1 MEDICION Y CONTROL POR COMPUTADORA

1.1 Sistemas de tiempo real

La primera propuesta para usar un computador operando en *tiempo real* como parte de un sistema de control fue descrita en un documento por Brown y Campbell, en 1950. Allí aparecía una gráfica que mostraba un computador en un control realimentado.

La aplicación de computadores digitales al control industrial comenzó en la década de los 50´s. La primera instalación industrial de un sistema computarizado se realizó en setiembre de 1958, por parte de la Louisiana Power and Light Company, en USA. Sin embargo, no era un sistema de control. El primer sistema de control a lazo cerrado por computador fue hecho por la Texaco Company , en Texas, USA.

A partir de este inicio, se ha tenido un gran incremento en el uso de computadores digitales en el ámbito industrial. El hardware (HW) involucrado (procesadores, memorias, dispositivos de entrada/salida, etc.) se ha mejorado notablemente. Adicionalmente, el software (SW) utilizado (lenguajes de programación de tiempo real, aplicaciones de control listas para ser utilizadas por el usuario, etc.) ha tenido un gran desarrollo. Con los costos cada día más bajos de estos sistemas, se ha logrado una gran penetración de los mismos en casi toda actividad industrial moderna.

1.1.1 Elementos de un Sistema de Control Digital

El esquema básico de un sistema de control digital se muestra en la figura 1.1.

En esta figura, la *planta* hace referencia a todos los elementos (equipos, aparatos, máquinas, etc.) que conforman uno o más procesos de producción, manufactura, etc.

Los *dispositivos de entrada* son utilizados por el computador para leer los datos del proceso, requeridos por el sistema. Estos datos corresponden a los valores de las variables físicas del mismo, que pueden ser: señales discretas (estado de contactos o interruptores), o señales analógicas (valor de la presión, temperatura, pH, etc.).

Los *dispositivos de salida* son utilizados por el computador para entregar al mundo externo las señales requeridas para que el sistema de control actúe sobre el proceso. Estas salidas pueden ser: señales discretas o señales analógicas.

En el interior del computador se realizan las *tareas* programadas (corresponden al software del sistema), en acuerdo con los objetivos del control. Las *tareas de entrada* procesan las señales entregadas por los dispositivos de entrada. Realizan procesos tales como: linealización, cambio de escala, filtraje, etc.

Las *tareas de salida* procesan los datos calculados, antes de entregarlos a los dispositivos de salida. Realizan funciones tales como: generación de alarmas, cambio de escala, etc.

Las *tareas del control* son los diferentes algoritmos de control (PID, ON/OFF, etc.) utilizados por el sistema. Estas tareas se aplican a aquellos lazos de control a los cuales se les haya especificado o programado.

Las *tareas de comunicaciones* se utilizan para dar a conocer al usuario del sistema (operador, ingeniero de planta, gerente) las condiciones de todo el sistema. A su vez, son requeridas para poder modificar a voluntad las condiciones de trabajo de la planta.

La *interface de comunicaciones* adecua las señales entregadas por el computador digital a los diferentes *dispositivos de comunicación* que se tiene en el sistema de control. Los dispositivos de comunicación son necesarios para llevar a cabo la relación entre los seres humanos (usuario, operador) y el sistema de control.

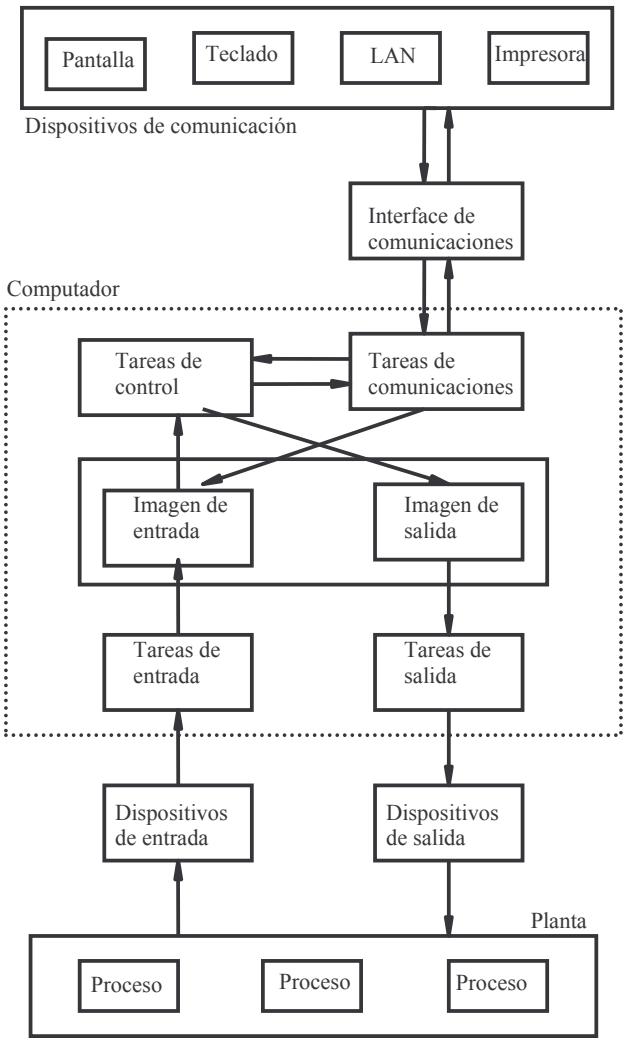


Figura 1.1 Esquema básico de un sistema de control digital

1.1.2 Definición de Tiempo Real

El sistema mostrado en la figura 1.1 trabaja en tiempo real. Pero ¿ qué significa: en tiempo real?. El autor J. E. Cooling lo define como:

Son sistemas en los cuales debe producirse la respuesta correcta en un espacio definido de tiempo. Si la respuesta del computador excede ese espacio de tiempo, entonces se obtendrá una degradación del desempeño o un mal funcionamiento.

El espacio de tiempo al que se refiere el párrafo anterior está determinado por los requerimientos del proceso. Así, una transacción bancaria puede aceptar tiempos del orden de los segundos, pero un sistema de control de velocidad de un motor requiere generalmente tiempos mucho menores.

1.1.3 Clasificación de los Sistemas de Tiempo Real

Dependiendo del tipo de sincronización que se tiene entre los procesos externos al computador y las tareas internas del mismo, los sistemas de tiempo real pueden clasificarse en:

Sistemas periódicos ó basados en un reloj. En este caso las tareas se realizan según un esquema de tiempo cíclico predefinido. Un ejemplo típico se ve en un lazo de control digital realimentado, en donde se debe muestrear la variable a controlar cada cierto tiempo; este tiempo de muestreo generalmente es constante y, por lo tanto, es cíclico.

Sistemas basados en eventos. Aquí, la ocurrencia de una tarea está determinada por la ocurrencia de un evento externo, el cual es típicamente no periódico. Por ejemplo, el sistema reacciona ante la apertura o el cierre de un interruptor, o ante la presencia de una condición de alarma porque una variable ha sobrepasado cierto valor.

Sistemas interactivos. En estos sistemas, la relación entre las acciones en el computador y el mundo externo no están definidas de forma muy fuerte, en cuanto hace referencia al tiempo en que se realizan. Típicamente el requerimiento es tal, que un conjunto de operaciones en el computador debería ser completado dentro de un cierto intervalo de tiempo. Ejemplos de estos sistemas son los sistemas bancarios de tarjetas y la reserva de vuelos áereos.

Otra forma de clasificar los sistemas de tiempo real (y las tareas de tiempo real) es la de dividirlos en las dos siguientes categorías:

1. Tiempo real duro (hard real-time)
2. Tiempo real suave (soft real-time)

Se tiene el primer caso cuando el sistema debe satisfacer siempre y en cada caso, los requerimientos de tiempo. El ejemplo indicado anteriormente sobre el tiempo de muestreo de una variable controlada es aplicable también en este caso. Por ejemplo, el sistema debe muestrear una señal de velocidad de un motor cada 10 milisegundos, ya que de otra manera el sistema de control no funcionaría correctamente.

En los sistemas de tiempo real suave, se espera que el sistema responda dentro de un cierto tiempo promedio. Por ejemplo, se espera que un cajero automático responda la petición de servicio de un usuario en un tiempo promedio de 10 segundos. Sin embargo, para una petición en particular, el tiempo de respuesta puede ser distinto de este valor. Lo que no se consideraría satisfactorio es que el tiempo de respuesta sea de 5 minutos en promedio.

1.1.4 Clasificación de los Programas

Los sistemas de control digitales requieren de *programas* (el software) para su funcionamiento, que *corren* en la memoria del sistema. Para desarrollar los programas se requieren de *lenguajes de programación*. Se pueden considerar tres tipos de programación usados en estos lenguajes:

1. Programación Secuencial. En este tipo de programación las acciones se ordenan en una estricta secuencia. Cada acción está determinada por una *instrucción, comando o declaración* propia del lenguaje. El programa consta de una serie de instrucciones que se ejecutan una a continuación de la otra. Un ejemplo típico es el lenguaje BASIC de hace algunos años.
2. Programación Multitarea. En este caso, el programa está constituido por un cierto número de *procesos o tareas* que se ejecutan en forma concurrente ó paralela (los procesos son módulos de programas cuyas instrucciones o comandos se ejecutan generalmente en forma secuencial). Los procesos se comunican entre sí a través de variables comunes o señales de sincronización. Por ejemplo, el Windows utiliza programación multitarea.
3. Programación en Tiempo real. En un programa de tiempo real, la secuencia de algunas de sus acciones no son determinadas por el diseñador sino por el medio ambiente en el que trabaja el sistema, es decir, por eventos externos que ocurren en tiempo real y sin ninguna referencia a las operaciones internas del computador. Inherentemente este tipo de programación involucra programación multitarea. Ejemplos de lenguajes de programación en tiempo real son: Modula 2, Ada, FORTH, LabVIEW, C sobre RT Linux, Java Real Time, etc.

1.2 Conceptos sobre Control por Computadora

Los procesos o aplicaciones industriales pueden ser clasificados de la siguiente manera:

1. Proceso por lotes (o batch). Ocurre cuando el proceso consiste en realizar una secuencia de operaciones para producir una cierta cantidad de un producto (un *lote*). Para producir otros lotes, las operaciones deben ser ejecutas nuevamente. Un ejemplo de un proceso en lotes es la producción del concreto.
2. Proceso continuo. Un proceso es continuo cuando la producción puede mantenerse por largos períodos de tiempo sin interrupción (en teoría, por un tiempo infinito). En estos procesos la materia prima se alimenta, se transforma y se obtiene el producto final en forma continua. Ejemplos del mismo son la producción del papel y del cemento.
3. Procesos de laboratorio o de prueba. En este caso el computador es utilizado para controlar cierto experimento complejo, o para controlar un equipo para probar automáticamente en una fábrica, tarjetas electrónicas u otros equipos. Un ejemplo típico es el control y el análisis de los datos generados por un cromatógrafo de fase de vapor.

En cualquiera de los procesos clasificados en la forma anterior, las actividades que se desarrollan por parte del sistema incluyen las siguientes:

- Adquisición de datos
- Control secuencial
- Lazos de Control Digital Directo
- Control supervisor
- Análisis de datos
- Almacenamiento de datos
- Interface hombre-máquina

Los objetivos que se persiguen al usar un computador para controlar el proceso incluyen:

- Eficiencia en la operación
- Facilidad de operación
- Seguridad
- Producto de mayor calidad
- Reducción del desperdicio
- Reducción del impacto ambiental debido a la producción
- Reducción del tiempo de producción
- Control Secuencial

El control secuencial es muy usado en las industrias químicas y alimenticias en donde las operaciones que se realizan a menudo involucran mezcla de materias primas, realización de algún proceso y luego la descarga del producto final. El control secuencial predomina en los procesos a lotes. El objetivo del control es la realización de una serie de tareas que deben realizarse en un estricto orden secuencial para la correcta obtención del producto final. Generalmente involucra acciones temporizadas, contar objetos, contar el número de veces que se realiza una acción, etc. Un ejemplo de la acciones típicas de un control secuencial es el siguiente:

1. Abrir la válvula A
2. Esperar 20 segundos
3. Abrir la válvula B
4. Si el interruptor S1 está abierto cerrar la válvula A, en caso contrario abrir la válvula C
5. Monitorear la temperatura. Cuando alcance el valor de referencia dado, cerrar todas las válvulas
6. ...

Es común encontrar una mezcla de control secuencial y *control realimentado* (o continuo), por ejemplo: controles de temperatura, de presión, etc.

El control secuencial puede variar desde sistemas muy complejos hasta bastante pequeños (el control de una lavadora doméstica, por ejemplo).

En la actualidad se fabrican equipos específicamente diseñados para efectuar control secuencial, como son los *Controladores Lógicos Programables* (o PLC: Programable Logic Controller). Como su nombre lo indica, el equipo puede ser utilizado en una variedad de procesos diversos, dependiendo de la programación que el usuario haga del mismo.

1.2.1 Lazos de Control Digital Directo

En un lazo de Control Digital Directo el computador se encuentra dentro del bucle de control. Dicho de otra manera, el controlador del sistema lo constituye el computador. Las ventajas de utilizar estos sistemas, sobre el control analógico, son: menor costo (con el uso masivo de los microprocesadores y microcontroladores, un controlador digital puede ser mas barato que uno analógico), mejor desempeño (mayores rangos de ajuste y precisión) y mayor seguridad (menos propensos a fallos).

En la actualidad el algoritmo de control más utilizado en estos sistemas es el bien conocido PID utilizado en el control realimentado, aunque también se está incrementando el uso de sistemas inteligentes de control (sistemas expertos, lógica difusa, redes neuronales, etc.). Las aplicaciones de sistemas de control digital directo van desde un simple lazo de control hasta las que contienen cientos y hasta miles de lazos.

El control digital directo no está limitado únicamente al lazo simple de control realimentado. Otras técnicas incluyen: control *inferencial*, control por *adelanto de la señal* (feedforward control) y el *control adaptivo*.

1.2.2 Control inferencial

El control inferencial es aquel en el cual, las variables en las que se basa la señal realimentada de control no se miden directamente, sino que se infieren a partir de mediciones efectuadas de otras variables del proceso (las cuales sí pueden ser medidas ó ser medidas más fácilmente), como se muestra en la figura 1.2. Un ejemplo típico del uso de este tipo de control es la medición de caudal a través de la diferencia de presión..

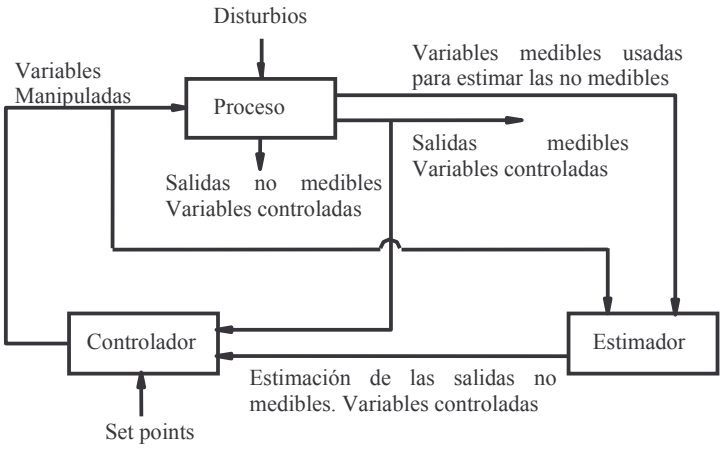


Figura 1.2 Sistema de control inferencial

1.2.3 Control por adelanto de la señal

El control por adelanto de la señal se usa frecuentemente en los procesos industriales. En este caso lo que se miden son los disturbios en el sistema en lugar de la salida del mismo, como se muestra en la figura 1.3. Por ejemplo, en una laminadora de acero, si la temperatura del acero es conocida antes de entrar al primer par de rodillos laminadores, la separación requerida entre los rodillos (para producir una lámina de un determinado calibre) puede ser calculada con precisión y puede estimarse la reducción de esta separación en cada rodillo en etapa subsiguientes del laminado. De esta manera es posible minimizar el tiempo total de laminado y la cantidad de desperdicio producido (lámina de calibre fuera de las tolerancias permitidas).

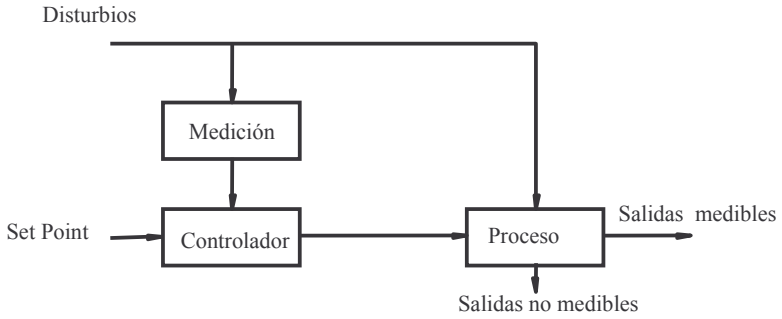


Figura 1.3 Sistema de control por adelanto de la señal

El efecto de introducir el control por adelanto de la señal es el de incrementar la velocidad de respuesta del sistema a los disturbios; sin embargo solo puede ser usado si los disturbios pueden ser medidos y en aquellas plantas en donde los efectos de los disturbios pueden ser predichos con precisión.

1.2.4 Control adaptivo

En el control adaptivo se modifica el ajuste de los parámetros del controlador de acuerdo a cambios que se presentan en el proceso, buscando que la respuesta del sistema sea siempre la misma, independientemente de las variaciones producidas en el proceso. Existen varias formas de control adaptivo, siendo uno de ellos el denominado *regulador de autosintonía* (self-tuning regulator), que se muestra en la figura 1.4.

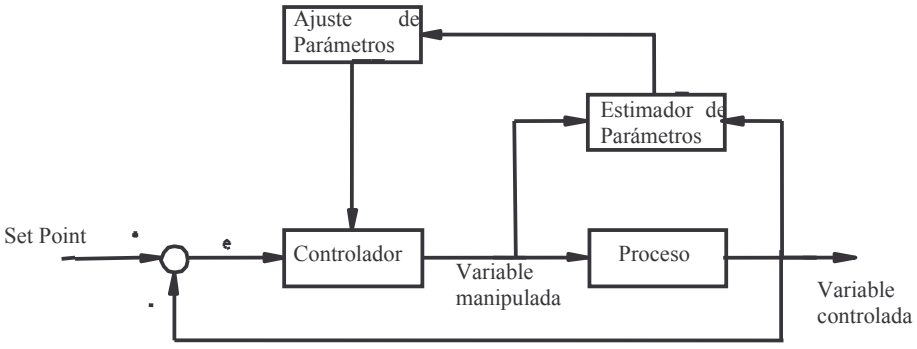


Figura 1.4 Sistema de control adaptivo

Como se puede observar, se requiere *identificar* el proceso para obtener los parámetros del mismo. Utilizando estos parámetros se reajustan los parámetros del controlador para seguir obteniendo la respuesta deseada.

Una forma alternativa de auto-sintonía se encuentra frecuentemente en controladores PID comerciales. En forma periódica, el controlador inyecta un pequeño disturbio al proceso y mide la respuesta. Esta respuesta es comparada con alguna respuesta deseada y los parámetros del controlador se ajustan para que la respuesta del proceso se ajuste a la deseada. La comparación puede estar basada en el porcentaje de sobrepaso ó en algo más complejo.

1.2.5 Control Supervisor

La idea básica del control supervisor se muestra en la figura 1.5. Los círculos marcados con la letra C representar controladores individuales (efectuando control digital directo) con su propio lazo de control.

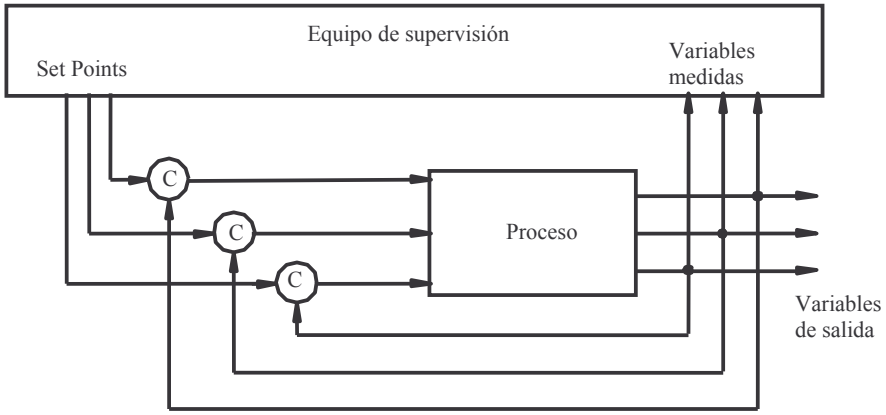


Figura 1.5 Sistema de control supervisor

En este caso el computador no hace control en forma directa, sino que se utiliza más bien para supervisar el funcionamiento total de la planta, permitiendo la modificación de los set-points de los controladores y el conocimiento de los valores de la variablesinvolucradas en los procesos. Esta información, útil para gerentes e ingenieros, permite visualizar más fácilmente el estado de la planta.

La mayoría de las aplicaciones del control supervisor son relativamente simples y están basadas en las características de estado estable de la planta. Sin embargo, cada día se incrementa la complejidad de los algoritmos de control utilizados. Un área actualmente en crecimiento son los *sistemas expertos* aplicados al control supervisor.

1.2.6 Sistemas Jerárquicos

En estos sistemas las tareas son divididas de acuerdo a la función que ejecutan, por ejemplo, un computador realiza operaciones de control digital directo, actuando como sirviente de otro que realiza funciones de control supervisor. Se obtiene así una estructura piramidal para el sistema, como se muestra en la figura 1.6.

Cada elemento de decisión recibe comandos del nivel superior y envía información de vuelta a dicho nivel.

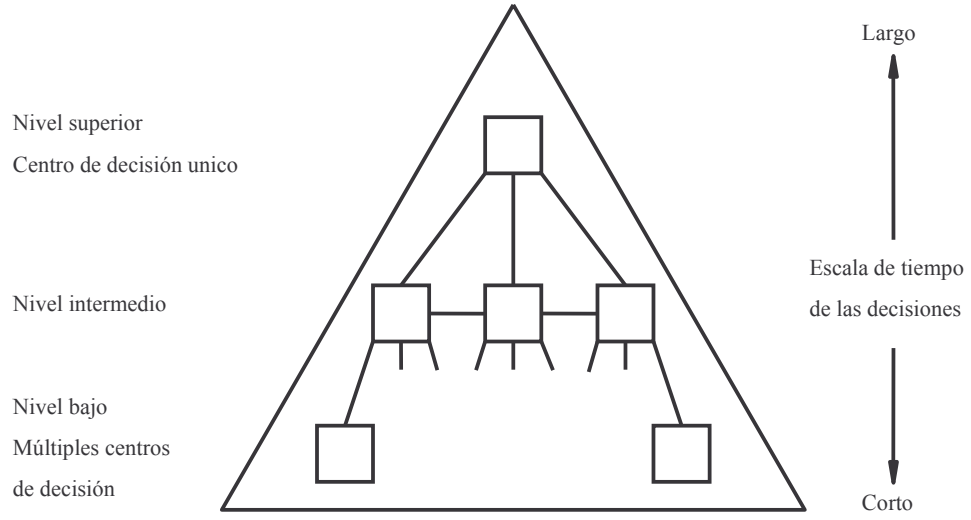


Figura 1.6 Sistema de control jerárquico

Los elementos en un mismo nivel pueden enviarse información entre sí. En la base de la jerarquía, típicamente se requiere respuestas rápidas (en milisegundos o segundos) a problemas simples. A medida que se avanza hacia niveles superiores de la jerarquía, la complejidad de los cálculos se incrementa y los tiempos de respuestas requeridos son también mayores (más lenta).

Como ejemplo de un sistema jerárquico podemos considerar el caso de una fábrica con una estructura de tres niveles: un nivel de administración, encargado de funciones de programación de la producción -; un nivel de supervisión, que conoce las *recetas* de los productos a producir y las secuencias de las operaciones a realizar y un nivel de control, encargado de operar las válvulas e interruptores, controlar las temperaturas, etc.

La mayoría de los sistemas jerárquicos involucran alguna forma de red distribuida y, por lo tanto, la mayoría de los sistemas es una mezcla de control jerárquico y control distribuido.

1.2.7 Sistemas Distribuidos

En un sistema distribuido muchos computadores trabajan en paralelo. Sus características principales son:

- Cada unidad realiza esencialmente las mismas tareas que las demás
- En el caso de falla ó de sobrecarga de una unidad en particular, todo o parte de su trabajo puede ser transferido a otras unidades

En otras palabras, el trabajo no se divide por función y designado a un computador en particular como en el sistema jerárquico, sino que el trabajo total es dividido y repartido en varios computadores. En la práctica se puede hacer una distribución menos compleja,

como por ejemplo: un computador realiza todas las entradas y salidas que no pertenecen a los procesos, otro ejecuta todos los cálculos de los controles digitales directos, otro realiza adquisición de datos y un cuarto computador realiza el control de los actuadores.

En los esquemas más modernos se utiliza una mezcla de control distribuido y jerárquico. Las tareas de medición, control digital directo, etc., son distribuidas entre varios computadores, los cuales son conectados entre sí vía un sistema de comunicación y configurados en una estructura jerárquica.

1.2.8 Interface Hombre-Máquina

La clave del éxito en la adopción de un sistema de control computarizado reside a menudo en las facilidades que provee el sistema a los operadores y usuarios del mismo. El sistema debe ser capaz de mostrar fácilmente el estado de la planta en un momento dado; facilidades de interactuar con los procesos (modificación de set-points, reconocimiento de alarmas, etc.); producir reportes, gráficos e informes estadísticos; historia de la operación de la planta; información económica; facilidades propias para el ingeniero de planta; etc.

Todos los aspectos anteriores se engloban generalmente en lo que se denomina la *interfaz hombre-máquina* del sistema. Un buen diseño de esta interfaz provee una interacción más agradable y sencilla con todo el sistema, facilitando el trabajo de los operadores.

1.2.9 El Ingeniero de Control

En toda planta industrial es necesaria la presencia de ingenieros con conocimientos en técnicas, métodos y tecnologías de control. Es también recomendable que este ingeniero tenga conocimientos sobre los procesos de producción que se realizan en la planta, facilitando así su comunicación con el personal dedicado a otras áreas que se manejan en la planta.

Las responsabilidades de un ingeniero de control son las siguientes:

1. Definir la estrategia apropiada de control para alcanzar los requerimientos del sistema
2. Definir las variables a medir, manipular y controlar, y establecer las constantes de escala, filtrado, puntos de alarmas, intervalos de muestreo, etc.
3. Definir los controladores a utilizar y las conexiones con los otros elementos del sistema
4. Sintonizar o ajustar los controladores de acuerdo a la especificación escogida
5. Definir y programar los procedimientos de control secuencial necesarios para la operación de la planta
6. Determinar e implementar el esquema de control supervisor que se requiera

1.3 Variantes en el desarrollo de aplicaciones

Existen una gran variedad de medios técnicos para el desarrollo de un sistema automatizado, desde medios convencionales hasta inteligentes, medios de cómputos de propósitos específicos, etc. Así mismo el software se ha desarrollado grandemente existiendo diferentes sistemas operativos, software para el desarrollo de aplicaciones,

software específicos desarrollados por diferentes firmas de automatización, etc. En esta sección se dará una visión general de las posibilidades que existen hoy en día.

1.3.1 Componentes de un sistema automatizado.

Un sistema automatizado por computadora tiene bien definido varios elementos que lo conforman:

1. El hardware.
2. El software.
3. La documentación.
4. La parte organizativa.
5. El personal de operación.

En el hardware encontramos todos los medios técnicos necesarios como son: los sensores, transmisores de señal, actuadores, controladores, los medios de computación, etc.

El software está constituido por los programas necesarios en el sistema, que incluye algoritmos, los modelos matemáticos de los procesos, los métodos utilizados, etc.

La documentación describe la información requerida por el sistema automatizado, las normas, las bases de datos, los ajustes requeridos por los controladores, etc. La documentación es de gran importancia sobre todo en sistemas grandes.

La parte organizativa está formada por el conjunto de instrucciones y reglamentos para el personal de operación, que incluye:

- organización laboral de los puestos de trabajo, turnos, etc.
- normas de operación de cada puesto de trabajo.
- especificaciones de calidad.
- procedimientos a ejecutar en caso de avería.
- instrucciones de seguridad.

El personal de operación se encarga de manejar la planta y está compuesto por:

- ingenieros, que se ocupan de la toma de decisiones de más alto nivel en el manejo de la planta
- tecnólogos, que tienen a su cargo el manejo rutinario del proceso.
- personal de mantenimiento, que garantiza el correcto funcionamiento del sistema automatizado.

Para el desarrollo de una aplicación industrial es importante tener presente todas las componentes, pues todas ellas son importantes para lograr una buena operación con eficiencia en todos los sentidos.

1.3.2 Variantes de los sistemas de medición y control.

En la actualidad existe un gran número de medios técnicos disponibles para el desarrollo de un sistema automatizado, como por ejemplo:

- **Unidades convencionales:** Lo constituyen diferentes instrumentos, como son: los sensores, transmisores de señal, controladores, etc., que unidos entre sí conforman el sistema de automatización. Cada instrumento es un dispositivo distinto.
- **Sistemas modulares**: Generalmente son pequeños con microprocesadores o microcontroladores para realizar funciones específicas de medición y control, como son los Controladores Lógicos Programables (PLC). Generalmente utiliza tarjetas insertables o módulos, escogidas de acuerdo a las exigencias de la aplicación.
- **Sistemas de control distribuido (DCS)**: Son sistemas de control diseñados para aplicaciones de tamaño mediano a grandes. Generalmente constan de una diversidad de equipos conectados en red, como son: acondicionadores de señal, unidades de control, consolas de trabajo computarizadas, monitores de despliegue de información, impresoras, etc. Estos sistemas generalmente son muy costosos.

En un sistema automatizado pueden utilizarse varios de los medios disponibles. En la actualidad se busca que los medios tengan gran conectividad entre sí.

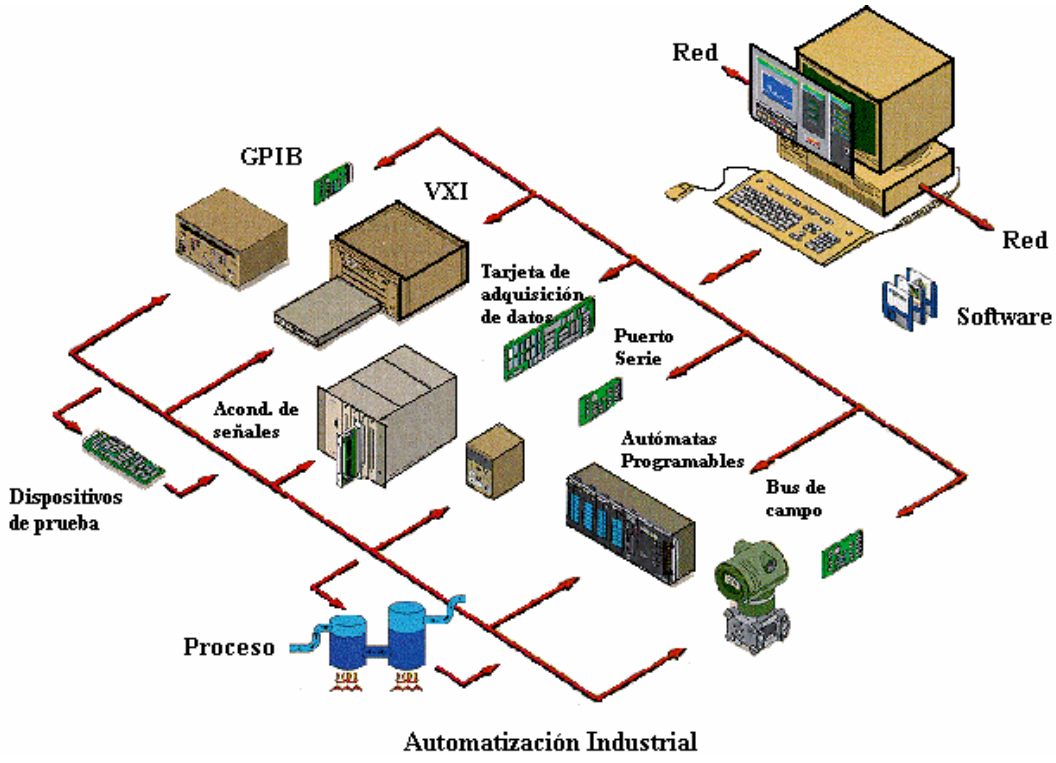


Figura 1.7 Distintas posibilidades de conexión de un sistema automatizado

La escogencia de los medios técnicos depende de las necesidades del usuario y de sus posibilidades económicas.

En la figura 1.7 se muestran diferentes posibilidades de conexión de diversos medios técnicos, en la conformación de un sistema automatizado. Se pueden plantear las siguientes:

- Instrumentos con norma GPIB (General Purpose Interface Bus), a través de tarjetas de interfase GPIB, insertable en una computadora.
- Interfaces para adquisición y acondicionamiento de señales.
- Tarjetas de adquisición de datos insertables en módulos externos ó en un computador.
- Instrumentos que se comunican a través de interfaces serial standard (RS232).
- Interfase para bus de campo para la atención de dispositivos en el campo.
- Controladores lógicos programables

1.3.3 Variantes en la programación de un sistema.

Para la programación de una aplicación se puede seguir tres estrategias:

- **Utilizar sistemas operativos (SO) y lenguajes estándar**: Esta estrategia es la de menor costo, mayor tiempo de proyección y menor confiabilidad. En estos casos hay que realizar las propias funciones para el trabajo en tiempo real como por ejemplo reprogramar el servicio de interrupción de tiempo de la máquina, establecer prioridades, manejo de recursos, etc. Para sistemas pequeños en tiempo real puede utilizarse esta variante.
- **Utilizar un SO de tiempo real y lenguajes con bibliotecas para tiempo real**: Esta alternativa requiere un costo medio, ya que se tienen facilidades para la programación y funciones de específicas para un sistema de automatización. Ejemplo: Time-slicer sobre DOS y "C" con biblioteca, LabVIEW, LabWindows sobre windows. Permite flexibilidad para realizar aplicaciones a la medida. Para lograr una alta confiabilidad es necesario mucha depuración off-line.
- **Sistemas específicos para medición y control**: Con ellos se logra una rápida implementación del sistema. Pero tiene el inconveniente de que es costoso. Por ejemplo: Lookout, Genie, Intouch, Lookout, BridgeVIEW, Factory link., Sistema SCAP, RSVIEW.

1.3.4 Tareas típicas en un sistema de medición y control

En un sistema de medición y control por computadoras se realizan tareas que se ejecutan en diferentes medios técnicos. A continuación se mostrarán varias de las tareas mas típicas.

1. Adquisición y procesamiento primario de la información.

Son tareas para la adquisición, validación, filtraje, escalado, etc. de la variable de proceso. Típicamente son periódicas y pueden existir varias con diferentes periodos de muestreo. Es usual tomar un periodo de muestreo básico, la menor, y las demás con múltiplos de estas, por ejemplo: 1, 2, 5, 10 ..seg. Si tenemos que por deducción teórica,

por la experiencia o porque se determinó un período de muestreo para una señal que no esté en esos valores se toma el próximo inferior. Por ejemplo, si tenemos que una variable debe ser muestreada cada 8.5 seg , entonces la muestreamos cada 5 seg.

Estas tareas son ejecutadas cercanas al proceso, en muchos casos en dispositivos específicos. Son de la más alta prioridad, pues son la base para muchas tareas de nivel superior (regulación, presentación, transmisión, etc.).

Estas tareas por lo general pueden dar entre otras cosas: el resultado de la medición, su estado, esto es, si es válida, si entró en alarma, si está inhabilitada, etc.

2. Atención a alarmas, arranque, paradas, condiciones anormales, etc.

Estas tareas se ejecutan normalmente a solicitud de otras (por ejemplo, por violación de límites) o por eventos (ejemplo: solicitud del operador, señal ON-OFF proveniente del proceso de violación de límite). Su ejecución es aperiódica con alta prioridad.

3. Regulación.

Consiste en la ejecución de un algoritmo de control. Pueden existir varias con diferentes periodos de muestreo. Generalmente pueden tomar la información de las tareas de medición. Los lazos de regulación deben tener las opciones de automático y manual, y tener presente el paso de manual a automático sin saltos.

Estas tareas pueden estar en un primer o segundo nivel de automatización. Su prioridad alta, pero más baja que la de medición.

4. Cálculos.

Pueden existir tareas periódicas para el cálculo de variables que no se midan directamente, como son: integraciones, cálculos de eficiencia, promedios, etc. Muchas son consideradas como variables calculadas y pueden posee todos los parámetros típicos de una variable de proceso, como por ejemplo los límites de alarmas. Normalmente los periodos son mas altos que los de medición; por ejemplo, 5 minutos para los cálculos de integración.

Además, existen tareas aperiódicas a solicitud del operador o por ocurrencia de algún evento, por ejemplo: cálculos de resúmenes, cálculos económicos, etc.

Muchas de las variables calculadas sirven de base para los reportes periódicos.

5. Presentación de la información.

Tarea comúnmente llamada de refrescamiento de pantalla. Toman la información a presentar de las tareas de medición, cálculos, reportes, etc.. Las pantallas de interfaces hombre-máquina poseen su propio periodo de refrescamiento, determinado por la respuesta del operador humano, que no distingue cambios rápidos de información. El periodo típico de estas tareas es de 1 seg.

Desde una pantalla activa, se puede llamar y presentar a otras, por ejemplo, una representación general y de ahí presentar representaciones de áreas, pantallazos de alarma, etc.

Estas tareas presentan baja prioridad.

6. Atención al operador.

Tarea que atiende teclado, mouse, lápiz óptico, etc. Es deseado que esta tarea se active por interrupción. Pasa la información a otras tareas. Generalmente esta tarea es la de mas baja prioridad, para que la lentitud del operador no afecte la ejecución del sistema.

7. Reportes

Los reportes son tareas generalmente calendarias coincidiendo con cambios de turnos, meses, etc. Pueden existir reportes a solicitud del operador en el momento que estime conveniente. Pueden existir reportes de incidencias, alarmas, análisis técnico-económicos, etc. Los reportes de alarmas suelen visualizarse o imprimirse según van apareciendo. Los reportes se pueden realizar en pantalla, impresora o grabar en disco. Presentan baja prioridad.

8. Controles lógicos secuenciales

Tareas aperiódicas que pueden activarse por solicitud externa o del operador. Se caracterizan por acciones lógicas, abrir, cerrar, bloquear, etc. Suelen ejecutarse independientemente de las de medición y regulación. Por ejemplo: establecer la secuencia lógica y set-point para el arranque de una caldera. Típicamente estas tareas son las que realiza un Controlador lógico programable (PLC). Pero en los sistemas de control modernos, basados en técnicas digitales, pueden implementarse estas funciones para que coexistan en un mismo equipo con las tareas de regulación.

9. Comunicación

Estas tareas se encargan de la transmisión de la información a los diferentes niveles de automatización en tiempo real. Pueden ser tareas periódicas, como por ejemplo para la transmisión de las señales medidas al nivel superior. También puede ser a solicitud de los diferentes niveles de automatización. En otros casos estas tareas se activan con la ocurrencia de algún evento, por ejemplo, cambio de algún valor discreto que se debe informar.

Pueden tener alta prioridad, dependiendo de la importancia que se le de y de los buffer que se disponga para la comunicación. Suelen utilizarse buffers de entrada y de salida, de manera tal que todas las tareas que necesiten enviar información la depositen en los buffers y las tareas de comunicación se encargan de enviarla al respectivo destino.

10. Optimización, despacho, análisis técnico-económicos, etc.

Estas tareas se ejecutan en los niveles superiores de automatización y tienen como objetivo establecer los ajustes y valores deseados más convenientes para la producción. Pueden ser periódicas con períodos relativamente grandes o aperiódicas por solicitud del operador u otras tareas. Son de baja prioridad.

Ejemplo de tareas

Como ejemplo de las tareas existentes en un sistema de automatización, sea un control preliminar de la densidad de la pulpa en una fábrica de papel, cuyo esquema simplificado es mostrado en la figura 1.8. Se regula el nivel de la pulpa en el tanque de la izquierda. Se regula el nivel de agua en el tanque de la derecha. La pulpa y el agua se mezcla en el tanque del centro de la figura. La densidad de la mezcla de salida es regulada con la entrada de agua: se aumenta la entrada de agua si de desea una menor densidad de pulpa a la salida y se disminuye si se desea una menor densidad. Obsérvese que se tienen sensores para el nivel en los tanques y para la densidad del mezcla de salida, y reguladores para los flujos de entrada de la pulpa y el agua.

Nos interesa establecer las variables a medir y su objetivo, y establecer las tareas para un sistema de medición, control, y supervisión por computadora.

La tabla mostrada a continuación muestra las variables de proceso escogidas para ser medidas, su tipo (si es discreta u on-off, o analógica) y el objetivo del sistema de supervisión sobre las mismas (si se implementa una alarma, un bloqueo sobre un flujo, etc.). Así mismo, en otra tabla se muestran las variables que deben calcularse a partir de las variables medidas, así como el objetivo de las mismas. Por último se hace un listado de las tareas básicas que debe realizar el sistema.

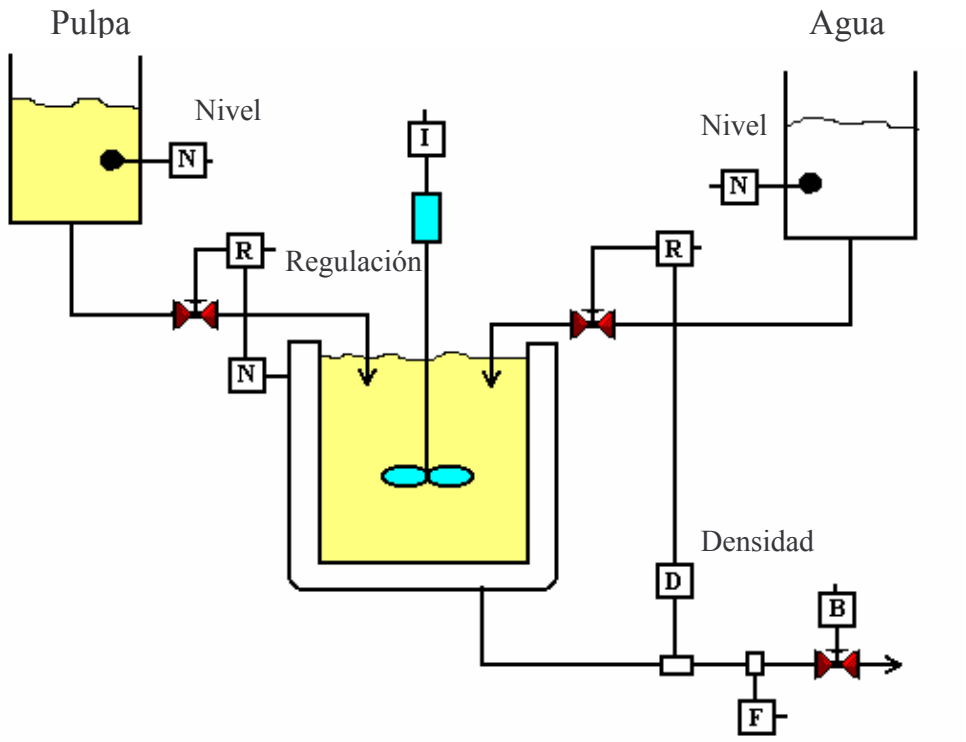


Figura 1.8 Esquema simplificado del proceso de la pulpa en una fábrica de papel

Por ejemplo, se debe detectar el nivel mínimo de la pulpa en el tanque. Para ello se utiliza una indicación de tipo on-off que indica si la pulpa alcanzó en su nivel mínimo o no. Además, se debe medir el nivel de la pulpa en el tanque, para lo cual se requiere de un sensor de nivel de tipo analógico.

Variable medida	**Tipo**	**Objetivo**
Nivel mínimo de la pulpa	on-off	Alarma, bloqueo
Nivel máximo de la pulpa	on-off	Alarma, bloqueo entrada de pulpa
Nivel mínimo del agua	on-off	Alarma, bloqueo
Nivel máximo del agua	on-off	Alarma, bloqueo entrada de agua
Agitador funcionando	on-off	Alarma, bloqueo
Corriente máxima del agitador	on-off	Alarma, bloqueo

Nivel	analógica	Regulación, indicación, registro, Alarma
Densidad	analógica	Regulación, indicación, registro, Reporte
Flujo de salida	analógica	Indicación, reporte

Variable calculada	**Objetivo**
Densidad media de la pulpa de salida.	Indicación, reporte.
Cantidad de pulpa producida.	Indicación, reporte.

Las tareas básicas asociadas a esta sección son:

1. Adquisición y procesamiento de las variables analógicas. Estas tareas serán periódicas en dependencia de los periodos de muestreo, con alta prioridad.
2. Adquisición y procesamiento de las variables discretas. Periódica, con bajo periodo de muestreo, o por interrupción en dependencia del hardware. Máxima prioridad.
3. Regulación. Periódica en dependencia de los periodos de muestreo de las señales realimentadas o de los algoritmos de control.
4. Procesamiento de las variables calculadas. Periódicas, con periodos de muestreo relativamente alto, o a solicitud del operador por teclado (o mouse)
5. Presentación de la información. Refrescamiento cada 1 seg.

   Pantallas de:

   - Selección general y estado del sistema. Es la que se presenta en el inicio.
   - Mímico. Representación mímica de la sección y la indicación de variables y estados de alarmas.
   - Registros. Pantalla para visualizar las variables a registrar.
   - Alarmas. Pantalla donde se refleja la información de las alarmas ocurridas y su estado presente.
   - Reportes. Pantalla que refleja la información resumen de una jornada de trabajo, puede ser solicitada en cualquier momento por el operador.
   - Densidad media.
   - Pulpa producida.

6. Comunicación con el nivel superior. Almacenaje en fichero compatible con bases de datos de los parámetros del sistema, así como los estados y resultados actualizados.

1.4 Ejemplos de software para el desarrollo de aplicaciones de medición y control.

A continuación se presentarán las características mas generales de algunos softwares utilizados para aplicaciones de tiempo real.

1.4.1 Algunas características del RSVIEW.

- Software de para medición, control, monitoreo, supervisión de la firma Rockwell Software.
- La programación consiste en la configuración de los diferentes objetos tanto para la aplicación como del sistema.
- Facilidades para el trabajo directo con los PLCs Allen Braeley.
- Trabajo sobre el sistema operativo Windows lo que posibilita el uso de herramientas del propio sistema.
- Comunicación con otras aplicaciones por DDE, OLE.
- Facilidades para la creación de representaciones gráficas del proceso con animación.
- Facilidades para el trabajo en red. Uso de TCP/IP.
- Trabajo directo con bases de datos y uso del SQL lo que permite que otras aplicaciones puedan utilizar los resultados.
- Posee varias posibilidades para el tratamiento de alarmas.
- Se establecen varios niveles de seguridad para el acceso a las diferentes partes del sistema.

1.4.2 Algunas características del LOOKOUT

- Constituye un software para la automatización industrial.
- Totalmente orientado a objeto
- No existe límite en cuanto a la cantidad de objetos, paneles, puntos de E/S que puede gobernar
- Permanece On-line mientras se agrega algún objeto a la aplicación.
- Incluye una amplia biblioteca de gráficos, no obstante se pueden crear sus propios gráficos.
- Admite files: Windows metafiles (.WMF), bitmap (.BMP), AutoCad, Photographs.
- Permite intercambio dinámico de datos (DDE) con otras aplicaciones.
- Amplio tratamiento de las alarmas
- 10 niveles de seguridad

1.4.3 Algunas características del LabVIEW para Windows:

- Usa un lenguaje de programación gráfica llamado "G".
- Diseñado para instrumentación, procesamiento, control, presentación de la información de forma eficiente.
- Consiste en construir programas denominados Instrumentos Virtuales (VI) por su similitud con instrumentos reales.
- Usa la programación por flujo de datos.
- Velocidad del código compilado similar al "C".
- Multitarea. Se pueden crear caminos de flujos de datos independientes o paralelo. Se pueden ejecutar otras aplicaciones mientras se ejecuta una aplicación.
- Gran cantidad de funciones para ayudar a la programación.
- Biblioteca específica de aplicaciones para:
  Adquisición de datos.
  Control de instrumentos seriales y con buses GPIB

Análisis de datos.
Presentación de la información.
Almacenaje de datos.

- Funciones específicas para
  Generación de señales.
  Procesamiento de señales.
  Filtros.
  Ventanas.
  Estadísticas.
  Regresión.
  Algebra lineal.
- Manejo de arreglos.
- Controles e indicadores para la presentación de la información:
  Gráficos.
  Interruptores, indicadores visuales.
  Tanques, termómetros, etc.
- Análisis en tiempo real, posee funciones para el trabajo en tiempo real.
- Diferentes posibilidades para la comunicación con el exterior, uso de puertos, drivers en DDL, comunicación serie, enlace con redes, protocolo TCT/IP para la comunicación en redes, etc.
- LabVIEW posee VIs para controlar instrumentos a través de los buses GPIB, VXI, RS232. Posee "drivers" para más de 150 interfases de 40 fabricantes.
- Oferta drivers para la comunicación con diferentes PLCs.
- Programación modular y jerárquica. Cada VI confeccionado puede así mismo ser una función de otra aplicación.
- Posibilidad del uso de intercambio dinámico de datos DDE.
- Posibilidad de uso de objetos de otras aplicaciones a través de OLE.
- Manejo de base de datos, posibilidades de uso de SQL.
- Fácil, su puesta a punto.
- En el LabVIEW se promueve el concepto de programación modular. Esto es, una aplicación consta de varios subVIs y estos a su vez de otros. Poseen la característica a diferencia de otros lenguajes, que los subVIs pueden ejecutarse independientemente, facilitando su puesta a punto. Por estas propiedades de forma muy fácil un VI una vez desarrollada se puede aplicar a varias aplicaciones.

Algunas características de la programación por flujo de datos:

- Para explotar al máximo el paralelismo de un programa se utiliza la programación por flujo de datos. La idea básica consiste en posibilitar la ejecución de una instrucción en cuanto sus operandos estén disponibles. Por lo que no se necesitan contadores de programas.
- La inicialización de las instrucciones depende de la disponibilidad de los datos, con independencia de su ubicación física en el programa.
- Cada elemento, llamado nodo, consiste de: operador, los receptores de operandos y los destinos de los resultados.
- La regla de disparo de un nodo exige que todos los receptores estén ocupados por los valores, o sea, que todos los datos de entrada estén disponibles.
- Los programas por flujos de datos se representan por grafos dirigidos que muestran el flujo de datos entre los nodos.

- No existe el concepto de memoria de datos compartidos por lo que no se producen efectos colaterales.
- La programación por flujo de datos es apropiada para la implementación distribuida. Se puede usar en aplicaciones altamente concurrente. Como desventaja se le sitúa que tiende a gastar excesivo espacio de memoria.

1.5 Introducción al LabVIEW

LabVIEW es una aplicación para el desarrollo de programas de propósitos generales, como C o BASIC. Pero a diferencia de éstos, que son lenguajes de programación basados en texto para crear líneas de código, LabVIEW usa un lenguaje de programación gráfico, el G, para crear programas en forma de diagrama de bloques.

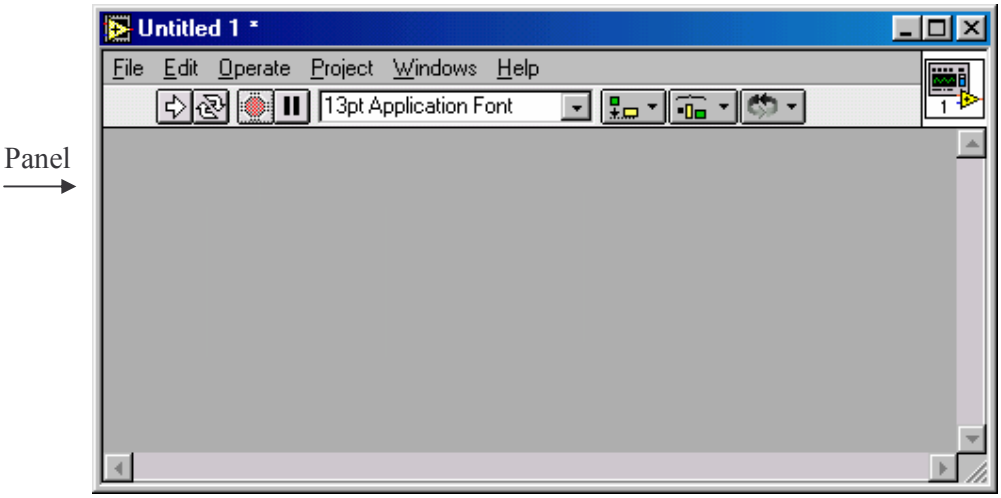
Como cualquier otro lenguaje de programación moderno, LabVIEW posee una extensa librería de funciones y subrutinas para realizar cualquier tarea de programación. Pero además, contiene librerías de aplicación específica para adquisición de datos, GPIB y control por puerto serial, análisis de datos, representación almacenamiento de datos. También incluye herramientas convencionales para el desarrollo de programas, como son: puntos de ruptura en un programa, ejecución animada y ejecución paso a paso, que facilitan la depuración y el desarrollo. LabVIEW ha sido desarrollado por National Instruments Corporation.

1.5.1 Componentes del LabVIEW

Los programas creados en LabVIEW tienen la extensión **.vi**, sinónimo de *Virtual Instrument*. Esta denominación de *Instrumento Virtual* nace de la idea de que el lenguaje permite la creación de interfaces gráficas en la pantalla de un computador, que semejan la cara frontal de un instrumento real, como por ejemplo un osciloscopio, que el operador puede operar mediante el teclado o el ratón del computador.

Un VI consta de los siguientes elementos:

1. El *Panel*, en donde se desarrolla la interfaz gráfica del programa



2. El *Diagrama*, en donde se desarrolla el algoritmo del programa, en forma gráfica.

Diagrama →

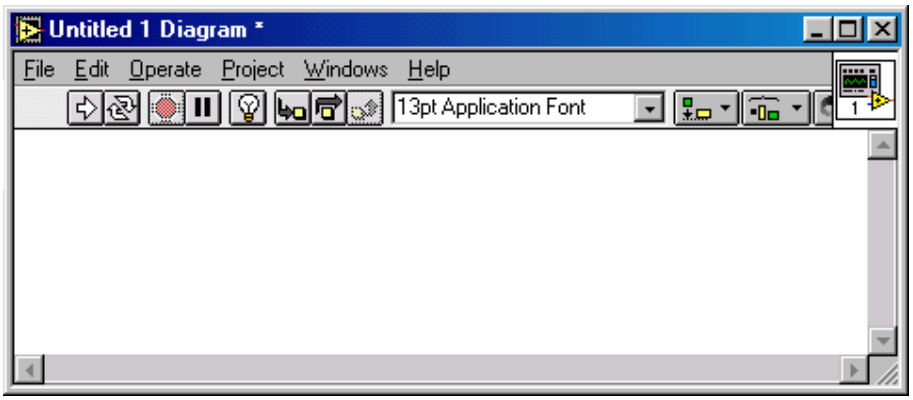


Figura 1.9 El *Panel* y el *Diagrama* del entorno de programación del LabVIEW

En la figura 1.10 se muestra un programa sencillo (el panel y el diagrama) desarrollado en LabVIEW. El nombre de programa es *Medida.vi*.

En el panel se muestran dos objetos gráficos con los nombres de *Entrada* y *Salida*: el primero simula un potenciómetro y el segundo un metro de salida. Al correr el programa, si el operador acciona la entrada, el metro de salida indicará el valor introducido por el operador. Para que el programa realice esta función es necesario agregarle el "algoritmo" respectivo, lo cual se efectúa en el diagrama.

En el diagrama del centro de la figura 1.10, vemos dos rectángulos con los nombres de *Entrada* y *Salida* que, lógicamente, representan a los objetos del panel. El "algoritmo" en este caso consiste simplemente en unir (o "cablear") los dos rectángulos mediante una línea (o "cable") de tal manera que los datos generados por la entrada pasen a la salida (flujo de datos). Esta tarea debe ejecutarla el programador. El resultado se muestra en el diagrama inferior de la figura.

El desarrollo de cualquier VI en LabVIEW sigue entonces los siguientes pasos:

1. Realizar la interfaz gráfica en el panel
2. Cablear los elementos respectivos en el diagrama. El cableado debe hacerse de tal manera que el programa realice la función que se desea, según el algoritmo apropiado.

Con el objeto de realizar una interfaz apropiada, en LabVIEW es posible modificar el tamaño de los objetos colocados en el panel, así como su color. Se puede agregar texto y líneas decorativas que mejoran el aspecto de la interfaz, del color que se desee.

Otro componente importante de cada programa o VI implementado en LabVIEW es el *icono* por intermedio del cual se le identifica. En la figura 1.10 se muestra el icono que por defecto se coloca inicialmente a todo VI. El usuario puede modificar el icono a voluntad. Se recomienda hacer un dibujo y/o un texto que identifique fácilmente la función del VI. Cuando un VI se utiliza en otro VI, el icono del mismo lo representa.

Adicionalmente a icono, se deben establecer el *conector* del VI. En el conector se sitúan los puntos de entrada y salida del VI, para que pueda ser cableado a otros VI's.

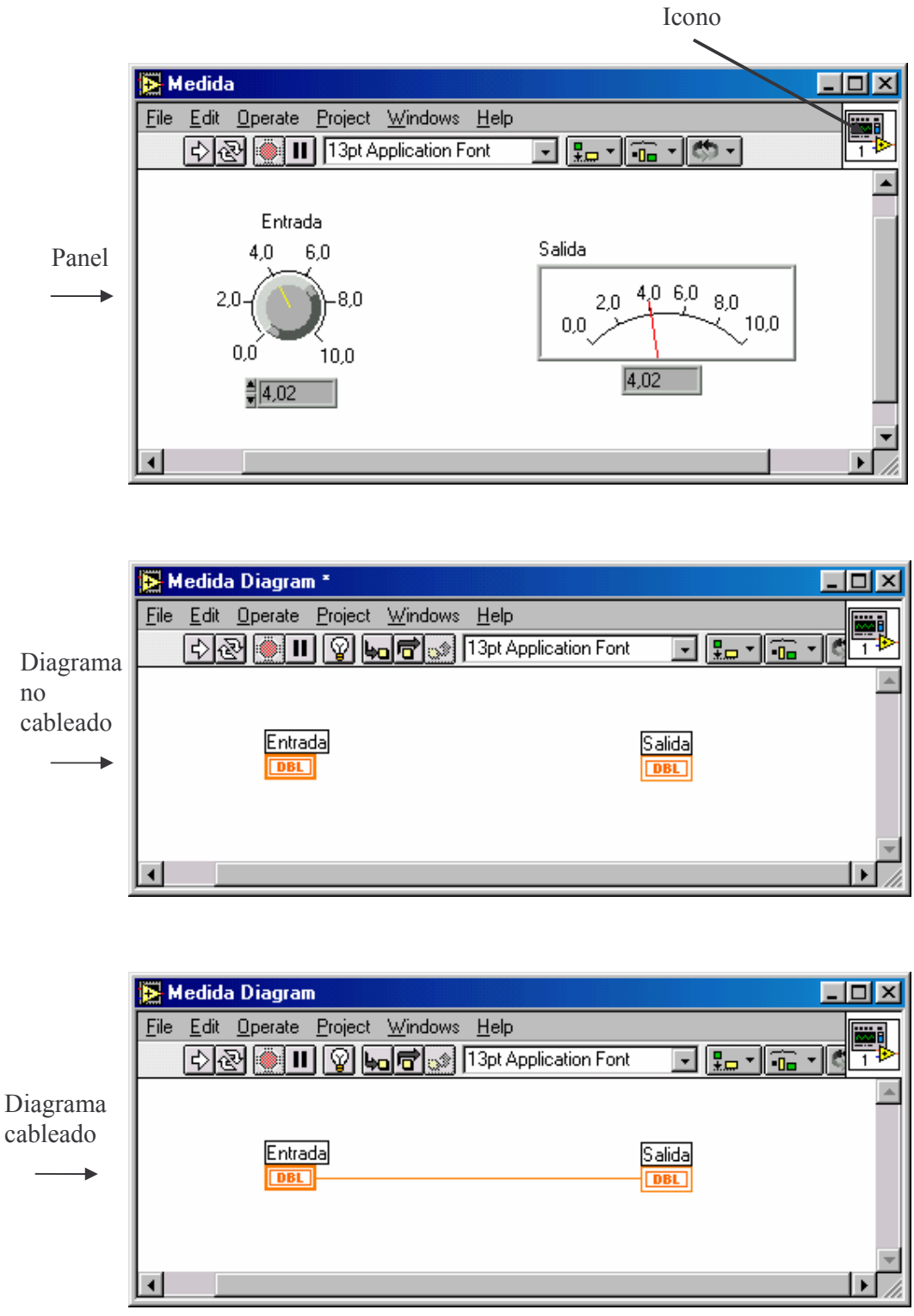


Figura 1.10 Programa sencillo en LabVIEW

1.5.2 Las paletas (*Palette*) de LabVIEW

Para las labores de programación, LabVIEW posee tres paletas:

1. La *Paleta de Herramienta* (Tools palette). Aquí es posible escoger (mediante el ratón), herramientas que permiten mover y dimensionar los objetos, accionar los controles cuando se ejecuta el programa, modificar el color, introducir puntos de ruptura en un programa, cablear, introducir texto, etc.
2. La *Paleta de Controles* (Control palette). Se accede a esta paleta cuando se trabaja en el panel. Permite escoger los distintos tipos de objetos que pueden colocarse en el panel para establecer la interfaz gráfica del programa. Cualquier objeto puede funcionar ya sea como *Control*, si permite al operador introducir datos a la aplicación, o como *Indicador*, cuando sirve para mostrar datos al operador.
3. La *Paleta de Funciones* (Function palette). Se accede a esta paleta cuando se trabaja en el diagrama. Permite escoger las distintos funciones que posee LabVIEW para realización de cálculos, para el manejo de los diferentes tipos de datos, etc. Sirven para la realización del algoritmo del programa.

La figura 1.11 muestra las tres paletas mencionadas anteriormente.

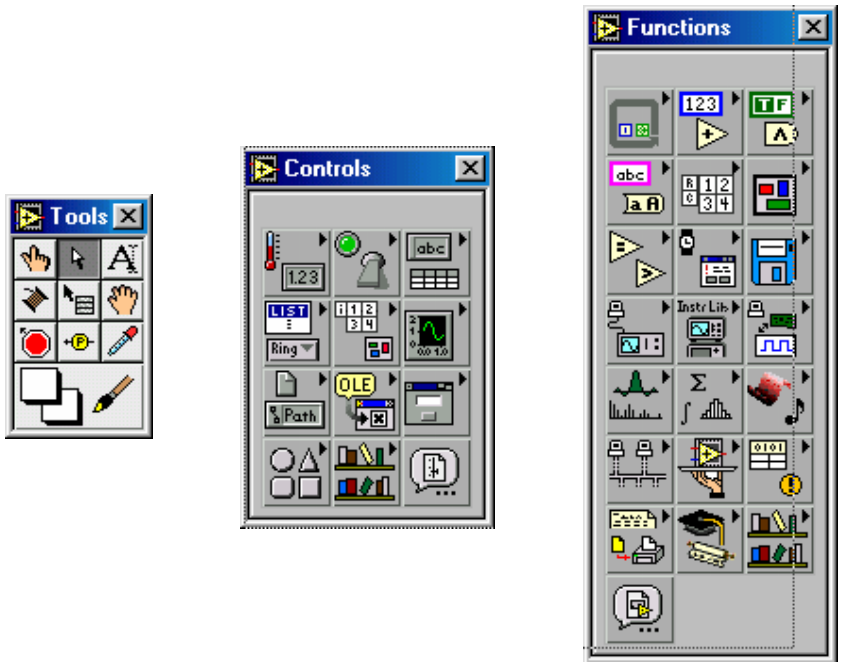


Figura 1.11 Las paletas de LabVIEW

En la figura 1.11 observamos que la paleta de controles está dividida en una retícula de 3x4 iconos, cada uno con un dibujo diferente. Accediendo con el ratón a estos iconos se despliega otra sub-paleta. Por ejemplo, al situarse en el primer icono se despliega la subpaleta *Numeric* (*numérica*) en donde es posible escoger el objeto que se desee colocar en el panel. Generalmente, cada subpaleta permite escoger objetos asociados con un determinado *tipo de datos*. Así, la subpaleta numérica contiene objetos que

permiten introducir o mostrar datos del tipo numérico. La subpaleta *String* (*Cadena*) contiene objetos asociados al tipo de dato de cadena o alfanumérico.

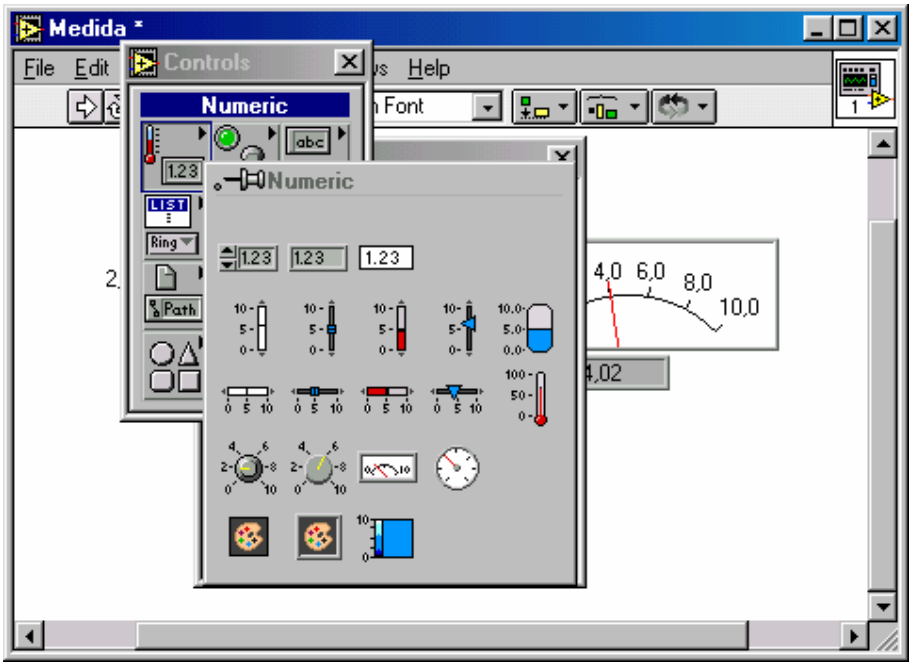


Figura 1.12 La subpaleta *Numeric* de la Paleta de Controles

El la paleta de controles se tienen las siguientes subpaletas:

- Numeric controls: para introducir y desplegar datos numéricos.
- Boolean controls: para introducir y desplegar datos booleanos (True/False).
- String & Table controls: para introducir y desplegar textos.
- List & Ring controls: para desplegar y/o seleccionar desde una lista de opciones.
- Array & Cluster controls: para agrupar un conjunto de datos.
- Graph controls: para graficar datos numéricos.
- Path & Refnum controls: para introducir y desplegar rutas de archivo y pasar números de referencia de un VI a otro.
- Decorations: para añadir decoraciones en la presentación de los VI.
- User controls: para grabar en disco controles creados por el usuario.
- ActiveX: para dar soporte al manejo de objetos ActiveX.
- Select a control ...: para acceder a los controles creados por el usuario.

En forma análoga a la paleta de controles, la Paleta de Funciones también está subdividida en subpaletas, como se puede observar en la figura 1.11. Las subpaletas son las siguientes.

- Structures functions: contiene las estructuras de control de programas.
- Numeric functions: funciones para el manejo de datos numéricos.
- Boolean functions: funciones para el manejo de datos booleanos.
- String functions: funciones para el manejo de datos de cadena.

- Array functions: : funciones para el manejo de los arreglos.
- Cluster functions: funciones para el manejo de los conjuntos de datos.
- Comparison functions: funciones de comparación entre tipos de datos.
- Time & Dialog functions: funciones relativas al tiempo y a las ventanas de diálogo.
- File I/O functions: funciones relativas al manejo de archivos.
- Instrument I/O functions: funciones para la interface con instrumentos externos.
- Instrument drivers functions: colección de drivers para el manejo de instrumentos.
- Data adquisition functions: funciones para el manejo de tarjetas de adquisición de datos.
- Signal processing functions: funciones para el procesamiento digital de señales.
- Mathematics functions: funciones para el manejo de ecuaciones matemáticas.
- Graphic & sound functions: funciones para el manejo de gráficas y sonido.
- Communicatios functions: funciones relativas a la comunicaciones.
- Application controls functions: funciones sobre la Ayuda, los Menús y la impresión.
- Advanced functions: grupo de funciones avanzadas.
- Report generation functions: funciones para la generación de reportes.
- Tutorial: acceso al tutorial de LabVIEW.
- User libraries: acceso a librerías creadas por el usuario.
- Select a VI ...: acceso a un VI creado por el usuario y almacenado en disco.

1.5.3 Ejemplo de una aplicación en LabVIEW

La figura 1.13 muestra un ejemplo de una aplicación en LabVIEW

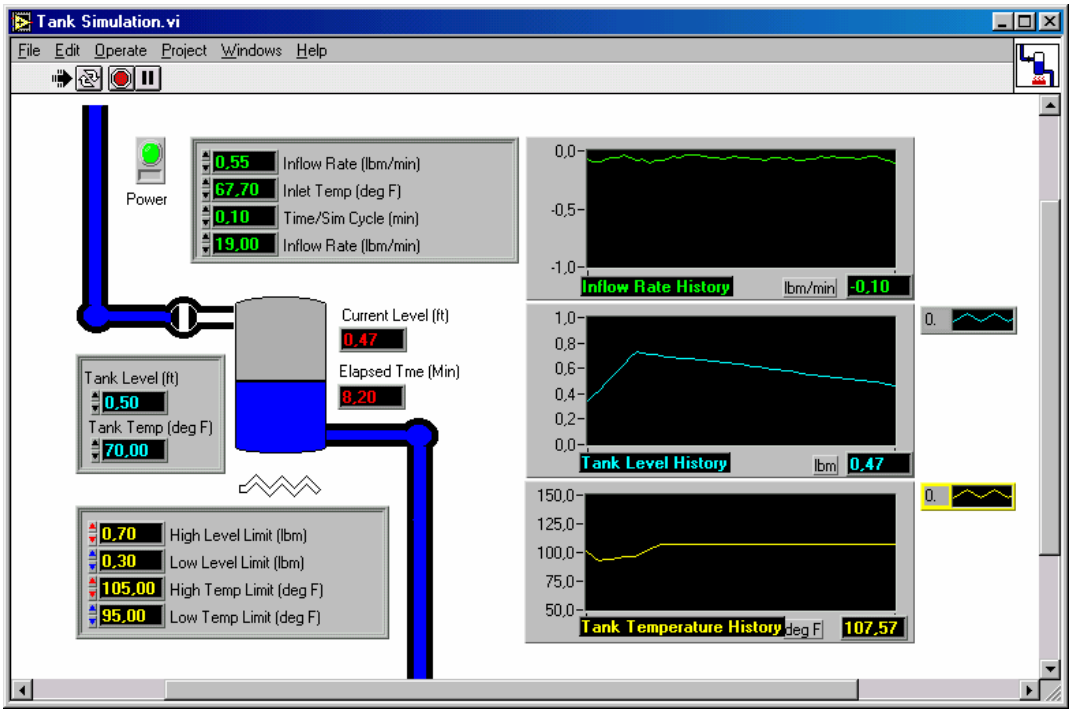


Figura 1.13 Ejemplo de una aplicación en LabVIEW (Panel)

Como se observa, es posible diseñar una interfaz gráfica adecuada para el manejo de un proceso por parte de un operador. En realidad, la interfaz mostrada corresponde a una simulación de un proceso, pero bien podría tratarse de un proceso real.

La figura 1.14 muestra el diagrama correspondiente al panel de la figura 1.13.

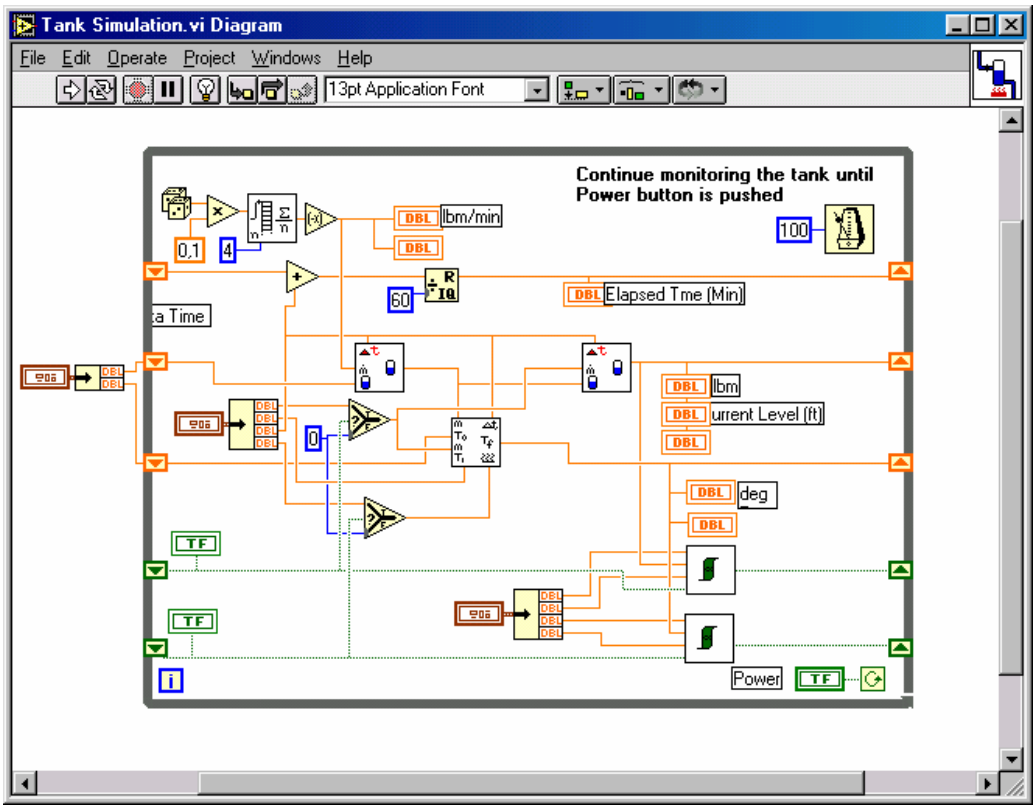


Figura 1.14 Ejemplo de una aplicación en LabVIEW (Diagrama)

Puede observarse claramente que el programa está desarrollado mediante el "cableado" apropiado de diversos componentes gráficos. Es interesante aclarar que algunos de estos componentes (los que aparecen como cuadrados con fondo blanco) son VI's en sí mismos. Haciendo doble click sobre ellos con el ratón, se abre una ventana con el panel correspondiente a dicho VI. Lógicamente, este panel tiene asociado su correspondiente diagrama, en donde se desarrolla el algoritmo del VI. Es decir, a todo VI desarrollado por un usuario se le asigna un icono (cuyo aspecto puede ser dibujado por el usuario) que sirve para utilizar dicho VI en otro VI de nivel superior. De este modo se dice que el LabVIEW es modular, ya que todo VI puede utilizarse como componente dentro de otro VI, y el primero como componente de uno nuevo y así sucesivamente.

1.6 Conectividad entre aplicaciones.

Una característica importante de las aplicaciones modernas es la gran conectividad que existe entre ellas. El sistema operativo Windows ofrece varias posibilidades para la conexión entre aplicaciones. Los sistemas de supervisión actuales explotan estas facilidades. Entre estas facilidades se pueden plantear:

- Atención a puertos de E/S
- Comunicación serial.
- Manejo de ficheros.
- Enlace dinámico de bibliotecas (DDL).
- Ejecución de comandos del sistema
- Conexión a redes, uso del protocolo TCP/IP.
- Uso de drivers especifico para adquisición de datos (DAQ, GPIB).
- Intercambio dinámico de datos (DDE).
- Trabajo con bases de datos (SQL).
- Enlace con objetos (OLE y ActiveX)

1.6.1 Atención a Puertos de E/S.

Muchos software presentan funciones para el trabajo directo con los puertos, pudiendo utilizarse ya sean bytes o words (dos bytes). Esto permite el trabajo con cualquier interfase similar al ensamblador. En el LabVIEW que presenta dos funciones para ello:

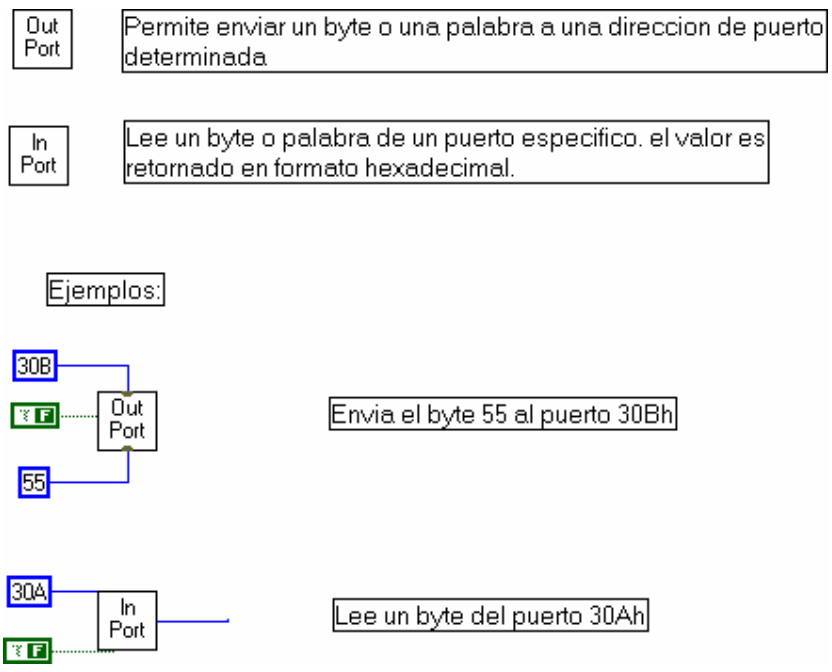


Figura 1.15 Manejo de puertos en LabVIEW.

1.6.2 Comunicación serie

Para la comunicación serial generalmente se tienen gran variedad de funciones. Los parámetros de la comunicación se pueden establecer según se requiera o utilizar los establecidos por defecto. Es común que los sistemas trabajen en tiempo real por interrupción y la información recibida o transmitida se almacena en los buffers de comunicación de manera tal que la aplicación al enviar o transmitir lo hace a través de dichos buffers. Esto hace que se pueda estar ejecutando varias tareas y realizarse la comunicación de forma transparente para las aplicaciones. Toda comunicación requiere establecer un protocolo para que las partes puedan entenderse entre sí.

Muchas aplicaciones poseen sus propios drivers para la comunicación serial con dispositivos específicos como por ejemplo autómatas programables (Lookout, RSVIEW, LabVIEW, Intouch).

Algunas funciones para el manejo de comunicación serial en LabVIEW son:

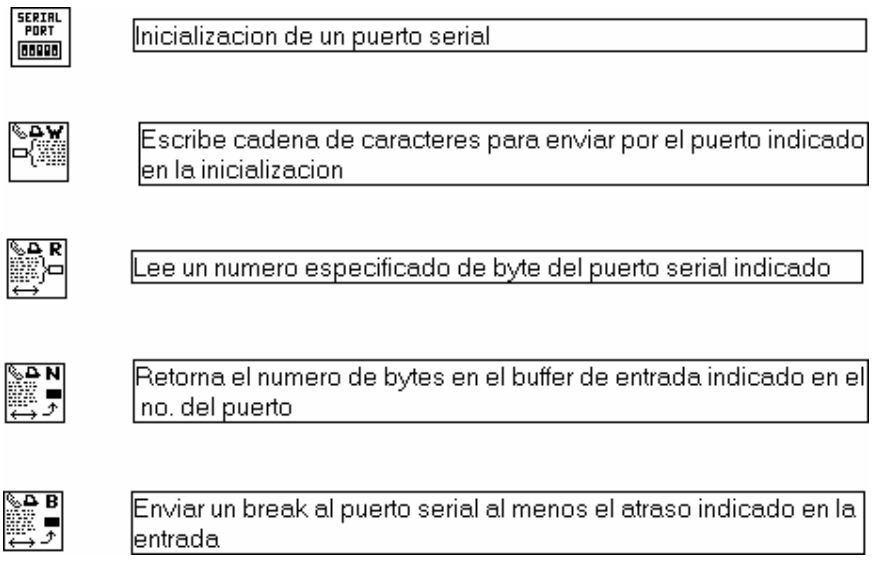


Figura 1.16 Funciones para la comunicación serie en LabVIEW.

1.6.3 Trabajo con ficheros

Todo lenguaje de programación contiene funciones para el trabajo con ficheros, entre las que se encuentran: abrir, leer, escribir, cerrar, copiar, mover información sobre un fichero, etc.

A continuación se presentan como un ejemplo, las funciones para la lectura y escritura de una cadena de caracteres en LabVIEW. La primera función abre el fichero, escribe una cadena de caracteres (al comienzo o se agrega) y luego cierra dicho fichero. La segunda función realiza una tarea similar, pero lee la cadena de caracteres del fichero. El LabVIEW posee una gran variedad de funciones para estos propósitos.

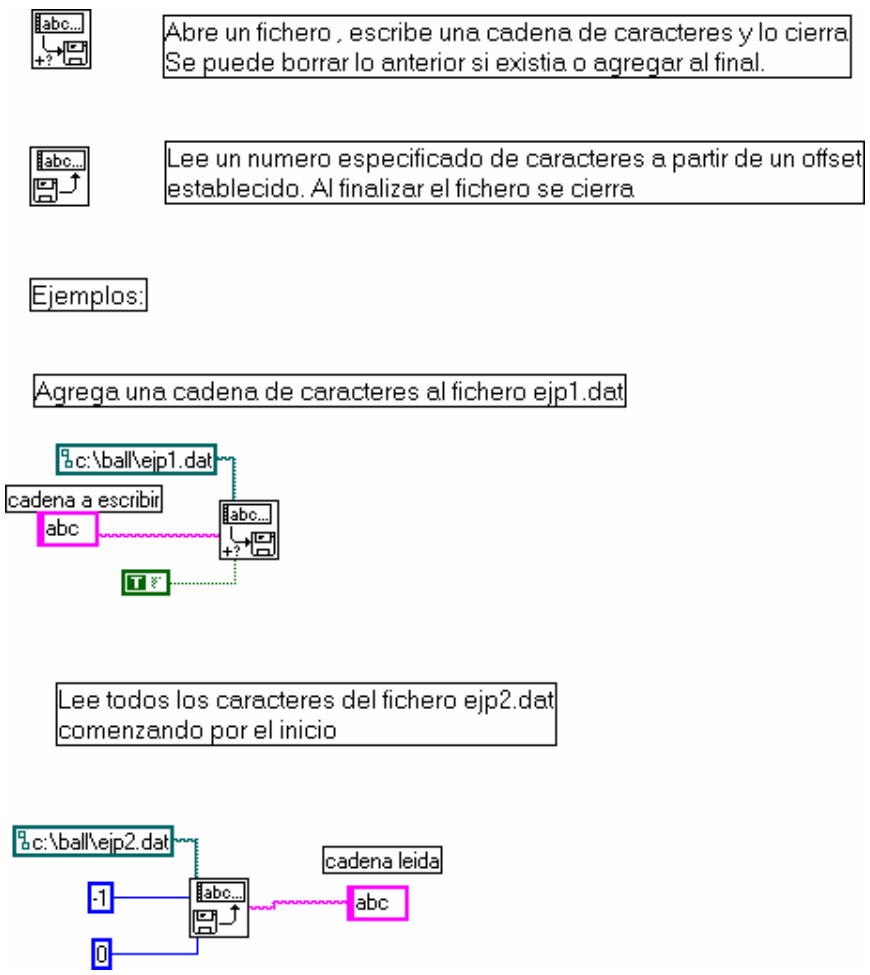


Figura 1.17 Ejemplo de trabajo con fichero en LabVIEW.

1.6.4 Manejo de DLL

El entorno Windows utiliza bibliotecas para almacenar funciones, API's, etc., que pueden ser utilizadas por cualquier aplicación que las requiera. El Enlace Dinámico de Bibliotecas (Dynamic Link Library, DLL), es un mecanismo que enlaza las aplicaciones a estas bibliotecas, en tiempo de ejecución. Las bibliotecas permanecen en su propio fichero y no son copiadas al fichero ejecutable de la aplicación, lo que permite que el tamaño de aplicaciones complejas pueda se reducido. Las DDL's se enlazan con la aplicación cuando se está ejecutando. A su vez las DDL's pueden tener enlaces con otras DDL's.

Las DDL usan un formato que es común a una gran variedad de sistemas, por lo que las aplicaciones sobre windows tendrán acceso a una gran cantidad de funciones realizadas con diferentes software.

En el ejemplo que se plantea mas adelante se utiliza una DDL propia de Windows, para hacer uso de la multimedia desde una aplicación de LabVIEW

La idea es utilizar la función “Call Library Function” propia de LabVIEW para llamar un procedimiento almacenado en una DLL de Windows, para reproducir un archivo tipo .wav a través del sistema de sonido de la computadora. Para poder utilizar la función es necesario darle una serie de parámetros, como son: nombre de la DLL, nombre de la función, el prototipo de la función, etc.

Icono de la función en LabVIEW:

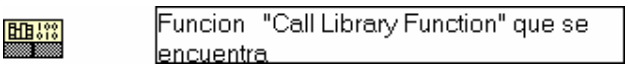


Diagrama de bloques del programa:

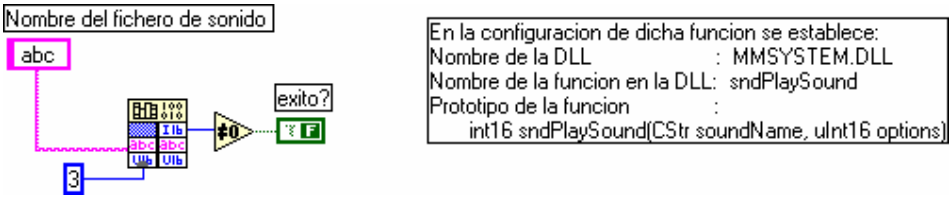


Figura 1.18 Ejemplo de uso de una DLL en LabVIEW.

1.6.5 Ejecución de comandos del sistema operativo:

En muchas aplicaciones, a menudo es necesario ejecutar comandos del sistema operativo o ejecutar otra aplicación. Todos los lenguajes de programación incluye una instrucción para realizar esta tarea; por ejemplo la instrucción RUN en Basic.

En LabVIEW, la función *Exec* realiza esta función:

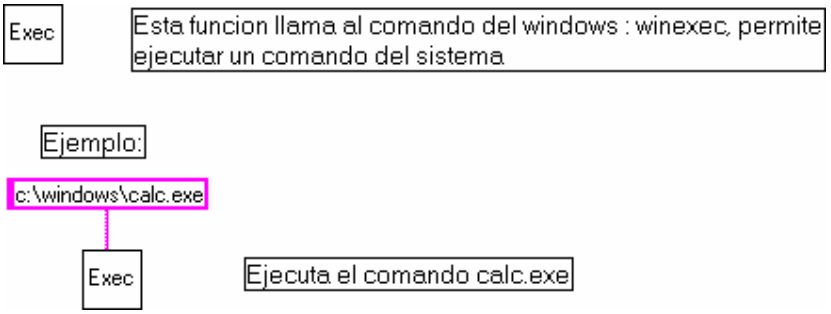


Figura 1.19 Ejecución de comandos del sistema operativo en LabVIEW.

1.6.6 Conexión a redes. Uso de TCP/IP

La conectividad de aplicaciones a través de redes es de fundamental importancia en la actualidad. Prácticamente todos los sistemas de control y supervisión modernos utilizan las redes para la comunicación entre sus distintos componentes. Para establecer la comunicación es necesario utilizar un protocolo de comunicaciones, como por ejemplo el UDP (User Datagram Protocol) y el TCP/IP (Transmision Control Protocol / Internet Protocol).

Con el protocolo UDP, no se necesita establecer la comunicación antes de recibir o enviar datos. La destinación del dato es especificada cuando éste es enviado. Puede enviarse un número limitado de datos en un paquete. Para gran cantidad de datos debe dividirse en secciones. El UDP posee funciones para abrir, leer, escribir, cerrar la comunicación.

El protocolo TCP/IP proviene de los protocolos TCP (Transmision Control Protocol), e IP (Internet Protocol) (y UDP). Su uso está muy extendido ya que es el utilizado en la red Internet. Permite la comunicación entre diferentes máquinas incluyendo el trabajo en Internet. En este protocolo, la comunicación se establece entre un *servidor* y un *cliente*.

Algunas funciones de LabVIEW para la conexión en red utilizando TCP/IP son:

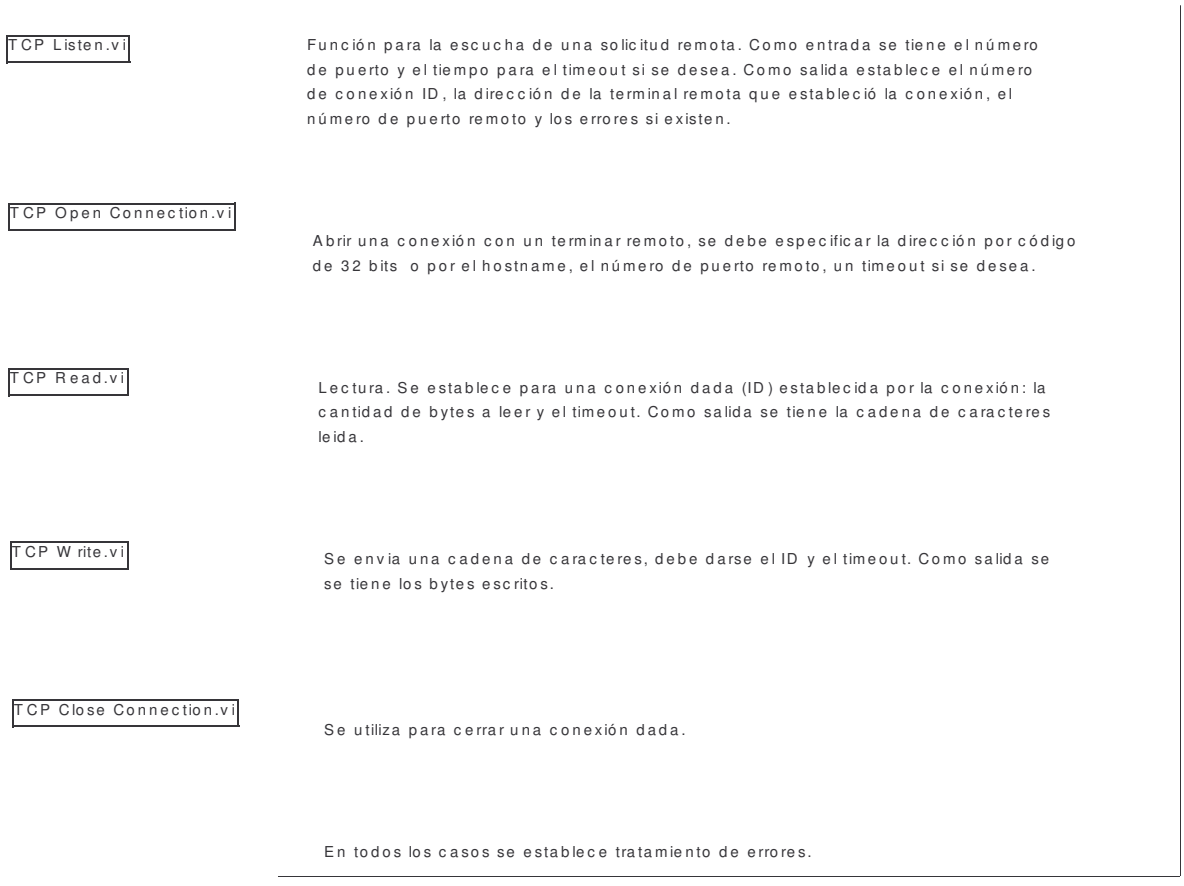


Figura 1.20 Algunas funciones para el trabajo con TCP/IP en LabVIEW.

1.6.7 Adquisición de datos.

Se ha vuelto común la utilización de tarjetas de adquisición de datos insertables en una ranura (ISA o PCI) de un computador, permitiendo una forma ágil de leer los valores de variables discretas o analógicas procedentes de un proceso.

Las aplicaciones para el monitoreo y supervisión poseen drivers para la atención a dichas tarjetas de adquisición de datos, atención a módulos lejanos por protocolos fieldbus, etc. La National Instruments oferta drivers para mas de 150 tarjetas de adquisición de datos de más de 40 firmas. El LabVIEW posee cerca de 100 funciones entre las que se incluye:

- entradas analógicas simples y múltiples,
- salidas analógicas,
- entradas y salidas discretas,
- medición de periodo, frecuencia,
- conteo de pulsos,
- generación de pulsos,
- calibración,
- linealización de termopares,
- lectura de termistores, strain gauge, RTD.

Estas funciones pueden utilizarse para cualquier tarjeta de adquisición de datos, siempre que antes se instale el driver apropiado para dicha tarjeta.

Otros software específicos para supervisión poseen drivers para el trabajo con autómatas, como por ejemplo: Lookout, LabVIEW, RSVIEW, Intouch.

1.6.8 Intercambio dinámico de datos (DDE, Dynamic Data Exchange)

El DDE es un protocolo para el intercambio dinámico de datos entre aplicaciones de Windows que lo permitan. Esto hace que, por ejemplo, se puede enviar datos de forma dinámica, en ejecución, entre dos aplicaciones, como por ejemplo entre RSVIEW, LabVIEW, Microsoft Word, Matlab, Excel, etc.

LabVIEW posee funciones para DDE bajo Windows. Para ello, cada aplicación establece sus Servicios, Tópicos e Item, requeridos para el trabajo sobre DDE.

1.6.9 Manejo de bases de datos

En general, toda aplicación de control requiere el manejo de muchos datos: variables, parámetros, configuraciones, etc. El conjunto de todos estos datos constituyen la base de datos del la aplicación. Por ello, los software para el trabajo en tiempo real poseen facilidades para el trabajo con bases de datos de forma dinámica. La base de datos puede ser interna (o propia) de la aplicación, o pueden utilizarse bases de datos estandarizadas, externas a la aplicación, pero que puede ser accedida por la aplicación. Así tenemos que muchos softwares poseen facilidades para el trabajo con tablas compatibles con las bases de datos standard. Otros utilizan el lenguaje standard SQL (Standard Query Lenguage) para trabajar directamente con bases de datos SQL. Tenemos como ejemplo: el LabVIEW oferta un toolkit para el trabajo hasta con 35 bases de datos, el RSVIEW posee el Watcom SQL.

El trabajo con SQL Server posee la ventaja adicional de trabajar con bases de datos distribuidas con gran nivel de seguridad. El LabVIEW posee esta posibilidad.

1.6.10 Enlace con objetos.

En aplicaciones sobre Windows se puede utilizar métodos y funciones propias de una aplicación, por otra aplicación, logrando con ello una gran conectividad. Con OLE (Object Linking and Embedding) se brinda una gran flexibilidad. Muchos softwares hacen uso de OLE como Matlab, LabVIEW, RSVIEW, Delphi, etc.

En el libro *Inside OLE*, Kraig Brockschmidt, define: OLE es un ambiente unificado de servicios basados en objetos ... con el único propósito de permitir una rica integración entre componentes.

El termino usado para referirse a todos los servicios es *componente*. Para ser más preciso, un servicio esta compuesto de uno o más componentes, un componente lo componen uno o más objetos, donde cada objeto provee su funcionalidad y su contenido a través de una o más interfaces. Estas interfaces contienen uno o más funciones por medio de las cuales uno puede acceder a las cosas que brinda el componente.

En LaBVIEW existen diferentes funciones como: ejecutar método, obtener propiedades, listar métodos, etc. De esta manera, es posible utilizar un objeto de otra aplicación, como por ejemplo Excel, desde una aplicación de LabVIEW.