



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA MECÁNICA ELÉCTRICA

**PLANTEAMIENTO DEL USO DE INSTRUMENTACIÓN VIRTUAL
PARA ADQUISICIÓN DE DATOS**

Mario Edgar Luis Tarot Gálvez
ASESORADO POR ING. GUSTAVO ADOLFO VILLEDA VÁSQUEZ

GUATEMALA, OCTUBRE DE 2004

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

PLANTEAMIENTO DEL USO DE INSTRUMENTACIÓN VIRTUAL PARA ADQUISICIÓN DE DATOS

TRABAJO DE GRADUACIÓN
PRESENTADO A JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

MARIO EDGAR LUIS TAROT GÁLVEZ

ASESORADO POR ING. GUSTAVO ADOLFO VILLEDA VÁSQUEZ

AL CONFERÍRSELE EL TÍTULO DE
INGENIERO ELECTRÓNICO

GUATEMALA, OCTUBRE DE 2004

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA

DECANO	Ing. Sydney
Milson	
VOCAL I	Ing. Recinos
VOCAL II	Lic.
VOCAL III	Ing.
VOCAL IV	Br. Ruiz
VOCAL V	Br. Leiva
SECRETARIO	Ing. Pedro Antonio Aguilar Polanco

DIRECTIVA

Alexander Samuels
Murphy Olympo Paiz
Amahán Sánchez Álvarez
Julio David Galicia Celada
Kenneth Issur Estrada
Elisa Yazminda Vides

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

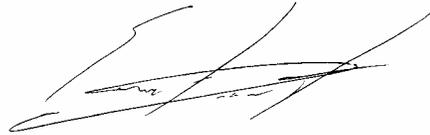
DECANO	Ing. Sydney Alexander Samuels Milson
EXAMINADOR	Ing. Enrique Edmundo Ruiz Carballo
EXAMINADOR	Ing. Jorge Fernando Álvarez Girón
EXAMINADOR	Ing. Erwin Efraín Segura Castellanos
SECRETARIO	Ing. Pedro Antonio Aguilar Polanco

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

PLANTEAMIENTO DEL USO DE INSTRUMENTACIÓN VIRTUAL PARA ADQUISICIÓN DE DATOS

Tema que me fuera asignado por la dirección de la Escuela de Ingeniería Mecánica Eléctrica con fecha del 26 de septiembre de 2001.



Mario Edgar Luis Tarot Gálvez.

AGRADECIMIENTOS

**A DIOS Y A LA
VIRGEN MARÍA**

Por el regalo de la vida, y por todas las bondades recibidas en ella.

A MIS PADRES

Por su apoyo, consejos, y su esfuerzo para que triunfe en todos los momentos de mi vida.

A MIS HERMANOS

Por la unión que siempre nos ha caracterizado.

A MIS HIJOS

Por hacerme tan dichoso.

A MI ESPOSA

Por su cariño, nobleza y su apoyo incondicional.

A MIS ABUELOS

Por guiarnos a mí y a mi familia por el camino del bien.

A MI FAMILIA

Por su aporte a mi desarrollo integral.

A MIS AMIGOS **Por ser parte de mi vida.**

A MI ASESOR ***Por su colaboración para el desarrollo de este trabajo.***

A LA FACULTAD ***Por la formación que he recibido.***
DE INGENIERÍA

DEDICATORIA

A DIOS Nuestro señor.

A LA VIRGEN MARÍA Madre de Dios.

A MIS PADRES María Olga Galvez Aldana de Tarot y
Mario Eugenio Tarot Lanza
Con especial cariño por todos sus sacrificios.

A MIS HERMANOS Evelyn Jhannette
Mario Eugenio
Lidia Carmelina

A MIS HIJOS Anais y Jafet. Para que sea un parámetro en su vida.

A MI ESPOSA Karla A. Chavarría M. de Tarot

A MIS SOBRINOS Evelyn María Regina (QEPD). Para mi angelito con mucho
cariño de tu tío EKA.
Carlos Efraín
Ana María del Rosario
Mario Eugenio

Para que sea un ejemplo a seguir.

A MIS ABUELOS

José Domingo Tarot (QEPD)
Carmelina Lanza de Tarot
Gregorio Gálvez (QEPD)
Lidia Aldana (QEPD)

A MI FAMILIA

Luis Rodolfo, Guísela
Primos y primas
Byron Gálvez
Tíos y tías

A MI CUÑADO

Carlos E. Chacón González.

A MI CUÑADA

Wendy Chavarria Montesflores

A MIS AMIGOS

Juan Carlos Flores Mayen (QEPD). Viejo, te dije que algún día iba a ser Ingeniero.

Heber Meda, Jaime Sepulveda, Guillermo de la Vega

Angel y Anael Argueta, Arnoldo García, Rodolfo López Vega.

Kenneth Estrada, Estuardo Tello, Romeo Muñoz, Erick Reyes,

Rualdo Soto, Erick Cruz, Harold Letona, Mario Guillen, Víctor

Sandoval, Abel Gómez, José María Del Cid, Jimmy González ,

Sergio Juárez, Guillermo Cotto, Carlos Muñoz Wung, Obed

Ajanel, Sergio González, Hector Ovalle, Edie Esau González,

Juan F. Catalán, Ferdi Rodríguez, Ivan Morales.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	VI
LISTA DE SÍMBOLOS.....	IX
GLOSARIO.....	X
RESUMEN.....	XIX
OBJETIVOS.....	XXI
INTRODUCCIÓN.....	XXIII
1. INSTRUMENTOS TRADICIONALES VERSUS VIRTUALES.....	1
1.1 Instrumentos tradicionales.....	1
1.1.1 Sistemas de instrumentación.....	1
1.1.2 Sistemas de medida.....	1
1.1.3 Sistemas electrónicos básicos de medida.....	2
1.1.4 Sistemas de medida de datos múltiples.....	2
1.1.5 Sistemas de telemetría.....	3
1.1.6 Sistemas de control.....	3
1.2 Instrumentos virtuales.....	6
1.3 Comparación de instrumentación tradicional versus virtual.....	8
1.3.1 Beneficios de los instrumentos basados en <i>PC</i>	11
1.3.1.1 Mayor rapidez en las pruebas.....	11
1.3.1.2 Menor costo de inversión.....	11
1.3.1.4 Integración simplificada del sistema.....	11
2. INSTRUMENTACIÓN VIRTUAL.....	13

2.1 Construcción de instrumentos virtuales.....	14
2.1.1 Estrategia base para diseños de instrumentos virtuales.....	16
2.1.2 Estrategia para implementar un sistema de instrumentación.....	
virtual.....	17
2.2 Patrones, calibración, precisión de la instrumentación virtual.....	18
2.2.1 Clasificación de los patrones.....	18
2.2.2 Calibración.....	19
2.2.2.1 Calibración de las tarjetas de adquisición de datos.....	20
2.3 Aplicación en la industria en general.....	21
2.3.1 Aplicaciones de prueba y medición.....	21
2.3.2 Aplicaciones de automatización industrial.....	22
2.3.3 Aplicaciones en las telecomunicaciones.....	22
2.4 Aplicación didáctica.....	23
2.4.1 Características de las aplicaciones.....	23
2.4.2 Entrenadores didácticos basados en instrumentación virtual.....	24
2.4.3 Potenciación de entrenadores existentes.....	25
2.4.4 Entrenador de electricidad básica asistido por.....	
computadora.....	26
2.5 Sistemas de computación.....	26
2.5.1 Plataforma de computación y tipo de sistema.....	26
3. SOFTWARE ASOCIADO A LA INSTRUMENTACIÓN VIRTUAL.....	29
3.1 <i>Software</i>	29
3.2 PROGRAMACIÓN TRADICIONAL VERSUS	
PROGRAMACIÓN GRÁFICA.....	31
3.2.1 Programación tradicional.....	31
3.2.2 Programación gráfica.....	32

3.3	VVE.....	33
3.4	Labview	36
3.4.1	Características de Labview.....	38
3.4.2	Adquisición de datos con Labview.....	39
3.5	Lookout.	41
3.5.1	Características de Lookout.....	43
	3.5.1.1 Arquitectura basada en	
	objetos.....	43
	3.5.1.2 Conexión en red.....	43
	3.5.1.3 Listo para internet.	43
3.6	LabWindows/cvi.....	45
3.6.1	Características de Labwindows/cvi.....	46
3.7	Interface con autómatas.....	47
3.7.1	Simatic WinCC.....	47
3.7.1.1	Características de WinCC.....	48
3.7.1.2	Requisitos previos del entorno de trabajo de WinCC....	49
3.7.1.3	Interfaces con WinCC.....	50
3.7.2	Rsview.....	52
3.7.2.1	Características adicionales de RSView32.....	54
3.7.2.2	Aplicaciones con RSView32.....	55
3.8	Características económicas del software asociado a la instrumentación... Virtual.....	57
3.8.1	Análisis económico de WinCC.....	57
3.8.2	Análisis económico de RSView.....	59
3.8.3	Resumen económico de Labview.....	61
3.8.4	Resumen económico de Lookout.....	62
4.	EJEMPLO PRÁCTICO DE INSTRUMENTACIÓN VIRTUAL.....	63
4.1	Uso de la computadora para crear sistemas de registro de datos.....	64
4.1.1	Partes de un sistema de registro de datos.....	65
5.	ADQUISICIÓN DE DATOS Y ACONDICIONAMIENTO DE SEÑALES.....	67
5.1	Adquisición de datos.....	67
5.1.1	Sistema de adquisición de datos local.....	67

5.1.2 Sistema de acondicionamiento de señales para tarjetas de.....	
adquisición de datos.....	71
5.1.3 Sistema de adquisición de datos remoto.....	73
5.2 Acondicionamiento de transductores.....	75
5.2.1 Acondicionamiento de señales.....	76
5.2.2 Aislamiento de señales.....	77
5.2.3 Amplificación de señales.....	77
5.2.4 Atenuación de señales.....	77
5.2.5 Multiplexión de señales.....	78
5.2.6 Filtrado de señales.....	78
5.2.7 Excitación de transductores.....	78
5.2.8 Linearización de señales.....	79
5.2.9 Compensación de junta fría de transductores.....	79
5.2.10 Muestreo simultaneo de señales.....	79
5.3 Tipos de transductores utilizados comúnmente.....	80
5.3.1 Termopares ó termocuplas.....	80
5.3.2 <i>RTD</i>	82
5.3.3 Galgas extensiométricas.....	83
5.3.4 Acelerómetros.....	85
5.3.5 Señales de corriente.....	86
6. SIMULACIÓN Y CONTROL DE PROCESOS CON PROGRAMACIÓN.....	
GRÁFICA.....	87
6.1 La simulación de sistemas de control en la empresa.....	91
6.2 Sistemas que utilizan programación gráfica.....	95
6.3 Simulación de sistemas.....	98
CONCLUSIONES.....	105
RECOMENDACIONES.....	107
BIBLIOGRAFÍA.....	109

ÍNDICE DE ILUSTRACIONES

Figuras

1.	Ejemplo de programación gráfica.....	32
2.	Objeto de dialogo de archivo <i>VEE</i>	34
3.	Modelo de <i>VEE</i>	34
	4. <i>Panel de interfase con el usuario en</i> <i>VEE</i>	35
5.	Diagrama de bloques y su respectivo panel frontal.....	36
6.	Instrumento virtual realizado con <i>Labview</i>	37
7.	Monitoreo remoto y control.....	38

8.	Diagrama de bloque.....	38
9.	Ejemplos de instrumentos virtuales.....	39
10.	Captura de una forma de onda senoidal por medio de Labview.....	40
11.	Aplicaciones de <i>Lookout</i>	41
12.	Ejemplo de control de proceso industrial.....	41
13.	Panel frontal de <i>Labwindows/cvi</i>	45
14.	Gráfica de muestreo de temperatura.....	46
15.	Sistema cliente / servidor usando <i>WinCC</i>	51
16.	Arquitectura con <i>OPC</i>	52
17.	Conectividad usando <i>Rsview</i>	54
18.	Pantalla de <i>HMI/SCADA</i> utilizando <i>Rsview</i>	54
19.	Adquisición de datos en una hoja de cálculo.....	63
20.	Adquisición de datos en una planta de producción.....	66
21.	MODULO DE ADQUISICIÓN DE DATOS.....	68
22.	Sistema <i>Fieldpoint</i>	70
23.	Sistema de la serie 6B.....	70
24.	Ejemplo de adquisición de datos remoto.....	73
25.	Acondicionamiento de señales.....	75
26.	<i>Termopar</i>	81
27.	RTD.....	83
28.	Conexión de galgas extensiométricas.....	84
29.	Galgas extensiométricas.....	84
30.	Acelerómetros.....	85
31.	Señales de corriente.....	86
32.	Porcentaje de presupuesto contra tiempo.....	88
33.	Áreas de una industria.....	92
34.	Tipos de señales eléctricas.....	92

35.	Simulación de un circuito	
<u>RLC.....</u>	<u>94</u>	
36.	Simulación de un proceso <i>PID</i>	95
37.	Diagrama de bloques de programación	
gráfica.....	96	
38.	Programación	
gráfica.....	97	
39.	Simulación de	
sistemas.....	99	
40.	Ejemplo de un sistema en una	
industria.....	100	
41.	Sistema de	
control.....	101	

TABLAS

I.	Instrumentos tradicionales vrs. virtuales.....	8
II.	Características de medición.....	10
III.	Plataformas de instrumentos basados en	
computadora.....	12	
IV.	Definiciones utilizadas en	
instrumentación.....	19	
V.	Prueba y	
medición.....	21	

VI.	Automatización
<u>industrial</u>	<u>22</u>
VII. Aplicaciones de telecomunicaciones.....	22
VIII. Tipos de computadora.....	26
IX. Sistemas operativos.....	27
X. Tipo de bus o canal de comunicación.....	27
XI. Ejemplo de configuración de sistema.....	28
XII. Algunos programas para instrumentación virtual.....	30
XIII. Características de <i>VEE</i>	33
XIV. Lista de funciones de <i>Lookout</i>	44
XV. Aplicaciones con <i>WinCC</i>	49
XVI. Configuración de sistema.....	50
XVII. Precios de <i>WinCC</i> según los tags requeridos.....	58
XVIII. Características importantes del <i>hardware</i> de medición.....	65
XIX. Opciones disponibles para un sistema de adquisición de datos local...	68
XX. <i>Algunas de las opciones de sistemas disponibles</i>	72
XXI. <i>Comparación de transductores de temperatura</i>	83

LISTA DE SÍMBOLOS

A	Ampere, unidad de corriente
° C	Centígrado, unidad de temperatura
dB	Decibel
Hz	Hertz, un ciclo por segundo
K	Kilo (1×10^3)
M	Mega (1×10^6)
MA	<i>MICROAMPERE</i> (1×10^{-6})
mA	Miliampere (1×10^{-3})

mV	<i>MV</i>	<i>MICROVOLTIO (1X10⁻⁶)</i>
Ω		Milivoltio (1X10 ⁻³)
V		Ohm, unidad de resistencia
		Voltio, unidad de voltaje

GLOSARIO

Acelerómetro	Dispositivo utilizado para medir aceleración y vibración.
Active x	Tecnología que permite reutilizar código (escrito en cualquier lenguaje) que tiene una interfase definida.
Ansi c	Lenguaje de programación C estándar.
Api	Interfaz de programación de aplicación.
Ascii information	<i>American standard code for interchange</i> Código estándar americano para intercambio de información.
Ate	<i>Automated test equipment.</i> Sistema de prueba automática.
Bit	<i>Binary digit.</i> Dígito binario.
Byte	Cadena de ocho <i>bits</i> .
C++	Lenguaje de programación.
Compensación de junta fría	<i>Nivel artificial de referencia que compensa las</i> variaciones de temperatura ambiente en las mediciones de termocoplas.
Daq	<i>Data acquisition.</i> Recopilar información de las fuentes de medida por medio de sensores y transductores.

Data logger Instrumento que registra y/o traza uno o más canales de datos análogos o digitales.

Dcs *Distributed control systems.* Sistemas de control distribuido.

Dee Intercambio dinámico de datos.

Dll *Data link library.*

Driver Controlador que permite gestionar los periféricos que están conectados a la computadora.

Ethernet Red de área local de medios compartidos. Todos los mensajes se diseminan a todos los nodos en el segmento de red.

Fieldpoint Sistema modular, distribuido e inteligente de entradas y salidas para aplicaciones de monitorización y control.

Filtro paso bajo frecuencia *Circuito que atenúa los componentes de alta frecuencia en una señal analógica y sólo pasan las frecuencias bajas.*

Foundation fieldbus *Sistema serial de comunicaciones digital, bidireccional y multicada.*

Fuzzy logic Lógica difusa. Técnica matemática que se utiliza para resolver problemas que tienen varias soluciones en lugar de una sola, en sistemas de tiempo real en condiciones volátiles o imprevisibles.

Galga extensiométrica Dispositivo comúnmente usado en pruebas y mediciones mecánicas.

Galga varia extensiométrica de resistencia	Matriz de bobinas o cable muy fino el cual su resistencia linealmente dependiendo de la carga aplicada al dispositivo.
Gpib	<i>General purpose interface bus</i> . Puerto diseñado por <i>Hewlett Packard</i> , para establecer comunicación con instrumentos de medición.
Hardware	Se trata de todos los componentes físicos de una computadora.
Hmi máquina.	<i>Human machine interface</i> . Interfaz hombre
Html estructurar documentos.	<i>Hyper text markup language</i> . Sistema para
Ip	<i>Internet protocol</i> . Protocolo <i>internet</i> . Conjunto de reglas que regulan la transmisión de paquetes de datos a través de <i>internet</i> .
Instrumento	Dispositivo para determinar el valor o la magnitud de una cantidad o variable.
Instrumento virtual	Capa de <i>software</i> y <i>hardware</i> que se le agrega a una computadora en tal forma que permite a los usuarios interactuar con la computadora como si estuviesen utilizando su propio instrumento electrónico hecho a la medida.
Labview	Lenguaje de programación de alto nivel, de tipo gráfico, y enfocado al uso en instrumentación.

LABWINDOWS/CVI
SISTEMAS DE

AUTOMÁTICA PARA
APLICACIONES DE PRUEBA Y MEDIDA,
SUPERVISIÓN Y CONTROL DE
PROCESOS.

ENTORNO DE DESARROLLO DE

INSTRUMENTACIÓN DE PRUEBA
GENERAR

ADQUISICIÓN DE DATOS,
PROCESOS.

Lvdt *Linear voltage differential transformer.* Transformador de voltaje lineal diferencial. Utilizado para medir desplazamiento lineal.

Medidor *Consiste en una bobina que apoyada sobre cojinetes,*
D'arsonval puede girar debido al torque que sufre dentro de un campo magnético uniforme al circular la corriente continua a través de la misma.

Multiplexor Conjunto de semiconductores o interruptores electromecánicos en un arreglo para seleccionar una de varias entradas a una sola salida.

Ole *Object linking and embedding.*

Opc *Ole process controlled.*

Osi *Open system interconnect.*

Pc *Personal computer.* Computadora personal.

Pci *Peripheral component interconnect.* Bus local desarrollado por Intel.

Pcmcia Asociación internacional de tarjetas de memoria para ordenadores personales. Tarjeta estandarizada de expansión, del tamaño de una tarjeta de crédito, para ordenadores personales. En telecomunicaciones uno de sus principales usos es la transmisión de mensajes, datos, faxes, etc.

Pid *Proportional, integral, derivative*. Proporcional, integral, derivativo. Es un sistema de control que elimina la continua atención del operador ya que por medio de retroalimentación se trata de mantener el punto de ajuste.

Plc Controlador lógico programable.

Plug and play Característica del sistema operativo de una computadora para reconocer los dispositivos *hardware* a él conectados y ponerlos en funcionamiento de forma rápida y sencilla.

Puerto paralelo Interfase estándar multipin para computadoras, bidireccional y de alta velocidad.

Puerto serie Dispositivo asíncrono utilizado para transmisión y recepción de datos a través de distancias relativamente grandes, compatible con las normas RS232.

Pxi *PCI extensions for instrumentation*. Extensiones para instrumentación del bus *pci*.

Router Enrutador. Dispositivo que distribuye tráfico entre redes.
La decisión sobre a donde enviar los datos se realiza en base a información de nivel de red y tablas de direccionamiento.

Rs 232 *Recommended Standard 232*. Estándar de "Electronic Industries Association" (EIA) que especifica las características eléctricas, requerimientos de conector, y funciones de señal para una interfase serial.

Rs 485 *Recommended standard 485*. Ampliamente utilizado para adquisición de datos. Soporta 32 transmisores y 32 receptores, pueden conectarse con 2 cables o cuatro.

Rtd *Resistance-temperature detector*. Dispositivo que incrementa su resistencia con la temperatura.

Rtu *Remote terminal unit*. Unidad terminal remota.

Software Programa que se ejecuta en la computadora y que controla los elementos *hardware* y procesa los datos.

Scada Sistema de control y supervisión de adquisición de datos.

Scxi Extensiones de acondicionamiento de señal para la Instrumentación.

Strain Gages Véase galgas extensiométricas.

Teorema de Nyquist	La digitalización de una señal analógica ha de hacerse muestreando al menos al doble de la frecuencia máxima que se pretende capturar.
Termocopla	Transductor de temperatura que consiste en dos alambres de diferentes metales unidos entre si.
Tcp/ip	<i>Transmission control protocol/ internet protocol.</i>
Transductores	Dispositivos que convierten fenómenos físicos como temperatura, carga, presión o luz a señales eléctricas como voltaje y resistencia.
Ttl	Lógica transistor – transistor.
Uir	Recurso para interfaz de usuario.
Usb	<i>Universal serial bus.</i> Bus serial universal.
Vee	<i>Visual engineering environment.</i>
Visa	<i>Virtual instrument software architecture.</i> Software para arquitectura de instrumentación virtual.
Visual basic (Beginners en la	Lenguaje de programación basado en <i>Basic all purpose symbolic instruction code</i>) y programación orientada a objetos.
Vxi	<i>VME extensions for instrumentation.</i>

RESUMEN

Es necesario que los ingenieros conozcan los fundamentos básicos de los instrumentos industriales, los principios en que se basan, sus funciones y sus características, para hacer una correcta aplicación de ellos, tanto en los procesos industriales tradicionales o modernos en los que se desempeñe.

Por otra parte, es importante el conocimiento de la instrumentación virtual para familiarizar al estudiante con los conceptos modernos de instrumentación basados en instrumentos virtuales *VI* que procesan las señales del mundo real de acuerdo al tipo de análisis y diseño lógico que éste haya programado.

La instrumentación basada en computadora es la selección correcta para aumentar la productividad y reducir el costo de las mediciones. A medida que trabaja para extender los alcances de las pruebas, mientras encara presupuestos cada vez más reducidos, la selección de la instrumentación se torna muy importante.

Los instrumentos basados en computadora aumentan el rendimiento y reducen el costo de las pruebas, mediciones y aplicaciones de adquisición de datos.

**LOS INSTRUMENTOS BASADOS EN COMPUTADORA
SON MODULARES Y SE INTEGRAN FÁCILMENTE PARA**

SATISFACER LOS REQUISITOS PARTICULARES DE PRUEBAS.

Considerando que en nuestro país la instrumentación virtual es un tema que no se conoce a fondo, este trabajo de graduación dará a conocer a los estudiantes la importancia que tiene el uso de esta tecnología.

OBJETIVOS

General

- Proponer el uso de la instrumentación virtual para la adquisición de datos.

ESPECÍFICOS

1. Familiarizarse con los conceptos de instrumentación virtual.
2. Dar a conocer las ventajas y desventajas del uso de la instrumentación virtual, referentes a utilización, precisión, fiabilidad, etc.
3. Introducirse en la programación gráfica mediante un lenguaje de última generación.
4. Mostrar ejemplos del uso de instrumentación virtual.
5. Resaltar las ventajas de la implementación de la instrumentación virtual y mostrar el costo aproximado del mismo.

INTRODUCCIÓN

Ya sea que se requiera un sistema adaptable que simule los instrumentos tradicionales o que se busque una forma eficaz de modernizar un proceso de automatización, se puede tener mejores soluciones implantando un sistema basado en computadora.

En la actualidad el uso de la instrumentación estándar o tradicional como: osciloscopios, multímetros, generadores de datos, registradores de datos, controladores de temperatura y presión, entre otros tienen muchas limitaciones, como el rango de operación, el número de canales, la resolución, precisión, etc. Por lo que la funcionalidad de estos tipos de instrumentos es definida por el fabricante del equipo, y no por el usuario mismo.

Los sistemas basados en computadora *hardware & software* pueden efectuar las mismas tareas que los instrumentos y las herramientas de automatización tradicionales, con los beneficios adicionales de una mayor adaptabilidad y eficacia.

Con mínimas limitaciones dado el caso que se requiera un sistema de muy alto riesgo en el que se involucren procesos vitales que pongan en peligro la vida de las personas. En cuyos casos se podría tener un sistema diseñado con cierta redundancia y sistemas que cubran fallas energéticas.

Cuando se opta por un sistema basado en computadora, se define la función y combina la plataforma de computación, el *hardware*, *software* y todos los accesorios necesarios para su aplicación específica. Una vez que se ha creado un sistema, es fácil modificarlo y configurarlo para satisfacer aplicaciones diferentes o hacer cambios en la aplicación existente sin tener que comprar nuevos equipos.

La eficiencia y adaptabilidad ofrecida por un sistema basado en computadora no puede ser igualada por instrumentos tradicionales de funciones fijas, definidas por sus proveedores.

CON CADA NUEVO AVANCE EN LA TECNOLOGÍA CONVENCIONAL DE COMPUTACIÓN, LA MEDICIÓN Y LA AUTOMATIZACIÓN BASADAS EN COMPUTADORA SE VUELVEN MÁS DINÁMICAS Y PODEROSAS.

El constante mejoramiento del precio y el rendimiento de las computadoras actuales hace que los instrumentos basados en la misma tecnología sean más accesibles a los usuarios que estaban previamente limitados por métodos inadaptables y costosos de medición y automatización.

1. INSTRUMENTOS TRADICIONALES VERSUS VIRTUALES

1.1 Instrumentos tradicionales

El proceso de medición generalmente requiere el uso de un instrumento como medio físico para determinar la magnitud de una variable. Los instrumentos constituyen una extensión de las facultades humanas y en muchos casos permiten a las personas determinar el valor de una cantidad desconocida la cual no podría medirse utilizando solamente las facultades sensoriales.

1.1.1 Sistemas de instrumentación

Los sistemas de instrumentación están ampliamente categorizados tanto en sistemas de medida como en sistemas de control.

En un sistema de medida una magnitud es medida y su valor es convenientemente visualizado.

En los sistemas de control la información acerca de la magnitud que está siendo medida es usada para controlar la magnitud, de manera que su valor medido iguale un valor deseado. El valor medido puede ser visualizado o no.

1.1.2 Sistemas de medida

El sistema más sencillo consiste en un dispositivo de medida que, además, visualiza el valor medido.

Si se desea indicar el valor medido en un punto situado a cierta distancia del punto de medida, es preciso introducir un medio de transmisión de información entre el sistema de medida y el sistema de visualización.

Los sistemas de transmisión de medidas remotas más populares son los sistemas de medida electrónicos.

1.1.3 Sistemas electrónicos básicos de medida

- a) El transductor (sensor) convierte una magnitud física en una magnitud eléctrica en su salida.
- b) Acondicionador de señal que transforma la señal del transductor en una magnitud eléctrica adecuada a las exigencias de entrada del dispositivo visualizador.
- c) Fuente de alimentación que proporciona la energía eléctrica necesaria y adecuada para el circuito.
- d) Dispositivo visualizador que despliega la información sobre la magnitud a medir.

1.1.4 Sistemas de medida de datos múltiples

En este tipo de sistemas se pueden utilizar transductores del mismo tipo o de distinto tipo. El sistema más simple es aquel en que el transductor que se desea leer es seleccionado manualmente, por medio de un conmutador rotativo. Si se quiere visualizar una serie de magnitudes medidas en un monitor durante cortos intervalos de tiempo se puede conseguir mediante un seleccionador o secuenciador automático que explore las salidas de los transductores, y el visualizador incluye un número que identifica al transductor leído. La visualización simultánea de varias medidas puede realizarse mediante un registrador multicanal.

1.1.5 Sistemas de telemetría

Este término está usualmente reservado para los sistemas de medida múltiple que utilizan una portadora de alta frecuencia modulada para transmitir la información desde un punto a otro. Este modo se suele transmitir vía radio o también se puede transmitir mediante cable. En este sistema se utilizan multiplexores y demultiplexores.

La modulación de portadora, es la manera en que la señal portadora transmisora es modulada. En la modulación de amplitud, la frecuencia permanece constante, mientras que la amplitud varía con la señal modulada. En la modulación en fase, tanto la amplitud como la frecuencia permanecen constantes siendo su fase la que varía con la señal modulada. En la modulación en frecuencia, la amplitud de una portadora permanece constante en tanto que su frecuencia varía con la señal de moduladora.

1.1.6 Sistemas de control

Los sistemas de control son utilizados para controlar un proceso de algún sistema. Puede ser de forma automática o de forma manual. Por ejemplo:

- Control de todo / nada, como puede ser el controlar la temperatura de un horno doméstico.
- Control proporcional, como puede ser el mantener el interior de una nave de secado a una temperatura constante, precisa y sin variaciones.
- Control derivativo, en donde la salida del controlador es proporcional a la velocidad de cambio de la entrada.
- Control integral, en donde la velocidad de cambio de la salida es proporcional a la entrada.

Los instrumentos indicadores electromecánicos tales como:

- Galvanómetro de suspensión.
- Amperímetros de CD.
- Multímetro.
- Electrodinamómetros en mediciones de potencia.
- Termo instrumentos.
- Watthorímetro.
- Medidores de factor de potencia, etc.

Entre otros, utilizan el movimiento de un medidor electromagnético para medir voltaje, corriente, resistencia, potencia etc. Aunque puentes y multímetros hacen uso de componentes eléctricos para estas mediciones, los instrumentos descritos no utilizan amplificadores para incrementar la sensibilidad de las mediciones.

La parte principal de los instrumentos indicadores electromecánicos es el medidor *D'Arsonval*, que no se puede construir con una sensibilidad a escala completa menor de $50\mu\text{A}$.

El instrumento electrónico se basa en principios eléctricos o electrónicos para realizar una medición. Un instrumento electrónico puede ser un aparato relativamente sencillo y de construcción simple, como por ejemplo:

- El medidor de corriente directa.
- Voltímetro de CA.
- Multímetro electrónico.
- Voltímetro vectorial, etc.
- Medidor del vector de impedancia.

Sin embargo, el desarrollo de la tecnología, demanda la elaboración de mejores instrumentos y más exactos. Ésta se ha incrementado, produciéndose nuevos diseños y aplicaciones de instrumentos. Para optimizar el uso de estos dispositivos se necesita entender sus principios de operación y valorar la importancia de las aplicaciones deseadas.

La base de los instrumentos tradicionales de medición está en las consideraciones de lo que se conoce como teoría electromagnética, esta teoría nos muestra todas las aplicaciones y deducciones que se pueden obtener de ella para eventualmente llevar a cabo aplicaciones.

El uso de la instrumentación y comprender el funcionamiento de los instrumentos de mediciones comunes es de suma importancia para una gran variedad de personas, técnicos, ingenieros, etc. ya que es la base para la toma de decisiones de determinado proceso.

Hoy la mayor parte de estos instrumentos ha quedado casi obsoleto, esto, por el advenimiento de los nuevos equipos electrónicos y la miniaturización que ha venido a reemplazarlos.

1.2 Instrumentos virtuales

La industria de la instrumentación está sufriendo importantes cambios como resultado de la revolución de las computadoras personales. Estos cambios están ocurriendo tanto en el componente hardware como en el software. Un elevado número de científicos e ingenieros en todo el mundo usan computadoras personales para automatizar sus tareas de investigación, diseño y fabricación.

En este sentido, las compañías de instrumentación *National Instruments Corporation, Hewlett Packcard, Tektronix*, están introduciendo en el mercado nuevas herramientas que están revolucionando las interfaces de programación y el usuario. Ejemplos de estas herramientas son *VEE* de *HP* y *LabVIEW* y *LabWindow* de *National Instruments*.

Este desarrollo *hardware* y *software* de la *PC*, ha dado lugar a que aparezca un nuevo tipo de instrumentación, que es la denominada instrumentación virtual.

El término virtual, empleado en la industria de la instrumentación, hace referencia a la combinación de instrumentos programables con las computadoras personales.

Un instrumento virtual se define como un nivel de *software* y *hardware* añadido a un computador de propósito general de tal modo que los usuarios pueden interactuar con el computador como si fuese un instrumento electrónico tradicional.

Por tanto, un instrumento virtual consta de un elemento *hardware* y otro *software*. El primero está constituido por un computador personal, que realiza el procesamiento y visualización de los datos y un sistema específico, cuya función es la de adquirir los datos. Este último elemento, generalmente, está constituido por una placa de adquisición de datos, que se inserta en una ranura de expansión en la tarjeta principal de la computadora. El componente *software* consiste en un programa que se ejecuta en la computadora y que controla los elementos *hardware*, procesa y visualiza los datos.

Mientras más rápidas se realicen las pruebas de producción, más pronto se podrá aumentar la capacidad para enviar esos productos a los clientes. Los instrumentos basados en *PC* aumentan el rendimiento y reducen el costo de las pruebas, mediciones y aplicaciones de adquisición de datos.

1.3 Comparación de instrumentación tradicional versus virtual

Tabla I. Instrumentos tradicionales vrs. virtuales

Instrumento tradicional	Instrumento virtual
Definido por el fabricante	Definido por el usuario
Funcionalidad específica, con conectividad limitada.	Funcionalidad ilimitada, orientado a aplicaciones, conectividad amplia.
<i>Hardware</i> es la clave.	<i>Software</i> es la clave
Alto costo / función	Bajo costo / función, variedad de funciones, reusable.
Arquitectura "cerrada"	Arquitectura "abierta".
Lenta incorporación de nuevas tecnología.	Rápida incorporación de nuevas tecnologías, gracias a la plataforma <i>PC</i> .
Bajas economías de escala, alto costo de mantenimiento.	Altas economías de escala, bajos costos de mantenimiento.

La flexibilidad, el bajo costo de mantenimiento, la reutilización, la personalización de cada instrumento, la rápida incorporación de nuevas tecnologías, el bajo costo por función, el bajo costo por canal, etc. Son algunos de los beneficios que ofrece la instrumentación virtual.

La instrumentación virtual puede también ser implementada en equipos móviles *laptops* equipos distribuidos en campo *RS-485*, equipos a distancia (conectados vía radio, *Internet*, etc.), o equipos industriales *NEMA 4X*, etc.

Existe una tarjeta de adquisición de datos para casi cualquier bus o canal de comunicación, protocolos de comunicación y manejos de datos en *PC ISA*,

PCI, USB, serial RS-232/485, paralelo EPP, PCMCIA, CompactPCI, PCI, etc. y existe un controlador driver para casi cualquier sistema operativo WIN 3.1/95/NT, DOS, UNIX, MAC OS, etc.).

Las técnicas utilizadas normalmente para evaluar las características de medición de un multímetro digital *DMM o Digital Multi-Meter* pueden ser utilizadas para evaluar las características de medición de un instrumento virtual *VMM o Virtual Multi-Meter*.

Entre dichas características se encuentran las siguientes:

Tabla II. Características de medición

	DMM	VMM con tarjeta especializada	VMM con tarjeta de propósito general
<i>Hardware utilizado</i>	<i>HP 34401 A DMM</i>	<i>DAQCard 4050</i>	<i>PCI-MIO-16XE-10</i>
No. de Canales	1	1	16 (Diferencial)
Conversión AC	<i>True RMS</i>	<i>True RMS</i>	<i>True RMS (por software)</i>
Resolución (convertidor de 16-bits)	6 ¹ / ₂ – 4 ¹ / ₂ dígitos	5 ¹ / ₂ dígitos	4 ¹ / ₂ dígitos
Rango de entrada (ACV)	100 mV – 750 V	20 mV – 250 V	100mV – 250 V (con acondicionamiento SCXI)
Sensibilidad (ACV)	0.1 uV	0.1 uV	1.5 uV
Rango de Entrada (DCV)	100 mV – 1000 V	20 mV – 250 V	100 mV – 250 V
Sensibilidad (DCV)	0.1 uV	0.1 uV	1.5 uV
<i>NMRR</i>	60 dB	80 dB	Variable (80-120 dB)
<i>CMRR</i>	70 dB (AC), 140 dB (DC)	90 dB (AC), 30 dB (DC)	Variable (80-120 dB)
Velocidad de medición (lecturas/seg.)	5-1 K lecturas/seg.	10, 50, 60 K lecturas/seg.	100 K lecturas/seg.

1.3.1 Beneficios de los instrumentos basados en PC

1.3.1.1 Mayor rapidez en las pruebas

El tiempo de preparación y transferencia de datos es menor con instrumentos basados en *PC* que con instrumentos controlados por *GPIB*. Los instrumentos basados en *PC* usan directamente los buses de alta velocidad *PCI* y *CompactPCI* para mejorar la productividad del sistema de mediciones en más de 20 veces, cuando se compara con sistemas basados en instrumentos individuales.

1.3.1.2 Menor costo de inversión

Los precios de compra de instrumentos basados en *PC* son más económicos, costando aproximadamente la mitad del costo de instrumentos comparables. Además, el mantenimiento cuesta menos y requieren menos espacio en el banco de pruebas o en el bastidor que los instrumentos individuales.

1.3.1.3 Integración simplificada del sistema

Los instrumentos basados en *PC* son modulares y se integran fácilmente para satisfacer los requisitos particulares de pruebas.

Software completo de desarrollo y herramientas de integración, ambientes *LabVIEW* y *LabWindows/CVI* y *software* ejecutivo de pruebas *TestStand*, simplifican la integración del sistema.

El *software* le ofrece mayor adaptabilidad para crear la capacidad necesaria e implementar las rutinas de análisis no disponibles en instrumentos individuales.

Debido a que el *software* es familiar y diseñado sobre estándares abiertos, tales como *Windows*, la capacitación requerida por el usuario es mínima y el tiempo de desarrollo del sistema se reduce dramáticamente. Además, a medida que se desarrollan sus necesidades de medición, se conserva el valor de su inversión en el *software*.

Tabla III. Plataformas de instrumentos basados en computadora

Instrumentos basados en PC	PXI/CPCI	PCI	PCMCIA	ISA	USB
Osciloscopio	✓	✓	✓	✓	✓
Multímetros digitales	✓	✓	✓		
Interruptores	✓				
Generadores de formas de onda arbitraria		✓		✓	
Analizadores de señales dinámicas		✓			
Analizadores de datos en serie	✓	✓			
Probador <i>ISDN</i>			✓		
Registradores de temperatura	✓	✓	✓	✓	✓

2. INSTRUMENTACIÓN VIRTUAL

Un instrumento virtual se define como un nivel de *software* y *hardware* añadido a una computadora de propósito general de tal modo que los usuarios pueden interactuar con el computador como si fuese un instrumento electrónico tradicional.

La instrumentación virtual es un área de la ingeniería en la que se busca obtener dispositivos para registrar diferentes variables físicas, empleando técnicas de implementación híbrida *hardware* y *software*. Aplicando estos principios se logran instrumentos de medida muy flexibles, que facilitan el análisis y la presentación de resultados y que permiten reconfigurar sus parámetros de funcionamiento, haciéndolos a la medida para diferentes aplicaciones.

El concepto de instrumentación virtual nace a partir del uso de la computadora personal *PC* como instrumento de medición de tales señales como:

- Temperatura
- Presión
- Caudal, etc.

Es decir, la computadora comienza a ser utilizada para realizar mediciones de fenómenos físicos representados en señales de corriente por ejemplo 4-20 mA y/o voltaje por ejemplo 0-5 Vdc, como rangos estándar de los elementos de medición.

Sin embargo, el concepto de instrumentación virtual va más allá de la simple medición de corriente o voltaje, sino que también involucra el procesamiento, análisis, almacenamiento, distribución y despliegue de los

datos e información relacionados con la medición de una o varias señales específicas.

Es decir, el instrumento virtual no se conforma con la adquisición de la señal, sino que también involucra la interfaz hombre – máquina, las funciones de análisis y procesamiento de señales, las rutinas de almacenamiento de datos y la comunicación con otros equipos.

El término virtual nace precisamente a partir del hecho de que cuando se utiliza la computadora como instrumento es el usuario mismo quién, a través del *software*, define su funcionalidad y apariencia y por ello se dice que virtualizamos el instrumento, ya que su funcionalidad puede ser definida una y otra vez por el usuario y no por el fabricante.

El instrumento virtual es definido entonces como una capa de *software* y *hardware* que se le agrega a una computadora en tal forma que permite a los usuarios interactuar con ella como si estuviesen utilizando su propio instrumento electrónico hecho a la medida.

2.1 Construcción de instrumentos virtuales

Para construir un instrumento virtual, sólo se requiere de una computadora, una tarjeta de adquisición de datos con acondicionamiento de señales *PCMCIA*, *ISA*, *XT*, *PCI*, etc. y el *software* apropiado, los tres elementos clave en la conformación de un instrumento virtual, teniendo un chasis de acondicionamiento de señales como elemento opcional.

El acondicionamiento de señales es opcional, porque dependiendo de cada señal y/o aplicación, se puede o no requerir amplificación, atenuación, filtraje, aislamiento, etc. de cada señal. Si la señal está en el rango de los $\pm 5V_{dc}$ y no se requiere de

aislamiento o filtraje, la misma puede ser conectada directamente a la tarjeta de adquisición de datos.

Las características de los transductores definen muchos de los requerimientos de acondicionamiento de señales de un sistema de adquisición.

La mayoría de las señales requieren de preparación antes de poder ser digitalizadas. Por ejemplo, una señal de una termopar es muy pequeña y necesita ser amplificada antes de pasar por el digitalizador. Otros sensores como *RTD*, termistores, galgas extensiométricas y acelerómetros requieren de poder para operar. Aún las señales de voltaje puro pueden requerir de tecnología para bloquear señales grandes de modo común o picos. Todas estas tecnologías de preparación son formas de acondicionamiento de señal.

En el instrumento virtual, el *software* es la clave del sistema, a diferencia del instrumento tradicional, donde la clave es el *hardware*. Con el sistema indicado anteriormente, se podría construir un osciloscopio personalizado, con la interfaz gráfica que uno desee, agregándole inclusive más funcionalidad. Sin embargo, este mismo sistema puede también ser utilizado en la medición de temperatura, o en el control de arranque / parada de una bomba centrífuga. Es allí donde radica uno de los principales beneficios del instrumento virtual, su flexibilidad. Este instrumento virtual no sólo me permite visualizar la onda, sino que a la vez me permite graficar su espectro de potencia en forma simultánea.

2.1.1 Estrategia base para diseños de instrumentación virtual

A la hora de diseñar un sistema basado en el uso de instrumentos

virtuales, deben tomarse en cuenta algunos factores de importancia.

a) Identificar los tipos de señales de entrada y salida.

Cuando se usa una tarjeta de adquisición de datos se debe identificar los tipos de sensores y señales de entrada y salida con los que se trabajará. En cuanto a los tipos de entradas y salidas de un sistema de adquisición de datos, usualmente se consideran según el siguiente aspecto:

- Entradas analógicas, temperatura, precisión, voltaje, corriente, señales acústicas y vibración.

- Salidas analógicas, voltaje, corriente.

- Entradas y salidas digitales, entradas y salidas compatibles *TTL*, comunicación paralela, regulador de control.

- Entradas y salidas cronometradas, cronómetros y eventos, entradas y salidas de frecuencia. En estándar de 4-20mA, 0-5Vdc ó 0-10Vdc.

b) Escoger un método de acondicionamiento de señal.

Muchos tipos de señales, provenientes de diversos sensores, deben acondicionarse antes de ser conectados a la tarjeta de adquisición de datos. En esta parte se tiene que hacer una buena elección de transductores y convertidores que se ajusten a los requerimientos del proceso y cuyo costo no resulte muy elevado, para lograr un acople entre los elementos que integran el lazo de control.

Existen, en el mercado, dispositivos de acondicionamiento que pueden adquirirse de acuerdo a las necesidades e, incluso, puede pensarse en diseñarlos e implementarlos para el problema en particular, pues éstos son circuitos bastante sencillos.

- c) Escoger una tarjeta de adquisición adecuada de acuerdo a los criterios de selección.
- d) Escoger el cableado adecuado para la conexión entre la tarjeta y la computadora.
- e) Seleccionar el método de programación adecuado (*software*).

2.1.2 Estrategias para implementar un sistema de instrumentación virtual

- a) Estudiar si es necesaria la implementación de un instrumento virtual. Esto se consigue reconociendo que el instrumento virtual solucionará la necesidad existente produciendo beneficios palpables.
- b) Formar un equipo de funcionamiento cruzado. El cual está conformado por diferentes expertos de distintas áreas para compactar la información necesaria para el desarrollo del *software*.
- c) Entrenar el personal. Cada miembro del equipo estratégico de prueba entenderá la tecnología y base de la instrumentación virtual y cómo el *hardware* y el *software* se aplican a la situación en particular.
- d) Realizar un proyecto piloto. Se basa en probar la efectividad del instrumento virtual y comparar su desarrollo con el del instrumento tradicional.
- e) Implementar el Instrumento. Es la fase final donde se obtendrá el desarrollo y beneficio de la implementación de la estrategia.

2.2 Patrones, calibración, precisión de la instrumentación virtual

2.2.1 Clasificación de los patrones

Un patrón de medición es una representación física de una unidad de medición. Una unidad se realiza con referencia a un patrón físico arbitrario o a un fenómeno que incluye constantes físicas y atómicas.

Por ejemplo la unidad de masa del sistema internacional SI es el kilogramo. Además de las unidades fundamentales y derivadas de medición, hay diferentes tipos de patrones de medición, clasificados por su función y aplicación en las siguientes categorías:

- a) Patrones internacionales
- b) Patrones primarios
- c) Patrones secundarios
- d) Patrones de trabajo

Tabla IV. Definiciones utilizadas en instrumentación

Exactitud es la aproximación con la cual la lectura de un instrumento se acerca al valor real de la variable medida.
Precisión es la medida de la reproducibilidad de las mediciones; esto es, dado el valor fijo de una variable, la precisión es una medida del grado con el cual las mediciones sucesivas difieren una de otra.

Sensibilidad es la relación de la señal de salida o respuesta del instrumento respecto al cambio de la entrada o variable medida.
Resolución es el cambio más pequeño en el valor medido al cual responde el instrumento.
Error es la desviación a partir del valor real de la variable medida.

2.2.2 Calibración

Un instrumento no calibrado constituye un peligro potencial y no puede ser un dispositivo de medición y control confiable.

La calibración es parte esencial de la medición y el control industrial, además es la única garantía que los instrumentos tienen la exactitud y el rango requerido para mantener en operación y en condiciones económicamente controladas los sistemas.

Aunque la calibración no garantiza el buen funcionamiento de un instrumento, por lo general indica si su funcionamiento puede satisfacer o no las especificaciones de precisión y rango en las que se va a utilizar.

2.2.2.1 Calibración de las tarjetas de adquisición de datos

La calibración de un dispositivo de adquisición de datos es muy similar a la de un instrumento de escritorio. El objetivo es comprobar la precisión de una medición que se verifica contra un valor conocido de alta precisión. El resultado obtenido es entonces

comparado contra las especificaciones del instrumento.

Si la medición cae dentro del rango de las especificaciones, no es necesario ningún ajuste. Si la medición se sale del rango de la especificación, entonces se debe realizar un ajuste en el *hardware*.

La mayoría de los sistemas de adquisición de datos cuentan con dos tipos de calibración, la interna y la externa, las cuales ofrecen diferentes beneficios.

La calibración interna le permite al usuario ajustar su instrumento para ambientes diferentes a los lugares donde se realiza la calibración externa. En esencia la calibración interna funciona como una autocalibración o autocero en los instrumentos de escritorio. La calibración externa requiere de estándares externos de alta precisión. Este tipo de calibración generalmente es reservado para los laboratorios de metrología o instalaciones con estándares documentados.

2.3 Aplicación en la industria en general

2.3.1 Prueba y medición

Tabla V. Prueba y medición

Pruebas acústicas
Pruebas de maquinaria
Equipos de prueba automatizada
Inspección óptica automática (AOI)
Pruebas automotrices
Investigaciones biomédicas
Calibración
Pruebas de electrónicos
Alineación de fibras ópticas
Metrología y calibración óptica
Investigación y desarrollo
Pruebas de semiconductores
Pruebas de telecomunicaciones
Pruebas de vibración

2.3.2 Aplicaciones en automatización industrial

Tabla VI. Automatización industrial

Automatización de fábricas
Procesamiento de alimentos
Interfaz hombre – maquina (<i>HMI</i>)
Automatización de laboratorios
Control de máquinas
Visión de máquinas
Procesamiento de petróleo y gas
Procesamiento farmacéutico
Robots para colocación de elementos
Automatización de procesos
SCADA
Control estadístico de procesos
Robots para manejo de semiconductores

2.3.3 Aplicaciones en las telecomunicaciones

Tabla VII. Aplicaciones de telecomunicaciones

<i>AT&T</i> monitores sitios remotos
Evaluación de estaciones base celular
Automatización de sistemas de ayuda <i>Siemens</i> para evaluación de cableado telefónico

2.4 Aplicaciones didácticas

2.4.1 Características de las aplicaciones

Registro en bases de datos o en archivos planos de los estudiantes al acceder a la aplicación, Información, conceptualización utilizando archivos de sonido y vídeo por medio de *activex*.

Evaluación interactiva de términos o conceptos archivados en bases de datos actualizables, registro de evaluación, archivos, comunicación continua con el correo electrónico del profesor u orientador, activación de alarmas vía correo electrónico, simulación del proceso en cuestión, análisis del fenómeno, evaluación de la simulación, control real del proceso por medio de mímicos elaborados como *subVI* o por medio de fotografías dinámicas de los entrenadores existentes.

Programa cliente / servidor que permite multiplicar el entrenador desarrollado a través del control remoto desde diferentes terminales conectadas a la red interna o externa por medio del protocolo *TCP/IP*, inserción de fallas y evaluación en tiempo real suave, el estudiante tiene la posibilidad de armar sistemas relacionados con el área, de acuerdo a la selección del estudiante en la aplicación de los instrumentos virtuales, el computador se encarga de ordenar a los controladores seleccionados por el estudiante para actuar, si el estudiante selecciona una inadecuada configuración, el *software* activa las diferentes alarmas y precauciones permitiendo un análisis práctico del fenómeno.

Por último permite la evaluación de la etapa de control y realizar control estadístico cualitativo y cuantitativo de la evaluación, la evaluación es interactiva, instantánea, individual o colectiva, permite la generación de reportes para impresión o para guardar archivos con otra extensión.

Si se tiene una dirección *IP* válida se puede brindar capacitación a través de la red, sólo teniendo el programa cliente servidor, de tal forma que el estudiante pueda acceder al entrenador a través de la red, desde su casa.

Los entrenadores que se pueden elaborar pueden tener características de control y monitoreo de las variables relacionadas de forma manual (pulsadores e interruptores) o automático (a través de instrumentos virtuales).

2.4.2 Entrenadores didácticos basados en instrumentación virtual a través de *LabVIEW*

Si se tiene como reto, desarrollar *software* educativo, aprovechando la red de computadoras existente, los sistemas de adquisición de datos y las técnicas de multimedia para repotenciar y multiplicar los entrenadores didácticos con los que se cuentan en un centro educativo.

Se puede dar como solución el desarrollo de prácticas virtuales en *Labview* tipo cliente servidor, permitiendo: registro de estudiantes, información sobre prácticas, simulación de fenómenos físicos, evaluación interactiva, control del entrenador manual o desde el computador, mediante tarjeta *PCI6025E* y módulos de sensores según las prácticas, análisis de resultados,

evaluación del control, alarma de activación vía *e-mail*.

El propósito de los entrenadores es familiarizar al estudiante con el comportamiento de las variables para las cuales fueron desarrollados. Estos entrenadores son costosos y cerrados para prácticas, al complementarlos con instrumentación virtual se obtienen ventajas como *software* para adquirir más información sobre la variable en cuestión.

Gracias al uso de protocolos como *TCP/IP* y las grandes ventajas de *Labview*, pueden permitir desarrollar aplicaciones en refrigeración y aire acondicionado, electricidad, electrónica, y controles eléctricos, de modo que al entrenador se acceda desde aulas de formación virtual con 20 computadoras en red. Esta aplicación puede tener todas las características de índole didáctica.

2.4.3 Potenciación de los entrenadores existentes

Si un centro educativo cuenta con una cantidad de entrenadores que datan de años atrás, se podrían hacer aplicaciones elaboradas con *LabVIEW*, utilizando el sistema de adquisición de datos apropiado y el módulo de transductores adecuado, para lograr el fortalecimiento de las características propias del entrenador creando así un sistema alternativo que permite el acceso a él a través del computador. Así mismo se puede desarrollar comunicación a través de puerto serial entre computador y osciloscopio para potenciar el análisis de señales reales y controlar en forma virtual todos los parámetros del osciloscopio.

2.4.4 Entrenador de electricidad básica asistido por computadora

El entrenador podría constar de tres grupos de elementos pasivos variables: resistencia variable, capacitancia variable, inductancia variable, estos grupos pueden ser habilitados o deshabilitados por *software*, se puede configurar en serie o en paralelo o mixto a través de relés y lógica cableada, el entrenador tendría un transductor de corriente que permitirá a través de relés medir la corriente que circula en cada objeto o la corriente total del circuito ya sea en paralelo, en serie o mixto, de tal forma se obtiene instrumentos virtuales como voltímetros, vatímetros, amperímetros, medidor de ángulo de fase multiplicados a través de la red por medio del programa cliente servidor por medio del protocolo *TCP/IP*.

2.5 Sistemas de computación

2.5.1 Plataforma de computación y tipo de sistema

La plataforma de computación sobre la cuál se pretende implementar el sistema de adquisición de datos y control debe ser definida. La definición de la plataforma incluye:

Tabla VIII. Tipos de computadora

<i>PC</i> de escritorio
<i>PC</i> portable (<i>laptop</i>)
<i>PC</i> industrial (<i>rack mount, panel mount, Nema4x, etc</i>)
Estación de trabajo (<i>unix workstation</i>)

Tabla IX. Sistemas operativos

<i>DOS</i>
<i>MS Windows 3.1</i>
<i>MS Windows 95/98</i>
<i>MS Windows NT/2000/XP</i>
<i>MAC OS</i>
<i>Solaris (Unix)</i>
<i>AIX (Unix)</i>
<i>HP-UX (Unix)</i>
<i>Lynux</i>

Tabla X. Tipo de bus o canal de comunicación

<i>XT/ISA/EISA</i>
<i>PCI</i>
<i>Compact PCI</i>
<i>PXI (compactPCI-compatible)</i>
<i>PCMCIA o PC-card</i>
<i>Serial (RS-232, RS-422, RS-485, USB, IEEE-1394)</i>
<i>Puerto paralelo, EPP</i>
<i>GPIB o IEEE-488</i>
<i>VXI o VME</i>

Por ejemplo, un sistema para cualquier aplicación podría ser:

Tabla XI. Ejemplo de configuración de sistema

Tipo	Procesador	Sistema Operativo	<i>Memoria</i> RAM	Bus	Espacio en Disco duro Requerido	Resolución de pantalla
<i>PC</i> Escritorio	<i>Pentium MMX.</i> 233 MHz	<i>MS Windows</i> NT 4.0	64MB	PCI	100MB	1024 x 768 pixel

3. SOFTWARE ASOCIADO A LA INSTRUMENTACIÓN VIRTUAL

3.1 Software

El *Software* es la clave del instrumento virtual, ya que éste es el que sustituye al instrumento tradicional. El *software* juega un rol vital en el desarrollo de sistemas de adquisición de datos y control, además de dirigir la interacción de las especificaciones de *hardware*.

El *software* se elige de acuerdo a las necesidades y preferencias del usuario. Muchos factores afectan la elección del *software* incluyendo aplicaciones, requerimientos, el *hardware* del computador, sistema operativo, el costo y el *hardware* de instrumentación. El *software* que el usuario escoge deberá ser versátil, para adaptarse a diversas arquitecturas de computadoras, a diversos instrumentos y dispositivos de adquisición de datos.

El usuario deberá escoger el *software* de acuerdo a ciertas características, tales como: una arquitectura abierta, un costo reducido, desarrollo de actividades en diferentes plataformas, etc. Si existe un *software* que cubra las necesidades del usuario, entonces se dispondrá de él mismo, sin necesidad de empezar a desarrollar uno nuevo.

Estos programas incluyen un conjunto de librerías como *NI-DAQ* y *NI-IMAQ*. En ambos casos, estas librerías permiten desarrollar aplicaciones con lenguajes y compiladores estándar, tales como *C*, *C++*, *Visual Basic*, etc.

Estas librerías sólo incluyen las funciones básicas para escribir y leer a las tarjetas de adquisición de datos, junto a ciertas funciones de temporización, disparadores ó *triggers*, y configuración.

Sin embargo, si se desea contar con un sistema de desarrollo completo, que además de manejar las tarjetas de adquisición de datos sea capaz de generar interfaces hombre máquina, procesar las señales capturadas, ejecutar algoritmos de control, almacenar los datos capturados en bases de datos y comunicarse con otras aplicaciones locales o remotas, se recomienda utilizar alguno de los sistemas de desarrollo siguientes:

Tabla XII. Algunos programas para instrumentación virtual

Producto	Imagen	Lenguaje	Características
<i>LabVIEW</i>		G	Lenguaje 100% gráfico.
<i>LabWindows/CVI</i>		C/C++	Basado en C/C++.
<i>BridgeVIEW</i>		G	Lenguaje 100% gráfico, para sistemas SCADA
<i>ComponentWorks</i>		Visual Basic	Basado en <i>Visual Basic</i> , controles OCX.
<i>Measure</i>		Excel	Macros en <i>Excel</i> , VBA.
<i>Lookout</i>		Expresiones	Sistema SCADA, sin programación.

Dependiendo del lenguaje que se prefiera, bien sea C/C++ (*LabWindows/CVI*), *Visual Basic* (*ComponentWorks*), *Excel* (*Measure*) o un lenguaje gráfico como el G (*LabVIEW* y *BridgeVIEW*), se puede seleccionar cualquiera de estas opciones. Si no se desea programar, y se piensa utilizar

Fieldpoint o un controlador lógico programable *PLC* para controlar procesos, *Lookout* es la mejor alternativa.

3.2 Programación tradicional versus programación gráfica

3.2.1 Programación tradicional

La programación tradicional esta basada en texto. Las líneas de código son creadas usando palabras clave y reglas de sintaxis. Programar con lenguajes tradicionales es difícil debido a que el programador debe manejar la ejecución y el flujo de datos por medio de palabras clave, sintaxis, nombres de variables y estructuras de datos.

Los nombres de las variables son asignados por el programador y necesitan ser digitados dentro del código en numerosos lugares. Es relativamente fácil cometer errores con el nombre de la variable, usando el nombre equivocado o usando el nombre correcto en el lugar equivocado. El ambiente de depuración fue creado para permitir a los programadores ver la ejecución y el flujo de datos. Este puede mostrar cual es el problema pero no puede eliminar la raíz del problema. Mientras un programa es creado, el programador encripta la ejecución y el flujo de datos. Deducir el flujo de ejecución solo observando el código fuente, puede ser muy difícil, aun para el autor. Es por eso que es llamado código original.

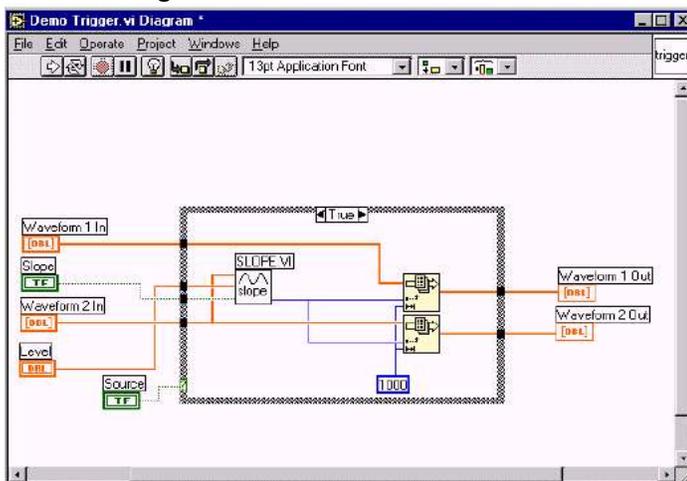
3.2.2 Programación gráfica

La programación gráfica es inmensamente diferente de la programación tradicional debido a que el flujo de datos y el flujo de

ejecución son manejados gráficamente. Un diagrama es creado usando objetos y líneas. Las líneas sobre todo representan los datos y los objetos analizan, crean o despliegan los datos.

El flujo de ejecución típicamente sigue el flujo de datos. Cuando el flujo de ejecución no puede seguir el flujo de datos, el flujo de ejecución aun es manejado gráficamente. Un programa gráfico es fácil de descifrar o dar soporte para el autor o para alguien más. La programación gráfica es relativamente sencilla debido a que los objetos son seleccionados de un menú los cuales proveen un efecto de palanca confiable que consiste en (ejecución, interfase del usuario y despliegue), y un útil código. La mayoría de los objetos utilizados podrían requerir cientos y aun miles de líneas de código tradicional.

Figura 1. Ejemplo de programación gráfica
Fuente: Programa de demostración de *Labview*



La ejecución y el flujo de datos son claramente representados en el diagrama creado que representa el programa. Existe menor posibilidad de errores humanos y los problemas son fáciles de localizar.

El código final es fácilmente interpretado por el autor así como por otros programadores.

3.3 VEE (*Visual Engineering Environment*)

Es un lenguaje de programación gráfico que se enfoca principalmente en problemas de ingeniería. Sobre todo que permite al usuario recolectar datos de un instrumento, analizar, filtrar y desplegar los datos de forma más significativa.

El usuario no necesita escribir programas convencionales. De este modo permite a los ingenieros de instrumentación el desarrollo rápido de complejos sistemas de control.

Tabla XIII. características de VEE

Dominio: HP VEE se enfoca en problemas de ingeniería que interactúan con instrumentos.
Componentes: Principalmente son los controladores <i>Drivers</i> de los instrumentos. Esas piezas de <i>software</i> que proveen una descripción gráfica y funcional de cómo interactúa uno mismo con los instrumentos. Muchos instrumentos (o instrumentos como componentes) se proveen con la distribución de VEE, pero VEE permite al usuario crear sus propios instrumentos.
Marco de Trabajo: Un modelo de flujo de datos transfiere vectores de los datos del instrumento para desplegar y procesar.
Lenguaje: Visual, ambiente gráfico usando conexiones alambradas a lo largo del flujo de datos.
Aplicaciones ejemplo: Son aplicaciones que también pueden ser cortadas y pegadas dentro de un nuevo programa.

VEE es razonablemente completo, dado que utiliza una programación gráfica. Sin embargo para aplicaciones que no sean de instrumentación por ejemplo: Acceso a bases de datos, manejo de interrupciones seriales de entrada y salida, cálculos complejos, etc. VEE será muy torpe.

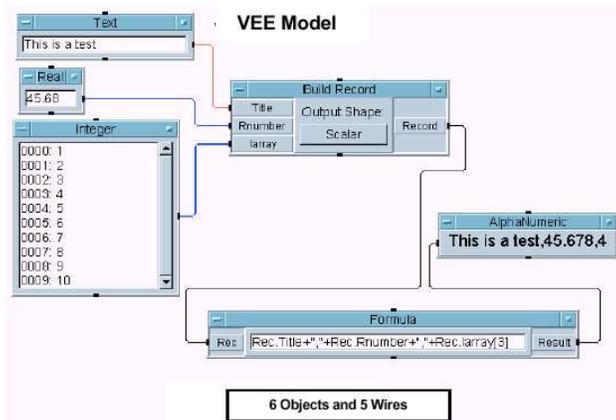
Figura 2. Objeto de diálogo de archivo VEE



Fuente: Programa VEE

En el frente de extensión abierta se pueden agregar nuevos instrumentos tanto como lo permita el modelo de interacción con el instrumento, sin embargo otra extensión será limitada. Los archivos de HP VEE son almacenados en formato ASCII lo que permite ventajas sobre transferencias de archivos entre sistemas operativos, edición y aun documentación.

Figura 3. Modelo de VEE



Fuente: Programa VEE

Figura 4. Panel de interfase con el usuario en VEE

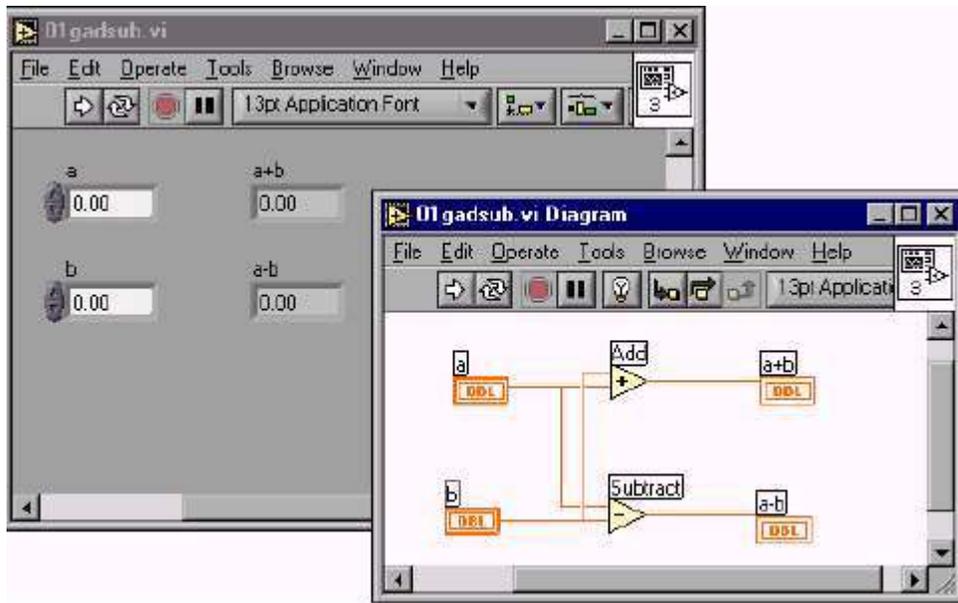


VEE User Interface Panel

Fuente: Programa VEE

3.4 LABVIEW (Laboratory Virtual Instrument Engineering Workbench)

Figura 5. Diagrama de bloques y su respectivo panel frontal



Fuente: Programa de demostración de *Labview*

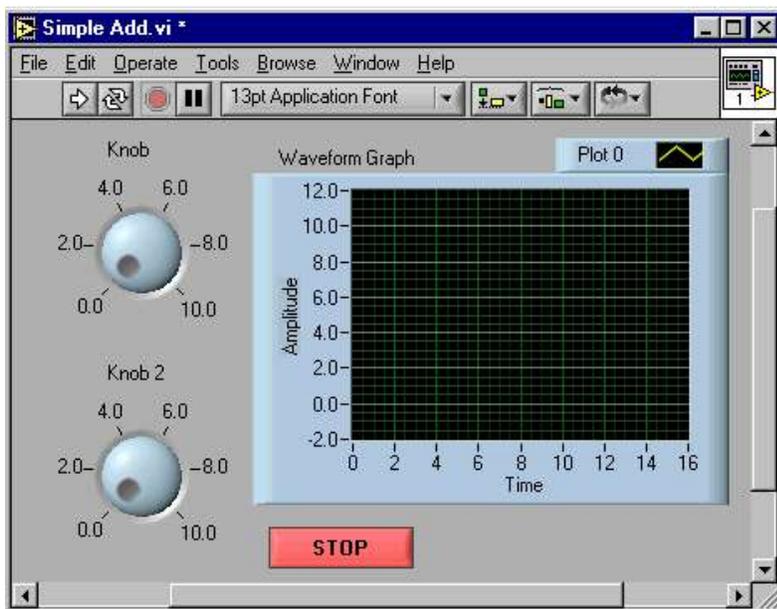
Labview es un lenguaje de programación gráfico para el diseño de sistemas de adquisición de datos, instrumentación y control. *Labview* permite diseñar interfaces de usuario mediante una consola interactiva basado en *software*.

Permite diseñar si se especifica un sistema funcional, un diagrama de bloques o una notación de diseño de ingeniería. *Labview* es a la vez compatible con herramientas de desarrollo similares y puede trabajar con programas de otra área de aplicación, como por ejemplo *Mathlab*. Tiene la ventaja que permite una fácil integración con *hardware*, específicamente con tarjetas de medición, adquisición y procesamiento de datos, incluyendo adquisición de imágenes.

Este lenguaje de programación permite desarrollar de una forma más rápida cualquier aplicación, especialmente de instrumentación, en comparación con lenguajes de programación tradicionales basados en texto, sin embargo si se desea una aplicación sencilla como un programa que sume dos números, definitivamente construirlo bajo *Labview* es más demorado y tedioso, seria más sencillo mediante un programador de texto donde simplemente se incluirá una línea.

Pero para un programa más complejo se puede diseñar un prototipo y modificarlo de una manera más rápida con *Labview* debido a que es un lenguaje programación gráfico.

Figura 6. Instrumento virtual realizado con *Labview*



Fuente: Programa de demostración de *Labview*

3.4.1 Características de Labview

Una de las principales características de *Labview* es su modularidad, es decir, la capacidad de utilizar bloques funcionales para la definición de la especificación. *Labview* permite conectarse a otras aplicaciones mediante un intercambio de datos como *Active X*, librerías dinámicas, bases de datos, *Excel* y/o a protocolos de comunicación como *DataSocket*, *TCP/IP*, *UDP*, *RS-232*, entre otras.

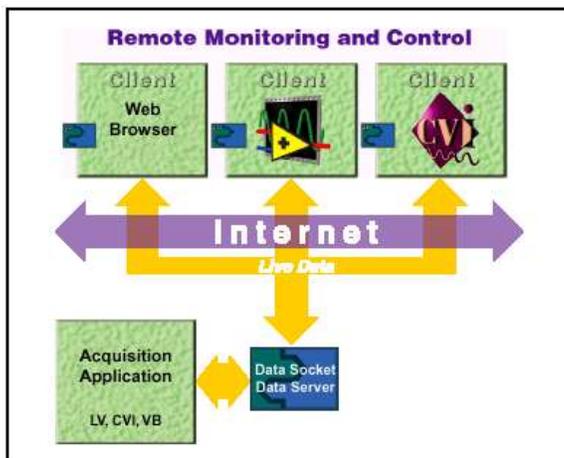
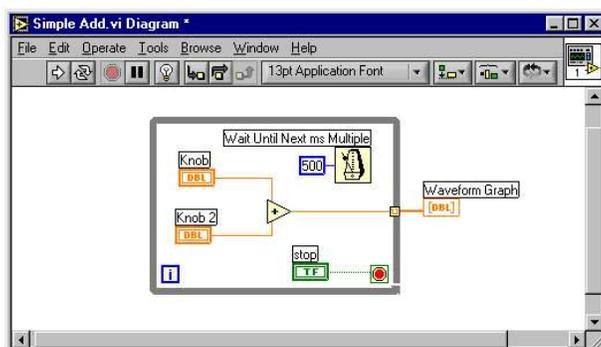


Figura 7. Monitoreo remoto y control

Fuente: Programa de demostración de *Labview*

Figura 8. Diagrama de bloque



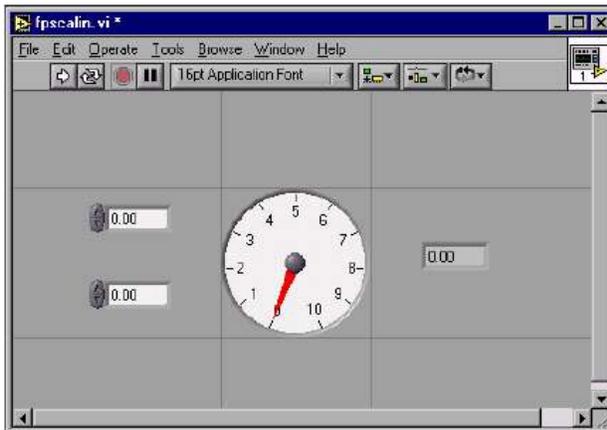
Fuente: Programa de demostración de *Labview*

3.4.2 Adquisición de datos con Labview

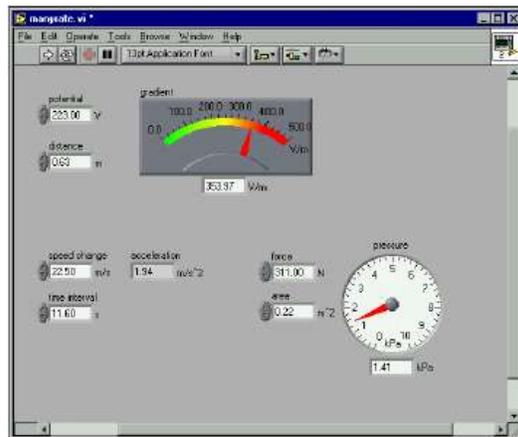
Labview es un lenguaje que se enfoca hacia el laboratorio, la realización de mediciones y por lo tanto la adquisición y análisis de datos. Mediante el uso de tarjetas es posible obtener señales análogas o digitales a partir de una conexión al bus *PCI* en una computadora, estas tarjetas se diferencian por el numero de muestras por segundo que pueden realizar, por el numero de *bits* ó resolución y por el numero de canales que manejan.

Por ejemplo una tarjeta de bajo costo como la *PCI-1200* realiza 100KS/s, con una resolución de 8 bits para 3 puertos digitales de entrada salida y dos de salida análoga, junto con 12 bits de resolución para 8 canales de entrada análogos, y puertos de temporización mediante una *PIT* (8253); la tarjeta esta conformada por *PPIs*, una *PIT* que usa la tarjeta para su programación y otra para disponibilidad del usuario, una *FPGA* y conversores digitales a análogo *DAC* y demás componentes básicos.

Figura 9. Ejemplos de instrumentos virtuales



PANEL FRONTAL CON OBJETO FIJADO A ESCALA.



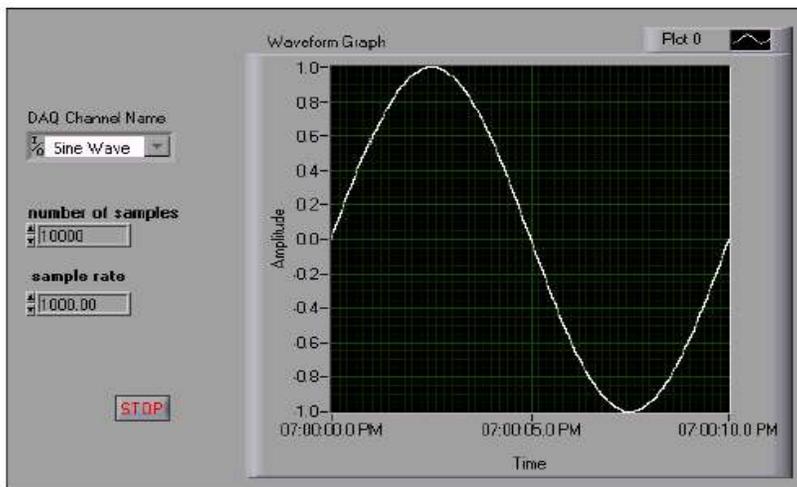
PANEL FRONTAL MULTIPLES RANGOS.

Fuente: Programa de demostración de *Labview*

La tarjeta se debe programar para definir los niveles de voltaje que debe manejar y si son bipolares o unipolares. Las características de la tarjeta se pueden configurar con la herramienta

Measurement & Automation que se encuentra en el explorador de *Windows* como un icono principal. Esta herramienta permite verificar que tarjetas hay instaladas en el computador, esto solo para el caso de tarjetas de adquisición de datos o de imágenes. *Labview* también permite comunicarse con otras tarjetas mediante el puerto paralelo o serial, utilizando la herramienta *LabWindows CVI*. Con este *software* se pueden diseñar en C.

Figura 10. Captura de una forma de onda senoidal por medio de



GRAFICA DE FORMA DE ONDA.

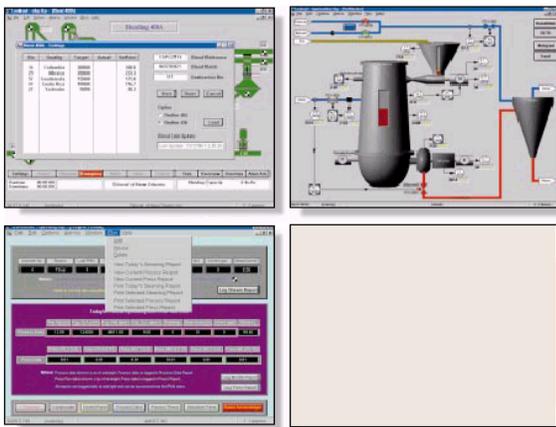
Labview

Fuente: Programa de demostración de *Labview*

3.5 Lookout

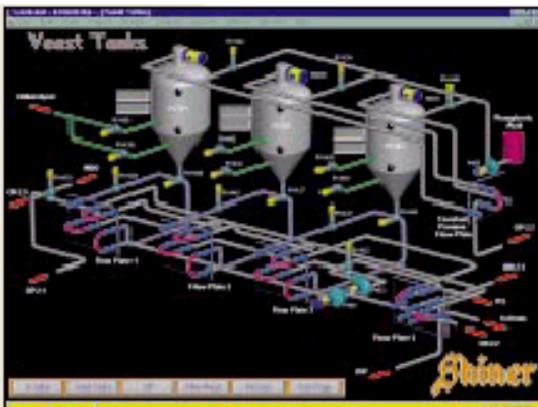
Es un programa para automatización industrial con conectividad a controles lógicos programables *PLC*, unidad terminal remota *RTU*, *FieldPoint*, adquisición de datos *DAQ*, *SCXI* dispositivos *foundation fieldbus*, etc. para crear interfaces hombre máquina *HMI* y sistemas de adquisición de datos y control supervisorio *SCADA* con alto desempeño bajo *Windows*, *Windows 95* y *Windows NT*.

Figura 11. Aplicaciones de *Lookout*



Fuente: Programa de demostración de *Lookout*

Figura 12. Ejemplo de control de proceso industrial



Fuente: Programa de demostración de *Lookout*

Lookout es un sistema orientado a objetos. Así que para crear una interfaz hombre máquina *HMI* simplemente se configuran los

objetos y se colocan en la pantalla gráfica. Entre las ventajas a destacar de *Lookout* esta la de permitir una auténtica configuración en línea. Mientras se van creando y modificando los objetos, éstos reflejan de forma inmediata el comportamiento real, incluso al operar en modo edición. Esta capacidad permite realizar cambios a la interfaz del operador sin detener o interrumpir el proceso industrial.

Adicionalmente *Lookout* posee una arquitectura basada en eventos, de forma que las aplicaciones son rápidas y aprovechan de manera eficaz los recursos de la computadora.

Es un software que permite fácilmente crear poderosas aplicaciones de monitoreo y control de procesos. Con *Lookout*, el desarrollo de una interfaz hombre máquina requiere de menos tiempo lo que permite un ahorro sustancial en el costo total de un proyecto. No requiere de programación, comandos tipo texto ni compilación.

Se tiene capacidad de realizar telemetría. Utiliza controladores drivers basados en *DLL (data link library)* conectividad *OPC (ole process controled)* intercambio dinámico de datos *DDE, ASCII (American Standard code for information interchange) (transfer code protocol/internet protocol)TCP/IP* y *UPD*.

3.5.1 Características de *lookout*

3.5.1.1 Arquitectura basada en objetos

Lookout elimina completamente la programación, *scripts* o compilación separada. Solamente tiene que configurar y conectar objetos para desarrollar aplicaciones de monitoreo y control. La arquitectura basada en objetos permite más fácilmente desarrollar y mantener aplicaciones, reduciendo aún más el costo total de un proyecto.

3.5.1.2 Conexión en red

El conectar múltiples servidores y clientes dentro de una planta o en locaciones remotas es muy fácil con *Lookout*. Solamente tiene que hacer un *browse* y seleccionar para poder conectarse a cualquier computadora en la red. *Lookout* cuenta también con las habilidades de distribuir la carga de su aplicación en una red y poder ver instantáneamente los puntos de entrada y salida de cualquier punto en la red.

3.5.1.3 Listo para *Internet*

Con *Lookout*, es posible monitorear y controlar su proceso usando un *browser* de *Web* como *Internet Explorer* sin necesidad de programar en *HTML* o *XML*. Solamente seleccionando la función de exportar su proceso como una página *web* en *Lookout* y esta listo.

Tabla XIV. Lista de funciones de *Lookout*

Lista de funciones en <i>Lookout</i>
Desempeño confiable para diversas aplicaciones
Conectividad a nivel empresa (<i>MES/ERP</i>)
Conectividad abierta
Herramientas de manejo de datos

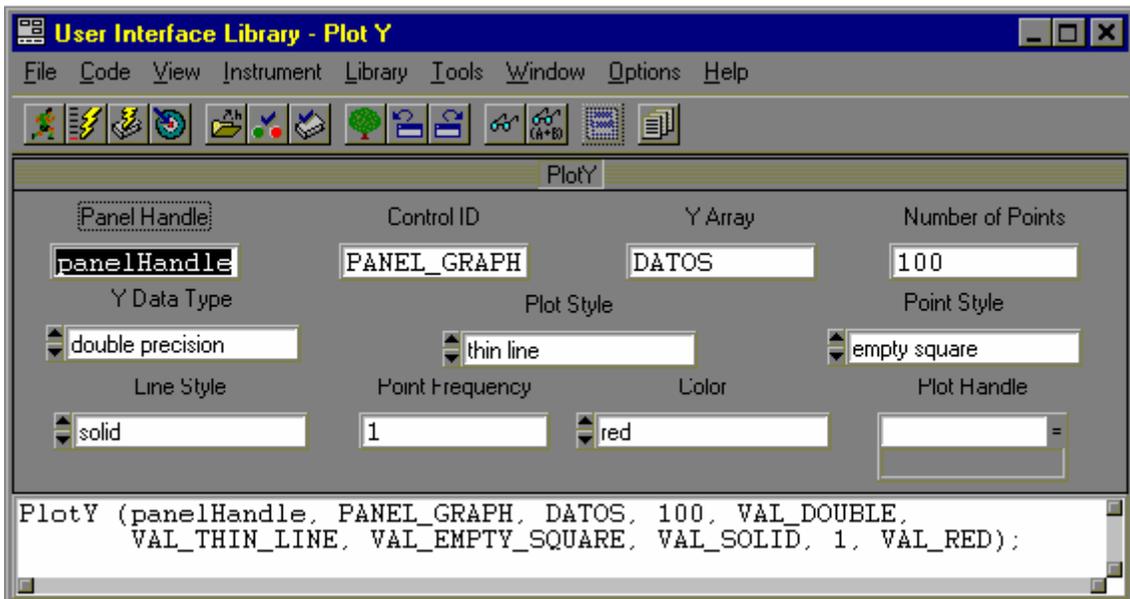
Generación de reportes
Visualización
Control supervisorio
Manejo de eventos
<u>Configuración en línea</u>
<i>Loggeo</i> distribuido
Alarmas y eventos distribuidos
Seguridad
Redundancia
Tendencias y gráficas

3.6 *LabWindows/cvi*

Es un ambiente integrado *ANSI C* diseñado para ingenieros y científicos que trabajan en la creación de aplicaciones de instrumentación virtual. Con bibliotecas de entrada y salida integradas, rutinas de análisis y herramientas de interfaz de usuario, esto permite crear sistemas avanzados de prueba y medición. Un ambiente integrado de *labwindows/cvi* incluye una gran variedad

de herramientas generadoras de código y utilidades para creación de prototipos para un desarrollo rápido y fácil de código C.

Figura 13. Panel frontal de *Labwindows/cvi*



Fuente: Programa de demostración de *Labwindows/cvi*

3.6.1 Características de *Labwindows/cvi*

Este programa contiene herramientas para desarrollo de aplicaciones para *Windows* y *Unix*. Además permite el desarrollo de controladores de instrumentos virtuales múltiples o unitarios.

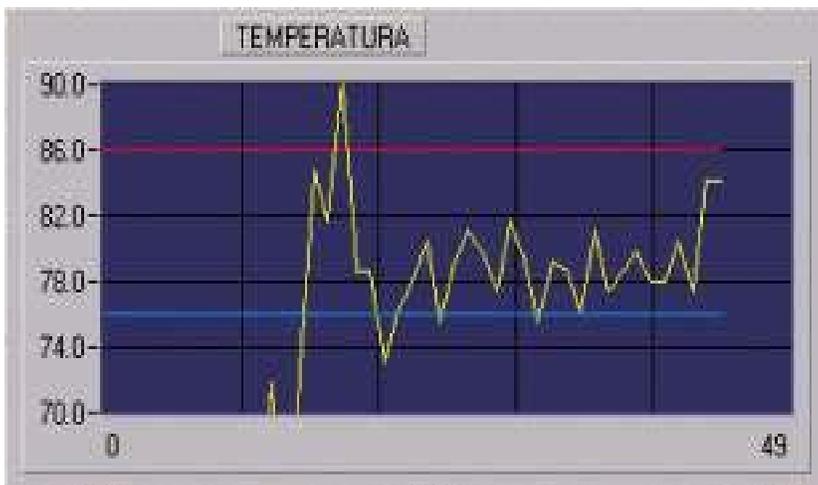
Desarrollo C más fácil y rápido usando un editor de recurso para interfaz de usuario *UIR drag and drop* y herramientas generadoras de código, ambiente

abierto ANSI C compatible con: *Microsoft Visual C++*, *Borland C++*, *Watcom C++*, *Symantec C++*.

Conectividad con bibliotecas de intercambio dinámico de datos *DDE* y (*transfer code protocol / Internet protocol*) *TCP/IP* y *UPD*.

Extensa biblioteca de análisis para el procesamiento de señales, estadísticas, aproximación de curvas y análisis complejo. Bibliotecas de instrumentación incorporadas que incluyen *GPIB*, adquisición de datos y análisis de señales.

Figura 14. Gráfica de muestreo de temperatura



Fuente: Programa de demostración de *Labwindows/cvi*

3.7 Interfase con autómatas

3.7.1 *Simatic WinCC (Windows control center)*

Es un sistema *SCADA* escalable con todas las funciones de manejo y visualización integradas. *WinCC* es el primer sistema de *software IHMI Integrated Human Machine Interface* en el mercado que integra el *software*

controlador de planta con los procesos de automatización. Los componentes de *WinCC* que se distinguen por su fácil uso permiten integrar sin problemas aplicaciones nuevas o existentes.

Con *WinCC* se dispone de un programa de visualización que permite supervisar todos los aspectos de los procesos de automatización.

Combina la arquitectura moderna de las aplicaciones de *Windows NT 4.0* con la sencillez de un programa de diseño gráfico, incluye todos los elementos necesarios para controlar y supervisar.

Está concebido para la visualización y manejo de procesos flexiblemente ampliable, diseñado para aplicaciones simples en la maquinaria hasta complejas aplicaciones multipuesto o incluso sistemas distribuidos con varios servidores redundantes y soluciones para grandes distancias en plantas, líneas de fabricación, e instalaciones. El volumen de funciones de este moderno sistema incluye la emisión de avisos de eventos en una forma adecuada para la aplicación industrial, el archivo de valores de medida y recetas y el listado de los mismos.

Con su potente acoplamiento al proceso, especialmente con *SIMATIC*, y su seguro archivo de datos, hace posible unas soluciones de alto nivel para la técnica de conducción de procesos.

3.7.1.1 Características de *WinCC*

Dentro de sus características destaca su potente sistema de almacenamiento, basado en *Microsoft SQL Server 2000*, con funcionalidad histórica integrada y archivación de altas prestaciones de valores de proceso y de mensajes de alarma.

Asimismo, integra funcionalidades mejoradas para los sistemas cliente-servidor, puede funcionar a través de routers *LAN* e incluye nuevos estándares abiertos con *Visual Basic* para aplicaciones (*VBA*) para la automatización de la configuración y *Visual Basic scripting (VBScript)* como nuevo lenguaje de programación en tiempo de ejecución adicional. Además, viene con un sistema de informes mejorado y un navegador *web* para acceso a *Internet*.

El tiempo de ejecución de *WinCC* permite a los operarios interactuar con la aplicación directamente en la máquina o desde un centro de control.

El entorno de ingeniería de proyectos de *WinCC* permite desarrollar:

Tabla XV. Aplicaciones con *WinCC*

Actividad	Aplicación
Dibujos	Diseñar representaciones de planta

Estructura de archivos	Guardar datos / eventos marcados con fecha y hora en una base de datos <i>SQL</i>
Generador de informes	Generar informes sobre los datos solicitados
Administración de datos	Definir y recopilar datos de toda la planta

El sistema básico de *WinCC* suministra todos los componentes y las funciones que se necesitan para implementar complejas tareas de visualización, además editores para la creación de imágenes, scripts, alarmas, tendencias e informes.

3.7.1.2 Requisitos previos del entorno de trabajo de *WinCC*

WinCC soporta todas las plataformas de computadoras comunes y compatibles con *IBM/AT*. Aunque se indican los valores para una configuración mínima, es conveniente utilizar los valores recomendados para la configuración máxima de modo que *WinCC* funcione eficazmente.

Tabla XVI. Configuración de sistema para *WinCC*

	Valores mínimos	Valores recomendados
<i>CPU</i>	<i>INTEL Pentium II 266</i>	<i>INTEL Pentium II 400 MHz</i>

	Valores mínimos	Valores recomendados
	MHz	
RAM	96 MB	128 MB
Controlador gráfico	SVGA (4 MB)	XGA (8 MB)
Resolución	800 * 600	1024 * 768
Disco duro	Espacio disponible: 500 MB	Espacio disponible: > 500 MB
	Unidad de <i>CD-ROM</i>	Unidad de <i>CD-ROM</i>

El rendimiento del sistema depende del número de tags en uso y del espacio de memoria disponible. Es necesario tener en cuenta que otros programas activos también pueden disminuir el rendimiento.

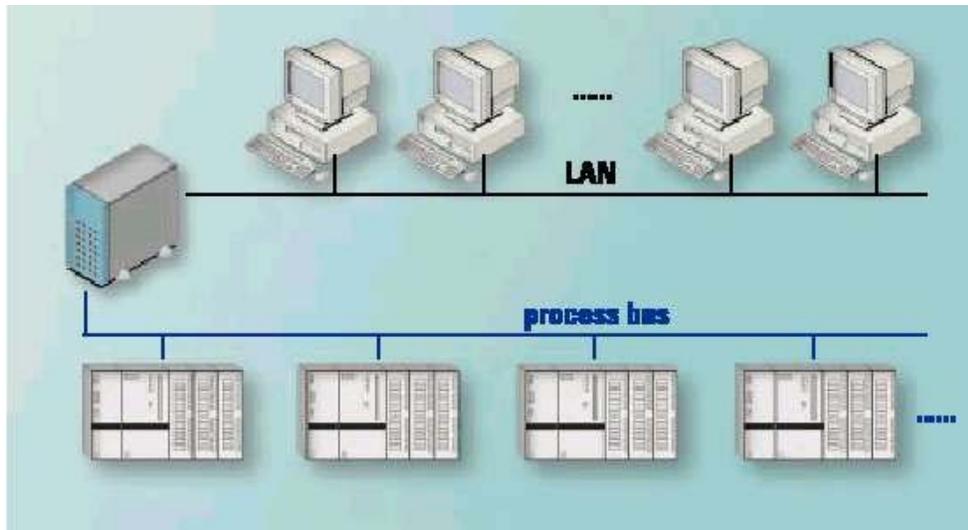
3.7.1.3 Interfaces con *WinCC*

Diversos paquetes de opciones *WinCC* y *Add-ons* han sido concebidos en base a las interfases abiertas de programación. Demuestran así la capacidad de ampliación modular y de utilización universal de *WinCC*.

Con la opción *WinCC Server* se puede transformar *WinCC* en un potente sistema cliente/servidor. De esta forma se puede operar varias estaciones de

manejo y visualización coordinadas en unión con los sistemas de automatización interconectados en red.

Figura 15. Sistema cliente/servidor usando *WinCC*



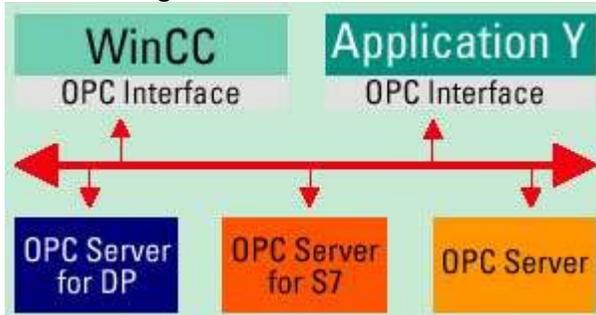
Fuente: Programa de demostración de *WinCC*

En la actualidad la integración de varios componentes, que además pueden ser de diferentes fabricantes, resulta con frecuencia penosa. Las aplicaciones requieren *software* controlador, o servidores desarrollados para cada aparato o bien para cada componente.

OPC (ole for process control) permite integrar componentes de automatización, en un sistema *plug and play*. Con *WinCC* el usuario puede concentrarse plenamente en sí al problema de automatización, ya que esta asegurada la integración de los componentes necesarios.

Figura 16. Arquitectura con OPC

Fuente: Programa de demostración de WinCC



3.7.2 RSVIEW

Es un software *HMI* integrado, basado en componentes para automatización y control de maquinaria y procesos. Diseñado para ambientes de *Microsoft Windows* y dispone de funciones de tendencias y alarmas sofisticadas además de una interfase sumamente gráfica para simplificar la configuración, funcionamiento y administración de procesos. Disponible en varios idiomas. Compatibilidad con tecnologías abiertas que permiten conectividad con otros productos de *software* y diversidad de aplicaciones.

Es completamente compatible con contenedores *OLE* para *ActiveX*, lo que facilita la inclusión de controles de este tipo suministrados por terceros. Incluye *VBA*, *Visual Basic* para aplicaciones como parte integrante de sus funciones, de modo que posibilita maneras ilimitadas de personalizar los proyectos.

Su compatibilidad con la tecnología cliente/servidor *OPC* le permite comunicarse con una amplia variedad de dispositivos de hardware.

El producto se complementa con *RSView32 Active Display System* y *RSView32 WebServer* (el primero para ver y controlar los proyectos

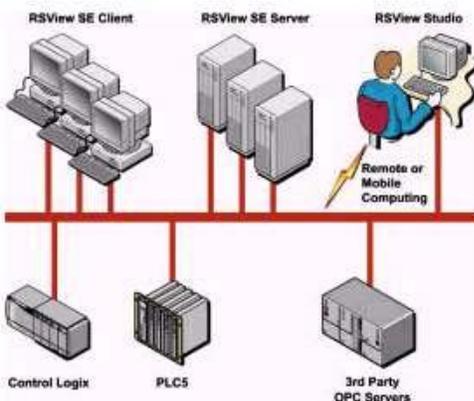
RSView32 desde localidades remotas y el segundo para que cualquier usuario autorizado pueda acceder a gráficas, etiquetas y alarmas mediante el uso de un navegador de Internet convencional.

El número límite de *tags* a utilizar podría ser por ejemplo 100,000 • 32,000 • 5,000 • 1,500 • 300 • 150 los cuales se pueden actualizar en cualquier momento sin reconfigurar todo el proyecto.

Las alarmas, configuración de *tags*, y registro de datos son compatibles con *ODBC*, lo cual permite manejar bases de datos como *Microsoft Access*, *SQL Server*, *Oracle*, *Sybase* y gráficamente ver las tendencias de los datos.

Debido a que *Rsview* incorpora el mismo lenguaje de programación que *Visual Basic*, *Microsoft Office* y otros productos de *software* con los mismos estándares, comparten una inherente interoperabilidad a nivel objeto, el cual provee una solución factible a un determinado proyecto. Los *drivers* de comunicación permiten conectarse con otros dispositivos Allen-Bradley o cualquier software *OPC (ole for process control)*.

Figura 17. Conectividad usando *Rsview*



Fuente: Manual de *Rsview*

Figura 18. Pantalla de HMI/SCADA utilizando *Rsview*



Fuente: Programa de demostración de *Rsview*

3.7.2.1 Características adicionales de *RSView32*

Con *VBA*, se pueden crear códigos con ramificaciones condicionales, dirigir los datos *RSView32* a aplicaciones de terceros y controlar *RSView32* desde una subrutina de *VBA* mediante la emisión de comandos *RSView32*.

Con las posibilidades que ofrece el contenedor de *RSView32 ActiveX* y *OLE* se puede aprovechar al máximo la avanzada tecnología.

Fácil integración de *RSView32* con otros productos *Rockwell Software*, por ejemplo *RSLogix 500*, *RSLogix 5*, *RSSql* y *RSWire*, para una solución de automatización completa.

Se pueden construir gráficos orientados al objeto, tales como medidores, tanques, tubería, placas frontales y botones u objetos personalizados. Provee soporte completo de vínculo e incorporación de objeto (*OLE*). Permite conectividad de la base de datos abierta de *Microsoft (ODBC)*.

3.7.2.2 Aplicaciones con *RSView32*

- Es posible utilizar la capacidad del contenedor *RSView32 ActiveX* y *OLE* para aprovechar la tecnología avanzada. Por ejemplo, puede incrustar *RSTools*, *Visual Basic* u otros componentes *ActiveX* en las pantallas gráficas de *RSView32* para ampliar las capacidades de éste.
- Crear y editar pantallas con las herramientas propias de los programas de *Microsoft* que se está utilizando. Mediante sofisticados gráficos y animaciones basados en objetos, más las técnicas simples de arrastrar colocar y cortar - pegar, se simplifica la configuración de la aplicación.
- Se puede utilizar el modelo de objetos *RSView32* y *VBA* para compartir datos con otros programas de *Windows*, tales como *Microsoft Access* y *SQL Server*, interactuar con otros programas de *Windows* tales como *Microsoft Excel*, así como personalizar y extender *RSView32* adaptándolo a sus necesidades específicas.
- Permite utilizar gráficos de las bibliotecas de gráficos *RSView32* o importar archivos de otros paquetes de dibujo tales como *CorelDRAW* y *Adobe Photoshop*.
- Desarrollar rápidamente su aplicación utilizando herramientas de productividad *RSView32* tales como el asistente de comandos, el examinador de *tags* y la ruta inteligente de objeto *Object Smart Path (OSP)*.
- Evitar introducir información repetida. Importando una base de datos de un *PLC* con el examinador de bases de datos de *PLC*.

- Es posible utilizar las funciones de alarmas de *RSView32* para monitorear incidentes ocurridos en el proceso con varios niveles de gravedad. Cree resúmenes de varias alarmas para obtener datos específicos sobre las alarmas en lugar de examinar las alarmas de la totalidad del sistema.
- Crear tendencias que muestren variables del proceso graficadas en relación con el tiempo. Se puede mostrar datos en tiempo real o históricos hasta con 16 plumas (*tags*) en cada tendencia.
- Registrar datos simultáneamente en varios archivos de registro o bases de datos *ODBC* remotas para proporcionar diversos registros de los datos de producción. Lleve los datos registrados directamente a programas de otros fabricantes tales como *Microsoft Excel* y *Seagate Crystal Reports* sin necesidad de convertir los archivos

3.8 Características económicas del *software* asociado a la instrumentación virtual

3.8.1 Análisis económico de *WinCC*

Este paquete según la versión del sistema operativo correrá:

<i>WinCC</i>	V5.1:	Para	<i>NT</i> ,	98
<i>WinCC</i>	V6:	<i>Windows</i>	2000,	<i>XP professional</i> .

Existen licencias para *WinCC server* o cliente, si es un sistema *stand-*

alone, entonces será la licencia cliente.

Luego está la opción de *Runtime* (RT) y *Configuration* (RC). La RT es para correr y poder realizar pequeños cambios (10 minutos únicamente), la RC es para correr y poder hacer cambios sin que le saque del paquete de configuración. Aparte existen muchos *Add-ons* para irle agregando al paquete (ejemplo: manejo de alarmas, clientes-servidores, manejo de varios monitores, *web-client*, almacenamientos, manejo de *chipcard*, etc.

Las licencias van según los tags que éstos son válidos para puntos de conexión únicamente, los tags internos son ilimitados (en teoría). Las licencias van de 128 tags, 256, 1024, 8k (solo para *WinCC V6.0*) y 64K.

Precios para *WinCC V6.0 Stand-alone IVA* incluido:

Tabla XVII. Precios de WinCC según los tags requeridos

RT64K	Q 85,950.00
RT8K	Q 69,920.00
RT1024	Q 58,200.00
RT256	Q 47,350.00
RT128	Q 30,820.00
RC 64K	Q 144,215.00
RC 8K	Q 125,275.00
RC 1024	Q 101,245.00
RC 256	Q 78,660.00
RC 128	Q 46,180.00

Se debe tomar nota que las *tags* pueden declararse como *BIT*, *Word*, *dword*, *float*, etc. y de forma interna se puede separar, es decir si en un motor tenemos los bits de *start*, *stop*, falla, manual, auto, estado, etc. se puede juntar en un *word* siendo una sola *tag* para el ahorro de variables.

Si se compra la de 128 *tags* existe posteriormente un único *upgrade* a mayor cantidad de variables.

3.8.2 Análisis económico de *Rsview*

Rsview tiene la opción de *Runtime* o configuration. La *Runtime* se refiere a que solamente se puede correr la aplicación sin modificaciones. La opción *configuration* permite correr el programa, además realizar cambios al mismo cuando se requiera.

El programa *Rsview* permite:

- Capacidad de almacenamiento de eventos por fecha, hora, operador.
- Almacenamiento de alarmas en formato *DBF wide/open format*, configurable dependiendo de la capacidad de disco duro de la computadora.

- Almacenamiento de históricos en formato *DBF wide/open format*, utilizando el *Trendx 4.0* configurable según las variables más importantes del proceso.
- 20 modelos distintos de tendencias cada uno con 10,000 variables, almacenado y compartido en formato *ODBC* o bases de datos propietarias.
- Conexión hacia la red por medio de *keepserver* y continuar con las tarjetas que actualmente hacen dicha conexión desde las computadoras existentes.

El costo aproximado para un modulo *Runtime* para *Rsview*, con un total máximo de 32K variables de control y visualización que incluye lo siguiente:

- Para este *software* esta estimada una computadora del tipo industrial marca *Allen-Bradley* con las siguientes características:
 - Procesador *Pentium IV 2.0 Ghz* de velocidad
 - 512 Mb RAM, 40Gb disco duro
 - *DVD-CD-RW drive*
 - Sistema operativo montado en fabrica *windows2000* o *WindowsXP PRO*
 - Monitor 17'' *Flat Panel Monitor, resistive touch screen* modelo 6186-M1ALTR.

Es de **US\$ 15,658.00**

3.8.3 Resumen económico de *Labview*

<i>LabVIEW for Windows, Mac OS, Linux (1 user)</i>	<i>US\$ 545</i>
<i>LabVIEW Datalogging and Supervisory Control Module</i>	<i>US\$ 435</i>
<i>LabVIEW Real-Time Module (1 user)</i>	<i>US\$ 435</i>
<i>LabWindows/CVI</i>	<i>US\$545</i>

LabVIEW Development Systems

LabVIEW for Windows 2000/NT/XP/Me/9x

<i>Full Development System</i>	<i>US\$ 2,195</i>
<i>Professional Development System</i>	<i>US\$ 3,845</i>
<i>NI Developer Suite Professional Edition</i>	<i>US\$ 4,395</i>

LabVIEW for Mac OS

<i>Professional Development System</i>	<i>US\$ 3,845</i>
<i>Full Development System</i>	<i>US\$ 2,195</i>

LabVIEW for Linux

<i>Full Development System</i>	<i>US\$ 2,195</i>
<i>Professional Development System</i>	<i>US \$3,845</i>

LabVIEW for Sun

<i>Full Development System</i>	<i>US\$ 3,295</i>
<i>Professional Development System</i>	<i>US\$ 4,945</i>

LabVIEW Datalogging and Supervisory Control Module

<i>LabVIEW Datalogging and Supervisory Control Module Development System</i>	<i>US\$ 2,195</i>
--	-------------------

3.8.4 Resumen económico de Lookout

<i>Lookout Enterprise Development/Run-Time Server (includes four client licenses) Unlimited I/O</i>	<i>US \$ 9,455</i>
<i>Lookout Enterprise Run-Time Only Server (includes four client licenses) Unlimited I/O</i>	<i>US\$ 6,815</i>
<i>Lookout Development/Run-Time Server (with no client licenses) Unlimited I/O</i>	<i>US\$ 7,255</i>
<i>Lookout Run-Time Only Server (with no client licenses) Unlimited I/O</i>	<i>US\$ 4,615</i>

Lookout Client Licenses

<i>Each additional concurrent client connection</i>	<i>US\$1,095</i>
<i>Lookout Enterprise Server Unlimited Client License</i>	<i>US\$ 10,995</i>

Lookout Server

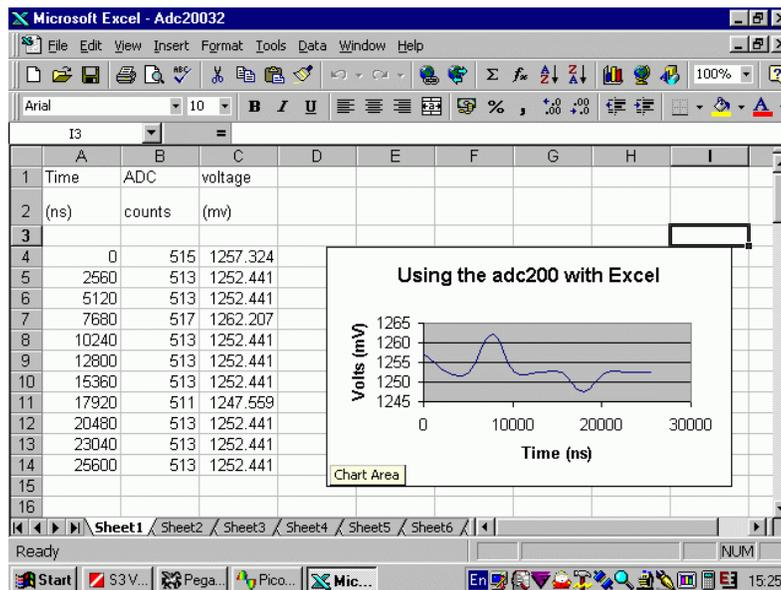
<u><i>Unlimited Client License</i></u>	<u><i>US\$ 13,195</i></u>
<i>Lookout Integrator Software (includes one client connection) Unlimited I/O</i>	<i>US\$ 875</i>
<i>Lookout Client Software Lookout Development/Run-Time</i>	<i>US\$ 875</i>

4. EJEMPLO PRÁCTICO DE INSTRUMENTACIÓN VIRTUAL

4.1 USO DE LA COMPUTADORA PARA CREAR SISTEMAS DE REGISTRO DE DATOS

Tradicionalmente las aplicaciones de registro de datos se han solucionado con equipos de llave en mano conocidos como *data loggers*. Estos equipos son dispositivos aislados que graban señales y las almacenan en memoria o las imprimen en papel. Típicamente los datos de estos dispositivos se tienen que transferir a una computadora porque la mayoría de las aplicaciones requieren también análisis, almacenamiento, presentación y distribución.

Figura 19. Adquisición de datos en una hoja de cálculo



Fuente: Programa *Microsoft Excel*

La transferencia de datos normalmente toma mucho tiempo porque es un proceso manual y además el usuario debe aprender a usar dos sistemas diferentes. Aunque estos dispositivos o dataloggers solucionan ciertas aplicaciones muy bien, hay otras ocasiones donde es más productivo y efectivo el adquirir y almacenar los datos directamente a la computadora.

Las tres ventajas principales de usar sistemas basados en computadora para el registro de datos son:

- 1) La integración del sistema de adquisición de datos o medición a la computadora evita el paso de transferencia de datos del dispositivo aislado a la computadora y esto le ahorra tiempo.
- 2) Las modernas herramientas de *software* de hoy en día le permiten al usuario automatizar sus tareas de registro de datos, incluyendo el almacenamiento, análisis, presentación y distribución.
- 3) Los sistemas de registro de datos basados en computadora le brindan la flexibilidad al usuario de poder usar una amplia variedad de *hardware* para adquisición y acondicionamiento de datos.

Al usar la computadora como la base de un sistema de registro de datos, toma ventaja de una arquitectura abierta y flexible la cual se puede expandir y modificar cuando los requerimientos del sistema cambien.

4.1.1 PARTES DE UN SISTEMA DE REGISTRO DE DATOS

SI SE EXAMINAN LAS PARTES QUE FORMARÍAN UN SISTEMA DE REGISTRO DE DATOS, SE TENDRÍA QUE, DEL LADO DEL *HARDWARE*, HAY TRES COMPONENTES BÁSICOS: LA COMPUTADORA, EL *HARDWARE* DE MEDICIÓN Y LOS SENSORES. LA COMPUTADORA PUEDE SER UNA DE ESCRITORIO, UNA PORTÁTIL *LAPTOP* O UNA COMPUTADORA INDUSTRIAL COMO *PXI* POR EJEMPLO. LOS SENSORES CONVIERTEN LOS PARÁMETROS FÍSICOS A SEÑALES ELÉCTRICAS COMO POR EJEMPLO LOS TERMOPARES, GALGAS, TRANSDUCTORES DE PRESIÓN, ACELERÓMETROS, ETC. EL *HARDWARE* DE MEDICIÓN ES EL COMPONENTE QUE CONVIERTE A LA COMPUTADORA EN UN SISTEMA DE REGISTRO DE DATOS. ALGUNAS CARACTERÍSTICAS QUE SE DEBEN DE BUSCAR EN EL *HARDWARE* DE MEDICIÓN INCLUYEN:

TABLA XVIII. CARACTERÍSTICAS IMPORTANTES DEL *HARDWARE* DE MEDICIÓN

Alta integración de componentes diseñados para que trabajen en conjunto
Sistema modular que permita rápida expansión o modificación
Hardware que soporte varias opciones de tipos de sensores, número de canales y tipo de conectores
Especificaciones detalladas de precisión y calibración que le permitan verificar la calidad de sus datos.
Ajustarse a los estándares ya establecidos

El *software* que se use para un sistema de registro de datos basado en *PC* va a dictar la productividad y flexibilidad del sistema. El *software* que se use deberá integrarse de manera transparente con el *hardware* y deberá proveer las herramientas para configurar el *hardware*, usar escalas en los datos de los canales y calibrar el sistema.

En pocas palabras, el *software* deberá permitir desarrollar la aplicación completa incluyendo la presentación de los datos, análisis y almacenamiento.

**HAY DOS TIPOS DE *SOFTWARE* QUE SE PUEDE USAR
PARA CREAR SISTEMAS DE REGISTRO DE DATOS
BASADOS EN *PC* LOS CUALES SON:**

A) PAQUETES BASADOS EN CONFIGURACIÓN

B) HERRAMIENTAS DE DESARROLLO.

**LOS PAQUETES BASADOS EN CONFIGURACIÓN SON
APLICACIONES LISTAS PARA EJECUTARSE QUE
PROVEEN UN AMBIENTE AMIGABLE PARA QUE SE
PUEDA CONFIGURAR RÁPIDAMENTE LA TAREA DE
REGISTRO DE DATOS. UNA DE LAS DESVENTAJAS DE
USAR APLICACIONES BASADAS EN CONFIGURACIÓN ES**

QUE, A MENOS QUE CUENTEN CON UN MÉTODO PARA PERSONALIZAR, EL SISTEMA ESTARÁ LIMITADO POR LA FUNCIONALIDAD DEFINIDA POR EL FABRICANTE.

Por el otro lado, las herramientas de desarrollo le permiten crear una aplicación personalizada a lo que el usuario exactamente necesite. Con las herramientas de desarrollo se podrá modificar cualquier aplicación a medida que los requerimientos cambien, integrar funciones de análisis y generación de reportes personalizados y automatizar un sistema.

Figura 20. Adquisición de datos en una planta de producción



Fuente: Autores varios. Revista *Instrumentation*. Cuarto trimestre. Pág. 5

Los sistemas de registro de datos basados en *PC* pueden traer muchas ventajas a una aplicación porque integran las tareas de registro, almacenamiento, presentación, análisis y distribución de datos. Estas mejoras ofrecen mayor productividad, flexibilidad y capacidades mejoradas.

5. ADQUISICIÓN DE DATOS Y ACONDICIONAMIENTO DE SEÑALES

5.1 Adquisición de datos

Los sistemas de adquisición de datos *DAQ* basados en *PC* y dispositivos insertables son usados en un amplio rango de aplicaciones en los laboratorios, en el campo y en el piso de una planta de manufactura. La selección de la tarjeta de adquisición de datos y control, y del sistema de acondicionamiento de señales, están estrechamente ligadas.

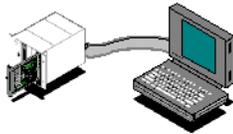
Dependiendo de la cantidad y tipo de señales, el tipo de adquisición ya sea local o remoto y la velocidad con se necesita adquirir las señales, entre otros factores, se puede utilizar diferentes sistemas de acondicionamiento de señales y tarjetas de adquisición de datos. Otros factores importantes son, la resolución deseada por ejemplo 12, 16 o 24 *bits* para la conversión analógica - digital, y la necesidad o no de aislamiento por cada señal. Recordemos que para definir la velocidad de muestreo requerida para la tarjeta de adquisición de datos, debemos referirnos al teorema de *Nyquist*, el cuál nos indica que la misma debe ser al menos dos veces la velocidad de la señal que tenga la frecuencia más rápida a muestrear.

5.1.1 Sistema de adquisición de datos local

Este tipo de sistema es normalmente implementado en una computadora de escritorio o una portable. Se dice que es local, porque la tarjeta de adquisición de datos reside dentro de la computadora.

Está conectada a uno de sus puertos seriales/paralelo, y el sistema de acondicionamiento de señales está directamente conectado a la computadora, a corta distancia, menos de 10 metros.

FIGURA 21. MODULO DE ADQUISICIÓN DE DATOS



FUENTE: *INDUSTRIAL SOLUTIONS*, PÁG. 42.

TABLA XIX. OPCIONES DISPONIBLES PARA UN SISTEMA DE ADQUISICIÓN DE DATOS LOCAL

Tipo de PC	PC (XT)	AT	PCI	PXI	PCMCIA	Puerto Serial	Puerto Paralelo
Portable					X	X	X
Escritorio	X	X	X	X (i)	X (ii)	X(iii)	X

Notas:

(i) Sólo con computador *PXI*.

(ii) Se requiere de adaptador *PCMCIA* para *PC*.

(iii) Incluye el nuevo puerto *USB*, disponible en las *PC* de fabricación reciente.

Las tarjetas *PC* (para el bus *XT*) son típicamente de menor velocidad de muestreo, tienden a desaparecer o ser discontinuadas. Además, su funcionalidad es limitada, requieren configuración manual del nivel de interrupción, canales *DMA* y direcciones de memoria, en comparación con las tarjetas *PCMCIA*, *PXI* y *PCI* que son totalmente *Plug-and-play* o auto configurables.

Las tarjetas *AT* (para el bus *ISA*) tienen una mayor capacidad funcional que las de la serie *PC*, sin llegar al rendimiento y funcionalidad de las tarjetas

PCI o *PXI*, y sin permitir la configuración automática *Plug-and-play* o *PnP* que el bus *PCI* o el *PCMCIA* permiten. Estas también tienden a desaparecer o ser discontinuadas, a medida que el bus *PCI* y otras opciones se convierten en los nuevos estándares.

Las tarjetas *PCI* (para el bus *PCI*) ofrecen el mayor rendimiento y funcionalidad posible, gracias al ancho de banda del bus *PCI* (133 *Mbytes/seg.*) y las facilidades *Plug-and-play* o *PnP* que este bus ofrece.

Las tarjetas *PXI* son similares a las *PCI* pero están especialmente diseñadas para ser utilizadas en el nuevo bus *PXI*, compatible con el bus *CompactPCI*. La especificación *CompactPCI* es la versión industrial y más robusta del bus *PCI*. Por otro lado, las tarjetas *DAQCard*, para el bus *PCMCIA*, son la opción para los sistemas de adquisición de datos y control basados en computadoras portables o *laptops*, y las tarjetas *DAQPAD* están diseñadas para utilizar el puerto paralelo *EPP* de la *PC*.

Por último, queda la opción de utilizar el puerto serial de la *PC* (*RS-232*) para adquirir datos y generar señales de control, siempre a menor velocidad que todas las opciones anteriores. En este caso, las opciones disponibles son el sistema de acondicionamiento modular *Fieldpoint*, y la serie 6B; ambos digitalizan las señales y las envían a la *PC* vía puerto serial *RS-232*.

Figura 22. Sistema *Fieldpoint*



Fuente: Catálogo *Measurement and automation*, Pág. 573.

Figura 23. Sistema de la Serie 6B



Fuente: Catálogo, *Measurement and automation*, Pág. 570.

Ambos sistemas son completamente modulares, contando con módulos para señales de entradas y salidas analógicas (4-20mA) y digitales o discretas (0-24Vdc). También existen módulos para señales de termopares, *RTD*, etc., así como salidas discretas de potencia (24Vdc, 240 Vdc, etc.). Estos sistemas son en sí mismos sistemas de acondicionamiento de señales, y como tales no requieren de accesorios o acondicionadores adicionales.

La principal diferencia entre ambos, es que el sistema *Fieldpoint* utiliza módulos de varios canales (en lugar de un canal por módulo, como sucede con el sistema 6B), y maneja directamente señales digitales (el sistema 6B requiere de los módulos *SSR* para el manejo de señales digitales).

Adicionalmente, el sistema *Fieldpoint* es más compacto y económico (para más de 16 entradas analógicas), y ofrece posibilidades mayores de crecimiento o expansión.

5.1.2 Sistema de acondicionamiento de señales para tarjetas de adquisición de datos

Se mencionó anteriormente que el sistema *Fieldpoint* es en sí mismo un sistema de acondicionamiento de señales que no requiere de una tarjeta de adquisición de datos para funcionar. Sin embargo, todas las otras opciones basadas en tarjetas de adquisición de datos para el bus *XT*, *ISA*, *PCI*, *PXI* o *PCMCIA*, requieren normalmente de un sistema de acondicionamiento de señales específico para cada tipo de señal, tales como las generadas por los siguientes sensores o transductores, que se consideran estándares industriales:

- Termopares (tipo K, J, T, etc.)
- *RTD* (por ejemplo: Pt-100)
- Termistores
- Galgas de deformación (*Strain Gages*)
- 4-20 mA, etc.

Las bondades de un sistema de acondicionamiento de señales son muchas, entre las cuáles se encuentran las siguientes:

Filtraje de ruido, aislamiento, atenuación y/o amplificación de la señal, linealización de la señal por *hardware*, por ejemplo termopares. Alimentación de lazos de control y/o *RTD*, *strain-gages*, multiplexación de señales, etc.

SE DICE QUE EL SISTEMA DE ACONDICIONAMIENTO DE SEÑALES ES OPCIONAL PORQUE, SI LA SEÑAL A MEDIR SE ENCUENTRA EN EL RANGO DE +/- 5VDC Y NO SE REQUIERE AISLAMIENTO, LA MISMA SE PUEDE CONECTAR DIRECTAMENTE A LA TARJETA DE ADQUISICIÓN DE DATOS POR MEDIO DE UN BLOQUE DE BORNAS Y UN CABLE EXTERNO. EL RIESGO EN ESTE CASO ES QUE LA TARJETA NO CONTARÍA CON AISLAMIENTO.

Sin embargo, cuando se desea contar con las bondades del acondicionamiento de señales, y/o el tipo de señales a medir así lo requiere, se recomienda agregarle a la tarjeta de adquisición de datos un sistema de acondicionamiento de señales externo.

Adicionalmente a las señales ya comentadas, también existen necesidades de capturar y procesar otros tipos de señales, tales como:

- audio, vídeo, movimiento, etc.

Tabla XX. Algunas de las opciones de sistemas disponibles

Nombre	Imagen	Ventajas	Desventajas
--------	--------	----------	-------------

5B Series		Sólo para señales analógicas, modular, un módulo por señal.	Alto costo por señal para menor de 16 AI, módulos dedicados, ruido, no es portable.
SSR		Sólo para señales digitales, modular, un módulo por señal.	Alto costo por señal para menor de 24 DI, módulos dedicados, no es portable.
SCC		Sólo para señales analógicas, modular, un módulo por señal, es portable	Alto costo por señal para menor de 16 AI, módulos dedicados, ruido.
SCXI		Para todo tipo de señales, un módulo acondiciona diferentes señales, compacto, poco ruido, robusto, alta-precisión, bajo costo por canal para mas de 16 AI, semi-portable.	Alto costo por canal para menos de 16 AI, pesado, utiliza 1 puerto digital de la tarjeta DAQ para comunicación, velocidad limitada.

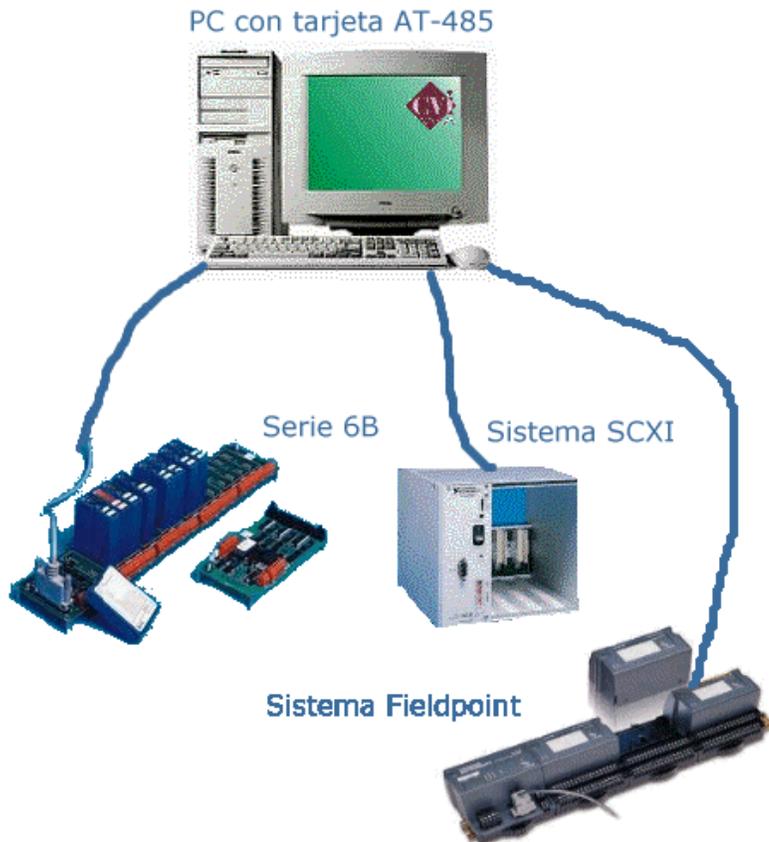
Fuente: Catálogo, *Measurement and automation*, páginas 196, 246, 320, 347.

5.1.3 Sistema de adquisición de datos remoto

En el punto anterior, se explicó lo referente a los sistemas locales de adquisición de datos, aquellos en los cuáles la computadora y el sistema de acondicionamiento de señales están a corta distancia, y las tarjetas de adquisición de datos residen dentro de la computadora o se conectan con ella a través del puerto serial *RS-232* o el puerto paralelo.

Ahora veamos lo referente a los sistemas remotos de adquisición de datos, aquellos en donde la computadora y el sistema de adquisición y acondicionamiento de señales están distantes uno del otro a más de 20 metros.

Figura 24. Ejemplo de adquisición de datos remoto



Fuente: *Data Acquisition and signal conditioning*, Pág. 14.

Un sistema de adquisición de datos remoto se basa en una conexión serial *RS-485* o *Ethernet*. El estándar *RS-485* permite conectar hasta 32 dispositivos o chasis de adquisición y/o acondicionamiento de datos conectados en forma multicáda a lo largo de un mismo cable de 2 ó 4 hilos trenzados.

En el estándar *RS-485*, el cable que interconecta todos los dispositivos de adquisición/acondicionamiento de señales puede tener una longitud de 467 metros (1400 pies), lo cual permite que la computadora se encuentre bien distante de los módulos de acondicionamiento de señales. Utilizando un módem, esta distancia puede ser ilimitada.

Se puede mencionar tres opciones para los sistemas remotos de adquisición de datos basados en el estándar *RS-485*:

- Sistema *Fieldpoint* (modo *RS-485*)
- Serie 6B (modo *RS-485*)
- Sistema *SCXI*

En cualquiera de los tres casos, se requiere que la computadora cuente con un puerto *RS-485*, por ejemplo la tarjeta *AT/PCI/PCMCIA-485*, mediante el cuál se realiza la conexión con los módulos de acondicionamiento de señales remotas.

Para seleccionar cualquiera de estos sistemas, se puede referir a las ventajas y desventajas presentadas en el punto anterior, siendo algunas de las diferencias entre las tres opciones el hecho de que un módulo de acondicionamiento de señales del sistema *SCXI* puede alcanzar a acondicionar diferentes tipos de señales, ofrecen mayor precisión y amplificación de señales, adicionalmente, el sistema *SCXI* permite una mayor capacidad de expansión.

A diferencia de esto, cada módulo del sistema *Fieldpoint* sólo acondiciona un mismo tipo de señal, aunque cada módulo maneja varias señales del mismo tipo en forma simultánea. Por último, y como ya se mencionó, la serie 6B requiere de un módulo por cada señal.

5.2 Acondicionamiento de transductores

Figura 25. Acondicionamiento de señales



Fuente: *Industrial solutions*, Pág. 40.

Típicamente, los dispositivos de adquisición de datos *DAQ* insertables son instrumentos de propósito general diseñados para medir señales de voltaje.

El problema es que la mayoría de los sensores y transductores generan señales que debe acondicionar antes de que un dispositivo de adquisición *DAQ* pueda adquirir con precisión la señal. Este procesamiento al frente, conocido como acondicionamiento de señal, incluye funciones como amplificación, filtrado, aislamiento eléctrico y multiplexación. Así la mayoría de los sistemas de adquisición *DAQ* basados en computadora incluyen algún tipo de acondicionamiento de señal además del dispositivo de adquisición *DAQ* y la *PC*.

SISTEMAS DE CONMUTACIÓN AL FRENTE TAMBIÉN INCREMENTAN LA FUNCIONALIDAD DE SU SISTEMA DE MEDICIÓN Y AUTOMATIZACIÓN. CONMUTACIÓN DE PROPÓSITO GENERAL LE PROPORCIONA CONTROL DIGITAL DE LA PRESENCIA O AUSENCIA DE UNA SEÑAL EN EL SISTEMA, COMO LA ALIMENTACIÓN A UN MOTOR.

CONFIGURACIONES DE MULTIPLEXORES/MATRIZ DE RELEVADORES CONTROLAN LA FUENTE Y RUTA DE LAS

SEÑALES EN SU SISTEMA O ACTÚAN COMO MULTIPLEXORES PARA DISPOSITIVOS COMO MULTÍMETROS DIGITALES *DMMS*.

5.2.1 Acondicionamiento de señales

LOS TRANSDUCTORES SON DISPOSITIVOS QUE CONVIERTEN FENÓMENOS FÍSICOS COMO TEMPERATURA, CARGA, PRESIÓN O LUZ A SEÑALES ELÉCTRICAS COMO VOLTAJE Y RESISTENCIA. LAS CARACTERÍSTICAS DE LOS TRANSDUCTORES DEFINEN MUCHOS DE LOS REQUERIMIENTOS DE ACONDICIONAMIENTO DE SEÑALES DE UN SISTEMA DE ADQUISICIÓN *DAQ*.

La mayoría de las señales requieren de preparación antes de poder ser digitalizadas. Por ejemplo, una señal de una termopar es muy pequeña y necesita ser amplificada antes de pasar por el digitalizador. Otros sensores como *RTD*, termistores, galgas extensiométricas y acelerómetros requieren de poder para operar. Aún las señales de voltaje puro pueden requerir de tecnología para bloquear señales grandes de modo común o picos. Todas estas tecnologías de preparación son formas de acondicionamiento de señal.

Puesto que existe un amplio rango de tecnologías, el papel que desempeña y la necesidad de cada una de estas se puede volver confuso. A continuación se proporciona una guía de los tipos de acondicionamiento de señal más comunes, su funcionalidad y ejemplos de cuando usarlos.

5.2.2 Aislamiento de señales

Las señales de voltaje fuera del rango del digitalizador pueden dañar el sistema de medición y ser peligrosas para el operador. Por esta razón,

normalmente es preciso tener el aislamiento y la atenuación para proteger al sistema y al usuario de voltajes de alta tensión o picos.

También se puede necesitar aislamiento si el sensor está en un plano de tierra diferente al del sensor de medición (como un termopar montado en una máquina).

5.2.3 Amplificación de señales

Cuando los niveles de voltaje que va a medir son muy pequeños, la amplificación se usa para maximizar la efectividad de su digitalizador. Al amplificar la señal de entrada, la señal acondicionada usa más efectivamente el rango del convertidor analógico – digital *ADC* y mejora la precisión y resolución de la medición. Algunos sensores que típicamente requieren de amplificación son los termopares y galgas extensiométricas.

5.2.4 Atenuación de señales

La atenuación es lo opuesto a la amplificación. Es necesario cuando el voltaje que se va a digitalizar es mayor al rango de entrada del digitalizador. Esta forma de acondicionamiento de señal disminuye la amplitud de la señal de entrada, para que la señal acondicionada este dentro del rango del *ADC*. La atenuación es necesaria para medir voltajes altos.

5.2.5 Multiplexión de señales

Típicamente, el digitalizador es la parte más costosa del sistema de adquisición de datos. Al multiplexar, usted puede rutear secuencialmente un

cierto número de señales a un solo digitalizador, logrando así un sistema de bajo costo y extendiendo el número de conteo de señales de un sistema. Multiplexar las señales es necesario para cualquier aplicación de alto conteo de canales.

5.2.6 Filtrado de señales

Los filtros son necesarios para remover cualquier componente de frecuencia no deseada en una señal, principalmente para prevenir *aliasing* y reducir la señal de ruido. Algunas mediciones de termopares generalmente requieren de filtros pasa bajos para remover el ruido de las líneas de poder. Las mediciones de vibración normalmente requieren de filtros *antialiasing* para remover componentes de señales más allá del rango de frecuencias del sistema de adquisición de datos.

5.2.7 Excitación de transductores

Muchos sensores, como *RTD*, galgas y acelerómetros, requieren de alguna fuente de poder para hacer la medición. La excitación es la tecnología de acondicionamiento de señal requerida para proveer esa fuente. Esta excitación puede ser voltaje o corriente dependiendo del tipo de sensor.

5.2.8 Linearización de señales

Algunos tipos de sensores producen señales de voltaje que no son lineales en relación con la cantidad física que están midiendo. La linearización, es el

proceso de interpretar la señal del sensor como una medición física, puede realizarse a través de acondicionamiento de señal o *software*. Los termopares son un ejemplo típico de un sensor que requiere linearización.

5.2.9 Compensación de junta fría de transductores

Otra tecnología requerida para mediciones de termopares es la compensación de junta fría *CJC*. Siempre que se conecta un termopar a un sistema de adquisición de datos, la temperatura de la conexión debe ser conocida para poder calcular la temperatura verdadera que el termopar está midiendo. Un sensor *CJC* debe estar presente en el lugar de las conexiones.

5.2.10 Muestreo simultáneo de señales

Cuando es crítico medir dos o más señales en un mismo instante, el muestreo simultáneo es indispensable. El acondicionamiento de señal apropiado al frente de un sistema es una opción de bajo costo para realizar esta operación sin tener que comprar un digitalizador para cada canal. Algunas de las aplicaciones que podrían requerir de muestreo simultáneo incluyen las mediciones de vibración y mediciones de diferencias de fase.

5.3 Tipos de transductores típicos

5.3.1 Termopares o termocuplas

El transductor más utilizado para medir temperatura es el termopar o termocupla. Aunque el termopar es económico, resistente y puede operar en un amplio rango de temperaturas, el termopar requiere de acondicionamiento de señal especial.

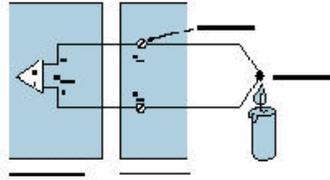
Un termopar opera bajo el principio de que una junta de metales no similares genera un voltaje que varía con la temperatura.

Además al conectar el cable del termopar al cable que lo conecta al dispositivo de medición se crea una junta termoeléctrica adicional conocida como junta fría. Entonces el voltaje medido, V_{MES} incluye el voltaje del termopar y los voltajes de junta fría (V_{CJ}). El método para compensar estos voltajes de junta fría no deseados es conocido como compensación de junta fría.

La mayoría de los productos de acondicionamiento de señal compensan las juntas frías usando un sensor adicional, como un termistor o sensor *IC*. Este sensor es colocado en el conector de señales o bloque terminal para medir la temperatura ambiente en la junta fría directamente. El *software* después puede calcular la compensación apropiada para los voltajes termoeléctricos indeseados.

Sensibilidad y ruido son otros factores importantes a considerar cuando se miden termopares.

Figura 26. Termopar



La conexión del cable de un termopar al sistema de medición crea una junta termoeléctrica adicional, llamada junta fría. Esta debe de ser compensada con acondicionamiento de señal.

Fuente: Catálogo, *Measurement and automation*, Pág. 780.

Las salidas de los termopares son muy pequeñas y cambian de 7 a $50\mu\text{V}$ por cada grado ($1\text{ }^{\circ}\text{C}$) de cambio en temperatura haciendo a las señales muy susceptibles a los efectos de ruido eléctrico. Por esto los acondicionadores de termopares incluyen filtros de ruido paso bajo para suprimir el ruido de 50 y 60 Hz. Además incluyen amplificadores de instrumentación de alta ganancia para aumentar el nivel de la señal.

Amplificar la señal del termopar también incrementa la resolución o sensibilidad de la medición. Por ejemplo, un típico dispositivo de adquisición con un rango de entrada de convertidor analógico digital ADC de $\pm 10\text{ V}$ y una ganancia en tarjeta de 50 tiene una resolución de $98\text{ }\mu\text{V}$. Esto corresponde a aproximadamente $2\text{ }^{\circ}\text{C}$ para un termopar tipo J o K.

Al añadir un acondicionamiento de señal con una ganancia adicional de 100, la resolución incrementa a $1\mu\text{V}$, lo cual corresponde a una fracción de un grado *Celsius*.

5.3.2 RTD (resistance temperature detector)

OTRO POPULAR SENSOR DE TEMPERATURA ES EL DETECTOR DE TEMPERATURA-RESISTENCIA *RESISTANCE-TEMPERATURE DETECTOR (RTD)*, UN DISPOSITIVO QUE INCREMENTA SU RESISTENCIA CON LA TEMPERATURA. EL TIPO DE *RTD* MÁS UTILIZADO ESTA HECHO DE PLATINO Y CUENTA CON UNA RESISTENCIA NOMINAL DE 100 Ω A 0 °C. PORQUE EL *RTD* ES UN DISPOSITIVO RESISTIVO, SE DEBE DE PASAR CORRIENTE A TRAVÉS DEL *RTD* PARA PRODUCIR UN VOLTAJE QUE EL DISPOSITIVO DE ADQUISICIÓN *DAQ* PUEDA LEER.

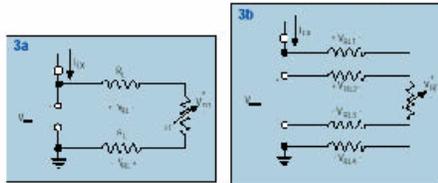
CON RESISTENCIAS RELATIVAMENTE BAJAS (100 Ω) QUE CAMBIAN MUY POCO CON LA TEMPERATURA (MENOS DE 0.4 $\Omega/^\circ\text{C}$), LOS *RTDS* REQUIEREN DE ACONDICIONAMIENTO DE SEÑAL CON FUENTES DE EXCITACIÓN DE CORRIENTE ALTAMENTE PRECISAS, AMPLIFICADORES DE ALTA GANANCIA Y CONECTORES PARA MEDICIONES DE 4 Y 3 HILOS.

Por ejemplo, una medición de *RTD* de 2 incluye errores de caída de voltaje causados por la corriente de excitación pasando por la punta resistiva R_L . Estos errores, los cuales pueden ser significativos se eliminan usando un *RTD* de 4 hilos. La configuración de 4 hilos usa un segundo par de cables para pasar la corriente de excitación. De esta manera una corriente casi insignificante fluye a través de los cables del sensor y así el error de punta resistiva es muy pequeño.

La contrapartida es que se necesita el doble de cables y el doble de canales de adquisición de datos. La técnica de 3 hilos ofrece una solución

intermedia que elimina un cable, pero no es tan precisa. Aunque es la más ampliamente utilizada.

Figura 27. RTD



Errores causados por la resistencia de punta R_p , se pueden minimizar usando un RTD de cuatro hilos.

Fuente: Catálogo, *Measurement and automation*, Pág. 782

Tabla XXI. Comparación de transductores de temperatura

	RTD	Termistor	Sensor de IC	Termopar
Ventajas	Más estable. Más preciso. Más lineal que los Termopares.	Alto rendimiento Rápido Medida de dos hilos	El más lineal El de más alto rendimiento Económico	Autoalimentado Robusto Económico Amplia variedad de formas físicas Amplia gama de temperaturas
Desventajas	Caro. Lento. Precisa fuente de Alimentación. Pequeño cambio de resistencia. Medida de 4 hilos Autocalentable	No lineal. Rango de Temperaturas limitado. Frágil. Precisa fuente de alimentación. Autocalentable	Limitado a $< 250\text{ }^{\circ}\text{C}$ Precisa fuente de alimentación Lento Autocalentable Configuraciones limitadas	No lineal Baja tensión Precisa referencia El menos estable El menos sensible

EN LA ANTERIOR TABLA SE COMPARAN LOS CUATRO TIPOS DE TRANSDUCTORES DE TEMPERATURA MÁS UTILIZADOS, Y REFLEJA LOS FACTORES QUE DEBEN TENERSE EN CUENTA: LAS PRESTACIONES, EL ALCANCE EFECTIVO, EL PRECIO Y LA COMODIDAD.

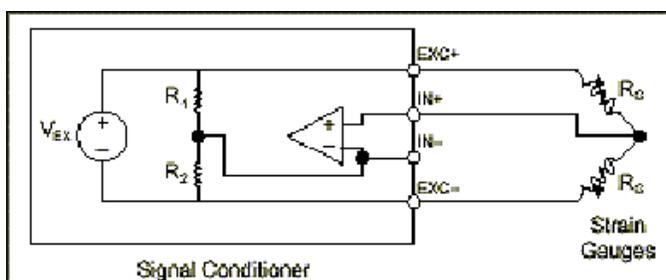
5.3.3 Galgas extensiométricas

La galga extensiométrica es un dispositivo comúnmente usando en pruebas y mediciones mecánicas. La galga más común, la galga extensiométrica de resistencia, consiste de una matriz de bobinas o cable muy fino el cual varia su resistencia linealmente dependiendo de la carga aplicada al dispositivo.

Cuando se usa una galga extensiométrica, se pega la galga directamente al dispositivo bajo prueba, aplica fuerza y mide la carga detectando los cambios en resistencia. Las galgas extensiométricas también son usados en sensores que detectan fuerza, aceleración, presión y vibración.

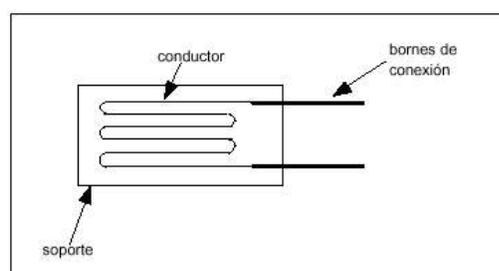
Las mediciones de galgas extensiométricas involucran sensar cambios muy pequeños de resistencia. Por esto una selección apropiada y el uso de un puente, acondicionamiento de señal, cableado y adquisición de datos son componentes requeridos para lograr mediciones precisas.

Figura 28. Conexión de galgas extensiométricas



Fuente: Catálogo *Measurement and automation*, Pág. 784.

Figura 29. Galgas extensiométricas



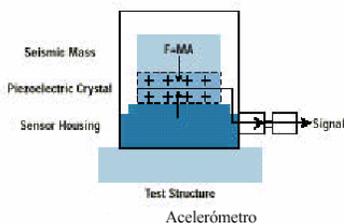
Fuente: Catálogo *Measurement and automation*, Pág. 785.

5.3.4 Acelerómetros

Los acelerómetros son dispositivos usados para medir aceleración y vibración. El dispositivo consiste de una masa conocida pegada a un elemento piezoeléctrico. A medida que el acelerómetro se mueve, la masa aplica fuerza al cristal generando una carga. Al leer esta carga se puede determinar la aceleración. Los acelerómetros son direccionales, esto quiere decir que solo miden aceleración en un eje. Para monitorear aceleración en tres dimensiones, use un acelerómetro multieje.

Existen acelerómetros de dos tipos, pasivos y activos. Los acelerómetros pasivos envían la carga generada por el elemento piezoeléctrico. Ya que la señal es muy pequeña, los acelerómetros pasivos requieren de un amplificador para amplificar la señal. Los acelerómetros activos incluyen circuitería interna para convertir la carga del acelerómetro a una señal de voltaje, pero requieren de una fuente constante de corriente para alimentar el circuito.

Figura 30. Acelerómetros



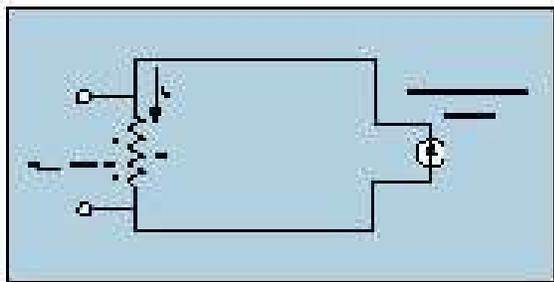
Fuente: Catálogo, *Measurement and automation*, Pág. 786

5.3.5 Señales de corriente

Muchos de los dispositivos y transmisores que se usan en aplicaciones de control y monitoreo de procesos generan una señal de corriente, normalmente de 0 a 20 mA o de 4 a 20 mA. Las señales de corriente se usan porque son menos propensas a los errores causados por ruido o caídas de voltaje en cables que son muy largos. Los acondicionadores de señal convierten las señales de corriente a señales de voltaje al pasar la corriente a través de una resistencia de precisión.

El voltaje que resulta ($V_{MEAS} = I_S R$) después puede ser digitalizado.

Figura 31. Señales de corriente



Fuente: Catálogo, *Measurement and automation*, Pág. 788.

6. SIMULACIÓN Y CONTROL DE PROCESOS CON PROGRAMACIÓN GRÁFICA

La modelación y simulación se ha constituido en una poderosa herramienta para el diseño, análisis y optimización de sistemas y procesos industriales. La disponibilidad de computadoras personales cada vez más poderosas, de menor costo y de fácil uso, acompañados por *software* ó programas de aplicación y lenguajes de programación altamente flexibles, ha permitido la masificación del uso de diferentes técnicas de simulación y control de procesos.

Este trabajo se concentra en la aplicación de las técnicas de modelaje, simulación y control de procesos haciendo uso de la tecnología *PC* y un lenguaje de programación 100% gráfico que permite la construcción de modelos mediante un paradigma de programación con “iconos interconectados por cables gráficos”, cuyo código es ejecutado sobre la base del principio de “flujo de datos” por los “cables” que interconectan los diferentes “iconos” o funciones.

Combinando este lenguaje de programación 100% gráfico con tarjetas de adquisición y control (entradas/salidas) para la computadora, se puede entonces configurar un sistema de simulación y control capaz de manejar señales reales (analógicas y digitales, de entrada y de salida) y ejecutar modelos matemáticos de control de procesos en tiempo real.

La simulación de sistemas nos permite, entre otras cosas:

- Predecir el resultado de las acciones que se tomen sobre el proceso o sistema de control.

- Comprender porqué los eventos ocurren.
- Identificar áreas problemáticas antes de la implantación del sistema.
- Explorar los efectos de las modificaciones.
- Evaluar ideas y su viabilidad, e identificar sus ineficiencias.
- Estimular el pensamiento creativo y entrenar al personal.
- Optimizar los procesos (ahorros de energía, cuellos de botella, mejoras de los rendimientos, etc.)

Lo más temprano que se utilice la simulación de sistemas en un proyecto de automatización u optimización de procesos, mayor es el potencial de ahorro y efectividad que se logrará con dicho sistema. Esto se debe a que según estudios realizados, se ha podido confirmar que en un proyecto de automatización y control, más del 80% de los costos del sistema de control ya están comprometidos en la fase de diseño. Es decir, si se logra optimizar el sistema de control en su etapa de diseño, la reducción en los costos asociados con el mismo y su efectividad de maximizar.

Figura 32. Porcentaje de presupuesto contra tiempo



Un mejor diseño se traduce entonces en un mejor producto, menores costos, mayor control, mayor seguridad, menor variabilidad del proceso, etc. Adicionalmente, a través de la simulación de sistemas se minimiza el ensayo y error, ya que se trabaja con modelos matemáticos de comportamiento predecible y controlable por los diferentes algoritmos de control que se apliquen al proceso real o simulado.

Para poder simular un sistema de control, se debe contar con un “modelo” matemático de la planta o proceso que se quiere controlar, el cual puede optimizarse vía interacciones múltiples, bajo múltiples escenarios, variables, parámetros que se evalúan individual ó simultáneamente.

En un modelo matemático, las entidades del sistema y sus atributos se representan mediante variables matemáticas. Las actividades se describen mediante funciones matemáticas que interrelacionan las variables. Los modelos matemáticos dinámicos normalmente se resuelven mediante métodos numéricos, y la simulación es una de las herramientas más comunes.

Un modelo no es solo el sustituto de un sistema, sino también una simplificación del mismo. Para obtener un modelo se debe al menos:

- Determinar su estructura (entidades, atributos, actividades y fronteras).
- Definir los datos (valores de los atributos y relaciones involucradas en las actividades).

Existe una estrecha interrelación entre el proceso de recolección de datos para el modelo, y el análisis de los mismos: las suposiciones relativas al sistema orientan la recolección de datos y el análisis de éstos confirma o refuta las suposiciones. Los pasos básicos para modelar un sistema de control son:

- Definir el modelo matemático y programar el sistema de control.
- Seleccionar los parámetros y los valores iniciales.
- Ejecutar el modelo.
- Procesar los resultados (incluye la visualización y la investigación adicional).
- Interactuar con el modelo hasta lograr optimización.

Con el modelaje y simulación de los sistemas de control, se busca satisfacer la necesidad de controlar la fabricación de los productos mediante el mantenimiento de las variables de proceso en los valores más estables posibles.

Los controladores comparan las variables de proceso con el punto de consigna *setpoint* y a partir de su diferencia o error calculan cual debe ser la señal de salida al elemento final de control. Para controlar el proceso, se utiliza el controlador más apropiado:

- Relé (*On-Off*).
- *Ratio*.

- *PID* (proporcional, integral, diferencial).
- *Fuzzy Logic* (Lógica difusa).
- Otros (adaptivo, algoritmos genéticos, redes neuronales, etc.)

Los procesos industriales son simulados de dos formas: utilizando la teoría clásica de control o mediante el concepto de espacio de estado.

LA ARQUITECTURA *PC* Y LA PROGRAMACIÓN GRÁFICA NOS PERMITEN REALIZAR PRUEBAS EN TIEMPO REAL CON SEÑALES SIMULADAS, REALES (“*HARDWARE IN THE LOOP*” TESTING) O HÍBRIDAS (SIMULADAS O REALES).

6.1 La simulación de sistemas de control en la empresa

Existen cuatro áreas básicas en toda empresa manufacturera que se pueden beneficiar del modelaje y simulación de sistemas

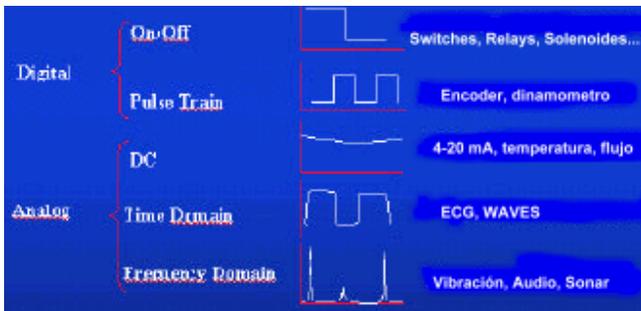
- Diseño (Ingeniería).
- **OPERACIONES (PRODUCCIÓN).**
- Control de calidad (pruebas y medidas).
- Mantenimiento.

Figura 33. Áreas de una industria



Para que un lenguaje de programación pueda ser utilizado exitosamente en aplicaciones de modelaje y simulación de sistemas de control, debe contar con una gran variedad de funciones matemáticas, gráficas y de procesamiento de señales.

Figura 34. Tipos de Señales eléctricas



Tal lenguaje, debe ser fácil de aprender y usar, y debe permitir tanto el manejo de señales reales de tipo analógico y digital, como señales simuladas. A estos tipos de señales eléctricas, debemos agregar las señales de vídeo (analógico y digital) que generan las cámaras utilizadas en los sistemas de visión de máquinas y captura de imágenes en tiempo real, y las señales de control de movimiento (analógica y digital) utilizadas para controlar motores de paso y servo.

De esa forma, el lenguaje de simulación puede ser utilizado en cualquiera de las cuatro áreas críticas antes mencionadas: ingeniería, operaciones, control de calidad y mantenimiento.

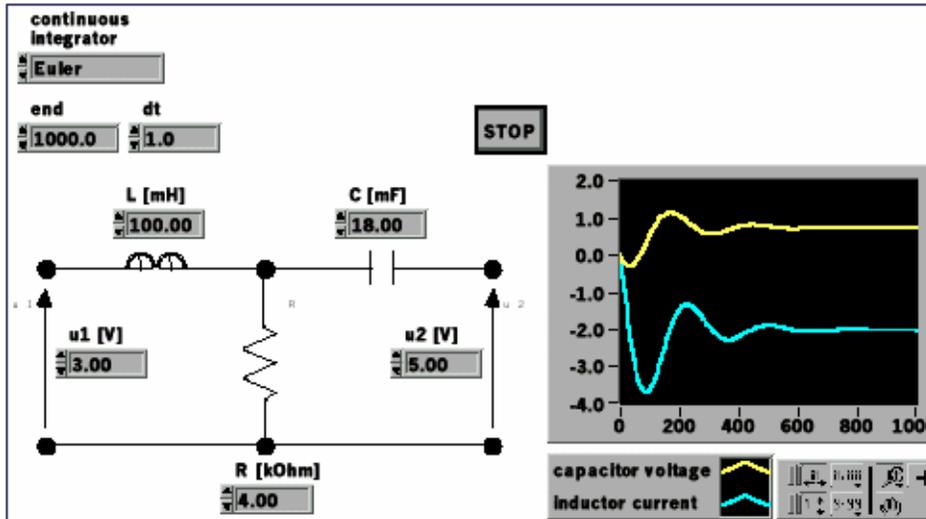
Los sistemas de simulación tradicionales, utilizan principalmente señales simuladas a través de funciones o datos precapturados, mientras que los lenguajes de programación permiten el uso de señales reales analógicas y digitales, junto a las simuladas.

El manejo de señales reales con los modelos de simulación, permite que el modelo sea optimizado y ajustado en tiempo real, de acuerdo a los patrones de comportamiento de las señales de entrada y salida del sistema. Es decir, el modelo puede ser ajustado sobre la base de la respuesta del proceso o sistema real a las señales de estímulo generadas por el modelo mismo.

Esto hace que la transición del modelo/simulación hacia el sistema final sea rápida y efectiva, disminuyéndose el número de interacciones normalmente requeridas para adaptar o implantar el modelo en el proceso real.

Por ejemplo, se podría modelar y simular un circuito RLC completamente en *software*, y luego construir físicamente el circuito RLC y aplicarle el modelo de control previamente simulado.

Figura 35. Simulación de un circuito RLC

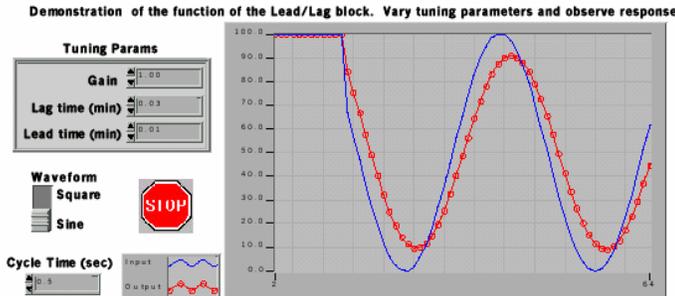


Fuente: Programa DAQ designer 2002.

Bajo este esquema, la transición del modelo simulado al modelo físico, es casi transparente, y solo requiere de la sustitución de la “planta” o proceso simulado, por uno real. El sistema de control sería el mismo para ambos casos, y puede ser validado inmediatamente tanto con el modelo físico como con el proceso real.

Igualmente, si queremos aplicarle un controlador *PID* o un *Lead-Lag* a un proceso simulado, podemos primero aplicárselo al modelo simulado (función de transferencia, ecuación diferencial, ecuaciones de espacio de estado, etc.), y posteriormente al proceso real.

Figura 36. Simulación de un proceso *PID*



Fuente: Programa *DAQ designer 2002*.

Si se demuestra que el controlador *PID* no es el adecuado, o si simplemente se desea evaluar otros controladores sobre el mismo sistema o proceso, se puede entonces aplicar un controlador de lógica difusa, u otros algoritmos de control (genéticos, redes neuronales, control adaptivo, etc.)

6.2 Sistemas que utilizan programación gráfica

Existen muchos lenguajes de programación en el mercado. Algunos de ellos utilizan la denominación “visual” en su nombre (Ej. *Visual C*, *Visual Basic*, etc.). Lo cierto es que la mayoría de estos lenguajes son no más del 50% visuales, ya que están basados en una combinación de objetos gráficos y texto.

Sin embargo, sí existen lenguajes de programación 100% gráficos, es decir, lenguajes que permiten programar y desarrollar una aplicación sin utilizar texto. Programas que permiten la programación sin texto, utilizando sólo iconos, cables gráficos y controles o indicadores (objetos gráficos).

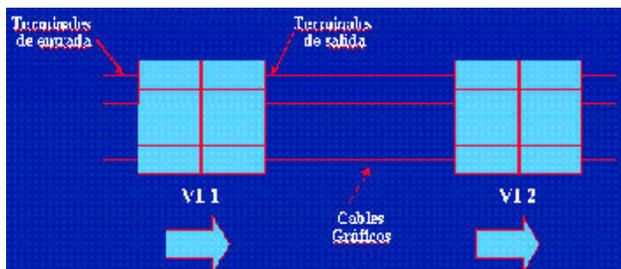
El principio o paradigma de programación utilizado por este lenguaje gráfico se basa en cuatro elementos básicos:

- El uso de iconos, cables gráficos y controles-indicadores.

- Programación sobre la base de un diagrama de bloques.
- El uso de dos ventanas de programación: panel de controles y panel del diagrama.
- Ejecución del programa sobre la base del flujo de datos, en forma paralela.

La programación con diagramas de bloques se basa en iconos o bloques gráficos que se interconectan entre sí a través de “cables gráficos”.

Figura 37. Diagrama de bloques de programación gráfica



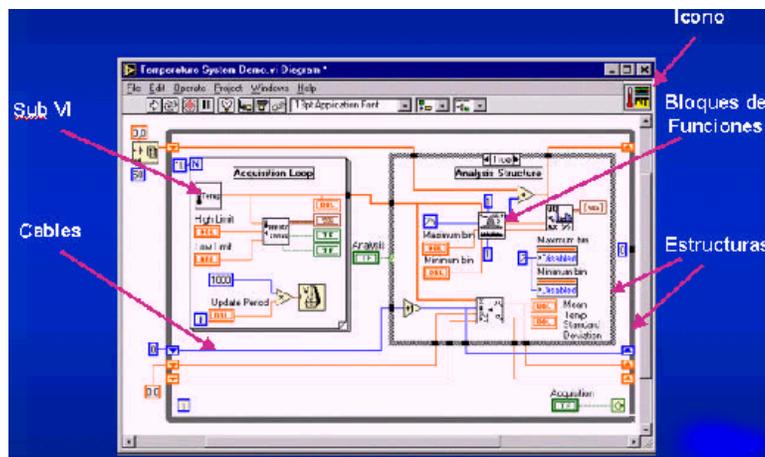
Los datos “fluyen” entre los bloques o funciones a través de los “cables gráficos”. Cada bloque o función cuenta con terminales de entrada en el lado izquierdo, y terminales de salida, en el lado derecho.

Cada bloque se ejecuta cuando todos los datos de entrada son recibidos en los terminales de entrada, permitiendo la ejecución paralela de más de un bloque en un mismo programa. Cada programa se denomina “Instrumento Virtual” ó VI (según sus siglas en el idioma inglés), y puede incluir uno ó más sub-Vis como una especie de subrutina.

De esta forma, se puede desarrollar una aplicación completa, basada en un lenguaje de programación altamente intuitivo, fácil de aprender, pero a la vez, tan poderoso y rápido como cualquier otro lenguaje.

A cada tipo de dato (entero, punto flotante, cadena de caracteres, etc.) se le asigna un color específico, lo que permite su rápida identificación en el código fuente.

Figura 38. Programación gráfica



Fuente: Programa de demostración de Labview

El lenguaje de programación gráfica cuenta con un compilador de alta eficiencia, que permite crear ejecutables *EXE* para la posterior distribución del modelo.

6.3 Simulación de sistemas

Un sistema es un conjunto de objetos reunidos en alguna interacción o interdependencia regular. Los sistemas están compuestos por entidades, atributos y actividades. El estado de un sistema describe la condición actual de sus entidades, atributos y actividades. Su progreso se estudia siguiendo los cambios del estado del sistema.

Un modelo es una descripción lógica de cómo un sistema, proceso o componente se comporta. La simulación incluye el diseño de un modelo de un sistema, proceso o componente, y la realización de experimentos sobre el mismo. El propósito de los experimentos es determinar cómo el sistema real se desempeña, y pronosticar el efecto de los cambios sobre el mismo en el tiempo. Se utiliza la simulación para responder tales preguntas como:

- ¿Resultará este cambio en el proceso en una mayor calidad, eficiencia ó producción?
- ¿Podemos estabilizar el sistema en un menor tiempo aplicando este algoritmo de control?
- ¿Se reduce el consumo de materia prima realizando estos cambios al sistema de control?
- ¿Se reduce la variación del proceso utilizando este nuevo algoritmo?

Existen actividades endógenas que sólo ocurren dentro del sistema. También existen actividades exógenas que sólo ocurren fuera del sistema (en el entorno). Si el sistema es un sistema “cerrado”, entonces se dice que no cuenta

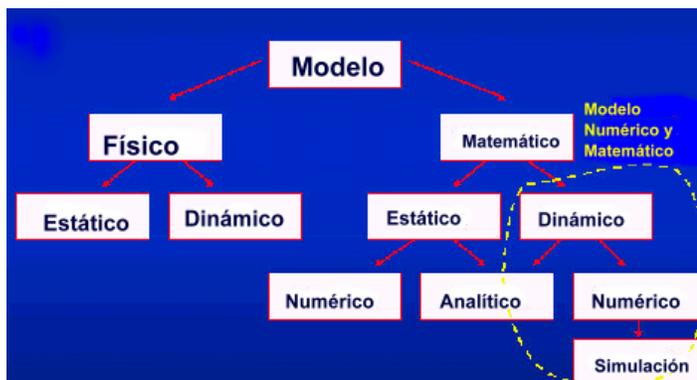
con actividad exógena. Si el sistema es “abierto”, entonces si cuenta con actividad exógena. Los cambios en actividades exógenas (del entorno) pueden afectar el sistema.

Si el resultado de una actividad se puede describir completamente en términos de su entrada, se denomina “determinística”. Si los efectos de la actividad varían aleatoriamente en diferentes salidas, se denomina “estocástica”.

Si la ocurrencia de una actividad es aleatoria, se le atribuye al entorno o medio ambiente y por lo tanto es exógena. Se utilizan distribuciones de probabilidad para describir el carácter aleatorio de una actividad.

Los modelos son estáticos o dinámicos. Los modelos dinámicos se subdividen en continuos y discretos.

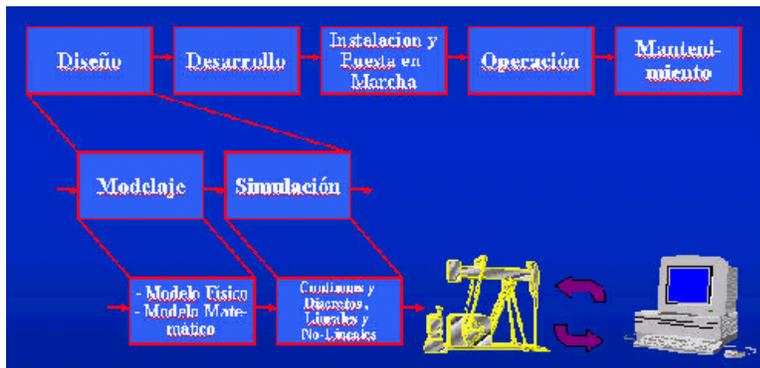
Figura 39. Simulación de sistemas



Los sistemas discretos son aquellos en los que los cambios ocurren en forma discontinua. Se describe sobre la base de los eventos que producen los cambios en el estado del sistema, y a veces se simplifican considerando que los cambios ocurren en forma continua.

Hay pocos sistemas totalmente discontinuos o totalmente discretos. Sin embargo, siempre predomina un tipo de cambio (suave o discontinuo), por lo que se pueden definir como continuos o discretos.

Figura 40. Ejemplo de un sistema en una industria



En un modelo matemático, las entidades del sistema y sus atributos se representan mediante variables matemáticas. Las actividades se describen mediante funciones matemáticas que interrelacionan las variables.

Los modelos matemáticos dinámicos normalmente se resuelven mediante métodos numéricos, y la simulación es una de las herramientas más comunes. Un modelo no es solo el sustituto de un sistema, sino también una simplificación del mismo. Para obtener un modelo se debe al menos:

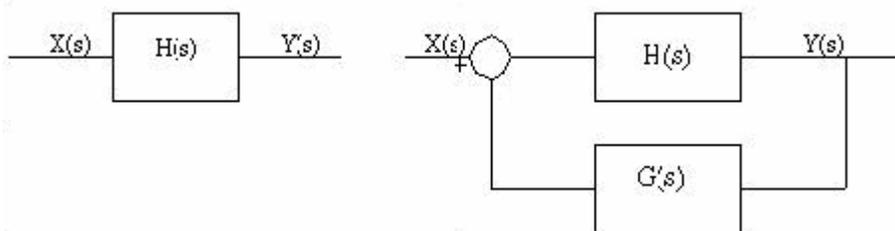
- Determinar su estructura (entidades, atributos, actividades y fronteras).
- Definir los datos (valores de los atributos y relaciones involucradas en las actividades).

- Definir interrelación: las suposiciones relativas al sistema orientan la recolección de datos y el análisis de éstos confirma o refuta las suposiciones.

Los pasos básicos para modelar un sistema de control son:

- Definir el modelo matemático y programar el sistema de control.
- Seleccionar los parámetros y los valores iniciales.
- Ejecutar el modelo (simulación).
- Procesar los resultados (incluye la visualización y la investigación adicional).
- Interactuar con el modelo hasta lograr optimización.

Figura 41. Sistema de control



Con el diseño y análisis de los sistemas de control, su modelaje y simulación, se busca satisfacer la necesidad de controlar la fabricación de los productos mediante el mantenimiento de las variables de proceso en los valores más estables posibles.

Los controladores comparan las variables de proceso con el punto de consigna *setpoint* y a partir de su diferencia o error calculan cual debe ser la señal de salida al elemento final de control (ver Figura 41). Se utiliza el controlador más apropiado: Relé (*On-Off*), *Ratio*, *PID*, *Fuzzy* y otros.

PARA AYUDAR EN EL DISEÑO DE LOS SISTEMAS DE CONTROL, EL LENGUAJE GRÁFICO, PERMITE REALIZAR LA TRANSFORMACIÓN ENTRE LAS DIFERENTES REPRESENTACIONES MATEMÁTICAS DEL SISTEMA:

- Ecuaciones de espacio-estado, a función de transferencia.
- Función de transferencia, a ecuaciones de espacio-estado.
- Cálculo de polos y ceros a partir de la función de transferencia.
- Cálculo de parámetros de interés tales como: índice de amortiguamiento, frecuencias naturales, ganancia DC, polos y ceros para la determinación de la estabilidad del sistema, etc.

La programación gráfica permite utilizar el diagrama de polos y ceros, y moverlos en forma gráfica para lograr la estabilidad del sistema, y calcular la ganancia de retroalimentación $G(s)$ para el nuevo sistema de lazo cerrado.

Una vez que se diseña el sistema de control, el lenguaje de programación gráfica le permite analizarlo en forma totalmente simulada o con señales reales capturadas o generadas en tiempo-real, a través de diagramas de *Bode*, *Nyquist*, y de ubicación de raíces para el análisis del sistema de control y su respuesta en el dominio de frecuencia.

CONCLUSIONES

1. La industria de la instrumentación está sufriendo importantes cambios como resultado de la revolución de las computadoras personales.
Estos cambios están ocurriendo tanto en el componente *hardware* como en el *software*. Un elevado número de científicos e ingenieros en el mundo usan computadoras personales para automatizar sus tareas de investigación, diseño y fabricación. En este sentido, las compañías de instrumentación *National Instruments Corporation*, *Hewlett Packcard*, *Tektronix*, entre otras, están introduciendo en el mercado nuevas herramientas que están revolucionando las interfaces de programación y el usuario. Ejemplos de estas herramientas son *VEE* de *Hewlett Packcard*, y *LabVIEW* y *LabWindows* de *National Instruments*.
2. El desarrollo de *hardware* y *software* de computadora, ha dado lugar a que aparezca un nuevo tipo de instrumentación, que es la denominada instrumentación virtual.
3. El término virtual, empleado en la industria de la instrumentación, hace referencia a la combinación de instrumentos programables con las computadoras de propósito general.
4. Un instrumento virtual se define como un nivel de *software* y *hardware* añadido a un computador de propósito general de tal modo que los usuarios pueden interactuar con el computador como si fuese un instrumento electrónico tradicional.

5. Un instrumento virtual consta de un elemento *hardware* y otro *software*. El primero está constituido por un computador personal, que realiza el procesamiento y visualización de los datos y un sistema específico, cuya función es la de adquirir los datos, generalmente está constituido por una placa de adquisición de datos, que se inserta en una ranura de expansión de la computadora. El componente *software* consiste en un programa que se ejecuta en la computadora y que controla los elementos *hardware* y procesa y visualiza los datos.

6. La instrumentación basada en *PC* es la selección correcta para aumentar la productividad y reducir el costo de las mediciones.

7. Con cada nuevo avance en la tecnología convencional de computación, la medición y la automatización basadas en computadora se vuelven más dinámicas y poderosas.

RECOMENDACIONES

1. Identificar y definir las señales de entrada y salida así como los tipos de sensores con los que se trabajará el sistema de adquisición de datos.
2. Muchos tipos de señales, provenientes de diversos sensores, deben acondicionarse antes de ser conectados a la tarjeta de adquisición de datos, por lo que, se debe escoger un método de acondicionamiento de señal de transductores y convertidores que se ajusten a los requerimientos del proceso y cuyo costo no resulte muy elevado, para lograr un acople entre los elementos que integran el lazo de control.
3. Es importante definir el cableado adecuado para la conexión entre la tarjeta y la computadora.
4. Es aconsejable realizar un proyecto piloto para probar la efectividad del instrumento virtual y comparar su desarrollo con el del instrumento tradicional.
5. Estudiar si es necesaria la implementación de un instrumento virtual, esto se consigue reconociendo que el instrumento virtual solucionará la necesidad existente produciendo beneficios palpables.

BIBLIOGRAFÍA

1. Cooper, William y Albert D. Helfrick. **Instrumentación electrónica moderna y técnicas de medición**. México: Editorial Prentice-Hall Hispanoamericana S.A.1991.
2. **CREUS SOLÉ, ANTONIO**. INSTRUMENTACIÓN INDUSTRIAL. **6ªED. COLOMBIA: EDITORIAL ALFAOMEGA. 1999.**
3. **KARCZ, ANDRÉS M.** FUNDAMENTOS DE METROLOGÍA ELÉCTRICA. **MÉXICO: EDITORIAL ALFAOMEGA. 1994.**
4. **MALONEY, TIMOTHY J.** ELECTRÓNICA INDUSTRIAL MODERNA. **3ª ED. MÉXICO: EDITORIAL PRENTICE – HALL. 1997.**
5. Méndez López, Mario Alberto. **Diseño de una red de comunicación industrial red Simatic para una planta de alimentos**. Tesis de Ing. Electrónica. Guatemala, Universidad de San Carlos de Guatemala, Facultad de Ingeniería, 2000, 79 p. Ilus.
6. **SOISSON, HAROLD E.** INSTRUMENTACIÓN INDUSTRIAL. **MÉXICO: EDITORIAL LIMUSA.1992.**
7. National Instruments. www.ni.com/latam. Agosto 2001.

