



UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO

---

FACULTAD DE INGENIERÍA

Diseño e implementación de una interfaz máquina–usuario en  
LabVIEW, para el control de un sistema de posicionamiento  
automatizado

**T E S I S**

QUE PARA OBTENER EL TÍTULO DE:

**INGENIERO EN COMPUTACIÓN**

**(ÁREA: INGENIERÍA BIOMÉDICA)**

PRESENTA:

**WILLIAM HIDBER CRUZ**

**DIRECTOR DE TESIS: DR. PEDRO ACEVEDO CONTLA**  
**CODIRECTOR: M. C. ISRAEL SÁNCHEZ DOMÍNGUEZ**



MÉXICO, D.F.

2010

## Dedicatoria

*A mis padres, hermano y amigos de la universidad. A los profesores que me apoyaron en la realización de la tesis.*

## Agradecimientos

*Proyecto PAPIIT-1N113610, por el apoyo a la realización del presente trabajo.*

Dedicatoria .....	i
Agradecimientos .....	i
Índice .....	ii
<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>Capítulo 1</b>	
<b>Sistemas de Instrumentación y control</b>	
1.1. Instrumentación .....	3
1.2. Sistemas de instrumentación .....	4
1.3. Instrumentación virtual y real .....	5
1.4. LabVIEW .....	8
1.5. Sistemas de control .....	11
1.6. Dispositivos para el control de un posicionador .....	13
1.6.1. Motores a pasos .....	13
1.6.1.1. Motor de imán permanente .....	14
1.6.1.2. Motor de reluctancia variable .....	15
1.6.1.3. Motores Híbridos .....	17
1.6.1.4. Motores unipolares y bipolares .....	17
1.7. Control de movimiento basado en PC usando LabVIEW .....	18
<b>Capítulo 2</b>	
<b>Adquisición de datos</b>	
2.1. Introducción .....	21
2.2. Conversión analógico/digital .....	22
2.3. Muestreo de señales .....	23
2.4. Tipos de convertidores .....	24
2.4.1. Aproximaciones sucesivas .....	24
2.4.2. Convertidor ADC rampa .....	26
2.4.3. Convertidor ADC Doble rampa .....	26
2.4.4. Convertidor de Tensión Frecuencia .....	27
2.5. El osciloscopio .....	28
2.6. Buses de transmisión de datos .....	33
2.6.1. GPIB .....	33
2.6.2. PXI .....	35
2.6.3. USB .....	36
2.6.4. Ethernet .....	37

## Capítulo 3

### Diseño e Implementación

3.1. Introducción .....	39
3.2. Herramientas básicas de LabVIEW .....	41
3.3. Comunicación con la tarjeta NI USB-6501 .....	42
3.4. Caracterización del Posicionador .....	43
3.4.1. Caracterización del eje X .....	47
3.4.2. Caracterización del eje Y .....	47
3.4.3. Caracterización del eje Z .....	48
3.4.4. Resultados en la caracterización del equipo .....	48
3.5. Osciloscopio .....	49
3.5.1. Programación del osciloscopio .....	50
3.5.2. Programación del Osciloscopio en LabVIEW .....	54
3.6. Diseño de la Interfaz .....	58
3.6.1. Estructura del panel principal .....	58
3.6.2. Menú archivo .....	59
3.6.3. Menú adquisición .....	61
3.6.4. Menú ventanas .....	63
3.7. Distribución del programa .....	64

## Capítulo 4

### Pruebas, Resultados y Conclusiones

4.1. Introducción .....	65
4.2. Pruebas mecánicas .....	65
4.3. Pruebas acústicas .....	71
4.4. Resultados y conclusiones .....	76

<b>Bibliografía</b> .....	79
---------------------------	----

### ANEXOS

A. Descripción del código .....	82
B. Guía rápida de manejo .....	93
C. Hojas de datos de los dispositivos .....	99
a. NI-USB 6105 .....	99
b. Osciloscopio .....	100

# Introducción

---

Dentro de las líneas de investigación del Departamento de Ingeniería de Sistemas Computacionales y Automatización (DISCA) del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS) de la UNAM, se encuentra la de imagenología ultrasónica. En los proyectos relacionados con esta línea se desarrollan nuevas técnicas de adquisición y despliegue de señales ultrasónicas, también se diseñan, modifican, construyen y caracterizan transductores ultrasónicos. Para llevar a cabo la caracterización de este tipo de transductores es necesario contar con un sistema de posicionamiento automatizado y a su vez este sistema debe de contar con una interfaz eficiente que garantice el manejo, control y funcionalidad de manera amigable de dicho sistema. El sistema en cuestión es un posicionador XYZ que consta de un conjunto de rieles controlados por motores a pasos.

Por lo tanto, el objetivo principal de la presente tesis es: desarrollar una aplicación funcional que permita la interacción entre el operador y un sistema de posicionamiento de 3 ejes, tanto para el control como para la adquisición de datos, factores básicos en la caracterización de transductores ultrasónicos.

El presente trabajo está dividido en 4 capítulos, cuyo contenido es el siguiente:

**Capítulo 1.** Se da una breve explicación sobre lo que son los sistemas de instrumentación y control, además de explicar el porqué es más fácil y económico migrar a sistemas virtuales de instrumentación y control.

**Capítulo 2.** Se comentan los conceptos básicos para entender lo que son los sistemas de adquisición de datos, además de explicar el funcionamiento de uno de los dispositivos más usados para adquirir datos, el osciloscopio. Igual se explican las diferentes interfaces de comunicación de estos sistemas con la PC.

**Capítulo 3.** Este capítulo constituye la parte central de la tesis, ya que aquí se explica paso a paso el desarrollo de ésta. Primero se explican los requisitos a cumplir por parte del software, posteriormente se da la explicación de porque se decidió programar el software sobre LabVIEW, además de los pasos iniciales antes de entrar a fondo en la programación del software (la caracterización del equipo y la comunicación entre los dispositivos). Y por último, se da una explicación a fondo sobre los paneles frontales de la versión final del software.

**Capítulo 4.** Se describen las pruebas que se le hicieron al sistema una vez finalizado el desarrollo, con el fin de comprobar que se cumplió el objetivo planteado.

**Anexos.** Finalmente se presenta una breve explicación del código del software, además de presentar las hojas de datos de los dispositivos que se ocuparon en el desarrollo de la tesis, Estos últimos necesarios para una mejor comprensión del trabajo.

## Sistemas de Instrumentación y Control

### 1.1. Instrumentación

La instrumentación trata los sistemas integrados cuya finalidad es medir magnitudes físicas de un sistema externo, elaborar la información asociada a ellas y presentarlas a un operador. En la mayoría de los circuitos eléctricos, electrónicos o de fluidos, se toman mediciones para indicar la presencia y nivel de una cantidad medible, tal como voltaje, presión, velocidad o temperatura, durante las condiciones normales de operación, algunas mediciones también se toman cuando se detectan problemas en los procesos. Las mediciones se pueden tomar con instrumentos portátiles, como son los voltímetros o los multímetros, cuando se trata de localizar fallas, pero cuando se desean tomar lecturas para indicaciones o desplegados permanentes, los instrumentos de medición se montan en tableros. Tomar y desplegar lecturas durante la operación normal da una indicación visual del comportamiento de un circuito o de un proceso, por ejemplo la observación del nivel de presión, temperatura, voltaje, corriente, velocidad, etcétera, en cualquier momento, permite tener sistemas seguros y productivos. Tomar mediciones cuando se tiene algún problema puede dar una clara idea del problema mismo en la operación del sistema y de los posibles problemas a futuro, de hecho, siempre se deben tomar mediciones cuando se observe o prevea un funcionamiento anormal. [1]

## 1.2. Sistemas de Instrumentación

Un sistema de instrumentación (figura 1.1) es una estructura compleja que agrupa un conjunto de instrumentos, como son, un dispositivo o sistema en el que se mide, las conexiones entre estos elementos y por último, y más importante, los programas que se encargan de automatizar el proceso y de garantizar la repetibilidad de las medidas. El objetivo básico de un sistema de instrumentación es la adquisición de información del mundo físico a la máxima velocidad posible, con la mayor exactitud que se pueda obtener y con el menor costo. Si esto se usa para adquirir datos, hablaremos de sistemas de instrumentación o de medición. Si esta adquisición de información se usa para determinar la respuesta a los ensayos a los que se somete un circuito integrado, un sistema electrónico o mecánico, etc., hablaremos de sistemas automáticos de prueba (Automatic test equipment: ATE). Históricamente el termino ATE se ha reservado a aquellos sistemas destinados a realizar ensayos en circuitos integrados, componentes electrónicos discretos, placas de circuito impreso o sistemas electrónicos completos, pero puede generalizarse a otros tipos de ensayo como térmicos o mecánicos. La diferencia estructural entre los sistemas de medida y de prueba radicaría en la existencia en estos últimos de un subsistema destinado a aplicar excitación al elemento que se somete a ensayo. Este subsistema puede sustituirse, a nivel formal, por la existencia de una hipótesis sobre el comportamiento del sistema físico en el que se mide. [2]

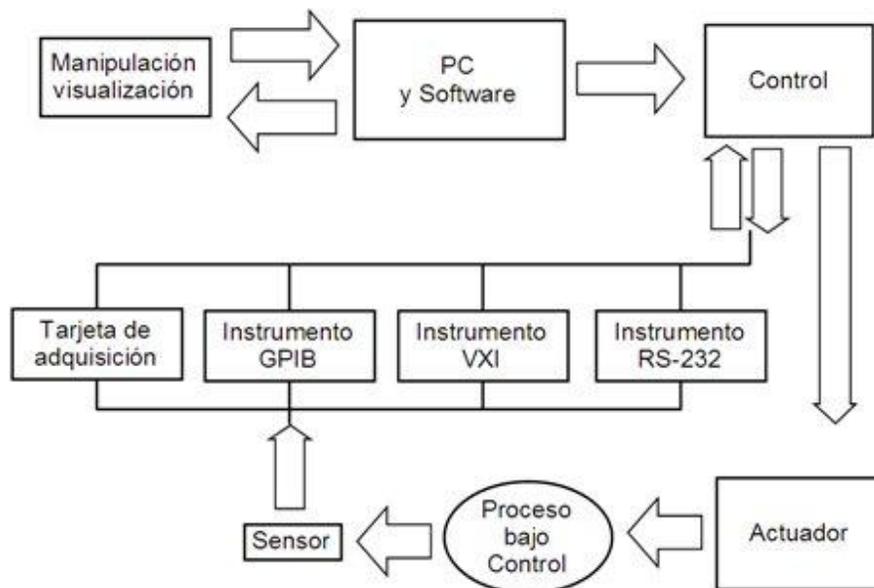


Figura 1.1. Diagrama general de un sistema de instrumentación y control.

Para poder llevar a cabo los objetivos mencionados anteriormente es necesaria la utilización de una plataforma de hardware basada en una computadora y una serie de programas que realicen el control de la circuitería. Por otra parte, la utilización de una interfaz gráfica de usuario facilitará la interacción entre el sistema y el operador. El software de control del sistema tiene que ser lo suficientemente flexible para poder adaptarse a las diferentes necesidades de medida y prueba. Además, a bajo nivel tenemos los *drivers*<sup>1</sup> de control de las diferentes opciones de adquisición. Los programas de aplicación tienen que incorporar otros *drivers* de los diferentes instrumentos de laboratorio, y de esta forma hacer más general el sistema de instrumentación. Así, por ejemplo, un *driver* de un generador de funciones mediante interfaz IEEE-488 (GPIB) tiene que ofrecer al operador una interfaz gráfica que simule el panel frontal del generador. Eso significa que desde la pantalla del ordenador tiene que poder controlarse el generador de una manera similar a como se haría manualmente. Este *driver* tiene que permitirnos, aprovechando todos los comandos GPIB disponibles, desarrollar prácticamente todas las funciones del instrumento en la pantalla de nuestro ordenador. Si a estas funciones somos capaces de añadir otras procedentes de las librerías propias del paquete de software, tenemos como resultado un instrumento virtual. [3]

### 1.3. Instrumentación Virtual y Real

Antes de hablar de instrumentos reales y virtuales, lo que nos interesa, primero es necesario saber que es un instrumento; un instrumento es un objeto, herramienta, aparato, etc., que sirve para hacer una operación, producir un efecto, etc., es decir, es una herramienta que nos ayuda a comprender mejor los eventos físicos que ocurren a nuestro alrededor. Un instrumento virtual consiste de una computadora del tipo industrial, o una estación de trabajo equipada con poderosos programas (software), hardware económico, tales como tarjetas para insertar, y controladores (*drivers*) que cumplen en conjunto, las funciones de instrumentos tradicionales. Los instrumentos virtuales representan un apartado fundamental en los sistemas de instrumentación que están basados completamente en hardware, ya que aprovechan la potencia de cálculo, productividad, exhibición y capacidad de conexión de las populares computadoras de escritorio y estaciones de trabajo para sustituir parte de este hardware por un software que haga la misma tarea. Aunque la PC y la tecnología de circuitos integrados han experimentado avances significativos en las últimas dos décadas, es el software el que realmente provee la ventaja para construir sobre una potente base de hardware para crear instrumentos virtuales, proveyendo mejores maneras de innovar y de reducir costos significativamente. Con los instrumentos virtuales, los ingenieros construyen sistemas de medición y automatización que se ajustan exactamente a

---

<sup>1</sup> Llamado normalmente controlador es un programa informático que permite al sistema operativo interactuar con un periférico, haciendo una abstracción del hardware y proporcionando una interfaz para usarlo.

sus necesidades, en lugar de estar limitados por instrumentos tradicionales de funciones fijas.

Los instrumentos tradicionales, tales como osciloscopios y generadores de ondas, son muy poderosos, caros y diseñados para llevar a cabo una o más tareas específicas definidos por el fabricante. Sin embargo, el usuario por lo general no puede extender o personalizar esas tareas. Las perillas y botones del instrumento, sus circuitos electrónicos y las funciones disponibles para el usuario son todas específicas a la naturaleza del instrumento. Además, debe desarrollarse tecnología especial y costosos componentes para construirlos, los cuales los hace muy caros y lentos para adaptarlos. Los ingenieros y científicos cuyas necesidades, aplicaciones y requerimientos varían muy rápidamente, necesitan flexibilidad para crear sus propias soluciones, por lo que muchas veces optan por los instrumentos virtuales. Los instrumentos virtuales aprovechan los beneficios de la última tecnología de las computadoras, haciéndolos igual de robustos que un instrumento tradicional. Uno puede adaptar un instrumento virtual a las necesidades particulares sin necesidad de reemplazar todo el instrumento dado que posee el software de aplicación instalado en la computadora y al amplio rango de hardware para instalar en ella. El instrumento virtual muchas veces sólo está limitado por el software que se utiliza. Las diferencias entre instrumentos tradicionales y virtuales se pueden resumir en la tabla 1.1.

**Tabla 1.1.** Comparación entre un instrumento tradicional y un instrumento virtual.

<b>Instrumentos tradicionales</b>	<b>Instrumentos virtuales</b>
Definidos por el fabricante	Definidos por el usuario.
Función específica, conectividad limitada	Sistema orientado a la aplicación con conectividad a redes, periféricos y aplicaciones.
El hardware es la clave	El software es la clave.
Caro	Bajo costo, reutilizable.
Cerrado, Funcionalidad fija	Abierto, Funcionalidad flexible.
Cambios lentos en la tecnología (5-10 años de ciclo de vida)	Adaptación rápida a los cambios tecnológicos.
Costos de desarrollo y mantenimiento grandes	El software minimiza los costos de desarrollo y mantenimiento.

El software sobre el que se programa el instrumento virtual y que se ejecuta sobre el controlador, en este caso la computadora, ha evolucionado con el tiempo. Los primeros entornos de programación únicamente permitían el control de instrumentos o dispositivos externos a la

computadora. Algunos fabricantes de interfaces para computadora suministraban, primero, un conjunto de funciones que se dejaban residentes en memoria y a las que se accedía mediante interrupciones de software. Posteriormente suministraban librerías de funciones que se podían llamar desde lenguajes de alto nivel, que ejecutaban las interrupciones software. Estas primeras herramientas facilitaban el control de la interfaz de comunicaciones con instrumentos externos y eran independientes del instrumento a controlar. No obstante no incluían utilidades para el análisis de datos ni la presentación de los mismos. De aquí podemos ver que la mayoría de los lenguajes de programación que se utilizaban y en algunos casos se siguen utilizando son los lenguajes basados en texto. Sin embargo, existen otros grupos de lenguajes de programación, los lenguajes gráficos. Su aparición en el mercado es más reciente. El desarrollo de aplicaciones es totalmente diferente, permite crear al usuario soluciones completas uniendo iconos de una forma totalmente gráfica y según una estructura jerárquica. Al igual que los lenguajes de programación clásicos se dispone de múltiples tipos de datos y estructuras de programación (bucles, condiciones, E/S, etc.) incluyendo algunos entornos un compilador para aumentar la velocidad de ejecución. El ciclo de aprendizaje y programación se reduce drásticamente al ser entornos que imitan una forma de programación muy parecida a un diagrama de flujo o bloques. Esta simplificación en la forma de programación lleva asociada algunas limitaciones. La velocidad final de la aplicación será mucho menor que su versión en lenguaje C y la utilización de muchos elementos gráficos e iconos requiere grandes cantidades de memoria y potencia de cálculo.

Existen en el mercado diversos paquetes de programación gráfica, tales como:

El visual DAS de Keithley, el cual es un API<sup>2</sup> de VISUAL BASIC.

VEE (Hewlett-Packard): es un entorno de programación gráfico propio que funciona sobre diferentes plataformas, Macintosh, Windows y estaciones de trabajo HP y SUN, dispone de librerías para la manipulación de datos, control de flujo, interfaces de usuario y *drivers* de instrumentos compatibles de HP, no dispone de librerías para el control de tarjetas de adquisición. El editor gráfico no incluye compilador y existe una versión run-time para la aplicación final.

DT VEE (Data translation): Es un entorno propio, disponible para Windows. Está orientado a la programación de sus tarjetas de adquisición y presentación de datos. No dispone de *drivers* para instrumentos controlables.

El software más utilizado en la instrumentación virtual y que igual está clasificado en los entornos de programación gráficos es LabVIEW.

---

<sup>2</sup> Una interfaz de programación de aplicaciones o API (Application Programming Interface) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

## 1.4. LabVIEW

LabVIEW (figura 1.2) es un lenguaje de programación gráfico diseñado para ingenieros y científicos para desarrollar aplicaciones de pruebas, control y medidas. La naturaleza intuitiva de la programación gráfica de LabVIEW lo hace fácil de usar. LabVIEW es el acrónimo de *Laboratory Virtual Instrument Engineering Workbench*. National Instruments es la empresa desarrolladora y propietaria de LabVIEW, comenzó en 1976 en Austin Texas y sus primeros productos eran dispositivos para el bus de instrumentación GPIB. En abril de 1983 comenzó el desarrollo de lo que sería su producto estrella: LabVIEW, que vería la luz en octubre de 1986 con el lanzamiento de LabVIEW 1.0. Originalmente este programa estaba orientado a aplicaciones de control de instrumentos electrónicos usados en el desarrollo de sistemas de instrumentación, lo que se conoce como instrumentación virtual. Por este motivo los programas creados en LabVIEW se guardan en ficheros llamados VIs (*Virtual Instruments*) y con la misma extensión.

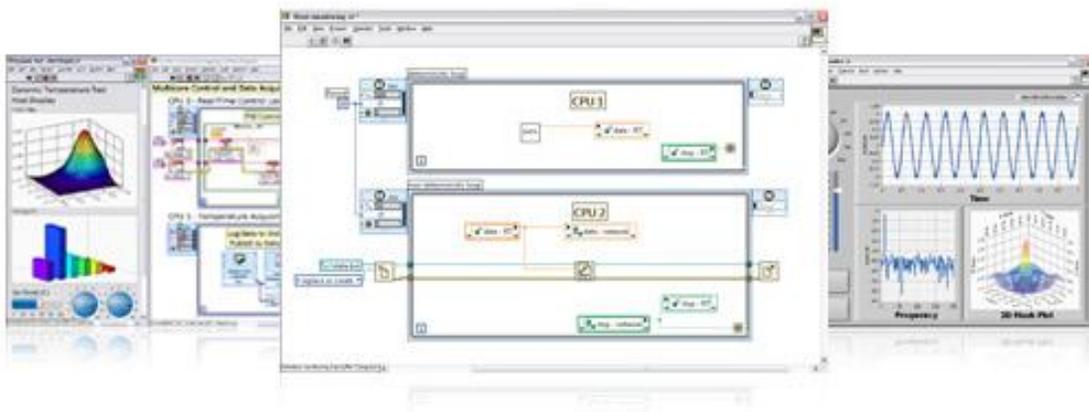


Figura 1.2. Entorno de programación LabVIEW.

Los VIs imitan a los instrumentos físicos tanto en su apariencia como en su operación. Entre las ventajas que encontramos al utilizar LabVIEW tenemos:

La primera ventaja de usar LabVIEW es que es compatible con herramientas de desarrollo similares y puede trabajar a la vez con programas de otra área de aplicación, como Matlab o Excel. Además se puede utilizar en muchos sistemas operativos, incluyendo Windows y UNIX, siendo el código transportable de uno a otro.

Otra de las ventajas más importantes que tiene este lenguaje de programación es que permite una fácil integración con hardware, específicamente con tarjetas de medición, adquisición y procesamiento de datos (incluyendo adquisición de imágenes).

Es muy simple de manejar, debido a que está basado en un nuevo sistema de programación gráfica, llamado lenguaje G.

Es un programa enfocado hacia la instrumentación virtual, por lo que cuenta con numerosas herramientas de presentación, en gráficas, botones, indicadores y controles, los cuales son muy esquemáticos y versátiles. Estos serían complicados de realizar en bases como C++, donde el tiempo de programación para lograr el mismo efecto sería muchas veces mayor.

Es un programa que contiene librerías especializadas para manejos de tarjetas de adquisición de datos (DAQ: Data Acquisition), redes, comunicaciones, análisis estadístico, comunicación con bases de datos (útil para una automatización de una empresa a nivel total).

Como se programa creando subrutinas en módulos de bloques, se pueden usar otros bloques creados anteriormente como aplicaciones por otras personas.

Los VIs de LabVIEW contienen tres componentes principales: el panel frontal, el diagrama de bloques, y el panel de icono/conector (figuras 1.3 a y 1.3 b).

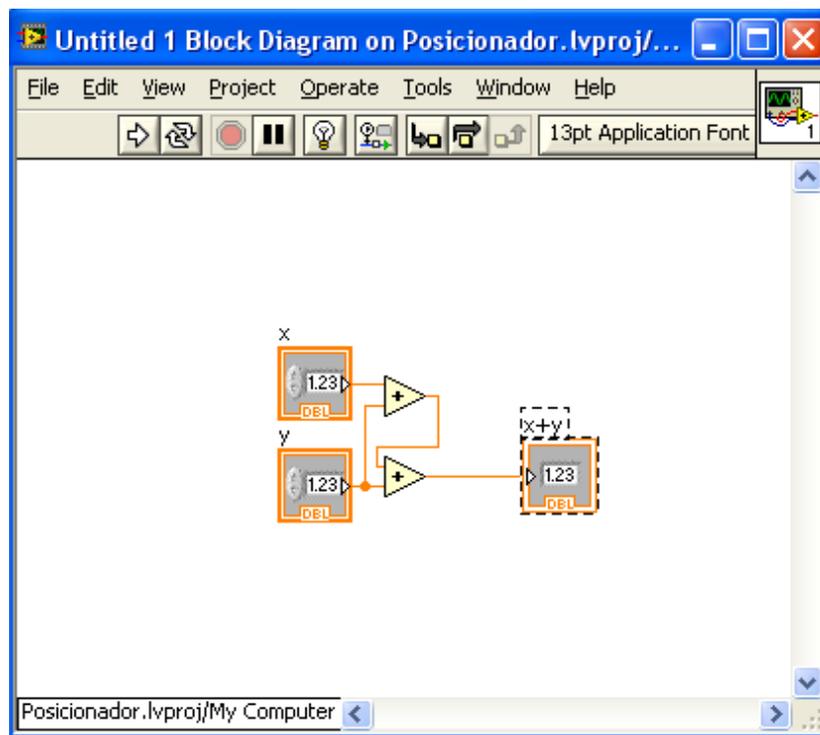


Figura 1.3. a) Componentes de LabVIEW, diagrama de bloques (Donde se encuentra el código del VI).



Figura 1.3. b) Componentes de LabVIEW, Panel frontal.

El panel frontal (figura 1.3a) es la interfaz que observa el usuario, en este panel es donde se colocan los controles, indicadores, gráficos, etcétera, todos interactivos con el usuario. Los controles los constituyen perillas, conmutadores, potenciómetros deslizantes y otros dispositivos de entrada, que suministran los datos de entrada al VI. Los indicadores son gráficos, leds y otros tipos de displays que simulan los dispositivos de salida y visualizan los datos adquiridos o generados por el VI.

El diagrama de bloques (figura 1.3b) es donde se introduce el código, donde se realiza la programación, permite al usuario crear soluciones completas uniendo iconos de una forma totalmente gráfica y según una estructura jerárquica, los bloques son módulos software pre programados que aparecen como iconos en la pantalla, algunos iconos son estándares para cualquier aplicación, pero otros corresponden a un hardware específico del sistema de medida o creados por el programador. Dentro de este panel igual se encuentran los iconos y sus respectivos conectores (se le conoce como panel icono conector).

Una vez construidos el panel frontal y el diagrama de bloques, se puede diseñar el icono y conector, ya que cada VI se puede usar como un subVI. Cada VI muestra su icono en la parte superior derecha de las ventanas de panel frontal y diagrama de bloques. El icono es una representación gráfica del VI, puede contener texto, imágenes o una combinación de ambas. Al utilizar el VI como un subVI el icono identifica a éste en el diagrama de bloques del VI principal. Al hacer doble *click* sobre él se puede editar para su personalización. También es necesario diseñar una rejilla de conexión para la utilización del VI como subVI. La rejilla de conexión es un conjunto

de terminales que corresponden a los controles e indicadores del VI. Esto es similar a la lista de parámetros en una llamada a una función en los lenguajes de programación basados en texto. La rejilla de conexión define las entradas y salidas que se pueden conectar al VI, la rejilla de conexión recibe los datos en los terminales de entrada y los pasa al código del diagrama de bloques a través de los controles del panel frontal y recibe los resultados en sus salidas a través de los indicadores del panel frontal. Un subVI es análogo a un método en los lenguajes de programación basados en texto.

### 1.5. Sistemas de Control

Desde el punto de vista de la teoría del control, un sistema o proceso está formado por un conjunto de elementos relacionados entre sí para alcanzar una determinada finalidad, este sistema ofrece señales de salida en función de señales o datos de entrada. El conocimiento del comportamiento de la señal de salida respecto a una variación en la señal de entrada permitirá seleccionar la acción de control adecuada para lograr mejoras en la respuesta prevista del sistema. En todo sistema de control, el objetivo principal es mantener la variable a controlar dentro de ciertos límites permisibles por el sistema, en el control de movimiento la variable típica a controlar es la posición. El funcionamiento de un sistema depende de las características e interacciones entre sus elementos. Se denomina sistema de control (figura 1.4), aquel sistema constituido por un conjunto de elementos conectados de forma que permiten regular o gobernar a otro sistema, denominado planta, sistema o proceso. El aspecto más importante de un sistema es el conocimiento de su dinámica de funcionamiento, un conocimiento preciso de la relación entrada/salida permite predecir la dinámica del sistema y seleccionar la acción de control adecuada para mejorarla. De esta manera, el diseñador, conociendo cual es la dinámica deseada, ajustará la acción de control para conseguir el objetivo final. [4]



Figura 1.4. Sistema general de control.

**Sistemas de control en lazo abierto:** Son denominados así a aquellos sistemas en los cuales la salida no tiene efecto sobre la acción de control (figura 1.5). Es decir, en un sistema de control en lazo abierto no se mide la salida ni se realimenta para compararla con la entrada. Un ejemplo práctico es una lavadora. El remojo, el lavado y el centrifugado en la lavadora operan con una base de tiempo. La máquina no mide la señal de salida, que es la limpieza de la ropa. [4]

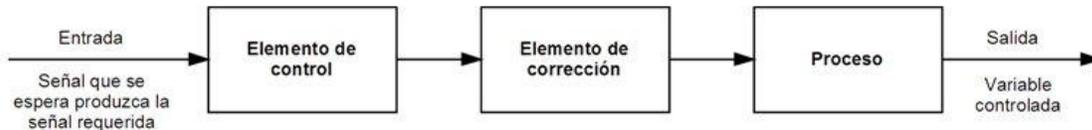


Figura 1.5. Sistema general de control en lazo abierto.

Debido a que en un sistema de control de lazo abierto la salida no se compara con la entrada de referencia, a cada entrada de referencia le corresponde una condición operativa fija; como resultado, la precisión del sistema depende de la calibración. Ante la presencia de perturbaciones, un sistema de control en lazo abierto no realiza la tarea deseada. En la práctica, el control en lazo abierto sólo se utiliza si se conoce la relación entre la entrada y la salida y si no hay perturbaciones internas ni externas. En el caso de un sistema de lazo abierto para el control de motores a pasos hay que tomar en cuenta que la realimentación no existe si no hay algún dispositivo que contemple la posición actual, la cual nos permita controlar la generación de pulsos para enviárselas al motor y así tener un mayor control en la posición. En la figura 1.6 se muestra el diagrama para el control de un motor a pasos en lazo abierto. El controlador del motor recibe la señal de dirección y genera la secuencia lógica para desplazarlo, la cual ingresa a una etapa de potencia, la cual genera la corriente de excitación. Cuando los pulsos son aplicados al controlador del motor, los estados de las terminales de salida cambian para girar el motor un paso en la dirección enviada. En este caso la precisión en el control depende del ángulo de paso del motor, debido a que los pulsos son generados en ciertas aplicaciones por un microprocesador, éste puede controlarlos de manera eficiente y confiable, por lo que la velocidad del motor está controlada. Pero el funcionamiento bajo este tipo de modo presenta ciertas desventajas. Por ejemplo, si la frecuencia del tren de pulsos es demasiado alta, el motor puede que no reaccione al tren de pulsos, y si hay un error de posicionamiento éste tiende a ser acumulativo.

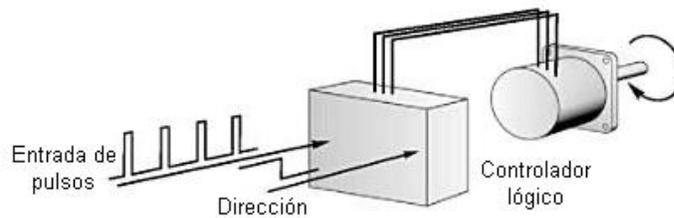


Figura 1.6. Diagrama de un sistema de control en lazo abierto para el control de un motor a pasos.

**Sistemas de control en lazo cerrado:** Los sistemas de control realimentados se denominan también sistemas de control en lazo cerrado. En la práctica, los términos control realimentado y control en lazo cerrado se usan indistintamente. En un sistema de control en lazo cerrado, se alimenta al controlador la señal de error de actuación, que es la diferencia entre la señal de entrada y la señal de realimentación (que puede ser la propia señal de salida o una función de la señal de salida y sus derivada y/o integrales), con el fin de reducir el error y llevar la salida del sistema a un valor deseado. El término control en lazo cerrado siempre implica el uso de una acción de control realimentado para reducir el error del sistema. [4]

## 1.6. Dispositivos para el control de un posicionador

Para el control de las variables cinemáticas, son los motores los que se utilizan para operar un sinnúmero de aplicaciones, y el motor de pasos en específico ha sido el móvil básico de una gran cantidad de sistemas de control de movimiento de precisión.

### 1.6.1. Motores a pasos

Un motor a pasos, o motor de velocidad gradual, es un dispositivo electromecánico que convierte pulsos eléctricos en movimientos mecánicos discretos. Éstos se producen en la inversión de la corriente por los devanados, que a su vez es controlada por unos interruptores de estado sólido de potencia. Dependiendo de su diseño, un motor a pasos puede avanzar  $90^\circ$ ,  $45^\circ$ ,  $18^\circ$  o incluso una fracción de grado por pulso. Estos son motores que se usan especialmente en aplicaciones donde se tiene que controlar la posición y el movimiento con precisión. Variando la velocidad de los pulsos, podemos hacer que el motor avance rápida o lentamente.

### Ventajas

- El ángulo de rotación del motor es proporcional a los pulsos de entrada
- El motor mantiene su torque mientras esté energizado.
- Precisión en su posicionamiento.
- Excelente respuesta al inicio, parada y reversa.

Algunas aplicaciones de estos motores son: máquinas herramientas, trazadores X-Y, caseteras, válvulas e impresoras. Existen principalmente 3 tipos de motores a paso:

- De imán permanente
- Reluctancia variable
- Híbridos

#### 1.6.1.1. Motores de imán permanente

Estos motores tienen la característica de que utilizan un imán permanente cerámico cilíndrico en el rotor, es decir estos imanes están adheridos al rotor, los polos N y S son permanentes (figura 1.7). El imán está magnetizado radialmente en una serie de polos. El estator está constituido por láminas de material ferro magnético, bobinado con el mismo número de polos que el rotor.

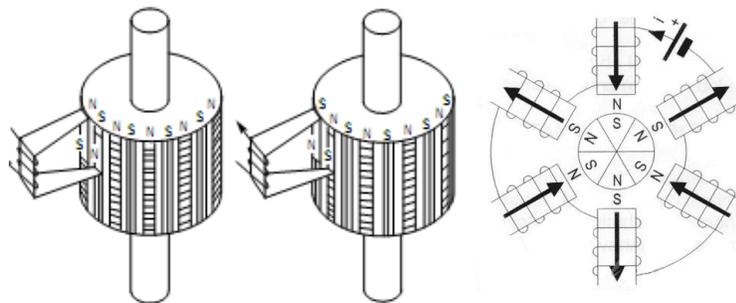


Figura 1.7. Corte transversal de un motor a pasos de imán permanente.

En la figura 1.8 se muestra el corte transversal de un motor a pasos de dos polos y dos fases (se llama así porque el estator tiene dos bobinas y el rotor dos polos magnéticos), supongamos que el devanado *bs* está a circuito abierto y que se aplica una corriente positiva constante por el devanado *as*. El resultado es que esta corriente establece un polo sur de estator, en el diente de

estator donde está el devanado  $as_1$ , y se establece el polo norte del estator en el diente en el que está devanado  $as_2$ . El rotor se colocará en  $\theta_{rm} = 0$ . Ahora se desenergiza el devanado  $as$  al mismo tiempo que se energiza el devanado  $bs$ , con una corriente positiva. El rotor se mueve la longitud de un paso en dirección contraria a la de las manecillas del reloj. Para continuar los pasos en esa dirección, se desenergiza el devanado  $bs$  y se energiza el  $as$  con una corriente negativa. Esto es, se hacen pasos en sentido contrario al de las manecillas del reloj con la secuencia de corriente  $i_{as'} i_{bs} -i_{as} -i_{bs} i_{as'} i_{bs} \dots$ . La rotación en sentido de las manecillas del reloj se logra con una secuencia de corriente  $i_{as'} -i_{bs} -i_{as} i_{bs} i_{as'} -i_{bs} \dots$  [5]

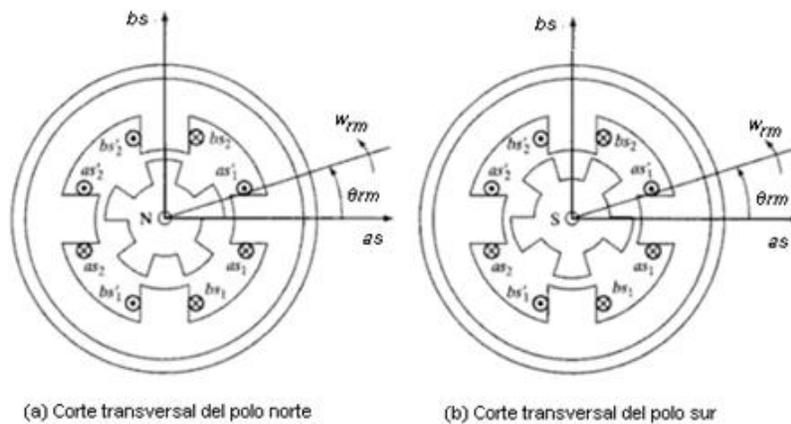


Figura 1.8. Corte transversal de un motor a pasos.

El ángulo de paso de este tipo de motores depende del número de polos del estator y el rotor. Debido a las características del material magnético utilizado en la construcción del rotor, el número de polos de éste es limitado, por lo que los ángulos que se consiguen con este tipo de motor son grandes. Por lo tanto el número de pasos viene dado por:

$$n = n_e * n_p \quad (1)$$

Donde:

$n_e$  = número de devanados (fases)

$n_p$  = número de polos

### 1.6.1.2. Motores de Reluctancia Variable

En los motores de reluctancia variable su rotor es de hierro dulce laminado, con varios dientes en dirección radial, mientras que el estator tiene un número de polos diferente también en

dirección radial (figura 1.9). Si el número de fases del estator es  $q$  y su número de polos es  $N_s$ , entonces el número de dientes del rotor  $N_r$ , se suele elegir como  $N_r = N_s \pm (N_s/q)$ . El número de dientes del rotor es menor que el número de dientes del estator, de modo que sólo un par de polos del estator y su correspondiente par de polos del rotor pueden estar alineados por fase. [6]

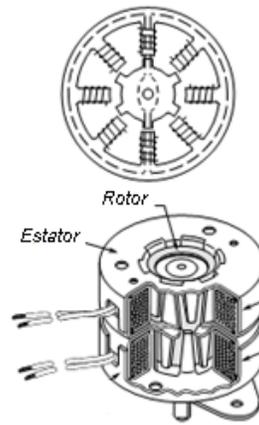


Figura 1.9. Motor a pasos de reluctancia variable.

La ranuración del rotor conlleva a una variación de la reluctancia en función de su posición angular. Al aplicar un impulso  $i_a$  a la fase del estator, el rotor (hierro) se acerca al electroimán para disminuir el entrehierro, y con ello la reluctancia magnética en el circuito, y así facilitar el paso del flujo magnético. Partiendo de la posición de equilibrio con la fase 1 activada (figura 1.10 a), un paso se obtiene situando la fase 1 a estado *OFF* y la fase 2 a estado *ON*; en este instante, los polos del rotor más próximos a los polos de la fase dos del estator, son atraídos en el intento de circular las líneas de flujo magnético (figura 1.10 (b)), produciendo un movimiento de rotación entre ambos polos hasta quedar alineados (figura 1.10 (c)). Con esta operación obtenemos un paso del rotor, seguimos haciendo lo mismo sucesivamente para hacer el desplazamiento de más pasos.

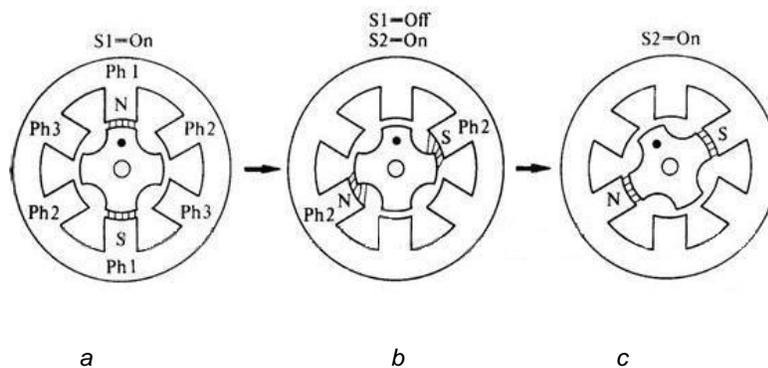


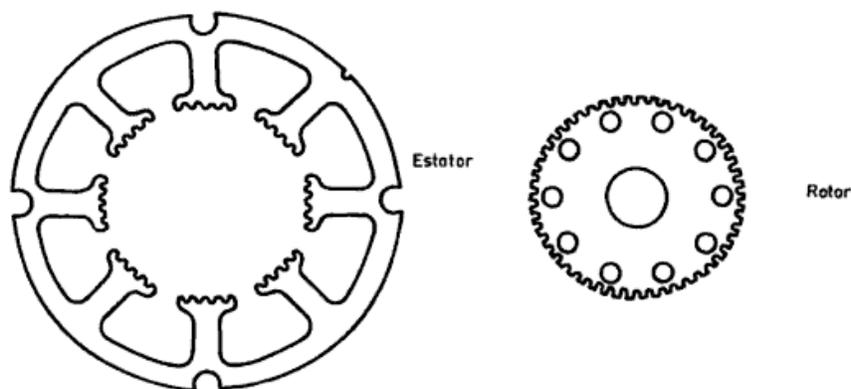
Figura 1.10. Secuencia de excitación para producir un paso.

El paso angular es:

$$\frac{360}{qN_r} \quad (2)$$

### 1.6.1.3. Motores Híbridos

Los motores híbridos son una combinación de los dos tipos mencionados anteriormente. Se denominan también motores de inductancia síncrona. Su estator consta de varias bobinas, mientras que su rotor consiste en un imán cilíndrico magnetizado axialmente (en la dirección del eje del motor), dispuesto entre dos piezas de hierro dulce laminado que tiene varios dientes, en número ligeramente distinto al de las bobinas del estator (figura 1.11). [6]



1.11 Motor paso a paso híbrido.

### 1.6.1.4. Motores unipolares y bipolares

Para los motores unipolares el bobinado por cada fase es doble (figura 1.12), unido en el interior y puesto en serie nos entrega 6 hilos, agrupados de tres en tres para cada fase (uno de estos es el punto común). Existen tres tipos de secuencia para manejar estos tipos de motores (figura 1.12), tipo *wave drive*, se activa una sola bobina a la vez, secuencia normal, se activan dos bobinas a la vez y secuencia de medio pasó.

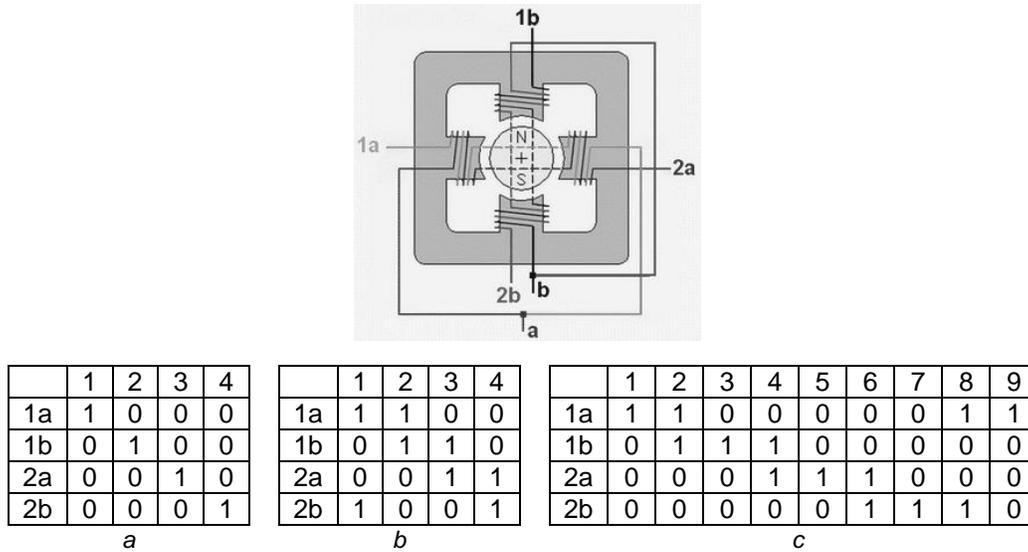


Figura 1.12. Secuencia de pulsos para un motor a pasos unipolar, a) Tipo wave drive, b) Secuencia normal, c) Secuencia de medio paso.

La configuración de los motores bipolares (figura 1.13), requiere que las bobinas reciban corriente en uno y otro sentido, y no solamente un encendido-apagado como en los unipolares. Cada inversión de la polaridad provoca el movimiento del eje en un paso.

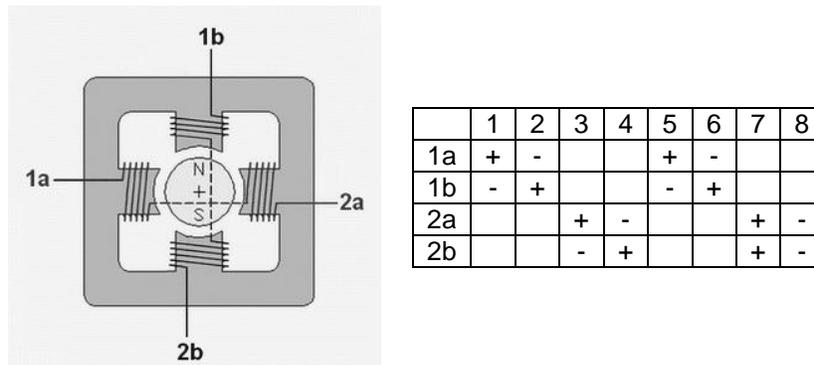


Figura 1.13. Secuencia de manejo para un motor a pasos bipolar.

### 1.7. Control de movimiento basado en PC usando LabVIEW

Durante años la industria confió en sistemas basados en PLCs y control de movimientos propietarios. Sin embargo, un creciente número de ingenieros están hoy optando a las soluciones de control de movimientos basados en PC para automatizar procesos fabriles y ensayar maquinaria. Estos sistemas son más fáciles de programar y ofrecen mayor flexibilidad y mayor

rendimiento que los sistemas basados, por ejemplo, en PLCs. Las aplicaciones pueden personalizarse para atender requerimientos actuales y futuros, eliminando así la necesidad de volver a invertir en un nuevo sistema. Este cambio a las PCs es debido principalmente a que el rendimiento de las computadoras mejora regularmente y así los científicos e ingenieros pueden integrar fácilmente tecnología de control de movimientos basado en PC con otros sistemas de medición, tales como visión y adquisición de datos. De manera entonces que la nueva tecnología ahorra dinero y provee una flexibilidad adicional.

Un sistema de control de movimiento (figura 1.14), consiste de cinco principales componentes, el dispositivo mecánico que se está moviendo, el motor (servo o por pasos) con retroalimentación y E/S de movimiento, el amplificador, el controlador inteligente y el software de interfaz de programación/operación. Mientras las soluciones que provienen de otros vendedores emplean sistemas de arquitectura cerrada, la flexibilidad agregada y potencial para costos más bajos que está disponible con soluciones para movimiento basado en computador están haciendo que aumente su popularidad. Los productos de movimiento para varios ejes de National Instruments suministran exactitud y movimiento de alto desempeño para todas las aplicaciones con motores servo o por pasos.

Una parte importante de un sistema de control de movimiento basado en PC es el software que integra todos los componentes. Una ventaja del sistema de control de movimiento basado en PC sobre los sistemas propietarios cerrados es que los controladores de los primeros a menudo poseen funcionalidad de control de movimiento incorporada, permitiendo al usuario tener más tiempo para invertir en el desarrollo del software de comando y monitoreo y proveyendo un controlador de *display* basado en PC. Un software adaptable en controladores de movimiento significa que la aplicación es compatible con diferentes tipos de motores, incluyendo los diferentes motores paso a paso y servomotores. A medida que se generan nuevas versiones del software, no hay necesidad de invertir en nuevo hardware, en lugar de eso simplemente se puede actualizar a la versión de software más reciente. Esta compatibilidad hacia atrás preserva la inversión en el software, ahorrando tiempo y dinero. Este software también debería ser compatible con importantes entornos de software de aplicación. La facilidad de uso es un factor importante cuando se instala y configura un sistema de control de movimientos basado en PC. Por ejemplo, algunos software permiten verificar el número de ranura y probar los recursos asignados cuando se instala el controlador de movimiento. Luego, simplemente se ejecutan las rutinas de inicialización para inicializar y verificar el sistema. Con esta metodología se puede operar y verificar el sistema de control de movimiento antes de hacer cualquier programación. Otra consideración a tener en cuenta al elegir el software de control de movimientos es su habilidad para verificar cada componente del sistema sin crear programas personalizados. Utilizando un medio ambiente configurable, se puede fácilmente ensayar y configurar el sistema sin programación alguna. Algunos de los parámetros de configuración y ensayo incluyen: configuración de conmutación



# Adquisición de Datos

## 2.1. Introducción

**A**l hablar de *adquisición de datos* nos referimos a la recolección de información de sistemas que trabajan en tiempo real y se caracterizan por su habilidad para adquirir datos o para realizar una tarea de control dentro de una ventana de tiempo aceptable. La naturaleza se mueve en forma de ondas analógicas, ya sean las olas del océano, un terremoto, un estampido sónico, una explosión, el sonido a través del aire, o la frecuencia natural de un cuerpo en movimiento. La energía, las partículas vibratorias y otras fuerzas invisibles, impregnan nuestro universo físico. Incluso la luz (en parte partícula y en parte onda) tiene una frecuencia fundamental que se puede observar como un color determinado. Todos estos procesos que se presentan en el mundo real producen señales analógicas cuya variación es continua, en la figura 2.1 se muestran algunas señales captadas de aparatos de uso cotidiano. La velocidad de esta variación a veces es muy lenta, como es el caso de las variaciones de la temperatura, pero también pueden ser muy rápidas, como en un sistema de audio, la mejor manera de describir los procesos analógicos es mediante números decimales y letras del alfabeto.

No es fácil guardar, manejar, comparar, calcular o recuperar datos de manera precisa utilizando tecnología analógica, por lo que muchas veces nos apoyamos de una computadora. Sin embargo,

las computadoras trabajan con señales digitales, por lo tanto es necesario convertir cualquier señal analógica que ocupemos a digital, es decir convertir estas señales analógicas a intervalos discretos. Por este motivo surgió la necesidad de contar con convertidores que sirvan para vincular entre sí a los mundos analógico y digital. Mediante los convertidores analógicos a digital (ADC: Analog Digital Converter) el mundo analógico se comunica con las computadoras. El hardware utilizado para la adquisición de datos lo llamamos “Sistema de adquisición de datos”, estos sistemas funcionan como una interfaz entre el mundo real con sus parámetros físicos reales, que son analógicos y el mundo de las señales digitales de las computadoras.

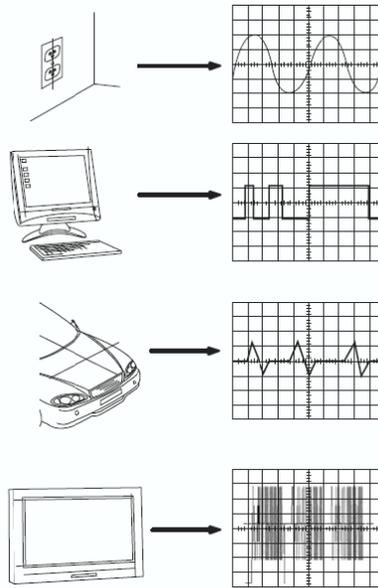


Figura 2.1. Formas de onda habituales.

## 2.2. Conversión Analógico/Digital

El muestreo de señales eléctricas, usualmente voltajes, se efectúa de manera más común con dos dispositivos, el de muestreo y retención (M/R) y el convertidor analógico digital. A veces estos dispositivos se acoplan en conjunto en un módulo electrónico. La excitación del M/R es el voltaje analógico en su entrada, y cuando se le agrega un reloj, reproduce ese voltaje a la salida como respuesta y lo retiene hasta que se vuelve a activar el reloj para adquirir otro voltaje. La adquisición de la señal del voltaje de entrada del M/R ocurre durante el tiempo de apertura, que es el ancho del pulso del reloj. Durante el pulso de reloj la señal de voltaje de salida se mueve con mucha rapidez desde su valor anterior para seguir la excitación. Al final del pulso del reloj la señal del voltaje de salida se mantiene en un valor fijo hasta que ocurre el siguiente pulso del reloj. [7]

Los convertidores A/D son dispositivos electrónicos que establecen una relación entre el valor de la señal en su entrada y la palabra digital obtenida en su salida. La relación se establece, en la

mayoría de los casos, con la ayuda de una tensión de referencia. La conversión de señales analógicas a digitales se realiza en dos etapas: primero se cuantifica la señal y luego se codifica.

La cuantificación consiste en representar la amplitud (continua) de la señal mediante un número finito de valores distintos, en instantes determinados; si el conversor es de  $n$  bits, hay  $2^n$  valores o estados posibles. La codificación es la representación del valor asignado a la señal, mediante combinaciones de símbolos que se representan habitualmente con dos niveles de tensión lógicos (1,0). La cuantificación y codificación las realiza el convertidor analógico-digital (ADC). [6]

En la figura 2.2 se muestra una señal senoidal y su versión digital, utilizando para ello un ADC de 3 bits. En este caso hay 8 estados de salida, representados por los códigos desde 000 hasta 111. Si el margen  $M$  de tensiones de entrada del ADC es de 0 a 10 V, cada código representa un intervalo de amplitudes de  $10 \text{ V}/8 = 1.25 \text{ V}$ . [8, 9, 10]

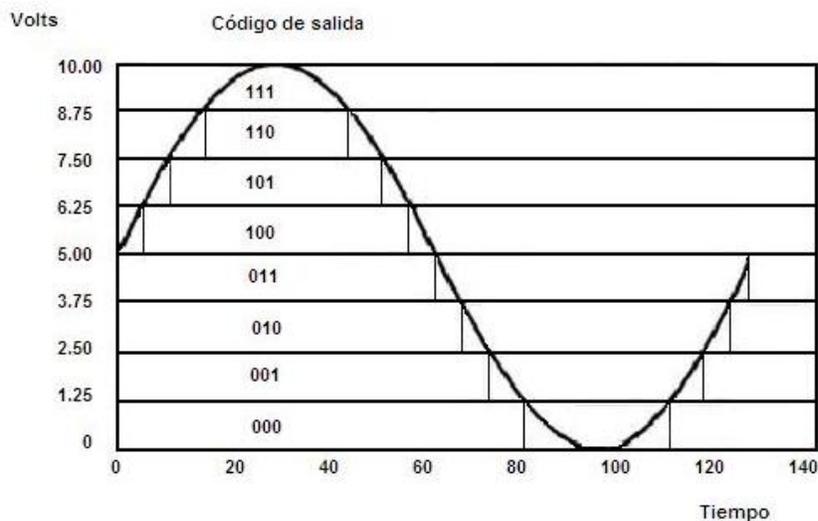


Figura 2.2. Señal senoidal y su versión digital con una resolución de 3 bits.

### 2.3. Muestreo de señales

El muestreo es un proceso lineal mediante el cual se transforma una señal continua en el tiempo y de banda limitada, en una sucesión (continua) de valores de amplitud en instantes discretos, que constituyen la denominada señal muestreada o serie temporal. Esta transformación se puede hacer de distintas formas, pero lo más habitual es que la señal muestreada esté formada por los valores de la señal original en instantes de tiempo equiespaciados. Consiste en tomar muestras periódicas de la amplitud de una señal analógica, siendo el intervalo entre las muestras constante. El ritmo de este muestreo se denomina frecuencia o tasa de muestreo y determina el número de muestras que se toman en un intervalo de tiempo.

El teorema del muestreo fija la frecuencia de digitalización (frecuencia de muestreo) mínima necesaria para que no se produzca pérdida de información al transformar una señal continua  $x(t)$  en una serie de valores discretos  $(x_k)$ . El teorema de muestreo afirma que si  $x(t)$  es limitada en banda<sup>1</sup>, la función  $x(t)$  en cualquier punto puede ser reconstruida a partir de sus valores muestreados  $(x_k)$  si la digitalización se ha efectuado con un  $\Delta \leq 1/2 f_{max}$ . [11]

Teorema de Nyquist: "Si una señal se muestrea para todo tiempo a una tasa mayor que el doble de la frecuencia más alta a la cual su TFTC es distinta de cero, entonces puede reconstruirse exactamente a partir de las muestras."

## 2.4. Tipos de convertidores

Hay varias formas de convertidores analógico/digitales, siendo los más comunes el de aproximaciones sucesivas, flash, rampa, doble rampa y tensión-frecuencia. Los de aproximaciones sucesivas, flash y rampa son ejemplos de los que pueden conocerse como ADC de muestreo; ellos proporcionan el valor digital equivalente a la tensión en el instante en que la señal es muestreada. Los de doble rampa y tensión-frecuencia son ejemplos de ADC integradores en los que se promedia el valor de tensión sobre un tiempo de medida fijo. Los convertidores de este tipo tardan más tiempo para dar una medida pero tienen mejor rechazo al ruido.

### 2.4.1. Aproximaciones Sucesivas

El ADC de aproximaciones sucesivas es uno de los ADCs más usados y tiene como parte de sus características fundamentales la conversión en lapsos de tiempo corto. Su modo de operación es el siguiente: toma una muestra de la tensión analógica de entrada y, después, se compara con la tensión que crece en incrementos hasta que se llega al valor total de la tensión de entrada. Este crecimiento incremental de tensión se produce por un reloj que emite una secuencia regular de impulsos que se cuentan y convierten en una señal analógica mediante un convertidor digital/analógico. La señal analógica resultante se compara con la tensión de entrada y cuando crece al nivel superior, se detienen los impulsos del reloj para leer el contador y, por tanto, la lectura del contador es el equivalente digital de la tensión de entrada analógica, en la figura 2.3 se muestra el diagrama de bloques de este ADC. [12]

---

<sup>1</sup> Una función  $x(t)$  se dice que es limitada en banda si  $X(f)$  es nula para frecuencias  $f$  tales que  $|f| > f_{max}$ .

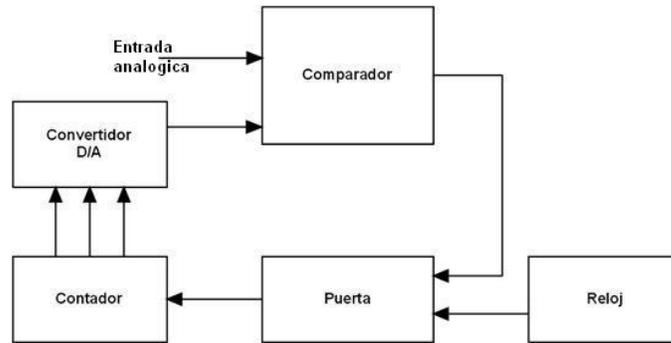


Figura 2.3. Convertidor A/D de aproximaciones sucesivas.

La figura 2.4 muestra el algoritmo del convertidor, se empieza por el bit de mayor peso hasta el bit de menor peso. El primer nivel de comparación es la mitad de la tensión de fondo de escala (FE) y el incremento del nivel de comparación es a cada paso la mitad del incremento del paso anterior.

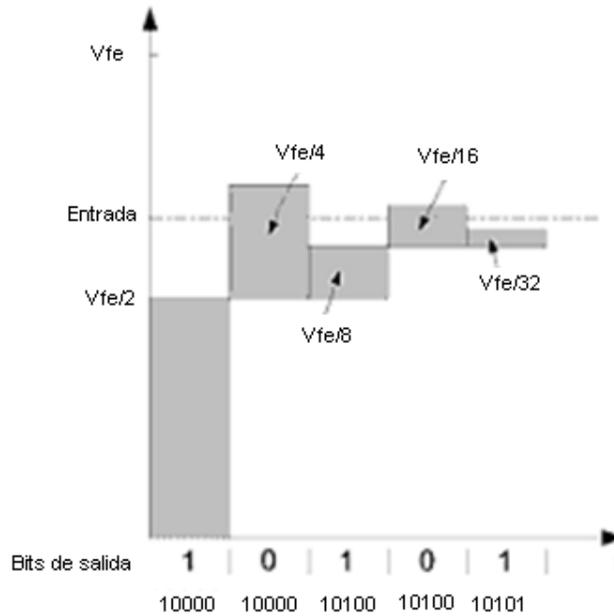


Figura 2.4. Diagrama del algoritmo del convertidor.

El tiempo de conversión de este método es una constante dada por:

$$T = \frac{N}{f_{clk}} \tag{3}$$

Donde:

$N$  = Número de bits en el conversor

$f_{clk}$  = Frecuencia del reloj

### 2.4.2. Convertidor ADC rampa

El convertidor de rampa es el más simple y barato (figura 2.5), la tensión analógica de entrada se aplica al comparador y se toma un tiempo para que la rampa de tensión suba desde 0 V hasta el valor de tensión analógica medida. Este tiempo se obtienen en forma digital mediante un contador del número de impulsos producidos por un reloj durante el tiempo en que la puerta está abierta; la puerta está abierta cuando empieza la rampa y cerrada cuando la rampa y la tensión analógica son iguales. [9]

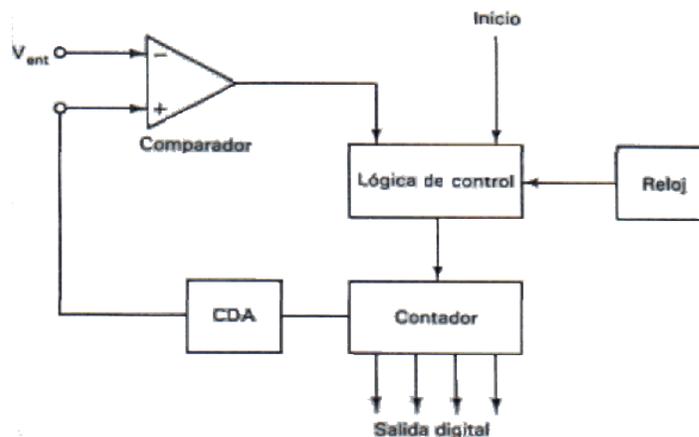


Figura 2.5. Convertidor rampa.

### 2.4.3. Convertidor ADC Doble Rampa

En un ADC de doble rampa (figura 2.6) se convierte la tensión analógica de entrada en una corriente proporcional que se integra cargando un capacitor durante un tiempo fijo determinado por un oscilador de frecuencia estable (primera rampa). El condensador se descarga luego (segunda rampa) mediante una corriente proporcional a una tensión de referencia interna, conocida y estable, y de signo opuesto a la entrada. El tiempo que tarda en descargarse el capacitor es proporcional a la tensión de entrada. Esta clase de convertidor A/D tiene la ventaja del rechazo al ruido. [10]

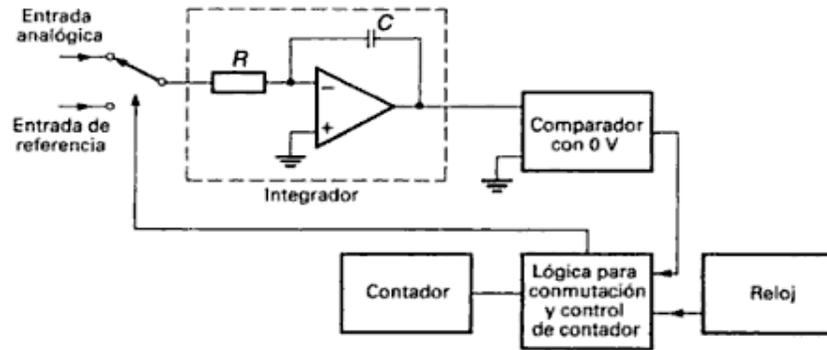


Figura 2.6. Convertidor doble rampa.

#### 2.4.4. Convertidor de Tensión Frecuencia

Con el convertidor tensión frecuencia (figura 2.7), la tensión analógica de entrada se convierte en una colección de impulsos cuya frecuencia es proporcional al tamaño de la tensión de entrada. La frecuencia se determina entonces contando el número de impulsos existentes en un intervalo de tiempo fijo. Se utiliza un integrador para integrar la tensión analógica de entrada  $V_{in}$  sobre el tiempo que tarda la salida del integrador para ir de 0 a  $.V_{ref}$ . Al final del tiempo, un generador de impulsos emite un impulso simple. Como la entrada al integrador todavía está presente, el proceso se repite de nuevo obteniéndose finalmente una secuencia de impulsos con una frecuencia determinada por  $V_{in}$ . [13]

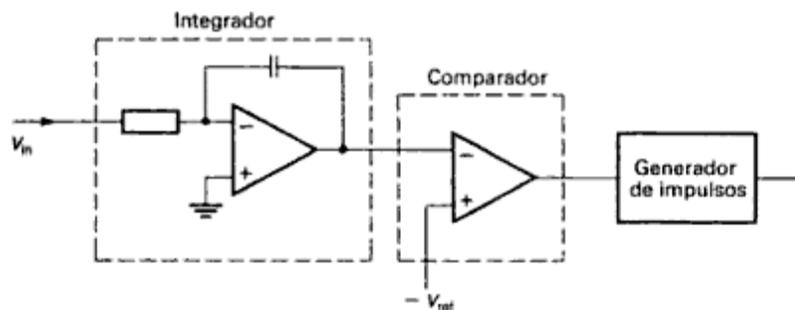


Figura 2.7. Convertidor A/D de tensión-frecuencia.

## 2.5. El Osciloscopio

El osciloscopio es un dispositivo de presentación de gráficas, es decir traza una gráfica de una señal eléctrica, en la mayoría de las aplicaciones, esta gráfica muestra cómo cambia la señal con el tiempo: el eje vertical (Y) representa el voltaje, el eje horizontal (X) representa el tiempo. La intensidad o brillo de la pantalla se denomina, a veces, eje Z (ver figura 2.8).

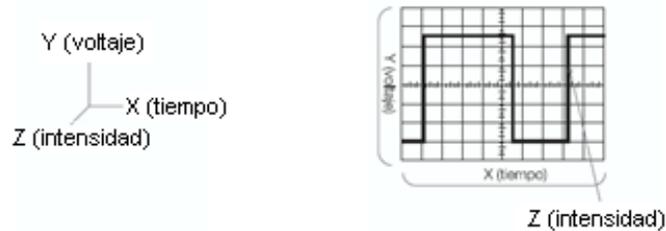


Figura 2.8. Representación gráfica en un osciloscopio.

Los osciloscopios se pueden clasificar en analógicos y digitales. Para muchas aplicaciones dará igual un osciloscopio analógico que uno digital. Sin embargo, cada uno de ellos tiene características únicas que lo pueden hacer más o menos conveniente para aplicaciones específicas. Los osciloscopios digitales, a su vez, pueden ser clasificados en osciloscopios de memoria digital (DSO: Digital Storage Oscilloscope), osciloscopios de fósforo digital (DPO: Digital phosphor oscilloscopes), y osciloscopios de muestreo.

Un osciloscopio digital utiliza un convertidor analógico digital para convertir el voltaje medido en información digital. Estos osciloscopios adquieren la forma de onda como una serie de muestras y las almacenan hasta que acumulan muestras suficientes como para describir una forma de onda. El osciloscopio digital reconstruye entonces la forma de onda para su representación en pantalla. La técnica digital permite que el osciloscopio pueda representar cualquier frecuencia dentro de su rango, con estabilidad, brillante y claridad. Para señales repetitivas, el ancho de banda del osciloscopio digital es una función del ancho de banda analógico de los componentes de entrada del osciloscopio, comúnmente conocido como el punto de atenuación -3dB. Para eventos transitorios y de disparo único, tales como pulsos y escalones, el ancho de banda puede verse limitado por la velocidad de muestreo del osciloscopio. Un osciloscopio digital convencional se conoce como osciloscopio de memoria digital o simplemente, osciloscopio digital (DSO). Los osciloscopios de memoria digital permiten la captura y visualización de eventos que ocurren solamente una vez y a los que se conoce como transitorios. Debido a que la información de la forma de onda existe en forma digital, como una serie de valores binarios almacenados, ésta puede ser almacenada, analizada, archivada y procesada de cualquier otra forma dentro del propio

osciloscopio o por un ordenador externo. No es necesario que la forma de onda sea continua, y puede ser mostrada en pantalla incluso cuando la señal ha desaparecido. Los DSO consisten en subsistemas de procesado de datos que se utilizan para recomponer y mostrar los datos de la forma de onda completa. Un DSO utiliza una arquitectura de procesado en serie para capturar y mostrar una señal en su pantalla, como se indica en la figura 2.9. [14]



Figura 2.9. Arquitectura de procesado en serie de un osciloscopio digital DSO.

La primera etapa de un DSO es un amplificador vertical. Los controles verticales permiten ajustar la amplitud y el rango de posición. El convertidor analógico digital (CAD) del sistema horizontal muestrea las señales en puntos aislados en el tiempo y convierte el voltaje de la señal presente en estos puntos en valores digitales, denominados puntos de muestreo. El reloj de muestreo del sistema horizontal determina con qué frecuencia el CAD recoge muestras. Esta velocidad se llama velocidad de muestreo y se expresa en muestras por segundo (S/s). Las muestras del CAD son almacenadas en la memoria de adquisición como puntos de la forma de onda. Varias muestras pueden conformar un punto de la forma de onda. Todos los puntos de la forma de onda en conjunto conforman el registro de forma de onda. El número de puntos de forma de onda utilizados para crear un registro se denomina longitud de registro. El disparo del sistema determina los puntos de comienzo y parada del registro.

El osciloscopio de fósforo digital (DPO) utiliza una arquitectura en paralelo (figura 2.10) para llevar a cabo las mismas funciones que el DSO. La arquitectura del DPO utiliza hardware ASIC (Circuito Integrado para Aplicaciones Específicas) para adquirir imágenes de las formas de onda, proporcionando altas velocidades de captura de forma de onda, que redundan en un elevado nivel de visualización de la señal. Esta prestación aumenta la probabilidad de observar los eventos transitorios que ocurren en sistemas digitales, tales como pulsos de escasa amplitud, espurios y errores de transición. La primera etapa de un DPO es similar a la de un osciloscopio analógico, un amplificador vertical, y su segunda etapa es similar a la de un DSO, un ADC. En cualquier osciloscopio (analógico, DSO, DPO) hay siempre un tiempo de retención (holdoff) durante el cual el instrumento procesa los datos más recientemente adquiridos, restaura el sistema, y espera el siguiente evento de disparo. Durante este tiempo, el osciloscopio está ciego ante cualquier actividad de la señal. La probabilidad de ver un evento infrecuente o de baja repetición, disminuye conforme aumenta este tiempo de retención. Debe tenerse en cuenta que es imposible determinar la probabilidad de captura simplemente analizando la velocidad de actualización de la pantalla. Si

se depende solamente de la velocidad de actualización, es fácil cometer el error de creer que el osciloscopio está capturando toda la información pertinente acerca de la forma de onda cuando, en realidad no es así. [14]

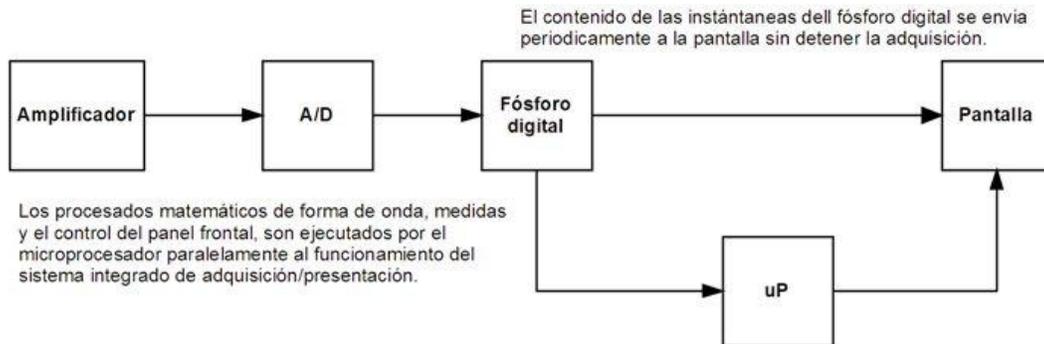


Figura 2.10. Arquitectura en paralelo de un osciloscopio de fósforo digital (DPO).

El DPO explora los datos digitalizados de la forma de onda sobre una base de datos de fósforo digital (así se conoce a la base de datos donde se almacenan las formas de onda). Cada 1/30 de segundo se envía directamente al sistema de presentación una instantánea de la imagen de la señal que está almacenada en la base de datos. Esta exploración directa de los datos de la forma de onda y su copia directa en la memoria de presentación desde la base de datos, elimina el cuello de botella del procesado de datos (que se da en una arquitectura en serie). El resultado es una mejora del tiempo de actividad del osciloscopio y una viva actualización de la presentación. Los detalles de la señal, los eventos intermitentes y las características dinámicas de la señal, se capturan en tiempo real. El microprocesador del DPO trabaja en paralelo con este sistema de adquisición integrado para la gestión de la presentación, la automatización de las medidas y el control del instrumento, para que ello no afecte a la velocidad de adquisición del osciloscopio. Un DPO utiliza un fósforo digital puramente electrónico que, en realidad, es una base de datos constantemente actualizada. Los osciloscopios de fósforo digital (DPO) son apropiados para la visualización de altas y bajas frecuencias, para formas de onda repetitivas, transitorios y para variaciones de la señal en tiempo real. Solamente un DPO proporciona el eje Z (intensidad) en tiempo real, el cual no está disponible en los DSO convencionales. Una vez adquirida la señal analógica es preciso muestrearla para así poder ser procesada digitalmente. En la pantalla de un osciloscopio digital, se reconstruye un conjunto de puntos muestreados, mostrando la amplitud medida en el eje vertical y el tiempo en el eje horizontal, tal como se ilustra en la figura 2.11. La señal de la forma de onda en la figura 2.8 aparece como una serie de puntos en la pantalla. Si los puntos están ampliamente espaciados y es difícil interpretarlos como una forma de onda, pueden ser conectados utilizando un proceso denominado interpolación. La interpolación interconecta los puntos con líneas o vectores. Se dispone de determinados métodos de interpolación que se

pueden utilizar para crear una representación precisa y continua de una señal de entrada. [14]

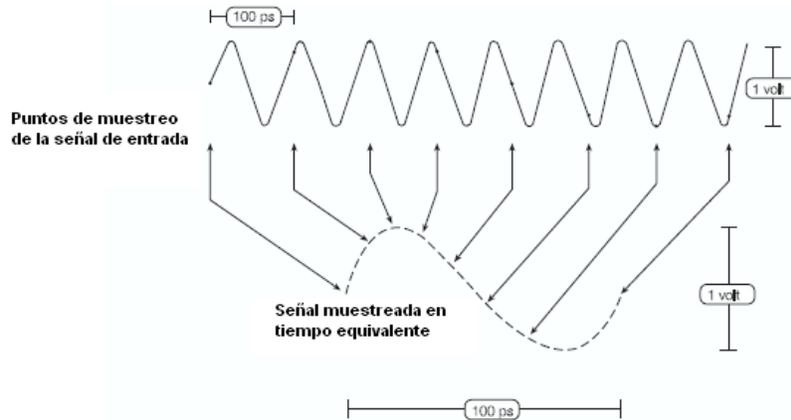


Figura 2.11. Muestreo básico. Los puntos muestreados se conectan por interpolación para crear una forma de onda continua.

Algunos osciloscopios digitales proporcionan una alternativa en el método de muestreo, ya sea en tiempo real o en tiempo equivalente. Los controles de adquisición disponibles en estos osciloscopios permiten seleccionar un método de muestreo para la adquisición de las señales. Cada método posee diferentes ventajas, dependiendo del tipo de medidas que se desee realizar.

El **muestreo en tiempo real** resulta ideal para señales cuyo rango de frecuencia es menor que la mitad de la velocidad máxima de muestreo del osciloscopio. En este caso, el osciloscopio puede adquirir puntos más que suficientes con un solo "barrido" de la forma de onda como para conformar una imagen precisa (figura 2.12), El muestreo en tiempo real es la única forma de capturar señales transitorias rápidas de ocurrencia única con un osciloscopio digital.

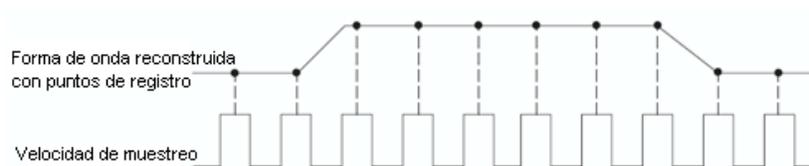


Figura 2.12. Método de muestreo en tiempo real.

El muestreo en tiempo real representa una gran dificultad para los osciloscopios digitales debido a la velocidad de muestreo que se requiere para digitalizar con precisión eventos transitorios de alta frecuencia. En el caso **del muestreo en tiempo real con interpolación** los osciloscopios digitales toman muestras individuales de la señal que puede ser presentada, en algunos casos puede resultar difícil visualizar la señal representada por puntos, sobre todo porque puede haber

solamente unos pocos puntos representando porciones de alta frecuencia de la señal. Para ayudar en la visualización de las señales, los osciloscopios digitales típicamente tienen modos de presentación con interpolación. Al utilizar muestreo en tiempo real con interpolación, el osciloscopio recoge unos pocos puntos de muestreo de la señal en un solo barrido y en modo de tiempo real, y utiliza la interpolación para rellenar los espacios intermedios. La interpolación lineal conecta los puntos de las muestras mediante líneas rectas (figura 2.14). Este método está limitado a la reconstrucción de señales de flancos rectos, tales como las ondas cuadradas. La interpolación seno  $x/x$  es más versátil, ya que conecta los puntos de las muestras mediante curvas (figura 2.14). Este tipo de interpolación es un proceso matemático en el que se calculan los puntos que rellenarán el espacio entre las muestras reales. La interpolación mencionada se presta más a formas de señales curvadas e irregulares, que son mucho más habituales en el mundo real que las puras ondas cuadradas y los pulsos. En consecuencia, la interpolación seno  $x/x$  es el método preferido para aplicaciones donde la velocidad de muestreo es de 3 a 5 veces el ancho de banda del sistema. [14]

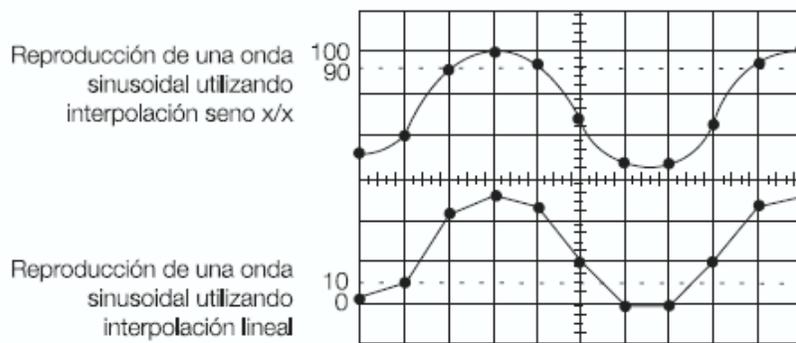


Figura 2.14. Métodos de interpolación.

Cuando se miden señales de alta frecuencia, el osciloscopio puede no ser capaz de recoger suficientes muestras en un barrido. El **muestreo en tiempo equivalente** se puede utilizar entonces para adquirir con precisión señales cuya frecuencia excede la mitad de la velocidad de muestreo del osciloscopio. Los digitalizadores (sistemas de muestreo) en tiempo equivalente se aprovechan de la circunstancia de que la mayoría de los eventos naturales y los producidos por el hombre son repetitivos. El muestreo en tiempo equivalente construye una imagen de una señal repetitiva, capturando tan solo una parte de la información en cada repetición. La forma de onda se va creando lentamente como una cadena de puntos luminosos, que se ilumina uno a uno. Este modo permite al osciloscopio capturar con precisión señales cuyas componentes de frecuencia son mucho más elevadas que la velocidad de muestreo del osciloscopio.

## 2.6. Buses de transmisión de datos

Para establecer la comunicación entre nuestro dispositivo de medición y la PC, y poder transmitir datos, es necesario tener en cuenta que interfaz se va a emplear, también hay que tomar en cuenta que desde el punto de vista de instrumentación virtual, el aspecto más importante de estos instrumentos será su posibilidad de ser controlados remotamente. Los buses más empleados en instrumentación son el USB, Ethernet, GPIB y PXI.

### 2.6.1. GPIB

GPIB (General Purpose Interface Bus, figura 2.15) es un estándar de conexión que permite la comunicación de una PC con instrumentos electrónicos de medida, como pueden ser generadores de funciones, osciloscopios, etc. El bus GPIB fue inventado por Hewlett Packard, a finales de los años 60s.

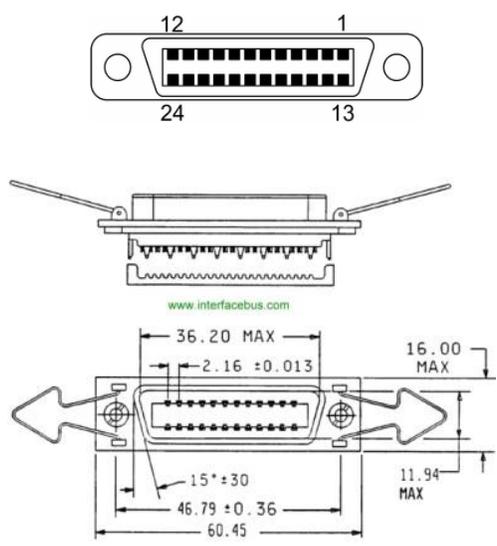


Figura 2.15. Bus GPIB.

La intención de crear el bus GPIB era la de disponer de un bus fiable, especialmente diseñado para conectar computadoras e instrumentos en una configuración de red que poseyera las características requeridas por un equipo de medida. Se le denominó originalmente HP-IB y se popularizó con rapidez, debido a sus altas tasas de transferencia de datos (8 Mbytes/s). En 1975, para evitar la dispersión de características, los principales fabricantes acordaron la estandarización del GPIB (IEEE 488.1), centrándose en las características eléctricas y mecánicas del bus. En 1987 tuvo lugar una segunda estandarización (IEEE 488.2), que delimitó de forma más concreta la programación del GPIB, definiendo comandos de aparatos, formato de mensajes y estado de los instrumentos. En 1990 se adoptó el formato de comandos SCPI, incluido en el 488.2, que estructura las órdenes a los aparatos de forma coherente, permitiendo una compatibilidad más extensa. National Instruments introdujo una extensión retro-compatible a IEEE 488.1, conocida originalmente como HS-488. Esta incrementa la velocidad máxima a 8 MB/s, aunque la velocidad

disminuye a medida que se conectan más dispositivos al bus. Fue incorporada al estándar en 2003, como IEEE 488.1-2003. El bus de transmisión de datos GPIB es de 8 bits en paralelo, y lógica negativa con niveles TTL estándar (T si el voltaje < 0.8 V y F si el voltaje es > 2.0 V). El IEEE-488 permite que hasta 15 dispositivos inteligentes compartan un simple bus paralelo, mediante conexión en cadena, con el dispositivo más lento determinando la velocidad de transferencia. La máxima velocidad de transmisión está sobre 1 Mbps en el estándar original y en 8 Mbps con IEEE-488.1-2003 (HS-488). Las 16 líneas que componen el bus están agrupadas en tres grupos de acuerdo con sus funciones: 8 de bus de datos, 3 de bus de control de transferencia de datos y 5 de bus general. Algunas de ellas tienen retornos de corrientes comunes y otras tienen un retorno propio, lo que provoca un aumento del número de líneas totales (8 masas). La separación máxima entre dos dispositivos es de 4 m, y la separación promedio en toda la red debe ser menor de 2 m (National Instruments comercializa un extensor de fibra óptica (GPIB 140 y 140/2) que permite alcanzar una longitud de hasta 2 km), la longitud total de la red no debe exceder los 20 m. En la tabla 2.1 se muestra la asignación de terminales del bus GPIB. [15,16]

**Tabla 2.1.** Asignación de terminales para el bus GPIB.



Pin	Signal	Description
Pin 1	DIO1	Data input/output bit.
Pin 2	DIO2	Data input/output bit.
Pin 3	DIO3	Data input/output bit.
Pin 4	DIO4	Data input/output bit.
Pin 5	EOI	End-of-identify.
Pin 6	DAV	Data valid.
Pin 7	NRFD	Not ready for data.
Pin 8	NDAC	Not data accepted.
Pin 9	IFC	Interface clear.
Pin 10	SRQ	Service request.
Pin 11	ATN	Attention.
Pin 12	SHIELD	
Pin 13	DIO5	Data input/output bit.
Pin 14	DIO6	Data input/output bit.
Pin 15	DIO7	Data input/output bit.
Pin 16	DIO8	Data input/output bit.
Pin 17	REN	Remote enable.
Pin 18	GND	(emparejado con DAV)
Pin 19	GND	(emparejado con NRFD)
Pin 20	GND	(emparejado con NDAC)
Pin 21	GND	(emparejado con IFC)
Pin 22	GND	(emparejado con SRQ)
Pin 23	GND	(emparejado con ATN)

### 2.6.2. PXI

Extensiones PCI<sup>2</sup> para Instrumentación (PXI: PCI eXtensions for Instrumentation) es una plataforma basada en PC que ofrece una solución de despliegue de alto rendimiento y bajo costo para sistemas de medida y automatización. PXI (figura 2.16) combina el bus eléctrico de Interconexión de Componentes Periféricos (PCI) con el robusto y modular paquete Eurocard de CompactPCI, y añade buses de sincronización especializados y características clave de software. PXI también añade características mecánicas, eléctricas y de software que definen sistemas completos para aplicaciones de pruebas y medidas, de adquisición de datos y de manufactura. Estos sistemas son útiles para aplicaciones tales como militares, aeroespaciales y automotrices, y para pruebas tales como de manufactura e industriales. National Instruments desarrolló y anunció la especificación PXI en 1997 y la lanzó en 1998 como una especificación abierta de industria para cubrir la creciente demanda de sistemas complejos de instrumentación. Actualmente, PXI es gobernada por PXI Systems Alliance (PXISA), un grupo de más de 50 compañías comisionadas para promocionar el estándar, asegurar su inter operatividad y conservar la especificación PXI. Debido a que PXI es una especificación abierta, cualquier proveedor puede construir productos PXI. CompactPCI, el estándar regulado por el PCI Industrial Computer Manufacturers Group (PICMG), y los módulos PXI pueden ocupar el mismo sistema PXI sin ningún problema debido a que la inter operatividad entre CompactPCI y PXI es una característica clave en la especificación PXI. [17]

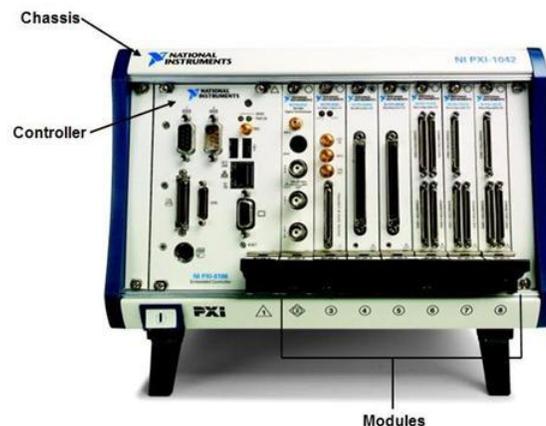


Figura 2.16. Sistema PXI.

De la misma manera en que la industria de PCs comerciales mejoró significativamente el ancho de banda del bus disponible al evolucionar del PCI a PCI Express, a finales del 2005, PXI también incorporó mayores capacidades de ancho de banda de bus con la introducción de PXI Express.

---

<sup>2</sup> *Peripheral Component Interconnect, consiste en un bus de PC estándar para conectar dispositivos periféricos directamente a su placa base*

PXI tiene la habilidad de cubrir aún más necesidades de aplicación al integrar PCI Express al estándar PXI. La tecnología PCI Express se puede integrar al plano trasero mientras conserva la compatibilidad con la gran base de instalación de sistemas existentes. La ranura del controlador del sistema es capaz de conectarse a enlaces PCI Express de hasta 16x (ancho del bus) además de enlaces de 1x, 4x y 8x, los cuales proporcionan un ancho de banda de hasta 6 GB/s al plano trasero del PXI Express. Al aprovechar la tecnología PCI Express, PXI Express incrementa el ancho de banda disponible de 132 MB/s con PXI a 6 GB/s para un mejoramiento en ancho de banda multiplicado por 45, mientras aún mantiene compatibilidad de software y hardware con módulos PXI. Con este rendimiento mejorado, PXI puede alcanzar muchas nuevas áreas de aplicación, muchas de las cuales sólo podían ser cubiertas por hardware costoso y patentado.

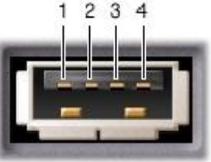
La mayoría de los chasis contienen una ranura de controlador del sistema, por lo cual uno tiene opciones a elegir al determinar cuál es el mejor controlador de sistema para una aplicación, como son, controladores remotos desde una PC de escritorio o controladores embebidos de alto rendimiento. Los controladores embebidos eliminan la necesidad de una PC externa, proporcionando un sistema completo contenido dentro del chasis PXI. Estos controladores embebidos poseen dispositivos estándares tales como CPU integrado, disco duro, RAM, Ethernet, video, teclado/mouse, puerto serial, USB y otros periféricos, así como Windows Microsoft y todos los controladores de dispositivos instalados. Para controlarlos desde la PC, hace falta una conexión que haga puente entre el PXI y la PC, por lo regular este puente lo hace una tarjeta PCI. Los módulos PXI se pueden conectar en cascada, una vez conectados en la PC.

### **2.6.3. USB**

Actualmente han ido ganando auge los dispositivos para instrumentación USB (Universal Serial Bus), ya que estos permiten la portabilidad y compatibilidad con casi todas las PCs actuales. El USB es un puerto que sirve para conectar periféricos a una computadora. Fue creado en 1996 por siete empresas: IBM, Intel, Northern Telecom, Compaq, Microsoft, Digital Equipment Corporation y NEC. El diseño del USB tenía en mente eliminar la necesidad de adquirir tarjetas separadas para poner en los puertos bus ISA o PCI, y mejorar las capacidades plug-and-play, permitiendo a esos dispositivos ser conectados o desconectados al sistema sin necesidad de reiniciar. Las señales del USB se transmiten en un cable de par trenzado con impedancia característica de  $90 \Omega \pm 15\%$ , cuyos hilos se denominan D+ y D-. Estos, colectivamente, utilizan señalización diferencial en full dúplex para combatir los efectos del ruido electromagnético en enlaces largos. D+ y D- suelen operar en conjunto y no son conexiones simples. Los niveles de transmisión de la señal varían de 0 a 0.3 V para niveles de voltaje bajos ("0") y de 2.8 a 3.6 V para niveles de voltaje altos ("1") en las versiones 1.0 y 1.1, y en  $\pm 400$  mV en alta velocidad (2.0). En las primeras versiones, los alambres

de los cables no están conectados a tierra, pero en el modo de alta velocidad se tiene una terminación de  $45 \Omega$  a tierra o un diferencial de  $90 \Omega$  para acoplar la impedancia del cable. Este puerto sólo admite la conexión de dispositivos de bajo consumo, es decir, que tengan un consumo máximo de 100 mA por cada puerto; sin embargo, en caso de que estuviese conectado un dispositivo que permite 4 puertos por cada salida USB (extensiones de máximo 4 puertos), entonces la energía del USB se asignará en unidades de 100 mA hasta un máximo de 500 mA por puerto. La tabla 2.2 muestra la asignación de terminales para este bus. [18]

**Tabla 2.2.** Asignación de terminales para el bus USB.

	Terminal 1	VCC
	Terminal 2	-Data
	Terminal 3	+Data
	Terminal 4	Ground

Los dispositivos USB se clasifican según su velocidad de transferencia de datos:

- Baja velocidad (1.0): Tasa de transferencia de hasta 1,5 Mbps (192 KB/s). Utilizado en su mayor parte por dispositivos de interfaz humana (Human interface device) como los teclados, los ratones y artículos del hogar.
- Velocidad completa (1.1): Tasa de transferencia de hasta 12 Mbps (1,5 MB/s), según este estándar, pero se dice en fuentes independientes que habría que realizar nuevamente las mediciones. Ésta velocidad fue la más rápida antes de la especificación USB 2.0, y muchos dispositivos fabricados en la actualidad trabajan a esta velocidad.
- Alta velocidad (2.0): Tasa de transferencia de hasta 480 Mbps (60 MB/s) pero por lo general de hasta 125Mbps (16MB/s). Está presente en el 99% de los CPUs actuales. El cable USB 2.0 dispone de cuatro líneas, un par para datos, una de voltaje y una de toma de tierra.

#### 2.6.4. Ethernet

Recientemente, los fabricantes de instrumentos han empezado a incluir Ethernet como una interface de comunicación alternativa. A pesar de que Ethernet es nuevo en las aplicaciones de control, es una tecnología madura que es ampliamente utilizada para sistemas de medición. Con más de 100 millones de computadoras alrededor del mundo con capacidad Ethernet, el argumento de utilizarlo como una solución para el control de instrumentos tiene mucho peso.

Las aplicaciones de control de instrumentos con Ethernet pueden tomar ventajas de las características únicas del bus, las cuales incluyen control remoto, facilidad para compartir los

instrumentos entre usuarios, y una fácil publicación de los datos obtenidos. Por otra parte, los usuarios toman ventaja de las extensas redes Ethernet existentes en sus compañías y laboratorios. Sin embargo, esta ventaja podría ser un problema en algunas compañías, ya que requerirá la intervención de administradores de red en las aplicaciones tradicionales de ingeniería.

Otros factores a considerar al examinar Ethernet para el control de instrumentos son: la velocidad de transferencia, el determinismo y la seguridad. La mayoría de las redes Ethernet de hoy en día son del tipo 10BaseT ó 100BaseTX, las cuales transmiten datos a 10 Mb/s ó 100 Mb/s respectivamente. Sin embargo, estas tasas de transferencia son teóricas debido a factores como: tráfico en la red, retrasos e ineficiencia en la transferencia de datos. Por otra parte, debido a la incertidumbre en las tasas de transferencia, no se puede asegurar un determinismo en la comunicación a través de Ethernet. Finalmente, los usuarios que requieran integridad y privacidad en los datos deberán tomar medidas de precaución especiales. En la tabla 2.3 se pueden ver las características para cada tecnología de Ethernet. [19]

**Tabla 2.3.** Tecnologías del bus Ethernet.

Tecnología	Velocidad de transmisión	Tipo de cable	Distancia máxima
10Base2	10 Mbps	Coaxial	185 m
10BaseT	10 Mbps	Par Trenzado	100 m
10BaseF	10 Mbps	Fibra óptica	2000 m
100BaseT4	100Mbps	Par Trenzado (categoría 3UTP)	100 m
100BaseTX	100Mbps	Par Trenzado (categoría 5UTP)	100 m
100BaseFX	100Mbps	Fibra óptica	2000 m
1000BaseT	1000Mbps	4 pares trenzado (categoría 5e ó 6UTP )	100 m
1000BaseSX	1000Mbps	Fibra óptica (multimodo)	550 m
1000BaseLX	1000Mbps	Fibra óptica (monomodo)	5000 m

# Diseño e Implementación

### 3.1. Introducción

El objetivo principal de la aplicación es facilitar el manejo y control de un sistema posicionador de 3 ejes, el cual se utilizara principalmente en la caracterización de transductores ultrasónicos. Los datos adquiridos con el sistema serán almacenados en la PC para su posterior procesamiento. En la figura 3.1 podemos observar un diagrama general del sistema a controlar. El sistema cuenta básicamente con dos etapas: la de control del posicionador y la de adquisición de datos acústicos provenientes del posicionador XYZ, por medio del osciloscopio. En la etapa de control contamos con lo que es el circuito de control, la etapa de potencia y los motores a controlar. Para mandar las señales de control hacia los motores se empleó una tarjeta digital E/S NI USB-6501 de 24 canales digitales E/S (circuito de control); debido a que los motores a pasos ocupados necesitan una alimentación de 600mA es necesario añadir una etapa de potencia se optó por utilizar un arreglo de transistores Darlington, los motores a pasos, son unipolares de 200 pasos por vuelta, con una alimentación de 12v. Adicionalmente al control del posicionador también es necesario un dispositivo para adquirir los datos, para esto se emplea un osciloscopio de fósforo digital Tektronix (DAQ). Las características de cada dispositivo se muestran en el anexo A. Con

base en las características del hardware y el conocimiento de las pruebas que en ella se realizarán, se estableció que el software debe cumplir con los siguientes requerimientos:

- Adquisición automática de datos.
- Control de la operación del osciloscopio.
- Control manual (desde el teclado) del posicionador.
- Mostrar en pantalla los datos que se estén adquiriendo.

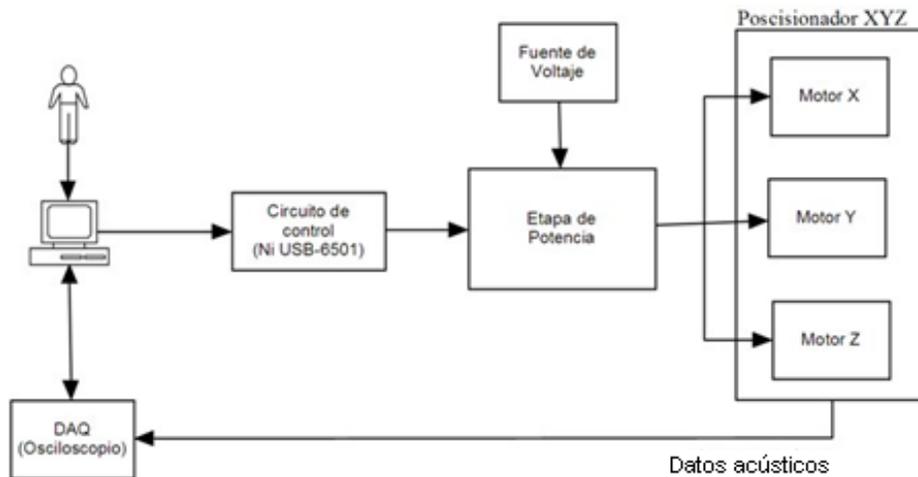


Figura 3.1. Diagrama general del sistema.

Una vez establecidos los dispositivos con los que cuenta el sistema posicionador XYZ, en los siguientes temas se dará una breve explicación de los principales elementos que permiten la programación en LabVIEW (Necesario para entender el código de LabVIEW mostrado en figuras posteriores). Posteriormente se explica la forma en la que se realizó la caracterización del posicionador XYZ con ayuda de una aplicación hecha en LabVIEW. Por último se menciona la forma de programar un VI para el control del osciloscopio. Los pasos que se siguieron para la realización del proyecto fueron:

- 1.\_ Caracterización del osciloscopio
- 2.\_ Desarrollo de un VI para el control óptimo de los motores a pasos.
- 3.\_ Desarrollo de un VI para el control óptimo del osciloscopio.
- 4.\_ Desarrollo de la interfaz (Se conjuntaron los VIs que se programaron de forma independiente y se completó la programación de la interfaz).

Cabe resaltar que se contaba con todo el equipo desde el inicio, por lo que únicamente el proyecto está enfocado en la construcción de la interfaz de usuario, sin meternos a profundidad en la parte física (hardware).

### 3.2. Herramientas básicas de LabVIEW

Como ya se había indicado anteriormente, un programa creado en LabVIEW consta de dos partes: el panel frontal y el diagrama de bloques. El panel frontal (panel de control) es el que simula el panel de un instrumento físico, nos muestra la interfaz con el usuario. En lo que se conoce en LabVIEW como “diagrama de bloques” se aprecia la estructura del programa (el código del programa que se está desarrollando), en el cual los datos “fluyen” a través de líneas. Tanto en el panel frontal como en el diagrama de bloques, existe una paleta de herramientas (figura 3.2), que sirve para editar el programa.

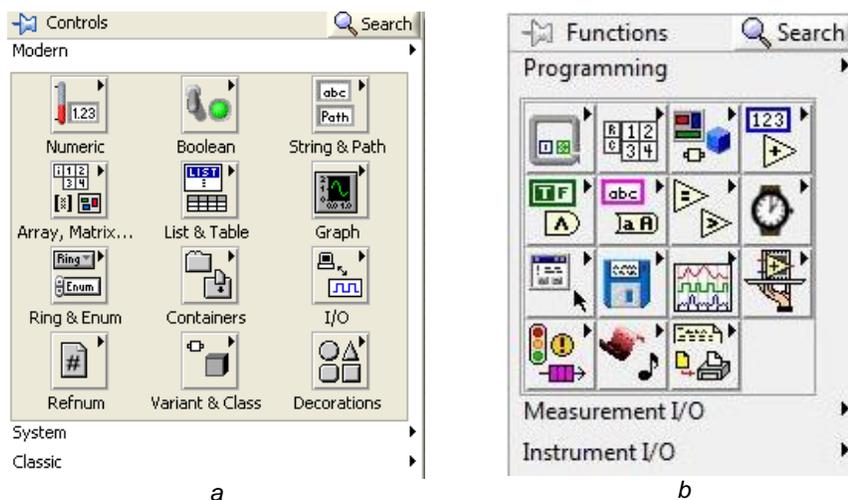


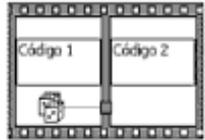
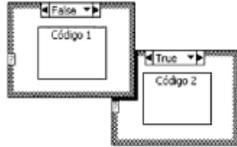
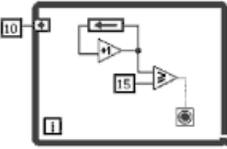
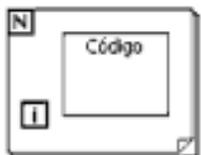
Figura 3.2. a) Paleta de control, b) Paleta de funciones.

En el panel frontal existe la paleta de controles (Controls Palette) que contiene indicadores pudiéndolos visualizar como tablas, gráficas, botones, etc. En el “diagrama de bloques” tenemos la paleta de funciones, donde se encuentran todas las funciones utilizadas dentro de la programación. Hay que tener en cuenta que cuando se realiza una conexión en un programa, para que exista concordancia con el flujo de datos, estos deben coincidir con el mismo tipo de datos. El flujo de datos va de izquierda a derecha en el “diagrama de bloques” y está determinado por las operaciones o funciones que procesan los datos. La ejecución del programa es secuencial, es decir, una tarea no se inicia hasta no tener todas sus variables de entrada de información, o que las tareas que le preceden hayan terminado de ejecutarse.

Las instrucciones de control permiten a un programa ejecutar un código de forma condicional o repetirlo cierto número de veces. En LabVIEW estas instrucciones son estructuras que encierran

en su interior el código al que afectan. Las estructuras que encontramos se muestran en la tabla 3.1.

**Tabla 3.1.** Principales estructuras de LabVIEW.

	<p>La estructura secuencial ejecuta el código en el orden que se encuentre dentro, es decir, primero ejecuta lo que se encuentre en la primera estructura, luego lo que se encuentre en la siguiente.</p>
	<p>La estructura "case" se emplea cuando dos o mas acciones alternativas dependen de una condición.</p>
	<p>La estructura "while" ejecuta una acción, hasta que se le indique lo contrario.</p>
	<p>La estructura "for" repite el código que se encuentre en su interior un determinado número de veces.</p>

Con base en las estructuras mencionadas podemos observar dos tipos de programación, estructurada y modular. En la estructurada el código se ejecuta de forma secuencial, como ya se mencionó, en la modular se divide el programa en partes que tengan una personalidad propia, es decir, dividir el programa en varios subprogramas que ahorren tiempo y esfuerzo a la hora de realizar y ejecutar el programa principal. [20,21]

### 3.3. Comunicación con la tarjeta NI USB-6501

Antes de abordar el problema planteado inicialmente en el trabajo, debemos saber como mandar señales lógicas con la tarjeta de control, para esto LabVIEW tiene implementados su propio *driver* para sus tarjetas (Ni-DAQ), la cual facilita la comunicación tanto con tarjetas I/O analógicas como con tarjetas I/O digitales. En la figura 3.3 observamos el código ocupado para la comunicación con la tarjeta. Este código nos sirve para realizar la programación de los motores, el cual se explicara en el tema siguiente.



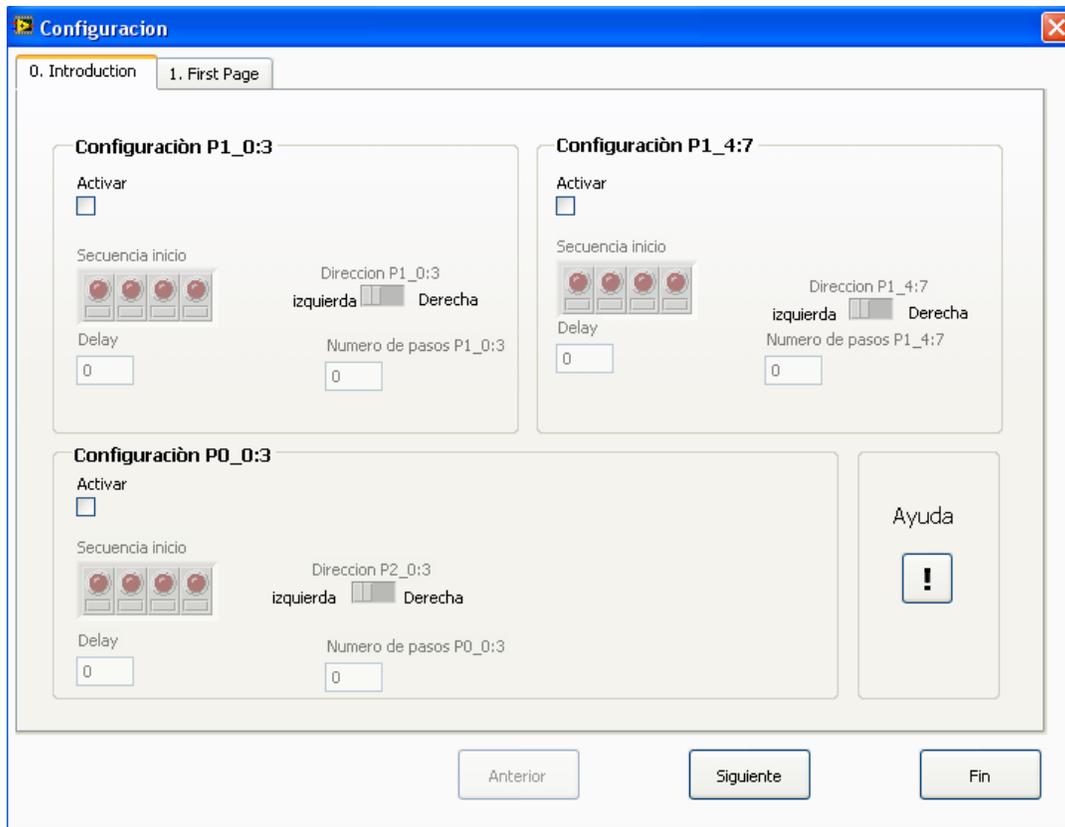


Figura 3.4. Interfaz de la aplicación para la caracterización del posicionador.

El funcionamiento básico de la aplicación reside en la forma en la que controlamos los motores, en la figura 3.5 podemos observar un diagrama de flujo del funcionamiento de los motores en la aplicación.

- Establecemos el retardo entre pasos.
- Establecemos la secuencia de inicio.
- Establecemos el número de pasos que vamos a avanzar.
- Avanzamos un paso, después activamos el retardo, avanzamos otro paso, activamos el retardo, avanzamos otro paso, así sucesivamente, el número de pasos avanzados lo comparamos con el valor dado al principio (el número de pasos que queríamos avanzar), cuando sean iguales detenemos el motor.

Se utilizó la secuencia tipo *wave drive* cuya secuencia se puede apreciar en el capítulo 1, además del funcionamiento de los motores se explica en el mismo capítulo, cabe recordar que los motores son del tipo imán permanente.

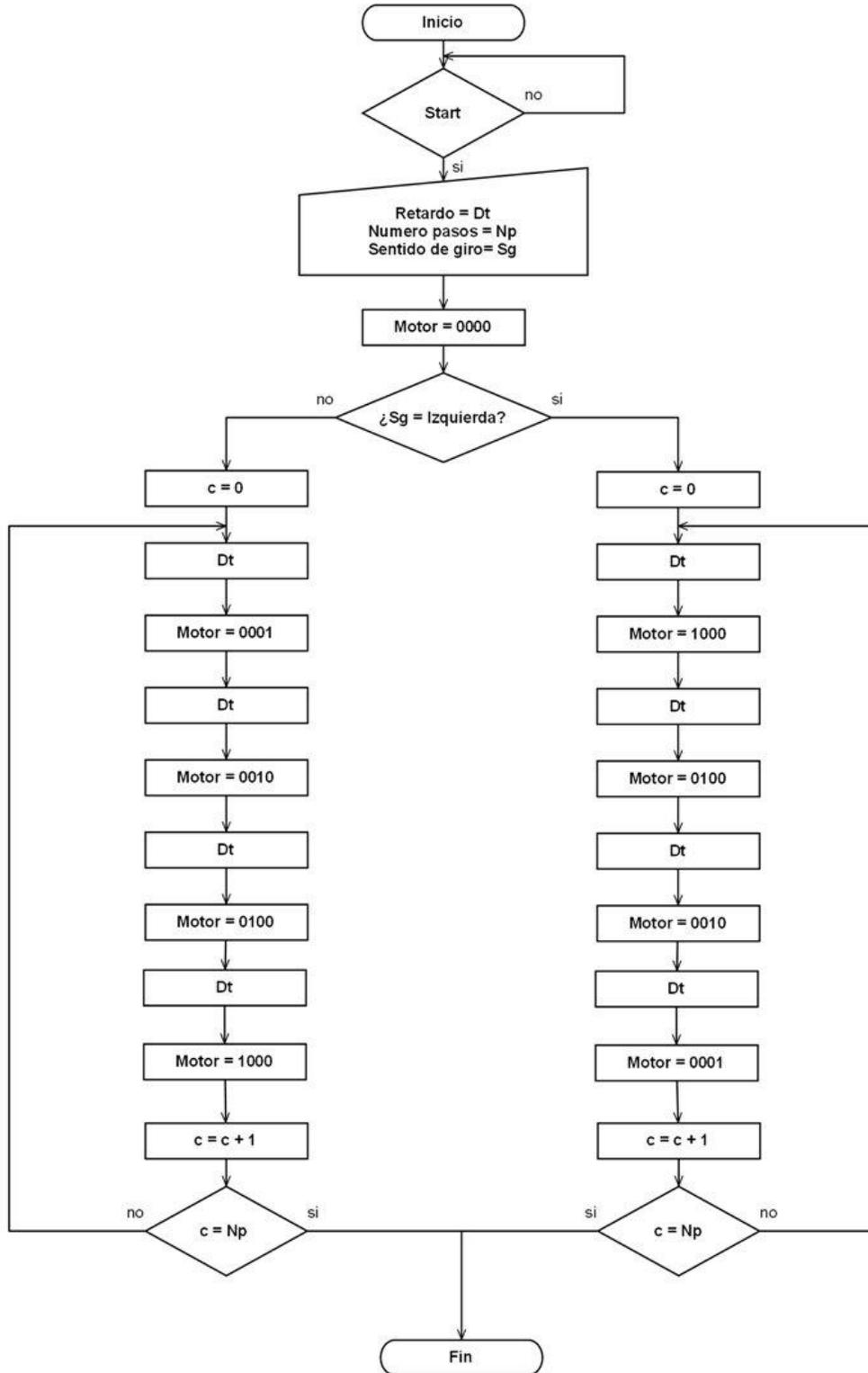


Figura 3.5. Diagrama de flujo del funcionamiento de los motores.

Una vez que se termino de programar la aplicación, Lo que se hizo después fue poner a barrer los ejes, esto con la finalidad de aflojar los tornillos para lograr un avance suave entre los ejes. Durante este proceso se presentó un problema que evitó que se hiciera una caracterización correcta. El instrumento no estaba bien calibrado, por lo que al hacer los barridos en los ejes estos tendían a quedar atorados en algún punto. En el laboratorio se intentó volver a calibrar los ejes, pero sin éxito, motivo por el cual el sistema se envió a revisar con las personas encargadas de su construcción. Una vez que se observó que no había dificultad en el barrido de cada eje, se pasó a caracterizar cada eje. Las pruebas hechas consistían en los siguientes pasos:

1. Hacer avances en cada eje de los pasos asignados
2. Medir con un vernier electrónico el avance en micrometros.
3. Regresar el número de pasos avanzados, es decir regresar al inicio
4. Volver a hacer el avance de pasos.
5. Volver a medir con el vernier el avance en micrometros.
6. Los datos recolectados se guardaron en una tabla.
7. Se obtuvo un promedio para calcular el avance por paso en unidades de longitud.

De la tarjeta de control únicamente se ocupan 12 canales, se optó por definir desde el principio los canales que se asignarían a los motores (tabla 3.2), sin que haya posteriormente una posibilidad de cambiarlos; esto se realizó para que el usuario no se esté preocupando por estar configurando las terminales que se conectaran a los motores y únicamente llegue y empiece a realizar las caracterizaciones.

**Tabla 3.2.** Asignación de terminales en la tarjeta NI USB-6501.

Terminal	Conexión
1	GND
17	Motor X
18	Motor X
19	Motor X
20	Motor X
27	Motor Y
28	Motor Y
29	Motor Y
30	Motor Y
3	Motor Z
4	Motor Z
5	Motor Z
6	Motor Z

### 3.4.1. Caracterización del eje X

Para la caracterización del eje X se hicieron barridos de 25, 50,100, 200, 400, 600, 800, 1000, 2000 pasos, los datos obtenidos se pueden observar en la tabla 3.3.

**Tabla 3.3.** Caracterización del eje X.

Pasos	Posición de inicio (mm)	Posición final (mm)	Desplazamiento total (mm)	Desplazamiento por paso (mm)
25	107.54	107.66	0.12	0.0048
50	107.54	107.77	0.23	0.0046
100	107.54	108.28	0.74	0.0074
200	107.54	109.1	1.56	0.0078
400	107.54	110.67	3.13	0.0078
600	107.54	112.19	4.65	0.0077
800	107.54	113.78	6.24	0.0078
1000	107.54	115.31	7.77	0.0077
2000	107.54	123.3	15.76	0.0078
			Promedio	0.0070

### 3.4.2. Caracterización del eje Y

Al igual que el eje X, para la caracterización del eje Y se hicieron barridos de 25, 50,100, 200, 400, 600, 800, 1000, 2000 pasos, los datos obtenidos se pueden observar en la tabla 3.4.

**Tabla 3.4.** Caracterización del eje Y.

Pasos	Posición de inicio (mm)	Posición final (mm)	Desplazamiento total (mm)	Desplazamiento por paso (mm)
25	49.04	49.19	0.15	0.0060
50	49.04	49.36	0.32	0.0064
100	49.04	49.72	0.68	0.0068
200	49.04	50.72	1.68	0.0084
400	49.04	52.32	3.28	0.0082
600	49.04	53.91	4.87	0.0081
800	49.04	55.36	6.32	0.0079
1000	49.04	57.14	8.1	0.0081
2000	49.04	64.84	15.8	0.0079
			Promedio	0.0075

### 3.4.3. Caracterización del eje Z

Al igual que el eje X, Y, para la caracterización del eje Z se hicieron barridos de 25, 50, 100, 200, 400, 600, 800, 1000, 2000 pasos, los datos obtenidos se pueden observar en la tabla 3.5.

**Tabla 3.5.** Caracterización del eje Z.

Pasos	Posición de inicio (mm)	Posición final (mm)	Desplazamiento total (mm)	Desplazamiento por paso (mm)
25	159.04	159.17	0.13	0.0052
50	159.04	159.32	0.28	0.0056
100	159.04	159.71	0.67	0.0067
200	159.04	160.6	1.56	0.0078
400	159.04	162.31	3.27	0.0081
600	159.04	163.89	4.85	0.0080
800	159.04	165.34	6.3	0.0078
1000	159.04	167.13	8.09	0.0080
2000	159.04	174.83	15.79	0.0078
			Promedio	0.0072

### 3.4.4. Resultados en la caracterización del equipo

De acuerdo a los datos mostrados en las tablas anteriores se llegó a la siguiente conclusión (tabla 3.6).

**Tabla 3.6** Características del posicionador XYZ.

Dimensiones de la estructura: 90 x 90 x 90 [cm]  
 Barrido máximo: eje X: 66 [cm], eje Y: 66 [cm], eje Z: 36.5 [cm]  
 Barrido efectivo: eje X: 7cm–60 cm, (real 6.6 cm–59.4 cm), eje Y: 7cm–60 cm, (real 6.6 cm–59.4 cm), eje Z: 7cm – 36 cm, (real 7.3 cm–36.5 cm).  
 Desplazamiento por paso en el eje X: 8  $\mu$ m.  
 Desplazamiento por paso en el eje Y: 8  $\mu$ m.  
 Desplazamiento por paso en el eje Z: 8  $\mu$ m.  
 Retardo entre cada paso: 20 ms.



**Nota:** Se decidió establecer el desplazamiento por paso en 8  $\mu$ m, ya que las personas encargadas de la construcción del posicionador dieron este valor como teórico, y los datos recolectados se aproximan a este valor. Para fijar el desplazamiento efectivo se tomó el 10 % de cada lado en los ejes X,Y, para el eje Z la mediciones se toman a partir del 20%, este valor obtenido se redondeó por facilidad a la hora de tomar las medidas, se dejó este espacio como medida de seguridad para que los motores no lleguen a tocar el inicio de carrera.

Ya que teníamos calibrado el equipo, procedimos a realizar el VI encargado del control del osciloscopio.

### 3.5. Osciloscopio

El osciloscopio empleado para realizar la tarea de la adquisición de datos fue un osciloscopio de fósforo digital marca Tektronix DPO 3014, cuyas características principales se muestran en la tabla 3.7, en el anexo B podemos ver las características completas del dispositivo..

**Tabla 3.7.** Características principales del osciloscopio Tektronix DPO 3014.

4 Canales de entrada

Ancho de banda analógico de 100MHz

Frecuencia de muestreo de 2.5 GS/s

Comunicación RS-232,USB,Ethernet

Captura máxima de 50,000 ondas/s

Grabado máximo de datos de 5 Mega muestras por cada canal



#### 3.5.1. Programación del osciloscopio

El osciloscopio utilizado admite el control remoto desde la PC, ya sea vía Ethernet, GPIB o USB. Para la transmisión de datos entre el osciloscopio y la PC se optó por utilizar el puerto USB (figura 3.6).



*Figura 3.6.* Conexión mediante bus USB.

En los siguientes párrafos se explica de forma general la manera en la que se mandan los comandos de control al osciloscopio, éstos comandos funcionan para toda la serie de osciloscopio de la marca Tektronix DPO 3000. Dependiendo del modelo, al conectar el osciloscopio a la PC, éste nos lo detecta como un dispositivo USB con alguno de los parámetros de la tabla 3.8.

**Tabla 3.8.** Parámetros de conexión.

Parámetro	Valor
ID del Fabricante	0x0699 (decimal 1689)
ID del Producto	0x0410 (decimal 1040) DPO3012
	0x0411 (decimal 1041) DPO3014
	0x0412 (decimal 1042) DPO3032
	0x0413 (decimal 1043) DPO3034
	0x0414 (decimal 1044) DPO3052
	0x0415 (decimal 1045) DPO3054
Número de serie	Número de serie
Descripción del fabricante	"Tektronix"
Descripción de la interface	"USBTMC-USB488"

El control de las operaciones y funciones a través del puerto USB se realiza por medio de comandos y consultas (*queries*). Los comandos y *queries* que se mostrarán a continuación utilizan la notación de Backus-Naur<sup>1</sup>, los símbolos usados se describen en la tabla 3.9.

**Tabla 3.9.** Símbolos de funciones y operaciones para los puertos.

Símbolo	Significado
<>	Define elemento
=	Es definido como
	OR exclusiva
{ }	Grupo; Un elemento es requerido
[ ]	Opcional; Puede ser omitido
...	Elemento anterior, puede ser repetido

<sup>1</sup> Es una metasintaxis usada para expresar gramáticas libres de contexto: es decir, una manera formal de describir lenguajes formales.

**Sintaxis de Comandos.** Los comandos consisten en un *set* de órdenes e instrucciones de consulta (generalmente llamados comandos y consultas). Los comandos modifican la configuración del osciloscopio o le dicen al osciloscopio que realice una acción específica. Las consultas regresan información acerca del osciloscopio.

**Mensajes:** Un mensaje es un comando o nombre de consulta seguido por cualquier información que el osciloscopio necesita para ejecutar otro comando o consulta. Los mensajes pueden contener cinco tipos de elementos, definidos en la tabla 3.10.

**Tabla 3.10.** Elementos de comando.

Símbolo	Significado
<Header>	Es el nombre del comando. Si el encabezado termina como una pregunta (?), entonces se trata de un <i>query</i> . El encabezado puede empezar con el símbolo “:”. Si el comando está concatenado con otros comandos entonces es necesario incluir el símbolo “:”.
<Mnemonic>	Es la subfunción del encabezado. Algunos comandos sólo tienen un mnemónico. Si el comando tiene más de un mnemónico, entonces es necesario separarlos con el símbolo “:”.
<Argument>	Es una propiedad del comando asociada con el encabezado. Algunos comandos no tienen argumentos y otros tienen múltiples argumentos, un espacio separa el argumento del encabezado y una <coma> separa los argumentos entre sí.
<comma>	Una sola coma es usada para separar los argumentos.
<space>	Un espacio en blanco es usado para separar el argumento del encabezado.

**Comandos:** Los comandos causan que el osciloscopio realice una función específica. La estructura básica de los comandos es la siguiente:

[:]<Header>[<Space><Argument>[<Comma><Argument>] ...]

El encabezado de un comando consiste en uno o más mnemónicos dispuestos en una estructura jerárquica o de árbol. El primer mnemónico es la base o raíz del árbol y cada mnemónico siguiente

tiene una jerarquía más baja que el anterior mnemónico. Los comandos que se encuentren en un nivel superior de la jerarquía afectan a niveles inferiores. El signo (:) siempre regresa a la base del árbol de comandos, como se muestra en el siguiente ejemplo.

[:]<Header>[<Space><Argument>[<Comma><Argument>][:]<Header>[<Space><Argument>  
>[<Comma><Argument>] ...]

**Consultas:** Las consultas causan que el osciloscopio regrese información acerca de su estado. Las consultas tienen la siguiente estructura:

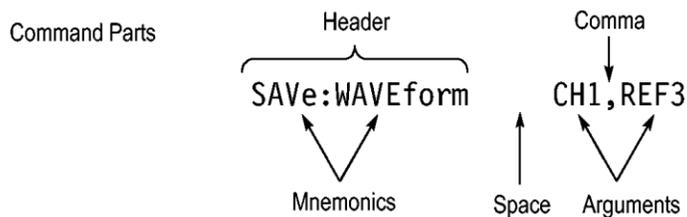
- [:]<Header>
- [:]<Header>[<Space><Argument>[<Comma><Argument>]...]

Puede especificar un comando de consulta en cualquier nivel dentro del árbol de comandos a menos que se indique lo contrario. Estas ramas de consultas devuelven información acerca de todos los mnemónicos debajo de la rama determinada o nivel.

**Encabezados (Headers):** El comando *HEADer* es utilizado cuando queremos que el osciloscopio nos regrese un encabezado como parte de la respuesta a la consulta hecha. Si está en “on” el Header, entonces la respuesta de la consulta regresa los encabezados de los comandos. Cuando está en “off” el Header, la respuesta sólo incluye los valores. Esto hace que sea más fácil extraer los valores, para su posterior análisis. En la tabla 3.11 se muestra la diferencia en las respuestas.

**Tabla 3.11.** Diferencias entre los distintos encabezados de control.

Consulta	Header Off	Header On
TIME?	14:30:00	:TIME "14:30:00"
ACQUIRE:NUMAVG?	100	:ACQUIRE:NUMAVG 100



**Puesta a ceros el buffer de datos:** Se puede limpiar y reiniciar (reset) la cola de salida del *buffer* de datos del osciloscopio para recibir nuevas consultas y comandos, esto es usando la función *DLC* (Device Clear), más adelante se mostrara como usar este comando.

**Entrada de comandos:** Las siguientes reglas aplican al ingresar comandos:

- Se pueden introducir comandos en mayúsculas y minúsculas.
- Cualquier comando puede ir precedido de espacios en blanco. Los espacios en blanco incluyen cualquier combinación de los caracteres de control ASCII 00 a 09 y 0B a 20 hexadecimal (0 a 9 y 11 a 32 en decimal).
- El osciloscopio ignora comandos que consisten en cualquier combinación de espacios en blanco y avance de línea.

**Abreviación:** Se pueden abreviar la mayor parte de los comandos.

**Concatenación:** Se puede concatenar cualquier combinación de comandos y consultas mediante un punto y coma (;). El osciloscopio ejecuta los comandos en el orden recibido. Para concatenar los comandos se deben seguir las siguientes reglas:

- Se separan completamente los comandos por un punto y coma, además del signo correspondiente al encabezado del comando que sigue, excepto el primer encabezado. Por ejemplo, el comando *TRIGger:MODE NORMAL* y *ACQuire:NUMAVg 8*, puede ser concatenado como sigue:

*TRIGger:MODE NORMAL;:ACQuire:NUMAVg 8*

- Si se concatenan comandos donde sólo difieren en el último mnemónico, se puede abreviar el segundo comando y eliminar su correspondiente signo(:). Por ejemplo, para concatenar los comandos *ACQuire:MODE ENvelope* y *ACQuire:NUMAVg 8* en una sola línea:

*ACQuire:MODE ENvelope; NUMAVg 8*

- Nunca preceder un asterisco con dos puntos.
- Cuando se concatenan consultas, sus respuestas regresan en una sola línea.
- Un set de comandos y consultas pueden ser concatenadas en el mismo mensaje. Estos comandos y consultas son ejecutados en el orden recibido. Por ejemplo:

*ACQuire:MODE SAMple;NUMAVg?;STATE*

**Terminación:** Se ocupa el símbolo *EOM* (End Of Message) para representar la terminación de un mensaje.

**Tipos de argumentos:** Los argumentos son los valores que le pasamos a las consultas o los comandos. Los comandos usan argumentos tales como enumeración, número, cadenas de caracteres y bloques. A continuación se explica cada una:

**Enumeración:** Estos argumentos se introducen como una cola de caracteres. Como palabras reservadas.

**Número:** Varios comandos requieren argumentos numéricos. La sintaxis muestra el formato que el osciloscopio regresa en respuesta a una consulta.

**String:** Algunos comandos aceptan como argumento una cadena de caracteres, esta cadena de caracteres se pone en su forma ASCII, cuando el comando así lo requiera.

Para ver la lista completa de comandos (mnemónicos, consultas y comandos) se debe consultar la guía de programación DPO 3000 Tektronix, la cual se encuentra en la página del fabricante. Hasta aquí se explica la forma en la que los comandos son mandados desde y hacia la PC u osciloscopio. Estos comandos pueden ser implementados en algún lenguaje de programación, por ejemplo C/C++, o en algún lenguaje gráfico que facilite la programación, por ejemplo LabVIEW.

### 3.5.2. Programación del Osciloscopio en LabVIEW

El concepto de instrumentación virtual nace a partir del uso de la computadora personal como instrumento de medición de señales, tales como presión, temperatura, caudal, o cualquier fenómeno físico medible. Pero un instrumento virtual no sólo se encarga de hacer mediciones, sino 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. Por ejemplo, el osciloscopio tradicional tiene una funcionalidad ya predefinida desde la fábrica donde lo diseñan, producen y ensamblan. Es decir, la funcionalidad de este tipo de instrumento es definida por el fabricante del equipo, y no por el usuario mismo. El término "virtual" nace precisamente a partir del hecho de que cuando se utiliza la PC como "instrumento" es el usuario mismo quién, a través del software, define su funcionalidad y "apariencia" y por ello decimos que "virtualizamos" el instrumento, ya que su funcionalidad puede ser definida una y otra vez por el usuario y no por el fabricante. Por lo tanto el instrumento virtual a desarrollar debe cumplir básicamente con las siguientes características:

- Control total desde la PC, es decir, que no haya necesidad de estar manejando algún dispositivo físico.
- Capacidad de desplegar los datos adquiridos.
- Capacidad para almacenar los datos adquiridos en la PC.
- Fácil manejo.

Cada instrumento programable de prueba y medición entiende un conjunto de comandos. En este caso los comandos que le enviaremos al osciloscopio son los vistos anteriormente. Se optó por desarrollar la aplicación de adquisición de datos en LabVIEW, esto debido a la facilidad de programación que nos ofrece. La comunicación del osciloscopio con LabVIEW la hacemos por medio del puerto USB. LabVIEW cuenta con una API de comunicación llamada VISA (Virtual Instrument Software Architecture), la cual ocuparemos para mandar y recibir datos de nuestro dispositivo (figura 3.7).



Figura 3.7. Comunicación a través de NI-VISA.

**VISA:** Es una API o librería desarrollada por varios fabricantes de equipos que proporciona un software estándar para las operaciones de lectura y escritura en instrumentación. NI-VISA es la implementación de National Instruments de este estándar, puede establecer comunicaciones de GPIB, serie, PXI, VXI, Ethernet u otro bus. Como VISA soporta varias interfaces de comunicación suele ser el método elegido para crear *drivers* de control de equipos. VISA por sí solo no provee capacidad de programación. VISA es un API de alto nivel que llama a las funciones de los *drivers* que se encuentran en un nivel más bajo de abstracción. [14,15]

Una de las ventajas de VISA es que utiliza muchas de las mismas operaciones para comunicarse con los instrumentos, independientemente del tipo de interfaz (bus de comunicación). Por ejemplo, el comando de VISA para escribir una cadena ASCII a un instrumento que se basa en mensajes es el mismo si el instrumento es de serie, GPIB o VXI. Por lo tanto, VISA proporciona independencia de la interfaz. Esto puede hacer que sea fácil cambiar de interfaz y además da a los programadores facilidad de programar para diferentes instrumentos con un solo lenguaje que aprender. VISA también está diseñada para que los programas escritos usando las llamadas a funciones VISA sean fácilmente transferibles de una plataforma a otra. Para asegurar esto, VISA

define estrictamente sus propios tipos de datos tal que las cuestiones como el tamaño de una variable entera de una plataforma a otra no debe afectar a un programa de VISA. Las llamadas a funciones VISA y sus parámetros asociados son uniformes en todas las plataformas; el software puede ser portado a otras plataformas y ser recompilado. En otras palabras, un programa en C usando VISA puede ser portado a otras plataformas que soporten C. Un programa de LabVIEW puede ser portado a otras plataformas con soporte de LabVIEW. Una de las ventajas que ofrece LabVIEW es que hace el API de VISA totalmente transparente al usuario. La mayor ventaja de VISA, quizás, es que es un lenguaje fácil de aprender y usar. Su estructura orientada a objetos hace que el lenguaje y sus operaciones sean muy intuitivos para aprender. Esto se debe en parte al hecho de que VISA provee la independencia de la interfaz mediante el uso de las mismas operaciones para interfaces diferentes y también por el hecho de que VISA presenta una API fácil de usar. VISA proporciona la funcionalidad más utilizada para la programación en instrumentación en un conjunto de comandos muy compacto. Varios instrumentos de medición y control necesitan un *driver* adicional para ocupar todas sus funcionalidades con VISA. LabVIEW implementa un método fácil para la descarga de *drivers*, basta únicamente con conectar el dispositivo, ejecutar LabVIEW, ir a la sección de descarga de *drivers*, automáticamente nos busca e instala el *driver* del dispositivo conectado. Si no existe el dispositivo, se tiene que crear manualmente, ésto creando VIs, apoyándonos completamente en VISA (figura 3.8).

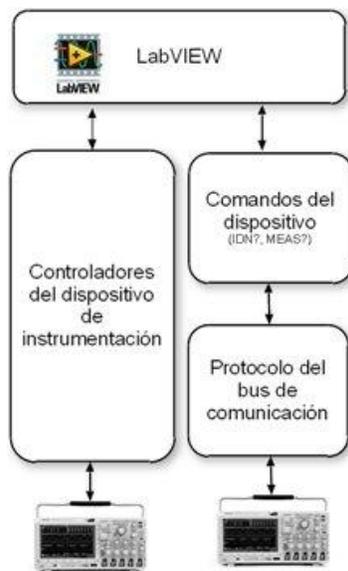


Figura 3.8. Comunicación de dispositivos físicos mediante LabVIEW.

Con respecto a LabVIEW hay docenas de VIs dedicados a VISA, los más importantes son VISA Open, VISA Write, VISA Read Y VISA Close. En la siguiente tabla 3.12 se listan la utilidad de cada uno.

**Tabla 3.12.** Principales funciones de NI-VISA.

VI	Función
 VISA open	Abre una sesión de VISA , hay que indicarle la interfaz que se va a ocupar.
 VISA Close	Cierra una sesión de VISA.
 VISA Write	Escribe datos desde el buffer hasta el otro dispositivo.
 VISA Read	Hay que indicarle el número de bytes que debe leer del buffer.
 VISA Clear	Limpia el Buffer, tanto de entrada como de salida.

En la figura 3.9 podemos observar la forma en la que llamamos un dispositivo desde LabVIEW, en este caso el osciloscopio ocupado.

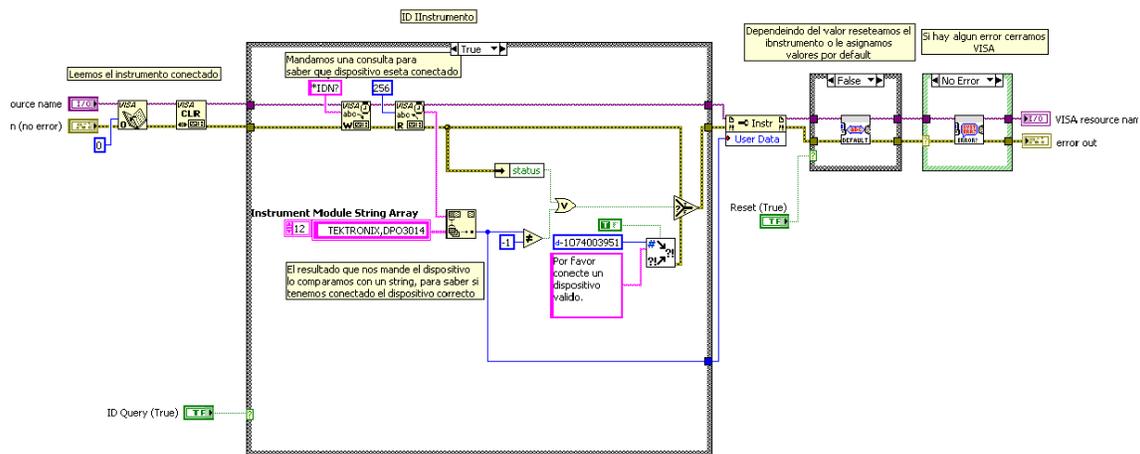


Figura 3.9. Ejemplo para lectura de un instrumento en LabVIEW.

Con base en la figura primero con *VISA open* leemos el dispositivo que esté conectado en ese momento, procedemos a limpiar el buffer de entrada y salida (*VISA clear*), Mandamos una consulta (*IDN?*) a nuestro dispositivo (*VISA Write*), recibimos la respuesta (*VISA Read*), la respuesta recibida la comparamos con una tabla de dispositivos, si está conectado el dispositivo correcto no mandamos ningún error, en caso contrario se manda un mensaje de error. Si hay un error cerramos la sesión de VISA (*VISA Close*), en caso contrario seguimos con la sesión de VISA. Todas las

consultas y comandos que mandemos desde VISA al osciloscopio son en su forma *Backus-Naur*, esto nos simplifica la programación del osciloscopio, usando VISA se hace más transparente su programación.

### 3.6. Diseño de la Interfaz

Ya que tenemos programado el VI de control de los motores y el VI de control del osciloscopio pasamos a desarrollar la interfaz. La interfaz para el manejo y control del posicionador, también fue desarrollado bajo el entorno de LabVIEW 8.5. Tomando en cuenta los requerimientos iniciales, se decidió programar el software de manera que tuviera una estructura de fácil comprensión para el usuario final, por lo que su apariencia sería similar a una clásica ventana de Windows OS. Únicamente se explica la estructura de la interfaz, para ver parte del código pasar al anexo A.

#### 3.6.1. Estructura del panel principal

En la figura 3.10 se muestra la ventana principal con la que el usuario interactúa al iniciar el programa. Las partes con las cuales cuenta dicha ventana son:

- Barra de menús: Donde están contenidas las diferentes acciones a elegir para que el software realice determinada tarea. Los menús con los que cuenta son: *Archivo*, *Adquisición*, *Ventanas*, *Acerca de*.
- Panel Dispositivos: Muestra los dispositivos conectados, como son, el dispositivo para realizar el control del posicionador y el dispositivo para realizar las adquisiciones.
- Panel Posición actual: Muestra la posición de barrido cuando está en modo automático.
- Panel Control manual: Cuando se está manipulando el posicionador manualmente, se nos muestra una ventana con el valor de avance que se esté haciendo.
- Ventana de Mensajes: Es un indicador de que nos muestra si ocurre algún error durante la adquisición o algún otro evento (al cambiar la configuración de barrido, cancelar la adquisición de datos, etc).
- Panel Gráfica: Panel donde se muestra la gráfica de los datos adquiridos por el osciloscopio.

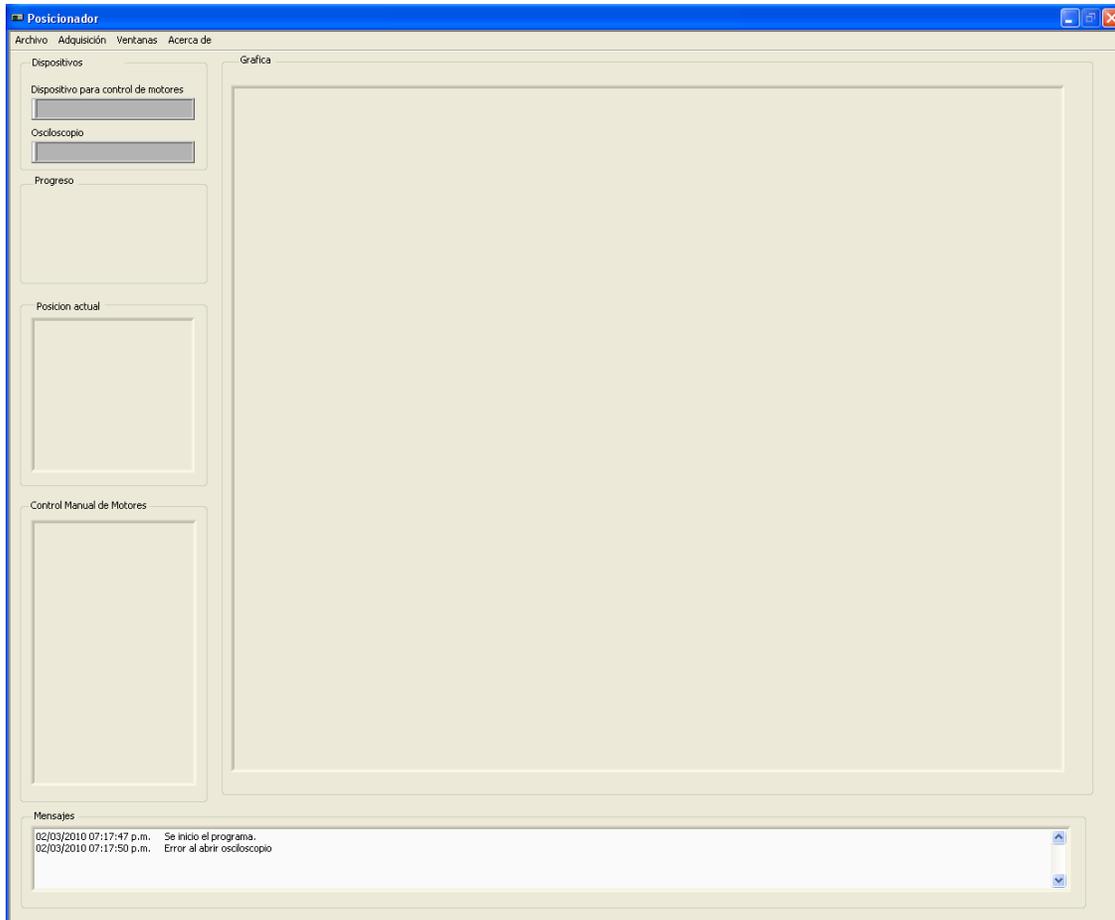


Figura 3.10. Panel principal del software.

### 3.6.2. Menú Archivo

Al ejecutar el programa lo primero que debe hacer el usuario es definir ciertos parámetros para que se realice una adquisición correcta. El menú archivo cuenta con las opciones de: guardar y configuración, Los cuadros de diálogo se muestran en la figura 3.11.

**Configuración:** Nos permite configurar ciertas características del software, tales como la ubicación del directorio donde se guardaran los datos recolectados en las adquisiciones automáticas. Además de cambiar el dispositivo con el cual se efectuaran las adquisiciones. En la pestaña datos aparecen tres opciones para configurar, Cambiar, nos permite activar las casillas que se encuentran debajo de estas. Nombre de la carpeta, definimos el nombre de la carpeta donde se guardaran los datos. Directorio raíz, Es la ruta donde se creará la carpeta donde se guardarán los datos. La pestaña Dispositivos, contiene dos elementos para configurar. Cambiar, la cual nos permite activar la casilla que se encuentra debajo de esta. Osciloscopio, en caso de que no se

haya conectado el osciloscopio al iniciar el programa, desde aquí podemos seleccionarlo una vez esté conectado. Además se cuenta con la posibilidad de cambiar el sentido de los ejes.

**Guardar:** Sirve para guardar las adquisiciones que se estén haciendo cuando se está manejando de forma manual el equipo.

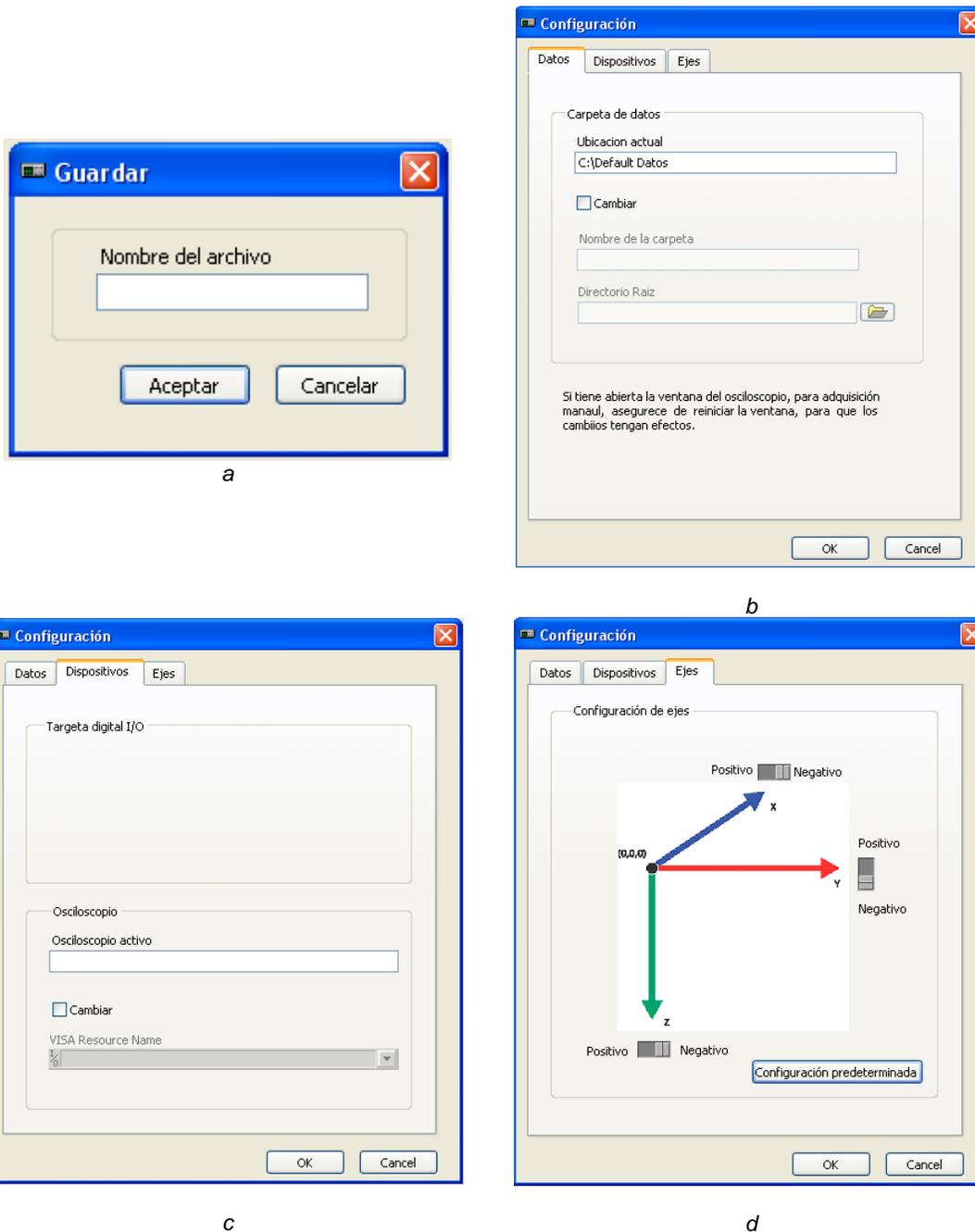
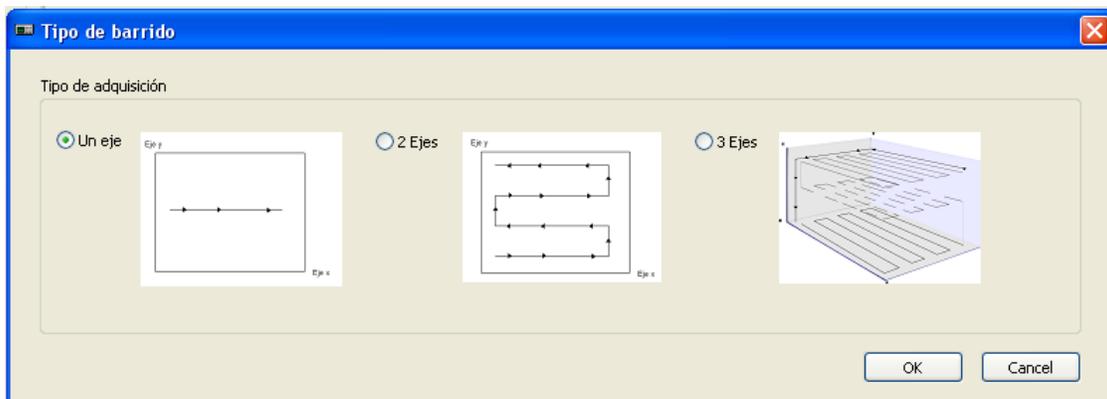


Figura 3.11. Paneles de control, a) Guardar, b) Configuración para el guardado de datos, c) Selección del dispositivo de adquisición, d) Configuración de los ejes.

Una vez que se hayan hecho las correspondientes configuraciones del software el usuario podrá aceptar o cancelar los cambios hechos. En el anexo A se explica el funcionamiento del código para ambas opciones de configuración.

### 3.6.3. Menú Adquisición

Mediante el menú adquisición definimos los parámetros correspondientes a la adquisición de datos, éste es, el tipo de adquisición, si es en uno, dos o tres ejes, además de iniciar la prueba correspondiente. La configuración para las adquisiciones se hace mediante un cuadro de diálogo, como se puede ver en la figura 3.12.



3.12. Cuadro de diálogo que nos permite seleccionar la manera en la que se barren los ejes.

Una vez que tenemos seleccionado el tipo de barrido que vamos a hacer y tras haber aceptado la opción, se nos desplegará otro cuadro de diálogo. El cuadro de diálogo será diferente dependiendo del tipo de barrido seleccionado. En el caso de que se haya seleccionado la primera opción (un eje), en la primer pantalla que nos aparece, tenemos la posibilidad de seleccionar el eje en el cual se efectuara el barrido, después seleccionaremos la distancia entre muestras, ya sea en micras o en milímetros y el número de muestras que se tomaran, En la última pantalla configuramos diferentes aspectos del DAQ (osciloscopio), tales como el canal donde se hará la adquisición, el tiempo total de la señal en segundos, así como el número de puntos que tomara por señal para ser almacenados. En el caso de una adquisición en dos ejes, los pasos a seguir son los mismos que para un eje, con la única diferencia que podemos seleccionar un máximo de dos ejes, la configuración es independiente por cada eje seleccionado. Para la configuración de tres ejes tenemos que configurar cada eje, la configuración por eje es independiente entre sí. Para la configuración del DAQ las diferentes opciones a configurar son las mismas que en los otros barridos. En las figuras 3.13a y 3.13b se muestran los cuadros de diálogo para los barridos. Se

decidió implementar la configuración de esta forma, ya que así se hace más simple la configuración de las adquisiciones para el usuario, haciendo más transparente su manejo. Una vez que ya se haya terminado de configurar el sistema podemos ya iniciar la adquisición. Si deseamos parar la adquisición en curso, en el mismo menú se encuentra la opción de detener, también podemos detener la adquisición presionando la tecla “ESC” desde el teclado.

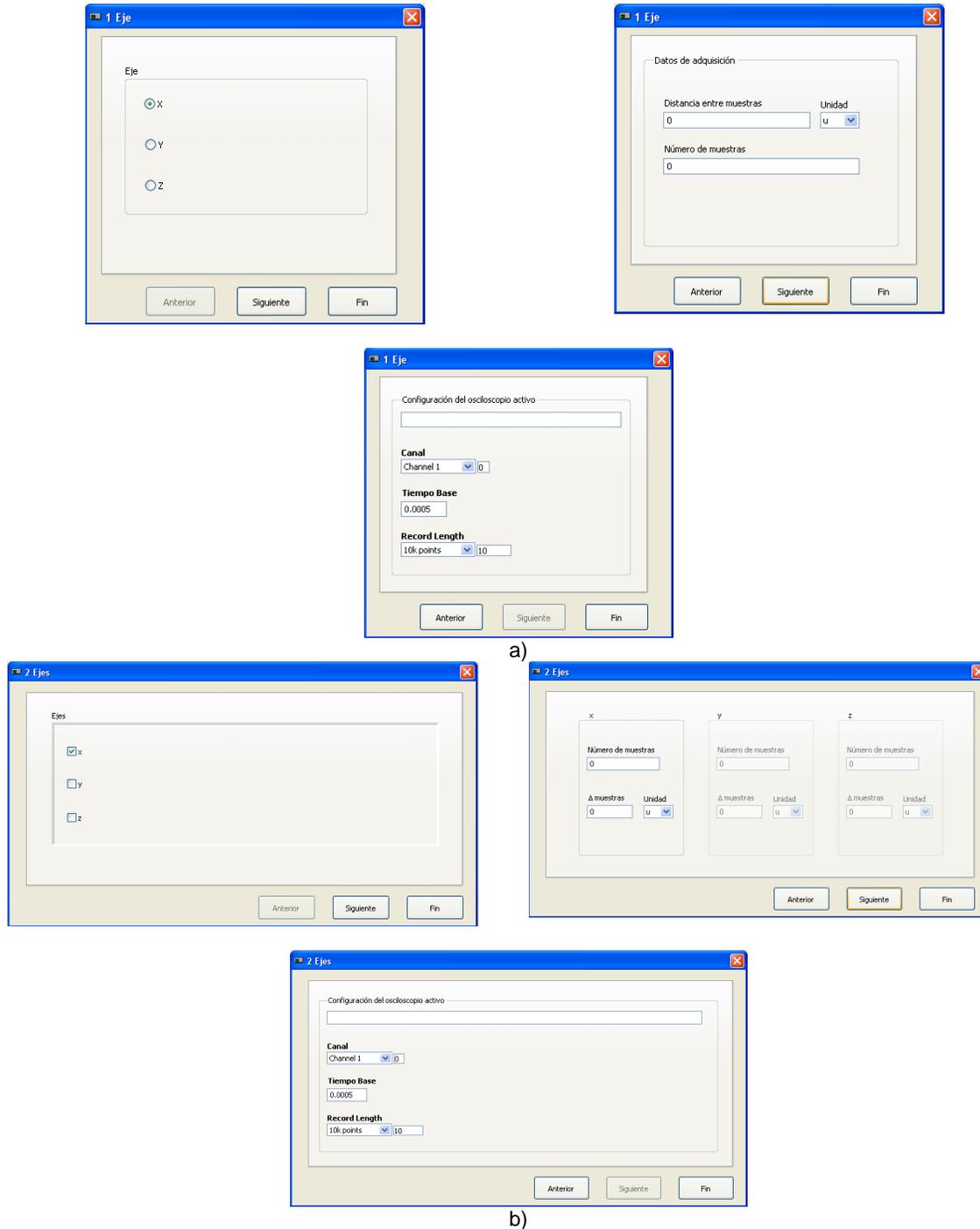


Figura 3.13a. Cuadros de dialogo para configurar las adquisiciones, a) Un eje, b) dos ejes.

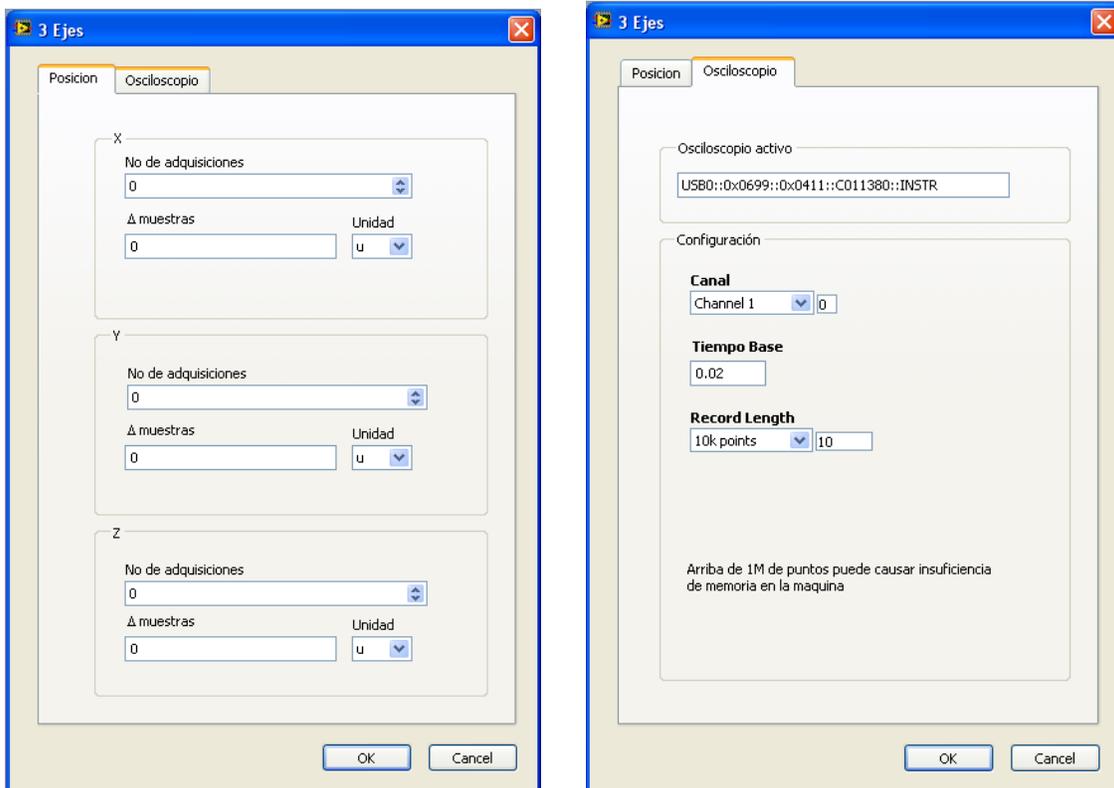


Figura 3.13b. Cuadros de dialogo para configurar las adquisiciones en tres ejes.

### 3.6.4. Menú Ventanas

Este menú cuenta con tres opciones, Osciloscopio, Motores y Posición, la función de cada categoría es:

- Osciloscopio, desde aquí activamos el panel del osciloscopio, al seleccionar este ítem se nos muestran dos opciones, mostrar el panel con o sin controles. Si seleccionamos el panel sin controles, el manejo del osciloscopio se hace desde el dispositivo físico, si seleccionamos la opción con controles, el manejo del osciloscopio se hace desde la misma interfaz.
- Motores, nos permite controlar el posicionador desde el teclado. Son las teclas de dirección las que se ocupan para hacer el barrido.
- Posición, únicamente se nos muestra la posición, esto, cuando la adquisición se hace de forma manual.

### 3.7. Distribución del programa

Una vez que se finalizó el desarrollo, se debe entregar el software terminado e independiente en la forma de un ejecutable o DLL. El Constructor de Aplicaciones de LabVIEW ofrece la funcionalidad para elaborar ejecutables, DLLs, distribuciones de código fuente y archivos zip. También puede utilizarlo para elaborar instaladores de Windows que le brindan al usuario una experiencia muy directa durante la instalación. Los instaladores juntan los ejecutables y los DLLs con los controladores del hardware de NI, documentación, licencias y archivos configurables en un solo paquete. Fue entregado un ejecutable.

# Pruebas, Resultados y Conclusiones

## 4.1. Introducción

Una vez que se concluyó con el desarrollo del software pasamos a realizar diferentes pruebas de funcionamiento, con la finalidad de validar la funcionabilidad de la interfaz. Las pruebas que se le hicieron al sistema fueron 2.

- Pruebas mecánicas
- Pruebas acústicas

## 4.2. Pruebas mecánicas

La repetibilidad es la capacidad de reproducción de la pluma o del índice del instrumento al medir repetidamente valores idénticos de la variable en las mismas condiciones de servicio y en el mismo sentido de variación. Con esta prueba lo que se busca es saber que tan confiable es el sistema para obtener los mismos valores en pruebas idénticas. La prueba consistió en hacer barridos a determinadas distancias en los 3 ejes, de acuerdo a estándares internacionales, partiendo de un punto inicial y observar la variación en el desplazamiento para regresar a su posición original. En la tabla 4.1 se muestran los datos obtenidos para el eje X. Para efectos de

presentación, por utilidad se presentan solo “x” número de datos.

**Tabla 4.1.** Datos obtenidos en el eje X.

100 pasos			
Posición inicial (mm)	Posición Final (mm)	Delta (mm)	Avance por paso (mm)
85	85.68	85	0.0068
85	85.6	85	0.006
85	85.66	85	0.0066
85	85.66	85	0.0066
85	85.66	85	0.0066
85	85.66	85	0.0066
85	85.66	85	0.0066
85	85.66	85	0.0066
85	85.65	85	0.0065
85	85.66	85	0.0066
85	85.66	85	0.0066
85	85.66	85	0.0066
85	85.66	85	0.0066
84.84	85.51	84.84	0.0067
84.84	85.51	84.84	0.0067
84.84	85.51	84.84	0.0067
84.84	85.5	84.84	0.0066
84.84	85.51	84.84	0.0067
84.84	85.48	84.83	0.0064
84.83	85.48	84.83	0.0065
84.83	85.48	84.83	0.0065
84.83	85.48	84.83	0.0065
84.83	85.49	84.83	0.0066
84.83	85.5	84.84	0.0067
84.84	85.49	84.84	0.0065
84.84	85.5	84.84	0.0066
84.84	85.5	84.84	0.0066
84.84	85.49	84.84	0.0065
84.84	85.49	84.82	0.0065
84.82	85.48	84.82	0.0066
84.82	85.48	84.82	0.0066
	Desviación Estándar		0.000136289
	Varianza		1.85747E-08
	Promedio		0.006573333

200 pasos

Posición inicial (mm)	Posición Final (mm)	Delta (mm)	Avance por paso (mm)
84.82	86.12	84.71	0.0065
84.71	86.13	84.71	0.0071
84.71	86.08	84.65	0.00685
84.65	86	84.58	0.00675
84.58	85.96	84.58	0.0069
84.58	85.96	84.58	0.0069
84.58	85.96	84.58	0.0069
84.58	85.96	84.57	0.0069
84.57	85.95	84.57	0.0069
84.57	85.95	84.57	0.0069
Desviación Estándar			0.000152388
Varianza			2.32222E-08
Promedio			0.00686

En la tabla 4.2 se muestran los datos obtenidos para el eje Y.

**Tabla 4.2.** Datos obtenidos en el eje Y.

#### 20 pasos

Posición inicial (mm)	Posición Final (mm)	Delta (mm)	Avance por paso (mm)
18.47	18.59	18.47	0.006
18.47	18.59	18.47	0.006
18.47	18.6	18.47	0.0065
18.47	18.6	18.47	0.0065
18.47	18.6	18.47	0.0065
Desviación Estándar			0.00027386
Varianza			7.5E-08
Promedio			0.0063

#### 50 pasos

Posición inicial (mm)	Posición Final (mm)	Delta (mm)	Avance por paso (mm)
18.47	18.78	18.47	0.0062
18.47	18.83	18.47	0.0072
18.47	18.81	18.47	0.0068
18.47	18.81	11:16	0.0068
18.47	18.81	18.47	0.0068
Desviación Estándar			0.00035777
Varianza			1.28E-07
Promedio			0.00676

#### 100 pasos



Posición inicial (mm)	Posición Final (mm)	Delta (mm)	Avance por paso (mm)
18.4	19.96	18.4	0.0078
18.4	19.96	18.4	0.0078
18.4	19.96	18.4	0.0078
18.4	19.96	18.4	0.0078
18.4	19.96	18.4	0.0078
Desviación Estándar			9.6974E-19
Varianza			9.404E-37
Promedio			0.0078

En la tabla 4.3 se muestran los datos obtenidos para el eje Z.

**Tabla 4.3.** Datos obtenidos en el eje Z.

### 20 pasos

Posición inicial (mm)	Posición Final (mm)	Delta (mm)	Avance por paso (mm)
48.92	49.09	48.93	0.0085
48.93	49.07	48.93	0.007
48.93	49.08	48.93	0.0075
48.93	49.08	48.94	0.0075
48.94	49.08	48.93	0.007
Desviación Estándar			0.00061237
Varianza			3.75E-07
Promedio			0.0075

### 50 pasos

Posición inicial (mm)	Posición Final (mm)	Delta (mm)	Avance por paso (mm)
48.93	49.3	48.93	0.0074
48.93	49.38	48.94	0.009
48.94	49.38	48.93	0.0088
48.93	49.35	48.93	0.0084
48.93	49.32	48.93	0.0078
Desviación Estándar			0.00067231
Varianza			4.52E-07
Promedio			0.00828

### 100 pasos

Posición inicial (mm)	Posición Final (mm)	Delta (mm)	Avance por paso (mm)
-----------------------	---------------------	------------	----------------------

48.93	49.67	48.93	0.0074
48.93	49.68	48.93	0.0075
48.93	49.72	48.93	0.0079
48.93	49.7	48.93	0.0077
48.93	49.68	48.93	0.0075
48.93	49.69	48.93	0.0076
48.93	49.69	48.93	0.0076
48.93	49.69	48.93	0.0076
48.93	49.69	48.93	0.0076
48.93	49.69	48.93	0.0076
48.93	49.68	48.93	0.0075
48.93	49.69	48.93	0.0076
48.93	49.69	48.93	0.0076
48.93	49.69	48.93	0.0076
48.93	49.69	48.93	0.0076
48.93	49.69	48.92	0.0076
48.92	49.7	48.93	0.0078
48.93	49.68	48.93	0.0075
48.93	49.68	48.93	0.0075
48.93	49.68	48.93	0.0075
48.93	49.69	48.93	0.0076
48.93	49.69	48.93	0.0076
48.93	49.7	48.93	0.0077
48.93	49.69	48.93	0.0076
48.93	49.69	48.93	0.0076
48.93	49.7	48.94	0.0077
48.94	49.7	48.94	0.0076
48.94	49.7	48.93	0.0076
48.93	49.69	48.93	0.0076
48.93	49.69	48.93	0.0076
	Desviación Estándar		9.4686E-05
	Varianza		8.9655E-09
	Promedio		0.0076
200 pasos			
Posición inicial (mm)	Posición Final (mm)	Delta (mm)	Avance por paso (mm)

48.93	50.43	48.92	0.0075
48.92	50.45	48.92	0.00765
48.92	50.43	48.92	0.00755
48.92	50.43	48.92	0.00755
48.92	50.42	48.91	0.0075
Desviación Estándar			6.1237E-05
Varianza			3.75E-09
Promedio			0.00755

Los datos obtenidos son satisfactorios, ya que de acuerdo a los datos que observamos en las tablas anteriores podemos concluir que se cumple la repetibilidad del sistema, con una variación mínima en el proceso de desplazamiento.

### 4.3. Prueba Acústica

Una vez que se sabe que mecánicamente hay una repetibilidad, podemos garantizar que el sistema es confiable, desde el punto de vista mecánico, ahora hay que garantizar que hay una confiabilidad a la hora de adquirir los datos de los transductores, para esto se realizó una prueba acústica en el sistema. Esta prueba consiste en hacer los siguientes barridos:

- Eje X: El barrido se hace en un solo sentido, adquiriendo los datos en el origen, a los 10 pasos, 100 pasos y 250 pasos.
- Eje Y: Los barridos se hace en dos sentidos, primero se adquieren los datos del origen, luego se hace un barrido en el sentido negativo del eje de 10 pasos, en el sentido positivo se toman muestras a los 10, 20 y 30 pasos.
- Eje Z: Al igual que el eje y, se hacen barridos tanto en el sentido positivo del eje como en el negativo. En el sentido negativo se toman las muestras a los 10 pasos. En el sentido positivo las muestras son tomadas a los 10 y 20 pasos.

En la figura 4.1 se muestra los avances en pasos para tomar las adquisiciones de datos y de esta forma validar la prueba acústica.

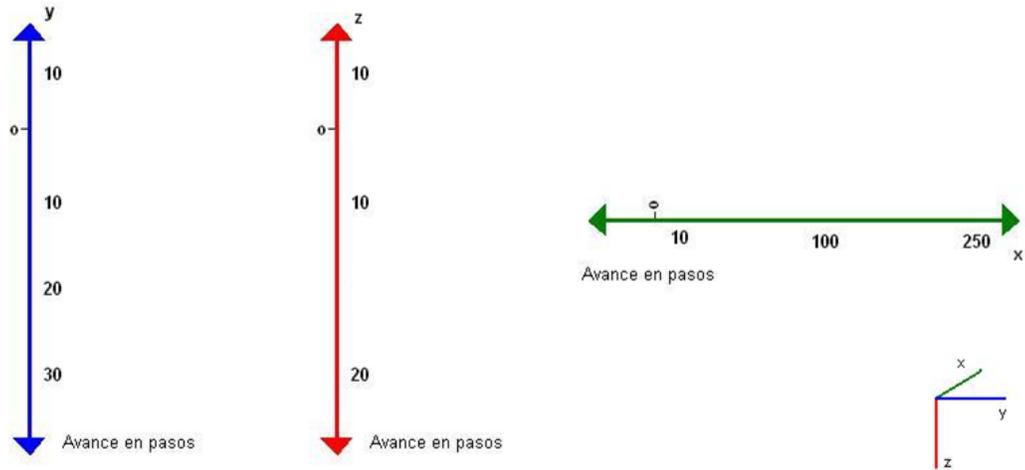


Figura 4.1. Caracterización acústica.

Para llevar a cabo la prueba se utilizaron dos transductores de ultrasonido, un Krautkramer a 2.6 MHz y un Aerotech a 3.5 MHz. En la figura 4.2 se observa la conexión que se realizó.

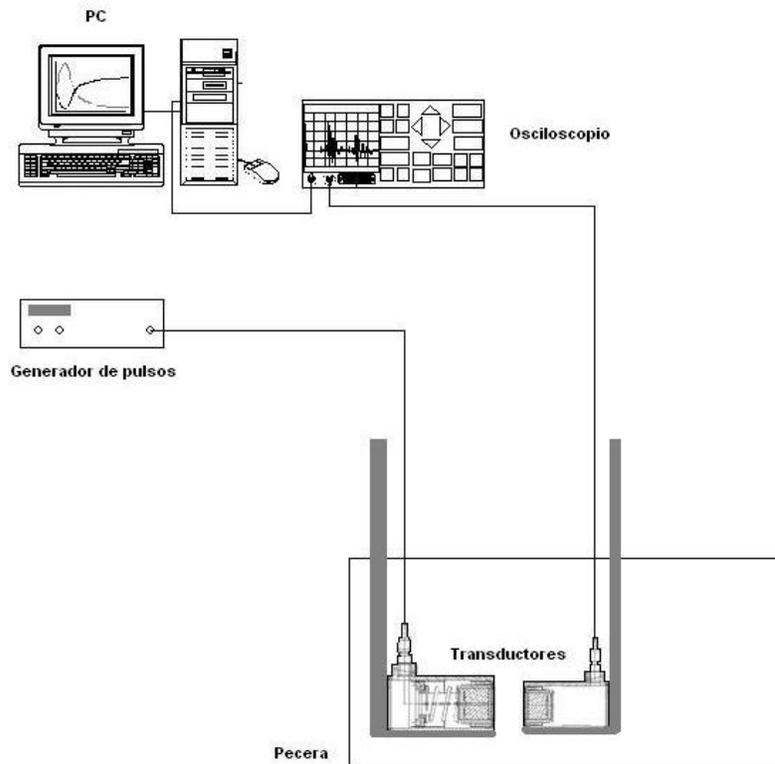


Figura 4.2. Sistema montado para la caracterización acústica.

El transductor Krautkramer se dejó fijo, este transductor es el que manda la señal, el transductor móvil es el Aerotech, el cual capta las señales, las manda al osciloscopio, y este las manda a la PC. Ambos transductores permanecen sumergidos dentro de un liquido, en este caso es agua. En la figura 4.3, se muestra el sistema completo.



a)



b)



c)

Figura 4.3 .Sistema para la caracterización de transductores, a) Osciloscopio, b) tarjeta de control y etapa de potencia, c) Posicionador XYZ.

Una vez adquiridos los datos, se graficaron y se observaron las características que presentaron, las figuras 4.4, 4.5 y 4.6 muestran estas gráficas, se puede observar que hay un disminución en la amplitud, lo que representa una pérdida de la señal, esta pérdida se produce al alejarse el transductor a caracterizar, del transductor emisor.

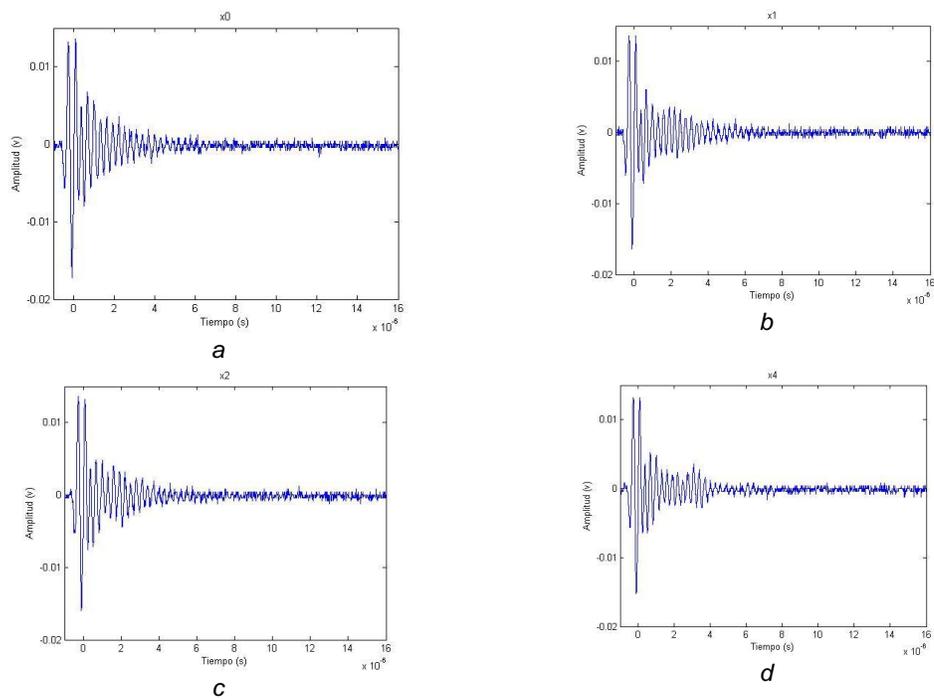


Figura 4.4. Formas de onda obtenidas en la adquisición con los transductores, eje X, a) origen b) 10 pasos, c) 100 pasos, d) 250 pasos.

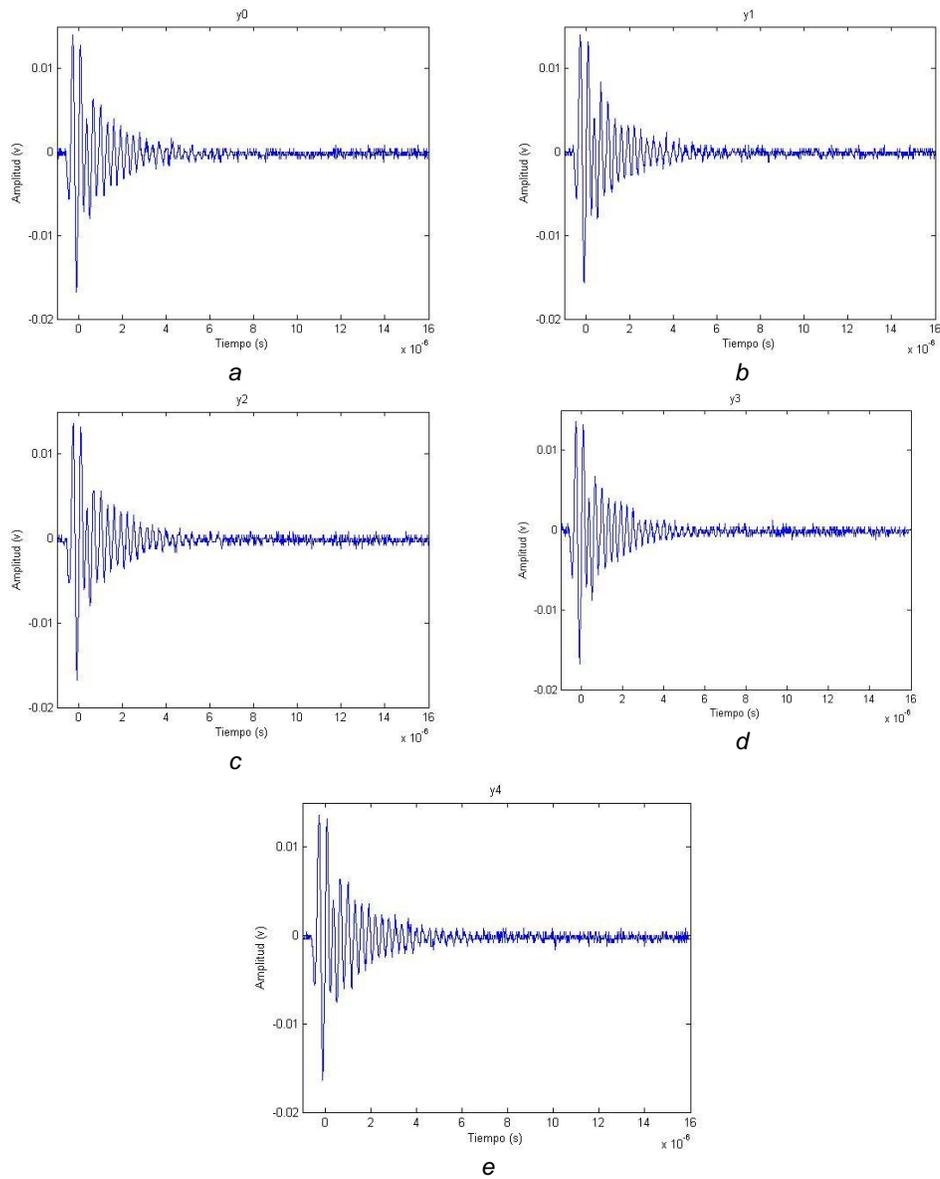


Figura 4.5. Formas de onda obtenidas en la adquisición con los transductores, eje Y, a) origen b) -10 pasos, c) 10 pasos, d) 20 pasos, e) 30 pasos.

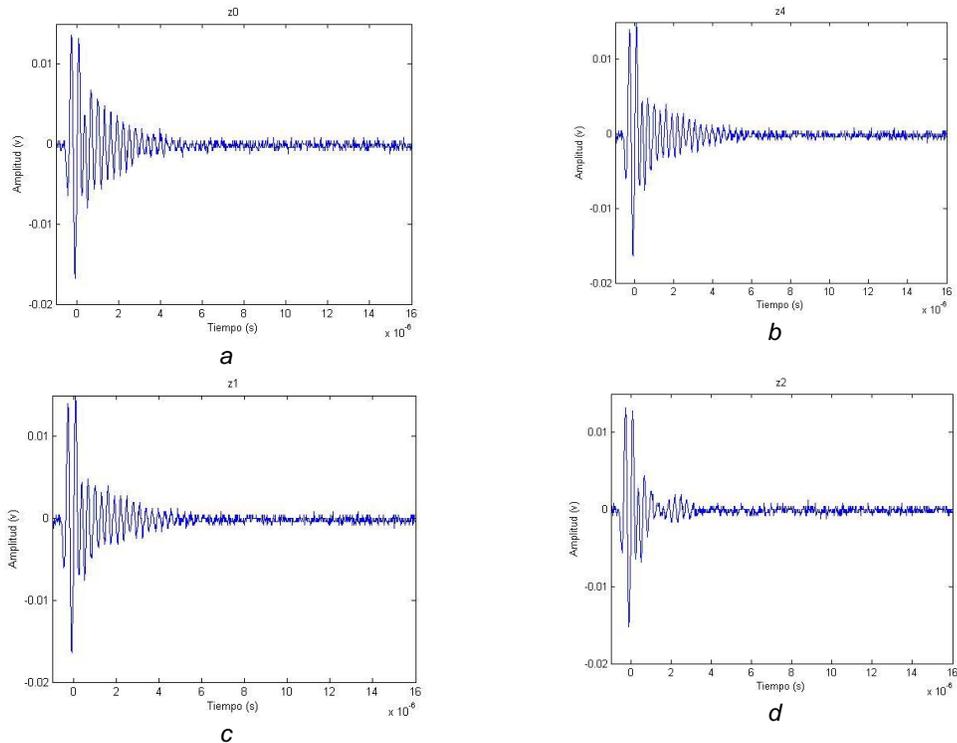


Figura 4.6. Formas de onda obtenidas en la adquisición con los transductores, eje Z, a) origen b) -10 pasos, c) 10 pasos, d) 20 pasos.

#### 4.4. Resultados y Conclusiones

Los resultados que se pueden obtener a partir de las pruebas realizadas al sistema, son:

- El sistema posicionador presenta una repetibilidad alta en el movimiento mecánico
- Una variación mínima en los desplazamientos por eje.
- Gracias a la repetibilidad del sistema podemos mencionar que las señales adquiridas en pruebas acústicas son las correctas.

Por lo tanto podemos concluir que el software para el control del sistema de posicionamiento muestra un funcionamiento estable en varios ítems de prueba (continuo, pausado, acústico, etc.), lo que nos permite tener un mejor control al hacer la caracterización de los transductores ultrasónicos. Este proceso tiene por objetivo comprobar la respuesta mecánica y acústica de los transductores bajo el control de la interface.

El software nos permite el control independiente por cada eje, el manejo independiente del osciloscopio, control manual del barrido de los ejes desde el teclado, control automático para realizar pruebas (el control de los ejes y el osciloscopio lo hace el mismo software). Se logró el objetivo principal, desarrollar un sistema capaz de controlar y adquirir datos de una forma sencilla y amigable. El software es compatible con cualquier sistema de posicionamiento que ocupe el mismo tipo de motor y línea de osciloscopios, permitiendo su portabilidad y escalamiento en sistemas más robustos, respetando las características básicas de control. El software puede ser modificado sin que se presenten problemas en la programación de la interfaz.



## Bibliografía

- [1] El ABC de la instrumentación en el control de Procesos Industriales, Gilberto Enríquez Harper. Editorial Limusa, 2004.
- [2] Sistemas de instrumentación, Pere J. Riu Costa, Javier Rosell Ferrer, Juan Ramos Castro. Edicions UPC, 1995.
- [3] Instrumentación virtual: adquisición, procesado y análisis de señales, Antoni Mànuel, Joaquim Olivé, Biel Soleanchez Robertlive, Antoni Mànuel, Francesc J. Ediciones Alfa Omega, 2002.
- [4] Ingeniería de control moderna, Katsuhiko Ogata, Pearson Educación, 2003.
- [5] Electrónica de potencia: circuitos, dispositivos y aplicaciones, Muhammad H. Rasid, Pearson Educación, 2004.
- [6] Adquisición y distribución de señales, Ramón Pallás Areny, Marcombo, 2005.
- [7] Señales y sistemas, Michael Roberts, McGraw-Hill, 2005.
- [8] Instrumentos electrónicos básicos, Ramón Pallás Areny, Marcombo 2006.
- [9] Amplificadores operacionales y circuitos integrados lineales. Robert F. Coughlin, Frederick F. Driscoll, Pearson Educación, 1999.
- [10] Fundamentals of Operational Amplifiers and Linear Integrated Circuits, Howard M. Berlin, Frank C. Getz, Prentice Hall, 1990.
- [11] Introducción a la bioingeniería, José Mompín Poblet, Marcombo 1988.
- [12] Mediciones y pruebas eléctricas y electrónicas, Bill Bolton, Marcombo, 1995.
- [13] Máquinas eléctricas y sistemas de potencia, Théodore Wildi, Rodolfo Navarro Salas, Pearson Educación, 2007.
- [14] El XYZ de los osciloscopios Tektronix
- [15] GPIB: [www.uhu.es/diego.lopez/ICI/GPIB.pdf](http://www.uhu.es/diego.lopez/ICI/GPIB.pdf)
- [16] GPIB: <http://sine.ni.com/np/app/culdesac/p/ap/ictrl/lang/es/pg/1/sn/n17:ictrl,n21:25/docid/tut-3419>
- [17] PXI: <http://www.ni.com/pxi/esa/>
- [18] USB: <http://info geekcatamarca.wordpress.com/2010/04/03/caracteristicas-de-transmision-de-un-puerto-usb/>

[19]<http://zone.ni.com/devzone/cda/tut/p/id/5934>

[20]Labview: entorno gráfico de programación, José Pelegrí Sebastián, Marcombo, 2007.

[21]LabVIEW for everyone: graphical programming made even easier, Lisa K. Wells, Jeffrey Travis, Prentice Hall, 1996.

# ANEXOS

---

## Anexo A. Descripción del código.

El panel principal (figura A.1a) está controlado mediante un ciclo while y un cuadro de eventos (figura A.1b y A.1c). Las principales acciones que se realizan en esta parte son:

- Al ejecutar el programa, lo primero que se hace es crear un archivo donde se van guardando las diferentes variables que se ocupan durante la ejecución. Como son la ruta de los directorios donde se guardan los datos, las posiciones de los motores, los nombres de los dispositivos de control y adquisición. Esto se hace tener almacenados los datos de la última ejecución que se realizó.
- En el cuadro de eventos manejamos lo que son las opciones de la barra de menús, cada vez que se realiza alguna acción en la barra, activamos un evento, lo que obliga a ejecutar una tarea en específico. La estructura de eventos se encuentra dentro de un ciclo while, esto para ejecutarla continuamente mientras el programa principal este en ejecución. El ciclo termina cuando seleccionamos la opción salir del menú, o de la ventana principal. Dentro del cuadro de eventos, la mayoría del código son subVIs, es decir son rutinas independientes que se mandan a llamar al ejecutar cierto evento.
- El panel de mensajes es controlado mediante un ciclo while. Cada vez que ocurre cierto evento durante la ejecución del programa, este guarda cierto mensaje dentro de una cola, la cual se va almacenando dentro de un arreglo de datos, para posteriormente guardarlos en un archivo de texto. El ciclo while termina cuando cerramos el panel principal.

El VI guardar (figura A.2a) lo ocupamos cuando estamos haciendo las adquisiciones en modo manual y queremos almacenar los datos que se estén recopilando. En la figura A.2b, podemos observar el diagrama de bloques de este cuadro de diálogo, lo que hace es pasar únicamente el nombre del archivo al VI mostrado en la figura A.3. Este VI manda el valor de guardar al VI OSciloscopioSC.vi, el cual guarda los datos que se estén adquiriendo en ese momento. OSciloscopio.vi, es el VI encargado de hacer la comunicación con el osciloscopio. Su funcionamiento se explica en el Capítulo 3.5.2. En la figura A.4b se muestra el código para el panel tipo de barrido Figura A.4a. La figura A.5b muestra el código para el panel configuración (figura A.5a). La figura A.6 muestra el diagrama de flujo del VI encargado del control del barrido para las adquisiciones.

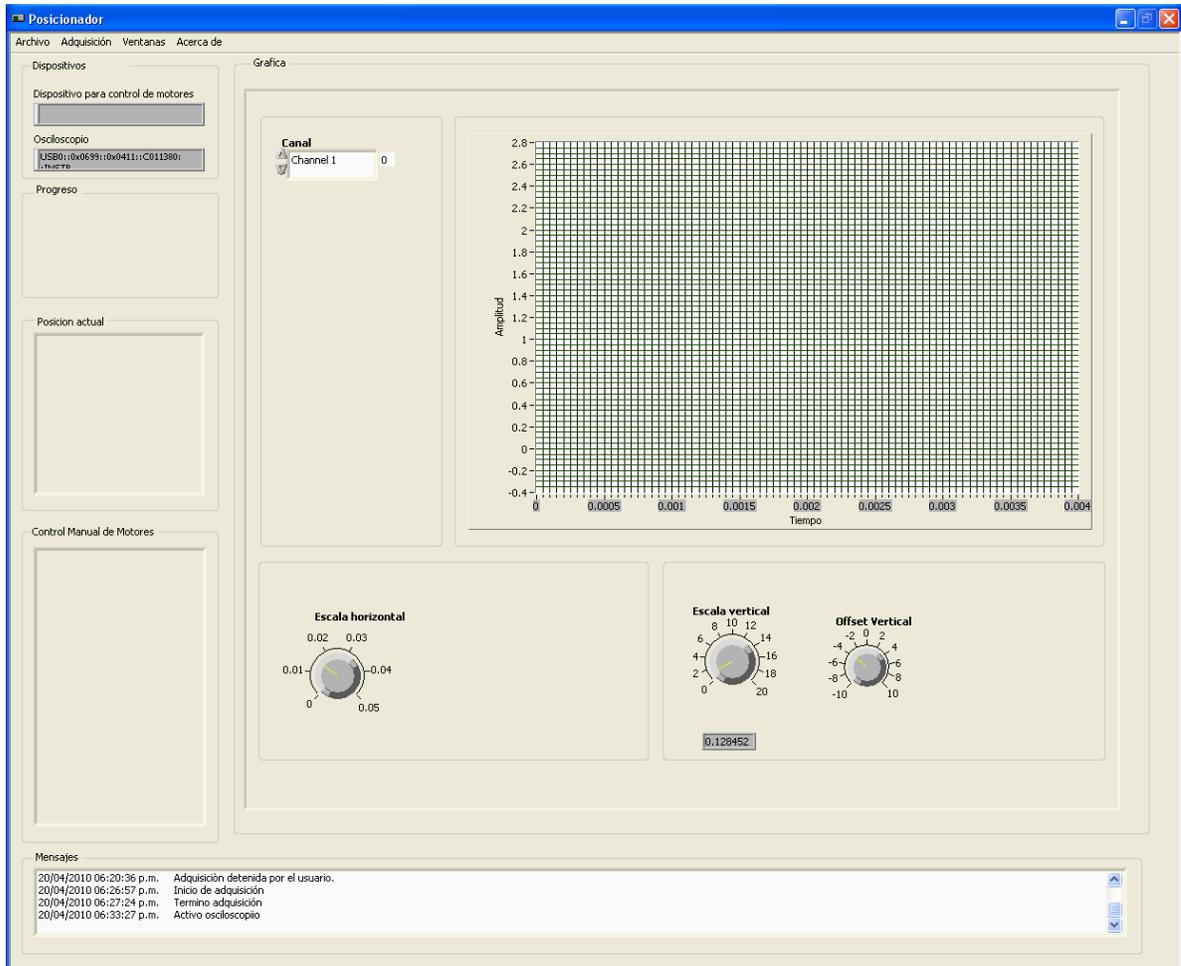


Figura A.1a. Panel principal.

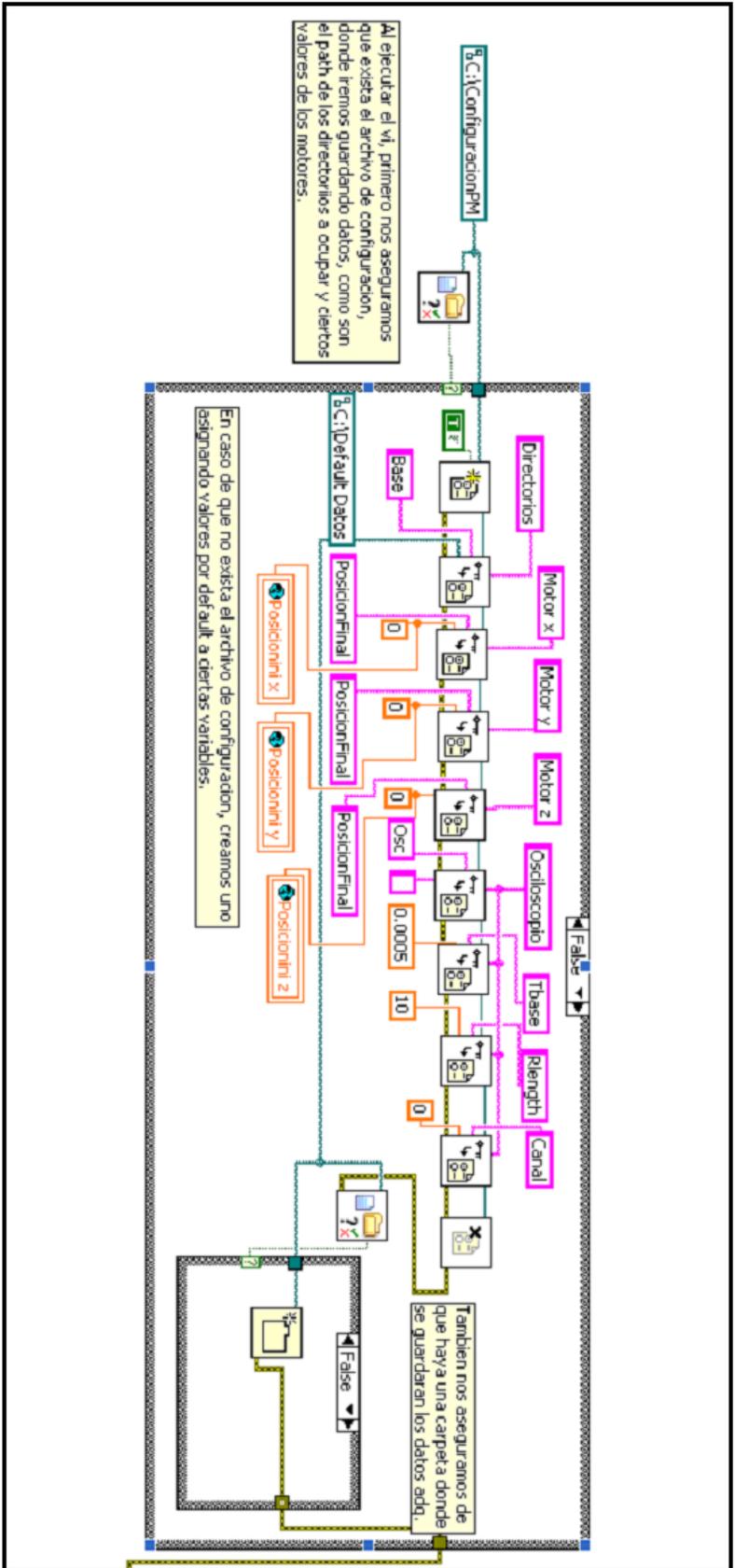


Figura A.1b. Diagrama de bloques del panel principal. a) Sección 1, Creación del archivo de configuración.

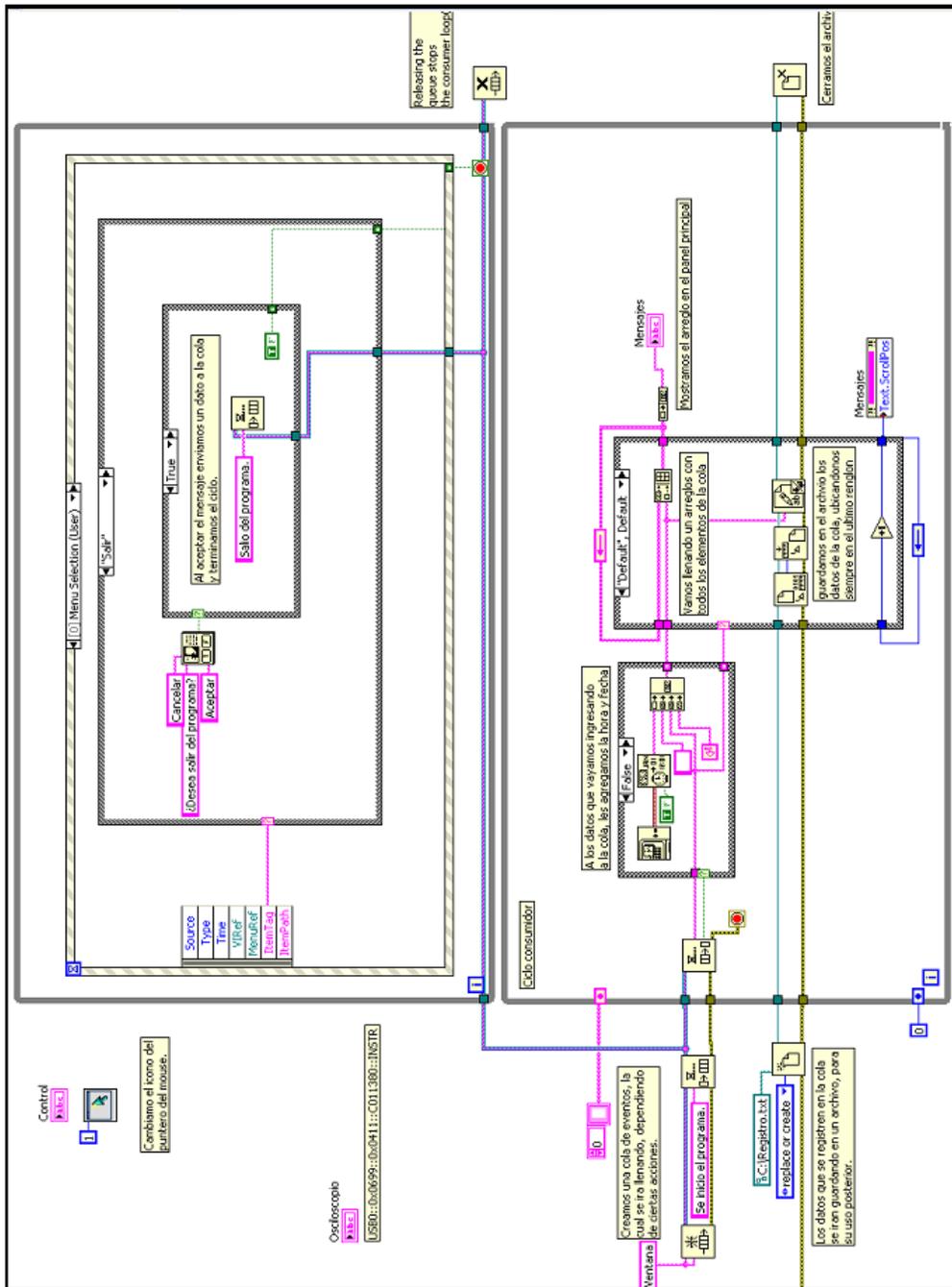
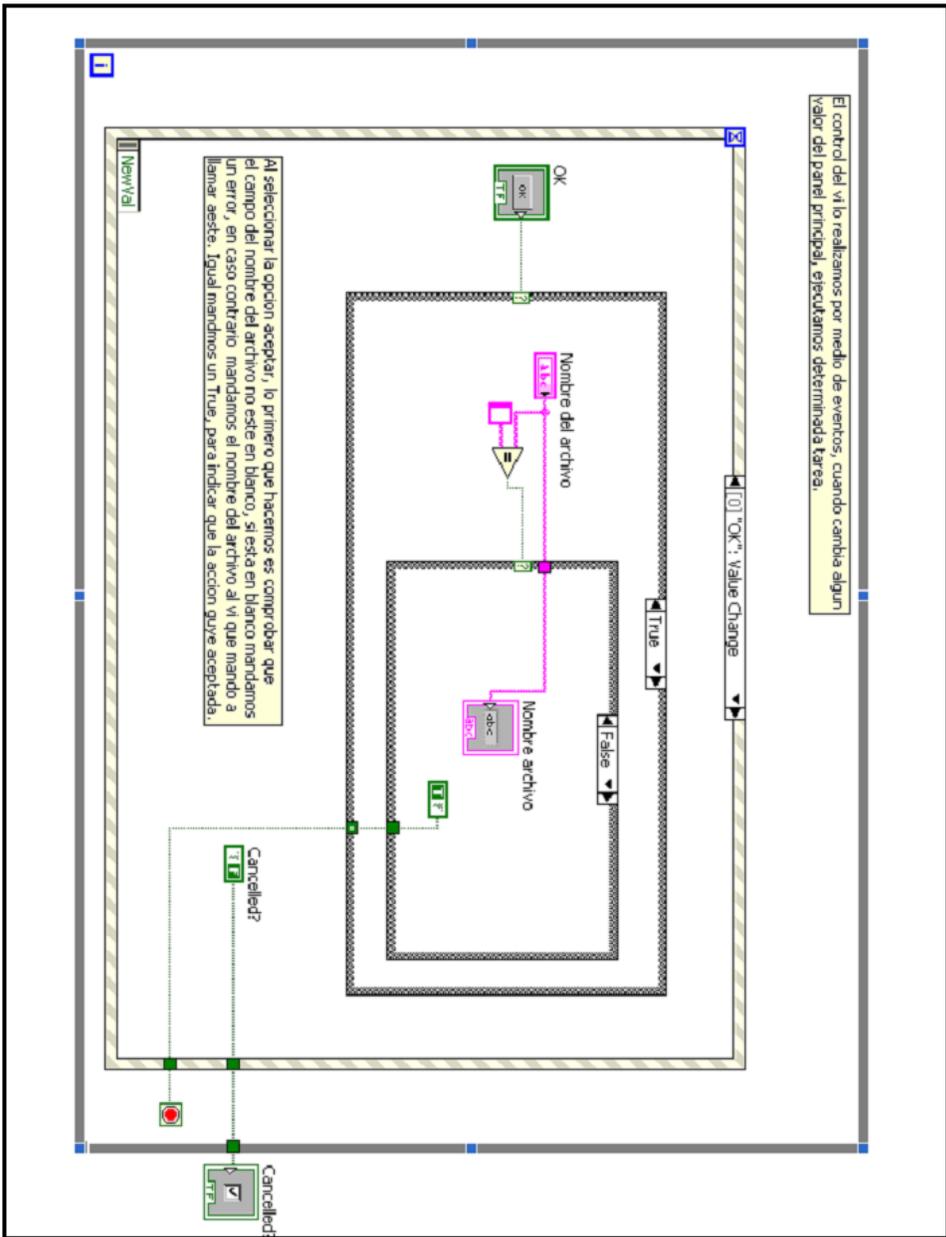


Figura A.1c. Diagrama de bloques del panel principal, Sección 2, Manejo de los eventos para las diferentes opciones del panel.



a)



b)

Figura A.2. a) Cuadro de dialogo guardar, b) Diagrama de bloques del cuadro de dialogo guardar.

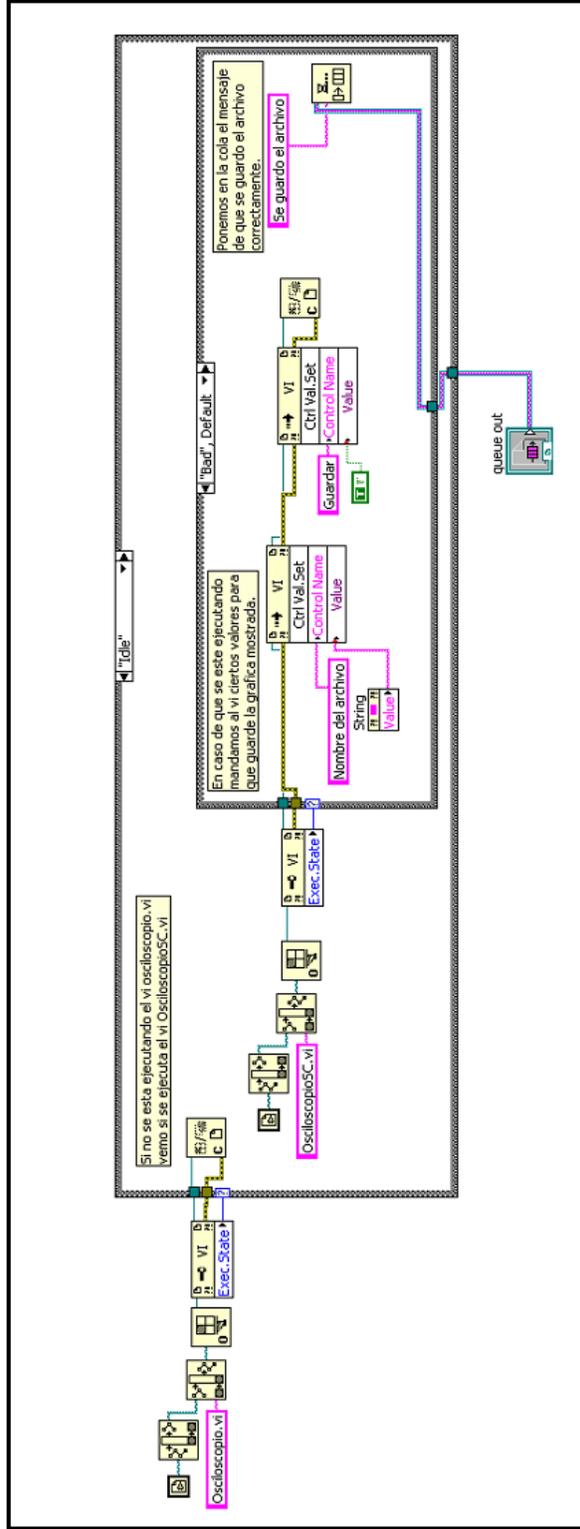


Figura A.3. diagrama de bloques que se encarga de guardar los datos adquiridos. Cuando se están desplegando algún panel del osciloscopio y seleccionamos la opción guardar, lo que se hace es pasar un valor de verdadero, desde este código, al diagrama de bloques del panel del osciloscopio, para que ejecute la acción de guardar.

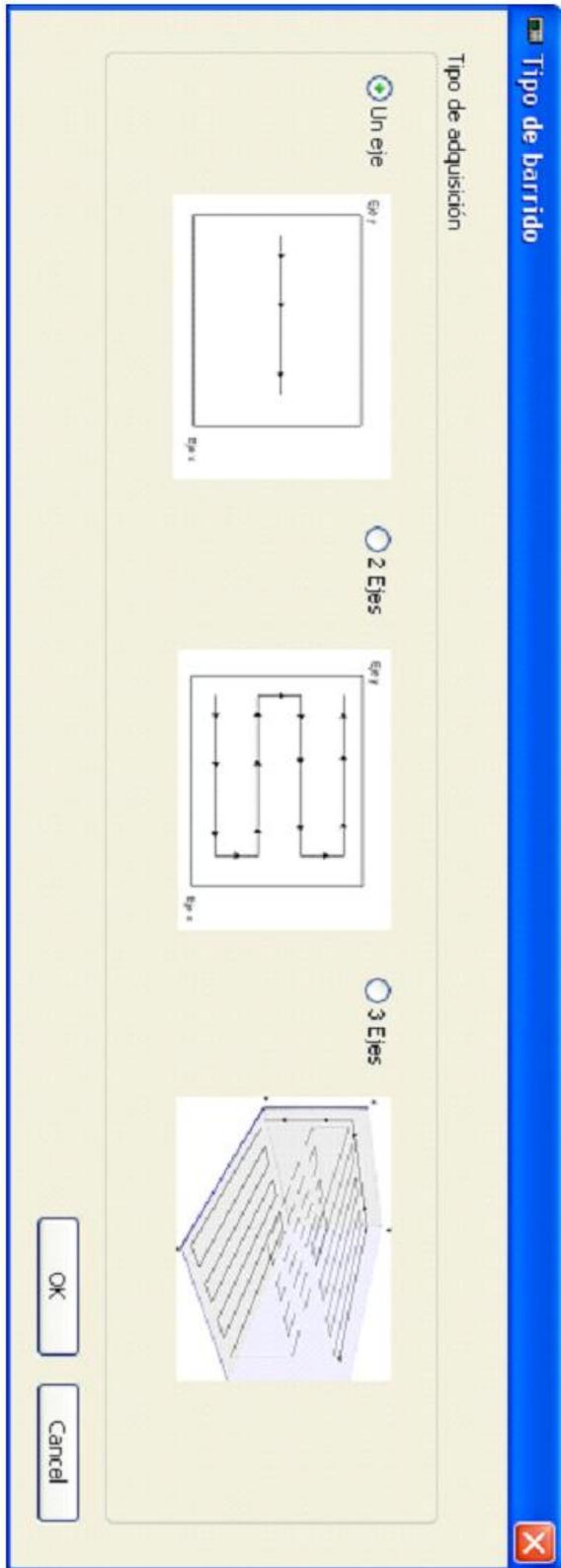


Figura A. 4a. Tipo de barrido.

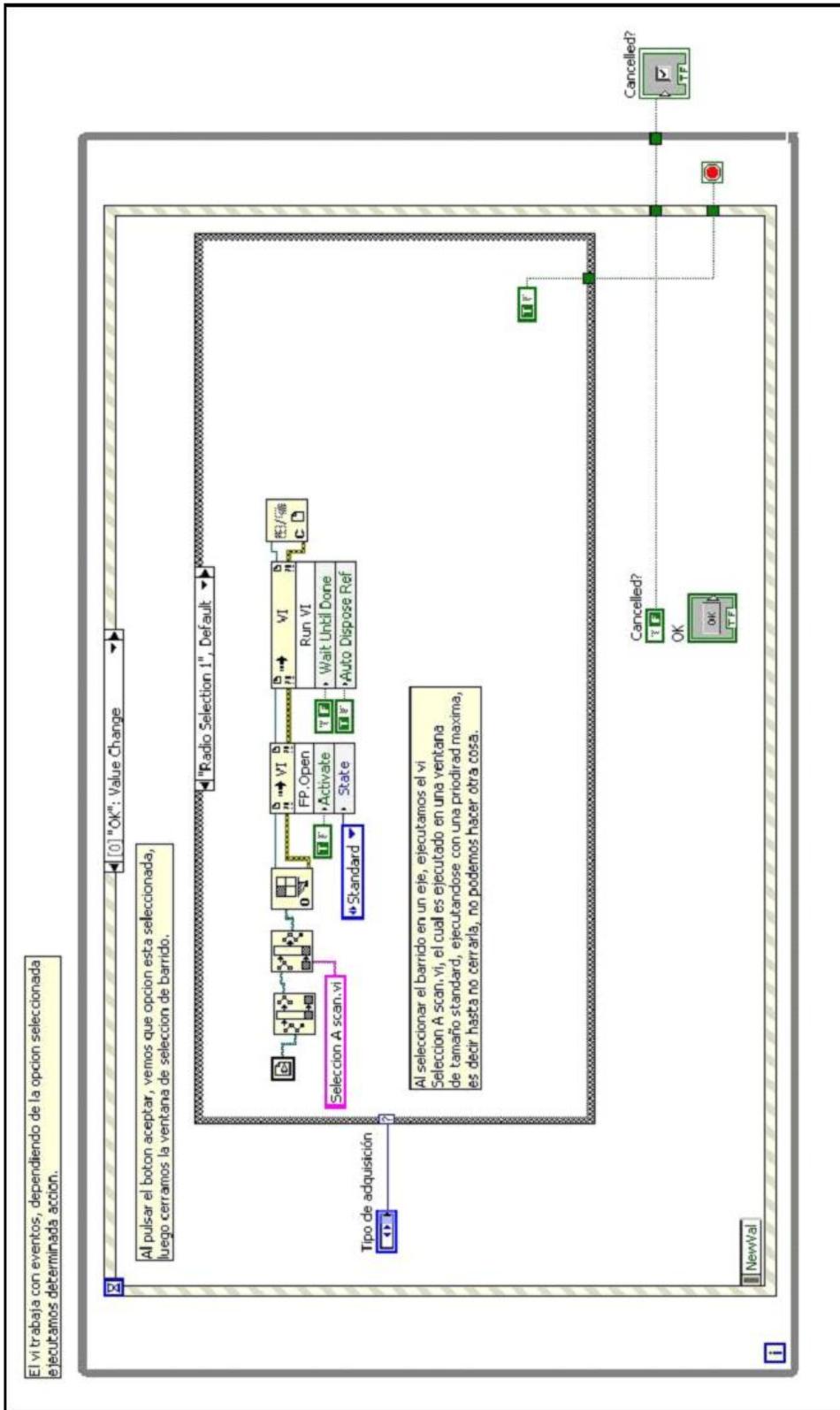


Figura A. 4b. Diagrama de bloques del panel "Tipo de barrido", funciona a través de eventos, dependiendo de la selección que hayamos efectuado, mandamos a llamar al VI correspondiente y cerramos este.

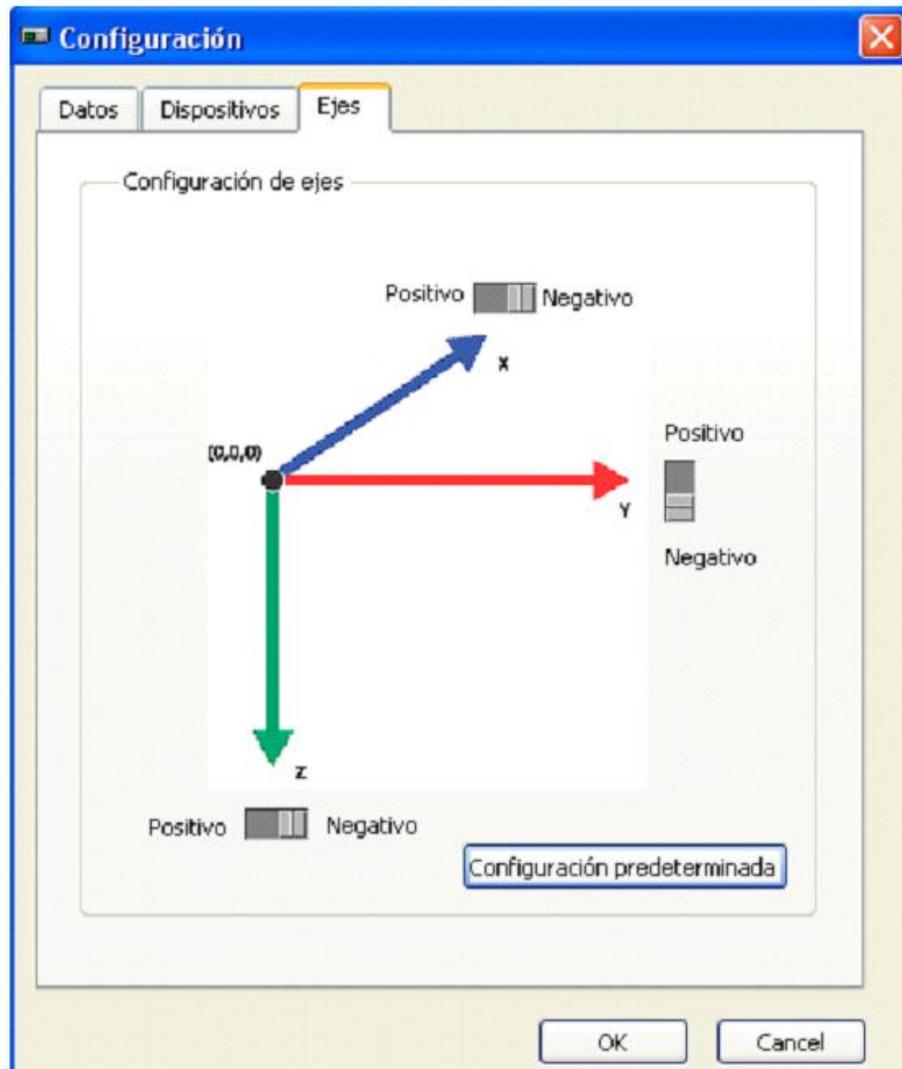


Figura A.5a. Panel configuración.

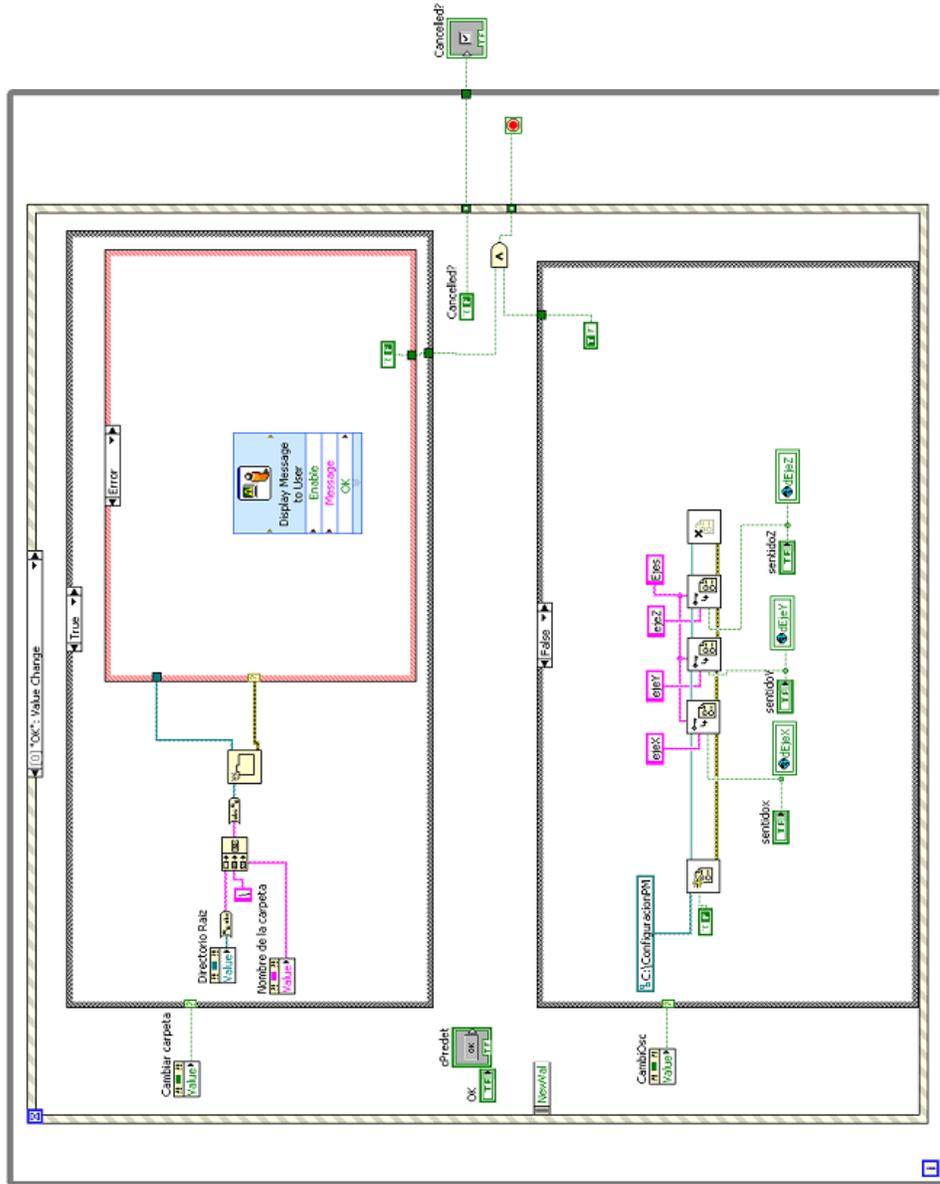


Figura A.5b. Diagrama de bloques del panel "Configuración", su funcionamiento se basa en eventos, los cambios únicamente se guardan al seleccionar el botón "aceptar" del panel frontal, los cambios son guardados en un archivo de configuración, para su uso posterior.

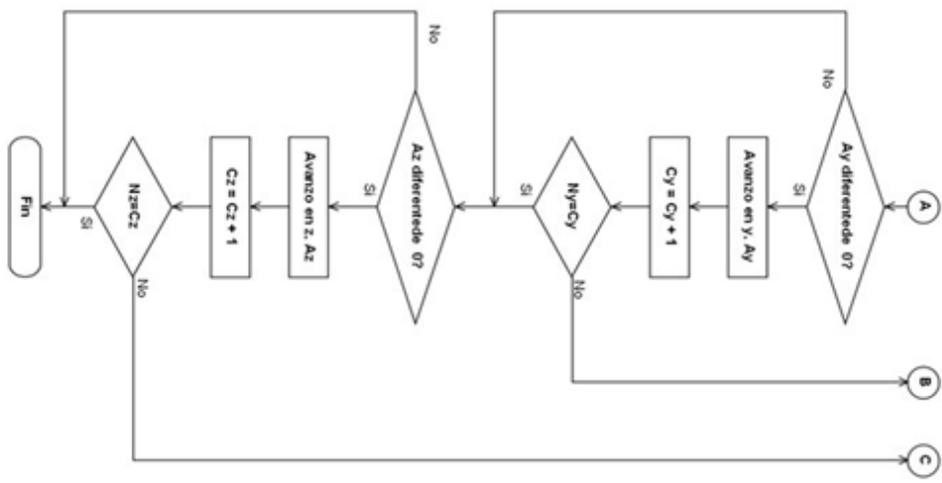
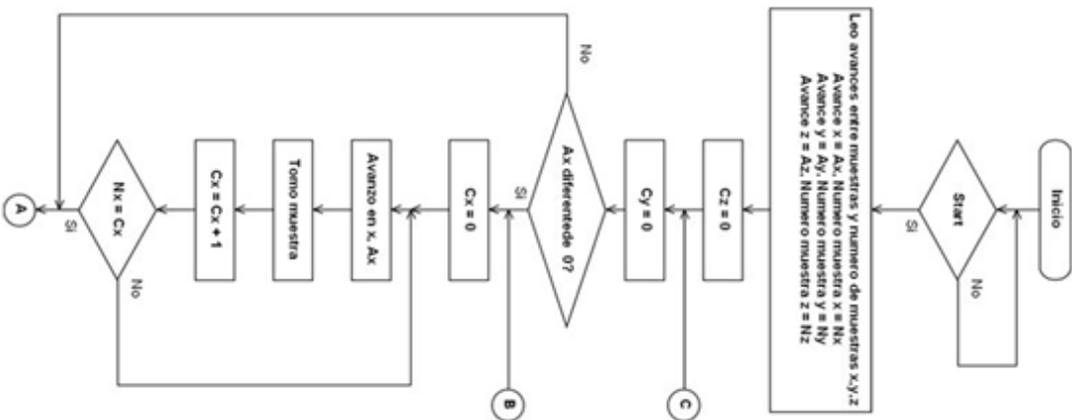


Figura A.6. Diagrama de flujo del funcionamiento del VI encargado de realizar las adquisiciones.

## Anexo B. Guía rápida de manejo

Al ejecutar el programa se nos presenta una pantalla de inicio (figura B.1a), durante la cual se comprueba que la tarjeta de control este conectada a la PC, en caso de que no se encuentre conectada, se nos desplegara un error (figura B.1b), y no podremos entrar al panel principal del software.

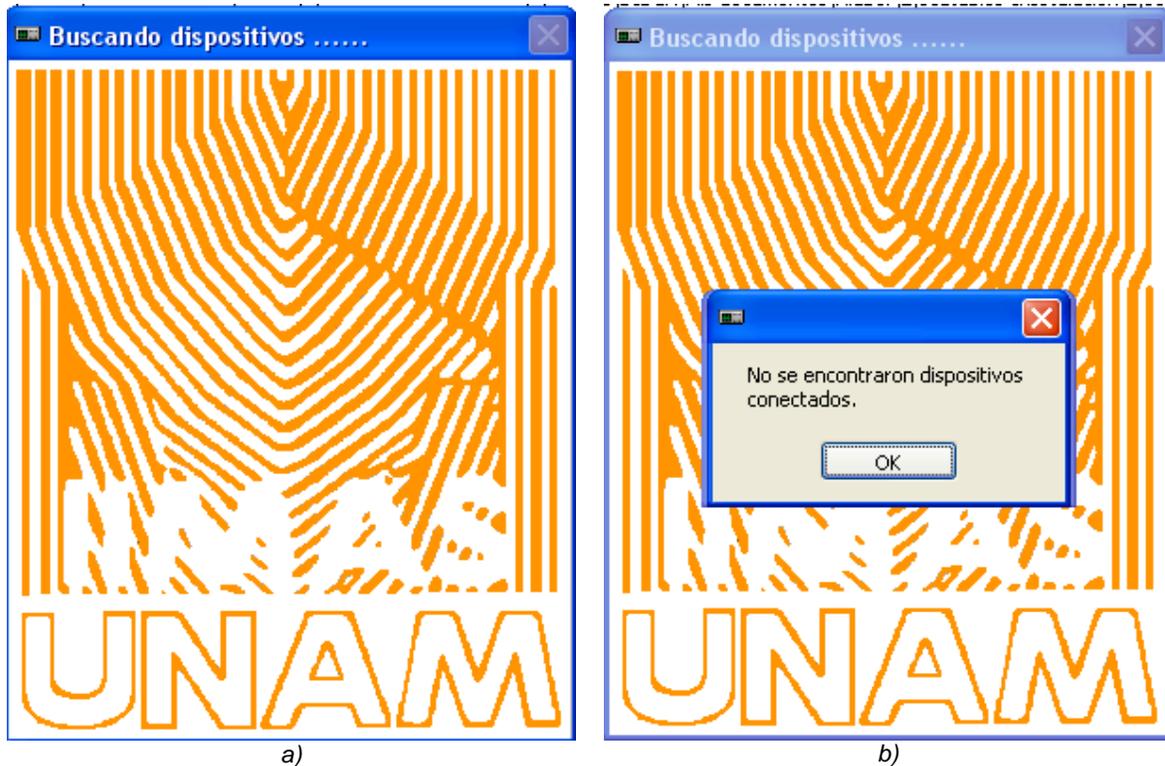


Figura B.1. Splash de inicio del software.

Una vez que ya estemos en el panel principal, ya podemos empezar a realizar una nueva adquisición. Para realizar una adquisición automática, primero debemos seleccionar la carpeta donde se guardan los datos, por defecto los datos se guardan en la ruta **C:\Default Datos**, para seleccionar una nueva ruta nos situamos en el **submenú configuración**, que se encuentra en el **menú Archivo**, una vez que se haya seleccionado la opción se nos abrirá el cuadro de diálogo de configuración, nos situamos en la pestaña **Datos**, y seleccionamos la opción **cambiar** (figura B.2). Una vez que se liberen las opciones que se encuentran abajo, seleccionamos la ruta donde se guardaran los datos, además del nombre de nuestra carpeta.

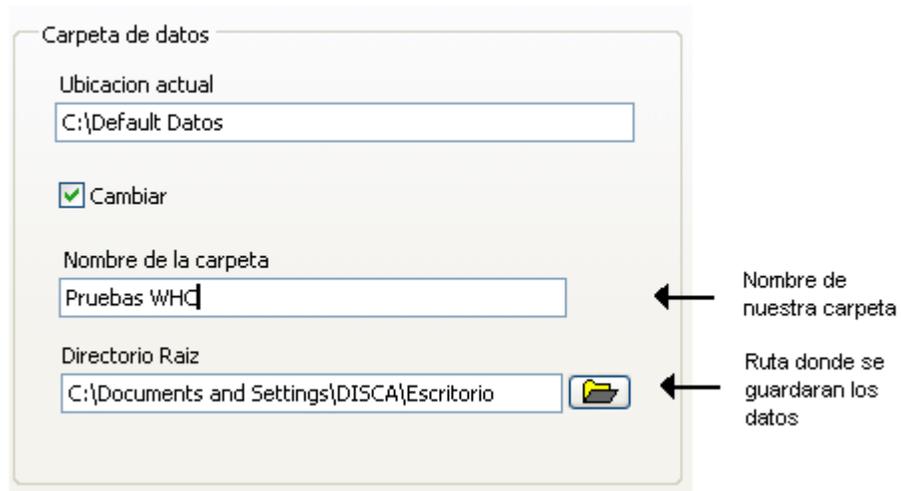


Figura B.2. Configuración de la carpeta donde se guardaran los datos.

Desde este mismo cuadro de diálogo tenemos la posibilidad de seleccionar el osciloscopio a ocupar (figura B.3a), esto se hace desde la pestaña **Dispositivos**, además de configurar los el sentido de barrido de cada eje, desde la pestaña ejes (figura B.3b).

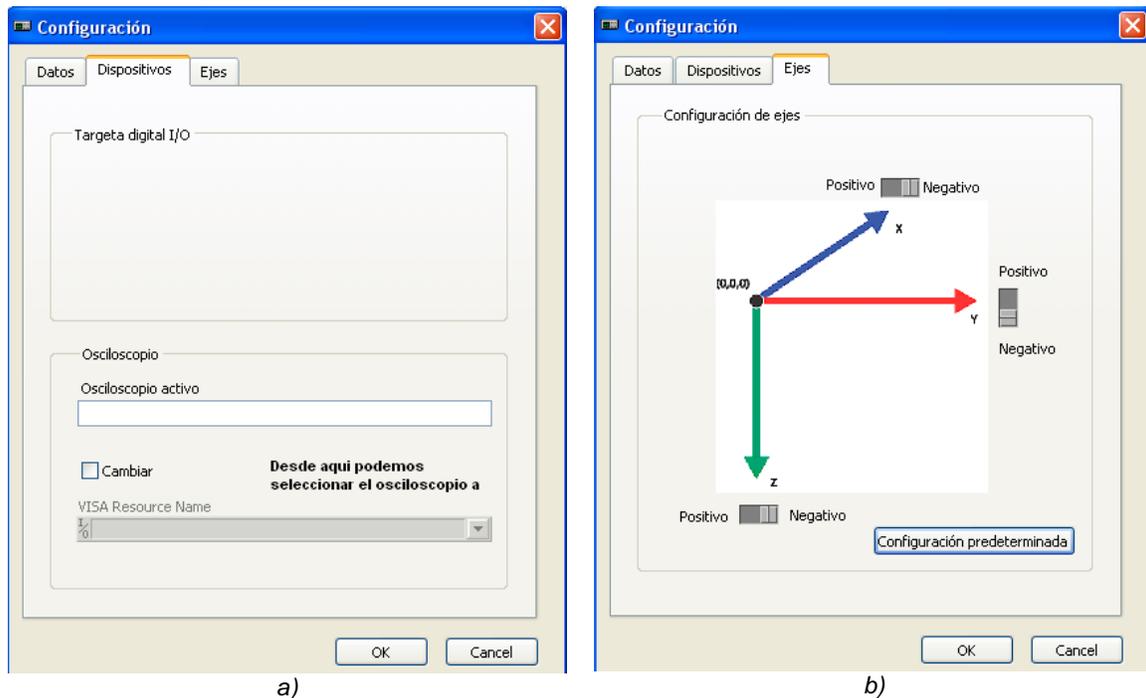


Figura B.3. a) Para seleccionar un osciloscopio, activamos la casilla “cambiar”, para después seleccionar el osciloscopio, b) Sentido de barrido de los ejes.

Después de haber hecho los cambios pertinentes, hacemos clic en el botón “OK” del **cuadro de diálogo**, para guardar los cambios. Ahora procedemos a configurar el tipo de adquisición que vamos a realizar, se dará el ejemplo para una adquisición en un solo eje. Primero nos situamos en el submenú “Nueva” (figura B.4a), se abrirá un cuadro de diálogo, desde donde seleccionaremos el tipo de barrido a realizar (figura B.4b), ya sea en uno, dos o tres ejes. Seleccionamos la primera opción, un eje, hacemos clic en “OK”.

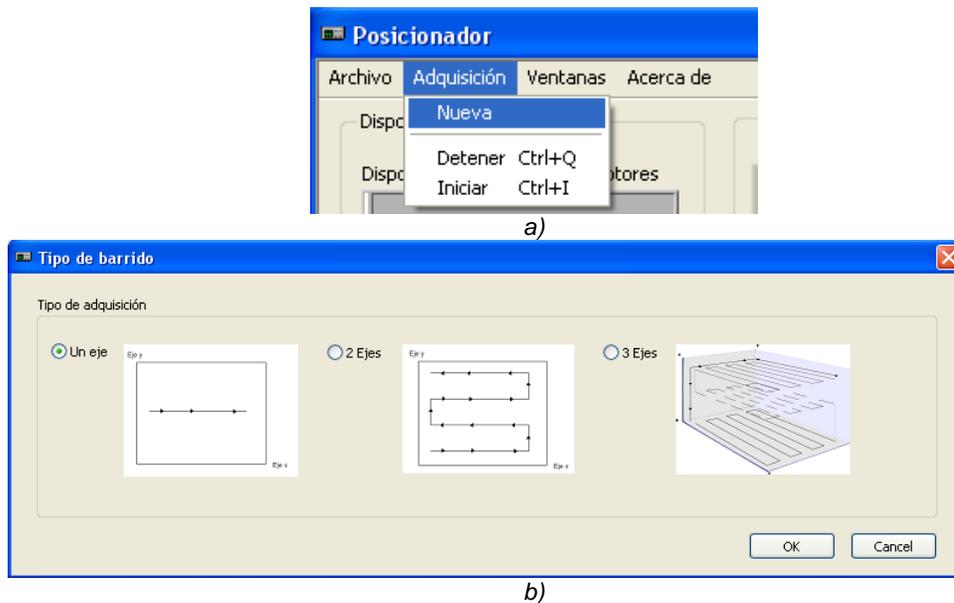


Figura B.4. Configuración para una nueva adquisición, a) Submenú “Nueva”, b) Cuadro de dialogo para seleccionar el tipo de barrido.

En el nuevo cuadro de diálogo que aparece (figura B.5a), configuramos el eje donde se hará el barrido, hacemos clic en siguiente, ahora configuramos la distancia entre muestras y el número de muestras a realizar, como se muestra en la figura B.5b, para la distancia entre muestras tenemos la posibilidad de poner las unidades del valor, ya sea en milímetros o micrómetros.

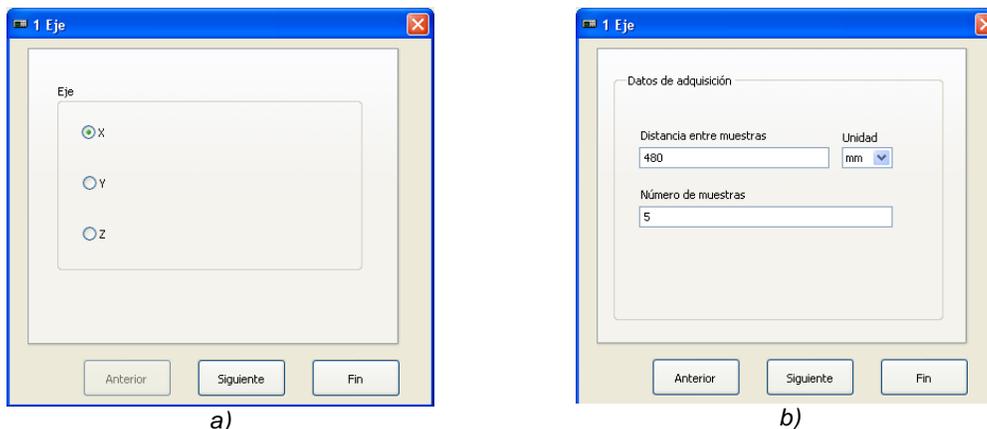


Figura B.5. a) Selección del eje, b) Configuración del barrido.

Por ultimo configuramos algunas características del osciloscopio (figura B.6), como son el canal desde donde se adquieran los datos, la longitud en tiempo de la muestra a tomar, y el número de puntos que contendrá la muestra adquirida, es decir el número de datos que se guardaran por cada muestra, para después ser reconstruidos.

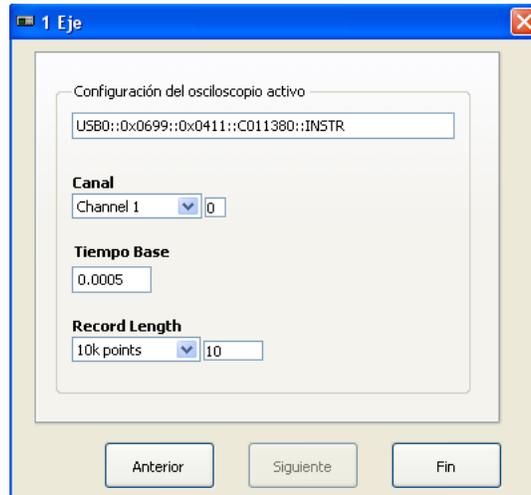


Figura B.6. Configuración del osciloscopio.

Ahora ya estamos listos para iniciar la adquisición, para esto nos situamos en el **submenú Iniciar**, que se encuentra dentro del **menú Adquisición**, o también podemos iniciar la adquisición con la combinación de teclas **Ctrl + I**. La figura B.7 muestra la pantalla de una adquisición.

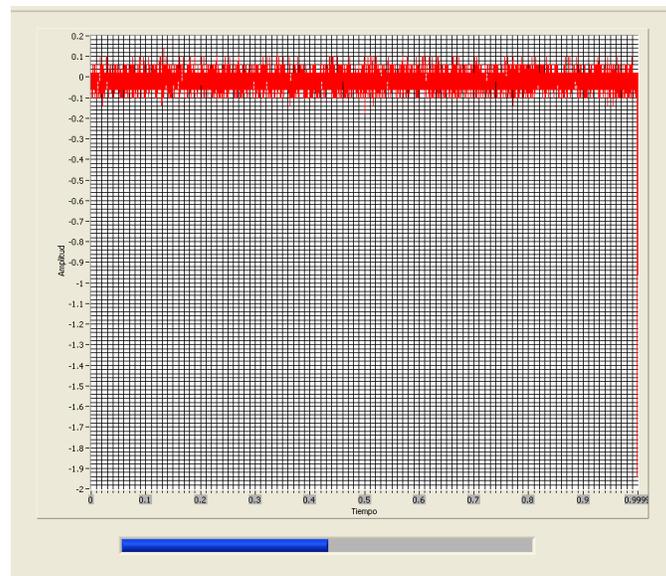


Figura B.7. Debajo de la gráfica se muestra una barra de progreso, que nos indica el avance que se lleva de la adquisición.

Una vez que se termina la adquisición el panel donde se muestra la gráfica se cierra, además que en la ventana de eventos se nos indica que se termino de realizar la adquisición (figura B.8).

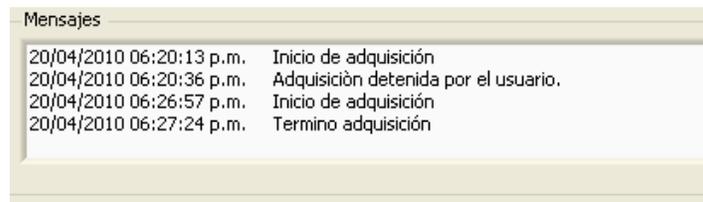
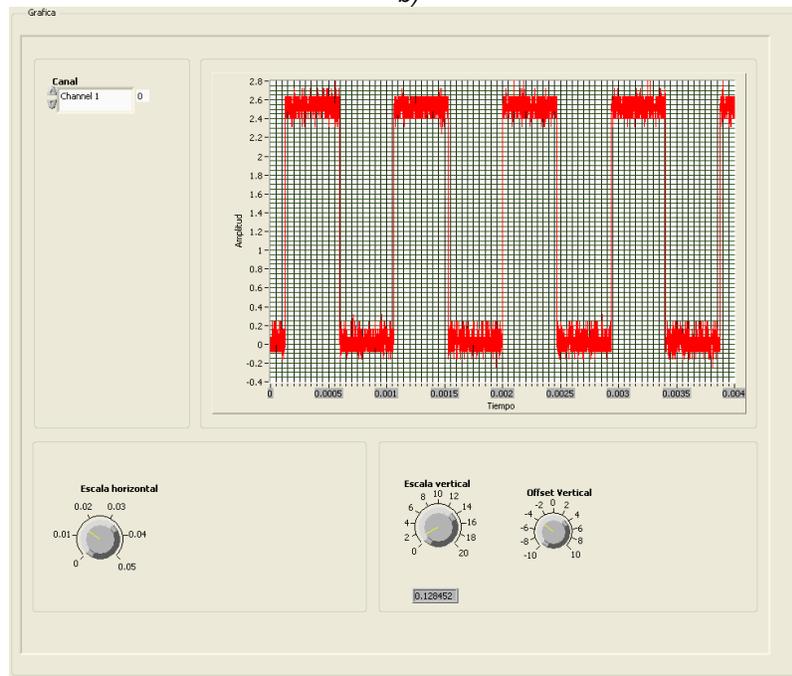


Figura B.8. Ventana de mensajes, donde se nos indica que ya se termino de realizar la adquisición en curso.

Para realizar una adquisición manual primero activamos el osciloscopio desde el **submenú osciloscopio** (figura B.9a), se nos darán dos opciones, la de controlar el osciloscopio desde la PC o desde el mismo dispositivo (hardware físico). En el panel gráfica, se mostrara el instrumento virtual (figura B.9b).



b)



b)

Figura B.9. a) Submenú Osciloscopio, c) Panel del dispositivo virtual.

Ahora activamos el control manual de barrido, seleccionamos el **submenú motor** (figura B.10a), ya podemos mover los ejes desde el teclado. Si queremos mover el eje X, presionamos la **tecla “X”**, para mover el eje Y, presionamos la **tecla “Y”**, para mover el eje Z, presionamos la **tecla “Z”**, solo podemos mover un eje a la vez. Para guardar los datos que estemos viendo en pantalla nos vamos al **submenú guardar** (figura B.10b) y hacemos click.



a)   
 b)   
 Figura B.10. a) Submenú Motor, b) Submenú Guardar.

Ya que términos de ocupar el programa, para salirnos tenemos tres opciones, desde el icono superior derecho (donde se encuentra el botón cerrar “x”), desde el **submenú Salir** (que se encuentra dentro del **menú Archivo**) o presionando la combinación de teclas **Ctrl + x**.

## Low-Cost Digital I/O Device for USB

### NI USB-6501

- Small, portable digital I/O device
- 24 digital I/O lines, one 32-bit counter
- Overvoltage protection, 8.5 mA current drive
- Full-speed USB (12 Mb/s) bus interface
- Built-in screw terminals with removable connectors for easier connectivity
- OEM version available with 34-pin IDC mass termination connectivity

#### Operating Systems

- Windows Vista, XP/2000
- Mac OS X<sup>1</sup>
- Linux<sup>®1</sup>

#### Recommended Software

- LabVIEW
- LabWindows™/CVI

#### Measurement Services Software (included)

- NI-DAQmx driver software
- Measurement & Automation Explorer configuration utility

<sup>1</sup>Mac OS X and Linux users need to download NI-DAQmx Base.



### Overview

The National Instruments USB-6501 is a portable digital I/O device, providing reliable data acquisition and control at a low price. With plug-and-play USB connectivity, the NI USB-6501 is simple enough for home/academic applications but robust and versatile enough for laboratory/industrial applications. The board-only version is ideal for embedded OEM applications.

Product	Bus	Digital I/O Lines	Counter/ Timers	Current Drive	Logic Level	Industrial Feature Set
NI 6501	USB	24	1	8.5 mA	5V TTL / CMOS	–

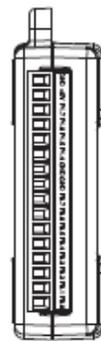
Table 1. USB-6501 Features

### Hardware

The USB-6501 is a full-speed USB device that provides 24 DIO lines and a 32-bit counter. The 24 digital lines are arranged in three ports: P0.<0..7>, P1.<0..7>, and P2.<0..7>. P2.7 can also function as a 32-bit counter. You can individually program each of the USB-6501 DIO lines as a DI or DO line and use the DIO lines to monitor or control digital signals. All samples of the DI lines and updates of the DO lines are software-timed.

The default configuration of the USB-6501 DIO ports is open-drain, for 5 V operation with an onboard 4.7 k $\Omega$  pull-up resistor. You can add an external, user-provided, pull-up resistor to increase the source current drive up to 8.5 mA per line.

Each DIO signal is protected against overvoltage, undervoltage, and overcurrent conditions, as well as ESD events. At system startup and reset, the hardware sets all DIO lines to high-impedance inputs.



Terminal	Signal
1	GND
2	+5 V
3	P1.7
4	P1.6
5	P1.5
6	P1.4
7	GND
8	GND
9	P2.7
10	P2.6
11	P2.5
12	P2.4
13	P2.3
14	P2.2
15	P2.1
16	P2.0



Terminal	Signal
17	P0.0
18	P0.1
19	P0.2
20	P0.3
21	P0.4
22	P0.5
23	P0.6
24	P0.7
25	GND
26	GND
27	P1.0
28	P1.1
29	P1.2
30	P1.3
31	+5 V
32	GND

Figure 1. USB-6501 Front



# Mixed Signal Oscilloscopes

## MSO3000 Series, DPO3000 Series Data Sheet



### Features & Benefits

#### Key Performance Specifications

- 500, 300, 100 MHz Bandwidth Models
- 2 and 4 Analog Channel Models
- 16 Digital Channels (MSO Series)
- 2.5 GS/s Sample Rate on All Channels
- 5 Megapoint Record Length on All Channels
- >50,000 wfms Maximum Waveform Capture Rate
- Suite of Advanced Triggers

#### Ease of Use Features

- Wave Inspector® Controls Provide Easy Navigation and Automated Search of Waveform Data
- 29 Automated Measurements, and FFT Analysis for Simplified Waveform Analysis
- TekVPI® Probe Interface Supports Active, Differential, and Current Probes for Automatic Scaling and Units
- 9 in. (229 mm) WVGA Widescreen Color Display
- Small Footprint and Lightweight – Only 5.8 in. (147 mm) deep and 9 lb. (4 kg)

#### Connectivity

- USB 2.0 Host Port on both the Front Panel and Rear Panel for Quick and Easy Data Storage, Printing, and Connecting a USB Keyboard
- USB 2.0 Device Port on Rear Panel for Easy Connection to a PC or Direct Printing to a PictBridge®-compatible Printer
- Integrated 10/100 Ethernet Port for Network Connection and Video Out Port to Export the Oscilloscope Display to a Monitor or Projector

#### Optional Serial Triggering and Analysis

- Automated Serial Triggering, Decode, and Search Options for I<sup>2</sup>C, SPI, CAN, LIN, RS-232/422/485/UART, and I<sup>2</sup>S/LJ/RJ/TDM

#### Mixed Signal Design and Analysis (MSO Series)

- Automated Triggering, Decode, and Search on Parallel Buses
- Multichannel Setup and Hold Triggering
- MagniVu™ High-speed Acquisition Provides 121.2 ps Fine Timing Resolution on Digital Channels

#### Optional Application Support

- Power Analysis
- HDTV and Custom Video Analysis

### Feature-rich Tools for Debugging Mixed Signal Designs

With the MSO/DPO3000 Mixed Signal Oscilloscope Series, you can analyze up to 20 analog and digital signals with a single instrument to quickly find and diagnose problems in complex designs. Bandwidths up to 500 MHz and a minimum of 5x oversampling on all channels ensure you have the performance you need for many of today's mainstream applications. To capture long windows of signal activity while maintaining fine timing resolution, the MSO/DPO3000 offers a deep record length of 5 Mpoints standard on all channels.

With Wave Inspector® controls for rapid waveform navigation, automated serial and parallel bus analysis, and automated power analysis - the MSO/DPO3000 Oscilloscope Series from Tektronix provides the feature-rich tools you need to simplify and speed debug of your complex design.



## Characteristics

### Vertical System Analog Channels

Characteristic	MSO3012 DPO3012	MSO3014 DPO3014	MSO3032 DPO3032	MSO3034 DPO3034	MSO3054 DPO3054
Input Channels	2	4	2	4	4
Analog Bandwidth (-3 dB)	100 MHz	100 MHz	300 MHz	300 MHz	500 MHz
Calculated Rise Time 5 mV/div (typical)	3.5 ns	3.5 ns	1.17 ns	1.17 ns	700 ps
Hardware Bandwidth Limits	20 MHz		20 MHz, 150 MHz		
Input Coupling	AC, DC, GND				
Input Impedance	1 M $\Omega$ $\pm$ 1%, 75 $\Omega$ $\pm$ 1%, 50 $\Omega$ $\pm$ 1%				
Input Sensitivity Range, 1 M $\Omega$	1 mV/div to 10 V/div				
Input Sensitivity Range, 75 $\Omega$ , 50 $\Omega$	1 mV/div to 1 V/div				
Vertical Resolution	8 bits (11 bits with Hi Res)				
Maximum Input Voltage, 1 M $\Omega$	300 V <sub>RMS</sub> with peaks $\leq$ $\pm$ 450 V				
Maximum Input Voltage, 75 $\Omega$ , 50 $\Omega$	5 V <sub>RMS</sub> with peaks $\leq$ $\pm$ 20 V				
DC Gain Accuracy	$\pm$ 1.5% for 5 mV/div and above $\pm$ 2.0% for 2 mV/div $\pm$ 2.5% for 1 mV/div				
Channel-to-Channel Isolation (Any Two Channels at Equal Vertical Scale)	$\geq$ 100:1 at $\leq$ 100 MHz and $\geq$ 30:1 at >100 MHz up to the rated BW				

### Offset Range

Range	1 M $\Omega$	50 $\Omega$ , 75 $\Omega$
1 mV/div to 99.5 mV/div	$\pm$ 1 V	$\pm$ 1 V
100 mV/div to 995 mV/div	$\pm$ 10 V	$\pm$ 5 V
1 V/div	$\pm$ 100 V	$\pm$ 5 V
1.01 V/div to 10 V/div	$\pm$ 100 V	NA

### Vertical System Digital Channels

Characteristic	All MSO3000 Models
Input Channels	16 Digital (D15 to D0)
Thresholds	Threshold per set of 8 channels
Threshold Selections	TTL, CMOS, ECL, PECL, User Defined
User-defined Threshold Range	-15 V to +25 V
Maximum Input Voltage	-20 V to +30 V
Threshold Accuracy	$\pm$ (100 mV +3% of threshold setting)
Maximum Input Dynamic Range	50 V <sub>pk-pk</sub> (threshold setting dependent)
Minimum Voltage Swing	500 mV <sub>pk-pk</sub>
Input Impedance	101 k $\Omega$
Probe Loading	8 pF
Vertical Resolution	1 bit

### Horizontal System Analog Channels

Characteristic	All MSO3000 Models All DPO3000 Models
Maximum Sample Rate (all channels)	2.5 GS/s
Maximum Record Length (all channels)	5 Mpoints
Maximum Duration of Time Captured at Highest Sample Rate (all channels)	2 ms
Time-base Range (s/div)	1 ns to 1000 s
Time-base Delay Time Range	-10 divisions to 5000 s
Channel-to-Channel Deskew Range	$\pm$ 100 ns
Time-base Accuracy	$\pm$ 10 ppm over any $\geq$ 1 ms interval

### Horizontal System Digital Channels

Characteristic	All MSO3000 Models
Maximum Sample Rate (Main, all channels)	500 MS/s (2 ns resolution)
Maximum Record Length (Main, all channels)	5 Mpoints
Maximum Sample Rate (MagniVu, all channels)	8.25 GS/s (121.2 ps resolution)
Maximum Record Length (MagniVu, all channels)	10 kpoints centered on the trigger
Minimum Detectable Pulse Width	2.0 ns
Channel-to-Channel Skew	500 ps typical