

**TUTORIAL PARA LA PROGRAMACION DE LAS TARJETAS NI-USB  
6008/6009 CON LA TOOLBOX DE AQUISICION DE DATOS DE MATLAB**

**KARINA CARREÑO JAIMES  
NATALIA GABRIEL JIMENEZ**

**UNIVERSIDAD PONTIFICIA BOLIVARIANA  
ESCUELA DE INGENIERIAS Y ADMINISTRACION  
FACULTAD DE INGENIERIA ELECTRONICA  
BUCARAMANGA  
2009**

**TUTORIAL PARA LA PROGRAMACION DE LAS TARJETAS NI-USB  
6008/6009 CON LA TOOLBOX DE AQUISICION DE DATOS DE MATLAB**

**KARINA CARREÑO JAIMES  
NATALIA GABRIEL JIMENEZ**

**Trabajo de Grado para optar al título de  
Ingeniero Electrónico**

**Director:  
EDGAR BARRIOS URUEÑA  
Ing. MSc. Eléctrico**

**UNIVERSIDAD PONTIFICIA BOLIVARIANA  
ESCUELA DE INGENIERIAS Y ADMINISTRACION  
FACULTAD DE INGENIERIA ELECTRONICA  
BUCARAMANGA**

**2009**

**Nota de aceptación**

---

---

---

---

Firma del jurado

---

Firma del jurado

Bucaramanga, 26 de Marzo de 2009

A Dios por darme la oportunidad de existir y la fuerza para lograr todas mis metas y mis sueños.

A mis padres, porque sin excusa alguna se han sacrificado para formarme y porque con amor, confianza, amistad y esfuerzo me han brindado apoyo para hacer posible la culminación de mi carrera profesional.

A mi hermana Carolina, por darme un ejemplo a seguir, por todo su apoyo y por todos sus regaños que me han servido para ser siempre mejor.

A mi novio "pirri", por su ánimo en todo momento, por acompañarme a la universidad todos los días de vacaciones y por todo su amor.

Gracias

Natalia Gabriel Jiménez

A Dios por permitirme estar viva para escribir esto y por la fuerza espiritual q me ha brindado, sin esto no hubiese sido posible continuar.

A mis padres profundamente por la paciencia, confianza, apoyo y amor incondicional q constantemente me han proporcionado sin importar cada una de mis decisiones. Ellos son la base de mi vida y a con quienes estaré agradecida eternamente

A mis hermanas que me han acompañado en los buenos y malos momentos y siempre me animaron durante toda la realización de este proyecto.

A mis amigos, q estuvieron conmigo los primeros semestres y a los q están conmigo ahora por todos los momentos q disfrutamos, por sus consejos, deseos y buenas intenciones hacia mí y sobre todo por hacer de mi paso por la universidad algo único e inolvidable.

A aquellas personas que, en los últimos meses, con su presencia me han aligerado del peso que suponen mis obligaciones.

Karina Carreño Jaimes

## **AGRADECIMIENTOS**

Las autoras expresamos agradecimientos a:

Al Ingeniero Edgar Barrios Urueña, director del proyecto, por su orientación y porque nos dio la libertad y confianza necesaria para la realización del proyecto.

Al Ingeniero Juan Carlos Villamizar Rincón porque nos brindó su ayuda en situaciones de dificultad.

A la universidad por todos los años de una formación profesional integra.

## INDICE DE CONTENIDO

	Pág.
1 INTRODUCCION .....	3
2 OBJETIVOS .....	4
2.1 Objetivo General.....	4
2.2 Objetivos específicos .....	4
3 MARCO TEORICO .....	5
3.1 SISTEMA DE ADQUISICIÓN DE DATOS .....	5
3.1.1 Descripción general.....	5
3.2 MUESTREO .....	10
3.2.1 Hardware de escaneo: .....	11
3.2.2 Hardware de Muestreo y Captura simultaneas:.....	11
3.2.3 ¿Cómo son adquiridas las muestras?.....	12
3.2.4 Configuración de canal.....	13
3.3 MEDICIONES CON CALIDAD .....	15
3.3.1 ¿Qué se está midiendo?.....	15
3.4 TOOLBOX DE ADQUISICION DE DATOS .....	25
3.4.1 Componentes de la Toolbox .....	25
3.4.2 Motor de adquisición de datos .....	27
3.4.3 Adaptadores de Hardware:.....	29
3.5 TARJETA NI USB-6008/6009 .....	30
3.5.1 Diferencias entre las tarjetas NI-USB 6008 y 6009 .....	30

3.5.2	Software de la Tarjeta .....	30
3.5.3	Hardware.....	31
3.5.4	Connector de Entradas y Salidas.....	32
3.5.5	Entradas Análogas.....	34
3.5.6	Trigger digital .....	37
3.5.7	Salidas Análogas.....	37
3.5.8	Entradas y Salidas Digitales:.....	38
3.5.9	Protecciones en las entradas y salidas de la Tarjeta.....	39
4	METODOLOGIA Y DESCRIPCION DEL PROYECTO .....	40
4.1	Estudio de la Toolbox y la Tarjeta .....	40
4.2	Diseño del Tutorial .....	41
4.2.1	Selección del contenido .....	41
4.2.2	Diseño de la estructura .....	42
4.3	ELABORACIÓN DEL TUTORIAL.....	44
4.4	MÓDULO DE PRUEBAS DIGITAL Y ANÁLOGO.....	45
4.4.1	Módulo digital .....	46
4.4.2	Módulo análogo .....	47
4.5	Laboratorios .....	49
5	RESULTADOS .....	50
6	CONCLUSIONES.....	52
7	RECOMENDACIONES .....	54
8	REFERENCIAS .....	55
9	BIBLIOGRAFIA.....	56

## INDICE DE FIGURAS

	Pág.
Figura 1. Componentes de un Sistema de Adquisición de Datos .....	6
Figura 2 . Hardware de Adquisición de Datos.....	7
Figura 3. Hardware de escaneo.....	11
Figura 4. Entrada Diferencial .....	14
Figura 5. Entrada de Nodo Simple.....	15
Figura 6. Exactitud y precisión .....	17
Figura 7. Teorema de Nyquist.....	23
Figura 8. Muestreo de una Señal.....	24
Figura 9. Componentes de la Toolbox .....	26
Figura 10. Flujo de Entrada de Datos .....	28
Figura 11. Flujo de Salida de Datos.....	29
Figura 12. Hardware de la tarjeta.....	31
Figura 13. Circuito de Entrada Análoga .....	34
Figura 14. Conexión de una señal de voltaje diferencial.....	35
Figura 15. Ejemplo de una Medición Diferencial de 20V .....	36
Figura 16. Señal con Exceso de mas de +10V .....	36
Figura 17. Conexión de una Señal Referenciada de Nodo Simple .....	37
Figura 18. Circuito de Salida Análoga.....	38
Figura 19. Conexión de una Carga a la Salida Análoga .....	38
Figura 20. Ejemplos de conexión de Entradas y Salidas Digitales .....	39
Figura 21. Estructura de una GUI del tutorial.....	43
Figura 22. GUI con Acceso a Temas Complementarios .....	44
Figura 23. GUI de Tema extra .....	45
Figura 24. Cirucitos del Módulo Digital .....	47
Figura 25. Circuito del generador de onda sinusoidal.....	48
Figura 26. Diseño del circuito impreso en EAGLE .....	48
Figura 27. GUI Inicial del Tutorial.....	50
Figura 28. GUI de un Ejemplo tipo DEMO .....	51



## INDICE DE TABLAS

	Pág.
Tabla 1. Relación entre Rango de Entrada, Ganancia, y precisión.....	19
Tabla 2. Diferencias entre las Tarjetas NI-USB 6008 y 6009.....	30
Tabla 3. Terminal Análogo.....	32
Tabla 4. Terminal Digital.....	33

## **INDICE DE ANEXOS**

ANEXO 1. TUTORIAL PARA LA PROGRAMACION DE LAS TARJETAS NI-USB 6008/6009 CON LA TOOLBOX DE ADQUISICION DE DATOS. GUIA RAPIDA DE USUARIO .....	58
ANEXO 2. LISTA DE FUNCIONES Y PROPIEDADES PARA LA TOOLBOX DE ADQUISICIÓN DE DATOS DE MATLAB.....	67
ANEXO 3. EJEMPLOS RESUELTOS PRESENTADOS EN EL TUTORIAL.	77

## **RESUMEN**

**TÍTULO:** TUTORIAL PARA LA PROGRAMACION DE LAS TARJETAS NI-USB 6008/6009 CON LA TOOLBOX DE AQUISICION DE DATOS DE MATLAB.

**AUTORES:** KARINA CARREÑO JAIMES  
NATALIA GABRIEL JIMENEZ

**FACULTAD:** INGENIERÍA ELECTRONICA

**DIRECTOR:** MSc. EDGAR BARRIOS URUEÑA

Teniendo en cuenta la importancia de la adquisición de datos en sistemas donde implica el control del mismo, el análisis y la visualización de datos se quiere con este proyecto de grado dar un conocimiento general acerca de la programación de las tarjetas NI-USB 6008/6009 para extraer y entregar datos digitales y análogos con la Toolbox de Adquisición de Datos de MATLAB.

El proyecto consiste en un tutorial donde se presenta toda la información teórica acerca de las tarjetas y de la Toolbox, explicación de las funciones y propiedades básicas y necesarias para la programación, ejemplos demostrativos y laboratorios para desarrollar.

El tutorial cuenta un hardware llamado “Módulo de pruebas Digital y Análogo”, diseñado para mostrar en él, los ejemplos desarrollados y los laboratorios. Este permite tomar entradas y salidas digitales, una entrada análoga a la tarjeta de un generador de una señal seno y una cuadrada.

## **ABSTRACT**

**TITLE:** NI-USB 6008/6009 BOARD PROGRAMMING TUTORIAL WITH MATLAB DATA ACQUISITION TOOLBOX.

**AUTHOR:** KARINA CARREÑO JAIMES  
NATALIA GABRIEL JIMENEZ

**FACULTY:** ELECTRONIC ENGINEERING.

**DIRECTOR:** MSc. EDGAR BARRIOS URUEÑA

Given the importance of the data acquisition in systems where is implicated the control of itself, the analysis and visualization of data, with this graduation project, is wanted to give a general knowledge about the programming of the NI-USB 6008/6009 boards, to extract and submit digital and analog data with the MATLAB's Data Acquisition Toolbox.

The project is a tutorial wich is presents all the theoretical information of the board and the Toolbox, explanation of the basic functions and properties necessary for the programming, demonstrative examples and laboratories for development.

The tutorial counts with a hardware module called "Digital and Analog Test Module", designed to show the developed examples and the laboratories. This module can take digital inputs and digital outputs, an analog input to the board from a generator of a sinus signal and a square signal inputs and digital outputs, an analog input to the board from a generator of a sinus signal and a square signal.

## 1 INTRODUCCION

En los últimos años la Tarjeta de Adquisición de Datos USB 6008/6009 de National Instruments ha sido muy utilizada por la Universidad Pontificia Bolivariana en los laboratorios de control, proyectos de grados y hasta ha sido incluida como herramienta básica en el diplomado de LABVIEW. En todos los casos el software de programación es LABVIEW, por lo tanto el objetivo del tutorial para la programación de la tarjeta con la Toolbox de Adquisición de Datos es brindar otro medio para su uso, explicando claramente sus características, la descripción de la Toolbox y las instrucciones básicas para realizar una sesión de adquisición.

La Toolbox de adquisición de datos por medio de controladores da soporte a varios tipos de hardware de adquisición incluyendo las tarjetas de la National Instruments, utiliza el lenguaje C de MATLAB y exige la configuración de propiedades para el hardware y la creación de un objeto ya sea de entrada o salida digital o análoga, el cual permite el acceso y control del comportamiento del hardware según las necesidades de la aplicación de adquisición.

El tutorial tiene una metodología de enseñanza elemental, donde se toma desde lo básico hasta lo general, para dar un entendimiento claro y conciso, al final se pretende que el alumno o la persona que lo realice sea capaz de desarrollar unas prácticas planteadas con el módulo de servomecanismo presente el laboratorio y así tome la habilidad de usar la tarjeta con la Toolbox en aplicaciones donde se necesite la manipulación de datos.

## 2 OBJETIVOS

### **2.1 Objetivo General**

Crear un software interactivo para la enseñanza del manejo y la programación de la tarjeta NI-USB 6008/6009 de National Instruments utilizando la Toolbox de Adquisición de Datos de MATLAB para adquirir y entregar datos de señales análogas y digitales.

### **2.2 Objetivos específicos**

- Conocimiento de las características, funcionamiento y modos de uso de las tarjetas NI-USB 6008/6009.
- Conocimiento de las funciones, instrucciones y aplicaciones de la Toolbox de Adquisición de datos de MATLAB para la obtención y entrega de datos por medio de las tarjetas NI-USB 6008/6009.
- Plantear y desarrollar aplicaciones que muestren el uso de las tarjetas NI-USB 6008/6009 con la Toolbox de adquisición de datos para adquisición y entrega de datos análogos y digitales.
- Desarrollar un tutorial dirigido al estudiante que le facilite el entendimiento general de las tarjetas y de las aplicaciones con la Toolbox.

## 3 MARCO TEORICO

### 3.1 SISTEMA DE ADQUISICIÓN DE DATOS [1]

#### 3.1.1 Descripción general

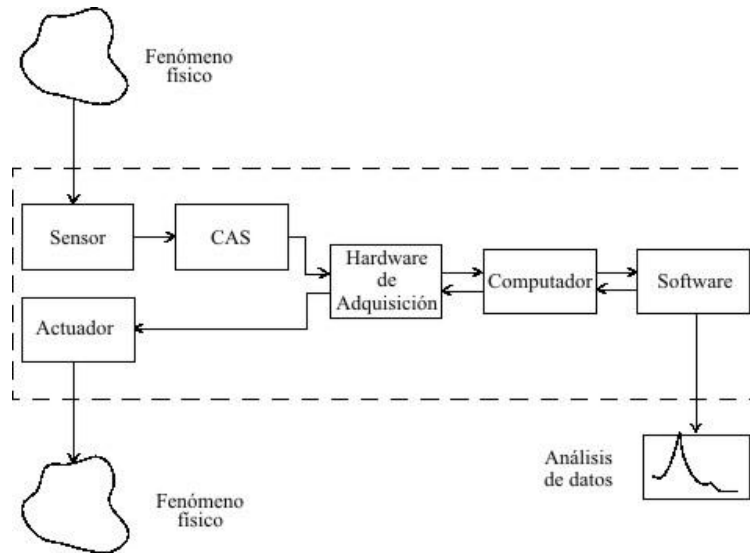
El objetivo de cualquier sistema de adquisición de datos es proporcionar las herramientas y recursos necesarios para tomar señales físicas y convertirlas en datos que posteriormente se puedan procesar y mostrar.

Un sistema de adquisición de datos se podría tomar como un grupo de hardware y software que permiten interactuar con el mundo real, consta de estos componentes:

- **Hardware de Adquisición:** Es el corazón de cualquier sistema de adquisición de datos. La función principal es hacer la conversión de señales análogas a señales digitales y señales digitales a análogas. Conversión A/D y D/A.
- **Sensores y Actuadores (Transductores):** Un transductor es un dispositivo que convierte un tipo de energía de entrada en una energía de salida de otra forma.
- **Acondicionador de señal:** Las señales de los sensores a menudo son incompatibles con el hardware de adquisición de datos, y para superar esto las señales deben ser acondicionadas. Por ejemplo, las señales podrían ser amplificadas o volverlas en señales sin componentes de frecuencias indeseadas. Las señales de salida también pueden ser acondicionadas.
- **Computador:** Proporciona un procesador, un sistema de reloj, un bus de datos, memoria y espacio en el disco para almacenar datos.
- **Software:** Permite el intercambio de información entre el computador y el hardware.

Los componentes y la relación entre cada uno, se presenta a continuación en la figura 1.

**Figura 1.** Componentes de un Sistema de Adquisición de Datos



Fuente: Data Acquisition Toolbox™ 2, User's Guide, MathWorks.

La figura 1, describe las dos características importantes de un sistema de adquisición de datos:

- Las entradas son la salida de un sensor, convertidas en datos que el computador puede leer y analizar para extraer información significativa.
- Los datos desde el computador son convertidos en una señal análoga y son la salida a un actuador.

### 3.1.1.1 Hardware de adquisición de datos

El hardware de adquisición de datos puede instalarse de manera interna, directamente en una ranura de expansión del computador o de manera externa, que se conecta al computador a través de un cable.

Se caracteriza por los subsistemas que éste posee. Un subsistema es un componente del hardware que realiza una tarea específica. Los más comunes son:

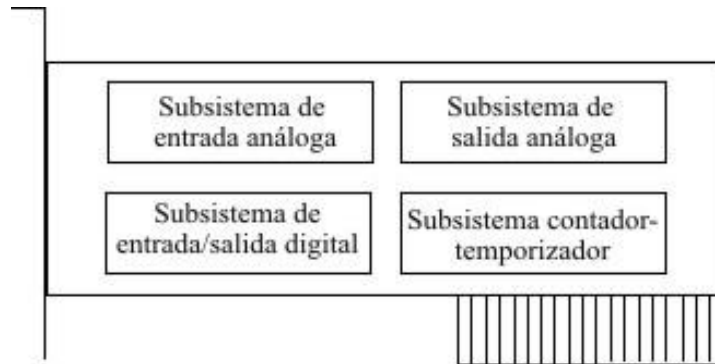
- Entradas análogas
- Salidas análogas
- Entradas/salidas digitales



- Contador/Temporizador

Un Hardware consta de múltiples subsistemas, como se muestra a continuación en la figura 2,

**Figura 2 . Hardware de Adquisición de Datos**



Fuente: Data Acquisition Toolbox™ 2, User's Guide, MathWorks.

- **Subsistemas de entrada analógica:** Convierte las señales analógicas reales desde el sensor en señales (bits) que pueden ser leídas por el PC.
- **Subsistemas de salida analógica:** Convierten datos digitales almacenados en el computador en una señal analógica real. Realizan la conversión inversa de los subsistemas de entrada analógica. Estos subsistemas son llamados también subsistemas AO, convertidores D/A o DACs.
- **Subsistemas de entradas/salidas digitales:** Están diseñados para valores de entradas y salidas digitales (valores lógicos). Estos valores suelen ser manejados ya sea como bits simples, líneas de bits o como un puerto que normalmente consta de ocho líneas de bits.
- **Subsistemas de contador/temporizador:** Son usados para eventos de conteo, medición de frecuencia y periodo, y para la generación de tren de pulsos.

### 3.1.1.2 Sensores

Un sensor convierte una variable física en una variable que es usada como entrada en el hardware de adquisición. Hay dos tipos de sensores según la salida que produce: sensor digital y sensor analógico.

El sensor digital produce en la salida una señal digital que es la representación de la señal de entrada, mide valores discretos en tiempo discreto. El nivel lógico de la salida de los sensores digitales debe ser compatible con el receptor digital. Algunos niveles estándares de niveles lógicos son los de TTL (transistor-transistor logic) y ECL (emitter-coupled logic).

El sensor análogo produce una señal de salida análoga que es directamente proporcional a la señal medida, de magnitudes continuas en el tiempo. Las variables más típicas tal como la temperatura, presión y aceleración son medidas con sensores análogos.

### 3.1.1.3 Acondicionamiento de señal

Las señales de los sensores normalmente son incompatibles con el hardware de adquisición de datos. Para superar esa incompatibilidad, la señal debe ser acondicionada. El tipo de acondicionamiento depende del sensor que se utilice. Las formas más comunes de acondicionar una señal son:

- **Amplificación:** Las señales de niveles bajos, menores de 100mV necesitan ser amplificadas. Los niveles altos, también podrían requerir de amplificación dependiendo del rango de entrada de la medición. La amplificación permite reducir y hacer uso del mayor rango y así aumentar la resolución de la medición.
- **Filtrado:** Elimina ruidos indeseados de la señal. El filtro de ruido es usado en señales que varían lentamente, como la temperatura, para atenuar las señales de alta frecuencia, que pueden reducir la precisión de la medición. Las señales de rápida variación como la vibración, requieren de un tipo de filtro diferente, conocido como filtro de antialiasing, que eliminan las frecuencias más altas, que podrían dar lugar a medidas erróneas.
- **Aislamiento eléctrico:** Evita los transitorios de señales de alto voltaje.
- **Multiplexación:** Permite enviar distintas señales sobre un mismo canal.

Se debe tener en cuenta que la señal de conmutación de un multiplexor tenga tiempo suficiente para realizar esto.

#### 3.1.1.4 Equipo

Ofrece un procesador, un sistema de reloj, un bus para la transferencia de datos, memoria y espacio en el disco para almacenar datos.

El procesador controla la rapidez de datos aceptados por el convertidor. El sistema de reloj proporciona información acerca del tiempo de los datos adquiridos. Sabiendo la lectura del sensor no es **suficiente, se** necesita saber cuándo se produjo la medición.

Los datos son transferidos desde el hardware al sistema de memoria a través de la memoria dinámica de acceso (DMA) o interrupciones. La DMA es controlada por hardware y por lo tanto es extremadamente rápida. La tasa de adquisición de datos está determinada por la arquitectura del sistema del bus.

#### 3.1.1.5 Software

Independientemente del hardware que use, en una sesión de adquisición se necesita enviar información al hardware y recibir información desde el mismo. Se envía información al hardware tal como la tasa de muestreo, y recibe información desde el hardware tal como, mensajes de estado y mensajes de error. Se podría necesitar también suministrar al hardware con información, para que éste pueda integrarse con otro hardware y con recursos del computador. Este intercambio de información se realiza a través del software.

Hay dos tipos de software:

- Controladores
- Software de aplicación

**Controladores:** Los controladores permiten el acceso y el control de las capacidades del hardware. Entre otras cosas, los controladores básicos permiten:

- Traer datos y obtener datos desde y fuera de la tarjeta.
- Control de la tasa a la cual los datos son adquiridos.
- Integrar el hardware de adquisición de datos con los recursos del computador.
- Integrar el hardware de adquisición de datos con el hardware de acondicionamiento de señales.
- Acceso múltiple a subsistemas de la tarjeta
- Acceso múltiple a tarjetas de adquisición de datos

**Software de aplicación:** Proporciona una interfaz al controlador.

El software de aplicación básico permite:

- Reporte pertinente sobre la información tal como el número de muestras adquiridas.
- Generar eventos
- Administrar los datos almacenado en la memoria del computador
- Condicionar una señal
- Graficar los datos adquiridos

### **3.2 MUESTREO [1]**

Durante el muestreo se realiza una captura instantánea de la señal en tiempo discreto. Para la mayoría de las aplicaciones, el intervalo de tiempo entre las muestras se mantiene constante (por ejemplo, muestra cada milisegundo).

Para muchos de los convertidores digitales, la toma de muestras es realizada por un circuito de muestreo y captura (S/H), consiste en una señal de amortiguación seguido por un interruptor electrónico conectado a un condensador.

El funcionamiento de un circuito S/H sigue estos pasos:

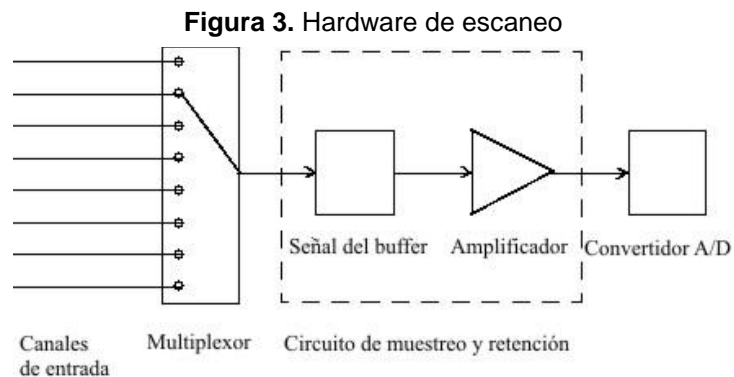
1. En un determinado instante de muestreo, el conmutador conecta el buffer y un condensador a la entrada.
2. El condensador se carga a la tensión de entrada.
3. La carga se mantiene hasta que el convertidor A/D digitaliza la señal.
4. Para múltiples canales (multiplexado) por un convertidor A/D, los pasos anteriores se repiten para cada canal de entrada.
5. Todo el proceso se repite para el próximo instante de muestreo.

El hardware puede ser dividido en dos categorías principales basadas en cómo las señales son muestreadas: Hardware de muestreo, la cual muestrea señales secuencialmente, y hardware de muestreo y captura simultánea, la cual muestrea todas las señales al mismo tiempo.

### 3.2.1 Hardware de escaneo:

El Hardware de escaneo muestrea una única señal de entrada, convierte la señal a un valor digital y luego, repite el proceso para cada canal de entrada utilizado. En otras palabras, cada canal de entrada es muestreado secuencialmente.

Como se muestra en la figura 3, la mayoría de dispositivos de adquisición de datos tiene un convertidor A/D que es multiplexado a múltiples canales de entrada.



Fuente: Data Acquisition Toolbox™ 2, User's Guide, MathWorks.

Por lo tanto, si se usa múltiples canales, estos canales no pueden ser muestreados simultáneamente y existe un desfase de tiempo entre los canales muestreados consecutivos.

La máxima tasa de muestreo por canal esta dado por la fórmula:

$$\text{Máxima tasa de muestreo} = \frac{\text{Máxima tasa del hardware}}{\text{Número de canales escaneados}}$$

### 3.2.2 Hardware de Muestreo y Captura simultaneas:

El hardware de muestreo simultáneo y retención (SS/H) muestrea todas las señales de entrada al mismo tiempo y retiene los valores hasta que el convertidor

digitaliza todas las señales. En algunos sistemas puede haber un convertidor para cada canal.

Por ejemplo, se necesita medir simultáneamente la aceleración de múltiples acelerómetros para determinar la vibración de un dispositivo bajo prueba. Para ello, se debe usar un hardware SS/H, porque este no tiene un desfase de tiempo entre los canales muestreados consecutivos. En general, puede ser que se necesite utilizar un hardware SS/H si la señal del sensor cambia significativamente en un momento en que es inferior al desfase de tiempo entre los canales muestreados consecutivos, o si se necesita usar una función de transferencia o llevar a cabo una correlación del dominio de frecuencia.

### 3.2.3 ¿Cómo son adquiridas las muestras?

Las muestras son adquiridas a partir de un subsistema de entrada análoga a una tasa específica por un reloj. Como cualquier sistema temporizado, los relojes de adquisición de datos se caracterizan por su resolución y precisión. La resolución del tiempo se define como el intervalo de tiempo más pequeño que se puede medir de manera precisa. La precisión del tiempo es afectada por el JITTER del reloj. El JITTER crece cuando un reloj produce valores diferentes para un intervalo de tiempo dado.

Para cualquier sistema de adquisición de datos, existen tres fuentes de reloj que puede utilizar: el reloj incorporado de la tarjeta de adquisición de datos, el reloj del computador, o un reloj externo. La Toolbox de Adquisición de Datos soporta todo el software de estas fuentes de reloj, dependiendo de los requisitos del hardware:

- **Reloj de la tarjeta.** El reloj incluido en la tarjeta o *reloj interno* suele ser un chip temporizador en la tarjeta de hardware que está programado para generar un pulso de corriente en la tasa deseada. El reloj de la tarjeta en general tiene una alta precisión y bajo JITTER en comparación con el reloj del computador. Siempre se debe usar el reloj de la tarjeta, cuando la tasa de muestreo es alta, y cuando se necesite un intervalo de tiempo fijo entre muestras.

- **Reloj del computador.** El reloj del computador o reloj de software se utiliza para las tarjetas que no poseen un reloj interno. El reloj del computador es menos preciso y tiene más JITTER que el reloj que está incluido en la tarjeta, y se limita generalmente a tasas de muestreo por debajo de 500 Hz.

- **Reloj externo.** Un reloj externo se utiliza a menudo cuando la tasa de muestreo es baja y no constante.

### 3.2.4 Configuración de canal

Los canales de entrada pueden ser configurados de una de estas dos maneras:

- Modo Diferencial
- Modo de Nodo Simple

La elección de la configuración del canal de entrada podría depender de si la señal de entrada es referenciada a tierra o es una señal flotante.

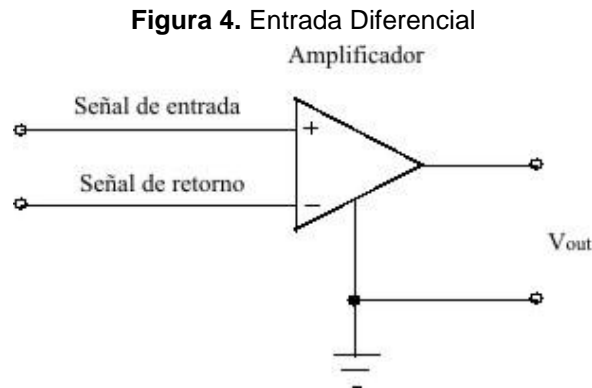
Una señal flotante usa una referencia aislada de la tierra y no está conectado a la tierra del sistema. Como resultado de ello, la señal de entrada y el dispositivo de hardware no están conectados a una referencia común, lo que puede causar que la señal de entrada exceda el rango válido del hardware. Para evitar este problema, se debe conectar la señal a la tierra de la tarjeta del dispositivo. Ejemplo de fuentes de señales flotantes incluyen termocuplas y baterías.

Una señal referenciada a tierra está conectada a la misma referencia del dispositivo. Como resultado de ello, la señal de entrada y el hardware están conectados a una referencia común.

#### - **Entrada diferencial:**

Cuando se configura el hardware para entrada diferencial, hay dos cables de señal asociada con cada señal de entrada – uno para la señal de entrada y el otro para la señal de referencia (retorno). La medición es la diferencia de tensión entre los dos cables, los cuales ayudan a reducir el ruido y cualquier voltaje que es común a ambos cables.

Como se muestra en la figura 4, la señal de entrada está conectada a la entrada positiva del amplificador y el retorno de señal está conectado a la entrada negativa del amplificador. El amplificador tiene un tercer conector que permite que estas señales sean referenciadas a tierra.



Fuente: Data Acquisition Toolbox™ 2, User's Guide, MathWorks.

National Instruments recomienda que se use entradas diferenciales bajo cualquiera de estas condiciones:

- La señal de entrada es de bajo nivel (menos de 1 voltio).
- Las puntas que conecta a la señal es mayor que 10pies.
- La señal de entrada requiere un punto de referencia a tierra separado o señal de retorno.
- Las puntas de la señal se mueven a través de un medio ruidoso.

- **Entradas de nodo simple:**

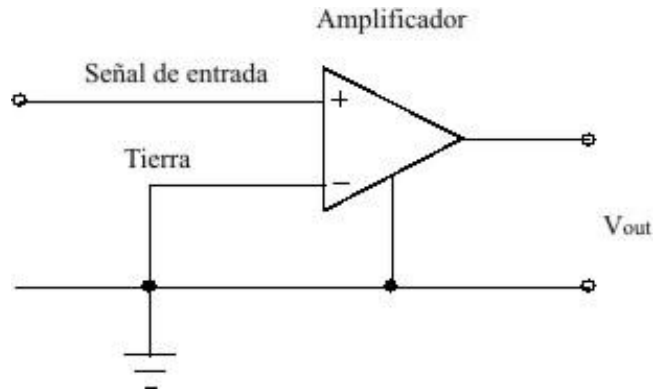
Cuando se configura el hardware para una entrada de simple nodo, hay un cable de señal asociado a cada señal de entrada, y cada señal de entrada está conectada a la misma tierra. Las mediciones de simple nodo son más susceptibles al ruido que las mediciones diferenciales debido a las diferencias en la trayectoria de la señal.

Como se muestra a continuación en la figura 5, la señal de entrada está conectada



a la entrada positiva del amplificador y la tierra está conectada a la entrada negativa del amplificador.

**Figura 5.** Entrada de Nodo Simple



Fuente: Data Acquisition Toolbox™ 2, User's Guide, MathWorks.

National Instruments recomienda usar entradas de simple nodo bajo cualquiera de estas condiciones:

- La señal de entrada es de alto nivel (más de 1 voltio).
- Las puntas que conecta a la señal es menor que 10pies.
- La señal de entrada puede compartir un punto de referencia común con otras señales.

### **3.3 MEDICIONES CON CALIDAD [1]**

#### **3.3.1 ¿Qué se está midiendo?**

Para la mayoría de las aplicaciones de adquisición de datos, es necesario medir la señal producida por el sensor a una tasa específica.

En muchos casos, la señal del sensor es un nivel de voltaje que es proporcional a los fenómenos físicos de interés (por ejemplo, temperatura, presión o aceleración).

Si se está midiendo gradualmente cambios de fenómenos como la temperatura, una tasa de muestreo lenta generalmente es suficiente. Si se está midiendo cambios rápidos (dinámica) de fenómenos como las vibraciones o mediciones acústicas, se necesita una tasa de muestreo rápida.

Para hacer mediciones de alta calidad, se debe seguir estas reglas:

- Maximizar la precisión y exactitud
- Minimizar el ruido
- comparar el rango del sensor de acuerdo al rango del A / D.

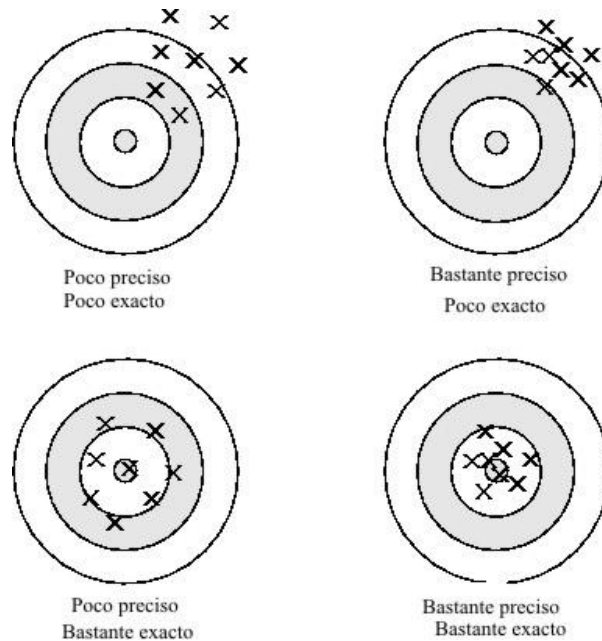
### **3.3.1.1 Exactitud y precisión**

Cada vez que se adquieren los datos medidos, se debe hacer todo lo posible para maximizar la exactitud y la precisión. La calidad de la medición depende de la exactitud y la precisión de todo el sistema de adquisición de datos, y puede ser limitada por factores tales como la resolución de la tarjeta o el ruido ambiental.

En términos generales, la exactitud de una medición determina qué tan cerca llegan a ser las mediciones al valor real. Por lo tanto, indica la exactitud del resultado. La precisión de la medición refleja que tan exacto es un resultado determinado sin importar qué signifique el resultado. La precisión relativa indica la incertidumbre en una medición como una fracción del resultado.

Por ejemplo, suponer que se mide la parte superior de una mesa con un metro y se encuentra que su longitud es de 1.502 metros. Este número indica que el metro (y los ojos) pueden determinar distancias de por lo menos un milímetro. Bajo la mayoría de las circunstancias, se considera que esto es una medición bastante precisa en relación con una precisión de alrededor de 1/1500. Sin embargo, supongamos que se realiza la medición de nuevo y se obtiene un resultado de 1,510 metros. Después de una cuidadosa consideración, se descubre que la técnica inicial para la lectura del metro era defectuosa porque no ha leído directamente por encima. Por lo tanto, la primera medición no era exacta. La precisión y la exactitud se ilustran en la figura 6:

**Figura 6.** Exactitud y precisión



Fuente: Data Acquisition Toolbox™ 2, User's Guide, MathWorks.

Para los subsistemas de entrada análoga, la exactitud suele ser limitada por errores de calibración, mientras que la precisión es normalmente limitada por el convertidor A/D.

### 3.3.1.2 Exactitud:

La exactitud se define como la relación entre una cantidad medida y el verdadero valor de esa cantidad. Cada componente que aparece en la trayectoria de la señal análoga afecta la exactitud del sistema y el rendimiento. La exactitud total del sistema está dada por el componente con la peor exactitud.

Para el hardware de adquisición de datos, la exactitud suele expresarse como un porcentaje o una fracción del bit menos significativo (LSB). Bajo circunstancias ideales, la exactitud de la tarjeta es de  $\pm 0.5 \text{ LSB}$ . Por lo tanto, un convertidor de 12 bits tiene sólo 11 bits utilizables.

Muchas tarjetas incluyen un amplificador de ganancia programable, que se encuentra justo antes de la entrada del convertidor. Para evitar que la exactitud del

sistema sea degradada, la exactitud y la linealidad de la ganancia deben ser mejor que la del convertidor A/D. La exactitud específica de una tarjeta también se ve afectada por la tasa de muestreo y el tiempo de resolución del amplificador. El tiempo de resolución se define como el tiempo necesario para que el amplificador de instrumentación realice su función para una exactitud determinada. Para mantener una exactitud plena, la salida del amplificador debe fijar a un nivel determinado para la magnitud de  $0.5 \text{ LSB}$  antes de la próxima conversión, y es sobre el orden de varias décimas de milisegundo de la mayoría de las tarjetas.

El tiempo de resolución es una función de la tasa de muestreo y el valor de ganancia. Altas tasas, configuraciones de ganancia altas requieren ajustes de tiempo más largos mientras tasas bajas, configuraciones de ganancia bajas requieren ajustes de tiempo más cortos.

### **3.3.1.3 Precisión:**

El número de bits usados para representar una señal analógica determina la precisión (resolución) del dispositivo. Cuantos más bits proporcione la tarjeta, más precisa será la medición. Una alta precisión, alta resolución del dispositivo divide el rango de entrada en más divisiones permitiendo con eso percibir valores de voltaje más pequeños. Una baja precisión, baja resolución del dispositivo divide el rango de entrada en un menor número de divisiones aumentando con eso los valores de tensión detectables.

La precisión general del sistema de adquisición de datos es determinada por el convertidor A/D, y se especifica por el número de bits utilizados para representar la señal análoga. La mayoría de las tarjetas usan 12 o 16 bits. La precisión de la medición está dada por:

$$\text{Precisión} = \text{una parte en } 2^{\text{número de bits}}$$

La precisión en voltios está dada por:

$$\text{Precisión} = \frac{\text{Rango de voltaje}}{2^{\text{número de bits}}}$$

Por ejemplo, si se está usando un convertidor de 12 bits A/D configurado para un rango de 10 voltios, entonces,

$$\text{Precisión} = \frac{10\text{voltios}}{2^{12}}$$

Esto significa que el convertidor puede detectar diferencias de voltaje en el nivel de 0,00244 voltios (2,44 mV).

### 3.3.1.4 ¿Cómo están relacionados la ganancia, el rango y las mediciones de precisión?

Cuando se configura el rango de entrada y la ganancia del subsistema de entrada analógica, el resultado final debería maximizar la resolución de la medición y reducir al mínimo las posibilidades de una condición de sobre paso del rango. El rango de entrada actual está dado por la fórmula:

$$\text{Rango de entrada actual} = \frac{\text{Rango de entrada}}{\text{Ganancia}}$$

La relación entre la ganancia, rango de entrada real, y la precisión para una señal unipolar y bipolar teniendo un rango de entrada de 10 V, se muestra a continuación en la tabla 1:

**Tabla 1.** Relación entre Rango de Entrada, Ganancia, y precisión

Rango de entrada	Ganancia	Rango de entrada actual	Precisión (12Bits A/D)
0 a 10V	1.0	0 a 10V	2.44mV
	2.0	0 a 5V	1.22mV
	5.0	0 a 2V	0.488mV
	10.0	0 a 1V	0.244mV
-5 a 5V	0.5	-10 a 10V	4.88mV
	1.0	-5 a 5V	2.44mV
	2.0	-2.5 a 2.5V	1.22mV
	5.0	-1 a 1V	0.448mV
	10.0	-0.5 a 0-5V	0.244mV

Como se muestra en la tabla 1, la ganancia afecta la precisión de la medida. Si se selecciona una ganancia que disminuye el rango de entrada actual, entonces la precisión incrementa. Contrario a esto, si se selecciona una ganancia que incrementa el rango de entrada actual, entonces la precisión disminuye. Esto es porque el rango de entrada actual varía pero el número de bits usados por el convertidor A/D se mantiene fijo.

---

**Nota:** Con Toolbox de Adquisición de Datos no se tiene que especificar el rango y la ganancia. En lugar de esto, simplemente se especifica el rango de entrada deseado.

---

### **3.3.1.5 Ruido**

El ruido es considerado como una medida que no es parte del fenómeno físico de interés. El ruido puede ser generado dentro de los componentes eléctricos de la entrada del amplificador (ruido interno), o éste puede ser adherido a la señal como mientras es transferida por los cables de entrada (ruido externo). Las técnicas que se pueden reducir para reducir el efecto del ruido son descritas a continuación:

- **Eliminación de ruido interno:**

El ruido interno se deriva de los efectos térmicos del amplificador. Los amplificadores suelen generar unos microvoltios de ruido interno, lo que limita la resolución de la señal a este nivel. La cantidad de ruido añadido a la señal depende del ancho de banda de la entrada del amplificador.

Para reducir el ruido interno, hay que seleccionar un amplificador con un ancho de banda que corresponda con el ancho de banda de la señal de entrada.

- **Eliminación de ruido externo:**

El ruido externo surge de muchas fuentes. Por ejemplo, muchas aplicaciones de adquisición de datos están sujetas a un ruido de 60 Hz generado por los circuitos de alimentación de CA. Este tipo de ruido se conoce como pick-up o zumbido, y aparece como una señal de interferencia sinusoidal en el circuito de medición. Otra fuente común de interferencia es la iluminación fluorescente. Estas luces generan un arco a dos veces la frecuencia de la línea de poder (o alimentación) (120Hz)

El ruido se añade al circuito de adquisición desde esas fuentes externas, porque las puntas de la señal actúan como antenas que toman actividad eléctrica del ambiente.

Gran parte de este ruido es común a ambos cables de señal. Para eliminar la mayor parte de este voltaje en modo común, se debe:

- Configurar los canales de entrada en modo diferencial.
- Utilice cables de señal que son trenzados.
- Mantenga los cables de señal lo más corto posible.
- Mantenga los cables de señal lo más lejos posible de la actividad eléctrica del medio ambiente.

- **Filtrado:**

También reduce el ruido de la señal. Para muchas aplicaciones de adquisición de datos, un filtro pasa bajo es beneficioso. Como su nombre indica, un filtro pasa-bajo pasa los componentes de frecuencia baja, pero atenúa los componentes de frecuencias altas.

La frecuencia de corte del filtro debe ser compatible con las frecuencias presentes en la señal de interés y la tasa de muestreo utilizada para la conversión A / D. Un filtro pasa-bajo que es utilizado para evitar frecuencias más altas a causa de la distorsión introducida en la señal digitalizada, es conocido como un filtro antialiasing si el corte se produce en la frecuencia de Nyquist. Es decir, el filtro elimina las frecuencias más grandes que la mitad de la frecuencia de muestreo. Estos filtros generalmente tienen un corte más agudo que el filtro normal paso bajo usado para condicionar una señal. Filtros de antialiasing se especifican en función de la tasa de muestreo del sistema y debe haber un filtro por la señal de entrada.

### **3.3.1.6 Relación entre el rango del sensor y el rango del convertidor A/D**

Cuando la señal es digitalizada por un convertidor A/D, se debe tener en cuenta estos dos puntos:

- El rango esperado de los datos de la señal. Este rango depende de los fenómenos físicos que se miden y del rango de salida del sensor.
- El rango del convertidor A/D. Para muchos dispositivos, el rango del hardware esta especificado por la ganancia y la polaridad.

Se debe seleccionar el rango del hardware y del sensor de tal manera que se obtenga la máxima precisión, y el rango dinámico total de la señal de entrada sea cubierta.

Por ejemplo, se usa un micrófono con un rango dinámico de 20 dB a 140 dB y una sensibilidad de salida de 50mV/Pa. Si se mide el ruido en la calle, entonces se podría esperar que el nivel de sonido nunca exceda los 80 dB, lo que corresponde a una magnitud de presión sonora de 200mPa y una tensión de salida del micrófono de 10mV. En estas condiciones, se debe definir el rango de entrada de la tarjeta de adquisición de datos para una amplitud de señal máxima de 10mV, o un poco más.

### **3.3.1.7 ¿Qué tan rápido debe ser muestreada una señal?**

Cuando una señal continua es muestreada, parte de la información se pierde. La clave es muestrear a una tasa tal que la señal de interés este bien caracterizada y la cantidad de información perdida sea mínima.

Si se muestrea a una tasa que es muy baja, entonces, se puede producir aliasing en la señal. El aliasing puede ocurrir para señales que varían rápidamente como para señales que varían lenta

El aliasing ocurre cuando la señal muestreada contiene componentes de frecuencias mayores a la mitad de la tasa de muestreo. Los componentes de frecuencias podrían proceder de la señal, en tal caso se está sobre muestreando y podría incrementar la tasa de muestreo. También podrían proceder del ruido, en este caso se podría necesitar el acondicionamiento de la señal usando un filtro. Las reglas usadas para prevenir el antialiasing está dado por el Teorema de Nyquist, que establece que:

- Una señal análoga puede ser reconstruida sin errores únicamente desde las muestras tomadas con un intervalo de tiempo igual.
- La tasa de muestreo puede ser igual o dos veces más grande que el mayor componente de frecuencia de la señal análoga. Una frecuencia de la mitad de la tasa de muestreo es llamada como frecuencia de Nyquist.

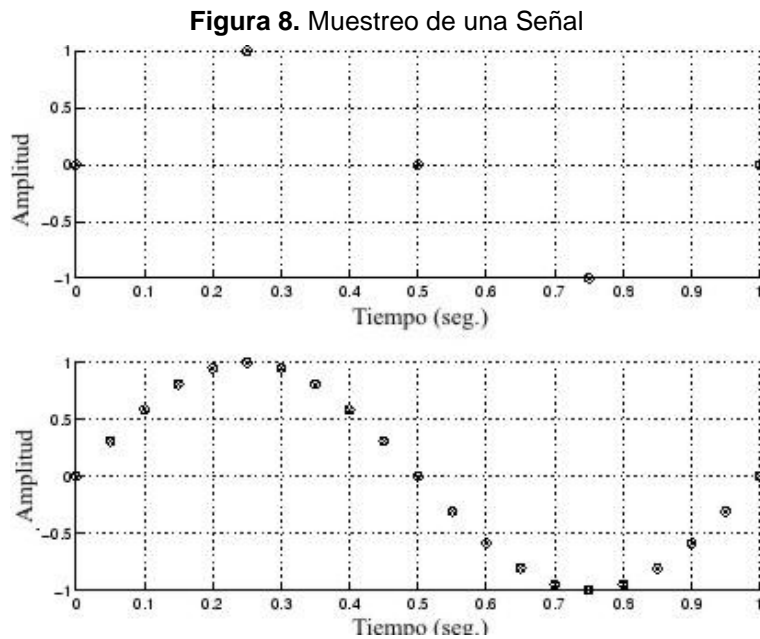
Sin embargo, si la señal de entrada está afectada por el ruido, entonces, el aliasing puede ocurrir aún.





Donde  $n$  es cero o cualquier entero positivo. Para este ejemplo, la señal actual puede ser de 3Hz, 5Hz, 7Hz, 9Hz, y así sucesivamente. La relación  $0.25 * (\text{Tasa de muestreo})$  es llamada el “alias” de la señal que puede estar en otra frecuencia. En otras palabras, El “aliasing” ocurre cuando una la identidad de otra frecuencia.

Si se muestrea la señal de entrada por lo menos dos veces más rápido que el componente de frecuencia más alto, la señal podría ser caracterizada singularmente, pero esta tasa no imita la forma de onda muy cerca. Como se muestra en la figura 8, se necesita una tasa de muestreo de aproximadamente 10 a 20 veces la frecuencia más alta.



Fuente: Data Acquisition Toolbox™ 2, User's Guide, MathWorks.

Como se muestra en el figura 8, la baja tasa de muestreo produce una señal muestreada que parece ser una forma de onda triangular. Como se muestra en la parte inferior de la figura 8, una señal muestreada de alta fidelidad se produce cuando la tasa de muestreo es mayor.

En este último caso, la señal muestreada en realidad se parece a una onda sinusoidal.

### **3.4 TOOLBOX DE ADQUISICION DE DATOS [1]**

La Toolbox de Adquisición de Datos es una colección de funciones de archivos .m y archivos .mex construido sobre el entorno de MATLAB. La Toolbox también incluye varios vínculos de librerías dinámicas (DLL's) llamados adaptadores, los cuales permiten interactuar con un hardware específico. La Toolbox tiene las siguientes características:

- Un marco para traer datos reales, medidos en el espacio de trabajo de MATLAB usando un hardware de adquisición de datos compatible con el PC.
- Soporte para entradas análogas (AI), salidas análogas (AO) y entradas y salidas digitales (DIO). Incluyendo conversiones simultáneas para entradas y salidas análogas.
- Soporte para estos tipos de hardware más populares:
  - Advantech®. Tarjetas que usen el administrador de dispositivos de Advantech.
  - Módulos VXI E1432A/33A/34A de Agilent Technologies®.
  - Keithley®. Tarjetas que usen DriverLINX.
  - Tarjetas de Measurement Computing™ Corporation.
  - National Instruments®. Tarjetas que usan el software tradicional NI-DAQ o NI-DAQmx.
  - Puertos paralelos LPT1 y LPT3.
  - Microsoft® Windows®. Tarjetas de sonido.

Adicionalmente, se puede usar un Kit de adaptador de interface para dispositivos que no soportan la Toolbox.

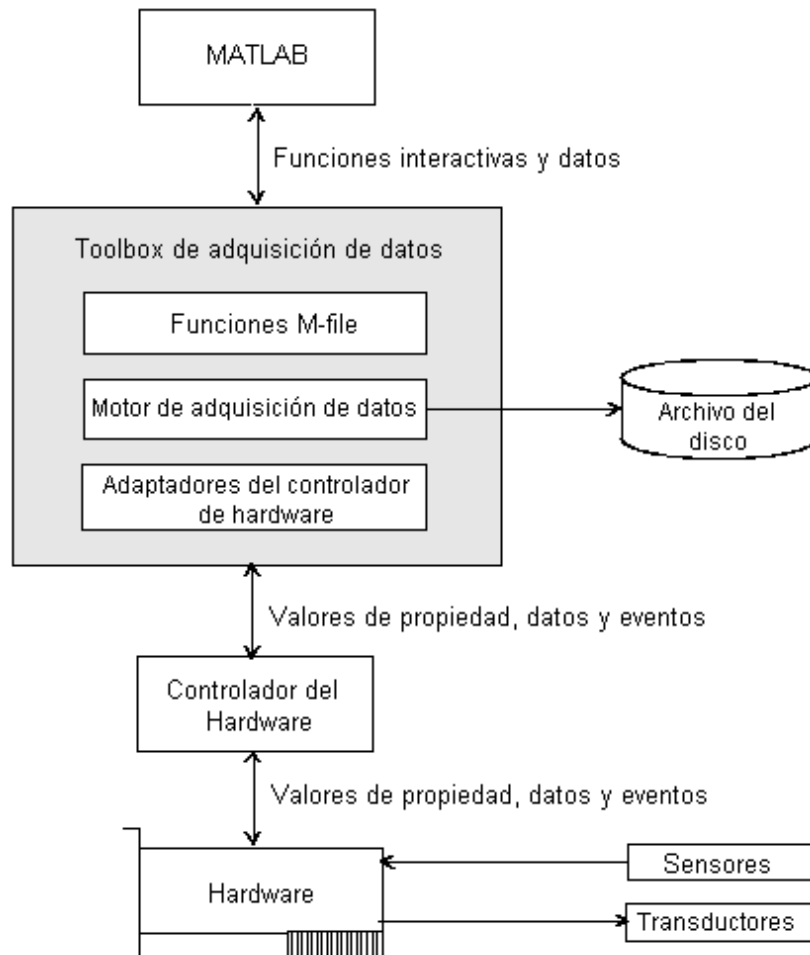
#### **3.4.1 Componentes de la Toolbox**

La "Toolbox" de Adquisición de Datos consta de tres componentes:

- Funciones de archivos m
- Motor de adquisición de datos
- Adaptadores de Hardware

La figura 9, muestra que estos componentes permiten pasar información entre MATLAB y el hardware de adquisición de datos.

**Figura 9.** Componentes de la Toolbox



Fuente: Data Acquisition Toolbox™ 2, User's Guide, MathWorks.

- **Valores de Propiedad:** Es posible controlar el comportamiento de la aplicación de adquisición de datos, configurando los valores de propiedad. En general, se puede pensar que una propiedad es como una característica de la Toolbox o del controlador del hardware que puede ser manipulado para satisfacer las necesidades de la aplicación.

- **Datos:** Se puede adquirir datos desde un sensor conectado a un subsistema de entrada análoga y guardarla en MATLAB, o sacar datos de MATLAB a un transductor conectado a un subsistema de salida análoga. Adicionalmente puede transferir valores (1s y 0s) entre MATLAB y un subsistema de entrada y salida digital.
  
- **Eventos:** Un evento ocurre en un instante particular después que una condición es cumplida y puede resultar en uno o más llamados que esté especificado. Los eventos pueden ser generados solo después que se configure las propiedades asociadas. Algunas de las maneras en las que se puede usar los eventos incluye la iniciación del análisis después que una predeterminada cantidad de datos sea adquirida, o mostrando un mensaje al espacio de trabajo de MATLAB después que un error ocurra.

### 3.4.2 Motor de adquisición de datos

El motor de la adquisición de datos es un archivo MEX que:

- Almacena los objetos de dispositivo y valores de propiedades asociadas que controla la adquisición de datos.
  - Controla la sincronización de eventos.
  - Controla el almacenamiento de los datos adquiridos o de los datos en cola.
- Mientras que el motor realiza esas tareas, se puede realizar otras tareas en MATLAB como analizar los datos adquiridos. En otras palabras, el motor y MATLAB son asíncronos. La relación entre adquirir datos, sacar datos y el flujo de datos es descrito a continuación.

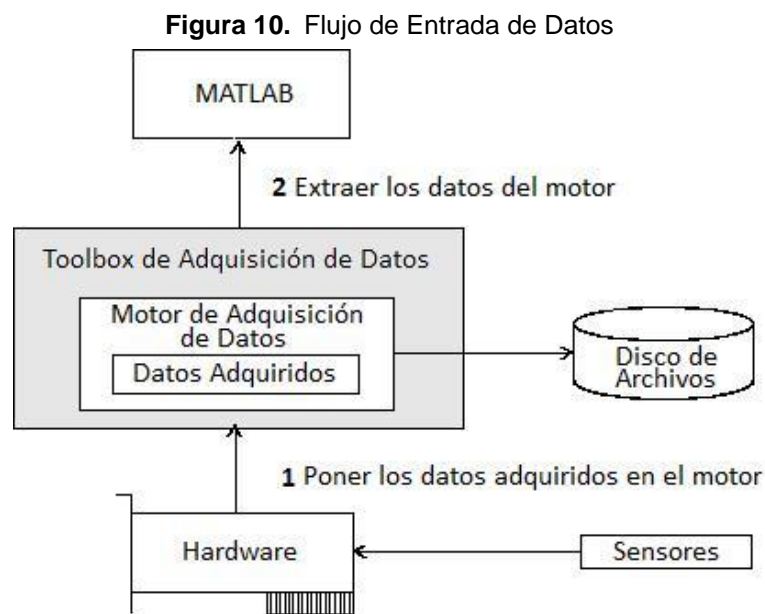
#### 3.4.2.1 Flujo de datos adquiridos:

En el flujo de datos, los datos son almacenados temporalmente en la memoria porque estos se van sobrescribiendo. La tasa con la cual los datos son sobrescritos depende de factores incluyendo la memoria disponible, la tasa con la que los datos son adquiridos y el número de canales del hardware.

Los datos almacenados no están disponibles automáticamente en el área de trabajo de MATLAB. Por ello, se tiene que extraer explícitamente del motor usando la función *getdata*.

El flujo de datos adquiridos como se muestra en la figura 10, consiste de dos pasos independientes:

1. Los datos adquiridos desde el hardware son almacenados en el motor.
2. Los datos son extraídos desde el motor y almacenados en el espacio de trabajo de MATLAB o sacados como archivo.



Fuente: Data Acquisition Toolbox™ 2, User's Guide, MathWorks.

### 3.4.2.2 Flujo de salida de datos:

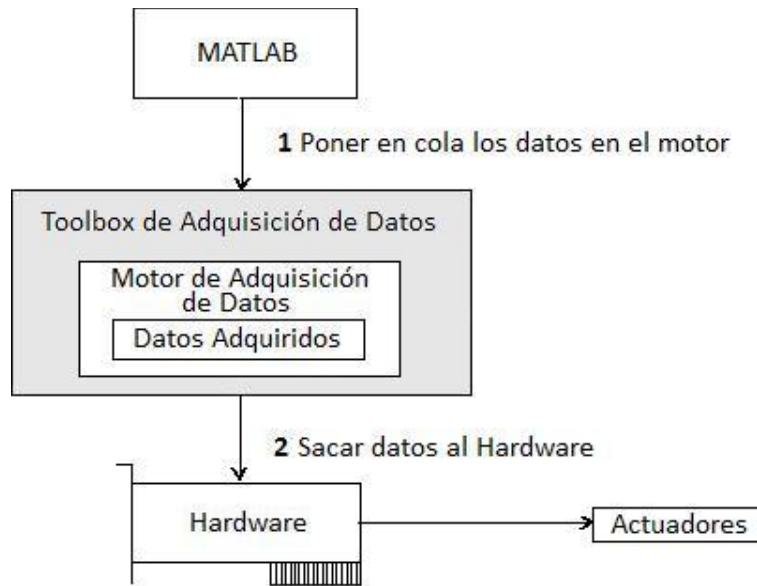
El flujo de salida de datos se refiere al flujo de datos desde el motor de adquisición de datos al hardware. Sin embargo, antes que los datos son sacados, se deben poner en cola en el motor con la función *PUTDATA*. La cantidad de datos que se puede poner en cola depende de factores incluyendo la memoria disponible, el número de canales del hardware y el tamaño de cada muestra de dato.

El flujo de salida de datos consiste de dos pasos independientes:

1. Los datos son puestos en cola en el motor desde el espacio de trabajo de MATLAB.
2. Los datos en cola en el motor son sacados al hardware.

Estos pasos se ilustran en la figura 11.

**Figura 11.** Flujo de Salida de Datos



Fuente: Data Acquisition Toolbox™ 2, User's Guide, MathWorks.

### 3.4.3 Adaptadores de Hardware:

El adaptador es la interfaz entre el motor de adquisición de datos y el controlador del hardware.

El propósito principal del adaptador es pasar información entre MATLAB y el hardware por medio de sus controladores.

Para adquirir datos usando una tarjeta de National Instruments, la versión apropiada del driver NI-DAQ debe ser instalada en su plataforma.

### 3.5 TARJETA NI USB-6008/6009 [2]

La tarjeta NI USB-6008/6009 brinda una conexión de 8 canales de entradas análogas (AI), dos canales de salidas análogas (AO), 12 canales de entrada/salida digitales y un contador de 32-bit con una interfaz USB de alta velocidad.

#### 3.5.1 Diferencias entre las tarjetas NI-USB 6008 y 6009

La tabla 2 muestra las diferencias entre las tarjetas de adquisición de datos USB 6008 y 6009 de National Instruments.

**Tabla 2.** Diferencias entre las Tarjetas NI-USB 6008 y 6009

Característica	USB 6008	USB 6009
Resolución en las entradas análogas	12 Bits en modo diferencial, 11 Bits para modo de nodo simple	14 Bits en modo diferencial, 13 Bits para modo de nodo simple
Máximo frecuencia de muestreo en las entradas análogas Canal individual*	10 kS/s	48 kS/s
Máximo frecuencia de muestreo en las entradas análogas Múltiples canales (Total*)	10 kS/s	42 kS/s
Configuración DIO	Colector abierto	Colector abierto o control activo
*Pueden ser dependiendo del sistema		

#### 3.5.2 Software de la Tarjeta

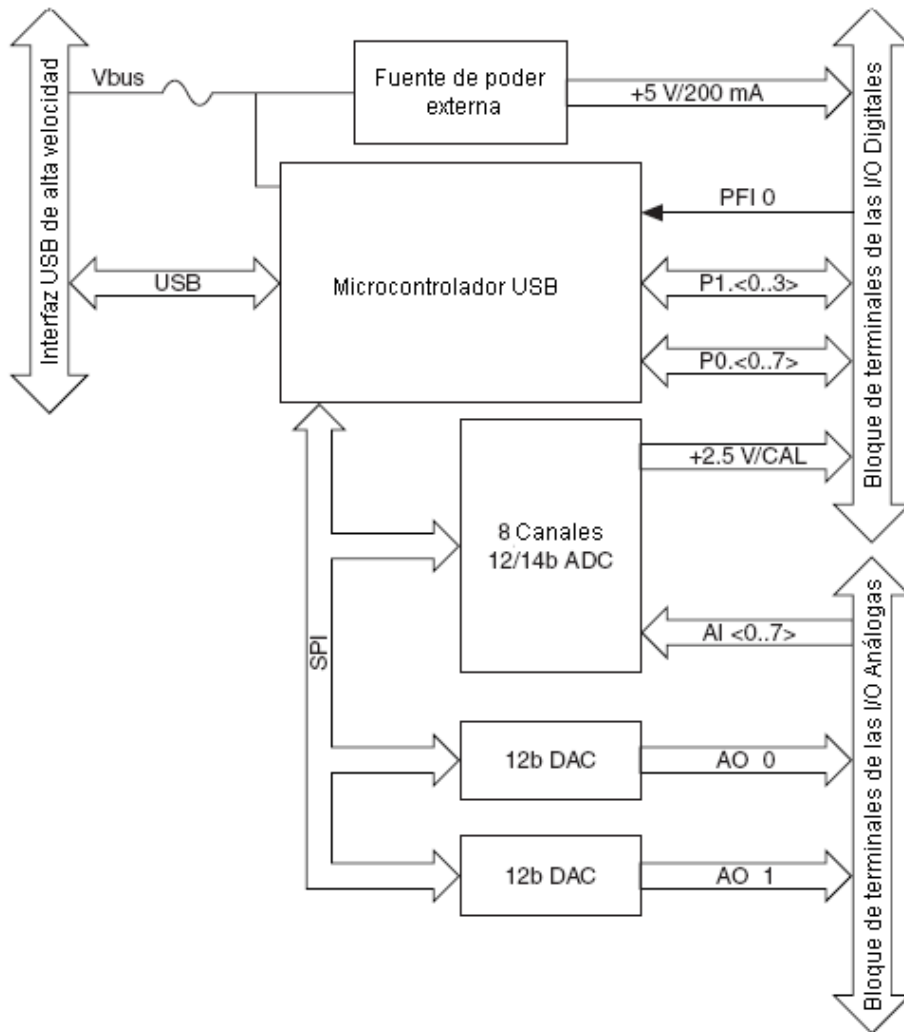
El software que soporta USB-6008 para Windows 2000/XP es NI-DAQmx.



### 3.5.3 Hardware

En la figura 12 muestra en un diagrama de bloques los principales componentes de las Tarjetas NI-USB 6008/6009.

Figura 12. Hardware de la tarjeta

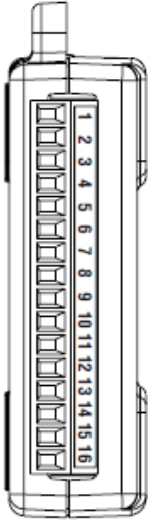


Fuente: Tarjetas DAQ, USB 6008/6009. User Guide and Specifications, National Instruments

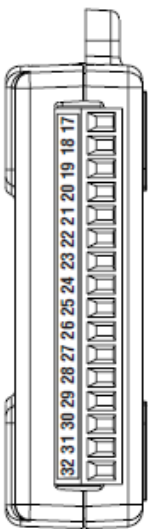
### 3.5.4 Conector de Entradas y Salidas

Hay un bloque de terminales para señales análogas y otro para señales digitales. Los bloques de terminales permiten 16 conexiones para usarlos con alambre 16 AWG a 28 AWG. La tabla 3 lista la ubicación de terminales análogos y la tabla 4 lista la de terminales digitales.

**Tabla 3.** Terminal Análogo

Modulo	Terminal	Señal, Nodo Simple	Señal, Diferencial
	1	GND	GND
	2	AI 0	AI 0+
	3	AI 4	AI 0-
	4	GND	GND
	5	AI 1	AI 1+
	6	AI 5	AI 1-
	7	GND	GND
	8	AI 2	AI 2+
	9	AI 6	AI 2-
	10	GND	GND
	11	AI 3	AI 3+
	12	AI 7	AI 3-
	13	GND	GND
	14	AO 0	AO 0
	15	AO 1	AO 1
	16	GND	GND

**Tabla 4.** Terminal Digital

Modulo	Terminal	Señal
	17	PO.0
	18	PO.1
	19	PO.2
	20	PO.3
	21	PO.4
	22	PO.5
	23	PO.6
	24	PO.7
	25	PI.0
	26	PI.1
	27	PI.2
	28	PI.3
	29	PFI 0
	30	+2.5V
	31	+5V
	32	GND

### 3.5.4.1 Descripción de las Señales

GND: punto de referencia para las medidas de entrada análoga de simple nodo, voltajes de salida, señales digitales en los conectores de I/O, fuente de +5v, y en los +2.5vdc de referencia.

AI<0....7> (entradas análogas): para medidas en modo s-t solo son las entradas del canal análogo en voltaje. En modo diferencial AI0 a AI4 son entradas positivas y negativas para las entradas análogas diferenciales del canal 0.

AO0 (salida con referencia en GND): suministra un voltaje de salida para el canal 0.

AO1 (salida con referencia en GND): suministra un voltaje de salida para el canal1.

P1<0...3> y P0<0...7> (referencia en tierra): Puertos de entrada o salidas digitales. Cada puerto se puede configurar como entrada o salida individualmente.

+2.5V (referencia en GND): referencia externa de 2.5Voltios para pruebas en realimentación.

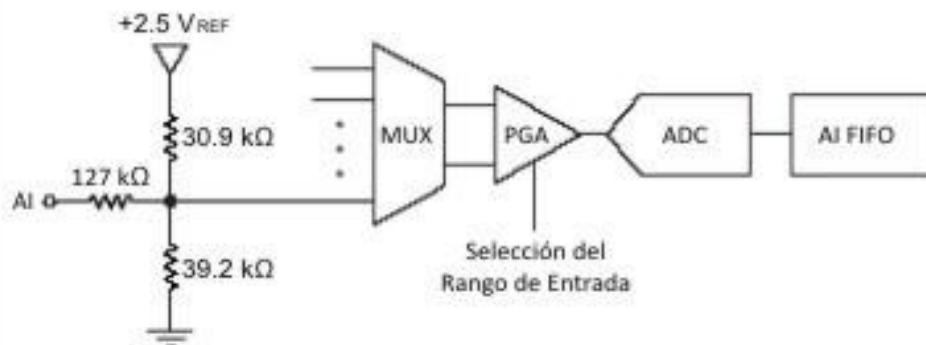
+5V (referencia en tierra): fuente de 5v, a 200mA (salida).

### 3.5.5 Entradas Análogas

*Circuito:*

En la figura 13 se muestra un diagrama de bloques de los componentes del circuito interno de las entradas análogas de la Tarjeta

**Figura 13.** Circuito de Entrada Análoga



Fuente: Tarjetas DAQ, USB 6008/6009. User Guide and Specifications, National Instruments

**Mux:** La USB 6008/6009 tiene un convertidor análogo a digital. El multiplexor enruta un canal de AI a tiempo con la PGA.

**PGA:** Amplificador de ganancias programables. Provee ganancia de entradas 1, 2, 4, 5, 8, 10, 16, o 20 para la configuración en medidas diferenciales y ganancia unitaria cuando se configuran para entrada sencilla. La ganancia del PGA es automáticamente calculada según el rango de voltaje seleccionado en la aplicación.

**A/D convertidor:** El convertidor digitaliza la señal de entrada analoga convirtiendo el voltaje análogo en un código digital.

**AI FIFO:** La tarjeta USB puede realizar conversiones A/D simples y multiples de codigo externo.

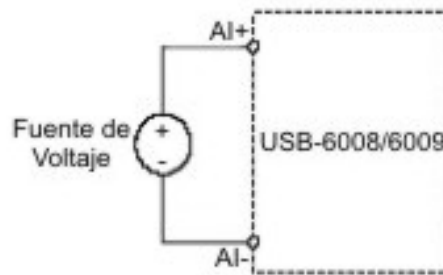
### 3.5.5.1 Modos de entradas analogas

Estas entradas se pueden configurar para tomar medidas simples o en modo diferencial.

- **Modo diferencial:**

conectar el pin positivo de la señal al terminal AI+ y negativo a terminal AI-.

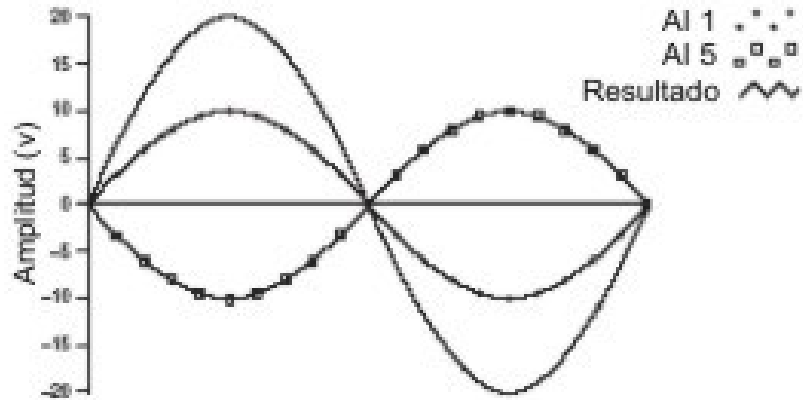
**Figura 14.** Conexión de una señal de voltaje diferencial



Fuente: Tarjetas DAQ, USB 6008/6009. User Guide and Specifications, National Instruments

En el modo de entrada diferencial se puede medir señales de  $\pm 20\text{v}$  en el rango de mas o menos  $\pm 20\text{v}$ . sin embargo el maximo voltaje en cualquier pin es de  $\pm 10\text{v}$  con respecto a tierra. Por ejemplo si AI1 es  $+10\text{v}$  y AI5 es de  $-10\text{v}$ , entonces la medida tomada por el dispositivo es de  $+20\text{v}$ , como se muestra en la figura 15.

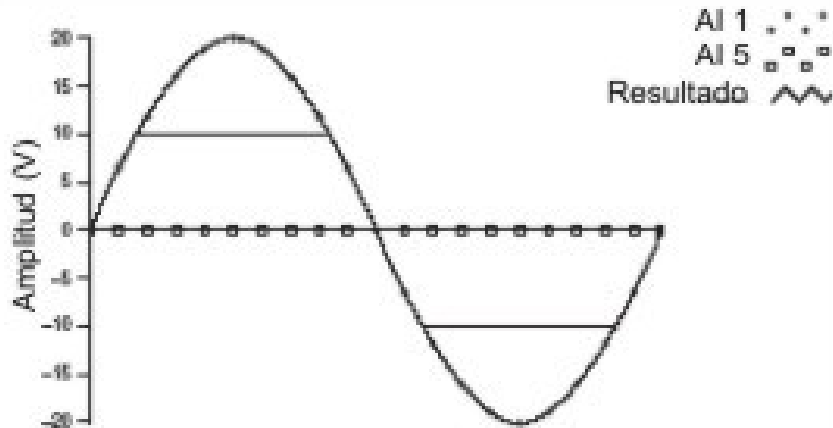
**Figura 15.** Ejemplo de una Medición Diferencial de 20V



Fuente: Tarjetas DAQ, USB 6008/6009. User Guide and Specifications, National Instruments

Conectando una señal de más de  $\pm 10\text{V}$  en cualquier pin resulta una señal de la figura 16.

**Figura 16.** Señal con Exceso de mas de +10V

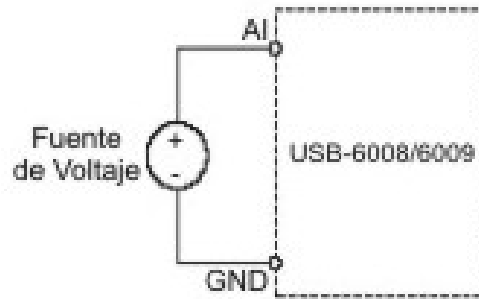


Fuente: Tarjetas DAQ, USB 6008/6009. User Guide and Specifications, National Instruments

- **Modo de nodo simple:**

Para conectar una señal de voltaje con referencia(RSE) a la tarjeta, conecte el positivo de la señal deseada al terminal AI y la tierra al terminal GND, tal como se muestra en la figura 17.

**Figura 17.** Conexión de una Señal Referenciada de Nodo Simple



Fuente: Tarjetas DAQ, USB 6008/6009. User Guide and Specifications, National Instruments

### 3.5.6 Trigger digital

Cuando una AI es definida, se puede configurar PF10 como un trigger digital. Cuando el trigger digital es activado la forma en AI espera un desborde en PF10 antes de empezar la adquisición. Para usar ai/starttrigger con uan fuente digital especifique PF10 como una fuente y seleccione el desborde.

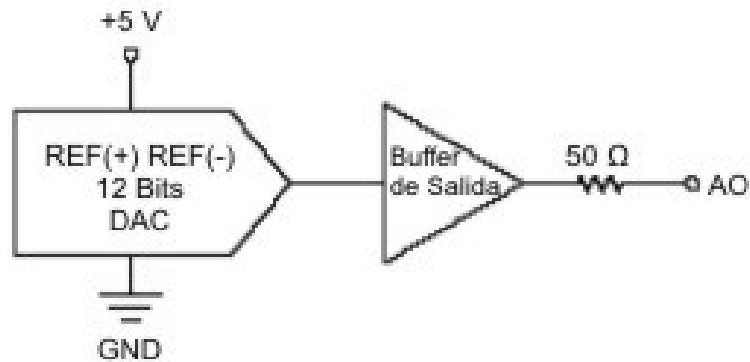
### 3.5.7 Salidas Análogas

la tarjeta USB 6008/6009 tiene dos canales de salidas analogas que pueden gernarar una salida para 0-5v.

*Circuito:*

En la figura 18 se muestra un diagrama de bloques de los componentes principales del circuito interno de las salidas análogas de la Tarjeta

**Figura 18.** Circuito de Salida Análoga



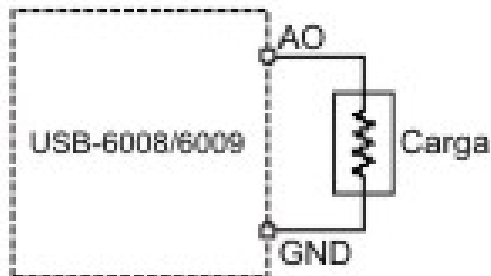
Fuente: Tarjetas DAQ, USB 6008/6009. User Guide and Specifications, National Instruments

*DACs (convertidores de digital a analógico):* convierte un código digital en voltajes analógicos.

*Conexión de carga a la salida analógica:*

Para conectar cargas a la tarjeta, conecte el pin positivo de la carga al terminal AO, y conecte la tierra de la carga al terminal, tal como se ilustra en la figura 19.

**Figura 19.** Conexión de una Carga a la Salida Análoga



Fuente: Tarjetas DAQ, USB 6008/6009. User Guide and Specifications, National Instruments

### 3.5.8 Entradas y Salidas Digitales:

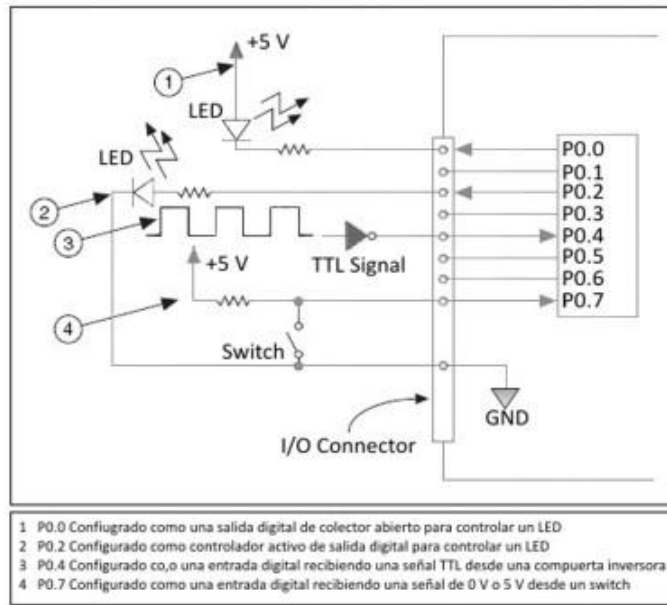
La tarjeta tiene dos puertos digitales, uno 8 líneas PO<0....7> y el otro de cuatro líneas PI<0....3>, los cuales pueden ser tomados como entradas o como salida. Tierra es la referencia del puerto.



Circuito:

En la figura 20, se muestra algunas formas de conexión de entradas y salidas digitales a la Tarjeta.

**Figura 20.** Ejemplos de conexión de Entradas y Salidas Digitales



Fuente: Tarjetas DAQ, USB 6008/6009. User Guide and Specifications, National Instruments

### 3.5.9 Protecciones en las entradas y salidas de la Tarjeta

Para proteger la USB-6008/6009 contra condiciones de sobre voltaje, bajo voltaje y sobre corriente se debe tener en cuenta las siguientes directrices:

- Si se configura una línea DIO como salida no conectarla a fuente externa, señal a tierra o fuente de potencia.
- Si se configura una línea DIO como salida verificar que la corriente de carga conectada no excede los límites de corriente del hardware DAQ.
- Si configura una línea DIO línea como entrada no alimentar la línea con voltajes fuera del rango normal de operación. Las líneas DIO tienen un rango de operación mas bajo que las señales AI.
- Tratar el hardware como sensible a la estática.
- Conectar a tierra apropiadamente el operador y el equipo cuando se manipula o conecta el aparato DAQ.

## 4 METODOLOGIA Y DESCRIPCION DEL PROYECTO

Buscando el interés de los estudiantes hacia el reconocimiento de las Tarjetas NI-USB 6008/6009 como una herramienta útil y de fácil manejo en soluciones de problemas o proyectos tanto en la vida estudiantil como profesional se presenta como proyecto un tutorial donde se expone las características de las tarjetas y cómo programarlas con la Toolbox de Adquisición de Datos de MATLAB, ofreciendo una nueva alternativa de programación ya que éstas han sido utilizadas en los últimos tiempos con LabVIEW.

### 4.1 *Estudio de la Toolbox y la Tarjeta*

Como primer paso para el desarrollo del proyecto se tradujeron los manuales de la Toolbox de Adquisición de Datos y de la tarjeta NI-USB 6008/6009 para lograr el entendimiento del funcionamiento y las características de cada uno.

Del manual de la Toolbox se hizo un estudio detallado de los 8 primeros capítulos, los cuales contienen la explicación teórica de los componentes de un sistema de adquisición de datos, instrucciones y características para realizar una sesión de adquisición, ejemplos de cada tema utilizando la tarjeta de sonido incluida en los computadores, luego se realizaron los ejemplos del manual de la Toolbox, con la tarjeta de sonido del computador. Durante el estudio de la Toolbox se observó que la Toolbox no tiene soporte al subsistema de contador de la tarjeta, por lo tanto no fue posible cumplir con el objetivo planteado en el anteproyecto para realizar eventos de conteo y generación de pulsos con el uso del contador de la tarjeta.<sup>1</sup>

Seguido a esto se procedió a conocer las propiedades, especificaciones, cuidados básicos y limitaciones de las tarjetas NI-USB 6008/6009. Una vez realizado esto,

---

<sup>1</sup> Para corroborar esta observación dirigirse a la pagina 7-2 del manual de la Toolbox. (Data Acquisition Toolbox™ 2, User's Guide. pag7-2)

los ejemplos desarrollados con la tarjeta de sonido, fueron ajustados a las propiedades de la tarjeta DAQ y probados con esta, teniendo en cuenta sus limitaciones.

En este punto se comprobó una de las limitaciones de la tarjeta en las salidas análogas, puesto que esta no permite entregar datos como un flujo de datos, sino dato por dato; es decir que la tarjeta entrega una señal análoga muestreada, y solo conserva en la memoria el último dato entregado.

## **4.2 *Diseño del Tutorial***

El diseño del tutorial se basó en dos pasos:

1. Selección del contenido
2. Diseño de la estructura

### **4.2.1 Selección del contenido**

Una vez completado el estudio de la Tarjeta y de la Toolbox de MATLAB, se pasó a la selección y organización de los temas del contenido del tutorial, escogiendo la información necesaria que se debe tener para extraer y entregar datos digitales y análogos con la tarjeta por medio de la Toolbox de adquisición. La organización de los temas se realizó con una metodología de aprendizaje que parte de lo básico a lo complejo.

El contenido del tutorial está dividido en seis capítulos, que son mencionados a continuación:

- Capítulo 1: Tarjeta NI-USB 6008/6009
- Capítulo 2: Toolbox de Adquisición de Datos
- Capítulo 3: Entrada/Salida Digital
- Capítulo 4: Salida Análoga
- Capítulo 5: Entrada Análoga
- Capítulo 6: Laboratorios

El capítulo 1, inicia con una descripción general de la tarjeta y de sus conectores análogos y digitales. Brinda una explicación del uso y conexión de sus entradas y salidas digitales, entradas y salidas análogas, y las diferencias entre las tarjetas 6008 y 6009. Además, ofrece temas adicionales y ayudas para el uso de la Tarjeta, tales como una breve descripción del hardware, la instalación y especificaciones de la tarjeta.

En el capítulo 2, se expone una breve descripción de la Toolbox y de sus componentes, algunos comandos de ayuda que ésta ofrece y el proceso que se realiza en una adquisición de datos para extraer y entregar datos.

En los capítulos del 3 al 5, se explican los pasos a seguir para realizar una sesión de adquisición de datos de entradas/salidas digitales, salidas análogas y entradas análogas respectivamente.

Estos tres capítulos plantean paso por paso la forma como se programa la tarjeta para efectuar una adquisición de datos con la Toolbox. Muestran la función o propiedades que se deben usar para realizar cada paso, sus respectivas sintaxis, descripción de las mismas, y breves ejemplos ilustrativos.

Cada uno de estos capítulos, presenta unos ejemplos tipo DEMO, en los cuales se aplica la teoría explicada en el capítulo. Estos ejemplos se muestran con su respectiva programación en el Anexo 3 del presente libro.

El capítulo 6, presenta dos prácticas. Una para que el usuario la desarrolle y la otra un ejecutable que muestra la aplicación de una adquisición de datos para el control de posición utilizando el módulo de servomecanismos, las cuales serán explicadas más adelante.

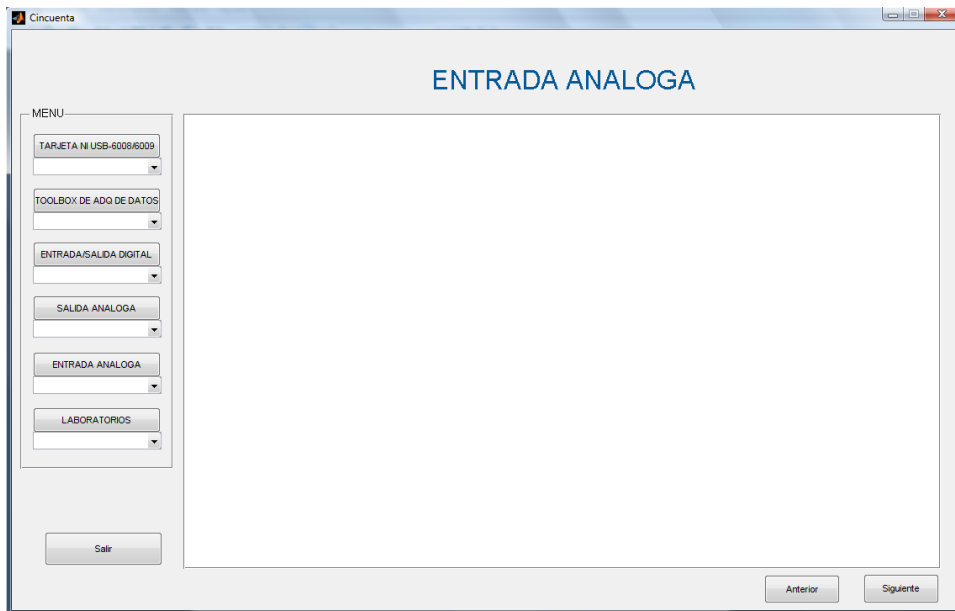
#### **4.2.2 Diseño de la estructura**

Teniendo distribuida la parte teórica, se realizó el diseño en papel una estructura del tutorial y se pasó a realizar la interfaz gráfica de usuario (GUI) en MATLAB de la estructura previamente diseñada. En esta parte fue necesario recordar por medio de videos y foros publicados en Internet y del "HELP" de MATLAB, cómo se

crean una GUI, la utilización de las herramientas de la GUI y la programación de cada herramienta.

En la siguiente figura (figura 21) se muestra el diseño de la estructura del tutorial en la GUI.

**Figura 21.** Estructura de una GUI del tutorial



Las GUIs del tutorial están divididas en tres áreas, las cuales son:

- Menú
- Presentación del contenido
- Direccionamiento básico

#### **4.2.2.1 Menú**

Consta de seis botones acompañados cada uno de una lista desplegable.

Cada botón conduce al inicio de un capítulo y la lista desplegable dirige a un tema específico del capítulo.

#### **4.2.2.2 Presentación de contenido**

El área de presentación de contenido es un marco o “Frame” donde se muestra la explicación de los temas.

### 4.2.2.3 Direccinamiento básico

Consiste en dos botones, “Siguiente” y “Anterior” que permiten avanzar en el orden propuesto de los temas o retroceder a temas anteriores.

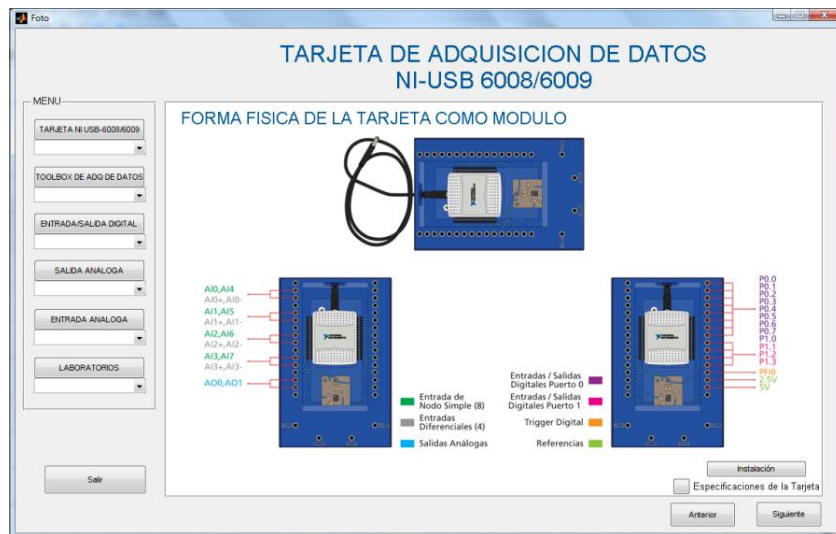
## 4.3 ELABORACIÓN DEL TUTORIAL

La elaboración de las GUIs del tutorial consistió en la digitalización de la teoría en textos estáticos y en la adaptación de la estética de colores, tamaños de elementos, algunas animaciones realizadas con programación y el uso de imágenes previamente diseñadas.

Las imágenes ilustrativas se crearon como complemento a la explicación teórica, utilizando el software de diseño COREL PHOTO PAINT X3.

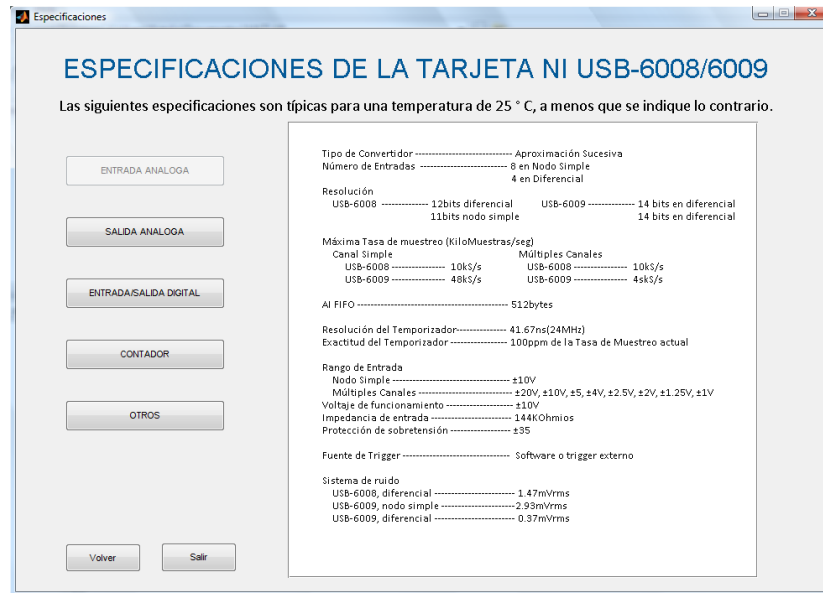
A medida que se fueron desarrollando los temas en las GUIs se observó que era necesario crear otro tipo de GUI extra que tiene como fin la profundización o complementación en ciertos temas del tutorial. El acceso a estas GUIs es por medio de unos botones que aparecen dentro del área de presentación del contenido en la parte inferior derecha y una vez consultadas, tienen la opción de regresar a la GUI de la cual fueron llamadas o salir del tutorial. Por ejemplo, la figura 22 muestra una interfaz del capítulo 1 del tutorial que permite el acceso a la explicación de la instalación de las tarjetas y a las especificaciones de la tarjeta.

Figura 22. GUI con Acceso a Temas Complementarios



Para las GUIs de temas no se realizó un diseño de estructura, sino que se iban creando de acuerdo a la información adicional que pretende mostrar. En la figura 23 se observa una interfaz de una GUI extra, la cual presenta las especificaciones de la tarjeta.

**Figura 23.** GUI de Tema extra



Una vez creada la presentación teórica del tutorial, se pasó al planteamiento de ejemplos y al diseño de las GUIs de éstos.

Cada ejemplo planteado fue debidamente probado en el laboratorio de electrónica de la UPB; para las pruebas se uso un generador de funciones, el osciloscopio y un multímetro digital, además de los elementos usados en los circuitos de prueba.

#### **4.4 MÓDULO DE PRUEBAS DIGITAL Y ANÁLOGO**

Con base a las aplicaciones de los ejemplos se diseñó un circuito de tal forma que éste reuniera los elementos necesarios para cada aplicación y poder comprobar el funcionamiento de los ejemplos.

El módulo está dividido en dos secciones que son:

- Módulo Digital
- Módulo Análogo

#### **4.4.1 Módulo digital**

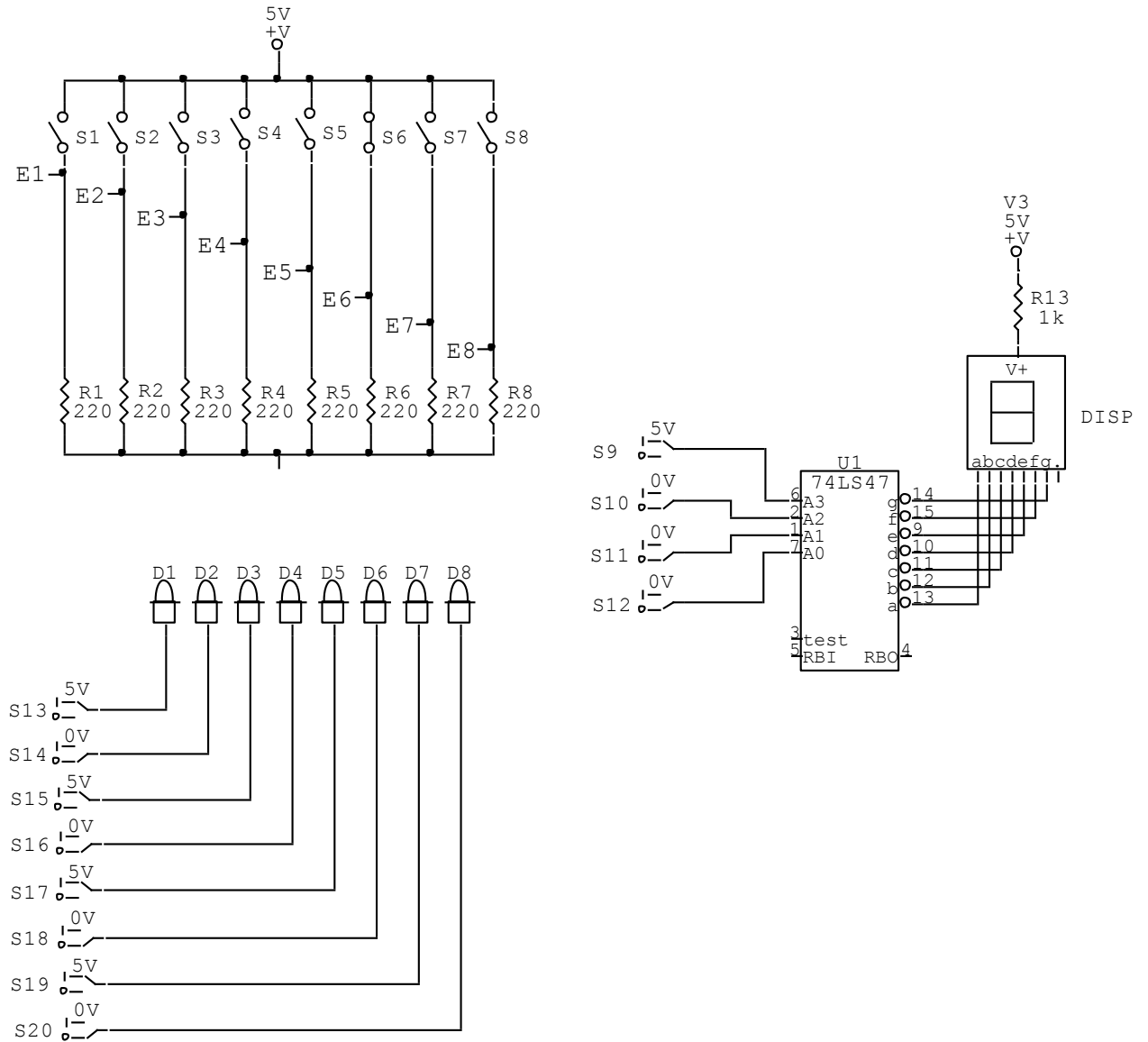
El módulo digital proporciona los siguientes elementos:

- Ocho terminales de un DIP SWITCH que pueden ser tomados como entradas digitales a la tarjeta. Cuando se ajustan al estado de encendido “ON” toma una tensión de 5V.
- Ocho LEDs con los terminales negativos conectados a tierra y los terminales positivos conectados cada uno a una resistencia los cuales pueden ser alimentados con las salidas digitales de la tarjeta.
- Codificador BCD a 7 segmentos conectado a un display de cátodo común. Las entradas de control del codificador sirven como salidas digitales de la tarjeta y con esto se puede realizar conteos que implican los números del 0 al 9.  
Es posible conectar directamente 7 salidas de la tarjeta a los terminales del display.

En la figura 24 se muestra los circuitos en circuit Maker del módulo digital, los interruptores digitales S9 al S20 simulan las salidas de la tarjeta y las entradas a la tarjeta son tomadas desde los interruptores S1 al S8 y son marcadas como E1 al E8.



**Figura 24. Circuitos del Módulo Digital**

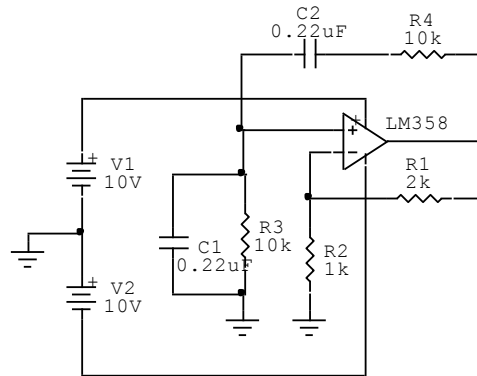


#### 4.4.2 Módulo análogo

Contiene un generador de una onda seno<sup>2</sup> y una onda cuadrada de 5Vpp a una frecuencia de 90Hz. Las dos salidas del generador son tomadas como entradas análogas a la tarjeta.

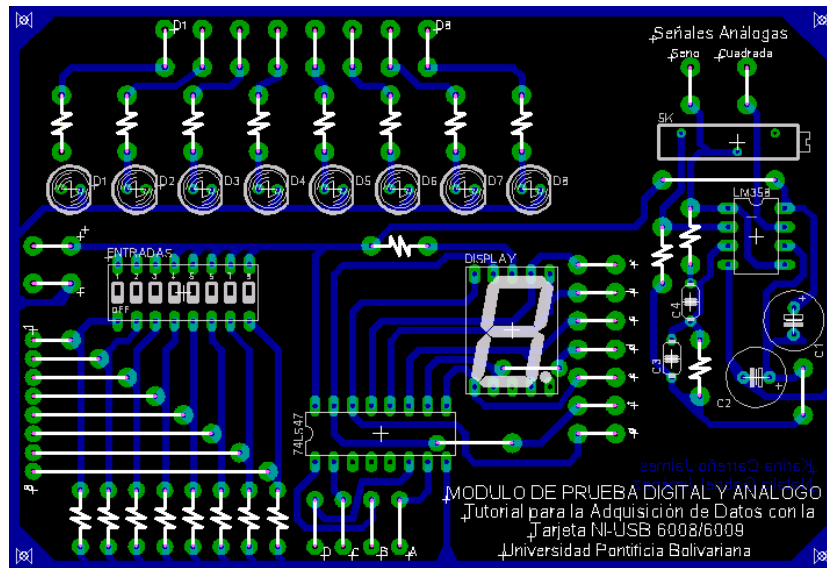
<sup>2</sup> El circuito fue tomado del MANUAL DE PRÁCTICAS PARA EL LABORATORIO DE ELECTRONICA INDUSTRIAL, del profesor JUAN CARLOS VILLAMIZAR RINCON para las Unidades Tecnológicas de Santander, adicionalmente se le agregó un comparador a su salida para generar lo onda cuadrada.

**Figura 25.** Circuito del generador de onda sinusoidal



Teniendo los circuitos, se procedió a realizar los montajes en PROTOBOARD, luego se probó el funcionamiento de los circuitos. Seguido a esto se hicieron las conexiones entre el módulo y la tarjeta para comprobar los ejemplos del tutorial. Una vez realizada las pruebas se pasó al diseño de la circuito impreso en EAGLE, el cual se muestra en la figura 25.

**Figura 26.** Diseño del circuito impreso en EAGLE



#### **4.5 Laboratorios**

Para la realización de las prácticas como primer paso se estudiaron las partes y el funcionamiento del módulo de servomecanismo del laboratorio de control, se realizó una adquisición para tomar algunas señales del módulo con la tarjeta. Seguido de esto se planteó la toma de muestras de señales como una práctica para el usuario haciendo la GUI donde se describe brevemente el módulo de servomecanismo y se muestra los pasos a realizar para la práctica.

Se planteó una aplicación con el módulo de servomecanismo para el control de posición, la cual es ejecutada por el usuario, se hizo la GUI de la aplicación y se probó su funcionamiento.

## 5 RESULTADOS

Del proyecto de grado se obtuvo un tutorial para la programación de las tarjetas NI-USB 6008/6009 elaborado como una interfaz gráfica de usuario de MATLAB. La figura 27 muestra la GUI de presentación inicial del tutorial

**Figura 27.** GUI Inicial del Tutorial



El tutorial contiene cinco capítulos teóricos que ofrecen conocimientos sobre la tarjeta y la Toolbox, instrucciones, funciones y pasos básicos para realizar una sesión de adquisición de datos.

Cuenta con un módulo de pruebas digital y análoga, el cual es usado como medio físico para la realización de las aplicaciones de los ejemplos tipo DEMO presentados al final de programación.

En figura 28, muestra un ejemplo de GUI de un ejemplo

**Figura 28.** GUI de un Ejemplo tipo DEMO



Como finalización del tutorial en capítulo 6 se presenta una aplicación de una sesión de adquisición para el control de posición de un motor usando el módulo de servomecanismos, la GUI de la aplicación se muestra en la figura 30.

## 6 CONCLUSIONES

Se obtuvo un tutorial creado como una interfaz gráfica de usuario en MATLAB con el cual se obtiene los conocimientos necesarios para realizar una sesión de adquisición para leer y escribir datos digitales y análogos con las tarjetas NI-USB 6008/7009 por medio de la Toolbox de adquisición de datos.

La Toolbox de Adquisición de Datos ofrece soporte a las librerías y controladores de muchos dispositivos, haciendo fácil el intercambio de información entre MATLAB y el Hardware, puesto que no es necesario incluir instrucciones para soportar la comunicación serial entre ellos.

El manejo de Toolbox con los dispositivos a los cuales tiene soporte es similar en instrucciones pero ajustando las propiedades a las de cada dispositivo, por lo cual es posible programar otro dispositivo teniendo las especificaciones del mismo, una vez se haya desarrollado el tutorial y los ejemplos presentes en él.

Para realizar una sesión de adquisición independientemente de la aplicación que se tenga se siguen los mismos pasos los cuales son, crear el objeto, agregarle las líneas o los canales al objeto, configurar propiedades (para objetos de entrada análoga), ejecutar el objeto (para objetos de entrada análoga), extraer o escribir los datos y borrar el objeto si éste no se va a utilizar más.

El extraer datos al entorno de MATLAB permite guardar los datos como archivo para un posterior análisis, visualización o para ser manipulados y así ejecutar una acción de salida que permita realizar una acción de control.

El ancho de banda de las tarjetas NI-USB 6008/6009 es de 10.000 muestras por segundo y 48.000 muestras por segundo respectivamente, pero para una visualización de una señal que represente la señal física muestreada, es recomendable muestrear a una tasa de 10 a 20 veces mayor al componente de frecuencia máxima de la señal.

El muestreo en las tarjetas cumple el teorema de Nyquist el cual dice que con dos puntos tomados en de una señal es posible realizar la reconstrucción de la misma, por lo tanto con base a las tasas de muestreo soportadas por las tarjetas, es posible muestrear bajo este teorema señales entre 1,2Hz y 5KHz con la tarjetas 6008 y señales con frecuencias entre 1,2Hz y 24KHz con la tarjeta 6009.

El tutorial es una herramienta de fácil uso que permite el previo estudio de las tarjetas y de la Toolbox para realizar posteriores proyectos.

## **7 RECOMENDACIONES**

Se recomienda realizar proyectos donde se vea la utilización de las tarjetas con la Toolbox de Adquisición de Datos de MATLAB.

Es recomendable que las tarjetas sean utilizadas en proyectos donde se involucre el muestreo de señales con bajas frecuencias.

El tutorial podría ser mejorado adicionándole más aplicaciones donde se utilicen los puertos digitales en conjunto con las entradas y salidas análogas.

Con base a lo explicado en el tutorial, podría aplicarse esos conocimientos para la elaboración de prácticas con los módulos del laboratorio de control.



## 8 REFERENCIAS

**[1]** Data Acquisition Toolbox 2, User's Guide. MathWorks.

**[2]** Tarjetas DAQ, USB 6008/6009. User Manual, National Instruments.

## 9 BIBLIOGRAFIA

- 1- Data Acquisition Toolbox 2, User's Guide. MathWorks.
- 2- Data Acquisition Toolbox 2, User's Guide. MathWorks.
- 3- Data Acquisition Toolbox, Quick Reference Guide. MathWorks.
- 4- Data Acquisition Toolbox 2.8. Data Sheet, MathWorks.
- 5- Matlab r2007b, Help. MathWorks.
- 6- VILLAMIZAR RINCON, Juan Carlos. MANUAL DE PRÁCTICAS PARA EL LABORATORIO DE ELECTRONICA INDUSTRIAL. Unidades Tecnológicas de Santander

## **ANEXOS**

**ANEXO 1. TUTORIAL PARA LA PROGRAMACION DE LAS TARJETAS NI-USB  
6008/6009 CON LA TOOLBOX DE ADQUISICION DE DATOS. GUIA RAPIDA DE  
USUARIO**

# **TUTORIAL PARA LA PROGRAMACION DE LAS TARJETAS NI-USB 6008/6009 CON LA TOOLBOX DE ADQUISICION DE DATOS DE MATLAB**

## **GUIA RAPIDA DE USUARIO**

**KARINA CARREÑO JAIMES  
NATALIA GABRIEL JIMENEZ**



En los últimos años la Tarjeta de Adquisición de Datos USB 6008/6009 de National Instruments ha sido muy utilizada por la Universidad Pontificia Bolivariana en los laboratorios de control, proyectos de grados y hasta ha sido incluida como herramienta básica en el diplomado de LABVIEW.

En todos los casos el software de programación es LABVIEW, por lo tanto el objetivo del tutorial para la programación de la tarjeta con la Toolbox de Adquisición de Datos es brindar otro medio para su uso, explicando claramente sus características, la descripción de la Toolbox y las instrucciones básicas para realizar una sesión de adquisición, por medio de teoría complementada con imágenes, ejemplos resueltos por temas y dos experiencias propuestas para que el usuario las desarrolle en la finalización del tutorial.

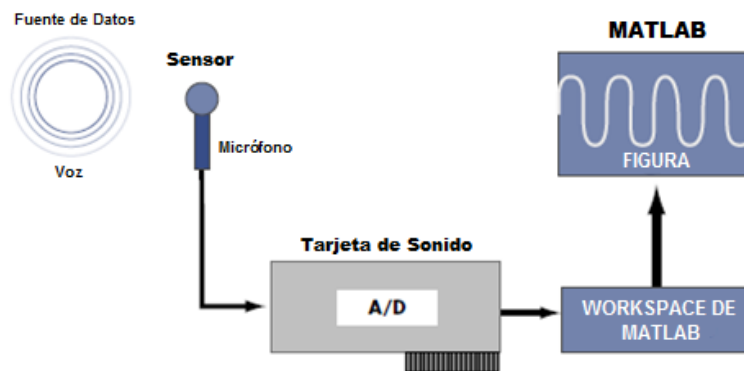
Esta guía describe cómo usar el tutorial y los requerimientos para su ejecución y la de los ejemplos.

### SISTEMA DE ADQUISICION DE DATOS [1]

El objetivo de cualquier sistema de adquisición de datos es proporcionar las herramientas y recursos necesarios para tomar señales físicas y convertirlas en datos que posteriormente se puedan procesar y mostrar.

Un sistema de adquisición de datos se podría tomar como un grupo de hardware y software que permiten interactuar con el mundo real, como muestra la figura 1, consta de estos componentes:

**Figura 1.** Ejemplo de Configuración de un Sistema de Adquisición de Datos



FUENTE: Data Acquisition Toolbox 2.8. Data Sheet, MathWorks

- **Hardware de Adquisición:** Es el corazón de cualquier sistema de adquisición de datos. La función principal es hacer la conversión de señales análogas a señales digitales y señales digitales a análogas. Conversión A/D y D/A.
- **Sensores y Actuadores (Transductores):** Un transductor es un dispositivo que convierte un tipo de energía de entrada en una energía de salida de otra forma.

- **Acondicionador de señal:** Las señales de los sensores a menudo son incompatibles con el hardware de adquisición de datos, y para superar esto las señales deben ser acondicionadas. Por ejemplo, las señales podrían ser amplificadas o volverlas en señales sin componentes de frecuencias indeseadas. Las señales de salida también pueden ser acondicionadas.
- **Computador:** Proporciona un procesador, un sistema de reloj, un bus de datos, memoria y espacio en el disco para almacenar datos.
- **Software:** Permite el intercambio de información entre el computador y el hardware.

En el tutorial se maneja como hardware de adquisición la tarjeta NI-USB 6008/6009, software la Toolbox de Adquisición de Datos y toma como sensores las entradas y salidas digitales y señales análogas desde el Módulo de Pruebas Digital y Análogo.

## ***TOOLBOX DE ADQUISICIÓN DE DATOS [2]***

La Toolbox de Adquisición de Datos proporciona un conjunto de herramientas de entrada análoga, salida análoga y entradas o salidas digitales para una variedad de hardware de adquisición compatibles con el PC.

La Toolbox permite la configuración de dispositivos externos para leer datos en MATLAB para un inmediato análisis o visualización y enviar datos de salida.

Realiza automáticamente las conversiones A/D y D/A, para recibir o enviar datos.

Se puede ejecutar las funciones de la Toolbox de Adquisición de Datos como una línea de comandos o programas a través de MATLAB.

La Toolbox proporciona funciones para la creación de objetos que están directamente asociados con el hardware. Estos objetos incluyen las propiedades base, tales como la tasa muestreo, configuración del trigger (disparo) y



propiedades de canales. También incluyen las propiedades que permiten el acceso a las características específicas y capacidades del hardware.

Los objetos del dispositivo proveen un acceso a la funcionalidad del hardware y permiten el control del comportamiento de la adquisición. Por ejemplo, se puede extraer datos análogos a través de un objeto de entrada análoga creado en MATLAB.

## Referencias

[1] Data Acquisition Toolbox 2, User's Guide. MathWorks.

[2] Data Acquisition Toolbox 2.8. Data Sheet, MathWorks

---

[1] Data Acquisition Toolbox™ 2, User's Guide, MathWorks.  
[2] Data Acquisition Toolbox 2.8. Data Sheet, MathWorks.

## DESCRIPCION Y MANEJO DEL TUTORIAL

---

Antes de tener una descripción del tutorial o de aprenderlo a utilizarlo es necesario verificar que se cumple con los requerimientos y pasos necesarios para la ejecución.

### ***Requerimientos para correr el Tutorial***

Para la ejecución del tutorial se debe tener previamente instalado MATLAB y la Toolbox de Adquisición de Datos, en computador y realizar los siguientes pasos para la ejecución:

1. Del CD-ROM copiar la carpeta de imágenes en la dirección C:\ con el nombre "ImágenesTutorialToolbox".
2. Copiar del CD-ROM los archivos de la carpeta llamada TUTORIAL en la carpeta de trabajo de MATLAB del computador. Esta carpeta está ubicada en algunos casos en la carpeta "Mis documentos" o "Documentos" con el nombre de MATLAB o WORK y en otros casos en C:\Program Files\MATLAB\, con el nombre WORK.
3. Abrir MATLAB
4. En la ventana de comandos de MATLAB digitar:

```
Tutorial
```

En esta parte se la GUI de presentación del tutorial que se muestra en la figura 2.

**Figura 2.** GUI Inicial del Tutorial



## ***DESCRIPCION DEL TUTORIAL***

Los temas del capítulo están divididos en 5 capítulos, más un capítulo de practica.

El capítulo 1 y 2 trata las características principales de la tarjeta y la Toolbox de Adquisición de Datos.

En los capítulos 3,4 y 5 se dan los pasos necesarios para realizar una adquisición para extraer y entregar datos digitales y análogos. Cada capítulo explica las funciones y propiedades de cada paso.

En estos capítulos se incluye una serie de ejemplos resueltos a ejecutar con el módulo de pruebas, los cuales muestran la síntesis de los pasos que se llevan a cabo en la adquisición según la aplicación que tiene cada uno.

Como finalización del tutorial, en el capítulo 6 se plantea una experiencia para que el usuario ponga en práctica los conocimientos adquiridos y se presenta una aplicación en la cual se realiza el control de posición de un motor en el módulo de servomecanismo.

## **MANEJO DEL TUTORIAL**

Para acceder a los temas principales el tutorial ofrece dos formas de manejo, las cuales son:

- Botones de direccionamiento básico
- Menú

### **Botones de direccionamiento básico:**

Se encuentran ubicados en la parte inferior derecha de la GUI. Consiste en dos botones, “Siguiente” y “Anterior”, permiten avanzar en el orden propuesto de los temas o retroceder a temas anteriores.

### **Menú:**

Está ubicado en la parte izquierda de la GUI. Permite un acceso rápido a los temas del tutorial.

Consta de seis botones acompañados cada uno de una lista desplegable. Cada botón conduce al inicio de un capítulo y la lista desplegable dirige a un tema específico del capítulo.

Además de los temas principales, dentro de algunos capítulos se presentan contenidos de profundización y ayuda como complemento para el aprendizaje en la realización del tutorial. Entre éstos están la instalación, especificaciones de las tarjetas.

El acceso a estos contenidos es por medio de unos botones que aparecen en algunas GUIs ubicados en la parte inferior derecha del área de explicación del contenido.

## ANEXO 2. LISTA DE FUNCIONES Y PROPIEDADES PARA LA TOOLBOX DE ADQUISICIÓN DE DATOS DE MATLAB

### Comandos de información y ayuda

Función	Descripción
daqhelp	Muestra información para objetos de dispositivo, constructores, adaptadores, funciones y Propiedades.
daqhwinfo	Muestra información del hardware de adquisición de datos
propinfo	Retorna características de las propiedades para objetos de dispositivos, canales o líneas

### Funciones

Las funciones de la Toolbox y los objetos de dispositivo con los que están asociados, están organizadas dentro de los grupos mostrados a continuación. Los objetos compatibles incluyen entradas análogas (AI), salidas análogas (AO) y I/O digital (DIO).

Crear objetos de dispositivo		AI	AO	DIO
analoginput	Crea un objeto de entrada análoga	✓		
analogoutput	Crea un objeto de salida análoga		✓	
digitalio	Crea un objeto de I/O digitales			✓

Agregar Canales y líneas		AI	AO	DIO
addchannel	Agrega canales de hardware a un objeto de entrada análoga o salida análoga	✓	✓	
addline	Agrega líneas de hardware a un objeto de I/O digitales			✓
addmuxchannel	Agrega canales de hardware cuando se utiliza una tarjeta multiplexadora (Únicamente de National Instruments)	✓		

Tomar y configurar propiedades		AI	AO	DIO
get	Retorna las propiedades del objeto	✓	✓	✓
set	Configura o muestra las propiedades del objeto	✓	✓	✓
setverify	Configura y retorna la propiedad específica	✓	✓	✓

<b>Ejecutar el objeto</b>		AI	AO	DIO
start	Arranca un objeto	✓	✓	✓
stop	Ejecuta manualmente un disparador	✓	✓	✓
trigger	Configura y retorna la propiedad específica	✓	✓	
wait	Espera que el objeto se detenga	✓	✓	

<b>Trabajar con datos</b>		AI	AO	DIO
flushdata	Remueve datos del motor (memoria) de adquisición de datos	✓		
getdata	Toma datos, tiempo e información del evento desde el motor (memoria) de adquisición de datos	✓		
getsample	Adquiere inmediatamente una muestra	✓		
getvalue	Lee valores desde las líneas			✓
peekdata	Muestra los datos adquiridos más recientes	✓		
putdata	Enfila datos en el motor (memoria) para una salida eventual		✓	
putsample	Toma inmediatamente una muestra		✓	
Putvalue	Escribe valores a las líneas			✓

<b>Tomar información y ayuda</b>		AI	AO	DIO
daqhelp	Muestra ayuda para los objetos, constructores, adaptadores, funciones y propiedades	✓	✓	✓
daqhwinfo	Muestra información del hardware de adquisición de datos	✓	✓	✓
daqpropedit	Abre el editor de propiedades de adquisición de datos	✓	✓	✓
propinfo	Retorna características de propiedades para los objetos, canales o líneas	✓	✓	✓

<b>Propósitos generales</b>		<b>AI</b>	<b>AO</b>	<b>DIO</b>
<code>binvec2dec</code>	Convierte un binvec a un valor decimal			✓
<code>clear</code>	Remueve dispositivos de objetos del área de trabajo de MATLAB	✓	✓	✓
<code>daqcallback</code>	Una función de llamado que muestra Información del evento para el evento especificado	✓	✓	✓
<code>daqfind</code>	Retorna objetos, canales o líneas desde el motor (memoria) hacia el área de trabajo de MATLAB	✓	✓	✓
<code>daqmem</code>	Asigna o muestra recursos de memoria	✓	✓	
<code>daqread</code>	Lee un archivo de la Toolbox de Adquisición de Datos	✓		
<code>daqregister</code>	Registra o borra un adaptador de controlador de hardware	✓	✓	✓
<code>daqreset</code>	Remueve objetos de dispositivo y DLL's de adquisición de datos de la memoria	✓	✓	✓
<code>dec2binvec</code>	Convierte un valor decimal a un binvec	✓		✓
<code>delete</code>	Remueve objetos de dispositivos, canales o líneas del motor (memoria) de adquisición de datos	✓	✓	✓
<code>disp</code>	Muestra información resumida para los objetos de dispositivo, canales o líneas	✓	✓	✓
<code>ischannel</code>	Verifica valores por canales	✓	✓	✓
<code>isdioline</code>	Verifica valores por líneas	✓	✓	✓
<code>isvalid</code>	Determina si un objeto, grupo de canales o grupo de líneas son validos	✓	✓	✓
<code>length</code>	Retorna la longitud de un objeto, grupo de canales o grupo de líneas	✓	✓	✓
<code>load</code>	Carga objetos de dispositivos, canales o líneas dentro del área de trabajo de MATLAB	✓	✓	✓
<code>makenames</code>	Genera una lista descriptiva de canales o nombres de líneas	✓	✓	✓
<code>muxchanidx</code>	Retorna un índice de canales escaneados multiplexados	✓		
<code>obj2mfile</code>	Convierte objetos de dispositivos, canales o líneas a código de MATLAB	✓	✓	✓
<code>save</code>	Guarda objetos de dispositivos a un archivo-MAT	✓	✓	✓
<code>showdaqevents</code>	Muestra información de la bitácora del evento	✓	✓	

### Propiedades base de entradas análogas

Las propiedades base de entradas análogas están divididas en 2 grandes categorías: propiedades comunes y propiedades de canal. Las propiedades comunes aplican a cualquier canal contenido por el objeto de entrada análoga, mientras que las propiedades de canal pueden ser configuradas para canales individuales.

### Propiedades comunes

<b>Propiedades de configuración básica de entradas análogas</b>	
SamplesPerTrigger	Especifica el número de muestras a adquirir por cada miembro del grupo de canales por cada disparador que ocurra
SampleRate	Especifica el rango por canal en el cual los datos análogos son convertidos en datos digitales
TriggerType	Especifica el tipo de disparador a ejecutar

<b>Propiedades de registro de entradas análogas</b>	
LogFileName	Especifica el nombre del archivo del disco en el cual la información es registrada
Logging	Indica si los datos se están registrando en la memoria o en un archivo del disco
LoggingMode	Especifica el destino para la información adquirida
LogToDiskMode	Especifica si los datos, eventos e información de hardware son guardadas en un archivo del disco en múltiples archivos



### **Propiedades del disparador de entrada análoga**

InitialTriggerTime	Indica el tiempo absoluto del primer disparador
ManualTriggerHwOn	Especifica que el dispositivo del hardware comienza cuando un disparador manual es emitido
TriggerFcn	Especifica la función de llamado de un archivo .m para ser ejecutada cuando un disparador ocurra
TriggerChannel	Especifica el canal que sirve como fuente para un disparador
TriggerCondition	Especifica la condición que debe ser satisfecha para que un disparador se ejecute
TriggerConditionValue	Especifica uno o mas valores de voltaje que deben ser satisfechos para que un disparador se ejecute
TriggerDelay	Especifica el valor de retraso para el registro de datos
TriggerDelayUnits	Especifica las unidades en las cuales los datos del retraso del disparador son tomados
TriggerRepeat	Especifica el número de veces adicionales que se ejecuta el disparador
TriggersExecuted	Indica el numero de disparadores que se ejecutan
TriggerType	Especifica el tipo de disparador para ejecutar

### **Propiedades del estado de entradas análogas**

Logging	Indica si los datos están siendo registrados en la memoria o un archivo del disco
Running	Indica si el objeto está corriendo
SamplesAcquired	Indica el numero de muestras adquiridas por canal
SamplesAvailable	Indica el numero de muestras disponibles por canal en el motor (memoria)

### **Propiedades de configuración de hardware de entradas análogas**

ChannelSkew	Especifica el tiempo entre canales de hardware escaneados consecutivos
ClockSource	Especifica el reloj usado para gobernar el valor de conversión de hardware
InputType	Especifica la configuración del canal de hardware de entrada análoga
SampleRate	Especifica el valor por canal en el cual los datos análogos se convierten en datos digitales

<b>Propiedades de llamado de entradas análogas</b>	
DataMissedFcn	Especifica la función de llamado del archivo .m para ser ejecutada cuando los datos no se encuentren
InputOverRangeFcn	Especifica la función de llamado del archivo .m para ser ejecutada cuando los datos adquiridos excedan el valor válido del hardware
RuntimeErrorFcn	Especifica la función de llamado del archivo .m para ser ejecutada cuando ocurra un error en el tiempo de ejecución
SamplesAcquiredFcn	Especifica la función de llamado del archivo .m para ser ejecutada cada vez que un número predefinido de muestras sean adquiridas por cada miembro del grupo de canales
SamplesAcquiredFcnCount	Especifica la función de llamado del archivo .m para ser ejecutada cada vez que un número predefinido de muestras sean adquiridas por cada miembro del grupo de canales
StartFcn	Especifica la función de llamado del archivo .m para ser ejecutada justo antes que el objeto arranque
StopFcn	Especifica la función de llamado del archivo .m para ser ejecutada justo después que el objeto se detenga
TimerFcn	Especifica la función de llamado del archivo .m para ser ejecutada cuando pase un período de tiempo predefinido
TimerPeriod	Especifica el período de tiempo entre eventos de conteo
TriggerFcn	Especifica la función de llamado del archivo .m para ser ejecutada cuando un disparador ocurra

<b>Propiedades de propósito general de entradas análogas</b>	
BufferingConfig	Especifica la memoria asignada por canal
BufferingMode	Especifica como la memoria es asignada
Channel	Contiene canales de hardware agregados al objeto
EventLog	Almacena información de eventos específicos
Tag	Especifica una etiqueta al objeto
Timeout	Especifica un tiempo de espera adicional para extraer datos
Type	Indica el tipo del objeto
UserData	Almacena datos que el usuario quiera asociar con un objeto

## Propiedades de canales

<b>Propiedades de canales de entradas análogas</b>	
ChannelName	Especifica un nombre descriptivo para el canal
HwChannel	Especifica el ID del canal de hardware
Index	Indica el índice de MATLAB® de un canal de hardware
InputRange	Especifica el rango del subsistema de entrada análoga
NativeOffset	Indica la compensación a usar cuando se convierte entre datos en formato nativos y dobles
NativeScaling	Indica la escala a usar cuando se convierte entre datos en formato nativo y dobles
SensorRange	Especifica el rango de datos que se espera del sensor
Units	Especifica la etiqueta de unidades que se están utilizando
UnitsRange	Especifica el rango de datos como unidades

## Propiedades bases de salidas análogas

Las propiedades bases de salidas análogas están divididas en dos categorías principales: propiedades comunes y propiedades de canal. Las propiedades comunes aplican para cada canal contenido por el objeto de salida análoga, mientras que las propiedades de canal pueden ser configuradas para canales individuales.

## Propiedades comunes

<b>Propiedades básicas de configuración de salidas análogas</b>	
SampleRate	Especifica el valor por canal en la cual los datos digitales son convertidos en datos análogos
TriggerType	Especifica el tipo de disparador a ejecutar

<b>Propiedades de disparadores de salida análoga</b>	
InitialTriggerTime	Indica el tiempo absoluto para el primer disparador
TriggerFcn	Especifica la función de llamado del archivo .m para ser ejecutada cuando un disparo ocurre
TriggersExecuted	Indica el numero de disparadores que ejecuta
TriggerType	Especifica el tipo de disparador a ejecutar

<b>Propiedades de estado de salidas análogas</b>	
Running	Indica si el objeto está corriendo
TriggerType	Especifica el tipo de disparador a ejecutar
SamplesAvailable (memoria)	Indica el número de muestras disponibles por canal en el motor
SamplesOutput (memoria)	Indica el numero de salida de muestras por canal del motor
Sending	Indica si los datos están siendo enviados al dispositivo de hardware

<b>Propiedades básicas de configuración de salidas análogas</b>	
ClockSource	Especifica el reloj usado para gobernar el valor de conversión de hardware
SampleRate	Especifica el valor por canal en el cual los datos digitales son convertidos en datos análogos

<b>Propiedades de administración de datos de salida análoga</b>	
MaxSamplesQueued	Indica el número máximo de muestras que pueden ser enlistadas en el motor (memoria)
RepeatOutput	Especifica el número de de tiempos adicionales en que salen los datos enlistados
Timeout	Especifica un tiempo adicional para enlistar datos

<b>Propiedades llamado de salida análoga</b>	
RuntimeErrorFcn	Especifica la función de llamado del archivo .m para ser ejecutada cuando ocurra un error en el tiempo de ejecución
HwChannel	Especifica el ID del canal de hardware
Index	Indica el índice de MATLAB de un canal de hardware
InputRange	Especifica el rango del subsistema de entrada análoga
NativeOffset	Indica la compensación a usar cuando se convierte entre datos en formato nativos y dobles
NativeScaling	Indica la escala a usar cuando se convierte entre datos en formato nativo y dobles
SensorRange	Especifica el rango de datos que se espera del sensor
Units	Especifica la etiqueta de unidades que se están utilizando
UnitsRange	Especifica el rango de datos como unidades

<b>Propiedades de propósito general de salida análoga</b>	
BufferingConfig	Especifica la memoria asignada por canal
BufferingMode	Especifica como la memoria es asignada
Channel	Contiene canales de hardware añadidos al objeto
EventLog	Almacena información de los eventos específicos
Name	Especifica un nombre para el objeto
OutOfDataMode	Especifica como el valor mantenido por el subsistema de salida análoga es determinado
Tag	Especifica una etiqueta al objeto
Type	Indica el tipo del objeto
UserData	Almacena datos que se quieran asociar con un objeto

<b>Propiedades de canales de salida análoga</b>	
ChannelName	Especifica un nombre para el canal
DefaultChannelValue	Especifica el valor mantenido por el subsistema de salida análoga
HwChannel	Especifica el ID del canal de hardware
Index	Indica el índice MATLAB de un canal de hardware
NativeOffset	Indica el offset a usar cuando se convierten datos entre formato nativo y dobles
NativeScaling	Indica el escala a utilizar cuando se convierten datos entre formato nativo y dobles
OutputRange	Especifica el rango de salida análoga del subsistema hardware
Units	Especifica la etiqueta de unidades de ingeniería
UnitsRange	Especifica el rango de data como unidades de ingeniería

### **Propiedades base de I/O digital**

Las propiedades base de I/O digital se dividen en 2 categorías principales: propiedades comunes y propiedades de línea. Las propiedades comunes aplican para cada línea contenida en el objeto de entrada/salida digital, mientras que las propiedades de línea pueden ser configuradas para líneas individuales.

## Propiedades comunes

<b>Propiedades comunes de I/O digital</b>	
Line	Contiene líneas de hardware agregadas al objeto
Name	Especifica un nombre para el objeto
Running	Indica si el objeto está corriendo
Tag	Especifica una etiqueta para el objeto
TimerFcn	Especifica la función de llamado del archivo .m para ser ejecutada cuando pase un periodo de tiempo predeterminado
TimerPeriod	Especifica el periodo de tiempo entre eventos
Type	Indica el tipo de objeto
UserData	Almacena datos que se quieren asociar con un objeto

## Propiedades de línea

<b>Propiedades comunes de I/O digital</b>	
Direction	Especifica si una línea es usada para entrada o salida
HwLine	Especifica el ID de una línea de hardware
Index	Indica el índice MATLAB de una línea de hardware
LineName	Especifica un nombre para una línea
Parent	Indica el objeto (parent) de una línea
Port	Especifica el ID del puerto

### ANEXO 3. EJEMPLOS RESUELTOS PRESENTADOS EN EL TUTORIAL

Como se mencionó anteriormente, los capítulos 3, 4 y 5 culminan con unos ejemplos resueltos, donde se muestra el uso de las propiedades y funciones básicas para realizar una sesión de adquisición. Estos ejemplos son mostrados a continuación con sus respectivas soluciones:

#### EJEMPLOS DE SALIDA DIGITAL

##### Ejemplo 1:

Este ejemplo ilustra los pasos básicos para escribir en un objeto digital (DIO) de salida individual, dos valores digitales con un tiempo de espera de 2 segundos entre ellos.

```
DIO=digitalio('nidaq','Dev1');
Linea=addline(DIO,0,'out');
n=2;
putvalue(DIO,0)
pause(n)
putvalue(DIO,1)
delete(DIO)
clear DIO
```

##### Ejemplo 2:

En este ejemplo se tiene un arreglo de decimales (conteo del 0 al 15) que son escritos en un objeto digital (DIO) de salida en bus, que contiene las líneas del 0 al 3 del puerto 1 de la tarjeta.

```
DIO=digitalio('nidaq','Dev1');
Lineas=addline(DIO,0:3,1,'out');
n=1.5;
datos=[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15];
for i=1:length(datos)
```

```

        putvalue (DIO, datos (i))
        pause (n)
    end
    delete (DIO)
    clear DIO

```

## EJEMPLOS DE ENTRADA DIGITAL

### Ejemplo 1:

En este ejemplo se crea un objeto digital DIO y se le agrega la línea 5 del puerto 0 con dirección de entrada, luego se crea un ciclo repetitivo, para extraer 10 veces el valor que hay en la entrada y se guarda todos los valores en una variable.

```

DIO = digitalio('nidaq','Dev1');
linea = addline(DIO,5,0,'in');
n=2;
a=10;
for i=1:a
    valores(i,:) = getvalue(DIO);
    pause(n)
end
delete (DIO)
clear DIO

```

### Ejemplo 2:

En este ejemplo se tiene un objeto digital (DIO), al cual se le agrega las líneas del 0 al 3 del puerto 0 como salidas y la línea 0 del puerto 1 como entrada (la línea toma el índice 5 de MATLAB).

Mientras el valor en la línea de entrada es 1, se realiza un conteo regresivo del 9 al 0 con las líneas de salida conectadas al decodificador del display 7 segmentos de cátodo común.



```

DIO=digitalio('nidaq','Dev1');
Salidas=addline(DIO,0:3,'out');
Entrada=addline(DIO,0,1,'in','entrada');
datos=[9 8 7 6 5 4 3 2 1 0];
while getvalue(DIO.line(5))==1
for i=1:length(datos)
    putvalue(DIO.line(1:4),datos(i))
    pause(1)
end
end
delete (DIO)
clear DIO

```

### Ejemplo 3:

En este ejemplo se usa un TIMER para muestrear continuamente las líneas del 0 al 4 de un objeto digital DIO, hasta que sea detenido manualmente con la función STOP (stop(tim)) en la ventana de comandos de MATLAB.

Cada 2 segundos se cumple un periodo donde llama una función de nombre “ent”, en la cual se extrae y guarda los valores en una variable en forma de matriz. Esto hace que el objeto esté muestreando continuamente hasta que el TIMER es detenido.

```

a=0;
tim=timer;
tim.TimerFcn='ent';
tim.Period=2;
tim.ExecutionMode='FixedDelay';

DIO=digitalio('nidaq','Dev1');
Entradas=addline(DIO,0:4,'in');
start(tim)

```

#### - Función “ent”:

```

a=a+1;
datos=getvalue(DIO);

```

## EJEMPLOS DE SALIDA ANÁLOGA

### Ejemplo 1:

Se crea un objeto de salida análoga AO de un canal, para sacar una señal seno de 0 a 5Voltios, creada con la función Linspace, como un vector fila de 100 valores desde 0 hasta  $2\pi$ .

```
AO = analogoutput('nidaq', 'Dev1');
Canal = addchannel(AO,0);
datos = 2.5*sin(linspace(0,2*pi,100))+2.5;
pause(1)
for i=1:length(datos)
    putsample(AO,datos(i))
end
delete (AO)
clear AO
```

### Ejemplo 2:

Por medio de un Timer con un periodo de 1.5 segundos, escribir dos vectores de datos a un objeto de salida análoga de dos canales. La interrupción se detiene cuando todos los datos son enviados totalmente. Se crea el archivo salida.m como función del Timer.

```
tim=timer;
tim.TimerFcn='salida';
tim.Period=1.5;
tim.ExecutionMode='fixedDelay';

a=0;
vector1=[1.6 2 3 4 5 4 2.3 2.5];
vector2=[4 5 3.2 2 4.7 3.1 2 1];
AO=analogoutput('nidaq', 'Dev1');
Canales = addchannel(AO,0:1);
start(tim)
```

- Función “salida”:

```
a=a+1;
putsample(AO,[vector1(a) vector2(a)])
if a==length(vector1)
    stop(tim)
end
```

## EJEMPLOS DE ENTRADA ANÁLOGA

### Ejemplo 1:

Se crea un objeto de entrada análoga (AI), con un canal de nodo simple, para muestrear durante un segundo la salida del generador de onda seno (F=90Hz) del modulo de pruebas, a una tasa de muestreo de 1000Hz y se extraen 200 muestras del motor de adquisición para visualizarlas.

```
AI = analoginput('nidaq','Dev1');
Canal=addchannel(AI,0:1);

duracion=1;
set(AI,'InputType','SingleEnded')
set(AI,'SampleRate',1000)
TasaActual=get(AI,'SampleRate');
set(AI,'SamplesPerTrigger',TasaActual*duracion)

start(AI)
wait(AI,duracion+1)

[data,t] = getdata(AI,200);
figura
plot(t,data),
grid on
delete(AI)
clear AI
```

### Ejemplo 2:

En este ejemplo se inicia la adquisición de datos con un objeto de entrada análoga individual con el canal 0 durante 1 segundo, justo después que un flanco positivo es detectado en el trigger digital (PFI0). Una vez terminada la adquisición se extraen los datos del motor para la visualización de la señal de entrada.

```
AI = analoginput('nidaq', 'Dev1');
addchannel(AI,0);
set(AI, 'TriggerType', 'HwDigital');
set(AI, 'TriggerCondition', 'PositiveEdge');

start(AI)

wait(AI,10)
data=getdata(AI,100);
plot(data);

delete(AI)
clear AI
```

### Ejemplo 3:

Con un objeto de entrada análoga de un canal se quiere hacer una previsualización de los datos mientras éstos son extraídos, con una tasa de muestreo de 1KHz y un trigger manual.

```
AI = analoginput('nidaq', 'Dev1');
chan = addchannel(AI,0);

duracion = 10;
set(AI, 'SampleRate', 1000)
TasaActual = get(AI, 'SampleRate');
set(AI, 'SamplesPerTrigger', duracion*TasaActual)
set(AI, 'TriggerType', 'Manual')
```

```

previsualizacion = duracion*TasaActual/100;
subplot(211)
P = plot(zeros(previsualizacion,1)); grid on
title('Previsualización de los Datos')
xlabel('Muestras')
ylabel('Tensión(Volts)')

start(AI)
trigger (AI)

while AI.SamplesAcquired < previsualizacion
end
while AI.SamplesAcquired < duracion*TasaActual
    datos = peekdata(AI,previsualizacion);
    set(P, 'ydata', datos)
    drawnow
end

wait(AI,duracion+100)

datos=getdata(AI,300);

subplot(212), plot(datos), grid on
title('Datos Adquiridos')
xlabel('Muestras')
ylabel('Tensión(volts)')

delete(AI)
clear AI

```

#### Ejemplo 4:

Este ejemplo muestra la forma para iniciar una adquisición con un objeto de entrada análoga de dos canales cuando la señal seno conectada al canal 0 pase por 1.5 voltios, con un retardo del trigger de 0.5 segundos antes que se ejecute el trigger.

Este ejemplo muestra la forma para iniciar una adquisición con un objeto de entrada análoga de dos canales y un trigger por software; el trigger se activa cuando la señal conectada al canal 0 pasa por 1.5 voltios con pendiente positiva y toma datos durante 0.2 segundos antes que el trigger se ejecute; este proceso se repite una vez.

```
AI = analoginput('nidaq', 'Dev1');
chan1 = addchannel(AI,0); %Señal seno
chan2 = addchannel(AI,1); %Señal cuadrada

duracion = 0.1;
set(AI, 'SampleRate', 1000)
ActualRate = get(AI, 'SampleRate');
set(AI, 'SamplesPerTrigger', ActualRate*duracion)

% Configuración del trigger por software
Num_Arranque = 1.5;
set(AI, 'TriggerChannel', chan1)
set(AI, 'TriggerType', 'Software')
set(AI, 'TriggerCondition', 'Rising')
set(AI, 'TriggerConditionValue', Num_Arranque)
set(AI, 'TriggerRepeat', 1)
set(AI, 'TriggerDelayUnits', 'Samples')
set(AI, 'TriggerDelay', -100)

start(AI)
wait(AI, duracion+1)

[d1,t1] = getdata(AI);
```

```
[d2,t2] = getdata(AI);  
subplot(211), plot(t1,d1), grid on,  
title('Muestras del Primer Trigger')  
subplot(212), plot(t2,d2), grid on  
title('Muestras del Segundo Trigger')  
delete(AI)  
clear AI
```