



Universidad Nacional Autónoma De México

Facultad de Ingeniería

TESIS:

**ADQUISICION Y PROCESAMIENTO DE
INFORMACION CON BASE EN LA NORMA
IEEE C37.118-2011 DE SINCRFASORES
PARA SISTEMAS ELECTRICOS DE
POTENCIA.**

Que para Obtener El Titulo De: Ingeniero Eléctrico Electrónico

Alumno:

Baez López Enrique

Tutor:

Ing. Juan José López Marquez

Mexico DF 2015

I

*Dedicado a
mi familia*

Agradecimientos

A mis padres y hermanos.
A mis maestros y guías.

Índice general

Agradecimientos	II
Indice de Figuras	IV
Indice de Cuadros	VII
1. Introducción	1
1.1. Medición Fasorial	1
1.1.1. Fasor	1
1.1.2. Parámetros de la medición fasorial	3
1.1.3. Instrumentos de medición fasorial	4
1.1.4. Retos de la medición fasorial	4
1.2. En resumen	4
2. Adquisición de datos	5
2.1. Justificación	5
2.2. TCP/IP	5
2.2.1. Significado y aplicación	5
2.2.2. Modelo de capas	6
2.2.3. El modelo TCP/IP	6
2.3. Dirección IP	8
2.3.1. Qué es la dirección IP	8
2.3.2. Estructura	9
2.3.3. IP estático	10
2.3.4. IP dinámico	10
2.4. Detección de errores CRC	10
3. Decodificación de datos	12
3.1. C37.118.2-2011	12
3.1.1. Configuración general	12
3.1.2. Comandos	15
3.1.3. Frame de configuración 2	16
3.1.4. Frame de datos	24
4. Implementación	30
4.1. Breve descripción del entorno de Labview	30
4.1.1. Panel Frontal	31
4.1.2. Diagrama De Bloques	31

4.2.	Programación gráfica aplicada a adquisición y publicación de información proveniente de los equipos de medición.	31
4.3.	Adquisición.vi	32
4.3.1.	Consulta de los datos de los PMUs	35
4.3.2.	Procesamiento de datos generales de los PMU	37
4.3.3.	Procesamiento de los datos particulares de un PMU.	38
4.3.4.	Preparación de los frame de comandos	39
4.3.5.	Comunicación con el PMU y cese del flujo de datos.	41
4.3.6.	Peticion, recepcion y publicacion del frame de configuracion	43
4.3.7.	Peticion del frame de datos	44
4.3.8.	Recepción, y publicación de frame de datos.	44
4.4.	Visualizador.vi	46
4.4.1.	Consulta de los datos de las variables compartidas	50
4.4.2.	Procesamiento de datos de las variables compartidas en inicialización de títulos de graficas	50
4.4.3.	Lectura de variables compartidas	52
4.4.4.	Decodificación de frame de configuración.	52
4.4.5.	Procesamiento de frame de datos	63
4.4.6.	Decodificación de frame de datos	67
4.4.7.	Tiempo y sincronización	89
4.4.8.	Procesamiento de Waveform	92
4.4.9.	Graficado	95
4.5.	Guardado.vi	99
4.5.1.	Inicio del reporte	102
4.5.2.	Inserción de datos	104
4.5.3.	Formato, guardado y cierre del reporte	106
4.5.4.	Preparación de encabezado y datos	107
4.6.	Alarma.vi	113
4.6.1.	Búfer	117
4.6.2.	Condición de alerta	120
4.6.3.	Numero de alertas	129
4.6.4.	Centrado del evento	131
4.6.5.	Guardado y envío	133
5.	Conclusiones	138

Índice de figuras

1.1. Representación gráfica de un fasor	2
4.1. Adquisición Algoritmo Básico	33
4.2. Adquisición Panel Frontal	34
4.3. Adquisición Diagrama de Bloques	35
4.4. Control de dirección	36
4.5. adquisición 1	37
4.6. adquisición 2	38
4.7. adquisición 3	40
4.8. adquisición 4	42
4.9. adquisición 5	43
4.10. adquisición 6	44
4.11. adquisición 7	45
4.12. Visualizador Algoritmo Básico	47
4.13. Visualizador Panel frontal	48
4.14. Visualizador Diagrama de Bloques	49
4.15. Visualizador 1	50
4.16. Visualizador 2	51
4.17. Visualizador 3	52
4.18. Visualizador 4	53
4.19. Decodificador del frame de configuración 2 (Panel frontal)	54
4.20. Decodificador del frame de configuración 2 (Diagrama de bloques)	55
4.21. Visualizador 5	56
4.22. Visualizador 6	58
4.23. Dec. Conf. Particular (Panel frontal)	59
4.24. Dec. Conf. Particular (Diagrama de bloques)	60
4.25. Visualizador 7	62
4.26. Visualizador 8	63
4.27. Visualizador 8 (Panel frontal)	64
4.28. Visualizador 8 (Diagrama de bloques)	65
4.29. Decodificador del frame de datos (Panel Frontal)	67
4.30. Decodificador del frame de datos (Diagrama de bloques)	68
4.31. Visualizador 9	70
4.32. Visualizador 10	71
4.33. Visualizador 10 (Panel frontal)	73
4.34. Visualizador 10 (Diagrama de bloques)	74
4.35. Visualizador 11	75
4.36. Visualizador 12	77

4.37. Visualizador 12.1	79
4.38. Visualizador 12.2	80
4.39. Visualizador 12.3	81
4.40. Visualizador 12.4	82
4.41. Visualizador 13	83
4.42. Visualizador 14	85
4.43. Visualizador 15	85
4.44. Visualizador 16	87
4.45. Visualizador 17	88
4.46. Visualizador 18	90
4.47. Visualizador 19	91
4.48. Visualizador 20	92
4.49. Visualizador 21 (Panel frontal)	93
4.50. Visualizador 21 (Diagrama de bloques)	93
4.51. Visualizador 22	94
4.52. Visualizador 23	96
4.53. Diferencia angular (Panel frontal)	97
4.54. Diferencia angular (Diagrama de bloques)	97
4.55. Caso 0: el resultado pasa directamente	98
4.56. Caso 1: se resta 360 al resultado	98
4.57. Caso 2: se suma 360 al resultado	98
4.58. Guardado Algoritmo Básico	99
4.59. Guardado (Panel frontal)	100
4.60. Guardado (Diagrama de bloques)	101
4.61. Sección de guardado correspondiente al visualizador	102
4.62. Guardado 1	103
4.63. Guardado 1.1	104
4.64. Guardado 2	105
4.65. Guardado 3	106
4.66. Guardado 3.1	107
4.67. Guardado 4	108
4.68. Guardado 5 (Panel frontal)	109
4.69. Guardado 5(Diagrama de bloques)	110
4.70. Identificadores (Parametro de entrada)	111
4.71. Parámetros de entrada (fijos en la programación)	112
4.72. Encabezados (de las columnas de la hoja de excel)	112
4.73. Alarma Algoritmo Básico	114
4.74. Alarma (Panel frontal)	115
4.75. Alarma (Diagrama de bloques)	116
4.76. Sección de Alarma correspondiente al visualizador	117
4.77. Bufer	118
4.78. Búfer (Panel frontal)	119
4.79. Búfer (Diagrama de bloques)	119
4.80. comparador	121
4.81. Alarma 1	122
4.82. Alarma 2	123
4.83. Ejemplo de perturbación en la tensión	124
4.84. Ejemplo de perturbación en la frecuencia	125
4.85. Alarma 3	126
4.86. Alarma 4 (Panel frontal)	127

4.87. Alarma 4 (Diagrama de bloques)	128
4.88. Perturbación de una subestación	129
4.89. Perturbación del sistema	130
4.90. Alarma 5	131
4.91. Alarma 6	132
4.92. Alarma 7	133
4.93. Ejemplo de correo recibido	135
4.94. Ejemplo de Archivo recibido (Nombre del archivo)	135
4.95. Ejemplo de Archivo recibido (hoja de voltajes)	136
4.96. Ejemplo de Archivo recibido (hoja de frecuencias)	137

Índice de cuadros

3.1. Equivalencia de bits entre sistema hexadecimal y binario	13
3.2. Configuración general	14
3.3. tabla de comandos	15
3.4. Ejemplo de frame de comando	15
3.5. Frame de configuración 2	18
3.6. Ejemplo de frame de configuracion 2	24
3.7. Frame de fatos	25
3.8. Ejemplo de frame de datos	27
3.9. valores decodificados	29
4.1. Formato General Adquisición	89
4.2. Formato Especifico de Visualizador.vi	90
4.3. Formato con potencias y tensión	95

Capítulo 1

Introducción

1.1. Medición Fasorial

1.1.1. Fasor

Un Fasor es la representación gráfica de un número complejo que se utiliza para representar un fenómeno periódico;

Matemáticamente el Fasor Y se puede expresar como:

$$Y = A[\cos(\omega t + \phi) + \sin(\omega t + \phi)j] \quad (1.1)$$

Donde:

A es una magnitud que gira sobre el origen del plano complejo $(0,0j)$, en sentido anti horario.

ϕ es angulo de desplazamiento de la magnitud A con respecto a otra magnitud (si no existe otra magnitud de referencia su valor es 0).

ω es la velocidad angular a la que gira A y se calcula:

$$\omega = 2\pi f \quad (1.2)$$

Donde:

f es la frecuencia del sistema.

Gráficamente se puede representar mediante la figura

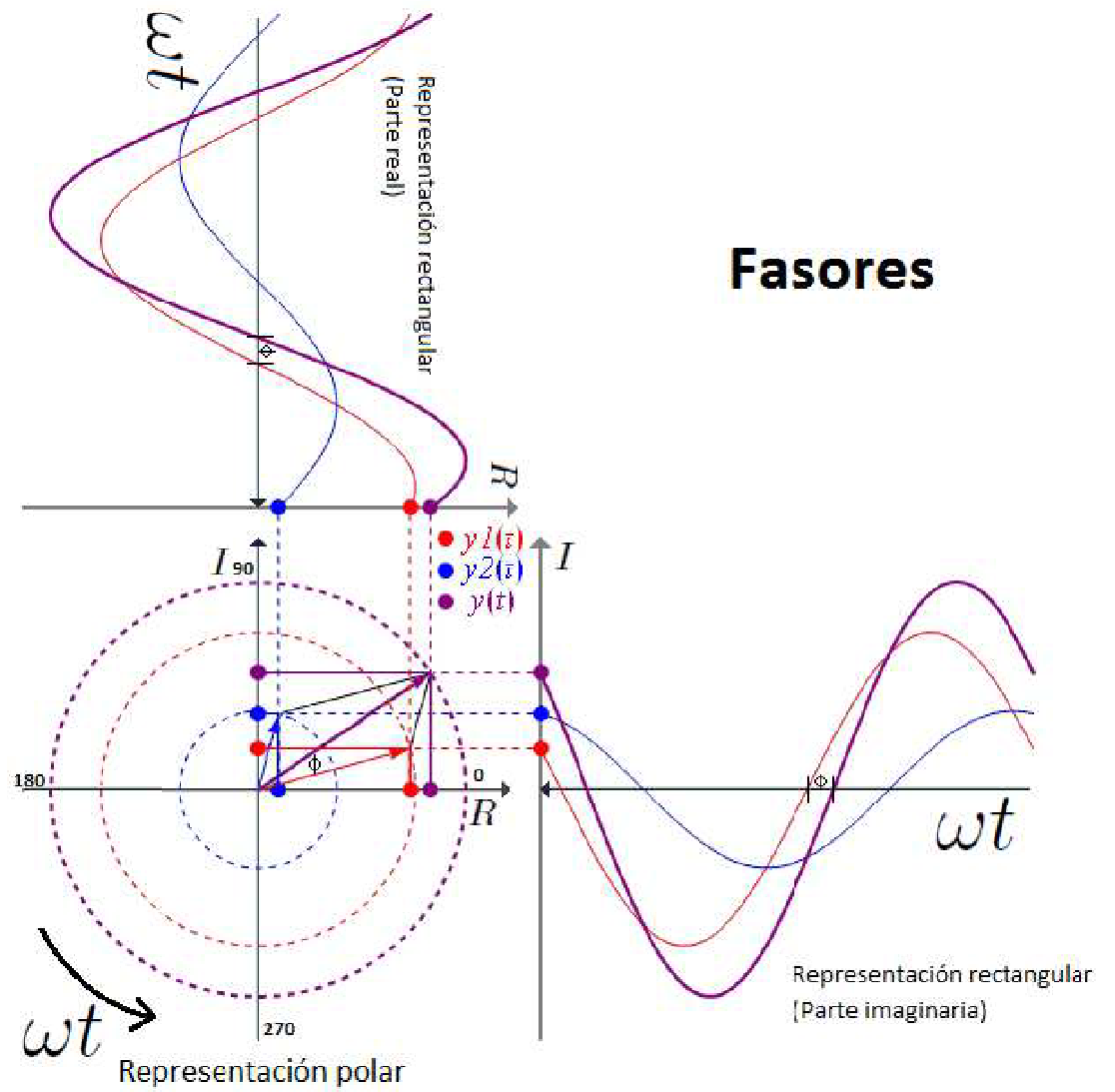


Figura 1.1: Representación gráfica de un fasor

Para un instante dado, se puede determinar el valor del ángulo instantáneo del fasor θ mediante la ecuación:

$$\theta = \omega t + \phi \quad (1.3)$$

Sustituyendo la ecuación 1.3 en 1.1 se tiene que, la ecuación que describe el valor instantáneo del fasor Y es:

$$Y = A[\cos(\theta) + \sin(\theta)j] \quad (1.4)$$

1.1.2. Parámetros de la medición fasorial

Tensión y Corriente

En el sector eléctrico los fenómenos de principal interés son la tensión y corriente del sistema. Estos fenómenos son de tipo periodico razón por la cual se emplean los fasores.

Para conocer el comportamiento general de los fasores, basta con medir los parámetros de frecuencia (f), magnitud (A) y angulo instantáneo (θ) y repetir la medición a un intervalo constante.

Calculo de potencias y diferencia angular

Otros fenómenos de gran interés son la diferencia angular entre la tensión y la corriente; la diferencia entre el ángulo de dos puntos del sistema; y las potencias activa y reactiva de una linea, bus o subestación. Todos estos parámetros pueden ser medidos de forma directa, pero para efectos prácticos se calculan posteriormente mediante las siguientes ecuaciones.

Diferencia angular:

Sea la diferencia del angulo (ϕ) entre la tensión y la corriente, o la diferencia entre la tensión en dos puntos distintos del sistema (ej. dos subestaciones), este parámetro se calcula restando el angulo instantáneo del fasor en cuestión (θ_1) al angulo instantáneo del fasor de referencia (θ_0) como se muestra en la ecuación:

$$\phi_1 = \theta_1 - \theta_0 \quad (1.5)$$

la razón es que la frecuencia (f) de dos puntos del sistema son muy aproximadas, y con base en la ecuación (1.2) se tiene: $\omega_1 \approx \omega_0$ por lo tanto:

$$\theta_1 - \theta_0 = (\omega_1 t + \phi_1) - (\omega_0 t + 0) \approx \omega t - \omega t + \phi_1 = \phi_1$$

Potencia aparente:

Siendo V el fasor de la tensión e I el fasor de la corriente se puede calcular la potencia aparente mediante la ecuación:

$$S = VI^* \quad (1.6)$$

Donde I^* es el conjugado de la corriente

Potencia activa:

Se define como la parte real de la potencia aparente

$$P = \text{Re}\{S\} \quad (1.7)$$

Potencia reactiva:

Se define como la parte imaginaria de la potencia aparente

$$Q = \text{Im}\{S\} \quad (1.8)$$

1.1.3. Instrumentos de medición fasorial

Para realizar las mediciones de los parámetros de los fasores que describen los fenómenos eléctricos, en el sector eléctrico se han desarrollado instrumentos digitales que realizan múltiples tomas de muestras por segundo y las codifican para su posterior procesamiento. A estos instrumentos se les denomina unidades de medición fasorial o PMU por sus siglas en inglés (Phasor Measurement Unit).

Tipos de PMU

En la Gerencia Regional de Transmisión Central (GRTC) de la Comisión Federal de Electricidad (CFE), se tiene registros de PMUs de las marcas ABB, SEL, SIEMENS, ROCHESTER y algunas unidades de medición implementadas como un PMU sin ser tal.

1.1.4. Retos de la medición fasorial

Pese a ser instrumentos de gran utilidad, los PMUs se encuentran con algunos retos por superar tanto para su implementación como su estudio tal como es:

- La distribución tan alejada de los equipos, los cuales están instalados en subestaciones y líneas situadas a varios kilómetros de distancia entre si y hacia los equipos que requieren la información.
- La limitada cantidad de usuarios que pueden consultar la información generada por un PMU (algunos PMU solo pueden soportar 2 usuarios).
- La diversidad de empresas que fabrican los PMUs y el software específico para cada equipo.

1.2. En resumen

La finalidad de este trabajo es elaborar un software que permita adquirir y procesar información proveniente de los PMU superando los retos intrínsecos de los mismos a través del protocolo C37.118.2-2011.

Capítulo 2

Adquisición de datos

2.1. Justificación

En la actualidad existen diversos medios, modelos y protocolos de comunicación, tanto de forma alámbrica (cableada punto a punto), como de forma inalámbrica (Wireles, ondas de radio, micro ondas, bluetooth); pero resalta la comunicación en red (e Internet) usando el modelo TCP/IP por sus ventajas en la transmisión de datos de gran volumen y su uso generalizado en equipos que requieren comunicarse de forma remota. En el caso de las unidades de medición fasorial (PMU), estas ya tienen integrado un sistema que permite publicar y recibir datos de esta forma.

2.2. TCP/IP

1

2.2.1. Significado y aplicación

TCP/IP es un conjunto de protocolos. La sigla TCP/IP significa “Protocolo de Control de Transmisión/Protocolo de Internet” y se pronuncia “T-C-P-I-P”. En algunos aspectos, TCP/IP representa todas las reglas de comunicación para Internet y se basa en la noción de dirección IP, es decir, en la idea de brindar una dirección IP a cada equipo de la red para poder enrutar paquetes de datos. Está diseñado para cumplir con una cierta cantidad de criterios, entre ellos:

- Dividir mensajes en paquetes;
- Usar un sistema de direcciones;
- Enrutar datos por la red;
- Detectar errores en las transmisiones de datos.

El conocimiento del conjunto de protocolos TCP/IP no es esencial para un simple usuario, de la misma manera que un espectador no necesita saber cómo

¹KIOSKEA.NET TCP/IP (Bibliografía [1])

funciona su red audiovisual o de televisión. Sin embargo, para las personas que desean administrar o brindar soporte técnico a una red TCP/IP, su conocimiento es fundamental.

2.2.2. Modelo de capas

Para poder aplicar el modelo TCP/IP en cualquier equipo, es decir, independientemente del sistema operativo, el sistema de protocolos TCP/IP se ha dividido en diversos módulos. Cada uno de éstos realiza una tarea específica. Además, estos módulos realizan sus tareas uno después del otro en un orden específico, es decir que existe un sistema estratificado. Ésta es la razón por la cual se habla de modelo de capas.

El término capa se utiliza para reflejar el hecho de que los datos que viajan por la red atraviesan distintos niveles de protocolos. Por lo tanto, cada capa procesa sucesivamente los datos (paquetes de información) que circulan por la red, les agrega un elemento de información (llamado encabezado) y los envía a la capa siguiente.

La importancia de un sistema de capas

El objetivo de un sistema en capas es dividir el problema en diferentes partes (las capas), de acuerdo con su nivel de abstracción. Cada capa del modelo se comunica con un nivel adyacente (superior o inferior). Por lo tanto, cada capa utiliza los servicios de las capas inferiores y se los proporciona a la capa superior.

Encapsulación de datos

Durante una transmisión, los datos cruzan cada una de las capas en el nivel del equipo remitente. En cada capa, se le agrega información al paquete de datos. Esto se llama encabezado, es decir, una recopilación de información que garantiza la transmisión. En el nivel del equipo receptor, cuando se atraviesa cada capa, el encabezado se lee y después se elimina. Entonces, cuando se recibe, el mensaje se encuentra en su estado original.

En cada nivel, el paquete de datos cambia su aspecto porque se le agrega un encabezado. Por lo tanto, las designaciones cambian según las capas:

- El paquete de datos se denomina mensaje en el nivel de la capa de aplicación.
- El mensaje después se encapsula en forma de segmento en la capa de transporte.
- Una vez que se encapsula el segmento en la capa de Internet, toma el nombre de datagrama.
- Finalmente, se habla de trama en el nivel de capa de acceso a la red.

2.2.3. El modelo TCP/IP

El modelo TCP/IP, utiliza el enfoque modular (utiliza módulos o capas), pero sólo contiene cuatro:

Capa de acceso a la red: especifica la forma en la que los datos deben enrutarse, sea cual sea el tipo de red utilizado;

Capa de Internet: es responsable de proporcionar el paquete de datos (datagrama);

Capa de transporte: brinda los datos de enrutamiento, junto con los mecanismos que permiten conocer el estado de la transmisión;

Capa de aplicación: incorpora aplicaciones de red estándar (Telnet, SMTP, FTP, etc.).

A continuación se describen mas detalladamente:

■ Capa de acceso a la red

La capa de acceso a la red es la primera capa de la pila TCP/IP. Ofrece la capacidad de acceder a cualquier red física, es decir, brinda los recursos que se deben implementar para transmitir datos a través de la red.

Por lo tanto, la capa de acceso a la red contiene especificaciones relacionadas con la transmisión de datos por una red física, cuando es una red de área local (Red en anillo, Ethernet, FDDI), conectada mediante línea telefónica u otro tipo de conexión a una red. Trata los siguientes conceptos:

- Enrutamiento de datos por la conexión.
- Coordinación de la transmisión de datos (sincronización).
- Formato de datos.
- Conversión de señal (análoga/digital).
- Detección de errores a su llegada.

Afortunadamente, todas estas especificaciones son invisibles al ojo del usuario, ya que en realidad es el sistema operativo el que realiza estas tareas, mientras los drivers de hardware permiten la conexión a la red (por ejemplo, el driver de la tarjeta de red).

■ Capa de Internet

La capa de Internet es la capa “más importante” (si bien todas son importantes a su manera), ya que es la que define los datagramas y administra las nociones de direcciones IP. Permite el enrutamiento de datagramas (paquetes de datos) a equipos remotos junto con la administración de su división y ensamblaje cuando se reciben.

La capa de Internet contiene 5 protocolos:

- El protocolo IP.
- El protocolo ARP.
- El protocolo ICMP.
- El protocolo RARP.
- El protocolo IGMP.

Los primeros tres protocolos son los más importantes para esta capa.

- Capa de transporte

Los protocolos de las capas anteriores permiten enviar información de un equipo a otro. La capa de transporte permite que las aplicaciones que se ejecutan en equipos remotos puedan comunicarse. El problema es identificar estas aplicaciones. De hecho, según el equipo y su sistema operativo, la aplicación puede ser un programa, una tarea, un proceso, etc. Además, el nombre de la aplicación puede variar de sistema en sistema.

Es por ello que se ha implementado un sistema de numeración para poder asociar un tipo de aplicación con un tipo de datos. Estos identificadores se denominan puertos.

La capa de transporte contiene dos protocolos que permiten que dos aplicaciones puedan intercambiar datos independientemente del tipo de red (es decir, independientemente de las capas inferiores). Estos dos protocolos son los siguientes:

- TCP, un protocolo orientado a conexión que brinda detección de errores;
- UDP, un protocolo no orientado a conexión en el que la detección de errores es obsoleta.

- Capa de aplicación

La capa de aplicación se encuentra en la parte superior de las capas del protocolo TCP/IP. Contiene las aplicaciones de red que permiten la comunicación mediante las capas inferiores. Por lo tanto, el software en esta capa se comunica mediante uno o dos protocolos de la capa inferior (la capa de transporte), es decir, TCP o UDP.

Existen diferentes tipos de aplicaciones para esta capa, pero la mayoría son servicios de red o aplicaciones brindadas al usuario para proporcionar la interfaz con el sistema operativo. Se pueden clasificar según los servicios que brindan:

- Servicios de administración de archivos e impresión (transferencia).
- Servicios de conexión a la red.
- Servicios de conexión remota.

2.3. Dirección IP

2

2.3.1. Qué es la dirección IP

Todo equipo conectado a Internet (o a cualquier red) posee una identificación única, llamada dirección IP (en inglés, Internet Protocol), compuesta por cuatro combinaciones de números (p.ej. 187.25.14.190).

²INFORMÁTICA HOY (Bibliografía [2])

Estos números, llamados octetos, pueden formar más de cuatro billones de direcciones diferentes. Cada uno de los cuatro octetos tiene una finalidad específica. Los dos primeros grupos se refieren generalmente al país y tipo de red (clases). Este número es un identificador único en el mundo: en conjunto con la hora y la fecha, puede ser utilizado, por ejemplo, por las autoridades, para saber el lugar de origen de una conexión.

El concepto de IP va ligado al TCP. Un protocolo de red es como un idioma, si dos personas están conversando en idiomas diferentes ninguna entenderá lo que la otra quiere decir. Con los equipos ocurre una cosa similar, dos equipos que están conectados físicamente por una red deben “hablar” el mismo idioma para que uno entienda los requisitos del otro. El protocolo TCP estandariza el cambio de información entre las computadoras y hace posible la comunicación entre ellas. Es el protocolo más conocido actualmente pues es el protocolo estándar de Internet.

El protocolo TCP contiene las bases para la comunicación de computadoras dentro de una red, pero así como nosotros cuando queremos hablar con una persona tenemos que encontrarla e identificarla, las computadoras de una red también tienen que ser localizadas e identificadas. En este punto entra la dirección IP. La dirección IP identifica a una computadora en una determinada red. A través de la dirección IP sabemos en que red está la computadora y cual es la computadora. Es decir verificado a través de un número único para aquella computadora en aquella red específica.

2.3.2. Estructura

La dirección IP consiste en un número de 32 bits que en la práctica vemos siempre segmentado en cuatro grupos de 8 bits cada uno (xxx.xxx.xxx.xxx). Cada segmento de 8 bits varía de 0-255 y están separados por un punto.

Esta división del número IP en segmentos posibilita la clasificación de las direcciones IPs en 5 clases: A, B, C, D e E. Cada clase de dirección permite un cierto número de redes y de computadoras dentro de estas redes.

Direcciones IP Clase A

En las redes de clase “A” los primeros 8 bits de la dirección son usados para identificar la red, mientras los otros tres segmentos de 8 bits cada uno son usados para identificar a las computadoras.

Una dirección IP de clase A permite la existencia de 126 redes y 16 777 214 computadoras por red. Esto pasa porque para las redes de clase A fueron reservados por la IANA (Internet Assigned Numbers Authority) los IDs de “0” hasta “126”.

Direcciones IP Clase B

En las redes de clase B los primeros dos segmentos de la dirección son usados para identificar la red y los últimos dos segmentos identifican las computadoras dentro de estas redes.

Una dirección IP de clase B permite la existencia de 16 384 redes y 65 534 computadoras por red. El ID de estas redes comienza con “128.0” y va hasta “191.255”.

Direcciones IP Clase C

Redes de clase C utilizan los tres primeros segmentos de dirección como identificador de red y sólo el último segmento para identificar la computadora.

Una dirección IP de clase C permite la existencia de 2 097 152 redes y 254 computadoras por red. El ID de este tipo de red comienza en “192.0.1” y termina en “223.255.255”.

Direcciones IP Clase D

En las redes de clase D todos los segmentos son utilizados para identificar una red y sus direcciones van de “224.0.0.0” hasta “239.255.255.255” y son reservados para los llamados multicast. Las redes de clase Y, así como las de clase D, utilizan todos los segmentos como identificadores de red y sus direcciones se inician en “240.0.0.0” y van hasta “255.255.255.255”. La clase Y es reservada por la IANA para uso futuro.

2.3.3. IP estático

El IP estático (o fijo) es un número IP asignado permanentemente a una computadora, o sea, su dirección IP no cambia, excepto si dicha acción se fuera realizada manualmente.

2.3.4. IP dinámico

El IP dinámico, es un número que es asignado a una computadora cuando esta se conecta a la red, pero que cambia cada vez que se establece la conexión.

La rotación de direcciones IPs (IP dinámicos) funciona de la siguiente forma: un determinado proveedor de acceso a Internet (Ej. Arnet), posee X números IPs para usar. Cada vez que una máquina se conecta a internet, el proveedor le asigna una dirección IP aleatoria, dentro de una cantidad de direcciones IPs disponibles. El proceso más utilizado para esta distribución de IPs dinámicos es el Dynamic Host Configuration Protocol (DHCP).

2.4. Detección de errores CRC

El método de control de errores por redundancia cíclica (CRC), permite la verificación de todos los bits del mensaje mediante la utilización de un algoritmo matemático.

Básicamente:

En el equipo transmisor:

- Se divide el mensaje que se quiere enviar, entre un polinomio conocido en ambos extremos del canal de comunicaciones llamado generador.
- Como resultado de esta división, se obtendrán dos polinomios: el cociente y el resto o CHK (abreviatura de check).
- El CHK se transmite junto con la secuencia de bits sobre la que se quiere efectuar el control de errores.

En el receptor:

- Se reciben el mensaje a controlar junto con el CHK de la división.
- El mensaje a controlar se divide entre el polinomio generador en forma idéntica que en el transmisor, y se obtiene un nuevo resto (CHK).

Finalmente se compara el CHK obtenido por los cálculos realizados en el receptor, con el CHK recibido de los cálculos efectuados en el transmisor. Si no han habido errores, ambos restos deben ser iguales.

Este método se basa en la detección de errores por medio de operaciones aritméticas con polinomios que utilizan la técnica conocida como modulo 2.

Capítulo 3

Decodificación de datos

Una vez establecida la comunicación con los PMU, el siguiente paso es conocer la organización y formato de la información, y decodificarla o “traducirla” del lenguaje maquina a los valores fasoriales de los parámetros eléctricos.

3.1. C37.118.2-2011

¹

Se trata de la Norma IEEE para Transferencia datos de sincrofasores para Sistemas de Potencia, en ella se especifica la configuración de la información y los estándares utilizados para poder transmitir, recibir y decodificar los datos adquiridos por sistemas de medición eléctrica de potencia, tales como los PMU.

3.1.1. Configuración general

La información se transmite en formato digital en forma de mensajes llamados FRAMES.

Formato digital

Se refiere a que la información es recibida mediante secuencias ordenadas de 0 y 1 lógicos.

En el formato digital encontramos los siguientes términos como estructuras básicas:

- Bit. Es el acrónimo Binary digit (“dígito binario”). Puede representar el valor de 0 o 1.
- Byte. Es la agrupación de 8 bits.

Existen dos maneras básicas de representar un byte:

- De forma binaria con 8 dígitos. ejemplo: 1111 1000
- De forma hexadecimal con 2 dígitos. ejemplo: F8

¹C37.118.2-2011 IEEE Standard for Synchrophasor Data Transfer for Power Systems (Bibliografía [4])

En ambos ejemplos anteriores se representa el mismo byte.

HEX	BIN	HEX	BIN
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

Cuadro 3.1: Equivalencia de bits entre sistema hexadecimal y binario

Frames

Se trata de mensajes no continuos con secuencias de valores binarios. Estos valores de acuerdo con su posición contienen información o configuración de los fasores.

En la Norma IEE C37.118.2-2011 se especifica, que para las comunicaciones desde y hacia un equipo de medición existen cuatro tipos distintos de frames:

- De datos (Data Frame)
- De configuración (Configuration frame)
- De comandos (Command frame)
- De encabezado (Header frame)

Todos estos frames contienen un orden particular en sus bits, pues cada uno tiene un propósito específico, pero en todos ellos se sigue una estructura básica la cual se expresa en el cuadro 3.2:

Posición	Campo	Bytes	Descripción y comentarios
1	SYNC	2	El primer byte es el valor constante "AA" El segundo byte nos indica el tipo de frame y la versión Este se divide de la siguiente manera: Bit 7: Reservado (valor constante "0") Bits 6-4: bin (hex) 000 (0): Data Frame (datos) 010 (1): Configuration Frame 1 011 (3): Configuration Frame 2 101 (5): Configuration Frame 3 100 (4): Command Frame (comando) Bits 3-0
continúa tab. 3.2 ...			

Posición	Campo	Bytes	Descripción y comentarios
			0001(1): versión 1 (solo código definido en IEEE Std C37.118-2005) 0002(2): versión 2 (con código agregado en IEEE Std C37.118.2-2011)
2	FRAMESIZE	2	-Número total de bytes en el frame incluyendo el CHK. Numero entero de 16 bits sin signo; rango: [0,65535]
3	IDCODE	2	-Código de identificación (definido por el usuario) Número entero de 16 bits sin signo; rango: [1,65534], (0 y 65535 están reservados)
4	SOC	4	-Estampa de tiempo: Numero de segundos transcurridos desde la media noche del 1° de enero de 1970 (base UNIX) Numero entero de 32 bits sin signo.
5	FRACSEC	4	-Fracción de segundo y calidad de tiempo: El primer Byte nos indica la calidad de tiempo. Este se divide le la siguiente manera: Bits 7-4: Definidos en 6.2.2 de IEEE Std C37.118.2-2011 3-0: bin (hex) 0000 (0): Operación normal. sincronizado con el UTC 0001 (1): Desviación de 10-9 s del UTC 0010 (2): Desviación de 10-8 s del UTC 0011 (3): Desviación de 10-7 s del UTC 0100 (4): Desviación de 10-6 s del UTC 0101 (5): Desviación de 10-5 s del UTC 0110 (6): Desviación de 10-4 s del UTC 0111 (7): Desviación de 10-3 s del UTC 1000 (8): Desviación de 10-2 s del UTC 1001 (9): Desviación de 10-1 s del UTC 1010 (A): Desviación de 1 s del UTC 1011 (B): Desviación de 10 s del UTC 1111 (F): Fallo del reloj, tiempo no confiable Los 3 bytes restantes indican la fracción de segundo. Numero entero de 24 bits.
-	-	-	Configuración particular de cada tipo de frame
Ultima	CHK	2	“Resto” de la verificación CRC.

Cuadro 3.2: Configuración general

Calculo del tiempo

El valor del tiempo se divide en dos partes, la parte entera (expresada en el SOC) y la parte fraccional, (expresada en los últimos 3 bytes del FRACSEC).

Para calcular el tiempo se tiene la ecuación 3.1:

$$tiempo = SOC + \frac{(FRACSEC)}{(TIME_BASE)} \quad (3.1)$$

Donde TIME_BASE es un valor que se encuentra en los frame de configuración que se describirán adelante (cap 3.1.3).

3.1.2. Comandos

Se trata de instrucciones que sirven para pedir información de la configuración del frame o iniciar o detener el flujo de datos.

Los posibles comandos que se pueden enviar a un PMU están expresados en el cuadro 3.3

Tabla de comandos		
Binario	Hex	Descripción
0000 0000 0000 0001	00 01	Detener transmisión de datos
0000 0000 0000 0010	00 02	Iniciar transmisión de datos
0000 0000 0000 0011	00 03	Enviar Header frame
0000 0000 0000 0100	00 04	Enviar frame de configuración 1
0000 0000 0000 0101	00 05	Enviar frame de configuración 2
0000 0000 0000 0110	00 06	Enviar frame de configuración 3
0000 0000 0000 1000	00 08	Frame extendido

Cuadro 3.3: tabla de comandos

Ejemplo de configuración general aplicada a frame de comandos

Se pretende enviar el comando de “Enviar frame de configuración 2” a un PMU, para lo cual se emplea la siguiente tabla auxiliar:

Indice	Hex		Binario		Descripción	Valor
	Byte +0	Byte +1				
0	AA	41	10101010	01000001	Tipo de instrucción:	Comando
					Versión	V1
2	00	12	00000000	00010010	Bytes de lectura:	18
4	00	01	00000000	00000001	Código de identificación:	1
6	00	00	00000000	00000000	Segundos	0
8	00	00	00000000	00000000		
10	00	00	00000000	00000000	Calidad del tiempo	00
12	00	00	00000000	00000000	Fracción de segundo:	0
14	00	05	00000000	00000101	Comando	5
16	E5	E8	11100101	11101000	CHK	E5E8

Cuadro 3.4: Ejemplo de frame de comando

Dado que el mensaje a enviar es una instrucción y el tiempo no es relevante para el equipo que la recibe, los bits que describen tiempo se deja en sus valores predeterminados.

Se pasan los valores de la tabla a una trama:

AA41 0012 0001 0000 0000 0000 0000 0005 E5E8

Este mensaje ya puede ser enviado mediante un programa de comunicaciones o algún lenguaje que acepte el protocolo TCP/IP al PMU , especificando su dirección IP, el puerto y el código de identificación.

3.1.3. Frame de configuración 2

Para esta aplicación, todos los PMUs tienen una configuración de tipo 2. La organización del frame de configuración 2 se describe en el cuadro 3.5.

Posición	Campo	Bytes	Descripción y comentarios
1	SYNC	2	Descrito en el cuadro 3.2 ²
2	FRAMESIZE	2	Descrito en el cuadro 3.2
3	IDCODE	2	Descrito en el cuadro 3.2
4	SOC	4	Descrito en el cuadro 3.2
5	FRACSEC	4	Descrito en el cuadro 3.2
6	TIME.BASE	4	-Base de tiempo. Divisor de FRACSEC empleado en el cálculo para obtener el tiempo. Se compone de dos partes: Bits 31-24: Reservados Bits 23-0: Base de tiempo (Divisor de FRACSEC) Numero entero de 24 bits sin signo.
7	NUM.PMU	2	-Numero de PMUs que se recibirán en el frame de datos (Data frame) Numero entero de 16 bits sin signo.
8	STN	16	-Nombre de estación (definido en la programación del PMU) 16 bytes en formato ASCII (1 letra por cada byte)
9	IDCODE	2	-código de identificación de cada bloque Numero entero de 16 bits sin signo.
10	FORMAT	2	-Formato de presentación de los PMUs Bits 15-4: No se utilizan Bit 3: Tipo de numero con que se presenta la frecuencia (FREQ) y/o diferencial de frecuencia(DFREQ) 0= fijo con entero de 16-bits (requiere posterior procesamiento) 1= punto flotante de 32bits formato IEEE Bit 2: Tipo de numero con que se presentan los valores análogos (Analog)
continúa cuadro 3.5...			

²Cuadro de configuración general

Posición	Campo	Bytes	Descripción y comentarios
			0= entero de 16-bits (requiere Factor de conversión) 1= punto flotante de 32bits formato IEE Bit 1: Tipo de numero con que se presentan los fasores (Phasors) 0= entero de 16-bits (requiere Factor de conversión) 1= punto flotante de 32bits formato IEE Bit 0: Forma en que se presentan los fasores (Phasors) 0= RECTANGULAR (Parte real y parte imaginaria) 1= POLAR (magnitud y ángulo)
11	PHNMR	2	-Numero de Fasores que se recibirán en el frame de datos (Data frame)
12	ANNMR	2	-Numero de valores análogos que se recibirán en el frame de datos (Data frame)
13	DGNMR	2	-Número de entradas digitales que se recibirán en el frame de datos (Data frame)
14	CHNAM	16 x N	-Nombre de Fasores y Banderas(definido en la programación del PMU) 1 letra formato ASCII por cada byte N=No. de fasores+ No. Análogos + 16x No. de entradas digitales
15	PHUNIT	4x PHNMR	-Factor de conversión de Fasores Bits 31-24 (byte más significativo): HEX (00): Tensión (01): Corriente Bits 23-0 (3 Bytes menos significativos): Factor de conversión requerido en "FORMAT" Entero de 24 bits sin signo, multiplicado por un factor de 10 ⁵
16	ANUNIT	4x ANNMR	-Factor de conversión de análogos Bits 31-24: Bits 23-0:
17	DIGUNIT	4x DGNMR	-Mascaras para entradas digitales (definido por el usuario) Sugerencia de uso: Los 2 bytes más significativos pueden ser el estado predeterminado de las entradas Los 2 bytes menos significativos pueden ser el estado actual de las entradas
18	FNOM	2	-Frecuencia nominal Bits 15-1: Reservados (valor predeterminado 0) Bit 0: 0= FNOM 60 [Hz] 1= FNOM 50 [Hz]
19	CFGNCT	2	Contador de cambios de configuración Entero de 16 bits sin signo
-	REPITE 8-19	-	En caso de existir más de un PMU se repiten los puntos 8-19 por cada PMU adicional
continúa cuadro 3.5...			

Índice	Byte Hex		Binario		Descripción	Valor	
	+0	+1					
104	20	20	00100000	00100000			
106	20	20	00100000	00100000			
108	20	20	00100000	00100000			
110	41	54	01000001	01010100	Nom. Fasor	AT2-1	A T
112	32	2D	00110010	00101101			2 -
114	31	20	00110001	00100000			1
116	20	20	00100000	00100000			
118	20	20	00100000	00100000			
120	20	20	00100000	00100000			
122	20	20	00100000	00100000			
124	20	20	00100000	00100000			
126	41	33	01000001	00110011	Nom. Fasor	A3W30-TTH1-I	A 3
128	57	33	01010111	00110011			W 3
130	30	5F	00110000	01011111			0 -
132	54	54	01010100	01010100			T T
134	48	31	01001000	00110001			H 1
136	2D	49	00101101	01001001			- I
138	20	20	00100000	00100000			
140	20	20	00100000	00100000			
142	41	33	01000001	00110011	Nom. Fasor	A3W40-TTH2-I	A 3
144	57	34	01010111	00110100			W 4
146	30	5F	00110000	01011111			0 -
148	54	54	01010100	01010100			T T
150	48	32	01001000	00110010			H 2
152	2D	49	00101101	01001001			- I
154	20	20	00100000	00100000			
156	20	20	00100000	00100000			
158	50	48	01010000	01001000	Nom. Fasor	PHASOR8	P H
160	41	53	01000001	01010011			A S
162	4F	52	01001111	01010010			O R
164	38	20	00111000	00100000			8
166	20	20	00100000	00100000			
168	20	20	00100000	00100000			
170	20	20	00100000	00100000			
172	20	20	00100000	00100000			
174	54	45	01010100	01000101	Nom. canal digital	TEST-ACTIVE	T E
176	53	54	01010011	01010100			S T
178	2D	41	00101101	01000001			- A
180	43	54	01000011	01010100			C T
182	49	56	01001001	01010110			I V
184	45	20	01000101	00100000			E
186	20	20	00100000	00100000			
188	20	20	00100000	00100000			
190	42	31	01000010	00110001	Nom. canal digital	B1-VTMCB.OPEN	B 1
192	5F	56	01011111	01010110			- V
194	54	4D	01010100	01001101			T M

continúa cuadro 3.6 ...

Indice	Byte Hex		Binario		Descripción	Valor		
	+0	+1						
196	43	42	01000011	01000010			C	B
198	5F	4F	01011111	01001111			-	O
200	50	45	01010000	01000101			P	E
202	4E	20	01001110	00100000			N	
204	20	20	00100000	00100000				
206	42	32	01000010	00110010	Nom. canal digital	B2.VTMCB.OPEN	B	2
208	5F	56	01011111	01010110			-	V
210	54	4D	01010100	01001101			T	M
212	43	42	01000011	01000010			C	B
214	5F	4F	01011111	01001111			-	O
216	50	45	01010000	01000101			P	E
218	4E	20	01001110	00100000			N	
220	20	20	00100000	00100000				
222	45	58	01000101	01011000	Nom. canal digital	EXTERN.BI8	E	X
224	54	45	01010100	01000101			T	E
226	52	4E	01010010	01001110			R	N
228	5F	42	01011111	01000010			-	B
230	49	38	01001001	00111000			I	8
232	20	20	00100000	00100000				
234	20	20	00100000	00100000				
236	20	20	00100000	00100000				
238	45	58	01000101	01011000	Nom. canal digital	EXTERN.BI9	E	X
240	54	45	01010100	01000101			T	E
242	52	4E	01010010	01001110			R	N
244	5F	42	01011111	01000010			-	B
246	49	39	01001001	00111001			I	9
248	20	20	00100000	00100000				
250	20	20	00100000	00100000				
252	20	20	00100000	00100000				
254	4E	6F	01001110	01101111	Nom. canal digital	No Connection	N	o
256	20	43	00100000	01000011				C
258	6F	6E	01101111	01101110			o	n
260	6E	65	01101110	01100101			n	e
262	63	74	01100011	01110100			c	t
264	69	6F	01101001	01101111			i	o
266	6E	20	01101110	00100000			n	
268	20	20	00100000	00100000				
270	4E	6F	01001110	01101111	Nom. canal digital	No Connection	N	o
272	20	43	00100000	01000011				C
274	6F	6E	01101111	01101110			o	n
276	6E	65	01101110	01100101			n	e
278	63	74	01100011	01110100			c	t
280	69	6F	01101001	01101111			i	o
282	6E	20	01101110	00100000			n	
284	20	20	00100000	00100000				
286	4E	6F	01001110	01101111	Nom. canal digital	No Connection	N	o

continúa cuadro 3.6 ...

Indice	Byte Hex		Binario		Descripción	Valor			
	+0	+1							
380	20	20	00100000	00100000					
382	4E	6F	01001110	01101111	Nom. canal digital	No Connection	N	o C o n e t i o n	
384	20	43	00100000	01000011					
386	6F	6E	01101111	01101110					
388	6E	65	01101110	01100101					
390	63	74	01100011	01110100					
392	69	6F	01101001	01101111					
394	6E	20	01101110	00100000					
396	20	20	00100000	00100000					
398	4E	6F	01001110	01101111	Nom. canal digital	No Connection	N	o C o n e t i o n	
400	20	43	00100000	01000011					
402	6F	6E	01101111	01101110					
404	6E	65	01101110	01100101					
406	63	74	01100011	01110100					
408	69	6F	01101001	01101111					
410	6E	20	01101110	00100000					
412	20	20	00100000	00100000					
414	4E	6F	01001110	01101111	Nom. canal digital	No Connection	N	o C o n e t i o n	
416	20	43	00100000	01000011					
418	6F	6E	01101111	01101110					
420	6E	65	01101110	01100101					
422	63	74	01100011	01110100					
424	69	6F	01101001	01101111					
426	6E	20	01101110	00100000					
428	20	20	00100000	00100000					
430	00	47	00000000	01000111	Parametro medido:	tension			
432	E4	CE	11100100	11001110	factor	47.1163			
434	00	47	00000000	01000111	Parametro medido:	tension			
436	E4	CE	11100100	11001110	factor	47.1163			
438	01	02	00000001	00000010	Parametro medido:	Corriente			
440	FA	F6	11111010	11110110	factor	1.95318			
442	01	01	00000001	00000001	Parametro medido:	Corriente			
444	7D	7B	01111101	01111011	factor	0.97659			
446	01	03	00000001	00000011	Parametro medido:	Corriente			
448	B9	B4	10111001	10110100	factor	2.44148			
450	01	02	00000001	00000010	Parametro medido:	Corriente			
452	FA	F6	11111010	11110110	factor	1.95318			
454	01	02	00000001	00000010	Parametro medido:	Corriente			
456	FA	F6	11111010	11110110	factor	1.95318			
458	01	02	00000001	00000010	Parametro medido:	Corriente			
460	FA	F6	11111010	11110110	factor	1.95318			
462	00	00	00000000	00000000	Estado inicial	0000000000000000			
464	00	1F	00000000	00011111	Estado actual	0000000000011111			
466	00	00	00000000	00000000	Frecuencia nominal	60 [Hz]			
continúa cuadro 3.6 ...									

Indice	Byte Hex		Binario		Descripción	Valor
	+0	+1				
468	00	00	00000000	00000000	Contador de cambios en configuración	0
470	00	14	00000000	00010100	Frames por segundo	20
472	04	A4	00000100	10100100	CHK	04A4

Cuadro 3.6: Ejemplo de frame de configuracion 2

3.1.4. Frame de datos

La parte medular de los PMUs son los datos que este envía sobre las mediciones de los parámetros eléctricos, estos datos se reciben mediante un frame cuya estructura se describe en el cuadro 3.7:

Indice	Campo	Bytes	Descripción y comentarios
1	SYNC	2	Descrito anteriormente
2	FRAMESIZE	2	Descrito anteriormente
3	IDCODE	2	Descrito anteriormente
4	SOC	4	Descrito anteriormente
5	FRACSEC	4	Descrito anteriormente
6	STAT	2	- Estado del PMU Bits 15-14: Errores 00= Buenos datos de medición, no hay errores 01= No hay información acerca de los datos 10= PMU en modo de prueba (no usar valores) 11= Error general (no usar valores) Bit 13: sincronización PMU 0= sincronización con una fuente de tiempo UTC Bit 12-0: Definidos en 6.3 Data frame de IEEE Std C37.118.2-2011
7	PHASORS	4X PHNMR	-Valores de medición fasorial 4 bytes si las unidades son entero de 16 bits (2 valores) (definido en los frame de configuración) -Si se representan de forma rectangular, el valor real va primero. Tipo de numero: entero de 16 bits con signo. Rango [- 32 767 , 32 767] (Requieren procesamiento posterior) ³ . -Si se presenta en forma polar, la magnitud va primero. Tipo de número para magnitud: entero de 16 bits sin signo. Rango de magnitud: [0 , 65535] (Requieren procesamiento posterior) ⁴ .
continúa...			

³Posteriormente se dividen entre el factor de conversión indicado en PHUNIT del frame de configuración

⁴Posteriormente se dividen entre el factor de conversión indicado en PHUNIT del frame de configuración

Indice	Campo	Bytes	Descripción y comentarios
			Tipo de número para ángulo: entero de 16 bits con signo. Rango de Angulo (-31416, 31416) ⁵ 8 bytes si las unidades son punto flotante de 32bits (2 valores) (definidos en los frame de configuración) -Si se representan de forma rectangular, el valor real va primero -Si se presenta en forma polar, la magnitud va primero Tipo de numero para todos los casos: IEEE floating-point format
8	FREQ	4 o 2	-Valores de medición de frecuencia: 2 bytes si se representa con numero entero de 16 bits (definidos en los frame de configuración) -Tipo de número: entero de 16 bits con signo Rango [- 32 767 , 32 767] ⁶ 4 bytes si el tipo de numero es punto flotante de 32bits IEEE (definidos en los frame de configuración)
9	DFREQ	4 o 2	-Valores de medición de diferencial de frecuencia: Mismo tratamiento que FREQ
10	ANALOG	2 X ANNMR 4 X ANNMR	-Valores análogos Son mediciones de tipo analógico de instrumentos añadidos o externos al PMU. Los rangos y usos son definidos por el usuario. 2 bytes si las unidades son entero de 16 bits 4 bytes si las unidades son punto flotante de 32bits (definidos en los frame de configuración)
11	DIGITAL	2 X DGNMR	-Estados digitales Son banderas de tipo digital cuyos propósitos son definidos por el usuario
-	REPITE 6-11	-	En caso de existir más de un PMU se repiten los puntos 6-11 por cada PMU adicional
Ultima	CHK	2	Descrito anteriormente

Cuadro 3.7: Frame de fatos

Ejemplo de recepción y decodificación de frame de datos

Una vez recibida y decodificada la configuración, el siguiente paso es recibir y decodificar los datos de las mediciones. Para ello, primero se envía el comando “Iniciar transmisión de datos”, a lo cual el PMU responde la siguiente trama:

AA01 005C 0001 54D3 BECE 000D BB9F 0000 47DC 8776 484B B366 47DA 9A26 484A E8AB
3DA1 590B BD32 CDCE 3B01 AFEC BD25 E82F 3DD3 9BE2 3CD5 79A5 3BD2 8940 3EF9 7968

⁵(para determinar el ángulo se multiplica este valor por 0.0001)(rango: $[-\pi, \pi]$).

⁶(para calcular la frecuencia se le suma a la frecuencia nominal (FNOM del frame de configuración) el valor de FEQ multiplicado por 0.001)

BD3B 0709 BDFA 83DC 3DDB FC3C BDC4 4752 4270 2AB4 3C4D 429C 0000 116A

De acuerdo a los datos del frame de configuración 2 adquirido anteriormente (cuadro 3.6); los Fasores, valores análogos, frecuencia y diferencial de frecuencia se han enviado en formato “punto flotante de 32 bits IEE” (descrito en los bytes 38 y 39) por lo tanto se tomaran 8 bytes por cada uno de estos valores.

Basándose en esa información se decodifica el frame de datos auxiliándose del cuadro .

Indice	HEX +0 +1		BIN		Descripción	Valor
0	AA	01	10101010	00000001	Tipo de instrucción Versión	Datos IEEE Std C37.118- 2005 V1
2	00	5C	00000000	01011100	Bytes de lectura:	92
4	00	01	00000000	00000001	Codigo de identificacion:	1
6	54	D3	01010100	11010011	Segundos ⁷	1423163086
8	BE	CE	10111110	11001110		
10	00	0D	00000000	00001101	Calidad del tiempo	00
12	BB	9F	10111011	10011111	Fraccion de segundo:	899999
14	00	00	00000000	00000000	Estado de la señal	buenos datos de medición, no hay errores
16	47	DC	01000111	11011100	Fasor 1	47DC8776
18	87	76	10000111	01110110		
20	48	4B	01001000	01001011		
22	B3	66	10110011	01100110		
24	47	DA	01000111	11011010	Fasor 2	47DA9A26
26	9A	26	10011010	00100110		
28	48	4A	01001000	01001010		
30	E8	AB	11101000	10101011		
32	3D	A1	00111101	10100001	Fasor 3	3DA1590B
34	59	0B	01011001	00001011		
36	BD	32	10111101	00110010		
38	CD	CE	11001101	11001110		
40	3B	01	00111011	00000001	Fasor 4	3B01AFEC
42	AF	EC	10101111	11101100		
44	BD	25	10111101	00100101		
46	E8	2F	11101000	00101111		
48	3D	D3	00111101	11010011	Fasor 5	3DD39BE2
50	9B	E2	10011011	11100010		
52	3C	D5	00111100	11010101		
54	79	A5	01111001	10100101		
56	3B	D2	00111011	11010010	Fasor 6	3BD28940
58	89	40	10001001	01000000		
continúa cuadro 3.8...						

⁷transcurridos desde 1/1/1970

Indice	HEX		BIN		Descripción	Valor
	+0	+1				
60	3E	F9	00111110	11111001		3EF97968
62	79	68	01111001	01101000		
64	BD	3B	10111101	00111011	Fasor 7	BD3B0709
66	07	09	00000111	00001001		
68	BD	FA	10111101	11111010		
70	83	DC	10000011	11011100		
72	3D	DB	00111101	11011011	Fasor 8	3DDBFC3C
74	FC	3C	11111100	00111100		
76	BD	C4	10111101	11000100		
78	47	52	01000111	01010010		
80	42	70	01000010	01110000	Frecuencia	42702AB4
82	2A	B4	00101010	10110100		
84	3C	4D	00111100	01001101	Delta de frecuencia	3C4D429C
86	42	9C	01000010	10011100		
88	00	00	00000000	00000000	Digital	0000000000000000
90	11	6A	00010001	01101010	CHK	116A

Cuadro 3.8: Ejemplo de frame de datos

De manera inmediata se tiene que:

- Se recibieron todos los bits.
- El tiempo está en operación normal, y sincronizado con el UTC.
- El PMU está operando de forma correcta (buenos datos de medición, no hay errores).
- El código de verificación por redundancia cíclica es correcto (se requieren operaciones externas)

Para conocer los datos de las mediciones (tiempo, fasores, valores análogos y digitales), se necesita procesar los valores del frame de datos (tab. 3.8), con base en la información del frame de configuración (tab. 3.6).

Decodificación de tiempo

- Del frame de configuración (tab. 3.6), se tiene que el tiempo base es:
TIME_BASE=1000000
- Del frame de datos (tab. 3.8), se tiene que los segundos son:
SOC= 1423163086
- Del frame de datos (tab. 3.8), se tiene que la Fraccion de segundo es:
FRACSEC=899999

Estos datos son sustituidos en la ecuación descrita en “calculo de tiempo” (Ec. 3.1)teniendo asi:

$$tiempo = SOC + \frac{(FRACSEC)}{(TIME_BASE)} = 1423163086 + \frac{899999}{1000000} = 1423163086,899999$$

Como se indica en el cuadro 3.5 este valor representa los segundos transcurridos desde la media noche del 1° de enero de 1970.

Agregando a esa fecha el valor obtenido del cálculo, se tiene que la fecha y hora de la muestra es:

05/02/2015 01:04:46.900 p.m.

Decodificación de valores Fasoriales

- Del frame de configuración 2 (tab. 3.6), se tiene que el tipo de numero de número de los Fasores es:

Punto flotante de 32 bits IEE.

Por lo tanto:

- Se compone de dos valores de 32 bits (8 bytes en total como se indico al inicio del ejemplo).
 - No necesita factores de conversión ni ecuaciones auxiliares.
- Del frame de configuración 2 (tab. 3.6), se tiene que la forma de representación de los Fasores es:

Rectangular

Por lo tanto:

- La parte real del número complejo se expresa en los primeros 4 bytes
 - La parte imaginaria del número complejo se expresa en los últimos 4 bytes
- Del frame de configuración 2 (tab. 3.6), se tiene que los nombres de estos Fasores son:
 - V1_B2
 - V1_B1
 - A3370_PRD-I
 - AT1I
 - AT21
 - A3W30_TTH1-I
 - A3W40_TTH2-I

- PHASOR8

- Del frame de configuración 2 (tab. 3.6), los parámetros medidos por los Fasores son:

- tensión
- tensión
- corriente
- corriente
- corriente
- corriente
- corriente
- corriente

- Del frame de configuración 1 (tab. 3.8), se toman los valores hexadecimales de los fasores, los cuales se convierten y organizan de acuerdo al frame de configuración 2 (tab. 3.6) lo cual nos da finalmente:

Hex	Dec	Rectangular	Nombre	Unidades
47DC8776 484BB366	112911 208590	112911+208590j	V1.B2	V
47DA9A26 484AE8AB	111924 207779	111924+207779j	V1.B1	V
3DA1590B BD32CDCE	0.0787831 -0.0436533	0.0787831-0.0436533j	A3370_PRD-I	A
3B01AFEC BD25E82F	0.00197887 -0.0405046	0.00197887-0.0405046j	AT1-I	A
3DD39BE2 3CD579A5	0.103325 0.026059	0.103325+0.026059j	AT2-1	A
3BD28940 3EF97968	0.00642505 0.487254	0.00642505+0.487254j	A3W30_TTH1-I	A
BD3B0709 BDF A83DC	-0.045661 -0.122322	-0.045661-0.122322j	A3W40_TTH2-I	A
3DDBFC3C BDC44752	0.107415 -0.0958392	0.107415-0.0958392j	PHASOR8	A

Cuadro 3.9: valores decodificados

Capítulo 4

Implementación

1

4.1. Breve descripción del entorno de Labview

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) es un sistema de desarrollo basado en programación gráfica, orientado a desarrollar aplicaciones para instrumentación que integra una serie de librerías para comunicación con instrumentos electrónicos como GPIB, RS232 o RS485 con tarjetas de adquisición de datos, sistemas de adquisición y acondicionamiento como VXI o SCXI, comunicaciones en redes TCP/IP, UDP, o en los estándares de software COM, OLE, DDE, DLL o ActiveX para Windows, así como AppleEvents para MacOS o PIPE para UNIX.

Los programas realizados en LabVIEW se llaman instrumentos virtuales “Vis”, ya que tienen la apariencia de los instrumentos reales, sin embargo, poseen analogías con funciones provenientes de lenguajes de programación convencionales.

Las principales características de los Vis se pueden describir como:

- Los Vis contienen una interfaz interactiva de usuario, la cual se llama panel frontal, ya que simula el panel de un instrumento físico. Se puede entrar datos usando el teclado o el ratón y tener una visualización de los resultados en la pantalla del computador. El Panel Frontal es la interfaz hombre-máquina de un VI.
- Los Vis reciben instrucciones de un diagrama de bloques construido en lenguaje G el cual suministra una solución gráfica a un problema de programación. El diagrama de bloques es el código fuente de un VI.
- Los Vis usan una estructura hereditaria y modular que permite realizar programas por niveles o hacer programas con otros programas o sub programas. Un VI contenido en otro VI es denominado subVI. Todo VI se puede convertir en subVI sin ningún tipo de cambio en su estructura.

¹Curso LabVIEW6i (Bibliografía [3])

4.2. PROGRAMACIÓN GRÁFICA APLICADA A ADQUISICIÓN Y PUBLICACIÓN DE INFORMACIÓN PRO

Con estas características LabVIEW permite dividir un programa en una serie de tareas las cuales son divisibles nuevamente hasta que una aplicación complicada se convierte en una serie de sub tareas simples. Todos los anteriores conceptos están de acuerdo con las concepciones modernas de la programación modular.

Además LabVIEW puede ser usado con poca experiencia en programación pues utiliza metodologías familiares a técnicos, ingenieros, doctores y la comunidad científica en general.

Cada VI de LabVIEW cuenta con dos interfaces: panel frontal y diagrama de bloques. Éstas cuentan con paletas que contienen los objetos necesarios para implementar y desarrollar tareas. La figura (poner referencia e imagen) muestra estas interfaces dentro de un entorno Windows.

(falta imagen)

4.1.1. Panel Frontal

Es la interface gráfica que simula el panel de un instrumento real, permite la entrada y salida de datos, puede contener pulsadores, perillas, botones, gráficos y en general controles e indicadores. Figura (poner referencia e imagen).

(falta imagen)

Los controles son objetos que sirven para introducir datos al programa y pueden ser manipulados por el usuario. Los controles son variables de entrada. Los indicadores sirven para presentar los resultados entregados por el programa y no pueden ser manipulados por el usuario. Los indicadores son variables de salida.

4.1.2. Diagrama De Bloques

El diagrama de bloques contiene el código fuente gráfico del VI, posee funciones y estructuras que relacionan las entradas con las salidas creadas en el panel frontal. En un diagrama se distinguen: Terminales, que representan los controles e indicadores del panel. Funciones y SubVIs, que realizan tareas específicas. Estructuras y Cables que determinan el flujo de los datos en el programa. En general, cualquiera de estas partes del diagrama de un VI se denomina NODO.

4.2. Programación gráfica aplicada a adquisición y publicación de información proveniente de los equipos de medición.

Como se mencionó en el capítulo 1.1.4, uno de los retos a resolver es la limitada cantidad de equipos que pueden consultar la información de un PMU al mismo tiempo. Para este fin, la solución implementada fue centralizar la

capacidad de manipulación y consulta de todos los PMUs en un solo equipo; este equipo obtendrá los frames de configuración y de datos, y los publicara mediante variables compartidas

4.3. Adquisición.vi

Se trata de un programa cuya función es establecer las comunicaciones con los PMUs, recibir los datos binarios de los mismos y compartirlos. Este programa solo se ejecuta en el equipo central al cual responden los PMUs. Se basa en dos objetivos esenciales: recibir la información proveniente de los PMUs y publicarla mediante variables compartidas.

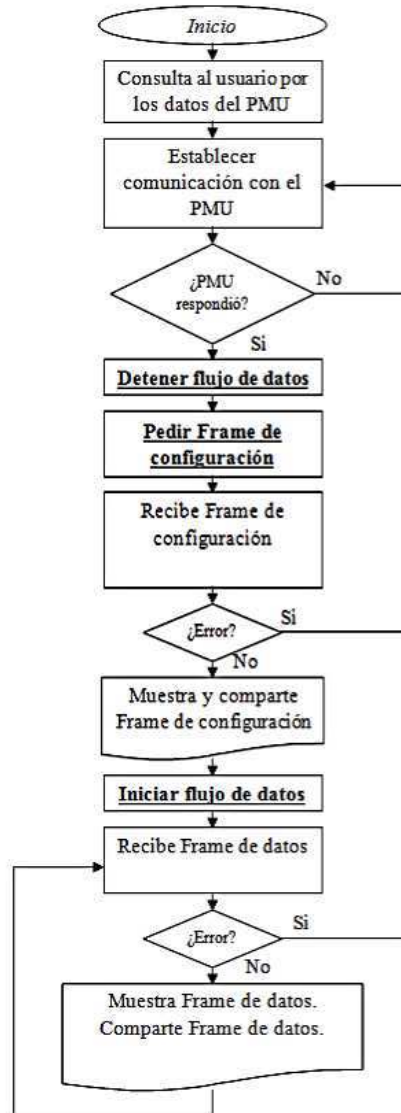


Figura 4.1: Adquisición Algoritmo Básico

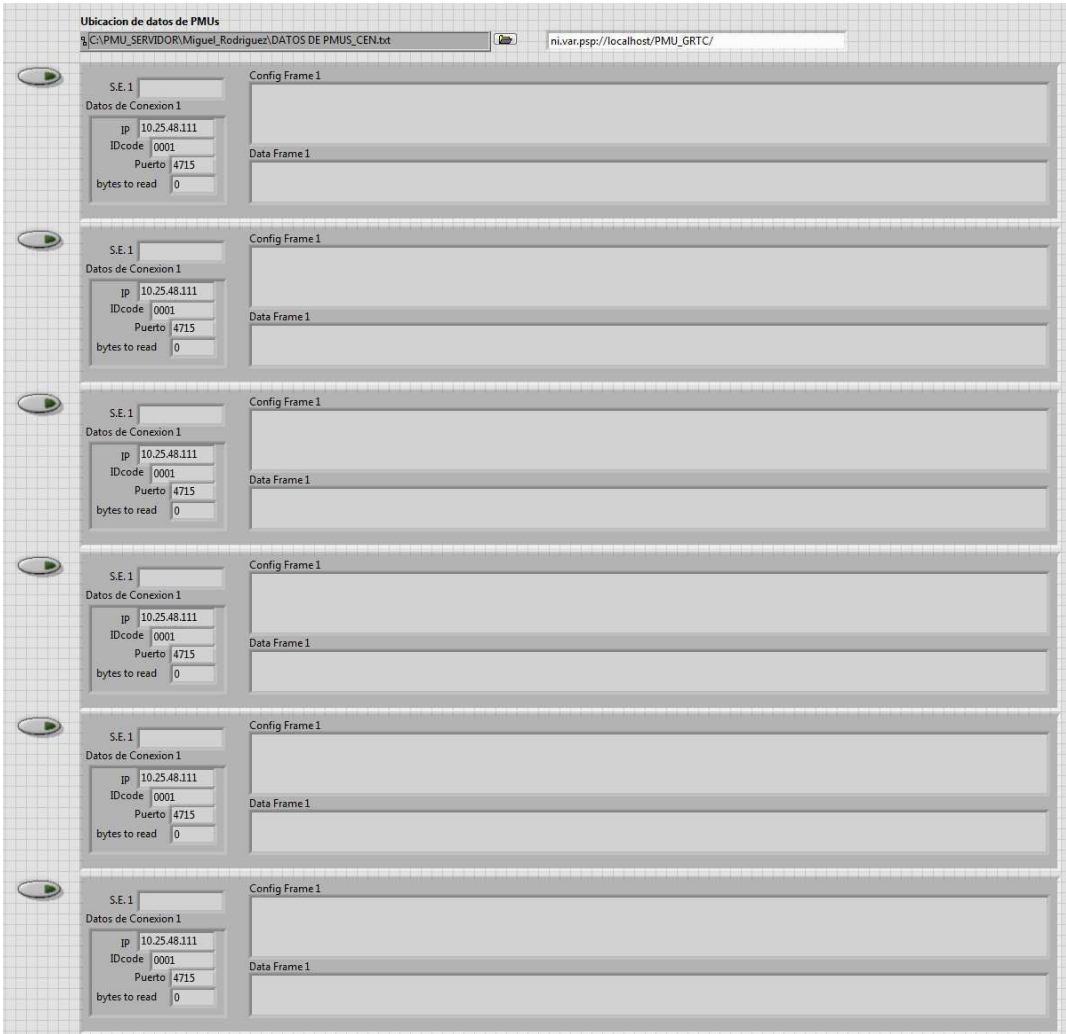


Figura 4.2: Adquisición Panel Frontal

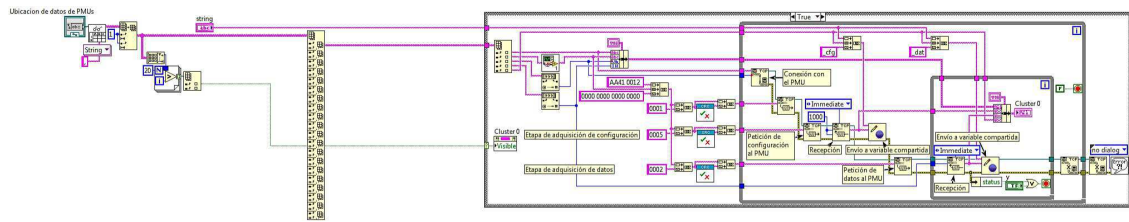


Figura 4.3: Adquisición Diagrama de Bloques

4.3.1. Consulta de los datos de los PMUs

En el caso particular de esta aplicación, los datos como dirección IP del equipo, puerto, código de identificación (ID) y bits de lectura se encuentran en un archivo “.txt”. Por lo tanto en panel frontal del VI de adquisición se especifica la ruta y nombre del archivo mediante el control de dirección mostrado en la figura: 4.3.1

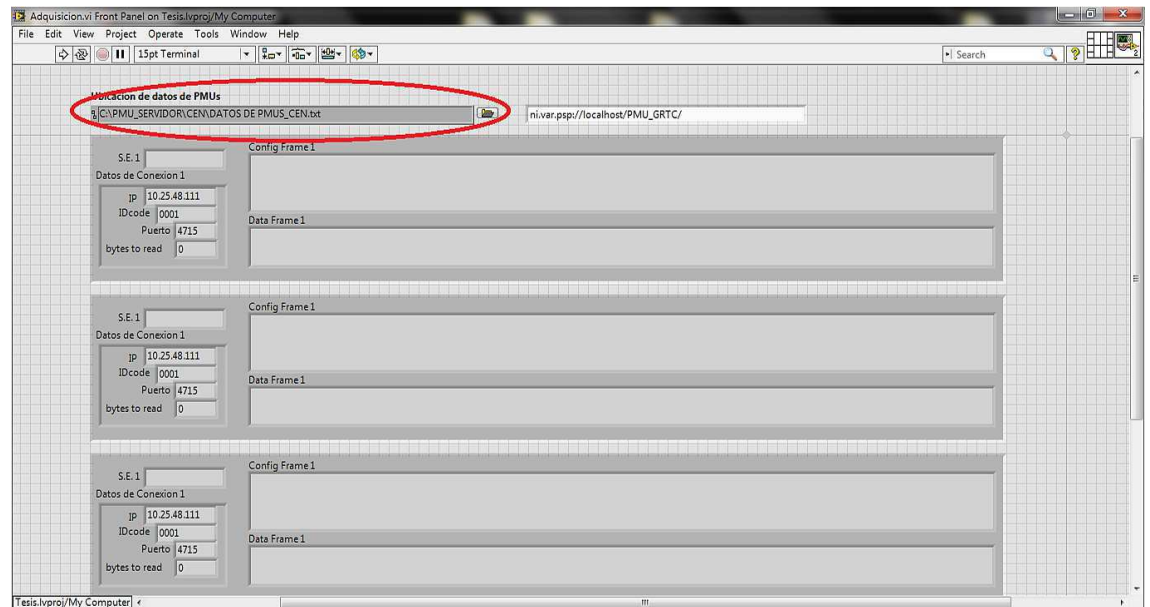


Figura 4.4: Control de dirección

La estructura que se sigue en el archivo es:

No,IP,IDCODE,Puerto,Bytes_de_lectura,Identificador

Dónde:

- No: Índice que indica su posición en el archivo .txt (sin otro uso relevante)
- IP: Dirección TCP/IP del PMU
- IDCODE: Código de identificación (ID) del PMU
- Puerto: Puerto de escucha del PMU
- Bytes_de_lectura: Bytes que contendrá el frame de datos
- Identificador: Nombre asignado por la compañía, para identificar la sub-estación de donde provienen los datos.

Ejemplo de texto en el archivo:

No,IP,IDCODE,Puerto,Bytes_de_lectura,Identificador

1,10.25.182.74,1,4712,92,CEN_TUL2
 2,10.25.156.36,16,4715,474,CEN_TCL1
 3,10.25.147.3,1,4715,84,CEN_PBD1

Nótese que no lleva espacios y los parámetros van delimitados por comas.

4.3.2. Procesamiento de datos generales de los PMU

Una vez indicada la dirección del archivo con los datos de los PMUs, en el diagrama de bloques se “interconectan” los nodos y VIs que seccionaran la información para su posterior uso, como se muestra en la figura 4.5:

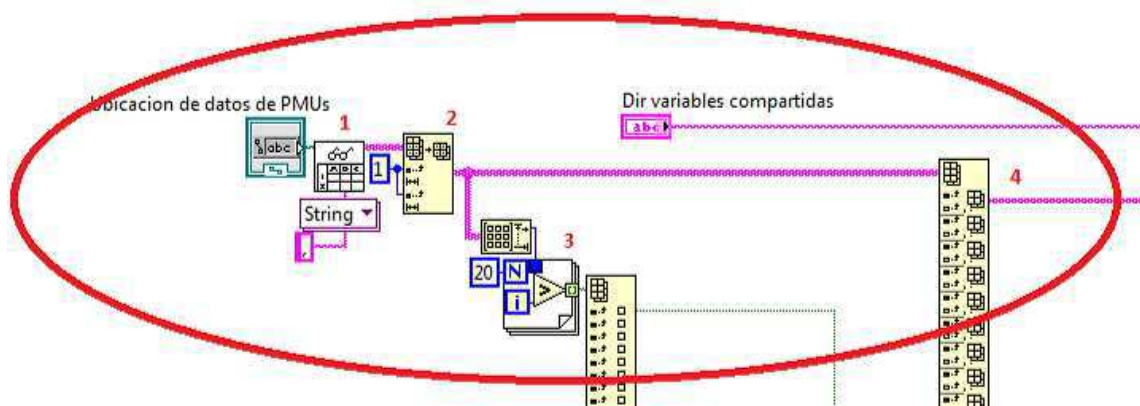


Figura 4.5: adquisición 1

La función de esta sección es:

1. Transformar el texto a una “tabla” (arreglo bidimensional) de valores “string” (cadena de caracteres)
2. Eliminar los datos irrelevantes (como encabezado e índice del archivo)
3. Contar el número de PMUs que se procesaran.

4. Seccionar el arreglo bidimensional (los datos de todos los PMUs) en arreglos unidimensionales (datos correspondientes de cada PMU).

El número de PMUs se muestra como un arreglo de valores booleanos (cable verde) mientras los valores se muestran como un arreglo bidimensional de cadenas de caracteres “string” (cable rosa). Al término de este proceso se tienen los datos del PMU extraídos del archivo .TXT listos para su utilización en el programa.

4.3.3. Procesamiento de los datos particulares de un PMU.

Una vez procesados los datos generales de todos los PMUs se procede a procesar los datos particulares de cada PMU en su módulo correspondiente como se muestra en la figura: 4.6

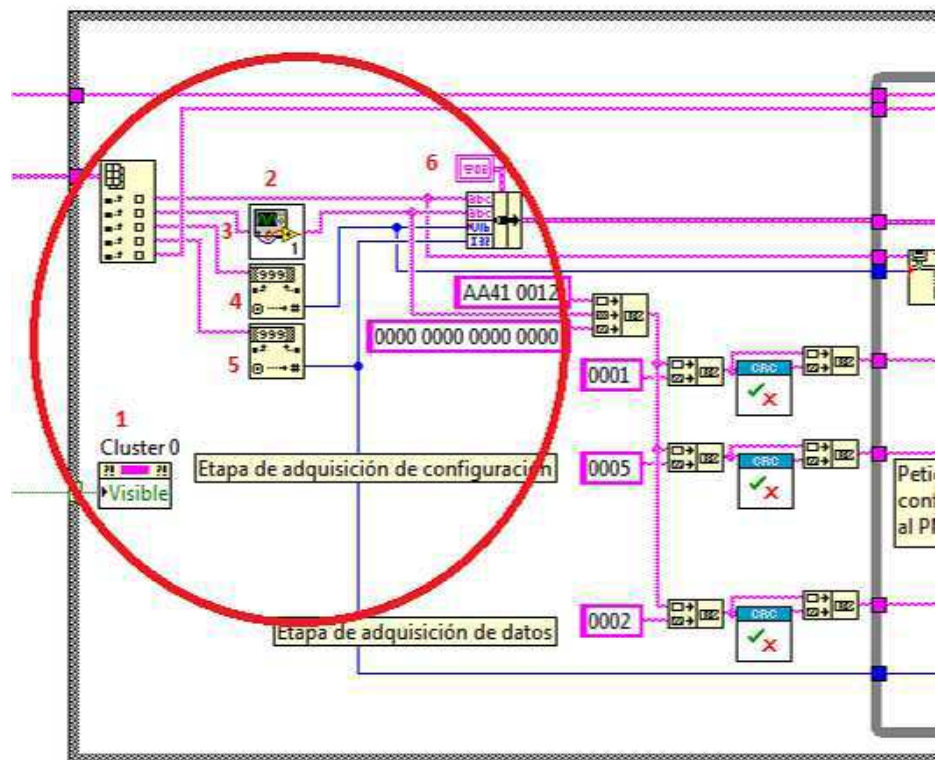


Figura 4.6: adquisición 2

La función de esta sección es:

1. Habilitar o deshabilitar su procesamiento y ocultar o hacer visible su indicador “Cluster n” en el panel frontal.
2. Pasar la dirección IP del equipo directamente.

3. Convertir el código de identificación del PMU de código ASCII a hexadecimal
4. Convertir el puerto de escucha del PMU de código ASCII a decimal.
5. Convertir los bites de lectura del frame de datos de código ASCII a decimal.
6. Concentrar todos los datos anteriores en un clúster (equivalente a una variable de tipo objeto en programación orientada a objetos).

Al termino de este proceso se tienen los datos del PMU agrupados en un clúster (cable rosa con puntos blancos) y de forma individual para su utilización (IP e ID_code cables rosas, Puerto y Bytes de lectura cables azules).

4.3.4. Preparación de los frame de comandos

Como se puede observar en el diagrama de flujo del programa adquisición (en la figura (poner referencia)), el programa utiliza básicamente tres comandos para comunicarse con el PMU:

1. Detener flujo de datos
2. Enviar frame de configuración 2
3. Iniciar flujo de datos

Para que se puedan enviar dichos comandos y que el PMU los procese y responda, se tiene que preparar un frame con la estructura definida en el capítulo 3.1.1

Para la elaboración de estas tramas, de forma teórica, su estructura debería ser diseñada de manera individual y con datos específicos de tiempo, PMU e instrucción; pero de forma práctica se puede tomar una trama “común” la cual solo variará de un PMU a otro por el código de identificación del PMU, y de un comando a otro por el código de dicho comando.

En forma programada, este proceso se implementó como se muestra en la figura 4.7

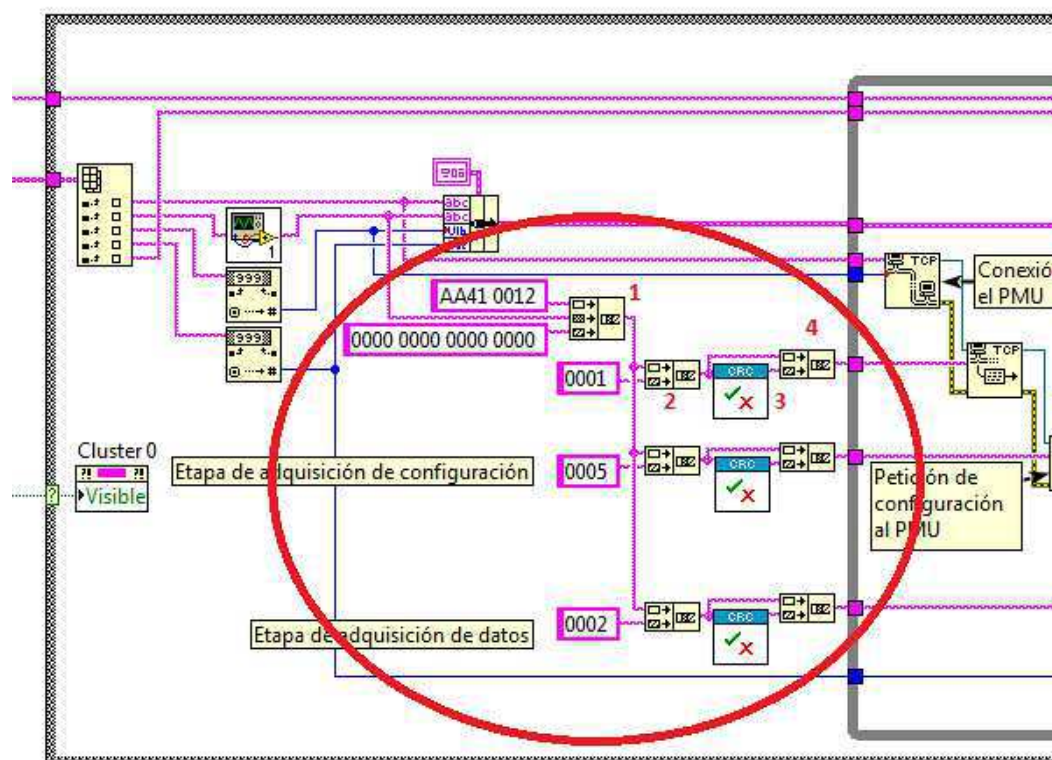


Figura 4.7: adquisición 3

La función de esta sección es:

1. Concatenar los valores constantes e introducir el código de identificación "ID" del PMU en su lugar correspondiente de acuerdo a lo establecido en el capítulo 3.1.1.
2. Concatenar la trama específica del PMU con el código específico del comando, tal como se explica en el capítulo 3.1.2.
 - 0001: "Detener flujo de datos"
 - 0005: "Enviar frame de configuración 2"
 - 0002: "Iniciar flujo de datos").
3. Calcular el CHK del CRC
4. Concatenar el CHK Con la trama específica de cada comando.

Al termino de este proceso se tienen los 3 frames de comandos listos para ser enviados (3 cables rosas).

4.3.5. Comunicación con el PMU y cese del flujo de datos.

Una vez que se tienen los datos del PMU y los comandos que se le enviarán, se procede a establecer la comunicación con el equipo.

El protocolo para esta aplicación es TCP/IP por lo que se requiere la dirección IP del equipo, el puerto y el código de identificación (intrínseco en los comandos que se prepararon en el capítulo anterior).

En LabVIEW existen 4 nodos para comunicaciones TCP/IP

1. Inicio de comunicaciones
2. Escritura
3. Lectura
4. Cierre de comunicaciones

Todas ellas interactúan entre sí mediante un conector específico de comunicaciones (cable verde) y un conector de estado, el cual indica si existe error o el proceso se realizó exitosamente (cable amarillo con puntos negros). Los comandos seguirán al puerto abierto por el nodo de inicio de comunicaciones y ese se mantendrá así hasta que llegue al cierre de comunicaciones.

En caso de que se genere un error, los nodos de escritura y lectura dejarán de operar y arrojarán una excepción o mensaje el cual seguirá hasta que sea corregido (generalmente hasta que se cierra el puerto). En forma programada, el proceso de comunicación con el PMU y la primera instrucción se implementó como se muestra en la figura: 4.8

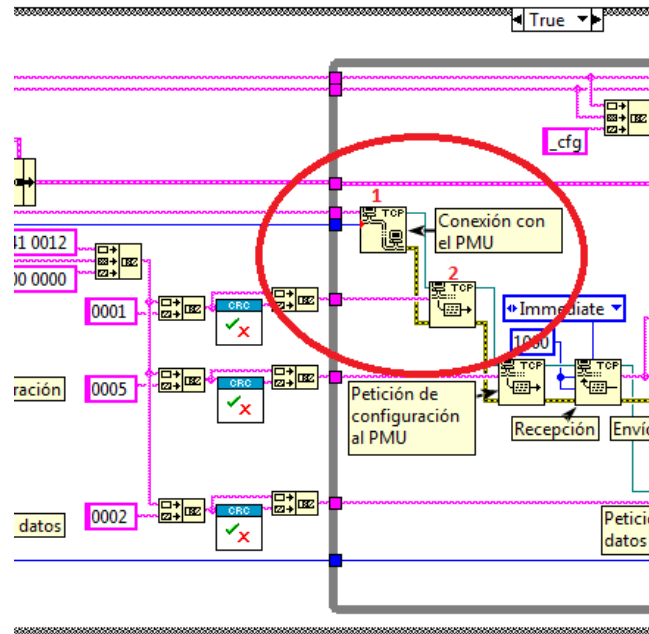


Figura 4.8: adquisición 4

La función de esta sección es:

1. Establecer comunicación con el PMU mediante el protocolo TCP/IP, indicando la dirección IP del PMU (cable rosa) y el puerto de escucha (cable azul).
2. Una vez establecida la comunicación (cables verde y amarillo), se envía el frame de comando correspondiente a la instrucción “detener flujo de datos” (cable rosa)

Al término de este proceso se tiene que el PMU ya no está enviando información, y se tiene listo el puerto de comunicaciones TCP/IP (cables verde y amarillo).

En caso de que la comunicación con el PMU no se hubiese logrado o fuera errónea, los demás procesos se deshabilitan mediante el conector amarillo hasta que llegue a la etapa de cierre y encontrándose en un ciclo while el programa regresaría a este punto para intentar establecer la comunicación nuevamente.

4.3.6. Petición, recepción y publicación del frame de configuración

El siguiente paso es enviar el frame de comando con la instrucción: “enviar frame de configuración 2”, recibir la trama de dicho parametro y publicarla.

En forma programada, este proceso se implementó como se muestra en la figura 4.9

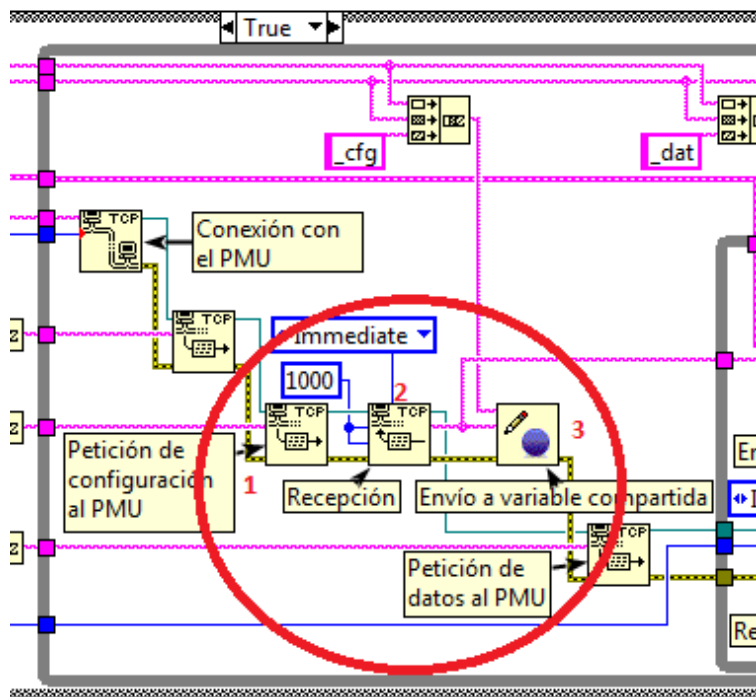


Figura 4.9: adquisición 5

La función de esta sección es:

1. Enviar el frame de comando con la instrucción: “enviar frame de configuración 2” mediante el nodo de escritura TCP/IP de labview
2. Recibir el frame de configuración 2 (cable rosa) mediante el nodo de lectura (parámetros: 1000bytes de lectura y recepción inmediata)
3. Publicar el frame de configuración mediante la variable compartida “nombre del PMU_cfg” de tipo string.

Este proceso se repetirá en un ciclo while hasta que se encuentre un error o hasta que se detenga el programa.

En forma programada, se implementó como se muestra en la figura 4.11

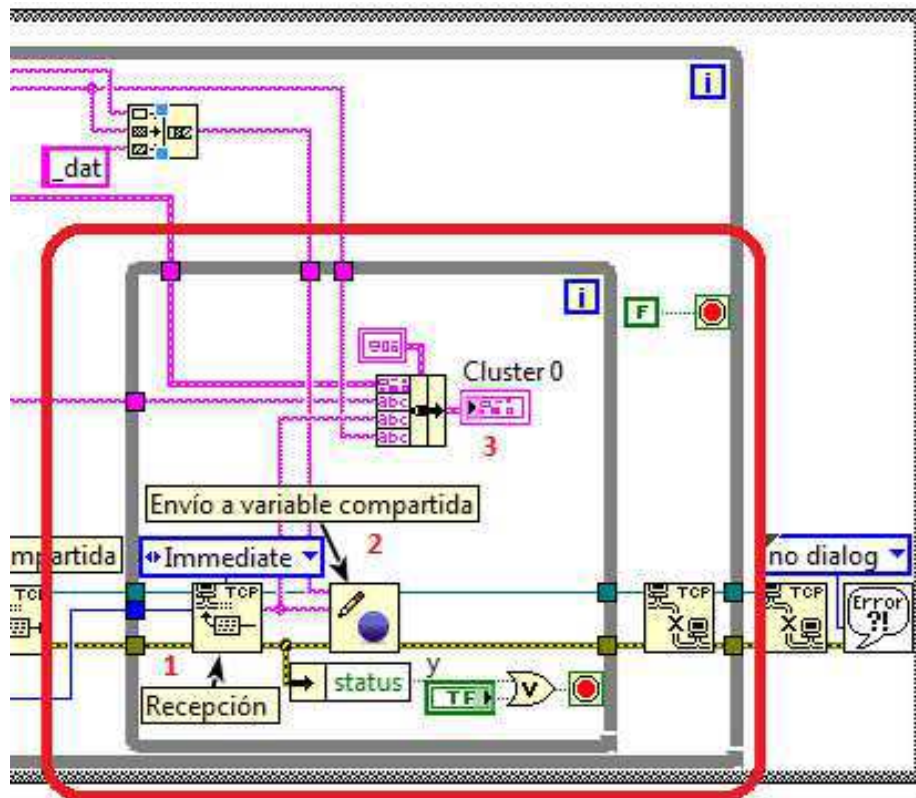


Figura 4.11: adquisición 7

Las funciones de esta sección son:

1. Recibir el frame de configuración
2. Publicar el frame de configuración
3. Concentrar los datos de adquisición del PMU, el nombre de identificación del PMU, el frame de configuración y el frame de datos más reciente en un clúster.

Al término de este proceso se tiene que:

El frame de datos se ha compartido y guardado mediante una variable compartida de tipo string.

4.4. Visualizador.vi

Se trata de un programa que recibirá los datos publicados por el programa de adquisición y mostrara los valores de los MPU en forma gráfica y en tiempo real.

Este programa se ejecutara en aquellos equipos conectados a la red del equipo central, los cuales requieran únicamente visualizar valores Las funciones de este programa son:

- Adquirir los frames de configuración y datos, publicados por equipo central mediante el programa de adquisición.
- Decodificar los frames de datos y configuración, y transformarlos en valores fasoriales.
- Darle el procesamiento necesario a los valores fasoriales y mostrarlos mediante graficas o indicadores.

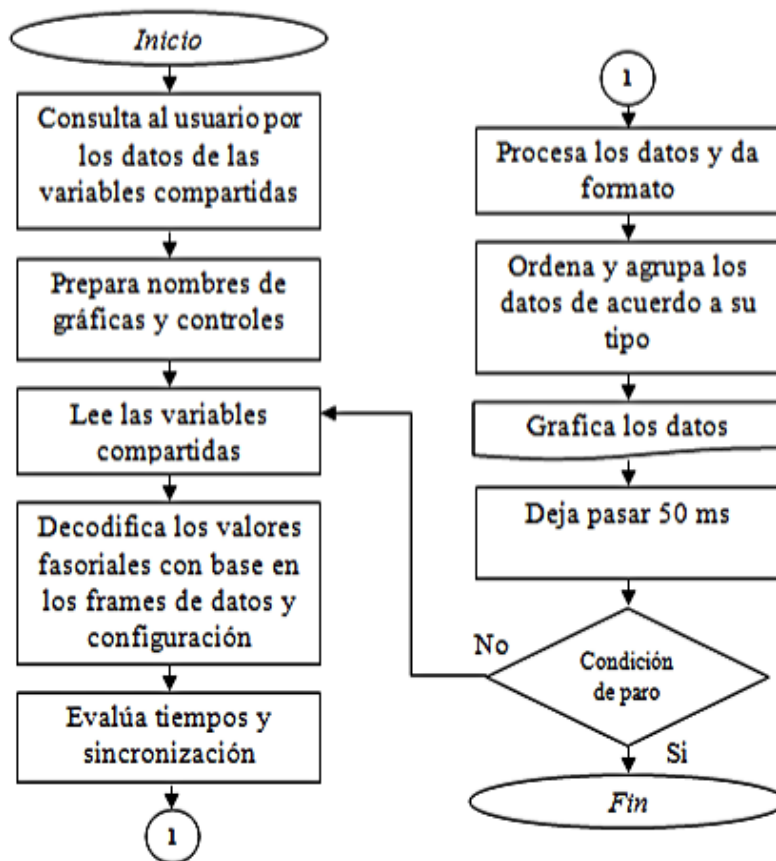


Figura 4.12: Visualizador Algoritmo Básico

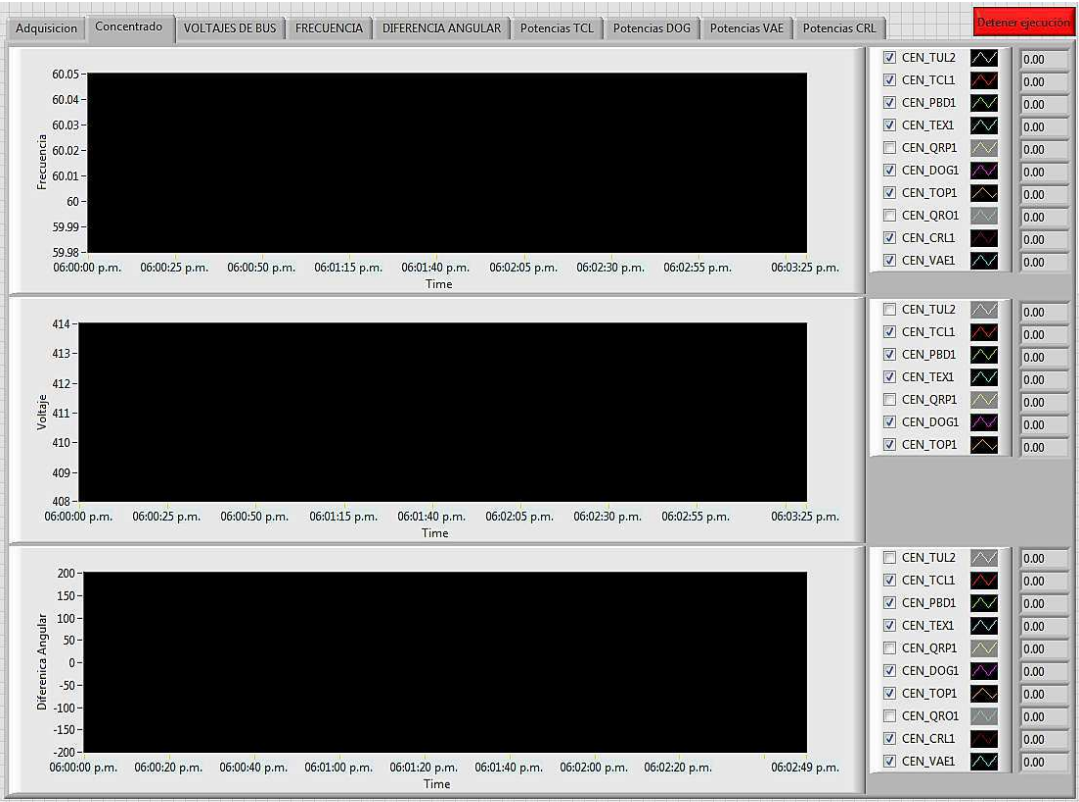


Figura 4.13: Visualizador Panel frontal

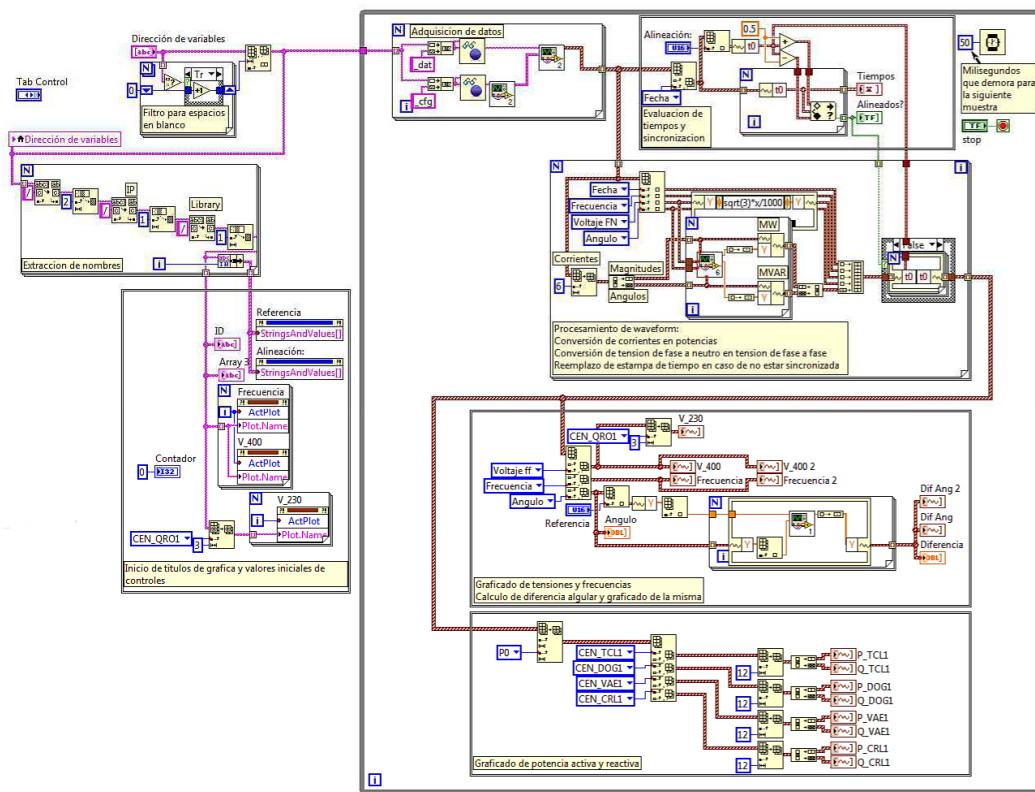


Figura 4.14: Visualizador Diagrama de Bloques

4.4.1. Consulta de los datos de las variables compartidas

Para este proyecto como se explicó en el capítulo 4.3, el programa de “adquisición” se encarga de adquirir y publicar los frames de datos y configuración de cada PMU en el equipo central mediante “variables compartidas”; estas variables compartidas son un tipo de dato propio de LabView que permite compartir información entre distintos programas y equipos (siempre que estén conectados a la misma red). Por lo tanto el primer paso de este programa “Visualizador” es obtener los datos de dichas variables para su posterior aprovechamiento.

Para este fin, en el panel frontal, en la pestaña de adquisición se encuentra un control de tipo string para introducir la dirección de las variables como se muestra en la figura 4.15



Figura 4.15: Visualizador 1

4.4.2. Procesamiento de datos de las variables compartidas en inicialización de títulos de graficas

Una vez insertadas las direcciones de las variables, estas se procesan para extraer los identificadores de los PMU y para leer sus datos.

En forma programada se implementó como se muestra en la figura 4.16

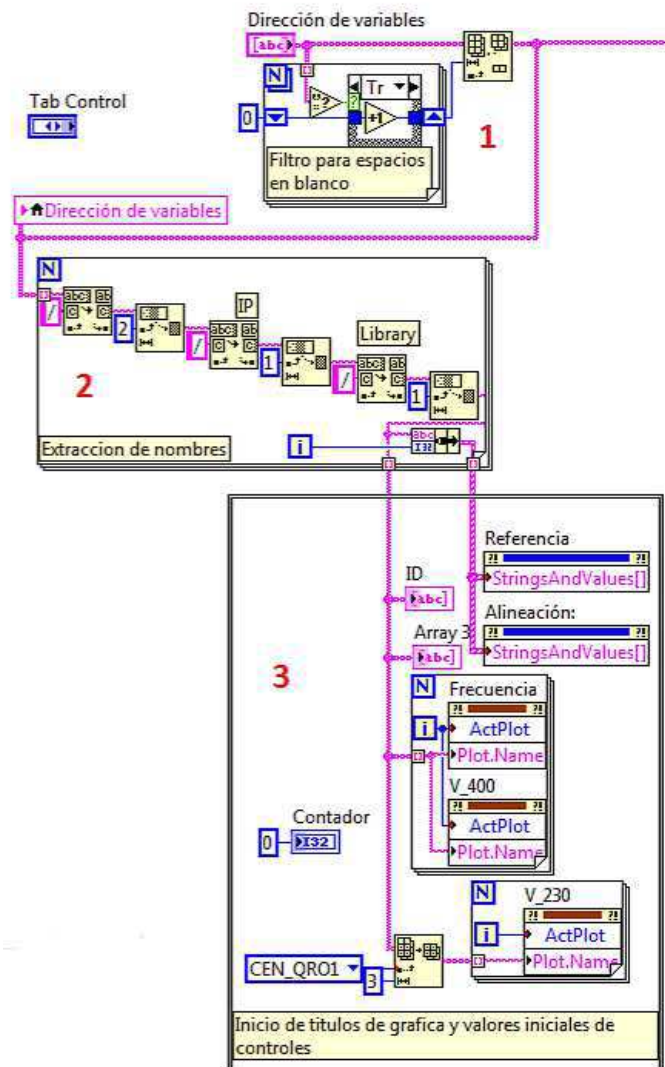


Figura 4.16: Visualizador 2

La función de esta sección es:

1. Eliminar los espacios en blanco del control de dirección de variables (para evitar errores en la lectura de las variables compartidas).
2. Extraer los identificadores de las subestaciones.
3. Inicializar los nombres de las tablas, gráficas y controles. Las direcciones de las variables se muestran como un arreglo de tipo "string" (cable rosa).

Al término de este proceso se tienen las direcciones de las variables, y las gráficas, tablas y controles tienen los títulos correspondientes de cada identificador.

4.4.3. Lectura de variables compartidas

Una vez que se tienen los datos de las direcciones de las variables, estos se introducen a un nodo cuya función es buscar y entregar el contenido de las mismas.

Puesto que el número de variables que se van a procesar es indefinido, estos están agrupados en un arreglo, el cual se introducen en un ciclo FOR que realiza el proceso de lectura de variables y decodificación, tantas veces como el tamaño del arreglo (razón por la cual se filtran los espacios en blanco). A la salida del ciclo se tiene un segundo arreglo de variables de tipo waveform, esta variable es propia del lenguaje LabView y contiene tanto el valor numérico del parámetro como la hora del PMU y es empleada para mostrar los valores en forma de graficas dinámicas.

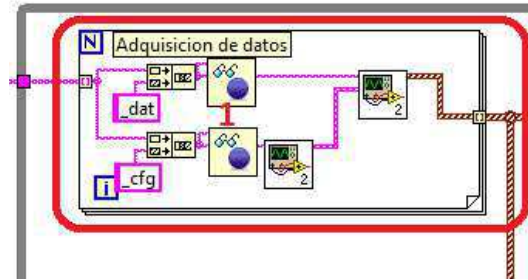


Figura 4.17: Visualizador 3

La función primera de esta sección es:

1. Leer las variables compartidas de acuerdo a las direcciones introducidas en el panel frontal.

Nota: este proceso lo realiza tantas veces como el tamaño del arreglo que contiene las direcciones.

Al final de este primer proceso se obtienen el frame de datos y el frame de configuración publicado por el programa de “Adquisición”.

4.4.4. Decodificación de frame de configuración.

El proceso de decodificación del frame de configuración se concentra en un subVi como se muestra en la figura 4.18



Figura 4.18: Visualizador 4

Este subVI: Recibe como entrada el frame de configuración.

Entrega a la salida un clúster con los datos relevantes de la configuración (descritos en el capítulo 3.1.3) para la decodificación de los valores fasoriales tales como:

- IDCODE (código de identificación del PMU)
- TIME_BASE (base de tiempo)
- NUM_PMU (número de PMUs)
 - STN (nombre de la estación)
 - IDCODE (código de identificación de la estación)
 - FORMAT (formato de los valores fasoriales y análogos)
 - PHNMR (número de fasores)
 - ANNMR (número de análogos)
 - DGNMR (número de digitales)
 - CHNAME PHASOR (nombres de canal correspondientes a los fasores)
 - CHNAME ANALOG (nombres de canal correspondientes a los analógicos)
 - CHNAME DIGITALES (nombres de canal correspondientes a los digitales)
 - PHUNIT (factor de conversión para fasores y análogos)
 - PHASOR_V_OR_I (unidades en que se presentan los fasores)
 - FNOM (frecuencia nominal de operación del PMU)
 - CFGCNT (contador de cambios en la configuración)
 - DATA_RATE (flujo de datos)
- CHK (resto de la verificación CRC)

Para comprender su funcionamiento el subVI, ahora se presenta en forma de VI y se tiene:

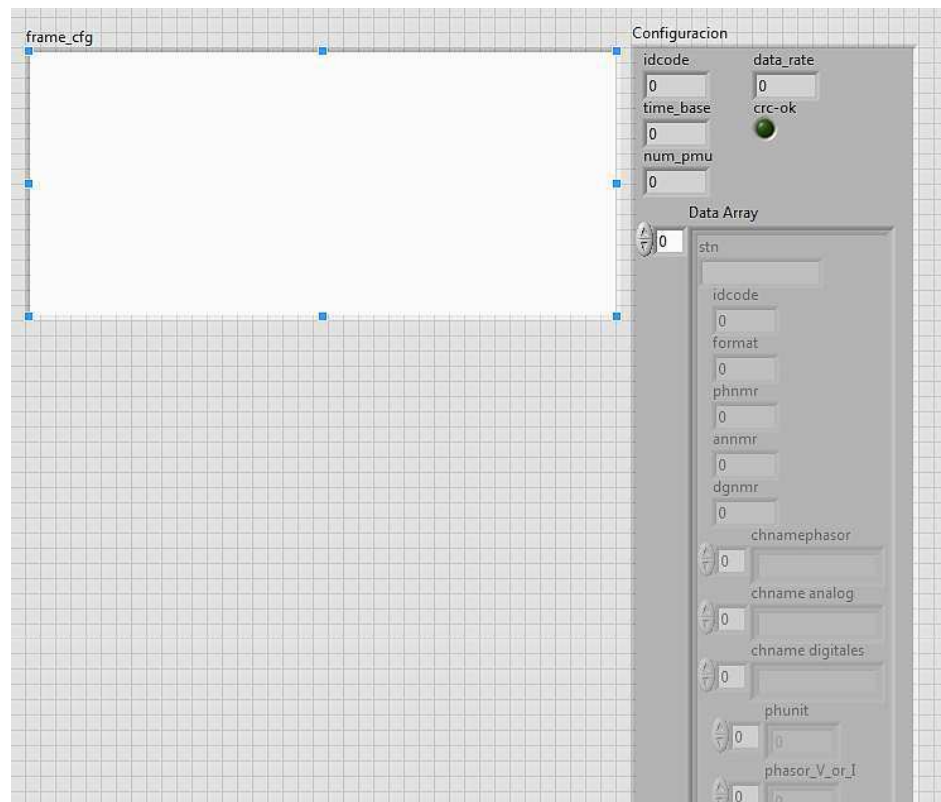


Figura 4.19: Decodificador del frame de configuración 2 (Panel frontal)

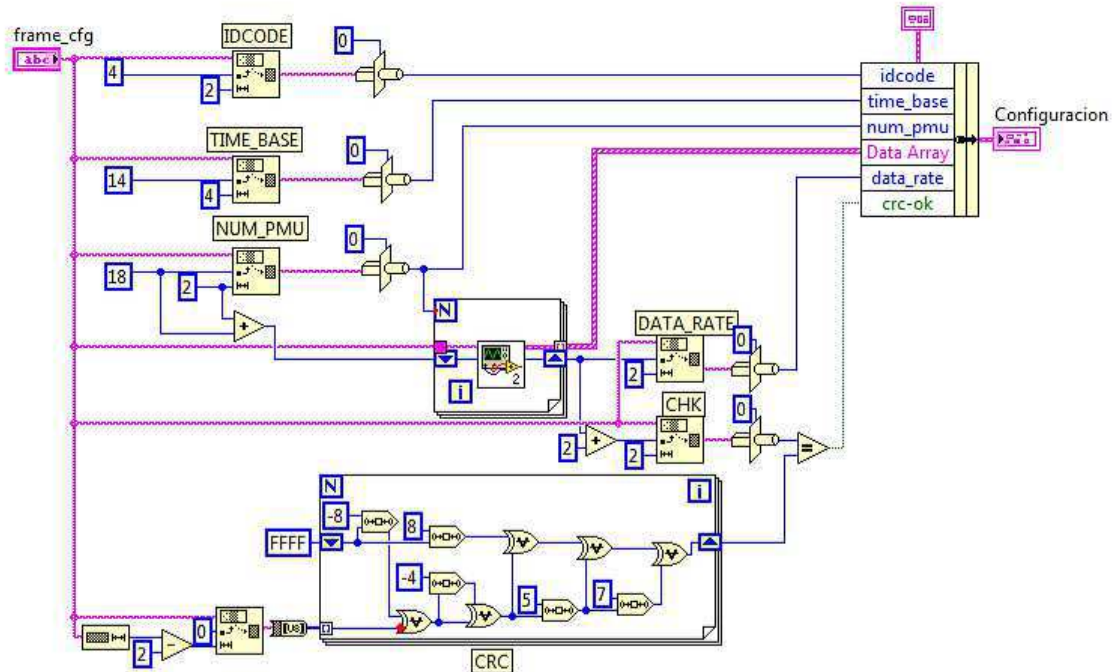


Figura 4.20: Decodificador del frame de configuración 2 (Diagrama de bloques)

Extracción de configuración general del PMU

Para su procesamiento, el frame de configuración 2 se puede dividir en dos bloques, los valores generales que describen al PMU y los valores particulares de cada estación de PMU. Los valores generales del PMU son:

1. IDCODE (código de identificación del PMU)
2. TIME_BASE (base de tiempo)
3. NUM_PMU (número de PMUs)
4. DATA_RATE (flujo de datos)
5. CRC-OK (resultado de la verificación CRC)

La forma adquirir estos valores consiste en extraer del frame de configuración, los bits correspondientes a cada parámetro y transformarlos de acuerdo a lo establecido en las especificaciones del frame según la norma IEE C37.118.2-2011 descrita en el capítulo 3.1.3

En el caso de los parámetros de IDCODE, TIME_BASE, NUM_PMU y DATA_RATE esta transformación consiste en pasar el número de hexadecimal al

tipo de número establecido en las especificaciones del frame. En labview esto se logra mediante un nodo denominado TypeCast, el cual solo requiere el dato de entrada y una muestra del dato al que se desea convertir.

Para la verificación del CRC se necesita extraer de la trama los últimos 2 bytes, estos valores son los correspondientes al CHK calculado en el equipo emisor; el siguiente paso consiste en aplicar el CRC a la trama recibida, para obtener el CHK calculado en el receptor; finalmente se comparan ambos CHK y si son iguales se tiene que la recepción de la trama fue correcta.

En forma programada, este proceso se implementó como se muestra en la figura 4.21

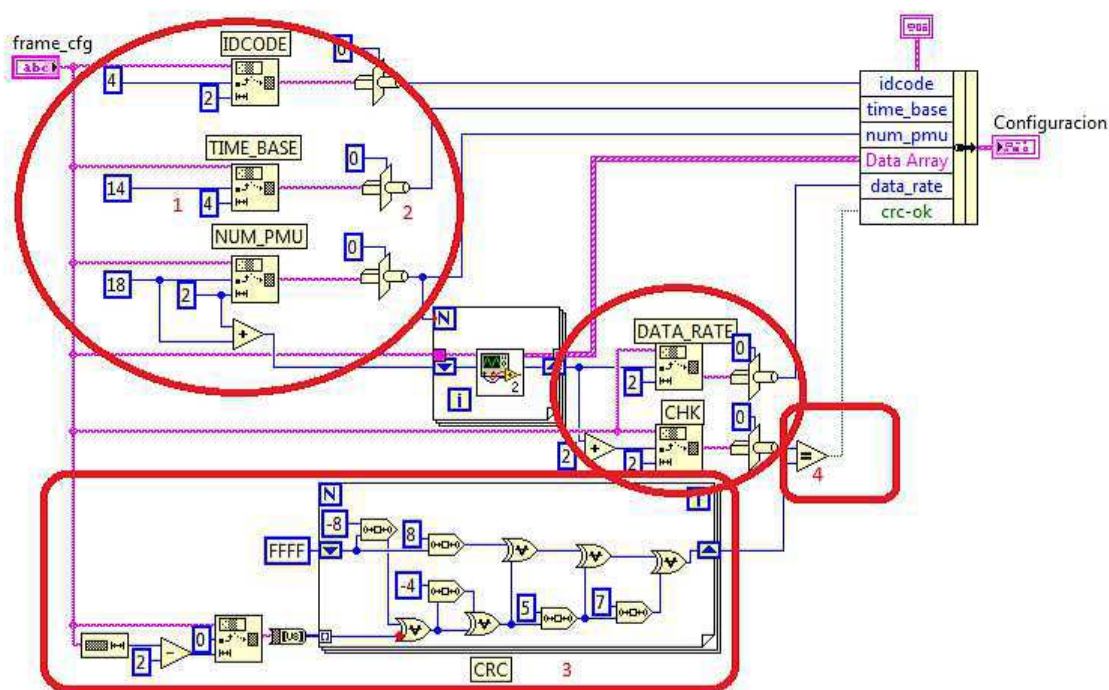


Figura 4.21: Visualizador 5

La función de esta sección es:

1. Extraer de la trama los bytes correspondientes a cada parámetro
2. Transformar el código hexadecimal a su correspondiente formato de número
3. Calcular el CHK del equipo receptor

4. Comparar el CHK del equipo receptor y el emisor

Al término de este proceso se tienen:

- Los valores de IDCODE, TIME_BASE, NUM_PMU y DATA_RATE (cables azules)
- Si la trama se recibió exitosamente (valor booleano) (cable verde)

Extracción de configuración particular de cada Estación de PMU

Los valores particulares de cada estación son:

1. STN (nombre de la estación)
2. IDCODE (código de identificación de la estación)
3. FORMAT (formato de los valores fasoriales y análogos)
4. PHNMR (número de fasores)
5. ANNMR (número de análogos)
6. DGNMR (número de digitales)
7. CHNAME PHASOR (nombres de canal correspondientes a los fasores)
8. CHNAME ANALOG (nombres de canal correspondientes a los analógicos)
9. CHNAME DIGITALES (nombres de canal correspondientes a los digitales)
10. PHUNIT (factor de conversión para fasores y análogos)
11. PHASOR_V_OR_I (unidades en que se presentan los fasores)
12. FNOM (frecuencia nominal de operación del PMU)
13. CFGCNT (contador de cambios en la configuración)

La extracción de estos valores se realiza tantas veces como número de estaciones de PMU existan, por lo que para su procesamiento se creó un subVI el cual decodifica de manera particular cada estación y se ejecuta tantas veces como lo indique el parámetro de NUM_PMU (número de PMUs).

En forma programada, este proceso se concentra en un subVI como lo muestra la figura: 4.22

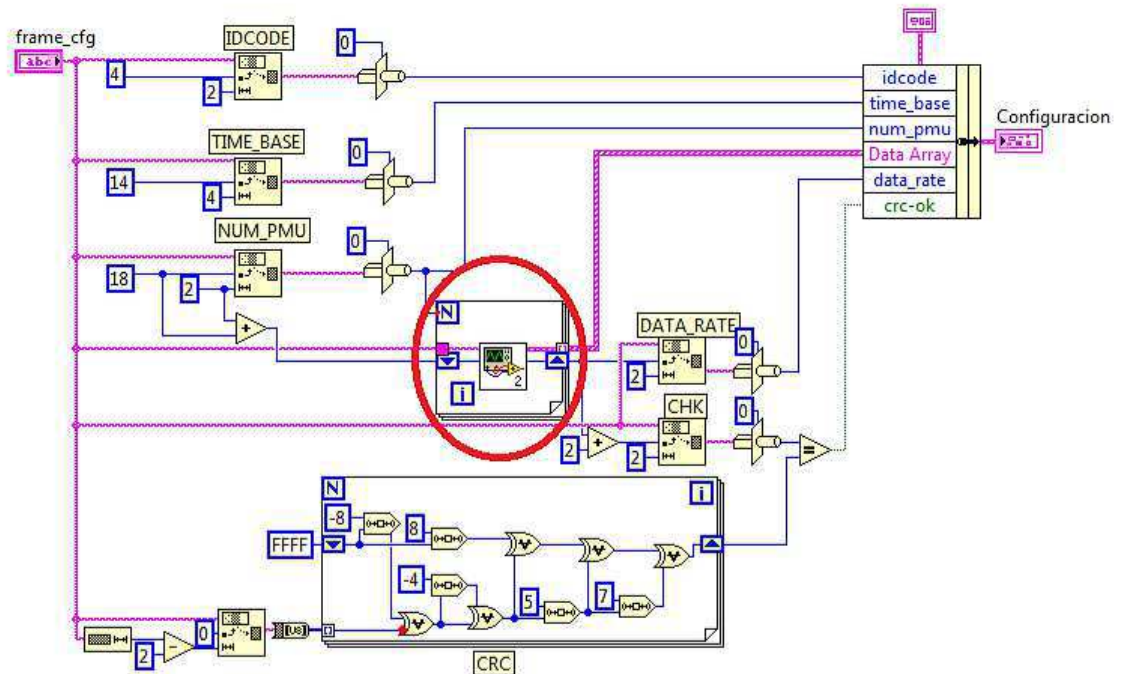


Figura 4.22: Visualizador 6

Este subVI:

Recibe como datos de entrada:

- La trama del frame de configuración 2 (cable rosa)
- El número de PMUs del frame (cable azul conectado al número de repeticiones del ciclo FOR)
- El índice que indica la posición del frame donde se comenzaran a decodificar los datos de la estación (cable azul).

Entrega como datos de salida:

- Un arreglo de clúster con los datos de cada estación (cable rosa con puntos blancos).
- El índice que indica la posición del frame donde se realizara el siguiente proceso (cable azul)

Para comprender su funcionamiento el subVI, ahora se presenta en forma de VI y se tiene:

The image shows the front panel of a LabVIEW subVI titled "Dec. Conf. Particular". The panel is organized into three main sections: "data_cfg", "data_inter", and "Index1".

- data_cfg:** Contains a numeric input field labeled "index 0" with the value "0".
- data_inter:** Contains several input fields and controls:
 - A numeric input field labeled "stn" with the value "0".
 - A numeric input field labeled "idcode" with the value "0".
 - A numeric input field labeled "format" with the value "0".
 - A numeric input field labeled "phnmr" with the value "0".
 - A numeric input field labeled "annmr" with the value "0".
 - A numeric input field labeled "dgnmr" with the value "0".
 - A numeric input field labeled "chnamephasor" with the value "0".
 - A numeric input field labeled "chname analog" with the value "0".
 - A numeric input field labeled "chname digitales" with the value "0".
 - A numeric input field labeled "phunit" with the value "0".
 - A numeric input field labeled "phasor_V_or_I" with the value "0".
 - A numeric input field labeled "fnom" with the value "0".
 - A numeric input field labeled "cfgcnt" with the value "0".
- Index1:** Contains a numeric input field labeled "Index1" with the value "0".

Figura 4.23: Dec. Conf. Particular (Panel frontal)

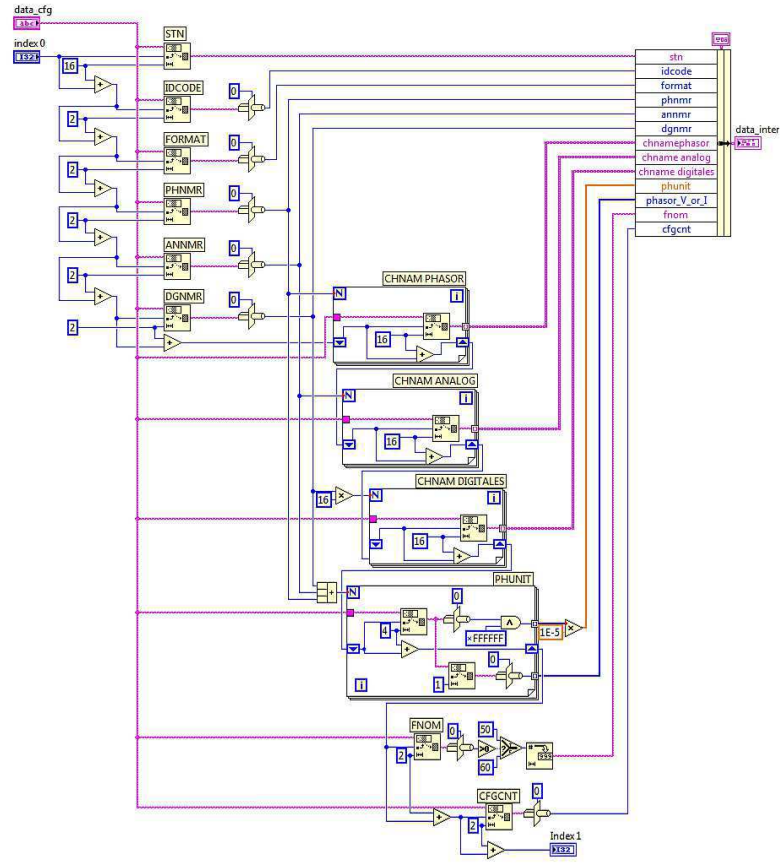


Figura 4.24: Dec. Conf. Particular (Diagrama de bloques)

A diferencia de los valores generales, los valores particulares de cada estación del PMU consisten en números de distinto formato, nombres y constantes que pueden depender de solo unos Bits y no de todo el Byte.

También, la posición de cada parámetro en la cadena depende de la posición y la extensión en bytes del parámetro anterior.

Para realizar toda esta decodificación:

- En los valores numéricos se procede nuevamente mediante un TypeCast
- En los valores que contienen nombres solo se corta la trama pues el cambio

de hexadecimal a ASCII en LabView es transparente (solo hay que cambiar el formato de presentación en el panel frontal)

- En los valores donde la información se encuentra solo en algunos bits de toda la cadena de Bytes:
 - Si son los bits menos significativos se procede realizando un Type-Cast y posteriormente aplicando una operación AND con el valor equivalente a los bites de lectura (dejando en 0 los valores que no se deberán tomar en cuenta),
 - si son los bits más significativos se realiza un segundo corte a la cadena indicando el tamaño de bytes que se tomaran.
 - Si es un bite con solo dos opciones se puede realizar una comparación lógica y mediante una llave de selección, y se asigna un valor al parámetro de acuerdo a la comparación.

En forma programada, este proceso se implementó como se muestra en la figura 4.25

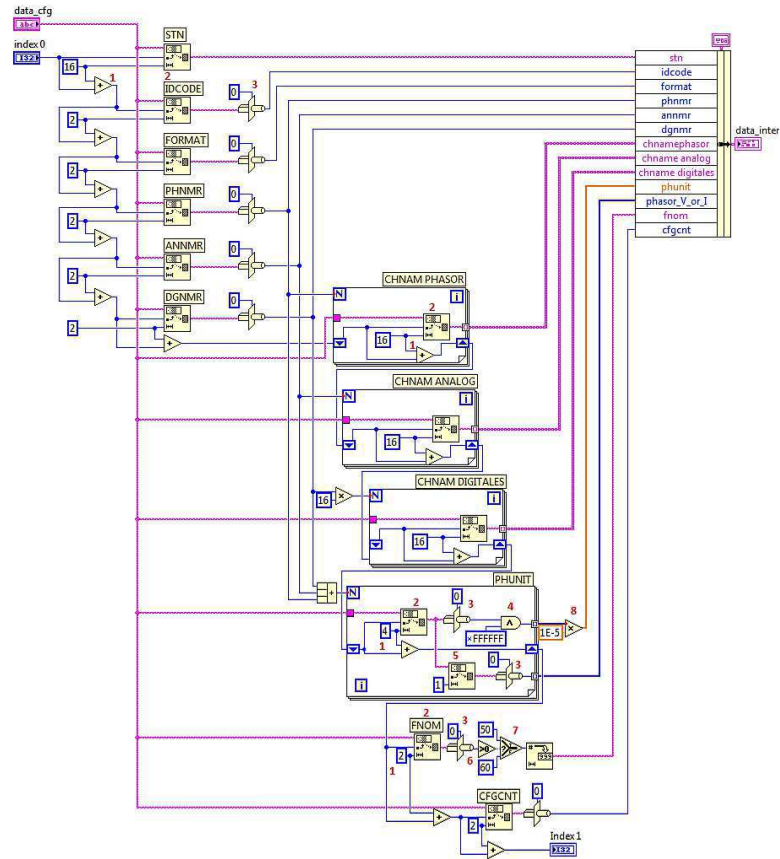


Figura 4.25: Visualizador 7

Las funciones de esta sección son:

1. Calcular el índice del siguiente parámetro, sumando al índice del parámetro actual la longitud en bytes del mismo.
2. Seccionar del frame de configuración los valores correspondientes al parámetro actual.
3. Realizar el cambio de formato mediante el Type Cast

4. Seccionar de los valores del parámetro, los bits necesarios para la conversión mediante una operación AND
5. Seccionar de los valores del parámetro, los bits necesarios para la conversión
6. Realizar una comparación lógica cuando parámetro tiene solo dos opciones de valor
7. Asignar el valor al parámetro de acuerdo al resultado de la operación lógica
8. Multiplicar el valor obtenido por una constante de acuerdo a las especificaciones del “frame de configuración 2”

Al término de este proceso se tienen los valores particulares de una estación del PMU agrupados en un clúster, este parámetro se pasa al vi de decodificación del “frame de configuración 2” que lo ejecuta tantas veces como número de PMUs y todos resultados obtenidos se agrupan en un arreglo llamado “Data Array”.

4.4.5. Procesamiento de frame de datos

El procesamiento del frame de datos se concentra en un subVi como se muestra en la figura:

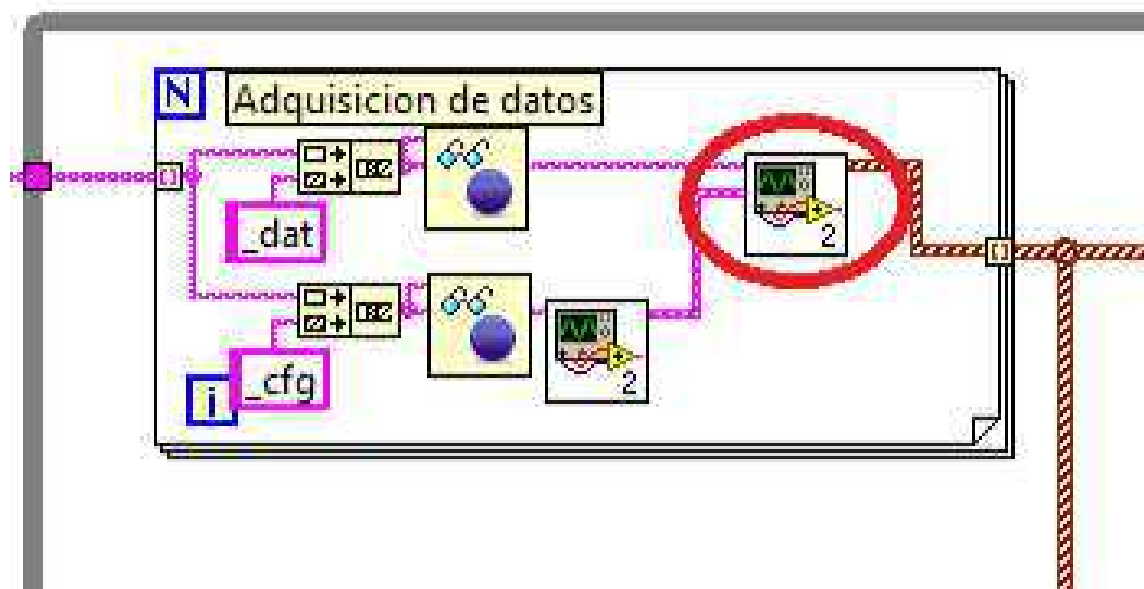


Figura 4.26: Visualizador 8

Este subVI:

- Recibe como entrada el frame de datos y el clúster con los datos relevantes del frame de configuración.
- Entrega a la salida un arreglo de Waveform con los valores fasoriales decodificados del frame de datos (fecha/hora, frecuencias, magnitudes y ángulos) y su respectiva estampa de tiempo (tiempo del PMU).

Para comprender su funcionamiento el subVI, ahora se presenta en forma de VI y se tiene:

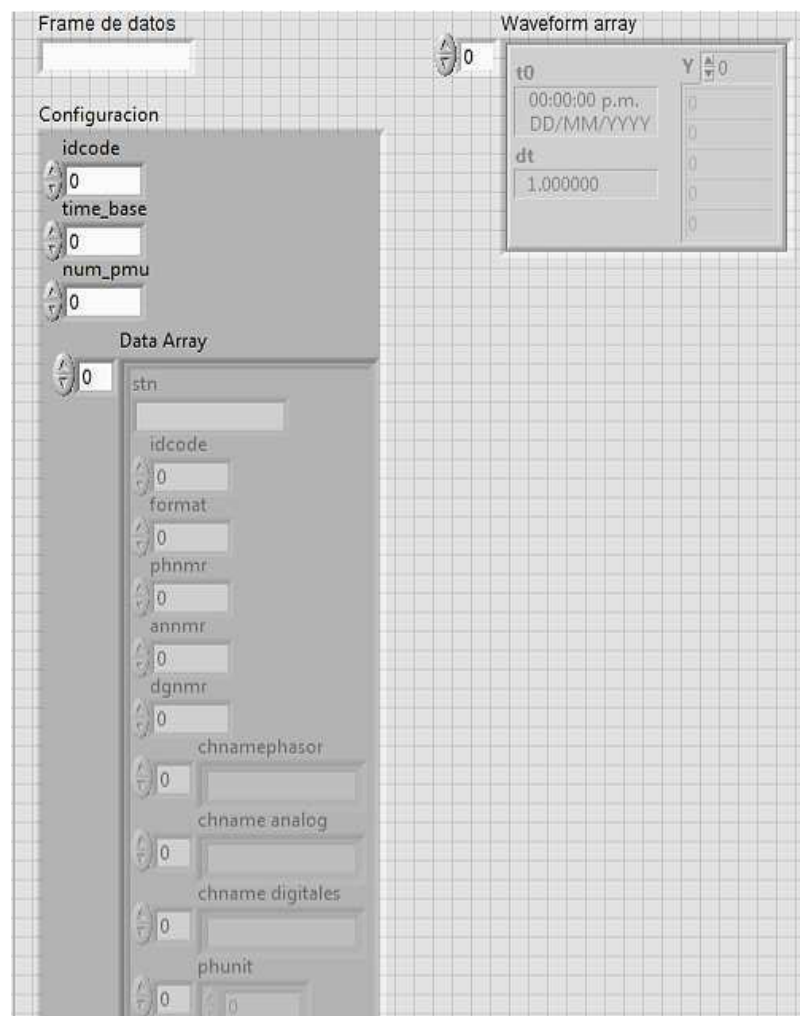


Figura 4.27: Visualizador 8 (Panel frontal)

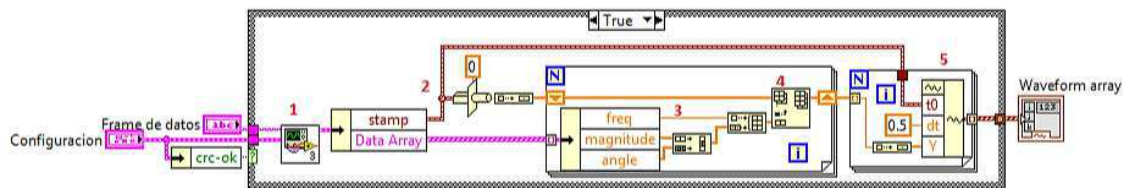


Figura 4.28: Visualizador 8 (Diagrama de bloques)

(Este VI es solo una capa auxiliar que se encarga de organizar los datos en un arreglo.)

Las funciones de esta sección son:

1. Decodificar el frame de datos con base en los datos del frame de configuración 2
2. Extraer la fecha y hora
3. Extraer y organizar los valores fasoriales de cada PMU (no incluye valores analógicos, ni digitales)
4. Ordena los valores de los Fasores en el siguiente orden:
 - Fecha/hora (general del PMU)
 - Frecuencia (particular de la estación 0)
 - Magnitud (particular del Fasor 00)
 - Angulo (particular del Fasor 00)
 - Magnitud (particular del Fasor 0X)
 - Angulo (particular del Fasor 0X)
 - Frecuencia (particular de la estación N)

- Magnitud (particular del Fasor N0)
 - Angulo (particular del Fasor N0)
 - Magnitud (particular del Fasor NY)
 - Angulo (particular del Fasor NY)
5. Convierte los valores de tipo double a waveform chart (formato para graficas de labview)

4.4.6. Decodificación de frame de datos

En el apartado anterior la decodificación del frame de datos se mostró como un subVI, en forma más detallada se muestra su estructura como un VI donde se tiene:

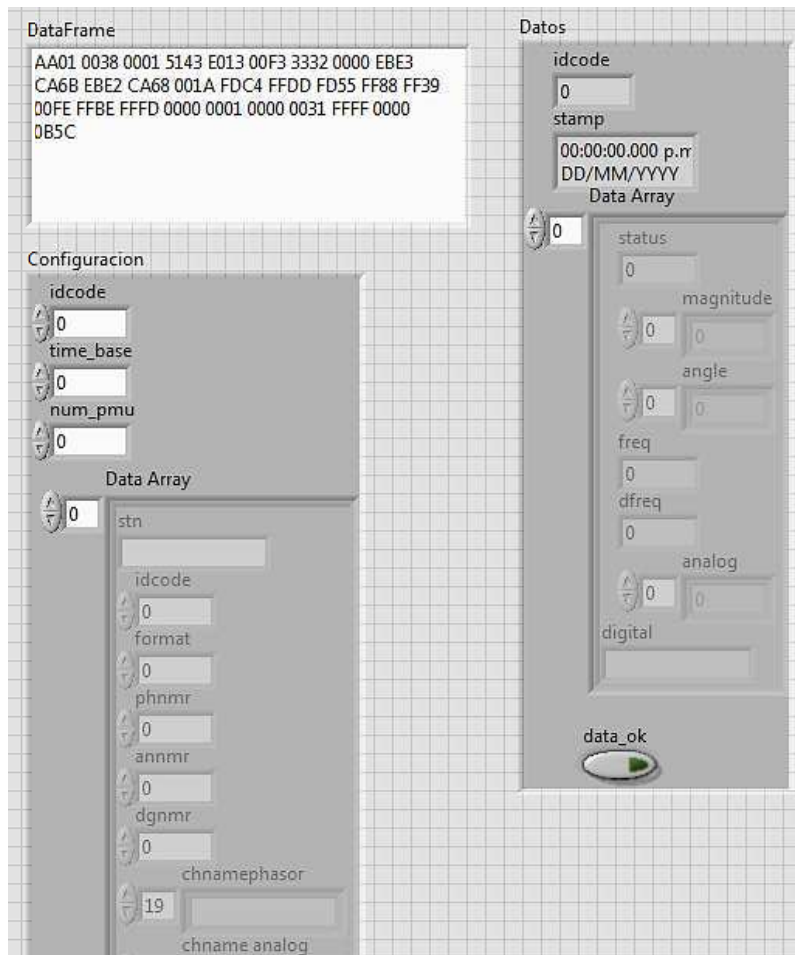


Figura 4.29: Decodificador del frame de datos (Panel Frontal)

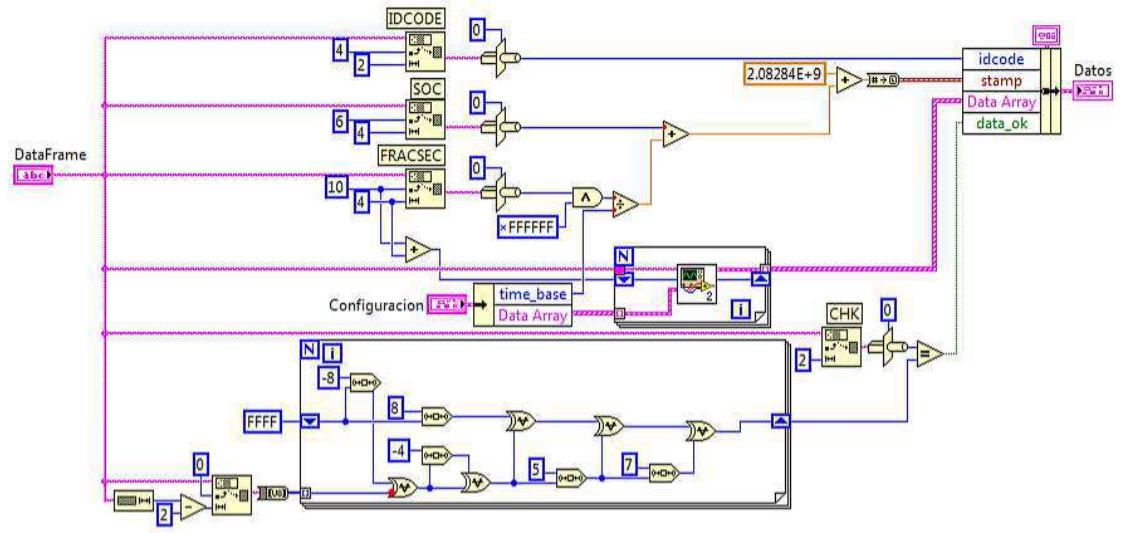


Figura 4.30: Decodificador del frame de datos (Diagrama de bloques)

Este subVI:

- Recibe como entrada el frame de datos y el clúster con los datos relevantes del frame de configuración.
- Entrega a la salida un clúster con los datos del PMU, tales como:
 - IDCODE (código de identificación del PMU)
 - STAMP (estampa de tiempo del PMU)
 - STAT (estado de la medición del PMU)
 - PHASORS (Valores de los Fasores del PMU)
 - FREQ (frecuencia del PMU)
 - DFREQ (diferencial de frecuencia del PMU)
 - ANALOG (valores analógicos externos del PMU)
 - DIGITAL (valores digitales del PMU)
 - DATA.OK (resultado de la verificación CRC)

Descritos en el capítulo 3.1.4

Extracción de datos generales del PMU

De forma similar al frame de configuración 2, los datos del PMU se pueden dividir en dos bloques, los valores generales del PMU y los valores particulares de cada estación de PMU.

Los valores generales del PMU son:

- IDCODE (código de identificación del PMU)
- SOC (parte entera de los segundos de la fecha y hora de la medición)
- FRACSEC (fracción de segundo de la hora y fecha de la medición)
- CHK (resto de la verificación CRC)

La forma adquirir estos valores consiste en extraer del frame de configuración, los bits correspondientes a cada parámetro y transformarlos de acuerdo a lo establecido en las especificaciones del frame de datos (que incluye relaciones con los datos del frame de configuración 2).

- En el caso de IDCODE se procede a realizar una conversión con un TypeCast
- Para los parámetros de SOC y FRACSEC se realiza también un TypeCast y posteriormente se realizan las operaciones definidas en el capítulo 3.1.1 las cuales incluyen también al valor de TIME_BASE del frame de configuración 2, para así formar la estampa de tiempo.
- Para la verificación del CRC se procede de la misma manera que en el procesamiento del frame de configuración 2 (capítulo 4.4.4).

En forma programada, este proceso se implementó como se muestra en la figura 4.31

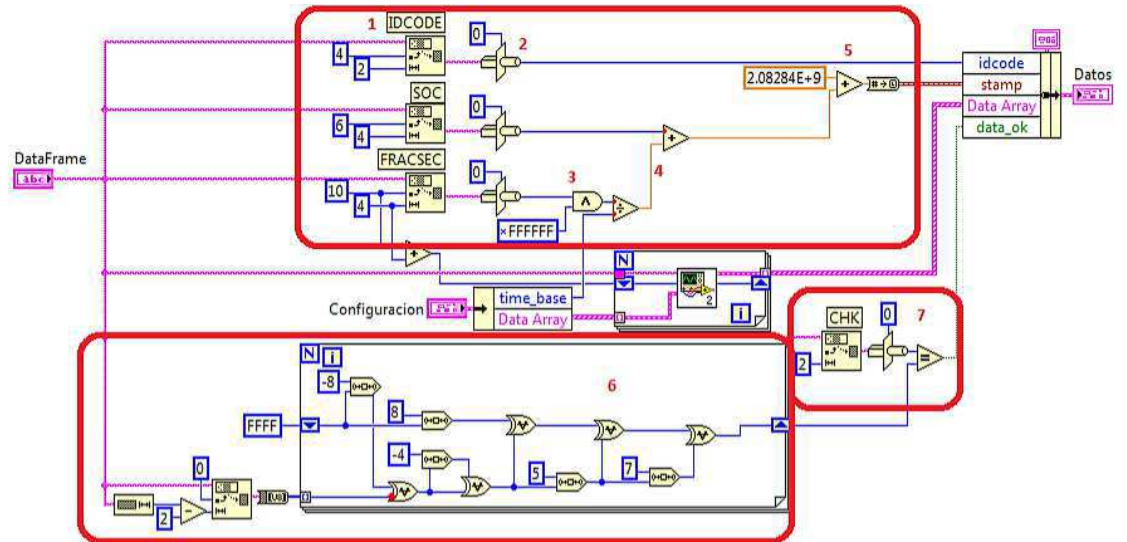


Figura 4.31: Visualizador 9

La función de esta sección es:

1. Extraer de la trama los bytes correspondientes a cada parámetro
2. Transformar el código hexadecimal mediante un TypeCast a su correspondiente formato de número
3. Extraer los bits necesarios del byte de datos para la siguiente operación
4. Realizar el cálculo de tiempo de acuerdo al Capítulo 3.1.1
5. Convertir el tiempo de base unix a base de tiempo de labview (investiga cual es)
6. Calcular el CHK del equipo receptor
7. Comparar el CHK del equipo receptor y el emisor

Al término de este proceso se tienen:

- El valor de IDCODE (cables azules)
- La estampa de tiempo del PMU STAMP (cable café)
- Si la trama se recibió exitosamente (cable verde)

Extracción de datos particulares de cada Estación de PMU

Los datos particulares de cada estación son:

- STAT (estado de la medición del PMU)
- PHASORS (Valores de los Fasores del PMU)
- FREQ (frecuencia del PMU)
- DFREQ (diferencial de frecuencia del PMU)
- ANALOG (valores analógicos externos del PMU)
- DIGITAL (valores digitales del PMU)

La extracción de estos datos se realiza tantas veces como número de PMUs existan, por lo que para su procesamiento se creó un subVI el cual decodifica de manera particular esta sección y se ejecuta tantas veces como el tamaño del arreglo de datos (Que a su vez indica el número de PMUs).

En forma programada, este proceso se concentra en un subVI como lo muestra la figura 4.32

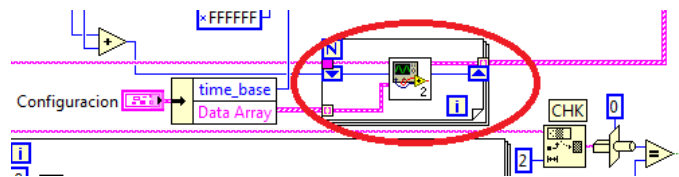


Figura 4.32: Visualizador 10

Este subVI:

Recibe como datos de entrada la trama del frame de datos (cable rosa), el índice que indica la posición en dicho frame donde se comenzaran a decodificar los datos de la estación (cable azul) y el clúster con los datos relevantes del frame de configuración 2 (cable rosa con puntos blancos).

Entrega como datos de salida un arreglo de clúster con los datos de cada estación (cable rosa con puntos blancos), y el índice de la posición en el frame de datos donde se realizara el siguiente proceso (cable azul).

Para comprender su funcionamiento el subVI, ahora se presenta en forma de VI y se tiene:

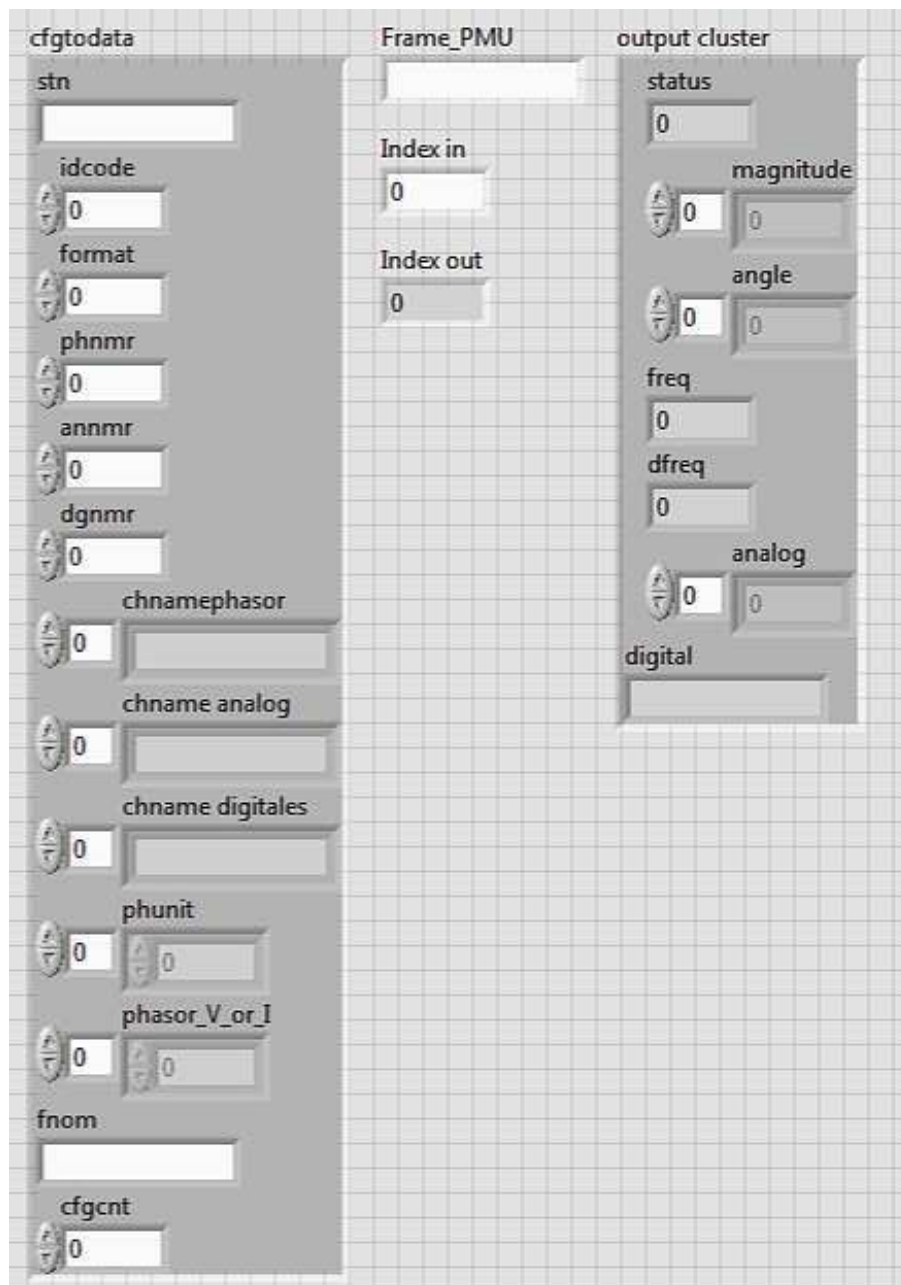


Figura 4.33: Visualizador 10 (Panel frontal)

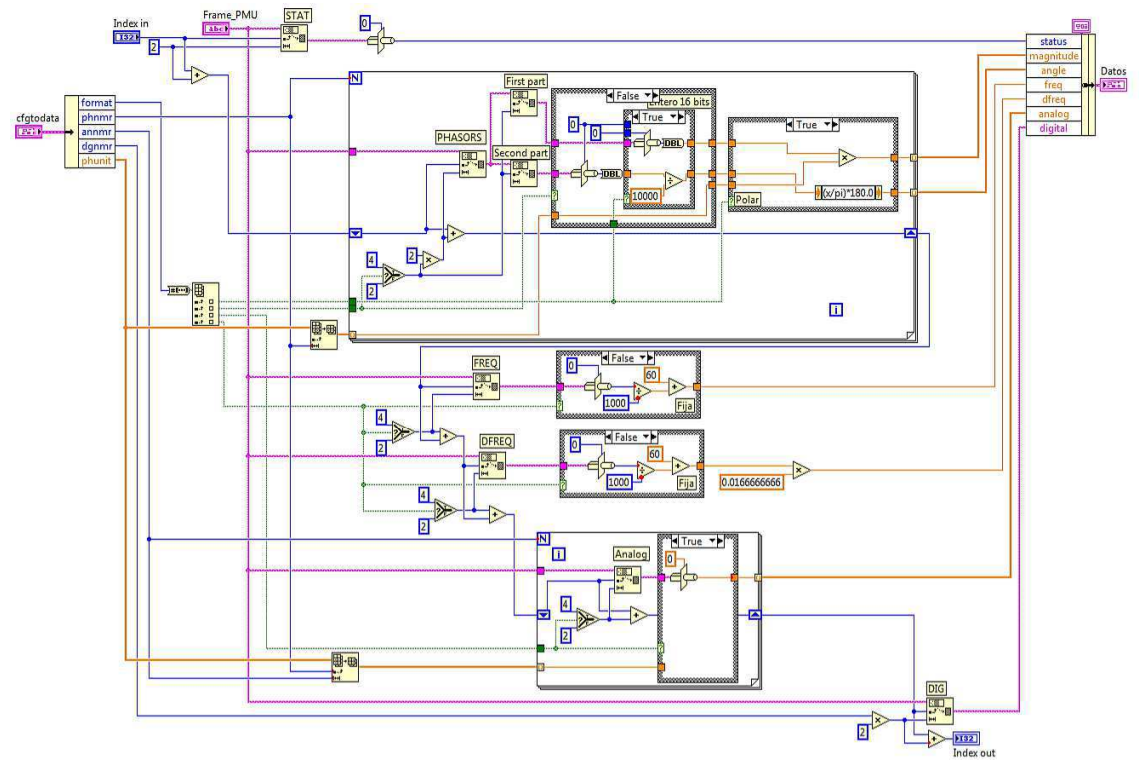


Figura 4.34: Visualizador 10 (Diagrama de bloques)

El funcionamiento de este vi se basan las especificaciones de la norma IEE C37.118.2-2011 descrita en el capítulo 3.1.4 y abarca todas sus posibles variantes.

Para detallar las opciones de cada parámetro se analizarán de forma individual

STATUS

El parámetro de status indica el estado y calidad de la medición mediante banderas digitales, su tamaño es de 2 bytes y para su utilización se le dio como formato un número entero sin signo de 16 bits. En forma programada, este proceso se implementó como se muestra en la figura 4.35.

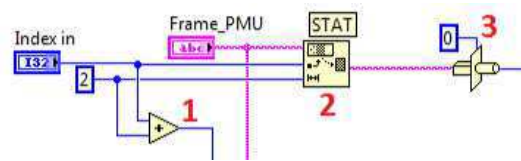


Figura 4.35: Visualizador 11

Las funciones de esta sección son:

1. Calcular el índice del siguiente parámetro, sumando al índice del parámetro actual la longitud en bytes del mismo.
2. Seccionar del frame de datos los bytes correspondientes al parámetro actual, introduciendo el índice donde se encuentra y su longitud.
3. Realizar el cambio de formato mediante un Type Cast indicando que el formato de número a la salida será: entero de 16bits sin signo

PHASORS

Según la norma IEE C37.118.2-2011 cada fasor está compuesto por dos valores como se especifica en el cuadro 3.7 y estos constituyen la parte central del programa. Así mismo para estos dos valores existen dos formas de representación: rectangular y polar; y dos tipos de formato de número: entero de 16 bits y punto flotante de 32bits.

Según lo expuesto en el cuadro 3.7:

De acuerdo a su representación, en cuestión del orden:

- Si se trata de representación rectangular:
la parte real va primero seguida de la parte imaginaria.
- Si se trata de representación polar:
la magnitud va primero seguida del ángulo.

De acuerdo a su formato, en cuestión de conversiones

- Si se trata de punto flotante de 32 bits no se requieren más conversiones.
- Si se trata de entero de 16 bits, el valor resultante se tiene que multiplicar por una constante.
 - Para la representación rectangular:
 - Tanto la parte real como la imaginaria tienen un formato de 16 bits CON SIGNO, y posteriormente se debe multiplicar este valor por el parámetro PHUNIT del frame de configuración 2
 - Para la representación polar:
 - La magnitud tiene un formato de 16 bits SIN SIGNO y posteriormente se debe multiplicar este valor por el parámetro PHUNIT del frame de configuración 2.
 - El ángulo tiene un formato 16 bits CON SIGNO y posteriormente se debe dividir este valor entre 1000. Los valores fasoriales pueden tener solo una de las dos opciones y el parámetro “FORMAT” del “frame de configuración 2” es el que nos indica con cual se trabaja.

Todos estos procesos se implementaron como se muestra en la figura 4.36

7. Seleccionar de la sub trama los bytes correspondientes al segundo valor fasorial.
8. Procesar los valores de acuerdo a su formato numérico.
9. Procesar los valores de acuerdo a su representación fasorial.

Al final de este proceso se tienen 2 arreglos con los valores de las magnitudes y ángulos de los valores fasoriales

En forma más detallada se muestran a continuación las cuatro posibles combinaciones de formato y representación.

Caso 1: formato de punto flotante de 32 bits y representación polar.

En forma programada:

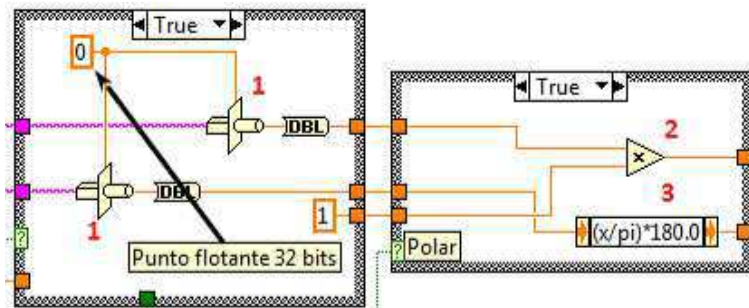


Figura 4.37: Visualizador 12.1

Las funciones de esta sección son:

1. Realizar la conversión de los dos valores mediante un Type Cast indicando que el formato de número a la salida será: punto flotante de 32bits
2. Multiplicar la magnitud por su constante correspondiente (en este caso al no requerir conversión es 1).
3. Convertir el ángulo de radianes a grados.

Caso 2: formato de punto flotante de 32 bits y representación rectangular.

En forma programada:

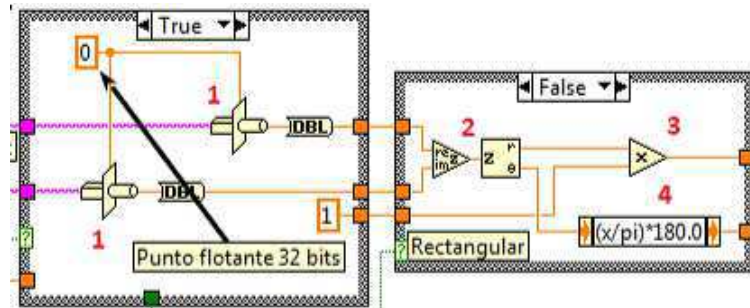


Figura 4.38: Visualizador 12.2

Las funciones de esta sección son:

1. Realizar la conversión de los dos valores mediante un Type Cast indicando que el formato de número a la salida será: punto flotante de 32bits.
2. Convertir de representación rectangular a polar.
3. Multiplicar la magnitud por su constante correspondiente (en este caso al no requerir conversión es 1).
4. Convertir el ángulo de radianes a grados.

Caso 3: formato entero de 16 bits y representación rectangular.

En forma programada:

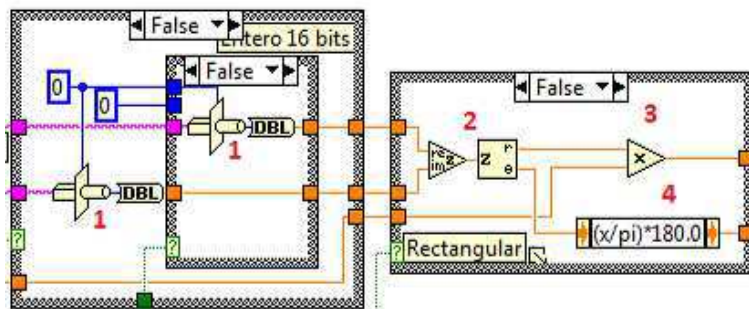


Figura 4.39: Visualizador 12.3

Las funciones de esta sección son:

1. Realizar la conversión de los dos valores mediante un Type Cast indicando que el formato de número a la salida será: entero de 16 bits CON SIGNO
2. Convertir de representación rectangular a polar
3. Multiplicar la magnitud por su constante correspondiente (en este caso proviene de PHUNIT).
4. Convertir el ángulo de radianes a grados.

Caso 4: formato entero de 16 bits y representación polar.

En forma programada:

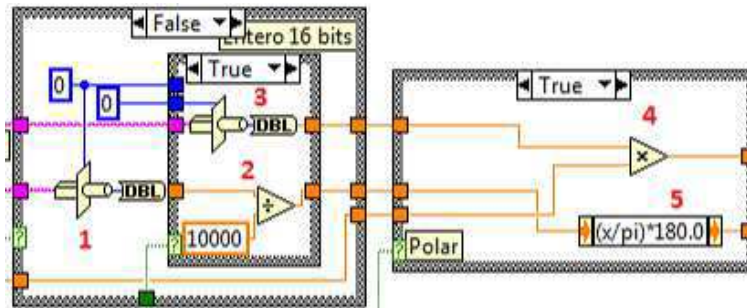


Figura 4.40: Visualizador 12.4

Las funciones de esta sección son:

1. Realizar la conversión del valor correspondiente al ángulo mediante un Type Cast indicando que el formato de número a la salida será: entero de 16 bits CON SIGNO
2. Dividir el valor del ángulo entre 1000
3. Realizar la conversión del valor correspondiente a la magnitud mediante un Type Cast indicando que el formato de número a la salida será: entero de 16 bits SIN SIGNO
4. Multiplicar la magnitud por su constante correspondiente (en este caso proviene de PHUNIT).
5. Convertir el ángulo de radianes a grados.

FREQ y DFREQ

Los parámetros de frecuencia y diferencial de frecuencia reciben el mismo tratamiento y de forma similar a los valores fasoriales tienen dos tipos de formato numérico: Entero de 16 bits y punto flotante de 32 bits.

- Si el formato es punto flotante no se requieren más conversiones
- Si el formato es entero el resultado se dividirá entre 1000 y se le sumará a la frecuencia nominal (60 Hz).

En forma programada, este proceso se implementó como se muestra en la figura: 4.41

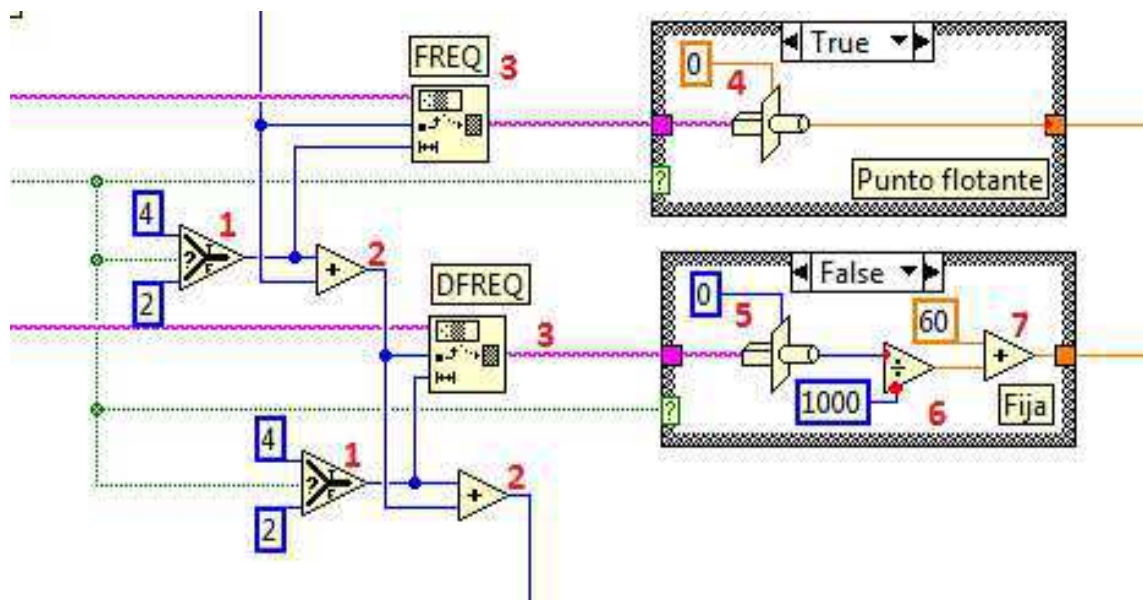


Figura 4.41: Visualizador 13

Las funciones de esta sección son:

1. Establecer la longitud de los valores de frecuencia y diferencial de frecuencia dentro de la trama (2 Bytes si su formato es entero de 16 Bits, 4 Bytes si es punto flotante de 32 Bits)
2. Calcular el índice del siguiente parámetro, sumando al índice del parámetro actual la longitud en bytes del parámetro actual.

3. Seccionar del frame de datos los bytes correspondientes al parámetro actual, introduciendo el índice donde se encuentra y su longitud.
4. Realizar el cambio de formato mediante un Type Cast indicando que el formato de número a la salida será: punto flotante de 32bits (en caso de que el formato así lo indique)
5. Realizar el cambio de formato mediante un Type Cast indicando que el formato de número a la salida será: entero de 16 bits CON SIGNO (en caso de que el formato así lo indique)
6. Dividir el valor entre 1000
7. Sumar el valor a la frecuencia nominal (60 Hz)

Nota: el procesamiento de FREC y DFREC dependen del mismo parámetro contenido en FORMAT del frame de configuración 2

ANALOG

Se tratan de mediciones de tipo analógicas que son útiles en el estudio de parámetros eléctricos, la aplicación de estas mediciones son definidas por el usuario y los parámetros que censan son variados, por ejemplo: potencia activa, potencia reactiva, temperatura, etc.

De forma similar a los valores fasoriales tienen dos tipos de formato numérico: Entero de 16 bits y punto flotante de 32 bits.

- Si el formato es punto flotante no se requieren más conversiones
- Si el formato es entero el resultado se dividirá entre una constante definida en el parámetro ANUNIT del “frame de configuración 2” (en esta aplicación se encuentra concatenado al final de phunit)

En forma programada, este proceso se implementó como se muestra en la figura: 4.42 y la figura 4.43

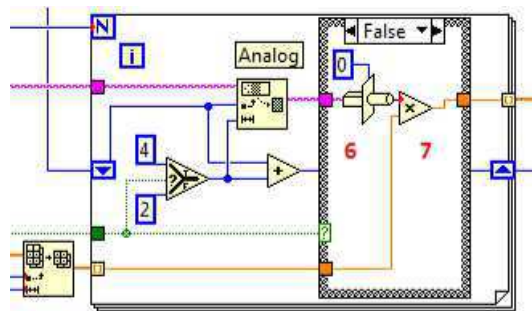


Figura 4.42: Visualizador 14

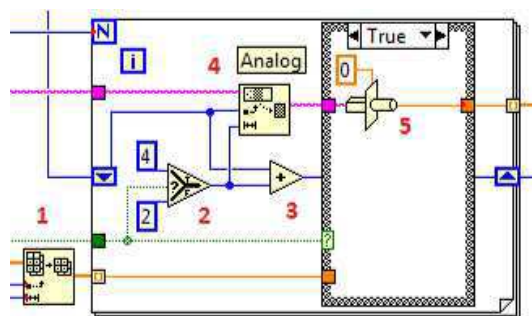


Figura 4.43: Visualizador 15

Las funciones de esta sección son:

1. Seleccionar del arreglo “phunit” las constantes por las que se multiplicara el valor si su formato es entero
2. Establecer la longitud de los valores analogos dentro de la trama (2 Bytes si su formato es entero de 16 Bits, 4 Bytes si es punto flotante de 32 Bits)
3. Calcular el índice del siguiente parámetro, sumando al índice del parámetro actual la longitud en bytes del parámetro actual.
4. Seccionar del frame de datos los bytes correspondientes al parámetro actual, introduciendo el índice donde se encuentra y su longitud.
5. Realizar el cambio de formato mediante un Type Cast indicando que el formato de número a la salida será: punto flotante de 32bits (en caso de que el formato así lo indique)
6. Realizar el cambio de formato mediante un Type Cast indicando que el formato de número a la salida será: entero de 16 bits CON SIGNO (en caso de que el formato así lo indique)
7. Multiplicar el valor por su respectiva constante

DIGITAL

Se trata de banderas digitales cuyo uso es definido por el usuario, pueden indicar el estado de interruptores habilitación de líneas, alertas etc.

En forma programada, este proceso se implementó como se muestra en la figura: 4.44

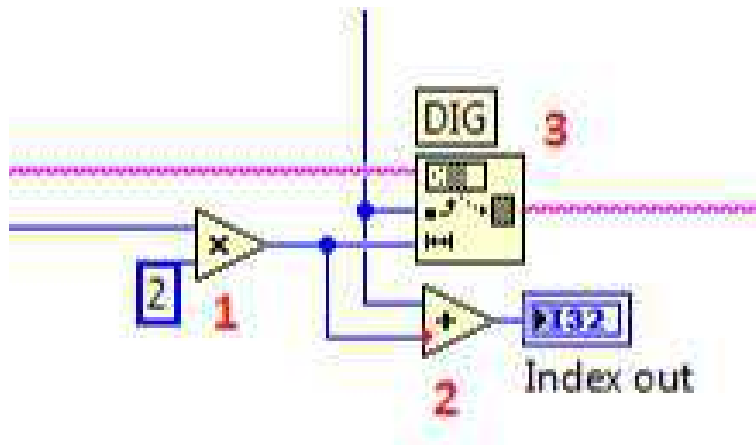


Figura 4.44: Visualizador 16

Las funciones de esta sección son:

1. Establecer la longitud de los valores digitales dentro de la trama (2 Bytes por cada valor digital)
2. Calcular el índice del siguiente parámetro, sumando al índice del parámetro actual la longitud en bytes del mismo.
3. Seccionar del frame de datos los bytes correspondientes al parámetro actual, introduciendo el índice donde se encuentra y su longitud.

Nota: Al término de este proceso no se realiza conversión alguna pues se desconoce el uso o aplicación de las banderas.

Al término de todos estos procesos se tienen los valores particulares de una estación del PMU agrupados en un clúster, este parámetro se pasa al vi de decodificación del “frame de datos” que lo ejecuta tantas veces como estaciones de PMUs y todos resultados obtenidos se pasaran al Vi superior que los transformara a un arreglo de waveforms.

Agrupación de valores

Regresando al programa central, el ciclo for que incluye la decodificación del frame de configuración y la decodificación de datos une los arreglos de valores de cada PMU en un arreglo de arreglos (arreglo de dimensión 2) como se muestra en la figura: 4.45

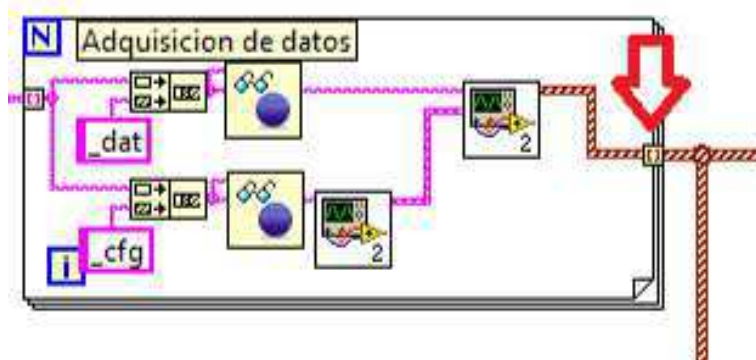


Figura 4.45: Visualizador 17

La estructura de este arreglo en forma de tabla es:
Dónde:

- n es el número de variables introducidas al programa (número de PMUs)
- m el número de fasores dentro del PMU (organizados de acuerdo a su orden de aparición dentro del PMU)

Como se mencionó en el capítulo 3.1.4, los valores de dichos fasores pueden ser únicamente frecuencias o tensiones y para conocer a cual hacen referencia, se puede revisar el byte más significativo del parámetro PHUNIT del frame de configuración o la variable “PHASOR_V_OR_I” del clúster “data inter” incrustado en el clúster “data configuración” de este programa.

Para esta aplicación en particular se sabe que todas las unidades de medición fasorial tienen el siguiente formato:

- Fasor 1: Voltaje de bus de fase a neutro de la subestación (expresada en volts)
- Fasor 2: Voltaje de bus de fase a neutro de la subestación (expresada en [V]) (respaldo)
- Fasor 3: Corriente de línea 1 (expresadas en Amperes)

	PMU 0	PMU 1	...	PMU n-1	PMU n
Fecha					
Frecuencia					
Magnitud 0					
Ángulo 0					
Magnitud 1					
Ángulo 1					
...					
...					
Magnitud m-2					
Ángulo m-2					
Magnitud m-1					
Ángulo m-1					

Cuadro 4.1: Formato General Adquisición

- Fasor 4: Corriente de línea 2 (expresadas en [A])
- Fasor x: corriente de línea x-2 (expresadas en [A])

Por lo tanto la tabla 4.4.6 para esta aplicación en particular será:

4.4.7. Tiempo y sincronización

En el estudio de los parámetros eléctricos cuya frecuencia es de 60 [Hz] (repetición del fenómeno 60 veces en cada segundo) el tiempo es un factor importante.

Los PMU en teoría están sincronizados por equipos auxiliares, pero en ocasiones debido a factores externos como la distancia (que retrasa unos segundos la recepción) o la ubicación (que no permite llegar la señal de tiempo al equipo) las mediciones se desfasan.

En este programa, previendo estos casos, se extraen los parámetros de tiempo de todos los PMU que se van a procesar y se comparan con el tiempo de un PMU seleccionado por el usuario, si al realizar la comparación del tiempo, este se encuentra en un rango aceptable (± 0.5 segundos según los operadores), se considera que se mantiene alineado, caso contrario su estampa de tiempo no se considera apta para su utilización.

En forma programada, este proceso se implementó como se muestra en las figuras 4.46 y 4.47

	PMU 0	PMU 1	...	PMU n-1	PMU n
Fecha					
Frecuencia					
Voltaje fn					
Ángulo V					
Voltaje2 fn					
Ángulo V2					
Corriente 1					
Ángulo I2					
...					
...					
Corriente m-1					
Ángulo I m-1					
Corriente m					
Ángulo I m					

Cuadro 4.2: Formato Especifico de Visualizador.vi

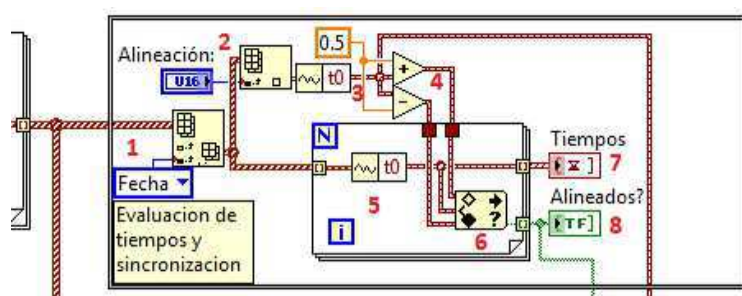


Figura 4.46: Visualizador 18

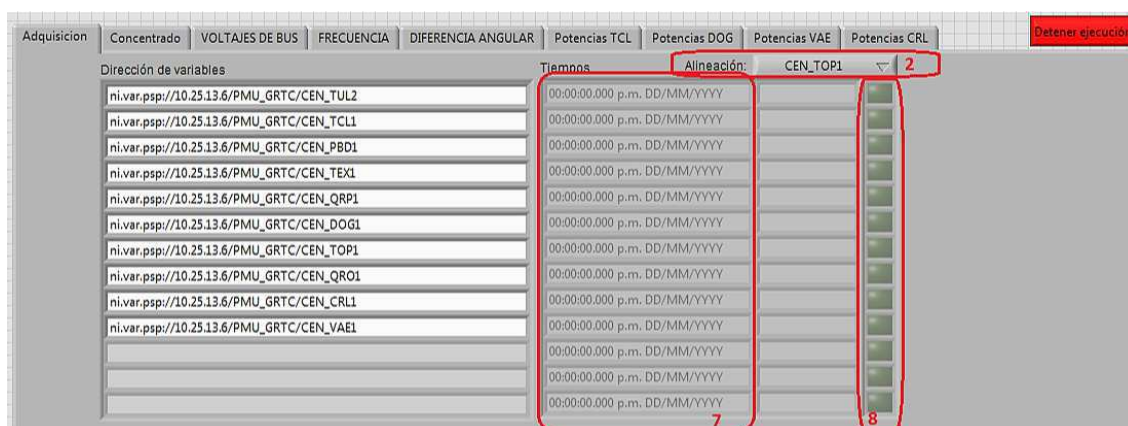


Figura 4.47: Visualizador 19

Las funciones de esta sección son:

1. Extraer del arreglo, el renglón correspondiente a los tiempos.
2. Seleccionar mediante el panel frontal la subestación cuyo tiempo se tomara como referencia.
3. Extraer la estampa de tiempo de referencia
4. Calcular el rango superior e inferior de la comparación
5. Extraer las estampas de tiempo de todos los PMU que se procesaran
6. Determinar si la estampa de tiempo se encuentra en el rango
7. Mostrar el tiempo de todos los PMU
8. Mostrar el resultado de la comparación

Al término de este proceso se tiene el tiempo de referencia (cable café), y la alineación de los tiempos (cable verde)

4.4.8. Procesamiento de Waveform

Como se explicó con anterioridad, los valores de los fasores son tensiones de fase a neutro y corrientes de línea expresadas en unidades de volts [V] y amperes [A]. De forma práctica para su estudio y visualización, la tensión de fase a neutro se transforma a tensión de fase a fase y se expresa en unidades de [KV], mientras las corrientes se transforman a potencia activa [MW] y reactiva [MVAR]. En cuestión de tiempos, aquellas estampas que no están sincronizadas son extraídas y reemplazadas por el tiempo de referencia. En forma programada, este proceso se implementó como se muestra en la figura 4.48

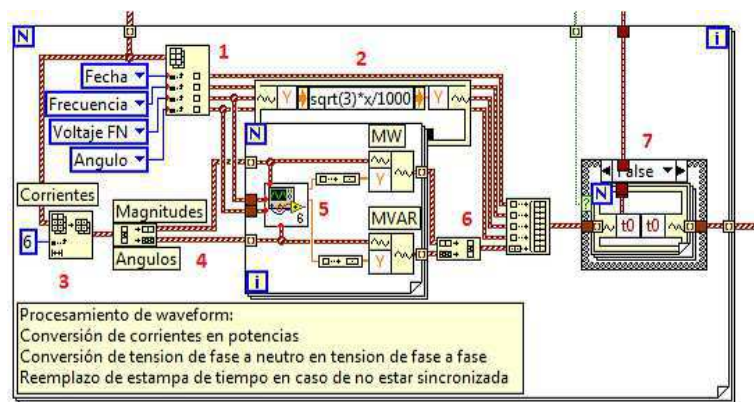


Figura 4.48: Visualizador 20

Las funciones de esta sección son:

1. Extraer del arreglo, los renglones correspondientes a la fecha, frecuencia, voltaje de fase a neutro y ángulo del voltaje.
2. Convertir la tensión de fase a neutro en tensión de fase a fase expresada en [kV].
3. Extraer del arreglo, los renglones correspondientes a las corrientes.
4. Separar los valores de las corrientes en magnitudes y ángulos.
5. Con los datos de tensión y corriente, calcular las potencias.
6. “Empalmar” los valores de potencia activa y reactiva.
7. En caso de que el tiempo este fuera de fase, reemplazar la estampa de tiempo con el tiempo de referencia.

Calculo de potencias

En el apartado anterior el cálculo de potencias se mostró como un subVI, pero para comprender su funcionamiento ahora se presenta en forma de VI y se tiene:

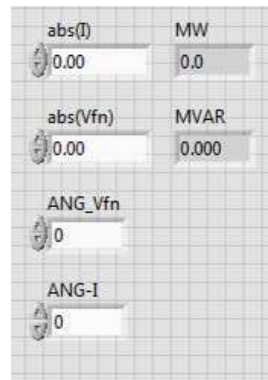


Figura 4.49: Visualizador 21 (Panel frontal)

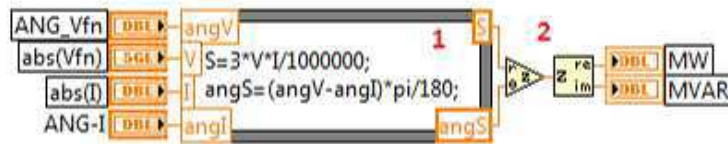


Figura 4.50: Visualizador 21 (Diagrama de bloques)

Las funciones de esta sección son:

1. El cálculo de la potencia mediante un nodo de formula empleando la ecuación 1.6.
2. La conversión de representación polar a rectangular y la separación de los valores real e imaginario.

Al término de este proceso se tienen las potencias activa y reactiva representadas en unidades de [MW] y [MVAR] respectivamente.

Agrupación de valores

Regresando al programa central, el ciclo for que incluye la conversión de corrientes a potencias, y la tensión de fase a neutro a tensión de fase a fase vuelve a agrupar los valores de todos los PMU como se muestra en la figura: 4.51

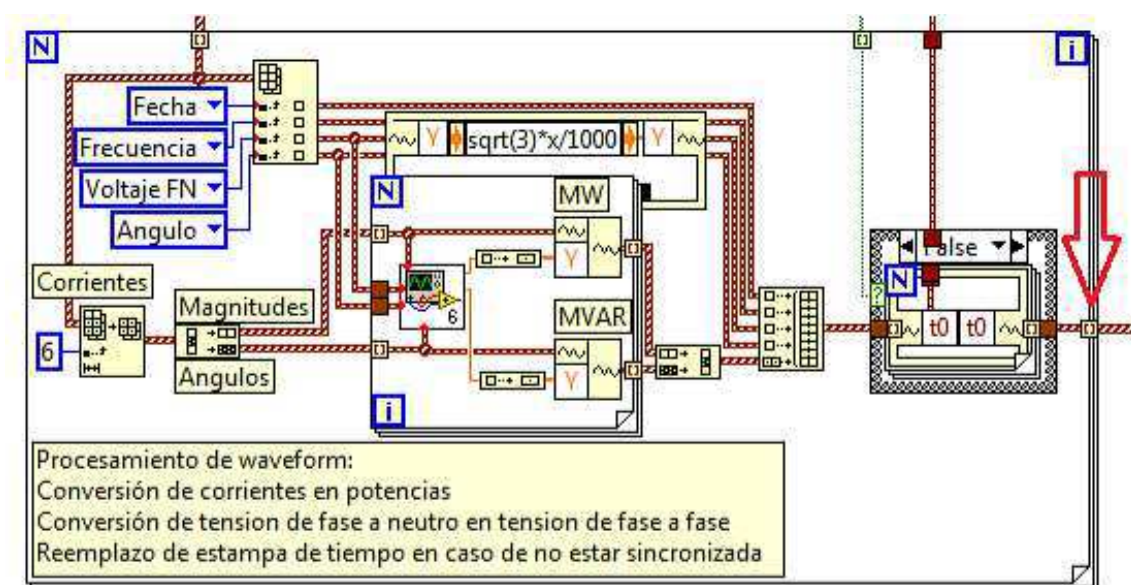


Figura 4.51: Visualizador 22

La nueva configuración del arreglo se puede describir mediante la tabla:

	PMU 1	PMU 2	...	PMU n-1	PMU n
Fecha					
Frecuencia					
Voltaje FF [kV]					
Ángulo v					
P1 [MW]					
Q1 [MVAR]					
P2					
Q2					
...					
...					
P m-1					
Q m-1					
P m					
Q m					

Cuadro 4.3: Formato con potencias y tensión

4.4.9. Graficado

De forma visual según el criterio de los operadores, los elementos más relevantes son: la magnitud de la tensión, la frecuencia, la diferencia angular, y las potencias de las líneas. Bajo ese criterio estos serán los valores que se graficarán. En forma programada, este proceso se implementó como se muestra en la figura: 4.52

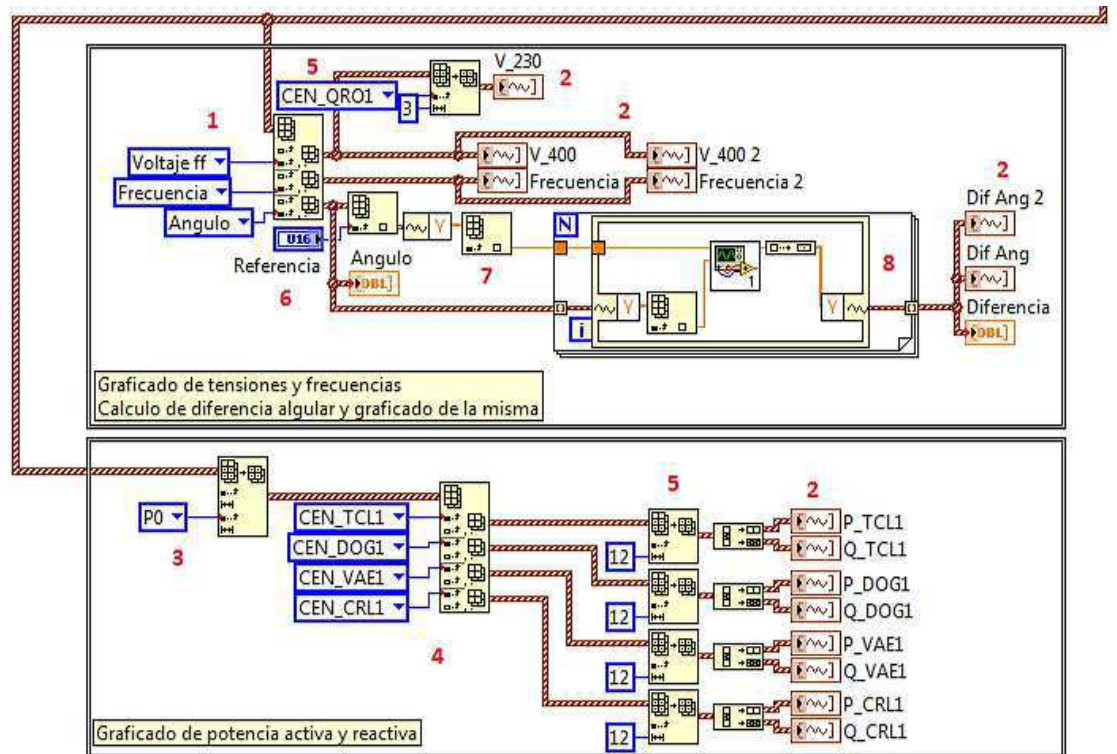


Figura 4.52: Visualizador 23

Las funciones de esta sección son:

1. Selección del parámetro a graficar
2. Graficado de los parámetros
3. Sub seccionamiento de toda la trama comenzando por el parámetro indicado en la constante
4. Selección de la subestación (columna) de la que se mostraran los parámetros (potencias)
5. Delimitación de la longitud de los parámetros (número de elementos en el arreglo) y separación de potencias en activa y reactiva.
6. Selección de referencia para el parámetro “diferencia angular”
7. Extracción del valor del ángulo
8. Calculo de la diferencia angular

Al término de este proceso se tienen graficados los valores de los parámetros más relevantes.

Cálculo de diferencia angular

En el apartado anterior el cálculo de la diferencia angular se mostró como un subVI, pero para comprender su funcionamiento ahora se presenta en forma de VI y se tiene:

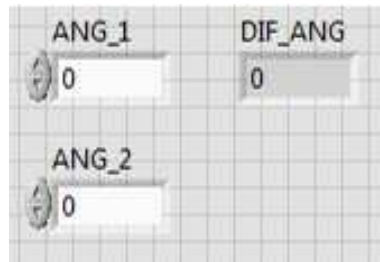


Figura 4.53: Diferencia angular (Panel frontal)

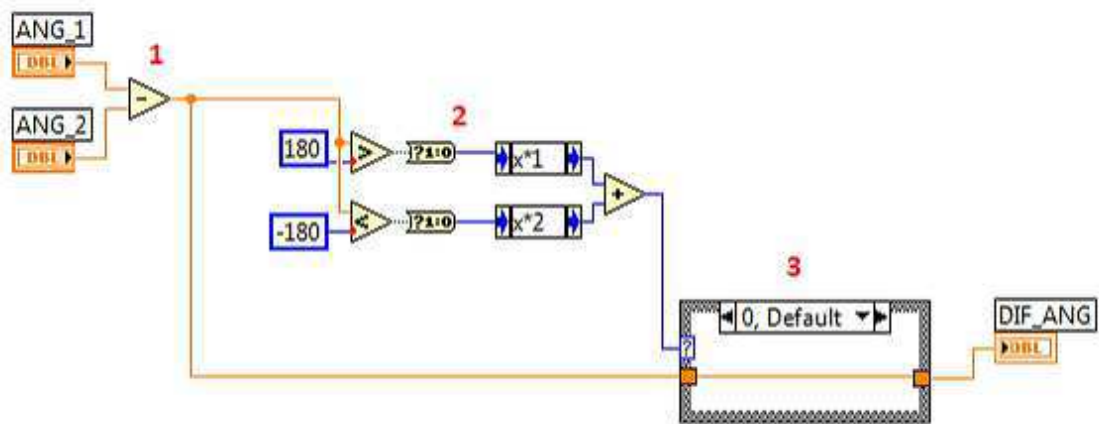


Figura 4.54: Diferencia angular (Diagrama de bloques)

Las funciones de esta sección son:

1. El cálculo de la diferencia angular
2. La asignación de un numero dependiendo el resultado de la diferencia
 - 0 si : $-180 < \text{dif} < 180$
 - 1 si : $180 < \text{dif}$
 - 2 si : $\text{dif} < -180$
3. Procesamiento del resultado de acuerdo al caso

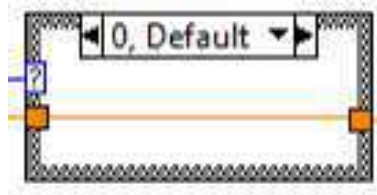


Figura 4.55: Caso 0: el resultado pasa directamente

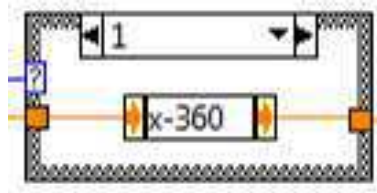


Figura 4.56: Caso 1: se resta 360 al resultado

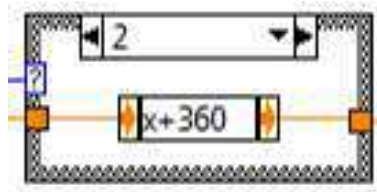


Figura 4.57: Caso 2: se suma 360 al resultado

Al término de este proceso se tienen los valores de la diferencia angular.

4.5. Guardado.vi

Es un programa basado en el visualizador, al cual se le agrega la función de guardar un registro de los valores que se van presentando en forma de tablas en formato .xls (Excel) Este programa se ejecutara en aquellos equipos conectados a la red del equipo central, los cuales requieran visualizar valores y guardar un registro de los mismos. Las funciones de este programa son: Adquirir los frames de configuración y datos, publicados por equipo central mediante el programa de adquisición. Decodificar los frames de datos y configuración, y transformarlos en valores fasoriales. Darle el procesamiento necesario a los valores fasoriales y mostrarlos mediante graficas o indicadores. Guardar los datos en forma de tablas de valores en formato .xlsx (excel)

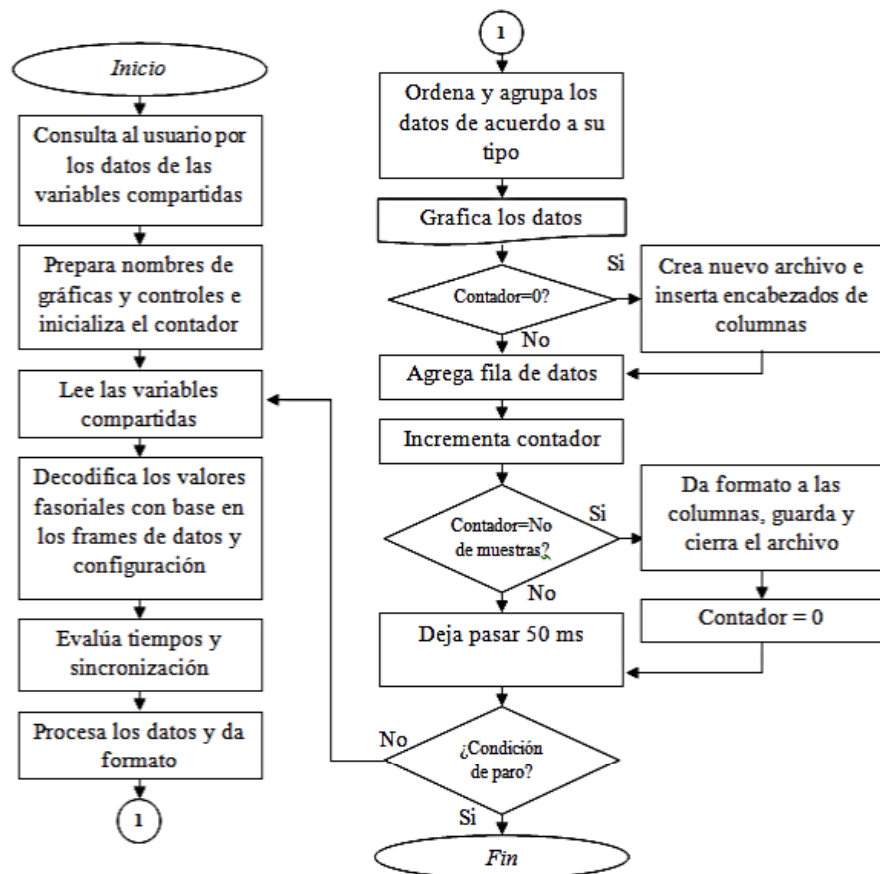


Figura 4.58: Guardado Algoritmo Básico

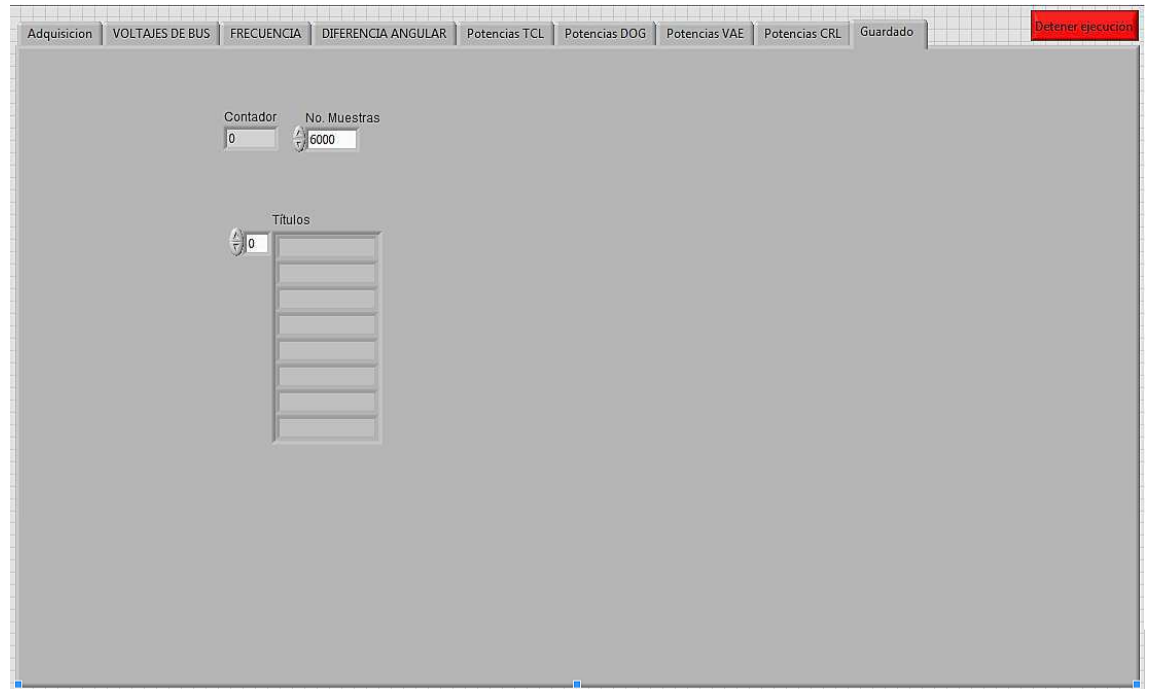
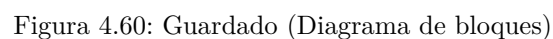


Figura 4.59: Guardado (Panel frontal)



Como se mencionó en su descripción, el programa “guardado” se trata de un programa basado en el programa “visualizador” por lo que la parte de procesamiento y muestra de los valores fasoriales son idénticas, como se puede observar en el algoritmo básico y en la figura 4.61, donde la sección del programa encerrada con rojo se trata del mismo código que el visualizador y su funcionamiento se explicó en el capítulo 4.4.

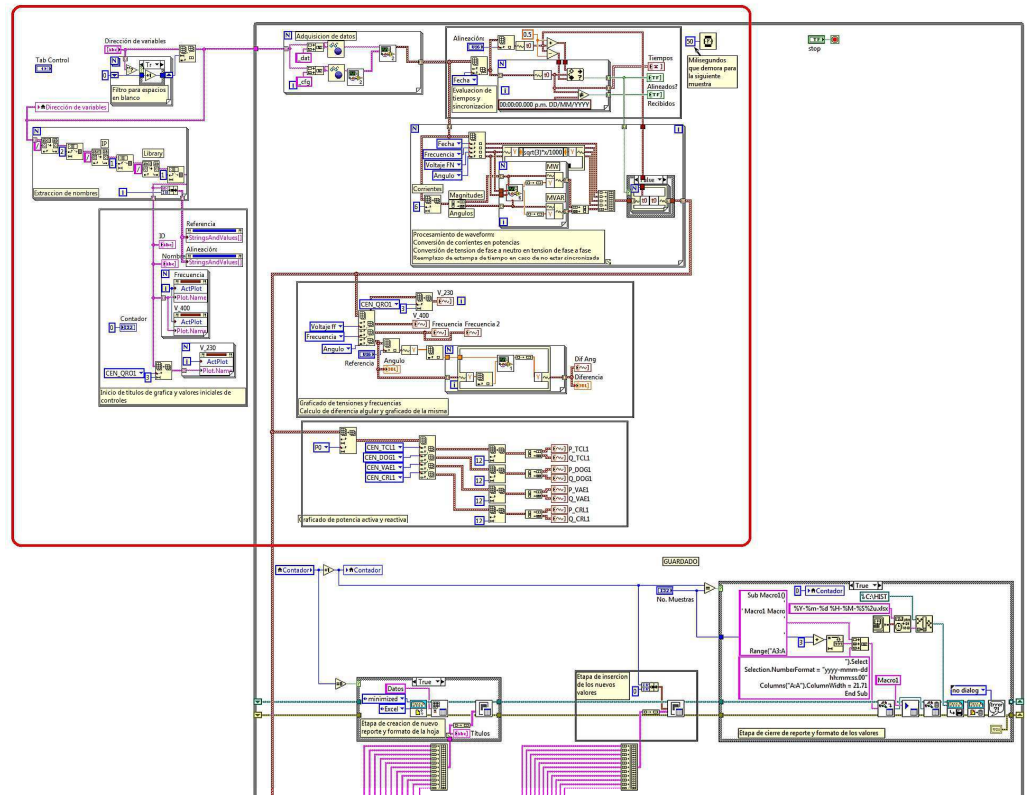


Figura 4.61: Sección de guardado correspondiente al visualizador

4.5.1. Inicio del reporte

Como se puede observar en el algoritmo básico del programa, cuando este comienza y cuando el contador vale 0, se genera un nuevo reporte en Excel y se insertan los encabezados de las columnas de los datos que se van a guardar.

En forma programada, este proceso se implementó como se muestra en la figura 4.62

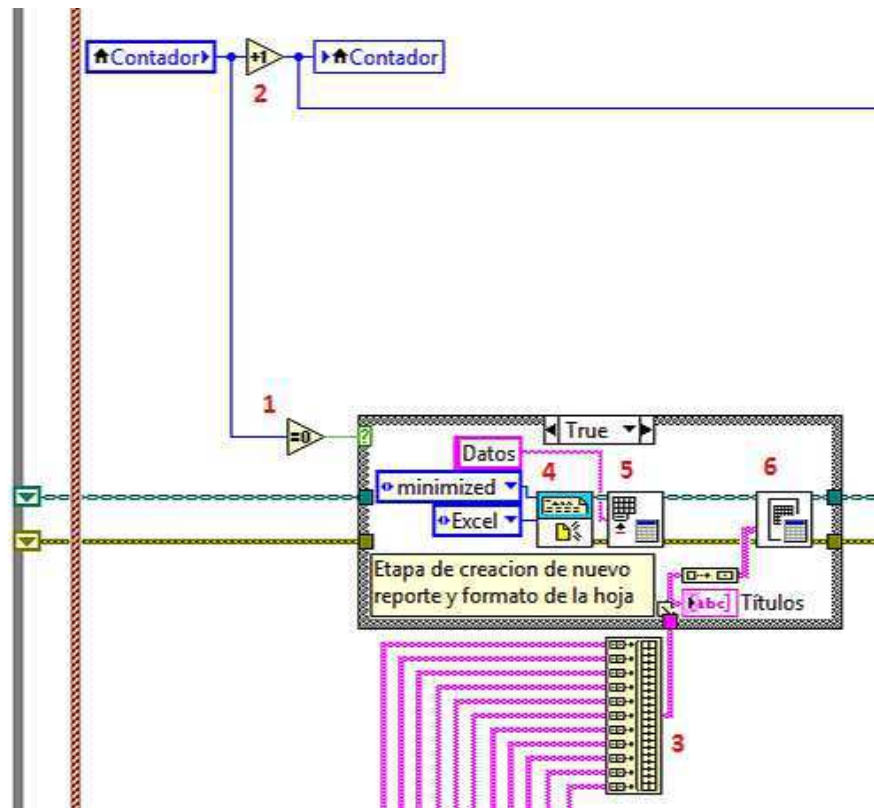


Figura 4.62: Guardado 1

La función de esta sección es:

1. Determinar si el valor del contador es 0
2. Incrementar el contador
3. Agrupar los nombres de encabezados de columnas

Si el valor del contador era 0:

4. Iniciar un nuevo reporte
5. Nombrar la hoja de Excel como "Datos"
6. Insertar los encabezados de columnas

Si el valor del contador era distinto de 0, esta sección no realiza ninguna acción.

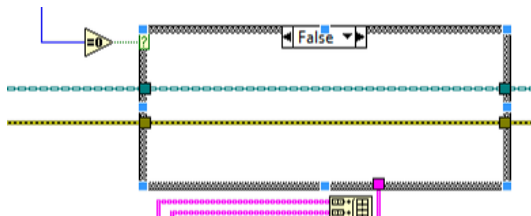


Figura 4.63: Guardado 1.1

4.5.2. Inserción de datos

Una vez que inicia el programa, cada vez que se repita el ciclo de “adquisición-decodificación-visualización” se insertara un renglón con los valores de los datos actuales de los PMU.

En forma programada, este proceso se implementó como se muestra en la figura 4.64

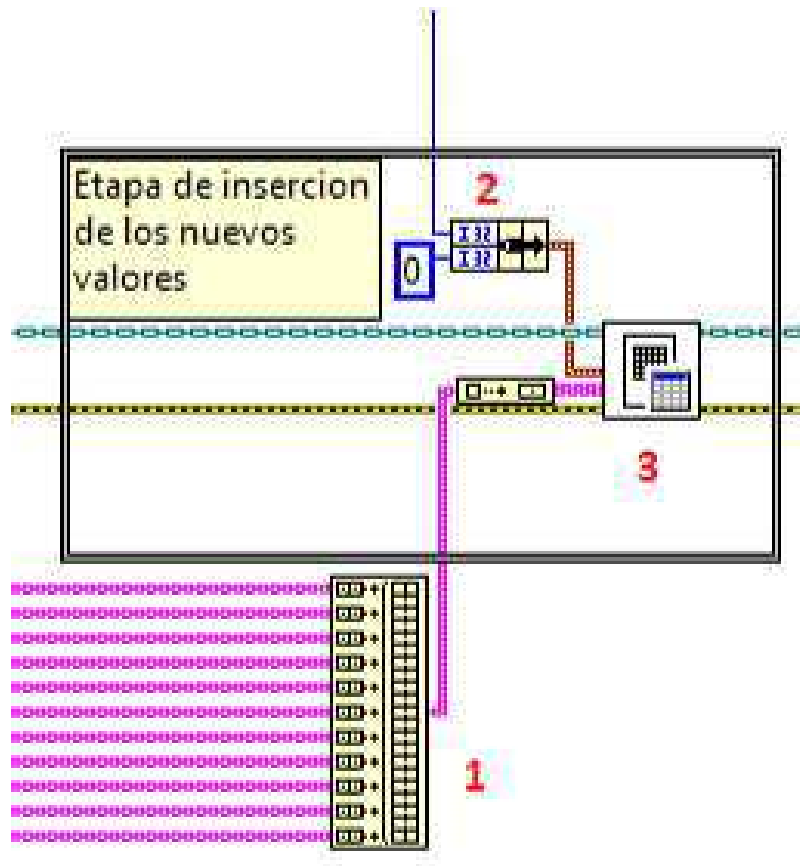


Figura 4.64: Guardado 2

La función de esta sección es:

1. Agrupar los valores actuales de los PMU
2. Determinar la fila donde se insertaran los valores con base en al valor del contador post incrementado.
3. Insertar los datos.

4.5.3. Formato, guardado y cierre del reporte

Cuando el contador del reporte llega al número de muestras indicado por el usuario, el programa procede a dar formato a las celdas, guardar el reporte de acuerdo a la fecha de la última muestra y posteriormente cerrarlo.

En forma programada, este proceso se implementó como se muestra en la figura: 4.65

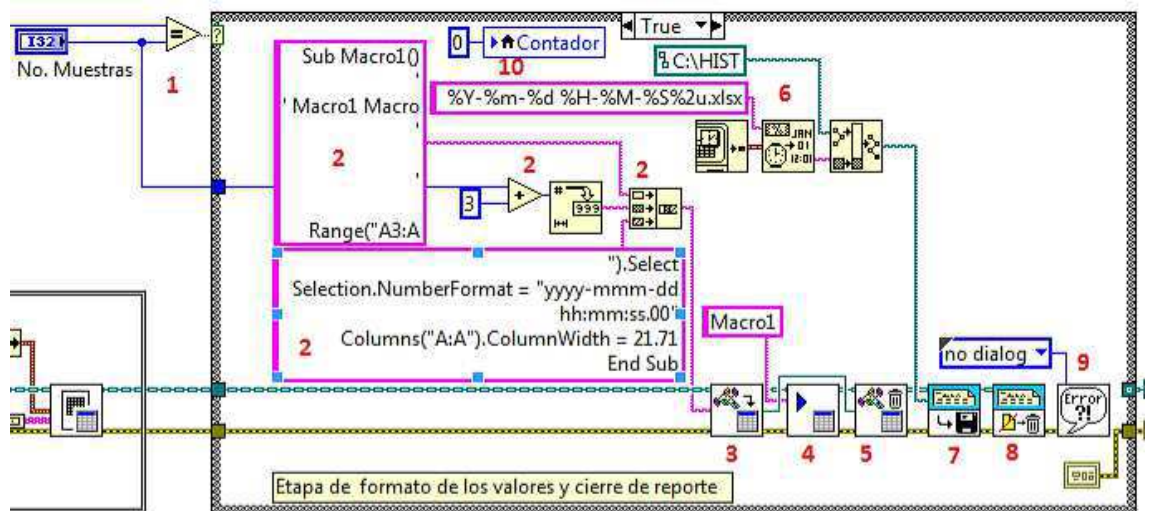


Figura 4.65: Guardado 3

La función de esta sección es:

1. Determinar si el contador ha alcanzado el número de muestras definido por el usuario

Si el contador ha llegado al número de muestras:

2. Crear el código de una macro que dará formato a la columna de fecha y hora.
3. Insertar la macro

4. Ejecutar la macro
5. Eliminar la macro
6. Crear el nombre del archivo y su ruta de guardado
7. Guardar el archivo
8. Eliminar el reporte
9. Procesar un posible error
10. Reiniciar el contador

Si el contador aun no llega al número de muestras, esta sección no realiza ninguna acción

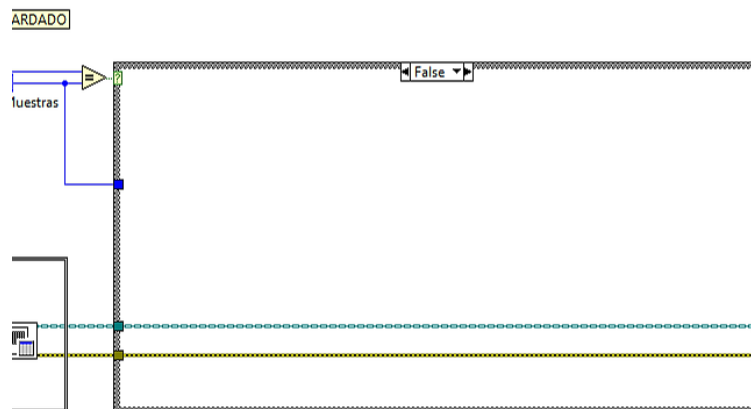


Figura 4.66: Guardado 3.1

4.5.4. Preparación de encabezado y datos

En los apartados de inicio del reporte e inserción de datos, se habló de la agrupación de los datos de los PMU y de los encabezados, para determinar dichos encabezados y para hacer la conversión de los datos numéricos a “string” (tipo de dato que permite Excel), se creó un subVi que realiza ambos procesos.

En forma programada, este proceso se implementó como se muestra en la figura 4.67

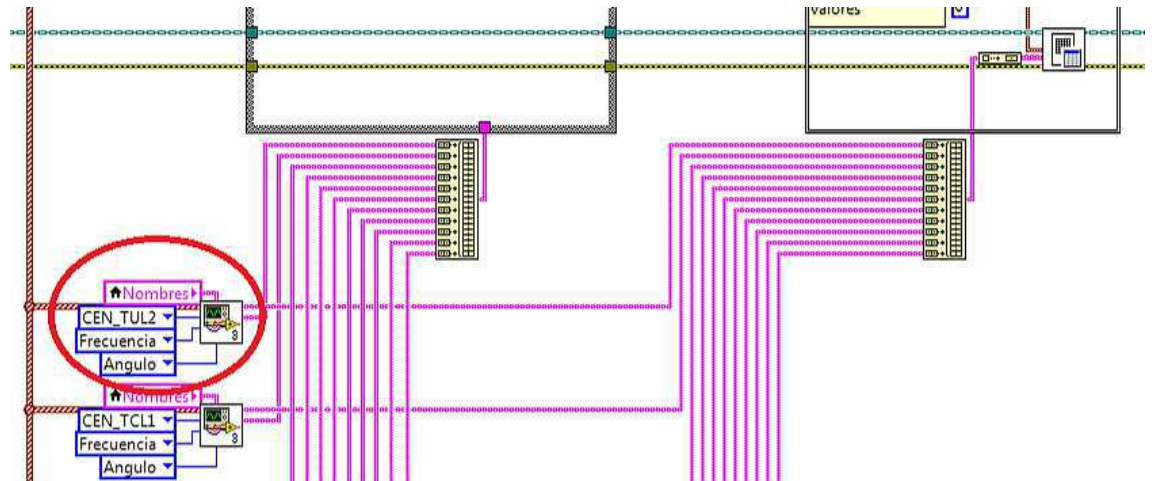


Figura 4.67: Guardado 4

Este VI recibe como datos de entrada:

- El arreglo de waveforms con los datos de todos los fasores
- El arreglo de nombres de las subestaciones que se están procesando
- El índice de la subestación
- El índice del primer parámetro a guardar
- El índice del último parámetro a guardar

Entrega como datos de salida:

- Los encabezados de las columnas de los datos seleccionados.
- Los datos actuales seleccionados.

Para comprender su funcionamiento el subVI, ahora se presenta en forma de VI y se tiene:

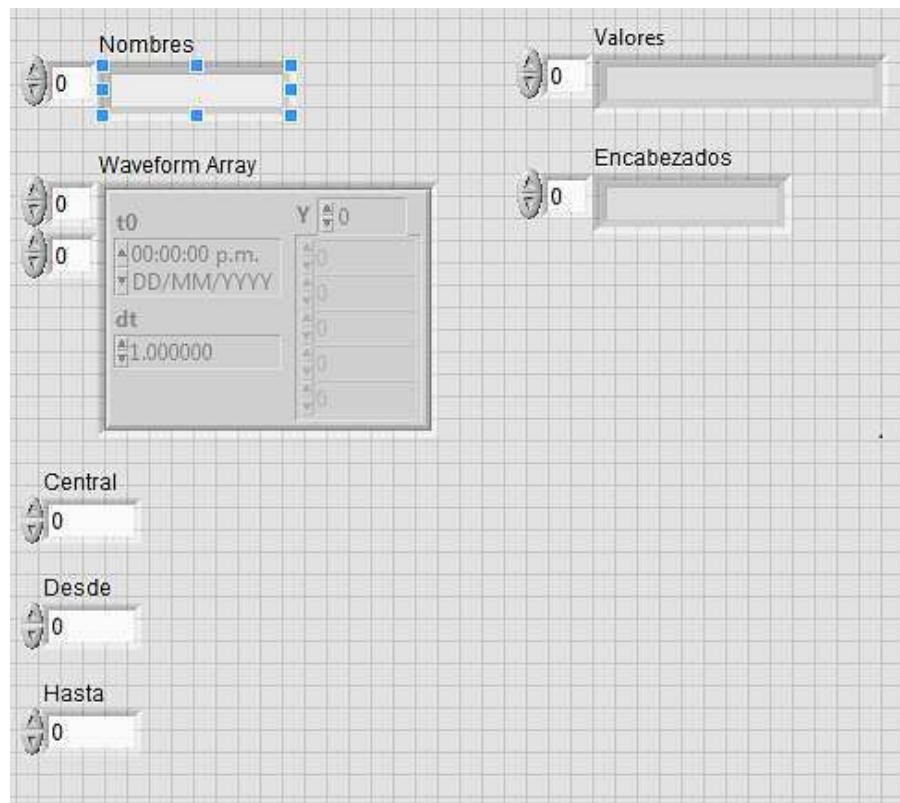


Figura 4.68: Guardado 5 (Panel frontal)

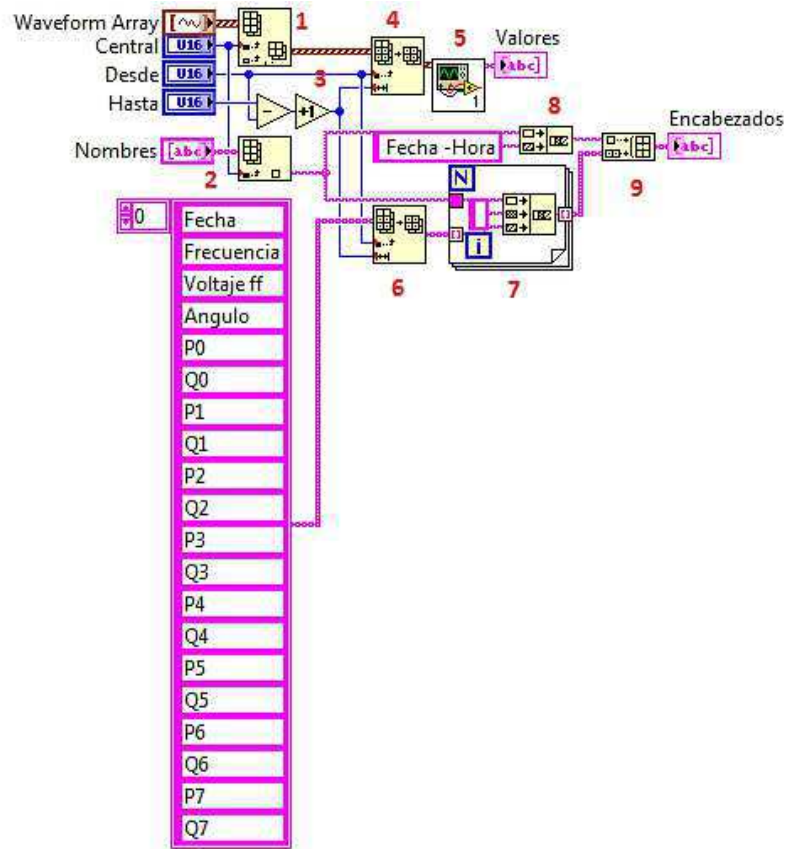


Figura 4.69: Guardado 5(Diagrama de bloques)

La función de esta sección es:

1. Seleccionar del arreglo de waveform, la columna correspondiente a los datos de la central
2. Seleccionar del arreglo de nombres, el identificador de la central
3. Calcular la longitud del sub arreglo
4. Seccionar de los datos de la central, los elementos que se van a guardar, indicando el índice del primer valor y la longitud del nuevo arreglo

5. Convertir los valores de waveform a string
6. Seccionar del arreglo constante de encabezados, los correspondientes a los elementos que se van a guardar, indicando el índice del primer valor y la longitud del nuevo arreglo.
7. Concatenar el nombre del identificador y los parámetros seccionados
8. Concatenar el nombre del identificador y el parámetro “Fecha-Hora”
9. Agrupar todos los encabezados

Al término de este proceso se tienen los siguientes encabezados y sus respectivos valores:

Encabezados VAE Potencias CRL Guardado		
Pos	Alineación:	CEN_TOP1
D/MM/YYYY		CEN_TUL2
D/MM/YYYY		CEN_TCL1
D/MM/YYYY		CEN_PBD1
D/MM/YYYY		CEN_TEX1
D/MM/YYYY		CEN_QRP1
D/MM/YYYY		CEN_DOG1
D/MM/YYYY		CEN_TOP1
D/MM/YYYY		CEN_QRO1
D/MM/YYYY		CEN_CRL1
D/MM/YYYY		CEN_VAE1
D/MM/YYYY		
D/MM/YYYY		
D/MM/YYYY		

Figura 4.70: Identificadores (Parametro de entrada)



Figura 4.71: Parámetros de entrada (fijos en la programación)

Encabezados	
0	CEN_TUL2 Fecha -Hora
	CEN_TUL2 Frecuencia
	CEN_TUL2 Voltaje ff
	CEN_TUL2 Angulo
	CEN_TUL2 P0
	CEN_TUL2 Q0
	CEN_TUL2 P1
	CEN_TUL2 Q1
	CEN_TUL2 P2
	CEN_TUL2 Q2
	CEN_TUL2 P3
	CEN_TUL2 Q3

Figura 4.72: Encabezados (de las columnas de la hoja de excel)

4.6. Alarma.vi

Es un programa basado en el visualizador, al cual se le agrega la función de mantener un registro temporal de valores, y hasta que se cumple cierta condición en el comportamiento de los parámetros los guarda, grafica y envía por correo electrónico.

Este programa se ejecutara en aquellos equipos conectados a la red del equipo central, cuya función sea monitorear los parámetros de los PMUs. Las funciones de este programa son:

- Adquirir los frames de configuración y datos, publicados por equipo central mediante el programa de adquisición.
- Decodificar los frames de datos y configuración, y transformarlos en valores fasoriales.
- Darle el procesamiento necesario a los valores fasoriales y mostrarlos mediante graficas o indicadores.
- Mantener temporalmente, un histórico de los valores fasoriales cuya una longitud será definida por el usuario (búfer).
- Guardar, graficar y enviar por correo electrónico los valores fasoriales cuando se cumple una condición en su comportamiento.

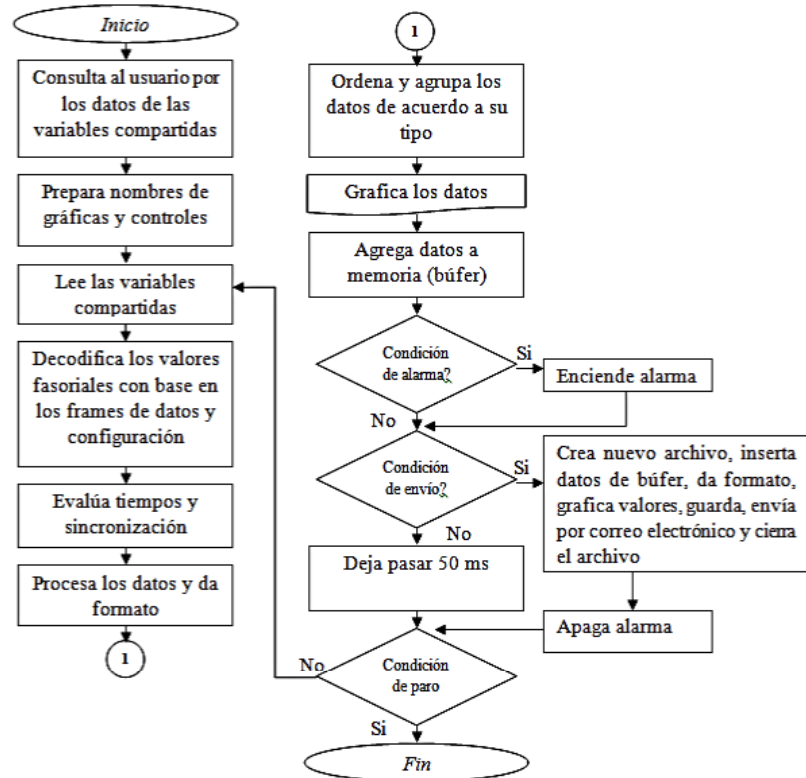


Figura 4.73: Alarma Algoritmo Básico

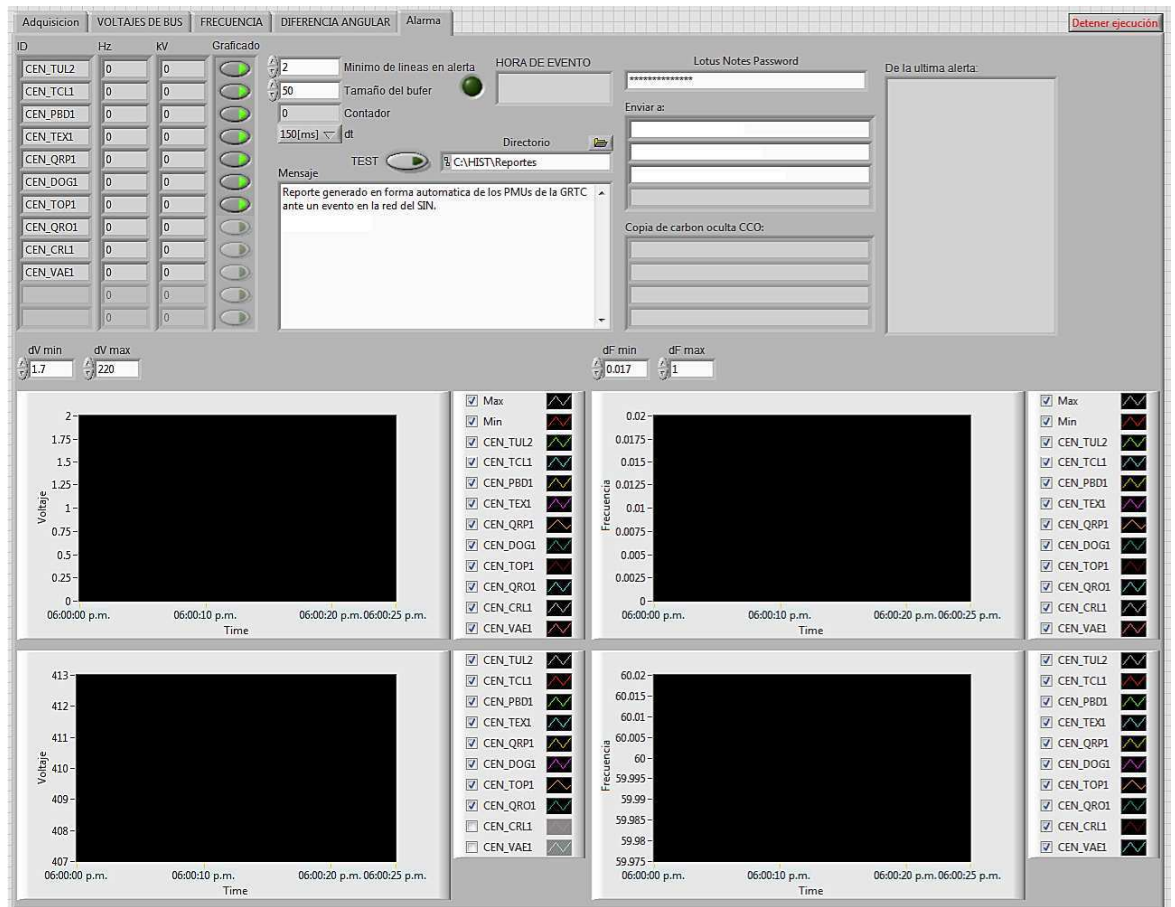


Figura 4.74: Alarma (Panel frontal)

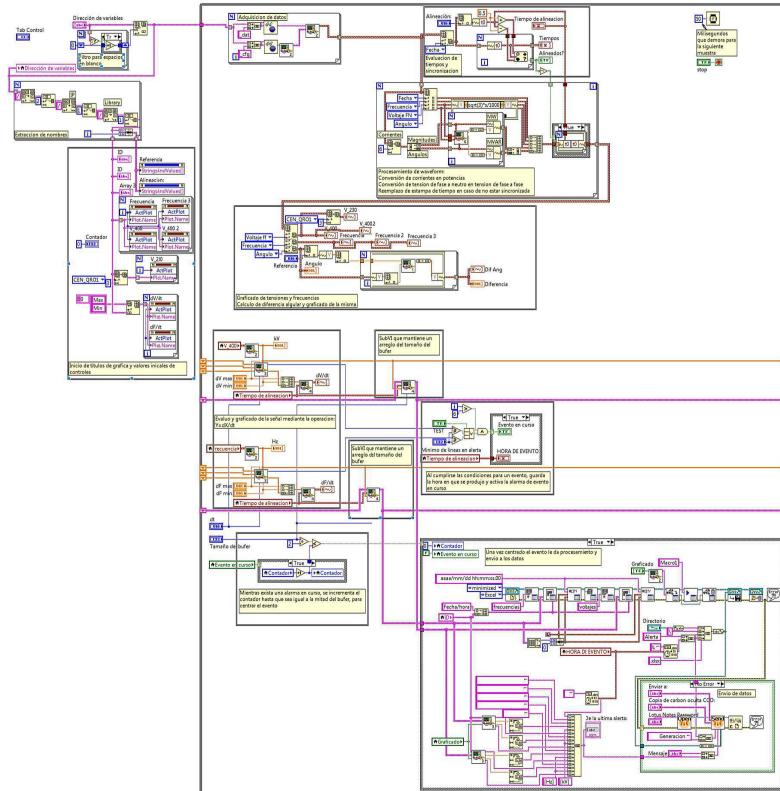


Figura 4.75: Alarma (Diagrama de bloques)

Como se mencionó en su descripción, el programa “alarma” se trata de un programa basado en el programa “visualizador” por lo que la parte de procesamiento y muestra de los valores fasoriales son idénticas, como se puede observar en el algoritmo básico y en la figura 4.76, donde la sección del programa encerrada con rojo se trata del mismo código que el visualizador y su funcionamiento se explicó en el capítulo 4.4:

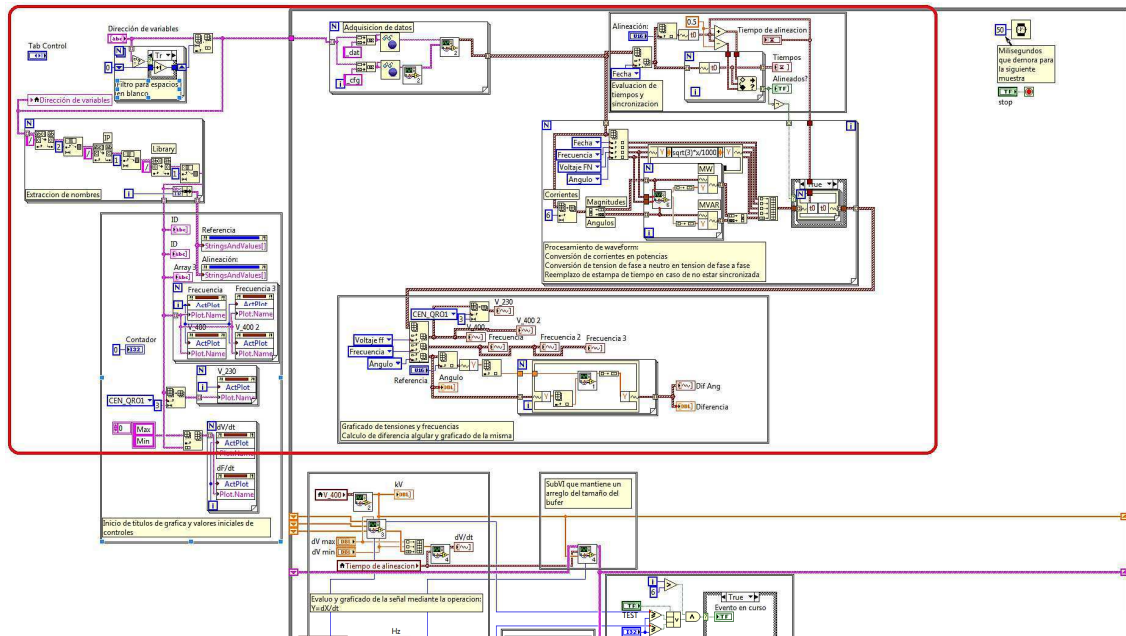


Figura 4.76: Sección de Alarma correspondiente al visualizador

4.6.1. Búfer

Como se mencionó en su descripción, este programa mantiene un registro temporal de las mediciones fasoriales, esto se logra con el subVi mostrado en la figura 4.77



Figura 4.78: Búfer (Panel frontal)

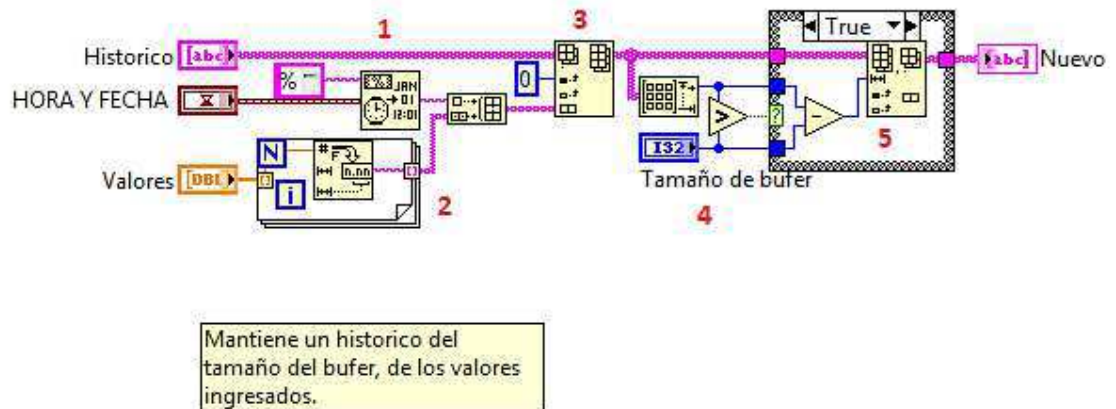


Figura 4.79: Búfer (Diagrama de bloques)

Las funciones de esta sección son:

1. Convertir el tiempo a string
2. Convertir los valores a string
3. Agrupar el tiempo y los valores
4. Determinar si el tamaño del búfer ha rebasado el tamaño que definió el usuario

Si el tamaño del búfer ha rebasado el tamaño definido por el usuario

5. Eliminar del búfer tantos renglones como sean necesarios para mantener el tamaño definido por el usuario

4.6.2. Condición de alerta

Para comenzar este estudio, se definieron dos parámetros a monitorear, uno es la magnitud de las tensiones de bus y otro la frecuencia. De forma teórica el valor de la frecuencia es de 60 [Hz] y de la tensión monitoreada 400[kV], pero en la práctica estos valores pueden variar +- 5.

Es así que una primera aproximación fue realizar una condición de alerta en los siguientes casos:

- La frecuencia es mayor a 63 [Hz]
- La frecuencia es menor a 57 [Hz]
- La magnitud de la tensión es mayor a 420 [kV]
- La magnitud de la tensión es menor a 380 [kV]

Su implementación fue sencilla y consistía en un comparador de ventana. En forma programada, este proceso se implementó como se muestra en la figura 4.80

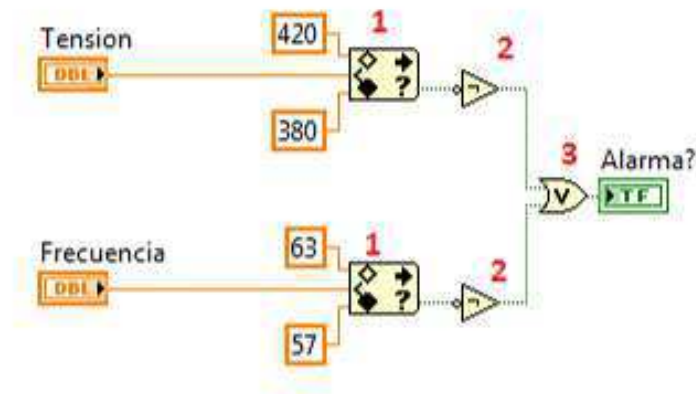


Figura 4.80: comparador

La función de esta sección es:

1. Determinar si el valor de parámetro se encuentra en el rango establecido
2. Invertir el valor de la salida
3. Sumar los resultados mediante una compuerta OR

Los resultados de esta primera implementación mostraron, que en ocasiones los parámetros operaban muy cerca de los límites llegando a cruzarlos repetidamente disparando alertas múltiples sin ser precisamente una situación anormal. También se daban falsas alerta, cuando por alguna razón se perdía comunicación con los PMU y sus valores marcaban 0.

Además si los parámetros se encontraban muy cerca del valor teórico, los eventos más relevantes como el disparo de una línea, tiro de carga, entrada de una línea en servicio, entrada de una unidad generadora entre otros, se manifestaban como oscilaciones abruptas pero sus valores se mantenían dentro de los rangos de tolerancia y eran ignoradas como se puede observar en las figuras 4.81 y 4.82

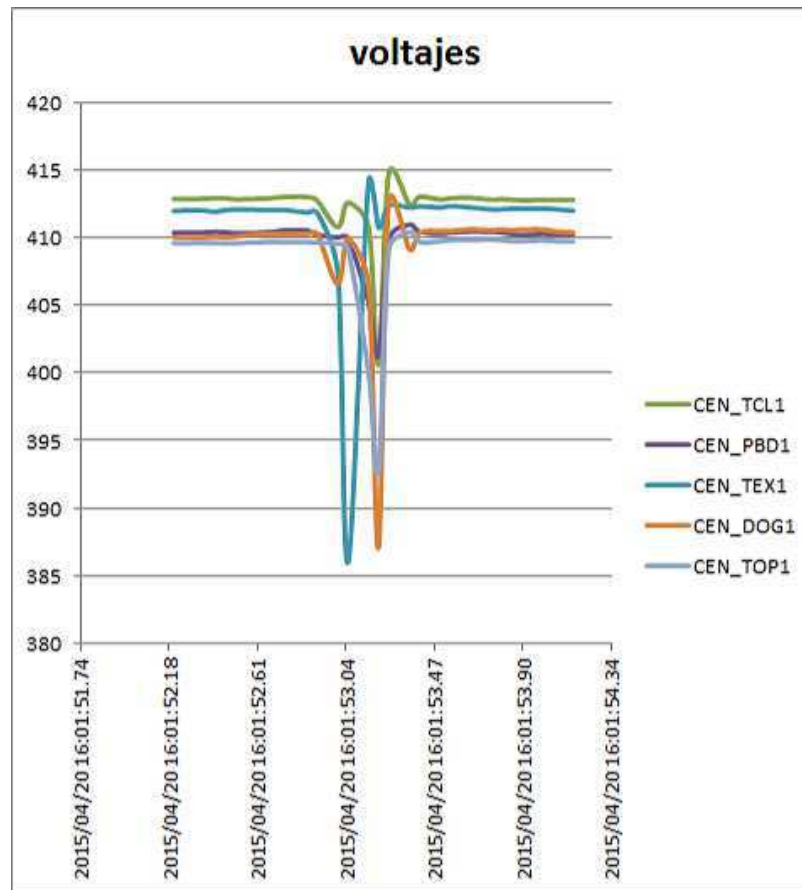


Figura 4.81: Alarma 1

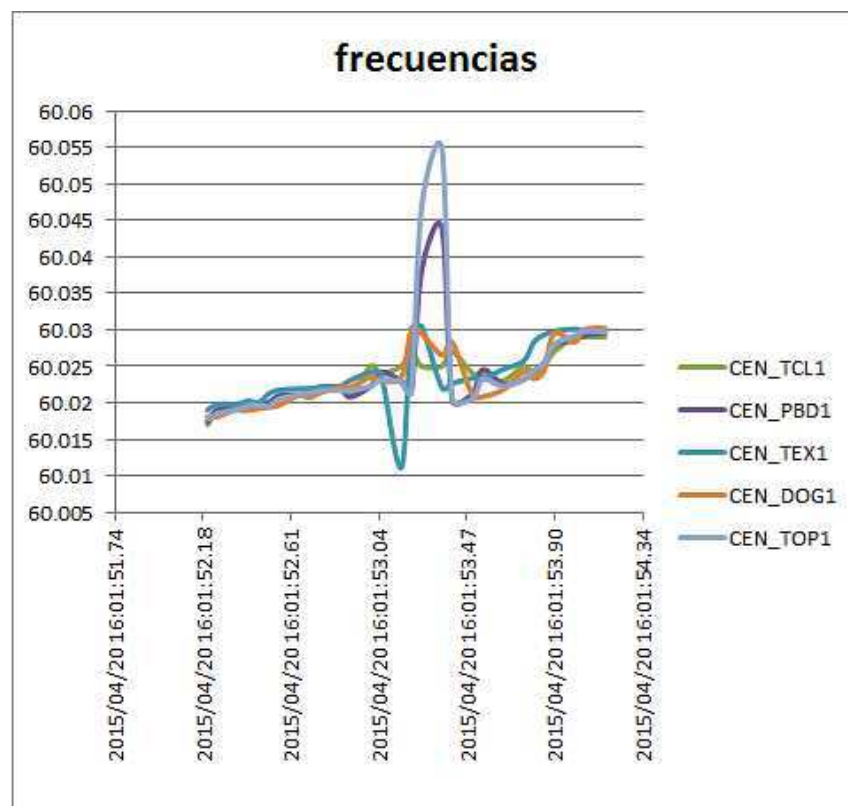


Figura 4.82: Alarma 2

El siguiente criterio consistió en calcular la relación de cambio de los valores con respecto al tiempo, en forma matemática puede describirse como $\frac{dx}{dt}$.

La razón es que al ocurrir un cambio repentino en los valores de los fasores esta relación aumenta drásticamente, pero si el cambio es gradual la relación se mantiene con un valor cercano a 0. Como se puede mostrar en las figuras (poner referencia) donde la gráfica superior corresponde a la relación $\frac{\Delta x}{\Delta t}$ y la gráfica inferior a los valores que la originan:

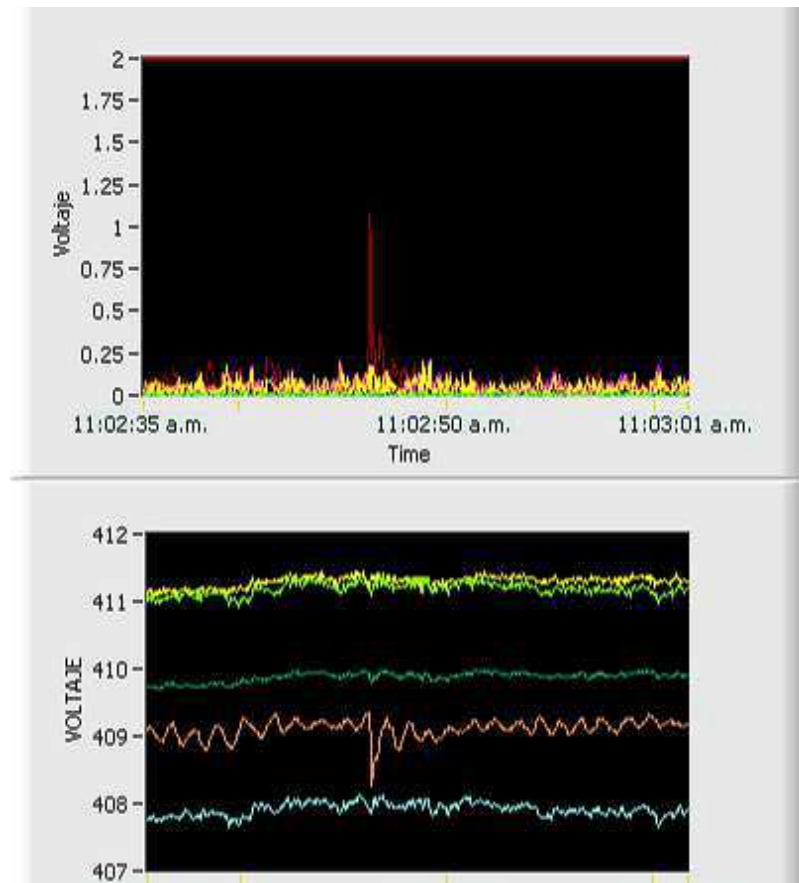


Figura 4.83: Ejemplo de perturbación en la tensión

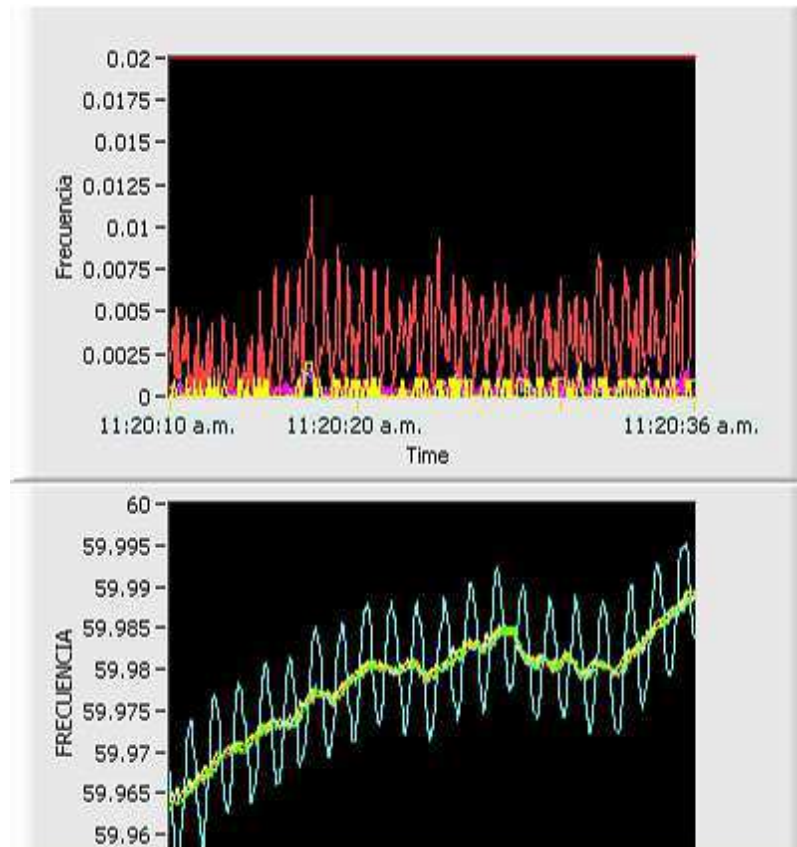


Figura 4.84: Ejemplo de perturbación en la frecuencia

Una vez establecida la relación se procede a establecer un rango para el cual se considerara un evento como tal y no solo una perturbación sin relevancia.

En forma programada, este proceso se implementó como se muestra en la figura 4.85



Figura 4.85: Alarma 3

Las funciones de esta sección son:

1. Adquirir del control del panel frontal los valores de relación máximos y mínimos para determinar la alerta
2. Calcular $\frac{\Delta x}{\Delta t}$ y determinar si existen líneas en alerta y su numero
3. Agrupar los valores máximos y mínimos y los valores de $\frac{\Delta x}{\Delta t}$
4. Convertir los valores a waveform
5. Graficar

En el punto número 2 se muestra el cálculo de $\frac{\Delta x}{\Delta t}$ como un subVI el cual: Recibe como datos de entrada:

- Los valores actual, pasado, antepasado y ante antepasado de los fasores
- Los rangos máximos y mínimos para ser considerado una alerta
- Un índice para seleccionar con qué valor pasado se efectuara el calculo

Entrega como datos de salida:

- El cálculo de $\frac{\Delta x}{\Delta t}$
- El número de líneas que cumplen con la condición de alerta

Para comprender su funcionamiento el subVI, ahora se presenta en forma de VI y se tiene:

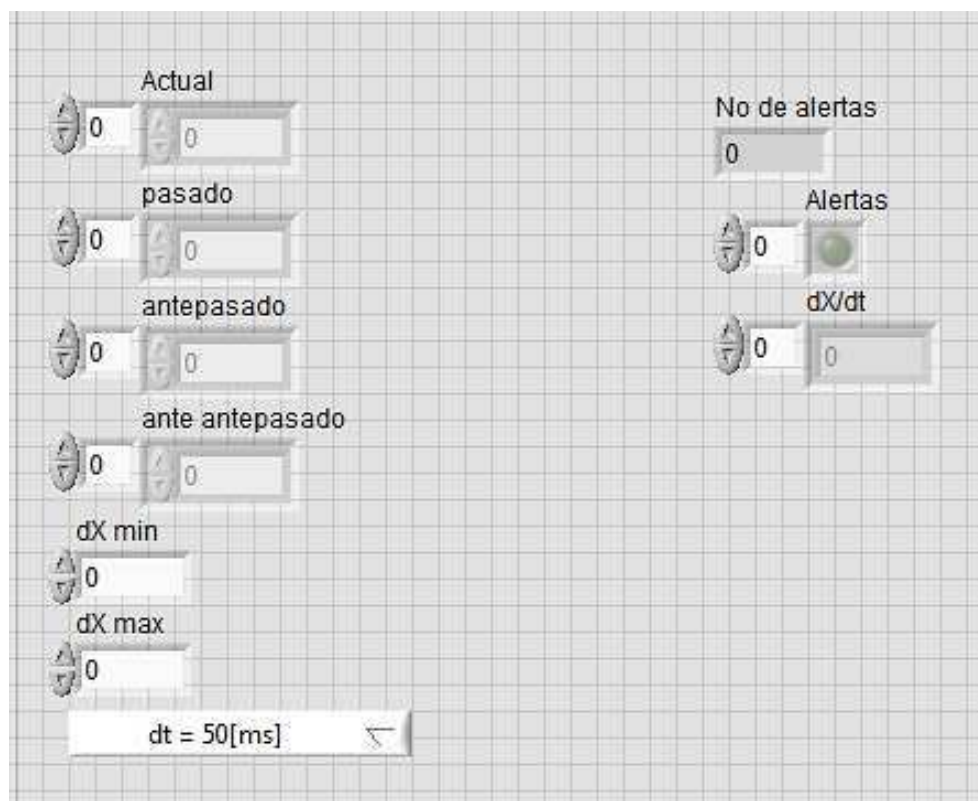


Figura 4.86: Alarma 4 (Panel frontal)

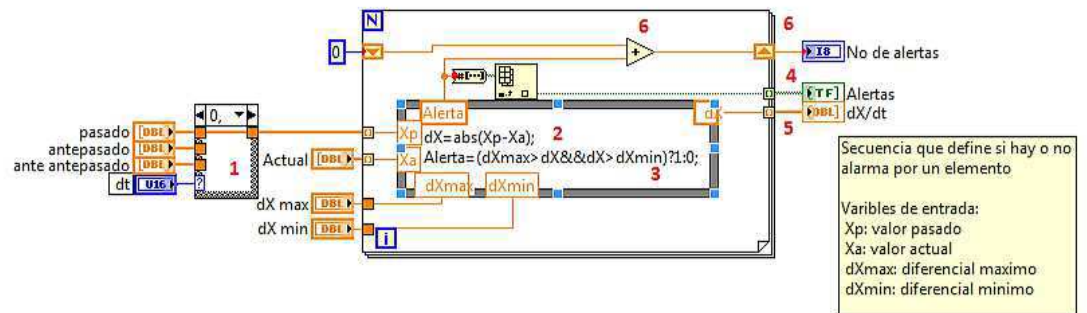


Figura 4.87: Alarma 4 (Diagrama de bloques)

Las funciones de esta sección son:

1. Determinar el valor pasado con el que se realizará la operación.
2. Calcular $\frac{\Delta x}{\Delta t}$
 Nótese que el programa ejecuta toda la subrutina cada 50[ms], por lo tanto:
 - si se selecciona el valor pasado $\Delta t = 50[ms]$
 - si se selecciona el valor antepasado $\Delta t = 100[ms]$
 - si se selecciona el valor ante antepasado $\Delta t = 150[ms]$
3. Determinar si $\frac{\Delta x}{\Delta t}$ se encuentra en el rango de la alerta.
4. Agrupar las alertas
5. Agrupar los valores de $\frac{\Delta x}{\Delta t}$
6. Determinar el número de alertas en total

4.6.3. Numero de alertas

Una vez establecido el parámetro que activara las alertas, se evaluaron los resultados de esta segunda implementación.

Se encontró que en ocasiones una línea puede sufrir una perturbación, y que puede manifestarse como una oscilación en el voltaje o frecuencia de la subestación pero sin considerarse una alerta (como la quema de caña de azúcar, papalote atorado en la línea etc.), pues este cambio no afecta a todo el sistema. Una muestra de ello se puede observar en la figura 4.88

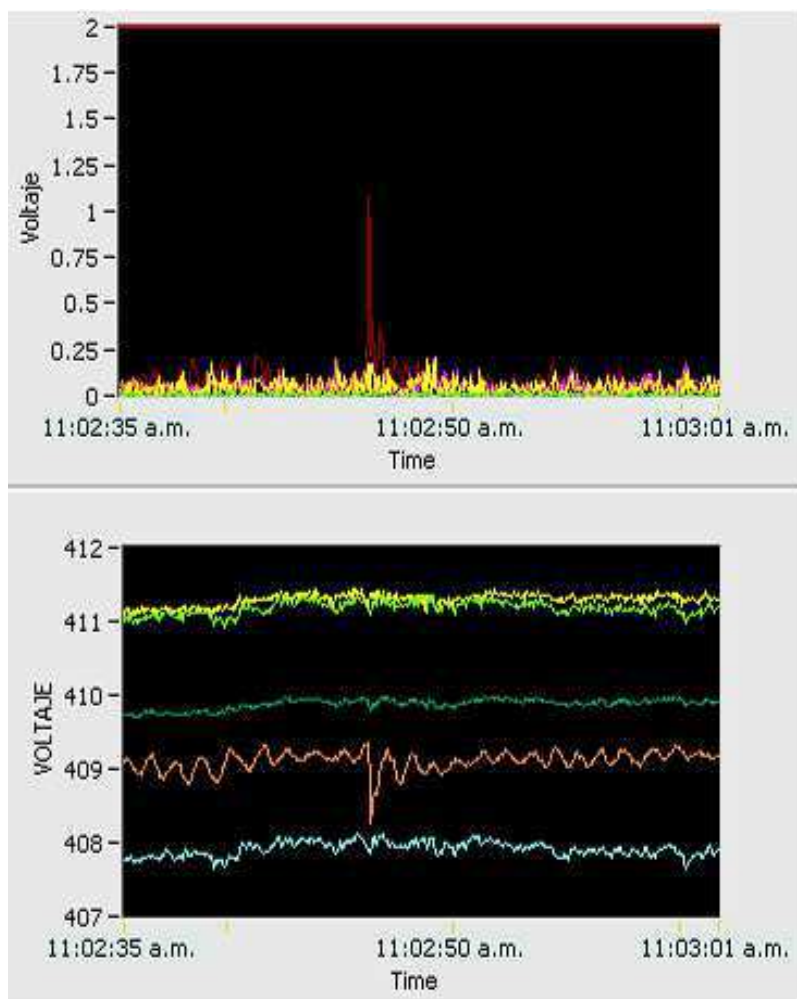


Figura 4.88: Perturbación de una subestación

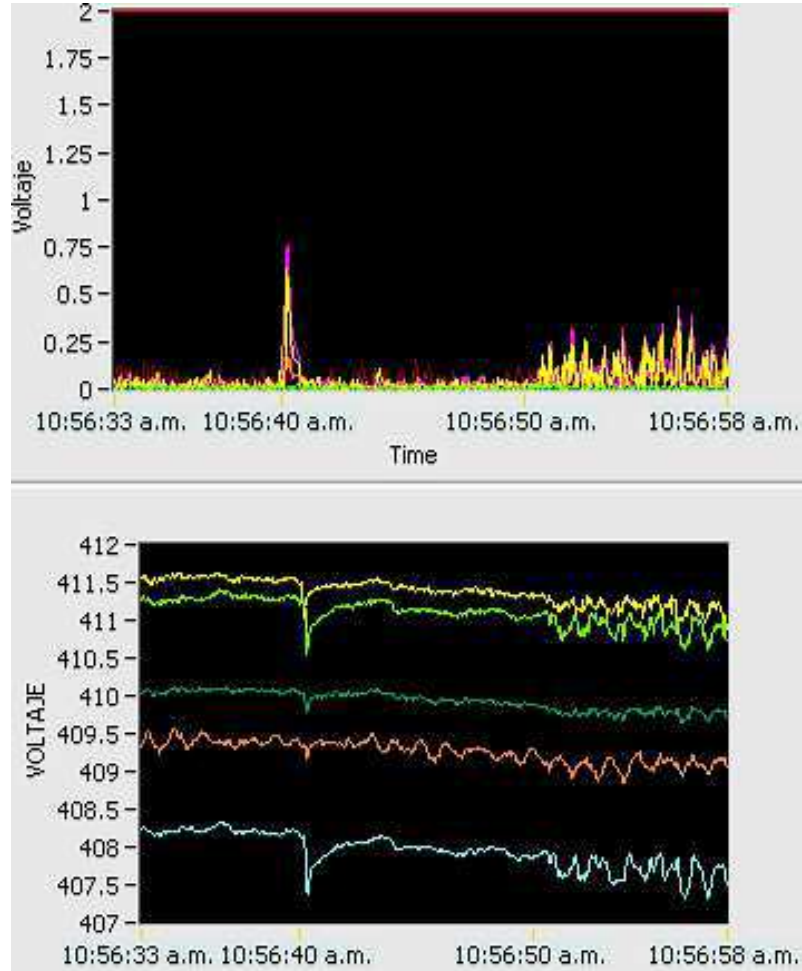


Figura 4.89: Perturbación del sistema

Contrastando la figura 4.87 que muestra una perturbación de una subestación particular, con la figura 4.88 que muestra una perturbación del sistema, se puede observar que en la perturbación particular se dispara un solo valor de $\frac{\Delta x}{\Delta t}$, mientras en la del sistema se disparan más de un valor. Es así que la segunda condición para activar una alarma es tener mínimo dos subestaciones en alerta.

En forma programada, este proceso se implementó como se muestra en la figura:4.90

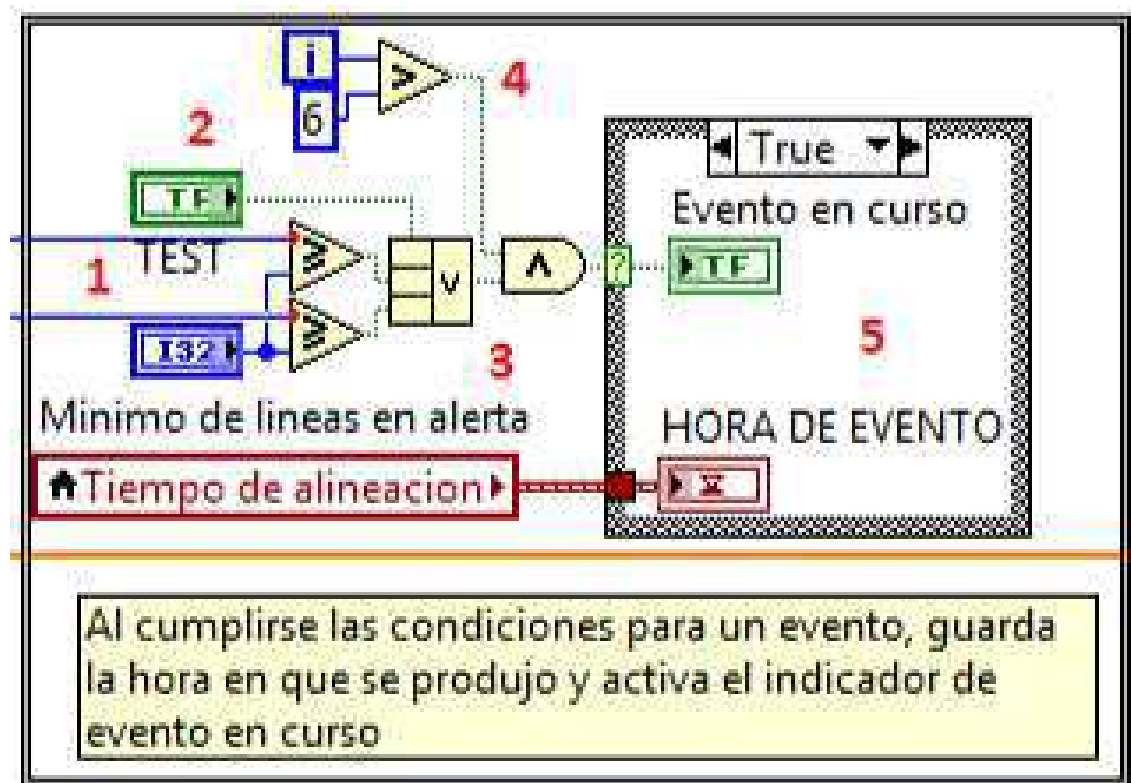


Figura 4.90: Alarma 5

Las funciones de esta sección son:

1. Determinar si el número de alertas es superior al mínimo.
2. Implementación de un botón para ejecutar el programa de forma manual.
3. Sumar las alertas manual, de frecuencia y de tensión mediante una compuerta OR.
4. Verificar que la alarma no se ejecute en las primeras repeticiones del programa.
5. Activar los indicadores de evento en curso y guardar la hora del evento en una variable local.

4.6.4. Centrado del evento

Una vez establecida la alarma y su correcto funcionamiento se encontró que si el graficado comenzaba a partir de que se accionaba la alarma, se perdía la

información previa al evento como podía ser oscilaciones múltiples, picos de tensión o frecuencia etc.

Para solucionar esta condición se implementó el búfer que mantendría los valores pasados, y se estableció un contador el cual comenzara a incrementarse a partir de que se acciona la alarma, y se detendrá cuando alcance un valor igual a la mitad del tamaño del búfer. Es en este momento que los valores temporales del búfer se guardan y procesan, de este modo se tiene que el inicio de la alarma se sitúa justo a la mitad del archivo.

En forma programada, este proceso se implementó como se muestra en la figura 4.91

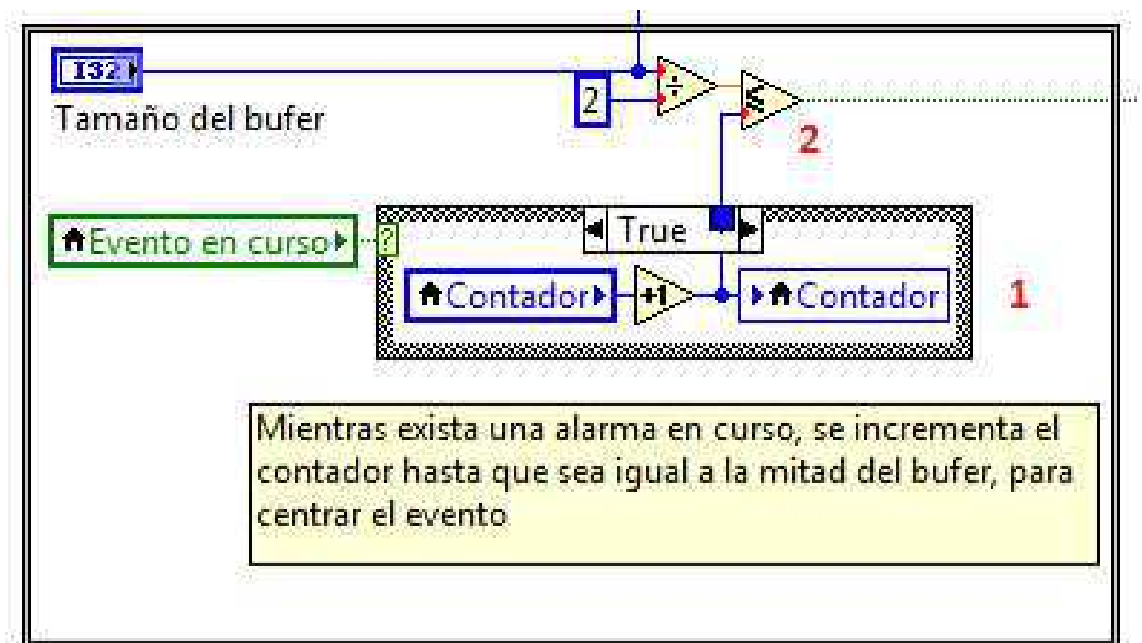


Figura 4.91: Alarma 6

Las funciones de esta sección son:

1. Incrementar el contador mientras se encuentre activo el indicador de "evento en curso"
2. Determinar si el contador es igual o mayor a la mitad del tamaño del búfer (valor definido por el usuario).

4.6.5. Guardado y envío

El proceso es similar al programa de guardado salvo que la apertura del reporte, la inserción de datos, el formato y guardado de los valores se corren en una sola rutina y no en tres diferentes, y una vez guardado este archivo se envía por correo electrónico mediante el programa “Lotus Notes” En forma programada, este proceso se implementó como se muestra en la figura: 4.92

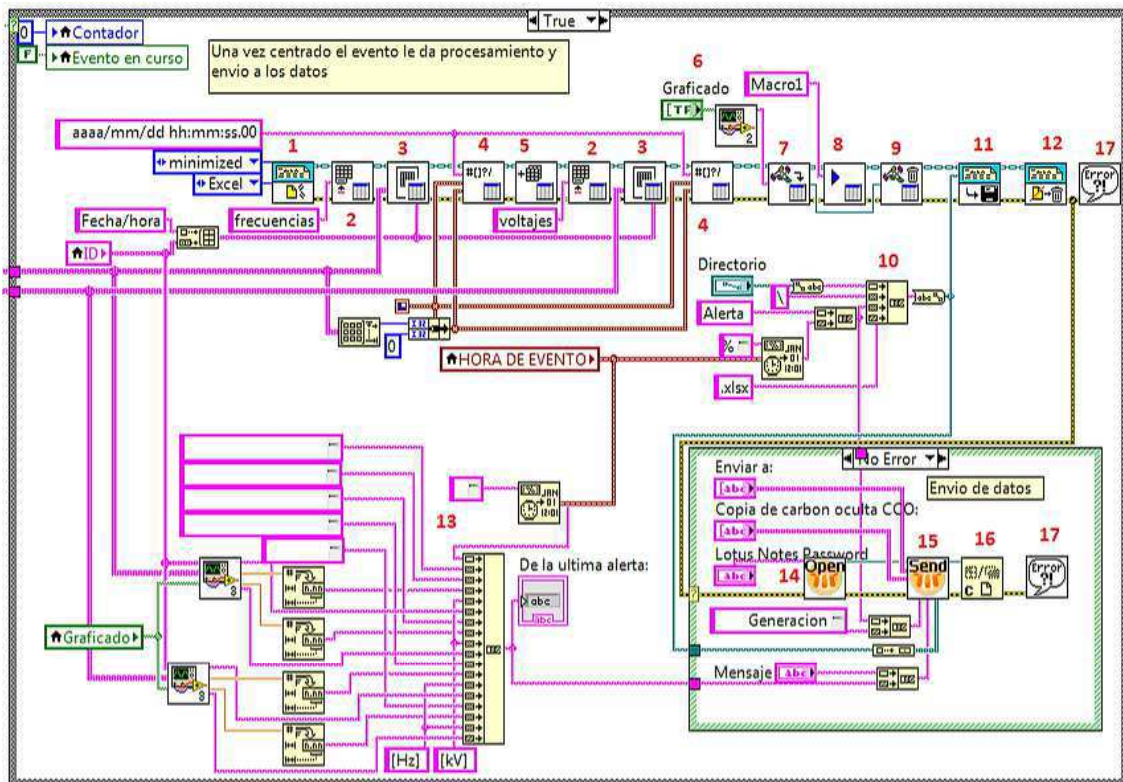


Figura 4.92: Alarma 7

Las funciones de esta sección son:

1. Iniciar un reporte con Excel
2. Cambiar el nombre de la hoja de Excel
3. Insertar los valores del búfer
4. Dar formato a las columnas del tiempo
5. Agregar una nueva hoja de Excel
6. Crear el código de la macro que creara las graficas
7. Insertar la macro al libro de Excel

8. Ejecutar la macro
9. Eliminar la macro del libro de Excel
10. Crear el nombre del archivo con la hora del evento y definir su ruta de guardado
11. Guardar el archivo
12. Cerrar el reporte
13. Crear el mensaje del cuerpo del correo
14. Abrir conexión con el programa Lotus notes introduciendo la contraseña de la cuenta.
15. Enviar un correo vía Lotus notes a los destinatarios, con la hora del evento como asunto, el mensaje del punto 13 y el archivo recién guardado como dato adjunto.
16. Cierre de lotus notes
17. Procesamiento de algún error

Ejemplo del mensaje enviado:

Reporte generado en forma automatica de los PMUs de la GRTC ante un evento en la red del SIN.

[Ing. JJLM]

Datos del evento:

Fecha: 2015/04/20

Hora: 16:01:53.55

Máximos y mínimos valores registrados:

V min: 386.001235[kV] CEN_TUL2

V max: 414.788258[kV] CEN_TEX1

F min: 59.994942[Hz] CEN_PBD1

F max: 60.054821[Hz] CEN_TOP1

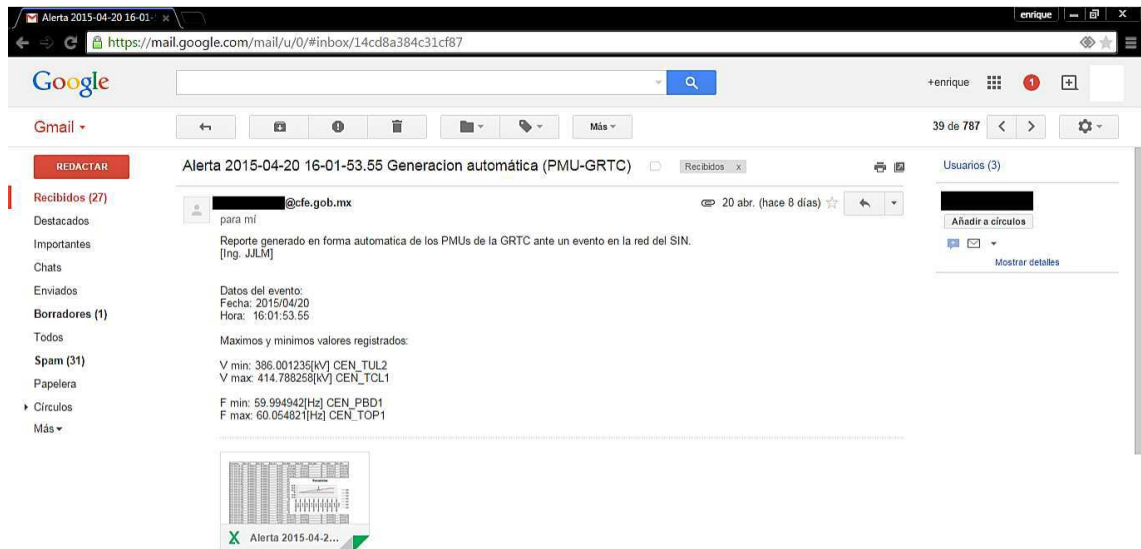


Figura 4.93: Ejemplo de correo recibido

Ejemplo de archivo de Excel recibido

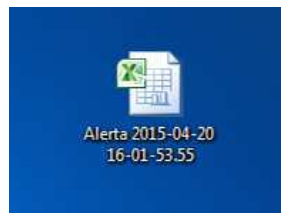


Figura 4.94: Ejemplo de Archivo recibido (Nombre del archivo)

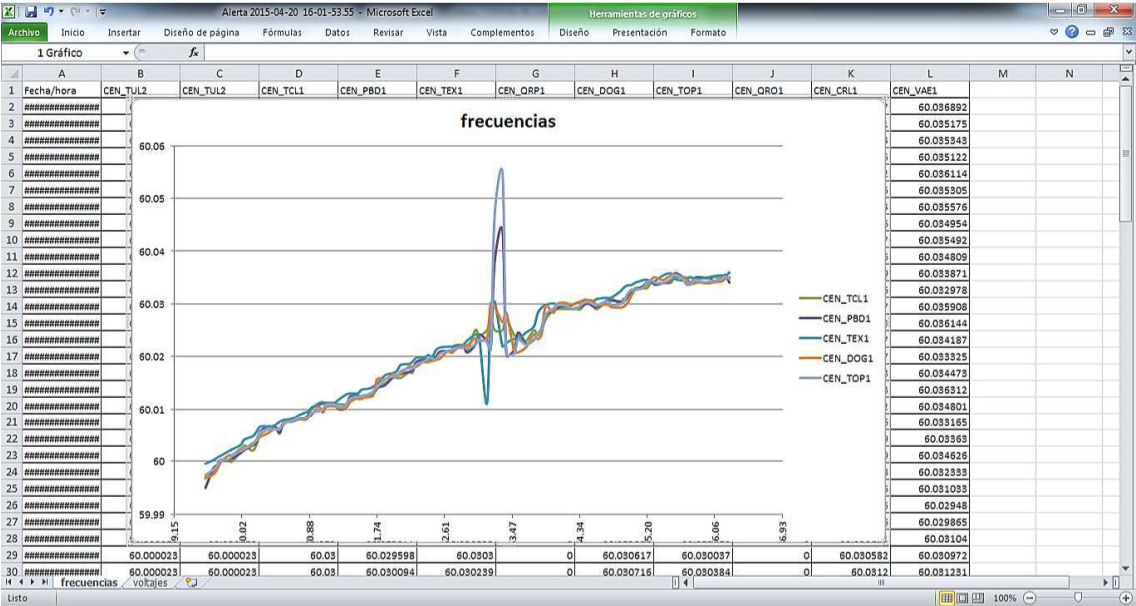


Figura 4.95: Ejemplo de Archivo recibido (hoja de voltajes)



Figura 4.96: Ejemplo de Archivo recibido (hoja de frecuencias)

Capítulo 5

Conclusiones

En el Sistema Eléctrico, los parámetros que muestran el comportamiento del sistema se pueden concentrar en 4: el valor de la frecuencia, los fasores de tensión y corriente y sus respectivos ángulos; la razón es que a partir de estos parámetros primarios se pueden calcular parámetros secundarios como potencia, diferencia angular factor de potencia etc.

Para adquirir los datos de estos parámetros primarios se emplean instrumentos de medición denominados unidades de medición fasorial PMU.

Existen para los PMU, retos intrínsecos como por ejemplo:

- Su ubicación.
- La diversidad de compañías que fabrican los PMU y por consiguiente su software particular que impide interacciones entre los datos de diversos equipo.
- Los límites en los usuarios que pueden consultar su información.

Para resolver el reto de los PMU sobre la distancia entre los equipos y los centros donde se necesitan las mediciones, los PMU cuentan con un protocolo de comunicación TCP/IP, este protocolo es aprovechado en esta aplicación debido a su estructura robusta en capas y la “transparencia” de su protocolo hacia el usuario, puesto que el usuario para establecer comunicación con la unidad solo requiere conocer la dirección IP del equipo, el puerto, y su identificador. Haciendo una analogía con las líneas telefónicas, el usuario para comunicarse con otra persona necesita saber su número de teléfono, extensión, y el nombre de con quién desea comunicarse, sin poner atención al funcionamiento de las líneas telefónicas, del aparato receptor, conmutador, etc.

Previendo la posibilidad de errores en la transmisión o recepción de datos se emplea el método de detección de errores por redundancia cíclica (CRC), que consiste en un algoritmo que trabaja con la composición de todo el mensaje (asignando a cada bit un valor numérico por su posición y efectuando divisiones) lo cual permite identificar de manera precisa cualquier anomalía en el mismo, y

en algunos casos la posibilidad de recuperar información perdida (con algoritmos secundarios que hacen la operación inversa: multiplicación).

Para resolver el reto de los PMU sobre las múltiples compañías y modelos, la organización IEEE estableció una norma que unifica el envío y recepción de información de todos los PMU, esta norma se identifica con el código “C37.118.2-2011”.

Dicha norma establece que

- La información proveniente de los PMUs se recibe en forma de mensajes denominados “FRAMES”
- Para adquirir los valores de los PMU se necesitan básicamente 2 frames:
 1. El frame de configuración, que describe cómo se organiza el frame de datos.
 2. El frame de datos, que contiene la información de los valores Fasoriales, analógicos y digitales.
- Para operar los PMU existe un frame de comandos con 7 posibles comandos, de los cuales se emplean principalmente 3:
 1. Detener flujo de datos.
 2. Enviar frame de configuración 2.
 3. Inicial flujo de datos.

Para procesar la información de múltiples PMU, y realizar interacciones entre estos de forma automática, basándose en la norma IEEE C37.118.2-2011 se procedió a diseñar 4 programas:

1. Adquisición
2. Visualizador
 - a) Guardado
 - b) Alarmas

El programa de adquisición cumple la función de obtener los frames de datos y configuración, y publicarlos mediante “variables compartidas”. Para ello establece conexión con todos los PMU y envía los comandos necesarios para conseguir dichos frames. Se implementó para resolver el reto de los PMU sobre los límites en los usuarios que pueden consultar su información.

El programa de visualización, cumple la función de graficar los parámetros eléctricos. Para ello decodifica el frame de datos con base en el frame de configuración, y una vez que tiene los valores de los Fasores, valores analógicos y valores digitales, realiza las operaciones necesarias para calcular las potencias propias y diferencia angular entre subestaciones.

El Programa de guardado, cumple la función de generar un registro de los valores de los PMU. Se trata de un programa basado en el visualizador al que se le agrego la función de insertar y guardar los datos de los PMUs en un libro de Excel.

El Programa de alarma, cumple la función de, en caso de un evento, enviar un correo electrónico con los datos de los parámetros eléctricos. Se trata de un programa basado en el visualizador, al que se le agregaron las funciones de identificar cuando existe una perturbación en el sistema, y evaluar si esta perturbación corresponde a un evento.

En resumen, con las nociones las nociones básicas de eléctrica de potencia y el conocimiento de la norma IEEE C37.118.2-2011 se puede implementar un sistema integral de visualización y monitoreo que facilite el estudio de los parámetros de la red eléctrica, el cual emplea como instrumentos de medición la mayoría de equipos disponibles en el mercado, permitiendo interacciones entre los mismos.

Bibliografía

- [1] KIOSKEA.NET, *TCP/IP*, [página web en línea].
KIOSKEA.NET [citado 01/5/2015].
Disponible en internet:
<http://es.kioskea.net/contents/282-tcp-ip>
- [2] INFORMÁTICA HOY, *QUE ES LA DIRECCIÓN IP*, [página web en línea].
INFORMÁTICA HOY [citado 01/5/2015].
Disponible en internet:
<http://www.informatica-hoy.com.ar/aprender-informatica/Que-es-la-direccion-IP.php>
- [3] INSTITUTO TECNOLÓGICO DE COSTARICA, *Curso LabVIEW6i*, [Recurso digital en línea]
INSTITUTO TECNOLÓGICO DE COSTARICA [citado 01/5/2015].
Disponible en internet:
<http://www.ie.itcr.ac.cr/einteriano/control/Labview/ParaAprender/Curso%20LabVIEW6i.pdf>
- [4] IEEE STANDARDS ASSOCIATION, *C37.118.2-2011 IEEE Standard for Synchrophasor Data Transfer for Power Systems*, [Recurso digital en línea]
IEEE STANDARDS ASSOCIATION [citado 01/5/2015].
Disponible en internet:
<https://standards.ieee.org/findstds/standard/C37.118.2-2011.html>