



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

“Diseño y creación de interfaces gráficas en
LabVIEW, para la caracterización de sensores y el
monitoreo experimental del SIBEO”

TESIS

que para obtener el título de

INGENIERO MECATRÓNICO

PRESENTA:

DAVID MARÍN GARCIA



Director de Tesis:

M.I. Billy Arturo Flores Medero Navarro

Ciudad Universitaria, México, octubre 2011

Gracias...

A Dios, por darme la oportunidad de vivir, salir adelante y superarme. Por darme la fuerza y tenacidad necesaria en aquellos momentos en que el camino se tornaba difícil...

Mamá, por apoyarme toda la vida, por enseñarme a ser autosuficiente en todos los ámbitos posibles, gracias a ti lleve a cabo mis estudios de una manera responsable y autodidacta...

Abuelos, por enseñarme con su experiencia y sabios consejos, por inculcarme que la dignidad de una persona está en el esfuerzo de su trabajo y en los logros que de ahí emanen...

Tíos, por ser ejemplos de superación y éxito, porque en ustedes encontré un orgullo, la experiencia y la motivación para abrirme paso en el mundo...

Familia, porque cada uno de ustedes en algún momento, en algún espacio, en alguna circunstancia supo hacer o decir justamente lo que necesitaba para avanzar y no decaer...

A la Universidad Nacional Autónoma de México, por abrirme sus puertas y brindarme un mundo de conocimientos, así como las habilidades necesarias para desarrollarme como profesional...

A mis Profesores de Carrera, por compartir conmigo sus amplios conocimientos, gracias a cada uno de ellos por su gran vocación de enseñanza...

A mis Amigos, por caminar a mi lado en esta etapa, por ser compañeros fieles y por apoyarme en aquellos momentos en los que el trabajo de muchos era necesario...

Al Doc. Steven Czitrom Baus y al Proyecto SIBEO, por darme la oportunidad de aplicar mis conocimientos y habilidades, ha sido sin duda una experiencia muy enriquecedora y el inicio de una vida profesional...

A todos y cada uno de ustedes les dedico este trabajo.

David Marín García

Índice

Objetivo	(2)
Introducción	(2)
Capítulo 1. Antecedentes	
1.1 Sistema de bombeo por energía de oleaje (SIBEO).....	(5)
1.1.1 Principio de funcionamiento.....	(5)
1.1.2 Importancia del monitoreo experimental.....	(6)
1.2 Problemática.....	(6)
1.3 Breve estado del arte.....	(7)
1.4 Solución.....	(13)
1.4.1 LabVIEW.....	(14)
1.4.2 Caracterización de sensores.....	(15)
1.5 Programación en base al modelo <i>cascada</i>	(19)
Capítulo 2. Diseño conceptual	
2.1 Metodología de desarrollo.....	(21)
2.2 Requerimientos para el desarrollo.....	(22)
2.3 Instrumentación a emplear.....	(23)
2.4 Diseño gráfico de interfaces.....	(24)
2.5 Algoritmos de programación.....	(27)
2.6 Diagramas de flujo.....	(31)
Capítulo 3. Desarrollo de interfaces	
3.1 Funcionamiento de comandos básicos.....	(34)
3.1.1 Adquisición de datos externos.....	(34)
3.2 Programación de la interfaz “Modo Calibración”.....	(35)
3.2.1 Ajuste de requerimientos y reprogramación.....	(41)
3.3 Programación de la interfaz “Modo Experimental”.....	(43)
3.3.1 Ajuste de requerimientos y reprogramación.....	(48)
3.4 Pruebas de laboratorio.....	(49)
Capítulo 4. Manual de operación	
4.1 Manual de operación de la interfaz “Modo Calibración”.....	(55)
4.2 Manual de operación de la interfaz “Modo Experimental”.....	(58)
Conclusiones	(61)
Anexos	
Anexo 1. Tarjeta de adquisición de datos NI USB-6009.....	(62)
Bibliografía	(65)

Objetivo

El presente trabajo tiene como objetivo describir el desarrollo de interfaces gráficas, creadas en LabVIEW, para la caracterización de sensores con salida en voltaje y para el monitoreo de las variables críticas presentes en los experimentos realizados con el Sistema de Bombeo por Energía de Oleaje (SIBEO) propiedad de la UNAM. Ya que, el SIBEO es un invento de recién concepción, requiere de un análisis experimental detallado para el mejoramiento de su desempeño en distintos ambientes de trabajo y potenciar así su campo de aplicación, a fin de beneficiar al hombre y al medio ambiente.

Introducción

El proyecto SIBEO cuenta con paqueterías de *software* tradicionales y procesos de análisis poco eficientes que proveen tiempos de respuesta considerables. Este trabajo propone implementar y desarrollar nuevas herramientas tecnológicas capaces de procesar una mayor cantidad de información experimental en un menor tiempo.

En la actualidad existen en el mercado, en forma comercial o libre, distintos paquetes de *software* con los que es posible realizar una caracterizar sensores a través de métodos numéricos como la regresión lineal y polinomial, sin embargo, estos son únicamente sistemas informáticos en donde no es posible ingresar datos desde un dispositivo externo, los datos deben ser ingresados manualmente o importados desde un archivo externo. En lo que a monitoreo de señales refiere, son muy pocas las herramientas que se pueden encontrar de manera comercial y adaptables a las necesidades de cada aplicación; los sistemas disponibles son creados para una sola área de actuación.

LabVIEW es un lenguaje de programación gráfico creado por la empresa National Instruments, que utiliza iconos para proveer un ambiente de programación más sencillo que los entornos tradicionales. En contraste con los lenguajes de programación basados en línea de código, este *software* usa la programación en “Flujo de Datos”, donde el flujo de los datos a través de los nodos en un diagrama de bloque determina el orden de ejecución de las funciones. En LabVIEW es posible construir instrumentos virtuales (VIs), utilizando una variedad de herramientas y objetos, en una interfaz de usuario conocida como “Panel Frontal” (PF), después se agrega el código de programación en un “Diagrama de Bloque” (DB) usando representaciones gráficas de las funciones para controlar los objetos del PF. Este código de recurso gráfico es también conocido como código G o código de diagrama de bloque.

El trabajo de creación de nuevas interfaces es un proyecto complejo, por lo que es necesario recurrir a principios de desarrollo específicos. Estos principios definen el campo de la ingeniería de *software*. El mayor componente de este campo son los modelos de ciclo de vida, los cuales describen los pasos a seguir cuando se desarrolla un *software*. El “Modelo Cascada” es clásico en la ingeniería de *software* y es el primero de los modelos más antiguos. Es altamente usado en proyectos gubernamentales y en muchas grandes compañías, debido a que enfatiza la planeación en las primeras etapas, pudiendo identificar las fallas de diseño antes de ser desarrolladas. Es por esto que la metodología de desarrollo se basara en este modelo, considerando los siguientes aspectos.

1. Obtención de los requerimientos funcionales y operacionales.
2. Descripción de los sensores y equipos a emplear.
3. Elaboración de prototipos rápidos para cada interfaz.

4. Definir los algoritmos de programación.
5. Elaboración de los diagramas de flujo de programación.
6. Desarrollar el código de programación.
7. Detección e implementación de mejoras.
8. Pruebas de funcionamiento en laboratorio.
9. Elaboración del manual de operación.

Una herramienta de *software* robusta y eficaz implica un procedimiento de mejora continua para corregir errores no previstos y optimizar su funcionamiento. En el caso de las interfaces a desarrollar se analizarán también algunas áreas de oportunidad para mejorar su funcionamiento, en base a la experiencia y como trabajo a futuro.

Capítulo 1. Antecedentes

1.1 Sistema de bombeo por energía de oleaje (SIBEO)

En medio de la inquietud por el posible agotamiento de los yacimientos petroleros y la contaminación ambiental causada por la quema de combustibles fósiles, se han dado pasos para explotar la energía solar, eólica y del oleaje marino. En el Instituto de Ciencias del Mar y Limnología de la UNAM, un grupo de investigadores dirigidos por el Dr. Steven Czitrom Baus han diseñado tecnología de punta para aprovechar esta última. Se trata de un sistema que permite bombear agua limpia y oxigenada a cuerpos de agua contaminados.

En todo el mundo existen cuerpos de agua costeros (como lagunas) cerrados de manera natural que al recibir y acumular contaminantes, pueden llegar a presentar niveles tóxicos peligrosos. Una de las alternativas para sanearlos es bombear a su interior agua limpia y oxigenada del exterior, sobre todo cuando dichos cuerpos de agua contienen materia orgánica acumulada.

Otra aplicación del SIBEO es el manejo biológico en cuerpos de agua costeros. En muchas lagunas de México y del mundo, en donde se desarrollan larvas de camarón y otras especies de la cadena alimenticia, se encuentran aisladas del mar por barreras de arena y por la disminución de lluvias, a consecuencia de la deforestación. Con el uso del SIBEO se puede generar un intercambio controlado de agua de esas lagunas costeras y el mar para incrementar la piscicultura y, por lo tanto, el nivel económico de los pescadores de la región. Debido a que el sistema no tiene partes móviles, los organismos marinos pasan a través de él sin sufrir ningún daño. De esa manera se pueden inyectar larvas de especies marinas para fertilizar los cuerpos de agua costeros, así como manejar los niveles de nutrientes, de oxígeno, de salinidad, y mejorar las condiciones ambientales para la producción.

1.1.1 Principio de funcionamiento

El SIBEO es una herramienta de manejo costero que puede usarse, además del saneamiento de áreas marinas contaminadas, para la acuicultura o la recuperación de lagunas costeras aisladas para la crianza de especies.

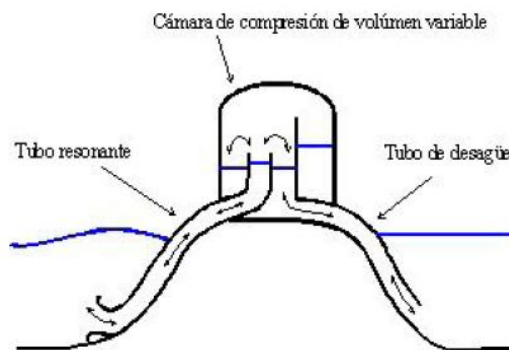


Figura 1. Diagrama esquemático del SIBEO.

La figura anterior muestra un diagrama esquemático del SIBEO. La señal de presión inducida por el oleaje en la boca del tubo resonante provoca un flujo oscilatorio, que derrama agua en la cámara de compresión con el paso de cada ola. El agua derramada se acumula en la cámara y luego desciende por gravedad al cuerpo de agua receptor. El aire en la cámara de compresión se comporta como un

resorte contra el cual oscila el agua en ambos tubos, de tal manera que el sistema tiene una frecuencia natural de oscilación que depende de la dureza del resorte y de las masas de agua en los tubos. Cuando esta frecuencia coincide con la del oleaje incidente, el sistema entra en resonancia, de tal forma que se maximizan las oscilaciones en los tubos y por lo tanto la eficiencia de bombeo. El sistema puede mantenerse en resonancia, para cualquier frecuencia de oleaje, mediante un sistema de sintonización (Patente del Dr. Czitrom) que varía la dureza del resorte de la cámara de compresión cambiando su volumen (por cuestiones de confidencialidad no será analizado o mostrado dicho sistema de sintonización).

1.1.2 Importancia del monitoreo experimental

Ya que el SIBEO es una herramienta concebida y proyectada para ser aplicada, actualmente se incursiona en generar energía eléctrica. Con este objetivo en mente es necesario reanalizar en el laboratorio los modelos a escalas, implementar las modificaciones pertinentes y validar su respuesta mediante un sistema de monitoreo experimental. Será un proceso continuo y sistemático, en el cual se verificará la eficiencia y eficacia de los cambios, identificando debilidades y logros.

Las variables más importantes a analizar son:

- Nivel de altura en el tubo resonante.
- Nivel de altura en el tubo de desagüe.
- Nivel de altura del oleaje.
- Nivel de altura del flujo.
- Presión en la cámara de compresión.
- Presión en la boca de entrada.

1.2 Problemática

Desde la creación de los primeros prototipos del SIBEO, se han realizado pruebas de laboratorio para conocer su comportamiento ante modificaciones de diseño y sintonización, con diversos sistemas de *software* y hardware, implementados de manera sistemática. Sin embargo, cada prueba y proceso muchas veces resulta lento y por tanto poco eficiente.

En el caso de los equipos de medición, ya sea fabricados por los mismos colaboradores del proyecto o adquiridos en el mercado, el proceso de su caracterización emplea en primera instancia un *software* para la adquisición de voltajes, empleando sistemas electrónicos para adquisición de datos (*Data Acquisition Systems*, DAQS). Posteriormente los datos son ingresados a otro *software* que gráfica dichas señales contra niveles patrón, mostrando en pantalla la respuesta del sensor. Posteriormente esta gráfica es analizada en otro *software* para obtener la regresión lineal correspondiente o modelo matemático lineal. Los coeficientes del modelo son incorporados en el procesamiento de datos y adecuar así una respuesta lineal.

De igual forma, el proceso para el monitoreo experimental es sistemático y emplea distintos paquetes de *software* para su ejecución, uno de estos es LabVIEW *SignalExpress*. Este ha sido de utilidad para el SIBEO ya que sus parámetros de funcionamiento son totalmente personalizables y es posible visualizar en pantalla la gráfica de comportamiento de la variable de interés, además de contar también con la opción de generar reportes en Excel con el concentrado de datos. Sin embargo, por las características propias del *software*, no es posible visualizar en pantalla las gráficas de las

seis variables más importantes de manera clara e independiente, algo que de ser posible, ayudaría a comprender con mayor facilidad el análisis y el mismo monitoreo.

Los reportes generados por LabVIEW *SignalExpress* presentan también detalles, como es el exceso de información contenida y el espacio en disco duro a ocupar, sin contar que la cantidad de información que se puede almacenar en los archivos de Excel esta limitada a cierto número de celdas.

La siguiente figura muestra el ambiente de LabVIEW *SignalExpress*.

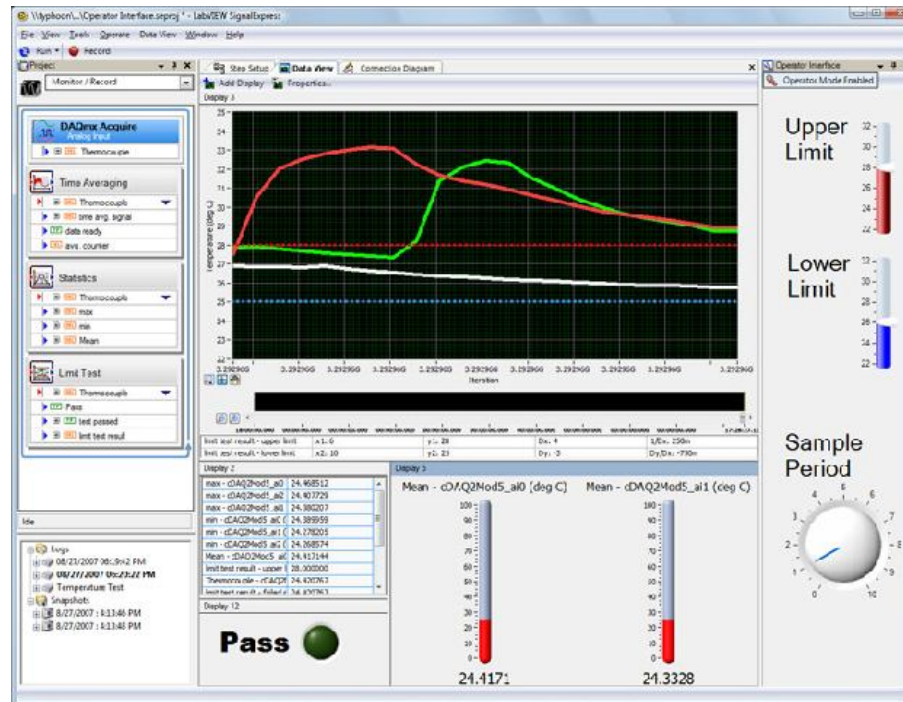


Figura 2. Ambiente de LabVIEW *SignalExpress*.

Por lo anterior, surge la necesidad de contar con sistemas informáticos especializados en la caracterización de sensores y al monitoreo de variables físicas, a fin de solventar las deficiencias existentes y brindar al SIBEO herramientas más eficientes.

1.3 Breve estado del arte

En la actualidad es posible encontrar de manera comercial o en Internet, distintos paquetes de *software* capaces de caracterizar sensores a través de métodos numéricos como la regresión lineal, la regresión polinomial, etc. En estos es necesario ingresar de manualmente (o por importación) los datos y establecer los parámetros de procesamiento para generar las funciones y los coeficientes que mejor se ajustan a la respuesta del sensor. A continuación se describen algunos de estos paquetes de *software*.

Programa 13.2 Regresión lineal y polinomial

Este programa realiza, dada una muestra, un ajuste usando el criterio de mínimos cuadrados. Las opciones son regresión lineal, polinomial de grado 2, 3,..., 6 o regresión lineal a los modelos intrínsecamente lineales: exponencial, potencial, logarítmico y recíproco.

Principales características:

- Opción para ingresar datos manualmente o por importación.
- Opción para escoger el tipo de regresión a ajustar.
- Provee los coeficientes principales de cada regresión.
- Opción para realizar interpolaciones.
- Graficación de los datos tabulares y la regresión resultante.
- Generación de archivos con los resultados obtenidos.

A continuación se presenta el ambiente de este sistema.

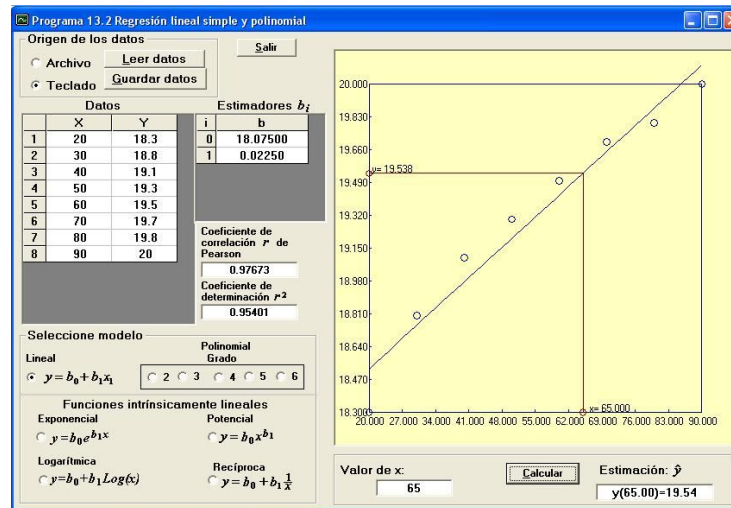


Figura 3. Ambiente gráfico del Programa 13.2 .

STATISTICA

Software compatible con Windows para obtener regresiones lineales y no lineales, así como estimaciones estadísticas.

Principales características:

- Opción para escoger el tipo de regresión a ajustar.
- Generación de regresiones superficiales.
- Se requieren varios pasos para establecer la configuración de la regresión deseada.
- Provee los coeficientes principales de cada regresión.

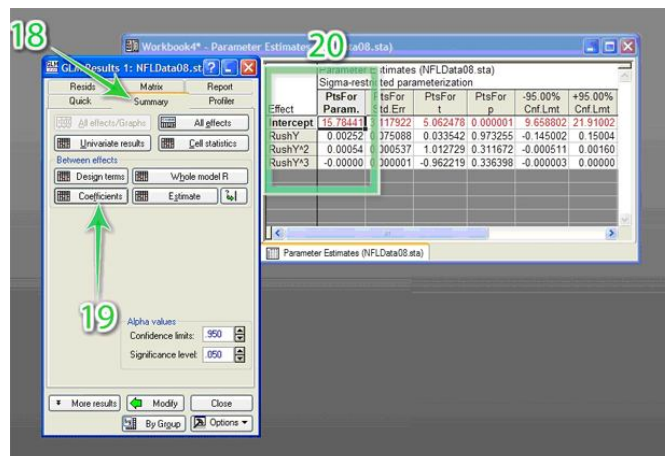


Figura 4. Ambiente gráfico de STATISTICA.

NLREG

Es un programa de análisis estadístico, el cual puede desarrollar regresiones lineales y no lineales, superficies y ajustes de curva. NLREG determina el valor de los parámetros de una ecuación partiendo de un conjunto de datos especificados por el usuario.

Principales características:

- Capaz de obtener cualquier tipo de función.
- Para el ajuste de funciones emplea lenguajes de programación similares a C.
- Capacidad de graficar la dispersión de datos y su correspondiente curva o superficie de ajuste.
- Capacidad de manejar hasta 5 variables y parámetros (versión estándar) provenientes de los datos de observación.
- Capacidad de encontrar la raíz o valor mínimo de una función no lineal.

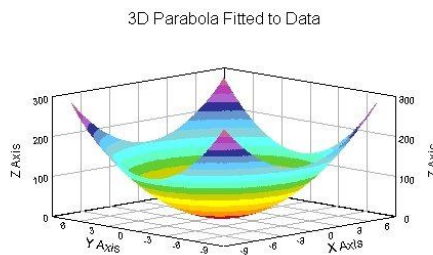


Figura 5. Gráfica en 3D generada en NLREG.

Microsoft Office Excel®

Excel es la herramienta más utilizada para el análisis de datos y estadísticas. Como parte de la paquetería de Microsoft Office, hoy en día es posible encontrar este *software* en cualquier computadora con sistema operativo Windows.

Principales características:

- Opción para importar datos o ingresarlos de manera manual.
- Graficación de datos rápida e intuitiva.
- Obtención de la línea de tendencia.
- Generación de la ecuación de línea de tendencia (lineal y no lineal).
- Generación de archivos compatibles con otros sistemas de procesamiento de datos.
- Opción para generar una gran cantidad de funciones matemáticas, estadísticas, análisis, etc.

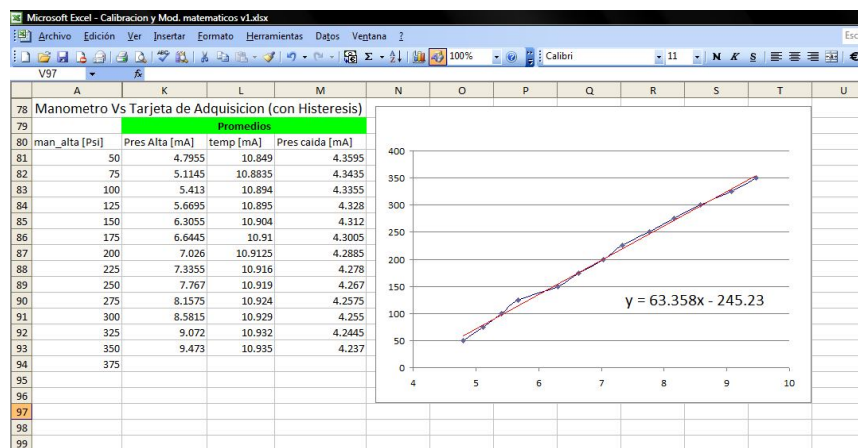


Figura 6. Ambiente de Microsoft Excel®.

Es posible encontrar también:

CaEst 1.2

Entorno de cálculo especializado en operaciones estadísticas, índice de Gini, regresiones lineales, etc.

Origin 8

Entorno para el análisis de datos y el desarrollo de publicaciones científicas de cálculo, generar regresiones lineales, polinómicas, etc.

DataFit 8.2

Cálculos matemáticos con regresiones no lineales y lineales; es posible definir los propios modelos de regresión o escoger entre más de 350.

Kalkulator 2.50

Una calculadora científica muy completa para Windows, con la que es posible generar histogramas, regresiones polinómicas, lineales, logarítmicas, exponenciales, etc.

Cabe mencionar que los sistemas anteriores no cuentan con la opción para ingresar datos desde un dispositivo externo, por lo que el ingresar datos es un aspecto limitado.

En lo que a monitoreo de señales se refiere, los sistemas que se pueden encontrar de manera comercial y adecuados a las distintas necesidades de cada aplicación, son creados para un cierto tipo de aplicación. Es importante mencionar, que si bien no existen sistemas suficientes, si existen varias herramientas informáticas de desarrollo capaces de crear aplicaciones en base a requerimientos especiales. Algunos de estos son: LabVIEW, Visual C#, Delphi® y MatLab.

Visual C#

Microsoft C#® es un lenguaje con seguridad de tipos y orientado a objetos sencillo, que permite a los programadores crear una gran variedad de aplicaciones. Junto a .NET Framework® y Visual C#® permite la creación de aplicaciones para Windows, servicios Web, herramientas de bases de datos, componentes, controles, etc.

.NET Framework es un entorno de ejecución y desarrollo que permite el trabajo conjunto de diferentes bibliotecas y lenguajes de programación para crear aplicaciones basadas en Windows, Web, Mobile y Office.

Principales características:

- Con .NET Framework, las aplicaciones desarrolladas poseen una gran compatibilidad con Windows y sus programas.
- Desarrollo de aplicaciones en lenguaje C.
- Programación intuitiva que acelera el proceso de desarrollo.
- Amplia biblioteca de controles gráficos como: botones, CheckBox, ComboBox, etc.

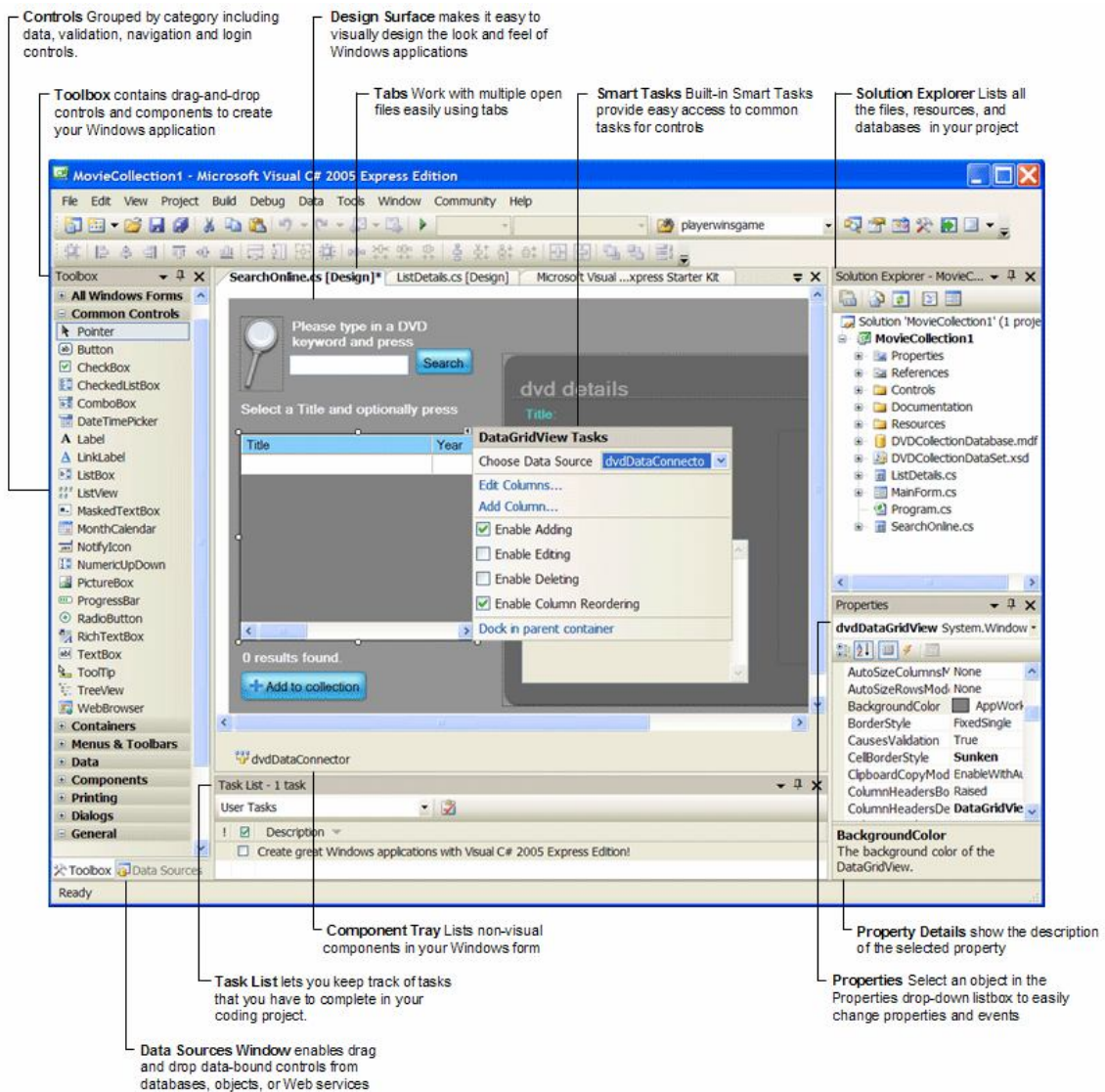


Figura 7. Entorno de desarrollo de Microsoft Visual C# 2005 Express Edition

Delphi® XE

Es una opción adecuada para desarrollar aplicaciones muy rápidas en Windows. Reduce el tipo de programación por código y crea aplicaciones muy rápidas por ser un entorno de desarrollo integrado (IDE, *Integrated Development Environment*). Con Delphi® es posible crear aplicaciones con diseño visual, aplicaciones que requirieren visualización rápida de los datos, control de hardware en tiempo real, manipulación de objetos en 3D, modelos financieros, imágenes, sistemas con punto de escala, etc. Delphi® XE esta disponible en versiones para uso profesional, *enterprise* y arquitectónico.

Principales características:

- RAD IDE con rápido diseño *drag-and-drop*.
- Lenguaje de programación *Object Pascal*.
- Mas de 500 controles visuales.
- Compatible con JSON, REST, HTTP, HTTPS, COM y XML.
- Métricas de código UML.
- Compatible con Windows 2000, XP, Vista, Windows 7 y versiones anteriores de Delphi®.



Figura 8. Interfaz gráfica generada en Delphi® XE

MatLab

Es un lenguaje de alto nivel y ambiente interactivo que permite llevar a cabo tareas computacionales intensivas, en su mayoría cálculos matemáticos, mas rápidamente que con lenguajes de programación tradicionales como C, C++ y *Fortran*. Es posible utilizar MatLab en un gran número de aplicaciones incluyendo procesamiento de señales e imágenes, comunicaciones, diseño de control, mediciones y pruebas, análisis y modelado financiero, etc.

Principales características:

- Desarrollo de ambientes para la administración de códigos, archivos y datos.
- Herramientas matemáticas para algebra lineal, estadística, análisis de *Fourier*, filtración, optimización e integración numérica.
- Funciones gráficas 2D y 3D para visualización de datos.
- Herramientas para crear interfaces gráficas de usuario a la medida de las necesidades.
- Funciones para integrar algoritmos basados en MatLab con aplicaciones y lenguajes externos como C, C++, *Fortran*, Java, COM y Microsoft Excel®.

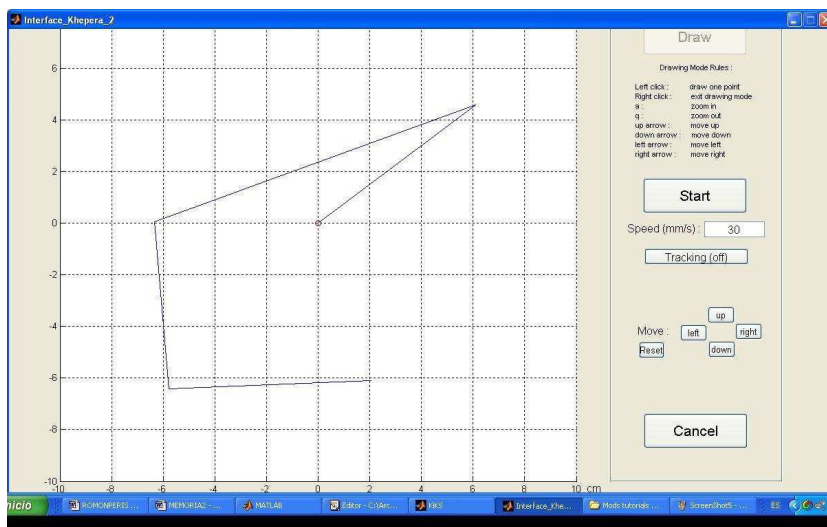


Figura 9. Plataforma gráfica para el control experimental de un robot.

1.4 Solución

Desarrollar dos interfaces gráficas que en base a requerimientos específicos, puedan incrementar la eficiencia del análisis de resultados y el proceso de experimentación. La primera se enfocara en obtener el modelo matemático que mejor describa el comportamiento de cada sensor (caracterización de sensores). La segunda interfaz se enfocará en el monitoreo de las variables físicas mas importantes del proceso de experimentación con el SIBEO. Ambas interfaces serán capaces de generar reportes, en los cuales se almacenen los resultados obtenidos en cada proceso.

Dado que LabVIEW es un entorno de desarrollo gráfico con amplias herramientas y funciones para crear aplicaciones, será empleado para llevar a cabo dicha solución. Ante este panorama el proyecto del SIBEO cuenta con una licencia de uso (LabVIEW 8.5 *Professional Development System with MathScript*) con todas las funciones y comandos necesarios.



Figura 10. Pantalla de inicio de LabVIEW 8.5.

Para la adquisición de datos, ya sea para el proceso de caracterización o bien para el monitoreo de variables críticas, se emplearan dos tarjetas de adquisición de datos, cuyos puertos de entrada sean capaces de aceptar señales de voltaje en un rango mínimo de 5 a 10 volts y con una interfaz de comunicación hacia la computadora. El proyecto dispone de dos DAQS de la marca National Instrument, modelo NI USB-6009 de tipo *plug and play* (las especificaciones técnicas se encuentran en el Anexo 1) muy bien adecuados para solventar esta parte.



Figura 11. Tarjeta de Adquisición de Datos Multifuncional NI USB-6009

El valor agregado mas importante que se puede resaltar de la solución técnica mencionada, es que el proyecto no requiere realizar ninguna adquisición adicional de equipos o paquetes de software, lo que denota en ahorros considerables e inmediatos al presupuesto aprobado para el proyecto.

1.4.1 LabVIEW¹

LabVIEW (*Laboratory Virtual Instrument Engineering Workbench*) es un lenguaje de programación gráfico, creado por la empresa National Instruments que utiliza iconos en lugar de líneas de texto para crear aplicaciones. Provee un ambiente de programación más sencillo que los entornos tradicionales. En contraste con los lenguajes de programación basados en texto, donde las instrucciones determinan el orden de la ejecución de los programas, LabVIEW usa la programación en “Flujo de Datos”, donde el flujo de los datos a través de los nodos en un diagrama de bloque determina el orden de ejecución de las funciones.

Posee un compilador el cual genera un código optimizado que hace que las aplicaciones tengan velocidades de ejecución comparables con programas C compilados, así como su compatibilidad con varios sistemas operativos como Windows®, Macintosh®, UNIX® y sistemas de tiempo real.

Principales usos

Es usado por ingenieros y científicos para tareas como:

- Adquisición de datos y análisis matemático.
- Comunicación y control de instrumentos de cualquier fabricante.
- Diseño de controladores: simulación, prototipo rápido y validación.
- Control y supervisión de procesos.
- Visión artificial y control de movimiento.
- Domótica.

Principales características

Su principal característica es la facilidad de uso, tanto para programadores profesionales como para personas con pocos conocimientos en programación ya que pueden hacer programas relativamente complejos. Los programas en LabVIEW son llamados instrumentos virtuales (VIs) y es posible crear programas de miles de VIs (equivalente a millones de páginas de código texto) para aplicaciones complejas, programas de automatización de miles de puntos de entradas/salidas, proyectos para combinar nuevos VIs con VIs ya creados, etc.

Instrumentos virtuales o VIs

En LabVIEW es posible construir VIs utilizando una variedad de herramientas y objetos. Estos VIs son construidos en una interfaz de usuario conocida como “Panel Frontal”, después se agrega el código de programación en un “Diagrama de Bloque” usando representaciones gráficas de las funciones para controlar los objetos del panel frontal. Este código de recurso gráfico es también conocido como código G o código de diagrama de bloque.

Un VI contiene los siguientes tres elementos:

- *Panel Frontal*: Sirve como interfaz de usuario.
- *Diagrama de Bloque*: Contiene los códigos gráficos que definen la funcionalidad del VI.
- *Icono-Conector*: Identifica la interfase del VI por lo que es posible usar el VI en otro VI, conocido como SubVI. Un SubVI corresponde a una subrutina en lenguajes de programación basados en texto.

1.- Para una consulta mas detallada: *LabVIEW Fundamentals*, National Instrument, April 2003 Edition.

1.4.2 Caracterización de sensores

Un sensor es un dispositivo que transforma un tipo de energía o variable física en otra, normalmente en energía eléctrica, y se utiliza para medir la variable en cuestión.

Los sensores pueden clasificarse en dos tipos básicos, dependiendo de la forma de la señal convertida:

- Analógicos
- Digitales

Los sensores analógicos proporcionan una señal analógica continua, por ejemplo voltaje o corriente eléctrica. Esta señal puede ser tomada como el valor de la variable física que se mide.

Los sensores digitales producen una señal de salida digital, en forma de un conjunto de bits de estado en paralelo o formando una serie de pulsaciones que pueden ser contadas. En una u otra forma, las señales digitales representan el valor de la variable medida. Los sensores digitales suelen ofrecer la ventaja de ser más compatibles con las computadoras que los sensores analógicos.

Los siguientes términos se emplean para definir el funcionamiento de los sensores y los sistemas de medición.

Rango y margen. El rango de un sensor define los límites entre los cuales puede variar la entrada. El margen es el valor máximo de la entrada menos el valor mínimo.

Error. Es la diferencia entre el resultado de la medición y el valor verdadero de la cantidad que se mide.

$$\text{Error} = \text{valor medido} - \text{valor real} \dots \dots \dots (1.1)$$

Exactitud. Es el grado hasta el cual un valor producido por un sistema de medición podría estar equivocado. Es por lo tanto igual a la suma de todos los errores posibles más el error en la exactitud de calibración del transductor.

Sensibilidad. Es la relación que indica que tanta salida se obtiene por unidad de entrada, es decir, salida/entrada. Es frecuente que este término también se utilice para indicar la sensibilidad de otras entradas.

Resolución. Cuando la entrada varía continuamente en todo el rango, las señales de salida de algunos sensores pueden cambiar a pequeños intervalos. La resolución es el cambio mínimo del valor de la entrada capaz de producir un cambio observable a la salida.

Impedancia de salida. Cuando un sensor que produce una salida eléctrica, se conecta a un circuito electrónico, es necesario conocer la impedancia de salida, debido a que esta se va a conectar en serie o en paralelo a dicho circuito. Al incluir el sensor, el comportamiento del sistema con el que se conecta se podría modificar de manera considerable

Cualquier sensor necesita estar calibrado y caracterizado para ser útil como dispositivos de medida. La caracterización es el proceso por el cual se establece una relación entre la variable medida y la señal de salida obtenida. En este proceso se debe obtener el modelo matemático que mejor describa el comportamiento del sensor, así como los parámetros que lo forman.

Modelo Matemático

Un modelo matemático se define como una descripción, desde el punto de vista de las matemáticas, de un hecho o fenómeno del mundo real, desde el tamaño de la población, hasta fenómenos físicos como la velocidad, aceleración o densidad. El objetivo del modelo matemático es entender ampliamente el fenómeno y tal vez predecir su comportamiento en el futuro.

Es importante mencionar que un modelo matemático no es completamente exacto con problemas de la vida real, de hecho, se trata de una idealización.

Hay una gran cantidad de funciones que representan relaciones observadas en el mundo real, a continuación se presentan dos de las más importantes, tanto algebraicamente como gráficamente.

Modelos Lineales. Una función es lineal cuando su gráfica es una línea recta y por consecuencia tiene la forma:

$$y = f(x) = m \cdot x + b \dots\dots\dots (1.2)$$

Donde **m** representa la pendiente de la recta y **b** la ordenada al origen (el punto en el que la recta intersecta al eje de las "y"). Es importante mencionar que este tipo de funciones crecen a tasa constante; su dominio e imagen son todos los números reales.

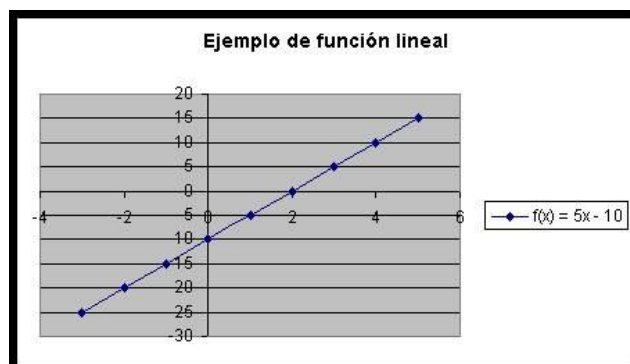


Figura 12. Función lineal

Modelos Polinómicos. Una función es un polinomio si tiene la forma:

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0 \dots\dots\dots (1.3)$$

Donde **n** representa un entero positivo y los números **a₀, a₁, a₂, ..., a_n**, son constantes llamadas coeficientes del polinomio. El dominio de todos los polinomios son todos los números reales.

Los polinomios se nombran de acuerdo al grado del primer término. Los polinomios de grado uno son de la forma: $P(x) = mx + b$, y son funciones lineales. Los polinomios de segundo grado son llamados funciones cuadráticas y presentan la forma $P(x) = ax^2 + bx + c$; su gráfica es una parábola. Una función de tercer grado, es llamada función cúbica y tiene la forma: $P(x) = ax^3 + bx^2 + cx + d$.

Para obtener un modelo matemático a partir de una dispersión de puntos, se emplean generalmente las teorías de métodos numéricos como: regresión lineal, para obtener modelos lineales o regresión polinomial, para obtener modelos polinómicos.

Regresión Lineal

Un análisis de regresión investiga la relación algebraica que existe entre una variable dependiente (Y) y una o más variables independientes (X₁, X₂, X₃,...). Para poder realizar esta investigación, se

debe postular una relación funcional entre las variables. Debido a su simplicidad analítica, la forma funcional que más se utiliza en la práctica es la relación lineal. Cuando sólo existe una variable independiente, esto se reduce a una función lineal:

$$Y = b_0 + b_1X \dots\dots\dots (1.4)$$

El problema consiste en obtener estimaciones de estos coeficientes a partir de una muestra de observaciones sobre las variables Y y X. En el análisis de regresión, estas estimaciones se obtienen por medio del método de mínimos cuadrados.

Para poder visualizar el grado de relación que existe entre las variables, como primer paso es conveniente elaborar un diagrama de dispersión (figura 16), que es una representación en un sistema de coordenadas cartesianas de los datos numéricos observados.

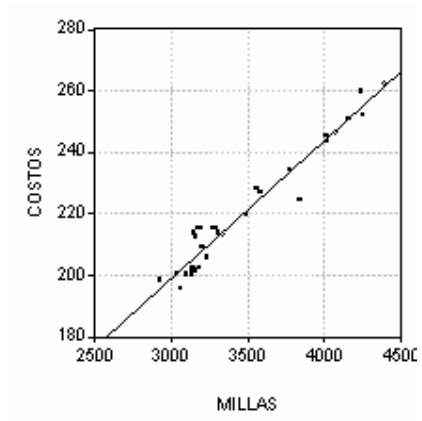


Figura 13. Diagrama de dispersión millas contra costos

Se denomina **diagrama de dispersión** porque no existe una relación matemáticamente exacta entre las variables. Si entre estas variables existiera una relación lineal perfecta, entonces todos los puntos caerían a lo largo de la recta de regresión, que también ha sido trazada y que muestra la relación "promedio" que existe entre las dos variables.

Para estimar los coeficientes por medio de mínimos cuadrados, se utilizan las siguientes fórmulas:

$$b_1 = \frac{\sum XY - \bar{y} \sum X}{\sum X^2 - \bar{x} \sum X} \dots\dots\dots (1.5)$$

$$b_0 = \bar{y} - b_1 \bar{x} \dots\dots\dots (1.6)$$

Regresión Polinomial

Debido a que muchos fenómenos reales son representados pobremente mediante una regresión lineal recta, es necesario ajustar curvas en lugar de rectas. Una alternativa es emplear la regresión polinomial.

El procedimiento de mínimos cuadrados se puede extender fácilmente y ajustar datos a un polinomio de m-esimo grado:

$$y = a_0 + a_1x + a_2x^2 + \dots\dots\dots + a_mx^m \dots\dots\dots (1.7)$$

En este caso la suma de los cuadrados de los residuos es:

$$S_r = \sum_{i=1}^n (y_i - a_0 - a_1 x_i - a_2 x_i^2 - \dots - a_m x_i^m)^2 \dots\dots\dots (1.8)$$

Se toma la derivada de la ecuación 1.8 con respecto a cada uno de los coeficientes del polinomio, para obtener:

$$\frac{\partial S_r}{\partial a_0} = -2 \sum (y_i - a_0 - a_1 x_i - a_2 x_i^2 - \dots - a_m x_i^m) \dots\dots\dots (1.9)$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum x_i (y_i - a_0 - a_1 x_i - a_2 x_i^2 - \dots - a_m x_i^m) \dots\dots\dots (1.10)$$

$$\frac{\partial S_r}{\partial a_2} = -2 \sum x_i^2 (y_i - a_0 - a_1 x_i - a_2 x_i^2 - \dots - a_m x_i^m) \dots\dots\dots (1.11)$$

⋮

$$\frac{\partial S_r}{\partial a_m} = -2 \sum x_i^m (y_i - a_0 - a_1 x_i - a_2 x_i^2 - \dots - a_m x_i^m) \dots\dots\dots (1.11)$$

Estas ecuaciones se pueden igualar a cero y reordenar de tal forma que se obtenga el siguiente conjunto de ecuaciones normales:

$$a_0 n + a_1 \sum x_i + a_2 \sum x_i^2 + \dots + a_m \sum x_i^m = \sum y_i \dots\dots\dots (1.12)$$

$$a_0 \sum x_i + a_1 \sum x_i^2 + a_2 \sum x_i^3 + \dots + a_m \sum x_i^{m+1} = \sum x_i y_i \dots\dots\dots (1.13)$$

$$a_0 \sum x_i^2 + a_1 \sum x_i^3 + a_2 \sum x_i^4 + \dots + a_m \sum x_i^{m+2} = \sum x_i^2 y_i \dots\dots\dots (1.14)$$

⋮

$$a_0 \sum x_i^m + a_1 \sum x_i^{m+1} + a_2 \sum x_i^{m+2} + \dots + a_m \sum x_i^{2m} = \sum x_i^m y_i \dots\dots\dots (1.15)$$

En donde todas las sumatorias van desde **i = 1** hasta **n**. Notese que las **m+1** ecuaciones anteriores son lineales y tienen **m+1** incógnitas: a_0, a_1, \dots, a_m . Los coeficientes de las incógnitas se pueden calcular directamente de los datos observados. Por lo tanto, el problema de determinar polinomios de grado **m** con mínimos cuadrados es equivalente a resolver un sistema de **m+1** ecuaciones lineales simultáneas.

1.5 Programación en base al modelo *cascada*

Los proyectos de desarrollo de *software* son complejos y para tratar con estas complejidades, muchos desarrolladores recurren a los principios de desarrollo específicos. Estos principios definen el campo de la ingeniería de *software*. El mayor componente de este campo son los modelos de ciclo de vida, los cuales describen los pasos a seguir cuando se desarrolla un *software* (desde el concepto inicial hasta la culminación de la versión, el mantenimiento y subsecuentemente la actualización).

Actualmente existen diferentes modelos de ciclo de vida², cada uno tiene ventajas y desventajas en términos de tiempo de finalización, calidad y administración de riesgos. Para cada modelo es importante conocer los requerimientos y especificaciones del proyecto, contemplando posibles cambios.

El modelo de ciclo de vida es una fundación para la totalidad del proceso de desarrollo. Las buenas decisiones pueden repercutir en la calidad y disminuir los tiempos que se están tomando para desarrollarlo.

Modelo Cascada

El modelo de cascada es clásico en la ingeniería de *software*. Este modelo es el primero de los modelos más antiguos. Es altamente usado en proyectos gubernamentales y en muchas grandes compañías, debido a que enfatiza la planeación en las primeras etapas, pudiendo identificar las fallas de diseño antes de ser desarrolladas. También, porque trabaja bien para proyectos en los cuales el control de calidad es el aspecto más importante.

El modelo simple consiste de varios estados interconectados sistemáticamente donde la información fluye y se reanaliza en cada estado, la figura 14 muestra lo anterior. Comienza con establecer los requerimientos del sistema y de software, y continúa con el diseño arquitectural, diseño detallado, código, pruebas y mantenimiento. Este modelo también sirve de punto de referencia para muchos otros modelos de ciclo de vida.

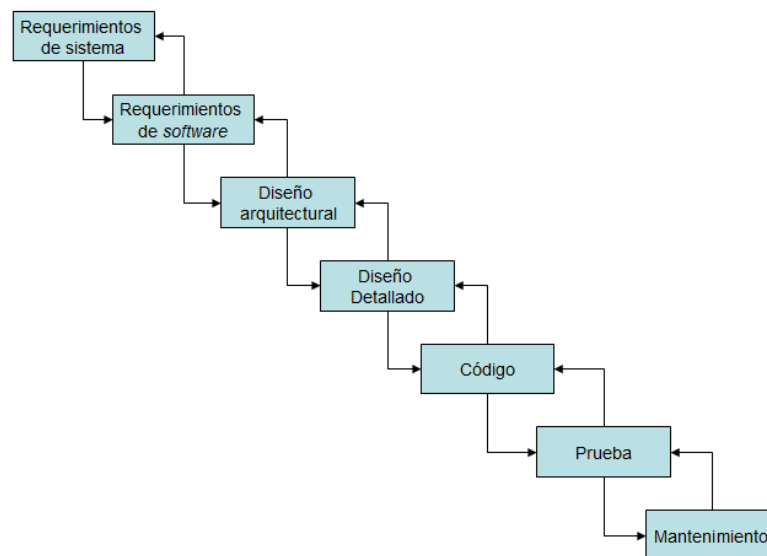


Figura 14. Diagrama conceptual del modelo *cascada*

2.- En "LabVIEW™ Intermediate I: Successful Development Practices Course Manual" se presenta cada uno de estos modelos de ciclo de vida.

- *Requerimientos del sistema:* Establecen los componentes para construir el sistema, incluyendo los requerimientos de hardware, herramientas de software y otros componentes necesarios. Ejemplos de estos requerimientos incluyen decisiones en hardware como: número de canales, velocidad de adquisición, etc., y decisiones en software como bases de datos y librerías.
- *Requerimientos de software:* Establece las expectativas para la funcionalidad del sistema e identifica que requerimientos influyen. El análisis de los requerimientos incluye determinar la interacción necesaria con otras aplicaciones y bases de datos, requerimientos de rendimiento, requerimiento de la interfaz de usuario, etc.
- *Diseño Arquitectural:* Determina la estructura en un sistema para conocer los requerimientos específicos. El diseño define los componentes principales y la interacción entre ellos, sin embargo no define la estructura de cada componente.
- *Diseño detallado:* Examina los componentes de software definidos en el diseño arquitectural. Especifica como cada componente debe ser implementado.
- *Código:* Implementa las especificaciones del diseño detallado.
- *Pruebas:* Determina si el software cumple con los requerimientos específicos y detecta cualquier error presente en el código.
- *Mantenimiento:* Canaliza los problemas y sugiere mejoras después al término del software.

En cada fase se debe crear la documentación que explique los objetivos y describa los requerimientos para esa fase. Al final de cada fase, es necesario establecer una revisión para determinar si el proyecto puede continuar a la siguiente fase.

Mucha gente cree que no es posible aplicar este modelo a todas las situaciones, por ejemplo, con el modelo sencillo, es indispensable establecer los requerimientos antes de comenzar con el diseño, y establecer un diseño completo antes de comenzar con el código. No hay superposición entre fases. En el desarrollo del mundo real, se descubren problemas durante las fases de diseño o código que provienen directamente de los requerimientos.

El modelo cascada no prohíbe regresar a una anterior, por ejemplo, desde la fase de diseño hasta la fase de requerimientos. Sin embargo, esto involucra re-trabajos costosos. Cada fase completada requiere una revisión formal y un la elaboración de la documentación correspondiente.

Debido a que el desarrollo es tardío en el proceso, no es posible ver resultados inmediatos. Este retraso debe ser considerado en el rendimiento. Aunque el modelo cascada tiene sus limitantes, es instructivo porque enfatiza importantes fases para el desarrollo de un proyecto. Aun si no se aplica el modelo, es importante considerar cada una de sus fases y relación en cada proyecto.

Capítulo 2. Diseño conceptual

2.1 Metodología de desarrollo

Para el desarrollo del presente trabajo la metodología se basa en el modelo *casca*, pues su estructura es sistemática, eficaz e ideal para resolver la problemática expuesta. A continuación los pasos a seguir.

“Requerimientos de Sistema y Software”

1.- Obtención de los requerimientos funcionales y operacionales.

Con la participación del director del proyecto SIBEO, establecer los requerimientos funcionales y operativos que cada interfaz debe cumplir.

2.- Sensores y equipos a emplear.

Identificar el equipo de medición a utilizar, en especial su tipo de señal de salida.

“Diseño Arquitectural”

3.- Elaboración de prototipos rápidos.

Definir el diseño gráfico de cada interfaz empleando prototipos rápidos de VIs. De igual forma, en este punto se contará con la colaboración del director del proyecto SIBEO, a fin de establecer conjuntamente el ambiente gráfico más agradable y útil.

“Diseño Detallado”

4.- Creación del algoritmo de programación.

Definir el conjunto finito de pasos e instrucciones que de manera detallada describan las funciones que deban aplicarse en cada situación de la ejecución.

5.- Elaboración de diagramas de flujo.

Elaborar diagramas de flujo para visualizar gráficamente el procesamiento de la información; este paso es sumamente importante ya que a partir de estos diagramas es posible crear los códigos de programación e identificar posibles errores en el desarrollo.

“Código”

6.- Programación de interfaces de caracterización y monitoreo.

Elaboración de los códigos de programación tanto para la interfaz de caracterización como para la interfaz de monitoreo experimental.

“Pruebas”

7.- Incorporación de mejoras.

Identificar los errores, corregirlos y validarlos, a fin de mejorar en la mayor medida posible el rendimiento de los sistemas.

8.- Pruebas de laboratorio.

Realizar pruebas de validación bajo ambientes controlados y asegurar que los sistemas son capaces de satisfacer los requerimientos.

9.- Manual de operación.

Elaborar el manual de operación de cada interfaz, que permita al usuario operar los sistemas, sin la necesidad de leer el código de programación o sin intervención de alguna capacitación personal en un futuro.

2.2 Requerimientos para el desarrollo

En coordinación con el Dr. Steven Czitrom Baus, director del proyecto del SIBEO, se establecen los siguientes requisitos mínimos para cada interfaz.

- A través del software LabVIEW desarrollar dos interfaces que trabajen de manera independiente. Una de estas se llamara “Modo Calibración” y servirá para realizar el proceso de caracterización de sensores. La segunda interfaz se llamara “Modo Experimental” y será utilizada para llevar a cabo el monitoreo de las variables críticas del SIBEO.
- Emplear las dos tarjetas NI USB-6009, propiedad del proyecto, para adquirir los datos de ambas interfaces.

La interfaz Modo Calibración:

- Debe mostrar la hora y fecha en la que se llevo a cabo la caracterización, en formato día, mes y año (Ej. domingo, 23 de agosto de 2009) , en horas y minutos (Ej. 6:50pm).
- Debe contar con una función para especificar la ruta en la cual se deben almacenar los archivos de resultado.
- La caracterización de los sensores debe ser preferentemente de fácil visualización, es decir, ubicar con facilidad aquellos apartados en donde se este llevando a cabo, o bien, poder alternar entre apartados de manera sencilla.
- El apartado de cada sensor debe admitir (de manera manual) valores arbitrarios de referencia, para después mostrar los niveles de voltaje correspondientes, presentes en la tarjeta de adquisición.
- El conjunto de valores formado por los datos arbitrarios y adquiridos deben ser mostrados en una gráfica para facilitar su análisis.
- Mediante una regresión lineal, el sistema debe calcular los parámetros de la recta que mejor represente la respuesta del sensor. Después debe mostrar estos parámetros en pantalla.
- Debe generar dos archivos compatibles con programas alfanuméricos de edición como Microsoft Excel® y Block de Notas®, ambos deben contener fecha y hora de la caracterización. Uno de ellos almacenara los datos de referencia y su correspondiente nivel de voltaje adquirido; el otro, los parámetros que resultaron de cada regresión lineal.
- El nombre del archivo generado será: Fecha_Hora_Calib (Ej. 23ago2009_6:50pm_Calib).

La Interfaz Modo Experimental:

- Debe mostrar la fecha y hora en la que se llevo a cabo la prueba experimental, en formato día, mes y año (Ej. domingo, 23 de agosto de 2009), en horas y minutos (Ej. 6:50pm). También,

debe mostrar los parámetros de experimentación como: Periodo (cs), Amplitud (cm), Marea (mm) y No. de Experimento (enteros), estos deben ser ingresados manualmente por el usuario.

- Debe contar con una opción para cargar el archivo generado en la interfaz “Modo Calibración” (archivo Calib), pues los parámetros de la regresión lineal deberán ser utilizados por el sistema para procesar los niveles de voltaje y obtener niveles físicos correspondientes.
- Debe mostrar simultáneamente las gráficas de monitoreo de los 6 sensores, sólo para identificar el momento en que los niveles llegan a un estado estable. Dichas gráficas deben ser de tipo barrido.
- En el momento que los niveles físicos presenten un comportamiento estable, se deberá disponer de un control cuya función guarde los voltajes en memoria por un lapso definido por el usuario, a una frecuencia de muestreo de 10ms; esta frecuencia también es de monitoreo. Se propone que mediante un botón y un *Led* indicador se active esta función.
- Contar con un botón para detener la aplicación.
- Al momento de detener la aplicación, los datos guardados en memoria deberán almacenarse en archivos compatibles con programas alfanuméricos comunes como Excel y Block de Notas, a manera de reporte, en la misma ruta de la cual se extrajo el archivo generado por la interfaz “Modo Calibración”. Estos archivos deberán contener los datos de experimentación y el nombre del archivo de *Calib* para referencia. Uno de estos archivos contendrá los datos crudos del experimento y otro los niveles de altura y presión experimental.

2.3 Instrumentación a emplear

El proyecto SIBEO ha diseñado y construido seis sensores de bajo nivel, empleando transductores de tipo capacitivo, filtros pasivos, etc. para el acondicionamiento de la señal. Varios de estos han sido probados bajo ambientes controlados y han demostrado tener gran precisión de lectura. Proveen una señal de voltaje variable en un rango de 0 a 10 volts. Dado que estos sensores serán utilizados en el monitoreo de las variables críticas del SIBEO serán también empleados en el proceso de caracterización.

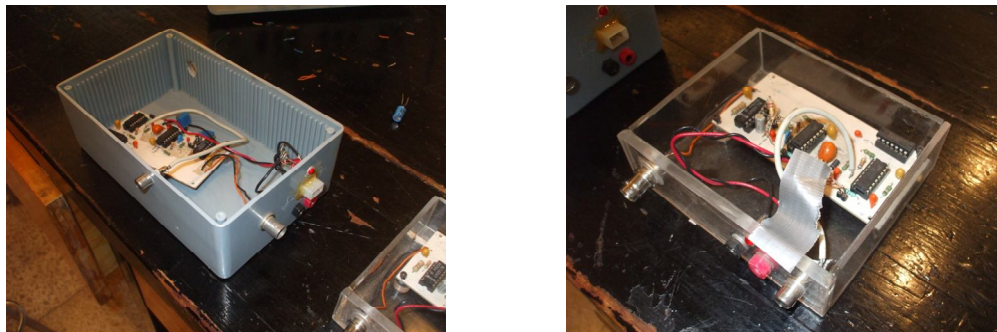


Figura 15 y 16. Sensores fabricados por el proyecto SIBEO

En el caso de que la estabilidad y precisión de estos sistemas se vea alterada por la constate manipulación, o bien por el proceso de desarrollo en el que varios aun se encuentran, el proyecto a adquirido dos sensores de presión, marca KELLER, de diseño robusto y de uso industrial. Estos también proveen una salida de 0 a 10 volts. Dado su alto nivel de estabilidad y precisión, su

constante caracterización no resulta primordial, por lo que serán empleados principalmente en el proceso de monitoreo.



Figura 17. Sensor de presión KELLER para el monitoreo experimental

2.4 Diseño gráfico de interfaces

En este apartado se muestran los distintos ambientes gráficos propuestos a lo largo del proyecto, así como sus principales características y desventajas. La siguiente figura muestra el primer diseño gráfico propuesto para la interfaz “Modo Experimental”, contemplando los elementos básicos requeridos.

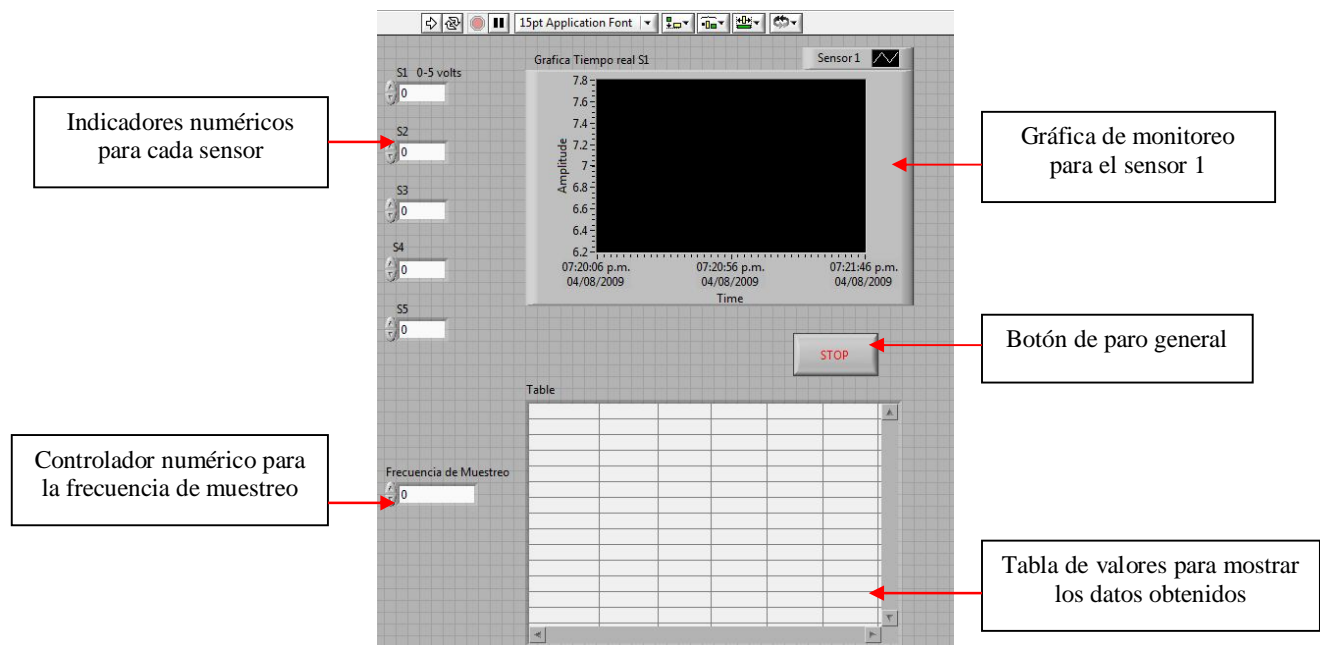


Figura 18. Primer prototipo para la interfaz de monitoreo

Como se puede apreciar, este prototipo se encuentra alejado de los requerimientos, pues sólo presenta un bosquejo de algunos elementos básicos, sin contar que no existe claridad en los indicadores de algunos elementos como por ejemplo: S1, S2, Table, etc. La “Gráfica Tiempo real S1” muestra información de sobra (la fecha) en su eje de tiempo. Existe una falta de coherencia en la ubicación de elementos pues el botón “Stop” se encuentra a la mitad de la interfaz. Carece de colores y contrastes entre elementos, lo que propicia la dificultad para ubicar con facilidad cada elemento.

Estos y otros principios de diseño se encuentran enlistados en la Norma ISO 9241³ (UNE-EN ISO 9241) la cual trata el diseño ergonómico de programas para equipos con pantallas de visualización de datos.

La figura 19 muestra el diseño gráfico para la interfaz de “Modo Calibración”, el cual cuenta con módulos de caracterización, lo cual facilita su identificación, concentrando la información de manera independiente.

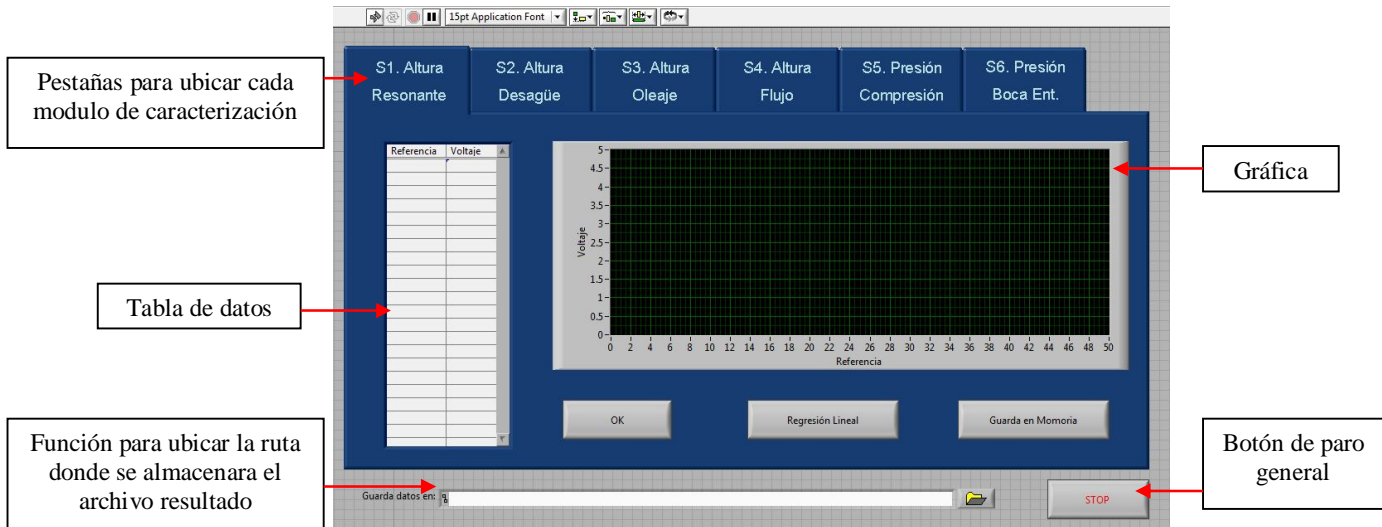


Figura 19. Prototipo inicial de la interfaz “Modo Calibración”

Cuenta con la función “Tabla de datos” donde es posible ingresar los valores de referencia y los adquiridos. Esta tabla de valores serviría para generar una gráfica con la respuesta del sensor. El botón “Ok” ordena a las tarjetas electrónicas adquirir los valores de voltaje. El botón “Regresión Lineal” realiza el procesamiento de datos interno para obtener el modelo lineal mejor adecuado. El botón “Guarda en Memoria” almacenaría todos los datos obtenidos en memoria.

Siguiendo la Norma ISO 9241 se puede apreciar un aumento en el manejo de colores y contrastes, así como una coherencia en la ubicación de elementos, información concisa e intuitiva en la tabla de valores y botones. La proximidad entre los botones de control refleja una relación funcional entre ellos: el control sobre los datos de la gráfica.

El principal defecto de esta interfaz, es la complejidad en su programación, pues cada modulo tiene su propia función “tabla de datos”, “grafica”, botones de regresión lineal y guarda en memoria, etc., que implica realizar una misma función en varias ocasiones pero empleando distintas variables, esto afecta el rendimiento general del sistema considerablemente.

Tras realizar diversas modificaciones se establecieron las siguientes interfaces para desarrollar principalmente por presentar un ambiente de manejo más sencillo para los colaboradores del proyecto y por reunir la mayor cantidad de principios de diseño según la normativa ISO.

3.- En el libro “Sistemas SCADA” de Anquillo Rodríguez Penin se muestra más a detalle cada principio de diseño para la creación de pantallas de visualización de datos.

Modo Calibración

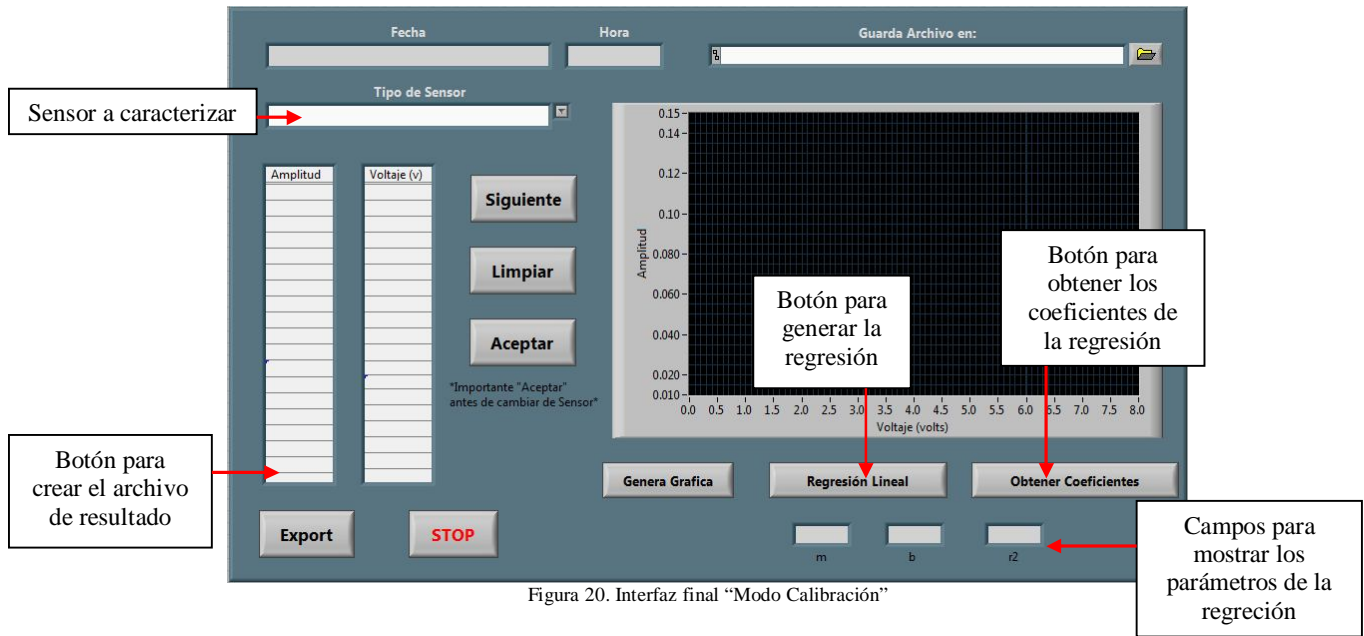


Figura 20. Interfaz final "Modo Calibración"

Esta interfaz, además de mostrar la información de fecha, hora, etc., cuenta también con el control "Tipo de Sensor" el cual despliega una lista de los sensores empleados en el SIBEO (figura 15 y 16), de tal forma que el usuario pueda seleccionar uno de estos para iniciar el proceso. Debajo de este control se encuentra el botón "Siguiete" el cual tiene la misma función que el botón "OK" en la primera interfaz (figura 19). El Botón "Limpiar" tendrá la función de borrar todos los datos mostrados en pantalla, en el caso de que exista algún error. "Aceptar" tendrá la misma función que el botón "Guarda en Memoria" en la primera interfaz (figura 19).



Figura 21. Lista de sensores a caracterizar.

Una vez ingresados los datos de referencia y los datos de voltaje, es posible generar una distribución de valores con el botón "Genera Gráfica", mostrando en pantalla la respuesta del sensor. Con el botón "Regresión Lineal" se obtendrá la función lineal que mejor se adecue a la respuesta. Para determinar los coeficientes de pendiente (m), ordenada al origen (b) y error (r^2), será necesario presionar el botón "Obtener Coeficientes". La función "Export" creará los archivos resultado, en la ruta definida.

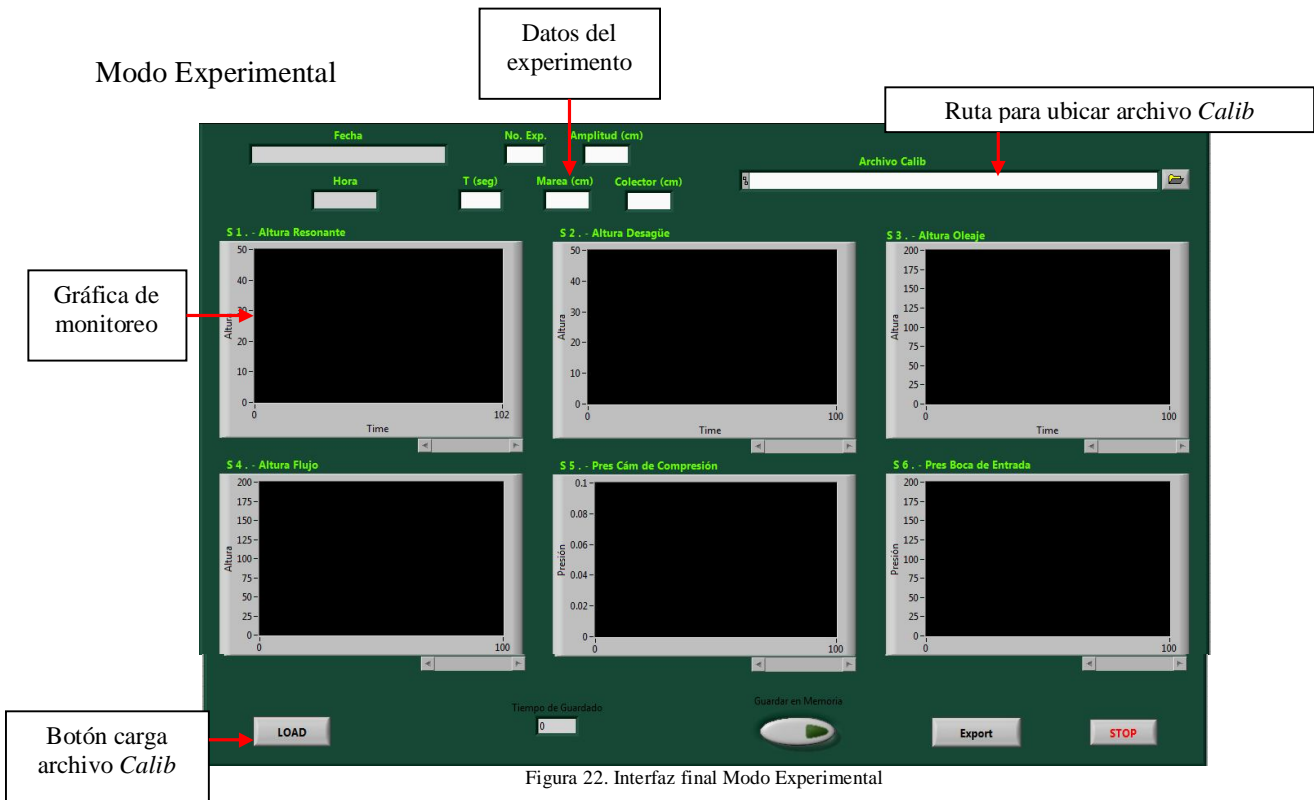


Figura 22. Interfaz final Modo Experimental

Esta interfaz contiene todos y cada uno de los elementos necesarios para realizar el monitoreo experimental: campos para mostrar la fecha y hora de la realización del experimento, parámetros de identificación (número del experimento, amplitud del oleaje, periodo de oscilación, nivel de marea y nivel en el colector). Cuenta también con el control “Archivo Calib” para especificar la ruta del archivo *Calib*, ya sea ingresando la ruta manualmente o seleccionándola con un navegador de archivos. Cuenta con seis gráficas de barrido para cada sensor, como lo estipula los requerimientos.

La principal característica de esta interfaz es el comando “LOAD”, cuya función es extraer del archivo *Calib* los coeficientes de cada regresión lineal. Estos valores serán incorporados al procesamiento interno del sistema para mostrar en cada grafica los niveles de altitud y presión determinados.

Uno de los requisitos más importantes para esta interfaz fue el poder almacenar en memoria los niveles de altitud y presión por un determinado tiempo. Esto se logra ingresando un “Tiempo de guardado” en segundos y presionando el botón “Guardar en Memoria”, cuyo *Led* se encenderá en color verde para indicar que se esta llevando a cabo el proceso.

El comando “Export” tendrá la misma función que en la interfaz “Modo Calibración”, crear los archivos resultado.

2.5 Algoritmos de programación

Antes de enlistar el algoritmo de programación, es importante considerar que en LabVIEW muchos aspectos de programación están sobre-entendidos al momento de utilizar un objeto, por ejemplo, en varios programas de desarrollo por línea de código, antes de utilizar una variable, esta debe ser declarada e inicializada, en LabVIEW los objetos aparecen por defecto con un tipo de dato

especifico (I16, abc, DBL, etc.) y una inicialización a el valor “Null”. Por tal motivo no se detallaran estos tipos de código predefinido.

Pensando en un sistema dedicado e intransferible se establece un diagrama de conexión para las tarjetas de adquisición de datos y los sensores a emplear, designando una tarjeta únicamente para los sensores de altura y la otra para los sensores de presión (figura 23), distribuyendo los sensores en cada puerto de entrada analógica (EA).

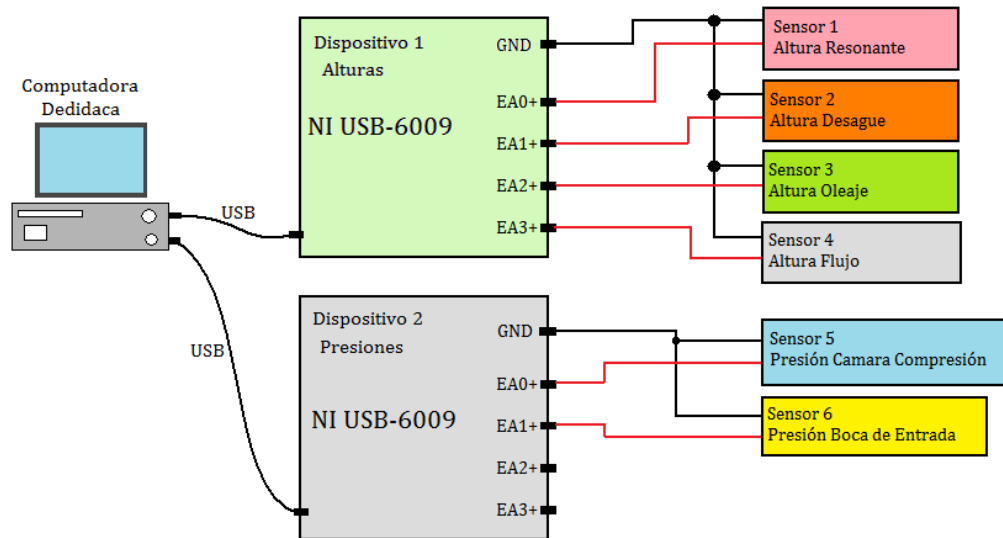



Figura 23. Diagrama de conexiones entre los DAQS y los sensores

Considerando los puntos anteriores, se procede a generar el algoritmo de programación para cada interfaz:

Modo Calibración

- 1.- Activar el botón “Run”  para comenzar la aplicación.
- 2.- Obtener la fecha y hora instantánea de la computadora.
- 3.- Mostrar fecha y hora en los campos correspondientes, en formato: *Lunes, 28 de julio de 2010 y 3:00pm.*
- 4.- Inicializar en cero las tablas, la gráfica, variables de control y procesamiento interno de la información.
- 5.- Si el usuario selecciona el tipo de sensor:
 - **Sensor 1. Altura Resonante**, indicar a la tarjeta que utilizará el puerto de entrada analógica 0 del dispositivo 1 (Disp1/EA0) para adquirir datos.
 - **Sensor 2. Altura Desagüe**, indicar a la tarjeta que utilizará el puerto de entrada analógica 1 del dispositivo 1 (Disp1/EA1) para adquirir datos.
 - **Sensor 3. Altura Oleaje**, indicar a la tarjeta que utilizará el puerto de entrada analógica 2 del dispositivo 1 (Disp1/EA2) para adquirir datos.

- **Sensor 4. Altura Flujo**, indicar a la tarjeta que utilizará el puerto de entrada analógica 3 del dispositivo 1 (Disp1/EA3) para adquirir datos.
- **Sensor 5. Presión Cámara Compresión**, indicar a la tarjeta que utilizará el puerto de entrada analógica 0 del dispositivo 2 (Disp2/EA0) para adquirir datos.
- **Sensor 6. Presión Boca de Entrada**, indicar a la tarjeta que utilizará el puerto de entrada analógica 1 del dispositivo 2 (Disp2/EA1) para adquirir datos.

6.- Si el botón “Siguiente” es activado, configurar el DAQS para adquirir el nivel de voltaje existente en el puerto de entrada seleccionado, en un rango de 0 a 10 volts, y mostrarlo en la columna de datos “Voltaje (V)”. Colocando cada valor leído debajo del anterior. Si el botón no es activado no ejecutar ninguna acción.

7.- Si el botón “Genera Gráfica” es activado, tomar los valores tanto de la columna “Amplitud” como de la columna “Voltaje (V)” y mostrar la distribución de puntos voltaje contra amplitud en la gráfica. Si no es activado el botón no se ejecuta ninguna acción.

Nota: Antes de activar este botón el usuario debe asentar todos los valores de referencia suficientes para la caracterización y que correspondan a la cantidad de valores adquiridos por el DAQS. Los valores de referencia pueden ser definidos antes de adquirir los valores o en forma simultanea.

8.- Si el botón “Regresión Lineal” es activado, analizar la distribución de puntos y obtener el modelo matemático que mejor se adecúe a dicha distribución, a través de una regresión lineal, para después mostrar en la gráfica la función obtenida mediante una línea en color rojo. Si el botón no es activado no ejecutar ninguna acción.

9.- Si el botón “Obtener Coeficientes” es activado, extraer el conjunto de coeficientes que conforman el modelo matemático, los cuales también son mostrados en los campos: “m”, “b”, “r2”. Si el botón no es activado, no se debe ejecutar ninguna acción.

Nota: El usuario debe obtener primero la regresión lineal, de lo contrario solo obtener valores igual a cero.


10.- Si el botón “Limpiar” es activado, todos los datos obtenidos y mostrados en pantalla, son igualados a cero, con excepción de los valores especificados en la columna “Amplitud”. Si no es activado el botón, no se ejecuta ninguna acción.

11.- Si el botón “Aceptar” es activado, los coeficientes obtenidos son almacenados en memoria, así como las columnas de “Voltaje” y “Amplitud”, conservando la relación con cada sensor. Si no es activado el botón, no se ejecuta ninguna acción.

12.- Si el botón “Export” es activado, generar 2 archivos alfanuméricos sin extensión, con todos los datos almacenados en memoria. Estos archivos son guardados en la ruta especificada en el campo “Guarda Archivo en:” y sus nombres conllevan la fecha y hora de la caracterización, diferenciando un archivo de otro por medio del identificador “Calib” y “Tablas”. El archivo “Tablas” contiene todas las tablas introducidas para caracterización. El archivo “Calib” contiene los conjuntos de coeficientes de cada regresión. Si no es activado el botón, no se ejecuta ninguna acción.

13.- Si el botón “Stop” es activado, finalizar la aplicación, si no, volver a ejecutar los comandos desde el punto 5.

Modo Experimental

- 1.- Comenzar la aplicación activando el botón “Run” .
- 2.- Obtener la fecha y hora instantánea de la computadora.
- 3.- Inicializar con cero todas las gráficas de monitoreo y las variables de procesamiento a utilizar.
- 4.- Mostrar la fecha y hora en los campos correspondientes, en formato: *Lunes, 28 de julio de 2010 y 3:00pm.*
- 5.- Configurar todos los DAQS para adquirir, en todos sus puertos de entrada analógica, niveles de voltaje en un rango de 0 a 10 volts.
- 6.- Adquirir los niveles de voltaje presentes en los puertos de entrada analógica de manera simultánea, con un tiempo de muestreo de 10ms. Este tiempo también sirve para determinar el tiempo de guardado en memoria.
- 7.- Dado que la información es recolectada por la computadora como un arreglo ordenado de datos, descomponer este arreglo para cada entrada analógica (EA0...EA3).
- 8.- Aplicar las operaciones matemáticas de la función lineal ($F[x] = m \cdot (x) + b$) y sus coeficientes (m y b), a cada nivel de voltaje adquirido, a fin de obtener las amplitudes (altura y presión) en función de dichos voltajes:

$$\text{Amplitud [Volts]} = m * (\text{Volts}) + b \dots\dots\dots (1.16)$$

- 9.- Mostrar en las gráficas de monitoreo las amplitudes obtenidas (Altura Resonante, Altura Desagüe, Presión Boca de Entrada, etc.) en forma de barrido.
- 10.- Si el botón “LOAD” es activado, extraer del archivo “Calib”, cuya ruta de ubicación se encuentra especificada en el apartado “Archivo Calib”, los coeficientes de la caracterización e ingresarlos al procesamiento interno. Si el botón no es activado no ejecutar ninguna acción.

Nota: Si no existe ninguna ruta especificada en el apartado “Archivo Calib”, abrir una ventana de búsqueda para que el usuario pueda especificar la ubicación del archivo.

- 11.- Si el botón “Guardar en memoria” es activado, almacenar en memoria los niveles de voltaje adquiridos (valores brutos) y las amplitudes obtenidas (valores físicos) que en ese momento se encuentren procesando. Esta función permanece activa por el lapso de tiempo definido en “Tiempo de Guardado”. Si el botón no es activado, no ejecutar ninguna acción.

Nota: Lo anterior se logra analizando la relación entre el periodo de muestreo y el tiempo que se requiere mantener activa la función.

Si:

$$\text{Periodo de Muestreo} = 0.01 [\text{seg/muestra}] \dots\dots\dots (1.17)$$

Y por ejemplo:

$$\text{Tiempo de Guardado} = 30 [\text{seg}] \dots\dots\dots (1.18)$$

La cantidad de muestras que debemos **CONTAR** es:

$$\text{No. de Muestras} = \text{Tiempo de Guardado} / \text{Periodo de Muestreo} \dots\dots (1.19)$$

$$\text{No. de Muestras} = 30 [\text{seg}] / 0.01 [\text{seg/muestra}]$$

$$\text{No. de Muestras} = 3000$$

Por lo tanto el sistema debe contar una cantidad de 3000 muestras para asegurar un tiempo de guardado de 30 [seg].

12.- Si el botón “Export” es activado, generar 2 archivos alfanuméricos sin extensión, con todos los datos y tablas en memoria. Estos archivos son guardados en la misma carpeta de la cual se extrajo el archivo “Calib” y sus nombres se diferencian uno de otro por el indicador “Exp_Voltajes” y “Exp_DatosFisicos”. El archivo “Exp_Voltajes” contiene una tabla con los niveles de voltaje almacenados en memoria. El archivo “Exp_DatosFisicos” contiene una tabla con los valores de amplitud almacenados en memoria. Ambos archivos contienen en su encabezado fecha, hora y no. de experimento, así como la ruta del archivo “Calib”, nivel de mare, amplitud, periodo, etc. Si no es activado el botón, no se ejecuta ninguna acción.

Nota: Si los datos de: marea, amplitud, colector, etc., no se encuentran especificados antes de general dichos archivos, asignar el valor de “Null” a cada uno de ellos.

13.- Si el botón “Stop” es activado, finalizar la aplicación, si no, volver a ejecutar los comandos desde el punto 5.

2.6 Diagramas de flujo

Con los algoritmos de programación ya definidos, es posible elaborar los diagramas de flujo que sirven bien en todo proceso de desarrollo de software, para analizar el flujo de los datos a través de cada acción e identificar los riesgos y posibles errores presentarse al momento de realizar el código de programación.

A continuación se presenta el diagrama de flujo de la interfaz de caracterización:

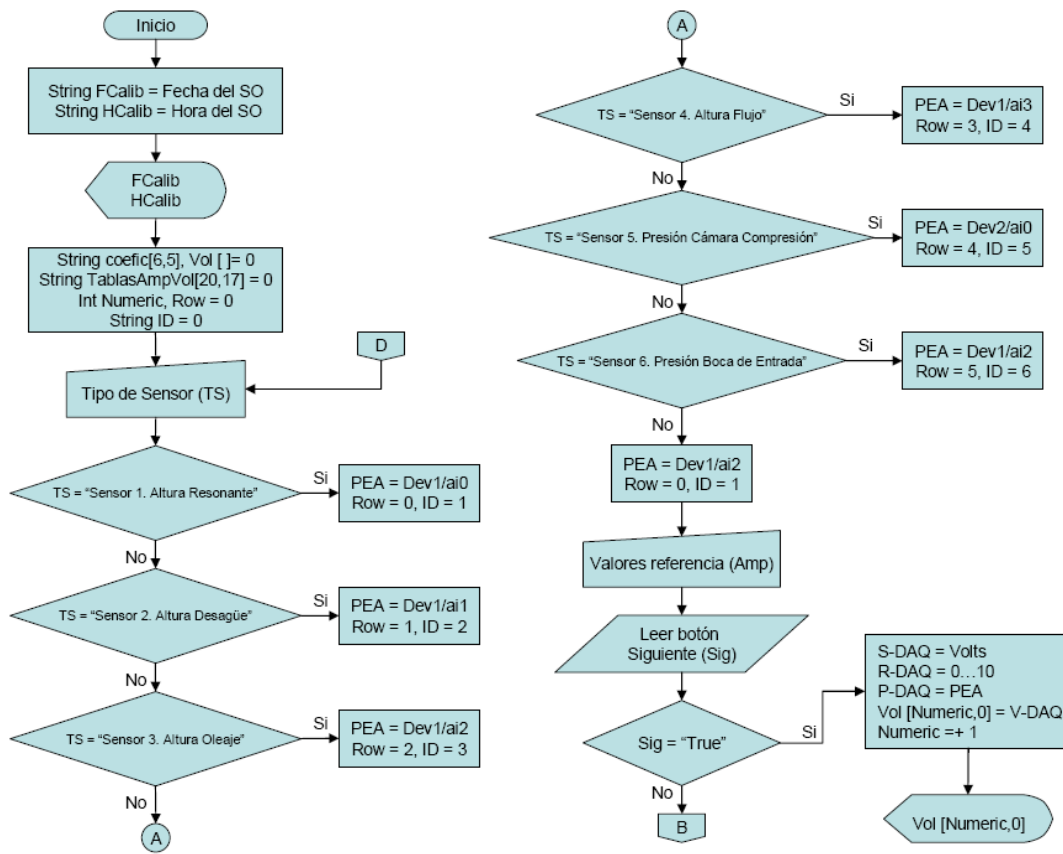


Figura 24. Diagrama de flujo de la interfaz de caracterización – Parte 1

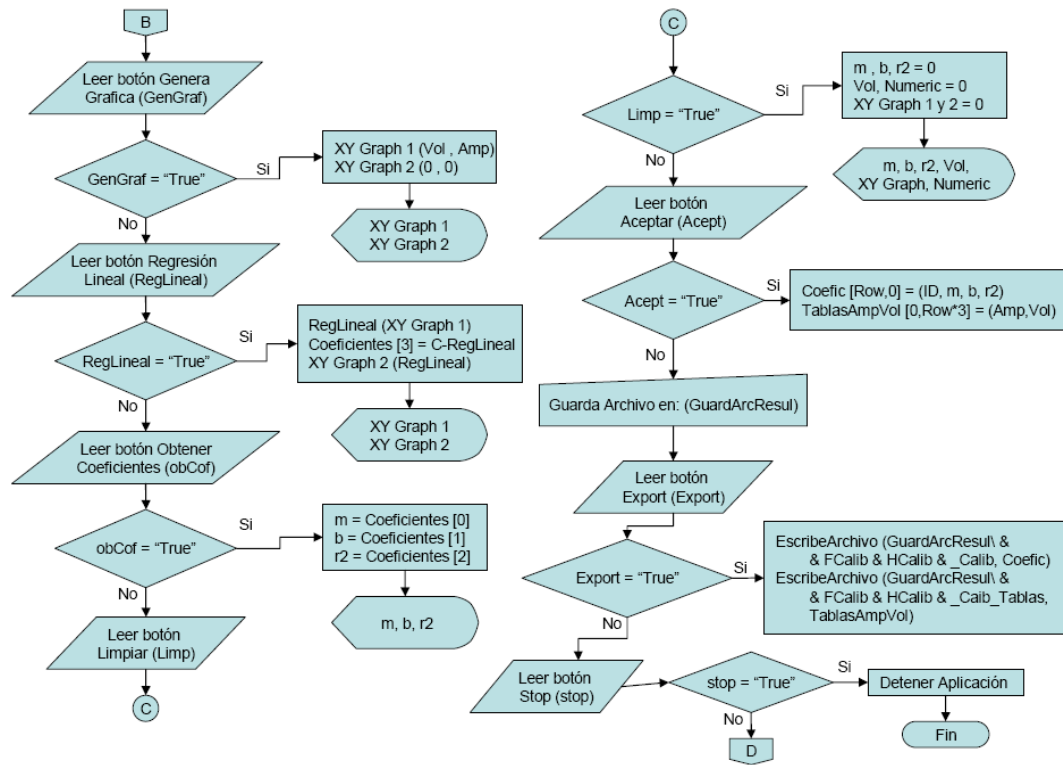


Figura 25. Diagrama de flujo de la interfaz de caracterización – Parte 2

A continuación el diagrama de flujo de la interfaz de experimentación:

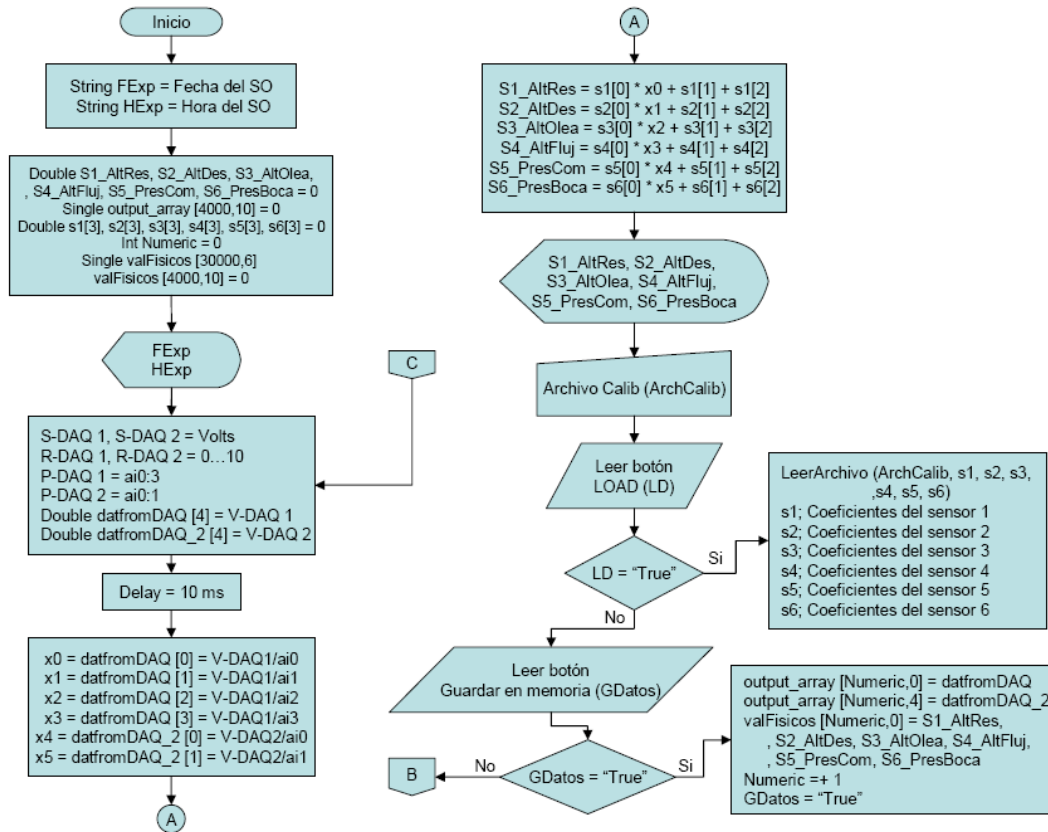


Figura 26. Diagrama de flujo de la interfaz experimental – Parte 1

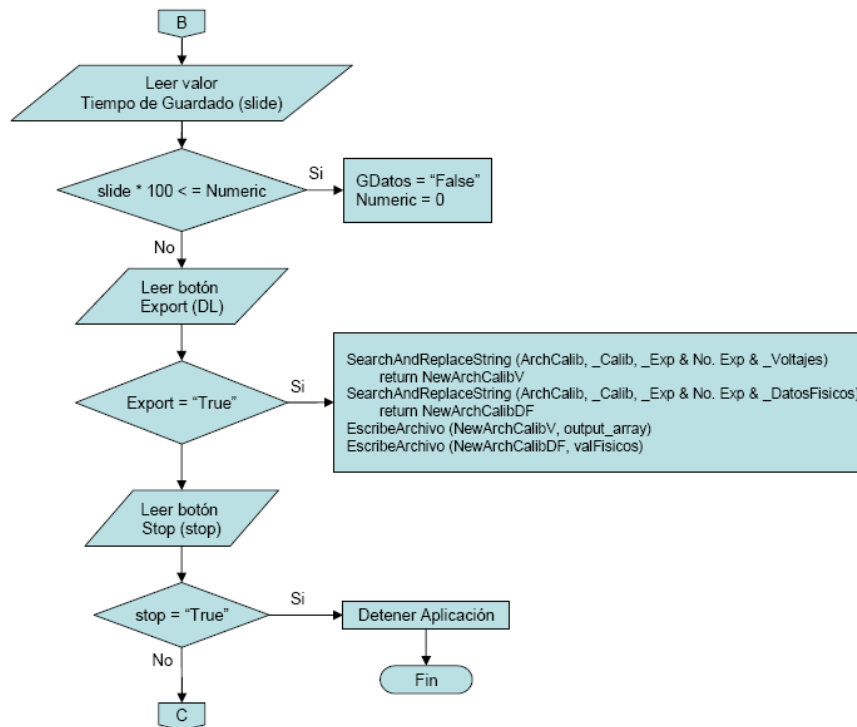


Figura 27. Diagrama de flujo de la interfaz experimental – Parte 2

3. Desarrollo de interfaces

3.1 Funcionamiento de comandos básicos

3.1.1 Adquisición de datos externos

Para entender el proceso de adquisición de valores a través de las tarjetas de adquisición de datos, en el panel frontal se crea el siguiente PF.

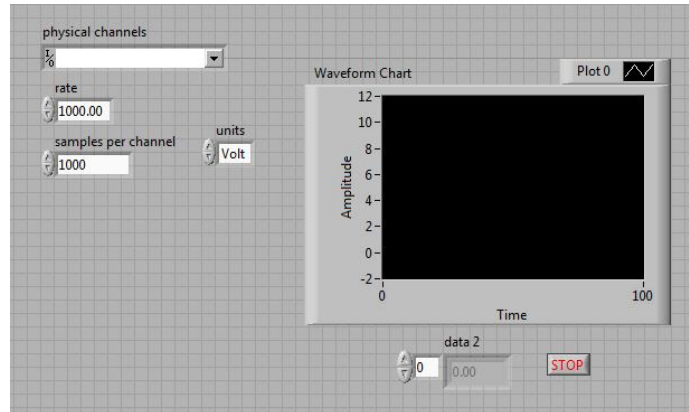
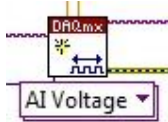


Figura 28. Interfaz para adquirir datos del DAQS

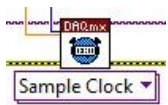
A continuación las imágenes de programación que se crearán en el DB:



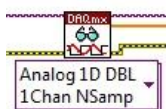
Las siguientes funciones de LabVIEW habilitan, leen y cierran los canales de entrada de los DAQS:



Habilitar y configurar un canal del DAQS. Sus parámetros de entrada principales son: tipo de señal a medir, canal físico de lectura, nivel máximo y mínimo de medida; sus parámetros de salida principales son: canal físico habilitado y configurado, validación de error.



Establecer velocidad de muestreo. Con parámetros de entrada principales: canal físico de lectura, validación de error, número de estimación y número de muestras por canal; sus parámetros de salida son: canal físico acotado y validación de error.



Leer dato. Parámetros de entrada: canal físico de lectura, validación de error, dimensión del arreglo y punto flotante de los dato; parámetros: dato leído, validación de error, canal físico de lectura.



Borra/cerrar canal. Sus parámetros de entrada principales son: canal físico a borrar y validación de error.

Para repetir el proceso de manera cíclica es necesario contener todos los elementos en una estructura *while*, la cual repetirá el proceso hasta que el botón “Stop” sea activado. Concluido lo anterior, se alambran los elementos entre si como se muestra en la siguiente figura.

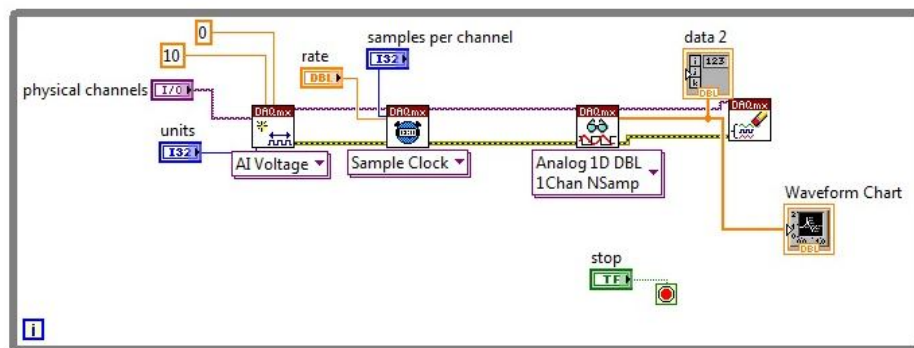


Figura 29. Diagrama de bloque para adquirir datos del DAQS

En primera instancia el canal físico, el tipo de señal a monitorear y el número de muestreos son definidos por el usuario; el nivel máximo y mínimo de dicha señal (diez y cero correspondientemente) son definidos el personal de mantenimiento.

En el nodo siguiente los datos obtenidos son mostrados tanto en el indicador “data2” como en la gráfica “Waveform Chart”, este elemento maneja datos de tipo *double*.

Finalmente se limpia el canal y se verifica que el proceso no contenga errores. Por último y con los DAQS preparados es posible asignar valores aleatorios en cada control y observar la respuesta en la frecuencia de muestreo y adquisición de datos.

3.2 Programación de la interfaz “Modo Calibración”

En esta sección se describe la programación de la interfaz “Modo Calibración” en base a los algoritmos de programación y diagramas de flujo antes mencionados, así como empleando el diseño gráfico mostrado en el apartado 2.4 de este trabajo.

Se comienza obteniendo el tiempo instantáneo de la computadora mediante el comando “Get Date/Time In Seconds” y a través del comando “Gate Date/Time String” se generan los datos tanto de fecha como de hora de manera independiente. Para especificar el formato de visualización (*Friday, January 21, 1994*) se utiliza el identificador *long*. Estos datos son unidos a los indicadores de tipo *string* “FCalib” y “HCalib” correspondientemente, mostrando así la información en pantalla.

Después se declaran e inicializan los elementos de almacenamiento temporal “Coefic” como un arreglo 6x5 de tipo *string* y un valor inicial igual a *null*; “TablasAmpVol” como un arreglo 20x17 de tipo *string* y con un valor inicial igual a *null*; “Numeric” como un indicador numérico de tipo “Int” y con un valor inicial igual a cero. Posteriormente, se encierran estos elementos en una estructura secuencial “Flat Sequence Structure” como se muestra en la figura 30.

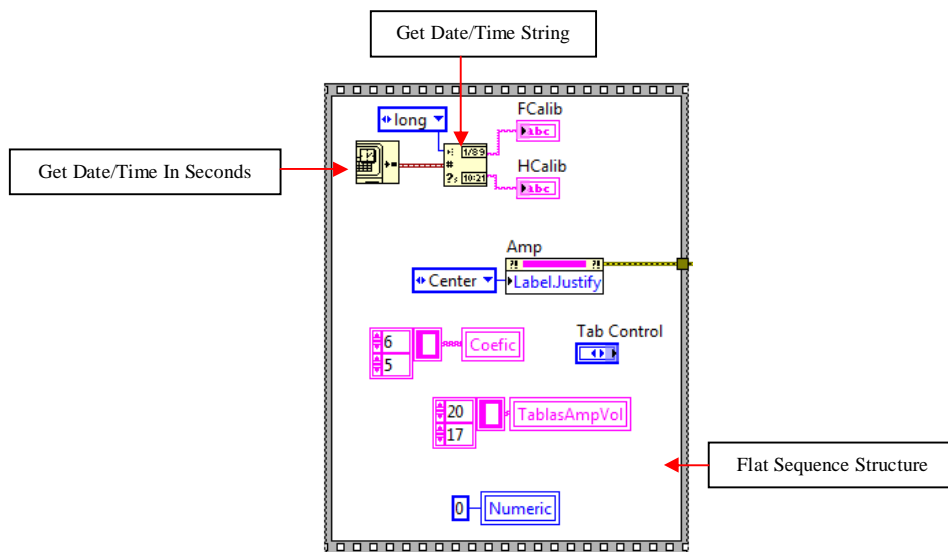


Figura 30. Diagrama de bloque que declara e inicializa los elementos a utilizar

El elemento “Amp” es un nodo de propiedad y declara la justificación del texto en la columna del elemento “Amplitud”, su salida de error proporciona continuidad de flujo de programación.

Empleando una *case structure* y en base al estado de “TS” (Tipo de Sensor) se asignan valores de “Row” e “ID” al canal de lectura como se muestra en el diagrama de bloque de la figura 31.



Figura 31. *Case structure* con valores a asignar en base al estado de “TS”

Empleando una estructura *case* se configuran los DAQS de acuerdo al rango permitido de señales de entrada, al canal físico seleccionado (figura 31) y al estado del elemento “Sig” (botón “Siguiente” en el PF). Una vez adquirido el valor se construye un arreglo de dimensión 1, el cual se inserta en un arreglo de dos dimensiones donde el renglón de inserción lo determina la variable “Numeric”. El arreglo obtenido se convierte de tipo *double* a tipo *string* mediante el comando “Number To Fractional String” y es mostrado en pantalla a través del elemento “Vol”. Se incrementa el valor de “Numeric” en 1. En el caso de que el estado de “Sig” sea *false*, únicamente se procesara el valor del arreglo de dos dimensiones. La siguiente figura muestra el diagrama de bloque de la programación descrita.

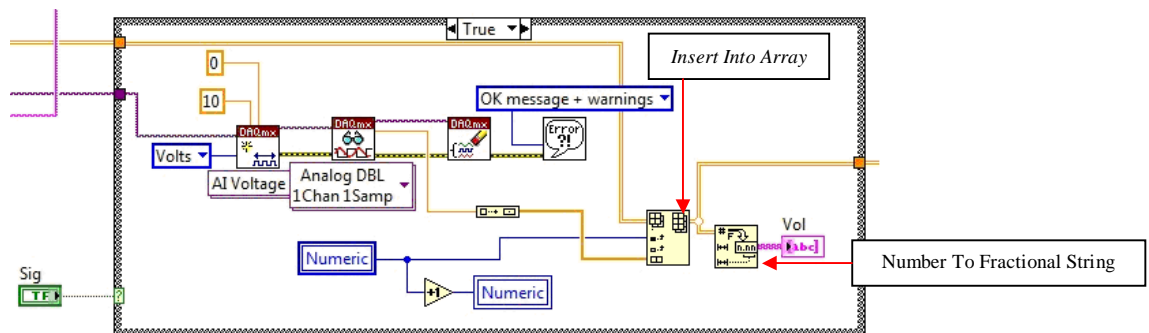


Figura 32. *Case Structure* del elemento “Sig” para configurar, adquirir y mostrar en pantalla los valores del DAQS

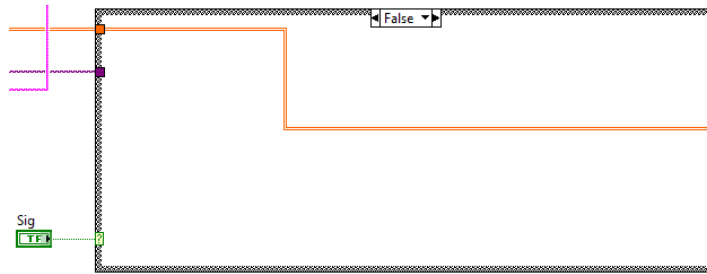


Figura 33. Case Structure en modo False

Una vez adquiridos los valores en los DAQS, el programa revisa el estado del elemento “GenGraf” mediante una estructura *Case*, si su estado es *true* se extrae la columna 0 de los elementos “Vol” y “Amp”. Dichas columnas se convierten de tipo *string* a tipo *Double* con la función “Fract/Exp String to Number” y se unen a través de un “Bundle” para formar un *cluster* de dos elementos que, junto a un *cluster* con valores igual a cero, se construye una matriz de dimensión 1, la cual es alambrada al elemento “XY Graph”. De esta manera se mostrara en pantalla la gráfica amplitud contra voltaje. En el caso de que el estado de “GenGraf” sea *false*, no se ejecuta ninguna acción. El diagrama resultante aparece en la figura 34.

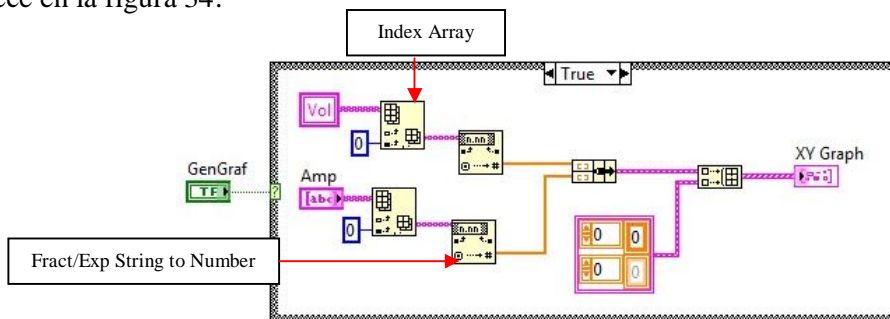


Figura 34. Diagrama de bloque para generar la gráfica XY a partir de los valores de amplitud contra voltaje

Con ayuda de una estructura *case* se obtiene la regresión lineal de la gráfica si el estado del elemento “RegLineal” es *true*. A través del nodo de propiedad “XY Graph – Value” se obtiene un arreglo de todos sus valores presentes. Con la función “Index Array” se extrae el conjunto de valores que corresponden únicamente a la gráfica. La función “Linear Fit” emplea tanto los valores del eje X como del eje Y para generar la regresión correspondiente. Los coeficientes que resultan, se almacenan en el elemento “Coeficientes”.

Para mostrar en pantalla la gráfica de la regresión obtenida y la gráfica de amplitud contra voltaje simultáneamente, se construye un arreglo de una dimensión con dos *bundles*, cada uno con dos arreglos para los conjuntos de valores X y Y. En el caso de que el estado del elemento “RegLineal” sea *false* no se ejecuta ninguna acción. En la figura 35 se muestra el diagrama descrito.

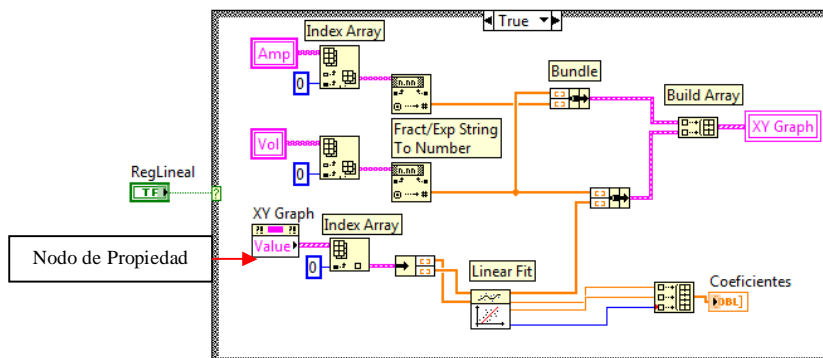


Figura 35. Diagrama de bloque para obtener la regresión lineal de la gráfica XY y mostrarla en pantalla

El proceso para mostrar los coeficientes de la regresión, se ejecuta al momento en que el estado del elemento “obCof” es *true* y es recibido en una estructura *case*. Inicia con la extracción de los valores contenidos en el elemento “Coeficientes” considerando la siguiente agrupación:

- Índice 0: pendiente
- Índice 1: ordenada al origen
- Índice 2: error

Estos valores son convertidos de tipo *double* a tipo *string* y mostrados en pantalla a través de los indicadores “m”, “b” y “r2”. En el caso de que el estado de “obCof” sea *false* no se ejecuta ninguna acción. Lo anterior se presenta en la siguiente figura.

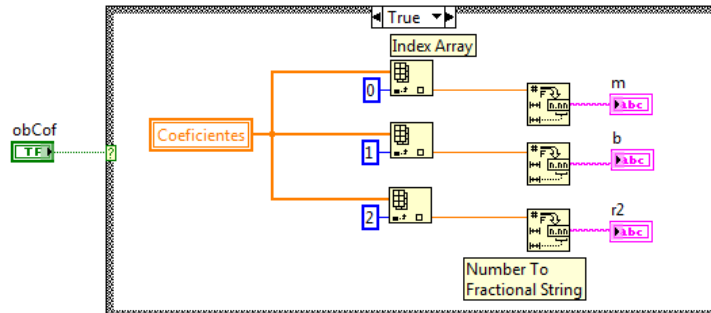


Figura 36. Diagrama de bloque para mostrar en pantalla los coeficientes de la regresión lineal

Si el botón “Limpiar” es activado el estado del elemento “Limp” cambia a *true*. Ante esto el programa asigna constantes igual a 0 a cada nodo de propiedad *value* de los elementos “Vol”, “m”, “b”, “r2” y “XY Graph”, con excepción del elemento “Numeric”, ya que en este emplea su variable local. La figura 37 muestra el diagrama de bloque de la estructura *case* que contiene las funciones mencionadas.

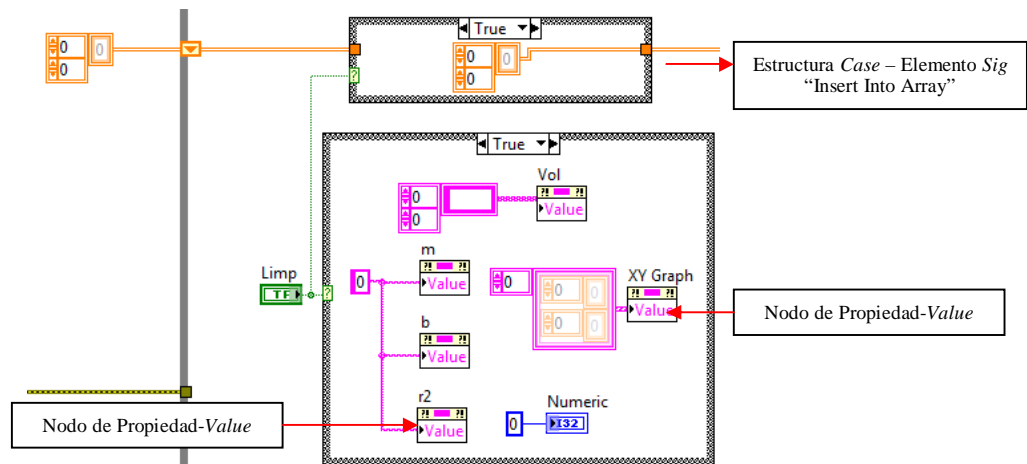


Figura 37. Diagrama de bloque que asigna el valor de 0 a cada variable

Ahora bien, si el botón “Aceptar” es activado el estado del elemento “Acept” cambia a *true* activando la condición correspondiente de una estructura *case*. En esta se construye un arreglo con el valor de los elementos “ID”, “m”, “b”, “r2” y el resultado es ingresado en la función “Replace Array Subset” la cual tiene como base el elemento “Coefic”. La matriz es ingresada con el índice determinado por el valor “Row”, de tal forma que el arreglo resultado sea un arreglo de valores enlistados según el número de sensor caracterizado, el resultado es asignado al elemento “Coefic”.

Mediante la función “Index Array” se obtienen los valores de los elementos “Amp” y “Vol” empleando el índice 0. Los valores de “Amp” son los primeros en ser procesados e insertados en el elemento “TablasAmpVol”, el resultado se ingresa a una función “Replace Array Subset” como arreglo base. Los valores del elemento “Vol” son insertados en este nuevo arreglo en la ubicación $(0, Row \times 3 + 1)$, el arreglo resultado es almacenado en el elemento “TablasAmpVol”, con esto se asegura que tal elemento contiene todos y cada uno de los valores de interés. Las formulas de ubicación forman tablas de datos ordenadas y separadas por una columna entre si. En el caso de que el estado del elemento “Accept” sea *false*, no se ejecuta ninguna acción y el campo permanece vacío. La siguiente figura refleja la programación mencionada.

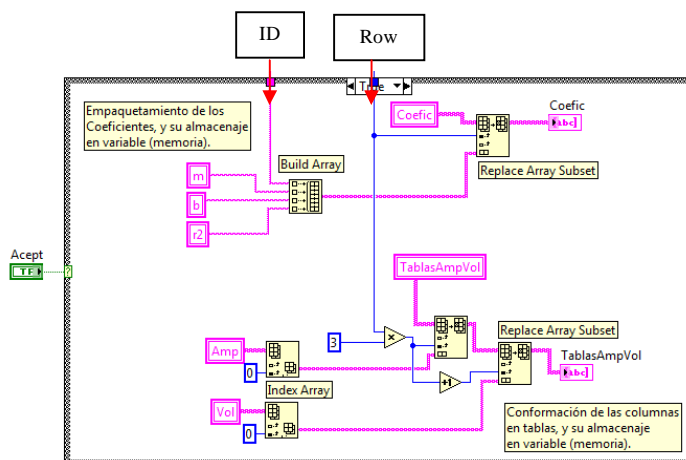


Figura 38. Diagrama de bloque para formar los arreglos de datos con los valores de interés

Para generar los archivos con los valores adquiridos, el programa espera que el estado del elemento “Export” sea *trae*, si esta condición se cumple se emplean variables de almacenamiento temporal para obtener el valor de “FCalib” y “HCalib”. A través de funciones “Search and Replace String” se reemplazan los caracteres que no pueden ser procesados por Windows: /, :, con la intención de acondicionar dichos valores. En el caso de “FCalib” se realizan los cambios en formato de tal manera que 99/99/99 sea 99-99-99, así como en el caso de “HCalib” donde se modifica 99:99pm. por 99-99pm. Del elemento “GuardArcResul” se obtiene la ruta de almacenamiento, y empleando la función “Path To String”, se convierte a tipo *string*. Estos datos se unen de manera ordenada a través de la función “Concatenate String” para obtener como resultado la ruta completa de almacenamiento, siguiendo la siguiente estructura:

“GuardArcResul”\”FCalib”_”HCalib”_Calib

Se forma un arreglo con la función “Build Array” con las etiquetas: **Sens**, **m**, **b**, **r2**, este servirá como encabezado en el archivo a formar. El arreglo es introducido en la matriz “Coefic” en el renglón 0. El arreglo que resulta es ingresado en la función “Write To Spreadsheet File”. De igual forma se ingresa la ruta de tipo *path* previamente formada como se muestra en la figura 39.

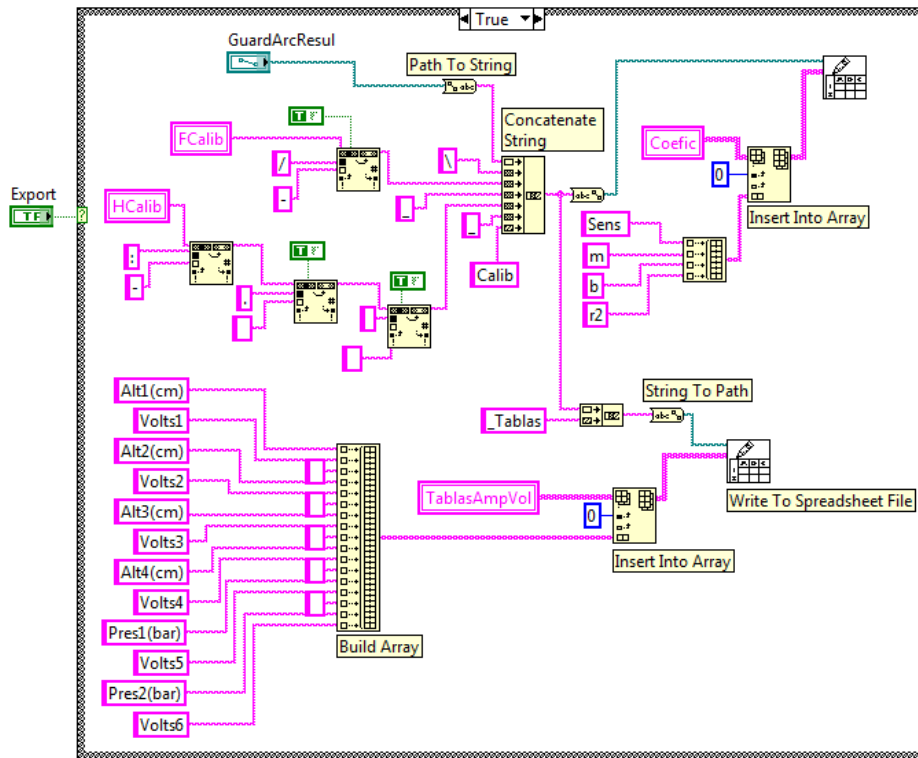


Figura 39. Diagrama de bloque que genera los archivos con los valores obtenidos y procesados.

Dentro de esta estructura se forma un arreglo ordenado con las etiquetas: “Alt1(cm), Volts1, Alt2(cm), Volts2, Alt3(cm), Volts3, Alt4(cm), Volts4, Alt5(cm), Volts5, Alt6(cm), Volts6”, estas formaran parte del encabezado para las tablas de valores. El arreglo que resulta es almacenado en el elemento “TablasAmpVol” a través de la función “Insert Into Array” en el índice 0. Dicho arreglo es la base para formar el archivo resultado, mediante la función “Write To Spreadsheet File”. La ruta de almacenamiento es concatenada con la etiqueta “_Tablas” y convertida de tipo *string* a tipo *path* y es ingresada en la función “Write To Spreadsheet File”. Si el valor de “Export” es *false*, no se ejecuta ninguna acción y el apartado correspondiente permanece en vacío.

Toda sintaxis de programación hasta ahora descrita es contenida en ciclo *while*, el cual por defecto trae consigo un botón de paro (elemento “stop”), lo que brinda un análisis cíclico de cada función y elemento el caso en que el valor del elemento “stop” sea *true*.

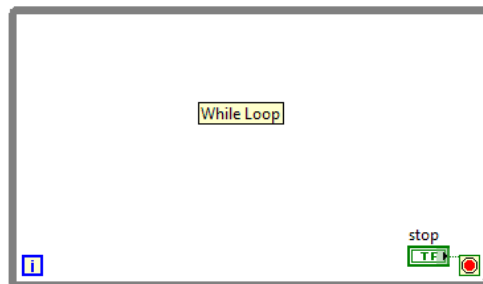


Figura 40. Ciclo while en del DB

Los indicadores gráficos correspondientes a las variables de almacenamiento temporal: “Coeficientes”, “Coefic” y “Numeric” aparecen en el PF fuera del marco azul, con el fin de no confundir al usuario al momento de realizar la operación, lo anterior se ilustra en la figura 41.

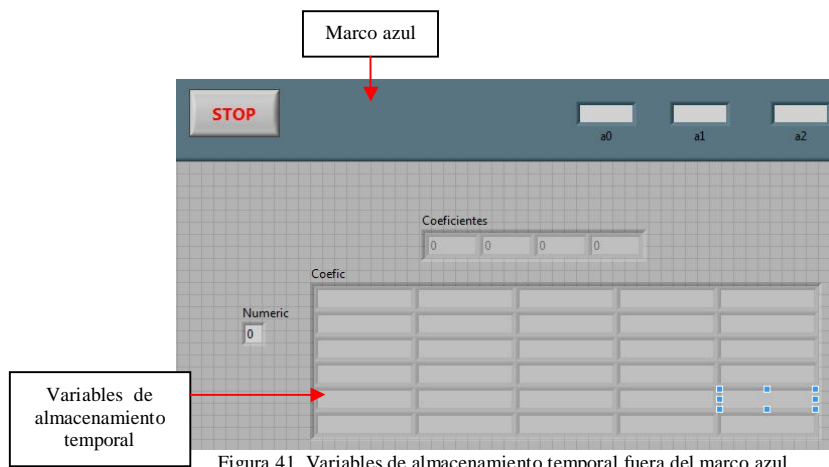


Figura 41. Variables de almacenamiento temporal fuera del marco azul.

3.2.1 Ajuste de requerimientos y reprogramación

Concluida la interfaz Modo Calibración se debe analizar a detalle el proceso para obtener la regresión lineal.

En la respuesta de los sensores construidos en el SIBEO de manera convencional, se observa un estado de saturación natural. Esta se debe a que su construcción está basada en dispositivos electrónicos básicos y de manufactura sencilla como capacitores, resistencias, etc. Estos no están diseñados para un alto rendimiento y precisión a través del tiempo. Debido a este problema los modelos matemáticos presentarán un gran porcentaje de error e incertidumbre.

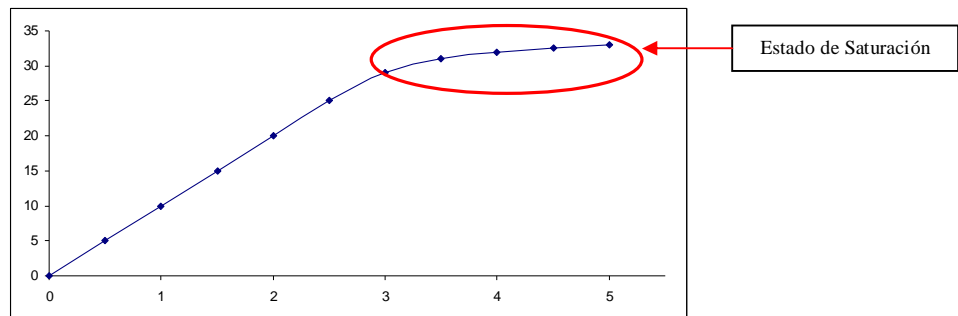


Figura 42. Respuesta de un sensor con estado de saturación.

Para solventar esta situación se propone obtener una regresión polinomial de 3er. grado en lugar de una regresión lineal. Dicha regresión será capaz de adecuarse mejor a respuestas lineales y no lineales. Es necesario entonces realizar las modificaciones correspondientes en la programación de la interfaz y obtener así el desempeño deseado.

En la estructura *case* del elemento “RegLineal” se cambia la función “Linear Fit” por la función “General Polynomial Fit” con un orden 3. Esta función retorna los coeficientes de la función polinomial en el orden: a0, a1, a2,... con dimensión igual a 1. Este arreglo es contenido en la variable “Coeficientes”, el diagrama de bloque se muestra en la figura 43.

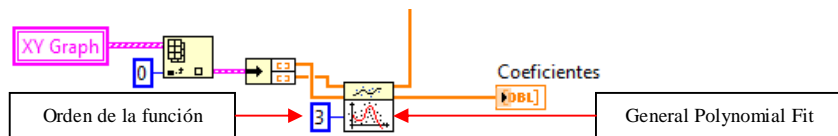


Figura 43. Adaptación de la regresión polinomial para solventar el estado de saturación

En la estructura *case* del elemento “obCof”, los datos convertidos por la función “Number to Fractional String” son mostrados en los elementos “a0”, “a1”, “a2” y “a3”, los cuales reemplazan los indicadores “m”, “b” y “r2” en el PF.

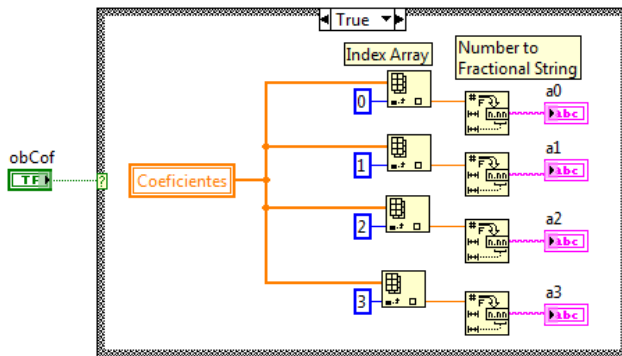


Figura 44. Estructura case de “obCof” modificada

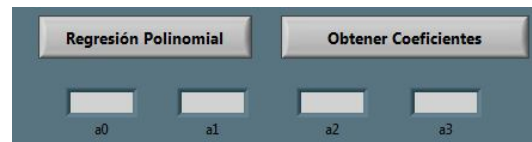


Figura 45. Indicadores modificados en el PF

Así mismo en la estructura del elemento “Acept”, se construye un arreglo con las variables locales de “a0”, “a1”, “a2” y “a3”, como se muestra a continuación.

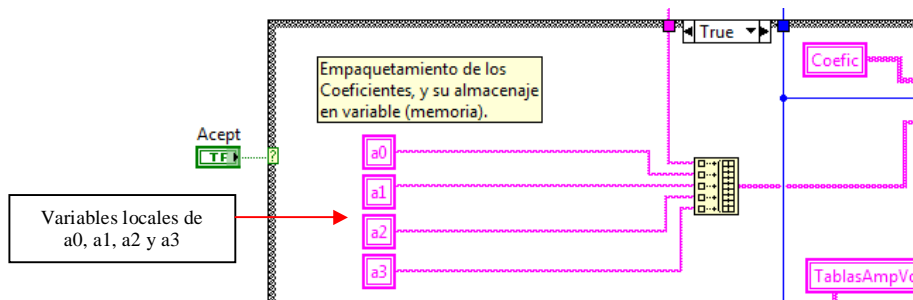


Figura 46. Diagrama de bloque de la estructura case del elemento “Acept” modificada

Estas modificaciones también repercuten en la estructura *case* del elemento “Export”, ya que deben ser sustituidas las etiquetas de los encabezados del arreglo “Coefic”.

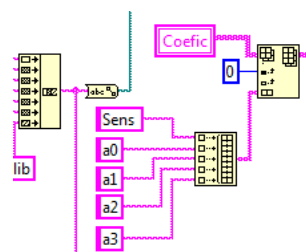


Figura 47. Diagrama de bloque de los nuevos encabezados del arreglo “Coefic”

3.3 Programación de la interfaz “Modo Experimental”

A continuación se presenta la programación de la interfaz “Modo Experimental” en base a los algoritmos de programación y empleando el diseño gráfico del PF definido en el apartado 2.5 de este trabajo.

En primera instancia, los nodos de propiedad “History” de los elementos “S1_AltRes”, “S2_AltDes”, “S3_AltOlea”, “S4_AltFluj”, “S5_PresCom” y “S6_PresBoca” se inicializan en cero.

Se adquiere la hora y fecha del sistema operativo con la función “Get Date/Time In Seconds” y a través de la función “Get Date Time String” se decodifica y se muestra en los indicadores “FExp” y “HExp” la información correspondiente. Se establece el rango de los arreglos “output_array”, “output_array 2” y “valFisicos” en 4000,10 y se inicializan en cero, así como los arreglos de dimensión 1: “s1”, “s2”, “s3”, “s4”, ”s5”, “s6” y “Numeric”. Estos objetos se agrupan en una estructura secuencial para asegurar el correcto flujo de procesamiento. En la figura siguiente se presenta el DB mencionado.

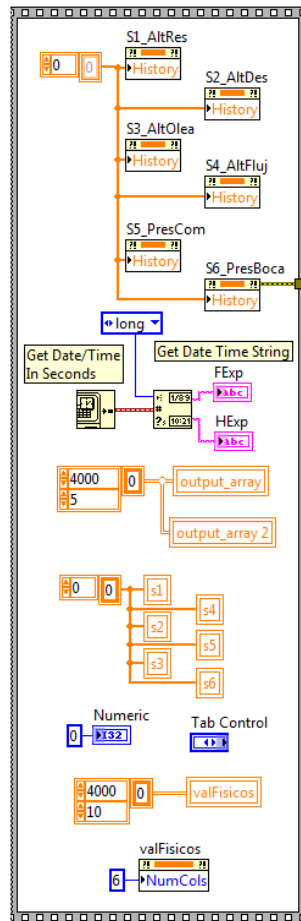


Figura 48. Diagrama de bloque que inicializa los elementos a utilizar

Se crea un ciclo *while* y dentro de este se crean el siguiente programa.

A través de la función “Create Virtual Channel” se configuran los DAQS para adquirir señales analógicas de voltaje, habilitar los puertos de entrada de “ai0” a “ai3” del dispositivo 1 y de “ai0” a ai1” del dispositivo 2, así como establecer rangos de lectura de 0 a 10 *volts*. Con la función “Read” se adquieren los niveles de señal de cada DAQS y se almacenan en el elemento “atfromDAQ” y “datfromDAQ 2”, estos son arreglos de dimensión 1 y de tipo *single*. Con la función “Clear Task” se limpian los canales. La figura 49 y 50 muestran el programa descrito.

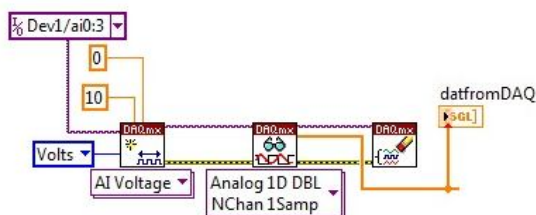


Figura 49. Diagrama de bloque para adquirir voltajes del dispositivo 1

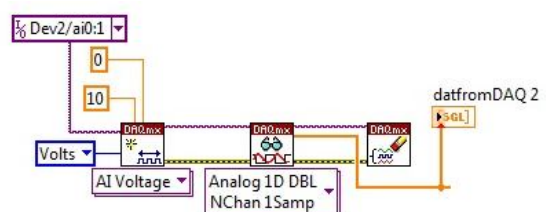


Figura 50. Diagrama de bloque para adquirir voltajes del dispositivo 2

Empleando la función “Wait Until Next ms Multiple” se establece un tiempo de espera de 10ms para todos los procesos dentro del ciclo *while*, con el cual indirectamente también se define el periodo de muestreo de cada DAQS.

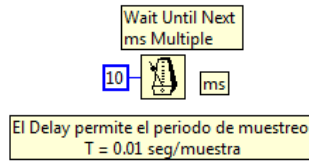


Figura 51. Definición del tiempo de espera aplicable a todo el ciclo *while*

Con los valores de voltaje contenidos en “datfromDAQ” y “datfromDAQ_2” y a través de la función “Index Array”, se extrae cada valor según el índice de referencia. Se emplea la misma función para extraer de las variables locales “s1”, “s2”, “s3”, y “s4” las constantes del modelo matemático: índice de referencia 0 para el valor de “a0”, índice de referencia 1 para el valor de “a1” y así sucesivamente para el valor “a3”. Aplicando la función “Compound Arithmetic” en tipo de suma y multiplicación, se calcula el modelo matemático según la forma del polinomio de 3 grado en función de los valores de voltaje. De esta manera se obtienen las variables críticas de interés: altura resonante, altura de desagüe, etc. La figura 52 muestra los elementos “S1_AltRes”, “S2_AltDes”, “S3_AltOlea” y “S4_AltFluj”, los cuales presentan en el PF los valores procesados.

De igual forma se descompone el elemento “datfromDAQ_2” y junto a los elementos “s4” y “s5”, se determinan las variables de: presión en el compresor y presión en la boca de desagüe. La figura 53 ilustra el diagrama de bloque de este proceso.

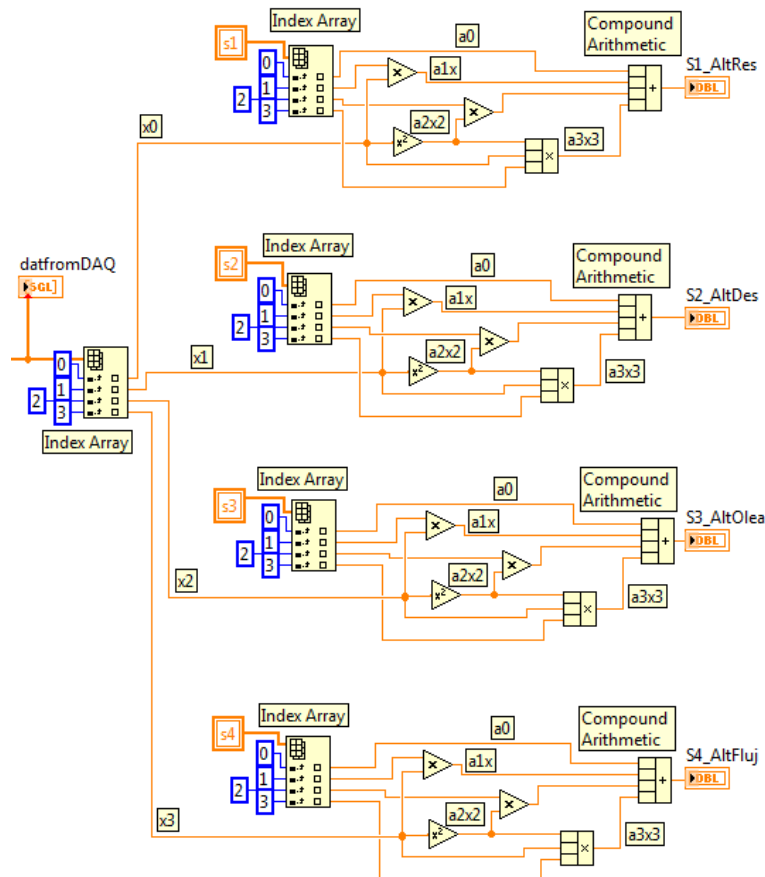


Figura 52. Tratamiento matemático de “datfromDAQ” para determinar las variables de interés.

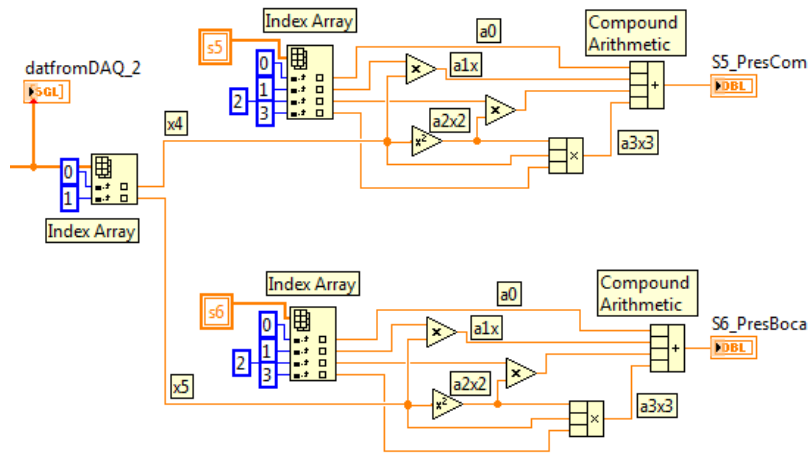


Figura 53. Tratamiento matemático de “datfromDAQ 2” para determinar las variables de interés.

En el momento en que el botón *LOAD* es activado, el elemento “LD” adquiere el valor *true* se extrae la información del archivo “Calib” a través de la función “Read From Spreadsheet File” en formato *string*. Para fines prácticos se eliminan los encabezados del renglón 0 y los indicadores de sensor de la columna 0 a través de la función “Delete From Array”. La función “Fract/Exp String To Number” es utilizada para transformar los datos contenidos a tipo *double*. Con la función “Index Array” se extraen los coeficientes, según el índice de referencia en el cual fueron almacenados. En la siguiente figura se puede apreciar como los coeficientes son almacenados en las variables de almacenamiento temporal “s1”, “s2”, “s3”, “s4”, “s5” y “s6”.

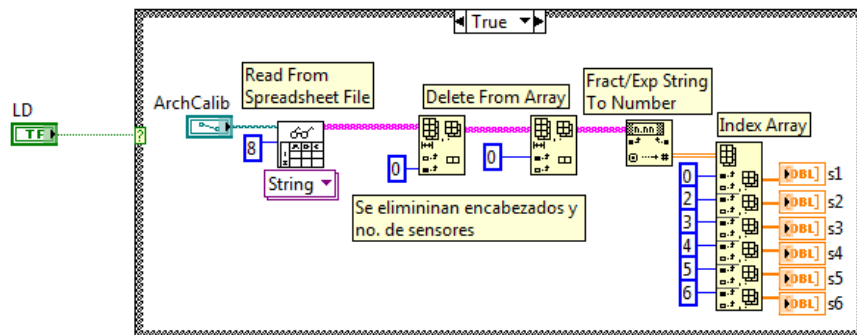


Figura 54. Diagrama de bloque para extraer del archivo “Calib” el conjunto de coeficientes de cada modelo matemático

Para el caso en que el botón “Guardar en Memoria” es activado, el elemento “Gdatos” adquiere el valor de *true*. Al cumplirse esta condición el arreglo “datfromDAQ” es ingresado al elemento “output_array” a través de la función “Replace Array Subset” en el renglón definido por el elemento “Numeric”, el cual incrementa su valor en uno cada vez que se cumple dicha condición; este mismo procedimiento se sigue para incorporar el arreglo “datfromDAQ 2” en el elemento “output_array 2”. Adicional a este proceso se construye un arreglo con los contenidos en los elementos “S1_AltRes”, “S2_AltDes”, “S3_AltOlea”, etc. Dicho arreglo se inserta en la variable de almacenamiento temporal “valFisicos” en el índice de renglón definido por “Numeric”. Para mantener el valor de “Gdatos” con un valor *true*, se asigna este valor booleano a su varia local. La figura 55 que a continuación se presenta, ilustra el proceso mencionado.

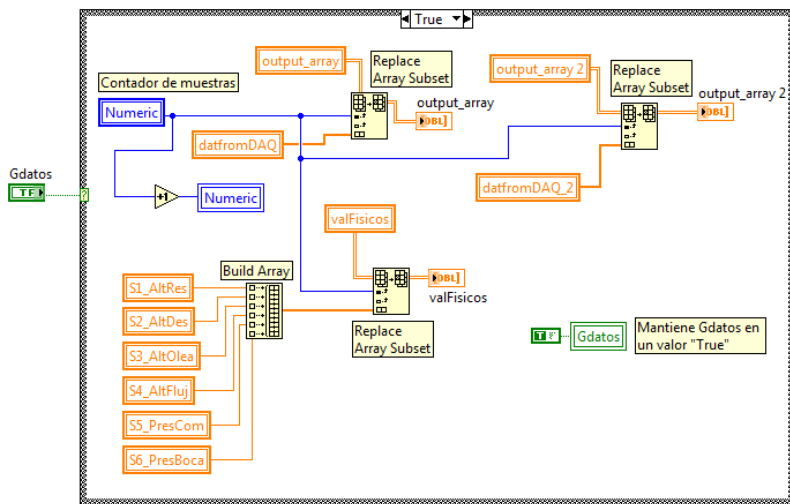


Figura 55. Diagrama de bloque que almacena los valores brutos y físicos en variables locales

El tiempo en segundos que el sistema debe permanecer almacenando datos en memoria esta definido por el elemento “Slide” en el PF. Cuando el valor que resulta multiplicar el número entero de este elemento por 100 sea igual o menor al valor de “Numeric”, se asigna el valor *false* al elemento “Gdatos” lo que provoca que el proceso de almacenamiento en memoria se detenga y se reinicie el contador “Numeric”. Este proceso esta fuera de cualquier otra estructura y esta contenida únicamente en la estructura general *while*.

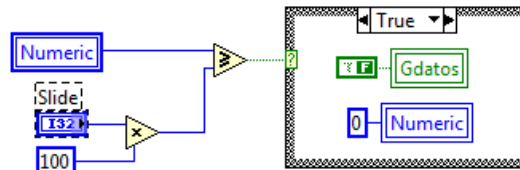


Figura 56. Diagrama de bloque que condiciona el tiempo de almacenado

Para generar el archivo con el registro de los valores brutos, el sistema espera que el botón “Export” sea presionado para asignar el valor de *true* al elemento “DL”. Esto forma un arreglo con las etiquetas: “Fecha”, “Hora”, “No. Exp.”, “T (seg)”, “A (cm)”, “M (cm)”, “C (cm)” y “Archivo Calib”, dicho arreglo se ingresa en el índice de renglón 0 de una matriz, originalmente con un valor igual a *null*. El arreglo que resulta se utiliza como base y en el renglón 1 se insertan los valores de los elementos “FExp”, “HExp”, “No. Exp”, “PerT”, “Amp”, “Mar”, “colec” y “ArchCalib”. El nuevo arreglo se utiliza como base nuevamente para insertar en su renglón número 3 los encabezados: “Dev1/a0”, “Dev1/a1”, “Dev1/a2”, “Dev1/a3”, “Dev2/a0”, “Dev2/a1”. Posteriormente se ingresa en su renglón número 4 la unión de los arreglos “output_array” y “output_array 2”. Con la función “Write To Spreadsheet File” se crea un archivo con el último arreglo obtenido. La ruta de almacenamiento es una modificación de la ruta “ArchCalib”, pues es convertida a tipo *string* y mediante la función “Search and Replace String” se cambian los caracteres “_Calib” por los caracteres “_Exp&No.Exp&_Voltajes”, generando la ruta en la cual se debe almacenar el primer archivo. Lo anterior puede apreciarse en el diagrama de bloque superior de la figura 57.

Para obtener el archivo con los valores procesados o físicos, el “Encabezado 1” es insertado en una matriz de dimensión 2 a través de la función “Insert Into Array” en el índice de renglón 1. En el renglón 1 del arreglo resultado se insertan los valores correspondientes a dicho encabezado. El arreglo que resulta es utilizado como base y en su renglón número 3 se insertan las etiquetas “Alt Res”, “Alt Des”, “Alt Ole”, “Alt Flu”, “PresCamCom”, “PresBocEnt”. El elemento “valFisicos” se

inserta en el renglón número 4 del arreglo resultado, el cual se almacena a la función “Write To Spreadsheet File”. La ruta de almacenamiento es modificada de tal forma que los caracteres “_Calib” son cambiados por el concatenado “_Exp&No.Exp_DatosFisicos”.

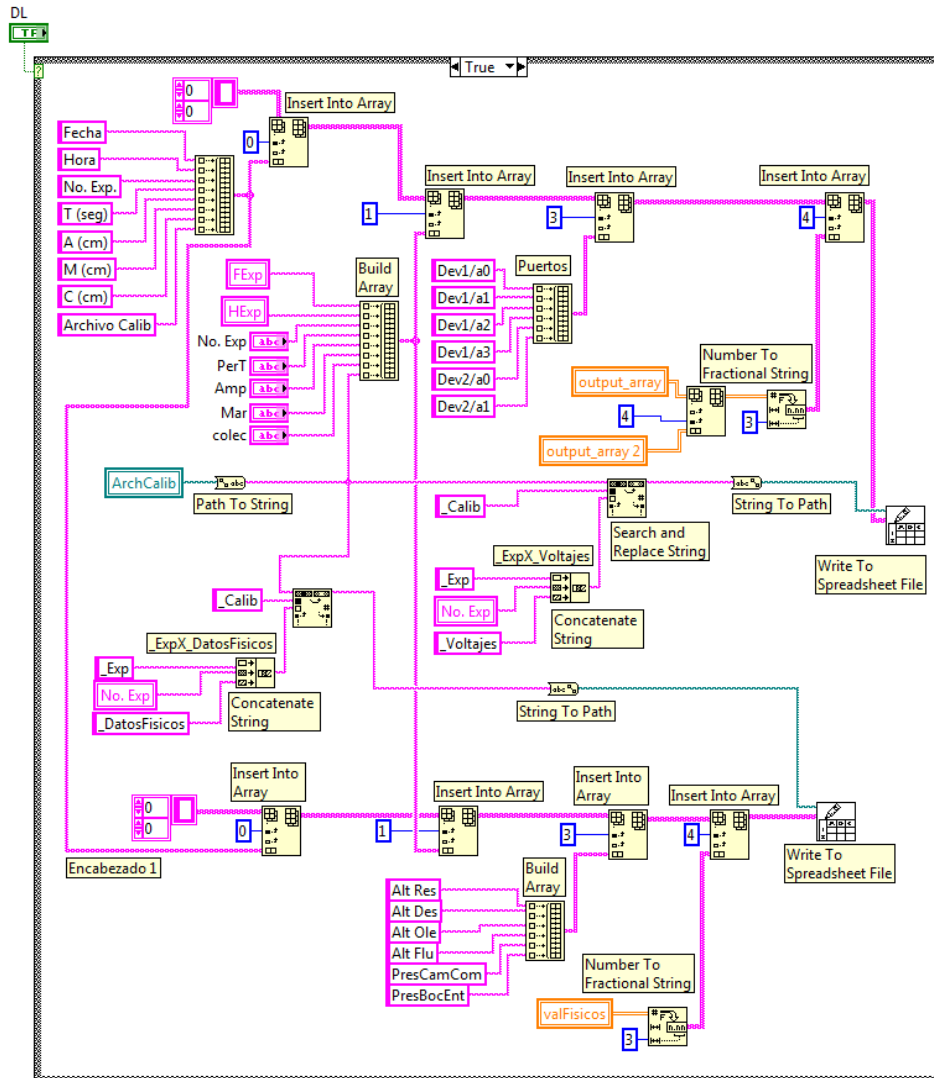


Figura 57. Diagrama de bloque que genera los archivos que contienen los valores brutos y físicas del experimento

3.3.1 Ajuste de requerimientos y reprogramación

De la experiencia con ensayos experimentales previos, se descubre que el tiempo en el cual el sistema debe guardar los valores procesados, oscila entre 20 y 40 segundos. Ante esto se determina cambiar la configuración del control “slide” y acotarlo únicamente a dicho rango, como una oportunidad de mejora. Se aprovecha para cambiar también la presentación de dicho objeto y emplear un control de tipo barra con valores predefinidos. Esto no representa un cambio en la programación, únicamente las características del control en el PF cambiarían.



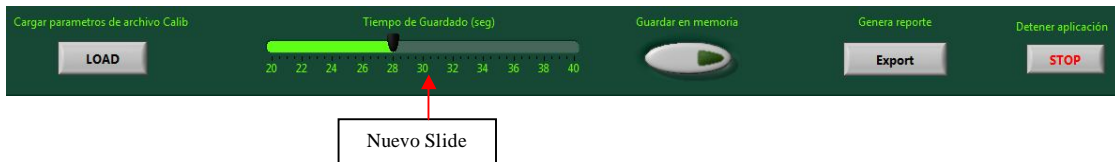


Figura 58. Control slide anterior y posterior a la mejora en el rango de valores

3.4 Pruebas de laboratorio

Las pruebas de laboratorio se realizaron en el laboratorio de acústica de la Facultad de Ciencias (segundo piso del edificio de física) Ciudad Universitaria, con el fin de validar el correcto funcionamiento de cada interfaz.

Para estas pruebas fue necesario contar con los siguientes equipos:

- Fuente de poder Matrix®, MPS-3005L-3, con voltaje y corriente variable, 1 canal de salida fijo en 5 Vcc y 2 canales de salida variable hasta 24 Vcc.
- Laptop Dell® con sistema operativo Windows XP®, memoria RAM de 1GB, procesador Duo e interfaces “Modo Calibración” y “Modo Experimental” precargadas.
- Dos tarjetas de adquisición de datos National Instrument®, NI USB-6009, 8 entradas, 14-bits, multifuncional y con cables USB para comunicación con PC.
- Sensor de presión KELLER, PR-41X, rango de medición de 0 a 0.1 bars y salida de 0 a 10 Vcc.
- Contenedor de plástico con agua.



Figura 59. Fuente de poder



Figura 60. Laptop



Figura 61. Tarjetas de adquisición de datos



Figura 62. Sensor de presión KELLER

Para comenzar el desarrollo se conectaron los DAQS a la computadora, a través de la interfaz USB y posteriormente, empleando la fuente de poder, se alimento el sensor con un voltaje de 15 Vcc (de un rango de 13 a 28 Vcc), su señal de salida fue conectada a la entrada “ai0” de la tarjeta identificada con la etiqueta “S. Presión”.



Figura 63. Conexión de las DAQS con la PC y el sensor de presión

Concluido lo anterior se comenzó encendiendo la fuente de poder y poniendo en marcha la interfaz “Modo Calibración”. Se procedió a sumergir el sensor dentro del contenedor de agua según la siguiente tabla de profundidades, considerando que la altura máxima del SIBEO es de 20cm aproximadamente.

No.	Profundidad (cm)
1	15
2	12
3	9
4	6
5	3
6	0



Figura 64. Sensor de presión sumergido

En cada caso, la interfaz registro el nivel de voltaje proporcional, el cual fue verificado a través de un multímetro digital conectado en paralelo a los DAQS, obteniendo errores de precisión de 0.05%. La interfaz fue capaz de generar de manera instantánea la curva de caracterización y los coeficientes de la regresión polinomial aplicada.

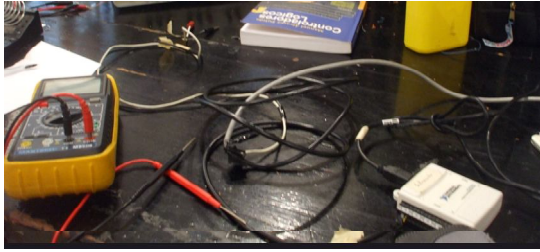


Figura 65. Implementación de un multímetro para la validación del voltaje

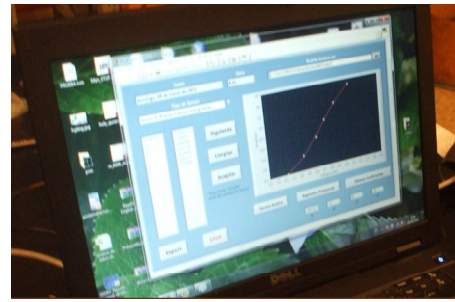


Figura 66. Obtención de la curva de caracterización

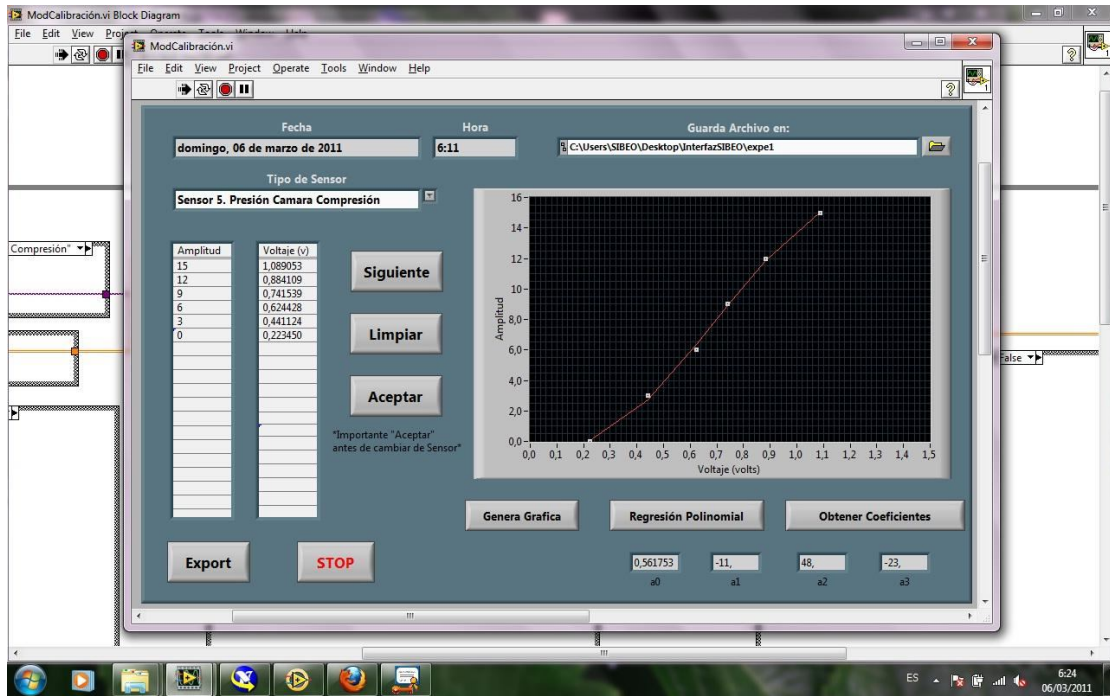


Figura 67. Impresión de pantalla de la interfaz al momento de obtener la curva de caracterización

Tanto los coeficientes como las tablas de datos creadas fueron almacenadas en archivos alfanuméricos y posteriormente visualizados a través de Microsoft Excel®.

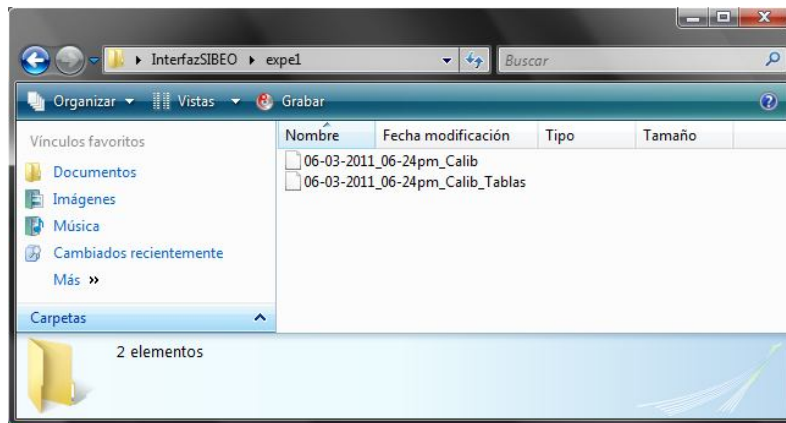


Figura 68. Creación de archivos en la PC

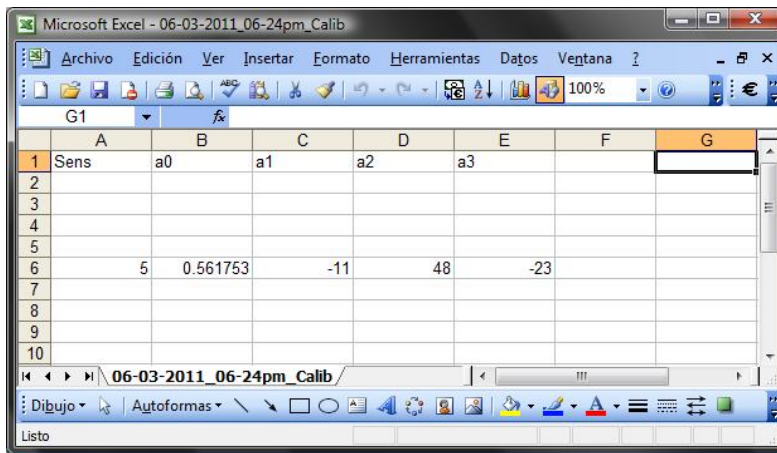


Figura 69. Visualización del archivo “Calib” a través de Microsoft Excel®

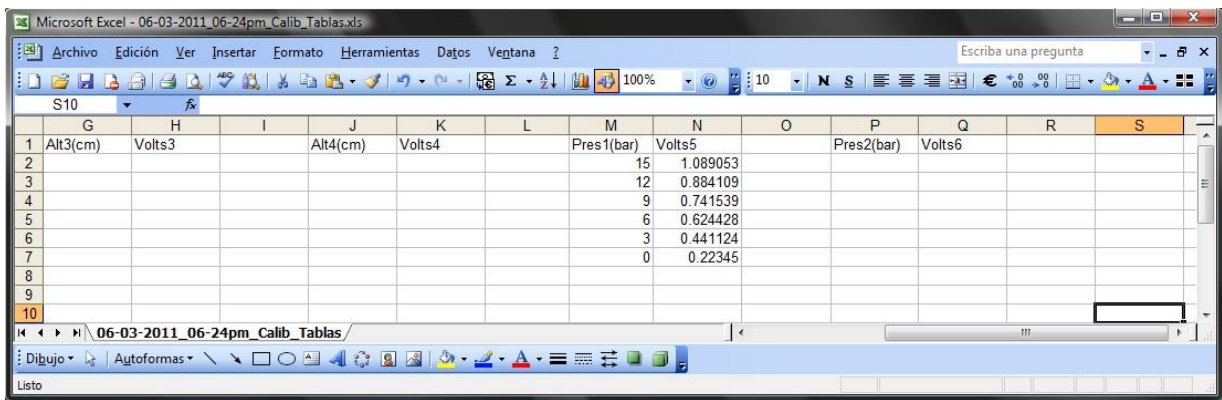


Figura 70. Visualización del archivo “Calib_Tablas” a través de Microsoft Excel®

Para realizar las pruebas de la interfaz “Modo Experimental”, no se requirió realizar ninguna modificación en la conexión de los equipos, el sensor de presión permaneció sumergido en el recipiente con agua, únicamente por cuestiones de seguridad la fuente de poder fue apagada. La interfaz “Modo Calibración” fue detenida activando el botón “STOP”.

Para comenzar con las pruebas de esta segunda interfaz, se dio inicio al sistema activando el botón “Run”, se introdujo el valor de los parámetros del experimento: No. experimento, amplitud, Periodo, etc., y se especifico ubicación de archivo “Calib” acabado de obtener.

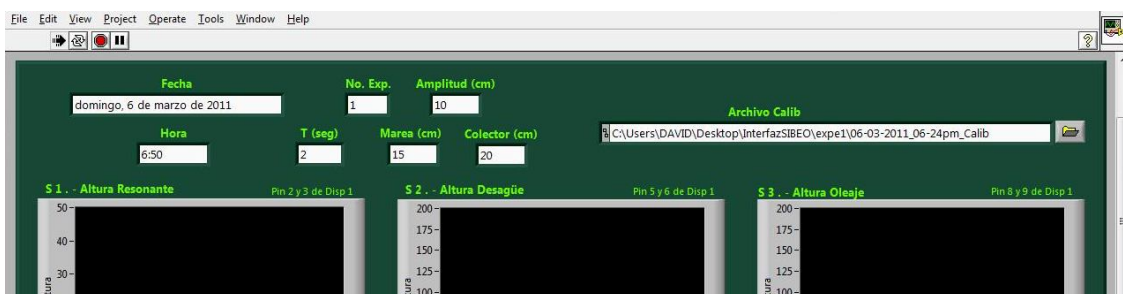


Figura 71. Variables del experimento ingresadas por el usuario

Una vez cargados los datos mediante el botón “LOAD”, se realizaron movimientos ascendentes y descendentes del sensor dentro del recipiente de agua (figura 64) con la intención de simular variaciones de presión reales dentro del SIBEO y visualizarlas en la gráfica correspondiente.

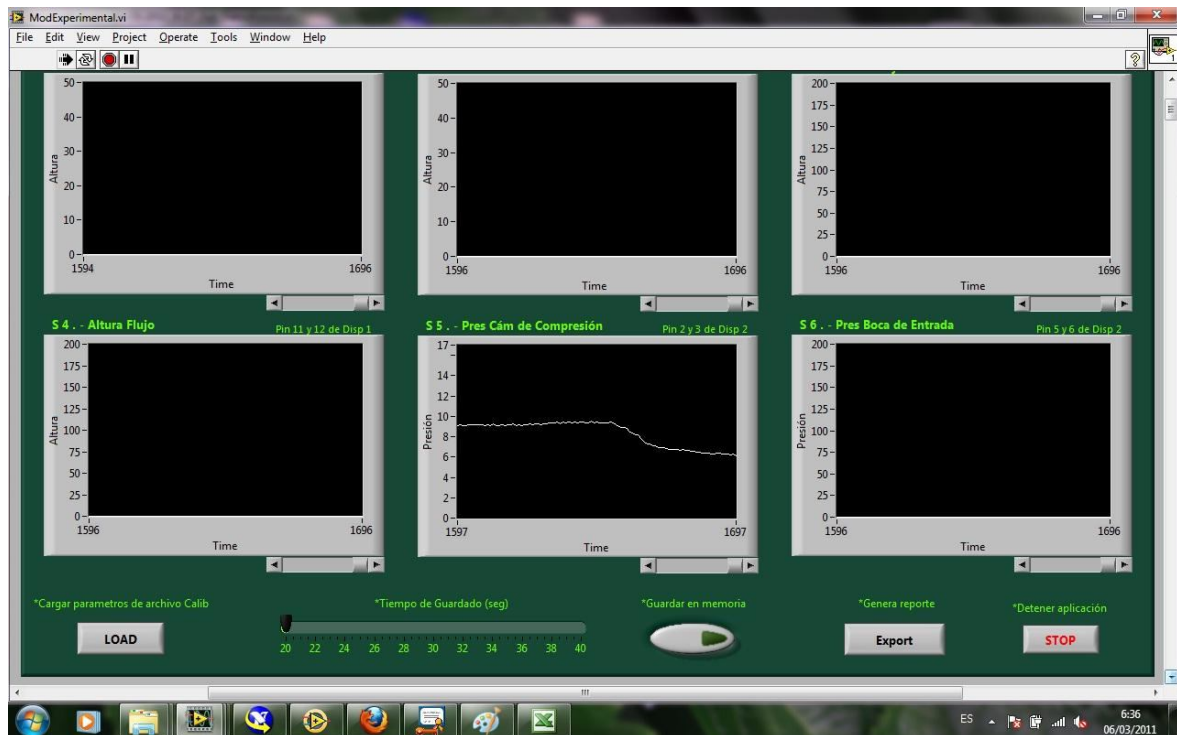


Figura 72. Visualización en pantalla de las variaciones de presión provocadas intencionalmente

Se estableció un tiempo de guardado de 30 segundos para activar el botón “Guardar en Memoria”. Transcurrido el tiempo y activando el botón “Export”, se generaron los archivos de reporte esperados.

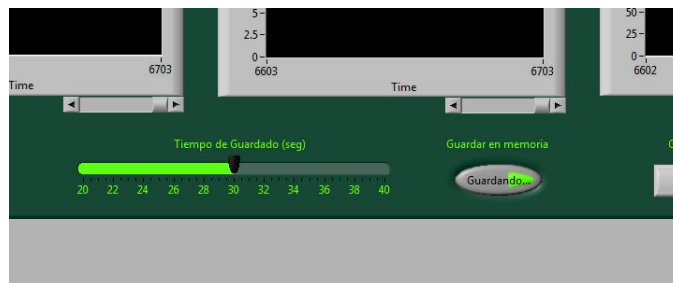


Figura 73. Proceso en curso de almacenado en memoria

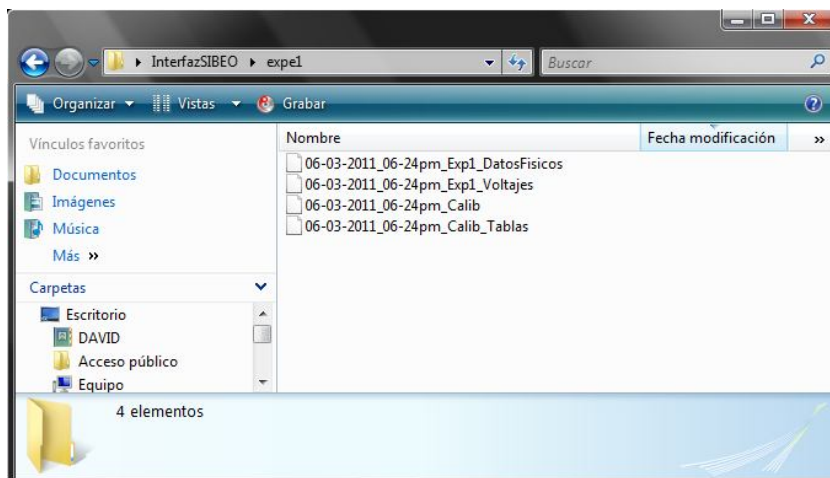


Figura 74. Archivos generados en la misma carpeta que el archivo “Calib”

Empleando Microsoft Excel® fue posible conocer el contenido de los archivos y corroborar que la información contenida era correcta, la figura 75 muestra dicho contenido en Excel®. Cabe mencionar que tales archivos también pueden ser visualizados a través del software Bloc de Notas®, la figura 76 muestra la implementación de este programa para visualizar el contenido.

1	Fecha	B	C	D	E	F	G	H	I	J	K
2	domingo, 6 de marzo de 2011	Hora	No. Exp.	T (seg)	A (cm)	M (cm)	C (cm)	Archivo Calib			
3		06:50	1	2	10	15	20	C:\Desktop\InterfazSIBEO\expe1\06-03-2011_06-24pm_Ca			
4								0			
5	Alt Res	Alt Des	Alt Ole	Alt Flu	PresCamCom	PresBocEnt					
6	0	0	0	0	0.961	0	0	0	0	0	0
7	0	0	0	0	0.961	0	0	0	0	0	0
8	0	0	0	0	0.961	0	0	0	0	0	0
9	0	0	0	0	0.961	0	0	0	0	0	0
10	0	0	0	0	0.961	0	0	0	0	0	0
11	0	0	0	0	4.187	0	0	0	0	0	0
12	0	0	0	0	4.187	0	0	0	0	0	0
13	0	0	0	0	4.187	0	0	0	0	0	0
14	0	0	0	0	0.961	0	0	0	0	0	0
15	0	0	0	0	0.961	0	0	0	0	0	0
16	0	0	0	0	0.961	0	0	0	0	0	0
17	0	0	0	0	4.187	0	0	0	0	0	0
18	0	0	0	0	4.187	0	0	0	0	0	0
19	0	0	0	0	8.493	0	0	0	0	0	0
20	0	0	0	0	8.493	0	0	0	0	0	0
21	0	0	0	0	4.187	0	0	0	0	0	0

Figura 75. Visualización del archivo “_DatosFisicos” con Microsoft Excel®

```

Archivo Edición Formato Ver Ayuda
Fecha Hora No. Exp. T (seg) A (cm) M (cm) C (cm) Archivo Calib
domingo, 6 de marzo de 20.00011 6:50.000 1 2 10 15 20.000 c:\Desktop\InterfazSIBEO\expe1\06-03-2011_06-24pm_Ca

Alt Res Alt Des Alt Ole Alt Flu PresCamCom PresBocEnt
0.000 0.000 0.000 0.000 0.961 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.961 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.961 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.961 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.961 0.000 0.000 0.000
0.000 0.000 0.000 0.000 4.187 0.000 0.000 0.000
0.000 0.000 0.000 0.000 4.187 0.000 0.000 0.000
0.000 0.000 0.000 0.000 4.187 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.961 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.961 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.961 0.000 0.000 0.000
0.000 0.000 0.000 0.000 4.187 0.000 0.000 0.000
0.000 0.000 0.000 0.000 4.187 0.000 0.000 0.000
0.000 0.000 0.000 0.000 4.187 0.000 0.000 0.000
0.000 0.000 0.000 0.000 8.493 0.000 0.000 0.000
0.000 0.000 0.000 0.000 8.493 0.000 0.000 0.000
0.000 0.000 0.000 0.000 4.187 0.000 0.000 0.000
0.000 0.000 0.000 0.000 4.187 0.000 0.000 0.000
0.000 0.000 0.000 0.000 4.187 0.000 0.000 0.000
0.000 0.000 0.000 0.000 4.187 0.000 0.000 0.000
0.000 0.000 0.000 0.000 4.187 0.000 0.000 0.000

```

Figura 76. Visualización del archivo “_DatosFisicos” a través de Bloc de Notas®

4. Manual de operación

Este Manual de Usuario expone el proceso sistemático de operación de cada interfaz, a fin de que el usuario final cuente con una guía clara de cómo utilizar los herramientas que ahora posee para realizar tanto la caracterización de sensores como del monitoreo del SIBEO.

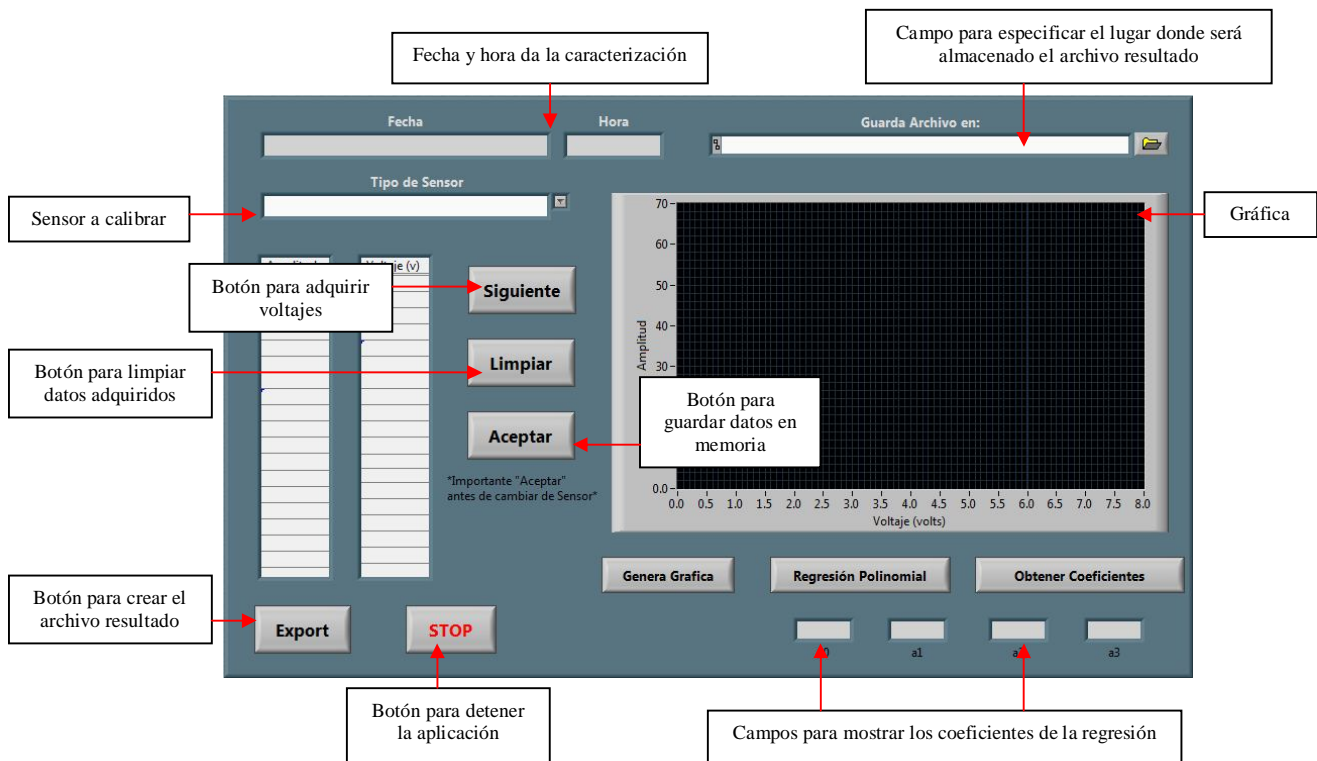
4.1 Manual de operación de la interfaz “Modo Calibración”

La interfaz de calibración fue desarrollada tras la necesidad de caracterizar sensores de presión y altura en el menor tiempo posible para optimizar así el trabajo en campo.


Para el funcionamiento, es necesario primero ingresar las alturas y presiones a las cuales los sensores estarán expuestos, para después adquirir por medio de una tarjeta de adquisición de datos, los valores en voltajes correspondientes a estos parámetros de referencia. Con los datos recolectados es posible generar una gráfica de voltaje contra dichos parámetros, con la intención de tener una idea más clara del comportamiento real del sensor. A través de comandos en forma de botones se obtiene la regresión polinomial que mejor representa dicho comportamiento. Los coeficientes del modelo matemático correspondientes son, de igual forma, obtenidos por comandos similares y presentados en pantalla mediante indicadores especiales. Estos coeficientes serán utilizados en la interfaz de monitoreo para visualizar en tiempo real las presiones y alturas críticas del SIBEO.

Elementos de la Interfaz

A continuación se muestran los elementos de la interfaz.




Operación

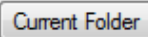
Para comenzar a correr el sistema es necesario presionar el botón “Run”  ubicado en la parte superior-izquierda de la ventana.

Automáticamente aparecerá la fecha y hora de la computadora en los campos correspondientes.

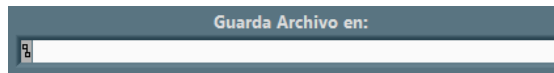


Fecha Hora

A través del botón de navegación  seleccionar la carpeta en la que serán almacenados los archivos resultantes, uno de estos contendrá los coeficientes de regresión polinomial de los distintos sensores caracterizados y el otro guardara las tablas de datos obtenidas en el proceso.

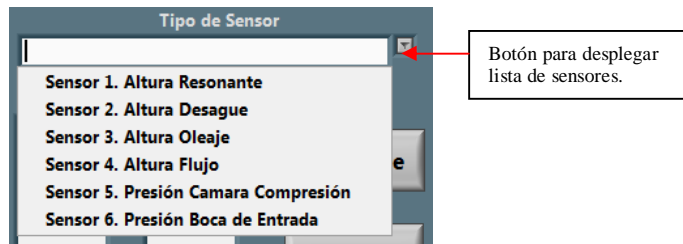
Nota: Para seleccionar la carpeta actual presionar el botón .

La ruta de la carpeta seleccionada aparecerá en el campo “Guardar Archivo en:”



Guarda Archivo en:

A continuación se debe seleccionar el sensor a calibrar presionando el pequeño botón ubicado a la derecha del campo “Tipo de Sensor”, este desplegara una lista de los sensores.



Tipo de Sensor

- Sensor 1. Altura Resonante
- Sensor 2. Altura Desague
- Sensor 3. Altura Oleaje
- Sensor 4. Altura Flujo
- Sensor 5. Presión Camara Compresión
- Sensor 6. Presión Boca de Entrada

Botón para desplegar lista de sensores.

Aparecerá en pantalla el sensor a calibrar.

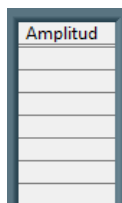


Tipo de Sensor

Sensor 1. Altura Resonante

Nota: Si no se selecciona ningún sensor, el sistema tomara por defecto el sensor 1.

Los valores de altura y presión que el usuario debe definir como parámetros de referencia, deben ser colocados en el campo denominado amplitud.




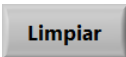
Amplitud

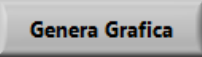
Es importante poner especial atención al momento de ingresar estos datos, ya que es una matriz de celdas en la que se puede mover, y hasta perder de vista, la columna “Amplitud”, si esto ocurre basta con dar *click* con el botón izquierdo del *mouse* en la parte superior-izquierda de cada celda.


Nota: No es necesario correr el programa para ingresar estos datos, ya que pueden ser introducidos previamente.


En este momento el sistema se encuentra listo para comenzar a adquirir voltajes desde de la tarjeta de adquisición NI USB-6009. Es necesario corroborar que dichas tarjetas estén debidamente conectadas a la computadora mediante los puertos USB.

Para adquirir los voltajes de las tarjetas se debe presionar el botón . Se presionara este botón cuantas veces sea necesario cubrir la cantidad de valores de amplitud prefijados para la caracterización.

Si existe algún error en los valores adquiridos, se debe presionar el botón  para borrar dichos valores.

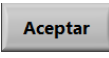
Una vez listas las tablas de amplitud y voltaje, se puede generar la gráfica correspondiente al presionar el botón , este dibujara puntos blancos la relación.

Al presionar el botón  se crea la regresión polinomial (línea en color rojo sobre la gráfica) que mejor se ajusta al comportamiento.

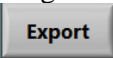
Mediante el botón  los coeficientes del modelo matemático de la regresión a parecen en los siguientes campos:




a0	a1	a2	a3
----	----	----	----

Si la caracterización resulta satisfactoria, se debe presionar el botón  para almacenar tanto la tabla de valores físicos como los coeficientes de la regresión en memoria.

Nota: Es muy importante cerciorarse de que los valores a almacenar correspondan al sensor especificado en el campo “Sensor a Calibrar” ya que si este difiere, estos valores se almacenaran haciendo referencia a otro sensor.

Llegado hasta este punto, el proceso puede darse por concluido. Es importante presionar el botón  para descargar la información almacenada en un archivo alfa-numérico ubicado en la ruta especificada en el campo “Guarda archivo en:”, este podrá ser visualizado con cualquier editor de texto, sin embargo, es recomendable utilizar Microsoft Excel® o Block de Notas®.

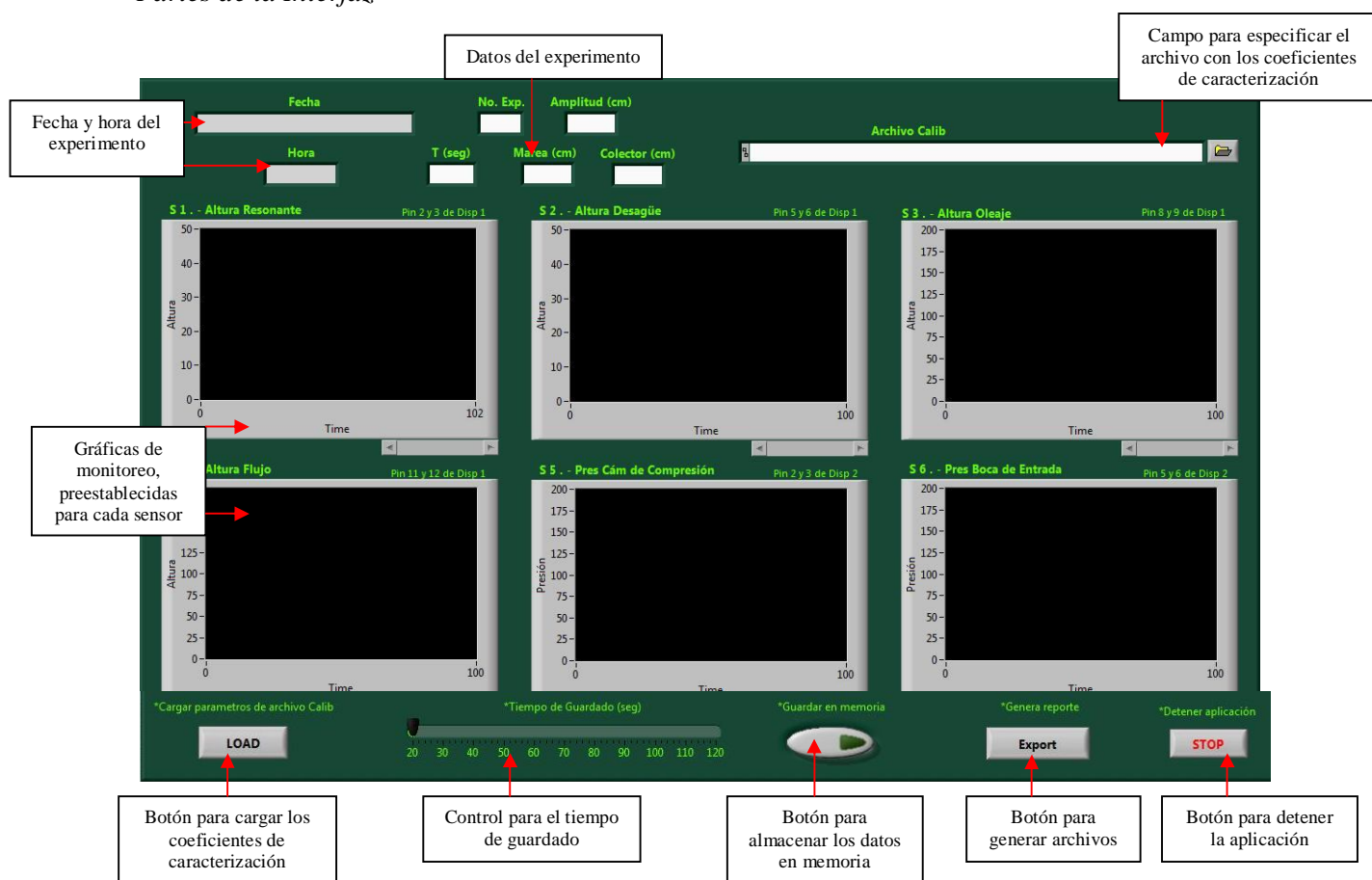
Para finalizar la aplicación basta oprimir .

4.2 Manual de operación de la interfaz “Modo Experimental”


La presente interfaz tiene como objetivo servir al proyecto del SIBEO en el monitoreo de las distintas variables críticas, visualizando de manera general su comportamiento en tiempo real. Esta herramienta genera archivos con el concentrado de valores tanto físicos (procesados) como adquiridos por los sistemas de adquisición de datos en forma de memoria de campo.

Para su funcionamiento primero es necesario ingresar de manera manual los parámetros conocidos del experimento: amplitud de ola, periodo, altura de marea y altura en el colector, así como el archivo que contiene los coeficientes de la caracterización de cada sensor. Después se cargan dichos coeficientes en el procesamiento interno y comenzar así el monitoreo en tiempo real. Si se desea almacenar en memoria los valores visualizados, debe especificarse el tiempo de guardado en un rango de 20 a 120 segundos. La creación de los archivos se lleva a cabo al momento de activar un botón de control.

Partes de la Interfaz





Operación

Para correr el sistema es necesario presionar el botón “Run”  ubicado en la parte superior-izquierda de la ventana.

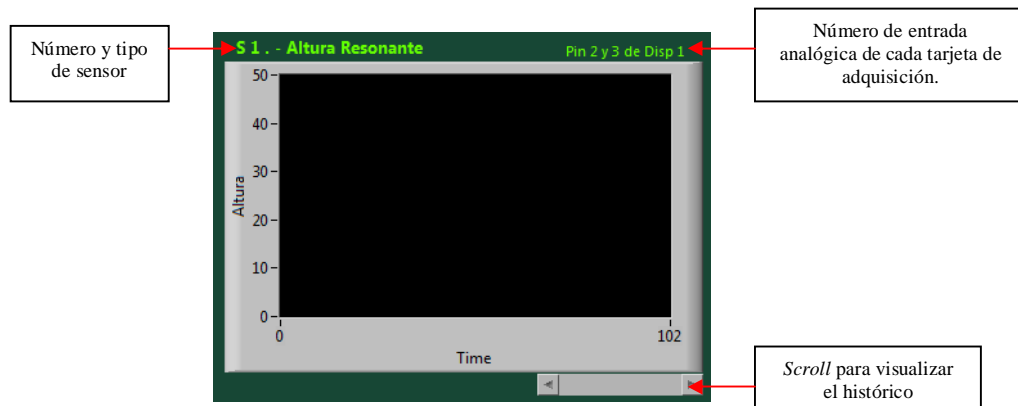
Automáticamente aparecerá la fecha y hora en los campos correspondientes:

Después se debe ingresar los datos del experimento: número de experimento (No. Exp.), amplitud de la marea (cm), periodo de la marea [T (seg)], marea (cm) y altura en el colector (cm). Estos datos aparecerán en los encabezados de los archivos resultantes a fin de identificar claramente el experimento a realizar.

Se deberá seleccionar el archivo con los coeficientes de caracterización en el campo “Archivo Calib”, ya que en este se encuentran los coeficientes de los modelos matemáticos de las distintas regresiones polinomial para cada sensor. Es posible escribir la ruta de dicho archivo o ubicarlo mediante el botón , el cual permite navegar por los archivos de la computadora hasta localizar el archivo “Calib”.



A continuación presionar el botón  para extraer los coeficientes del archivo “Calib” e ingresarlos en el procesamiento, con el fin de mostrar en cada gráfica la variable física del modelo matemático aplicado.

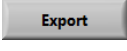
Cada gráfica contiene:





Para cambiar la escala del eje de “Altura” dar doble *click* sobre el último número en la escala y editar su valor, el rango se ajustara automáticamente.

En el momento que las gráficas muestre un comportamiento estable, se especifica el tiempo en que los valores tanto físicos como brutos (voltajes), se deben almacenar en memoria, haciendo uso del siguiente control:

Para iniciar el proceso de almacenaje en memoria, oprimir el botón , este cambiara de estado a  y permanecerá así hasta que el tiempo haya transcurrido.

Posteriormente se genera el archivo resultado oprimiendo el botón , el cual creara dos archivo alfanumérico en la misma ruta del “Archivo Calib” y podrán ser leídos con Microsoft Excel® o bien con Bloc de Notas®.

Nota: Si se desea volver a ejecutar el proceso, se puede volver a presionar el botón , este reemplazara los valores previos.

Para finalizar la aplicación presionar .

Conclusiones

Habiendo alcanzado los requerimientos de diseño satisfactoriamente y mejorado las deficiencias encontradas en las pruebas de laboratorio (estado de saturación y control *slide*), es posible asegurar que las interfaces desarrolladas, y descritas en este trabajo, se encuentran listas para ser empleadas como herramientas adicionales en los procesos de caracterización de sensores y monitoreo experimental del SIBEO. Sin embargo, por ser sistemas desarrollados recientemente, en un tiempo relativamente corto, existen aun algunas áreas de oportunidad que pueden ser analizadas con el fin de mejorar en conjunto el rendimiento general. Estas áreas de oportunidad se presentan a continuación:

- Limpieza individual de celdas y no por elemento. En la interfaz “Modo Calibración” el botón “Limpiar” envía a cero toda la columna de voltaje, pero de la experiencia adquirida en las pruebas de laboratorio, resultaría mejor asignar el valor de cero a una sola celda, pues se podría borrar valores erróneos de manera independiente.
- Utilización de la biblioteca “Report Generation”. En esta biblioteca de LabVIEW se encuentran funciones avanzadas con las que es posible generar reportes con un formato ordenado, en versión HTML o directo a impresoras. Lo anterior pensando en la posibilidad de generar reportes de carácter ejecutivo, de manera inmediata al termino de cada experimento e informar a los directivos a cargo de los resultados obtenidos.
- Empleo de SubVIs para los modelos matemáticos. A fin de hacer más entendible el DB de la interfaz “Modo Experimental”, es conveniente crear SubVIs con las operaciones y funciones matemáticas empleadas para aplicar los modelos matemáticos, pues son procesos repetitivos con igual número de operaciones matemáticas. El uso de SubVIs brinda una presentación mas clara del código de programación.

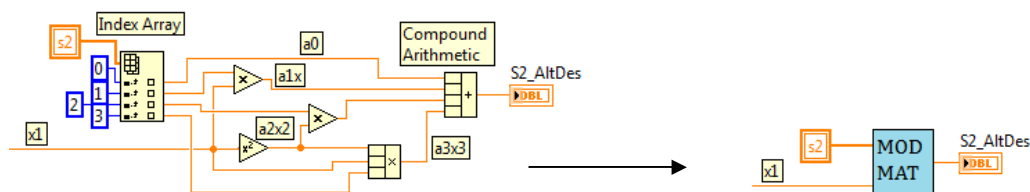


Figura 77. Implementación de un SubVI para las operaciones del modelo matemático

- Empleo de una sola tarjeta de adquisición de datos en lugar de dos. Esta es talvez la mejora mas significativa que podría realizarse. A pesar de que la implementación de dos tarjetas fue uno de los requerimientos para el desarrollo de las interfaces, es importante notar que tanto a nivel de hardware como de software, un solo DAQS es capaz de realizar los procesos de caracterización y monitoreo descritos. Esto traería mejoras directamente al rendimiento del programa pues se eliminarían las funciones utilizadas para el procesamiento de datos de la segunda tarjeta.

Anexos

Anexo 1. NI USB-6009

Low-Cost, Bus-Powered Multifunction DAQ for USB, 12- or 14-Bit, Up to 48 kS/s, 8 Analog Inputs



Up to 48 kS/s	Digital triggering
2 analog outputs at 12 bits, software-timed	Bus-powered
12 TTL/CMOS digital I/O lines	1-year warranty
One 32-bit, 5 MHz counter	

Overview

With recent bandwidth improvements and new innovations from National Instruments, USB has evolved into a core bus of choice for measurement applications. The NI USB-6008 and USB-6009 are low-cost data acquisition (DAQ) devices with easy screw connectivity and a small form factor. With plug-and-play USB connectivity, these devices are simple enough for quick measurements but versatile enough for more complex measurement applications.

Analog Input	
Converter type	Successive approximation
Analog inputs	8 single-ended, 4 differential, software selectable
<i>Input resolution</i>	
NI USB-6008	12 bits differential, 11 bits single-ended
NI USB-6009	14 bits differential, 13 bits single-ended
<i>Max sampling rate (aggregate)</i>	
NI USB-6008	10 kS/s
NI USB-6009	48 kS/s
AI FIFO	512 bytes
Timing resolution	41.67 ns (24 MHz timebase)
Timing accuracy	100 ppm of actual sample rate
<i>Input range</i>	
Single-ended	±10 V
Differential	±20 V ² , ±10 V, ±5 V, ±4 V, ±2.5 V, ±2 V, ±1.25 V, ±1 V
Working voltage	±10 V

Input impedance	144 k Ω
Overvoltage protection	± 35
Trigger source	Software or external digital trigger
<i>System noise</i>	
Single-ended	
± 10 V range	5 mVrms
<i>Differential</i>	
± 20 V range	5 mVrms
± 1 V range	0.5 mVrms

Analog Output

Analog outputs	2
Output resolution	12 bits
Maximum update rate	150 Hz, software-timed
Output range	0 to +5 V
Output impedance	50 Ω
Output current drive	5 mA
Power-on state	0 V
Slew rate	1 V/ μ s
Short circuit current	50 mA
Absolute accuracy (no load)	7 mV typical, 36.4 mV maximum at full scale

Digital I/O

<i>Digital I/O</i>	
P0.<0..7>	8 lines
P1.<0..3>	4 lines
Direction control	Each channel individually programmable as input or output
<i>Output driver type</i>	
NI USB-6008	Open collector (open-drain)
NI USB-6009	Each channel individually programmable as active drive (push-pull) or open collector (open-drain)
Compatibility	TTL, LVTTTL, CMOS
Absolute maximum voltage range	-0.5 to 5.8 V with respect to GND
Pull-up resistor	4.7 k Ω to 5 V
Power-on state	Input

Digital Logic Levels			
Level	Min	Max	Units
Input low voltage	-0.3	0.8	V
Input high voltage	2.0	5.8	V
Input leakage current	—	50	μA
Output low voltage (I = 8.5 mA)	—	0.8	V
<i>Output high voltage</i>			
Active drive (push-pull), I = -8.5 mA	2.0	3.5	V
Open collector (open-drain), I = -0.6 mA, nominal	2.0	5.0	V
Open collector (open-drain), I = -8.5 mA, with external pull-up resistor	2.0	—	V

External Voltage

+5 V output (200 mA maximum)	+5 V typical, +4.85 V minimum
+2.5 V output (1 mA maximum)	+2.5 V typical
+2.5 V accuracy	0.25% max
Reference temperature drift	50 ppm/°C max

Power Requirements

USB

4.10 to 5.25 VDC	80 mA typical, 500 mA max
USB suspend	300 μA typical, 500 μA max

Physical Characteristics

Dimensions

Without connectors	6.35 cm × 8.51 cm × 2.31 cm
With connectors	8.18 cm × 8.51 cm × 2.31 cm
I/O connectors	USB series B receptacle, (2) 16 position terminal block plug headers

Weight

With connectors	84 g
Without connectors	54 g
Screw-terminal wiring	16 to 28 AWG
Torque for screw terminals	0.22–0.25 N · m

Bibliografía

Czitrom, S. P. R., Godoy R., Prado E., Pérez P., Peralta-Fabi. *Hydrodynamics of an oscillating water column seawater pump, Part I: theoretical aspects*. Ocean Engineering 27 (2000), 1181-1198.

Stewart, James. *Cálculo, trascendentes tempranas*, 4ta. ed. Tr. de Andrés Sestier, Thomson, México 2002.

Rodríguez Penin, Aquilino. *Sistemas SCADA*, 2da. ed. Alfaomega / Marcombo, México.

López R., Leobardo. *Programación Estructurada, un Enfoque Algorítmico*, 2da. ed. Alfaomega, México.

Cetinkut, Sabri. *Mecatrónica*, 1ra. ed. Grupo Editorial Patria, México 2007.

Creus Solé, Antonio. *Instrumentación industrial*, 6ta. ed. Alfaomega / Marcombo, México 1997.

LabVIEW Fundamentals, National Instrument, April 2003 Edition.

LabVIEW™ Intermediate I: Successful Development Practices Course Manual

Sapag Chain, Nassir y Sapag Chain, Reinaldo. *Preparación y evaluación de proyectos*, 3ra. ed. McGraw-Hill, Colombia 1996.

Benjamin Niku, Saeed. *Creative design of products and systems*. John Wiley & Sons, United States of America 2009.

G. Alciature, David y B. Hestand, Michael. *Introducción a la mecatrónica y los sistemas de medición*, 3ra. ed. McGraw-Hill, México 2007.

http://www.profesormolina.com.ar/tecnologia/sens_transduct/que_es.htm

<http://www.revistaciencias.com/publicaciones/EpyuVklAAALeskXxCG.php>

<http://www.scribd.com/doc/2634854/MODELOS-MATEMATICOS-DE-SISTEMAS-FISICOS>

http://www.mcgraw-hill-educacion.com/pye01e/cap13/programa13_2.html

<http://www.statsoftiberica.com/es/Blog/RegresionPolinomial.asp>