CAPÍTULO 3. INTERFASE CON EL USUARIO

3.1 Interfaz con la computadora

La etapa en dónde se hace la transformación de la señal analógica a digital es en la tarjeta de adquisición de datos NI-USB6008, ésta además de capturar la señal analógica obtenida de los amplificadores de instrumentación comunica al sistema electrónico con una computadora personal por medio de un puerto USB (*Universal Serial Bus*).

El *software* de *National Instruments Labview* es capaz de interpretar la señal digital, manipularla y enviar señales de control mediante una tarjeta de adquisición de la misma familia. Es precisamente esta parte lo que diferencia instrumentación electrónica de virtual. En la figura 3.1 se muestra el bloque gráfico de *Labview* llamado *DAQ Assitant*. Este bloque ayuda a configurar el Instrumento Virtual (concepto desarrollado por *National Instruments* para referirse a un programa de *Labview*).

Fig. 3.1 Bloque *DAQ Assistant*

En este bloque podemos configurar entradas, salidas, frecuencia, rango de voltaje de entrada, canales utilizados, si son analógicos o digitales, configuración de las terminales, escalas, diagrama de conexión, modo de adquisición, parámetros de reloj (esto se refiere a frecuencia de muestreo y número de muestras).

Para la realización de este sistema se configuraron los siguientes parámetros:

1. Se utilizaron 8 canales nombrados “Entrada” para cada una de las señales del sistema de adquisición de datos. Se configuraron para que estuvieran referenciadas a tierra (RSE) para utilizar al 100% la capacidad de la tarjeta de adquisición de datos. Ver figura 3.2.

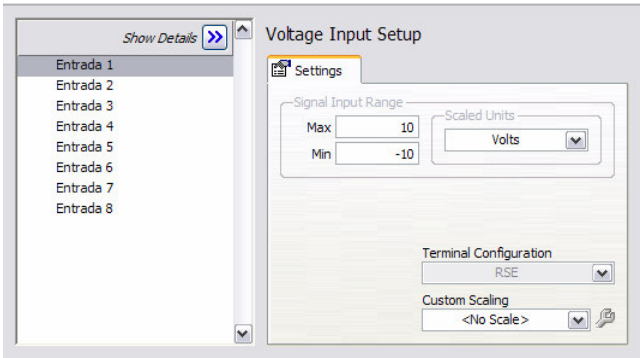

Fig. 3.2 Configuración de los 8 canales

2. Se configuró el rango de voltaje de entrada de -10 a 10 volts, para observar si hay un *offset* negativo.

3. La velocidad de muestreo es 2000 muestras a 1 kHz en modo continuo. Esto con el propósito de actualizar la información de manera rápida y utilizar al 100% la capacidad del sensor. Ver figura 3.3.

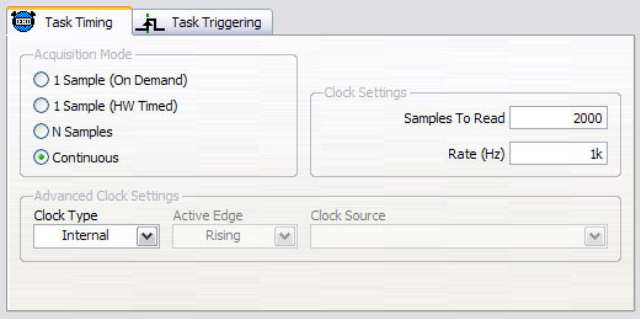

Fig. 3.3 Velocidad de muestreo

3.2 Programación en *NI-Labview*.

3.2.1 ¿Qué es *Labview*?

Labview es un ambiente de programación diseñado para crear interfaces con cualquier instrumento de medición. *Labview* reúne asistentes interactivos, generación de códigos y conectividad con diversos dispositivos; gracias a esto último *Labview* puede ser incorporado a sistemas existentes. Este *software* es capaz de adquirir y generar señales desde dispositivos USB, tarjetas de entrada como PCI (Componente Periférico de Interconexión) y sistemas basados en *Ethernet*. Otra característica es, que cuenta con más de 600 funciones para la síntesis de señales, análisis de frecuencia, probabilidad, estadística, matemáticas, ajuste de curva, interpolación, procesamiento digital y análogo. Se pueden extender las aplicaciones a más específicas para sonido, vibraciones, máquinas de visión, comunicaciones de radiofrecuencia y otras. Después de adquirir, procesar y analizar los datos, *Labview* es capaz de mostrar la información de una forma sencilla y exacta. Genera reportes, guarda la información obtenida, puede publicar ésta en Internet, cuenta con conectividad con bases de datos y otros [17].

3.2.2 Instrumento Virtual (VI)

Un programa hecho en *Labview* se denomina instrumento virtual o VI, esto es porque tiene la capacidad de sustituir a un multímetro, osciloscopio o analizador de espectros.

En esta sección se explicará detalladamente cada parte del VI, pero antes, se muestran los bloques de *Labview* utilizados, dando su nombre y funcionamiento.

El instrumento virtual desarrollado en esta tesis es algo complejo, ya que reúne elementos de programación de alto nivel, multiplexado, un sistema de adquisición de datos, desplegado de información en diferentes formas, análoga y digital, y finalmente la parte de captura y escritura en un archivo; por lo que para explicarlo vamos a dividirlo en diferentes etapas. Ver figura 3.4.

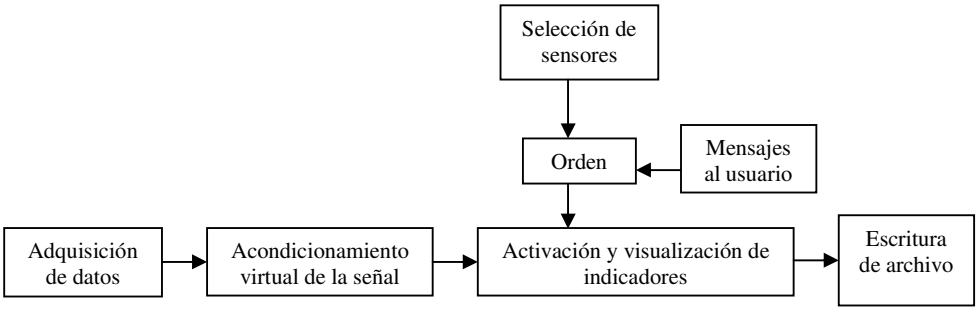

Fig. 3.4 Etapas de programación en *Labview*.

La primer etapa que es la de adquisición de datos realizada con ayuda del asistente de adquisición de datos de *Labview* como fue explicado en la sección 3.1.

3.2.3 Acondicionamiento virtual de la señal

Para poder acondicionar la señal, que es la segunda etapa de este VI, fue necesario utilizar un ajustador de curva, esto se hizo con la finalidad de reunir todas las muestras tomadas en un tiempo determinado y ajustar una curva. Se configuró un modelo polinomial de 5º orden que es el que da un mejor resultado. Con otros había problemas de ajuste o de tiempo de procesamiento, es decir, volvía el programa más lento. Ver figuras 3.5 y 3.6.

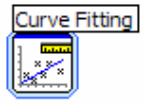

Fig. 3.5 Bloque Ajustador de curva

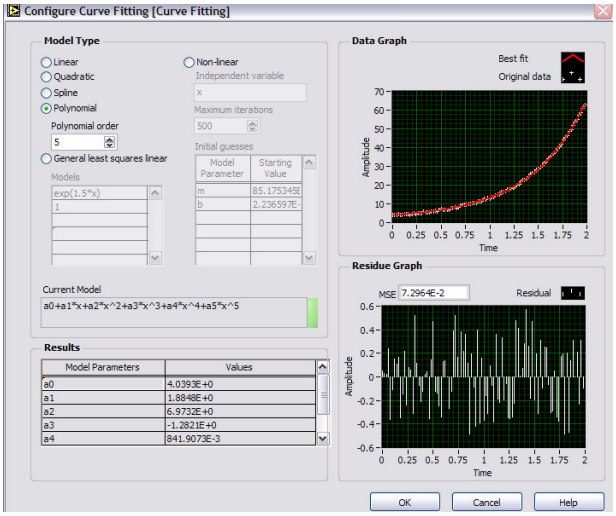

Fig. 3.6 Ajuste de curva con modelo polinomial de 5º orden

Posteriormente hubo que hacer un filtrado, para evitar ruido indeseable; como se trabajó en 1 kHz, se implementó un filtro pasabanda centrado en esta frecuencia. Se configuró la frecuencia de corte inferior en 999 Hz y la superior en 1001 Hz utilizando la topología Butterworth de 3er. orden, con el objetivo de evitar la lectura de alguna señal en cualquier otra frecuencia, por ejemplo el ruido electromagnético que generan las lámparas en donde está localizada la planta hidráulica. Ver figuras 3.7 y 3.8.

Fig. 3.7 Bloque *Filter*

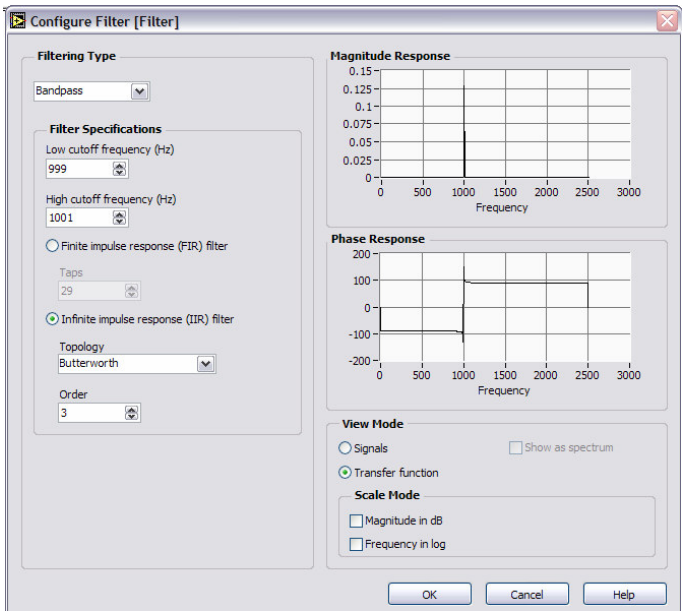

Fig. 3.8 Configuración de filtro pasabanda *Butterworth* de 3er. orden.

La señal tuvo que ser acondicionada para que nos brindara valores en psi. La fórmula aplicada fue implementada de la siguiente manera: La señal entregada por el sistema de adquisición de datos es de 0 a 4 V proporcional de 0 a 50 kPa, así que para obtener el factor por el cual hay que multiplicar la señal para obtener el valor de presión equivalente a 1 V efectuamos la siguiente operación:

$$(4V)X = 50\times10^{3}\,Pa$$

$$X = \frac{50\times10^{3}\,Pa}{4V} \tag{3.1}$$

$$X = 12.5\frac{kPa}{V}$$

El valor solicitado por el departamento de Ingeniería Civil y Ambiental es psi $\left(\frac{lb}{in^2}\right)$, así que se tiene que hacer la conversión. Utilizando la ecuación 1.2, la cual dice que 1 kilo Pascal (kPa) es igual a 0.145 libras sobre pulgadas cuadradas (psi), obtenemos el siguiente factor multiplicativo, así que la señal de voltaje que obtengamos del sistema de adquisición de datos (X) vamos a multiplicarla de la siguiente forma:

$$X\left(12.5\frac{kPa}{V}\right)\left(0.145\frac{psi}{kPa}\right) = \text{ medición en psi} \tag{3.2}$$

En la figura 3.9 se muestra el bloque de *Labview* para representar a la fórmula y en la figura 3.10 cómo se configuró este bloque.

Fig. 3.9 Bloque *Formula*

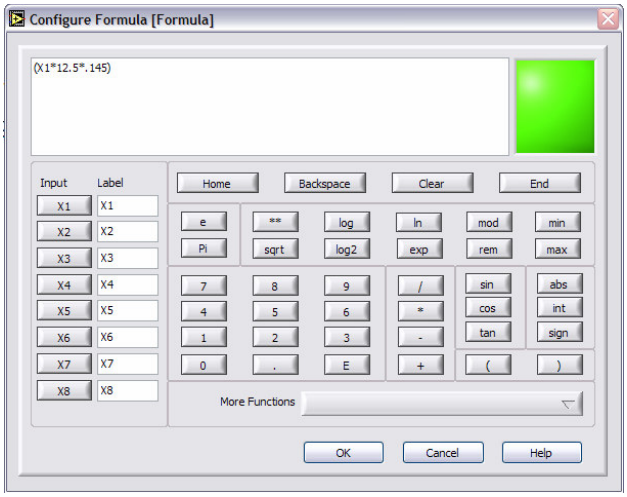

Fig. 3.10 Configuración del bloque fórmula para obtener una medición real en psi

Ahora se hace la caracterización del sistema de instrumentación virtual, utilizando los valores antes mencionados para obtener valor en psi y no en volts. Se obtuvo la siguiente tabla de valores:

CAMARA	MUETRAS DE PRESIÓN EN EL SISTEMA DESARROLLADO (PSI)									
TRIAXIAL	1	2	3	4	5	6	7	8	9	10
1 psi	1.6	1.3	1.2	1.3	1.6	1.3	1.4	1.5	1.4	1.5
2 psi	3	2.9	3	2.9	2.8	2.9	3	3	3	2.9
3 psi	4.6	4.3	4.3	4.5	4.5	4.3	4.4	4.5	4.6	4.5
4 psi	6.2	5.9	5.8	5.9	6.1	6.1	5.9	5.9	5.9	6
5 psi	7.8	7.3	7.5	7.6	7.8	7.7	7.8	7.7	7.8	7.7
6 psi	9.3	8.8	9	9.1	9.2	9	9	9.1	9.1	9.1
7 psi	10.7	10.6	10.6	10.9	10.3	10.7	10.8	10.6	10.5	10.7

Tabla 3.1 Pruebas realizadas variando 1 psi.

Como podemos observar el primer resultado no fue muy satisfactorio, ya que los valores no son parecidos a los obtenidos en el medidor de presión *Spec Scientific 840065*. Al obtener la gráfica de todas las muestras se puede observar una desviación de la curva del sistema desarrollado contra la del medidor de presión. Ver figura 3.11.

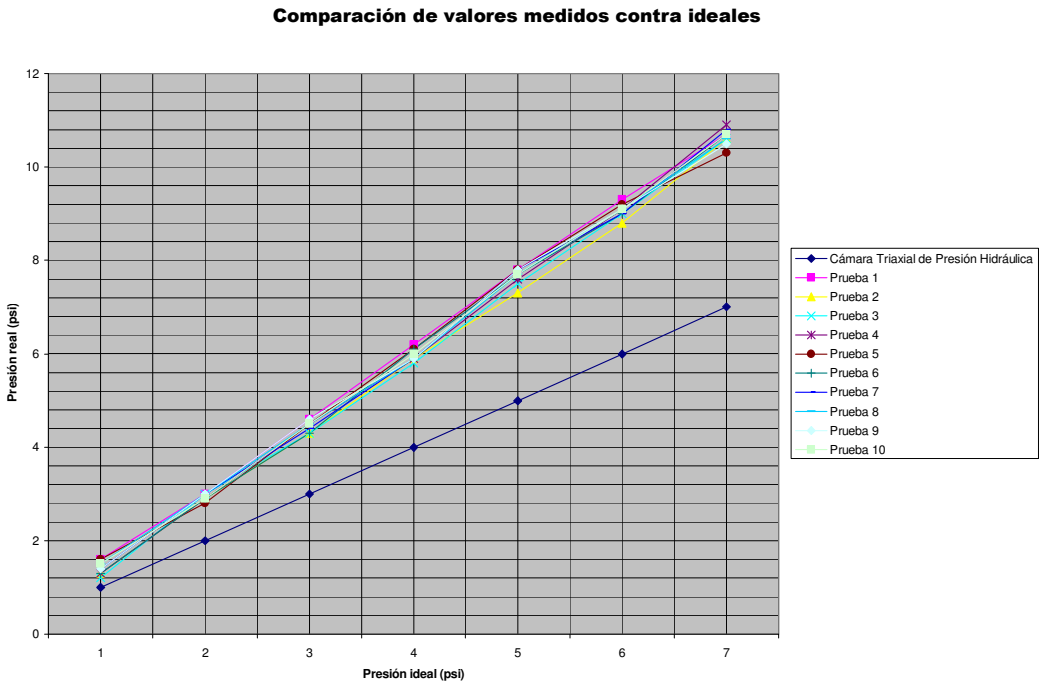

Fig. 3.11 Comparación de valores medidos contra ideales

En la figura 3.11 se observa que hay cierta linealidad entre las 10 pruebas que se hicieron; así como también se puede observar una notoria desviación respecto a la curva ideal, que es la obtenida por el medidor de presión antes mencionado.

3.3 Etapa de acondicionamiento de la señal

En la etapa anterior, realizamos la caracterización del sistema desarrollado en esta tesis y obtuvimos una curva desviada de la ideal, ésta cuenta con un *offset* y una pendiente. En esta sección vamos a estudiar este comportamiento y se va a realizar el ajuste de curva para obtener una muy aproximada a la ideal. .

Analizando los valores de la tabla 3.1 y la figura 3.11, observamos que éstos varían en cada prueba a una presión constante, por ejemplo, cuando fijamos la presión a 1 psi, obtenemos:

CAMARA	MUETRAS DE PRESIÓN EN EL SISTEMA DESARROLLADO (PSI)									
TRIAXIAL	1	2	3	4	5	6	7	8	9	10
1 psi	1.6	1.3	1.2	1.3	1.6	1.3	1.4	1.5	1.4	1.5

Tabla 3.2 Valores de presión en el sistema desarrollado a 1 psi constante.

Hay que tomar en cuenta que el medidor de presión con el que hicimos la calibración del sistema desarrollado, sólo nos brinda valores pares, por lo que cuando este marca 1 psi, es probable que la medición real sea 0.9 psi, o que la presión esté oscilando entre 0.9 y 1 psi, y el medidor de presión esté haciendo un redondeo a 1 psi. Así, las variaciones en el sistema desarrollado son mínimas si tomamos esto en cuenta.

Para poder hacer el ajuste de curva, lo primero que se hizo fue obtener un promedio para cada presión, de todas las muestras obtenidas de la siguiente forma:

	CÁMARA TRIAXIAL	MUETRAS DE PRESIÓN EN EL SISTEMA DESARROLLADO (PSI)										
		A	B	C	D	E	F	G	H	I	J	
1	1 psi	1.6	1.3	1.2	1.3	1.6	1.3	1.4	1.5	1.4	1.5	$\bar{x}1$
2	2 psi	3	2.9	3	2.9	2.8	2.9	3	3	3	2.9	$\bar{x}2$
3	3 psi	4.6	4.3	4.3	4.5	4.5	4.3	4.4	4.5	4.6	4.5	$\bar{x}3$
4	4 psi	6.2	5.9	5.8	5.9	6.1	6.1	5.9	5.9	5.9	6	$\bar{x}4$
5	5 psi	7.8	7.3	7.5	7.6	7.8	7.7	7.8	7.7	7.8	7.7	$\bar{x}5$
6	6 psi	9.3	8.8	9	9.1	9.2	9	9	9.1	9.1	9.1	$\bar{x}6$
7	7 psi	10.7	10.6	10.6	10.9	10.3	10.7	10.8	10.6	10.5	10.7	$\bar{x}7$

Tabla 3.3 Tabla para calcular los promedio.

Variable	Fórmula	Promedio ($\bar{x}$)
$\bar{x}1$	(A1 + B1 + C1 + D1 + E1 + F1 + G1 + H1 + I1 + J1) / 10	1.41
$\bar{x}2$	(A2 + B2 + C2 + D2 + E2 + F2 + G2 + H2 + I2 + J2) / 10	2.94
$\bar{x}3$	(A3 + B3 + C3 + D3 + E3 + F3 + G3 + H3 + I3 + J3) / 10	4.45
$\bar{x}4$	(A4 + B4 + C4 + D4 + E4 + F4 + G4 + H4 + I4 + J4) / 10	5.97
$\bar{x}5$	(A5 + B5 + C5 + D5 + E5 + F5 + G5 + H5 + I5 + J5) / 10	7.67
$\bar{x}6$	(A6 + B6 + C6 +D6 + E6 + F6 + G6 + H6 + I6 + J6) / 10	9.07
$\bar{x}7$	(A7 + B7 + C7 + D7 + E7 + F7 + G7 + H7 + I7 + J7) / 10	10.64

Tabla 3.4 Tabla de fórmulas para calcular los promedio.

Con los promedios o medias aritméticas obtenidas, se obtiene la siguiente gráfica:

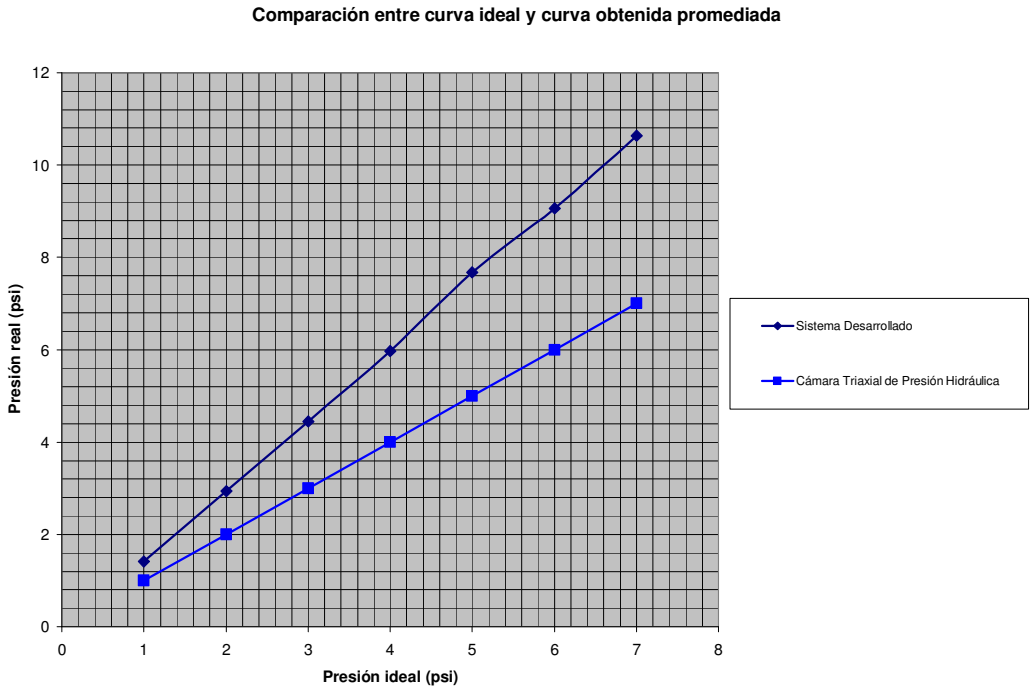

Fig. 3.12 Comparación de valores medios contra ideales.

Con los datos medios o estandarizados de la respuesta del sistema desarrollado, suponemos una recta, ya que estos presentan un comportamiento muy lineal, es entonces necesario obtener la ecuación de ambas rectas. Con la fórmula de una recta, obtenemos cada ecuación.

$$y - y_1 = m(x - x_1) \tag{3.3}$$

$$m = \frac{y_2 - y_1}{x_2 - x_1} \tag{3.4}$$

Donde m es la pendiente, x_1 es un valor sobre la recta perteneciente al eje de las abscisas y x_2 es otro valor en el mismo eje, sólo tiene que cumplirse que $x_1 < x_2$. Lo mismos sucede con y_1 y y_2, son valores sobre la recta pero estos son pertenecientes al eje de las ordenadas; también tiene que cumplirse que $y_1 < y_2$.

Tomando los valores extremos de la gráfica encontramos que: $x_1=1$, $x_2=7$, $y_1=1.41$, $y_2= 10.64$. Al sustituirlos en la fórmula obtenemos:

$$m = \frac{10.64 - 1.41}{7 - 1} = 1.5383 \tag{3.5}$$

$$y - 1.41 = 1.5383(x - 1) \tag{3.6}$$

$$y = 1.5383x - 0.1283 \tag{3.7}$$

La ecuación de la recta del comportamiento de la cámara triaxial de presión hidráulica es completamente lineal y proporcional, es decir, para cada valor de x, le corresponde el mismo valor de y.

$$x = y \tag{3.8}$$

Para comprobar nuestros resultados y hacerlos más exacto, utilizamos la ayuda de la paquetería *Microsoft Excel*. Éste además de graficar, obtiene la ecuación de la recta automáticamente.

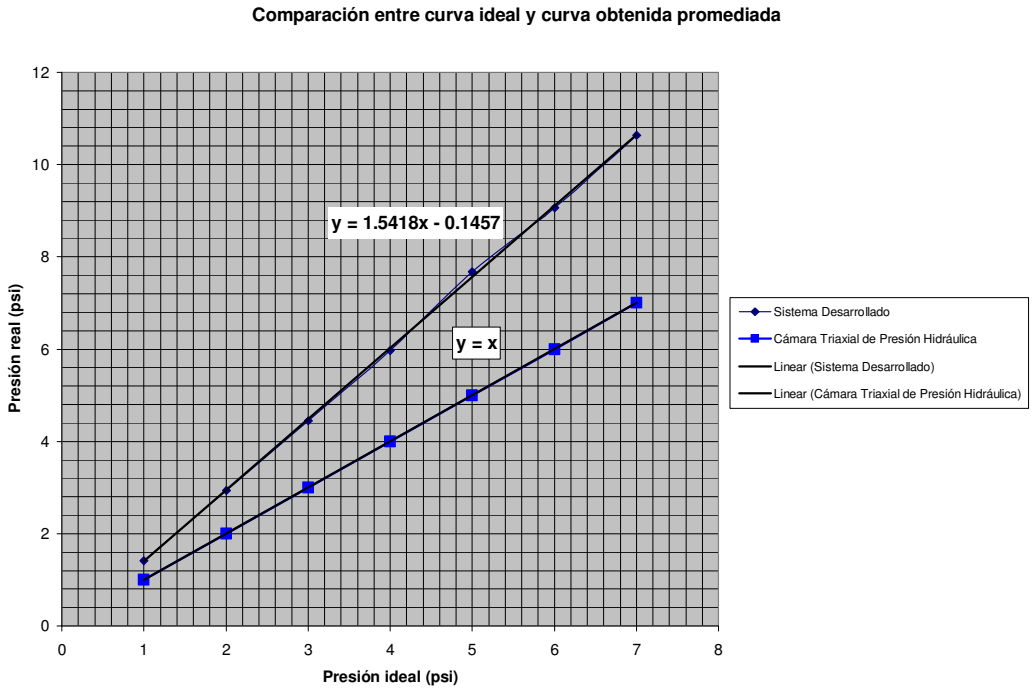

Fig. 3.13 Cálculo de las ecuaciones de las rectas.

Como podemos observar los resultados obtenidos se asemejan a los calculados por Excel, además de que la gráfica obtenida con el sistema desarrollado, se aproxima bastante a una recta. Ahora necesitamos encontrar un factor que haga que la recta obtenida con el sistema desarrollado se iguale a la recta ideal.

$$y = 1.5418x - 0.1457 \tag{3.9}$$

$$x = y \tag{3.10}$$

Ahora tenemos que hacer que la señal que obtuvimos, con el sistema desarrollado, mostrada en la figura 3.13, se comporte como la recta ideal, a cada valor de "x" le corresponda uno igual a "y", por lo que tenemos que manipular la ecuación 3.9 para que se comporte como la ecuación 3.10. Es necesario obtener un valor para el bloque *Formula* de *Labview*, con el fin de hacer un ajuste de curva. Si aplicamos las operaciones inversas a la ecuación 3.9 obtenemos el factor que estamos buscando. Ver ecuación 3.11.

$$y = \left(\frac{1}{1.5418}\right)1.5418x - 0.1457 + 0.1457 = x \tag{3.11}$$

$$\frac{A}{1.5418} + 0.1457 \tag{3.12}$$

Donde A es el valor obtenido previamente en la ecuación 3.2. Es así que el bloque de *Labview* llamado fórmula, hay que introducir la siguiente fórmula:

$$\frac{X(12.5)(0.145)}{1.5418} + 0.1457 \quad \begin{array}{l}\text{= medición en psi} \\ \text{con curva ajustada a la ideal}\end{array} \tag{3.13}$$

En la figura 3.14 se observa el cambio realizado al bloque Fórmula del sistema virtual en *Labview*.

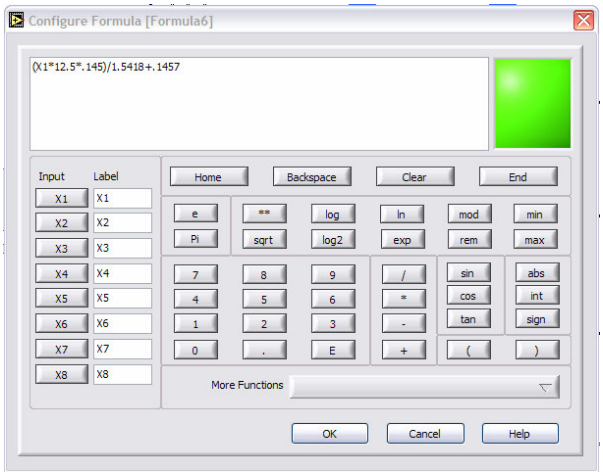

Fig. 3.14 Fórmula para ajuste de curva en bloque Formula de *Labview*

3.3.1 Selección de sensores

En esta etapa de la programación, se tiene que seleccionar 8 de los 63 sensores colocados en la planta hidráulica, por lo se tuvo que hacer una planta virtual con sus 63 sensores como se muestra en a figura 3.15.

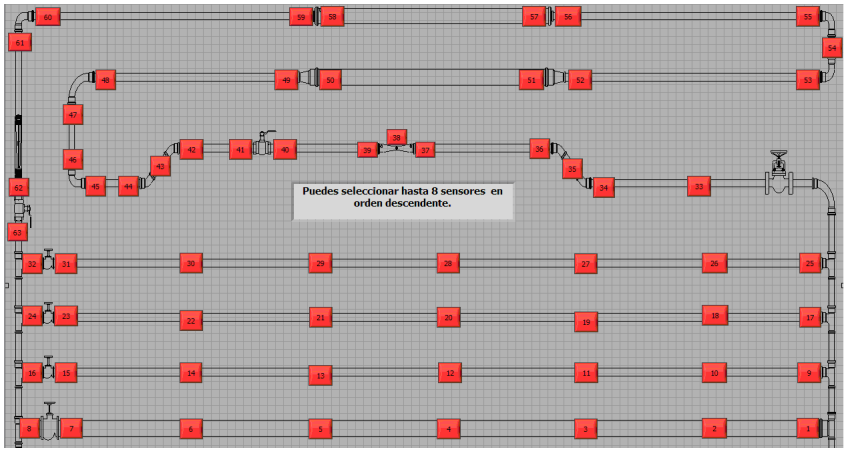

Fig. 3.15 Diagrama de la planta hidráulica con 63 botones.

Para realizar la selección del sensor que se quiere medir, primero se colocaron 63 botones que representan cada uno de los sensores, además de su ubicación. Después se etiquetaron con el número de sensor correspondiente. En el código del programa se observa que hay una red de botones conectados todos a un arreglo con dimensiones 1×63 como se puede observar en las figuras 3.19 y 3.20.

Fig. 3.16 Bloque Botón OK

Fig. 3.17 Bloque de selección

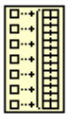

Fig. 3.18 Bloque para crear arreglos unidimensionales

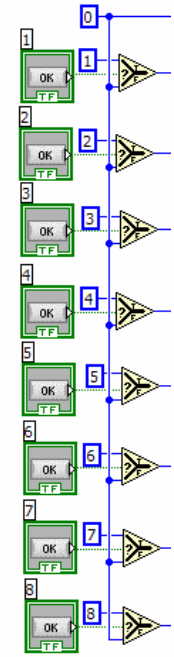

Fig. 3.19 Botones con lógica *booleana.*

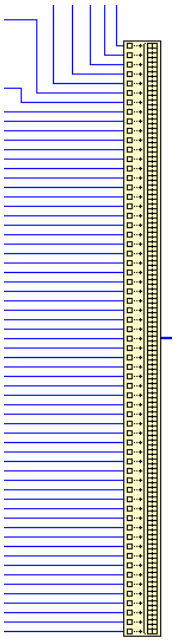

Fig. 3.20 Arreglo con dimensiones 1 X 63.

La lógica *booleana* aplicada en esta parte de la programación es la siguiente: Al no presionar el botón, éste manda un cero lógico al bloque de selección el cual activa un cero decimal. Al presionar el botón, éste manda un uno lógico el cual activa el número decimal que identifica a cada sensor en el bloque de selección. Todos los números decimales son mandados al arreglo, el cual los va a almacenar en el orden que fueron conectados a éste. En otras palabras, se van a guardar en desorden. Es importante hacer la aclaración que *Labview* trabaja con diferentes tipos de variables como por ejemplo decimales, de 32 y 64 bits, así como variables *booleanas*, cuerdas, y muchas otras. Para realizar operaciones de cualquier clase, es necesario trabajar con variables del mismo tipo, si no fuera así, *Labview* marca error. Es por esto que se utilizó un selector que se activa con variables *booleanas* y manda valores decimales.

3.3.2 Orden

En esta etapa se ordenan los valores almacenados en el arreglo anterior. Se colocó un bloque que hace el orden de forma ascendente automática. Como el arreglo es de una dimensión, se va a comportar como un vector, así que al ordenarlos de forma ascendente, los botones que no se presionen quedan en las primeras posiciones y los botones con valor decimal mayor en las últimas. Por ejemplo, si presionáramos 60, 45, 32, 6, 1 en un arreglo de 1 X 8, los valores quedarían acomodados de la siguiente forma:

32
0
0
6
60
0
45
1

Fig. 3.21 Arreglo a[7] con valores.

0
0
0
1
6
32
45
60

Fig. 3.22 Arreglo a[7] ordenado ascendentemente.

En el arreglo ordenado ascendentemente, note que a[0], a[1] y a[2] no fueron presionados. En las demás posiciones son colocados los valores de los botones presionados. Otro punto importante en esta lógica de programación, es que se guarda el mismo orden en que fueron presionados los botones: 60 se presionó primero, 45, 32 y 6 después, y al final se presionó el 1. Al realizar esto se puede extraer las posiciones a[7], a[6], a[5], a[4] y a[3] para obtener el valor de los botones en el orden que fueron presionados. Esta misma lógica, pero aplicada a un arreglo unidimensional de 63 posiciones es aplicada al VI desarrollado. Ver figuras 3.23, 3.24, 3.25, 3.26 y 3.27.

Fig. 3.23 Bloque que ordena un arreglo unidimensional ascendentemente

Fig. 3.24 Bloque que extrae un valor de un arreglo por medio de su índice o posición

Fig. 3.25 Bloque que indica cuando un valor es diferente de cero mandando un 1 lógico

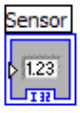

Fig. 3.26 Bloque que despliega un valor decimal de hasta 32 bits de longitud

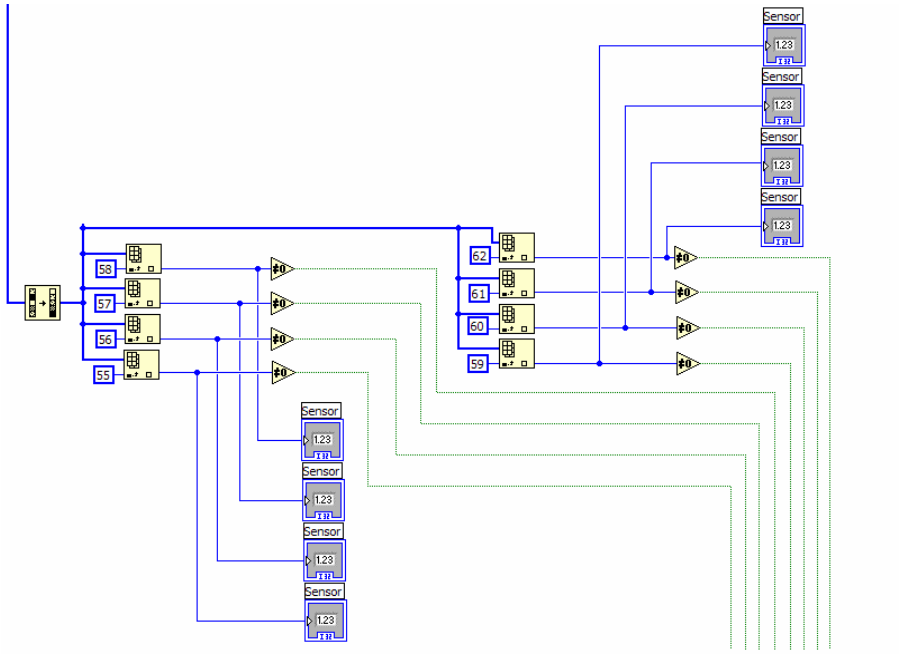

Fig. 3.27 Etapa en la que ordena, extrae valores, los despliega y enciende relevadores.

En la figura 3.27 se ordena el arreglo de forma ascendente, después son extraídos y desplegados los valores de las 8 últimas posiciones, si existe un valor diferente de cero, el comparador manda un 1 lógico, mientras el valor sea cero decimal, el comparador va a estar mandando un 0 lógico. Los cables azules son referentes a valores decimales enteros de 32 *bits* de longitud y los verdes a valores booleanos.

3.3.3 Mensajes al usuario

En esta sección se explica cómo se despliega el mensaje de error mostrado en la parte central de la planta hidráulica virtual (Fig. 3.15).

Fig. 3.28 Bloque selector de cuerda o texto

Fig. 3.29 Bloque que despliega cuerda o texto

Fig. 3.30 Bloque que manda una cuerda vacía

En esta parte, se utiliza un bloque selector de cuerda; en el argot de programación, cuerda (*string*) se le llama tipo de variable alfanumérica, en otras palabras es todo aquello que sea texto. Este selector funciona de la siguiente manera, se coloca una cuerda en la parte superior izquierda, otra en el lugar marcado con "T" (*True*) y otra en el marcado con "F" (*False*); el selector funciona con la variable lógica que se coloca en el lugar marcado con "?", cuando el la variable lógica es verdadera (1), el selector tiene como salida la primer cuerda y le añade la cuerda marcada con "T"; ahora, cuando la variable es falsa (0), el selector va tener de salida la primer cuerda y va a añadir la cuerda marcada con "F". Ver figura 3.28.

Sabiendo el funcionamiento del bloque selector de cuerda, hacemos el análisis de la parte de programación en dónde se manda un mensaje de error. El sistema tiene la capacidad de capturar 8 señales, si intentamos seleccionar una novena, esta no va a ser capturada, pero el usuario tiene que saber que seleccionó una opción inválida.

Tenemos un arreglo de 63 posiciones, de las cuales las únicas útiles son las 8 últimas, después de haber pasado por el bloque que ordena el arreglo de forma ascendente, de la posición 55 a la posición 62; si se llegara a almacenar un valor en la posición 54, ésta sería un valor que el VI desarrollado no va a ser capaz de capturar, por lo que es necesario que el usuario sepa de esto. Se utiliza el bloque que extrae un valor de un arreglo unidimensional, mostrado en la figura 3.24; se extrae el valor de la posición 54, si éste es diferente de 0, es decir que fue presionado un noveno botón, el bloque de comparación, mostrado en la figura 3.25, manda un 1 lógico y activa el selector de cuerda, éste tiene de salida un mensaje que dice: "ERROR. Sólo puedes seleccionar 8"; por el contrario, si no se presiona un noveno botón el selector de cuerda

manda un mensaje que dice: “Puedes seleccionar hasta 8 sensores en orden descendente”. Los cables rosas indican variables de tipo alfanuméricas (cuerdas) y el bloque de la figura 3.30, manda una cuerda vacía, esto con el objetivo de que el selector de cuerdas, no mande algo previo de los mensajes antes mencionados. Ver figura 3.31.

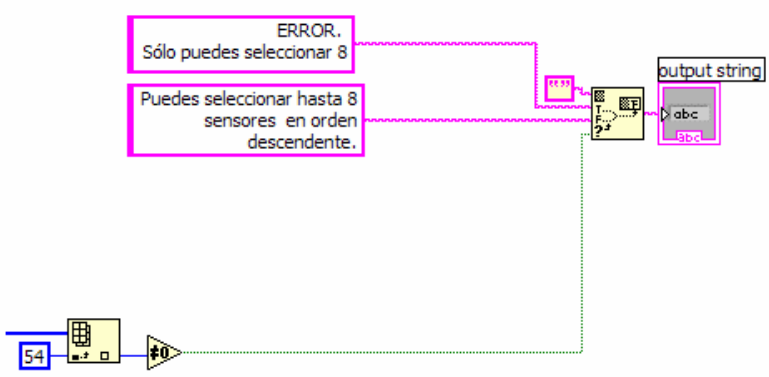

Fig. 3.31 Mensaje de error.

3.3.4 Activación y visualización de indicadores

En la siguiente etapa de programación se hace la activación de los dos tipos de indicadores que utilizamos en esta tesis. Se utilizaron indicadores analógicos y para mejorar la precisión de las mediciones, se utilizaron indicadores digitales, como se muestra en las figuras 3.32 y 3.33

Fig. 3.32 Indicador análogo

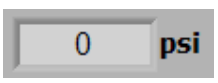

Fig. 3.33 Indicador digital

Fig. 3.34 Bloque para indicador análogo

Fig. 3.35 Bloque para indicador digital

Recordando la etapa de Orden de este VI, los comparadores mandan un 1 lógico cuando el valor alojado en la posición marcada es diferente de cero, y un 0 lógico cuando el valor alojado es cero, ver figura 3.27. La variable lógica viaja a través de los cables verdes y llegan a los relevadores. Ver figura 3.36.

Fig. 3.36 Relevador

El funcionamiento del relevador utilizado es el siguiente: El relevador se encuentra abierto, al recibir una señal lógica verdadera (1) se cierra, es decir se convierte en un corto circuito; por otro lado, al recibir una señal lógica falsa (0), éste se va a quedar abierto, se comporta como un circuito abierto. Ver figura 3.37 y 3.38.

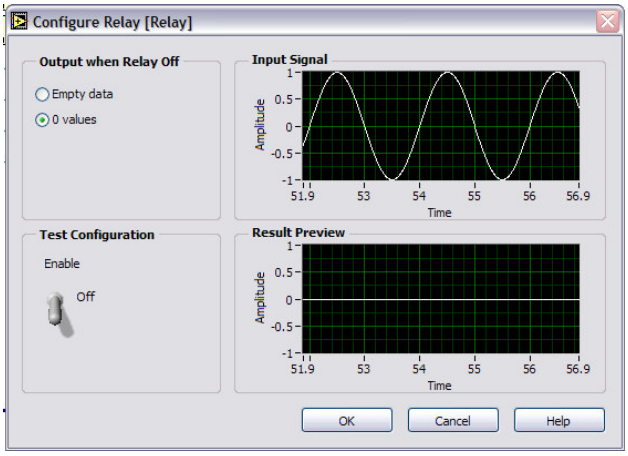

Fig. 3.37 Funcionamiento del relevador apagado

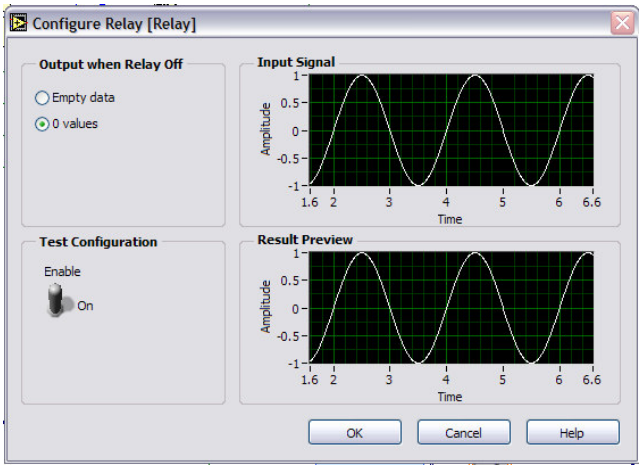

Fig. 3.38 Funcionamiento del relevador encendido

Se colocaron 8 relevadores, uno para cada señal. Son activados en la etapa de Orden y la señal proviene de la etapa de Acondicionamiento virtual de la señal. La única función del relevador es controlar el paso de la señal hacia los indicadores. Ver figura 3.39.

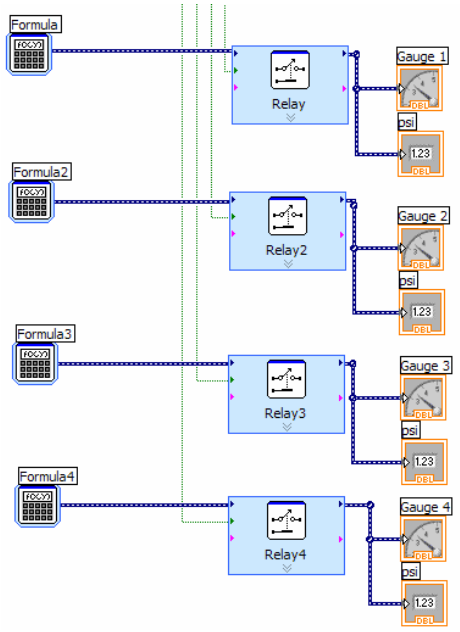

Fig. 3.39 Activación de indicadores

En la figura 3.39 se muestran sólo 4 relevadores, con sus respectivas señales e indicadores; en realidad son 8 sólo que para fines explicativos se decidió mostrar sólo 4.

3.3.5 Escritura de archivo

En esta sección de la programación se va a explicar cómo se hizo para escribir un archivo, es decir, cómo se registró la información. Primero hay que presentar los bloques de *Labview* que vamos a utilizar. Ver figuras 3.40, 3.41, 3.42, 3.43, 3.44 y 3.45.

Fig. 3.40 Bloque para reunir varias señales en un bus

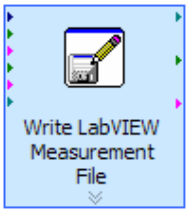

Fig. 3.41 Bloque para escribir un Labview Measurement File

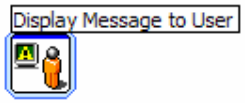

Fig. 3.42 Bloque para desplegar un mensaje al usuario

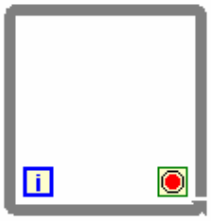

Fig. 3.43 Lazo *While*

Fig. 3.44 Bloque para detener proceso

Fig. 3.45 Botón para detener proceso

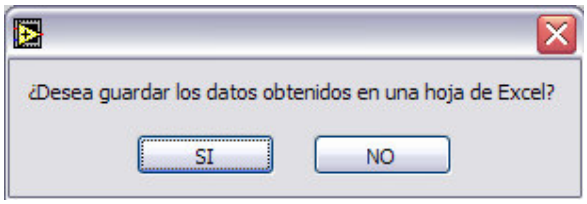

Fig. 3.46 Mensaje al usuario

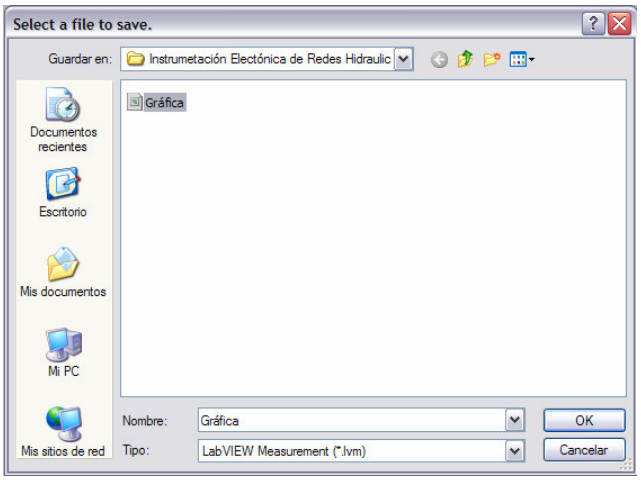

Fig. 3.47 Ventana para guardar archivo

El VI realizado en esta tesis es un sistema de adquisición de datos, donde la información es manipulada por botones; todo sistema de adquisición de datos debe ir forzosamente dentro de un lazo *While*, este funciona de la siguiente manera: se va a repetir todo proceso dentro de él, hasta que sea mandada una señal lógica verdadera a su bloque de paro que se encuentra en la parte inferior derecha. Ver figura 3.43. El instrumento virtual realizado, tiene éste botón de paro y se muestra en la figura 3.45, el del panel frontal y en la figura 3.44 el que se encuentra en el diagrama de bloques.

La etapa de escribir un archivo empieza cuando el usuario del VI presiona el botón de paro rotulado con la leyenda "DETENER SISTEMA", mostrado en la figura 3.45; al presionarlo, éste manda una señal lógica verdadera (1 lógico) al bloque que despliega un mensaje al usuario (figura 3.42), donde se le pregunta si quiere guardar la información en una hoja de *Excel*; ver figura 3.46. Si el usuario presiona el botón rotulado con la leyenda "NO", el VI se detiene y no realiza otra acción, en cambio, si presiona el rotulado con "SI", el VI se detiene, y se manda una señal lógica verdadera al bloque que escribe archivos LMF, (*Labview Measurement*) mostrado en la figura 3.41, éste se activa y captura todas las señales contenidas en el bus de datos, (hecho por bloque mostrado en la figura 3.40) en el instante de tiempo en el que es presionado el botón de paro, posteriormente, el bloque abre una ventana para guardar la información en un archivo con la extensión "*.lvm", cuyo significado es "*Labview Measurement*" o "Medición de *Labview*". El detalle en esta etapa de programación es que el usuario forzosamente tiene que sobrescribir el archivo llamado "Gráfica". Ver figura 3.47 y 3.48.

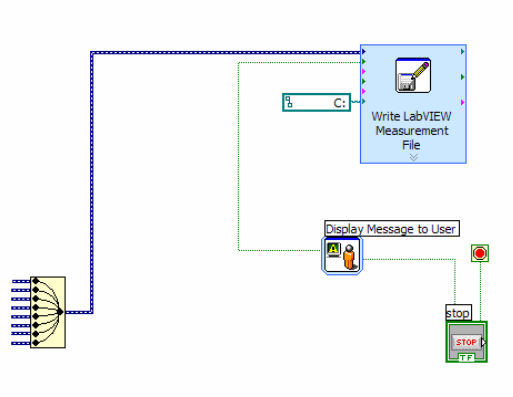

Fig. 3.48 Etapa de escritura de archivo.

El VI fue configurado para que al ser abierto se ejecute automáticamente, además fue protegido con una contraseña por si una persona no autorizada quiere modificarlo, la contraseña programada es “presion” en minúsculas y sin acentos, es importante que sea escrito de esta forma por que si no fuera así, el programa no va a dar acceso al usuario. Ver figura 3.49.

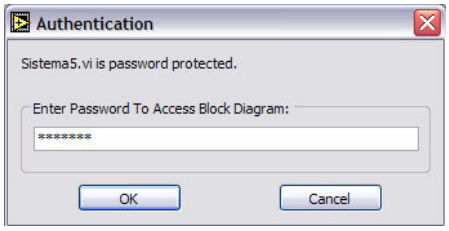

Fig. 3.49 Autenticado de usuario para acceder al modo Edición

En esta etapa concluye el VI, por lo que sólo falta explicar la parte en dónde se manipula la información obtenida en este programa.

3.4 Creación de macro en Microsoft Excel

En la sección anterior realizamos un VI, finalizando con la escritura de un archivo con extensión (*.lvm), por lo que, en esta sección se va a realizar la graficación de los datos obtenidos en la sección anterior.

Al sobrescribir el archivo "Gráfica", *Labview* crea un archivo con nombre "Gráfica.lvm", el cual puede ser leído por *Microsoft Excel*. Éste archivo tiene las siguientes características.

- Archivo que puede ser leído por *Microsoft Excel* y *NI-Labview*.
- Captura 2000 muestras en el instante de tiempo que se presiona " SI " ventana emergente mostrada en la figura 3.46.
- Guarda nombre del operador, éste es fijo y no se puede cambiar.
- Captura el número de canales que se utilizaron en el VI.
- Captura el tiempo en que fue tomada la muestra, este es fijo y no se puede cambiar, utiliza un reloj interno para tomarlo, no es la hora en que fue detenido el VI.

La mayoría de ésta información no nos es útil, por lo que se decidió hacer una *macro* en *Microsoft Excel* con la ayuda de *Visual Basic*. *Macro* es un segmento de programación, en el cual se programa una rutina y cada vez que se ejecute esta macro va a realizar dicha rutina.

La rutina que se ejecutó fue la siguiente:

1. Abrir el archivo “Gráfica.lvm”.
2. Seleccionar un instante de tiempo fijo, en este caso se utilizó la primer captura.
3. Seleccionar una casilla correspondiente a un canal, el cual representa la captura de una señal en un instante de tiempo.
4. Copiar la casilla.
5. Abrir el archivo “Graficación.xls”
6. Pegar la casilla copiada en un lugar determinado, éste es explicado más adelante.
7. Hacer lo mismo para los 8 canales en un tiempo fijo.
8. Dar formato a los datos pegados.

Esta rutina fue guardada con el acceso rápido “Ctrl+a”, lo cual, al presionar esta combinación de teclas, la macro va a ser ejecutada.

El código del programa realizado se puede observar detalladamente en el apéndice A. El rango de casillas en dónde es pegado cada valor obtenido del archivo “Gráfica.lvm” es: [B17:I17], éste rango fue seleccionado para crear una gráfica, la cuál va a graficar cada uno de los 8 canales; la gráfica creada tiene la capacidad de ajustar su escala automáticamente, por lo que nos facilita la visualización de los datos mostrados.

El objetivo de esta gráfica es observar el cambio de presión en cada uno de los 8 canales capturados; cada canal representa la medición en un instante de tiempo de un

sensor de presión diferencial. El resultado final se observa en la tabla 3.5 y en la figura 3.50.

1er. Sensor	*2o. Sensor*	*3er. Sensor*	*4o. Sensor*	*5o. Sensor*	*6o. Sensor*	*7o. Sensor*	*8o. Sensor*
0.587295	0.538658	0.472177	0.395031	0.372827	0.332048	0.311351	0.192651

Tabla 3.5 Tabla realizada por *macro* hecha en *Visual Basic*, tomando muestras en una tubería.

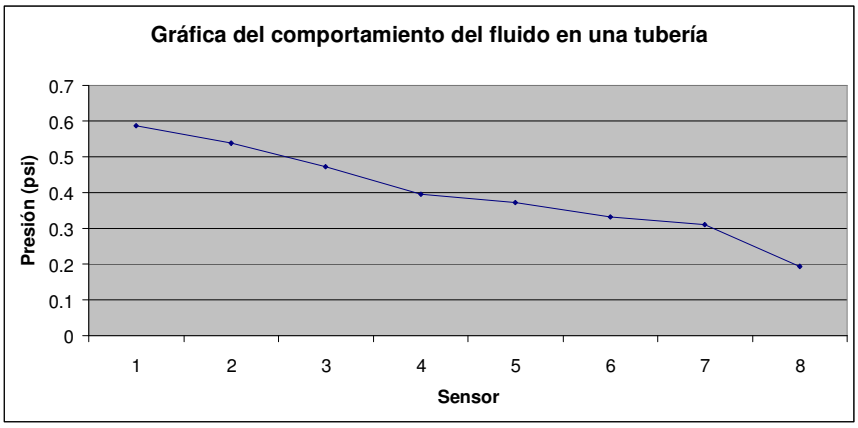

Fig. 3.50 Gráfica de la tabla realizada por *macro* hecha en *Visual Basic*, tomando muestras en 1 tubería.