

CAPÍTULO III

3. DISEÑO E IMPLEMENTACIÓN

El siguiente capítulo detalla el funcionamiento e implementación del sistema de detección de fallas por cortocircuitado en los bobinados de motores de inducción.

3.1 REQUERIMIENTOS GENERALES

El proyecto está dirigido a monitorear el estado de motores de inducción siguiendo los siguientes atributos [19].

- Debe ser confiable y no debe dar falsas indicaciones.
- Debe ser fácil y seguro de operar.
- Debe proveer una buena relación costo-beneficio.
- Debe ser no invasivo.
- Debe ser amigable al usuario.
- Debe ser capaz de adquirir, procesar, interpretar y almacenar varios conjuntos de datos.
- Debe poseer criterios de confiabilidad, escalabilidad y de optimización de recursos [34].
- Debe ser implementado con transductores que se puedan encontrar en el mercado y cuyo desempeño haya sido efectivo en otras aplicaciones.
- Debe mostrar el diagnóstico al tiempo de terminar la prueba.
- Debe ser capaz de generar reportes automáticamente.

3.2 SOFTWARE A UTILIZAR.

El sistema está desarrollado en Labview: *Laboratory Virtual Instrument Engineering Workbench*.

LabVIEW es una aplicación para desarrollo de programas parecido a otros sistemas de desarrollo basados en C++ o BASIC. La diferencia más importante es que Labview no es basado en líneas de código sino en programación gráfica: Lenguaje G [35]. La idea principal es que LabVIEW use diagramas de bloques por los cuales se genere y fluya la información. Además posee librerías específicas de análisis de señales, y adquisición de datos. La ejecución de los programas no los hace línea por línea sino basado en como fluyen los requerimientos de la información.

Esta herramienta tiene además la facilidad de hacer interfaces de usuario muy fácilmente en sus “Paneles Frontales” mientras que el código desarrollado queda transparente al usuario en su “Diagrama de bloques”.

Las estructura de programación es similar a la de un lenguaje basado en texto, así posee librerías, funciones, subrutinas, depuradores y compiladores.

La última versión para Windows XP, disponible hasta el comienzo de la implementación del proyecto es la 8.0 que tiene la facilidad de levantar en los procesos de sistema un administrador de dispositivos que ayuda con la flexibilidad de los dispositivos periféricos como sistemas de adquisición de datos para que sean reconocidos y relacionados con sus controladores de manera ágil.

A continuación, está la explicación exhaustiva de cómo han sido implementados los diferentes bloques del la Figura 2.1.

3.3 MOTOR

El proyecto está diseñado para analizar cualquier tipo de motor que cumpla con las características de la Sección 2.1.3. De aquí que para el proyecto se decidió utilizar un motor con las siguientes características:

- Alimentación Monofásica fase-neutro.
- Tipo jaula de ardilla.
- Frecuencia: 60 Hz.
- Potencia: 1/3 HP.
- Numero de polos 4.
- Velocidad nominal: 1724 RPM.
- Voltaje nominal 120 V.
- Corriente nominal: 3A.

En un inicio el motor se encuentra en buen estado y después se cortocircuitan algunas bobinas para pruebas. Este proceso se detalla en el capítulo IV.

3.4 TRANSDUCTOR

3.4.1 TRANSDUCTOR DE CORRIENTE

Una vez que se han analizado las opciones de sensado de corriente de la sección 2.2.1 y comparándolas con los requerimientos del proyecto descritas en la sección 2.2.2, se puede concluir que la mejor opción que además cumple con cada uno de ellos es la pinza amperimétrica Fluke 80i-600A.

De esta manera, en la Tabla 3.1, se puede comparar las características de la pinza amperimétrica con los requerimientos del sistema.

Tabla 3.1. Comparación de requerimientos de sistema y pinza

Requerimiento de sistema	Característica de pinza amperimétrica 80i-600A
Método no invasivo.	Método no invasivo.
Rango de medición de corriente al menos del valor nominal o de plena carga del motor a ser evaluado (3 A).	Rango de medición de 1 a 600 A.
La salida debe conservar todas las características de la corriente sensada.	Tiene una precisión de 3%.
La salida del sensor debe poder ser acondicionada a un nivel de voltaje de hasta $\pm 15V$ pico.	La salida en Voltios es proporcional a la corriente en Amperios, por lo que puede acondicionarse.
Ancho de banda de 0-500 Hz mínimo.	Ancho de banda 0-50kHz.

A la salida de la pinza, se debe poner una resistencia de 12Ω . Con eso se puede tener una salida de 4.24 V pico dentro del rango deseado.

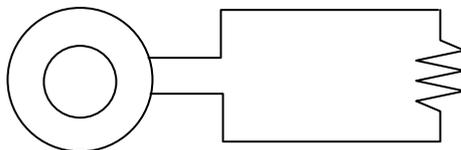


Figura 3.1. Pinza con Resistencia de acondicionamiento.

3.4.2 TRANSDUCTOR DE VOLTAJE

Una vez que se han analizado las opciones de sensado de voltaje de la sección 2.2.3 y comparándolas con los requerimientos del proyecto descritas en la sección 2.2.4, se puede concluir que la mejor opción que además cumple con cada uno de ellos es el transformador de voltaje.

De esta manera, en la Tabla 3.2, se puede comparar las características del transformador de voltaje con los requerimientos del sistema.

Tabla 3.2. Comparación de requerimientos de sistema y transformador de voltaje.

Requerimiento de sistema	Característica del transformador de voltaje
Método no invasivo.	Método no invasivo.
Debe ser capaz de sensar el voltaje de alimentación al motor, que para una red en bajo voltaje puede ser 208/120 V	Medición de 120 o 208 V. a 9-18V con tap central.
La salida debe conservar todas las características de la corriente sensada.	Tiene alta precisión, es decir, se pueden diferenciar variaciones de magnitudes muy pequeñas.
Salida del sensor debe poder ser acondicionada a un nivel de voltaje de hasta $\pm 15V$ pico.	La salida en Voltios es proporcional y reducida y puede acondicionarse.
Ancho de banda de 0-500 Hz mínimo.	Ancho de banda mayor a 500 Hz.

A la salida del transformador hay que poner una resistencia que limite la corriente en el secundario. El valor de esta resistencia es de $1\text{ k}\Omega$ para que por ahí circulen 9 mA , que es una corriente suficientemente pequeña para ser manejada por el equipo de adquisición y tan grande como para que no sea comparable con el ruido.

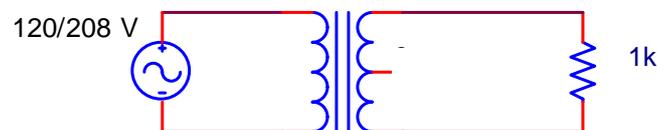


Figura 3.2. Transformador de voltaje

3.5 ADQUISICIÓN DE DATOS.

3.5.1 FILTRO *ANTI-ALIASING*

La banda de paso del filtro debe ser lo más plana posible y la banda de transición lo más angosta posible. Por esto el filtro es de tipo Butterworth, de séptimo orden diseñado utilizando el programa *Filterlab* distribuido en línea por *Microchip* y simulado en ORCAD PSpice. El esquemático se muestra en la Figura 3.4:

Esta implementación tiene la siguiente salida en simulación:

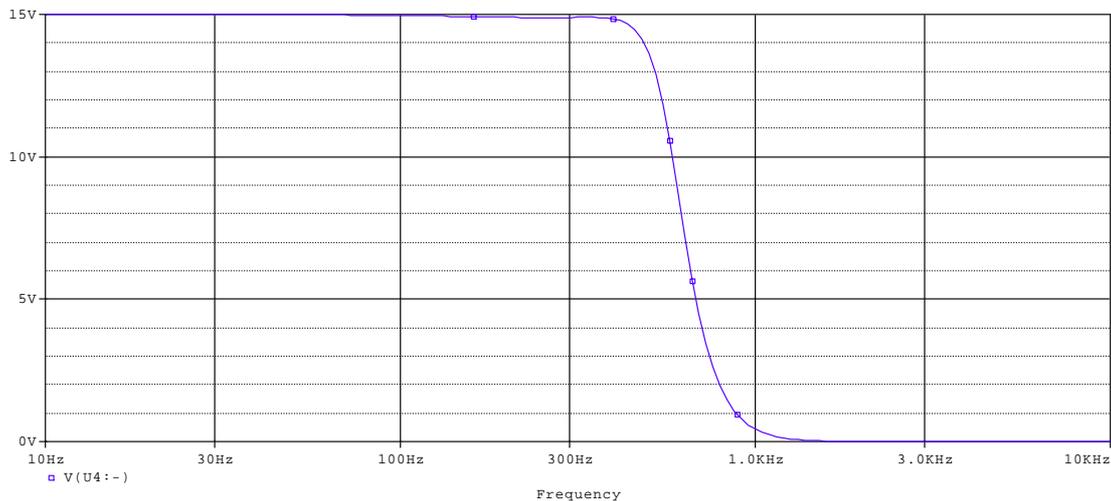


Figura 3.3. Respuesta en frecuencia del filtro *Anti-Aliasing*

Como se observa, la magnitud de salida a una señal de entrada de 15 V en 2 kHz es de 4.2 mV, lo que corresponde a una atenuación aproximadamente de -71 dB que está en un nivel bastante aceptable para estas frecuencias.

Debido a que los Amplificadores Operacionales están polarizados con $\pm 15V$ su salida no puede exceder estos valores dando una protección adicional a la entrada de la DAQ. Se implementaron dos filtros idénticos, uno para la señal de corriente y otro para la señal de voltaje.

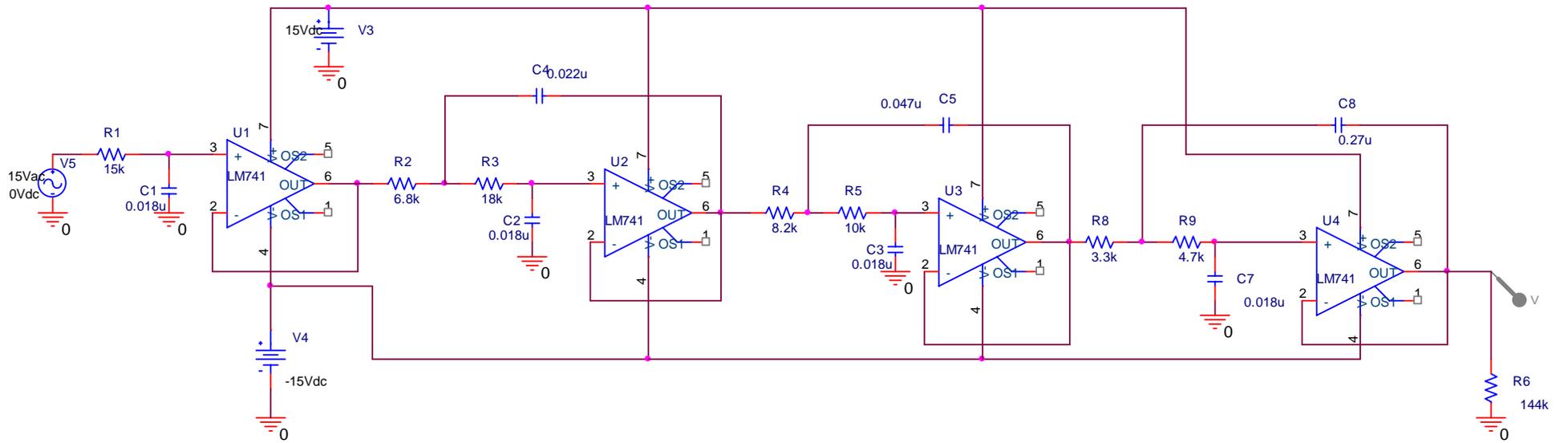
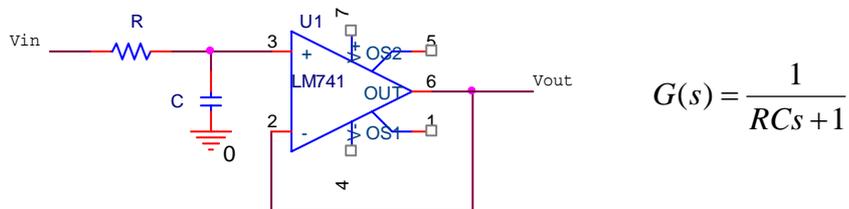


Figura 3.4. Esquemático del Filtro *Anti-Aliasing*

El circuito es implementado en cuatro etapas con topología Sallen-Key en cascada en cascada.

La primera etapa es de primer grado, su esquemático y su función de transferencia son:



$$G(s) = \frac{1}{RCs + 1}$$

Figura 3.5. a) Primera etapa filtro *anti-aliasing*, a) esquema, b) función de transferencia

Donde:

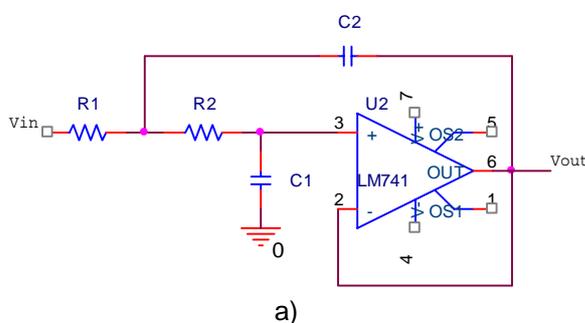
$$R = 15 \text{ k}\Omega$$

$$C = 0.018 \text{ }\mu\text{F}$$

de esto que la función de transferencia para esta etapa es:

$$G(s) = \frac{1}{0,00027s + 1} \text{ y un punto de media potencia de } 587 \text{ Hz.}$$

Las siguientes tres etapas son de segundo orden, y cada una tiene el siguiente esquemático, función de transferencia y frecuencia natural:



$$G(s) = \frac{1}{R_1 R_2 C_1 C_2 s^2 + (R_1 C_1 + R_2 C_1) s + 1}$$

b)

$$f_0 = \frac{1}{2\pi \sqrt{R_1 R_2 C_1 C_2}}$$

c)

Figura 3.6. a) Etapas 2 a 4 de filtro *anti-aliasing*, a) esquema, b) función de transferencia, c) frecuencia de corte rad/s

Para cada etapa se tienen los siguientes parámetros:

Tabla 3.3. Parámetros de las etapas del filtro.

Etapa	Valores de elementos	Función de transferencia	Frecuencia de corte (Hz)
Segunda	$R_1 = 6,8 \text{ k}\Omega$ $R_2 = 18 \text{ k}\Omega$ $C_1 = 0,018 \text{ }\mu\text{F}$ $C_2 = 0.022 \text{ }\mu\text{F}$	$G(s) = \frac{1}{4,85E - 8s^2 + 5,18E - 4s + 1}$	722
Tercera	$R_1 = 8,2 \text{ k}\Omega$ $R_2 = 10 \text{ k}\Omega$ $C_1 = 0,018 \text{ }\mu\text{F}$ $C_2 = 0.047 \text{ }\mu\text{F}$	$G(s) = \frac{1}{6,94E - 8s^2 + 3,28E - 4s + 1}$	637
Cuarta	$R_1 = 3.3 \text{ k}\Omega$ * $R_2 = 4,7 \text{ k}\Omega$ $C_1 = 0,018 \text{ }\mu\text{F}$ $C_2 = 0.27 \text{ }\mu\text{F}$	$G(s) = \frac{1}{7,53E - 8s^2 + 1.44E - 4s + 1}$	580

* Este valor es un potenciómetro de precisión para regular la salida.

Las respuestas en frecuencia de cada una de estas funciones están en

la Figura 3.7:

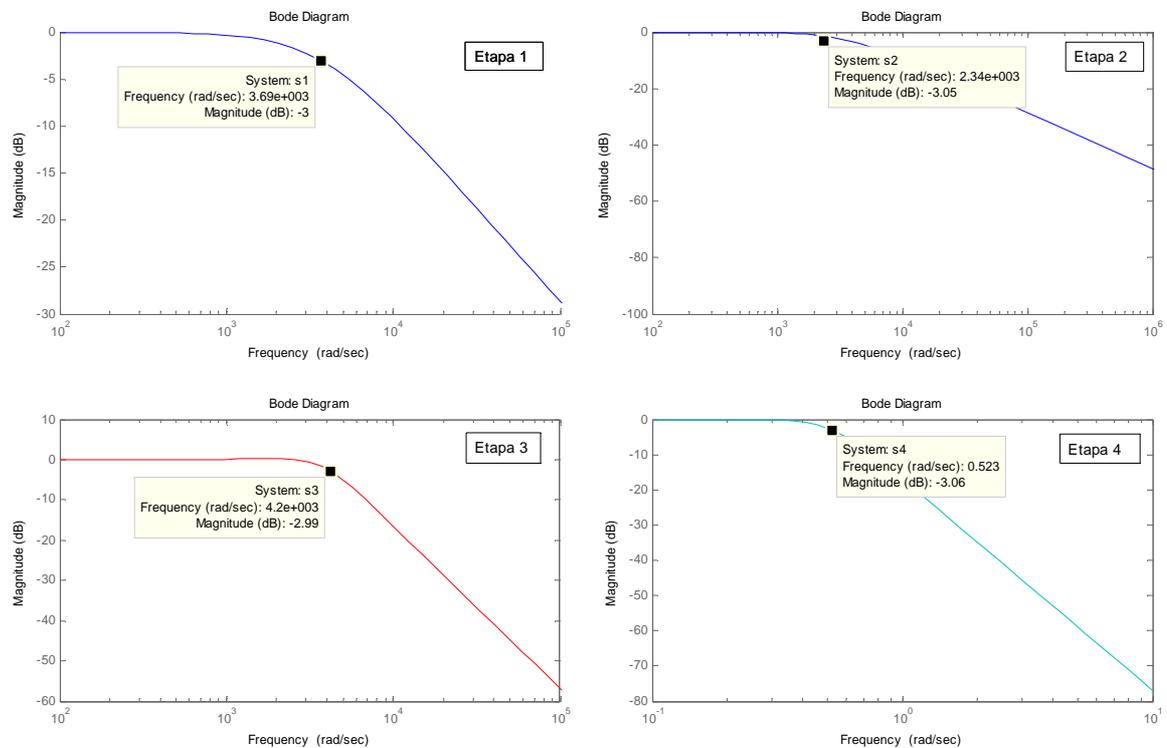


Figura 3.7. Respuesta en frecuencia de cada etapa del filtro *anti-aliasing*

De los datos anteriores llegamos a la función de transferencia total:

$$G(s) = \frac{1}{6,843E-26s^7 + 1,439E-21s^6 + 1,317E-17s^5 + 7,49E-14s^4 + 2,879E-10s^3 + 7,522E-7s^2 + 1,26E-3s + 1}$$

y la respuesta en frecuencia es la siguiente:

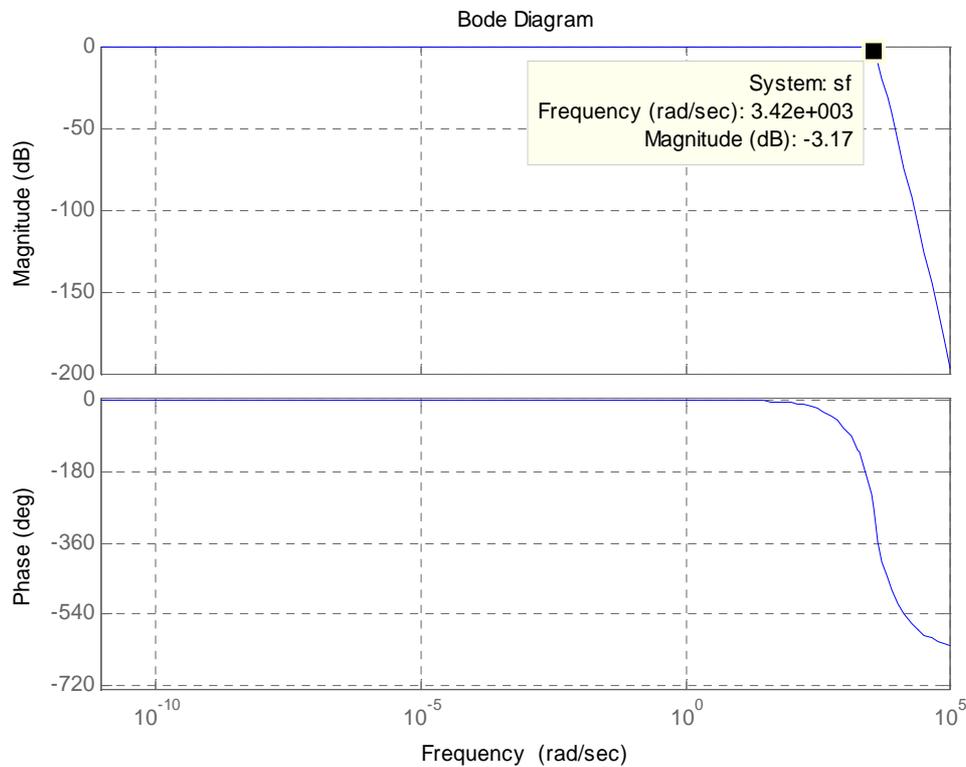


Figura 3.8. Respuesta en frecuencia del filtro *anti-aliasing*

El circuito se ha implementado con amplificadores operacionales LM741, ya que su ancho de banda está entre 0,437 y 1,5 MHz. Éste es un rango muy superior a la frecuencia que trabaja el filtro, por lo que no habrá atenuación debida a las características de construcción del amplificador operacional.

Existen dos filtros idénticos, uno para la señal de corriente y uno para la señal de voltaje.

3.5.2 EQUIPO DE ADQUISICIÓN DE DATOS

El equipo con el cual se realiza la implementación del proyecto debe cumplir con las especificaciones del sistema de adquisición de datos de la sección 2.3.4. En vista de estos requerimientos, la mejor solución que cumple con todas las características es hacer la implementación con una Tarjeta de Adquisición de Datos (DAQ).

En el mercado existen algunas opciones de DAQs que responden correctamente a las necesidades del proyecto, de esta manera es que se ha elegido la DAQ USB-6009, fabricado por la *National Instruments (NI)* y totalmente compatible con LabVIEW.

La apariencia física de la DAQ USB-6009 es la siguiente:



Figura 3.9. DAQ USB-6009 fotografía www.ni.com

En la Tabla 3.4 se realiza una comparación entre los requerimientos del sistema y las características del equipo con el que se va a implementar.

Tabla 3.4. Comparación entre requerimientos de sistema y características del equipo a utilizar.

REQUERIMIENTOS DEL SISTEMA	CARACTERÍSTICAS DEL EQUIPO
Sistema de adquisición que utilice una interfase USB.	Comunicación USB 2.0 <i>Full speed</i>
El sistema debe tener un rango de $\pm 15V$.	Rango máximo de $\pm 20V$ (diferencial)
El ADC puede ser por aproximaciones sucesivas o flash.	ADC por aproximaciones sucesivas.
El ADC debe tener una resolución mínima de 12 bits	Resolución de 14 bits
El SQNR debe ser de al menos 72 dB	SQNR = 82 dB
El sistema debe ser capaz de muestrear en el rango de 2,5 –4 kS/s y enviar los datos a una PC, preferiblemente con DMA	Velocidad de muestreo máximo de 48 kS/s. Envío de datos mediante búfer directo a memoria RAM.
Debe tener dos canales de entrada analógica con muestreo simultáneo o por multiplexación, y dos de E/S digitales.	8 canales de entrada analógica referidas a tierra o 4 diferenciales. Muestreo simultáneo. 10 E/S digitales.

Las características completas de la DAQ USB-6009 se encuentran en la página web de *National Instruments*.

La velocidad de muestreo, es configurada por software a través del *Measurement and Automation Studio*, que es compatible con LabVIEW, de la siguiente manera en la Figura 3.10:

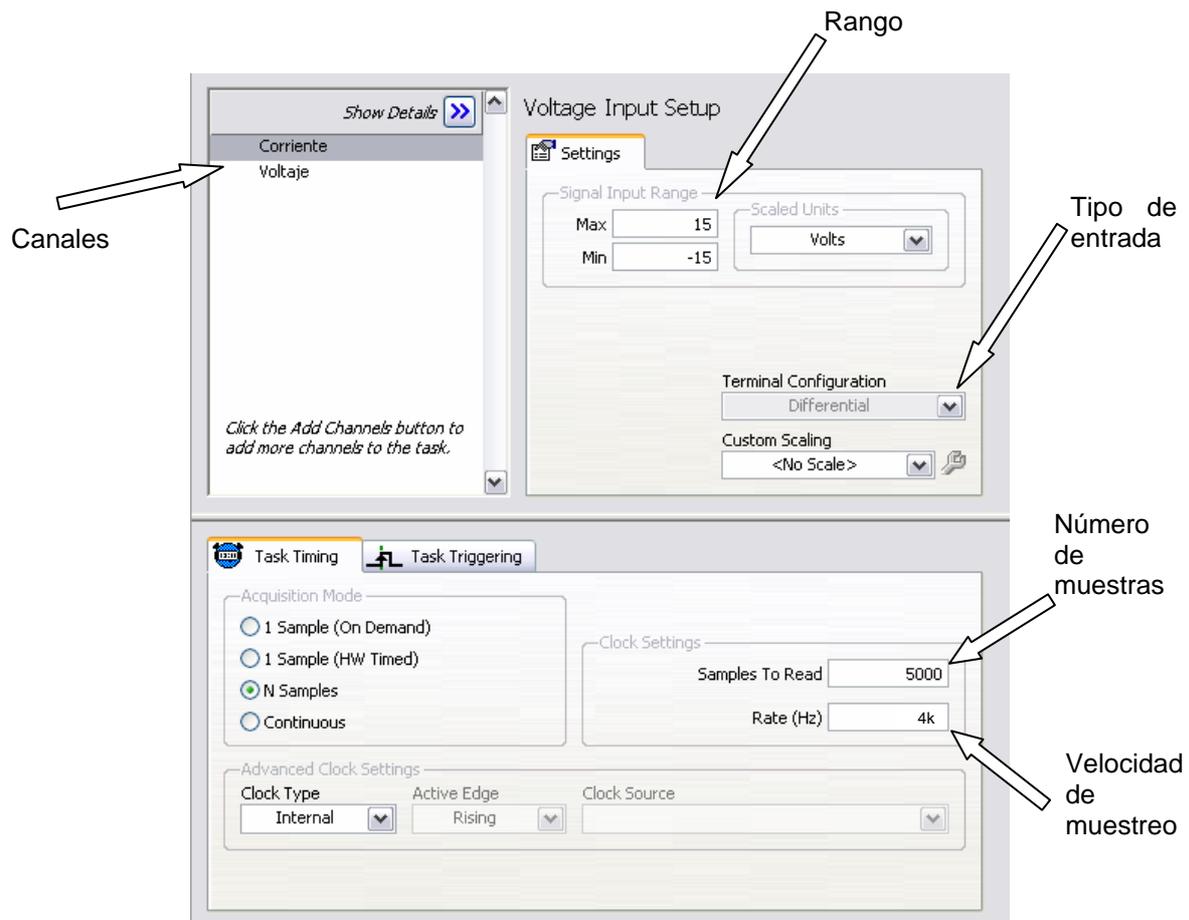


Figura 3.10. Configuración de la DAQ.

En el programa deberá haber dos tareas: una llamada *Corriente* y otra llamada *Voltaje*.

Tarea “Corriente”

Es la encargada de adquirir la señal de corriente desde el transductor. Los límites de esta deben estar entre ± 15 V. La velocidad elegida en la práctica es de 4 kS/s.

Tarea “Voltaje”

Es la encargada de adquirir la señal de voltaje desde el transductor. Los límites de este deben estar entre $\pm 15V$ y tener un escalamiento lineal a 0-169V con la siguiente configuración:

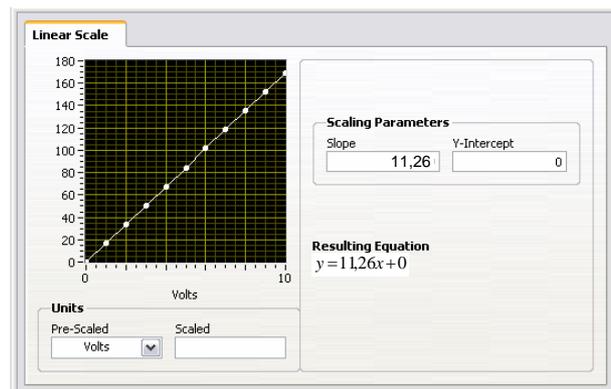


Figura 3.11 Implementación de tarea Voltaje

La escala debe ser lineal y obedecer la ecuación $y = 11,26x + 0$ ya que 15 voltios en la DAQ deben representar $120\sqrt{2} = 169$ V pico de voltaje de entrada al motor. Pero debido a que es importante notar la forma de onda de corriente en el mismo gráfico, no se usará este escalamiento.

Tipo de entrada al dispositivo

El tipo de entrada al dispositivo es el diferencial, así que las entradas deben ir conectadas de la siguiente manera (del reporte generado por *Measurement and Automation Explorer*):

Task Name: Medicion Analogica

Channel Name	Physical Channel	Device Type	Measurement Type
Corriente	Dev1/ai0	USB-6009	Voltage
Voltaje	Dev1/ai1	USB-6009	Voltage

Corriente

Point 1

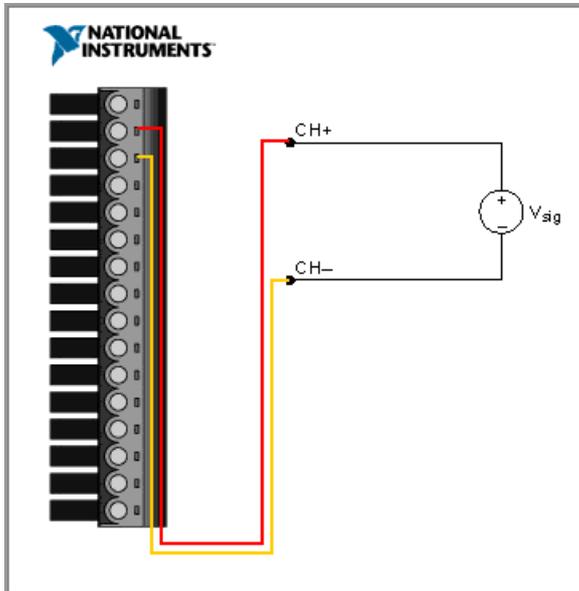
Voltage/CH+

Voltage/CH-

Point 2

16PinCombicon/2

16PinCombicon/3



Voltaje

Point 1

Voltage/CH+

Voltage/CH-

Point 2

16PinCombicon/5

16PinCombicon/6

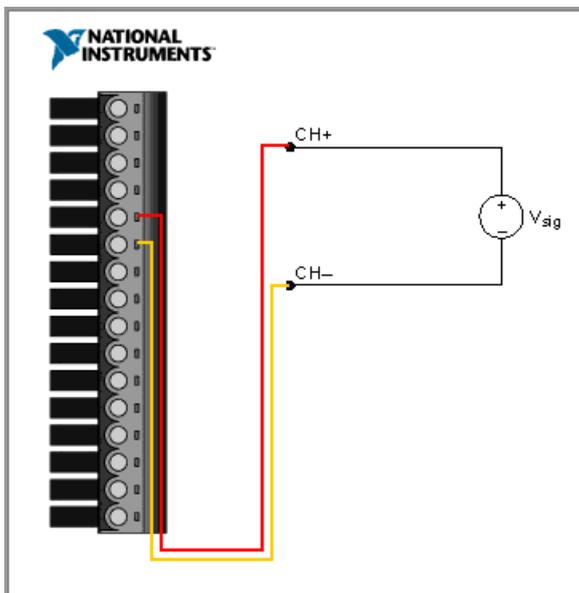


Figura 3.12. Conexión de entradas en la DAQ.

De esta manera se consiguen las ventajas discutidas en la sección 2.3.2.

La implementación de la lectura en el programa se la realiza a través de la librería DAQmx de LabVIEW, con el siguiente código:

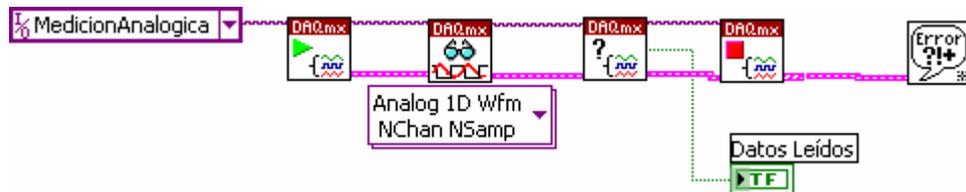


Figura 3.13. Implementación de adquisición de datos

En el primer bloque, se elige la tarea que va a trabajar, después, se inicia comunicación con la DAQ, se leen las muestras con un formato de arreglo unidimensional para forma de onda, se verifica que la tarea se haya realizado y se escribe una notificación, se cierra comunicación con la DAQ en esta tarea, y por último, se completa con un manejador de errores que envía un mensaje relacionado si existiera.

3.5.3 AISLAMIENTO A DIFERENTES FUENTES DE RUIDO

La eliminación a distintas fuentes externas de ruido se la realiza principalmente utilizando par trenzado para la conexión entre la DAQ, el filtro *Anti-Aliasing* y las señales de los transductores, además de tener un plano de tierra bordeando la tarjeta impresa en donde se encuentra el filtro.

3.6 PROCESAMIENTO DIGITAL DE SEÑALES

3.6.1 ELIMINACIÓN DE LA PERIODICIDAD EN FRECUENCIA DEBIDO AL MUESTREO, DE RUIDO Y COMPONENTES PARÁSITAS.

Como se discutió en la sección 2.4.2, se ha decidido implementar un filtro IIR tipo pasabajas de topología Butterworth de cuarto orden en los que la frecuencia de corte es de 500 Hz.

Entre el software de *National Instruments* que viene integrado con LabVIEW está el denominado *Express* que permite hacer implementaciones con aplicaciones precompiladas y configurables en tiempo de programación. Es así que se tiene la siguiente configuración del filtro digital para el proyecto:

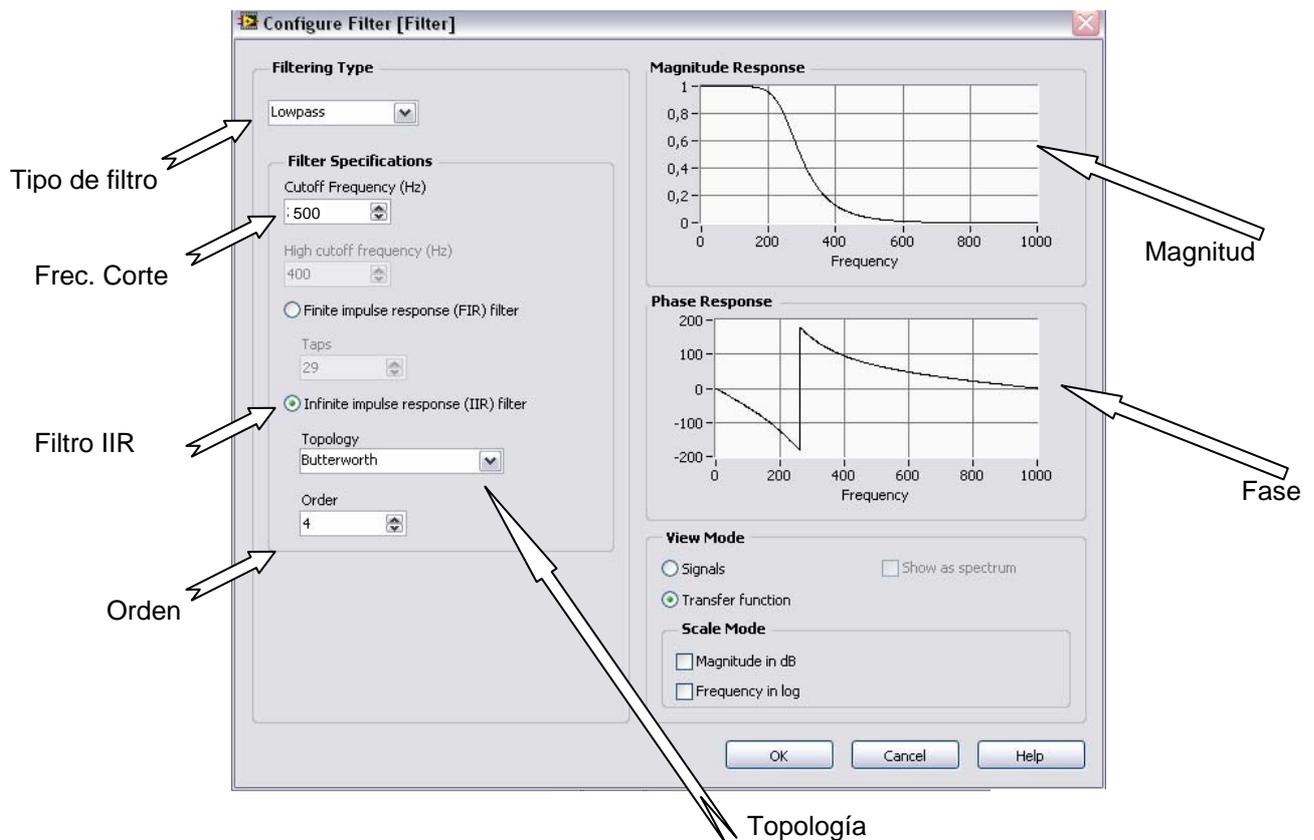


Figura 3.14. Configuración del filtro digital

Estas configuraciones se guardan en memoria y se las utiliza en el programa principal mediante un ícono específico:

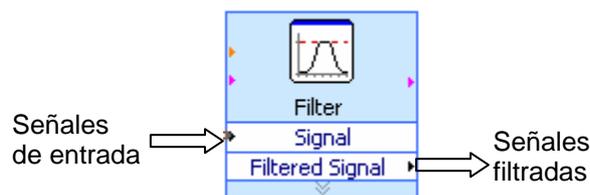


Figura 3.15. Implementación del filtro digital en LabVIEW

3.6.2 ANÁLISIS EN FRECUENCIA

Según la discusión de la sección 2.4.4, las características del análisis en frecuencia son:

- Análisis en frecuencia por transformada de Fourier.
- Implementación mediante FFT.
- La secuencia de entrada debe usar *zero padding*.
- Optimización de FFT mediante promedio RMS.
- Se hará una ponderación de mediciones de tipo lineal.
- Implementación con ventana tipo Hanning

Para la implementación del algoritmo FFT, tanto para corriente como para voltaje, se decidió utilizar la herramienta *Power Spectrum* de *Signal Express* de LabVIEW. Esta herramienta es implementada con *zero padding* [30]. Se la ha configurado de la siguiente manera:

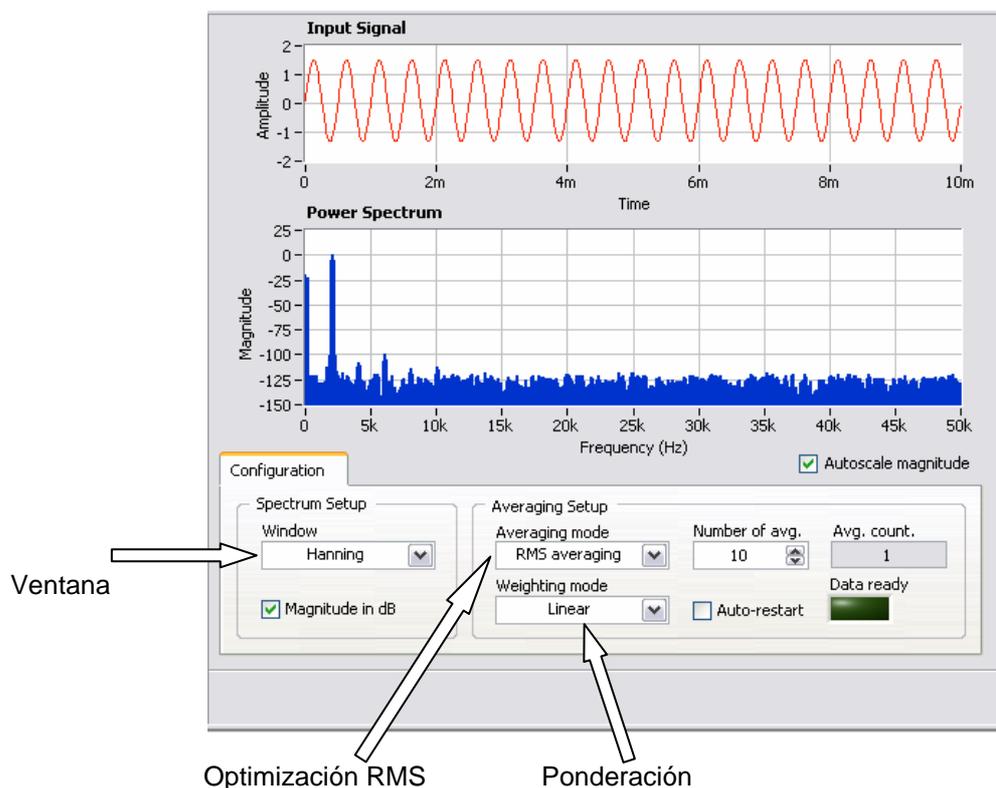


Figura 3.16. Configuración de análisis espectral

Las funciones que se utilizan para la programación general son las siguientes:

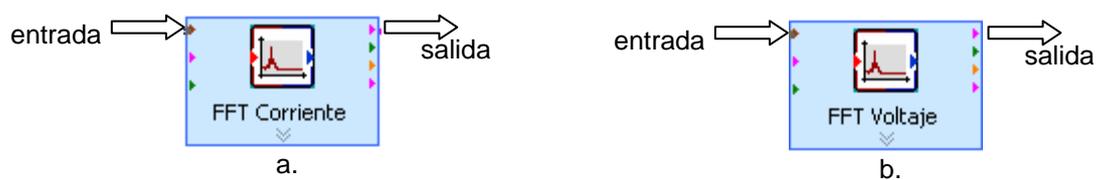


Figura 3.17. Implementación de FFT de corriente y voltaje

En el programa se implementa de la siguiente manera:

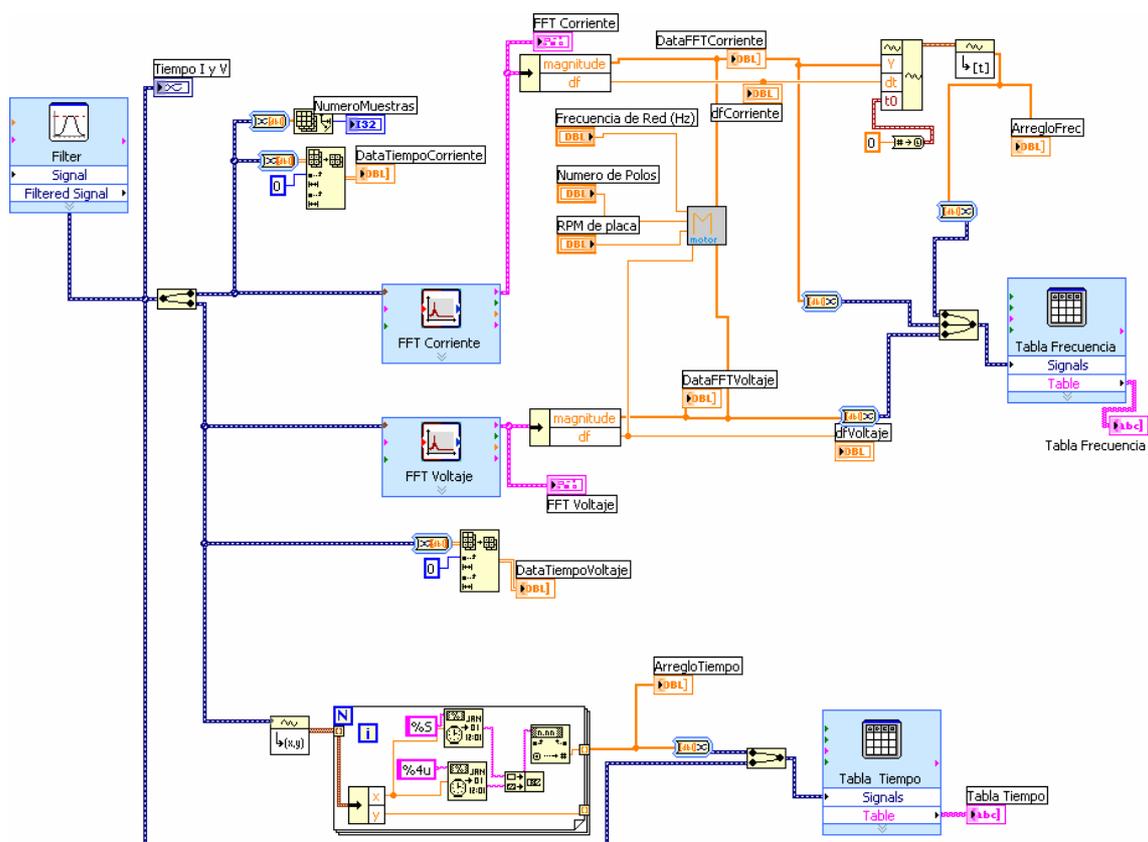


Figura 3.18. Procesamiento digital de señales y MCSA

Las señales adquiridas entran al filtro, y luego para cada una se halla su espectro en frecuencia. De esta manera se implementa el algoritmo de MCSA en el bloque MOTOR que hace la detección de la falla. Al mismo tiempo, se escriben en las tablas que serán accesibles al usuario en tiempo de ejecución.

3.7 ALGORITMO DE MCSA

El algoritmo de MCSA es implementado en la función MOTOR que recibe como datos:

- Un arreglo de los valores de magnitud de corriente y voltaje en frecuencia.
- Espaciamiento en frecuencia
- Frecuencia de red
- Número de polos
- Velocidad real del motor.

Con estos datos se calculan las frecuencias de las bandas central y las correspondientes a donde se hallarían las fallas. Se encuentran los equivalentes en número de muestra a dichas frecuencias y se desplaza dos muestras en cada una.

Por otra parte, en los arreglos de corriente y voltaje, se detectan todos los picos existentes a través de la función “detector de picos” precompilada en LabVIEW. La salida de esta función que se usará, es la localización de cada pico en número de muestra, de tal manera que entrega un arreglo en la que cada elemento es la localización de un pico (arreglo de picos).

A través de la función de “En rango y coerción” se determina si alguna de las localizaciones de algún pico está entre el número de muestra correspondiente a las frecuencias de interés menos dos, y éste último sumado cinco muestras. Es decir, se toma el intervalo que se encuentra entre las frecuencias de interés menos dos muestras y más dos muestras. Esto se realiza para evitar que una falla no sea detectada por un error en la medición de la velocidad del rotor, dándole así, flexibilidad al sistema.

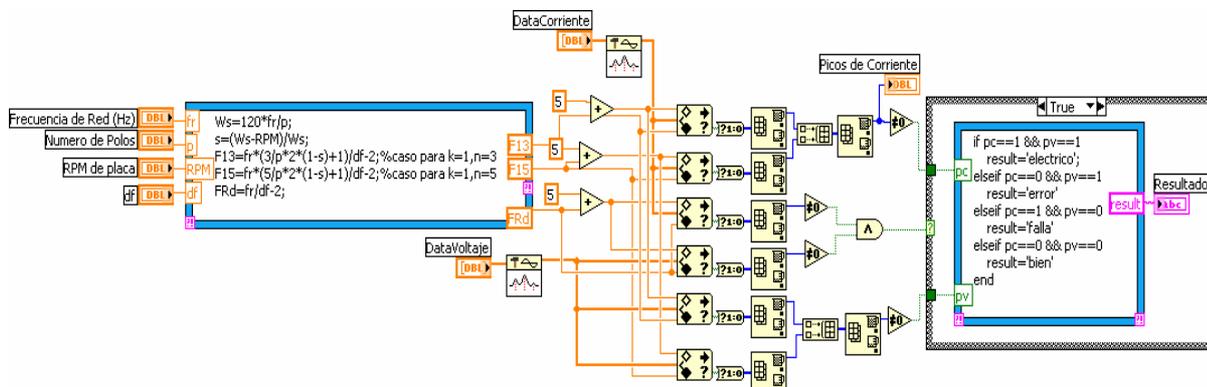


Figura 3.19. Implementación MCSA

Lo que la función “En rango” devuelve como resultado es un arreglo de booleanos en el que cada elemento es 0 si el pico detectado en el correspondiente al arreglo de picos no cae entre los intervalos explicados anteriormente y 1 en el caso contrario.

Después, se detecta el máximo de cada uno de estos arreglos de booleanos de tal manera que este será 1 si efectivamente hubo un pico en la frecuencia deseada o 0 si no lo hubo.

Si el programa detecta un pico en la banda central, se procede a la detección, si no, es que ha existido un error en el procesamiento o en la conexión del hardware. Por lo que se enviará un mensaje de error, según la Tabla 3.6.

3.8 DETECCIÓN DE LA FALLA

Como se trató anteriormente, la sección 2.5 da el criterio de detección de la falla. De aquí que en la implementación de estos criterios, que se hace en la misma función MOTOR, se toman en cuenta como entradas una estructura tipo “case” cuyo controlador es un booleano que indica 1 si existió un pico en la frecuencia fundamental con lo cual se permite la detección según el criterio de MCSA de voltaje y corriente de la sección 2.5. Si el booleano indica 0, se envía un mensaje de error a la variable “resultado”.

Al lazo de código entran dos cantidades booleanas que dicen si existió o no un pico en las frecuencias de interés y escriben un mensaje en la variable “resultado” según la Tabla 3.5.

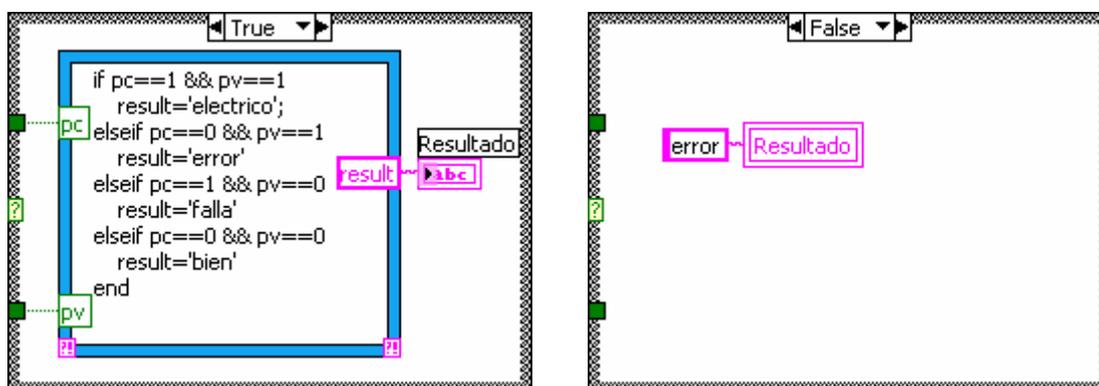


Figura 3.20. Implementación de Detección de falla

En este código se consideran todas las posibilidades que puedan existir, y además se asegura que haya una discriminación entre un proceso correcto, con datos coherentes y uno con datos erróneos o problemas eléctricos.

Tabla 3.5. Criterio de selección de falla

Pico de corriente	Pico de voltaje	Salida
1*	1	eléctrico
0	1	error
1	0	falla
0	0	bien

* 1: Sí se presentó pico, 0: no se presentó pico

La variable “resultado” entra o a una función que se encarga de generar los parámetros de los reportes (sección 2.7) si este se ha elegido para crearse, o si no, a una subrutina más simple de escritura. En ambos casos, el programa escribe el diagnóstico y una recomendación en el panel frontal de acuerdo a la siguiente tabla:

Tabla 3.6. Diagnósticos y recomendaciones que muestra el equipo.

Salida	Diagnóstico a mostrar	Recomendación a mostrar
“error”	Existe un error en el proceso de detección.	Revisar el estado de la red eléctrica y las configuraciones de hardware y software.
“electrico”	Es probable que exista un error en la red eléctrica.	Revisar la red eléctrica a la que esta conectada el motor.
“falla”	El motor tiene bobinas cortocircuitadas en su interior.	Sacar de operación el motor y llevarlo a mantenimiento.
“bien”	El motor se encuentra en buen estado..	No se necesita tomar ninguna acción.

3.9 ALMACENAMIENTO DE DATOS

Cumpliendo con los criterios de la sección 2.6.1, se debe almacenar las siguientes variables:

- Variación de Tiempo.
- Señal de corriente en el dominio del tiempo.
- Señal de voltaje en el dominio del tiempo.
- Variación de Frecuencia.
- Señal de corriente en el dominio de la frecuencia.
- Señal de voltaje en el dominio de la frecuencia.

en las siguientes aplicaciones:

- En hoja de cálculo.
- En base de datos estandarizada.

En lo que concierne a hojas de cálculo, la aplicación será MS EXCEL, ya que es la más común entre los sistemas sobre Windows, mientras que en base de datos se han elegido dos aplicaciones: MS Access y MySQL.

El usuario debe elegir en qué aplicación almacenar los datos, o simplemente obviar este proceso.

3.9.1 ALMACENAMIENTO EN MS EXCEL

LabVIEW tiene una librería que permite la comunicación y el almacenamiento de los datos adquiridos en una hoja de cálculo, y

especificando su nombre, y con la extensión *.xls se completa el proceso de creación de un documento en MS Excel.

El código está implementado como sigue:

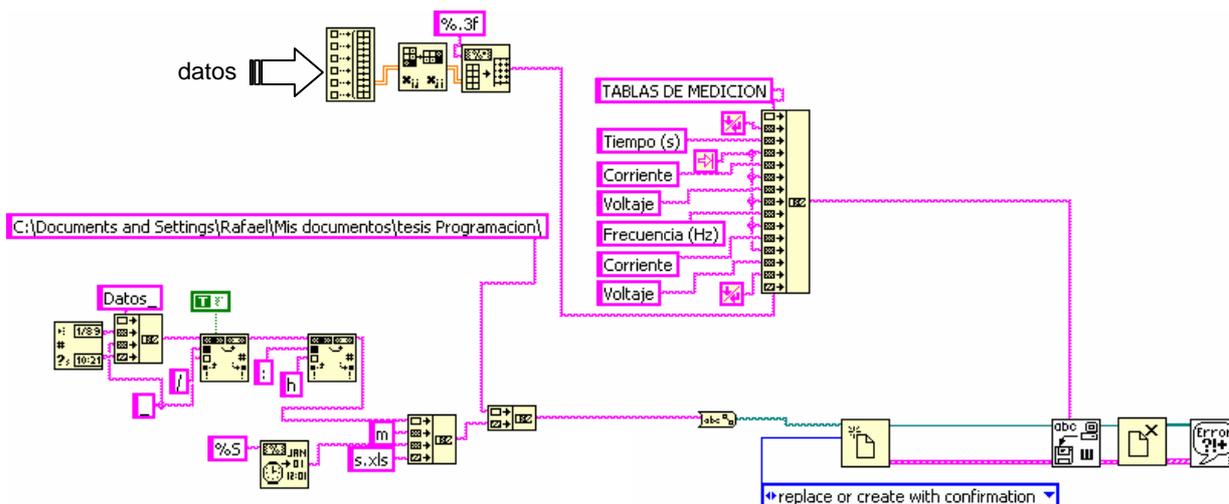


Figura 3.21. Implementación de almacenamiento en Excel

Se crean los títulos de las columnas, se llenan las filas y las columnas, se establece una dirección dinámica que crea archivos asegurándose que siempre tenga un nombre distinto para evitar conflicto, y por último se cierra el archivo.

El formato del nombre del archivo es:

Datos_dd_mm_aa_HHhMMmSSs.xls donde *dd*, *mm* y *aa* son el día, mes y año seguido de los enteros respectivos de hora, minutos y segundos al tiempo de ejecución.

Debido a la rapidez del proceso, se pueden enviar los datos adquiridos en su totalidad.

3.9.2 ALMACENAMIENTO EN BASE DE DATOS MS ACCESS

LabVIEW tiene una librería que permite la comunicación directa con cualquier base de datos estandarizada, dependiendo del Nombre de Origen de Datos (*Data Source Name DSN*) instalado en el Administrador de Orígenes de Datos ODBC.

La configuración está descrita en la Figura 3.22, en donde aparece creada el DSN *DatosNumericosAccess*, entre otros, que comunica a cualquier aplicación con la Base de Datos llamada de la misma forma.

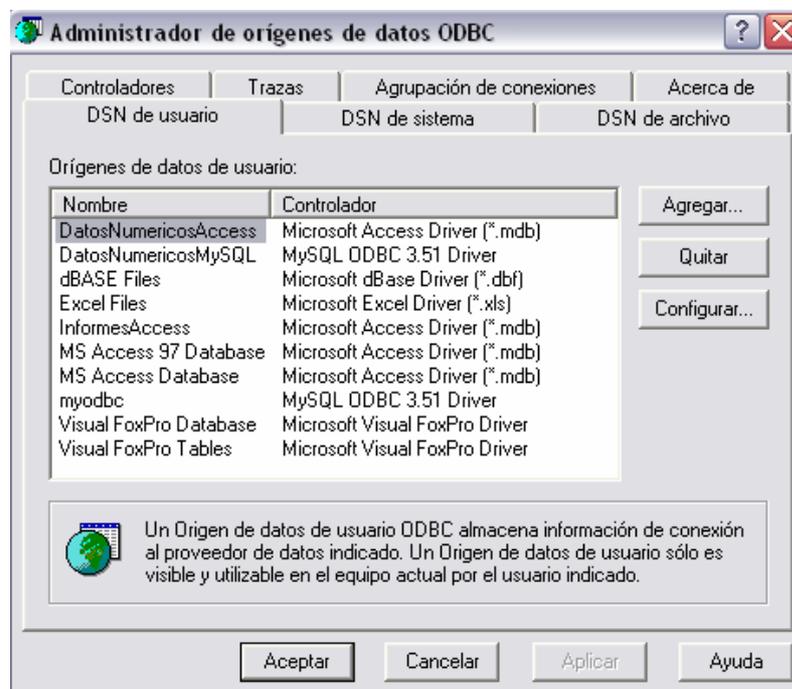


Figura 3.22. Administrado de orígenes de datos configurado para el proyecto.

La implementación en LabVIEW se hace según la Figura 3.23, a través de una secuencia en donde se especifica el DSN al inicio de la subrutina, eso quiere decir que se trabaja en la Base de Datos llamada *DatosNumericosAccess*. Dentro de esta, se crea una tabla con nombre

dinámico, de manera similar como se nombró las tablas en MS Excel. Una vez creada la tabla, se especifican los nombres de las columnas y se empiezan a almacenar los datos a través de un lazo. Debido a que son bastantes los datos a recopilar, la comunicación desde una aplicación a través de ODBC es relativamente lenta y por último no hay la necesidad de almacenar todos los datos, se envía el número de muestras correspondientes al intervalo de 0-500 Hz (el número de datos depende del espaciamiento en frecuencia).

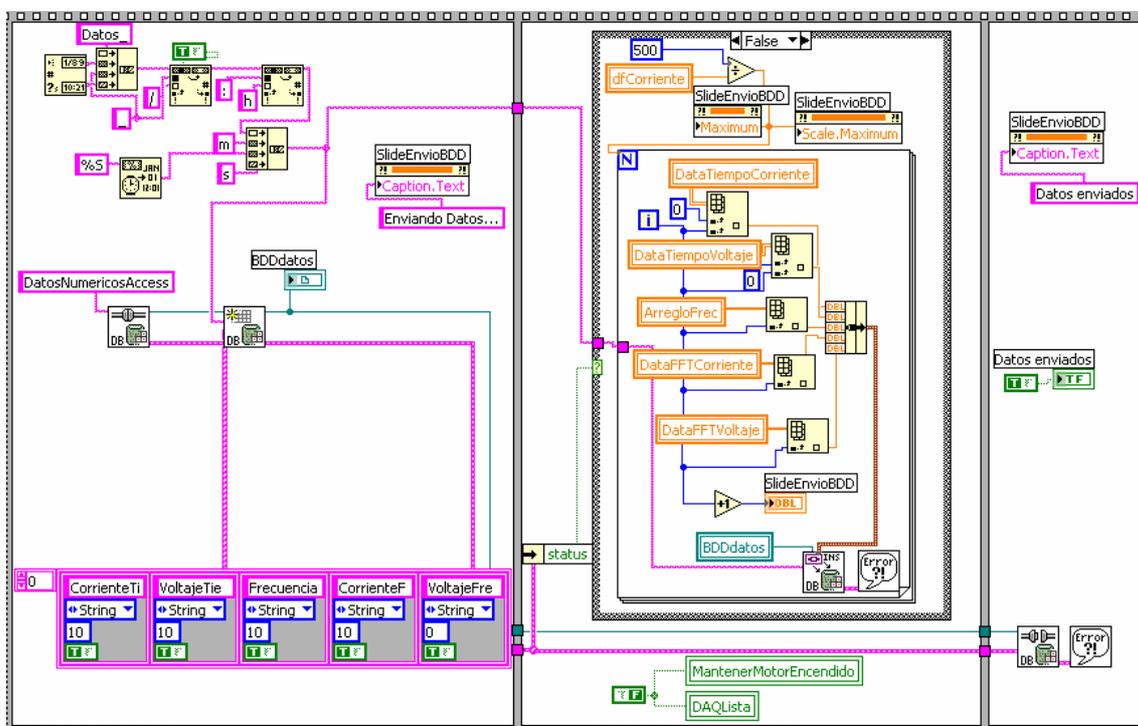


Figura 3.23. Implementación de almacenamiento de datos en MS Access

3.9.3 ALMACENAMIENTO EN BASE DE DATOS MySQL.

El objetivo de realizar un almacenamiento en MySQL es para que el sistema sea más versátil y pueda tener comunicación con sistemas más robustos que Windows, como lo es Linux o UNIX, además de poder integrar el sistema a otro basado en redes seguras con la posibilidad de que el proyecto sea parte de un sistema SCADA o DCS[36].

Para esta implementación se ha instalado un controlador especial de un DSN para Linux. Este comunica al *host* del proyecto con una computadora externa a través de una red de área local LAN en donde reside la base de datos en MySQL y está previamente creada la Base de datos: *DatosNumericos*.

Las direcciones de red para esta configuración deben ser las siguientes:

- *Host* de proyecto: 192.168.1.80, estática.
- *Host* de Base de datos: 192.168.1.90, estática.

Esto debe ser configurado a través de las propiedades de LAN del computador en el que se esté trabajando, mientras que la implementación en LabVIEW es similar al caso anterior, y toma las mismas consideraciones, pero se cambia el nombre de DSN a utilizar:

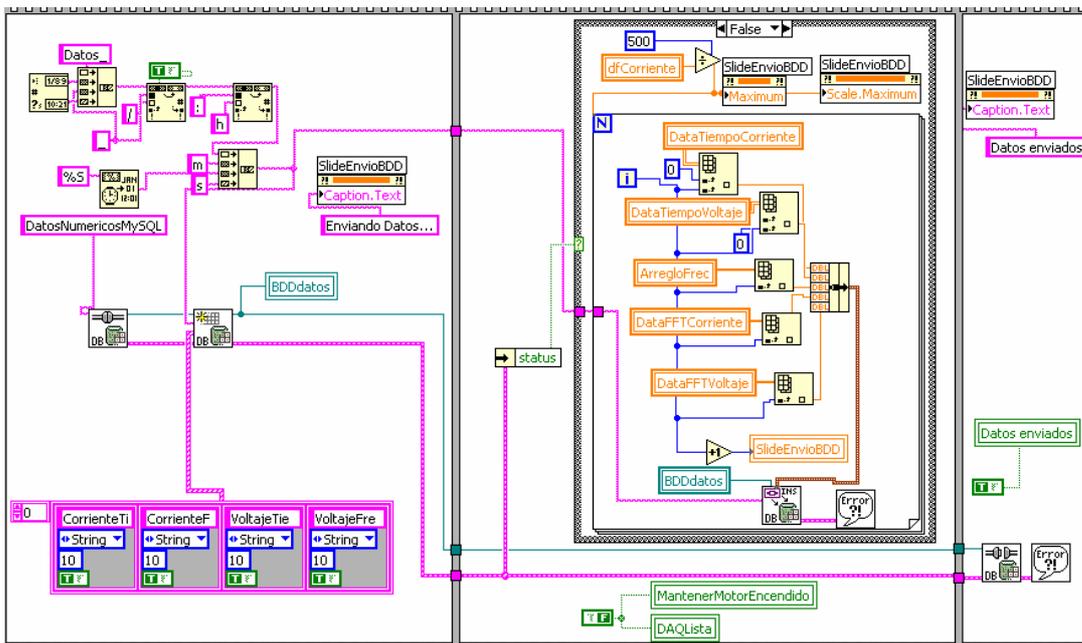


Figura 3.24. Implementación de almacenamiento de datos en MySQL

En todas las subrutinas anteriores se han implementado señales visuales que muestran el estado del porcentaje del almacenamiento a través de una barra de progreso en el panel frontal.

3.10 GENERACIÓN DE REPORTES

3.10.1 IMPLEMENTACIÓN

La mejor manera de implementar la generación de reportes es a través de DIAdem 9.0 que es un software especializado en esta tarea y está totalmente integrado con LabVIEW.

En primer lugar se ha diseñado una plantilla con la interfase gráfica de DIAdem, en la que se incluye:

- Gráfica general para las formas de onda en dominio del tiempo para corriente y voltaje.
- Gráfica general para las formas de onda de corriente en frecuencia.
- Gráfica general para las formas de onda de voltaje en frecuencia.
- Diagnóstico.
- Entrada de texto para incluir los siguientes datos del motor analizado.
 - Frecuencia de Red.
 - RPM Real.
 - Número de Polos.
- Recomendación.
- Lugar adonde se enviaron los datos.
- Fecha y hora de la ejecución del proceso.

El archivo es generado en un documento de tipo *.pdf asignado siempre con un nombre dinámico generado como se explicó en la sección 3.9.2 de almacenamiento de datos y cuyo formato es el siguiente:

Reporte_dd_mm_aa_HHhMMmSSs.pdf donde *dd*, *mm* y *aa* son el día, mes y año seguido de los enteros respectivos de hora, minutos y segundos al tiempo de ejecución.

La plantilla es el archivo *STplantillareporte.tdr* y se muestra a continuación:

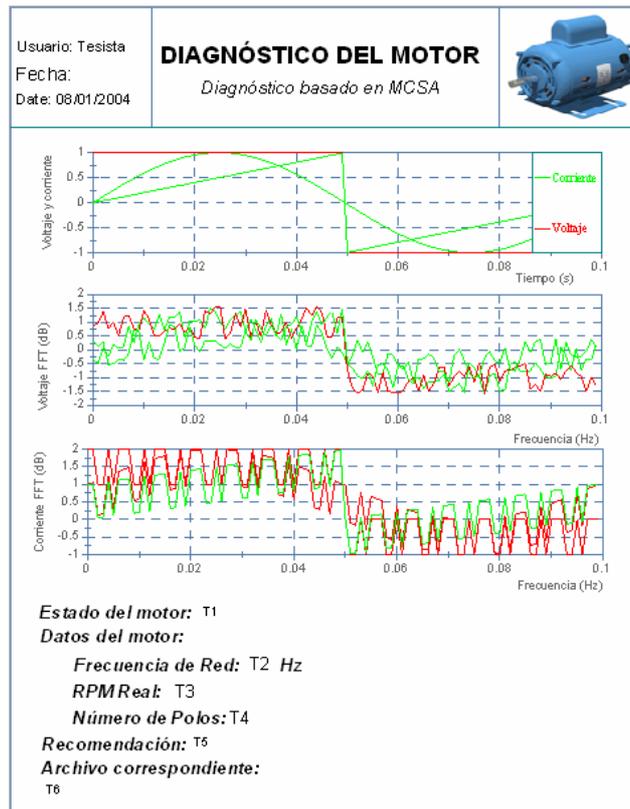


Figura 3.25. Plantilla de reportes

Esta implementación se realiza a través de la Función *Diadem Report* precompilada en LabVIEW:



Figura 3.26. Implementación de generación de reportes

En las entradas *signals 1-3* van las formas de señal entre 0 y 300 Hz, en *Text values* entra una matriz que reemplaza en la plantilla los valores T1 a T6 durante la ejecución por sus respectivos correspondientes numéricos.

Para que esto se haga dinámicamente se implementa una función llamada *Reporte* que funciona de la siguiente manera:

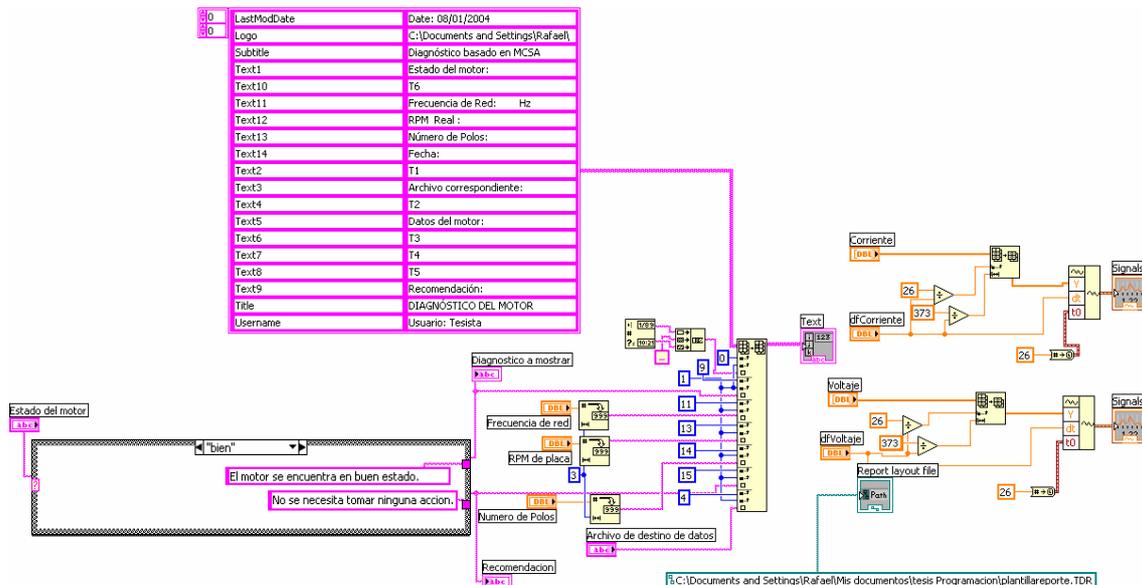


Figura 3.27. Función *Reporte*

Las entradas de esta función son: la variable de salida de la función MOTOR, frecuencia de red, RPM real, número de polos y dirección de almacenamiento de datos. Reemplazando los valores de la celda de una matriz, que contiene los parámetros de la plantilla predefinida con los actuales, se obtiene una manera dinámica de asignarlos. Además, se asignan también los rangos de las señales en frecuencia que se enviarán al reporte.

Las salidas serán: la matriz de texto con datos actuales, las señales requeridas ya acotadas, la dirección de referencia de la plantilla, el diagnóstico y la recomendación a mostrarse en el panel de control.

3.11 VISIÓN GENERAL DEL SISTEMA

El sistema tiene una interfaz basada en cuatro paletas:

- Configuración.
- Procesamiento y señales de tiempo.
- Señales de frecuencia y diagnóstico.
- Tablas.

Cada una de ellas tiene una función especial y está diseñada para que sea confortable al usuario y fácil de manejar.

3.11.1 INTERFAZ GRÁFICA

Paleta “Configuración”.

La paleta “Configuración” es la pantalla principal de la aplicación y su apariencia se encuentra en la Figura 3.28.

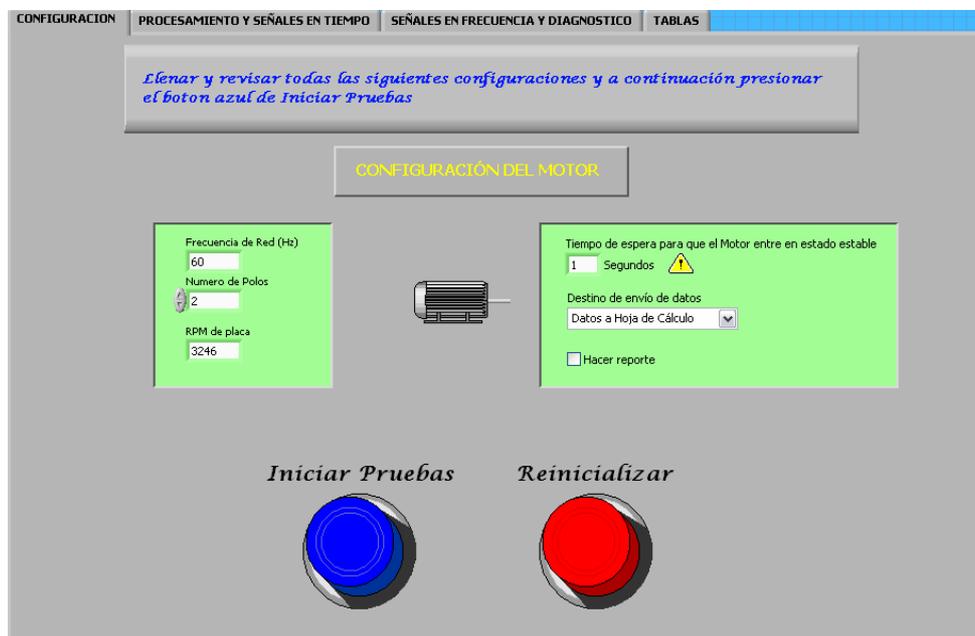


Figura 3.28. Pantalla “Configuración”

Aquí se encuentra una interfaz en la que el usuario ingresa los datos particulares del motor a diagnosticar. Estos son:

- Frecuencia de red. Predeterminado en 60 Hz.
- Número de polos. Predeterminado en 4.
- Tiempo de espera para que el motor entre en estado estable. Predeterminado en 2 segundos.
- RPM real del motor. Predeterminado en 1765
- Destino de envío de Datos. Con las opciones: Datos a Hoja de Cálculo, Datos a Access, Datos a MySQL y No enviar (Predeterminado).
- Opción para hacer o no Reporte.

Además, tiene dos botones principales, uno azul que inicia las pruebas, y uno rojo que reinicializa todo el programa a los valores predeterminados.

Paleta “Procesamiento y señales de tiempo”.

En esta paleta se muestra en un principio, el avance en el proceso de la detección a través de tres barras de estado que parpadean o se llenan según la tarea que se esté ejecutando en algún momento. Estas son:

- Estado Lectura de Datos. (Parpadeo)
- Estado Envío de Datos. (Llenado)
- Estado Creación de Reporte. (Parpadeo)

Durante y una vez terminadas estas etapas, se escriben mensajes al usuario describiendo el estado de la tarea.

Además, en la parte media-inferior se encuentra un visualizador para mostrar las formas de onda en el dominio del tiempo una vez que se ha terminado de adquirir las señales.

La paleta se muestra a continuación:

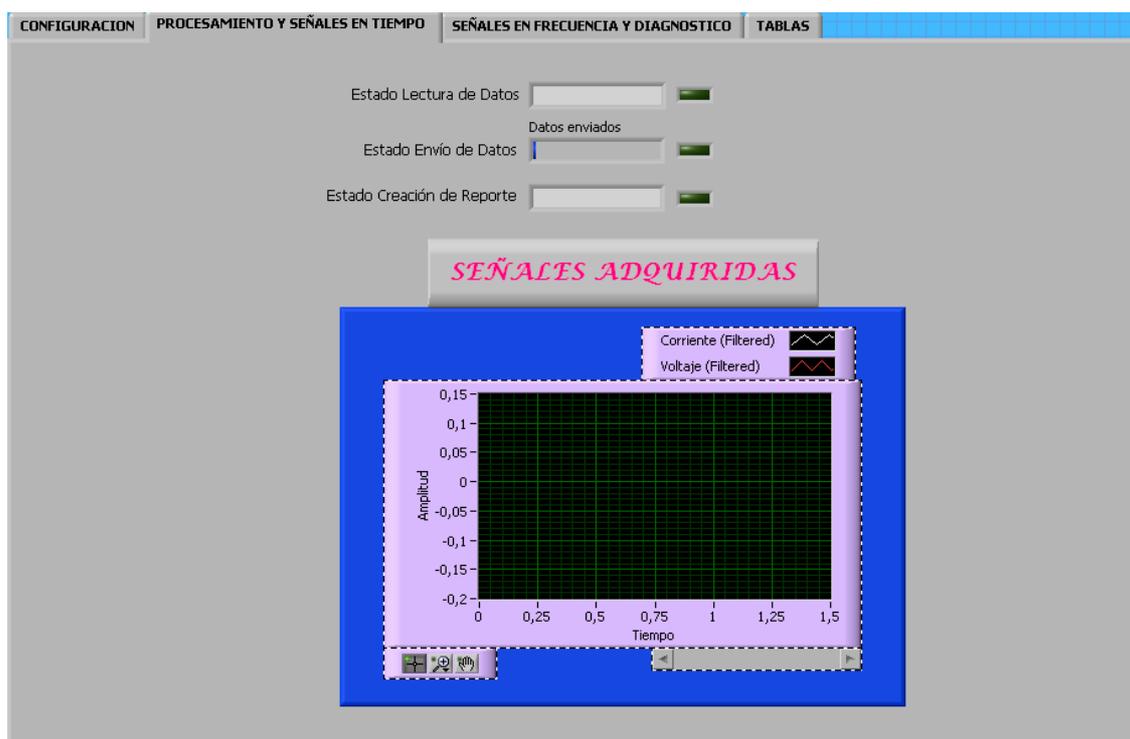


Figura 3.29. Pantalla “Procesamiento y señales en tiempo”

Paleta “Señales de frecuencia y diagnóstico”.

Esta es la paleta que proporciona el resultado mismo del diagnóstico. Aquí se muestran las formas de onda de corriente y voltaje en el dominio de la frecuencia en un rango óptimo para la visualización de las frecuencias de interés (0-300 Hz en donde se ubicarán con seguridad los picos de interés), con la opción de revisar el resto del espectro con una barra de desplazamiento en cada visualizador.

Por otra parte, se muestra el tipo de falla y la recomendación según la Tabla 3.6.

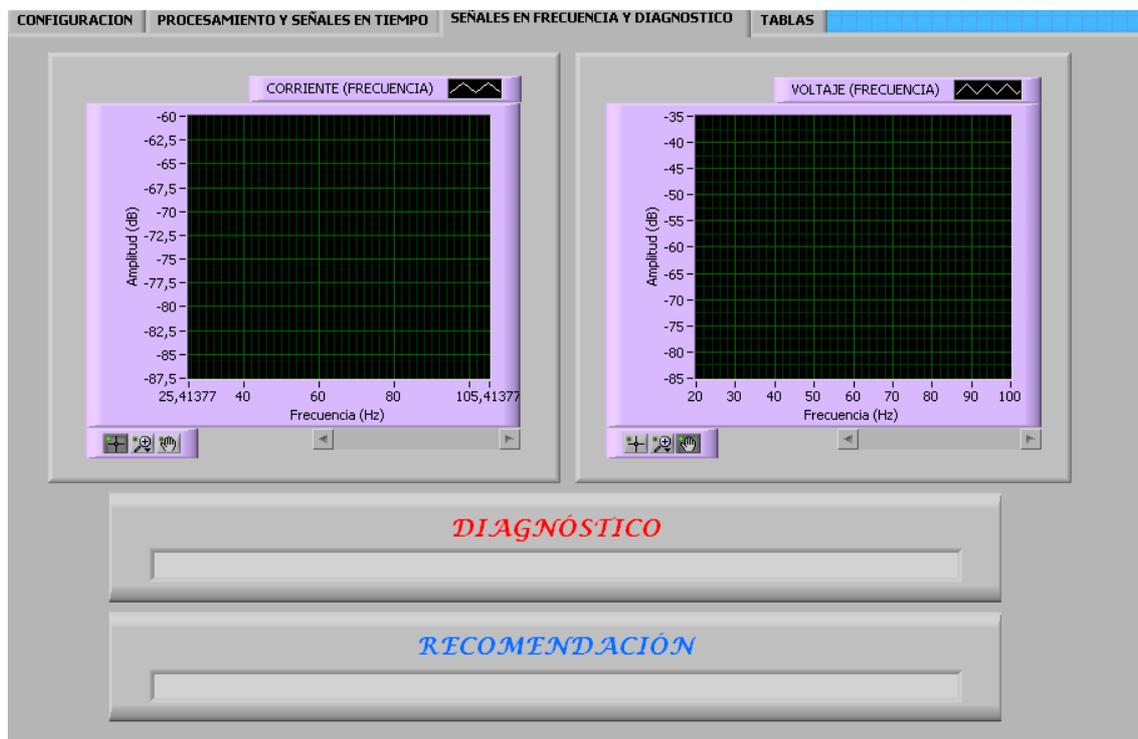


Figura 3.30. Pantalla “Señales en frecuencia y diagnóstico”

Paleta “Tablas”.

Por último, hay una paleta en las que se encuentran las tablas con todos los datos del arreglo de tiempo con sus respectivos valores de voltaje y corriente y lo propio para el dominio de la frecuencia.

El objetivo de esta paleta, es brindar al usuario una manera ágil y rápida de encontrar los valores exactos correspondientes para cierto tiempo o frecuencia de manera que puedan ser utilizados para un eventual cálculo de revisión para constatar que el programa está haciendo lo correcto.

La paleta “Tablas” se muestra en la Figura 3.31.

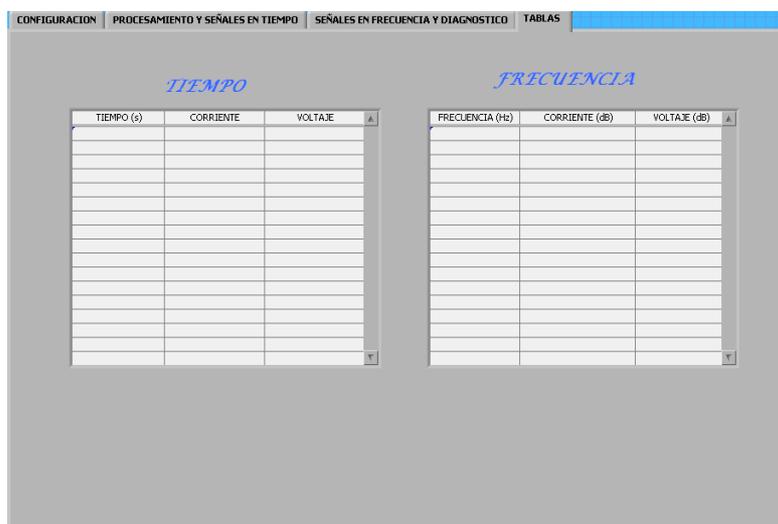


Figura 3.31. Pantalla “Tablas”

Paro de Emergencia.

Como en todo proceso, se necesita una protección de seguridad que suspenda cualquier tarea. El “Paro de Emergencia” es ejecutado a través de la función PARO que aparece y es accesible únicamente cuando se está ejecutando la detección y a la vez, bloquea cualquier acceso al resto de las pantallas. Al instante de presionar este botón, se apaga el motor, se detiene la DAQ, y se detiene cualquier tarea ejecutada por LabVIEW.

Este botón, que cuando aparece lo hace en la parte inferior derecha de la pantalla, se ve de la siguiente manera:



Figura 3.32. Función Paro de Emergencia

3.11.2 FUNCIONAMIENTO GENERAL DEL SOFTWARE

- Se inicia en la paleta “configuración”. El software solo recibe datos de configuración y se mantiene en estado de espera hasta que llegue una orden de arranque de proceso.
- Cuando se presiona el botón “Iniciar” empieza el proceso.
- Se bloquea la pantalla, aparece al mismo tiempo el paro de emergencia.
- Se resetean la DAQ, pantallas, visualizadores, tablas y variables internas.
- Se activa el relevador de protección de filtros y DAQ.
- Se lee los datos.
- Se hace la evaluación de parámetros obtenidos, tareas digitales, detección e identificación de falla.
- Se muestra resultados en pantalla.
- Se almacena los datos.
- Se crea el reporte.
- Se desactiva paro de emergencia.
- Se queda en estado de espera nuevamente.

3.11.3 FUNCIONAMIENTO GENERAL DE HARDWARE

El hardware es implementado bajo el esquema de la Figura 3.33.

La alimentación se realiza a través de un enchufe a la red de 110 o 220V. La energía va a dos transformadores cuyos secundarios dan 24 Vrms con tap central. A continuación, se realiza la rectificación mediante dos puentes de diodos y un filtro de dos capacitores de 470 uF en serie para el uno, y un capacitor de 1000 uF para el otro. Adicionalmente, se pone

reguladores de voltaje de 15, -15, 5 y 12 V. Las dos primeras sirven para la polarización de los amplificadores operacionales de los filtros *anti-aliasing*, mientras que la tercera sirve para la polarización del circuito integrado cambiador de nivel. El último es para el funcionamiento de los relevadores (A,B,C) de protección de las señales adquiridas.

Existe también un contactor (H1) de 10 A de capacidad para encender y apagar motor según la prueba lo requiera pero no es obligatorio que el motor sea controlado por éste. El programa no se interrumpe si el motor nunca interactúa con el contactor, obteniéndose así, la característica de ser un sistema no-invasivo.

Protecciones de Hardware.

Los dispositivos principales a proteger son la DAQ, que es el elemento más delicado del sistema, y los filtros.

En el arranque, el motor tiene una corriente mayor que la nominal, por lo que un sobrevoltaje en la señal de corriente puede estar presente al inicio de la prueba. Si el arranque del motor se sincroniza con la lectura de datos, un sobrevoltaje en los sensores puede ser aislado evitando así que otras etapas se dañen.

A través de dos señales digitales entregadas por la DAQ coordinado por software para que active primero el relevador C que a la vez activa el contactor H1 que enciende en motor. Se espera un cierto tiempo en el que se estima que el transiente ha pasado, (esta es una variable manipulada por el usuario) y se encienden los relevadores B y C, conectados en paralelo que cierran el circuito de conexión entre los sensores y los filtros. Cuando estos dos se activan, también se activan señales visuales a través de LEDs

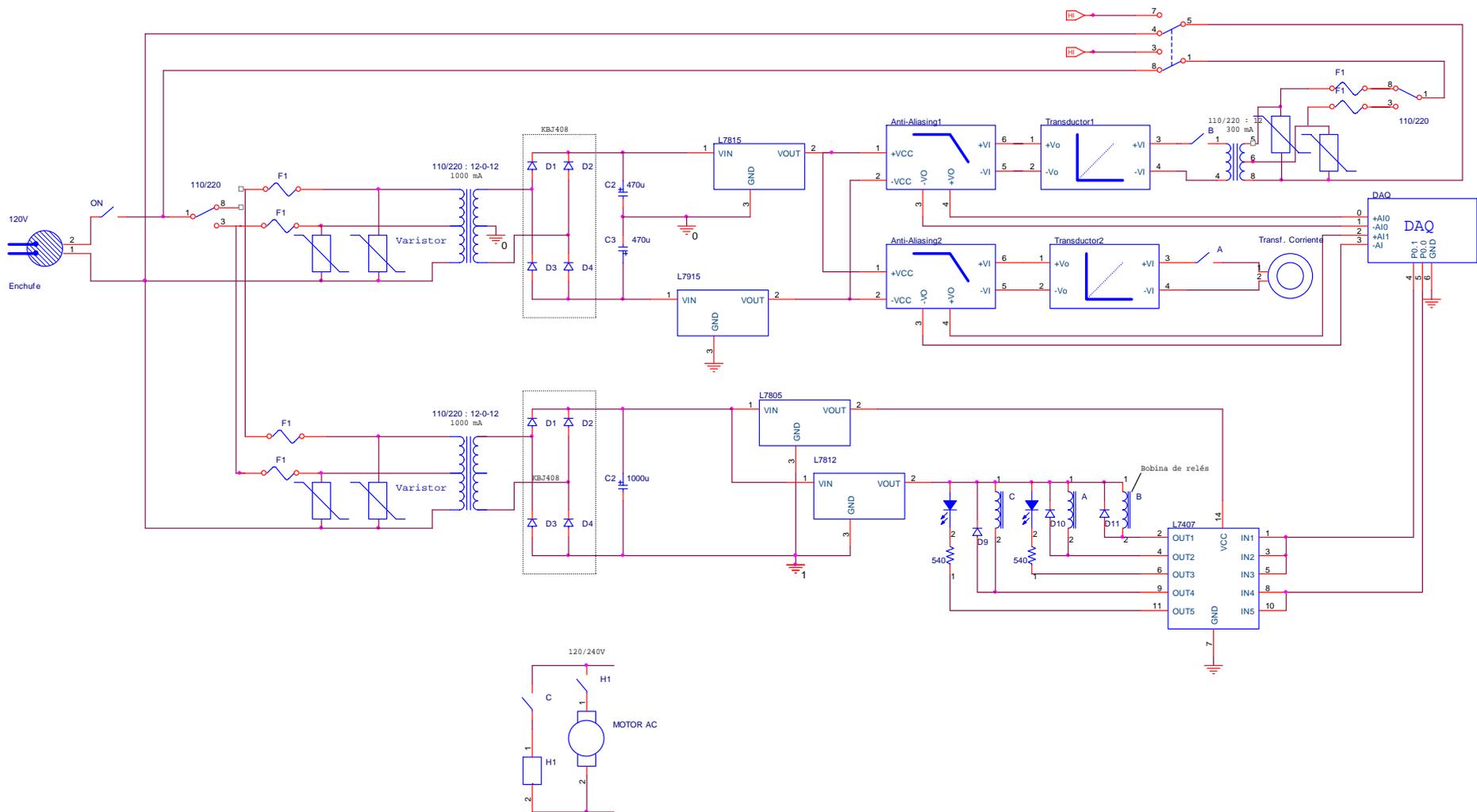


Figura 3.33. Esquema general del Hardware

Cuando la prueba ha concluido, se desconecta primero los relevadores A y B, y después el C.

En vista de que la DAQ no maneja una corriente suficiente como para encender un relevador, se ha utilizado un cambiador de nivel TTL que funciona como búfer. Los relevadores toman su corriente desde los reguladores de voltaje de 12 V.

En el sistema también existen protecciones de sobrevoltaje y sobrecorriente con fusibles y un varistor a la entrada de energía.

Los elementos usados son los siguientes:

Tabla 3.7. Elementos utilizados en la implementación.

Función	Elemento	Características principales
Varistor	CNR20D241k	Diseñado para actuar en picos de 200 V
Fusibles		250 mA
Puente de diodos	KBJ408	
Regulador 15V	7815	Corriente máxima: 1 A
Regulador -15V	7915	Corriente máxima: 1 A
Regulador 12V	7812	Corriente máxima: 1 A
Regulador 5V	7805	Corriente máxima: 1 A
Relevador	4123-(T71)	Bobina 12V, contacto 120/240V maneja 10/6A
Contactador H1		10 Amperios.
Búfer	7407	Maneja salida de colector abierto hasta de 40 mA.