

4. La interfaz

Como ya se dijo en capítulos anteriores, el prototipo requiere de una interfaz para comunicarse con la computadora.

El tubo acústico recibe el tono que la computadora le envía y a su vez le regresa la señal que genera el micrófono al desplazarse dentro del tubo. La interfaz se encargará de realizar las funciones adicionales (generación de la onda y visualización) para el manejo completo de las señales que se reciben en la computadora.

El programa de la interfaz debe realizar lo siguiente:

- a) Generar un tono puro (señal senoidal de amplitud y frecuencia conocidas) que será enviado al amplificador del tubo y que se percibirá en la bocina. También será capaz de cambiar la frecuencia y la amplitud del tono a voluntad del usuario.
- b) Recibir la señal que envía el micrófono.
- c) Recibir también la señal que sale por la bocina.
- d) Mostrar al usuario la frecuencia de la señal enviada y su magnitud.
- e) Mostrar ambas señales (micrófono y bocina) en el tiempo.
- f) Realizar la comparación de las señales en XY y mostrarlas en una gráfica (Figuras de Lissajous).
- g) Indicar mediante una señal luminosa (led virtual), la igualdad o diferencia de fase entre las señales que procesa.
- h) Mostrar la velocidad del sonido teórica a la temperatura ambiente.
- i) Determinar la velocidad del sonido en función de la distancia recorrida.
- j) Calcular el error entre la velocidad del sonido teórica y la velocidad obtenida en el experimento.

Para realizar todas estas funciones, emplearemos la plataforma llamada *LabVIEW*.

4.1 LabVIEW 8.5

LabVIEW es un software de desarrollo enfocado a la creación de instrumentos virtuales (VI's) para la medición de diferentes procesos. En nuestro caso, desarrollaremos un VI que utiliza la tarjeta de sonido que las propias computadoras tienen, empleando la salida de audio (borne verde) para el envío de la señal que conocemos (tono) y la entrada *line in* (borne azul) para adquirir de regreso la señales que provienen del micrófono y de la bocina.

El modo de programación que utiliza *LabVIEW* es gráfico (cajas y módulos). Tiene controles e indicadores pre-programados que se pueden interconectar entre si para formar paneles de instrumentos con las características que se deseen.

En la plataforma de programación de *LabVIEW* se emplean dos vistas: El diagrama de bloques (*block diagram*) y el panel frontal (*front panel*).

En el diagrama de bloques se realizan todas las conexiones que necesitan los diferentes indicadores y controles que emplea el *LabVIEW*, por supuesto, estas conexiones son de distintas naturalezas pues el software maneja diferentes tipos de datos (matrices, cadenas, constantes, etc), además posee instrumentos “*express*” que simplifican notablemente el tratamiento de las señales. También maneja lazos de control como otros lenguajes de programación (*for, while, case, switch, etc.*).

El Panel Frontal muestra todos los indicadores, controles, perillas y demás accesorios que el instrumento virtual necesite. Este es el panel que el usuario final verá y operará.

En nuestro caso deseamos que la operación de la interfaz sea simple y muy gráfica pues esto ayudará a la comprensión del fenómeno acústico de interés (velocidad del sonido) por parte del usuario final (alumno de bachillerato). Ver la siguiente figura:

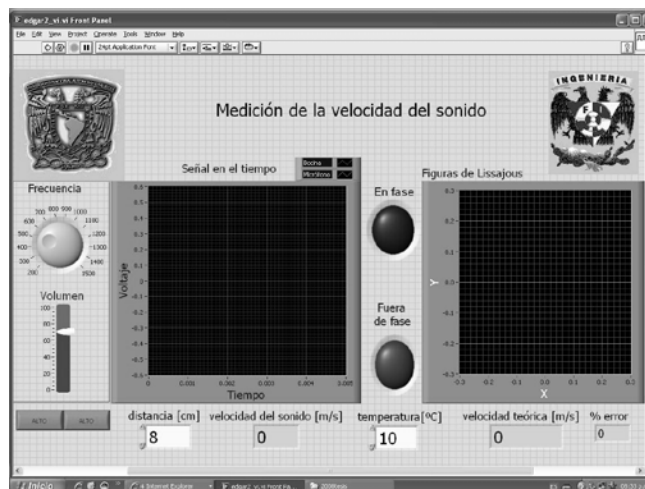


Figura 39. Interfaz del usuario o panel frontal

Y el diagrama de bloques correspondiente al panel es el siguiente:

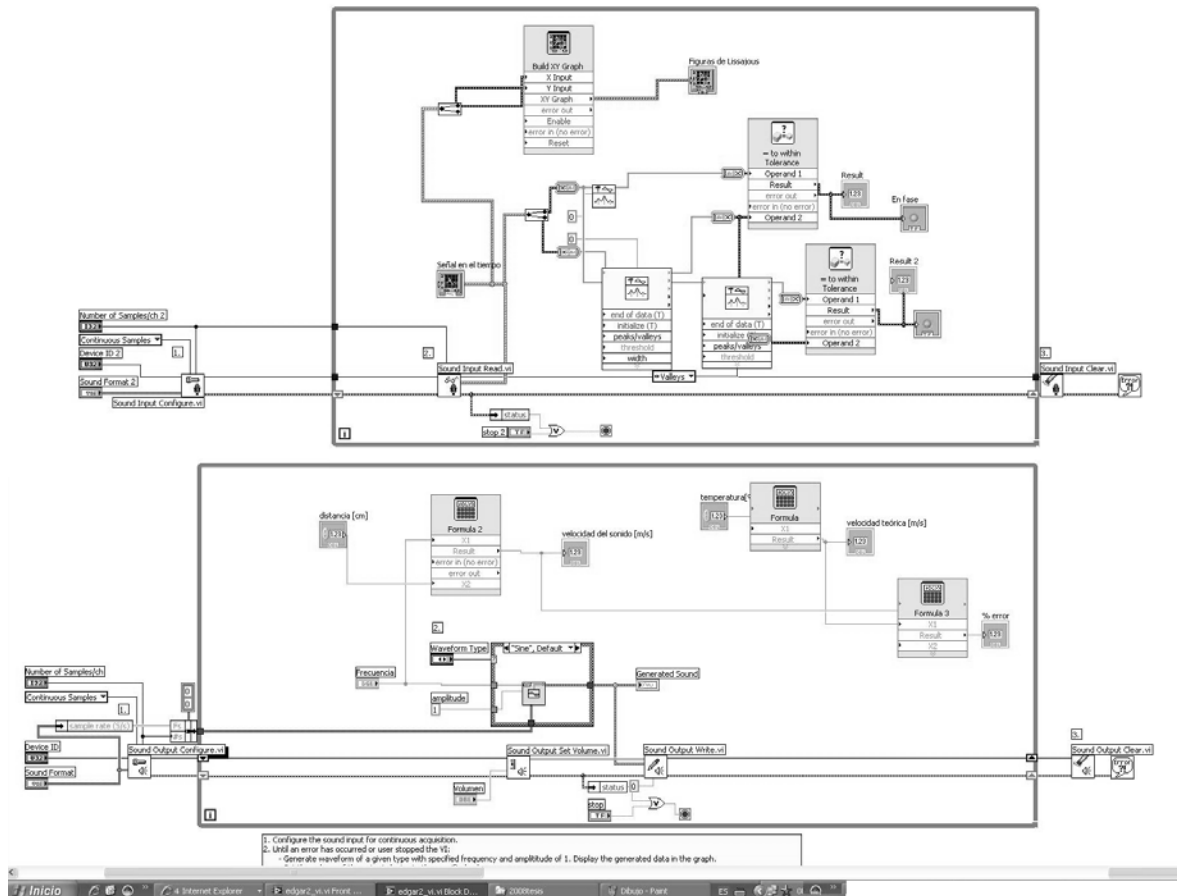


Figura 40. Diagrama de bloques del programa final.

4.2 Programación de la interfaz

La programación debe hacerse de acuerdo al listado siguiente:

- a) Generar un tono puro. Esto se logra mediante el ejemplo “*generate sound*” del propio *LabVIEW*. Gracias a que la plataforma posee ya algunos ejemplos, nos serviremos de ellos para darles las características que necesitamos sin tener que generarlos desde cero. Esto representa una gran ventaja en tiempo y funcionalidad.

El ejemplo nos permite generar una señal senoidal con la amplitud y frecuencia que necesitemos especificando también el número de muestras que deseamos que tenga. Ver el ejemplo en la siguiente figura:

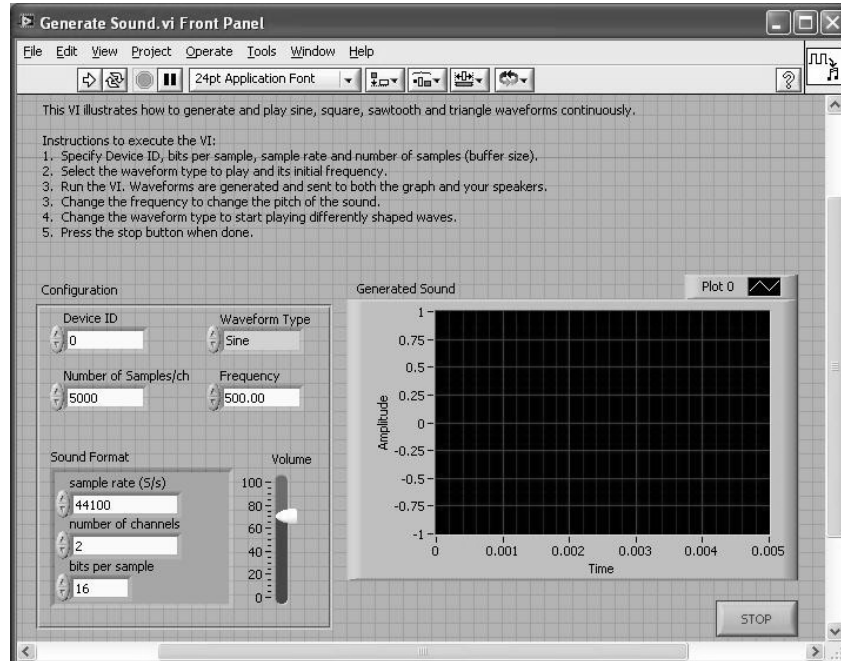


Figura 41. Ejemplo de cómo generar un sonido (panel frontal)

Cada panel está asociado a un diagrama de bloques, esto nos permite la programación de manera gráfica y por bloques. Cada una de las características del ejemplo puede ser modificada por nosotros de acuerdo a lo que necesitemos, también es posible “desactivar” algunos controles del panel, de modo que no aparezcan en la interfaz final y no puedan ser operados por el usuario final. La programación en este caso solo es posible a través del diagrama de bloques.

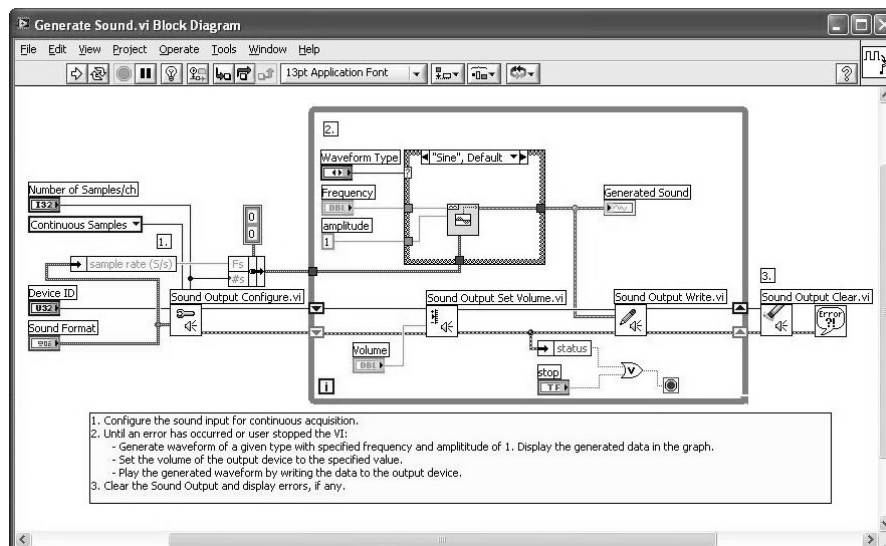


Figura 42. Diagrama de bloques de “generando un sonido”

- b) Recibir la señal que proviene del micrófono
- c) Recibir la señal que sale por la bocina.

Ambas señales se reciben por la tarjeta de sonido de la computadora. Una vez más, empleamos el ejemplo o aplicación previa que tiene la plataforma de *LabVIEW*.

Continuous Sound Input (entrada de sonido continuo) nos permite adquirir una señal que entra a la computadora a través de la tarjeta de sonido para su manipulación. Los parámetros que la aplicación nos muestra se observan en la figura 43:

- Forma de la señal a través de una grafica en tiempo.
- Tipo de tarjeta. (solo poseemos una)
- Numero de muestras por canal
- Tasa de muestreo
- Numero de puntos por muestra
- Numero de canales

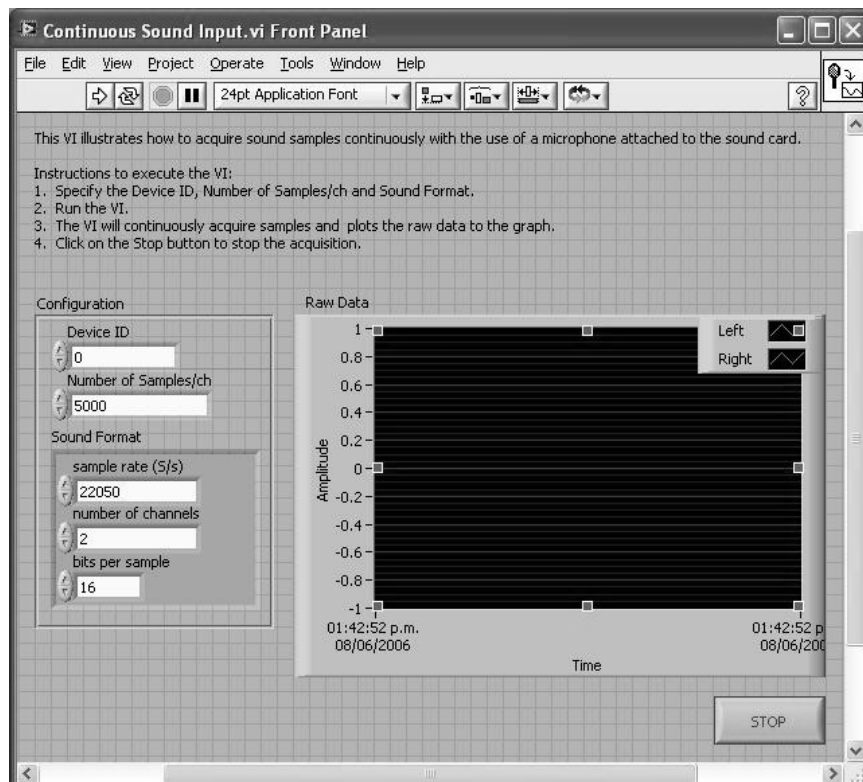


Figura 43. Panel frontal de Entrada de sonido continuo

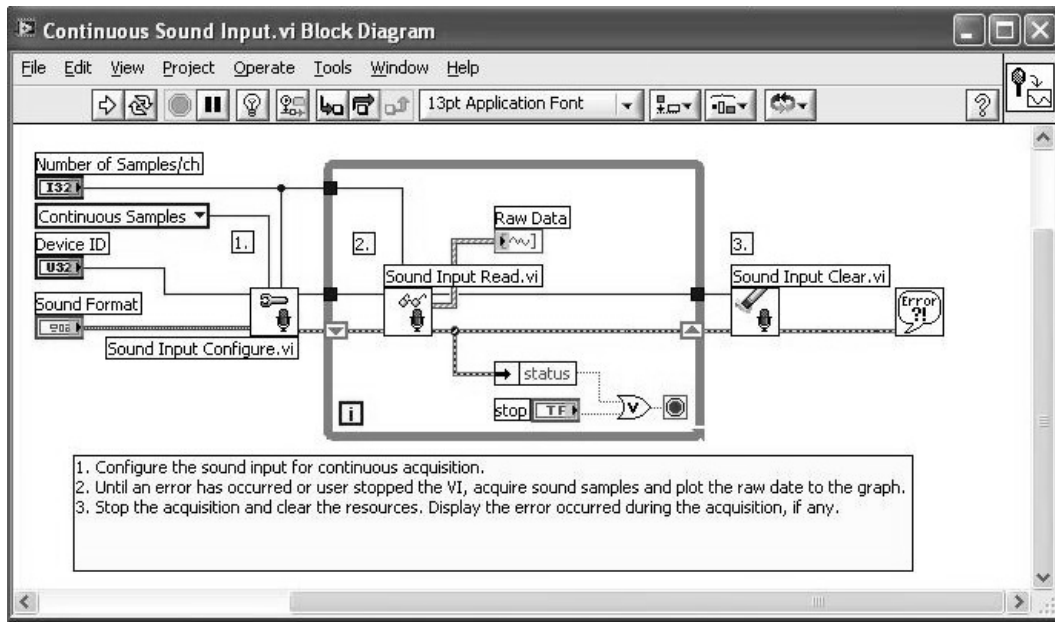


Figura 44. Diagrama de bloques de entrada de sonido continuo.

En la figura 44 se muestra el diagrama de bloques que corresponde a la programación del ejemplo *Continuos Sound Input*.

- d) Mostrar al usuario la frecuencia de la señal enviada y su magnitud. Esto se logra a través del dial de frecuencia y la barra de volumen, no solo se pueden ver sino que además el usuario puede modificar los valores que desee (para el volumen de 0 a 1 [V], y la frecuencia de 200 a 1500 [Hz]), ver la siguiente figura:

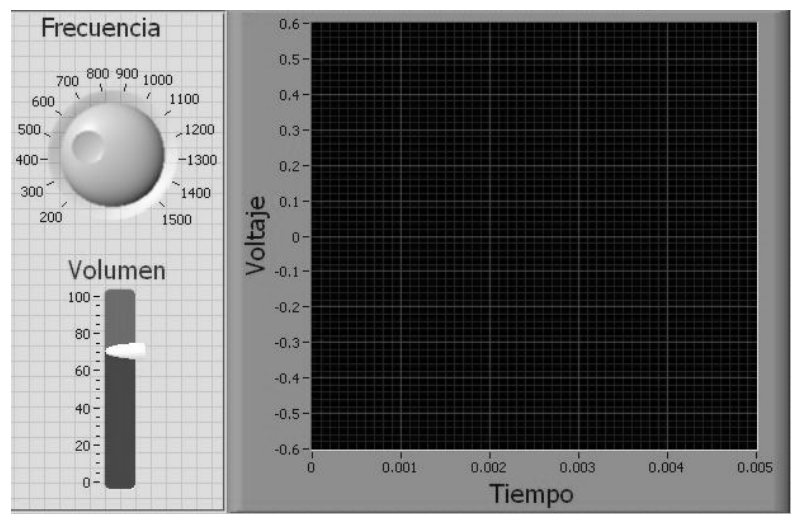


Figura 45. Controles de frecuencia, volumen y gráfica de las señales en tiempo

- e) Mostrar las señales del micrófono y la bocina en el tiempo.
Se logra con la grafica mostrada en la figura 45.
- f) Realizar la comparación de las señales (Figuras de Lissajous). El ejemplo nos muestra las señales que se reciben en la tarjeta de sonido en el tiempo, pero si queremos ver las figuras de Lissajous será necesario hacer la transformación de variables y hacer un cambio en el tipo de grafica.

En la figura 46 se muestran los bloques necesarios para la conversión.

Utilizamos un VI (Instrumento virtual) Express, el *Build XY graph*, que ya viene pre-configurado y que nos facilita la obtención de las Figuras. En el cambio de variable solo usamos un “*split*” que nos divide la señal que viene como estéreo en dos canales distintos. Una vez hecha esta separación, el *Build XY* envía la comparación a una grafica en X y Y.

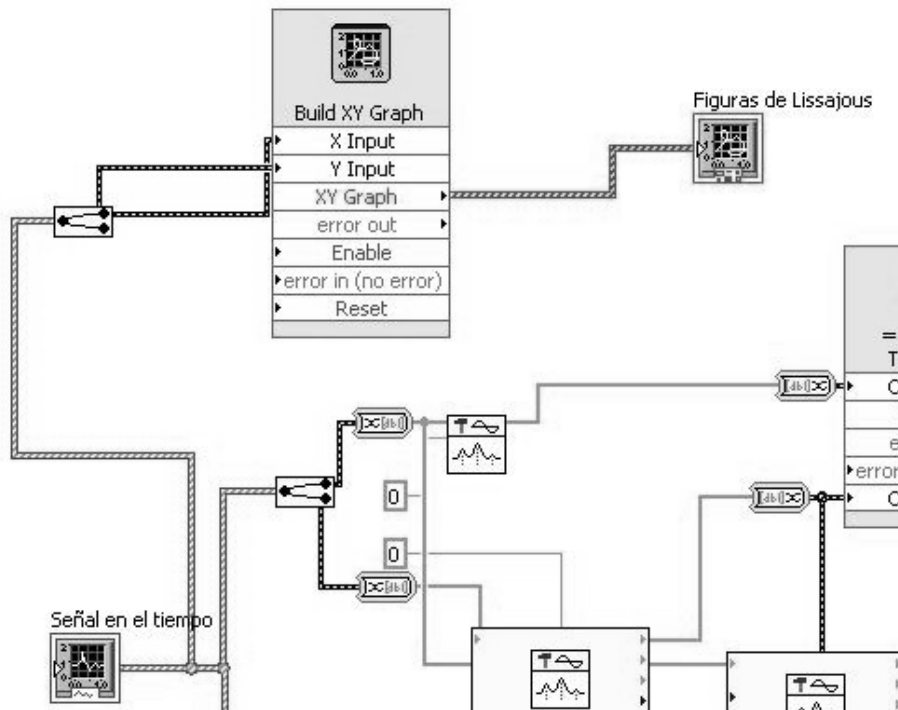


Figura 46. Bloques para realizar la gráfica de las Figuras de Lissajous

- g) Indicar mediante una señal luminosa (*led* virtual), el desfase de 0 ó 180 grados entre las señales que procesa. En esta parte recurrimos a diferentes bloques de control e indicadores para lograr que los *leds* enciendan según sea el caso (fase o desfase) ver figura 47.

Solo se consideran los casos en que el desfase es de 0 grados, esto es, que no existe diferencia de fase entre las señales y cuando el desfase es de 180 grados, que implica que las señales son completamente opuestas (el valle de una corresponde al pico de la otra). Realmente ocurren diferentes desfases durante el envío, la amplificación y la recepción de las señales, pero todos ellos son constantes, de modo que el único cambio que nos interesa es el desfase que ocurre cuando el micrófono se desplaza en el interior del tubo. En el momento que la gráfica nos muestra “en fase” existe un desfase de 0 grados para ambas desde el punto de vista de la gráfica. Cuando desplazamos el micrófono, todos los desfases permanecen constantes menos el de la posición del micrófono, de manera que el “desfase” en la gráfica es únicamente función del movimiento del micrófono.

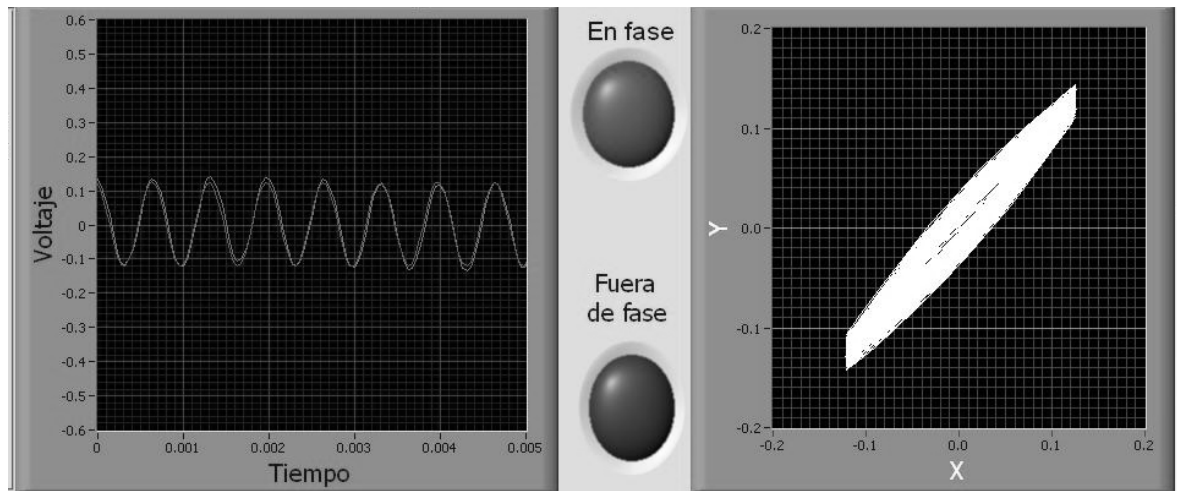


Figura 47. Gráfica de las señales en tiempo y figura de Lissajous con leds indicadores.

Para lograr lo anterior necesitamos “comparar” las señales. Antes de esto se requiere una detección de picos y valles, para poder comparar únicamente lo que nos interesa. Esto se logra con el VI. *Peak Detector*, verlo en la figura 48.

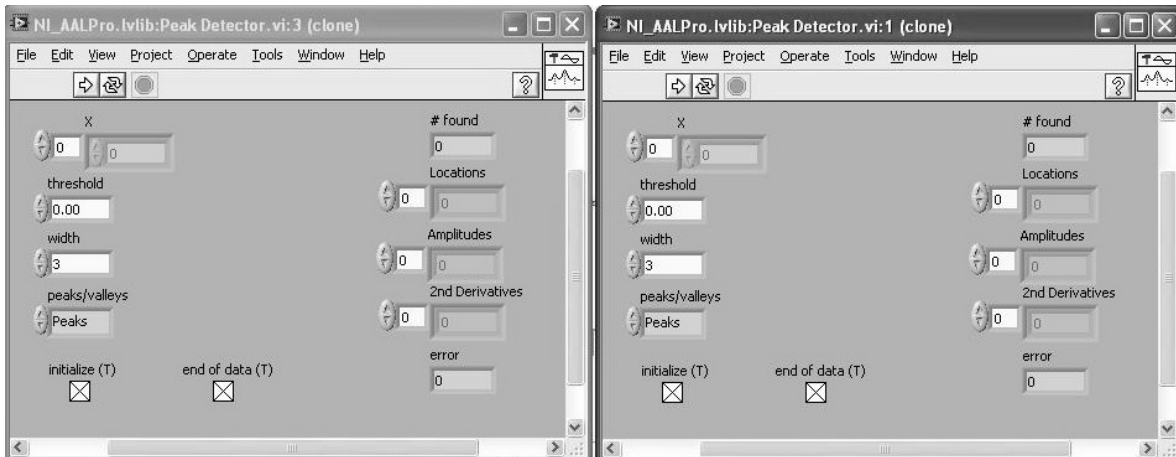


Figura 48. Propiedades del detector de picos

El detector de picos es una herramienta que nos permite encontrar los picos y valles de una señal. Mediante el empleo de ello encontraremos primero los picos de ambas señales y luego los valles de la señal del micrófono. La siguiente figura nos muestra los bloques correspondientes.

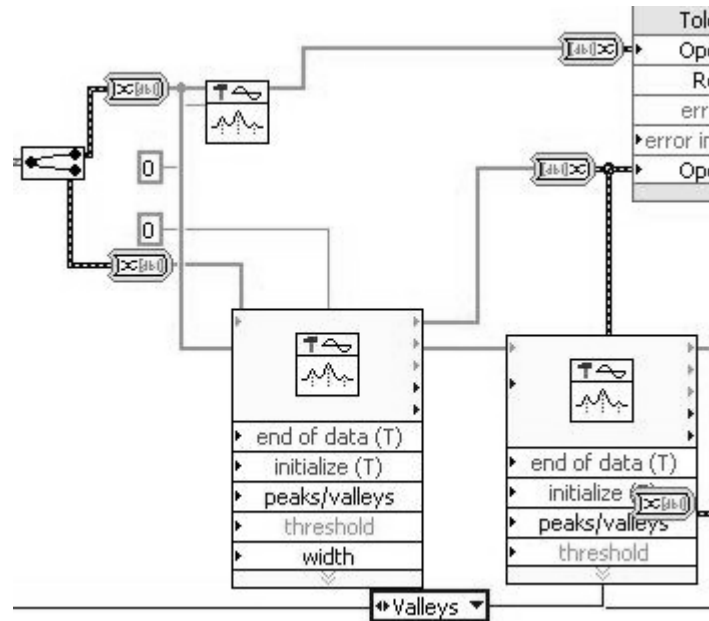


Figura 49. Diagrama de bloques del detector de picos

En el proceso de detección ocurre un cambio de datos, se pasa de un dato en forma de señal a un dato numérico y luego se regresa el dato a forma de señal para hacer la comparación, esto lo hace automáticamente el convertidor de tipo de datos.

Una vez obtenidos los picos y los valles de las señales establecemos la comparación mediante el uso del del VI express “*comparison*” ver figura 50.

El VI. *Comparison* nos permite comparar señales una a una de diferentes formas. La que nos interesa a nosotros es la de “igualdad con tolerancia” para que cuando exista la igualdad debida al movimiento de la señal del micrófono, se encienda un *led* si se trata de un desfase de cero grados o, se encienda otro si se trata de un desfase de 180 grados.

Los campos del comparador se muestran a continuación.

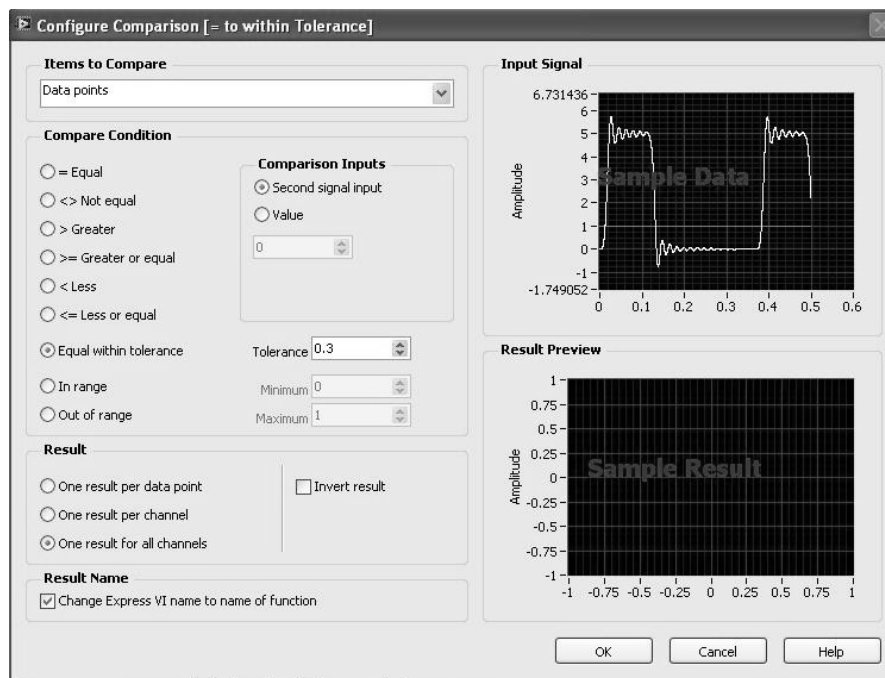


Figura 50. Propiedades del comparador

Debido a que la longitud de onda cambia en función de la frecuencia, el margen de tolerancia para el comparador también debería cambiar de acuerdo a la frecuencia, sin embargo, encontramos un valor para el cual los *leds* encienden aún cuando la apreciación de las señales en el tiempo y en XY se muestra confusa.

La salida de estos comparadores va directamente a los *leds* para determinar la diferencia de fase que existe entre las señales.

Los bloques correspondientes a las comparaciones se muestran en la siguiente figura:

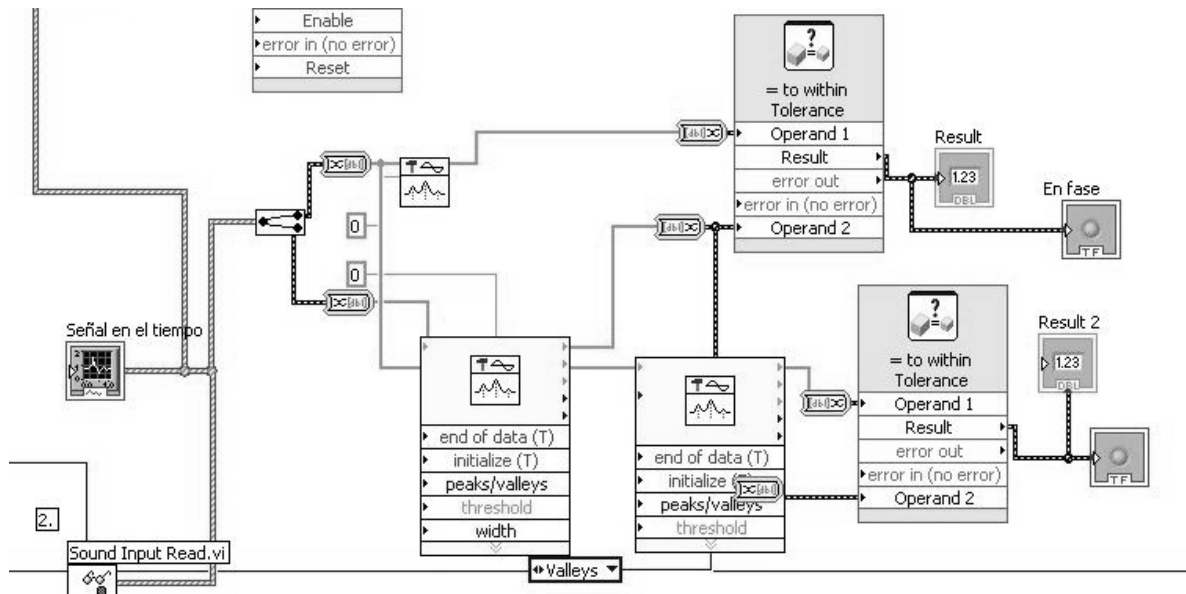


Figura 51. Bloques comparadores

Cuando se ha realizado la comparación y si el resultado es verdadero, se encienden los *leds* respectivos en la interfaz, como se muestra a continuación:

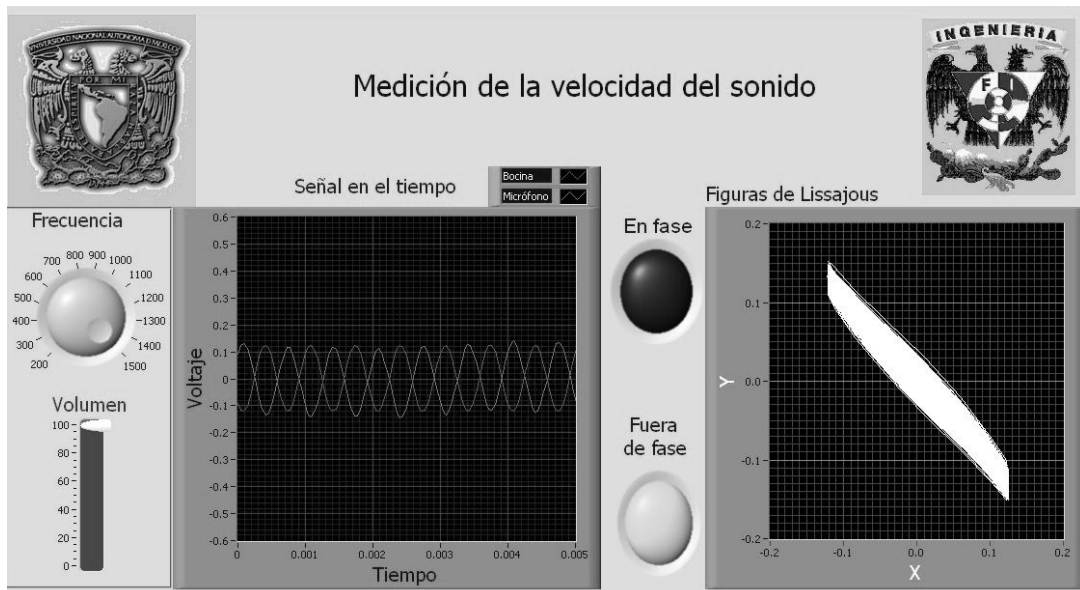


Figura 52. Detección de la fase en la interfaz

h) Mostrar la velocidad del sonido teórica dada la temperatura ambiente.
Como el prototipo cuenta con un termómetro digital, podemos calcular la velocidad teórica de acuerdo a la temperatura ambiente si ingresamos su valor a la interfaz.

El *LabVIEW* nos permite programar cajas de ingreso de datos y fórmulas de manera que el usuario final pueda obtener datos como resultado de alguna operación con sólo ingresar un valor o valores correspondientes.

De acuerdo a la ecuación 21, programamos la caja de fórmulas para obtener la velocidad en función de la temperatura, como se puede ver en las figuras 53 y 54.

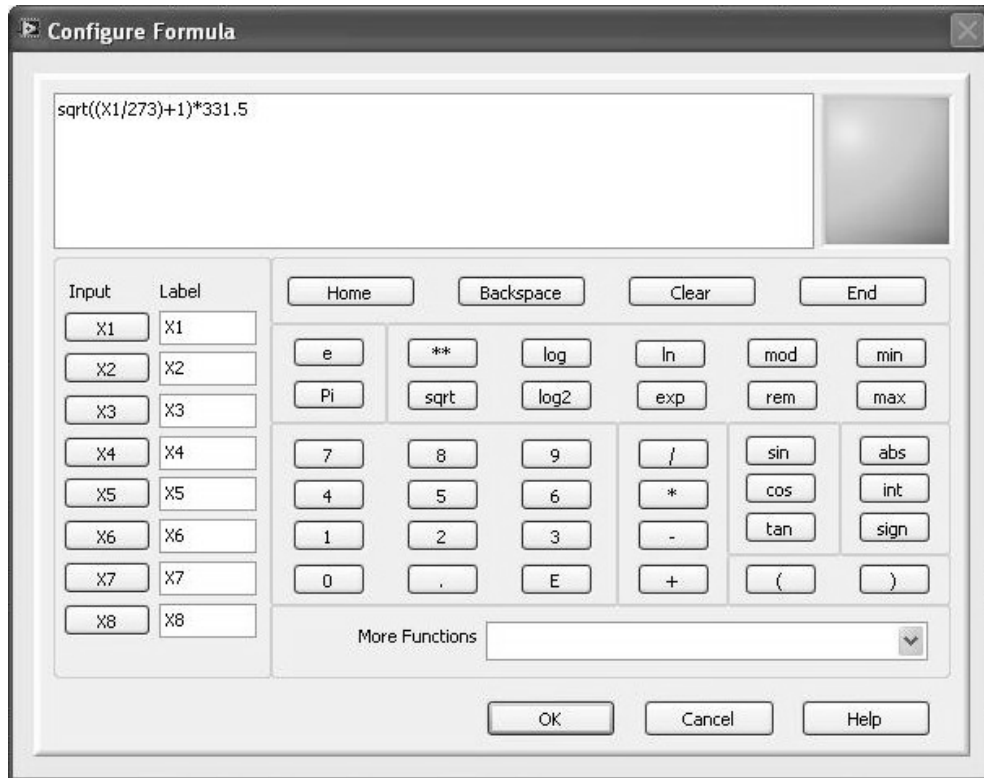


Figura 53. Fórmula para calcular la velocidad del sonido en función de la temperatura

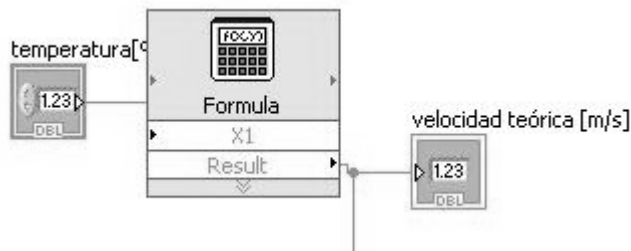


Figura 54. Diagrama de bloques del cálculo de la velocidad teórica

i) Determinar la velocidad del sonido en función de la distancia recorrida.

Nuevamente, para calcular la velocidad del sonido con la distancia registrada en el experimento, nos servimos de la caja de ingresos de datos y el V.I. Express de “*formula*”.

Como la distancia que recorre el micrófono desde que encuentra un pico (desfase cero) hasta que encuentra un valle (desfase de 180 grados) corresponde a la media longitud de onda, basados en esto podemos calcular la velocidad en función de la frecuencia de la onda (que ya conocemos) y la distancia (longitud de onda) que ingresemos a la caja de datos en la interfaz.

De la ecuación 1; λ es el doble de la distancia “d” recorrida por el micrófono dentro del tubo.

Programamos la fórmula para el cálculo de la velocidad del sonido experimental como se ve a continuación:

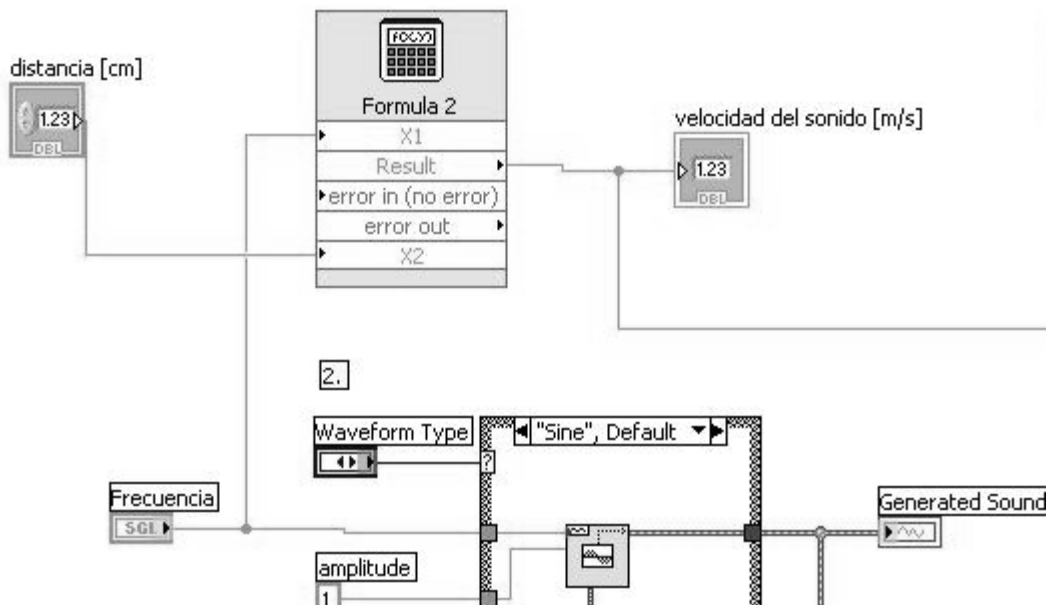


Figura 55. Diagrama de bloques del cálculo de la velocidad experimental

De manera adicional, podemos ingresar el valor de la distancia recorrida por el micrófono en centímetros [cm] y con un decimal [milímetros], el bloque de fórmula nos permite convertir a metros para homogeneizar las operaciones y obtener nuestro resultado en metros sobre segundo [m/s] (observar la figura).

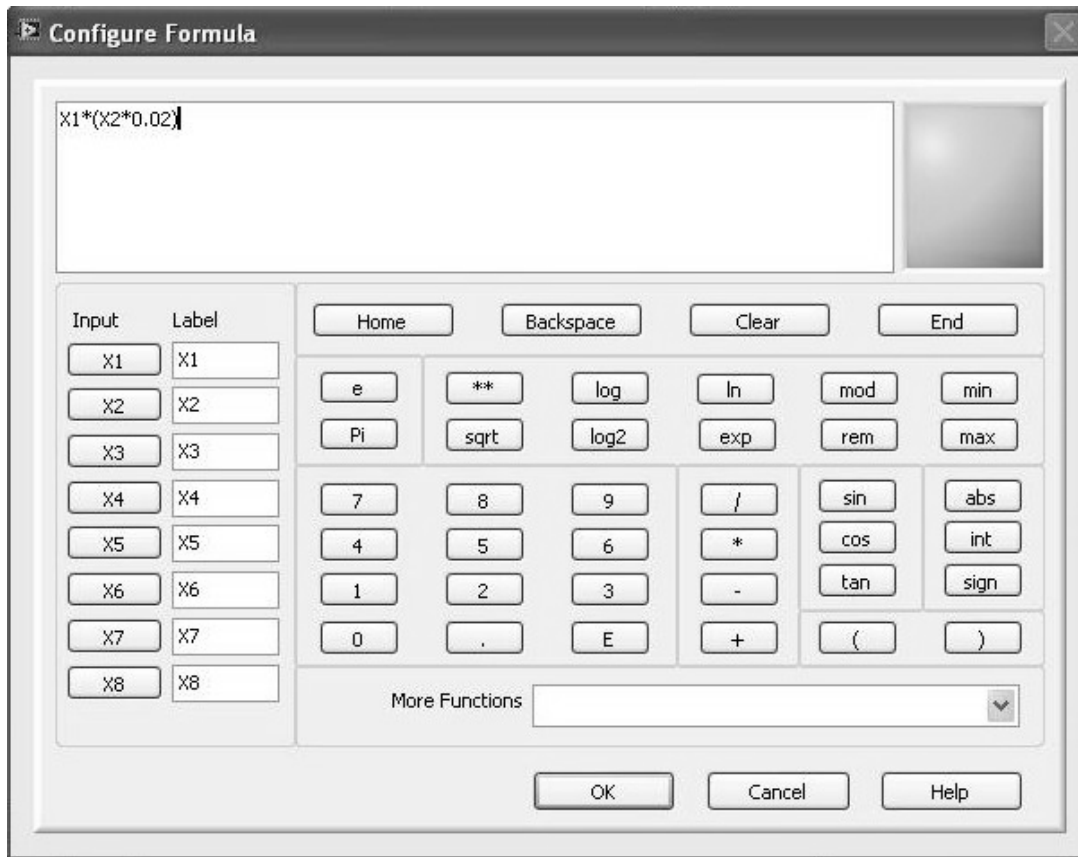


Figura 56. Fórmula para el cálculo experimental de la velocidad del sonido.

- j) Calcular el error entre la velocidad del sonido teórica y la velocidad obtenida en el experimento.

Una vez registras la velocidad teórica y la velocidad experimental podemos calcular el error que existe entre ambas. Este es un buen parámetro para determinar la funcionalidad de nuestro prototipo. Debemos decir que debido a la resolución en la toma de las distancias (1[mm]) para longitudes de onda demasiado pequeñas (menores de 15 [cm]) el error crece con cada milímetro de diferencia.

Programamos la fórmula de error de acuerdo a la ecuación 39. Se pueden apreciar las operaciones en la figura 57.

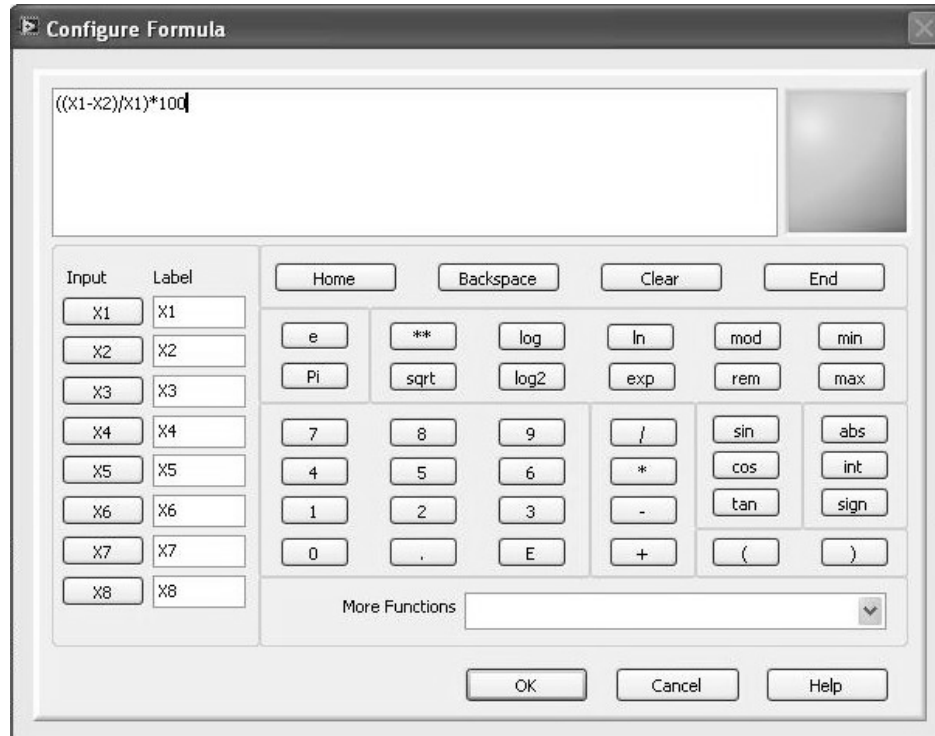


Figura 57. Programación del cálculo del error en *LabVIEW*

Una vez que la interfaz ha cumplido con todas la demandas que el prototipo requiere, se encuentra lista para someterse a las pruebas de conjunto y a las mediciones que los estudiantes de bachillerato puedan hacer. Ver figura 58

