, From: <http://www.ecofloat.es/index.htm> , Accedido en: 15-marzo-2007.
10. Corporation., B.S., 2004, Guía del desarrollador de servicios Web. EEUU., From: <http://www.enlaceinformatico.unicauca.edu.co/docs/sistemacontrolsupervisionindustrialmultiplataforma.pdf> , Accedido en: 15-Marzo-2007.
11. Curso de SQL, 2001, From: [http://www.aulaalic.es/sql/f\\_sql.htm](http://www.aulaalic.es/sql/f_sql.htm), Accedido en: 22-Febrero-2007.

12. D'Sousa C., 2000, Sistemas de Control, From: [http://www.gtc.iac.es/control/Control\\_es.asp](http://www.gtc.iac.es/control/Control_es.asp) , Accedido en: 14-mayo-2007.
13. D'Sousa C., 2005, Sistemas scada, From: <http://www.automatas.org/redes/scadas.htm> , Accedido en: 22-abril-2007.
14. DVT introduces OPC Server for machine vision, 2004, From: <http://www.manufacturing.net/ctl/article/CA450381.html> , Accedido en: 15-Marzo-2007.
15. Eisenberg A., et al., 2004, SQL 2003, From: <http://www.sigmod.org/sigmod/record/issues/0403/E.JimAndrew-standard.pdf> , Accedido en: 23-Marzo-2007.
16. Estrategia de sistemas abiertos de la administración del estado, 1990, From: <http://www.csi.map.es/csi/pg6050.htm> , Accedido en: 15-Mayo-2007.
17. Fernández Lozano J., 2005, Supervisión y Control de Procesos, From: <http://www.etsii.uma.es/descargas/asignaturas/ia905.pdf> , Accedido en: 10-marzo-2007.
18. Ferreiro García R., 2003, Tecnologías actuales de comunicación de las variables de campo en la industria de proceso, From: [http://www.cea-ifac.es/actividades/jornadas/XXII/documentos/A\\_03\\_IC.pdf](http://www.cea-ifac.es/actividades/jornadas/XXII/documentos/A_03_IC.pdf) , Accedido en: 2-febrero-2007.
19. Figueras S., 1999, Memoria del proyecto de Ingeniería Técnica en Informática de Sistemas presentada. From: <http://personal.redestb.es/efigueras/memoria.htm> , Accedido en: 13-Marzo-2007.
20. Figueroa S., 2002, Modelos OSI y TCP/IP., From: <http://www.mygnet.com/articulos/redes/726/> , Accedido en: 18-abril-2007.
21. González M., 1997, Monografía SQL SERVER., From: <http://www.monografias.com/trabajos14/sqlserver/sqlserver.shtml> , Accedido en: 22-Febrero-2007.
22. Gustabello R., 2003, Servidores/Clientes para Redes Industriales con Ethernet en Tiempo Real. Tesis de Maestría. Cuba, Departamento de Automática y Sistemas Computacionales, Universidad Central “Marta Abreu” de Las Villas.
23. LabVIEW, User Manual, 1998, From: <http://www.ni.com/pdf/manuals/320999b.pdf> , Accedido en: 26-abril-2007.

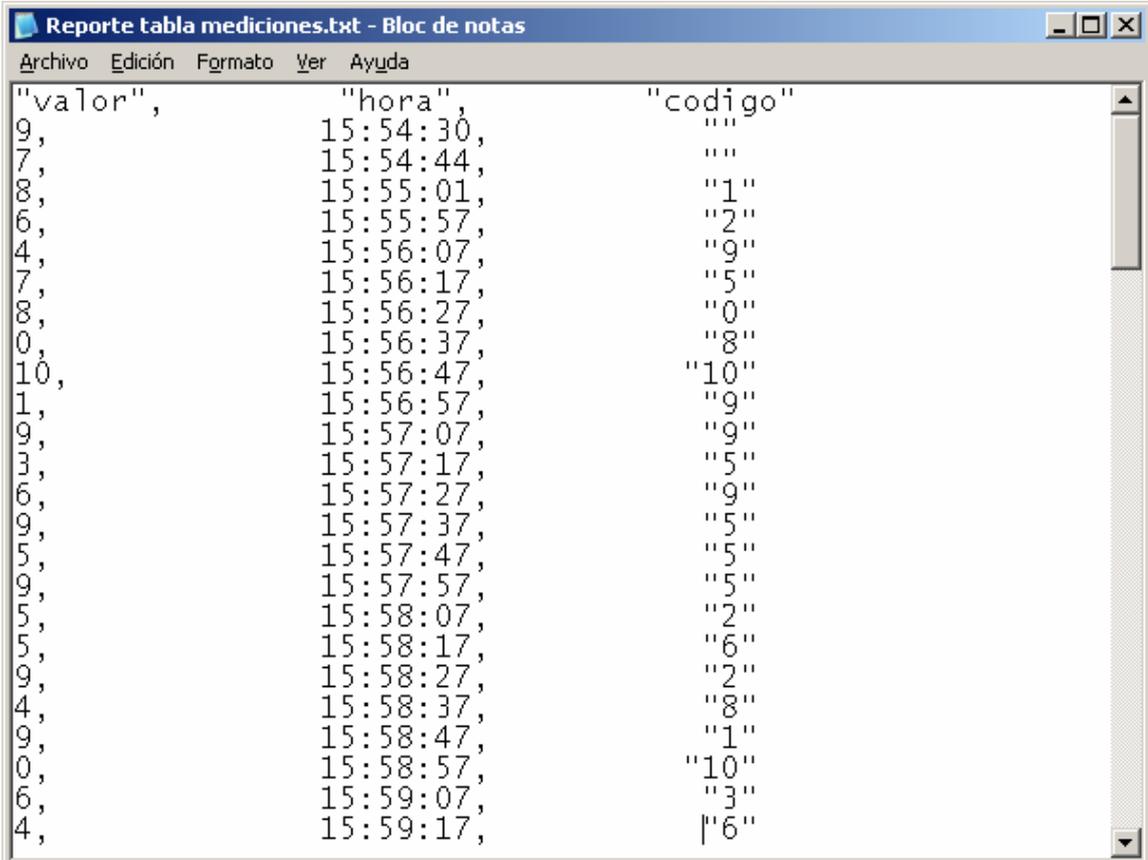
24. López D., 2002, Introducción a los SCADA, From: <http://www.isa.uniovi.es/felipe/files/infindII/documentos/scadas.pdf> , Accedido en: 13-marzo-2007.
25. Lozano C., R.M., Cristóbal., 2002, Introducción a SCADA, From: <http://www.santiagoapostol.net/srca/buses/scada.pdf>, Accedido en: 23-Febrero-2007.
26. Meléndez J., C.J., 2002, Supervisión de Procesos Industriales, From: [http://eia.udg.es/~quimmel/docencia/superv\\_proc\\_ind\\_iitap/super\\_proc\\_ind\\_iitap.html](http://eia.udg.es/~quimmel/docencia/superv_proc_ind_iitap/super_proc_ind_iitap.html) , Accedido en: 15-mayo-2007.
27. Melton, J., 2002, Ion Enterprise SCADA, From: <http://www.IonEnterpriseCompany/manualdereferencia/IonEnterpriseScada.html> , Accedido en: 20-marzo-2007.
28. Michael V., 2001, Quo vadis industrial Ethernet, From: <http://www.isa.org/.../ContentManagement/ContentDisplay.cfm&Content>, Accedido en: 22-Enero-2007.
29. Molineaux R., J.D., 2003, La Instrumentación Virtual, From: [http://www.aadeca.org/articulos/Tracnova-La\\_Instrumentacion\\_Virtual.pdf](http://www.aadeca.org/articulos/Tracnova-La_Instrumentacion_Virtual.pdf) , Accedido en: 16-febrero-2007.
30. OPC releases Data Exchange Specification, 2003, From: <http://groups.google.com/cu/groups?q=OPC+releases+Data+eXchange+specification&hl=es&lr=&ie=UTF-8&sa=X&oi=groups&ct=title>, Accedido en: 20-marzo-2007.
31. Padron J. O., 2003, Sistema Supervisor de Variables Energéticas. Trabajo de Diploma. Cuba, Departamento de Automática y Sistemas Computacionales, Universidad Central “Marta Abreu” de Las Villas.
32. Rosen J., 2000, El Modelo Cliente-Servidor., From: <http://sopa.dis.ulpgc.es/diplomatura/practicass9798/ipc/interc01.htm>, Accedido en: 25-Marzo-2007.
33. Samper Márquez J., 2005, Introducción a los Sistemas Expertos, From: <http://www.redcientifica.com/doc/doc199908210001.html> , Accedido en: 26-mayo-2007.

34. Sherstuk Y., 2004, Solutions on oil refinery information storage construction on the basis of MS SQL server, From: <http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/s/Sherstuk:Yuri.html> , Accedido en: 15-Febrero-2007.
35. Siles R., 2002, Análisis de seguridad de la familia de protocolos TCP/IP y sus servicios asociados, From: [http://ulises.adi.uam.es/pub/mirror/LuCAS/Manuales-LuCAS/doc-seguridad-tcpip/Seguridad\\_en\\_TCP-IP\\_Ed1.pdf](http://ulises.adi.uam.es/pub/mirror/LuCAS/Manuales-LuCAS/doc-seguridad-tcpip/Seguridad_en_TCP-IP_Ed1.pdf) , Accedido en: 20-marzo-2007.
36. Simbron N., 1997, Teoría General de Sistema, From: <http://www.monografias.com/trabajos/tgralsis/tgralsis.shtml> , Accedido en: 10-febrero-2007.
37. Soto M., 2001, Protocolos TCP/IP, From <http://www.monografias.com/trabajos/protocolotcpip/protocolotcpip.shtml>, Accedido en: 20-marzo-2007.
38. Stanek William R., 2000, Microsoft sql server 2000. manual del administrador., From: <http://www.agapea.com/Microsoft-SQL-Server-2000-Manual-del-administrador-n8980i.htm> , Accedido en: 15-febrero-2007.
39. Tutorial del LabVIEW (1999) From: [http://www.gte.us.es/~galvan/IE\\_4T/Tutorial%20de%20Labview.pdf](http://www.gte.us.es/~galvan/IE_4T/Tutorial%20de%20Labview.pdf), Accedido en: 30-marzo-2007.
40. Vegas J., 2002, Creación de un Portal Web docente, From: <http://www.abcdatos.com/webmasters/tutorial/15923.html> , Accedido en: 22-mayo-2007.
41. Vetter M., 2002, La especificación OPC, From: <http://www.opceurope.org> , Accedido en: 20-abril-2007.
42. Vidal P., 2002, OPC: Un estandar en las redes industriales y buses de campo, From: [http://www.senacitel.cl/senacitel2002\\_5.php](http://www.senacitel.cl/senacitel2002_5.php), Accedido en: 12-marzo-2007.

## Anexo 1: Estructura de conexión entre las pantallas del SupClien 1.1

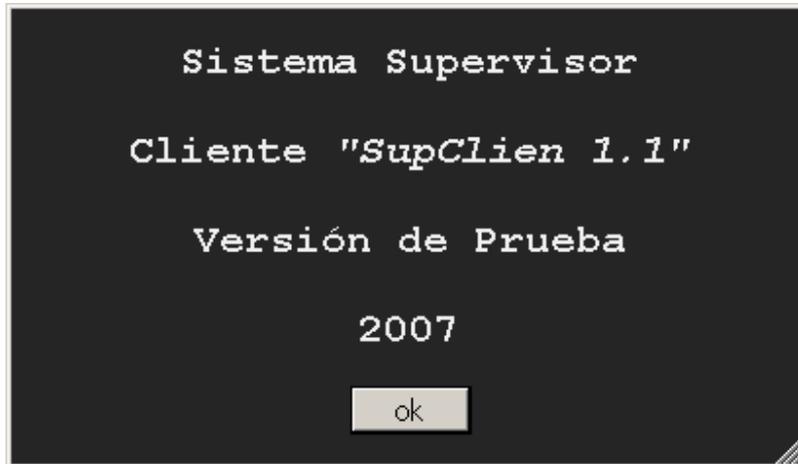


**Anexo 2: Fragmento del documento .txt en el que se registran las variables y sus características.**



```
Reporte tabla mediciones.txt - Bloc de notas
Archivo Edición Formato Ver Ayuda
"valor",          "hora",          "codigo"
9,               15:54:30,       ""
7,               15:54:44,       ""
8,               15:55:01,       "1"
6,               15:55:57,       "2"
4,               15:56:07,       "9"
7,               15:56:17,       "5"
8,               15:56:27,       "0"
0,               15:56:37,       "8"
10,              15:56:47,       "10"
1,               15:56:57,       "9"
9,               15:57:07,       "9"
3,               15:57:17,       "5"
6,               15:57:27,       "9"
9,               15:57:37,       "5"
5,               15:57:47,       "5"
9,               15:57:57,       "5"
5,               15:58:07,       "2"
5,               15:58:17,       "6"
9,               15:58:27,       "2"
4,               15:58:37,       "8"
9,               15:58:47,       "1"
0,               15:58:57,       "10"
6,               15:59:07,       "3"
4,               15:59:17,       "6"
```

**Anexo 3: Versión del SupClien 1.1**



**Anexo 4: Carpeta que contiene los archivos .txt los cuales muestran en el nombre el día en que fueron creados.**

