

# Universidad Tecnológica Nacional

Proyecto Final

---

## Filtro Armónico Activo

---

Autores:

- Di Lazzaro, Ángel Gabriel
- Dolaz, Leandro Agustín
- Ubiedo, Lautaro Ezequiel

Director:

- Ing. Agustín Chort

*Proyecto final presentado para cumplimentar los requisitos académicos para acceder al título de Ingeniero Electrónico*

*en la*

**Facultad Regional Paraná**

21 de abril de 2025

## **Declaración de autoría:**

Nosotros declaramos que el Proyecto Final "Filtro Armónico Activo" y el trabajo realizado son propios. Declaramos:

Este trabajo fue realizado en su totalidad, o principalmente, para acceder al título de grado de Ingeniero Electrónico, en la Universidad Tecnológica Nacional, Regional Paraná. Se establece claramente que el desarrollo realizado y el informe que lo acompaña no han sido previamente utilizados para acceder a otro título de grado o pre-grado. Siempre que se ha utilizado trabajo de otros autores, el mismo ha sido correctamente citado. El resto del trabajo es de autoría propia. Se ha indicado y agradecido correctamente a todos aquellos que han colaborado con el presente trabajo. Cuando el trabajo forma parte de un trabajo de mayores dimensiones donde han participado otras personas, se ha indicado claramente el alcance del trabajo realizado.

Firmas:

Fecha:

## Agradecimientos:

A nuestro director de proyecto, Ing. Agustín Chort, por brindarnos todas las herramientas necesarias y poner siempre a disposición el Laboratorio de Potencia para nuestras pruebas a lo largo del proyecto.

A nuestras familias, por su apoyo y soporte a lo largo de todos estos años brindándonos todos los recursos necesarios para poder dedicarnos de manera exclusiva a nuestro desarrollo profesional.

A nuestros amigos, por el acompañamiento y apoyo a lo largo de toda la carrera. Así como también su constante predisposición a colaborar en los aspectos que lo requeríamos.

A nuestros docentes, en especial a los ingenieros Fabio Vincitorio y Gustavo Yarce por brindarnos recomendaciones y sugerencias de gran valor durante el desarrollo del proyecto.

Di Lazzaro Ángel Gabriel  
Dolaz Leandro Agustín  
Ubiedo Lautaro Ezequiel

Universidad Tecnológica Nacional

## *Abstract*

Facultad Regional Paraná

Ingeniero en Electrónica

### **Filtro Armónico Activo**

Di Lazzaro Ángel Gabriel

Dolaz Leandro Agustín

Ubiedo Lautaro Ezequiel

#### **Abstract:**

We developed a prototype of a single-phase shunt active harmonic filter. This works with a visualization application developed in Python. By real-time measuring the load current using the Raspberry Pi Pico development board and the SCT-013 current sensor, we input these data into an adaptive filter implemented on the Tang Nano 9k FPGA to generate, through a full-bridge inverter, the harmonic currents of the load in counterphase, this is the signal that is added to the load current to reduce the Total Harmonic Distortion of current (THDi). The implementation of this prototype, scalable in power and adaptable to three phases, consistently achieved a resulting THDi below 5% when tested with current generated by a dimmer and resistive load with a THDi exceeding 70%, even with real-time variations in the load.

**Keywords:** FPGA, Raspberry Pi Pico, Python, Inverter.

**Resumen:**

Realizamos un prototipo de filtro armónico activo monofásico en derivación. El mismo funciona junto con una aplicación de visualización desarrollada en Python. Midiendo en tiempo real la corriente de carga mediante la placa de desarrollo Raspberry Pi Pico y el sensor de corriente SCT-013, ingresamos estos datos a un filtro adaptativo implementado en la FPGA Tang Nano 9k para generar, mediante un inversor de puente completo, los armónicos de la corriente de carga en contrafase, es esta señal la que se suma a la corriente de carga para disminuir la distorsión total armónica de corriente (THDi). La implementación de este prototipo, escalable en potencia y trasladable a tres fases, logró mantener un THDi resultante por debajo del 5 % de forma constante al usar como prueba una corriente generada por un dimmer y una carga resistiva con un THDi mayor al 70 %, e incluso realizando variaciones de la carga en tiempo real.

**Palabras Clave:** FPGA, Raspberry Pi Pico, Python, Inversor.

# Índice

Capítulo 1: Introducción	1
Capítulo 2: Desarrollo	3
1    Etapa de Sensado: . . . . .	3
1.1    Sensado de Corrientes de Carga y Armónica: . . . . .	6
1.2    Sensado de la Corriente Resultante: . . . . .	20
2    Etapa de Filtrado y Generación de señales PWM . . . . .	25
2.1    Modelado de Algoritmos en Punto Fijo . . . . .	26
2.2    Compensación Armónica: El Filtro Adaptativo . . . . .	27
2.3    Marco teórico . . . . .	27
2.4    Diagrama de Bloques Completo de la FPGA . . . . .	30
2.5    Prefiltro . . . . .	31
2.6    Filtro de Media Móvil . . . . .	31
2.7    Filtro de Media Móvil Recursivo . . . . .	33
2.8    Simulación del Filtro de Media Móvil Recursivo . . . . .	34
2.9    El Filtro FIR y el Algoritmo LMS . . . . .	35
2.10    Generador de PWM . . . . .	38
2.11    Generación del punto muerto . . . . .	40
2.12    Simulación del Algoritmo PWM y Tiempo Muerto . . . . .	41
2.13    Modulación PWM de los Armónicos . . . . .	42
3    Etapa de Generación de Armónicos: . . . . .	43
4    Modelado y Fabricación del Gabinete: . . . . .	55
5    Aplicación de Visualización: . . . . .	58
Capítulo 3: Resultados	61
1    Recursos utilizados en la FPGA . . . . .	61
2    Prestaciones del dispositivo . . . . .	63
Capítulo 4: Análisis de Costos	71
Capítulo 5: Discusión y Conclusión	76
Capítulo 6: Referencias	80

# Lista de Figuras

1	Vista del Chip RP2040 [1]. . . . .	3
2	Comunicación entre FPGA y RP2040. . . . .	4
3	Diagrama de Flujo del Software de la RP2040. . . . .	5
4	Circuito de Medición de Corrientes. . . . .	6
5	Sensores de Corriente SCT-013[2]. . . . .	7
6	Diagrama de Conexión del Sensor SCT-013[2]. . . . .	8
7	Filtro Antialiasing. . . . .	8
8	Respuesta en Frecuencia del Filtro Antialiasing. . . . .	9
9	Simulación del Filtro Antialiasing. . . . .	9
10	Amplificador No Inversor con MCP41010 [3]. . . . .	10
11	Adición de Capacitor [4]. . . . .	11
12	Funcionamiento Completo del Circuito de Sensado de Corriente. . . . .	12
13	Configuración de Pruebas Preliminares. . . . .	13
14	Medición de Referencia. . . . .	14
15	Medición de Prueba Lámpara 300[W]. . . . .	15
16	Medición de Prueba Plancha 1100[W]. . . . .	16
17	Medición de Prueba Lámpara 75[W]. . . . .	17
18	Medición de Cada Bit del Puerto Paralelo. . . . .	18
19	Gráfico de las Muestras del Puerto Paralelo. . . . .	19
20	Circuito de Medición de la Señal Resultante. . . . .	20
21	Señal de Salida del R2R. . . . .	21
22	Circuito de Protección CMOS del CD4013[5]. . . . .	22
23	Esquemático Completo del Circuito de Sensado. . . . .	23
24	PCB Principal de la Etapa de Sensado. . . . .	24
25	PCB del Conversor Analógico Digital. . . . .	24
26	Aritmética de Punto Fijo [6]. . . . .	25
27	Compensación Armónica: El Filtro Adaptativo [7]. . . . .	27
28	Ecuador Adaptativo[8]. . . . .	28
29	Curva de error cuadrático medio[9]. . . . .	28
30	Coeficientes del Filtro FIR. . . . .	29
31	Diagrama en Bloques de la FPGA. . . . .	30
32	Algoritmo en Bloques de la FPGA. . . . .	31
33	Filtro de Media Móvil[10]. . . . .	32
34	Filtrado de la Corriente Armónica. . . . .	32
35	Filtros de media móvil en cascada. . . . .	33
36	Filtro de Media Móvil Recursivo[10]. . . . .	33
37	Simulación del Filtro de Media Móvil Recursivo. . . . .	34
38	Filtros de Media Móvil en cascada. . . . .	34

39	Implementación del Filtro de Media Móvil Recursivo. . . . .	35
40	Respuesta temporal del filtro adaptativo. . . . .	36
41	Respuesta espectral del filtro adaptativo. . . . .	36
42	Simulación del filtro adaptativo en Vivado. . . . .	37
43	Bloque del Filtro Adaptativo. . . . .	37
44	Circuito Interno del Filtro Adaptativo. . . . .	38
45	Comparativa entre una señal moduladora y las portadoras. . . . .	39
46	Generación del Punto Muerto. . . . .	40
47	Simulación del Algoritmo PWM y el Tiempo Muerto. . . . .	41
48	Relación entre la Frecuencia y la Modulación[11]. . . . .	42
49	Circuito de Generación de Armónicos. . . . .	43
50	Tiempo Muerto para las Señales PWM. . . . .	45
51	Señal PWM Filtrada. . . . .	45
52	Salida Senoidal Filtrada del Inversor. . . . .	46
53	Señal de Prueba con Armónicos. . . . .	46
54	Respuesta en Frecuencia del Filtro RLC. . . . .	47
55	Tensión de Salida del Filtro RLC. . . . .	47
56	FFT de la Tensión de Salida del Filtro RLC. . . . .	48
57	Primera Medición de Corriente del Inversor con la App. . . . .	48
58	Medición Final de Corriente del Inversor con la App. . . . .	49
59	PCB de la Etapa de Generación de Armónicos. . . . .	50
60	Problema de Cortocircuito. . . . .	51
61	Circuito de Adaptación de Señales PWM. . . . .	52
62	Tiempos de Propagación 6N137[12] . . . . .	53
63	PCB de los Optoacopladores. . . . .	53
64	Señales de Salida de los Optoacopladores. . . . .	54
65	Circuito Completo de la Etapa de Generación de Armónicos. . . . .	54
66	[V]ista Exterior del Gabinete. . . . .	55
67	Vista Interior del Gabinete. . . . .	56
68	Vista Exterior del Gabinete. . . . .	57
69	Panel Frontal del Gabinete . . . . .	57
70	Aplicación Final Desarrollada. . . . .	59
71	Ventanas Emergentes de Configuración. . . . .	60
72	Tabla de recursos utilizados por bloque implementado. . . . .	61
73	Lógica combinacional utilizada. . . . .	61
74	Lógica secuencial utilizada. . . . .	62
75	Elementos DSP (multiplicadores). . . . .	62
76	Recursos usados contra libres. . . . .	63
77	Medición Realizada con Osciloscopio y Pinza Amperométrica. . . . .	65
78	Medición Realizada del Filtro Activo de Potencia con la App. . . . .	66

79	Prueba de Funcionamiento con Lámpara de 60[W]. . . . .	68
80	Prueba de Funcionamiento con Plancha de 1000[W]. . . . .	69
81	Prueba de Funcionamiento con Plancha de 1000[W] y Más Conducción. . . .	70

## Lista de Tablas

1	Comparación FFT de la Aplicación y del Osciloscopio. . . . .	67
2	Costos Componentes de la Etapa de Sensado. . . . .	71
3	Costos Componentes del Inversor. . . . .	73
4	Costos de Materiales para Fabricar las PCBs. . . . .	74
5	Costos de Materiales para el Gabinete. . . . .	75
6	Comparación entre Filtro Comercial y el Propuesto. . . . .	78

# Lista de Abreviaciones y Símbolos

**A/D** Analógico/Digital. 36

**ADC** Conversor Analógico-Digital (Analog-Digital Converter). 3–5, 10, 11, 13, 18, 20–22, 72, 79

**APF** Filtro Armónico Activo (Active Power Filter). 1, 2

**CMOS** Semiconductor de Óxido Metálico Complementario. 22

**DMA** Acceso Directo a Memoria (Direct Memory Access). 4

**FET** Transistor de Efecto de Campo (Field Effect Transistor). 8

**FFT** Transformada Rápida de Fourier (Fourier Fast Transform). 17, 47–49, 59, 60, 65, 67, 68

**FIR** Respuesta al Impulso Finita. 27–31, 35, 37

**FPGA** Matriz de puerta programable en campo. 1, 3, 4, 13, 18, 20, 21, 25, 30, 31, 43, 52, 53, 61, 68, 69, 78, 79

**GPIO** Entrada/Salida de Uso Genérico (General Purpose Input/Output). 4

**IGBT** Transistor Bipolar de Compuerta Aislada (Isolated Gate Bipolar Transistor). 40, 43, 44, 64, 72–74, 76

**LED** Diodo Emisor de Luz (Light Emitter Diode). 44, 50, 52, 55

**LMS** Mínimo Cuadrado Medio (Least Mean Square). 35

**MAF** Filtro de Media Móvil (Moving Average Filter). 34

**MOSFETS** Transistor de Efecto de Campo de Metal-Óxido-Semiconductor (Metal-Oxide-Semiconductor Field Effect Transistor). 76

**PCB** Placa de Circuito Impreso (Printed Circuit Board). 74

**PIO** Entrada/Salida Programable (Programmable Input/Output). 4

**PLA** Poli Ácido Láctico. 57, 75

**PWM** Modulación por Ancho de Pulsos. 1, 27, 30, 40, 42, 43, 45–47, 51–53, 74, 76

**RMS** Raíz de la Media del Cuadrado (Root Mean Square). 14–16, 49, 58, 60, 69

**SPWM** Modulación por Ancho de Pulso Sinusoidal (Sinusoidal Pulse Width Modulation). 31

**THD** Distorsión Armónica Total. 2, 49, 65, 68–70, 79


**THDi** Distorsión Armónica Total de Corriente. 20, 58, 59, 65, 69, 70, 76, 78, 79

**UTN** Universidad Tecnológica Nacional. 57

**VSI** Inversor de Voltaje Conmutado. 74

## **Dedicado a:**

Nuestras familias, por su apoyo incondicional a lo largo de todos estos años.

 <b>UTN Regional Paraná</b>	<b>Filtro Armónico Activo</b>
<b>Ingeniería en Electrónica</b>	<b>1.Introducción</b>

## Capítulo 1: Introducción

Las cargas no lineales, como equipos electrónicos y motores con control de velocidad, consumen corriente de manera no sinusoidal que distorsiona la forma de onda de la corriente. Esta distorsión genera armónicos, que son componentes de frecuencia múltiple de la frecuencia fundamental, los cuales pueden causar varios problemas en el sistema eléctrico, como el sobrecalentamiento de transformadores y motores, la degradación del aislamiento de los cables, la reducción de la eficiencia energética y el mal funcionamiento de equipos sensibles[13]. Para solucionar estos problemas asociados con la distorsión armónica hemos desarrollado un prototipo del Filtro Armónico Activo (APF) de forma de establecer un método eficiente para eliminar la mayor cantidad posible de armónicos de la red en tiempo real[14].

El desarrollo de nuestro APF fue llevado a cabo en etapas, donde primero realizamos la etapa de medición de las corrientes, luego la etapa de generación de corriente y por último el software de visualización que permite ver las corrientes de interés y las mediciones necesarias para que el usuario pueda corroborar el correcto funcionamiento del dispositivo. Para la medición de las corrientes usamos los sensores SCT-013, debido a que podemos colocarlos en la instalación eléctrica sin tener que modificarla, en conjunto con la placa de desarrollo Raspberry Pi Pico que nos permite tomar la información a la velocidad necesaria y transmitirla tanto a la FPGA como a la computadora al mismo tiempo. Para la implementación del algoritmo del APF empleamos la FPGA Tang Nano 9K debido a que tiene la capacidad computacional necesaria y un pequeño tamaño. Es en la FPGA donde se analiza la corriente de carga y se extrae de la misma la información de los armónicos en ella presentes, para generar una señal con los armónicos de la corriente de carga, hasta 1[KHz], con su misma amplitud pero en contrafase con la idea de que al sumar esta señal generada con la corriente de carga original los armónicos sean eliminados de la red eléctrica. Para poder generar esta señal armónica primero la modulamos empleando el método de ancho de pulso (PWM) y luego usamos estas señales PWM para controlar un inversor monofásico de puente completo el cual generará esta corriente armónica alterna. Al tratarse de un prototipo nuestra implementación no realiza la inyección de corriente pero si realizamos la suma algebraica de las señales, para poder mostrar en la aplicación de computadora, desarrollada en Python, la corriente de carga, la corriente de armónicos generada y la corriente que resulta de aplicar este filtro.



Nuestro objetivo principal fue lograr que la señal obtenida al aplicar nuestro APF tenga en todo momento una distorsión total armónica (THD) menor al 5 % y que aunque se varíe la carga en tiempo real este resultado se mantenga. Al realizar pruebas empleando un dimmer controlando cargas resistivas pudimos generar corrientes con una alta carga de armónicos (con un THD mayor al 70 %) y nuestro APF pudo controlar estos armónicos y mantuvo en todo momento un THD en la señal resultante menor al 5 %. Además, hemos logrado que nuestro diseño tanto de software como de hardware sea compatible con aplicaciones reales en ambientes industrial ya sean monofásicos o trifásicos, teniendo también la posibilidad de mejorar su capacidad de generación de potencia armónica la cual está actualmente con esta implementación en torno a los 500[W].



## Capítulo 2: Desarrollo

### 2.1. Etapa de Sensado:

Para el funcionamiento del filtro necesitamos medir la corriente de la carga, cuyos armónicos queremos eliminar de la red, la corriente que genera el inversor, para corroborar su correcto funcionamiento, y la corriente resultante que se obtiene al aplicar el filtrado. La medición de estas corrientes se realizó mediante los sensores SCT-013-010 de la marca YHDC (más adelante hablaremos de todas sus especificaciones técnicas) a los cuales se les aplicó un filtrado para limitar el espectro de frecuencias y luego una amplificación para así aprovechar al máximo la resolución del ADC del microcontrolador RP2040. Por otra parte, la señal de la corriente resultante es generada por la FPGA y enviada también al microcontrolador.

El microcontrolador cumple con tres tareas de forma simultánea, en primer lugar lee los datos provenientes de ambos sensores de corriente y de la FPGA, también envía los datos de la corriente de carga a la FPGA a través de un puerto paralelo de 8 bits, y a su vez envía los datos de las tres señales de interés por puerto serie a la computadora para poder visualizarlos mediante nuestra aplicación. Para ello aprovechamos que el microcontrolador RP2040 está conformado por cuatro núcleos que pueden emplearse a discreción, dos núcleos del procesador Cortex M0+ (Proc0 y Proc1) y dos núcleos o bloques PIO (programmable input/output) denominados PIO0 y PIO1 [1], tal como se ve en la Fig. 1.

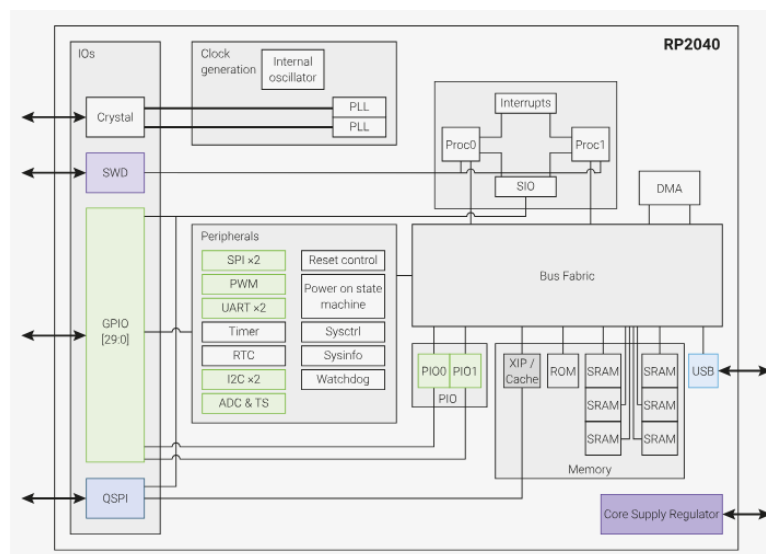


Fig. 1: Vista del Chip RP2040 [1].



Los núcleos del procesador pueden programarse de forma individual y ejecutarán el código que se les asigne de forma simultánea. Estos núcleos se programan en C y funcionan a la velocidad de clock de la placa Raspberry Pi Pico (130[MHz]). Por otro lado, podemos entender a los bloques PIO como máquinas de estado que solo pueden realizar operaciones sobre los pines de salida y/o entrada de forma más básica. Los PIO se deben programar en Assembler y si bien emplean el mismo clock estos bloques son interfaces de hardware muy versátiles ya que cuentan con conexiones dedicadas al bus principal, al controlador de interrupciones y a los GPIO lo que hace que la velocidad de ejecución de instrucciones de los PIO sea mayor a la de los núcleos del procesador principal[1].

Para maximizar la eficiencia de nuestro software de sensado, hicimos uso de la posibilidad que nos ofrece la RP2040 de realizar la lectura de los tres canales del ADC de forma independiente a una tasa de muestreo total de 500[Ksps] (quedando 166[Ksps] por canal) y emplear el acceso directo en memoria (DMA) para almacenar los datos. De esta forma el proceso de lectura se realiza de forma independiente, por lo que tenemos un núcleo del procesador enviando los datos por puerto serie a la computadora, al mismo tiempo que otro núcleo se encarga de enviar cada dato de la corriente de carga al registro de salida del PIO0, el cual se comunica directamente con la FPGA de forma sincrónica para enviarle el dato de 8 bits por puerto paralelo usando 8 GPIO para ello, al ritmo que demande la FPGA.

La comunicación entre la FPGA y la RP2040, como mencionamos previamente, es sincrónica y con transmisión de datos de forma paralela (Fig. 2). La RP2040 pone un 1 en un GPIO, el cual la FPGA considerará como la habilitación para comenzar la comunicación, iniciando así la señal de clock. A partir de ese momento, en cada flanco ascendente de la señal de clock se enviará un dato por el puerto paralelo de 8 bits.

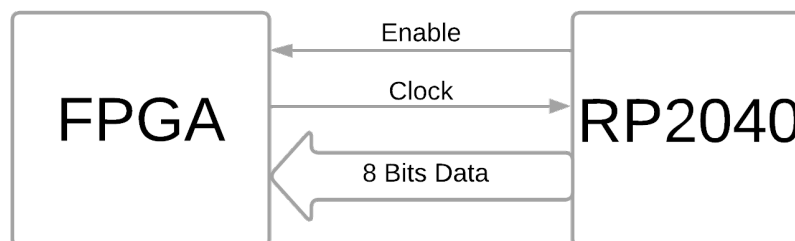


Fig. 2: Comunicación entre FPGA y RP2040.



Debemos mencionar también que, además del envío de datos hacia la computadora, el núcleo Proc0 inicializa y configura todas las interrupciones, periféricos y pines de entrada y salida. Siendo además este núcleo el encargado del control automático de ganancia, el cual asegura que sin importar la corriente a medir, siempre se emplee la totalidad de resolución del ADC.

De modo de resumen, antes de pasar a explicar la electrónica detrás de la etapa de sensado, en la Fig. 3 podrá usted observar un diagrama de flujo del software de la RP2040.

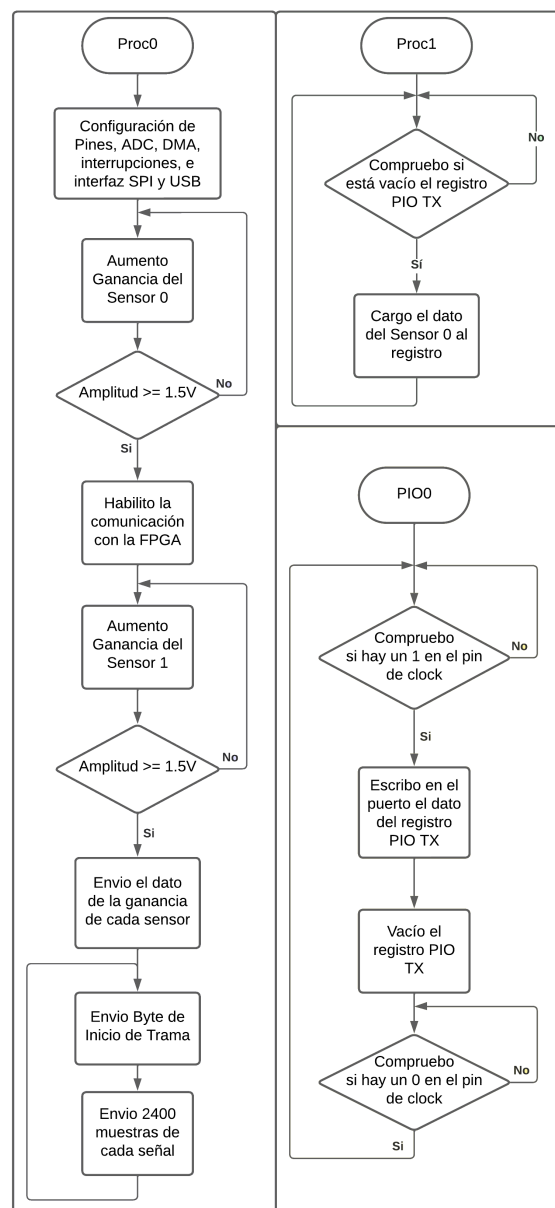


Fig. 3: Diagrama de Flujo del Software de la RP2040.



### 2.1.1. Sensado de Corrientes de Carga y Armónica:

En nuestra aplicación necesitamos medir la corriente de la red con el consumo de la carga y la corriente de salida del inversor sobre la carga de prueba por lo que contamos con dos etapas de sensado de corriente, siendo ambas exactamente iguales (como se puede observar en la Fig. 4) por lo que haremos una explicación detallada de una sola etapa.

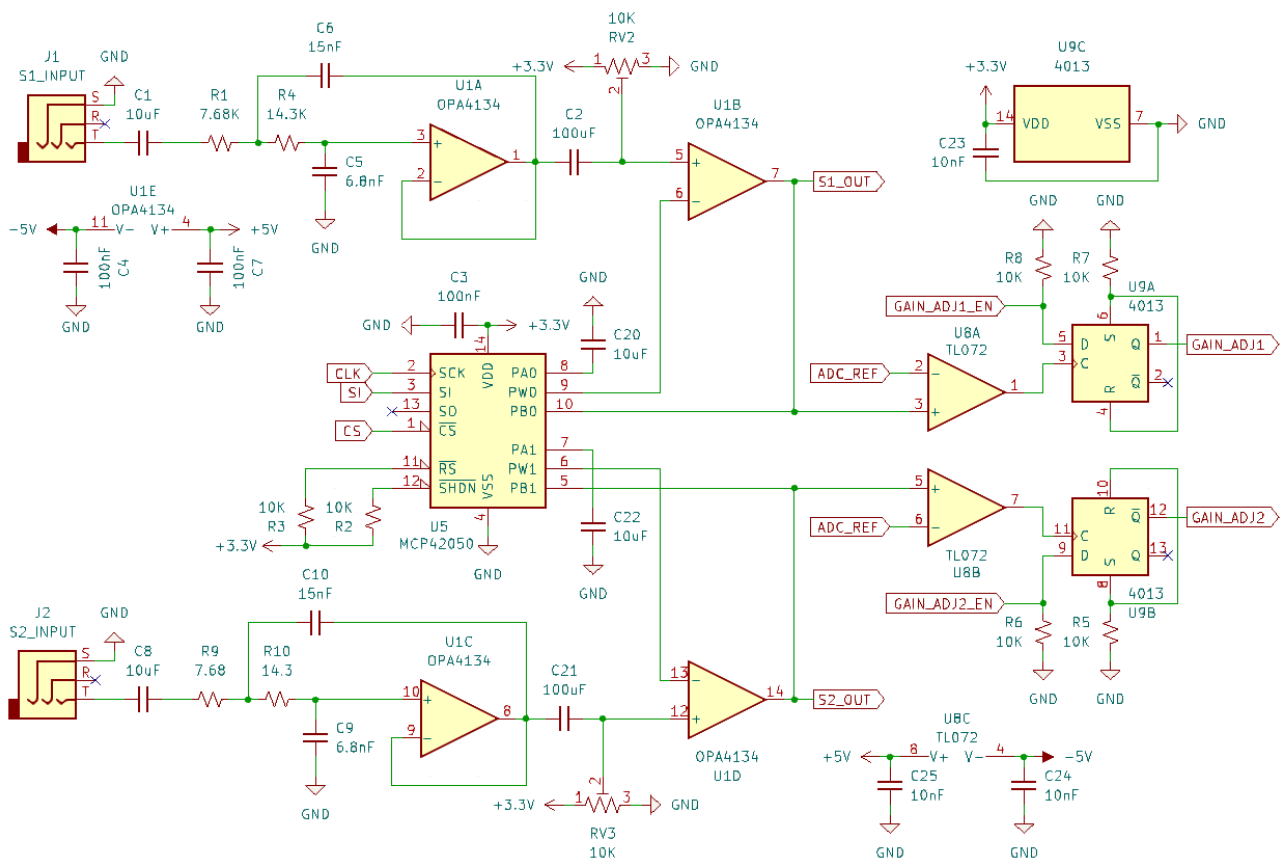


Fig. 4: Circuito de Medición de Corrientes.



Antes de entrar en detalle sobre el funcionamiento del circuito, debemos conocer los sensores a usar y sus especificaciones técnicas. Los sensores de corriente que usamos fueron los modelos SCT-013-010 de la marca YHDC, que tienen un rango de corriente de entrada de 0[A] a 10[A] con un rango de tensión de salida de 0[V] a 1[V] de acuerdo con su correspondiente hoja de datos [2].

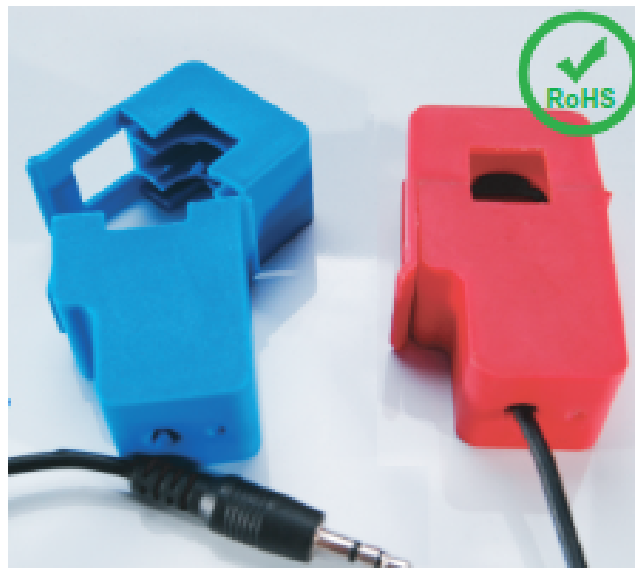


Fig. 5: Sensores de Corriente SCT-013[2].

Como se puede observar en la Fig. 5, los sensores constan de un núcleo de ferrite partido lo cual permite abrirlos y colocarlos sobre una instalación eléctrica ya finalizada sin tener que hacer modificaciones a la misma.

Su funcionamiento se basa en la inducción del campo magnético generado por la corriente alterna que origina una corriente en el bobinado de los sensores, los cuales, al tener una resistencia conectada a la salida del bobinado, hacen que lo que midamos a la salida de los sensores sea una señal de tensión alterna.

Decidimos usar estos sensores ya que nos aíslan de la red y nos permiten medir un rango de corriente adecuado para nuestro propósito, que es de baja potencia. Además, nos aseguran un error de fase máximo de 4 grados sexagesimales y un error de amplitud máximo de 1 % [15] lo cual es fundamental para nuestra implementación, debido a que los errores de fase y tensión se ven luego reflejados en el funcionamiento final del filtro.



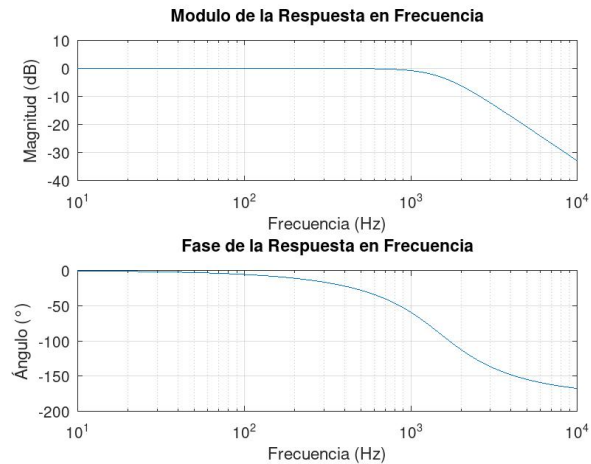


Fig. 8: Respuesta en Frecuencia del Filtro Antialiasing.

Empleamos este filtro antialiasing para así eliminar el ruido de alta frecuencia que puede llegar a encontrarse en la corriente de red, debido a que nuestro proyecto está limitado a tratar los armónicos hasta una frecuencia de 1[KHz]. En cuanto al orden del filtro, el hecho de usar una frecuencia de muestreo total de 500[Ksps] nos permite usar un filtro de orden bajo ya que la imagen de la señal estará muy alejada, en el dominio de la frecuencia, de la señal de interés[18].

Para comprobar el funcionamiento del filtro, empleando GNU Octave, generamos una señal alterna con una frecuencia fundamental de 50[Hz], su tercer armónico de 150[Hz], su quinto armónico de 250[Hz] y un armónico de 50[KHz], (el cual podría presentarse en una red doméstica debido a las altas frecuencias de conmutación de algunas fuentes de alimentación) y hemos aplicado el filtro a esta señal obteniendo el resultado de la Fig. 9.

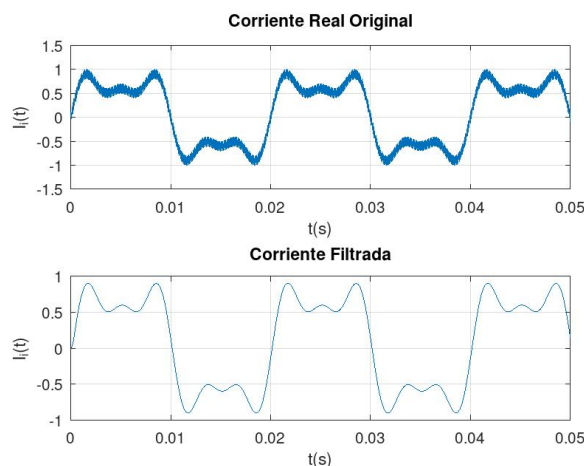


Fig. 9: Simulación del Filtro Antialiasing.



La siguiente etapa es la adición de una tensión continua de offset colocando el punto medio de un potenciómetro multivueeltas a la salida del filtro así obtuvimos la señal de tensión alterna sumada a una tensión de continua para así poder medir la señal con el ADC del microcontrolador. Como el ADC puede medir señales desde 0[V] hasta 3[V] la tensión de offset la ajustamos manualmente para centrar la señal alterna en los 1.5[V] de modo de poder amplificarla lo máximo posible y así aprovechar toda la resolución del ADC.

Teniendo la señal filtrada y con el offset necesario, aplicamos una amplificación en la siguiente etapa. El amplificador no inversor basado también en el OPA4134 cuenta con un potenciómetro digital (MCP42050), la ganancia del amplificador es ajustada a necesidad mediante el microcontrolador. El circuito del amplificador no inversor en conjunto con el potenciómetro digital está basado en el de la Fig. 10.

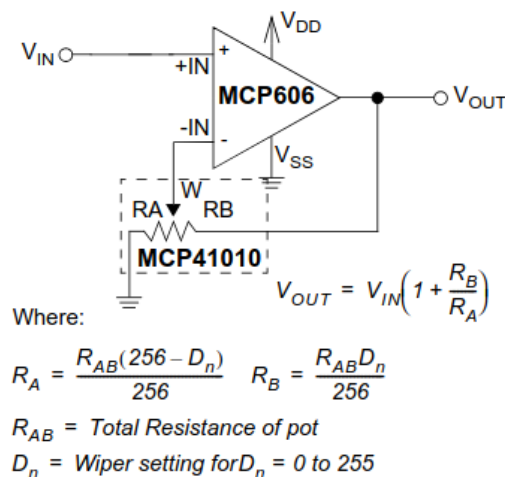


Fig. 10: Amplificador No Inversor con MCP41010 [3].

El MCP42050 es un potenciómetro digital de dos canales independientes, por lo que tenemos dos potenciómetros de 50[KΩ] cuyos pivotes, puntos medios o wipers (en inglés) pueden ser colocados en cualquiera de las 256 posiciones equidistantes de cada potenciómetro al enviar los correspondientes comandos al integrado mediante comunicación serie con protocolo SPI [3].

Gracias al uso del MCP42050 la ganancia del amplificador no inversor ( $G$ ) se puede controlar fácilmente estableciendo la posición del wiper ( $D_n$ ) que puede tomar cualquier valor entero entre 0 y 255.

$$G = 1 + \frac{D_n}{256 - D_n} \quad (1)$$



Para controlar la ganancia del amplificador no inversor implementamos un circuito con un comparador y un flipflop tipo D. Con este circuito de control, la salida del amplificador se compara con la tensión de referencia del ADC, es decir 3[V], y cómo la señal es alterna al ir aumentando la ganancia del amplificador no inversor llegará un momento en que la señal tendrá sus picos por encima de los 3[V] haciendo que la salida del comparador ahora sea una señal cuadrada. Esta señal cuadrada ingresa al flipflop tipo D actuando como señal de clock, y al tener un 1 lógico en el pin D, en el pin de salida Q tendremos también un 1 lógico.

De esta manera, la lógica de control del amplificador inversor es la siguiente:

1. Inicialmente la ganancia es  $G = 1[V/V]$ , hay un 0 lógico en  $D$  y también en  $Q$ .
2. Se incrementa la ganancia aumentando de uno en uno el valor de  $D_n$  cada 50[mS] y se coloca un 1 lógico en el pin  $D$  del flipflop.
3. Cuando los picos de la señal alterna superan la tensión de referencia (3[V]) tenemos a la salida del comparador una señal cuadrada que es la señal de clock del flipflop tipo D.
4. El flipflop al tener clock coloca un 1 lógico en  $Q$ .
5. El microcontrolador detecta el flanco ascendente y detiene el incremento de la ganancia.
6. Para la ganancia final se reduce en 1 el valor de  $D_n$  para no sobrepasar la tensión de referencia.

Antes de terminar con esta etapa debemos mencionar el capacitor a ground que tiene cada uno de los potenciómetros digitales del MCP42050, ya que estos permiten que el amplificador operacional aplique la ganancia dada únicamente a la componente alterna de la señal sin modificar así la tensión de offset que de otro modo se vería también afectada. El agregado de este capacitor hace que el circuito funcione tal cual como se ve en la Fig 11.

DC Bias of Opamps – Non Inverting Amplifier

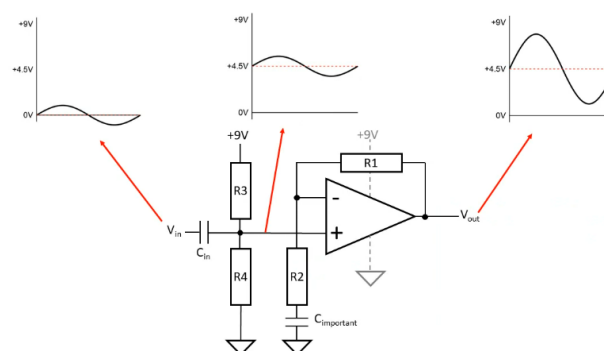


Fig. 11: Adición de Capacitor [4].



Si bien esta técnica se emplea cuando se miden señales alternas con amplificadores operacionales de alimentación simple, en nuestro caso usamos un amplificador con alimentación simétrica y el agregado del capacitor ( $C_{important}$  en la Fig. 11) hace que el offset dado anteriormente no se modifique.

Para resumir, las mediciones de corrientes se realizan a través del sensor SCT-013-10 y pasan primero por un filtro pasabajos, luego se les añade un offset y por último se ajustan las ganancias de las señales para aprovechar al máximo la resolución del conversor analógico digital de aproximaciones sucesivas que posee el microcontrolador RP2040. La evolución de la señal al pasar por este circuito se puede observar en la Fig. 12.

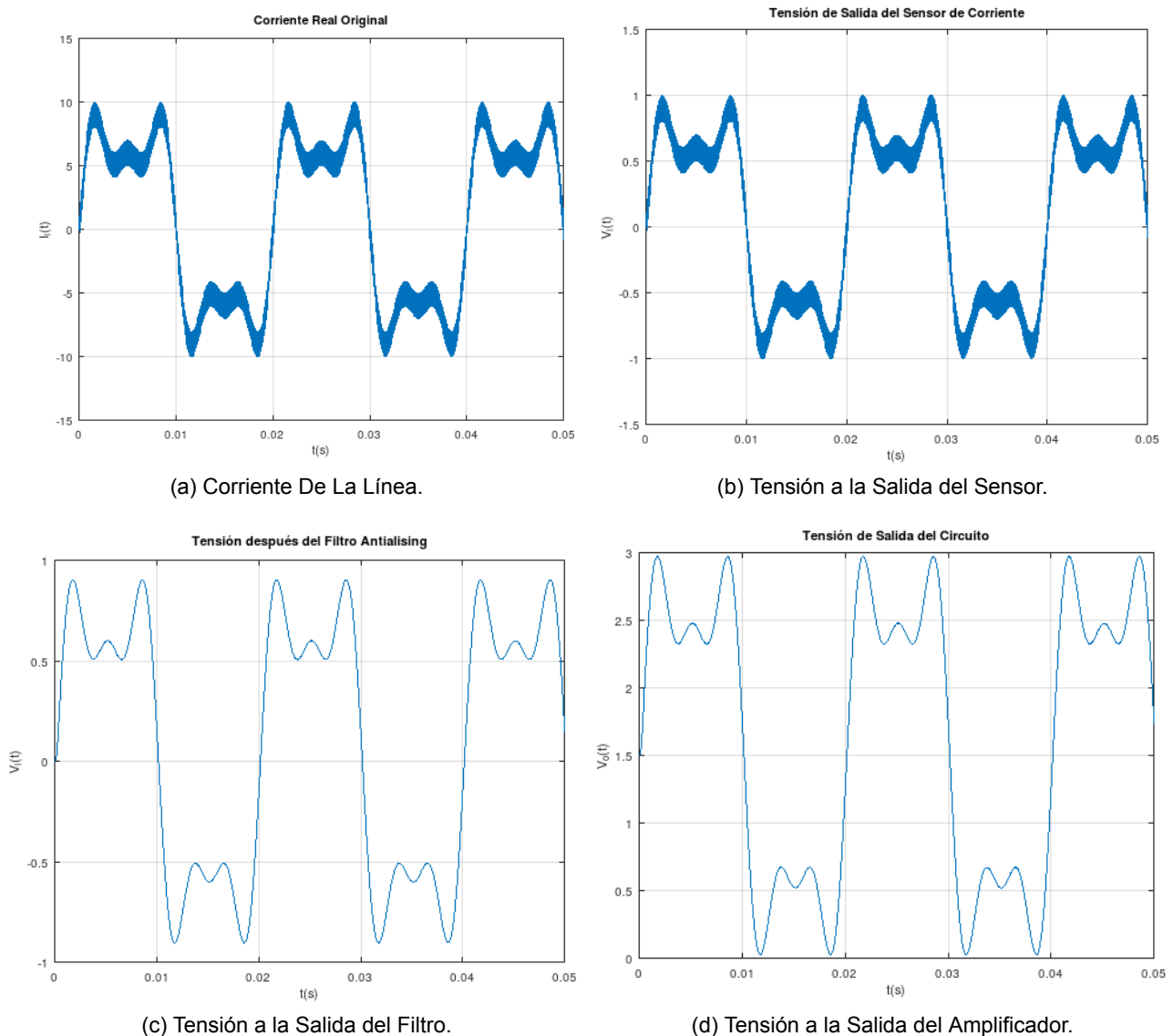


Fig. 12: Funcionamiento Completo del Circuito de Sensado de Corriente.



Como parte de la etapa de desarrollo, lo siguiente que hicimos fue diseñar y fabricar un primer prototipo de la placa de sensado, la cual no cuenta con la sección de control para el ajuste automático de ganancia ya que al comienzo establecíamos la ganancia manualmente para así poder ver como afecta el tener mayor o menor amplitud al proceso de filtrado. El diseño de la misma se realizó empleando el software gratuito de diseño asistido KiCad versión 8.0.

Al hacer las pruebas con distintas ganancias pudimos notar que la falta de información al no emplear todo el rango posible del ADC afecta directamente al funcionamiento del filtro, es por esto que luego implementamos en protoboard el ajuste automático de ganancia para así tener siempre el mejor rendimiento posible. De esta manera nuestra configuración de pruebas se puede observar en la Fig. 13.

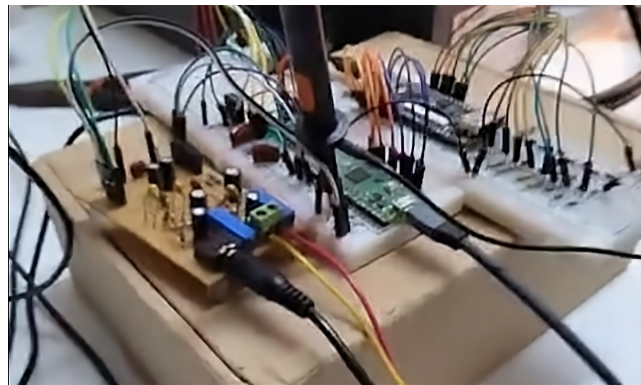


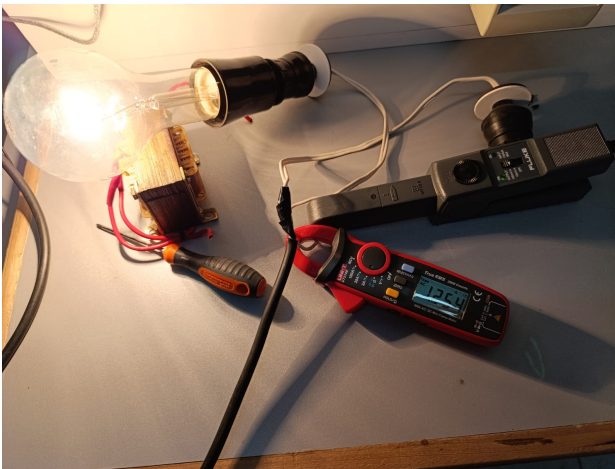
Fig. 13: Configuración de Pruebas Preliminares.

Para las pruebas preliminares usamos dos lámparas incandescentes, una de 75[W] y otra de 300[W], una plancha de 1100[W] y también un dimmer para generar armónicos. Una vez que verificamos con un osciloscopio que la señal efectivamente llega sin inconvenientes a la RP2040, todas las pruebas se realizaron empleando el software de visualización desarrollado en Python. Además, usamos un osciloscopio aislado Hantek DSO1062B con una pinza de medición de corriente de la marca Fluke como instrumento patrón para verificar la veracidad de nuestras mediciones.

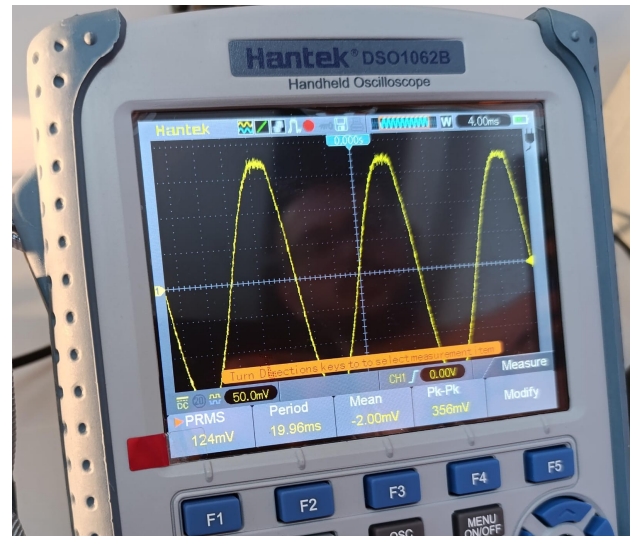
Como mencionamos previamente, la RP2040 no solo debe medir y mostrar mediante la aplicación la información sino que también debe enviársela a la FPGA que se encargará del filtrado. Para comprobar esta parte del funcionamiento usamos un analizador lógico de 24[MHz] y 8 canales de la marca Saleae con el cual pudimos ver bit a bit el comportamiento del puerto paralelo que transmite los datos desde la RP2040 a la FPGA.



La primera prueba que realizamos fue midiendo la corriente que consume una lámpara incandescente de 300[W], para lo cual dispusimos del montaje de la Fig. 14 en el Laboratorio de Potencia ya que el mismo cuenta con todos los elementos y sistemas de seguridad para pruebas con tensión alterna.



(a) Configuración de la Prueba.



(b) Osciloscopio Hantek con Pinza Fluke.

Fig. 14: Medición de Referencia.

Si bien se ve en la Fig. 14 una pinza amperométrica, la misma sirve para tener una referencia a simple vista de la corriente pero nuestra medición patrón es la dada por el osciloscopio Hantek junto con la pinza Fluke. De acuerdo con el fabricante [19] la medición de la pinza Fluke tiene un error de  $\pm 3\%$ , es por esto por lo que lo consideraremos como nuestro instrumento patrón.

Al tener un ajuste la pinza de  $100[mV/1A]$  como medimos  $124[mV]$  de tensión RMS en el osciloscopio Hantek, esto significa que tenemos  $1.24[A]$  de corriente. Usar el osciloscopio también nos sirvió para visualizar la forma de onda y compararla con nuestra aplicación de modo de asegurar que no hay una introducción de ruido o deformación no deseada.



Luego de tener nuestra medición patrón pasamos a medir la corriente que consume la lámpara incandescente con nuestra placa de sensado y aplicación desarrollada en Python.

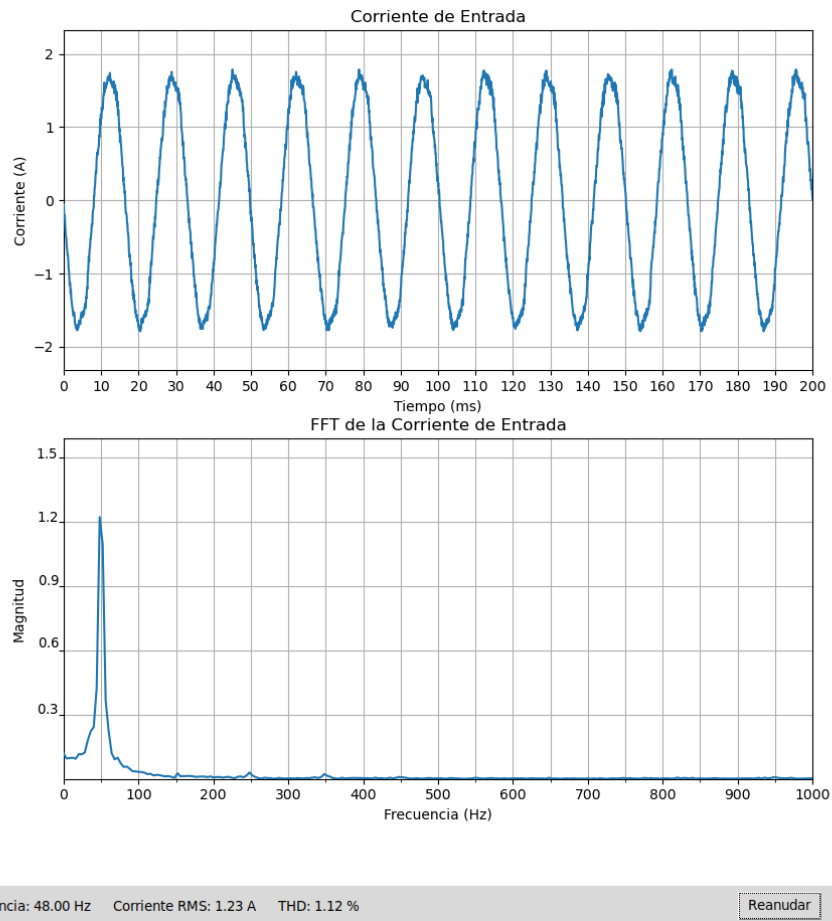


Fig. 15: Medición de Prueba Lámpara 300[W].

Como se puede observar en la Fig. 15 la corriente RMS medida fue de 1.23[A] por lo que tenemos un error de 10[mA] (0.8%) con respecto a la medición realizada por la pinza Fluke con el osciloscopio Hantek. Si bien se pueden observar que la escala temporal del gráfico no está correcta, en estas primeras pruebas buscamos comprobar el error de nuestra medición en cuanto a valor RMS de la corriente unicamente, ya que estos otros errores son debidos a cálculos del software y no debido a la electrónica que es la que estamos corroborando en primera instancia.



Después pasamos a repetir la prueba pero con la plancha de 1100[W]. Ahora obtuvimos, mediante la pinza Fluke en conjunto con el osciloscopio Hantek, una medición patrón de 4.82[A], la cual en este caso fue coincidente con nuestra medición por parte de la placa de sensado en conjunto con nuestra aplicación, como se puede observare en la Fig.16 .

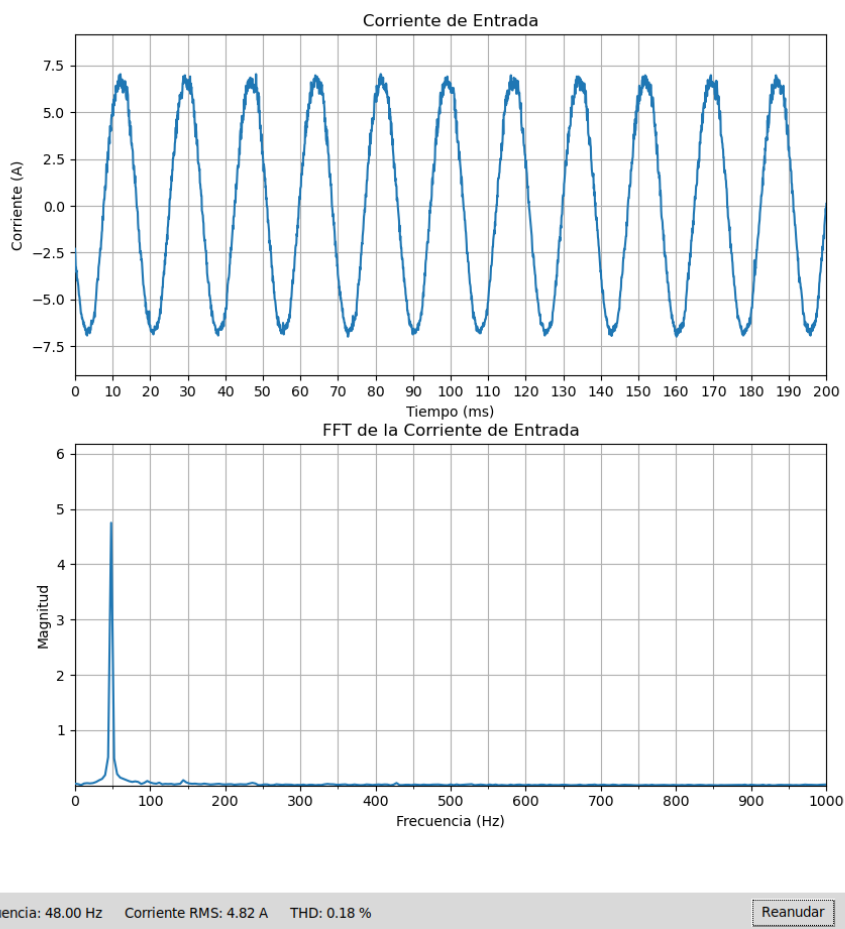


Fig. 16: Medición de Prueba Plancha 1100[W].

Con estas pruebas pudimos verificar el correcto funcionamiento de la electrónica de la etapa de sensado, al comprobar que nuestras mediciones de corriente RMS son concordantes con las de nuestro instrumento patrón. Un análisis más a detalle se verá luego en la sección Resultados de este mismo informe.



Luego lo que hicimos fue realizar una medición de la corriente que consume una lámpara incandescente de 75[W] pero esta vez con una mayor cantidad de muestras enviadas a la aplicación, pasando de 1200 a 2400 muestras por canal, y con una mayor tasa de muestreo, pasando de 450[Ksps] a 500[Ksps]. En este caso solamente hicimos la prueba mediante nuestra aplicación ya que lo más importante era ver si la visualización de la señal mejoraba, tanto en el dominio temporal como en el dominio frecuencial.

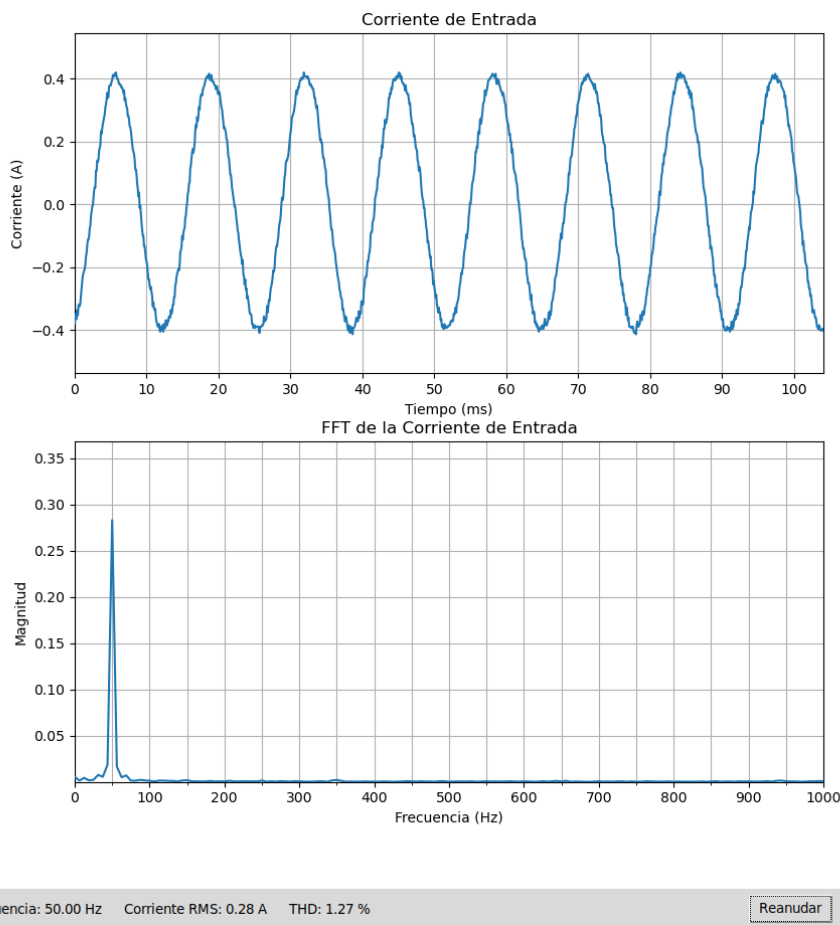


Fig. 17: Medición de Prueba Lámpara 75[W].

Como se puede observar en la Fig. 17 ganamos en definición haciendo que la señal se vea con más detalle y también la FFT resultante ahora tiene también menor base de ruido y una mejor definición del pico de potencia a la frecuencia medida. Esto se puede observar comparando esta medición con la de la Fig. 15, donde al no tener una gran amplitud, la frecuencia principal en la FFT se ve claramente la presencia de ruido base y además no está tan bien definido el gráfico de la FFT como lo está en esta última medición.



Una vez que comprobamos que las mediciones cuantitativas realizadas por nuestros sensores son coherentes con las mediciones de nuestro instrumento patrón, y teniendo ya establecido la frecuencia de muestreo en 500[KSPs], la siguiente prueba fue transmitir la información por un puerto paralelo implementado por ocho pines de uso general de la RP2040 siguiendo la frecuencia de clock dada por la FPGA.

Como se mencionó previamente, empleamos para esta prueba el analizador lógico de 24[MHz] y 8 canales de la marca Saleae, el cual configuramos para capturar los 8 bits del puerto paralelo a una tasa de 50 KSPs ya que la frecuencia de clock empleada para el envío de datos es de 15[KHz]. Usando el software provisto por Saleae se puede observar cada uno de los bits y luego exportar el valor decimal que forman los 8 bits cada 66.67[uS], lo que sería equivalente a tomar el valor decimal a la misma frecuencia del clock provisto por la FPGA.

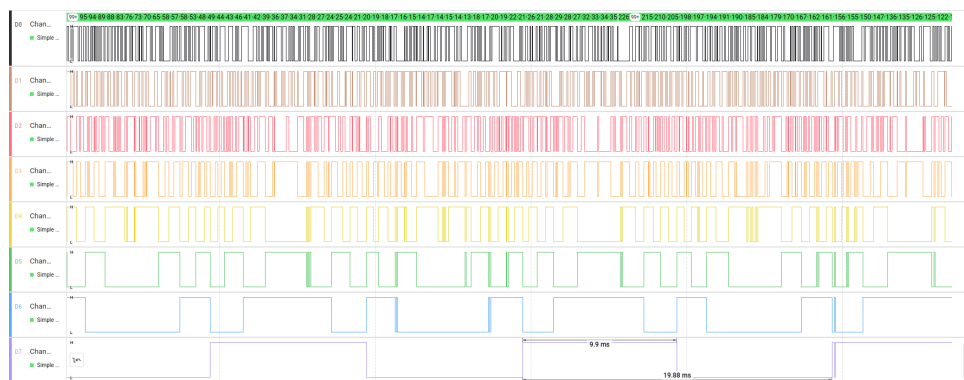


Fig. 18: Medición de Cada Bit del Puerto Paralelo.

La medición obtenida por el analizador lógico además de servir para visualizar el comportamiento del puerto paralelo, nos sirve para verificar la correcta medición de la señal ya que de estar correcto el offset aplicado, la señal se ubicará centrada en la totalidad del rango del ADC, por lo tanto, al ser la señal medida una senoidal sabemos que si esta correctamente centrada está pasará la misma cantidad de tiempo con valores por encima de 128 que con valores por debajo de 128, y esto lo podemos corroborar con la medición de la frecuencia del bit más significativo. En la Fig. 18 se puede observar esta comprobación que indica que la señal estaba correctamente centrada debido al offset aplicado ya que la frecuencia del bit más significativo es de aproximadamente 50[Hz].



Si bien vemos cada uno de los bits por separado en la aplicación del analizador lógico, para corroborar que la señal se está transmitiendo correctamente lo que hicimos fue tomar el archivo con formato csv generado con todos los valores decimales del puerto paralelo tomados cada 66.67 us y graficarlos empleando el software Calc de LibreOffice, de esta forma veremos la representación de la señal y comprobaremos que efectivamente es una senoidal de 50[Hz].

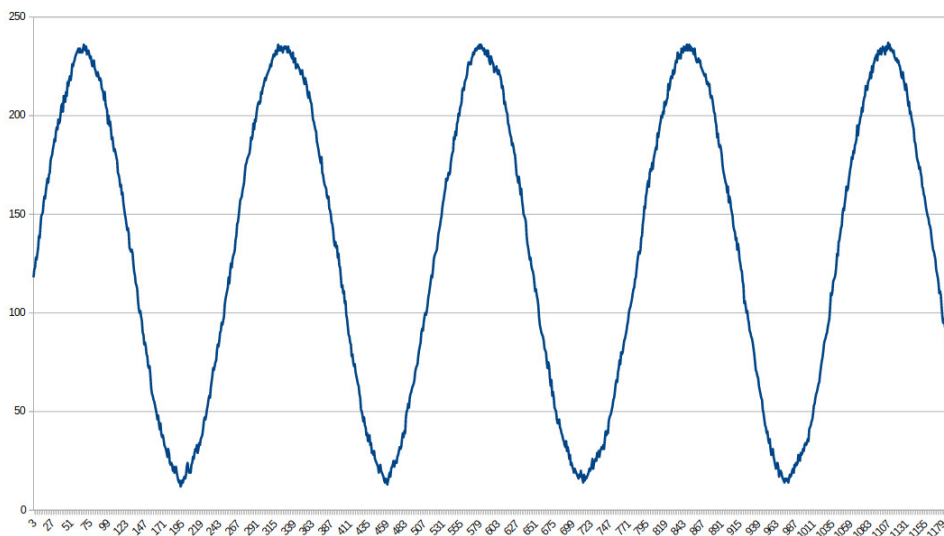


Fig. 19: Gráfico de las Muestras del Puerto Paralelo.

En la Fig. 19 se puede ver que la información transmitida por el puerto paralelo corresponde efectivamente a una señal senoidal. Además, sabiendo que cada muestra está tomada a 66.67[us] podemos calcular el período de la señal, el cual nos dio en este caso 18[mS] aproximadamente, muy cercano a los 20[mS] de una señal senoidal de exactos 50[Hz]. De este modo, concluimos que el puerto paralelo comandado por el PIO0 de la RP2040 funciona correctamente.



### 2.1.2. Sensado de la Corriente Resultante:

Debido a que no hacemos una inyección a la red de la corriente generada por el inversor controlado por la FPGA, la forma de observar la funcionalidad del filtro en si mismo es aprovechar que la FPGA contiene la información de cada una de las corrientes (La de carga a filtrar y la de los armónicos) y realizar en la misma la suma de ambas señales, mostrando el resultado obtenido a través de 8 pines digitales a los cuales le conectamos un conversor digital analógico usando la topología del circuito R2R y un amplificador operacional en configuración de seguidor de tensión como se ve en la Fig. 20.

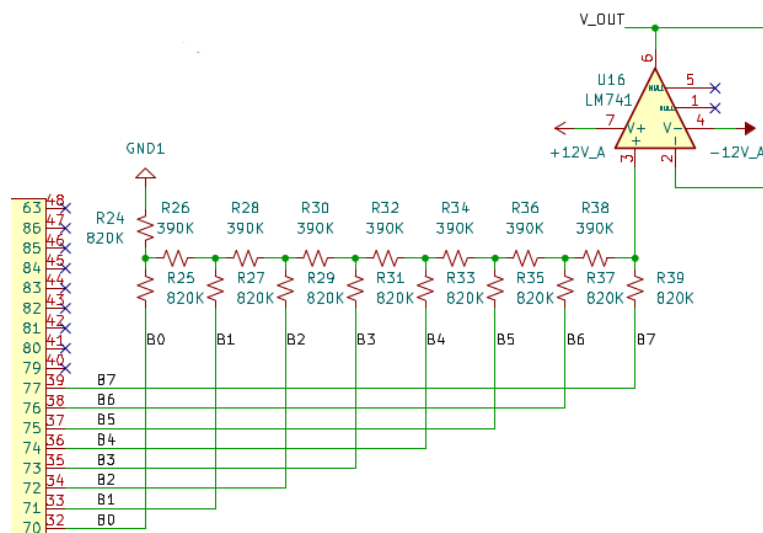


Fig. 20: Circuito de Medición de la Señal Resultante.

De esta forma la señal va directamente al tercer canal del ADC de la RP2040 para poder realizar las mediciones necesarias. Hay que tener en cuenta que la FPGA emplea las señales ya normalizadas en amplitud por lo que esta señal resultante también estará normalizada (Fig. 21). Sin embargo, esto no representa ningún inconveniente ya que lo más importante de esta señal es poder medir el THDi obtenido para así corroborar la efectividad del filtro implementado.

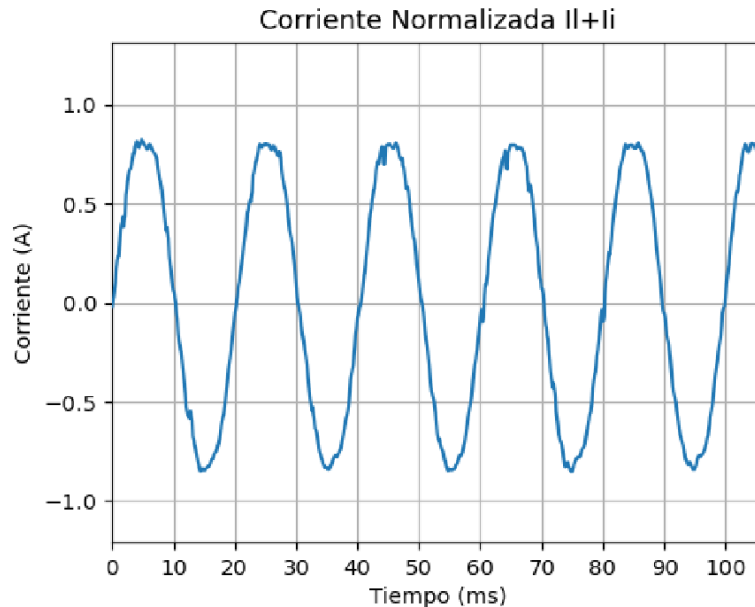


Fig. 21: Señal de Salida del R2R.

Una vez comprobamos el total funcionamiento del circuito de la etapa de filtrado, diseñamos su correspondiente placa de circuito impreso en el programa KiCad. En la Fig. 23 se puede observar el circuito completo contemplando el circuito ya explicado anteriormente sumado a las conexiones del microcontrolador RP2040 y la FPGA y la fuente de alimentación, la cual consta de tres reguladores de tensión para obtener así +12[V], -12[V] y 5[V], de donde el par de tensiones diferenciales irán a los amplificadores operacionales mientras que los 5[V] alimentarán a la RP2040 y a la FPGA.

Además de estas tensiones de alimentación, se puede observar que hay un voltaje de referencia de 3[V] provisto por el LM4040, esto lo hicimos siguiendo la recomendación del fabricante del microcontrolador ya que se especifica en la hoja de datos de la placa Raspberry Pi Pico que el ADC tiene un mejor funcionamiento si se limita su rango de tensión desde 0[V] a 3[V], en lugar de 3.3[V] [1].



También colocamos pines y jumpers para realizar pruebas de funcionamiento. Tenemos pines en paralelo a los conectores de los sensores para así poder ver las señales de entrada, pines con cada una de las tensiones para así ver que este funcionando correctamente la etapa de alimentación, pines en los canales del ADC de la RP2040 para así ver que señales está recibiendo el microcontrolador, y también pines hembra donde irán conectadas otras dos placas, una que será la placa del conversor digital analógico y otra que será necesaria para el inversor (esta otra placa se explica en la sección de Generación de Armónicos). Se ve además un jumper que es para poder dejar o quitar la alimentación por parte de la placa en el caso que estemos realizando pruebas o queramos programarla y solamente nos haga falta alimentar la RP2040 mediante el USB.

Además, tuvimos que hacer una corrección al circuito original ya que a pesar de que durante todas las pruebas preliminares funcionó perfectamente, observando la hoja de datos del CD4013[5], vimos que tiene un circuito de protección CMOS en cada pin de entrada (Fig. 22), es decir, tiene un diodo en directa a VDD (3.3[V]) y un diodo en inversa a VSS (GND), y como la salida del amplificador operacional TL072 será una cuadrada que varia desde +12[V] hasta -12[V], cuando se tengan 12[V] el diodo en directa a VDD conduciría conectando la la salida del operacional a los 3.3[V] de la RP2040 y cuando se tengan -12[V] el diodo en inversa a VSS conduciría conectando la salida del operacional a GND.

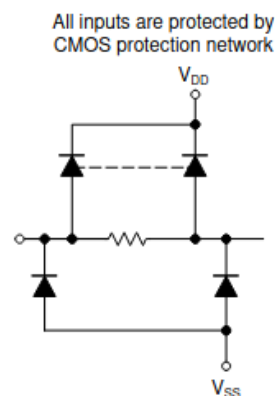


Fig. 22: Circuito de Protección CMOS del CD4013[5].



Si bien no tuvimos ningún inconveniente en las pruebas preliminares en protoboard, si observamos un sobrecalentamiento del operacional al pasar al circuito final con componentes SMD, por lo que investigamos y encontramos este inconveniente el cual fue solucionado agregando un diodo a la salida del operacional, eliminando la tensión negativa, y seguido de un divisor resistivo así ahora a la entrada del CD4013 se tendrá una cuadrada que variará desde 0 a 3.3[V]. De esta forma, el circuito completo que corresponde a la medición o sensado de señales se puede observar en la siguiente figura.

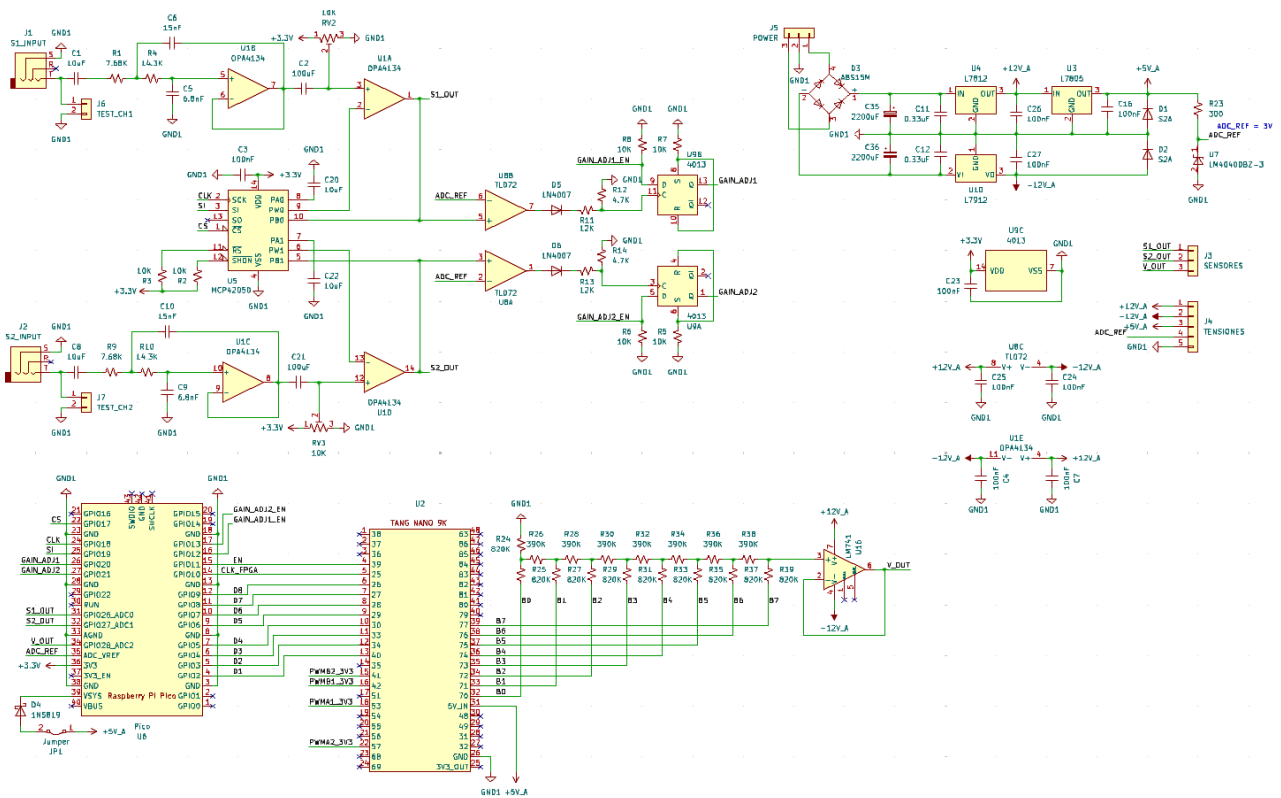


Fig. 23: Esquemático Completo del Circuito de Sensado.



Una vez listo el diseño del PCB principal, generamos los archivos gerbers correspondientes y la enviamos a fabricar. Una vez recibida la placa, al ya contar con todos los componentes necesarios comenzamos con la etapa de soldadura de los mismo obteniendo como resultado final la placa que se observa a continuación.

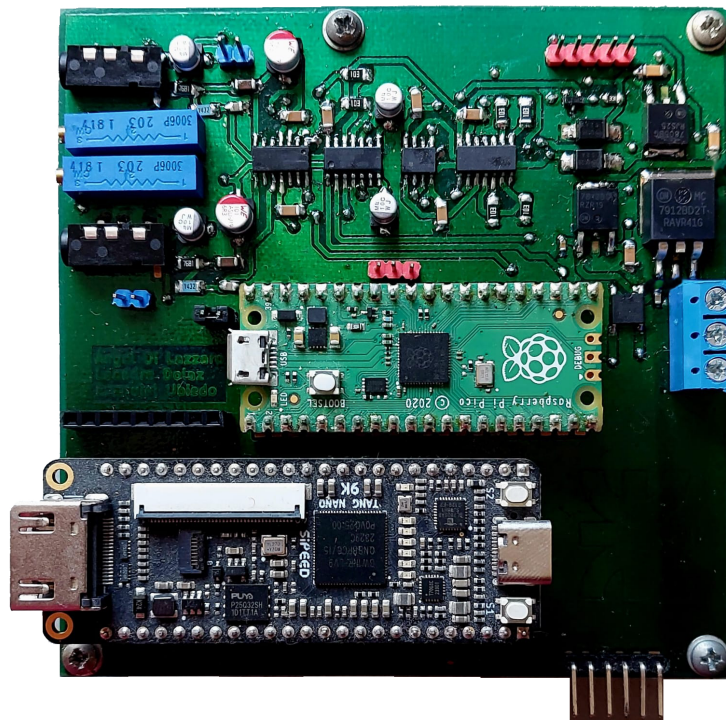


Fig. 24: PCB Principal de la Etapa de Sensado.

Para la placa del convertor analógico digital, la misma decidimos fabricarla de manera casera debido a las pequeñas dimensiones de la misma, quedando finalmente de la siguiente manera.

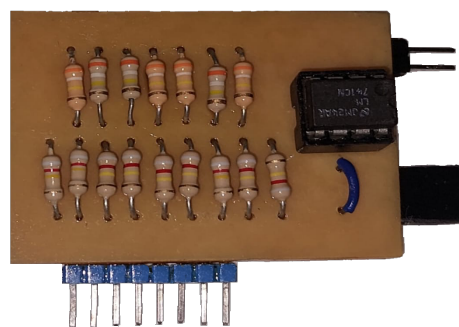


Fig. 25: PCB del Convertor Analógico Digital.



## 2.2. Etapa de Filtrado y Generación de señales PWM

Para la etapa de filtrado, utilizamos una FPGA debido a su versatilidad, ya que permite implementar circuitos digitales optimizados mediante un lenguaje de descripción de hardware (HDL)[20]. Esto nos brindó flexibilidad para configurar los recursos lógicos y aritméticos según los requerimientos del sistema, priorizando la reducción de área y consumo de potencia sin afectar el rendimiento[21]. Para optimizar aún más el uso de hardware, empleamos aritmética de punto fijo en lugar de punto flotante, lo que permitió representar valores numéricos de manera eficiente y con menor costo en términos de recursos[6].

La representación en punto fijo es un tipo de representación numérica que mantiene el punto decimal fijo dentro de un conjunto de bits. El mismo establece una determinada cantidad de bits para el signo, la parte entera y la parte fraccionaria. A diferencia del punto flotante, el formato de punto fijo realiza las operaciones aritméticas con números enteros, lo que facilita su implementación en hardware[6]. Nuestra tarea consistió en analizar el impacto del error de cuantización, seleccionando formatos de representación adecuados según la precisión y el rango de valores necesarios.

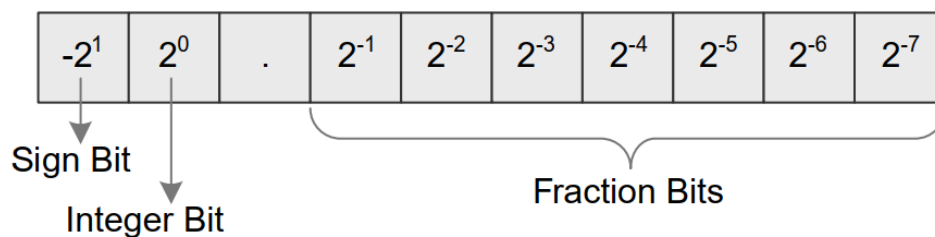



Fig. 26: Aritmética de Punto Fijo [6].

 <b>UTN Regional Paraná</b>	<b>Filtro Armónico Activo</b>
<b>Ingeniería en Electrónica</b>	<b>2.Desarrollo</b>

### 2.2.1. Modelado de Algoritmos en Punto Fijo

Para evaluar la implementación en punto fijo, desarrollamos modelos equivalentes en punto flotante y punto fijo, empleando el sistema de cómputo numérico MATLAB en su versión R2021b. Así, el hardware requerido para realizar operaciones aritméticas en punto fijo se reduce sencillamente a operaciones con variables del tipo entero. Sin embargo, la cuantización de las variables reales introduce distintos tipos de errores que se propagan a lo largo del procesamiento de los distintos bloques digitales. Es por ello que inicialmente se modela el algoritmo a implementar tanto en punto flotante como en punto fijo, de manera que su análisis permita observar sus comportamientos y evaluar conclusiones. El objetivo es que ambos modelos sean lo más parecido posible en cuanto a su comportamiento.

Para modelar un algoritmo en punto fijo se deben conocer las magnitudes máximas y mínimas involucradas tanto de las variables internas y externas como las constantes. Esto permite establecer un tipo de formato de punto fijo para cada etapa del algoritmo con el fin de determinar una resolución adecuada para todo el rango de valores presentes. Un algoritmo en punto fijo involucra la cuantización de las constantes, de las señales de entrada y salida así como también de las variables internas. Todas ellas se relacionan mediante operaciones aritméticas que involucran un incremento en la cantidad de bits, ya que en operaciones como la suma se debe contemplar bits de acarreo, así como también en las multiplicaciones el valor resultante tiene un ancho de palabra igual a la suma de sus operandos. Para reducir los bits se emplean métodos de saturación, desbordamiento, truncamiento y redondeo con la finalidad de mantener la precisión del resultado.



## 2.2.2. Compensación Armónica: El Filtro Adaptativo

La compensación armónica de red requiere un procesamiento digital de señales en tiempo real. La misma genera la corriente armónica presente en la corriente de carga con el fin de realizar una cancelación mediante la expresión  $i_s = i_L - i_F$ , donde  $i_s$  representa la corriente de red,  $i_L$  la corriente de carga e  $i_F$  la corriente armónica generada por el algoritmo digital. Ambas señales deben estar en fase para que la operación sea efectiva. Para su implementación, diseñamos una etapa que convirtió la señal  $i_F$  del dominio digital a una señal de potencia en el dominio analógico, empleando una modulación PWM seguida de un inversor o convertidor DC-AC conectado a la red. Las señales PWM se generaron a partir de comparar continuamente la señal  $i_F$  con una señal portadora de mayor frecuencia.

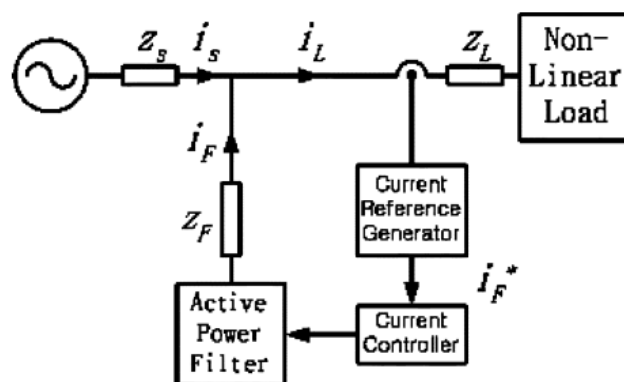


Fig. 27: Compensación Armónica: El Filtro Adaptativo [7].

## 2.2.3. Marco teórico

El filtro adaptativo consiste en un algoritmo que ajusta continuamente los coeficientes del filtro FIR hasta obtener la respuesta esperada, que resulta nada menos que la fundamental de 50[Hz]. Se emplea en aplicaciones con señales cambiantes, como cargas no lineales con contenido armónico variable o canales de comunicación con respuesta en frecuencia variable. A diferencia de los filtros convencionales que introducen retardo de fase, el filtro adaptativo permite generar una señal armónica en fase con la corriente de carga, facilitando la cancelación de armónicos.

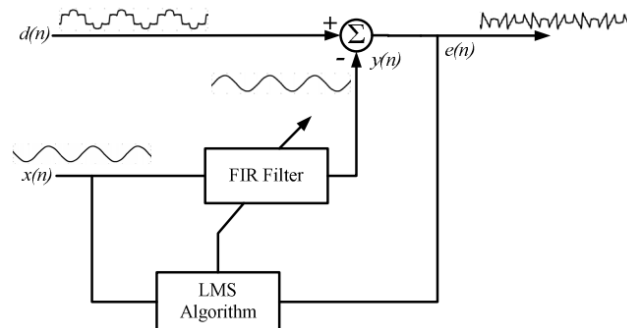


Fig. 28: Ecualizador Adaptativo[8].

Para actualizar los coeficientes del filtro, aplicamos el criterio de error cuadrático medio (LMS), el cual genera una señal de error que resulta de la diferencia entre la señal con contenido armónico y la salida del filtro FIR. Esta señal de error es la que contiene los armónicos que posteriormente serán inyectados en contrafase a la red. La señal de entrada del filtro FIR consiste en una señal de referencia que tiene correlación con la señal armónica. La misma proviene de la fundamental de 50[Hz] de la corriente de carga, ya que la señal de error que se genera debe carecer de esta componente, pues es la que se debe preservar en la red. El funcionamiento del filtro adaptativo se resume en (2).

$$\begin{aligned}
 y(n) &= w(n)x(n) \\
 e(n) &= d(n) - y(n) \\
 w(n+1) &= w(n) + 2\mu e(n)x(n)
 \end{aligned}
 \tag{2}$$

El filtro adaptativo procesa cuatro señales principales: la señal  $x[n]$ , que utilizamos como señal de referencia; la señal  $d[n]$ , que representa la corriente armónica; la señal de error  $e[n]$ , que refleja la diferencia entre  $d[n]$  y la salida filtrada  $y[n]$ ; y los coeficientes  $w[n]$  del filtro FIR, que son ajustados dinámicamente para minimizar el error. Esto ocurre a partir de emplear el opuesto al gradiente del error cuadrático medio, lo que permite encontrar el mínimo absoluto de la función de error, que resulta ser el contenido armónico de  $d[n]$ . La velocidad con la que el filtro realiza esta función depende de la constante  $\mu$ .

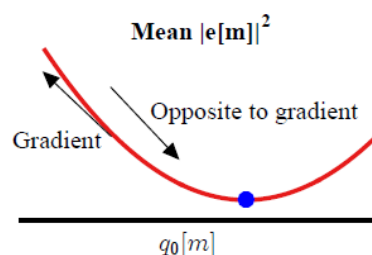


Fig. 29: Curva de error cuadrático medio[9].



La Fig. 30 representa la evolución de los coeficientes del filtro FIR en el tiempo. Durante esta etapa transitoria los coeficientes convergen hasta un valor final en el cual el error alcanza su mínimo absoluto. Debido a que la señal de error contiene los armónicos de  $d[n]$  y es empleado para ajustar continuamente los coeficientes, el mismo genera cierta variabilidad en los coeficientes. Para minimizarlo, se debe reducir el parámetro  $\mu$  a costa de producir una respuesta dinámica más lenta del filtro, lo que lo puede volver inestable. La solución a este inconveniente consiste en emplear un criterio para emplear distintos valores de  $\mu$  durante el ajuste de los coeficientes, con el objetivo de obtener una respuesta dinámica rápida y estable.

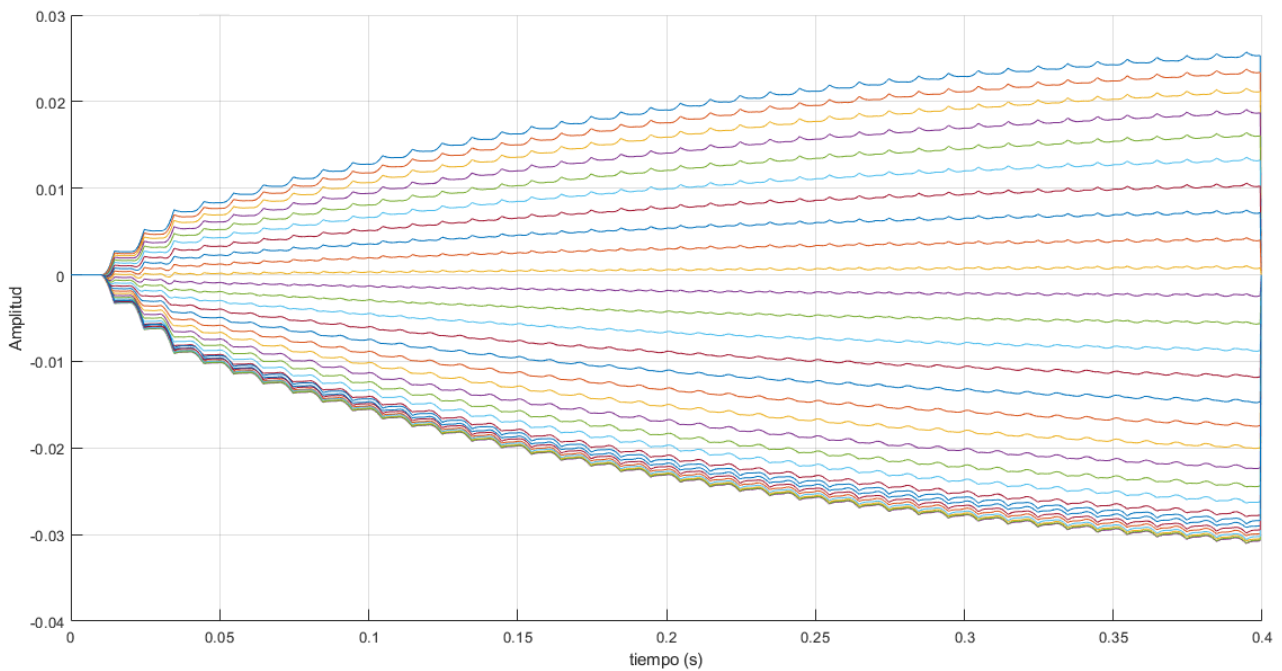


Fig. 30: Coeficientes del Filtro FIR.



## 2.2.4. Diagrama de Bloques Completo de la FPGA

A continuación, se presenta el diagrama de bloques general de todo el algoritmo que implementamos dentro de la FPGA. Para programar la FPGA empleamos el software que dispone GOWIN EDA para ello.

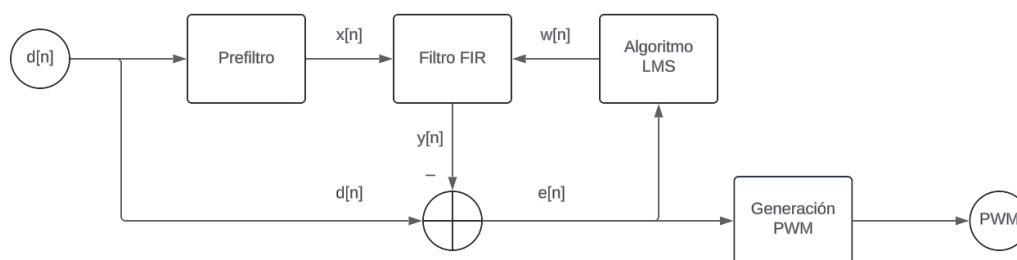


Fig. 31: Diagrama en Bloques de la FPGA.

A nivel general se tuvieron cuatro bloques principales:

- **El prefiltro:** Empleamos un filtro digital para eliminar el contenido armónico de  $d[n]$ , con el objetivo de generar la señal de referencia  $x[n]$ , correspondiente a la corriente fundamental. Al mantener una respuesta constante, cualquier atenuación en la banda de paso se compensa a la salida. La fase introducida por el filtro no afecta la generación de la corriente armónica, ya que el filtro adaptativo se encarga de compensar la fase.
- **El filtro FIR:** Implementamos un filtro digital con coeficientes variables en tiempo real. La cantidad de coeficientes dependió de la capacidad del mismo para filtrar de manera óptima los armónicos. Realizamos varias simulaciones hasta encontrar que con 60 coeficientes el filtro tenía un comportamiento aceptable.
- **Algoritmo LMS:** Basado en el error cuadrático medio, se buscó minimizar el error  $e[n]$  resultado de la diferencia entre  $d[n]$  y  $y[n]$ , representando la corriente armónica en  $d[n]$ . Este resultado lo utilizamos para actualizar los coeficientes del filtro FIR y adaptar su respuesta en tiempo real.
- **Generación de PWM:** A partir de la señal de error  $e[n]$ , generamos el PWM para el inversor de potencia, para así tener la capacidad de inyectar la corriente armónica en la red si se aumentara la tensión del inversor (es de 12[V] en nuestro prototipo). Derivamos el PWM de dos portadoras triangulares, una positiva y otra negativa, comparadas ciclo a ciclo con  $e[n]$ . Implementamos tiempos muertos para evitar la conducción simultánea de los IGBTs.



La Fig. 32 muestra el algoritmo completo implementado en la FPGA, detallando los relojes y reset de cada bloque. Incluimos divisores de frecuencia (LSCLK) para generar las distintas frecuencias necesarias para el funcionamiento del algoritmo, como la frecuencia de muestreo del filtro y la del generador interno del bloque SPWM encargado de modular la señal de error  $e[n]$ .

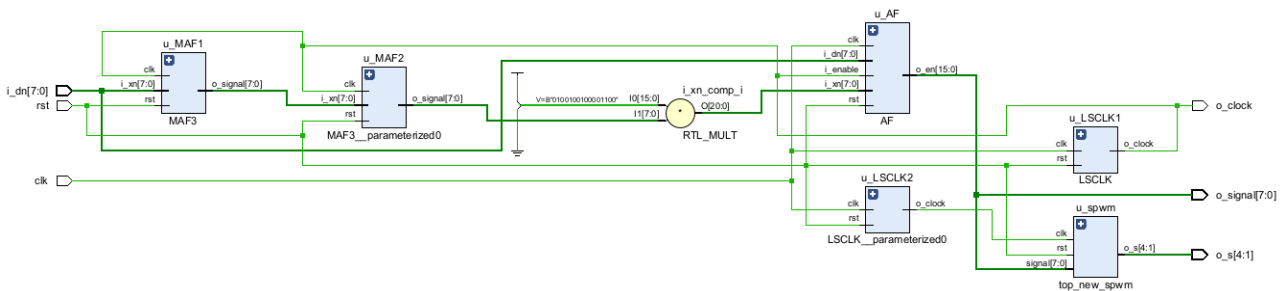


Fig. 32: Algoritmo en Bloques de la FPGA.

### 2.2.5. Prefiltro

El prefiltro cumple la función de eliminar el contenido armónico de la señal  $d[n]$  para generar la señal de referencia  $x[n]$ . La selección del tipo de filtro la basamos en su complejidad y velocidad de respuesta. Dado que las corrientes armónicas impares predominan en la red eléctrica, optamos por un filtro de media móvil.

### 2.2.6. Filtro de Media Móvil

El filtro de media móvil es un filtro FIR relativamente simple en el que todos sus coeficientes son iguales. Su operación consiste en promediar un conjunto de muestras. Si el promedio lo realizamos sobre un ciclo completo de un armónico específico, este lo eliminamos por completo junto con sus múltiplos. La Fig. 33 representa el diagrama de bloques del filtro, en donde las muestras pasadas de la entrada se generan a partir de los bloques de retardo unitario "D". En (3) se tiene la expresión matemática resultado de promediar L muestras de la señal de entrante.

$$y[n] = \frac{1}{L} \cdot (x[n] + x[n - 1] + \dots + x[n - L + 1]) \tag{3}$$

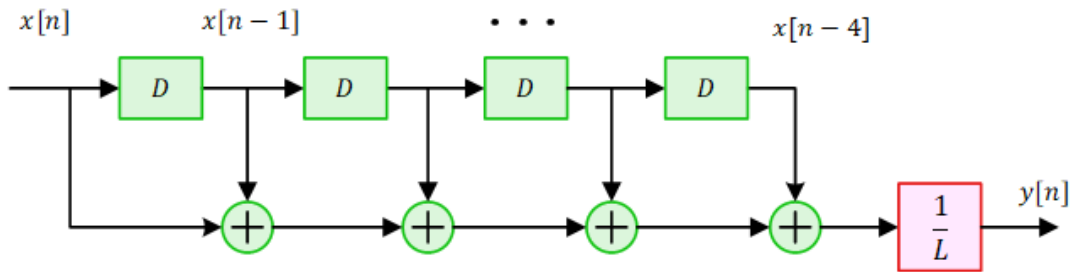


Fig. 33: Filtro de Media Móvil[10].

Para eliminar los armónicos de tercer y quinto orden en una red de 50[Hz], implementamos dos filtros de media móvil en cascada. Con una frecuencia de muestreo de 15[KHz], requerimos 100 y 60 muestras, respectivamente, para promediar un ciclo de 150[Hz] y otro de 250[Hz]. La combinación de ambos filtros resultó en una atenuación suficiente para eliminar los armónicos impares, como se muestra en la Fig. 34.

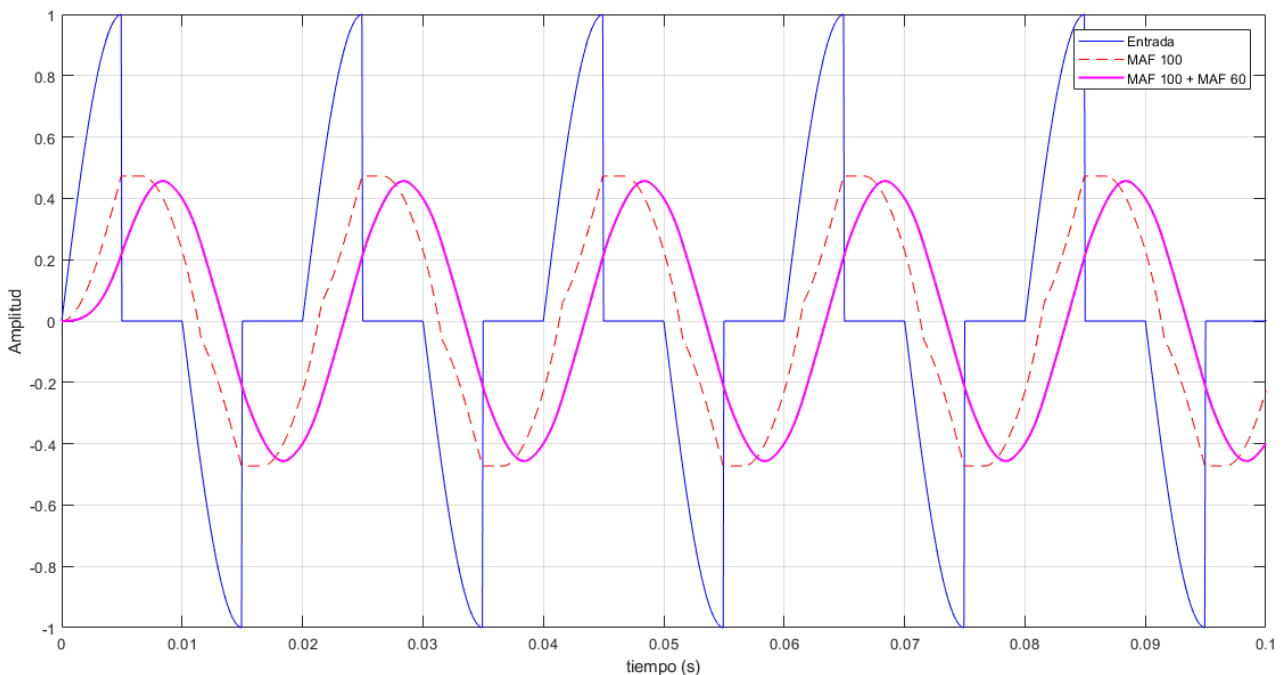


Fig. 34: Filtrado de la Corriente Armónica.



Cada filtro por si solo introduce inicialmente una atenuación de aproximadamente 13[dB] en la banda de rechazo, incrementándose con la frecuencia. La atenuación total en la frecuencia fundamental de 50[Hz] fue de -2.233[dB], lo que requirió amplificar la salida en un factor lineal de 1.29 para compensar la amplitud. Al conectar los filtros en cascada, logramos mayor atenuación para armónicos como el séptimo (350[Hz]), superando los 30[dB], tal como se muestra en la Fig. 35.

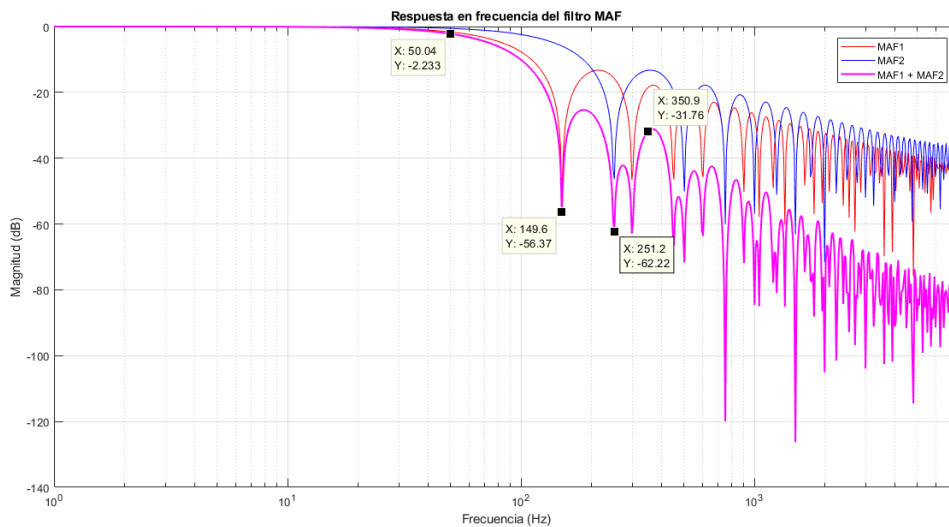


Fig. 35: Filtros de media móvil en cascada.

### 2.2.7. Filtro de Media Móvil Recursivo

El filtro de media móvil tradicional requiere la suma de todas las muestras en cada iteración, lo que demanda mayor cantidad de operaciones. Una alternativa más eficiente fue el filtro de media móvil recursivo, que empleó un par de sumas y una multiplicación, como se muestra en la Fig. 36.

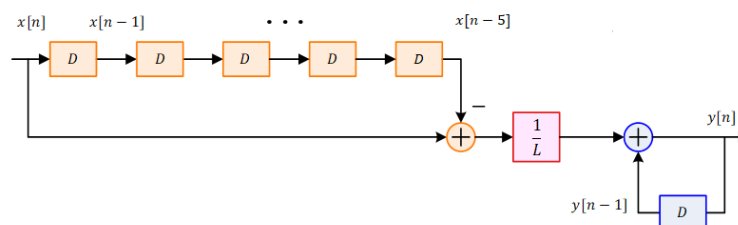


Fig. 36: Filtro de Media Móvil Recursivo[10].

Este filtro mantiene el buffer de  $L$  muestras y calcula la salida restando la muestra más antigua y sumando el promedio anterior.



### 2.2.8. Simulación del Filtro de Media Móvil Recursivo

Una vez seleccionada la arquitectura del filtro, realizamos su descripción en Verilog e implementamos un banco de pruebas para evaluar su funcionamiento en el software Vivado de Xiling. La simulación confirmó que la salida del filtro contuvo solo la frecuencia fundamental de 50[Hz].

En la herramienta de simulación, visualizamos las variables en formato analógico, facilitando el análisis del comportamiento del filtro, como se muestra en la Fig. 37. La amplitud de la salida resultante de los dos filtros en cascada se corresponde con el de la Fig. 34.

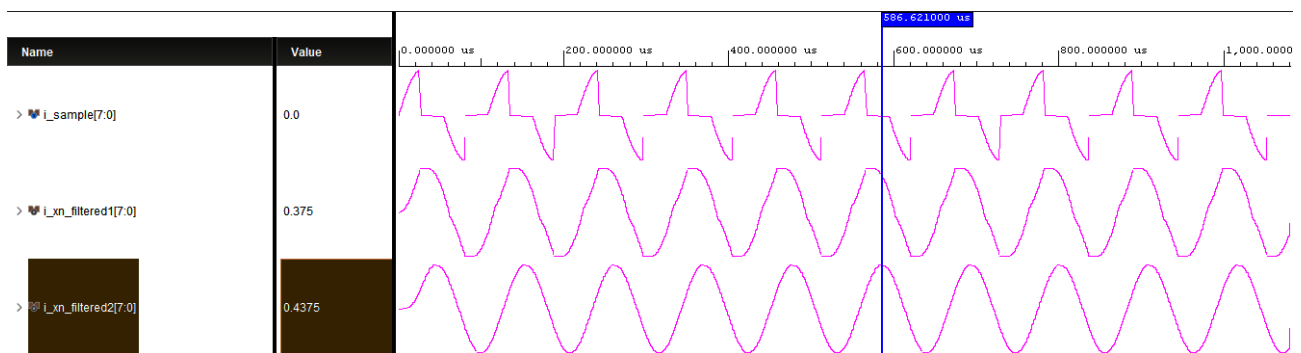


Fig. 37: Simulación del Filtro de Media Móvil Recursivo.

En la Fig. 38 se encuentran los dos filtros MAF en cascada seguido del multiplicador que compensa la atenuación producida en los 50[Hz], mientras que la Fig. 39 ilustra los elementos combinatoriales (suma, resta, y multiplicación) y secuenciales (registros) implementados en el filtro de media móvil recursivo.

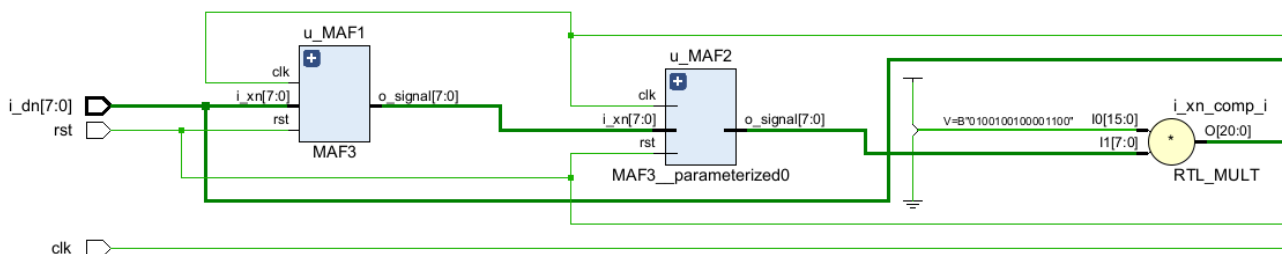


Fig. 38: Filtros de Media Móvil en cascada.

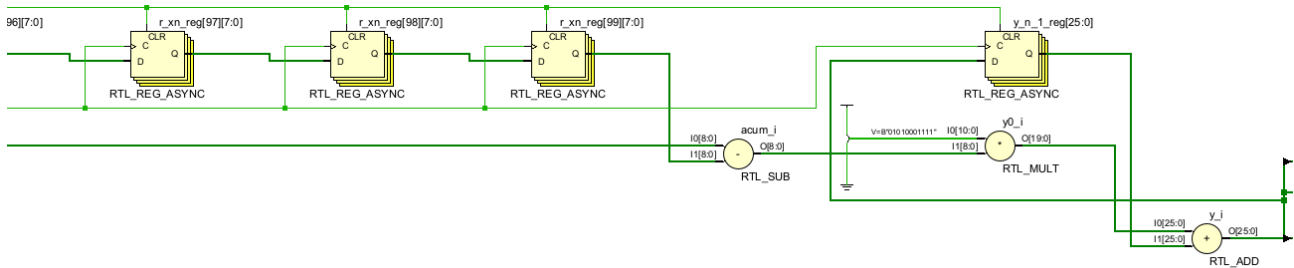


Fig. 39: Implementación del Filtro de Media Móvil Recursivo.

### 2.2.9. El Filtro FIR y el Algoritmo LMS

El filtro FIR y el algoritmo LMS conforman el filtro adaptativo, dado que uno realiza el filtrado mientras que el otro ajusta sus coeficientes de forma dinámica. Debido a la complejidad del algoritmo, definimos rigurosamente el formato de punto fijo de todas las variables y operaciones involucradas. Por ello, modelamos el algoritmo en punto fijo previo a su descripción en hardware, con el objetivo de que fuera una copia fiel del modelo ideal en punto flotante.

- **Filtro FIR:** Implementamos un filtro de 60 coeficientes, con  $x[n]$  como señal de entrada y  $y[n]$  como salida, la cual utilizamos para realizar la substracción  $e[n] = d[n] - y[n]$ . La estructura del filtro fue del tipo directo, por lo que, a medida que registramos las muestras de  $x[n]$ , realizamos las multiplicaciones por los coeficientes seguidas de una suma para conformar la salida. Optimizamos la cantidad de hardware utilizado, evitando la implementación de 60 multiplicadores a cambio de uno solo, a cambio de incrementar la frecuencia interna de funcionamiento.
- **Algoritmo LMS:** Computamos el valor siguiente de los coeficientes del filtro FIR. Utilizamos el parámetro  $\mu$  para determinar la velocidad de convergencia y la precisión del sistema. Un valor reducido de  $\mu$  implica una respuesta dinámica lenta pero precisa, mientras que un valor elevado permite una respuesta rápida pero con posibles oscilaciones. Implementamos un método para modificar dinámicamente  $\mu$ , logrando mayor velocidad de convergencia al inicio y mejor precisión al final.



La Fig. 40 y la Fig. 41 son el resultado de simular las respuestas del filtro en el dominio del tiempo y la frecuencia, comparando los modelos en punto fijo y punto flotante. Al no observarse diferencias significativas, validamos la exactitud del modelo en punto fijo. La cantidad de bits utilizada estuvo determinada por la señal  $d[n]$ , obtenida a partir de un conversor A/D de 8 bits. Las demás variables internas tuvieron una resolución mayor con el objetivo de incrementar la precisión de los cálculos.

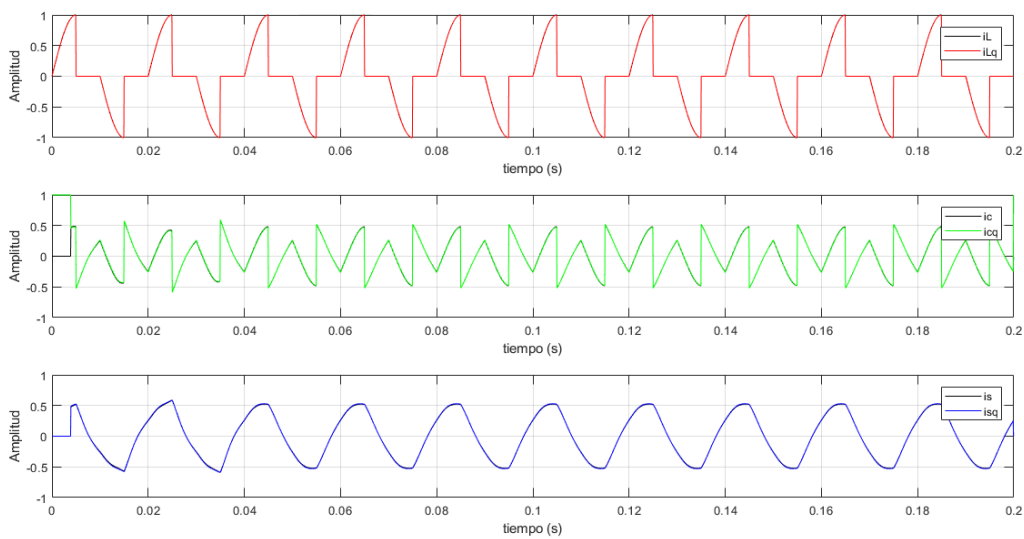


Fig. 40: Respuesta temporal del filtro adaptativo.

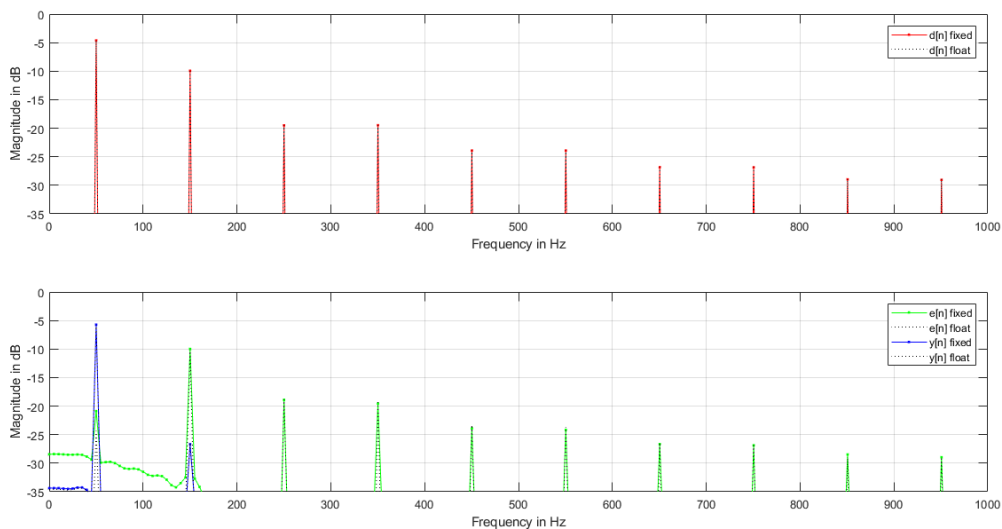


Fig. 41: Respuesta espectral del filtro adaptativo.



Una vez validado el modelo en punto fijo, procedimos con la descripción en hardware, respetando los formatos de cada variable para replicar el comportamiento obtenido en simulación, como se observa en la Fig. 42.

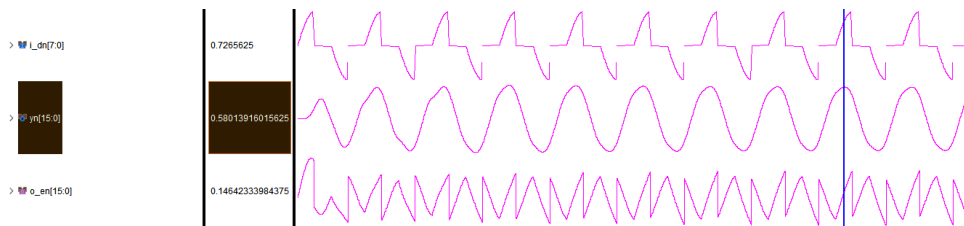


Fig. 42: Simulación del filtro adaptativo en Vivado.

Implementamos el filtro adaptativo con las entradas  $d[n]$  y  $x[n]$ , y la salida correspondiente a la corriente de error  $e[n]$ , que contiene los armónicos de 50[Hz]. La principal complejidad de este bloque se debió al gran número de coeficientes del filtro FIR y la cantidad de multiplicadores requeridos. Para optimizar el uso de recursos, utilizamos un solo multiplicador, aumentando la frecuencia interna de procesamiento. De este modo, ejecutamos 60 multiplicaciones a partir de un solo multiplicador en el tiempo equivalente a un solo ciclo de muestreo. Esto lo logramos mediante una frecuencia de reloj interna 60 veces mayor que la señal de habilitación  $i\_enable$ .

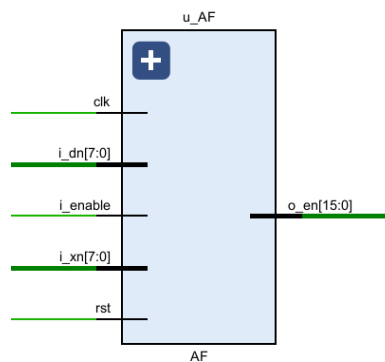


Fig. 43: Bloque del Filtro Adaptativo.

El circuito interno del filtro adaptativo estuvo compuesto principalmente por registros de muestras de entrada, memorias para almacenar coeficientes, multiplexores, sumadores y un multiplicador. La selección de la multiplicación a realizar la gestionamos mediante un contador de ciclos de reloj y un multiplexor, asignando en cada ciclo la operación correspondiente a  $x[n - N] \cdot h[N]$ . Este bloque representa la mayor complejidad en términos de recursos utilizados dentro del sistema, lo cual se reflejó en los resultados obtenidos y en el análisis de eficiencia del hardware implementado.



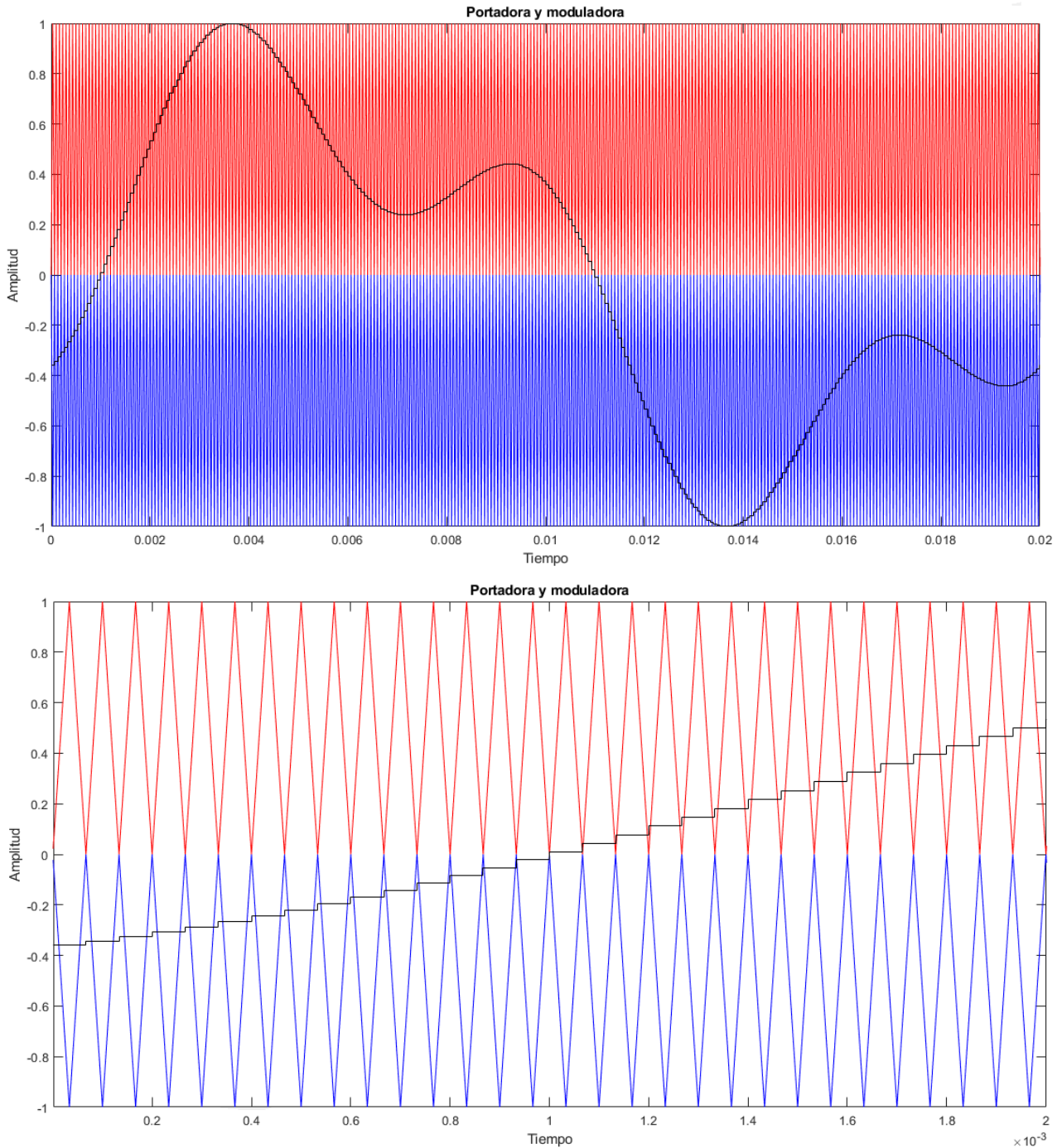


Fig. 45: Comparativa entre una señal moduladora y las portadoras.



### 2.2.11. Generación del punto muerto

Dado que las muestras de  $e[n]$  se encontraron sincronizadas con el periodo de la señal portadora  $p[n]$ , facilitamos la generación de los tiempos muertos. El mismo lo obtuvimos comparando la distancia entre el valor de  $p[n]$  y la muestra actual de  $e[n]$ . Definimos un intervalo donde el centro fue el valor de la muestra y el radio la distancia máxima entre la muestra y el valor de la portadora por el cual activamos la generación del tiempo muerto. Dado que las señales de control PWM son digitales, la generación de punto muerto consistió en una señal de control normalmente en 1, mientras que se activa poniéndose en 0. Entonces, si se hace una operación lógica AND con las señales PWM al activarse el tiempo muerto las señales se ponen en 0 en ese lapso, dándole tiempo a los IGBT de conmutarse sin entrar en cortocircuito.

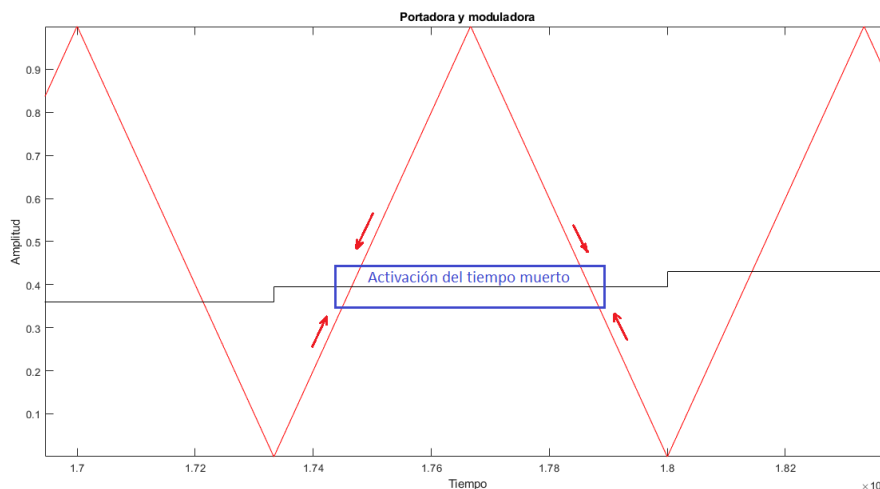


Fig. 46: Generación del Punto Muerto.



### 2.2.12. Simulación del Algoritmo PWM y Tiempo Muerto

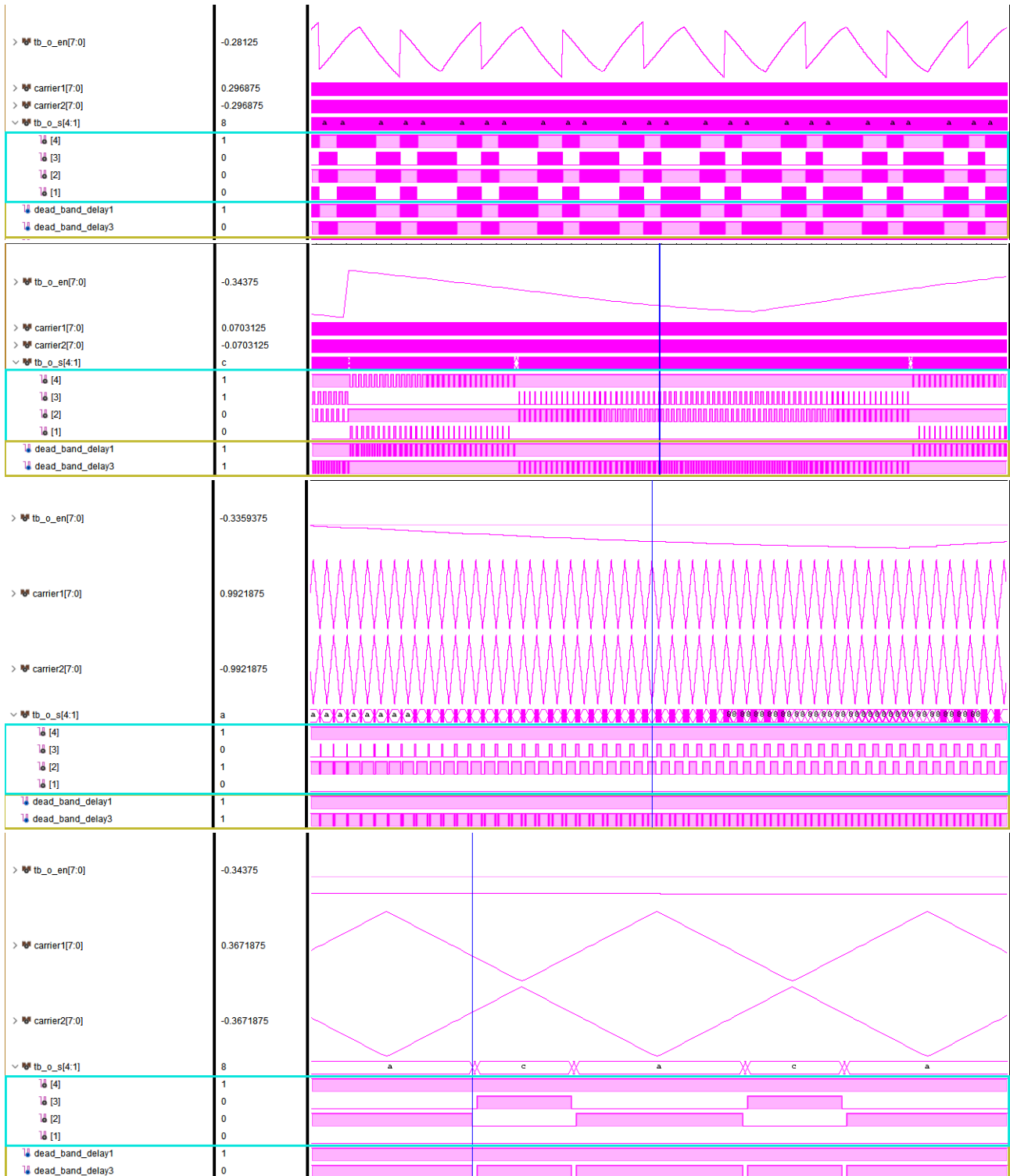


Fig. 47: Simulación del Algoritmo PWM y el Tiempo Muerto.



### 2.2.13. Modulación PWM de los Armónicos

Para garantizar una correcta modulación de la señal armónica  $e[n]$ , establecimos un criterio basado en la relación entre la frecuencia de la portadora y la moduladora. Considerando que la literatura sugiere una razón de al menos 5 pulsos de la portadora por medio ciclo de la moduladora, determinamos que una portadora de  $15[KHz]$  debería modular correctamente armónicos de hasta  $1,5[KHz]$  (armónico 30). La otra razón para emplear esta frecuencia es que, al coincidir con la frecuencia del PWM, permite tener una frecuencia de conmutación relativamente alta y alejada de los armónicos de red, con el fin de filtrarlos adecuadamente con el filtro de salida del circuito inversor, garantizando una correcta implementación del esquema propuesto.

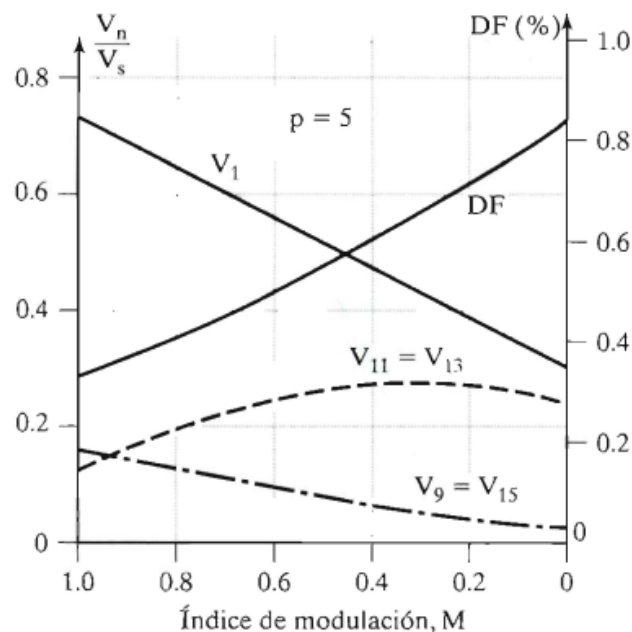


Fig. 48: Relación entre la Frecuencia y la Modulación[11].



### 2.3. Etapa de Generación de Armónicos:

Como se mencionó anteriormente, la FPGA es la encargada de generar las señales PWM para controlar un inversor, el cual nos da la corriente de armónicos necesaria. Para mantener nuestro objetivo de realizar un prototipo escalable en cuanto a potencia y tensión lo que hicimos fue diseñar un inversor en puente completo basado en dos medios puentes, de esta forma se puede replicar este medio puente una vez más para tener un inversor trifásico tal cuál se explica en el Capítulo 15 de la Sección II de [11]. Por otro lado, en cuanto a potencia este circuito está preparado para poder entregar hasta 500W debido a los componentes empleados, además, de acuerdo con [22] este inversor se puede acoplar a un transformador elevador de tensión y así poder trabajar con la tensión de red e inyectar la corriente a la misma.

En la Fig. 49 se puede observar el diagrama esquemático del circuito del inversor, el cual está basado en dos controladores o drivers IR2110 y cuatro IGBT modelos IKD10N60RC2. Seleccionamos este driver debido a la familiaridad que tenemos con el mismo por su frecuente uso en la cátedra de Electrónica de Potencia, además según [23] este driver nos permite controlar tensiones de hasta 600[V] con una lógica compatible con 3.3[V], así podemos controlarlo sin inconvenientes con la FPGA y tenemos un circuito escalable a usar con más tensión en un futuro. Por otro lado, los IGBT según [24] toleran hasta 600[V] entre colector y emisor, pudiendo también tolerar una corriente de colector máxima de 10[A] (8.9[A] para el diodo de marcha libre en paralelo), esto hace que el inversor sea escalable usando exactamente este mismo diseño.

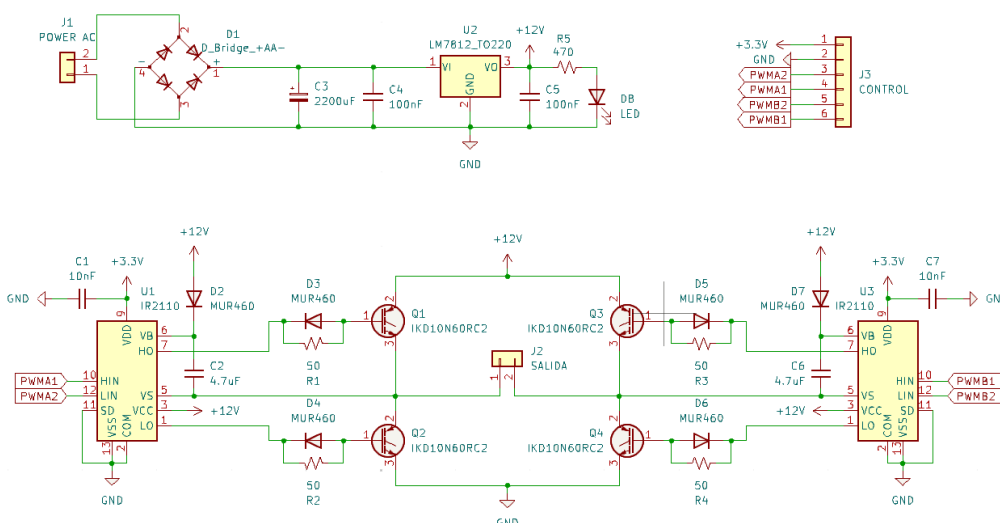



Fig. 49: Circuito de Generación de Armónicos.

 <b>UTN Regional Paraná</b>	<b>Filtro Armónico Activo</b>
<b>Ingeniería en Electrónica</b>	<b>2.Desarrollo</b>

Lo primero que encontramos en el circuito de la Fig. 49 es un regulador de tensión de 12[V], ya que esta será la tensión que empleará el inversor. Debido a que no realizaremos la inyección de corriente a la red, no es necesario emplear una alta tensión de continua. Luego del puente rectificador tenemos un capacitor para disminuir el ripple de la tensión y, posteriormente, tenemos un LM7812 con la configuración recomendada por el fabricante [25], además contamos con un diodo LED para así saber cuánto está encendido el inversor. Esta sección donde se establece la tensión continua de trabajo se conoce comúnmente como el *Bus DC*, este circuito puede funcionar sin inconvenientes con un Bus DC de 311[V] para así generar una tensión alterna de salida compatible para su inyección a la red.

Pasando ahora a los drivers, colocamos un capacitor de 10[nF] cercano a la alimentación de cada uno de los IR2110 de acuerdo con lo descrito por el fabricante [23] y luego diseñamos la red de bootstrap siguiendo los pasos establecidos por International Rectifier en [26] ya que los drivers IR2110 son de la misma empresa. De acuerdo con los cálculos basado en [26] nuestro capacitor de bootstrap será de 4.7[uF], mientras que cumpliendo con [26] el diodo necesario será un MUR460 debido a que se necesita que el tiempo de recuperación inverso sea menor a 100nS y el del diodo MUR460 es de 50nS según el fabricante [27].

Por último, para el gate del IGBT decidimos seguir la recomendación del fabricante [24] y emplear una resistencia de 50 [ $\Omega$ ] a la cual le colocamos en antiparalelo un diodo MUR460, de acuerdo con [28], para disminuir el tiempo de conmutación a la desconexión.

En cuanto a las señales de control decidimos emplear cuatro señales, una para cada IGBT para así poder tener más control sobre los tiempos muertos de las señales. Estos tiempos muertos son fundamentales para evitar cualquier posible cortocircuito, ya que al asegurarnos de que están en corte los IGBT antes de pasar de un estado al otro sabemos así que no hay posibilidad de un IGBT del lado alto de un medio puente se ponga en conducción antes de que el IGBT del lado bajo del mismo medio puente se desconecte.

Debido a que los IGBT IKD10N60RC2 son de montaje superficial decidimos implementar en una PCB de prototipo el circuito del inversor para así poder realizar todas las pruebas necesarias y corroborar el correcto funcionamiento del circuito antes de su etapa final ya integrado con la etapa de sensado y filtrado.



Para realizar las primeras pruebas y corroborar el correcto funcionamiento del inversor empleamos un simple programa realizado en Arduino para el microcontrolador ESP32 ya que lo único que queremos implementar es la generación de señales PWM con tiempo muerto. El primer programa de pruebas se basa en recorrer un vector de 200 posiciones el cual almacena un periodo completo de una senoidal de 50[Hz], la lectura de los valores del vector se realiza a una frecuencia de 15[KHz] y cada vez que se lee el dato (un valor entre 0 y 255) se ajusta el ciclo de trabajo de forma proporcional. Además de la modulación PWM de la senoidal de 50[Hz], le añadimos a cada dato un tiempo muerto de 1.30[ $\mu$ S] como se puede observar en la Fig. 50.

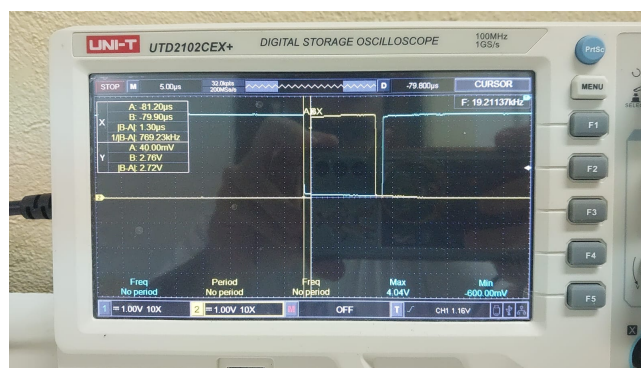


Fig. 50: Tiempo Muerto para las Señales PWM.

Para saber si el ESP32 estaba modulando correctamente la señal senoidal de 50[Hz], lo que hicimos fue colocar esta señal como entrada a un filtro pasabajos RC y así logramos observar que efectivamente tenemos una señal senoidal modulada en esta señal PWM, tal como se muestra en la Fig. 51.

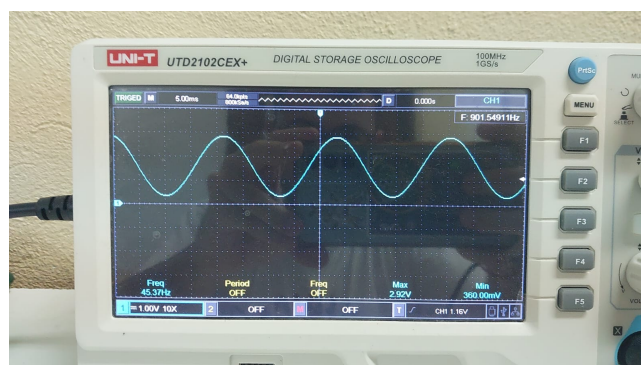


Fig. 51: Señal PWM Filtrada.

Si bien se ve que la frecuencia no es de 50[Hz], esto se debe a la poca precisión del ESP32 ya que se usaron comandos de alto nivel para hacer más sencillo el programa pero esto baja mucho la precisión en cuanto a controles de tiempo se refiere.



Ahora que sabemos que las señales de control PWM modulan correctamente esta señal senoidal de pruebas, lo que hicimos fue emplear el inversor con estas señales y medir la tensión de salida del mismo. Al igual que como hicimos para observar la senoidal de la señal PWM de control, a la salida del inversor conectamos el mismo filtro pasabajos RC conformado por una resistencia serie de  $220[K\Omega]$  y un capacitor en derivación de  $10[nF]$  de esta forma aunque el filtro es de primer orden, teniendo una frecuencia de corte de  $72.34[Hz]$  la atenuación de las componentes de alta frecuencia es suficiente. Es así como pudimos observar efectivamente que teníamos la misma señal senoidal a la salida del inversor que la que veíamos al analizar la señal PWM del controlador (Fig. 52), aunque esta vez tenemos un pequeño offset de aproximadamente  $500[mV]$ .

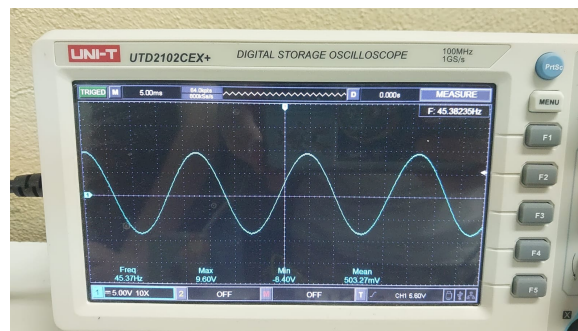


Fig. 52: Salida Senoidal Filtrada del Inversor.

Sabiendo que el inversor funciona, la siguiente prueba es ver si es capaz de replicar una señal con armónicos. Para ello generamos en GNU Octave un período de una señal con componentes de  $50[Hz]$ ,  $150[Hz]$  y  $250[Hz]$ , como debemos tener una señal que varíe desde 0 hasta 255, sin llegar a estos extremos (para tener un índice de modulación menor a 1), la señal se puede expresar como:

$$x(t) = 127 + 100 \cdot \sin(2\pi ft) + 50 \cdot \sin(6\pi ft) + 25 \cdot \sin(10\pi ft)$$

Siendo  $f = 50[Hz]$ , un período de esta señal representado por un vector de 200 valores se ve en la Fig. 53.

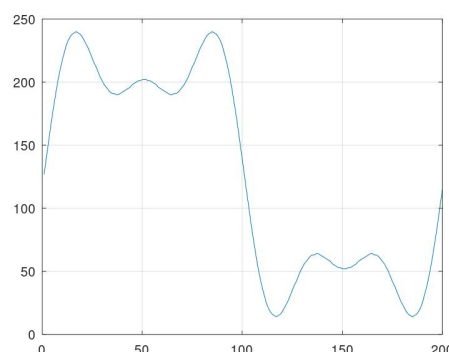


Fig. 53: Señal de Prueba con Armónicos.



Nuevamente, empleamos estos datos en el programa del microcontrolador ESP32 y generamos así las señales PWM de control para el inversor. Esta vez, para ver como funciona el inversor entregando un poco más de potencia lo que hicimos fue implementar a la salida del circuito un filtro pasabajos RLC compuesto por un inductor de 10[mH] en serie conectado a un paralelo en derivación de un capacitor de 1[uF] y dos resistencias las cuales equivalen a 27[Ω], de esta forma tenemos una potencia de salida de 3W aproximadamente.

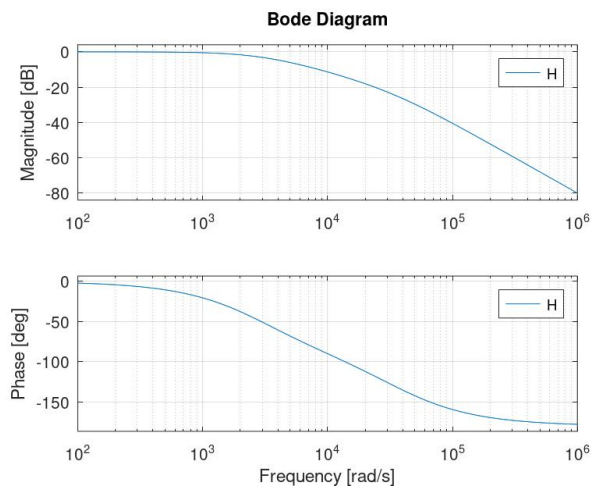


Fig. 54: Respuesta en Frecuencia del Filtro RLC.

Como se puede observar en la Fig. 54 la frecuencia de corte del filtro es de 1591.55[Hz], por lo que no alterará los armónicos que queremos generar pero si eliminará la alta frecuencia de la modulación PWM. Empleando este filtro a la salida del inversor, colocando una punta del osciloscopio en la resistencia del RLC y la masa al otro extremo de la resistencia, la tensión que medimos es la mitad de la real aplicada a la carga como se ve en Fig. 55, pero nos facilita la medición de los parámetros de frecuencia y también nos posibilita realizar la FFT.

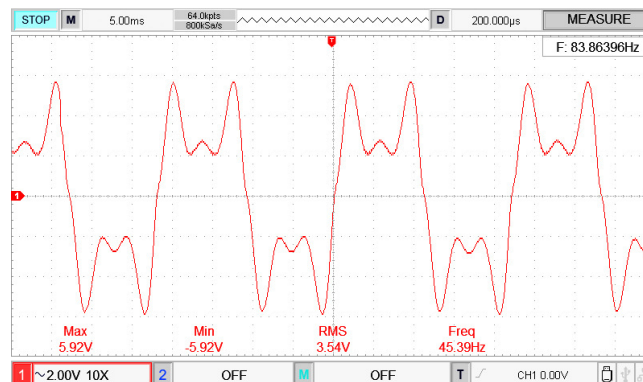


Fig. 55: Tensión de Salida del Filtro RLC.



Si vemos la Fig. 55 comparada con la Fig. 53, son muy similares pero es la FFT aplicada a la tensión la que nos brinda más información sobre la señal. Al observar la Fig. 56 notamos que no se sigue exactamente la relación de amplitudes que establecimos en un principio pero si es muy cercana. Debido a que para ver la señal completa se debe colocar cada punta del osciloscopio (referenciadas a masa) a cada salida del inversor y ver la suma de estas señales, pero hacer esto no nos permite realizar la FFT, por lo que solo medimos una salida del inversor. Sin embargo, podemos decir que la generación de armónicos por parte del inversor funciona sin inconvenientes.

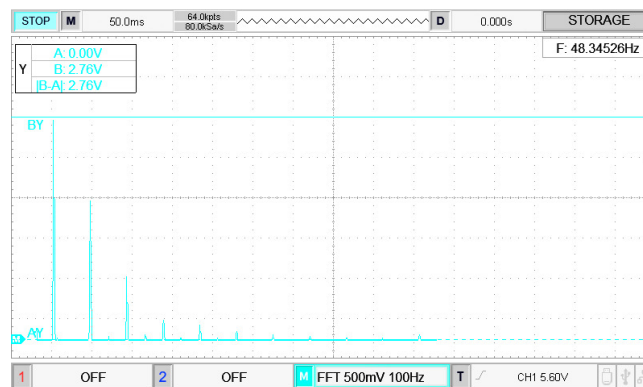


Fig. 56: FFT de la Tensión de Salida del Filtro RLC.

A pesar de esta pequeña diferencia con la señal ideal, culminamos las pruebas de funcionamiento del inversor considerando que no presenta inconveniente alguno y es capaz de generar las señales de armónicos que necesitamos. Para corroborar esto hemos probado nuestra etapa de sensado midiendo la corriente de salida del inversor con el filtro RLC conectado, de esta forma pudimos ver que efectivamente tenemos la misma forma de señal de corriente que de tensión, es decir, una suma de los armónicos 1ro, 3ro y 5to.

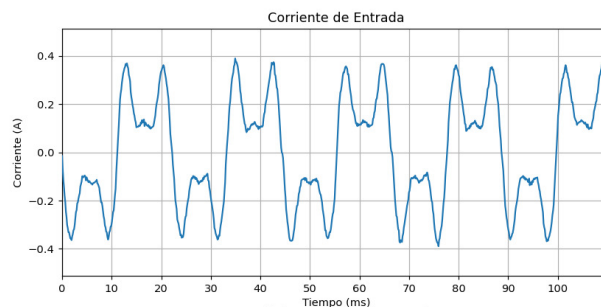


Fig. 57: Primera Medición de Corriente del Inversor con la App.



En la Fig. 57 se puede ver que la corriente se correlaciona con la señal de tensión medida con el osciloscopio previamente, aunque se ve también que tenemos un error a la hora de graficar el eje temporal, esto se solucionó para la prueba final. Solucionado el inconveniente de la escala temporal en la aplicación de visualización, volvimos a repetir la prueba y obtuvimos la siguiente medición.

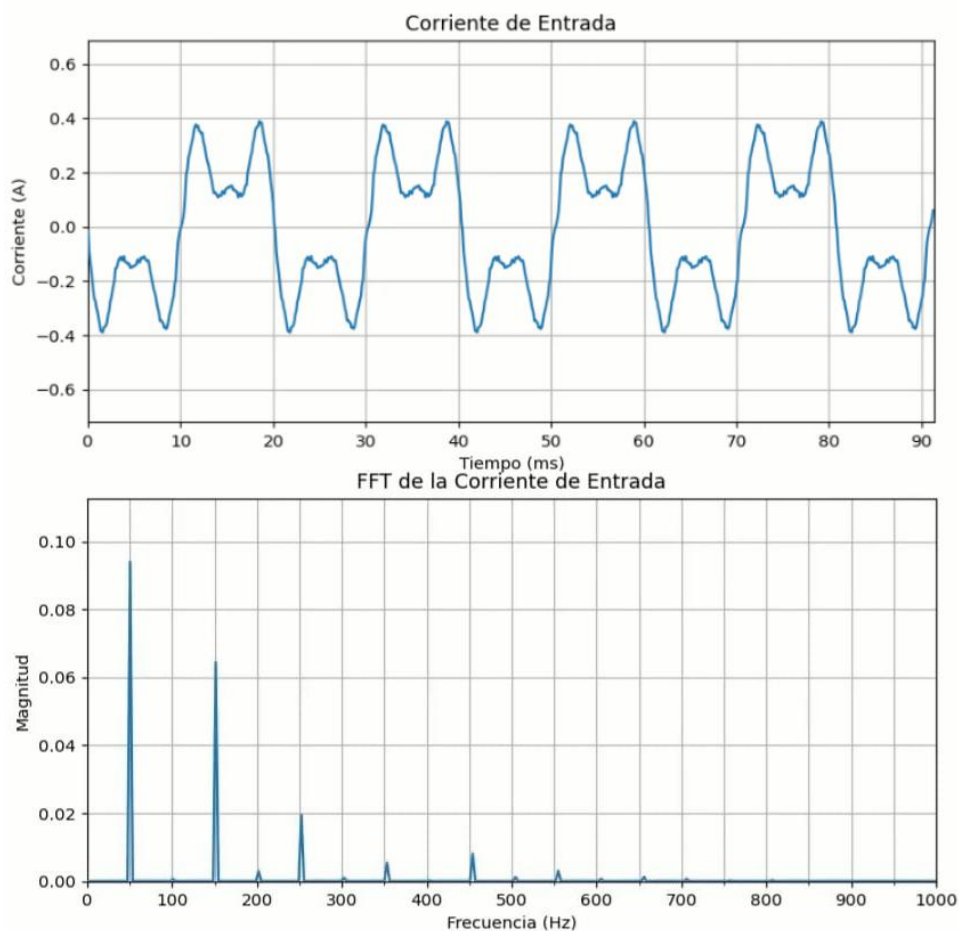


Fig. 58: Medición Final de Corriente del Inversor con la App.

En la Fig. 58 se ve claramente como la señal de corriente y su FFT es proporcional a la del osciloscopio ya que la impedancia del filtro RLC para estas frecuencias no es más que la resistencia, de este modo podemos hacer una comparación entre la corriente RMS que medimos con la aplicación y la corriente medida indirectamente con el osciloscopio, además, el THD de la corriente debería ser igual al de la tensión. Por un lado, observando las mediciones realizadas con el osciloscopio vemos que tenemos un consumo de 260[mA] RMS y un THD del 70 % aproximadamente, mientras que nuestra aplicación nos arrojó un consumo de 240[mA] RMS y un THD del 72.5 % por lo que consideramos que nuestro inversor está funcionando correctamente.



Llegados a este punto decidimos hacer una prueba midiendo esta misma corriente pero solamente usando la resistencia como carga, considerando que el filtro de la etapa de sensado podría llegar a ser suficiente para eliminar la alta frecuencia. Al realizar la prueba correspondiente vimos los mismos resultados usando solo la resistencia que usando el filtro RLC, por lo que para mayor comodidad de ahora en adelante usaremos solamente la resistencia como carga del inversor.

Una vez ya probado el circuito de la Fig. 49, generamos sus correspondientes archivos gerbers mediante la aplicación KiCad y los enviamos para la fabricación de la placa de circuito impreso del inversor. Una vez obtuvimos la placa, soldamos todos los componentes y colocamos un disipador al regulador de tensión lineal LM7812 ya que aunque deberá proveer menos corriente de su máximo posible, esta corriente multiplicada por la diferencia de tensión entre sus pines de entrada y de salida genera que el regulador tenga una disipación de potencia la cuál puede afectar el rendimiento del mismo en tiempos de prueba prolongados, es por ello que decidimos colocarle un disipador con su respectiva pasta de conducción térmica.

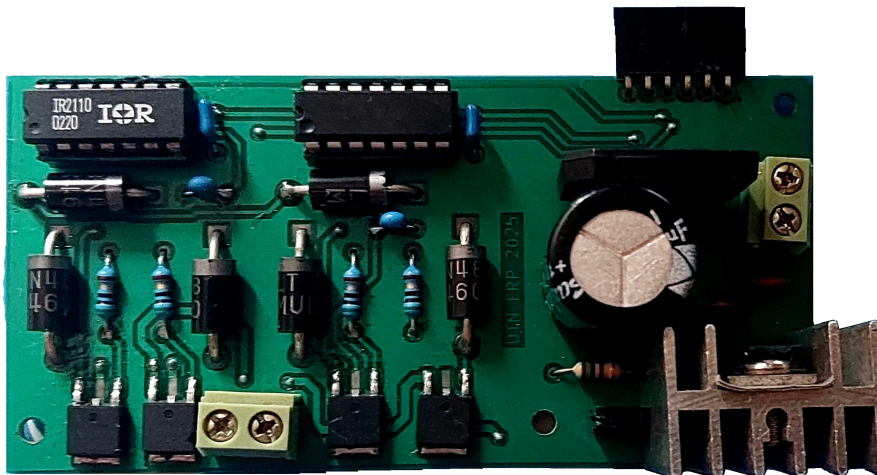


Fig. 59: PCB de la Etapa de Generación de Armónicos.

Se puede observar en la Fig. 59 que a la izquierda del regulador de tensión se encuentran dos pines hembra, estos representan el LED que indicará que el inversor esta encendido, y se colocaron pines hembra para así poder colocar el LED en el gabinete y conectarlo mediante cables a la placa.



Teniendo las dos placas terminadas hicimos una primera prueba preliminar para ver el funcionamiento de todo el conjunto y observamos un sobrecalentamiento en el puente rectificador de la placa del inversor y esto se debe al uso de un mismo transformador para alimentar ambas placas. Para comprender mejor esto se presenta en la siguiente imagen ambos circuitos de alimentación, por un lado el de la placa de sensado y por otro el de la placa del generador de armónicos.

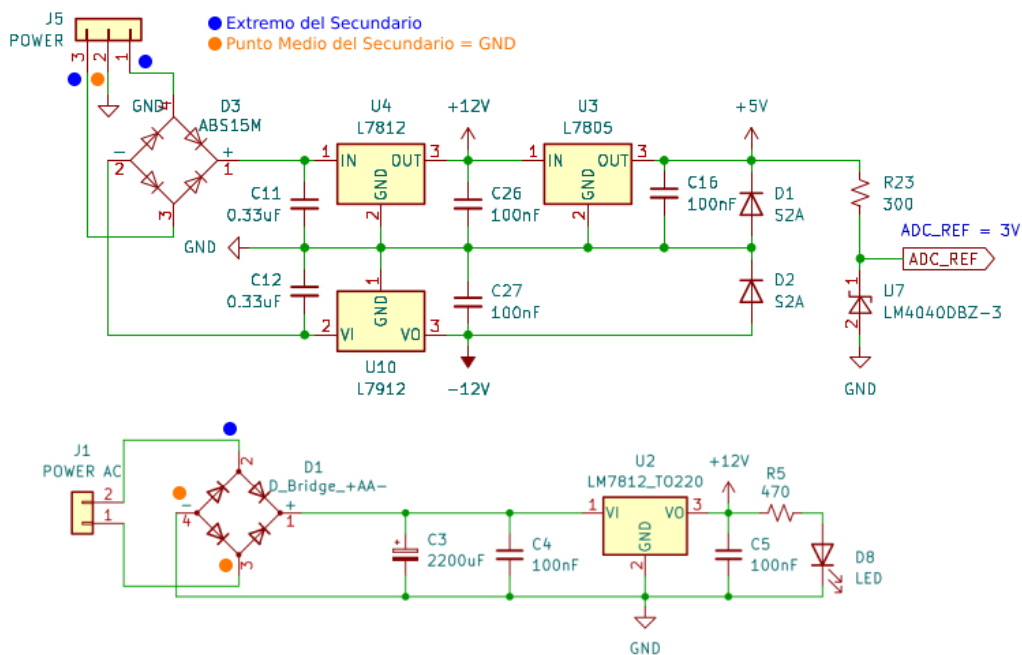


Fig. 60: Problema de Cortocircuito.

Como se puede observar en la Fig. 60 al ser el punto medio del transformador la referencia o GND del circuito de la etapa de sensado, en la etapa de generación de armónicos tenemos el punto medio en cortocircuito, durante cada medio ciclo, con un extremo del secundario del transformador. Para nosotros es imprescindible unir las masas ya que al enviar las señales PWM de control al IR2110 necesitamos que el driver tenga la misma referencia que las señales así se detectan correctamente los valores lógicos y como el IR2110 nos obliga a compartir la masa del sector de control con el sector de potencia es que ocurre este inconveniente que genera un cortocircuito entre punto medio y extremo del secundario del transformador a través de uno de los diodos del puente rectificador de la placa del inversor.



Para solucionar esto es que decidimos hacer una nueva placa que se colocará entre medio del inversor y de la placa de sensado. Esta nueva placa constará de cuatro optoacopladores 6N137, de modo que la información se transmita desde la FPGA a los drivers IR2110 manteniendo aislados ambos circuitos. Entonces, a los 6N137 entrarán las señales PWM de nivel lógico de 3.3[V] y tendremos a la salida las mismas señales pero invertidas y con un nivel lógico de 5[V], debido a que usaremos un 7805 para alimentar este circuito (haciendo uso de los 12[V] del inversor).

Así, el circuito a implementar se puede observar en la Fig. 61. Esta nueva placa irá conectada en una de las tiras de pines hembra que tenemos disponibles en la placa de sensado.

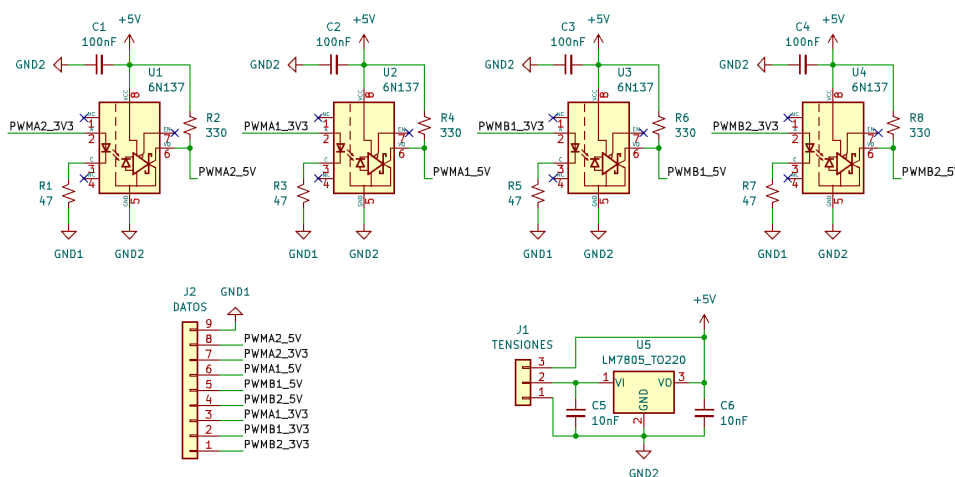


Fig. 61: Circuito de Adaptación de Señales PWM.

De esta forma mantenemos las masas de cada placa separadas y evitamos tener este inconveniente previamente descrito. Debemos mencionar que el hecho de usar los optoacopladores 6N137 y no otros es debido a su alta velocidad de respuesta[12] (tiempo de propagación de alto a bajo y viceversa, Fig. 62, de 50nS) lo que hace que no tengamos retrasos y deformaciones de las señales PWM. Además, la corriente necesaria por el LED del 6N137 para funcionar (mínimo 5[mA][12]) es adecuada para la capacidad de entrega de corriente de la FPGA[29].

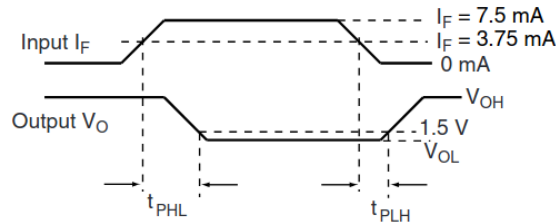


Fig. 62: Tiempos de Propagación 6N137[12]

Por otra parte, la inversión de las señales que provocan estos optoacopladores se solucionó fácilmente generando previamente una inversión de las señales en el código de la FPGA, de modo que al volver a invertirse en los 6N137 obtenemos las señales originales para los drivers IR2110.

Una vez diseñado el circuito en KiCad realizamos la placa de circuito impreso de manera casera como se ve en la Fig. 63 en este caso debido a la sencillez que presenta la misma. Esta placa se conecta en una tira de pines hembras que tiene la placa de sensado quedando de forma vertical sobre la misma, y así entonces en el armado final tendremos la placa del inversor conectada de forma horizontal a los pines machos de la placa de sensado y la placa de los optoacopladores conectada de forma vertical a los pines hembra.

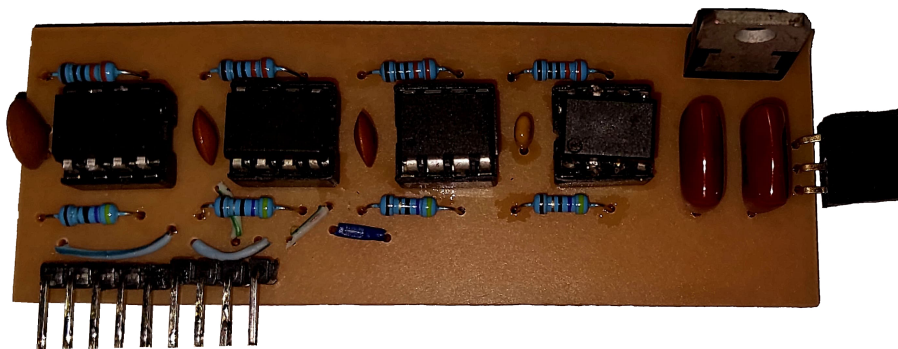


Fig. 63: PCB de los Optoacopladores.

Probando la placa se ven ahora las señales PWM de control con niveles lógicos de 5[V] y respetando los tiempos muertos dados como se ve puede observar en Fig. 64. Además se ve como al invertir la señal en la FPGA luego de los optoacopladores ya tenemos las señales originales sin ningún inconveniente.

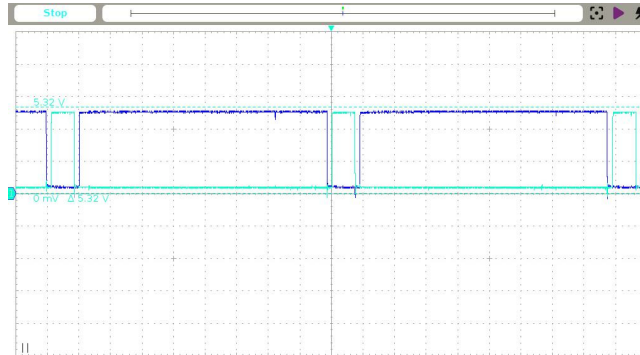


Fig. 64: Señales de Salida de los Optoacopladores.

Habiendo solucionado este inconveniente ahora la sección de potencia, o de generación de armónicos, completa consta del inversor de puente completo, los optoacopladores de adaptación y aislación, y finalmente la carga resistiva pura.

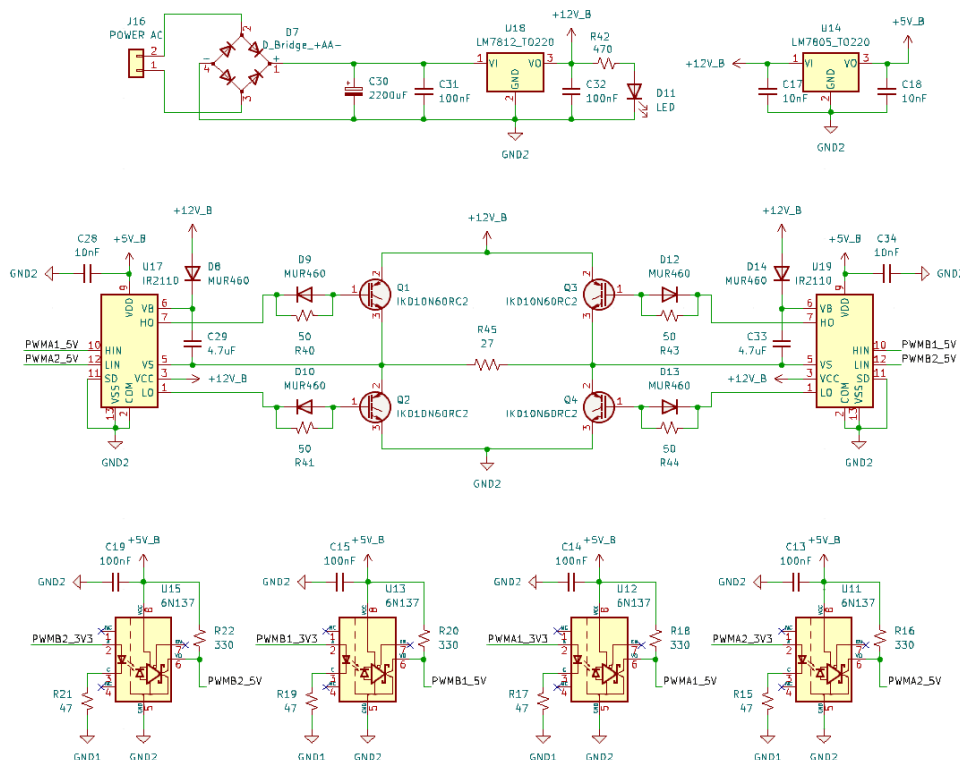


Fig. 65: Circuito Completo de la Etapa de Generación de Armónicos.



## 2.4. Modelado y Fabricación del Gabinete:

Una vez teníamos todas las placas terminadas comenzamos el diseño del gabinete empleando el programa gratuito de FreeCad. Nuestro gabinete contiene todas las placas y además el transformador con secundario de punto medio el cual alimenta tanto a la placa de sensado como al inversor generador de armónicos. Además, el gabinete permite el acceso a los potenciómetros de ajuste para eliminar el offset de las señales, a las entradas auxiliares de 3.5[mm] para los sensores y también al USB de la Raspberry Pi Pico.

Por último, el gabinete tiene dos conectores banana hembra para conectar cualquier tipo de carga al inversor, junto con un orificio para el LED indicador de estado del inversor. En cuanto a alimentación y encendido colocamos un conector IEC 320 C14 para proveer de 220[V] AC al transformador, esta alimentación pasa también por un fusible en serie de 1.5[A] (el cual puede accederse externamente), y luego tenemos tres interruptores independientes uno para encender el transformador, otro para encender la placa de sensado y un tercer interruptor para encender el inversor.



(a) Perspectiva Anterior.



(b) Perspectiva Posterior.

Fig. 66: [V]ista Exterior del Gabinete.



Como se puede observar en la Fig. 66 el gabinete consta de dos partes de igual tamaño que tienen 200[mm] de ancho, 200[mm] de largo y 50[mm] de alto, dando al final un gabinete de medidas exteriores de 200[mm] de ancho, 200[mm] de largo y 100[mm] de alto. El grosor de las paredes del gabinete es de 2[mm] para así darle la rigidez necesaria a la estructura sin encarecer demasiado la fabricación del mismo.

Ambas partes del gabinete están unidas mediante tornillos de 3[mm] de diámetro y 38[mm] de largo los cuales atraviesan los canales de unión internos que cuentan con una tuerca en su base, de esta manera se logra una mayor resistencia de la unión que al usar otros métodos como podrían ser encastres o una tapa que encaje a presión con la base. En cuanto a las placas, estas cuentan con separadores hexagonales de 6[mm] de diámetro y 20[mm] de largo, de este modo aseguramos cada placa al suelo del gabinete y lo dejamos a una altura un poco mayor para así tener cables más cortos a las conexiones externas (interruptores, salidas).

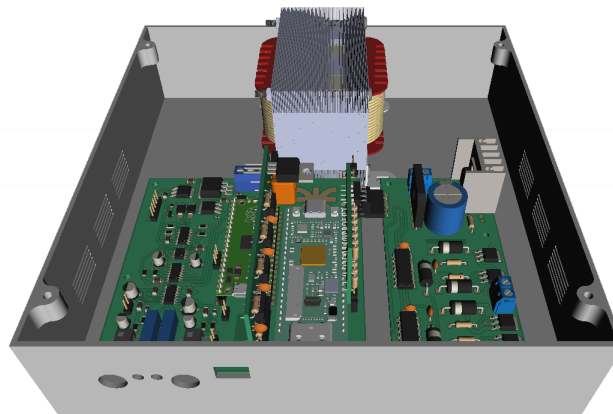



Fig. 67: Vista Interior del Gabinete.

Debemos mencionar que además de emplear el software libre de diseño 3D FreeCad, también hemos empleado el plugin KiCadStepUp para poder importar al modelo del gabinete cada uno de los PCB, esto facilitó mucho el diseño a la hora de dimensionar el espacio interno requerido y la ubicación de los conectores externos. Además, hemos empleado la librería pública de la comunidad de GrabCad de la cual hemos obtenido varios modelos 3D como el del transformador y de otros varios componentes electrónicos que no estaban originalmente en KiCad.

 <b>UTN Regional Paraná</b>	<b>Filtro Armónico Activo</b>
<b>Ingeniería en Electrónica</b>	<b>2.Desarrollo</b>

Finalmente, enviamos a imprimir el gabinete usando filamentos PLA blanco y PLA negro, este último para el logo de UTN ubicado en la parte superior. Además de la impresión del gabinete, decidimos diseñar e imprimir autoadhesivos para mejor la presentación del gabinete y también indicar donde se conecta cada cosa y como operar el instrumento en sí mismo.



(a) Perspectiva Anterior.

(b) Perspectiva Posterior.

Fig. 68: Vista Exterior del Gabinete.

Por último, para una mayor apreciación, dejamos una imagen de la vista frontal del gabinete donde se deben conectar los sensores de corriente, la carga del inversor y el cable USB para la visualización de la información en la computadora.

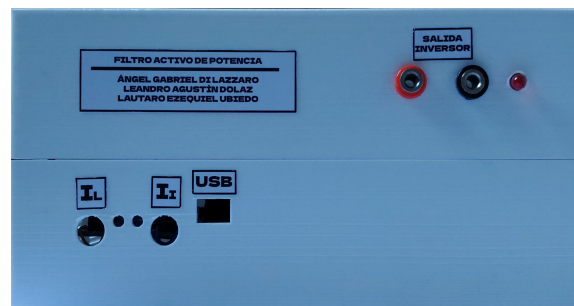



Fig. 69: Panel Frontal del Gabinete

 <b>UTN Regional Paraná</b>	<b>Filtro Armónico Activo</b>
<b>Ingeniería en Electrónica</b>	<b>2.Desarrollo</b>

## 2.5. Aplicación de Visualización:

Como venimos mostrando durante esta sección del informe, hemos desarrollado una aplicación para poder ver las señales del filtro. Esta aplicación esta desarrollada totalmente en el lenguaje Python y hemos usado el software Visual Studio Code para llevar a cabo la programación completa y todas las pruebas necesarias ya que nos permite poder hacer un debug completo en tiempo real viendo el estado de cada una de las variables implicadas de forma de poder simplificar la detección de errores.

La aplicación muestra las 3 señales que tenemos de interés, que son la corriente de carga, la corriente con los armónicos generados y la corriente resultante al aplicar el filtrado. A cada una de las señales además de mostrarlas en el dominio temporal se les aplica una transformada rápida de Fourier para mostrarlas también en el dominio de la frecuencia. Sumado a esto, la aplicación nos permite medir y mostrar la frecuencia, el valor RMS y el THDi de la corriente de carga además de la frecuencia y el THDi de la corriente resultante.

Además de mostrar toda esta información en tamaño de pantalla completa, la aplicación dispone de tres botones, uno para finalizar los procesos de la aplicación y cerrarla, otro para pausar la actualización automático del gráfico y otro para ajustar las escalas de los gráficos. El botón de pausado de la actualización automática es muy útil para poder hacer mediciones o mirar con mas detalle las señales sin la necesidad de mantener encendido el filtro ya que se puede apagar y la imagen permanecerá en la aplicación hasta que la misma se cierre. Por otro lado, el botón de re ajuste de las escalas sirve ya que el filtro funciona en tiempo real por lo que se puede variar la carga y el filtro continua funcionando pero puede suceder que la escala inicial seleccionada por la aplicación no sea la adecuada al haber variado la corriente por lo que tenemos la posibilidad de reajustarla para mayor comodidad.

El funcionamiento básico de la aplicación se basa en esperar al mensaje de comienzo de trama por el puerto serie, cuando se detecta ese mensaje el programa comienza a guardar los datos de cada señal en un vector, teniendo así al final tres vectores, uno para cada señal. Luego estos vectores se trabajan exactamente de la misma manera, primero generando el vector de tiempo conociendo la cantidad de muestras y el tiempo entre muestras (directamente relacionado con el baudrate, 115200[bps] y la frecuencia de muestreo, 500[Ksps]), luego se grafican estos vectores y también sus espectros de frecuencia. A partir de acá usamos, dependiendo del caso, el vector de datos temporales o de datos frecuenciales para realizar las mediciones de frecuencia, THDi y valor RMS.



Una muestra de la aplicación funcionando se puede ver a continuación en la Fig. 70

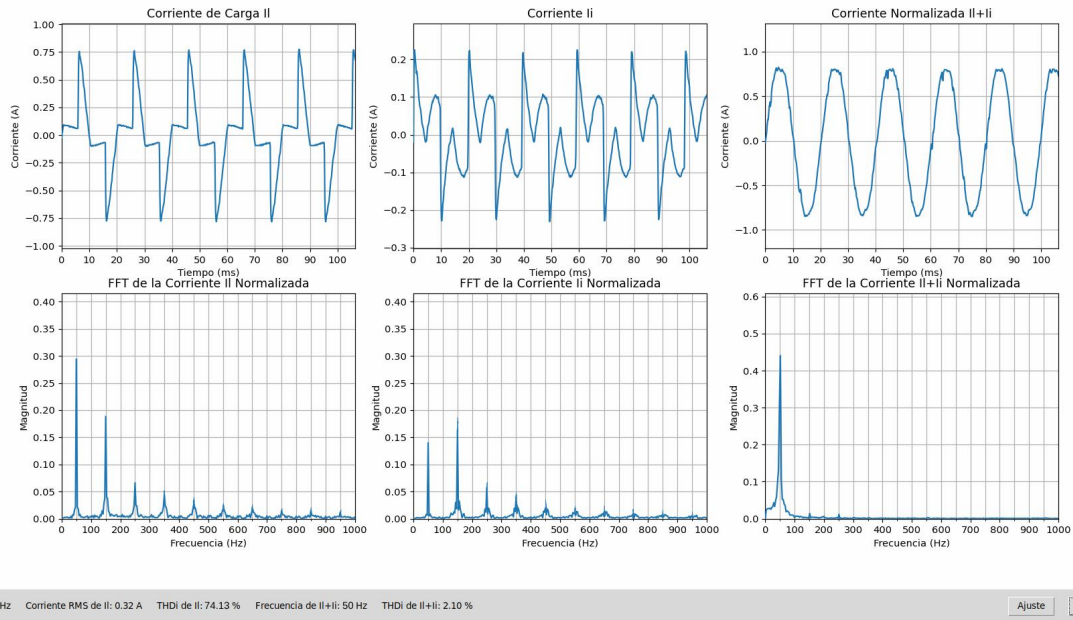



Fig. 70: Aplicación Final Desarrollada.

Quando realizamos la FFT a la señal obtenemos un vector con las magnitudes del espectro y un vector con las frecuencias, es con estos datos que calculamos cual es la frecuencia de la señal. El algoritmo implementado lo que hace es buscar en el vector de magnitud el valor máximo y obtener el índice del vector en ese valor, luego se busca en el vector de frecuencias este índice y el valor obtenido es la frecuencia de la señal.

Con una dinámica similar hacemos la medición del THDi de las señales. Como sabemos, cada armónico esta distanciado del siguiente por 100[Hz], esta distancia es proporcional en los índices del vector de la FFT de la señal por lo tanto lo que hacemos en la aplicación es buscar el índice correspondiente a los 50[Hz] y luego buscamos los múltiplos impares para así ubicarnos en 150[Hz], 250[Hz], etc. y así tenemos los valores de magnitud de cada uno de los armónicos y de la fundamental para poder calcular entonces el THDi de la siguiente manera:

$$THDi = \frac{\sqrt{\sum_{n=3}^{19} |I_n|^2}}{|I_{fundamental}|} \cdot 100 \% \quad (4)$$

 <b>UTN Regional Paraná</b>	<b>Filtro Armónico Activo</b>
<b>Ingeniería en Electrónica</b>	<b>2.Desarrollo</b>

Para la medición del valor RMS de cada corriente, lo que hacemos es usar los vectores de las señales en el dominio temporal ya que el usar la librería *numpy* nos permite elevar al cuadrado la señal completa, luego calculamos el valor medio de este vector y aplicamos la raíz cuadrada de forma que estamos calculando el valor RMS de la siguiente forma:

$$RMS = \sqrt{\frac{1}{N} \cdot \sum_{i=1}^N y^2(i)} \quad (5)$$

Finalmente, debemos también comentar que al conectar el Filtro a la computadora debemos ver el puerto serie de la Raspberry Pi Pico primeramente para saber que ganancia se aplicó a cada canal de los sensores de corriente, ya que estos datos son necesarios para poder mostrar los valores reales de las corrientes. Una vez sabido esto se puede ejecutar la aplicación de python y ahí el usuario ve una ventana emergente que pide el ingreso del nombre del puerto serie para conectarse al microcontrolador y luego también aparecen dos ventanas emergentes para ingresar la ganancia de cada canal.

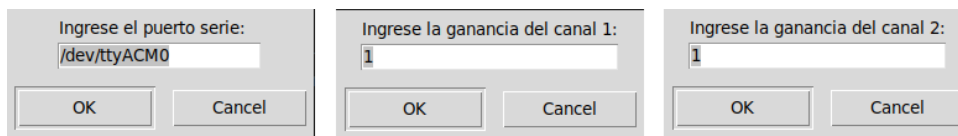


Fig. 71: Ventanas Emergentes de Configuración.

Una vez realizado esto, la aplicación inicia y hace por primera vez un ajuste de las escalas de los gráficos donde para los gráficos temporales lo que se hace es que el eje Y vaya desde un 30 % menos que el valor mínimo hasta un 30 % mayor que el valor máximo, mientras que el eje X está limitado a mostrar 800 muestras (en lugar de las 2400 totales) esto para poder ver de la mejor manera posible la señal. Por otro lado, en el dominio de la frecuencia el eje X se limita a los 1000[Hz] ya que es nuestro límite de funcionamiento, y el eje Y va desde 0 hasta un valor 30 % mayor al máximo detectado de las magnitudes de la FFT. Este ajuste, como se dijo anteriormente, se aplica automáticamente al inicio y luego puede el usuario volver a aplicarlo pulsando el botón *Ajuste*.



### Capítulo 3: Resultados

#### 3.1. Recursos utilizados en la FPGA

Se ha visto que la FPGA implementa diversos bloques de relativa complejidad, que incluyen gran cantidad de operaciones aritméticas. Los recursos de la FPGA presentes en la placa de desarrollo Tang Nano 9K se pueden encontrar en [29]. Entre los bloques que dispone se encuentran elementos de registros, multiplicadores embebidos, bloques de memoria RAM, etc. Desde la Fig. 72 a la Fig. 76 se realiza una comparación entre los recursos que emplea cada bloque implementado y posteriormente se comparan los recursos totales utilizados contra los disponibles por la FPGA, dando cuenta de la capacidad para implementar algoritmos de mayor complejidad.

Bloque	Logica		Registros	DSP (18x18)	BRAM (18Kb)	Comentarios
	LUT	ALU				
LSCLK1	18	0	10	0	0	Generador 15kHz
LSCLK2	4	0	2	0	0	Generador 900kHz
MAF1	178	6	106	2	1	Filtra 3er armónico
MAF2	122	5	64	2	1	Filtra 5to armónico
AF	945	64	2473	8	0	Filtro adaptativo
PWM	8	16	0	0	0	Genera PWM
Portadora	42	0	6	0	0	Señal triangular
Top level	37	0	0	1	0	Integra los bloques
<b>Total utilizada</b>	<b>1354</b>	<b>91</b>	<b>2661</b>	<b>13</b>	<b>2</b>	
<b>Total libre</b>	<b>7195</b>	<b>7195</b>	<b>3819</b>	<b>7</b>	<b>24</b>	

Fig. 72: Tabla de recursos utilizados por bloque implementado.

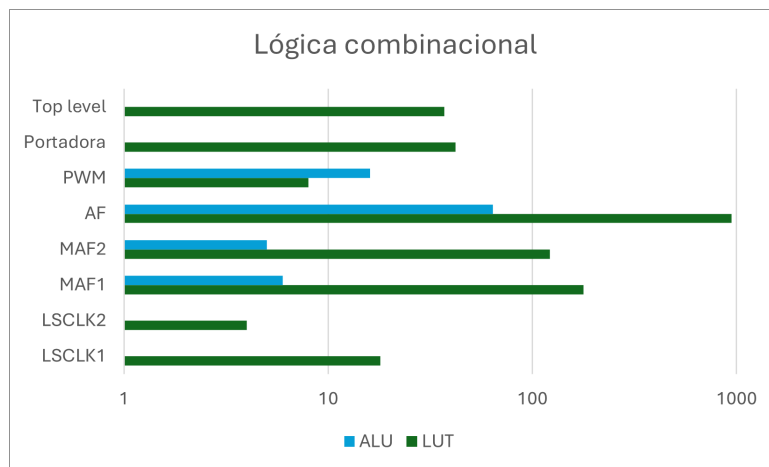


Fig. 73: Lógica combinacional utilizada.

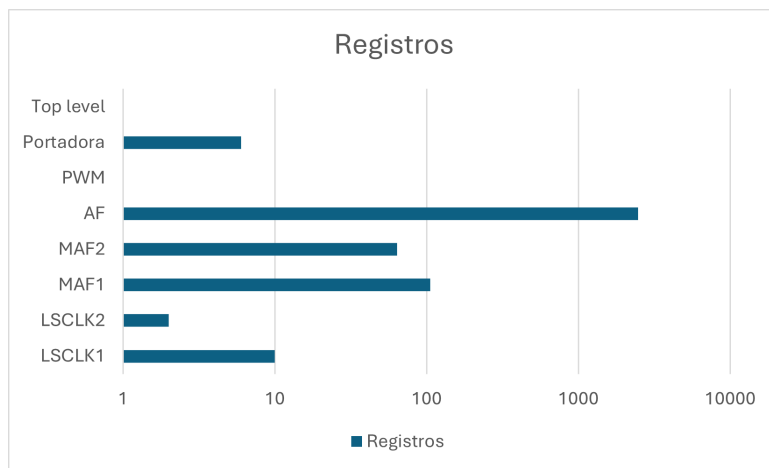


Fig. 74: Lógica secuencial utilizada.

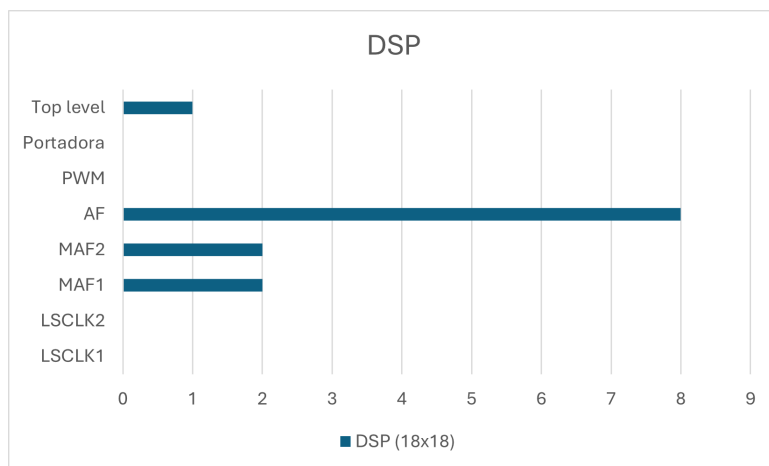


Fig. 75: Elementos DSP (multiplicadores).

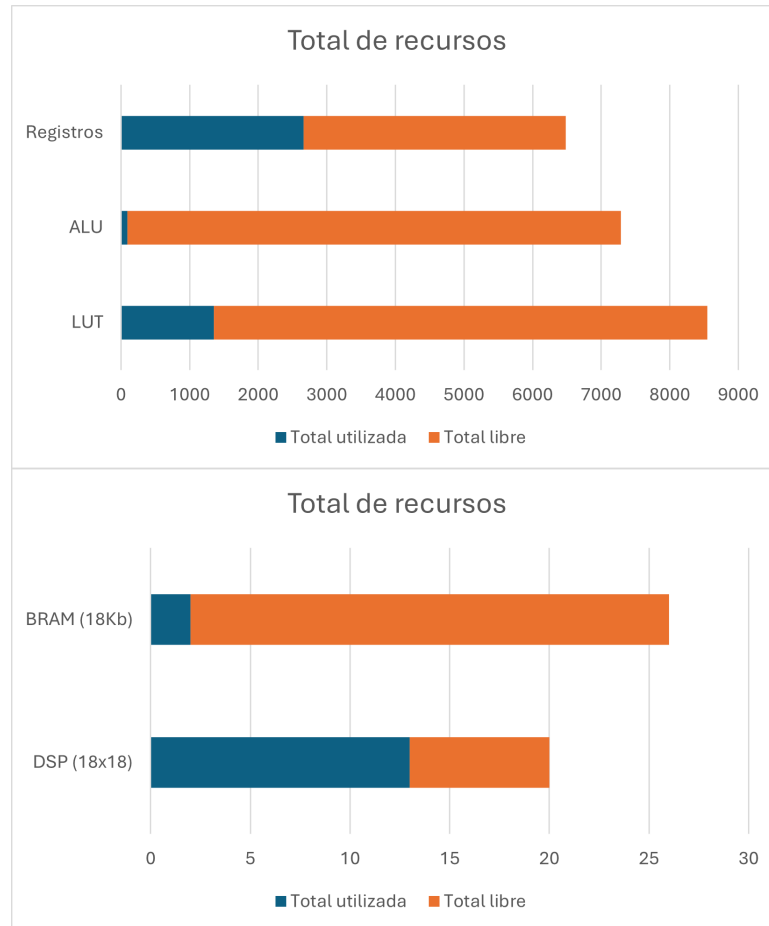



Fig. 76: Recursos usados contra libres.

### 3.2. Prestaciones del dispositivo

Para comenzar a hablar de las prestaciones de nuestro dispositivo como resultado de las pruebas de funcionamiento realizadas comentaremos primero cual es el rango de corrientes que podemos medir. El rango de corriente de carga que puede medir nuestro filtro armónico activo viene establecido por el modelo de sensor que se utilice. La familia de sensores de corriente SCT013 de la marca YHDC tiene modelos de: 5, 10, 15, 20, 30, 50, 60 y 100 amperes, cada uno de ellos provee como salida una tensión alterna de entre 0[V] a 1[V], por lo tanto nuestro diseño se puede emplear sin ninguna modificación con cualquiera de los sensores de corriente de esta familia, dependiendo así el rango de corriente del modelo escogido.

 <b>UTN Regional Paraná</b>	<b>Filtro Armónico Activo</b>
<b>Ingeniería en Electrónica</b>	<b>3.Resultados</b>

En nuestro caso empleamos un sensor de 10[A], por lo que podemos medir corrientes menores a 10[A] sin ningún inconveniente. De acuerdo con [15] midiendo corrientes de hasta 1.5[KHz] el sensor tiene un error de amplitud por debajo del 1 % mientras que tiene un error de fase de hasta 5 grados. Según lo establecido en [15] el sensor SCT-013 puede medir sin inconvenientes corrientes de hasta 2.5[KHz] pero es nuestro filtro el que limita el ancho de banda ya que posee una frecuencia de corte de 1.5[KHz]. El filtro implementado es de topología activa Sallen Key, siendo un filtro pasa bajos de segundo orden. Teniendo esto en cuenta, podemos medir corrientes hasta 10[A] con un error menor al 1 % y con un ancho de banda de 1.5[KHz]. Esto nos garantiza cumplir nuestro objetivo de poder medir, y por ende filtrar, hasta el vigésimo armónico (1[KHz]) sin ningún inconveniente.

Ahora, en cuanto a la potencia que puede ser capaz de proveer el inversor al generar las corrientes armónicas, esto depende principalmente de los transistores empleados en el diseño del puente completo. En nuestro caso estamos usando los IGBT IKD10N60RC2 de la marca Infineon los cuales soportan una tensión entre colector y emisor de hasta 600[V] y una corriente de colector de hasta 10[A], según [24], por ende si se usara una tensión DC de 311[V] (que sería la forma mas sencilla de obtener alta tensión DC, rectificando la tensión de línea) en el inversor el circuito sería capaz de proveer hasta 500[W] sin ningún problema ya que estamos hablando de una corriente 2.3A aproximadamente lo cual está muy por debajo del máximo permitido por el transistor, y además la tensión entre colector y emisor sería también menor al máximo permitido. Cabe aclarar, que esta tensión también la soporta el driver IR2110 ya que según [23] el canal flotante puede tolerar hasta 600[V].

En resumen, el diseño del inversor que hemos implementado es capaz de proporcionar una potencia de 500[W] si así se requiere. Además, cambiando el transistor como se explicó previamente en la sección de Desarrollo, se puede aumentar la potencia del inversor usando el mismo circuito y también se puede trasladar este circuito a un sistema trifásico de ser necesario ya que solo se requeriría implementar un medio puente más de igual forma que los dos medios puente ya implementados que conforman el inversor monofásico de puente completo.



Una vez establecidos los rangos de corriente de entrada para la etapa de sensado y de potencia del inversor generador de armónicos, lo que hicimos fue corroborar las mediciones de THD realizadas por nuestra aplicación, esto verificando la veracidad de nuestros datos de la FFT de la señal comparada con la FFT realizada por un osciloscopio, ya que son estos datos los que empleamos en el cálculo del THD (también comparamos el THD calculado por el osciloscopio con el nuestro). Como hemos mencionado antes en la sección de Desarrollo, la medición del THDi que realizamos en la aplicación se basa en el espectro frecuencial de la señal, es por ello que decidimos medir la corriente con nuestro dispositivo y también con la pinza amperométrica Fluke 80i-110s, que tiene un error del  $\pm 3\%$ [19], en conjunto con el osciloscopio Tektronic TDS 1002B, que tiene un error de voltaje de  $\pm 3\%$  y de tiempo de  $0,005\%$ [30], para así corroborar el error de nuestra medición al realizar la FFT.

Para esta prueba estamos usando como carga una lámpara incandescente de 60[W] junto con un dimmer para así generar los armónicos necesarios. Al conectar el osciloscopio a la computadora, usando su software dedicado, pudimos extraer todos los datos que necesitábamos, una representación gráfica se ve puede observar en la Fig. 77, en formato csv de forma de poder hacer una comparación detallada entre los datos obtenidos por medio del osciloscopio Tektronix junto con la pinza amperométrica Fluke y nuestro filtro de potencia activo en conjunto con su aplicación desarrollada en Python, cuya medición se puede ver en la Fig. 78.

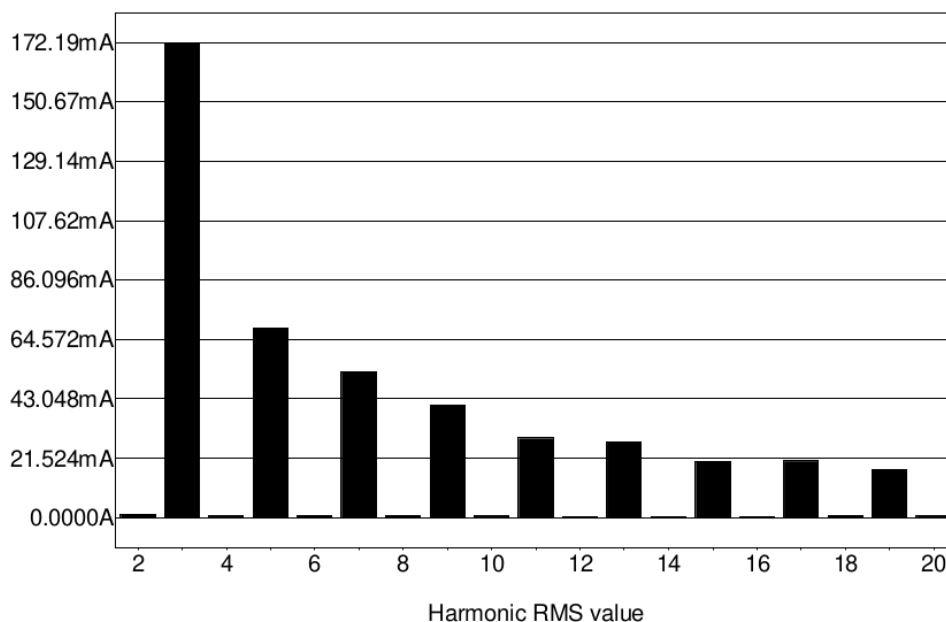


Fig. 77: Medición Realizada con Osciloscopio y Pinza Amperométrica.

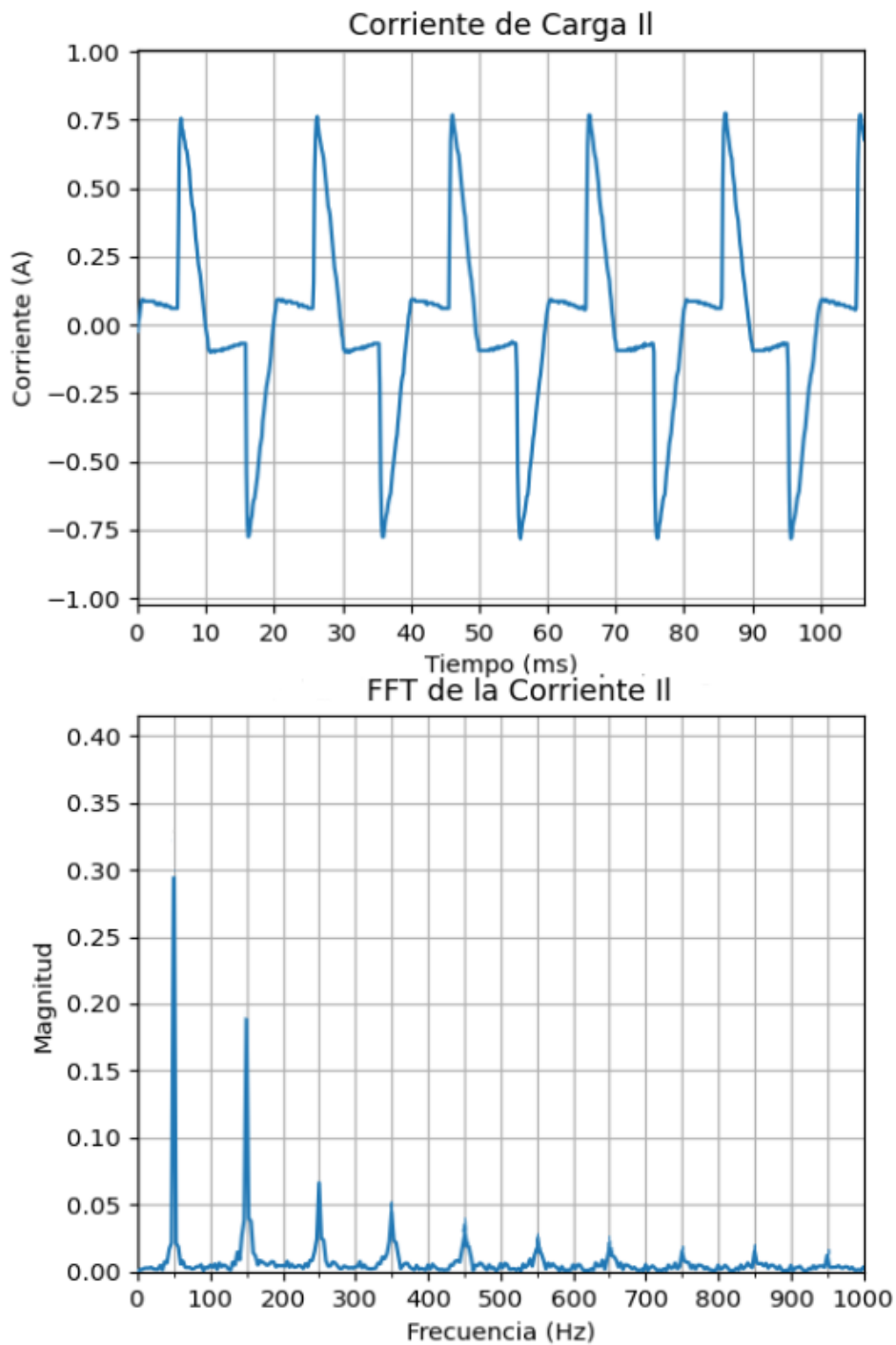


Fig. 78: Medición Realizada del Filtro Activo de Potencia con la App.



Si bien al comparar los gráficos se ve que las mediciones son muy similares, hemos usado la opción de debug del programa Visual Studio Code para así ver y guardar los datos de las magnitudes de la FFT de la corriente (datos en [mA]). Una vez que teníamos todos los datos lo que hicimos fue comparar los valores donde estaban los picos, siendo estas las magnitudes de cada armónico, y establecer el error que teníamos en cada armónico medido mediante la aplicación comparado con los valores arrojados por el osciloscopio, dando como resultado la Tabla 1.

Armónicos	App [mA]	Osciloscopio [mA]	Error (%)
3ro	180,62	172,19	4,90
5to	65,23	68,34	4,55
7mo	50,39	53,03	4,98
9no	38,95	40,83	4,60
11ro	27,63	29,06	4,92
13ro	26,31	27,73	5,12
15to	19,34	20,41	5,24
17mo	19,75	20,85	5,28
19no	16,61	17,53	5,25

Tabla 1: Comparación FFT de la Aplicación y del Osciloscopio.

Observando las mediciones realizadas vimos que el tercer armónico lo detectamos un poco mayor comparado con el osciloscopio pero luego los demás armónicos los detectamos menores, además se ve que a medida que aumenta el orden del armónico va incrementando el error. Creemos que el aumento del error a medida que aumenta la frecuencia del armónico se puede deber al filtro pasabajos cuya frecuencia de corte idealmente está ubicada en los 1.5[KHz] pero en la realidad podría ser menor y tal vez atenúe algunos armónicos mayores.

Por otro lado, si bien no se ve en esta comparación, la magnitud de la fundamental también resultó mayor en nuestra aplicación en comparación con el osciloscopio, por lo que creemos que este comportamiento de mostrar más magnitud en las bajas frecuencias se pueda deber al sensor SCT-013 ya que si bien su ancho de banda es mayor su uso no está focalizado en mediciones de armónicos. Teniendo esto en cuenta nuestras mediciones tienen un error del  $\pm 5\%$ , usando la pinza amperométrica Fluke en conjunto con el osciloscopio Tektronix como patrón.



Ahora bien, pasando a una prueba completa de funcionamiento, continuamos usando la misma lámpara incandescente de 60[W] alimentada a través del dimmer, y el resultado se puede apreciar en la Fig. 79 , donde a la izquierda se puede observar la corriente de la carga, al medio se ubica la corriente generada por el inversor, y a la derecha la corriente resultante de aplicar el filtrado. Cabe aclarar que la FFT de las corrientes de carga (II) y del inversor (Ii) se encuentran normalizadas de forma de que el usuario pueda observar que los armónicos en ambos gráficos deben ser del mismo tamaño para así tener un buen filtrado. Por otro lado, la señal de la derecha es generada directamente por la FPGA, ya que no realizamos una inyección de la corriente del inversor a la red la FPGA genera la señal de armónicos desfasados 180 grados y la suma a la corriente de carga obteniendo así la señal resultante que sale por medio de 8 bits de forma digital y luego se convierte a una señal analógica normalizada que es la que el usuario ve en la aplicación. Es el THD de esta tercera señal el que nos indicara la efectividad de nuestro filtro activo de potencia.

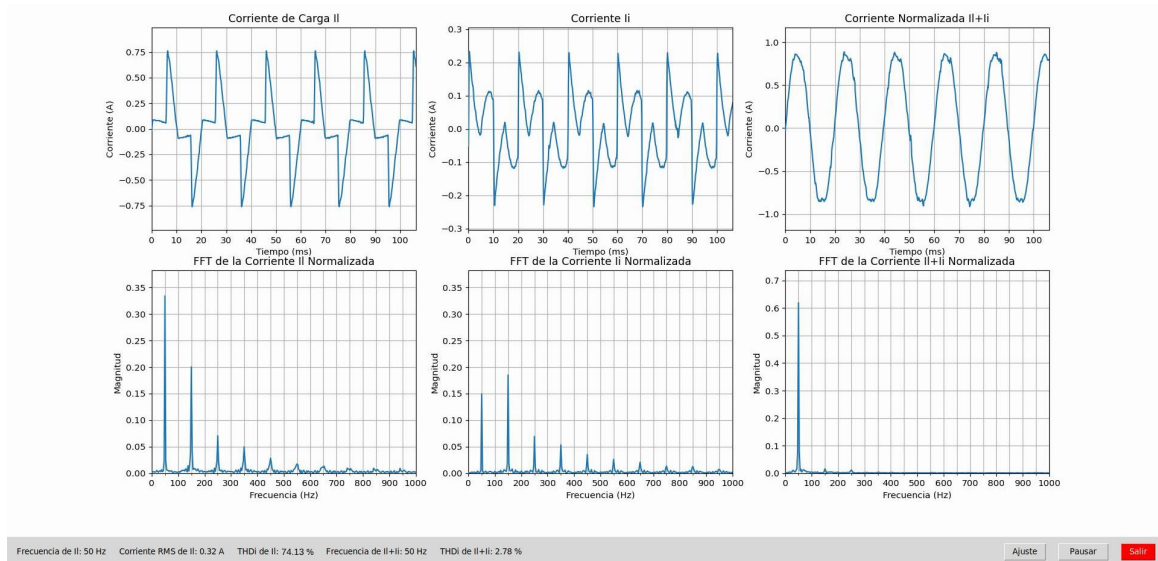


Fig. 79: Prueba de Funcionamiento con Lámpara de 60[W].



Debemos aclarar que aunque en la aplicación se vea que están en fase la corriente de carga con la del inversor esto es sólo para la visualización gráfica, la FPGA genera todos los armónicos desfasados 180 grados como se explicó anteriormente. Finalmente, en la sección inferior de la imagen se ven las mediciones realizadas, donde se puede apreciar la frecuencia fundamental de la corriente de carga y de la corriente resultante, siendo 50[Hz] en ambos casos, también se ve el valor RMS de la corriente de carga el cual al usar el debug vimos que fue de 324,76[mA] mientras que la medición con el osciloscopio nos arrojó un valor de 340,83[mA] por lo que nuestro error en este caso es menor al 5 %, y finalmente, tenemos el THDi de la corriente de carga y de la señal resultante, donde se ve que pasamos de tener un THDi del 74,13 % a uno del 2,78 %, logrando así un THD menor al 5 % mostrando que el filtrado es totalmente funcional.

Al realizar la medición del THDi de la corriente de cargar con la pinza Fluke en conjunto con el osciloscopio Tektronix obtuvimos un valor de 76,54 %, por lo que podemos considerar que es coherente nuestra aplicación y nuestro filtro activo de potencia está funcionando correctamente.

Después de corroborar el correcto funcionamiento del filtro, cambiamos la carga y empleamos una plancha 1000[W] haciendo que el ángulo de disparo del dimmer sea mayor para ver como reacciona el filtro a un mayor THDi de la corriente de carga. El resultado de esta prueba se puede observar en la Fig. 80.

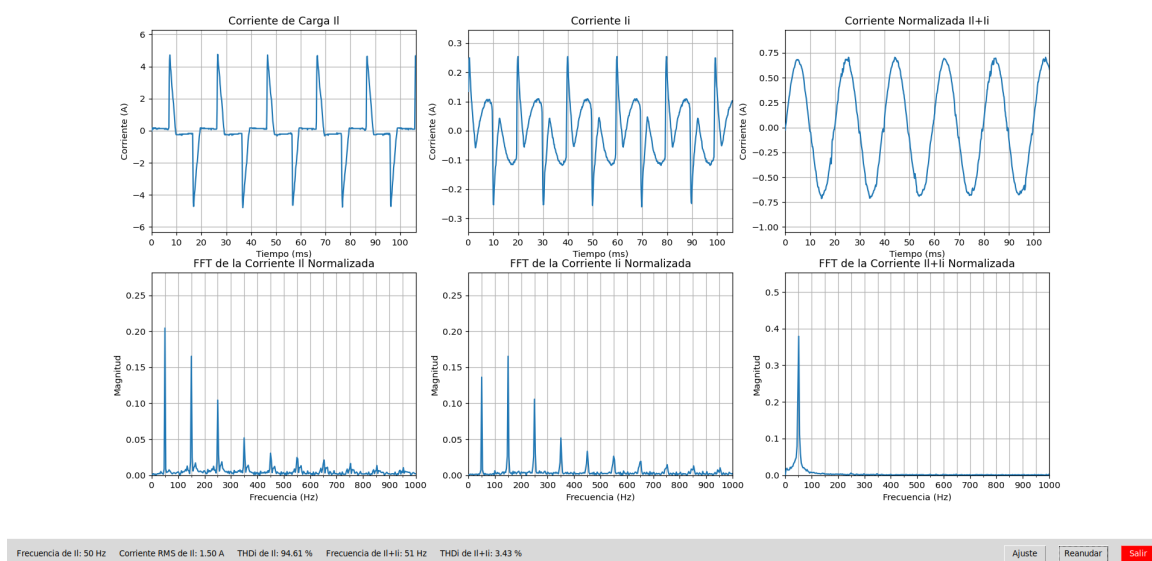


Fig. 80: Prueba de Funcionamiento con Plancha de 1000[W].



Como se puede apreciar, al disminuir el tiempo de conducción del dimmer la magnitud de los armónicos aumenta, disminuyendo también la magnitud de la fundamental y esto genera el elevado THDi que en este caso vale 94,61 % y al aplicar el filtrado activo obtenemos una señal senoidal cuyo THD es del 3,43 %, si bien se ve que es mayor con respecto a la prueba anterior, aún así estamos por debajo del 5 %.

Para mostrar como reacciona el filtro activo de potencia a un cambio de la corriente en tiempo real hemos aumentado el tiempo de conducción del dimmer y ahora obtuvimos un THDi de la corriente de carga de 49,16 %, mientras que la señal resultante al aplicar el filtrado tiene un THD de 3,08 % como se ve en la figura siguiente.

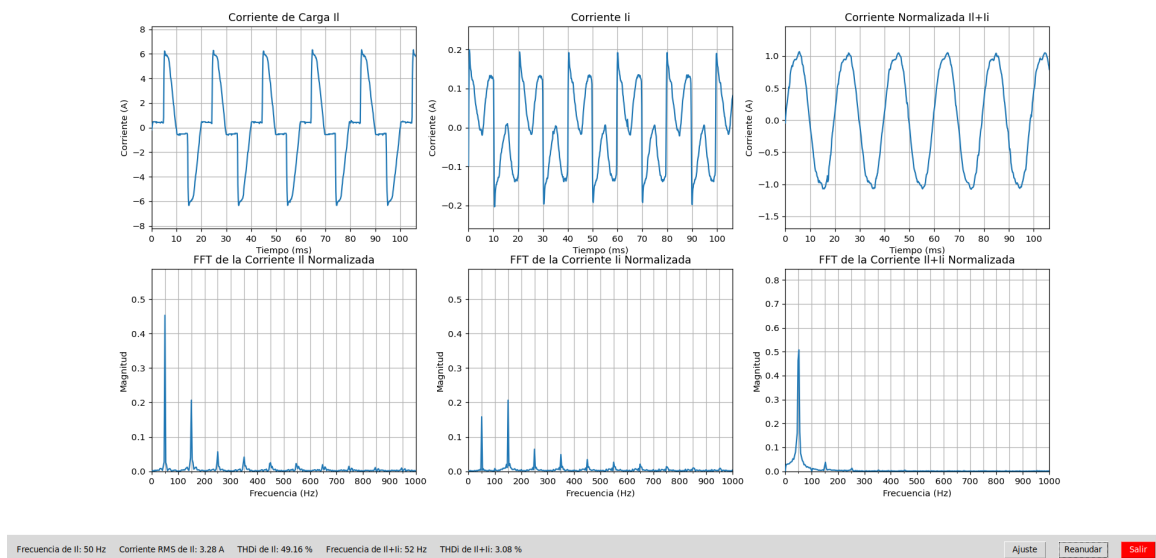


Fig. 81: Prueba de Funcionamiento con Plancha de 1000[W] y Más Conducción.

Terminando esta sección queremos presentar un resumen de las características de nuestro filtro activo de potencia implementado:

- Rango Corriente de Carga: 0-10[A].
- Rango de Armónicos: desde el 3ro hasta 20mo.
- Capacidad de Potencia del Inversor: 500[W].
- THDi Resultante: < 5 %.
- Error Promedio de las Mediciones:  $\pm 5$  %.
- Tensión de Alimentación: 220[V]/50[Hz]




## Capítulo 4: Análisis de Costos

Para una mejor organización hemos dividido los costos de nuestro proyecto en 3 partes. Por un lado tenemos los costos de componentes para la etapa de sensado, luego tenemos los costos de componentes del inversor y por último los costos de fabricación e insumos varios. Hemos decidido hacer esta separación para así poder plantear que costos aumentarían dependiendo de que mejora se requiera realizar, por ejemplo, si se desea aumentar la potencia solamente se deberá modificar el inversor. Dicho esto, comenzamos presentando en la Tabla 2 los costos de los componentes empleados en la etapa de sensado. Cabe recalcar que todos los precios se verán en dólares estadounidenses (U\$D).

Componente	Cantidad	Precio Unitario (U\$D)	Precio (U\$D)
Conector Jack 3,5[mm]	2	\$0,780	\$1,560
Capacitor 16[V] 10[uF]	10	\$0,078	\$0,780
Capacitor 25[V] 0,1[uF]	10	\$0,058	\$0,580
Capacitor 6,3[V] 100[uF]	2	\$0,160	\$0,320
Capacitor 250[V] 15[nF]	2	\$0,860	\$1,720
Capacitor 16[V] 6,8[nF]	2	\$0,360	\$0,720
Capacitor 25[V] 0,33[uF]	2	\$0,100	\$0,200
Resistor 250[mW] 7,68[K $\Omega$ ]	2	\$0,100	\$0,200
Resistor 250[mW] 14,3[K $\Omega$ ]	2	\$0,100	\$0,200
Resistor 250[mW] 10[K $\Omega$ ]	10	\$0,031	\$0,310
Resistor 250[mW] 820[K $\Omega$ ]	10	\$0,140	\$1,400
Resistor 250[mW] 390[K $\Omega$ ]	10	\$0,140	\$1,400
MC7812	1	\$0,530	\$0,530
MC7912	1	\$0,990	\$0,990
MC7805	1	\$0,270	\$0,270
Diodo S2A	2	\$0,070	\$0,140
LM4040	1	\$0,860	\$0,860
Resistor 250[mW] 330[ $\Omega$ ]	1	\$0,100	\$0,100
Puente Rectificador ABS15M	1	\$0,200	\$0,200
Ampl. Operacional TL072	1	\$0,230	\$0,230
Ampl. Operacional OP4134	1	\$5,810	\$5,810
Ampl. Operacional UA741	1	\$1,442	\$1,442
Flip-Flop Tipo D CD4013	1	\$0,330	\$0,330
Digipot MCP42050	1	\$2,820	\$2,820
Bornera de 3 Puntos	1	\$0,501	\$0,501
Tira de Pines Machos Rectos	1	\$0,926	\$0,926
Tira de Pines Machos 90°	1	\$0,849	\$0,849
Tira de Pines Hembra Rectos	4	\$1,333	\$5,332
Trimmer Potenciómetro 10[K $\Omega$ ]	2	\$0,700	\$1,400

Tabla 2: Costos Componentes de la Etapa de Sensado.

 <b>UTN Regional Paraná</b>	<b>Filtro Armónico Activo</b>
<b>Ingeniería en Electrónica</b>	<b>4.Análisis de Costos</b>

Si hacemos una suma de todos los componentes listados en la Tabla 2, agregándole el costo de los sensores SCT-013 (\$37,886) la Raspberry Pi Pico (\$8,13) y la FPGA Tang Nano 9K (\$13), vemos que tenemos un costo total de \$90,546 dólares estadounidenses en cuanto a componentes para la etapa de sensado se refiere. Si se quisiera adaptar esta etapa para un filtro activo trifásico pasaríamos de medir dos corrientes (carga e inversor) a medir seis corrientes, tres sensores para la corriente de carga y tres sensores para la corriente del inversor por lo cual los componentes desde el sensor YHDC SCT-013 hasta el digipot MCP42050, deberían multiplicarse por tres sus cantidades, además se deberá añadir un multiplexor para poder leer las seis señales haciendo uso del mismo microcontrolador ya que solo cuenta con tres canales ADC. De esta forma podemos muy fácilmente escalar nuestra etapa de sensado ya que la lectura de cada una de las corrientes se realiza de forma independiente y los componentes usados fueron seleccionados con el fin de facilitar este proceso ya que se podría copiar exactamente el circuito presentado tres veces o usar los componentes de la misma familia como el OPA134, OPA2134, TL074, MCP41050,[16][31][3] etc. pudiendo así mantener la calidad del sistema usando distintas combinaciones de componentes de la misma familia logrando el diseño más eficiente posible.

Debemos mencionar que al seguir usando las Raspberry Pi Pico RP2040 la tasa de muestreo del ADC seguirá siendo la misma (500[Ksps]) tanto si el sistema es trifásico como si fuera monofásico, por lo que la resolución será menor pero aún así será suficiente para el correcto funcionamiento del filtro.

Ahora veamos los costos de los componentes del inversor (Tabla 3), esta etapa es la que determina cuál será la potencia máxima que es capaz de manejar el filtro. Cabe destacar que esta etapa podría también ser subdividida en dos, por un lado los drivers y los IGBT y por otro lado la fuente de tensión DC. En este caso al tratarse de un dispositivo prototipo no estamos trabajando con alta tensión DC, por lo que para su implementación en un sistema real y funcional debe considerarse usar un rectificador de la tensión de red más un banco de capacitores con un arrancador suave para así tener una tensión estable de aproximadamente 311[V ]o rectificar un tensión menor usando un transformador (como es este caso) y luego emplear un elevador de tensión DC, en este caso la complejidad del circuito aumenta pero disminuye el tamaño final del sistema.



Componente	Cantidad	Precio Unitario (U\$D)	Precio (U\$D)
Zócalo DIP14	2	\$0,433	\$0,866
Driver IR2110	2	\$5,463	\$10,926
Optoacoplador 6N137	4	\$2,108	\$8,432
IGBT IKD10N60RC2	4	\$0,690	\$2,760
Diodo Rápido MUR460	6	\$0,450	\$2,700
Capacitor 25[V] 4.7[uF]	2	\$0,340	\$0,680
Capacitor 25[V] 10[nF]	2	\$0,183	\$0,366
Resistor 250[mW] 50[Ω]	4	\$0,100	\$0,400
Resistor 250[mW] 470[Ω]	1	\$0,210	\$0,210
Resistor 250[mW] 330[Ω]	5	\$0,140	\$0,700
Resistor 250[mW] 47[Ω]	5	\$0,140	\$0,700
LED 5[mm]	1	\$0,099	\$0,099
LM7812	1	\$1,626	\$1,626
LM7805	1	\$1,134	\$1,134
Capacitor 25[V] 100[nF]	6	\$0,245	\$1,470
Capacitor 25[V] 10[nF]	2	\$0,169	\$0,339
Capacitor 35[V] 2200[uF]	1	\$0,902	\$0,902
Puente Rectificador GBU4M	1	\$0,490	\$0,490
Bornera de 2 Puntos	2	\$0,811	\$1,622
Tira de Pines Hembra 90°	1	\$1,951	\$1,951
Tira de Pines Macho 90°	1	\$0,849	\$0,849

Tabla 3: Costos Componentes del Inversor.

Al igual que en los costos de la etapa de sensado, todos los precios están en dólares estadounidenses (U\$D). Si vemos la Tabla 3 podemos calcular un costo total para esta etapa de \$39,222 dólares estadounidenses. Este costo aumenta rápidamente si decidimos implementar una fuente de tensión DC de alrededor de 311[V] para poder realizar la inyección de corriente a la red, sumado también al costo del filtro necesario para conectar la salida del inversor a la red.

Como se mencionó anteriormente, este circuito inversor permite manejar una potencia de hasta 500[W], pudiendo aumentarla simplemente cambiando los modelos de IGBT empleados, considerando siempre que puedan ser controlados por el driver IR2110. Por ejemplo, ahora los IGBT permiten una corriente máxima de 10[A] pero si se los cambia por unos IGBT GT50JR21 de la marca Toshiba, se podrían manejar corrientes de hasta 50[A] [32] aumentando así la potencia máxima que es capaz de entregar el filtro. Ahora bien, el mayor costo del aumento de potencia viene del lado de la fuente de tensión DC, ya que tanto si se usa una simple rectificación de la tensión de red como si se usara un transformador con rectificación y con un circuito elevador de tensión DC, los costos de estas fuentes se elevarían notablemente.




Por otro lado, si se quisiera trasladar el circuito a un sistema trifásico, la modificación a realizar sería usar un medio puente más, teniendo así tres drivers IR2110 y seis IGBT para así configurarlos como un VSI trifásico. Obviamente, esta modificación vendrá de la mano con la modificación del software para generar ahora seis señales de control PWM, una para cada uno de los IGBT. Debe tenerse en cuenta que cualquier modificación, tanto en aumento de potencia como en cambio de sistema de monofásico a trifásico, además de implicar un aumento en los costos, incrementa bastante el tamaño físico del filtro completo.

Finalmente, llegamos a la parte de los costos de fabricación. Las dos placas principales y más grandes decidimos enviarlas a fabricar a empresas dedicadas a esto, costando la placa de sensado (Fig. 24) \$19,688 dólares estadounidenses, mientras que la placa del inversor (Fig. 59) costó \$15,454 dólares estadounidenses. El costo de fabricación de cada PCB incluye el costo de la placa, terminación con máscara antisoldante, pads con flux y también el envío a domicilio. Por otro lado, las dos placas más pequeñas las fabricamos de forma casera para economizar los costos, así los gastos en materiales para ello se listan en la Tabla 4.

Material	Cantidad	Precio Unitario (USD)	Precio (USD)
Placa de Cobre Simple Faz 5x5[cm]	1	\$0,911	\$0,911
Placa de Cobre Simple Faz 10x5[cm]	1	\$1,749	\$1,749
Flux Líquido 10[cc]	1	\$2,083	\$2,083
Estaño 67 % 0.5[mm] 500[gr]	1	\$2,128	\$2,128
Alcohol Isopropílico 125[mL]	1	\$2,204	\$2,204
Impresión en Papel Satinado	2	\$0,228	\$0,456
Limpiador Antigrasa CIF 900[mL]	1	\$2,213	\$2,213
Cloruro Férrico 250[cc]	1	\$1,854	\$1,854

Tabla 4: Costos de Materiales para Fabricar las PCBs.

Las PCB que fabricamos manualmente las hicimos primero limpiándolas con un paño de micro fibra con anti grasa, de forma de eliminar todas las impurezas o suciedades del cobre sin dejar marcas ni ralladuras. Luego con la técnica de la termotransferencia pasamos el diseño del papel satinado a la placa para su posterior ataque químico con cloruro férrico. Finalmente realizamos las perforaciones necesarias y nos ayudamos del flux líquido para realizar las soldaduras de la mejor forma posible. Para resumir, nuestros costos de fabricación totales son de \$48,740 dólares estadounidenses.

 <b>UTN Regional Paraná</b>	Filtro Armónico Activo
Ingeniería en Electrónica	4.Análisis de Costos

Finalmente nos falta el costo del gabinete, cuya impresión 3D en filamento PLA nos costó \$10,646 dólares estadounidenses. A esto debemos sumarle el costo de las pegatinas personalizadas, los interruptores, conectores, el cable de alimentación, el porta fusible con su respectivo fusible y los tornillos de fijación (Tabla 5). Por lo que el costo total del gabinete es de \$31,671 dólares estadounidenses.


Material	Cantidad	Precio Unitario (U\$D)	Precio (U\$D)
Ficha Banana Hembra	2	\$0,433	\$0,866
Interruptor para Chasis	3	\$1,749	\$5,247
Porta Fusible A Rosca	1	\$0,757	\$0,757
Fusible 1.5[A]	1	\$0,473	\$0,473
Conector IEC 60320 C13	1	\$2,090	\$2,090
Cable de Alimentación	1	\$3,401	\$3,401
Tornillos con Tuercas 3x40[mm]	4	\$0,112	\$0,449
Tornillos con Tuercas 3x20[mm]	4	\$0,110	\$0,443
Separadores Hexagonales	4	\$1,540	\$6,159
Plancha de Pegatinas	1	\$1,140	\$1,140

Tabla 5: Costos de Materiales para el Gabinete.

Resumiendo, los costos desglosados del proyecto son los siguientes:

- Componentes de Sensado: \$90,546
- Componentes del Inversor: \$39,222
- Costos de Fabricación de PCB: \$48,740
- Costo del Gabinete: \$31,671
- Transformador 12+12[V] 3[A]: \$22,813

Lo que indica que el costo total de fabricación de nuestro filtro activo de potencia es de \$232,992 dólares estadounidenses.

 <b>UTN Regional Paraná</b>	Filtro Armónico Activo
Ingeniería en Electrónica	5.Discusión y Conclusión

## Capítulo 5: Discusión y Conclusión


Antes de comenzar debemos recordar los objetivos planteados para este proyecto, los cuales son:

- Tener la capacidad de filtrar hasta el vigésimo armónico.
- Conseguir bajar el THDi en torno al 5 %.
- Presentar un inversor escalable capaz de suministrar una potencia de 500[W].
- Lograr un diseño funcional, económico, competitivo y trasladable a un ambiente industrial.
- Cumplir con las normativas y estándares anteriormente establecidas.

Como mencionamos en la sección de Resultados, al emplear los sensores de corriente SCT-013 de la marca YHDC en conjunto con un filtro pasa bajos de arquitectura Sallen-Key con una frecuencia de corte de 1.5[KHz], el ancho de banda queda limitado. Esto lo vimos claramente al medir la corriente que generaba el inversor en las pruebas y ver que las imágenes del espectro que se generan debido a la modulación PWM no aparecen luego en las mediciones que realiza la etapa de sensado. Además midiendo cargas con un THDi del 94,61 % vimos que podemos medir y generar corrientes de hasta el vigésimo armónico sin inconvenientes, cumpliendo así con este objetivo.

Al mismo tiempo, en las pruebas realizadas pudimos disminuir la distorsión total armónica en un caso de 74,13 % a 2,78 %, en otro caso de 94,61 % a 3,43 % y también en un caso de 49,16 % a 3,08 % demostrando así que logramos mantener un THDi resultante siempre por debajo del 5 %. Incluso al aplicar variaciones en tiempo real a la corriente de carga.

Por otro lado, en lo que respecta a nuestro inversor, el mismo se basa en un circuito de puente completo con drivers de control lo cual permite suministrar tanta potencia como lo posibiliten los IGBT o MOSFETS empleados. Nuestro diseño implementado es capaz de emplear 311[V] de continua para generar así una corriente alterna que permita la reinyección a la red como lo hacen los filtros armónicos activos comerciales, pudiendo entregar hasta 500[W] de potencia. En nuestra implementación no realizamos esta reinyección de corriente debido a que no era nuestra intención implementar esta etapa de potencia pero aún así diseñamos el circuito inversor para que sea capaz de funcionar de esta manera, y en nuestra implementación lo usamos con una tensión de 12[V]. Además, se puede copiar un medio puente de modo de tener tres medios puentes en lugar de dos y configurarlos para así tener un sistema trifásico.

 <b>UTN Regional Paraná</b>	<b>Filtro Armónico Activo</b>
<b>Ingeniería en Electrónica</b>	<b>5.Discusión y Conclusión</b>

Es gracias al diseño que optamos por implementar que podemos decir que nuestro filtro armónico activo es escalable en potencia, ya que requiere solamente del cambio de transistores, y también trasladable a un circuito trifásico, debido a que solo se debería implementar un medio puente adicional.

En cuanto al diseño en general del filtro, los transistores IKD10N60RC2 cuentan con un diodo intrínseco de marcha libre lo que evita los problemas que podrían generar las cargas inductivas. Esto junto con el diseño robusto del chip de acuerdo con el fabricante[24], ya que soporta ambientes húmedos y también puede funcionar en un rango de temperaturas de entre -40 a 175 grados centígrados, y la capacidad del driver IR2110, de acuerdo con [23], de poder funcionar en un rango de temperaturas de -40 a 125 grados centígrados teniendo además, en cada entrada de datos, una resistencia de pull-down y un detector Schmitt trigger todo esto hace que el circuito completo del inversor sea apto para funcionar en ambientes industriales.

Considerando entonces que el inversor usa componentes aptos para el ambiente industrial y como usamos optoacopladores para aislar el inversor de la etapa de sensado, se puede separar estas etapas y conectarlas mediante un cable mallado para así colocar la etapa de sensado en un lugar más seguro y el inversor cerca de la carga cuya corriente se quiere filtrar. Además, al haber usado buenas prácticas, de acuerdo con [33], a la hora de diseñar la placa de la etapa de sensado, usando dos planos de masa y componentes SMD para así disminuir lo máximo posible las interferencias o ruido electromagnético se podría emplear un gabinete metálico conectado a tierra para así proteger aún más el circuito de las interferencias electromagnéticas que pueden estar presentes en el ambiente industrial, como sugiere [33] y así la etapa de sensado sería apta también para ambientes industriales con esta pequeña modificación.

Para establecer un punto de comparación entre nuestro filtro armónico activo implementado y uno comercial, buscamos las características del filtro monofásico Mini AHF de la marca COEPOWER [34] enfocándonos en el costo, el voltaje y frecuencia de funcionamiento, la capacidad de entrega de corriente, el rango de filtrado, la frecuencia switching de trabajo, que controlador usa, que protocolo de comunicación emplea y por último su tamaño y peso. La comparación de todos estos datos entre el filtro comercial y el propuesto en este trabajo se puede observar en la Tabla 6.




Características	COEPOWER Mini AHF	Filtro Propuesto
Costo	\$800	\$233
THDi	%5,0	%5,0
Voltaje Funcionamiento	220[V]	220[V]
Entrega de Corriente	15[A]	2,72[A]
Frecuencia de Funcionamiento	50[Hz]	50[Hz]
Rango de Filtrado	3er-50mo	3er-20mo
Frecuencia Switching	20[KHz]	15[KHz]
Controlador	DSP+FPGA	FPGA
Comunicación	MODBUS / Ethernet	USB
Tamaño	452,5x450x86 [mm]	200x200x100 [mm]
Peso	14[Kg]	1,96[Kg]

Tabla 6: Comparación entre Filtro Comercial y el Propuesto.

Al comparar nuestro filtro con el comercial, vimos que logramos proveer el mismo valor de THDi pero nuestra capacidad de corriente es menor y también nuestro rango de filtrado. Sin embargo tenemos las ventajas de un mucho menor tamaño y peso, junto también a un menor costo. Aunque al costo debemos agregarle el costo de trabajo personal, estaríamos aún muy por debajo del costo total final del filtro comercial. Por lo tanto, en la comparación nuestro dispositivo resulta tener las mismas prestaciones en cuanto a filtrado se refiere que uno comercial, mientras que tiene menores prestaciones en cuanto a potencia pero tiene un menor tamaño, menor peso y también menor costo.

Ahora bien, si lo comparamos con un filtro armónico activo que emplea el mismo algoritmo que el nuestro, es decir que se basa en el uso de un filtro adaptativo predictivo [7] pero implementado con tecnología DSP podemos ver que el uso de la tecnología FPGA significa una mejora en cuanto al filtrado. En [7] el filtro desarrollado es también del tipo shunt y funciona también en sistemas monofásicos pero para una tensión de 110[V] y 60[Hz], en este caso los autores del artículo emplearon una carga conformada por una resistencia y un inductor para así generar armónicos y lograron pasar de un THDi original de 25,3 % a uno del 5,93 %. Si bien no hemos realizado pruebas midiendo cargas de ese estilo, empleando un dimmer y una carga resistiva pudimos generar una corriente con un THDi de 94,61 % y logramos reducirlo a un 3,43 % lo cual representa una capacidad de filtrado mucho mayor a la lograda en [7] siendo que estamos empleando el mismo algoritmo.

 <b>UTN Regional Paraná</b>	<b>Filtro Armónico Activo</b>
<b>Ingeniería en Electrónica</b>	<b>5.Discusión y Conclusión</b>

Las prestaciones finales de nuestro filtro armónico activo se encuentran listadas en la Tabla 6, donde se puede ver entre lo más destacado la capacidad de mantener un THDi por debajo del 5% filtrando hasta el vigésimo armónico, el pequeño tamaño de 200x200x100[mm], su poco peso de 1,96[Kg] y su muy bajo costo de \$233 dólares estadounidenses. Con estas características en mente debemos decir que la principal diferencia de nuestro filtro frente a otros similares es la poca corriente que es capaz de suministrar, pero sabemos que este diseño es fácilmente escalable para mayores potencias por lo que no es un problema difícil de solucionar. Por otro lado, el rango de filtrado de nuestro dispositivo si es menor frente al rango de filtros comerciales similares y es algo mucho más complejo de modificar ya que esto incumbe a los sensores de corriente empleados, a los circuitos de filtrado previos al ADC y también a la capacidad de procesamiento de la FPGA. Sin embargo, queremos destacar el correcto funcionamiento de nuestro filtro el cuál aún con corrientes con gran carga de armónicos y aplicando variaciones en tiempo real de la carga siempre respondió manteniendo por debajo del 5% el THD de la señal resultante lo que indica una gran robustez y fiabilidad.

Para finalizar, creemos que este filtro armónico activo podría ser una excelente alternativa para comercios o pequeñas empresas donde los requisitos de potencia no sean tan elevados ya que se podría colocar este pequeño filtro por un muy bajo costo y asegurar así el correcto filtrado de los armónicos para prevenir así todos los inconvenientes que esto conlleva. Además, como posible mejora queremos mencionar el reemplazar el microcontrolador Raspberry Pi Pico por su modelo Pico W el cual mantiene la misma disposición de pines y procesador pero añade la funcionalidad de conexión inalámbrica mediante WiFi, lo que serviría para hacer el envío de datos por Internet en lugar de por un cable USB, lo que daría la oportunidad de usar herramientas del internet de las cosas (IoT) para realizar un servidor web que este mostrando en tiempo real las mediciones y el funcionamiento del filtro para que cualquier usuario de forma remota pueda acceder al mismo y corroborar su funcionamiento.



## Capítulo 6: Referencias

- [1] R. Pi, "A microcontroller by raspberry pi," 2019, [Revisado Feb. 2025].
- [2] YHDC, "Split core current transformer," SCT-013 Datasheet, 2005, [Revisado Oct. 2012].
- [3] Microchip, "Single/dual digital potentiometer with spi interface," MCP42050 Datasheet, 1995, [Revisado Jul. 2003].
- [4] trevortjes, "10 opamps: Dc bias for single supply circuits," 2020. [Online]. Available: <https://www.youtube.com/watch?v=Ut-m8dl7INA&t=186s> [Accedido Sept. 2024].
- [5] T. Instruments, "Cd4013b cmos dual d-type flip-flop," CD4013 Datasheet, 1998, [Revisado Sept. 2016].
- [6] S. A. Khan, *Digital Design of Signal Processing Systems*. Wiley, 2011.
- [7] B.-M. Han, B.-Y. Bae, and S. J. Ovaska, "Reference signal generator for active power filter using improved adaptive predictive filter," *IEEE Transactions on Industrial Electronics*, 2005.
- [8] R. R. Pereira, C. H. da Silva, L. E. B. da Silva, and G. Lambert-Torres, "Application of adaptive filters in active power filters," in *2009 Brazilian Power Electronics Conference*, 2009.
- [9] Q. Chaudhari, "Least mean square (lms) equalizer - a tutorial," 2024. [Online]. Available: <https://wirelesspi.com/least-mean-square-lms-equalizer-a-tutorial/> [Accedido Mar. 2025].
- [10] —, "Cascaded integrator comb (cic) filters - a staircase of dsp," 2024. [Online]. Available: <https://wirelesspi.com/cascaded-integrator-comb-cic-filters-a-staircase-of-dsp/> [Accedido Mar. 2025].
- [11] M. H. Rashid, *Power Electronics Handbook*. Elsevier, 2011.
- [12] V. Semiconductors, "High speed optocoupler, single and dual, 10mbd," 6N137 Datasheet, 1998, [Revisado Feb. 2017].
- [13] W. W. K Eichert, T. Mangold, "Power quality issues and their solutions," in VII Seminario de Electrónica de Potencia, Valparaíso, Chile, Abril 1999.
- [14] M. El-Habrouk, M. Darwish, and P. Mehta, "Active power filters: A review," *Electric Power Applications, IEE Proceedings* -, vol. 147, pp. 403 – 413, 10 2000.
- [15] M. Robert Wall B.Sc., C.Eng., "A report on the properties of the yhdc current transformer," OpenEnergy, Tech. Rep., 2014.
- [16] T. Instruments, "Opax134 soundplus high performance audio operational amplifiers," OPA4134 Datasheet, 1997, [Revisado Nov. 2024].
- [17] C. F. Ian Williams, "Parasitics in precision pcb layouts," *Texas Instruments*, 2013.
- [18] P. Butler, "Antialiasing filtering considerations for high precision sar analog-to-digital converters," *Analog Dialogue*, 2018.
- [19] Fluke, "Pinza amperimétrica fluke ca/cc 80i-110s," Fluke CA/CC 80i-110s Datasheet, 2024, [Revisado Dec. 2024].
- [20] J. Schneider and I. Smalley, "¿qué es una matriz de puertas programables en campo (fpga)?" 2024. [Online]. Available: <https://www.ibm.com/mx-es/think/topics/field-programmable-gate-arrays#:~:text=Una%20FPGA%20es%20un%20tipo,y%20la%20creaci%C3%B3n%20de%20prototipos.> [Accedido Mar. 2025].
- [21] Y. Krinin, "How fpgas work, and why you'll buy one," 2013. [Online]. Available: <https://www.fpgarelated.com/showarticle/357/how-fpgas-work-and-why-you-ll-buy-one> [Accedido Mar. 2025].
- [22] F. L. et all, "Dc to ac inverter prototype for small scale power supply with spwm method," *Control Systems and Optimization Letters*, 2023.
- [23] I. Rectifier, "High and low side driver," IR2110 Datasheet, 2003, [Revisado Jun. 2019].
- [24] Infineon, "Cost effective monolithically integrated igbt with diode," 2020, [Revisado Sept. 2020].
- [25] Fairchild, "Lm78xx 3-terminal 1 a positive voltage regulator," 2006, [Revisado Sept. 2014].
- [26] J. Adams, *Design Tip DT98-2a*. International Rectifier, 2001.
- [27] D. Incorporated, "Super fast glass passivated rectifier," MUR460 Datasheet, 2002, [Revisado Oct. 2021].
- [28] Toshiba, *MOSFET Gate Drive Circuit*. Toshiba, 2018.
- [29] GOWIN, "Gw1nr series of fpga products," FPGA Tang Nano 9K Datasheet, 2018, [Revisado Oct. 2021].
- [30] Tektronix, "Digital storage oscilloscopes tds1000b series," Tektronix TDS1002B Datasheet, 2010, [Revisado Feb. 2016].

- [31] T. Instruments, "TI07xx low-noise fet-input operational amplifiers," 1978, [Revisado Abr. 2023].
- [32] TOSHIBA, "Discrete silicon n-channel igbt," 2012, [Revisado Ene. 2014].
- [33] T. Instruments, *PCB Design Guidelines For Reduced EMI*. Texas Instruments, 1999.
- [34] C. Powers, "Mini active harmonic filter," 2021. [Online]. Available: <https://www.coepowers.com/product/mini-active-harmonic-filter-ahf/> [Accedido Mar. 2025].