

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE
TELECOMUNICACIÓN



UNIVERSITAS
Miguel Hernández

"Estudio y análisis de diferentes tecnologías inalámbricas
de localización usando dispositivos LED, ultrasonidos y
láser"

TRABAJO FIN DE GRADO

Junio - 2024

AUTORA: María Belén Espinosa Martínez

DIRECTOR: Pablo Corral González



RESUMEN

Este Trabajo de Fin de Grado tiene como objetivo analizar diferentes sistemas de localización en tiempo real, mediante estimación de distancias y triangulación, utilizando los algoritmos RSSI (Received Signal Strength Indication), TOA (Time of Arrival) y TOF (Time of Flight) implementados sobre diferentes plataformas. La finalidad es medir y analizar la precisión de la localización de objetos con diferentes algoritmos y tecnologías, ofreciendo una solución práctica y eficaz para aplicaciones en la industria.

El proyecto se ha llevado a cabo en varias etapas, comenzando con el montaje del hardware necesario mediante la integración de sensores de ultrasonido HC-SR04, sensores láser VL6180X y fotodetectores BPW21 y BPW77NB con placas Arduino Uno y Mega 2560. Los fotodetectores han sido fundamentales para la implementación del algoritmo RSSI, proporcionando mediciones precisas de la intensidad de la señal recibida. El sensor de ultrasonidos se ha usado para implementar el algoritmo TOA. Y finalmente, con el sensor láser VL6180X se ha implementado el algoritmo TOF.

Posteriormente, se han implementado y programado los algoritmos de localización (TOA y TOF) en el entorno Arduino IDE, y se han creado scripts en Visual Studio Code específicos para la automatización de la toma de datos y la optimización de las mediciones.

Para la captura y análisis de las señales de los fotodetectores (algoritmo RSSI), se han utilizado herramientas como el osciloscopio digital WaveForms y el Analog Discovery 2. Los datos obtenidos han sido procesados y visualizados con herramientas de software como Visual Studio Code, permitiendo la creación de gráficos y tablas comparativas que han facilitado la evaluación del rendimiento y precisión de cada algoritmo.

El análisis de los resultados ha permitido identificar las ventajas y limitaciones de cada tecnología y algoritmo utilizado, contribuyendo al desarrollo de sistemas de localización en tiempo real más precisos y eficientes.

Palabras clave: Localización en tiempo real, RSSI, TOA, TOF, Arduino, sensores de ultrasonido, sensores láser, fotodetectores, Analog Discovery 2.



ABSTRACT

This Final Degree Project aims to analyze different real-time localization systems, by means of distance estimation and triangulation, using RSSI (Received Signal Strength Indication), TOA (Time of Arrival) and TOF (Time of Flight) algorithms implemented on different platforms. The aim is to measure and analyze the accuracy of object localization with different algorithms and technologies, offering a practical and efficient solution for industrial applications.

The project has been carried out in various stages, beginning with the required hardware set up by integrating HC-SR04 ultrasound and VL6180X laser sensors and BPW21 and BPW77NB photodetectors with Arduino Uno and Mega 2560 plaques. The photodetectors have been fundamental for the implementation of the RSSI algorithm, providing accurate measurements of the received signal strength. The ultrasound sensor has been used to implement the TOA algorithm. And finally, the VL6180X laser sensor has been used to implement the TOF algorithm.

Subsequently, the localization algorithms (TOA and TOF) have been implemented and programmed in the Arduino IDE environment, and specific scripts have been created in Visual Studio Code for the automation of data acquisition and optimization of the measurements.

To capture and analyze the signals from the photodetectors (RSSI algorithm), tools such as the WaveForms digital oscilloscope and Analog Discovery 2 have been used. The data obtained were processed and visualized with software tools such as Visual Studio Code, allowing the creation of graphs and comparative tables that facilitated the evaluation of the performance and accuracy of each algorithm.

The analysis of the results has allowed us to identify the advantages and limitations of each technology and algorithm used, contributing to the development of more accurate and efficient real-time localization systems.

Keywords: Real-time localization, RSSI, TOA, TOF, Arduino, ultrasound sensors, laser sensors, photodetectors, Analog Discovery 2.





ÍNDICE

RESUMEN	3
ABSTRACT	5
ÍNDICE	8
Contenido	8
ÍNDICE DE TABLAS	12
ÍNDICE DE ILUSTRACIONES	14
ÍNDICE DE GRÁFICAS	16
1. INTRODUCCIÓN Y OBJETIVOS	20
1.1 Contexto y motivación	20
1.2 Objetivos	21
2. MATERIAL Y MÉTODOS.....	24
2.1. Hardware	24
2.2. Software.....	35
2.3. Fundamentos teóricos	38
2.3.1. Algoritmo RSSI.....	38
2.3.2. Algoritmo Estimación del Tiempo de Llegada (TOA).....	40
2.3.3. Algoritmo TOF.....	41
2.4. Montaje eléctrico para toma de medidas	42
2.4.1. Analog Discovery 2 con fotodetector BPW21 y Led comercial (RSSI para estimación de distancias).....	43
2.4.2. Analog Discovery 2 con tres fotodetectores BPW21 y Led comercial (RSSI para triangulación)	43
2.4.3. Arduino UNO con sensor de ultrasonidos HC-SR04 (TOA para estimación de distancias)	45

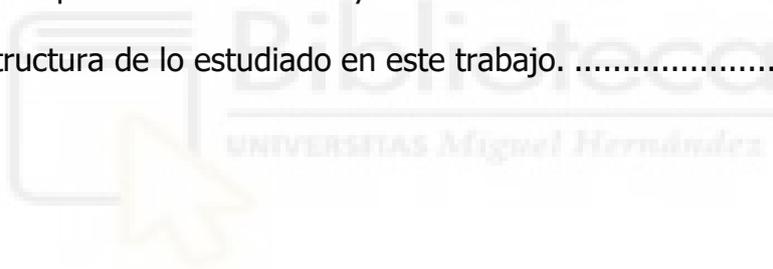
2.4.4.	Arduino Uno con 3 Ultrasonidos (TOA para triangulación)	46
2.4.5.	Arduino Mega 2560 con sensor láser VL6180X (TOF para estimación de distancias)	48
2.5.	Programación	49
2.5.1.	Código RSSI con fotodetector BPW21 en Analog Discovery 2. ...	49
2.5.2.	Código RSSI tres fotodetectores BPW21 en Analog Discovery 2.	49
2.5.3.	Código TOA con sensor de ultrasonidos HC-SR04 en Arduino IDE para estimación de distancias.	49
2.5.4.	Código TOA con sensor de ultrasonidos HC-SR04 en Arduino IDE para triangulación.....	51
2.5.5.	Código TOF con sensor láser VL6180X en Arduino IDE para estimación de distancias.....	52
3.	RESULTADOS.....	56
3.1.	Algoritmo RSSI estimación distancia led comercial y fotodetector BPW21	56
3.2.	Algoritmo RSSI triangulación con led comercial y tres fotodetectores BPW21	64
3.3.	Algoritmo TOA para estimación de distancias con ultrasonidos HC-SR04	78
3.4.	Algoritmo TOA para triangulación con tres ultrasonidos.....	89
3.5.	Algoritmo TOF con estimación de distancias y un sensor láser VL6180X	109
4.	CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN	116
4.1.	Conclusiones.....	116
4.2.	Líneas futuras.....	119
5.	ANEXOS.....	122

5.1. Código TOA con sensor de ultrasonidos HC-SR04 en Arduino IDE para estimación de distancias.....	122
5.2. Código Python para gráficas TOA con estimación de distancias.....	123
5.3. Código Python para cálculos TOA para estimación de distancias.....	123
5.4. Código TOA con triangulación con tres sensores de ultrasonidos HC-SR04 en Arduino IDE.....	126
5.5. Código Python para gráficas TOA con triangulación.....	127
5.6. Código Python para gráficas Error Relativo TOA con triangulación...	130
5.7. Código Python para cálculos TOA con triangulación.....	132
5.8. Código Sensor láser VL6180X en Arduino IDE para estimación de distancias	134
5.9. Código Python para gráficas Sensor Láser para estimación de distancias	136
5.10. Código Python para gráficas Sensor láser VL6180X para estimación de distancias Valor de Referencia VS Error Relativo.....	139
5.11. Código Python para cálculos Sensor láser VL6180X para estimación de distancias.....	140
6. BIBLIOGRAFÍA	144



ÍNDICE DE TABLAS

Tabla 1. Tabla valores RSSI estimación de distancias en mV.	62
Tabla 2. Tabla valores RSSI estimación de distancias en mV.	75
Tabla 3. Tabla valores RSSI triangulación en mV para el fotodetector 2.	75
Tabla 4. Tabla valores RSSI triangulación en mV para el fotodetector 3.	76
Tabla 5. Resumen de medidas para algoritmo TOA con un sensor Ultrasonidos HC-SR04.	88
Tabla 6. Resumen de medidas para algoritmo TOA con triangulación con tres sensores DE Ultrasonidos HC-SR04.	107
Tabla 7. Valores para el Error Relativo máximo del sensor láser VL6180X.	110
Tabla 8. Valores para el Error Relativo mínimo del Sensor láser VL6180X.	110
Tabla 9. Valores para el RTT máximo y mínimo del Sensor láser VL6180X.	112
Tabla 10. Estructura de lo estudiado en este trabajo.	116





ÍNDICE DE ILUSTRACIONES

Ilustración 1. Arduino UNO.	24
Ilustración 2. Arduino MEGA 2560.	25
Ilustración 3. Sensor HC-SR04.....	26
Ilustración 4. Sensor láser VL6180X.....	27
Ilustración 5. Fotodetector BPW21.	28
Ilustración 6. Diagrama de radiación del fotodetector BPW21.....	29
Ilustración 7. Fotodetector BPW77NB.	30
Ilustración 8. Diagrama de radiación del fotodetector BPW77NB.	31
Ilustración 9. Led 10 mm blanco de alta luminosidad.....	32
Ilustración 10. Fuente óptica Lambertiana.....	33
Ilustración 11. Analog Discovery 2.....	34
Ilustración 12. Esquema conexiones osciloscopio digital Analog Discovery 2... 34	
Ilustración 13. Captura de pantalla del entorno de programación de Arduino IDE.	35
Ilustración 14. Captura de pantalla del monitor serie de Arduino IDE.....	36
Ilustración 15. Ejemplo de captura de pantalla de medición del Osciloscopio de WaveForms.....	37
Ilustración 16. Captura de pantalla del entorno Visual Studio Code.....	38
Ilustración 17. Esquema de conexión Analog Discovery 2 con fotodetector BPW21 y LED COMERCIAL.....	43
Ilustración 18. Esquema de conexión dos Analog Discovery 2 con tres fotodetectores BPW21	44
Ilustración 19. Esquema de conexión de Arduino UNO con Sensor HC-SR04. .	45
Ilustración 20. Esquema de conexión de Arduino UNO con tres Sensores HC- SR04.	46

Ilustración 21. Disposición real de los tres sensores HC-SR04.....	47
Ilustración 22. Montaje de conexión Sensor láser VL6180X con Arduino Mega 2560.....	48
Ilustración 23. Diagrama de Flujo del código de Arduino UNO y un sensor HC-SR04.	50
Ilustración 24. Diagrama de Flujo del código de Arduino UNO y tres sensores HC-SR04.	52
Ilustración 25. Diagrama de Flujo del código de Arduino UNO y tres sensores HC-SR04.	54
Ilustración 26. Imagen Osciloscopio a 8 cm y 10 kHz en amarillo la señal transmitida y en azul la señal recibida.	63
Ilustración 27. Imagen Osciloscopio a 10 cm y 100 kHz, en amarillo la señal transmitida y en azul la señal recibida.	63
Ilustración 28. Imagen Osciloscopio a 10 cm y 1000 kHz, en amarillo la señal transmitida y en azul la señal recibida.	64
Ilustración 29. Osciloscopio del fotodetector 1 a 6 cm y 50 kHz, en amarillo la señal transmitida y en azul la señal recibida.....	77
Ilustración 30. Osciloscopio del fotodetector 2 (amarillo) y 3 (azul) a 1 cm y 1000 kHz.	78

ÍNDICE DE GRÁFICAS

Gráfica 1. Comparación del voltaje Peak2Peak VS Distancia para 1 kHz.	56
Gráfica 2. Comparación del voltaje Peak2Peak VS Distancia para 2 kHz.	57
Gráfica 3. Comparación del voltaje Peak2Peak VS Distancia para 5 kHz.	57
Gráfica 4. Comparación del voltaje Peak2Peak VS Distancia para 10 kHz.....	57
Gráfica 5. Comparación del voltaje Peak2Peak VS Distancia para 20 kHz.....	58
Gráfica 6. Comparación del voltaje Peak2Peak VS Distancia para 50 kHz.....	58
Gráfica 7. Comparación del voltaje Peak2Peak VS Distancia para 100 kHz.....	58
Gráfica 8. Comparación del voltaje Peak2Peak VS Distancia para 200 kHz.....	59
Gráfica 9. Comparación del voltaje Peak2Peak VS Distancia para 500 kHz.....	59
Gráfica 10. Comparación del voltaje Peak2Peak VS Distancia para 1000 kHz. .	59
Gráfica 11. Combinación de valores RSSI estimación de distancias.	62
Gráfica 12. Comparación del voltaje Peak2Peak VS Distancia con tres fotodetectores para 1 kHz.....	65
Gráfica 13. Comparación del voltaje Peak2Peak VS Distancia con tres fotodetectores para 2 kHz.....	66
Gráfica 14. Comparación del voltaje Peak2Peak VS Distancia con tres fotodetectores para 5 kHz.....	67
Gráfica 15. Comparación del voltaje Peak2Peak VS Distancia con tres fotodetectores para 10 kHz.....	68
Gráfica 16. Comparación del voltaje Peak2Peak VS Distancia con tres fotodetectores para 20 kHz.....	69
Gráfica 17. Comparación del voltaje Peak2Peak VS Distancia con tres fotodetectores para 50 kHz.....	70
Gráfica 18. Comparación del voltaje Peak2Peak VS Distancia con tres fotodetectores para 100 kHz.....	71

Gráfica 19. Comparación del voltaje Peak2Peak VS Distancia con tres fotodetectores para 200 kHz.	72
Gráfica 20. Comparación del voltaje Peak2Peak VS Distancia con tres fotodetectores para 500 kHz.	73
Gráfica 21. Comparación del voltaje Peak2Peak VS Distancia con tres fotodetectores para 1000 kHz.	74
Gráfica 22. Comparación de Valores Obtenidos VS Valores Reales para la Constante 0.034.....	79
Gráfica 23. Comparación de Error Relativo (%) VS Valor de Referencia para la Constante 0.034.....	80
Gráfica 24. Comparación de Error Relativo (%) VS Valor de Referencia para la Constante 0.034 ampliado.	80
Gráfica 25. Comparación de Valores Obtenidos VS Valores Reales para la Constante 0.0353.....	81
Gráfica 26. Comparación de Error Relativo (%) VS Valor de Referencia para la Constante 0.0353.....	82
Gráfica 27. Comparación de Error Relativo (%) VS Valor de Referencia para la Constante 0.0353 ampliado.....	83
Gráfica 28. Comparación de Valores Obtenidos VS Valores Reales para la Constante 0.03432.....	84
Gráfica 29. Comparación de Error Relativo (%) VS Valor de Referencia para la Constante 0.03432.....	85
Gráfica 30. Comparación de Error Relativo (%) VS Valor de Referencia para la Constante 0.03432 ampliado.....	85
Gráfica 31. Comparación de Valores Obtenidos VS Valores Reales para la Constante 0.03483.....	86
Gráfica 32. Comparación de Error Relativo (%) VS Valor de Referencia para la Constante 0.03483.....	87

Gráfica 33. Comparación de Error Relativo (%) VS Valor de Referencia para la Constante 0.03483 ampliado.....	87
Gráfica 34. Comparación de Valores Obtenidos VS Valores Reales del Sensor 1 para la Constante 0.0353.....	91
Gráfica 35. Comparación de Valores Obtenidos VS Valores Reales del Sensor 2 para la Constante 0.0353.....	92
Gráfica 36. Comparación de Valores Obtenidos VS Valores Reales del Sensor 3 para la Constante 0.0353.....	93
Gráfica 37. Comparación de Error Relativo (%) VS Valor de Referencia de los sensores para la Constante 0.0353.....	94
Gráfica 38. Comparación de Error Relativo (%) VS Valor de Referencia de los sensores para la Constante 0.0353 ampliado.....	95
Gráfica 39. Comparación de Valores Obtenidos VS Valores Reales para el Sensor 1.....	97
Gráfica 40. Comparación de Valores Obtenidos VS Valores Reales para el Sensor 2.....	98
Gráfica 41. Comparación de Valores Obtenidos VS Valores Reales para el Sensor 3.....	99
Gráfica 42. Comparación de Error Relativo (%) VS Valor de Referencia de los sensores para la constante 0.03432.....	100
Gráfica 43. Comparación de Error Relativo (%) VS Valor de Referencia de los sensores para la constante 0.03432 ampliado.....	101
Gráfica 44. Comparación de Valores Obtenidos VS Valores Reales para el Sensor 1.....	102
Gráfica 45. Comparación de Valores Obtenidos VS Valores Reales para el Sensor 2.....	103
Gráfica 46. Comparación de Valores Obtenidos VS Valores Reales para el Sensor 3.....	104

Gráfica 47. Comparación de Error Relativo (%) VS Valor de Referencia de los sensores para la constante 0.03483.....	105
Gráfica 48. Comparación de Error Relativo (%) VS Valor de Referencia de los sensores para la constante 0.03483 ampliado.	106
Gráfica 49. Comparación de Valores Obtenidos VS Valores Reales para el sensor láser VL6180X.	109
Gráfica 50. Comparativa del Valor de Referencia (cm) VS Error Relativo (%) para el Sensor láser VL6180X.	111



1. INTRODUCCIÓN Y OBJETIVOS

1.1 Contexto y motivación

Los sistemas de localización en tiempo real son de gran interés en diferentes campos. Los algoritmos RSSI, TOA y TOF son técnicas de localización ampliamente utilizadas para medir la posición de objetos o personas en tiempo real. Sin embargo, la implementación práctica de estos algoritmos puede ser un desafío significativo debido a la complejidad del procesamiento de señales y la necesidad de hardware especializado.

Los fotodetectores son dispositivos esenciales en sistemas de localización y detección de luz. Su principal función es convertir la luz en señales eléctricas, lo que los hace de gran valor en diversas aplicaciones tecnológicas. En el contexto de la localización en tiempo real, los fotodetectores juegan un papel crucial en la implementación de algoritmos como RSSI. En este caso, haremos uso del fotodetector y el led conectado directamente al dispositivo Analog Discovery 2, que hace las veces de osciloscopio así como de generador de funciones.

La plataforma Arduino, conocida por su accesibilidad y versatilidad, ofrece una excelente base para desarrollar sistemas de localización en tiempo real. Su capacidad de integración con diversos sensores y su facilidad de programación la convierten en una opción atractiva para proyectos tanto educativos como profesionales.

Para este trabajo se hará uso de una placa Arduino Uno junto a un sensor de ultrasonidos HC-SR04 para estimación de distancias y triangulación, de modo que podamos evaluar el algoritmo TOA en circunstancias diferentes. En este caso, iremos variando la distancia del ultrasonido con respecto al objeto y así evaluar la eficacia de los resultados.

Del mismo modo, para estimación de distancias con el algoritmo TOF acoplaremos un sensor láser VL6180X a la placa de Arduino Mega 2560 e iremos variando su distancia con respecto al obstáculo.

Este Trabajo de Fin de Grado tiene como objetivo analizar diferentes sistemas de localización en tiempo real, mediante estimación de distancias y triangulación,

utilizando los algoritmos RSSI (Received Signal Strength Indication), TOA (Time of Arrival) y TOF (Time of Flight) implementados sobre diferentes plataformas (Arduino y Analog Discovery 2). El objetivo es medir y analizar la precisión de la localización de objetos con diferentes algoritmos y tecnologías, ofreciendo una solución práctica y eficaz para aplicaciones en la industria.

Al abordar este tema, se pretende no solo contribuir al conocimiento técnico en la materia, sino también ofrecer una guía práctica para la implementación de estos sistemas en proyectos reales, destacando los beneficios y limitaciones de cada enfoque. La combinación de hardware accesible con algoritmos avanzados de localización tiene el potencial de revolucionar la forma en que interactuamos con nuestro entorno, mejorando la eficiencia y precisión en una amplia gama de aplicaciones con la intención de obtener una solución óptima que integre una o varias tecnologías.

1.2 Objetivos

El propósito principal de este trabajo es investigar y comparar la precisión de los algoritmos RSSI (Received Signal Strength Indication), TOA (Time of Arrival) para estimación de distancias y triangulación y TOF (Time of Flight) para estimación de distancias en la toma de medidas de localización. Este estudio es fundamental para determinar qué método es más efectivo en función de la distancia, para todos los casos, y particularmente en función de la distancia y la frecuencia para el algoritmo RSSI.

Para llevar a cabo esta investigación, se utilizará una variedad de dispositivos de hardware, principalmente basados en la plataforma ARDUINO. Esto incluye placas como Arduino Uno y Mega 2560, junto con sensores como el de ultrasonidos HC-SR04 y el sensor láser VL6180X. Estos dispositivos proporcionarán la base para la recolección de datos y la implementación de los algoritmos de localización.

Además, para el análisis detallado del algoritmo RSSI, se empleará el Analog Discovery 2, junto con fotodetectores y LEDs. Este equipo permitirá capturar y

analizar las señales de voltaje pico a pico, lo que proporcionará información crucial sobre la fuerza de la señal y su variación en el espacio.





2. MATERIAL Y MÉTODOS

2.1. Hardware

Para el desarrollo de este trabajo se han implementado diversos materiales de hardware, todos ellos girando entorno al principal que ha sido Arduino UNO

- Arduino UNO R3

Arduino UNO se trata de una placa de microcontrolador de código abierto. Esta placa está equipada con un conjunto de pines de E/S digitales y analógicos, lo que permiten conectar placas de expansión y otros circuitos. Esta placa está formada por 14 pines digitales, de los cuales 6 se pueden usar como salidas PWM, 6 entradas analógicas, una conexión USB, un conector ICSP, un oscilador de cristal de 16 MHz y un botón de reset.

(BricoGeek, 2014)



Ilustración 1. Arduino UNO.

- Arduino MEGA 2560

El Arduino Mega 2560 representa una placa de control basada en el ATmega2560. Cuenta con una amplia gama de características, incluyendo 54 pines digitales (con 15 configurables como salidas PWM), 16 entradas analógicas, 4 UART para conexiones serie de hardware, un cristal de oscilador de 16 MHz, conectividad USB, un puerto de alimentación, un encabezado ICSP y un botón de reinicio. Esta placa ofrece todo lo necesario para respaldar el funcionamiento del

microcontrolador. Simplemente se conecta la placa a una computadora mediante un cable USB, o se alimenta con un adaptador de corriente alterna a corriente continua o una batería para iniciar su operación. Además, es compatible con la mayoría de los campos diseñados para Uno, así como con placas anteriores como Duemilanove o Diecimila.

(BricoGeek, 2014)



Ilustración 2. Arduino MEGA 2560.

- Ultrasonidos HC-SR04

El sensor HC-SR04 es un dispositivo económico utilizado para medir distancias mediante ultrasonido en un rango de 2 a 450 cm. Destaca por su tamaño compacto, bajo consumo de energía, precisión y precio accesible. Es ampliamente utilizado en proyectos de robótica y automatización debido a la abundante información y proyectos disponibles en línea. El sensor consta de un emisor y un receptor piezoeléctricos, así como electrónica para su funcionamiento. Emite 8 pulsos de ultrasonido tras recibir la orden del pin TRIG. Estas ondas de sonido viajan en el aire y rebotan al encontrar el objeto. El sonido de rebote es detectado por el receptor piezoeléctrico, por lo que el pin ECHO cambia a Alto (5V) por el tiempo que la onda tarda en ir y volver desde que fue emitida hasta que fue detectada. El tiempo del pulso ECHO es medido por el

microcontrolador y así se puede calcular la distancia al objeto. La distancia se calcula utilizando la siguiente fórmula:

$$distancia [m] = \frac{((Tiempo\ del\ pulso\ ECHO) * (Velocidad\ del\ sonido\ [\frac{m}{s}]))}{2} \quad (1)$$

Su funcionamiento no se ve afectado por la luz solar o materiales negros, aunque puede tener dificultades con materiales acústicamente blandos.

(Naylamp, 2014)



El sensor HC-SR04 emite pulsos de ultrasonido (a una frecuencia de 40 KHz) después de recibir una orden en el pin TRIG. Estos pulsos rebotan en los objetos y son detectados por el receptor piezoeléctrico. El tiempo que tarda en regresar la señal se mide en el pin ECHO (que trabaja a 5V), permitiendo calcular la distancia al objeto. No se ve afectado por la luz solar ni por objetos de color negro, pero puede tener dificultades con materiales blandos como la tela o la lana.

- Sensor láser VL6180X

La placa de expansión X-NUCLEO-6180XA1 está equipada con el sensor ALS, de gestos y de proximidad VL6180X, el cual se basa en la tecnología de tiempo de vuelo FlightSense™ de ST. Esta placa de evaluación ofrece una introducción a las capacidades de proximidad, alcance y detección de luz del módulo VL6180X. Es

compatible con la familia de placas STM32 Núcleo y se ajusta al diseño del conector Arduino UNO R3. A través de los conectores Arduino, es posible superponer varias placas de expansión ST, lo que facilita el desarrollo de aplicaciones VL6180X con interfaces Bluetooth o WiFi, entre otras posibilidades.

(RS-online, Junio)



Ilustración 4. Sensor láser VL6180X.

El sensor VL53L0X, sensor láser en el que se basa nuestro sensor láser, es un sensor láser de última generación que, junto con Arduino, puede medir distancias precisas de 50 mm a 2000 mm. La variante VL6180X es adecuada para rangos más cercanos, de 5 mm a 200 mm.

(Luis Llamas, 2024)

La longitud de onda infrarroja es de 850 nm y utiliza una interfaz de comunicación I2C a 400 kHz, con una dirección del dispositivo en el protocolo de comunicación I2C de 7 bits de 0x29.

- Fotodetector BPW21

Los fotodiodos BPW20 y BPW21 de *Vishay Semiconductor* son componentes de detección de luz que se presentan en encapsulados estándar TO-5. Estos encapsulados están herméticamente sellados, lo que los hace adecuados para aplicaciones que requieren precisión lineal. Los fotodiodos BPW20 y BPW21

encuentran su utilidad en una variedad de aplicaciones, como sensores para la medición de luz en cámaras, fotómetros, analizadores de color, fotómetros y otros dispositivos de medición y control en entornos médicos e industriales.

Las características distintivas de los fotodiodos BPW20 y BPW21 incluyen su encapsulado TO-5 con una lente de vidrio plano, un diámetro de 8,13 mm y un área sensible a la radiación de 7,5 mm². Estos fotodiodos ofrecen una alta sensibilidad a la luz, con un ángulo de media intensidad de 50°, baja corriente oscura y alta linealidad. Esta combinación de características los hace apropiados para aplicaciones donde se requiere una detección precisa y confiable de la luz infrarroja.

Para usar este fotodetector, se conecta al Analog Discovery 2 y así poder medir la respuesta del fotodiodo a la luz.

(Naylamp, 2014)

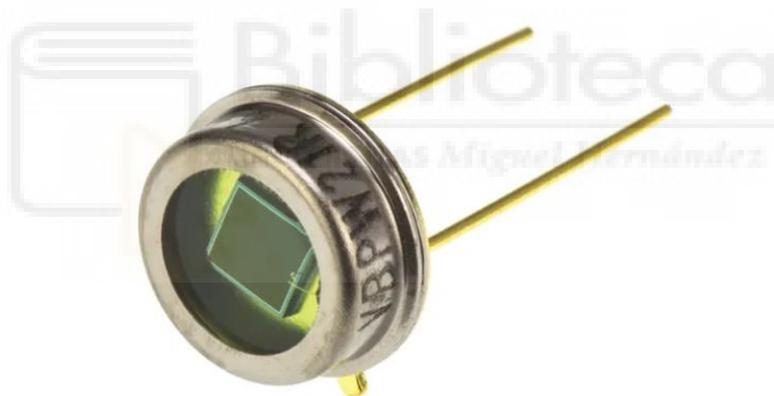


Ilustración 5. Fotodetector BPW21.

$$S_{rel} = f(\varphi)$$

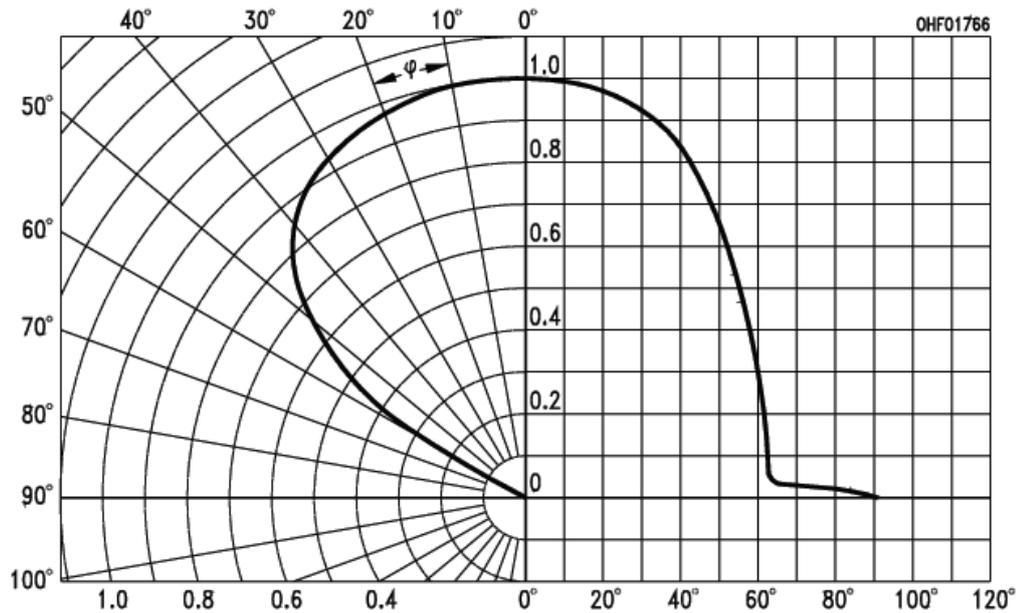


Ilustración 6. Diagrama de radiación del fotodetector BPW21.

En el diagrama, el eje horizontal representa el ángulo de incidencia de la luz (φ), que va desde 0° hasta 120° . El eje vertical indica el ángulo relativo de sensibilidad S_{rel} que varía de 0 a 1, donde 1 indica la máxima sensibilidad.

La curva empieza en su máximo en 0° , indicando la máxima sensibilidad cuando la luz incide directamente de frente al sensor. A medida que el ángulo aumenta, la sensibilidad disminuye, reflejando que el sensor es menos eficiente al captar luz que llega en ángulos más agudos respecto a la superficie del sensor. Alrededor de los 40° - 50° , la sensibilidad ha disminuido significativamente, y después de 90° , la sensibilidad es mínima o nula, lo que implica que el sensor prácticamente no detecta luz incidente más allá de este ángulo.

(Vishay, 2024)

- Fotodetector BPW77NB

El detector de luz VPW77NB es un dispositivo electrónico diseñado para transformar la energía lumínica en una señal eléctrica. Este componente es ampliamente utilizado en diversas aplicaciones que requieren la detección de luz, como sistemas de control automático, instrumentación óptica, comunicaciones ópticas, entre otros.

Las series BPW77NA y BPW77NB de *Vishay Semiconductor* constituyen una familia de fototransistores NPN de silicio. Estos fototransistores se encuentran encapsulados en TO-18, garantizando un sellado hermético con una lente de cristal. Los BPW77NA y BPW77NB son altamente sensibles a la radiación infrarroja visible y cercana, lo que los convierte en opciones ideales para detectores en circuitos de control y accionamiento electrónicos.

Las características principales de los fototransistores BPW77NA y BPW77NB incluyen su encapsulado TO-18 con un diámetro de 4,7 mm, montaje en orificio pasante, alta fotosensibilidad, sensibilidad radiante destacada, tiempos de respuesta rápidos y un amplio rango de temperatura de funcionamiento de -40 a +125 °C.

Estos fototransistores infrarrojos de *Vishay Semiconductor* ofrecen un rendimiento confiable y versátil para una variedad de aplicaciones en electrónica.

(RS-online, 2015)



Ilustración 7. Fotodetector BPW77NB.

Su diagrama de radiación es:

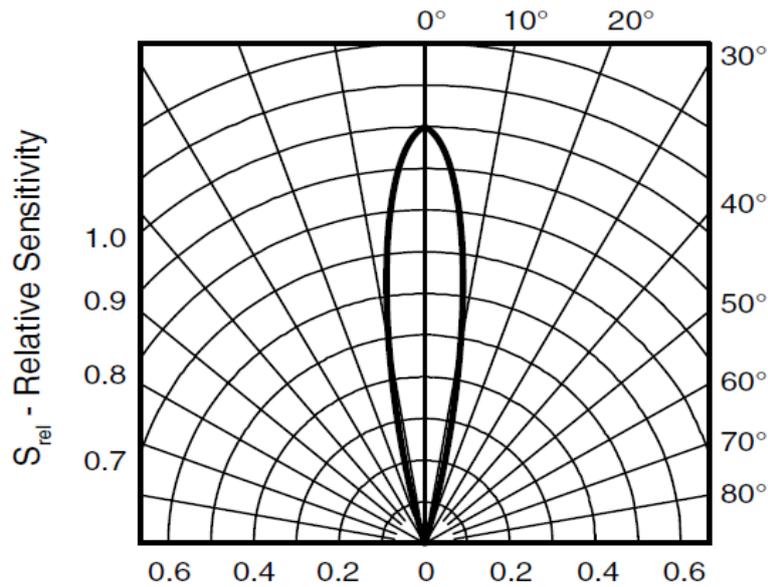


Ilustración 8. Diagrama de radiación del fotodetector BPW77NB.

El diagrama de radiación para el fotodetector BPW77NB ilustra la sensibilidad relativa (S_{rel}) del dispositivo en función del ángulo de incidencia de la luz. Esta representación es especialmente importante para entender cómo el sensor responde a la luz que llega desde diferentes direcciones, lo cual es crucial para aplicaciones donde la dirección de la luz puede variar o necesita ser controlada precisamente.

Eje Vertical (Sensibilidad Relativa): Representa la sensibilidad del detector normalizada respecto a su máxima sensibilidad, que es 1.0 en el centro.

Eje Angular: Varía desde 0° en el centro hasta 80° en los extremos, mostrando cómo cambia la sensibilidad del detector a medida que la luz incide desde ángulos más oblicuos respecto a la perpendicular del sensor.

La curva de sensibilidad es simétrica alrededor del 0°, lo que indica que la sensibilidad es igual para ángulos positivos y negativos respecto a la normal.

La sensibilidad máxima se observa cuando la luz incide perpendicularmente al fotodetector (0°). Esto es típico en muchos fotodetectores, donde la configuración óptica está diseñada para ser más eficiente cuando la luz incide directamente.

A medida que el ángulo aumenta, la sensibilidad disminuye rápidamente, lo que muestra que el detector es altamente sensible a los cambios en la dirección de la incidencia de la luz. Esto afectará significativamente en este trabajo.

(Vishay, 2024)

- Led 10 mm blanco alta luminosidad

Se trata de un diodo emisor de luz (LED) que emite luz blanca con una intensidad de iluminación mayor que la de los LED's estándar. Este tipo de LED se usa con aplicaciones donde se requiera una iluminación brillante y eficiente.



Ilustración 9. Led 10 mm blanco de alta luminosidad.

El siguiente diagrama representa una fuente óptica Lambertiana y cómo varía la intensidad luminosa emitida con respecto al ángulo de emisión θ respecto a la normal de la superficie emisora. Este tipo de representación es fundamental para entender el comportamiento de fuentes de luz que siguen la ley de Lambert, una regla esencial en la óptica y la fotometría.

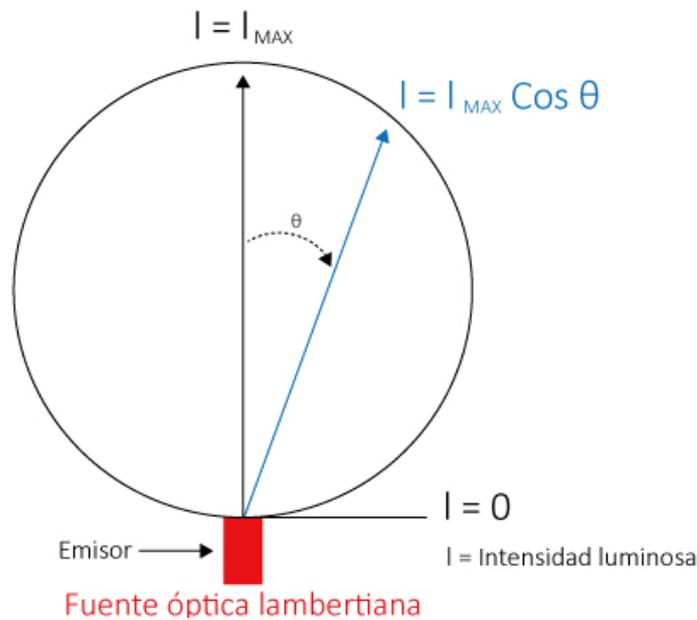


Ilustración 10. Fuente óptica Lambertiana.

La fuente en el diagrama es marcada como el punto de emisión de la luz.

Eje Vertical (Intensidad Luminosa I) es el eje que pasa verticalmente a través de la fuente y representa la dirección de la normal (0°). La intensidad luminosa es máxima (I_{MAX}) cuando la luz se emite directamente a lo largo de esta normal.

El ángulo de Emisión (θ) es el ángulo entre la dirección de emisión de luz y la normal a la superficie de la fuente. Este ángulo es crucial para determinar la intensidad de la luz emitida.

La intensidad de la luz emitida en cualquier ángulo θ se describe por la fórmula

$$I = I_{MAX} \cos \theta \quad (2)$$

Esto indica que la intensidad disminuye con el coseno del ángulo de emisión. En el eje horizontal, donde $\theta=90^\circ$, la intensidad es 0, lo que corresponde a la emisión perpendicular a la normal, donde el coseno de 90° es 0.

El círculo en el diagrama representa una sección transversal del patrón de radiación. La distancia de cualquier punto en este círculo al centro (la fuente) es proporcional a la intensidad luminosa en esa dirección. El círculo completo muestra cómo la intensidad es máxima en la normal y disminuye hasta cero en ángulos de 90° respecto a la normal.

- Analog Discovery 2



Ilustración 11. Analog Discovery 2.

Este dispositivo es un osciloscopio USB que permite a sus usuarios medir, generar, registrar y controlar circuitos de señal mixta de diversas características. Cuenta con entradas y salidas tanto analógicas como digitales.

En este caso, se ha utilizado los dos canales para los dos casos que se estudian en este trabajo. La señal recibida se muestra por la pantalla del Wave Forms, que es el software que generará archivos con medidas, histograma y capturas de pantalla.

Las conexiones de este dispositivo se pueden apreciar en este esquema:

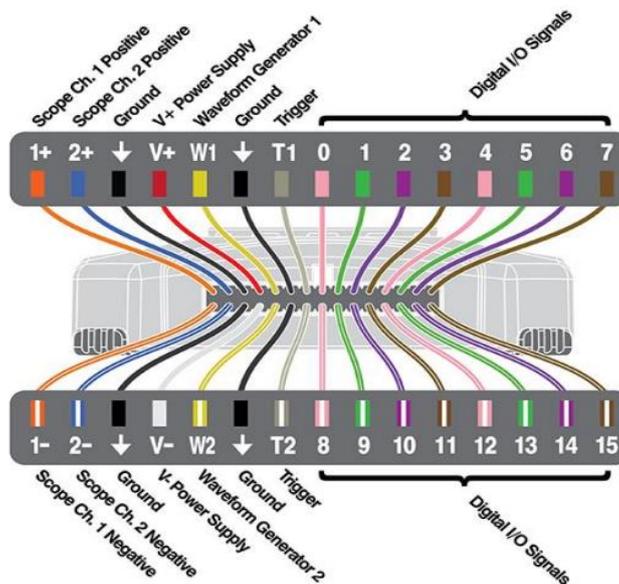


Ilustración 12. Esquema conexiones osciloscopio digital Analog Discovery 2.

2.2. Software

- Arduino IDE

El software de Arduino es fundamental para programar la placa Arduino, y para ello es necesario descargar el IDE (Integrated Development Environment), que proporciona herramientas para escribir, depurar y grabar código. Este entorno facilita el desarrollo de proyectos gracias a su accesibilidad.

Para comenzar a utilizar el IDE, se debe descargar e instalar desde el sitio web oficial de Arduino según el sistema operativo del usuario. Una vez instalado, se abre una interfaz dividida en diferentes áreas: barra de menús, barra de botones, editor de código, barra de mensajes y barra de estado.

El editor de código es donde se trabaja principalmente, y desde allí se pueden realizar acciones como verificar el código, cargarlo en la placa Arduino, crear nuevos sketches, abrir y guardar proyectos.

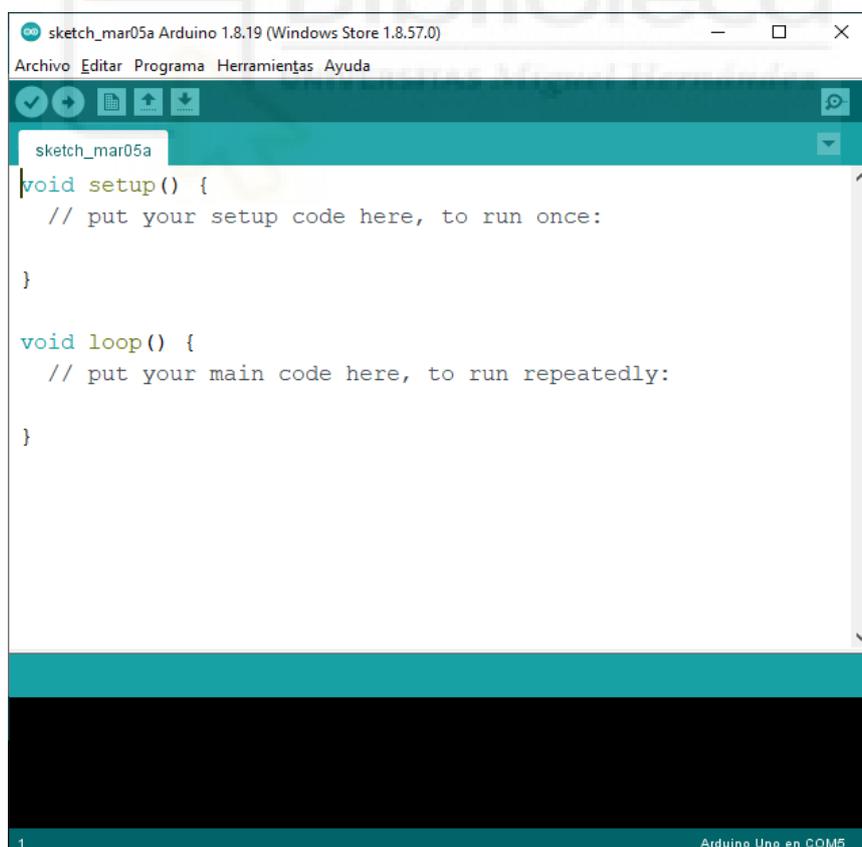


Ilustración 13. Captura de pantalla del entorno de programación de Arduino IDE.

El menú ofrece diversas opciones, como abrir y guardar archivos, compilar el código, importar librerías y herramientas adicionales para formatear el código o grabar un *bootloader* en la placa.

El *monitor serial* es una herramienta importante que permite enviar y recibir datos entre la placa Arduino y el computador a través del cable USB. Es crucial configurar la misma velocidad de transmisión en el monitor serial que en el programa ejecutado en la placa para garantizar la correcta transferencia de datos.



Ilustración 14. Captura de pantalla del monitor serie de Arduino IDE.

(ARDUINO.cl, 2023)

- Waveforms

WaveForms es un software desarrollado por *Digilent*, una empresa líder en herramientas y dispositivos para la educación y la ingeniería. Este programa ofrece una plataforma versátil diseñada para interactuar con una variedad de dispositivos de hardware, como placas de desarrollo FPGA (Field-Programmable Gate Array), placas de desarrollo de microcontroladores e instrumentos de prueba y medición.

Una de las características destacadas de WaveForms es su capacidad para visualizar y controlar señales tanto analógicas como digitales. Los usuarios

pueden capturar y analizar formas de onda, así como controlar la generación de señales de salida, lo que facilita el análisis y la manipulación de datos.

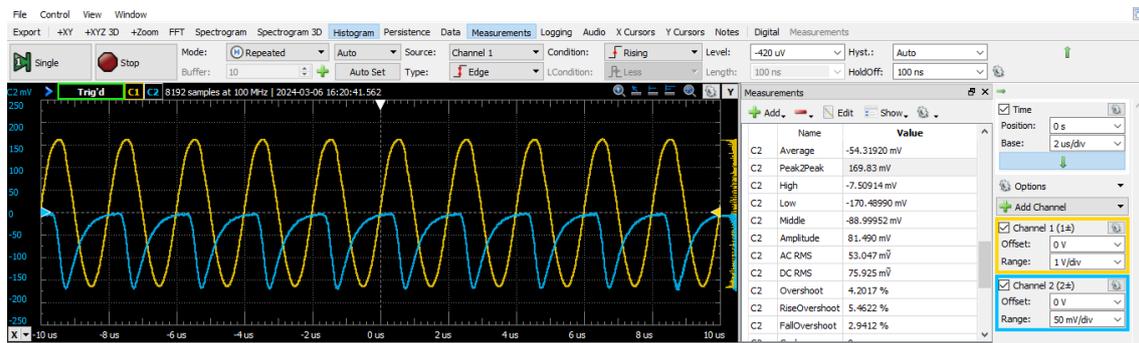


Ilustración 15. Ejemplo de captura de pantalla de medición del Osciloscopio de WaveForms.

Además, WaveForms incluye una amplia gama de instrumentos virtuales, como osciloscopios, generadores de funciones, analizadores de espectro y analizadores de protocolo. Estos instrumentos virtuales ofrecen funcionalidades similares a sus contrapartes físicas, pero dentro del entorno de software de WaveForms, lo que permite un análisis más profundo y una mayor flexibilidad en la configuración.

La interfaz de usuario de WaveForms es intuitiva y fácil de usar, lo que facilita la configuración y el control de los dispositivos conectados. Los usuarios pueden acceder a todas las funciones y herramientas a través de una interfaz gráfica amigable, lo que reduce la curva de aprendizaje y mejora la eficiencia en el uso del software.

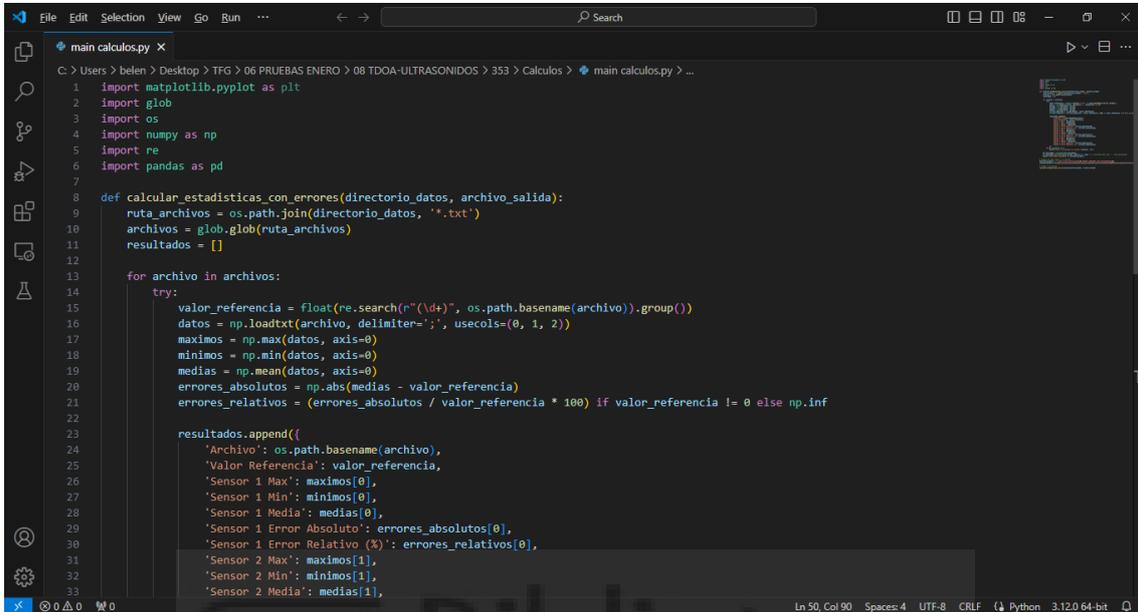
Además, WaveForms es compatible con múltiples plataformas, incluyendo Windows, macOS y Linux, lo que lo hace accesible para una amplia audiencia de usuarios. Esto lo convierte en una herramienta versátil que puede utilizarse en una variedad de entornos y configuraciones.

Por último, WaveForms es una opción popular en entornos educativos, donde se utiliza para enseñar conceptos de electrónica, FPGA y otras áreas relacionadas. Su interfaz intuitiva y sus herramientas poderosas lo convierten en una herramienta invaluable para la enseñanza y el aprendizaje práctico.

- Visual Studio Code

Visual Studio Code (VS Code), creado por Microsoft, es un editor de código abierto que no tiene coste y es compatible con plataformas como Windows,

macOS y Linux. Este editor es popular entre los desarrolladores por su versatilidad en múltiples lenguajes, como Python, y su capacidad para ser personalizado mediante una amplia gama de complementos, temas y ajustes configurables.



```
1 import matplotlib.pyplot as plt
2 import glob
3 import os
4 import numpy as np
5 import re
6 import pandas as pd
7
8 def calcular_estadisticas_con_errores(directorio_datos, archivo_salida):
9     ruta_archivos = os.path.join(directorio_datos, '*.txt')
10    archivos = glob.glob(ruta_archivos)
11    resultados = []
12
13    for archivo in archivos:
14        try:
15            valor_referencia = float(re.search("(\\d+)", os.path.basename(archivo)).group())
16            datos = np.loadtxt(archivo, delimiter=';', usecols=(0, 1, 2))
17            maximos = np.max(datos, axis=0)
18            minimos = np.min(datos, axis=0)
19            medias = np.mean(datos, axis=0)
20            errores_absolutos = np.abs(medias - valor_referencia)
21            errores_relativos = (errores_absolutos / valor_referencia * 100) if valor_referencia != 0 else np.inf
22
23            resultados.append({
24                'Archivo': os.path.basename(archivo),
25                'Valor Referencia': valor_referencia,
26                'Sensor 1 Max': maximos[0],
27                'Sensor 1 Min': minimos[0],
28                'Sensor 1 Media': medias[0],
29                'Sensor 1 Error Absoluto': errores_absolutos[0],
30                'Sensor 1 Error Relativo (%)': errores_relativos[0],
31                'Sensor 2 Max': maximos[1],
32                'Sensor 2 Min': minimos[1],
33                'Sensor 2 Media': medias[1],
34            })
35        except Exception as e:
36            print(f"Error al procesar el archivo {archivo}: {e}")
37
38    return resultados
```

Ilustración 16. Captura de pantalla del entorno Visual Studio Code.

2.3. Fundamentos teóricos

2.3.1. Algoritmo RSSI

El Received Signal Strength Indicator (RSSI) es una medida utilizada para evaluar la potencia de la señal recibida en sistemas de comunicación inalámbrica. Representa la potencia (mW) de la señal recibida en un receptor desde un transmisor, y se expresa generalmente en decibelios (dBm). Los valores de RSSI son fundamentales para determinar la calidad del enlace y estimar la distancia entre nodos en redes inalámbricas de sensores (WSN), WiFi, Bluetooth, y otros sistemas de comunicación.

Características del RSSI:

Medición de Intensidad: RSSI mide la potencia de la señal recibida, lo cual ayuda a determinar la proximidad del transmisor y la calidad de la comunicación.

Comúnmente se expresa en dBm, donde valores más cercanos a 0 dBm indican una señal más fuerte, y valores más negativos indican una señal más débil.

Uso en Estimación de Distancia: RSSI puede ser utilizado para estimar la distancia entre el transmisor y el receptor basado en la atenuación de la señal.

Calidad del Enlace: Un RSSI alto (menos negativo) sugiere un enlace de alta calidad, mientras que un RSSI bajo (más negativo) puede indicar problemas de conectividad.

(Rappaport, 2002)

Para convertir las mediciones de intensidad luminosa en un equivalente de RSSI en un contexto de señales ópticas, se puede seguir un proceso sistemático que se asemeja a la conversión de potencia de señal en comunicaciones inalámbricas.

Paso 1: Establecer una Referencia. Primero, se necesita identificar el valor de intensidad máxima (I_{max}) en las mediciones, que será el valor más alto de voltaje medido en milivoltios cuando el fotodetector esté lo más cerca posible del LED.

Paso 2: Calcular la Intensidad Relativa. Para cada medición de intensidad I obtenida a diferentes distancias en milivoltios, se calcula la intensidad relativa en comparación con la intensidad máxima

Este valor será un número entre 0 y 1, donde 1 significa que la intensidad es igual a la intensidad máxima y valores más bajos indican una menor intensidad.

$$Intensidad\ Relativa = \frac{I}{I_{max}} = \frac{V_{Peak2Peak}}{I_{max}} \quad (3)$$

Este valor estará entre 0 y 1, donde 1 indica que la intensidad de la medición es igual a la intensidad máxima y valores más bajos indican una menor intensidad.

Paso 3: Convertir a RSSI. Para convertir la intensidad relativa a un valor de RSSI, puedes usar una escala logarítmica similar a la que se usa en las señales de radio. La fórmula común para esto en comunicaciones RF es la siguiente:

$$RSSI = 10 * \log_{10}(Intensidad\ Relativa) \quad (4)$$

Esta conversión te dará un RSSI en decibelios, donde valores más altos (más cercanos a cero) indican una mayor intensidad, y valores más bajos (más negativos) indican una menor intensidad. Si I es igual a I_{\max} , RSSI será 0 dB, lo que representa la máxima intensidad.

Paso 4: Calibración y Ajustes. Si es necesario, se puede ajustar la escala del RSSI para que los valores sean más manejables o informativos según la aplicación específica. Por ejemplo, se puede añadir un factor de escala o un offset si es necesario para alinear con otras mediciones o para mejorar la interpretación. Es posible que se deba calibrar el método si se encuentra que la respuesta del fotodetector no es completamente lineal o si hay otros factores como la temperatura o la interferencia lumínica afectando las mediciones.

Paso 5: Implementación en Software. Finalmente, se implementa el cálculo en Python para analizar los datos del osciloscopio. Esto permite automatizar el cálculo del RSSI y usar estos valores para análisis posteriores.

2.3.2. Algoritmo Estimación del Tiempo de Llegada (TOA)

La estimación del tiempo de llegada, ToA, es una técnica de posicionamiento que permite medir la distancia para establecer la ubicación. En un escenario, se utilizan múltiples nodos de referencia (NR) que, en sincronía, permiten localizar un nodo de destino (ND) a través de la triangulación. La localización del nodo destino es posible si se asume que todas las coordenadas de los nodos de referencia son conocidas. En un entorno coplanar (2D), con las posiciones de los NR conocidas, solo tres NR y tres mediciones de distancias ToA son necesarias para localizar el ND. Sin embargo, en un entorno no coplanar (3D), se necesitan cuatro nodos de referencia (NR).

La posición del ND puede determinarse ya sea a través de los NR estableciendo una estación base de referencia, o mediante el propio ND.

ToA, siendo una técnica robusta, presenta ciertos inconvenientes. Requiere que todos los nodos (tanto los NR como los ND) tengan una sincronización precisa.

Un pequeño error en la temporización puede generar un gran error en el cálculo de la distancia (R_i) entre el nodo destino y uno de los nodos de referencia específicos. Además, la señal transmitida debe estar "etiquetada" con un "sello de tiempo" para permitir que los NR determinen el momento en que la señal fue iniciada por el ND. Este sello de tiempo añade complejidad a la señal transmitida y puede ser una fuente adicional de error.

Otro inconveniente es que a menos que se conozcan las características de la señal para permitir que el receptor determine con precisión el tiempo de llegada bajo condiciones de canal pobres, este método requerirá de altas relaciones señal-ruido (SRN), alta relación señal a interferencia, y una limitada distorsión multitrayecto en todos los receptores. La exactitud del método puede mejorarse utilizando información conocida a partir de señales moduladas digitalmente. La posición del ND puede determinarse con precisión si existe una línea de vista completa entre la estación móvil o ND y las demás estaciones base o NR's. No obstante, si no hay línea de vista (NLOS), la señal tomará un trayecto más largo hacia el receptor del NR y el ToA medido generalmente será mayor que el tiempo de llegada de una señal con línea de vista.

(Andrés Felipe García, 2015)

2.3.3. Algoritmo TOF

El algoritmo Time of Flight (TOF) es una técnica crucial en el ámbito de la medición de distancias y la captura de imágenes tridimensionales. Utilizado ampliamente en aplicaciones de escaneo 3D y sensores de proximidad, el algoritmo TOF mide el tiempo que tarda un pulso de luz en viajar desde un emisor hasta un objeto y regresar al receptor. Esta medida de tiempo, conocida como tiempo de vuelo, permite calcular con precisión la distancia entre el sensor y el objeto.

Su funcionamiento consiste en la emisión de un pulso de luz, generalmente emitido por un láser o LED, es dirigido hacia el objeto objetivo. El pulso de luz se refleja en la superficie del objeto. El sensor capta el pulso de luz reflejado. Se

mide el tiempo que tarda el pulso de luz en viajar desde el emisor hasta el objeto y regresar al sensor. Utilizando la velocidad de la luz, se calcula la distancia basada en el tiempo de vuelo medido.

El algoritmo TOF tiene diversas aplicaciones, entre las que destacan escaneo 3D, los sensores de Proximidad y la medición de distancias, que es la aplicación en la que basaremos parte de este trabajo.

Ventajas del Algoritmo TOF

Alta Precisión, ya que es capaz de medir distancias con gran exactitud. Y la rapidez, debido a que la medición es rápida, lo que permite su uso en aplicaciones en tiempo real.

Desventajas del Algoritmo TOF

Es sensible a la luz ambiental, por lo que puede verse afectado por la luz ambiental, aunque los sensores modernos incluyen tecnologías para mitigar este problema. Los sensores TOF pueden ser más costosos que otros métodos de medición de distancia.

(AndroidPhoria, 2023)

2.4. Montaje eléctrico para toma de medidas

En este apartado, se procede a analizar los diversos montajes eléctricos realizados para la toma de medidas en cada caso.

Para una mejor explicación del montaje, se emplea una representación gráfica gracias a la herramienta de Fritzing, la cual permite representar esquemática y visualmente este tipo de circuitos.

2.4.1. Analog Discovery 2 con fotodetector BPW21 y Led comercial (RSSI para estimación de distancias)

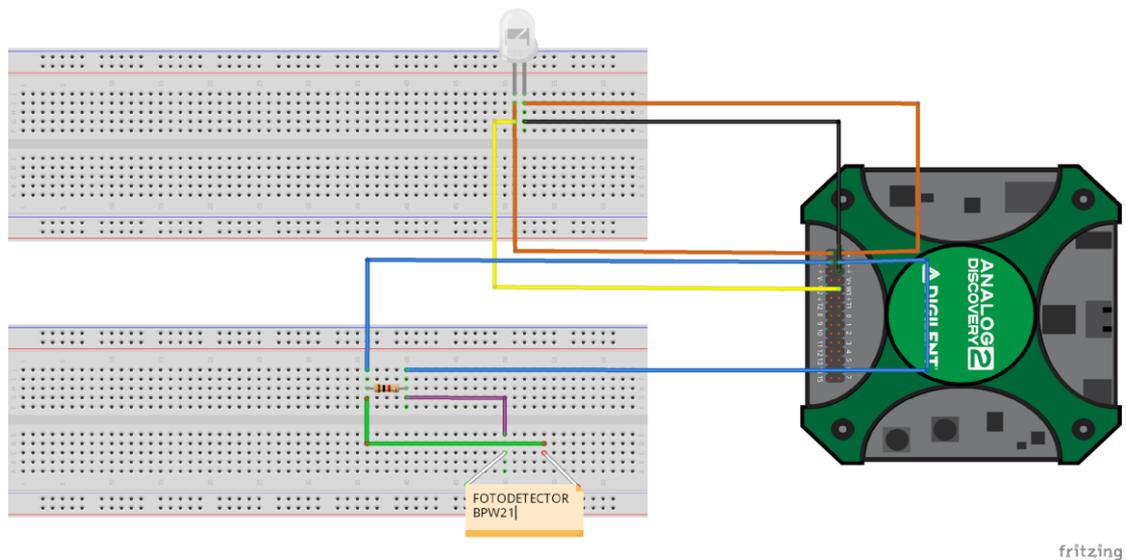


Ilustración 17. Esquema de conexión Analog Discovery 2 con fotodetector BPW21 y LED COMERCIAL.

En este esquema se conecta por un lado la parte de emisión, que en nuestro caso es el diodo LED utilizado para emitir la señal. Este led se alimenta a través del generador de señales W1 del osciloscopio Analog Discovery 2, y la otra pata del LED se conecta a tierra (GND).

Se conecta una resistencia de 1 K Ω con el objetivo de normalizar el valor obtenido. Dado que el osciloscopio tiene una resistencia interna de 1 M Ω , logramos que la corriente circule a través de los bornes del canal 2 del osciloscopio, donde estamos realizando la medición. En el canal 2 (W2) tenemos la conexión del fotodetector BPW21 que es el que se encarga de recibir el voltaje de la señal que el LED emite.

Finalmente, se conectan los canales W1 y W2 al osciloscopio para poder visualizar la señal emitida por el led (W1) y la recibida por el fotodetector BPW21 (W2).

2.4.2. Analog Discovery 2 con tres fotodetectores BPW21 y Led comercial (RSSI para triangulación)

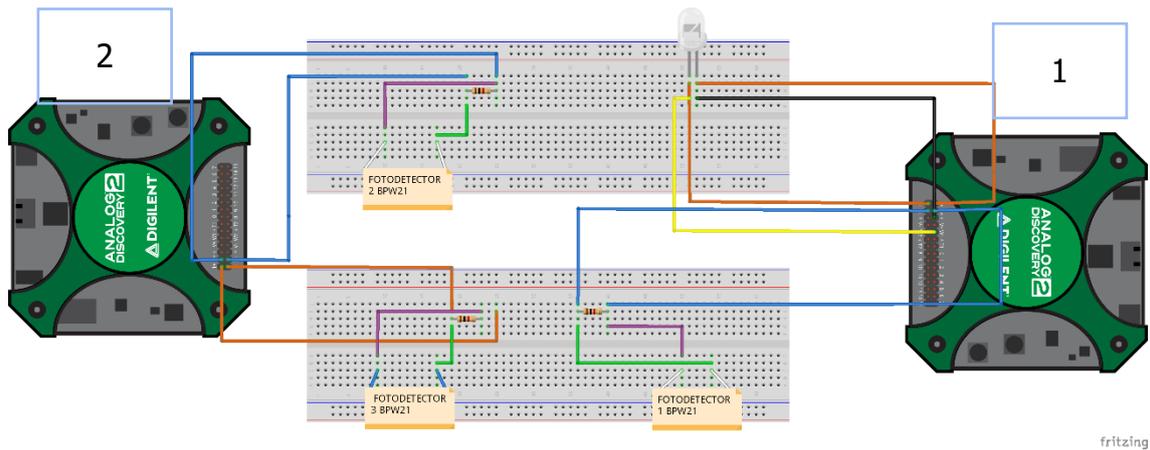


Ilustración 18. Esquema de conexión dos Analog Discovery 2 con tres fotodetectores BPW21

En este esquema se han utilizado dos dispositivos Analog Discovery 2 para analizar las señales emitidas y recibidas por tres fotodetectores BPW21.

Parte de Emisión:

Un diodo LED se utiliza para emitir la señal.

Este LED se alimenta a través del generador de señales W1 del primer osciloscopio Analog Discovery 2.

La otra pata del LED se conecta a tierra (GND).

Una resistencia de $1\text{ K}\Omega$ se conecta en serie con el LED para limitar la corriente.

Parte de Recepción:

Se utilizan tres fotodetectores BPW21 para recibir la señal emitida por el LED.

Cada fotodetector está conectado a uno de los canales de los osciloscopios.

El primer fotodetector BPW21 se conecta al canal W2 del primer osciloscopio Analog Discovery 2.

El segundo fotodetector BPW21 se conecta al canal W1 del segundo osciloscopio Analog Discovery 2.

El tercer fotodetector BPW21 se conecta al canal W2 del segundo osciloscopio Analog Discovery 2.

Conexiones de Medición:

El canal W1 del primer osciloscopio se utiliza para visualizar la señal emitida por el LED.

Los canales W2 del primer osciloscopio y W1 y W2 del segundo osciloscopio se utilizan para medir las señales recibidas por los fotodetectores BPW21.

De esta manera, se pueden comparar las señales emitidas y recibidas, observando la variación de las mismas al pasar por los diferentes fotodetectores. Los tres fotodetectores están dispuestos estratégicamente, separados por 120°, formando una disposición simétrica que permite una captura más precisa y equilibrada de la señal luminosa en diferentes direcciones.

2.4.3. Arduino UNO con sensor de ultrasonidos HC-SR04 (TOA para estimación de distancias)

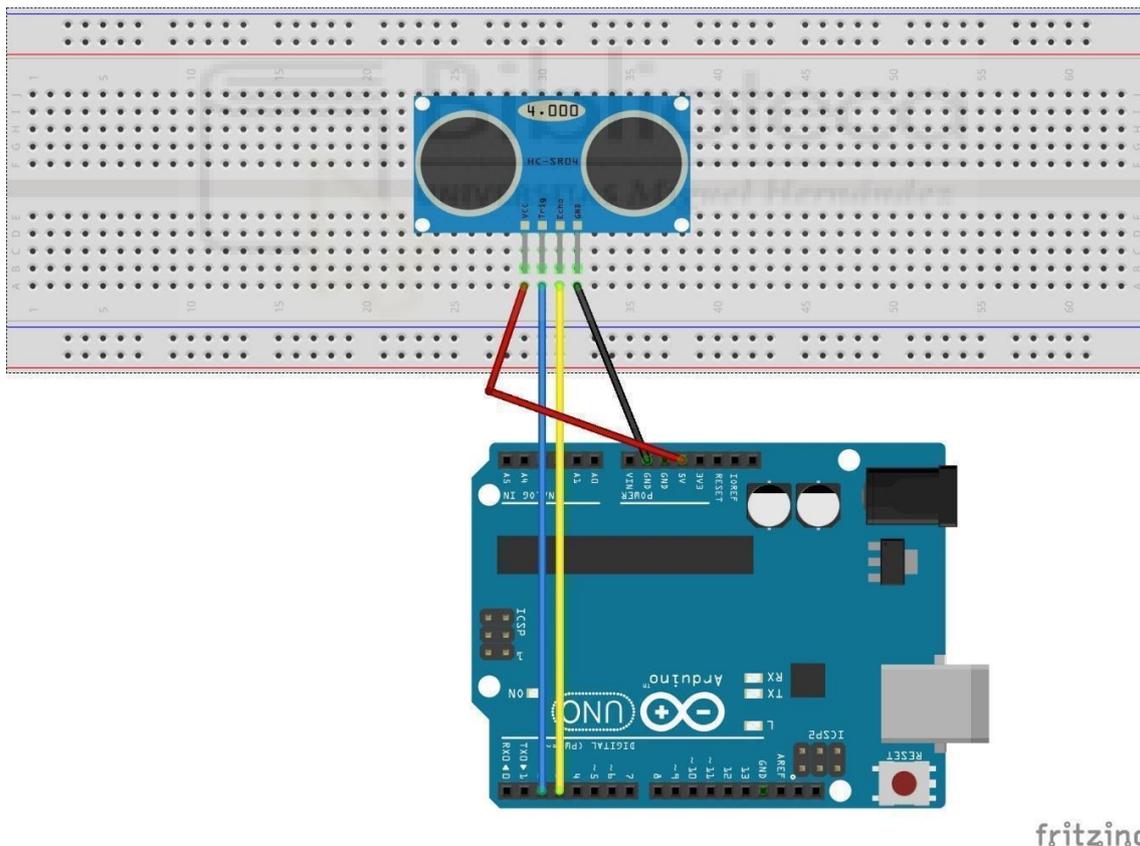


Ilustración 19. Esquema de conexión de Arduino UNO con Sensor HC-SR04.

En esta imagen se muestra un esquema de conexión de un sensor de ultrasonido HC-SR04 con una placa Arduino Uno. El sensor HC-SR04 se utiliza para medir

distancia mediante ultrasonidos. El esquema representa cómo conectar el sensor al Arduino para poder enviar y recibir señales que permitan calcular la distancia a objetos cercanos. Aquí se explica cómo está configurado:

Pines del sensor HC-SR04:

- VCC: Conectado a la línea de +5V del Arduino para alimentar el sensor.
- Trig (Trigger): Conectado al pin digital 9 del Arduino. Este pin se utiliza para enviar un pulso ultrasónico.
- Echo: Conectado al pin digital 10 del Arduino. Este pin recibe el eco del pulso ultrasónico que rebota en los objetos.
- GND (Ground): Conectado a uno de los pines GND del Arduino para completar el circuito.

2.4.4. Arduino Uno con 3 Ultrasonidos (TOA para triangulación)

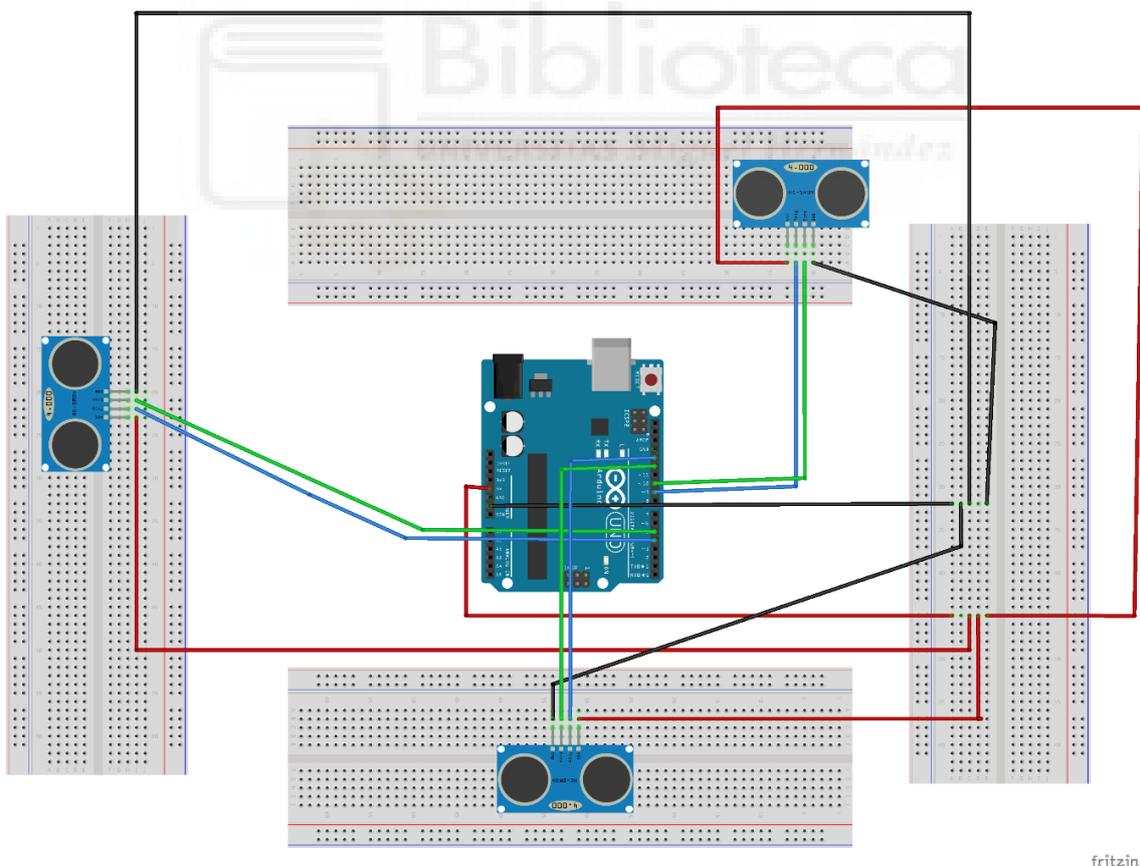


Ilustración 20. Esquema de conexión de Arduino UNO con tres Sensores HC-SR04.

Este esquema parece un poco más complejo que el anterior, ya que involucra múltiples sensores de ultrasonidos HC-SR04 conectados a una placa Arduino. En este caso, cada sensor puede medir distancias independientemente, lo cual es útil en aplicaciones que requieren percepción espacial de un entorno, como la robótica o sistemas de seguridad.

Sensores HC-SR04: cada sensor está conectado similarmente con cuatro pines: VCC, Trig, Echo y GND. Los sensores están distribuidos en diferentes partes de la protoboard. En este caso, la herramienta no permite representar al completo la disposición de nuestro esquema, sólo la distribución de pines.

En este caso, cada sensor debería estar a 120° de separación con respecto al otro, formando entre los 3 un círculo perfecto.

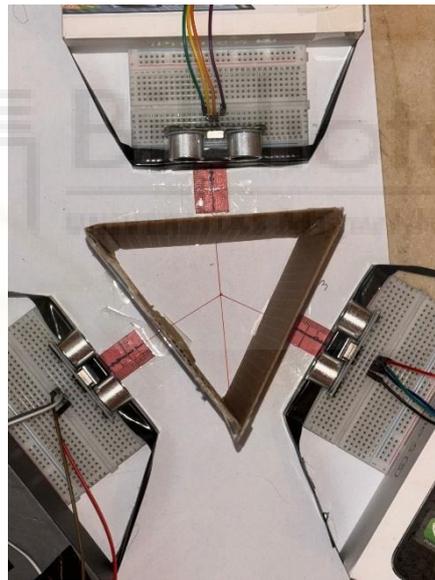


Ilustración 21. Disposición real de los tres sensores HC-SR04.

2.4.5. Arduino Mega 2560 con sensor láser VL6180X (TOF para estimación de distancias)



Ilustración 22. Montaje de conexión Sensor láser VL6180X con Arduino Mega 2560.

En la imagen se observa un montaje de un sensor láser VL6180X conectado a una placa Arduino Mega 2560.

El sensor láser VL6180X encaja perfectamente sobre la placa de Arduino Mega 2560. Es compatible con el diseño del conector Arduino UNO R3, utilizado también en la Arduino Mega 2560.

VCC del sensor VL6180X a 5V de la Arduino Mega 2560.

GND del sensor VL6180X a GND de la Arduino Mega 2560.

SCL (Clock) del sensor VL6180X a SCL (pin 21) de la Arduino Mega 2560.

SDA (Data) del sensor VL6180X a SDA (pin 20) de la Arduino Mega 2560.

El sensor láser se conecta mediante el protocolo I2C. En cuanto al Arduino Mega 2560, se elige este porque ofrece múltiples pines de entrada/salida facilitando la conexión de diversos sensores o actuadores si hiciera falta.

2.5. Programación

2.5.1. Código RSSI con fotodetector BPW21 en Analog Discovery 2.

Para este apartado, se ha utilizado el código desarrollado previamente para la automatización de medidas por un compañero en un Trabajo de Fin de Grado anterior.

(Jorge García Brea, 2022)

2.5.2. Código RSSI tres fotodetectores BPW21 en Analog Discovery 2.

Para este apartado, se ha modificado el código desarrollado previamente por un compañero en un Trabajo de Fin de Grado anterior, adaptándolo para que el segundo Analog Discovery 2 tenga dos canales en recepción.

(Jorge García Brea, 2022)

2.5.3. Código TOA con sensor de ultrasonidos HC-SR04 en Arduino IDE para estimación de distancias.

Este código es un programa de Arduino que utiliza un sensor ultrasónico, gestionado a través de la biblioteca "NewPing", para realizar hasta 100 mediciones de distancia. Se establecen constantes para los pines del sensor (trigPin y echoPin) y la distancia máxima detectable de 100 cm. El programa crea un objeto "NewPing" configurado con estos parámetros.

En la función "setup()", el código inicia la comunicación serial a 9600 bps, lo que permite visualizar los resultados de las mediciones en tiempo real en el monitor serial. Dentro del bucle principal "loop()", se ejecuta un ciclo que se repite hasta

que se alcanzan 100 mediciones. En cada iteración, se introduce una pausa de 50 milisegundos para evitar interferencias entre mediciones consecutivas.

El sensor realiza una medición mediante el método "ping()", que devuelve la duración en microsegundos del sonido reflejado desde un objeto. Esta duración se convierte en una distancia en centímetros utilizando la fórmula:

$$distancia = \left(\frac{duración}{2}\right) * 0.034 \quad (5)$$

donde la duración se divide por dos (para calcular solo el trayecto de ida) y se multiplica por 0.034, que corresponde a la velocidad aproximada del sonido en cm/μs. Esta constante es la que se ha ido modificando en cada una de las pruebas para ver cuál de ellas es la óptima para hacer estas mediciones.

Cada resultado de la distancia calculada se imprime en el monitor serial en formato simple, mostrando sólo el valor de la distancia en centímetros. Además, un contador de mediciones se incrementa en uno tras cada ciclo, terminando el programa cuando se han realizado 100 mediciones. Este programa es ideal para aplicaciones donde se requieren mediciones de distancia precisas y repetitivas, como en sistemas de automatización o robótica.

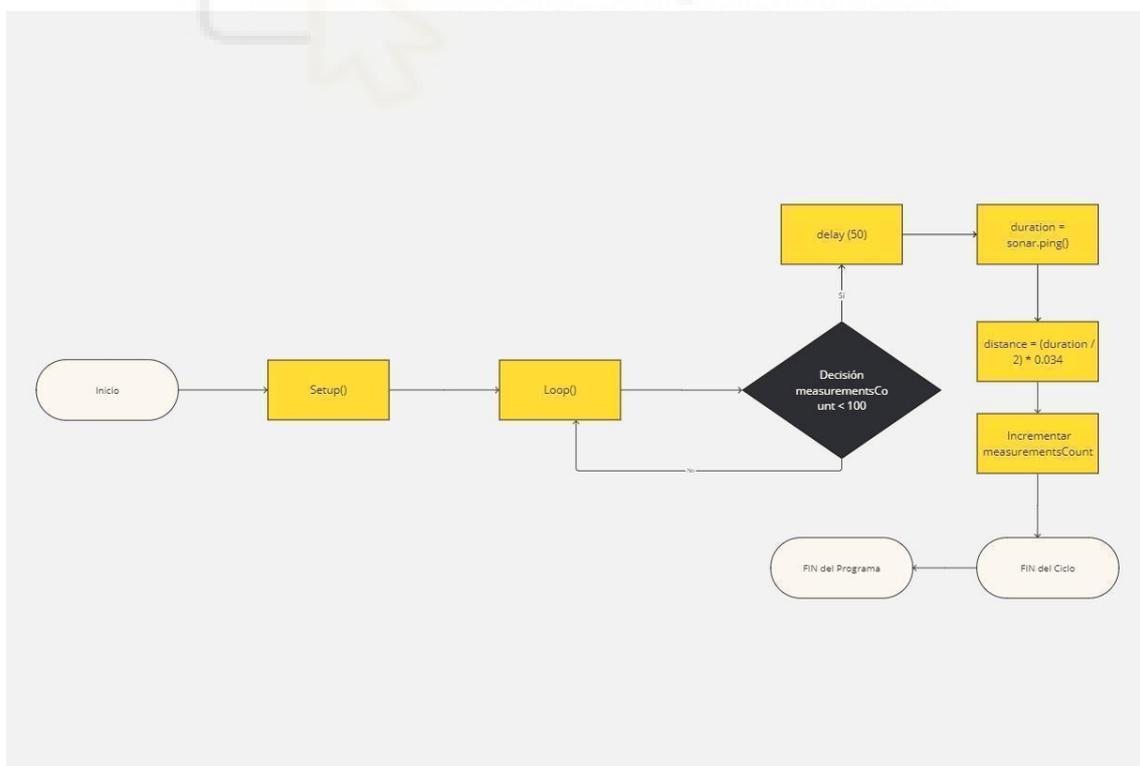


Ilustración 23. Diagrama de Flujo del código de Arduino UNO y un sensor HC-SR04.

2.5.4. Código TOA con sensor de ultrasonidos HC-SR04 en Arduino IDE para triangulación.

Este código utiliza la biblioteca "NewPing" para manejar tres sensores ultrasónicos conectados a un Arduino, destinados a realizar hasta 100 mediciones de distancia. Inicialmente, se incluye la biblioteca "NewPing" y se definen constantes que representan los pines de conexión para los sensores, así como la distancia máxima de detección configurada a 100 cm. Para cada sensor, se instancian objetos "NewPing", asociados a sus respectivos pines de disparo y eco. En la función "setup()", se inicializa la comunicación serial a 9600 bps, lo cual permite la visualización de los datos en el monitor serial. Durante el bucle principal "loop()", el programa verifica si se han realizado menos de 100 mediciones. Si es así, se introduce una pausa de 50 milisegundos entre mediciones para estabilizar las señales y se realizan mediciones de distancia con cada sensor utilizando el método "ping_median()". Este método es útil para obtener una medición más precisa al calcular la mediana de varias mediciones. Las distancias se calculan convirtiendo el tiempo medido en microsegundos a centímetros, utilizando la velocidad del sonido.

Las distancias medidas se imprimen en el monitor serial en el formato "distancia1; distancia2; distancia3", y después de cada ciclo de mediciones, se incrementa el contador de mediciones. Este proceso continúa hasta alcanzar el límite de 100 mediciones. Este enfoque proporciona una forma eficiente y precisa de monitorear las distancias en aplicaciones que requieran múltiples puntos de detección, como en robótica o sistemas de seguridad.

El cálculo de la distancia se hace con

$$distancia = \left(\frac{duración}{2}\right) * 0.034 \quad (6)$$

Calcula la distancia basándose en la duración del eco. El cálculo se hace dividiendo la duración por 2 (para obtener la duración del viaje de ida) y multiplicando por 0.034, que es aproximadamente la velocidad del sonido en cm/ μ s. Esta constante es la que se ha ido modificando en cada una de las pruebas

para ver cuál de ellas es la óptima para hacer estas mediciones. Se ha ido modificando en cada uno de los sensores.

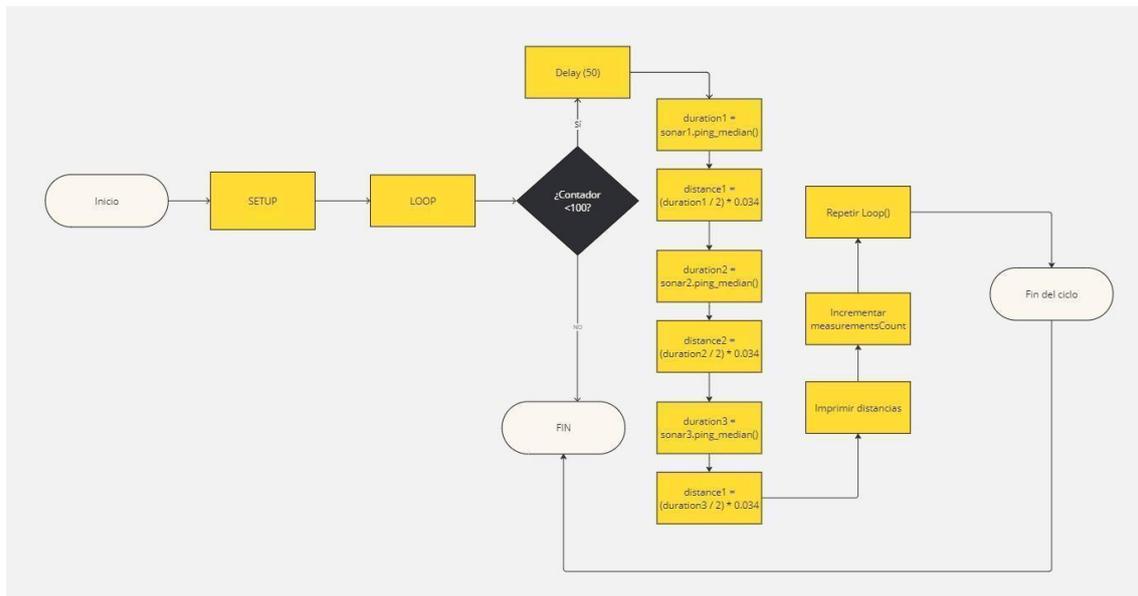


Ilustración 24. Diagrama de Flujo del código de Arduino UNO y tres sensores HC-SR04.

2.5.5. Código TOF con sensor láser VL6180X en Arduino IDE para estimación de distancias.

Este código utiliza la biblioteca "VL6180X" para manejar un sensor de proximidad conectado a un Arduino. El objetivo es realizar hasta 100 mediciones de distancia, registrando y analizando los tiempos de respuesta de las señales.

Para comenzar, se incluyen las librerías necesarias: Wire.h y VL6180X.h. La primera permite la comunicación I2C, esencial para interactuar con el sensor VL6180X, mientras que la segunda proporciona las funciones necesarias para inicializar y leer datos del sensor.

Se declara una instancia del objeto VL6180X para interactuar con el sensor, y se define una variable "measurementsCount" para contar el número de mediciones realizadas, con un límite de 100.

En la función "setup()", se configuran los parámetros iniciales del sistema. Se inicializa la comunicación serial a 9600 baudios para permitir la visualización de los datos en el monitor serial. También se inicia la comunicación I2C con

"Wire.begin()", se inicializa el sensor VL6180X mediante "sensor.init()", se configuran los parámetros por defecto del sensor con "sensor.configureDefault()", y se establece un tiempo máximo de espera de 1000 ms para la lectura del sensor mediante "sensor.setTimeout(1000)".

En la función "loop()", el código realiza el proceso de medición y registro de datos hasta alcanzar 100 mediciones. Durante cada iteración, se siguen varios pasos clave:

Primero, se registra el tiempo inicial en microsegundos utilizando la función "micros()". Esto se hace antes de enviar la señal de medición para tener un punto de referencia temporal.

A continuación, se mide la distancia en milímetros utilizando el método "sensor.readRangeSingle()". Esta función envía un pulso ultrasónico y mide el tiempo que tarda en regresar, calculando así la distancia.

Después de obtener la distancia, se registra nuevamente el tiempo en microsegundos para calcular el tiempo total de viaje de la señal. Este tiempo se almacena en la variable "end_time".

El siguiente paso es calcular el tiempo de viaje de la señal, restando "start_time" de "end_time". Este valor representa el tiempo de ida de la señal en microsegundos.

Para facilitar la interpretación de los datos, se convierte el tiempo de viaje de microsegundos a milisegundos dividiendo por 1000. Luego, se convierte la distancia de milímetros a centímetros dividiendo por 10.

Se calcula el tiempo de ida y vuelta (Round Trip Time o RTT) multiplicando el tiempo de viaje en milisegundos por 2. Esto proporciona una medida completa del tiempo que tarda la señal en viajar hasta el objeto y regresar.

Los resultados de la medición, es decir, la distancia en centímetros y el RTT en milisegundos, se imprimen en el monitor serie para su análisis. Si ocurre un tiempo de espera durante la medición, se imprime un mensaje de "TIMEOUT"

para indicar que el sensor no pudo completar la medición dentro del tiempo establecido.

Finalmente, se incrementa el contador de mediciones "measurementsCount". Si se alcanzan 100

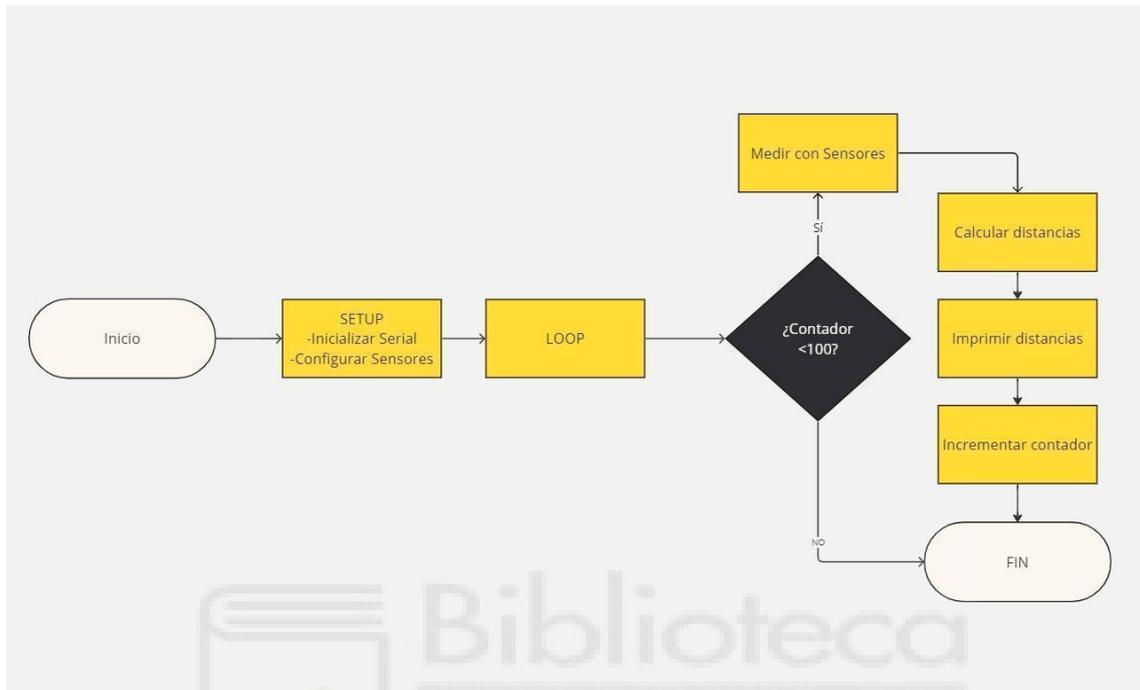


Ilustración 25. Diagrama de Flujo del código de Arduino UNO y tres sensores HC-SR04.



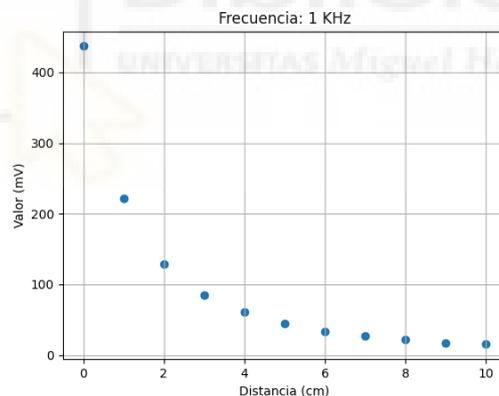
3. RESULTADOS

Para estos resultados hay que tener en cuenta que:

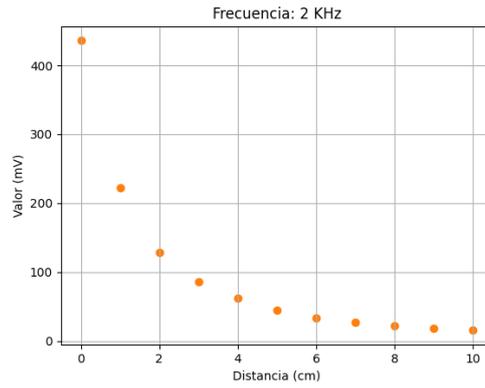
- Para RSSI con estimación de distancias o con triangulación usando leds y fotodetectores, podemos variar la distancia y la frecuencia.
- Para TOA con estimación de distancias o triangulación y haciendo uso de un sensor de ultrasonidos HC-SR04, se puede cambiar la distancia y la constante de la velocidad del sonido, manteniendo la misma frecuencia.
- Para TOF con estimación de distancias únicamente y usando un sensor láser VL6180X, se cambia la distancia.

3.1. Algoritmo RSSI estimación distancia led comercial y fotodetector BPW21

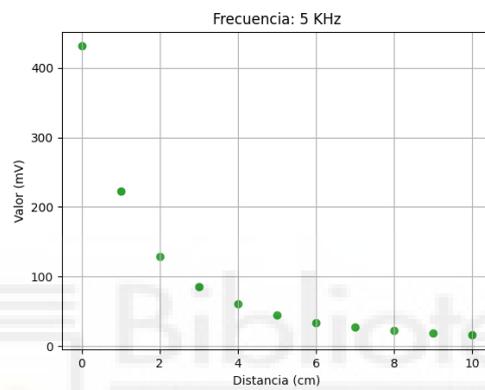
Para el estudio del algoritmo en este caso, se ha dispuesto un fotodetector BPW21 y un led comercial y se fue variando la distancia del fotodetector con respecto al led y la frecuencia, obteniendo los siguientes resultados.



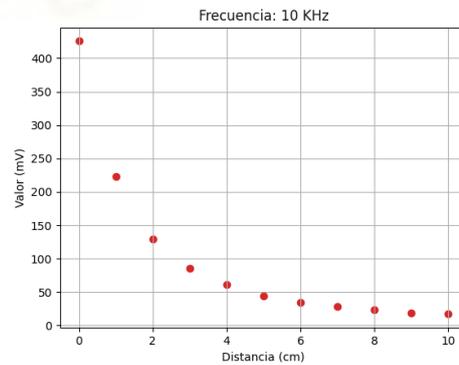
Gráfica 1. Comparación del voltaje Peak2Peak VS Distancia para 1 kHz.



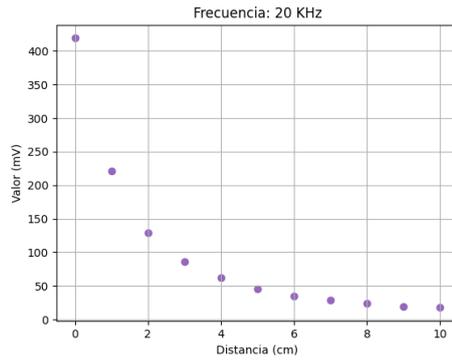
Gráfica 2. Comparación del voltaje Peak2Peak VS Distancia para 2 kHz.



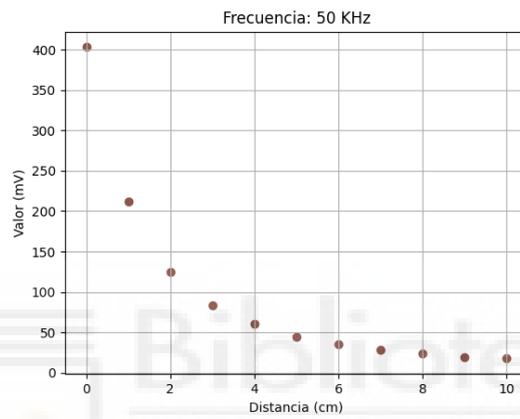
Gráfica 3. Comparación del voltaje Peak2Peak VS Distancia para 5 kHz.



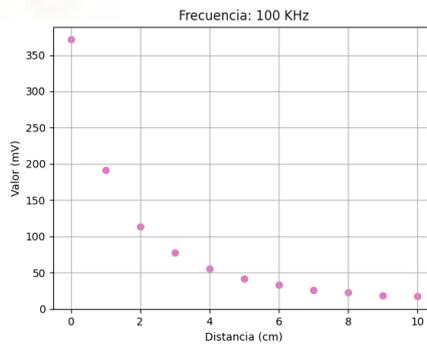
Gráfica 4. Comparación del voltaje Peak2Peak VS Distancia para 10 kHz.



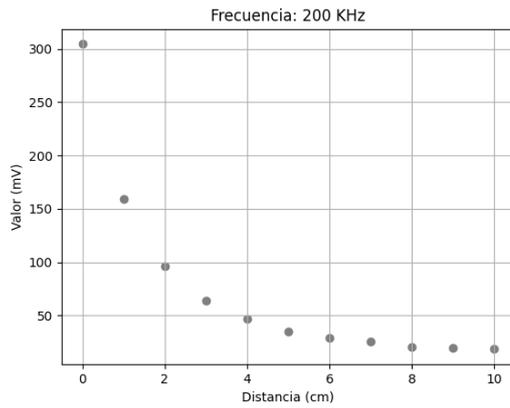
Gráfica 5. Comparación del voltaje Peak2Peak VS Distancia para 20 kHz.



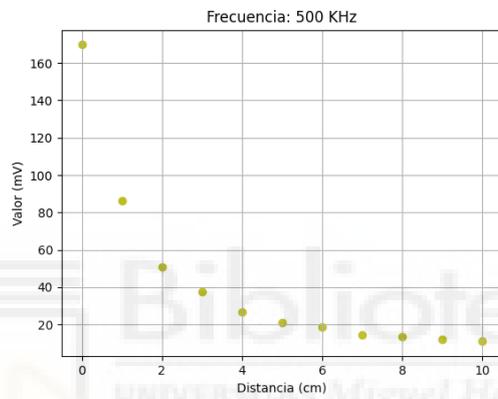
Gráfica 6. Comparación del voltaje Peak2Peak VS Distancia para 50 kHz.



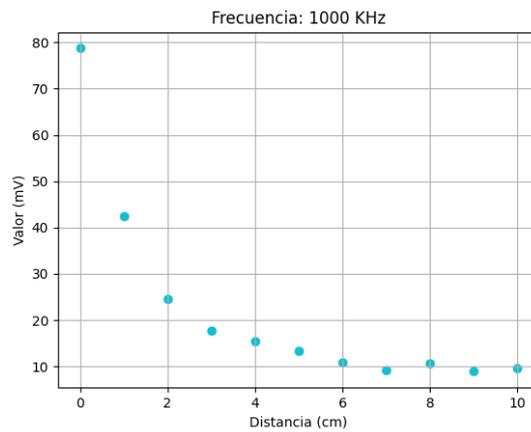
Gráfica 7. Comparación del voltaje Peak2Peak VS Distancia para 100 kHz.



Gráfica 8. Comparación del voltaje Peak2Peak VS Distancia para 200 KHz.



Gráfica 9. Comparación del voltaje Peak2Peak VS Distancia para 500 KHz.



Gráfica 10. Comparación del voltaje Peak2Peak VS Distancia para 1000 KHz.

- Gráfica 1 kHz

La señal Peak to Peak es máxima (más de 400 mV) a una distancia de 0 cm y disminuye rápidamente a medida que la distancia aumenta, estabilizándose cerca de 0 mV más allá de los 10 cm.

A esta frecuencia, la señal se atenúa fuertemente y se reduce casi a cero a mayores distancias, indicando una señal más limpia con menos ruido.

- Gráfica 2 kHz

Similar a la gráfica de 1 kHz, pero la atenuación parece menos pronunciada. La señal Peak-to-Peak inicial también es alta (más de 400 mV) y disminuye a menos de 10 mV a 10 cm.

La frecuencia más alta aún se atenúa rápidamente con la distancia, pero parece mantener un poco más de intensidad que a 1 kHz a mayores distancias.

La señal se atenúa rápidamente, manteniendo niveles bajos de ruido a mayores distancias.

- Gráfica 5 kHz

La señal inicial es alta (más de 400 mV) y se atenúa, estabilizándose en valores muy bajos (cerca de 0 mV) después de 10 cm.

La señal se reduce significativamente, con poco ruido detectado a mayores distancias.

- Gráfica 10 kHz

Nuevamente, la señal es alta al inicio y se atenúa rápidamente, manteniéndose en valores bajos a distancias mayores.

La señal muestra una disminución rápida, con poco ruido a distancias mayores.

- Gráfica 20 kHz

La señal inicial es alta y sigue una tendencia similar de atenuación rápida.

La señal aún muestra una rápida atenuación y bajos niveles de ruido a mayores distancias.

- Gráfica 50 kHz

La señal inicial sigue siendo alta, pero la atenuación parece menos pronunciada que en frecuencias más bajas.

La señal se atenúa, pero no tanto, con algo de ruido detectado a mayores distancias.

- Gráfica 100 kHz

La señal inicial es alta (aproximadamente 250 mV) y se atenúa gradualmente.

La señal se mantiene más fuerte a mayores distancias, pero el aumento del valor del voltaje Peak2Peak puede estar relacionado con el ruido.

- Gráfica 200 kHz

La señal inicial es alrededor de 300 mV y se atenúa, pero mantiene valores más altos a mayores distancias.

A esta frecuencia, la señal muestra un nivel mayor de ruido a distancias mayores.

- Gráfica 500 kHz

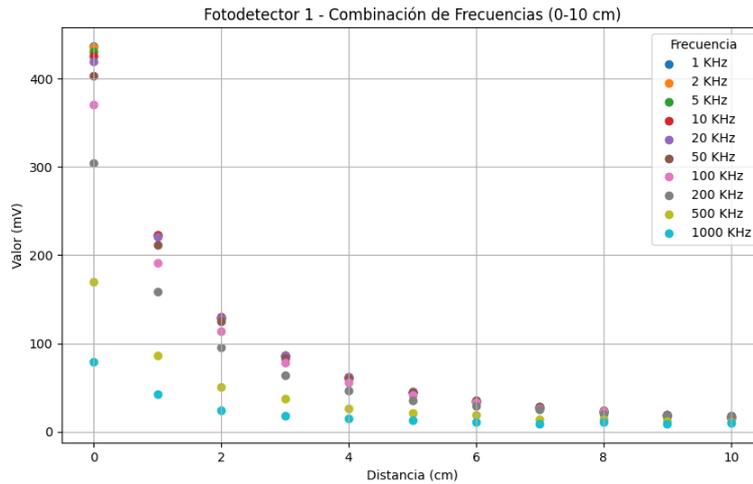
La señal inicial es alta (aproximadamente 160 mV) y se atenúa menos pronunciadamente.

La señal muestra una atenuación menor, pero con más ruido a mayores distancias.

- Gráfica 1000 kHz

La señal inicial es aproximadamente de 80 mV y muestra una atenuación menor a lo largo de la distancia.

A esta frecuencia, la señal tiene una menor atenuación, pero los picos de voltaje son en su mayoría ruido captado por el osciloscopio a mayores distancias.



Gráfica 11. Combinación de valores RSSI estimación de distancias.

Tabla 1. Tabla valores RSSI estimación de distancias en mV.

f (kHz)	1 kHz	2 kHz	5 kHz	10 kHz	20 kHz	50 kHz	100 kHz	200 kHz	500 kHz	1000 kHz
0	436.90	438.87	431.08	425.60	419.09	403.00	371.16	304.39	169.83	78.751
1	222.22	222.90	223.24	222.56	221.19	211.94	191.4	158.87	86.284	42.457
2	128.40	1299.08	129.43	129.77	129.43	124.97	113.68	95.871	51.017	24.653
3	85.257	28.942	85.942	86.284	86.626	83.202	77.724	63.686	37.664	17.808
4	60.947	61.631	61.631	61.631	61.974	60.262	56.153	46.908	26.707	15.408
5	44.196	44.512	44.512	44.854	45.196	44.512	41.772	35.267	21.229	13.353
6	33.555	33.555	33.897	34.24	35.267	35.267	33.212	29.446	18.832	10.957
7	26.707	26.707	27.256	28.077	28.761	28.077	26.365	25.337	14.381	9.2447
8	21.913	22.256	22.256	23.283	23.625	24.31	22.941	20.201	19.696	10.614
9	17.462	18.147	18.487	19.174	19.517	19.517	19.176	19.517	12.326	8.9023
10	15.408	15.408	16.093	17.12	17.462	17.805	17.462	18.489	10.957	9.5871

Frecuencias bajas (1 kHz - 20 kHz), muestran una alta atenuación de la señal con la distancia, con niveles bajos de ruido a distancias mayores.

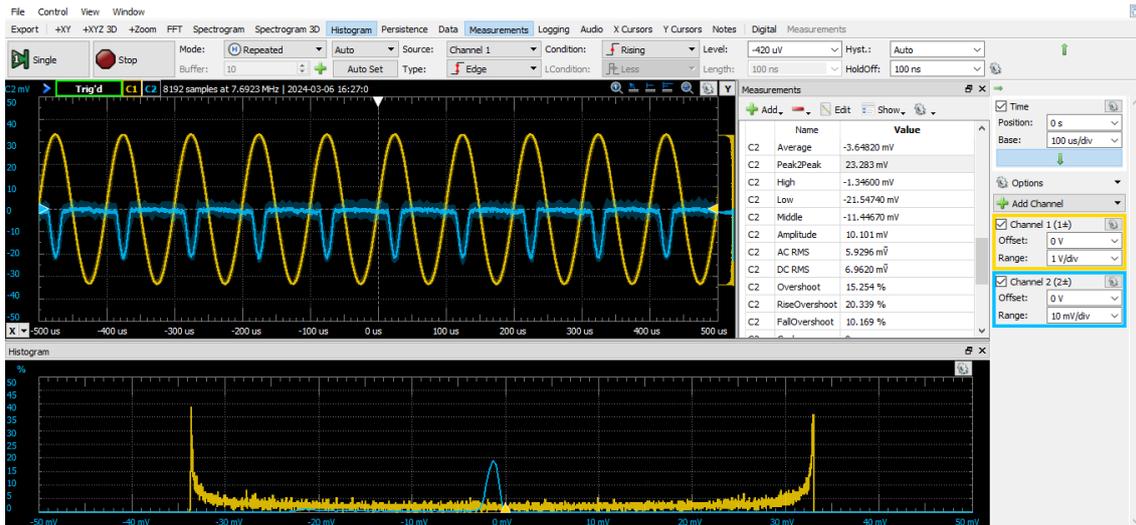


Ilustración 26. Imagen Osciloscopio a 8 cm y 10 kHz en amarillo la señal transmitida y en azul la señal recibida.

Frecuencias intermedias (50 kHz - 200 kHz), a medida que la frecuencia aumenta, el nivel de señal baja sin una gran variación en función de la distancia, pero aumenta el nivel de ruido detectado.

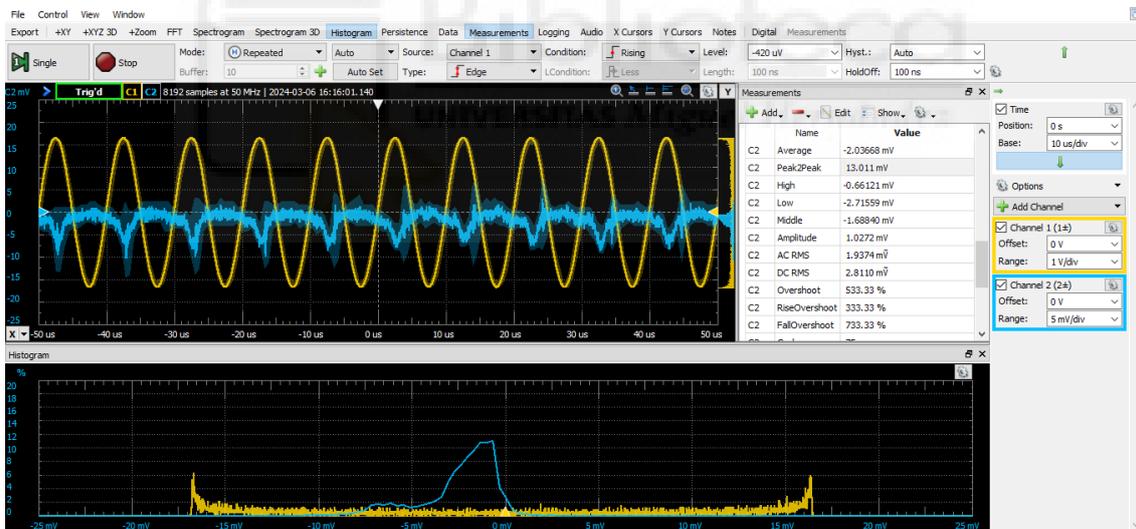


Ilustración 27. Imagen Osciloscopio a 10 cm y 100 kHz, en amarillo la señal transmitida y en azul la señal recibida.

Frecuencias altas (500 kHz - 1000 kHz), el nivel de señal es similar a las frecuencias medias, pero es debido al efecto que produce el ruido, siendo imposible discernir la onda recibida (parece una señal completamente aleatoria).

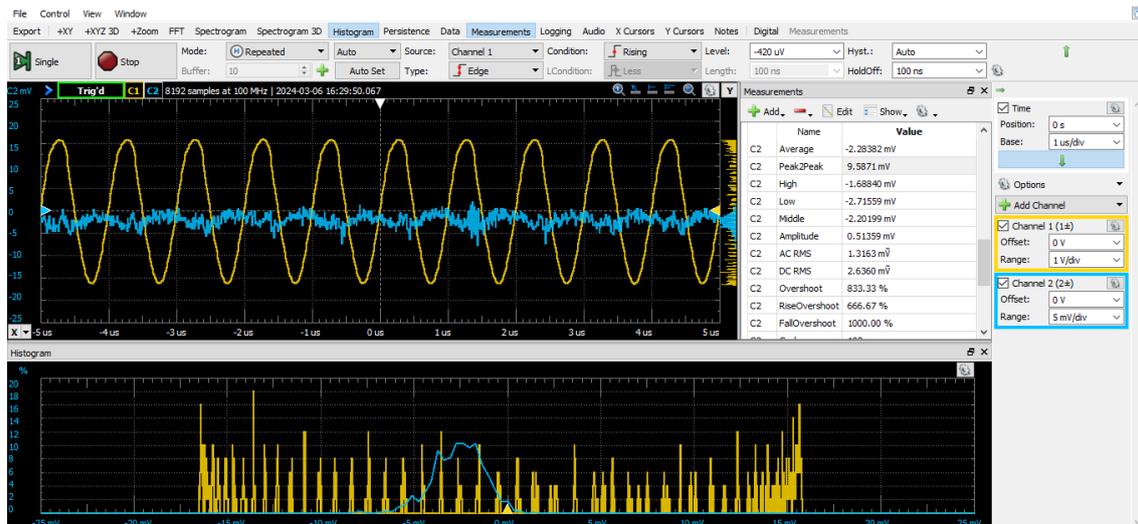
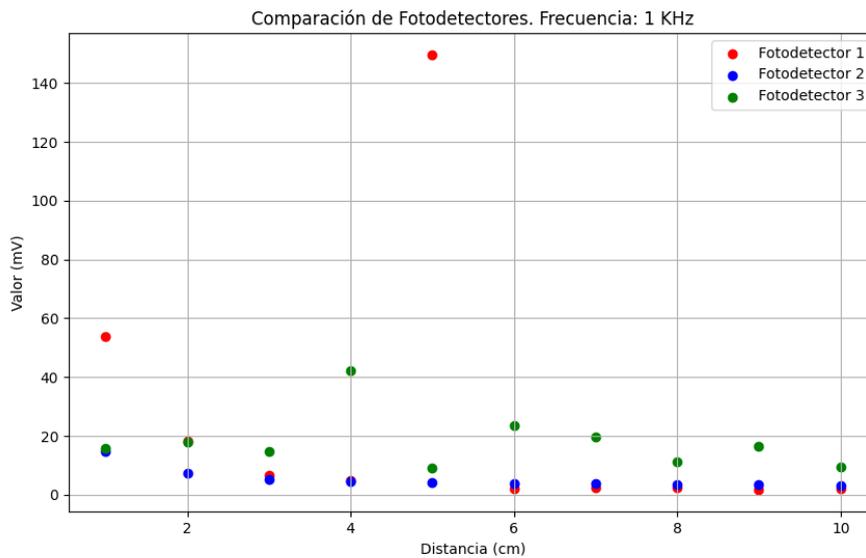


Ilustración 28. Imagen Osciloscopio a 10 cm y 1000 kHz, en amarillo la señal transmitida y en azul la señal recibida.

Estas observaciones indican que, aunque las frecuencias más altas muestran menor atenuación de la señal, el aumento en el ruido a mayores distancias sugiere que la señal útil se ve afectada por interferencias.

3.2. Algoritmo RSSI triangulación con led comercial y tres fotodetectores BPW21

Para el estudio del algoritmo en este caso, se han dispuesto tres fotodetectores BPW21 separados 120° y un led comercial en el centro. A continuación, se fue variando las distancias de los fotodetectores con respecto al led y la frecuencia, obteniendo los siguientes resultados.

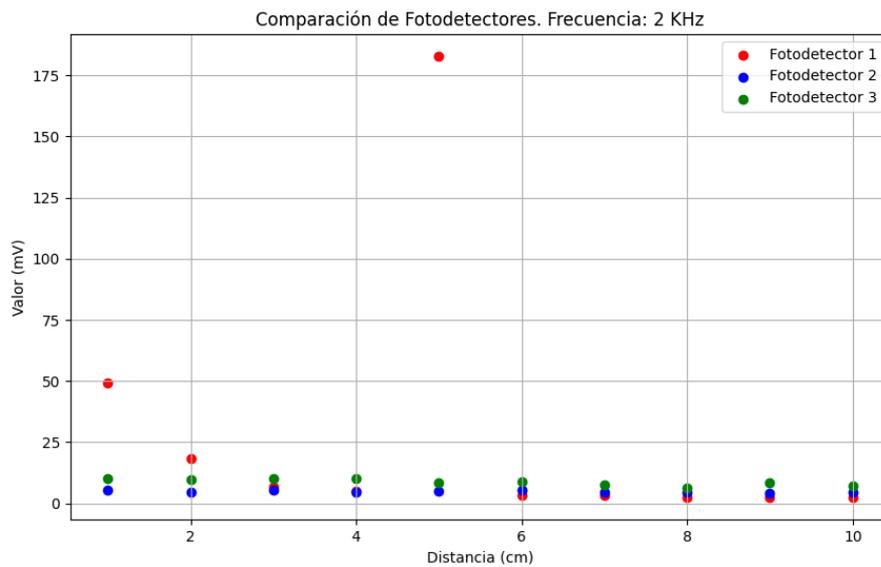


Gráfica 12. Comparación del voltaje Peak2Peak VS Distancia con tres fotodetectores para 1 kHz.

Fotodetector 1: Exhibe picos altos y variabilidad errática, especialmente alrededor de los 5 cm, indicando alta sensibilidad al ruido o cambios en el entorno. La tendencia general es una disminución después de ciertos picos.

Fotodetector 2: Muestra una disminución constante y uniforme de valores con la distancia, sugiriendo mejor consistencia y menor sensibilidad al ruido.

Fotodetector 3: Tiene alta variabilidad y un pico notable a 4 cm, indicando sensibilidad a ciertas condiciones de medición. La dispersión de valores sugiere inconsistencia y mayor sensibilidad en ciertas distancias.

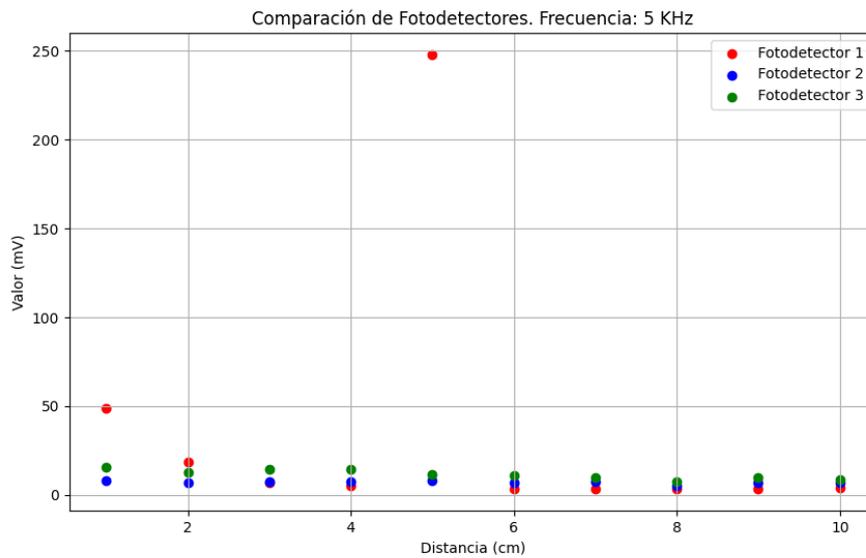


Gráfica 13. Comparación del voltaje Peak2Peak VS Distancia con tres fotodetectores para 2 kHz.

Fotodetector 1: A distancias cortas (2.5-5 cm), los voltajes son altos, alcanzando picos de hasta 175 mV, pero disminuyen drásticamente a casi 0 mV con el aumento de la distancia.

Fotodetector 2: Los voltajes son bajos, comenzando alrededor de 5.4 mV y decreciendo ligeramente con la distancia, mostrando una tendencia decreciente constante.

Fotodetector 3: Los voltajes iniciales son más altos que en el fotodetector 2, comenzando cerca de 10 mV. Aunque hay una tendencia decreciente similar al fotodetector 2, la dispersión de los valores es más evidente.

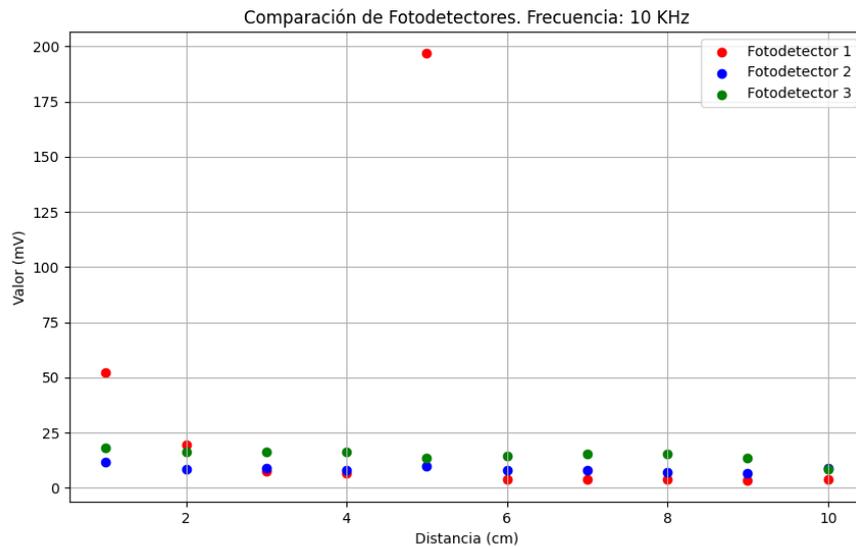


Gráfica 14. Comparación del voltaje Peak2Peak VS Distancia con tres fotodetectores para 5 kHz.

Fotodetector 1: Los voltajes son significativamente altos a distancias cortas, alcanzando picos de hasta 250 mV, pero disminuyen drásticamente con el aumento de la distancia.

Fotodetector 2: Los voltajes son considerablemente más bajos que en el fotodetector 1, iniciando en aproximadamente 8 mV y decreciendo ligeramente con la distancia.

Fotodetector 3: Muestra una caída gradual de los valores de milivoltios a medida que aumenta la distancia, con una leve variabilidad.

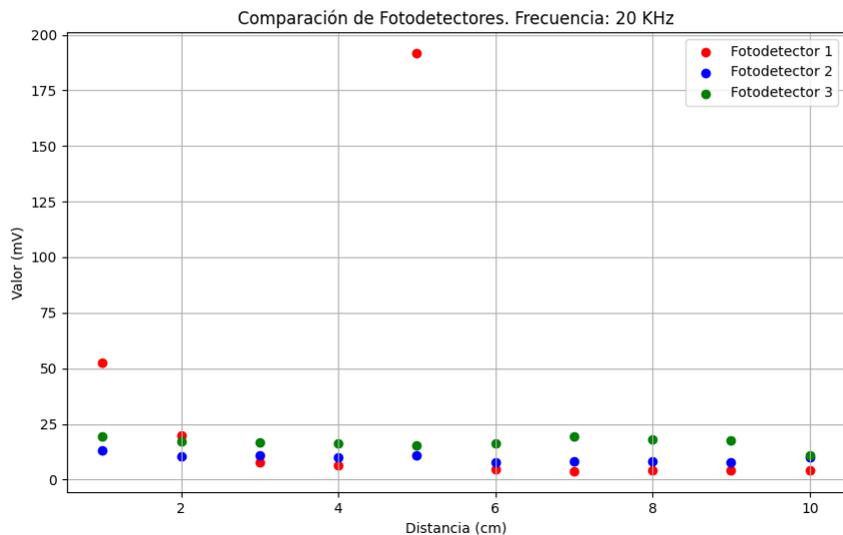


Gráfica 15. Comparación del voltaje Peak2Peak VS Distancia con tres fotodetectores para 10 kHz.

Fotodetector 1: Exhibe picos altos y gran variabilidad, con comportamiento errático especialmente alrededor de los 5 cm. A distancias mayores, los valores son bajos y estables, pero la inconsistencia sugiere alta sensibilidad al ruido o cambios en el entorno.

Fotodetector 2: Tiene menor sensibilidad inicial comparado con el fotodetector 1, con valores de voltaje que decrecen gradualmente con la distancia. La tendencia es decreciente con cierta variabilidad.

Fotodetector 3: Muestra una caída gradual de los milivoltios con la distancia, con leve variabilidad. La tendencia es estable sin picos significativos, indicando buena sensibilidad y consistencia en las mediciones.

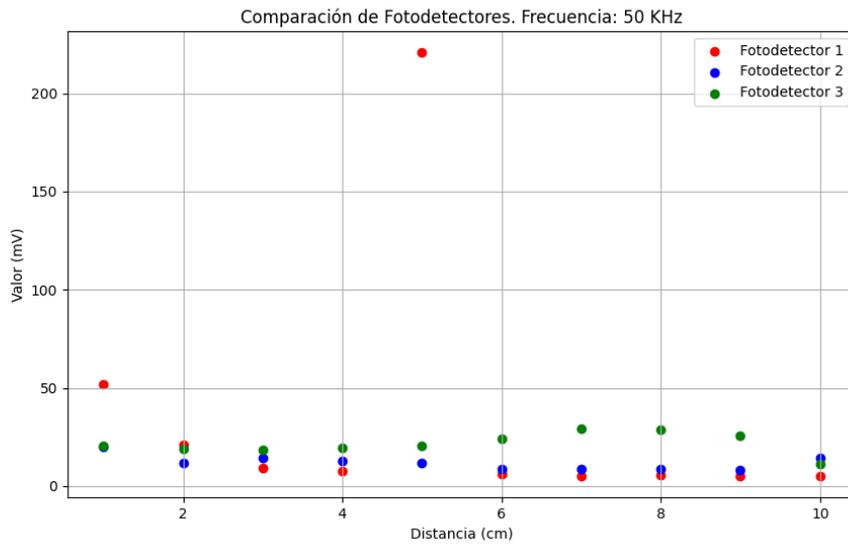


Gráfica 16. Comparación del voltaje Peak2Peak VS Distancia con tres fotodetectores para 20 kHz.

Fotodetector 1: Exhibe picos muy altos y gran variabilidad, con comportamiento errático especialmente alrededor de los 5 cm. A distancias mayores, los valores son bajos, casi 0, sugiriendo alta sensibilidad al ruido o cambios en el entorno.

Fotodetector 2: Los valores de voltaje comienzan cerca de 13 mV a distancias cortas y decrecen significativamente con la distancia, llegando a alrededor de 8 mV a 10 cm. La tendencia decreciente es clara, aunque con alguna dispersión en los datos.

Fotodetector 3: Los valores iniciales de voltaje son más altos, comenzando cerca de 18 mV, y disminuyen a alrededor de 12 mV a 10 cm. Aunque hay variabilidad en los datos, la tendencia decreciente es clara.

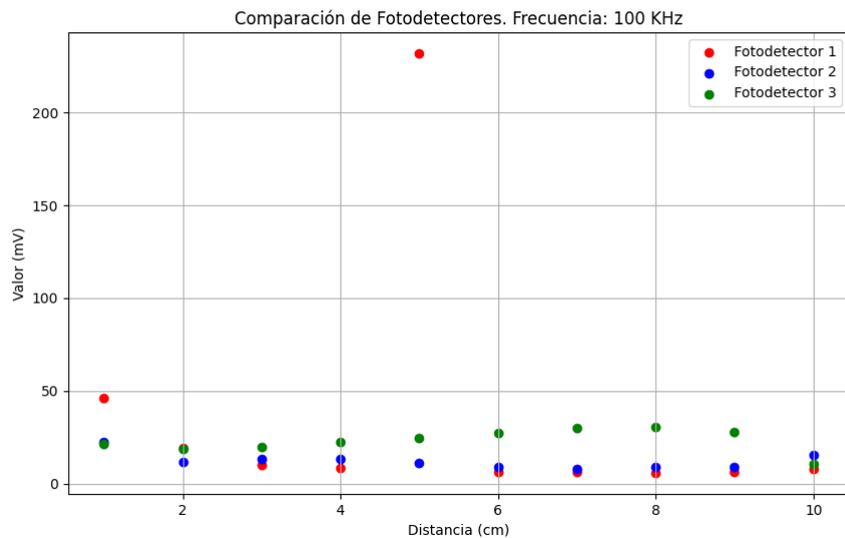


Gráfica 17. Comparación del voltaje Peak2Peak VS Distancia con tres fotodetectores para 50 kHz.

Fotodetector 1: Muestra alta sensibilidad a distancias cortas con picos de voltaje significativos que disminuyen rápidamente con la distancia. A mayores distancias, el voltaje se estabiliza en niveles más bajos, aunque con algunos picos aislados.

Fotodetector 2: Presenta una disminución gradual de los valores de voltaje desde niveles iniciales altos (aproximadamente 20 mV) hasta niveles más bajos (sobre los 8 mV) a 10 cm. La tendencia es claramente decreciente con ligera dispersión en los datos.

Fotodetector 3: Inicia con los valores de voltaje más altos (sobre los 30 mV) y muestra una disminución constante a medida que aumenta la distancia. Hay una notable caída a 10 cm, con dispersión moderada en los datos.

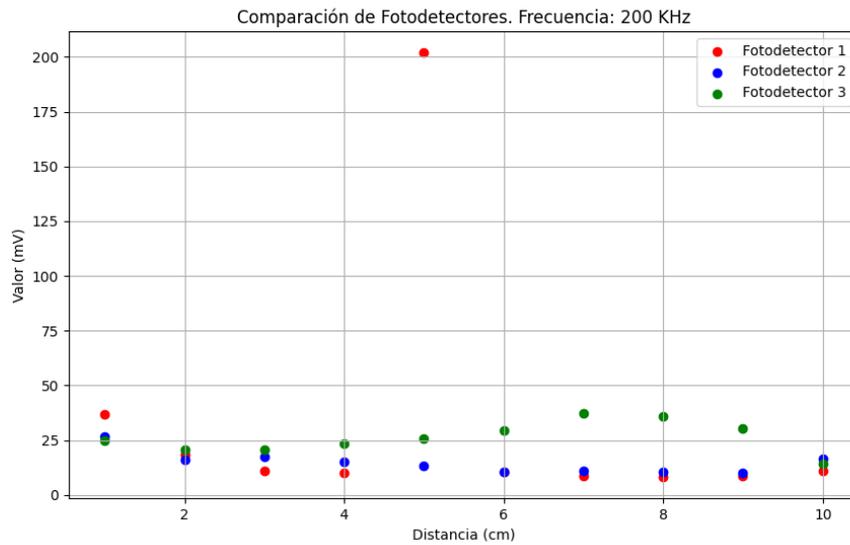


Gráfica 18. Comparación del voltaje Peak2Peak VS Distancia con tres fotodetectores para 100 kHz.

Fotodetector 1: Exhibe picos muy altos y gran variabilidad, con comportamiento errático especialmente alrededor de los 5 cm. A distancias mayores, los valores son bajos y estables, pero la inconsistencia sugiere alta sensibilidad al ruido o cambios en el entorno.

Fotodetector 2: Muestra una disminución constante y uniforme de los valores medidos a medida que la distancia aumenta. La estabilidad sugiere mejor rendimiento en términos de consistencia y menor sensibilidad al ruido en comparación con los otros dos fotodetectores.

Fotodetector 3: Presenta una variabilidad significativa en los valores de milivoltios a medida que aumenta la distancia. Los valores aumentan hasta cierta distancia antes de disminuir, indicando sensibilidad variable dependiendo de la distancia.

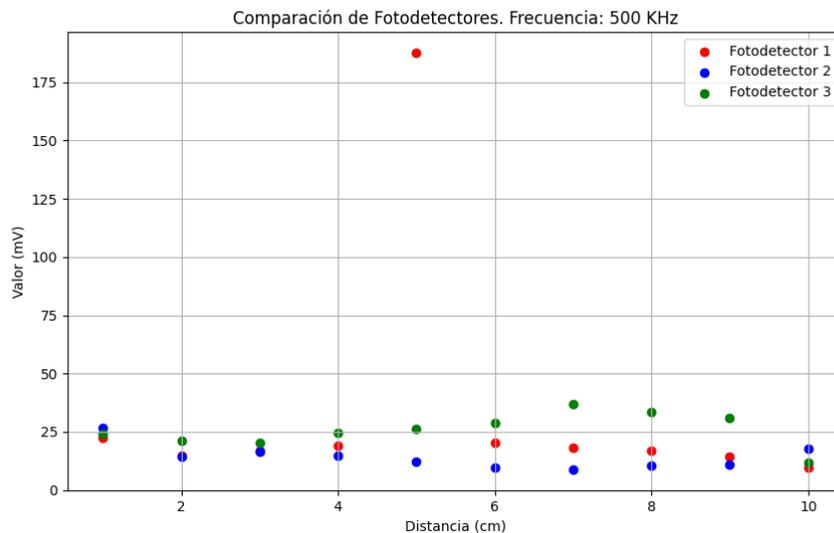


Gráfica 19. Comparación del voltaje Peak2Peak VS Distancia con tres fotodetectores para 200 kHz.

Fotodetector 1: Presenta alta variabilidad con picos significativos a 5 cm y 15 cm. La mayoría de los valores en otras distancias se mantienen en un rango de 0-25 mV, pero con fluctuaciones considerables, sugiriendo alta sensibilidad a fluctuaciones o ruido en el entorno de medición.

Fotodetector 2: Comienza con valores de voltaje alrededor de 26 mV a distancias cortas, decreciendo de manera significativa y constante con el aumento de la distancia. La baja dispersión de los datos indica mayor consistencia en las mediciones, aunque los valores son muy bajos.

Fotodetector 3: Muestra variabilidad significativa en los valores de milivoltios a medida que aumenta la distancia. Los valores aumentan hasta aproximadamente 8 cm antes de disminuir, sugiriendo sensibilidad variable según la distancia. La variabilidad es notable, con los valores más altos alcanzados en distancias intermedias.

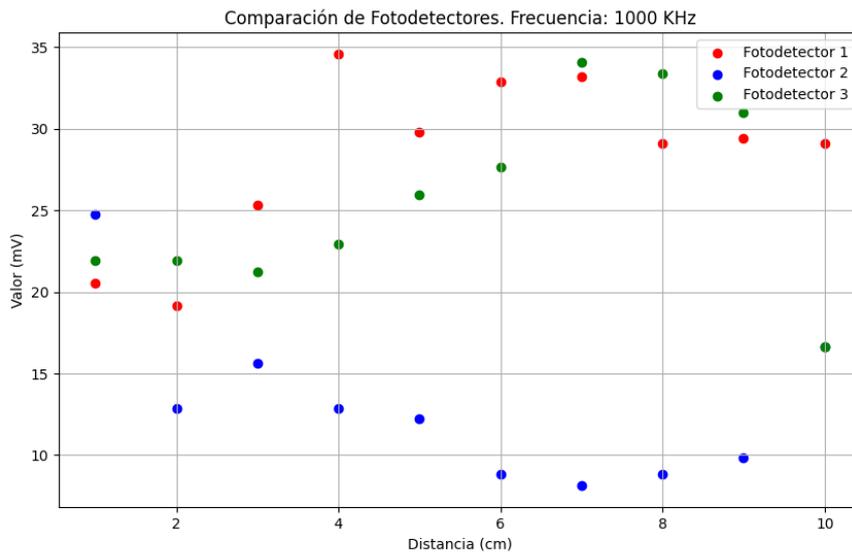


Gráfica 20. Comparación del voltaje Peak2Peak VS Distancia con tres fotodetectores para 500 kHz.

Fotodetector 1: Presenta alta variabilidad con picos significativos a 5 cm, 12 cm y 15 cm. La mayoría de los valores en distancias menores a 10 cm se mantienen alrededor de 25 mV, pero con fluctuaciones considerables, sugiriendo alta sensibilidad a fluctuaciones o ruido en el entorno de medición.

Fotodetector 2: Muestra una disminución constante y uniforme de los valores medidos a medida que la distancia aumenta. La estabilidad en los valores indica mejor rendimiento en términos de consistencia y menor sensibilidad al ruido en comparación con el Fotodetector 1. No se observan picos significativos, indicando una medición más uniforme.

Fotodetector 3: Presenta variabilidad significativa en los valores de milivoltios a medida que aumenta la distancia. Los valores aumentan hasta aproximadamente 8 cm antes de disminuir, sugiriendo sensibilidad variable según la distancia. La variabilidad es notable, con los valores más altos alcanzados en distancias intermedias, lo que podría ser útil en aplicaciones específicas.



Gráfica 21. Comparación del voltaje Peak2Peak VS Distancia con tres fotodetectores para 1000 kHz.

Fotodetector 1: Exhibe picos muy altos y gran variabilidad, con comportamiento errático especialmente alrededor de los 15 cm. A distancias mayores, los valores son bajos y estables, pero la inconsistencia sugiere alta sensibilidad al ruido o cambios en el entorno.

Fotodetector 2: Muestra una disminución constante y uniforme de los valores medidos a medida que la distancia aumenta. La estabilidad en los valores indica mejor rendimiento en términos de consistencia y menor sensibilidad al ruido, sin picos significativos.

Fotodetector 3: Presenta variabilidad significativa en los valores de milivoltios a medida que aumenta la distancia. Los valores aumentan hasta cierta distancia (6-8 cm) antes de disminuir, sugiriendo sensibilidad variable dependiendo de la distancia. La variabilidad es notable en estos valores intermedios.

En la siguiente tabla podemos ver los valores que se han obtenido tras la medición para cada uno de los fotodetectores.

Tabla 2. Tabla valores RSSI estimación de distancias en mV.

f (kHz)	1 kHz	2 kHz	5 kHz	10 kHz	20 kHz	50 kHz	100 kHz	200 kHz	500 kHz	1000 kHz
0	58.654	53.621	50.325	50.753	53.956	52.895	49.965	39.855	23.365	20.333
1	53.757	49.306	48.62	52.045	52.387	51.703	46.224	36.979	22.599	20.544
2	18.147	18.147	18.832	19.517	19.859	20.887	19.174	18.147	14.723	19.174
3	6.5056	6.848	7.1904	7.5328	7.8752	7.2248	10.272	10.957	17.120	25.338
4	4.7936	5.136	5.136	6.5056	6.5056	7.5328	8.56	10.272	18.832	34.583
5	149.63	182.84	247.9	196.88	191.74	220.85	231.81	202.02	187.83 5	29.789
6	2.0544	3.0816	3.424	3.7664	4.4512	6.1632	6.5056	10.614	20.202	32.871
7	2.3968	3.0816	3.424	3.7664	3.7664	5.136	3.1632	8.9024	18.147	63.213
8	2.3968	2.3968	3.424	3.7664	4.1088	5.4784	5.8208	8.2176	17.120	29.104
9	1.712	2.3968	3.424	3.4240	4.1088	5.1336	6.1613	8.56	14.381	29.447
10	18.147	2.3968	3.7664	3.4240	3.7664	5.1360	6.5056	9.2448	17.462	29.104

Tabla 3. Tabla valores RSSI triangulación en mV para el fotodetector 2.

f (kHz)	1 kHz	2 kHz	5 kHz	10 kHz	20 kHz	50 kHz	100 kHz	200 kHz	500 kHz	1000 kHz
0	14.585	5.450	7.914	10.243	12.143	16.100	16.666	20.668	19.334	17.932
1	14.585	5.4270	8.1405	11.532	13.228	20.012	22.726	26.457	26.796	24.761
2	7.4621	4.7486	6.7838	8.4797	10.515	11.872	11.872	15.942	14.246	12.889
3	5.0878	5.4270	7.4621	9.1581	10.854	14.585	13.568	17.638	16.281	15.603
4	4.4094	4.7486	7.4621	8.1405	10.176	12.889	13.568	14.924	14.585	12.889
5	4.0703	5.0878	8.1405	9.8362	10.854	11.872	11.193	13.228	12.211	12.211
6	3.7311	5.4270	7.1230	7.8013	7.8013	8.8189	9.1581	10.515	9.8365	8.8189
7	3.7311	4.7486	7.4621	8.1405	8.1405	8.8189	8.1405	10.854	8.8189	8.1405

8	3.3919	4.7486	5.0878	7.123	8.1405	8.4797	8.8189	10.515	10.515	8.8189
9	3.3919	4.0703	7.1230	6.4446	7.8013	8.1405	8.8189	10.176	10.854	9.8365
10	3.0527	4.4094	6.7838	8.8189	9.8365	14.246	15.263	16.620	17.638	16.620

Tabla 4. Tabla valores RSSI triangulación en mV para el fotodetector 3.

f (kHz)	1 kHz	2 kHz	5 kHz	10 kHz	20 kHz	50 kHz	100 kHz	200 kHz	500 kHz	1000 kHz
0	23.470	10.767	15.981	18.701	18.589	20.140	21.735	23.196	23.691	22.972
1	15.846	10.115	15.846	18.206	19.218	20.567	21.578	24.612	23.938	21.915
2	17.869	9.7775	12.475	16.521	17.195	18.881	18.544	20.904	21.241	21.915
3	17.835	10.115	14.498	16.521	16.858	18.544	19.892	20.567	20.229	21.241
4	42.144	10.115	14.498	16.183	16.183	19.555	22.252	23.601	24.612	22.927
5	9.1032	8.4289	11.800	13.486	15.509	20.567	24.612	25.961	26.298	25.961
6	23.601	8.7661	10.789	14.498	16.183	23.938	27.310	29.670	28.995	27.647
7	19.555	7.7546	9.7775	15.172	19.555	29.333	30.007	37.424	36.750	34.053
8	11.126	6.4060	7.7546	15.172	18.206	28.995	30.681	35.739	33.378	33.378
9	16.251	8.4289	9.7775	13.486	17.532	25.624	27.984	30.344	31.018	31.018
10	9.4404	7.0803	8.7661	8.4289	10.789	11.126	10.789	14.161	11.800	11.463

Tras un análisis detallado, se concluye que los tres fotodetectores presentan serias deficiencias en su rendimiento. Los valores obtenidos son generalmente bajos, con menos de 50 milivoltios en casi todas las frecuencias y distancias, y muestran una considerable cantidad de ruido. Esto afecta negativamente su

fiabilidad y utilidad en aplicaciones precisas, ya que los bajos niveles de señal implican una menor capacidad para captar y procesar la luz de manera efectiva.

Los datos presentan alta variabilidad con picos irregulares y fluctuaciones significativas, indicando una elevada sensibilidad al ruido. Este comportamiento errático compromete la precisión y la repetibilidad de los resultados, haciéndolos inadecuados para aplicaciones donde se requiere exactitud.

Fotodetector 1: Exhibe picos muy altos y gran variabilidad en varias frecuencias, con fluctuaciones drásticas que sugieren alta sensibilidad al ruido y a las fluctuaciones, haciendo menos confiable para mediciones precisas.

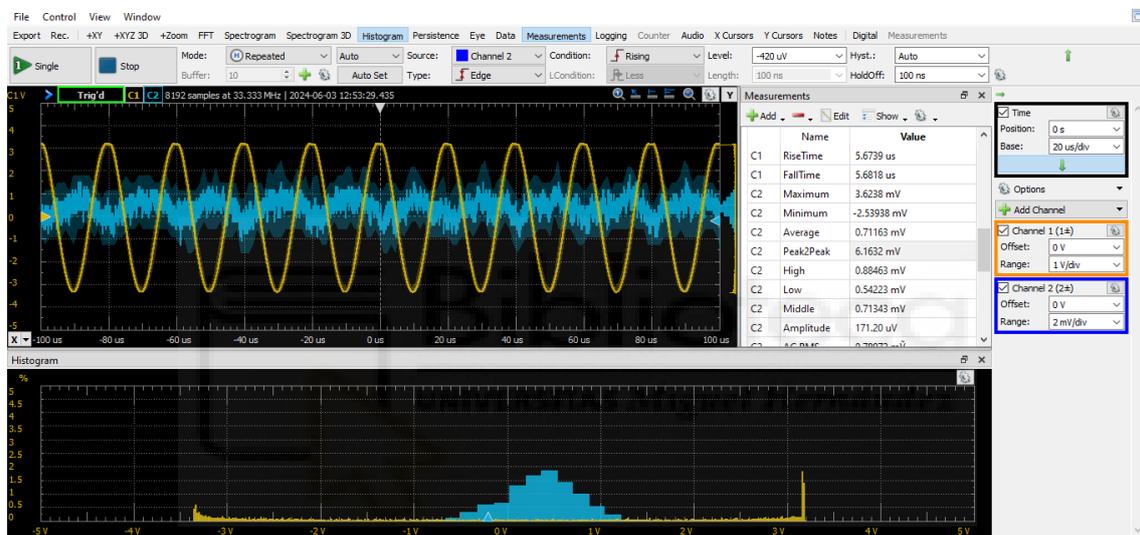


Ilustración 29. Osciloscopio del fotodetector 1 a 6 cm y 50 kHz, en amarillo la señal transmitida y en azul la señal recibida.

En cuanto al fotodetector 2, aunque muestra una tendencia más uniforme y estable, los valores medidos son aún demasiado bajos. La falta de picos significativos es positiva en términos de estabilidad, pero no compensa los bajos niveles de señal y la presencia de ruido.

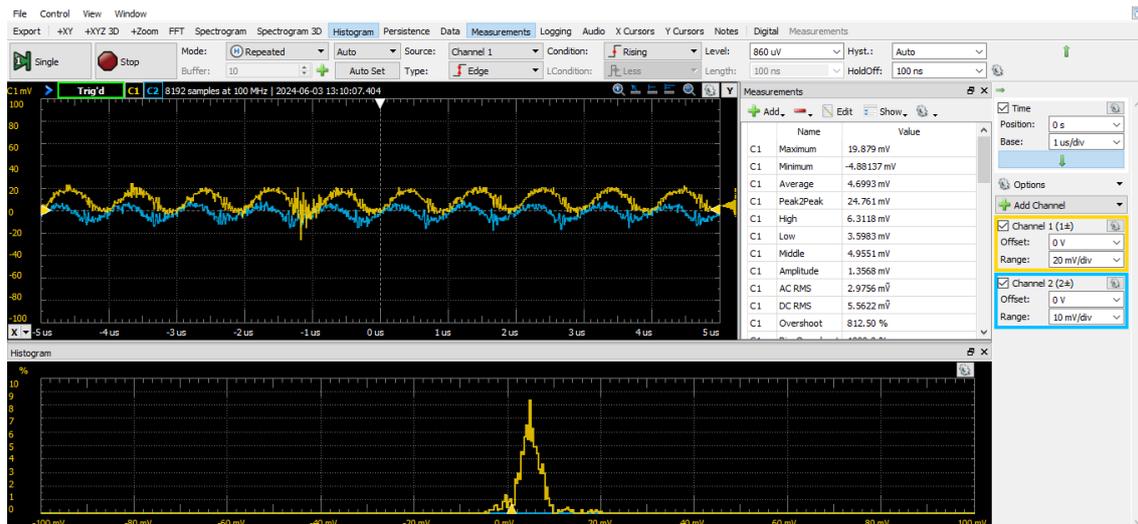


Ilustración 30. Osciloscopio del fotodetector 2 (amarillo) y 3 (azul) a 1 cm y 1000 kHz.

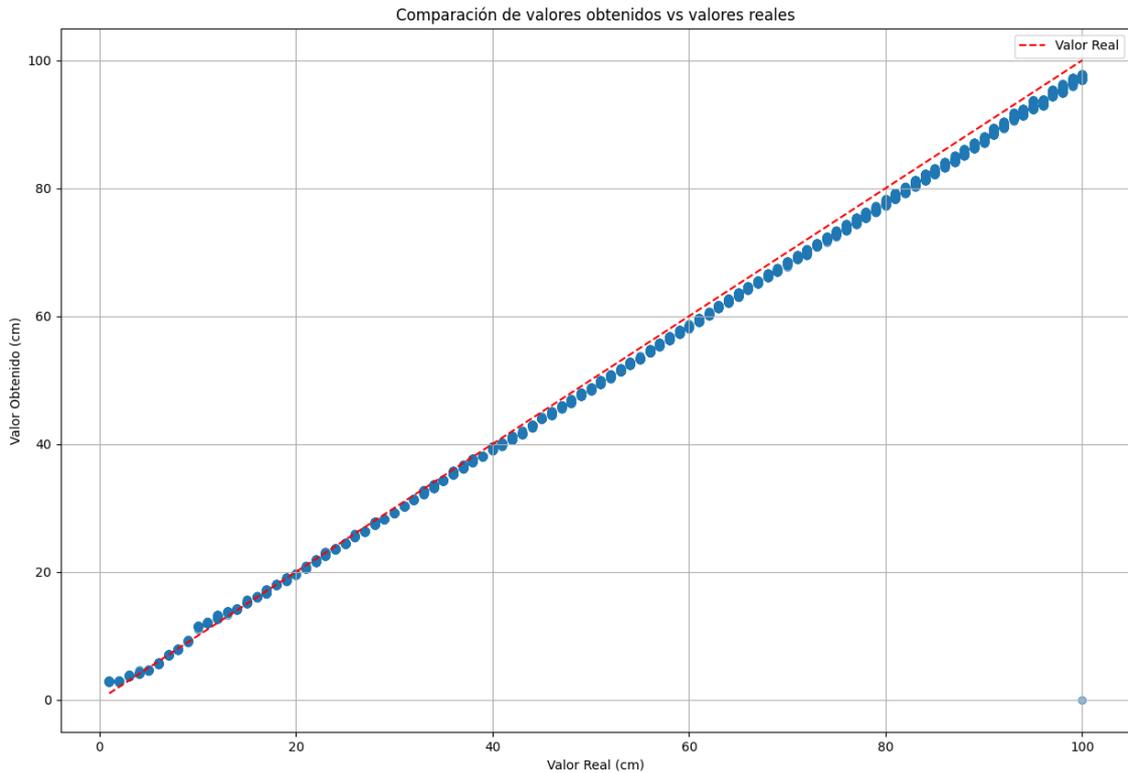
Finalmente, el fotodetector 3 muestra una variabilidad significativa, especialmente a mayores distancias, con un comportamiento que sugiere una sensibilidad variable. Aunque en algunas frecuencias presenta mejores valores que los otros dos fotodetectores, sigue siendo afectado por el ruido y la baja señal.

3.3. Algoritmo TOA para estimación de distancias con ultrasonidos HC-SR04

Para este estudio, se empleó un algoritmo de Time of Arrival (TOA) que requirió la adaptación constante de un parámetro, como se ha señalado anteriormente, con el objetivo de determinar la configuración que arroja los mejores resultados. Se llevaron a cabo 100 mediciones utilizando tres valores diferentes para dicho parámetro en un total de 100 centímetros. Posteriormente, utilizando Python, se procesaron los datos obtenidos para generar gráficos comparativos entre los valores reales y los valores calculados por el algoritmo. Para acceder a un análisis detallado de todas las medidas recopiladas, se invita a consultar los enlaces proporcionados.

- Constante 0.034

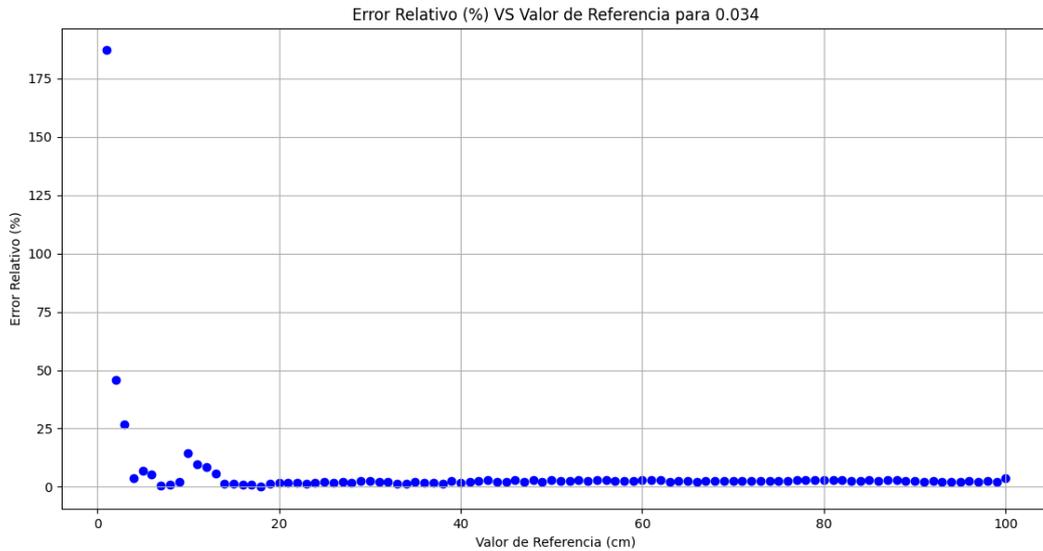
Iniciamos el estudio con la constante teórica que el algoritmo nos propone.



Gráfica 22. Comparación de Valores Obtenidos VS Valores Reales para la Constante 0.034.

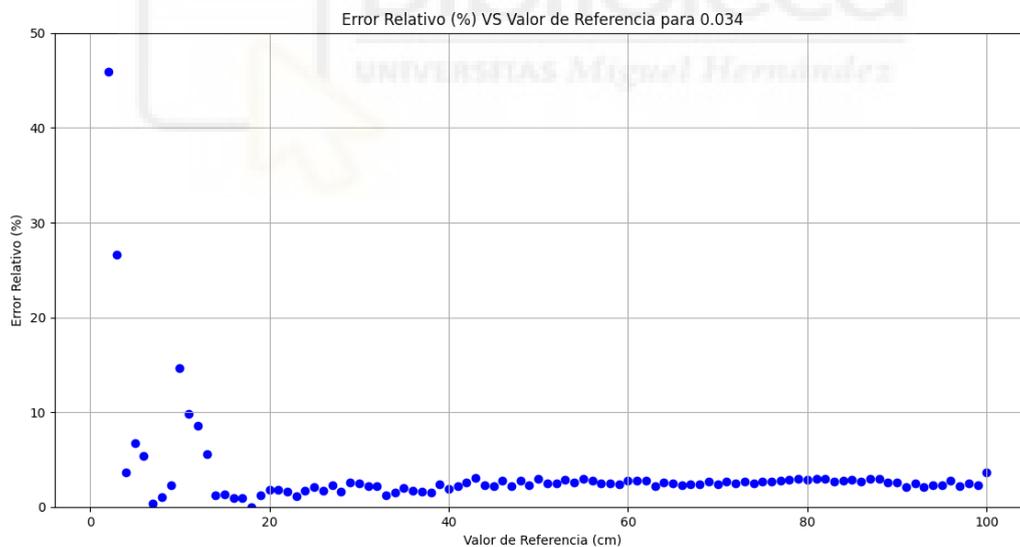
El análisis revela una relación lineal entre los valores obtenidos y los valores reales, como se evidencia en la alineación de los puntos a lo largo de la línea roja discontinua, lo que indica una fuerte correlación del método de medición. Sin embargo, alrededor del centímetro 40, la línea ideal se sitúa ligeramente por encima de los valores medidos, mostrando una leve desviación. En el extremo derecho del gráfico, donde los valores reales son más altos (70 a 100 cm), los puntos comienzan a separarse de la línea roja, sugiriendo una menor precisión del método a distancias mayores de 80 cm. Además, hay un punto significativamente alejado de la línea de tendencia en el extremo superior derecho, donde el valor obtenido es mucho más bajo que el valor real.

Observando una gráfica comparativa del error relativo con el valor de referencia obtenemos en este caso



Gráfica 23. Comparación de Error Relativo (%) VS Valor de Referencia para la Constante 0.034.

Como hemos concluido anteriormente, el mayor Error Relativo se encuentra en el primer centímetro. Si lo vemos más detalladamente, ampliando la gráfica al primer rango del Error Relativo:



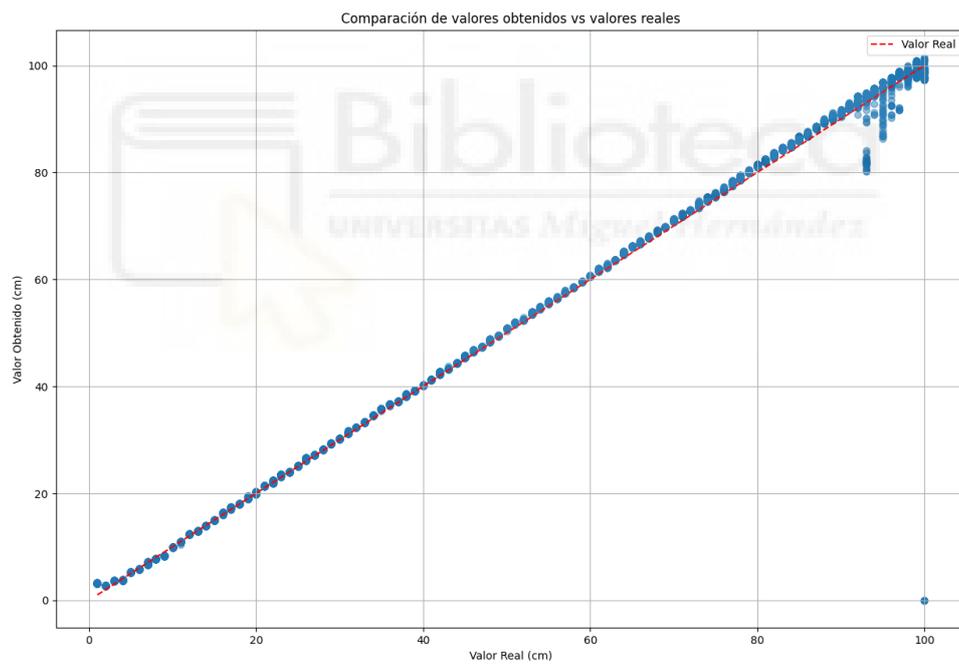
Gráfica 24. Comparación de Error Relativo (%) VS Valor de Referencia para la Constante 0.034 ampliado.

El error relativo en la región analizada es altamente variable, alcanzando hasta un 50% en algunos puntos, especialmente en los valores de referencia más bajos. En los primeros 10 cm del valor de referencia, hay varios puntos con errores relativos significativamente altos. Sin embargo, a medida que los valores de referencia aumentan, el error relativo disminuye considerablemente y se

estabiliza. Después de aproximadamente 20 cm, el error relativo se mantiene generalmente bajo, por debajo del 10%, con la mayoría de los puntos acercándose al 0%. Esto indica que el sistema o método de medición es menos confiable para valores de referencia bajos, pero mejora significativamente con valores de referencia más altos. Se sugiere modificar la constante utilizada en el método para obtener resultados más precisos en las mediciones.

Podemos continuar el análisis modificando la constante utilizada en el código para ver la variabilidad de los resultados. Se modifica la constante 0.034 por tres constantes distintas, haciendo un análisis diferente para cada una. Comenzamos con una constante más "alejada" a la velocidad del sonido.

- Constante 0.0353



Gráfica 25. Comparación de Valores Obtenidos VS Valores Reales para la Constante 0.0353.

La relación entre los valores obtenidos y los valores reales muestra una notable linealidad, evidenciada por la alineación de los puntos a lo largo de la línea roja discontinua, indicando una fuerte correlación y precisión en el método de medición. Sin embargo, en el extremo derecho del gráfico (80-100 cm), los puntos comienzan a desviarse de la línea, sugiriendo una menor precisión a

grandes distancias (más de 85 cm). Además, hay un punto significativamente alejado de la línea de tendencia en el extremo superior derecho, donde el valor obtenido es mucho más bajo que el valor real, lo que podría ser un outlier o un error de medición.

Observando una gráfica comparativa del error relativo con el valor de referencia obtenemos en este caso



Gráfica 26. Comparación de Error Relativo (%) VS Valor de Referencia para la Constante 0.0353.

Como hemos concluido anteriormente, el mayor Error Relativo se encuentra en el primer centímetro. Si lo vemos más detalladamente, ampliando la gráfica al primer rango del Error Relativo:

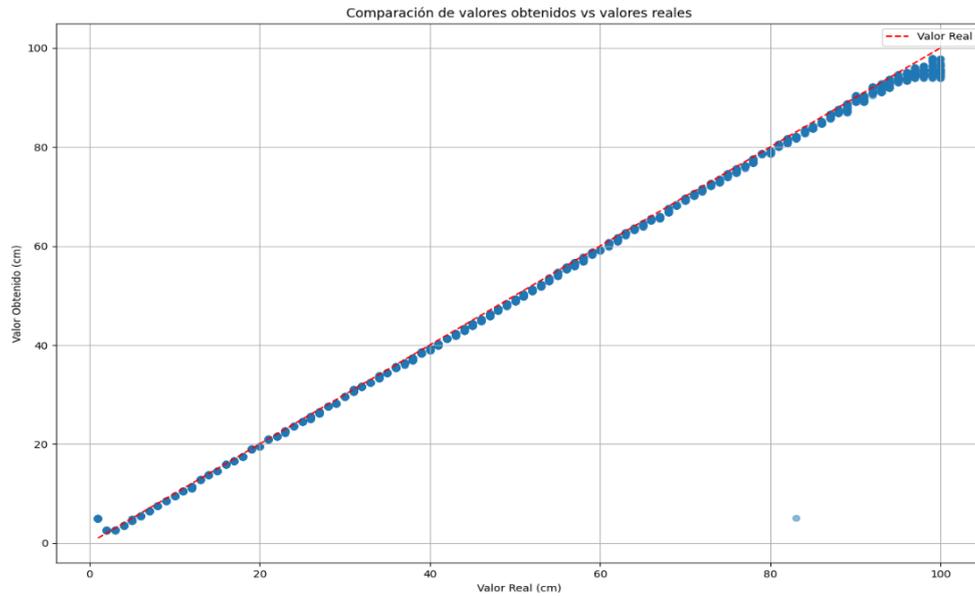


Gráfica 27. Comparación de Error Relativo (%) VS Valor de Referencia para la Constante 0.0353 ampliado.

El error relativo disminuye notablemente a medida que aumenta el valor de referencia, con una alta variabilidad al comienzo y una reducción significativa y rápida, estabilizándose por debajo del 1% después de los primeros 20 cm. Los valores más altos de error relativo se registran en los valores de referencia más bajos, superando el 30% y el 15%, lo que indica dificultades en lograr buena precisión con valores pequeños. Después de 20 cm, el error relativo se mantiene estable y bajo, sugiriendo una mayor precisión del método o instrumento de medición a medida que aumenta el valor de referencia. Algunos puntos se desvían de la tendencia general en la gráfica, posiblemente representando errores esporádicos en las mediciones o variaciones específicas que podrían necesitar revisión. Hacia el extremo derecho de la gráfica, cerca de 100 cm, se observa un leve incremento en el error relativo, lo que podría indicar un cambio en la precisión del instrumento o metodología en valores altos de referencia.

Seguimos el análisis modificando la constante a 0.03432, un poquito más ajustada a la velocidad del sonido.

- Constante 0.03432



Gráfica 28. Comparación de Valores Obtenidos VS Valores Reales para la Constante 0.03432.

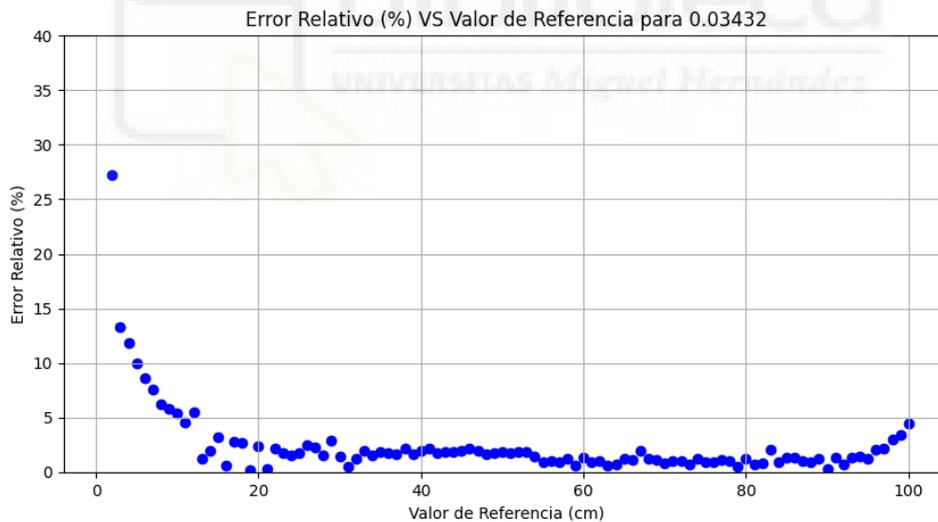
Esta gráfica presenta una correlación casi perfecta y lineal, con una dispersión mínima en todos los rangos de valores. Todos los puntos están muy cercanos a la línea ideal (línea discontinua). Los puntos se alinean muy bien con la línea roja, con excepción de un punto alrededor del valor real 80 cm, donde el valor obtenido es ligeramente inferior al esperado. Este pequeño desajuste podría ser resultado de un error puntual o de una leve variabilidad en la medición. Parece haber otro outlier en la parte inferior derecha, cerca del valor real 100 cm, donde el valor obtenido cae notablemente por debajo de la línea de tendencia.

Observando una gráfica comparativa del error relativo con el valor de referencia obtenemos en este caso



Gráfica 29. Comparación de Error Relativo (%) VS Valor de Referencia para la Constante 0.03432.

Como hemos concluido anteriormente, el mayor Error Relativo se encuentra en el primer centímetro. Si lo vemos más detalladamente, ampliando la gráfica al primer rango del Error Relativo:



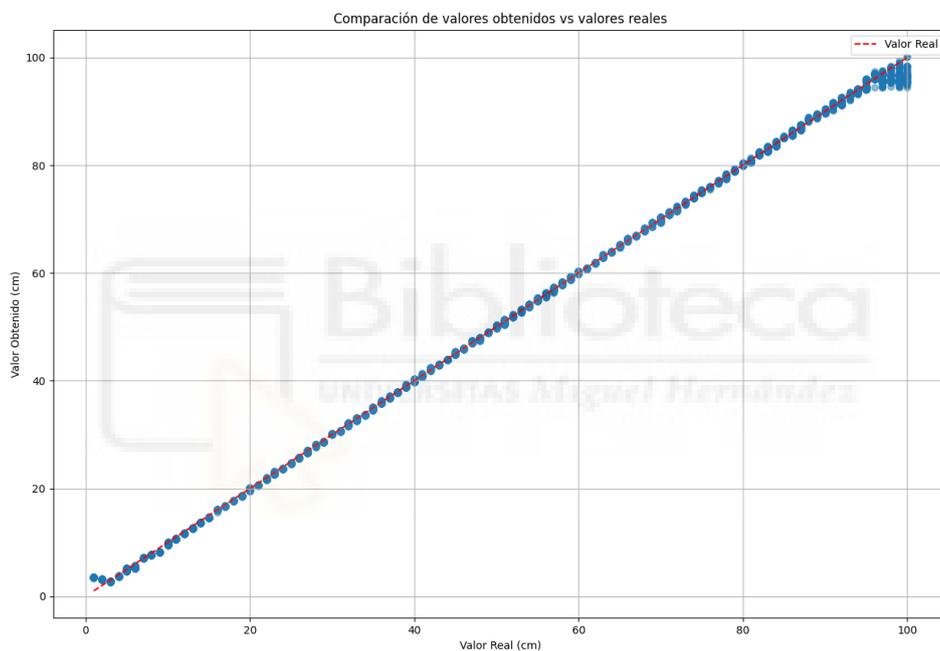
Gráfica 30. Comparación de Error Relativo (%) VS Valor de Referencia para la Constante 0.03432 ampliado.

El error relativo disminuye significativamente a medida que aumenta el valor de referencia, con una caída rápida hasta aproximadamente los 20 cm, seguida de una estabilización. Los valores más altos de error relativo se observan en valores de referencia bajos, especialmente cerca de 1 cm, donde el error alcanza hasta el 400%, indicando dificultades del método de medición en estos rangos. A partir

de los 20 cm, el error relativo se estabiliza en torno al 1% o menos, mejorando la precisión con valores más altos. Algunos puntos en el rango de 20 a 100 cm muestran un error ligeramente mayor, pero son pocos y podrían ser errores puntuales. Cerca de los 100 cm, se observa un ligero aumento en el error relativo, lo que podría indicar limitaciones del método o instrumento en valores altos.

Finalmente, si ajustamos un poquito más el valor de la constante con la velocidad del sonido, estudiamos la constante 0.03483.

- Constante 0.03483



Gráfica 31. Comparación de Valores Obtenidos VS Valores Reales para la Constante 0.03483.

Similar a la segunda gráfica, muestra una excelente correlación lineal entre los valores obtenidos y los valores reales. Sin embargo, hacia el extremo superior, algunos puntos ligeramente se desvían de la línea ideal.

Observando una gráfica comparativa del error relativo con el valor de referencia obtenemos en este caso



Gráfica 32. Comparación de Error Relativo (%) VS Valor de Referencia para la Constante 0.03483.

Como hemos concluido anteriormente, el mayor Error Relativo se encuentra en el primer centímetro, con un valor del caso 250%. Si lo vemos más detalladamente, ampliando la gráfica al primer rango del Error Relativo, aunque no es significativamente alto.



Gráfica 33. Comparación de Error Relativo (%) VS Valor de Referencia para la Constante 0.03483 ampliado.

Al principio, especialmente en los valores más bajos cerca de segundo centímetro, el error relativo es bastante alto, superando incluso el 15%. Esto indica que el método de medición tiene dificultades para alcanzar precisión en mediciones de valores pequeños.

A medida que el valor de referencia aumenta, el error relativo disminuye significativamente, especialmente después de los 20 cm, llegando a estabilizarse cerca del 0%. Esto sugiere que el método de medición es más preciso en rangos más altos.

Se han hecho una serie de cálculos en los que se calcula cual es el valor de error relativo para todas las medidas y en la siguiente tabla se muestran los mejores y peores resultados de error relativo para cada una de las constantes estudiadas.

Tabla 5. Resumen de medidas para algoritmo TOA con un sensor Ultrasonidos HC-SR04.

Constante	Archivo	Valor Referencia (cm)	Máximo	Mínimo	Media	Error Relativo (%)
0.034	1.txt	1	2.89	2.77	2.8708	187.08
0.034	18.txt	18	18.14	17.94	17.9973	0.015
0.0353	1.txt	1	3.23	3.02	3.1446	214.46
0.0353	18.txt	18	18.06	17.97	18.0083	0.046111111
0.03432	1.txt	1	5.06	4.92	4.9654	396.57
0.03432	19.txt	19	19	18.79	18.9517	0.0254210526
0.03483	1.txt	1	3.6	3.4	2.4804	248.04
0.03483	32.txt	32	32.24	31.61	31.9999	0.0003125

Las gráficas presentan una relación lineal muy fuerte entre los valores obtenidos y los reales, indicando un método efectivo de medición o estimación.

En las gráficas, se observa que el primer centímetro tiene un error relativo demasiado elevado, siendo el mayor de todos un 450% en la gráfica correspondiente a la constante 0.3432.

El análisis de la constante 0.034 inicialmente parece prometedor, pero muestra un desliz significativo que se magnifica con la distancia. Ajustando la constante, los resultados se aproximan mejor a la realidad.

En el segundo caso (0.0353) exhibe picos de error altos (más del 35%) en los primeros 2-5 cm. El error se reduce drásticamente y se mantiene casi plano en cerca del 0% a lo largo del rango de medición, con un pequeño aumento cerca de los 100 cm.

Para el tercer caso (0.03432) comienza con un pico de error muy alto (más del 30%) en los primeros 2-5 cm. El error relativo cae significativamente y se estabiliza cerca del 0% para valores superiores a 20 cm, con una consistencia notable y un aumento sutil al final.

Finalmente, el cuarto caso (0.3483) el error relativo es alto en valores bajos de referencia, pero disminuye rápidamente y se mantiene bajo para valores superiores a 20 cm, con un aumento notable cerca de 100 cm.

Todas las gráficas muestran una disminución rápida del error relativo después de los valores iniciales y estabilidad en errores bajos para la mayoría del rango. Sin embargo, la tercera gráfica (0.03483) ofrece el mejor desempeño en consistencia, con picos iniciales que se reducen rápidamente y una línea estable en casi todo el rango, con un aumento leve al final, indicando un ajuste y calibración efectivos del instrumento de medición para un amplio rango de valores de referencia.

3.4. Algoritmo TOA para triangulación con tres ultrasonidos

En el desarrollo del algoritmo TOA (Time of Arrival) para triangulación es imprescindible la adaptación continua de un parámetro específico para determinar cuál constante optimiza los resultados del sistema. Para este propósito, se realizaron 100 mediciones para cada uno de los tres sensores dispuestos en una configuración geométrica equiangular, con una separación de 120 grados entre cada sensor. El objetivo de este análisis no solo incluye la determinación de la constante más eficaz, sino también la evaluación de cuál de

los sensores proporciona un rendimiento superior y las razones subyacentes de esta superioridad.

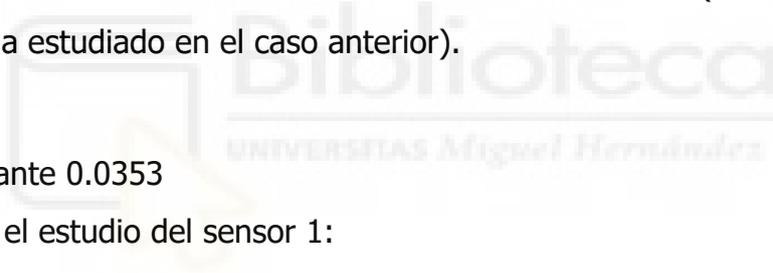
Aunque la constante 0.034 inicialmente proporcionó resultados con una desviación menor en comparación con otros estudios de constantes, se opta por omitirla. Esto se debe a que, a pesar de su menor desviación, los valores obtenidos siguen estando menos cercanos al valor real, provocando un sutil error que tiende a aumentar con la distancia, en lugar de atenuarse, provocando así que no se dieran los mejores resultados.

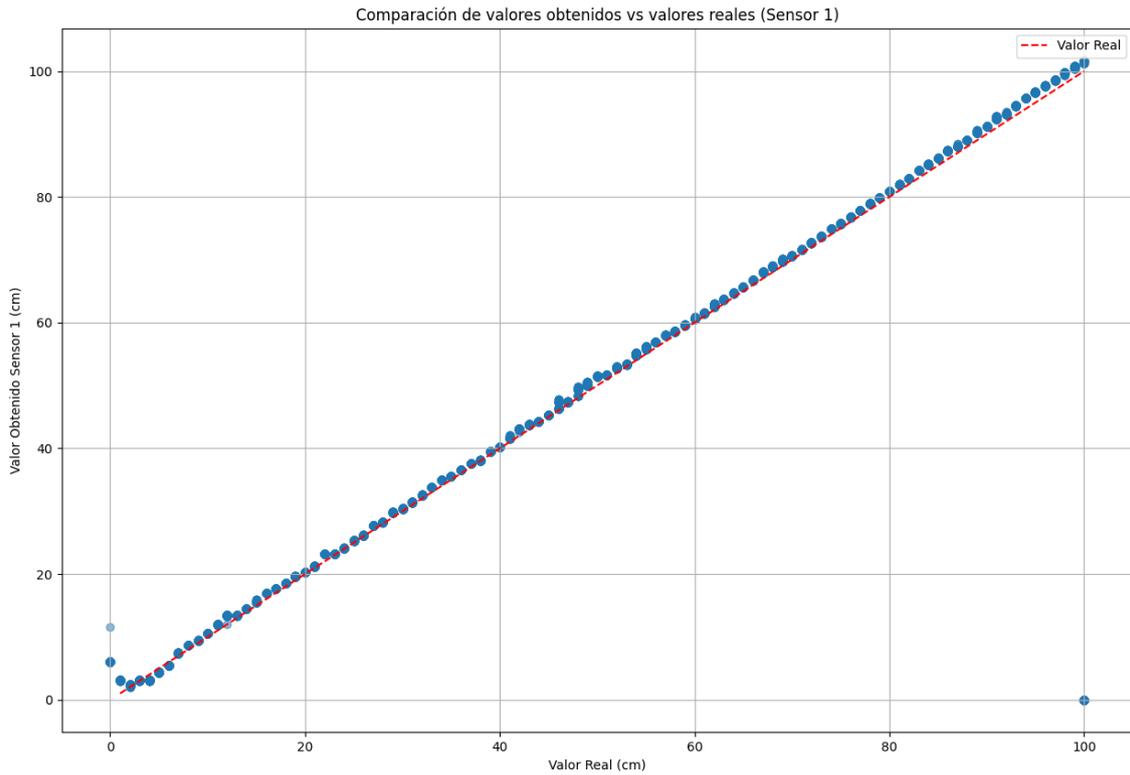
La flexibilidad para ajustar la constante permite una mayor adaptabilidad y precisión en la obtención de resultados más fieles a la realidad.

Iniciamos este estudio con el valor de la constante 0.0353 (mismo valor que también se ha estudiado en el caso anterior).

- Constante 0.0353

Se inicia con el estudio del sensor 1:

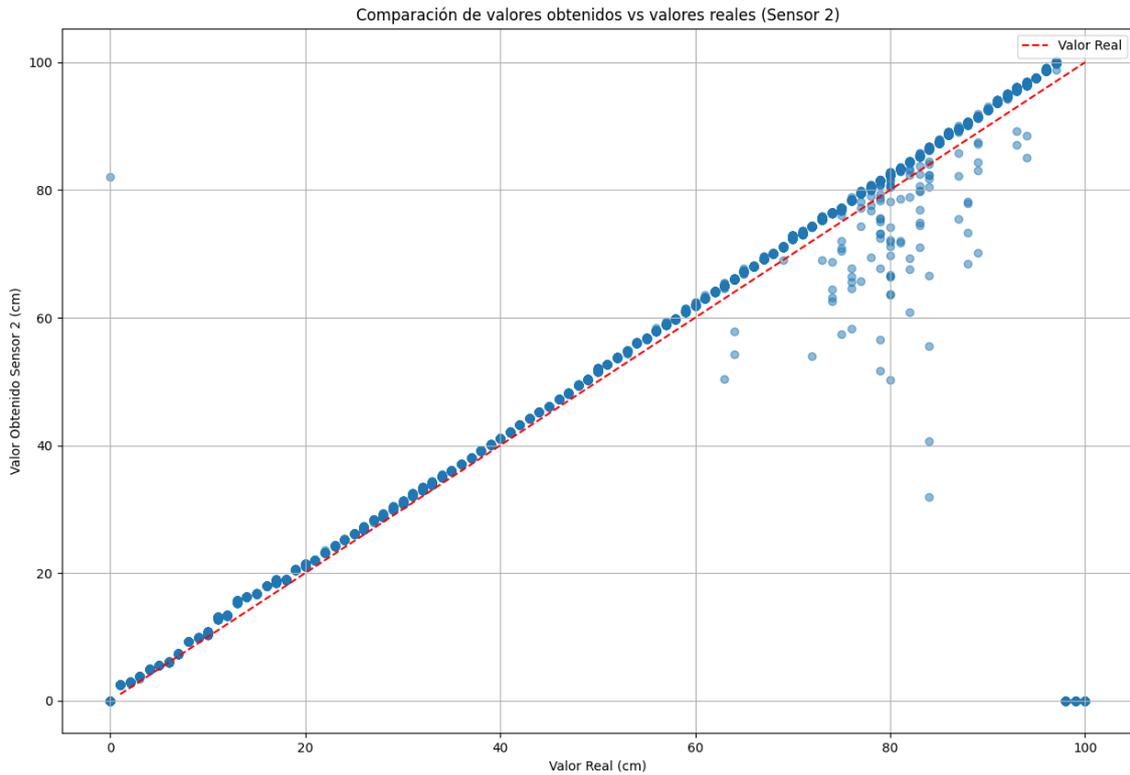




Gráfica 34. Comparación de Valores Obtenidos VS Valores Reales del Sensor 1 para la Constante 0.0353.

Los puntos en esta gráfica siguen de cerca la línea punteada roja, aunque con ligeras desviaciones en valores bajos y altos. Aunque hay algunas discrepancias menores, en general, el sensor 1 parece ser bastante preciso y confiable.

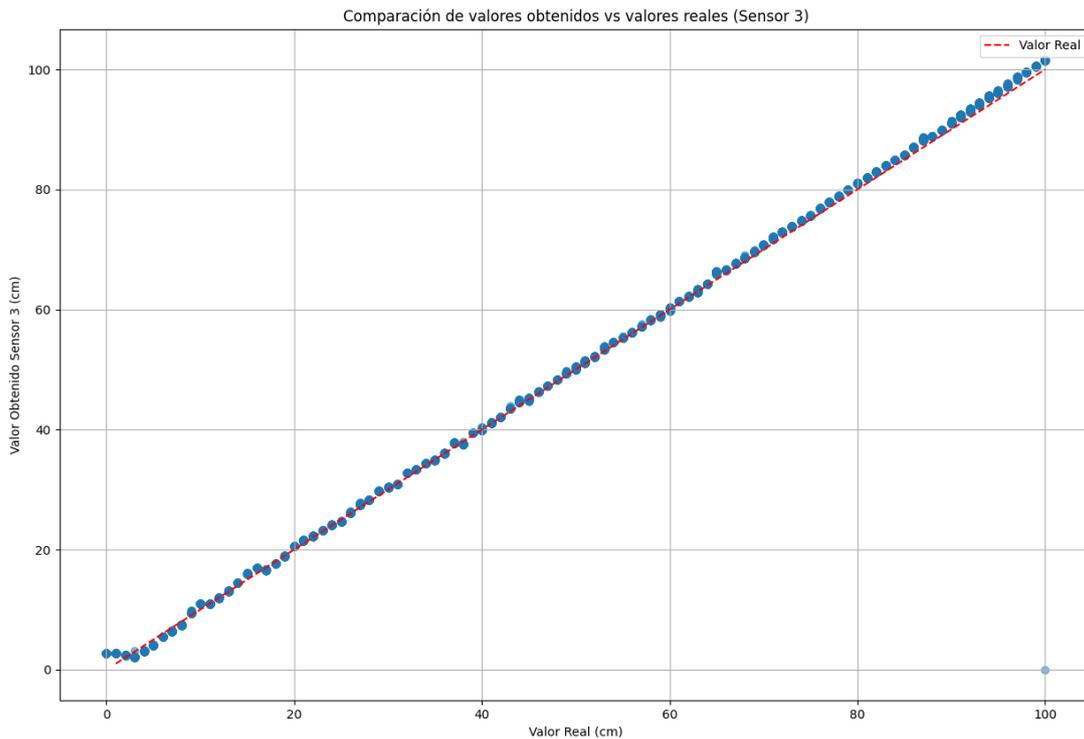
Si ahora se estudia el sensor 2 obtenemos:



Gráfica 35. Comparación de Valores Obtenidos VS Valores Reales del Sensor 2 para la Constante 0.0353.

En esta gráfica, la mayoría de los puntos están cerca de la línea roja punteada, indicando una buena correspondencia entre los valores medidos y los reales. Sin embargo, hay cierta dispersión de puntos alejados de esta línea, especialmente cuando los valores reales superan los 60 cm, sugiriendo posibles inexactitudes o inconsistencias en la medición del sensor en este rango.

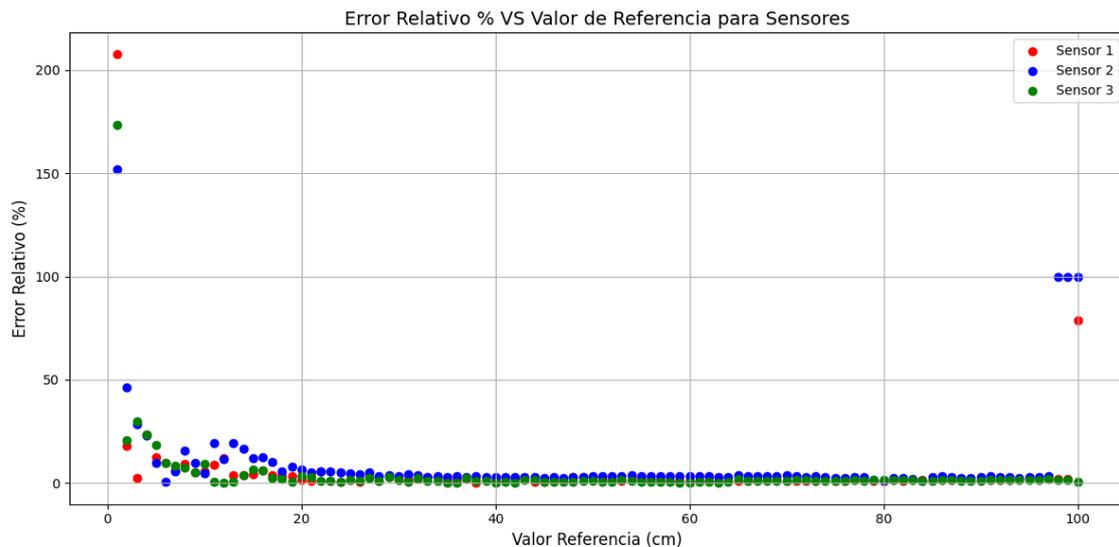
Finalmente, se estudia el sensor 3 y obtenemos:



Gráfica 36. Comparación de Valores Obtenidos VS Valores Reales del Sensor 3 para la Constante 0.0353.

Similar al sensor 1, esta gráfica muestra una alineación casi perfecta de los puntos con la línea de valor real. Los valores obtenidos por el sensor son muy precisos en comparación con los valores reales a lo largo de todo el rango de medición. Esta consistencia indica que el sensor 3 tiene un alto nivel de precisión y confiabilidad.

Observando las gráficas comparativas de cada sensor:



Gráfica 37. Comparación de Error Relativo (%) VS Valor de Referencia de los sensores para la Constante 0.0353.

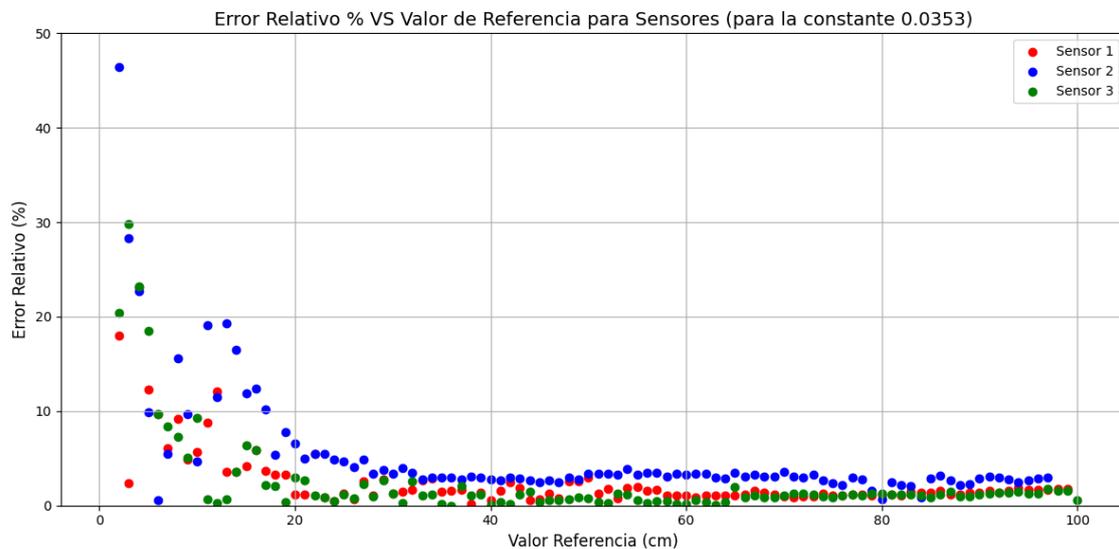
En el sensor 1, el punto de mayor error relativo se encuentra en el primer centímetro. La mayoría de los puntos están agrupados cerca del 0% de error relativo, indicando que, en general, el sensor tiene un error relativo muy bajo y, por ende, alta precisión. Sin embargo, existen varios puntos que se desvían hacia valores más altos de error relativo, representando mediciones con un error considerablemente mayor, especialmente en los extremos del rango de medición (0-20 centímetros).

Para el sensor 2, en valores de referencia cercanos a 0, el error relativo es extremadamente alto, alcanzando hasta más de 140% en el primer centímetro. Este pico de error disminuye significativamente a medida que aumenta el valor de referencia. A medida que el valor de referencia se incrementa, el error relativo tiende a estabilizarse cerca del 0%, mostrando una mayor precisión en las mediciones del sensor en este rango. Sin embargo, para valores superiores a 95 centímetros, el error relativo vuelve a ser alto, alcanzando el 100%.

En el sensor 3, se observa que, en valores de referencia bajos, especialmente en el primer centímetro, los errores relativos son muy altos, con algunos puntos alcanzando más de 175%. La precisión del sensor mejora significativamente para mediciones que superan los 20 centímetros de valor de referencia, volviéndose

muy confiable en este rango, lo que es ideal para aplicaciones donde las mediciones generalmente superan esta distancia.

Si ampliamos la gráfica:



Gráfica 38. Comparación de Error Relativo (%) VS Valor de Referencia de los sensores para la Constante 0.0353 ampliado.

En cuanto al sensor 1, para valores de referencia cercanos a 0, el error relativo es extremadamente alto, alcanzando más del 140% en el primer centímetro. Este error decrece significativamente a medida que el valor de referencia aumenta y tiende a estabilizarse cerca del 0%, mostrando mayor precisión en este rango. Sin embargo, para valores superiores a 95 centímetros, el sensor tiene un error relativo del 100%, indicando baja precisión en altos valores de referencia.

En el sensor 2, en los valores de referencia a partir de 2 centímetros, se observan errores relativos más altos, llegando hasta alrededor del 50%. La precisión del sensor mejora significativamente para valores de referencia más altos.

Finalmente, para el sensor 3, en valores de referencia bajos, especialmente en el primer centímetro, se observan errores relativos muy altos, con algunos puntos alcanzando más del 175%. La precisión del sensor mejora significativamente y se vuelve muy confiable para mediciones superiores a 20 cm. El error relativo disminuye significativamente a medida que el valor de referencia aumenta,

estabilizándose cerca del 0% alrededor de los 30 cm y manteniendo esa precisión hasta los 100 cm.

El análisis de los gráficos de rendimiento para los sensores 1, 2 y 3 revela diferencias clave en precisión y consistencia a lo largo de un rango de valores de referencia de 0 a 100 cm:

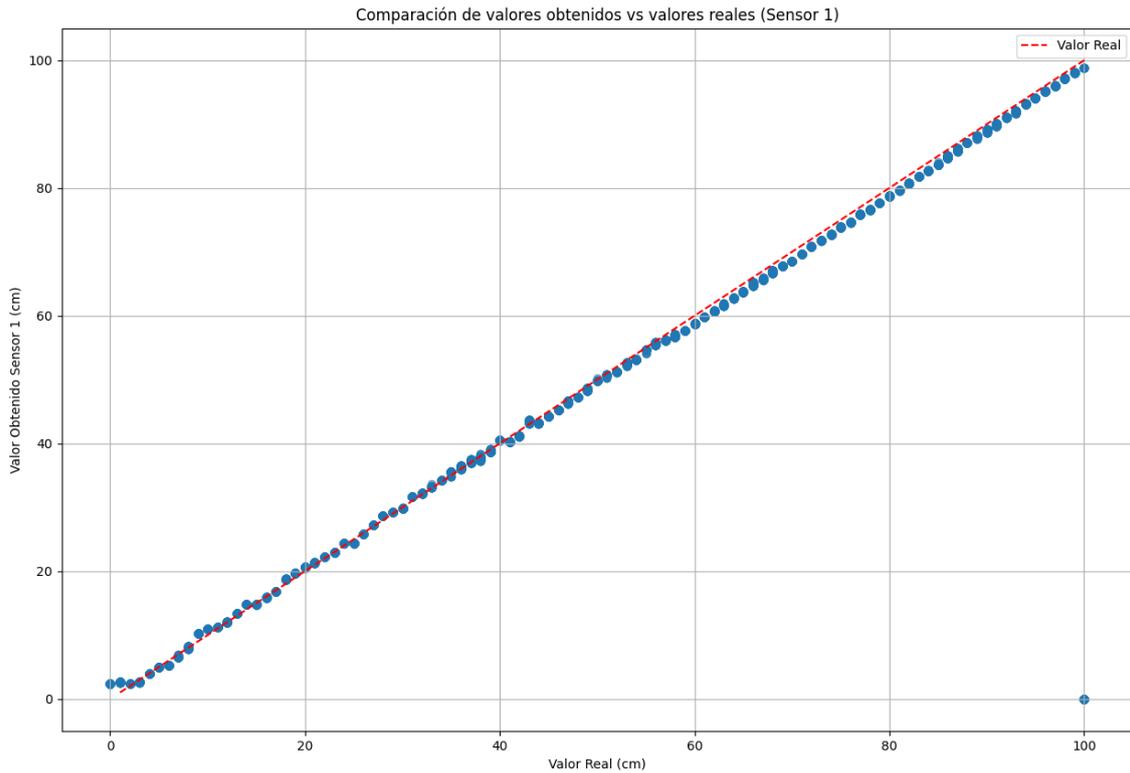
Ambos sensores muestran un desempeño excepcionalmente alto. La alineación casi perfecta de los puntos azules con la línea roja discontinua, que representa los valores reales, indica una precisión notable. Esto refleja una mínima variabilidad entre los valores medidos y los valores reales, demostrando alta confiabilidad y precisión.

El Sensor 2, aunque generalmente sigue la línea de valores reales, presenta una dispersión notable de puntos, especialmente en el extremo superior del rango de medición. Esto sugiere una menor precisión en comparación con los Sensores 1 y 3. La presencia de puntos significativamente alejados de la línea de valor real puede indicar problemas de calibración, estabilidad del sensor bajo ciertas condiciones, o menor sensibilidad en rangos de medición más altos.

Ajustando un poquito el valor de la constante a la velocidad del sonido, continuamos evaluando la constante 0.03432

- Constante 0.03432

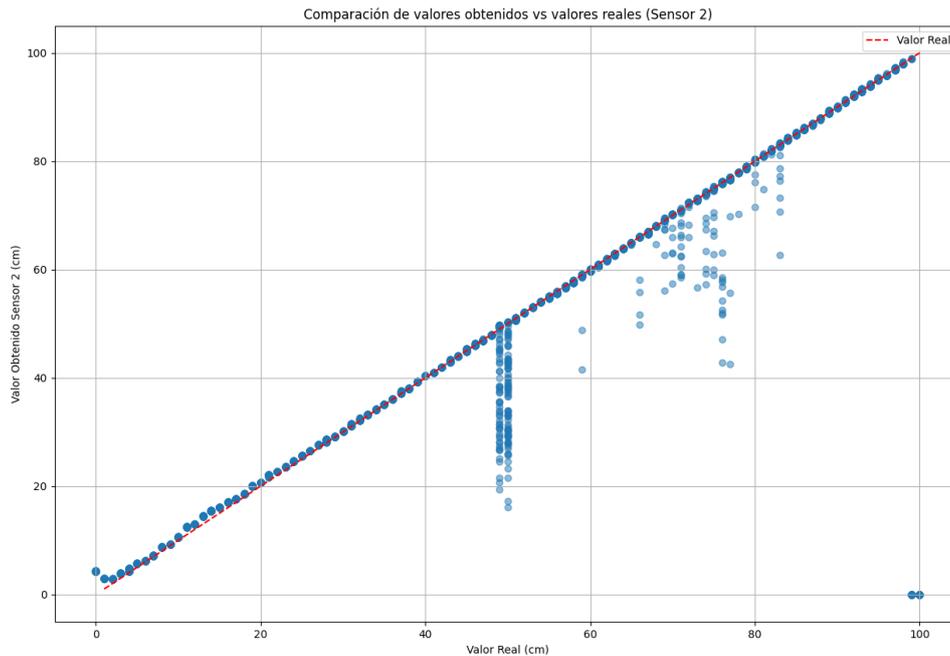
Se inicia con el estudio del sensor 1:



Gráfica 39. Comparación de Valores Obtenidos VS Valores Reales para el Sensor 1.

El análisis de las mediciones del sensor revela que los puntos azules siguen muy de cerca la línea roja discontinua, sugiriendo que las mediciones son muy precisas y consistentes con los valores reales. No hay desviaciones notables ni errores sistemáticos grandes entre las mediciones y los valores reales. El sensor mantiene una precisión constante en un rango amplio, desde valores cercanos a cero hasta alrededor de 100 cm. La proporcionalidad directa y lineal entre las mediciones y los valores reales indica que el sensor es confiable para medir distancias en este rango específico sin necesidad de ajustes o correcciones adicionales.

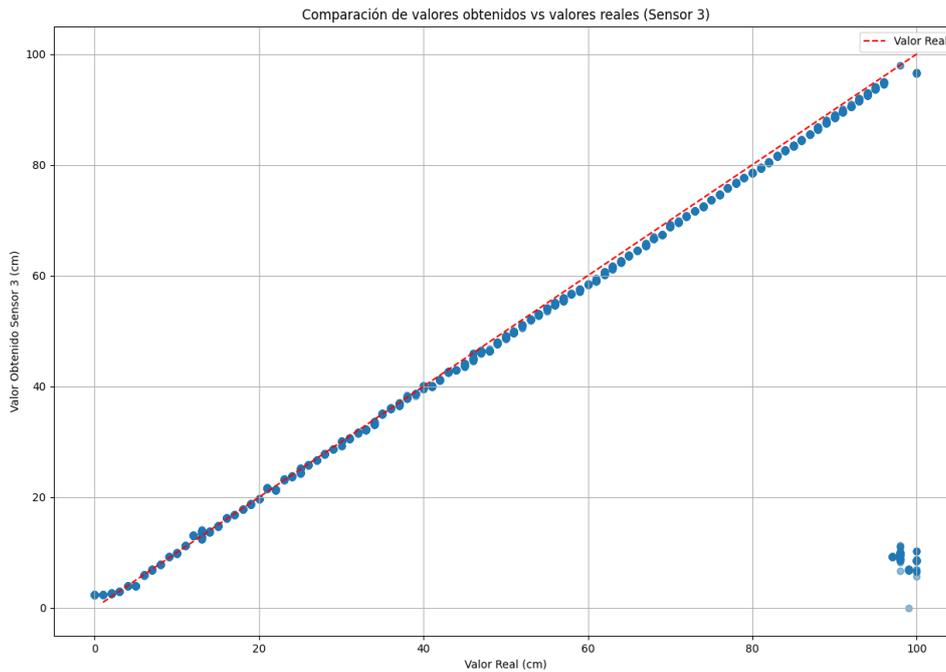
Continuamos el estudio con el sensor 2:



Gráfica 40. Comparación de Valores Obtenidos VS Valores Reales para el Sensor 2.

La gráfica muestra que el Sensor 2 realiza mediciones precisas y consistentes a distancias menores, con una notable degradación en la precisión a medida que la distancia aumenta, especialmente más allá de los 50 cm. Este patrón sugiere que el sensor puede no ser adecuado para aplicaciones que requieren mediciones precisas a distancias más largas, a menos que se identifiquen y corrijan las causas de la variabilidad observada.

Y finalmente para el Sensor 3:



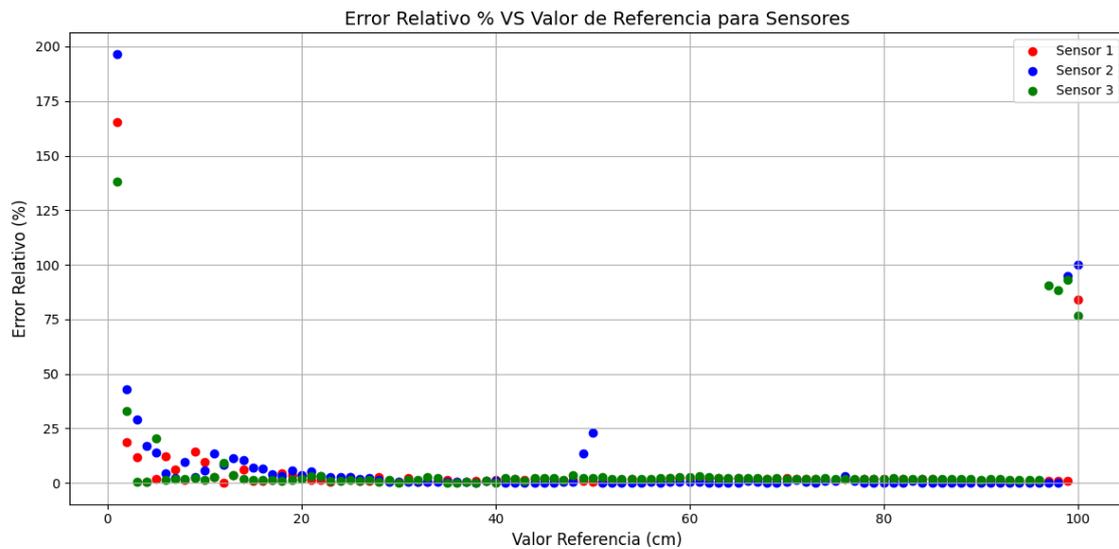
Gráfica 41. Comparación de Valores Obtenidos VS Valores Reales para el Sensor 3.

Desde 0 cm hasta aproximadamente 70 cm, los puntos azules siguen muy de cerca la línea roja discontinua, lo que indica que las mediciones del sensor son precisas y consistentes con los valores reales en este rango. La mayoría de las mediciones se alinean bien con la línea de valores reales, mostrando una alta precisión y baja variabilidad.

Hay una notable anomalía a partir de los 80 cm, donde varios puntos se agrupan lejos de la línea roja, con valores medidos que disminuyen drásticamente en comparación con los valores reales esperados, algunos incluso cayendo cerca del cero.

Esta desviación sugiere que el sensor puede tener limitaciones o fallas al medir distancias mayores, particularmente en el rango de 80 a 100 cm.

Si ahora estudiamos las gráficas comparativas de cada sensor tenemos:



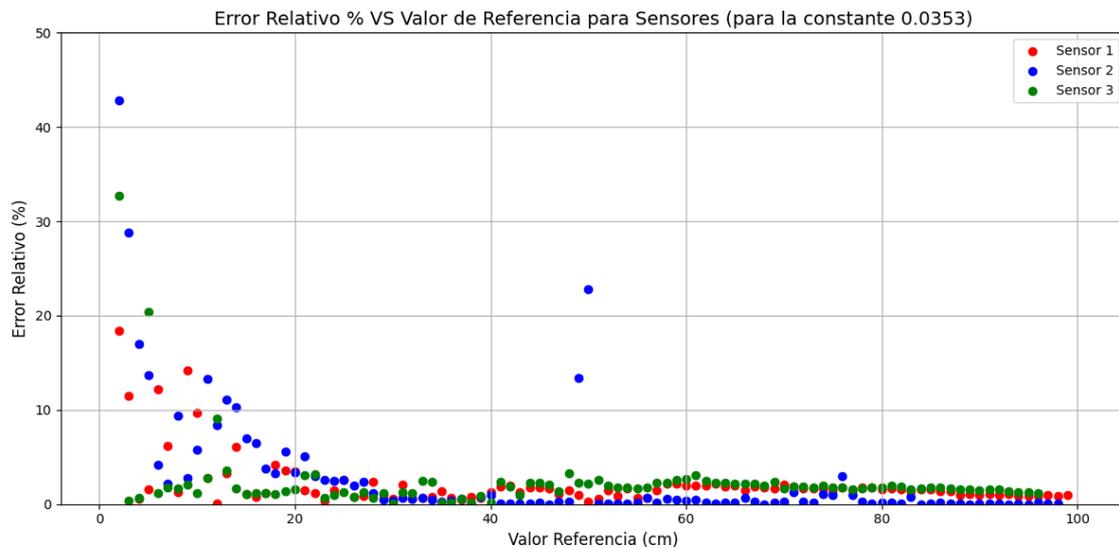
Gráfica 42. Comparación de Error Relativo (%) VS Valor de Referencia de los sensores para la constante 0.03432.

El análisis del error relativo para el sensor revela alta precisión en la mayoría de los valores de referencia, con datos agrupados cerca del 0% de error relativo. Sin embargo, se observan algunos puntos con errores significativamente altos, incluyendo uno cercano al 150% cuando el valor de referencia es 0 cm y otro alrededor del 85% a 100 cm.

El sensor 2 muestra el error más alto en el primer centímetro, alcanzando casi el 200% de error relativo. A medida que la distancia de referencia aumenta, el error relativo disminuye significativamente, estabilizándose mayormente por debajo del 5% para valores superiores a 20 cm.

En comparación con los sensores 1 y 2, este sensor 3 presenta un patrón diferente: aunque tiene un pico de error inicial de casi el 140%, este no es tan alto como en los otros sensores. Sin embargo, se observan más picos de error en distancias superiores a 95 cm, con errores relativos de más del 80%.

Ampliando la gráfica:



Gráfica 43. Comparación de Error Relativo (%) VS Valor de Referencia de los sensores para la constante 0.03432 ampliado.

Aun observando la gráfica más de cerca, vemos que el error relativo de mayor valor se encuentra por debajo de los 20 centímetros en los tres casos, llegando a alcanzar un valor de casi 50% para el sensor 2. Los errores tienden a disminuir y estabilizarse a medida que la distancia de referencia aumenta. A partir de aproximadamente 20 cm, la mayoría de los errores son inferiores al 5%, lo que sugiere que el sensor es más preciso en este rango.

Se observa para el sensor 2, que a valores menores de 50 centímetros hay puntos de error relativo muy elevados (entre el 15 y 25 %)

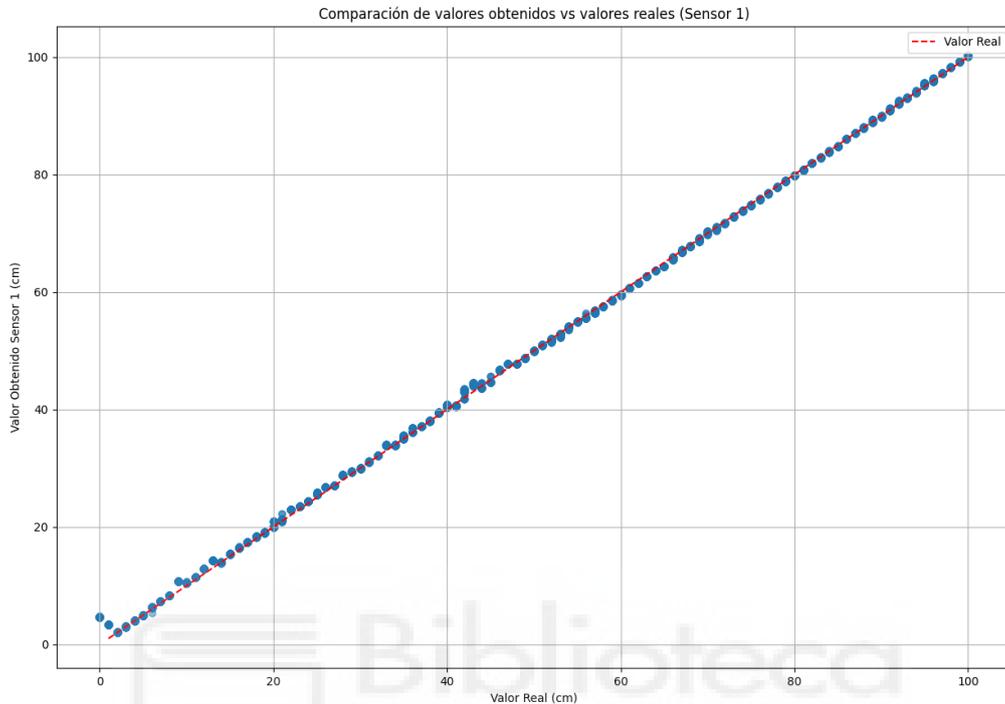
En cuanto al sensor 3, hay valores cercanos al 0% de error relativo, pero menos que en otros casos. Aun así, en este caso se observa que hay menos picos de error relativo alto.

Podemos concluir que todos los sensores exhiben un pico de error en mediciones cercanas a 0 cm y un incremento leve hacia el final del rango. El sensor 3 es el único que no tiene un único pico tan elevado al principio, sino que presenta una serie de picos más pequeños a lo largo del rango. Además, el sensor 3 muestra un aumento más definido y consistente en el error al final del rango.

Finalmente, como en el apartado anterior, evaluamos la constante 0.03483

- Constante 0.03483

Se inicia el estudio con el sensor 1:

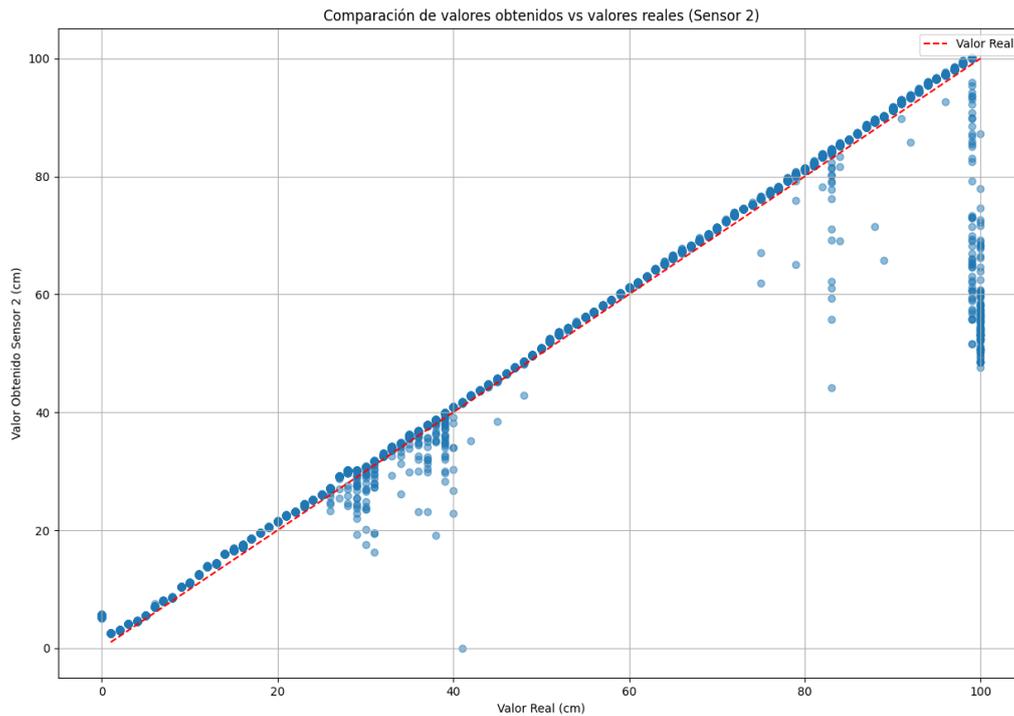


Gráfica 44. Comparación de Valores Obtenidos VS Valores Reales para el Sensor 1.

La proximidad de los puntos azules a la línea roja discontinua sugiere que el sensor es bastante preciso, ya que la mayoría de los puntos se alinean muy cerca o sobre la línea roja. Esto indica que los valores medidos por el sensor son muy cercanos a los valores reales.

Aunque la mayoría de los puntos están cerca de la línea ideal, algunos puntos parecen desviarse ligeramente. Analizar el error de medición estadísticamente (como el error medio cuadrático) podría dar una idea más clara de la precisión general del sensor.

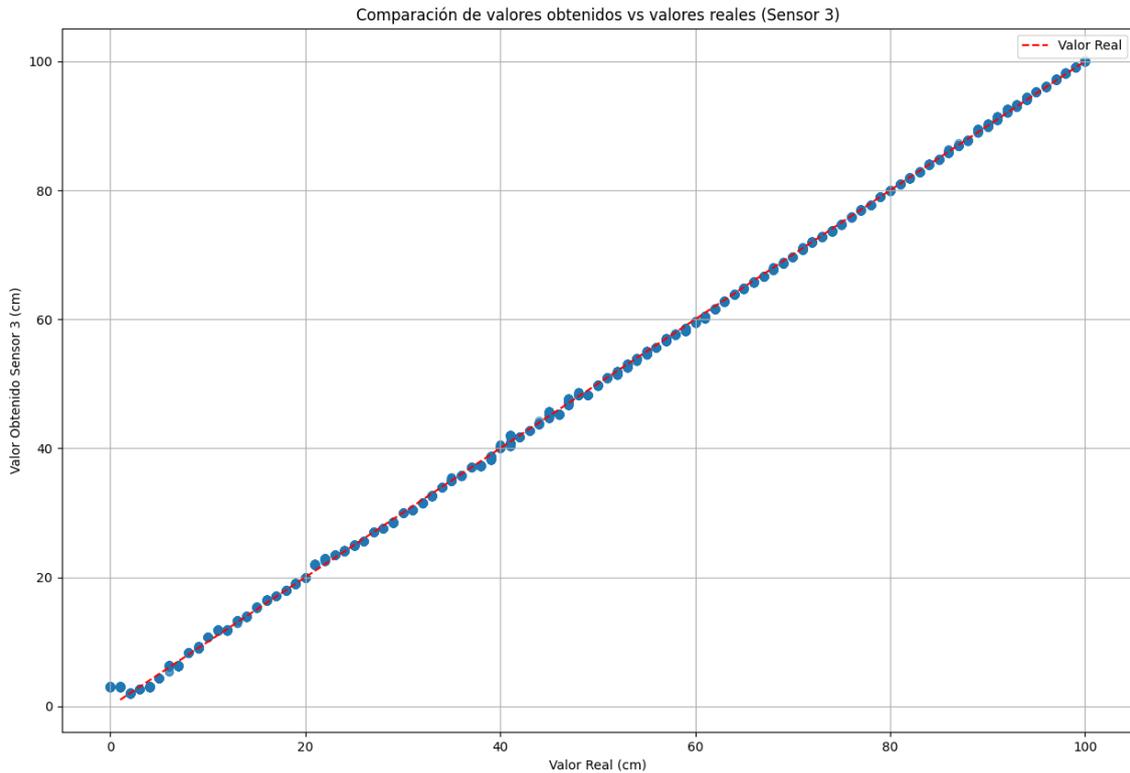
Continuando con el estudio del sensor 2 tenemos:



Gráfica 45. Comparación de Valores Obtenidos VS Valores Reales para el Sensor 2.

Los valores en la mayoría del rango son lineales y siguen de cerca la línea roja, sugiriendo una alta precisión en esos rangos. Sin embargo, hay un patrón distinto en ciertos rangos, en particular cerca de los 40 cm y en la región más alta, cerca de 80 a 100 cm.

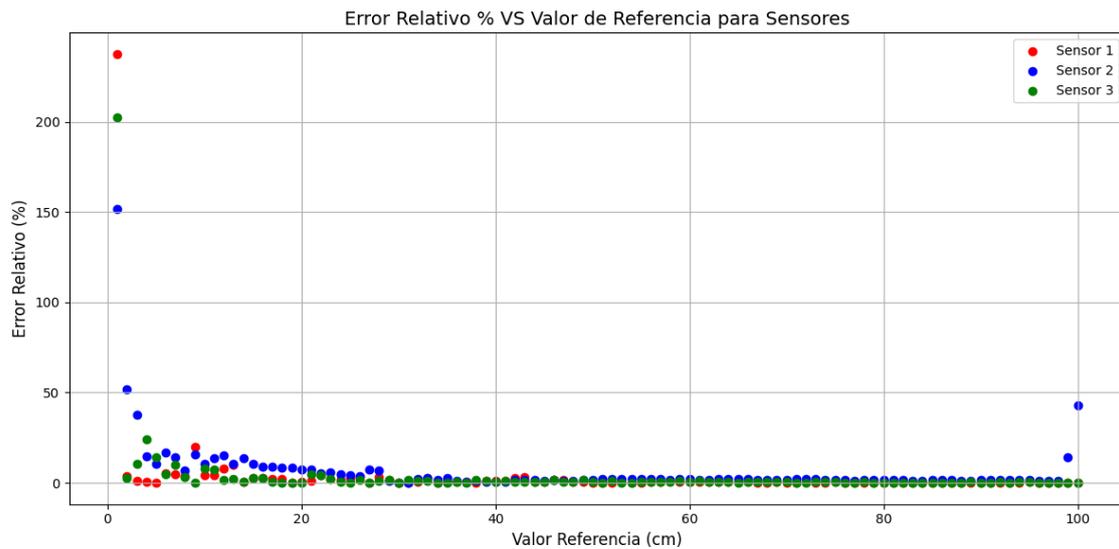
Finalmente, para el sensor 3:



Gráfica 46. Comparación de Valores Obtenidos VS Valores Reales para el Sensor 3.

La proximidad de los puntos azules a la línea roja sugiere que el sensor tiene una alta precisión, dado que la mayoría de los puntos se ubican sobre o muy cerca de la línea de referencia ideal. El sensor muestra una consistencia en las mediciones a lo largo de todo el rango de 0 a 100 cm. No se observan variaciones significativas o patrones anormales en ninguna parte específica del rango, lo cual es un indicativo de un buen desempeño general del sensor.

Si ahora observamos las gráficas comparativas para cada sensor:



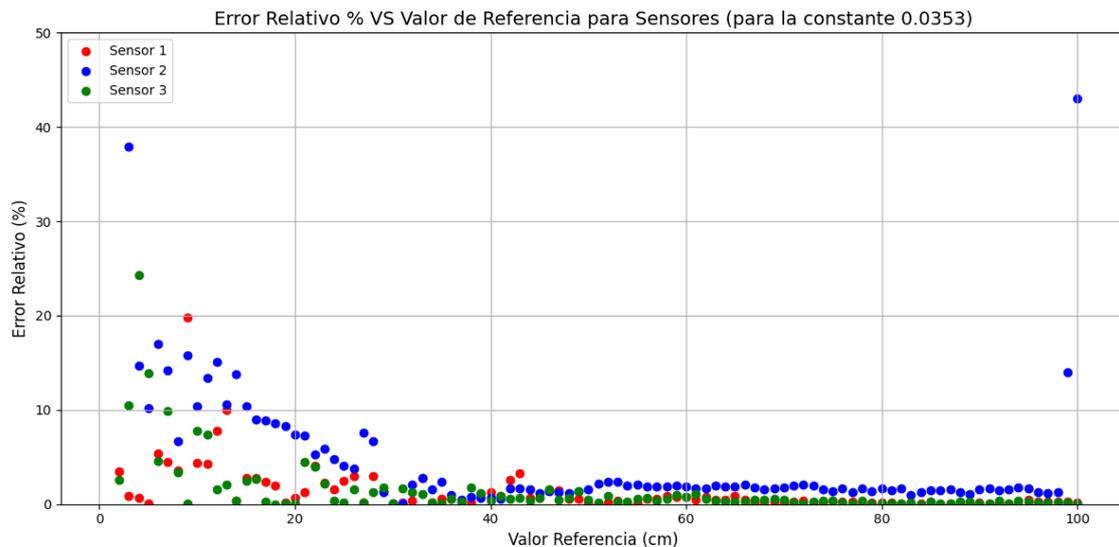
Gráfica 47. Comparación de Error Relativo (%) VS Valor de Referencia de los sensores para la constante 0.03483.

En el sensor 1, el mayor error relativo se encuentra en el primer centímetro (más del 200%). Para valores superiores a 20 cm, el error es bajo, pero hay variabilidad en valores menores a 20 cm, alcanzando hasta el 30%.

Para el sensor 2, se obtiene un error muy alto (más del 140%) en el primer centímetro, que disminuye rápidamente. El error se estabiliza cerca del 0% para valores mayores, mostrando alta precisión en rangos superiores.

Finalmente, el sensor 3 presenta un error extremadamente alto (más del 200%) en el primer centímetro, que también decrece drásticamente. El error se estabiliza cerca del 0% para valores mayores, indicando alta precisión.

Si ampliamos la gráfica:



Gráfica 48. Comparación de Error Relativo (%) VS Valor de Referencia de los sensores para la constante 0.03483 ampliado.

El sensor 1 presenta en distancias cortas, el sensor muestra un error relativo significativo, hasta un 20%, lo que indica menor precisión en estos rangos. A partir de 20 cm, el error relativo se estabiliza y se mantiene consistentemente bajo, por debajo del 10%, indicando mayor confiabilidad para distancias mayores.

Para el sensor 2, se presenta un error relativo alto, más del 40%, en el primer centímetro. Este error disminuye rápidamente y se estabiliza cerca de 0% a medida que aumenta la distancia, mostrando mayor precisión. Para valores superiores a 95 cm, el error relativo se acerca al 50%, indicando menor precisión en altos valores de referencia.

El sensor 3, en valores de referencia cercanos a 0, el error relativo es notablemente alto, más del 30% en los primeros centímetros. Este error decrece significativamente con la distancia. A partir de 20 cm, el error relativo se mantiene consistentemente bajo, por debajo del 5%, indicando alta precisión en el rango de 20 a 100 cm.

Todos los sensores presentan errores altos en distancias muy cortas, pero mejoran significativamente con la distancia.

El Sensor 1 es más confiable a partir de 20 cm, manteniendo un error bajo.

El Sensor 2 tiene alta precisión en rangos medios, pero pierde precisión en distancias superiores a 95 cm.

El Sensor 3 muestra alta precisión en la mayoría de los valores de referencia, especialmente entre 20 y 100 cm.

Podemos concluir que el sensor 1: Presenta un error relativo extremadamente alto, superando el 200% en el primer centímetro. A medida que el valor de referencia aumenta, el error disminuye y se estabiliza cerca de 0%, mostrando mayor precisión en rangos de 20 a 100 cm.

El Sensor 2 muestra un error relativo alto, superando el 140% en valores cercanos a 0 cm. El error disminuye y se estabiliza por debajo del 20% a partir de 20 cm, pero vuelve a aumentar para valores superiores a 95 cm.

Y el sensor 3 tiene un error relativo extremadamente alto al inicio, superando el 200% en el primer centímetro. El error se reduce significativamente y se mantiene por debajo del 25% para valores mayores a 5 cm, con alta precisión en el rango de 20 a 100 cm.

Se han hecho una serie de cálculos en los que se calcula cual es el valor de error relativo para todas las medidas y en la siguiente tabla se muestran los mejores y peores resultados de error relativo para cada una de las constantes estudiadas y cada uno de los sensores que intervienen en este estudio.

Tabla 6. Resumen de medidas para algoritmo TOA con triangulación con tres sensores DE Ultrasonidos HC-SR04.

Cte.	Sensor	Valor Ref. (cm)	Máximo (cm)	Mínimo (cm)	Media (cm)	Error Relativo (%)
0.03432	1	1	2.73	2.59	2.6516	165.16
0.03432	1	12	12.06	11.93	12.014	0.1166666667
0.03432	2	1	3	2.87	2.9652	196.25

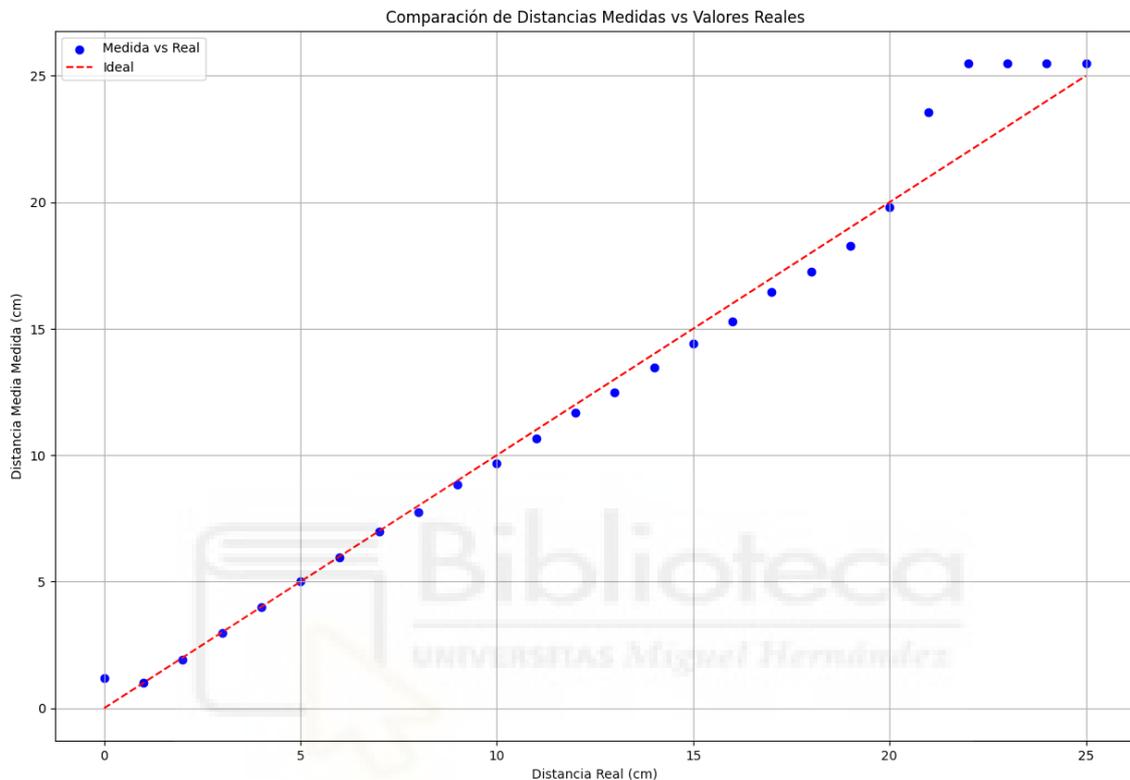
0.03432	2	68	68.14	64.64	67.9987	0.001911764706
0.03432	3	1	2.39	2.32	2.3816	138.16
0.03432	3	30	30.12	29.29	30.0151	0.050333333333
0.0353	1	1	3.09	3.02	30.774	207.74
0.0353	1	38	38.11	38.04	38.0512	0.134736
0.0353	2	1	2.59	2.45	2.52	152
0.0353	2		6.05	5.98	6.0332	0.553333
0.0353	3	1	2.81	2.67	2.733	173.3
0.0353	3	36	36.06	35.92	63.0054	0.015
0.03483	1	1	3.4	3.33	3.3741	237.41
0.03483	1	94	94.3	93.88	94.0081	0.008617021277
0.03483	2	1	2.56	2.49	2.5159	151.59
0.03483	2	30	30.84	17.54	29.9888	0.037333333333
0.03483	3	1	3.05	2.98	3.0262	202.62
0.03483	3	84	84.06	83.92	83.9921	0.009404761905

La selección de la constante depende del balance entre el error máximo y mínimo. Basándonos en la tabla resumen anterior podemos concluir que la constante 0.03432 podría ser la mejor opción, ya que ofrece un buen balance general para todos los sensores, con errores máximos y mínimos razonables. Tiene el mínimo error más bajo en algunos sensores y no tiene errores máximos extremadamente altos en comparación con los otros.

Por lo tanto, se recomienda utilizar la constante **0.03432** para el algoritmo TOA con triangulación basado en los resultados comparativos de los sensores.

3.5. Algoritmo TOF con estimación de distancias y un sensor láser VL6180X

En este caso, hemos estudiado este algoritmo acompañado de un sensor láser VL6180X que se acopla perfectamente a la placa de Arduino Mega 2560. Para la toma de medidas se ha ido variando la distancia del sensor láser con respecto al obstáculo.



Gráfica 49. Comparación de Valores Obtenidos VS Valores Reales para el sensor láser VL6180X.

En el gráfico, los puntos azules representan las observaciones de las distancias medidas en relación con los valores reales (distancia real), mientras que la línea roja punteada representa la relación ideal donde las distancias medidas serían exactamente iguales a los valores reales.

La relación entre las distancias medidas y las distancias reales parece ser bastante lineal, como indica la línea roja punteada, lo que sugiere que la medición es proporcional al valor real.

Hasta aproximadamente 15 cm, las mediciones parecen seguir muy de cerca la línea ideal, indicando alta precisión en las mediciones en este rango. Más allá de 15 cm, aunque las mediciones siguen siendo lineales, comienzan a alejarse

ligeramente de la línea ideal, lo cual sugiere un cambio en la precisión del instrumento de medición.

Si ahora hacemos diversos cálculos, se observa que la peor medida es la realizada a 22 cm, ya que tiene un error relativo del 15.9%.

Tabla 7. Valores para el Error Relativo máximo del sensor láser VL6180X.

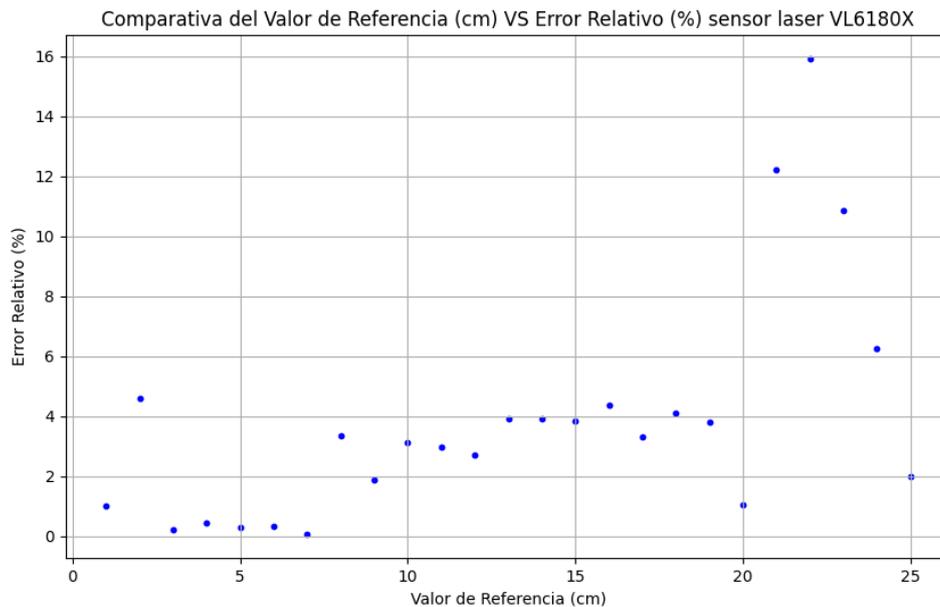
Archivo	Valor Referencia (cm)	Distancia Máxima	Distancia Mínima.	Distancia media	Error Relativo (%)
22.txt	22	25.5	25.5	25.5	15.90909091

De otro modo, obtenemos que la mejor medida es para 7 centímetros con un error relativo del 0.057142857 %.

Tabla 8. Valores para el Error Relativo mínimo del Sensor láser VL6180X.

Archivo	Valor Referencia (cm)	Distancia Máxima	Distancia Mínima	Distancia media	Error Relativo (%)
7.txt	7	7.4	6.6	6.996	0.027142857

Con este análisis ya se puede concluir que este método de medición es más preciso en valores de referencia bajos, pero si seguimos analizando detalladamente se observa que:



Gráfica 50. Comparativa del Valor de Referencia (cm) VS Error Relativo (%) para el Sensor láser VL6180X.

Los datos parecen agruparse más en la parte baja del gráfico, lo que indica que el error relativo es generalmente bajo cuando el valor de referencia es bajo. Sin embargo, hay una dispersión más notable a medida que el valor de referencia aumenta, indicando posiblemente una variabilidad más alta en el error con mayores distancias.

Hay una tendencia aparente de que a medida que el valor de referencia aumenta, también lo hace el error relativo. Esto es más notable particularmente después de los 15 cm, donde los puntos de datos se sitúan en la banda de error del 4%. Una vez superan los 20 cm, el error se dispara a valores más elevados.

Por otro lado, ahora nos centramos en analizar el tiempo de ida y vuelta (RTT, por sus siglas en inglés) el cual es una métrica fundamental en muchas aplicaciones de sensores, como la medición de distancia con el sensor VL6180X en un Arduino Uno, se obtienen otros resultados mostrados en las tablas siguientes. El RTT se refiere al tiempo que tarda una señal en ir desde el sensor hasta un objeto y volver al sensor. Este tiempo es crucial para calcular distancias con precisión.

Tabla 9. Valores para el RTT máximo y mínimo del Sensor láser VL6180X.

Archivo	Valor Referencia (cm)	RTT Máximo (ms)	RTT Mínimo (ms)	RTT Media (ms)
22.txt	22	31.42	26.79	27.339
7.txt	7	25.78	21.08	21.2212

Ambos conjuntos de datos muestran variaciones en los tiempos de RTT, aunque los valores de RTT en el archivo 7 cm son más consistentes en comparación con los del archivo 22 cm.

Los resultados obtenidos son erráticos y no son comparables con los obtenidos en el algoritmo TOF para estimación de distancias. Causas potenciales de mediciones incorrectas de RTT en el Arduino Uno con un sensor VL6180X:

- Reflexiones y absorciones indebidas. Si el objeto objetivo no es totalmente reflectante o si es de color oscuro, parte de la luz emitida por el sensor puede ser absorbida en lugar de reflejada de vuelta, afectando así la medición del tiempo de ida y vuelta.
- Interferencias externas. Las interferencias de luz ambiental, como la luz solar directa o fuentes de luz artificial fuertes, pueden afectar la señal del sensor. Este tipo de interferencia puede causar que el sensor registre tiempos de RTT incorrectos.
- Distancia del objetivo. Si el objeto está fuera del rango óptimo del sensor, las mediciones pueden volverse inestables o imprecisas. El sensor VL6180X tiene un rango de funcionamiento recomendado que debe respetarse para obtener mediciones precisas.
- Errores en el código que maneja la lectura de datos del sensor VL6180X pueden resultar en mediciones imprecisas, ya que solo es capaz de medir hasta milisegundos, no microsegundos, perdiendo así gran precisión de información.

En conclusión, el algoritmo TOF para estimación de distancias combinado con el sensor láser VL6180X demuestra una alta precisión en mediciones hasta aproximadamente 15 cm, siguiendo muy de cerca la línea ideal. Sin embargo, a medida que la distancia supera los 15 cm, la precisión disminuye y el error relativo aumenta, especialmente notable a 22 cm con un error del 15.9%. La mejor precisión se observa a 7 cm con un error relativo mínimo de 0.057%. Además, hemos de tener en cuenta que los análisis del tiempo de ida y vuelta (RTT) también muestran más consistencia en distancias más cortas, pero que factores como reflexiones indebidas, interferencias externas, distancia del objetivo y posibles errores en el código influyen decisivamente en un comportamiento errático de dichas mediciones, por ese motivo, desechamos su uso en este TFG. Por otro lado, Para el caso del algoritmo TOF con estimación de distancias, al igual que en casos anteriores, la idea principal era modificar algunos parámetros para alcanzar la mejor precisión, pero debido a los buenos resultados en las medidas de TOF a distancias cortas, se considera que no hace falta modificar ningún parámetro al respecto.

Por último, hemos de considerar que el algoritmo TOF para estimación de distancias combinado con el sensor láser VL6180X presenta una gran precisión en la estimación de distancias en el rango corto, particularmente entre 0 y 15 centímetros. Sin embargo, es fundamental tener en cuenta que este algoritmo viene acompañado de un coste elevado y enfrenta un desafío significativo relacionado con la sensibilidad a la luz ambiental.

La interferencia de la luz ambiental puede impactar negativamente en la capacidad del sensor TOF, como el VL6180X, para llevar a cabo mediciones precisas. La presencia de una intensa luz ambiental, especialmente en forma de luz infrarroja, puede perturbar el haz láser del sensor, causar reflexiones adicionales en los objetos a medir y, en casos extremos, saturar el sensor. Como resultado, las mediciones de distancia pueden ser inconsistentes o incorrectas en entornos con alta luminosidad.

Dado este desafío inherente a la sensibilidad del sensor TOF a la luz ambiental, se concluye que no se considera necesario llevar a cabo pruebas adicionales de triangulación.





4. CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN

4.1. Conclusiones

En apartados anteriores, hemos estudiado los resultados de diferentes algoritmos en diferentes circunstancias:

Tabla 10. Estructura de lo estudiado en este trabajo.

RSSI para estimación de distancias	RSSI para triangulación
TOA para estimación de distancias	TOA para triangulación
TOF para estimación de distancias	

Finalmente, se puede concluir:

Al utilizar un solo fotodetector, utilizando el algoritmo RSSI para estimación de distancias, se observó un comportamiento favorable en las frecuencias bajas, en comparación con intermedias o altas.

Bajas frecuencias (1 kHz - 20 kHz): la señal mostró una atenuación alta con la distancia, manteniendo niveles bajos de ruido a mayores distancias.

Frecuencias intermedias (50 kHz - 200 kHz): la señal presentó una menor variación, pero el aumento en el ruido captado a mayores distancias comprometió la utilidad de la señal.

Altas Frecuencias (500 kHz - 1000 kHz): Aunque se observó una menor variabilidad de la señal con la frecuencia y la distancia, los picos de voltaje debido al ruido hicieron que las señales útiles fueran difíciles de distinguir.

A pesar de un desempeño inicial favorable, el uso de un solo fotodetector no es adecuado para aplicaciones de alta precisión debido a la alta atenuación de la señal y la presencia de ruido a mayores distancias.

Al utilizar tres fotodetectores simultáneamente, haciendo uso del algoritmo RSSI mediante triangulación, la variabilidad de las señales aumentó significativamente. Los valores obtenidos eran inconsistentes y mostraban una gran cantidad de ruido, especialmente en distancias intermedias y largas.

Bajas Frecuencias (1 kHz - 20 kHz): los tres fotodetectores presentaron picos irregulares y fluctuaciones significativas, indicando una alta sensibilidad al ruido y cambios en el entorno.

Frecuencias Intermedias (50 kHz - 200 kHz): la señal mostró una tendencia más uniforme con un fotodetector en comparación con los otros dos, pero en general, la variabilidad y el ruido seguían siendo altos.

A altas frecuencias (500 kHz - 1000 kHz): la señal mostró menor atenuación, pero los valores altos de ruido dificultan la interpretación de las señales útiles.

El uso de tres fotodetectores no mejoró la precisión ni la fiabilidad de las mediciones. La alta variabilidad y el ruido significativo sugieren que esta configuración no es adecuada para aplicaciones que requieren señales consistentes y claras si no llevan aparejada alguna técnica de procesamiento de señal que reduzca el efecto del ruido indeseado.

La estimación de distancias usando el algoritmo Tiempo de Llegada (TOA) y ultrasonidos presentó una buena linealidad en las mediciones, especialmente en distancias medias. Sin embargo, el error relativo fue notablemente alto en distancias cortas, disminuyendo significativamente para valores superiores a 20 cm.

El algoritmo TOA es eficaz para distancias intermedias, pero presenta limitaciones en distancias muy cortas y largas. Se debe considerar la implementación de técnicas de calibración y filtrado para mejorar la precisión en estos rangos.

La constante 0.03483 se identificó como el mejor valor para el algoritmo TOA para estimación de distancias con un único dispositivo. Esta constante ofreció la mejor combinación de precisión y linealidad en las mediciones de distancia, con un rendimiento óptimo en distancias intermedias de 20 a 100 cm. Aunque se observaron algunos errores en las distancias muy cortas y largas, la constante 0.03483 proporcionó los resultados más consistentes y confiables.

El algoritmo TOA con triangulación mostró consistencia en sus mediciones utilizando tres sensores de ultrasonidos. La precisión fue alta en rangos de 20 a 100 centímetros, aunque se observaron errores significativos en las mediciones iniciales.

El TOA con triangulación es adecuado para aplicaciones de localización en tiempo real en distancias intermedias, pero como con los otros algoritmos, la precisión en distancias muy cortas necesita mejoras.

Por otro lado, la mejor constante para el algoritmo TOA con triangulación es 0.03432, si bien se observa que el comportamiento de los sensores no es uniforme siendo el mejor sensor evaluado el número 3. Esta combinación ofreció la mejor precisión y fiabilidad en las mediciones de distancia, especialmente en los rangos intermedios de 20 a 100 cm.

El sensor láser VL6180X con el algoritmo TOF, demostró una alta precisión en rangos cortos hasta 15 cm, con las mediciones siguiendo de cerca la línea ideal. Más allá de 15 cm, aunque las mediciones mantuvieron su linealidad, se alejaron ligeramente de la línea ideal, indicando un cambio en la precisión.

El algoritmo TOF ofrece alta precisión y rapidez en la medición de distancias, ideal para aplicaciones en tiempo real. Sin embargo, su sensibilidad a la luz ambiental puede afectar su rendimiento, si bien los sensores modernos suelen contar con tecnologías de mitigación. Aunque tenga mayor coste, los sensores TOF destacan por su exactitud y eficiencia en numerosos escenarios de medición.

El peor error relativo se observó a 22 cm con un 15.9%, mientras que la mejor medida fue a 7 cm con un 0.057%, lo que confirma que el algoritmo TOF para estimación de distancias es ideal para distancias bajas.

El algoritmo RTT (Round Trip Time) no fue adecuado para este proyecto debido a las limitaciones del hardware de Arduino. La plataforma Arduino no permite medir tiempos en microsegundos con la precisión necesaria para obtener resultados confiables con el algoritmo RTT. Esta limitación afecta significativamente la precisión de las mediciones de tiempo, haciendo que los

resultados sean imprecisos y poco fiables para aplicaciones de localización en tiempo real.

Se concluye que para estimación de distancias sería muy interesante el uso de un sensor láser VL6180X con el algoritmo TOF para distancias de 0 – 15 centímetros y, para distancias de 15 – 100 centímetros, se recomienda el uso de un sensor de ultrasonidos HC-SR04 con algoritmo TOA. Por tanto, implementar un dispositivo híbrido que implemente ambas tecnologías podría suponer unos resultados de estimación de distancias muy cercanos a la realidad con un coste no excesivamente elevado.

En contraste, en el caso de la triangulación, se contempla la posibilidad de reemplazar el sensor láser por fotodetectores con RSSI incorporando técnicas de procesamiento para corregir las desviaciones de señal indeseadas. Esta alternativa ofrece un coste considerablemente más reducido, aunque con la necesidad de un procesamiento adicional para mitigar la falta de precisión en la estimación de distancias.

4.2. Líneas futuras

Sería interesante enriquecer el script de automatización con una integración de inteligencia artificial, que simultáneamente analice los datos del osciloscopio digital mientras se realizan las mediciones. De esta manera, se logrará una mayor precisión y eficiencia en el proceso, fusionando tecnología avanzada con la automatización de tareas con el objetivo de obtener resultados en tiempo real.

También sería interesante, tal y como se ha planteado en las conclusiones, implementar algoritmos avanzados de procesamiento de señales y técnicas de aprendizaje automático para mejorar la precisión y fiabilidad de los sistemas de localización en tiempo real, principalmente cuando se trabaja con RSSI.

Como continuación a este trabajo fin de grado, se podría llevar a cabo el desarrollo de un hardware híbrido que combine un sensor de ultrasonidos para medir distancias de 15-100 centímetros y un sensor láser para la medida de distancias en rangos de 0-15 centímetros. Esta sinergia mejoraría la precisión y brindaría nuevas posibilidades para diversas aplicaciones.

Otras posibles mejoras podrían ser implementar técnicas de aprendizaje automático y procesamiento avanzado de señales para el algoritmo TOA, principalmente a cortas distancias. Los modelos de inteligencia artificial podrían ser entrenados para reconocer y corregir patrones de error específicos, mejorando así la fiabilidad del sistema en tiempo real.

Futuras investigaciones podrían centrarse en reducir el ruido y la interferencia que afectan las mediciones de TOA con triangulación. Esto incluye el desarrollo de técnicas de filtrado avanzadas y el uso de materiales que minimicen la interferencia electromagnética y acústica.

Con respecto a la tecnología Time-of-Flight (TOF) se plantea explorar las posibles aplicaciones prácticas. Ésta, representa un área de estudio interesante, ya que podría potenciar la precisión y versatilidad de las medidas de los sensores que utilizan TOF. Al comprender mejor cómo integrar TOF en sistemas de triangulación, podemos desbloquear todo su potencial y aprovecharlo para resolver desafíos complejos en ingeniería.

Al combinar tecnologías avanzadas, inteligencia artificial y un profundo entendimiento de los factores ambientales, estamos en camino hacia sistemas de localización más precisos, eficientes y robustos. La innovación y la dedicación a la mejora constante son las claves para avanzar en este fascinante campo de estudio.



5. ANEXOS

5.1. Código TOA con sensor de ultrasonidos HC-SR04 en Arduino IDE para estimación de distancias

```
#include <NewPing.h>

#define trigPin 2
#define echoPin 3
#define MAX_DISTANCE 100

NewPing sonar(trigPin, echoPin, MAX_DISTANCE);
float duration, distance;
int measurementsCount = 0; // Variable para contar las mediciones realizadas

void setup() {
  Serial.begin(9600);
}

void loop() {
  if (measurementsCount < 100) { // Realizar máximo 100 mediciones
    delay(50);

    duration = sonar.ping();
    distance = (duration / 2) * 0.034;

    Serial.print("");
    Serial.print(distance);
    Serial.println("");

    measurementsCount++; // Incrementar el contador de mediciones realizadas
```

```
}  
}
```

5.2. Código Python para gráficas TOA con estimación de distancias

```
import pandas as pd  
import matplotlib.pyplot as plt  
  
# carga de datos  
file_path = 'C:/Users/belen/Desktop/0.034 Gráficas/Graficas distancia cvs  
valor real unido/resultados.xlsx' # Update this path  
data = pd.read_excel(file_path, sheet_name='Sheet1')  
  
# dibujar  
plt.figure(figsize=(10, 6))  
plt.scatter(data['Valor de Referencia'], data['Error Relativo'], color='blue')  
  
# titulos  
plt.title('Error Relativo (%) VS Valor de Referencia para 0.034')  
plt.xlabel('Valor de Referencia (cm)')  
plt.ylabel('Error Relativo (%)')  
  
# Plot  
plt.grid(True)  
plt.show()
```

5.3. Código Python para cálculos TOA para estimación de distancias

```
import numpy as np  
import pandas as pd  
import glob  
import os
```

```

def process_file(file_path):
    # Extraer el valor de referencia del nombre del archivo
    file_name = os.path.basename(file_path)
    valor_referencia = float(os.path.splitext(file_name)[0])

    # Leer los datos y filtrar líneas vacías
    with open(file_path, 'r') as file:
        data = [float(line.strip()) for line in file if line.strip()]

    # Calcular los valores requeridos
    max_value = np.max(data)
    min_value = np.min(data)
    mean_value = np.mean(data)
    error_abs = abs(mean_value - valor_referencia)
    error_rel = (error_abs / valor_referencia) * 100

    return {
        'Archivo': file_name,
        'Valor de Referencia': valor_referencia,
        'Máximo': max_value,
        'Mínimo': min_value,
        'Media': mean_value,
        'Error Absoluto': error_abs,
        'Error Relativo': error_rel
    }

def process_all_files(directory):
    # Lista de archivos

```

```

file_paths = glob.glob(os.path.join(directory, '*.txt'))

# Inicializar una lista para almacenar los resultados
resultados = []

# Procesar cada archivo
for file_path in file_paths:
    resultado = process_file(file_path)
    resultados.append(resultado)

return resultados

# Directorio que contiene los archivos
directorio = 'C:/Users/belen/Desktop/0.034 Gráficas'

# Procesar todos los archivos y obtener los resultados
resultados = process_all_files(directorio)

# Convertir resultados a DataFrame
df = pd.DataFrame(resultados)

# Guardar resultados en un archivo Excel
df.to_excel('C:/Users/belen/Desktop/0.034 Gráficas/resultados.xlsx',
index=False)

print('Cálculos guardados en resultados.xlsx')

```

5.4. Código TOA con triangulación con tres sensores de ultrasonidos HC-SR04 en Arduino IDE

```
#include <NewPing.h>

#define trigPin1 9
#define echoPin1 10
#define trigPin2 4
#define echoPin2 5
#define trigPin3 12
#define echoPin3 13
#define MAX_DISTANCE 100

NewPing sonar1(trigPin1, echoPin1, MAX_DISTANCE);
NewPing sonar2(trigPin2, echoPin2, MAX_DISTANCE);
NewPing sonar3(trigPin3, echoPin3, MAX_DISTANCE);

float distance1, distance2, distance3;
int measurementsCount = 0; // Variable para contar las mediciones realizadas

void setup() {
  Serial.begin(9600);
}

void loop() {
  if (measurementsCount < 100) { // Realizar máximo 100 mediciones
    delay(50);

    // Realizar mediciones con los tres sensores
    unsigned int duration1 = sonar1.ping_median();
```

```

distance1 = (duration1 / 2.0) * 0.0353;

unsigned int duration2 = sonar2.ping_median();
distance2 = (duration2 / 2.0) * 0.0353;

unsigned int duration3 = sonar3.ping_median();
distance3 = (duration3 / 2.0) * 0.0353;

// Imprimir las distancias medidas por cada sensor
Serial.print(" ");
Serial.print(distance1);
Serial.print(" ; ");
Serial.print(distance2);
Serial.print(" ; ");
Serial.print(distance3);
Serial.println("");

measurementsCount++; // Incrementar el contador de mediciones realizadas
}
}

```

5.5. Código Python para gráficas TOA con triangulación

```

import matplotlib.pyplot as plt

import glob

import os

import numpy as np

import re

def graficar_valores_obtenidos_vs_reales(directorio_datos,
directorio_graficas):

```

```

# Patrón para buscar todos los archivos .txt en el directorio especificado
ruta_archivos = os.path.join(directorio_datos, '*.txt')

archivos = glob.glob(ruta_archivos)

# Intentar ordenar los archivos basándose en números extraídos del nombre
del archivo

try:
    archivos = sorted(archivos, key=lambda x: float(re.search(r"(\d+)",
os.path.basename(x)).group()))
except AttributeError:
    print(f"Error: No se pudo encontrar un número en el nombre de uno de
los archivos.")

    for archivo in archivos:
        if re.search(r"(\d+)", os.path.basename(archivo)) is None:
            print(f"Archivo sin número en el nombre: {archivo}")
        return # Terminar la ejecución si hay error

# Listas para almacenar todos los valores obtenidos y los valores reales
para cada sensor

valores_obtenidos_sensor1 = []
valores_obtenidos_sensor2 = []
valores_obtenidos_sensor3 = []
todos_valores_reales = []

# Procesar cada archivo

for archivo in archivos:
    # Extraer el número del archivo para el valor real usando expresión
regular
    valor_real = float(re.search(r"(\d+)",
os.path.basename(archivo)).group())

```

```

# Intentar leer los valores obtenidos del archivo
try:
    datos = np.loadtxt(archivo, delimiter=';', usecols=(0, 1, 2)) #
Ajuste para leer correctamente los datos
    valores_obtenidos_sensor1.extend(datos[:, 0])
    valores_obtenidos_sensor2.extend(datos[:, 1])
    valores_obtenidos_sensor3.extend(datos[:, 2])
    todos_valores_reales.extend([valor_real] * len(datos))
except Exception as e:
    print(f"Error al leer el archivo {archivo}: {e}")
    continue # Continuar con el siguiente archivo

# Verificar que hay datos para graficar
if not todos_valores_reales:
    print("No hay datos para graficar.")
    return

# Crear la gráfica para cada sensor
for i, sensor_values in enumerate([valores_obtenidos_sensor1,
valores_obtenidos_sensor2, valores_obtenidos_sensor3], start=1):
    plt.figure(figsize=(15, 10))
    plt.scatter(todos_valores_reales, sensor_values, alpha=0.5)
    plt.plot(range(1, int(max(todos_valores_reales))+1), range(1,
int(max(todos_valores_reales))+1), 'r--', label='Valor Real')
    plt.title(f'Comparación de valores obtenidos vs valores reales (Sensor
{i})')
    plt.xlabel('Valor Real (cm)')
    plt.ylabel(f'Valor Obtenido Sensor {i} (cm)')
    plt.legend()
    plt.grid(True)

```

```

# Verificar y crear el directorio si no existe
if not os.path.exists(directorio_graficas):
    os.makedirs(directorio_graficas)

# Guardar la gráfica
plt.savefig(os.path.join(directorio_graficas,
f'comparacion_valores_sensor{i}.png'), bbox_inches='tight')
plt.close()

# Rutas de los directorios
directorio_datos = r'C:/Users/belen/Downloads/353-20240421T171852Z-
001/353/Medidas'
directorio_graficas = r'C:/Users/belen/Downloads/353-20240421T171852Z-
001/353/DISTANCIA VS VALOR UNIDO'

# Llamar a la función
graficar_valores_obtenidos_vs_reales(directorio_datos, directorio_graficas)

```

5.6. Código Python para gráficas Error Relativo TOA con triangulación

```

import pandas as pd
import matplotlib.pyplot as plt

# Directorios
directorio_datos = r'C:\Users\belen\Desktop\08 TD0A-
ULTRASONIDOS\353\Calculos\estadisticas_sensores.csv'
directorio_salida = r'C:\Users\belen\Desktop\08 TD0A-
ULTRASONIDOS\353\Calculos'

# Cargar los datos

```

```

sensor_data = pd.read_csv(directorio_datos, sep=';')

# Crear una sola gráfica con los datos de los tres sensores
fig, ax = plt.subplots(figsize=(12, 6))

colors = ['red', 'blue', 'green']

sensor_labels = ['Sensor 1', 'Sensor 2', 'Sensor 3']

for i in range(3):
    color = colors[i]
    sensor_label = sensor_labels[i]

    ax.scatter(sensor_data['Valor Referencia (cm)'],
               sensor_data[f'{sensor_label} Error Relativo (%)'],
               color=color,
               label=sensor_label)

ax.set_title('Error Relativo % VS Valor de Referencia para Sensores (para la
constante 0.0353)', fontsize=14)
ax.set_xlabel('Valor Referencia (cm)', fontsize=12)
ax.set_ylabel('Error Relativo (%)', fontsize=12)
ax.grid(True)
ax.legend()

# Limitar el eje Y hasta el 50%
#ax.set_ylim(bottom=0, top=50)

# Ajustar layout y guardar como PNG
plt.tight_layout()

plt.savefig(f'{directorio_salida}\\Error_Relativo_Sensores_UpTo_50_prueba.png
')

plt.close(fig)

```

5.7. Código Python para cálculos TOA con triangulación

```
import matplotlib.pyplot as plt

import glob

import os

import numpy as np

import re

import pandas as pd

def calcular_estadisticas_con_errores(directorio_datos, archivo_salida):

    # Construye la ruta para los archivos de datos

    ruta_archivos = os.path.join(directorio_datos, '*.txt')

    # Obtiene la lista de archivos que coinciden con la ruta

    archivos = glob.glob(ruta_archivos)

    resultados = []

    for archivo in archivos:

        try:

            # Extrae el valor de referencia del nombre del archivo usando una

            # expresión regular

            valor_referencia = float(re.search(r"(\d+)",

            os.path.basename(archivo)).group())

            # Carga los datos del archivo, asumiendo que las columnas están

            # delimitadas por ';'

            datos = np.loadtxt(archivo, delimiter=';', usecols=(0, 1, 2))

            # Calcula los máximos, mínimos y medias para cada sensor

            maximos = np.max(datos, axis=0)

            minimos = np.min(datos, axis=0)

            medias = np.mean(datos, axis=0)

            # Calcula los errores absolutos y relativos en función de la media

            # y el valor de referencia
```

```

errores_absolutos = np.abs(medias - valor_referencia)

errores_relativos = (errores_absolutos / valor_referencia * 100)
if valor_referencia != 0 else np.inf

# Agrega los resultados a la lista
resultados.append({
    'Archivo': os.path.basename(archivo),
    'Valor Referencia': valor_referencia,
    'Sensor 1 Max': maximos[0],
    'Sensor 1 Min': minimos[0],
    'Sensor 1 Media': medias[0],
    'Sensor 1 Error Absoluto': errores_absolutos[0],
    'Sensor 1 Error Relativo (%)': errores_relativos[0],
    'Sensor 2 Max': maximos[1],
    'Sensor 2 Min': minimos[1],
    'Sensor 2 Media': medias[1],
    'Sensor 2 Error Absoluto': errores_absolutos[1],
    'Sensor 2 Error Relativo (%)': errores_relativos[1],
    'Sensor 3 Max': maximos[2],
    'Sensor 3 Min': minimos[2],
    'Sensor 3 Media': medias[2],
    'Sensor 3 Error Absoluto': errores_absolutos[2],
    'Sensor 3 Error Relativo (%)': errores_relativos[2]
})
except Exception as e:
    # Imprime un mensaje de error si ocurre un problema al procesar un
archivo
    print(f"Error al procesar el archivo {archivo}: {e}")

# Convierte la lista de resultados a un DataFrame de pandas

```

```

df_resultados = pd.DataFrame(resultados)

# Guarda el DataFrame en un archivo CSV con ';' como delimitador
df_resultados.to_csv(archivo_salida, index=False, sep=';')

print(f"Resultados guardados en {archivo_salida}")

# Cómo llamar a la función

directorio_datos = r'C:\Users\belen\Desktop\TFG\06 PRUEBAS ENERO\08 TOA CON
TRIANGULACIÓN-ULTRASONIDOS\353'

archivo_salida = r'C:\Users\belen\Desktop\TFG\06 PRUEBAS ENERO\08 TOA CON
TRIANGULACIÓN-ULTRASONIDOS\353\Calculos\estadisticas_sensores.csv'

# Llamar a la función

calcular_estadisticas_con_errores(directorio_datos, archivo_salida)

```

5.8. Código Senzor láser VL6180X en Arduino IDE para estimación de distancias

```

#include <Wire.h>
#include <VL6180X.h>

VL6180X sensor;

int measurementsCount = 0;

void setup()
{
  Serial.begin(9600);
  Wire.begin();

  sensor.init();
  sensor.configureDefault();
  sensor.setTimeout(1000);

```

```

}

void loop()
{
  if (measurementsCount < 100) {
    unsigned long start_time, end_time;

    // Registrar el tiempo actual
    start_time = micros();

    // Enviar el pulso ultrasónico
    int distance_mm = sensor.readRangeSingle(); // Leer la distancia en
milímetros

    // Registrar el tiempo nuevamente
    end_time = micros();

    // Calcular el tiempo de ida en microsegundos
    unsigned long travel_time = end_time - start_time;

    // Convertir el tiempo de ida a milisegundos
    float travel_time_ms = travel_time / 1000.0;

    // Convertir la distancia a centímetros
    float distance_cm = distance_mm / 10.0;

    // Calcular el RTT (ida y vuelta)
    float rtt = travel_time_ms * 2;

    // Imprimir la medida en centímetros acompañada de su RTT

```

```

Serial.print("Distancia: ");
Serial.print(distance_cm);
Serial.print(" cm\t RTT: ");
Serial.print(rtt);
Serial.println(" ms");

// Verificar si se agotó el tiempo de espera
if (sensor.timeoutOccurred() {
    Serial.print(" TIMEOUT");
}

measurementsCount++;
} else {
    // Si ya se han realizado 100 mediciones, salir del bucle loop
    while (true) {}
}
}
}

```

5.9. Código Python para gráficas Sensor Láser para estimación de distancias

```

import matplotlib.pyplot as plt
import glob
import os
import re
import numpy as np

def graficar_comparacion_distancias(directorio_datos, directorio_graficas):
    # para buscar todos los archivos .txt en el directorio especifico
    ruta_archivos = os.path.join(directorio_datos, '*.txt')

```

```

archivos = sorted(glob.glob(ruta_archivos), key=lambda x:
float(os.path.splitext(os.path.basename(x))[0]))

# extraer los datos de distancia
regex = re.compile(r'Distancia: (\d+\.\d+) cm')

# Listas para almacenar los valores medios de las distancias medidas y los
valores reales
distancias_medidas = []
distancias_reales = []

# Procesar cada archivo
for archivo in archivos:
    with open(archivo, 'r') as file:
        lineas = file.readlines()

        # Extraer el valor real de la distancia del nombre del archivo
        valor_real = float(os.path.splitext(os.path.basename(archivo))[0])
        distancias_reales.append(valor_real)

    distancias = []

    # Extraer los datos de distancia de cada línea
    for linea in lineas:
        match = regex.search(linea)
        if match:
            distancia = match.groups()[0]
            distancias.append(float(distancia))

```

```

        # Calcular el valor medio de las distancias medidas y añadirlo a la
lista
        if distancias:
            distancia_media = np.mean(distancias)
            distancias_medidas.append(distancia_media)
        else:
            distancias_medidas.append(0) # En caso de no encontrar datos,
añadir un 0

        # Crear una gráfica para comparar las distancias medidas con los valores
reales

        plt.figure(figsize=(15, 10))

        plt.scatter(distancias_reales, distancias_medidas, color='b',
label='Medida vs Real')

        plt.plot([min(distancias_reales), max(distancias_reales)],
[ min(distancias_reales), max(distancias_reales)], 'r--', label='Ideal')

        plt.title('Comparación de Distancias Medidas vs Valores Reales')
        plt.xlabel('Distancia Real (cm)')
        plt.ylabel('Distancia Media Medida (cm)')
        plt.legend()
        plt.grid(True)

        # Asegurarse de que el directorio de gráficas existe
        if not os.path.exists(directorio_graficas):
            os.makedirs(directorio_graficas)

        # Guardar la gráfica
        plt.savefig(os.path.join(directorio_graficas,
'comparacion_distancias.png'), bbox_inches='tight')

        plt.close()

```

```

# Rutas de los directorios

directorio_datos = 'C:/Users/belen/Desktop/TFG/06 PRUEBAS ENERO/10 SESNSOR
LASER'

directorio_graficas = 'C:/Users/belen/Desktop/TFG/06 PRUEBAS ENERO/10 SESNSOR
LASER/GRaficas Valor real VS Valor obtenido'

# Llamar a la función

graficar_comparacion_distancias(directorio_datos, directorio_graficas)

```

5.10. Código Python para gráficas Sensor láser VL6180X para estimación de distancias Valor de Referencia VS Error Relativo

```

import os

import pandas as pd

import matplotlib.pyplot as plt

def crear_grafica(archivo_datos, carpeta_salida):
    # Cargar los datos desde el archivo CSV
    data = pd.read_csv(archivo_datos, sep=';')

    # Filtrar datos para evitar infinidades en la gráfica
    filtered_data = data[data['Error Relativo (%)'] != float('inf')]

    # Crear gráfica
    plt.figure(figsize=(10, 6))

    plt.scatter(filtered_data['Valor Referencia (cm)'], filtered_data['Error
Relativo (%)'], color='blue', marker='o', s=10)

    plt.title('Comparativa del Valor de Referencia (cm) VS Error Relativo (%)
sensor láser VL6180X')

    plt.xlabel('Valor de Referencia (cm)')

    plt.ylabel('Error Relativo (%)')

```

```

plt.grid(True)

# Guardar la gráfica en el directorio deseado
output_path = os.path.join(carpeta_salida,
'error_relativo_vs_valor_referencia.png')
plt.savefig(output_path)
plt.close()
print(f"Gráfica guardada en: {output_path}")

# Rutas de los archivos
archivo_datos = r'C:\Users\belen\Desktop\TFG\06 PRUEBAS ENERO\10 SESNSOR
LASER\Calculos\calculos_sensor.csv'
carpeta_salida = r'C:\Users\belen\Desktop\TFG\06 PRUEBAS ENERO\10 SESNSOR
LASER\Calculos'

# Llamar a la función
crear_grafica(archivo_datos, carpeta_salida)

```

5.11. Código Python para cálculos Sensor láser VL6180X para estimación de distancias

```

import os
import glob
import numpy as np
import pandas as pd
import re

def procesar_archivos(directorio_datos, archivo_salida):
    ruta_archivos = os.path.join(directorio_datos, '*.txt')
    archivos = glob.glob(ruta_archivos)
    resultados = []

```

```

# Compilar las expresiones regulares para extraer los datos
distancia_re = re.compile(r"Distancia:\s*(\d+\.\d+)\s*cm")
rtt_re = re.compile(r"RTT:\s*(\d+\.\d+)\s*ms")

for archivo in archivos:
    try:
        valor_referencia = float(re.search(r"(\d+)",
os.path.basename(archivo)).group())

        with open(archivo, 'r') as f:
            distancias = []
            rttts = []

            for line in f:
                distancia_match = distancia_re.search(line)
                rtt_match = rtt_re.search(line)
                if distancia_match and rtt_match:
                    distancias.append(float(distancia_match.group(1)))
                    rttts.append(float(rtt_match.group(1)))

# Calcular estadísticas y errores
dist_media = np.mean(distancias)
resultados.append({
    'Archivo': os.path.basename(archivo),
    'Valor Referencia (cm)': valor_referencia,
    'Distancia Max': np.max(distancias),
    'Distancia Min': np.min(distancias),
    'Distancia Media': dist_media,
    'Error Absoluto': np.abs(dist_media - valor_referencia),
    'Error Relativo (%)': (np.abs(dist_media - valor_referencia) /
valor_referencia * 100) if valor_referencia != 0 else np.inf,

```

```

        'RTT Max': np.max(rtts),
        'RTT Min': np.min(rtts),
        'RTT Media': np.mean(rtts)
    })
except Exception as e:
    print(f"Error al procesar el archivo {archivo}: {e}")

# Guardar los resultados en un archivo CSV
df_resultados = pd.DataFrame(resultados)
df_resultados.to_csv(archivo_salida, index=False, sep=';')
print(f"Resultados guardados en {archivo_salida}")

# Rutas de los directorios
directorio_datos = r'C:\Users\belen\Desktop\TFG\06 PRUEBAS ENERO\10 SESNSOR
LASER'
archivo_salida = r'C:\Users\belen\Desktop\TFG\06 PRUEBAS ENERO\10 SESNSOR
LASER\Calculos\calculos_sensor.csv'

# Llamar a la función
procesar_archivos(directorio_datos, archivo_salida)

```



6. BIBLIOGRAFÍA

- Andrés Felipe García, C. G. (2015). Algoritmos de Radiolocalización basados en ToA,. *Revista Ingeniería y Región*, 2-4.
- AndroidPhoria. (30 de Septiembre de 2023). *AndroidPhoria*. Obtenido de <https://androidphoria.com/curiosidades/que-hace-el-sensor-tof>
- BricoGeek. (Septiembre de 2014). *BricoGeek*. Obtenido de https://tienda.bricogeek.com/arduino/306-arduino-mega-2560.html?gad_source=1&gclid=CjwKCAjw88yxBhBWEiWA7cm6pVwhB-w_cUIRNaQfibooVQ05S4hrozIM5-vqj9Yv8e3pfV0b1e-XhoCcAQQAvD_BwE
- Daboín Angela., V. G. (Abril 2012). *MEDICIÓN DE RSSI, LQI Y PRUEBAS DE COBERTURA PARA DIFERENTES*. Venezuela: Universidad Nacional Experimental Politécnica "Antonio José de Sucre".
- Jorge García Brea. (Junio de 2022). <http://dspace.umh.es/bitstream/11000/28556/1/TFG-Garc%C3%ADa%20Brea%2C%20Jorge.pdf>. Obtenido de Trabajo de Fin de Grado UMH: <http://dspace.umh.es/bitstream/11000/28556/1/TFG-Garc%C3%ADa%20Brea%2C%20Jorge.pdf>
- Jorge García Brea. (Junio de 2022). *TRABAJO FIN DE GRADO UMH*. Obtenido de <http://dspace.umh.es/bitstream/11000/28556/1/TFG-Garc%C3%ADa%20Brea%2C%20Jorge.pdf>
- Luis Llamas. (Junio de 2024). *Luis Llamas, Ingeniería, informática y diseño*. Obtenido de <https://www.luisllamas.es/arduino-sensor-distancia-vl53l0x/>
- Naylamp. (Septiembre de 2014). *Naylam Mechatronics*. Obtenido de <https://naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html>
- Rappaport, T. S. (2002). *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ: Prentice Hall.

RS-online. (Junio de 2015). *RS-online*. Obtenido de <https://es.rs-online.com/web/p/fototransistores/7082810>

RS-online. (2015 de Junio). *Placa de expansión STMicroelectronics X-Nucleo - RS*. Obtenido de <https://docs.rs-online.com/a63b/0900766b8147345e.pdf>

Salud., O. M. (8 de OCTUBRE de 2014). *Organización Mundial de la Salud*. . Obtenido de <https://www.who.int/es/news-room/fact-sheets/detail/electromagnetic-fields-and-public-health-mobile-phones>

Vishay . (Mayo de 2024). *rs-online*. Obtenido de rs-online: <https://docs.rs-online.com/80eb/0900766b80e2fb20.pdf>

Vishay. (Mayo de 2024). *rs-online*. Obtenido de rs-online: <https://docs.rs-online.com/ee92/0900766b817157b5.pdf>

