

DESARROLLO DE SOFTWARE PARA MEJORAR LA OBTENCIÓN DE LA  
RESPUESTA EN FRECUENCIA EN UN TRASFORMADOR PILOTO  
ENERGIZADO.

JOHN EDWIN GAVIRIA GONZALEZ

UNIVERSIDAD DEL VALLE  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA  
ÁREA DE SISTEMAS DE POTENCIA  
SANTIAGO DE CALI  
2016

DESARROLLO DE SOFTWARE PARA MEJORAR LA OBTENCIÓN DE LA  
RESPUESTA EN FRECUENCIA EN UN TRASFORMADOR PILOTO  
ENERGIZADO.

JOHN EDWIN GAVIRIA GONZALEZ

Proyecto de grado  
presentado como requisito para optar al título de  
INGENIERO ELECTRICISTA.

DIRECTORES:

Ing. GUILLERMO APONTE MAYOR M.Sc. Ph.D.  
Ing. EDUARDO GÓMEZ LUNA Ph.D.

UNIVERSIDAD DEL VALLE  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA  
ÁREA DE SISTEMAS DE POTENCIA  
SANTIAGO DE CALI  
2016

## **AGRADECIMIENTOS**

Primero que nada agradecer a Dios quien me dio la fuerza y fortaleza para permitir llegar al final de una de las etapas más importantes de mi vida personal y profesional.

A mis padres Hector Fabio Gaviria, Maria Gonzalez; y hermana Diana Katherine Gaviria, quienes con su apoyo incondicional, esfuerzo, sacrificio y amor, hicieron posibles que saliera adelante.

A familiares y amigos por el apoyo incondicional, y motivación durante el desarrollo de mis estudios universitarios.

A mis directores de trabajo de grado, por su acompañamiento, paciencia y tiempo invertido en asesorías para terminar satisfactoriamente con este proyecto.

Al ingeniero Christian Fabian Guerrero Chaves, por su amabilidad y tiempo invertido en asesorías, para contribuir en el desarrollo de este proyecto, permitiéndome utilizar material de su propiedad.

Al personal del laboratorio de alta tensión GRALTA, por permitir servir de apoyo y prestar sus equipos para el desarrollo de las pruebas.

A los profesores de la Universidad, en especial a los de la Escuela de Ingeniería Eléctrica y Electrónica por servir de guías, promover la disciplina y brindar los conocimientos necesarios durante mi etapa formativa en la universidad.

Un agradecimiento especial a mis compañeros de Ingeniería Eléctrica por su apoyo y amistad durante el desarrollo de la carrera.

A la Universidad del Valle, por su misión de llevar la educación superior de excelente calidad a los sectores menos favorecidos de la comunidad.

## RESUMEN

El trabajo de grado presentado trata acerca del desarrollo de una aplicación de software, para mejorar la obtención de la respuesta en frecuencia en un transformador piloto energizado. Dentro del estudio realizado se inicia con una recopilación de información acerca de las generalidades del método “Análisis de Respuesta en Frecuencia (FRA)”, como base teórica del presente trabajo; en el que se incluye la descripción del método tradicional y las tendencias en instrumentación virtual para optimizar este método. Luego de esto se realiza una recopilación de las alternativas de software, para desarrollar la aplicación con instrumentación virtual, así pues con la información recolectada se realiza la selección del software, y posteriormente se describen sus principales herramientas y entorno de trabajo. Luego se realiza la descripción del desarrollo de la aplicación y se utiliza una herramienta computacional desarrollada en Matlab, para validar la aplicación hecha, además se realiza un análisis de los resultados obtenidos entre la aplicación desarrollada y la aplicación de Matlab. Por último se entregan conclusiones, trabajos futuros, y una guía de usuario para el manejo de la aplicación software.

**Palabras Clave:** FRA, Instrumentación Virtual, Transformada Wavelet, Transformador Piloto, Labview, Matlab.

## ABSTRACT

The showed degree work expose the development of a software application, to improve the obtaining of frequency response in a pilot transformer energized. The study starts by gathering information about the generalities of the method “frequency response analysis (FRA)”; it includes the description of the traditional method and the trends in virtual instrumentation to optimize this method. After this, a gather of information of software alternatives is done to develop the application with virtual instrumentation, subsequently with the information collected the software selection is done, and then the description of the main tools and work environment. The next step is the development of the application and the validation of it through a computational tool developed in Matlab. Besides is done an analysis of the results obtained between the application developed in Matlab and the application made in this degree work. Finally, the conclusions, future works and a user guide for managing the software application are delivered.

**Keywords:** FRA, Virtual Instrumentation, Wavelet Transformed, Pilot Transformer, Labview, Matlab.

## CONTENIDO

pág.

|                                                                                                              |    |
|--------------------------------------------------------------------------------------------------------------|----|
| 1. GENERALIDADES DE LA RESPUESTA EN FRECUENCIA CON EL TRANSFORMADOR EN SERVICIO.....                         | 11 |
| 1.1 La tecnica FRA. ....                                                                                     | 11 |
| 1.2 La técnica IFRA .....                                                                                    | 13 |
| 1.3 La técnica SFRA.....                                                                                     | 13 |
| 1.4 El metodo FRA on-line.....                                                                               | 13 |
| 1.5 El metodo IFRA on-line.....                                                                              | 14 |
| 1.5.1 Etapa 1. ....                                                                                          | 15 |
| 1.5.2 Etapa 2. ....                                                                                          | 16 |
| 1.5.3 Etapa 3. ....                                                                                          | 16 |
| 1.6 Metodo IFRA on-line empleando Instrumentación virtual .....                                              | 17 |
| 1.6.1 Metodología propuesta para el desarrollo del método IFRA on-line con instrumentación virtual. ....     | 18 |
| 1.6.2 Especificaciones técnicas de los equipos para el método IFRA on-line con instrumentación virtual. .... | 19 |
| 1.6.3 Sistema de medición y adquisición de señales.....                                                      | 20 |
| 2. SOFTWARE PARA LA OBTENCIÓN DE LA CURVA FRA DE MANERA DINAMICA .....                                       | 23 |
| 2.1 Alternativas de Software programables para la prueba IFRA on-line.....                                   | 24 |
| 2.1.1 Scada. ....                                                                                            | 24 |
| 2.1.2 Delphi. ....                                                                                           | 25 |
| 2.1.3 Neplan. ....                                                                                           | 25 |
| 2.1.4 Powerworld. ....                                                                                       | 26 |
| 2.1.5 Matlab.....                                                                                            | 27 |
| 2.1.6 Labview. ....                                                                                          | 27 |
| 2.2 Labview, software seleccionado para desarrollar la prueba IFRA on-line                                   | 28 |
| 2.3 Plataforma para instrumentación virtual labview.....                                                     | 29 |
| 2.3.1 Panel frontal.....                                                                                     | 29 |
| 2.3.2 Diagrama de bloques.....                                                                               | 30 |

|                                                                                                        |    |
|--------------------------------------------------------------------------------------------------------|----|
| 2.3.3 Paletas de herramientas.....                                                                     | 31 |
| 2.3.4 Paletas de controles. ....                                                                       | 31 |
| 2.3.5 Barra de herramientas del panel frontal.....                                                     | 33 |
| 2.3.6 Paleta de funciones .....                                                                        | 33 |
| 2.3.7 Barra de herramientas del diagrama de bloques .....                                              | 34 |
| 2.3.8 Clases de datos en Labview. ....                                                                 | 34 |
| 2.3.9 Estructuras de programación de Labview.....                                                      | 35 |
| 2.3.10 Programación basada en texto. ....                                                              | 36 |
| 3. DESARROLLO DE LA APLICACIÓN SOFTWARE PARA EL MÉTODO IFRA<br>ON-LINE .....                           | 39 |
| 3.1 etapas implementadas del desarrollo de la aplicación software para el<br>método IFRA on-line ..... | 40 |
| 3.1.1 Etapa adquisición de datos.....                                                                  | 40 |
| 3.1.2 Etapa de filtrado.....                                                                           | 49 |
| 3.1.3 Ventana de análisis. ....                                                                        | 54 |
| 3.1.4 Procesamiento de datos. ....                                                                     | 57 |
| 4. DESARROLLO DE PRUEBAS Y ANALISIS DE RESULTADOS.....                                                 | 64 |
| 4.1 Realización de pruebas. ....                                                                       | 67 |
| 4.1.1 Prueba 1. ....                                                                                   | 67 |
| 4.1.2 Prueba 2. ....                                                                                   | 71 |
| 4.1.3 Prueba 3. ....                                                                                   | 73 |
| 4.1.4 Prueba 4. ....                                                                                   | 74 |
| 4.1.5 Prueba 5. ....                                                                                   | 76 |
| 4.1.6 Prueba 6. ....                                                                                   | 78 |
| 5. CONCLUSIONES Y TRABAJOS FUTUROS.....                                                                | 80 |
| 5.1 Conclusiones .....                                                                                 | 80 |
| 5.2 Trabajos futuros .....                                                                             | 81 |
| 6. BIBLIOGRAFÍA.....                                                                                   | 82 |

## LISTA DE FIGURAS

Pág.

|            |                                                                                |    |
|------------|--------------------------------------------------------------------------------|----|
| Figura 1.  | Respuesta en Frecuencia del transformador antes y después de algún evento..... | 12 |
| Figura 2.  | Metodología implementada para la técnica IFRA-online.....                      | 14 |
| Figura 3.  | Esquema del circuito de inyección de pulso para el método IFRA on-line.....    | 15 |
| Figura 4.  | Módulo LC para la realización de ensayos en sistemas de 15Kv. ....             | 16 |
| Figura 5.  | Esquema del método IFRA on-line con medición de las señales.....               | 16 |
| Figura 6.  | Esquema del método IFRA on-line con instrumentación virtual .....              | 18 |
| Figura 7.  | Diagrama esquemático del transformador piloto .....                            | 19 |
| Figura 8.  | Punta de tensión Tektronix P2220. ....                                         | 20 |
| Figura 9.  | Bobina Rogowski.....                                                           | 20 |
| Figura 10. | Esquema de conexión de la tarjeta national instruments PCI 6115 ...            | 22 |
| Figura 11. | Panel frontal o interfaz de Labview .....                                      | 30 |
| Figura 12. | Diagrama de bloques de Labview .....                                           | 30 |
| Figura 13. | Paleta de herramientas .....                                                   | 31 |
| Figura 14. | Paletas de controles.....                                                      | 32 |
| Figura 15. | Control e indicador numérico .....                                             | 32 |
| Figura 16. | Control e indicador booleano .....                                             | 32 |
| Figura 17. | Controles e indicadores de cadena de caracteres .....                          | 32 |
| Figura 18. | Control e indicador gráfico .....                                              | 33 |
| Figura 19. | Barra de herramientas del panel frontal .....                                  | 33 |
| Figura 20. | Paleta de funciones.....                                                       | 34 |
| Figura 21. | Barra de herramientas del diagrama de bloques .....                            | 34 |
| Figura 22. | Clases de datos en Labview .....                                               | 35 |
| Figura 23. | Estructuras de programación en Labview .....                                   | 36 |
| Figura 24. | Formula node .....                                                             | 37 |
| Figura 25. | Mathscript Node .....                                                          | 38 |
| Figura 26. | Matlab script Node .....                                                       | 38 |
| Figura 27. | Secuencia de de bloques VIs que contiene la etapa de adquisición de datos..... | 41 |
| Figura 28. | DAQmx Create Task.VI.....                                                      | 41 |
| Figura 29. | DAQmx Create Channel (AI-Voltage-Basic).VI .....                               | 42 |
| Figura 30. | DAQmx Timing (Sample Clock).VI .....                                           | 44 |
| Figura 31. | DAQmx Control Task.VI.....                                                     | 46 |
| Figura 32. | DAQmx Read (Analog 1D Wfm NChan NSamp).....                                    | 47 |
| Figura 33. | DAQmx Wait Until Done.VI .....                                                 | 48 |
| Figura 34. | DAQmx Stop Task.VI .....                                                       | 49 |
| Figura 35. | DAQmx Clear Task.VI .....                                                      | 49 |
| Figura 36. | Secuencia de bloques VIs que contiene la etapa de filtro.....                  | 50 |
| Figura 37. | WA Detrend (Waveform).VI.....                                                  | 50 |
| Figura 38. | WA Denoise VI.....                                                             | 52 |

|            |                                                                                        |    |
|------------|----------------------------------------------------------------------------------------|----|
| Figura 39. | Dialog Box Options del Trigger and Gate Express. VI.....                               | 55 |
| Figura 40. | Trigger and Gate express. VI .....                                                     | 55 |
| Figura 41. | Convert from Dynamic Data .....                                                        | 57 |
| Figura 42. | Convert to Dynamic Data .....                                                          | 58 |
| Figura 43. | Aplicaciones de la transformada wavelet .....                                          | 58 |
| Figura 44. | Representación gráfica del análisis de la transformada wavelet .....                   | 59 |
| Figura 45. | Wavelet Morlet modificada .....                                                        | 60 |
| Figura 46. | Algoritmo para obtener la curva FRA, mediante la herramienta Matlab Script Node.....   | 63 |
| Figura 47. | Build XY Graph. VI .....                                                               | 63 |
| Figura 48. | Interfaz gráfica utilizada para validar la aplicación desarrollada en Labview.....     | 64 |
| Figura 49. | Diagramas Esquemáticos de las pruebas realizadas.....                                  | 67 |
| Figura 50. | Señales de tensión y corriente obtenidas de manera dinámica con Labview, prueba 1..... | 68 |
| Figura 51. | Señales de tensión y corriente obtenidas de manera off-line con Matlab, prueba 1 ..... | 68 |
| Figura 52. | Pulsos de tensión y corrientes filtrados con Labview, prueba 1. ....                   | 69 |
| Figura 53. | Pulsos de tensión y corrientes filtrados con Matlab, prueba 1. ....                    | 69 |
| Figura 54. | Curva FRA obtenida de manera dinámica con Labview, prueba 1.....                       | 70 |
| Figura 55. | Curva FRA obtenida de manera off-line con Matlab, prueba 1. ....                       | 70 |
| Figura 56. | Curvas FRA obtenidas con la aplicación de Matlab y Labview, prueba 1.....              | 70 |
| Figura 57. | Curva FRA obtenida de manera dinámica con Labview, prueba 2.....                       | 71 |
| Figura 58. | Curva FRA obtenida de manera off-line con Matlab, prueba 2. ....                       | 72 |
| Figura 59. | Curvas FRA obtenidas con la aplicación de Matlab y Labview, prueba 2.....              | 72 |
| Figura 60. | Curva FRA obtenida de manera dinámica con Labview, prueba 3.....                       | 73 |
| Figura 61. | Curva FRA obtenida de manera off-line con Matlab, prueba 3. ....                       | 73 |
| Figura 62. | Curvas FRA obtenidas con la aplicación de Matlab y Labview, prueba 3.....              | 74 |
| Figura 63. | Curva FRA obtenida de manera dinámica con Labview, prueba 4.....                       | 75 |
| Figura 64. | Curva FRA obtenida de manera off-line con Matlab, prueba 4. ....                       | 75 |
| Figura 65. | Curvas FRA obtenidas con la aplicación de Matlab y Labview, prueba 4.....              | 76 |
| Figura 66. | Curva FRA obtenida de manera dinámica con Labview, Prueba 5. ....                      | 77 |
| Figura 67. | Curva FRA obtenida de manera off-line con Matlab, prueba 5 .....                       | 77 |
| Figura 68. | Curvas FRA obtenidas con la aplicación de Matlab y Labview, prueba 5.....              | 78 |
| Figura 69. | Curva FRA obtenida de manera dinámica con Labview, prueba 6 ....                       | 78 |
| Figura 70. | Curva FRA obtenida de manera off-line con Matlab, prueba 6 .....                       | 79 |
| Figura 71. | Curvas FRA obtenidas con la aplicación de Matlab y Labview, prueba 6.....              | 79 |

## LISTA DE TABLAS

|                                                                               | Pág. |
|-------------------------------------------------------------------------------|------|
| Tabla 1. Especificaciones técnicas del transformador piloto. ....             | 19   |
| Tabla 2. Especificaciones técnicas de la Bobina Rogowski. ....                | 21   |
| Tabla 3. Especificaciones de la punta de tensión P2220 Tektronix. ....        | 21   |
| Tabla 4. Especificaciones de la tarjeta PCI 6115. ....                        | 21   |
| Tabla 5. Pruebas realizadas con el transformador conectado a 120 V ....       | 66   |
| Tabla 6. Pruebas realizadas con el transformador sin conectarse a 120 V. .... | 66   |

## INTRODUCCIÓN

Los transformadores conforman una de las partes más fundamentales dentro de un sistema de potencia, por lo que es necesario que se mantenga la confiabilidad de estos dentro de la red eléctrica; para una subestación es de vital importancia que el transformador no salga de servicio, ni que presente fallos durante su operación o vida útil; Para ello estos son sometidos a una serie de pruebas de diagnóstico, con el fin de evaluar el estado en que se encuentra el transformador, y poder detectar cualquier anomalía antes de que se genere una falla en este.

Una de las pruebas de diagnóstico, es el análisis de respuesta en frecuencia FRA, (Frequency Response Analysis), la cual evalúa las deformaciones o desplazamiento en las bobinas o núcleo que se lleguen a presentar en el transformador; dicha prueba se realiza con el transformador fuera de servicio, lo que implica que no se realice con frecuencia y solo se haga cuando se tenga cierta sospecha de un posible fallo en el equipo, disminuyendo así, la confiabilidad de estos dentro de la red eléctrica.

Actualmente se han logrado algunos avances en búsqueda de obtener la respuesta en frecuencia con el transformador en servicio, en [1] se realiza la propuesta de un método para obtener la respuesta en frecuencia a partir de señales transitorias, obteniendo excelentes resultados, sin embargo; el procesamiento de datos para obtener la curva no es en tiempo real, es decir; si bien este método obtiene las mediciones con el transformador bajo carga, el procesamiento de estas es off-line, lo que implica que se genere un retardo de tiempo al momento de diagnosticar estos equipos.

Este trabajo de grado se enfoca en proponer una aplicación de software que genere de manera dinámica la obtención de la curva de la respuesta en frecuencia con el transformador en servicio, buscando así, mejorar el método IFRA on-line desarrollado en [1].

Este trabajo de grado, se compone de cinco capítulos, el primer capítulo menciona las tendencias actuales del método FRA e instrumentación virtual, el capítulo dos realiza un breve análisis bibliográfico de las alternativas planteadas para seleccionar el software, además en este capítulo se menciona las principales herramientas y el entorno de trabajo del software seleccionado; en el capítulo tres se presenta la descripción del desarrollo de la aplicación, en el cuatro se presenta la validación teniendo como referencia la aplicación desarrollada en [1], y finalmente en el capítulo cinco se presentan las conclusiones y trabajos futuros.

## **1. GENERALIDADES DE LA RESPUESTA EN FRECUENCIA CON EL TRANSFORMADOR EN SERVICIO**

En la actualidad existen chequeos que se realizan a los transformadores en servicio, como es el análisis de gases disueltos (DGA), análisis de los cambiadores de tap's, emisión acústica (AE), descargas parciales (PD), tangente delta (DT), termografía por infrarrojo y capacitancia (C) [2]. Sin embargo es necesario complementar estas pruebas con las que permitan determinar el estado mecánico de los bobinados y núcleo del transformador.

Una de estas pruebas es el análisis de respuesta en frecuencia FRA (Frequency Response Analysis). es una de las técnicas de diagnóstico más utilizadas para transformadores a nivel mundial, esta técnica se centra en detectar las deformaciones o desplazamientos ocurridos en las bobinas y núcleo del transformador; es decir, dicha técnica es utilizada para la detección de posibles fallas mecánicas ocurridas en el equipo. En la actualidad esta prueba es realizada con la condición de que el transformador debe encontrarse sin carga o fuera de servicio.

Las pruebas realizadas con el transformador fuera de servicio, implican que se generen impactos negativos, los cuales se ven reflejados en los gastos que se dan para la conexión y desconexión del transformador en la red, cortes de energía que sufre un determinado sector al instante en que este sale de servicio, lo que causa que dicha prueba no se realice frecuentemente; y solo se haga cuando se tenga una cierta sospecha de un posible fallo en el transformador.

Con base a lo anterior se aprecia la necesidad de buscar nuevas alternativas para mejorar dicha prueba y así poder garantizar mayor seguridad y confiabilidad en una red eléctrica. Hasta el momento se han realizado algunos avances con el transformador bajo carga; sin embargo el procesamiento de datos es fuera de línea. [1]

A continuación se define en que consiste esta prueba, los métodos existentes para la obtención de resultados bajo esta, los avances que se han logrado para realizar dicha prueba con el transformador energizado; y tendencias en instrumentación virtual para obtener un diagnóstico en tiempo real del estado del transformador conectado a la red.

### **1.1 LA TÉCNICA FRA.**

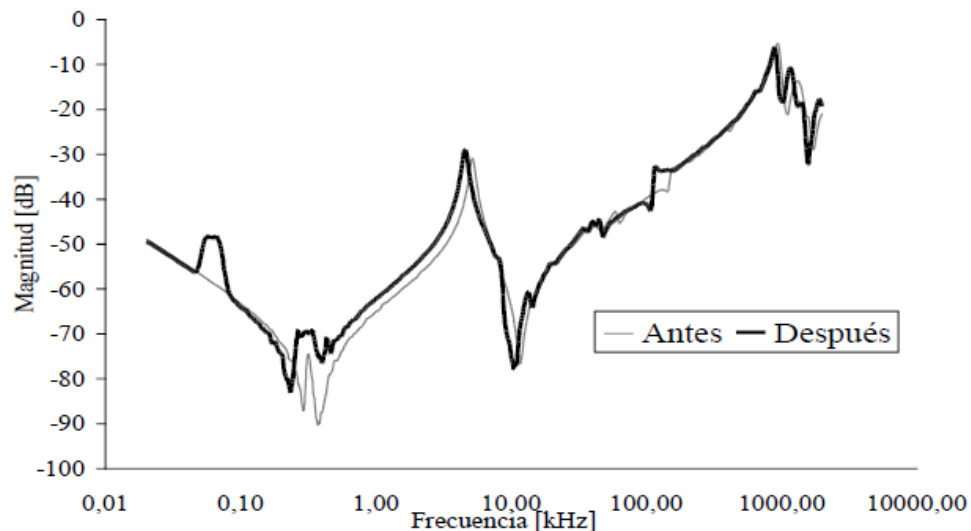
El análisis FRA (Frequency Response Analysis). Es una técnica utilizada comúnmente en la industria para evaluar el estado mecánico de los bobinados y núcleo del transformador, los cuales se pueden ver afectados por un fallo en el

cambiador de tap, un daño generado en el transporte del equipo, o un fallo después de un evento sísmico, entre otros. [3]

La técnica se basa en obtener una curva de referencia, la cual es única para cada transformador, esta se obtiene antes de que el transformador se vea expuesto a algún evento, dicha curva es comparada con otras que se obtienen posteriormente (después del evento), con el fin de percibir cambios, en la figura 1 se muestran dos gráficas, la curva de referencia y la que se obtiene después de haber sometido al transformador a un evento extremo (desplazamiento de sus bobinas o núcleo).

Dichos cambios pueden estar asociados a deformaciones o desplazamientos de los bobinados o núcleo, causando posibles fallos en el funcionamiento del transformador.

Figura 1. Respuesta en Frecuencia del transformador antes y después de algún evento.



Fuente: (Gómez Luna, Obtención de la respuesta en frecuencia de transformadores en servicio a partir de la medida de señales transitorias, 2013)

Existen dos métodos para obtener la curva de impedancia del transformador mediante FRA:

- IFRA (Análisis de la respuesta en frecuencia ante un impulso)
- SFRA (Análisis de la respuesta en frecuencia ante un barrido frecuencial)

## 1.2 LA TÉCNICA IFRA

La técnica IFRA o Análisis de la respuesta en frecuencia ante un impulso, consiste en aplicar un impulso de tensión transitorio en cuestión de microsegundos por uno de los lados del transformador, la obtención del espectro frecuencial se obtiene mediante el cociente, entre la transformada rápida de Fourier (FFT) de la tensión aplicada a la entrada, y la transformada a la intensidad de la corriente medida; para que luego dicha relación pueda ser graficada en función de la frecuencia.

Debido a la naturaleza del método IFRA, el desarrollo de la prueba es muy rápido, contando con solo unos segundos para generar la curva FRA, este método se hace ideal para detectar deformaciones de bobinado incluso en condiciones severas. [4], [5]

## 1.3 LA TÉCNICA SFRA

El método SFRA (Sweep Frequency Response Analysis) o simplemente SFRA, actualmente es el método utilizado con mayor frecuencia en el diagnóstico de estos equipos (SFRA off-line); consiste en la aplicación por un lado del transformador, de una serie de señales sinusoidales de bajo voltaje aproximadamente de 1 a 10 Volts pico a pico [6], estas son aplicadas en diferentes frecuencias que varían en un ancho de banda entre los Hz y los MHz, en este método la función de transferencia se define directamente como el cociente entre las mediciones tomadas a la salida (corriente), y las mediciones tomadas a la entrada (voltaje), para todos los valores de frecuencia medidos.

En esta prueba se puede identificar que las fallas asociadas con el núcleo son identificadas en baja frecuencia, mientras que fallas asociadas con desplazamiento pequeños, problemas en los conductores, leve deformación, son identificadas en media y alta frecuencia. [7]

A continuación se presentan los avances obtenidos con el método FRA con el transformador en servicio (técnica FRA on-line).

## 1.4 EL METODO FRA ON-LINE

“La técnica FRA con el transformador en carga (on-line), consiste en aplicar los métodos IFRA o SFRA con el transformador conectado al resto del sistema de potencia, actualmente los primeros avances de la prueba FRA se han llevado a cabo con el método IFRA, debido a que la mayoría de la investigaciones han estudiado este primer caso con mayor frecuencia, mediante el uso de señales no controladas provenientes del sistema de potencia” [8], [1], [9].

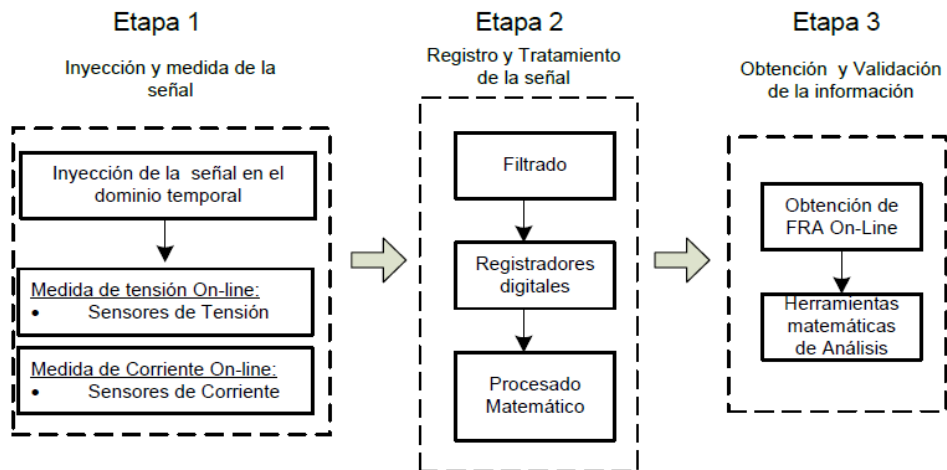
Durante los últimos años se han realizado investigaciones con el fin de mejorar el método FRA; En [10] se evidencia el desarrollo de plataformas para el estudio de esta prueba de manera on-line, así como el desarrollo de nuevas técnicas para la interpretación de la curva [11]; además con el avance de las nuevas tecnologías se han buscado nuevas alternativas para lograr obtener dicha curva con el transformador energizado.[8]

Una de estas alternativas es el método IFRA on-line (obtención de la respuesta en frecuencia mediante la inyección de pulsos con el transformador en servicio), a continuación se menciona en que consiste este método, el cual fue base para el desarrollo del presente trabajo.

### 1.5 EL METODO IFRA ON-LINE

Es un método desarrollado en [1], el cual consiste en que mediante una serie de pulsos inyectados por uno de los terminales del transformador, obtener la respuesta en frecuencia con el transformador en servicio, sin embargo la obtención de dicha curva, y procesamiento de los pulsos se realiza hasta el momento de manera off-line. Este método se basa en la técnica IFRA y se divide en tres etapas tal como se indica en la figura 2.

Figura 2. Metodología implementada para la técnica IFRA-online.

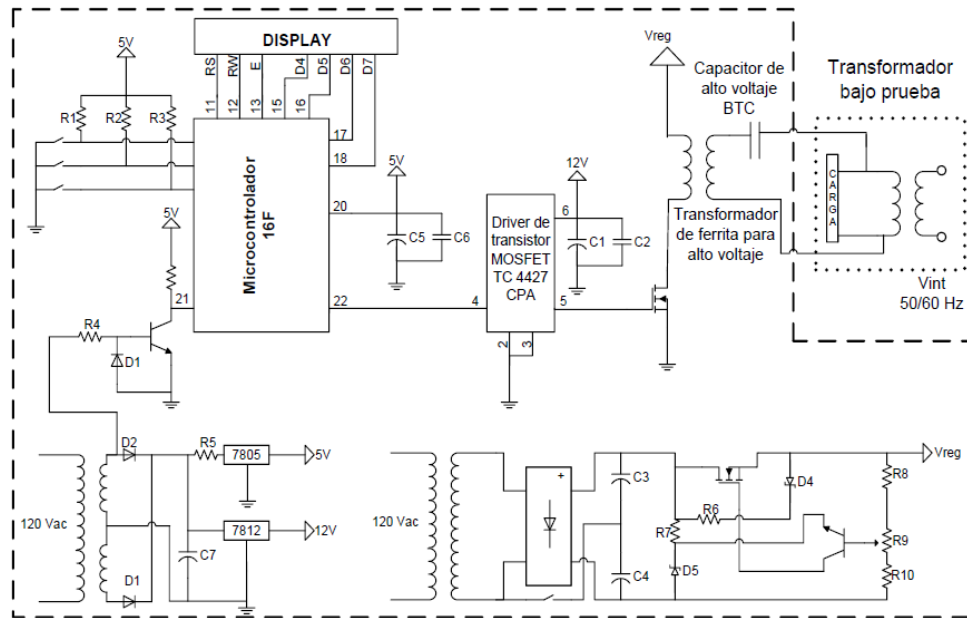


Fuente: (Gómez Luna, Obtención de la respuesta en frecuencia de transformadores en servicio a partir de la medida de señales transitorias, 2013)

### 1.5.1 Etapa 1.

Esta primer etapa hace referencia a la inyección de la señal de excitación, para ello en [1] se diseñó un circuito electrónico que permite inyectar pulsos sobre la señal de 60 Hz, este circuito permite controlar la amplitud, el periodo y la fase del pulso generado. A continuación en la figura 3, se presenta el esquema desarrollado para este circuito de inyección.

Figura 3. Esquema del circuito de inyección de pulso para el método IFRA on-line.



Fuente: (Gómez Luna, Obtención de la respuesta en frecuencia de transformadores en servicio a partir de la medida de señales transitorias, 2013)

A este sistema de inyección se le realizó una modificación para inyectar los pulsos con mayor energía, y así poder utilizar dichos pulsos en pruebas en línea con transformadores de potencia. Este sistema es un circuito LC el cual permite amplificar la magnitud de los pulsos y servir como aislamiento de hasta 15kV, entre el circuito de inyección y el circuito de potencia. Está conformado por un transformador de ferrita y capacitores de alta tensión tal como se indica en la figura 4, donde se muestran los componentes para este sistema.[12]

Figura 4. Módulo LC para la realización de ensayos en sistemas de 15Kv.



Fuente: (Cuadros Dixon, Identificación de las variaciones en la curva FRA on-line ante cambios internos y externos de un transformador piloto, 2016)

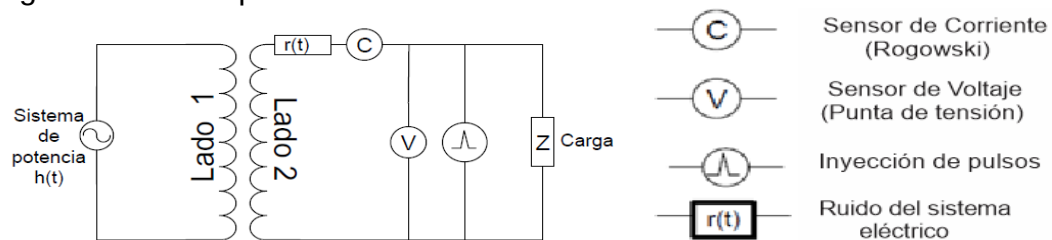
### 1.5.2 Etapa 2.

Hace referencia a la etapa de filtrado, en la cual se hace el uso de la transformada wavelet en la modalidad del Análisis de Multirresolución (MRA) permitiendo de forma simultanea el filtrado de los pulsos en la señal de red (50/60 Hz) y eliminando los efectos no deseados (ruido eléctrico) obtenidos en la medición.

### 1.5.3 Etapa 3.

En esta etapa se realiza la obtención de los espectros de frecuencia de los pulsos inyectados; para esto también se hace uso de la transformada wavelet, la cual posee mejores características que Fourier en cuestión de análisis en señales transitorias, esta herramienta permite el estudio de la señal transitoria en detalle tanto en el dominio de tiempo como en el dominio de la frecuencia utilizando pequeñas ventanas temporales para altas frecuencias y grandes ventanas para bajas frecuencias; permitiendo así, determinar con exactitud la frecuencia y el tiempo exacto de cada uno de los componentes de dichas señales. En la figura 5, se presenta el esquema de método IFRA on-line con la inyección de pulsos controlados y la medición de la señal.

Figura 5. Esquema del método IFRA on-line con medición de las señales



Fuente: (Gómez Luna, Obtención de la respuesta en frecuencia de transformadores en servicio a partir de la medida de señales transitorias, 2013)

Si bien el método IFRA on-line presenta una gran ventaja con respecto al método tradicional SFRA off-line, ya que puede obtener la curva FRA con el equipo en servicio; sin embargo, el inconveniente de este método es que la obtención de la respuesta en frecuencia es algo demorada, ya que si bien las mediciones de los pulsos de tensión y corriente se toman de manera on-line, el procesamiento y análisis de estas es off-line (fuera de línea); y es aquí donde el trabajo de grado desarrollado presenta un gran aporte, logrando el monitoreo y la obtención de manera dinámica de la curva FRA con el transformador en servicio; es decir el nuevo método IFRA on-line obtiene las mediciones de los pulsos, realiza el procesamiento y análisis de manera on-line, por lo que el tiempo de obtención de la respuesta en frecuencia es cuestión de segundos, convirtiéndolo en un método más eficiente y de mayor confiabilidad.

A continuación se hace referencia a la tecnología que se implementó para la obtención y monitoreo de manera dinámica de la curva FRA, con el transformador en servicio.

## 1.6 METODO IFRA ON-LINE EMPLEANDO INSTRUMENTACIÓN VIRTUAL

“En la actualidad la adquisición de datos, sigue una arquitectura tradicional, la cual emplea microcontroladores con funciones para establecer el tiempo en el que se debe de tomar la muestra de los sensores, transformar señales análogas en digitales, almacenarlas en una memoria, y finalmente controlar un módulo de comunicación para transmitir la información” [13]. Sin embargo con el avance de la tecnología se ha desarrollado el método de usar la instrumentación virtual para la adquisición de datos y procesamiento de la señal, donde el elemento principal es el ordenador, generando así una manera más exacta, eficaz y precisa de obtener las mediciones; a diferencia del método tradicional donde el elemento principal es el hardware. [14]

La instrumentación virtual se ha convertido en tendencia para el desarrollo de nuevos equipos electrónicos. Los usuarios pueden obtener un instrumento de medición por medio de un ordenador, una tarjeta de adquisición de datos y un software especializado. A continuación se mencionan algunos avances que se han logrado con la instrumentación virtual.

En [15] se presenta una aplicación desarrollada en Labview para el control y monitoreo de una producción industrial mediante la instrumentación virtual, en esta aplicación se busca reemplazar el uso varios controladores lógicos programables (PLC), por un ordenador equipado.

En [16] se presenta monitoreo on-line para el análisis y rendimiento de un motor de inducción en el que se implementa la instrumentación virtual como medio para obtener los valores de tensión, corriente, potencia, velocidad del motor, lo anterior

simplemente utilizando una tarjeta de adquisición de datos y un ordenador con el que se obtuvieron excelentes resultados.

En el campo del diagnóstico de transformadores se han realizado algunas aplicaciones con instrumentación virtual, una de ellas es la automatización de las pruebas de rutina que se les realizan a estos equipos, los resultados de dichas pruebas se observan de manera on-line a través de una tarjeta de adquisición y pueden almacenarse en una base de datos para su futura recuperación. [17]

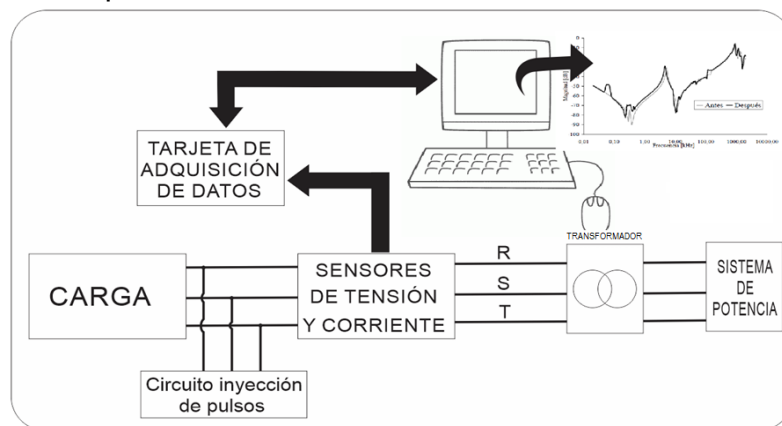
La instrumentación virtual posee un amplio alcance en el análisis de los equipos eléctricos, como los análisis transitorios, efecto térmico, detección de fallas, monitoreo, entre otras aplicaciones. Por lo que se visualiza como la tecnología base para emplear el método IFRA on-line de manera dinámica y de esta manera obtener la respuesta en frecuencia con el transformador en servicio.

#### 1.6.1 Metodología propuesta para el desarrollo del método IFRA on-line con instrumentación virtual.

Para llevar a cabo el método IFRA on-line con instrumentación virtual, es necesario contar con los siguientes equipos: sensores de medición de corriente y tensión con amplio ancho de banda para capturar las señales de alta frecuencia, un circuito de inyección de pulsos mencionado anteriormente, un hardware para la adquisición de las medidas (tensión y corriente), y un computador con un software programable para el procesamiento y análisis de las señales. [1]

A continuación en la figura 6 se aprecia el método IFRA on-line con instrumentación virtual, seguido de las especificaciones técnicas de cada uno de los equipos empleados para el desarrollo de este método.

Figura 6. Esquema del método IFRA on-line con instrumentación virtual



### 1.6.2 Especificaciones técnicas de los equipos para el método IFRA on-line con instrumentación virtual.

#### Transformador.

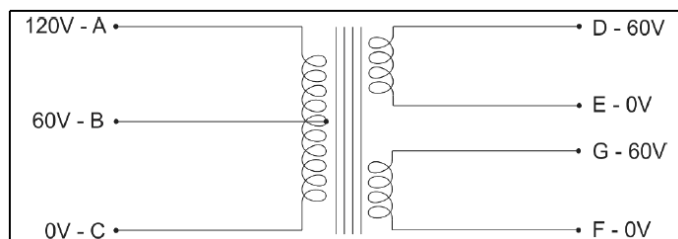
Transformador piloto de la marca Feedback, de referencia TT179, que hace parte de un módulo de entrenamiento en el que se desarrollan pruebas experimentales en condición de vacío y con carga con el fin de observar el comportamiento del equipo, en la tabla 1 se presenta las especificaciones técnicas del transformador bajo prueba.

Tabla 1. Especificaciones técnicas del transformador piloto.

| Descripción                    | Valor  |    |
|--------------------------------|--------|----|
| Potencia                       | 240    | VA |
| Tensión nominal del primario   | 120    | V  |
| Tensión nominal del secundario | 120-60 | V  |
| Corriente Nominal              | 2      | A  |
| Regulación                     | 15     | %  |
| Eficiencia                     | 83     | %  |
| Temperatura max operación      | 70     | °C |

El diagrama esquemático del transformador piloto se muestra en la figura 7 donde se muestran sus terminales de conexión.

Figura 7. Diagrama esquemático del transformador piloto



Fuente: (Cuadros Dixon, Identificación de las variaciones en la curva FRA on-line ante cambios internos y externos de un transformador piloto, 2016)

### 1.6.3 Sistema de medición y adquisición de señales.

Las mediciones de las señales serán realizadas por los siguientes equipos:

- La medición de la tensión es realizada mediante una punta de tensión P2220 de marca tektronix
- La medición de corriente es tomada mediante una Bobina Rogowski

El uso de estos equipos (Bobina Rogowski y la punta de tensión) es debido a su amplio ancho de banda para la medición de señales de alta frecuencia; sin embargo, su medición es planteada, y debidamente argumentada en [1]. En la figuras 8 y 9 se muestran las fotos de la bobina Rogowski y la punta de tensión utilizada; en las tablas 2 y tabla 3 se muestra un resumen de las especificaciones técnicas de estos dos equipos.

Figura 8. Punta de tensión Tektronix P2220.



Fuente: (SgLabs-Test and measurement instruments.(2016).Perugia Italia. Recuperado de <http://www.sglabs.it/en/product.php?s=tektronix-p2220&id=1560>)

Figura 9. Bobina Rogowski.



Fuente: (TestEquipment center.(2015).New York,EU. Recuperado de <https://testequipment.center/RFI-EMI-EMC-Compliance/RF-Current-Monitor-Probes>)

Tabla 2. Especificaciones técnicas de la Bobina Rogowski

| Descripción              | Valor           |
|--------------------------|-----------------|
| Referencia               | 2100            |
| Sensibilidad             | 1 Volt/ Ampere  |
| Ancho de banda           | 125 Hz - 20 MHz |
| Tiempo de subida         | 20 ns           |
| Tasa de bajada           | 0.08 % $\mu$ s  |
| Resistencia de salida    | 50 ohms         |
| Corriente pico max       | 500 A           |
| Corriente rms max        | 7.5 A           |
| Temperatura de operación | 0 a 65°C        |

Tabla 3. Especificaciones de la punta de tensión P2220 Tektronix.

| Descripción                          | Atenuación |          |
|--------------------------------------|------------|----------|
|                                      | x10        | x1       |
| Ancho de banda (Hz)                  | 0 – 200M   | 0 – 6M   |
| Precisión                            | ±2%        | ±2%      |
| Resistencia de entrada (M $\Omega$ ) | 10         | 1        |
| Capacitancia de entrada (pF)         | 13 – 17    | 80 – 110 |
| Tiempo útil de subida (ns)           | <2.2       | <50      |
| Voltaje eficaz máximo (V)            | 300        | 150      |

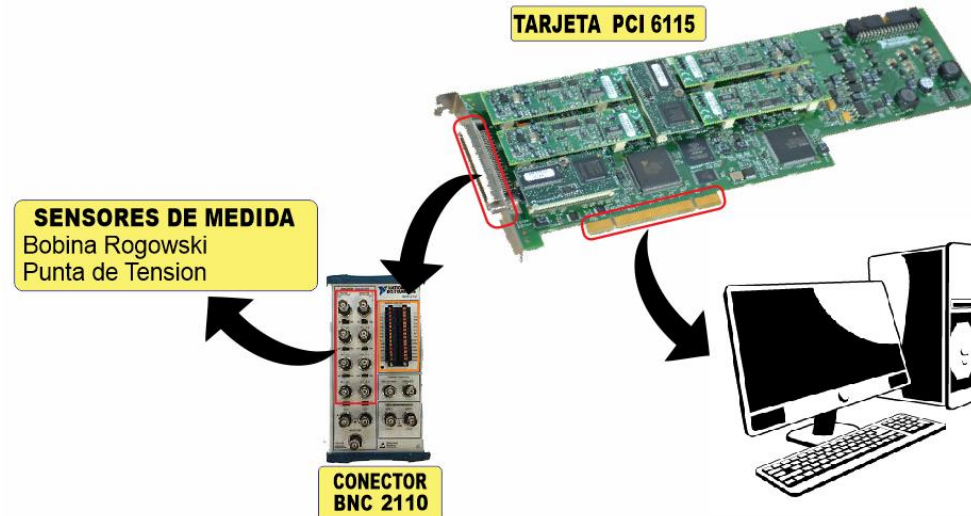
La medición de dichas señales es enviada a una tarjeta de adquisición de datos; la tarjeta con la que se cuenta para este proyecto es la NI PCI-6115 de la National Instruments, esta viene conectada al bloque NI BNC 2110, el cual es el medio de conexión entre los sensores y esta, la tarjeta 6115 se encarga de adquirir las señales sensadas de manera dinámica (adquiere 10 millones de muestras por segundo), dicha tarjeta cuenta con las siguientes especificaciones de fábrica, ver tabla 4.

Tabla 4. Especificaciones de la tarjeta PCI 6115.

| Descripción          | Valor                         |
|----------------------|-------------------------------|
| Entradas analógicas  | 4 entradas, 10 MS/s por canal |
| Memoria interna      | 32 o 64 MS                    |
| Salidas analógicas   | 2 salidas, 4 MS/s             |
| Entradas digitales   | 8                             |
| Voltaje pico máximo  | ± 42V                         |
| driver               | NI-DAQmx                      |
| Software interactivo | Labview Signal Express        |

A continuación se presenta el bosquejo de la conexión entre la tarjeta PCI 6115 y los otros equipos.

Figura 10. Esquema de conexión de la tarjeta National Instruments PCI 6115



Fuente: (Gómez Luna, Obtención de la respuesta en frecuencia de transformadores en servicio a partir de la medida de señales transitorias, 2013)

## **2. SOFTWARE PARA LA OBTENCIÓN DE LA CURVA FRA DE MANERA DINAMICA**

Con los avances obtenidos en instrumentación virtual se hace necesario contar con un determinado software, que pueda llevar a cabo las tareas de adquisición de datos y procesamiento de las señales; así pues con el fin de cumplir con estos objetivos se preseleccionaron los siguiente software como tentativos para manejar el método IFRA on-line con instrumentación virtual: Scada (Supervisory Control And Data Acquisition), Matlab (MATrix LABoratory), Delphi, Neplan, Powerworld, y Labview (Laboratory Virtual Instrument Engineering Workbench); son los software que han obtenido grandes avances en la tareas de monitoreo; y en la actualidad se emplean para desarrollar aplicaciones de software asociadas al campo de la ingeniería eléctrica.

Un software como Scada es especialmente diseñado para desarrollar funciones asociadas al monitoreo en tiempo real, o permitir el control en una producción.

Matlab es un entorno de desarrollo de software Implementado para la representación de datos, creación de interfaces usuario (GUI) y la comunicación con otros programas que funcionan con lenguajes diferentes al de Matlab.

Delphi es un entorno de desarrollo de software destinado para la programación de propósito general, implementada para proyectos como aplicaciones de consola, aplicaciones web.

Neplan es una herramienta de software desarrollada para planificar, optimizar y simular redes eléctricas. La facilidad de la interfaz permite al usuario realizar casos de estudio de manera muy eficiente.[18]

Powerworld es un paquete de simulación del sistema de potencia interactivo diseñado para simular el funcionamiento de alto voltaje en un marco de tiempo que va desde varios minutos hasta varios días.

Labview es un lenguaje de entorno de programación gráfico diseñado para la adquisición de datos, monitoreo o medición de una variable en un proceso.

En el presente capítulo se exponen los avances obtenidos asociados al monitoreo con cada uno de los software preseleccionados, y posterior a ello se presentan las razones de la selección del software con el que se desarrollará el nuevo método IFRA on-line, así como su entorno de trabajo y sus principales herramientas para el manejo de este.

## 2.1 ALTERNATIVAS DE SOFTWARE PROGRAMABLES PARA LA PRUEBA IFRA ON-LINE

Con el objetivo de desarrollar el método IFRA on-line y con los software preseleccionados; se presenta un análisis de la información bibliográfica de estas alternativas, y así escoger el software más adecuado para implementar el método.

### 2.1.1 Scada.

Es un sistema por medio del cual se pueden monitorear y controlar las distintas variables que se presentan en un proceso o planta. Para ello se requieren aplicaciones de software, sistemas de comunicación, unidades remotas que permiten controlar el proceso de forma automática desde la pantalla de un ordenador. Scada es un sistema en el que aplicaciones de monitoreo en tiempo real, simulaciones y control son frecuentemente desarrolladas bajo este criterio, las aplicaciones desarrolladas con este sistema interactúan de manera dinámica con los equipos externos mejorando la confiabilidad y eficiencia del proceso. A continuación se presentan algunos avances obtenidos con este sistema.

En [19] se desarrolló la construcción de un modelo reducido de una subestación, en la que se realizan diferentes tipos de pruebas de automatización, tales como desconexión de cargas, coordinación de protecciones, estudios de flujos de carga y análisis de fallas. Con el fin de optimizar el funcionamiento de una subestación ante disturbios causados por la red eléctrica, además de servir como plataforma para el desarrollo de pruebas de protección y control en una subestación.

En [20] se propone el desarrollo de una aplicación mediante el software DIGFAULT para automatizar el diagnóstico de fallas de un sistema de potencia, permitiendo así estimar el origen de un fallo, brindar una solución óptima en caso de múltiples fallas, o en caso que los dispositivos de protección no funcionen.

En [21] se desarrolla una aplicación para monitorear el rendimiento de una turbina eólica con el fin de mejorar su disponibilidad, fiabilidad y vida útil dentro de la red eléctrica, optimizando la detección temprana de fallas y anomalías en estas turbinas y reduciendo así; el número de mantenimientos no programados debido a una falla.

En [22] se propone el desarrollo de una aplicación para el diagnóstico on-line del transformador mediante el sistema Scada, donde se monitorean variables tales como temperaturas, humedad, análisis de gases y cambiador de TAPS. Proporcionando que el transformador sea más confiable y eficiente dentro de la red eléctrica.

En resumen se puede concluir que es un sistema en el que se pueden desarrollar aplicaciones de software para el monitoreo on-line, diagnósticos y simulaciones en tiempo real, permitiendo obtener un control y monitoreo de un sistema o proceso, sin embargo existen multitud de productos Scada en el mercado, por lo que la búsqueda de la selección de software mediante este sistema, se extendería debido a la variedad de aplicaciones desarrolladas.

### 2.1.2 Delphi.

Delphi es un Lenguaje de programación basado en programación visual de propósito general, el uso habitual del Delphi es el desarrollo de aplicaciones visuales, y bases de datos cliente-servidor, aplicaciones de consola, aplicaciones web, servicios del sistema operativo.[23]

En algunas aplicaciones Delphi ha sido implementado como medio de simulación de las redes eléctricas [24], así como también se han logrado aplicaciones para la adquisición de datos permitiendo así el monitoreo de un proceso o sistema [25] [26], también se evidencia la gran capacidad de este software en términos de precisión, flexibilidad y velocidad al momento de responder en un proceso [27] [28], quedando así demostrado, que es un analizador robusto ideal para el desarrollo de cualquier aplicación.

Sin embargo en el campo de la instrumentación virtual con este software se han obtenido pocos avances debido a que es una herramienta que está enfocada al tema de la base de datos y al desarrollo de manera general de múltiples aplicaciones.

### 2.1.3 Neplan.

Es un software implementado para el estudio de sistemas de potencia, a través de simulaciones realizadas por un ordenador, empleando herramientas para el análisis de flujos de potencia, cortocircuito y análisis de estabilidad de pequeñas señales [29]; Esto le proporciona a los estudiantes, la capacidad de construir modelos, realizar cálculos y analizar resultados.

Neplan permite realizar simulaciones de cualquier tipo de sistema de potencia; en los últimos años se han logrado avances para la simulación de Smart grid o redes eléctricas inteligente. [30]

En [31] utilizan Neplan para analizar el efecto que puede generar un sistema fotovoltaico interconectado a un circuito ramal de una red de distribución eléctrica, permitiendo evidenciar así excelentes resultados.

Sin embargo en el análisis on-line se han desarrollado pocas aplicaciones mediante este software, una de ella es en [32] donde se implementa Neplan para analizar los parámetros de los componentes de una red eléctrica de manera on-line demostrando así confiabilidad y buena capacidad para la comunicación con datos y software externos, mediante el uso de librerías dinámicas o DLL. (Dynamic Link Library)

Con base a lo anterior se puede concluir que el software Neplan es una herramienta muy potente y con una gran aplicabilidad para estudiar los efectos que se generan en un sistema potencia sin embargo; dicha herramienta no se puede visualizar como la mejor alternativa para el estudio de una prueba tan específica como la respuesta en frecuencia en un transformador, ya que su aplicabilidad se encuentra con un mayor enfoque en los temas asociados a pérdidas de potencia, flujos de potencia, análisis de cortocircuito, estabilidad de pequeñas señales, entre otros factores que influyen en la confiabilidad y rendimiento de una red eléctrica.

#### 2.1.4 Powerworld.

Es una herramienta potente diseñada para la simulación interactiva de sistemas de alta tensión o sistemas de potencia, que comprenden un marco de tiempo que va desde varios minutos hasta varios días [33]. Algunas aplicaciones desarrolladas mediante este software, brindan soluciones óptimas en la reducción de los costos de operación de generadores, también permiten modelar controladores para brindar soluciones en el mejoramiento de estabilidad dentro de la red eléctrica. [34] y [35]

Sumado a esto, se han desarrollado aplicaciones que buscan la integración de este software con otros; con el fin de brindar mejores soluciones para el sistema de potencia o despacho óptimo de energía. [36]

Sin embargo los avances para aplicaciones en tiempo real han sido pocos, hasta el momento se han desarrollado técnicas de visualización en tiempo real para la observación del flujo de potencia, información acerca de la demanda de energía, evaluación de ubicación y magnitud de sobrecargas, entre otros parámetros, para monitorear frecuentemente el estado del sistema de potencia, o una micro red eléctrica. [37] y [38]

De lo anterior se puede concebir que el software Powerworld ha obtenido grandes avances en el desarrollo de herramientas y aplicaciones para el estudio de los sistemas de potencia, sin embargo las aplicaciones y herramientas que contiene este, no se encuentran enfocadas para realizar un tratamiento y análisis adecuado de señales transitorias.

### 2.1.5 Matlab.

Matlab, *MATrix LABoratory* ("laboratorio de matrices"); es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Implementado para representación de datos, creación de interfaces usuario (GUI), y la comunicación con otros programas en otros lenguajes. [39]

En el desarrollo de aplicaciones donde se requiere la adquisición de datos, monitoreo o análisis de señales transitorias se ha implementado este software como medio para estudiar el comportamiento del sistema de manera on-line.[40] y [41]

También se ha utilizado para desarrollar modelos de sistemas de control, integración con micro controladores y tarjetas de adquisición con el fin obtener modelos dinámicos y controlar de manera eficiente los disturbios generados en el área de los motores. [42] y [43]

Sumado a estos avances, Matlab también ha desarrollado aplicaciones para el estudio de señales transitorias tal como se expone en [44], [45] donde se implementa la transformada wavelet en el software Matlab, para la localización de fallas en una línea de transmisión, monitoreo de descargas parciales y análisis de las señales transitorias medidas, en [46] se implementa esta transformada con Matlab para el diseño de filtros de señales electrocardiográficas, las cuales se ven frecuentemente distorsionadas por el ruido.

En resumen se puede evidenciar el potencial que tiene el software Matlab para el análisis y procesamiento de señales transitorias, su facilidad para la integración con otros software; y sus desarrollos de aplicaciones para el estudio on-line. Convirtiéndolo así, en una de las mejores alternativas para el desarrollo de este trabajo de grado.

### 2.1.6 Labview.

Un Software como Labview facilita el desarrollo de los sistemas virtuales, utilizado en el análisis y sistemas de vibraciones ,instrumentación, control de instrumentos, procesamiento de señales, programación, o cualquier disciplina que requiere la adquisición de datos, control y análisis de datos.[47]

A continuación se exponen algunos desarrollos en instrumentación virtual asociados a la parte eléctrica realizados por Labview.

En [48] se ha utilizado Labview para desarrollar aplicaciones de monitoreo en una red eléctrica donde se busca realizar mediciones de tensión, corriente, energía, y

analizar armónicos de manera on-line, generando así un sistema flexible y viable en comparación con los métodos convencionales.

En el campo del diagnóstico de transformadores se han desarrollado aplicaciones con Labview que permiten realizar una serie de pruebas a un transformador en tiempo real contando con un sistema de adquisición de datos, para tomar medidas, registrar y analizar los datos generados por los equipos externos, esto con el fin de determinar los parámetros necesarios como corriente, tensión, potencia, frecuencia y temperatura dinámicamente. [17] [49]

También se han desarrollado aplicaciones para determinar la detección continua del contenido de gases disueltos que se dan en el equipo permitiendo así el monitoreo y el diagnóstico de manera on-line. [50]

Sumado a esto, en [51] se desarrolló una aplicación para obtener la respuesta en frecuencia de un transformador mediante Labview, esta aplicación permite diagnosticar anomalías en el transformador de una manera menos costosa, en comparación con el sistema convencional SFRA comercial.

No obstante con los avances presentados en el campo del monitoreo on-line, tiempo real y diagnóstico de transformadores, Labview también permite la integración con otros software para desarrollar de aplicaciones más complejas, o que requieren un mayor procesamiento matemático, tal como se presenta en [52], [53], [54], [55]; donde se implementa la integración de Labview con Matlab con el fin de desarrollar aplicaciones en el campo de la robótica, análisis de señales transitorias, sistemas de control, entre otros.

En resumen Labview es una potente herramienta para la adquisición, análisis y procesamiento de datos, donde haciendo uso de lenguajes de programación gráfico (lenguaje G) o el lenguaje M, permite desarrollar aplicaciones complejas mediante la interconexión de sus instrumentos virtuales, convirtiéndolo así en una herramienta de desarrollo poderosa y flexible. [53]

## 2.2 LABVIEW, SOFTWARE SELECCIONADO PARA DESARROLLAR LA PRUEBA IFRA ON-LINE

Con referencia a los avances que se han logrado con cada software se puede visualizar a Labview como la mejor alternativa para el desarrollo del trabajo de grado; debido a que presenta las siguientes ventajas: presenta compatibilidad con el equipo de adquisición de datos, cuenta con las principales aplicaciones desarrolladas en tiempo real, cuenta con amplia compatibilidad para integrar herramientas y programas de otras aplicaciones dentro de los que se tiene la integración de este con el software de Matlab, permitiendo así la facilidad de implementar el lenguaje M dentro de este; y convirtiéndolo en una herramienta potente y flexible para el desarrollo de cualquier aplicación [53],[56],[57],[58].

A continuación se expone a Labview como medio para implementar la instrumentación virtual sobre la técnica IFRA o respuesta en frecuencia ante un impulso.

### 2.3 PLATAFORMA PARA INSTRUMENTACIÓN VIRTUAL LABVIEW

Labview (*Laboratory Virtual Instrument Engineering Workbench*) es un software orientado a desarrollar aplicaciones para la instrumentación, utilizando un lenguaje de programación gráfica denominado lenguaje G, el cual permite programar las aplicaciones mediante la interconexión lógica entre diversos elementos o instrumentos virtuales. A diferencia de otros sistemas de programación que utilizan líneas de código tipo texto para la creación de aplicaciones, Labview cuenta con una serie de diagramas de bloques para la programación como lo son: librerías para la adquisición de datos, acondicionamiento de datos, comunicación en redes, análisis de señales, control y almacenamiento de datos [59]. Cada programa desarrollado en Labview (VIs) cuenta con dos interfaces o dos entornos de trabajo:

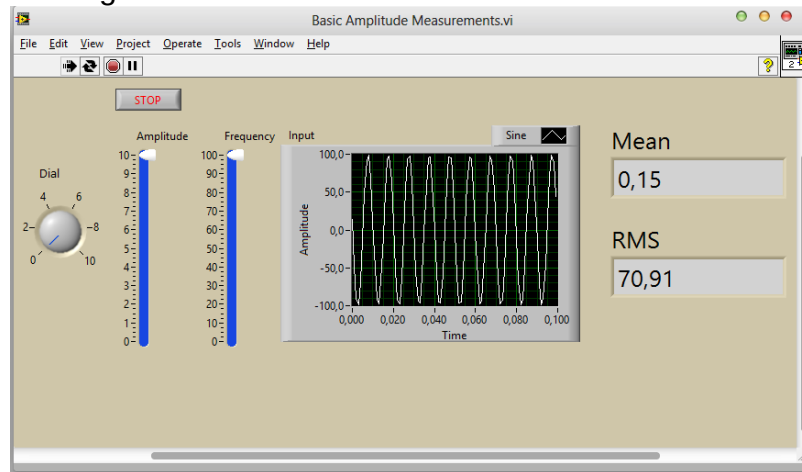
- panel frontal
- diagrama de bloques

Cada elemento de entrada o salida presente en la interfaz de usuario o panel frontal, tiene su correspondiente representación en el diagrama de bloques, a continuación se realiza una breve descripción de estos dos entornos de trabajo.

#### 2.3.1 Panel frontal.

Simula los instrumentos físicos y facilita el ingreso de datos a través de controles como pulsadores, perillas y botones; El resultado de cada aplicación o proceso, puede ser visualizada mediante medidores o indicadores numéricos y gráficos. Permitiendo así, establecer un control por parte del usuario sobre un algún proceso. En la figura 11 se presenta el panel frontal o interfaz de una aplicación desarrollada para la variación de parámetros tales como amplitud y frecuencia de una señal sinusoidal.

Figura 11. Panel frontal o interfaz de Labview

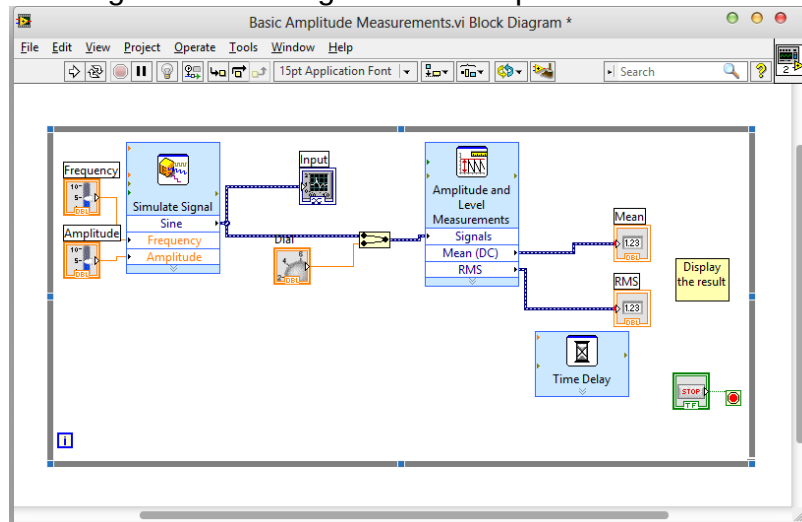


Fuente: (ejemplo Basic Amplitude Measurements, software Labview 2011)

### 2.3.2 Diagrama de bloques.

Es el código fuente de los VIs; representa los controles e indicadores del panel de control o interfaz, además cuenta con funciones y estructuras que realizan tareas específicas para determinar el flujo de datos y resultados en el programa. En la figura 12 se presenta el código fuente de la aplicación mostrada en la figura 11.

Figura 12. Diagrama de bloques de Labview



Fuente: (ejemplo Basic Amplitude Measurements, software Labview 2011)

A continuación se mencionan las paletas de herramientas, controles e indicadores, las barras de herramientas, paletas de funciones, clases de datos y por ultimo las estructuras de programación que son frecuentemente utilizados en Labview.

### 2.3.3 Paleta de herramientas.

Contiene las herramientas necesarias para editar y depurar los objetos tanto en el panel frontal como en el diagrama de bloques, esta paleta se encuentra en *view / tools palette*, en la figura 13 se presenta esta paleta de herramientas con sus componentes.

Figura 13. Paleta de herramientas



Fuente: (Holguin G, Perez S, Orozco A. (2002).Curso básico Labview 6i.

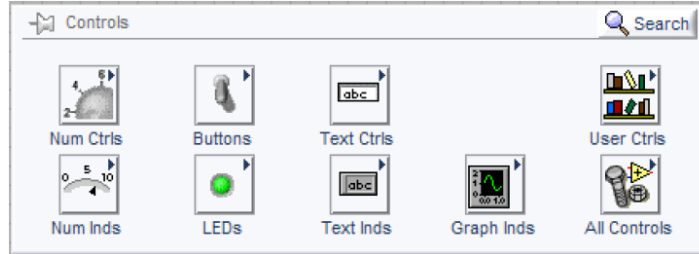
Recuperado de

<http://www.ie.itcr.ac.cr/einteriano/control/labview/ParaAprender/Curso%20LabVIEW6i.pdf>)

### 2.3.4 Paleta de controles.

Se utiliza únicamente en el panel frontal, la cual contiene los objetos necesarios para la creación de interfaz (controladores e indicadores). En la figura 14 se presenta esta paleta con sus componentes.

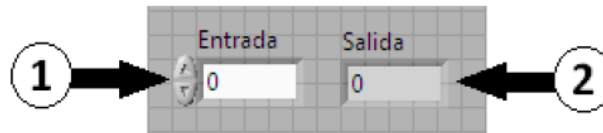
Figura 14. Paleta de controles



Fuente: (software Labview 2011)

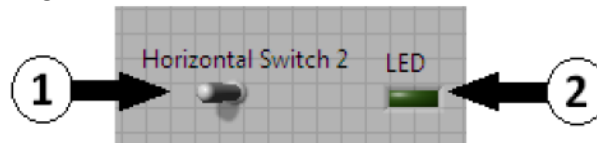
A continuación en las figuras 15, 16, 17 y 18 se visualizan algunos controles con los diferentes tipos de indicadores que se encuentran en el panel frontal, tales como indicadores numéricos, booleanos, indicadores tipo cadena de caracteres, y los tipos gráficos.

Figura 15. Control e indicador numérico



Fuente: (Guerrero Christian, Monitoreo del transformador en servicio, mediante la respuesta en frecuencia haciendo uso de la plataforma Labview, 2013)

Figura 16. Control e indicador booleano



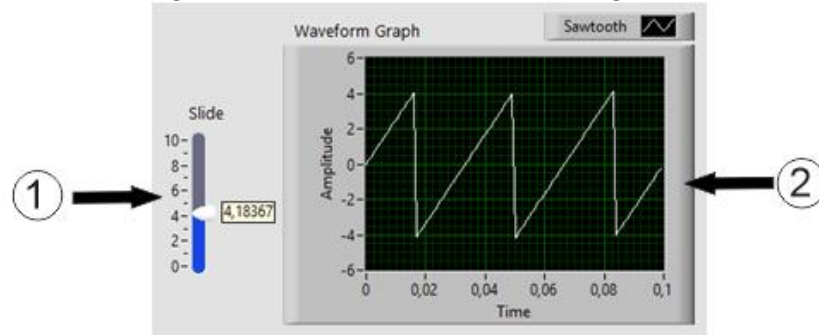
Fuente: (Guerrero Christian, Monitoreo del transformador en servicio, mediante la respuesta en frecuencia haciendo uso de la plataforma Labview, 2013)

Figura 17. Controles e indicadores de cadena de caracteres



Fuente: (Guerrero Christian, Monitoreo del transformador en servicio, mediante la respuesta en frecuencia haciendo uso de la plataforma Labview, 2013)

Figura 18. Control e indicador gráfico

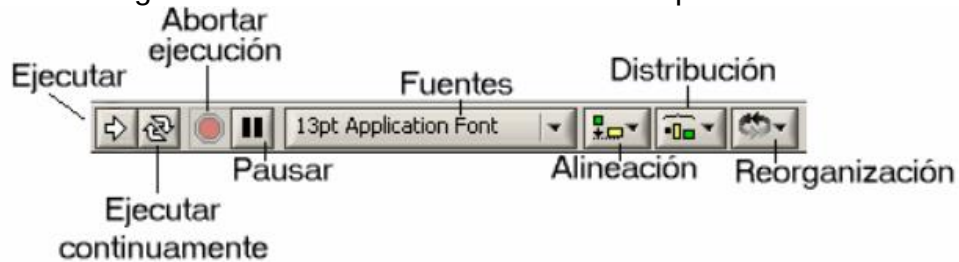


Fuente: (software Labview 2011)

### 2.3.5 Barra de herramientas del panel frontal.

Esta barra se encuentra en la parte superior del panel frontal, la barra de herramientas permite la ejecución, depuración y organización de los VIs, tal como se indica en la figura 19.

Figura 19. Barra de herramientas del panel frontal



Fuente: (Holguin G, Perez S, Orozco A. (2002).Curso básico Labview 6i.

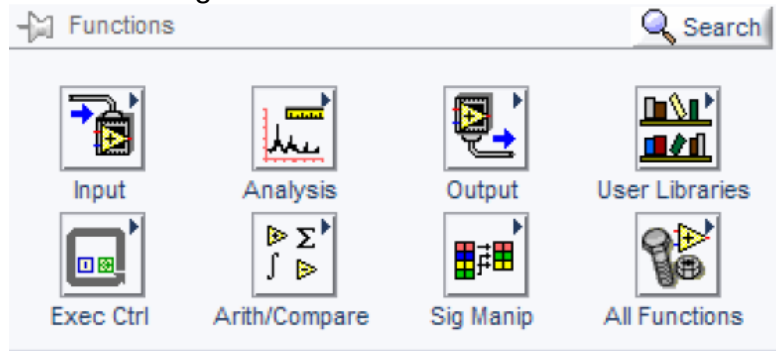
Recuperado de

<http://www.ie.itcr.ac.cr/einteriano/control/labview/ParaAprender/Curso%20LabVIEW6i.pdf>

### 2.3.6 Paleta de funciones

Se utiliza únicamente en el diagrama de bloques, contiene los objetos necesarios para crear y editar el código fuente de la aplicación, está compuesta por submenús con funciones aritméticas, estadísticas, cálculo y trigonometría, además de contar con estructuras de programación como For loop, while loop, case y sequence entre otras funciones de lógica booleana, adquisición de datos, procesamiento de señales como filtros, análisis de espectros, el análisis de Fourier, análisis wavelet, comunicaciones con equipos externos entre otras, tal como se indica en la figura 20.

Figura 20. Paleta de funciones

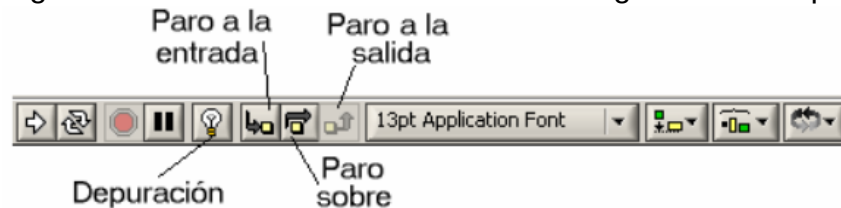


Fuente: (software Labview 2011)

### 2.3.7 Barra de herramientas del diagrama de bloques

Barra que contiene además de los mismos botones del panel frontal, las herramientas de depuración, ver figura 21.

Figura 21. Barra de herramientas del diagrama de bloques



Fuente: (Holguin G, Perez S, Orozco A. (2002).Curso básico Labview 6i.

Recuperado de

<http://www.ie.itcr.ac.cr/einteriano/control/labview/ParaAprender/Curso%20LabVIEW6i.pdf>)

### 2.3.8 Clases de datos en Labview.

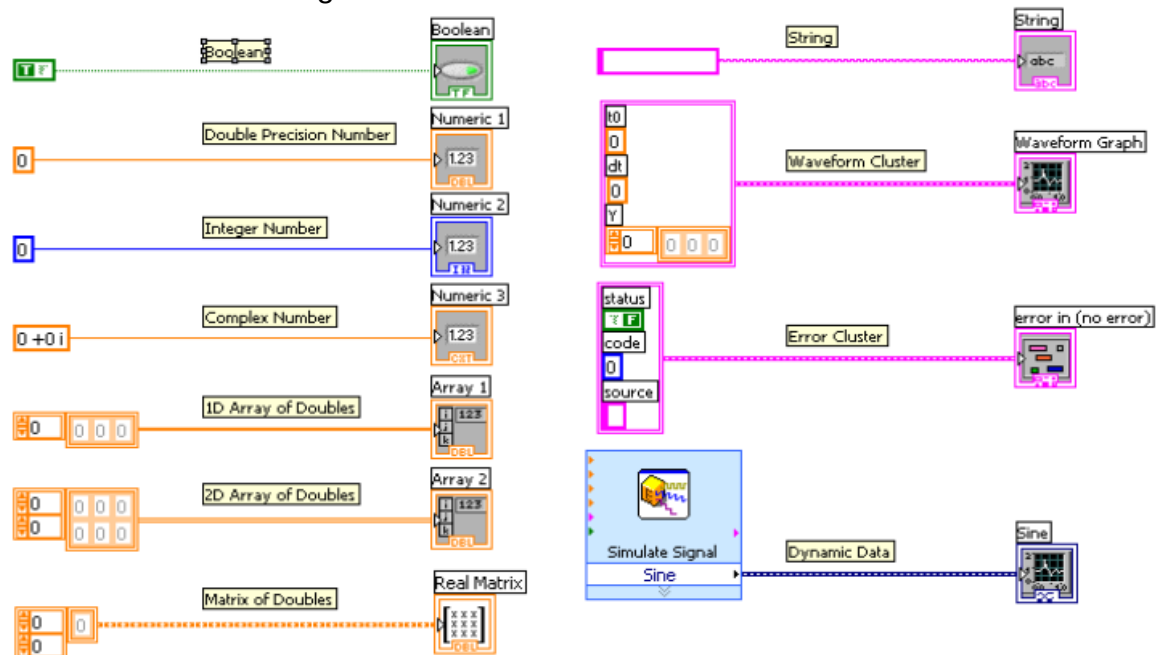
Los tipos de datos que se implementan en el diagrama de bloques son:

- Boolean: Representa los dos estados lógicos TRUE (verdadero) o FALSE (falso).
- Numeric: Dato principal de Labview, representa cualquier tipo de número según su tipo (real, imaginario, complejo, con signo, entero).
- String: Representa con un color rosa, representa letras palabras y oraciones.

- Cluster: Agrupan elementos o datos de tipo mixto; ejemplo: un cluster de error de Labview combina un valor booleano, un valor numérico y una cadena de caracteres.
- Waveform data: Tipos de datos específicos para las ondas, mediante el cual los datos se guardan por medio de matrices de una dimensión.

A continuación en la figura 22 se visualizan las clases de datos que se usan en el diagrama de bloques

Figura 22. Clases de datos en Labview



Fuente: (Corcuera Pedro. Introducción a Labview. Recuperado de [http://personales.unican.es/corcuerp/LabView/Slides/Intro\\_LabView.pdf](http://personales.unican.es/corcuerp/LabView/Slides/Intro_LabView.pdf))

### 2.3.9 Estructuras de programación de Labview.

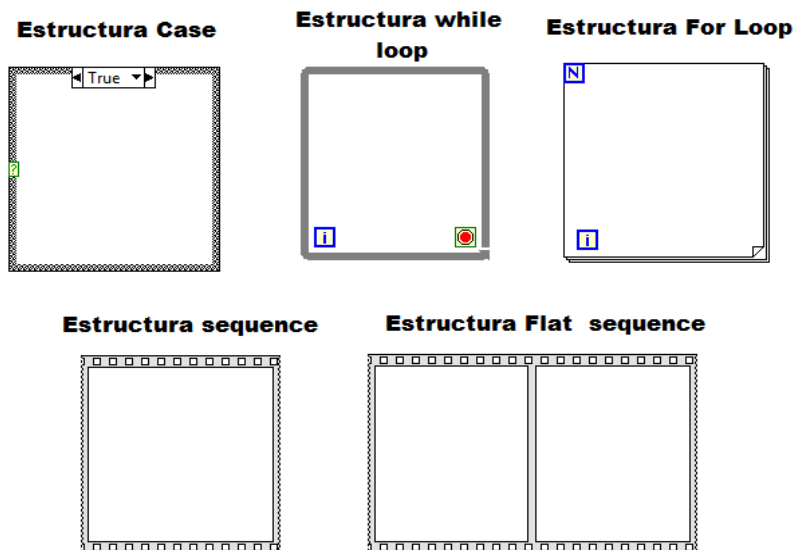
Labview cuenta con estructuras de programación que son implementadas con frecuencia en otros software, a continuación se mencionan las más utilizadas.

- *Stacked sequence structure:* permite establecer un orden de ejecución entre tareas donde no existe una dependencia de datos.

- *Flat sequence structure*: permite utilizar varias estructuras tipo sequence simultáneamente y asegurar un orden de ejecución para cada subdiagrama permitiendo que se ejecute antes o después del otro subdiagrama.
- *Estructura Case*: Estructura condicional. Presenta varios subdiagramas (casos) de los cuales solo se ejecuta un subdiagrama según el dato que llega al selector de este.
- *Estructura for loop*: permite ejecutar N veces todas las operaciones colocadas dentro su subdiagrama.
- *Estructura While Loop*: Estructura de iteración. Permite ejecutar todas las operaciones colocadas dentro de su subdiagrama, mientras que una determinada condición sea “true” ó “false”.

A continuación en la figura 23 se presentan los esquemas de cada una de estas de estructuras.

Figura 23. Estructuras de programación en Labview



#### 2.3.10 Programación basada en texto.

Existe evidencia significativa de que los estudiantes de ingeniería presentan dificultades al momento del aprendizaje en programación basados en texto [60], estudios encuentran que la gran mayoría suelen ser más visuales e intuitivos en cuanto a los estilos de aprendizaje, el lenguaje de programación gráfico o lenguaje

G proporciona una interfaz intuitiva y un entorno de programación amigable para el desarrollador o usuario, el objetivo principal del lenguaje G es proporcionar un fácil apoyo para ejecutar tareas como medición, control, monitoreo, entre otras [60]. Sin embargo el lenguaje de programación gráfico no es el más adecuado al momento de representar fórmulas o ecuaciones matemáticas, la cuales pueden ser representadas fácilmente en texto.

Labview cuenta con tres herramientas para evaluar dichas expresiones matemáticas, estas se encuentran dentro del diagrama de bloques, a continuación se mencionan y describen cada una de ellas.

- **Formula node:** Es una herramienta que evalúa las formulas y expresiones matemáticas utilizando la estructura de sintaxis de C++, dicho nodo cuenta con funciones matemáticas básicas como lo son: sqrt, log2, máximo, mínimo, abs, acos, acosh, asin, asinh, atan, atanh, cos, cosh. Tal como se presenta en la figura 24 donde se utiliza la función suma.
- **Mathscript Node:** Este nodo adiciona matemática textual al entorno de Labview, cuenta con más de 800 funciones incorporadas y la posibilidad ejecutar algunos archivos.m, los cuales se han desarrollado en el software Matlab o GNU Octave. En la figura 25 se presenta el nodo utilizando algunas herramientas matemáticas como la transformada de Fourier, el valor absoluto, o determinando la longitud de un vector.
- **Matlab script node:** Esta herramienta llama al software Matlab para ejecutar secuencias de comandos o funciones desarrollados en Labview, para el funcionamiento de dicha herramienta se debe tener como requisito el software de Matlab versión 6.5 o superior instalado en el equipo, este nodo o punto de conexión invoca el servidor con que cuenta Matlab para ejecutar los script escritos dentro de este. Permitiendo así trabajar con cualquier función o librería de Matlab dentro de Labview. En la figura 26 se presenta esta herramienta utilizando funciones especiales de Matlab como la transformada wavelet continua, y su wavelet madre.

Figura 24. Formula node

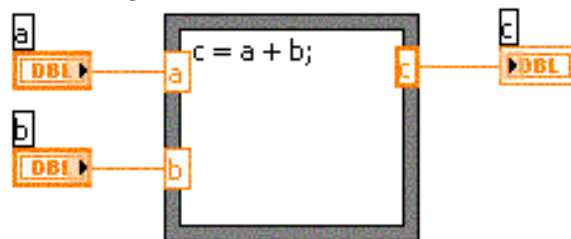


Figura 25. Mathscript Node

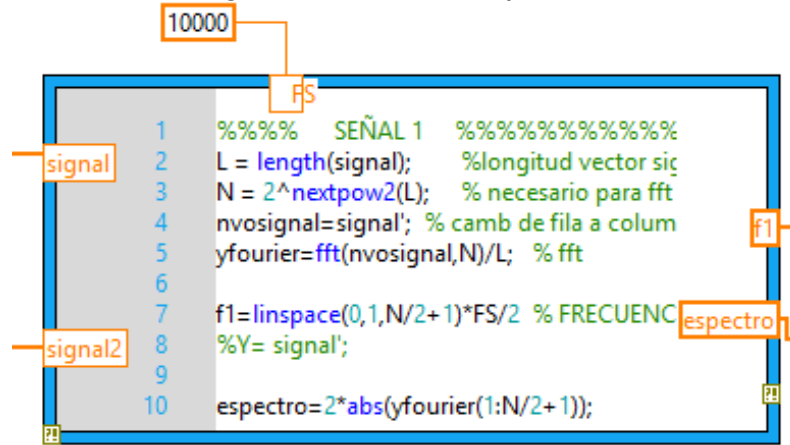
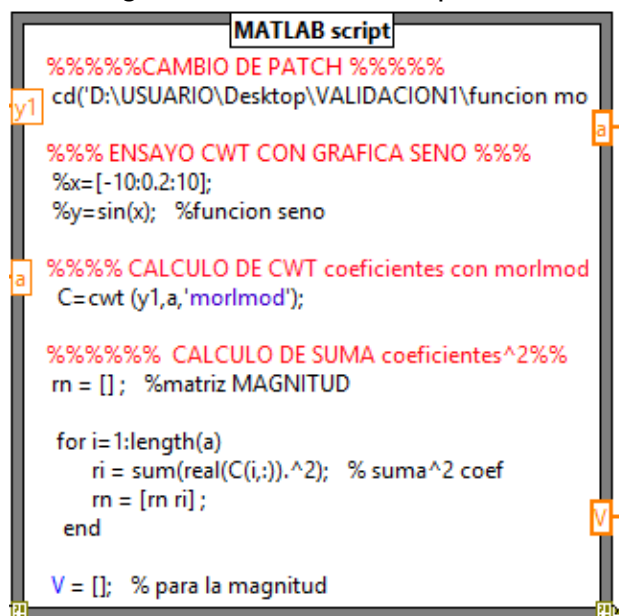


Figura 26. Matlab script Node



### **3. DESARROLLO DE LA APLICACIÓN SOFTWARE PARA EL MÉTODO IFRA ON-LINE**

El monitoreo del transformador mediante la respuesta en frecuencia es de vital importancia para contar con un sistema, el cual permita identificar y diagnosticar en tiempo real, posibles fallas que se puedan presentar en el equipo mientras este se encuentre en servicio; y así poder brindar una condición óptima para el funcionamiento del equipo dentro de la red eléctrica.

En el presente capítulo se describe cada una de las etapas que comprenden el desarrollo de la aplicación software, teniendo en cuenta la metodología realizada en el método IFRA-online y el trabajo realizado en [13]. A continuación se mencionan brevemente las etapas que comprenden esta aplicación, y posteriormente se realiza una descripción de manera más detallada del funcionamiento de cada una y de los VIs (instrumentos virtuales) involucrados.

- **Adquisición:** En la etapa de adquisición de datos se contará con la tarjeta NI PCI 6115 de la National Instruments, que será necesaria para adquirir las señales de tensión y corriente inyectadas en el transformador.
- **Filtro de 60 Hz:** Etapa implementada para eliminación de la señal de 60 Hz y el ruido presente en el sistema; con el fin de filtrar y obtener los pulsos inyectados en el transformador.
- **Filtro Denoising:** Etapa implementada para la eliminación del ruido eléctrico, presente en los pulsos que se encuentran bajo estudio.
- **Ventana de Análisis “Trigger”:** Etapa implementada para extraer un segmento de la señal filtrada (pulsos), y así proceder al análisis matemático de los pulsos encontrados.
- **Procesamiento de Datos:** Para el análisis de señales transitorias de muy corta duración (pulsos), es necesario contar una herramienta matemática como la transformada wavelet, la cual es defendida y argumentada en [1].

### 3.1 ETAPAS IMPLEMENTADAS DEL DESARROLLO DE LA APLICACIÓN SOFTWARE PARA EL MÉTODO IFRA ON-LINE

#### 3.1.1 Etapa adquisición de datos.

El desarrollo de la etapa de adquisición de datos, está compuesta por dos partes, la primer parte es el hardware, el cual es el medio físico para adquirir las señales sensadas; y la segunda parte es el desarrollo software, el cual permite el procesamiento de las señales que se desean adquirir. A continuación se menciona el contenido y funcionamiento de estas dos partes.

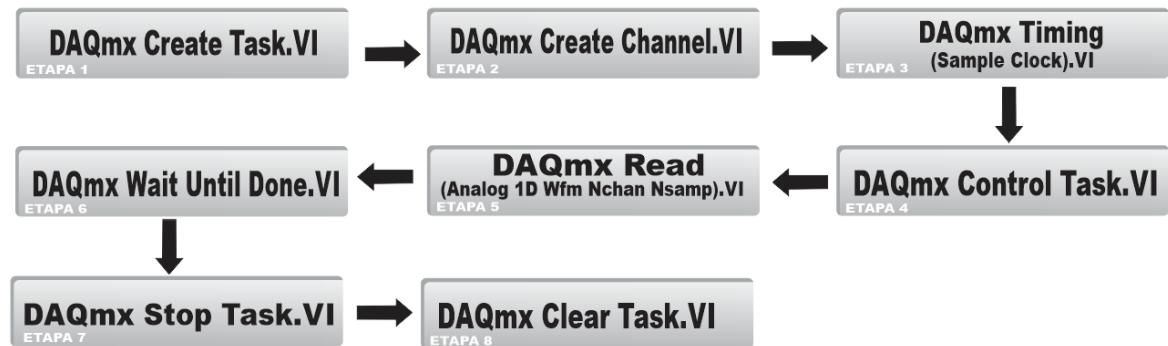
##### 3.1.1.1 Hardware.

Las características del hardware para la adquisición de datos es de vital importancia al momento de obtener las mediciones en el menor tiempo posible, una tarjeta de gran capacidad para la toma de muestras reduce el tiempo de pruebas, el ahorro de energía, y mejora la precisión en la toma de datos. En el capítulo uno se mencionó la tarjeta NI PCI 6115 y sus especificaciones como medio, para obtener las mediciones de tensión y corriente realizadas por la punta de tensión y la bobina Rogowski.

##### 3.1.1.2 Software.

En esta sección, se mencionan los VIs comprendidos dentro de la etapa de adquisición de datos, posteriormente se describe el funcionamiento y los parámetros de configuración de cada uno de ellos; dichos VIs hacen parte del conjunto de bloques con que cuenta la toolset NI-DAQmx, y su implementación es argumentada con base al conjunto de especificaciones de la tarjeta PCI 6115 y al trabajo realizado en [13]. En la figura 27 se presenta la secuencia de VIs utilizados para obtener la adquisición de señales de tensión y corriente en tiempo real.

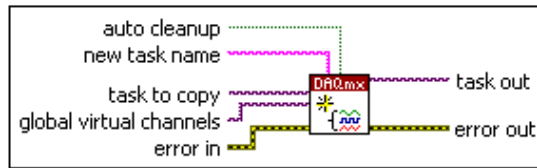
Figura 27. Secuencia de de bloques VIs que contiene la etapa de adquisición de datos



#### 3.1.1.2.1. DAQmx Create Task.VI

Este VI se encarga de crear y adicionar canales virtuales por tarea, si se emplea el DAQmx Create Task dentro de un bucle (while loop o for loop) creará una nueva tarea en cada iteración. Es la primer etapa del proceso de adquisición, y se implementó dentro de un ciclo while loop, para adquirir continuamente los datos. En la figura 28 se presenta este VI con sus terminales de entrada y salida. Y posteriormente sus parámetros de configuración.

Figura 28. DAQmx Create Task.VI



##### 3.1.1.2.1.1. Parámetros de configuración del bloque DAQmx Create Task.VI

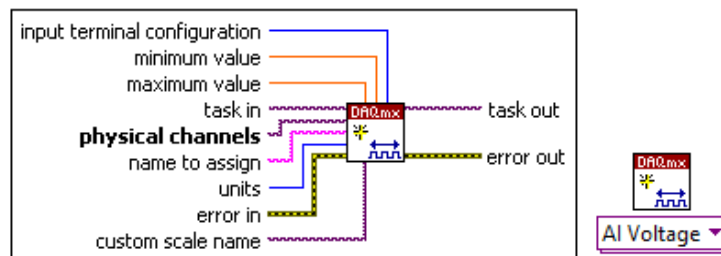
- **Auto cleanup:** Parámetro de estado lógico o booleano implementado para determinar si la tarea impuesta se decide eliminar al momento de finalizar su aplicación, si el estado se encuentra en “Falso” Labview no puede eliminar dicha tarea automáticamente, por lo que para el proceso de adquisición se recomienda colocarlo en estado “verdadero”.
- **New task name:** Parámetro de tipo string, utilizado para asignar un nombre a la tarea que se encuentra en ejecución; sin embargo, en esta aplicación no fue necesario configurar este parámetro.

- Task to copy: Parámetro para asignar el nombre de una tarea y crear una copia de esta, sin embargo para la aplicación no fue necesario configurarlo.
- Global virtual Channels: Parámetro para especificar el canal físico que se está utilizando, sin embargo en este caso no fue necesario utilizar este parámetro.
- Error in: Parámetro que describe las condiciones o errores que se generan antes de colocar en funcionamiento el DAQmx Create Task.vi
- Error out: Describe el estado de error, que este VI o función genera con respecto a alguna anomalía dentro de este, generalmente va conectado al error in del siguiente bloque de la etapa.
- Task out: Hace referencia a la nueva tarea o salida de datos que se encuentran en ejecución, generalmente este terminal va conectado al terminal “task in” del siguiente bloque de la etapa.

#### 3.1.1.2.2. DAQmx Create Channel.VI

Es un VI polimórfico (que posee varias funciones), utilizado para la creación de canales virtuales; en este caso se implementa para la creación de las entradas análogas referentes a tensión y corriente [13]. Con este VI prácticamente se configura los parámetros de entradas que va tener cada canal y así poder visualizar las señales análogas adquiridas. En la figura 29 se presenta este VI utilizando la función AI-Voltage-Basic para la creación de las entradas de tensión y corriente. Y posteriormente, se presenta una breve descripción de sus parámetros de configuración.

Figura 29. DAQmx Create Channel (AI-Voltage-Basic).VI



### 3.1.1.2.2.1. Parámetros de configuración del bloque DAQmx Create Channel (AI-Voltage-Basic).VI

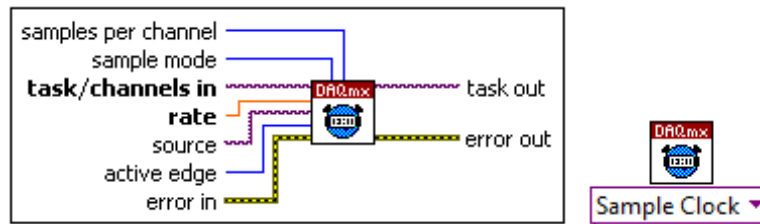
- Input terminal configuration: Parámetro que especifica el tipo configuración que tendrá el terminal de entrada para la creación de los canales virtuales; las configuraciones con que cuenta este VI son: entrada diferencial, pseudodiferencial y entrada de referencia única, las cuales son argumentadas y especificadas en el mando de ayuda de Labview, en el proceso de adquisición de la aplicación se recomienda utilizar la entrada pseudodiferencial.
- Minimum value: Parámetro que especifica el valor mínimo de medición (en este proyecto se especifico un valor minimo de  $-10$  debido a que la señales a medir son superiores a este valor).
- Maximun value: Parámetro que especifica el valor máximo de medición (en este proyecto se especifico un valor maximo de  $10$  debido a que la señales a medir son inferiores a este valor).
- Task in: Parámetro que especifica la tarea, que se asignará a los canales virtuales creados. Por lo general este terminal se encuentra conectado con el task out del bloque anterior; en este caso esta conectado al task out del bloque DAQmx Create Task para ir ejecutando las tareas etapa por etapa.
- Physical channels: Parámetro para especificar el numero de los canales físicos, que se usan en la creación de los canales virtuales (En el proyecto se especificó el número "0" para corriente, y el número "1" para voltaje; sin embargo, se puede optar por la selección de otros canales).
- Name to assign: Parámetro para asignar el nombre de cada canal virtual creado. (En este proyecto se asignaron los nombres de corriente y voltaje para cada canal).
- Units: Parámetro que especifica el tipo de unidades a utilizar para generar las lecturas de las muestras obtenidas, en este caso son las unidades de de tensión (Volts) y corriente (Ampere).
- Custom scale name: Parámetro que especifica el tipo escala deseada para las señales bajo estudio (en este proyecto se determino un valor de escala de 0).

- Error in: Parámetro que describe las condiciones o errores que se generan antes de colocar en funcionamiento el DAQmx Create Channel.
- Error out: Describe el estado de error, que este VI o función genera con respecto a alguna anomalía dentro de este.
- Task out: Hace referencia a la nueva tarea o salida de datos que se encuentran en ejecución, generalmente este terminal va conectado al terminal “task in” del siguiente bloque de la etapa.

#### 3.1.1.2.3. DAQmx Timing (Sample Clock).VI

Es un VI polimórfico utilizado para controlar el número y la velocidad con la que son adquiridas las muestras, en el desarrollo de la aplicación se empleó este VI en la configuración Sample Clock, permitiendo así configurar el número de muestras por adquisición o generación, según los requerimientos dados por el usuario [13]. En la figura 30 se presenta este VI con sus terminales de entradas y salida. Y posteriormente, una breve descripción de sus parámetros de configuración.

Figura 30. DAQmx Timing (Sample Clock).VI



##### 3.1.1.2.3.1. Parámetros de configuración del bloque DAQmx Timing (Sample Clock).VI

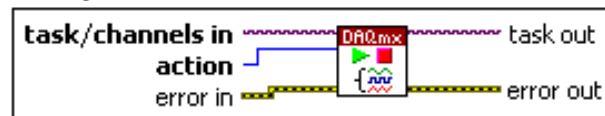
- Sample per chanel: Parámetro que especifica el número de muestras para adquirir o generar en cada canal creado; si el número de muestras que se dan son continuas, el NI DAQmx utiliza este valor para determinar el tamaño del buffer, en este proyecto se designó el valor de 2E6 con el fin de poder adquirir mayor numero de muestras que las tomadas con Matlab (1E6), y poder tener así una mayor precision en la obtencion de la curva.

- Sample mode: Parámetro que determina si el número de muestras a adquirir o a generar se dan en forma continua o se genera en un número finito, en la etapa de adquisición se recomienda colocar que las muestras se obtienen en un número finito.
- Task/channels in: Nombre de la tarea o lista de canales virtuales seleccionados a los que se le desea aplicar la función de este VI, este parámetro permite controlar la adquisición y velocidad del número de muestras adquiridas, en la aplicación se recomienda conectarlo al task out del DAQmx create channel.
- Rate: Parámetro que especifica la velocidad de muestreo por canal en unidades de muestras por segundo; en este proyecto se designó una velocidad de 10E6 debido a que es la velocidad con la que cuenta la tarjeta de adquisición.
- Source: Parámetro para especificar el terminal donde se encuentra la alimentación para el funcionamiento del reloj, por lo general esta entrada se deja sin conexión para utilizar el valor por defecto a bordo del reloj del dispositivo.
- Active edge: Parámetro que especifica en cual flanco de los pulsos, se desea adquirir o generar las muestras. En dicho parámetro se pueden escoger los flancos ascendentes o los flancos descendentes para el muestreo de datos. Sin embargo, este parámetro Active Edge no fue necesario tenerlo en cuenta en el desarrollo de la adquisición de señales.
- Error in: Parámetro que describe las condiciones o errores que se generan antes de colocar en funcionamiento el DAQmx Timing (Sample Clock).VI.
- Error out: Describe el estado de error, que este VI o función genera con respecto a alguna anomalía dentro de este.
- Task out: Hace referencia a la nueva tarea o salida de datos que se encuentran en ejecución, generalmente este terminal va conectado al terminal "task in" del siguiente bloque de la etapa.

#### 3.1.1.2.4. DAQmx Control Task.VI

Este VI modifica la operación de una tarea de acuerdo con la acción que se especifique, esta acción puede ser la verificación de parámetros, cancelación de la tarea, reprogramar la configuración del hardware, entre otros [13]; básicamente este VI ejerce un control sobre la tarea que está siendo ejecutada. En la figura 31 se presenta este VI con sus terminales de entradas y salida. Y posteriormente, se presentan sus parámetros de configuración.

Figura 31. DAQmx Control Task.VI



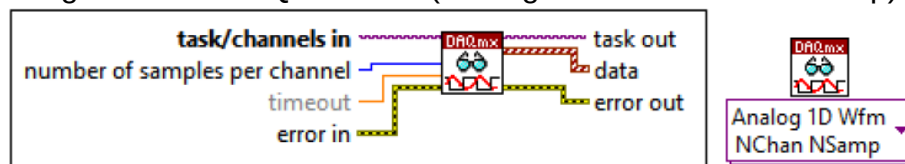
##### 3.1.1.2.4.1. Parámetros de configuración del bloque DAQmx Control Task.VI

- Task/channels in: Parámetro que permite seleccionar el nombre de la tarea o lista de canales virtuales a los que se le desea aplicar la función de este VI , en el proceso de adquisición se recomienda conectar este parámetro con el task out del DAQmx Timing.
- Action: Especifica la forma en que se modificará la tarea que se encuentra en ejecución, utilizando acciones de control como: Verificar, guardar, programar, liberar y cancelar la tarea en ejecución. En la aplicación se recomienda colocar “commit” (programar), para obtener todas las muestras adquiridas.
- Error in: Parámetro que describe las condiciones o errores que se generan antes de colocar en funcionamiento el DAQmx Control Task.VI
- Error out: Describe el estado de error, que este VI o función genera con respecto a alguna anomalía dentro de este.
- Task out: Hace referencia a la nueva tarea o salida de datos que se encuentran en ejecución, generalmente este terminal va conectado al terminal “task in” del siguiente bloque de la etapa.

#### 3.1.1.2.5. DAQmx Read (Analog 1D Wfm NChan NSamp).VI

Es un VI polimórfico, que permite leer uno o múltiples canales a la vez para posteriormente mostrarlos a la salida como una función gráfica [13], de modo que, esta herramienta permite leer y procesar una muestra de cada canal (tensión y corriente) para posteriormente, mostrarlos mediante un indicador gráfico. En la figura 32 se presenta este VI utilizando la función Analog 1D Wfm NChan NSamp para la lectura de los canales de tensión y corriente. Y posteriormente se presenta sus parámetros de configuración.

Figura 32. DAQmx Read (Analog 1D Wfm NChan NSamp)



##### 3.1.1.2.5.1. Parámetros de configuración del bloque DAQmx Read (Analog 1D Wfm NChan NSamp).

- Task/channels in: Parámetro que permite seleccionar el nombre de la tarea o lista de canales virtuales, a los que se le desea aplicar la función de este VI, en este proyecto se recomienda conectar este parámetro con el task out del DAQmx Control Task.
- Number of samples per channel: Parámetro que especifica el número de muestras a leer, si dicho parámetro se deja sin conexión o se le asigna como constante -1, permite que el NI-DAQmx determine el número de muestras de manera automática. En este caso se recomienda colocarlo en -1.
- Timeout: Parámetro que especifica el tiempo en segundos en el que las muestras estarán disponibles. si se establece el tiempo de espera en -1 el VI esperará indefinidamente, si se establece el tiempo de espera en 0, el VI intentará leer las muestras solicitadas de manera inmediata; sumado a esto; el timeout puede generar un error en el VI en caso de no ser capaz de realizar dicha tarea por lo que es recomendable configurarlo con valores cercanos a cero (0.6, 0.5, 0.4).

- Data: Parámetro de salida que genera una matriz de datos de una dimensión, referentes a las señales leídas. En este caso se obtienen los datos de las señales de tensión y corriente, para posteriormente mostrarlas en un indicador gráfico.
- Error in: Parámetro que describe las condiciones o errores que se generan antes de colocar en funcionamiento el DAQmx Read.VI
- Error out: Describe el estado de error, que este VI o función genera con respecto a alguna anomalía dentro de este.
- Task out: Hace referencia a la nueva tarea o salida de datos que se encuentran en ejecución, generalmente este terminal va conectado al terminal “task in” del siguiente bloque de la etapa, en este caso va conectado al task/channels in del bloque DAQmx stop task.

#### 3.1.1.2.6. DAQmx Wait Until Done.VI

Este VI permite asegurar que la adquisición de datos sea completada, antes de limpiar las tareas (liberar recursos en el buffer), para ello este VI cuenta con la entrada timeout que especifica el tiempo de espera en segundos, antes de que los recursos sean liberados; para esta aplicación se recomienda dejarlo en 40. En la figura 33, se presenta este VI con sus terminales de entrada y salida. Sus parámetros de configuración, “error in, error out, task out”, tienen la misma función que los descritos en el bloque DAQmx Control Task.VI.

Figura 33. DAQmx Wait Until Done.VI



#### 3.1.1.2.7. DAQmx Stop Task.VI

Este VI permite detener la tarea o proceso y devolverla al estado inicial. En la figura 34, se presenta este VI con sus terminales de entrada y salida. Sus parámetros de configuración, “error in, error out, task out”, tienen la misma función que los descritos en el bloque DAQmx Control Task.VI.

Figura 34. DAQmx Stop Task.VI



#### 3.1.1.2.8. DAQmx Clear Task.VI

Este VI libera los recursos de la memoria temporal evitando que el buffer se sature. En la figura 35, se presenta este VI con sus terminales de entrada y salida. Sus parámetros de configuración, “error in, error out, task int”, tienen la misma función que los descritos en el bloque DAQmx Control Task.VI.

Figura 35. DAQmx Clear Task.VI



#### 3.1.2 Etapa de filtrado.

Labview cuenta con herramientas para el procesamiento de aplicaciones, estas se basan en el análisis avanzado de señales, dentro de las que se encuentran la toolset de procesamiento de señales (SPT), esta toolset implementa herramientas avanzadas para el procesamiento de señales en las que la transformada de Fourier no es la más indicada, se divide en tres grandes grupos.

- **Análisis tiempo-frecuencia:** Toolkit que contiene algoritmos y herramientas para analizar el contenido espectral de señales no estacionarias, incluyendo algoritmos como la transformada ventana de Fourier, la transformada Gabor, la distribución Choi-Williams, la distribución Winger-Ville entre otros [61].
- **Análisis de series temporales:** Toolkit que se emplea en aplicaciones donde se necesita realizar predicciones lineales, compresión de datos, y estimaciones de datos este toolkit contiene algoritmos para el análisis estadístico como la media, la varianza, matriz de correlación, estimación espectral para analizar las señales en el dominio de la frecuencia [61].
- **Análisis wavelet toolkit:** Empleado para la extracción de señales de corta duración incluyendo algoritmos o VIs como la transformada wavelet

continua, la wavelet discreta, los tipos de wavelet madre o bancos de filtros entre otras funciones [61].

Para el desarrollo de la etapa de filtrado, se implementó el toolkit de análisis wavelet, la cual es muy usada en el procesamiento de análisis de señales transitorias, para el desarrollo de la aplicación se implementó esta herramienta para extraer los pulsos de la señal de 60 Hz y eliminar el ruido eléctrico presente en ellos, a continuación en la figura 36 se menciona la secuencia de los Vis involucrados en el desarrollo de esta etapa; y posteriormente se describe el funcionamiento de estos.

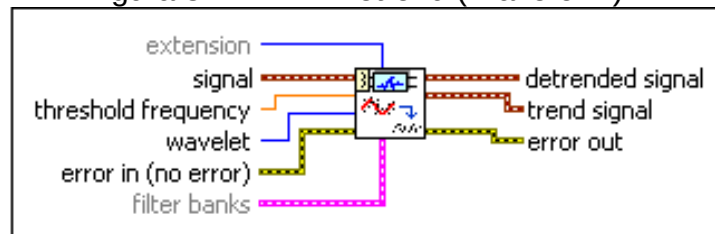
Figura 36. Secuencia de bloques VIs que contiene la etapa de filtro.



#### 3.1.2.1 WA Detrend. VI

Es un VI polimórfico (que posee varias funciones), el cual se encarga de eliminar tendencias de distorsión de la señal, mediante la transformada wavelet discreta (DWT), utilizando coeficientes de aproximación a cero. En este caso este VI eliminará toda señal que se encuentre dentro de la frecuencia de 60 Hz. El tipo de señal de entrada que se encuentre en el parámetro “signal”, determinara la clase de función que se escogerá en este bloque. (Si el parámetro “signal” es un tipo de dato waveform la función que se debe escoger es “WA Detrend (Waveform)” [13]. En la figura 37 se presenta el WA Detrend con sus terminales de entrada y salida, y posteriormente sus parámetros de configuración.

Figura 37. WA Detrend (Waveform).VI



##### 3.1.2.1.1. Parámetros de configuración del bloque WA Detrend VI

- Extensión: Especifica el método en que se utilizarán los datos de entrada, seleccionar el tipo de método realizará la transición entre la señal de

entrada, y los datos que se desean extraer lo más suave posible, una transición suave genera un menor número de coeficientes en gran detalle y optimiza la eficiencia de la representación de la señal, para ello este terminal cuenta con las opciones: “zero padding, symmetric y periodic” que se implementan según el tipo de señal y requerimiento de la aplicación. Estos son mencionados de manera más detallada en el manual de ayuda de Labview, en la aplicación de filtrar la señal de 60 Hz se recomienda colocarlo en la opción “periodic”.

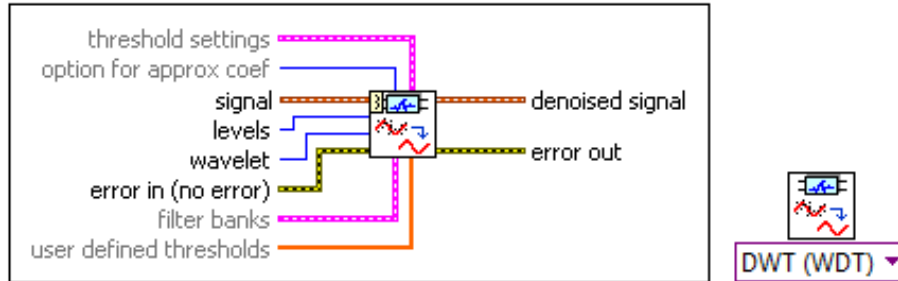
- Signal: Parámetro que especifica la señal de entrada, en este caso se conecta al parametro data del DAQmx Read.
- Threshold frequency: Especifica el límite superior de frecuencia en el que se elimina toda señal que este contenida dentro de este valor. La frecuencia umbral determina el nivel de la transformada wavelet. Si el valor asignado es -1 significa que este VI establece la frecuencia umbral de manera automática, (para este proyecto se configuró este parámetro con un valor de 600, eliminando así la señales no deseadas inferiores a esta referencia).
- Wavelet: Parámetro que especifica el tipo de wavelet a usar como la Haar, la Daubechies, Coiflets, symmlets, o la biortogonal. Las wavelets ortogonales como la Daubechies son ideales para la eliminación de ruido y compresión de una imagen, en la aplicación se implementó la Daubechies “db10”, la cual fue utilizada en la aplicación de matlab.
- Error in: Parámetro que describe las condiciones o errores que se generan antes de colocar en funcionamiento el WA Detrend (Waveform).VI
- Filter Banks: Especifica los bancos de filtros que se usarán para la descomposición de la onda, dichos bancos utilizan el análisis de multirresolución de la wavelet discreta, para descomponer las señales temporales en datos de baja y alta frecuencia; este parámetro se compone de dos subíndices que son; el “Analysis filters”, que determina si el análisis de los coeficientes será en filtros paso bajos o filtros pasó altos y el “Synthesis filters” que especifica la síntesis de coeficientes en los filtros paso bajo o filtros paso altos. Sin embargo para el desarrollo de la aplicación no fue necesario configurar este parámetro.

- Detrend signal: Genera la señal de salida (señal filtrada), en este caso devuelve los pulsos que se inyectaron sin la señal de 60 Hz.
- Trend signal: Devuelve la tendencia residual de la señal.
- Error out: Describe el estado de error, que este VI o función genera con respecto a alguna anomalía dentro de este.

### 3.1.2.2 WA Denoise. VI

Es un VI polimórfico, que se encarga de eliminar el ruido eléctrico presente en las señales, valiéndose de la transformada wavelet discreta (DWT) como medio para generar una señal más limpia y de mejor procesamiento. Sin embargo, se debe tener en cuenta el tipo de señal de entrada, ya que de esta dependerá la selección de la función que tendrá este VI. (Si la señal de entrada es un tipo de dato waveform la función que se debe escoger es “WA DWT (Waveform)” [13]. En la figura 38 se presenta este VI utilizando la función WA DWT (Waveform). Y posteriormente sus parámetros de configuración.

Figura 38. WA Denoise VI



#### 3.1.2.2.1. Parámetros de configuración del bloque WA Denoise.VI

- Threshold settings: Parámetro de clase cluster (que posee varios tipos de datos), el cual especifica el tipo de umbral o límite que se usará para obtener los coeficientes de descomposición wavelet de la señal. Este parámetro se compone de dos subíndices el “Thresholding rule”, que especifica el tipo de selección del umbral (límite), y el “Rescaling method”, que establece el tipo de método para estimar los coeficientes, estos dos subíndices son mencionados de manera más detallada en el manual de ayuda de Labview. En la aplicación se recomienda asignar al subíndice “Thresholding rule” la opción “universal”, y al subíndice “Rescaling method”

la opción “single”, en los anexos se puede observar de manera más detallada estos dos subíndices con sus asignaciones.

- Option for approx coef: Especifica el tipo de operación que se le aplica a los coeficientes, a partir de la descomposición de la wavelet. Para ello este terminal cuenta con tipos de operaciones tales como: “None, Threshold y Detrend”, dichas operaciones determinan si el valor de dichos coeficientes se mantendrá igual, se aproximarán a cero o se llevarán al umbral de estos. Si se especifica un determinado valor; este VI omite la configuración realizada, sin embargo en esta etapa se recomienda utilizar el “threshold” para así eliminar el ruido eléctrico.
- Signal: Parámetro que especifica la señal de entrada, en la aplicación va conectado al “Detrend signal” del bloque WA Detrend.
- Levels: Parámetro que especifica el número de niveles en el análisis wavelet discreta, en este proyecto se utilizó el nivel 10, el cual fue implementado en la aplicación de matlab.
- Wavelet: Parámetro que especifica el tipo de wavelet a usar como la Haar, la Daubechies, Coiflets, symmlets, o la biortogonal, las wavelets ortogonales como la Daubechies son ideales para la eliminación de ruido y compresión de una imagen en este proyecto se implementó la Daubechies “db 10” la cual se implementó en la aplicación de matlab.
- Error in: Parámetro que describe las condiciones o errores que se generan antes de colocar en funcionamiento el WA Denoise VI
- Filter Banks: Especifica los bancos de filtros para la onda, si se especifica un valor en este terminal, el VI omite la configuración de la señal de entrada. Este parámetro se compone de dos subíndices; el “Analysis filters”, que determina si el análisis de los coeficientes será en filtros paso bajos o filtros paso altos y el “Synthesis filters”, que especifica la síntesis de coeficientes en los filtros paso bajo o filtros paso altos, sin embargo para el desarrollo de la aplicación no fue necesario configurar este parámetro.
- User defined thresholds: Especifica el límite para cambiar la escala de los coeficientes de la señal; sin embargo, para propósitos de la aplicación no fue necesario configurar este parámetro.

- Denoised signal: Genera la señal de salida sin el ruido eléctrico.
- Error out: Contiene información del error generado durante la ejecución de este VI.

A continuación se menciona la etapa para extraer los pulsos en el instante de tiempo en que estos se generan.

### 3.1.3 Ventana de análisis.

Para la realización de la prueba FRA se requiere analizar los pulsos inyectados, para ello se hace necesaria una ventana de análisis en la que se identifique el momento en el cual estos aparecen en la señal, esto con el fin de liberar recursos de máquina y optimizar el proceso; permitiendo así, que dichos pulsos sean procesados de manera más rápida. A continuación se describe la función del VI que se implementó.

#### 3.1.3.1 Trigger and Gate express. VI

Este VI se encarga de usar la función Trigger and gate, para extraer o determinar una ventana de análisis en la señal bajo estudio, las condiciones de activación pueden estar basadas en un umbral de inicio o parada, cuando una condición Trigger es estática, el trigger se ejecuta inmediatamente generando un número predefinido de muestras.

Este instrumento virtual presenta dos condiciones para su configuración.

La primera es el cuadro de dialogo “Dialog Box Options”, que se genera, al momento de llamar a este bloque, ver figura 39; y la segunda condición son los parámetros de entrada que trae consigo este VI, ver figura 40.

Figura 39. Dialog Box Options del Trigger and Gate Express. VI

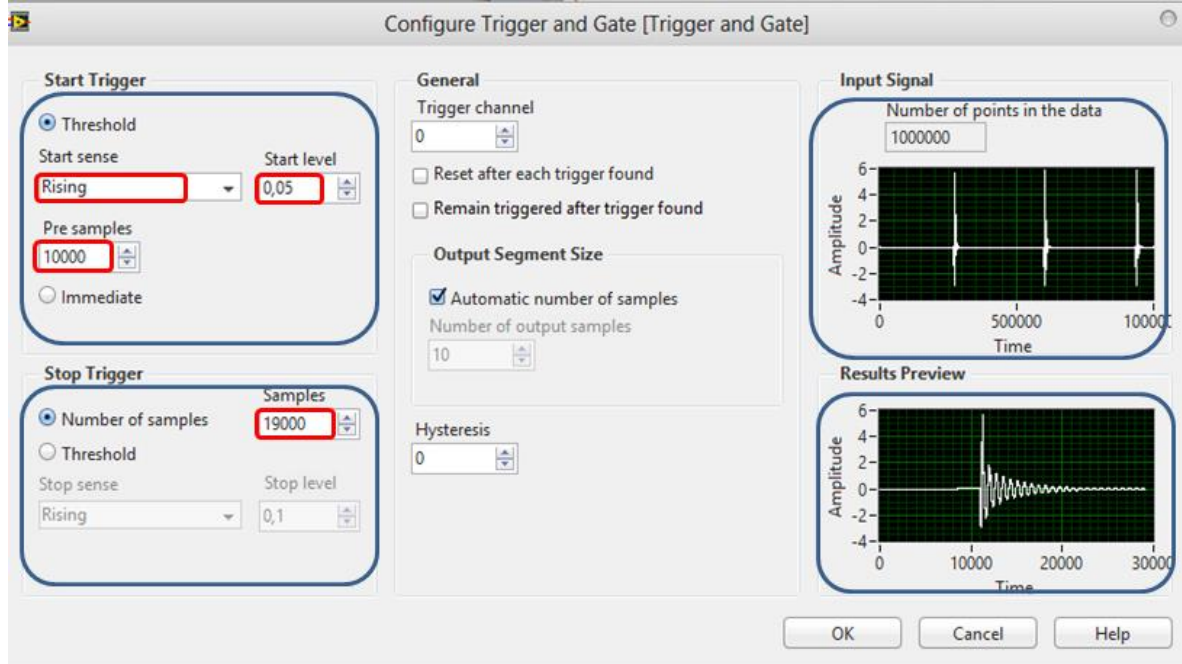
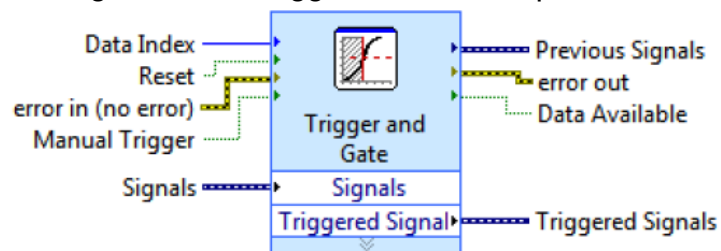


Figura 40. Trigger and Gate express. VI



A continuación se menciona la primer condición para la configuración del trigger and gate el “Dialog Box Options”, y posterior a ello, la segunda condición, la cual hace referencia a los parámetros entrada.

#### 3.1.3.1.1. Parámetros de configuración del bloque Trigger and Gate Express.

##### 3.1.3.1.1.1. Dialog Box Options.

Cuadro de dialogo que trae consigo un conjunto de Parámetros que se generan al momento de invocar este VI, y que son necesarios para el funcionamiento del bloque, estos se visualizan en la figura 39. Este cuadro de dialogo cuenta con dos subíndices; el Star trigger y el Stop trigger.

- Start trigger: Conjunto de parámetros que especifica el punto de referencia en el que se activa la ventana de análisis o Trigger and Gate. Está compuesto por los siguientes parámetros: el “threshold”, el “start sense”, el “start level”, las “pre samples” y el parametro “immediate”, estos se muestran en la figura 39 con sus respectivos valores de configuración; sin embargo, la función de cada uno de estos parámetros se menciona de manera detallada en el manual de ayuda de Labview.
- Stop Trigger: Conjunto de parámetros que especifica el punto de referencia en el que se desactiva la ventana de análisis y se guarda el número de muestras recogidas, Está compuesto por dos subíndices; el “number of samples” y el “threshold”, estos se muestran en la figura 39 con su respectivos valores de configuración, la función de estos parámetros es mencionada de manera detallada en el manual de ayuda de Labview.

#### 3.1.3.1.2. Parámetros de entrada del Trigger and Gate express.

Parámetros de entrada necesarios para el funcionamiento del bloque, estos se visualizan en la figura 40.

- Data Index: Contiene el registro de la detección del trigger
- Reset: Parámetro para controlar el arranque interno de este VI
- Error in: Parámetro que describe las condiciones o errores que se generan antes de colocar en funcionamiento el Trigger and Gate Express.
- Manual Trigger: Parámetro para Iniciar el Trigger inmediatamente, e ignorar los ajustes de inicio de activación.
- Signals: Señal de entrada
- Previous Signals: Parámetro de salida, que genera el resultado del último segmento de la señal.
- Data Available: Indica los datos disponibles, necesarios para la ejecución del Trigger.
- Triggered signal: Genera el segmento resultante de la señal que se encuentra entre el inicio del Trigger y el fin de este, es decir la señal de interés.

#### 3.1.4 Procesamiento de datos.

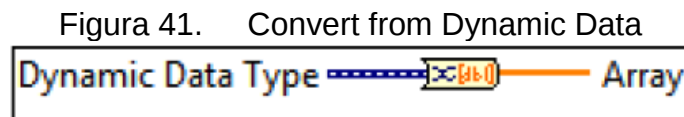
La aplicación software desarrollada en Labview, ha sido diseñada integrando el software de Matlab, con el fin de poder analizar la información de las señales transitorias, y obtener la respuesta en frecuencia. En el capítulo 1 se menciona la transformada wavelet como medio para el desarrollo de la respuesta en frecuencia mediante señales transitorias. Dicha transformada se encuentra incorporada dentro del software de Matlab y Labview.

Sin embargo, en Matlab esta herramienta cuenta con funciones especiales que permiten acondicionar la wavelet continua a fines específicos de este proyecto; para ello se integró el software de Matlab con Labview mediante la herramienta "Matlab script node" la cual es mencionada en capítulo 2, esta herramienta de Labview, permite programar la transformada wavelet para encontrar los espectros de los pulsos. Sin embargo, antes de utilizar dicha herramienta, se requiere que los datos adquiridos y filtrados de manera dinámica, sean convertidos a tipos de dato numérico, esto con el objetivo de utilizar el "Matlab script node".

A continuación se hace referencia a la función que se utilizó para convertir los datos dinámicos a numéricos, y viceversa, permitiendo así que estos sean procesados para obtener la curva FRA. Posteriormente se menciona los pasos necesarios para obtener la respuesta en frecuencia mediante esta transformada, y por último se menciona el VI utilizado para mostrar las curvas obtenidas.

##### 3.1.4.1 Función Convert from Dynamic Data.

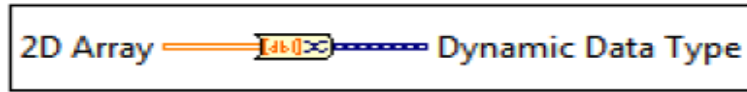
Convierte el tipo de dato dinámico a un tipo de dato numérico, Booleano, forma de onda, tipos de datos de matriz, para su uso con otros VIs e integración con funciones. En la figura 41 se presenta este VI con sus terminales de entrada y salida.



##### 3.1.4.2 Función Convert to Dynamic Data.

Convierte el dato numérico, Booleano, forma de onda o tipos de datos de matriz; al tipo de dato dinámico, para su uso con un Express Vis, (instrumentos que permiten ser configurados mediante un cuadro de dialogo). En la figura 42 se presenta este VI con sus terminales de entrada y salida.

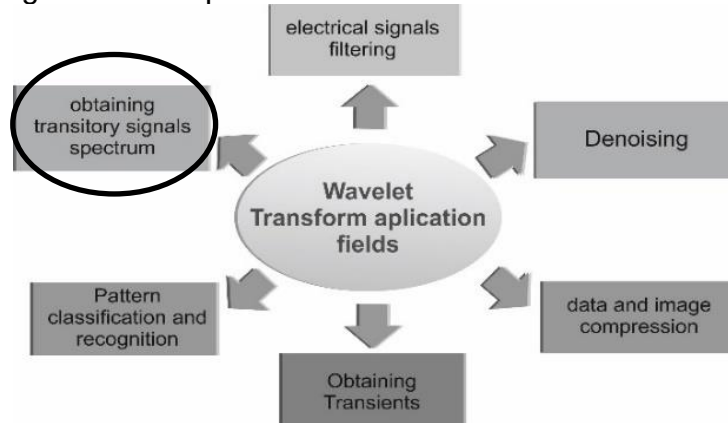
Figura 42. Convert to Dynamic Data



#### 3.1.4.3 Transformada wavelet.

La transformada wavelet es una poderosa herramienta que permite determinar el espectro de una señal transitoria, desarrollo de filtros, procesamiento de imágenes, entre otras aplicaciones, en la figura 43 se presenta los campos de aplicación de esta herramienta.

Figura 43. Aplicaciones de la transformada wavelet



Esta transformada se basa en descomponer una señal en diferentes componentes de frecuencia, pero de tal manera que la descomposición de la señal provenga de una familia de funciones que son traslaciones o dilataciones de una función principal  $(t)$  llamada wavelet madre. [1].

La expresión matemática de esta transformada se ve expresada en la ecuación 1

$$\psi_{\tau,a} = \frac{1}{\sqrt{a}} \psi\left(\frac{t-\tau}{a}\right) \quad (1)$$

Donde:

$a$  : Es el parámetro de dilatación o escalamiento de la función, el cual permite que la señal se dilate o se contraiga al presentarse alguna variación, debido a ello se reduce o aumenta la frecuencia de la señal, obteniendo así la información de los componentes de esta.[62]

$\tau$  : Parámetro de traslación de la función, dicho parámetro nos permite ubicarnos en un lugar específico de tiempo.

Existen dos modalidades de análisis wavelet que se implementan dependiendo de la naturaleza de la señal, estas son la transformada wavelet continua (CWT) y la wavelet discreta (DWT) sin embargo para la obtención de los componentes de frecuencia de los pulsos, se hace referencia a la CWT debido a que la naturaleza de las señales bajo estudio son continuas.

#### 3.1.4.4 Transformada wavelet continua (CWT).

Transformada que permite el análisis de la señal en un segmento localizado en esta, consiste en expresar una señal continua como una expansión de términos o coeficientes del producto interno entre la señal y la wavelet madre [63] , la expresión matemática se ve expresada en la ecuación 2.

$$C(\tau, a) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \psi\left(\frac{t-\tau}{a}\right) dt \quad (2)$$

Dónde:

$\overline{\psi(t)}$  : Indica el complejo de la wavelet madre

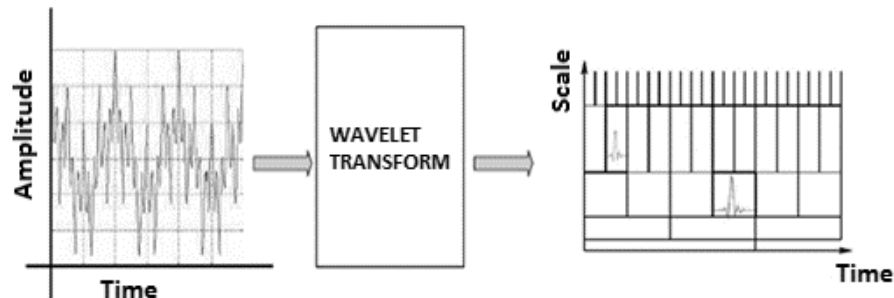
$a$  : Parámetro de dilatación o escalamiento.

$\tau$  : Parámetro de traslación

$x(t)$ : Indica la función de la señal transitoria

La CWT utiliza la wavelet madre como función base para desplazarse y extenderse a través del dominio señal, y así poder llevar la señal del dominio del tiempo al dominio de la escala. Como se indica en la figura 44.

Figura 44. Representación gráfica del análisis de la transformada wavelet



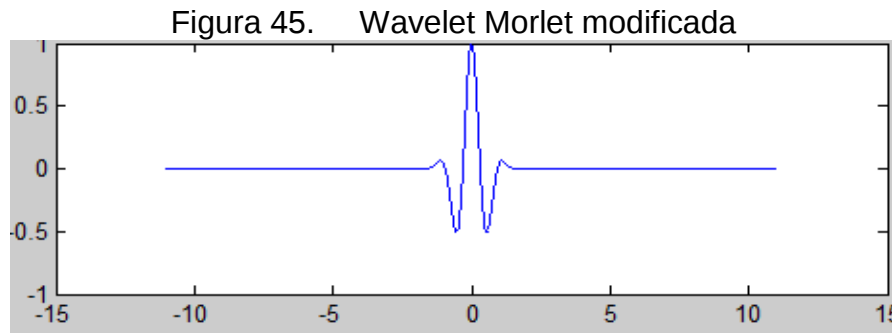
Fuente: (Gómez Luna, Obtención de la respuesta en frecuencia de transformadores en servicio a partir de la medida de señales transitorias, 2013)

#### 3.1.4.5 Wavelet Madre.

Como se mencionó anteriormente la descomposición de la señal se realiza mediante una familia de funciones denominadas wavelets madre, Cada wavelet madre cuenta con unas propiedades específicas que la hacen apropiadas para determinadas aplicaciones. En [1] se mencionan y describen las características y propiedades de cada una de estas. A continuación se menciona la wavelet madre que se utilizó para descomponer las señales transitorias (pulsos) en el desarrollo de la aplicación.

#### 3.1.4.6 Wavelet Madre tipo Morlet modificada.

La wavelet madre denominada “morlet modificada”, es un tipo de wavelet diseñada en [1]. Tiene su forma de onda, basada en el tipo wavelet madre “morlet”; con la ventaja de que esta wavelet diseñada, cuenta con un mayor tiempo de atenuación, logrando así, un ancho de banda más estrecho para generar una menor dispersión en frecuencia y mayor selectividad en las señales transitorias, dicha wavelet madre se presenta de forma general en la figura 45 y ecuación (3).



$$\psi(t) = \cos(2\pi t)e^{-\alpha t^2} \quad (3)$$

#### 3.1.4.7 Desarrollo de algoritmo referente a la wavelet.

Para la obtención de la respuesta en frecuencia, es necesario determinar los espectros de los pulsos inyectados, para ello se requiere trasladar las señales del dominio del tiempo al dominio de la escala, y así poder determinar la energía (espectros) que contienen estos pulsos, a continuación se presenta el proceso para obtener dichos espectros utilizando la herramienta de Labview “Matlab script node” como medio.

Para obtener el espectro de la señal y trasladar correctamente la señal del dominio del tiempo al dominio de la escala es necesario considerar la ecuación (4)

$$a = \frac{fc}{fa \cdot \Delta t} \quad (4)$$

Dónde:

$a$ : es la escala.

$fc$ : es la frecuencia central de la wavelet madre.

$fa$ : es la frecuencia asociada a la escala determinada.

$\Delta t$ : es el periodo de muestreo de la señal a analizar.

Sumado a esto también se debe considerar la señal transitoria  $x(t)$  la cual puede verse como una suma de coeficientes asociados a cada escala wavelet; esta se presenta en la ecuación (5), donde  $F_a(t)$  es la señal analizada en cada nivel de la escala y que es presentada en la ecuación (6).

$$x(t) = \sum_{ai}^{af} F_a(t) \quad (5)$$

$$F_a(t) = \sum_{\tau i}^{\tau f} C(\tau, a) \psi_{a,\tau}(t) \quad (6)$$

Para determinar el espectro de dichas señales se requiere determinar la energía de los coeficientes calculados, la cual es determinada a partir de teorema de Paserval, el cual propone que la energía de la señal en un dominio dado, es igual a la energía de la señal en el dominio de la transformada. Por lo tanto en la ecuación (7) se obtiene la energía asociada a cada coeficiente de escala.

$$E_a = \sum_{\tau i}^{\tau f} C(\tau, a)^2 \quad (7)$$

Finalmente para determinar el espectro de la señal transitoria se obtiene empleando la ecuación (8).

$$\|F_a\| = \sqrt{\sum_{\tau i}^{\tau f} C(\tau, a)^2} \quad (8)$$

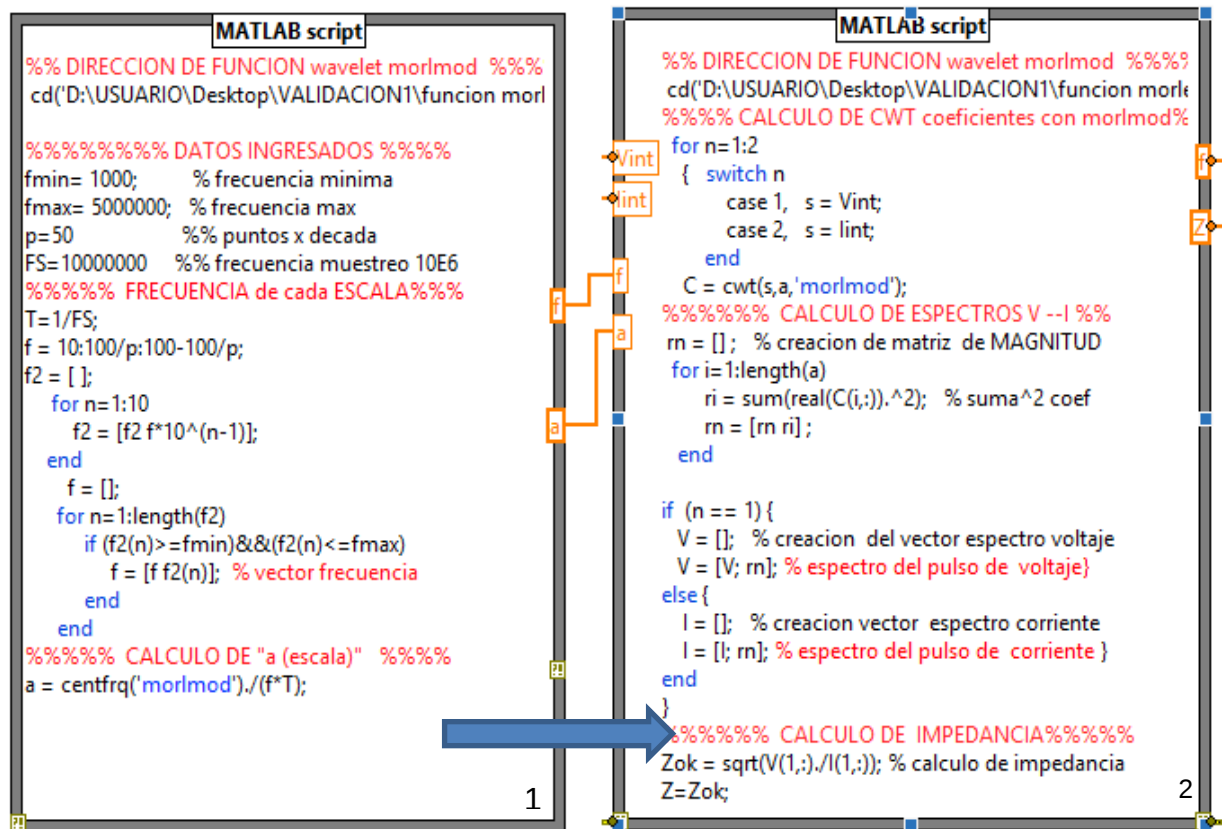
La respuesta en frecuencia puede ser vista como la variación de impedancia o fase, que es sometida a un barrido de frecuencias, por lo que, para determinar la respuesta en frecuencia al transformador, es necesario aplicar el anterior procedimiento a un pulso de tensión  $V_a$  y a un pulso de corriente  $I_a$  obteniendo así la información contenida en estos.

Con base a lo anterior y a partir de los valores de espectros obtenidos para tensión y corriente, la expresión de la respuesta en frecuencia queda definida en (9).

$$\|Z\| = \frac{\|V_a\|}{\|I_a\|} \quad (9)$$

En la figura 46, se presenta el anterior procedimiento utilizando el “Matlab Script Node”, en el cuadro 1 se muestra el algoritmo para determinar la frecuencia asociada a cada escala, también el cálculo del vector escala, y por último las variables de salida “a” (vector escala), y “f” (vector de frecuencia), mientras que en el cuadro 2 se muestra el algoritmo para el cálculo de los coeficientes de los pulsos, el cálculo de los espectros, el cálculo de la impedancia, sus variables de entrada “Vint” (pulsos de tension), e “Iint” (pulsos de corriente), y sus variables de salida “z” (vector de impedancia) y “f” (vector de frecuencia) .

Figura 46. Algoritmo para obtener la curva FRA, mediante el Matlab Script Node

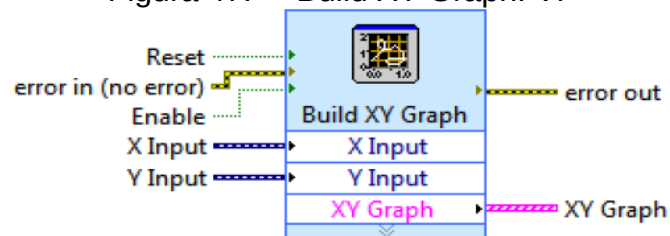


A continuación se presenta el Vi utilizado para mostrar la gráfica de la curva FRA.

### 3.1.4.8 Módulo Build XY Graph.

Módulo que permite que los datos adquiridos y procesados se muestran en un indicador gráfico, de este modo se puede visualizar de manera continua la curva FRA, y los cambios que se presentan en esta. En la figura 47 se presenta este VI con sus terminales en los que se aprecia las entradas (X,Y) y salidas (XY Graph).

Figura 47. Build XY Graph. VI

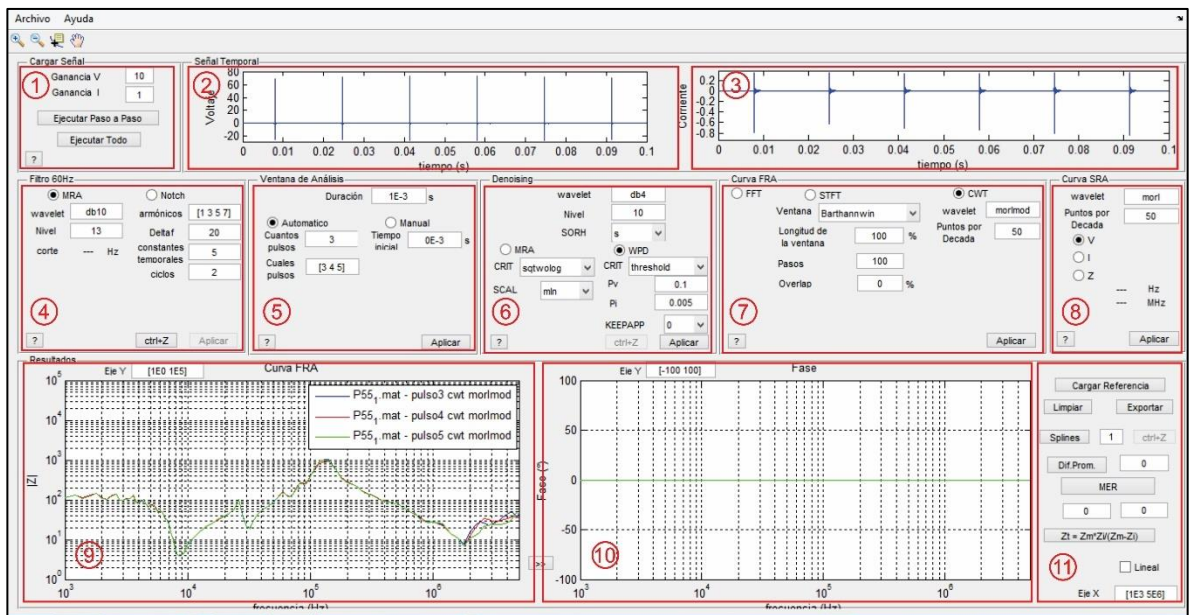


#### 4. DESARROLLO DE PRUEBAS Y ANALISIS DE RESULTADOS

En este capítulo se procede a validar el nuevo método IFRA on-line desarrollado en Labview con la aplicación desarrollada en [1]. El sistema implementado para la validación, es una interfaz gráfica de usuario (GUI), desarrollada en el software de Matlab, esta interfaz realiza el tratamiento de las señales medidas y obtención de la curva FRA de manera offline. Sin embargo, cabe resaltar que los datos bajo estudio con la GUI de Matlab hacen parte de mediciones realizadas con el transformador conectado a una señal de 120 V. Esta aplicación permite tener una referencia al momento de comparar los resultados experimentales obtenidos dinámicamente.

A continuación se presenta el sistema de validación utilizado y el funcionamiento de sus etapas.

Figura 48. Interfaz gráfica utilizada para validar la aplicación desarrollada en Labview



Fuente: (Cuadros Dixon, Identificación de las variaciones en la curva FRA on-line ante cambios internos y externos de un transformador piloto, 2016)

En la figura 48 se muestra el sistema de validación con las etapas utilizadas para el procesamiento de las señales, donde se presentan la etapa de adquisición, filtro y obtención de la curva FRA, a continuación se explica el proceso de la obtención de dicha curva con esta GUI, teniendo en cuenta la enumeración presentada en la figura anterior.

1. Etapa de adquisición: Se cargan las señales de tensión y corriente adquiridas y se les da una ganancia dependiendo de la relación de los sensores de tensión y corriente, [12]
2. Visualización de la señal de tensión: Se aprecia visualmente cada etapa del procesamiento de señal [12].
3. Visualización de la señal de Corriente: Se aprecia visualmente cada etapa del procesamiento de señal, [12].
4. Etapa de eliminación de la señal de 60 Hz: En esta etapa se escoge una wavelet madre, se calcula la frecuencia central de esta y mediante la transformada wavelet discreta en su modalidad de análisis multirresolución (MRA) se elimina la señal de 60 Hz dejando solo las señales transitorias que contienen la información necesaria para la obtención de la respuesta en frecuencia, [12].
5. Etapa de identificación de la ventana de análisis: En esta etapa ya sea de manera automática o manual se realiza la identificación de la ventana de análisis, aquí se trata de obtener solo el contenido espectral necesario para el trazo de la curva FRA, [12].
6. Etapa de eliminación de ruido: En esta etapa hay un filtro que elimina el ruido eléctrico de la ventana de análisis mediante el uso de transformada wavelet ya sea mediante análisis multirresolución (MRA) o la descomposición en paquetes (WPD), [12].
7. Etapa de obtención de FRA: Aquí se traza la curva de respuesta en frecuencia ya sea mediante transformada de Fourier ya sea en su forma rápida (FFT) o en su modalidad de ventana (STFT), o se obtiene la curva de respuesta en frecuencia mediante la transformada Wavelet continua (CWT), [12].
8. Etapa de obtención de la curva de respuesta en la escala SRA: En este espacio es posible trazar la curva de respuesta en función de la escala; la cual es el dominio en el que trabaja la transformada wavelet en su forma continua, este sistema está bajo estudio y gracias a los avances al respecto se tiene esta herramienta para la obtención de la respuesta en la escala, [12].
9. Visualización de la magnitud de la curva de respuesta en frecuencia: En este espacio se visualiza la curva de respuesta en frecuencia en un rango de frecuencias y magnitud variable, [12].
10. Visualización de la fase de la curva de respuesta en frecuencia: En este espacio se visualiza la fase de la respuesta en frecuencia, esta parte está bajo estudio ante el uso de wavelets complejas para la obtención de la fase, [12].
11. Etapa de análisis de los resultados: En esta etapa se permite la comparación de curvas de respuesta en frecuencia ya sea ante una curva de referencia o entre ellas mismas mediante herramientas matemáticas como la diferencia promedio o el error relativo medio (MER). Esta etapa también permite exportar los gráficos de magnitud y fase de la respuesta en frecuencia, [12].

Para la validación del método se desarrollaron una serie de pruebas en la que se sometió al transformador piloto a unos escenarios, con el fin de validar la repetitividad de la curva FRA, y tiempo de obtención de esta para cada aplicación (Matlab y Labview), para ello en las tablas 1 y 2 se presentan las pruebas que se realizaron y con las que se obtuvieron las curvas mediante Labview y Matlab.

Tabla 5. Pruebas realizadas con el transformador conectado a 120 V

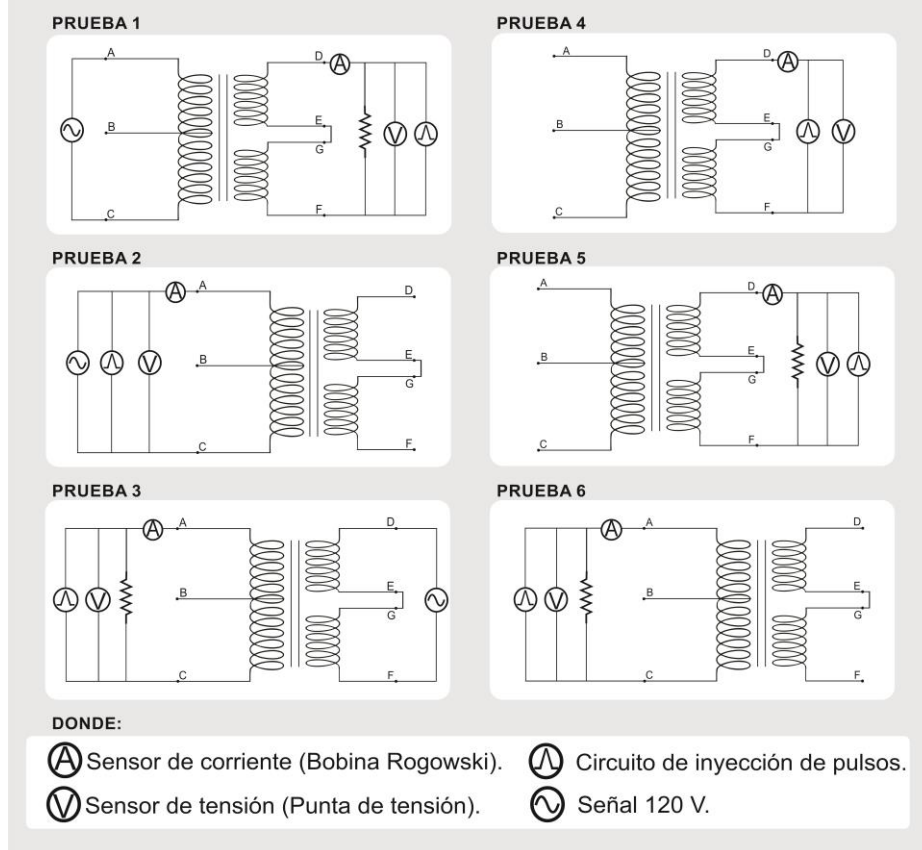
| Prueba | Descripción                                                                                                                                                                |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| P1     | Inyección entre terminales D-F (Devanado secundario) con los terminales D-F en carga y una conexión entre E-G (particiones secundarias en serie). Alimentando A-C          |
| P2     | Inyección entre terminales A-C (devanado primario) con los terminales D-F en circuito abierto y una conexión entre E-G (particiones secundarias en serie). Alimentando A-C |
| P3     | Inyección entre terminales A-C (devanado primario) con los terminales A-C (devanado primario) en carga. Alimentando D-F (Particiones secundarias en serie)                 |

Tabla 6. Pruebas realizadas con el transformador sin conectarse a 120 V

| Prueba | Descripción                                                                                                                                                                              |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| P4     | Inyección entre terminales D-F con una conexión entre E-G (particiones secundarias en serie) y los terminales A-B-C (devanado primario) en circuito abierto.                             |
| P5     | Inyección entre terminales D-F (Devanado secundario) con los terminales D-F en carga y una conexión entre E-G (particiones secundarias en serie). A-C en circuito abierto.               |
| P6     | Inyección entre terminales A-C (devanado primario) con los terminales F-G (partición secundaria) en cortocircuito y A-C (devanado primario) en carga. Terminales D-E en circuito abierto |

Los diagramas esquemáticos de las pruebas realizadas para las tablas 5 y 6, se presentan en la figura 49.

Figura 49. Diagramas Esquemáticos de las pruebas realizadas



#### 4.1 REALIZACIÓN DE PRUEBAS.

En montaje experimental de la prueba IFRA on-line con instrumentación virtual, consta de un transformador piloto, un circuito de inyección de pulsos, un Variac (Generador de señal de 120 V), sensor de medición de tensión (punta de tensión) sensor de corriente (bobina Rogowski), una carga de  $7,3 \Omega$ , un computador con la tarjeta 6115 y su conector; y por último el software de Labview y Matlab.

Los resultados experimentales obtenidos en Labview, son comparados con los resultados obtenidos en la aplicación de Matlab. A continuación se describe y se muestran las gráficas obtenidas en la etapa de adquisición, etapa de filtrado, y la etapa de procesamiento de datos, donde se visualiza la curva FRA teniendo en cuenta los ensayos propuestos en la tabla 5 y tabla 6.

##### 4.1.1 Prueba 1.

Esta primera prueba consistió en inyectar los pulsos en paralelo a una carga resistiva por el lado de baja tensión, manteniendo el lado de alta conectado a 120

V, y realizando las mediciones de tensión y corriente por el lado de baja tensión. En las figuras 50, 51, 52, 53 y 54 se muestran los resultados obtenidos en el transformador, para cada etapa mencionada bajo esta configuración.

#### 4.1.1.1 Adquisición de datos.

En las figuras 50 y 51 se presentan los resultados obtenidos en el proceso de adquisición, para la aplicación de Labview (figura 50), y para la aplicación de Matlab (figura 51).

Figura 50. Señales de tensión y corriente obtenidas de manera dinámica con Labview, prueba 1

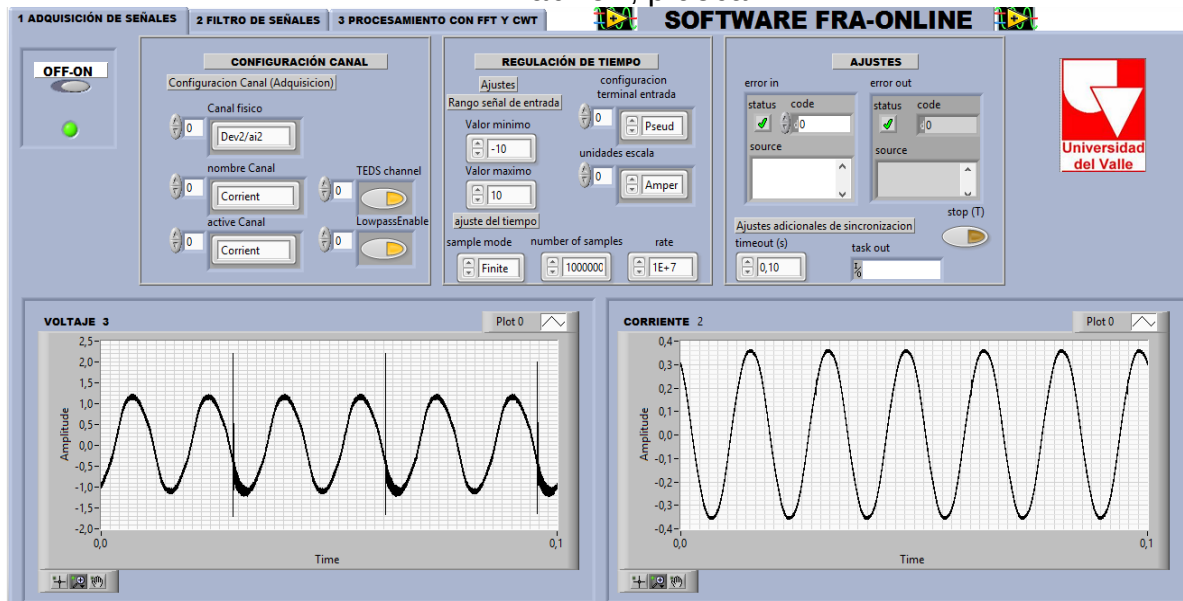
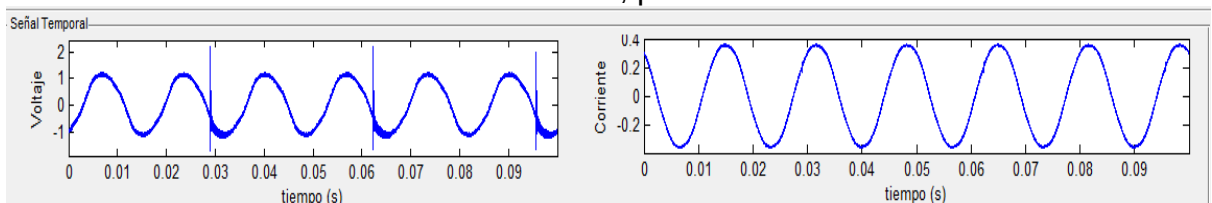


Figura 51. Señales de tensión y corriente obtenidas de manera off-line con Matlab, prueba 1



#### 4.1.1.2 Etapa de filtrado.

En las figuras 52 y 53 se presentan los resultados obtenidos en la etapa de filtrado, para la aplicación de Labview (figura 52), y para la aplicación de Matlab (figura 53).

Figura 52. Pulsos de tensión y corrientes filtrados con Labview, prueba 1.

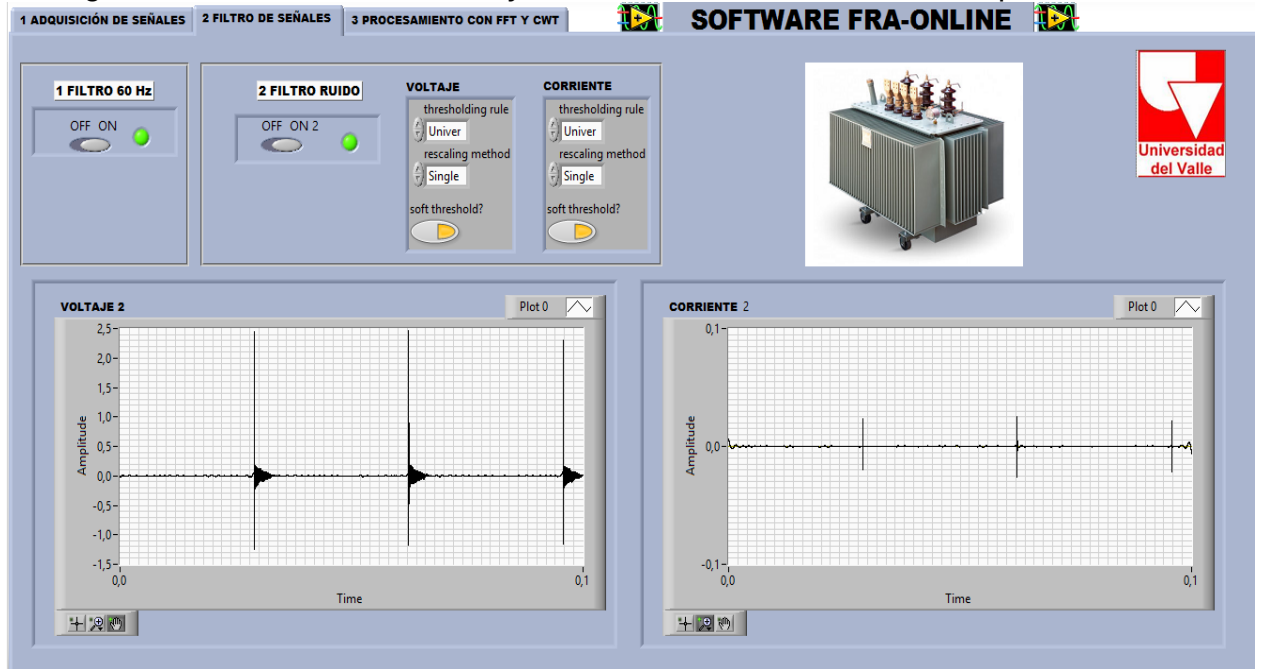
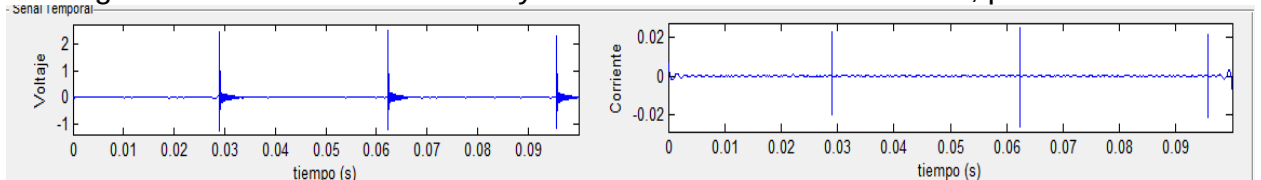


Figura 53. Pulsos de tensión y corrientes filtrados con Matlab, prueba 1.



#### 4.1.1.3 Etapa de procesamiento.

En las figuras 54 y 55, se presentan los resultados obtenidos de las curvas FRA, para la aplicación de Labview (figura 54), y la obtención de la curva mediante la aplicación de Matlab (figura 55).

Figura 54. Curva FRA obtenida de manera dinámica con Labview, prueba 1.

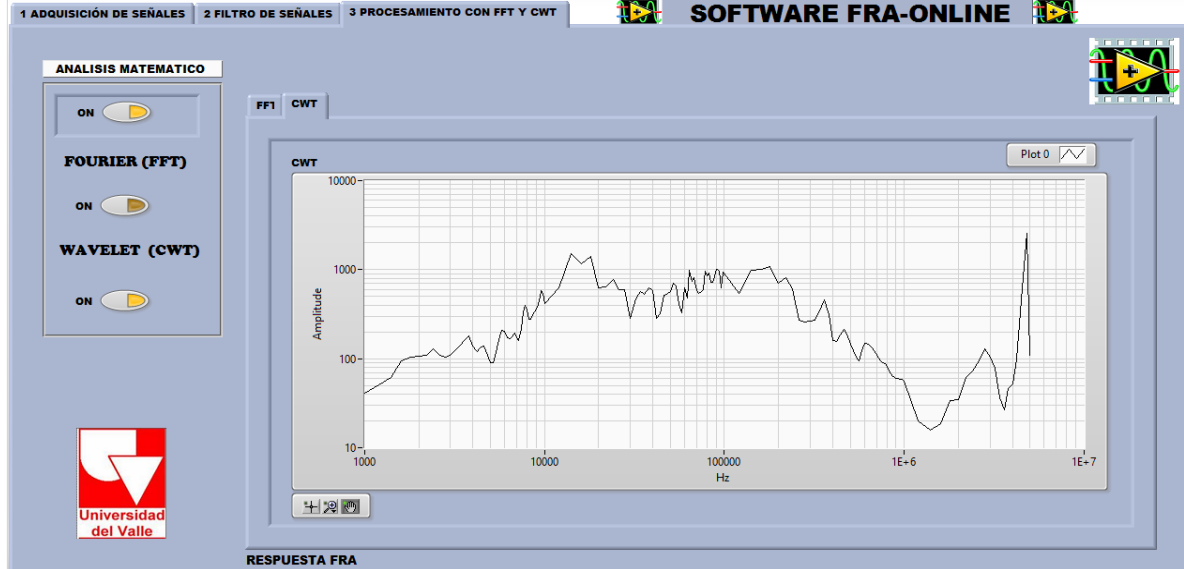
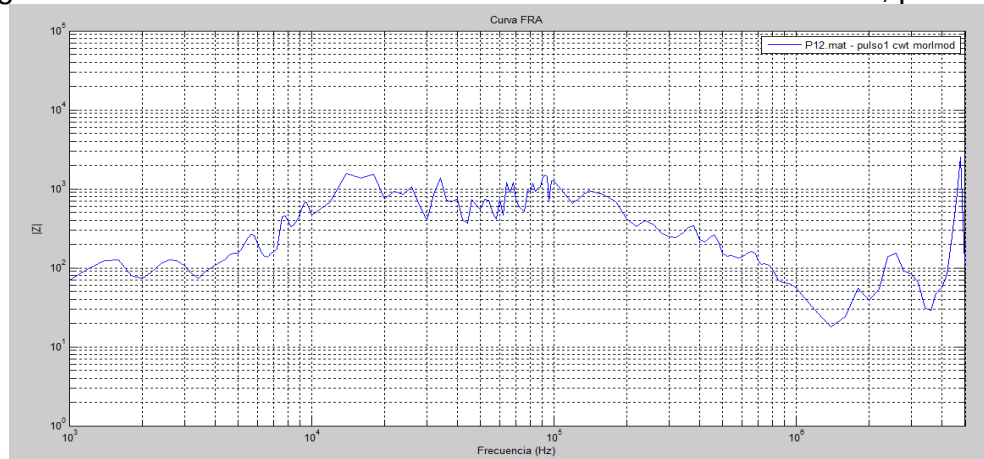


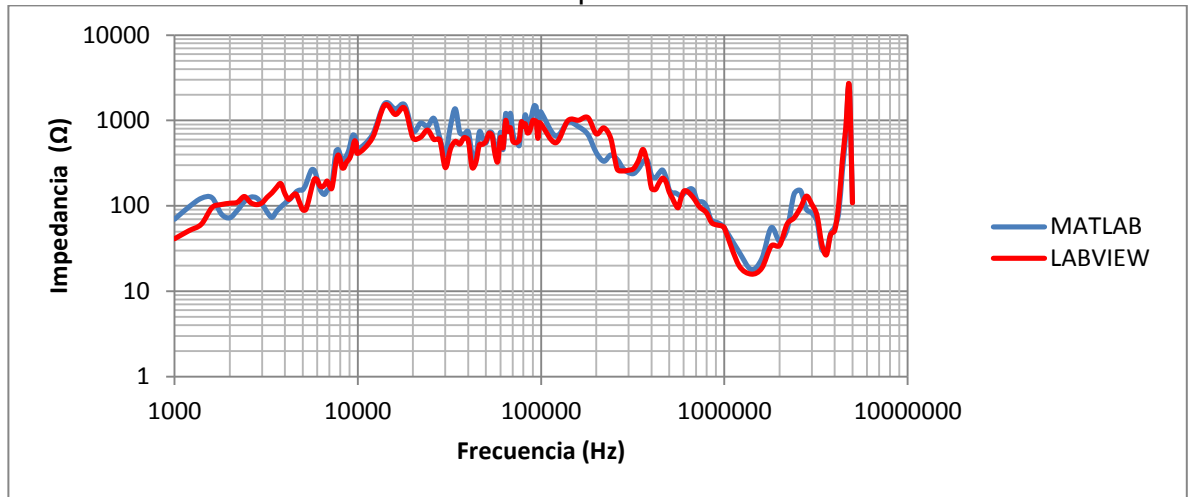
Figura 55. Curva FRA obtenida de manera off-line con Matlab, prueba 1.



De acuerdo con los resultados de la figura 54, y figura 55, en esta primer prueba se observa una buena repetitividad en las curvas obtenidas mediante los dos métodos, además se pudo observar que durante el desarrollo de las pruebas, el tiempo de obtención de la curva en Labview fue aproximadamente 10 segundos, mientras que la obtenida en Matlab fue de 16 minutos.

En la figura 56 se presenta la gráfica de comparación entre la curva obtenida con Labview, y la curva obtenida en Matlab, en esta se puede observar que las dos respuestas presentan una buena repetitividad dentro del rango de frecuencias de los 10 kHz a los 3000 kHz, este alto nivel de similitud entre las dos gráficas, es atribuido a que en esta prueba el transformador se encontraba con carga.

Figura 56. Curvas FRA obtenidas con la aplicación de Matlab y Labview , prueba 1.



#### 4.1.2 Prueba 2.

La segunda prueba, consistió en inyectar los pulsos por el lado de Alta tensión, manteniendo este lado conectado a 120 V y el lado de baja en vacío, las mediciones se realizaron por el lado de alta tensión. Sin embargo, con el fin de resumir el procedimiento, solo se presentarán los resultados obtenidos en la última etapa (etapa de procesamiento), en la que se analiza curva FRA obtenida con Labview y Matlab; en las figura 57, y figura 58 se muestran los resultados obtenidos en el transformador bajo esta configuración.

Figura 57. Curva FRA obtenida de manera dinámica con Labview, prueba 2.

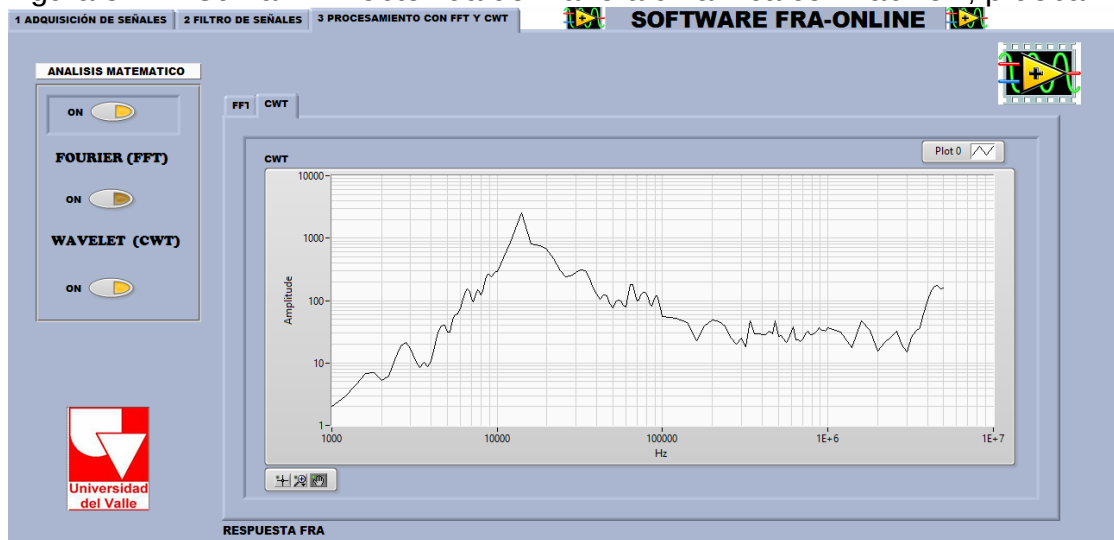
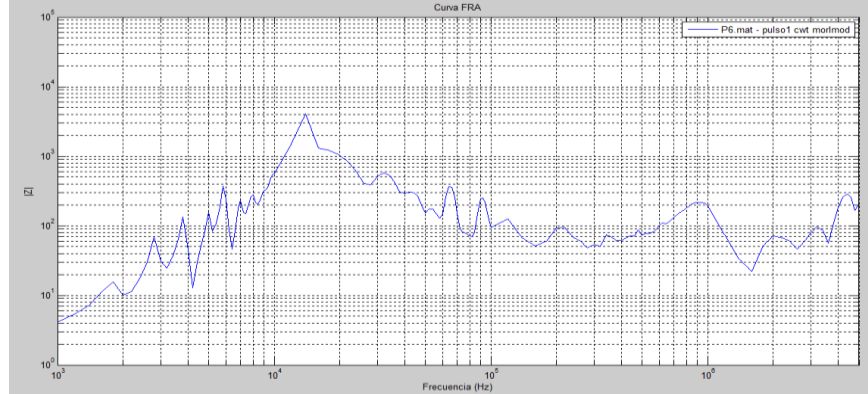


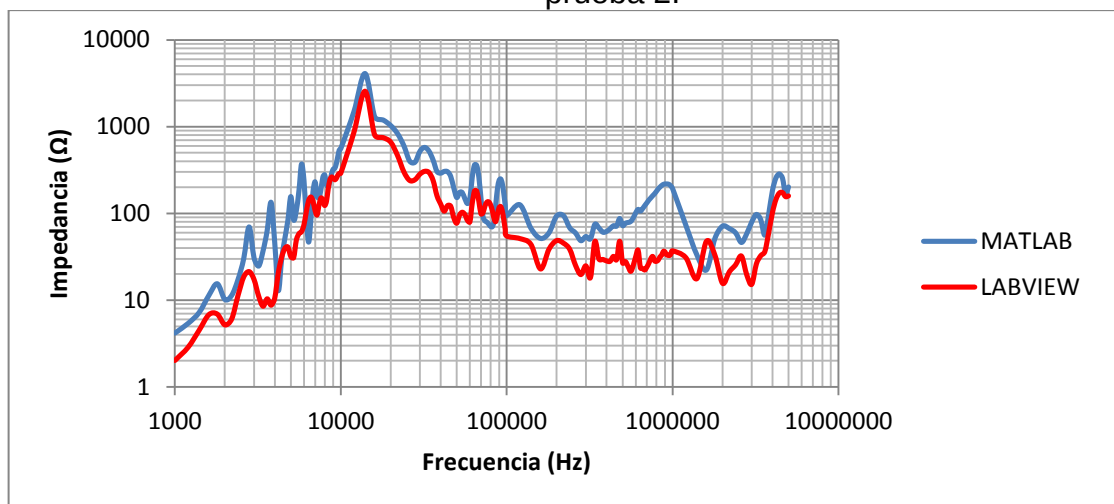
Figura 58. Curva FRA obtenida de manera off-line con Matlab, prueba 2.



De acuerdo con los resultados de la figura 57, y figura 58, en esta segunda prueba se observa algunos cambios en la repetitividad de las curvas obtenidas, además durante el desarrollo de las pruebas se tomó que el tiempo de obtención de la curva en Labview, fue aproximadamente 9 segundos, mientras que la obtenida en Matlab fue de 15 minutos.

En la figura 59 se presenta la gráfica de comparación entre las dos curvas, donde se pueden observar la similitud entre ellas y algunas diferencias en la amplitud y forma de onda; las cuales son atribuidas al procesamiento de los bloques con que cuenta Labview, al tiempo en que se inyectaron los pulsos para cada aplicación, y a la ausencia de carga.

Figura 59. Curvas FRA obtenidas con la aplicación de Matlab y Labview, prueba 2.



#### 4.1.3 Prueba 3.

La tercera prueba, consistió en inyectar los pulsos en paralelo a una carga resistiva por el lado de alta tensión, manteniendo el lado de baja conectado a 120 V, y realizando las mediciones de tensión y corriente por el lado de alta tensión. Sin embargo, con el fin de resumir el procedimiento, solo se presentarán los resultados obtenidos en la última etapa (etapa de procesamiento), en la que se analiza curva FRA obtenida con Labview y Matlab; en las figuras 60, y 61 se muestran los resultados obtenidos en el transformador bajo esta configuración.

Figura 60. Curva FRA obtenida de manera dinámica con Labview, prueba 3.

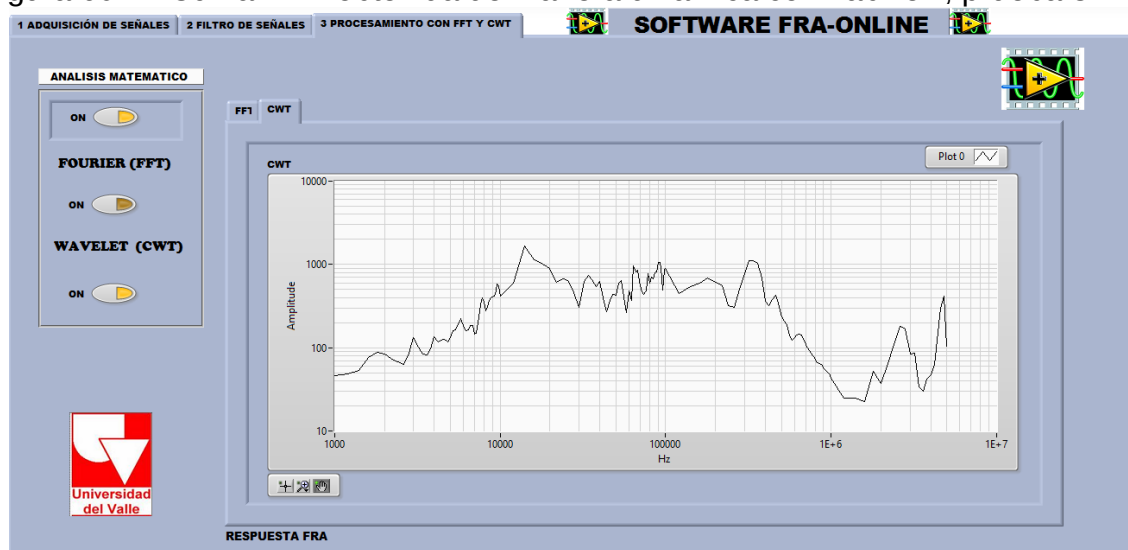
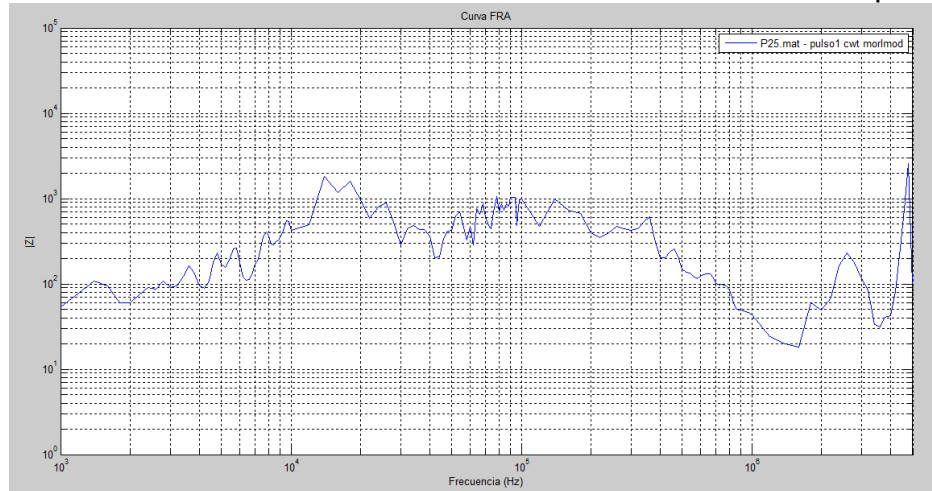


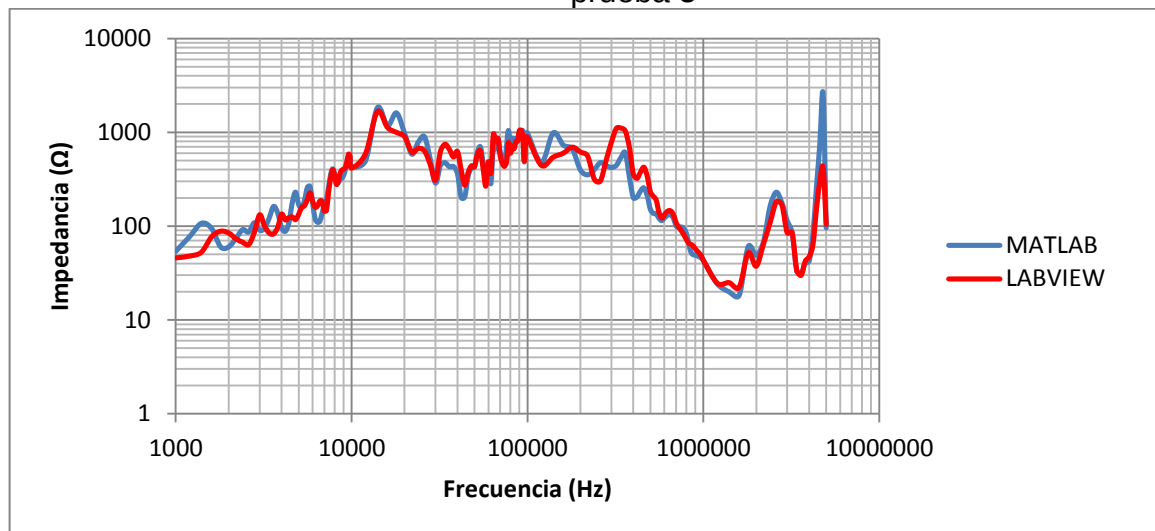
Figura 61. Curva FRA obtenida de manera off-line con Matlab, prueba 3.



De acuerdo con los resultados de la figura 60, y figura 61, en esta tercer prueba se observa, una mejor repetitividad en las curvas obtenidas, además durante el desarrollo de las pruebas se tomó que el tiempo de obtención de la curva en Labview, fue aproximadamente 8 segundos, mientras que la obtenida en Matlab fue de 14 minutos.

En la figura 62 se presenta la gráfica de comparación entre la curva obtenida con Labview, y la curva obtenida en Matlab, donde se observa que las dos respuestas presentan una buena similitud dentro del rango de frecuencias de los 10 KHz a los 4000 KHz, este alto nivel de similitud entre las dos gráficas, es atribuido a que en esta prueba el transformador se encontraba con carga.

Figura 62. Curvas FRA obtenidas con la aplicación de Matlab y Labview, prueba 3



#### 4.1.4 Prueba 4.

La cuarta prueba, consistió en solamente inyectar los pulsos por el lado de baja tensión, manteniendo el lado de alta en vacío, y realizando las mediciones de tensión y corriente, por el lado de baja. En las figuras 63 y 64, se muestran los resultados obtenidos con Labview y Matlab bajo esta configuración.

Figura 63. Curva FRA obtenida de manera dinámica con Labview, prueba 4.

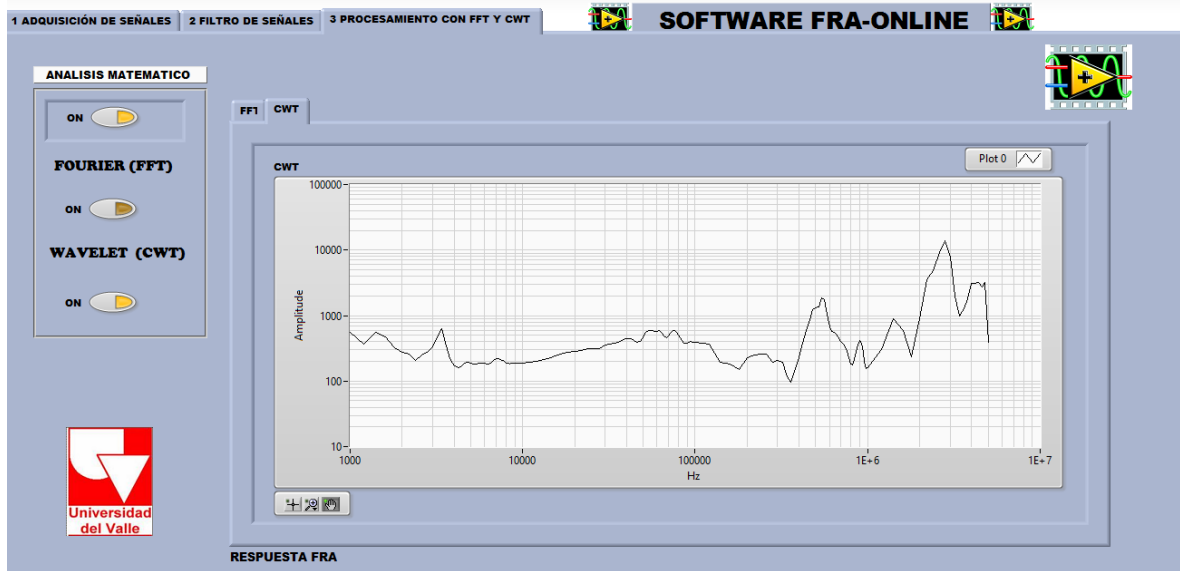
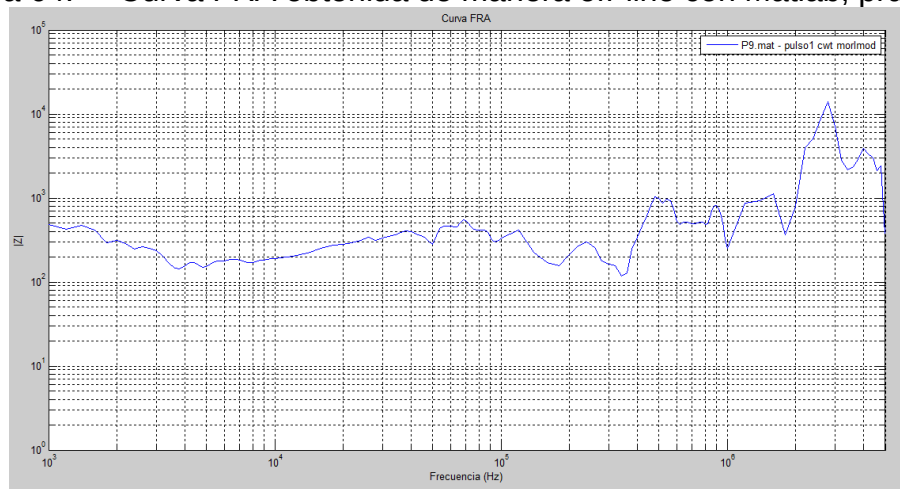
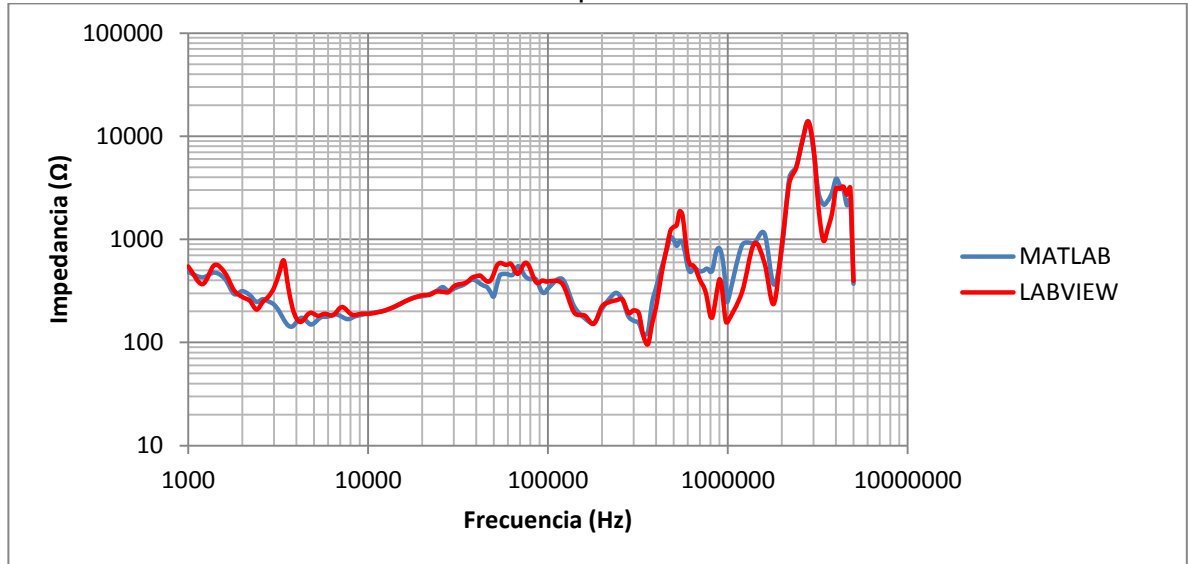


Figura 64. Curva FRA obtenida de manera off-line con Matlab, prueba 4.



En la figura 65 se presenta la gráfica de comparación entre las dos curvas, donde se observa que las dos respuestas presentan una buena similitud dentro del rango de frecuencias de los 1 kHz a los 500 kHz, este bajo nivel de similitud es atribuido al procesamiento de los bloques, al tiempo en que se inyectaron los pulsos, y a la ausencia de carga, sin embargo el tiempo de obtención de la curva para Labview fue de 9 segundos, mientras que para Matlab fue 16 minutos.

Figura 65. Curvas FRA obtenidas con la aplicación de Matlab y Labview, prueba 4.



#### 4.1.5 Prueba 5.

La quinta prueba, consistió en inyectar los pulsos en paralelo a una carga resistiva por el lado de baja tensión, manteniendo el lado de alta en vacío, y realizando las mediciones de tensión y corriente por el lado de baja. En las figuras 66 y 67 se presentan los resultados obtenidos en la última etapa (etapa de procesamiento), en la que se analiza curva FRA obtenida con Labview y Matlab.

Figura 66. Curva FRA obtenida de manera dinámica con Labview, Prueba 5.

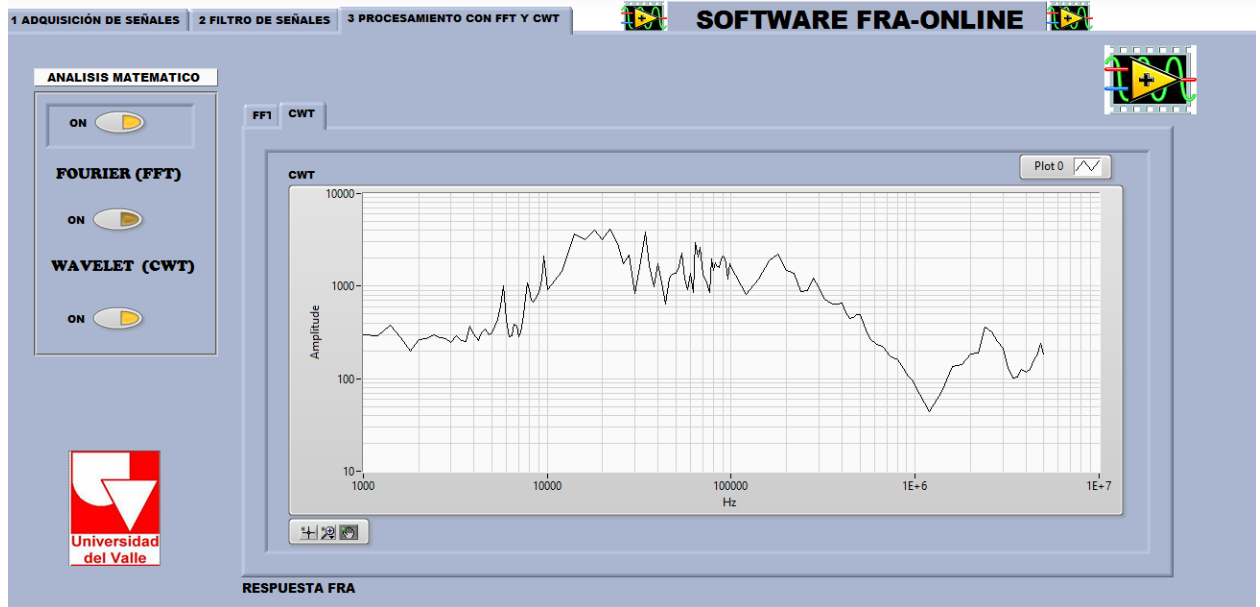
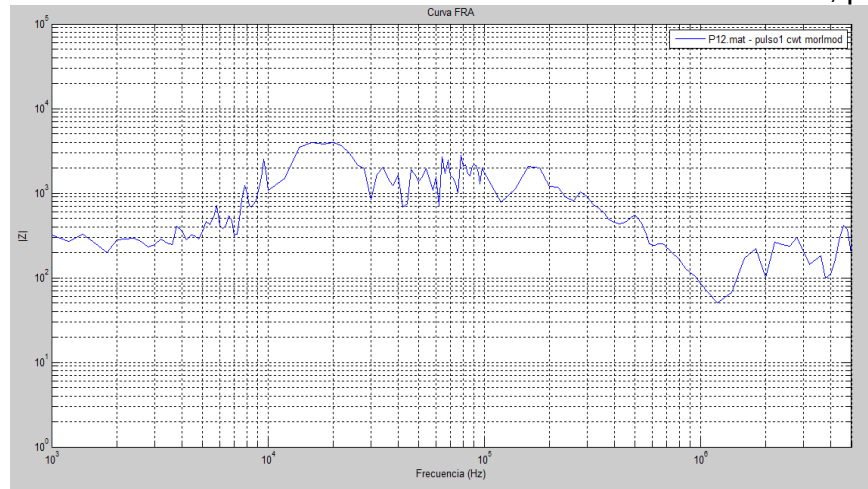
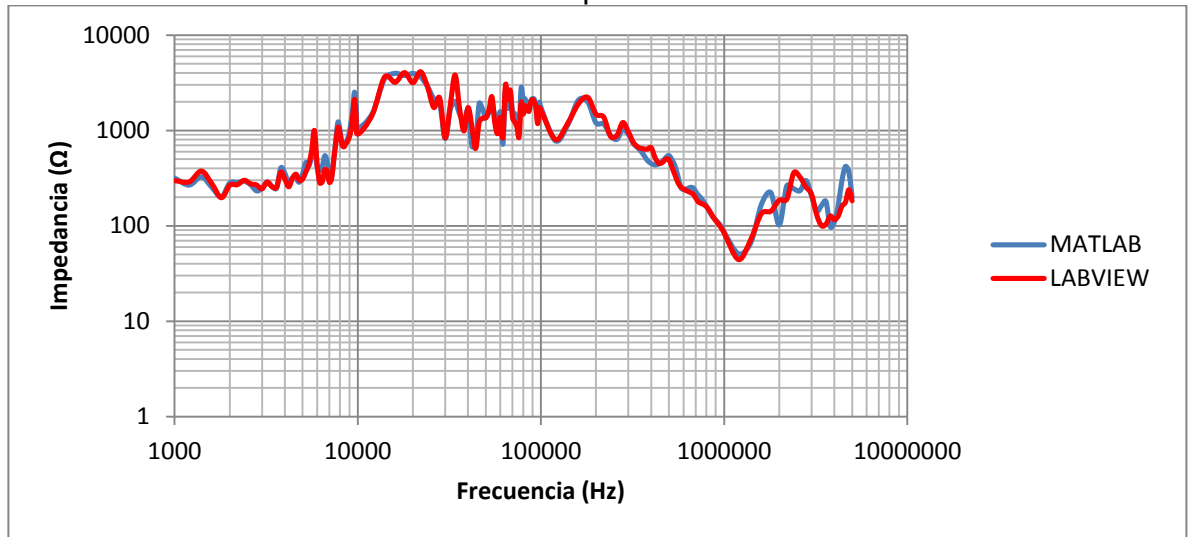


Figura 67. Curva FRA obtenida de manera off-line con Matlab, prueba 5



En la figura 68 se presenta la gráfica de comparación entre la curva obtenida con Labview, y la curva obtenida en Matlab, donde se observa que las dos respuestas presentan un alto nivel de similitud dentro del rango de frecuencias de 1 kHz a los 1000 kHz, este nivel de similitud es atribuido en gran parte a la carga conectada en el transformador, y a la ausencia de la señal de 120V. Además de notar que el tiempo de obtención de la curva para Labview fue de 8 segundos, mientras que para Matlab fue de 15 minutos.

Figura 68. Curvas FRA obtenidas con la aplicación de Matlab y Labview, prueba 5



#### 4.1.6 Prueba 6.

La sexta prueba, consistió en inyectar los pulsos en paralelo a una carga resistiva por el lado de alta tensión, manteniendo el lado de baja en vacío, y realizando las mediciones de tensión y corriente, por el lado de alta. En las figuras 69 y 70 se muestran los resultados obtenidos con Labview y Matlab de la curva FRA.

Figura 69. Curva FRA obtenida de manera dinámica con Labview, prueba 6

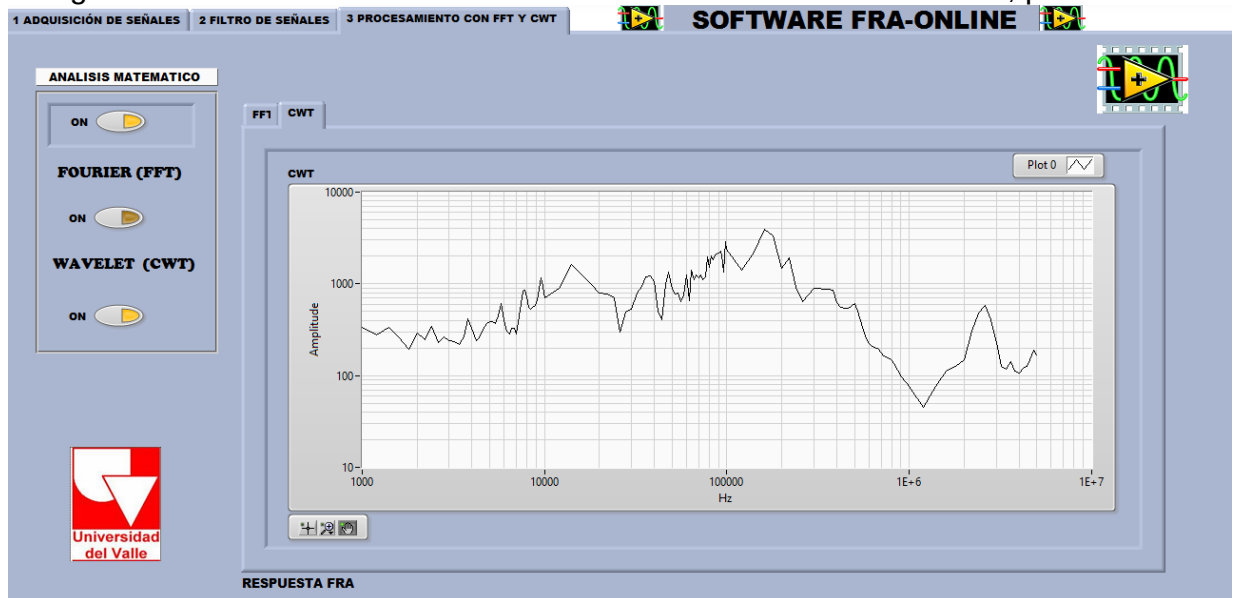
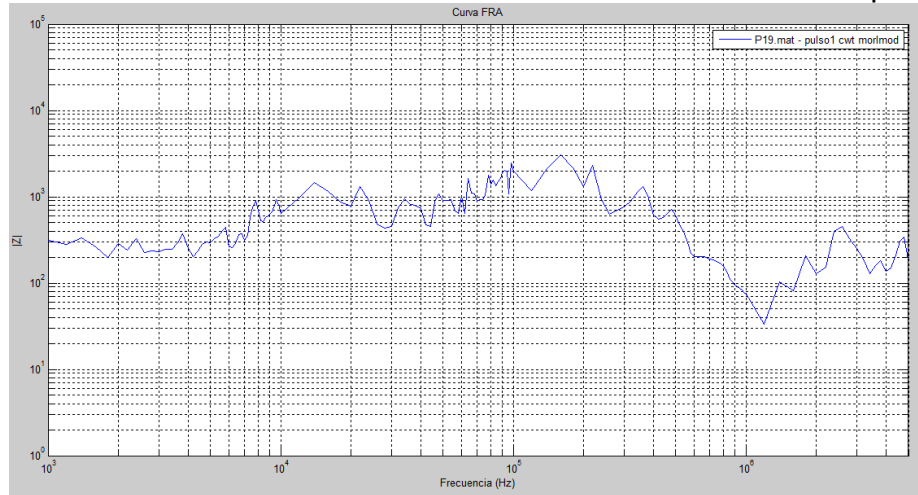
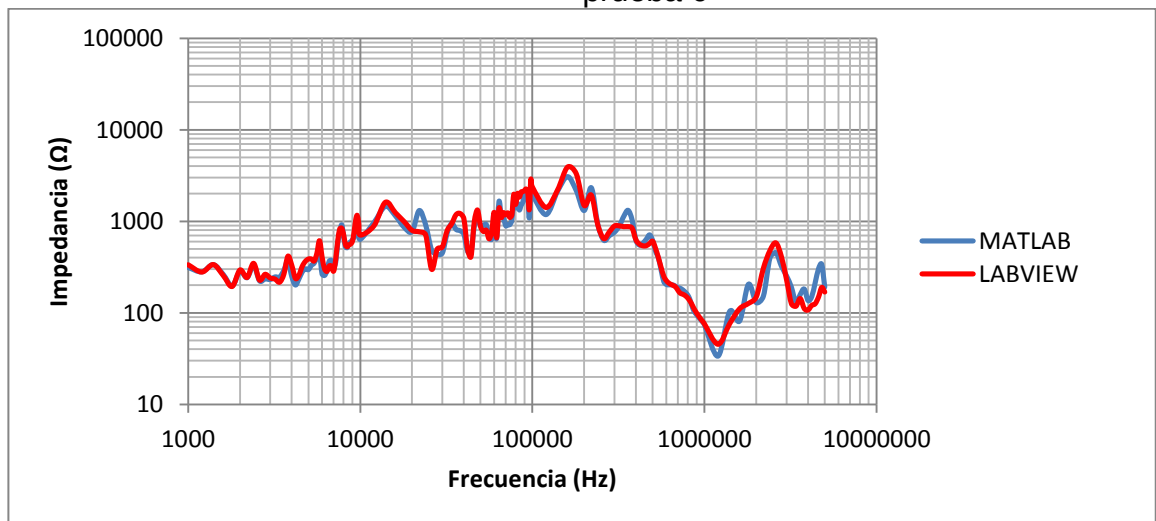


Figura 70. Curva FRA obtenida de manera off-line con Matlab, prueba 6



En la figura 71 se presenta la gráfica de comparación entre la curva obtenida con Labview, y la curva obtenida en Matlab, donde se observa que las dos respuestas presentan un alto nivel de similitud dentro del rango de frecuencias de los 1 kHz a los 1000 kHz, esta alta similitud es atribuida en gran parte a la carga conectada al transformador y a la ausencia de la señal de 120 V. además de observar que el tiempo de obtención de la curva para Labview fue de 8 segundos, mientras que para Matlab fue de 15 minutos.

Figura 71. Curvas FRA obtenidas con la aplicación de Matlab y Labview, prueba 6



## 5. CONCLUSIONES Y TRABAJOS FUTUROS

De acuerdo con el análisis realizado entre las curvas obtenidas dinámicamente y las obtenidas de manera off-line se puede apreciar los buenos resultados que se obtuvieron con la aplicación desarrollada, y el gran aporte de este trabajo para el método IFRA-online, a continuación se presentan las conclusiones del presente trabajo de grado y se mencionan los posibles trabajos futuros.

### 5.1 CONCLUSIONES

Los resultados obtenidos presentan una buena repetitividad, desde 10 kHz hasta 1 MHz entre las dos aplicaciones (Labview y Matlab); logrando obtener un monitoreo de manera dinámica de los devanados del transformador.

El tiempo para la obtención de la respuesta en frecuencia con la nueva aplicación software presenta una disminución notoria, con respecto al tiempo de obtención, de la aplicación de Matlab (alrededor de 8 segundos en Labview mientras que para la aplicación de Matlab, tarda alrededor de 14 minutos).

Se pudo observar con el trabajo realizado que en los escenarios del transformador bajo carga y vacío, la curva IFRA obtenida mediante el software Labview presentaba algunas pequeñas variaciones en comparación con las curvas obtenidas en Matlab; esto se debe a que al obtener la curva en tiempo real, se tiene una mayor precisión de la información dado la dinámica de las señales temporales.

Se aprecia la potencialidad que tiene la transformada wavelet para el estudio de los sistemas de potencia los cuales poseen un espectro de frecuencias único y que puede variar si se presenta una anomalía en sus componentes internos, reafirmando su ventaja sobre el método tradicional con Fourier y dando a conocer que es pieza clave en el estudio de señales transitorias.

El presente trabajo de grado, logró usar la transformada wavelet continua para obtener la curva IFRA en tiempo real usando el software Labview, complementado trabajos de grado anteriores donde solamente se había utilizado la transformada Fourier.

Se presenta a Labview como una gran alternativa para el monitoreo de aplicaciones complejas ya que no solo posee herramientas para el sistema de control e instrumentación, sino que además cuenta con herramientas avanzadas como la SPT (toolset de procesamiento de señales) que permite realizar aplicaciones de mayor complejidad y el Matlab Script Node que permite la

integración de Matlab con Labview exponiendo la versatilidad y potencialidad de este software frente a cualquier tipo de aplicación.

## 5.2 TRABAJOS FUTUROS

Para seguir con la línea de investigación y dar un paso hacia optimizar el método IFRA-online se propone:

- Ejecutar la aplicación desarrollada, para transformadores de potencia y de distribución variando sus parámetros internos para evaluar su efectividad en el diagnostico on-line.
- Realizar una serie de pruebas y posteriormente un análisis entre las curvas que se obtienen en tiempo real para verificar el diagnóstico en transformadores con la aplicación desarrollada.
- Implementar un sistema de telecomunicaciones para establecer un monitoreo remoto en un transformador con la aplicación desarrollada. Permitiendo así diagnosticar de manera óptima y eficiente los transformadores de distribución o de potencia cuando se encuentren en servicio desde una estación de control.

## 6. BIBLIOGRAFÍA

- [1] E. Gómez-Luna, "Obtencion de la respuesta en frecuencia de transformadores en servicio apartir de la medida de señales transitorias," universidad del valle, 2013.
- [2] M. Wang, a. J. Vandermaar, and K. . Srivastava, "Review of Condition Assessment of Power Transformers in Service," *IEEE Electr. Insul. Mag.*, vol. 18, no. 6, pp. 12–25, 2002.
- [3] P. Picher, C. Rajotte, and C. Tardif, "Experience with frequency response analysis (FRA) for the mechanical condition assessment of transformer windings," *IEEE Electr. Insul. Conf.*, pp. 220–224, 2013.
- [4] S. Ma and H. Ren, "Method for detecting system on power transformer winding deformation," *IEEE Int. Conf. Electr. Mach. Syst. ICEMS 2009*, 2009.
- [5] C. González, J. Pleite, V. Valdivia, and J. Sanz, "An overview of the on line application of frequency response analysis (FRA)," *IEEE Int. Symp. Ind. Electron.*, pp. 1294–1299, 2007.
- [6] Omicrom, "FRAnalyzer," 2011.
- [7] J. R. Secue, E. Mombello, and C. V. Cardoso, "Revisión del Analisis de Respuesta en Frecuencia (SFRA) para evaluación de desplazamientos y deformaciones de devanados en transformadores de potencia," *IEEE Lat. Am. Trans.*, vol. 5, no. 5, pp. 321–328, 2007.
- [8] E. Gomez-Luna, G. Aponte Mayor, C. Gonzalez-Garcia, and J. Pleite Guerra, "Current status and future trends in frequency-response analysis with a transformer in service," *IEEE Trans. Power Deliv.*, vol. 28, no. 2, pp. 1024–1031, 2013.
- [9] E. Gomez-Luna, G. Aponte Mayor, J. Pleite Guerra, D. F. Silva Salcedo, and D. Hinestroza Gutierrez, "Application of wavelet transform to obtain the frequency response of a transformer from transient signals-part 1: Theoretical analysis," *IEEE Trans. Power Deliv.*, vol. 28, no. 3, pp. 1709–1714, 2013.
- [10] R. Rampersad, S. Member, A. Singh, and S. Bahadoorsingh, "Platform for studying online frequency response analysis of power transformers," *IEEE Electr. Insul. Conf.*, no. June, pp. 7–10, 2015.

- [11] M. Mahvi, V. Behjat, and E. Rahimpour, "New statistical approach to interpret power transformer frequency response analysis: non-parametric statistical methods," *IET Sci. Meas. Technol.*, vol. 10, no. 4, pp. 364–369, 2016.
- [12] D. Cuadros, "Identificación de las variaciones en la curva FRA on-line ante cambios en los parametros internos y externos de un transformador piloto," 2016.
- [13] C. F. Guerrero, "Monitoreo del transformador en servicio, mediante la respuesta en frecuencia haciendo uso de la plataforma labview," p. 111, 2013.
- [14] J. S. Oyola, G. Gordillo, and C. A. Arredondo, "Desarrollo de Prototipo de Sistema de Monitoreo en Tiempo Real del Ni- vel de Agua en Ríos Usando Instrumentación Virtual," *RCF Rev. Colomb. Física*, vol. 40, no. 2, pp. 371–374, 2008.
- [15] M. a. Lemos, D. M. Brunini, G. Botura, M. a. Marques, and L. C. Rosa, "Virtual instrumentation: A practical approach to control and supervision process," *IEEE Int. Conf. Comput. Sci. Netw. Technol.*, vol. 4, pp. 2570–2573, 2011.
- [16] A. Singhal and A. Garg, "Online Parameter Determination and Performance Analysis of Three Phase Induction Motor using Virtual Instrumentation," *IEEE Int. Conf. Power Electron. Drivers Energy Syst.*, pp. 1–6, 2012.
- [17] A. Emami and T. Yazdi, "Designing and constructing transformer test system with aid of labview," *IEEE 2nd Int. Conf. Adapt. Sci. Technol.*, vol. 4, pp. 169–174, 2009.
- [18] NEPLAN smarter tools, "Software Neplan," 2015, 2015. [Online]. Available: <http://www.neplan.ch/neplanproduct/en-electricity/>.
- [19] M. Swathi Ramesh, S. Smrithi, P. Akhilraj Koti, P. Kanakasabapathy, a. N. Ravishankar, S. Pal, and P. Poornachandran, "Development of SCADA automated 33kV substation model as testing platform," *IEEE Int. Conf. Ind. Instrum. Control. ICIC*, no. Icic, pp. 1442–1447, 2015.
- [20] a L. Oliveira, O. C. B. De Araújo, a F. Brum, G. C. Jr, a P. De Moraes, and G. Marchesan, "An automatic fault diagnosis solution for electrical power systems," *IEEE Trans. Power Syst.*, pp. 1–5, 2011.

- [21] X. Ye and L. Zhou, "Using SCADA data fusion by swarm intelligence for wind turbine condition monitoring," *IEEE 4th Glob. Congr. Intell. Syst. GCIS*, pp. 210–215, 2013.
- [22] W. Ziomek, K. Vijayan, and D. G. Sanders, "Advanced Transformer Fleet Monitoring System," *IEEE Trans. Power Syst.*, pp. 0–4, 2014.
- [23] H. arturo Diaz domingo, torres juan, Bolivar jorge, Zerpa jhoan, "lenguaje de programacion delphi," 2012, 2012. [Online]. Available: <http://lenguajedeprogramaciondelphi.blogspot.com.co/>.
- [24] Y. E. Colak I, "Simulation of the parallel connection of electircal power stations," *IEEE Inf. Syst. Manag.*, pp. 41–46, 2000.
- [25] G. Ochoa, "Programa para la adquisición y filtrado de datos a partir del Monitor de Bioactividad LKB-2277," *Rev. CENIC Ciencias Biológicas*, vol. 39, no. 2, pp. 133–138, 2008.
- [26] L. Fan, Jinhui Jia, Songmin Wei and J. Wang, Zhihong Li, Xiuzhi Sheng, "Application of LVDT Sensor Data Acquisition System Based on PCI-1716," *IEEE Mod. Electron. Tec.*, pp. 548–552, 2011.
- [27] L. Li, C. Li, S. Sarkar, J. Zhang, S. Witham, Z. Zhang, L. Wang, N. Smith, M. Petukh, and E. Alexov, "DelPhi: a comprehensive suite for DelPhi software and associated resources," *BMC Biophys.*, vol. 5, p. 9, 2012.
- [28] S. Huss, J. Schmitz, D. Goltz, H.-P. Fischer, R. Büttner, and R. Weiskirchen, "Development and evaluation of an open source Delphi-based software for morphometric quantification of liver fibrosis.," *Fibrogenesis Tissue Repair*, vol. 3, no. 1, p. 10, 2010.
- [29] D. Bica, C. Moldovan, and M. Muji, "Power engineering education using NEPLAN software," *43rd Int. Univ. Power Eng. Conf.*, pp. 1–3, 2008.
- [30] S. Lazarou, R. Musca, M. Demitras, G. Fulli, and I. Colak, "Modeling of a Smart Grid System Using Real Time Data on Neplan," *IEEE 4th Int. Conf. Power Eng.*, no. May, pp. 13–17, 2013.
- [31] J. Cabrera, I. Medina, J. Hernández, and R. De Distribución, "Effects of PV Interconnection System to a Circuit of a Distribution Network," *IEEE Trans. Power Deliv.*, pp. 2–7, 2012.
- [32] S. Breganni, F. D. Urso, A. Giarratana, P. Gregoraci, M. Manzini, E. Pons, M. Repetto, C. Palestro, C. Duca, and H. A. L. S. Srl, "Online Optimizer for

- Distribution Networks on Neplan,” *IEEE Electr. Power Energy Conf.*, pp. 1–6, 2015.
- [33] PowerWorld corporation, “Software power World,” 2015, 2015. [Online]. Available: <http://www.powerworld.com/products/simulator/overview>.
  - [34] M. Elizondo, K. Karanjit, and W. Zhang, “Frequency Responsive Demand in U . S . Western Power System Model,” *IEEE Trans. Power Deliv.*, pp. 1–5, 2015.
  - [35] H. Kaur and J. S. Randhawa, “Optimal Power Flow Using Power World,” *IEEE Electr. Power Energy Conf.*, pp. 1–6, 2010.
  - [36] D. Zhang and S. Li, “Integrating PowerWorld and MatLab for Agent-Based Modeling and Simulation of Competitive Electric Power Markets,” *IEEE Electr. Insul. Mag.*, pp. 1–5, 2014.
  - [37] R. P. Klump and J. D. Weber, “Real-Time Data Retrieval and New Visualization Techniques for the Energy Industry PowerWorld Corporation Visualization techniques,” *35 th Hawaii Int. Conf. Syst. Sci.*, vol. 00, no. c, pp. 1–6, 2002.
  - [38] D. Zhang, S. Li, P. Zeng, and C. Zang, “Optimal microgrid control and power-flow study with different bidding policies by using powerworld simulator,” *IEEE Trans. Sustain. Energy*, vol. 5, no. 1, pp. 282–292, 2014.
  - [39] Departamento de Matematicas Universidad de los Andes, “software Matlab,” 2013.[Online].Available: [http://pentagono.uniandes.edu.co/tutorial/Matlab/tutorial\\_matlab.pdf](http://pentagono.uniandes.edu.co/tutorial/Matlab/tutorial_matlab.pdf).
  - [40] E. Dokur, A. Ozdemir, and S. Ilhan, “Matlab Gui Based Data Acquisition and Processing of Lightning Impulse Voltages,” *IEEE Electr. Power Int. Conf.*, no. 3, pp. 2–5, 2015.
  - [41] S. Li and D. Zhang, “Developing smart and real-time demand response mechanism for residential energy consumers,” *IEEE Electr. Power Energy Conf.*, pp. 1–5, 2014.
  - [42] M. N. Elya, M. Loqman, M. Aqilah, and S. Murniati, “Development of Simple Setup for Model Identification Using Matlab Data Acquisition,” *IEEE Int. Conf. Control Syst. Comput. Eng.*, pp. 52–57, 2013.

- [43] A. Panda, H. Wong, V. Kapila, and S. Lee, "Matlab Data Acquisition and Control Toolbox for Basic Stamp Microcontrollers," *IEEE Conf. Decis. Control*, pp. 3918–3925, 2006.
- [44] T. J. Cavicchi, "Experimentation and analysis: SigLab/MATLAB data acquisition experiments for signals and systems," *IEEE Trans. Educ.*, vol. 48, no. 3, pp. 540–550, 2005.
- [45] P. Ray, a. K. Maitra, and A. Basuray, "On-line partial discharge monitoring using MATLAB interface with DSO," *2013 IEEE 1st Int. Conf. Cond. Assess. Tech. Electr. Syst. IEEE CATCON 2013 - Proc.*, pp. 383–385, 2013.
- [46] N. Sultana and Y. Kamatham, "Mitigation of Noise and Interference in ECG Signals with Savitzky-Golay Least Squares Polynomials and Discrete Wavelet Transform," *Electr. Comput. Commun. Technol.*, pp. 1–5, 2015.
- [47] S. S. Murthy, R. K. Mittal, A. Dwivedi, G. Pavitra, and S. Choudhary, "Online performance monitoring and testing of electrical equipment using Virtual Instrumentation," *7th Int. Conf. Power Electron. Drive Syst.*, pp. 1608–1612, 2007.
- [48] S. H. Laskar and M. Muhammad, "Power Quality Monitoring by Virtual Instrumentation using LabVIEW," *46 th Int. Univ. Power Eng. Conf.*, pp. 1–6, 2011.
- [49] N. Khera and S. Jain, "Development of Lab VIEW based Electrical Parameter Monitoring System for Single Phase Supply," *Int. Conf. Commun. Control Intell. Syst.*, pp. 1–4, 2015.
- [50] Q. F. Niu, W. Li, and Y. P. Shi, "Realization of a power transformer on-line monitoring and diagnosis system based on DGA and PNN," *Int. Conf. Comput. Mechatronics, Control Electron. Eng. C.*, vol. 4, pp. 210–213, 2010.
- [51] R. Sriphuek and S. Chotigo, "Low-cost frequency response analyzer for transformer diagnosis," *IEEE Int. Conf. Cond. Monit. Diagnosis, C.*, no. September, pp. 437–441, 2012.
- [52] Q. Chengdun, "Design of virtual instrument based on MATLAB and LABView," *Fifth Conf. Meas. Technol. Mechatronics Autom.*, pp. 466–469, 2013.
- [53] A. Abdulhady and R. Bicker, "A Simulation of Non-stationary Signal Analysis Using Wavelet Transform Based on LabVIEW and Matlab," *IEEE Eur. Model. Symp.*, pp. 138–144, 2014.

- [54] E. Salami, I. B. Tijani, and A. U. Jibia, "Development of real-time software interface for multicomponent transient signal analysis using Labview and Matlab," *IEEE Int. Conf. Mechatronics*, no. May, pp. 1–5, 2011.
- [55] H. Singhal, S. Arora, and A. Kaur, "Hands on Educational Platform for Control System Models using MATLAB and LabVIEW," *IEEE Int. Conf. Control Appl.*, no. 1, pp. 489–494, 2013.
- [56] D. Vuljaj, Z. Hanić, and M. Vražić, "LabVIEW Application for Visualisation of Synchronous Machine Real-Time Capability Diagram with Variable Operational Limits," *Int. Conf. Electr. Drives Power Electron.*, pp. 21–23, 2015.
- [57] M. Upadhye, P. Borole, and A. Sharma, "Real-Time Wireless Vibration Monitoring System Using LabVIEW," *IEEE Int. Conf. Ind. Instrum. Control*, no. Icic, pp. 925–928, 2015.
- [58] L. Hadjidemetriou, G. Nicolaou, D. Stavrou, and E. Kyriakides, "Low-Cost Real-Time Monitoring of a Laboratory Scale Power System," *IEEE Mediterr. Electrotech. Conf.*, no. April, pp. 1–6, 2016.
- [59] G. Holguín, S. Pérez, and Á. Orozco, *Curso Básico de LabVIEW 6i*. 2002.
- [60] G. Bucks and W. Oakes, "Work in Progress - Impact of Graphical Programming Environments on Learning and Understanding Programming Concepts," *IEEE Front. Educ. Conf.*, pp. 23–24, 2008.
- [61] N. I. Corporation, "LabVIEW Tools for Time-Frequency, Time-Series, and Wavelet Analysis," *Natl. Instruments*.
- [62] X. Rios, "Diseño e implementación de un sistema de compresión de imagenes usando dispositivos reprogramables," Universidad del Valle, 2011.
- [63] N. Natalia and O. D. Marcela, "EL USO DE LA TRANSFORMADA WAVELET DISCRETA EN LA RECONSTRUCCIÓN DE SEÑALES SENOSOIDALES," no. 38, pp. 381–386, 2008.

## ANEXOS

### Manual de usuario aplicación IFRA on-line

El manual se divide en dos partes; la primera parte es la instalación óptima de la aplicación, la segunda parte es una guía de usuario del manejo de esta. Como requisito a tener en cuenta antes de utilizar la aplicación; es que se debe contar en el computador con el software de Matlab versión 6.5 o superior, y que dicha aplicación solo funciona con tarjetas de adquisición de datos que tengan como controladores el NI-DAQmx.

### INSTALACIÓN

1. Se descomprime el archivo de la imagen 1.



Imagen 1. Archivo Software IFRA-online

2. Se realiza la instalación mediante el siguiente procedimiento:

Ejecutar el setup siguiendo la dirección: software IFRA-online >> setup >> Volume >> setup.exe, ver imagen 2.

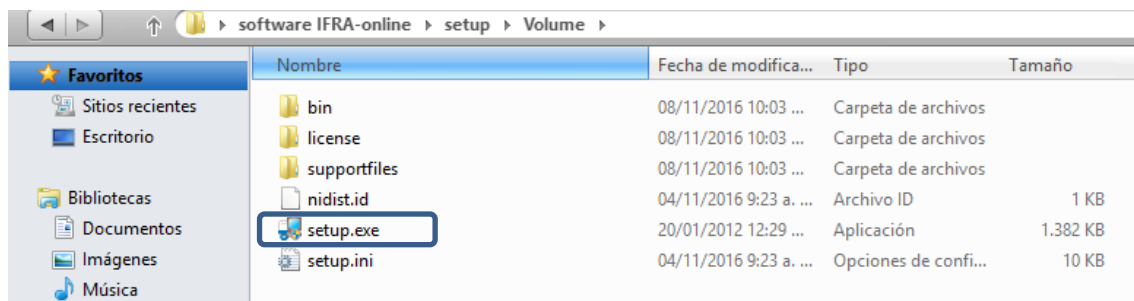
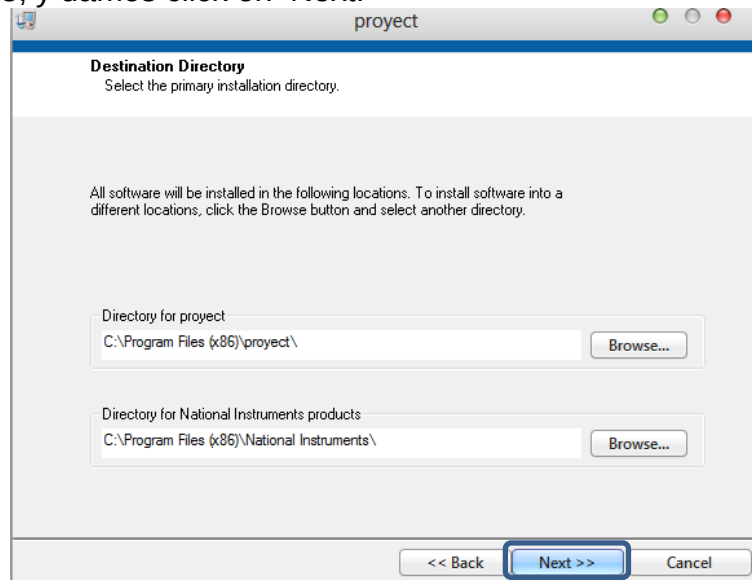


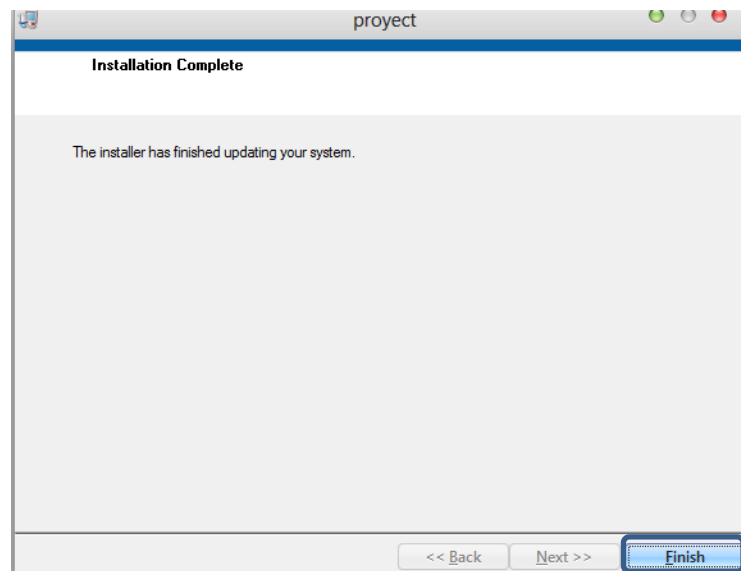
Imagen 2. Instalador del software IFRA-online

Luego de ello, deberá aparecer el siguiente cuadro de dialogo tal como se indica en la imagen 3, y damos click en Next.



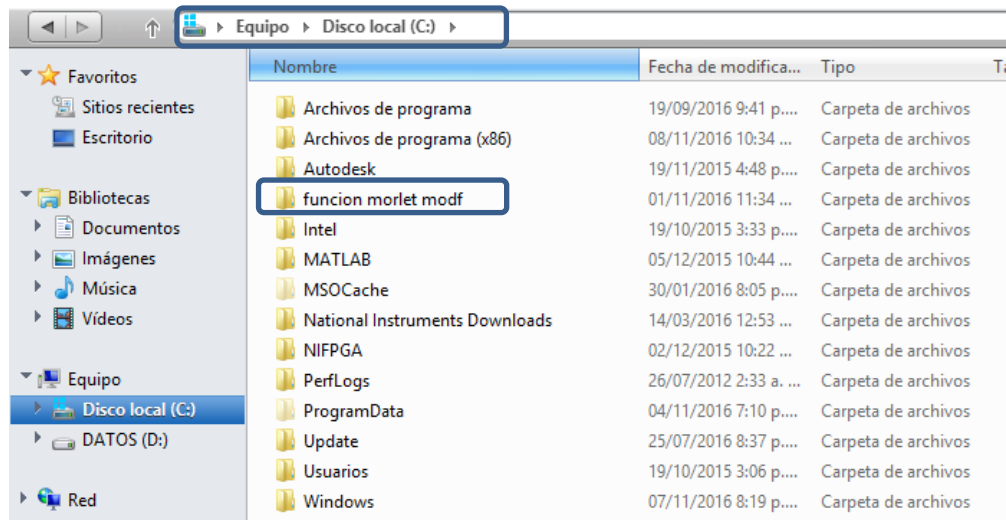
**Imagen 3. Cuadro de dialogo de instalación**

Esperaremos a que se instale y se procede a dar click en Finish tal como se indica en la imagen 4.



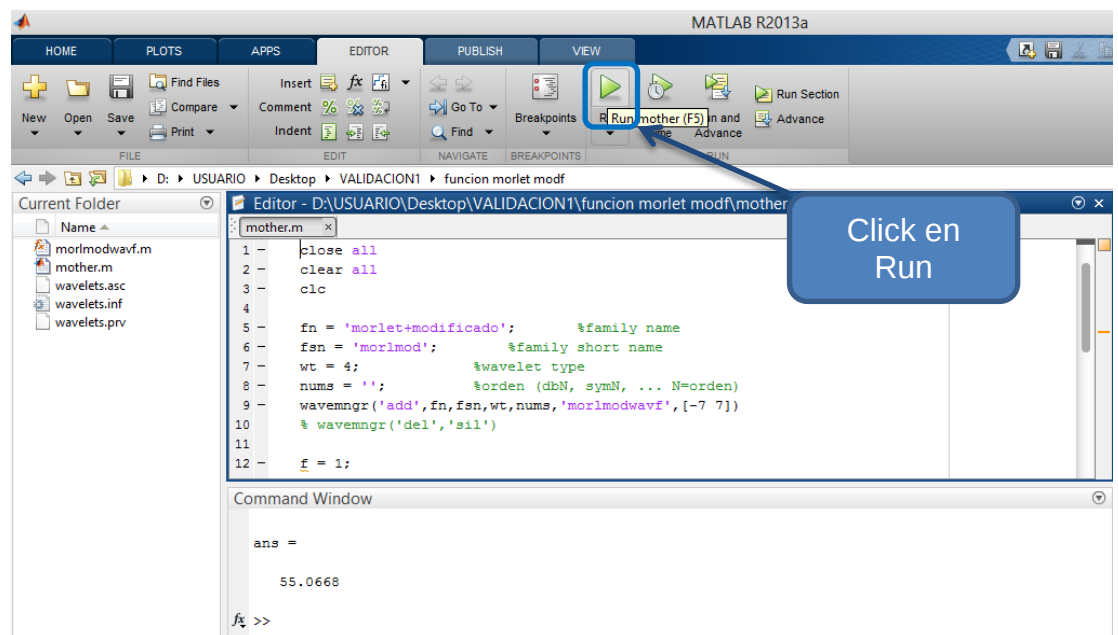
**Imagen 4. Finalización de la instalación**

Seguido de ello, copiar la carpeta llamada “funcion morlet modf”, que se encuentra en la carpeta software IFRA-online, y pegarla en disco local C, tal como se indica en la imagen 5.



**Imagen 5. Dirección para pegar la carpeta función morlet modf**

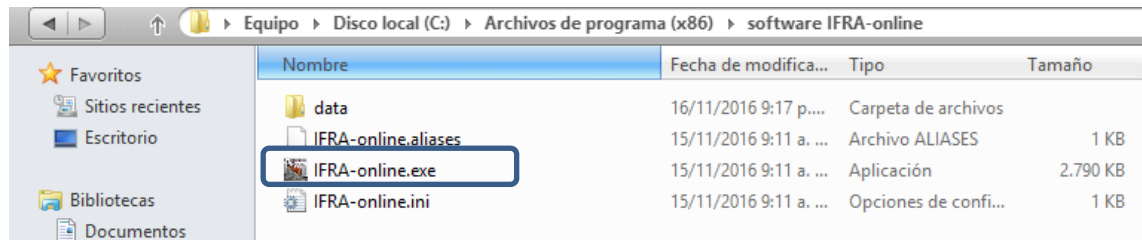
- Posteriormente abrir el archivo “mother.m”, que se encuentra en la carpeta función morlet modf, este archivo debe compilarse tal como se indica la imagen 6.



**Imagen 6. Compilación del archivo mother.m**

Este procedimiento solo se realiza la primera vez que se hace la instalación de la aplicación software.

Después de haber realizado este paso se procede a cerrar el archivo “mother.m”; y finalmente para esta primera parte, se ejecuta la aplicación que queda ubicada en la siguiente dirección: Disco local C >> archivos de programa (x86) >>software IFRA-online >> IFRA-online.exe; ver imagen 6.



**Imagen 7. Ejecución de la aplicación**

## **GUIA PARA OBTENER LA RESPUESTA EN FRECUENCIA**

Después de haber realizado la primer parte, se procede a describir la manera de cómo se debe configurar y utilizar dicha aplicación, inicialmente esta comprende 3 etapas, la primera, es la etapa de adquisición de datos, donde se hace referencia a los controles para obtener las señales sensadas, la segunda; es la etapa del filtrado y eliminación del ruido, donde se realiza el filtro y acondicionamiento de los pulsos sobre la señal de 60 Hz; la tercer y última etapa es el procesamiento matemático, donde se visualiza la curva de respuesta en frecuencia; en esta se visualizan dos curvas, una obtenida con la transformada de Fourier y otra curva obtenida mediante la transformada wavelet.

A continuación se presenta la manera secuencial como se obtiene la curva de respuesta en frecuencia mediante la aplicación desarrollada, sin embargo antes de correr la aplicación se debe de tener en cuenta que número de canal físico se utilizó con los equipos de medición.

### **Etapas 1. Adquisición de datos**

Permite la opción continua de N iteraciones para adquirir las señales sensadas, en la imagen 8, se presenta la interfaz de configuración, mediante la cual se controla la adquisición de muestras, se presentan los controles de configuración y seguido de ello, las recomendaciones que se deben de tener en cuenta al momento de parametrizarlos. Para acceder a ellos dar click en la pestaña de adquisición de señales y dar click en el boton “off-on”.

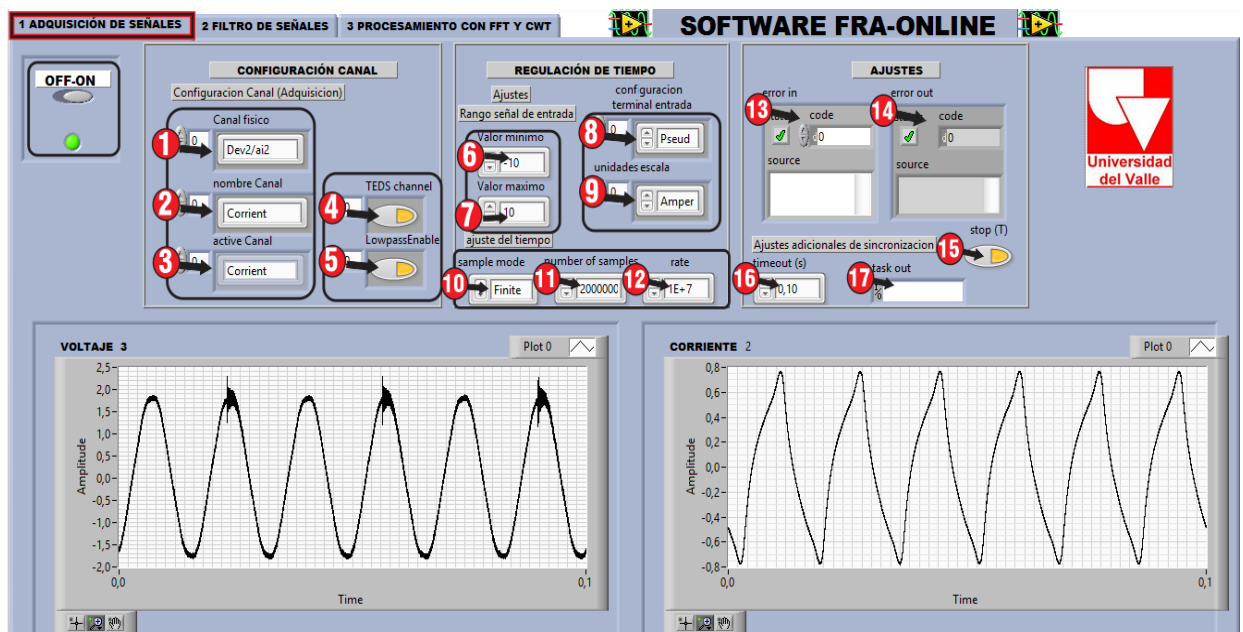


Imagen 8. Secuencia de pasos para la adquisición de señales con la aplicación software.

1. **Canal físico:** Nombre del canal físico mediante el cual se conectara la señal que se piensa sensar. La tarjeta cuenta con 8 canales disponibles, los cuales van enumerados desde el canal 0, al canal 7. Es decir si en la interfaz se selecciona el canal 0 para la adquisición de la señal de voltaje, físicamente en la entrada 0 de la tarjeta 6115 debe estar conectado dicha señal.
2. **Nombre del canal:** La señal que se encuentre en etapa de adquisición, es necesario que coincida con el nombre del canal en la interfaz, esto con el fin de realizar un adecuado procesamiento de esta, en este caso el canal 0 se encargara de las señales referentes a los pulsos de corriente y el canal 1 se encargara de los de voltaje.
3. **Activación del canal:** Nombre del canal que se piensa activar para el proceso de adquisición
4. **TEDS channel:** Activa o desactiva el canal
5. **Low Pass Enable:** Activa o desactiva el filtro de un canal
6. **Valor Mínimo:** Valor mínimo del rango para las señales que se van adquirir, para la aplicación FRA, este valor es de -10

7. **Valor Máximo:** Valor máximo del rango para las señales que se van adquirir, para la aplicación FRA, este valor es de 10
8. **Configuración del terminal de entrada:** Existen cinco tipos de entradas o de medidas para adquirir las señales, estas son Default, differential, NRSE, Pseudodifferential, y RSE, la función de estos tipos de medida son mencionadas de manera más detallada en manual de ayuda de labview, sin embargo para la aplicación se recomienda utilizar la "Pseudodiferencial", ya que permite proporcionar un rechazo de señales en modo común o un rechazo al ruido eléctrico presente en las señales.
9. **Unidades de escala:** Parámetro que configura las unidades de los canales de entrada, para la aplicación FRA, se definió la unidad de ampere para el canal 0 y Volts para el canal 1.
10. **Modo de Muestreo: "sample mode"** Parámetro que determina si el número de muestras a adquirir o a generar se dan en forma continua o se genera en un número finito, en la etapa de adquisicion se recomienda colocar que las muestras se obtienen en un numero finito ("finite").
11. **Number of samples:** Parámetro que especifica el número de muestras para adquirir o generar en cada canal creado; si el número de muestras que se dan son continuas, el NI DAQmx utiliza este valor para determinar el tamaño del buffer, en esta ventana se recomienda colocarlos en 2E6, Debido a la cantida de muestras dadas en los pulsos.
12. **Velocidad de muestreo "rate":** Valor máximo de frecuencia de muestreo admisible para la tarjeta de adquisición 6115, de este valor depende el número de muestras y la frecuencia para el procesamiento de las señales. Debido a las especificaciones que trae consigo la tarjeta se recomienda colocarlo en 10E6.
13. **Error de entrada** Presenta una breve descripción de un error generado en el proceso antes que se ejecute el VI, o la función.
14. **Error de salida** Presenta una breve descripción de un error generado en el proceso después que se ejecute el VI, o la función.
15. **Stop** Detiene el proceso de adquisición de datos.

**16. Time out** Tiempo de espera para la adquisición de datos. si se establece el tiempo de espera en -1 el VI esperara indefinidamente, si se establece el tiempo de espera en 0, el VI intentara leer las muestras solicitadas de manera inmediata; sumado a esto; el timeout puede generar un error en el VI en caso de no ser capaz de realizar dicha tarea por lo que es recomendable configurarlo con valores cercanos a cero (0.1, 0.2, 0.3).

**17. Task out** En caso de que se genera algún error este Indicador permite visualizar la tarea que no se ejecuto correctamente en el proceso de adquisición de datos.

## Etapa 2. Filtro de señales

Permite la opción de filtrar las señales sensadas, es decir excluye los pulsos inyectados de la señal de 60 HZ y posteriormente realiza la eliminación del ruido eléctrico.

En la imagen 2, se presenta la interfaz de configuración, mediante la cual se realiza el filtro de los pulsos; para utilizarla dar click en la pestaña 2 “filtro de señales”. Seguido de ello, dar click en los botones de filtro de 60Hz Y filtro de Ruido; tal como se indica en la imagen,

A continuación se realiza una breve descripción de los botones de esta etapa.

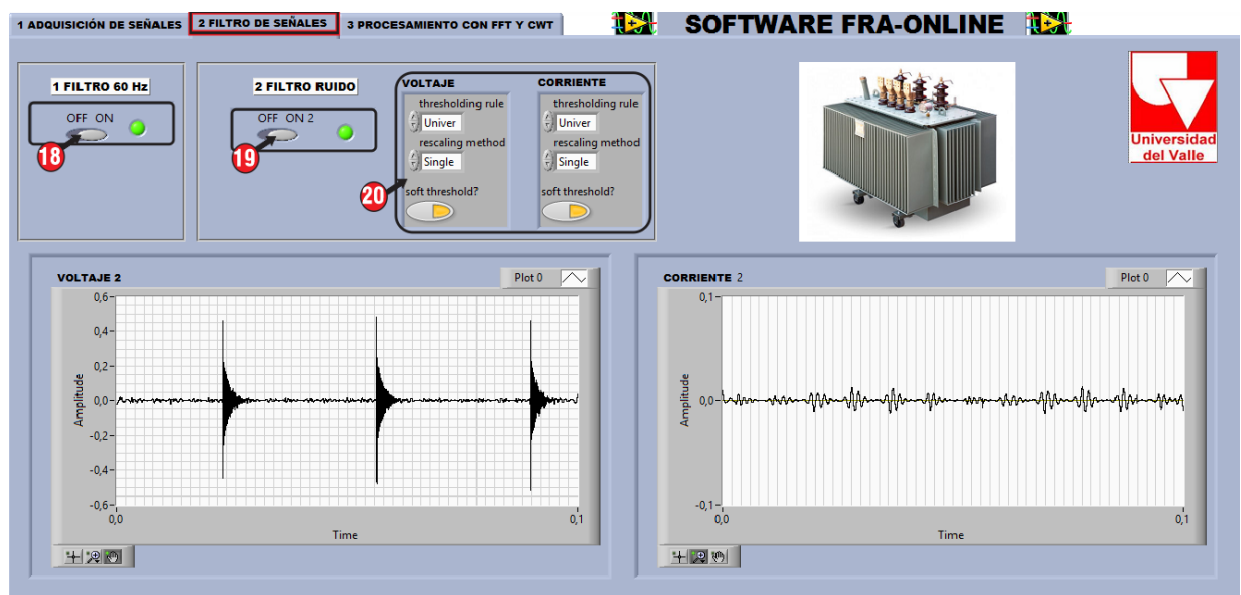


Imagen 9. Secuencia de pasos para el filtro de señales con la aplicación software.

**18. Filtro 60 Hz** Etapa para activar el filtro que eliminara toda señal que se encuentre dentro de la frecuencia de los 60 Hz. Para activarlo dar click, en el botón de “filtro de 60Hz” tal como se indica en la imagen 2.

**19. Filtro denoising** Activa la etapa de eliminación del ruido eléctrico, para activarlo dar click, en el botón de “filtro ruido” tal como se indica en la imagen 2.

**20. Ajustes:** Se recomienda colocar los ajustes del Thresholding rule en “universal” y el Rescaling method en “single”.

### Etapa 3. Procesamiento Matemático

Permite la opción de realizar el procesamiento matemático para obtener la respuesta en frecuencia del transformador, esta etapa tiene la versatilidad de obtener la respuesta en frecuencia con la transformada de Fourier, o con la transformada wavelet.

En la imagen 3, se presenta la interfaz de configuración, mediante la cual se realiza la obtención de la curva FRA; para utilizarla dar click en la pestaña 3 “procesamiento con FFT y CWT”. A continuación se presenta la descripción de los botones de esta etapa.

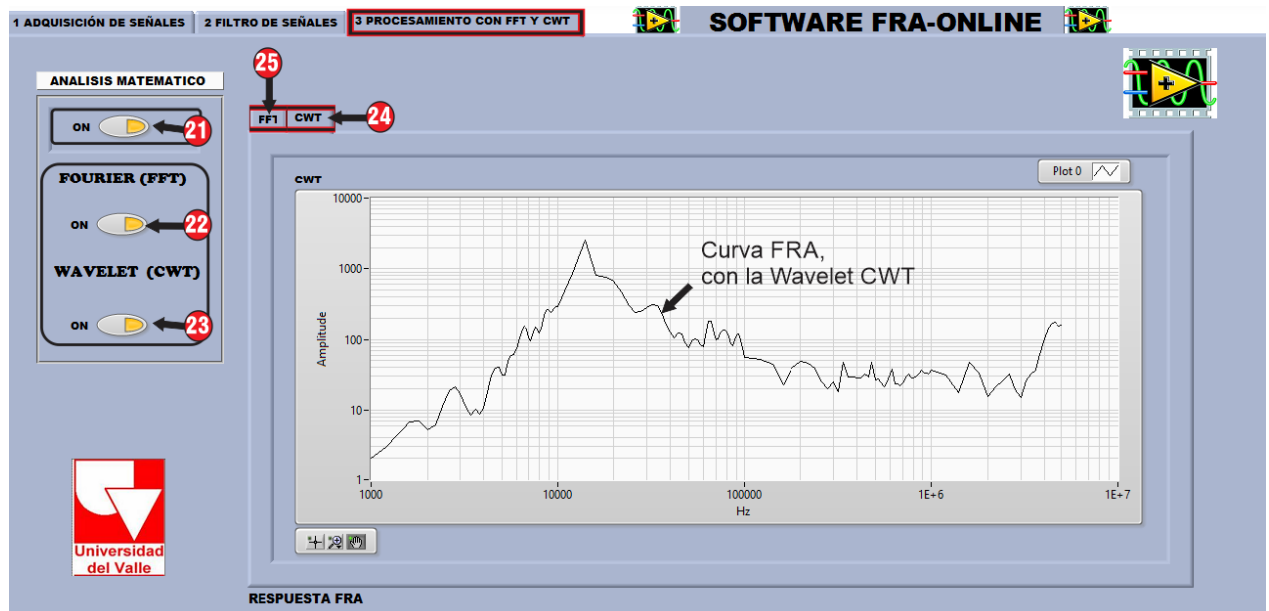


Imagen 10 pasos para visualizar la curva con la aplicación desarrollada

- 21. **Botón on:** activa el procesamiento matemático para utilizar ya sea la transformada de Fourier o la transformada wavelet.
- 22. **Botón on (Fourier):** activa el proceso de obtención de la curva FRA con la transformada de Fourier.
- 23. **Botón on (Wavelet):** activa el proceso de obtención de la curva FRA con la transformada wavelet.
- 24. **Pestaña CWT:** permite visualizar la respuesta en frecuencia con la transformada wavelet.
- 25. **Pestaña FFT:** permite visualizar la respuesta en frecuencia con la transformada de Fourier.