



Universidad Tecnológica Nacional  
Facultad Regional Bahía Blanca

**Ingeniería Electrónica**

**Proyecto Final**

# **MEDIDOR DE CALIDAD ENERGETICA**

**Autores**

**Acosta Hernan Adalberto**

**Pérez Artica Ramiro José**

**Trovato Joaquín Ezequiel**

**Tutores**

**Mg Guillermo Friedrich**

**Ing. Adrián Laiuppa**

Bahía Blanca | 13 de Marzo de 2024

## **AGRADECIMIENTOS**

Este proyecto no hubiese sido posible sin la formación académica que hemos recibido durante nuestros años en la Facultad Regional Bahía Blanca y el acompañamiento de los docentes que han demostrado el interés y la vocación de enseñar. Compartimos el esfuerzo de trabajar en paralelo a estudiar esta carrera y damos fé de que, si bien por momentos nos hemos visto superados, esta Facultad y nuestras familias siempre nos acompañaron para cumplir este sueño tan anhelado durante años.

## RESUMEN

Este estudio se enfoca en el desarrollo de un dispositivo para la medición y análisis de la calidad de la energía eléctrica, con especial atención en el contenido armónico, utilizando el microcontrolador ESP32 por su eficiente conectividad a Internet y capacidad de procesamiento.

La motivación detrás de este proyecto radica en la necesidad de superar las limitaciones de las herramientas comerciales de diagnóstico eléctrico, que frecuentemente omiten la medición de armónicos.

Se adoptó una metodología centrada en la implementación de algoritmos de Transformada Rápida de Fourier (FFT) para el análisis detallado de armónicos y la utilización de Google Spreadsheets para el almacenamiento y análisis remoto de datos.

Los resultados obtenidos demuestran que el dispositivo no solo cumple, sino que supera las expectativas iniciales, proporcionando mediciones precisas y ofreciendo una visión exhaustiva del consumo energético.

Esto permite a los usuarios tomar decisiones informadas para optimizar el uso de energía. Además, el diseño modular del sistema asegura su escalabilidad.

Las conclusiones destacan el impacto significativo del proyecto en la ingeniería electrónica, al ofrecer una herramienta integral que sirva de apoyo para la mejora de eficiencia energética, la seguridad operacional y el cumplimiento normativo, estableciendo un precedente para el futuro desarrollo de sistemas de monitorización energética.



Figura 1 Imagen del frente del equipo desarrollado

## **PALABRAS CLAVE**

Consumo energético; Eficiencia energética; Contenido armónico; Internet de las Cosas;

# Índice

Índice.....	4
Índice de Figuras .....	6
Índice de Tablas.....	8
1. Introducción .....	9
1.2. Objetivos .....	12
2. Desarrollo .....	13
2.1 Diseño e implementación .....	15
2.1.1 Placa Madre.....	15
2.1.2 ESP-32.....	16
2.1.3 Display ILI9486 320x480.....	17
2.1.4 Medición de tensión SI8920BC .....	17
2.1.5 Fuentes de alimentación 5V.....	18
2.1.6 Pinza Sensor De Corriente 30a Sct013 .....	19
2.1.7 Diseño del Gabinete .....	21
2.2 Tensión circuito.....	22
2.2.1 Análisis en DC para la adquisición de tensión .....	23
2.2.2 Análisis en AC para la adquisición de tensión .....	24
2.3 Corriente circuito .....	25
2.3.1 Análisis en DC para la adquisición de corriente .....	25
2.3.2 Análisis en AC para la adquisición de corriente .....	26
3. Gestión de memoria y de procesos en sistemas embebidos.....	26
3.1 ESP32 – Arquitectura.....	26
3.2 Uso de FREERTOS .....	28
3.3 Declaración de funciones/tareas .....	29
3.3 Uso de Interrupciones.....	31
3.4 Muestreo periódico.....	31

3.5 Método de detección de cruce por cero y cálculo de frecuencia .....	33
3.6 Cálculo de valores rms .....	35
3.7 Cálculo de espectro y THD .....	36
3.7.1 Transformada de Fourier .....	37
3.7.2 Transformada Discreta de Fourier (DFT).....	37
3.7.3 Algoritmo Cooley-Tukey, método de la mariposa.....	39
3.8 Filtros Digitales Butterworth HPF y LPF orden 2 .....	47
3.8.1 Propiedades de los filtros .....	47
3.8.2 Transformación Bilineal (Método Tustin): .....	51
3.8.3 Implementación Filtro Pasabajo orden 2 .....	54
3.8.4 Implementación filtro pasa altos HPF de orden 2.....	55
3.9 Entorno gráfico y uso de librerías para gráficos.....	57
4. Posteo periódico Google Spreadsheets .....	58
5. Redes de Datos. Protocolos de Comunicación ESP32 .....	61
5.1 Comunicación WiFi.....	62
5.2 Protocolo TCP/IP .....	62
5.3 Administrador de conexión inalámbrica .....	63
6. Ensayos realizados.....	65
6.1 Pruebas de Medición de Tensión AC.....	66
6.2 Pruebas de Medición de Corriente AC .....	67
6.3 Pruebas de Medición Armónicos .....	68
7. Conclusión .....	73
8. Anexos .....	74
9. Referencias Bibliográficas .....	77
9.2 Normas técnicas .....	77

## Índice de Figuras

Figura 1 Imagen del frente del equipo desarrollado .....	2
Figura 2 Representación de onda senoidal pura y distorsionada.....	10
Figura 3 Representación de armónicos como múltiplo de la frecuencia fundamental.....	11
Figura 4 Diagrama en bloques funcionamiento general del código implementado .....	14
Figura 5 Diseño Placa Madre utilizada vista superior.....	15
Figura 6 Diseño Placa Madre utilizada vista inferior .....	15
Figura 7 ESP32 con descripción de pines de salida/entrada .....	16
Figura 8 Display utilizado para el equipo.....	17
Figura 9 Encapsulado DIP-8 de operacional Si8920 .....	17
Figura 10 Esquemático completo circuito de adaptación de niveles .....	18
Figura 11 Diagrama en Bloque Funcional SI8920.....	18
Figura 12 Fuente Switching 5V utilizada en el equipo.....	19
Figura 13 Pinza utilizada para la medición de corriente.....	20
Figura 14 Circuito adaptador de niveles y filtrado. ....	20
Figura 15 Esquema de conexión plug Pinza amperométrica.....	21
Figura 16 Gabinete diseñado en impresora 3D vista superior .....	21
Figura 17 Gabinete diseñado en impresora 3D vista lateral .....	22
Figura 18 Circuito utilizado para la adaptación de señal tensión a valores lógicos.....	22
Figura 19 Esquemático filtro de RC entrada .....	24
Figura 20 Simulación respuesta del filtro RC implementado en el circuito.....	25
Figura 21 Esquemático adaptación de niveles pinza.....	25
Figura 22 Descripción de arquitectura ESP32 .....	27
Figura 23 Método de detección de cruce por cero .....	34
Figura 24 Ejemplo genérico de etapas mariposa de N=8.....	43
Figura 25 Representación diagrama en bloques filtro dos ramas.....	49
Figura 26 Representación canónica diagrama en bloques filtro .....	49
Figura 27 Display con explicación de contenido.....	57
Figura 28 Código implementado en AppScript Google .....	60
Figura 29 Captura de pantalla de Google Spreadsheets .....	61
Figura 30 Gráficos históricos de hoja de datos Google Spreadsheets.....	61
Figura 31 Captura contenido armónico hoja de datos.....	61
Figura 32 Código QR para acceder a Hoja de Datos Google Spreadsheets .....	61
Figura 33 SSID de red de WiFi en modo AP .....	64
Figura 34 Pantalla principal de portal de configuración de conexión WiFi .....	64
Figura 35 Selección de SSID de red a conectar y escritura de contraseña .....	65
Figura 36 Guardado de credenciales de conexión exitosa .....	65
Figura 37 Pinza de medición voltaje en comparativa con equipo. ....	66
Figura 38 Gráfica de dispersión entre tensión medida en pinza y equipo. ....	67
Figura 39 Curva de medición corriente .....	68
Figura 40 Medición de corriente comparativa entre pinza y equipo en tablero eléctrico .....	68
Figura 41 Comparación armónicos equipo y osciloscopio Rigol .....	69
Figura 42 FFT Osciloscopio fundamental y 6to armónico THD 5.0 %.....	69
Figura 43 FFT Equipo fundamental y 6to armónico THD 3.0 %.....	70
Figura 44 FFT Equipo fundamental y 3er armónico THD 3%.....	71
Figura 45 FFT Osciloscopio fundamental y armónico 6 y 12, THD 3,8 %.....	71

Figura 46 FFT Equipo fundamental y armónico 6 y 12 THD 3,7 %..... 72



## Índice de Tablas

Tabla 1 Contenido Armónico de Voltajes menores a 1000v por EN 50160.....	12
Tabla 2 Valores de escalado de señal hacia ADC tensión. ....	24
Tabla 3 Asignación de pines ESP32 para conexión de ILI9486 .....	28
Tabla 4 Representación de método de inversión de bits FFT .....	45
Tabla 5 Comparación de medidas de Tensión UT203+ y equipo, cálculo de error.....	66
Tabla 6 Comparación de medidas Corriente UT203+ y equipo, cálculo de error.....	67
Tabla 7 Comparación de medidas armónicos Rigol DS2072A y equipo, cálculo de error .....	72
Tabla 8 Límites de Emisión Resolución ENRE 099/1997.....	74
Tabla 9 Límites de Emisión Resolución ENRE 099/1997.....	75
Tabla 10 Límites de Emisión Resolución ENRE 184/2000.....	76
Tabla 11 Niveles de Compatibilidad tensiones armónicas IEC 61000 -2 -2 .....	76

## 1. Introducción

La creciente dependencia de la energía eléctrica en la vida cotidiana y la integración de una multitud de dispositivos eléctricos y electrónicos en nuestros hogares, plantean desafíos significativos para la gestión eficiente y responsable del consumo energético. Los dispositivos modernos, especialmente aquellos que utilizan fuentes de alimentación switching, como luminarias LED, convertidores AC-DC y routers, introducen contenido armónico en las redes eléctricas domiciliarias, afectando negativamente la calidad de la energía y, por ende, el funcionamiento de las instalaciones eléctricas.

La presencia de armónicos en la red puede provocar una serie de problemas técnicos, incluyendo sobrecalentamiento de equipos, interferencias en dispositivos electrónicos, incremento en las pérdidas de transmisión y reducción de la vida útil de los componentes de la instalación. Esto no solo implica un desafío para la estabilidad y seguridad de las redes eléctricas domésticas, sino que también representa una barrera para alcanzar un consumo energético más eficiente y sostenible.

Frente a este panorama, la necesidad de contar con un sistema avanzado de monitoreo y análisis se hace evidente. La tecnología del Internet de las Cosas (IoT) emerge como una solución potencialmente transformadora, permitiendo la implementación de dispositivos inteligentes capaces de medir y analizar el consumo de energía y el contenido armónico en tiempo real. Este dispositivo puede ofrecer una visión detallada de cómo la energía es consumida y afectada por los distintos aparatos eléctricos y electrónicos en el hogar.

Además, la capacidad del sistema para almacenar datos históricos en la nube abre la puerta a un análisis profundo y a largo plazo del consumo de energía. Esto no solo facilita la identificación de patrones y tendencias de consumo, sino que también permite la detección temprana de problemas potenciales en la instalación eléctrica, contribuyendo así a la prevención de fallos y a la optimización del consumo energético.

En resumen, la adopción de tecnologías de monitoreo y análisis basadas en IoT representa una estrategia clave para abordar los desafíos asociados con la calidad de la energía en las redes eléctricas domésticas. Al proporcionar una comprensión más clara y detallada del consumo energético y sus efectos, estos sistemas pueden jugar un papel crucial en la promoción de un uso más eficiente, responsable y sostenible de la energía eléctrica en el mundo contemporáneo.

### **La causa de los armónicos.**

La forma de onda ideal de la fuente de alimentación de CA es una onda sinusoidal pura de 50 Hz sin ningún ruido. Pero desafortunadamente, en realidad, es difícil tener una fuente de alimentación de CA ideal. Esto se debe a que, a veces, la carga que conectamos causa distorsión en la forma de onda de la fuente de alimentación de CA.

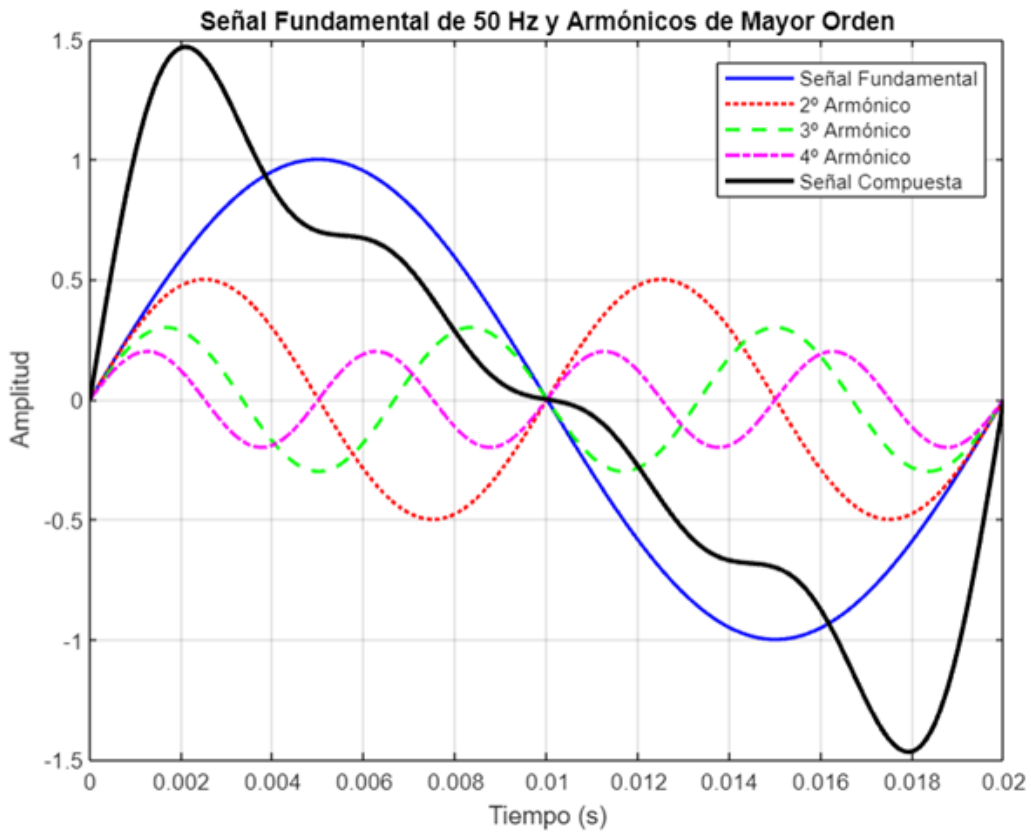


Figura 2 Representación de onda senoidal pura y distorsionada

La distorsión es causada por dispositivos de conversión de energía como inversores para motores eléctricos y tiristores industriales. Un dispositivo comúnmente llamado VFD (Variable Frequency Drive o Variador de Frecuencia es uno de ellos. El inversor regula la velocidad del motor convirtiendo el voltaje y la frecuencia a valores adecuados. Esto permite afinar el comportamiento del equipo. Pueden ayudar a reducir el consumo de energía y mejorar el control de salida, pero también son fuentes de armónicos.

### ¿Qué son los armónicos?

Los armónicos son múltiplos enteros de la frecuencia fundamental. Por ejemplo, si la frecuencia fundamental es 50 Hz, la frecuencia de segundo orden es 100 Hz, la de tercer orden 150 Hz y la de cuarto orden 200 Hz.

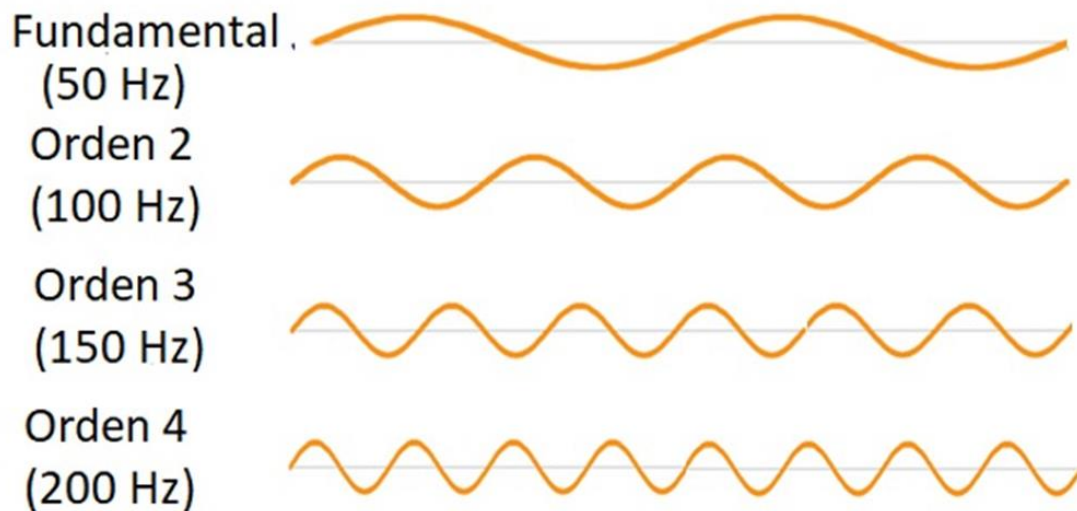


Figura 3 Representación de armónicos como múltiplo de la frecuencia fundamental

Los armónicos causan distorsión en la forma de onda del voltaje y corriente, lo que, en el peor de los casos, puede provocar accidentes como el sobrecalentamiento de las instalaciones o daños en los equipos. Por razones de seguridad, es mejor tener la menor cantidad posible de armónicos.

#### ¿Qué es THD (distorsión armónica total)?

THD es la relación entre el componente armónico total y el componente fundamental. En concreto, se expresa mediante la siguiente fórmula de cálculo. El numerador es la suma de los valores RMS de cada orden a partir del segundo orden. El denominador es el valor ~~rms~~ de la forma de onda fundamental.

$$THD = \frac{\sum_{n=2}^{\infty} V_n}{V_1} * 100\%$$

El valor de la Distorsión Armónica Total (THD) sirve como un indicador claro y sintético para identificar los armónicos existentes en la señal de una fuente de alimentación. Este parámetro resume en un único valor el grado de distorsión que la forma de onda original ha sufrido debido a la presencia de armónicos, facilitando así su comprensión y análisis. Por tanto, el THD emerge como un criterio crucial para la evaluación de la presencia y el impacto de los armónicos en cualquier sistema de alimentación eléctrica.

#### Estándares relacionados con THD

Estándares como IEC61000-3-2 (Comisión Electrotécnica Internacional, 2018), EN50160 (Comité Europeo de Normalización, 2010) e IEEE 519 (Instituto de Ingenieros Eléctricos y Electrónicos, 2014) especifican el límite para el contenido armónico y/o THD de cada orden. Estos límites se clasifican en varias categorías, como tensión de alimentación, corriente nominal del instrumento, etc. El valor THD en el voltaje es típicamente del 1% a menos del 8%. En cuanto a la corriente, el valor puede superar el 100% cuando hay muchos armónicos. La Tabla 1 muestra los valores recomendados para las normas EN 50160.

Tabla 1 Contenido Armónico de Voltajes menores a 1000v por EN 50160

Ordenar	Valor límite
THD	8.0%
2	2.0%
3	5.0%
4	1.0%
5	6.0%
6	0.5%
7	5.0%
9	1.5%
11	3.5%

### Efecto de los armónicos

Los armónicos pueden afectar no solo el sitio donde se usa la energía, sino también las instalaciones que lo rodean. Esto es especialmente cierto en lugares como fábricas y plantas donde se utilizan instalaciones con equipos de gran potencia. Al instalar un dispositivo de conversión de energía, es mejor verificar los efectos de los armónicos generados por el dispositivo. Si existe la preocupación de que los efectos de los armónicos empeoren, puede ser necesario instalar equipos como reactores y filtros pasivos para reducir los efectos de los armónicos.

### 1.2. Objetivos

El objetivo principal de este Proyecto es el desarrollo de un sistema de monitoreo para variables eléctricas, que no solo abarca el análisis detallado de armónicos, sino que también promueve una utilización energética eficiente. Este sistema está diseñado para proporcionar control y acceso remoto a los datos, facilitando su visualización y almacenamiento en la nube, lo cual representa un paso significativo hacia la gestión inteligente de la energía eléctrica. A partir de esta premisa fundamental, se establecen varios objetivos específicos:

1. **Desarrollo de un Sistema Integral de Medición:** Se busca diseñar y construir un sistema comprensivo que recoja datos eléctricos de manera eficiente y los presente de forma clara y comprensible para el usuario, facilitando así la interpretación y el análisis de la información.

2. **Creación de una Solución Económica y Escalable:** El sistema debe ser costeable en comparación con opciones comerciales existentes, ofreciendo además la flexibilidad de ser escalado según las necesidades específicas del usuario, adaptándose así a diversos contextos de uso.
3. **Visualización en Tiempo Real de Parámetros Eléctricos:** Es esencial habilitar la observación instantánea de las mediciones eléctricas, tanto localmente a través del display integrado en el dispositivo, como remotamente mediante el uso del servidor de Google Spreadsheets, asegurando acceso constante y en tiempo real a la información.
4. **Fomento de la Eficiencia Energética:** Utilizando los datos recabados sobre el consumo eléctrico, el sistema debe servir como una herramienta para promover un uso más racional y eficiente de la energía, contribuyendo de esta manera a la optimización del consumo y a la prolongación de la vida útil de los equipos conectados al sistema.
5. **Diseño Compacto para Fácil Integración:** La solución debe ser diseñada con dimensiones reducidas, permitiendo una integración sencilla en instalaciones eléctricas residenciales, sin requerir modificaciones extensivas o equipos adicionales.

## 2. Desarrollo

En la creación de este proyecto, se otorgó especial importancia a diseñar un dispositivo de dimensiones compactas, adecuadas para su fácil integración en tableros eléctricos residenciales, siguiendo las dimensiones estándar de interruptores automáticos tetrapolares o térmicas. Esta necesidad de compatibilidad dimensional guió la selección de una carcasa personalizada, apta para ser fabricada mediante impresión 3D, que cumple con los requisitos de tamaño y adaptabilidad.

La elección del microcontrolador ESP32 como núcleo del sistema fue una decisión estratégica, influenciada por sus destacadas capacidades de procesamiento y su eficiencia energética, así como su inmejorable adaptación para el desarrollo de dispositivos dentro del paradigma del IoT. La capacidad de conectividad WiFi del ESP32 se destaca como un componente crucial, brindando al prototipo una conectividad inalámbrica, esencial para la transmisión de datos y el acceso remoto a las mediciones.

Para asegurar un desarrollo coherente y estructurado del proyecto, se elaboró un diagrama detallado que delineó la arquitectura de programación del dispositivo. Este diagrama funcionó como una hoja de ruta esencial, clarificando la lógica de programación y facilitando la implementación del código. Esta organización previa fue decisiva para comprender la estructura funcional del dispositivo, garantizando una base sólida para su desarrollo posterior.

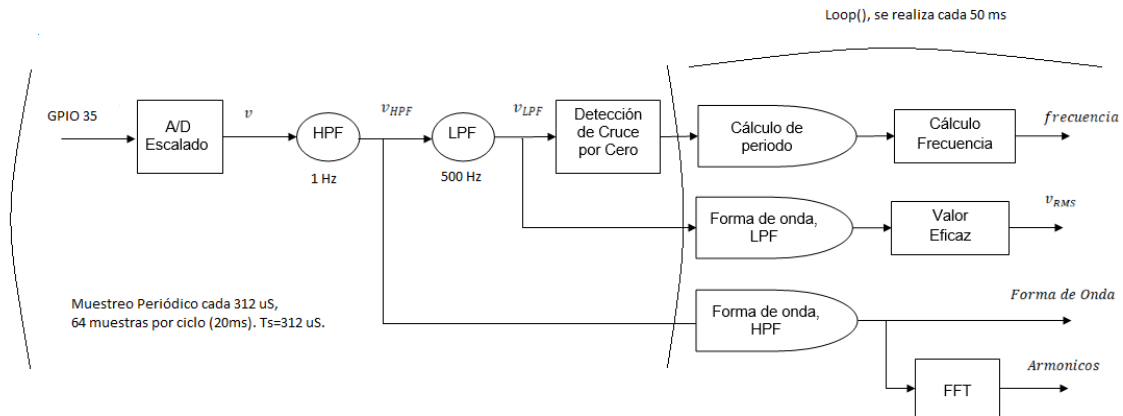


Figura 4 Diagrama en bloques funcionamiento general del código implementado

El diseño del sistema para la medición de tensión y corriente en este proyecto se caracteriza por enfocarse en la precisión y la seguridad. La cadena de medición inicia con la conexión de la línea de tensión AC a una bornera, seguida por un circuito adaptador de niveles que ajusta la tensión a un rango seguro y lógico para el sistema. Este ajuste es vital para la compatibilidad con el microcontrolador y para prevenir daños potenciales.

La incorporación de un amplificador operacional con aislamiento galvánico juega un papel crucial en la protección contra posibles descargas eléctricas, tanto para el sistema como para el usuario. Este elemento se complementa con un filtro de entrada y una red de resistencias, optimizando la señal antes de su procesamiento por el Conversor Analógico-Digital (ADC) situado en el pin GPIO35 del ESP32. Este arreglo garantiza que la señal esté adecuadamente centrada para su posterior análisis.

Desde la perspectiva del software, se implementa un muestreo periódico de la señal cada 312,5  $\mu$ s, acumulando 64 muestras durante un ciclo de 20 ms para capturar de manera efectiva las fluctuaciones de la señal eléctrica. La señal muestreada es procesada inicialmente por un filtro digital de PasaAltos (HPF) Butterworth de segundo orden, con una frecuencia de corte de 1 Hz, eliminando así la componente de corriente continua (CC) y preparando la señal para un análisis más exhaustivo.

El análisis espectral, incluyendo el cálculo de armónicos y el de Distorsión Armónica Total (THD), se realiza sobre la señal filtrada por el HPF. Este paso es esencial para identificar las frecuencias específicas que contribuyen a la distorsión armónica. Para la detección de cruces por cero y el cálculo de la frecuencia y valor eficaz, la señal es adicionalmente filtrada por un filtro de PasaBajos (LPF) Butterworth de segundo orden con una frecuencia de corte de 500 Hz. Este filtro se encarga de suprimir ruidos y señales indeseadas que podrían afectar la precisión en la detección de cruces por cero y en los cálculos derivados.

Este enfoque integral para el procesamiento de la señal, ejecutado aproximadamente cada 50 ms, destaca la eficiencia del sistema para monitorear en tiempo real y con gran detalle las características eléctricas. Cada etapa de este proceso ha sido cuidadosamente diseñada para asegurar mediciones exactas y confiables, fundamentales para el análisis detallado y la mejora del uso energético. Este proceso detallado se explicará con mayor profundidad en secciones posteriores para asegurar una comprensión completa del sistema desarrollado.

## 2.1 Diseño e implementación

### 2.1.1 Placa Madre

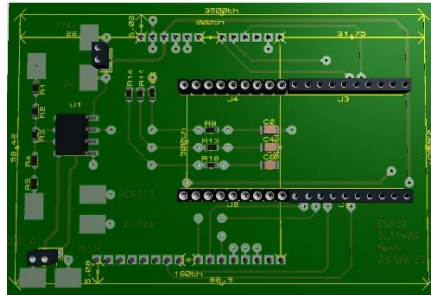


Figura 5 Diseño Placa Madre utilizada vista superior

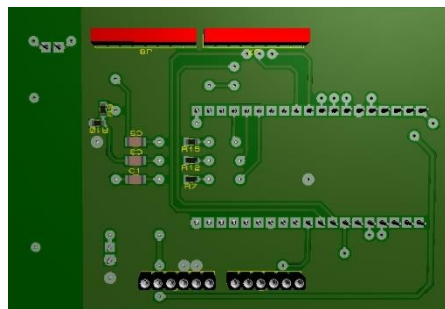


Figura 6 Diseño Placa Madre utilizada vista inferior

El diseño y desarrollo de una placa madre específica para este proyecto representa un paso importante desde donde nace la electrónica necesaria. La motherboard no solo aloja la placa de desarrollo ESP32, sino que también integra los pines de entrada utilizados, un display para la visualización de datos en tiempo real, y los circuitos de adaptación de niveles y protección requeridos para garantizar la funcionalidad y seguridad del sistema.

El display permite una interacción directa con el usuario, mostrando en tiempo real los parámetros eléctricos medidos, lo cual es fundamental para el monitoreo y la toma de decisiones basadas en datos. Esta característica es especialmente útil para la visualización inmediata de información, facilitando así el diagnóstico y la gestión de la energía.

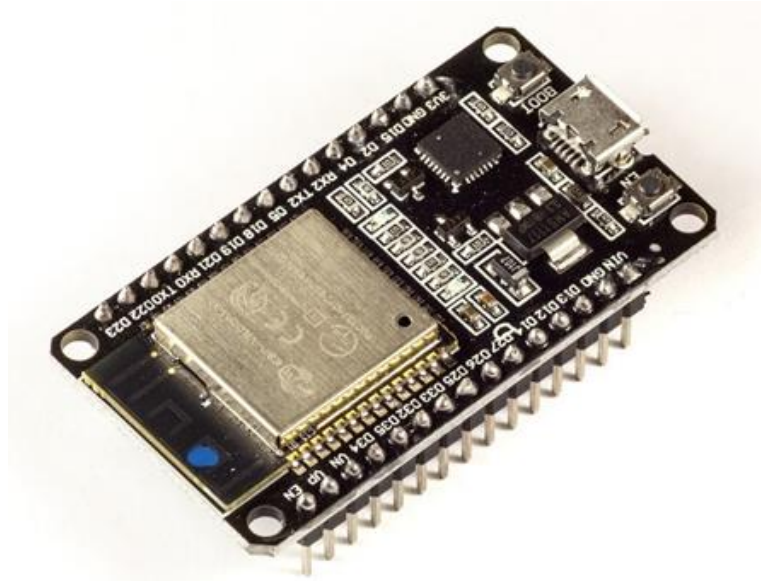
Los circuitos de adaptación de niveles son esenciales para convertir las señales eléctricas de alta tensión a niveles compatibles con los dispositivos electrónicos de baja tensión, como el microcontrolador ESP32, sin comprometer la precisión de las mediciones. Estos circuitos aseguran que la señal de entrada sea adecuadamente preparada para su procesamiento posterior, manteniendo la integridad de los datos.

Por último, los circuitos de protección incorporados en la placa madre desempeñan un papel vital en la salvaguarda de la electrónica del proyecto contra sobretensiones, cortocircuitos, y otros eventos eléctricos adversos. Estos circuitos protegen no solo el hardware, sino también la seguridad del usuario, haciendo del proyecto una solución robusta y confiable para el monitoreo eléctrico.

En conjunto, la placa madre diseñada es el corazón del proyecto, integrando de manera eficaz todos los componentes necesarios para el monitoreo y análisis de las variables eléctricas, y estableciendo una base sólida para la realización exitosa del proyecto.



## 2.1.2 ESP-32



*Figura 7 ESP32 con descripción de pines de salida/entrada*

La incorporación del microcontrolador ESP32 (ESP32 Pinout reference, s.f.) en este proyecto aporta una serie de ventajas técnicas significativas, fundamentales para su éxito. Este componente se distingue por su arquitectura de alto rendimiento y eficiencia energética, basada en un núcleo Tensilica LX6 dual-core que opera a una frecuencia de hasta 240 MHz. Esta capacidad de procesamiento permite al ESP32 manejar complejas operaciones de análisis y procesamiento de señales eléctricas con gran eficacia, lo cual es crucial para la monitorización en tiempo real y el análisis de armónicos, THD (Total Harmonic Distortion), y otras métricas eléctricas.

Desde la perspectiva del IoT, el ESP32 ofrece capacidades avanzadas de conectividad, incluyendo WiFi 802.11 b/g/n, que facilitan la transmisión de datos recolectados hacia plataformas en la nube o sistemas de gestión locales sin la necesidad de hardware adicional. Esta conectividad integrada es esencial para implementar funcionalidades como el acceso remoto y el monitoreo en tiempo real, permitiendo una gestión eficiente de la energía eléctrica y el diagnóstico a distancia de las instalaciones.

Además, el ESP32 integra una amplia gama de funcionalidades periféricas y de interfaz, incluyendo ADCs (Convertidores Analógicos a Digitales) de alta resolución, interfaces SPI, I2C, y UART, así como capacidades de PWM (Modulación por Ancho de Pulso), lo que lo hace extraordinariamente versátil para adaptarse a las necesidades específicas del proyecto. Esta flexibilidad permite la conexión directa con diversos dispositivos externos, como displays para la visualización de datos.

Por último, la arquitectura del ESP32 soporta técnicas avanzadas de gestión de energía, lo que permite optimizar el consumo energético del dispositivo en aplicaciones que requieren operación continua. Esto es particularmente beneficioso en el contexto de sistemas embebidos para el monitoreo eléctrico, donde la eficiencia energética puede tener un impacto significativo en la viabilidad y sostenibilidad del proyecto a largo plazo.

El ESP32 se posiciona como un componente idóneo para este proyecto, ofreciendo una combinación inigualable de rendimiento de procesamiento, conectividad IoT integrada,

versatilidad en la interfaz de periféricos, y eficiencia energética, estableciendo así una plataforma robusta y adaptable para el desarrollo de sistemas avanzados de monitoreo eléctrico.

### 2.1.3 Display ILI9486 320x480

El display ILI9486 (ILI9486, s.f.) es un controlador LCD TFT con una resolución de 320x480 píxeles, perfecto para proyectos que demandan una interfaz gráfica detallada. Ofrece una amplia paleta de colores y utiliza una interfaz paralela de 16 bits para una comunicación eficiente con microcontroladores, como el ESP32. Este display resulta ideal para visualizar información compleja, incluyendo gráficos de armónicos y valores de THD, en aplicaciones como sistemas de monitoreo eléctrico y dispositivos IoT. Su alta resolución y capacidad de color, junto con su eficiencia energética, lo hacen adecuado para presentar datos eléctricos de manera clara y accesible, mejorando la experiencia del usuario y la interpretación de las métricas críticas de calidad de energía.

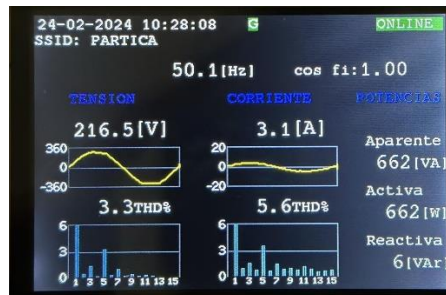


Figura 8 Display utilizado para el equipo

### 2.1.4 Medición de tensión SI8920BC

Se trata de un integrado con encapsulado DIP-8 que consta de un amplificador operacional aislado que fue utilizado para separar la medición de alta tensión de la parte lógica, para proteger tanto a los dispositivos como al usuario.

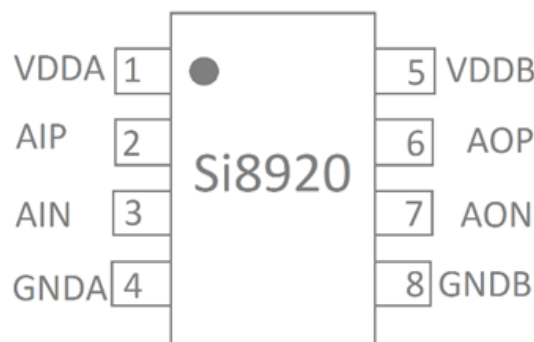


Figura 9 Encapsulado DIP-8 de operacional Si8920

El Si8920 (Si8920, s.f.) es un amplificador analógico aislado galvánicamente. La entrada diferencial de bajo voltaje es ideal para medir voltaje a través de un resistor de derivación de corriente o para cualquier lugar donde un sensor debe estar aislado del sistema de control. La salida es un diferencial analógico de señal amplificada por 8.1x o 16.2x.

## Esquemático

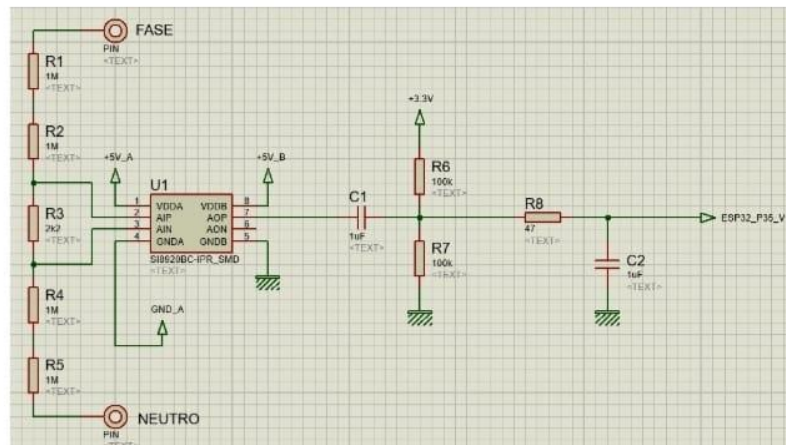


Figura 10 Esquemático completo circuito de adaptación de niveles

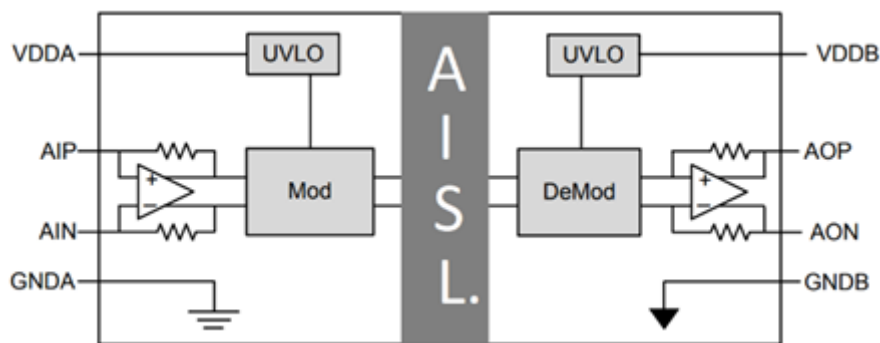


Figura 11 Diagrama en Bloque Funcional Si8920

### 2.1.5 Fuentes de alimentación 5V

Una fuente de alimentación de AC-DC (AC-DC power module 5V 700mA, s.f.) que proporciona 5V y 700mA es un dispositivo que convierte la corriente alterna (AC) de la red eléctrica en una corriente continua (DC) de baja tensión y corriente específicas. Estas fuentes de alimentación son comunes en una gran variedad de dispositivos electrónicos, como cargadores de teléfonos, routers de internet, y muchos otros aparatos que requieren una alimentación de bajo voltaje y corriente moderada para su funcionamiento. Diseñadas para ser compactas y ligeras, estas fuentes de alimentación suelen tener un tamaño reducido para facilitar su integración en dispositivos o para ser utilizadas como cargadores portátiles. Los modelos conmutados suelen ser más eficientes y ligeros, pero pueden generar más ruido eléctrico o armónicos que deben ser filtrados adecuadamente.



Figura 12 Fuente Switching 5V utilizada en el equipo

### 2.1.6 Pinza Sensor De Corriente 30a Sct013

La pinza SCT-013 es un sensor de corriente no invasivo ampliamente utilizado en proyectos de monitoreo de energía eléctrica, particularmente en aplicaciones de medición y análisis de consumo energético. Este tipo de sensor, también conocido como transformador de corriente (TC), permite medir la corriente que fluye a través de un cable sin necesidad de hacer contacto directo con él o interrumpir el circuito eléctrico. Esto lo hace ideal para implementaciones de monitoreo de energía en entornos residenciales o comerciales, donde la seguridad y la facilidad de instalación son prioritarias.

Características y ventajas del SCT013 de 30A incluyen:

1. **Medición No Invasiva:** Permite medir la corriente sin interrumpir el circuito, lo que es ideal para aplicaciones de monitoreo y diagnóstico en sistemas eléctricos.
2. **Rango de Medición:** Capaz de medir corrientes de hasta 30A, lo que lo hace adecuado para la mayoría de las aplicaciones domésticas y muchas industriales.
3. **Facilidad de Uso:** Su diseño de pinza permite una instalación rápida y sencilla, simplemente abriendo la pinza y cerrándola alrededor del conductor.
4. **Salida Analógica:** Típicamente proporciona una salida analógica que puede ser fácilmente convertida a una señal digital mediante un ADC para su procesamiento con microcontroladores como el ESP32.
5. **Seguridad:** Al ser un método de medición aislado, mejora la seguridad del usuario y del equipo de medición, minimizando el riesgo de accidentes eléctricos.

El SCT013 es ideal para integrar en sistemas de monitoreo de energía, como parte de un sistema de gestión de carga o simplemente para obtener lecturas de consumo de corriente en tiempo real.



Figura 13 Pinza utilizada para la medición de corriente

El sensor de corriente SCT013 proporciona una salida proporcional a la corriente detectada en forma de una señal senoidal que oscila entre -1 y 1 V. Para que esta señal pueda ser leída correctamente por un ADC (Convertor Analógico Digital) de un microcontrolador como el ESP32, que típicamente mide voltajes en un rango de 0 a 3.3 V, se necesita modificar la señal para incluir un offset que desplace la señal completa al rango positivo.

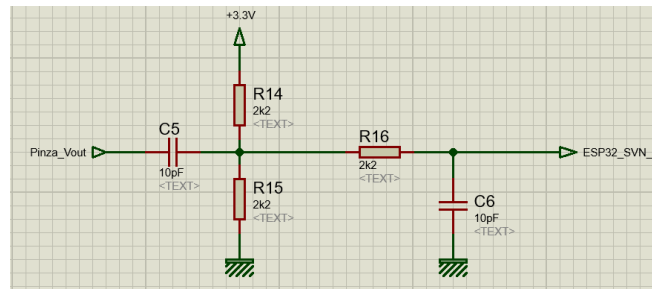
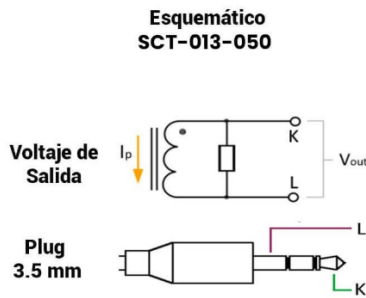


Figura 14 Circuito adaptador de niveles y filtrado.

Esta modificación se realiza a través de un circuito de adaptación de señal. La idea es que la senoidal de -1 a 1 V se transforme en una señal que oscile alrededor del punto medio del rango de entrada del ADC (1.65 V para el ADC de 3.3 V que implementa el microcontrolador), manteniendo la misma forma y amplitud. Esto asegura que la señal de AC se pueda medir efectivamente en su totalidad, ya que el ADC solo puede leer valores positivos. La señal resultante tendría una forma de onda que oscila entre  $1.65 \text{ V} \pm 0.5 \text{ V}$  si el rango de medición es de 0 a 3.3 V. Con esta adaptación, el ESP32 puede procesar la señal y calcular la corriente real que fluye a través del conductor. La conexión se realiza con un conector de audio estéreo estándar de 3.5 mm (Plug). La conexión central o anillo (Ring) no está conectada.

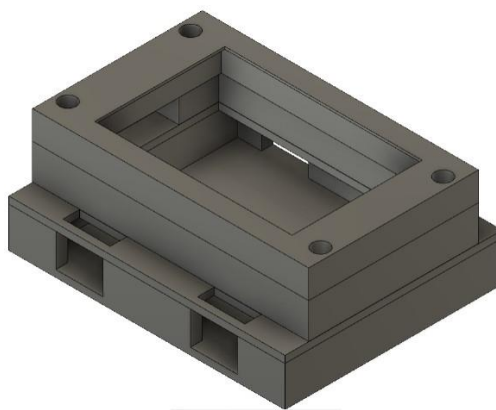


*Figura 15 Esquema de conexión plug Pinza amperométrica*

### 2.1.7 Diseño del Gabinete

Una premisa fundamental del proyecto era el desarrollo de un dispositivo con dimensiones compactas, que permitiera su instalación directa en un tablero eléctrico residencial sin requerir modificaciones significativas en la infraestructura existente. Con este objetivo en mente, se tomó la decisión de diseñar un gabinete personalizado que se ajustara a las dimensiones estándar de un interruptor automático tetrapolar, asegurando así su compatibilidad con los sistemas de distribución eléctrica domiciliarios. Este enfoque en el diseño permitió que el dispositivo se pudiera montar fácilmente en un Riel-Din, un sistema de fijación universalmente adoptado en tableros eléctricos, lo cual facilita enormemente su integración en diferentes entornos y configuraciones eléctricas.

La adaptación del diseño a las medidas específicas de una térmica tetrapolar no solo responde a un criterio de eficiencia espacial, sino que también refleja un compromiso con la accesibilidad y la facilidad de instalación. Este diseño asegura que el dispositivo no solo cumpla con las expectativas funcionales y técnicas del proyecto, sino que también se alinee con los requisitos prácticos y estéticos de los sistemas eléctricos residenciales.



*Figura 16 Gabinete diseñado en impresora 3D vista superior*

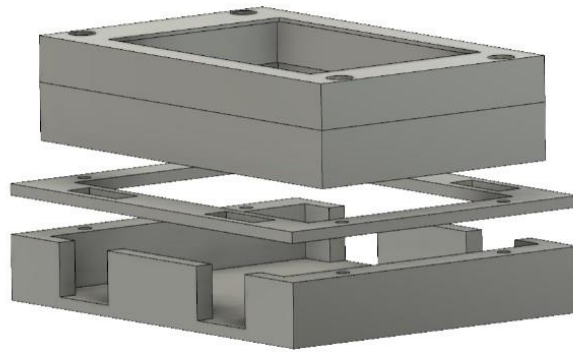


Figura 17 Gabinete diseñado en impresora 3D vista lateral

## 2.2 Tensión circuito

Para poder adecuar los valores de tensión a medir que deben ingresar al ADC se siguen los siguientes pasos, disminuir la amplitud, montar la parte alterna de la señal al nivel central del ADC y realizar un filtrado para eliminar las componentes frecuenciales indeseadas.

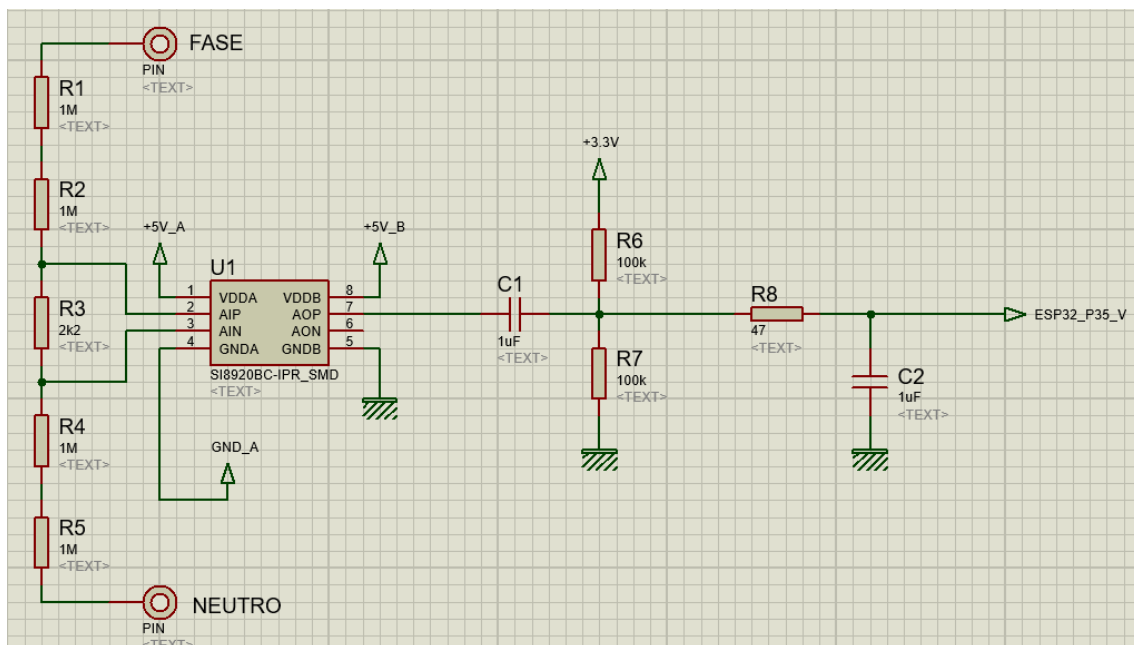


Figura 18 Circuito utilizado para la adaptación de señal tensión a valores lógicos.

Para disminuir la amplitud de tensión la elección de un divisor de tensión fue la opción tomada. Se utilizaron varios resistores en lugar de solo dos por cuestiones de disipación y también de aislación ya que entre extremos de cada resistor la tensión es solo una parte del total.

$$I = \frac{V}{R_1 + R_2 + R_3 + R_4 + R_5} = \frac{220V_{rms}}{5002200\Omega} = 4,398 \cdot 10^{-5} Arms$$

$$P_{R1} = I^2 \cdot R_1 = (4,398 \cdot 10^{-5} A)^2 \cdot 1000000\Omega = 1,93 mW$$

$$V_{R1} = I \cdot R_1 = 4,398 \cdot 10^{-5} A \cdot 1000000\Omega = 43,98 V_{rms}$$



Las resistencias smd del tamaño 0805 se seleccionaron por cumplir con los requerimientos, por ser fáciles de montar y también por su disponibilidad a la hora de comprarlas.

El divisor de tensión compuesto por R1, R2, R3, R4 y R5 tiene una relación:

$$\frac{V_o}{V_i} = \frac{R_3}{R_1 + R_2 + R_3 + R_4 + R_5} = \frac{2200}{5002200} = 4,39 \cdot 10^{-4}$$

Con esta relación teniendo una tensión pico de entrada de 311,12 V (220Vrms) sobre R3 tenemos 136,83 mV.

Sabiendo que el amplificador operacional admite un máximo de 200 mV de pico la máxima tensión pico que podemos medir es de 454,74 Vp o de 321,55 Vrms.

$$V_{ip} = \left(\frac{V_o}{V_i}\right)^{-1} V_{op} = \frac{5002200}{2200} 0,2Vp = 454,74 Vp$$
$$V_{irms} = \frac{V_{ip}}{\sqrt{2}} = \frac{454,74 Vp}{\sqrt{2}} = 0,2Vp = 321,55 Vrms$$

Teniendo en cuenta lo anterior se determina que en condiciones normales de 220 Vrms se estaría midiendo en un 68% de la escala completa (220 Vrms/321,55 Vrms. 100) lo cual nos deja un gran margen de seguridad y también nos aleja de los límites del componente en donde existen alinealidades que no son deseadas.

El amplificador operacional SI8920 tiene una ganancia de tensión 8 pero esto ocurre entre los pines 6(AoP) y 7(AoN). En nuestro caso para simplificar el diseño se toma la salida desde el pin 6 pero respecto al 8 que es en GND del dispositivo y en esta configuración la ganancia en tensión se ve reducida a 4.

### 2.2.1 Análisis en DC para la adquisición de tensión

El capacitor C1 bloquea la componente de continua de salida del amplificador operacional que es de 2,5V. La red compuesta por R6 y R7 es alimentada por la tensión de 3,3V regulada por la placa del ESP32 y tiene como objetivo montar la señal que pasa por C1 justo en la mitad del rango del ADC del microcontrolador.

Los valores de R6 y R7 fueron elegidos teniendo en cuenta que no sean muy bajos para sobrecargar el circuito y que no sean muy altos como para que no exista alguna contaminación por ruidos. 100K fue el resultado de varias pruebas realizadas con resistencias en protoboard.

Teniendo en cuenta que el ADC tiene una impedancia de entrada muy alta el efecto de R8 puede despreciarse y C2 en continua también puede despreciarse.

En la siguiente tabla pueden apreciarse los rangos y escalados de señal hacia el ADC.



Tabla 2 Valores de escalado de señal hacia ADC tensión.

Rango de medición	Entrada		Salida del divisor	Salida SI8920	Después de C1 R6 R7	ADC	Rango ADC
%	Vrms	Vpico	Vpico	Vpico	Vpico	valor	%
100,0	321,6	454,7	0,200	3,30	2,45	3040	74,2
68,4	220,0	311,1	0,137	3,05	2,20	2727	66,6
0,0	0,0	0,0	0,000	2,50	1,65	2048	50,0
68,4	220,0	-311,1	-0,137	1,95	1,10	1367	33,4
100,0	321,6	-454,8	-0,200	1,70	0,85	1055	25,8

### 2.2.2 Análisis en AC para la adquisición de tensión

C1 y el paralelo de R6 y R7 conforman un filtro pasa altos y los valores fueron elegidos teniendo en cuenta lo disponible en el mercado junto con el resultado de tener en frecuencia de corte de una década menor a 50Hz, de esta forma a la frecuencia de trabajo no tendríamos problemas.

$$f_c = \frac{1}{2 \cdot \pi \cdot (R_{11} || R_{12}) \cdot C_1} = \frac{1}{2 \cdot \pi \cdot 50K\Omega \cdot 1\mu F} = 3,18Hz$$

R8 y C2 conforman un filtro pasa bajos. La frecuencia de muestreo del ADC es de 3200 Hz y el motivo de montar este filtro es el de tener un efecto anti-aliasing. Los capacitores son smd del tamaño 1206 y se tomó el mismo valor que en el filtro anterior para no tener problemas a la hora del montaje ya que no habría que distinguir un componente de otro distinto.

$$f_c = \frac{1}{2 \cdot \pi \cdot R_8 \cdot C_2} = \frac{1}{2 \cdot \pi \cdot 47\Omega \cdot 1\mu F} = 3386,27 Hz$$

Suponiendo valores de impedancia de entrada y de salida para el conjunto de filtros, se realiza una simulación con software para obtener la respuesta de ambos filtros juntos.

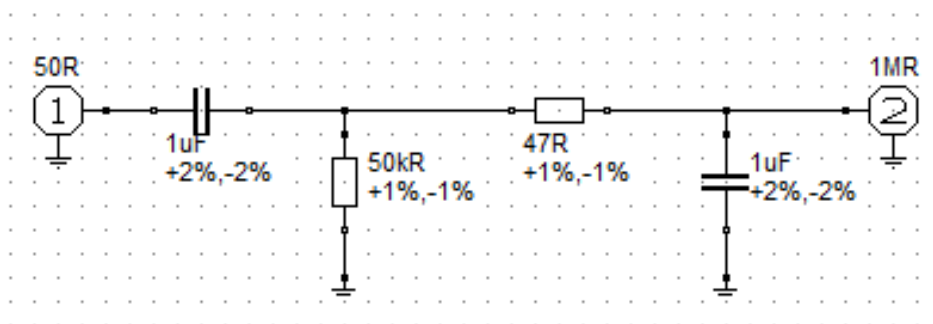


Figura 19 Esquemático filtro de RC entrada

La siguiente gráfica muestra la respuesta del circuito anterior, el eje de ordenadas va de -5 dB a 5 dB y el de abscisas de 1 Hz a 5 kHz. El valor de -3 dB se obtiene en 1,66 Hz y en 3200 Hz según la gráfica.

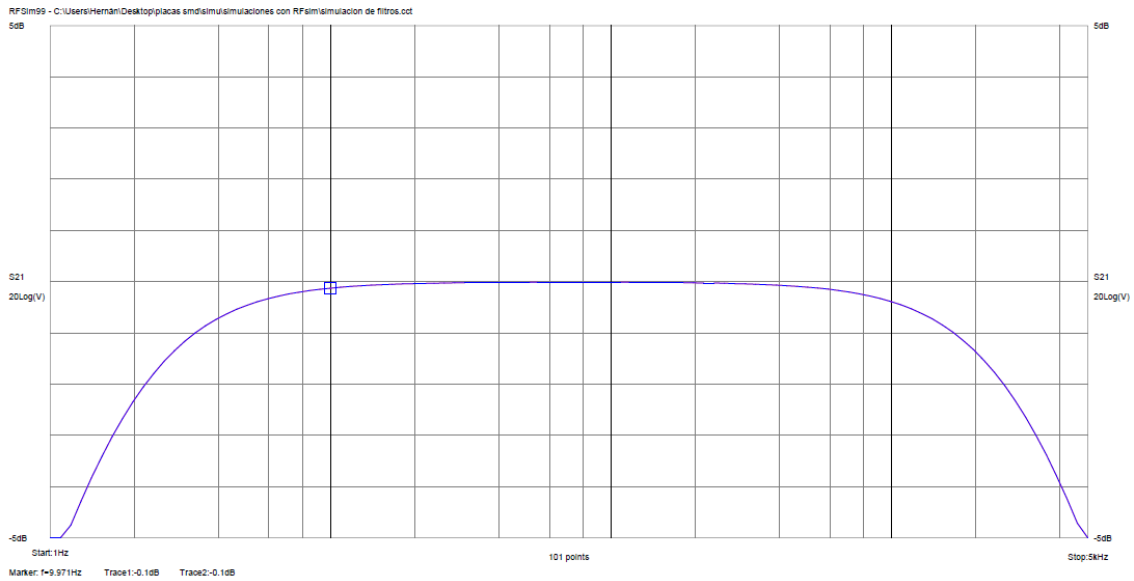


Figura 20 Simulación respuesta del filtro RC implementado en el circuito

### 2.3 Corriente circuito

Para poder adecuar los valores de corriente a medir que deben ingresar al ADC se siguen los siguientes pasos, disminuir la amplitud, montar la parte alterna de la señal al nivel central del ADC y realizar un filtrado para eliminar las componentes frecuenciales indeseadas.

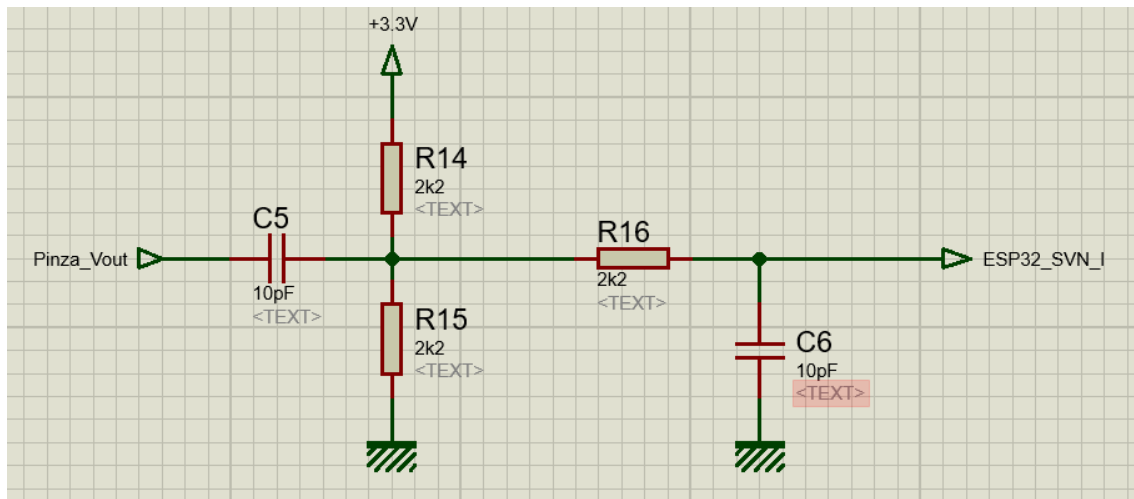


Figura 21 Esquemático adaptación de niveles pinza

#### 2.3.1 Análisis en DC para la adquisición de corriente

La señal que genera la pinza SCT-013, nos entrega una señal alterna de 0 a 1 v, para poder medir correctamente esta señal con el ADC, es necesario realizar algunas modificaciones.

En primer lugar, se observa el capacitor C5 que tiene como objetivo bloquear la continua de salida del sensor de corriente. La red compuesta por R14 y R15 es alimentada por la tensión de

3,3 V regulada por la placa del ESP32 y tiene como objetivo montar la señal que pasa por C5 justo en la mitad del rango del ADC del microcontrolador.

Los valores de R14 y R15 fueron elegidos teniendo en cuenta que no sean muy bajos para sobrecargar el circuito y que no sean muy altos como para que no exista alguna contaminación por ruidos. 2,2 k $\Omega$  fue el resultado de varias pruebas realizadas con resistencias en protoboard.

Teniendo en cuenta que el ADC tiene una impedancia de entrada muy alta el efecto de R16 puede despreciarse y C6 en continua también puede despreciarse.

### 2.3.2 Análisis en AC para la adquisición de corriente

Los componentes C5, R14, R15, R16 y C6 cumplen la misma función que C1, R6, R7, R8 y C2 en el circuito de adquisición de tensión.

## 3. Gestión de memoria y de procesos en sistemas embebidos

### 3.1 ESP32 – Arquitectura

Como se indicó en la sección 2.1.2., el microcontrolador elegido para el Proyecto es el ESP32. El motivo de su elección no es sólo por la potencia del mismo y su bajo consumo, sino que se encuentra orientado al desarrollo de dispositivos para el “Internet de las cosas” o IoT. Además, brinda una integración con la conectividad WiFi, esencial para el desarrollo de nuestro prototipo. Una de las características fundamentales y que destacan a esta placa, es su arquitectura doble núcleo, función implementada en la realización de nuestro Proyecto.

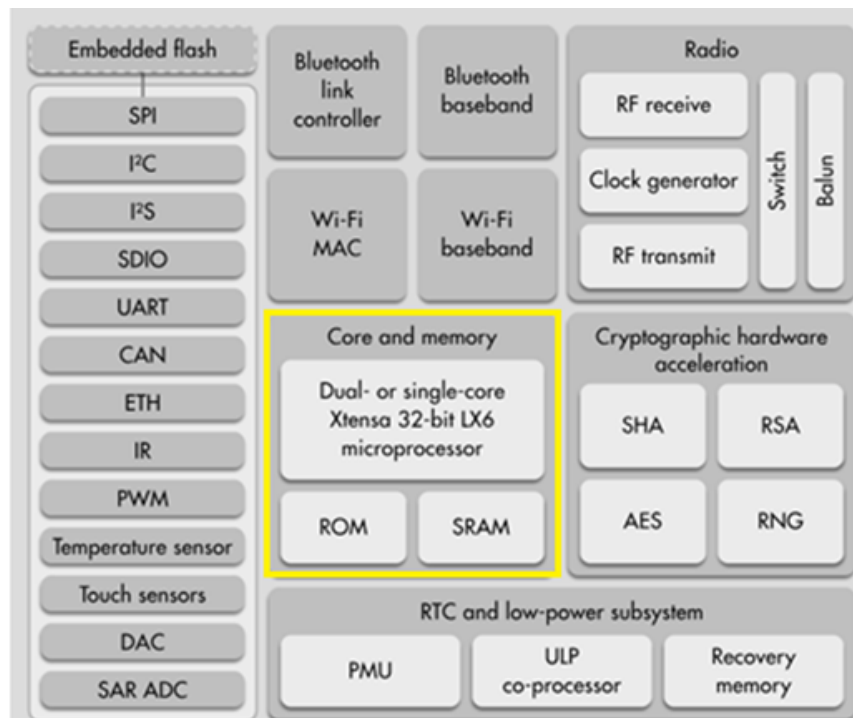


Figura 22 Descripción de arquitectura ESP32

El procesador Xtensa Dual-Core LX6 de 32 bits a 160 ó 240 MHz, y coprocesador de ultra baja potencia (ULP), fueron utilizados para implementar el proyecto mediante la multitarea, realizando una asignación de las diversas funciones a ambos núcleos, de una manera estratégica. En las próximas secciones, se detallarán estas asignaciones e implementación. A continuación, se seleccionarán y desarrollarán aquellas características utilizadas para la implementación en el proyecto.

### Conectividad WiFi

El chip cuenta con conectividad WiFi, siendo compatible con 802.11 b/g/n en la banda de los 2.4GHz, alcanzando velocidades de hasta 150 Mbits/s.

### Interfaces Periféricas

Entre las interfaces disponibles en ESP32, podemos destacar I2C, SPI, PWM, CAN, entre otros. Para el conexionado de la pantalla ILI9486 se utilizó el bus paralelo.

### Bus Paralelo 8 bits

El ILI9486 es un controlador de pantalla LCD popular que soporta resoluciones de hasta 320x480 píxeles y puede manejar colores de 18 bits. A continuación, se enumeran las conexiones necesarias para su funcionamiento.

1. **Líneas de Datos:** D0-D7 para transferencia de datos de 8 bits.
2. **RS (Registro Select/Command):** Selecciona si el byte enviado es un comando (RS bajo) o un dato (RS alto).

3. **WR (Write):** Señal de escritura, activa la escritura de datos o comandos al display cuando está en bajo.
4. **RD (Read):** Señal de lectura, activa la lectura de datos desde el display cuando está en bajo (no siempre se utiliza).
5. **CS (Chip Select):** Activa el display cuando está en bajo.
6. **RST (Reset):** Reinicia el controlador del display.

**Pasos para la conexión.**

1. **Conectar las líneas de alimentación (Vcc y GND)** entre el ESP32 y el display.
2. **Conectar las líneas de datos (D0-D7)** desde el ESP32 a los pines correspondientes en el display.
3. **Conectar las líneas de control (RS, WR, RD, CS, y RST)** desde el ESP32 a los pines respectivos en el display.

Esquema de conexión utilizado:

*Tabla 3 Asignación de pines ESP32 para conexión de ILI9486*

PIN ESP32	PANTALLA ILI9486
4	WR
2	RD
15	RS (D/C)
12	D0
13	D1
26	D2
25	D3
17	D4
16	D5
27	D6
14	D7

### 3.2 Uso de FREERTOS

FreeRTOS (*FreeRTOS, s.f.*) es un sistema operativo en tiempo real (RTOS) popular para microcontroladores y sistemas embebidos. Proporciona una manera eficiente de gestionar múltiples tareas (hilos de ejecución) dentro de un dispositivo, facilitando la programación de aplicaciones complejas que requieren paralelismo y temporización precisa. A continuación, se enuncian algunos conceptos clave y características de FreeRTOS.

**Características Principales**

- **Multitarea:** FreeRTOS permite la creación y gestión de múltiples tareas, cada una con su propia pila y prioridad, lo que facilita la organización del código y la ejecución concurrente.

- **Sincronización y Comunicación entre Tareas:** Proporciona mecanismos como semáforos, colas de mensajes y mutex para sincronizar tareas y compartir datos de manera segura entre ellas.
- **Temporizadores:** Soporta temporizadores de software para ejecutar tareas después de un periodo de tiempo o de manera periódica.
- **Gestión de Memoria:** Ofrece opciones para la gestión estática y dinámica de la memoria, permitiendo un control fino sobre el uso de los recursos.
- **Portabilidad:** Está diseñado para ser portátil y se puede ejecutar en una amplia gama de microcontroladores y procesadores, desde pequeños dispositivos de 8 bits hasta potentes unidades de 32 bits.

En FreeRTOS, el programador define y crea tareas que el sistema operativo luego gestiona, asignándoles tiempo de CPU según su prioridad y el algoritmo de planificación (scheduling) elegido. Un "scheduler" (planificador) de tareas se encarga de determinar cuál tarea debe ejecutarse en cada momento, basándose en su estado y prioridad. Las tareas pueden estar en diferentes estados, como "listo para ejecutar", "en espera" o "suspendido".

Hay diversas tareas que pueden implementarse en un sistema operativo de tiempo real:

- **Tareas periódicas.** Atienden eventos que ocurren constantemente y a una frecuencia determinada. Por ejemplo, la tarea de muestreo de la señal de entrada del ADC.
- **Tareas aperiódicas.** Atienden eventos que no se saben cuándo van a ocurrir. Estas tareas están inactivas (bloqueadas) hasta que no ocurre el evento de interés. En nuestro caso, el envío de datos al servidor
- **Tareas de procesamiento continuo.** Tareas que trabajan en forma permanente.

Un aspecto importante es que no todas estas tareas requieren la misma frecuencia de operación. Por ejemplo, la lectura de los sensores tendrá que realizarse más frecuentemente que, por ejemplo, el envío de datos al servidor. Otro punto importante es que no todas esas tareas tienen la misma prioridad. Teniendo en cuenta estos conceptos, los códigos implementados se desarrollaron mediante la definición de funciones específicas (tareas), donde se buscó un equilibrio de tal forma de maximizar su funcionamiento.

### 3.3 Declaración de funciones/tareas

#### Declaración de una Función de Tarea

Una función de tarea en FreeRTOS debe tener el siguiente prototipo:

```
/ Se define el "handler" o manejador de las tareas
TaskHandle_t taskHandle_sample, taskHandle_GSS;
```

La siguiente función es una ISR (Interrupt Service Routine) para un temporizador. Se marca con **IRAM\_ATTR** para asegurar que se almacene en la RAM de instrucciones, lo cual es necesario para las ISR en el ESP32. Dentro de la ISR, se llama a **xTaskNotifyFromISR()** para enviar una

notificación a la tarea **muestreo** (a través de su manejador **taskHandle\_sample**). La notificación puede despertar la tarea si está esperando esta notificación.

```
// Timer ISR
void IRAM_ATTR onTimer()
{
    BaseType_t taskWoken;
    xTaskNotifyFromISR(taskHandle_sample, 0, eIncrement, &taskWoken);
}
```

La tarea **muestreo** espera indefinidamente una notificación (**xTaskNotifyWait**) antes de proceder con su lógica de procesamiento. Este mecanismo de espera y notificación es eficiente para sincronizar tareas con eventos externos o interrupciones sin ocupar CPU innecesariamente. La tarea **muestreo** es la que se utiliza para leer el ADC, tanto del GPIO34 como del GPIO35, se trata de la tarea principal de todo el proyecto.

```
// Tarea ejecutada por Timer ISR:
// Muestreo de señales, detección de cruce por cero y del desfase
void muestreo(void *pvParameters)
{
    while (true)
    {
        // espera notificacion
        xTaskNotifyWait(0, 0, &ulNotifiedValue, portMAX_DELAY);
        // inicio tarea muestreo

        .
        .
        .

    } // fin tarea muestreo
}
```

La llamada **xTaskCreateUniversal()** crea la tarea **muestreo**, asignándole un tamaño de pila, una prioridad y un núcleo de CPU específico para su ejecución. El identificador de la tarea (**taskHandle\_sample**) se utiliza para referencias futuras, como enviar notificaciones a la tarea.

Creación de la tarea

```
xTaskCreateUniversal(
    muestreo,           // Función que se ejecutará como tarea
    "muestreo",        // Nombre
    8192,              // Tamaño de Pila
    NULL,              // Parametro de entrada de tarea
    24,                // Prioridad (la mas alta)
    &taskHandle_sample, // Identificador de tarea
    PRO_CPU_NUM        // Nucleo en el que se debe ejecutar
);
```

### 3.3 Uso de Interrupciones

Una interrupción es una señal que pausa la ejecución del programa principal para que el sistema pueda ejecutar una función específica, conocida como "servicio de interrupción" o ISR (Interrupt Service Routine). Después de que se completa el ISR, el sistema retoma la ejecución del programa principal.

```
// Crea timer para interrupcion
hw_timer_t * timer = NULL;
timer = timerBegin(0, 40, true);
timerAttachInterrupt(timer, &onTimer, true);
timerAlarmWrite(timer, 625, true);
timerAlarmEnable(timer);
```

El código configura un temporizador en el ESP32 para generar interrupciones a intervalos regulares, utilizando las funciones de la API de ArduinoIDE (*ArduinoIDE*, s.f.) para ESP32:

1. **timerBegin(0, 40, true);** Inicializa el temporizador 0 con un divisor de preescaler de 40, contando hacia arriba. Esto ajusta la frecuencia de conteo basada en la frecuencia del CPU principal cada 0,5  $\mu$ s.
2. **timerAttachInterrupt(timer, &onTimer, true);** Asocia la función **onTimer** como ISR (Interrupt Service Routine) al temporizador, configurado para interrupciones de flanco.
3. **timerAlarmWrite(timer, 625, true);** Se generará una interrupción cada 625 conteos, es decir, cada 0,5  $\mu$ s x 625 = 312,5  $\mu$ s.
4. **timerAlarmEnable(timer);** Activa el temporizador para comenzar a generar interrupciones.

En resumen, se configura una interrupción timer cada 312,5  $\mu$ s.

### 3.4 Muestreo periódico

En primer lugar, se debe muestrear el voltaje del sistema de CA de 220 V en intervalos regulares. Al crear un muestreo periódico con FreeRTOS, es decir, una interrupción periódica:

1. Se creó una función que describa el proceso para operar periódicamente y se colocó la función `xTaskNotifyWait()` que espera la notificación al comienzo del ciclo infinito con `while (true)`.
2. Se agregó la función anterior como una tarea en la función `setup()`.
3. Del mismo modo, se configuró la interrupción del temporizador y se configuró la función de interrupción en la función `setup()`.
4. Se creó la función de interrupción anterior y envió una notificación a la tarea (función) creada en 1 y 2 arriba con la función `xTaskNotifyFromISR()`.



```

// Muestro de señales, detección de cruce por cero y del desfase
void muestreo(void *pvParameters)
{
    uint32_t ulNotifiedValue;
    while (true)
    {
        // Wait for notification
        xTaskNotifyWait(0, 0, &ulNotifiedValue, portMAX_DELAY);

        // Sample analog signal and apply filters
        v_adc[i] = analogRead(GPIO_ADC_V);
        i_adc[i] = analogRead(GPIO_ADC_I);

        v_scaled[i] = static_cast<float>((v_adc[i] / ADC_FULLSCALE - 0.5) * V_AC_PP);
        i_scaled[i] = static_cast<float>((i_adc[i] / ADC_FULLSCALE - 0.5) * I_AC_PP);

        v_hpf[i] = hpf.apply(v_scaled[i]);
        i_hpf[i] = hpf_I.apply(i_scaled[i]);

        v_lpf[i] = lpf.apply(v_hpf[i]);
        i_lpf[i] = lpf_I.apply(i_hpf[i]);

        // Detect zero-crossing
        i_cxc++;
        int i_pre;
        i_pre = i > 0 ? i - 1 : N_MUESTRAS - 1;
        if (v_lpf[i_pre] < 0 && v_lpf[i] >= 0) //entra si tension cruza por cero
        {
            // Check zero-crossing timing
            if (i_cxc > N_PLOT - N_VENT && i_cxc < N_PLOT + N_VENT) // N_PLOT = 64; N_VENT = 8; entra en 56 < i_cxc < 72 en tiempo es 20ms +- 2,5ms
            {
                // Save zero-crossing information
                v_cruce_cero = true;
                cxc[k].N = i_cxc;
                cxc[k].V_neg = v_lpf[i_pre];
                cxc[k].V_pos = v_lpf[i];
                k = (k + 1) % N_CXC;
            }
            i_cxc = 0;
        }

        if(v_cruce_cero) // entra si tension cruza por cero v_cruce_cero = false;
        {
            if(i_lpf[i_pre] < 0 && i_lpf[i] >= 0) // entra si corriente cruza por cero. 64 muestras por ciclo de V, I puede entrar en el rango de 0 a 16 y en el rango de 48 a 64
            {
                if((i_cxc >= 0 && i_cxc <= 16) || (i_cxc >= 48 && i_cxc < 64))
                {
                    diferencia_de_muestras = i_cxc; // diferencia_de_muestras = 0;
                    v_cruce_cero = false;
                    dift = diferencia_de_muestras * T_SAMPLE_US / 1000; // queda en mseg
                }
            }
        }

        // Incremento de indice
        i = (i + 1) % N_MUESTRAS;
    } //fin while
} // fin muestreo

```

a función **muestreo** implementa una tarea de FreeRTOS que realiza el muestreo de señales analógicas, aplica filtros de paso alto y paso bajo, detecta cruces por cero en la señal de tensión, y calcula la diferencia de tiempo entre los cruces por cero de las señales de tensión y corriente. A continuación, se describe el funcionamiento y algunos puntos clave del código:

### Descripción General

1. **Espera de Notificación:** La tarea espera indefinidamente una notificación mediante **xTaskNotifyWait**. Esta notificación es enviada por una ISR (como un temporizador) para indicar que es momento de realizar una nueva medición.
2. **Muestreo de Señales:** Se leen valores analógicos de los dos canales voltaje y corriente cada 312.5  $\mu$ s mediante **analogRead** y se escalan. La escala convierte los valores del ADC a unidades físicas basadas en la amplitud pico a pico de las señales y el valor máximo del ADC (**ADC\_FULLSCALE**).
3. **Aplicación de Filtros:** Se aplican filtros pasaalto (HPF) y pasabajo (LPF) a las señales escaladas. En función de donde se va a utilizar la señal muestreada es que filtro se debe utilizar. Por ejemplo, para el caso de THD y armónicos, sólo se utiliza el HPF. Pero para el cálculo de cruces por cero, y frecuencias, se utiliza la señal filtrada por ambos filtros.

4. **Detección de Cruce por Cero:** Se realiza una detección de cruce por cero en la señal de voltaje filtrada. Si el cruce ocurre dentro de un rango específico de muestras (definido por **N\_PLOT** y **N\_VENT**), es decir entre 44,4 Hz y 57,1 Hz se registra como un evento de cruce por cero válido, y se reinicia el contador de muestras (**i\_cxc**).
5. **Cálculo de Diferencia de Fase:** Si se detecta un cruce por cero en la señal de voltaje, se espera detectar un cruce por cero en la señal de corriente dentro de ciertos rangos de muestras. Si esto sucede, se calcula la diferencia de muestras entre los cruces, lo que implica una diferencia de fase entre las señales de voltaje y corriente. Posteriormente usado para el cálculo de grados de desfase.
6. **Gestión de Índices y Variables:** El índice **i** se incrementa y se maneja modularmente con **N\_MUESTRAS** para circular a través de un búfer de almacenamiento. Se utilizan variables como **v\_cruce\_cero** para manejar el estado del análisis entre iteraciones del bucle.

### 3.5 Método de detección de cruce por cero y cálculo de frecuencia

La medición precisa de la frecuencia de una señal senoidal requiere la detección del período de la señal, que es el inverso de la frecuencia. En un contexto donde la señal puede estar contaminada por ruido, la detección del cruce por cero se convierte en una herramienta esencial para determinar el período con exactitud. En el sistema descrito, se detectan únicamente los cruces por cero ascendentes, que son los momentos en los que la señal cambia de un valor negativo a un valor positivo.

Para implementar esta técnica, se realiza un muestreo de la señal cada 312,5  $\mu$ s. Si en un muestreo el voltaje pasa de un valor negativo a un valor positivo, se considera que ha ocurrido un cruce por cero. Sin embargo, para aumentar la robustez del sistema contra el ruido, la señal se filtra primero a través de un filtro pasa bajos con una frecuencia de corte de 500 Hz antes de realizar la detección de cruce por cero.

Se establece una ventana de tiempo de  $20 \pm 2,5$  ms para aceptar un cruce por cero, lo que corresponde a un rango de frecuencias de 44,4 a 57,1 Hz, teniendo en cuenta las fluctuaciones comunes en la frecuencia de la red eléctrica. Cuando se detecta un cruce por cero, el sistema registra:

- El número de tiempos de muestreo **N** desde el último cruce por cero detectado.
- El valor de la señal negativa  $v_{neg}$  justo antes del cruce por cero.
- El valor de la señal positiva  $v_{pos}$  en el punto exacto del cruce por cero.

Con estos datos, es posible calcular la frecuencia de la señal de manera precisa. Este método mejora la fiabilidad de la medición de la frecuencia y minimiza los efectos del ruido

Luego se almacenan en una matriz (un búfer circular para 100 cruces por cero) de clase "cxc" (cruces por cero).

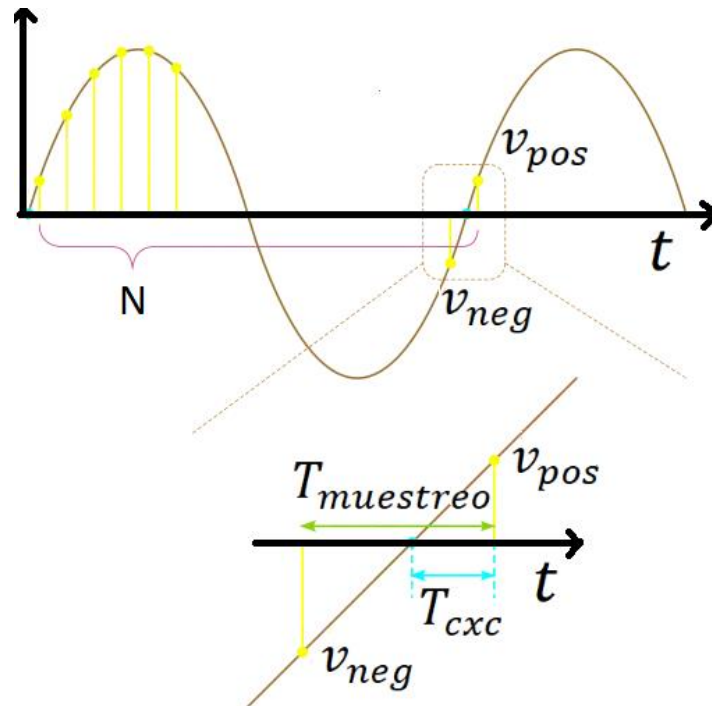


Figura 23 Método de detección de cruce por cero

El verdadero cruce por cero no se detecta debido al muestreo periódico constante. El tiempo  $T_{cxc}$ , tiempo en el que cruza por cero se puede estimar mediante la siguiente fórmula:

$$T_{cxc} = \frac{v_{positivo}}{v_{positivo} - v_{negativo}} * T_{muestreo}$$

$V_n < 0$ , por lo que el denominador es positivo.

Para calcular el tiempo  $T_M$  que corresponde a  $M$  cruces por cero (o  $M$  ciclos completos de una señal de corriente alterna), se emplea una técnica de medición precisa que contabiliza el intervalo temporal entre cruces por cero consecutivos. En el contexto descrito,  $M$  es igual a 100, lo que indica que la medición se extiende a través de 100 ciclos de la señal de CA.

La fórmula para calcular  $T_M$  se basa en la suma acumulativa de los productos de  $N$  (el número de muestreos entre dos cruces por cero consecutivos) y  $T_{muestreo}$  (el período de muestreo fijo, que en este caso es de 312,5  $\mu$ s). Esto se representa matemáticamente como:

$$T_M = \sum_{i=0}^{M-1} T_{muestreo} * N[i]$$

Sin embargo, para una precisión adicional, también se hace un ajuste añadiendo el tiempo al primer cruce por cero detectado y restando el tiempo al último cruce por cero detectado. Esto corrige cualquier desfase en la detección del cruce por cero debido al tiempo de muestreo. La fórmula ajustada quedaría:

$$T_M = \sum_{i=0}^{M-1} T_{muestreo} * N[i] + T_{cxc,0} - T_{cxc,M-1}$$

Este método de medición de tiempo permite calcular con precisión la frecuencia de la señal de CA, ya que la frecuencia ( $f$ ) es el recíproco del período de un ciclo ( $T$ ), y al trabajar con múltiples ciclos se mejora la exactitud de la medición de la frecuencia. La frecuencia se puede expresar como:

$$f = \frac{M}{T_M}$$

Donde  $M$  es el número total de ciclos medidos y  $T_M$  es el tiempo total acumulado para esos ciclos. Este enfoque reduce significativamente el error de medición que podría resultar de un solo ciclo, especialmente en presencia de ruido o variaciones transitorias en la señal de CA.

A continuación, se observa parte del código implementado para cálculo de los cruces por cero.

```

i_cxc++;
int i_pre;
i_pre = i > 0 ? i - 1 : N_MUESTRAS - 1;
if (v_lpf[i_pre] < 0 && v_lpf[i] >= 0) //entra si tension cruza por cero
{
    // Check zero-crossing timing
    if (i_cxc > N_PLOT - N_VENT && i_cxc < N_PLOT + N_VENT) // N_PLOT = 64; N_VENT = 8;
    {
        // Save zero-crossing information
        v_cruce_cero = true;
        cxc[k].N = i_cxc;
        cxc[k].V_neg = v_lpf[i_pre];
        cxc[k].V_pos = v_lpf[i];
        k = (k + 1) % N_CXC;
    }
    i_cxc = 0;
}

// Calculo de frecuencia con la informacion del cruce por cero-----
float calcFreq()
{
    float T = 0;

    T += cxc[0].V_pos / (cxc[0].V_pos - cxc[0].V_neg);
    for (int j = 1; j < N_CXC; j++)
    {
        T += cxc[j].N;
    }
    T -= cxc[N_CXC - 1].V_neg / (cxc[N_CXC - 1].V_pos - cxc[N_CXC - 1].V_neg);
    T *= T_SAMPLE_US * 1e-6;

    float f = 1.0 / T * (N_CXC - 1);
    return f;
}

```

### 3.6 Cálculo de valores rms

El cálculo de la tensión eficaz o valor RMS (Root Mean Square) es un componente esencial en el análisis de señales de corriente alterna, ya que proporciona una medida de la potencia promedio entregada por la señal. Para calcular el valor RMS ( $V_{rms}$ ) de la señal de corriente alterna después de aplicar un filtro pasa bajos ( $V_{LPF}$ ), se emplea la siguiente fórmula matemática:

$$V_{rms} = \sqrt{\frac{1}{N} * \sum_{i=0}^{N-1} (V_{LPF}[i])^2}$$

En esta fórmula, N representa el número total de muestras tomadas durante el periodo de muestreo, que para 8 ciclos de una señal de 50 Hz sería de 512 muestras considerando que se muestrea a una frecuencia superior a la frecuencia de la señal para cumplir con el teorema de Nyquist.

La suma de los cuadrados de cada muestra de la señal filtrada por el filtro pasabajos ( $V_{LPF}$ ) proporciona el total de potencia cuadrática en el intervalo muestreado. Dividir este total por N da el promedio de potencia por muestra. La raíz cuadrada del resultado proporciona el valor RMS, que equivale a la tensión de una fuente de corriente continua que entregaría la misma cantidad de potencia a una carga resistiva.

Es importante recalcar que el valor RMS calculado de esta manera refleja con precisión la potencia real si la señal ha sido correctamente filtrada para eliminar el ruido y otras componentes de alta frecuencia que podrían alterar la medición. Por ello, el uso del filtro pasa bajos antes del cálculo RMS es crucial para obtener una lectura precisa.

```
// Calculo de VRMS-----
float calcRMS_V()
{
    float V_rms = 0;
    for (int j = 0; j < N_MUESTRAS; j++)
    {
        V_rms += pow(v_lpf[j], 2);
    }
    V_rms /= N_MUESTRAS;
    V_rms = sqrt(V_rms);
    return V_rms;
}
```

### 3.7 Cálculo de espectro y THD

La señal  $V_{HPF}$  después de HPF se almacena en el búfer circular durante 8 ciclos. Como contramedida de ruido, se creó una matriz de tipo `complex<float>` promediando 8 ciclos (512 puntos) para 1 ciclo (64 puntos) y llenando la parte imaginaria con cero, y se pasó como un puntero a la función FFT la cual realizará el cálculo correspondiente.

Dado que solo hay 64 puntos de datos, solo se puede obtener el orden 0 (CC), el orden 1 (onda fundamental) y el orden 31 de armónicos. Teniendo en cuenta que el componente de CC es cero porque se aplica el HPF. Para cada uno de los órdenes 2 y superiores, la relación con el orden 1 (onda fundamental) se expresa como un porcentaje y se almacena en una matriz separada. Además, la THD (distorsión armónica total) se calcula a partir de la siguiente fórmula (hasta el armónico 31).

$$THD = \frac{\sqrt{\sum_{k=2}^{31} V_k^2}}{V_1}$$

Si la frecuencia fluctúa, la premisa de "64 muestras = 1 ciclo" se romperá, pero por la misma razón que en el caso anterior del valor eficaz, se decidió no tenerlo en cuenta. Dado que la frecuencia debe estar entre 49,9 y 50,1 Hz la mayor parte del tiempo, la diferencia entre  $312,5 \mu s \times 64 = 20 \text{ ms}$  y un ciclo es de aproximadamente  $40 \mu s = 0,128$  muestras. Por lo tanto, creemos que no afecta tanto el cálculo de FFT.

### 3.7.1 Transformada de Fourier

La Transformada de Fourier es una herramienta matemática que descompone una señal en el dominio del tiempo en sus componentes de frecuencia. Esto es esencial para entender y analizar señales en términos de sus componentes espectrales. Existen varias formas de la Transformada de Fourier, siendo la Transformada de Fourier Continua (FTC) y la Transformada Discreta de Fourier (DFT) dos de las más comunes. Aquí nos centraremos principalmente en la DFT.

### 3.7.2 Transformada Discreta de Fourier (DFT)

La DFT se aplica a señales discretas, es decir, aquellas que están muestreadas en intervalos regulares. Para una señal discreta  $x(n)$  de longitud  $N$ , la DFT se define mediante la siguiente fórmula.

Ecuación fundamental de análisis de DFT de  $N$  muestras.

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N} = \frac{1}{N} \sum_{n=0}^{N-1} x(n)[\cos(2\pi nk/N) - j \sin(2\pi nk/N)]$$

- $X(k)$  es la componente en frecuencia en el índice  $k$
- $X(n)$  es el valor de la señal en el instante de tiempo discreto  $n$
- $N$  es longitud de la señal
- "j" unidad imaginaria.

La DFT convierte una señal discreta en el dominio del tiempo  $x(n)$  en una representación en el dominio de la frecuencia  $X(k)$ . Cada  $X(k)$  indica la contribución de la frecuencia  $k$  en la señal original.

La DFT produce un conjunto de coeficientes complejos  $X(k)$  donde la magnitud de cada coeficiente  $|X(k)|$  representa la amplitud de la componente de frecuencia correspondiente, y el ángulo  $\arg X(k)$  indica la fase. La DFT se organiza comúnmente en un gráfico conocido como espectro de amplitud, que representa cómo se distribuyen las amplitudes a lo largo de las diferentes frecuencias. De esta forma es como obtenemos los armónicos para realizar el análisis espectral de tensión y corriente de nuestro proyecto.

Los armónicos son componentes frecuenciales que aparecen en una señal periódica y que son múltiplos enteros de la frecuencia fundamental. En otras palabras, si una señal tiene una frecuencia fundamental  $f_0$ , los armónicos estarán en  $2f_0$ ,  $3f_0$ ,  $4f_0$  y así sucesivamente.

Cuando se trabaja con transformadas de Fourier discretas (DFT) a partir de una señal analógica muestreada, como es el caso de la señal que se obtiene al utilizar el ADC de la ESP32, es crucial entender el concepto de frecuencia de muestreo y cómo se relaciona con los armónicos. Aquí hay algunas claves para entender esta relación:

1. **Teorema de Muestreo de Nyquist:** Según el teorema de Nyquist, para evitar la pérdida de información al muestrear una señal analógica, la frecuencia de muestreo  $f_s$  debe ser al menos el doble de la frecuencia máxima en la señal analógica  $f_{max}$ . Es decir,  $f_s \geq 2 * f_{max}$
2. **Frecuencia de Nyquist:** La frecuencia de Nyquist  $f_{Nyquist}$  es la mitad de la frecuencia de muestreo  $f_{Nyquist} = \frac{f_s}{2}$ . Es el límite superior de frecuencia que se puede representar de manera única en una señal muestreada. Es decir, no se podrá obtener información de más del doble de la frecuencia de muestreo.
3. **Relación con los Armónicos:** En el contexto de una DFT aplicada a una señal muestreada, los armónicos de la señal original y sus frecuencias aparecerán reflejados alrededor de la frecuencia de Nyquist. Este fenómeno se conoce como el "plegado espectral" o "aliasing". Si un armónico cae por encima de la frecuencia de Nyquist, se pliega de nuevo dentro del rango y puede superponerse con otros componentes espectrales. Por eso es importante respetar la premisa explicada en el inciso 1.
4. **Muestreo y Resolución Espectral:** El número de puntos en la DFT está relacionado con la resolución espectral. Cuantos más puntos en la DFT, mayor será la resolución y mejor será la capacidad para distinguir entre componentes espectrales cercanos. Sin embargo, también es importante recordar que la resolución espectral está directamente relacionada con la duración de la señal que estás analizando.

Al trabajar con transformadas de Fourier discretas en señales analógicas muestreadas, es fundamental considerar la frecuencia de muestreo y cómo afecta la representación de los armónicos. Un muestreo inadecuado puede llevar a problemas de "aliasing" y distorsionar la representación espectral de la señal. Asegurarse de que la frecuencia de muestreo cumple con el teorema de Nyquist es esencial para obtener una representación precisa de los componentes frecuenciales de la señal muestreada.

Teniendo esto en consideración, vamos a explicar algunas de las decisiones que se tomaron a la hora de realizar el algoritmo de cálculo de la DFT.

Sabemos que la frecuencia que vamos a muestrear se encuentra en el rango cercano a los 50 Hz, ya que la red eléctrica domiciliaria debe por norma encontrarse cercana a este valor. Es por esta razón que se estableció una ventana de  $20\text{mS} \pm 2,5 \text{ mS}$ , que representa un rango de 44,4 Hz a 57,14 Hz.

Por lo que la frecuencia de muestreo se estableció en 3200 Hz ( $T_s=312 \mu\text{S}$ ). Cumpliendo con el teorema de Nyquist para evitar que se produzca Aliasing, y tener una cantidad considerable de muestras por periodo. Teniendo en cuenta la frecuencia de muestreo podemos observar que por período podemos encontrar 64 muestras.

### 3.7.3 Algoritmo Cooley-Tukey, método de la mariposa

El algoritmo Cooley-Tukey es un algoritmo eficiente para calcular la Transformada Rápida de Fourier (FFT). Esta técnica divide recursivamente una DFT (Transformada Discreta de Fourier) de tamaño  $N$  en dos DFT más pequeñas de tamaño  $N/2$  y luego combina los resultados de manera eficiente utilizando el método de la mariposa (butterfly). Es por esta razón que se han desarrollado métodos computacionales para aprovechar algunas características de la DFT, eliminando operaciones redundantes. Principalmente se aprovechan las características de Periodicidad y Simetría de los coeficientes.

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{-kn} \quad \text{Ecuación general} \quad \text{Donde: } W_N = e^{-2\pi i/N} \text{ Operador rotación}$$

Recordando la ecuación general de la DFT vemos 2 sumas realizadas, reduciendo el número de muestras a la mitad y separando la suma en pares a impares multiplicando por el operador rotacional.

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x_{10}(n)W_N^{-2kn} + \sum_{n=0}^{\frac{N}{2}-1} x_{11}(n)W_N^{-2kn+1} \quad \text{se separa la parte par de la impar}$$

$$x_{10}(n) = x(2n) \quad \text{Pares} \quad x_{11}(n) = x(2n + 1) \quad \text{Impares} \quad 0 \leq n \leq \frac{N}{2} - 1$$

$$W_N^{-2kn} = W_{\frac{N}{2}}^{-kn}$$

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x_{10}(n)W_{\frac{N}{2}}^{-kn} + W_N^{-k} \sum_{n=0}^{\frac{N}{2}-1} x_{11}(n)W_{\frac{N}{2}}^{-kn}$$

$$X(k) = X_{10}(k) + W_N^{-k}X_{11}(k) \quad (1)$$

Como  $X_{10}(k)$  y  $X_{11}(k)$  son periódicas con periodo  $\frac{N}{2}$

$$X\left(k + \frac{N}{2}\right) = X_{10}\left(k + \frac{N}{2}\right) + W_N^{-k+\frac{N}{2}}X_{11}\left(k + \frac{N}{2}\right)$$

$$\text{Si: } W_N = e^{-2\pi j/N}$$

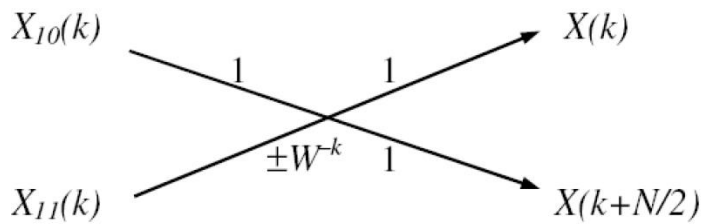
$$W_N^{k+\frac{N}{2}} = e^{-\frac{j2\pi}{N}\left(k+\frac{N}{2}\right)} = e^{-\frac{j2\pi}{N}k} e^{-j\pi}$$

$$\text{Por Euler sabemos que } e^{-j\pi} = -1 \quad W_N^{k+\frac{N}{2}} = -W_N^k$$

$$X\left(k + \frac{N}{2}\right) = X_{10}(k) - W_N^{-k}X_{11}(k) \quad (2)$$

Las Ecuaciones (1) y (2) pueden representarse con el siguiente gráfico que se asemeja a una "mariposa". De allí su nombre.





Lo mismo se puede realizar con  $X_{10}(k) = \sum_{n=0}^{N-1} x(n)W_{N/2}^{-kn}$  que se puede dividir en dos sumatorias.

$$X_{10}(k) = \sum_{n=0}^{\frac{N}{4}-1} x_{20}(n)W_{\frac{N}{4}}^{-kn} + W_N^{-2k} \sum_{n=0}^{\frac{N}{4}-1} x_{21}(n)W_{\frac{N}{4}}^{-kn}$$

$$x_{20}(n) = x_{10}(2n) \quad \text{Pares} \quad x_{21}(n) = x_{10}(2n + 1) \quad \text{Impares}$$

Esta secuencia se puede realizar  $N/2$  veces.

Para explicar el cálculo del método de la mariposa para una Transformada Rápida de Fourier (FFT) con  $N=8$  muestras, desglosamos el proceso paso a paso. Este método se basa en el algoritmo de Cooley-Tukey, que es una forma eficiente de computar la FFT mediante la descomposición de la transformada de Fourier de tamaño  $N$  en transformadas de menor tamaño.

### Paso 1: División en Problemas más Pequeños

El primer paso del algoritmo es dividir el problema inicial de tamaño  $N=8$  en problemas más pequeños. Esto se logra separando las muestras en grupos de pares e impares:

- Para  $N=8$ , se divide en dos grupos de  $N/2=4$ .
- Las muestras en posiciones pares (incluyendo el 0) forman un grupo y las muestras en posiciones impares forman otro.

### Paso 2: Aplicación Recursiva de la FFT

Cada uno de estos grupos de 4 muestras se somete de nuevo al mismo proceso, dividiéndose a su vez en grupos más pequeños de 2 muestras hasta llegar a problemas de tamaño 1, donde la FFT de una muestra es la muestra misma.

- **Primera Aplicación Recursiva (N=4 para pares):** Se divide nuevamente en dos grupos de 2 (posiciones 0 y 2 en un grupo, y 4 y 6 en otro).
- **Segunda Aplicación Recursiva (N=4 para impares):** Similarmente, las muestras en las posiciones 1, 3, 5, y 7 se dividen en dos grupos de 2 (posiciones 1 y 3, por otro lado 5 y 7).

### Paso 3: Cálculo de la FFT en Problemas de Tamaño 1

Cuando se llega a grupos de tamaño 1, el cálculo de la FFT es directo, ya que la FFT de una muestra es simplemente la muestra misma. Este es el caso base de la recursión.

### Paso 4: Combinación usando el Método de la Mariposa

La combinación de los resultados de los problemas más pequeños se realiza utilizando el método de la mariposa. Este método calcula la FFT del conjunto original de muestras a partir de las FFTs de los subconjuntos.

- **Aplicación del Método de la Mariposa:** Para cada par de resultados de FFTs de tamaño menor, se calculan dos nuevos resultados mediante una combinación lineal, utilizando coeficientes conocidos como "factores de rotación" ( $W_N^{-k}$ ), que son raíces de la unidad.
- **Primera Combinación:** Por ejemplo, para dos FFTs de tamaño 2, se combinan para formar una FFT de tamaño 4. Esto se repite para todos los grupos.
- **Segunda Combinación:** Las FFTs de tamaño 4 (de los pasos anteriores) se combinan para formar la FFT final de tamaño 8, completando el proceso.

### Paso 5: Ordenamiento Final

Finalmente, se realiza un ordenamiento que asegura que las salidas de la FFT estén en el orden correcto. Este proceso se conoce como "bit reversal" o inversión de bits.

### Visualización del Método de la Mariposa con Valores reales

Para una FFT de  $N=8$ , el método de la mariposa implica la realización de cálculos específicos que mezclan las muestras de entrada de manera que al final, después de aplicar las operaciones de mariposa y los factores de rotacionales apropiados, se obtiene la FFT deseada. Este proceso reduce significativamente el número de operaciones necesarias en comparación con la definición directa de la DFT, aprovechando las simetrías y periodicidades de las raíces de la unidad.

Para mostrar cómo cada muestra  $N$  se relaciona con su respectivo factor rotacional en una FFT de tamaño  $N=8$ , primero debemos entender qué es un factor rotacional y cómo se calcula.

Un factor rotacional es una raíz de la unidad y se calcula como:

$$W_N = e^{\frac{-j2\pi k}{N}}$$

donde:

- $N$  es el número total de muestras,
- $k$  es el índice específico para el cual se calcula el factor,
- $j$  es la unidad imaginaria.

Para una FFT de  $N=8$ , los factores rotacionales se calculan para  $k=0$  hasta  $k=N-1$ , pero debido a la simetría y periodicidad de las raíces de la unidad, en la práctica, se usan solo los necesarios según la etapa de combinación en el algoritmo.

### Paso 1: Primer Nivel de Descomposición

Para  $N=8$ , las muestras se dividen inicialmente en grupos de tamaño  $N/2=4$ . En este nivel, no necesitamos aplicar factores rotacionales ya que estamos solo dividiendo el problema.

### Paso 2: Segundo Nivel de Descomposición

En el segundo nivel, trabajamos con grupos de  $N/4=2$  muestras. Aquí comenzamos a aplicar los factores rotacionales al combinar las FFTs más pequeñas.

### Paso 3: Cálculo de los Factores Rotacional

Vamos a calcular y relacionar los factores rotacionales para la combinación final que forma la FFT completa de  $N=8$ . Para una FFT de 8 puntos, los factores se utilizan cuando combinamos los resultados de las FFTs de tamaño menor.

En una FFT de 8 puntos, los factores rotacional relevantes para la etapa final de combinación son  $W_8^0, W_8^1, W_8^2$  y  $W_8^3$  que estos se aplican a las operaciones de la mariposa para combinar FFTs de tamaño 4 en la FFT final de tamaño 8.

$$W_8^k = e^{\frac{-j\pi k}{4}}$$

Para  $k=0,1,2,3$ , tenemos:

$$W_8^0 = e^{\frac{-j\pi 0}{4}} = 1$$

$$W_8^1 = e^{\frac{-j\pi 1}{4}} = \cos\left(\frac{\pi}{4}\right) - j \sin\left(\frac{\pi}{4}\right)$$

$$W_8^2 = e^{\frac{-j\pi 2}{4}} = \cos\left(\frac{\pi}{2}\right) - j \sin\left(\frac{\pi}{2}\right)$$

$$W_8^3 = e^{\frac{-j\pi 3}{4}} = \cos\left(\frac{3\pi}{4}\right) - j \sin\left(\frac{3\pi}{4}\right)$$

### Paso 4: Aplicación de los Factores Rotacional

En la combinación final para  $N=8$ , estos factores rotacionales se aplican de la siguiente manera:

- Las muestras en la primera mitad de la FFT (índices 0 a 3) se combinan con sus contrapartes en la segunda mitad (índices 4 a 7) usando los factores rotacionales correspondientes.
- Por ejemplo, para combinar las dos FFTs de 4 puntos en una de 8 puntos, la muestra en la posición  $n$  de la primera FFT se combina con la muestra en la posición  $n$  de la segunda FFT multiplicada por  $W_8^n$ .

Esto significa que, para cada par de muestras que se combinan, se utiliza un factor rotacional específico para ajustar la fase de la muestra de la segunda mitad antes de sumarla o restarla de la muestra correspondiente en la primera mitad, completando así la FFT de 8 puntos.

Vamos a calcular los valores exactos de  $W_8^1, W_8^2$  y  $W_8^3$  para ver cómo se aplican en términos de sus componentes reales e imaginarios.

Los valores de los factores rotacional para  $k=1,2,3$  en términos de sus componentes reales e imaginarios son:

- $W_8^1 = 0,7071 - j0,7071$
- $W_8^2 = 0 - j1,0$  (aproximadamente 0 debido a la precisión numérica, pero conceptualmente es 0)
- $W_8^3 = -0,7071 - j0,7071$

Estos factores se utilizan para ajustar las fases de las muestras en la segunda mitad de la FFT antes de su combinación con las muestras de la primera mitad, siguiendo el método de la mariposa. Cada factor rotacional ajusta la muestra de la segunda mitad multiplicando por:

- $W_8^1$  para la primera combinación de mariposa,
- $W_8^2$  para la segunda combinación, y
- $W_8^3$  para la tercera combinación.

Por ejemplo, en la operación de la mariposa para el índice  $k=1$ , la muestra en el índice 1 se combina con la muestra en el índice 5 después de multiplicar esta última por  $0,7071-j0,7071$ , ajustando así su fase y amplitud antes de sumarla o restarla con la muestra en el índice 1. Similarmente, se procede con los otros índices utilizando sus respectivos factores rotacionales.

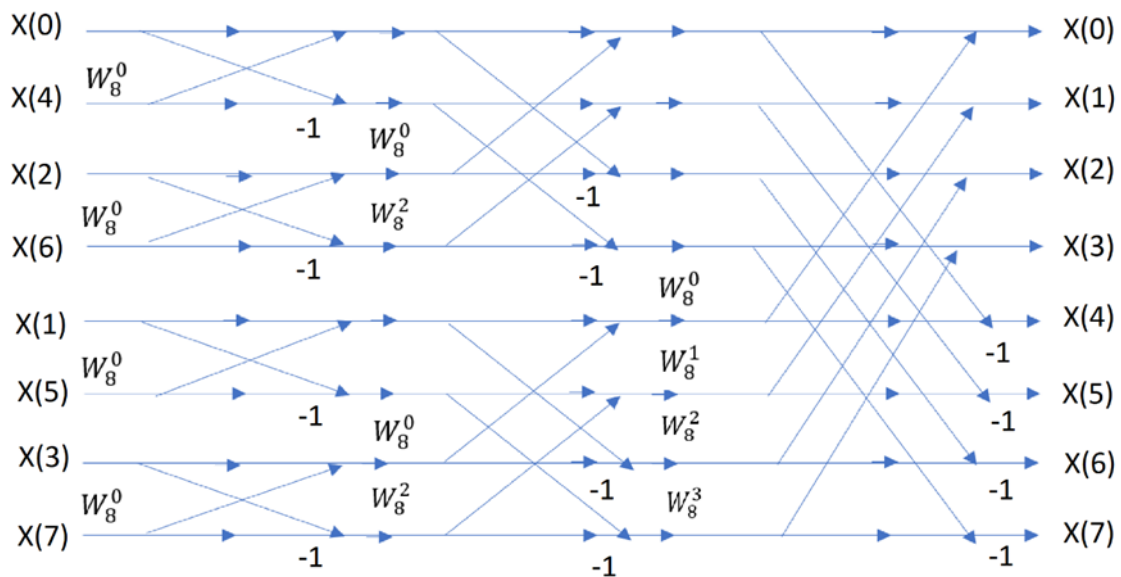


Figura 24 Ejemplo genérico de etapas mariposa de  $N=8$

A continuación, se explican los pasos del algoritmo implementado:

**1. División:**

- Dada una secuencia de datos de longitud  $N$ , el algoritmo comienza dividiendo la secuencia en dos partes iguales.
- Cada mitad representa una DFT de longitud  $N/2$ .

```
int n = N / 2;
```

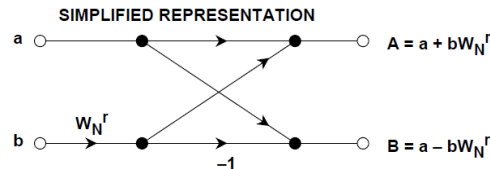
**2. Cálculo del operador de rotación:**

- Se calcula el operador de rotación  $W_N$  que se utiliza en el cálculo de las mariposas. Este operador se define como  $W_N = e^{-2\pi i/N}$ .

```
Complex W; // Se declara una variable de tipo complejo
W.polar(1.0, -2 * M_PI * i / N); // se calcula operador
```

**3. Cálculo de mariposas:**

- Se realiza el cálculo de las mariposas para combinar los resultados de las DFT parciales de las dos mitades.
- En cada nivel de recursión, se realiza la operación de mariposa en pares de puntos de la secuencia.
- La operación de mariposa combina dos puntos usando el operador de rotación  $W_N$ .



Lo que se busca es descomponer una secuencia de  $N$  muestras, en  $N$  secuencias de 1 muestra.

```
// Calculo por método Mariposa
Complex f_tmp = f[i] - f[n + i]; //se guarda en tmp la resta de un
par menos un impar
f[i] += f[n + i]; // se sobrescribe con la suma del par y el impar
f[n + i] = W * f_tmp; // Se multiplica por el factor rotacional
```

#### 4. Llamada recursiva:

- Se realiza una llamada recursiva a la función para las dos mitades de la secuencia.
- Este paso se repite hasta que la longitud de la secuencia es 1, en cuyo caso no se realizan más divisiones.

```
// Llamada recursiva a la función
FFT_1(&f[0], n); // Primera mitad pares incluyendo el cero
FFT_1(&f[n], n); // Segunda mitad impares
```

#### 5. Permutación:

- Después de completar las llamadas recursivas, se realiza una permutación simple de los resultados para reorganizar la secuencia utilizando la reversión de bits (bits reversal). Recordemos que las variables leídas por el ADC se leen en orden y se guardan en la memoria, por lo que luego del uso de la mariposa es necesario reordenar las variables para seguir respetando el orden original. Cambiando los "0" por los "1".

Tabla 4 Representación de método de inversión de bits FFT

Sample numbers in normal order		Sample numbers after bit reversal	
Decimal	Binary	Decimal	Binary
0	0000	0	0000
1	0001	8	1000
2	0010	4	0100
3	0011	12	1100
4	0100	2	0010
5	0101	10	1010
6	0110	6	0100
7	0111	14	1110
8	1000	1	0001
9	1001	9	1001
10	1010	5	0101
11	1011	13	1101
12	1100	3	0011
13	1101	11	1011
14	1110	7	0111
15	1111	15	1111

```
// Bit reversal o inversión de bits
Complex F[N]; // Se declara un arreglo complex como variable aux.
for (int i = 0; i < n; i++)
{
    F[2 * i] = f[i];
    F[2 * i + 1] = f[n + i];
}
// sobrescribe la salida a la entrada
for (int i = 0; i < N; i++)
{
    f[i] = F[i];
}
```

**6. Resultados finales:**

- Los resultados finales de la FFT se obtienen al final de todas las llamadas recursivas.

El proceso de "mariposa" es fundamental para la eficiencia del algoritmo. Consiste en combinar los resultados de las DFT parciales de dos subsecuencias mediante la multiplicación por el operador de rotación. Este paso se repite en cada nivel de la recursión hasta que se obtiene la FFT completa. Esta técnica reduce significativamente el número de operaciones requeridas para calcular la FFT, convirtiéndola en una de las formas más eficientes de realizar este cálculo. La

cantidad de operaciones mariposa que debe usarse son  $\frac{N}{2} \times \log_2 N$  en lugar de las  $N^2$  requeridas para el cálculo de la DFT. La cantidad de memoria que se necesita para calcular la FFT es de  $2N$ .

### 3.7.4 Implementación algoritmo FFT

```
void FFT_1(Complex *f, int N)
{
    if (N > 1)
    {
        int n = N / 2; // Se divide el Numero de muestras N en dos

        for (int i = 0; i < n; i++)
        {

            Complex W; // Se declara variable de tipo complejo
            W.polar(1.0, -2 * M_PI * i / N); // se calcula operador rotacional

            // Calculo método Mariposa
            Complex f_tmp = f[i] - f[n + i];
            f[i] += f[n + i];
            f[n + i] = W * f_tmp;
        }

        // Llamada recursiva a la función
        FFT_1(&f[0], n); // Primera mitad
        FFT_1(&f[n], n); // Segunda mitad

        / Inversion de bits

        int i = 0;
        for (int j = 1; j < N - 1; j++)
        {
            for (int k = N >> 1; k > (i ^ k); k >>= 1);
            if (j < i)
            {
                Complex f_tmp = f[j];
                f[j] = f[i];
                f[i] = f_tmp;
            }
        }
    }
}
```

### 3.8 Filtros Digitales Butterworth HPF y LPF orden 2

Los filtros Butterworth son conocidos por su característica de tener una respuesta en frecuencia máximamente plana en la banda de paso, lo que significa que no introducen ondulaciones ni variaciones en la amplitud de las señales dentro de esta banda. Esto es especialmente útil en aplicaciones donde se requiere mantener la integridad de la señal original sin distorsiones.

La elección de un filtro Butterworth de orden 2 se justifica por su respuesta en frecuencia plana en la banda de paso, su transición suave hacia la banda de atenuación, la simplicidad de su diseño e implementación, su estabilidad y la previsibilidad de su comportamiento, junto con una fase relativamente lineal. Estas características lo hacen adecuado para aplicaciones donde la calidad y la integridad de la señal son críticas.

Para el diseño del proyecto se crearon dos filtros Butterworth de orden 2, un HPF con frecuencia de corte de 1 Hz y un LPF con frecuencia de corte de 500 Hz.

#### 3.8.1 Propiedades de los filtros

Los filtros digitales son herramientas fundamentales en el campo de la ingeniería electrónica y el procesamiento de señales, con una amplia gama de aplicaciones prácticas. A diferencia de los filtros analógicos, que procesan señales continuas mediante componentes electrónicos físicos (como resistencias, capacitores e inductores), los filtros digitales procesan señales discretas (digitales) mediante operaciones matemáticas ejecutadas en un procesador, un microcontrolador, o un circuito especializado como un DSP (Digital Signal Processor).

La implementación de filtros digitales se realiza a través de algoritmos que definen cómo se modifica la señal de entrada para producir la señal de salida. Estos algoritmos se basan en ecuaciones diferenciales discretas que describen la relación entre la entrada, la salida, y los coeficientes del filtro, que definen su respuesta en frecuencia.

Un filtro digital puede ser representado mediante el siguiente diagrama en bloque



El análisis de un filtro digital es el proceso de determinar la respuesta de un filtro ante una determinada excitación. El diseño de un filtro digital es el proceso de sintetizar e implementar un filtro digital de tal manera que cumpla con las especificaciones prescritas.

En general los filtros digitales deben cumplir con las siguientes características:

- Invariancia en el tiempo
- Causalidad
- Linealidad

Por otra parte, los filtros digitales pueden clasificarse como:

- Filtros no recursivos (también llamados FIR o de respuesta finita al impulso).
- Filtros recursivos (también llamados IIR o de respuesta infinita al impulso).



Ecuación en diferencias del filtro no recursivo:

$$y[n] = \sum_{i=0}^N b_i x[n-i] \quad \text{Donde N: orden del filtro}$$

Ecuación en diferencias del filtro recursivo:

$$y[n] = \sum_{i=0}^N b_i x[n-i] - \sum_{i=1}^N a_i y[n-i]$$

- Invariancia en el tiempo:

Un sistema es invariante en el tiempo si ante el desplazamiento de una dada excitación produce la misma salida, pero afectada por el mismo desplazamiento que la excitación

- Causalidad

Para que un filtro sea causal, es condición indispensable que su salida en un instante no dependa de la entrada en tiempos posteriores.

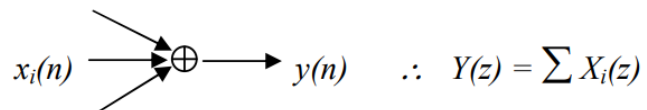
- Linealidad

Deben ser lineales, además deben cumplir con la propiedad de homogeneidad, conmutatividad y aditividad. Una señal puede descomponerse en un grupo de componentes aditivas, pasar cada una por un sistema lineal y sumar las salidas individuales para obtener la salida total (síntesis).

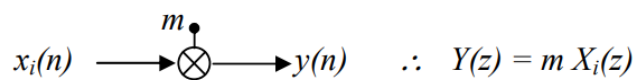
Redes de Filtrado:

Para la representación de un filtro se pueden utilizar combinaciones de los siguientes elementos:

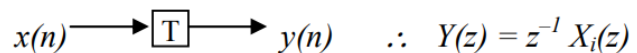
sumador:  $y(n) = \sum x_i(n)$



multiplicador:  $y(n) = m x(n)$



retardo:  $y(n) = x(n - 1)$



En la imagen inferior se puede observar una red que representa un filtro IIR

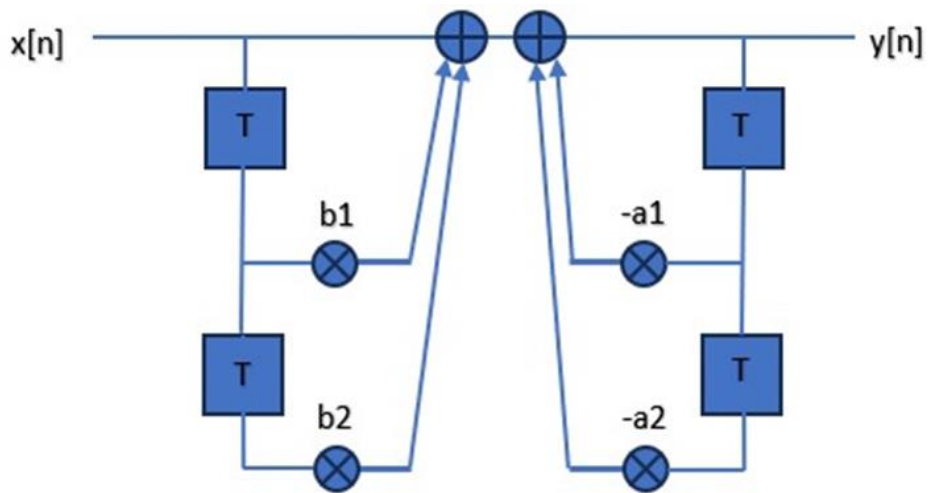


Figura 25 Representación diagrama en bloques filtro dos ramas

Vemos en el ejemplo anterior un diagrama que representa la función transferencia del filtro, donde se observa la rama directa y la rama recursiva. La realización directa requiere una cantidad de retardos del doble del orden del filtro y cada retardo implica un requerimiento de memoria a ser usado en el próximo ciclo. Por lo que puede optarse por usar la forma canónica del filtro, en el que los retardos se reducen a la mitad.

Se puede observar que hay pares de valores repetidos en los retardos, por lo que el diagrama podría simplificarse sin alterar la función transferencia del filtro.

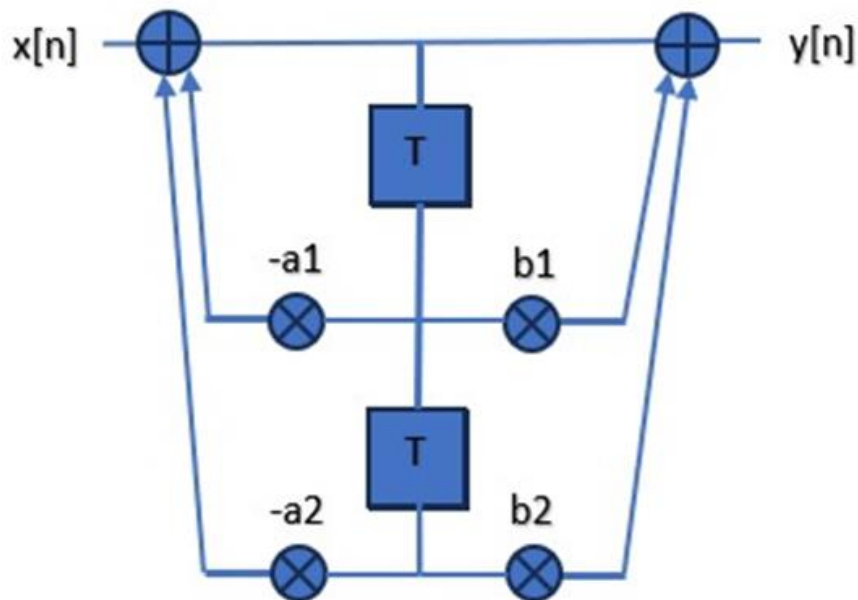


Figura 26 Representación canónica diagrama en bloques filtro

La función transferencia de un filtro analógico puede ser expresado de la siguiente forma:

$$\frac{V_o(s)}{V_i(s)} = H(s) = \frac{N(s)}{D(s)}$$

Siendo  $N(s)$  y  $D(s)$  polinomios en función de  $s = \sigma + j\omega$

- Atenuación en dB:  $A(\omega) = 20 \log \left| \frac{V_i(j\omega)}{V_o(j\omega)} \right| = 20 \log \left| \frac{1}{H(j\omega)} \right| = 10 \log L(\omega^2)$
- Función de atenuación  $L(\omega^2) = \frac{1}{H(j\omega)H(-j\omega)}$
- Desplazamiento de fase:  $\theta(j\omega) = \arg H(j\omega)$
- Retardo de grupo:  $\tau = -\frac{d\theta(j\omega)}{d\omega}$

Para un filtro pasabajos normalizado:

$$A(\omega) = -3dB \quad \omega = 1 [rad/seg]$$

La normalización de la frecuencia es un paso en el diseño de filtros analógicos y digitales que simplifica el proceso de diseño al estandarizar la frecuencia de corte del filtro a un valor conveniente, típicamente 1 rad/s.

Si el filtro debe operar en frecuencias específicas distintas de las normalizadas, es necesario "desnormalizar" el filtro. Esto significa ajustar su diseño para que las características de frecuencia del filtro coincidan con las especificaciones deseadas de la aplicación real.

La desnormalización implica ajustar los parámetros de la función de transferencia del filtro. Este proceso ajusta los polos y ceros de la función de transferencia para reflejar las nuevas frecuencias de operación.

Partiendo de un filtro digital pasabajos normalizado, podemos obtener filtros pasabajos, pasaltos, pasabandas y rechazabandas desnormalizados. Para ello se emplean las siguientes transformaciones.

$$\text{PasaBajos} \rightarrow \text{PasaBajos: } s = \lambda \bar{s}$$

$$\text{PasaBajos} \rightarrow \text{PasaAltos: } s = \frac{\lambda}{\bar{s}}$$

$$\lambda = \frac{\omega_{oN}}{\omega_{oD}}$$

$\omega_{oN}$  representa la frecuencia normalizada

$\omega_{oD}$  representa la frecuencia desnormalizada

Usando la transformada Z, un filtro digital puede ser caracterizado mediante una función transferencia discreta en el tiempo que juega el mismo rol que una función transferencia continua en el tiempo para un filtro analógico. Si se considera un filtro digital causal y recursivo, tal como el siguiente:

$$y(n) = \sum_{i=0}^N a_i x(n-i) - \sum_{i=1}^N b_i y(n-i)$$

Aplicando transformada Z

$$\mathbb{Z}y(n) = \sum_{i=0}^N a_i z^{-i} \mathbb{Z}x(n) - \sum_{i=1}^N b_i z^{-i} \mathbb{Z}y(n)$$

$$Y(z) = X(z) \sum_{i=0}^N a_i z^{-i} - Y(z) \sum_{i=1}^N b_i z^{-i}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^N a_i z^{-i}}{1 + \sum_{i=1}^N b_i z^{-i}}$$

Factoreando:

$$\frac{H_0 \prod_{i=0}^N (z - z_i)}{\prod_{i=1}^N (z - p_i)}$$

Donde  $z_i$  son los ceros y  $p_i$  son los polos

Recordando que el filtro es estable si  $|p_i| < 1$

### 3.8.2 Transformación Bilineal (Método Tustin):

La transformación bilineal se utiliza para mapear filtros analógicos en el plano S al plano Z (dominio del tiempo discreto) mediante una transformación bilineal. La relación entre las variables S (en el dominio analógico) y z (en el dominio discreto) está dada por:

$$s = \frac{2}{T_s} \frac{z - 1}{z + 1}$$

Donde:

- $T_s$  es el período de muestreo (el tiempo entre muestras).

La transformación bilineal tiene como objetivo que la respuesta en frecuencia del filtro digital tenga la misma respuesta en frecuencia que el filtro analógico. Sin embargo, la respuesta en frecuencia del filtro digital presenta algunas diferencias y debe ser adaptada para que ambos se comporten de igual manera.

La transformación bilineal introduce un efecto de distorsión de la frecuencia, conocido como "warping". Para compensarlo y lograr la frecuencia de corte deseada en el diseño digital, necesitamos realizar "prewarping" de la frecuencia de corte analógica  $\omega_c$  antes de aplicar la transformación bilineal:

$$\omega_c = \frac{2}{T} \tan\left(\frac{\Omega_c T}{2}\right)$$

siendo  $\omega_c$  la frecuencia de corte del filtro analógico

siendo  $\Omega_c$  la frecuencia de corte del filtro digital deseada

Para obtener los coeficientes del filtro digital resultante a partir de la transformación bilineal, se puede aplicar esta transformación a los coeficientes del filtro analógico. Vamos a considerar un filtro analógico de Butterworth de segundo orden como ejemplo.

La función de transferencia general de un filtro Butterworth de orden  $n$  en el dominio de Laplace es:

$$H_{aN}(s) = \frac{1}{1 + \sqrt[2]{2} s + s^2}$$

$$H_{aD}(\check{s}) = \frac{1}{1 + \sqrt[2]{2} \frac{s}{\omega_c} + \left(\frac{s}{\omega_c}\right)^2}$$

Donde:

- $\omega_c$  es la frecuencia de corte en el dominio analógico.
- $H_{aN}(s)$  *normalizado*
- $H_{aD}(\check{s})$  *desnormalizado*

Para transformar esto al dominio discreto utilizando la transformación bilineal, sustituimos  $s$  por la expresión de la transformación bilineal:

$$H_d(z) = H_a \frac{2}{T_s} \frac{z-1}{z+1}$$

Para un filtro Butterworth de segundo orden, la función de transferencia digital resultante sería:

$$H_d(z) = \frac{1}{1 + 2 \frac{T_s \omega_c}{z-1} + \left(\frac{T_s \omega_c}{z-1}\right)^2}$$

Donde:

- $T_s$  el período de muestreo.

De esta función de transferencia digital, se pueden extraer los coeficientes del numerador y del denominador para utilizar en la implementación del filtro digital en una forma como la siguiente:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

Los coeficientes específicos  $b_0, b_1, b_2, a_1, a_2$  se obtendrán al expandir la función de transferencia y realizar algunos pasos algebraicos.

Veamos el ejemplo para un filtro Butterworth pasa-bajo de orden 2.

### Paso 1: Especificaciones Iniciales

- Frecuencia de corte ( $f_c$ ): 500 Hz

- Frecuencia de muestreo ( $F_s$ ):  $f_s = \frac{1}{T_s} = \frac{1}{312.5 \mu s} = 3200 \text{ Hz}$
- Factor de amortiguamiento ( $\zeta$ ): 0,707

### Paso 2: Prewarping

Para un filtro digital, primero realizamos el prewarping de la frecuencia de corte analógica para compensar el efecto de la transformación bilineal:

$$\omega = \frac{2}{312.5 \mu s} \tan\left(\frac{2\pi * 500 * 312.5 \mu s}{2}\right) = 3420.87 \text{ rad/s}$$

$$f_c = \frac{\omega}{2\pi} = \frac{3420.87}{2\pi} = 544.44 \text{ Hz}$$

### Paso 3: Función de Transferencia Analógica Normalizada

La función de transferencia analógica de un filtro Butterworth de orden 2 es:

$$H_{aN}(s) = \frac{1}{1 + \sqrt{2}s + s^2}$$

$$H_{aD}(\check{s}) = \frac{1}{1 + \sqrt{2} \frac{s}{\omega_c} + \left(\frac{s}{\omega_c}\right)^2}$$

La última expresión representa la función transferencia desnormalizada.

### Paso 4: Transformación al Dominio Digital

Aplicamos la transformada  $H_d(z) = H_a \frac{z}{T_s} \frac{z-1}{z+1}$  para obtener la función de transferencia digital  $H(z)$ .

$$H_d(z) = \frac{1}{1 + 2 \frac{T_s \omega_c}{z-1} + \left(\frac{T_s \omega_c}{z-1}\right)^2}$$

Ahora solo resta calcular los coeficientes del filtro digital. Operando matemáticamente se llega a las siguientes fórmulas de cálculo de los mismos.

$$M = \frac{4\zeta}{\omega_c T_s} = 2,465$$

$$N = \frac{4}{(\omega_c T_s)^2} = 3,50$$

$$a_1 = \frac{2(1-N)}{1+M+N} = -0,7178$$

$$a_2 = \frac{1-M+N}{1+M+N} = 0,143$$

$$b_0 = \frac{1}{1+M+N} = 0,143$$

$$b_1 = \frac{2}{1+M+N} = 0,287$$

$$b_2 = b_0 = 0,143$$

En el constructor `SecondOrderLPF`, se calculan y almacenan los coeficientes del filtro basados en la transformada bilineal (método de Tustin) utilizando los parámetros dados: el periodo de muestreo `Ts`, el factor de amortiguamiento  $\zeta$  (`zeta`), y la frecuencia natural `omega_n`. También se inicializan los valores anteriores de entrada (`u_old`) y salida (`y_old`) del filtro a cero, lo que es necesario para el primer cálculo de la salida.

La función `apply(float u)` toma una entrada `u` y calcula la salida `y` del filtro utilizando los coeficientes calculados durante la inicialización y los valores antiguos de entrada y salida almacenados. Luego actualiza los valores antiguos con los actuales para su uso en el próximo cálculo. Este proceso se repite cada vez que se llama a `apply` con una nueva entrada, lo que hace que el filtro procese la señal de entrada muestra por muestra de manera iterativa.

### 3.8.3 Implementación Filtro Pasabajo orden 2

```
// Second-order low-pass filter class
class SecondOrderLPF {
public:
    SecondOrderLPF(float Ts, float zeta, float omega_n);
    ~SecondOrderLPF(void);
    float apply(float u);

private:
    float m, n;
    float a1, a2, b0, b1, b2;
    float y_old[2], u_old[2];
};

*/
// Constructor
SecondOrderLPF::SecondOrderLPF(float Ts, float zeta, float omega_n) {
    // Calculo de coeficientes a partir de la transformacion bilineal
    m = 4.0 * zeta / (omega_n * Ts);
    n = 4.0 / pow(omega_n * Ts, 2.0);

    a1 = 2.0 * (1.0 - n) / (1.0 + m + n);
    a2 = (1.0 - m + n) / (1.0 + m + n);
    b0 = 1.0 / (1.0 + m + n);
    b1 = 2.0 / (1.0 + m + n);
    b2 = b0;

    // Inicializacion de valores anteriores
    y_old[0] = y_old[1] = 0;
    u_old[0] = u_old[1] = 0;
}
// Destructor
SecondOrderLPF::~SecondOrderLPF(void) {}
```

```

// Aplica filtro digital
float SecondOrderLPF::apply(float u) {
    float y = b0 * u + b1 * u_old[0] + b2 * u_old[1] - a1 * y_old[0] - a2 *
y_old[1];
    y_old[1] = y_old[0]; // Actualiza salidas anteriores
    y_old[0] = y;
    u_old[1] = u_old[0]; // Actualiza retardos anteriores
    u_old[0] = u;
    return y;
}

```

### 3.8.4 Implementación filtro pasa altos HPF de orden 2

De manera similar se calcula el filtro digital HPF de orden 2 con frecuencia de corte 1 Hz, sus coeficientes y se presenta el código implementado. El algoritmo funciona de manera análoga al del LPF, la diferencia principal radica en la ecuación de transferencia que caracteriza al filtro HPF en el dominio de Laplace, a partir desde donde se obtiene la ecuación transferencia en dominio Z. Como se puede observar cambian los coeficientes que conforman la función transferencia del filtro.

$$H_{AD}(s) = \frac{s^2}{1 + \sqrt[2]{2} \frac{s}{\omega_c} + \left(\frac{s}{\omega_c}\right)^2}$$

$$H_{HP}(z) = \frac{(z-1)^2}{(z+1)^2 - \sqrt[2]{2} \frac{\omega_c}{T_s} (z+1) + \frac{\omega_c^2}{T_s^2}}$$

Cálculo frecuencia de corte

$$\omega = \frac{2}{312.5\mu S} \tan\left(\frac{2\pi * 1 * 312.5\mu S}{2}\right) = 6.283 \text{ rad/s}$$

$$f_c = \frac{\omega}{2\pi} = \frac{6.283}{2\pi} = 1.00 \text{ Hz}$$

$$M = \frac{4\zeta}{\omega_n T_s} = 2,465$$

$$N = \frac{4}{(\omega_n T_s)^2} = 3,50$$

$$a_1 = \frac{2(1-N)}{1+M+N} = -0,7178$$

$$a_2 = \frac{1-M+N}{1+M+N} = 0,1435$$

$$b_0 = \frac{N}{1+M+N} = 0,50$$

$$b_1 = \frac{-2N}{1+M+N} = -1$$

$$b_2 = b_0 = 0,5$$



```

// Second-order high-pass filter class
class SecondOrderHPF {
public:
    SecondOrderHPF(float Ts, float zeta, float omega_n);
    ~SecondOrderHPF(void);
    float apply(float u);

private:
    float m, n;
    float a1, a2, b0, b1, b2;
    float y_old[2], u_old[2];
};
*/
// Constructor
SecondOrderHPF::SecondOrderHPF(float Ts, float zeta, float omega_n) {
    // Calculo de coeficientes basados en la transformación bilineal
    m = 4.0 * zeta / (omega_n * Ts);
    n = 4.0 / pow(omega_n * Ts, 2.0);

    a1 = 2.0 * (1.0 - n) / (1.0 + m + n);
    a2 = (1.0 - m + n) / (1.0 + m + n);
    b0 = n / (1.0 + m + n);
    b1 = -2.0 * n / (1.0 + m + n);
    b2 = b0;

    // Inicializacion de valores anteriores
    y_old[0] = y_old[1] = 0;
    u_old[0] = u_old[1] = 0;
}

// Destructor
SecondOrderHPF::~SecondOrderHPF(void) {}

// Aplica filtro digital
float SecondOrderHPF::apply(float u) {
    float y = b0 * u + b1 * u_old[0] + b2 * u_old[1] - a1 * y_old[0] - a2 *
y_old[1];
    y_old[1] = y_old[0]; // Actualiza salidas anteriores
    y_old[0] = y;
    u_old[1] = u_old[0]; // Actualiza retardos anteriores
    u_old[0] = u;
    return y;
}

```

### 3.9 Entorno gráfico y uso de librerías para gráficos

El display ILI9486 fue seleccionado por su capacidad para ofrecer una visualización a color de alta resolución, lo cual es esencial para representar con claridad las señales y espectros que varían en el tiempo. La comunicación entre el ESP32 y el display se facilita mediante el uso de la librería de código abierto LovyanGFX, que proporciona un conjunto robusto de herramientas para el manejo de gráficos y texto en diversos modelos de displays. LovyanGFX destaca por su eficiencia y flexibilidad, permitiendo una integración sencilla y una personalización extensa de la interfaz gráfica de usuario.

Para la representación gráfica de las señales de tensión y corriente en tiempo real, así como para la visualización de los espectros obtenidos a partir de la FFT, se han empleado librerías especializadas como `tdplot` y `spectrum`. Estas herramientas complementan a LovyanGFX, facilitando la generación de gráficos detallados y escalas de frecuencia adecuadas para el análisis espectral. La combinación de estas librerías permite no solo mostrar las formas de onda en su dominio temporal, sino también desglosar sus componentes frecuenciales, crucial para estudios de calidad de energía y diagnósticos eléctricos.

El desafío de actualizar la información visualizada en el display cada 125 ms fue abordado mediante una optimización cuidadosa del código y la gestión eficiente de los recursos del ESP32. Esto implicó un manejo adecuado de las interrupciones y el uso de algoritmos para el cálculo de la FFT y la renderización de los gráficos. La elección de este intervalo de actualización proporciona un equilibrio entre una visualización fluida de las señales y la carga computacional, asegurando que el sistema pueda operar de manera continua.

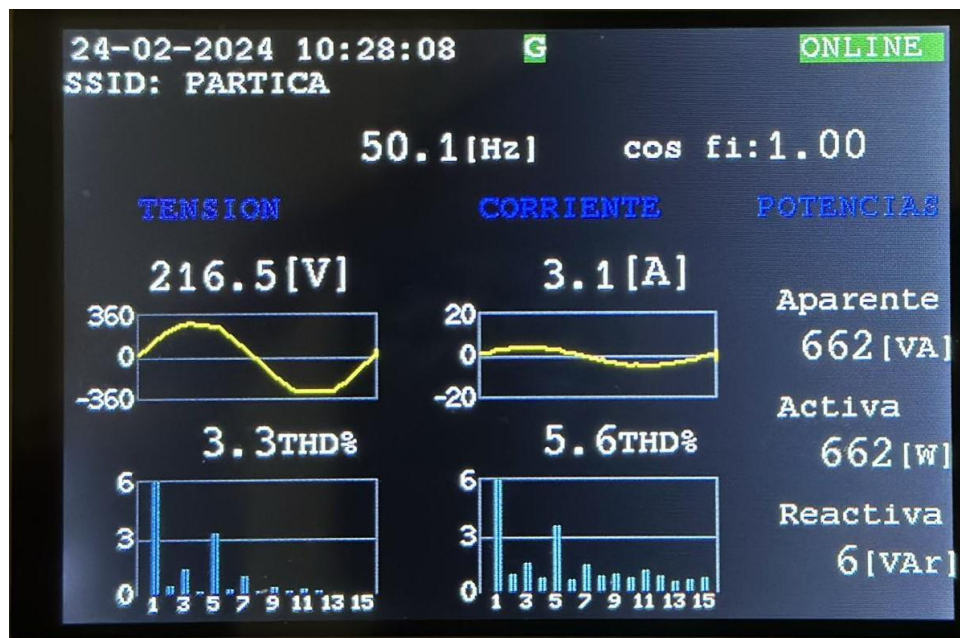


Figura 27 Display con explicación de contenido

La interfaz gráfica desplegada en la pantalla está estructurada para ofrecer una monitorización en tiempo real de parámetros eléctricos críticos. En la parte superior, la pantalla presenta la fecha y hora actual, proporcionando un sello temporal esencial para el registro de datos. Adicionalmente, esta sección superior muestra el estado de la conexión a internet, de vital

importancia para la funcionalidad de transmisión de datos remota, indicado por el ícono "ONLINE", y el SSID de la red Wi-Fi a la que está conectado el dispositivo, permitiendo la confirmación rápida de la conectividad de red.

Centrado en la parte superior de la pantalla, el usuario es informado acerca de las transmisiones de datos hacia el servidor de Google Spreadsheets, lo que refleja la integración del sistema con plataformas de almacenamiento en la nube y garantiza la accesibilidad remota a los datos recolectados.

Inmediatamente debajo del encabezado, se proporcionan las mediciones en tiempo real de la frecuencia de la red eléctrica y el factor de potencia, parámetros importantes para la evaluación del rendimiento del sistema de energía.

La pantalla está diseñada para dividir visualmente la información de tensión y corriente. Se ofrecen las mediciones instantáneas de tensión y corriente en la parte central izquierda y central, respectivamente, acompañadas de sus correspondientes formas de onda sinusoidales, que permiten al usuario apreciar visualmente la calidad y estabilidad de la señal eléctrica en el tiempo. Estas representaciones gráficas son de especial importancia en aplicaciones donde la fidelidad de la onda sinusoidal es indicativa de la calidad de energía.

Además, debajo de cada forma de onda, se presentan gráficos de barras que ilustran el contenido armónico y la Distorsión Armónica Total (THD) para tanto la tensión como la corriente, proporcionando un análisis detallado de la calidad de la señal y permitiendo identificar la presencia de armónicos que podrían afectar el rendimiento de los equipos conectados a la red.

Finalmente, en el margen derecho de la pantalla, se exhiben los valores de potencia aparente, activa y reactiva. Estos indicadores son esenciales para comprender la carga total que se impone a la red eléctrica, diferenciando claramente entre el consumo real de energía y la energía que es devuelta o almacenada en el sistema debido a cargas reactivas.

Cada elemento de la interfaz ha sido diseñado con el objetivo de proporcionar claridad y accesibilidad inmediata a la información más relevante, asegurando que los usuarios puedan realizar evaluaciones rápidas y precisas del estado actual del sistema eléctrico.

#### 4. Posteo periódico Google Spreadsheets

A continuación, se muestra la tarea implementada para el envío de los datos desde el dispositivo hacia la hoja de datos, el código es el siguiente.

```

void postGSS(void *pvParameters)
{
    uint32_t ulNotifiedValue;
    char pubMessage[256];

    const int bufferSize = 64;
    // Crear un buffer de caracteres
    char floatBuffer[bufferSize];
    char floatBufferI[bufferSize];

    while (true)
    {
        // espera notificacion
        xTaskNotifyWait(0, 0, &ulNotifiedValue, portMAX_DELAY);

        // Flag up
        postingNow = true;
        snprintf(floatBuffer, bufferSize, "%.2f,%.2f,%.2f,%.2f,%.2f,%.2f", THD_V_PROM, s_v[3], s_v[5], s_v[7], s_v[9], s_v[11]);
        snprintf(floatBufferI, bufferSize, "%.2f,%.2f,%.2f,%.2f,%.2f,%.2f", THD_I_PROM, s_i[3], s_i[5], s_i[7], s_i[9], s_i[11]);
        // Crea mensaje JSON
        StaticJsonDocument<800> doc;
        JsonObject object = doc.to<JsonObject>();

        // Relleno de datos del mensaje JSON
        object["time"] = datebuf;
        object["freq"] = f_PROM;
        object["v_rms"] = V_rms_PROM;
        object["i_rms"] = I_rms_PROM;
        object["grad2"] = grad_PROM;
        object["THD"] = floatBuffer;
        object["THDI"] = floatBufferI;

        serializeJson(doc, pubMessage);

        // Postea a Google Spreadsheet
        http.begin(apiURL);
        httpCode = http.POST(pubMessage);

        // Flag down
        postingNow = false;

        // espera
        delay(100);
    } // fin while
} //fin postGSS

```

La tarea **postGSS** permanece en espera hasta que recibe una notificación, momento en el cual se activa una bandera denominada **postingNow**. Esta bandera sirve como indicador dentro del sistema que señala que la publicación de datos está en proceso, garantizando así que no se realicen otras operaciones que puedan interferir con la transmisión de datos.

Una vez activada, la tarea procede a formatear los valores a enviar en una estructura JSON. Este formato estructurado es esencial para la integración con la API de Google Spreadsheets, permitiendo una transferencia de datos coherente y organizada.

Tras la preparación del mensaje JSON, **postGSS** inicia una sesión HTTP y realiza una solicitud POST con el contenido JSON. El éxito de la operación se verifica a través de un código de respuesta HTTP, tras lo cual la bandera **postingNow** se restablece, indicando la conclusión del proceso de publicación.

Este mecanismo asegura una actualización regular y fiable de los datos en la nube, permitiendo un acceso remoto casi inmediato a las mediciones recientes para análisis y revisión.

Desde el lado del servidor, se muestra el código implementado en Google Apps Script para la recepción y manejo de los datos enviados a la hoja de datos Spreadsheets de Google donde son almacenados los datos.

```

function doPost(e) {
  // Extract POST'ed data
  var params = JSON.parse(e.postData.getDataAsString());
  var time = params.time;
  var freq = params.freq;
  var v_rms = params.v_rms;
  var i_rms = params.i_rms;
  var grad2 = params.grad2;
  var THD = params.THDI;
  var THDI = params.THDI;
  // Obtain spreadsheet and sheet handler
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  var sheet = ss.getActiveSheet();
  // Append data to the last row
  sheet.appendRow([time, freq, v_rms, i_rms, grad2, THD, THDI]);
}

```

Figura 28 Código implementado en AppScript Google

La función **doPost(e)** presentada es un manejador de eventos en Google Apps Script que se activa cuando se realiza una solicitud HTTP POST a la aplicación web publicada correspondiente. Este script se ejecuta en el contexto de una hoja de cálculo de Google (Google Spreadsheet), y aquí se detallan sus funcionalidades:

1. **Extracción de Datos:** La función comienza analizando los datos recibidos en el cuerpo de la solicitud POST (**e.postData.getDataAsString()**) y los convierte de una cadena JSON a un objeto JavaScript mediante **JSON.parse**.
2. **Variables de Datos:** Extrae los datos individuales del objeto JSON, asignando cada valor relevante a una variable local (**time**, **freq**, **v\_rms**, **i\_rms**, **grad2**, **THD**, **THDI**). Estas variables representan respectivamente el tiempo, la frecuencia de la señal, el valor RMS (Root Mean Square) de la tensión y la corriente, el ángulo de fase de la corriente respecto a la tensión (grados cuadráticos), y la Distorsión Armónica Total tanto de la tensión como de la corriente.
3. **Manejador de Hoja de Cálculo:** Obtiene la hoja de cálculo activa y la hoja actual usando **SpreadsheetApp.getActiveSpreadsheet()** y **getActiveSheet()**. Esto permite al script operar en la hoja de cálculo desde la que se ejecuta.
4. **Registro de Datos:** Finalmente, agrega una nueva fila al final de la hoja actual con los datos recibidos, utilizando el método **appendRow**. Cada elemento del array corresponde a una celda en la nueva fila de la hoja de cálculo.

Esta función permite automatizar la recopilación de datos en una hoja de cálculo de Google, proporcionando un medio eficiente para almacenar y analizar datos recibidos desde el dispositivo que envía periódicamente sus mediciones a través de una solicitud POST.

Los datos son actualizados cada 10 segundos, lo que nos permite tomar una muestra a lo largo de un tiempo lo suficientemente significativa para analizar el comportamiento en cualquier momento que se deseé. La hoja de cálculo fue pensada para facilitar el análisis de variables y poder obtener las gráficas en una ventana de tiempo como así también una lectura de los picos máximos, mínimos y el aproximado de la energía consumida en dicha ventana.

Desde el lado de Google Spreadsheets, se observa lo siguiente.

Fecha	hora	freq	v_rms	THD_V	Lyms	THD_I	Pot (P)	Pot (Q)	Pot (S)	cos FI	i3	i4	i7	i8	i11	i3	i5	i7	i9	i11	
30-01-2024	15:30:00	min	50.00	202.16	4.72	7.57	8.12	1483.19	184.65	1773.43	-0.49	0.09	0.18	0.20	0.08	0.08	0.28	4.25	0.15	0.01	0.06
		max	60.27	209.59	5.52	12.94	13.10	2623.51	601.72	2643.95	0.96	4.43	3.80	3.70	2.95	2.16	6.79	5.84	1.73	0.80	1.03
30-01-2024	16:00:00	Tempo																			
Fecha y hora del último dato	24-02-2024 19:46:29	datos	50.09	263.41	5.03	35.28	nan	9288.54	50.80	9288.40	0.31	17.34	2.49	1.89	1.57	0.78	0.00	0.00	0.00	0.00	0.00
Energía Consumida en dicho tiempo [kWh]	1.0509314																				

Figura 29 Captura de pantalla de Google Spreadsheets

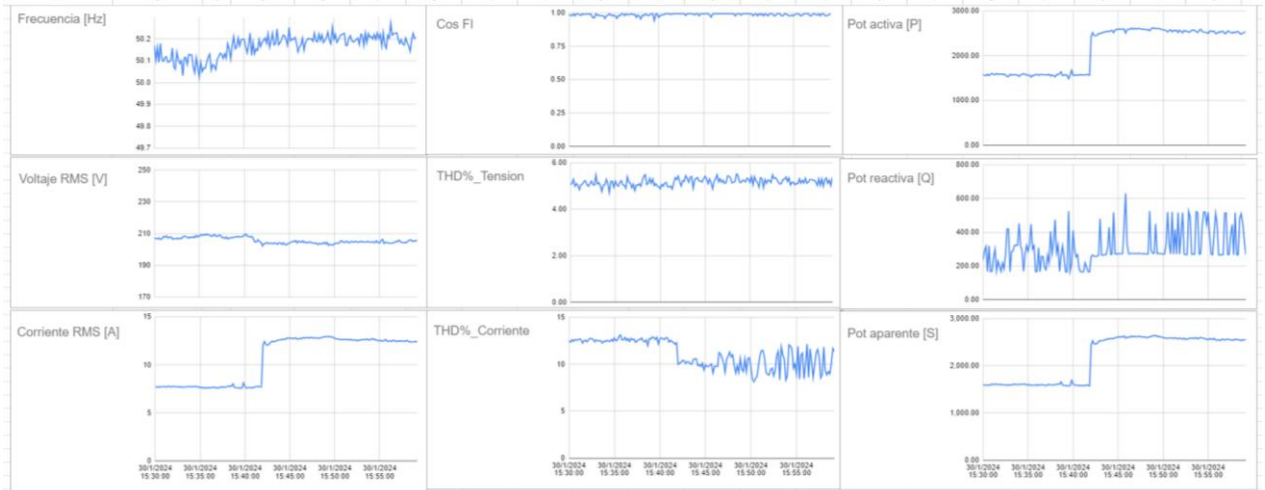


Figura 30 Gráficos históricos de hoja de datos Google Spreadsheets

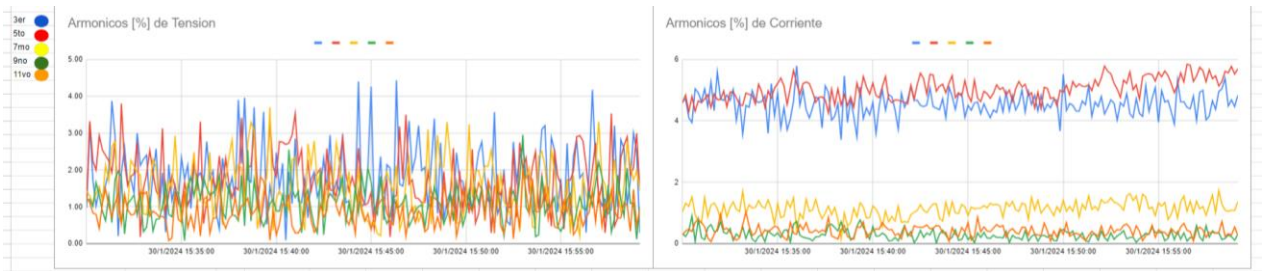


Figura 31 Captura contenido armónico hoja de datos

Con el siguiente código QR, que se encuentra impreso sobre la carcasa del dispositivo se accede a la Hoja de datos de Google Spreadsheets.

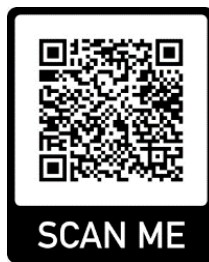


Figura 32 Código QR para acceder a Hoja de Datos Google Spreadsheets

## 5. Redes de Datos. Protocolos de Comunicación ESP32

En el proyecto, el ESP32 juega un papel fundamental como controlador de comunicaciones aprovechando su capacidad nativa de conectividad WiFi y la robustez del protocolo TCP/IP para la transferencia de datos. Este protocolo es esencial para el desarrollo del proyecto ya que

proporciona un método confiable y estandarizado para el envío de paquetes de datos a través de redes complejas y es ampliamente aceptado, facilitando la interoperabilidad con diversas plataformas y servicios en la nube.

## 5.1 Comunicación WiFi

La comunicación WiFi del ESP32 se maneja mediante el estándar IEEE 802.11, que permite la conectividad inalámbrica a la red local y, por extensión, a Internet. El proceso de conexión a la red WiFi comienza con la autenticación y la asociación con el punto de acceso, utilizando el SSID y las credenciales de seguridad pertinentes. Una vez establecida la conexión, el ESP32 puede comunicarse con otros dispositivos en la red local o con servidores en Internet.

El ESP32 gestiona activamente la conexión WiFi, reconectando en caso de pérdida de señal y manteniendo una conexión estable incluso en condiciones de señal fluctuante, lo cual es crucial para la confiabilidad del sistema.

Para el proyecto, la conectividad WiFi no solo es una vía de comunicación, sino que también es una parte integral del sistema de adquisición de datos. Al utilizar la infraestructura WiFi existente, el dispositivo puede operar sin la necesidad de cableado de red adicional, lo que facilita su instalación y reducción de costos. La habilidad para transmitir datos sin ataduras físicas también proporciona una gran flexibilidad en la ubicación del dispositivo y en la escalabilidad del sistema.

## 5.2 Protocolo TCP/IP

El protocolo TCP/IP se divide en varias capas, cada una encargada de diferentes aspectos de la comunicación en la red:

1. **Capa de Enlace:** En el caso del ESP32, la capa de enlace se maneja a través de su hardware de WiFi, que realiza la comunicación con el punto de acceso local y gestiona el control de acceso al medio.
2. **Capa de Internet:** Aquí es donde el protocolo IP asigna direcciones a los dispositivos y dirige los paquetes de datos a su destino correcto a través de la red.
3. **Capa de Transporte:** El protocolo TCP opera en esta capa, estableciendo una conexión segura y confiable entre el ESP32 y el servidor remoto. El TCP garantiza que los paquetes de datos lleguen en orden y sin errores, retransmitiendo cualquier dato que se pierda o se corrompa en el camino.
4. **Capa de Aplicación:** En esta capa se construyen los mensajes de datos que se enviarán, a menudo en formato JSON para facilidad de interpretación y manejo en aplicaciones web.

En nuestro proyecto, el ESP32 utiliza el stack TCP/IP para enviar datos de mediciones eléctricas a un servidor remoto, específicamente a Google Spreadsheets para su registro y análisis. Para lograr esto, se llevan a cabo los siguientes pasos:

1. **Conexión:** El ESP32 se conecta a la red local y establece una conexión TCP con el servidor de Google utilizando su stack WiFi integrado.



2. **Formato de Datos:** Se recopilan mediciones como la frecuencia, voltaje RMS, corriente RMS, y otros parámetros eléctricos, y se formatean en un objeto JSON.
3. **Publicación de Datos:** Mediante una solicitud HTTP POST, el ESP32 envía el objeto JSON a través de la conexión TCP. La solicitud HTTP es un estándar de la capa de aplicación que utiliza el protocolo TCP para asegurar la entrega.
4. **Manejo de Estado:** La tarea de publicación utiliza una bandera, **postingNow**, que se activa durante la transmisión de datos para indicar que la publicación está en curso. Esto es vital para manejar el estado del sistema y evitar la sobreposición de tareas de publicación.

### 5.3 Administrador de conexión inalámbrica

El ESP32 opera en dos modos principales: como un **punto de acceso (AP)** y como una **estación (STA)**. Este dualismo permite que el dispositivo se comporte tanto como un emisor de señal WiFi para que otros dispositivos se conecten directamente a él, como también pueda conectarse a una red WiFi existente.

1. **Modo Punto de Acceso:** Al encenderse, el ESP32 se configura en modo AP, permitiendo que otros dispositivos se conecten a él como si fuera un router WiFi. Esto es particularmente útil cuando el dispositivo se está configurando por primera vez o cuando no hay redes conocidas disponibles.
2. **Conexión al Portal:** Los usuarios se conectan a la red WiFi generada por el ESP32 y navegan a una dirección IP predefinida, típicamente **192.168.4.1**, que sirve como portal de configuración.
3. **Configuración de la Red:** A través de la interfaz del portal, se ingresan los detalles de la red WiFi a la que se conectará el sensor, como el SSID y la contraseña. Estos datos se almacenan en la memoria no volátil del ESP32, lo que permite que el dispositivo se reconecte automáticamente en el futuro sin necesidad de configuración adicional.
4. **Guardado y Reconexión:** Una vez guardada la configuración, el ESP32 se reiniciará y utilizará los parámetros almacenados para conectarse como una estación a la red WiFi seleccionada.

El portal es una solución elegante que simplifica la configuración inicial, ya que no requiere que el usuario instale software adicional. Solo se necesita un dispositivo con capacidad WiFi y un navegador para acceder al portal y configurar la conexión.

A continuación, se enumeran los pasos a seguir para su configuración.

1. En primer lugar, se debe acceder desde el dispositivo desde donde se va a configurar a las redes WiFi disponibles, y seleccionar AutoConnectAP. Es el nombre de red que crea cuando se inicia en modo punto de acceso AP.





Figura 33 SSID de red de WiFi en modo AP

2. Normalmente al conectarse a esa red, automáticamente se abre el navegador en la dirección ip correspondiente, en este caso 193.168.4.1. En caso de no abrirse se accede al navegador de preferencia y se escribe en el buscador la dirección proporcionada. Una vez dentro del portal se accede a un menú como el que se presenta a continuación. En donde se debe presionar Configure WiFi para iniciar la configuración.

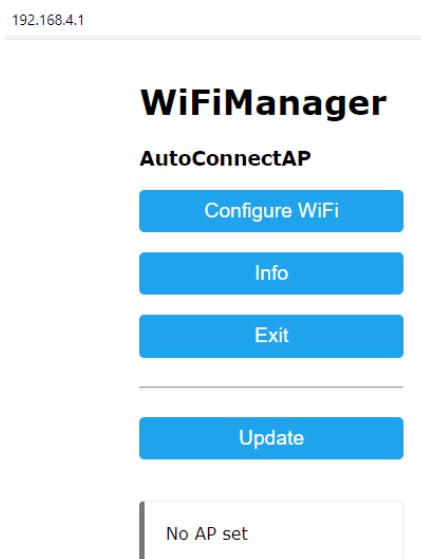


Figura 34 Pantalla principal de portal de configuración de conexión WiFi

3. Una vez dentro del menú, en primer lugar, se debe seleccionar la SSID (nombre de la red) a la que se desea conectar el dispositivo, posteriormente será necesario escribir la clave y presionar el botón Save para que se guarden los datos.

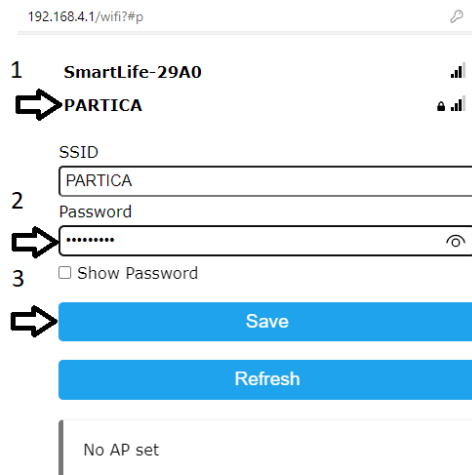


Figura 35 Selección de SSID de red a conectar y escritura de contraseña

4. Luego de los pasos anteriores, se presenta la siguiente pantalla.

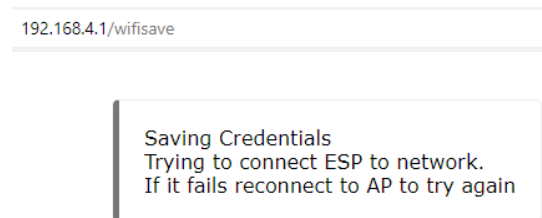


Figura 36 Guardado de credenciales de conexión exitosa

A partir de este momento, cada vez que se inicie el dispositivo va a intentar conectarse a la última red configurada y almacenada en memoria. En caso de no tener éxito en la conexión, nuevamente se iniciará en modo AP esperando una nueva configuración. Puede suceder que, ante la ausencia de acceso a internet, el dispositivo no logre conectarse. Pero después de una espera prudencial, inicia el resto de las tareas, y se conectará automáticamente en cuanto la conexión a internet se restablezca.

## 6. Ensayos realizados

En el marco de este estudio sobre la medición y análisis de la calidad de la energía eléctrica, los ensayos realizados para validar la precisión y fiabilidad del dispositivo desarrollado se llevaron a cabo utilizando una pinza amperométrica UNI-T UT203+ como patrón de referencia. Este instrumento, reconocido por su precisión y confiabilidad en la medición de corrientes en conductores sin necesidad de interrumpir el circuito, permitió establecer un estándar de comparación para las mediciones obtenidas por el dispositivo. La pinza amperométrica facilitó la captura de valores de corriente precisos en tiempo real, los cuales se utilizaron para calibrar y verificar las lecturas del sistema desarrollado, asegurando así su validez y efectividad en la monitorización de la calidad energética.

A continuación, se presentarán las tablas y gráficos con las muestras tomadas para las distintas medidas, detallando frente que equipos patrones se contrastaron las medidas. Luego con dichas mediciones se calculará el error con la siguiente fórmula.

$$\% \text{ de error} = \frac{\text{Valor de referencia} - \text{Valor medido}}{\text{Valor de referencia}} \times 100\%$$

Siendo el valor de referencia los equipos comerciales tomados como patrón y el valor medido por el dispositivo desarrollado.

## 6.1 Pruebas de Medición de Tensión AC

Tabla 5 Comparación de medidas de Tensión UT203+ y equipo, cálculo de error

Tensión [Vrms]		
UT203+	Equipo	error [%]
190	192	1,05
195,4	197	0,82
200	201,5	0,75
205,2	206,2	0,49
210,4	210,8	0,19
215,2	214,9	0,13
220,5	220,4	-0,05
225,3	225,2	-0,04
230,9	230,8	-0,04
234	234,8	0,34
240	240	0,00

Para las mediciones de tensión y corriente, se optó por conectar el dispositivo junto con la pinza en un tablero eléctrico domiciliario para medir consumos reales. Las mismas fueron tomadas sobre la fase que mayores consumos tenía para poder ir agregando cargas para contrastar ambas mediciones.



Figura 37 Pinza de medición voltaje en comparativa con equipo.

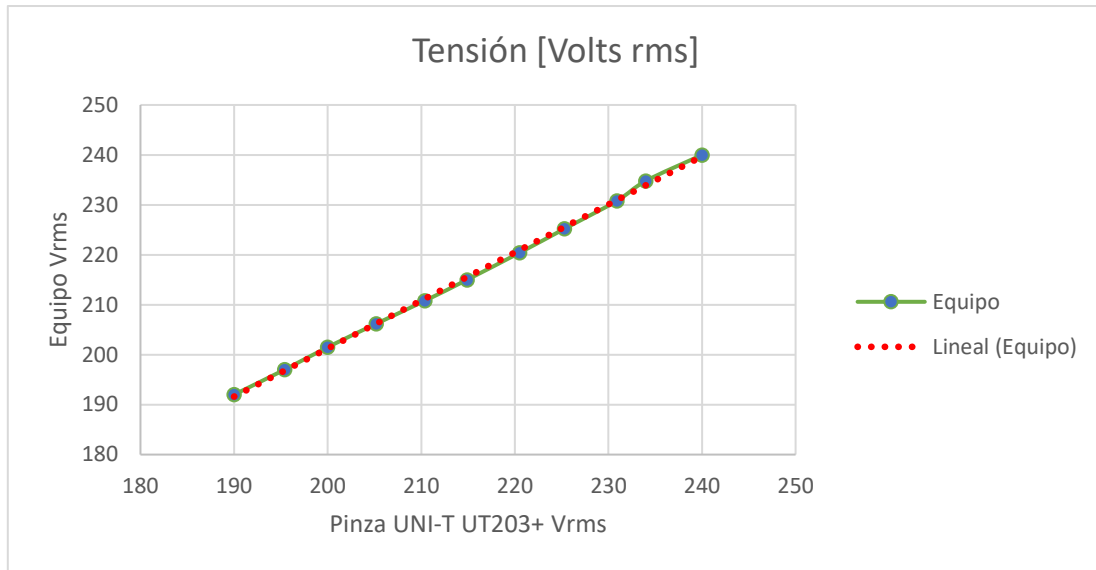


Figura 38 Gráfica de dispersión entre tensión medida en pinza y equipo.

## 6.2 Pruebas de Medición de Corriente AC

Para las mediciones de corrientes se fueron agregando cargas a dicha fase, siendo las más representativas en cuanto a consumo dos unidades de Aire Acondicionado, con potencias de 3,40 kW y 5,30 kW. Adicionalmente, se conectaron a los tomacorrientes de dicha fase, más cargas de tipo resistivo puras, como plancha, secador de pelo y tostadora hasta alcanzar un valor cercano a los 22 A.

Tabla 6 Comparación de medidas Corriente UT203+ y equipo, cálculo de error

Corriente [Irms]		
Tester	Equipo	error [%]
0,12	0,15	-25,0
0,35	0,365	-4,3
0,5	0,60	-20,0
1,95	1,87	-1,5
3,00	2,99	-0,6
4,45	4,50	-1,1
6,33	6,40	-1,1
10,62	11,0	-3,6
13,13	13,4	-2,1
16,96	16,87	0,5
21,9	22,1	-0,9

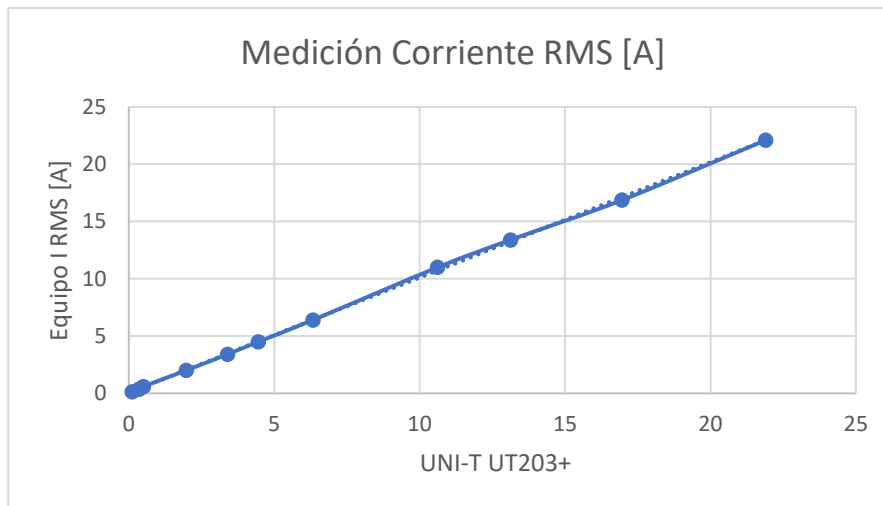


Figura 39 Curva de medición corriente



Figura 40 Medición de corriente comparativa entre pinza y equipo en tablero eléctrico

### 6.3 Pruebas de Medición Armónicos

Para el análisis de armónicos en este estudio, se empleó un osciloscopio Rigol DS2072A dotado de la función matemática de Transformada Rápida de Fourier (FFT). Este enfoque metodológico permitió una evaluación precisa de los componentes armónicos presentes en las señales eléctricas analizadas. La FFT, al descomponer las señales en sus frecuencias constituyentes, facilitó la identificación y cuantificación de los armónicos, ofreciendo así una visión detallada de la calidad de la energía eléctrica. El uso del osciloscopio, un instrumento de alta precisión y capacidad de resolución, aseguró la recopilación de datos fiables y exactos, esenciales para el análisis armónico profundo.

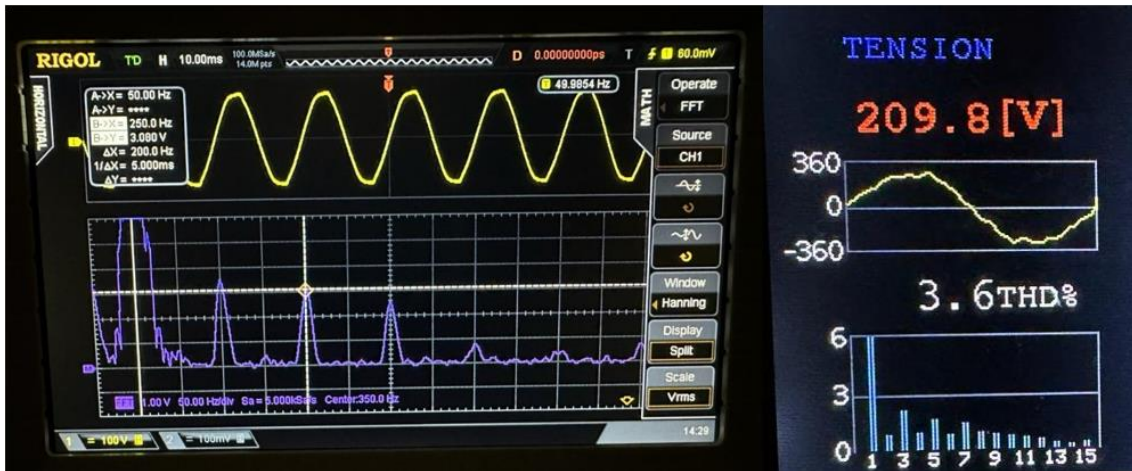


Figura 41 Comparación armónicos equipo y osciloscopio Rigol

En la siguiente imagen se observan las mediciones del osciloscopio realizando la función matemática FFT con ventana Hanning. Vemos que el cursor A se sitúa sobre la fundamental, en 50 Hz y posee una amplitud de 524,0 mV, mientras que el sexto armónico está en 300 Hz y tiene una amplitud de 28,0 mV. Realizando el cálculo de THD, observamos que da en ambos equipos el mismo valor.

$$THD = \frac{\sqrt{\sum_{n=2}^N (V_n)^2}}{V_1} * 100\% = \frac{\sqrt{28^2}}{524} * 100\% = 5,3 \%$$

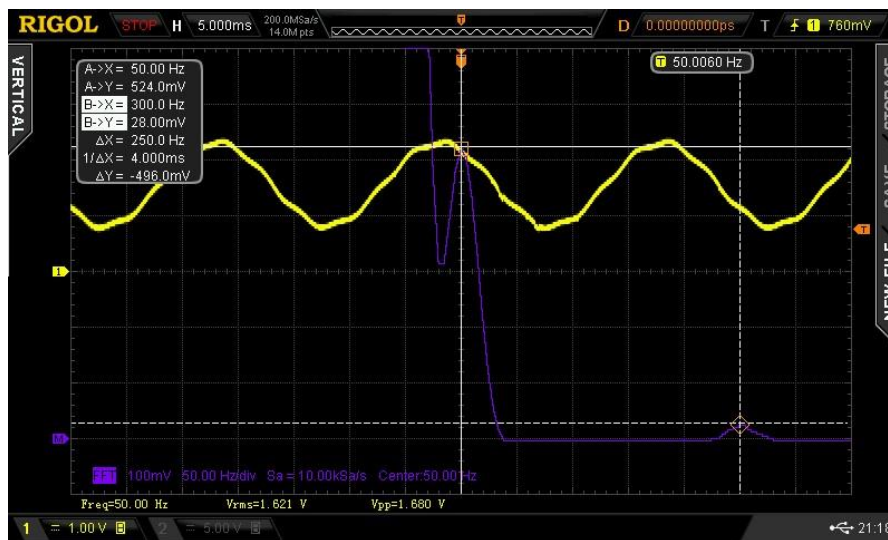


Figura 42 FFT Osciloscopio fundamental y 6to armónico THD 5.0 %

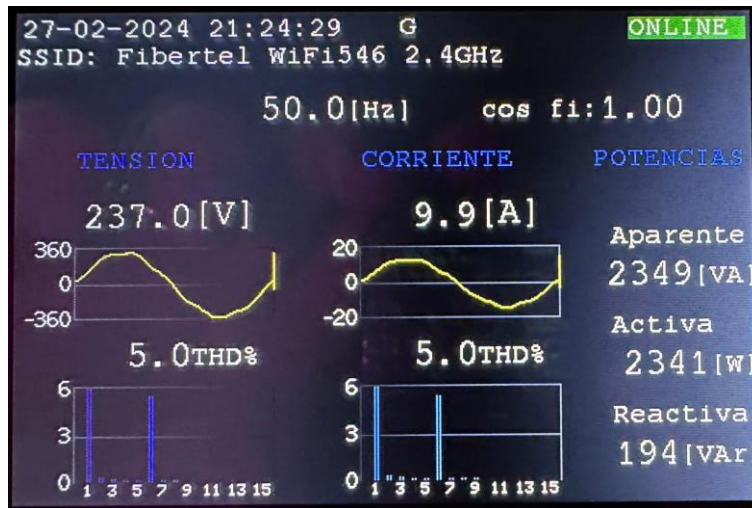
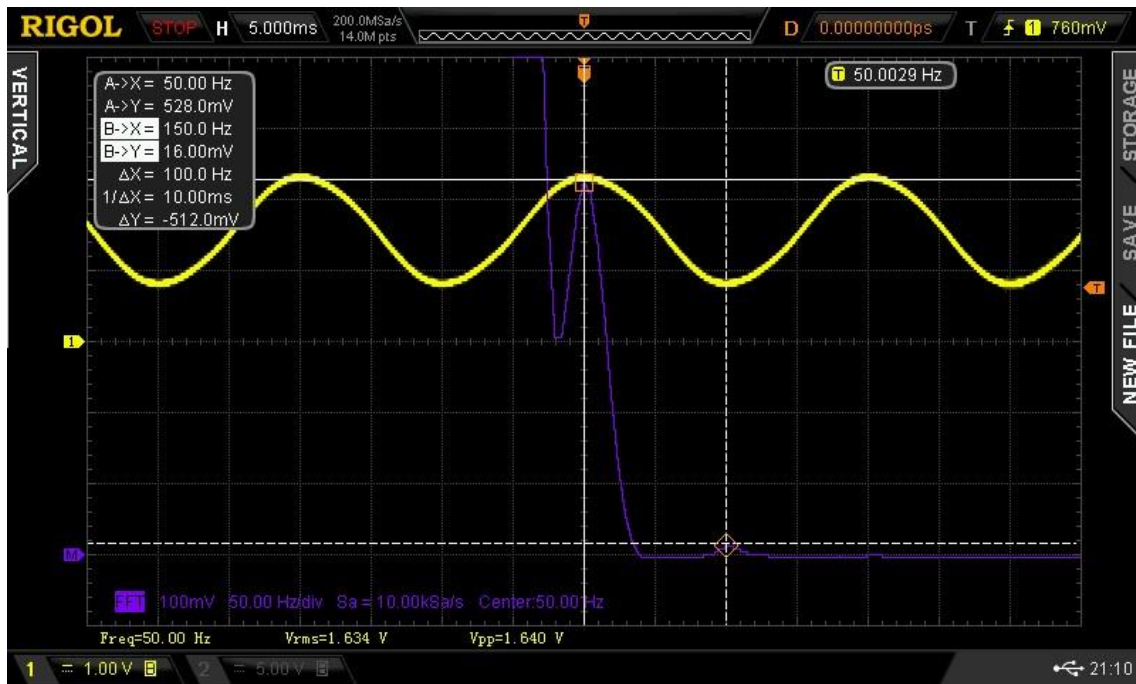


Figura 43 FFT Equipo fundamental y 6to armónico THD 3.0 %

Para el caso del tercer armónico, tenemos la fundamental en 528,0 mV y el armónico 3 en 16,0 mV. Operando de la misma forma calculamos el THD.

$$THD = \frac{\sqrt{\sum_{n=2}^N (V_n)^2}}{V_1} * 100\% = \frac{\sqrt{16^2}}{524} * 100\% = 3,0 \%$$



FFT Osciloscopio fundamental y 3er armónico THD 3.0 %



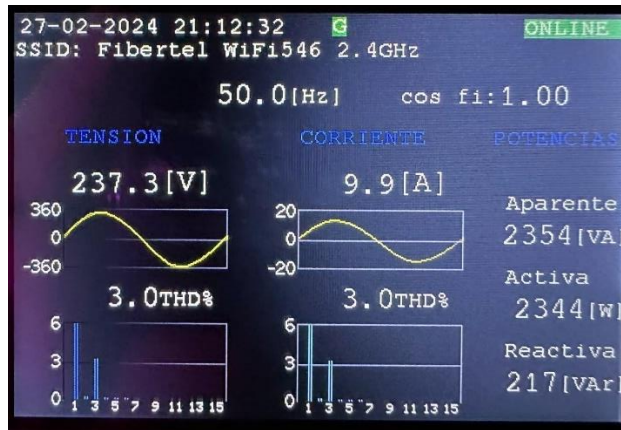


Figura 44 FFT Equipo fundamental y 3er armónico THD 3%

$$THD = \frac{\sqrt{\sum_{n=2}^N (V_n)^2}}{V_1} * 100\% = \frac{\sqrt{16^2 + 12^2}}{524} * 100\% = 3,81 \%$$

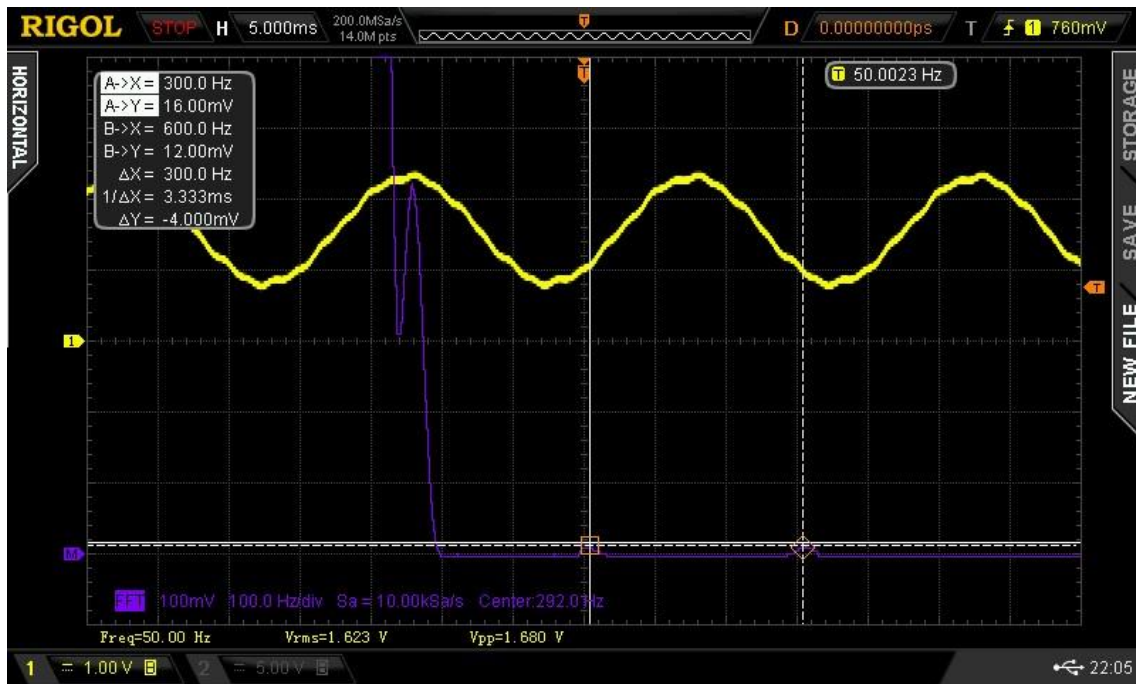


Figura 45 FFT Osciloscopio fundamental y armónico 6 y 12, THD 3,8 %



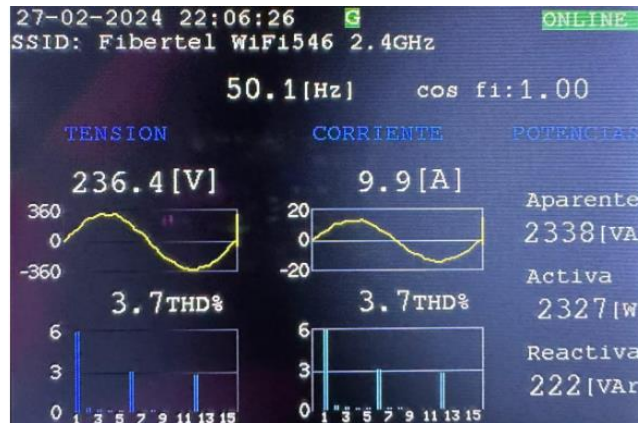


Figura 46 FFT Equipo fundamental y armónico 6 y 12 THD 3,7 %

Tabla 7 Comparación de medidas armónicos Rigol DS2072A y equipo, cálculo de error

Rigol DS2072A	Equipo	error %
1,33	1,3	2,26
2,22	2,2	0,90
3,45	3,4	1,45
3,81	3,7	2,89
4,68	4,6	1,71
5,3	5	5,66
5,34	5,3	0,75
5,97	6	0,50

## 7. Conclusión

El desarrollo de este dispositivo innovador constituye un salto cualitativo en el campo de la ingeniería electrónica, proporcionando una herramienta integral para la medición y análisis de la calidad de la energía eléctrica, incluyendo la capacidad de evaluar con precisión el contenido armónico. Este proyecto surgió de la identificación de una laguna existente en las soluciones de diagnóstico eléctrico disponibles comercialmente, muchas de las cuales carecen de la funcionalidad para medir y analizar armónicos, a pesar de su importancia crítica para la evaluación de la calidad energética con el advenimiento de tantos dispositivos generadores de armónicos a nivel domiciliario.

Utilizando el microcontrolador ESP32, seleccionado por su robusta conectividad a Internet y capacidad de procesamiento, junto con la implementación de algoritmos para el análisis armónico, este dispositivo no solo aborda una necesidad no satisfecha, sino que también allana el camino hacia la creación de sistemas de monitorización energética más avanzados, seguros y confiables.

Los beneficios derivados de la implementación de esta tecnología son extensos y significativos. Contribuye a una mayor eficiencia energética mediante la identificación y corrección de ineficiencias en el consumo de energía. Refuerza la seguridad operacional al reducir el riesgo de fallos y daños en la infraestructura causados por una calidad de energía comprometida. Además, facilita el cumplimiento de las normativas vigentes en materia de calidad energética, un aspecto fundamental en la operación de instalaciones eléctricas contemporáneas.

El análisis del prototipo final revela resultados altamente satisfactorios que superan las expectativas iniciales. El dispositivo no solo ofrece mediciones precisas del consumo energético, sino que también proporciona a los usuarios una comprensión detallada de su consumo eléctrico, permitiéndoles tomar decisiones informadas para optimizar el uso y, por ende, generar ahorros significativos. La capacidad de visualizar los datos en tiempo real, tanto en el dispositivo, a través de su display, como en cualquier dispositivo inteligente conectado a Internet, añade un valor adicional al proyecto.

Si bien el proyecto ha alcanzado y superado sus objetivos iniciales, se identificaron áreas de mejora y expansión futura. Entre ellas, la adaptación del sistema para análisis en redes trifásicas y la incorporación de sensores de mayor amperaje adaptados a entornos industriales, lo cual ampliaría significativamente su aplicabilidad.

Adicionalmente, el desarrollo de un servidor web dedicado específicamente para este dispositivo mejoraría la autonomía y fiabilidad del sistema, liberándolo de la dependencia de servidores de terceros.

En conclusión, este proyecto no solo cumple con los objetivos planteados, sino que establece un nuevo estándar en la monitorización de la calidad de la energía eléctrica, demostrando el impacto transformador de la ingeniería electrónica en el avance hacia sistemas energéticos más eficientes, seguros y sostenibles.

## 8. Anexos

### ASPECTOS NORMATIVOS

La presencia de armónicos en las instalaciones está sujeta a diferentes normativas y reglamentos:

Reglamentación aplicable a los Usuarios de Energía Eléctrica: **Resolución ENRE 099/1997** (Ente Nacional Regulador de la Electricidad, 1997).

Reglamentación aplicable a las Distribuidoras de Energía Eléctrica: **Resolución ENRE 0184/2000** (Ente Nacional Regulador de la Electricidad, 2000).

La generación de armónicos en los equipos tiene su tratamiento en:

Norma **IEC 61000-3-2** para equipos de BT con corriente nominal hasta 16 A.

Norma **IEC 61000-3-4** para equipos de BT con corriente nominal superior a 16 A.

#### **Resolución ENRE 099/1997**

Define el Límite de Emisión Individual de un usuario como aquel nivel de perturbación que puede inyectar en la red en su punto de suministro, y que no podrá ser superado en más de un 5% del tiempo total del período de medición

*Tabla 8 Límites de Emisión Resolución ENRE 099/1997*

Límites de Emisión Individuales de Intensidades Armónicas para usuarios con tarifa T-1, T-2 y T-3.			
Orden de la armónica	Usuario T1	Usuarios T2 y T3 en BT y MT	Usuarios T3 en AT
(n)	Intensidad armónica máxima, en (A)	Intensidad armónica máxima, como el % de la corriente de carga contratada.	Intensidad armónica máxima, como el % de la corriente de carga contratada.
5	2,28	12	6
7	1,54	8,5	5,1
11	0,66	4,3	2,9
13	0,42	3	2,2
17	0,26	2,7	1,8
19	0,24	1,9	1,7
23	0,2	1,6	1,1
25	0,18	1,6	1,1
>25	4,5/n	0,2 + 0,8 * 25/n	0,4

Tabla 9 Límites de Emisión Resolución ENRE 099/1997

Límites de Emisión Individuales de Intensidades de Armónicos para usuarios con tarifa T-1, T-2 y T-3.			
<b>IMPARES MÚLTIPLOS DE 3</b>			
3	4,6	16,6	7,5
9	0,8	2,2	2,2
15	0,3	0,6	0,6
21	0,21	0,4	0,4
>21	4,5 /n	0,3	0,3
<b>PARES</b>			
2	2,16	10	10
4	0,86	2,5	3,8
6	0,6	1	1,5
8	0,46	0,8	0,5
10	0,37	0,8	0,5
12	0,31	0,4	0,5
>12	3,68/n	0,3	0,5
<b>TDTI (en %)</b>		<b>20</b>	<b>12</b>

#### PERÍODO DE MEDICIÓN:

El período de medición para que el control tenga las adecuadas características de fiabilidad y representatividad será de una semana de registros válidos, es decir deberá contener como mínimo 1008 registros de períodos de “10 minutos”.

#### MEDICIÓN DE LA EMISIÓN DE ARMÓNICOS:

En el caso de usuarios con instalaciones que puedan estar emitiendo armónicas por encima del Límite de Emisión Individual correspondiente, la DISTRIBUIDORA realizará las mediciones correspondientes para comprobar si cumple con los Límites de Emisión Individuales fijados en las tablas anteriores. Durante la medición se registrarán en el punto de suministro las tensiones armónicas, las intensidades armónicas y los correspondientes ángulos de desfase entre ellas, en intervalos de 10 minutos, a fin de determinar el flujo de potencia activa de cada armónica controlada de forma tal de comprobar la responsabilidad del usuario.

#### Resolución ENRE 184/2000

Es la base metodológica para el control de las Distribuidoras:

#### PERTURBACIONES:

El Control del nivel de perturbaciones existentes en la red (Flicker y Armónicas).

#### PENALIDADES:

Las sanciones las pagará la DISTRIBUIDORA a los usuarios afectados por la mala calidad de la tensión, aplicando bonificaciones en las facturas inmediatamente posteriores al semestre en que se detectó la anomalía.

Tabla 10 Límites de Emisión Resolución ENRE 184/2000

Tabla 3. Niveles de Referencia para las Armónicas de tensión en BT (U≤1 kV), que no deben ser superados durante más del 5% del periodo de medición.					
Impares no múltiplos de 3		Impares múltiplos de 3		Pares	
Orden de la armónica (n)	Nivel de Referencia de la armónica (en % con respecto a la fundamental)	Orden de la armónica (n)	Nivel de Referencia de la armónica (en % con respecto a la fundamental )	Orden de la armónica (n)	Nivel de Referencia de la armónica (en % con respecto a la fundamental)
5	6	3	5	2	2
7	5	9	1,5	4	1
11	3,5	15	0,3	6	0,5
13	3	21	0,2	8	0,5
17	2	>21	0,2	10	0,5
19	1,5			12	0,2
23	1,5			>12	0,2
25	1,5				
>25	1,5				
<b>Tasa de Distorsión Total: TDT 8 %</b>					

## RECOMENDACIONES SEGÚN IEC

Niveles de Compatibilidad para las tensiones armónicas individuales en una red pública de BT según IEC 61000 -2 -2 (Comisión Electrotécnica Internacional, 2008).

Tabla 11 Niveles de Compatibilidad tensiones armónicas IEC 61000 -2 -2

Armónicos impares no múltiplos de 3		Armónicos impares múltiplos de 3		Armónicos pares	
Rango armónico n	Tensión armónico %	Rango armónico n	Tensión armónico %	Rango armónico n	Tensión armónico %
5	6	3	5	2	2
7	5	9	1,5	4	1
11	3,5	15	0,3	6	0,5
13	3	21	0,2	8	0,5
17	2	>21	0,2	10	0,5
19	1,5			12	0,2
23	1,5			>12	0,2
>25	$0,2 + 0,5 \times 25 / n$				

## 9. Referencias Bibliográficas

### 9.1 Sitios Web

AC-DC power module 5V 700mA. (s.f.). [https://wiki.iteadstudio.com/AC-DC Power Module 5V 700mA](https://wiki.iteadstudio.com/AC-DC-Power-Module-5V-700mA)

ArduinoIDE. (s.f.). Software. <https://www.arduino.cc/en/software>

ESP32 Pinout reference. (s.f.). Pinout-reference. <https://lastminuteengineers.com/esp32-pinout-reference/>

FreeRTOS. (s.f.). Real Time Operating System for embedded systems. <https://www.freertos.org/>

ILI9486. (s.f.). Datasheet. [https://www.waveshare.com/w/upload/7/78/ILI9486 Datasheet.pdf](https://www.waveshare.com/w/upload/7/78/ILI9486-Datasheet.pdf)

Si8920. (s.f.). Datasheet <https://www.skyworksinc.com/-/media/Skyworks/SL/documents/public/data-sheets/si8920-datasheet.pdf>

### 9.2 Normas técnicas

Comisión Electrotécnica Internacional. (2008). Compatibilidad electromagnética (EMC). Parte 2-2: Entorno electromagnético - Límites de compatibilidad para armónicos de corriente (corriente de entrada) producidos por equipos de alumbrado de baja tensión y equipos electrónicos de potencia. (Norma IEC 61000-2-2, 3.ª ed.). Ginebra, Suiza: IEC.

Comisión Electrotécnica Internacional. (2018). Compatibilidad electromagnética (EMC). Parte 3-2: Límites de emisión para armónicos de corriente (corriente de entrada) producidos por equipos de alumbrado de baja tensión y equipos electrónicos de potencia. (Norma IEC 61000-3-2, 5.ª ed.). Ginebra, Suiza: IEC.

Comité Europeo de Normalización. (2010). Compatibilidad electromagnética. Requisitos de emisión de los equipos eléctricos y electrónicos de baja tensión. (Norma EN 50160, 2.ª ed.). Bruselas, Bélgica: CEN.

Ente Nacional Regulador de la Electricidad. (1997). Resolución ENRE N° 099/1997. Normas de calidad de la energía eléctrica. Buenos Aires, Argentina: ENRE.

Ente Nacional Regulador de la Electricidad. (2000). Resolución ENRE N° 184/2000. Normas de calidad de la energía eléctrica. Buenos Aires, Argentina: ENRE.

Instituto de Ingenieros Eléctricos y Electrónicos. (2014). IEEE Recommended Practice and Requirements for Harmonic Control in Electric Power Systems. (Norma IEEE 519, 1.ª revisión). Piscataway, NJ, Estados Unidos: IEEE.