

**DISEÑO DEL PROTOTIPO DEL MÓDULO PARA LA MONITORIZACIÓN DE  
LAS PERTURBACIONES DE TENSIÓN PARA UNA UNIDAD CONSTRUCTIVA  
EN EL ANÁLISIS DE LOS PARÁMETROS DE CALIDAD DE POTENCIA  
ELÉCTRICA CON BASE EN EL ESTÁNDAR IEC 61000-4-30**

**ANDRÉS MAURICIO SALINAS CALA  
ÓSCAR JAVIER CORREDOR MONTERO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE  
TELECOMUNICACIONES  
BUCARAMANGA**

**2009**

**DISEÑO DEL PROTOTIPO DEL MÓDULO PARA LA MONITORIZACIÓN DE  
LAS PERTURBACIONES DE TENSIÓN PARA UNA UNIDAD CONSTRUCTIVA  
EN EL ANÁLISIS DE LOS PARÁMETROS DE CALIDAD DE POTENCIA  
ELÉCTRICA CON BASE EN EL ESTÁNDAR IEC 61000-4-30**

**ANDRÉS MAURICIO SALINAS CALA**

**Trabajo de grado para optar al título de Ingeniero Electrónico**

**ÓSCAR JAVIER CORREDOR MONTERO**

**Trabajo de grado para optar al título de Ingeniero Electricista**

**Director**

**Profesor Gabriel Ordóñez Plata**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE  
TELECOMUNICACIONES**

**BUCARAMANGA**

**2009**

## **DEDICATORIA**

A nuestros padres, familiares y amigos por brindarnos su apoyo

## AGRADECIMIENTOS

Los autores expresan su agradecimiento a:

**Profesores Gabriel Ordóñez Plata, Cesar Antonio Duarte Gualdrón y Jorge Hernando Ramón Suárez**, por la colaboración y orientación prestada durante el desarrollo del trabajo de grado.

A los integrantes del Seminario de Investigación por sus aportes en los debates y discusiones que se llevaron a cabo durante el desarrollo del seminario.

## CONTENIDO

|   | <b>Pág.</b> |
|---|-------------|
| <b>1. INTRODUCCIÓN</b>  | 1           |
| <b>2. DEFINICIONES DEL ESTÁNDAR IEC 61000-4-30:2003</b>   | 4           |
| 2.1 DEFINICIONES PARA EVENTOS DE ESTADO ESTACIONARIO  | 4           |
| 2.2 DEFINICIONES PARA EVENTOS DE ESTADO TRANSITORIO   | 5           |
| <b>3. ANÁLISIS DE LA REGULACIÓN Y LA NORMATIVIDAD PARA LA<br/>MONITORIZACIÓN DE PERTURBACIONES DE TENSIÓN</b> | 7           |
| 3.1 REGLAMENTACIÓN COLOMBIANA   | 7           |
| 3.1.1 Resolución 024 de 2005 y resolución 016 de 2007 de la Comisión de<br>Regulación de Energía y Gas (CREG) | 8           |
| 3.1.1.1 Estado estacionario   | 8           |
| 3.1.1.2 Estado transitorio  | 9           |
| 3.1.2 Norma Técnica Colombiana -NTC 5001 (2008)   | 11          |
| 3.1.2.1 Estado estacionario   | 11          |
| 3.1.2.2 Estado transitorio  | 12          |
| 3.2 NORMATIVIDAD INTERNACIONAL (IEC 61000-4-30 de 2003)   | 12          |
| 3.2.1 Estándar IEC 61000-4-30 (2003)  | 12          |
| 3.2.1.1 Estado estacionario   | 12          |
| 3.2.1.2 Estado transitorio  | 14          |

|   |           |
|---|-----------|
| <b>4. ALGORITMOS UTILIZADOS PARA LA MONITORIZACIÓN DE<br/>PERTURBACIONES DE TENSIÓN</b> | <b>16</b> |
| 4.1 EFECTOS DE LAS VARIACIONES DE FRECUENCIA EN LA MEDICIÓN<br>DE EVENTOS               | 17        |
| 4.2 MONITORIZACIÓN DE EVENTOS DE ESTADO ESTACIONARIO                                    | 18        |
| 4.2.1 Medida básica   | 18        |
| 4.2.2 Desbalance  | 18        |
| 4.2.2.1 Componente de secuencia negativa (inversa)                                      | 19        |
| 4.2.2.2 Componente de secuencia cero  | 22        |
| 4.2.3 Amplitud de la tensión de alimentación  | 25        |
| 4.2.4 Sobretensión y subtensión   | 27        |
| 4.2.5 Evaluación de la medida   | 29        |
| 4.2.5.1 Agregación de ciento ochenta períodos   | 29        |
| 4.2.5.2 Agregación de diez minutos  | 30        |
| 4.2.5.3 Agregación de dos horas   | 34        |
| 4.3 MONITORIZACIÓN DE EVENTOS DE ESTADO TRANSITORIO                                     | 35        |
| 4.3.1 Medida Básica   | 37        |
| 4.3.2 Huecos de Tensión   | 42        |
| 4.3.2.1 Detección   | 42        |
| 4.3.2.2 Evaluación  | 43        |
| 4.3.2.3 Incertidumbre de la medida  | 45        |

|   |           |
|---|-----------|
| 4.3.2.4 Diagrama de flujo   | 45        |
| 4.3.3 Sobretensión Temporal   | 48        |
| 4.3.3.1 Detección   | 48        |
| 4.3.3.2 Evaluación  | 49        |
| 4.3.3.3 Diagrama de flujo   | 50        |
| 4.3.4 Interrupción de tensión   | 52        |
| 4.3.4.1 Detección   | 52        |
| 4.3.4.2 Diagrama de flujo   | 53        |
| 4.4 INFORME CREG  | 55        |
| 4.4.1 Eventos de estado estacionario  | 55        |
| 4.4.2 Eventos de estado transitorio   | 56        |
| <b>5. VERIFICACIÓN DE LA CONFORMIDAD CON EL ESTÁNDAR<br/>IEC 61000-4-30 (2003)</b>                        | <b>57</b> |
| 5.1 PRUEBAS REALIZADAS AL ALGORITMO DE MEDIDA PARA LA<br>MONITORIZACIÓN DE EVENTOS DE ESTADO ESTACIONARIO | 60        |
| 5.1.1 Análisis de resultados  | 62        |
| 5.1.1.1 Desbalance  | 63        |
| 5.1.1.2 Amplitud de la tensión de alimentación  | 67        |
| 5.1.1.3 Sobretensión y subtensión   | 70        |
| 5.1.2 Análisis con otras señales de prueba  | 71        |
| 5.1.2.1 Ruido   | 71        |
| 5.1.2.2 Cuantificación de la señal  | 75        |

|  |            |
|--|------------|
| 5.2 PRUEBAS REALIZADAS AL ALGORITMO DE MEDIDA PARA LA MONITORIZACIÓN DE EVENTOS DE ESTADO TRANSITORIO  | 76         |
| 5.2.1 Características de las señales sintéticas de prueba  | 77         |
| 5.2.2 Análisis de resultados   | 80         |
| 5.2.2.1 Huecos de tensión  | 80         |
| 5.2.2.2 Sobretensión temporal  | 83         |
| 5.2.2.3 Comparación de errores en la estimación del valor eficaz de la tensión con muestras constantes y muestras variables con falta de sincronismo | 85         |
| 5.2.2.4 Incertidumbre de la duración de la medida  | 87         |
| 5.2.3 Análisis con otras señales de prueba   | 89         |
| <b>6. CONCLUSIONES Y FUTUROS DESARROLLOS</b>   | <b>96</b>  |
| <b>BIBLIOGRAFÍA</b>  | <b>102</b> |
| <b>ANEXOS</b>  | <b>105</b> |

## LISTA DE FIGURAS

|  | <b>Pág.</b> |
|--|-------------|
| Figura 1. Tensión eficaz de doce ciclos  | 18          |
| Figura 2. Esquema para la estimación de la Componente de secuencia negativa            | 21          |
| Figura 3. Esquema para la estimación de la Componente de secuencia cero                | 24          |
| Figura 4. Diagrama de flujo para el cálculo del valor eficaz de doce ciclos            | 25          |
| Figura 5. Esquema para la medición de la amplitud de la tensión de alimentación        | 26          |
| Figura 6. Esquema para la medición de los valores de sobretensión y subtensión         | 28          |
| Figura 7. Diagrama de flujo para la agregación de un tiempo de ciento ochenta períodos | 30          |
| Figura 8. Diagrama de flujo para el tiempo de agregación de diez minutos               | 32          |
| Figura 9. Diagrama de flujo para el tiempo de agregación de dos horas                  | 34          |

|   |    |
|---|----|
| Figura 10. Diagrama de flujo del medidor trifásico de perturbaciones transitorias de Tensión  | 37 |
| Figura 11. Representación de la medida básica para la medición de eventos transitorios  | 38 |
| Figura 12. Diagrama de flujo para la detección de los cruces por cero para la estimación del valor eficaz con un número de muestras fijo                  | 39 |
| Figura 13 Diagrama de flujo para la detección de los cruces por cero para la estimación del valor eficaz de la tensión con un número de muestras variable | 41 |
| Figura 14. Diagrama de flujo de la evaluación de la tensión eficaz  | 42 |
| Figura 15. Umbrales para la detección de un hueco de tensión  | 43 |
| Figura 16. Representación de la evaluación de la medida de un hueco de tensión  | 45 |
| Figura 17. Diagrama de flujo para la detección y evaluación de un hueco de tensión  | 47 |
| Figura 18. Umbrales de detección de una sobretensión temporal   | 48 |
| Figura 19. Representación de la evaluación de la medida de una sobretensión temporal  | 50 |

|   |    |
|---|----|
| Figura 20. Diagrama de flujo para la detección y evaluación de una sobretensión temporal        | 51 |
| Figura 21. Diagrama de flujo para la detección y evaluación de una interrupción de tensión      | 54 |
| Figura 22. Representación del inicio y final de la duración del evento                          | 87 |
| Figura 23. Hueco de tensión ideal con dos cambios en la amplitud de la señal                    | 90 |
| Figura 24. Resultado del algoritmo de muestra variable  | 91 |
| Figura 25. Cambio de fase de la señal producido por un hueco de tensión                         | 92 |
| Figura 26. Resultado del algoritmo de muestra variable al ocurrir un cambio de fase en la señal | 93 |
| Figura 27. Interrupción de tensión  | 94 |
| Figura 28. Resultado del algoritmo de muestra variable  | 94 |

## LISTA DE TABLAS

|   | <b>Pág.</b> |
|---|-------------|
| Tabla 1. Cuadro comparativo de las principales características para la monitorización de perturbaciones de tensión  | 16          |
| Tabla 2. Condiciones de ensayo de incertidumbre para los métodos clase A  | 58          |
| Tabla 3. Margen de variación de las magnitudes de influencia (señales de entrada) para el método clase A  | 58          |
| Tabla 4. Especificaciones de ensayo para el clasificador de flicker   | 59          |
| Tabla 5. Errores en la estimación de la componente de secuencia negativa bajo las condiciones de ensayo 1   | 63          |
| Tabla 6. Error en la estimación de la componente de secuencia negativa bajo las condiciones de ensayo 2   | 63          |
| Tabla 7. Errores en la estimación de la componente de secuencia negativa bajo las condiciones de ensayo 3   | 64          |
| Tabla 8. Errores en la estimación de la componente de secuencia negativa a 60 [Hz] con una frecuencia de muestreo de 10235 [Hz] (2047 muestras en 12 ciclos) en las condiciones de ensayo 3 | 64          |

|  |    |
|--|----|
| Tabla 9. Errores en la estimación de la componente de secuencia negativa a 60 [Hz] con una frecuencia de muestreo de 10245 [Hz] (2049 muestras) bajo las condiciones de ensayo 3 | 65 |
| Tabla 10. Errores en la estimación de la componente de secuencia cero bajo las condiciones de ensayo 1   | 65 |
| Tabla 11. Errores en la estimación de la componente de secuencia cero bajo las condiciones de ensayo 2   | 66 |
| Tabla 12. Errores en la estimación de la componente de secuencia cero bajo las condiciones de ensayo 3   | 66 |
| Tabla 13. Errores en la estimación de la componente de secuencia cero a 60 [Hz] con una frecuencia de muestreo de 10235 [Hz] (2047 muestras) bajo las condiciones de ensayo 3    | 67 |
| Tabla 14. Errores en la estimación de la componente de secuencia cero a 60 [Hz] con una frecuencia de muestreo de 10245 [Hz] (2049 muestras) bajo las condiciones de ensayo 3    | 67 |
| Tabla 15. Errores en la estimación del valor eficaz de la tensión de alimentación bajo las condiciones de ensayo 1   | 68 |
| Tabla 16. Errores en la estimación del valor eficaz de la tensión de alimentación bajo las condiciones de ensayo 2   | 68 |
| Tabla 17. Errores en la estimación del valor eficaz de la tensión de alimentación bajo las condiciones de ensayo 3   | 68 |

|   |    |
|---|----|
| Tabla 18. Errores en la estimación del valor eficaz de la tensión de alimentación a 60 Hz con una frecuencia de muestreo de 10235 [Hz] (2047 muestras) bajo las condiciones de ensayo 3 | 69 |
| Tabla 19. Errores en la estimación del valor eficaz de la tensión de alimentación a 60 Hz con una frecuencia de muestreo de 10245 [Hz] (2049 muestras) bajo las condiciones de ensayo 3 | 69 |
| Tabla 20. Errores en la estimación de los valores de sobretensión y subtensión bajo las condiciones de ensayo 1   | 70 |
| Tabla 21. Errores en la estimación de los valores de sobretensión y subtensión bajo las condiciones de ensayo 2   | 70 |
| Tabla 22. Errores en la estimación de los valores de sobretensión y subtensión bajo las condiciones de ensayo 3   | 71 |
| Tabla 23. Errores máximos en la medida de los eventos de estado estacionario cuando se presenta el fenómeno de ruido blanco   | 72 |
| Tabla 24. Errores máximos en la medida de los eventos de estado estacionario cuando se presenta el fenómeno de ruido rojo   | 73 |
| Tabla 25. Errores máximos en la medida de los eventos de estado estacionario cuando se presenta el fenómeno de ruido azul   | 73 |
| Tabla 26. Errores máximos en la medida de los eventos de estado estacionario cuando se presenta el fenómeno de ruido violeta  | 74 |

|   |    |
|---|----|
| Tabla 27. Errores máximos en la medida de los eventos de estado estacionario cuando se presenta el fenómeno de ruido marrón | 74 |
| Tabla 28. Errores máximos en la medida de los eventos de estado estacionario cuando se presenta el fenómeno de ruido rosado | 75 |
| Tabla 29. Errores máximos por la cuantificación de la señal para la medida de los eventos de estado estacionario            | 76 |
| Tabla 30. Errores en la medición de huecos de tensión bajo las condiciones de ensayo 1                                      | 81 |
| Tabla 31. Errores en la medida de huecos de tensión bajo las condiciones de ensayo 2  | 82 |
| Tabla 32. Errores en la medida de huecos de tensión bajo las condiciones de ensayo 3  | 82 |
| Tabla 33. Errores en la medición de sobretensiones temporales bajo las condiciones de ensayo 1                              | 83 |
| Tabla 34. Errores en la medida de sobretensiones temporales bajo las condiciones de ensayo 2                                | 84 |
| Tabla 35. Errores en la medida de sobretensiones temporales bajo las condiciones de ensayo 3                                | 84 |
| Tabla 36. Errores en la medición del valor eficaz bajo desincronización de frecuencia en las condiciones de ensayo 1        | 85 |

|  |    |
|--|----|
| Tabla 37. Errores en la medición del valor eficaz bajo desincronización de frecuencia en las condiciones de ensayo 2 | 86 |
| Tabla 38. Errores en la medición del valor eficaz bajo desincronización de frecuencia en las condiciones de ensayo 3 | 86 |
| Tabla 39. Errores en la estimación de la duración de un hueco de tensión de duración 0,2164 [s]                      | 88 |
| Tabla 40. Errores en la estimación de la duración de un hueco de tensión de duración 0,3666 [s]                      | 88 |
| Tabla 41. Errores en la estimación de la duración de un hueco de tensión de duración 0,4248 [s]                      | 88 |
| Tabla 42. Errores en la estimación de la duración de un hueco de tensión de duración 0,3082 [s]                      | 89 |

## LISTA DE ANEXOS

|   | <b>Pág.</b> |
|---|-------------|
| <b>ANEXO A. ALGORITMO PARA LA MEDICIÓN DE LAS PERTURBACIONES ESTACIONARIAS DE TENSIÓN EN LENGUAJE C</b> |             |
| a) Algoritmo de agregación de ciento ochenta períodos   | 105         |
| b) Algoritmo de agregación de diez minutos y dos horas  | 109         |
| <b>ANEXO B. ALGORITMO PARA LA MEDICIÓN DE LAS PERTURBACIONES ESTACIONARIAS DE TENSIÓN EN MATLAB</b>     |             |
| a) Algoritmo de agregación de ciento ochenta períodos   | 115         |
| b) Algoritmo de agregación de 10 minutos y dos horas  | 118         |
| <b>ANEXO C. ALGORITMO PARA LA MEDICIÓN DE LAS PERTURBACIONES TRANSITORIAS DE TENSIÓN EN LENGUAJE C</b>  |             |
| c1) Medidor monofásico implementado con muestra fija.   | 124         |
| c2) Medidor monofásico implementado con muestra variable.   | 133         |
| c3) Medidor trifásico implementado con muestra fija.  | 142         |
| c4) Medidor trifásico implementado con muestra variable.  | 157         |
| <b>ANEXO D1. ALGORITMO PARA LA MEDICIÓN DE LAS PERTURBACIONES TRANSITORIAS DE TENSIÓN EN MATLAB.</b>    |             |
| d1) Medidor monofásico implementado con muestra fija.   | 174         |

|   |     |
|---|-----|
| d2) Medidor monofásico implementado con muestra variable. | 186 |
| d3) Medidor trifásico implementado con muestra fija.      | 200 |
| d4) Medidor trifásico implementado con muestra variable.  | 218 |

#### ANEXO D2. GENERADOR DE SEÑALES SINTÉTICAS EN TIEMPO DISCRETO.

|               |     |
|---------------|-----|
| a) Monofásico | 231 |
| b) Trifásico  | 232 |

#### ANEXO D3. GENERADOR DE SEÑALES SINTÉTICAS EN TIEMPO CONTÍNUO.

|               |     |
|---------------|-----|
| a) Monofásico | 235 |
| b) Trifásico  | 236 |

#### ANEXO D4. MEDIDOR RMS DE 1/2 CICLO EN TIEMPO CONTINUO.

|               |     |
|---------------|-----|
| a) Monofásico | 239 |
| b) Trifásico  | 241 |

## RESUMEN

**TÍTULO:** DISEÑO DEL PROTOTIPO DEL MÓDULO PARA LA MONITORIZACIÓN DE LAS PERTURBACIONES DE TENSIÓN PARA UNA UNIDAD CONSTRUCTIVA EN EL ANÁLISIS DE LOS PARÁMETROS DE CALIDAD DE POTENCIA ELÉCTRICA CON BASE EN EL ESTÁNDAR IEC 61000-4-30\*

**AUTORES:** ANDRÉS MAURICIO SALINAS CALA, ÓSCAR JAVIER CORREDOR MONTERO\*\*

**PALABRAS CLAVES:** Calidad de Potencia, huecos de tensión, sobretensiones temporales, interrupciones, desbalance, amplitud de la tensión, sobretensión y subtensión.

### DESCRIPCIÓN:

En este documento se examinan las exigencias establecidas en las resoluciones de la Comisión de Regulación de Energía y Gas (CREG) sobre monitorización de la calidad de la energía eléctrica y como éstas deben tenerse en cuenta en la medición de perturbaciones de tensión tanto de estado estacionario (desbalance, regulación de tensión, subtensiones y sobretensión), como de estado transitorio (huecos de tensión, sobretensión temporal e interrupciones).

El análisis incluye las exigencias establecidas en la norma IEC 61000-4-30, ya que ésta es la base para la realización de los algoritmos de monitorización de la tensión. Se realiza un estudio detallado de la norma con el fin de extraer las características principales en procesos de monitorización de perturbaciones de tensión como son la medida básica de tensión, la detección de un transitorio de tensión y la evaluación e incertidumbre, para generar los algoritmos que describen el comportamiento del proceso y obtener el código en un archivo de texto de Matlab y lenguaje de programación C. Se explica la generación de los códigos con base en los diagramas de flujo del algoritmo de medición de perturbaciones de tensión.

Finalmente, se presenta un análisis de los resultados presentados por los algoritmos de estimación de tensión y se simulan diferentes escenarios para verificar el cumplimiento de las especificaciones de la norma IEC 61000-4-30.

---

\* Trabajo de Grado

\*\* Facultad de ingenierías Físico-mecánicas, Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director:  
Dr. Gabriel Ordóñez Plata

## ABSTRACT

**TITLE:** DESIGN OF THE PROTOTYPE OF THE MODULE FOR THE VOLTAGE DISTURBANCES MEASUREMENTS FOR A CONSTRUCTIVE UNIT IN THE ANALYSIS OF THE PARAMETERS OF THE ELECTRIC POWER QUALITY BASED ON THE STANDARD IEC 61000-4-30\*

**AUTHORS:** ANDRÉS MAURICIO SALINAS CALA, ÓSCAR JAVIER CORREDOR MONTERO\*\*

**KEYWORDS:** power quality, voltage dips, swells, interruption, unbalance, voltage amplexness.

### DESCRIPTION:

In this document the requirements established by the Energy and Gas Regulation Commission (CREG) resolutions on electric power quality monitoring are analyzed. Likewise, it is also discussed how these requirements have to be taken into account in voltage disturbances measurement of stationary events (unbalance, voltage regulation, under-voltage over-voltage), and transient events (sag, swells and interruptions).

The analysis includes those demands established in the standard IEC 61000-4-30 because it is the base document when designing voltage measurement algorithms. This standard is analyzed in depth in order to consider the necessary information for monitoring process like: basic measurement, transient voltage detection, measure evaluation and uncertainty to generate the algorithms description to process and obtain the code in a Matlab and programming language C text archive and explain codes generation based on the flow diagrams of the voltage disturbances measurements algorithms.

Finally, the analyses of the algorithms results were presented and several scenarios were simulated to verify the compliance with the standard IEC 61000-4-30 specifications.

---

\* Degree Work

\*\* Faculty of Engineering Physic-Mechanicals, School of Engineering Electrical Electronics and Telecommunications.

Director: Dr. Gabriel Ordóñez Plata

## 1. INTRODUCCIÓN

Las perturbaciones de tensión son un factor relevante en las pérdidas económicas de las empresas. Por ejemplo, una interrupción del servicio de energía eléctrica en una fábrica de dispositivos semiconductores tiene unos costos elevados, llegando en algunos casos a pérdidas económicas cercanas al millón de dólares.

Los huecos de tensión (*Sag* o *Dip*), las interrupciones de tensión y las sobretensiones temporales (*Swell*) son fenómenos que ocurren a diario en los sistemas de distribución de energía eléctrica, por múltiples causas, que en muchos casos no son ocasionados por el sistema que los sufre.

La monitorización de las variaciones de tensión de alimentación, desbalances, sobretensiones y subtensiones permite hacer un seguimiento de la tensión de suministro y llevar un registro de su comportamiento para contrastarlo con los fenómenos ocurridos y así determinar la causa de la falla.

Para realizar esta monitorización se han expedido normativas nacionales e internacionales que proponen los métodos de medición a seguir, para establecer una metodología normalizada en diferentes países que permita posteriormente realizar estudios y comparaciones.

En Colombia, la Comisión de Regulación de Energía y Gas (CREG) ha expedido dos resoluciones relacionadas con la monitorización de la señal de tensión para que los operadores de red realicen un seguimiento de la calidad de onda que

están entregando a sus usuarios. Estas resoluciones establecen metodologías propuestas por la Comisión Electrotécnica Internacional (IEC) y el Instituto de Ingenieros Electricistas y Electrónicos (IEEE) para el seguimiento de la señal de tensión.

La aplicación adecuada de estas metodologías requiere de un análisis de los procedimientos descritos en las normas internacionales y en este trabajo de grado, específicamente se estudian los propuestos para la monitorización de cambios en la señal de tensión en la norma IEC 61000-4-30.

En el trabajo se realiza un estudio detallado de la norma para considerar la información necesaria en procesos de monitorización como son: la medida básica de tensión, la detección de un transitorio de tensión, la evaluación e incertidumbre de la medida. A partir del análisis realizado se diseñan los algoritmos de estimación de tensión y se simulan diferentes escenarios para verificar el cumplimiento de las especificaciones de la norma.

La organización del documento se describe a continuación:

En el capítulo dos se presentan algunas definiciones básicas que permiten comprender el desarrollo del algoritmo propuesto para la monitorización de perturbaciones de la señal de tensión de estado transitorio así como sus variaciones en estado estacionario.

El estándar IEC 61000-4-30, las resoluciones de la CREG y la Norma Técnica Colombiana NTC 5001 se analizan en el capítulo tres.

El capítulo cuatro hace un completo desarrollo de los algoritmos para la medición de eventos estacionarios y transitorios, explica el diseño de los algoritmos en

Matlab y lenguaje C a través de diagramas de flujo, y de qué manera se logró establecer la evaluación de la medida de cada evento.

El quinto capítulo hace referencia a la verificación de los algoritmos, esto se realiza con base en las pruebas descritas en el capítulo seis del estándar IEC 61000-4-30. Además, se realiza una serie de pruebas con señales adicionales a las propuestas por el estándar IEC 61000-4-30 para contrastar el efecto de dichas perturbaciones en la medida de los algoritmos.

Finalmente, en el capítulo seis se presentan las conclusiones y recomendaciones para trabajos futuros.

Como complemento se anexan los programas realizados en lenguaje C que permiten la implementación de los algoritmos que monitorizan las perturbaciones de la señal de tensión.

## 2. DEFINICIONES DEL ESTÁNDAR IEC 61000-4-30:2003

A continuación se presentan una serie de definiciones de parámetros propuestos en la norma IEC 61000-4-30 para la monitorización de los eventos de estado estacionario y transitorio de la señal de tensión.

### 2.1 DEFINICIONES PARA EVENTOS DE ESTADO ESTACIONARIO

**Tensión eficaz de doce ciclos ( $U_{ef}$ ).** Se define como la raíz cuadrada de la media aritmética de los cuadrados de los valores instantáneos de una magnitud de tensión durante doce ciclos de la señal y con un ancho de banda determinado [IEC 61000-4-30, 2003].

**Valor Alto ( $U_{alta}$ ).** Es el valor absoluto de la diferencia entre el valor medido y el valor nominal de un parámetro, únicamente cuando el valor medido del parámetro es superior al valor nominal [IEC 61000-4-30, 2003]. Otros términos equivalentes utilizados son: tensión alta y sobretensión.

**Valor bajo ( $U_{baja}$ ).** Es el valor absoluto de la diferencia entre el valor medido y el valor nominal de un parámetro, únicamente cuando el valor medido es inferior al valor nominal [IEC 61000-4-30, 2003]. Otros términos equivalentes utilizados son: tensión baja y subtensión.

**Amplitud de la tensión.** Corresponde al valor de la tensión eficaz de doce ciclos. [IEC 61000-4-30, 2003].

**Tensión de secuencia positiva (directa).** Se define como la componente simétrica obtenida a partir de la matriz de transformación de *Fortescue*, formada por un conjunto de tensiones trifásicas balanceadas de secuencia positiva.

**Tensión de secuencia negativa (inversa).** Corresponde a la componente simétrica obtenida a partir de la matriz de transformación de *Fortescue*, formada por un conjunto de tensiones trifásicas balanceadas de secuencia negativa.

**Tensión de secuencia cero (homopolar).** Es la componente simétrica obtenida a partir de la matriz de *Fortescue*, formada por un conjunto de tensiones trifásicas que se encuentran en fase.

**Componente de secuencia negativa ( $U_2$ ).** Es la relación porcentual entre la tensión inversa y la tensión directa [IEC 61000-4-30, 2003]. En la resolución 016 de 2007 de la CREG, se define como la relación entre la tensión de secuencia negativa y la tensión de secuencia positiva.

**Componente de secuencia cero ( $U_0$ ).** Corresponde a la relación porcentual existente entre la tensión homopolar y la tensión directa [IEC 61000-4-30, 2003].

**Agregación de medidas.** Las agregaciones de las medidas se obtienen como la raíz cuadrada de la media aritmética de los cuadrados de los valores de entrada [IEC 61000-4-30, 2003].

## 2.2 DEFINICIONES PARA EVENTOS DE ESTADO TRANSITORIO

**Tensión eficaz refrescada cada medio ciclo ( $U_{ef(1/2)}$ ).** Corresponde al valor de la tensión eficaz estimado en un periodo que comienza en el cruce por cero de la componente fundamental, y refrescada en cada semiperiodo [IEC 61000-4-30, 2003]. Esta es la medida básica para la medición de eventos transitorios.

**Tensión de entrada declarada ( $U_{din}$ ).** Se define en función de la tensión de la red o es la tensión definida por el usuario o distribuidor. Este valor es utilizado para definir los umbrales que permitirán detectar los eventos transitorios.

**Umbral de tensión.** Valor de tensión que permite determinar el inicio y fin de un evento. Éste umbral puede ser variable o constante y es a su vez un porcentaje de la referencia dada por el usuario u operador de red.

**Tensión de referencia deslizante ( $U_{rd}$ ).** Se define en función de la tensión eficaz de doce ciclos ( $U_{ef}$ ) y de la tensión de referencia ( $U_{rd}$ ) anterior (ver Ecuación 1). Este valor es utilizado para definir los umbrales de detección de los eventos transitorios.

$$U_{rd(n)} = 0.9976 * U_{rd(n-1)} + 0.0033U_{ef} \quad (1)$$

**Datos marcados.** Cuando ocurre un evento transitorio, éste influye sustancialmente en la estimación de los parámetros de estado estacionario, por tanto ellos se deben marcar para que el usuario conozca que son resultados que pueden diferir sustancialmente con los estimados en otras circunstancias.

**Histéresis.** Es la diferencia de amplitud entre los umbrales de entrada y salida [IEC 61000-4-30, 2003].

### **3. ANÁLISIS DE LA REGULACIÓN Y LA NORMATIVIDAD PARA LA MONITORIZACIÓN DE PERTURBACIONES DE TENSIÓN**

Este capítulo analiza las resoluciones de la CREG y el estándar [IEC 61000-4-30, 2003] con relación a la medición de los eventos transitorios (huecos, sobretensiones temporales e interrupciones) y estacionarios (desbalance, amplitud de la tensión, sobretensión y subtensión).

#### **3.1 REGLAMENTACIÓN COLOMBIANA**

A finales de la década de los 90, la CREG como parte de sus atribuciones legales expidió el reglamento de distribución de energía eléctrica [CREG 070, 1998] algunas pautas con el fin de mejorar la calidad de la potencia suministrada a los usuarios a través de cinco indicadores:

- Frecuencia y tensión.
- Contenido de armónicos de las ondas de tensión y corriente.
- Flicker.
- Factor de potencia.
- Transitorios electromagnéticos rápidos y fluctuaciones de tensión.

Posteriormente se expidieron las resoluciones CREG 024 de 2005 y CREG 016 de 2007 donde se establecen los parámetros principales que debe medir la unidad constructiva. En la resolución [CREG 024, 2005] se establecen las pautas que debe cumplir la unidad constructiva en cuanto a medición y almacenamiento de eventos, además de presentar un anexo con la modificación a los numerales 6.2.1 y 6.2.2 de la resolución [CREG 070, 1998]. En numeral 6.2.1 se modifica los estándares de calidad de potencia suministrada referente a los rangos de

variación de la frecuencia, magnitud estacionaria de la tensión y los límites máximos para la distorsión armónica de la onda de tensión. En el numeral 6.2.2 se aprueba la posibilidad para el operador de red de instalar equipos auxiliares cerca de los equipos de medida de los usuarios con el fin de detectar la fuente de distorsiones o fluctuaciones en el sistema.

La resolución [CREG 016, 2007] modifica la resolución [CREG 024, 2005] en los artículos 3º, 4º y los literales b), c) y g) del artículo 5º. En el artículo 3º, se expone la modificación a las características mínimas que deben tener los equipos de medición, entre las cuales sobresale la incorporación del desempeño clase A para la estimación del desbalance, huecos y sobretensiones. En el artículo 4º se presenta un cambio notable en el plan para la instalación de registro donde se obliga a los operadores de red a poder realizar mediciones en el 100% de las barras de las subestaciones y en los circuitos de tensión 4, 3 y 2, ya que en [CREG 024, 2005] este porcentaje de cobertura solo se aplicaba a las barras de las subestaciones. Finalmente, en el artículo 5º se separa el efecto del Pst en el almacenamiento de las interrupciones y las fluctuaciones estacionarias de tensión.

### **3.1.1 Resoluciones 024 de 2005 y 016 de 2007 de la Comisión de Regulación de Energía y Gas (CREG).**

**3.1.1.1 Estado Estacionario.** En la resolución 016 de 2007 de la Comisión de Regulación de Energía y Gas (CREG), se establece que la monitorización de las perturbaciones de tensión de estado estacionario realizada por parte de los equipos de calidad debe incluir los siguientes parámetros:

- “Medir la relación de voltaje de secuencia negativa y el voltaje de secuencia positiva ( $V(2)/V(1)$ ) para el barraje, con desempeño clase A”. [CREG 016, 2007]

- “Medir la desviación estacionaria de la tensión eficaz (duración superior a un minuto) por debajo del 90% y por encima del 110% de la tensión nominal.” [CREG 016, 2007]

Comparando las definiciones presentadas en las Resoluciones [CREG 024, 2005] y [CREG 016, 2007], con lo establecido en [IEC 61000-4-30, 2003], se encuentran algunas inconsistencias a saber:

La resolución 016 de 2007 indica que el equipo de Calidad con exigencia clase A debe medir únicamente la relación entre la tensión de secuencia negativa y la de secuencia positiva, definida como componente inversa, mientras que el estándar IEC 61000-4-30, expresa que se debe realizar la medición tanto de la componentes inversa como de la componente homopolar para verificar la conformidad con las normas y así certificar la exigencia de la clase A.

Por otro lado, la definición de sobretensión y subtensión (valor alto, valor bajo) queda abierta a interpretación, debido a que no especifica la medida básica para el valor eficaz de estado estacionario, ni el método de medida a utilizar. Por lo tanto, se deja a criterio del diseñador de la unidad constructiva la forma de medida del evento y la clase de exactitud de la misma, ya que ésta no se establece en la resolución.

**3.1.1.2 Estado transitorio.** El artículo 2º de la resolución 024 de 2005 establece los parámetros que deberá medir la unidad constructiva. Para el presente informe dichos parámetros son:

- Medir hundimientos y picos de acuerdo con el Estándar IEC 61000-4-30 (2003-02) con desempeño clase A. [CREG 024, 2005]

- Medir la continuidad del suministro (frecuencia y duración de las interrupciones superiores a un minuto). [CREG 024, 2005]

Además en el literal c) del artículo 4º se expresa el almacenamiento y medición de interrupciones, así:

“Las discontinuidades en la prestación del servicio superiores a un minuto y con tensión menor al 10% de la tensión nominal, se deberán almacenar en forma separada. Se dejará constancia de la existencia de éstas en los registros de Pst”. [CREG 024, 2005].

Queda claro como se debe realizar la medición de hundimientos y picos y la clase de exactitud de la medida. Sin embargo, la interrupción se deja abierta a cualquier método de medida (se puede hablar de  $U_{ef(1/2)}$ ,  $U_{avg}$  o  $U_{ef}$  entre otros,) ya que no especifican si toman una interrupción como un caso especial de un hueco o es un caso aparte, además no se establece la precisión de la medida (Clase A o B).

En un futuro se puede registrar inconformidad por parte de los operadores de red al determinar que ocurrió, un hueco o una interrupción o los dos eventos al mismo tiempo.

Dos parámetros importantes para la detección de los eventos de estado transitorio son los umbrales y la histéresis. En la resolución [CREG 024, 2005] no se establecen dichas cantidades. Los umbrales encontrados son asignados a la detección de la desviación estacionaria de la tensión eficaz (numeral 6.2.1 del anexo 1 de [CREG 024, 2005]). A pesar de la confusión entre el significado de eventos transitorios y estacionarios, se adoptaron los umbrales para la detección de la desviación estacionaria en la detección de los eventos transitorios (huecos, sobretensiones temporales e interrupciones de tensión)

El único evento con umbral asignado es la interrupción de tensión. Sin embargo, no está asignado con base en [IEC 61000-4-30, 2003], el umbral es del 10% de la tensión nominal.

En el caso de la histéresis, la CREG no hace algún comentario al respecto por tanto se asume como 0%.

### **3.1.2 Norma Técnica Colombiana - NTC 5001 (2008)**

**3.1.2.1 Estado estacionario.** En la norma Técnica Colombiana los eventos de estado estacionario se encuentran definidos como eventos de larga duración, encontrándose entre ellos el desbalance, las variaciones de tensión de estado estable (Amplitud de la tensión), y los valores de sobretensión y subtensión. La estimación del desbalance se realiza con base en la componente de secuencia negativa (relación entre la tensión de secuencia negativa y positiva), pero no se hace algún comentario acerca de la medida de desbalance a partir de la componente de secuencia cero (homopolar).

La evaluación de la medida del desbalance y las variaciones de tensión se realiza mediante la agregación de medida de diez minutos. Sin embargo, la definición de agregación presentada en el documento es inadecuada, ya de acuerdo con [IEC 61000-4-30, 2003], los intervalos de doce ciclos comprendidos en la ventana de diez minutos no son constantes, se encuentran en función del reloj de tiempo absoluto, y no a partir de una medida de frecuencia.

Para los valores de sobretensión y subtensión de estado estacionario, la [NTC 5001, 2008] no establece la medida básica para la monitorización de estos eventos, quedando abierta a los requerimientos del diseñador del equipo de medida. Además, la evaluación de medida se realiza a partir de la duración y la

amplitud de la sobretensión y subtensión, con lo cual se difiere totalmente de lo consignado en [IEC 61000-4-30, 2003], donde este proceso se debe efectuar a partir de los intervalos de agregación de ciento ochenta periodos, diez minutos y horas.

**3.2.1.2 Estado transitorio.** En la [NTC 5001, 2008] se encuentran definidos los umbrales para la monitorización de eventos de estado transitorio (huecos, sobretensiones temporales, interrupciones). En los sistemas monofásicos, para los huecos de tensión el umbral es entre el (10%-90%) de  $U_{din}$ , para las sobretensiones por encima del 110%, y en el caso de las interrupciones menor al 10% de  $U_{din}$ . Por lo cual, se concluye que la [NTC 5001, 2008] considera el hueco de tensión y la interrupción como dos eventos independientes. No se hace ningún comentario sobre los valores de histéresis, ni se menciona como se debe realizar la detección y evaluación de estos eventos transitorios para las redes trifásicas.

## **3.2 NORMATIVIDAD INTERNACIONAL**

**3.2.1 ESTÁNDAR IEC 61000-4-30 (2003).** Este documento fue aprobado por el Comité Europeo de Normalización Electrotécnica (CENELEC) con el fin adoptar los métodos de medida de la calidad de suministro a todos los países miembros de esta organización. La CREG decidió aplicar una parte de esta norma en Colombia para mejorar y controlar la calidad de la energía a los usuarios.

**3.2.1.1 Estado Estacionario.** Los eventos de estado estacionario estudiados en este documento son: Amplitud de la tensión, desbalance de tensiones en sistemas trifásicos, sobretensión y subtensión, cuya unidad básica de medida es el valor eficaz de la tensión cada doce ciclos ( $U_{ef}$ ), mientras que la evaluación de la

medida se efectúa a partir de los intervalos de agregación de ciento ochenta períodos, diez minutos y dos horas.

**Valor eficaz de la tensión cada doce ciclos.** De acuerdo con [IEC 61000-4-30, 2003] el valor eficaz de la tensión cada doce ciclos debe estimarse en intervalos que deben ser contiguos y no deben solaparse. Sin embargo, no se especifica si el inicio de las medidas debe ser en el instante de encender el equipo de medida o al primer cruce por cero de la señal de tensión. Para el diseño del algoritmo de monitorización propuesto en este trabajo se ha optado por definir el comienzo de los intervalos una vez sea encendido el equipo con el fin de considerar la información de la mayor cantidad de muestras durante el proceso.

**Agregación de ciento ochenta períodos.** “La agregación de ciento ochenta períodos se realiza a partir de quince intervalos de doce ciclos” [IEC 61000-4-30, 2003]. Se puede presentar una confusión en la agregación de medida de 180 períodos denominada igualmente como agregación de 3 segundos. Sin embargo, el uso del término “3 segundos” es inadecuado ya que dependiendo del valor de la frecuencia del sistema, la duración del intervalo de ciento ochenta períodos de la señal puede ser superior o inferior a 3 segundos. En [Bollen, 2006] se explica con mayor detalle esta ambigüedad. Por consiguiente, en este documento se adopta el uso del término agregación de ciento ochenta períodos.

Además, en [IEC 61000-4-30, 2003] no se establece una relación clara entre la agregación de 180 períodos y la agregación de medidas de diez minutos y dos horas, ya que la agregación de diez minutos se realiza a partir de la medida básica de doce ciclos, y la de dos horas mediante los valores de diez minutos. También es importante señalar que en [IEC 61000-4-30, 2003] no se asigna una referencia de tiempo para cada una de las medidas de ciento ochenta períodos, lo que dificulta definir el instante de inicio y final del primer intervalo de doce ciclos requeridos para la agregación.

Por lo tanto, en este trabajo se ha establecido que el inicio de la primera ventana de doce ciclos para la agregación coincida con el encendido del equipo de medida para la monitorización de los eventos, mientras que el final se obtiene a partir de un número fijo de muestras que se encuentran comprendidas en la ventana de observación, considerando una correcta sincronización de la frecuencia de muestreo con la frecuencia del sistema.

**Agregación de diez minutos.** El inicio y final de los intervalos de diez minutos se encuentran determinados por el reloj de tiempo absoluto (Por ejemplo: 12:00, 12:10, 12:20). Si la última ventana de doce ciclos se solapa con el límite del reloj absoluto, esta ventana se debe incluir para la agregación de diez minutos, por lo cual el número de intervalos de doce ciclos comprendidos en la ventana de agregación no es un valor fijo.

Además, se debe tener en cuenta que los valores de doce ciclos son diferentes para la agregación de ciento ochenta períodos y diez minutos, ya que las ventanas de doce ciclos comprendidas entre el encendido del equipo de medida y el primer intervalo del reloj de tiempo absoluto no deben incluirse en la agregación de diez minutos. Por consiguiente, se requiere que el algoritmo de monitorización de eventos de estado estacionario sea ejecutado simultáneamente dos veces.

**Agregación 2 horas.** “Los datos del intervalo de dos horas se deben agregar a partir de doce intervalos de diez minutos” [IEC 61000-4-30, 2003].

**3.2.1.2 Estado Transitorio.** El estándar deja abierto las opciones de generación de las señales para comprobar las incertidumbres planteadas en el documento. Por lo tanto, se diseñaron señales sintéticas con los datos suministrados por la tabla 3 del capítulo 6 de [IEC 61000-4-30, 2003] con el fin de determinar si el algoritmo cumple o no con las incertidumbres establecidas. Al generar estas

señales no se tiene certeza que las características de la señal estuvieran en ella, por ejemplo, si la amplitud de los armónicos es de una cantidad determinada, al igual que los interarmónicos y Flicker, ¿se tiene la seguridad de que dichas características se encuentren en la señal sin que se perturben entre sí? No se logró una solución a este problema porque no fue posible determinar la influencia de los diferentes eventos en la señal de prueba manteniendo las características especificadas en la tabla 3 del capítulo seis del estándar [IEC 61000-4-30, 2003].

¿Cómo se debe medir un hueco de tensión cuando la señal es de cero voltios?, esta inquietud generó inconvenientes en el diseño del algoritmo, debido a que la medida básica es dependiente de la detección de cruces por cero de la señal fundamental. Si la tensión de la red es cero volts, entonces ¿Cómo se detectan los cruces por cero? En este sentido el estándar no hace referencia a esta situación, para corregir este problema se decidió refrescar la medida básica medio periodo de forma arbitraria mientras se mantenga la tensión de la red a cero volts, puesto que no puede detectar el cruce por cero.

#### 4. ALGORITMOS UTILIZADOS PARA LA MONITORIZACIÓN DE PERTURBACIONES DE TENSIÓN

Los eventos de las perturbaciones de tensión que se deben monitorizar, están propuestas en el estándar [IEC 6100-4-30, 2003]. Estos eventos se han dividido en dos partes: unos para estado estacionario y otros para estado transitorio. Los eventos de estado estacionario son: La amplitud de la tensión de alimentación, el desbalance de la tensión de la red y los valores de subtensión y sobretensión, mientras que los eventos de estado transitorio son: La interrupción de tensión, hueco (hundimiento) de tensión y la sobretensión temporal. La tabla 1 presenta una descripción general de los eventos con algunas características que se detallarán posteriormente. Para la monitorización de todos los eventos, el parámetro utilizado es el valor eficaz de la tensión y se diferencian en el tiempo en el cual debe refrescarse el valor de este parámetro.

Tabla 1. Cuadro comparativo de las principales características para la monitorización de perturbaciones de tensión

|                         | Evento estacionario  | Evento transitorio  |
|-------------------------|--|---|
| Eventos medidos         | Amplitud de la tensión de alimentación<br>Desbalance de la tensión de la red<br>Subtensión y<br>Sobretensión | Interrupción de Tensión<br>Huecos de Tensión<br>Sobretensión Temporal |
| Medida Básica           | Tensión eficaz de doce ciclos  | Tensión eficaz refrescada cada semiperiodo                            |
| Evaluación de la medida | Algoritmos de agregación   | Duración del evento<br>Amplitud del evento                            |
| Incertidumbre           | Medida del evento  | Medida y duración del evento  |

El algoritmo para la monitorización de perturbaciones de tensión que se explicará a continuación, fue desarrollado e implementado en la herramienta de Software MATLAB y lenguaje de programación C. El algoritmo de eventos de estado

estacionario se encuentra consignado en los anexos A y B, mientras que el algoritmo de eventos de estado transitorio en los anexos C y D.

#### **4.1 EFECTOS DE LAS VARIACIONES DE FRECUENCIA EN LA MEDICIÓN DE EVENTOS**

El algoritmo para la monitorización de eventos de estado estacionario requiere una sincronización entre la frecuencia de la red y la frecuencia de muestreo con la que se adquieren los datos. Esto se realiza con el fin de tener un número constante de muestras (2048) en cada una de las ventanas de observación de doce ciclos y a partir de cada intervalo realizar el cálculo de la medida básica (tensión eficaz de doce ciclos) adecuadamente.

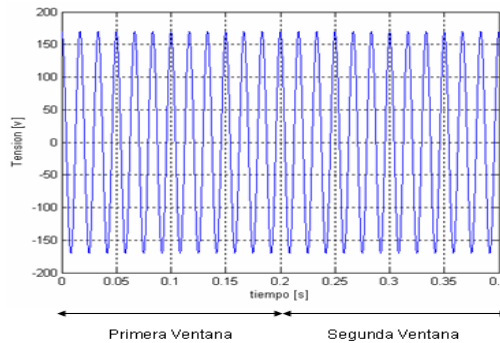
En caso de tener variaciones en la frecuencia del sistema y que ésta no pueda sincronizarse con la frecuencia de muestreo, no se tendrán la cantidad de muestras deseada en la ventana de doce ciclos (2048), produciéndose errores en la medida básica y en la evaluación de los eventos; los cuales dependiendo del grado de desincronización pueden llegar a ser mayores que la incertidumbre permisible para el cumplimiento de la exigencia clase A en la medición del valor eficaz de la tensión.

Para los eventos transitorios la ventana de observación comprende un periodo de la señal, y la frecuencia de muestreo determinada por la cantidad de muestras por ciclo. En la mayoría de las aplicaciones de la monitorización se realiza el muestreo con 256 muestras por ciclo para lo cual se requiere sincronizar el proceso de muestreo con la frecuencia del sistema para reducir la incertidumbre en la estimación del valor eficaz. En el capítulo 5 se presentan los errores máximos cuando no se sincroniza la frecuencia de muestreo con la del sistema.

## 4.2 MONITORIZACIÓN DE EVENTOS DE ESTADO ESTACIONARIO

**4.2.1 Medida Básica.** La medida básica para la monitorización de eventos de estado estacionario (desbalance, amplitud de la tensión, sobretensión y subtensión) es el valor de la tensión eficaz evaluada en doce ciclos, definida en la sección 2.1.1; teniendo en cuenta que en el caso de desbalance en sistemas trifásicos, el valor eficaz de la tensión corresponde al de la componente fundamental. Los intervalos de doce ciclos deben ser contiguos, y no deben solaparse como se aprecia en la Figura 1.

Figura 1. Tensión eficaz de doce ciclos.



Fuente: Autores

**4.2.2 Desbalance.** De acuerdo con el estándar IEC 61000-4-30 la estimación del desbalance se efectúa a partir de la componente de secuencia negativa (inversa) y y cero (homopolar), definida en las secciones 2.1.8 y 2.1.9 respectivamente. Sin embargo, la implementación directa a partir de las componentes simétricas involucra algunos inconvenientes debido a que la transformación de *Fortescue* emplea operaciones entre números reales y complejos e introduce la necesidad de realizar desfases de las ondas de tensión, lo cual puede llegar a generar errores y dificultades para la monitorización de los eventos.

Por las razones mencionadas anteriormente, se ha optado por implementar un algoritmo alternativo presentado así mismo en el estándar IEC 61000-4-30 sección 5.71, mostrado en las ecuaciones (5) y (6).

Para la estimación de la componente de secuencia cero, la norma no propone una ecuación adicional para realizar dicho proceso. Con el fin de solucionar los inconvenientes mencionados anteriormente, se ha propuesto un algoritmo de medida alternativo para la monitorización del desbalance que consiste en el cálculo de las tensiones de secuencia cero y secuencia positiva. (Ver ecuaciones 7 y 8) [Baggini, 2008].

**4.2.2.1 Componente de secuencia negativa (inversa).** La medición de la componente inversa se realiza a partir de los procesos de la obtención de las tensiones de línea, atenuación de armónicos y el cálculo de los parámetros beta ( $\beta$ ) y  $u_2$ .

**Obtención de las tensiones de línea.** El proceso de calcular las tensiones de línea a partir de las tensiones de fase, se realiza empleando la ley de tensiones que Kirchoff en una red trifásica, como se muestra en las ecuaciones (2), (3) y (4):

$$U_{ab} = U_{an} - U_{bn} \quad (2)$$

$$U_{bc} = U_{bn} - U_{cn} \quad (3)$$

$$U_{ca} = U_{cn} - U_{an} \quad (4)$$

Es importante señalar que si se tiene acceso a las tensiones de línea, se puede omitir este proceso.

**Atenuación de armónicos.** En el estándar [IEC 61000-4-30,2003] se proponen dos métodos para la atenuación de armónicos: con un filtro pasabajas o mediante

el algoritmo de la transformada discreta de Fourier (DFT). Debido a que en la respuesta natural del filtro pasabajas se produce un error en la primera medida y además éste permite el paso de subarmónicos (ondas con una frecuencia inferior a 60 Hz) se ha seleccionado el algoritmo de la DFT.

La implementación del algoritmo se realiza a través de dos procesos. El primero se encarga de agrupar bloques de 2048 muestras y en cada proceso toma el grupo siguiente existente durante el tiempo de simulación de la señal de entrada.

El segundo calcula la transformada discreta de Fourier (DFT) a cada grupo de 2048 muestras con el fin de extraer de ella la magnitud del valor eficaz de doce ciclos de la componente fundamental de las tensiones de línea escalada un factor igual a  $\sqrt{2}/2048$ .

**Parámetros beta y U2 (componente de secuencia negativa).** La evaluación de los parámetros beta ( $\beta$ ) y U2 , se realiza a través de las expresiones presentadas en el estándar IEC 61000-4-30 sección 5.71, mostradas en las ecuaciones (5) y (6):

$$\beta = \frac{U_{abfund}^4 + U_{bcfund}^4 + U_{cafund}^4}{(U_{abfund}^2 + U_{bcfund}^2 + U_{cafund}^2)^2} \quad (5)$$

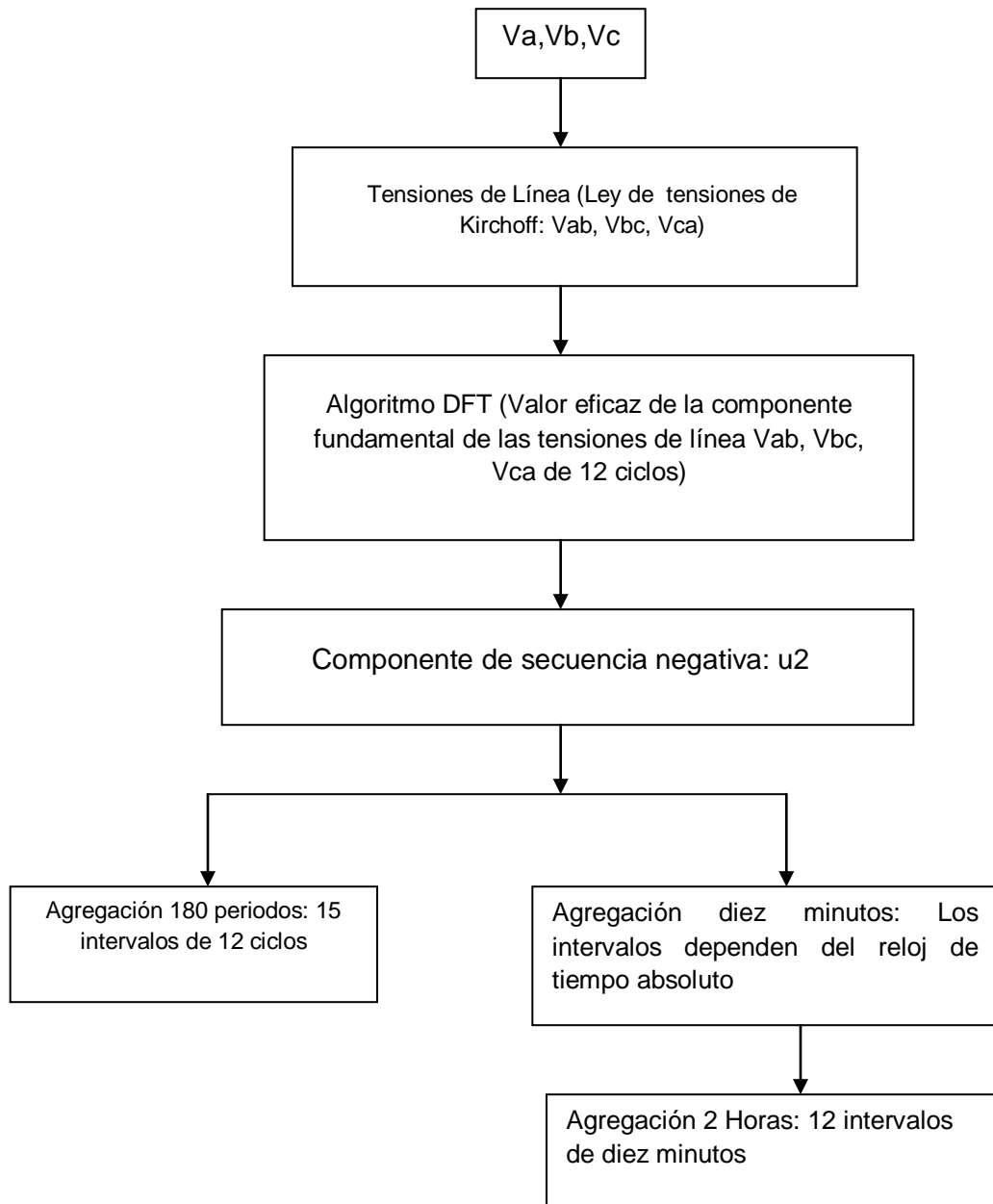
$$U_2 = \sqrt{\frac{1 - \sqrt{3 - 6 * \beta}}{1 + \sqrt{3 - 6 * \beta}}} * 100 \% \quad (6)$$

Donde:

$U_{abfund}, U_{bcfund}, U_{cafund}$  Son las tensiones eficaces de la componente fundamental de las tensiones de línea.

$\beta$  es un factor que depende de la magnitud de las componentes fundamentales de las tensiones de línea. En la figura 2 se muestra el diagrama de flujo de la estimación de la componente de secuencia negativa.

Figura 2. Esquema para la estimación de la componente de secuencia negativa.



Fuente: Autores

#### 4.2.2.2 Componente de secuencia cero.

**Tensión de secuencia cero.** La evaluación de la tensión de secuencia cero se realiza mediante la suma de las tensiones de fase escaladas por un factor igual a  $\frac{1}{\sqrt{3}}$ . A la función resultante, se le calcula el valor eficaz de la componente fundamental utilizando la Transformada Discreta de Fourier (DFT) de forma similar como se ha expuesto en la sección 4.2.2.1 Atenuación de armónicos para la componente inversa, como se aprecia en la ecuación (7):

$$tensionsecuenciacer0 = U_{effund} \left( \frac{1}{\sqrt{3}} * (U_{an} + U_{bn} + U_{cn}) \right) \quad (7)$$

Siendo:

$U_{an}$ ,  $U_{bn}$ ,  $U_{cn}$  las señales muestreadas para las tensiones de fase.

**Tensión de secuencia positiva.** La medida de la tensión de secuencia positiva se realiza a través de la implementación de la ecuación (8) [Baggini, 2008], cuyas variables de entrada ( $U_{abfund}$ ,  $U_{bcfund}$ ,  $U_{cafund}$ ) corresponden a la tensión eficaz de doce ciclos de la componente fundamental de las tensiones de línea obtenida a través del proceso de atenuación de armónicos descrito para la componente de secuencia negativa (inversa).

$$tensionsecpositiva = \sqrt{\frac{1}{12} \left[ K + \left( \sqrt{3} * U_{abfund} + \sqrt{4U_{bcfund}^2 + (U_{abfund} - K)^2} \right)^2 \right]} \quad (8)$$
$$K = \left( \frac{U_{abfund}^2 - U_{cafund}^2}{U_{abfund}} \right)^2$$

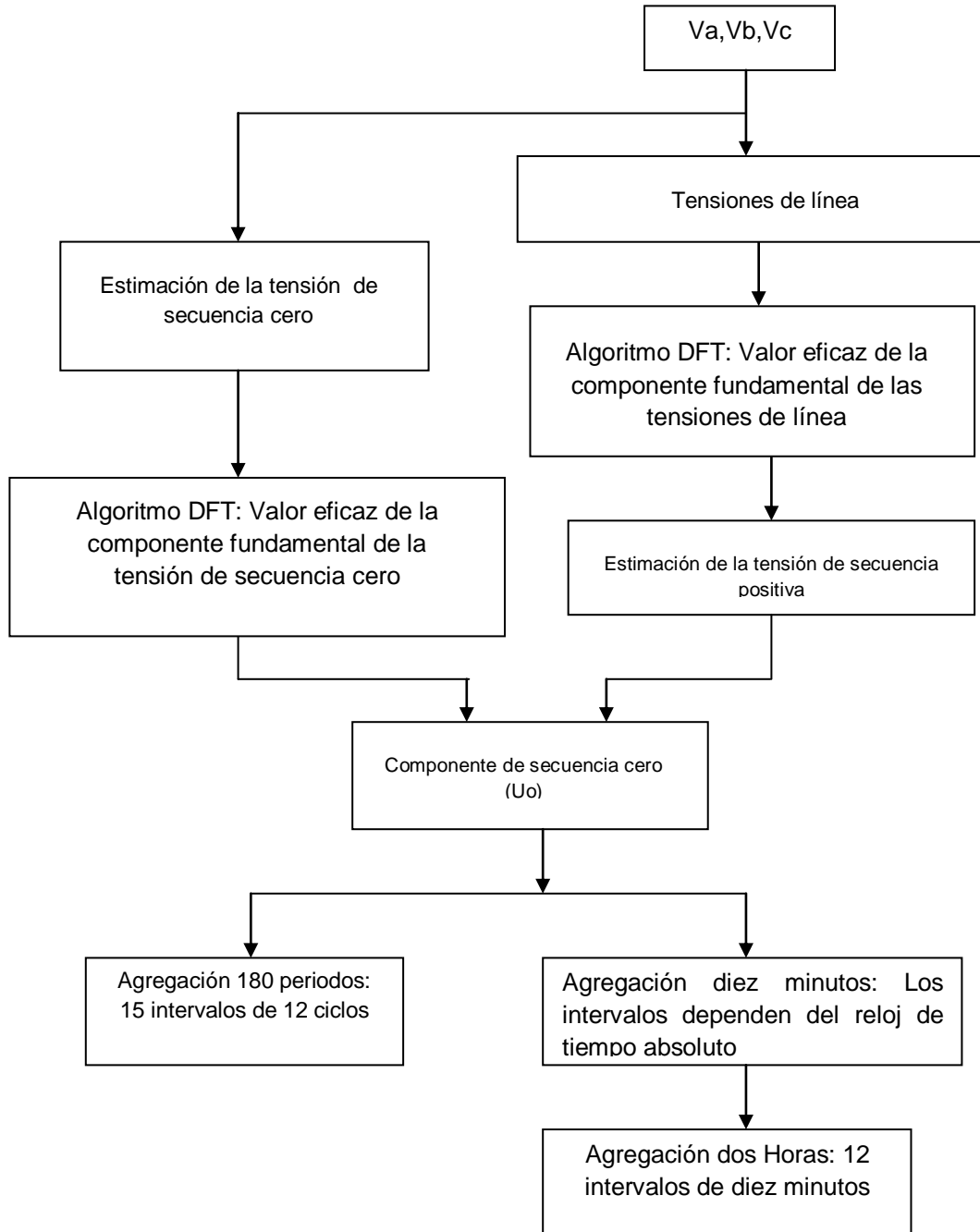
Donde:

$U_{abfund}$ ,  $U_{bcfund}$ ,  $U_{cafund}$  son las tensiones eficaces de las componentes

fundamentales de las tensiones de línea.

**Parámetro  $U_0$  (Componente de secuencia cero).** La estimación de la componente de secuencia cero (homopolar) es igual a la relación en porcentaje entre la tensión de secuencia cero y la tensión de secuencia positiva determinadas a partir de las ecuaciones (7) y (8) respectivamente. En la Figura 3 se muestra el diagrama de flujo para el cálculo de la componente de secuencia cero.

Figura 3. Esquema para la estimación de la componente de secuencia cero.

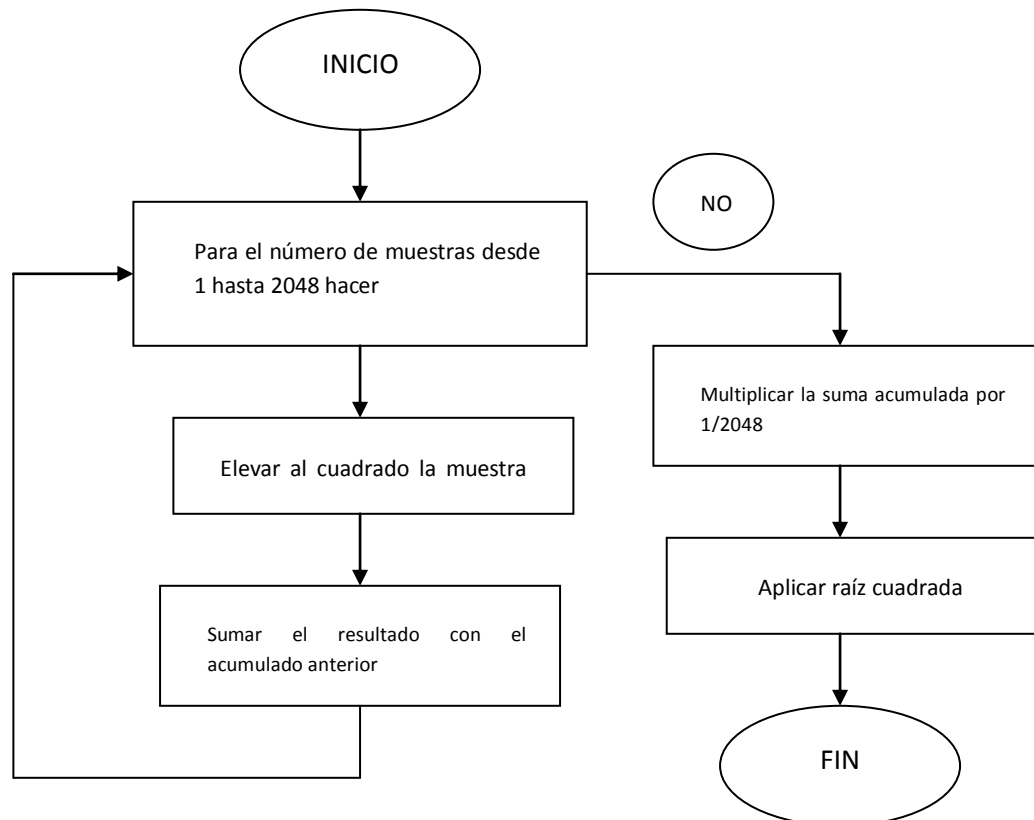


Fuente: Autores

**4.2.3 Amplitud de la tensión de alimentación.** La medida del valor eficaz de la tensión de alimentación se realiza cada doce ciclos, incluyendo todas las perturbaciones que pueden afectar a la señal trifásica, como los armónicos e interarmónicos.

La lógica de procesamiento de los datos para el cálculo del valor eficaz de doce ciclos se presenta en la Figura 4. Como se explicó en la sección 4.1, el algoritmo considera una adecuada sincronización de la frecuencia del sistema con la frecuencia de muestreo con el fin de cada ventana de doce ciclos tenga un número fijo de muestras, que para este caso son 2048.

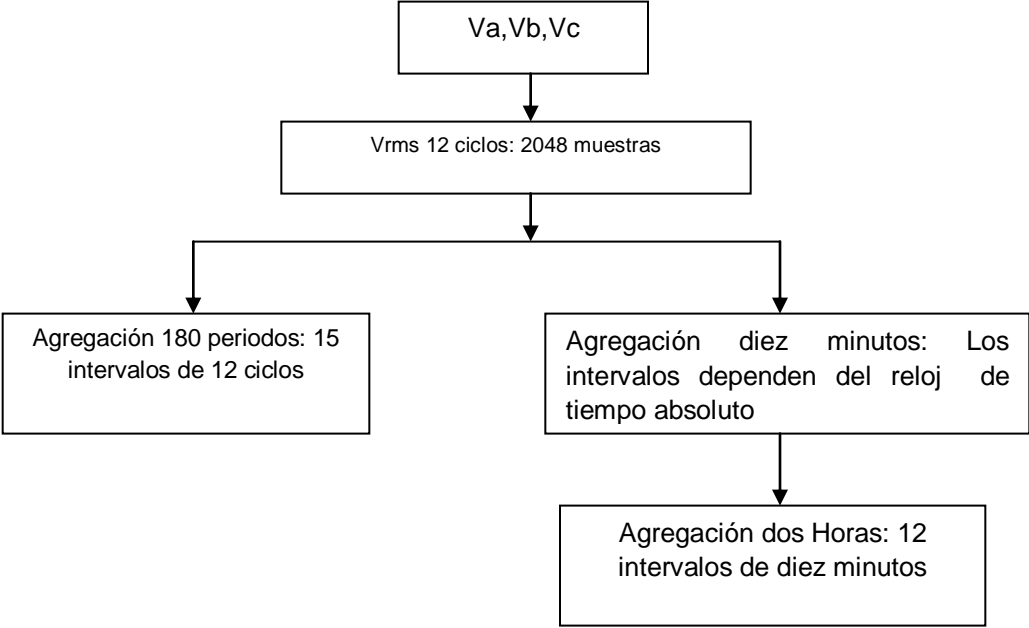
Figura 4. Diagrama de flujo para el cálculo de valor eficaz de doce ciclos



Fuente: Autores

Este procedimiento se realiza en forma paralela para cada una de las tensiones de fase de la señal de entrada. En la Figura 5 se muestra el esquema donde se resume el proceso completo para la medida de la amplitud de la tensión de alimentación.

Figura 5. Esquema para la medición de la amplitud de la tensión de alimentación



Fuente: Autores

**4.2.4 Sobretensión y subtensión.** La estimación de los valores de sobretensión y subtensión, se realizan a partir de las ecuaciones (9), (10), (11) y (12), donde ( $U_{ef}$ ) son los valores eficaces de doce ciclos de las tensiones de fase, obtenidas en la sección 4.2.3 para la amplitud de la tensión de alimentación, y ( $U_{din}$ ) hace referencia a la tensión de entrada declarada.

- Si la  $U_{ef}$  es menor o igual que  $U_{din}$ , entonces:

$$U_{baja} = \frac{U_{din} - U_{ef}}{U_{din}} * 100 \quad (9)$$

$$U_{alta} = 0 \quad (10)$$

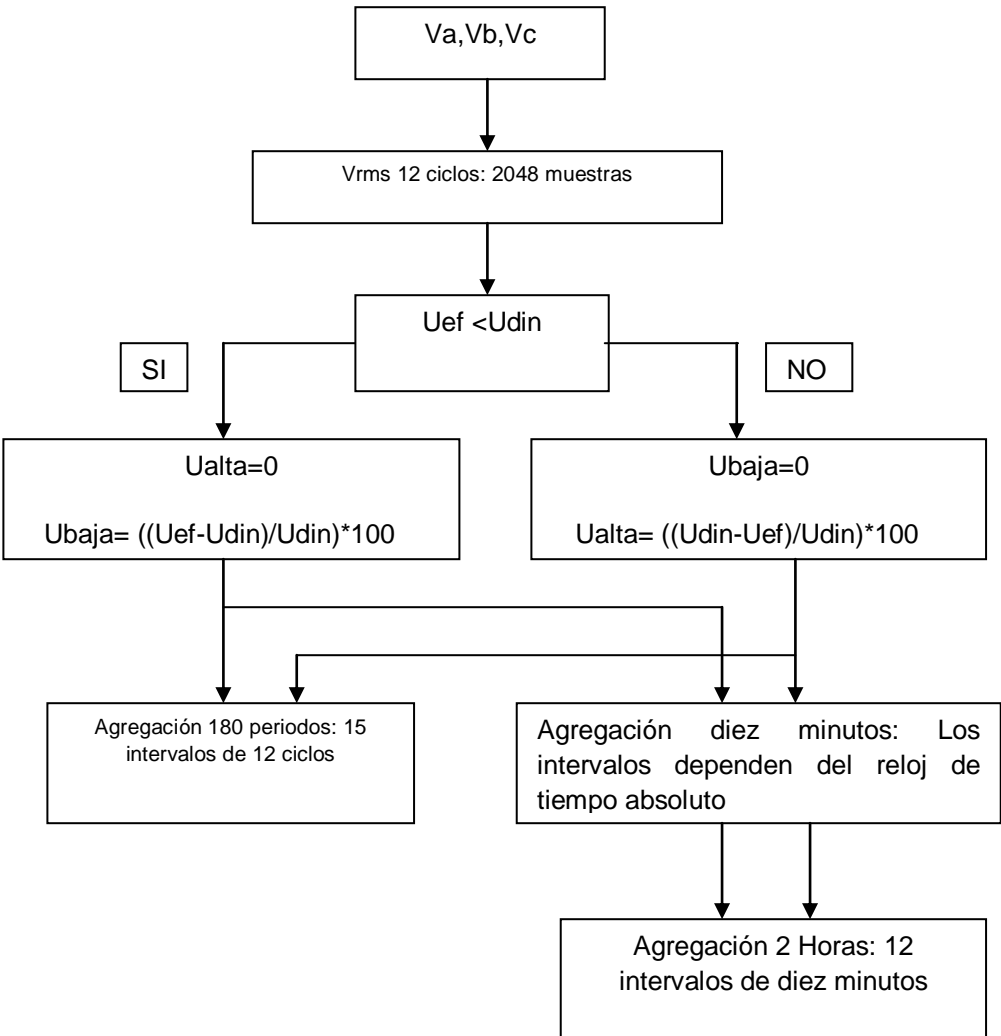
- En el caso contrario ( $U_{ef}$  mayor que  $U_{din}$ ):

$$U_{alta} = \frac{U_{ef} - U_{din}}{U_{din}} * 100 \quad (11)$$

$$U_{baja} = 0 \quad (12)$$

En la Figura 6 se presenta el diagrama de flujo para el cálculo de los valores de sobretensión y subtensión.

Figura 6. Esquema para la medición de los valores de sobretensión y subtensión



Fuente: Autores

**4.2.5 Evaluación de la medida.** Para cada uno de los eventos de estado estacionario (desbalance, amplitud de la tensión de alimentación, sobretensión y subtensión) la evaluación de la medida se realiza mediante los intervalos de agregación de ciento ochenta períodos, diez minutos y dos horas.

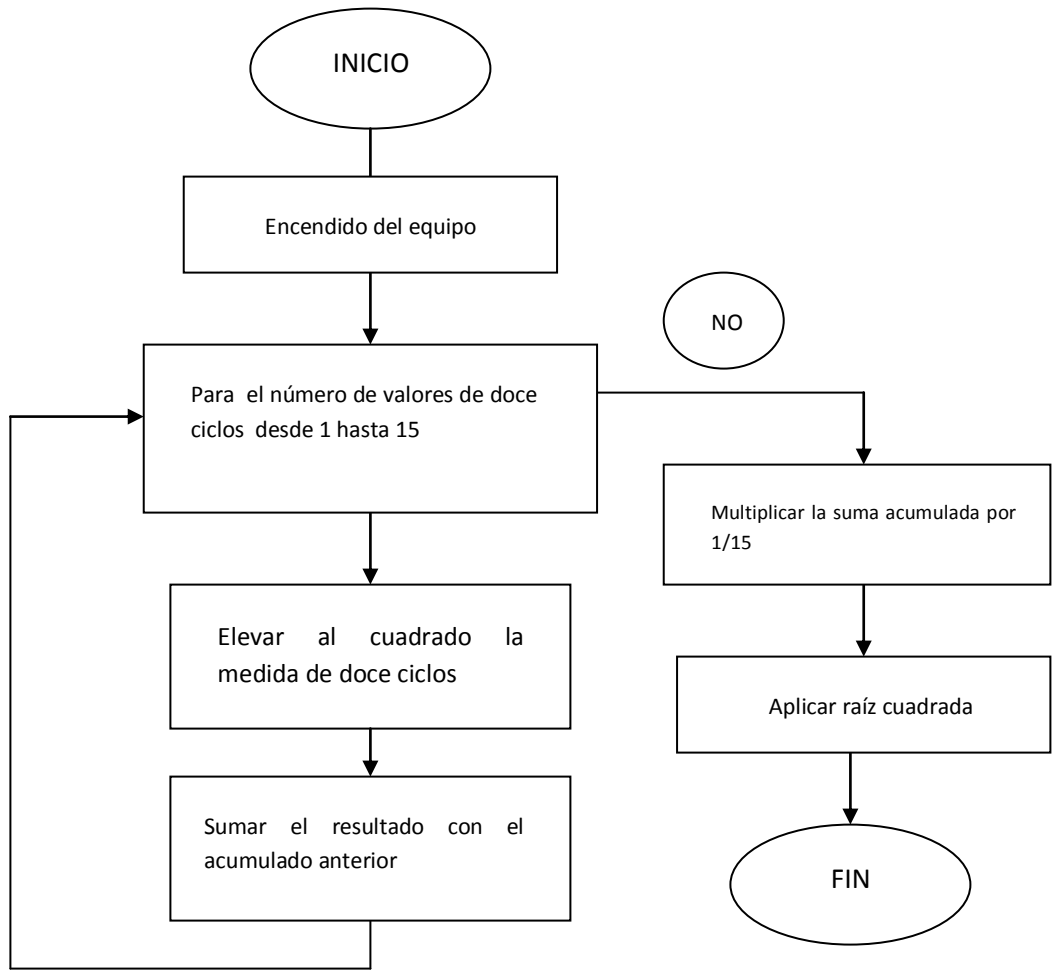
**4.2.5.1 Agregación de ciento ochenta períodos.** La agregación de 180 períodos se realiza a partir de quince intervalos de doce ciclos. En la Figura 7 se presenta la rutina para la realización del proceso. El inicio del primer intervalo de doce ciclos coincide con el encendido del equipo de medida mientras que el final se encuentra determinado por un número constante de muestras comprendidas en la ventana de doce ciclos. Los intervalos son contiguos hasta llegar al intervalo quince donde finalmente se realiza la evaluación de la primera medida (ver ecuación 13) y el inicio para la siguiente agregación.

$$U_{180} = \sqrt{\frac{1}{15} \sum_{i=1}^{15} U_{doce\text{ciclos}}(i)^2} \quad (13)$$

Donde:

$U_{doce\text{ciclos}}$ , son las medidas de los eventos estacionarios en un intervalo de doce ciclos.

Figura 7. Diagrama de flujo para la agregación de un tiempo de ciento ochenta períodos



Fuente: Autores

**4.2.5.2 Agregación de diez minutos.** El inicio y final de los intervalos de agregación se encuentran determinados por el reloj de tiempo absoluto. En la Figura 8 se muestra el diagrama de flujo del proceso, el cual se encuentra dividido de la siguiente forma: La primer parte consiste ir actualizando las muestras existentes desde que se enciende el equipo de medida hasta la hora absoluta mas

cercana (X: 00, X: 10, X: 20, X: 30, X: 40, X: 50), debido a que estas muestras no se deben incluir para el proceso de agregación. Además, se guarda la hora absoluta para actualizar el siguiente intervalo de agregación. En la segunda parte se ejecuta el algoritmo de medida de doce ciclos de eventos estacionarios a partir de la última muestra actualizada mientras no se llegue al próximo intervalo de tiempo absoluto. Al cumplirse la hora de tiempo absoluto se actualiza la muestra donde esto ocurre ya que esa muestra indica el comienzo del siguiente intervalo de agregación, y se guarda el tiempo con el fin de actualizar el nuevo tiempo absoluto para la agregación. Se completa la última ventana de doce ciclos y se guarda el número de valores de doce ciclos comprendidos en el intervalo de observación de diez minutos. Finalmente, la agregación de diez minutos se efectúa mediante la raíz cuadrada del promedio de los cuadrados de los valores de doce ciclos comprendidos en el intervalo de diez minutos (ver ecuación 14).

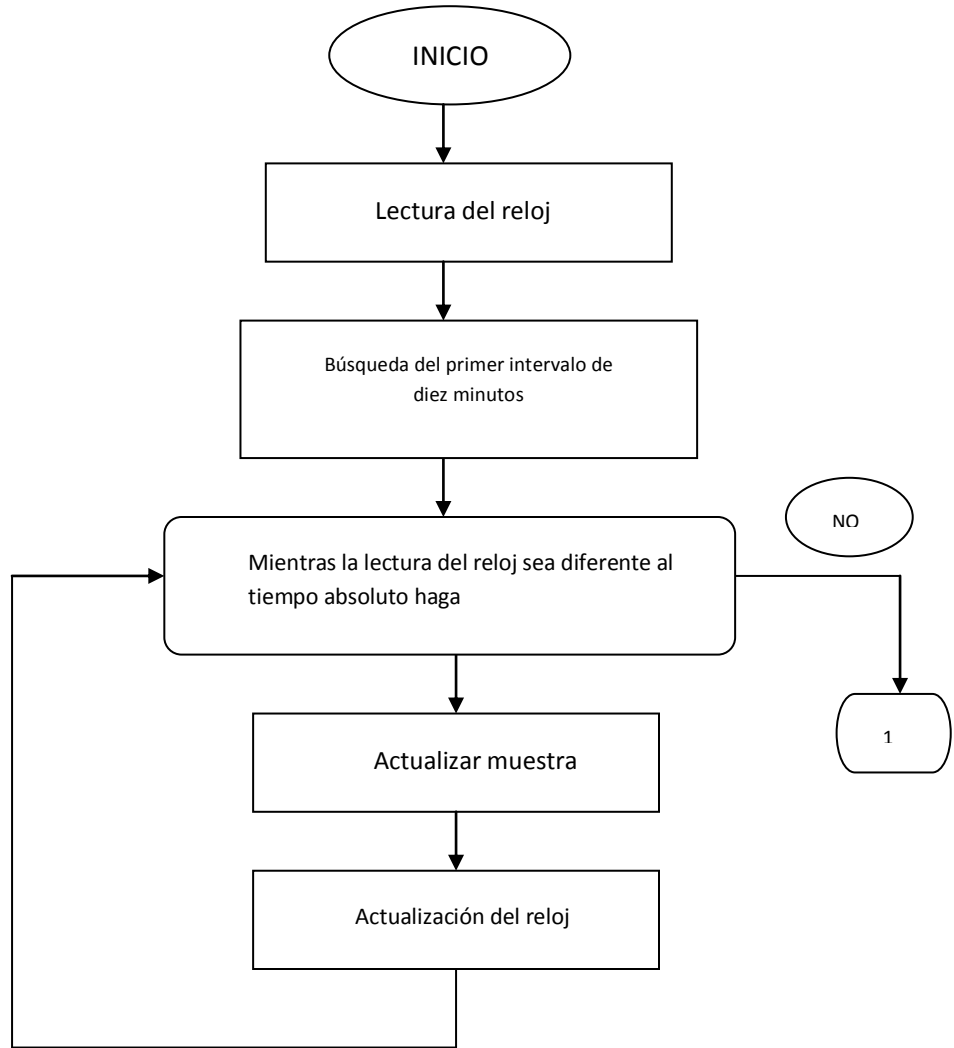
$$U_{10min} = \sqrt{\frac{1}{N} \sum_{1}^N U_{doce ciclos}(i)^2} \quad (14)$$

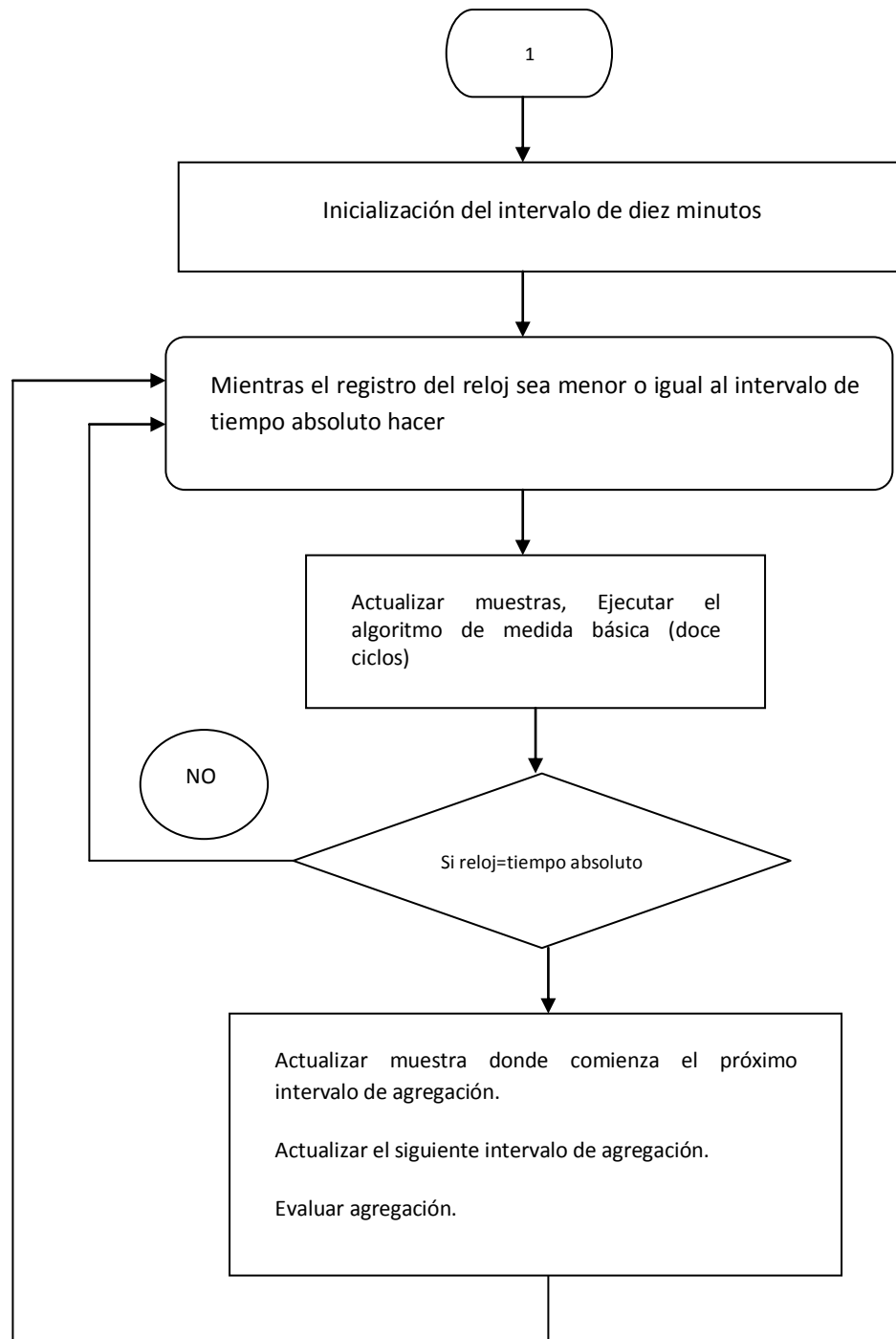
Donde:

$U_{doce ciclos}$ , son las medidas de los eventos estacionarios en un intervalo de doce ciclos.

N, es el número de intervalos de doce ciclos existentes en el intervalo de agregación de diez minutos.

Figura 8. Diagrama de flujo para el tiempo de agregación de diez minutos





Fuente: Autores

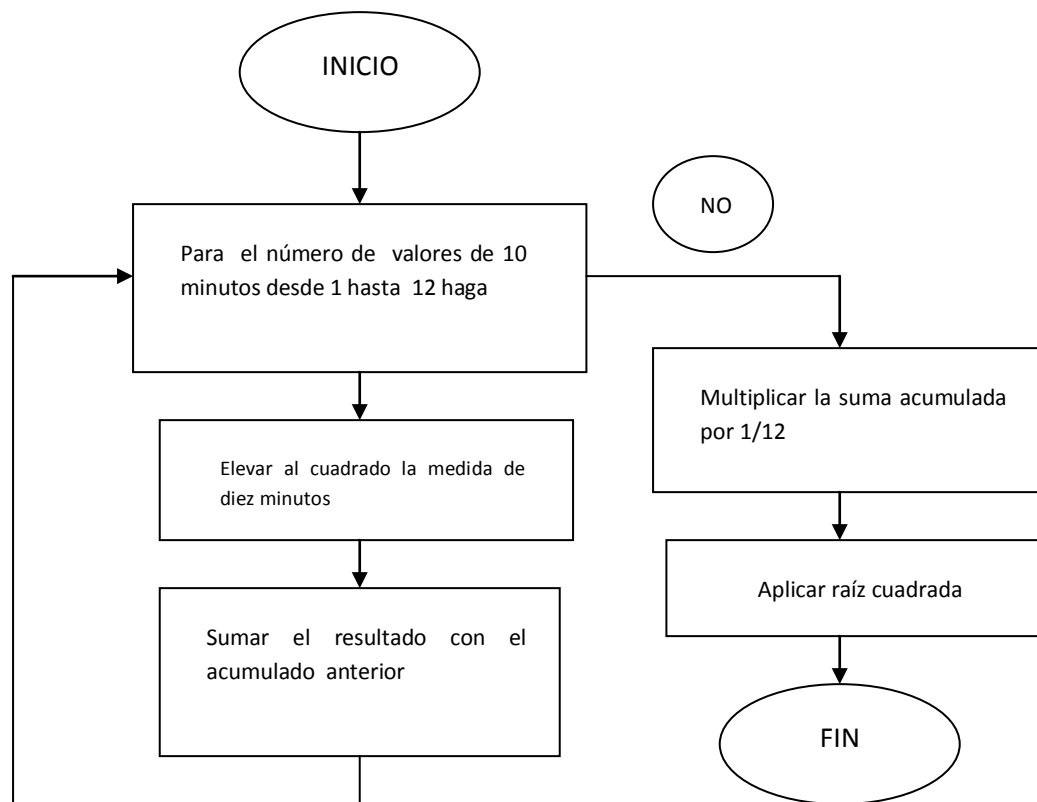
**4.2.5.3 Agregación de dos horas.** La agregación de dos horas se realiza a partir de doce intervalos de diez minutos. En la Figura 9 se presenta la rutina para la realización del proceso. Los valores de diez minutos se evalúan en tiempo absoluto y cuando se llegan a tener doce de ellos se efectúa la evaluación de la medida de dos horas (ver ecuación 15) y comienza la siguiente agregación.

$$U_{2hr} = \sqrt{\frac{1}{12} \sum_{1}^{12} U_{10min}(i)^2} \quad (15)$$

Donde:

$U_{10min}$  son las medidas de los eventos estacionarios en un intervalo de diez minutos.

Figura 9. Diagrama de flujo para el tiempo de agregación de dos horas



Fuente: Autores

#### 4.3 MONITORIZACIÓN DE EVENTOS DE ESTADO TRANSITORIO

Los eventos de estado transitorio son aquellos en los cuales se produce un cambio inesperado en la tensión de la red con una duración no superior a un minuto. Estos eventos se presentan por cambios de las características del funcionamiento del sistema como la incidencia de una descarga atmosférica en la red eléctrica, la conexión o desconexión de cargas eléctricas de magnitudes importantes, especialmente motores y condensadores, la presencia de una falla en el sistema eléctrico entre otros. Estos eventos se clasifican en:

- Interrupción de tensión.
- Huevo (hundimiento) de tensión.
- Sobretensión temporal.

En el párrafo 3.1.1.2 se presentó que la CREG estableció el estándar IEC 61000-4-30 como norma a cumplir para el diseño de los algoritmos de medición de estas perturbaciones. A continuación se hace un análisis de los parámetros a tener en cuenta en el desarrollo de estos algoritmos, medida básica (tensión eficaz refrescada cada semiperiodo  $U_{ef(1/2)}$ ), detección, evaluación e incertidumbre.

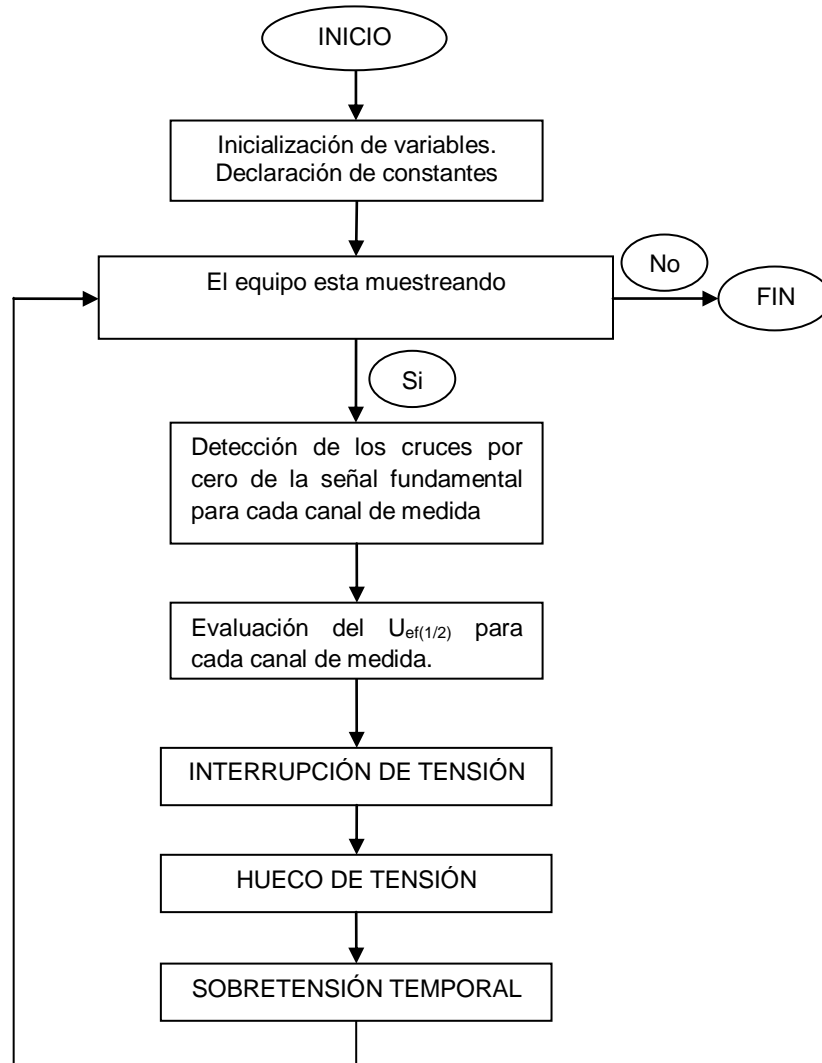
La detección hace referencia a los umbrales que indican si ha ocurrido un evento o no. Existen dos umbrales, el inicial y el final, cada uno está en función de un porcentaje de  $U_{din}$  o  $U_{rd}$  más un valor de histéresis. La detección se puede realizar en las redes monofásicas y trifásicas, por lo que el estándar presenta dos definiciones. Para analizar mejor los resultados se deben registrar la fecha y hora en que la tensión  $U_{ef(1/2)}$  traspasa un umbral con el fin de contrastarlo con posibles causas del mismo.

La evaluación hace referencia al parámetro que caracteriza cada evento y que permite analizar las perturbaciones de la señal de tensión. El hueco de tensión se evalúa con la tensión residual y su duración; la sobretensión temporal con la amplitud máxima de la sobretensión y su duración y finalmente la interrupción de tensión se evalúa con la duración. Dado que una interrupción de tensión es un caso especial del hueco de tensión (establecido en el estándar [IEC 61000-4-30, 2003]), no se requiere otro parámetro de evaluación diferente.

La incertidumbre de la medida permite cuantificar la clase de exactitud y la precisión del equipo de medición. En este sentido, el estándar [IEC 61000-4-30, 2003] propone un valor de incertidumbre para cada evaluación de la medida. Para establecer el cumplimiento de los requisitos propuestos por la norma con relación a la incertidumbre, se realizaron pruebas con señales sintéticas construidas a partir de los requerimientos exigidos por la norma.

En la Figura 10 se presenta el diagrama de flujo del proceso de medición. El algoritmo utilizado detecta el primer cruce por cero de la componente fundamental de la señal, después evalúa el valor eficaz de la tensión en un periodo comenzando en el cruce por cero detectado (definido en [IEC 61000-4-30, 2003]). Obtenido el valor eficaz de la tensión, verifica si ha ocurrido un evento comparando el valor estimado con los umbrales establecidos para cada evento; posteriormente, se detecta el siguiente cruce por cero, y se repite el proceso. Es decir, la estimación del valor eficaz se realiza (refresca) cada medio ciclo.

Figura 10. Diagrama de flujo del medidor trifásico de perturbaciones transitorias de tensión

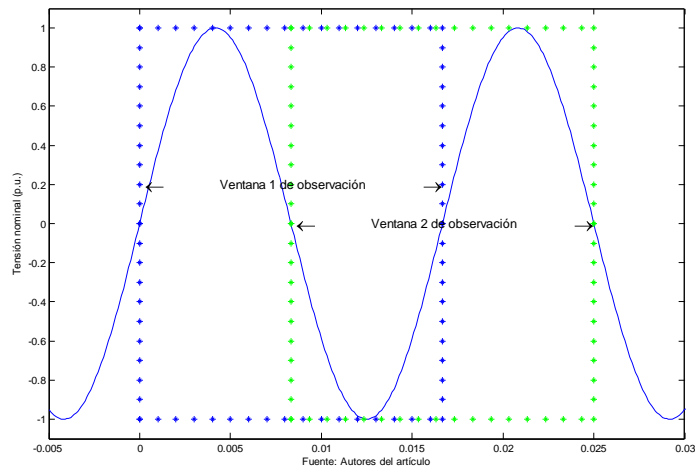


Fuente: Autores

**4.3.1 Medida básica.** El estándar [IEC 61000-4-30, 2003] propone como medida básica para los tres eventos, el valor eficaz de la tensión evaluada en un ciclo y refrescada cada medio ciclo  $U_{ef(1/2)}$ . Este valor se compara con los umbrales de detección. La Figura 11 presenta un esquema que representa la definición de la

medida básica. La tensión  $U_{ef(1/2)}$  es medida sobre una ventana de observación de un periodo de duración, comenzando en un cruce por cero de la componente fundamental y refrescada cada semiperiodo. Después se detecta el siguiente cruce por cero y se repite nuevamente la estimación (ventana 2 de observación) como se muestra en la Figura 11. Para una adecuada estimación es necesario que la detección del cruce por cero se realice con la componente fundamental de la señal de tensión, debido a que la distorsión armónica que se puede presentar en la señal de la red puede hacer que existan varios cruces por cero en un periodo, lo cual dificultaría establecer el inicio de la medida.

Figura 11 Representación de la medida básica para la medición de eventos transitorios

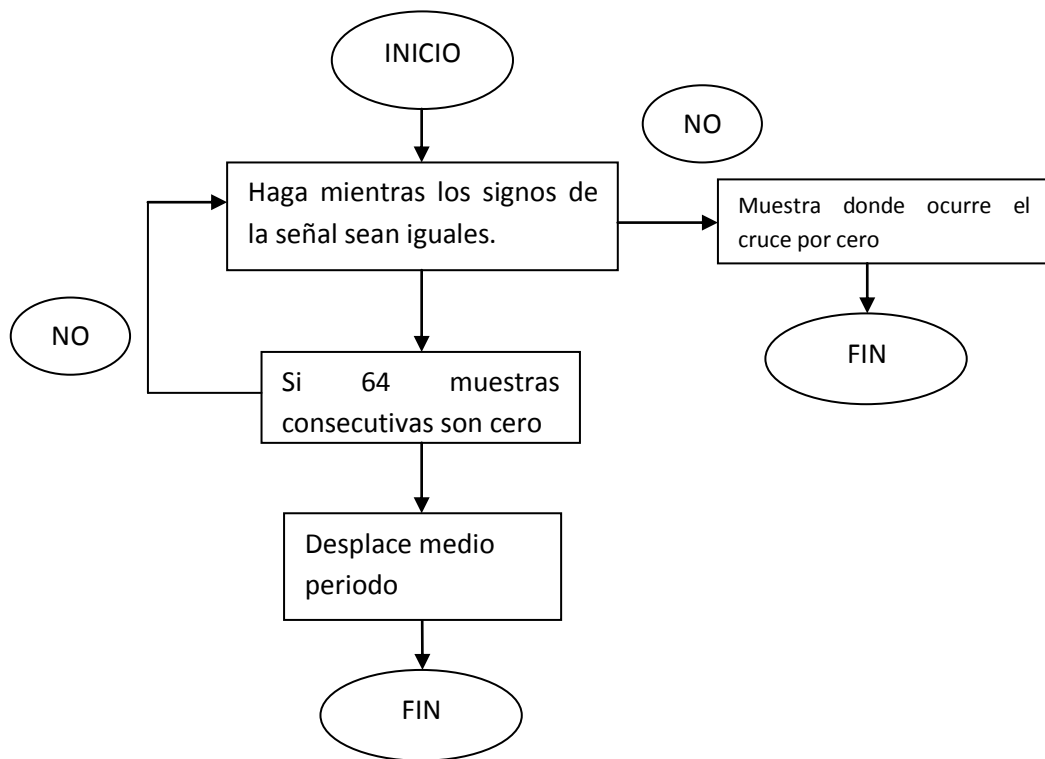


Fuente: Autores

El valor de la tensión eficaz se puede realizar de dos formas: con muestras fijas o con muestras variables en una ventana de observación. La Figura 12 muestra el diagrama de flujo de la detección de los cruces por cero, para la estimación del valor eficaz de la tensión con muestras fijas. El algoritmo compara los signos de dos muestras consecutivas durante medio periodo hasta que dichos signos sean

diferentes. En ese instante sale del ciclo y la muestra donde ocurrió el cruce queda almacenada para ser utilizada por la rutina de evaluación de  $U_{ef(1/2)}$ . Esta rutina realiza la medición sobre las siguientes N muestras después del cruce por cero, lo cual indica que si ocurre una falta de sincronización el algoritmo presentará errores en la medida puesto que la ventana de observación será mayor o menor que un periodo de la señal. Cuando 64 muestras consecutivas son cero se asume que ocurrió una caída total de la tensión de la red, por tanto no se puede determinar la  $U_{ef(1/2)}$ , puesto que no se conoce el cruce por cero. Para corregir este problema se optó por desplazar arbitrariamente la señal en medio periodo puesto que ese es el tiempo mínimo para encontrar un nuevo cruce por cero.

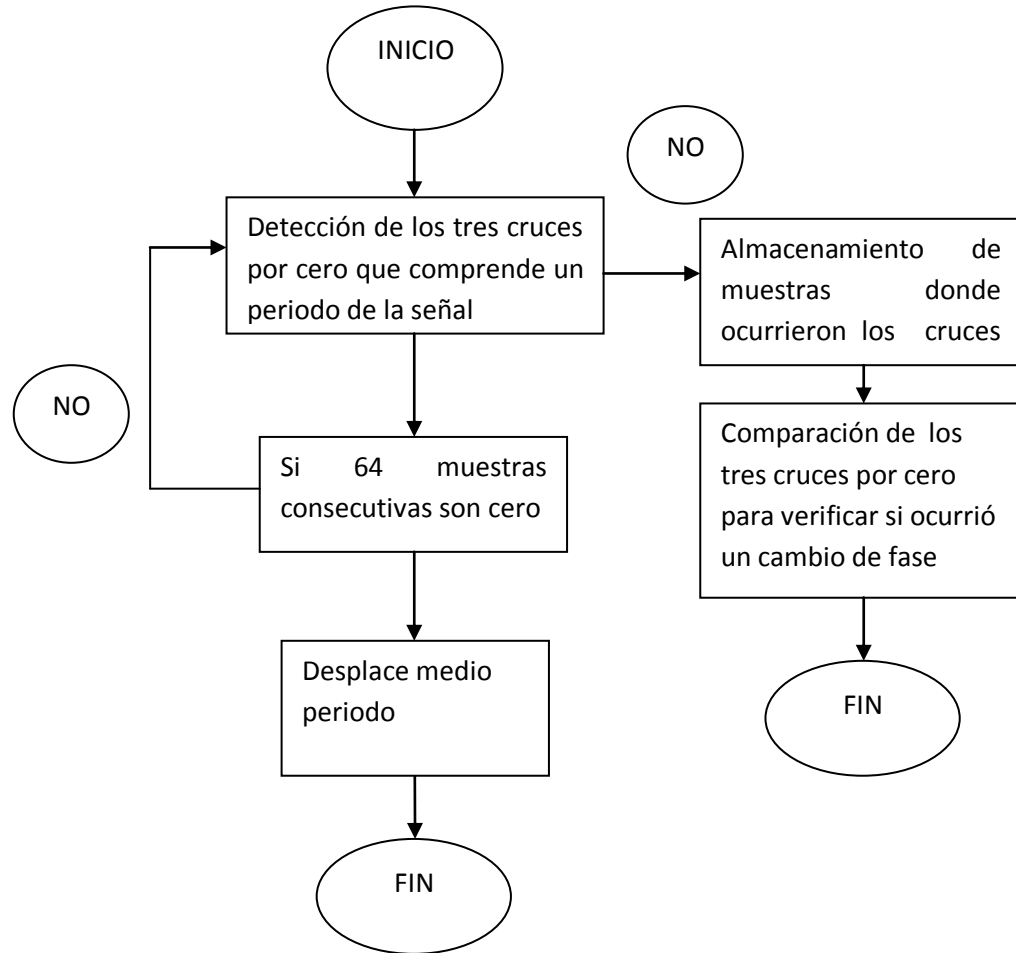
Figura 12 Diagrama de flujo para la detección de los cruces por cero para la estimación del valor eficaz de la tensión con un número de muestras fijo.



Fuente: Autores

La Figura 13 muestra la rutina para la detección de los cruces por cero para una medida básica del valor eficaz de la tensión con un número de muestras variable. El algoritmo detecta el cruce por cero al inicio del periodo, el cruce por cero donde ocurre el cambio polaridad del periodo y el cruce por cero al final del periodo. Después de detectar los cruces por cero se comparan entre ellos para verificar si ocurrió un cambio de fase, ya que de ocurrir este evento no sería una medida óptima. Se evalúa la distancia entre el primer y tercer cruce con respecto al cruce de cambio de polaridad para compararlo con un rango en el cual se supone que el semiciclo cumple con la forma senoidal de la señal y no con una señal amorfa producida por un cambio de fase. El valor eficaz de la tensión es estimado con las muestras existentes entre el primer y el tercer cruce por cero.

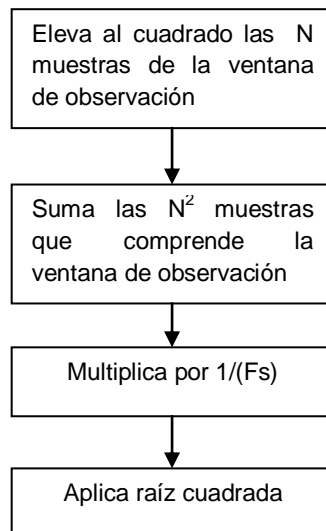
Figura 13 Diagrama de flujo para la detección de los cruces por cero para la estimación del valor eficaz de la tensión con un número de muestras variable.



Fuente: autores

La rutina para la estimación del valor eficaz de la tensión  $U_{ef(1/2)}$  se muestra en la Figura 14. La frecuencia  $F_s$  es la frecuencia de muestreo en muestras por ciclo.

Figura 14. Diagrama de flujo de la evaluación del valor eficaz de la tensión

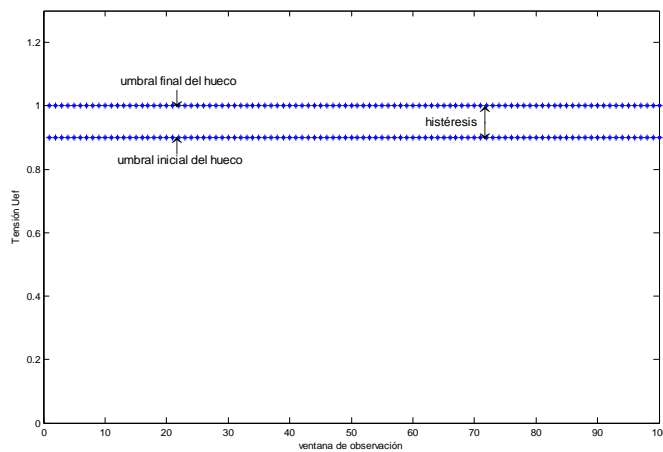


Fuente: Autores

**4.3.2 Hueco de tensión.** Un hueco (hundimiento) de tensión se define como la reducción del valor eficaz de la tensión  $U_{ef(1/2)}$  en algún punto de la red eléctrica por debajo de un umbral especificado.

**4.3.2.1 Detección.** El estándar IEC 61000-4-30 define dos umbrales, el primero corresponde al inicio del evento y el segundo al final del mismo, este último se compone de un valor umbral base (igual al umbral inicial del hueco u otro que defina el usuario) más un valor de histéresis. La Figura 15 muestra un esquema con esta definición.

Figura 15 Umbrales para la detección de un hueco de tensión.



Fuente: Autores

El estándar define la detección para redes monofásicas y redes trifásicas de la siguiente forma:

- “Redes monofásicas. Un hueco de tensión comienza cuando la tensión  $U_{ef(1/2)}$  cae por debajo del umbral del hueco y termina cuando la tensión  $U_{ef(1/2)}$  es igual o superior al umbral del evento más la tensión de histéresis” [IEC 61000-4-30, 2003].
- “Redes polifásicas. Un hueco comienza cuando la tensión  $U_{ef(1/2)}$  de uno o varios canales cae por debajo del umbral del hueco y se termina cuando la tensión  $U_{ef(1/2)}$  en todos los canales medidos es igual o superior al umbral del hueco mas la tensión de histéresis” [IEC 61000-4-30, 2003].

**4.3.2.2 Evaluación.** La caracterización de un hueco de tensión se realiza estimando tres parámetros característicos:

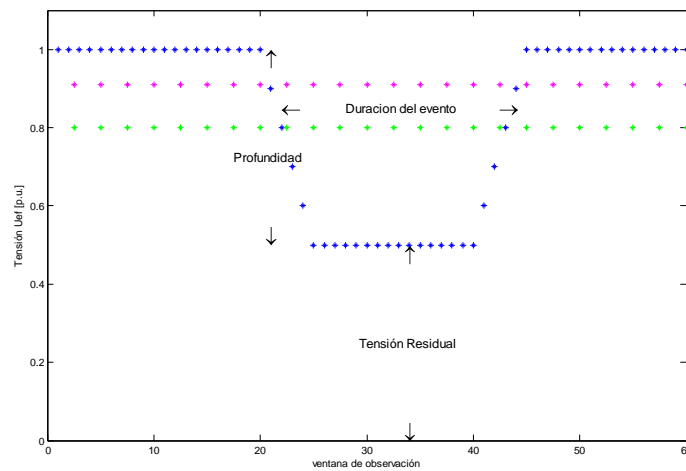
- Tensión residual. Definida como la mínima tensión  $U_{ef(1/2)}$  registrada en el canal de medida durante la ocurrencia del evento en una red monofásica y

la mínima tensión  $U_{ef(1/2)}$  registrada en cualquier canal de medida en una red polifásica. Este dato es referenciado a 0 [V] como lo muestra la Figura 16.

- Profundidad. La profundidad se define como la resta entre  $U_{din}$  (o  $U_{rd}$ ) y la tensión residual, por tanto este dato se encuentra referenciado a  $U_{din}$  (o  $U_{rd}$ ).
- Duración del evento. Es la diferencia entre el inicio y el final del evento. Este parámetro es afectado en gran proporción por la selección de los umbrales.

En redes polifásicas se puede presentar simultáneamente un hueco de tensión y una sobretensión temporal por lo tanto el valor eficaz de la tensión se debe comparar con todos los umbrales, caso contrario ocurre cuando el evento es una interrupción ya que para que suceda esta perturbación todos los valores eficaces de las tensiones de las tres fases deben estar por debajo del umbral inicial de la interrupción de tensión.

Figura 16. Representación de la evaluación de la medida de un hueco de tensión.



Fuente: Autores

**4.3.2.3 Incertidumbre.** En la estimación del valor eficaz para la detección de un hueco de tensión, el estándar exige una incertidumbre de  $\pm 0,2\%$  de  $U_{din}$ , y para la duración la incertidumbre es de un periodo de la señal.

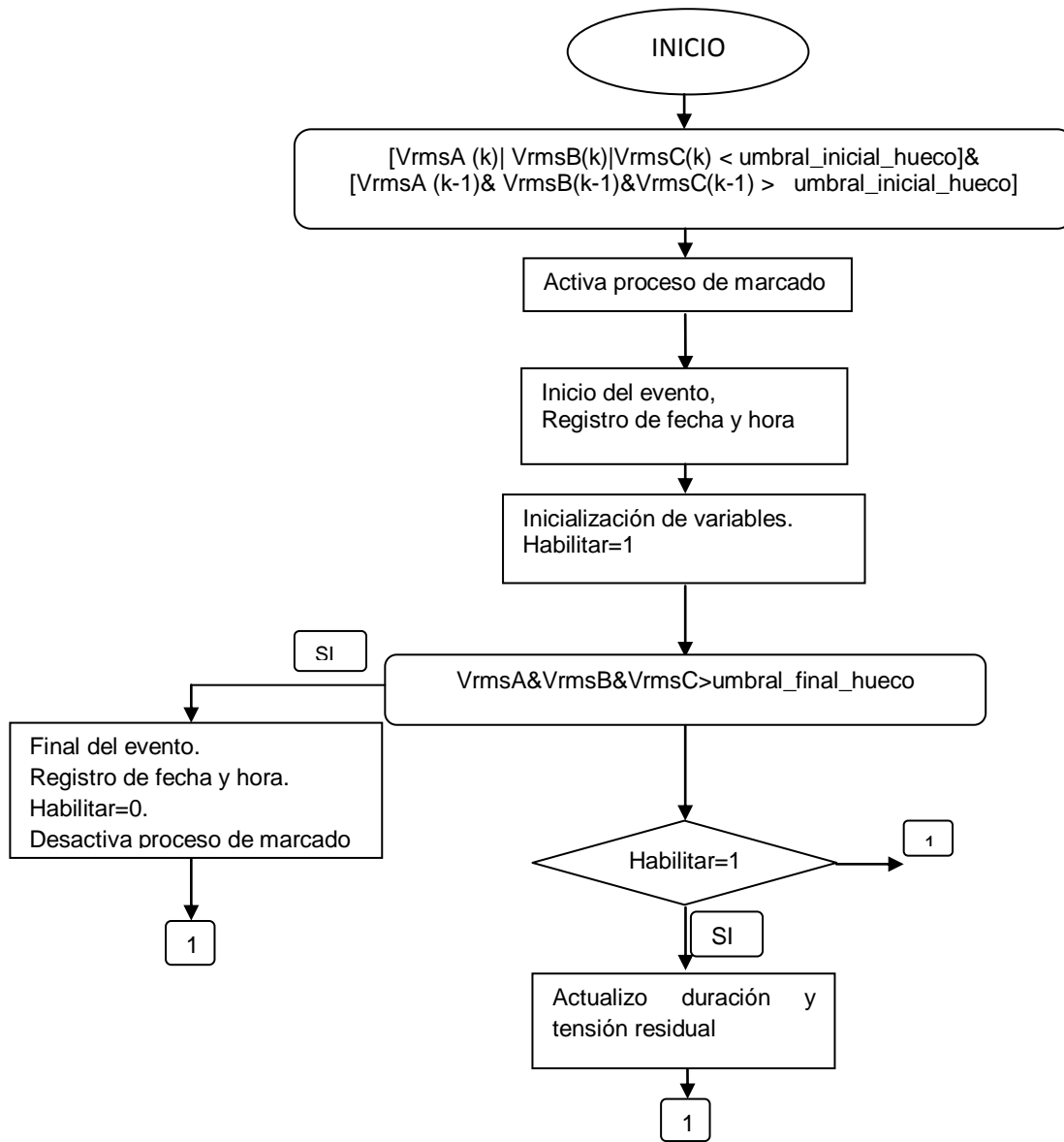
**4.3.2.4 Diagrama de flujo.** La Figura 17 presenta el diagrama de flujo para la estimación de los parámetros del hueco de tensión. Inicialmente se verifica el comienzo del evento comparando las amplitudes actuales y anteriores de la red cumpliendo con lo enunciado en el párrafo 4.3.2.1 para redes polifásicas. Si se cumple esta condición se activa la bandera (Habilitar) que habilita la evaluación de la medida, se registra la fecha en ocurrió el mismo y se activa el proceso de marcado para dejar constancia de que los parámetros estimados para los eventos estacionarios pueden tener incertidumbres altas comparadas con las establecidas para funcionamiento normal.

Después se verifica si los valores eficaces actualizados de las tensiones están por encima del umbral final del hueco. Si esta condición no se cumple, se procede a evaluar la medida, actualizar el registro de tensión residual y estimar la duración

del evento. Si la condición se cumple, se procede a registrar la fecha y hora en que terminó el evento, inhabilitar el proceso de evaluación y a desactivar el proceso de marcado. Posteriormente el valor eficaz de la tensión actualizado se comparará con el siguiente evento.

La medición de la duración del evento se realiza sumando todos los semiperíodos ocurridos durante la evento, escalados por la mitad de la frecuencia de muestreo y por el tiempo de muestreo, por eso es tan importante la sincronización del sistema.

Figura 17. Diagrama de flujo para la detección y evaluación de un hueco de tensión.

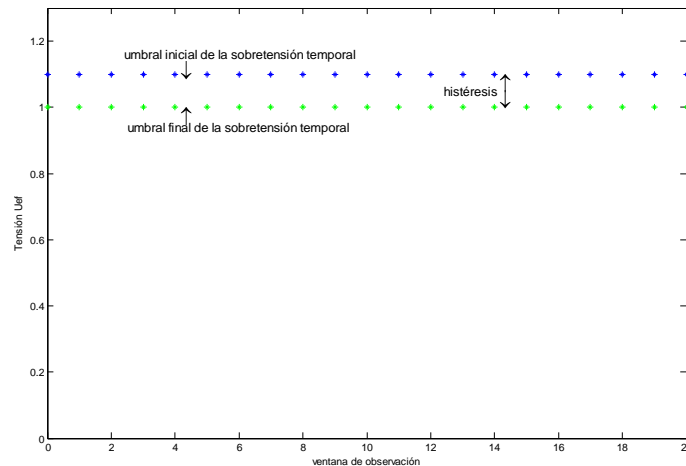


Fuente: Autores

**4.3.3 Sobretensión temporal.** Una sobretensión temporal se define como una elevación de la tensión  $U_{ef(1/2)}$  por encima del umbral.

**4.3.3.1 Detección.** El estándar IEC 61000-4-30 define dos umbrales, el primero corresponde al inicio del evento y el segundo al final del mismo, este se compone de un valor umbral base (igual al umbral inicial de la sobretensión temporal u otro que defina el usuario) menos la tensión de histéresis. La Figura 18 presenta un esquema de esta definición.

Figura 18. Umbrales de detección de una sobretensión temporal.



Fuente: Autores

Al igual que para la caracterización de huecos de tensión, la norma define la detección para redes monofásicas y redes trifásicas de la sobretensión temporal así:

- Redes monofásicas. Una sobretensión temporal comienza cuando la tensión  $U_{ef(1/2)}$  eleva por encima del umbral de la sobretensión temporal, y se termina cuando la tensión  $U_{ef(1/2)}$  es igual o inferior al umbral de la

sobretensión temporal menos la tensión de histéresis [IEC 61000-4-30, 2003].

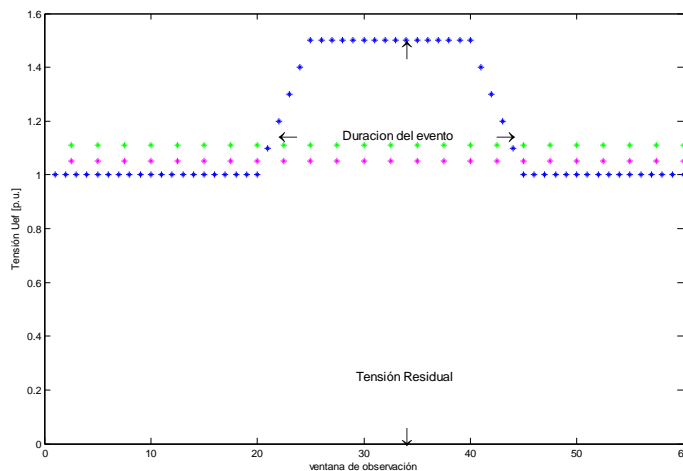
- Redes polifásicas. Una sobretensión temporal comienza cuando la tensión  $U_{ef(1/2)}$  de uno o más canales se eleva por encima del umbral de la sobretensión temporal, y se termina cuando la tensión  $U_{ef(1/2)}$  en todos los canales de medida es inferior o igual al umbral de la sobretensión temporal menos la tensión de histéresis [IEC 61000-4-30, 2003].

**4.3.3.2 Evaluación.** La evaluación de una sobretensión temporal se realiza determinando dos valores, ellos son:

- La amplitud máxima de la sobretensión temporal. Definida como la máxima tensión  $U_{ef(1/2)}$  registrada en el canal de medida durante la ocurrencia del evento en una red monofásica y la máxima tensión  $U_{ef(1/2)}$  registrada en cualquier canal de medida en una red polifásica. Este dato es referenciado a 0 [V].
- Duración del evento. Es la diferencia entre el inicio y el final del evento. Este parámetro es afectado en gran proporción por la selección de los umbrales.

En la Figura 19 se muestra la representación de la evaluación de este evento.

Figura 19. Representación de la evaluación de la medida de una sobretensión temporal.



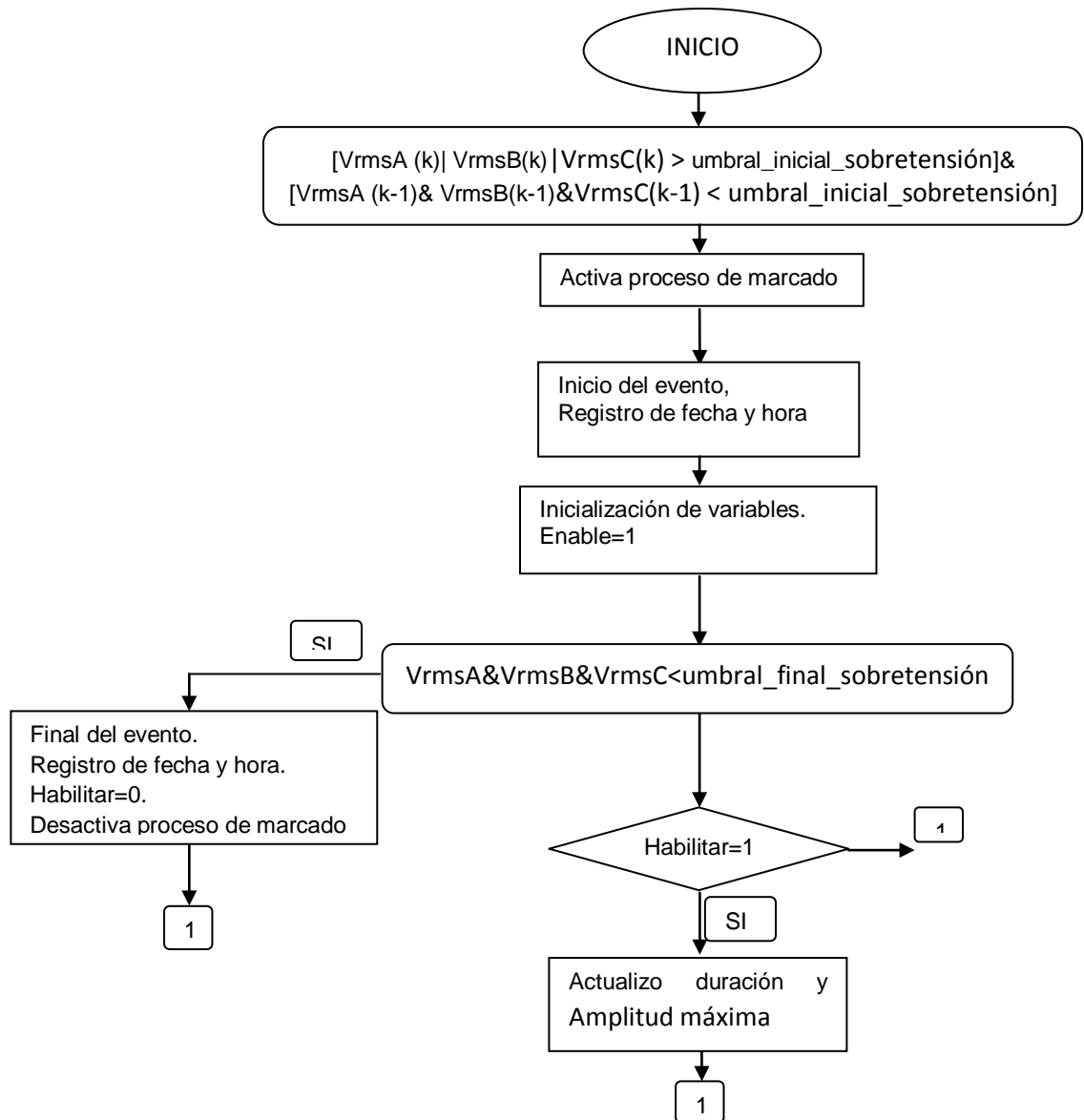
Fuente: Autores

**4.3.3.3 Diagrama de flujo.** La Figura 20 presenta el diagrama de flujo para la estimación de los parámetros de la sobretensión temporal. Inicialmente se verifica el inicio del evento comparando las amplitudes actuales y anteriores de la red cumpliendo con lo enunciado en la sección 4.2.3.1 para redes polifásicas. Si se cumple esta condición se activa la bandera (Habilitar) que habilita la evaluación de la medida, se registra la fecha en ocurrió el mismo y se activa el proceso de marcado para dejar constancia de que los parámetros estimados para los eventos estacionarios pueden tener incertidumbres altas comparadas con las establecidas para funcionamiento normal.

Después se verifica si los valores eficaces de las tensiones actualizadas están por debajo del umbral final de la sobretensión temporal, si esta condición no se cumple se procede a estimar la medida, actualizar el registro de amplitud máxima de la sobretensión y duración del evento. Si la condición se cumple, se procede a registrar la fecha y hora en que terminó el evento, inhabilitar el proceso de evaluación y a desactivar el proceso de marcado. Posteriormente el valor eficaz

actualizado de la tensión se compara con los valores de los umbrales para verificar si se presenta otro evento.

Figura 20. Diagrama de flujo para la detección y evaluación de una sobretensión temporal



Fuente: Autores

#### 4.3.4 Interrupción de tensión.

Una interrupción de tensión se define como una reducción de la tensión  $U_{ef(1/2)}$  por debajo de un umbral específico.

**4.3.4.1. Detección.** El estándar IEC 61000-4-30 define dos umbrales, el primero corresponde al inicio del evento y el segundo al final del mismo, este último se compone de un valor umbral base (igual al umbral inicial de la interrupción u otro que defina el usuario) más la tensión de histéresis. El estándar [IEC 61000-4-30, 2003] aclara que una interrupción de tensión es un caso especial de un hueco de tensión, por tanto solo es necesario comentar que las características de evaluación e incertidumbre son idénticas al hueco de tensión salvo por la duración del evento ya que este dato es diferente al de un hueco de tensión.

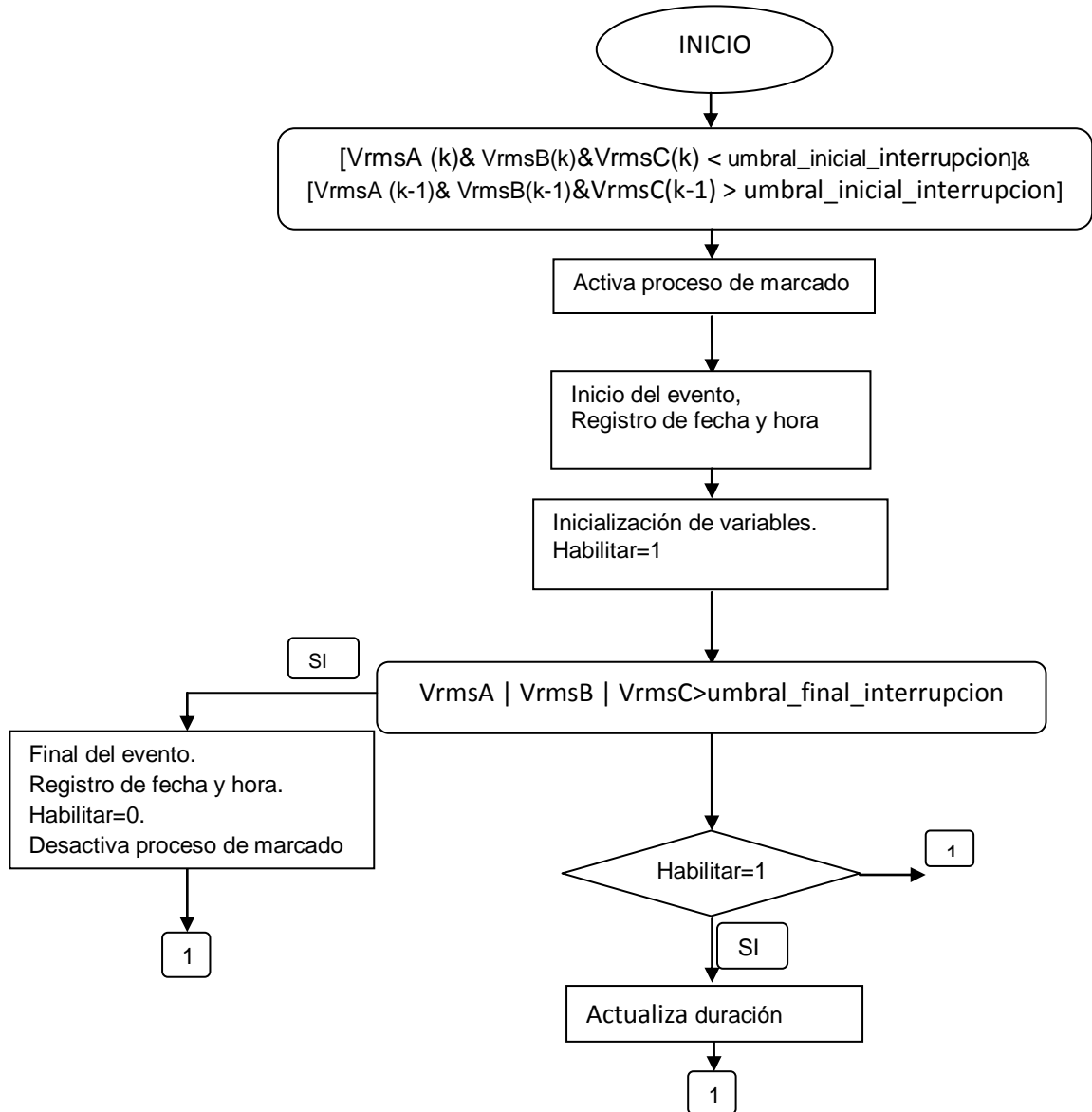
Para la monitorización de las interrupciones de tensión, la norma define la detección para redes monofásicas y redes trifásicas de la siguiente forma:

- Redes monofásicas. Una interrupción de tensión comienza cuando la tensión  $U_{ef(1/2)}$  está por debajo del umbral de tensión de tensión y se termina cuando el valor de tensión  $U_{ef(1/2)}$  es igual o superior al umbral de interrupción de tensión más la histéresis [IEC 61000-4-30, 2003].
- Redes polifásicas. Una interrupción de tensión comienza cuando la tensión  $U_{ef(1/2)}$  de todos los canales cae por debajo del umbral de interrupción de tensión y se termina cuando la tensión  $U_{ef(1/2)}$  de cualquier canal es igual o superior al umbral de interrupción de tensión más la histéresis [IEC 61000-4-30, 2003].

**4.3.4.2 Diagrama de flujo.** La Figura 21 presenta el diagrama de flujo para la medición de la interrupción de tensión. Inicialmente se verifica el comienzo del evento comparando las amplitudes actuales y anteriores de la red de acuerdo con lo enunciado en la sección 4.2.4.1 para redes polifásicas. Si se cumple esta condición se activa la bandera (Habilitar) que habilita la evaluación de la medida, se registra la fecha en ocurrió el mismo y se activa el proceso de marcado para dejar constancia de que los parámetros estimados para los eventos estacionarios pueden tener incertidumbres altas comparadas con las establecidas para funcionamiento normal.

Después se verifica si los valores eficaces actualizados de las tensiones eficaces actuales están por encima del umbral final de la interrupción. Si esta condición no se cumple se procede a estimar la medida y actualizar el registro de duración del evento. Si la condición se cumple, se procede a registrar la fecha y hora en que terminó el evento, inhabilitar el proceso de evaluación y a desactivar el proceso de marcado. Posteriormente, el valor eficaz actualizado de la tensión se compara con los valores de los umbrales para verificar si se presenta una perturbación.

Figura 21. Diagrama de flujo para la detección y evaluación de una interrupción de tensión.



Fuente: Autores

Los procesos enunciados anteriormente son aplicables a sistemas monofásicos (es solo aplicarlo a una fase). Para la comparación se requiere verificar el inicio y fin con el valor eficaz de la tensión de una sola fase. Los otros procesos son

idénticos. Se explicó el algoritmo trifásico porque será utilizado en la unidad constructiva, las pruebas serán realizadas con el algoritmo monofásico ya que la medida básica y evaluación tienen las mismas características.

#### **4.4 INFORME CREG**

En este apartado se describen las exigencias de presentación de los reportes relacionados con la calidad de la energía eléctrica según las resoluciones de la CREG 024 del 2005 y 016 del 2007.

**4.4.1 Eventos de estado estacionario.** De acuerdo con la modificación de la resolución 024 de 2005, presentada en la resolución 016 de 2007 el almacenamiento de eventos se debe realizar así:

##### ARTÍCULO 4. Plan de Recolección de datos

b) Almacenamiento de las fluctuaciones estacionarias de tensión. Las desviaciones, en valor absoluto, de la tensión eficaz de duración superior a un minuto y superiores o iguales al 10% de la tensión nominal, serán almacenadas de forma separada de las discontinuidades por interrupción, de duración superior a un minuto, pero su efecto será tenido en cuenta para la evaluación de los valores obtenidos de Pst según lo indicado en el literal f) [CREG 016, 2007].

e) Almacenamiento Voltaje de Secuencia Negativa. Se tomará al menos una medida de la relación  $V(2)/V(1)$  ( Voltaje de Secuencia Negativa sobre Voltaje de Secuencia Positiva) en cada barraje de subestación donde se conecten las unidades constructivas reconocidas, según lo descrito en el Artículo 3º. En caso que  $V(2)$  y  $V(1)$  sean simultáneamente menores al 10% la relación tomará el valor de cero. Se dejará constancia de esta medición en los registros de Pst según lo indicado en el literal f) [CREG 016, 2007].

**4.4.2 Eventos de estado transitorio.** La CREG solicita un archivo del tipo CSV en el cual queden registrados algunos eventos solicitados por la entidad en el artículo 3° de [CREG 024, 2005].

El archivo será llamado CEL\_Semana\_ j\_PM.csv, donde j corresponde al número de de la semana, y PM corresponde al número de punto de medida. Las características que deben ser registradas por el operador de red son:

Número y duración de interrupciones de tensión

Número de desviación y duración (en segundos) estacionarias de tensión (DET)

El valor de Pst de cada fase.

Los resultados almacenados en el archivo CSV deben tener cuatro cifras decimales utilizando el siguiente formato:

“dd/mm/aaaa, hh:mm, NI, DI ,NDET ,DDET ,Pst\_R, Pst\_S, Pst\_T, V2V1”. (dd = día, mm= mes , aaaa = año , hh = hora , mm = minuto , NI = Número de interrupciones , DI = Duración de interrupciones, NDET = Número de DET, DDET = Duración de las DET, Pst\_R S ó T = Pst por fases, V2V1=relación  $V^{(2)}/V^{(1)}$  ).

## **5. VERIFICACIÓN DE LA CONFORMIDAD CON EL ESTÁNDAR IEC 61000-4-30 (2003)**

Con el fin de verificar la conformidad con el estándar IEC 61000-4-30 se ha realizado una serie de pruebas al algoritmo de medida de perturbaciones de tensión de estado transitorio y estacionario, para obtener los errores que permitan determinar si dichas medidas se encuentran en el margen de incertidumbre exigido para la clase A.

En el estándar IEC 61000-4-30 se encuentra consignado la metodología para realizar las pruebas de desbalance, y amplitud de la tensión de alimentación. Sin embargo, no se especifican pruebas para los valores de sobretensión y subtensión. Por consiguiente, se ha adoptado aplicar las mismas señales que en el caso de la amplitud de la tensión de alimentación, ya que la unidad de medida básica es la misma (valor eficaz de la tensión de doce ciclos). En el caso de los eventos de estado transitorio, el estándar [IEC 61000-4-30,2003] no propone pruebas para esta clase de eventos por lo cual se ha planteado utilizar un proceso similar con las mismas señales sintéticas para verificar la conformidad para el desempeño clase A.

Las señales de prueba se encuentran consignadas en la tabla 3 del capítulo 6 de [IEC 61000-4-30, 2003], las cuales constan de tres condiciones de ensayo y cinco puntos equidistantes en el margen de influencia a medir como se puede apreciar en las tablas 2 y 3 respectivamente.

Tabla 2. Condiciones de ensayo de incertidumbre para los métodos clase A

| Magnitud de influencia | Condiciones de ensayo 1      | Condiciones de ensayo 2   | Condiciones de ensayo 3   |
|------------------------|------------------------------|---|---|
| Frecuencia             | $F_{nom} \pm 0,5 \text{ Hz}$ | $F_{nom} -1 \text{ Hz} \pm 0,5 \text{ Hz}$  | $F_{nom} +1 \text{ Hz} \pm 0,5 \text{ Hz}$  |
| Armónicos              | 0% a 3% de $U_{din}$         | 10%±3% de $U_{din}$ 3 <sup>er</sup> a 0°<br>5%±3% de $U_{din}$ 5° a 0°<br>5%±3% de $U_{din}$ 29° a 0° | 10%±3% de $U_{din}$ 7° a 0°<br>5%±3% de $U_{din}$ 13° a 0°<br>5%±3% de $U_{din}$ 25° a 0° |
| Interarmónicos         | 0% a 0,5% de $U_{din}$       | 1%±0,5% de $U_{din}$ a $7.5 \cdot F_{nom}$  | 1%±0,5% de $U_{din}$ a $3.5 \cdot F_{nom}$  |
| Flicker                | $P_{st} < 0,1$               | $P_{st} = 1 \pm 0,1$<br>Modulación Rectangular de 39 cambios por minuto                               | $P_{st} = 4 \pm 0,1$<br>Modulación Rectangular de 110 cambios por minuto                  |

Fuente: IEC 61000-4-30:2003

Tabla 3. Margen de variación de las magnitudes de influencia (señales de entrada) para el método clase A

| Magnitudes de Influencia | Margen de Variación    |
|--------------------------|------------------------|
| 0%-5%                    | Desbalance             |
| 0%-200%                  | Amplitud de la tensión |

Fuente: IEC 61000-4-30:2003

Para incluir la perturbación del fenómeno Flicker, fue necesario realizar un procedimiento adicional, debido a que en el estándar [IEC 61000-4-15,2003] no se dan a conocer pruebas que permitan obtener las características de la señal modulada (amplitud y frecuencia de modulación) a partir de un  $P_{st}$  cuyo valor sea diferente a 1. Sin embargo, es posible a partir de estos datos iniciales calcular las características de la señal teniendo en cuenta la linealidad existente en un rango determinado para una frecuencia de modulación constante entre la amplitud de la modulación y el valor del  $P_{st}$ . En [Oviedo y Rueda, 2009] se presenta con mayor detalle las pruebas realizadas para verificar la característica lineal del medidor de

Flicker. En la tabla 4 se muestra las pruebas para obtener los valores de Pst igual a 1, de acuerdo con [IEC 61000-4-15,2003].

Tabla 4. Especificaciones de ensayo para el clasificador de flicker

| <b>Cambios Rectangulares por minuto</b> | <b>Frecuencia de Modulación [Hz]</b> | <b>Amplitud de tensión <math>\Delta V/V</math> [%]</b> |
|---|--------------------------------------|--|
| 39                                      | 0,325                                | 1,044  |
| 110                                     | 0,917                                | 0,841  |
| 1620                                    | 13,5                                 | 3,166  |

Fuente: IEC 61000-4-15: 2003

Para las condiciones de ensayo uno, considerando que no se especifica la frecuencia de modulación, se ha optado por utilizar una frecuencia de modulación de 13,5 [Hz]. Debido a la linealidad del medidor, para obtener el valor de Pst aproximadamente igual a 0,1 que es igual a multiplicar el Pst de 1 por 0,1 se requiere multiplicar por el mismo factor la amplitud de la tensión ( $\Delta V/V$ ) por 0,1, luego el  $\Delta V/V$  necesario es  $3,166\% \cdot 0,1 = 0,316\%$  para la misma frecuencia de modulación.

En la condición de ensayo dos, no se requieren realizar los cálculos anteriores ya que se tienen las características de la señal modulada para un Pst de 1 con una frecuencia de 39 cambios por minuto, y para la condición de ensayo 3, el valor requerido para el  $\Delta V/V$  es  $4 \cdot 0,841\% = 3,364\%$  con una modulación rectangular de 110 cambios por minuto.

## 5.1 PRUEBAS REALIZADAS AL ALGORITMO DE MEDIDA PARA LA MONITORIZACIÓN DE EVENTOS DE ESTADO ESTACIONARIO

Las condiciones de ensayo realizadas incluyen el efecto de armónicos, interarmónicos, variaciones en la frecuencia del sistema y Flicker. Para las condiciones de ensayo 1 se ha tomado la decisión de realizar la prueba con los 10 primeros armónicos y todos con la amplitud máxima que es igual a 3%  $U_{din}$ . En el caso de interarmónicos se ha optado por evaluar un interarmónico a la mitad de la frecuencia por cada grupo de armónicos comprendidos entre los 10 primeros armónicos (8 interarmónicos). Finalmente, la prueba de frecuencia se ha efectuado para los valores límites del margen de variación, que este caso son de 50,5 [Hz] a una frecuencia de muestreo sincronizada de 8168,66666 [Hz] y 69,5 [Hz] con frecuencia de muestreo de 11861,3333[Hz].

Para la condición de ensayo dos se ha adoptado por realizar la prueba a la amplitud máxima de armónicos e interarmónicos, la cual es del 13% de  $U_{din}$  para el tercer armónicos, y 8% para el armónico quinto y veintinueve, y de 1,5% de  $U_{din}$  para el interarmónico. La prueba de frecuencia se ha efectuado para los valores límites del margen de variación, que son de 49,5 [Hz] a una frecuencia de muestreo sincronizada de 8448 [Hz] y 68,5 [Hz] con frecuencia de muestreo de 11690,66666[Hz].

En la condición de ensayo tres, las amplitudes usadas para el armónico séptimo, trece y veinticinco fueron 13%, 8% y 8% de  $U_{din}$  respectivamente, y para el interarmónico de 1,5% de  $U_{din}$ . La prueba de frecuencia se ha efectuado para los valores límites para el margen de variación, que en este caso es de 51,5 [Hz] a una frecuencia de muestreo sincronizada de 8789,3333 [Hz] y 70,5 [Hz] con frecuencia de muestreo de 12032 [Hz].

Finalmente, en caso de no tener sincronización de frecuencia, puede ser requerido una muestra de más o de menos si la relación entre el error de sincronismo proveniente de la tarjeta de adquisición de datos y la medida de frecuencia es menor que 1 o 0,5. Por consiguiente, se ha realizado unas pruebas empleando las señales sintéticas definidas previamente con el fin de cuantificar el error por utilizar durante el proceso de monitorización 2048 muestras cuando en realidad existen 2047 o 2049 muestras en la ventana de doce ciclos.

En el caso de la estimación del desbalance, los cinco puntos equidistantes utilizados dentro del margen de influencia (0-5%) fueron: 0%, 1,25%, 2,5%, 3,75% y 5%. El método empleado para calcular cada punto fue mantener constante la amplitud de la componente fundamental de la fase A e ir variando las amplitudes de la componente fundamental de las fases B y C en la misma proporción, hasta llegar al porcentaje de desbalance deseado. La metodología utilizada fue a través de pruebas de ensayo y error, debido a que no es posible calcular analíticamente las tensiones de fase a partir de las tensiones de línea (la matriz obtenida del sistema de ecuaciones no es invertible).

Este proceso aplica tanto a la componente de secuencia positiva como a la componente de secuencia cero ya que si se mantienen los ángulos equilibrados y únicamente se modifican los valores de amplitud de la señal, se llega a que ambas medidas son iguales.

Para la medición de la amplitud de la tensión de alimentación, y los valores de sobretensión y subtensión, los cinco puntos equidistantes utilizados en el margen de influencia (0-200% de  $U_{din}$ ) fueron: 50% de  $U_{din}$ , 75% de  $U_{din}$ , 100% de  $U_{din}$ , 125% de  $U_{din}$ , 150% de  $U_{din}$ . El método empleado para calcular cada punto es que de acuerdo a la variación que tuvieran las amplitudes de los armónicos e interarmónicos de la señal de tensión en cada una de las condiciones de ensayo, se fijara la amplitud de la componente fundamental de la señal de entrada, de tal

forma que el valor eficaz de la tensión de doce ciclos de la onda se mantuviera constante en 50%, 75%, 100%, 125% y 150% de  $U_{din}$  respectivamente, como se aprecia en las siguientes ecuaciones:

- En la condición de ensayo 1:

$$V_{effund} = \sqrt{V_{eftotal}^2 - 9 * V_{efarm}^2 - 8 * V_{efint}^2} \quad (16)$$

- En las condiciones de ensayo 2 y 3:

$$V_{effund} = \sqrt{V_{eftotal}^2 - 9 * V_{efarm}^2 - V_{efint}^2} \quad (17)$$

Donde:

$V_{eftotal}$  , Tensión eficaz de doce ciclos para la señal total

$V_{effund}$  , Tensión eficaz de doce ciclos de la componente fundamental de la señal

$V_{efarm}$  , Tensión eficaz de doce ciclos para los armónicos

$V_{efint}$  , Tensión eficaz de doce ciclos para los interarmónicos

**5.1.1 Análisis de resultados.** A continuación se presentan las tablas de errores obtenidos para la estimación del desbalance, amplitud de la tensión, sobretensión y subtensión bajo las condiciones de ensayo. En el caso de las componentes de secuencia negativa y secuencia cero, la incertidumbre de la medida es de  $\pm 0,15\%$  mientras que para la amplitud de la tensión, sobretensión y subtensión es de  $0,1\%$  de  $U_{din}$ .

Es importante señalar que las pruebas de frecuencia para la amplitud de la tensión, y los valores de sobretensión y subtensión se efectuaron para los valores límites del margen de variación, pero al obtenerse que los errores calculados eran iguales, se pueden resumir en un solo resultado.

### 5.1.1.1 Desbalance

#### a) Componente de secuencia negativa

Las tablas 5, 6 y 7 muestran los errores en la estimación de la componente de secuencia negativa bajo las condiciones de los ensayos propuestos en la norma IEC 61000-4-30 (ver Tabla 2).

Tabla 5. Errores en la estimación de la componente de secuencia negativa bajo las condiciones de ensayo 1.

| Señal sintética            | Componente de secuencia negativa |            |            |           |           |
|----------------------------|----------------------------------|------------|------------|-----------|-----------|
|                            | 0%                               | 1,25%      | 2,5%       | 3,75%     | 5%        |
| Armónicos e Interarmónicos | 1,504e-06                        | 1,299e-013 | 8,971e-013 | 8,40e-013 | 7,09e-013 |
| Frecuencia                 |                                  |            |            |           |           |
| 50,5 [Hz]                  | 1,501e-06                        | 1,099e-013 | 3,299e-013 | 2,19e-013 | 6,00e-013 |
| 69,5 [Hz]                  | 1,407e-06                        | 3,199e-013 | 3,099e-013 | 7,02e-014 | 4,90e-013 |
| Flicker                    | 0,010503                         | 0,008181   | 0,008335   | 0,008092  | 0,007985  |

Tabla 6. Errores en la estimación de la componente de secuencia negativa bajo las condiciones de ensayo 2.

| Señal sintética            | Componente de secuencia negativa |            |            |           |           |
|----------------------------|----------------------------------|------------|------------|-----------|-----------|
|                            | 0%                               | 1,25%      | 2,5%       | 3,75%     | 5%        |
| Armónicos e Interarmónicos | 1,509e-06                        | 1,298e-013 | 5,799e-013 | 8,20e-013 | 8,29e-013 |
| Frecuencia                 |                                  |            |            |           |           |
| 49,5 [Hz]                  | 1,051e-06                        | 3,399e-013 | 7,600e-013 | 8,02e-013 | 6,00e-013 |
| 68,5 [Hz]                  | 1,062e-06                        | 1,209e-012 | 1,02e-012  | 8,40e-013 | 7,09e-013 |
| Flicker                    | 2,535e-04                        | 9,1021e-05 | 7,6457e-05 | 1,568e-04 | 2,529e-04 |

Tabla 7. Errores en la estimación de la componente de secuencia negativa bajo las condiciones de ensayo 3.

| Señal sintética            | Componente de secuencia negativa |            |            |           |           |
|----------------------------|----------------------------------|------------|------------|-----------|-----------|
|                            | 0%                               | 1,25%      | 2,5%       | 3,75%     | 5%        |
| Armónicos e Interarmónicos | 1,501e-06                        | 1,299e-012 | 5,799e-013 | 8,20e-013 | 8,29e-013 |
| Frecuencia                 |                                  |            |            |           |           |
| 51,5 [Hz]                  | 1,541e-06                        | 1,204e-012 | 9,796e-013 | 8,41e-013 | 6,00e-013 |
| 70,5 [Hz]                  | 1,497e-06                        | 1,209e-012 | 1,021e-012 | 8,40e-013 | 8,20e-013 |
| Flicker                    | 8,7938e-04                       | 7,1675e-04 | 5,4897e-04 | 7,817e-04 | 8,771e-04 |

Como se puede observar en las tablas 5, 6 y 7, los errores máximos obtenidos en las condiciones de ensayo 1, 2 y 3 son inferiores a la incertidumbre (0,15%) de la medida para certificar la clase A. El error mínimo es de  $7,02e-014$  y el máximo de 0,01053%. Además, se verifica que la medida del desbalance no se ve afectada por el ingreso de armónicos e interarmónicos, y de la variación de la frecuencia del sistema desde que se encuentre sincronizada con la frecuencia de muestreo.

Tabla 8. Errores en la estimación de la componente de secuencia negativa a 60 [Hz] con una frecuencia de muestreo de 10235 [Hz] (2047 muestras en 12 ciclos) en las condiciones de ensayo 3

| Señal sintética            | Componente de secuencia negativa |          |           |          |          |
|----------------------------|----------------------------------|----------|-----------|----------|----------|
|                            | 0%                               | 1,25%    | 2,5%      | 3,75%    | 5%       |
| Armónicos e Interarmónicos | 0,0244255                        | 0,024584 | 0,0247439 | 0,024487 | 0,024364 |
| Flicker                    | 0,023546                         | 0,023705 | 0,0238614 | 0,023609 | 0,023487 |

Tabla 9. Errores en la estimación de la componente de secuencia negativa a 60 [Hz] con una frecuencia de muestreo de 10245 [Hz] (2049 muestras) bajo las condiciones de ensayo 3

| Señal sintética            | Componente de secuencia negativa |          |          |          |          |
|----------------------------|----------------------------------|----------|----------|----------|----------|
|                            | 0%                               | 1,25%    | 2,5%     | 3,75%    | 5%       |
| Armónicos e Interarmónicos | 0,0244136                        | 0,02425  | 0,024002 | 0,024216 | 0,024287 |
| Flicker                    | 0,025292                         | 0,025059 | 0,024880 | 0,025094 | 0,025163 |

Como se puede observar en las tablas 8 y 9, los errores que se presentan cuando no hay sincronización de frecuencia son mayores que en el caso sincronizado. Sin embargo, el error máximo en las condiciones de ensayo (0,025292%) sigue siendo inferior al error permisible (0,15%) para certificar la clase A.

b) Componente de secuencia cero

Las tablas 10, 11 y 12 muestran los errores en la estimación de la componente de secuencia cero bajo las condiciones de los ensayos propuestos en la norma IEC 61000-4-30 (ver Tabla 2).

Tabla 10. Errores en la estimación de la componente de secuencia cero bajo las condiciones de ensayo 1.

| Señal sintética            | Componente de secuencia cero |            |            |           |           |
|----------------------------|------------------------------|------------|------------|-----------|-----------|
|                            | 0%                           | 1,25%      | 2,5%       | 3,75%     | 5%        |
| Armónicos e Interarmónicos | 0,628e-012                   | 7,898e-013 | 7,296e-014 | 7,30e-013 | 7,20e-013 |
| Frecuencia                 |                              |            |            |           |           |
| 50,5 [Hz]                  | 1,071e-012                   | 6,994e-014 | 3,899e-013 | 5,08e-014 | 4,86e-013 |
| 69,5 [Hz]                  | 5,897e-013                   | 5,995e-014 | 5,995e-014 | 7,30e-013 | 6,90e-013 |
| Flicker                    | 7,084e-014                   | 6,1222e-05 | 1,2480e-04 | 2,149e-04 | 4,205e-04 |

Tabla 11. Errores en la estimación de la componente de secuencia cero bajo las condiciones de ensayo 2.

| Señal sintética            | Componente de secuencia cero |            |            |           |           |
|----------------------------|------------------------------|------------|------------|-----------|-----------|
|                            | 0%                           | 1,25%      | 2,5%       | 3,75%     | 5%        |
| Armónicos e Interarmónicos | 0,681e-012                   | 6,899e-013 | 7,199e-013 | 7,39e-013 | 7,30e-013 |
| Frecuencia                 |                              |            |            |           |           |
| 49,5 [Hz]                  | 6,778e-013                   | 4,786e-013 | 4,996e-014 | 6,89e-013 | 7,19e-013 |
| 68,5 [Hz]                  | 6,709e-012                   | 6,998e-013 | 6,799e-013 | 6,70e-013 | 7,02e-013 |
| Flicker                    | 7,123e-014                   | 1,6571e-04 | 3,3631e-04 | 1,062e-04 | 1,311e-05 |

Tabla 12. Errores en la estimación de la componente de secuencia cero bajo las condiciones de ensayo 3.

| Señal sintética            | Componente de secuencia cero |            |            |           |           |
|----------------------------|------------------------------|------------|------------|-----------|-----------|
|                            | 0%                           | 1,25%      | 2,5%       | 3,75%     | 5%        |
| Armónicos e Interarmónicos | 0,620e-012                   | 5,799e-013 | 6,097e-013 | 6,30e-013 | 6,19e-013 |
| Frecuencia                 |                              |            |            |           |           |
| 51,5 [Hz]                  | 7,126e-013                   | 7,417e-013 | 7,456e-013 | 7,39e-013 | 7,40e-013 |
| 70,5 [Hz]                  | 6,712e-012                   | 7,298e-013 | 7,096e-013 | 7,30e-013 | 7,39e-013 |
| Flicker                    | 7,071e-014                   | 1,7363e-04 | 3,5239e-04 | 1,306e-04 | 4,617e-05 |

De acuerdo con los resultados de las tablas 10, 11 y 12, los errores obtenidos en las condiciones de ensayo 1, 2 y 3 son inferiores a la incertidumbre (0,15%) de la medida para certificar la clase A. El error mínimo es de 4,996e-014 y el máximo de 6,1222e-04. Además, se verifica que la medida de la componente de secuencia cero no se ve afectada por el ingreso de armónicos e interarmónicos, y de la variación de la frecuencia del sistema desde que se encuentre sincronizada con la frecuencia de muestreo.

Tabla 13. Errores en la estimación de la componente de secuencia cero a 60 [Hz] con una frecuencia de muestreo de 10235 [Hz] (2047 muestras) bajo las condiciones de ensayo 3.

| Señal sintética            | Componente de secuencia cero |            |            |           |           |
|----------------------------|------------------------------|------------|------------|-----------|-----------|
|                            | 0%                           | 1,25%      | 2,5%       | 3,75%     | 5%        |
| Armónicos e Interarmónicos | 0,0244255                    | 6,336e-003 | 6,327e-003 | 6,74e-003 | 6,59e-003 |
| Flicker                    | 7,037e-014                   | 1,3558e-04 | 2,735e-04  | 8,197e-04 | 0,0012361 |

Tabla 14. Errores en la estimación de la componente de secuencia cero a 60 [Hz] con una frecuencia de muestreo de 10245 [Hz] (2049 muestras) bajo las condiciones de ensayo 3.

| Señal sintética            | Componente de secuencia cero |            |            |           |           |
|----------------------------|------------------------------|------------|------------|-----------|-----------|
|                            | 0%                           | 1,25%      | 2,5%       | 3,75%     | 5%        |
| Armónicos e Interarmónicos | 0,012958                     | 6,78e-003  | 7,098e-003 | 7,29e-003 | 8,09e-003 |
| Flicker                    | 7,112e-014                   | 4,8182e-04 | 9,7639e-04 | 0,00107   | 0,001324  |

Como se puede observar en las tablas 13 y 14, los errores que se presentan cuando no hay sincronización de frecuencia son mayores que en el caso sincronizado. Sin embargo, el error máximo en las condiciones de ensayo (0,0244255%) sigue siendo inferior al error permisible (0,15%) para certificar la clase A.

**5.1.1.2 Amplitud de la tensión de alimentación.** Las tablas 15, 16 y 17 muestran los errores en la estimación del valor eficaz de la tensión del sistema bajo las condiciones de los ensayos propuestos en la norma IEC 61000-4-30 (ver Tabla 2).

Tabla 15. Errores en la estimación del valor eficaz de la tensión de alimentación bajo las condiciones de ensayo 1.

| Señal sintética            | Valor eficaz de la tensión |                         |                          |                          |                          |
|----------------------------|----------------------------|-------------------------|--------------------------|--------------------------|--------------------------|
|                            | 50% de U <sub>din</sub>    | 75% de U <sub>din</sub> | 100% de U <sub>din</sub> | 125% de U <sub>din</sub> | 150% de U <sub>din</sub> |
| Armónicos                  | 4,974e-012                 | 1,421e-011              | 0                        | 0                        | 0                        |
| Interarmónicos             | 2,131e-012                 | 7,105e-012              | 0                        | 0                        | 0                        |
| Armónicos e Interarmónicos | 2,842e-012                 | 9,947e-012              | 0                        | 0                        | 0                        |
| Frecuencia                 | 4,263e-012                 | 5,684e-012              | 0                        | 0                        | 0                        |
| Flicker                    | 0,0096680                  | 0,0096442               | 0,0094056                | 0,0096251                | 0,00966                  |

Tabla 16. Errores en la estimación del valor eficaz de la tensión de alimentación bajo las condiciones de ensayo 2.

| Señal sintética            | Valor eficaz de la tensión |                         |                          |                          |                          |
|----------------------------|----------------------------|-------------------------|--------------------------|--------------------------|--------------------------|
|                            | 50% de U <sub>din</sub>    | 75% de U <sub>din</sub> | 100% de U <sub>din</sub> | 125% de U <sub>din</sub> | 150% de U <sub>din</sub> |
| Armónicos                  | 1,207e-011                 | 7,105e-012              | 0                        | 0                        | 0                        |
| Armónicos e Interarmónicos | 7,816e-012                 | 5,684e-011              | 0                        | 0                        | 0                        |
| Frecuencia                 | 4,263e-012                 | 1,279e-011              | 0                        | 0                        | 0                        |
| Flicker                    | 1,184e-013                 | 3,157e-013              | 2,842e-012               | 7,57e-013                | 6,31e-014                |

Tabla 17. Errores en la estimación del valor eficaz de la tensión de alimentación bajo las condiciones de ensayo 3.

| Señal sintética            | Valor eficaz de la tensión |                         |                          |                          |                          |
|----------------------------|----------------------------|-------------------------|--------------------------|--------------------------|--------------------------|
|                            | 50% de U <sub>din</sub>    | 75% de U <sub>din</sub> | 100% de U <sub>din</sub> | 125% de U <sub>din</sub> | 150% de U <sub>din</sub> |
| Armónicos                  | 2,273e-011                 | 1,563e-011              | 0                        | 0                        | 0                        |
| Armónicos e Interarmónicos | 1,776e-011                 | 2,416e-011              | 0                        | 0                        | 0                        |
| Frecuencia                 | 9,237e-012                 | 4,263e-012              | 0                        | 0                        | 0                        |
| Flicker                    | 1,184e-013                 | 3,157e-014              | 1,184e-014               | 7,57e-014                | 6,31e-014                |

Las tablas 15,16 y 17 presentan los errores obtenidos en las condiciones de ensayo 1, 2 y 3 son inferiores a la incertidumbre (0,1%) de la medida para certificar la clase A. El error mínimo es cercano a 0 y el máximo es de 9,6680e-03%. Además, la medida del valor eficaz de la tensión no se ve afectada por la variación de la frecuencia del sistema siempre y cuando se encuentre sincronizada con la frecuencia de muestreo.

Tabla 18. Errores en la estimación del valor eficaz de la tensión de alimentación a 60 Hz con una frecuencia de muestreo de 10235 [Hz] (2047 muestras) bajo las condiciones de ensayo 3.

| Señal sintética            | Valor eficaz de la tensión |                         |                          |                          |                          |
|----------------------------|----------------------------|-------------------------|--------------------------|--------------------------|--------------------------|
|                            | 50% de U <sub>din</sub>    | 75% de U <sub>din</sub> | 100% de U <sub>din</sub> | 125% de U <sub>din</sub> | 150% de U <sub>din</sub> |
| Armónicos                  | 0,023258                   | 0,024406                | 0,023907                 | 0,024417                 | 0,024135                 |
| Interarmónicos             | 0,024417                   | 0,024204                | 0,024417                 | 0,023851                 | 0,024417                 |
| Armónicos e Interarmónicos | 0,027504                   | 0,024417                | 0,023743                 | 0,024417                 | 0,024016                 |
| Flicker                    | 0,0244007                  | 0,0243999               | 0,0240434                | 0,0241006                | 0,024417                 |

Tabla 19. Errores en la estimación del valor eficaz de la tensión de alimentación a 60 Hz con una frecuencia de muestreo de 10245 [Hz] (2049 muestras) bajo las condiciones de ensayo 3.

| Señal sintética            | Valor eficaz de la tensión |                         |                          |                          |                          |
|----------------------------|----------------------------|-------------------------|--------------------------|--------------------------|--------------------------|
|                            | 50% de U <sub>din</sub>    | 75% de U <sub>din</sub> | 100% de U <sub>din</sub> | 125% de U <sub>din</sub> | 150% de U <sub>din</sub> |
| Armónicos                  | 0,022508                   | 0,023365                | 0,023966                 | 0,024068                 | 0,024129                 |
| Interarmónicos             | 0,024095                   | 0,024199                | 0,024243                 | 0,024267                 | 0,024282                 |
| Armónicos e Interarmónicos | 0,023254                   | 0,023397                | 0,023738                 | 0,023918                 | 0,024011                 |
| Flicker                    | 0,0243450                  | 0,0243451               | 0,0243456                | 0,0243458                | 0,0243459                |

Como se observa en las tablas 18 y 19, los errores presentados cuando no hay sincronización de frecuencia son mayores que en el caso sincronizado. Sin

embargo, el error máximo en las condiciones de ensayo (0,024417%) es inferior al error permisible (0,1%) para certificar la clase A.

**5.1.1.3 Sobretensión y subtensión.** En las tablas 20, 21 y 22 se presentan los errores para los valores de tensión y sobretensión. Para los valores de amplitud igual a 50% de  $U_{din}$ , y 75% de  $U_{din}$ , los errores presentados se refieren por subtensión, mientras que para 125% y 150% de  $U_{din}$ , los errores son aplicables a valores de sobretensión. Para 100% de  $U_{din}$  el error calculado equivale a la tolerancia del programa (Matlab) para la medida de sobretensión y subtensión.

Tabla 20. Errores en la estimación de los valores de sobretensión y subtensión bajo las condiciones de ensayo 1.

| Señal sintética            | Sobretensión y subtensión |                  |                   |                   |                   |
|----------------------------|---------------------------|------------------|-------------------|-------------------|-------------------|
|                            | 50% de $U_{din}$          | 75% de $U_{din}$ | 100% de $U_{din}$ | 125% de $U_{din}$ | 150% de $U_{din}$ |
| Armónicos                  | 4,263e-012                | 1,208e-011       | 2,131e-011        | 1,21e-011         | 1,42e-011         |
| Interarmónicos             | 2,131e-012                | 9,947e-012       | 1,184e-011        | 2,13e-011         | 1,63e-011         |
| Armónicos e Interarmónicos | 4,261e-012                | 8,171e-012       | 1,421e-011        | 7,02e-011         | 1,64e-011         |
| Frecuencia                 | 2,124e-012                | 3,908e-012       | 7,10e-011         | 7,11e-011         | 9,24e-012         |
| Flicker                    | 0,0048340                 | 0,0072510        | 0,009668          | 0,012085          | 0,0145021         |

Tabla 21. Errores en la estimación de los valores de sobretensión y subtensión bajo las condiciones de ensayo 2.

| Señal sintética            | Sobretensión y subtensión |                  |                   |                   |                   |
|----------------------------|---------------------------|------------------|-------------------|-------------------|-------------------|
|                            | 50% de $U_{din}$          | 75% de $U_{din}$ | 100% de $U_{din}$ | 125% de $U_{din}$ | 150% de $U_{din}$ |
| Armónicos                  | 9,947e-012                | 6,04e-012        | 1,184e-012        | 1,70e-011         | 2,13e-011         |
| Armónicos e Interarmónicos | 5,684e-012                | 4,974e-012       | 1,591e-011        | 1,39e-011         | 9,95e-011         |
| Frecuencia                 | 2,132e-012                | 1,101e-011       | 5,192e-012        | 4,97e-012         | 2,12e-011         |
| Flicker                    | 7,105e-015                | 2,132e-014       | 2,368e-014        | 9,97e-014         | 4,93e-014         |

Tabla 22. Errores en la estimación de los valores de sobretensión y subtensión bajo las condiciones de ensayo 3.

| Señal sintética            | Sobretensión y subtensión |             |              |              |              |
|----------------------------|---------------------------|-------------|--------------|--------------|--------------|
|                            | 50% de Udin               | 75% de Udin | 100% de Udin | 125% de Udin | 150% de Udin |
| Armónicos                  | 1,918e-011                | 1,314e-011  | 2,605e-011   | 1,71e-011    | 3,27e-011    |
| Armónicos e Interarmónicos | 1,492e-011                | 1,989e-011  | 3,431e-011   | 1,69e-011    | 5,68e-012    |
| Frecuencia                 | 4,974e-012                | 3,908e-012  | 1,539e-012   | 7,11e-012    | 5,69e-012    |
| Flicker                    | 7,105e-015                | 2,136e-014  | 1,184e-014   | 9,94e-014    | 4,97e-014    |

Como se muestra en las tablas 20, 21 y 22, los errores obtenidos en las condiciones de ensayo 1, 2 y 3 son inferiores a la incertidumbre (0,1%) de la medida para certificar la clase A. El error mínimo por subtensión es de 2,131e-012% y el máximo de 7,2510e-03%. El error mínimo por sobretensión es de 5,68e-012% y el máximo de 0,0145021%. Además, la medida de sobretensión y subtensión no se ve afectada de manera apreciable por la variación de la frecuencia del sistema siempre y cuando se encuentre sincronizada con la frecuencia de muestreo.

**5.1.2 Análisis con otras señales de prueba.** Existen otras perturbaciones a parte de los mencionados anteriormente (armónicos, interarmónicos, variaciones de frecuencia, flicker) que afectan las medidas para la monitorización los eventos de estado estacionario, entre las cuales se encuentra las señales de ruido, y la cuantificación de bits en el proceso de muestreo y adquisición de los datos. A continuación, se presenta una serie de pruebas para estimar los errores, cuando se consideran estas perturbaciones en la señal de entrada.

**5.1.2.1 Ruido.** Las clases de señales de ruido incluidas en las pruebas son: el blanco y sus derivados como el rojo, azul, violeta, marrón, rosado. El ruido blanco o Gaussiano presenta una densidad de potencia uniforme sobre todo el espectro

de frecuencia, y es el que se considera regularmente en el tratamiento de señales. A partir del filtrado a baja frecuencia (pasabajas) del ruido blanco se obtiene el ruido rojo. El ruido rosado es el resultado de integrar el ruido blanco mientras que el ruido azul y violeta se obtiene de la diferenciación y doble diferenciación de la señal de ruido blanco respectivamente. En [Villamizar, 2008] se presenta con mayor detalle estudios realizados para el análisis y clasificación de las señales de ruido.

Las señales sintéticas de ruido fueron creadas con la ayuda del Software de Simulación de Perturbaciones de Tensión [Cobos y Vargas, 2009], para un rango de variación de amplitudes de la señal de ruido entre 1 y 30 Voltios pico a pico.

Las tablas 23 a 28 presentan los errores debidos a las diferentes clases de señales de ruido mencionadas previamente.

Tabla 23. Errores máximos en la medida de los eventos de estado estacionario cuando se presenta el fenómeno de ruido blanco.

| Amplitud Señal sintética [Vpp] | Eventos de estado estacionario |                        |                           |
|--------------------------------|--------------------------------|------------------------|---------------------------|
|                                | Desbalance                     | Amplitud de la tensión | Sobretensión y Subtensión |
| 1                              | 2,89389e-03                    | 2,044917e-03           | 0,003635588               |
| 5                              | 9,37055e-03                    | 0,021843333            | 0,037393568               |
| 10                             | 0,019870643                    | 0,105779666            | 0,106547687               |
| 15                             | 0,026926271                    | 0,135016666            | 0,238478093               |
| 20                             | 0,033703189                    | 0,263645333            | 0,275636040               |
| 25                             | 0,085139241                    | 0,452381666            | 0,388860559               |
| 30                             | 0,093805611                    | 0,548914916            | 0,457297505               |

Tabla 24. Errores máximos en la medida de los eventos de estado estacionario cuando se presenta el fenómeno de ruido rojo.

| Amplitud Señal sintética [Vpp] | Eventos de estado estacionario |                        |                           |
|--------------------------------|--------------------------------|------------------------|---------------------------|
|                                | Desbalance                     | Amplitud de la tensión | Sobretensión y Subtensión |
| 1                              | 7,0166e-014                    | 1,0034995e-08          | 1,0034722e-08             |
| 5                              | 2,7404e-012                    | 2,5086832e-09          | 2,5086811e-07             |
| 10                             | 4,1602e-012                    | 1,0034725e-06          | 1,0034721e-06             |
| 15                             | 4,1602e-012                    | 2,2578124e-06          | 2,2578124e-06             |
| 20                             | 4,1602e-012                    | 4,0138891e-06          | 4,0138878e-06             |
| 25                             | 4,1602e-012                    | 6,2717008e-06          | 6,2717012e-06             |
| 30                             | 4,1602e-012                    | 9,0312491e-06          | 9,0312496e-06             |

Tabla 25. Errores máximos en la medida de los eventos de estado estacionario cuando se presenta el fenómeno de ruido azul.

| Amplitud Señal sintética [Vpp] | Eventos de estado estacionario |                        |                           |
|--------------------------------|--------------------------------|------------------------|---------------------------|
|                                | Desbalance                     | Amplitud de la tensión | Sobretensión y Subtensión |
| 1                              | 1,079912e-04                   | 0,001165712            | 0,00102548101             |
| 5                              | 1,495039e-04                   | 0,015062267            | 0,01545227512             |
| 10                             | 8,694743e-04                   | 0,062481911            | 0,06050259018             |
| 15                             | 0,00119186                     | 0,1430622269           | 0,1329324486              |
| 20                             | 0,00246977                     | 0,2459426517           | 0,2505102258              |
| 25                             | 0,00354871                     | 0,3821057349           | 0,3770491616              |
| 30                             | 0,00518856                     | 0,5274690745           | 0,5400412377              |

Tabla 26. Errores máximos en la medida de los eventos de estado estacionario cuando se presenta el fenómeno de ruido violeta.

| Amplitud Señal sintética [Vpp] | Eventos de estado estacionario |                        |                           |
|--------------------------------|--------------------------------|------------------------|---------------------------|
|                                | Desbalance                     | Amplitud de la tensión | Sobretensión y Subtensión |
| 1                              | 4,66506e-06                    | 0,0018252515           | 0,0017314155              |
| 5                              | 8,67291e-06                    | 0,0442236727           | 0,0455262359              |
| 10                             | 2,03929e-05                    | 0,1735127219           | 0,1830423368              |
| 15                             | 1,11028e-04                    | 0,4130591756           | 0,4275961753              |
| 20                             | 1,45611e-04                    | 0,7045997227           | 0,7303963861              |
| 25                             | 1,65441e-04                    | 1,1446181806           | 1,1495684678              |
| 30                             | 2,08318e-04                    | 1,5318900985           | 1,6176098140              |

Tabla 27. Errores máximos en la medida de los eventos de estado estacionario cuando se presenta el fenómeno de ruido marrón.

| Amplitud Señal sintética [Vpp] | Eventos de estado estacionario |                        |                           |
|--------------------------------|--------------------------------|------------------------|---------------------------|
|                                | Desbalance                     | Amplitud de la tensión | Sobretensión y Subtensión |
| 1                              | 1,9989e-010                    | 6,116707e-010          | 9,9246241e-010            |
| 5                              | 4,2541e-010                    | 7,9599971e-09          | 8,51085720e-09            |
| 10                             | 2,05510e-09                    | 1,2495000e-08          | 1,24948703e-08            |
| 15                             | 3,28582e-09                    | 1,7780005e-08          | 1,61852753e-08            |
| 20                             | 3,46533e-09                    | 3,5914161e-08          | 2,45816877e-08            |
| 25                             | 7,32445e-09                    | 3,9249916e-08          | 3,28676345e-08            |
| 30                             | 9,41324e-09                    | 4,1914169e-08          | 4,08578411e-08            |

Tabla 28. Errores máximos en la medida de los eventos de estado estacionario cuando se presenta el fenómeno de ruido rosado.

| Amplitud Señal sintética [Vpp] | Eventos de estado estacionario |                        |                           |
|--------------------------------|--------------------------------|------------------------|---------------------------|
|                                | Desbalance                     | Amplitud de la tensión | Sobretensión y Subtensión |
| 1                              | 7,067287e-06                   | 4,4947814e-05          | 5,250088647e-05           |
| 5                              | 3,819130e-05                   | 6,4576122e-05          | 1,209031725e-04           |
| 10                             | 5,030893e-05                   | 2,6097577e-04          | 5,79790294e-04            |
| 15                             | 1,505339e-04                   | 3,5862073e-04          | 5,96404308e-04            |
| 20                             | 3,132555e-04                   | 9,2714211e-04          | 9,43084337e-04            |
| 25                             | 5,186215e-04                   | 0,0013609057           | 0,00163547607             |
| 30                             | 6,87435e-04                    | 0,0014578671           | 0,00174149221             |

En los resultados mostrados en las tablas 23 a 28, se puede observar que el efecto de la señal de ruido es más apreciable para las estimaciones del valor eficaz y las perturbaciones de sobretensión y subtensión que en la estimación del desbalance. Además, se concluye que los mayores errores se obtienen con la señal de ruido blanco, mientras que los más pequeños ocurren con el ruido rojo y marrón. En el caso del desbalance, los errores bajo todas las señales de ruido son inferiores al permisible para certificar la clase A (0,15%), mientras que para la estimación del valor eficaz de la tensión y los valores de sobretensión, subtensión son superiores al máximo permisible por el norma IEC 61000-4-30 (0,1%) cuando la señal de ruido es blanco, azul y violeta con una amplitud de tensión pico a pico superior a 10 [V].

**5.1.2.2 Cuantificación de la señal.** Los resultados de la prueba se muestran en la Tabla 29 y con ello se busca establecer los errores que se tienen al realizar la conversión analógica digital de una señal considerando el efecto de los bits del convertidor, para lo cual se consideraron los efectos de convertidores de 2 a 20, considerando que la señal de ruido introducida tiene una distribución de probabilidad uniforme con valores máximo y mínimo determinados por la resolución establecida por los bits del convertidor.

Tabla 29. Errores máximos por la cuantificación uniforme de la señal para la medida de los eventos de estado estacionario.

| Señal sintética [Nº-Bits] | Eventos de estado estacionario |                        |                           |
|---------------------------|--------------------------------|------------------------|---------------------------|
|                           | Desbalance                     | Amplitud de la tensión | Sobretensión y Subtensión |
| 2                         | 1,7601073816                   | 7,1428571428           | 8,33333333333             |
| 4                         | 0,1670163014                   | 0,6644591369           | 0,66450694333             |
| 6                         | 0,0168286891                   | 0,0613946077           | 0,06146788011             |
| 8                         | 0,0086879385                   | 0,0286909931           | 0,02806219097             |
| 10                        | 0,0015066690                   | 0,0048134153           | 0,00484161759             |
| 12                        | 1,876900e-04                   | 5,9782665e-04          | 5,9767555e-04             |
| 14                        | 1,057334e-04                   | 3,3552832e-04          | 3,3552712e-04             |
| 16                        | 5,613021e-06                   | 1,7819587e-05          | 1,78195910e-05            |
| 18                        | 2,820561e-06                   | 8,9541576e-06          | 8,9544564e-06             |
| 20                        | 1,214359e-06                   | 3,8478444e-06          | 3,8476438e-06             |

De los resultados obtenidos (ver Tabla 29), se puede observar que el uso de una mayor cantidad de bits para representar o codificar la señal de entrada reduce los errores de cuantificación en la medida de las perturbaciones de tensión pero en la implementación del conversor analógico digital se requiere de circuitos más complejos y costosos. Además, se puede observar que a partir de 6 bits se cumple con la exigencia clase A, ya que el error para el desbalance es inferior al 0,15% y para el valor eficaz de la tensión y los valores de sobretensión y subtensión el error es inferior al 0,1%.

## 5.2 PRUEBAS REALIZADAS AL ALGORITMO DE MEDIDA PARA LA MONITORIZACIÓN DE EVENTOS DE ESTADO TRANSITORIO.

Para establecer las incertidumbres del algoritmo de medición de perturbaciones transitorias de tensión se realizó un proceso similar al de los eventos de estado estacionario con algunas modificaciones. La primera modificación es que no se

utilizará la tabla 1 de la sección 6.1 de la norma IEC 61000-4-30 donde presenta los márgenes de variación de las perturbaciones de tensión. La segunda modificación esta relacionada a la no utilización de la metodología planteada en la sección 6.2 de la norma IEC 61000-4-30 donde verifican la realización de los métodos de medida para las perturbaciones de estado estacionario. A continuación se describe como se generaron las señales sintéticas para diferentes condiciones de ensayo.

**5.2.1 Características de las señales sintéticas de prueba.** Teniendo en cuenta las condiciones de ensayo descritas en la tabla 2 que contienen las características de la señal sintética se tomarán cinco puntos equidistantes en el intervalo de cada perturbación y cada prueba queda de la siguiente manera:

Condiciones de ensayo 1.

Frecuencia: 59,5; 59,75; 60; 60,25; 60,5 [Hz]

Armónicos: 0; 0,75; 1,5; 2,25; 3 [% de U<sub>din</sub>]      Componentes: 3, 5 y 29

Interarmónicos: 0; 0,125; 0,25; 0,375; 0,5 [%]      Componente interarmónica:  
7,5\*F<sub>nom</sub>

Condiciones de ensayo 2.

Frecuencia: 58,5; 58,75; 59; 59,25; 59,5 [Hz]

Armónicos: 7; 8,5; 10; 11,5; 13 [% de U<sub>din</sub>]      Componente armónica: 3.

2; 3,5; 5; 6,5; 8 [% de U<sub>din</sub>]      Componente armónica: 5 y 29.

Interarmónico: 0,5; 0,75; 1; 1,25; 1,5 [% de U<sub>din</sub>]      Componente interarmónica:  
7,5\*F<sub>nom</sub>

Condiciones de ensayo 3.

Frecuencia: 60,5; 60,75; 61; 61,25; 61,5 [Hz]

Armónicos: 7; 8,5; 10; 11,5; 13 [% de Udin]      Componente armónica: 7.

2; 3,5; 5; 6,5; 8 [% de Udin]      Componente armónica: 13 y 25.

Interarmónico: 0,5; 0,75; 1; 1,25; 1,5 [% de Udin] Componente interarmónica:  
3.5\*Fnom

Las pruebas se dividen en cuatro partes: La primera se realiza solo considerando componentes armónicas y se llama **armónicos**, la segunda se realiza considerando componentes interarmónicas y se llama **interarmónicos**, la tercera es la mezcla de las dos anteriores, llamada **armónicos e interarmónicos**, cabe resaltar que las amplitudes de los armónicos son iguales con el fin de reducir la cantidad de pruebas. Los puntos equidistante para la frecuencia se utilizarán para determinar la frecuencia de muestreo y realizar pruebas cuando existe desincronización en el sistema (la frecuencia del conversor analógico digital no está sincronizada con la frecuencia de la red), en este caso la frecuencia de la red es constante igual a 60 [Hz] y la frecuencia de muestreo se modificará respecto a la frecuencias de las condiciones de ensayo.

Las características de cada una de las pruebas son las siguientes:

- **Pruebas armónicos.** Se realizan cinco pruebas que corresponde a un punto equidistante del intervalo de cada perturbación. Todas las amplitudes de las componentes armónicas se consideran iguales.
- **Pruebas interarmónicos.** Se realizaron cinco pruebas que corresponde a un punto equidistante del intervalo de cada perturbación. Todas las amplitudes de las componentes interarmónicas son iguales.
- **Pruebas armónicas e interarmónico.** Estas pruebas se realizaron con los cinco puntos equidistantes de los armónicos e interarmónicos.

La señal sintética de la prueba monofásica se generó de forma discreta y continua con el archivo `generador_señal_conformidad_un_tramo.m`. Este archivo genera una señal dividida en tres tramos para simular un evento, como se observa en la Figura 17. En el anexo D2-b se encuentra el archivo en Matlab para la generación de las señales

En la generación de esta señal es posible modificar la frecuencia de cada tramo al igual que las amplitudes de la componente fundamental, las componentes armónicas y las componentes interarmónicas, así como el comienzo y final de cada tramo. Todos los tramos tienen las mismas características de armónicos e interarmónicos. Cabe señalar que la señal discreta se encuentra muestreada de forma sincronizada con la frecuencia de la red. En el anexo D2-b se describe este procedimiento.

El archivo `generador_senal_conformidad_un_tramo_continuo.m` (ANEXO D3) contiene la señal continua con iguales características de la señal discreta, la salida de este archivo es la señal de la red elevada al cuadrado ya que se utilizará para evaluar la tensión eficaz de forma continua con el comando `QUAD` para determinar la integral de la señal en la ventana de observación.

El archivo `rms_analogico_un_tramo.m` (ANEXO D4) realiza el mismo procedimiento del algoritmo de la unidad constructiva pero considerando que la señal es continua en el tiempo. En este programa se utiliza el comando `QUAD` presentado en el párrafo anterior. Sin embargo este proceso no es viable totalmente ya que la función `Quad` está en función de la tolerancia, entre más pequeño sea este valor más preciso es el resultado, en las pruebas iniciales se evidenció que al reducir la tolerancia se producían resultados dudosos por lo cual

se aumenta este valor para evitarlos pero se perdía precisión, los resultados presentados más adelante fueron desarrollados con una tolerancia de  $10^{-10}$ .

La señal discreta es la que ingresa al algoritmo llamado `medidor_monofasico_unidad_constructiva.m` (ANEXO D1-a) y `medidor_trifásico_unidad_constructiva.m` (ANEXO D1-b) para ser evaluada. Las entradas y salidas del algoritmo son descritas en su respectivo anexo.

Finalmente se compara el resultado del proceso con la señal continua con el obtenido con la señal discreta para determinar las incertidumbres del algoritmo.

Como el valor eficaz de tensión depende de la amplitud de las componentes armónicas, no es posible que este valor sea constante para todas las pruebas, por tanto la amplitud de la componente fundamental es constante y las amplitudes de las componentes armónicas e interarmónicas serán modificadas en función de los porcentajes presentados en la Tabla 2.

**5.2.2 Análisis de resultados.** Las señales sintéticas de prueba fueron creadas con el generador de perturbaciones. Al final de este programa se encuentra el comando `save` que guarda el archivo de texto `Prueba_XXX_datos`, el cual almacena las características principales de la señal: amplitud de las componentes armónicas e interarmónicas, frecuencia de la red, entre otras, con el fin de reconstruirlas posteriormente con el generador de perturbaciones.

Las tablas 30 a 35 contienen el número de la prueba y la diferencia entre el valor real y el resultado del algoritmo; para que este cumpla con lo propuesto en la norma IEC 61000-4-30 la diferencia debe ser menor a 0,254.

**5.2.2.1 Huecos de tensión.** Las pruebas presentadas a continuación fueron realizadas considerando que: la duración del hueco de tensión es de 0,5 [s], y que

la componente fundamental tiene un valor eficaz de 9 [V] (que corresponde al 7,09% de la tensión nominal). Las características de las componentes armónicas e interarmónicas son presentadas en cada una de las tablas, dependiendo de la respectiva condición de ensayo; además se considera que el sistema se encuentra sincronizado a la frecuencia de la red. Los detalles de la señal sintética se encuentran en los archivos de texto llamados prueba\_xxx\_datos.

Las tablas 30, 31 y 32 muestran los errores para las condiciones de: ensayo 1, ensayo 2 y ensayo 3, respectivamente.

Tabla 30. Errores en la medición de huecos de tensión bajo las condiciones de ensayo 1

| <b>Señal sintética</b>     | <b>NºPrueba</b> | <b>Diferencia</b> |
|----------------------------|-----------------|-------------------|
| Armónicos                  | 1               | 0,000000000000020 |
|                            | 2               | 0,00000000012013  |
|                            | 3               | 0,00000000001459  |
|                            | 4               | 0,00000000378615  |
|                            | 5               | 0,00000000499765  |
| Interarmónicos             | 6               | 0,00000878160147  |
|                            | 7               | 0,00001756166198  |
|                            | 8               | 0,00002633603973  |
|                            | 9               | 0,00003510093294  |
| Armónicos e Interarmónicos | 10              | 0,00003364458387  |
|                            | 11              | 0,00011438296681  |
|                            | 12              | 0,00023556973846  |
|                            | 13              | 0,00038861089354  |

Tabla 31. Errores en la medida de huecos de tensión bajo las condiciones de ensayo 2

| <b>Señal sintética</b>     | <b>NºPrueba</b> | <b>Diferencia</b> |
|----------------------------|-----------------|-------------------|
| Armónicos                  | 18              | -0,00000000218791 |
|                            | 19              | -0,00000000016797 |
|                            | 20              | -0,00000000272600 |
|                            | 21              | -0,00000001247970 |
|                            | 22              | -0,00000000015070 |
| Interarmónicos             | 23              | 0,00003510093294  |
|                            | 24              | 0,00005258459312  |
|                            | 25              | 0,00006998200937  |
|                            | 26              | 0,00008725957191  |
|                            | 27              | 0,00010439043057  |
| Armónicos e Interarmónicos | 28              | 0,00028451557404  |
|                            | 29              | 0,00059709721720  |
|                            | 30              | 0,00096180607480  |
|                            | 31              | 0,00135197225910  |
|                            | 32              | 0,00175346704400  |

Tabla 32. Errores en la medida de huecos de tensión bajo las condiciones de ensayo 3

| <b>Señal sintética</b>     | <b>NºPrueba</b> | <b>Diferencia</b> |
|----------------------------|-----------------|-------------------|
| Armónicos                  | 38              | 0,00000000027236  |
|                            | 39              | -0,00000000036063 |
|                            | 40              | -0,00000000011870 |
|                            | 41              | -0,00000000100220 |
|                            | 42              | -0,00000000219080 |
| Interarmónicos             | 43              | 0,00003495871707  |
|                            | 44              | 0,00005243136178  |
|                            | 45              | 0,00006985494826  |
|                            | 46              | 0,00008719767567  |
|                            | 47              | 0,00010442789001  |
| Armónicos e Interarmónicos | 48              | 0,00010789637963  |
|                            | 49              | 0,00032353700720  |
|                            | 50              | 0,00059434870670  |
|                            | 51              | 0,00089491290000  |
|                            | 52              | 0,00121080065230  |

Para las condiciones de ensayo uno los errores son pequeños y aumentan a medida que se aumentan las amplitudes de los armónicos e interarmónicos. Los resultados de las pruebas que consideran contenido de componentes armónicas e interarmónicas son los que presentan errores mayores de los ensayos, debido a que el algoritmo de valor eficaz es un filtro que atenúa las componentes armónicas [Duarte, 2004]. Sin embargo, la componente interarmónica no es eliminada produciendo un error en la estimación.

**5.2.2.2 Sobretensión temporal.** Las tablas 33, 34 y 35 muestran los errores para las condiciones de: ensayo 1, ensayo 2 y ensayo 3, respectivamente.

Tabla 33. Errores en la medición de sobretensiones temporales bajo las condiciones de ensayo 1

| Señal sintética            | NºPrueba | Diferencia         |
|----------------------------|----------|--------------------|
| Armónicos                  | 96       | 0,0000000000000000 |
|                            | 97       | 0,00000000165801   |
|                            | 98       | 0,0000000000000300 |
|                            | 99       | 0,0000000000000600 |
|                            | 100      | 0,00000000860300   |
| Interarmónicos             | 101      | -0,000008780939    |
|                            | 102      | -0,000017561705    |
|                            | 103      | -0,000026342298    |
|                            | 104      | -0,000035122697    |
| Armónicos e Interarmónicos | 105      | -0,000010288929    |
|                            | 106      | -0,000023591973    |
|                            | 107      | -0,000039905719    |
|                            | 108      | -0,000059225775    |

Tabla 34. Errores en la medida de sobretensiones temporales bajo las condiciones de ensayo 2

| <b>Señal sintética</b>     | <b>NºPrueba</b> | <b>Diferencia</b> |
|----------------------------|-----------------|-------------------|
| Armónicos                  | 113             | 0,000000000005    |
|                            | 114             | 0,000000000469    |
|                            | 115             | 0,000000000031    |
|                            | 116             | 0,000000000041    |
|                            | 117             | 0,000000000006    |
| Interarmónicos             | 118             | -0,000035122697   |
|                            | 119             | -0,000052682834   |
|                            | 120             | -0,000070242008   |
|                            | 121             | -0,000087800101   |
|                            | 122             | -0,000105356994   |
| Armónicos e Interarmónicos | 123             | -0,000054310230   |
|                            | 124             | -0,000099471985   |
|                            | 125             | -0,000156564123   |
|                            | 126             | -0,000225509543   |
|                            | 127             | -0,000306214335   |

Tabla 35. Errores en la medida de sobretensiones temporales bajo las condiciones de ensayo 3.

| <b>Señal sintética</b>     | <b>NºPrueba</b> | <b>Diferencia</b> |
|----------------------------|-----------------|-------------------|
| Armónicos                  | 133             | 0,000000000000    |
|                            | 134             | 0,000000000022    |
|                            | 135             | 0,000000000021    |
|                            | 136             | 0,000000000005    |
|                            | 137             | 0,000000000036    |
| Interarmónicos             | 138             | -0,000034897795   |
|                            | 139             | -0,000052341962   |
|                            | 140             | -0,000069782822   |
|                            | 141             | -0,000087220257   |
|                            | 142             | -0,000104654153   |
| Armónicos e Interarmónicos | 143             | -0,000040840307   |
|                            | 144             | -0,000076161652   |
|                            | 145             | -0,000121355539   |
|                            | 146             | -0,000176358087   |
|                            | 147             | -0,000241090513   |

Los errores de la estimación de la sobretensión temporal son menores comparadas con los obtenidos en la estimación de los huecos de tensión, debido a que la amplitud de las componentes armónicas e interarmónica son menores comparadas con la amplitud del evento. Para el hueco de tensión simulado, la amplitud de la componente fundamental es comparable con la amplitud de las componentes armónicas e interarmónicas razón por la cual el efecto de estas componentes es importante en los errores de estimación.

**5.2.2.3 Comparación de los errores entre la estimación del valor eficaz de la tensión con muestras constantes y con muestras variables con falta de sincronismo.** Los resultados mostrados en las tablas 36 a 38 comparan los errores en la estimación del valor eficaz de una señal de tensión con sólo componente fundamental y considerando que la frecuencia es de 60 [Hz]. La amplitud del hueco de tensión es de 40 [Vp] (que corresponde a un 22,27% del valor nominal) y se considera que la frecuencia de muestreo varía de acuerdo al rango de frecuencia establecido en la sección 5.2.1.

Tabla 36. Errores en la medición del valor eficaz bajo desincronización de frecuencia en las condiciones de ensayo 1

|            | Frecuencia de muestreo [ Hz] | Algoritmo muestra variable | Algoritmo muestra fija |
|------------|------------------------------|----------------------------|------------------------|
|            |                              | Diferencia 1               | Diferencia 2           |
| Prueba 153 | 7616                         | 0,0074276875514            | 0,1180852230114        |
| Prueba 154 | 7648                         | 0,0590038701945            | 0,0590038701945        |
| Prueba 155 | 7712                         | 0,0512225044731            | -0,0587433814351       |
| Prueba 156 | 7744                         | 0,1017208296160            | -0,1170808314110       |

Tabla 37. Errores en la medición del valor eficaz bajo desincronización de frecuencia en las condiciones de ensayo 2

|            | Frecuencia de muestreo [Hz] | Algoritmo muestra variable | Algoritmo muestra fija |
|------------|-----------------------------|----------------------------|------------------------|
|            |                             | Diferencia 1               | Diferencia 2           |
| Prueba 157 | 7488                        | 0,0226456619133            | 0,3522095869002        |
| Prueba 158 | 7520                        | 0,0749323388327            | 0,2943808706293        |
| Prueba 159 | 7552                        | 0,0149752109982            | 0,2358678622594        |
| Prueba 160 | 7584                        | 0,0669099953576            | 0,1769987253651        |
| Prueba 161 | 7616                        | 0,0074276875514            | 0,1180852230114        |

Tabla 38. Errores en la medición del valor eficaz bajo desincronización de frecuencia en las condiciones de ensayo 3

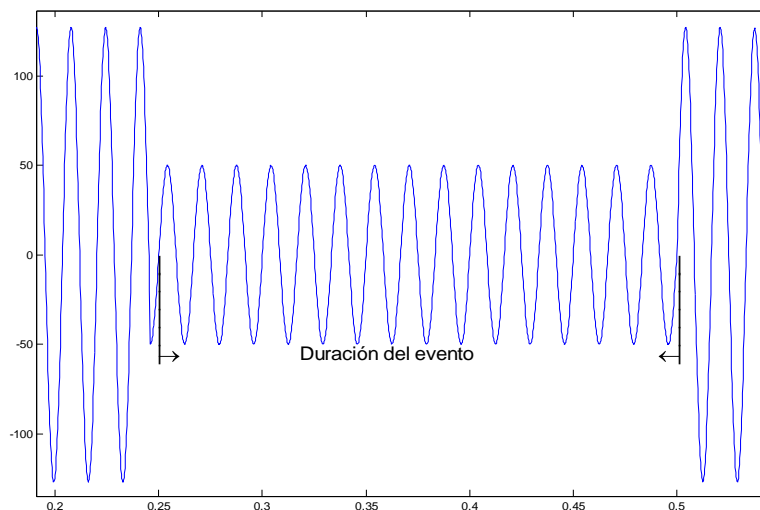
|            | Frecuencia de muestreo [Hz] | Algoritmo muestra variable | Algoritmo muestra fija |
|------------|-----------------------------|----------------------------|------------------------|
|            |                             | Diferencia 1               | Diferencia 2           |
| Prueba 162 | 7744                        | 0,1017208296160            | -0,1170808314110       |
| Prueba 163 | 7776                        | 0,0435600828331            | -0,1747272882578       |
| Prueba 164 | 7808                        | 0,0937255409358            | -0,2319610705110       |
| Prueba 165 | 7840                        | 0,0360203440070            | -0,2879121015272       |
| Prueba 166 | 7872                        | 0,0858480600268            | -0,3431219808904       |

Los resultados de las tablas 36, 37 y 38 muestran que el algoritmo con muestra variable permite mantener el error en niveles menores o iguales a 0,101720V, sin importar la frecuencia de muestreo. Además, los resultados son menores a la incertidumbre establecida en la norma IEC 61000-4-30 de  $\pm 0,2\%$  de  $U_{din}$ , es decir  $\pm 0,254V$  para una señal con una tensión eficaz de 127 [Vrms]. En el caso de muestra fija se observa que el sistema tiene un rango en la frecuencia de muestreo, entre 7584 y 7776 [Hz], en el cual se mantiene en el rango de incertidumbre de la clase A.

**5.2.2.4 Incertidumbre de la duración de la medida.** La duración de un evento es presentado en la Figura 22, de acuerdo con [Markiewicz y Klajn, 2004] el inicio y final de un evento comienza en el primer y último cruce por cero del evento respectivamente. Teniendo lo anterior, se presentará a continuación una serie de ensayos con señales sintéticas para determinar si cumple con la incertidumbre de la estimación de la duración del evento. Para comparar los algoritmos de muestra variable y muestra fija, se modificará la frecuencia de muestreo para verificar el desempeño de cada uno, se supone que la frecuencia de la red está a 60 [Hz] y la frecuencia medida es 58, 59, 60, 61 y 62 [Hz]. La frecuencia de muestreo es el producto entre la frecuencia medida y la cantidad de muestras por ciclo (en este caso es de 128 muestras/ciclo). A la frecuencia de muestreo de 7680 [Hz] el sistema se encuentra sincronizado con la frecuencia de la red.

La amplitud del hueco de tensión es de 50 [Vp] (corresponde a un 27,83% de la tensión nominal) durante el intervalo en que ocurre el evento. En las tablas 39 a 42 se muestran los errores en la estimación de la duración del hueco de tensión considerando diferentes tiempos y los algoritmos de muestras fija y variable.

Figura 22. Representación del inicio y final de la duración del evento.



Fuente: Autores

Tabla 39. Errores en la estimación de la duración de un hueco de tensión de duración 0,2164 [s]

| Frecuencia de muestreo [Hz] | Duración muestra fija [s] | Error [ms] | Duración muestra variable [s] | Error [ms] |
|-----------------------------|---------------------------|------------|-------------------------------|------------|
| 7424                        | 0,232758                  | 16,35      | 0,225484                      | 9,084      |
| 7552                        | 0,2203389                 | 3,938      | 0,216896                      | 0,496      |
| 7680                        | 0,2083333                 | 8,066      | 0,2083333                     | 8,0666     |
| 7808                        | 0,1967213                 | 19,67      | 0,19979                       | 16,61      |
| 7936                        | 0,193548                  | 22,85      | 0,19959                       | 16,81      |

Tabla 40. Errores en la estimación de la duración de un hueco de tensión de duración 0,3666 [s]

| Frecuencia de muestreo [Hz] | Duración muestra fija [s] | Error [ms] | Duración muestra variable [s] | Error [ms] |
|-----------------------------|---------------------------|------------|-------------------------------|------------|
| 7424                        | 0,387931                  | 21,2643    | 0,375808                      | 9,14134    |
| 7552                        | 0,3813559                 | 14,689     | 0,375397                      | 8,73034    |
| 7680                        | 0,36666                   | 0          | 0,36666                       | 0          |
| 7808                        | 0,352459                  | 14,207     | 0,357966                      | 8,7006     |
| 7936                        | 0,330645                  | 36,02166   | 0,3435609                     | 23,10576   |

Tabla 41. Errores en la estimación de la duración de un hueco de tensión de duración 0,4248 [s]

| Frecuencia de muestreo [Hz] | Duración muestra fija [s] | Error [ms] | Duración muestra variable [s] | Error [ms] |
|-----------------------------|---------------------------|------------|-------------------------------|------------|
| 7424                        | 0,456896                  | 32,096     | 0,442618                      | 17,818     |
| 7552                        | 0,432203                  | 7,403      | 0,42545                       | 0,65       |
| 7680                        | 0,416666                  | 8,1334     | 0,41666                       | 8,1334     |
| 7808                        | 0,409836                  | 14,964     | 0,416239                      | 21,439     |
| 7936                        | 0,395161                  | 29,639     | 0,40751                       | 17,29      |

Tabla 42. Errores en la estimación de la duración de un hueco de tensión de duración 0,3082 [s].

| Frecuencia de muestreo [Hz] | Duración muestra fija [s] | Error [ms] | Duración muestra variable [s] | Error [ms] |
|-----------------------------|---------------------------|------------|-------------------------------|------------|
| 7424                        | 0,336206                  | 28,006m    | 0.3257                        | 17,5       |
| 7552                        | 0,322033                  | 13,833m    | 0,317002                      | 8,802      |
| 7680                        | 0,3                       | 8,2m       | 0,3                           | 8,2        |
| 7808                        | 0,303278                  | 4,922 m    | 0,308017                      | 0,183      |
| 7936                        | 0,290322                  | 17,878m    | 0,301663                      | 6,537      |

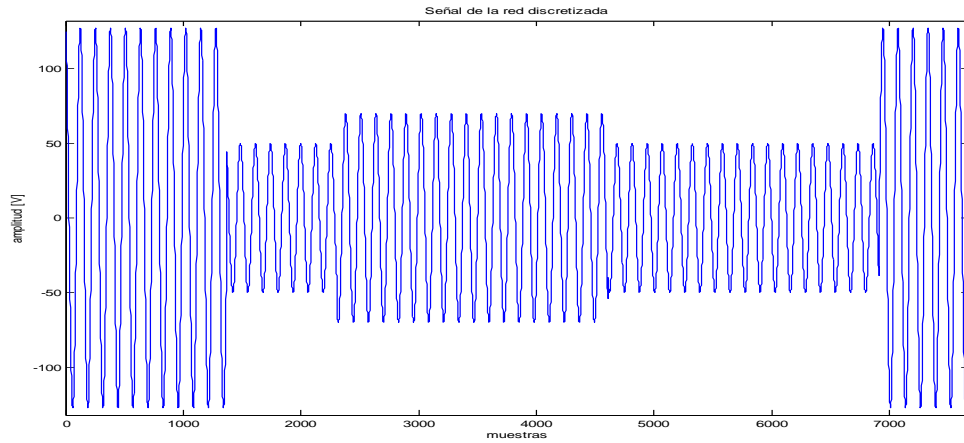
Los resultados de las tablas 39 a 42 muestran que el algoritmo de muestra variable presenta el mejor comportamiento en cuanto que genera una menor incertidumbre en la estimación de la duración del hueco de tensión.

### 5.2.3 Análisis con otras señales de prueba.

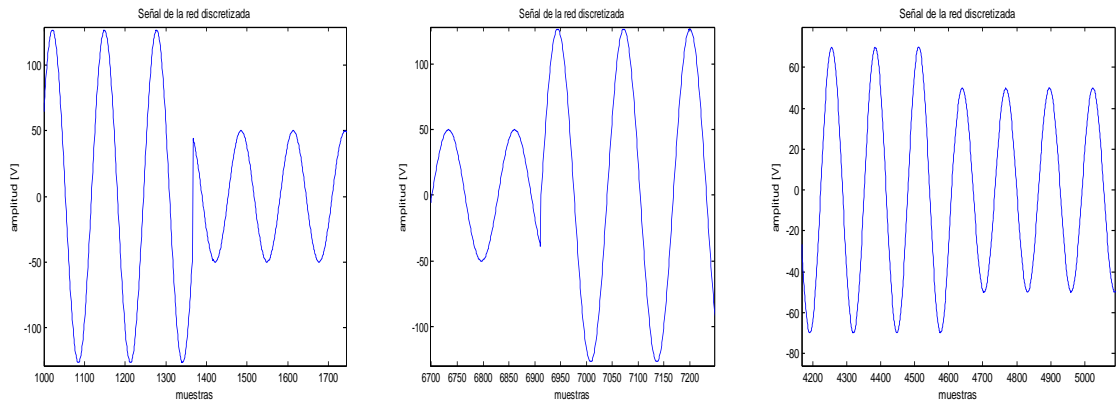
Los resultados previos permiten establecer que el algoritmo de muestra variable genera una mejor incertidumbre en la estimación de los parámetros que caracterizan un hueco de tensión. A continuación se presentará un análisis con diversas señales sintéticas ideales diferentes a las estipuladas en el estándar IEC 61000-4-30 con el fin de verificar el comportamiento de dicho algoritmo.

La señal de la Figura 23.a es un hueco de tensión ideal con sólo componente fundamental y con dos cambios en la amplitud de la señal, la amplitud mínima de 50 [Vp] y la amplitud máxima de 70 [Vp]. La amplitud en estado estacionario es de 127 [Vp].

Figura 23. Hueco de tensión ideal con dos cambios en la amplitud de la señal



a) Señal de la red discretizada



b) Aumento de la señal de la red en los cambios de amplitud donde comenzó (figura izquierda) y finalizó (figura central) el hueco de tensión, y donde finalizó el cambio de amplitud (figura derecha).

Fuente: Autores

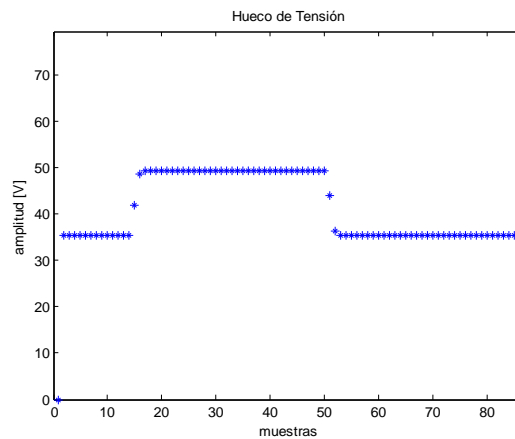
La Figura 23.b presenta un aumento de la señal de la Figura 23.a con el fin de resaltar los cambios ocurridos al comienzo y final del hueco de tensión.

La Figura 24 presenta el resultado de la estimación del valor eficaz con el algoritmo con muestra variable durante la perturbación. Se observa que este

algoritmo sigue de forma adecuada la señal de la red. La definición del estándar [IEC 61000-4-30, 2003] es claro en señalar que el valor eficaz debe ser medido desde el primer cruce por cero de la señal de la red durante un periodo, por tanto los cambios de fase debido a un hueco de tensión o una sobretensión temporal no debe ser tenidos en cuenta por el algoritmo.

Si la señal de tensión del sistema presenta un cambio en su amplitud de forma suave sin que se produzca un cambio de fase o un cambio brusco en la amplitud como se muestra en la tercera gráfica de la Figura 23.b, el algoritmo no lo detectará y realizará la medición según lo estipulado en el estándar [IEC 61000-4-30, 2003], esta es la razón por la cual se observan dos tensiones eficaces en la Figura 24 entre las dos medidas de huecos de tensión constantes.

Figura 24. Resultado del algoritmo de muestra variable.

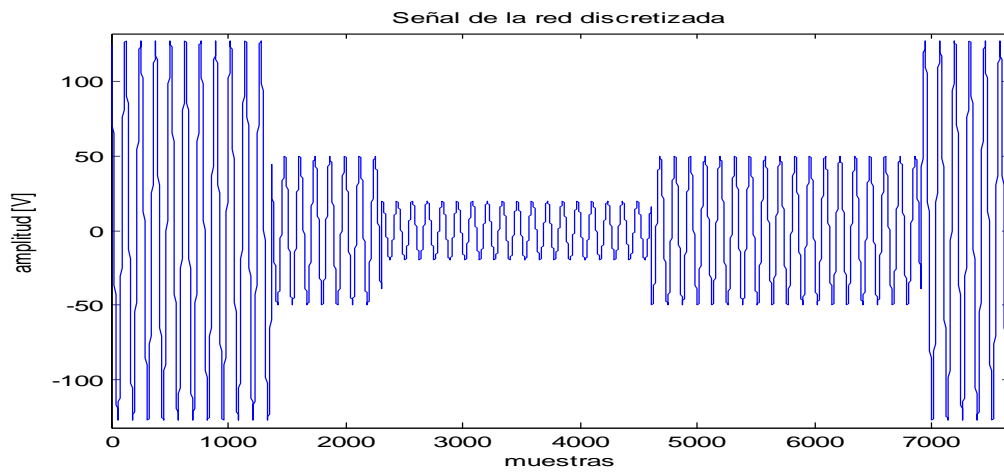


Fuente: Autores

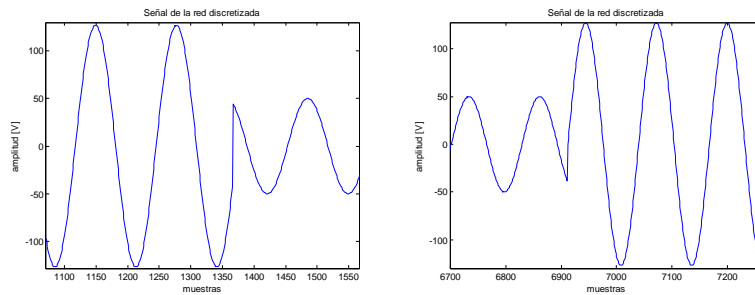
Ahora se presentarán los resultados obtenidos al modificar la señal de la Figura 23.a con el objetivo de generar además de los cambios de amplitud, cambios de fase tanto en el inicio como final del hueco. La Figura 25 muestra la señal

resultante, resaltando los cambios de fase al inicio y la final de la perturbación (ver literales b y c de la Figura 25).

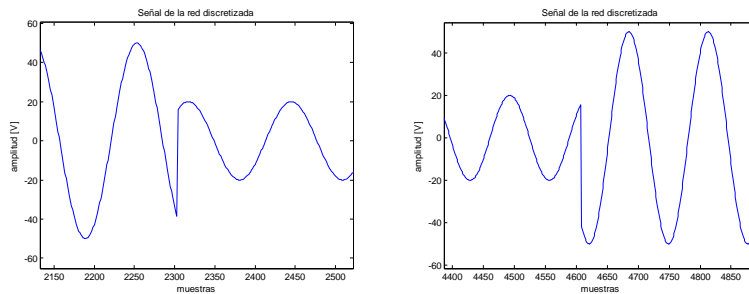
Figura 25. Cambio de fase de la señal producido por un hueco de tensión



a) Señal de red discretizada



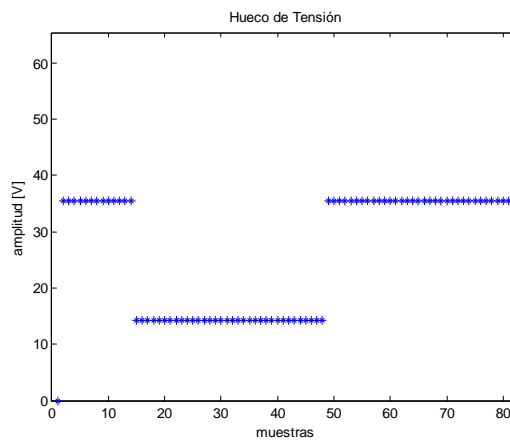
b) Cambios de fase producidos por el hueco de tensión



c) Cambios de fase producidos durante el hueco de tensión elevando la amplitud de la tensión.

Fuente: Autores

Figura 26. Resultado del algoritmo de muestra variable al ocurrir un cambio de fase en la señal



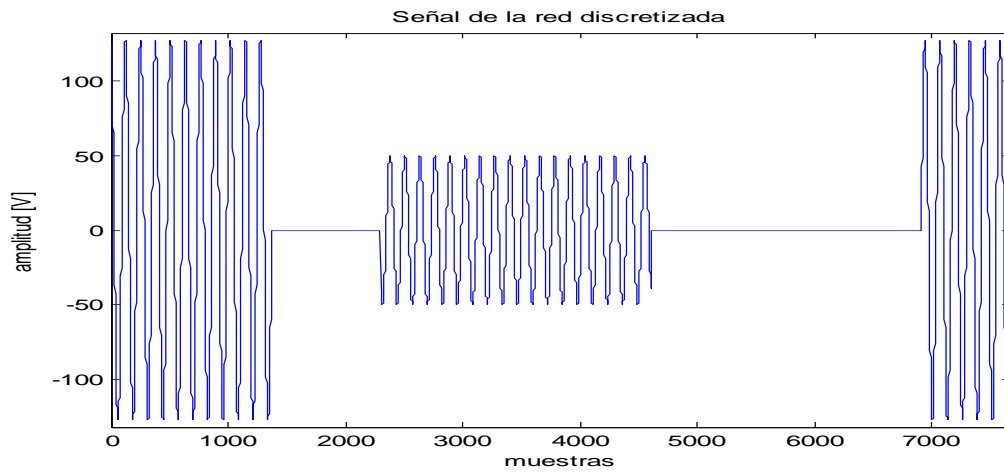
Fuente: Autores

En la Figura 26 se muestra el seguimiento que realiza el algoritmo de muestra variable a los cambios del valor eficaz de la señal de tensión. En ella se ve de que se está detectando de forma adecuada los cambios de fase producidos durante un evento, realizando la medida básica de manera óptima.

Finalmente, se simula una mezcla de perturbaciones que incluye un hueco y una interrupción de tensión con amplitud de 0 [Vp], para verificar el seguimiento del

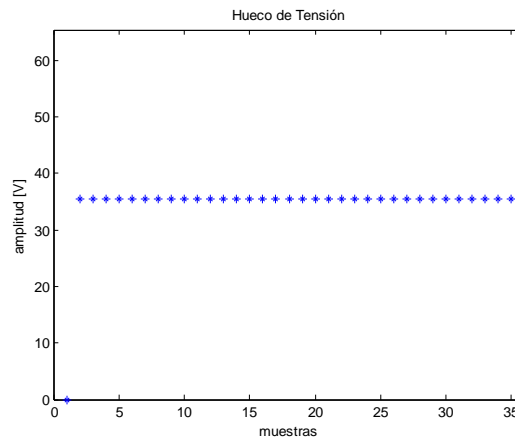
algoritmo de muestra variable a esta perturbación. La Figura 27 presenta la forma de la señal utilizada como prueba.

Figura 27. Interrupción de tensión



Fuente: Autores

Figura 28. Resultado del algoritmo de muestra variable.



Fuente: Autores

La Figura 28 solo presenta el seguimiento que hace el algoritmo del hueco de tensión donde la amplitud de la señal es diferente de 0 [Vp], ya que es innecesario

almacenar muestras de valor cero. El algoritmo realiza la medición de la interrupción de tensión como un caso diferente al hueco, por tanto el resultado será de dos interrupciones de tensión y un hueco de tensión que incluye las dos interrupciones mencionadas.

## **6. CONCLUSIONES Y FUTUROS DESARROLLOS**

En este trabajo se ha realizado un análisis sobre los antecedentes y estado actual en materia de reglamentación y normalización de la monitorización de la calidad de la energía eléctrica en los aspectos relacionados con las perturbaciones de la señal de tensión. Para ello se ha tenido en cuenta las resoluciones expedidas por la Comisión de Regulación de Energía y Gas (CREG) en Colombia, así como las normas que sobre esta temática ha establecido la Comisión Electrotécnica Internacional.

En este análisis se ha establecido el impacto de estos documentos en el desarrollo de los algoritmos de monitorización incluyendo algunas inconsistencias en las que se ha incurrido, a juicio de los autores, y que pueden generar conflicto en la estimación de perturbaciones de la señal de tensión.

El estudio realizado permitió perfeccionar el diseño de los algoritmos de monitorización de eventos tanto de estado estacionario (desbalance en sistemas trifásicos, amplitud de la tensión, sobretensión y subtensión) como de estado transitorio (huecos de tensión, sobretensiones temporales e interrupciones) considerando las principales características que describen el comportamiento del proceso como son: la medida básica, la detección del evento y la evaluación e incertidumbre de la medida.

Para la estimación del desbalance del sistema trifásico se ha efectuado un análisis de las implicaciones que puede tener la implementación de las componentes de secuencia negativa y cero a partir de los métodos contemplados en la norma IEC 61000-4-30, encontrándose que es mejor utilizar expresiones matemáticas de estimación de estas componentes que no involucren la realización de operaciones con números complejos o desfases de las ondas de tensión. También, se exploran las alternativas propuestas por la norma IEC 61000-4-30 para la atenuación de armónicos con el fin de obtener el valor eficaz de la componente fundamental de las señales de tensión, observando que mediante el algoritmo de la transformada discreta de Fourier (DFT) se realiza el proceso introduciendo menores errores en la medida.

El algoritmo de estimación del valor eficaz de la tensión, y los valores sobretensión y subtensión cuya medida básica es la tensión de doce ciclos, considera que siempre se tendrá un número constante de muestras en cada una de las ventanas de doce ciclos (2048), lo cual requiere la actualización de la frecuencia de muestreo de acuerdo con los cambios que experimente la frecuencia del sistema. En caso de no cumplirse esta condición, el algoritmo tendrá errores que dependiendo de la desincronización de esta frecuencia pueden estar por fuera de la incertidumbre establecida para cumplir con la exigencia de la clase de la medida. Si la desincronización es tal que exista una muestra de más o menos en el intervalo de observación de doce ciclos (2047 o 2049), el error aumenta con relación al obtenido con el caso sincronizado (2048); sin embargo, los errores máximos presentados se encuentran dentro del margen de incertidumbre establecido para la verificación de la conformidad de clase A con la norma IEC 61000-4-30.

La evaluación de la medida de los eventos de estado estacionario se realiza a partir de los intervalos de agregación de ciento ochenta períodos, diez minutos y dos horas. De acuerdo con el estándar IEC 61000-4-30, la agregación de ciento ochenta períodos se efectúa a partir de quince intervalos de doce ciclos. Sin

embargo, en el estándar no se especifica el inicio y final para el proceso de agregación. Por tal motivo, en este documento se ha determinado el encendido del equipo de medición de calidad de potencia como el inicio para la agregación de ciento ochenta períodos. En la agregación de diez minutos, el inicio y final para la identificación de los intervalos de doce ciclos, se encuentran determinados a través de un reloj de tiempo absoluto (Por ejemplo a las 6:00, 6:10, 6:20), pero no se encuentra una relación directa con la agregación de ciento ochenta períodos, debido a que esta última no se evalúa a partir de medidas absolutas de tiempo, sino con base en la medida de frecuencia. Por consiguiente, los intervalos de observación de doce ciclos empleados para las agregaciones de ciento ochenta períodos y diez minutos son diferentes por lo cual puede ser requerido que el algoritmo de medida básica de eventos de estado estacionario sea ejecutado simultáneamente dos veces. La agregación de dos horas se realiza a partir de doce intervalos de diez minutos.

Con respecto a los eventos de estado transitorio, el algoritmo de monitorización se puede realizar de dos formas: con muestras constantes comenzando en un cruce por cero de la señal fundamental o con muestras variables, es decir, tomando las muestras en un periodo comenzando en un cruce por cero y terminando en el cruce por cero donde se completa un periodo de la señal fundamental.

El primero es utilizado para cuando la sincronización del convertidor analógico-digital con la frecuencia de la señal de la red es estable, si esto no ocurre, el algoritmo no tomará la cantidad de muestras necesarias en la ventana de observación y el error crecerá a medida que aumente la falta de sincronización. El segundo caso es útil cuando no existe sincronización del convertidor analógico-digital con la señal de la red. Los errores presentados en esta situación son pequeños, debido a que el algoritmo toma la cantidad de muestras contenida en un periodo de la señal desde el primer cruce por cero del periodo hasta el último cruce por cero al final del periodo y de esta manera mantiene el error reducido.

Se desarrolló un artículo en el que se explica los pormenores del proyecto con el fin de presentarlo como ponencia en el V Simposio Internacional sobre Calidad de la Energía Eléctrica realizado en Santa Fé de Bogotá entre el 4 y 6 de agosto de 2009.

La monitorización de la calidad de la potencia se ha convertido en una prioridad para los entes reguladores que exigen a los operadores de red mejorar el servicio a sus usuarios. Este proyecto ha permitido conocer las ventajas y desventajas de los métodos exigidos para la medición de las perturbaciones de tensión y los requerimientos expuestos por los organismos internacionales para obtener medidas precisas que permiten fiabilidad en la detección. El trabajo en grupo fue prioritario para la finalización de los algoritmos y el desarrollo de este informe en el que se plasma los resultados mas relevantes en el análisis de los algoritmos de medición de los eventos de tensión enunciados en las resoluciones CREG 024 de 2005 y 016 de 2007 y en el Estándar IEC 6100-4-30. La interacción con diferentes personalidades de la calidad de potencia permitieron clarificar diferentes conceptos poco tangibles en los diferentes documentos de los entes gubernamentales e internacionales.

Finalmente, se han realizado las pruebas de conformidad con el estándar IEC 61000-4-30 para la monitorización de eventos de estado estacionario, encontrándose que los errores obtenidos en las condiciones de ensayo 1, 2 y 3 para desbalance, amplitud de la tensión de alimentación, y sobretensión y subtensión, están en el margen de incertidumbre tolerable para certificar la clase A. Para los eventos de estado transitorio, se ha planteado emplear señales sintéticas similares a las de estado estacionario, ya que en estándar IEC 61000-4-30 no se menciona como realizar la verificación de la conformidad para esta clase de eventos.

### **Para trabajos Futuros:**

Los algoritmos de monitorización de eventos de estado estacionario y transitorio se han realizado con base en el estándar IEC 61000-4-30 y las resoluciones 024 y 016 de la Comisión de Regulación de Energía y Gas (CREG). Sin embargo, como se ha podido observar a través del documento, el estándar IEC 61000-4-30 deja abierto de interpretación algunas consideraciones en el proceso de monitorización de perturbaciones de tensión las cuales se podrían tomar en cuenta para la realización de estudios posteriores:

- El inicio y final de la medida básica (tensión eficaz de doce ciclos) y la agregación de ciento ochenta períodos para eventos de estado estacionario. En el documento, el inicio de la medida se ha definido una vez se encienda el equipo de calidad de potencia eléctrica, mientras que el final de los doce ciclos se identifica con un número fijo de muestras, bajo la consideración de una correcta sincronización de frecuencia. Una alternativa podría ser en definir el inicio y final a partir de cruces por cero de la componente fundamental de la señal de entrada, de la misma manera que se ha realizado para eventos de estado transitorio.
- Medición de un hueco de tensión al ocurrir una interrupción total. En el documento, se optó por refrescar medio periodo de la señal arbitrariamente, hasta que la señal sea diferente de cero. Se pueden plantear otras soluciones a este suceso al cual el estándar no hace referencia.
- Pruebas de conformidad con el estándar IEC 61000-4-30 para eventos de estado transitorio. Se pueden plantear una propuesta de protocolo de pruebas para las perturbaciones de tensión de tipo transitorio.
- Pruebas con Flicker. Las pruebas que incluyeron la perturbación del fenómeno flicker son válidas únicamente con el medidor digital diseñado en

[Oviedo y Rueda, 2009] debido a que el estándar [IEC 61000-4-15, 2003] deja cierto grado de libertad en algunos aspectos de diseño como el número de cifras decimales para la frecuencia de modulación, el número de clases y el margen de ganancia por lo que los resultados de Pst pueden variar de forma considerable si se emplea otro medidor. Por consiguiente, como se menciona en [Oviedo y Rueda, 2009] se deben realizar estudios que permitan unificar criterios en la medición de la perturbación del fenómeno flicker, y de esta forma poder efectuar con una mayor certeza las pruebas de conformidad con el estándar IEC 61000-4-30.

## BIBLIOGRAFÍA

[Baggini, 2008] Baggini Angelo., “Handbook of POWER QUALITY”, Wiley, 642 pages, June 2008.

[Bollen, 2006] Bollen, Math H. J. y Gu Irene., “Signal Processing of Power Quality Disturbances”, Willey-IEEE press, 861 pages, August 2006.

[Cobos y Vargas, 2009] Cobos Jorge y Vargas Juan., “Simulación de un Generador de Perturbaciones de los Parámetros de la Calidad de la Potencia Eléctrica para Verificar la Conformidad con el Estándar CEI 61000-4-30”. Trabajo de Grado (Ingeniería Eléctrica), Universidad Industrial de Santander, 2009.

[CREG 070, 1998] Comisión de Regulación de Energía y Gas – CREG, “Resolución CREG 070 de 28 de Mayo de 1998”, Comisión de regulación de Energía y Gas – CREG, Santa Fé de Bogotá, Mayo de 1998.

[CREG 024, 2005] Comisión de Regulación de Energía y Gas – CREG, “Resolución CREG 024 de 26 de Abril de 2005”, Comisión de regulación de Energía y Gas – CREG, Santa Fé de Bogotá, Abril de 2005.

[CREG 016, 2007] Comisión de Regulación de Energía y Gas – CREG, “Resolución CREG 024 de 26 de Febrero de 2007”, Comisión de regulación de Energía y Gas – CREG, Santa Fé de Bogotá, Febrero de 2007.

[Duarte, 2004] Duarte, Cesar., “Técnicas de Procesamiento de Señales para la Monitorización de la Calidad de la Energía Eléctrica”, Trabajo de investigación para optar el título de Magíster en Potencia Eléctrica (Ingeniería Eléctrica), Universidad Industrial de Santander, 2004.

[IEC 61000-4-15, 2003] INTERNATIONAL ELECTROTECHNICAL COMMISSION. IEC 61000-4-15. “Electromagnetic compatibility (EMC)-Part 4-15: Testing and measurements techniques-Flickermeter. Functional and design specifications”. 16 p., Geneve, Switzerland, 2003.

[IEC 61000-4-30, 2003] INTERNATIONAL ELECTROTECHNICAL COMMISSION. IEC 61000-4-30. “Electromagnetic compatibility (EMC)-Part 4-30: Testing and measurements techniques- Power quality measurements methods”. 89 p., Geneve, Switzerland, 2003.

[Markiewicz y Klajn, 2004] Markiewicz Henryk, Klajn Antoni.VOLTAGE DISTURBANCES ESTÁNDAR EN 50160. Voltage characteristics in Public Distribution Systems, Wroclaw University of Technology, July 2006.

[NTC 5001, 2008] Norma Técnica Colombiana 5001: “Calidad de Potencia Eléctrica (CPE). Límites y Metodología de Evaluación en Punto de Conexión Común”, Instituto Colombiano de Normas Técnicas (ICONTEC), 51 páginas, Mayo de 2008.

[Oviedo y Rueda, 2009] Oviedo Silvia y Rueda Laura., “Diseño del Prototipo del Módulo para la Monitorización del Fenómeno Flicker para una Unidad Constructiva de Medición de Parámetros de Calidad de Potencia Eléctrica, con base en el

estándar CEI 61000-4-15". Trabajo de Grado (Ingeniería Eléctrica y Electrónica), Universidad Industrial de Santander, 2009.

[Villamizar, 2008] Villamizar Montes, Libardo, "Prototipo de Herramienta Software para la estimación de parámetros de la Máquina Síncrona bajo condiciones de Carga", Trabajo de investigación para optar el título de Magíster en Ingeniería Eléctrica (Ingeniería Eléctrica), Universidad Industrial de Santander, 2008.

## ANEXO A. ALGORITMO PARA LA MEDICIÓN DE LAS PERTURBACIONES ESTACIONARIAS DE TENSIÓN EN LENGUAJE C

### a) Algoritmo de agregación de ciento ochenta períodos

```
/*Declaración de librerías*/
#include<stdio.h>
#include<conio.h>
#include<math.h>

/*Declaración de funciones*/
float fft(float Va[],int Nsample);
float inversa(float Uabfund, float Ubcfund, float Ucafund);
float positiva(float Uabfund, float Ubcfund, float Ucafund);
float sobretension(float Ueficaz,float Udin);
float subtension(float Ueficaz, float Udin);
float pi=3.14159265358979;

main ()
{
    /*Variables de entrada*/
    float pi=3.14159265358979;
    float fs=10240;
    float Ts=1/fs;
    float fsis=60;
    float Udin=120;
    float Vmaxa=120*sqrt(2),Vmaxb=140*sqrt(2),Vmaxc=140*sqrt(2);
    int Nsample=2048, Nmax=15*2048;
    int i,cont12=0,cont180=0,k,n=0,w1=1;
    float t,Va,Vb,Vc,Vab,Vbc,Vca;
    float homopolar, V1[Nsample], V2[Nsample], V3[Nsample], homp[Nsample];
    float sumaa, sumab, sumac;
    float Ueficaza, Ueficazb, Ueficazc, Uabfund, Ubcfund, Ucafund, u2,
    float tensionhomopolar;
    float positivo, u0, Ualtaa, Ubajaa, Ualtab, Ubajab, Ualtac, Ubajac;

    /*Variables para la agregación de 180 periodos*/
    float u2180[5],u0180[5],Ueficaza180[5],Ueficazb180[5],Ueficazc180[5];
    float Ualtaa180[5], Ualtab180[5],Ualtac180[5],Ubajaa180[5],Ubajab180[5],Ubajac180[5];

    /*Inicialización de las variables para la agregación de 180 periodos*/
    float sumau2=0, sumau0=0, sumaueficaza=0, sumaueficazb=0, sumaueficazc=0, sumaualtaa=0, sumaubajaa=0;
    float sumaualtab=0, sumaubajab=0, sumaualtac=0, sumaubajac=0;

    /*Inicio del algoritmo*/
    while(n<Nmax)
    {
        k=0;
        sumaa=0;
        sumab=0;
        sumac=0;
        for (i=n;i<n+Nsample;i++)
        {
            /*Señal trifásica de entrada: tensiones de fase*/
            t=i*Ts;
            Va=Vmaxa*sin(2*pi*fsis*t);
            Vb=Vmaxb*sin(2*pi*fsis*t-(2/3.0)*pi);
            Vc=Vmaxc*sin(2*pi*fsis*t+(2/3.0)*pi);

            /* Tensiones de línea*/
            Vab=Va-Vb;
            Vbc=Vb-Vc;
            Vca=Vc-Va;
            /*Suma de las muestras al cuadrado para la señal*/
            sumaa=sumaa+pow(Va,2);
```

```

sumab=sumab+pow(Vb,2);
sumac=sumac+pow(Vc,2);

/*Componente de secuencia cero*/
homopolar=(1/sqrt(3))*(Va+Vb+Vc);

/*Almacenamiento de la señal de entrada*/
V1[k]=Vab;
V2[k]=Vbc;
V3[k]=Vca;
homp[k]=homopolar;
k=k+1;
}
Ueficaza=sqrt(sumaa/Nsample);
Ueficazb=sqrt(sumab/Nsample);
Ueficazc=sqrt(sumac/Nsample);

/*Tensión eficaz de la componente fundamental de las tensiones de línea*/
Uabfund=fft(V1,Nsample);
Ubcfund=fft(V2,Nsample);
Ucafund=fft(V3,Nsample);

/*Tensión eficaz de la componente fundamental de la tensión homopolar*/
tensionhomopolar=fft(homp,Nsample);

/*Componente Inversa*/
u2=invrsa(Uabfund,Ubcfund,Ucafund);

/*Componente de Secuencia positiva*/
positivo=positiva(Uabfund,Ubcfund,Ucafund);

/*Componente homopolar*/
u0=(tensionhomopolar/positivo)*100;

/*Sobretension y subtension*/
/*Para la fase A*/
Ualtaa=sobretension(Ueficaza,Udin);
Ubajaa=subtension(Ueficaza,Udin);

/*Para la fase B*/
Ualtab=sobretension(Ueficazb,Udin);
Ubajab=subtension(Ueficazb,Udin);

/*Para la fase C*/
Ualtac=sobretension(Ueficazc,Udin);
Ubajac=subtension(Ueficazc,Udin);

/*Suma parciales para la Agregación de 180 periodos*/
sumau2=sumau2+pow(u2,2);
sumau0=sumau0+pow(u0,2);
sumaueficaza=sumaueficaza+pow(Ueficaza,2);
sumaueficazb=sumaueficazb+pow(Ueficazb,2);
sumaueficazc=sumaueficazc+pow(Ueficazc,2);
sumaualtaa=sumaualtaa+pow(Ualtaa,2);
sumaubajaa=sumaubajaa+pow(Ubajaa,2);
sumaualtab=sumaualtab+pow(Ualtab,2);
sumaubajab=sumaubajab+pow(Ubajab,2);
sumaualtac=sumaualtac+pow(Ualtac,2);
sumaubajac=sumaubajac+pow(Ubajac,2);

/*Evaluación de la Agregación de 180 ciclos*/
if (cont12==(w1*15)-1)
{
/*Desbalance*/
u2180[cont180]=sqrt(sumau2/15.0);
u0180[cont180]=sqrt(sumau0/15.0);
}

```

```

/*Amplitud de la tension*/
Ueficaza180[cont180]=sqrt(sumaueficaza/15.0);
Ueficazb180[cont180]=sqrt(sumaueficazb/15.0);
Ueficazc180[cont180]=sqrt(sumaueficazc/15.0);

/*Sobretension y subtension*/
Ualtaa180[cont180]=sqrt(sumaualtaa/15.0);
Ubajaa180[cont180]=sqrt(sumaubajaa/15.0);
Ualtab180[cont180]=sqrt(sumaualtab/15.0);
Ubajab180[cont180]=sqrt(sumaubajab/15.0);
Ualtac180[cont180]=sqrt(sumaualtac/15.0);
Ubajac180[cont180]=sqrt(sumaubajac/15.0);

/*Actualización de Variables*/
sumau2=0; sumau0=0; sumaueficaza=0; sumaueficazb=0; sumaueficazc=0; sumaualtaa=0;
sumaubajaa=0; sumaualtab=0; sumaubajab=0; sumaualtac=0; sumaubajac=0;
cont180=cont180+1;
w1=w1+1;
}

/*Actualización de variables*/
n=n+2048;
cont12=cont12+1;
}

/*Resultados para la agregación de 180 ciclos*/
for( i=0;i<cont180;i++)
{
printf("\nLa tension eficaz para la fase A es %f\n",Ueficaza180[i]);
printf("\nLa tension eficaz para la fase B es %f\n",Ueficazb180[i]);
printf("\nLa tension eficaz para la fase C es %f\n",Ueficazc180[i]);
printf("\nDesbalance: componente inversa es %f\n",u2180[i]);
printf("\nDesbalance: componente homopolar es %f\n",u0180[i]);
printf("\nLa sobretension para la fase A es %f\n",Ualtaa180[i]);
printf("\nLa subtension para la fase A es %f\n",Ubajaa180[i]);
printf("\nLa Sobretension para la fase B es %f\n",Ualtab180[i]);
printf("\nLa subtension para la fase B es %f\n",Ubajab180[i]);
printf("\nLa sobretension para la fase C es %f\n",Ualtac180[i]);
printf("\nLa subtension para la fase C es %f\n",Ubajac180[i]);
}
getch();
}

/*Función para la implementación del algoritmo de la DFT*/

float fft(float Va[],int Nsample)
{
float V[Nsample], Vreal[Nsample],Vimag[Nsample];
float sumreal, sumimag,fundamental;
int i,j;

/*Parte real de la DFT*/
for (i=0;i<Nsample;i++)
{
sumreal=0;
for (j=0; j<Nsample;j++)
{
sumreal=sumreal+(Va[j]*cos((2*pi*i*j)/Nsample));
}
Vreal[i]=sumreal;
}

/*Parte imaginaria de la DFT*/
for (i=0;i<Nsample;i++)
{
sumimag=0;
for (j=0;j<Nsample;j++)

```

```

        {
            sumimag=sumimag-(Va[j]*sin((2*pi*i*j)/Nsample));
        }
        Vimag[i]=sumimag;
    }

    /*Magnitud de los coeficientes de la DFT*/
    for (i=0;i<Nsample;i++)
    {
        V[i]=(sqrt(2)/Nsample)*sqrt(pow(Vreal[i],2)+pow(Vimag[i],2));
    }

    /*Componente fundamental de la señal de entrada*/
    fundamental=V[12];

    return fundamental;
}

/*Función para el calculo de la componente inversa*/

float inversa(float Uabfund, float Ubcfund, float Ucafund)
{
    float aux,beta,u2;
    aux= pow(Uabfund,2)+pow(Ubcfund,2)+pow(Ucafund,2);
    beta=(pow(Uabfund,4)+pow(Ubcfund,4)+pow(Ucafund,4)) /pow(aux,2);
    u2=fabs((sqrt((1-sqrt(3-(6*beta))))/(1+sqrt(3-(6*beta))))*100);
    return u2;
}

/*Función para calcular la componente de secuencia positiva*/
float positiva(float Uabfund, float Ubcfund, float Ucafund)
{
    float w,x,y,z,positivo;
    w=((pow(Ubcfund,2))-(pow(Ucafund,2)))/Uabfund);
    x=sqrt(4*pow(Ubcfund,2)-pow(Uabfund+w,2));
    y=pow(((sqrt(3)*Uabfund+x)),2);
    z=y+(pow(w,2));
    positivo=sqrt((1/12.0)*z);
    return positivo;
}

/*función para calcular el valor de sobretensión*/
float sobretension(float Ueficaz,float Udin)
{
    float Ualta;
    if (Ueficaz>=Udin)
        Ualta=((Ueficaz-Udin)/Udin)*100;
    else Ualta=0;
    return Ualta;
}

/*Función para calcular el valor de subtensión*/
float subtension(float Ueficaz, float Udin)
{
    float Ubaja;
    if (Ueficaz<Udin)
        Ubaja=((Udin-Ueficaz)/Udin)*100;
    else Ubaja=0;
    return Ubaja;
}

```

## b) Algoritmo de agregación de diez minutos y dos horas

```
/*Declaración de librerías*/
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<time.h>

/*Declaracion de funciones*/
int intervaloaux(int minutos);
int intervalo(int minutos);
int reloj(int minutos);
float fft(float Va[],int Nsample);
float inversa(float Uabfund, float Ubcfund, float Ucafund);
float positiva(float Uabfund, float Ubcfund, float Ucafund);
float sobretension(float Ueficaz,float Udin);
float subtension(float Ueficaz, float Udin);
float pi=3.14159265358979;

/*Variables de entrada*/

main()
{
    float pi=3.14159265358979;
    float fs=10240;
    float Ts=1/fs;
    float fsis=60;
    float Udin=120;
    float Vmaxa=120*sqrt(2),Vmaxb=140*sqrt(2),Vmaxc=140*sqrt(2);
    int Nsample=2048, Nmax=2048, muestras=1,actualizacion=0;
    int i,,cont10min=0,cont2hr=0,k,n=0,w2hr=1,minutos,segundos,tiempoabs,tiempoabsaux;
    float ti,Va,Vb,Vc,Vab,Vbc,Vca;
    float homopolar, V1[Nsample], V2[Nsample], V3[Nsample], homp[Nsample];
    float sumaa, sumab, sumac;
    float Ueficaza, Ueficazb, Ueficazc,Uabfund,Ubcfund,Ucafund,u2,tensionhomopolar;
    float positivo,u0,Ualtaa,Ubajaa,Ualtab,Uabajab,Ualtac,Ualtac,Uabajac;
    int muestrainicio,auxminutos,auxcont12=0;

    /*Variables para la agregacion de 10 minutos*/
    float u210min,u010min,Ueficaza10min,Ueficazb10min,Ueficazc10min;
    float Ualtaa10min,Ualtab10min,Ualtac10min,Uabajaa10min,Uabajab10min
    float Ubajac10min;

    /*Variables para la agregacion de 2 horas*/
    float u22hr,u02hr,Ueficaza2hr,Ueficazb2hr,Ueficazc2hr;
    float Ualtaa2hr, Ualtab2hr,Ualtac2hr,Uabajaa2hr,Uabajab2hr,Uabajac2hr;

    /*Inicializacion de las variables para la agregacion de 10 minutos*/
    float sumau2=0, sumau0=0, sumaueficaza=0, sumaueficazb=0, sumaueficazc=0, sumaualtaa=0;
    float sumaubajaa=0, sumaualtab=0, sumaubajab=0, sumaualtac=0, sumaubajac=0;

    /*Inicializacion de las variables para la agregacion de 2 horas*/
    float sumau210min=0, sumau010min=0, sumaueficaza10min=0, sumaueficazb10min=0, sumaueficazc10min=0;
    float sumaualtaa10min=0, sumaubajaa10min=0, sumaualtab10min=0, sumaubajab10min=0, sumaualtac10min=0;
    float sumaubajac10min=0;

    /*Inicio del reloj de tiempo*/
    struct tm* tptr;
    time_t t = time(NULL);
    tptr = localtime(&t);
    minutos=tptr->tm_min;
    segundos=tptr->tm_sec;

    /*Proceso que realiza el inicio de la agregacion de 10 minutos*/
    minutos=reloj(minutos);
    /*Actualizacion del siguiente intervalo de reloj de tiempo absoluto*/
```

```

tiempoabs=intervalo(minutos);
tiempoabsaux=intervaloaux(minutos);

/*Ejecución del algoritmo de medida basica */
while(minutos<=tiempoabsaux)

{
    /*Actualizacion del reloj*/
    time_t t = time(NULL);
    tptr = localtime(&t);
    minutos=tptr->tm_min;
    segundos=tptr->tm_sec;
    k=0;
    sumaa=0;
    sumab=0;
    sumac=0;

    for (i=n;i<n+Nsample;i++)
    {
        /*Actualizacion del reloj*/
        time_t t = time(NULL);
        tptr = localtime(&t);
        minutos=tptr->tm_min;
        segundos=tptr->tm_sec;

        if (minutos==tiempoabs)
        {
            muestrainicio=muestras;
            auxminutos=minutos;
            actualizacion=1;
        }
        /*Señal de entrada: tensiones de fase*/
        ti=i*Ts;
        Va=Vmaxa*sin(2*pi*fsis*ti);
        Vb=Vmaxb*sin(2*pi*fsis*ti-(2/3.0)*pi);
        Vc=Vmaxc*sin(2*pi*fsis*ti+(2/3.0)*pi);

        /*Señal de entrada: tensiones de linea*/
        Vab=Va-Vb;
        Vbc=Vb-Vc;
        Vca=Vc-Va;

        /*Suma de las muestras al cuadrado para la señal*/
        sumaa=sumaa+pow(Va,2);
        sumab=sumab+pow(Vb,2);
        sumac=sumac+pow(Vc,2);

        /*Componente de secuencia cero*/
        homopolar=(1/sqrt(3))*(Va+Vb+Vc);
        /*Almacenamiento de la señal de entrada*/
        V1[k]=Vab;
        V2[k]=Vbc;
        V3[k]=Vca;
        homp[k]=homopolar;
        k=k+1;

        /*conteo de muestras y actualizacion del reloj*/
        muestras=muestras+1;
    }
    Ueficaza=sqrt(sumaa/Nsample);
    Ueficazb=sqrt(sumab/Nsample);
    Ueficazc=sqrt(sumac/Nsample);

    /*Tension eficaz de la componente fundamental de las tensiones de linea*/
    Uabfund=fft(V1,Nsample);
    Ubcfund=fft(V2,Nsample);
    Ucafund=fft(V3,Nsample);

```

```

/*Tension eficaz de la componente fundamental de la tension homopolar*/
tensionhomopolar=fft(homp,Nsample);

/*Componente Inversa*/
u2=invrsa(Uabfund,Ubcfund,Ucafund);

/*Componente de Secuencia positiva*/
positivo=positiva(Uabfund,Ubcfund,Ucafund);

/*Componente homopolar*/
u0=(tensionhomopolar/positivo)*100;

/*Sobretension y subtension*/

/*Para la fase A*/
Ualtaa=sobretension(Ueficaza,Udin);
Ubajaa=subtension(Ueficaza,Udin);

/*Para la fase B*/
Ualtab=sobretension(Ueficazb,Udin);
Ubajab=subtension(Ueficazb,Udin);

/*Para la fase C*/
Ualtac=sobretension(Ueficazc,Udin);
Ubajac=subtension(Ueficazc,Udin);

/*Suma parciales para la Agregacion de 10 minutos*/
sumau2=sumau2+pow(u2,2);
sumau0=sumau0+pow(u0,2);
sumaueficaza=sumaueficaza+pow(Ueficaza,2);
sumaueficazb=sumaueficazb+pow(Ueficazb,2);
sumaueficazc=sumaueficazc+pow(Ueficazc,2);
sumaualtaa=sumaualtaa+pow(Ualtaa,2);
sumaubajaa=sumaubajaa+pow(Ubajaa,2);
sumaualtab=sumaualtab+pow(Ualtab,2);
sumaubajab=sumaubajab+pow(Ubajab,2);
sumaualtac=sumaualtac+pow(Ualtac,2);
sumaubajac=sumaubajac+pow(Ubajac,2);
auxcont12=auxcont12+1;

/*Evaluacion de la Agregacion de 10 minutos*/
if (actualizacion!=1)
{
    n=n+2048;
}

else
{
    /*Desbalance*/
    u210min=sqrt(sumau2/auxcont12);
    u010min=sqrt(sumau0/auxcont12);
    /*Amplitud de la tension*/
    Ueficaza10min=sqrt(sumaueficaza/auxcont12);
    Ueficazb10min=sqrt(sumaueficazb/auxcont12);
    Ueficazc10min=sqrt(sumaueficazc/auxcont12);

    /*Sobretension y subtension*/
    Ualtaa10min=sqrt(sumaualtaa/auxcont12);
    Ubajaa10min=sqrt(sumaubajaa/auxcont12);
    Ualtab10min=sqrt(sumaualtab/auxcont12);
    Ubajab10min=sqrt(sumaubajab/auxcont12);
    Ualtac10min=sqrt(sumaualtac/auxcont12);
    Ubajac10min=sqrt(sumaubajac/auxcont12);

    /*Actualizacion para la nueva ventana de agregacion de 10 minutos*/
    n=muestrainicio;
}

```

```

/*Actualizacion del intervalo de reloj de tiempo absoluto*/
tiempoabs=intervalo(auxminutos);
tiempoabsaux=intervaloaux(auxminutos);

/*Inicializacion de sumadores*/
sumau2=0; sumau0=0; sumaueficaza=0; sumaueficazb=0; sumaueficazc=0; sumaualtaa=0; sumaubajaa=0;
sumaualtab=0; sumaubajab=0; sumaualtac=0; sumaubajac=0;

/*Actualizacion de contadores*/
cont10min=cont10min+1;
auxcont12=0;
muestras=0;
cont12=0;
actualizacion=0;

/*Resultados para la agregacion de 10 minutos*/
printf("\nResultados agregacion 10 minutos\n");
printf("\nLa tension eficaz para la fase A es %f\n",Ueficaza10min);
printf("\nLa tension eficaz para la fase B es %f\n",Ueficazb10min);
printf("\nLa tension eficaz para la fase C es %f\n",Ueficazc10min);
printf("\nDesbalance: componente inversa es %f\n",u210min);
printf("\nDesbalance: componente homopolar es %f\n",u010min);
printf("\nLa sobretension para la fase A es %f\n",Ualtaa10min);
printf("\nLa subtension para la fase A es %f\n",Ubajaa10min);
printf("\nLa Sobretension para la fase B es %f\n",Ualtab10min);
printf("\nLa subtension para la fase B es %f\n",Ubajab10min);
printf("\nLa Sobretension para la fase C es %f\n",Ualtac10min);
printf("\nLa subtension para la fase C es %f\n",Ubajac10min);

/*Suma de los valores de 10 minutos para la agregacion de 2 horas*/
sumau210min=sumau210min+pow(u210min,2);
sumau010min=sumau010min+pow(u010min,2);
sumaueficaza10min=sumaueficaza10min+pow(Ueficaza10min,2);
sumaueficazb10min=sumaueficazb10min+pow(Ueficazb10min,2);
sumaueficazc10min=sumaueficazc10min+pow(Ueficazc10min,2);
sumaualtaa10min=sumaualtaa10min+pow(Ualtaa10min,2);
sumaubajaa10min=sumaubajaa10min+pow(Ubajaa10min,2);
sumaualtab10min=sumaualtab10min+pow(Ualtab10min,2);
sumaubajab10min=sumaubajab10min+pow(Ubajab10min,2);
sumaualtac10min=sumaualtac10min+pow(Ualtac10min,2);
sumaubajac10min=sumaubajac10min+pow(Ubajac10min,2);

if (cont10min==(w2hr*12)-1)
{
    /*Desbalance*/
    u22hr=sqrt(sumau210min/12.0);
    u02hr=sqrt(sumau010min/12.0)

    /*Amplitud de la tension*/
    Ueficaza2hr=sqrt(sumaueficaza10min/12.0);
    Ueficazb2hr=sqrt(sumaueficazb10min/12.0);
    Ueficazc2hr=sqrt(sumaueficazc10min/12.0);

    /*Sobretension y subtension*/
    Ualtaa2hr=sqrt(sumaualtaa10min/12.0);
    Ubajaa2hr=sqrt(sumaubajaa10min/12.0);
    Ualtab2hr=sqrt(sumaualtab10min/12.0);
    Ubajab2hr=sqrt(sumaubajab10min/12.0);
    Ualtac2hr=sqrt(sumaualtac10min/12.0);
    Ubajac2hr=sqrt(sumaubajac10min/12.0);

    /*Inicializacion de Variables*/
    sumau210min=0; sumau010min=0; sumaueficaza10min=0; sumaueficazb10min=0;
    sumaueficazc10min=0; sumaualtaa10min=0; sumaubajaa10min=0; sumaualtab10min=0;
    sumaubajab10min=0; sumaualtac10min=0; sumaubajac10min=0;
    /*Actualizacion del conteo*/

```

```

cont2hr=cont2hr+1;
w2hr=w2hr+1;

/*Resultados para la agregacion de 2 horas*/
printf("\nResultados agregacion 10 minutos\n");
printf("\nLa tension eficaz para la fase A es %f\n",Ueficaza2hr);
printf("\nLa tension eficaz para la fase B es %f\n",Ueficazb2hr);
printf("\nLa tension eficaz para la fase C es %f\n",Ueficazc2hr);
printf("\nDesbalance: componente inversa es %f\n",u22hr);
printf("\nDesbalance: componente homopolar es %f\n",u02hr);
printf("\nLa sobretension para la fase A es %f\n",Ualtaa2hr);
printf("\nLa subtension para la fase A es %f\n",Ubajaa2hr);
printf("\nLa sobretension para la fase B es %f\n",Ualtab2hr);
printf("\nLa subtension para la fase B es %f\n",Ubajab2hr);
printf("\nLa sobretension para la fase C %f\n",Ualtac2hr);
printf("\nLa subtension para la fase C es %f\n",Ubajac2hr[i]);
    }
}
}
getch();
}

/*Funcion para el inicio de la agregacion de 10 minutos*/
int reloj(int minutos)
{
    struct tm* tptr;
    if (minutos<10)
        while (minutos!=10)
        {
            time_t t = time(NULL);
            tptr = localtime(&t);
            minutos=tptr->tm_min;
        }
    else if (minutos<20)
        while (minutos!=20)
        {
            time_t t = time(NULL);
            tptr = localtime(&t);
            minutos=tptr->tm_min;
        }
    else if (minutos<30)
        while (minutos!=30)
        {
            time_t t = time(NULL);
            tptr = localtime(&t);
            minutos=tptr->tm_min;
        }
    else if (minutos<40)
        while (minutos!=40)
        {
            time_t t = time(NULL);
            tptr = localtime(&t);
            minutos=tptr->tm_min;
        }
    else if (minutos<50)
        while (minutos!=50)
        {
            time_t t = time(NULL);
            tptr = localtime(&t);
            minutos=tptr->tm_min;
        }
    else if (minutos<=59)
        while (minutos!=0)
        {
            time_t t = time(NULL);
            tptr = localtime(&t);
            minutos=tptr->tm_min;
        }
}

```

```

        }
        return minutos;
    }
}

/*Funcion para actualizar el siguiente intervalo de tiempo absoluto*/
int intervalo(int minutos)
{
    struct tm* tptr;
    int tiempoabs;
    switch (minutos)
    {
        case 10:
            tiempoabs=20;
            break;
        case 20:
            tiempoabs=30;
            break;
        case 30:
            tiempoabs=40;
            break;
        case 40:
            tiempoabs=50;
            break;
        case 50:
            tiempoabs=0;
            break;
        case 0:
            tiempoabs=10;
            break;
    }
    return tiempoabs;
}

/*Funcion para actualizar el siguiente intervalo de tiempo absoluto*/
int intervaloaux(int minutos)
{
    struct tm* tptr;
    int tiempoabsaux;

    switch (minutos)
    {
        case 10:
            tiempoabsaux=20;
            break;
        case 20:
            tiempoabsaux=30;
            break;
        case 30:
            tiempoabsaux=40;
            break;
        case 40:
            tiempoabsaux=50;
            break;
        case 50:
            tiempoabsaux=60;
            break;
        case 0:
            tiempoabsaux=10;
            break;
    }
    return tiempoabsaux;
}
}

```

## ANEXO B. ALGORITMO PARA LA MONITORIZACIÓN DE LAS PERTURBACIONES ESTACIONARIAS EN MATLAB

### a) Algoritmo de agregación de ciento ochenta períodos

```
clear all;
clc

%Variables de entrada
fs=10240;
Ts=1/fs;
fsis=60;
Udin=120;
Vmaxa=60*sqrt(2);
Vmaxb=140*sqrt(2);
Vmaxc=140*sqrt(2);
Nsample=2048;
Nmax=15*2048;
n=1;
cont12=1;
cont180=1;
w1=1;

%Inicialización de las variables para la agregación de 180 periodos
sumau2=0; sumau0=0; sumaueficaza=0; sumaueficazb=0; sumaueficazc=0; sumaualtaa=0; sumaubajaa=0;
sumaualtab=0; sumaubajab=0; sumaualtac=0; sumaubajac=0;

%Inicio del algoritmo
while n<=Nmax

    %Inicializacion de Variables para medida básica
    sumaa=0;
    sumab=0;
    sumac=0;

    k=1;

    for i=n:n+Nsample-1

        %Señal trifásica de entrada: tensiones de fase
        t=(i-1)*Ts;
        Va=Vmaxa*sin(2*pi*fsis*t);
        Vb=Vmaxb*sin(2*pi*fsis*t-(2/3)*pi);
        Vc=Vmaxc*sin(2*pi*fsis*t+(2/3)*pi);

        %Señal de entrada: tensiones de línea
        Vab=Va-Vb;
        Vbc=Vb-Vc;
        Vca=Vc-Va;

        %Componente de secuencia cero
        homopolar=(1/sqrt(3))*(Va+Vb+Vc);

        %Suma de las muestras al cuadrado para la señal
        sumaa=sumaa+Va^2;
        sumab=sumab+Vb^2;
        sumac=sumac+Vc^2;

        %Guardar de los datos las señal de entrada
        V1(k)=Vab;
```

```

V2(k)=Vbc;
V3(k)=Vca;
homp(k)=homopolar;
k=k+1;
end

%Valor eficaz de doce ciclos: Amplitud de la tensión de Alimentación
Ueficaza=sqrt(sumaa/Nsample);
Ueficazb=sqrt(sumab/Nsample);
Ueficazc=sqrt(sumac/Nsample);

%Transformada de Fourier para las tensiones de línea
ffta=fft(V1);
fftb=fft(V2);
fftc=fft(V3);

%Transformada de Fourier para la tensión homopolar
ffthomp=fft(homp);

magnituda=abs(ffta(13));
magnitudb=abs(fftb(13));
magnitudc=abs(fftc(13));
magnitudhomp=abs(ffthomp(13));

%Tension eficaz de la componente fundamental de las tensiones de línea
Uabfund=(sqrt(2)/Nsample)*magnituda;
Ubcfund=(sqrt(2)/Nsample)*magnitudb;
Ucafund=(sqrt(2)/Nsample)*magnitudc;

%Tension eficaz de la componente fundamental de la tension homopolar
tensionhomopolar=(sqrt(2)/Nsample)*magnitudhomp;

%Componente Inversa
u2=invrsa(Uabfund,Ubcfund,Ucafund);

%Componente de Secuencia positiva
positiva=secpositiva(Uabfund,Ubcfund,Ucafund,cont12);

%Componente homopolar
u0=(tensionhomopolar/positiva)*100;

%Sobretension y Subtension

%-----Para la fase A
[Ualtaa,Ubajaa]=valoraltobajo(Ueficaza,Udin);

%-----Para la fase B
[Ualtab,Ubajab]=valoraltobajo(Ueficazb,Udin);

%-----Para la fase C
[Ualtac,Ubajac]=valoraltobajo(Ueficazc,Udin);

%Suma acumulada de los valores de doce ciclos para la Agregación de 180 periodos
sumau2=sumau2+u2^2;
sumau0=sumau0+u0^2;
sumaueficaza=sumaueficaza+Ueficaza^2;
sumaueficazb=sumaueficazb+Ueficazb^2;
sumaueficazc=sumaueficazc+Ueficazc^2;
sumaualtaa=sumaualtaa+Ualtaa^2;
sumaubajaa=sumaubajaa+Ubajaa^2;
sumaualtab=sumaualtab+Ualtab^2;
sumaubajab=sumaubajab+Ubajab^2;
sumaualtac=sumaualtac+Ualtac^2;
sumaubajac=sumaubajac+Ubajac^2;

```

```

%Evaluación de la Agregación de 180 ciclos

if (cont12==w1*15)
    %Desbalance
    u2180(cont180)=sqrt(sumau2/15);
    u0180(cont180)=sqrt(sumau0/15);

    %Amplitud de la tensión
    Ueficaza180(cont180)=sqrt(sumaueficaza/15);
    Ueficazb180(cont180)=sqrt(sumaueficazb/15);
    Ueficazc180(cont180)=sqrt(sumaueficazc/15);

    %Sobretension y subtension
    Ualtaa180(cont180)=sqrt(sumaualtaa/15);
    Ubajaa180(cont180)=sqrt(sumaubajaa/15);
    Ualtab180(cont180)=sqrt(sumaualtab/15);
    Ubajab180(cont180)=sqrt(sumaubajab/15);
    Ualtac180(cont180)=sqrt(sumaualtac/15);
    Ubajac180(cont180)=sqrt(sumaubajac/15);

    %Inicialización de Variables
    sumau2=0; sumau0=0; sumaueficaza=0; sumaueficazb=0; sumaueficazc=0; sumaualtaa=0; sumaubajaa=0;
    sumaualtab=0; sumaubajab=0; sumaualtac=0; sumaubajac=0;

    cont180=cont180+1;
    w1=w1+1;
end

%Actualización de variables para el siguiente grupo de muestras
cont12=cont12+1;
n=n+2048;

end

%Resultados Obtenidos

fprintf('\nRESULTADOS OBTENIDOS\n');

%Resultados Agregación 180 periodos
fprintf('\nMEDIDA DE AGREGACION DE 180 PERIODOS\n');

fprintf('Componente Inversa\n');      disp(u2180)
fprintf('Componente Homopolar\n');  disp(u0180)
fprintf('Valor eficaz para la fase A\n'); disp(Ueficaza180)
fprintf('Valor eficaz para la fase B\n'); disp(Ueficazb180)
fprintf('Valor eficaz para la fase C\n'); disp(Ueficazc180)
fprintf('Sobretension para la fase A\n'); disp(Ualtaa180)
fprintf('Subtension para la fase A\n'); disp(Ubajaa180)
fprintf('Sobretension para la fase B\n'); disp(Ualtab180)
fprintf('Subtension para la fase B\n'); disp(Ubajab180)
fprintf('Sobretension para la fase C\n'); disp(Ualtac180)
fprintf('Subtension para la fase C\n'); disp(Ubajac180)

%Funciones auxiliares

%Función para la estimación de la Componente Inversa

function u2=inversa(Uabfund,Ubcfund,Ucafund)

beta=((Uabfund.^4)+(Ubcfund.^4)+(Ucafund.^4))./(((Uabfund.^2)+(Ubcfund.^2)+(Ucafund.^2)).^2);
u2=abs((sqrt((1-sqrt(3-(6*beta)))./(1+sqrt(3-(6*beta))))*100));

%Función para el cálculo de la componente de secuencia positiva

```

```

function positiva=sepositiva(Uabfund,Ubcfund,Ucafund)
w=((Ubcfund^2)-(Ucafund^2))/Uabfund);
x=sqrt(4*(Ubcfund^2)-((Uabfund+w)^2));
y=((sqrt(3)*Uabfund+x)^2);
z=y+(w^2);
positiva=sqrt((1/12)*z);

%Función para la evaluación de los parámetros de sobretensión y subtensión

function [Ualtaa,Ubajaa]=valoraltobajo(Ueficaza)
if Ueficaza<=Udin
    Ualtaa=0;
    Ubajaa=((Udin-Ueficaza)/Udin)*100;
else
    Ualtaa=((Ueficaza-Udin)/Udin)*100;
    Ubajaa=0;
end

```

## b) Algoritmo de agregación de diez minutos y dos horas

```

clear all;
clc

%Variables de entrada

fs=10240;
Ts=1/fs;
fsis=60;
Udin=120;
Vmaxa=120*sqrt(2);
Vmaxb=140*sqrt(2);
Vmaxc=140*sqrt(2);
Nsample=2048;
auxcont12=0;
cont12=1;
n=0;
cont10min=1;
cont2hr=1;
w2hr=1;

%Inicialización de las variables para la agregación de 10 minutos
sumau2=0; sumau0=0; sumaueficaza=0; sumaueficazb=0; sumaueficazc=0; sumaualtaa=0; sumaubajaa=0;
sumaualtab=0; sumaubajab=0; sumaualtac=0; sumaubajac=0;

%Inicialización de las variables para la agregación de 2 horas
sumau210min=0; sumau010min=0; sumaueficaza10min=0; sumaueficazb10min=0; sumaueficazc10min=0;
sumaualtaa10min=0; sumaubajaa10min=0; sumaualtab10min=0; sumaubajab10min=0; sumaualtac10min=0;
sumaubajac10min=0;

%Medida del reloj de tiempo

%reloj(5) se asignan los minutos y reloj(6) para los segundos
reloj=clock;
%Proceso que calcula el inicio para la agregación de 10 minutos

if reloj(5)<10
    while reloj(5)~=10
        reloj=clock;
    end
else if reloj(5)<20
    while reloj(5)~=20
        reloj=clock;
    end
else if reloj(5)<30
    while reloj(5)~=30

```

```

        reloj=clock;
    end
else if reloj(5)<40
    while reloj(5)~=40
        reloj=clock;
    end
else if reloj(5)<50
    while reloj(5)~=50
        reloj=clock;
    end
else if reloj(5)<=59
    while reloj(5)~=0
        reloj=clock;
    end
end
end
end
end
end
end
end

%Inicialización del siguiente intervalo de reloj de tiempo absoluto
switch reloj(5)
case 10
    tiempoabs=20;
    tiempoabsaux=20;
case 20
    tiempoabs=30;
    tiempoabsaux=30;
case 30
    tiempoabs=40;
    tiempoabsaux=40;
case 40
    tiempoabs=50;
    tiempoabsaux=50;
case 50
    tiempoabs=0;
    tiempoabsaux=60;
case 0
    tiempoabs=10;
    tiempoabsaux=10;
end

%Inicio del algoritmo
reloj=clock;
muestras=0;
actualizacion=0;

while reloj(5)<=tiempoabsaux
    %Inicialización de Variables para medida básica
    sumaa=0;
    sumab=0;
    sumac=0;
    k=1;
    for i=n:n+Nsample-1
        %Actualización del reloj
        reloj=clock;
        if (reloj(5)==tiempoabs) & reloj(6)==0
            muestrainicio=muestras;
            auxreloj=clock;
            actualizacion=1;
        end

        %Señal trifásica de entrada: tensiones de fase
        t=(i-1)*Ts;
        Va=Vmaxa*sin(2*pi*fsis*t);
        Vb=Vmaxb*sin(2*pi*fsis*t-(2/3)*pi);
    end
end

```

```

Vc=Vmaxc*sin(2*pi*fsis*t+(2/3)*pi);

%Señal de entrada: tensiones de línea
Vab=Va-Vb;
Vbc=Vb-Vc;
Vca=Vc-Va;

%Componente de secuencia cero
homopolar=(1/sqrt(3))*(Va+Vb+Vc);

%Suma de las muestras al cuadrado para la señal
sumaa=sumaa+Va^2;
sumab=sumab+Vb^2;
sumac=sumac+Vc^2;

%Almacenamiento de datos para la señal de entrada
V1(k)=Vab;
V2(k)=Vbc;
V3(k)=Vca;
homp(k)=homopolar;
k=k+1;

%Conteo de muestras y actualización del reloj
muestras=muestras+1;
reloj=clock;
end

%Valor eficaz de doce ciclos: Amplitud de la tensión de Alimentación
Ueficaza=sqrt(sumaa/Nsample);
Ueficazb=sqrt(sumab/Nsample);
Ueficazc=sqrt(sumac/Nsample);

%Transformada de Fourier para las tensiones de línea
ffta=fft(V1);
fftb=fft(V2);
fftc=fft(V3);

%Transformada de Fourier para la tension homopolar
ffthomp=fft(homp);
magnituda=abs(ffta(13));
magnitudb=abs(fftb(13));
magnitudc=abs(fftc(13));
magnitudhomp=abs(ffthomp(13));

%Tensión eficaz de la componente fundamental de las tensiones de línea
Uabfund=(sqrt(2)/Nsample)*magnituda;
Ubcfund=(sqrt(2)/Nsample)*magnitudb;
Ucafund=(sqrt(2)/Nsample)*magnitudc;

%Tensión eficaz de la componente fundamental de la tensión homopolar
tensionhomopolar=(sqrt(2)/Nsample)*magnitudhomp;
%Componente Inversa
u2=invversa(Uabfund,Ubcfund,Ucafund);

%Componente de Secuencia positiva
positiva=secpositiva(Uabfund,Ubcfund,Ucafund);

%Componente homopolar
u0=(tensionhomopolar/positiva)*100;

%Sobretension y Subtension

%-----Para la fase A
[Ualtaa,Ubajaa]=valoraltobajo(Ueficaza,Udin);

%-----Para la fase B
[Ualtab,Ubajab]=valoraltobajo(Ueficazb,Udin);

```

```

%-----Para la fase C
[Ualtac, Ubajac]=valoraltoabajo(Ueficazc,Udin);

%Suma de los valores de doce ciclos para la Agregación de 10 minutos
sumau2=sumau2+u2^2;
sumau0=sumau0+u0^2;
sumaueficaza=sumaueficaza+Ueficaza^2;
sumaueficazb=sumaueficazb+Ueficazb^2;
sumaueficazc=sumaueficazc+Ueficazc^2;
sumaualtaa=sumaualtaa+Ualtaa^2;
sumaubajaa=sumaubajaa+Ubajaa^2;
sumaualtab=sumaualtab+Ualtab^2;
sumaubajab=sumaubajab+Ubajab^2;
sumaualtac=sumaualtac+Ualtac^2;
sumaubajac=sumaubajac+Ubajac^2;

%Evaluacion de la Agregacion de 10 minutos
if (actualizacion~=1)
    n=n+2048;
    auxcont12=auxcont12+1;
    reloj=clock;
else

    %Desbalance
    u210min(cont10min)=sqrt(sumau2/(auxcont12+1));
    u010min(cont10min)=sqrt(sumau0/(auxcont12+1));

    %Amplitud de la tensión
    Ueficaza10min(cont10min)=sqrt(sumaueficaza/(auxcont12+1));
    Ueficazb10min(cont10min)=sqrt(sumaueficazb/(auxcont12+1));
    Ueficazc10min(cont10min)=sqrt(sumaueficazc/(auxcont12+1));

    %Sobretension y subtension
    Ualtaa10min(cont10min)=sqrt(sumaualtaa/(auxcont12+1));
    Ubajaa10min(cont10min)=sqrt(sumaubajaa/(auxcont12+1));
    Ualtab10min(cont10min)=sqrt(sumaualtab/(auxcont12+1));
    Ubajab10min(cont10min)=sqrt(sumaubajab/(auxcont12+1));
    Ualtac10min(cont10min)=sqrt(sumaualtac/(auxcont12+1));
    Ubajac10min(cont10min)=sqrt(sumaubajac/(auxcont12+1));

    %Actualización para la nueva ventana de agregación de 10 minutos
    n=muestrainicio;

    %Actualización del intervalo de reloj de tiempo absoluto
    switch auxreloj(5)
        case 10
            tiempoabs=20;
            tiempoabsaux=20;
        case 20
            tiempoabs=30;
            tiempoabsaux=30;
        case 30
            tiempoabs=40;
            tiempoabsaux=40;
        case 40
            tiempoabs=50;
            tiempoabsaux=50;
        case 50
            tiempoabs=0;
            tiempoabsaux=60;
        case 0
            tiempoabs=10;
            tiempoabsaux=10;
    end

    %Suma de los valores de 10 minutos para la agregación de 2 horas

```

```

sumau210min=sumau210min+u210min(cont10min)^2;
sumau010min=sumau010min+u010min(cont10min)^2;
sumaueficaza10min=sumaueficaza10min+Ueficaza10min(cont10min)^2;
sumaueficazb10min=sumaueficazb10min+Ueficazb10min(cont10min)^2;
sumaueficazc10min=sumaueficazc10min+Ueficazc10min(cont10min)^2;
sumaualtaa10min=sumaualtaa10min+Ualtaa10min(cont10min)^2;
sumaubajaa10min=sumaubajaa10min+Ubajaa10min(cont10min)^2;
sumaualtab10min=sumaualtab10min+Ualtab10min(cont10min)^2;
sumaubajab10min=sumaubajab10min+Ubajab10min(cont10min)^2;
sumaualtac10min=sumaualtac10min+Ualtac10min(cont10min)^2;
sumaubajac10min=sumaubajac10min+Ubajac10min(cont10min)^2;

if (cont10min==w2hr*12)
    %Desbalance
    u22hr(cont2hr)=sqrt(sumau210min/12);
    u02hr(cont2hr)=sqrt(sumau010min/12);

    %Amplitud de la tensión
    Ueficaza2hr(cont2hr)=sqrt(sumaueficaza10min/12);
    Ueficazb2hr(cont2hr)=sqrt(sumaueficazb10min/12);
    Ueficazc2hr(cont2hr)=sqrt(sumaueficazc10min/12);
    %Sobretension y subtension

    Ualtaa2hr(cont2hr)=sqrt(sumaualtaa10min/12);
    Ubajaa2hr(cont2hr)=sqrt(sumaubajaa10min/12);
    Ualtab2hr(cont2hr)=sqrt(sumaualtab10min/12);
    Ubajab2hr(cont2hr)=sqrt(sumaubajab10min/12);
    Ualtac2hr(cont2hr)=sqrt(sumaualtac10min/12);
    Ubajac2hr(cont2hr)=sqrt(sumaubajac10min/12);

    %Inicialización de Variables
    sumau210min=0; sumau010min=0; sumaueficaza10min=0; sumaueficazb10min=0;
    sumaueficazc10min=0; sumaualtaa10min=0; sumaubajaa10min=0; sumaualtab10min=0;
    sumaubajab10min=0; sumaualtac10min=0; sumaubajac10min=0;

    %Actualización del conteo
    cont2hr=cont2hr+1;
    w2hr=w2hr+1;
end

%Inicialización de sumadores
sumau2=0; sumau0=0; sumaueficaza=0; sumaueficazb=0; sumaueficazc=0; sumaualtaa=0; sumaubajaa=0;
sumaualtab=0; sumaubajab=0; sumaualtac=0; sumaubajac=0;

%Actualización de contadores
cont10min=cont10min+1;
auxcont12=0;
muestras=0;
cont12=0;
actualizacion=0;
end
cont12=cont12+1;
end

fprintf('\nRESULTADOS OBTENIDOS\n');

%Resultados Agregación 10 minutos

fprintf('\nMEDIDA DE AGREGACION DE 10 MINUTOS\n');

fprintf('Componente Inversa\n'); disp(u210min)
fprintf('Componente Homopolar\n'); disp(u010min)
fprintf('Valor eficaz para la fase A\n'); disp(Ueficaza10min)
fprintf('Valor eficaz para la fase B\n'); disp(Ueficazb10min)
fprintf('Valor eficaz para la fase C\n'); disp(Ueficazc10min)
fprintf('Sobretension para la fase A\n'); disp(Ualtaa10min)
fprintf('Subtension para la fase A\n'); disp(Ubajaa10min)

```

```
fprintf('Sobretension para la fase B\n'); disp(Ualtab10min)
fprintf('Subtension para la fase B\n'); disp(Ubajab10min)
fprintf('Sobretension para la fase C\n'); disp(Ualtac10min)
fprintf('Subtension para la fase C\n'); disp(Ubajac10min)
```

%Resultado Agregación de 2 horas

```
fprintf('\nMEDIDA DE AGREGACION DE 2 HORAS\n');
```

```
fprintf('Componente Inversa\n'); disp(u22hr)
fprintf('Componente Homopolar\n'); disp(u02hr)
fprintf('Valor eficaz para la fase A\n'); disp(Ueficaza2hr)
fprintf('Valor eficaz para la fase B\n'); disp(Ueficazb2hr)
fprintf('Valor eficaz para la fase C\n'); disp(Ueficazc2hr)
fprintf('Sobretension para la fase A\n'); disp(Ualtaa2hr)
fprintf('Subtension para la fase A\n'); disp(Ubajaa2hr)
fprintf('Sobretension para la fase B\n'); disp(Ualtab2hr)
fprintf('Subtension para la fase B\n'); disp(Ubajab2hr)
fprintf('Sobretension para la fase C\n'); disp(Ualtac2hr)
fprintf('Subtension para la fase C\n'); disp(Ubajac2hr)
```

## ANEXO C. ALGORITMO PARA LA MEDICIÓN DE LAS PERTURBACIONES TRANSITORIAS DE TENSIÓN EN LENGUAJE C

### c1) Medidor monofásico implementado con muestra fija.

```
//DISEÑO DEL PROTOTIPO DEL MÓDULO PARA LA MONITORIZACIÓN DE LAS PERTURBACIONES
//DE TENSIÓN PARA UNA UNIDAD CONSTRUCTIVA EN EL ANÁLISIS DE LOS PARÁMETROS
//DE CALIDAD DE POTENCIA ELÉCTRICA CON BASE EN EL ESTÁNDAR IEC 61000-4-30
```

```
//ALGORITMO PARA LA MEDICIÓN DE LAS PERTURBACIONES DE TENSIÓN DE ESTADO
//TRANSITORIO PARA UNA RED MONOFÁSICA
//HUECOS DE TENSIÓN, INTERRUPCIONES DE TENSIÓN Y SOBRETENSIONES TEMPORALES.
```

```
// _____ ENTRADAS _____
```

```
//Definiciones.
```

```
//"Tensión eficaz refrescada cada semiperiodo(Uef_medio_ciclo): Valor de la tensión
//eficaz medida en un periodo, comenzando en un paso por cero de la fundamnetal,
//y refrescada cada semiperiodo."[IEC 61000-4-30:2003.]
```

```
//-yfundamental: La componente fundamental de la señal de la red es
//utilizada para detectar el cruce por cero al inicio del periodo según lo
//establecido en el Estándar IEC 61000-4-30.
//Para la medicion con muestra constante, el algoritmo detecta el cruce por
//cero al comienzo del periodo y la muestra es almacenada en la variable n,
//en el caso de la medición con muestra variable, el algoritmo detecta los
//cruces por cero al inicio y al final del periodo el cruce por cero al
//inicio es almacenado en n y el cruce por cero al final del periodo se
//almacena en la variable N2.
```

```
//-yn: es la señal de la red, con armonicos, interarmonicos, flicker, etc.
```

```
//-Nsample: muestras por ciclo de la ventana de observación, en el caso de
//las pruebas realizadas este fue de 128 muestras/ciclo.
```

```
//-umbral_inicio_interrupcion[0]: umbral donde comienza la interrupción de tensión.
//-umbral_inicio_hueco[0]: umbral donde comienza el hueco de tensión.
//-umbral_inicio_sobretension[0]: umbral donde comienza la sobretensión temporal.
//-umbral_final_interrupcion[0]: umbral donde finaliza la interrupción de tensión.
//-umbral_final_hueco[0]: umbral donde finaliza el hueco de tensión.
//-umbral_final_sobretension[0]: umbral donde finaliza la sobretensión temporal.
```

```
//-Udin (Tensión de entrada declarada): tension declarada por el usuario u
//operador de red para tomarla como referencia en la declaración de la
//histeresis de los eventos.
```

```
//-Urd (Tensión de referencia deslizante): tension que varia cada doce ciclos,
//La tension Urd actual esta en función de la tensión Urd anterior mas la
//tensión eficaz de doce ciclos.
```

```
//histeresis_interrupcion: porcentaje de udin, usado para determinar el
//umbral final de la interrupción.
```

```
//histeresis_hueco:porcentaje de udin, usado para determinar el
//umbral final del hueco.
```

```
//histeresis_sobretension:porcentaje de udin, usado para determinar el
//umbral final de la sobretensión temporal.
```

```
//-umbral_variable: debido a que los umbrales pueden ser un porcentaje de
```

```

//Udin o Urd, se deja una variable para que el usuario decida que
//referencia utilizar.

//-Tm: Tiempo de muestreo del conversor. Esta variable está en funcion de
//la frecuencia medida. La duración de la medida esta en funcion de los
//semiciclos medidos, estos a su vez se multiplicam por el tiempo de
//muestreo medido, de allí la importancia de su precisión.

//_____ SALIDAS _____
//inicio_hueco_fecha /
//inicio_hueco_hora /
//final_hueco_fecha / Datos solicitados por la CREG
//final_hueco_hora /

//tiempo_hueco /
//Tension_Residual / Datos solicitados por el estándar
//Profundidad /
//muestra_hueco

//inicio_interrupcion_fecha /
//inicio_interrupcion_hora /
//final_interrupcion_fecha / Datos solicitados por la CREG
//final_interrupcion_hora /
//DES y FES
//Interrupciones superiores a un minuto/

//tiempo_interrupcion / Datos solicitados por el estándar

//inicio_sobretension_fecha /
//inicio_sobretension_hora /
//final_sobretension_fecha / Datos solicitados por la CREG
//final_sobretension_hora /

//Amplitud_maxima_sobretension /
//tiempo_sobretension / Datos solicitados por el estándar
//muestra_sobretension

/*Declaración de librerías*/
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<stdlib.h>
#define begin {
#define end }
main()
{
float Udin=120;
//_____ INICIALIZACIÓN DE VARIABLES _____

//_____ Interrupción _____
float umbral_inicio_interrupcion[10];
float umbral_final_interrupcion[10];
umbral_inicio_interrupcion[0]=0.1*Udin;
#define histeresis_interrupcion (5/100)*Udin;
umbral_final_interrupcion[0]=0.1*Udin+histeresis_interrupcion;
int logica_interrupcion_actual,logica_interrupcion_anterior,eneable_interrupcion;
int a1=0,a11=0,tiempo1=0,proceso_marcado,logica_max_interrupcion;
float muestra_interrupcion[300],tiempo_interrupcion[200];

//_____ Hueco de tensión _____
int logica_hueco_actual=0,logica_hueco_anterior=0,a22=0,a2=0,tiempo2,eneable_hueco,logica_max_hueco;
float umbral_inicio_hueco[10],muestra_hueco[250],Tension_Residual[100],profundidad[100];
float umbral_final_hueco[10],Urd[100],tiempo_hueco[100];
Urd[0]=Udin;

```

```

    umbral_inicio_hueco[0]=0.9*Udin;
    #define histeresis_hueco (5/100)*Udin;
    umbral_final_hueco[0]=0.9*Udin+histeresis_hueco;

// _____ Sobretensión temporal _____

int
logica_sobretension_actual,logica_sobretension_anterior,a44=0,a4=0,tiempo4,eneable_sobretension,logica_maxima_sobretension;
float muestra_sobretension[250],tiempo_sobretension[250],Amplitud_maxima_sobretension[250];
float umbral_inicio_sobretension[10],umbral_final_sobretension[10],error;
#define histeresis_sobretension (5/100)*Udin;
umbral_inicio_sobretension[0]=1.1*Udin;
umbral_final_sobretension[0]=1.1*Udin-histeresis_sobretension;

// _____ Cruces por cero _____
int cero1,cero2,senal_cero,signo1,signo2,cruce,tramo1,tramo2,logica_tramo1,logica_tramo2,f,umbral_variable=0;

// _____ Generación de la señal _____

int i;
float yfundamental[12800],Amplitud1=30,Amplitud2=40,pi=3.14159265358979,Tm=0.0001302083333;
float senal[12800],va[13000];
float Ts=0.0001302083333,fsis=60,Vmaxa=120*sqrt(2),suma,Uef_12_ciclo[1000];
int nmax=12800,inicio[2000],evento[1000],habilitar_Urd,resultado,senal_cero_interrupcion=1;
int n11=1678,n22=2500,n33=4800,n44=6300;

int var_umbral=0,nsample=128;
float Uef_medio_ciclo[700],maximo=0;
Uef_medio_ciclo[0]=120;
int k=1,n=0,j=0,h=-1,q=0,N2,var=0,muestra_variable=0;

// _____ Generación de la señal por tramos _____
for (i=0;i<12800;i++)
{
    if (i<n11)
    {
        va[i]=Vmaxa*sin(2*pi*fsis*i*Ts);
    }
    if (i>=n11 && i<=n22)
    {
        va[i]=Amplitud1*sqrt(2)*sin(2*pi*fsis*i*Ts);
    }
    if (i>=n22 && i<=n33)
    {
        va[i]=Vmaxa*sin(2*pi*fsis*i*Ts);
    }
    if (i>=n33 && i<=n44)
    {
        va[i]=Amplitud2*sqrt(2)*sin(2*pi*fsis*i*Ts);
    }
    if (i>n44)
    {
        va[i]=Vmaxa*sin(2*pi*fsis*i*Ts);
    }
    yfundamental[i]=va[i];
}

// _____ algoritmo general _____
n=1;
while (n<nmax-200)
begin
// _____ Detección de cruces por cero _____
    while (q!=1)
    begin
        n++;
// _____ búsqueda amplitud cero _____

```

```

if (yfundamental[n]==0)
begin
cero1=1;
end
else
begin
cero1=0;
end

if (yfundamental[n+1]==0)
begin
cero2=1;
end
else
begin
cero2=0;
end
//_____ Deteccion senal cero_____
if ((cero1 && cero2)==1)
begin
senal_cero++;
end
else
begin
senal_cero=0;
end

if (senal_cero==64)
begin
senal_cero_interrupcion=1;
senal_cero=0;

end
else
begin
senal_cero_interrupcion=0;
end

if (senal_cero_interrupcion==1)
begin
if (muestra_variable==0)
begin
inicio[0]=n;
end
else
begin
inicio[0]=n;
inicio[2]=n+nsample;
end
break;
end
else
begin
if (yfundamental[n]<0)
begin
signo1=1;
end
else
begin
signo1=0;
end

if (yfundamental[n+1]<0)
begin
signo2=1;
end
else

```

```

begin
signo2=0;
end
if (yfundamental[n]>maximo)
begin
maximo=yfundamental[n];
end
if (signo1!=signo2)
begin
n++;
for (i=n;i<n+nsample-1;i++)
begin
error=(fabs(yfundamental[i]-yfundamental[i+1]))/maximo;

if (error>0.05)
begin
f=0;
break;
end
else
begin
f=1;
end
end

if (f==0)
begin
n=n+nsample+2;
signo1=signo2=0;

if (eneable_hueco==1)
begin
tiempo2++;
tiempo_hueco[a2]=tiempo2*Tm*(nsample/2);
end
if (eneable_sobretension==1)
begin
tiempo4++;
tiempo_sobretension[a4]=tiempo4*Tm*(nsample/2);
end

end
else
begin
N2=n+nsample-1;
break;
end
signo1=signo2=0;
end
end
end

//_____ Estimación de la tensión eficaz muestra fija_____
suma=0;
for (j=n;j<N2;j++)
{
suma=suma+pow(va[j],2);
}
Uef_medio_ciclo[k]=sqrt(suma/nsample);

//_____ interrupción de tensión_____
if (Uef_medio_ciclo[k]<umbral_inicio_interrupcion[var_umbral])
begin

```

```

logica_interrupcion_actual=1;
end
else
begin
logica_interrupcion_actual=0;
end

if (Uef_medio_ciclo[k-1]>umbral_inicio_interrupcion[var_umbral])
begin
logica_interrupcion_anterior=1;
end
else
begin
logica_interrupcion_anterior=0;
end

if ((logica_interrupcion_actual && logica_interrupcion_anterior)==1)
begin
tiempo1=2;
muestra_interrupcion[a11]=0.1*Udin+0.1;
a11++;
a1++;
eneable_interrupcion=1;
proceso_marcado=1;
end

if (Uef_medio_ciclo[k]>umbral_final_interrupcion[var_umbral])
begin
logica_max_interrupcion=1;
end
else
begin
logica_max_interrupcion=0;
end

if ((logica_max_interrupcion && eneable_interrupcion)==1)
begin
eneable_interrupcion=proceso_marcado=0;
end

if (eneable_interrupcion==1)
begin
tiempo1++;
tiempo_interrupcion[a1]=tiempo1*Tm*(nsample/2);

if (Uef_medio_ciclo[k]!=0)
begin
a11++;
muestra_interrupcion[a11]=Uef_medio_ciclo[k];
end

end

// _____ Hueco de tensión _____

if (Uef_medio_ciclo[k]<umbral_inicio_hueco[var_umbral])
begin
logica_hueco_actual=1;
end
else
begin
logica_hueco_actual=0;
end

if (Uef_medio_ciclo[k-1]>umbral_inicio_hueco[var_umbral])
begin

```

```

logica_hueco_anterior=1;
end
else
begin
logica_hueco_anterior=0;
end

if ((logica_hueco_actual && logica_hueco_anterior)==1)
begin
a22++;
a2++;
tiempo2=1;
muestra_hueco[a22]=0.0;
eneable_hueco=1;
Tension_Residual[a2]=Uef_medio_ciclo[k];
profundidad[a2]=Uef_medio_ciclo[k];
proceso_marcado=1;
end

if (Uef_medio_ciclo[k]>umbral_final_hueco[var_umbral])
begin
logica_max_hueco=1;
end
else
begin
logica_max_hueco=0;
end

if ((logica_max_hueco && eneable_hueco)==1)
begin
// final_hueco_fecha[a2]=;
// final_hueco_hora[a2]=;
eneable_hueco=proceso_marcado=0;
end

if (eneable_hueco==1)
begin
tiempo2++;
tiempo_hueco[a2]=tiempo2*Tm*(nsample/2);

if (Uef_medio_ciclo!=0)
begin
a22++;
muestra_hueco[a22]=Uef_medio_ciclo[k];
end

if (Uef_medio_ciclo[k]<Tension_Residual[a2])
begin
if (umbral_variable==0)
begin
Tension_Residual[a2]=Uef_medio_ciclo[k];
profundidad[a2]=Udin-Uef_medio_ciclo[k];
end
else
begin
Tension_Residual[a2]=Uef_medio_ciclo[k];
profundidad[a2]=Urd[var_umbral]-Uef_medio_ciclo[k];
//Urd_hueco[a2]=Urd[var_umbral];
end
end

end

// _____ Sobretensión Temporal _____

if (Uef_medio_ciclo[k]>umbral_inicio_sobretension[var_umbral])
begin

```

```

logica_sobretension_actual=1;
end
else
begin
logica_sobretension_actual=0;
end

if (Uef_medio_ciclo[k-1]<umbral_inicio_sobretension[var_umbral])
begin
logica_sobretension_anterior=1;
end
else
begin
logica_sobretension_anterior=0;
end

    if ((logica_sobretension_actual && logica_sobretension_anterior)==1)
    begin
    a44++;
    a4++;
// inicio_sobretension_fecha[a4]=;
// inicio_sobretension_hora[a4]=;
    tiempo4=1;
    muestra_sobretension[a44]=0;
    eneable_sobretension=1;
    Amplitud_maxima_sobretension[a4]=Uef_medio_ciclo[k];
    proceso_marcado=1;
    end

    if (Uef_medio_ciclo[k]<umbral_final_sobretension[var_umbral])
    begin
    logica_maxima_sobretension=1;
    end
    else
    begin
    logica_maxima_sobretension=0;
    end

    if ((logica_maxima_sobretension && eneable_sobretension)==1)
    begin
// final_sobretension_fecha[a4]=;
// final_sobretension_hora[a4]=;
    eneable_sobretension=proceso_marcado=0;
    end

    if (eneable_sobretension==1)
    begin
    a44++;
    tiempo4++;
    tiempo_sobretension[a4]=tiempo4*Tm*(nsample/2);
    muestra_sobretension[a44]=Uef_medio_ciclo[k];

    if (Uef_medio_ciclo[k]>Amplitud_maxima_sobretension[a4])
    begin
    Amplitud_maxima_sobretension[a4]=Uef_medio_ciclo[k];
    end
    end

//
//-----

if (n>1536)
begin
var++;
evento[var]=proceso_marcado;
end

if (var==23)

```

```

begin
habilitar_Urd=1;
end
else
begin
habilitar_Urd=0;
end

//
if ((habilitar_Urd && umbral_variable)==1)
begin
resultado=0;
for (j=1;j=24;j++)
begin
resultado=resultado+evento[j];
end

if (resultado==0)
begin
var_umbral++;
Urd[var_umbral]=0.9967*Urd[var_umbral-1]+0.0033*Uef_12_ciclo[var_umbral];
end
else
begin
var++;
Urd[var_umbral]=Urd[var_umbral-1];
end

        umbral_inicio_interrupcion[var_umbral]=0.1*Urd[var_umbral];
        umbral_final_interrupcion[var_umbral]=0.1*Urd[var_umbral]+histeresis_interrupcion;

        umbral_inicio_hueco[var_umbral]=0.9*Urd[var_umbral];
        umbral_final_hueco[var_umbral]=0.9*Urd[var_umbral]+histeresis_hueco;

        umbral_inicio_sobretension[var_umbral]=1.1*Urd[var_umbral];
        umbral_final_sobretension[var_umbral]=1.1*Urd[var_umbral]-histeresis_sobretension;

var=0;
end

//

k++;

end

printf("_____ Interrupcion de tension _____ \n");
        for (j=1;j<=a1;j++)
        {
                printf("Duracion del evento %f\n",tiempo_interrupcion[j]);
        }
printf("\n");
printf("_____ Hueco de tension _____ \n");
        for (j=1;j<=a2;j++)
        {
                printf("Duracion del evento Tension Residual Profundidad %f %f %f
\n",tiempo_hueco[j],Tension_Residual[j],profundidad[j]);
        }
printf("\n");
printf("_____ Sobretension Temporal _____ \n");
        for (j=1;j<=a4;j++)
        {
                printf("Duracion del evento Amplitud maxima %f %f
%\n",tiempo_sobretension[j],Amplitud_maxima_sobretension[j],i);
        }
getch();
}

```

## c2) Medidor monofásico implementado con muestra variable.

//DISEÑO DEL PROTOTIPO DEL MÓDULO PARA LA MONITORIZACIÓN DE LAS PERTURBACIONES  
//DE TENSIÓN PARA UNA UNIDAD CONSTRUCTIVA EN EL ANÁLISIS DE LOS PARÁMETROS  
//DE CALIDAD DE POTENCIA ELÉCTRICA CON BASE EN EL ESTÁNDAR IEC 61000-4-30

//ALGORITMO PARA LA MEDICIÓN DE LAS PERTURBACIONES DE TENSIÓN DE ESTADO  
//TRANSITORIO PARA UNA RED MONOFÁSICA  
//HUECOS DE TENSIÓN, INTERRUPCIONES DE TENSIÓN Y SOBRETENSIONES TEMPORALES.

// \_\_\_\_\_ ENTRADAS \_\_\_\_\_

//Definiciones.

//"Tensión eficaz refrescada cada semiperiodo(Uef\_medio\_ciclo): Valor de la tensión  
//eficaz medida en un período, comenzando en un paso por cero de la fundamnetal,  
//y refrescada cada semiperiodo." [IEC 61000-4-30:2003.]

//-yfundamental: La componente fundamental de la señal de la red es  
//utilizada para detectar el cruce por cero al inicio del periodo según lo  
//establecido en el Estándar IEC 61000-4-30.  
//Para la medición con muestra constante, el algoritmo detecta el cruce por  
//cero al comienzo del periodo y la muestra es almacenada en la variable n,  
//en el caso de la medición con muestra variable, el algoritmo detecta los  
//cruces por cero al inicio y al final del periodo el cruce por cero al  
//inicio es almacenado en n y el cruce por cero al final del periodo se  
//almacena en la variable N2.

//-yn: es la señal de la red, con armónicos, interarmónicos, flicker, etc.

//-Nsample: muestras por ciclo de la ventana de observación, en el caso de  
//las pruebas realizadas este fue de 128 muestras/ciclo.

//-umbral\_inicio\_interrupcion[0]: umbral donde comienza la interrupción de tensión.  
//-umbral\_inicio\_hueco[0]: umbral donde comienza el hueco de tensión.  
//-umbral\_inicio\_sobretension[0]: umbral donde comienza la sobretensión temporal.  
//-umbral\_final\_interrupcion[0]: umbral donde finaliza la interrupción de tensión.  
//-umbral\_final\_hueco[0]: umbral donde finaliza el hueco de tensión.  
//-umbral\_final\_sobretension[0]: umbral donde finaliza la sobretensión temporal.

//-Udin (Tensión de entrada declarada): tensión declarada por el usuario u  
//operador de red para tomarla como referencia en la declaración de la  
//histeresis de los eventos.

//-Urd (Tensión de referencia deslizante): tensión que varía cada doce ciclos,  
//La tensión Urd actual está en función de la tensión Urd anterior más la  
//tensión eficaz de doce ciclos.

//histeresis\_interrupcion: porcentaje de udin, usado para determinar el  
//umbral final de la interrupción.

//histeresis\_hueco: porcentaje de udin, usado para determinar el  
//umbral final del hueco.

//histeresis\_sobretension: porcentaje de udin, usado para determinar el  
//umbral final de la sobretensión temporal.

//-umbral\_variable: debido a que los umbrales pueden ser un porcentaje de  
//Udin o Urd, se deja una variable para que el usuario decida que  
//referencia utilizar.

//-Tm: Tiempo de muestreo del convertidor. Esta variable está en función de  
//la frecuencia medida. La duración de la medida está en función de los

//semiciclos medidos, estos a su vez se multimplicam por el tiempo de  
//muestreo medido, de allí la importancia de su precisión.

// \_\_\_\_\_ SALIDAS \_\_\_\_\_

//inicio\_hueco\_fecha /  
//inicio\_hueco\_hora /  
//final\_hueco\_fecha / Datos solicitados por la CREG  
//final\_hueco\_hora /

//tiempo\_hueco /  
//Tension\_Residual / Datos solicitados por el estándar  
//Profundidad /  
//muestra\_hueco

//inicio\_interruccion\_fecha /  
//inicio\_interruccion\_hora /  
//final\_interruccion\_fecha / Datos solicitados por la CREG  
//final\_interruccion\_hora /  
//DES y FES  
//Interrupciones superiores a un minuto/

//tiempo\_interruccion / Datos solicitados por el estándar

//inicio\_sobretension\_fecha /  
//inicio\_sobretension\_hora /  
//final\_sobretension\_fecha / Datos solicitados por la CREG  
//final\_sobretension\_hora /

//Amplitud\_maxima\_sobretension /  
//tiempo\_sobretension / Datos solicitados por el estándar  
//muestra\_sobretension

/\*Declaración de librerías\*/

```
#include<stdio.h>  
#include<conio.h>  
#include<math.h>  
#include<stdlib.h>  
#define begin {  
#define end }
```

main()

```
{  
    float Udin=120;
```

// \_\_\_\_\_ INICIALIZACIÓN DE VARIABLES \_\_\_\_\_

```
// _____ Interrupción _____  
float umbral_inicio_interruccion[10];  
float umbral_final_interruccion[10];  
    umbral_inicio_interruccion[0]=0.1*Udin;  
    #define histeresis_interruccion (5/100)*Udin;  
    umbral_final_interruccion[0]=0.1*Udin+histeresis_interruccion;  
int logica_interruccion_actual,logica_interruccion_anterior,eneable_interruccion;  
int a1=0,a11=0,tiempo1=0,proceso_marcado,logica_max_interruccion;  
float muestra_interruccion[300],tiempo_interruccion[200];
```

```
// _____ Hueco de tensión _____  
int logica_hueco_actual=0,logica_hueco_anterior=0,a22=0,a2=0,tiempo2,eneable_hueco,logica_max_hueco;  
float umbral_inicio_hueco[10],muestra_hueco[250],Tension_Residual[100],profundidad[100];  
float umbral_final_hueco[10],Urd[100],tiempo_hueco[100];  
    Urd[0]=Udin;  
    umbral_inicio_hueco[0]=0.9*Udin;  
    #define histeresis_hueco (5/100)*Udin;  
    umbral_final_hueco[0]=0.9*Udin+histeresis_hueco;  
// _____ Sobretensión temporal _____
```

```

int
logica_sobretension_actual,logica_sobretension_anterior,a44=0,a4=0,tiempo4,eneable_sobretension,logica_maxima_sobretension;
float muestra_sobretension[250],tiempo_sobretension[250],Amplitud_maxima_sobretension[250];
float umbral_inicio_sobretension[10],umbral_final_sobretension[10],error;
#define histeresis_sobretension (5/100)*Udin;
umbral_inicio_sobretension[0]=1.1*Udin;
umbral_final_sobretension[0]=1.1*Udin-histeresis_sobretension;

// _____ Cruces por cero _____
int cero1,cero2,senal_cero,signo1,signo2,cruce,tramo1,tramo2,logica_tramo1,logica_tramo2,f,umbral_variable=0;

// _____ Generación de la señal _____

int i;
float yfundamental[12800],Amplitud1=30,Amplitud2=40,pi=3.14159265358979,Tm=0.0001302083333;
float senal[12800],va[13000];
float Ts=0.0001302083333,fsis=60,Vmaxa=120*sqrt(2),suma,Uef_12_ciclo[1000];
int nmax=12800,inicio[2000],evento[1000],habilitar_Urd,resultado,senal_cero_interrupcion=1;
int n11=1678,n22=2500,n33=4800,n44=6300;

int var_umbral=0,nsample=128;
float Uef_medio_ciclo[700],maximo=0;
Uef_medio_ciclo[0]=120;
int k=1,n=0,j=0,h=-1,q=0,N2,var=0;

// _____ Generación de la señal por tramos _____
for (i=0;i<12800;i++)
{
if (i<n11)
{
va[i]=Vmaxa*sin(2*pi*fsis*i*Ts);
}
if (i>=n11 && i<=n22)
{
va[i]=Amplitud1*sqrt(2)*sin(2*pi*fsis*i*Ts);
}
if (i>=n22 && i<=n33)
{
va[i]=Vmaxa*sin(2*pi*fsis*i*Ts);
}
if (i>=n33 && i<=n44)
{
va[i]=Amplitud2*sqrt(2)*sin(2*pi*fsis*i*Ts);
}
if (i>n44)
{
va[i]=Vmaxa*sin(2*pi*fsis*i*Ts);
}
yfundamental[i]=va[i];
}

// _____ algoritmo general _____
n=1;
while (n<nmax-200)
begin
// _____ Detección de cruces por cero _____
while (q!=1)
begin
n++;
// _____ búsqueda amplitud cero _____
if (yfundamental[n]==0)
begin
cero1=1;
end
else
begin

```

```

    cero1=0;
    end

    if (yfundamental[n+1]==0)
    begin
        cero2=1;
    end
    else
    begin
        cero2=0;
    end
// _____ Deteccion senal cero _____
    if ((cero1 && cero2)==1)
    begin
        senal_cero++;
    end
    else
    begin
        senal_cero=0;
    end

    if (senal_cero==64)
    begin
        senal_cero_interrupcion=1;
        senal_cero=0;

    end
    else
    begin
        senal_cero_interrupcion=0;
    end

    if (senal_cero_interrupcion==1)
    begin
        inicio[0]=n+64;
        inicio[2]=inicio[0]+nsample-1;

    break;
    end
    else
    begin
        if (yfundamental[n]<0)
        begin
            signo1=1;
        end
        else
        begin
            signo1=0;
        end

        if (yfundamental[n+1]<0)
        begin
            signo2=1;
        end
        else
        begin
            signo2=0;
        end
        if (yfundamental[n]>maximo)
        begin
            maximo=yfundamental[n];
        end

        if (signo1!=signo2)
        begin

            inicio[h]=n+1;

```

```

h++;
end

if (h==3)
begin

tramo1=inicio[1]-inicio[0];
tramo2=inicio[2]-inicio[1];
if (tramo1>(nsample/2-4) && tramo1<(nsample/2+4))
begin
logica_tramo1=1;
end
else
begin
logica_tramo1=0;
end

if (tramo2>(nsample/2-4) && tramo2<(nsample/2+4))
begin
logica_tramo2=1;
end
else
begin
logica_tramo2=0;
end

if ((logica_tramo1 && logica_tramo2)==1)
begin

for (i=inicio[0];i<(inicio[2]-1);i++)

begin
error=(fabs(yfundamental[i]-yfundamental[i+1]))/maximo;

if (error>0.05)
begin
f=0;
break;
end
else
begin
f=1;
end
end
maximo=0;
end

else
begin
f=0;
end

if (f==0)
begin
n=inicio[2]+2;
inicio[0]=inicio[1]=inicio[2]=0;
h=0;

if (eneable_hueco==1)
begin
tiempo2++;
tiempo_hueco[a2]=tiempo2*Tm*(nsample/2);
end
if (eneable_sobretension==1)
begin
tiempo4++;
tiempo_sobretension[a4]=tiempo4*Tm*(nsample/2);

```

```

                                end
                                end
                                else
                                begin
                                break;
                                end
                                signo1=signo2=0;
                                end
                                end
                                end
                                end

                                suma=0;
                                n=inicio[0]+2;

// _____ Estimación de la tensión eficaz muestra variable _____
                                for (j=inicio[0];j<inicio[2]-1;j++)
                                {
                                suma=suma+pow(va[j],2);
                                }
                                Uef_medio_ciclo[k]=sqrt(suma/(inicio[2]-inicio[0]+1));

                                h=0;

// _____ interrupción de tensión _____

if (Uef_medio_ciclo[k]<umbral_inicio_interrupcion[var_umbral])
begin
logica_interrupcion_actual=1;
end
else
begin
logica_interrupcion_actual=0;
end

if (Uef_medio_ciclo[k-1]>umbral_inicio_interrupcion[var_umbral])
begin
logica_interrupcion_anterior=1;
end
else
begin
logica_interrupcion_anterior=0;
end

if ((logica_interrupcion_actual && logica_interrupcion_anterior)==1)
begin
tiempo1=2;
muestra_interrupcion[a11]=0.1*Udin+0.1;
a11++;
a1++;
eneable_interrupcion=1;
proceso_marcado=1;
end

if (Uef_medio_ciclo[k]>umbral_final_interrupcion[var_umbral])
begin
logica_max_interrupcion=1;
end
else
begin
logica_max_interrupcion=0;
end

if ((logica_max_interrupcion && eneable_interrupcion)==1)
begin
eneable_interrupcion=proceso_marcado=0;

```

```

end

if (eneable_interrupcion==1)
begin
tiempo1++;
tiempo_interrupcion[a1]=tiempo1*Tm*(nsample/2);

if (Uef_medio_ciclo[k]!=0)
begin
a11++;
muestra_interrupcion[a11]=Uef_medio_ciclo[k];
end

end

// _____ Hueco de tensión _____

if (Uef_medio_ciclo[k]<umbral_inicio_hueco[var_umbral])
begin
logica_hueco_actual=1;
end
else
begin
logica_hueco_actual=0;
end

if (Uef_medio_ciclo[k-1]>umbral_inicio_hueco[var_umbral])
begin
logica_hueco_anterior=1;
end
else
begin
logica_hueco_anterior=0;
end

if ((logica_hueco_actual && logica_hueco_anterior)==1)
begin
a22++;
a2++;
tiempo2=1;
muestra_hueco[a22]=0.0;
eneable_hueco=1;
Tension_Residual[a2]=Uef_medio_ciclo[k];
profundidad[a2]=Uef_medio_ciclo[k];
proceso_marcado=1;
end

if (Uef_medio_ciclo[k]>umbral_final_hueco[var_umbral])
begin
logica_max_hueco=1;
end
else
begin
logica_max_hueco=0;
end

if ((logica_max_hueco && eneable_hueco)==1)
begin
// final_hueco_fecha[a2]=;
// final_hueco_hora[a2]=;
eneable_hueco=proceso_marcado=0;
end

if (eneable_hueco==1)
begin
tiempo2++;
tiempo_hueco[a2]=tiempo2*Tm*(nsample/2);

```

```

    if (Uef_medio_ciclo!=0)
    begin
    a22++;
    muestra_hueco[a22]=Uef_medio_ciclo[k];
    end

    if (Uef_medio_ciclo[k]<Tension_Residual[a2])
    begin
    if (umbral_variable==0)
    begin
    Tension_Residual[a2]=Uef_medio_ciclo[k];
    profundidad[a2]=Udin-Uef_medio_ciclo[k];
    end
    else
    begin
    Tension_Residual[a2]=Uef_medio_ciclo[k];
    profundidad[a2]=Urd[var_umbral]-Uef_medio_ciclo[k];
    //Urd_hueco[a2]=Urd[var_umbral];
    end
    end

    end

// _____ Sobretensión Temporal _____

if (Uef_medio_ciclo[k]>umbral_inicio_sobretension[var_umbral])
begin
logica_sobretension_actual=1;
end
else
begin
logica_sobretension_actual=0;
end

if (Uef_medio_ciclo[k-1]<umbral_inicio_sobretension[var_umbral])
begin
logica_sobretension_anterior=1;
end
else
begin
logica_sobretension_anterior=0;
end

    if ((logica_sobretension_actual && logica_sobretension_anterior)==1)
    begin
    a44++;
    a4++;
    // inicio_sobretension_fecha[a4]=;
    // inicio_sobretension_hora[a4]=;
    tiempo4=1;
    muestra_sobretension[a44]=0;
    eneable_sobretension=1;
    Amplitud_maxima_sobretension[a4]=Uef_medio_ciclo[k];
    proceso_marcado=1;
    end

    if (Uef_medio_ciclo[k]<umbral_final_sobretension[var_umbral])
    begin
    logica_maxima_sobretension=1;
    end
    else
    begin
    logica_maxima_sobretension=0;
    end

```

```

    if ((logica_maxima_sobretension && eneable_sobretension)==1)
    begin
//    final_sobretension_fecha[a4]=;
//    final_sobretension_hora[a4]=;
    eneable_sobretension=proceso_marcado=0;
    end

    if (eneable_sobretension==1)
    begin
    a44++;
    tiempo4++;
    tiempo_sobretension[a4]=tiempo4*Tm*(nsample/2);
    muestra_sobretension[a44]=Uef_medio_ciclo[k];

    if (Uef_medio_ciclo[k]>Amplitud_maxima_sobretension[a4])
    begin
    Amplitud_maxima_sobretension[a4]=Uef_medio_ciclo[k];
    end
    end

//
if (n>1536)
begin
var++;
evento[var]=proceso_marcado;
end

if (var==23)
begin
habilitar_Urd=1;
end
else
begin
habilitar_Urd=0;
end
//
if ((habilitar_Urd && umbral_variable)==1)
begin
resultado=0;
for (j=1;j=24;j++)
begin
resultado=resultado+evento[j];
end

if (resultado==0)
begin
var_umbral++;
Urd[var_umbral]=0.9967*Urd[var_umbral-1]+0.0033*Uef_12_ciclo[var_umbral];
end
else
begin
var++;
Urd[var_umbral]=Urd[var_umbral-1];
end

    umbral_inicio_interrupcion[var_umbral]=0.1*Urd[var_umbral];
    umbral_final_interrupcion[var_umbral]=0.1*Urd[var_umbral]+histeresis_interrupcion;

    umbral_inicio_hueco[var_umbral]=0.9*Urd[var_umbral];
    umbral_final_hueco[var_umbral]=0.9*Urd[var_umbral]+histeresis_hueco;

    umbral_inicio_sobretension[var_umbral]=1.1*Urd[var_umbral];
    umbral_final_sobretension[var_umbral]=1.1*Urd[var_umbral]-histeresis_sobretension;

var=0;
end

```

```

//
k++;
end
printf("_____ Interrupcion de tension _____ \n");
    for (j=1;j<=a1;j++)
    {
        printf("Duracion del evento %f\n",tiempo_interrupcion[j]);
    }
printf("\n");
printf("_____ Hueco de tension _____ \n");
    for (j=1;j<=a2;j++)
    {
        printf("Duracion del evento Tension Residual Profundidad %f %f %f
\n",tiempo_hueco[j],Tension_Residual[j],profundidad[j]);
    }
printf("\n");
printf("_____ Sobretension Temporal _____ \n");
    for (j=1;j<=a4;j++)
    {
        printf("Duracion del evento Amplitud maxima %f %f
%f\n",tiempo_sobretension[j],Amplitud_maxima_sobretension[j],i);
    }
getch();
}

```

### c3) Medidor trifásico implementado con muestra fija.

```

/*Declaración de librerías*/
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<stdlib.h>
#define begin {
#define end }
#include<time.h>

main()
{
//_____ Variables para la generación de la señal _____
float Ts=0.0001302083333,Vmax=120*sqrt(2),pi=3.14159265358979,Tm=0.0001302083333,fsis=60;
float YA[13000],YB[13000],YC[13000],yAfundamental[13000],yBfundamental[13000],yCfundamental[13000];
float AmplitudA=6,AmplitudB=150,AmplitudC=8,Udin=120;
int i,j,nmax=12800,Nsample=128,q=1,proceso_marcado;

//_____ Variables para la detección de cruces por cero fase A _____
int nA=0,ceroA1,ceroA2,senal_ceroA,signoA1,signoA2,senal_cero_intercepcionA,cruceA=0,tramo1A,tramo2A;
int logica_tramo1A,logica_tramo2A,inicioA[3],fA,h=0,senal_cero_interrupcionA;
float errorA;
//_____ Variables para la detección de cruces por cero fase B _____
int nB=0,ceroB1,ceroB2,senal_ceroB,signoB1,signoB2,senal_cero_intercepcionB,cruceB=0,tramo1B,tramo2B;
int logica_tramo1B,logica_tramo2B,inicioB[3],fB,senal_cero_interrupcionB;
float errorB;
//_____ Variables para la detección de cruces por cero fase C _____
int nC=0,ceroC1,ceroC2,senal_ceroC,signoC1,signoC2,senal_cero_intercepcionC,cruceC=0,tramo1C,tramo2C;
int logica_tramo1C,logica_tramo2C,inicioC[3],fC,senal_cero_interrupcionC;
float errorC;
//_____ Variables para la estimación de la tensión eficaz fase A _____
float sumaA,Uef_medio_ciclo_A[10000],maximoA=0;
int N2A;
//_____ Variables para la estimación de la tensión eficaz fase B _____
float sumaB,Uef_medio_ciclo_B[10000],maximoB=0;
int N2B;
//_____ Variables para la estimación de la tensión eficaz fase C _____

```

```

float sumaC,Uef_medio_ciclo_C[10000],maximoC=0;
int N2C;
//___Variables para la detección y evaluación de una interrupción de tensión___
float umbral_inicio_interrupcion[10];
float umbral_final_interrupcion[10];
    umbral_inicio_interrupcion[0]=0.1*Udin;
    #define histeresis_interrupcion (5/100)*Udin;
    umbral_final_interrupcion[0]=0.1*Udin+histeresis_interrupcion;
float actual_interrupcion,anterior_interrupcion,tiempo_interrupcion[1000];
int logica_interrupcion_actual,logica_interrupcion_anterior,a11=0,a1=0,tiempo1,habilitar_interrupcion;
int marcado_interrupcion,logica_max_interrupcion;
//___Variables para la detección y evaluación de un hueco de tensión___
float actual_hueco,anterior_hueco,muestra_hueco[1000],Tension_Residual[1000],tiempo_hueco[1000],Profundidad[1000];
int proceso_marcadoA,proceso_marcadoB,proceso_marcadoC,logica_hueco_actual,logica_hueco_anterior,habilitar_hueco;
int marcado_hueco,logica_max_hueco,tiempo2,a22=0,a2=0;
float umbral_inicio_hueco[10],umbral_final_hueco[10];
char fase_hueco[1000][3];
time_t inicio_hueco[1000], final_hueco[1000];
struct tm *tiempoComienzoPtr_hueco[1000], *tiempoFinalPtr_hueco[1000];

    umbral_inicio_hueco[0]=0.9*Udin;
    #define histeresis_hueco (5/100)*Udin;
    umbral_final_hueco[0]=0.9*Udin+histeresis_hueco;
//___Variables para la detección y evaluación de una Sobretensión Temporal___
float actual_sobretension,anterior_sobretension,muestra_sobretension[1000],tiempo_sobretension[1000];
float Amplitud_maxima_sobretension[1000];
int logica_sobretension_actual,logica_sobretension_anterior,a44=0,a4=0,tiempo4=0,marcado_sobretension;
int habilitar_sobretension,logica_max_sobretension;
float umbral_inicio_sobretension[10],umbral_final_sobretension[10];
char fase_sobretension[1000][3];
#define histeresis_sobretension (5/100)*Udin;
umbral_inicio_sobretension[0]=1.1*Udin;
umbral_final_sobretension[0]=1.1*Udin-histeresis_sobretension;
//___Variables para la detección de la ventana marcada de doce ciclos___
float Urd[1000],Uef_12_ciclos_A[1000],Uef_12_ciclos_B[1000],Uef_12_ciclos_C[1000];
Urd[0]=Udin;
Uef_medio_ciclo_A[0]=Uef_medio_ciclo_B[0]=Uef_medio_ciclo_C[0]=120;
int var_A=0,var_B=0,var_C=0,evento_A[1000],evento_B[1000],evento_C[1000],habilitar_Urd,umbral_variable=0;
int resultadoA,resultadoB,resultadoC,var_umbral=0,muestra_variable=0,k=1;

//_____ Generación de la señal _____
//_____ Fase A _____

    for (i=0;i<0+12800;i++)
    {
        if (i<2678)
        {
            YA[i]=Vmax*sin(2*pi*fsis*i*Ts);
        }
        if (i>=2678 && i<=6367)
        {
            YA[i]=AmplitudA*sqrt(2)*sin(2*pi*fsis*i*Ts);
        }
        if (i>6367)
        {
            YA[i]=Vmax*sin(2*pi*fsis*i*Ts);
        }
        yAfundamental[i]=YA[i];
    }

//_____ Fase B _____

    for (i=0;i<12800;i++)
    {
        if (i<2678)
        {
            YB[i]=Vmax*sin(2*pi*fsis*i*Ts);
        }
    }

```

```

        if (i>=2678 && i<=6367)
        {
            YB[i]=AmplitudB*sqrt(2)*sin(2*pi*fsis*i*Ts);
        }
        if (i>6367)
        {
            YB[i]=Vmax*sin(2*pi*fsis*i*Ts);
        }
        yBfundamental[i]=YB[i];
    }
// _____ Fase C _____
    for (i=0;i<12800;i++)
    {
        if (i<2678)
        {
            YC[i]=Vmax*sin(2*pi*fsis*i*Ts);
        }
        if (i>=2678 && i<=6367)
        {
            YC[i]=AmplitudC*sqrt(2)*sin(2*pi*fsis*i*Ts);
        }
        if (i>6367)
        {
            YC[i]=Vmax*sin(2*pi*fsis*i*Ts);
        }
        yCfundamental[i]=YC[i];
    }
// _____
while (nB<nmax)
begin

// _____ Cruce por cero Fase A _____
while (q!=-1)
begin
    while (q!=-1)
    begin
        nA=nA+1;
        if (yAfundamental[nA]==0)
        begin
            ceroA1=1;
        end
        else
        begin
            ceroA1=0;
        end
        if (yAfundamental[nA+1]==0)
        begin
            ceroA2=1;
        end
        else
        begin
            ceroA2=0;
        end

        if ((ceroA1 && ceroA2)==1)
        begin
            senal_ceroA++;
        end
        else
        begin
            senal_ceroA=0;
        end

        if (senal_ceroA==64)
        begin
            senal_cero_interrupcionA=1;
            senal_ceroA=0;
        end
    end
end

```

```

end
else
begin
senal_cero_interrupcionA=0;
end

if (senal_cero_interrupcionA==1)
begin

fA=0;
break;
end
else
begin
if (yAfundamental[nA]<0)
begin
signoA1=1;
end
else
begin
signoA1=0;
end

if (yAfundamental[nA+1]<0)
begin
signoA2=1;
end
else
begin
signoA2=0;
end
if (yAfundamental[nA]>maximoA)
begin
maximoA=yAfundamental[nA];
end
if (signoA1!=signoA2)
begin
nA++;

for (i=nA;i<(nA+Nsample-1);i++)

begin
errorA=(fabs(yAfundamental[i]-yAfundamental[i+1]))/maximoA;

if (errorA>0.05)
begin
fA=0;
break;
end
else
begin
fA=1;
end
end

if (fA==0)
begin

signoA1=signoA2=0;

break;
end
else
begin
N2A=nA+Nsample-1;
break;
end
end
end

```

```

end
end

end
signoA1=0;
signoA2=0;
// _____ Cruce por cero Fase B _____

while (q!=-1)
begin
nB++;
if (yBfundamental[nB]==0)
begin
cerob1=1;
end
else
begin
cerob1=0;
end
if (yBfundamental[nB+1]==0)
begin
cerob2=1;
end
else
begin
cerob2=0;
end

if ((cerob1 && cerob2)==1)
begin
senal_ceroB++;
end
else
begin
senal_ceroB=0;
end

if (senal_ceroB==64)
begin
senal_cero_interrupcionB=1;
senal_ceroB=0;
end
else
begin
senal_cero_interrupcionB=0;
end

if (senal_cero_interrupcionB==1)
begin

fB=0;
break;
end

else
begin

if (yBfundamental[nB]<0)
begin
signoB1=1;
end
else
begin
signoB1=0;
end

```

```

        if (yBfundamental[nB+1]<0)
        begin
        signoB2=1;
        end
        else
        begin
        signoB2=0;
        end
    if (yBfundamental[nB]>maximoB)
    begin
        maximoB=yBfundamental[nB];
    end
    if (signoB1!=signoB2)
    begin
        nB++;

                for (i=nB;i<nB+Nsample-1;i++)

                begin
                    errorB=(fabs(yBfundamental[i]-yBfundamental[i+1]))/maximoB;

                    if (errorB>0.05)
                    begin
                        fB=0;
                        break;
                    end
                    else
                    begin
                        fB=1;
                    end
                end

                if (fB==0)
                begin
                    signoB1=signoB2=0;
                    break;
                end
                else
                begin
                    N2B=nB+Nsample-1;
                    break;
                end

            end

        end

    end
    signoB1=0;
    signoB2=0;
// _____
// _____Cruce por cero Fase C_____

while (q!=-1)
begin
    nC=nC+1;
    if (yCfundamental[nC]==0)
    begin
        ceroC1=1;
    end
    else
    begin
        ceroC1=0;
    end
    if (yCfundamental[nC+1]==0)

```

```

begin
    ceroC2=1;
end
else
begin
    ceroC2=0;
end

if ((ceroC1 && ceroC2)==1)
begin
    senal_ceroC++;
end
else
begin
    senal_ceroC=0;
end

if (senal_ceroC==64)
begin
    senal_cero_interrupcionC=1;
    senal_ceroC=0;
end
else
begin
    senal_cero_interrupcionC=0;
end

if (senal_cero_interrupcionC==1)
begin
    fC=0;
    break;
end

else
begin

    if (yCfundamental[nC]<0)
    begin
        signoC1=1;
    end
    else
    begin
        signoC1=0;
    end

    if (yCfundamental[nC+1]<0)
    begin
        signoC2=1;
    end
    else
    begin
        signoC2=0;
    end
    if (yCfundamental[nC]>maximoC)
    begin
        maximoC=yCfundamental[nC];
    end
    if (signoC1!=signoC2)
    begin
        nC++;

        for (i=nC;i<nC+Nsample-1;i++)

            begin
                errorC=(fabs(yCfundamental[i]-yCfundamental[i+1]))/maximoC;

                if (errorC>0.05)

```

```

        begin
        fC=0;
        break;
        end
        else
        begin
        fC=1;
        end
        end

        if (fC==0)
        begin
        signoC1=signoC2=0;
        break;
        end
        else
        begin
        N2C=nC+Nsample-1;
        break;
        end
        end
    end
end

signoC1=0;
signoC2=0;
if (fA==0)
begin
    nA=nA+Nsample/2+2;
    nB=nA+Nsample/2+2;
    nC=nA+Nsample/2+2;
end
if (fB==0)
begin
    nB=nB+Nsample/2+2;
    nC=nB+Nsample/2+2;
    nA=nB+Nsample/2+2;
end
if (fC==0)
begin
    nC=nC+Nsample/2+2;
    nA=nC+Nsample/2+2;
    nB=nC+Nsample/2+2;
end
end

if ((fA | fB | fC)==0)
begin
    if (habilitar_hueco==1)
    begin
        tiempo2++;
        tiempo_hueco[a2]=tiempo2*Tm*(Nsample/2);
    end
    if (habilitar_sobretension==1)
    begin
        tiempo4++;
        tiempo_sobretension[a4]=tiempo4*Tm*(Nsample/2);
    end
end
end

if ((fA & fB & fC)==1)
begin
    h=4;
    break;
end
else
begin
    h=2;
end

```

```

end
end

// _____ Tensiones Eficaces de cada fase _____

sumaA=0;
for (i=nA;i<N2A;i++)
begin
    sumaA=sumaA+pow(YA[i],2);

end
    Uef_medio_ciclo_A[k]=sqrt(sumaA/Nsample);
    cruceA=0;

// _____

sumaB=0;
for (i=nB;i<N2B;i++)
begin
    sumaB=sumaB+pow(YB[i],2);
end

    Uef_medio_ciclo_B[k]=sqrt(sumaB/Nsample);
    cruceB=0;

// _____

sumaC=0;
for (i=nC;i<N2C;i++)
begin

    sumaC=sumaC+pow(YC[i],2);
end

    Uef_medio_ciclo_C[k]=sqrt(sumaC/Nsample);
    cruceC=0;

// _____

// _____ Interrupción de tension _____

    if (Uef_medio_ciclo_A[k]>Uef_medio_ciclo_B[k])
    begin
        actual_interrupcion=Uef_medio_ciclo_A[k];
    end
    else
    begin
        actual_interrupcion=Uef_medio_ciclo_B[k];
    end
    if (Uef_medio_ciclo_C[k]>actual_interrupcion)
    begin
        actual_interrupcion=Uef_medio_ciclo_C[k];
    end
    if (actual_interrupcion<umbral_inicio_interrupcion[var_umbral])
    begin
        logica_interrupcion_actual=1;
    end
    else
    begin
        logica_interrupcion_actual=0;
    end
end

    if (Uef_medio_ciclo_A[k-1]>Uef_medio_ciclo_B[k-1])
    begin
        anterior_interrupcion=Uef_medio_ciclo_A[k-1];
    end
    else

```

```

begin
    anterior_interrupcion=Uef_medio_ciclo_B[k-1];
end
if (Uef_medio_ciclo_C[k-1]>anterior_interrupcion)
begin
    anterior_interrupcion=Uef_medio_ciclo_C[k-1];
end
if (anterior_interrupcion>umbral_inicio_interrupcion[var_umbral])
begin
    logica_interrupcion_anterior=1;
end
else
begin
    logica_interrupcion_anterior=0;
end

if ((logica_interrupcion_actual && logica_interrupcion_anterior)==1)
begin
    a1++;
    a1++;
    tiempo1=0;
    habilitar_interrupcion=marcado_interrupcion=proceso_marcado=1;
end

    if (actual_interrupcion>umbral_final_interrupcion[var_umbral])
    begin
        logica_max_interrupcion=1;
    end
    else
    begin
        logica_max_interrupcion=0;
    end

if ((logica_max_interrupcion && habilitar_interrupcion)==1)
begin

    habilitar_interrupcion=0;
    marcado_interrupcion=proceso_marcado=0;
end

    if (habilitar_interrupcion==1)
    begin

        tiempo1=tiempo1+1;
        tiempo_interrupcion[a1]=tiempo1*Tm*(Nsample/2);

    end

// _____ Hueco de tension _____

if (Uef_medio_ciclo_A[k]<Uef_medio_ciclo_B[k])
begin
    actual_hueco=Uef_medio_ciclo_A[k];
    proceso_marcadoA=1;
end

else
begin
    actual_hueco=Uef_medio_ciclo_B[k];
    proceso_marcadoB=1;
end

if (Uef_medio_ciclo_C[k]<actual_hueco)
begin
    actual_hueco=Uef_medio_ciclo_C[k];
    proceso_marcadoC=1;
end

```

```

end
if (actual_hueco<umbral_inicio_hueco[var_umbral])
begin
logica_hueco_actual=1;
end
else
begin
logica_hueco_actual=0;
proceso_marcadoA=0;
proceso_marcadoB=0;
proceso_marcadoC=0;
end

if (Uef_medio_ciclo_A[k-1]<Uef_medio_ciclo_B[k-1])
begin
anterior_hueco=Uef_medio_ciclo_A[k-1];
end
else
begin
anterior_hueco=Uef_medio_ciclo_B[k-1];
end
if (Uef_medio_ciclo_C[k-1]<anterior_hueco)
begin
anterior_hueco=Uef_medio_ciclo_C[k-1];
end
if (anterior_hueco>umbral_inicio_hueco[var_umbral])
begin
logica_hueco_anterior=1;
end
else
begin
logica_hueco_anterior=0;
end

if ((logica_hueco_actual && logica_hueco_anterior)==1)
begin
a2++;
a2++;
tiempo2=0;
muestra_hueco[a22]=0;
habilitar_hueco=1;
Tension_Residual[a2]=actual_hueco;
Profundidad[a2]=Udin-actual_hueco;
marcado_hueco=1;

end

if (actual_hueco>umbral_final_hueco[var_umbral])
begin
logica_max_hueco=1;
end
else
begin
logica_max_hueco=0;
end

if ((logica_max_hueco && habilitar_hueco)==1)
begin
habilitar_hueco=0;
marcado_hueco=0;
end

char faseA1='A',faseA2='o',faseB1='B',faseB2='o',faseC1='C',faseC2='o';
if (habilitar_hueco==1)
begin
if (Uef_medio_ciclo_A[k-3]<umbral_final_hueco[var_umbral])
begin
j++;

```

```

fase_hueco[a2][j]=faseA1;

end
else
begin
j++;
fase_hueco[a2][j]=faseA2;
end
if (Uef_medio_ciclo_B[k-3]<umbral_final_hueco[var_umbral])
begin
j++;
fase_hueco[a2][j]=faseB1;
end
else
begin
j++;
fase_hueco[a2][j]=faseB2;
end
if (Uef_medio_ciclo_C[k-3]<umbral_final_hueco[var_umbral])
begin
j++;
fase_hueco[a2][j]=faseC1;
end
else
begin
j++;
fase_hueco[a2][j]=faseC2;
end
end

j=0;

```

```

tiempo2++;
tiempo_hueco[a2]=tiempo2*Tm*(Nsample/2);
if (actual_hueco!=0)
begin
a22++;
muestra_hueco[a22]=actual_hueco;
end
if (actual_hueco<Tension_Residual[a2])
begin
Tension_Residual[a2]=actual_hueco;
Profundidad[a2]=Udin-actual_hueco;
end
end

```

```

// _____ sobretension temporal _____
if (Uef_medio_ciclo_A[k]>Uef_medio_ciclo_B[k])
begin
actual_sobretension=Uef_medio_ciclo_A[k];
end
else
begin
actual_sobretension=Uef_medio_ciclo_B[k];
end
if (Uef_medio_ciclo_C[k]>actual_sobretension)
begin
actual_sobretension=Uef_medio_ciclo_C[k];
end
if (actual_sobretension>umbral_inicio_sobretension[var_umbral])
begin
logica_sobretension_actual=1;
end
else
begin

```

```

    logica_sobretension_actual=0;
end

if (Uef_medio_ciclo_A[k-1]>Uef_medio_ciclo_B[k-1])
begin
    anterior_sobretension=Uef_medio_ciclo_A[k-1];
end
else
begin
    anterior_sobretension=Uef_medio_ciclo_B[k-1];
end
if (Uef_medio_ciclo_C[k-1]>anterior_sobretension)
begin
    anterior_sobretension=Uef_medio_ciclo_C[k-1];
end
if (anterior_sobretension<umbral_inicio_sobretension[var_umbral])
begin
    logica_sobretension_anterior=1;
end
else
begin
    logica_sobretension_anterior=0;
end

if ((logica_sobretension_actual && logica_sobretension_anterior)==1)
begin
    a44++;
    a4++;
    tiempo4=0;
    muestra_sobretension[a44]=0;
    Amplitud_maxima_sobretension[a4]=actual_sobretension;
    marcado_sobretension=proceso_marcado=habilitar_sobretension=1;
end

if (actual_sobretension<umbral_final_sobretension[var_umbral])
begin
    logica_max_sobretension=1;
end
else
begin
    logica_max_sobretension=0;
end

if ((logica_max_sobretension && habilitar_sobretension)==1)
begin
    habilitar_sobretension=marcado_sobretension=0;
end

if (habilitar_sobretension==1)
begin

    if (Uef_medio_ciclo_A[k-3]>umbral_final_sobretension[var_umbral])
    begin
        j++;
        fase_sobretension[a4][j]=faseA1;
    end
    else
    begin
        j++;
        fase_sobretension[a4][j]=faseA2;
    end
    if (Uef_medio_ciclo_B[k-3]>umbral_final_sobretension[var_umbral])
    begin
        j++;
        fase_sobretension[a4][j]=faseB1;
    end
    else

```

```

begin
    j++;
    fase_sobretension[a4][j]=faseB1;
end
if (Uef_medio_ciclo_C[k-3]>umbral_final_sobretension[var_umbral])
begin
    j++;
    fase_sobretension[a4][j]=faseC1;
end
else
begin
    j++;
    fase_sobretension[a4][j]=faseC1;
end
end
j=0;

a44++;
tiempo4++;
tiempo_sobretension[a4]=tiempo4*Tm*(Nsample/2);
muestra_sobretension[a44]=actual_sobretension;

    if (actual_sobretension>Amplitud_maxima_sobretension[a4])
    begin
        Amplitud_maxima_sobretension[a4]=actual_sobretension;
    end
end

// _____ Deteccion ventana marcada de doce ciclos _____

if (nA>1536)
begin
    var_A++;
    evento_A[var_A]=proceso_marcadoA;
end
if (nB>1536)
begin
    var_B++;
    evento_B[var_B]=proceso_marcadoB;
end
if (nC>1536)
begin
    var_C++;
    evento_C[var_C]=proceso_marcadoC;
end
end

if (var_A==24)
begin
    habilitar_Urd=1;
end
else
begin
    habilitar_Urd=0;
end
end

// _____

if ((habilitar_Urd && umbral_variable)==1)
begin

    resultadoA=0;
    for (i=1;i<var_A;i++)
    begin
        resultadoA=evento_A[i]+resultadoA;
    end

    resultadoB=0;
    for (i=1;i<var_B;i++)

```

```

begin

    resultadoB=evento_B[i]+resultadoB;
end

resultadoC=0;
for (i=1;i<var_C;i++)
begin
    resultadoC=evento_C[i]+resultadoC;
end

    if (resultadoA==0)
begin
    var_umbral++;

    Urd[var_umbral]=0.9967*Urd[var_umbral-1]+0.0033*Uef_12_ciclos_A[var_umbral];
end
    if (resultadoB==0)
begin

    Urd[var_umbral]=0.9967*Urd[var_umbral-1]+0.0033*Uef_12_ciclos_B[var_umbral];
end
    if (resultadoC==0)
begin

    Urd[var_umbral]=0.9967*Urd[var_umbral-1]+0.0033*Uef_12_ciclos_C[var_umbral];
end
    else
begin

    var_umbral++;
    Urd[var_umbral]=Urd[var_umbral-1];

end

    umbral_inicio_interrupcion[var_umbral]=0.1*Urd[var_umbral];
    umbral_final_interrupcion[var_umbral]=0.1*Urd[var_umbral]+histeresis_interrupcion;

    umbral_inicio_hueco[var_umbral]=0.9*Urd[var_umbral];
    umbral_final_hueco[var_umbral]=0.9*Urd[var_umbral]+histeresis_hueco;

    umbral_inicio_sobretension[var_umbral]=1.1*Urd[var_umbral];
    umbral_final_sobretension[var_umbral]=1.1*Urd[var_umbral]-histeresis_sobretension;

    var_A=var_B=var_C=1;
end
k++;
j=0;
end
// _____

printf("_____ Interrupcion de tension _____\n");
    for (j=1;j<a1;j++)
    {
        printf("Duracion del evento %f\n",tiempo_interrupcion[j]);
    }
printf("\n");
printf("_____ Hueco de tension _____\n");
    for (j=1;j<a2;j++)
    {
        printf("Duracion del evento Tension Residual Profundidad Inicio %f %f %f %c %c %c
%s\n",tiempo_hueco[j],Tension_Residual[j],Profundidad[j],fase_hueco[j][1],fase_hueco[j][2],fase_hueco[j][3]);
    }
printf("\n");
printf("_____ Sobretension Temporal _____\n");
    for (j=1;j<=a4;j++)
    {

```

```

                printf("Duracion del evento Amplitud maxima  %f %f %c %c
%c\n",tiempo_sobretension[j],Amplitud_maxima_sobretension[j],fase_sobretension[j][1],fase_sobretension[j][2],fase_sobretension[j][3]);
            }

    getch();
}

```

#### c4) Medidor trifásico implementado con muestra variable.

```

/*Declaración de librerías*/
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<stdlib.h>
#define begin {
#define end }
#include<time.h>

main()
{
//_____Variables para la generación de la señal_____
float Ts=0.0001302083333,Vmax=120*sqrt(2),pi=3.14159265358979,Tm=0.0001302083333,fsis=60;
float YA[13000],YB[13000],YC[13000],yAfundamental[13000],yBfundamental[13000],yCfundamental[13000];
float AmplitudA=6,AmplitudB=150,AmplitudC=8,Udin=120;
int i,j,nmax=12800,Nsample=128,q=1,proceso_marcado;

//_____Variables para la detección de cruces por cero fase A_____
int nA=0,ceroA1,ceroA2,senal_ceroA,signoA1,signoA2,senal_cero_intercepcionA,cruceA=0,tramo1A,tramo2A;
int logica_tramo1A,logica_tramo2A,inicioA[3],fA,h=0,senal_cero_interrupcionA;
float errorA;
//_____Variables para la detección de cruces por cero fase B_____
int nB=0,ceroB1,ceroB2,senal_ceroB,signoB1,signoB2,senal_cero_intercepcionB,cruceB=0,tramo1B,tramo2B;
int logica_tramo1B,logica_tramo2B,inicioB[3],fB,senal_cero_interrupcionB;
float errorB;
//_____Variables para la detección de cruces por cero fase C_____
int nC=0,ceroC1,ceroC2,senal_ceroC,signoC1,signoC2,senal_cero_intercepcionC,cruceC=0,tramo1C,tramo2C;
int logica_tramo1C,logica_tramo2C,inicioC[3],fC,senal_cero_interrupcionC;
float errorC;
//_____Variables para la estimación de la tensión eficaz fase A_____
float sumaA,Uef_medio_ciclo_A[10000],maximoA=0;
int N2A;
//_____Variables para la estimación de la tensión eficaz fase B_____
float sumaB,Uef_medio_ciclo_B[10000],maximoB=0;
int N2B;
//_____Variables para la estimación de la tensión eficaz fase C_____
float sumaC,Uef_medio_ciclo_C[10000],maximoC=0;
int N2C;
//_____Variables para la detección y evaluación de una interrupción de tensión_____
float umbral_inicio_interrupcion[10];
float umbral_final_interrupcion[10];
    umbral_inicio_interrupcion[0]=0.1*Udin;
    #define histeresis_interrupcion (5/100)*Udin;
    umbral_final_interrupcion[0]=0.1*Udin+histeresis_interrupcion;
float actual_interrupcion,anterior_interrupcion,tiempo_interrupcion[1000];
int logica_interrupcion_actual,logica_interrupcion_anterior,a11=0,a1=0,tiempo1,habilitar_interrupcion;
int marcado_interrupcion,logica_max_interrupcion;
//_____Variables para la detección y evaluación de un hueco de tensión_____
float actual_hueco,anterior_hueco,muestra_hueco[1000],Tension_Residual[1000],tiempo_hueco[1000],Profundidad[1000];
int proceso_marcadoA,proceso_marcadoB,proceso_marcadoC,logica_hueco_actual,logica_hueco_anterior,habilitar_hueco;

```

```

int marcado_hueco,logica_max_hueco,tiempo2,a22=0,a2=0;
float umbral_inicio_hueco[10],umbral_final_hueco[10];
char fase_hueco[1000][3];
time_t inicio_hueco[1000], final_hueco[1000];
struct tm *tiempoComienzoPtr_hueco[1000], *tiempoFinalPtr_hueco[1000];

    umbral_inicio_hueco[0]=0.9*Udin;
    #define histeresis_hueco (5/100)*Udin;
    umbral_final_hueco[0]=0.9*Udin+histeresis_hueco;
//___Variables para la detección y evaluación de una Sobretensión Temporal___
float actual_sobretension,anterior_sobretension,muestra_sobretension[1000],tiempo_sobretension[1000];
float Amplitud_maxima_sobretension[1000];
int logica_sobretension_actual,logica_sobretension_anterior,a44=0,a4=0,tiempo4=0,marcado_sobretension;
int habilitar_sobretension,logica_max_sobretension;
float umbral_inicio_sobretension[10],umbral_final_sobretension[10];
char fase_sobretension[1000][3];
#define histeresis_sobretension (5/100)*Udin;
umbral_inicio_sobretension[0]=1.1*Udin;
umbral_final_sobretension[0]=1.1*Udin-histeresis_sobretension;
//___Variables para la detección de la ventana marcada de doce ciclos___
float Urd[1000],Uef_12_ciclos_A[1000],Uef_12_ciclos_B[1000],Uef_12_ciclos_C[1000];
Urd[0]=Udin;
Uef_medio_ciclo_A[0]=Uef_medio_ciclo_B[0]=Uef_medio_ciclo_C[0]=120;
int var_A=0,var_B=0,var_C=0,evento_A[1000],evento_B[1000],evento_C[1000],habilitar_Urd,umbral_variable=0;
int resultadoA,resultadoB,resultadoC,var_umbral=0,muestra_variable=0,k=1;
//_____Generación de la señal_____
//_____Fase A_____

    for (i=0;i<0+12800;i++)
    {
        if (i<2678)
        {
            YA[i]=Vmax*sin(2*pi*fsis*i*Ts);
        }
        if (i>=2678 && i<=6367)
        {
            YA[i]=AmplitudA*sqrt(2)*sin(2*pi*fsis*i*Ts);
        }
        if (i>6367)
        {
            YA[i]=Vmax*sin(2*pi*fsis*i*Ts);
        }
        yAfundamental[i]=YA[i];
    }
//_____Fase B_____

    for (i=0;i<12800;i++)
    {
        if (i<2678)
        {
            YB[i]=Vmax*sin(2*pi*fsis*i*Ts);
        }
        if (i>=2678 && i<=6367)
        {
            YB[i]=AmplitudB*sqrt(2)*sin(2*pi*fsis*i*Ts);
        }
        if (i>6367)
        {
            YB[i]=Vmax*sin(2*pi*fsis*i*Ts);
        }
        yBfundamental[i]=YB[i];
    }
//_____Fase C_____

    for (i=0;i<12800;i++)
    {
        if (i<2678)
        {
            YC[i]=Vmax*sin(2*pi*fsis*i*Ts);
        }
    }

```

```

    }
    if (i>=2678 && i<=6367)
    {
        YC[i]=AmplitudC*sqrt(2)*sin(2*pi*fsis*i*Ts);
    }
    if (i>6367)
    {
        YC[i]=Vmax*sin(2*pi*fsis*i*Ts);
    }
    yCfundamental[i]=YC[i];
}

//
while (nB<nmax)
begin

//
Cruce por cero Fase A
while (q!=-1)
begin
    while (q!=-1)
    begin
        nA=nA+1;
        if (yAfundamental[nA]==0)
        begin
            ceroA1=1;
        end
        else
        begin
            ceroA1=0;
        end
        if (yAfundamental[nA+1]==0)
        begin
            ceroA2=1;
        end
        else
        begin
            ceroA2=0;
        end

        if ((ceroA1 && ceroA2)==1)
        begin
            senal_ceroA++;
        end
        else
        begin
            senal_ceroA=0;
        end

        if (senal_ceroA==64)
        begin
            senal_cero_interrupcionA=1;
            senal_ceroA=0;
        end
        else
        begin
            senal_cero_interrupcionA=0;
        end

        if (senal_cero_interrupcionA==1)
        begin
            if (muestra_variable==0)
            begin
                inicioA[0]=nA;
            end
            else
            begin
                inicioA[0]=nA;
                inicioA[2]=nA+Nsample;
            end
        end
    end
end
end

```

```

end
break;
end
else
begin
if (yAfundamental[nA]<0)
begin
signoA1=1;
end
else
begin
signoA1=0;
end

if (yAfundamental[nA+1]<0)
begin
signoA2=1;
end
else
begin
signoA2=0;
end
if (yAfundamental[nA]>maximoA)
begin
maximoA=yAfundamental[nA];
end
if (signoA1!=signoA2)
begin
inicioA[cruceA]=nA+1;
cruceA=cruceA+1;
end
if (cruceA==3)
begin

tramo1A=inicioA[1]-inicioA[0];
tramo2A=inicioA[2]-inicioA[1];

if (tramo1A>60 & tramo1A<68)
begin
logica_tramo1A=1;
end
else
begin
logica_tramo2A=0;
end

if (tramo2A>60 & tramo2A<68)
begin
logica_tramo2A=1;
end
else
begin
logica_tramo2A=0;
end
//___
if ((logica_tramo1A && logica_tramo2A)==1)
begin
for (i=inicioA[0];i<(inicioA[2]-1);i++)

begin
errorA=(fabs(yAfundamental[i]-yAfundamental[i+1]))/maximoA;

if (errorA>0.05)
begin
fA=0;
break;
end

```

```

        else
        begin
        fA=1;
        end
        end
    end
    else
    begin
    fA=0;
    end

        if (fA==0)
        begin

        cruceA=tramo1A=tramo2A=0;
        signoA1=signoA2=0;
        break;
        end
        else
        begin
        nA=inicioA[0];
        N2A=inicioA[2]-1;
        cruceA=tramo1A=tramo2A=0;
        break;
        end
        //___
    end
end
end
signoA1=0;
signoA2=0;
//_____Cruce por cero Fase B_____

while (q!=-1)
begin
    nB=nB+1;
    if (yBfundamental[nB]==0)
    begin
        ceroB1=1;
    end
    else
    begin
        ceroB1=0;
    end
    if (yBfundamental[nB+1]==0)
    begin
        ceroB2=1;
    end
    else
    begin
        ceroB2=0;
    end

    if ((ceroB1 && ceroB2)==1)
    begin
        senal_ceroB++;
    end
    else
    begin
        senal_ceroB=0;
    end

    if (senal_ceroB==64)
    begin
        senal_cero_interrupcionB=1;
        senal_ceroB=0;
    end
end
end

```

```

else
begin
senal_cero_interrupcionB=0;
end

if (senal_cero_interrupcionB==1)
begin
if (muestra_variable==0)
begin
inicioB[0]=nB;
end
else
begin
inicioB[0]=nB;
inicioB[2]=nB+Nsample;
end
break;
end
else
begin

if (yBfundamental[nB]<0)
begin
signoB1=1;
end
else
begin
signoB1=0;
end

if (yBfundamental[nB+1]<0)
begin
signoB2=1;
end
else
begin
signoB2=0;
end
if (yBfundamental[nB]>maximoB)
begin
maximoB=yBfundamental[nB];
end
if (signoB1!=signoB2)
begin
inicioB[cruceB]=nB+1;
cruceB=cruceB+1;
end

if (cruceB==3)
begin

tramo1B=inicioB[1]-inicioB[0];
tramo2B=inicioB[2]-inicioB[1];

if (tramo1B>60 & tramo1B<68)
begin
logica_tramo1B=1;
end
else
begin
logica_tramo2B=0;
end

if (tramo2B>60 & tramo2B<68)
begin
logica_tramo2B=1;
end

```

```

else
begin
logica_tramo2B=0;
end

if ((logica_tramo1B && logica_tramo2B)==1)
begin

for (i=inicioB[0];i<(inicioB[2]-1);i++)
begin
errorB=(fabs(yBfundamental[i]-yBfundamental[i+1]))/maximoB;

if (errorB>0.05)
begin
fB=0;
break;
end
else
begin
fB=1;
end
end
end

else
begin
fB=0;
end

if (fB==0)
begin

cruceB=tramo1B=tramo2B=0;
signoB1=signoB2=0;
break;
end
else
begin
nB=inicioB[0];
N2B=inicioB[2]-1;
break;
end

//_____
end
end
signoB1=0;
signoB2=0;

//_____

//_____Cruce por cero Fase C_____

while (q!=-1)
begin
nC=nC+1;
if (yCfundamental[nC]==0)
begin
ceroC1=1;
end
else
begin
ceroC1=0;
end
if (yCfundamental[nC+1]==0)
begin
ceroC2=1;
end
end

```

```

else
begin
    ceroC2=0;
end

if ((ceroC1 && ceroC2)==1)
begin
    senal_ceroC++;
end
else
begin
    senal_ceroC=0;
end

if (senal_ceroC==64)
begin
    senal_cero_interrupcionC=1;
    senal_ceroC=0;
end
else
begin
    senal_cero_interrupcionC=0;
end

if (senal_cero_interrupcionC==1)
begin
    if (muestra_variable==0)
    begin
        inicioC[0]=nC;
    end
    else
    begin
        inicioC[0]=nC;
        inicioC[2]=nC+Nsample;
    end
    break;
end

else
begin

    if (yCfundamental[nC]<0)
    begin
        signoC1=1;
    end
    else
    begin
        signoC1=0;
    end

    if (yCfundamental[nC+1]<0)
    begin
        signoC2=1;
    end
    else
    begin
        signoC2=0;
    end
    if (yCfundamental[nC]>maximoC)
    begin
        maximoC=yCfundamental[nC];
    end
    if (signoC1!=signoC2)
    begin
        inicioC[cruceC]=nC+1;
        cruceC=cruceC+1;
    end
end
end

```

```

if (cruceC==3)
begin

    tramo1C=inicioC[1]-inicioC[0];
    tramo2C=inicioC[2]-inicioC[1];

    if (tramo1C>60 & tramo1C<68)
    begin
        logica_tramo1C=1;
    end
    else
    begin
        logica_tramo2C=0;
    end

    if (tramo2C>60 & tramo2C<68)
    begin
        logica_tramo2C=1;
    end
    else
    begin
        logica_tramo2C=0;
    end
    //_____
    if ((logica_tramo1C && logica_tramo2C)==1)
    begin
        for (i=inicioC[0];i<(inicioC[2]-1);i++)
        begin
            errorC=(fabs(yCfundamental[i]-yCfundamental[i+1]))/maximoC;

            if (errorC>0.05)
            begin
                fC=0;
                break;
            end
            else
            begin
                fC=1;
            end
        end
    end

    else
    begin
        fC=0;
    end

    if (fC==0)
    begin
        cruceC=tramo1C=tramo2C=0;
        signoC1=signoC2=0;
    end
    else
    begin
        nC=inicioC[0];
        N2C=inicioC[2]-1;
        break;
    end
    //_____
end

end
end
signoC1=0;
signoC2=0;
if (fA==0)
begin

```

```

        nA=inicioA[2]+2;
        nB=inicioA[2]+2;
        nC=inicioA[2]+2;
    end
    if (fB==0)
    begin
        nB=inicioB[2]+2;
        nC=inicioB[2]+2;
        nA=inicioB[2]+2;
    end
    if (fC==0)
    begin
        nC=inicioC[2]+2;
        nA=inicioC[2]+2;
        nB=inicioC[2]+2;
    end

    if ((fA | fB | fC)==0)
    begin
        if ((habilitar_hueco)==1)
        begin
            tiempo2++;
            tiempo_hueco[a2]=tiempo2*Tm*(Nsample/2);
        end
        if (habilitar_sobretension==1)
        begin
            tiempo4++;
            tiempo_sobretension[a4]=tiempo4*Tm*(Nsample/2);
        end
    end

    if ((fA & fB & fC)==1)
    begin
        h=4;
        break;
    end
    else
    begin
        h=2;
    end
end
//_____Tensiones Eficaces de cada fase_____

sumaA=0;
for (i=inicioA[0];i<inicioA[2]-1;i++)
begin
    sumaA=sumaA+pow(YA[i],2);
end
    Uef_medio_ciclo_A[k]=sqrt(sumaA/(inicioA[2]-inicioA[0]+1));
    cruceA=0;

//_____
sumaB=0;
for (i=inicioB[0];i<inicioB[2]-1;i++)
begin
    sumaB=sumaB+pow(YB[i],2);
end

    Uef_medio_ciclo_B[k]=sqrt(sumaB/(inicioB[2]-inicioB[0]+1));
    cruceB=0;

//_____
sumaC=0;
for (i=inicioC[0];i<inicioC[2]-1;i++)
begin
    sumaC=sumaC+pow(YC[i],2);
end

```

```

end

Uef_medio_ciclo_C[k]=sqrt(sumaC/(inicioC[2]-inicioC[0]+1));
cruceC=0;

// _____
// _____ Interrupción de tensión _____

if (Uef_medio_ciclo_A[k]>Uef_medio_ciclo_B[k])
begin
    actual_interrupcion=Uef_medio_ciclo_A[k];
end
else
begin
    actual_interrupcion=Uef_medio_ciclo_B[k];
end
if (Uef_medio_ciclo_C[k]>actual_interrupcion)
begin
    actual_interrupcion=Uef_medio_ciclo_C[k];
end
if (actual_interrupcion<umbral_inicio_interrupcion[var_umbral])
begin
    logica_interrupcion_actual=1;
end
else
begin
    logica_interrupcion_actual=0;
end

if (Uef_medio_ciclo_A[k-1]>Uef_medio_ciclo_B[k-1])
begin
    anterior_interrupcion=Uef_medio_ciclo_A[k-1];
end
else
begin
    anterior_interrupcion=Uef_medio_ciclo_B[k-1];
end
if (Uef_medio_ciclo_C[k-1]>anterior_interrupcion)
begin
    anterior_interrupcion=Uef_medio_ciclo_C[k-1];
end
if (anterior_interrupcion>umbral_inicio_interrupcion[var_umbral])
begin
    logica_interrupcion_anterior=1;
end
else
begin
    logica_interrupcion_anterior=0;
end

if ((logica_interrupcion_actual && logica_interrupcion_anterior)==1)
begin
    a11++;
    a1++;
    tiempo1=0;
    habilitar_interrupcion=marcado_interrupcion=proceso_marcado=1;
end

if (actual_interrupcion>umbral_final_interrupcion[var_umbral])
begin
    logica_max_interrupcion=1;
end
else
begin
    logica_max_interrupcion=0;
end
end

```

```

if ((logica_max_interrupcion && habilitar_interrupcion)==1)
begin

    habilitar_interrupcion=0;
    marcado_interrupcion=proceso_marcado=0;

end

                                if (habilitar_interrupcion==1)
                                begin

                                    tiempo1=tiempo1+1;
                                    tiempo_interrupcion[a1]=tiempo1*Tm*(Nsample/2);

                                end

// _____ Hueco de tension _____

if (Uef_medio_ciclo_A[k]<Uef_medio_ciclo_B[k])
begin
    actual_hueco=Uef_medio_ciclo_A[k];
    proceso_marcadoA=1;
end

else
begin
    actual_hueco=Uef_medio_ciclo_B[k];
    proceso_marcadoB=1;
end

if (Uef_medio_ciclo_C[k]<actual_hueco)
begin
    actual_hueco=Uef_medio_ciclo_C[k];
    proceso_marcadoC=1;
end
if (actual_hueco<umbral_inicio_hueco[var_umbral])
begin
    logica_hueco_actual=1;
end
else
begin
    logica_hueco_actual=0;
    proceso_marcadoA=0;
    proceso_marcadoB=0;
    proceso_marcadoC=0;
end

if (Uef_medio_ciclo_A[k-1]<Uef_medio_ciclo_B[k-1])
begin
    anterior_hueco=Uef_medio_ciclo_A[k-1];
end
else
begin
    anterior_hueco=Uef_medio_ciclo_B[k-1];
end
if (Uef_medio_ciclo_C[k-1]<anterior_hueco)
begin
    anterior_hueco=Uef_medio_ciclo_C[k-1];
end
if (anterior_hueco>umbral_inicio_hueco[var_umbral])
begin
    logica_hueco_anterior=1;
end
else
begin
    logica_hueco_anterior=0;

```

```

end

if ((logica_hueco_actual && logica_hueco_anterior)==1)
begin
a2++;
a2++;
tiempo2=0;
muestra_hueco[a2]=0;
habilitar_hueco=1;
Tension_Residual[a2]=actual_hueco;
Profundidad[a2]=Udin-actual_hueco;
marcado_hueco=1;

end

if (actual_hueco>umbral_final_hueco[var_umbral])
begin
logica_max_hueco=1;
end
else
begin
logica_max_hueco=0;
end

if ((logica_max_hueco && habilitar_hueco)==1)
begin
habilitar_hueco=0;
marcado_hueco=0;
end
char faseA1='A',faseA2='o',faseB1='B',faseB2='o',faseC1='C',faseC2='o';
if (habilitar_hueco==1)
begin
if (Uef_medio_ciclo_A[k-3]<umbral_final_hueco[var_umbral])
begin
j++;

fase_hueco[a2][j]=faseA1;

end
else
begin
j++;
fase_hueco[a2][j]=faseA2;
end
if (Uef_medio_ciclo_B[k-3]<umbral_final_hueco[var_umbral])
begin
j++;
fase_hueco[a2][j]=faseB1;
end
else
begin
j++;
fase_hueco[a2][j]=faseB2;
end
if (Uef_medio_ciclo_C[k-3]<umbral_final_hueco[var_umbral])
begin
j++;
fase_hueco[a2][j]=faseC1;
end
else
begin
j++;
fase_hueco[a2][j]=faseC2;
end

j=0;

```

```

tiempo2++;
tiempo_hueco[a2]=tiempo2*Tm*(Nsample/2);
if (actual_hueco!=0)
begin
a22++;
muestra_hueco[a22]=actual_hueco;
end
if (actual_hueco<Tension_Residual[a2])
begin
Tension_Residual[a2]=actual_hueco;
Profundidad[a2]=Udin-actual_hueco;
end
end

// _____ sobretension temporal _____
if (Uef_medio_ciclo_A[k]>Uef_medio_ciclo_B[k])
begin
actual_sobretension=Uef_medio_ciclo_A[k];
end
else
begin
actual_sobretension=Uef_medio_ciclo_B[k];
end
if (Uef_medio_ciclo_C[k]>actual_sobretension)
begin
actual_sobretension=Uef_medio_ciclo_C[k];
end
if (actual_sobretension>umbral_inicio_sobretension[var_umbral])
begin
logica_sobretension_actual=1;
end
else
begin
logica_sobretension_actual=0;
end

if (Uef_medio_ciclo_A[k-1]>Uef_medio_ciclo_B[k-1])
begin
anterior_sobretension=Uef_medio_ciclo_A[k-1];
end
else
begin
anterior_sobretension=Uef_medio_ciclo_B[k-1];
end
if (Uef_medio_ciclo_C[k-1]>anterior_sobretension)
begin
anterior_sobretension=Uef_medio_ciclo_C[k-1];
end
if (anterior_sobretension<umbral_inicio_sobretension[var_umbral])
begin
logica_sobretension_anterior=1;
end
else
begin
logica_sobretension_anterior=0;
end

if ((logica_sobretension_actual && logica_sobretension_anterior)==1)
begin
a44++;
a4++;
tiempo4=0;
muestra_sobretension[a44]=0;
Amplitud_maxima_sobretension[a4]=actual_sobretension;
marcado_sobretension=proceso_marcado=habilitar_sobretension=1;
end

```

```

        if (actual_sobretension<umbral_final_sobretension[var_umbral])
        begin
            logica_max_sobretension=1;
        end
        else
        begin
            logica_max_sobretension=0;
        end

        if ((logica_max_sobretension && habilitar_sobretension)==1)
        begin
            habilitar_sobretension=marcado_sobretension=0;
        end

        if (habilitar_sobretension==1)
        begin

            if (Uef_medio_ciclo_A[k-3]>umbral_final_sobretension[var_umbral])
            begin
                j++;
                fase_sobretension[a4][j]=faseA1;
            end
            else
            begin
                j++;
                fase_sobretension[a4][j]=faseA2;
            end
            if (Uef_medio_ciclo_B[k-3]>umbral_final_sobretension[var_umbral])
            begin
                j++;
                fase_sobretension[a4][j]=faseB1;
            end
            else
            begin
                j++;
                fase_sobretension[a4][j]=faseB1;
            end
            if (Uef_medio_ciclo_C[k-3]>umbral_final_sobretension[var_umbral])
            begin
                j++;
                fase_sobretension[a4][j]=faseC1;
            end
            else
            begin
                j++;
                fase_sobretension[a4][j]=faseC1;
            end
            end
            j=0;

            a44++;
            tiempo4++;
            tiempo_sobretension[a4]=tiempo4*Tm*(Nsample/2);
            muestra_sobretension[a44]=actual_sobretension;

            if (actual_sobretension>Amplitud_maxima_sobretension[a4])
            begin
                Amplitud_maxima_sobretension[a4]=actual_sobretension;
            end

        end

// _____ Deteccion ventana marcada de doce ciclos _____

        if (nA>1536)
        begin

```

```

    var_A++;
    evento_A[var_A]=proceso_marcadoA;
end
if (nB>1536)
begin
    var_B++;
    evento_B[var_B]=proceso_marcadoB;
end
if (nC>1536)
begin
    var_C++;
    evento_C[var_C]=proceso_marcadoC;
end

if (var_A==24)
begin
    habilitar_Urd=1;
end
else
begin
    habilitar_Urd=0;
end
end

```

//

```

if ((habilitar_Urd && umbral_variable)==1)
begin

    resultadoA=0;
    for (i=1;i<var_A;i++)
    begin
        resultadoA=evento_A[i]+resultadoA;
    end

    resultadoB=0;
    for (i=1;i<var_B;i++)
    begin

        resultadoB=evento_B[i]+resultadoB;
    end

    resultadoC=0;
    for (i=1;i<var_C;i++)
    begin
        resultadoC=evento_C[i]+resultadoC;
    end

    if (resultadoA==0)
    begin
        var_umbral++;

        Urd[var_umbral]=0.9967*Urd[var_umbral-1]+0.0033*Uef_12_ciclos_A[var_umbral];
    end
    if (resultadoB==0)
    begin

        Urd[var_umbral]=0.9967*Urd[var_umbral-1]+0.0033*Uef_12_ciclos_B[var_umbral];
    end
    if (resultadoC==0)
    begin

        Urd[var_umbral]=0.9967*Urd[var_umbral-1]+0.0033*Uef_12_ciclos_C[var_umbral];
    end
    else
    begin

        var_umbral++;
    end
end

```

```

        Urd[var_umbral]=Urd[var_umbral-1];

        end

        umbral_inicio_interrupcion[var_umbral]=0.1*Urd[var_umbral];
        umbral_final_interrupcion[var_umbral]=0.1*Urd[var_umbral]+histeresis_interrupcion;

        umbral_inicio_hueco[var_umbral]=0.9*Urd[var_umbral];
        umbral_final_hueco[var_umbral]=0.9*Urd[var_umbral]+histeresis_hueco;

        umbral_inicio_sobretension[var_umbral]=1.1*Urd[var_umbral];
        umbral_final_sobretension[var_umbral]=1.1*Urd[var_umbral]-histeresis_sobretension;

        var_A=var_B=var_C=1;
        end
        k++;
        j=0;
    end
//
printf("_____ Interrupcion de tension _____\n");
        for (j=1;j<a1;j++)
        {
            printf("Duracion del evento %f\n",tiempo_interrupcion[j]);
        }
printf("\n");
printf("_____ Hueco de tension _____\n");
        for (j=1;j<a2;j++)
        {
            printf("Duracion del evento Tension Residual Profundidad Inicio %f %f %f %c %c %c
%s\n",tiempo_hueco[j],Tension_Residual[j],Profundidad[j],fase_hueco[j][1],fase_hueco[j][2],fase_hueco[j][3]);
        }
printf("\n");
printf("_____ Sobretension Temporal _____\n");
        for (j=1;j<=a4;j++)
        {
            printf("Duracion del evento Amplitud maxima %f %f %c %c
%c\n",tiempo_sobretension[j],Amplitud_maxima_sobretension[j],fase_sobretension[j][1],fase_sobretension[j][2],fase_sobre
nsion[j][3]);
        }
        getch();
    }
}

```

## ANEXO D1. ALGORITMO PARA LA MEDICIÓN DE LAS PERTURBACIONES TRANSITORIAS DE TENSIÓN EN MATLAB.

d1) Medidor monofásico implementado con muestra fija.

```
% DISEÑO DEL PROTOTIPO DEL MÓDULO PARA LA MONITORIZACIÓN DE LAS PERTURBACIONES
% DE TENSIÓN PARA UNA UNIDAD CONSTRUCTIVA EN EL ANÁLISIS DE LOS PARÁMETROS
% DE CALIDAD DE POTENCIA ELÉCTRICA CON BASE EN EL ESTÁNDAR IEC 61000-4-30
```

```
% ALGORITMO PARA LA MEDICIÓN DE LAS PERTURBACIONES DE TENSIÓN DE ESTADO
% TRANSITORIO PARA UNA RED MONOFÁSICA
% HUECOS DE TENSIÓN, INTERRUPCIONES DE TENSIÓN Y SOBRETENSIONES TEMPORALES.
% AUTORES:
% ANDRÉS MAURICIO SALINAS CALA COD 2033708 ESTUDIANTE INGENIERÍA ELECTRÓNICA
% OSCAR JAVIER CORREDOR MONTERO COD 2020801 ESTUDIANTE INGENIERIA ELÉCTRICA
```

```
% _____ ENTRADAS _____
```

```
% Definiciones.
% "Tensión eficaz refrescada cada semiperiodo(Uef_medio_ciclo): Valor de la tensión
% eficaz medida en un periodo, comenzando en un paso por cero de la fundamental,
% y refrescada cada semiperiodo."[IEC 61000-4-30:2003.]
```

```
% -yfundamental: La componente fundamental de la señal de la red es
% utilizada para detectar el cruce por cero al inicio del periodo según lo
% establecido en el Estándar IEC 61000-4-30.
% Para la medición con muestra constante, el algoritmo detecta el cruce por
% cero al comienzo del periodo y la muestra es almacenada en la variable n.
```

```
% -yn: es la señal de la red, con armónicos, interarmónicos, flicker, etc.
```

```
% -Nsample: muestras por ciclo de la ventana de observación, en el caso de
% las pruebas realizadas este fue de 128 muestras/ciclo.
```

```
% -umbral_inicio_interrupcion(1): umbral donde comienza la interrupción de tensión.
% -umbral_inicio_hueco(1): umbral donde comienza el hueco de tensión.
% -umbral_inicio_sobretension(1): umbral donde comienza la sobretensión temporal.
% -umbral_final_interrupcion(1): umbral donde finaliza la interrupción de tensión.
% -umbral_final_hueco(1): umbral donde finaliza el hueco de tensión.
% -umbral_final_sobretension(1): umbral donde finaliza la sobretensión temporal.
```

```
% -interrupciones_CREG: Esta entrada lepermite al usuario habilitar los
% umbrales establecidos por la CREG en la resolución 024 de 2005.
```

```
% -Udin (Tensión de entrada declarada): tensión declarada por el usuario u
% operador de red para tomarla como referencia en la declaración de la
% histeresis de los eventos.
```

```
% -Urd (Tensión de referencia deslizante): tensión que varía cada doce ciclos,
% La tensión Urd actual está en función de la tensión Urd anterior más la
% tensión eficaz de doce ciclos.
```

```
% histeresis_interrupcion: porcentaje de udin, usado para determinar el
% umbral final de la interrupción.
```



```

a2=0;
a22=0;
tiempo2=0;
tiempo_hueco=0;
inicio_hueco=0;
inicio_hueco_fecha=0;
inicio_hueco_hora=0;
final_hueco=0;
final_hueco_fecha=0;
final_hueco_hora=0;
muestra_hueco=0;
Tension_Residual=0;
histeresis_hueco=(5/100)*Udin;
umbral_inicio_hueco(1)=0.9*Udin;
umbral_final_hueco(1)=0.9*Udin+histeresis_hueco;
Profundidad=0;
habilitar_hueco=0;
%
a4=0;
a44=0;
tiempo4=0;
tiempo_sobretension=0;
muestra_sobretension=0;
inicio_sobretension=0;
inicio_sobretension_fecha=0;
inicio_sobretension_hora=0;
final_sobretension=0;
final_sobretension_fecha=0;
final_sobretension_hora=0;
Amplitud_maxima_sobretension=0;
histeresis_sobretension=(5/100)*Udin;
umbral_inicio_sobretension(1)=1.1*Udin;
umbral_final_sobretension(1)=1.1*Udin-histeresis_sobretension;
habilitar_sobretension=0;
%
a1=0;
a11=0;
tiempo1=0;
habilitar_interrupcion=0;
tiempo_interrupcion=0;
muestra_interrupcion=0;
inicio_interrupcion=0;
inicio_interrupcion_fecha=0;
inicio_interrupcion_hora=0;
final_interrupcion=0;
final_interrupcion_fecha=0;
final_interrupcion_hora=0;
histeresis_interrupcion=(5/100)*Udin;
umbral_inicio_interrupcion(1)=0.1*Udin;
umbral_final_interrupcion(1)=0.1*Udin+histeresis_interrupcion;

%Si el usuario desea almacenar las interrupciones basados en las
%resoluciones de la CREG, entonces la variable Interrupciones CREG será
%igual a uno si no será igual a cero y toma los valores definidos de forma
%particular.

interrupciones_CREG=0;

if interrupciones_CREG==1,
    histeresis_interrupcion=0;
    umbral_inicio_interrupcion(1)=0.1*Udin;
    umbral_final_interrupcion(1)=0.1*Udin+histeresis_interrupcion;
end

%
Nsample=128;      %Muestras por ciclo
Tm=Tm3;          %Periodo de muestreo

```

```

umbral_variable=1;
maximo=0;
%
%|
%| INICIO DEL ALGORITMO
%|
% A continuación se presenta el algoritmo para la medición de eventos
% transitorios con base en el estándar IEC 61000-4-30:2003 y las
% resoluciones CREG 024 de 2005 y 016 de 2007. Esta se estructura de la
% siguiente manera:
% 1. Detección de los cruces por cero.
% 2. Evaluación de la tensión eficaz
% 3. Evaluación de la interrupción de Tensión
% 4. Evaluación del Hueco de Tensión
% 5. Evaluación de la Sobretensión Temporal.
% 6. Cambio de la referencia del umbral de detección.

n=1;
q=1;
Uef_medio_ciclo(4)=127;
k=5;
t=cputime;
while n<(numel(yfundamental)-4*Nsample),
%
% | EVALUACIÓN DEL VRMS REFRESCADO CADA MEDIO CICLO
%
% Rutina para la detección de Cruces por cero.

    while q~=-1,
        n=n+1;
% Inicialmente se verifica la existencia de una interrupcion total de
% tensión, se cuenta la cantidad de ceros consecutivos durante un periodo
% de la señal, si el resultado es igual a Nsample/2 implica que durante medio periodo no se
% presenta señal por lo no se puede detectar un cruce por cero, por tanto se
% procede a refrescar la señal medio periodo ya que este es el tiempo minimo
% para encontrar otro cruce por cero.

        if yfundamental(n)==0,
            cero1=1;
        else
            cero1=0;
        end
        if yfundamental(n+1)==0,
            cero2=1;
        else
            cero2=0;
        end
        if (cero1 & cero2)==1,
            senal_cero=senal_cero+1;
        else
            senal_cero=0;
        end
        end

        if senal_cero==Nsample/2,
            senal_cero_interrupcion=1;
            senal_cero=0;
            break
        else
            senal_cero_interrupcion=0;
        end

    end
%
% como se expresó anteriormente si ocurre una interrupcion total, se

```

```

% procede a refrescar en medio periodo el proceso de medición.
if senal_cero_interrupcion==1,

    n=n+Nsample/2;
    N2=n+Nsample-1;

    break

else

% Si no ocurre una interrupción total se procede a detectar los cruce por
% cero inicial y final durante el periodo de evaluación. Las variables
% signo1 y signo2 contiene el signo de las muestra actual y posterior de la
% componente fundamental de la señal de la red. El cruce por cero es
% almacenado en n.
    signo1=sign(yfundamental(n));
    signo2=sign(yfundamental(n+1));
    if yfundamental(n)>maximo,
        maximo=yfundamental(n);
    end
    if signo1~=signo2
        n=n+1;
% Después de detectar el cruce por cero se procede a verificar si ocurrió
% una cambio brusco en la amplitud,comparando muestras consecutivas desde n
% hasta n+Nsample-1.
        for i=n:Nsample-1,
            error=abs(yfundamental(i)-yfundamental(i+1))/maximo;

            if error>10,

                f=0;
                break
            else
                f=1;
            end

        end
        maximo=0;
% Si la señal no es la ideal para evaluar la medida se procede a buscar
% el siguiente cruce por cero.
        if f==0,
            n=n+Nsample+2;
            signo1=0;
            signo2=0;
            if (habilitar_hueco)==1,
                tiempo2=tiempo2+1;
                tiempo_hueco(a2)=tiempo2*Tm*(Nsample/2);
            end
            if habilitar_sobretension==1
                tiempo4=tiempo4+1;
                tiempo_sobretension(a4)=tiempo4*Tm*(Nsample/2);
            end
        else
% Si la señal es ideal para la detección de la tensión eficaz, se sale de
% la rutina de detección de cruces por cero.

            N2=n+Nsample-1;
            signo1=0;
            signo2=0;
            break
        end
    end

end

end

signo1=0;

```

```

signo2=0;
%
% Al encontrar el cruce por cero se procede a determinar la Uef_medio_ciclo
% (Tensión eficaz refrescada cada semiperiodo). Inicialmente se elevan las
% muestras al cuadrado, seguidamente se suman. El resultado es multiplicado
% por el inverso de la cantidad de muestras contenidas en el periodo de
% evaluación.

suma=0;
if (senal_cero_interrupcion)==1,
    Uef_medio_ciclo(k)=0;
else
    for i = n:N2,          %Se elevan las Nsample muestras de la señal original
                          %al cuadrado
        suma=yn(i).^2+suma;
    end
    Uef_medio_ciclo(k)=sqrt((1/abs(N2-n+1))*suma);
end

```

```

% El proceso de detección y evaluación de los eventos es similar para todos
% ellos. La primera parte detecta el inicio del evento a través de una
% lógica de comparación de las muestras actual y anterior con el umbral inicial
% del evento.
% La segunda parte detecta el final del evento con una lógica de
% comparación de la muestra actual con el umbral final del evento.
% Si el evento está en curso, el algoritmo evaluará el evento de acuerdo a
% lo establecido en el estandar IEC 61000-4-30:2003.
% Al comenzar un evento transitorio éste afecta las medidas para eventos
% estacionarios, por lo que se asigna la variable proceso_marcado para indicar
% que las medidas en estado estacionario son dudosas. Al finalizar el
% evento se procede a deshabilitar la variable proceso_marcado.
%

```

---

```

%                               EVALUACIÓN DE LA INTERRUPCIÓN
%

```

---

```

% Una interrupción de tensión comienza cuando la tensión eficaz cae por debajo
% del umbral inicial de la interrupción y finaliza cuando la tensión eficaz
% se eleva por encima del umbral final de la interrupción. Para este evento
% solo se determina la duración ya que según el estándar IEC 61000-4-30 una
% interrupción de tensión es un caso especial de un hueco por tanto la
% evaluación de la amplitud queda registrada en el evento hueco de tensión.
%

```

---

```

% En esta lógica se determinan las condiciones para detectar el inicio
% del evento, se evalúa la muestra actual y anterior para conocer si
% sobrepasaron los umbrales de tensión

```

```

if Uef_medio_ciclo(k)<umbral_inicio_interrupcion(var_umbral),
    logica_interrupcion_actual=1;
else
    logica_interrupcion_actual=0;
end

if Uef_medio_ciclo(k-1)>umbral_inicio_interrupcion(var_umbral),
    logica_interrupcion_anterior=1;
else
    logica_interrupcion_anterior=0;
end

```

```

% Si la muestra actual es menor que el umbral inicial de la interrupción y
% la muestra anterior es mayor que el umbral inicial de la interrupción
% comienza el evento. se realiza la inicialización de variables,
% registro de la fecha y hora del inicio del evento. Habilitación del
% proceso de marcado y evaluación de la medida.

```

```

if (logica_interrupcion_actual & logica_interrupcion_anterior)==1,
    a11=a11+1;
    a1=a1+1;
    inicio_interrupcion_fecha(a1)=floor(now);
    inicio_interrupcion_hora(a1)=rem(now,1);
    tiempo1=2;
    muestra_interrupcion(a11)=0.1*Udin+0.1;
    habilitar_interrupcion=1;
    proceso_marcado=1;
end

```

%

%En esta sección se determina si Uef\_medio\_ciclo está por encima del umbral de finalización de la interrupción. Si esto es verdadero, se procede a finalizar la evaluación del evento, se registra la fecha y hora además de inactivar el proceso de marcado y evaluación de la medida.

```

if Uef_medio_ciclo(k)>umbral_final_interrupcion(var_umbral),
    logica_max_interrupcion=1;
else
    logica_max_interrupcion=0;
end

```

```

if (logica_max_interrupcion & habilitar_interrupcion)==1,

```

```

    final_interrupcion_fecha(a1)=floor(now);
    final_interrupcion_hora(a1)=rem(now,1);
    habilitar_interrupcion=0;
    proceso_marcado=0;
end

```

%

% mientras Uef\_medio\_ciclo esté por debajo del umbral final de interrupción habilitar\_interrupcion será 1 y procederá a contabilizar la duración, la variable tiempo 1 contabiliza los semiciclos transcurridos durante el evento, en vista que la interrupción quedará almacenada en el evento hueco, no se necesita almacenar las muestras de la interrupción en esta el evento interrupción.

```

if (habilitar_interrupcion)==1,
    tiempo1=tiempo1+1;
    tiempo_interrupcion(a1)=tiempo1*Tm*((N2-n+1)/2);
    if Uef_medio_ciclo(k)~=0,
        a11=a11+1;
        muestra_interrupcion(a11)=Uef_medio_ciclo(k);
    end
end

```

%

% **EVALUACIÓN DEL HUECO DE TENSIÓN**

% Un hueco de tensión comienza cuando la tensión eficaz cae por debajo del umbral inicial del hueco de tensión y finaliza cuando la tensión eficaz se eleva por encima del umbral final del hueco. Según el estándar IEC 61000-4-30:2003 la evaluación de este evento esta determinada por la duración del evento y por la tensión residual y la profundidad. La profundidad es la diferencia entre la tensión de entrada declarada (Udin) y la mínima tensión medida durante el evento y la tensión residual es la mínima tensión medida durante el evento.

%

% Determinar el inicio del evento a través de la lógica de detección.

```

if Uef_medio_ciclo(k)<umbral_inicio_hueco(var_umbral),
    logica_hueco_actual=1;
else

```

```

        logica_hueco_actual=0;
    end

    if Uef_medio_ciclo(k-1)>umbral_inicio_hueco(var_umbral),
        logica_hueco_anterior=1;
    else
        logica_hueco_anterior=0;
    end

% Si la muestra actual está debajo del umbral inicial del hueco de tensión y la
% muestra anterior está por encima del umbral inicial del hueco de tensión
% entonces comienza el evento hueco de tensión. Se inicializan las variables
% del evento y se registra la fecha y hora en que comienza el mismo.
    if (logica_hueco_actual & logica_hueco_anterior)==1,
        a22=a22+1;
        a2=a2+1;
        inicio_hueco_fecha(a2)=floor(now);
        inicio_hueco_hora(a2)=rem(now,1);
        tiempo2=1;
        muestra_hueco(a22)=0;
        habilitar_hueco=1;
        Tension_Residual(a2)=Uef_medio_ciclo(k);
        Profundidad(a2)=Udin-Uef_medio_ciclo(k);
        proceso_marcado=1;
    end

%


---


%Lógica para determinar si la muestra actual esta por debajo del umbral
%final del hueco de tensión
    if Uef_medio_ciclo(k)>umbral_final_hueco(var_umbral),
        logica_max_hueco=1;
    else
        logica_max_hueco=0;
    end

%Si la tensión eficaz actual es mayor que el umbral final de hueco de tensión entonces
%se finaliza el hueco de tensión. El proceso de marcado y habilitación de
%la evaluación de la medida se desactiva. Se registra la fecha y hora de
%finalización del evento.
    if (logica_max_hueco & habilitar_hueco)==1,

        final_hueco_fecha(a2)=floor(now);
        final_hueco_hora(a2)=rem(now,1);
        habilitar_hueco=0;
        proceso_marcado=0;
    end

%


---


% si la tensión eficaz actual está por debajo del umbral final del hueco de
% tensión, aumenta la duración del evento, se actualiza la Tension_Residual,
% y Profundidad del evento. Además se almacenan las muestras del evento en
% muestra_hueco. La Profundidad esta referida a la tensión de entrada
% declarada o la tensión de referencia deslizando según sea el caso.
    if (habilitar_hueco)==1,
        tiempo2=tiempo2+1;
        tiempo_hueco(a2)=tiempo2*Tm*((N2-n+1)/2);
        if Uef_medio_ciclo(k)-=0;
            a22=a22+1;
            muestra_hueco(a22)=Uef_medio_ciclo(k);
        end

    if umbral_variable==0,

        if Uef_medio_ciclo(k)<Tension_Residual(a2),
            Tension_Residual(a2)=Uef_medio_ciclo(k);
            Profundidad(a2)=Udin-Uef_medio_ciclo(k);

```

```

        end
    else

        if Uef_medio_ciclo(k)<Tension_Residual(a2),
            Tension_Residual(a2)=Uef_medio_ciclo(k);
            Profundidad(a2)=Urd(var_umbral)-Uef_medio_ciclo(k);
            Urd_hueco(a2)=Urd(var_umbral);
        end
    end

end

%


---


%


---


% EVALUACIÓN DE LA SOBRETENSIÓN TEMPORAL
%


---


% La sobretensión temporal comienza cuando la tensión eficaz está por encima
% del umbral inicial de la sobretensión temporal y finaliza cuando la tensión
% eficaz esta por debajo del umbral final de la sobretensión temporal. El estándar
% IEC 61000-4-30:2003 evalúa la medida por medio de la amplitud máxima de la
% tensión eficaz durante el evento y la duración del mismo.
%


---



%Lógica para la detección del inicio del evento. Si la tensión eficaz
%actual esta por encima del umbral inicial de la sobretensión temporal y la
%tensión anterior está por debajo del umbral inicial de la sobretensión
%temporal entonces comienza el evento. Se inicializan las variables, el
%proceso de marcado de los datos estacionarios y se registra la fecha y
%hora de inicio del evento

    if Uef_medio_ciclo(k)>umbral_inicio_sobretension(var_umbral),
        logica_sobretension_actual=1;
    else
        logica_sobretension_actual=0;
    end

    if Uef_medio_ciclo(k-1)<umbral_inicio_sobretension(var_umbral),
        logica_sobretension_anterior=1;
    else
        logica_sobretension_anterior=0;
    end

    if (logica_sobretension_actual & logica_sobretension_anterior)==1,
        a44=a44+1;
        a4=a4+1;
        inicio_sobretension_fecha(a4)=floor(now);
        inicio_sobretension_hora(a4)=rem(now,1);
        tiempo4=0;
        muestra_sobretension(a44)=0;
        habilitar_sobretension=1;
        Amplitud_maxima_sobretension(a4)=Uef_medio_ciclo(k);
        proceso_marcado=1;
    end

%


---


% Lógica para la detección del final del evento. si la tensión eficaz
% actual está por debajo del umbral final de la sobretensión temporal se
% procede a finalizar el evento y al registro de la fecha y hora
    if Uef_medio_ciclo(k)<umbral_final_sobretension(var_umbral),
        logica_max_sobretension=1;
    else
        logica_max_sobretension=0;
    end
end

```

```

        if (logica_max_sobretension & habilitar_sobretension)==1,

            final_sobretension_fecha(a4)=floor(now);
            final_sobretension_hora(a4)=rem(now,1);

            habilitar_sobretension=0;
            proceso_marcado=0;
        end

%
% Si la tensión eficaz se encuentra por debajo del umbral final de la
% sobretensión temporal se actualiza la amplitud máxima de la sobretensión
% temporal y la duración del evento. Las muestras del evento son almacenadas
% en el vector muestra_sobretensión
        if (habilitar_sobretension)==1,
            a44=a44+1;
            tiempo4=tiempo4+1;
            tiempo_sobretension(a4)=tiempo4*Tm*((N2-n+1)/2);
            muestra_sobretension(a44)=Uef_medio_ciclo(k);

            if Uef_medio_ciclo(k)>Amplitud_maxima_sobretension(a4),
                Amplitud_maxima_sobretension(a4)=Uef_medio_ciclo(k);
            end
        end

%
% Si ocurrió un hueco de tensión automáticamente marca la ventana de 12
% ciclo y la tensión Urd no se actualiza. Esto se consigue con la
% variable Habilitar_Urd.
        k=k+1;
        if n>1536,
            var=var+1;
            evento(var)=proceso_marcado;
        end

        if var==25,
            habilitar_Urd=1;
        else
            habilitar_Urd=0;
        end

% Habilitar la tensión de referencia deslizante Urd
% Al principio del archivo se encuentra la variable umbral_variable, esta se
% utiliza para que el umbral esté en función de Udin o Urd.
% Si umbral_variable esta en 1, se habilita la tensión de referencia
% deslizante, si es cero los umbrales están en función de la tensión de
% entrada declarada.
        if (habilitar_Urd & umbral_variable)==1,
            resultado=0;
            for i = 1:24,
                resultado=evento(i)+resultado;
            end

            if (resultado)==0,
                var_umbral=var_umbral+1;
                %Evaluación de la tensión Urd definido
                %en el estándar IEC 61000-4-30:2003
                Urd(var_umbral)=0.9967*Urd(var_umbral-1)+0.0033*Uef_12_cycle(var_umbral);
            else
                % si existe una ventana de doce ciclos
                % que este marcada, la tensión de
                % referencia actual es igual a la tensión
                % de referencia anterior.
                var_umbral=var_umbral+1;
                Urd(var_umbral)=Urd(var_umbral-1);
            end
        end
    end
end

```

```

end
%Se actualizan los umbrales en función de la tensión de
%referencia deslizante que resulte del proceso
%anterior.

umbral_inicio_interrupcion(var_umbral)=0.1*Urd(var_umbral);
umbral_final_interrupcion(var_umbral)=0.1*Urd(var_umbral)+histeresis_interrupcion;

umbral_inicio_hueco(var_umbral)=0.9*Urd(var_umbral);
umbral_final_hueco(var_umbral)=0.9*Urd(var_umbral)+histeresis_hueco;

umbral_inicio_sobretension(var_umbral)=1.1*Urd(var_umbral);
umbral_final_sobretension(var_umbral)=1.1*Urd(var_umbral)-histeresis_sobretension;

var=1;
end
%----- Eliminar datos -----
%
%En esta sección se aumenta la variable de almacenamiento
%de las tensiones eficaces, después se pregunta por el valor de "k" con el fin de
%eliminar tensiones previas evitar el almacenamiento excesivo de datos..
if k==10
    Uef_medio_ciclo(1:5)=[];
    k=5;
end
%-----
end

% si el equipo termina de operar y en ese momento ocurría un evento se
% registra la fecha y hora del mismo para evitar problemas con las
% dimensiones de los vectores.
if habilitar_interrupcion==1
    final_interrupcion_fecha(a1)=floor(now);
    final_interrupcion_hora(a1)=rem(now,1);
end

if habilitar_hueco==1
    final_hueco_fecha(a2)=floor(now);
    final_hueco_hora(a2)=rem(now,1);
end

if habilitar_sobretension==1
    final_sobretension_fecha(a4)=floor(now);
    final_sobretension_hora(a4)=rem(now,1);
end

tiempo_proceso=cputime-t

%-----
%
% FINAL DEL ALGORITMO
%
%-----

%En esta sección se concatenan todas las variables de salida para que
%aparezcan de forma ordenada en la ventana de comandos de matlab
disp('____MEDICIÓN DE LA TENSIÓN EFICAZ CON MUESTRA VARIABLE_____')
disp('EVALUACIÓN SOBRETENSIÓN')
disp('Amplitud Maxima-Duracion- inicio del evento- final del evento')
disp(' Año Mes Día Hora Minuto Segundo - Año Mes Día Hora Minuto Segundo')
inicio_sobretension_fecha1=datevec(inicio_sobretension_fecha);
inicio_sobretension_hora1=datevec(inicio_sobretension_hora);
fecha_hora_inicio_sobretension=inicio_sobretension_fecha1+inicio_sobretension_hora1;
final_sobretension_fecha1=datevec(final_sobretension_fecha);
final_sobretension_hora1=datevec(final_sobretension_hora);
fecha_hora_final_sobretension=final_sobretension_fecha1+final_sobretension_hora1;
tiempo_sobretension1=tiempo_sobretension';
sobretension1=Amplitud_maxima_sobretension';

```

```

inicio_final_sobretension=cat(2,sobretension1,tiempo_sobretension1,fecha_hora_inicio_sobretension,fecha_hora_final_sobr
etension);
disp(inicio_final_sobretension)

disp('_____')
disp('EVALUACIÓN INTERRUPCIÓN')
disp('Duracion- inicio del evento- final del evento')
disp(' Año Mes Día Hora Minuto Segundo - Año Mes Día Hora Minuto Segundo' )

tiempo_interruccion1=tiempo_interruccion';
final_interruccion_fecha1=datevec(final_interruccion_fecha);
final_interruccion_hora1=datevec(final_interruccion_hora);
fecha_hora_final_interruccion=final_interruccion_fecha1+final_interruccion_hora1;

inicio_interruccion_fecha1=datevec(inicio_interruccion_fecha);
inicio_interruccion_hora1=datevec(inicio_interruccion_hora);
fecha_hora_inicio_interruccion=inicio_interruccion_fecha1+inicio_interruccion_hora1;

inicio_final_interruccion=cat(2,tiempo_interruccion1,fecha_hora_inicio_interruccion,fecha_hora_final_interruccion);
disp(inicio_final_interruccion)

disp('_____')
%Almacenamiento de interrupciones según la resolución CREG 024 de 2005
tiempo_interruccion_minutos=tiempo_interruccion/60;
h=1;
frecuencia_minutos=0;
interruccion_minutos=0;
for i=1:length(tiempo_interruccion_minutos);

    if tiempo_interruccion_minutos(i)>1

        interrupcion_minutos(h)=tiempo_interruccion_minutos(i);
        frecuencia_minutos=frecuencia_minutos+1;
        h=h+1;
    end
end

%Almacenamiento de las interrupciones superiores a un minuto y con tensión
%menor al 10% de la tensión nominal.
interruccion_minutos %Tiempo de duracion de cada evento superior a un minuto.
frecuencia_minutos %Cantidad de eventos ocurridos en el proceso de medición.

% Ahora se procede a determinar los DES y FES propuestos es la resolución
% CREG 025 de 1999 mientras el equipo esté en funcionamiento.

interruccion_hora=0;
tiempo_interruccion_hora=tiempo_interruccion/3600;

for i=1:length(tiempo_interruccion_hora)

    interrupcion_hora=tiempo_interruccion_hora(i)+interruccion_hora;
end

%Sumatoria del número de veces que el servicio es interrumpido en un
%circuito, mientras esté conectada la unidad constructiva a la red
%eléctrica
FES=a1

%Sumatoria del tiempo en horas de las interrupciones del servicio en un
%circuito, mientras esté conectada la unidad constructiva a la red
%eléctrica.
DES=interruccion_hora

disp('_____')
disp('EVALUACIÓN HUECOS')

```

```

disp('Tensión residual-Profundidad-Duración- inicio del evento- final del evento')
disp(' Año Mes Día Hora Minuto Segundo - Año Mes Día Hora Minuto Segundo')
inicio_hueco_fecha1 = datevec(inicio_hueco_fecha);
inicio_hueco_hora1 = datevec(inicio_hueco_hora);
fecha_hora_inicio_hueco=inicio_hueco_fecha1+inicio_hueco_hora1;
final_hueco_fecha1 = datevec(final_hueco_fecha);
final_hueco_hora1 = datevec(final_hueco_hora);
fecha_hora_final_hueco=final_hueco_fecha1+final_hueco_hora1;
visualizar_hueco=Tension_Residual';
visualizar_Profundidad=Profundidad';
tiempo_hueco1=tiempo_hueco';
inicio_final_hueco=cat(2,visualizar_hueco,visualizar_Profundidad,tiempo_hueco1,fecha_hora_inicio_hueco,fecha_hora_final_hueco);
disp(inicio_final_hueco)

disp('_____')

n=1:1:nfinal;
subplot(3,1,1)
plot(n,yn,'g')
title('Señal Monofásica Discreta ')
xlabel('muestras')
ylabel('amplitud')
axis([-10 length(yn)+10 min(yn)-10 max(yn)+10])

subplot(3,1,2)
n4=1:length(muestra_sobretension);
plot(n4,muestra_sobretension,'b*')
title('Sobretensión Temporal')
xlabel('muestras')
ylabel('amplitud [V]')
axis([0 length(muestra_sobretension)+1 0 max(muestra_sobretension)+30])

subplot(3,1,3)
n2=1:length(muestra_hueco);
plot(n2,muestra_hueco,'b*')
title('Hueco de Tensión')
xlabel('muestras')
ylabel('amplitud [V]')
axis([0 length(muestra_hueco)+1 0 max(umbral_final_hueco)+30])

```

## d2) Medidor monofásico implementado con muestra variable

```

% DISEÑO DEL PROTOTIPO DEL MÓDULO PARA LA MONITORIZACIÓN DE LAS PERTURBACIONES
% DE TENSIÓN PARA UNA UNIDAD CONSTRUCTIVA EN EL ANÁLISIS DE LOS PARÁMETROS
% DE CALIDAD DE POTENCIA ELÉCTRICA CON BASE EN EL ESTÁNDAR IEC 61000-4-30

```

```

% ALGORITMO PARA LA MEDICIÓN DE LAS PERTURBACIONES DE TENSIÓN DE ESTADO
% TRANSITORIO PARA UNA RED MONOFÁSICA
% HUECOS DE TENSIÓN, INTERRUPCIONES DE TENSIÓN Y SOBRETENSIONES TEMPORALES.
% AUTORES:
% ANDRÉS MAURICIO SALINAS CALA COD 2033708 ESTUDIANTE INGENIERÍA ELECTRÓNICA
% OSCAR JAVIER CORREDOR MONTERO COD 2020801 ESTUDIANTE INGENIERIA ELÉCTRICA

```

```

% _____ ENTRADAS _____

```

```

% Definiciones.
% "Tensión eficaz refrescada cada semiperiodo(Uef_medio_ciclo): Valor de la tensión
% eficaz medida en un periodo, comenzando en un paso por cero de la fundamnetal,
% y refrescada cada semiperiodo." [IEC 61000-4-30:2003.]

```

```

% -yfundamental: La componente fundamental de la señal de la red es
% utilizada para detectar el cruce por cero al inicio del periodo según lo

```

% establecido en el Estándar IEC 61000-4-30.  
 % Medición con muestra variable. El algoritmo detecta los  
 % cruces por cero al inicio y al final del periodo, el cruce por cero al  
 % inicio es almacenado en n y el cruce por cero al final del periodo se  
 % almacena en la variable N2.

% -yn: es la señal de la red, con armónicos, interarmónicos, flicker, etc.

% -Nsample: muestras por ciclo de la ventana de observación, en el caso de  
 % las pruebas realizadas este fue de 128 muestras/ciclo.

% -umbral\_variable: Los umbrales pueden ser un porcentaje de  
 % Udin o Urd por tanto se deja una variable para que el usuario decida que  
 % referencia utilizar. Si el usuario decide utilizar el umbral constante,  
 % el algoritmo utilizará los umbrales almacenados en la variable  
 % umbral\_inicio\_evento(1) y umbral\_final\_evento(1).

% -umbral\_inicio\_interrupcion(1): umbral donde comienza la interrupción de tensión.  
 % -umbral\_inicio\_hueco(1): umbral donde comienza el hueco de tensión.  
 % -umbral\_inicio\_sobretension(1): umbral donde comienza la sobretensión temporal.  
 % -umbral\_final\_interrupcion(1): umbral donde finaliza la interrupción de tensión.  
 % -umbral\_final\_hueco(1): umbral donde finaliza el hueco de tensión.  
 % -umbral\_final\_sobretension(1): umbral donde finaliza la sobretensión temporal.

% -Udin (Tensión de entrada declarada): tensión declarada por el usuario u  
 % operador de red para tomarla como referencia en la declaración de la  
 % histeresis y umbrales de los eventos.

% -Urd (Tensión de referencia deslizante): tensión que varía cada doce ciclos,  
 % La tensión Urd actual está en función de la tensión Urd anterior más la  
 % tensión eficaz de doce ciclos.

% histeresis\_interrupcion: porcentaje de udin, usado para determinar el  
 % umbral final de la interrupción.

% histeresis\_hueco: porcentaje de udin, usado para determinar el  
 % umbral final del hueco.

% histeresis\_sobretension: porcentaje de udin, usado para determinar el  
 % umbral final de la sobretensión temporal.

% -Tm: Tiempo de muestreo del convertidor. Esta variable está en función de  
 % la frecuencia medida. La duración de la medida está en función de los  
 % semiciclos medidos, y estos a su vez se multiplican por el tiempo de  
 % muestreo medido, de allí la importancia de su precisión.

% \_\_\_\_\_ SALIDAS \_\_\_\_\_

% inicio\_hueco\_fecha /  
 % inicio\_hueco\_hora /  
 % final\_hueco\_fecha / Datos solicitados por la CREG  
 % final\_hueco\_hora /

% tiempo\_hueco /  
 % Tension\_Residual / Datos solicitados por el estándar  
 % Profundidad /  
 % muestra\_hueco

% inicio\_interrupcion\_fecha /  
 % inicio\_interrupcion\_hora /  
 % final\_interrupcion\_fecha / Datos solicitados por la CREG  
 % final\_interrupcion\_hora /  
 % DES y FES  
 % Interrupciones superiores a un minuto/

% tiempo\_interrupcion / Datos solicitados por el estándar

```

% inicio_sobretension_fecha /
% inicio_sobretension_hora /
% final_sobretension_fecha / Datos solicitados por la CREG
% final_sobretension_hora /

% Amplitud_maxima_sobretension /
% tiempo_sobretension / Datos solicitados por el estándar
% muestra_sobretension
% _____ INICIALIZACIÓN DE VARIABLES _____
clc
close all

cuadrado=0;
suma=0;
udin=127/sqrt(2);
Urd(1)=udin;
signo1=0;
signo2=0;
k=5;
var_umbral=1;
var=1;
inicio=0;
tramo1=0;
tramo2=0;
proceso_marcado=0;
senal_cero_interrupcion=0;
habilitar_interrupcion=0;
habilitar_hueco=0;
habilitar_sobretension=0;
% _____
umbral_inicio_interrupcion=0;
umbral_final_interrupcion=0;
umbral_inicio_hueco=0;
umbral_final_hueco=0;
umbral_inicio_sobretension=0;
umbral_final_sobretension=0;
% _____
a2=0;
a22=0;
tiempo2=0;
tiempo_hueco=0;
inicio_hueco=0;
inicio_hueco_fecha=0;
inicio_hueco_hora=0;
final_hueco=0;
final_hueco_fecha=0;
final_hueco_hora=0;
muestra_hueco=0;
Tension_Residual=0;
histeresis_hueco=(5/100)*udin;
umbral_inicio_hueco(1)=0.9*udin;
umbral_final_hueco(1)=0.9*udin+histeresis_hueco;
Profundidad=0;
enable_hueco=0;
% _____
a4=0;
a44=0;
tiempo4=0;
tiempo_sobretension=0;
muestra_sobretension=0;
inicio_sobretension=0;
inicio_sobretension_fecha=0;
inicio_sobretension_hora=0;
final_sobretension=0;
final_sobretension_fecha=0;
final_sobretension_hora=0;
Amplitud_maxima_sobretension=0;

```

```

histeresis_sobretension=(5/100)*udin;
umbral_inicio_sobretension(1)=1.1*udin;
umbral_final_sobretension(1)=1.1*udin-histeresis_sobretension;
enable_sobretension=0;
%
a1=0;
a11=0;
tiempo1=0;
enable_interrupcion=0;
tiempo_interrupcion=0;
muestra_interrupcion=0;
inicio_interrupcion=0;
inicio_interrupcion_fecha=0;
inicio_interrupcion_hora=0;
final_interrupcion=0;
final_interrupcion_fecha=0;
final_interrupcion_hora=0;
histeresis_interrupcion=(5/100)*udin;
umbral_inicio_interrupcion(1)=0.1*udin;
umbral_final_interrupcion(1)=0.1*udin+histeresis_interrupcion;
%
Nsample=128;           %Muestras por ciclo
Tm=Tm3;               %Periodo de muestreo
umbral_variable=0;
%

%|
%|                     INICIO DEL ALGORITMO
%|

% A continuación se presenta el algoritmo para la medición de eventos
% transitorios con base en el estándar IEC 61000-4-30:2003 y las
% resoluciones CREG 024 de 2005 y 016 de 2007. Este se estructura de la
% siguiente manera:
% 1. Detección de los cruces por cero.
% 2. Evaluación de la tensión eficaz
% 3. Evaluación de la interrupción de Tensión
% 4. Evaluación del Hueco de Tensión
% 5. Evaluación de la Sobretensión Temporal.
% 6. Cambio de la referencia del umbral de detección.

n=1;
cruce=0;
maximo=0;
q=1;
t=cputime;
while n<(numel(yfundamental)-3*Nsample),
%
% |                     EVALUACIÓN DEL VRMS REFRESCADO CADA MEDIO CICLO
%

% Rutina para la detección de Cruces por cero.

    while q~-1,
        n=n+1;

% Inicialmente se verifica la existencia de una interrupcion total de
% tensión, se cuenta la cantidad de ceros consecutivos durante un periodo
% de la señal, si el resultado es 64 implica que durante medio periodo no se
% presenta señal por lo no se puede detectar un cruce por cero, por tanto se
% procede a refrescar la señal medio periodo ya que este es el tiempo minimo
% para encontrar otro cruce por cero.
        for i=n:n+Nsample,

            if yfundamental(i)==0,
                cero1=1;

```

```

else
    cero1=0;
end
if yfundamental(i+1)==0,
    cero2=1;
else
    cero2=0;
end
if (cero1 & cero2)==1,
    senal_cero=senal_cero+1;
else
    senal_cero=0;
end

    if senal_cero==64,
        senal_cero_interrupcion=1;
        senal_cero=0;
        break
    else
        senal_cero_interrupcion=0;
    end

end
%
% como se expresó anteriormente si ocurre una interrupción total, se
% procede a refrescar en medio periodo el proceso de medición.
if senal_cero_interrupcion==1,

    n=n+64;
    N2=n+Nsample-1;

    break

else

% Si no ocurre una interrupción total se procede a detectar los cruces por
% cero inicial y final durante el periodo de evaluación. Las variables
% signo1 y signo2 contiene el signo de las muestra actual y posterior de la
% componente fundamental de la señal de la red. Los cruces por cero son
% almacenados en la inicio(). La variable cruce lleva el conteo de la
% cantidad de cruces almacenado.
    signo1=sign(yfundamental(n));
    signo2=sign(yfundamental(n+1));
    if yfundamental(n)>maximo,
        maximo=yfundamental(n);
    %
    %     else
    %         maximo=yfundamental(n+1);
    end

    if signo1~=signo2
        cruce=cruce+1;
        inicio(cruce)=n+1;
    end

% cruce igual a tres indica que se tiene los cruces por cero de la señal
% fundamental. Ahora se procede a verificar si la señal es óptima para la
% evaluación de la tensión eficaz.
    if cruce==3,
maximo

% se determina la distancia entre el primero y segundo cruce y el segundo
% y tercero para verificar si ocurrió un cambio de fase.
        tramo1=inicio(2)-inicio(1);
        tramo2=inicio(3)-inicio(2);

        if tramo1>=60 && tramo1<=68
            logica_tramo1=1;
        else

```

```

logica_tramo2=0;
end

if tramo2>=60 && tramo2<=68
logica_tramo2=1;
else
logica_tramo2=0;
end

if (logica_tramo1 & logica_tramo2)==1
% Si la señal es optima en cuanto a las distancias entre los cruces por cero
% se procede a revisar las distancias entre muestras para determinar
% si ocurrió un cambio brusco en la tensión.

for i=inicio(1):inicio(3)-1,
error=abs(yfundamental(i)-yfundamental(i+1))/maximo;

if error>0.05,
error;
f=1;
break
else
f=0;
end

end
maximo=0;
else
f=1;

end

% Si la señal no es la ideal para evaluar la medida se procede a buscar
% tres nuevos cruces por cero.
if f==1
n=inicio(3)+2;
cruce=0;
inicio=0;
tramo1=0;
tramo2=0;
signo1=0;
signo2=0;
else
% Si la señal es la ideal se almacena el inicio y final del periodo y se
% finaliza esta rutina

n=inicio(1);
N2=inicio(3)-1;
cruce=0;
tramo1=0;
tramo2=0;
inicio=0;
break
end

end

end

end

signo1=0;
signo2=0;

%
% Al encontrar el cruce por cero se procede a determinar la Uef_medio_ciclo
% (Tensión eficaz refrescada cada medio ciclo). Inicialmente se elevan las
% muestras al cuadrado, seguidamente se suman. El resultado es multiplicado
% por el inverso de la cantidad de muestras contenidas en el periodo de
% evaluación.

```

```

suma=0;
if senal_cero_interrupcion==1,
    Uef_medio_ciclo(k)=0;
else
    for i = n:N2,          %Se elevan las Nsample muestras de la señal original
                            %al cuadrado
        suma=yn(i).^2+suma;
    end
    Uef_medio_ciclo(k)=sqrt((1/abs(N2-n+1))*suma);
end

```

% El proceso de detección y evaluación de los eventos es similar para todos  
% ellos. La primera parte detecta el inicio del evento a través de una  
% lógica de comparación de las muestras actual y anterior con el umbral inicial  
% del evento.  
% La segunda parte detecta el final del evento con una lógica de  
% comparación de la muestra actual con el umbral final del evento.  
% Si el evento está en curso, el algoritmo evaluará el evento de acuerdo a  
% lo establecido en el estandar IEC 61000-4-30:2003.  
%

---

% **EVALUACIÓN DE LA INTERRUPCIÓN**

---

% Una interrupción de tensión comienza cuando la tensión eficaz cae por debajo  
% del umbral inicial de la interrupción y finaliza cuando la tensión eficaz  
% se eleva por encima del umbral final de la interrupción. Para este evento  
% solo se determina la duración ya que según el estándar IEC 61000-4-30 una  
% interrupción de tensión es un caso especial de un hueco por tanto la  
% evaluación de la amplitud queda registrada en el evento hueco de tensión.  
%

```

% En esta lógica se determinan las condiciones para detectar el inicio
% del evento, se evalúa la muestra actual y anterior para conocer si
% sobrepasaron los umbrales de tensión
if Uef_medio_ciclo(k)<umbral_inicio_interrupcion(var_umbral),
    logica_interrupcion_actual=1;
else
    logica_interrupcion_actual=0;
end

if Uef_medio_ciclo(k-1)>umbral_inicio_interrupcion(var_umbral),
    logica_interrupcion_anterior=1;
else
    logica_interrupcion_anterior=0;
end

```

```

% Si la muestra actual es menor que el umbral inicial de la interrupción y
% la muestra anterior es mayor que el umbral inicial de la interrupción
% comienza el evento. se realiza la inicialización de variables,
% registro de la fecha y hora del inicio del evento. Habilitación del
% proceso de marcado y evaluación de la medida.
if (logica_interrupcion_actual & logica_interrupcion_anterior)==1,
    a1=a1+1;
    a1=a1+1;
    inicio_interrupcion_fecha(a1)=floor(now);
    inicio_interrupcion_hora(a1)=rem(now,1);
    tiempo1=2;
    muestra_interrupcion(a1)=0.1*udin+0.1;
    habilitar_interrupcion=1;
    proceso_marcado=1;
end

```

---

% En esta sección se determina si Uef\_medio\_ciclo está por encima del  
% umbral de finalización de la interrupción. Si esto es verdadero, se procede





```

else
    logica_sobretension_actual=0;
end

if Uef_medio_ciclo(k-1)<umbral_inicio_sobretension(var_umbral),
    logica_sobretension_anterior=1;
else
    logica_sobretension_anterior=0;
end

% Almacenamiento del inicio del evento.
if (logica_sobretension_actual & logica_sobretension_anterior)==1,
    a44=a44+1;
    a4=a4+1;
    inicio_sobretension_fecha(a4)=floor(now);
    inicio_sobretension_hora(a4)=rem(now,1);
    tiempo4=-1;
    muestra_hueco(a22)=0;
    muestra_sobretension(a44)=0;
    habilitar_sobretension=1;
    Amplitud_maxima_sobretension(a4)=Uef_medio_ciclo(k);
    proceso_marcado=1;
end

% -----
% Lógica para la detección del final del evento
if Uef_medio_ciclo(k)<umbral_final_sobretension(var_umbral),
    logica_max_sobretension=1;
else
    logica_max_sobretension=0;
end

% Detección del final del evento
% Si la tensión actual es menor que el umbral final de sobretensión temporal
% entonces se finaliza el evento
if (logica_max_sobretension & habilitar_sobretension)==1,
    final_sobretension(a4)=n+(3/2)*Nsample-1;
    final_sobretension_fecha(a4)=floor(now);
    final_sobretension_hora(a4)=rem(now,1);

    habilitar_sobretension=0;
    proceso_marcado=0;
end

% -----
if habilitar_sobretension==1,
    a44=a44+1;
    tiempo4=tiempo4+1;
    tiempo_sobretension(a4)=tiempo4*Tm*((n-N2)/2);
    muestra_sobretension(a44)=Uef_medio_ciclo(k);

    if Uef_medio_ciclo(k)>Amplitud_maxima_sobretension(a4),
        Amplitud_maxima_sobretension(a4)=Uef_medio_ciclo(k);
    end
end

% -----
% Si ocurrió un hueco de tensión automáticamente marca la ventana de 12
% ciclo y la tensión Urd no se actualiza. Esto se consigue con la
% variable Habilitar_Urd.
k=k+1;
if n>1536,
    var=var+1;
    evento(var)=proceso_marcado;
end

if var==24,
    habilitar_Urd=1;
end

```

```

else
    habilitar_Urd=0;
end
%-----Habilitar la tensión de referencia deslizante Urd-----
%Al principio del archivo se encuentra la variable umbral_variable, esta se
%utiliza para que el umbral esté en función de Udin o Urd.
% Si umbral_variable esta en 1, se habilita la tensión de referencia
% deslizante, si es cero los umbrales están en función de udin.

if (habilitar_Urd & umbral_variable)==1,
    resultado=0;
    for i = 1:24,
        resultado=evento(i)+resultado;
    end

    if (resultado)==0,
        var_umbral=var_umbral+1;
        %Evaluación de la tensión Urd con base
        %en la definición del estándar
        Urd(var_umbral)=0.9967*Urd(var_umbral-1)+0.0033*Uef_12_cycle(var_umbral);

    else
        % Si umbral variable esta deshabilitado
        % entonces no se actualiza la tensión
        % Urd.
        var_umbral=var_umbral+1;
        Urd(var_umbral)=Urd(var_umbral-1);

    end

    umbral_inicio_interrupcion(var_umbral)=0.1*Urd(var_umbral);
    umbral_final_interrupcion(var_umbral)=0.1*Urd(var_umbral)+histeresis_interrupcion;

    umbral_inicio_hueco(var_umbral)=0.9*Urd(var_umbral);
    umbral_final_hueco(var_umbral)=0.9*Urd(var_umbral)+histeresis_hueco;

    umbral_inicio_sobretension(var_umbral)=1.1*Urd(var_umbral);
    umbral_final_sobretension(var_umbral)=1.1*Urd(var_umbral)-histeresis_sobretension;

    var=1;
end
%----- Eliminar datos -----
%-----
%En esta sección se aumenta la variable de almacenamiento
%de las muestras después se pregunta por el valor de "k" con el fin de
%eliminar las muestras anteriores y evitar el almacenamiento excesivo de
%las muestras.
if k==10
    Uef_medio_ciclo(1:5)=[];
    k=5;
end
%-----
end
if habilitar_interrupcion==1
    final_interrupcion_fecha(a1)=floor(now);
    final_interrupcion_hora(a1)=rem(now,1);
end

if habilitar_hueco==1
    final_hueco_fecha(a2)=floor(now);
    final_hueco_hora(a2)=rem(now,1);
end

if habilitar_sobretension==1
    final_sobretension_fecha(a4)=floor(now);
    final_sobretension_hora(a4)=rem(now,1);
end
end

```

```
tiempo_proceso=cputime-t
```

```
%-----  
%                               FINAL DEL  
%                               ALGORITMO  
%-----
```

```
%En esta sección se concatenan todas las variables de salida para que  
%aparezcan de forma ordenada en la ventana de comandos de matlab
```

```
disp('_____MEDICIÓN DE LA TENSIÓN EFICAZ CON MUESTRA VARIABLE_____')  
disp('EVALUACIÓN SOBRETENSIÓN')  
disp('Amplitud Maxima-Duracion-      inicio del evento-      final del evento')  
disp('      Año Mes Día Hora Minuto Segundo - Año Mes Día Hora Minuto Segundo' )
```

```
inicio_sobretension_fecha1=datevec(inicio_sobretension_fecha);  
inicio_sobretension_hora1=datevec(inicio_sobretension_hora);  
fecha_hora_inicio_sobretension=inicio_sobretension_fecha1+inicio_sobretension_hora1;  
final_sobretension_fecha1=datevec(final_sobretension_fecha);  
final_sobretension_hora1=datevec(final_sobretension_hora);  
fecha_hora_final_sobretension=final_sobretension_fecha1+final_sobretension_hora1;  
tiempo_sobretension1=tiempo_sobretension';  
sobretension1=Amplitud_maxima_sobretension';  
inicio_final_sobretension=cat(2,sobretension1,tiempo_sobretension1,fecha_hora_inicio_sobretension,fecha_hora_final_sobr  
etension);  
disp(inicio_final_sobretension)
```

```
disp('_____')  
disp('EVALUACIÓN INTERRUPCIÓN')  
disp('Duracion-      inicio del evento-      final del evento')  
disp('      Año Mes Día Hora Minuto Segundo - Año Mes Día Hora Minuto Segundo' )
```

```
tiempo_interrupcion1=tiempo_interrupcion';  
final_interrupcion_fecha1=datevec(final_interrupcion_fecha);  
final_interrupcion_hora1=datevec(final_interrupcion_hora);  
fecha_hora_final_interrupcion=final_interrupcion_fecha1+final_interrupcion_hora1;
```

```
inicio_interrupcion_fecha1=datevec(inicio_interrupcion_fecha);  
inicio_interrupcion_hora1=datevec(inicio_interrupcion_hora);  
fecha_hora_inicio_interrupcion=inicio_interrupcion_fecha1+inicio_interrupcion_hora1;
```

```
inicio_final_interrupcion=cat(2,tiempo_interrupcion1,fecha_hora_inicio_interrupcion,fecha_hora_final_interrupcion);  
disp(inicio_final_interrupcion)
```

```
%Almacenamiento de interrupciones según la resolución CREG 024 de 2005
```

```
tiempo_interrupcion_minutos=tiempo_interrupcion/60;
```

```
h=1;
```

```
frecuencia_minutos=0;
```

```
interrupcion_minutos_tiempo=0;
```

```
for i=1:length(tiempo_interrupcion_minutos);
```

```
    if tiempo_interrupcion_minutos(i)>1
```

```
        interrupcion_minutos_tiempo(h)=tiempo_interrupcion_minutos(i);
```

```
        frecuencia_minutos=frecuencia_minutos+1;
```

```
        h=h+1;
```

```
    end
```

```
end
```

```
%Almacenamiento de las interrupciones superiores a un minuto y con tensión
```

```
%menor al 10% de la tensión nominal.
```

```
interrupcion_minutos_tiempo %Tiempo de duracion de cada evento superior a un minuto.
```

```
frecuencia_minutos %Cantidad de eventos ocurridos en el proceso de medición.
```

```
% Ahora se procede a determinar los DES y FES propuestos es la resolución
```

```

% CREG 025 de 1999.

interrupcion_hora=0;
tiempo_interrupcion_horas=tiempo_interrupcion/3600;

for i=1:length(tiempo_interrupcion_horas)

    interrupcion_hora=tiempo_interrupcion_horas(i)+interrupcion_hora;
end

%"Sumatoria del número de veces que el servicio es interrumpido en un
%circuito, mientras esté conectada la unidad constructiva a la red
%eléctrica"[CREG 024, 1999].
FES=a1

%"Sumatoria del tiempo en horas ed las interrupciones del servicio en un
%circuito, mientras esté conectada la unidad constructiva a la red
%eléctrica"[CREG 024, 1999].
DES=interrupcion_hora

disp('_____')
disp('EVALUACIÓN HUECOS')
disp('Tensión residual-Profundidad-Duración- inicio del evento- final del evento')
disp(' Año Mes Día Hora Minuto Segundo - Año Mes Día Hora Minuto Segundo')
inicio_hueco_fecha1 = datevec(inicio_hueco_fecha);
inicio_hueco_hora1 = datevec(inicio_hueco_hora);
fecha_hora_inicio_hueco=inicio_hueco_fecha1+inicio_hueco_hora1;
final_hueco_fecha1 = datevec(final_hueco_fecha);
final_hueco_hora1 = datevec(final_hueco_hora);
fecha_hora_final_hueco=final_hueco_fecha1+final_hueco_hora1;
visualizar_hueco=Tension_Residual';
visualizar_Profundidad=Profundidad';
tiempo_hueco1=tiempo_hueco;
inicio_final_hueco=cat(2,visualizar_hueco,visualizar_Profundidad,tiempo_hueco1,fecha_hora_inicio_hueco,fecha_hora_final_hueco);
disp(inicio_final_hueco)

% error1=minimo_hueco-Tension_Residual

disp('_____')

% n=1:1:nfinal;
% figure,subplot(2,3,[1 2])
% plot(n,yn,'g')
% title('Señal Monofásica Discreta ')
% xlabel('muestras')
% ylabel('amplitud')
% axis([-10 length(yn)+10 min(yn)-10 max(yn)+10])
%
% subplot(2,3,3)
% plot(n,yfundamental,'g')
% title('Señal fundamental Discreta ')
% xlabel('muestras')
% ylabel('amplitud')
% axis([-10 length(yn)+10 min(yn)-10 max(yn)+10])

% figure,subplot(1,3,1)
% n2=1:length(muestra_hueco);
% plot(n2,muestra_hueco,'b*')
% title('Hueco de Tensión')
% xlabel('muestras')
% ylabel('amplitud [V]')
% axis([0 length(muestra_hueco)+1 0 max(muestra_hueco)+30])
%

```

```

% subplot(1,3,2)
% n4=1:length(muestra_sobretension);
% plot(n4,muestra_sobretension,'b*')
% title('Sobretensión Temporal')
% xlabel('muestras')
% ylabel('amplitud [V]')
% axis([0 length(muestra_sobretension)+1 0 max(muestra_sobretension)+30])
%
%
% subplot(1,3,3)
% n1=1:length(muestra_interrupcion);
% plot(n1,muestra_interrupcion,'b*')
% title('Interrupción de Tensión')
% xlabel('muestras')
% ylabel('amplitud [V]')
% axis([0 length(muestra_interrupcion)+1 0 max(muestra_interrupcion)+7])

n2=1:length(muestra_hueco);
figure,plot(n2,muestra_hueco,'b*',n2,umbral_inicio_hueco,'g*',n2,umbral_final_hueco,'g*')
title('Hueco de Tensión')
xlabel('muestras')
ylabel('amplitud [V]')
axis([0 length(muestra_hueco)+1 0 max(umbral_final_hueco)+30])

% n1=1:length(muestra_interrupcion);
% figure,plot(n1,muestra_interrupcion,'b*',n2,umbral_inicio_interrupcion,'g*',n2,umbral_final_interrupcion,'g*')
% title('Interrupción de Tensión')
% xlabel('muestras')
% ylabel('amplitud [V]')
% axis([0 length(muestra_interrupcion)+1 0 max(umbral_final_interrupcion)+7])

n=1:1:nfinal;
figure,plot(n,yn) % discreto
title('Señal de la red discretizada')
xlabel('muestras')
ylabel('amplitud [V]')
axis([0 length(yn)+1 -max(yn)-5 max(yn)+5])
zoom on

% t=0:Tfinal/1000:Tfinal;
% figure,plot(t,yfundamental)

```

### d3) Medidor trifásico implementado con muestra fija.

% DISEÑO DEL PROTOTIPO DEL MÓDULO PARA LA MONITORIZACIÓN DE LAS PERTURBACIONES  
% DE TENSIÓN PARA UNA UNIDAD CONSTRUCTIVA EN EL ANÁLISIS DE LOS PARÁMETROS  
% DE CALIDAD DE POTENCIA ELÉCTRICA CON BASE EN EL ESTÁNDAR IEC 61000-4-30

% ALGORITMO PARA LA MEDICIÓN DE LAS PERTURBACIONES DE TENSIÓN DE ESTADO  
% TRANSITORIO PARA UNA RED TRIFÁSICA  
% HUECOS DE TENSIÓN, INTERRUPCIONES DE TENSIÓN Y SOBRETENSIONES TEMPORALES.  
% AUTORES:  
% ANDRÉS MAURICIO SALINAS CALA COD 2033708 ESTUDIANTE INGENIERÍA ELECTRÓNICA  
% OSCAR JAVIER CORREDOR MONTERO COD 2020801 ESTUDIANTE INGENIERIA ELÉCTRICA

% \_\_\_\_\_ ENTRADAS \_\_\_\_\_

% Definiciones.

% "Tensión eficaz refrescada cada semiperiodo(Uef\_medio\_ciclo): Valor de la tensión  
% eficaz medida en un periodo, comenzando en un paso por cero de la fundamental,  
% y refrescada cada semiperiodo." [IEC 61000-4-30:2003.]

% -yfundamentalA: La componente fundamental de la fase A de la red es  
% utilizada para detectar el cruce por cero al inicio del periodo según lo  
% establecido en el Estándar IEC 61000-4-30.  
% -yfundamentalB: La componente fundamental de la fase B de la red es  
% utilizada para detectar el cruce por cero al inicio del periodo según lo  
% establecido en el Estándar IEC 61000-4-30.  
% -yfundamentalC: La componente fundamental de la fase C de la red es  
% utilizada para detectar el cruce por cero al inicio del periodo según lo  
% establecido en el Estándar IEC 61000-4-30.

% Para la medicion con muestra constante, el algoritmo detecta el cruce por  
% cero al comienzo del periodo y la muestra es almacenada en la variable n.

% -yA: Señal de la fase A de la red eléctrica, con armónicos, interarmónicos, flicker, etc.  
% -yB: Señal de la fase B de la red eléctrica, con armónicos, interarmónicos, flicker, etc.  
% -yC: Señal de la fase C de la red eléctrica, con armónicos, interarmónicos, flicker, etc.

% -Nsample: muestras por ciclo de la ventana de observación, en el caso de  
% las pruebas realizadas este fue de 128 muestras/ciclo.

% -umbral\_inicio\_interrupcion(1): umbral donde comienza la interrupción de tensión.  
% -umbral\_inicio\_hueco(1): umbral donde comienza el hueco de tensión.  
% -umbral\_inicio\_sobretension(1): umbral donde comienza la sobretensión temporal.  
% -umbral\_final\_interrupcion(1): umbral donde finaliza la interrupción de tensión.  
% -umbral\_final\_hueco(1): umbral donde finaliza el hueco de tensión.  
% -umbral\_final\_sobretension(1): umbral donde finaliza la sobretensión temporal.

% -interrupciones\_CREG: Esta entrada lepermite al usuario habilitar los  
% umbrales establecidos por la CREG en la resolución 024 de 2005.

% -Udin (Tensión de entrada declarada): tension declarada por el usuario u  
% operador de red para tomarla como referencia en la declaración de la  
% histeresis de los eventos.

% -Urd (Tensión de referencia deslizante): tension que varia cada doce ciclos,  
% La tension Urd actual esta en función de la tensión Urd anterior mas la  
% tensión eficaz de doce ciclos.

% histeresis\_interrupcion: porcentaje de udin, usado para determinar el  
% umbral final de la interrupción.

% histeresis\_hueco: porcentaje de udin, usado para determinar el  
% umbral final del hueco.

```

% histeresis_sobretension:porcentaje de udin, usado para determinar el
% umbral final de la sobretensión temporal.

% -umbral_variable: debido a que los umbrales pueden ser un porcentaje de
% Udin o Urd, se deja una variable para que el usuario decida que
% referencia utilizar.

% -Tm: Tiempo de muestreo del convertor. Esta variable está en funcion de
% la frecuencia medida. La duración de la medida esta en funcion de los
% semiciclos medidos, estos a su vez se multiplicam por el tiempo de
% muestreo medido, de allí la importancia de su precisión.

% _____ SALIDAS _____
% inicio_hueco_fecha /
% inicio_hueco_hora /
% final_hueco_fecha / Datos solicitados por la CREG
% final_hueco_hora /

% tiempo_hueco /
% Tension_Residual / Datos solicitados por el estándar
% Profundidad /
% muestra_hueco

% inicio_interrupcion_fecha /
% inicio_interrupcion_hora /
% final_interrupcion_fecha / Datos solicitados por la CREG
% final_interrupcion_hora /
% DES y FES
% Interrupciones superiores a un minuto/

% tiempo_interrupcion / Datos solicitados por el estándar

% inicio_sobretension_fecha /
% inicio_sobretension_hora /
% final_sobretension_fecha / Datos solicitados por la CREG
% final_sobretension_hora /

% Amplitud_maxima_sobretension /
% tiempo_sobretension / Datos solicitados por el estándar
% muestra_sobretension

```

```

close all
clc
var_A=1;
var_B=1;
var_C=1;
var_umbral=1;
Urd(1)=127;

udin=127;
b=0;
k=2;
j=0;
%-----
a2=0;
a22=0;
tiempo2=0;
tiempo_hueco=0;
inicio_hueco=0;
inicio_hueco_fecha=0;
inicio_hueco_hora=0;
final_hueco=0;
final_hueco_fecha=0;
final_hueco_hora=0;

```

```

muestra_hueco=0;
Tension_Residual=0;
histeresis_hueco=(5/100)*udin;
umbral_inicio_hueco=0.9*udin;
umbral_final_hueco=0.9*udin+histeresis_hueco;
Profundidad=0;
maximo_hueco=0;
eneable_hueco=0;
%-----
a4=0;
a44=0;
tiempo4=0;
tiempo_sobretension=0;
muestra_sobretension=0;
inicio_sobretension=0;
inicio_sobretension_fecha=0;
inicio_sobretension_hora=0;
final_sobretension=0;
final_sobretension_fecha=0;
final_sobretension_hora=0;
Amplitud_maxima_sobretension=0;
histeresis_sobretension=(5/100)*udin;
umbral_inicio_sobretension=1.1*udin;
umbral_final_sobretension=1.1*udin-histeresis_sobretension;
eneable_sobretension=0;
%-----
a1=0;
a11=0;
tiempo1=0;
eneable=0;
tiempo_interrupcion=0;
muestra_interrupcion=0;
inicio_interrupcion=0;
inicio_interrupcion_fecha=0;
inicio_interrupcion_hora=0;
final_interrupcion=0;
final_interrupcion_fecha=0;
final_interrupcion_hora=0;
histeresis_interrupcion=(5/100)*udin;
umbral_inicio_interrupcion=0.1*udin;
umbral_final_interrupcion=0.1*udin+histeresis_interrupcion;
%-----
Nsample=128;
%Periodo de muestreo

Tm=(1/60)/Nsample;
Uef_medio_ciclo_A(1)=127;
Uef_medio_ciclo_B(1)=127;
Uef_medio_ciclo_C(1)=127;
%-----
%-----
yfundamentalA=yA;
yfundamentalB=yB;
yfundamentalC=yC;

yA=yA;
yB=yB;
yC=yC;
%|
%| INICIO DEL ALGORITMO
%|

```

```

% A continuación se presenta el algoritmo para la medición de eventos
% transitorios con base en el estándar IEC 61000-4-30:2003 y las
% resoluciones CREG 024 de 2005 y 016 de 2007 para una red trifásica.
% Esta se estructura de la siguiente manera:
% 1. Detección de los cruces por cero de cada fase.

```

```

% 2. Evaluación de la tensión eficaz de cada fase
% 3. Evaluación de la interrupción de Tensión en un sistema trifásico
% 4. Evaluación del Hueco de Tensión en un sistema trifásico
% 5. Evaluación de la Sobretensión Temporal en un sistema trifásico
% 6. Cambio de la referencia del umbral de detección.
maximoA=0;
maximoB=0;
maximoC=0;
nA=1;
nB=1;
nC=1;
umbral_variable=0;
q=1;
syms fase_hueco fase_sobretension

while ((nC)<(length(yfundamentalC)-3.5*Nsample)),
% -----
% Rutina para la detección de Cruces por cero de la fase A.
while q~=-1,
    while q~=-1,
        nA=nA+1;
% Inicialmente se verifica la existencia de una interrupción total de
% tensión, se cuenta la cantidad de ceros consecutivos durante un periodo
% de la señal en la fase A, si el resultado es igual a Nsample/2 implica
% que durante medio periodo no se presenta señal por lo no se puede
% detectar un cruce por cero, por tanto se procede a refrescar la señal
% medio periodo ya que este es el tiempo minimo para encontrar otro cruce por cero.

        if yfundamentalA(nA)==0,
            ceroA1=1;
        else
            ceroA1=0;
        end
        if yfundamentalA(nA+1)==0,
            ceroA2=1;
        else
            ceroA2=0;
        end
        if (ceroA1 & ceroA2)==1,
            senal_ceroA=senal_ceroA+1;
        else
            senal_ceroA=0;
        end

        if senal_ceroA==64,
            senal_cero_interrupcionA=1;
            senal_ceroA=0;
            break
        else
            senal_cero_interrupcionA=0;
        end

% como se expresó anteriormente si ocurre una interrupcion total, se
% procede a refrescar en medio periodo el proceso de medición.

        if senal_cero_interrupcionA==1,
            nA=nA+Nsample/2;
            break
        else

% Si no ocurre una interrupción total se procede a detectar el cruce por
% cero inicial durante el periodo de evaluación en la fase A. Las variables
% signoA1 y signoA2 contienen el signo de las muestra actual y posterior de la
% componente fundamental de la señal en la fase B. El cruce por cero es
% almacenado en nA.

            signoA1=sign(yfundamentalA(nA));

```

```

signoA2=sign(yfundamentalA(nA+1));
if yfundamentalA(nA)>maximoA,
    maximoA=yfundamentalA(nA);
end
if signoA1~=signoA2,
    nA=nA+1;

    for i=nA:nA+Nsample-1,
        errorA=abs(yfundamentalA(i)-yfundamentalA(i+1))/maximoA;

        if errorA>10,

            fA=0;
            break
        else
            fA=1;
        end

    end
    maximoA=0;

    if fA==0
        signoB1=0;
        signoB2=0;
    else

        nA=nA;
        N2A=nA+Nsample-1;
        break
    end

end

end

end

signoA1=0;
signoA2=0;
% Rutina para la detección de Cruces por cero de la fase B.
while q--=1,
    nB=nB+1;
% Inicialmente se verifica la existencia de una interrupción total de
% tensión, se cuenta la cantidad de ceros consecutivos durante un periodo
% de la señal en la fase B, si el resultado es igual a Nsample/2 implica
% que durante medio periodo no se presenta señal por lo no se puede
% detectar un cruce por cero, por tanto se procede a refrescar la señal
% medio periodo ya que este es el tiempo minimo para encontrar otro cruce
% por cero.
    if yfundamentalB(nB)==0,
        ceroB1=1;
    else
        ceroB1=0;
    end
    if yfundamentalB(nB+1)==0,
        ceroB2=1;
    else
        ceroB2=0;
    end
    if (ceroB1 & ceroB2)==1,
        senal_ceroB=senal_ceroB+1;
    else
        senal_ceroB=0;
    end

    if senal_ceroB==64,

```

```

    senal_cero_interrupcionB=1;
    senal_ceroB=0;
    break
else
    senal_cero_interrupcionB=0;
end

```

% como se expresó anteriormente si ocurre una interrupcion total, se  
% procede a refrescar en medio periodo el proceso de medición.

```

if senal_cero_interrupcionB==1,
    nB=nB+Nsample/2;
    break
else

```

% Si no ocurre una interrupción total se procede a detectar el cruce por  
% cero inicial durante el periodo de evaluación en la fase B. Las variables  
% signoB1 y signoB2 contiene el signo de las muestra actual y posterior de la  
% componente fundamental de la señal en la fase B. El cruce por cero es  
% almacenado en nB.

```

    signoB1=sign(yfundamentalB(nB));
    signoB2=sign(yfundamentalB(nB+1));
    if yfundamentalB(nB)>maximoB,
        maximoB=yfundamentalB(nB);
    end
    if signoB1~=signoB2
        nB=nB+1;
        for i=nB:nB+Nsample-1,
            errorB=abs(yfundamentalB(i)-yfundamentalB(i+1))/maximoB;

            if errorB>10,

                fB=0;
                break
            else
                fB=1;
            end

        end
        maximoB=0;
        if fB==0
            signoB1=0;
            signoB2=0;
        else
            nB=nB+1;
            N2B=nB+Nsample-1;
            break
        end
    end
end
end
signoB1=0;
signoB2=0;

```

% Rutina para la detección de Cruces por cero de la fase C.

```

while q--=-1,
    nC=nC+1;

```

% Inicialmente se verifica la existencia de una interrupción total de  
% tensión, se cuenta la cantidad de ceros consecutivos durante un periodo  
% de la señal en la fase C, si el resultado es igual a Nsample/2 implica  
% que durante medio periodo no se presenta señal por lo no se puede  
% detectar un cruce por cero, por tanto se procede a refrescar la señal  
% medio periodo ya que este es el tiempo minimo para encontrar otro cruce  
% por cero.

```

    if yfundamentalC(nC)==0,

```

```

    ceroC1=1;
else
    ceroC1=0;
end
if yfundamentalC(nC+1)==0,
    ceroC2=1;
else
    ceroC2=0;
end
if (ceroC1 & ceroC2)==1,
    senal_ceroC=senal_ceroC+1;
else
    senal_ceroC=0;
end

if senal_ceroC==64,
    senal_cero_interrupcionC=1;
    senal_ceroC=0;
    break
else
    senal_cero_interrupcion=0;
end

% como se expresó anteriormente si ocurre una interrupcion total, se
% procede a refrescar en medio periodo el proceso de medición.

if senal_cero_interrupcion==1,
    nC=nC+Nsample/2;
    break
else
    signoC1=sign(yfundamentalC(nC));
    signoC2=sign(yfundamentalC(nC+1));
    if yfundamentalC(nC)>maximoC,
        maximoC=yfundamentalC(nC);
    end
    if signoC1~=signoC2
% Si no ocurre una interrupción total se procede a detectar el cruce por
% cero inicial durante el periodo de evaluación en la fase C. Las variables
% signoC1 y signoC2 contiene el signo de las muestra actual y posterior de la
% componente fundamental de la señal en la fase C. El cruce por cero es
% almacenado en nC.

        for i=nC:nC+Nsample-1,
            errorC=abs(yfundamentalC(i)-yfundamentalC(i+1))/maximoC;

            if errorC>10,

                fC=0;
                break
            else
                fC=1;
            end

        end
        maximoC=0;
    if fC==0
        signoC1=0;
        signoC2=0;
    else
        nC=nC+1;
        N2C=nC+Nsample-1;
        break
    end
end
end
end
senal_cero_interrupcion=0;
senal_ceroC=0;

```

% Si algun periodo evaluado en la rutina de cruces por cero no es apta para  
 % la evluación de la medida, se reinicia el proceso comenzando en el tercer  
 % cruce por cero detectado. Si esto ocurre durannte un hueco de tensión, se  
 % perderia medio ciclo en la evaluación del tiempo del evento, por tanto se  
 % aumenta la variable tiempo\_hueco y hueco\_sobretensión en medio periodo.

```

    if fA==0,
        nA=inicioA(3)+2;
        nB=inicioA(3)+2;
        nC=inicioA(3)+2;
    elseif fB==0,
        nB=inicioB(3)+2;
        nC=inicioB(3)+2;
        nA=inicioB(3)+2;
    elseif fC==0,
        nC=inicioC(3)+2;
        nA=inicioC(3)+2;
        nB=inicioC(3)+2;
    end

    if (fA | fB | fC)==0
        if (habilitar_hueco)==1,
            tiempo2=tiempo2+1;
            tiempo_hueco(a2)=tiempo2*Tm*(Nsample/2);
        end
        if habilitar_sobretension==1
            tiempo4=tiempo4+1;
            tiempo_sobretension(a4)=tiempo4*Tm*(Nsample/2);
        end
    end

    if (fA & fB & fC)==1
        h=4;
        break
    else
        h=2;
    end
end

```

end

---

%  
 % EVALUACIÓN DE LA TENSIÓN EFICAZ EN CADA FASE  
 %

---

% Al encontrar el cruce por cero se procede a determinar la Uef\_medio\_ciclo  
 % (Tensión eficaz refrescada cada semiperiodo) de cada fase. Inicialmente  
 % se elevan las muestras al cuadrado, seguidamente se suman. El resultado  
 % es multiplicado por el inverso de la cantidad de muestras contenidas en  
 % el periodo de evaluación.

% Uef\_medio\_ciclo\_A. Tensión eficaz en la fase A  
 % Uef\_medio\_ciclo\_B. Tensión eficaz en la fase B  
 % Uef\_medio\_ciclo\_C. Tensión eficaz en la fase C

```

sumaA=0;
for i = nA:nA+Nsample-1,
    sumaA=yA(i).^2+sumaA;
end

    Uef_medio_ciclo_A(k)=sqrt((1/Nsample)*sumaA);

sumaB=0;
for i = nB:nB+Nsample-1,
    sumaB=yB(i).^2+sumaB;
end

    Uef_medio_ciclo_B(k)=sqrt((1/(Nsample))*sumaB);

```

%-----

```

sumaC=0;
for i = nC:nC+Nsample-1,
    sumaC=yC(i).^2+sumaC;
end

Uef_medio_ciclo_C(k)=sqrt((1/(Nsample))*sumaC);

```

```

%
%
%
%

```

---

### EVALUACIÓN DE LA INTERRUPCIÓN

---

```

% Una interrupción de tensión polifásica comienza cuando la tensión eficaz
% en las tres fases caen por debajo del umbral inicial de la interrupción y
% finaliza cuando la tensión eficaz en cualquiera de las fases se eleva por
% encima del umbral final de la interrupción. Para este evento solo se
% determina la duración ya que según el estándar IEC 61000-4-30 una
% interrupción de tensión es un caso especial de un hueco por tanto la
% evaluación de la amplitud queda registrada en el evento hueco de tensión.

```

```

% En esta lógica se determinan las condiciones para detectar el inicio
% del evento, se comparan las muestras actuales y anteriores para
% determinar si sobrepasaron el umbral inicial de interrupción.
% Si la tensión eficaz mas alta de la red polifásica es menor que el umbral inicial de la
% interrupción y la tensión eficaz mas alta de la red polifásica anterior es mayor
% que el umbral inicial de la interrupción entonces comienza una interrupción de tensión

```

```

if Uef_medio_ciclo_A(k)>Uef_medio_ciclo_B(k)
    actual_interrupcion=Uef_medio_ciclo_A(k);
else
    actual_interrupcion=Uef_medio_ciclo_B(k);
end
if Uef_medio_ciclo_C(k)>actual_interrupcion,
    actual_interrupcion=Uef_medio_ciclo_C(k);
end
if actual_interrupcion<umbral_inicio_interrupcion,
    logica_interrupcion_actual=1;
else
    logica_interrupcion_actual=0;
end

```

```

if Uef_medio_ciclo_A(k-1)>Uef_medio_ciclo_B(k-1)
    anterior_interrupcion=Uef_medio_ciclo_A(k-1);
else
    anterior_interrupcion=Uef_medio_ciclo_B(k-1);
end
if Uef_medio_ciclo_C(k-1)>anterior_interrupcion,
    anterior_interrupcion=Uef_medio_ciclo_C(k-1);
end
if anterior_interrupcion>umbral_inicio_interrupcion,
    logica_interrupcion_anterior=1;
else
    logica_interrupcion_anterior=0;
end

```

```

% Si comienza un evento se procede a inicializar las variables,
% registrar la fecha y hora del inicio del evento. Habilitar el
% proceso de marcado y evaluación de la medida.

```

```

if (logica_interrupcion_actual & logica_interrupcion_anterior)==1,

    a11=a11+1;
    a1=a1+1;
    inicio_interrupcion(a1)=nA;

```

```

inicio_interrupcion_fecha(a1)=floor(now);
inicio_interrupcion_hora(a1)=rem(now,1);
tiempo1=0;
muestra_interrupcion(a1)=0.11;
eneable=1;
marcado_interrupcion=1;
proceso_marcado=1;
end

```

%

---

```

% En esta sección se determina si la tensión eficaz actual mas alta en la red polifásica
% está por encima del umbral de finalización de la interrupción. Si esto es
% verdadero, se procede a finalizar la evaluación del evento, se registra la
% fecha y hora además de inactivar el proceso de marcado y evaluación de la medida.

```

```

if actual_interrupcion>umbral_final_interrupcion,
    logica_max_interrupcion=1;
else
    logica_max_interrupcion=0;
end

```

```

if (logica_max_interrupcion & eneable)==1,

```

```

    final_interrupcion(a1)=nA+(Nsample);
    final_interrupcion_fecha(a1)=floor(now);
    final_interrupcion_hora(a1)=rem(now,1);
    eneable=0;
    marcado_interrupcion=0;

```

```

end

```

```

% mientras la tensión eficaz actual mas alta esté por debajo del umbral final de interrupción,
% habilitar_interrupcion será 1 y procederá a incrementar la variable
% tiempo_1 que contabiliza el número de semiciclo registrados en el evento,
% la variable tiempo_interrupcion registra la duración del evento en
% segundos. La interrupción de tensión es un caso especial del hueco de
% tensión, por lo que las muestras quedan almacenadas en la rutina del
% hueco de tensión.

```

```

if eneable==1,

```

```

    tiempo1=tiempo1+1;
    tiempo_interrupcion(a1)=tiempo1*Tm*(Nsample/2);
end

```

%

---

## EVALUACIÓN DEL HUECO DE TENSIÓN

---

%

```

% Un hueco de tensión comienza cuando la tensión eficaz de cualquier fase cae por debajo del
% umbral inicial del hueco de tensión y finaliza cuando la tensión eficaz en todas las fases se
% elevan por encima del umbral final del hueco. Según el estándar
% IEC 61000-4-30:2003 la evaluación de este evento esta determinado por la
% duración del evento y por la tensión residual y su profundidad. La profundidad
% es la diferencia entre la tensión de entrada declarada (Udin) y la mínima
% tensión medida durante el evento y la tensión residual es la mínima tensión
% medida durante el evento.

```

```

% Determinar el inicio del evento a través de la lógica de detección.

```

```

% En esta lógica se determinan las condiciones para detectar el inicio
% del evento, se comparan las tensiones eficaces actuales y anteriores para
% determinar si sobrepasaron el umbral inicial de hueco de tensión.
% Si la tensión eficaz mas baja de la red polifásica es menor que el umbral inicial del hueco
% de tensión y la tensión eficaz mas baja de la red polifásica anterior es menor que el umbral inicial
% del hueco entonces comienza una interrupcion de tensión.

```

```

if Uef_medio_ciclo_A(k)<Uef_medio_ciclo_B(k)
    actual_hueco=Uef_medio_ciclo_A(k);
    proceso_marcadoA=1;

else
    actual_hueco=Uef_medio_ciclo_B(k);
    proceso_marcadoB=1;
end

if Uef_medio_ciclo_C(k)<actual_hueco,
    actual_hueco=Uef_medio_ciclo_C(k);
    proceso_marcadoC=1;
end
if actual_hueco<umbral_inicio_hueco,
    logica_hueco_actual=1;
else
    logica_hueco_actual=0;
    proceso_marcadoA=0;
    proceso_marcadoB=0;
    proceso_marcadoC=0;
end

if Uef_medio_ciclo_A(k-1)<Uef_medio_ciclo_B(k-1)
    anterior_hueco=Uef_medio_ciclo_A(k-1);
else
    anterior_hueco=Uef_medio_ciclo_B(k-1);
end
if Uef_medio_ciclo_C(k-1)<anterior_hueco,
    anterior_hueco=Uef_medio_ciclo_C(k-1);
end
if anterior_hueco>umbral_inicio_hueco,
    logica_hueco_anterior=1;
else
    logica_hueco_anterior=0;
end

```

% Si comienza un evento se procede a inicializar las variables,  
 % registrar la fecha y hora del inicio del evento. Habilitar el  
 % proceso de marcado y evaluación de la medida.

```

if (logica_hueco_actual & logica_hueco_anterior)==1,
    a22=a22+1;
    a2=a2+1;
    inicio_hueco(a2)=nA;
    inicio_hueco_fecha(a2)=floor(now);
    inicio_hueco_hora(a2)=rem(now,1);
    tiempo2=0;
    muestra_hueco(a22)=0;
    enable_hueco=1;
    Tension_Residual(a2)=actual_hueco;
    marcado_hueco=1;

end

```

% En esta sección se determina si la tensión eficaz actual mas baja en la red polifásica  
 % está por encima del umbral de finalización del hueco de tensión. Si esto es  
 % verdadero, se procede a finalizar la evaluación del evento, se registra la  
 % fecha y hora además de inactivar el proceso de marcado y evaluación de la medida.

```

if actual_hueco>umbral_final_hueco,
    logica_max_hueco=1;
else
    logica_max_hueco=0;
end

if (logica_max_hueco & enable_hueco)==1,
    final_hueco(a2)=nA+(Nsample+(Nsample/2-1));
    final_hueco_fecha(a2)=floor(now);

```

```

final_hueco_hora(a2)=rem(now,1);
eneable_hueco=0;
marcado_hueco=0;
end

```

% mientras la tensión eficaz actual mas baja esté por debajo del umbral final del hueco de tensión,  
 % habilitar\_hueco será 1 y procederá a incrementar la variable  
 % tiempo\_2 que contabiliza el número de semiciclo registrados en el evento,  
 % la variable tiempo\_hueco registra la duración del evento en  
 % segundos. Las muestras del evento se almacenan en la variable  
 % muestra\_hueco. fase\_hueco almacena la fase(s) donde se registró el evento

```

if eneable_hueco==1,
  if Uef_medio_ciclo_A(k-3)<umbral_final_hueco,
    j=j+1;
    fase_hueco(a2,j)='fase_A';
  else
    j=j+1;
    fase_hueco(a2,j)='0';
  end
  if Uef_medio_ciclo_B(k-3)<umbral_final_hueco,
    j=j+1;
    fase_hueco(a2,j)='fase_B';
  else
    j=j+1;
    fase_hueco(a2,j)='0';
  end
  if Uef_medio_ciclo_C(k-3)<umbral_final_hueco,
    j=j+1;
    fase_hueco(a2,j)='fase_C';
  else
    j=j+1;
    fase_hueco(a2,j)='0';
  end
end
j=0;

tiempo2=tiempo2+1;
tiempo_hueco(a2)=tiempo2*Tm*(Nsample/2);
actual_hueco;
if actual_hueco~=0;
  a22=a22+1;
  muestra_hueco(a22)=actual_hueco;
end
if actual_hueco<Tension_Residual(a2),

    Tension_Residual(a2)=actual_hueco;
    Profundidad(a2)=udin-actual_hueco;
end
end
end

```

%

---

% **EVALUACIÓN DE LA SOBRETENSIÓN TEMPORAL**

---

%

% La sobretensión temporal comienza cuando la tensión eficaz mas alta en la red polifásica está por encima  
 % del umbral inicial de la sobretensión temporal y finaliza cuando la tensión  
 % eficaz mas alta en la red polifásica esta por debajo del umbral final de la sobretensión temporal.  
 % El estándar IEC 61000-4-30:2003 evalúa la medida a través de la amplitud máxima de la  
 % tensión eficaz durante el evento y la duración del mismo.  
 %

---

% Determinar el inicio del evento a través de la lógica de detección.  
 % En esta lógica se determinan las condiciones para detectar el inicio  
 % del evento, se comparan las tensiones eficaces actuales y anteriores para  
 % determinar si sobrepasaron el umbral inicial de la sobretensión temporal.

% Si la tensión eficaz actual mas alta de la red polifásica es mayor que el umbral inicial de la  
 % sobretensión y la tensión eficaz anterior mas alta de la red polifásica es menor que el umbral inicial  
 % de la sobretensión entonces comienza una sobretensión temporal.

```

if Uef_medio_ciclo_A(k)>Uef_medio_ciclo_B(k)
    actual_sobretension=Uef_medio_ciclo_A(k);
else
    actual_sobretension=Uef_medio_ciclo_B(k);
end
if Uef_medio_ciclo_C(k)>actual_sobretension,
    actual_sobretension=Uef_medio_ciclo_C(k);
end
if actual_sobretension>umbral_inicio_sobretension,
    logica_sobretension_actual=1;
else
    logica_sobretension_actual=0;
end
  
```

```

if Uef_medio_ciclo_A(k-1)>Uef_medio_ciclo_B(k-1)
    anterior_sobretension=Uef_medio_ciclo_A(k-1);
else
    anterior_sobretension=Uef_medio_ciclo_B(k-1);
end
if Uef_medio_ciclo_C(k-1)>anterior_sobretension,
    anterior_sobretension=Uef_medio_ciclo_C(k-1);
end
if anterior_sobretension<umbral_inicio_sobretension,
    logica_sobretension_anterior=1;
else
    logica_sobretension_anterior=0;
end
  
```

% Si comienza un evento se procede a inicializar las variables,  
 % registrar la fecha y hora del inicio del evento. Habilitar el  
 % proceso de marcado y evaluación de la medida.

```

if (logica_sobretension_actual & logica_sobretension_anterior)==1,
    a44=a44+1;
    a4=a4+1;
    inicio_sobretension(a4)=nA-(Nsample-2);
    inicio_sobretension_fecha(a4)=floor(now);
    inicio_sobretension_hora(a4)=rem(now,1);
    tiempo4=0;
    muestra_sobretension(a44)=0;
    enenable_sobretension=1;
    Amplitud_maxima_sobretension(a4)=actual_sobretension;
    marcado_sobretension=1;
    proceso_marcado=1;
end
  
```

% En esta sección se determina si la tensión eficaz actual mas alta en la red polifásica  
 % está por debajo del umbral de finalización de la sobretensión temporal. Si esto es  
 % verdadero, se procede a finalizar la evaluación del evento, se registra la  
 % fecha y hora además de inactivar el proceso de marcado y evaluación de la medida.

```

if actual_sobretension<umbral_final_sobretension,
    logica_max_sobretension=1;
else
    logica_max_sobretension=0;
end
  
```

```

if (logica_max_sobretension & enenable_sobretension)==1,

    final_sobretension(a4)=nA+(Nsample+(Nsample/2-1));
    final_sobretension_fecha(a4)=floor(now);
    final_sobretension_hora(a4)=rem(now,1);
    enenable_sobretension=0;
    marcado_sobretension=0;
end
  
```

% mientras la tensión eficaz actual mas alta de la red polifásica esté por  
 % debajo del umbral final de la sobretensión,

```

% habilitar_sobretension será 1 y procederá a incrementar la variable
% tiempo_4 que contabiliza el número de semiciclo registrados en el evento,
% la variable tiempo_sobretension registra la duración del evento en
% segundos. Las muestras del evento se almacenan en la variable
% muestra_sobretension. fase_sobretension almacena la fase(s) donde se registró el evento
if eneeable_sobretension==1,

    if Uef_medio_ciclo_A(k-3)>umbral_final_sobretension,
        j=j+1;
        fase_sobretension(a4,j)='fase_A';
    else
        j=j+1;
        fase_sobretension(a4,j)='0';
    end
    if Uef_medio_ciclo_B(k-3)>umbral_final_sobretension,
        j=j+1;
        fase_sobretension(a4,j)='fase_B';
    else
        j=j+1;
        fase_sobretension(a4,j)='0';
    end
    if Uef_medio_ciclo_C(k-3)>umbral_final_sobretension,
        j=j+1;
        fase_sobretension(a4,j)='fase_C';
    else
        j=j+1;
        fase_sobretension(a4,j)='0';
    end
    j=0;

    a44=a44+1;
    tiempo4=tiempo4+1;
    tiempo_sobretension(a4)=tiempo4*Tm*(Nsample/2);
    muestra_sobretension(a44)=actual_sobretension;

    if actual_sobretension>Amplitud_maxima_sobretension(a4),
        Amplitud_maxima_sobretension(a4)=actual_sobretension;
    end

end

%
%-----
%En esta sección se aumenta la variable de almacenamiento de las muestras
%de las muestras después se pregunta por el valor de "k" con el fin de
%eliminar las muestras anteriores y evitar el almacenamiento excesivo de
%las muestras.
% desp
%
%-----
%Si ocurrió un hueco de tensión automáticamente marca la ventana de 12
%ciclo y la tensión de Urd no se actualiza. Esto se consigue con la
%variable Habilitar_Urd.

if nA>1536,
    var_A=var_A+1;
    evento_A(var_A)=proceso_marcadoA;
end
if nB>1536,
    var_B=var_B+1;
    evento_B(var_B)=proceso_marcadoB;
end
if nC>1536,
    var_C=var_C+1;
    evento_C(var_C)=proceso_marcadoC;
end

if var_A==24,
    habilitar_Urd=1;

```

```

else
    habilitar_Urd=0;
end

%_____Habilitar la tensión de udin deslizante Urd_____
%Al principio del archivo se encuentra la variable umbral_variable, esta se
%utiliza para que el umbral esté en función de Udin o Urd.

if (habilitar_Urd & umbral_variable)==1,

    resultadoA=0;
    for i = 1:length(evento_A),
        resultadoA=evento_A(i)+resultadoA;
    end

    resultadoB=0;
    for i = 1:length(evento_B),

        resultadoB=evento_B(i)+resultadoB;
    end

    resultadoC=0;
    for i = 1:length(evento_B),

        resultadoC=evento_C(i)+resultadoC;
    end

    if resultadoA==0,
        %Evaluación de la tensión Urd definido
        %en el estándar IEC 61000-4-30:2003
        var_umbral=var_umbral+1;

        Urd(var_umbral)=0.9967*Urd(var_umbral-1)+0.0033*Uef_12_cycle_A(var_umbral);

    elseif resultadoB==0,
        %Evaluación de la tensión Urd definido
        %en el estándar IEC 61000-4-30:2003
        Urd(var_umbral)=0.9967*Urd(var_umbral-1)+0.0033*Uef_12_cycle_B(var_umbral);
    elseif resultadoC==0,
        %Evaluación de la tensión Urd definido
        %en el estándar IEC 61000-4-30:2003
        Urd(var_umbral)=0.9967*Urd(var_umbral-1)+0.0033*Uef_12_cycle_C(var_umbral);
    else
        % si existe una ventana de doce ciclos
        % que este marcada, la tensión de
        % referencia actual es igual a la tensión
        % de referencia anterior.
        var_umbral=var_umbral+1;
        Urd(var_umbral)=Urd(var_umbral-1);
    end

    umbral_inicio_interrupcion(var_umbral)=0.1*Urd_A(var_umbral_A);
    umbral_final_interrupcion(var_umbral)=0.1*Urd_A(var_umbral_A)+histeresis_interrupcion;

    umbral_inicio_hueco(var_umbral)=0.9*Urd_A(var_umbral_A);
    umbral_final_hueco(var_umbral)=0.9*Urd_A(var_umbral_A)+histeresis_hueco;

    umbral_inicio_sobretension(var_umbral)=1.1*Urd_A(var_umbral_A);
    umbral_final_sobretension(var_umbral)=1.1*Urd_A(var_umbral_A)-histeresis_sobretension;

    var_A=1;
end

%_____Eliminar datos_____
%_____
%En esta sección se aumenta la variable de almacenamiento
%de las tensiones eficaces, después se pregunta por el valor de "k" con el fin de

```

%eliminar tensiones previas evitar el almacenamiento excesivo de datos..

```
k=k+1;
if k==10
    Uef_medio_ciclo_A(1:5)=[];
    Uef_medio_ciclo_B(1:5)=[];
    Uef_medio_ciclo_C(1:5)=[];
    k=5;
end
%-----
%-----
end

% si el equipo termina de operar y en ese momento ocurría un evento se
% registra la fecha y hora del mismo para evitar problemas con las
% dimensiones de los vectores.
if habilitar_interrupcion==1
    final_interrupcion_fecha(a1)=floor(now);
    final_interrupcion_hora(a1)=rem(now,1);
end

if habilitar_hueco==1
    final_hueco_fecha(a2)=floor(now);
    final_hueco_hora(a2)=rem(now,1);
end

if habilitar_sobretension==1
    final_sobretension_fecha(a4)=floor(now);
    final_sobretension_hora(a4)=rem(now,1);
end

disp('_____')
disp('EVALUACIÓN SOBRETENSIÓN')
disp('Amplitud Maxima-Duracion- inicio del evento- final del evento')
disp(' Año Mes Día Hora Minuto Segundo - Año Mes Día Hora Minuto Segundo' )

inicio_sobretension;
inicio_sobretension_fecha;
inicio_sobretension_hora;
inicio_sobretension_fecha1=datevec(inicio_sobretension_fecha);
inicio_sobretension_hora1=datevec(inicio_sobretension_hora);
fecha_hora_inicio_sobretension=inicio_sobretension_fecha1+inicio_sobretension_hora1;
final_sobretension;
final_sobretension_fecha;
final_sobretension_hora;
final_sobretension_fecha1=datevec(final_sobretension_fecha);
final_sobretension_hora1=datevec(final_sobretension_hora);
fecha_hora_final_sobretension=final_sobretension_fecha1+final_sobretension_hora1;
tiempo_sobretension;
tiempo_sobretension1=tiempo_sobretension;
Amplitud_maxima_sobretension;
sobretension1=Amplitud_maxima_sobretension;
inicio_final_sobretension=cat(2,sobretension1,tiempo_sobretension1,fecha_hora_inicio_sobretension,fecha_hora_final_sobretension);
disp(inicio_final_sobretension)
disp(fase_sobretension)

disp('_____')
disp('EVALUACIÓN INTERRUPCIÓN')
disp('Duracion- inicio del evento- final del evento')
disp(' Año Mes Día Hora Minuto Segundo - Año Mes Día Hora Minuto Segundo' )

inicio_interrupcion;
inicio_interrupcion_fecha;
inicio_interrupcion_hora;

tiempo_interrupcion;
```

```

tiempo_interruccion1=tiempo_interruccion';
final_interruccion;
final_interruccion_fecha;
final_interruccion_hora;
final_interruccion_fecha1=datevec(final_interruccion_fecha);
final_interruccion_hora1=datevec(final_interruccion_hora);
fecha_hora_final_interruccion=final_interruccion_fecha1+final_interruccion_hora1;

inicio_interruccion_fecha1=datevec(inicio_interruccion_fecha);
inicio_interruccion_hora1=datevec(inicio_interruccion_hora);
fecha_hora_inicio_interruccion=inicio_interruccion_fecha1+inicio_interruccion_hora1;

inicio_final_interruccion=cat(2,tiempo_interruccion1,fecha_hora_inicio_interruccion,fecha_hora_final_interruccion);
disp(inicio_final_interruccion)

%Almacenamiento de interrupciones según la resolución CREG 024 de 2005
interruccion_minutos_tiempo=0;
frecuencia_minutos=0;
if umbral_inicio_interruccion==0.1,

    tiempo_interruccion_minutos=tiempo_interruccion/60;
    h=1;

    for i=1:length(tiempo_interruccion_minutos);

        if tiempo_interruccion_minutos(i)>1

            interrupcion_minutos_tiempo(h)=tiempo_interruccion_minutos(i);
            frecuencia_minutos=frecuencia_minutos+1;
            h=h+1;
        end
    end
end

%Almacenamiento de las interrupciones superiores a un minuto y con tensión
%menor al 10% de la tensión nominal.
disp('_____')
disp('Datos Resolución CREG 024 de 2005')
interruccion_minutos_tiempo %Tiempo de duracion de cada evento superior a un minuto.
frecuencia_minutos %Cantidad de eventos ocurridos en el proceso de medición.

% Ahora se procede a determinar los DES y FES propuestos es la resolución
% CREG 025 de 1999.

interruccion_horas_tiempo=0;
tiempo_interruccion_horas=tiempo_interruccion/3600;

for i=1:length(tiempo_interruccion_horas)

    interrupcion_horas_tiempo=tiempo_interruccion_horas(i)+interruccion_horas_tiempo;

end
disp('_____')
disp('Datos Resolución CREG 089 de 1999 ')
%"Sumatoria del número de veces que el servicio es interrumpido en un
%circuito, mientras esté conectada la unidad constructiva a la red
%eléctrica".
FES=a1

%"Sumatoria del tiempo en horas ed las interrupciones del servicio en un
%circuito, mientras esté conectada la unidad constructiva a la red
%eléctrica".

```

DES=interrupcion\_horas\_tiempo

```
disp('_____')
disp('EVALUACIÓN HUECOS')
disp('Tensión residual-Profundidad-Duración- inicio del evento- final del evento')
disp(' Año Mes Día Hora Minuto Segundo - Año Mes Día Hora Minuto Segundo')
inicio_hueco_fecha1 = datevec(inicio_hueco_fecha);
inicio_hueco_hora1 = datevec(inicio_hueco_hora);
fecha_hora_inicio_hueco=inicio_hueco_fecha1+inicio_hueco_hora1;
final_hueco_fecha1 = datevec(final_hueco_fecha);
final_hueco_hora1 = datevec(final_hueco_hora);
fecha_hora_final_hueco=final_hueco_fecha1+final_hueco_hora1;
visualizar_hueco=Tension_Residual';
visualizar_Profundidad=Profundidad';
tiempo_hueco1=tiempo_hueco';
inicio_final_hueco=cat(2,visualizar_hueco,visualizar_Profundidad,tiempo_hueco1,fecha_hora_inicio_hueco,fecha_hora_final_hueco);
disp(inicio_final_hueco)
disp(fase_hueco)

nA=1:1:9000;
nB=1:1:9000;
nC=1:1:9000;

subplot(2,2,[1 2])
plot(nA,yA,'g',nB,yB,'m',nC,yC,'b')
% legend('tension rms','umbral inicial','umbral final');
title('Señal Trifásica Discreta ')
xlabel('muestras')
ylabel('amplitud')
axis([-10 length(yA)+10 -250 250])

subplot(2,3,4)
n2=1:length(muestra_hueco);
plot(n2,muestra_hueco,'b*')
% legend('tension rms','umbral inicial','umbral final');
title('Hueco de Tensión')
xlabel('muestras')
ylabel('amplitud')
axis([0 length(muestra_hueco)+1 0 max(muestra_hueco)+10])

subplot(2,3,5)
n4=1:length(muestra_sobretension);
plot(n4,muestra_sobretension,'b*')
title('Sobretensión Temporal')
xlabel('muestras')
ylabel('amplitud')

subplot(2,3,6)
n1=1:length(muestra_interrupcion);
plot(n1,muestra_interrupcion,'b*')
% legend('tension rms','umbral inicial','umbral final');
title('Interrupción de Tensión')
xlabel('muestras')
ylabel('amplitud')
axis([0 length(muestra_interrupcion)+1 0 max(muestra_interrupcion)+0.05])
```

#### d4) Medidor trifásico implementado con muestra variable.

% DISEÑO DEL PROTOTIPO DEL MÓDULO PARA LA MONITORIZACIÓN DE LAS PERTURBACIONES  
% DE TENSIÓN PARA UNA UNIDAD CONSTRUCTIVA EN EL ANÁLISIS DE LOS PARÁMETROS  
% DE CALIDAD DE POTENCIA ELÉCTRICA CON BASE EN EL ESTÁNDAR IEC 61000-4-30

% ALGORITMO PARA LA MEDICIÓN DE LAS PERTURBACIONES DE TENSIÓN DE ESTADO  
% TRANSITORIO PARA UNA RED MONOFÁSICA  
% HUECOS DE TENSIÓN, INTERRUPCIONES DE TENSIÓN Y SOBRETENSIONES TEMPORALES.  
% AUTORES:  
% ANDRÉS MAURICIO SALINAS CALA COD 2033708 ESTUDIANTE INGENIERÍA ELECTRÓNICA  
% OSCAR JAVIER CORREDOR MONTERO COD 2020801 ESTUDIANTE INGENIERIA ELÉCTRICA

% \_\_\_\_\_ ENTRADAS \_\_\_\_\_

% Definiciones.

% "Tensión eficaz refrescada cada semiperiodo(Uef\_medio\_ciclo): Valor de la tensión  
% eficaz medida en un periodo, comenzando en un paso por cero de la fundametal,  
% y refrescada cada semiperiodo." [IEC 61000-4-30:2003.]

% -yfundamental: La componente fundamental de la señal de la red es  
% utilizada para detectar el cruce por cero al inicio del periodo según lo  
% establecido en el Estándar IEC 61000-4-30.  
% Medición con muestra variable. El algoritmo detecta los  
% cruces por cero al inicio y al final del periodo, el cruce por cero al  
% inicio es almacenado en n y el cruce por cero al final del periodo se  
% almacena en la variable N2.

% -yn: es la señal de la red, con armonicos, interarmonicos, flicker, etc.

% -Nsample: muestras por ciclo de la ventana de observación, en el caso de  
% las pruebas realizadas este fue de 128 muestras/ciclo.

% -umbral\_variable: Los umbrales pueden ser un porcentaje de  
% Udin o Urd por tanto se deja una variable para que el usuario decida que  
% referencia utilizar. Si el usuario decide utilizar el umbral constante,  
% el algoritmo utilizará los umbrales almacenados en la variable  
% umbral\_inicio\_evento(1) y umbral\_final\_evento(1).

% -umbral\_inicio\_interrupcion(1): umbral donde comienza la interrupción de tensión.  
% -umbral\_inicio\_hueco(1): umbral donde comienza el hueco de tensión.  
% -umbral\_inicio\_sobretension(1): umbral donde comienza la sobretensión temporal.  
% -umbral\_final\_interrupcion(1): umbral donde finaliza la interrupción de tensión.  
% -umbral\_final\_hueco(1): umbral donde finaliza el hueco de tensión.  
% -umbral\_final\_sobretension(1): umbral donde finaliza la sobretensión temporal.

% -Udin (Tensión de entrada declarada): tension declarada por el usuario u  
% operador de red para tomarla como referencia en la declaración de la  
% histeresis y umbrales de los eventos.

% -Urd (Tensión de referencia deslizante): tension que varia cada doce ciclos,  
% La tension Urd actual esta en función de la tensión Urd anterior mas la  
% tensión eficaz de doce ciclos.

% histeresis\_interrupcion: porcentaje de udin, usado para determinar el  
% umbral final de la interrupción.

% histeresis\_hueco: porcentaje de udin, usado para determinar el  
% umbral final del hueco.

% histeresis\_sobretension: porcentaje de udin, usado para determinar el  
% umbral final de la sobretensión temporal.

% -Tm: Tiempo de muestreo del conversor. Esta variable está en funcion de

% la frecuencia medida. La duración de la medida esta en funcion de los  
 % semiciclos medidos, y estos a su vez se multiplican por el tiempo de  
 % muestreo medido, de allí la importancia de su precisión.

```

%_____SALIDAS_____
% inicio_hueco_fecha /
% inicio_hueco_hora /
% final_hueco_fecha / Datos solicitados por la CREG
% final_hueco_hora /

% tiempo_hueco /
% Tension_Residual / Datos solicitados por el estándar
% Profundidad /
% muestra_hueco

% inicio_interrupcion_fecha /
% inicio_interrupcion_hora /
% final_interrupcion_fecha / Datos solicitados por la CREG
% final_interrupcion_hora /
% DES y FES
% Interrupciones superiores a un minuto/

% tiempo_interrupcion / Datos solicitados por el estándar

% inicio_sobretension_fecha /
% inicio_sobretension_hora /
% final_sobretension_fecha / Datos solicitados por la CREG
% final_sobretension_hora /

% Amplitud_maxima_sobretension /
% tiempo_sobretension / Datos solicitados por el estándar
% muestra_sobretension
%_____INICIALIZACIÓN DE VARIABLES_____
clc
close all

cuadrado=0;
suma=0;
udin=127/sqrt(2);
Urd(1)=udin;
signo1=0;
signo2=0;
k=5;
var_umbral=1;
var=1;
inicio=0;
tramo1=0;
tramo2=0;
proceso_marcado=0;
senal_cero_interrupcion=0;
habilitar_interrupcion=0;
habilitar_hueco=0;
habilitar_sobretension=0;
%_____
umbral_inicio_interrupcion=0;
umbral_final_interrupcion=0;
umbral_inicio_hueco=0;
umbral_final_hueco=0;
umbral_inicio_sobretension=0;
umbral_final_sobretension=0;
%_____
a2=0;
a22=0;
tiempo2=0;
tiempo_hueco=0;
inicio_hueco=0;

```

```

inicio_hueco_fecha=0;
inicio_hueco_hora=0;
final_hueco=0;
final_hueco_fecha=0;
final_hueco_hora=0;
muestra_hueco=0;
Tension_Residual=0;
histeresis_hueco=(5/100)*udin;
umbral_inicio_hueco(1)=0.9*udin;
umbral_final_hueco(1)=0.9*udin+histeresis_hueco;
Profundidad=0;
enable_hueco=0;
%
a4=0;
a44=0;
tiempo4=0;
tiempo_sobretension=0;
muestra_sobretension=0;
inicio_sobretension=0;
inicio_sobretension_fecha=0;
inicio_sobretension_hora=0;
final_sobretension=0;
final_sobretension_fecha=0;
final_sobretension_hora=0;
Amplitud_maxima_sobretension=0;
histeresis_sobretension=(5/100)*udin;
umbral_inicio_sobretension(1)=1.1*udin;
umbral_final_sobretension(1)=1.1*udin-histeresis_sobretension;
enable_sobretension=0;
%
a1=0;
a11=0;
tiempo1=0;
enable_interrupcion=0;
tiempo_interrupcion=0;
muestra_interrupcion=0;
inicio_interrupcion=0;
inicio_interrupcion_fecha=0;
inicio_interrupcion_hora=0;
final_interrupcion=0;
final_interrupcion_fecha=0;
final_interrupcion_hora=0;
histeresis_interrupcion=(5/100)*udin;
umbral_inicio_interrupcion(1)=0.1*udin;
umbral_final_interrupcion(1)=0.1*udin+histeresis_interrupcion;
%
Nsample=128;           %Muestras por ciclo
Tm=Tm3;                %Periodo de muestreo
umbral_variable=0;
%

%|
%|          INICIO DEL ALGORITMO
%|

% A continuación se presenta el algoritmo para la medición de eventos
% transitorios con base en el estándar IEC 61000-4-30:2003 y las
% resoluciones CREG 024 de 2005 y 016 de 2007. Este se estructura de la
% siguiente manera:
% 1. Detección de los cruces por cero.
% 2. Evaluación de la tensión eficaz
% 3. Evaluación de la interrupción de Tensión
% 4. Evaluación del Hueco de Tensión
% 5. Evaluación de la Sobretensión Temporal.
% 6. Cambio de la referencia del umbral de detección.

```

```

n=1;
cruce=0;
maximo=0;
q=1;
t=cputime;
while n<(numel(yfundamental)-3*Nsample),
%
% | EVALUACIÓN DEL VRMS REFRESCADO CADA MEDIO CICLO
%
% Rutina para la detección de Cruces por cero.

    while q~-1,
        n=n+1;

% Inicialmente se verifica la existencia de una interrupcion total de
% tensión, se cuenta la cantidad de ceros consecutivos durante un periodo
% de la señal, si el resultado es 64 implica que durante medio periodo no se
% presenta señal por lo no se puede detectar un cruce por cero, por tanto se
% procede a refrescar la señal medio periodo ya que este es el tiempo minimo
% para encontrar otro cruce por cero.
        for i=n:Nsample,

            if yfundamental(i)==0,
                cero1=1;
            else
                cero1=0;
            end
            if yfundamental(i+1)==0,
                cero2=1;
            else
                cero2=0;
            end
            if (cero1 & cero2)==1,
                senal_cero=senal_cero+1;
            else
                senal_cero=0;
            end

            if senal_cero==64,
                senal_cero_interrupcion=1;
                senal_cero=0;
                break
            else
                senal_cero_interrupcion=0;
            end

        end
        %
% como se expresó anteriormente si ocurre una interrupcion total, se
% procede a refrescar en medio periodo el proceso de medición.
        if senal_cero_interrupcion==1,

            n=n+64;
            N2=n+Nsample-1;

            break

        else

% Si no ocurre una interrupción total se procede a detectar los cruce por
% cero inicial y final durante el periodo de evaluación. Las variables
% signo1 y signo2 contiene el signo de las muestra actual y posterior de la
% componente fundamental de la señal de la red. Los cruces por cero son
% almacenados en la inicio(). La variable cruce lleva el conteo de la
% cantidad de cruces almacenado.
            signo1=sign(yfundamental(n));

```

```

signo2=sign(yfundamental(n+1));
if yfundamental(n)>maximo,
    maximo=yfundamental(n);
else
%
%     maximo=yfundamental(n+1);
end

if signo1~=signo2
    cruce=cruce+1;
    inicio(cruce)=n+1;
end
% cruce igual a tres indica que se tiene los cruces por cero de la señal
% fundamental. Ahora se procede a verificar si la señal es óptima para la
% evaluación de la tensión eficaz.
    if cruce==3,
maximo

% se determina la distancia entre el primero y segundo cruce y el segundo
% y tercero para verificar si ocurrió un cambio de fase.
    tramo1=inicio(2)-inicio(1);
    tramo2=inicio(3)-inicio(2);

    if tramo1>=60 && tramo1<=68
        logica_tramo1=1;
    else
        logica_tramo2=0;
    end

    if tramo2>=60 && tramo2<=68
        logica_tramo2=1;
    else
        logica_tramo2=0;
    end

    if (logica_tramo1 & logica_tramo2)==1
% Si la señal es óptima en cuanto a las distancias entre los cruces por cero
% se procede a revisar las distancias entre muestras para determinar
% si ocurrió un cambio brusco en la tensión.

        for i=inicio(1):inicio(3)-1,
            error=abs(yfundamental(i)-yfundamental(i+1))/maximo;

            if error>0.05,
                error;
                f=1;
                break
            else
                f=0;
            end

        end
        maximo=0;
    else
        f=1;
    end

% Si la señal no es la ideal para evaluar la medida se procede a buscar
% tres nuevos cruces por cero.
    if f==1
        n=inicio(3)+2;
        cruce=0;
        inicio=0;
        tramo1=0;
        tramo2=0;
        signo1=0;
        signo2=0;
    else

```



```

if Uef_medio_ciclo(k-1)>umbral_inicio_interrupcion(var_umbral),
    logica_interrupcion_anterior=1;
else
    logica_interrupcion_anterior=0;
end

% Si la muestra actual es menor que el umbral inicial de la interrupción y
% la muestra anterior es mayor que el umbral inicial de la interrupción
% comienza el evento. se realiza la inicialización de variables,
% registro de la fecha y hora del inicio del evento. Habilitación del
% proceso de marcado y evaluación de la medida.
if (logica_interrupcion_actual & logica_interrupcion_anterior)==1,
    a11=a11+1;
    a1=a1+1;
    inicio_interrupcion_fecha(a1)=floor(now);
    inicio_interrupcion_hora(a1)=rem(now,1);
    tiempo1=2;
    muestra_interrupcion(a11)=0.1*udin+0.1;
    habilitar_interrupcion=1;
    proceso_marcado=1;
end

%


---


%En esta sección se determina si Uef_medio_ciclo está por encima del
%umbral de finalización de la interrupción. Si esto es verdadero, se procede
%a finalizar la evaluación del evento, se registra la fecha y hora además
%de inactivar el proceso de marcado y evaluación de la medida.
if Uef_medio_ciclo(k)>umbral_final_interrupcion(var_umbral),
    logica_max_interrupcion=1;
else
    logica_max_interrupcion=0;
end

if (logica_max_interrupcion & habilitar_interrupcion)==1,
    final_interrupcion(a1)=n+(Nsample);
    final_interrupcion_fecha(a1)=floor(now);
    final_interrupcion_hora(a1)=rem(now,1);
    habilitar_interrupcion=0;
    proceso_marcado=0;
end

%


---


% mientras Uef_medio_ciclo esté por debajo del umbral final de interrupción
% habilitar_interrupcion será 1 y procederá a contabilizar la duración, la
% variable tiempo 1 contabiliza los semiciclos transcurridos durante el
% evento, en vista que la interrupción quedará almacenada en el evento
% hueco, no se necesita almacenar las muestras de la interrupción en esta
% el evento interrupción.
if habilitar_interrupcion==1,
    tiempo1=tiempo1+1;
    tiempo_interrupcion(a1)=tiempo1*Tm*((N2-n+1)/2);
    if Uef_medio_ciclo(k)~=0,
        a11=a11+1;
        muestra_interrupcion(a11)=Uef_medio_ciclo(k);
    end
end

%


---


%
%
%


---


%
%
%


---


%
%
%


---


%
%
%


---


% Un hueco de tensión comienza cuando la tensión eficaz cae por debajo del
% umbral inicial del hueco de tensión y finaliza cuando la tensión eficaz se
% eleva por encima del umbral final del hueco. Según el estándar
% IEC 61000-4-30:2003 la evaluación de este evento esta determinada por la
% duración del evento y por la tensión residual y la profundidad. La profundidad

```

% es la diferencia entre la tensión de entrada declarada y la mínima  
 % tensión medida durante el evento y la tensión residual es la mínima tensión  
 % medida durante el evento.

%

%Determinar el inicio del evento a través de la lógica de detección

```

if Uef_medio_ciclo(k)<umbral_inicio_hueco(var_umbral),
    logica_hueco_actual=1;
else
    logica_hueco_actual=0;
end

if Uef_medio_ciclo(k-1)>umbral_inicio_hueco(var_umbral),
    logica_hueco_anterior=1;
else
    logica_hueco_anterior=0;
end

if (logica_hueco_actual & logica_hueco_anterior)==1,
    a22=a22+1;
    a2=a2+1;
    inicio_hueco_fecha(a2)=floor(now);
    inicio_hueco_hora(a2)=rem(now,1);
    tiempo2=1;
    muestra_hueco(a22)=0;
    habilitar_hueco=1;
    Tension_Residual(a2)=Uef_medio_ciclo(k);
    Profundidad(a2)=udin-Uef_medio_ciclo(k);
    proceso_marcado=1;
end
  
```

%

%Determinar el final del evento, registro de la muestra, fecha y hora

```

if Uef_medio_ciclo(k)>umbral_final_hueco(var_umbral),
    logica_max_hueco=1;
else
    logica_max_hueco=0;
end
  
```

%Si la tensión actual es mayor que el umbral final de hueco de tensión entonces  
 %se finaliza el hueco de tensión. El proceso de marcado y habilitación de  
 %la evaluación de la medida se desactiva.

```

if (logica_max_hueco & habilitar_hueco)==1,

    final_hueco(a2)=n+(3/2)*Nsample-1;
    final_hueco_fecha(a2)=floor(now);
    final_hueco_hora(a2)=rem(now,1);
    habilitar_hueco=0;
    proceso_marcado=0;
end
  
```

%

%si la tensión está por debajo del umbral final del hueco de tensión, se  
 %registra la duración, Tension\_Residual, Profundidad del evento. Además del  
 %almacenamiento de las muestras del evento.

```

if habilitar_hueco==1,
    tiempo2=tiempo2+1;
    tiempo_hueco(a2)=tiempo2*Tm*((N2-n+1)/2);
    if Uef_medio_ciclo(k)~=0;
        a22=a22+1;
        muestra_hueco(a22)=Uef_medio_ciclo(k);
    end

    if umbral_variable==0,
  
```



```

end
%
if habilitar_sobretension==1,
    a44=a44+1;
    tiempo4=tiempo4+1;
    tiempo_sobretension(a4)=tiempo4*Tm*((n-N2)/2);
    muestra_sobretension(a44)=Uef_medio_ciclo(k);

    if Uef_medio_ciclo(k)>Amplitud_maxima_sobretension(a4),
        Amplitud_maxima_sobretension(a4)=Uef_medio_ciclo(k);
    end
end

%
%Si ocurrió un hueco de tensión automáticamente marca la ventana de 12
%ciclo y la tensión Urd no se actualiza. Esto se consigue con la
%variable Habilitar_Urd.
k=k+1;
if n>1536,
    var=var+1;
    evento(var)=proceso_marcado;
end

if var==24,
    habilitar_Urd=1;
else
    habilitar_Urd=0;
end

% Habilitar la tensión de referencia deslizante Urd
%Al principio del archivo se encuentra la variable umbral_variable, esta se
%utiliza para que el umbral esté en función de Udin o Urd.
% Si umbral_variable esta en 1, se habilita la tensión de referencia
% deslizante, si es cero los umbrales están en función de udin.

if (habilitar_Urd & umbral_variable)==1,
    resultado=0;
    for i = 1:24,
        resultado=evento(i)+resultado;
    end

    if (resultado)==0,
        var_umbral=var_umbral+1;
        %Evaluación de la tensión Urd con base
        %en la definición del estándar
        Urd(var_umbral)=0.9967*Urd(var_umbral-1)+0.0033*Uef_12_cycle(var_umbral);
    else
        % Si umbral variable esta deshabilitado
        % entonces no se actualiza la tensión
        % Urd.
        var_umbral=var_umbral+1;
        Urd(var_umbral)=Urd(var_umbral-1);
    end

    umbral_inicio_interrupcion(var_umbral)=0.1*Urd(var_umbral);
    umbral_final_interrupcion(var_umbral)=0.1*Urd(var_umbral)+histeresis_interrupcion;

    umbral_inicio_hueco(var_umbral)=0.9*Urd(var_umbral);
    umbral_final_hueco(var_umbral)=0.9*Urd(var_umbral)+histeresis_hueco;

    umbral_inicio_sobretension(var_umbral)=1.1*Urd(var_umbral);
    umbral_final_sobretension(var_umbral)=1.1*Urd(var_umbral)-histeresis_sobretension;

var=1;
end
% Eliminar datos

```

```

%-----
%En esta sección se aumenta la variable de almacenamiento
%de las muestras después se pregunta por el valor de "k" con el fin de
%eliminar las muestras anteriores y evitar el almacenamiento excesivo de
%las muestras.
    if k==10
        Uef_medio_ciclo(1:5)=[];
        k=5;
    end
%-----
end
if habilitar_interrupcion==1
    final_interrupcion_fecha(a1)=floor(now);
    final_interrupcion_hora(a1)=rem(now,1);
end

if habilitar_hueco==1
    final_hueco_fecha(a2)=floor(now);
    final_hueco_hora(a2)=rem(now,1);
end

if habilitar_sobretension==1
    final_sobretension_fecha(a4)=floor(now);
    final_sobretension_hora(a4)=rem(now,1);
end
tiempo_proceso=cputime-t

%-----
%                               FINAL DEL
%                               ALGORITMO
%-----

%En esta sección se concatenan todas las variables de salida para que
%aparezcan de forma ordenada en la ventana de comandos de matlab
disp('_____MEDICIÓN DE LA TENSIÓN EFICAZ CON MUESTRA VARIABLE_____')
disp('EVALUACIÓN SOBRETENSIÓN')
disp('Amplitud Maxima-Duracion-      inicio del evento-      final del evento')
disp('      Año Mes Día Hora Minuto Segundo - Año Mes Día Hora Minuto Segundo')

inicio_sobretension_fecha1=datevec(inicio_sobretension_fecha);
inicio_sobretension_hora1=datevec(inicio_sobretension_hora);
fecha_hora_inicio_sobretension=inicio_sobretension_fecha1+inicio_sobretension_hora1;
final_sobretension_fecha1=datevec(final_sobretension_fecha);
final_sobretension_hora1=datevec(final_sobretension_hora);
fecha_hora_final_sobretension=final_sobretension_fecha1+final_sobretension_hora1;
tiempo_sobretension1=tiempo_sobretension;
sobretension1=Amplitud_maxima_sobretension;
inicio_final_sobretension=cat(2,sobretension1,tiempo_sobretension1,fecha_hora_inicio_sobretension,fecha_hora_final_sobretension);
disp(inicio_final_sobretension)

disp('_____')
disp('EVALUACIÓN INTERRUPCIÓN')
disp('Duracion-      inicio del evento-      final del evento')
disp('      Año Mes Día Hora Minuto Segundo - Año Mes Día Hora Minuto Segundo')

tiempo_interrupcion1=tiempo_interrupcion;
final_interrupcion_fecha1=datevec(final_interrupcion_fecha);
final_interrupcion_hora1=datevec(final_interrupcion_hora);
fecha_hora_final_interrupcion=final_interrupcion_fecha1+final_interrupcion_hora1;

inicio_interrupcion_fecha1=datevec(inicio_interrupcion_fecha);
inicio_interrupcion_hora1=datevec(inicio_interrupcion_hora);
fecha_hora_inicio_interrupcion=inicio_interrupcion_fecha1+inicio_interrupcion_hora1;

inicio_final_interrupcion=cat(2,tiempo_interrupcion1,fecha_hora_inicio_interrupcion,fecha_hora_final_interrupcion);

```

```

disp(inicio_final_interruccion)

%Almacenamiento de interrupciones según la resolución CREG 024 de 2005
tiempo_interruccion_minutos=tiempo_interruccion/60;
h=1;
frecuencia_minutos=0;
interruccion_minutos_tiempo=0;
for i=1:length(tiempo_interruccion_minutos);

    if tiempo_interruccion_minutos(i)>1

        interrupcion_minutos_tiempo(h)=tiempo_interruccion_minutos(i);
        frecuencia_minutos=frecuencia_minutos+1;
        h=h+1;
    end
end

%Almacenamiento de las interrupciones superiores a un minuto y con tensión
%menor al 10% de la tensión nominal.
interruccion_minutos_tiempo %Tiempo de duracion de cada evento superior a un minuto.
frecuencia_minutos %Cantidad de eventos ocurridos en el proceso de medición.

% Ahora se procede a determinar los DES y FES propuestos es la resolución
% CREG 025 de 1999.

interruccion_hora=0;
tiempo_interruccion_horas=tiempo_interruccion/3600;

for i=1:length(tiempo_interruccion_horas)

    interrupcion_hora=tiempo_interruccion_horas(i)+interruccion_hora;
end

%"Sumatoria del número de veces que el servicio es interrumpido en un
%circuito, mientras esté conectada la unidad constructiva a la red
%eléctrica"[CREG 024, 1999].
FES=a1

%"Sumatoria del tiempo en horas ed las interrupciones del servicio en un
%circuito, mientras esté conectada la unidad constructiva a la red
%eléctrica"[CREG 024, 1999].
DES=interruccion_hora

disp('_____')
disp('EVALUACIÓN HUECOS')
disp('Tensión residual-Profundidad-Duración- inicio del evento- final del evento')
disp(' Año Mes Día Hora Minuto Segundo - Año Mes Día Hora Minuto Segundo')
inicio_hueco_fecha1 = datevec(inicio_hueco_fecha);
inicio_hueco_hora1 = datevec(inicio_hueco_hora);
fecha_hora_inicio_hueco=inicio_hueco_fecha1+inicio_hueco_hora1;
final_hueco_fecha1 = datevec(final_hueco_fecha);
final_hueco_hora1 = datevec(final_hueco_hora);
fecha_hora_final_hueco=final_hueco_fecha1+final_hueco_hora1;
visualizar_hueco=Tension_Residual';
visualizar_Profundidad=Profundidad';
tiempo_hueco1=tiempo_hueco';
inicio_final_hueco=cat(2,visualizar_hueco,visualizar_Profundidad,tiempo_hueco1,fecha_hora_inicio_hueco,fecha_hora_final_hueco);
disp(inicio_final_hueco)

% error1=minimo_hueco-Tension_Residual

disp('_____')

```

```

% n=1:1:nfinal;
% figure,subplot(2,3,[1 2])
% plot(n,yn,'g')
% title('Señal Monofásica Discreta ')
% xlabel('muestras')
% ylabel('amplitud')
% axis([-10 length(yn)+10 min(yn)-10 max(yn)+10])
%
% subplot(2,3,3)
% plot(n,yfundamental,'g')
% title('Señal fundamental Discreta ')
% xlabel('muestras')
% ylabel('amplitud')
% axis([-10 length(yn)+10 min(yn)-10 max(yn)+10])

% figure,subplot(1,3,1)
% n2=1:length(muestra_hueco);
% plot(n2,muestra_hueco,'b*')
% title('Hueco de Tensión')
% xlabel('muestras')
% ylabel('amplitud [V]')
% axis([0 length(muestra_hueco)+1 0 max(muestra_hueco)+30])
%
% subplot(1,3,2)
% n4=1:length(muestra_sobretension);
% plot(n4,muestra_sobretension,'b*')
% title('Sobretensión Temporal')
% xlabel('muestras')
% ylabel('amplitud [V]')
% axis([0 length(muestra_sobretension)+1 0 max(muestra_sobretension)+30])
%
% subplot(1,3,3)
% n1=1:length(muestra_interrupcion);
% plot(n1,muestra_interrupcion,'b*')
% title('Interrupción de Tensión')
% xlabel('muestras')
% ylabel('amplitud [V]')
% axis([0 length(muestra_interrupcion)+1 0 max(muestra_interrupcion)+7])

n2=1:length(muestra_hueco);
figure,plot(n2,muestra_hueco,'b*',n2,umbral_inicio_hueco,'g*',n2,umbral_final_hueco,'g*')
title('Hueco de Tensión')
xlabel('muestras')
ylabel('amplitud [V]')
axis([0 length(muestra_hueco)+1 0 max(umbral_final_hueco)+30])

% n1=1:length(muestra_interrupcion);
% figure,plot(n1,muestra_interrupcion,'b*',n2,umbral_inicio_interrupcion,'g*',n2,umbral_final_interrupcion,'g*')
% title('Interrupción de Tensión')
% xlabel('muestras')
% ylabel('amplitud [V]')
% axis([0 length(muestra_interrupcion)+1 0 max(umbral_final_interrupcion)+7])

n=1:1:nfinal;
figure,plot(n,yn) % discreto
title('Señal de la red discretizada')
xlabel('muestras')
ylabel('amplitud [V]')
axis([0 length(yn)+1 -max(yn)-5 max(yn)+5])
zoom on

% t=0:Tfinal/1000:Tfinal;
% figure,plot(t,yfundamental)

```

## ANEXO D2. GENERADOR DE SEÑALES SINTÉTICAS EN TIEMPO DISCRETO.

### a) Monofásico

```

clear all
clc
close all
Nsample=128;
udin=127/sqrt(2);
Tm=(1/60)/Nsample;
Tfinal=0.7;
nfinal=fix(Tfinal/Tm);
n=1:1:nfinal;
#####
Fsist1=60;
Fsist2=60;
Fsist3=60;
#####
Amax1=127;
Amax2=15;
Amax3=127;
#####
Tm1=(1/Fsist1)/Nsample;
Tm2=(1/Fsist2)/Nsample;
Tm3=(1/Fsist3)/Nsample;
#####
n11=1000;
T11=n11*Tm;
T22=0.525;
n22=T22/Tm;
#####
armonico_3=(0/100)*udin;
armonico_5=(0/100)*udin;
armonico_29=(0/100)*udin;
interarmonico_a1=(0/100)*udin;
a3_armonico=3;
a5_armonico=5;
a29_armonico=29;
a1_interarmonico=7.5;
mod_flicker=35;
amp_flicker=0;
#####
Amax2=(-armonico_3.^2-armonico_5.^2-armonico_29.^2-interarmonico_a1.^2+324).^(1/2);
sqrt(Amax2^2+armonico_3^2+armonico_5^2+armonico_29^2+interarmonico_a1^2)
#####
y1=(Amax1*sin(2*pi*Fsist1*Tm1*n))+(armonico_3*sin(2*pi*a3_armonico*Fsist1*Tm1*n))+(armonico_5*sin(2*pi*a5_armonico
*Fsist1*Tm1*n))+(armonico_29*sin(2*pi*a29_armonico*Fsist1*Tm1*n))+(interarmonico_a1*sin(2*pi*a1_interarmonico*Fsist1*
Tm1*n));
y2=(Amax2*sin(2*pi*Fsist2*Tm2*n))+(armonico_3*sin(2*pi*a3_armonico*Fsist2*Tm2*n))+(armonico_5*sin(2*pi*a5_armonico
*Fsist2*Tm2*n))+(armonico_29*sin(2*pi*a29_armonico*Fsist2*Tm2*n))+(interarmonico_a1*sin(2*pi*a1_interarmonico*Fsist2*
Tm2*n));
y3=(Amax3*sin(2*pi*Fsist3*Tm3*n))+(armonico_3*sin(2*pi*a3_armonico*Fsist3*Tm3*n))+(armonico_5*sin(2*pi*a5_armonico
*Fsist3*Tm3*n))+(armonico_29*sin(2*pi*a29_armonico*Fsist3*Tm3*n))+(interarmonico_a1*sin(2*pi*a1_interarmonico*Fsist3*
Tm3*n));
#####
%Señal completa
yn=y1.*(n<n11)-y2.*(n>=n11)&(n<n22))+y3.*(n>=n22);%+(amp_flicker*square(2*mod_flicker*pi*n*Tm));
#####

```

```

y1f=(Amax1*sin(2*pi*Fsist1*Tm1*n));
y2f=(Amax2*sin(2*pi*Fsist2*Tm2*n));
y3f=(Amax3*sin(2*pi*Fsist3*Tm3*n));
#####
yfundamental=y1f.*(n<n11)-y2f.*((n>=n11) &(n<n22))+y3f.*(n>=n22);
#####
figure,plot(n,yn)
axis([0 length(yn)+1 -max(yn)-5 max(yn)+5])

```

## b) Trifásico

```

clear all
clc
close all
Nsample=128;
udin=127/sqrt(2);
Tm=(1/60)/Nsample;
Tfinal=0.5;           %Tiempo final de la señal sintética
nfinal=fix(Tfinal/Tm); %muestra final de la señal sintética
nA=1:1:nfinal;
nB=1:1:nfinal;
nC=1:1:nfinal;
t=0:0.00001:Tfinal;
n=nA;
%La señal sintética de prueba esta dividida en tres tramos con el fin de
%crear el evento en el segundo tramo.
#####
%__Frecuencia para cada tramo__
Fsist1=60;
Fsist2=61;
Fsist3=60;
#####
%__amplitud de la fundamental para cada tramo
Amax1A=127;
Amax2A=7;
Amax3A=127;
Amax1B=127;
Amax2B=155;
Amax3B=127;
Amax1C=127;
Amax2C=127;
Amax3C=127;
#####
%__Tiempo de muestreo para cada tramo__
Tm1=(1/Fsist1)/Nsample;
Tm2=(1/Fsist2)/Nsample;
Tm3=(1/Fsist3)/Nsample;
#####

T11A=0.05;           %final del primer intervalo, inicio del segundo intervalo
n11A=fix(T11A/Tm);
T22A=Tfinal-0.06;   %final del segundo intervalo, inicio del tercer intervalo
n22A=fix(T22A/Tm);

T11B=0.06;           %final del primer intervalo, inicio del segundo intervalo
n11B=fix(T11B/Tm);

```

T22B=Tfinal-0.06; %final del segundo intervalo, inicio del tercer intervalo  
n22B=fix(T22B/Tm);

T11C=0.05; %final del primer intervalo, inicio del segundo intervalo  
n11C=fix(T11C/Tm);  
T22C=Tfinal-0.06; %final del segundo intervalo, inicio del tercer intervalo  
n22C=fix(T22C/Tm);

#####  
%Variación de la amplitud de las componentes armónicas e interarmónicas de  
%las señales sintéticas creadas bajo el estándar IEC 61000-4-30:2003.  
%las constantes armonico\_x hacen referencia a las amplitudes de las  
%componentes que se encuentran en los ensayos 2 y 3.  
%La variable ax\_armonico indica la componente armónica incluida en la señal  
%sintética. en los ensayos 2 y 3 estas componentes son distintas por tanto  
%solo es cambiar la cantidad de la componente armónica. Igual se aplica para  
%la componente interarmónica.

armonico\_3=(0/100)\*udin;  
armonico\_5=(0/100)\*udin;  
armonico\_29=(0/100)\*udin;  
interarmonico\_a1=(0/100)\*udin;  
a3\_armonico=3;  
a5\_armonico=5;  
a29\_armonico=29;  
a1\_interarmonico=7.5;

mod\_flicker=35;  
amp\_flicker=0;

#####  
%Señal por tramos donde se incluye las componentes armónicas e  
%interarmónicas de la señal sintética, muestreada con tiempo de muestreo  
%Tmx donde x especifica el tramo.

y1A=(Amax1A\*sin(2\*pi\*Fsist1\*Tm1\*nA))+(armonico\_3\*sin(2\*pi\*a3\_armonico\*Fsist1\*Tm1\*nA))+(armonico\_5\*sin(2\*pi\*a5\_armonico\*Fsist1\*Tm1\*nA))+(armonico\_29\*sin(2\*pi\*a29\_armonico\*Fsist1\*Tm1\*nA))+(interarmonico\_a1\*sin(2\*pi\*a1\_interarmonico\*Fsist1\*Tm1\*nA));  
y2A=(Amax2A\*sin(2\*pi\*Fsist2\*Tm2\*nA))+(armonico\_3\*sin(2\*pi\*a3\_armonico\*Fsist2\*Tm2\*nA))+(armonico\_5\*sin(2\*pi\*a5\_armonico\*Fsist2\*Tm2\*nA))+(armonico\_29\*sin(2\*pi\*a29\_armonico\*Fsist2\*Tm2\*nA))+(interarmonico\_a1\*sin(2\*pi\*a1\_interarmonico\*Fsist2\*Tm2\*nA));  
y3A=(Amax3A\*sin(2\*pi\*Fsist3\*Tm3\*nA))+(armonico\_3\*sin(2\*pi\*a3\_armonico\*Fsist3\*Tm3\*nA))+(armonico\_5\*sin(2\*pi\*a5\_armonico\*Fsist3\*Tm3\*nA))+(armonico\_29\*sin(2\*pi\*a29\_armonico\*Fsist3\*Tm3\*nA))+(interarmonico\_a1\*sin(2\*pi\*a1\_interarmonico\*Fsist3\*Tm3\*nA));

y1B=(Amax1B\*sin(2\*pi\*Fsist1\*Tm1\*(nB-43)))+(armonico\_3\*sin(2\*pi\*a3\_armonico\*Fsist1\*Tm1\*(nB-43)))+(armonico\_5\*sin(2\*pi\*a5\_armonico\*Fsist1\*Tm1\*(nB-43)))+(armonico\_29\*sin(2\*pi\*a29\_armonico\*Fsist1\*Tm1\*(nB-43)))+(interarmonico\_a1\*sin(2\*pi\*a1\_interarmonico\*Fsist1\*Tm1\*(nB-43)));  
y2B=(Amax2B\*sin(2\*pi\*Fsist2\*Tm2\*(nB-43)))+(armonico\_3\*sin(2\*pi\*a3\_armonico\*Fsist2\*Tm2\*(nB-43)))+(armonico\_5\*sin(2\*pi\*a5\_armonico\*Fsist2\*Tm2\*(nB-43)))+(armonico\_29\*sin(2\*pi\*a29\_armonico\*Fsist2\*Tm2\*(nB-43)))+(interarmonico\_a1\*sin(2\*pi\*a1\_interarmonico\*Fsist2\*Tm2\*(nB-43)));  
y3B=(Amax3B\*sin(2\*pi\*Fsist3\*Tm3\*(nB-43)))+(armonico\_3\*sin(2\*pi\*a3\_armonico\*Fsist3\*Tm3\*(nB-43)))+(armonico\_5\*sin(2\*pi\*a5\_armonico\*Fsist3\*Tm3\*(nB-43)))+(armonico\_29\*sin(2\*pi\*a29\_armonico\*Fsist3\*Tm3\*(nB-43)))+(interarmonico\_a1\*sin(2\*pi\*a1\_interarmonico\*Fsist3\*Tm3\*(nB-43)));

y1C=(Amax1C\*sin(2\*pi\*Fsist1\*Tm1\*(nC+43)))+(armonico\_3\*sin(2\*pi\*a3\_armonico\*Fsist1\*Tm1\*(nC+43)))+(armonico\_5\*sin(2\*pi\*a5\_armonico\*Fsist1\*Tm1\*(nC+43)))+(armonico\_29\*sin(2\*pi\*a29\_armonico\*Fsist1\*Tm1\*(nC+43)))+(interarmonico\_a1\*sin(2\*pi\*a1\_interarmonico\*Fsist1\*Tm1\*(nC+43)));  
y2C=(Amax2C\*sin(2\*pi\*Fsist2\*Tm2\*(nC+43)))+(armonico\_3\*sin(2\*pi\*a3\_armonico\*Fsist2\*Tm2\*(nC+43)))+(armonico\_5\*sin(2\*pi\*a5\_armonico\*Fsist2\*Tm2\*(nC+43)))+(armonico\_29\*sin(2\*pi\*a29\_armonico\*Fsist2\*Tm2\*(nC+43)))+(interarmonico\_a1\*sin(2\*pi\*a1\_interarmonico\*Fsist2\*Tm2\*(nC+43)));  
y3C=(Amax3C\*sin(2\*pi\*Fsist3\*Tm3\*(nC+43)))+(armonico\_3\*sin(2\*pi\*a3\_armonico\*Fsist3\*Tm3\*(nC+43)))+(armonico\_5\*sin(2\*pi\*a5\_armonico\*Fsist3\*Tm3\*(nC+43)))+(armonico\_29\*sin(2\*pi\*a29\_armonico\*Fsist3\*Tm3\*(nC+43)))+(interarmonico\_a1\*sin(2\*pi\*a1\_interarmonico\*Fsist3\*Tm3\*(nC+43)));

```

#####
%Señal completa en el dominio de discreto y temporal
yAn=y1A.*(n<n11A)-y2A.*((n>=n11A)&(n<n22A))+y3A.*(n>=n22A);%+(amp_flicker*square(2*mod_flicker*pi*n*Tm));
% yAt=y1tA.*(t<T11A)-y2tA.*((t>=T11A)&(t<T22A))+y3tA.*(t>=T22A);%+(amp_flicker*square(2*mod_flicker*pi*t));

yBn=y1B.*(n<n11B)-y2B.*((n>=n11B)&(n<n22B))+y3B.*(n>=n22B);%+(amp_flicker*square(2*mod_flicker*pi*n*Tm));
% yBt=y1tB.*(t<T11B)-y2tB.*((t>=T11B)&(t<T22B))+y3tB.*(t>=T22B);%+(amp_flicker*square(2*mod_flicker*pi*t));

yCn=y1C.*(n<n11C)-y2C.*((n>=n11C)&(n<n22C))+y3C.*(n>=n22C);%+(amp_flicker*square(2*mod_flicker*pi*n*Tm));
% yCt=y1tC.*(t<T11C)-y2tC.*((t>=T11C)&(t<T22C))+y3tC.*(t>=T22C);%+(amp_flicker*square(2*mod_flicker*pi*t));

#####
%Componente fundamental en dominio discreto de la señal sintética por
%tramos
y1fA=(Amax1A*sin(2*pi*Fsist1*Tm1*n));
y2fA=(Amax2A*sin(2*pi*Fsist2*Tm2*n));
y3fA=(Amax3A*sin(2*pi*Fsist3*Tm3*n));

y1fB=(Amax1B*sin(2*pi*Fsist1*Tm1*n));
y2fB=(Amax2B*sin(2*pi*Fsist2*Tm2*n));
y3fB=(Amax3B*sin(2*pi*Fsist3*Tm3*n));

y1fC=(Amax1C*sin(2*pi*Fsist1*Tm1*n));
y2fC=(Amax2C*sin(2*pi*Fsist2*Tm2*n));
y3fC=(Amax3C*sin(2*pi*Fsist3*Tm3*n));

#####
% Componente fundamental de la señal sintética de prueba en el dominio
% y discreto
yAfundamental=y1fA.*(n<n11A)-y2fA.*((n>=n11A) &(n<n22A))+y3fA.*(n>=n22A);
% yAfundamental=y1fA.*(t<T11A)-y2fA.*((t>=T11A) &(t<T22A))+y3fA.*(t>=T22A);

yBfundamental=y1fB.*(n<n11B)-y2fB.*((n>=n11B) &(n<n22B))+y3fB.*(n>=n22B);
% yBfundamental=y1fB.*(t<T11B)-y2fB.*((t>=T11B) &(t<T22B))+y3fB.*(t>=T22B);

yCfundamental=y1fC.*(n<n11C)-y2fC.*((n>=n11C) &(n<n22C))+y3fC.*(n>=n22C);
% yCfundamental=y1fC.*(t<T11C)-y2fC.*((t>=T11C) &(t<T22C))+y3fC.*(t>=T22C);

#####
% figure,plot(t,yAt,t,yBt,t,yCt) %Grafica de la señal sintetica en tiempo
figure,plot(n,yAn,n,yBn,n,yCn) % discreto
axis([0 length(yCn)+1 -max(yCn)-5 max(yCn)+5])

```

## ANEXO D3. GENERADOR DE SEÑALES SINTÉTICAS EN TIEMPO CONTÍNUO.

### a) Monofásico

```
function [y2t] = generador_senal_conformidad_un_tramo_continuo(t)

Nsample=128;
Tm=(1/60)/Nsample;
udin=127/sqrt(2);
#####
%__Frecuencia para cada tramo__
Fsist1=60;
Fsist2=60;
Fsist3=60;
#####
%__amplitud de la fundamental para cada tramo
Amax1=-127;
Amax2=-40;
Amax3=127;
#####
Tfinal=0.8;
T11=0.15;      %final del primer intervalo, inicio del segundo intervalo
n11=fix(T11/Tm);
T22=Tfinal-0.15; %final del segundo intervalo, inicio del tercer intervalo
n22=fix(T22/Tm);
#####
%Varación de la amplitud de las componentes armónicas e interarmónicas de
%las señales sintéticas creadas bajo el estándar IEC 61000-4-30:2003.
%las constantes armonico_x hacen referencia a las amplitudes de las
%componentes que se encuentran en los ensayos 2 y 3.
%La variable ax_armonico indica la componente armónica incluida en la señal
%sintética. en los ensayos 2 y 3 estas componentes son distintas por tanto
%solo es cambiar el valor de la componente armónica. igual se aplica para
%la componente interarmónica.

armonico_3=(0/100)*udin;
armonico_5=(0/100)*udin;
armonico_29=(0/100)*udin;
interarmonico_a1=(0/100)*udin;
a3_armonico=3;
a5_armonico=5;
a29_armonico=29;
a1_interarmonico=7.5;

#####
%Señal por tramos de la señal sintética en el dominio del tiempo.
retardo1=35*Tm;
retardo2=45*Tm;

y1t=(Amax1*sin(2*pi*Fsist1*(t+retardo1)))+(armonico_3*sin(2*pi*a3_armonico*Fsist1*t))+(armonico_5*sin(2*pi*a5_armonico
*Fsist1*t))+(armonico_29*sin(2*pi*a29_armonico*Fsist1*t))+(interarmonico_a1*sin(2*pi*a1_interarmonico*Fsist1*t));
y2t=(Amax2*sin(2*pi*Fsist2*(t+retardo2)))+(armonico_3*sin(2*pi*a3_armonico*Fsist2*t))+(armonico_5*sin(2*pi*a5_armonico
*Fsist2*t))+(armonico_29*sin(2*pi*a29_armonico*Fsist2*t))+(interarmonico_a1*sin(2*pi*a1_interarmonico*Fsist2*t));
y3t=(Amax3*sin(2*pi*Fsist3*t))+(armonico_3*sin(2*pi*a3_armonico*Fsist3*t))+(armonico_5*sin(2*pi*a5_armonico*Fsist3*t))+(
armonico_29*sin(2*pi*a29_armonico*Fsist3*t))+(interarmonico_a1*sin(2*pi*a1_interarmonico*Fsist3*t));
mod_flicker=35;
```

```

amp_flicker=0;
%Señal completa
#####
yt=y1t.*(t<T11)-y2t.*((t>=T11)&(t<T22))+y3t.*(t>=T22);%+(amp_flicker*square(2*mod_flicker*pi*t));
y2t=yt.*yt; %Señal elevada al cuadrado, ya que se enciesita para determinar
      %la tensión rms de la señal tmeportal.

```

## b) Trifásico

```

FASE A
function [y2At] = generador_senal_conformidad_un_tramo_continuo_trifasico_A(t)

udin=127/sqrt(2);
%La señal sintética de prueba esta dividida en tres tramos con el fin de
%crear el evento en el segundo tramo.
#####
%__Frecuencia para cada tramo__
Fsist1=60;
Fsist2=60;
Fsist3=60;
#####
%__amplitud de la fundamental para cada tramo
Amax1=127;
Amax2=9;
Amax3=127;
#####
Tfinal=0.2; %Tiempo final de la señal sintética
T11A=0.03; %final del primer intervalo, inicio del segundo intervalo
T22A=Tfinal-0.06; %final del segundo intervalo, inicio del tercer intervalo

#####
%Varación de la amplitud de las componentes armónicas e interarmónicas de
%las señales sintéticas creadas bajo el estándar IEC 61000-4-30:2003.
%las constantes armonico_x hacen referencia a las amplitudes de las
%componentes que se encuentran en los ensayos 2 y 3.
%La variable ax_armonico indica la componente armónica incluida en la señal
%sintética. en los ensayos 2 y 3 estas componentes son distintas por tanto
%solo es cambiar el valor de la componente armónica. igual se aplica para
%la componente interarmónica.

armonico_3=(3/100)*udin;
armonico_5=(3/100)*udin;
armonico_29=(3/100)*udin;
interarmonico_a1=(0/100)*udin;
a3_armonico=3;
a5_armonico=5;
a29_armonico=29;
a1_interarmonico=7.5;

#####
%Señal por tramos de la señal sintética en el dominio del tiempo.

```

```

y1tA=(Amax1*sin(2*pi*Fsist1*t))+(armonico_3*sin(2*pi*a3_armonico*Fsist1*t))+(armonico_5*sin(2*pi*a5_armonico*Fsist1*t))
+(armonico_29*sin(2*pi*a29_armonico*Fsist1*t))+(interarmonico_a1*sin(2*pi*a1_interarmonico*Fsist1*t));
y2tA=(Amax2*sin(2*pi*Fsist2*t))+(armonico_3*sin(2*pi*a3_armonico*Fsist2*t))+(armonico_5*sin(2*pi*a5_armonico*Fsist2*t))
+(armonico_29*sin(2*pi*a29_armonico*Fsist2*t))+(interarmonico_a1*sin(2*pi*a1_interarmonico*Fsist2*t));
y3tA=(Amax3*sin(2*pi*Fsist3*t))+(armonico_3*sin(2*pi*a3_armonico*Fsist3*t))+(armonico_5*sin(2*pi*a5_armonico*Fsist3*t))
+(armonico_29*sin(2*pi*a29_armonico*Fsist3*t))+(interarmonico_a1*sin(2*pi*a1_interarmonico*Fsist3*t));

```

```

mod_flicker=35;
amp_flicker=0;
%Señal completa
#####
yAt=y1tA.*(t<T11A)-y2tA.*(t>=T11A)&(t<T22A))+y3tA.*(t>=T22A);%+(amp_flicker*sqrt(2*mod_flicker*pi*t));

```

```

y2At=yAt.*yAt; %Señal elevada al cuadrado, ya que se encesa para determinar
%la tensión rms de la señal ttemporal.
plot(t,y2At)

```

FASE B

```
function [y2Bt] = generador_senal_conformidad_un_tramo_continuo_trifasico_B(t)
```

```

udin=127/sqrt(2);
%La señal sintética de prueba esta dividida en tres tramos con el fin de
%crear el evento en el segundo tramo.
#####
%__Frecuencia para cada tramo__
Fsist1=60;
Fsist2=60;
Fsist3=60;
#####
%__amplitud de la fundamental para cada tramo
Amax1=127;
Amax2=9;
Amax3=127;
#####

```

```

Tfinal=0.2; %Tiempo final de la señal sintética
T11B=0.06; %final del primer intervalo, inicio del segundo intervalo
T22B=Tfinal-0.06; %final del segundo intervalo, inicio del tercer intervalo

```

```

#####
%Varación de la amplitud de las componentes armónicas e interarmónicas de
%las señales sintéticas creadas bajo el estándar IEC 61000-4-30:2003.
%las constantes armonico_x hacen referencia a las amplitudes de las
%componentes que se encuentran en los ensayos 2 y 3.
%La variable ax_armonico indica la componente armónica incluida en la señal
%sintética. en los ensayos 2 y 3 estas componentes son distintas por tanto
%solo es cambiar el valor de la componente armónica. igual se aplica para
%la componente interarmónica.

```

```

armonico_3=(3/100)*udin;
armonico_5=(3/100)*udin;
armonico_29=(3/100)*udin;
interarmonico_a1=(0/100)*udin;
a3_armonico=3;
a5_armonico=5;
a29_armonico=29;
a1_interarmonico=7.5;

```

```

#####
%Señal por tramos de la señal sintética en el dominio del tiempo.

```

```
desface=0.0055555553333333;
```

```

y1tB=(Amax1*sin(2*pi*Fsist1*(t-desface)))+(armonico_3*sin(2*pi*a3_armonico*Fsist1*(t-
desface)))+(armonico_5*sin(2*pi*a5_armonico*Fsist1*(t-desface)))+(armonico_29*sin(2*pi*a29_armonico*Fsist1*(t-
desface)))+(interarmonico_a1*sin(2*pi*a1_interarmonico*Fsist1*(t-desface)));
y2tB=(Amax2*sin(2*pi*Fsist2*(t-desface)))+(armonico_3*sin(2*pi*a3_armonico*Fsist2*(t-
desface)))+(armonico_5*sin(2*pi*a5_armonico*Fsist2*(t-desface)))+(armonico_29*sin(2*pi*a29_armonico*Fsist2*(t-
desface)))+(interarmonico_a1*sin(2*pi*a1_interarmonico*Fsist2*(t-desface)));
y3tB=(Amax3*sin(2*pi*Fsist3*(t-desface)))+(armonico_3*sin(2*pi*a3_armonico*Fsist3*(t-
desface)))+(armonico_5*sin(2*pi*a5_armonico*Fsist3*(t-desface)))+(armonico_29*sin(2*pi*a29_armonico*Fsist3*(t-
desface)))+(interarmonico_a1*sin(2*pi*a1_interarmonico*Fsist3*(t-desface)));

```

```

mod_flicker=35;
amp_flicker=0;
%Señal completa
#####

```

```

yBt=y1tB.*(t<T11B)-y2tB.*(t>=T11B)&(t<T22B))+y3tB.*(t>=T22B);%+(amp_flicker*sqrt(2*mod_flicker*pi*t));

```

```

y2Bt=yBt.*yBt; %Señal elevada al cuadrado, ya que se encenita para determinar
plot(t,y2Bt)
%la tensión rms de la señal tmeportal.

```

FASE C

```

function [y2Ct] = generador_senal_conformidad_un_tramo_continuo_trifasico_C(t)

```

```

udin=127/sqrt(2);
%La señal sintética de prueba esta dividida en tres tramos con el fin de
%crear el evento en el segundo tramo.
#####
%__Frecuencia para cada tramo__
Fsist1=60;
Fsist2=60;
Fsist3=60;
#####

```

```

%__amplitud de la fundamental para cada tramo
Amax1=127;
Amax2=9;
Amax3=127;
#####
Tfinal=0.2; %Tiempo final de la señal sintética
T11C=0.05; %final del primer intervalo, inicio del segundo intervalo
T22C=Tfinal-0.06; %final del segundo intervalo, inicio del tercer intervalo

```

```

#####
%Varación de la amplitud de las componentes armónicas e interarmónicas de
%las señales sintéticas creadas bajo el estándar IEC 61000-4-30:2003.
%las constantes armonico_x hacen referencia a las amplitudes de las
%componentes que se encuentran en los ensayos 2 y 3.
%La variable ax_armonico indica la componente armónica incluida en la señal
%sintética. en los ensayos 2 y 3 estas componentes son distintas por tanto
%solo es cambiar el valor de la componente armónica. igual se aplica para
%la componente interarmónica.

```

```

armonico_3=(3/100)*udin;
armonico_5=(3/100)*udin;
armonico_29=(3/100)*udin;
interarmonico_a1=(0/100)*udin;
a3_armonico=3;
a5_armonico=5;
a29_armonico=29;
a1_interarmonico=7.5;

```

```

#####

```

```

%Señal por tramos de la señal sintética en el dominio del tiempo.
desface=0.005555555333333333;
y1tC=(Amax1*sin(2*pi*Fsist1*(t+desface)))+(armonico_3*sin(2*pi*a3_armonico*Fsist1*(t+desface)))+(armonico_5*sin(2*pi*a
5_armonico*Fsist1*(t+desface)))+(armonico_29*sin(2*pi*a29_armonico*Fsist1*(t+desface)))+(interarmonico_a1*sin(2*pi*a1_
interarmonico*Fsist1*(t+desface)));
y2tC=(Amax2*sin(2*pi*Fsist2*(t+desface)))+(armonico_3*sin(2*pi*a3_armonico*Fsist2*(t+desface)))+(armonico_5*sin(2*pi*a
5_armonico*Fsist2*(t+desface)))+(armonico_29*sin(2*pi*a29_armonico*Fsist2*(t+desface)))+(interarmonico_a1*sin(2*pi*a1_
interarmonico*Fsist2*(t+desface)));
y3tC=(Amax3*sin(2*pi*Fsist3*(t+desface)))+(armonico_3*sin(2*pi*a3_armonico*Fsist3*(t+desface)))+(armonico_5*sin(2*pi*a
5_armonico*Fsist3*(t+desface)))+(armonico_29*sin(2*pi*a29_armonico*Fsist3*(t+desface)))+(interarmonico_a1*sin(2*pi*a1_
interarmonico*Fsist3*(t+desface)));

mod_flicker=35;
amp_flicker=0;
%Señal completa
%#####

yCt=y1tC.*(t<T11C)-y2tC.*(t>=T11C)&(t<T22C))+y3tC.*(t>=T22C);%+(amp_flicker*square(2*mod_flicker*pi*t));

y2Ct=yCt.*yCt; %Señal elevada al cuadrado, ya que se encesa para determinar
%la tensión rms de la señal tmeportal.

```

## ANEXO D4. MEDIDOR RMS DE 1/2 CICLO EN TIEMPO CONTINUO.

### a) Monofásico

```

Valor_rms_analogico=0;
l=0;
a1=0;
minimo=0;
limite_superior=0;
limite_inferior=0;
    signo1t=0;
    signo2t=0;

%#####

t=40;
n1=0;
tolerancia=1e-10;
%Se hace el procedimeito del algoritmo para la medicin de perturbaciones
%transitorias de tensión con base en estándar 61000-4-30:2003 pero en
%tiempo continuo.
while ((t)<(length(ytfundamental)-1000)),

    while signo1t==signo2t,
        if ytfundamental(t)==0,
            cero1t=1;
        else
            cero1t=0;
        end
        if ytfundamental(t+1)==0,
            cero2t=1;
        else
            cero2t=0;
        end
        if (cero1t & cero2t)==1,

```

```

        senal_cerot=1;
    else
        senal_cerot=0;
    end
    %-----
    %Si la señal es cero para dos muestras consecutivas
    %se refresca la señal en 64. es decir, se asume que
    %ocurrió una interrupcion total.
    if senal_cerot==1,
        t=t+834;
        break
    %-----
    else
        %Si no existe una interrupción total entonces se
        %busca el primer cruce por cero.
        signo1t=sign(ytfundamental(t));
        signo2t=sign(ytfundamental(t+1));
        t=t+1;
    end
end
    signo1t=0;
    signo2t=0;
n1=n1+1;
%Se determina la tensión rms comenzando en el cruce por cero de la
%fundamental. La señal de salida del archivo generador_senal_conformidad_un_tramo_continuo
%es usada por el comando quad para realizar la integral.
    t1=t*1e-5;
    Valor_rms_analogico(n1) =
sqrt((1/(1/Fsist2))*quad(@generador_senal_conformidad_un_tramo_continuo,t1,t1+1/Fsist2,tolerancia));

```

end

%SI OCURRE UN HUECO O SOBRETENSIÓN, SELECCIONE LA EXTRACCIÓN D EDATOS  
%CORRESPONDIENTE

%En esta seccion se busca la tensión minima del hueco de tensión

```

l1 = find(Valor_rms_analogico>0);
a1=Valor_rms_analogico(l1);
minimo_hueco=min(a1)
limite_superior1=minimo_hueco+(0.2/100)*udin
limite_inferior1=minimo_hueco-(0.2/100)*udin

```

%En esta sección se buscan la amplitud maxima de la sobretensión.

```

% l1 = find(Valor_rms_analogico>0);
% a1=Valor_rms_analogico(l1);
% maxima_sobretension=max(a1)
% limite_superior1=maxima_sobretension+(0.2/100)*udin
% limite_inferior1=maxima_sobretension-(0.2/100)*udin

```

#####

```

save Prueba_166_datos.txt Tfinal nfinal Tm1 Tm2 Tm3 T11 n11 T22 n22 Tm Nsample udin Fsist1 Fsist2 Fsist3 Amax1
Amax2 Amax3 armonico_3 armonico_5 armonico_29 interarmonico_a1 tolerancia a3_armonico a5_armonico a29_armonico
a1_interarmonico -ascii

```

## b) Trifásico

```
Valor_rms_analogico=0;
l=0;
a1=0;
minimo=0;
limite_superior=0;
limite_inferior=0;
    signo1tA=0;
    signo2tA=0;
    signo1tB=0;
    signo2tB=0;
    signo1tC=0;
    signo2tC=0;
Valor_rms_analogicoA=0;
Valor_rms_analogicoB=0;
Valor_rms_analogicoC=0;
#####
ta=40;
n1A=0;
n1B=0;
n1C=0;
tolerancia=1e-10;

%Se hace el procedimiento del algoritmo para la medición de perturbaciones
%transitorias de tensión con base en estándar 61000-4-30:2003 pero en
%tiempo continuo.
while ((ta)<(length(ytAfundamental)-1900)),

    while signo1tA==signo2tA,
        if ytAfundamental(ta)==0,
            cero1tA=1;
        else
            cero1tA=0;
        end
        if ytAfundamental(ta+1)==0,
            cero2tA=1;
        else
            cero2tA=0;
        end
        if (cero1tA & cero2tA)==1,
            senal_cerotA=1;
        else
            senal_cerotA=0;
        end
        %-----
        %Si la señal es cero para dos muestras consecutivas
        %se refresca la señal en 64. es decir, se asume que
        %ocurrió una interrupción total.
        if senal_cerotA==1,
            ta=ta+834;
            break
        %-----
        else
            %Si no existe una interrupción total entonces se
            %busca el primer cruce por cero.
            signo1tA=sign(ytAfundamental(ta));
            signo2tA=sign(ytAfundamental(ta+1));
            ta=ta+1;
        end
    end
```

```

end

    signo1tA=0;
    signo2tA=0;
n1A=n1A+1;
%Se determina la tensión rms comenzando en el cruce por cero de la
%fundamental. La señal de salida del archivo generador_senal_conformidad_un_tramo_continuo
%es usada por el comando quad para realizar la integral.
    tA=ta*1e-5;
    Valor_rms_analogicoA(n1A)=
sqrt((1/(1/Fsist2))*quad(@generador_senal_conformidad_un_tramo_continuo_trifasico_A,tA,tA+1/Fsist2,tolerancia));
end
% %
tb=40;
while ((tb)<(length(ytAfundamental)-1800)),

    while signo1tB==signo2tB,
        if ytBfundamental(tb)==0,
            cero1tB=1;
        else
            cero1tB=0;
        end
        if ytBfundamental(tb+1)==0,
            cero2tB=1;
        else
            cero2tB=0;
        end
        if (cero1tB & cero2tB)==1,
            senal_cerotB=1;
        else
            senal_cerotB=0;
        end
        %-----
        %Si la señal es cero para dos muestras consecutivas
        %se refresca la señal en 64. es decir, se asume que
        %ocurrió una interrupción total.
        if senal_cerotB==1,
            tb=tb+834;
            break
        %-----
        else
            %Si no existe una interrupción total entonces se
            %busca el primer cruce por cero.
            signo1tB=sign(ytBfundamental(tb));
            signo2tB=sign(ytBfundamental(tb+1));
            tb=tb+1;
        end
    end

    signo1tB=0;
    signo2tB=0;
n1B=n1B+1;
%Se determina la tensión rms comenzando en el cruce por cero de la
%fundamental. La señal de salida del archivo generador_senal_conformidad_un_tramo_continuo
%es usada por el comando quad para realizar la integral.
    tB=tb*1e-5;
    Valor_rms_analogicoB(n1B) =
sqrt((1/(1/Fsist2))*quad(@generador_senal_conformidad_un_tramo_continuo_trifasico_B,tB,tB+1/Fsist2,tolerancia));
end
% %
tc=40;
while ((tc)<(length(ytAfundamental)-1800)),

    while signo1tC==signo2tC,
        if ytCfundamental(tc)==0,

```

```

        cero1tC=1;
    else
        cero1tC=0;
    end
    if ytCfundamental(tc+1)==0,
        cero2tC=1;
    else
        cero2tC=0;
    end
    if (cero1tC & cero2tC)==1,
        senal_cerotC=1;
    else
        senal_cerotC=0;
    end
    %-----
    %Si la señal es cero para dos muestras consecutivas
    %se refresca la señal en 64. es decir, se asume que
    %ocurrió una interrupcion total.
    if senal_cerotC==1,
        tc=tc+834;
        break
    %-----
    else
        %Si no existe una interrupción total entonces se
        %busca el primer cruce por cero.
        signo1tC=sign(ytCfundamental(tc));
        signo2tC=sign(ytCfundamental(tc+1));
        tc=tc+1;
    end
end
    signo1tC=0;
    signo2tC=0;
n1C=n1C+1;
    %Se determina la tensión rms comenzando en el cruce por cero de la
    %fundamental. La señal de salida del archivo generador_senal_conformidad_un_tramo_continuo
    %es usada por el comando quad para realizar la integral.
    tC=tc*1e-5;
    Valor_rms_analogicoC(n1C) =
sqrt((1/(1/Fsist2))*quad(@generador_senal_conformidad_un_tramo_continuo_trifasico_C,tC,tC+1/Fsist2,tolerancia));
end

%_____

```

**%SI OCURRE UN HUECO O SOBRETENSIÓN, SELECCIONE LA EXTRACCIÓN DE DATOS  
%CORRESPONDIENTE**

**%En esta seccion se busca la tensión minima del hueco de tensión**

```

minimo_hueco=udin;
for k=1:length(Valor_rms_analogicoA)-1,

    if Valor_rms_analogicoA(k)>Valor_rms_analogicoB(k),
        minimo_hueco_actual=Valor_rms_analogicoB(k);
    else
        minimo_hueco_actual=Valor_rms_analogicoA(k);
    end
    if Valor_rms_analogicoC(k)<minimo_hueco,

```

```

    minimo_hueco_actual=Valor_rms_analogicoC(k);
end

if minimo_hueco_actual<minimo_hueco,
    minimo_hueco=minimo_hueco_actual;
end
end
minimo_hueco

%En esta sección se buscan la amplitud maxima de la sobretensión.
% I1 = find(Valor_rms_analogico>0);
% a1=Valor_rms_analogico(I1);
% maxima_sobretension=max(a1)
% limite_superior1=maxima_sobretension+(0.2/100)*udin
% limite_inferior1=maxima_sobretension-(0.2/100)*udin

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

save Prueba_005_datos.txt Nsample Tfinal T11A T22A T11B T22B T11C T22C udin Fsist1 Fsist2 Fsist3 Amax1 Amax2
Amax3 armonico_3 armonico_5 armonico_29 interarmonico_a1 tolerancia a3_armonico a5_armonico a29_armonico
a1_interarmonico -ascii

```