

**ESTIMACIÓN DE LOS PARÁMETROS ESTÁTICOS DE CORRECCIÓN DE LA  
SEÑAL ENTREGADA POR UN MAGNETÓMETRO ELECTRÓNICO DE TRES  
EJES**

MANUEL FERNANDO GALINDO SEMANATE



UNIVERSIDAD DEL CAUCA  
FACULTAD DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES  
DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL  
POPAYÁN  
2018



**ESTIMACIÓN DE LOS PARÁMETROS ESTÁTICOS DE CORRECCIÓN DE LA  
SEÑAL ENTREGADA POR UN MAGNETÓMETRO ELECTRÓNICO DE TRES  
EJES**

MANUEL FERNANDO GALINDO SEMANATE

Monografía Para Optar El Título De Ingeniero En Electrónica Y Telecomunicaciones

Director M.Sc. Lisbeth Lucía Martínez Ortega  
Codirector M.Sc. Elena Muñoz España

UNIVERSIDAD DEL CAUCA  
FACULTAD DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES  
DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL  
POPAYÁN  
2018



# CONTENIDO

---

	Pág.
<b>1. INTRODUCCIÓN.....</b>	<b>1</b>
1.1 PLANTEAMIENTO DEL PROBLEMA.....	1
1.2 OBJETIVOS.....	2
1.3 APORTE INVESTIGATIVO.....	2
1.4 METODOLOGÍA DE TRABAJO.....	3
<b>2. ESTADO DEL ARTE .....</b>	<b>6</b>
2.1 CONCEPTOS FUNDAMENTALES .....	6
2.1.1 Modelos de representación matemática de movimiento en el espacio.....	6
2.1.2 Campo magnético.....	9
2.1.3 Campo magnético terrestre.....	10
2.2 MAGNETÓMETROS.....	12
2.2.1 Técnicas de magnetometría.....	12
2.2.2 Capacidades y características de un magnetómetro.....	14
2.2.3 Tipos de magnetómetros.....	15
2.3 TRABAJOS RELACIONADOS.....	17
2.3.1 Fuentes de información.....	17
2.3.2 Criterios de búsqueda .....	18
2.3.3 Trabajos seleccionados .....	20
<b>3 DISEÑO DEL EXPERIMENTO .....</b>	<b>23</b>
3.1 CONSIDERACIONES SOBRE EL MAGNETÓMETRO.....	23
3.1.1 Magnetómetro Anisotrópico Magnetorresistivo.....	23
3.1.2 Efectos en un magnetómetro AMR .....	28
3.1.3 Modelo matemático .....	31
3.1.4 Especificaciones del sensor HMC5883L.....	32
3.2 CONDICIONES DE PRUEBA.....	33
3.2.1 Métodos de medición de prueba o calibración .....	33
3.2.2 Principio de medición.....	34
3.2.3 Método de medición.....	34
3.2.4 Proceso de medición.....	34
3.2.5 Método de calibración .....	35

---

3.3	PROCEDIMIENTO DEL EXPERIMENTO.....	35
3.3.1	Selección de entorno de medición.....	35
3.3.2	Arquitectura de hardware.....	35
3.3.3	Toma de muestras.....	36
3.3.4	Captura de datos y procesamiento.....	37
<b>4</b>	<b>DESARROLLO DE HERRAMIENTAS .....</b>	<b>38</b>
4.1	CONSTRUCCIÓN DE NODO.....	38
4.2	FIRMWARE DEL NODO.....	40
4.2.1	Configuración y lectura de sensores.....	40
4.2.2	Preprocesamiento de señal.....	42
4.2.3	Configuración del nodo.....	42
4.2.4	Protocolo de comunicación.....	44
4.3	PLATAFORMA ROTATIVA.....	48
4.4	APLICACIÓN WIMO.....	49
4.4.1	Panel de datos.....	49
4.4.2	Configuración.....	50
4.4.3	Optimización.....	50
<b>5</b>	<b>EVALUACIÓN.....</b>	<b>52</b>
5.1	COMPROBACIÓN DE EFECTO TÉRMICO.....	52
5.2	EVALUACIÓN DE RESPUESTA DE RUIDO.....	55
5.2.1	Prueba exterior.....	55
5.2.2	Prueba interior.....	59
5.2.3	Comparativa de resultados.....	62
5.3	CÁLCULO DE PARÁMETROS ESTÁTICOS.....	63
5.3.1	Toma de muestras.....	63
5.3.2	Compensación directa de corrimiento y ganancia.....	68
5.3.3	Adecuación a una esfera por mínimos cuadrados.....	70
5.3.4	Resultados.....	74
<b>6</b>	<b>CONCLUSIONES .....</b>	<b>80</b>
6.1	RESPECTO A LOS OBJETIVOS PLANTEADOS.....	80
6.2	APORTES Y EXPERIENCIAS.....	81
6.3	TRABAJOS FUTUROS.....	82
	<b>ANEXOS.....</b>	<b>89</b>

---

A - ESQUEMAS DE NODO DE SENSORES.....	89
B – INSTRUCTIVOS DE USO DEL NODO.....	92
C – DOCUMENTACIÓN DEL FIRMWARE EN EL NODO.....	97
D – SCRIPT DE CAPTURA DE DATOS “CAPTURAR.PY”.....	127
E – SCRIPT DE CORRECCIÓN TÉRMICA “THERMAL_ANALYSIS.M”.....	130
F – SCRIPT DE CÁLCULO DE PARÁMETROS “ELLIPSOIDS.M” .....	140
G – WIMO APP .....	149
H – PLATAFORMA ROTATIVA.....	158



## Lista de Tablas

---

Tabla 1. Relación de actividades y objetivos.....	5
Tabla 2. Comparativa de sensores magnéticos .....	17
Tabla 3. Palabras clave de búsqueda .....	19
Tabla 4. Resultados de búsquedas según criterios.....	19
Tabla 5. Descripción de variables del modelo matemático .....	31
Tabla 6. Características sensor HMC5883L.....	32
Tabla 7. Datos de sensores .....	41
Tabla 8. Parámetros de configuración de nodo.....	43
Tabla 9. Modos de ajuste con botón.....	43
Tabla 10. Datos de sensores – datagrama decodificado.....	45
Tabla 11. Parámetros de nodo - datagrama descodificado .....	46
Tabla 12. Recepción de parámetros - asignación de símbolos .....	47
Tabla 13. Promedios de desviación estándar de ruido en prueba exterior. ....	58
Tabla 14. Promedios de desviación estándar de ruido en prueba interior.....	61
Tabla 15. Comparativa de desviaciones estándar de ruido entre prueba exterior e interior. ....	62
Tabla 16. Factores de mejora en evaluaciones con los algoritmos de corrección .....	78

---



## Lista de Ilustraciones

---

Ilustración 1. Ángulos de Euler.....	6
Ilustración 2. Ejes de cardan en avión.....	7
Ilustración 3. Bloqueo de cardán en un avión.....	7
Ilustración 4. Desplazamiento del norte magnético con modelo CRGI.....	10
Ilustración 5. Elementos del campo magnético en WMM.....	11
Ilustración 6. Esquema SQUID.....	13
Ilustración 7. Esquema de una magnetorresistencia anisotrópica.....	23
Ilustración 8. Esquema básico de un sensor AMR de puente Wheatstone de un eje.....	24
Ilustración 9. Componentes de interacción en un AMR.....	25
Ilustración 10. Respuesta de magnetorresistencia anisotrópica básica.....	26
Ilustración 11. Estructura AMR de corriente a 45°.....	27
Ilustración 12. Respuesta de un sensor AMR en puente.....	28
Ilustración 13. Evaluación de magnetómetro 2D ideal.....	29
Ilustración 14. Efecto <i>Hard-Iron</i> 2D.....	29
Ilustración 15. Efecto <i>Soft-iron</i> 2D.....	29
Ilustración 16. Puente AMR Wheatstone.....	30
Ilustración 17. Ciclo de lectura HMC5883L.....	33
Ilustración 18. Proceso de medición.....	34
Ilustración 19. Arquitectura básica del hardware.....	36
Ilustración 20. Trazados de medición.....	36
Ilustración 21. Módulo GY-87.....	38

---

Ilustración 22. Módulo ESP-201 .....	38
Ilustración 23. Boceto 3D de nodo .....	39
Ilustración 24. Nodos fabricados.....	39
Ilustración 25. Datagrama IEEE 802.11 + UDP.....	44
Ilustración 26. Trama de recepción de parámetro .....	47
Ilustración 27. Plataforma rotativa .....	48
Ilustración 28. Eje de soporte .....	48
Ilustración 29. Panel de datos de aplicación .....	49
Ilustración 30. Panel de Configuración.....	50
Ilustración 31. Panel de optimización.....	51
Ilustración 32. Recipiente de prueba térmica.....	52
Ilustración 33. Wimo durante prueba .....	53
Ilustración 34. Variación de temperatura .....	53
Ilustración 35. Comportamiento térmico de magnetómetro.....	54
Ilustración 36. Magnetómetro en prueba térmica con compensación <i>alpha</i> activada. ....	55
Ilustración 37. Nodo acondicionado a prueba externa .....	55
Ilustración 38. Zona de prueba exterior .....	56
Ilustración 39. Temperatura en el nodo en prueba exterior. ....	57
Ilustración 40. Datos de magnetómetro en prueba exterior.....	57
Ilustración 41. Ruido extraído en prueba exterior .....	58
Ilustración 42. Histograma de señales de magnetómetro en prueba exterior. ....	59
Ilustración 43. Temperatura del nodo en prueba interior.....	60
Ilustración 44. Datos del magnetómetro en prueba interior. ....	60

---

Ilustración 45. Ruido extraído en prueba interior. ....	61
Ilustración 46. Histograma de señales de magnetómetro en prueba interior. ....	61
Ilustración 47. Detalle de señales en el histograma en prueba interior. ....	62
Ilustración 48. Histogramas de magnetómetro. ....	63
Ilustración 49. Cubo con sensores. ....	64
Ilustración 50. Cubo sobre plataforma. ....	64
Ilustración 51. Muestra de magnetómetro en plataforma. ....	65
Ilustración 52. Detalle de plano XZ. ....	66
Ilustración 53. Muestreo aleatorio de magnetómetro. ....	66
Ilustración 54. Cubo de sensores con masa metálica. ....	67
Ilustración 55. Respuesta del sensor a masa metálica. ....	67
Ilustración 56. Muestra aleatoria con la compensación. ....	69
Ilustración 57. Muestra en plano YZ con masa de metal. ....	69
Ilustración 58. Corrección Li/Griffiths en plano XY. ....	71
Ilustración 59. Corrección Vitali en plano XY. ....	72
Ilustración 60. Corrección Vitali en 3D. ....	73
Ilustración 61. Corrección Merayo en XY. ....	73
Ilustración 62. Corrección Petrov XY. ....	74
Ilustración 63. Configuración inicial <i>ellipsoids</i> . ....	76
Ilustración 64. Gráficas 3D generadas. ....	78
Ilustración 65. Gráfico comparativo de algoritmos. ....	79

---

## Glosario

---

<b>Access Point</b>	Término en inglés de Punto de Acceso, según el estándar IEEE 802.11 es un elemento de hardware de red que permite conectar una red de área local inalámbrica con una red cableada. Esta interfaz generalmente ofrece servicios de enrutamiento, asignación de direcciones IP o cifrado de red para que dispositivos clientes se conecten al punto de acceso formando un nodo de red.
<b>Actitud</b>	En aeronáutica, hace referencia a la posición de una nave u objeto sobre su propio eje respecto al horizonte.
<b>Alabeo</b>	En aeronáutica, es el movimiento rotacional sobre el eje longitudinal de un avión.
<b>AMR</b>	Siglas en inglés de magnetorresistencia anisotrópica, tipo de material metálico cuya resistencia varía con la presencia de un campo magnético.
<b>Anisótropo</b>	Según RAE (Real Academia Española), dicho de una sustancia o cuerpo: Que posee propiedades físicas distintas según la dirección en que se mide.
<b>Base64</b>	En informática, es un esquema de codificación de binario a texto en caracteres ASCII definido en la norma RFC 2045
<b>Cabeceo</b>	En aeronáutica, es el movimiento rotacional sobre el transversal de un avión.
<b>Datagrama</b>	En informática, es un paquete de datos binario organizado en bloques.
<b>Diamagnético</b>	Según RAE, dicho de un material: Que tiene menor permeabilidad magnética en el vacío, y es repelido por la acción de un fuerte imán.
<b>Double</b>	En informática, nombre para indicar un tipo de dato binario de coma flotante de 64bits definido según el estándar IEEE 754.
<b>EEPROM</b>	Siglas en inglés de Memoria de Solo Lectura Programable y Borrable Eléctricamente ( <i>Electrically Erasable Programmable Read-Only Memory</i> ). Es una memoria no volátil generalmente empleada en electrónica para almacenar datos de bajo volumen que no necesitan escritura constante.
<b>Exactitud</b>	Según VIM (Vocabulario Internacional de Metrología), proximidad entre un valor medido y un valor verdadero de un mensurando.
<b>Ferromagnetismo</b>	Según RAE, Propiedad de los materiales que, como el hierro, tienen muy alta permeabilidad magnética.
<b>Firmware</b>	En informática, término en inglés que hace referencia a un tipo de software de bajo nivel que controla lógicamente circuitos electrónicos de un dispositivo.

---

<b>Flash</b>	En informática, es un tipo de memoria EEPROM diseñada para el almacenamiento masivo de información digital.
<b>Geodesia</b>	Según RAE, ciencia matemática que tiene por objeto determinar la figura y magnitud del globo terrestre o de gran parte de él, y construir los mapas correspondientes.
<b>Guiñada</b>	En aeronáutica, es el movimiento rotacional sobre el eje vertical de un avión.
<b>Hardware</b>	Según RAE, equipo (conjunto de aparatos de una computadora)
<b>I<sup>2</sup>C</b>	Siglas en inglés de Circuito Inter-Integrado ( <i>Inter-Integrated Circuit</i> ), es un bus de datos en serie desarrollado por Philips utilizado principalmente para la comunicación entre distintas partes de un circuito electrónico.
<b>IMU</b>	Siglas en inglés de Unidad de Medición Inercial ( <i>Inertial Measurement unit</i> ), es un dispositivo electrónico empleado en navegación que informa sobre la velocidad, orientación y fuerzas gravitacionales. Se compone de acelerómetros y giroscopios.
<b>Inductancia</b>	Según RAE, relación entre la fuerza electromotriz producida en una bobina y la variación de la corriente, y que se mide en henrios.
<b>IP</b>	Siglas en inglés de Protocolo de Internet ( <i>Internet Protocol</i> ), suele usarse el término IP para referirse a una dirección IP, que es un número lógico que identifica a una interfaz dentro de una red.
<b>Latencia</b>	En informática, se refiere al tiempo que transcurre entre la emisión de un paquete de datos y la recepción.
<b>Magnetostricción</b>	Según RAE, cambio de las dimensiones de un material ferromagnético por la acción de un campo magnético.
<b>Maxwell, Ecuaciones de</b>	Son un conjunto de ecuaciones que describen por completo los fenómenos electromagnéticos reunidos por el científico James Clerk Maxwell
<b>MEMS</b>	Siglas en inglés de Sistemas Microelectromecánicos ( <i>Microelectromechanical systems</i> ), son dispositivos de tecnología electromecánica construidos en escala nanométrica. Suelen ser llamados también micro máquinas o micro sistemas.
<b>Mensurando</b>	Según VIM, magnitud que sea medir.
<b>Metrología</b>	Según VIM, Ciencia de las mediciones y sus aplicaciones.
<b>Mezanine</b>	En electrónica, interfaz electromecánica que permite ubicar un circuito impreso sobre otro.
<b>Momento</b>	Según RAE, en física, cantidad de movimiento. Momento de inercia: Suma de los productos que resultan de multiplicar la masa de cada elemento de un cuerpo por el cuadrado de su distancia a un eje de rotación. Momento de una fuerza: Magnitud resultante del producto del valor de una fuerza por su distancia a un punto de referencia. Momento lineal: cantidad de movimiento.

<b>Ohm, Ley de</b>	Postulación del científico Georg Simon Ohm, establece que la diferencia de potencial de un conductor es proporcional al producto de la intensidad de corriente que circula por la resistencia eléctrica del material.
<b>Piezolectricidad</b>	Según RAE, Propiedad que tienen ciertos cristales de polarizarse eléctricamente cuando son sometidos a presión, y a la inversa.
<b>Precisión</b>	Según VIM, proximidad entre las indicaciones o los valores medidos obtenidos en mediciones repetidas de un mismo objeto, o de objetos similares, bajo condiciones específicas.
<b>RAM</b>	Siglas en inglés de Memoria de Acceso Aleatorio (Random Access Memory), es una memoria volátil utilizada en computadoras para almacenar datos e instrucciones de forma temporal.
<b>Resistencia</b>	Según RAE, magnitud que mide la resistencia y cuya unidad en el sistema internacional es el ohmio. Determina la oposición al paso de corriente eléctrica de un material conductor.
<b>RPM</b>	Siglas de Revoluciones Por Minuto, es una unidad de frecuencia que mide el número de rotaciones completadas en un minuto por un cuerpo alrededor de un eje.
<b>Software</b>	Según RAE, conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.
<b>TAM</b>	Siglas en inglés de Magnetómetro de Tres ejes ( <i>Triaxial magnetometer</i> ), dispositivo que mide campo magnético en tres ejes espaciales.
<b>TCP</b>	Siglas de Protocolo de Control de Transmisión ( <i>Transmission Control Protocol</i> ), es un protocolo utilizado sobre el Protocolo de Internet empleado para transmitir datagramas con control de flujo definido en la norma RFC 1122
<b>Torque</b>	En física, Momento de fuerza.
<b>UDP</b>	Siglas en inglés de Protocolo de Datagramas de Usuario ( <i>User Datagram Protocol</i> ), al igual que TCP es un protocolo utilizado sobre el Protocolo de Internet, permite el envío de datagramas sin control o conexión en este nivel, está definido en la norma RFC 768
<b>Wheatstone, Puente de</b>	Es un circuito eléctrico de cuatro resistencias alimentado por una fuente de voltaje constante. Es utilizado para medir una resistencia desconocida.
<b>WiFi</b>	Siglas en inglés de Campo Inalámbrico ( <i>Wireless Field</i> ), es el nombre comercial utilizado para la tecnología de red local inalámbrica definida en la familia de estándares IEEE 802.11

\* RAE: Real Academia Española. VIM: Vocabulario Internacional de Metrología.

---

# 1. INTRODUCCIÓN

---

Hoy se está presenciando lo que los expertos denominan la tercera revolución industrial, sus inicios están marcados por tecnologías que han permitido el desarrollo de sistemas inteligentes capaces de mejorar la realización de tareas, que representa una reducción en las emisiones de gases contaminantes y de efecto invernadero. Como respuesta a esto, la industria ha enfatizado en la miniaturización de sistemas electrónicos, gracias a ello se logra construir equipos que demandan mucha menos energía y a la vez aumentan su rango de aplicación en cualquier sector. De lo anterior, parte la importancia de realizar investigaciones sobre el uso de estas nuevas tecnologías, como base de conocimiento para nuevas aplicaciones en escenarios donde la electrónica no se ha considerado como herramienta; esta es la motivación para realizar este trabajo, obviamente no es posible abarcar un tema tan amplio en una sola investigación.

En este caso, el documento está centrado en el uso de un magnetómetro electrónico de tres ejes, un sensor de un tamaño de 3mm por 3mm, diseñado para medir el campo magnético de la Tierra. A pesar de su reducido tamaño sus capacidades son amplias, ha sido evaluado en investigaciones como parte de sistemas de navegación de drones y satélites, detector de metales, medidor de tráfico o detector de tormentas solares. Sin embargo, dentro de esas investigaciones se hace constante una pobre descripción respecto a los procesos de captura de datos, los problemas que puede tener un sensor de este tipo durante las mediciones y el tratamiento para intentar solventarlos, por esta razón, este trabajo está orientado como un proceso descriptivo en el uso de un magnetómetro de tres ejes, en particular el modelo HMC5883L de Honeywell.

## 1.1 PLANTEAMIENTO DEL PROBLEMA

Los sistemas microelectromecánicos (MEMS, en inglés *Micro Electro Mechanical Systems*) están tomando relevancia dentro del mercado electrónico en el último tiempo gracias a su tamaño, bajo consumo y costo. Son sistemas construidos con técnicas de fotolitografía en tres dimensiones, similar a las utilizadas en circuitos integrados, con la diferencia de tener una construcción interna que permite una actuación mecánica. En el caso de este trabajo, se orienta a un magnetómetro MEMS de tres ejes, un sensor útil para sistemas de navegación y control.

El principio de operación de un magnetómetro MEMS tiene dos vertientes: La primera consiste en detección electrónica, utilizando un componente piezorresistivo o transductor electrostático que mide la fuerza de Lorentz. El segundo método es óptico, aplicando el efecto Faraday en un haz de luz láser donde se miden los cambios de polarización, aunque la complejidad de este método lo hace poco práctico a nivel comercial. Debido a esto último, el método más popular en distintas aplicaciones es el electrónico, con magnetorresistores anisotrópicos (AMR, en inglés *Anisotropic MagnetoResistance*), que es un tipo de piezorresistor útil para medir campos electromagnéticos o magnéticos dependiendo del compuesto químico empleado [1].

Los AMR utilizan compuestos ferrosos como níquel-hierro que naturalmente tienen el espín de sus electrones en un ángulo que produce una resistividad más alta, cuando este material se expone a un campo magnético la resistencia baja proporcionalmente a la cantidad de electrones

---

que se alinean al campo y sufren un cambio en su espín. Estos sensores suelen empaquetarse en circuitos integrados digitales, esto significa el uso de conversores analógico-digital que generan ruido, además tanto el conversor como el compuesto del AMR sufren variaciones de medición por la temperatura [2].

Cuando se emplea un magnetómetro de tres ejes (MTE) se espera obtener un vector que mide la magnitud y dirección de la línea de campo sobre la que se encuentra el sensor. A los errores que se presentan en la medición como el ruido, efecto térmico o defectos de fabricación se suman las diferencias que existen en los valores obtenidos de cada eje. Idealmente al caracterizar un sensor MTE con un campo constante y rotación sobre su mismo eje, la información de magnitud del campo debería ser igual, independiente de la posición del sensor. Pero dichas condiciones en la práctica no se dan, los factores que producen estos errores presentan características estáticas y dinámicas con componentes no lineales. Lo que implica un proceso de caracterización de, por lo menos, los factores más influyentes y la aplicación de tratamiento de señales para acondicionar información obtenida de un sensor MTE [3].

Entonces, si se desea mejorar la medición de un campo magnético considerando los factores más significativos ¿Cuáles son las características de un sistema para la estimación de los parámetros estáticos de corrección de la señal entregada por un magnetómetro electrónico de tres ejes?

## 1.2 OBJETIVOS

### Objetivo general

Proponer un sistema para la estimación de los parámetros estáticos de corrección de la señal entregada por un magnetómetro electrónico de tres ejes.

### Objetivos específicos

- Definir el modelo matemático de referencia y los parámetros estáticos asociados a este.
- Plantear los mecanismos necesarios para estimar los parámetros estáticos del modelo seleccionado.
- Evaluar el sistema propuesto en un entorno controlado con la asistencia de herramientas computacionales.

## 1.3 APORTE INVESTIGATIVO

El carácter descriptivo de esta investigación permitirá obtener información como:

- Equipamiento o condiciones de laboratorio mínimos necesarios para estimar los parámetros estáticos del magnetómetro MEMS seleccionado.
- Margen de mejora de la señal del magnetómetro con los parámetros estimados.



- Un conjunto entre software y hardware con potencial de aplicación en otros proyectos que empleen un tipo de sensor igual o similar al seleccionado en este trabajo.

## 1.4 METODOLOGÍA DE TRABAJO

Para el desarrollo de este proyecto se ha optado por la metodología definida en el “*Modelo Integral para un Profesional en Ingeniería*” [4], considerando la necesidad de adaptarlo a contexto de sistemas hardware-software. Dentro de ese modelo se identifican varios componentes ordenados: Construcción de una base de conocimiento aplicando el “*Modelo de Investigación Documental*” [5], posteriormente, se emplea un proceso orientado por una aproximación a una Metodología para Desarrollo para Sistemas Embebidos [6] que integra las consideraciones tanto en software como en hardware; finalmente la solución elaborada se evalúa a través del proceso de ejecución y análisis DESMET [7].

En cuanto al “*Modelo de Investigación Documental*”, este propone una serie de lineamientos adaptables a las condiciones del proyecto. Estos responden a cuatro fases:

**Fase Preparatoria:** Proceso en que se identifican y aclaran los conceptos principales como: MEMS, magnetómetro, modelos matemáticos y asociados.

**Fase Descriptiva:** Una vez identificados los elementos del núcleo de conocimiento, se realiza un proceso de estudio sobre el estado de dichas temáticas. Esto permite diferenciar la línea de investigación, además, establecer la disponibilidad de herramientas y algoritmos que apoyen el desarrollo del proyecto, en particular sobre el procesamiento de datos, mejoramiento de señales y modelos matemáticos de los parámetros estáticos del magnetómetro. Con la identificación correspondiente será posible integrarlos a los objetivos del proyecto.

**Fase de construcción teórica global:** Se establece la información descriptiva de la solución propuesta, ejecutando un balance para verificar limitaciones, problemas y logros en el desarrollo del proyecto.

**Fase de extensión y publicación:** Fase final en la que se hacen públicos los resultados del proyecto a través de mecanismos como la monografía, publicación de artículos y sustentación del proyecto.

Respecto a la aproximación *Metodología para Desarrollo para Sistemas Embebidos*, se emplea como referencia un sumario de técnicas que describen el modelamiento de la arquitectura, la programación, la orientación a tiempo real, análisis basado en modelamiento UML, entre otras. No existe una metodología rigurosa y específica debido a las variaciones técnicas y a la incertidumbre inicial [8] [9]. Para la ejecución se puede recurrir a componentes de metodologías de software como Scrum [10], XP [11], y de gestión de proyectos como PMI [12] sin necesidad de aplicarlas en rigor. Gracias a experiencias previas tanto en el campo académico como el laboral por parte del autor, se definen las actividades necesarias en paquetes de trabajo como recomienda PMI.

---

Por otra parte, DESMET es una metodología para la evaluación del proceso de desarrollo orientada a la ingeniería de software. En general, incluye la definición de la hipótesis de evaluación, variables, participantes, instrumentos, aplicación de instrumentos y análisis de información. Estas herramientas en especial permitirán evaluar la calidad de la información recabada por el sistema y establecer hasta qué punto se está mejorando la información obtenida del magnetómetro.

Atendiendo a lo anterior se han clasificado cinco paquetes de trabajo (*work package*) que recogen las actividades necesarias para la concreción de esta investigación, dichos paquetes son los siguientes:

**W.P.1** Revisión documental:

En esta actividad se realizará la recolección de información del estado del arte, esto será el insumo para definir el modelo matemático de referencia que se empleará en las evaluaciones con el magnetómetro.

**W.P.2** Planteamiento del sistema:

Con el modelo matemático definido, se eligen los parámetros que se desean medir del magnetómetro, para ello se planea el mecanismo que facilite la medición o cálculo de dichos parámetros.

**W.P.3** Desarrollo de hardware:

Se desarrolla el sistema planteado con los elementos hardware que se consideren necesarios.

**W.P.4** Evaluación:

En esta parte también se requiere un desarrollo, en este caso de software, que complemente y facilite la gestión de información de las mediciones obtenidas del sensor. Esta fase se caracteriza por una sucesión metódica de experimentos de medición para obtener información útil para su análisis.

**W.P.5** Documentación:

Todo el conocimiento generado se ordena y se presenta.

Estos paquetes guardan su relación con las actividades y los objetivos secundarios a cumplir, la Tabla 1 resume estas relaciones.

Con estos procesos se ha dado cumplimiento con los objetivos propuestos, los resultados de estas actividades están reflejados en los siguientes capítulos.

Tabla 1. Relación de actividades y objetivos

Objetivo	W.P.	Actividad	Herramienta	Entregable
Definir el modelo matemático de referencia y los parámetros estáticos asociados a este	1	1.1 Revisión documental	<i>Modelo de Investigación Documental</i> , Motores de búsqueda	Bibliografía clasificada
		1.2 Selección de modelo matemático	Análisis comparativo	Modelo y parámetros de referencia a emplear
Plantear los mecanismos necesarios para estimar los parámetros estáticos del modelo seleccionado	2	2.1 Selección de hardware de evaluación	Bibliografía clasificada	Componentes de sistema definidos
		2.2 Selección de entorno de pruebas	Bibliografía clasificada	Análisis de condiciones ambientales de la prueba
		2.3 Definición de modelo de pruebas	Bibliografía y componentes de sistema	Descripción del procedimiento para estimación de parámetros
Evaluar el sistema propuesto en un entorno controlado con la asistencia de herramientas computacionales.	3	3.1 Desarrollo de hardware de medición	<i>Metodología para Desarrollo para Sistemas Embebidos</i> , Documentación, s.w. de desarrollo	Hardware para la recolección de datos y su firmware
		3.2 Implementación de aplicativo de datos	Documentación, s.w. de desarrollo	Componentes software para la recepción y gestión de datos
		3.3 Consecución de herramientas adicionales	Bibliografía, documentación adicional	Hardware adicional necesario para pruebas
	4	4.1 Evaluación del modelo de pruebas	DESMET, Entregables anteriores	Base de datos de pruebas
		4.2 Análisis de datos	DESMET, Bases de datos de pruebas	Conclusiones sobre la propuesta
	5	Documentación final	Bibliografía y conclusiones	Artículo científico y monografía

s.w.: software

## 2. ESTADO DEL ARTE

---

Para tener una mejor comprensión de los parámetros y procedimientos requeridos en el proceso de caracterización del magnetómetro, es necesario contar con una definición de las tecnologías y conocimientos asociados. Esto asegura un mejor dimensionamiento de este trabajo y reduce la incertidumbre sobre los pasos a seguir en el desarrollo de la investigación.

### 2.1 CONCEPTOS FUNDAMENTALES

En esta sección se describen algunos fundamentos matemáticos y físicos que son tenidos en cuenta a lo largo de la investigación.

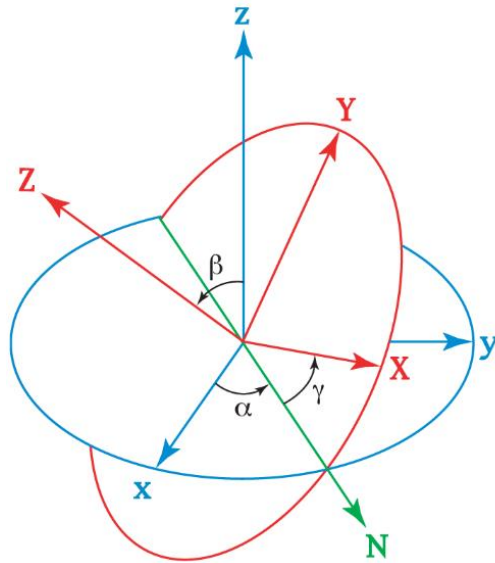
#### 2.1.1 Modelos de representación matemática de movimiento en el espacio

En los sistemas de navegación espacial, es decir de tres dimensiones, se emplean generalmente dos tipos de sistema de referencia para determinar la orientación de un objeto, lo que también se conoce como matriz de rotación. El primer sistema se le llama ángulos de Euler y el segundo como cuaterniones.

##### 2.1.1.1 Ángulos de Euler

Consiste en un sistema de tres dimensiones con ángulos ortogonales móviles respecto de una referencia de planos ortogonales inmóviles formados por los planos XY, XZ y YZ. Los ángulos formados por el movimiento del objeto respecto al plano se suelen llamar  $\alpha$ ,  $\beta$  y  $\gamma$  [13]. En la Ilustración 1 se observa la asignación de estos ángulos respecto a los planos.

Ilustración 1. Ángulos de Euler

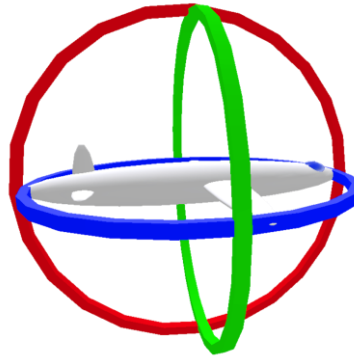


Fuente: [14]

---

Este sistema tiene como ventaja una fácil comprensión y representación por la proporcionalidad entre los ángulos y el movimiento, sin embargo presenta un problema llamado bloqueo de cardán; sucede cuando dos de sus tres ejes de libertad quedan en paralelo suprimiendo uno de estos. El bloqueo de cardán no significa un bloqueo físico de los ejes, estos continúan libres, pero uno de ellos respecto al otro no lo está [15]. En la Ilustración 2 se observa el modelo de un avión con sus tres ejes de cardán libres. Cuando se emplean los ángulos de Euler para representar la posición espacial de un avión reciben un nombre para cada eje: el cabeceo (*pitch*) que mide el ángulo elevación o caída de la cabeza respecto al horizonte, representado por el eje rojo. El segundo se le conoce como alabeo (*roll*), que indica el ángulo de inclinación de las alas respecto al eje transversal del avión visto desde la parte trasera, en la imagen representado por el eje verde. Finalmente, la guiñada (*yaw*) es el eje que mide el cambio de curso del avión respecto al norte, es decir una rotación sobre su propio eje visto desde arriba, se representa en el gráfico por eje azul.

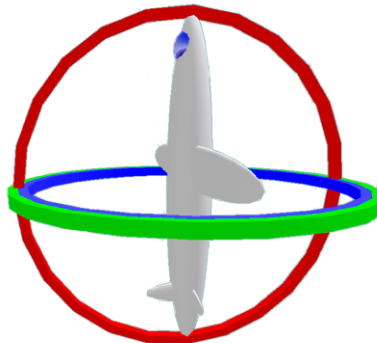
Ilustración 2. Ejes de cardan en avión



Fuente: [16]

Cuando el avión se eleva verticalmente, es decir su cabeceo es de  $90^\circ$ , sus ejes de alabeo y guiñada se ponen en paralelo como se observa en Ilustración 3, generando un bloqueo de cardán, ya que el movimiento de uno de estos dos ejes se afectará directamente.

Ilustración 3. Bloqueo de cardán en un avión



Fuente: [17]

No obstante, con que el avión simplemente cambie el ángulo de cabeceo a un valor distinto de  $90^\circ$  o  $-90^\circ$  los ejes de alabeo y guiñada de nuevo serán independientes, además el avión en ningún momento perdió su capacidad de movimiento, simplemente su notación matemática entró en una ambivalencia.

### 2.1.1.2 Cuaterniones

Los cuaterniones son hipercomplejos, una extensión de los números reales, añadiendo las unidades imaginarias  $i, j$  y  $k$  a los números reales tales que [18]:

$$\begin{aligned} q &= q_w + iq_x + jq_y + kq_z \\ i^2 &= j^2 = k^2 = -1 \\ ij &= k = -ji \\ jk &= i = -kj \\ ki &= j = -ik \end{aligned} \quad (2.1)$$

Gracias a esta representación es posible describir matemáticamente el movimiento angular de un objeto sin que se presente el bloqueo del cardán. El único problema es que convierte una información tridimensional a cuatro dimensiones, esto implica un costo computacional, más cuando se necesitan realizar transformaciones entre ángulos de Euler a cuaterniones y viceversa. En este trabajo se adopta la convención de Euler de NASA [19], esta descripción es ampliamente utilizada en sistemas inerciales y por ello se adapta bien a las necesidades aquí expuestas, la convención se muestra en la ecuación (2.2).

$$\begin{aligned} \mathbf{q} &= [q_0 \ q_1 \ q_2 \ q_3]^T = [q_w \ q_x \ q_y \ q_z]^T \\ |\mathbf{q}|^2 &= q_0^2 + q_1^2 + q_2^2 + q_3^2 = q_w^2 + q_x^2 + q_y^2 + q_z^2 = 1 \end{aligned} \quad (2.2)$$

Un cuaternion se relaciona con los ejes de Euler empleando las expresiones de la ecuación (2.3).

$$\begin{aligned} q_0 &= q_w = \cos\left(\frac{\alpha}{2}\right); \\ q_1 &= q_x = \sin\left(\frac{\alpha}{2}\right) \cos(\beta_x); \\ q_2 &= q_y = \sin\left(\frac{\alpha}{2}\right) \cos(\beta_y); \\ q_3 &= q_z = \sin\left(\frac{\alpha}{2}\right) \cos(\beta_z) \end{aligned} \quad (2.3)$$

Donde  $\alpha$  es un ángulo de rotación simple, mientras que los cosenos  $\beta$  indican la ubicación de los ejes de Euler [19]. Partiendo de estas definiciones, se establecen las ecuaciones de conversión entre Euler y cuaterniones con las ecuaciones (2.4) y (2.5).

$$\begin{aligned} C_r &= \cos(0.5 * \phi), & S_r &= \sin(0.5 * \phi) \\ C_p &= \cos(0.5 * \theta), & S_p &= \sin(0.5 * \theta) \\ C_y &= \cos(0.5 * \psi), & S_y &= \sin(0.5 * \psi) \end{aligned} \quad (2.4)$$


---

Donde:  $\phi$  es el ángulo de alabeo,  $\theta$  el ángulo de cabeceo y  $\psi$  el ángulo de guiñada. Luego los componentes del cuaternion w, x, y, z son:

$$\begin{aligned} \mathbf{q}_0 = \mathbf{q}_w &= C_y C_r C_p + S_y S_r S_p \\ \mathbf{q}_1 = \mathbf{q}_x &= C_y S_r C_p - S_y C_r S_p \\ \mathbf{q}_2 = \mathbf{q}_y &= C_y C_r S_p + S_y S_r C_p \\ \mathbf{q}_3 = \mathbf{q}_z &= S_y C_r C_p - C_y S_r S_p \end{aligned} \quad (2.5)$$

La operación inversa, es decir convertir de cuaternion a ángulos de Euler [20], requiere de la aplicación de la ecuación (2.6).

$$\begin{bmatrix} roll \\ pitch \\ yaw \end{bmatrix} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} atan2(2(q_w q_x + q_y q_z), 1 - 2(q_x^2 + q_y^2)) \\ asin(2(q_w q_y - q_z q_x)) \\ atan2(2(q_w q_z + q_x q_y), 1 - 2(q_y^2 + q_z^2)) \end{bmatrix} \quad (2.6)$$

### 2.1.2 Campo magnético

El campo magnético es un conjunto de vectores que describen la influencia magnética producida por una corriente eléctrica o un material magnético cercano. Existen distintos tipos de fuentes de campos magnéticos naturales como cobalto, hierro, níquel o tierras raras como el neodimio, que por sus propiedades atómicas pueden ser fuentes permanentes de campo magnético [21]. También, las interacciones de materiales producen campos naturales como lo es el caso del núcleo de la Tierra, que debido a la presión y rotación produce el campo magnético que protege al planeta de partículas emitidas del sol.

También el ser humano es causante de muchas emisiones de campo magnético a través de dispositivos como sistemas de comunicación, motores eléctricos e imanes. Además, la corriente eléctrica genera campos entorno a las líneas de transmisión, transformadores y equipos eléctricos o electrónicos. Lo cual significa un serio problema en la detección de campos naturales debido a la constante presencia de ruido en los sensores magnéticos.

El magnetismo también tiene manifestaciones físicas en interacciones con cargas eléctricas en lo que se conoce como electromagnetismo. Cuando un campo magnético actúa sobre una carga en movimiento, en una región del espacio, produce un efecto conocido como Fuerza de Lorentz y permite medir el campo magnético empleando la ecuación (2.7).

$$\mathbf{F} = q \cdot (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \quad (2.7)$$

En su definición la fuerza de Lorentz  $\mathbf{F}$  es producida cuando una carga puntual  $q$  atraviesa un campo magnético  $\mathbf{B}$  a una velocidad  $\mathbf{v}$  inicial y sumado a la fuerza de un campo eléctrico  $\mathbf{E}$  si lo hay. La fuerza de Lorentz es también conocida como fuerza magnética, inducción magnética o densidad de flujo magnético [22]. En el Sistema Internacional de Unidades el campo magnético se mide en la unidad Tesla (T) [23], su magnitud se calcula como se muestra en la ecuación (2.8).

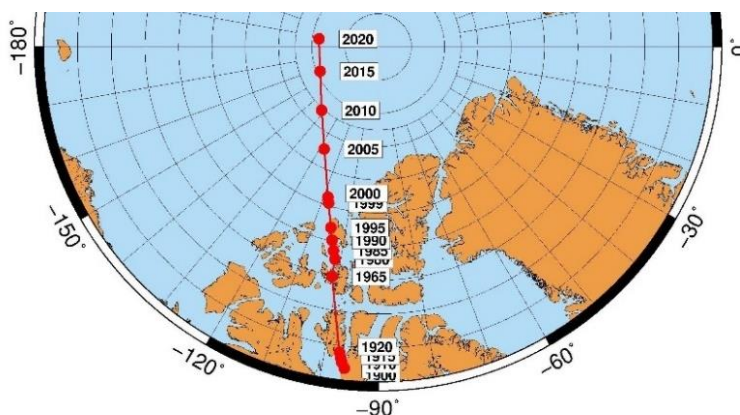
$$|\mathbf{B}| = \frac{|\mathbf{F}|}{|q||\mathbf{v}|} = \frac{1N}{1C \cdot 1 \frac{m}{s}} = \frac{1N}{A \cdot m} = 1T \quad (2.8)$$

### 2.1.3 Campo magnético terrestre

El campo magnético de la Tierra es producido por los movimientos de las aleaciones de hierro fundido en el núcleo externo del planeta y es conocido como geo-dinamo. Su comportamiento es similar a un dipolo magnético excepto por su desplazamiento en el tiempo. La magnitud en la superficie de la Tierra varía entre los  $25 \mu\text{T}$  a  $65 \mu\text{T}$ . Debido al movimiento del planeta y del núcleo, la inclinación del campo es variable como se ha podido constatar a través de estudios en rocas, que presentan alineaciones de materiales ferrosos hacia direcciones distintas a donde se encuentra el norte magnético hoy.

Gracias a los registros que se tienen del desplazamiento del norte magnético del planeta ha sido posible generar el modelo Campo de Referencia Geomagnético Internacional (IGRF, en inglés *International Geomagnetic Reference Field*), que predice su ubicación en función del tiempo, el modelo ha sido constatado y hasta el momento ha funcionado correctamente. El desplazamiento es lento, ha pasado de estar en el norte de Canadá inicios del siglo XX para encontrarse a escasos 100 kilómetros del norte geográfico de la Tierra hoy como se observa en Ilustración 4.

Ilustración 4. Desplazamiento del norte magnético con modelo CRGI



Fuente: [24]

También como referencia sobre el campo magnético de la Tierra se cuenta con el Modelo Magnético Mundial (WMM, en inglés *World Magnetic Model*). Este modelo es financiado en conjunto entre la Agencia Nacional de Inteligencia Geoespacial de los Estados Unidos (US NGA) y el Centro Geográfico de Defensa de Reino Unido (U.K. DGC), y administrado por el Centro de Datos Geofísicos de la Administración Nacional Atmosférica y Oceánica (US NOAA/NGDC) y el Servicio Geológico Británico (U.K. BGS). Estas organizaciones recolectan y procesan la información satelital del campo magnético de la Tierra y la emplean de referencia tanto en investigaciones civiles como militares entre los miembros de la Organización del Tratado del Atlántico Norte (OTAN) [25].





## 2.2 MAGNETÓMETROS

Un magnetómetro es un instrumento de medida que determina la magnitud de un campo magnético en un punto, según el tipo de sensor también puede estimar la orientación de este. Existen diferentes fenómenos físicos que permiten medir el campo como el efecto Hall [27] o el torque del momento magnético de un átomo [28], independiente de esto, siempre soportado en el principio de la fuerza de Lorentz (ver pág. 9)

### 2.2.1 Técnicas de magnetometría

La selección de una técnica u otra depende de las necesidades de medición que se tengan, algunos métodos resultan demasiado complejos y son útiles únicamente en entornos controlados de laboratorio. Las siguientes son las técnicas más conocidas:

#### 2.2.1.1 Fuerza de Faraday

Aplica una variante de la fuerza de Lorentz que indica que “*un gradiente de campo magnético espacial  $\nabla$  produce una fuerza  $\mathbf{F}$  que actúa sobre un objeto magnetizado  $M$* ” [29], esta afirmación se describe en la ecuación (2.9). Ese es el principio de funcionamiento de una brújula.

$$F = (M \cdot \nabla) \quad (2.9)$$

En un laboratorio se emplea un voladizo (*cantilever*) o carga suspendida que esté magnetizada, luego se mide el desplazamiento y se calcula la fuerza necesaria para que ese desplazamiento sea posible. Con el campo magnético de referencia y la fuerza ejercida por el campo a medir ya conocidos se despeja el campo a medir. Esta técnica resulta menos sensible y precisa por varias razones: se necesita generar un campo magnético conocido y para ello es requerida una referencia de calibración. La presencia de un torque en el campo afecta el modelo de medición y requiere estrategias para compensarlo. Es necesario generar un campo de referencia lo suficientemente grande como para que interactúe con el campo que se desea medir. Las ventajas que tiene son una gran tolerancia al ruido y facilidad en la implementación [29].

#### 2.2.1.2 Inducción

Se emplea una bobina de un material conductor y una muestra que se somete a magnetización cuando se aplica una corriente alterna controlada en la bobina, la reacción entre el campo generado por la bobina y la muestra permite establecer un diferencial de corriente en la bobina, lo que facilita la medición del campo aplicando las ecuaciones de Maxwell para calcularlo.

#### 2.2.1.3 SQUID

Su nombre proviene de las siglas en inglés Dispositivo de Interferencia Cuántica por Superconducción, es una aplicación del efecto Josephson, que es un fenómeno de súper corriente en una juntura de materiales superconductores sin aplicar voltaje [30]. Se emplean dos juntas Josephson en paralelo formando un anillo que crea un campo que interactúa con el que

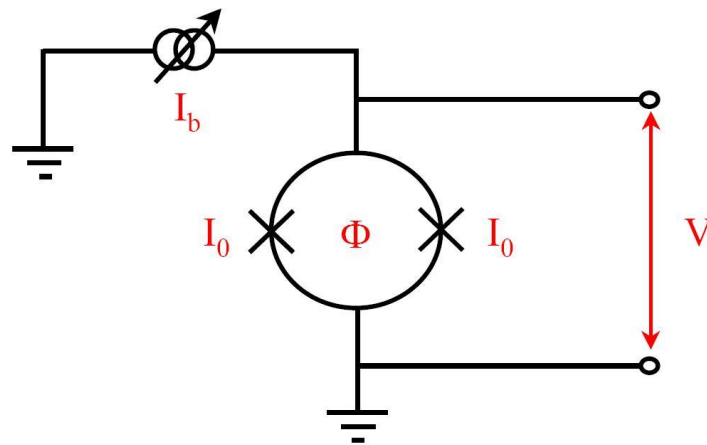
---

se desea medir, como se observa en Ilustración 6. Al ubicar un material resistivo seguido del anillo SQUID se obtiene un voltaje calculado con la ley de Ohm, para este caso:

$$\Delta V = \frac{R}{L} \cdot \Delta\Phi \quad (2.10)$$

Donde  $R$  es la resistencia de los materiales y  $L$  es la autoinductancia del anillo, que no suele ser muy elevada.  $\Delta\Phi$  es el resultado de la variación de flujo magnético producto de la interacción entre el campo generado por el anillo y el externo. Para medir este efecto se utiliza la ley de Ohm que determina la diferencia de potencial  $\Delta V$ , este será el indicador de diferencial de campo magnético del sensor [31].

Ilustración 6. Esquema SQUID



Fuente: [32]

Esta técnica de medición es muy útil por su sensibilidad que está en el orden de los atto Teslas (aT), no necesita refrigeración criogénica para funcionar, aunque su tamaño es de unos cuantos centímetros.

#### 2.2.1.4 Torque

En cuanto a esta técnica resulta distinta a las demás porque no mide directamente el campo magnético, para este caso se evalúa el diferencial en el torque del momento magnético de una muestra [33]. Este cambio se relaciona en la ecuación (2.11):

$$\tau = \mu \times B \quad (2.11)$$

En la ecuación el torque  $\tau$  cambia en función del campo magnético  $B$  que depende de un momento magnético constante  $\mu$ . Este torque es una manera de medir el magnetismo o la anisotropía del objeto de muestra. Para medir con este efecto, una forma común consiste en montar un voladizo y un objeto fijo cercano al que se mide su desplazamiento a través del cambio de capacitancia entre ambos. Otra manera de medir el cambio es a través de la piezorresistividad o por interferometría óptica.

## 2.2.2 Capacidades y características de un magnetómetro

Como se mostró en la sección anterior hay distintas maneras de construir un dispositivo que permita realizar una medición de campo magnético, por ello es importante tener en cuenta las características que dispone el sensor dependiendo de la aplicación. Los criterios presentados en esta sección son tomados de [28].

### 2.2.2.1 Ruido

Es una variación aleatoria en la medición producto de fuentes distintas al campo que se está midiendo. El ruido es medido en relación con el campo en Teslas por raíz cuadrada de Hercio, el componente de frecuencia se debe al ancho de banda y la cantidad de energía de ruido presente, calculado por raíz media cuadrada (RMS, en inglés *Root Mean Square*). La escala usual responde a nano o pico teslas dependiendo de la calidad del sensor.

### 2.2.2.2 Resolución

La resolución hace referencia a la mínima variación detectable por el sensor. Este valor también indica la diferencia entre un nivel y otro detectable en la señal, los valores intermedios por tanto no son medibles y se redondean, dando origen a lo que se conoce como error de cuantización. Este término aplica solamente a sensores digitales.

### 2.2.2.3 Sensibilidad

Esta palabra suele dar problemas por la ambigüedad entre los factores ancho de banda detectable y rango de campo medible. También algunos fabricantes emplean el vocablo haciendo referencia a la “sensibilidad de ruido” o incluso a la resolución. Lo conveniente es emplear sensibilidad para describir la mínima variación detectable independiente de ruido blanco y térmico sin considerar el muestreo digital.

### 2.2.2.4 Tasa de muestreo

Cuando se emplea un sensor digital el muestreo es crítico junto con la resolución, estos términos establecen la calidad del cuantificador digital empleado. Una tasa de muestreo más alta garantiza una mayor disponibilidad de información por unidad de tiempo, esto es útil en procesos de tratamiento de señal posteriores. Respecto al ciclo de lectura tiene que ver el tiempo que le toma al sensor responder a un cambio del campo magnético medido. Estas dos referencias son útiles para determinar la respuesta a cambios de posición rápidos, que es una característica crucial en aplicaciones de control o navegación.

### 2.2.2.5 Ancho de banda

Este parámetro guarda relación con el ciclo de lectura, tiene que ver con la capacidad de medir campos oscilatorios o en movimiento. Este parámetro también puede ser empleado como referencia para determinar la velocidad de respuesta del sensor al movimiento, pero también de su tolerancia a fuentes de campo magnético pulsantes como motores o antenas de radio, debido a la capacidad que tienen ciertos magnetómetros de ser empleados como detectores de espectro electromagnético.

### 2.2.2.6 Error absoluto y corrimiento

El error absoluto responde a la diferencia que existe entre el promedio de las mediciones y el promedio real del campo, para establecer este valor se necesita un patrón. El corrimiento es el cambio del error en el tiempo, suele ser más pequeño que el error absoluto y su origen es variado dependiendo de la técnica y materiales del sensor.

### 2.2.2.7 Tolerancia de gradiente

La forma del gradiente de un campo depende de su tamaño, un dipolo magnético tiene una alta concentración de líneas de campo en sus extremos y se extienden en el espacio, si el dipolo es muy pequeño como por ejemplo un imán, es posible que el magnetómetro evalúe muchas líneas de campo con distintas magnitudes y direcciones que, al ser cuantificadas pueden resultar en un valor errado. En el caso de medir grandes campos, como el de la Tierra, esos errores se hacen ínfimos. Un sensor será más tolerante al gradiente en cuanto tenga un tamaño reducido.

### 2.2.2.8 Zonas muertas

Ocurre cuando la topología del sensor no le permite alinearse con una línea de campo y por tanto no mide nada a pesar de estar dentro de un campo magnético.

### 2.2.2.9 Error de orientación

Este error corresponde a la diferencia entre la orientación real del campo magnético y el medido por el sensor, el nivel de error depende de la calidad en la construcción del sensor.

### 2.2.2.10 Rango de medición

Consiste en los límites máximos y mínimos medibles por el magnetómetro, tanto por las capacidades del sensor como por los límites del cuantificador. Por ejemplo, en los magnetómetros de navegación que miden el campo de la tierra, manejan un rango entre 20  $\mu\text{T}$  y 80  $\mu\text{T}$ .

## 2.2.3 Tipos de magnetómetros

Con las técnicas de magnetometría anteriormente expuestas es posible encontrar una gran cantidad de tipos de magnetómetros, sin embargo, para esta investigación es de interés aquellos que sean electrónicos y de un tamaño reducido. Los que cumplen con esta condición son los siguientes tipos de magnetómetros:

### 2.2.3.1 De Efecto Hall

El Efecto Hall fue descubierto por Edwin Hall en 1879, aplica la fuerza de Lorentz como base, para ello una corriente controlada es expuesta al campo magnético, la fuerza experimentada por los electrones los desvía de su dirección inicial provocando un voltaje. Esta señal de voltaje se describe en la ecuación (2.12).

$$V_H = \frac{R_H}{t} \cdot I \cdot B \cos(\alpha) \quad (2.12)$$

Donde  $R_H$  es el coeficiente Hall,  $I$  es la corriente pasando por la banda conductora del sensor,  $t$  es el grosor la banda,  $B$  es la magnitud del campo en medición y  $\alpha$  la diferencia entre la dirección del campo con el plano ortogonal de la banda conductora [27].

La sensibilidad de este tipo de sensores depende de la movilidad de electrones, es decir la resistividad de la banda y la corriente que pasa a través de esta. Estos sensores tienen una limitación en frecuencia en torno a 1 MHz y con una sensibilidad en escala de  $\mu\text{G}$  [27].

### 2.2.3.2 Magnetorresistencia anisotrópica (AMR)

La magnetorresistencia anisotrópica (AMR, en inglés *Anisotropic Magnetoresistance*) es un sensor construido con una película de níquel-hierro, la cual es atravesada por una corriente constante. Cuando se somete a un campo magnético la resistividad de la película cambia. En un sensor de este tipo se construye un arreglo de cuatro sensores como un puente de Wheatstone, la variación de voltaje indica la magnitud del campo magnético medido [34]. Este tipo de sensores es descrito en la sección Magnetómetro Anisotrópico Magnetorresistivo (ver pág. 23).

### 2.2.3.3 Magnetorresistencia gigante (GMR)

La magnetorresistencia gigante (GMR, en inglés *Gigant Magnetoresistance*) aplica el principio de dispersión dependiente del espín, consiste en un magnetorresistor de varias capas. Dos capas son ferromagnéticas, una capa no magnética y una anti ferromagnética que orientan los espines de las capas superiores, todas estas capas son atravesadas por una corriente constante paralela a las capas. Cuando se somete a un campo magnético, los espines de las capas ferromagnéticas comienzan a sufrir alteraciones en su orientación, haciendo que la conductividad cambie.

La sensibilidad de un sensor de este tipo depende del grosor de las capas. Estos sensores ofrecen alta sensibilidad y estabilidad térmica. Un campo magnético fuerte puede destruir las capas del sensor fácilmente [35].

### 2.2.3.4 Magnetorresistencia de túnel (TMR)

La magnetorresistencia de túnel (TMR, en inglés *Tunnel Magnetoresistance*) es un tipo de sensor con una construcción similar a GMR, con dos diferencias: la corriente atraviesa de forma transversal capa por capa, y la capa intermedia consiste en un material aislante de óxido metálico conocida como barrera. La resistencia del TMR cambia de manera similar a GMR por efectos de cambios en el espín. En ausencia de campo magnético las capas sufren magnetización antiparalela y en consecuencia una resistividad alta, cuando se somete a un campo se produce un efecto túnel alto y por tanto reduce la resistividad [36].

### 2.2.3.5 De efecto magneto-eléctrico (EM)

El efecto magneto-eléctrico (EM, en inglés *electric-magnetic effect*) trata de la aparición de una polarización eléctrica cuando hay presencia de un campo magnético. Este sensor se caracteriza por tener una capa de un material piezoeléctrico cubierto por capas de material magnetostrictivo, cuando se expone a un campo magnético alterno provoca excitación en las capas magnetostrictivas, esto en consecuencia genera estrés en el material piezoeléctrico que reacciona variando el voltaje que se aplica en el intermedio entre esta capa y las magnetostrictivas. En el caso de un campo estático, este genera un cambio de tensión constante medido en el material piezoeléctrico [37].

---

### 2.2.3.6 Comparativa

Cada tipo de sensor tiene ventajas y desventajas asociadas a su sensibilidad, resolución y tolerancia a ruido, pero desde la ingeniería la relación costo-beneficio importa para determinar la viabilidad del uso de un tipo de sensor en una aplicación. La Tabla 2 establece una comparativa y resume dichas características.

Tabla 2. Comparativa de sensores magnéticos

Tipo	Sensibilidad	Tamaño	Consumo energético	Ventaja	Desventaja	Costo
<b>Hall</b>	Medio/Bajo	$\mu\text{m}$	medio	Lineal	Baja sensibilidad	Muy bajo
<b>AMR</b>	Media	$\mu\text{m}$ -cm	medio	Estabilidad en corrimiento y sensibilidad	Rango de medición limitado	Bajo
<b>GMR</b>	Alta	$\mu\text{m}$	bajo	Pequeño tamaño	Ruido de baja frecuencia/histéresis	Medio
<b>TMR</b>	Alta	$\mu\text{m}$	bajo	Rango de magnetorresistividad alto	Histéresis	Medio
<b>EM</b>	Muy alta	mm-cm	Muy bajo	Pequeño tamaño/rango amplio	Pobre precisión para medir campos débiles	Muy bajo

Fuente: [27]

## 2.3 TRABAJOS RELACIONADOS

Como parte de la Metodología de trabajo (ver pág. 3) descrita anteriormente, el *Modelo de Investigación Documental* señala la importancia de recolectar y catalogar los trabajos de investigación que guardan relación con el presente documento, con el fin de establecer una base de conocimiento útil para mejorar la hipótesis de trabajo y determinar la pertinencia del mismo. En esta sección se exponen los recursos metódicos empleados y los resultados obtenidos de la investigación documental.

### 2.3.1 Fuentes de información

La existencia de bases de datos científicas ha facilitado la documentación de trabajos, gracias a la recolección de catálogos amplios y una clasificación en términos o palabras clave que ofrecen resultados más precisos en las búsquedas que se realizan. Estas son las bases de datos consultadas para esta investigación:

**IEEE Xplore:** Ofrece acceso a contenido científico y técnico publicado por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), lo que representa más cuatro millones de artículos. Dentro de Xplore se encuentra el contenido de 195 revistas, 1.400 actas de conferencias, 5.100 estándares técnicos y 2.000 libros. Mensualmente se registran alrededor 20.000 nuevos artículos en la base de datos [38].

**Google:** El motor de búsqueda más popular en la web también es un recurso útil, aunque no tiene un repositorio de artículos como tal, Google accede los índices de bases de datos científicas, recolecta citas y referencias que sirven como indicador sobre autores y trabajos relacionados con la búsqueda. Sin embargo, esta herramienta debe usarse con cautela porque también ofrecerá resultados de publicaciones de baja calidad, para compensar esto cuenta con la herramienta Scholar diseñada para filtrar el contenido científico y académico de fuentes reconocidas [39].

**ScienceDirect:** Hace parte de la editorial neerlandesa Elsevier, la mayor editorial de libros de medicina y literatura científica del mundo. ScienceDirect opera como plataforma digital para acceder a los contenidos que ofrece Elsevier. Cuenta con quince millones de publicaciones obtenidas de 3.800 revistas y más de 37.000 libros, de estos doscientos cincuenta mil son de libre acceso [40]. Dentro de la plataforma hay contenidos sobre ciencias físicas, de la vida, sociales y de salud todos revisados por pares científicos.

**SpringerLink:** Plataforma web que hace parte de la editorial Springer Science+Business Media. Ofrece contenidos de revistas, libros, series, protocolos y trabajos de referencia verificados por pares científicos. Se encuentran en línea más de doce millones publicaciones que engloban distintas ciencias al igual que ScienceDirect [41].

### 2.3.2 Criterios de búsqueda

Las fuentes de información seleccionadas cuentan con clasificadores que emplean palabras clave para ejecutar las búsquedas y ofrecer resultados, por esta razón es importante una selección de palabras relevantes y sus combinaciones que guarden relación con esta investigación. Para ello se recurre al título y la pregunta de investigación.

*Estimación de los parámetros estáticos de corrección de la señal entregada por un magnetómetro electrónico de tres ejes.*

*¿Cuáles son las características de un sistema para la estimación de los parámetros estáticos de corrección de la señal entregada por un magnetómetro electrónico de tres ejes?*

De estas dos frases se extractan: Magnetómetro, Parámetros, Corrección. De la palabra *tres ejes* se obtiene el concepto asociado de Sistema Inercial por su aplicación en sistemas de navegación. La palabra corrección también se puede asociar con Calibración, aunque en metrología estos términos distan por método y nivel de precisión, pero para evitar dualidades por la traducción se elige la palabra Calibración. Con estas referencias se han elegido las siguientes combinaciones, ordenadas en Tabla 3.

Con las palabras clave definidas para hacer la búsqueda, se realizan las consultas en las bases de datos, filtrando los resultados en un periodo entre 2012 y 2018. En el motor de Google se han suprimido las búsquedas de citas para evitar la redundancia de artículos. Los resultados de búsqueda se resumen en la Tabla 4.



Tabla 3. Palabras clave de búsqueda

Combinación	Criterio
a	Magnetometer
b	Magnetometer + Calibration
c	Magnetometer + Parameters
d	Magnetometer + Calibration + Parameters
e	Magnetometer + Inertial + Calibration
f	Magnetometer + Inertial + Parameters
g	Magnetometer + Inertial + Calibration + Parameters

Tabla 4. Resultados de búsquedas según criterios

Combinación	Xplore	Google	ScienceDirect	SpringerLink	Total	%
a	2.452	34.100	22.467	12.294	71.313	35,13
b	234	16.700	2.737	1.773	21.444	10,56
c	281	23.300	17.134	8.541	49.256	24,26
d	53	17.100	2.200	1.439	20.792	10,24
e	83	10.300	639	595	11.617	5,72
f	55	14.800	1.986	1.112	17.953	8,84
g	16	9.580	547	493	10.636	5,24
<b>Total</b>					<b>203.011</b>	<b>100</b>

El objetivo de este ejercicio está en obtener un indicador de pertinencia de los términos empleados en la búsqueda, determinar la disponibilidad de información y la importancia que podría tener la investigación, a menos resultados es probable que el tema no haya sido tratado.

Analizando los resultados, se observa que la palabra clave “Parameters” no tiene tanta influencia en la discreción de resultados como sucede con “Calibration”, de igual manera “Inertial” reduce la cantidad de resultados de forma significativa. Esto indica que los procesos de calibración para sistemas inerciales combinados con magnetómetros guardan una relación que no se ha explorado tanto, esto es un factor que puede resultar interesante en términos de aplicación de esta investigación, incluso puede orientar sobre opciones de desarrollo y evaluación que hace parte de los objetivos del presente trabajo.

En cuanto a las bases datos, Google guarda los mismos niveles de tendencia comparadas con las demás bases de datos, pero sus resultados son voluminosos y al evaluar la calidad del contenido se aprecia en sus resultados la inclusión de menciones en publicaciones web que no tienen algún tipo de indexación y que no han sido revisados por pares científicos, por esta razón se excluye a esta plataforma como fuente confiable de trabajos.

Por tanto, de esta búsqueda se obtienen 1.056 publicaciones que posiblemente resulten útiles, luego se revisan los títulos completos y resúmenes de los artículos que guardan mayor correspondencia con la investigación y se elige un grupo más pequeño que constituirán la base documental de este trabajo.

### 2.3.3 Trabajos seleccionados

Con la metodología aplicada en la sección anterior se han seleccionado catorce artículos y un trabajo de maestría que ofrecen información útil.

#### 2.3.3.1 Calibración integrada y compensación de perturbación magnética de un magnetómetro de tres ejes (*Integrated calibration and magnetic disturbance compensation of three-axis magnetometers*)

Aporte: Aplica una estrategia de calibración empleando perturbaciones *soft-iron* y *hard-iron* en un modelo matemático no lineal, compara el proceso de muestreo estático y dinámico [42].

Año: 2016

Origen: Equipment Academy, China; National University of Defense Technology, China.

#### 2.3.3.2 Aproximación pseudo-lineal de estimación de actitud y calibración de magnetómetro de un nanosatélite (*Pseudo-Linear Approach to Nanosatellite Attitude Estimation and Magnetometer Calibration*)

Aporte: Expone el modelo de un magnetómetro de tres ejes y determina el modelo apropiado para aplicar un filtro Kalman pseudo-lineal para emplearlo en el nanosatélite Aalto-1 [43].

Año: 2017

Origen: Universidad Aalto, Finlandia; Agencia Japonesa de Exploración Aeroespacial (JAXA).

#### 2.3.3.3 Calibración consistente de magnetómetro para determinación no lineal de actitud (*Consistent calibration of magnetometers for nonlinear attitude determination*)

Aporte: Muestra el modelo de tres dimensiones de un magnetómetro, expone el modelo de error y un proceso de calibración aplicando un algoritmo de aproximación a elipsoide y muestra un proceso experimental de validación [44].

Año: 2014

Origen: Universidad de Tecnología Sharif, Irán.

#### 2.3.3.4 Mejoras en modelamiento de error determinístico y calibración de sensores inerciales y magnetómetro (*Improvements in deterministic error modeling and calibration of inertial sensors and magnetometers*)

Aporte: Prueba un método con un grupo de sensores de tipo industrial, emplea en su proceso acelerómetro, giroscopio y magnetómetro. Muestra el proceso de determinación del modelo, extracción de parámetros y evaluación. El proceso aplica el algoritmo Levenberg-Marquardt [45].

Año: 2016

Origen: Universidad de Bilkent, Turquía.

#### 2.3.3.5 Calibración dinámica de magnetómetro y alineación con sensores inerciales con filtros Kalman (*Dynamic Magnetometer Calibration and Alignment to Inertial Sensors by Kalman Filtering*)

Aporte: Modelamiento y estimación de parámetros con un filtro Kalman extendido aplicando una correlación entre acelerómetro, magnetómetro y giroscopio para determinar la actitud de un sistema [46].

Año: 2018

Origen: Universidad Shanghai Jiao Tong, China.

**2.3.3.6 Calibración de magnetómetro asistido por giroscopio con filtros Kalman extendidos para dispositivos de electrónica de consumo** (*Extended Kalman Filter-Based Gyroscope-Aided Magnetometer Calibration for Consumer Electronic Devices*)

Aporte: Evaluación de filtros Kalman extendidos para ajustar un elipsoide de magnetómetro apoyado con un magnetómetro, expone el modelo y el proceso experimental de evaluación usando hardware comercial empleado en dispositivos como smartphones [47].

Año: 2017

Origen: Intel Asia Pacific Research and Development., Ltd, China.

**2.3.3.7 Calibración de magnetómetro usando sensores inerciales** (*Magnetometer Calibration Using Inertial Sensors*)

Aporte: Expone un algoritmo de calibración obteniendo parámetros y luego evaluando un filtro Kalman extendido para realizar una aproximación a elipsoide de un magnetómetro teniendo como referencia espacial los datos obtenidos de un acelerómetro y un giroscopio de un smartphone [48].

Año: 2016

Origen: Universidad de Linköping, Suecia; Universidad de Uppsala, Suecia.

**2.3.3.8 Calibración en vuelo de magnetómetro y sistema de navegación inercial para satélites de baja altitud** (*In-flight Calibration of Magnetometers and Low-cost SINS for Small Land-survey Satellite*)

Aporte: Estimación de parámetros de magnetómetros en un sistema de bajo costo para la navegación de un satélite [49].

Año: 2015

Origen: Universidad Técnica del estado de Samara, Rusia; Universidad Técnica de Estambul, Turquía.

**2.3.3.9 Calibración de magnetómetro usando la matriz de covarianza de un filtro Kalman para estimación en línea de orientación de campo magnético** (*Magnetometer Calibration Using Kalman Filter Covariance Matrix for Online Estimation of Magnetic Field Orientation*)

Aporte: Método de calibración con sensor estático, emplean un grupo de bobinas de Helmholtz para generar campos calibrados de evaluación. La calibración se realizar con filtros Kalman “unscented” [50].

Año: 2014

Origen: Universidad de Ljubljana, Eslovenia.

**2.3.3.10 Calibración automática y adaptative de sensores de campo tridimensionales** (*Automatic and adaptative calibration of 3D field sensors*)

Aporte: Artículo empleado como referencia en varios artículos, expone un modelo sencillo para evaluación de un magnetómetro de tres ejes para ajuste de elipsoide [51].

Año: 2008

Origen: Nokia Research Center, Finlandia.

---

**2.3.3.11 Estrategia y prueba general de calibración de magnetómetros de tres ejes**  
(*Calibration strategy and generality test of three-axis magnetometers*)

Aporte: Explica un diseño de experimento para calibrar y evaluar un magnetómetro, compara posteriormente la calidad del procedimiento empleando distintos algoritmos [52].

Año: 2013

Origen: National University of Defense Technology, China.

**2.3.3.12 Calibración rápida de una unidad de medición inercial de 9 grados de libertad usando un trazador de posición de 3 grados de libertad y una secuencia de movimiento semi-aleatoria**  
(*Fast calibration of a 9-DOF IMU using a 3-DOF position tracker and a semi-random motion sequence*)

Aporte: Presenta un método de calibración manual con un dispositivo inalámbrico de pequeño tamaño, describe el proceso de experimentación y evaluación aplicados en la investigación [53].

Año: 2016

Origen: University College Dublin, Irlanda.

**2.3.3.13 Autocalibración de magnetómetro aprovechando medidas de posición restringidas**  
(*Magnetometer Autocalibration Leveraging Measurements Locus Constraints*)

Aporte: Expone un modelo matemático detallado, es un trabajo empleado como referencia por otros artículos de esta lista [54].

Año: 2007

Origen: Universidad de Minnesota, Estados Unidos.

**2.3.3.14 Solución explícita de calibración de magnetómetro**  
(*Explicit solution for magnetometer calibration*)

Aporte: Es un artículo producto de la revisión del estado del arte para establecer un modelo matemático de ajuste a un elipsoide para calibrar un magnetómetro de tres ejes [55].

Año: 2014

Origen: Instituto Tecnológico de Aeronáutica, Brasil.

**2.3.3.15 Magnetómetro de magnetorresistencia anisotrópica para sistemas de medición inercial**  
(*Anisotropic Magnetoresistance Magnetometer for Inertial Navigation Systems*)

Aporte: Un estado del arte detallado sobre la estructura de un magnetómetro AMR con información sobre su modelo matemático [27].

Año: 2013

Origen: Politécnic Ecole, Francia.

### 3 DISEÑO DEL EXPERIMENTO

---

Con el Estado del Arte revisado se tiene una base de conocimientos suficiente para plantear el procedimiento que se aplicará en esta investigación. En este capítulo se tratarán los aspectos técnicos previos considerados el diseño del experimento aplicado como el hardware, el modelo matemático seleccionado para el sensor, método de experimentación y tratamiento de datos.

#### 3.1 CONSIDERACIONES SOBRE EL MAGNETÓMETRO

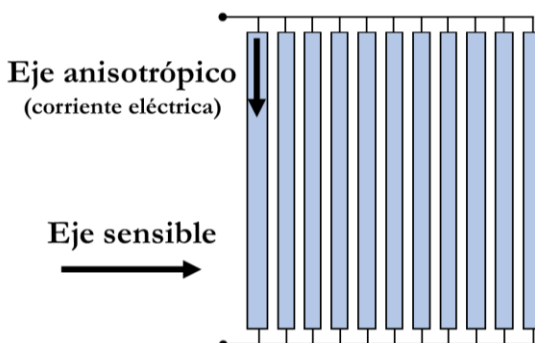
El eje de este trabajo radica en el magnetómetro que se elija, considerando la información mostrada en la Tabla 2 (ver pág. 17), se ha seleccionado el magnetómetro de tres ejes Honeywell HMC5883L por su disponibilidad comercial, costo y tipo; es un sensor AMR que según lo expresado en la tabla ya mencionada es una solución equilibrada. Teniendo eso presente, esta selección se profundizará en las características de este sensor que serán la base del experimento.

##### 3.1.1 Magnetómetro Anisotrópico Magnetorresistivo

En 1857 fue publicado el artículo “*Effect of magnetization on the electric conductivity of nickel and iron*” realizado por William Thomson [56]. En esta publicación se hacía el anuncio del descubrimiento de un efecto de cambios en la resistencia en función de la intensidad y dirección de campo magnético sobre un material de hierro y níquel, a esto se le llamó magnetorresistencia anisotrópica (AMR). Esta tecnología no fue práctica en su momento por el tamaño del sensor y la dificultad que existía para medir campos magnéticos con precisión, tuvo que pasar más de un siglo hasta que fuera posible construir una película fina lo suficientemente pequeña para emplearla en la construcción de sensores, a esta se le conoce comercialmente como Permalloy.

El efecto AMR se produce cuando un campo magnético altera la conductividad de un material al alinear el dominio magnético de este, normalmente el dominio se encuentra en un estado aleatorio entre cada átomo del material. Cuando se somete a un campo estos dominios comienzan a orientarse en la misma dirección del campo y cuantos más átomos alineados, menor será la resistencia eléctrica del material.

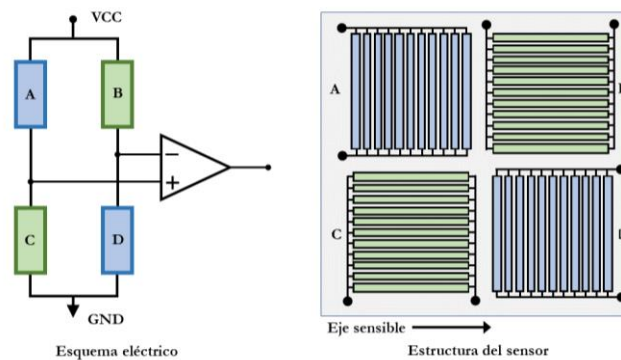
Ilustración 7. Esquema de una magnetorresistencia anisotrópica



Internamente una magnetorresistencia (MR) está construida a partir de líneas paralelas de forma similar a lo observado en la Ilustración 7, lo cual permite acomodar más material sensible en un área reducida. El eje sensible indica la dirección en la que el sensor tiene una mayor capacidad de detección de campo magnético, las líneas de la magnetorresistencia son perpendiculares al eje sensible y paralelas al llamado eje anisotrópico, que es la misma dirección que lleva la corriente a través del magnetorresistor.

Un AMR tiene una variabilidad en su resistencia entorno al 10% en presencia de un campo magnético, esto obliga al uso de una instrumentación sensible para medir las variaciones más pequeñas. Para solucionar este inconveniente, los sensores AMR comerciales emplean un arreglo de cuatro MR en configuración de puente de Wheatstone; que facilita la detección de cambios debido a la medición en dos MR a la vez. Además, la variación de voltaje se mide con un amplificador de instrumentación, llevando las diferencias de voltaje a niveles más altos, lo que facilita la digitalización de la medición con un conversor analógico-digital (ADC). El esquema básico de un sensor AMR se muestra en la Ilustración 8.

Ilustración 8. Esquema básico de un sensor AMR de puente Wheatstone de un eje

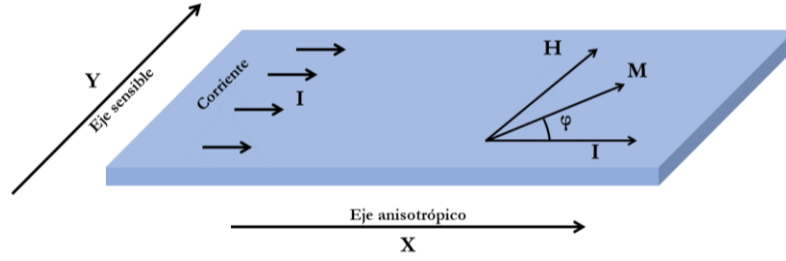


El funcionamiento del esquema anterior es el siguiente: el eje sensible es producto de las dos magnetorresistencias que tienen sus ejes anisotrópicos perpendiculares al sensible, en este caso A y D. Cuando se expone el sensor a un campo magnético paralelo al eje sensible la resistividad en A y D disminuirán, esto provocará un aumento de voltaje en el nodo A-C y una reducción de tensión en el nodo B-D. En la salida del amplificador de instrumentación la señal de voltaje presentará un incremento proporcional al cambio de los dos nodos. Para evitar que B y C constituyan un segundo eje sensible, el amplificador de instrumentación debe estar ajustado para ignorar cambios en estas dos magnetorresistencias o un procesamiento posterior si el sensor tiene salidas analógicas.

La razón de porqué el eje sensible es perpendicular, tiene relación con la regla de la mano derecha aplicada en electromagnetismo para determinar la orientación de las líneas de campo producidas por una corriente. Por esta razón el magnetorresistor debe tener una corriente para realizar mediciones, pues la interacción del campo de autoinducción y el campo a medir produce las interacciones en el material que generan cambios en la resistividad, en la Ilustración 9 se muestra una sección de un AMR.

---

Ilustración 9. Componentes de interacción en un AMR.



De la imagen anterior, el vector  $\mathbf{H}$  responde a la dirección del campo magnético a medir, el vector  $\mathbf{M}$  es el vector de magnetización producto de la corriente  $\mathbf{I}$  que fluye en el material. La corriente que es paralela al eje anisotrópico,  $\varphi$  indica el ángulo de diferencia entre  $\mathbf{I}$  y  $\mathbf{M}$  que es producto de la deformación del vector  $\mathbf{M}$  por la fuerza magnética con el campo  $\mathbf{H}$ . La energía producto de las interacciones electromagnéticas en el MR está dada por la siguiente ecuación:

$$E = E_{mag} + E_k = -\vec{H} \cdot \vec{M} + K_u \sin^2 \varphi \quad (3.1)$$

Donde  $\mathbf{E}_{mag}$  es la energía magneto-estática que depende de las fuerzas entre los campos  $\mathbf{M}$  y  $\mathbf{H}$ .  $\mathbf{E}_k$  la energía anisotrópica resultado de una constante anisotrópica, propia del material, es proporcional con el ángulo  $\varphi$  y  $K_u$  es una constante del material. Esta ecuación se expande aplicando las siguientes consideraciones:  $M_s = |\vec{M}|$  que es la saturación de magnetización,  $H_k = \frac{2K_u}{M_s}$  es el campo anisotrópico y los componentes de campo externo  $H_x$  y  $H_y$ . El resultado de la expansión es el siguiente:

$$E = -\mu_0 M_s (H_y \sin \varphi + H_x \cos \varphi) + \frac{\mu_0 M_s H_k}{2} \sin^2 \varphi \quad (3.2)$$

Derivando esta función en términos de  $\varphi$  se obtiene una ecuación instantánea de energía, para lograr la mínima energía se aplica la siguiente la condición  $\frac{\partial E}{\partial \varphi} = 0$ , luego la ecuación resultante es:

$$H_k \sin \varphi \cos \varphi = H_y \cos \varphi - H_x \sin \varphi \quad (3.3)$$

Si se asume que  $\varphi \ll 1$ , se aproximan las expresiones  $\sin \varphi \approx \varphi$  y  $\cos \varphi \approx 1$ .

Aplicando las aproximaciones en la ecuación (3.3) se obtiene la siguiente solución para  $\varphi$  (ecuación (3.4)):

$$\varphi = \frac{H_y}{H_k + H_x} \quad (3.4)$$

La ecuación (3.4) señala que si un campo externo es aplicado en la misma dirección del eje sensible ( $H_y$ ) el vector de magnetización  $\mathbf{M}$  rota. Durante esta rotación, la resistencia de la capa

permalloy cambia en función de dicho ángulo. La resistividad se determina aplicando la ley de Voigt-Thomson, esta ley describe el efecto de magnetorresistencia anisotrópica para determinar el cambio de la resistencia eléctrica del material en términos del ángulo de incidencia  $\varphi$ ; la ecuación de esta ley es la siguiente:

$$\rho(\varphi) = \rho_{\parallel} \cos^2 \varphi + \rho_{\perp} \sin^2 \varphi = \rho_{\parallel} - \Delta\rho \sin^2 \varphi \quad (3.5)$$

Donde  $\rho_{\parallel}$  y  $\rho_{\perp}$  son los componentes de resistividad paralelos y perpendicular respectivamente, la diferencia entre esos dos es  $\Delta\rho = \rho_{\parallel} - \rho_{\perp}$  y  $\varphi$  es el ángulo entre el vector de magnetización y la dirección de la corriente. Asumiendo que corriente tiene una dirección paralela al eje anisotrópico, tomando las ecuaciones (3.4) y (3.5), la variación de la resistencia será:

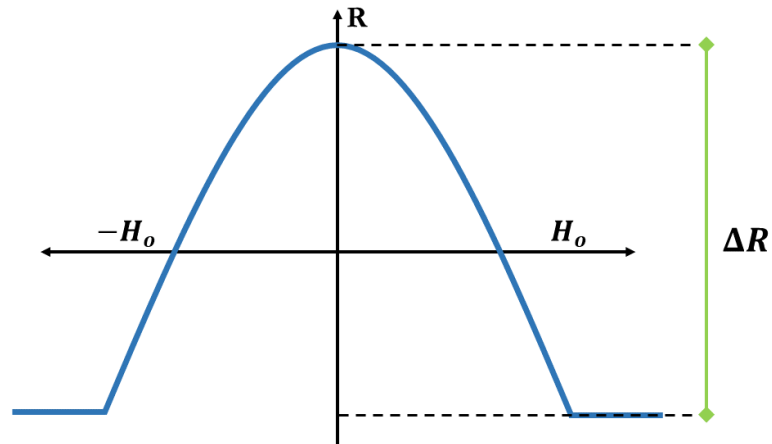
$$R_{\varphi} = R_{H=0} - \Delta R \sin^2 \varphi \quad (3.6)$$

De lo anterior  $R_{H=0}$  es la resistencia en un campo nulo paralelo al eje anisotrópico y  $\Delta R$  es la máxima variación de resistencia. Asumiendo ángulos  $\varphi \ll 1$ ,  $\sin^2 \varphi \approx \varphi^2$  y reemplazando la ecuación (3.4) en (3.6) se obtiene:

$$R_{\varphi} = R_{H=0} - \Delta R \left[ \frac{H_y}{H_k + H_x} \right]^2 \quad (3.7)$$

En la ecuación resultante del proceso anterior se encuentra un problema con el sensor AMR, debido a la respuesta exponencial que tiene respecto a un campo en el eje sensible  $H_y$ . En otras palabras, se tiene un comportamiento no lineal que es una condición no deseable, la respuesta del sensor está graficada en la Ilustración 10.

Ilustración 10. Respuesta de magnetorresistencia anisotrópica básica



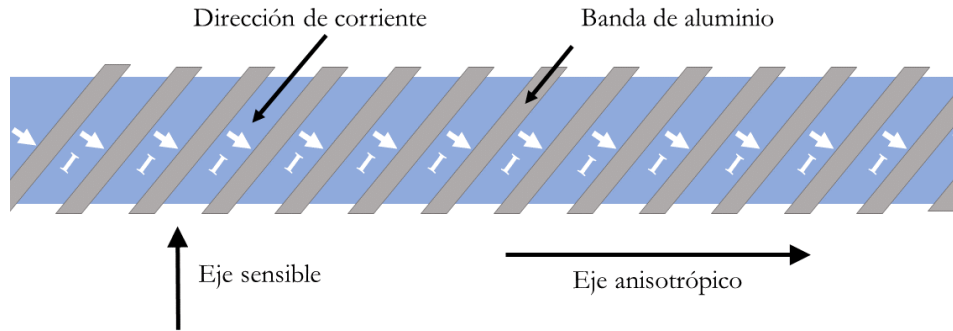
Para solucionar el problema de no linealidad, se modifica la dirección de la corriente sobre la lámina permalloy agregando un arreglo de bandas de aluminio con ángulo de  $45^\circ$ , se emplea

---



aluminio porque es un metal conductor diamagnético, por tanto, no afecta las interacciones de campo magnético entre el material AMR y el campo externo. La estructura se muestra en la siguiente imagen:

Ilustración 11. Estructura AMR de corriente a 45°



Con este cambio en la dirección de la corriente la ecuación (3.7) cambia de la siguiente forma:

$$R_{\varphi+45^\circ} = R_{H=0} - \Delta R \left[ \frac{1}{2} + \frac{H_y}{H_k + H_x} \sqrt{1 - \left( \frac{H_y}{H_k + H_x} \right)^2} \right] \quad (3.8)$$

De la ecuación anterior resulta en un componente  $K_1 = \frac{H_y}{H_k + H_x}$  que presenta un comportamiento lineal, el segundo componente  $K_2 = \sqrt{1 - \left( \frac{H_y}{H_k + H_x} \right)^2}$  resulta de tipo no lineal; ambos cambian con un campo magnético externo.

Al construir el sensor como un arreglo en puente, se obtienen dos magnetorresistencias que tienen el comportamiento de la ecuación (3.8) y otras dos que responderán a la siguiente ecuación:

$$R_{\varphi-45^\circ} = R_{H=0} - \Delta R \left[ \frac{1}{2} - \frac{H_y}{H_k + H_x} \sqrt{1 - \left( \frac{H_y}{H_k + H_x} \right)^2} \right] \quad (3.9)$$

Como se observa en la ecuación anterior, se mantienen los elementos  $K_1$  y  $K_2$ , con la diferencia del signo consecuente con el cambio de sentido en la posición del sensor respecto al campo externo a medir. Luego, con estas dos ecuaciones se define el voltaje de salida en el puente de Wheatstone formado por magnetorresistores anisotrópicos descrito por la siguiente ecuación:

$$V_{out} = V_{cc} \frac{R_{\varphi+45^\circ} - R_{\varphi-45^\circ}}{R_{\varphi+45^\circ} + R_{\varphi-45^\circ}} \quad (3.10)$$

Reemplazando las ecuaciones (3.8) y (3.9) en la expresión (3.10) se tiene:

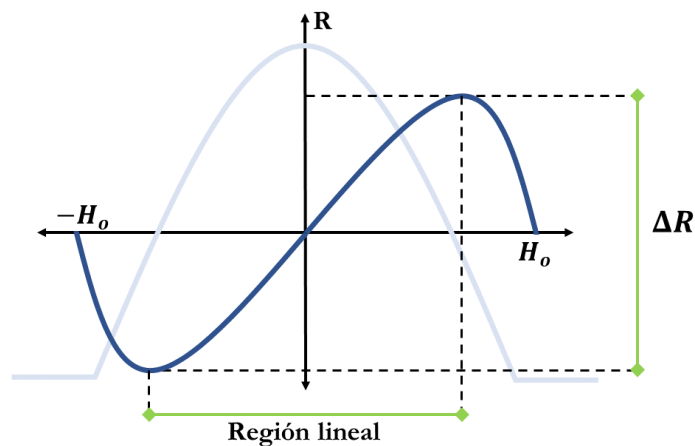
$$V_{out} = V_{cc} \frac{\Delta R}{R} \frac{H_y}{H_k + H_x} \sqrt{1 - \left(\frac{H_y}{H_k + H_x}\right)^2} \quad (3.11)$$

De la ecuación (3.11) se entiende que la razón por la cual se emplea un arreglo como puente es el aumento de la sensibilidad del sensor. Además, mediante la manipulación del factor  $H_k$  que es variable según la calidad y construcción del material, hace posible linealizar la respuesta de salida del puente AMR aplicando las condiciones  $H_k^2 \gg H_y^2$  y  $H_k^2 \gg H_x^2$ , resultando en la aproximación:

$$V_{out} \approx V_{cc} \frac{\Delta R}{R} \frac{H_y}{H_k + H_x} \quad (3.12)$$

La respuesta de esta aproximación lineal generada se muestra en la Ilustración 12.

Ilustración 12. Respuesta de un sensor AMR en puente



La respuesta del sensor con esta construcción muestra un comportamiento con una región que se aproxima a un comportamiento lineal, en consecuencia, el voltaje medido en el puente AMR será proporcional a la variación de campo magnético.

### 3.1.2 Efectos en un magnetómetro AMR

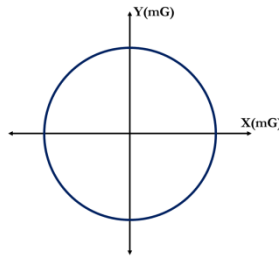
Los sensores no son perfectos y por tanto son susceptibles a presentar errores o alteraciones en sus mediciones, consecuencia de fenómenos físicos que en un entorno de aplicación no son controlables. En el caso de un magnetorresistor anisotrópico estos son los efectos a tener en cuenta durante las pruebas en esta investigación:

---

### 3.1.2.1 Efecto *Hard-Iron* y *Soft-Iron*

Estos dos efectos están presentes en todos los tipos de magnetómetros, son perturbaciones en el campo magnético que está midiendo producto de elementos magnéticos o ferromagnéticos cercanos al sensor; normalmente los dos efectos se presentan juntos. Para las siguientes descripciones, suponga que se evalúa un sensor magnético ideal de dos ejes, rotándolo para obtener un círculo como se observa en la Ilustración 13.

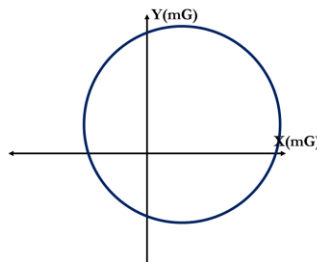
Ilustración 13. Evaluación de magnetómetro 2D ideal



#### **Efecto *Hard-Iron*:**

Esta distorsión es producida por la presencia de una fuente de campo magnético adicional a la que se desea medir, de la interacción resulta en un desplazamiento del centro del círculo graficado en la Ilustración 14.

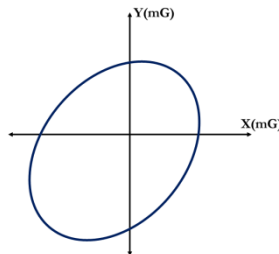
Ilustración 14. Efecto *Hard-Iron* 2D



#### **Efecto *Soft-Iron*:**

La cercanía de elementos ferromagnéticos resulta en una elongación del círculo esperado en la medición, el resultado se muestra en la Ilustración 15.

Ilustración 15. Efecto *Soft-iron* 2D

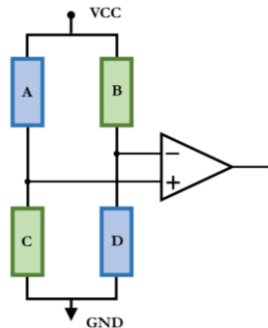


### 3.1.2.2 Efecto térmico

El aumento de temperatura ejerce dos reacciones en un sensor AMR en puente: primero, las resistencias aumentan linealmente. Segundo, la sensibilidad del Permalloy disminuye. Para comprender mejor esto se analiza matemáticamente.

Considere un puente de Wheatstone, cada magnetorresistencia del puente se llamará A, B, C y D respectivamente como se muestra en la siguiente imagen:

Ilustración 16. Puente AMR Wheatstone



El voltaje en la salida del puente está dado por la ecuación:

$$V_{out} = V_{cc} \left( \frac{C}{A + C} - \frac{D}{B + D} \right) \quad (3.13)$$

Cada MR responde a un coeficiente térmico  $(1 + \alpha\Delta\theta)$  propio de la aleación de níquel-hierro, donde  $\alpha$  es un coeficiente térmico y  $\Delta\theta$  es la variación de temperatura, que al reemplazarla en la ecuación (3.13) anterior resulta:

$$V_{out}(\theta) = V_{cc} \left( \frac{C(1 + \alpha\Delta\theta)}{(A + C)(1 + \alpha\Delta\theta)} - \frac{D(1 + \alpha\Delta\theta)}{(B + D)(1 + \alpha\Delta\theta)} \right) \quad (3.14)$$

Luego, con la ecuación generada se deriva en términos de temperatura, con esto se puede establecer qué factores alteran el voltaje:

$$\frac{dV_{out}(\theta)}{d\theta} = 0 \quad (3.15)$$

De la ecuación (3.15) queda indicado que teóricamente el puente AMR es independiente de los cambios de temperatura, pero esto es solo posible en condiciones ideales de voltaje y simetría de los magnetorresistores, cosa que en la práctica no sucede. Si se considera que el arreglo AC y BD tienen una corriente de entrada, la diferencia de voltaje en el puente es:

$$\Delta v(\theta) = IR_{total}(1 + \alpha\Delta\theta) \quad (3.16)$$


---

Al derivarse (3.16) en términos de la temperatura:

$$\frac{d\Delta v(\theta)}{d\theta} = \alpha I R_{total} \quad (3.17)$$

Finalmente, de la ecuación (3.17) se concluye: siempre que se tenga una corriente constante, la variación o corrimiento en las mediciones de campo en el sensor dependerán de un factor lineal  $\alpha$  propio de las láminas permalloy en el magnetómetro AMR.

### 3.1.3 Modelo matemático

Partiendo del conocimiento adquirido en el proceso de investigación documental, se ha seleccionado el modelo matemático más detallado y mejor justificado. La ecuación (3.18) obtenida del artículo *Magnetometer Calibration Using Inertial Sensors* [48] desarrolla y expone los efectos de cada factor dentro del modelo, la versión completa es la siguiente:

$$h_{m,t} = C_{sc} C_{no} (C_{si} R^{b_m b} R^{b_n} m^n + O_{hi}) + O_{zb} \quad (3.18)$$

Para explicar de mejor manera la ecuación (3.18), se describen en la Tabla 5 cada una de las variables:

Tabla 5. Descripción de variables del modelo matemático

Variable	Descripción	Tipo
$h_{m,t}$	Campo magnético dependiente de tiempo y fuerza magnética	Dinámico
$C_{sc}$	Matriz diagonal de sensibilidad de los ejes del magnetómetro	Estático
$C_{no}$	Matriz de no ortogonalidad de los ejes del magnetómetro	Estático
$C_{si}$	Matriz de efecto de campo <i>soft-iron</i>	Estático
$R^{b_m b}$	Matriz de rotación entre el cuerpo del equipo y el sensor	Estático
$R^{b_n}$	Matriz de rotación del cuerpo	Dinámico
$m^n$	Campo magnético local normalizado	Dinámico
$O_{hi}$	Vector de desviación por efecto <i>hard-iron</i>	Estático
$O_{zb}$	Vector de desviación por medición en el sensor	Estático

Fuente [48]

Debido a que la mayoría de las variables son estáticas, es posible simplificar la ecuación en una matriz 3x3 de distorsión  $\mathbf{D}$  y vector 3x1 de desviación  $\mathbf{O}$ :

$$\mathbf{D} = C_{sc} C_{no} C_{si} R^{b_m b}, \quad \mathbf{O} = C_{sc} C_{no} O_{hi} + O_{zb} \quad (3.19)$$

Para que estas consideraciones sean posibles, tanto  $\mathbf{D}$  como  $\mathbf{O}$  deben ser independientes del tiempo, eso significa que las distorsiones magnéticas sean constantes y que el sensor esté fijo de manera rígida sobre el cuerpo del equipo. Además, se asume que las desalineaciones pueden ser descritas como un matriz de rotación, además se adiciona el ruido como una variable dinámica  $e$ . Como resultado se obtiene la ecuación (3.20) :

$$h_{m,t} = DR^{bn}m^n + O + e_{m,t} \quad (3.20)$$

Este modelo se encuentra en los artículos seleccionados en el estado del arte en repetidas ocasiones y coinciden con la referencia a la publicación *Magnetometer Autocalibration Leveraging Measurement Locus Constraints* [54], la manera que expresan alternativamente el modelo es:

$$h_m = Ch^b + b + n \quad (3.21)$$

Donde  $C$  es una matriz 3x3 de corrección,  $h^b$  es el campo magnético medido,  $b$  el vector de desviación por efecto *hard-iron* y  $n$  el ruido del medio. La matriz de corrección es producto de la multiplicación de tres matrices:

$$C = C_s C_\eta C_\zeta = \begin{bmatrix} 1 + S_x & 0 & 0 \\ 0 & 1 + S_y & 0 \\ 0 & 0 & 1 + S_z \end{bmatrix} \begin{bmatrix} 1 & \eta_z & -\eta_y \\ -\eta_z & 1 & \eta_x \\ \eta_y & -\eta_x & 1 \end{bmatrix} \begin{bmatrix} 1 + \zeta_{xx} & \zeta & \zeta_{zx} \\ \zeta_{yx} & 1 + \zeta_{yy} & \zeta_{yz} \\ \zeta_{zx} & \zeta_{zy} & 1 + \zeta_{zz} \end{bmatrix} \quad (3.22)$$

Donde  $C_s$  contiene los factores de escala entre los ejes del sensor,  $C_\eta$  describe la no ortogonalidad y desalineación de los ejes del sensor respecto al cuerpo del dispositivo y  $C_\zeta$  expresa la deformación producida por el efecto *soft-iron*.

### 3.1.4 Especificaciones del sensor HMC5883L

Este es un magnetómetro de tres ejes de magnetorresistencias anisotrópicas resistivas, según el fabricante está recomendado para el diseño de sistemas de navegación terrestres y servicios basados en localización. Las principales características se mencionan en la siguiente tabla:

Tabla 6. Características sensor HMC5883L

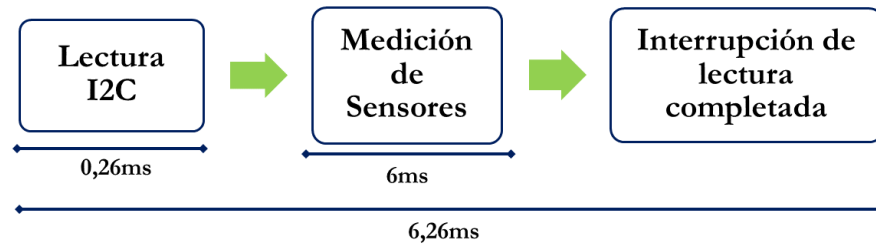
Característica	Detalle
Empaquetado	Cápsula de montaje superficial de (3.0x3.0x0.9) mm
Tipo	Sensores AMR digitales con ADC de 12 Bit de resolución.
Resolución	2 mG
Rango de medición	$\pm 8$ G
Tasa de muestreo	Hasta 160Hz, configurable
Resolución digital	(0.73 a 4.35) mG
Rangos de Ganancia	( $\pm 1$ , $\pm 2$ , $\pm 4$ y $\pm 8$ ) Gauss
Linealidad	0.1% FS en modo de ganancia $\pm 2$ G
Histéresis	$\pm 25$ ppm
Tolerancia de ganancia	$\pm 5\%$
Sensibilidad a temperatura	-0.3%/°C
Temperatura operativa	(-30 a 85)°C
Interfaz	I <sup>2</sup> C 100/400KHz
Voltaje operativo	1.8V, 2.5V, 3.3V
Adicionales	Bandas de desmagnetización integradas

Fuente: [57]

---

Internamente el sensor emplea un multiplexor y un solo convertor ADC para leer cada eje del magnetómetro. En consecuencia, aunque se realice un proceso de lectura acelerado no será posible obtener nuevas mediciones más allá de las que permitan el convertor y el multiplexor. Además tampoco es posible acceder únicamente a un solo eje puesto que internamente el controlador realiza la actualización de los tres ejes cada vez que se solicita una lectura. Los ciclos de lectura están resumidos en la Ilustración 17.

Ilustración 17. Ciclo de lectura HMC5883L



Fuente: [57]

El fabricante sugiere una rutina para obtener un factor de escala para la compensación de temperatura, en la que se toman una cantidad de muestras en un espacio de tiempo en la que la temperatura también ha variado, luego se toma el promedio de esas medidas dividiendo al último valor obtenido en condiciones “de calibración”. El factor obtenido se multiplica en cada nueva medición de su eje respectivo entregando un valor compensado.

## 3.2 CONDICIONES DE PRUEBA

La Organización Internacional de Metrología Legal (OIML), la Oficina Internacional de Pesos y Medidas (BIMP) y la Organización Internacional de Estandarización (ISO) son las entidades de referencia en cuanto a definiciones y procesos de metrología se refiere y que se tendrán en cuenta para el proceso de descripción de la prueba con el sensor en función de la disponibilidad de recursos para realizarla.

### 3.2.1 Métodos de medición de prueba o calibración

En primera instancia aplicando la normativa ISO/IEC 17025 [58] se definen los métodos de prueba o calibración según origen:

- Métodos Normalizados: Métodos descritos en normas de organizaciones técnicas reconocidas.
- Métodos desarrollados por laboratorio: Son métodos técnicos y documentados internamente pero que no cuentan con el respaldo de un estándar. Normalmente son procesos no estandarizados para complementar la ausencia de algún recurso dentro de un método estándar.
- Métodos no-normalizados: Son métodos experimentales que carecen de soporte completo y referencias estandarizadas.

En el caso de este trabajo, responde a un método **no-normalizado** por el carácter investigativo y experimental.

### 3.2.2 Principio de medición

Esta descripción proviene del Vocabulario Internacional de Metrología de OIML [59], en este caso tiene que ver la base científica de una medición. Considerando lo expuesto en los capítulos anteriores, no es posible alterar la técnica de medición para este trabajo porque el sensor tiene integrado el elemento sensor y los circuitos que acondicionan y digitalizan la señal.

### 3.2.3 Método de medición

Definido por OIML [59], hace referencia a las operaciones lógicas y ordenadas empleadas para la ejecución de las mediciones. El método de medida puede tener varias calificaciones como:

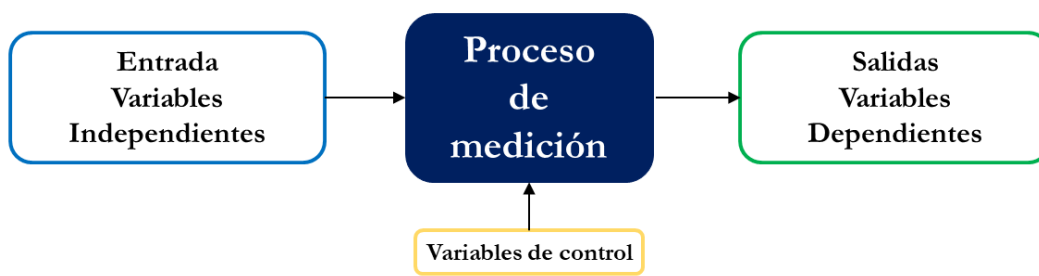
- Medición directa: Se emplea el instrumento de medición directamente sobre el fenómeno o elemento a medir.
- Medición indirecta: En este caso se mide un efecto producto del fenómeno a medir. Por ejemplo, determinar la temperatura midiendo el cambio en la densidad de un líquido.
- Método de sustitución: Utilizando un equipo auxiliar llamado comparador, se modifica al mensurando hasta llegar a un nivel de referencia.
- Método diferencial: Es considerado el método más exacto, consiste en determinar la diferencia entre un valor de referencia y uno desconocido.
- Método nulo o de cero: Con la ayuda de un detector de nulos como comparador se busca comprobar la igualdad entre el mensurando y un valor patrón.

Con los recursos actuales, sólo la medición directa es posible por la falta de instrumentos especializados y un laboratorio acondicionado.

### 3.2.4 Proceso de medición

Según el estándar ISO 10012 [60] la medición consiste en un conjunto de pasos ordenados para determinar el valor de una magnitud. Para ello hay que tener en cuenta los fenómenos físicos, efectos secundarios, condiciones ambientales y materiales que inciden la medición. Estas variables se modelan dentro de un proceso como se muestra en la Ilustración 18.

Ilustración 18. Proceso de medición





Para el caso particular de este trabajo las variables ya están determinadas en el modelo matemático elegido según el estado del arte como está indicado en la sección Modelo matemático (ver pág. 31). El inconveniente que superar está en las condiciones ambientales por la falta de un laboratorio con temperatura y humedad controlada, en este caso se tendrá que compensar de alguna manera de forma matemática.

### **3.2.5 Método de calibración**

En rigor del término en esta investigación no se está realizando una calibración, por esa razón en el título está indicado un proceso de estimación. En ese orden, el método no es tan estricto con el patrón de medición que es fundamental para una calibración. Para este trabajo no hay instrumentación disponible para medir campo magnético, por tanto, se realizará medición directa con los sensores HMC5883L a evaluar. Como patrón se utilizará la información del campo magnético terrestre obtenido del Modelo Magnético Mundial (ver pág. 10).

Finalmente, la estimación de parámetros con estas herramientas se ejecuta aplicando la técnica de calibración directa [59], donde el valor de patrón obtenido por WMM será utilizado para escalar la salida del sensor HMC5883L. Las diferencias entre el magnetómetro sin ajustar y MMM serán los factores de donde se obtienen los parámetros estáticos que se desean obtener.

## **3.3 PROCEDIMIENTO DEL EXPERIMENTO**

En este punto se tiene la base científica y técnica definida, permitiendo articular los pasos a realizar para obtener los parámetros estáticos de corrección de un magnetómetro. Dichos pasos son los siguientes:

### **3.3.1 Selección de entorno de medición**

Idealmente se emplearía un sitio con temperatura y humedad controlado, pero en el particular también se necesita un sitio alejado de metales, dispositivos de emisión de bajas frecuencias electromagnéticas, fuentes conmutadas, etc. Resulta difícil conseguir un laboratorio disponible que cumpla con estas condiciones en la región, por esta razón se evaluará el dispositivo en dos entornos: Primero será en un entorno abierto, en una zona rural, donde no hay presencia de ruido electromagnético artificial intenso y servirá de referencia para determinar el nivel de ruido en otras zonas de prueba. El segundo será una habitación, ubicando el magnetómetro alejado de metales a 1.5m a la redonda como mínimo, en horas de la noche donde las emisiones electromagnéticas son menores y las distorsiones por tormentas solares tienen menos efecto, si es que las hay.

### **3.3.2 Arquitectura de hardware**

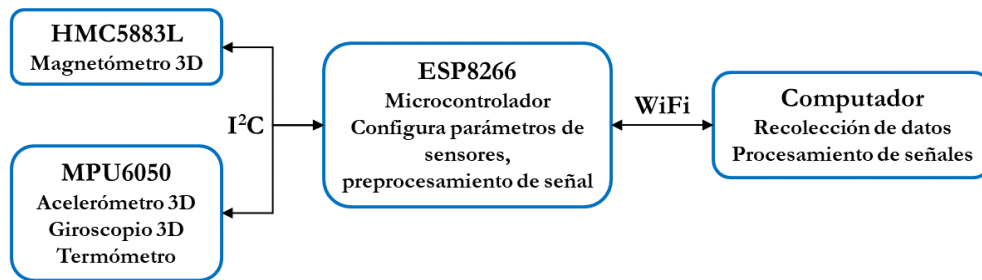
Como ya se ha mencionado, el magnetómetro seleccionado, es el integrado HMC5883L de Honeywell. Este dispositivo requiere el uso de un microcontrolador intermedio con protocolo I<sup>2</sup>C y otra interfaz que permita exportar los datos hacia un computador para su posterior análisis. El microcontrolador ESP8266 de Espressif es un chip de bajo costo compatible con varias interfaces de comunicación cableadas como SPI, UART e I<sup>2</sup>C y además cuenta con una

---

implementación TCP/IP para comunicarse por el protocolo IEEE 802.11 conocido comúnmente como WiFi; debido a la baja frecuencia de operación del magnetómetro la radiación del módulo de comunicación no interfiere en la medición de campo magnético.

El módulo que contiene el magnetómetro también cuenta la unidad de medición inercial (IMU, en inglés *Inertial Measurement Unit*) MPU6050 de Invensense, este dispone de un sensor de temperatura con resolución de 12 bit que resultará útil en las pruebas. Con esto, la arquitectura de comunicación entre el sensor y el computador será:

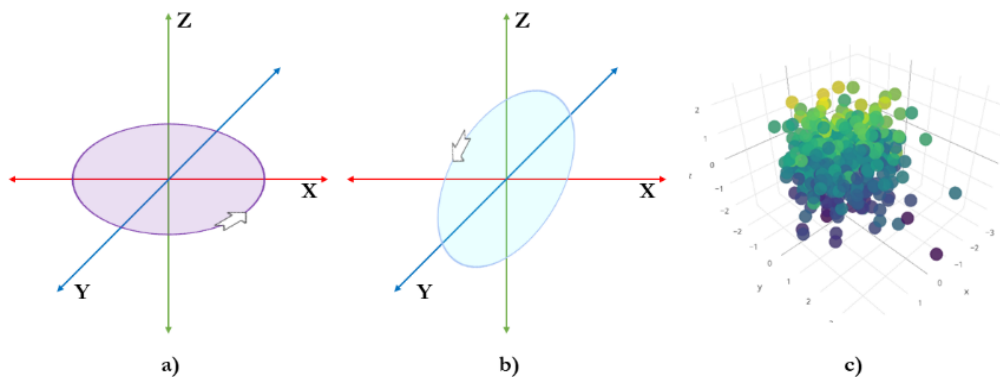
Ilustración 19. Arquitectura básica del hardware



### 3.3.3 Toma de muestras

Según lo consultado en el estado del arte, la toma de medidas sobre ejes ortogonales (Ilustración 20 a y b) no es garantía de calidad en las muestras, para solucionar esto algunos autores recomiendan realizar junto con las medidas ortogonales unas de movimiento libre (Ilustración 20 c) que ofrece una nube de puntos de medida más detallada. Las muestras tendrán marcas de tiempo y temperatura, esto permitirá detectar posibles eventos dinámicos asociados a la temperatura o variaciones eléctricas.

Ilustración 20. Trazados de medición



Fuente: a) propia, b) propia, c) [61]

Para lograr los movimientos exactos sobre los planos ortogonales el sensor debe disponer de un medio que permita ubicarlo de manera horizontal al plano terrestre, con esto la magnitud de campo H del WMM (ver pág. 11) será la referencia.

Para esto se propone el diseño de una plataforma construida con materiales diamagnéticos, en lo posible, para reducir potenciales efectos *hard-iron* y *soft-iron*. Esta contará con un motor paso a paso, ya que este tipo de motores están diseñados para mantener las líneas de campo encofradas en el chasis, lo cual reduce la intensidad de campo magnético detectable en las proximidades del motor. En el caso de las tomas de muestras con movimiento aleatorio se realizarán manualmente gracias a baja interacción magnética del cuerpo humano y la libertad de movimiento que tiene.

#### **3.3.4 Captura de datos y procesamiento**

Se requiere el diseño de una herramienta software que facilite la gestión de tomas de muestras con el magnetómetro. Como el dispositivo es inalámbrico, el firmware del microcontrolador y software de recepción en el computador deben garantizar en su comunicación una baja tasa de latencia y de pérdida de paquetes. Los datos recibidos se almacenarán en formato de valores separados por comas (CSV, en inglés *comma separated values*), que es fácil de generar y de abrir posteriormente en distintas soluciones de software.

Los archivos CSV generados en las tomas de muestra fueron evaluados en Matlab donde se realizaron los cálculos pertinentes para la estimación de parámetros que es el objetivo de este trabajo.

## 4 DESARROLLO DE HERRAMIENTAS

---

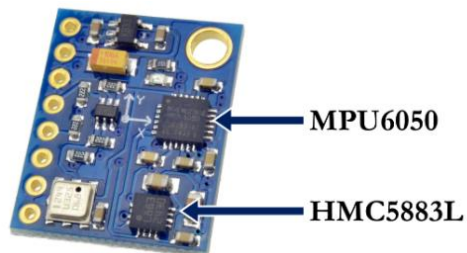
En esta sección se describen los procesos de desarrollo realizados para la consecución de las herramientas necesarias para la ejecución del experimento planeado, considerando la base científica expuesta anteriormente.

### 4.1 CONSTRUCCIÓN DE NODO

Se ha decidido emplear el término nodo para el dispositivo que integra el sensor, debido a que se trata de un elemento dentro de una estructura de red. Para construir el nodo se emplearon módulos electrónicos comerciales disponibles en Colombia, lo que redujo costos y facilitó la adquisición de varios de estos tanto para replicar nodos como para repararlos. Los módulos seleccionados son:

**GY-87:** Mostrado en la Ilustración 21. Es un módulo de tamaño (16.7 x 21.2) mm, integra magnetómetro HMC5883L, IMU MPU6050 y barómetro BMP180. Opera a 5V o 3.3V, todos los chips se comunican por el protocolo I<sup>2</sup>C. Diseñado para construir robots, en especial vehículos aéreos no tripulados.

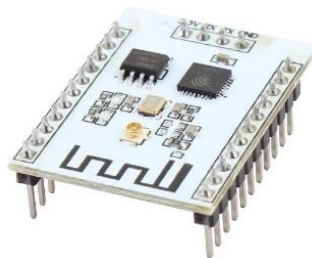
Ilustración 21. Módulo GY-87



Fuente: [62]

**ESP-201:** Este módulo del chip ESP8266 es una solución económica que deja a disposición del usuario una cantidad importante de pines, que son necesarios para programarlo y conectar el módulo GY-87.

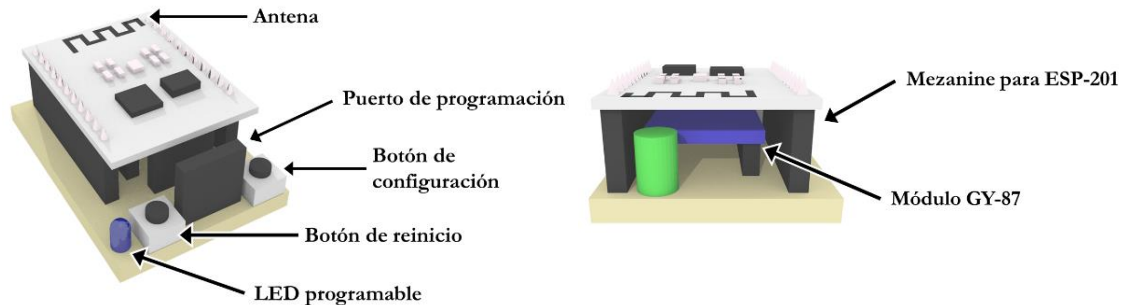
Ilustración 22. Módulo ESP-201



Fuente: [63]

Una vez seleccionados los dos módulos se hizo necesario el diseño de una tarjeta que permitiera integrarlos en un circuito que cumple con algunas condiciones: flexibilidad para las pruebas, de reparación fácil, conexión para programación y operación con baterías. Con la ayuda de la herramienta Blender, se modelaron los módulos ubicándolos sobre un prediseño de la tarjeta de integración. En la Ilustración 23 se observa el resultado de ese boceto tridimensional.

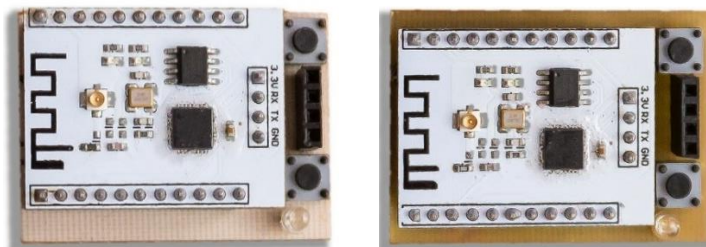
Ilustración 23. Boceto 3D de nodo



Como se observa en la ilustración anterior, se ha optado por apilar los dos módulos gracias a la diferencia de tamaño entre ellos reduciendo el espacio ocupado por el nodo, lo que facilitó ubicarlo en cualquier zona para pruebas. Además esta construcción reduce el flujo del aire sobre los sensores, esto ayudó a estabilizar la temperatura. El uso de un mezanine para instalar el módulo ESP-201 permite reemplazarlo con facilidad y acceder al módulo GY-87 cuando sea necesario. El botón de configuración tiene dos utilidades: la primera es de programación, ya que el microcontrolador necesita ser reiniciado mientras uno de sus pines está en cero lógico, pin que está asociado a este botón. Como segunda función, permite configurar modos de operación del nodo sin la necesidad de reprogramar o conectarse a la red WiFi.

Con el boceto como referencia, se diseñó el modelo del circuito impreso y se fabricaron dos unidades para utilizarlas en el montaje de dos nodos de prueba, los planos técnicos quedan consignados en el anexo A y las dos unidades fabricadas se muestran en la Ilustración 24.

Ilustración 24. Nodos fabricados



Imágenes a escala 1:1

Para más detalles técnicos consulte el anexo A - *Esquemas de nodo de sensores*.

## 4.2 FIRMWARE DEL NODO

Debido a la popularidad del microcontrolador ESP8266, ha sido portado el compilador de este chip para plataforma de desarrollo Arduino lo que facilita el desarrollo de código. El soporte es de código abierto y está bien documentado [64], tiene implementadas las funciones para configurar una comunicación por protocolo UDP o TCP y para la interfaz I<sup>2</sup>C. De igual manera los sensores HMC5883L y MPU6050 disponen de librerías de código abierto compatibles con Arduino. Con esta base de firmware lista, solo se hace necesario para este trabajo el desarrollo de código que configure e integre estos dispositivos y ofrezca las funcionalidades necesarias para realizar las mediciones. La documentación técnica del código se consigna en el anexo C – *Documentación Del Firmware En El Nodo*, respecto a la utilización consulte el anexo B – *Instructivos De Uso Del Nodo*. Para comprender el funcionamiento, los componentes más importantes se describen a continuación:

### 4.2.1 Configuración y lectura de sensores

Tanto el magnetómetro HMC5883L como la unidad de medición inercial MPU6050 tienen parámetros de configuración, muchos de estos no requieren ajuste para realizar una prueba estándar. Sin embargo, se ha decidido dejar algunos configurables, estos se editarán a través de las comunicaciones entre el nodo y computador para prevenir la necesidad de reprogramar el firmware.

#### Configuraciones

En el caso del magnetómetro según los datos del fabricante hay tres registros de configuración [57], se ha optado por dejar el siguiente plan de ajuste:

- Modo de operación: Single-Measurement, este modo permite la frecuencia máxima de actualización de los AMR que es de 160 Hz.
- Modo de medición: Normal, esto significa que no se forzará el sentido de corriente sobre los AMR, el magnetómetro emplea el flujo definido por el fabricante
- Filtro pasa bajas: Desactivado, las muestras no se promedian.
- Modo de energía: Siempre activo.
- Ganancia: Variable, el sensor tiene ocho niveles de ganancia ajustable y será un valor configurado por el usuario.

Para la IMU existe un mapa de registros de configuración mucho más extendido [65], no obstante, muchos de estos ni siquiera son usados a menos que se activen funciones especiales del sensor, los ajustes para este integrado son los siguientes:

- Sincronización externa: Desactivada, los sensores operan con el reloj interno como referencia.
- Digital Motion Processor: Desactivado, en este modo la IMU calcula internamente el cuaternión de posición. Se desactiva para procesar manualmente los datos.
- I<sup>2</sup>C By-Pass: Activada, en el módulo GY-87 es necesaria tener activada esta función para acceder al magnetómetro debido a que este último está conectado a la IMU y no al bus I<sup>2</sup>C.

- Fuente de reloj: Eje X de giroscopio, el fabricante recomienda utilizar como referencia de reloj uno de los ejes del giroscopio o un reloj externo para lograr estabilidad.
- Sleep: Desactivado, los sensores están siempre activos y en lectura.
- Filtro pasa bajas: Variable, el usuario podrá ajustar los niveles de filtrado o desactivarlo.
- Ganancia de giroscopio: Variable, la escala se podrá ajustar entre 0 y 3 que equivale a  $\pm 200^\circ/s$  y  $\pm 2000^\circ/s$  respectivamente.
- Ganancia de acelerómetro: Variable, la sensibilidad del acelerómetro podrá ajustarse en valores entre 0 y 3 equivalentes  $\pm 2g$  hasta  $\pm 16g$  (g son gravedades terrestres).

### Lectura

Tanto el magnetómetro como la IMU cuentan con pines de interrupción para señalar cuando hay nuevas lecturas completadas en sus respectivos sensores, pero actualizar estas lecturas por medio de interrupciones no es posible debido a que la comunicación WiFi puede verse cortada en uno de estos eventos y en consecuencia producir pérdidas de paquetes. Para compensar esta falencia, los integrados tienen registros de estado que permiten saber cuándo hay una nueva lectura y son usados en casos como este.

Para leer los sensores se programa un ciclo continuo que consulta los registros de estado, cuando se detecta un cambio en el magnetómetro o en la IMU y se actualizan sus lecturas en las respectivas variables de memoria en el microcontrolador para que sean transmitidas en el próximo paquete. Para reducir la pérdida de muestras al mínimo, el ESP8266 deberá emitir continuamente los datos de los sensores que tenga en memoria, dentro de un límite de tiempo controlado para evitar saturación en la red. El protocolo de transmisión elegido es UDP por su orientación a tiempo real, en cada paquete transmitido está incluida una marca de tiempo que permitirá saber si se ha perdido alguna de estas. Los datos tomados de los sensores para emisión se procesarán en un filtro de pasabajas configurable y por esta razón son de tipo *double*, las variables en el firmware están indicadas en la siguiente tabla:

Tabla 7. Datos de sensores

Nombre	Tipo	Descripción	Bytes
mag	Estructura, double x 3	Datos de magnetómetro	24
gyro	Estructura, double x 3	Datos de giroscopio – IMU	24
acc	Estructura, double x 3	Datos de acelerómetro – IMU	24
temp	double	Datos de termómetro - IMU	8
Total			80

La marca de tiempo se obtiene por un contador interno del microcontrolador que se incrementa cada 1 ms por interrupción de un *timmer* y está almacenada en una variable de 32 bits. A este dato de tiempo, se agrega la información del voltaje de entrada en el regulador del nodo en otra variable de 32 bits, esto es útil para detectar ruido en la alimentación del circuito o controlar la operación del nodo en caso de tener la batería baja. Junto con los datos de los sensores, estos suman 88 bytes, esta información es importante para calcular la velocidad de transmisión del nodo en la red.

#### 4.2.2 Preprocesamiento de señal

Del modelo matemático del magnetómetro (ver pág. 31) se ha establecido una dependencia de tres factores fundamentales que afectan la medición: corrimiento, ganancia y temperatura. Según los autores consultados, son fenómenos lineales y por tanto al operar factores de corrección con los valores de salida en cada eje se compensarían los efectos ya mencionados. Con ese objetivo han sido incluidos unos factores de corrección que se aplicarán directamente en el microcontrolador como ayuda al tratamiento matemático realizado en el computador posteriormente, estas correcciones se controlarán según la necesidad de prueba que tenga el usuario. Los factores en cuestión son:

##### **Factor *offset***

Son valores del mismo tipo de resolución digital que la de los sensores, es decir de 16 bits. Cada eje de los sensores tendrá asociado un factor de *offset* que, de estar habilitado, se le restará directamente.

##### **Factor *gain***

Estos factores *double* ajustarán la escala de cada uno de los ejes de los sensores, son números tipo *double* que se multiplican con su respectivo eje.

##### **Factor *alpha***

Según la descripción del fabricante y del modelo matemático, la temperatura tiene un efecto lineal sobre el magnetómetro, los factores *alpha* también de tipo *double*, serán multiplicados por la temperatura medida en la IMU y el resultado restado a la salida de cada eje.

Estos factores por cada eje del sensor ocupan 18 bytes, si se aplican a los nueve ejes del nodo estos factores ocuparían un total de 162 bytes.

#### 4.2.3 Configuración del nodo

El microcontrolador ESP8266 no tiene memoria interna, por esta razón debe tener conectada una unidad de memoria FLASH donde se almacena el código de ejecución e información útil para el usuario. Aprovechando este recurso de memoria se almacenan siete parámetros de configuración que podrán cambiarse a través de la comunicación inalámbrica, esto evitará la necesidad de reprogramar el nodo para ajustar configuraciones de los sensores o del microcontrolador, cada parámetro está representado por un dato de un byte. La Tabla 8 describe los parámetros implementados.

El nodo por defecto opera en modo de punto de acceso (*Access Point*), es decir que crea una red WiFi local a la que se puede conectar cualquier dispositivo de red compatible. El nombre de la red que crea el nodo está programado en el firmware y no cuenta con cifrado alguno. En el caso del modo Cliente, el dispositivo se conectará a una red configurada previamente en el firmware, estos ajustes de red no son posibles de cambiar a través de la comunicación inalámbrica debido a que estas funcionalidades requieren tiempo para la implementación y no resultan críticas para cumplir los objetivos de esta investigación.



Tabla 8. Parámetros de configuración de nodo

Nombre	Dir.	Descripción	Valores
CFG	0x00	Cada bit configura una función de programa	Bit 0: Modo Wifi – 0 WiFi en modo Access Point, 1 Modo cliente. Bit 1: Transmisión Serial – Envía datos de sensores en formato JSON Bit 2: <i>Offset</i> – Habilita la compensación de corrimiento. Bit 3: <i>Thermal</i> – Activa la compensación lineal de temperatura Bit 4: <i>Gain</i> – Configura la compensación de ganancia Bit 5-7: Reservado
ACC_SCALE	0x01	Escala de acelerómetro	0 - 3, Ajuste desde $\pm 2g$ hasta $\pm 16g^*$
GYRO_SCALE	0x02	Escala de giroscopio	0 - 3, Ajuste $\pm 250$ °/s a $\pm 2000$ °/s
MAG_SCALE	0x03	Escala de magnetómetro	0 - 7, ganancia de AMR desde $\pm 0.88Ga$ hasta $\pm 8.1Ga$
IMU_DLPF	0x04	Filtro pasabajas de IMU	0 - 6, Ancho de banda desde 260Hz hasta 5Hz
SWLPF	0x05	Filtro pasabajas en microcontrolador	0-255, filtro de ventana dependiente del tiempo de lecturas de cada sensor
TX_PERIOD	0x06	Periodo de emisión de lecturas	5-255, configura en milisegundos periodo de tiempo entre cada paquete.

\*g son gravedades terrestres según la hoja de datos del sensor [65]

El botón de configuración está habitado para tres funcionalidades: cambio de modo WiFi, activación de transmisión serial y reinicio de parámetros del nodo. Para ello se implementa un código de pulsaciones del LED en la tarjeta de integración. El procedimiento inicia al encender el nodo, en ese momento el usuario tiene dos segundos para presionar el botón antes que el nodo se active. El LED emitirá pulsaciones mientras el botón se mantenga presionado, la cantidad de pulsaciones indica el modo al que se quiere ajustar (ver Tabla 9), una vez encontrado el modo deseado se libera el botón. El nodo guardará las nuevas configuraciones y reiniciará automáticamente para luego ejecutarlas.

Tabla 9. Modos de ajuste con botón

Pulsaciones de LED	Función
1	Conmuta entre el modo WiFi de <i>Access Point</i> o Cliente.
2	Activa o deshabilita la transmisión de lectura de sensores por puerto serial.
3	Reinicia todas las configuraciones, borra parámetros <i>alpha</i> , <i>offset</i> y <i>gain</i> .
4 o más	Reservado, no se hace nada y el nodo continúa la inicialización.

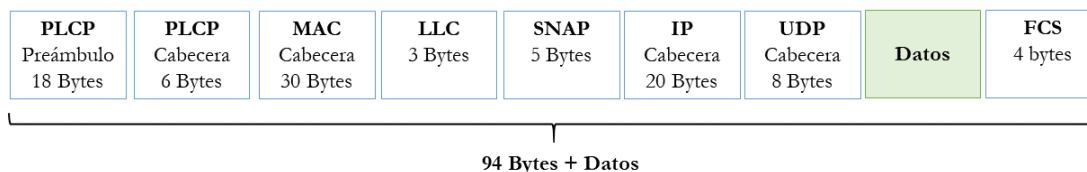
#### 4.2.4 Protocolo de comunicación

El microcontrolador ESP8266 integra una pila de protocolos para cumplir con el estándar IEEE802.11, esto significa que el chip soporta la implementación de una red de arquitectura IP con protocolos de transmisión TCP y UDP. TCP es un protocolo de transmisión controlado, orientado al flujo de datos sin pérdidas gracias al uso de marcas en los paquetes para identificarlos. Por su parte UDP emite datagramas sin la necesidad de crear una conexión, simplemente emite el flujo de datos de manera constante sin ningún control, esto da una ventaja para aplicaciones de tiempo real, porque si se pierde un datagrama simplemente se ignora y se continua con la emisión. En el caso del nodo de sensores, UDP es ideal porque los datos de los sensores son actualizados constantemente y una pérdida de una de esas lecturas no resulta crítica en un proceso en que se capturan cientos o miles de muestras. Considerando lo anterior, se necesitan los siguientes datos para establecer una conexión con un nodo:

- **Conexión física:** Se consigue conectando el nodo como cliente a una red existente, en la que necesita el nombre de la red (SSID) y la clave (PASS) si hay cifrado programado en el firmware previamente. Cuando el nodo opera en modo de punto de acceso, el computador debe conectarse a la red que haya creado el nodo con un SSID ya definido en el firmware.
- **Conexión lógica:** Refiere al direccionamiento lógico dentro de la red con direcciones IP, una aplicación necesita la dirección IP del nodo para acceder a los datos, para ello se deja definido en el firmware una dirección IP estática.
- **Conexión de transporte:** En el caso de UDP no se da una conexión como tal, simplemente el computador y el nodo tienen asignados un número identificador, conocido como puerto UDP. Por defecto el nodo tendrá un puerto UDP en el que espera un mensaje desde el computador para establecer la conexión, ese mensaje tendrá el dato del puerto asignado en el computador para recibir la información del nodo. Luego el nodo emitirá continuamente la información de los sensores, si el computador se desconecta el nodo no tendrá como saberlo y seguirá emitiendo hasta que sea apagado, por esta razón se dice que no existe una conexión formal. Como un paquete UDP finaliza con un carácter *Null* (cero) es necesario codificar la información del nodo en base64 para evitar pérdidas de datos, además funciona como un método de detección de errores en los paquetes recibidos.

Según el estándar IEEE 802.11 y RFC 768 [66] un datagrama UDP sobre un paquete 802.11 tendría la siguiente estructura:

Ilustración 25. Datagrama IEEE 802.11 + UDP



#### 4.2.4.1 Transmisión de muestras

Para enviar los datos de los sensores se necesitan emitir 88 Bytes, para identificarlo se adiciona el carácter “\*” al inicio totalizando 89 Bytes. Al codificar la trama en base64 se obtienen 120 Bytes al que se adiciona de nuevo el carácter “\*”, esta vez al final para emplearlo como bandera de cierre. Esto significa que sumando los 121 Bytes de datos más los 94 Bytes de protocolo, cada paquete necesita transmitir un total de 215 Bytes por paquete. Según el estándar 802.11 la peor velocidad de conexión media en una red de este tipo ofrece un ancho de banda de 1 Mbps, luego:

$$PPS = \frac{1Mbps}{215Bpp} = \frac{1 \times 10^6 bps}{1720 bpp} = 581.39 p/s \quad (4.1)$$

La ecuación (4.1) indica que en una mala condición de señal es posible transmitir hasta 581 paquetes en un segundo, considerando que el magnetómetro tiene una frecuencia máxima de operación de 160Hz y la IMU de 120Hz, el enlace de datos garantiza que no habrá pérdidas de información.

La estructura de datos que recibirá la aplicación en el computador después de la decodificación de base64 es la siguiente:

Tabla 10. Datos de sensores – datagrama decodificado

Posición	Tamaño(byte)	Tipo	Descripción
0	1	Uint8_t	Carácter de inicio “*”
4-1	4	Uint32_t	Marca de tiempo en milisegundos
8-5	4	Uint32_t	Voltaje de batería
16-9	8	Long	Temperatura
24-17	8	Long	Acelerómetro eje X
32-25	8	Long	Acelerómetro eje Y
40-33	8	Long	Acelerómetro eje Z
48-41	8	Long	Giroscopio eje X
56-49	8	Long	Giroscopio eje Y
64-57	8	Long	Giroscopio eje Z
72-65	8	Long	Magnetómetro eje X
80-73	8	Long	Magnetómetro eje Y
88-81	8	Long	Magnetómetro eje Z

#### 4.2.4.2 Comunicación de parámetros

Estos datos son de gran importancia y necesitan un mecanismo bidireccional. Para la emisión de estos valores se empleará el mismo modo de transmisión de muestras, con la diferencia de que el nodo no lo hará de forma automática, sino que dependerá de unos comandos textuales enviados desde el computador. Los comandos implementados son:

- *get\_parameters*: Ordena al nodo transmitir todos los parámetros cargados en la memoria RAM
- *save\_parameters*: Guarda los parámetros de la memoria RAM en la memoria EEPROM

- *apply\_compensation*: Cuando se cambian las configuraciones de aplicación de parámetros de corrección offset, gain y alpha no se hace automáticamente, este comando actualiza estas configuraciones en caliente, evitando la necesidad de reiniciar el nodo.
- *apply\_imu\_set*: Al modificar los parámetros de configuración ACC\_SCALE, GYRO\_SCALE, MAG\_SCALE e IMU\_DLPF (ver Tabla 8, pág. 43) no se aplican en caliente al modificarlos en la RAM, este comando ordena configurar el magnetómetro y la IMU con esos nuevos valores sin la necesidad de reiniciar el nodo.
- *reset*: Ordena al nodo reiniciarse.

La gestión de parámetros está en tres niveles: en la memoria FLASH, memoria RAM y variables de aplicación. Cuando el nodo es encendido los parámetros son leídos desde la FLASH para copiarlos en un arreglo de variables en la memoria RAM. Los parámetros alpha, gain y offset son utilizados para los cálculos de compensación anteriormente descritos, pero los valores de configuración funcionan de distinta manera porque no tienen que aplicarse y consultarse continuamente. Por esta razón están implementados los comandos de aplicación ya mencionados.

### Consulta de parámetros

Con la comunicación entre el nodo y el computador previamente establecida, la consulta de parámetros del nodo se realiza con la siguiente secuencia de pasos:

- 1) El computador envía el comando *get\_parameters*
- 2) El nodo responde construyendo una trama con la siguiente estructura:

Tabla 11. Parámetros de nodo - datagrama decodificado

Posición	Tamaño(byte)	Tipo	Descripción
0	1	Uint8_t	Carácter de inicio “&”
1	1	Uint8_t	CFG
2	1	Uint8_t	ACC_SCALE
3	1	Uint8_t	GYRO_SCALE
4	1	Uint8_t	MAG_SCALE
5	1	Uint8_t	IMU_DLPF
6	1	Uint8_t	SW_DLPF
7	1	Uint8_t	TX_PERIOD
79-8	8 x 9	Long x 9	Parámetros alpha*
97-80	2 x 9	Uint16_t x 2	Parámetros offset*
169-98	8 x 9	Long x 9	Parámetros gain*

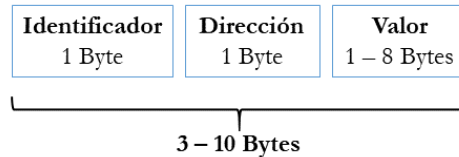
\*Es un arreglo asociado a los ejes de los sensores, en orden ascendente: acc.x, acc.y, acc.z, gyro.x, gyro.y, gyro.z, mag.x, mag.y, mag.z.

- 3) La trama generada es codificada en base64, se adiciona al final el carácter “&” y es transmitido.
- 4) El computador recibe el paquete, remueve el símbolo al final, decodifica en base64 y ordena las variables según el datagrama.

### Actualización de parámetros

Para reducir las colisiones entre los mensajes que envía el nodo con los datos de los sensores y los valores que recibe desde el computador para configurar parámetros, se opta por un protocolo sencillo en el que se configura un valor de parámetro a la vez. La trama tiene la siguiente forma:

Ilustración 26. Trama de recepción de parámetro



El tipo de parámetro que desea configurar con el mensaje debe corresponder con los identificadores consignados en la Tabla 12, en caso de no hacerlo el nodo desecha el mensaje. Es importante que la información del valor tenga el formato correspondiente, porque de no hacerlo se presentarán inconsistencias con el potencial de afectar el correcto funcionamiento del nodo.

Tabla 12. Recepción de parámetros - asignación de símbolos

Identificador	Dirección	Valor
# Configuración Tamaño de valor: 1 Byte	0	CFG
	1	ACC_SCALE
	2	GYRO_SCALE
	3	MAG_SCALE
	4	IMU_DLPF
	5	SW_DLPF
	6	TX_PERIOD
@      &      \$ alpha   gain   offset 8 Bytes   8 Bytes   2 Bytes	0	→ acc.x
	1	→ acc.y
	2	→ acc.z
	3	→ gyro.x
	4	→ gyro.y
	5	→ gyro.z
	6	→ mag.x
	7	→ mag.y
	8	→ mag.z

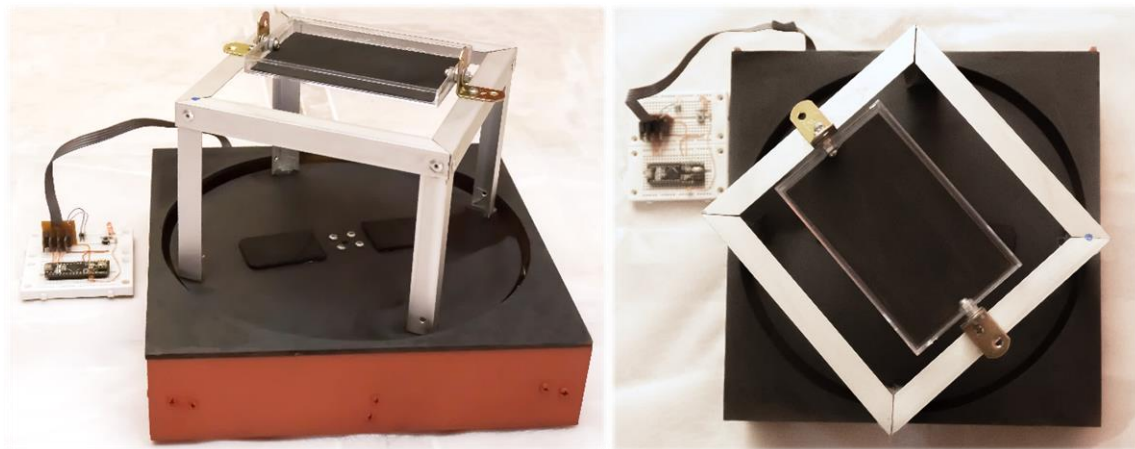
La secuencia de recepción de un parámetro es el siguiente:

- 1) El computador envía un valor de parámetro nuevo
- 2) El nodo recibe el parámetro y lo asigna en su espacio en RAM correspondiente
- 3) El computador consulta los parámetros y compara el valor cambiado. Si no hay correspondencia se repite el proceso.
- 4) Si el valor se actualizó correctamente y dependiendo de la situación, el usuario puede: Guardar el valor de forma permanente con el comando *save\_parameters*. Aplicar el cambio sobre configuración de magnetómetro e IMU con *apply\_imu\_set*. Aplicar la activación o desactivación de los compensadores con el comando *apply\_compensation*. Los valores de compensadores tienen efecto instantáneo al cambio cuando estos están activados.

### 4.3 PLATAFORMA ROTATIVA

Este instrumento permite de manera controlada ejecutar las mediciones de campo magnético de referencia. La plataforma está construida con un aglomerado de fibras de madera como material principal, uniones con tornillos que permiten ajustar la ubicación de las piezas buscando un ajuste preciso y para el movimiento un motor paso a paso. Como se presume, el motor es una fuente de emisión de campo y generador de distorsiones por sus componentes metálicos. Para solucionar esto, el disco rotativo que está sobre el motor tiene una estructura de aluminio adicional que aleja al nodo de sensores manteniendo el eje de rotación del disco. La plataforma construida se observa en la siguiente ilustración:

Ilustración 27. Plataforma rotativa



Como se observa en la Ilustración 27, la plataforma está acompañada de un microcontrolador destinado a manejar la velocidad y el sentido de rotación del motor, este dispositivo se comunica por comandos a través de su puerto serial adaptado a USB.

Sobre la estructura de aluminio está instalado un soporte plástico con un eje ajustable por tornillos, con el fin de tener control sobre nivel del horizonte en este soporte como se muestra en la Ilustración 28.

Ilustración 28. Eje de soporte



## 4.4 APLICACIÓN WIMO

Esta es una herramienta de software que tiene sus orígenes en una investigación previa para administrar parámetros de un autopiloto para un aeroplano pequeño. Emplea *electron*, que es una interfaz de programación para el desarrollo de aplicaciones de escritorio con lenguajes web como HTML, JavaScript y CSS. Su nombre deriva del inglés *Wireless Motion* (Wimo), esta aplicación está diseñada como complemento para manejar los datos generados en los nodos inalámbricos ya descritos, cuenta con la implementación completa para leer y ajustar los parámetros y una rutina para ajustarlos de manera asistida. La interfaz está compuesta por tres secciones llamadas: “Panel de datos”, “Configuración” y “Optimización” que se describen por funcionalidades y componentes gráficos:

### 4.4.1 Panel de datos

Tiene por objetivo presentar la información del sensor como un entorno para captura de datos en la realización de cualquier tipo de prueba con un nodo, como se muestra en la siguiente imagen:

Ilustración 29. Panel de datos de aplicación



### Funcionalidades:

- Panel para inicio de conexión por dirección IP del nodo.
- Visualización de datos de sensores en bruto con máximos y mínimos.
- Cálculo de cuaternión en modo 6 grados de libertad (acelerómetro y giroscopio) y 9 grados de libertad (acelerómetro, giroscopio y magnetómetro). Presentación numérica en formato de cuaternión NASA-JPL, ángulos de Euler y cubo animado en 3D.
- Presentación de estado del sensor: Tiempo de encendido, batería, temperatura y parámetros de configuración de nodo.
- Configuración de ubicación, necesario para el cálculo del WMM.
- Registro de la sesión a través de un archivo CSV.

#### 4.4.2 Configuración

Permite ajustar los parámetros descritos en la sección Configuración del nodo (ver pág. 42) mientras se visualizan las salidas de los sensores en graficas de tiempo real. Para acceder a esta pestaña de la aplicación es necesaria tener una sesión iniciada con un nodo.

Ilustración 30. Panel de Configuración



#### Funcionalidades:

- Configuración del registro CFG (ver Pág. 43), que permite activar los procesos de compensación *gain*, *offset* y *alpha*.
- Ajuste de configuraciones de ganancia y filtrado en magnetómetro y giroscopio.
- Visualización de cambios en configuración y salidas del nodo de sensores.

#### 4.4.3 Optimización

Esta sección resulta muy útil dentro del desarrollo de este trabajo, gracias a la rutina de cálculo de compensación térmica que requiere un proceso repetitivo y metódico para ajustar ese parámetro de corrección. También integra un mecanismo interactivo que ayuda a calcular los parámetros de offset y ganancia calculados directamente por el microcontrolador, sobre el tratamiento de datos y el uso de esos parámetros son detallados en el capítulo 5. El aspecto de esta sección de la aplicación se observa en la Ilustración 31.

#### Funcionalidades:

- Rutina de estimación de factores *alpha*.
- Tutorial interactivo para cálculo de parámetros *offset* y *gain*.
- Registro de los procesos anteriores.
- Visualización de señales de sensores en tiempo real.



Ilustración 31. Panel de optimización



Para mayor detalle sobre la aplicación consulte el anexo G – *WiMo APP*.

También se ofrece como un sustituto para captura de datos a un script Python, consulte el anexo D – *Script de captura de datos "CAPTURAR.PY"* para más detalles.

## 5 EVALUACIÓN

---

La base científica y las herramientas desarrolladas son el soporte de las pruebas que se describen en este capítulo. Los aspectos a verificar son: el efecto de la temperatura sobre los sensores, las condiciones de ruido entre un ambiente rural y uno urbano y finalmente la obtención de los parámetros estáticos del magnetómetro con su correspondiente validación.

### 5.1 COMPROBACIÓN DE EFECTO TÉRMICO

Al referirse al modelo matemático del magnetómetro (ver pág. 31) este excluye el efecto de la temperatura sobre el sensor. En las condiciones de prueba establecidas no hay un entorno disponible que permita excluir este parámetro sobre el sensor al momento de obtener muestras. Por esta razón, la caracterización sobre el efecto térmico es necesario.

#### **La estrategia para ejecutar esta medición:**

El nodo de sensores es inicialmente puesto a una temperatura cercana a 0 °C, luego el sensor es activado en posición estática en condiciones ambientales. El nodo está en modo *access point* para comunicarse debido a un defecto de operación del ESP8266 en modo cliente cuando opera a temperaturas menores a 5 °C. El filtro de software ajustado al máximo, promediando 255 muestras para reducir el efecto de ruido y con el magnetómetro en escala  $\pm 1.3$  G, que es la escala ideal para medir el campo magnético de la tierra sin la presencia de elementos magnéticos cercanos. El nodo se mantiene activo mientras alcanza un rango de variación de temperatura de por lo menos 15 °C más sobre la temperatura inicial de la prueba. Con esta información se analiza el comportamiento de los sensores ante una variación térmica para determinar si existe la relación lineal, descrita en la teoría analizada en los capítulos anteriores.

El cambio de temperatura en los circuitos produce una condensación de humedad que afecta la comunicación I<sup>2</sup>C entre los sensores y módulo del microcontrolador. Para prevenir este problema tanto el sensor como el módulo de comunicaciones son cubiertos con esmalte y como medida secundaria el nodo es contenido en un recipiente con bolsas de gel de sílice que absorben la humedad. En la Ilustración 32 se muestra la disposición del sensor dentro del recipiente mencionado.

Ilustración 32. Recipiente de prueba térmica



En la captura de información se usa el panel de datos de la aplicación Wimo, la conexión con el nodo se establece con la dirección IP 192.168.180.180 que es el valor por defecto en modo *Access point*, los compensadores integrados deben estar desactivados y preferiblemente los parámetros *alpha*, *gain* y *offset* en ceros. Una vez verificada la configuración y temperatura del nodo se habilita el registro de la sesión deslizando el interruptor “Crear registro”, automáticamente la aplicación crea un archivo CSV que se actualizará cada 2 minutos hasta que el usuario intervenga o el nodo pierda la conexión. En la aplicación el usuario verá datos como los mostrados en la siguiente imagen:

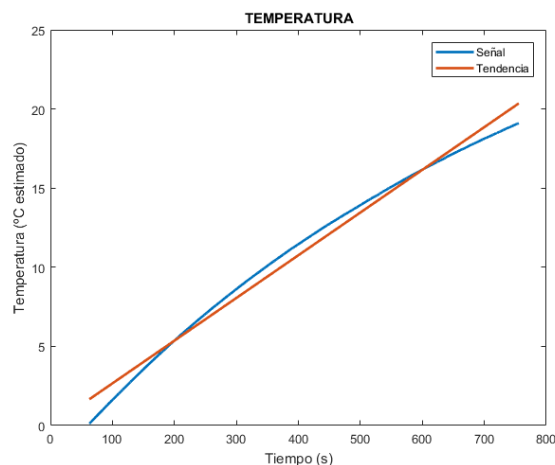
Ilustración 33. Wimo durante prueba



### Análisis de datos:

La información obtenida en la prueba es verificada con la ayuda de Matlab, empleando el script *thermal\_analysis* donde el archivo CSV es importado y se aplica un algoritmo de linealización en todos los datos de sensores entregados por el nodo. Como primera medida la variación de temperatura debe ser verificada, en la que debe existir un comportamiento constante de incremento similar al mostrado en la Ilustración 34.

Ilustración 34. Variación de temperatura



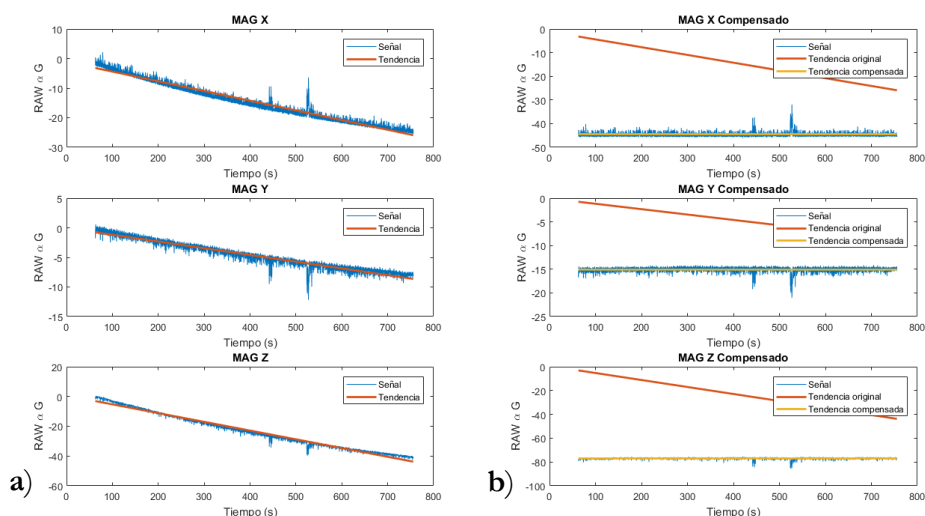
En el script está establecido un algoritmo que emplea las linealizaciones calculadas en los datos de los sensores. El factor de linealización de la señal del sensor se divide entre el factor de temperatura, luego este valor calculado se multiplica por la señal de sensor medida quedando la siguiente ecuación:

$$\Phi_{\text{comp}}(\mathbf{x}) = \Phi(\mathbf{x}) - \alpha\theta(\mathbf{x}) \quad (5.1)$$

Donde  $\mathbf{x}$  indica la posición de la muestra dentro del vector de datos.  $\Phi$  es la señal a compensar térmicamente,  $\theta$  responde a la temperatura en el momento de la muestra y  $\alpha$  el factor térmico calculado.

En la muestra obtenida del magnetómetro en cada uno de sus ejes mostrados en la Ilustración 35.a, se verifica que efectivamente las señales de los AMR son afectadas por la temperatura. Cuando se aplica el algoritmo de corrección estas curvas cambian a una tendencia plana, apreciables en la Ilustración 35.b, lo que confirma la correspondencia entre la temperatura y las magnitudes medidas en los sensores AMR de manera lineal.

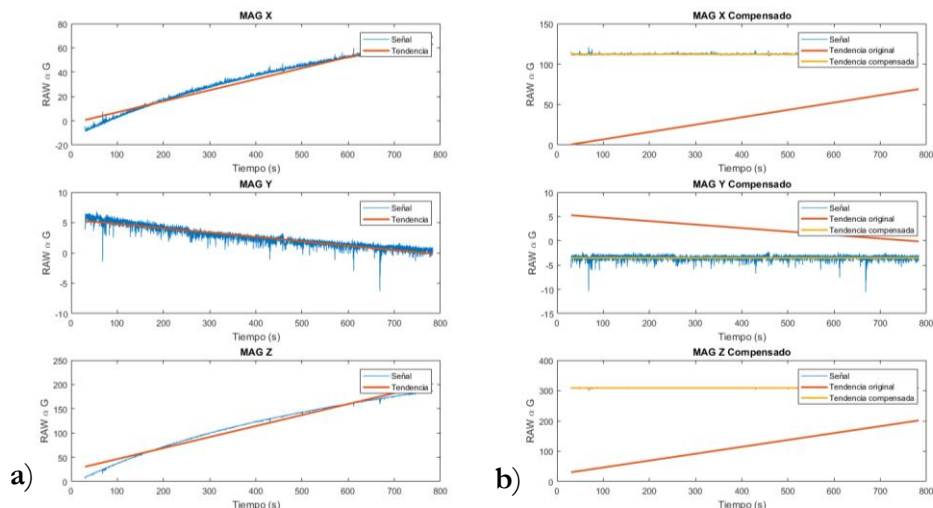
Ilustración 35. Comportamiento térmico de magnetómetro.



Los parámetros  $\alpha$  calculados son programados en la memoria y es activada la compensación térmica en el microcontrolador del nodo evaluado. Nuevamente el dispositivo es sometido a una prueba con cambio de temperatura, esta vez con la compensación que debería linealizar la salida de los sensores. Sin embargo, esto no sucedió como se muestra en la Ilustración 36 donde las tendencias de cada eje del magnetómetro presentan cambios de sentido, lo que demuestra que los parámetros si incidieron pero no con el valor adecuado.

### Conclusión:

En estas condiciones, se repitió el experimento diez veces más con resultados similares, esto permite concluir que el magnetómetro si tiene un factor térmico lineal, pero que es cambiante cada vez que se enciende el sensor, obligando a realizar la rutina de compensación en cada sesión de trabajo con el magnetómetro. Lo anterior mencionado se nota al comparar la Ilustración 35 y la Ilustración 36.

Ilustración 36. Magnetómetro en prueba térmica con compensación  $\alpha$  activada.

En el anexo E – *Script de corrección térmica “THERMAL\_ANALYSIS.M”* se ofrece el código fuente del script y un instructivo de uso.

## 5.2 EVALUACIÓN DE RESPUESTA DE RUIDO

Atendiendo a las condiciones de prueba especificadas en el capítulo 3 Diseño del Experimento, el comportamiento de ruido es estudiado bajo dos condiciones, la primera de ellas responde a una zona rural. La comparativa se realizará en los índices de señal a ruido y los histogramas generados de cada eje.

### 5.2.1 Prueba exterior

Para estas mediciones se acondicionó un sistema de alimentación de 5V que alimenta a el nodo y una carcasa plástica para evitar variaciones térmicas rápidas que provocan los vientos sobre los circuitos. El nodo es ubicado sobre una superficie nivelada como se observa en la siguiente imagen:

Ilustración 37. Nodo acondicionado a prueba externa



Con el fin de reducir el efecto de temperatura que produce la exposición directa al sol, el nodo es cubierto con una lona blanca que refleja la luz y ofrece una mejor condición térmica para la prueba. Previamente a las mediciones, el nodo es enfriado en una nevera y luego dejado en una zona a la sombra para ejecutar el cálculo de parámetros de corrección térmica, similar a la descrita en la sección inmediatamente anterior. Luego, el nodo es instalado en la zona de prueba (mostrada en la Ilustración 38) sin reiniciarlo para mantener la compensación de temperatura dentro de unos límites aceptables y dejado allí durante una hora con el filtro de ruido al mínimo.

Ilustración 38. Zona de prueba exterior



### **Análisis de datos:**

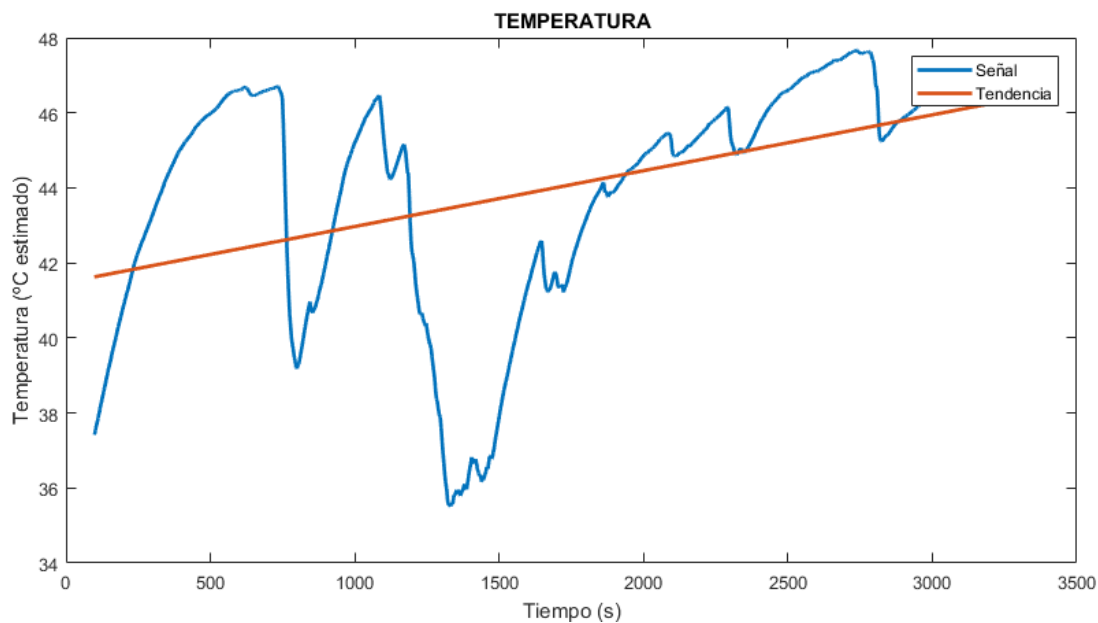
La prueba bajo las condiciones descritas se repitió tres veces para asegurar una validez mínima en los datos, también se tuvieron en cuenta las fuentes potenciales de interferencia producto de la actividad humana alrededor, independientes de la electrónica del dispositivo. Sobre estas fuentes, sus ubicaciones respecto al sensor son las siguientes:

- Transformador 60 kW: 200 m
- Casas ajenas a la prueba: 80 m mínimo
- Nevera: 16 m
- Televisor: 12 m
- Red eléctrica: 11m
- Computador: 10 m

En la ubicación de la prueba no hay cobertura de telefonía móvil; radio y televisión tienen una baja recepción. También se consultó en las casas vecinas si disponían de maquinaria industrial y si la usarían en el periodo de tiempo de la prueba, en todos los casos la respuesta fue negativa.

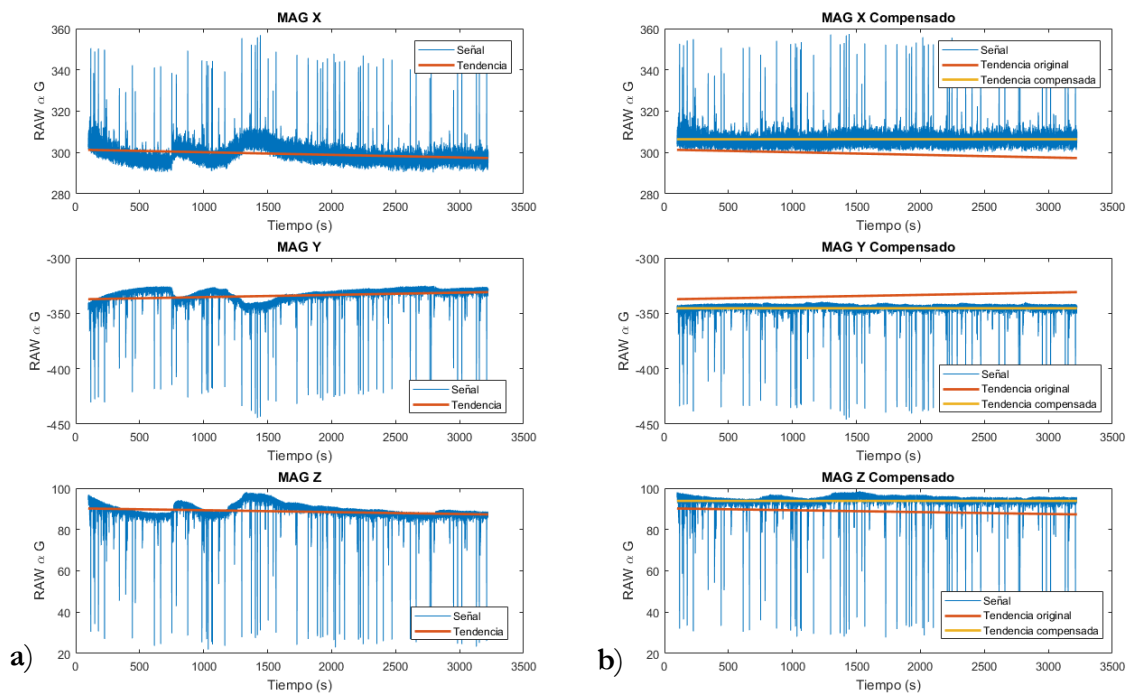
Con las condiciones de prueba en la mejor situación posible, se inició la captura de datos en un periodo entre las 9 a.m. y las 12 p.m., cada hora el nodo era reiniciado vía software para descartar cualquier problema que pudiera provocar el microcontrolador. El uso de la carcasa para mantener la temperatura en el nodo no evitó una fluctuación significativa y en el peor de los casos se registró una variación de 10 °C observable en la Ilustración 39.

Ilustración 39. Temperatura en el nodo en prueba exterior.



La gráfica anterior muestra una situación que fue constante durante las pruebas, la temperatura presentó variaciones significativas a pesar del uso de la carcasa y una cobertura para el sol. Aunque esto reduce el control de las condiciones de la prueba, la situación permite obtener muestras con una mayor afectación por cambios térmicos mostrados para evaluar el compensador, esto es notable en el comportamiento de las curvas en la Ilustración 40.

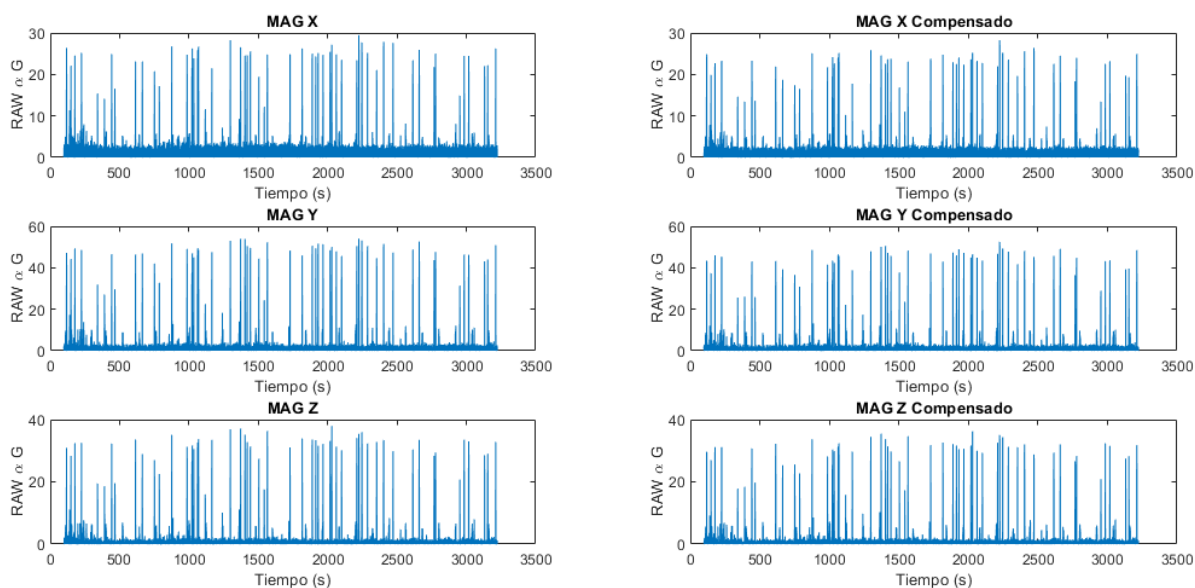
Ilustración 40. Datos de magnetómetro en prueba exterior.



Al comparar las gráficas resultantes, se observa que la envolvente de las curvas de la columna **a** al ser procesadas por el compensador, tienden hacia un comportamiento plano notable en la columna **b**. Teniendo en cuenta lo anterior, es posible afirmar que el compensador térmico funcionó y mantuvo la variación de la salida del sensor en  $\pm 10$  unidades sobre la media, mejora que fue posible con la linealización térmica planteada en la ecuación (5.1).

De las gráficas de la Ilustración 40 es evidente la presencia de distorsiones y por ello es necesario medirlo. Para separar el ruido de la señal se aplicó un proceso donde es calculada la tendencia de la señal empleando la función *smoothdata* de Matlab, luego la tendencia es restada de la señal original y calculado el valor absoluto de la operación, resultando en unas nuevas señales donde se ha extraído el ruido sin depender de la intensidad del campo magnético terrestre o el cambio proporcional de temperatura como se muestra en la Ilustración 41.

Ilustración 41. Ruido extraído en prueba exterior



Para cuantificar la cantidad de ruido presente en los datos extraídos se calcula la desviación estándar, esto ofrecerá un índice sobre la media del ruido presente en la muestra. En el caso particular de la prueba exterior los índices promedio de las tres sesiones de muestreo son consignadas en Tabla 13.

Tabla 13. Promedios de desviación estándar de ruido en prueba exterior.

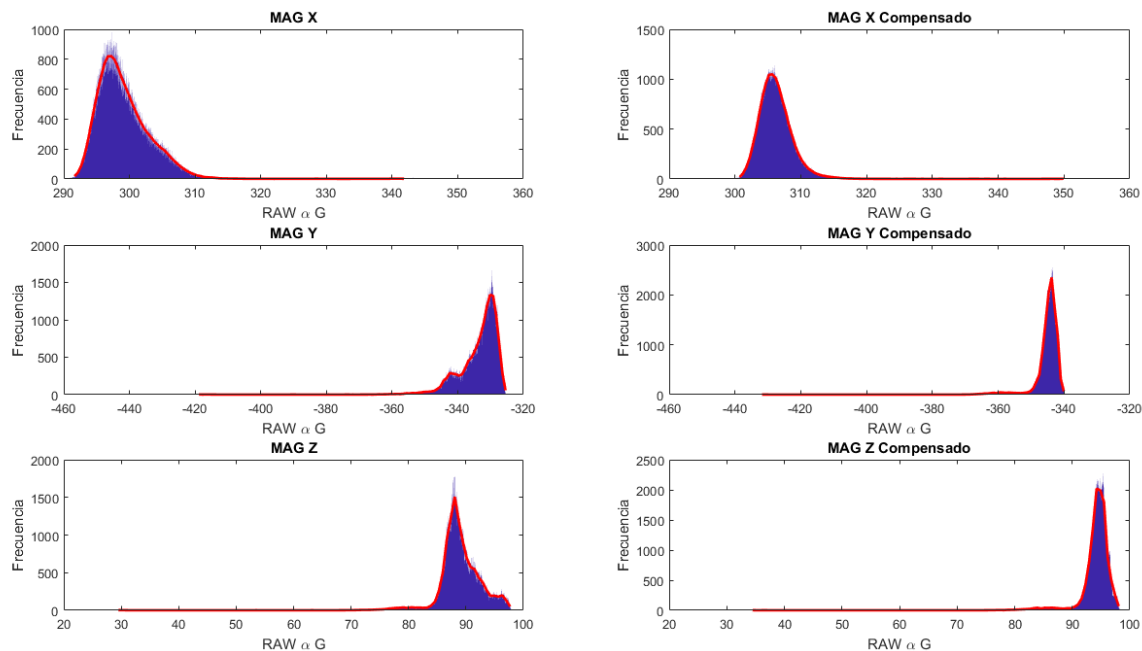
Eje	$\overline{SD}$	$\overline{SD}_{comp}$
<b>X</b>	2.0912	1.4139
<b>Y</b>	3.6393	2.7412
<b>Z</b>	2.3362	1.7334

**SD:** *Standard Deviation* – Desviación estándar



Como se observa en la tabla anterior los indicadores de desviación estándar se reducen al aplicar el compensador térmico, teniendo en cuenta que una menor desviación estándar significa una menor variación en la señal y en consecuencia menor ruido. De lo anterior es posible afirmar que conforme es aplicado el tratamiento de corrección térmica en la señal esta mejora significativamente, confirmando la efectividad del método en un caso donde la temperatura presentó un comportamiento irregular. Esto es apreciable en los histogramas en la Ilustración 42 generados con los datos del magnetómetro en la prueba.

Ilustración 42. Histograma de señales de magnetómetro en prueba exterior.



En todos los ejes es evidente que el tratamiento térmico concentra la energía de la señal cuando es aplicado.

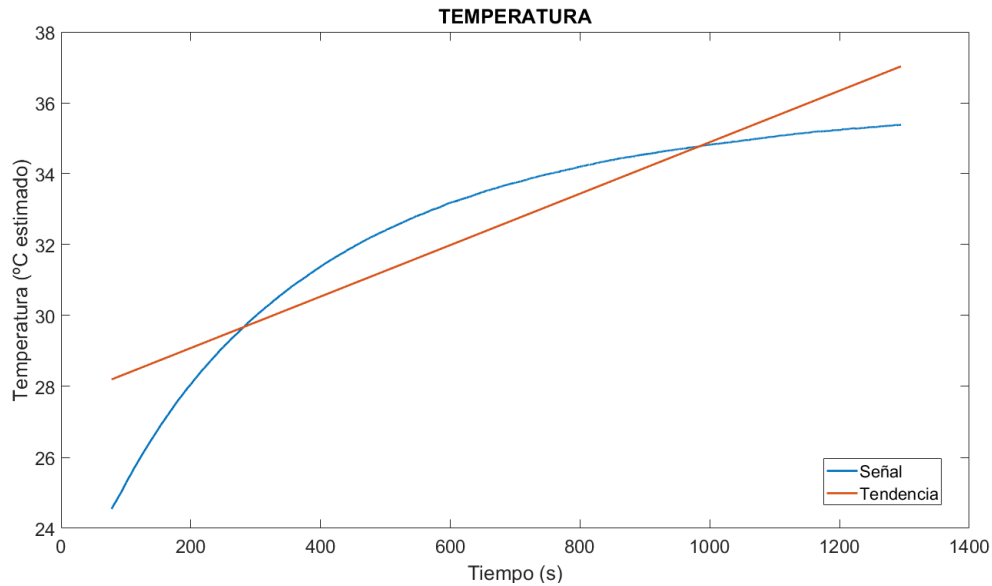
### 5.2.2 Prueba interior

Las mediciones se realizan con el mismo procedimiento general aplicado en la prueba exterior. Para ello, primero se ejecutó inicialmente la adquisición de factores de corrección de temperatura y posteriormente, sin apagar el dispositivo, medir el ruido ambiental dejando el nodo estático capturando datos, en este caso en una zona urbana con el potencial de contaminación de la señal producto de la actividad humana.

#### Análisis de la evaluación:

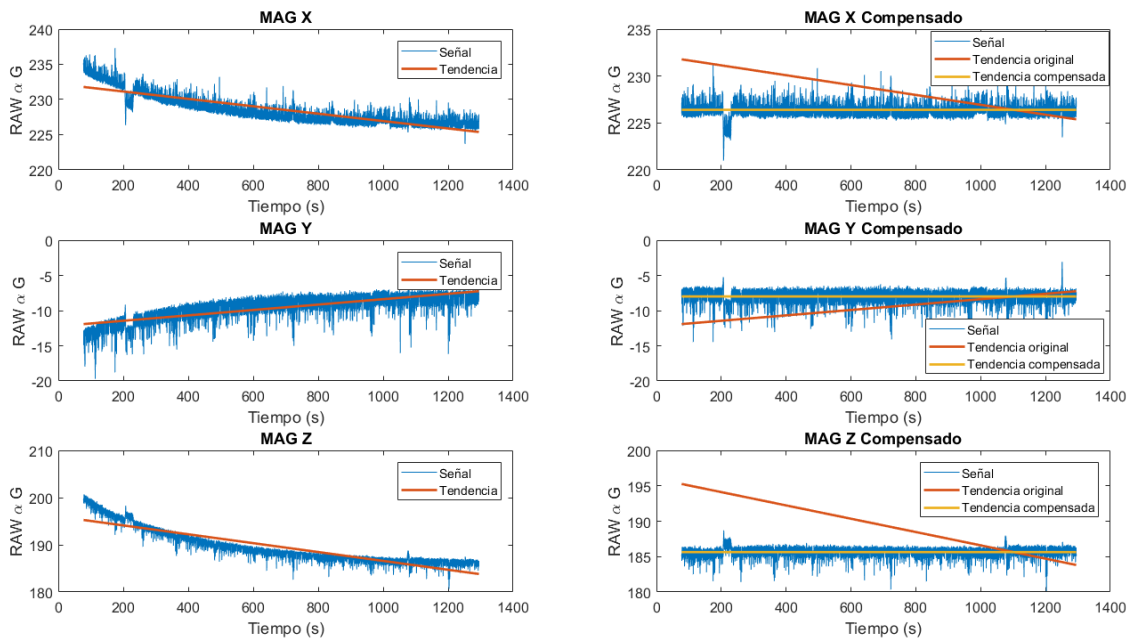
Para mantener la correspondencia con la prueba anterior, se realizaron tres mediciones sucesivas durante tres horas con un reinicio preventivo al microcontrolador cada hora. El comportamiento de cambio térmico, expuesto en la Ilustración 43, fue mucho más estable que en el exterior reduciendo el potencial efecto de ruido por temperatura.

Ilustración 43. Temperatura del nodo en prueba interior.



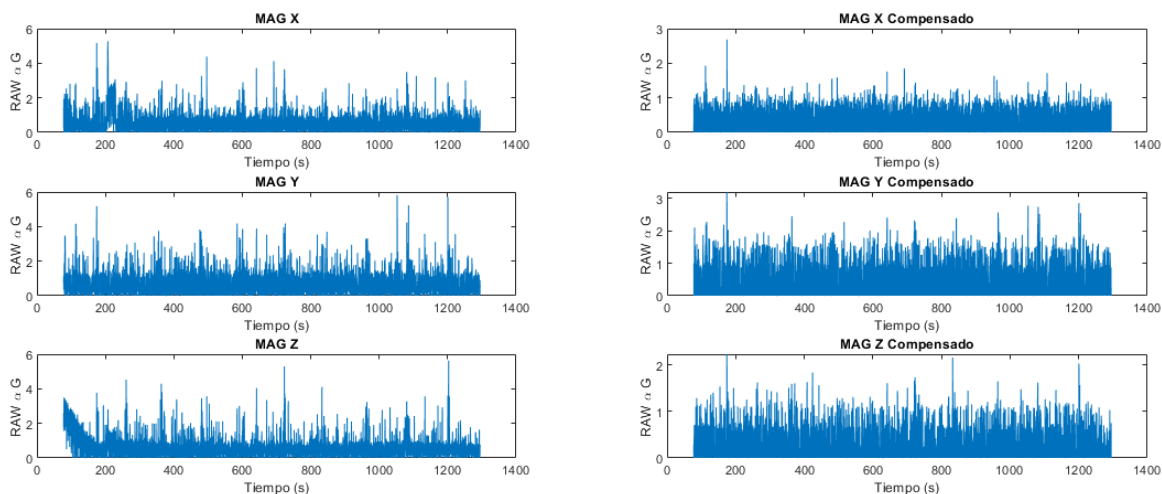
En cuanto la respuesta de los sensores, como era de esperarse en un entorno menos controlado, se detectaron eventos de ruido que son visibles como picos de señal en la Ilustración 44.

Ilustración 44. Datos del magnetómetro en prueba interior.



Nuevamente la compensación térmica aplicada en el microcontrolador redujo el efecto por el cambio de temperatura, pero no lo suprimió completamente, en la corrección térmica posterior en Matlab el efecto presenta un mejor comportamiento. Para efectos comparativos obtienen las señales de ruido de igual manera que en la prueba exterior como se muestra en la Ilustración 45

Ilustración 45. Ruido extraído en prueba interior.



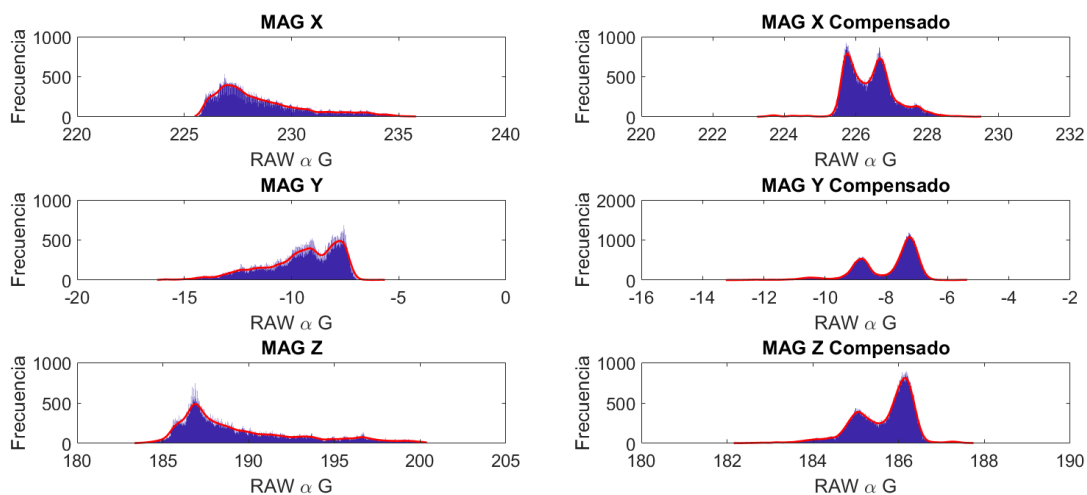
Con la información de las tres pruebas en interior, se obtienen los índices de señal a ruido promedio resumidos en la Tabla 14.

Tabla 14. Promedios de desviación estándar de ruido en prueba interior.

Eje	$\overline{SD}$	$\overline{SD}_{comp}$
<b>X</b>	0.7498	0.3064
<b>Y</b>	0.8525	0.4374
<b>Z</b>	0.8386	0.2969

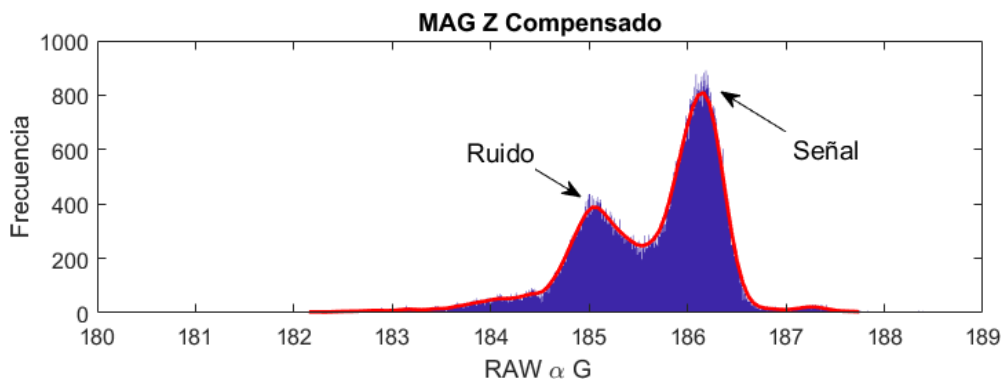
Los valores indicados en la tabla anterior muestran que tras la compensación térmica se reduce a más de la mitad la desviación estándar, lo que es muy significativo. Respecto a los histogramas presentados en la Ilustración 46, la concentración de la señal medida es apreciable a pesar de los eventos de ruido.

Ilustración 46. Histograma de señales de magnetómetro en prueba interior.



En la evaluación, son evidentes eventos de ruido que interfieren a la señal a medir, observables en detalle en la siguiente imagen:

Ilustración 47. Detalle de señales en el histograma en prueba interior.



### 5.2.3 Comparativa de resultados

Las dos pruebas suman seis horas de recolección de datos con los sensores, la transmisión de datos presentó una frecuencia promedio de 100 Hz por lo que se totalizan 2.16 millones de muestras en ese periodo de tiempo. La comparación de la prueba en el exterior e interior ofrece información sobre la incidencia de las emisiones electromagnéticas producidas por el hombre, los efectos de cambio de temperatura e incluso la interferencia por radiaciones solares. Como medida inicial se estimó la diferencia entre el ruido en el exterior e interior promediando los ejes del magnetómetro, los resultados de ese cálculo son los siguientes:

Tabla 15. Comparativa de desviaciones estándar de ruido entre prueba exterior e interior.

Ejes	$\overline{SD}$		$\overline{SD}_{comp}$	
	Exterior	Interior	Exterior	Interior
<b>X</b>	2.0912	1.4139	0.7498	0.3064
<b>Y</b>	3.6393	2.7412	0.8525	0.4374
<b>Z</b>	2.3362	1.7334	0.8386	0.2969
<b>Promedio</b>	2.6889	1.9628	0.8136	0.3469
<b>Relación (Exterior e Interior)</b>	<b>-27%</b>		<b>-57.36%</b>	

Se observa que el índice de ruido está a favor de la evaluación en el interior como un mejor ambiente de toma de muestras mostrando un 27% menos de ruido. Además la compensación térmica posterior mejora las condiciones de la señal llegando a un 57% menos de ruido que en la prueba interior, demostrando que a pesar de la falta de condiciones de control de temperatura en la prueba, es posible tratar los datos del magnetómetro para no depender del factor térmico.

#### Conclusión:

Las diferencias de ruido en los dos escenarios son consistentes a favor de las pruebas en el interior como un mejor sitio para evaluar al magnetómetro en búsqueda de los parámetros estáticos de este dispositivo, pero como se observó en la Ilustración 47, la presencia de eventos

de interferencia por la actividad del hombre es inevitable. En consecuencia, las siguientes pruebas realizadas con el magnetómetro requieren de ejecución bajo un par de consideraciones: primero, que exista un mínimo de certidumbre sobre las condiciones de emisiones electromagnéticas en las cercanías. Y segundo, una de temperatura estable, aunque no esté controlada.

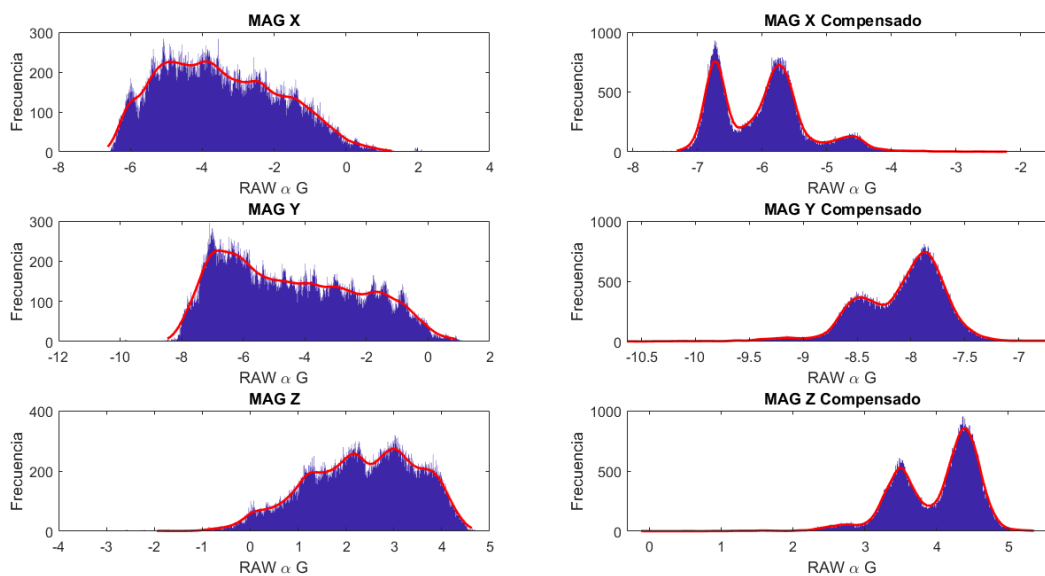
### 5.3 CÁLCULO DE PARÁMETROS ESTÁTICOS

Las comprobaciones anteriores han definido las condiciones necesarias para ejecutar una adecuada medición de señales. El procedimiento a seguir consiste en ejecutar la corrección de temperatura, tomar mediciones del magnetómetro y finalmente procesar la información obtenida, empleando algunos modelos de estimación de parámetros utilizados en las investigaciones consultadas en el estado del arte.

#### 5.3.1 Toma de muestras

Entre las 11 p.m. y las 3 a.m. es el horario seleccionado para tomar muestras con el magnetómetro, este periodo ha mostrado una escasa presencia de ruido en múltiples pruebas por la baja actividad humana y la estabilidad de temperatura en ese momento. Los histogramas mostrados en Ilustración 48 obtenidas de una de esas pruebas soporta la afirmación anterior, el rango de variación de la señal es menor a 8 unidades digitales del magnetómetro.

Ilustración 48. Histogramas de magnetómetro



En las evaluaciones previas se comprobó que un flujo de aire, incluso uno muy suave, cambia la temperatura de los sensores y en consecuencia afecta el proceso de cómputo de los factores de corrección térmica. Para prevenir este problema, se construyó un cubo en acrílico que contiene el nodo de sensores y un sistema de batería. La forma cúbica facilita la ubicación de los sensores cuando se realizan tomas de muestras sobre planos. El dispositivo construido se muestra en la Ilustración 49.

Ilustración 49. Cubo con sensores



Para iniciar la captura de datos, fue ejecutada una corrección térmica variando de 1 °C a 22 °C. Los parámetros *alpha* son calculados y aplicados con la asistencia de la herramienta Wimo, empleando el mismo algoritmo desarrollado en Matlab. Una vez programada la corrección de temperatura de los sensores, el nodo fue apagado y se tomaron muestras ubicándolo sobre la plataforma rotatoria al cubo tal como se aprecia en Ilustración 50.

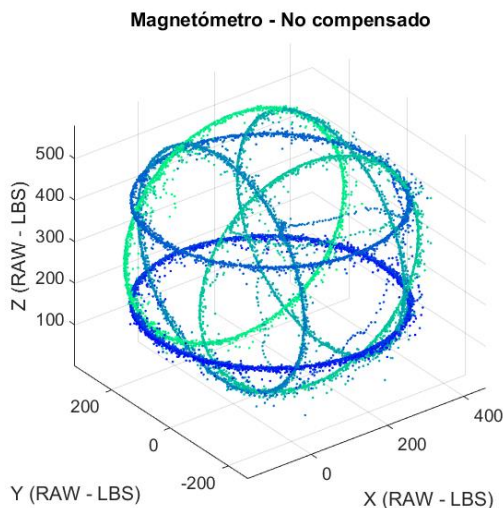
Ilustración 50. Cubo sobre plataforma



La plataforma se configuró a una velocidad de 8 RPM para asegurarse de no saturar al filtro o superar las capacidades de muestreo del sensor, respecto al filtro se ajustó para promediar 50 muestras. Los resultados de las mediciones fueron almacenados en un archivo CSV con la ayuda de la aplicación Wimo y posteriormente se graficó la información obtenida; para este caso se utilizó la comparativa de planos XY, XZ, YZ y una visualización 3D. La Ilustración 51 muestra el resultado de una de las sesiones de prueba, en el caso particular, el cubo fue rotado en cada una de sus caras mientras giraba sobre la plataforma.

---

Ilustración 51. Muestra de magnetómetro en plataforma



De la gráfica anterior se aprecia la formación de 6 anillos, esto se debe a que la línea de campo magnético detectado no es paralela al plano horizontal y al rotar sobre las caras del cubo el componente de campo en el eje Z desvía el anillo formado por las magnitudes correspondientes a los planos X, Y del sitio.

Según el WMM2015 las condiciones del sitio de prueba son:

Declinación:  $-5.4026^\circ$   
 Inclinación:  $25.2921^\circ$   
 Intensidad Horizontal:  $27033.1 \text{ nT}$   
 Intensidad Vertical:  $12773.9 \text{ nT}$   
 Intensidad Total:  $29899.18 \text{ nT}$

El sensor en las pruebas está configurado con una resolución digital de  $0.92 \text{ mG/LSB}$ , luego el radio ideal de los anillos es el siguiente:

$$B_H = 27033.1 \text{ nT} = 270.331 \text{ mG} \rightarrow B_{Hdigital} = \frac{270.331 \text{ mG}}{0.92 \text{ mG/LSB}} = 293.83 \text{ LSB} \quad (5.2)$$

Respecto a la separación de los anillos, la estimación es:

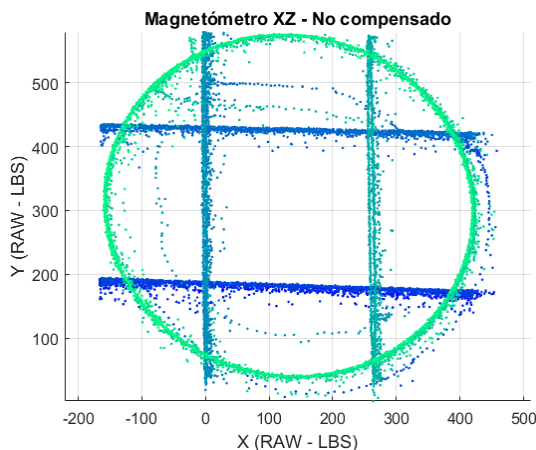
$$B_V = 12773.9 \text{ nT} = 127.739 \text{ mG} \rightarrow B_{Vdigital} = \frac{127.739 \text{ mG} * 2}{0.92 \text{ mG/LSB}} = 235.03 \text{ LSB} \quad (5.3)$$

El radio esperado de la esfera según el índice del WMM es el siguiente:

$$B_T = 29899.18 \text{ nT} = 298.991 \text{ mG} \rightarrow B_{Tdigital} = \frac{298.991 \text{ mG}}{0.92 \text{ mG/LSB}} = 324.99 \text{ LSB} \quad (5.4)$$

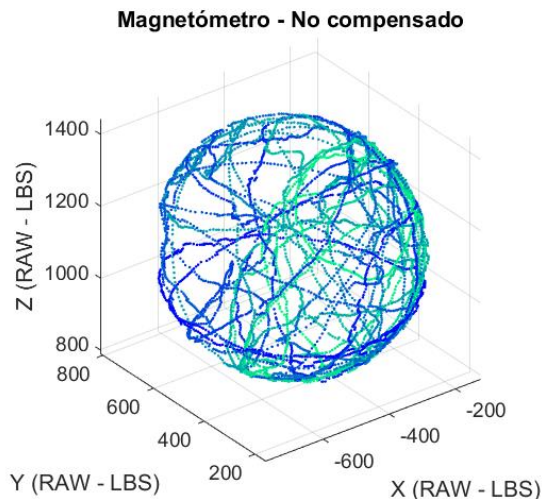
Las mediciones han indicado una aproximación al radio y separación de los anillos respecto a los valores calculados, no obstante, hay un efecto de corrimiento importante observado en la revisión de las gráficas en dos dimensiones, también es detectada una deformación de los anillos lo que indica la presencia de efectos *hard-iron* y *soft-iron* en el nodo, producto de los componentes internos y las imprecisiones propias del sensor como la construcción y la sensibilidad de cada eje. Una de las gráficas generadas en el plano XZ es ejemplo de lo mencionado anteriormente:

Ilustración 52. Detalle de plano XZ



La información adquirida sobre los planos ofrece la relación de corrección de escala y corrimiento en cada eje del magnetómetro. Algunas publicaciones consultadas recomiendan el uso de datos con posiciones aleatorias, según los autores esto mejora los resultados en los cálculos de parámetros de corrección. Por esta razón se toman muestras bajo los dos modos con el fin de comparar resultados.

Ilustración 53. Muestreo aleatorio de magnetómetro



La esfera no es perfecta y en el detalle del plano XZ persiste la deformación consistente con lo observado en la Ilustración 52.

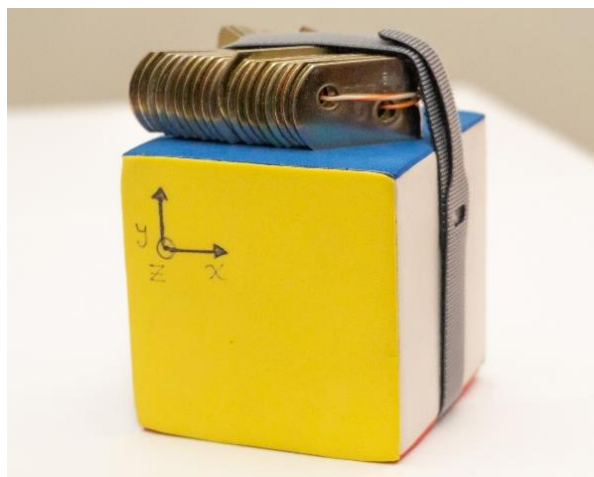
---



### Muestras de robustez

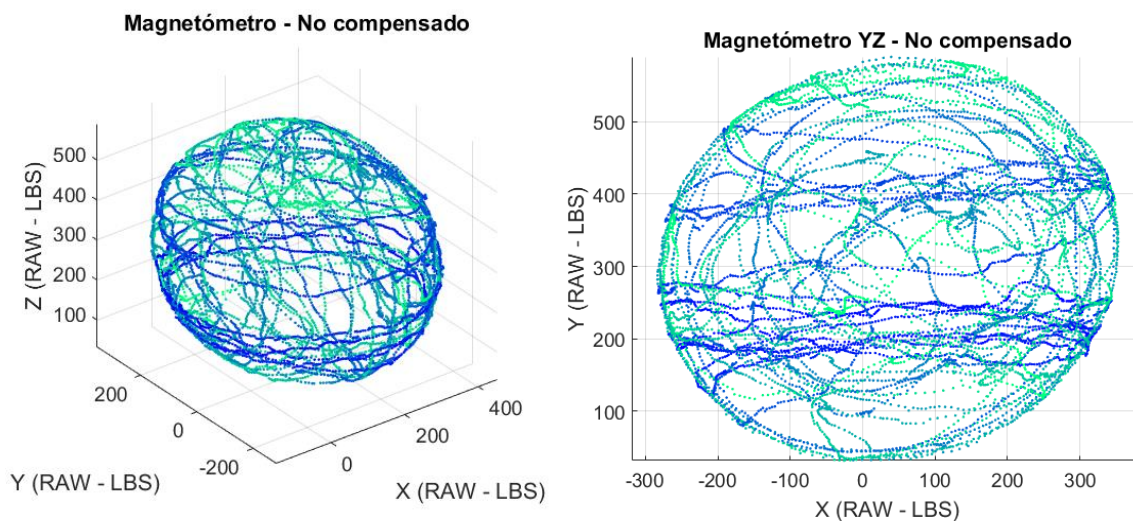
Aunque en los nodos de prueba presentan efectos *hard-iron* y *soft-iron*, no son lo suficientemente fuertes para comprobar la efectividad del método de estimación de parámetros de corrección, por esta razón se repiten las mediciones anteriores adicionando una masa de metal junto al sensor que provoca efectos de distorsión mayores.

Ilustración 54. Cubo de sensores con masa metálica



En las mediciones iniciales sin compensación de corrimiento o ganancia, los efectos *hard-iron* y *soft-iron* son notorios tal como se muestra en las siguientes gráficas:

Ilustración 55. Respuesta del sensor a masa metálica



Con las consideraciones de medición previas se realizan las pruebas de los procesos de cálculo para los parámetros de compensación.

### 5.3.2 Compensación directa de corrimiento y ganancia

El modelo de compensación aplicado en este caso parte de la propuesta de Yadav y Bleakeley [53], que consiste en un modo de cálculo simple. Utilizando los mismos factores implementados para compensar ganancia y corrimiento en el hardware del nodo (ver pág. 42), con un costo computacional bajo debido a su operación directa sobre cada eje del sensor. La intención de esta validación se debe a la presencia de compensadores del mismo tipo en el hardware de algunos sensores comerciales, estos operan restando un valor para ajustar el corrimiento y multiplicando por un segundo valor para corregir la ganancia.

El procedimiento para calcular los valores de compensación es el siguiente:

1) Calcular la intensidad de campo magnético horizontal según el WMM y emplearlo para ajustar la escala de los ejes XY de una muestra de rotación horizontal sobre estos dos ejes mencionados aplicando la siguiente ecuación:

$$\vartheta_{gain} = \frac{B_H}{\frac{\vartheta_{max} - \vartheta_{min}}{2} * \delta * 100} \quad (5.5)$$

Donde  $B_H$  es la magnitud de campo magnético horizontal del WMM,  $\vartheta$  es el eje del magnetómetro a evaluar,  $\delta$  responde la resolución digital del magnetómetro que por defecto es 0.92 mG/LSB. Este criterio se aplica con los máximos y mínimos detectados con los ejes X, Y que ofrecerán un valor de escala (diámetro del elipsoide) en cada eje para luego corregirlo con un factor a su valor ideal.

2) Con una muestra de rotación sobre el plano YZ, se determinan los máximos y mínimos del eje Z, luego es aplicada la ecuación (5.5) para estimar la ganancia del eje Z.

3) La información de rotación libre se utiliza para determinar los valores máximos y mínimos de cada eje, luego se calculan los factores de corrimiento para centrar el elipsoide aplicando la siguiente ecuación:

$$\vartheta_{offset} = \frac{\vartheta_{max} - \vartheta_{min}}{2} - \vartheta_{min} \quad (5.6)$$

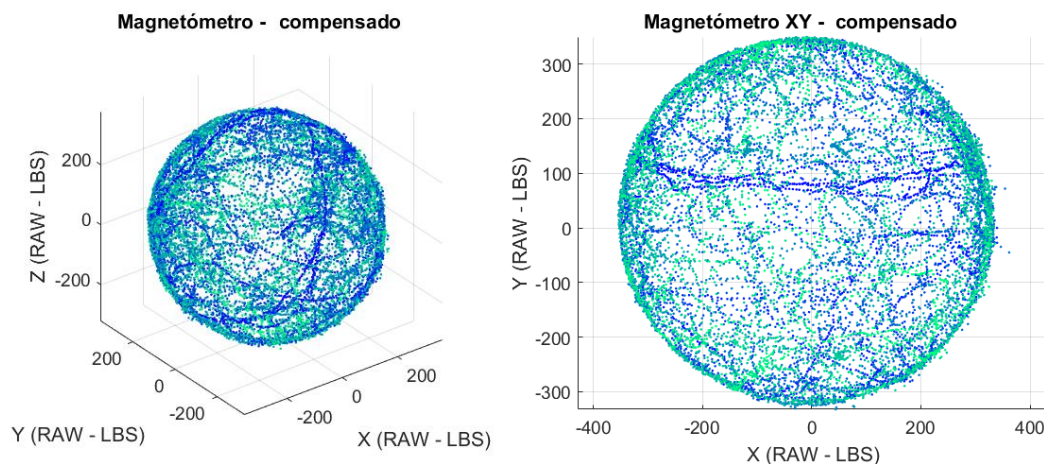
Los parámetros bajo este esquema se pueden calcular a través de Matlab o la aplicación Wimo que tiene implementado el procedimiento, para el caso particular se empleó la segunda opción.

Luego de obtener los parámetros de compensación *offset* y *gain*, son programados en el nodo y es realizada una toma de muestras aleatorias para comparar la corrección con las muestras iniciales, los resultados son mostrados en la Ilustración 56.

La escala de la esfera coincide con el radio calculado en la ecuación (5.4) de 324.99 unidades, el centroide del elipsoide graficado se encuentra muy cercano al centro. Al presentar una esfera con tan poca deformación en resultado, señala que no hay una incidencia fuerte de una no ortogonalidad en los ejes del sensor.

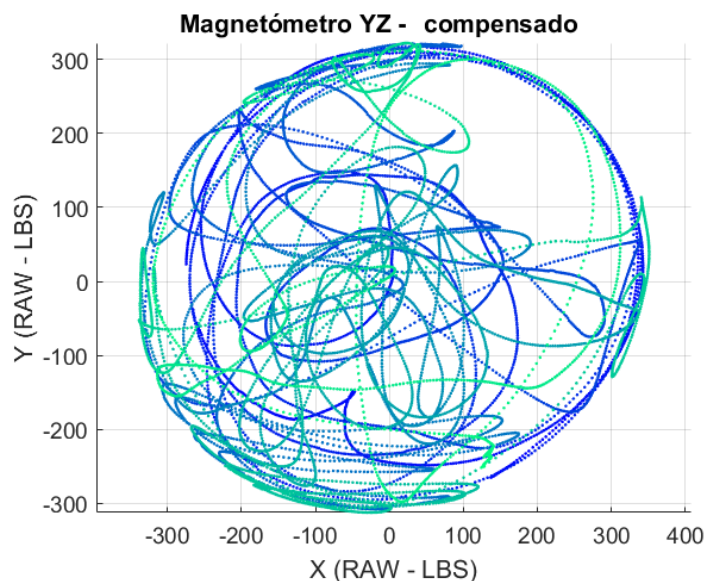
---

Ilustración 56. Muestra aleatoria con la compensación



Luego de esta verificación se repite el procedimiento adicionando la masa de metal al sensor, dando como resultado lo siguiente:

Ilustración 57. Muestra en plano YZ con masa de metal



La adición de la masa de metal permite confirmar que el método de compensación con los factores de corrimiento y ganancia no son suficientes para corregir la distorsión producida.

Recordando el modelo matemático simplificado:

$$h_{m,t} = Ch^b + b + n \quad (5.7)$$

El ruido  $n$  es compensado con el filtro de ventana implementado en el microcontrolador,  $b$  integra el factor de corrimiento producto del efecto *hard-iron* y la desalineación del sensor. Dentro de  $C$ , la matriz rotacional  $C_\eta$  se asume como una matriz unitaria debido la instalación paralela del sensor respecto al cuerpo del dispositivo (el cubo) y porque que no es posible precizarla con los recursos disponibles, la matriz  $C_S$  responde a los factores de ganancia calculados, pero no hay factores que cumplan con la matriz  $C_\zeta$ :

$$C = C_S C_\eta C_\zeta = \begin{bmatrix} 1 + S_x & 0 & 0 \\ 0 & 1 + S_y & 0 \\ 0 & 0 & 1 + S_z \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 + \zeta_{xx} & \zeta_{xy} & \zeta_{xz} \\ \zeta_{yx} & 1 + \zeta_{yy} & \zeta_{yz} \\ \zeta_{zx} & \zeta_{zy} & 1 + \zeta_{zz} \end{bmatrix} \quad (5.8)$$

En cuanto a cómo se aplican los factores de corrección calculados en el firmware del nodo:

```

if (gain_compensation) {
    mag.x *= GAIN[6];
    mag.y *= GAIN[7];
    mag.z *= GAIN[8];
}

if (offset_compensation) {
    mag.x -= OFFSETS[6];
    mag.y -= OFFSETS[7];
    mag.z -= OFFSETS[8];
}

```

La compensación de corrimiento/*hard-iron* (*offset*) y ganancia (*gain*) por el hardware es correcta según el modelo matemático, pero la aplicación de la matriz  $C_\zeta$  no estaría completa. Esto explica por qué la compensación no elimina la deformación del elipsoide producto del efecto *soft-iron*.

### 5.3.3 Adecuación a una esfera por mínimos cuadrados

Es un método idóneo en términos de precisión para obtener los parámetros estáticos según los autores consultados. Es repetitivo el uso de los algoritmos como el de Kanatani [67] o Gebre-Egziabher [54] que parten de la definición matemática de un elipsoide:

$$aX^2 + bY^2 + cZ^2 + f2XY + g2XZ + h2YZ + p2X + q2Y + r2Z + d = 0 \quad (5.9)$$

Donde X, Y, Z son los ejes del sensor y las letras restantes conforman un vector de coeficientes:

$$v = [a, b, c, f, g, h, p, q, r, d]^T \quad (5.10)$$

Con una nube de puntos como las que son generadas en las muestras aleatorias, son calculadas las ecuaciones para cada punto aplicando la ecuación (5.9), dependiendo del método de solución

del elipsoide es posible que adicione algunos factores a esa ecuación o no. Luego, es aplicado un método de solución numérico para despejar los factores del vector  $v$ .

Para reducir las posibilidades de error por mala implementación de un algoritmo, se han seleccionado cuatro scripts de Matlab publicados que permitan calcular los parámetros de ajuste a una esfera, dichos algoritmos son:

### 5.3.3.1 Ajuste específico a una elipsoide por mínimos cuadrados (*Least Squares Ellipsoid Specific Fitting*) – Li/Griffiths [68]

Este método emplea tres ecuaciones auxiliares generadas de la definición (5.9) que son:

$$I = a + b + c \quad (5.11)$$

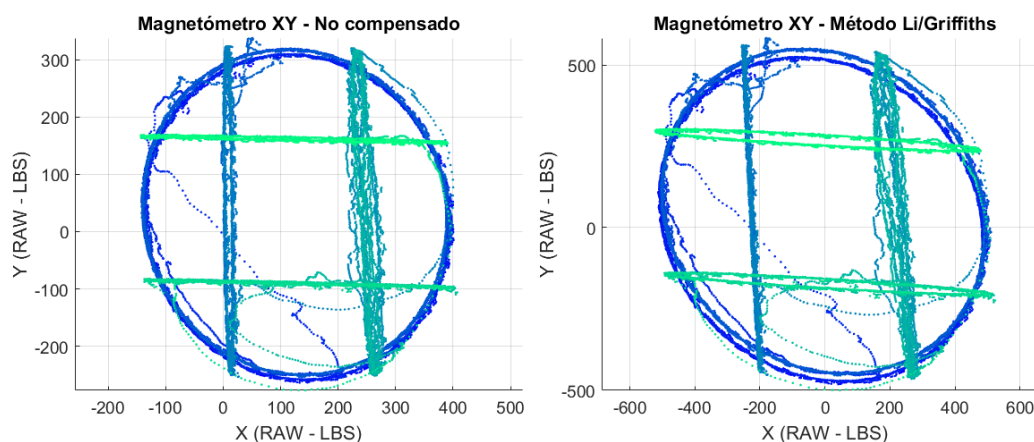
$$J = ab + bc + ac - f^2 - g^2 - h^2 \quad (5.12)$$

$$K = \begin{bmatrix} a & h & g \\ h & b & f \\ g & f & c \end{bmatrix} \quad (5.13)$$

Los autores exponen unas propiedades de  $kJ - I^2$  que según ellos permite hallar una solución a la matriz de parámetros de un elipsoide, utilizando una nube de puntos de pocas muestras y un factor K que define la forma del elipsoide.

Este algoritmo no entregó resultados al emplear muestras de rotación aleatoria del magnetómetro debido a indeterminaciones matemáticas, únicamente fue posible evaluar este algoritmo con la nube de puntos generada en la prueba de giro del sensor sobre sus ejes ortogonales con la masa de metal.

Ilustración 58. Corrección Li/Griffiths en plano XY



La matriz de corrección calculada con este método tiene un efecto negativo sobre la información original, los anillos presentan una desalineación mayor que en la medición sin procesar y no corrige la forma elíptica de los anillos del plano XY.

### 5.3.3.2 Ajuste a elipsoide o esfera para calibración de sensores (*Ellipsoid or sphere fitting for sensor calibration*) – Vitali (ST Microelectronics) [69]

Es un algoritmo de ajuste a esfera propuesto por ST Microelectronics para procesar la información de sensores como acelerómetros y magnetómetros. Para una elipse rotada se calcula una matriz  $9 \times n$ :

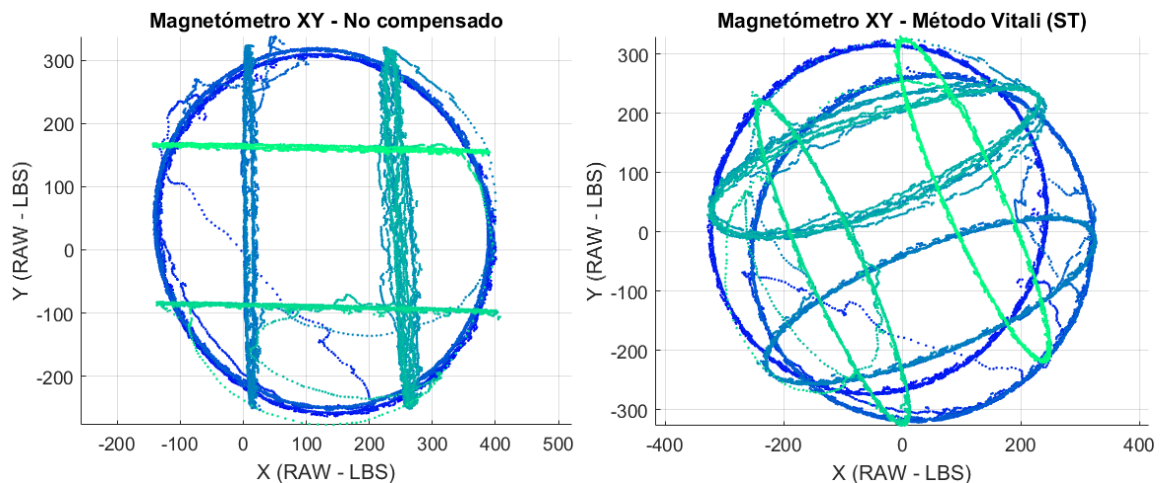
$$D = [x^2 \ y^2 \ z^2 \ 2xy \ 2xz \ 2yz \ 2x \ 2y \ 2z] \quad (5.14)$$

$$v = \frac{D^T * D}{D^T * \text{ones}(n, 1)} \quad (5.15)$$

De estas ecuaciones,  $v$  calcula los mínimos cuadrados de la matriz  $D$  generada, luego los factores generados se organizan para calcular los parámetros dependiendo del tipo de muestra, debido a que el algoritmo soporta corrección con el elipsoide en rotación, sin rotación o ajuste sobre planos 2D.

Para el caso particular se evalúa con el primer tipo de corrección, el resultado de la nube de puntos con el magnetómetro es el siguiente:

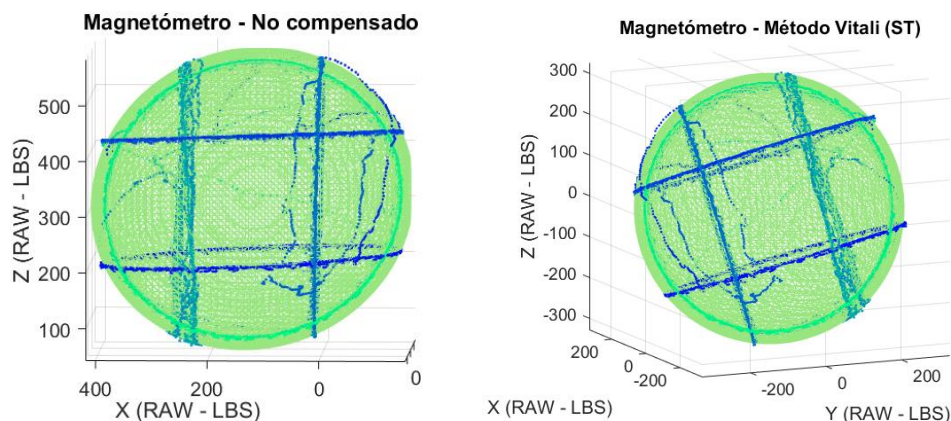
Ilustración 59. Corrección Vitali en plano XY



En los anillos del plano XY es notoria una corrección en la forma presentando una forma circular mejor definida, sin embargo, no se encuentran alineados lo que señala un desplazamiento rotacional de la esfera compensada, esto se confirma con los gráficos tridimensionales expuestos en la Ilustración 60.

La esfera muestra un desplazamiento rotación y además la esfera se muestra como un espejo en el plano Z, a favor de este método la esfera tiene forma esférica correcta y su centroide está cercano al origen del plano.

Ilustración 60. Corrección Vitali en 3D



### 5.3.3.3 Calibración escalar de magnetómetros vectoriales (*Scalar calibration of vector magnetometers*) – Merayo [70]

Algoritmo pre-vuelo evaluado en la calibración de satélites europeos como Ørsted, Astrid-2 o CHAMP. En este método se define una ecuación:

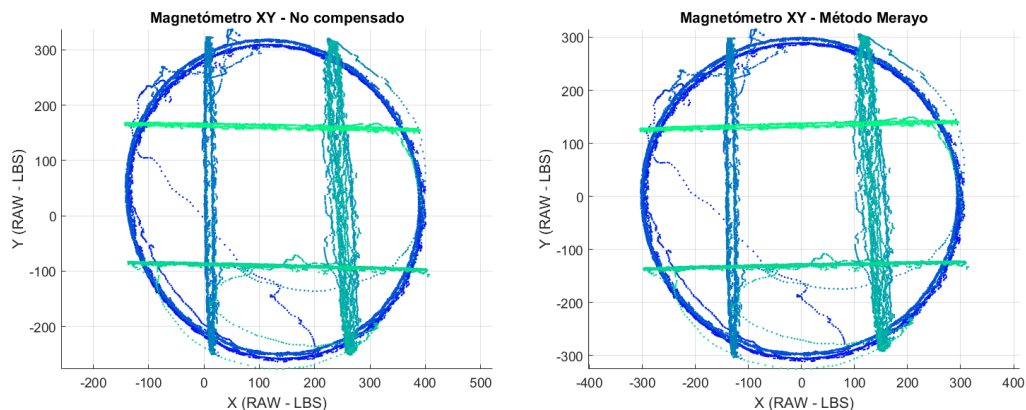
$$D = [x^2 \ y^2 \ z^2 \ xy \ xz \ yz \ x \ y \ z \ 1] \quad (5.16)$$

Luego, para calcular los mínimos cuadrados en Matlab:

```
D = qr(D); %Descomposición ortogonal-triangular
D = triu(D); %Selección de parte triangular superior de una matriz
[U, S, V] = svd(D); %Descomposición de valor singular
```

Con este tratamiento se calculan los parámetros base para generar la matriz de rotación, corrimiento y ganancia. Al aplicar este algoritmo con las muestras de anillos se tiene como resultado:

Ilustración 61. Corrección Merayo en XY

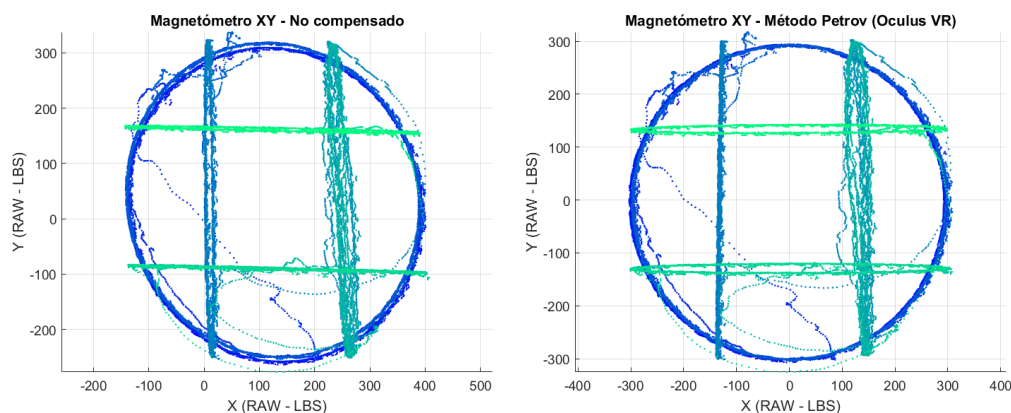


Los anillos tras la corrección con este método resultan satisfactorios, la proporción de los anillos es correcta, conserva la posición de la esfera, y está centrada en el origen del plano. De los parámetros de corrección que retorna Merayo este permite obtener la matriz de compensación  $C$  y el vector de centrado  $b$ .

### 5.3.3.4 Ajuste a elipsoide (*Ellipsoid fit*) -Petrov [68]

Es un algoritmo publicado en el repositorio de Matlab desarrollado por Yury Petrov. No se ha podido relacionar este algoritmo con una publicación, pero al comparar la corrección en la Ilustración 62 y del algoritmo Merayo en la Ilustración 61 se observa una similitud muy alta en los resultados. Del análisis del código de Petrov, resulta en un proceso más sencillo y computacionalmente de menor impacto. Al comparar los anillos respecto a los ejes es notoria una alineación de estos en comparación a Merayo, además la función retorna los parámetros necesarios para calcular de forma independiente la matriz de ganancia y la matriz de deformación por efecto *soft-iron*.

Ilustración 62. Corrección Petrov XY



### 5.3.4 Resultados

De los algoritmos de corrección evaluados se ha seleccionado el método Petrov, no solo por sus resultados, sino por la posibilidad de separar la matriz de ganancia de la matriz de corrección *soft-iron*/rotacional en los factores de salida de este algoritmo. Para estas evaluaciones se programó el script *ellipsoids.m* para Matlab, con este código es posible evaluar los cuatro algoritmos de ajuste a elipsoide y además calcula los parámetros de corrección para el magnetómetro.

La función *ellipsoid\_petrov* retorna cinco variables, tres de estas son las necesarias para los cálculos:

- $e\_center$  = vector  $(x,y,z)$  de corrimiento al origen del plano
- $e\_radii$  = vector  $(x,y,z)$  de tamaño de radio
- $e\_eigenvecs$  = matriz  $3 \times 3$  con vectores propios de los radios  $e\_radii$



Para aplicar la corrección sobre una muestra del magnetómetro contenida en una matriz 3 x n, donde n representa la cantidad de muestras, primero se ajusta el elipsoide al centro:

$$S = \begin{bmatrix} x_0 & \cdots & x_n \\ y_0 & \cdots & y_n \\ z_0 & \cdots & z_n \end{bmatrix} - \begin{bmatrix} e_{center_x} \\ e_{center_y} \\ e_{center_z} \end{bmatrix} \quad (5.17)$$

En cuanto a la matriz diagonal de escala:

$$C_{s1} = \begin{bmatrix} e_{radii_x} & 0 & 0 \\ 0 & e_{radii_y} & 0 \\ 0 & 0 & e_{radii_z} \end{bmatrix}^{-1} * \min(e_{radii}) \quad (5.18)$$

Previo al cálculo de la matriz de compensación C se generan dos matrices con los vectores propios:

$$I_{3x3} = e\_eigenvecs ; M_{3x3} = e\_eigenvecs^T \quad (5.19)$$

Luego la matriz de compensación C se obtiene a partir las ecuaciones (5.18) y (5.19):

$$C = I * C_{s1} * M \quad (5.20)$$

Los datos del magnetómetro ajustados a una esfera se obtienen utilizando las ecuaciones (5.17) y (5.20)

$$S_c = C * S \quad (5.21)$$

En este punto, la esfera está ajustada a un radio estimado por el algoritmo, pero no asegura la corrección de ganancia conforme a la magnitud de campo magnético que se esperaría del WMM con la escala configurada en el magnetómetro, entonces la matriz de ganancia corregida es:

$$C_s = C_{s1} * \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} = \begin{bmatrix} 1 + S_x & 0 & 0 \\ 0 & 1 + S_y & 0 \\ 0 & 0 & 1 + S_z \end{bmatrix} \quad (5.22)$$

Donde  $\lambda$  responde al resultado de dividir el valor de referencia de campo magnético y la magnitud de radio medio de la esfera corregida en la (5.21) aplicando lo siguiente:

$$\lambda = \frac{B_T (mG)}{RDS} * mediana \left( \sqrt{S_{cx}^2 + S_{cy}^2 + S_{cz}^2} \right) \quad (5.23)$$

Donde  $B_T$  corresponde a la magnitud de campo magnético total según el WMM,  $RDS$  es la resolución digital del sensor y  $S_c$  la matriz generada en (5.21).

Como el sensor está instalado de forma paralela a los planos del cubo exterior y la inspección de las gráficas no indica diferencias significativas de no ortogonalidad de los ejes, es posible asumir

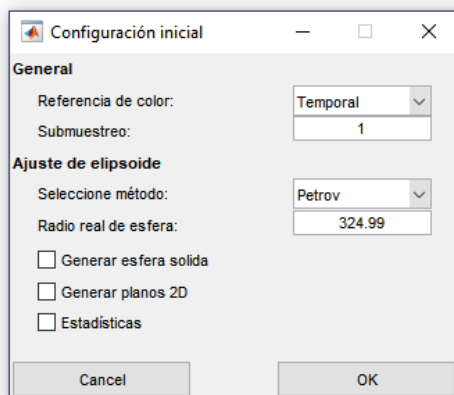
que la matriz rotacional  $C_\eta$  es diagonal unitaria, luego se despeja la matriz de factores *soft-iron*  $C_\zeta$  gracias a que la matriz de ganancia  $C_{s1}$  también es diagonal, resultando:

$$C_\zeta = C_{s1}^{-1} * C = \begin{bmatrix} 1 + \zeta_{xx} & \zeta_{xy} & \zeta_{xz} \\ \zeta_{yx} & 1 + \zeta_{yy} & \zeta_{yz} \\ \zeta_{zx} & \zeta_{zy} & 1 + \zeta_{zz} \end{bmatrix} \quad (5.24)$$

Respecto del vector *hard-iron* no es posible estimarlo, debido a que estos valores son sumados con el vector de corrimiento del sensor, Por lo tanto, para estimar las diferencias entre estos datos es necesario contar con un equipo de laboratorio que permita aislar al sensor de la fuente de efecto *hard-iron*. En consecuencia, el vector formado por la variable  $e\_center$  responde a la suma del vector de corrimiento (*offset*) del sensor y vector de efecto *hard-iron* que en el modelo matemático simplificado está representado por  $b$ .

Cuando se ejecuta el script *ellipsoids* en Matlab primero solicita unos valores de configuración a través de una ventana:

Ilustración 63. Configuración inicial *ellipsoids*



La referencia de color está disponible con base en el tiempo o la temperatura asociada a cada muestra, esto puede ser útil en especial si se quiere determinar un efecto de alteración por temperatura durante la prueba. La opción de submuestreo permite escalar el vector de muestras del archivo CSV para reducir el tiempo de cálculo en el caso de emplear datos voluminosos.

En los parámetros asociados al ajuste de elipsoide están disponibles las opciones para seleccionar el método de ajuste conforme los ya citados en la sección anterior, el radio de la esfera que corresponde al valor producido por  $\frac{B_T}{RDS}$  mencionado en la ecuación (5.23). Además, para facilitar la observación de las muestras es posible generar un elipsoide sólido con los puntos medidos o calculados y gráficos de planos en dos dimensiones. La opción de estadísticas ofrece un análisis sobre la media del radio y la desviación estándar de éste y las entrega en la línea de comandos de Matlab.

---

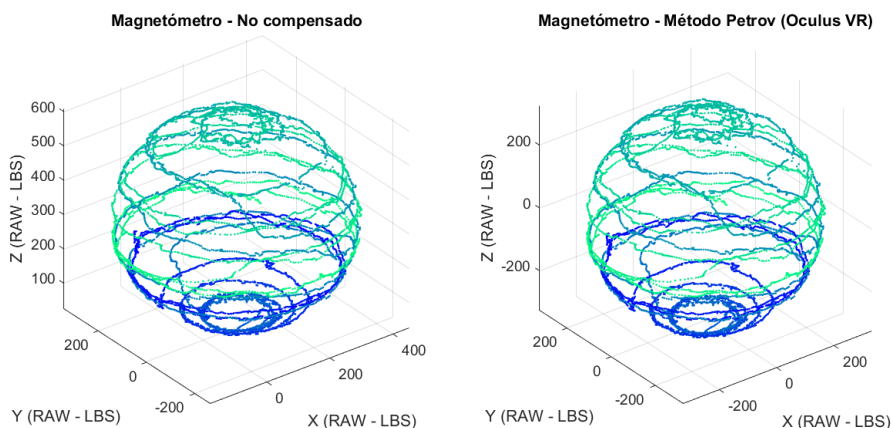
En la ventana de comandos de Matlab se presentan los factores calculados en la muestra, este es un ejemplo producto de una sesión en la que el cubo fue rotado sobre el eje Y:

```
*****
Fecha de la muestra:      05 Jun 2018  |  21:15:02 GMT
***** Parámetros - Petrov *****
Radio de la esfera Petrov:
    290.0522
Radio de la esfera real:
    324.9900
Vector de corrimiento + hard-iron:
    137.2564
     39.9224
    315.2397
Matriz de compensación C:
    1.0339    0.0063    0.0206
    0.0063    1.0657   -0.0035
    0.0206   -0.0035    1.1155
Matriz de escala:
    1.0278         0         0
         0    1.0669         0
         0         0    1.1205
Vector de factores S:
    0.0278
    0.0669
    0.1205
Matriz soft-iron:
    1.0060    0.0061    0.0200
    0.0059    0.9989   -0.0032
    0.0184   -0.0031    0.9956
```

### Validación de la corrección

Las gráficas generadas a partir de los datos de la prueba sin compensación y con corrección aplicando los parámetros obtenidos por el método de Petrov son las siguientes:

Ilustración 64. Gráficas 3D generadas



Finalmente son evaluados los algoritmos de ajuste a una esfera considerando como referencia la desviación estándar ( $\sigma$ ) de los  $n$  radios generados con cada punto de la muestra empleada sin compensar. Luego, la división entre las desviaciones estándar de los radios de la esfera no compensada y la compensada ofrecerá una relación porcentual de mejora sobre la esfera.

$$\text{Factor de mejora}(\%) = \frac{\sigma_{no\ compensada}}{\sigma_{compensada}} * 100 \quad (5.25)$$

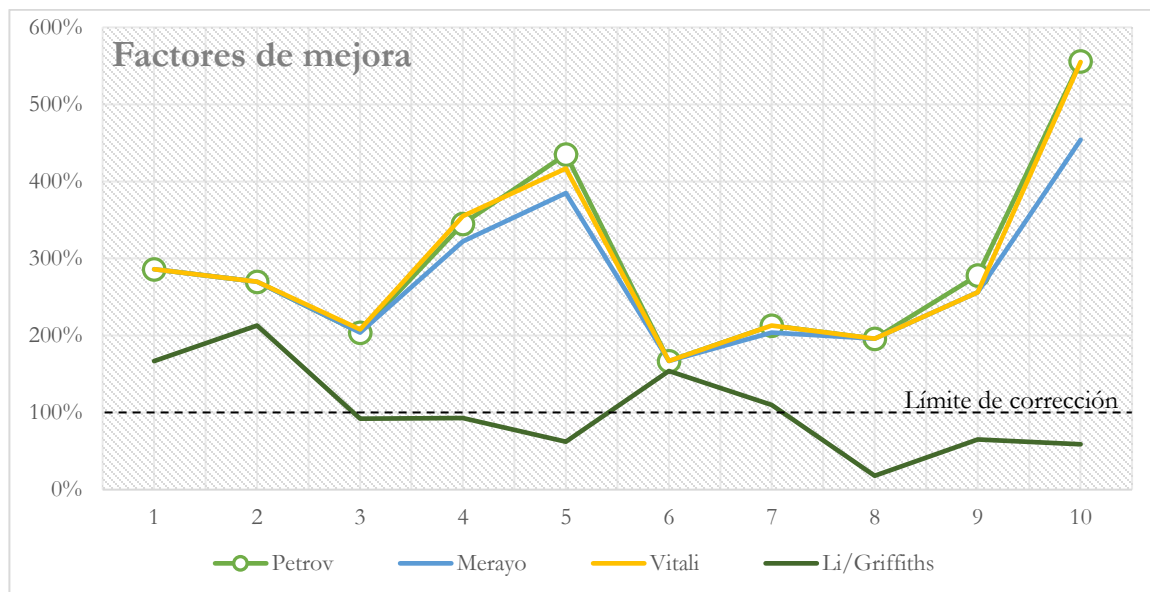
Cabe aclarar que hay efectos como errores de muestreo, ruido y la variación térmica que pueden afectar el valor de desviación estándar aún con una esfera de calidad aceptable tras la aplicación de los parámetros de corrección. La siguiente tabla muestra el rendimiento porcentual de los algoritmos en diez pruebas de rotación libre realizadas:

Tabla 16. Factores de mejora en evaluaciones con los algoritmos de corrección

Muestra	Petrov	Merayo	Vitali	Li/Griffiths
1	286%	286%	286%	167%
2	270%	270%	270%	213%
3	204%	204%	208%	92%
4	345%	322%	355%	93%
5	435%	385%	417%	62%
6	167%	167%	167%	154%
7	213%	204%	213%	110%
8	196%	196%	196%	18%
9	278%	256%	256%	65%
10	555%	454%	555%	59%

Según la definición del factor de mejora, aquellos valores menores a 100% indican potenciales deformaciones a la esfera original, 100% que la esfera no presentó cambios y superiores a 100% señalan mejoras en la forma de la esfera. Para facilitar el análisis de la Tabla 16 se genera una gráfica mostrada en Ilustración 65.

Ilustración 65. Gráfico comparativo de algoritmos



Se observa que los algoritmos Petrov, Merayo y Vitali tienen un rendimiento similar, encontrándose Petrov ligeramente por encima, lo que indica una mejora en comparación a Merayo y Vitali. Por su parte Li/Griffiths muestra problemas al tener factores de corrección menores al 100%, indicando que esas muestras empeoraron en su forma cuando fue aplicado este método de ajuste a esfera.

Esta comparativa confirma que este algoritmo de Petrov es el mejor de la selección y que los parámetros obtenidos con este método tienen una confiabilidad aceptable para corregir la señal de un magnetómetro de tres ejes.

En el anexo F – *Script de cálculo de parámetros "ELLIPSOIDS.M"* se encuentran más detalles del script *ellipsoids.m* además de un instructivo de uso.

## 6 CONCLUSIONES

---

### 6.1 RESPECTO A LOS OBJETIVOS PLANTEADOS

*Definir el modelo matemático de referencia y los parámetros estáticos asociados a este.*

- El estado del arte permitió establecer un modelo con facilidad debido a que muchos autores de artículos han recurrido a una ecuación ya probada con éxito en diferentes contextos, incluyendo sistemas de navegación para satélites.
- En varias publicaciones se evidenció que los parámetros estáticos no son tomados como una prioridad en un proceso de mejora o calibración de medición de un magnetómetro. Generalmente recurren a métodos experimentales para obtener un vector de corrección de corrimiento y de efecto *hard-iron*. El resto de correcciones quedan a cargo de un filtro Kalman, obligando a un esfuerzo computacional permanente para que dicho filtro funcione correctamente.
- El factor de temperatura no es tenido en cuenta dentro del modelo del magnetómetro por ninguno de los autores, aunque en contados casos se encontró que era considerado al momento de realizar experimentos bajo condiciones de control.

*Plantear los mecanismos necesarios para estimar los parámetros estáticos del modelo seleccionado.*

- Uno de los grandes retos de la investigación estaba en obtener datos que permitieran la evaluación del magnetómetro con las limitantes de recursos de laboratorio. Gracias a las consideraciones sobre el proceso de medición mostradas en el capítulo 3 este problema fue posible de solucionar.
- El entendimiento sobre la operación de un magnetómetro AMR resultó útil al ofrecer información sobre sus ventajas y desventajas, en especial la respuesta en función de la temperatura, que se habría convertido en un inconveniente en el proceso de validación de no ser por la previa implementación de un compensador.
- La arquitectura del sistema definida resultó en un mecanismo exitoso gracias a la estabilidad que ofreció, ya que fue posible recolectar masivamente una gran cantidad de datos con pocas pérdidas. El hardware se integró sin dificultad y la conectividad inalámbrica facilitó la ubicación del sensor en cualquier lugar para distintas pruebas.
- Incluir en el proceso una herramienta para la captura de datos permitió ejecutar dos experimentos simultáneos a fin de obtener muestras comparativas. Estos datos fueron referentes para determinar fuentes de ruido o distorsión.

*Evaluar el sistema propuesto en un entorno controlado con la asistencia de herramientas computacionales*

- El término “*entorno controlado*” resultó relativo tras las primeras pruebas. Las consideraciones en la planificación, junto con la implementación del compensador térmico y el filtro de ventana en el hardware, permitieron ejecutar pruebas sin la necesidad de un sitio totalmente libre de interferencias y temperatura controlada. Esto es posible gracias a que los parámetros del modelo simplificado totalizan los efectos de distorsión y lo corrigen.
- El método de compensación de ganancia (*gain*) y corrimiento (*offset*) implementado directamente en el hardware, según las pruebas, puede resultar suficiente en diversos casos gracias a que compensa en gran medida el efecto *hard-iron*. La condición a cumplir es que no tenga una fuente de campo magnético cercano que produzca un efecto *soft-iron* intenso.
- El ejercicio de evaluar varios algoritmos de aproximación a una elipse resulta interesante como un potencial aporte debido a que en el estado del arte no se encontró un trabajo comparativo. El caso particular del algoritmo Li/Griffiths es llamativo por su pobre rendimiento, sin embargo, es utilizado en algunos desarrollos de autopilotos para “calibrar” magnetómetros.
- La herramienta Wimo que inicialmente no hacía parte de la planificación de esta investigación resultó útil por una funcionalidad en especial: la representación del movimiento del cubo. Aunque las gráficas generadas en los scripts permitieron verificar los resultados de las pruebas con el sensor, no superaron la ventaja de observar rápidamente la reacción del sensor cuando es sometido a un cambio y la consecuente diferencia de posición entre el cubo virtual y el físico.
- Los algoritmos de Merayo y Petrov entregaron en todos los casos de evaluación unos elipsoides muy cercanos en forma a una esfera, considerando el hecho que la investigación se realizó con un magnetómetro comercial de bajo costo; esto es un logro importante que expande las posibilidades de aplicación de este tipo de sensores.

## 6.2 APORTES Y EXPERIENCIAS

- La utilización de un acelerómetro y un giroscopio integrado en el conjunto de hardware y software no es casual, esta investigación partió de una propuesta inicial de desarrollo de un autopiloto que tuvo avances en la selección del hardware y software. De esto derivó en la utilización del módulo GY-87 y la aplicación Wimo con el concepto de una interfaz de transmisión inalámbrica.
- El nodo implementado es de un tamaño reducido y consumo energético aceptable que ronda los 300mA, gracias a esto es posible emplearlo en futuros trabajos donde sea útil la detección de movimiento como: aplicaciones de realidad aumentada, caracterización mecánica, robótica, incluso rehabilitación física de una persona.

- El proceso de ajuste de ganancia y corrimiento por cálculo de máximos y mínimos también fue evaluado para el giroscopio y el acelerómetro, según las observaciones con la aplicación Wimo es posible afirmar en una primera instancia que este método funciona mejor para estos dos sensores que para el magnetómetro, lo suficiente para obtener mediciones aceptables.
- La corrección térmica fue una consideración útil con magnetómetro, que también mostró una efectividad similar sobre el giroscopio y el acelerómetro. Sin embargo, la necesidad de realizar una caracterización y cálculo de factores de corrección cada vez que se enciende el dispositivo no es una condición deseable en una aplicación.
- En los anexos se entrega la información técnica para replicar el nodo y su firmware, al ser utilizado dentro del entorno de Arduino se espera que esto ofrezca una base para el desarrollo de otro tipo de soluciones experimentales considerando los conceptos de internet de las cosas (IoT, en inglés *Internet of things*) o redes de sensores (WSN, en inglés *Wireless Sensor Network(s)*).

### 6.3 TRABAJOS FUTUROS

- Complementar los resultados de este trabajo con la estimación de parámetros dinámicos de un magnetómetro o de ser posible calibrar el sensor.
- Establecer por qué los factores térmicos que son lineales y estáticos durante una sesión de trabajo cambian al reiniciar el dispositivo. Esta situación aplica tanto al magnetómetro como al giroscopio y el acelerómetro.
- Evaluar los algoritmos de ajuste de elipsoide presentados en este trabajo sobre un acelerómetro, estos dos sensores comparten la característica de determinar la magnitud de un vector direccional en su proceso de medición.
- Extender las evaluaciones presentadas en esta investigación a otro tipo de magnetómetros de tres dimensiones.



## BIBLIOGRAFÍA

---

- [1] A. Zambrano y H. G. Kerkhoff, «Online digital compensation Method for AMR sensors,» de *2016 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, Tallinn, Estonia, 2016.
- [2] A. Platif, J. Kubik, M. Vopalensky y P. Ripka, «Precise AMR magnetometer for compass,» *Proceedings of IEEE Sensors 2003*, vol. 1, pp. 472-476, 2003.
- [3] Y. Wu y W. Shi, «On Calibration of Three-Axis Magnetometer,» *IEEE Sensors Journal*, vol. 15, n° 11, pp. 6424-6431, 2015.
- [4] C. E. Serrano Castaño, *Modelo integral para el profesional en ingeniería*, Popayán: Universidad del Cauca, 2005.
- [5] C. Hoyos Botero, *Un modelo para investigación documental. Guía teórico-práctica sobre construcción de estados del arte con importantes reflexiones sobre la investigación*, Medellín: Señal Editora, 2000.
- [6] K. Tanaka, *Embedded systems - Theory and desing methodology*, Croatia: InTech, 2012.
- [7] B. Kitchenham, S. Linkman y D. Law, «DESMET: a methodology for evaluating software engineering methods and tools,» *Computing Control Engineering Journal*, vol. 8, n° 3, pp. 120-126, 1997.
- [8] B. Kienhuis, E. F. Deprettere, P. v. d. Wolf y K. A. Vissers, «A Methodology to Design Programmable Embedded Systems - The Y-Chart Approach,» *Embedded Processor Design Challenges: Systems, Architectures, Modeling, and Simulation - SAMOS*, pp. 18-37, 2002.
- [9] L. a. M. C. a. V. E. Cordeiro, F. Cruz, D. Patrick, R. Barreto y V. Lucena, «A Platform-Based Software Design Methodology for Embedded Control Systems: An Agile Toolkit,» de *15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ecbs 2008)*, Belfast, 2008.
- [10] Y. Khmelevsky, X. Li y S. Madnick, «Software development using agile and scrum in distributed teams,» de *Systems Conference (SysCon), 2017 Annual IEEE International*, Montreal, 2017.
- [11] K. Beck, *Extreme Programming Explained: Embrace Change*, Boston: Addison-Wesley Longman Publishing Co., Inc., 2000.

- [12] Project Management Institute, Guía de los fundamentos para la dirección de proyectos, Newton Square, Pensilvania: PMI, 2013.
- [13] E. W. Weisstein, «Euler Angles,» Wolfram Math World, [En línea]. Available: <http://mathworld.wolfram.com/EulerAngles.html>. [Último acceso: 22 Abril 2018].
- [14] PAR, «Dos sistemas de coordenadas ortogonales en el que se muestran los ángulos de Euler. CC BY 2.5,» 2007.
- [15] Wikipedia.org, «Gimbal Lock,» [En línea]. Available: [https://en.wikipedia.org/wiki/Gimbal\\_lock](https://en.wikipedia.org/wiki/Gimbal_lock). [Último acceso: 22 Abril 2018].
- [16] MathsPoetry, «Normal situation - the three gimbals are independent,» 2009.
- [17] MathsPoetry, «Gimbal lock - two out of the three gimbals are in the same plane, one degree of freedom is lost,» 2009.
- [18] G. F. Torres del Castillo, «La representación de rotaciones mediante cuaterniones,» *Miscelánea Matemática*, n° 29, pp. 43-50, 1999.
- [19] NASA Mission Planning and Analysis Division, «Shuttle Program. Euler Angles, Quaternions and Transformation Matrices,» NASA, Houston, 1977.
- [20] J. L. Blanco, A tutorial on SE transformation parameterizations and on-manifold optimization, Málaga: Universidad de Málaga, 2013.
- [21] Instituto Latinoamericano de la Comunicación Educativa, «MATERIALES MAGNÉTICOS,» Biblioteca digital ILCE, [En línea]. Available: [http://bibliotecadigital.ilce.edu.mx/sites/ciencia/volumen2/ciencia3/088/html/sec\\_11.html](http://bibliotecadigital.ilce.edu.mx/sites/ciencia/volumen2/ciencia3/088/html/sec_11.html). [Último acceso: 22 Abril 2018].
- [22] W. H. J. Hayt y J. A. Buck, Teoría Electromagnética, México D.F.: McGraw-Hill, 2006.
- [23] Universidad de Sevilla, «Ley de Lorentz,» Universidad de Sevilla: Departamento de Física Aplicada, [En línea]. Available: [http://laplace.us.es/wiki/index.php/Ley\\_de\\_Lorentz](http://laplace.us.es/wiki/index.php/Ley_de_Lorentz). [Último acceso: 24 Abril 2018].
- [24] Woods Hole Oceanographic Institution, «The Magnetic North Pole,» Ocean Bottom Magnetology Laboratory, [En línea]. Available: <http://deeptow.whoi.edu/northpole.html>. [Último acceso: 24 Abril 2018].

- 
- [25] A. Chulliat, S. Macmillan, P. Alken, C. Beggan, M. Nair y B. Hamilton, *The US/UK World Magnetic Model for 2015-2020: Technical Report*, doi: 10.7289/V5TB14V7: National Geophysical Data Center, NOAA, 2015.
- [26] National Geophysical Data Center, NOAA, «The World Magnetic Model - Background,» [En línea]. Available: <https://www.ngdc.noaa.gov/geomag/WMM/back.shtml>. [Último acceso: 24 Abril 2018].
- [27] K. Mohamadabadi, *Anisotropic Magnetoresistance Magnetometer for inertial navigation systems*, Palaiseau: Ecole Polytechnique, 2013.
- [28] D. C. Hovde, M. D. Prouty, I. Hrvoic y R. E. Slocum, «Commercial magnetometers and their application,» *Optical Magnetometry*, pp. 387-405, 2013.
- [29] T. Sakakibara, H. Mitamura, T. Tayama y H. Amitsuka, «Faraday Force Magnetometer for High-Sensitivity Magnetization Measurements at Very Low Temperatures and High Fields,» *Japanese Journal of Applied Physics*, vol. 33, n° 9A, p. 5067, 1994.
- [30] Wikipedia, «Josephson effect,» [En línea]. Available: [https://en.wikipedia.org/wiki/Josephson\\_effect](https://en.wikipedia.org/wiki/Josephson_effect). [Último acceso: 27 Abril 2018].
- [31] J. Clarke y A. I. Braginski, *The SQUID handbook*, Weinheim: Wiley-VCH, 2006.
- [32] J. Olaf, «SQUID IV,» 2009.
- [33] D. F. J. Kimball y D. B. E. B. Alexandrov, «General principles and characteristics of optical magnetometers,» de *Optical Magnetometry*, Berkeley, Cambridge University Press, 2013, pp. 3-24.
- [34] Murata Manufacturing Co. Ltd, «Basic Knowledge of AMR Sensors (Magnetic Switches),» [En línea]. Available: <https://www.murata.com/products/sensor/amr/basic/principle>. [Último acceso: 05 Mayo 2018].
- [35] NVE Corporation, «How GMR Works,» [En línea]. Available: <https://www2.nve.com/gmrsensors/gmr-operation.htm>. [Último acceso: 2 Mayo 2018].
- [36] Multidimension Technology Co. Ltd., «Introduction to TMR Magnetic Sensors,» [En línea]. Available: <http://www.dowaytech.com/en/1776.html>. [Último acceso: 3 Abril 2018].
- [37] Y. Shen, «Applications of Magnetolectric Sensors,» Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2014.
-

- [38] IEEE, «About IEEE Xplore® Digital Library,» IEEE Xplore, [En línea]. Available: <https://ieeexplore.ieee.org/xpl/aboutUs.jsp>. [Último acceso: 3 Mayo 2018].
- [39] Wikipedia, «Google Académico,» [En línea]. Available: [https://es.wikipedia.org/wiki/Google\\_Académico](https://es.wikipedia.org/wiki/Google_Académico). [Último acceso: 3 Mayo 2018].
- [40] Elsevier, «What is ScienceDirect,» [En línea]. Available: <https://www.elsevier.com/solutions/sciencedirect>. [Último acceso: 3 Mayo 2018].
- [41] Springer International Publishing AG, «Knowledge matters. Choose SpringerLink,» [En línea]. Available: <https://www.springer.com/gp/products/springerlink>. [Último acceso: 3 Mayo 2018].
- [42] H. Pang, M. Pan, J. Chen, J. Li, Q. Zhang y S. Luo, «Integrated calibration and magnetic disturbance compensation of three-axis magnetometers,» *Measurement*, vol. 93, pp. 409-413, 2016.
- [43] O. Khurshid, J. Selkainaho, E. Kallio, A. Visala y H. E. Soken, «Pseudo-linear approach to nanosatellite attitude estimation and magnetometer calibration,» de *Recent Advances in Space Technologies (RAST), 2017 8th International Conference on*, Istanbul, 2017.
- [44] M. Kiani, S. H. Pourtakdoust y A. AkbarSheikhy, «Consistent calibration of magnetometers for nonlinear attitude determination,» *Measurement*, vol. 73, pp. 180-190, 2015.
- [45] G. Secer y B. Barshan, «Improvements in deterministic error modeling and calibration of inertial sensors and magnetometers,» *Sensors and Actuators A: Physical*, vol. 247, pp. 522-538, 2016.
- [46] Y. Wu, D. Zou, P. Liu y W. Yu, «Dynamic Magnetometer Calibration and Alignment to Inertial Sensors by Kalman Filtering,» *IEEE Transactions on Control Systems Technology*, vol. 26, n° 2, pp. 716-723, 2018.
- [47] K. Han, H. Han, Z. Wang y F. Xu, «Extended Kalman Filter-Based Gyroscope-Aided Magnetometer Calibration for Consumer Electronic Devices,» *IEEE Sensors Journal*, vol. 17, n° 1, pp. 63-71, 2017.
- [48] M. Kok y T. B. Schön, «Magnetometer Calibration Using Inertial Sensors,» *IEEE Sensors Journal*, vol. 16, n° 14, pp. 5679 - 5689, 2016.
- [49] Y. Somov y C. Hajiyev, «In-flight calibration of magnetometers and low-cost SINS for small land-survey satellite,» de *Electronics, Computers and Artificial Intelligence (ECAI), 2014 6th International Conference on*, Bucharest, 2015.

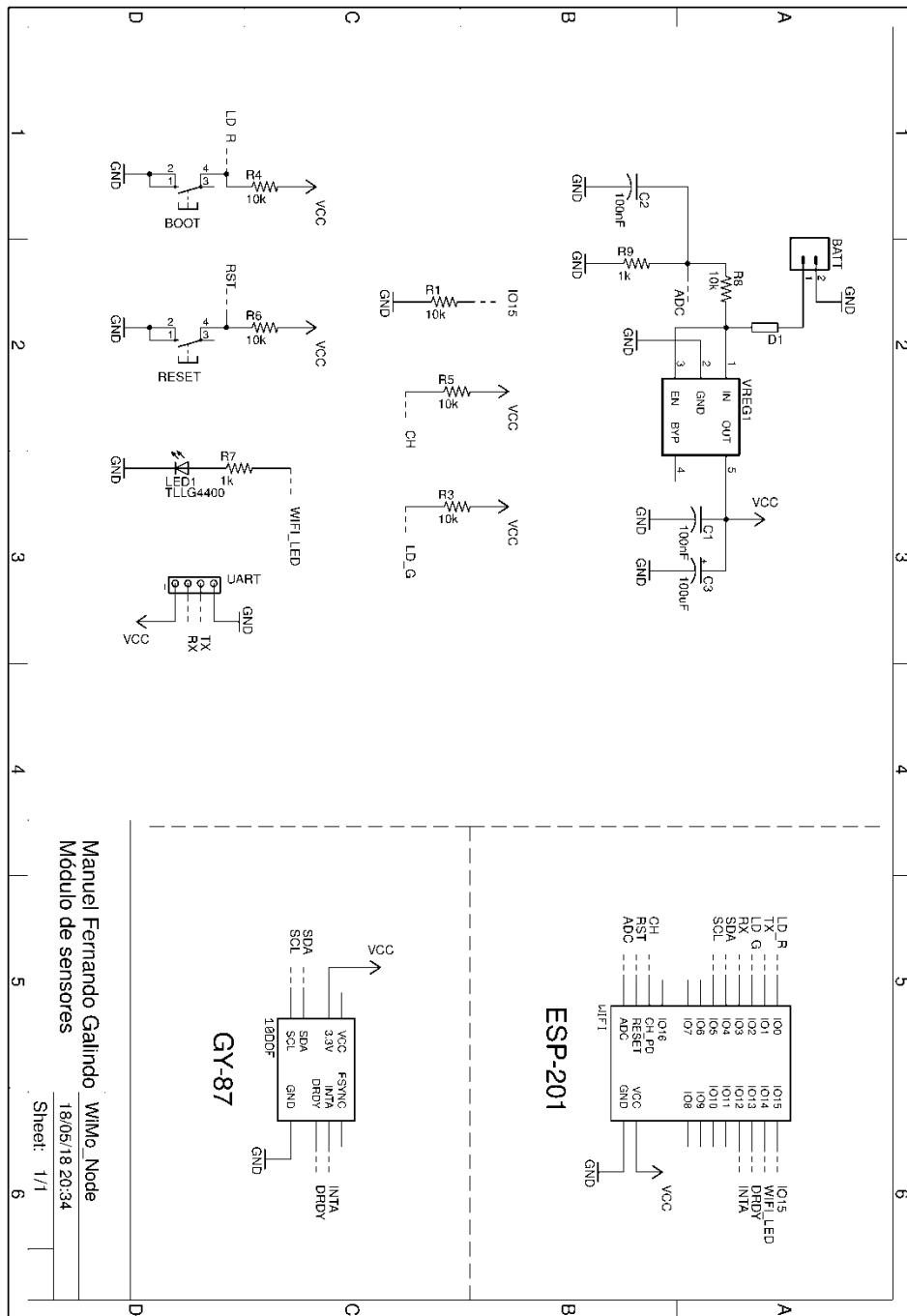
- 
- [50] T. Beravs, S. Beguš, J. Podobnik y M. Munih, «Magnetometer Calibration Using Kalman Filter Covariance Matrix for Online Estimation of Magnetic Field Orientation,» *IEEE Transactions on Instrumentation and Measurement*, vol. 63, n° 8, pp. 2013 - 2020, 2014.
- [51] T. Pylvänäinen, «Automatic and adaptive calibration of 3D field sensors,» *Applied Mathematical Modelling*, vol. 32, n° 4, pp. 575-587, 2008.
- [52] S. Luo, H. Pang, J. li, Q. Zhang, D. Chen, M. Pan y F. Luo, «Calibration strategy and generality test of three-axis magnetometers,» *Measurement*, vol. 46, n° 10, pp. 3918-3923, 2013.
- [53] N. Yadav y C. Bleakley, «Fast calibration of a 9-DOF IMU using a 3 DOF position tracker and a semi-random motion sequence,» *Measurement*, vol. 90, pp. 192-198, 2016.
- [54] D. Gebre-Egziabher, «Magnetometer Autocalibration Leveraging Measurement Locus Constraints,» *Journal of Aircraft*, vol. 44, n° 4, pp. 1361-1368, 2007.
- [55] E. M. Hemerly y F. A. A. Coelho, «Explicit Solution for Magnetometer Calibration,» *IEEE Transactions on Instrumentation and Measurement*, vol. 63, n° 8, pp. 2093-2095, 2014.
- [56] W. Thomson, «Effects of magnetization on the electric conductivity of nickel and of iron,» *On the electro-dynamic qualities of metals*, vol. 8, pp. 546-550, 1856.
- [57] Honeywell Inc., *3-Axis Digital Compass IC HMC5883L*, Plymouth, Minnesota, 2013.
- [58] W. Wong, «Center for Food Safety,» GovHK, 28 Abril 2009. [En línea]. Available: [http://www.cfs.gov.hk/english/food\\_leg/files/method\\_validation.pdf](http://www.cfs.gov.hk/english/food_leg/files/method_validation.pdf). [Último acceso: 16 Mayo 2018].
- [59] International Organization of Legal Metrology, «International Vocabulary of Metrology – Basic and General Concepts and Associated Terms (VIM),» OIML, 2012. [En línea]. Available: [https://www.oiml.org/en/files/pdf\\_v/v002-200-e12.pdf](https://www.oiml.org/en/files/pdf_v/v002-200-e12.pdf). [Último acceso: 16 Mayo 2018].
- [60] Instituto Colombiano de Normas Técnicas, «Norma Técnica Colombiana NTC-ISO 10012,» 26 Junio 2003. [En línea]. Available: <http://idrd.gov.co/sitio/idrd/sites/default/files/imagenes/ntc-iso10012%20medicion%20y%20equipos0.pdf>. [Último acceso: 16 Mayo 2018].
- [61] Plotly, «3D Scatter Plots in Python,» 2018.
- [62] ElectronicFans, «Gy-87 I2C/IIC 10dof,» China, 2018.
-

- [63] Home Appliance CO.,Ltd., «Esp8266 ESP-201,» China, 2018.
- [64] I. Grokhotkov, «Arduino core for ESP8266 WiFi chip,» [En línea]. Available: <https://arduino-esp8266.readthedocs.io/en/2.4.1/>. [Último acceso: 2018 Mayo 19].
- [65] InvenSense Inc., «MPU-6000 and MPU-6050 - Register Map and Descriptions,» Sunnyvale, CA, 2013.
- [66] D. Kliazovich y F. Granelli, «On packet concatenation with QoS support for wireless local area networks,» de *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*, Seúl, 2005.
- [67] K. Kanatani y P. Rangarajan, «Hyper least squares fitting of circles and ellipses,» *Computational Statistics & Data Analysis*, vol. 55, n° 6, pp. 2197-2208, 2011.
- [68] Q. Li y J. Griffiths, «Least squares ellipsoid specific fitting,» de *Geometric Modeling and Processing, 2004. Proceedings*, Beijing, 2004.
- [69] A. Vitali, «DT 0059 - Ellipsoid or sphere fitting for sensor calibration,» 2016.
- [70] J. M. G. Merayo, P. Brauer, F. Primdahl, J. R. Petersen y O. V. Nielsen, «Scalar calibration of vector magnetometers,» *Measurement Science and Technology*, vol. 11, n° 2, p. 120, 2000.
- [71] Y. Petrov, «Ellipsoid fit,» Matlab Central, 2015.

# ANEXOS

## A - ESQUEMAS DE NODO DE SENSORES

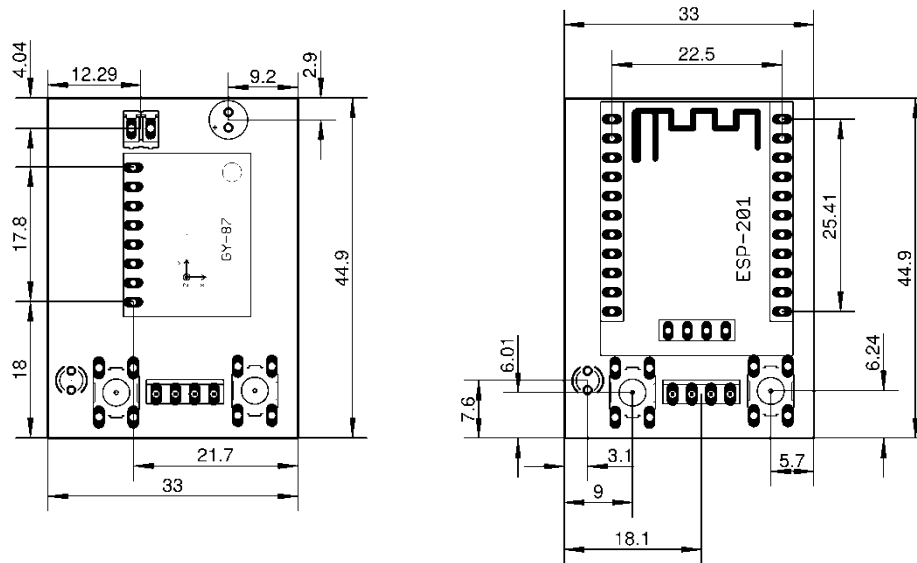
### A.1 Esquema eléctrico



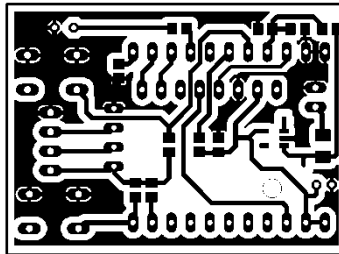
Manuel Ferrnando Gallindo WIMO\_Nodo  
Módulo de sensores

18/05/18 20:34  
Sheet: 1/1

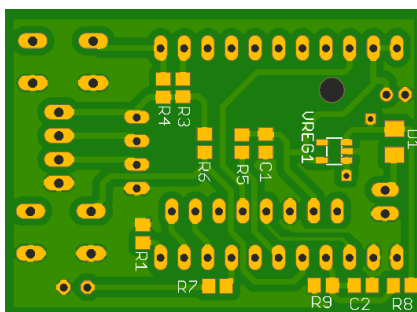
A.2 Esquemas mecánicos (vista superior, escala 1:1)



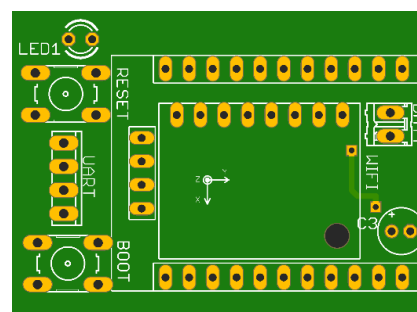
A.3 Esquema de circuito impreso (vista superior, escala 1:1)



A.4 Esquemas de ubicación de elementos



Vista inferior



Vista superior

**NOTA:** Recuerde modificar la antena del módulo ESP-201 que tiene un cortocircuito con GND.




---

**A.5 Bill of Materials**

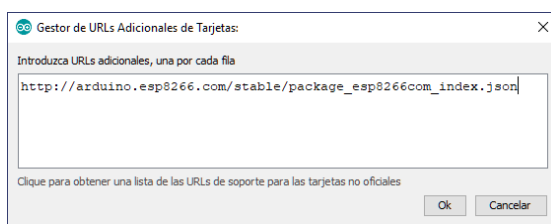
<b>Parte</b>	<b>Valor</b>	<b>Dispositivo</b>	<b>Paquete</b>
10DOF	GY-87	GY-87	GY-87
BATT		M02	02P
BOOT		10-XX	B3F-10XX
C1	100nF	C-USC0805	C0805
C2	100nF	C-USC0805	C0805
C3	100uF	CPOL-USE2-5	E2-5
D1		R-EU_M1206	M1206
LED1	TLLG4400	TLLG4400	LED3MM
R1	10k	R-US_R0805	R0805
R3	10k	R-US_R0805	R0805
R4	10k	R-US_R0805	R0805
R5	10k	R-US_R0805	R0805
R6	10k	R-US_R0805	R0805
R7	1k	R-US_R0805	R0805
R8	10k	R-US_R0805	R0805
R9	1k	R-US_R0805	R0805
RESET		10-XX	B3F-10XX
UART		MTA04-100	10X04MTA
VREG1		VREG_5PINS-SOT-23-5	SOT-23-5
WIFI	ESP-201	ESP-201	ESP201

## B – INSTRUCTIVOS DE USO DEL NODO

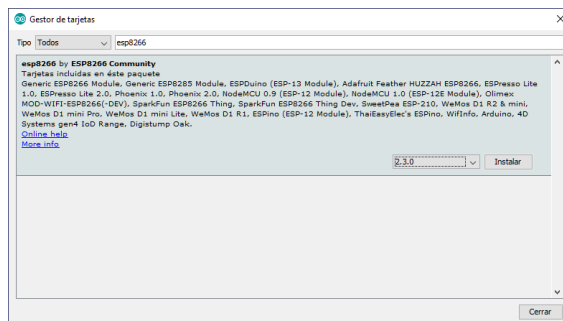
### 1. Configuración del entorno

Como requisito inicial se debe tener instalado el entorno de desarrollo Arduino, para este proyecto se utilizó la versión 1.8.5 en Linux x64 y Windows. Se procede a instalar las librerías de soporte del microcontrolador ESP8266, para ello se abre Arduino IDE y se accede la ventana de Preferencias en el menú Archivo > Preferencias. Luego, en la nueva ventana se hace clic en el botón  que abre la ventana del Gestor de URLs para adicionar tarjetas.

El enlace [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json) se adiciona en la lista y se presiona OK en las dos ventanas abiertas.



Ahora en Arduino IDE se ingresa al menú Herramientas > Placa > Gestor de Tarjetas. En la barra de búsqueda escriba *esp8266*, si el repositorio de librerías está disponible debería aparecer en la lista tras la búsqueda, la versión probada es la 2.3, seleccione la versión y presione Instalar.

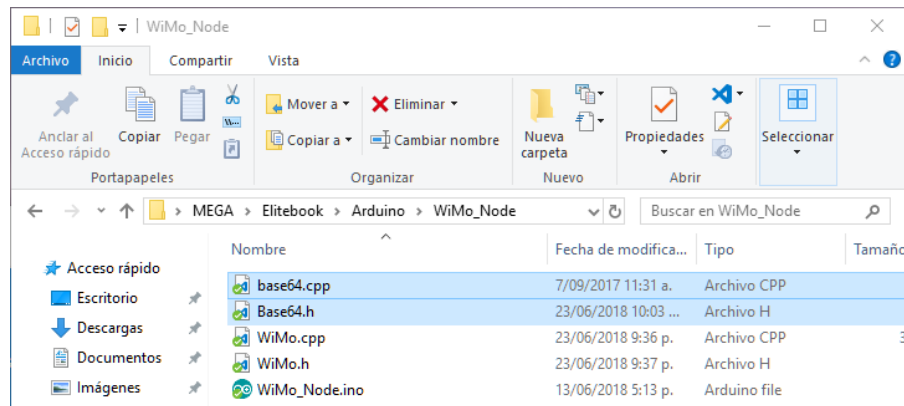


El proceso puede demorar dependiendo de la velocidad de conexión y la disponibilidad de los servidores donde están alojadas las librerías, una vez completado el proceso cierre la ventana.

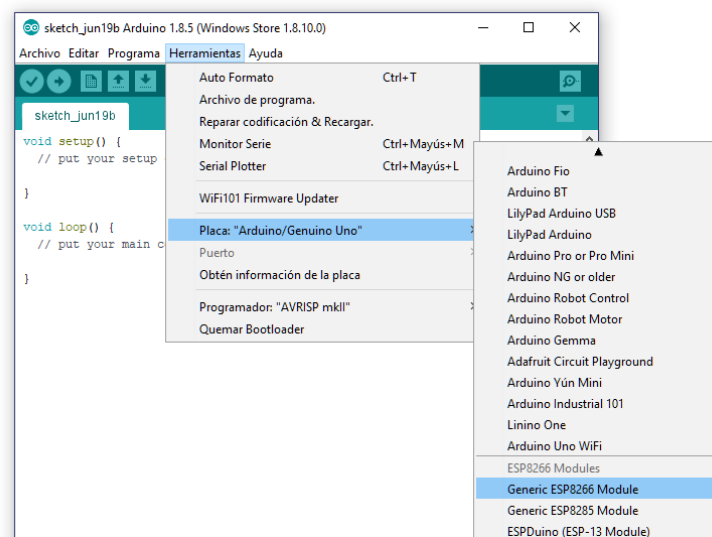
Ahora es el momento de adicionar las librerías I2Cdev, para ello acceda al repositorio en GitHub <https://github.com/jrowberg/i2cdevlib>, descargue las librerías desde el navegador con la opción *Clone or download* descargando el fichero Zip o clonando el repositorio con un software de gestión GIT instalado en su equipo.

De los archivos descargados del repositorio, ingrese en la carpeta Arduino y copie las carpetas I2Cdev, \_Stub, HMC5883L y MPU6050 en la ruta de librerías de Arduino en el equipo. En Windows la ruta por defecto está en Documentos/Arduino/libraries/ y en Linux en ~/Arduino/libraries.

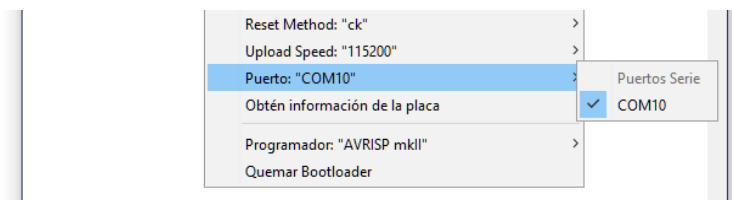
Dentro del proyecto *Wimo\_Node* se encuentra la librería de Base64, el compilador tiene problemas para importar estas funciones desde el directorio libraries, por ello se mantienen estos archivos junto con los del proyecto



Hecho lo anterior, seleccione la placa *Generic ESP8266 Module* en Herramientas > Placa

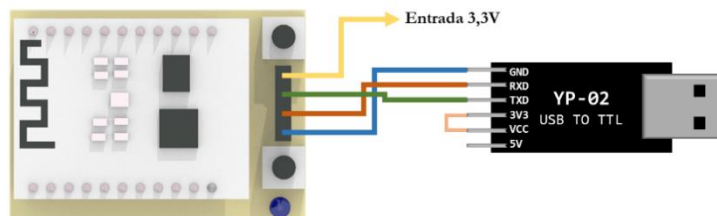


Conecte un adaptador USB a UART con niveles lógicos de 3.3V, configure el controlador del dispositivo si es necesario y luego seleccione el puerto asociado a ese adaptador en Arduino IDE en el menú Herramientas > Puerto:



## 2. Programación del nodo

Conecte el nodo al adaptador USB a TTL conforme al siguiente esquema:



Para alimentar el nodo están disponibles dos opciones: la primera, es emplear una fuente de 3.3V y conectarla a la entrada dispuesta en el puerto UART para programar. Como segunda opción está el alimentar el nodo empleando una batería, puerto USB o cualquier fuente de voltaje entre 5 y 8 voltios a través del conector de batería del nodo. Es importante asegurarse de que la fuente cumpla con una capacidad de corriente de por lo menos 300mA.

Mantenga presionado el botón de usuario y a la vez presione el botón de reinicio (que se encuentra al lado del LED), libere el botón de reinicio sin liberar el de usuario. El LED, que normalmente está activo, tras lo anterior ilumina tenuemente indicando que se encuentra en modo de programación.

En Arduino IDE abra el proyecto, realice los ajustes sobre el código que considere necesarios y luego ejecute el comando *Subir* ubicado en el menú “Programa”, si la compilación y la conexión con el nodo es correcta debería ver algo similar a lo mostrado en la siguiente imagen:

La imagen muestra la interfaz de usuario de Arduino IDE. En la parte superior, se ve el menú de archivo y el nombre del proyecto: 'WiMo\_Node Arduino 1.8.5 (Windows Store 1.8.10.0)'. El editor de código muestra un archivo llamado 'WiMo.cpp' con el siguiente código:

```
/**
 * @brief Ejemplo de implementación de clase WiMo en Arduino
 *
 * @file WiMo_Node.ino
 * @author Manuel Fernando Galindo
 * @date 2018-06-13
 * @par ©2018 - Copyright Manuel Fernando Galindo, Universidad del Cauca
 */

#include "WiMo.h"

uint16_t UDPPORT = 42100;          /**< Puerto UDP del nodo*/
char *AFSSID = "WiMo";           /**< Nombre de la red creada por el n

IPAddress APiPNode(192, 168, 180, 180); /**< Dirección IP del nodo en modo AP
IPAddress gateway(192, 168, 1, 1);    /**< Dirección IP de la ruta predeter
IPAddress subnet(255, 255, 255, 0);  /**< Máscara de red*/
char *ClientSSID = "nombre_de_red"; /**< Nombre de la red a conectarse en
char *ClientPWD = "contraseña_red"; /**< Contraseña de la red a conectarse
IPAddress ClientIP(192,168,1,180);   /**< Dirección IP manual del nodo en :


WiMo node(UDPPORT);                /**< Creación de objeto WiMo*/
```

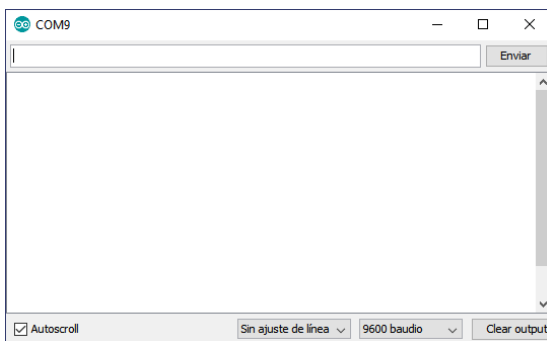
Debajo del código, se ve un indicador de progreso que dice 'Subiendo...' con una barra verde que está casi completa. En la parte inferior, se muestra el mensaje de compilación y subida:

```
Archiving built core (caching) in: C:\Users\mfgal\AppData\Local\Temp\arduino_cache
El Sketch usa 245469 bytes (56%) del espacio de almacenamiento de programa. El máx
Las Variables Globales usan 40012 bytes (48%) de la memoria dinámica, dejando 4190
Uploading 249616 bytes from C:\Users\mfgal\AppData\Local\Temp\arduino_build_699777
```

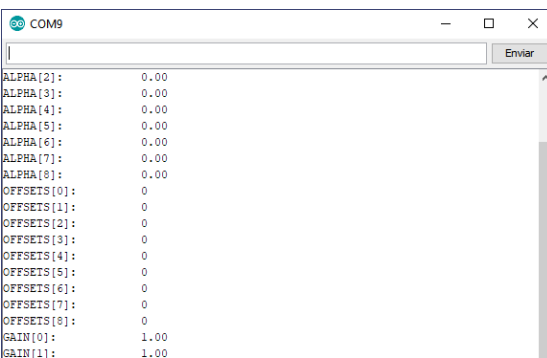
Al finalizar el nodo automáticamente ejecutará el nuevo firmware programado.

### 3. Información de salida por puerto UART

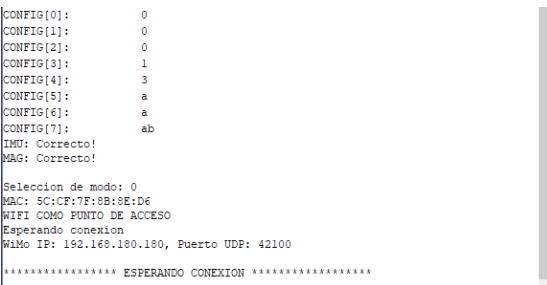
Con la conexión empleada en la sección anterior para el proceso de programación, es posible obtener información sobre la configuración y los sensores, estos datos resultan útiles para verificar el correcto funcionamiento del nodo. Utilizando Arduino IDE con el puerto ya configurado haga clic en el botón  o seleccione *Monitor Serie* en el menú herramientas, esta acción abrirá una ventana secundaria con un terminal como el que se muestra en la siguiente imagen:



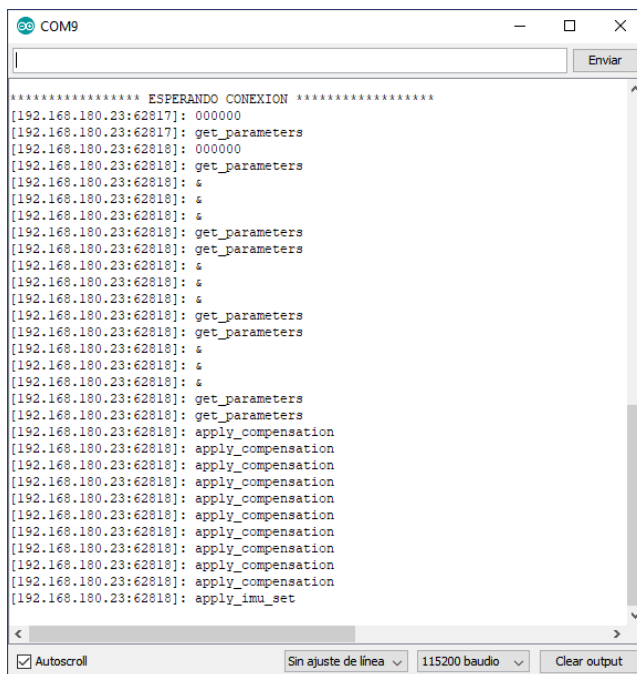
Cambie la velocidad de transferencia de 9600 baudios a 115200 baudios o la que haya configurado en el firmware, luego presione el botón de reinicio en el nodo, inmediatamente el nodo transmitirá los parámetros de corrección implementados en el firmware:



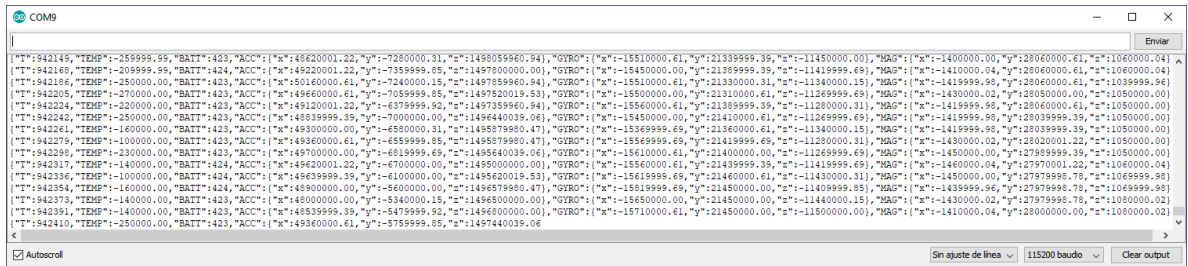
Por una cuestión de implementación de la clase Serial en Arduino, no es posible mostrar los valores double con más de dos decimales. Luego de mostrar los parámetros se informa sobre los ajustes de configuración del nodo, el estado de los sensores y posteriormente la configuración de red:



Una vez establecida la conexión e iniciada la recepción de mensajes desde la aplicación que esté consultado al nodo vía WiFi, los mensajes recibidos serán mostrados en el Monitor Serie con el formato [IP Origen: Puerto UDP]: Mensaje.



Si está activada la función transmitir datos por UART, se mostrarán mensajes con una estructura JSON que contendrá las mediciones obtenidas de los sensores, para compensar la pérdida de decimales los valores float o double en la estructura están multiplicados por cien mil, el usuario tendrá que dividir los valores posteriormente para corregir. Este método no se recomienda para la captura de datos porque no cuenta con una resolución tan alta como la que está disponible por UDP; el objetivo de esta implementación es ofrecer un medio de verificación de integridad y calidad de la conexión de red.



## C – DOCUMENTACIÓN DEL FIRMWARE EN EL NODO

Este proyecto ha sido compilado con la versión 1.8.5 de Arduino, el compilador xtensa con las librerías para ESP8266 mantenidas por la comunidad en su versión 2.3 y el conjunto de librerías I2Cdev para controlar los dispositivos HMC5883L y MPU6050.

Enlaces:

Arduino: <https://www.arduino.cc/>

ESP8266 Arduino Core: <https://github.com/esp8266/Arduino>

I2Cdev: <https://github.com/jrowberg/i2cdevlib>

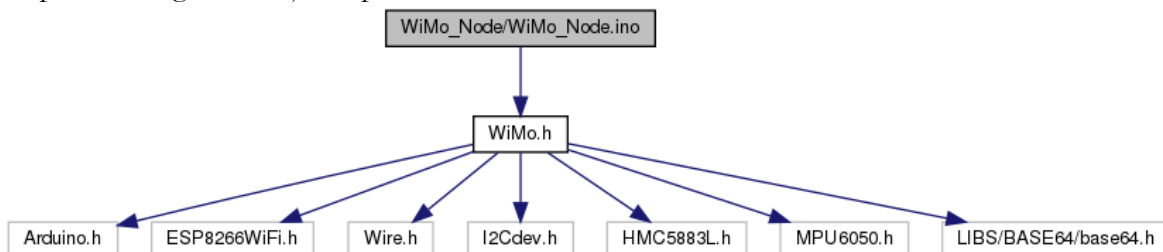
La siguiente información ha sido generada con la ayuda de doxygen:

### Referencia del Archivo WiMo\_Node/WiMo\_Node.ino

Ejemplo de implementación de clase **WiMo** en Arduino.

```
#include "WiMo.h"
```

Dependencia gráfica adjunta para WiMo\_Node.ino:



### Funciones

- IPAddress **APIPNode** (192, 168, 180, 180)
- IPAddress **gateway** (192, 168, 1, 1)
- IPAddress **subnet** (255, 255, 255, 0)
- IPAddress **ClientIP** (192, 168, 1, 180)
- void **setup** ()
- void **loop** ()

### Variables

- uint16\_t **UDPPORT** = 42100
- char \* **APSSID** = "WiMo"
- char \* **ClientSSID** = "nombre\_de\_red"
- char \* **ClientPWD** = "contrasena\_red"
- **WiMo node** (UDPPORT)

## Descripción detallada

Ejemplo de implementación de clase **WiMo** en Arduino.

Autor:

Manuel Fernando Galindo

Fecha:

2018-06-13

©2018 - Copyright Manuel Fernando Galindo, Universidad del Cauca

---

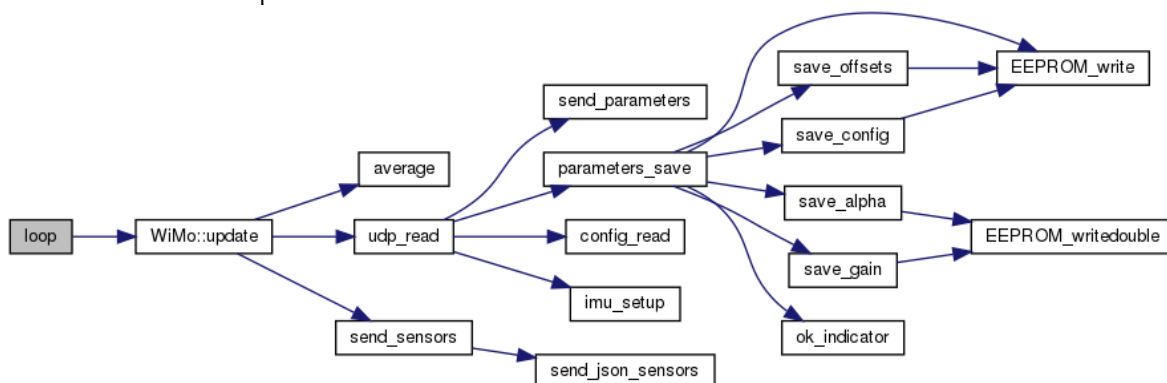
## Documentación de las funciones

### void loop ()

< Ejecución de tareas del nodo

Definición en la línea 33 del archivo WiMo\_Node.ino.

Gráfico de llamadas para esta función:



### void setup ()

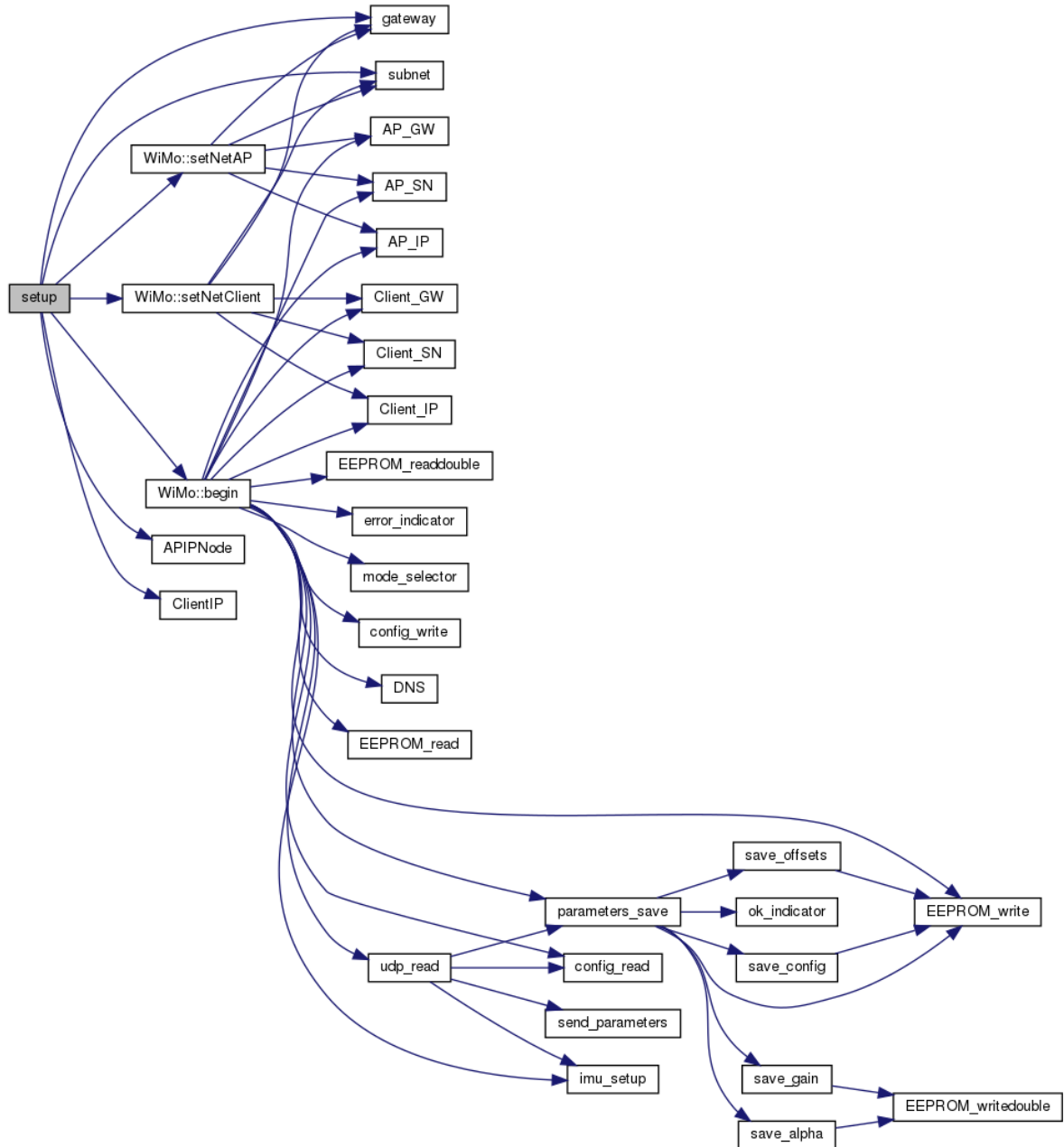
< Iniciar el nodo **WiMo**

Definición en la línea 24 del archivo WiMo\_Node.ino.

---



Gráfico de llamadas para esta función:



## Documentación de las variables

**char\* APSSID = "WiMo"**

Nombre de la red creada por el nodo

**char\* ClientSSID = "nombre\_de\_red"**

Nombre de la red a conectarse en modo cliente

**char\* ClientPWD = "contrasena\_red"**

Contraseña de la red a conectarse

**WiMo node(UDPPORT)**

Creación de objeto **WiMo**

**uint16\_t UDPPORT = 42100**

Puerto UDP del nodo

**IPAddress APIPNode (192 , 168 , 180 , 180 )**

Dirección IP del nodo en modo AP

**IPAddress ClientIP (192 , 168 , 1 , 180 )**

Dirección IP manual del nodo en modo cliente

**IPAddress gateway (192 , 168 , 1 , 1 )**

Dirección IP de la ruta predeterminada

**IPAddress subnet (255 , 255 , 255 , 0 )**

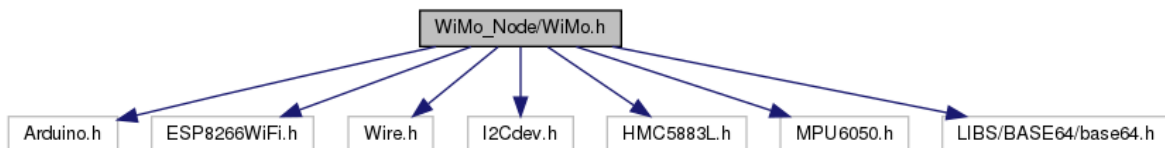
Máscara de red

## Referencia del Archivo **WiMo\_Node/WiMo.h**

Definiciones de clase **WiMo** implementada en **WiMo.cpp** y directivas de configuración y manejo de memoria.

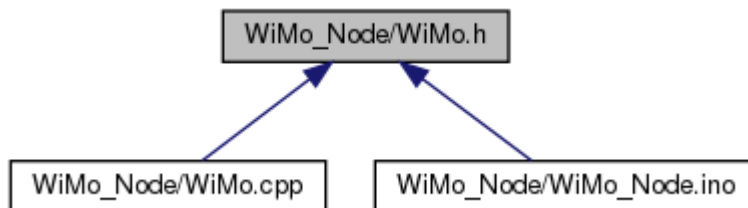
```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <Wire.h>
#include "I2Cdev.h"
#include "HMC5883L.h"
#include "MPU6050.h"
#include "LIBS/BASE64/base64.h"
```

Dependencia gráfica adjunta para **WiMo.h**:



---

Gráfico de los archivos que directa o indirectamente incluyen a este archivo:



## Clases

- class **WiMo**

Definición de clase WiMo.

## Defines

Mapa de memoria de elementos en EEPROM

- #define **CONFIG\_EEPROM\_LOC** 0
  - #define **EEADDR\_CFG** 0
  - #define **EEADDR\_ACC\_SCALE** 1
  - #define **EEADDR\_GYRO\_SCALE** 2
  - #define **EEADDR\_MAG\_SCALE** 3
  - #define **EEADDR\_IMU\_DLPF** 4
  - #define **EEADDR\_SWLFP** 5
  - #define **EEADDR\_TX\_PERIOD** 6
  - #define **EEADDR\_SIGN** 7
  - #define **ALPHA\_EEPROM\_LOC** 8
  - #define **EEADDR\_ALPHA\_ACCX** 8
  - #define **EEADDR\_ALPHA\_ACCY** 16
  - #define **EEADDR\_ALPHA\_ACCZ** 24
  - #define **EEADDR\_ALPHA\_GYROX** 32
  - #define **EEADDR\_ALPHA\_GYROY** 40
  - #define **EEADDR\_ALPHA\_GYROZ** 48
  - #define **EEADDR\_ALPHA\_MAGX** 56
  - #define **EEADDR\_ALPHA\_MAGY** 64
  - #define **EEADDR\_ALPHA\_MAGZ** 72
  - #define **GAIN\_EEPROM\_LOC** 80
  - #define **EEADDR\_ALPHA\_ACCX** 80
  - #define **EEADDR\_ALPHA\_ACCY** 88
  - #define **EEADDR\_ALPHA\_ACCZ** 96
  - #define **EEADDR\_ALPHA\_GYROX** 104
  - #define **EEADDR\_ALPHA\_GYROY** 112
  - #define **EEADDR\_ALPHA\_GYROZ** 120
  - #define **EEADDR\_ALPHA\_MAGX** 128
  - #define **EEADDR\_ALPHA\_MAGY** 136
  - #define **EEADDR\_ALPHA\_MAGZ** 144
  - #define **OFFSET\_EEPROM\_LOC** 160
-

- `#define EEADDR_OFF_ACCX 160`
- `#define EEADDR_OFF_ACCY 162`
- `#define EEADDR_OFF_ACCZ 164`
- `#define EEADDR_OFF_GYROX 166`
- `#define EEADDR_OFF_GYROY 168`
- `#define EEADDR_OFF_GYROZ 170`
- `#define EEADDR_OFF_MAGX 172`
- `#define EEADDR_OFF_MAGY 174`
- `#define EEADDR_OFF_MAGZ 176`

Firma de verificación de integridad de EEPROM

- `#define VERIFY_SIGN 0xAB`

Tamaño de buffer de filtro pasabajas

- `#define LPF_BUFFER_SIZE 255`

Asignación de pines en ESP8266

- `#define INTA_GPIO 12 //Interrupción de IMU`
- `#define DRDY_GPIO 13 //Interrupción de magnetómetro`
- `#define LED_GPIO 14 //Pin asignado a LED`
- `#define MODE_GPIO 0 //Pin asignado a botón de usuario`
- `#define BATT_GPIO A0 //Pin de medición de voltaje de batería`

---

## Descripción detallada

Definiciones de clase **WiMo** implementada en **WiMo.cpp** y directivas de configuración y manejo de memoria.

**Autor:**

Manuel Fernando Galindo

**Fecha:**

2018-06-13

©2018 - Copyright Manuel Fernando Galindo, Universidad del Cauca

## Referencia de la Clase WiMo

Definición de clase **WiMo**.

```
#include <WiMo.h>
```

### Métodos públicos

- **WiMo** (uint16\_t udp\_p)  
*Construye un nuevo objeto new Wi Mo.: Wi Mo.*

- void **setNetAP** (char \*SSID, IPAddress ip\_local, IPAddress **gateway**, IPAddress **subnet**)  
*Configuración de red en modo punto de acceso del nodo.*
- void **setNetClient** (char \*SSID, char \*Pwd, IPAddress ip\_local, IPAddress **gateway**, IPAddress **subnet**)  
*Configuración de red cifrada en modo cliente en modo manual.*
- void **setNetClient** (char \*SSID, IPAddress ip\_local, IPAddress **gateway**, IPAddress **subnet**)  
*Configuración de red no cifrada en modo cliente en modo manual.*
- void **setNetClient** (char \*SSID)  
*Configuración de la red no cifrada en modo cliente con direcciones asignadas por DHCP.*
- void **setNetClient** (char \*SSID, char \*Pwd)  
*Configuración de red cifrada en modo cliente con direcciones asignadas por DHCP.*
- void **begin** ()  
*Realiza las configuraciones del nodo para iniciar la operación.*
- void **update** ()  
*Actualiza las operaciones de lectura y transmisión del nodo.*

---

## Descripción detallada

Definición de clase **WiMo**.

Definición en la línea 120 del archivo WiMo.h.

---

## Documentación del constructor y destructor

### WiMo::WiMo (uint16\_t *udp\_p*)

Construye un nuevo objeto new Wi Mo:: Wi Mo.

COMM\_OP

Parámetros:

<i>udp_p</i>	puerto UDP del nodo
--------------	---------------------

Definición en la línea 738 del archivo WiMo.cpp.

---

## Documentación de las funciones miembro

### void WiMo::begin ()

Realiza las configuraciones del nodo para iniciar la operación.

Definición en la línea 746 del archivo WiMo.cpp.

Gráfico de llamadas para esta función:

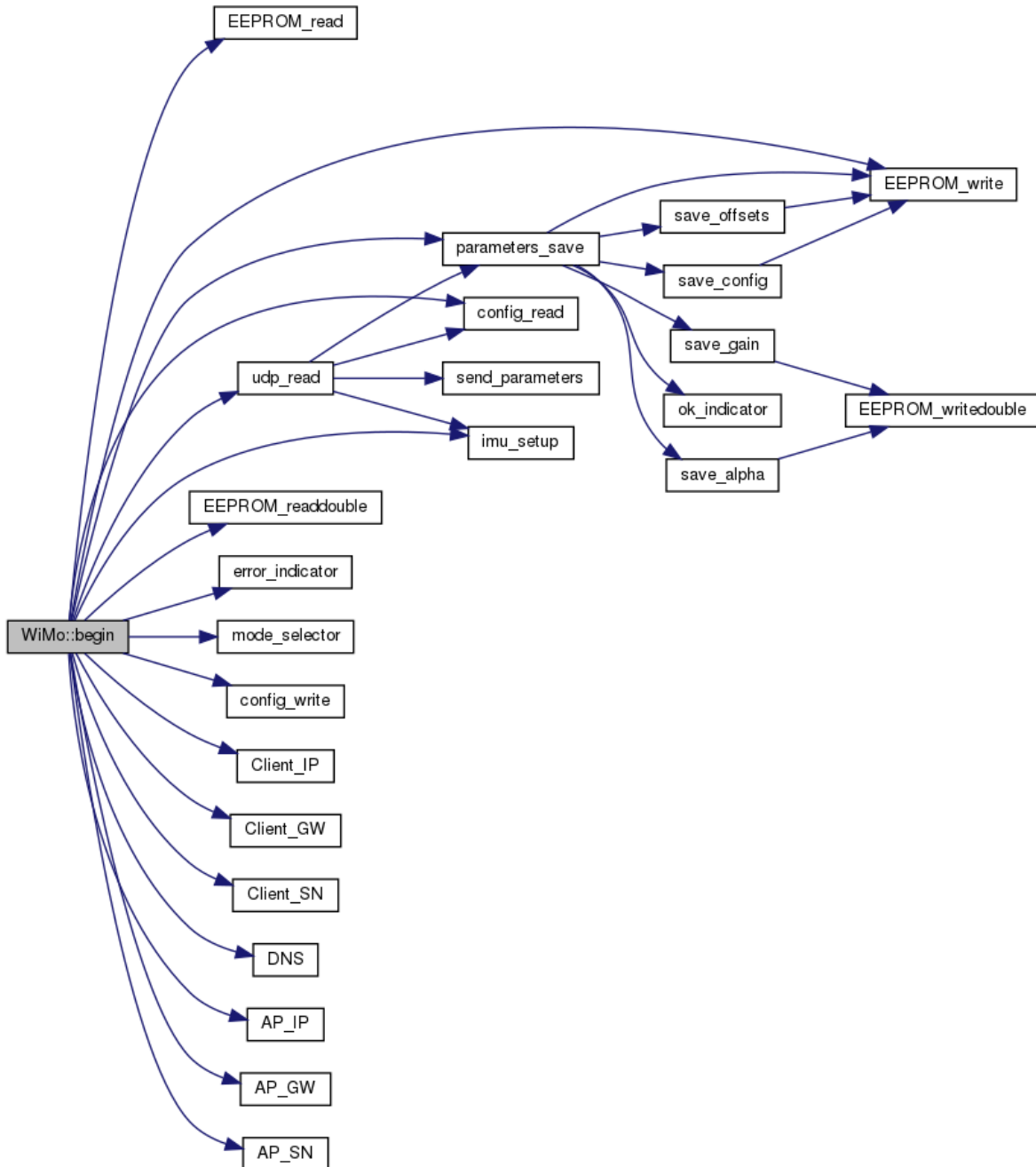


Gráfico de llamadas a esta función:



---

```
void WiMo::setNetAP (char * SSID, IPAddress ip_local, IPAddress gateway,
IPAddress subnet)
```

Configuración de red en modo punto de acceso del nodo.

Parámetros:

<i>SSID</i>	Nombre de la red creada por el nodo
<i>ip_local</i>	Dirección IP del nodo
<i>gateway</i>	Dirección IP de ruta predeterminada
<i>subnet</i>	Máscara de red

Definición en la línea 1039 del archivo WiMo.cpp.

```
void WiMo::setNetClient (char * SSID, char * Pwd, IPAddress ip_local, IPAddress
gateway, IPAddress subnet)
```

Configuración de red cifrada en modo cliente en modo manual.

Parámetros:

<i>SSID</i>	Nombre de la red a conectarse
<i>Pwd</i>	Contraseña de la red
<i>ip_local</i>	Dirección IP para asignar al nodo
<i>gateway</i>	Dirección IP de ruta predeterminada
<i>subnet</i>	Máscara de red

Definición en la línea 1055 del archivo WiMo.cpp.

```
void WiMo::setNetClient (char * SSID, IPAddress ip_local, IPAddress gateway,
IPAddress subnet)
```

Configuración de red no cifrada en modo cliente en modo manual.

Parámetros:

<i>SSID</i>	Nombre de la red a conectarse
<i>ip_local</i>	Dirección IP de nodo
<i>gateway</i>	Dirección IP de ruta predeterminada
<i>subnet</i>	Máscara de red

Definición en la línea 1073 del archivo WiMo.cpp.

```
void WiMo::setNetClient (char * SSID)
```

Configuración de la red no cifrada en modo cliente con direcciones asignadas por DHCP.

---

Parámetros:

<i>SSID</i>	Nombre de la red a conectarse
-------------	-------------------------------

Definición en la línea 1099 del archivo WiMo.cpp.

**void WiMo::setNetClient (char \* *SSID*, char \* *Pwd*)**

Configuración de red cifrada en modo cliente con direcciones asignadas por DHCP.

Parámetros:

<i>SSID</i>	Nombre de la red a conectarse
<i>Pwd</i>	Contraseña de la red

Definición en la línea 1087 del archivo WiMo.cpp.

**void WiMo::update ()**

Actualiza las operaciones de lectura y transmisión del nodo.

Definición en la línea 947 del archivo WiMo.cpp.

Gráfico de llamadas para esta función:

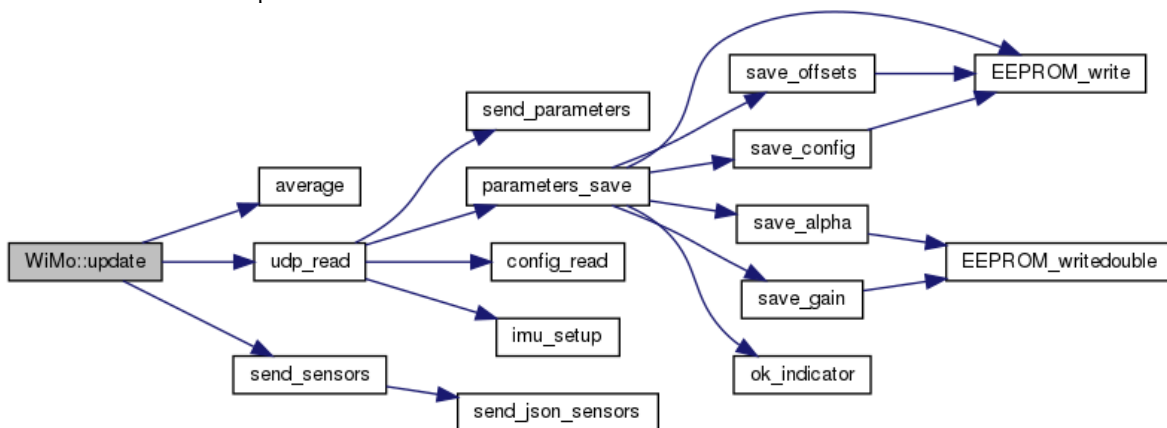


Gráfico de llamadas a esta función:



La documentación para esta clase fue generada a partir de los siguientes ficheros:

- WiMo\_Node/WiMo.h
  - WiMo\_Node/WiMo.cpp
-

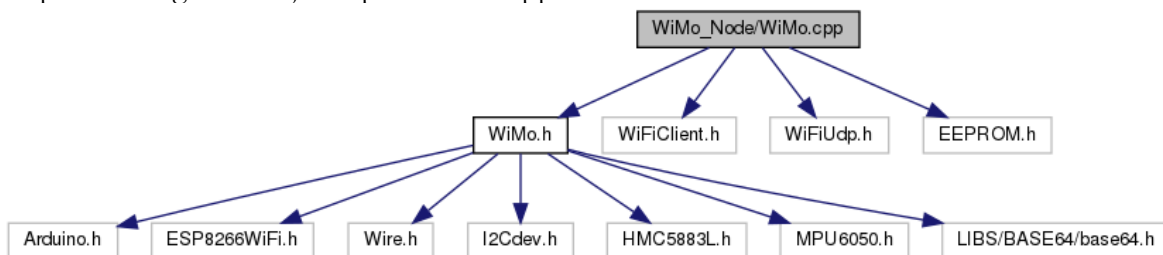


## Referencia del Archivo WiMo\_Node/WiMo.cpp

Este archivo contiene todas las funciones requeridas para la operación y captura de datos de un nodo de sensores con una unidad de medición inercial MPU6050 y un magnetómetro HMC5883L sobre un microcontrolador ESP8266.

```
#include "WiMo.h"
#include <WiFiClient.h>
#include <WiFiUdp.h>
#include <EEPROM.h>
```

Dependencia gráfica adjunta para WiMo.cpp:



## Clases

- struct **Vector3D**

Crea una estructura de tres dimensiones tipo double.

## Funciones

- IPAddress **AP\_IP** (192, 168, 180, 1)
- IPAddress **AP\_GW** (192, 168, 1, 1)
- IPAddress **AP\_SN** (255, 255, 255, 0)
- IPAddress **Client\_IP** (192, 168, 180, 1)
- IPAddress **Client\_GW** (192, 168, 1, 1)
- IPAddress **Client\_SN** (255, 255, 255, 0)
- IPAddress **DNS** (8, 8, 8, 8)
- template<typename RData > void **EEPROM\_read** (uint8\_t addr, RData &arg)  
*Lee un valor entero de cualquier tamaño de bit en EEPROM.*
- template<typename WData > void **EEPROM\_write** (uint8\_t addr, WData &arg)  
*Escribe un valor entero de cualquier tamaño de bit en EEPROM.*
- template<typename RData > void **EEPROM\_readdouble** (uint8\_t addr, RData &arg)  
*Lee un valor de punto flotante double de EEPROM.*
- template<typename WData > void **EEPROM\_writedouble** (uint8\_t addr, WData &arg)  
*Escribe un valor de punto flotante double en EEPROM.*
- void **save\_offsets** ()  
*Guarda en bloque los factores de corrimiento en EEPROM.*
- void **save\_gain** ()

- *Guarda en bloque los factores de ganancia en EEPROM.*
- void **save\_alpha** ()  
*Guarda en bloque los factores de corrección térmica en EEPROM.*
- void **save\_config** ()  
*Guarda en bloque los valores de configuración del nodo en EEPROM.*
- float **average** (int16\_t \*vec)  
*Calcula el promedio de un vector.*
- void **error\_indicator** ()  
*Rutina de LED para indicar error.*
- void **ok\_indicator** ()  
*Rutina de LED para indicar éxito en una operación.*
- uint8\_t **mode\_selector** ()  
*Rutina para determinar un valor de modo a través del botón de usuario.*
- void **parameters\_save** ()  
*Actualiza todos los valores de configuración en EEPROM.*
- uint8\_t **config\_read** (uint8\_t bit)  
*Lee bit de configuración del registro CFG de la variable config.*
- void **config\_write** (uint8\_t bit, uint8\_t value)  
*Escribe bit de configuración en registro CFG de la variable config.*
- void **imu\_setup** ()  
*Actualiza registros de configuración en MPU6050(IMU) y HMC5883L(Magnetómetro)*
- void **send\_parameters** ()  
*Construcción y envío de parámetros del nodo.*
- void **udp\_read** ()  
*Decodifica un mensaje UDP recibido.*
- void **send\_json\_sensors** ()  
*Cuando está habilitada la transmisión serial, se construye un mensaje JSON con la información de sensores.*
- void **send\_sensors** ()  
*Construye un mensaje UDP con la información de los sensores.*

## Variables

- MPU6050 **imu**
  - HMC5883L **magnetometer**
  - WiFiUDP **Udp**
  - uint16\_t **UDP\_PORT** = 42100
  - uint16\_t **remote\_port** = 0
  - IPAddress **client\_address**
  - char \* **AP\_SSID** = "WiMo"
  - char \* **Client\_SSID** = ""
  - char \* **Client\_PWD** = ""
  - bool **wifi\_mode** = false
  - bool **wifi\_manualIP** = false
  - bool **wifi\_withPWD** = false
-

- String **data\_msg**
- char **incomingPacket** [255]
- char **oPack** [170]
- char **b64** [256] = {0}
- uint8\_t **b64\_size** = 0
- uint8\_t **config** [] = {0,0,0,1,3,10,10,VERIFY\_SIGN}
- double **ALPHA** [9] = {0,0,0,0,0,0,0,0,0}
- int16\_t **OFFSETS** [9] = {0,0,0,0,0,0,0,0,0}
- double **GAIN** [9] = {1,1,1,1,1,1,1,1,1}
- **Vector3D** **acc**
- **Vector3D** **gyro**
- **Vector3D** **mag**
- double **temp**
- int16\_t **acc\_m** [3][**LPF\_BUFFER\_SIZE**] = {{0}}
- int16\_t **gyro\_m** [3][**LPF\_BUFFER\_SIZE**] = {{0}}
- int16\_t **mag\_m** [3][**LPF\_BUFFER\_SIZE**] = {{0}}
- int16\_t **temp\_m** [**LPF\_BUFFER\_SIZE**] = {0}
- uint8\_t **IMU\_SAMPLE\_COUNTER** = 0
- uint8\_t **MAG\_SAMPLE\_COUNTER** = 0
- bool **sensor\_status** = true
- uint32\_t **adread** = 0
- uint32\_t **last\_tx** = 0
- uint32\_t **system\_now** = 0
- uint16\_t **led\_value** = 0
- bool **led\_increment** = true
- bool **serial\_tx** = false
- bool **offset\_compensation** = false
- bool **thermal\_compensation** = false
- bool **gain\_compensation** = false

---

## Descripción detallada

Este archivo contiene todas las funciones requeridas para la operación y captura de datos de uno nodo de sensores con una unidad de medición inercial MPU6050 y un magnetómetro HMC5883L sobre un microcontrolador ESP8266.

### Autor:

Manuel Fernando Galindo

### Fecha:

2018-06-13

©2018 - Copyright Manuel Fernando Galindo, Universidad del Cauca

## Módulos en WiMo.cpp

### CLASS\_DEF

Creación de objetos de librerías.

### Variables

- MPU6050 **imu**
  - HMC5883L **magnetometer**
  - WiFiUDP **Udp**
- 

### Descripción detallada

Creación de objetos de librerías.

---

### Documentación de las variables

#### MPU6050 imu

MPU6050 etiquetada como "imu"

Definición en la línea 32 del archivo WiMo.cpp.

#### HMC5883L magnetometer

HMC5883L etiquetada como "magnetometer"

Definición en la línea 33 del archivo WiMo.cpp.

#### WiFiUDP Udp

Stack UPD etiquetada como "Udp"

Definición en la línea 34 del archivo WiMo.cpp.

### NET\_DEF

Declaraciones de variables de configuración de red.

### Funciones

- IPAddress **AP\_IP** (192, 168, 180, 1)
  - IPAddress **AP\_GW** (192, 168, 1, 1)
  - IPAddress **AP\_SN** (255, 255, 255, 0)
  - IPAddress **Client\_IP** (192, 168, 180, 1)
  - IPAddress **Client\_GW** (192, 168, 1, 1)
  - IPAddress **Client\_SN** (255, 255, 255, 0)
  - IPAddress **DNS** (8, 8, 8, 8)
-

## Variables

- `uint16_t UDP_PORT = 42100`
  - `uint16_t remote_port = 0`
  - `IPAddress client_address`
  - `char * AP_SSID = "WiMo"`
  - `char * Client_SSID = ""`
  - `char * Client_PWD = ""`
  - `bool wifi_mode = false`
  - `bool wifi_manualIP = false`
  - `bool wifi_withPWD = false`
- 

## Descripción detallada

Declaraciones de variables de configuración de red.

---

## Documentación de las variables

### **IPAddress AP\_GW (192 , 168 , 1 , 1 )**

Dirección de ruta predeterminada en modo AP

### **IPAddress AP\_IP (192 , 168 , 180 , 1 )**

Dirección IP del nodo cuando es AP (Access Point)

### **IPAddress AP\_SN (255 , 255 , 255 , 0 )**

Máscara de red en modo AP

### **IPAddress Client\_GW (192 , 168 , 1 , 1 )**

Dirección de ruta predeterminada en modo cliente

### **IPAddress Client\_IP (192 , 168 , 180 , 1 )**

Dirección IP del nodo en modo cliente

### **IPAddress Client\_SN (255 , 255 , 255 , 0 )**

Máscara de red en modo cliente

### **IPAddress DNS (8 , 8 , 8 , 8 )**

Dirección de servidor DNS

### **char\* AP\_SSID = "WiMo"**

Nombre de la red (SSID) en modo AP

Definición en la línea 54 del archivo WiMo.cpp.

### **IPAddress client\_address**

Dirección IP de destino PC cliente

Definición en la línea 46 del archivo WiMo.cpp.

### **char\* Client\_PWD = ""**

Contraseña de la red cliente

Definición en la línea 56 del archivo WiMo.cpp.

---

**char\* Client\_SSID = ""**

Nombre de la red en modo cliente  
Definición en la línea 55 del archivo WiMo.cpp.

**uint16\_t remote\_port = 0**

Puerto UDP de destino (PC Cliente)  
Definición en la línea 45 del archivo WiMo.cpp.

**uint16\_t UDP\_PORT = 42100**

Puerto UDP asignado al nodo  
Definición en la línea 44 del archivo WiMo.cpp.

**bool wifi\_manualIP = false**

Bandera de configuración IP. false: Automática(DHCP), true: Manual  
Definición en la línea 59 del archivo WiMo.cpp.

**bool wifi\_mode = false**

Según configuración CFG.Bandera de selección de modo WiFi. false:Modo AP, true:Modo cliente  
Definición en la línea 57 del archivo WiMo.cpp.

**bool wifi\_withPWD = false**

Bandera de indicación de red cifrada. false: Sin contraseña, true: Cifrada  
Definición en la línea 61 del archivo WiMo.cpp.

## MSG\_DEF

Reservas de memoria para construir/recibir mensajes.

## Variables

- String **data\_msg**
- char **incomingPacket** [255]
- char **oPack** [170]
- char **b64** [256] = {0}
- uint8\_t **b64\_size** = 0

---

## Descripción detallada

Reservas de memoria para construir/recibir mensajes.

---

## Documentación de las variables

**char b64[256] = {0}**

Buffer de salida codificado en base64  
Definición en la línea 75 del archivo WiMo.cpp.

---

**uint8\_t b64\_size = 0**

Tamaño del buffer base64

Definición en la línea 76 del archivo WiMo.cpp.

**String data\_msg**

Mensaje JSON para transmitir via UART

Definición en la línea 72 del archivo WiMo.cpp.

**char incomingPacket[255]**

Buffer para mensajes de entrada UDP

Definición en la línea 73 del archivo WiMo.cpp.

**char oPack[170]**

Buffer de mensaje de salida sin codificación

Definición en la línea 74 del archivo WiMo.cpp.

**CP\_DEF**

Espacio de memoria de parámetros del nodo.

**Variables**

- uint8\_t **config** [] = {0,0,0,1,3,10,10,VERIFY\_SIGN}
- double **ALPHA** [9] = {0,0,0,0,0,0,0,0,0}
- int16\_t **OFFSETS** [9] = {0,0,0,0,0,0,0,0,0}
- double **GAIN** [9] = {1,1,1,1,1,1,1,1,1}

**Descripción detallada**

Espacio de memoria de parámetros del nodo.

**Nota:**

sobre los vectores de compensación, el orden para cada eje es: 0:acc.x, 1:acc.y, 2:acc.z, 3:gyro.x, 4:gyro.y, 5:gyro.z, 6:mag.x, 7:mag.y, 8:mag.z

**Documentación de las variables****double ALPHA[9] = {0,0,0,0,0,0,0,0,0}**

Valores para compensación térmica

Definición en la línea 102 del archivo WiMo.cpp.

**uint8\_t config[] = {0,0,0,1,3,10,10,VERIFY\_SIGN}**

Vector con valores de configuración del nodo

0: CFG, valor con bits de configuración de funciones

0[bit 0]: Modo de WiFi 0[bit 1]: Transmisión serial

0[bit 2]: Compensación corrimiento

0[bit 3]: Compensación térmica

0[bit 4]: Compensación de ganancia

0[bit 5-7]: Reservado

1: ACC\_SCALE, Escala de ganancia de acelerómetro

2: GYRO\_SCALE, Escala de ganancia de giroscopio

3: MAG\_SCALE, Escala de ganancia de magnetómetro

4: IMU\_DLPF, Ajuste de filtro pasabajas de IMU

5: SWLPPF, Ajuste de filtro pasabajas de software

6: TX\_PERIOD, Periodo de transmisión en milisegundos

7: VERIFY\_SIGN, Valor de verificación de integridad EEPROM

Definición en la línea 87 del archivo WiMo.cpp.

**double GAIN[9] = {1,1,1,1,1,1,1,1,1}**

Valores de ajuste de ganancia

Definición en la línea 104 del archivo WiMo.cpp.

**int16\_t OFFSETS[9] = {0,0,0,0,0,0,0,0,0}**

Valores de compensación de corrimiento

Definición en la línea 103 del archivo WiMo.cpp.

## **SENS\_DEF**

Variables de sensores del nodo.

### **Variables**

- **Vector3D acc**
- **Vector3D gyro**
- **Vector3D mag**
- **double temp**
- **int16\_t acc\_m [3][LPF\_BUFFER\_SIZE] = {{0}}**
- **int16\_t gyro\_m [3][LPF\_BUFFER\_SIZE] = {{0}}**
- **int16\_t mag\_m [3][LPF\_BUFFER\_SIZE] = {{0}}**
- **int16\_t temp\_m [LPF\_BUFFER\_SIZE] = {0}**
- **uint8\_t IMU\_SAMPLE\_COUNTER = 0**
- **uint8\_t MAG\_SAMPLE\_COUNTER = 0**

---

## **Descripción detallada**

Variables de sensores del nodo.

---

---



## Documentación de las variables

### Vector3D acc

Vector de valores de acelerómetro

Definición en la línea 115 del archivo WiMo.cpp.

### int16\_t acc\_m[3][LPF\_BUFFER\_SIZE] = {{0}}

Matrix 3 x LPF\_BUFFER\_SIZE de buffer de acelerómetro

Definición en la línea 119 del archivo WiMo.cpp.

### Vector3D gyro

Vector de valores de giroscopio

Definición en la línea 116 del archivo WiMo.cpp.

### int16\_t gyro\_m[3][LPF\_BUFFER\_SIZE] = {{0}}

Matrix 3 x LPF\_BUFFER\_SIZE de buffer de giroscopio

Definición en la línea 120 del archivo WiMo.cpp.

### uint8\_t IMU\_SAMPLE\_COUNTER = 0

Contador utilitario de muestras de IMU

Definición en la línea 123 del archivo WiMo.cpp.

### Vector3D mag

Vector de valores de magnetómetro

Definición en la línea 117 del archivo WiMo.cpp.

### int16\_t mag\_m[3][LPF\_BUFFER\_SIZE] = {{0}}

Matrix 3 x LPF\_BUFFER\_SIZE de buffer de magnetómetro

Definición en la línea 121 del archivo WiMo.cpp.

### uint8\_t MAG\_SAMPLE\_COUNTER = 0

Contador utilitario de muestras de magnetómetro

Definición en la línea 124 del archivo WiMo.cpp.

### double temp

Valor de temperatura capturado

Definición en la línea 118 del archivo WiMo.cpp.

### int16\_t temp\_m[LPF\_BUFFER\_SIZE] = {0}

Vector de tamaño LPF\_BUFFER\_SIZE de buffer de temperatura

Definición en la línea 122 del archivo WiMo.cpp.

## UTL\_DEF

Variables utilitarias de ejecución.

### Variables

- bool **sensor\_status** = true
  - uint32\_t **adread** = 0
  - uint32\_t **last\_tx** = 0
-

- `uint32_t system_now = 0`
  - `uint16_t led_value = 0`
  - `bool led_increment = true`
  - `bool serial_tx = false`
  - `bool offset_compensation = false`
  - `bool thermal_compensation = false`
  - `bool gain_compensation = false`
- 

## Descripción detallada

Variables utilitarias de ejecución.

---

## Documentación de las variables

### **`uint32_t adread = 0`**

Valor ADC en medición el voltaje de la batería

Definición en la línea 137 del archivo WiMo.cpp.

### **`bool gain_compensation = false`**

Según la configuración CFG, faculta al nodo para compensar la ganancia de sensores true: facultado, false: no facultado

Definición en la línea 150 del archivo WiMo.cpp.

### **`uint32_t last_tx = 0`**

Referencia de tiempo en milisegundos de última transmisión de datos

Definición en la línea 138 del archivo WiMo.cpp.

### **`bool led_increment = true`**

Bandera de control de LED. true: PWM de LED incrementa false: PWM de LED decrementa

Definición en la línea 141 del archivo WiMo.cpp.

### **`uint16_t led_value = 0`**

Valor PWM en LED1

Definición en la línea 140 del archivo WiMo.cpp.

### **`bool offset_compensation = false`**

Según la configuración CFG, activa la compensación de corrimiento(offset) en sensores true: activado, false: desactivado

Definición en la línea 146 del archivo WiMo.cpp.

### **`bool sensor_status = true`**

Bandera de estado de sensores. false: los sensores no funcionan correctamente, true: sensores operan con normalidad

Definición en la línea 134 del archivo WiMo.cpp.

### **`bool serial_tx = false`**

Según la configuración CFG, habilita la transmisión de mensajes JSON por UART true: habilitado, false: deshabilitado

Definición en la línea 144 del archivo WiMo.cpp.

---

**uint32\_t system\_now = 0**

Indicador de tiempo actual en milisegundos  
Definición en la línea 139 del archivo WiMo.cpp.

**bool thermal\_compensation = false**

Según la configuración CFG, permite la compensación térmica de sensores true: permitido, false: no permitido  
Definición en la línea 148 del archivo WiMo.cpp.

**EEPROM\_OP**

Funciones de operación sobre EEPROM.

**Funciones**

- `template<typename RData > void EEPROM_read (uint8_t addr, RData &arg)`  
*Lee un valor entero de cualquier tamaño de bit en EEPROM.*
- `template<typename WData > void EEPROM_write (uint8_t addr, WData &arg)`  
*Escribe un valor entero de cualquier tamaño de bit en EEPROM.*
- `template<typename RData > void EEPROM_readdouble (uint8_t addr, RData &arg)`  
*Lee un valor de punto flotante double de EEPROM.*
- `template<typename WData > void EEPROM_writedouble (uint8_t addr, WData &arg)`  
*Escribe un valor de punto flotante double en EEPROM.*
- `void save_offsets ()`  
*Guarda en bloque los factores de corrimiento en EEPROM.*
- `void save_gain ()`  
*Guarda en bloque los factores de ganancia en EEPROM.*
- `void save_alpha ()`  
*Guarda en bloque los factores de corrección térmica en EEPROM.*
- `void save_config ()`  
*Guarda en bloque los valores de configuración del nodo en EEPROM.*

---

**Descripción detallada**

Funciones de operación sobre EEPROM.  
UTL\_DEF

---

**Documentación de las funciones**

**template<typename RData > void EEPROM\_read (uint8\_t *addr*, RData & *arg*)**

Lee un valor entero de cualquier tamaño de bit en EEPROM.

---

Template Parameters:

<i>RData</i>	Plantilla de tipo de dato
--------------	---------------------------

Parámetros:

<i>addr</i>	Dirección de memoria EEPROM a leer
<i>arg</i>	Valor a escribir en RAM

Definición en la línea 172 del archivo WiMo.cpp.

Gráfico de llamadas a esta función:



`template<typename RData > void EEPROM_readdouble (uint8_t addr, RData & arg)`

Lee un valor de punto flotante double de EEPROM.

Template Parameters:

<i>RData</i>	Plantilla de tipo de dato
--------------	---------------------------

Parámetros:

<i>addr</i>	Dirección de memoria EEPROM a leer
<i>arg</i>	Variable a escribir en RAM

Definición en la línea 212 del archivo WiMo.cpp.

Gráfico de llamadas a esta función:



`template<typename WData > void EEPROM_write (uint8_t addr, WData & arg)`

Escribe un valor entero de cualquier tamaño de bit en EEPROM.

Template Parameters:

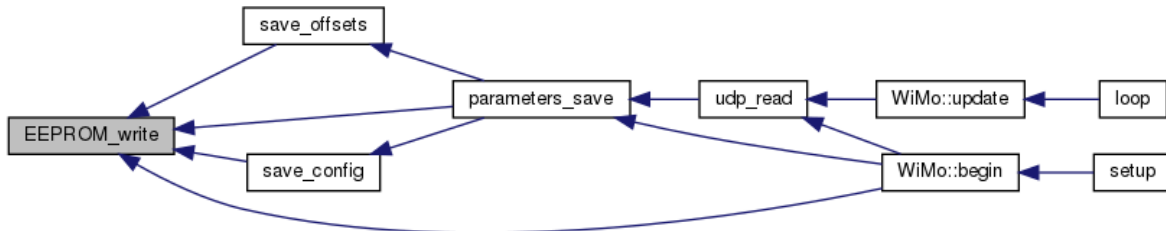
<i>WData</i>	Plantilla de tipo de dato
--------------	---------------------------

Parámetros:

<i>addr</i>	Dirección de memoria EEPROM
<i>arg</i>	Valor a escribir en EEPROM

Definición en la línea 192 del archivo WiMo.cpp.

Gráfico de llamadas a esta función:



**template<typename WData > void EEPROM\_writedouble (uint8\_t *addr*, WData & *arg*)**

Escribe un valor de punto flotante double en EEPROM.

Template Parameters:

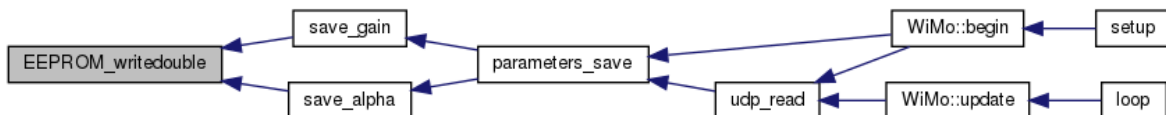
<i>WData</i>	Plantilla de tipo de dato
--------------	---------------------------

Parámetros:

<i>addr</i>	Dirección de memoria EEPROM
<i>arg</i>	Valor a escribir en EEPROM

Definición en la línea 232 del archivo WiMo.cpp.

Gráfico de llamadas a esta función:



**void save\_alpha ()**

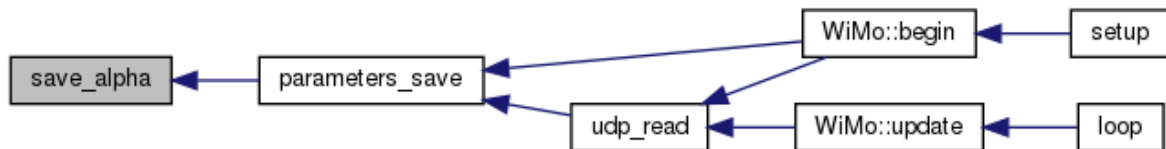
Guarda en bloque los factores de corrección térmica en EEPROM.

Definición en la línea 268 del archivo WiMo.cpp.

Gráfico de llamadas para esta función:



Gráfico de llamadas a esta función:



**void save\_config ()**

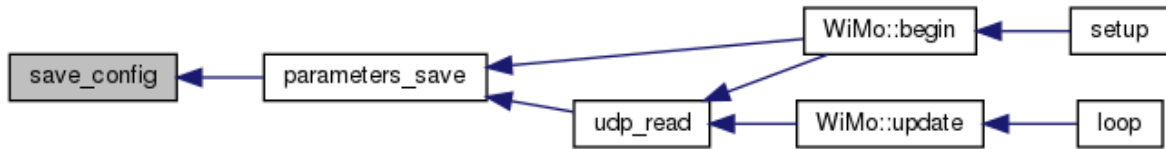
Guarda en bloque los valores de configuración del nodo en EEPROM.

Definición en la línea 278 del archivo WiMo.cpp.

Gráfico de llamadas para esta función:



Gráfico de llamadas a esta función:



### void save\_gain ()

Guarda en bloque los factores de ganancia en EEPROM.

Definición en la línea 258 del archivo WiMo.cpp.

Gráfico de llamadas para esta función:



Gráfico de llamadas a esta función:



### void save\_offsets ()

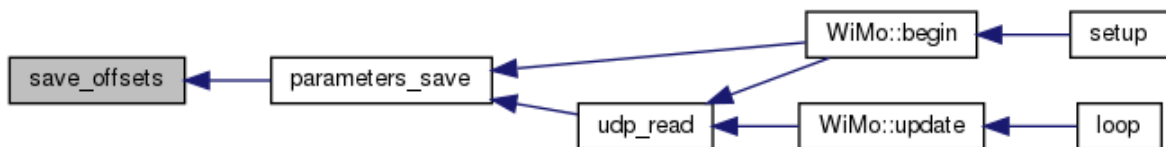
Guarda en bloque los factores de corrimiento en EEPROM.

Definición en la línea 248 del archivo WiMo.cpp.

Gráfico de llamadas para esta función:



Gráfico de llamadas a esta función:



## UTILS

Funciones utilitarias.

### Funciones

- float **average** (int16\_t \*vec)  
*Calcula el promedio de un vector.*

- void **error\_indicator** ()  
*Rutina de LED para indicar error.*
- void **ok\_indicator** ()  
*Rutina de LED para indicar éxito en una operación.*
- uint8\_t **mode\_selector** ()  
*Rutina para determinar un valor de modo a través del botón de usuario.*
- void **parameters\_save** ()  
*Actualiza todos los valores de configuración en EEPROM.*
- uint8\_t **config\_read** (uint8\_t bit)  
*Lee bit de configuración del registro CFG de la variable config.*
- void **config\_write** (uint8\_t bit, uint8\_t value)  
*Escribe bit de configuración en registro CFG de la variable config.*
- void **imu\_setup** ()  
*Actualiza registros de configuración en MPU6050(IMU) y HMC5883L(Magnetómetro)*

## Descripción detallada

Funciones utilitarias.

EEPROM\_OP

## Documentación de las funciones

### float average (int16\_t \* vec)

Calcula el promedio de un vector.

Parámetros:

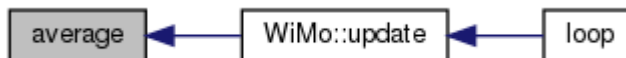
<i>vec</i>	Vector a leer
------------	---------------

Devuelve:

float Promedio calculado

Definición en la línea 302 del archivo WiMo.cpp.

Gráfico de llamadas a esta función:



### uint8\_t config\_read (uint8\_t bit)

Lee bit de configuración del registro CFG de la variable config.

Parámetros:

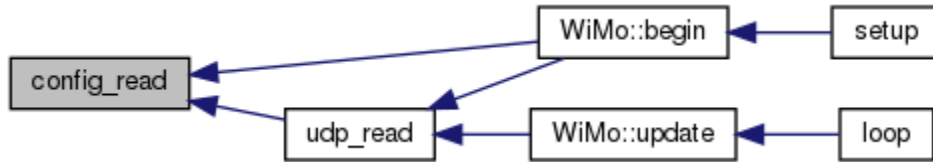
<i>bit</i>	Posición de bit a leer
------------	------------------------

**Devuelve:**

uint8\_t Valor obtenido

Definición en la línea 381 del archivo WiMo.cpp.

Gráfico de llamadas a esta función:



**void config\_write (uint8\_t bit, uint8\_t value)**

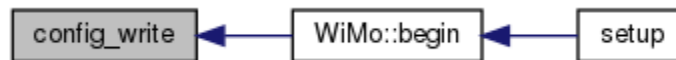
Escribe bit de configuración en registro CFG de la variable config.

**Parámetros:**

<i>bit</i>	Posición de bit a escribir
<i>value</i>	Valor lógico a escribir

Definición en la línea 391 del archivo WiMo.cpp.

Gráfico de llamadas a esta función:

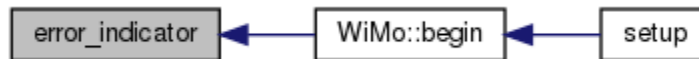


**void error\_indicator ()**

Rutina de LED para indicar error.

Definición en la línea 314 del archivo WiMo.cpp.

Gráfico de llamadas a esta función:

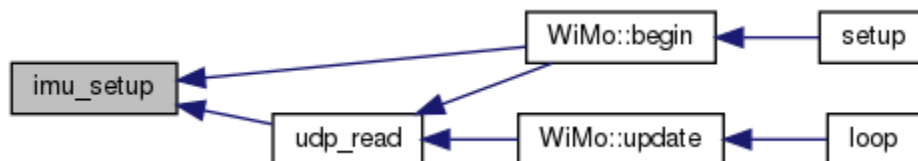


**void imu\_setup ()**

Actualiza registros de configuración en MPU6050(IMU) y HMC5883L(Magnetómetro)

Definición en la línea 402 del archivo WiMo.cpp.

Gráfico de llamadas a esta función:



**uint8\_t mode\_selector ()**

Rutina para determinar un valor de modo a través del botón de usuario.

---

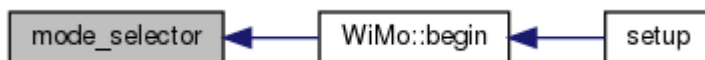


**Devuelve:**

uint8\_t valor seleccionado

Definición en la línea 342 del archivo WiMo.cpp.

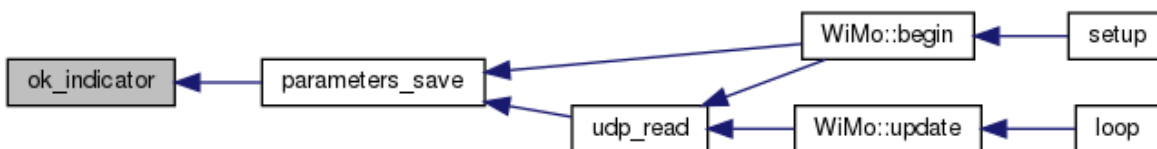
Gráfico de llamadas a esta función:

**void ok\_indicador ()**

Rutina de LED para indicar éxito en una operación.

Definición en la línea 328 del archivo WiMo.cpp.

Gráfico de llamadas a esta función:

**void parameters\_save ()**

Actualiza todos los valores de configuración en EEPROM.

**Nota:**

La firma de integridad de EEPROM se cambia al valor 0xBB durante el proceso, si no se completa el proceso al reiniciar se restauran los valores de configuración por defecto

Definición en la línea 362 del archivo WiMo.cpp.

Gráfico de llamadas para esta función:

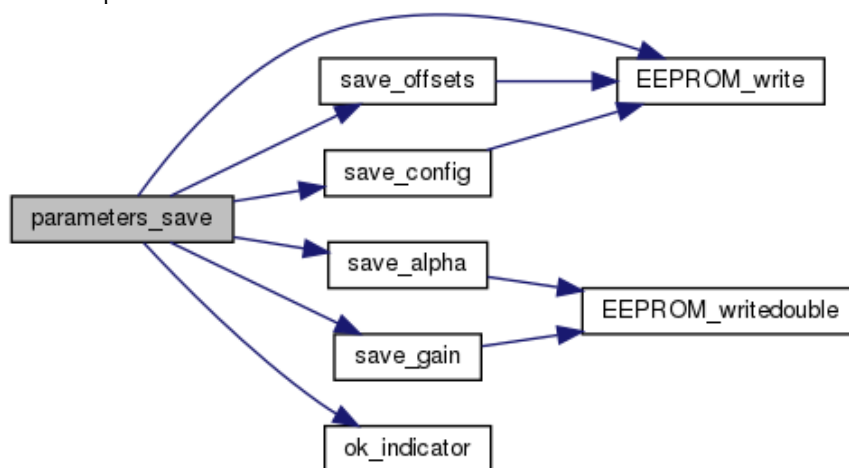
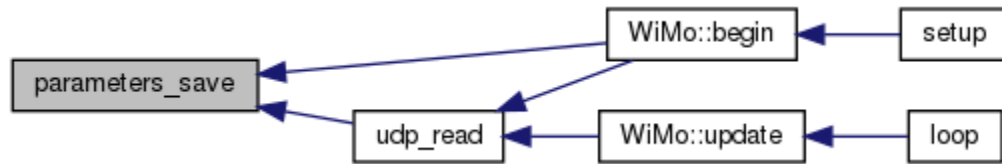


Gráfico de llamadas a esta función:



## COMM\_OP

Funciones de comunicación.

### Funciones

- void **send\_parameters** ()  
*Construcción y envío de parámetros del nodo.*
- void **udp\_read** ()  
*Decodifica un mensaje UDP recibido.*
- void **send\_json\_sensors** ()  
*Cuando está habilitada la transmisión serial, se construye un mensaje JSON con la información de sensores.*
- void **send\_sensors** ()  
*Construye un mensaje UDP con la información de los sensores.*

---

## Descripción detallada

Funciones de comunicación.

---

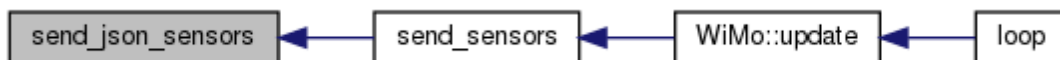
## Documentación de las funciones

### void send\_json\_sensors ()

Cuando está habilitada la transmisión serial, se construye un mensaje JSON con la información de sensores.

Definición en la línea 567 del archivo `WiMo.cpp`.

Gráfico de llamadas a esta función:



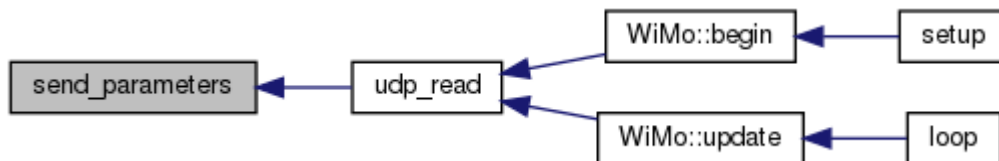
### void send\_parameters ()

Construcción y envío de parámetros del nodo.

Definición en la línea 428 del archivo `WiMo.cpp`.

---

Gráfico de llamadas a esta función:



### void send\_sensors ()

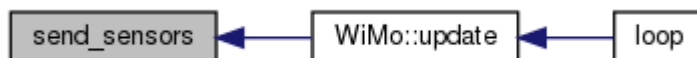
Construye un mensaje UDP con la información de los sensores.

Definición en la línea 606 del archivo WiMo.cpp.

Gráfico de llamadas para esta función:



Gráfico de llamadas a esta función:



### void udp\_read ()

Decodifica un mensaje UDP recibido.

Nota:

los parámetros recibidos tienen la estructura: Byte [0]:Tipo, Byte [1]:Dirección, Byte[2..5]:nuevo valor

Definición en la línea 483 del archivo WiMo.cpp.

Gráfico de llamadas para esta función:

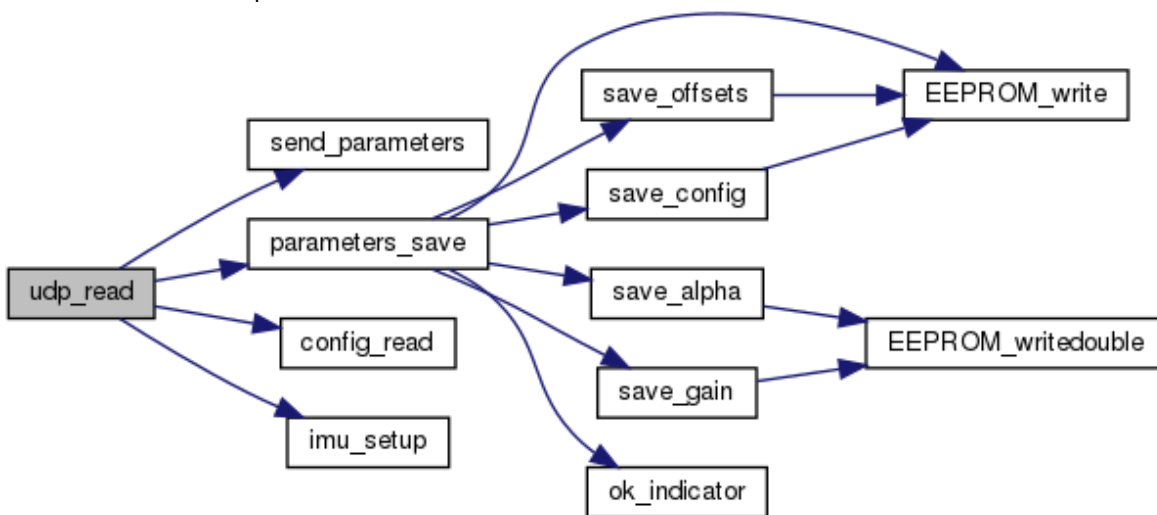
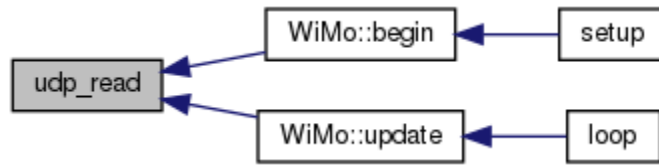


Gráfico de llamadas a esta función:



## D – SCRIPT DE CAPTURA DE DATOS “Capturar.py”

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import time
import base64
import struct
import csv
from socket import *
import datetime
import os
import argparse as arg

parser = arg.ArgumentParser(description='Captura datos de un nodo de sensores WiMo.',
                             prog='Capturar',
                             formatter_class=arg.ArgumentDefaultsHelpFormatter)
parser.add_argument("-a", "--address", nargs='?', default="192.168.181.180", help="Dirección IP del
nodo")
parser.add_argument("-p", "--port", nargs='*', default=42100, help="Puerto UDP del nodo")

args = parser.parse_args()
print (args)

#Configuración de conexión con el nodo de sensores
WiMo_IP = str(args.address) #Dirección IP
WiMo_PORT = int(args.port) #Puerto UDP asignado en el nodo

os.system("cls");
clientSocket = socket(AF_INET, SOCK_DGRAM)
clientSocket.settimeout(1)
message = 'hola wimo'
addr = (WiMo_IP, WiMo_PORT)
utc_datetime = datetime.datetime.utcnow()
filename = "WiMo_STD_" + utc_datetime.strftime("%d_%b_%Y_%H%M%S") + "_GMT.csv"
print "Conectando con el nodo en la dirección: "+str(WiMo_IP)+":"+str(WiMo_PORT)
print "Las muestras se guardarán en el archivo: "+filename
muestras = 0;
connected = False;

def convertMillis(millis):
    seconds=(millis/1000)%60
    minutes=(millis/(1000*60))%60
    hours=(millis/(1000*60*60))%24
    return seconds, minutes, hours

csv_salida = open(filename, "wb")
writer = csv.writer(csv_salida, delimiter=',')

for x in range(0,5):
    clientSocket.sendto(message, addr) #Mensaje de saludo al nodo para iniciar la transmisión

while True:
    try:
        data, server = clientSocket.recvfrom(255)
        data = base64.b64decode(data[:-1])
        if data[0] == '*': #Recepción de paquete con datos de sensores
            if not connected:
                os.system("cls")
                print "Conectado con el nodo en la dirección: "+str(WiMo_IP)+":"+str(WiMo_PORT)
                print "Tiempo encendido\tMuestras"
                connected = True;
            muestras+=1;
            wimo_data= [
                struct.unpack("<L", str(data[1:4+1]))[0], #Tiempo [uint32_t]
                struct.unpack("<d", str(data[9:16+1]))[0], #Temperatura [long]
```

```

struct.unpack("<d", str(data[17:24+1]))[0], #Acelerómetro X [long]
struct.unpack("<d", str(data[25:32+1]))[0], #Acelerómetro Y [long]
struct.unpack("<d", str(data[33:40+1]))[0], #Acelerómetro Z [long]
struct.unpack("<d", str(data[41:48+1]))[0], #Giroscopio X [long]
struct.unpack("<d", str(data[49:56+1]))[0], #Giroscopio Y [long]
struct.unpack("<d", str(data[57:64+1]))[0], #Giroscopio Z [long]
struct.unpack("<d", str(data[65:72+1]))[0], #Magnetómetro X [long]
struct.unpack("<d", str(data[73:80+1]))[0], #Magnetómetro Y [long]
struct.unpack("<d", str(data[81:88+1]))[0] ] #Magnetómetro Z [long]
con_sec, con_min, con_hour = convertMillis(int(wimo_data[0])) #Conversión de tiempo de
milisegundos a formato horas, minutos, segundos
wimo_time = ("{:0}:{1}:{2}".format(con_hour, con_min, con_sec))
print wimo_time+"\t\t\t"+str(muestras)," \r", #Muestra en consola de muestras y
tiempo de encendido en el nodo
writer.writerow(wimo_data) #Escritura de los valores recibidos en el archivo

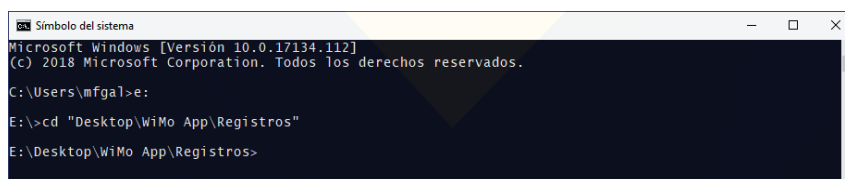
except timeout:
    print 'No es posible conectarse con WiMo'
    clientSocket.sendto(message, addr)

```

*Capturar.py* es un script de Python que permite recibir los datos de un nodo de sensores y guardarlos en un archivo CSV; para luego realizar cálculos ayudado de Matlab u otra herramienta de cómputo. Para su utilización es obligatorio tener instalado Python 2.x.

### Modo de empleo:

Abra la terminal de su sistema y monte la carpeta donde desea almacenar los archivos de muestras con el comando *cd*.



```

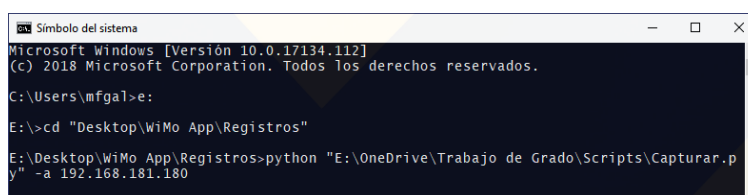
Microsoft Windows [Versión 10.0.17134.112]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\mfga>e:
E:\>cd "Desktop\WiMo App\Registros"
E:\Desktop\WiMo App\Registros>

```

Luego llame el script y adicione los parámetros de dirección IP y puerto(opcional):

```
python Capturar.py -a 192.168.181.180 -p 42100
```



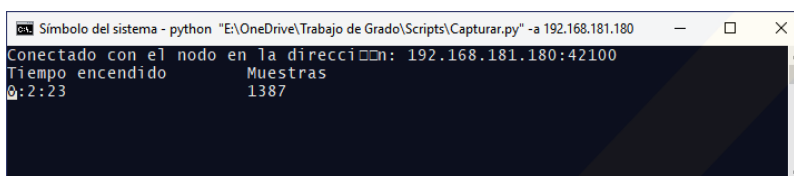
```

Microsoft Windows [Versión 10.0.17134.112]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\mfga>e:
E:\>cd "Desktop\WiMo App\Registros"
E:\Desktop\WiMo App\Registros>python "E:\OneDrive\Trabajo de Grado\Scripts\Capturar.py" -a 192.168.181.180

```

Si el nodo está operando, la conexión de red es correcta y la dirección IP señalada es la correspondiente, se observará lo siguiente:



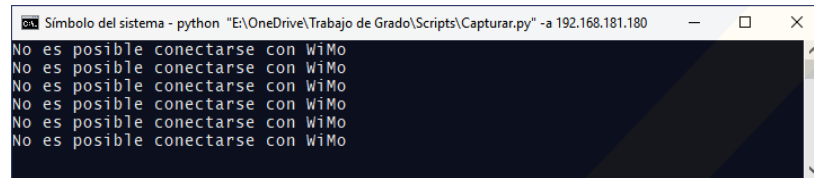
```

Microsoft Windows [Versión 10.0.17134.112]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\mfga>e:
E:\>cd "Desktop\WiMo App\Registros"
E:\Desktop\WiMo App\Registros>python "E:\OneDrive\Trabajo de Grado\Scripts\Capturar.py" -a 192.168.181.180
Conectado con el nodo en la dirección: 192.168.181.180:42100
Tiempo encendido Muestras
0:2:23 1387

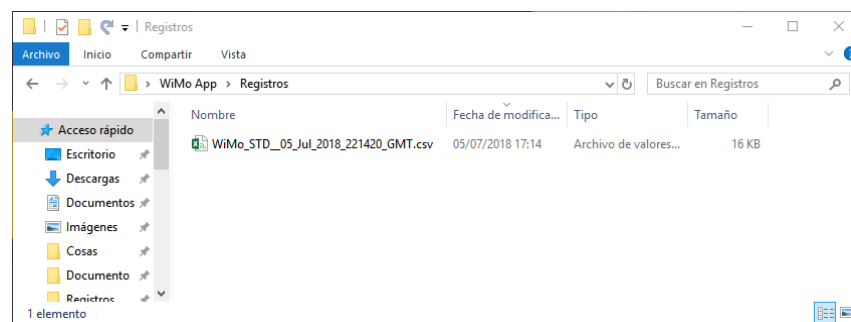
```

Si no es posible establecer la conexión se observará el mensaje “No es posible conectarse con WiMo”, se intentará nuevamente crear la conexión repetitivamente en caso de que el nodo esté apagado o la red desconfigurada.



```
Símbolo del sistema - python "E:\OneDrive\Trabajo de Grado\Scripts\Capturar.py" -a 192.168.181.180
No es posible conectarse con WiMo
No es posible conectarse con WiMo
No es posible conectarse con WiMo
No es posible conectarse con WiMo
No es posible conectarse con WiMo
No es posible conectarse con WiMo
```

Cuando finalice, termine la aplicación cerrando la terminal o matando la tarea, el archivo con los datos aparecerá en la ruta que fue seleccionada previamente:



## E – SCRIPT DE CORRECCIÓN TÉRMICA “thermal\_analysis.m”

```

clear all;

choice = questdlg('Desea mantener los gráficos de una sesión anterior?', ...
    'Visor de datos', ...
    'Si','No','No');
if choice == 'No'
    clc;
    close all;
end

fprintf("\r\n*****\r\n");
[fname, pname] = uigetfile('*.csv', 'Seleccione un archivo de registro WiMo', '/');
log_file = fullfile(pname, fname);
M = csvread(log_file,1,0);

dstr = strsplit(fname, '_');
fprintf("Fecha de la muestra:\t%s %s %s | %c%c:%c%c:%c%c
GMT\r\n",dstr{3},dstr{4},dstr{5},dstr{6}(1),dstr{6}(2),dstr{6}(3),dstr{6}(4),dstr{6}(5),d
str{6}(6));

switch dstr{2}
    case "THR"
        fprintf("Prueba de compensación térmica\r\n");
        compute_alpha = true;
    case "THX"
        fprintf("Prueba de mejora de compensación térmica\r\n");
        compute_alpha = true;
    case "STD"
        fprintf("Prueba tipo estándar\r\n");
        compute_alpha = true;
        choice = questdlg('Está seguro que estos datos son de una prueba térmica?',...
            'Pregunta',...
            'Si','No','No');
        if choice == 'No'
            clc; close all;
            compute_alpha=false;
        end
    otherwise
        fprintf("Prueba tipo no reconocido\r\n");
        clc; close all;
        compute_alpha=false;
end

plot_histogram = false;

choice = questdlg('Desea generar histogramas?',...
    'Pregunta',...
    'Si','No','No');
if choice == 'Si'
    plot_histogram = true;
end

if compute_alpha
    x_data=M(:,1)/1000;
    temp_data = M(:,2);
    p = polyfit(x_data,temp_data,2);

```



```

temp_tend = polyfit(x_data,temp_data,1);
x = linspace(min(x_data),max(x_data),101);
ypt = polyval(temp_tend,x);

%Accelerometer
figure('Name','Accelerometer');
subplot(3,2,1);
y_data=M(:,3);
p = polyfit(x_data,y_data,1);
accx_tend = polyfit(x_data, y_data,1);
y = polyval(p,x);
plot (x_data,y_data,'b-',x, y, 'r-')
title('ACC X')

subplot(3,2,2);
accx_alpha=(accx_tend(1)/temp_tend(1));
y_data=y_data-(accx_alpha*temp_data);
pc = polyfit(x_data,y_data,1);
yc = polyval(pc,x);
plot (x_data,y_data,'b-',x, y, 'r-',x,yc,'g-');
fprintf("ACC X Comp Tend:\t%.20g\r\n",median(yc));
title('ACC X Compensated');

subplot(3,2,3);
y_data=M(:,4);
accy_tend = polyfit(x_data, y_data,1);
p = polyfit(x_data,y_data,1);
y = polyval(p,x);
plot (x_data,y_data,'b-',x, y, 'r-')
title('ACC Y')

subplot(3,2,4);
accy_alpha=(accy_tend(1)/temp_tend(1));
y_data=y_data-(accy_alpha*temp_data);
pc = polyfit(x_data,y_data,1);
yc = polyval(pc,x);
plot (x_data,y_data,'b-',x, y, 'r-',x,yc,'g-');
fprintf("ACC Y Comp Tend:\t%.20g\r\n",median(yc));
title('ACC Y Compensated');

subplot(3,2,5);          % add second plot in 2 x 1 grid
y_data=M(:,5);
accz_tend = polyfit(x_data, y_data,1);
p = polyfit(x_data,y_data,1);
y = polyval(p,x);
plot (x_data,y_data,'b-',x, y, 'r-')
title('ACC Z')

subplot(3,2,6)
accz_alpha=(accz_tend(1)/temp_tend(1));
y_data=y_data-(accz_alpha*temp_data);
pc = polyfit(x_data,y_data,1);
yc = polyval(pc,x);
plot (x_data,y_data,'b-',x, y, 'r-',x,yc,'g-');
fprintf("ACC Z Comp Tend:\t%.20g\r\n",median(yc));
title('ACC Z Compensated');

%Gyroscope

```

```
figure('Name', 'Gyroscope');
subplot(3,2,1);
y_data=M(:,6);
gyrox_tend = polyfit(x_data, y_data,1);
p = polyfit(x_data,y_data,1);
y = polyval(p,x);
plot (x_data,y_data,'b-',x, y, 'r-') ;
title('GYRO X');

subplot(3,2,2);
gyrox_alpha=(gyrox_tend(1)/temp_tend(1));
y_data=y_data-(gyrox_alpha*temp_data);
pc = polyfit(x_data,y_data,1);
yc = polyval(pc,x);
plot (x_data,y_data,'b-',x, y, 'r-',x,yc,'g-');
fprintf("GYRO X Comp Tend:\t%.20g\r\n",median(yc));
title('GYRO X Compensated');

subplot(3,2,3);
y_data=M(:,7);
gyroy_tend = polyfit(x_data, y_data,1);
p = polyfit(x_data,y_data,1);
y = polyval(p,x);
plot (x_data,y_data,'b-',x, y, 'r-');
title('GYRO Y');

subplot(3,2,4);
gyroy_alpha=(gyroy_tend(1)/temp_tend(1));
y_data=y_data-(gyroy_alpha*temp_data);
pc = polyfit(x_data,y_data,1);
yc = polyval(pc,x);
plot (x_data,y_data,'b-',x, y, 'r-',x,yc,'g-');
fprintf("GYRO Y Comp Tend:\t%.20g\r\n",median(yc));
title('GYRO Y Compensated');

subplot(3,2,5);          % add second plot in 2 x 1 grid
y_data=M(:,8);
gyroz_tend = polyfit(x_data, y_data,1);
p = polyfit(x_data,y_data,1);
y = polyval(p,x);
plot (x_data,y_data,'b-',x, y, 'r-');
title('GYRO Z');

subplot(3,2,6);
gyroz_alpha=(gyroz_tend(1)/temp_tend(1));
y_data=y_data-(gyroz_alpha*temp_data);
pc = polyfit(x_data,y_data,1);
yc = polyval(pc,x);
plot (x_data,y_data,'b-',x, y, 'r-',x,yc,'g-');
fprintf("GYRO Z Comp Tend:\t%.20g\r\n",median(yc));
title('GYRO Z Compensated');

%Magnetometer
figure('Name', 'Magnetometer');
subplot(3,2,1);
y_data=M(:,9);
magx_tend = polyfit(x_data, y_data,1);
p = polyfit(x_data,y_data,1);
y = polyval(p,x);
```

```

plot (x_data,y_data,'b-',x, y, 'r-');
title('MAG X');

subplot(3,2,2);
magx_alpha=(magx_tend(1)/temp_tend(1));
y_data=y_data-(magx_alpha*temp_data);
pc = polyfit(x_data,y_data,1);
yc = polyval(pc,x);
plot (x_data,y_data,'b-',x, y, 'r-',x,yc,'g-');
fprintf("MAG X Comp Tend:\t%.20g\r\n",median(yc));
title('MAG X Compensated');

subplot(3,2,3);
y_data=M(:,10);
magy_tend = polyfit(x_data, y_data,1);
p = polyfit(x_data,y_data,1);
y = polyval(p,x);
plot (x_data,y_data,'b-',x, y, 'r-');
title('MAG Y');

subplot(3,2,4);
magy_alpha=(magy_tend(1)/temp_tend(1));
y_data=y_data-(magy_alpha*temp_data);
pc = polyfit(x_data,y_data,1);
yc = polyval(pc,x);
plot (x_data,y_data,'b-',x, y, 'r-',x,yc,'g-');
fprintf("MAG Y Comp Tend:\t%.20g\r\n",median(yc));
title('MAG Y Compensated');

subplot(3,2,5);          % add second plot in 2 x 1 grid
y_data=M(:,11);
magz_tend = polyfit(x_data, y_data,1);
p = polyfit(x_data,y_data,1);
y = polyval(p,x);
plot (x_data,y_data,'b-',x, y, 'r-');
title('MAG Z');

subplot(3,2,6);
magz_alpha=(magz_tend(1)/temp_tend(1));
y_data=y_data-(magz_alpha*temp_data);
pc = polyfit(x_data,y_data,1);
yc = polyval(pc,x);
plot (x_data,y_data,'b-',x, y, 'r-',x,yc,'g-');
fprintf("MAG Z Comp Tend:\t%.20g\r\n",median(yc));
title('MAG Z Compensated');

if plot_histogram
    fprintf("\r\n*****\r\n");
    %Accelerometer histograms
    figure('Name','Accelerometer Histogram');
    subplot(3,2,1);
    y_data=M(:,3);
    histogram(y_data);
    fprintf("ACC X SNR:\t%.20g\r\n",snr(y_data));
    title('ACC X')

    subplot(3,2,2);
    y_data=y_data-(accx_alpha*temp_data);
    histogram(y_data);

```

```
fprintf("ACC X Comp SNR:\t%.20g\r\n", snr(y_data));
title('ACC X Compensated');

subplot(3,2,3);
y_data=M(:,4);
histogram(y_data);
fprintf("ACC Y SNR:\t%.20g\r\n", snr(y_data));
title('ACC Y')

subplot(3,2,4);
y_data=y_data-(accy_alpha*temp_data);
histogram(y_data);
fprintf("ACC Y Comp SNR:\t%.20g\r\n", snr(y_data));
title('ACC Y Compensated');

subplot(3,2,5);          % add second plot in 2 x 1 grid
y_data=M(:,5);
histogram(y_data);
fprintf("ACC Z SNR:\t%.20g\r\n", snr(y_data));
title('ACC Z')

subplot(3,2,6)
y_data=y_data-(accz_alpha*temp_data);
histogram(y_data);
fprintf("ACC Z Comp SNR:\t%.20g\r\n", snr(y_data));
title('ACC Z Compensated');

%Gyroscope histograms
figure('Name', 'Gyroscope Histogram');
subplot(3,2,1);
y_data=M(:,6);
histogram(y_data);
fprintf("GYRO X SNR:\t%.20g\r\n", snr(y_data));
title('GYRO X')

subplot(3,2,2);
y_data=y_data-(gyrox_alpha*temp_data);
histogram(y_data);
fprintf("GYRO X Comp SNR:\t%.20g\r\n", snr(y_data));
title('GYRO X Compensated');

subplot(3,2,3);
y_data=M(:,7);
histogram(y_data);
fprintf("GYRO Y SNR:\t%.20g\r\n", snr(y_data));
title('GYRO Y')

subplot(3,2,4);
y_data=y_data-(gyroy_alpha*temp_data);
histogram(y_data);
fprintf("GYRO Y Comp SNR:\t%.20g\r\n", snr(y_data));
title('GYRO Y Compensated');

subplot(3,2,5);          % add second plot in 2 x 1 grid
y_data=M(:,8);
histogram(y_data);
fprintf("GYRO Z SNR:\t%.20g\r\n", snr(y_data));
title('GYRO Z')
```

```

subplot(3,2,6)
y_data=y_data-(gyroz_alpha*temp_data);
histogram(y_data);
fprintf("GYRO Z Comp SNR:\t%.20g\r\n",snr(y_data));
title('GYRO Z Compensated');

%Magnetometer histograms
figure('Name','Magnetometer Histogram');
subplot(3,2,1);
y_data=M(:,9);
histogram(y_data);
fprintf("MAG X SNR:\t%.20g\r\n",snr(y_data));
title('MAG X')

subplot(3,2,2);
y_data=y_data-(magx_alpha*temp_data);
histogram(y_data);
fprintf("MAG X Comp SNR:\t%.20g\r\n",snr(y_data));
title('MAG X Compensated');

subplot(3,2,3);
y_data=M(:,10);
histogram(y_data);
fprintf("MAG Y SNR:\t%.20g\r\n",snr(y_data));
title('MAG Y')

subplot(3,2,4);
y_data=y_data-(magy_alpha*temp_data);
histogram(y_data);
fprintf("MAG Y Comp SNR:\t%.20g\r\n",snr(y_data));
title('MAG Y Compensated');

subplot(3,2,5);          % add second plot in 2 x 1 grid
y_data=M(:,11);
histogram(y_data);
fprintf("MAG Z SNR:\t%.20g\r\n",snr(y_data));
title('MAG Z')

subplot(3,2,6)
y_data=y_data-(magz_alpha*temp_data);
histogram(y_data);
fprintf("MAG Z Comp SNR:\t%.20g\r\n",snr(y_data));
title('MAG Z Compensated');

end

temp_data = temp_data/340 + 35.63;
ypt = ypt/340 + 35.63;

figure('Name','Temperature');
plot(x_data,temp_data,'b-',x,ypt,'r-');
legend('Original Data','Fitted polynomial');
title('TEMP');

fprintf("\r\n*****\r\n");
fprintf("FACTORES ALFA\r\n");
fprintf("ACC X:\t%.20g\r\n",accx_alpha);
fprintf("ACC Y:\t%.20g\r\n",accy_alpha);
fprintf("ACC Z:\t%.20g\r\n\r\n",accz_alpha);

```

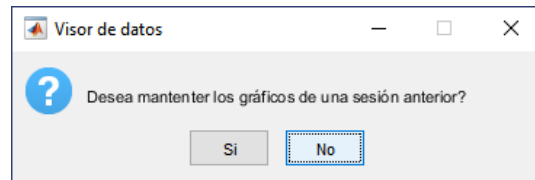
```
fprintf("GYRO X:\t.20g\r\n",gyrox_alpha);
fprintf("GYRO Y:\t.20g\r\n",gyroy_alpha);
fprintf("GYRO Z:\t.20g\r\n\r\n",gyroz_alpha);

fprintf("MAG X:\t.20g\r\n",magx_alpha);
fprintf("MAG Y:\t.20g\r\n",magy_alpha);
fprintf("MAG Z:\t.20g\r\n",magz_alpha);

fprintf("\r\n*****\r\n");
end
```

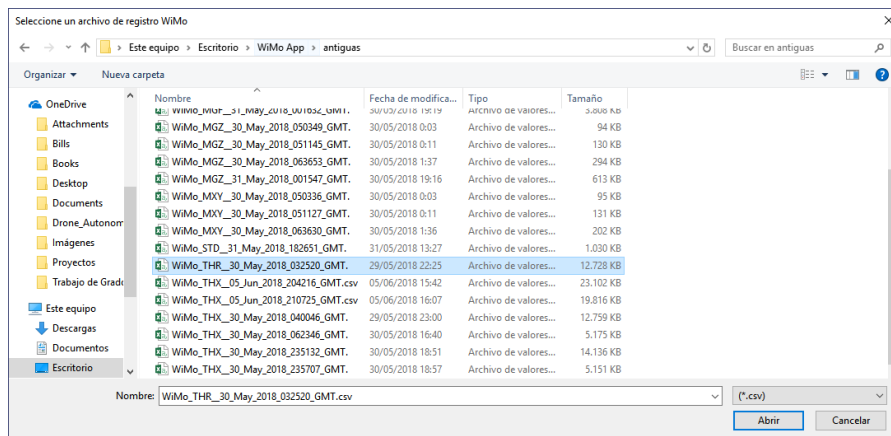
Este script de Matlab emplea datos de los sensores para calcular los factores lineales de corrección térmica, es necesario tomar muestras con el dispositivo en un entorno libre de ruido, completamente estático y con temperatura constante o controlada, lo deseable es que la temperatura cambie lentamente en aumento o decremento.

Al ejecutarlo se le preguntará al usuario si desea mantener gráficos de una sesión anterior, esto puede resultar útil para ejecutar comparativas de gráficas entre dos procesos de captura distintos.



Si responde que sí, las figuras en pantalla se mantendrán, en caso contrario serán cerradas. Si el equipo empleado cuenta con una baja cantidad de memoria RAM es recomendable no mantener más figuras en pantalla.

Luego, se abrirá una ventana de exploración para seleccionar un archivo CSV con los datos de muestra, si estos datos provienen de la aplicación Wimo solo son aceptados aquellos archivos que tengan en su nombre de archivo los prefijos STD (captura estándar), THR (sesión de cálculo de parámetros para compensación térmica) y THX (sesión de mejora de parámetros de compensación térmica).



Seleccione el archivo de interés y presione “abrir”

Posteriormente se preguntará si se desean generar histogramas, seleccione si o no según la necesidad. Los histogramas son particularmente útiles para detectar fuentes de distorsión o ruido, además de evaluar la calidad de los parámetros de corrección térmica.

Con la pregunta anterior resuelta, se realizarán los cálculos y gráficas, en la ventana de comandos de Matlab se mostrará la siguiente información:

```
*****
```

```
Fecha de la muestra:    30 May 2018   |   03:25:20 GMT
Prueba de compensación térmica
```

```
ACC X Comp Tend:  -56.944405866692883
ACC Y Comp Tend:  12.415895459516397
ACC Z Comp Tend:  749.21957664346621
GYRO X Comp Tend: -252.95706302973292
GYRO Y Comp Tend: 20.603877952385435
GYRO Z Comp Tend: -26.401721685366191
MAG X Comp Tend:  -44.256160707800802
MAG Y Comp Tend:  -15.298263038867612
MAG Z Comp Tend:  -77.583699391631285
```

```
*****
```

Lo anterior corresponde al factor de pendiente (inclinación) obtenido de la linealización de las curvas en cada eje de los sensores. Luego se calculan las relaciones de señal-ruido.

```
*****
```

```
ACC X SNR:  3.6876534764701057
ACC X Comp SNR:  -11.428027992568836
ACC Y SNR:  -9.4672707604109281
ACC Y Comp SNR:  -9.4712087825540721
ACC Z SNR:  20.524526862514755
ACC Z Comp SNR:  -9.9420451013631421
GYRO X SNR:  26.052278504949978
GYRO X Comp SNR:  -9.3642530818766438
GYRO Y SNR:  -2.3361755428113593
GYRO Y Comp SNR:  -2.3324292963052109
GYRO Z SNR:  -11.701335627762386
GYRO Z Comp SNR:  -11.706425360630242
MAG X SNR:  5.0508928905654606
MAG X Comp SNR:  -10.007435153115622
MAG Y SNR:  1.3754604713019294
MAG Y Comp SNR:  -11.322977815913699
MAG Z SNR:  11.293507084274664
MAG Z Comp SNR:  -9.5636676024186791
```

```
*****
```

Los valores SNR son obtenidos con la función `snr` de Matlab, generados mediante un proceso estadístico debido a la falta de información de la señal de origen o patrón. Los valores Comp

SNR son el resultado de calcular la relación de señal-ruído después de aplicar la compensación térmica. Finalmente, los factores de compensación térmica (alpha) son mostrados, para calcularlos se utilizan los valores de pendiente de cada eje divididos por la pendiente de la temperatura.

\*\*\*\*\*

FACTORES ALFA

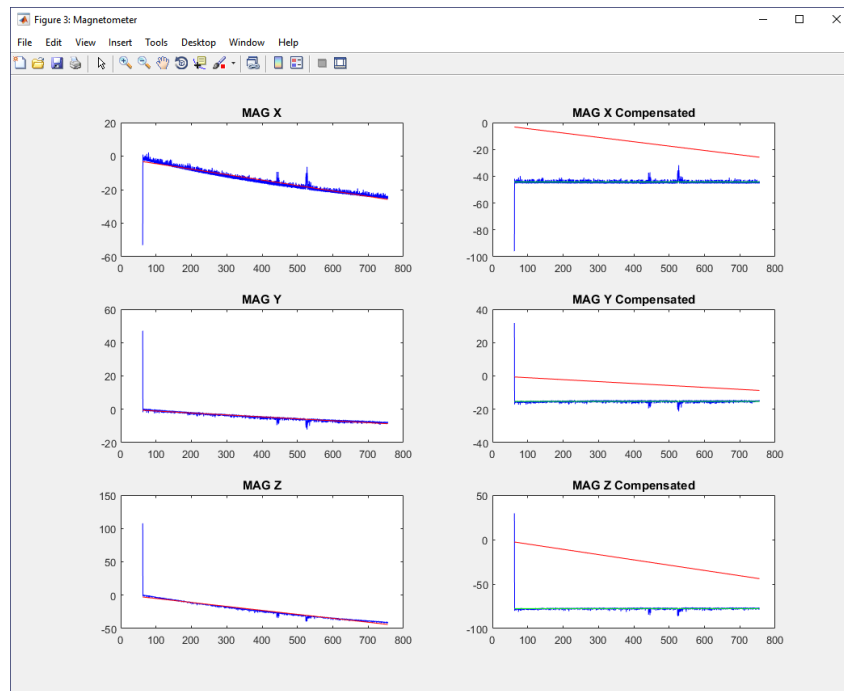
ACC X: -0.0053285595805352944  
 ACC Y: 0.001395707375463569  
 ACC Z: 0.068166349049498823

GYRO X: -0.021252230691863789  
 GYRO Y: 0.0012329170592637151  
 GYRO Z: -0.0021839741657188392

MAG X: -0.0035448641353923518  
 MAG Y: -0.0012704762489802164  
 MAG Z: -0.0064769547712882243

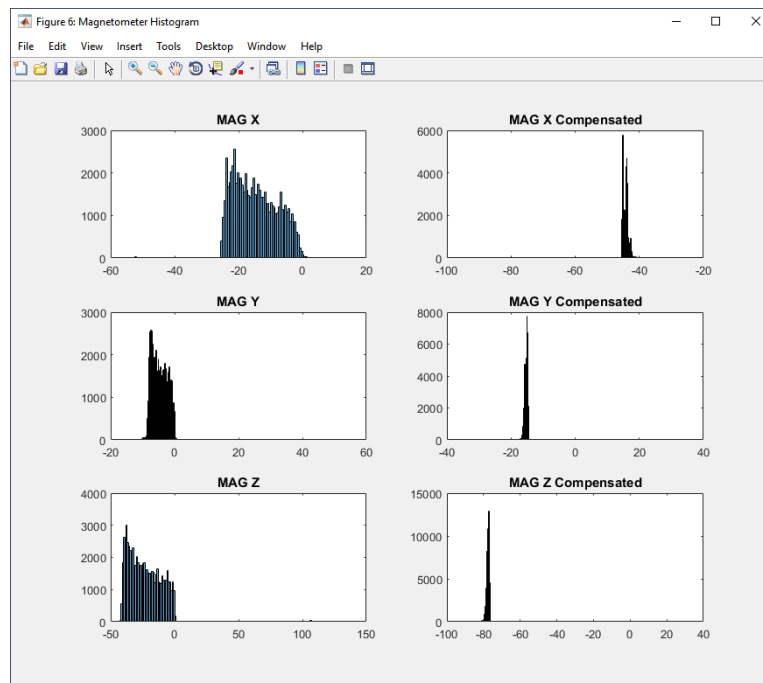
\*\*\*\*\*

Respecto a las gráficas se abrirá una por cada sensor, en la columna izquierda se muestran los datos en bruto junto a las pendientes como una línea roja. En la otra columna son expuestos los datos de sensores compensados térmicamente con dos líneas de pendiente, una roja que muestra la pendiente original y una verde con la nueva pendiente después de la compensación.

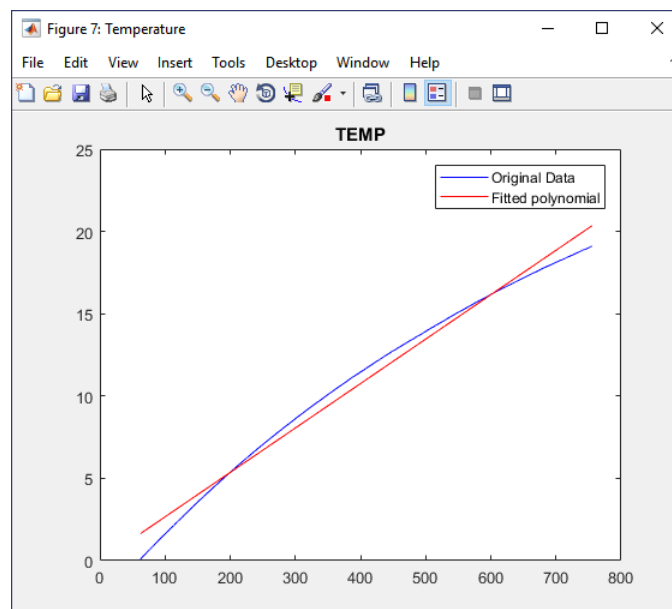




Si ha sido activada la generación de histogramas, son presentados en un formato similar a los datos de señal.



Finalmente son presentados los datos de temperatura con una aproximación a los valores en grados Celsius, recuerde que este sensor no cuenta con calibración.



Nótese como la temperatura cambió de manera sostenida producto del calentamiento de los componentes debido a la actividad de sí mismos, esto resulta útil en este proceso.

## F – SCRIPT DE CÁLCULO DE PARÁMETROS “ellipsoids.m”

```

clear all, close all;

[settings, button] = settingsdlg(...
    'title' , 'Configuración inicial', ...
    'separator' , 'General',...
    {'Referencia de color';'color_mode'} , {'Temporal','Térmico'}, ...
    {'Submuestreo';'rsmp'} , 1, ...
    'separator' , 'Ajuste de elipsoide',...
    {'Seleccione método';'sel_metodo'} , {'Petrov','Merayo','Vitali','Li/Griffiths','Ninguno'}, ...
    {'Radio real de esfera';'ref_scale'} , 324.99, ...
    {'Generar esfera solida';'plot_solid'} , false,...
    {'Generar planos 2D'; 'plot_2d'} , false, ...
    {'Estadísticas';'std'} , true)

clc;

fprintf("\r\n*****\r\n");
[fname, pname] = uigetfile('*.csv', 'Seleccione un archivo de registro WiMo', '/');
log_file = fullfile(pname, fname);
M = csvread(log_file,1,0);

dstr = strsplit(fname, '_');
fprintf("Fecha de la muestra:\t%s %s %s | %c%c:%c%c:%c%c\n",dstr{3},dstr{4},dstr{5},dstr{6}(1),dstr{6}(2),dstr{6}(3),dstr{6}(4),dstr{6}(5),dstr{6}(6));

%Referencia de color
if(settings.color_mode == "Temporal")
    t=M(:,1)/1000; %Temporal
    colors_pal = 'winter';
else
    t=M(:,2)/340 + 35.63; %Térmico
    colors_pal = 'parula';
end

if settings.rsmp<1
    settings.rsmp=1;
end

t2 = resample(t,1,settings.rsmp);%(30:end-30);
x = M(:,9);
y = M(:,10);
z = M(:,11);
x2 = resample(x,1,settings.rsmp);%(30:end-30);
y2 = resample(y,1,settings.rsmp);%(30:end-30);
z2 = resample(z,1,settings.rsmp);%(30:end-30);

%*****
nombre_metodo = "No compensado";
grap_title = "Magnetómetro - " + nombre_metodo;
figure('Name',grap_title)
scatter3(x2,y2,z2,4,t2,'filled');
axis equal;
colormap(colors_pal);
title(grap_title);
xlabel("X");
ylabel("Y");
zlabel("Z")
if(settings.std)
    fprintf("***** Estadísticas no compensado *****\r\n");
    SN = [x2 y2 z2];
    %offs = ((max(SN)-min(SN))/2)+min(SN);
    [ n_center ] =ellipsoid_petrov([x y z], '');

```

```

SN=[x2-n_center(1), y2-n_center(2),z2-n_center(3)];
[stt,md]=std_esfera(SN);
fprintf("Radio medio:\r\n");
disp(md);
fprintf("Desviación estándar de radio:\r\n");
disp(stt);
end

if(settings.plot_solid)
    p = generar_superficie([x2 y2 z2]);
    hold off;
    set( p, 'FaceColor', 'none', 'EdgeColor', [0.6 0.9 0.5] );
    %lighting flat;
    axis vis3d equal;
    %camlight;
    hold on;
end

if(settings.plot_2d)
    graficar_planos([x2 y2 z2], t2, nombre_metodo);
end

switch settings.sel_metodo
%*****
case "Vitali"
    fprintf("***** Parámetros - Vitali *****\r\n");
    nombre_metodo = "Método Vitali (ST)";
    grap_title = "Magnetómetro - " + nombre_metodo;
    tim_on = cputime;
    [ofs,gain,rotM]=ellipsoid_vitali([x2 y2 z2],0);
    S = [x2-ofs(1), y2-ofs(2), z2-ofs(3)];
    scal = inv(diag(gain));
    comp = rotM*scal;
    S=S*comp; % rotate to XYZ axes
    fit_scale = median(sqrt(S(:,1).^2 + S(:,2).^2 + S(:,3).^2));
    fit_scale = settings.ref_scale/fit_scale;
    fit_scale = diag([fit_scale fit_scale fit_scale]);
    XYZC = S*fit_scale;
    exe_time=cputime-tim_on;
    XC=XYZC(:,1);
    YC=XYZC(:,2);
    ZC=XYZC(:,3); % scale to sphere
    fprintf("Vector de corrimiento + hard-iron:\r\n");
    disp(ofs);
    fprintf("Matriz de compensación C (rotado):\r\n");
    disp(comp*fit_scale);
    fprintf("Matriz de escala:\r\n");
    disp(scal*fit_scale);
    if(settings.std)
        [stt,md]=std_esfera(XYZC);
        fprintf("Radio medio:\r\n");
        disp(md);
        fprintf("Desviación estándar de radio:\r\n");
        disp(stt);
        fprintf("Tiempo de ejecución:\r\n");
        disp(exe_time);
    end

    figure('Name',grap_title)
    scatter3(XC,YC,ZC,4,t2,'filled');
    axis equal;
    title(grap_title);
    xlabel("X");
    ylabel("Y");
    zlabel("Z");
    colormap(colors_pal);

```

```

if(settings.plot_solid)
    p = generar_superficie([XC YC ZC]);
    hold off;
    set( p, 'FaceColor', 'none', 'EdgeColor', [0.6 0.9 0.5] );
    %lighting flat;
    axis vis3d equal;
    %camlight;
    hold on;
end

if(settings.plot_2d)
    graficar_planos([XC YC ZC], t2, nombre_metodo);
end

%*****
case "Li/Griffiths"
    fprintf("***** Parámetros - Li/Griffiths *****\r\n");
    nombre_metodo = "Método Li/Griffiths";
    grap_title = "Magnetómetro - " + nombre_metodo;
    tim_on = cputime;
    [e_center, e_radii, e_eigenvecs]=elipsoid_li([x2 y2 z2]);
    S = [x2 - e_center(1), y2 - e_center(2), z2 - e_center(3)];
    scale = inv([e_radii(1) 0 0; 0 e_radii(2) 0; 0 0 e_radii(3)]) * min(e_radii);
    map = e_eigenvecs';
    invmap = e_eigenvecs;
    comp = invmap * scale * map;
    S = comp * S;
    fit_scale = median(sqrt(S(:,1).^2 + S(:,2).^2 + S(:,3).^2));
    fprintf("Radio de la esfera Li/Griffiths:\r\n");
    disp(fit_scale);
    fprintf("Radio de la esfera real:\r\n");
    disp(settings.ref_scale);
    fit_scale = settings.ref_scale/fit_scale;
    fit_scale = diag([fit_scale fit_scale fit_scale]);
    XYZC2 = fit_scale*S;
    soft_iron = inv(scale)*comp;
    exe_time = cputime-tim_on;
    XC2=XYZC2(1,:);
    YC2=XYZC2(2,:);
    ZC2=XYZC2(3,:);

    fprintf("Vector de corrimiento + hard-iron:\r\n");
    disp(e_center);
    fprintf("Matriz de compensación C:\r\n");
    disp(fit_scale*comp);
    fprintf("Matriz de escala:\r\n");
    temp = scale*fit_scale;
    disp(temp);
    fprintf("Vector de factores S:\r\n");
    temp = temp-diag([1 1 1]);
    temp = [temp(1,1);temp(2,2);temp(3,3)];
    disp(temp);
    fprintf("Matriz soft-iron:\r\n");
    disp(soft_iron);
    if(settings.std)
        [stt,md]=std_esfera(S');
        fprintf("Radio medio:\r\n");
        disp(md);
        fprintf("Desviación estándar de radio:\r\n");
        disp(stt);
        fprintf("Tiempo de ejecución:\r\n");
        disp(exe_time);
    end

    figure('Name',grap_title)

```

```

scatter3(XC2,YC2,ZC2,4,t2,'filled');
title(grap_title);
xlabel("X");
ylabel("Y");
zlabel("Z");
axis equal;
colormap(colors_pal);

if(settings.plot_solid)
    p = generar_superficie(XYZC2');
    hold off;
    set( p, 'FaceColor', 'none', 'EdgeColor', [0.6 0.9 0.5] );
    %lighting flat;
    axis vis3d equal;
    %camlight;
    hold on;
end

if(settings.plot_2d)
    graficar_planos(XYZC2', t2, nombre_metodo);
end

%*****
case "Merayo"
    fprintf("***** Parámetros - Merayo *****\n\n");
    nombre_metodo = "Método Merayo";
    grap_title = "Magnetómetro - " + nombre_metodo;
    tim_on = cputime;
    [U,c] = elipsoid_merayo([x2 y2 z2]');
    S = [x2-c(1), y2-c(2), z2-c(3)];
    S=S*U; % calibrated data
    fit_scale = median(sqrt(S(:,1).^2 + S(:,2).^2 + S(:,3).^2));
    fit_scale = settings.ref_scale/fit_scale;
    fit_scale = diag([fit_scale fit_scale fit_scale]);
    S=S*fit_scale;
    exe_time= cputime-tim_on;
    XC3=S(:,1);
    YC3=S(:,2);
    ZC3=S(:,3);
    fprintf("Vector de corrimiento + hard-iron:\n\n");
    disp(c);
    fprintf("Matriz de compensación C:\n\n");
    disp(U*fit_scale);
    if(settings.std)
        [stt,md]=std_esfera(S);
        fprintf("Radio medio:\n\n");
        disp(md);
        fprintf("Desviación estándar de radio:\n\n");
        disp(stt);
        fprintf("Tiempo de ejecución:\n\n");
        disp(exe_time);
    end

    figure('Name',grap_title)
    scatter3(XC3,YC3,ZC3,4,t2,'filled');
    axis equal;
    title(grap_title);
    xlabel("X");
    ylabel("Y");
    zlabel("Z");
    colormap(colors_pal);

    if(settings.plot_solid)
        p = generar_superficie([XC3 YC3 ZC3]);
        hold off;
        set( p, 'FaceColor', 'none', 'EdgeColor', [0.6 0.9 0.5] );

```

```

    %lighting flat;
    axis vis3d equal;
    %camlight;
    hold on;
end

if(settings.plot_2d)
    graficar_planos([XC3 YC3 ZC3], t2, nombre_metodo);
end

%*****
case "Petrov"
    fprintf("***** Parámetros - Petrov *****\r\n");
    nombre_metodo = "Método Petrov (Oculus VR)";
    grap_title = "Magnetómetro - " + nombre_metodo;
    tim_on=cputime;
    [e_center, e_radii, e_eigenvecs, e_v]=ellipsoid_petrov([x2 y2 z2]);
    S = [x2 - e_center(1), y2 - e_center(2), z2 - e_center(3)];
    scale = inv([e_radii(1) 0 0; 0 e_radii(2) 0; 0 0 e_radii(3)]) * min(e_radii);
    map = e_eigenvecs';
    invmap = e_eigenvecs;
    comp = invmap * scale * map;
    S = comp * S;
    fit_scale = median(sqrt(S(1,:).^2 + S(2,:).^2 + S(3,:).^2));
    fprintf("Radio de la esfera Petrov:\r\n");
    disp(fit_scale);
    fprintf("Radio de la esfera real:\r\n");
    disp(settings.ref_scale);
    fit_scale = settings.ref_scale/fit_scale;
    fit_scale = diag([fit_scale fit_scale fit_scale]);
    S=fit_scale*S;
    exe_time = cputime-tim_on;
    soft_iron = inv(scale)*comp;
    XC4=S(1,:);
    YC4=S(2,:);
    ZC4=S(3,:);
    fprintf("Vector de corrimiento + hard-iron:\r\n");
    disp(e_center);
    fprintf("Matriz de compensación C:\r\n");
    disp(fit_scale*comp);
    fprintf("Matriz de escala:\r\n");
    temp = scale*fit_scale;
    disp(temp);
    fprintf("Vector de factores S:\r\n");
    temp = temp-diag([1 1 1]);
    temp = [temp(1,1);temp(2,2);temp(3,3)];
    disp(temp);
    fprintf("Matriz soft-iron:\r\n");
    disp(soft_iron);
    if(settings.std)
        [stt,md]=std_esfera(S');
        fprintf("Radio medio:\r\n");
        disp(md);
        fprintf("Desviación estándar de radio:\r\n");
        disp(stt);
        fprintf("Tiempo de ejecución:\r\n");
        disp(exe_time);
    end
    figure('Name',grap_title)
    % draw original and compensated data
    scatter3(XC4,YC4,ZC4,4,t2,'filled');
    axis equal;
    title(grap_title);
    xlabel("X");
    ylabel("Y");
    zlabel("Z");

```

```

        colormap(colors_pal);

        if(settings.plot_solid)
            p = generar_superficie(S');
            hold off;
            set( p, 'FaceColor', 'none', 'EdgeColor', [0.6 0.9 0.5] );
            %lighting flat;
            axis vis3d equal;
            %camlight;
            hold on;
        end

        if(settings.plot_2d)
            graficar_planos(S', t2, nombre_metodo);
        end
    end

    %*****
function [stdm,med]=std_esfera(data)
    radd = sqrt(data(:,1).^2 + data(:,2).^2 + data(:,3).^2);
    stdm = std(radd);
    med = median(radd);
end

function graficar_planos(data,n,grafname)
x=data(:,1);
y=data(:,2);
z=data(:,3);
name = "Magnetómetro XY - " + grafname;

figure('Name',name)
scatter(x,y,4,n,'filled');
colormap(colors_pal);
grid on;
axis equal;
title(name);
xlabel('X');
ylabel('Y');

name = "Magnetómetro XZ - " + grafname;
figure('Name',name)
scatter(x,z,4,n,'filled');
colormap(colors_pal);
grid on;
axis equal;
title(name);
xlabel('X');
ylabel('Y');

name = "Magnetómetro YZ - " + grafname;
figure('Name',name)
scatter(y,z,4,n,'filled');
colormap(colors_pal);
grid on;
axis equal;
title(name);
xlabel('X');
ylabel('Y');
end

function [p] = generar_superficie(data)
x=data(:,1);
y=data(:,2);
z=data(:,3);
[ center, radii, evecs, v, chi2 ] =ellipsoid_petrov([x y z], '');
%draw fit
mind = min( [ x y z ] );

```

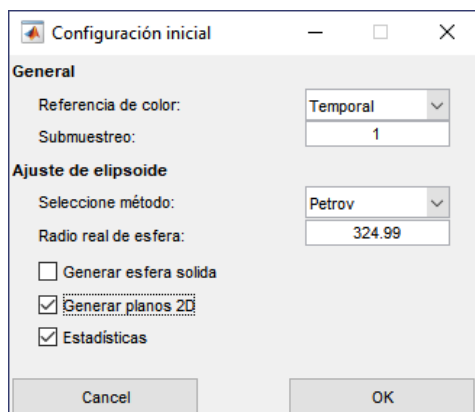
```

maxd = max( [ x y z ] );
nsteps = 50;
step = ( maxd - mind ) / nsteps;
[ x3, y3, z3 ] = meshgrid( linspace( mind(1) - step(1), maxd(1) + step(1), nsteps ), linspace(
mind(2) - step(2), maxd(2) + step(2), nsteps ), linspace( mind(3) - step(3), maxd(3) + step(3),
nsteps ) );

Ellipsoid = v(1) *x3.*x3 + v(2) * y3.*y3 + v(3) * z3.*z3 + ...
2*v(4) *x3.*y3 + 2*v(5)*x3.*z3 + 2*v(6) * y3.*z3 + ...
2*v(7) *x3 + 2*v(8)*y3 + 2*v(9) * z3;
p = patch( isosurface( x3, y3, z3, Ellipsoid, -v(10) ) );
end

```

Este es el segundo script de Matlab empleado, esta vez para calcular los parámetros estáticos atendiendo al modelo del magnetómetro consultado. Para ejecutar correctamente este script es necesario cumplir con algunas dependencias, para ello los archivos: ellipsoid\_li.m, ellipsoid\_merayo.m, ellipsoid\_petrov, ellipsoid\_vitali.m, ellipsoid\_parameters.m y settingsdlg.m se deben encontrar en el mismo path (ruta) de ellipsoids.m. Al ejecutarlo se abre un menú de configuración:



La referencia de color se relaciona con los datos empleados para definir la coloración de las gráficas en tres dimensiones, lo típico es emplear la referencia temporal, pero si son detectadas aberraciones en la lectura producto de variaciones en la temperatura puede resultar útil generar las imágenes en función de esta última variable.

El submuestreo es un valor entero para dividir la cantidad de muestras utilizadas para generar las gráficas, el valor de 1 está bien para la mayoría de casos. Sin embargo, si se cuenta con un equipo con poca memoria RAM o la muestra tiene varias horas de extensión es posible que tenga que incrementar este valor para reducir la carga de trabajo del computador.

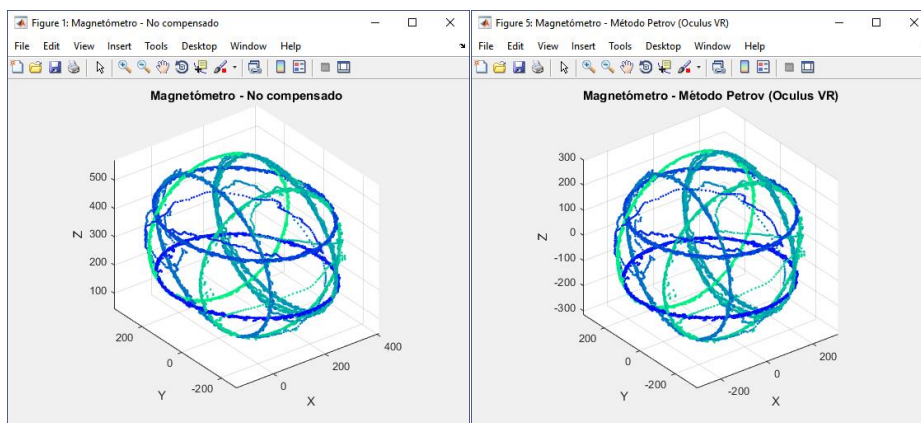
En cuanto a la sección “Ajuste de elipsoide”, están disponibles varias opciones que van acordes a la información presentada en el documento. Primero, el método responde a la técnica matemática aplicada para resolver el elipsoide, el radio real es el valor digital entregado por el sensor idealmente según el MMM y la resolución digital configurada. Generar esfera sólida activa el cálculo y renderización de una esfera de relleno para comparar. Generar planos 2D habilita la generación de gráficos en planos ortogonales XY, XZ, YZ como ayuda de análisis gráfico. Finalmente, la opción de estadísticas permite que se calculen las desviaciones estándar de cada



punto de medición respecto al centro, esto ofrece un indicador de calidad sobre el centrado del elipsoide (corrección offset) y deformación por efectos *soft-iron*.

Al presionar OK se abrirá una ventana de exploración para buscar y seleccionar un archivo CSV con la muestra que se desea analizar.

Con el archivo procesado, se generan automáticamente dos gráficos en 3D que ilustran los datos de la muestra antes y después del cálculo de los parámetros de corrección. Si están activados los gráficos en 2D se obtendrán seis imágenes adicionales con las vistas de planos.



En la ventana de comando de Matlab se observarán los datos generados en el proceso:

```

*****
Fecha de la muestra:      03 Jun 2018 | 02:44:29 GMT
***** Estadísticas no compensado *****

Radio medio:
    290.6357

Desviación estándar de radio:
    12.4647

***** Parámetros - Petrov *****

Radio de la esfera Petrov:
    267.1267

Radio de la esfera real:
    324.9900

Vector de corrimiento + hard-iron:
    127.4893
    33.4236
    
```

## Anexos

---

314.9787

Matriz de compensación C:

1.1239	0.0263	0.0348
0.0263	1.0504	-0.0418
0.0348	-0.0418	1.1971

Matriz de escala:

1.0267	0	0
0	1.1282	0
0	0	1.2166

Vector de factores S:

0.0267  
0.1282  
0.2166

Matriz soft-iron:

1.0947	0.0256	0.0339
0.0233	0.9311	-0.0371
0.0286	-0.0344	0.9840

Radio medio:

324.9900

Desviación estándar de radio:

2.4553

Tiempo de ejecución:

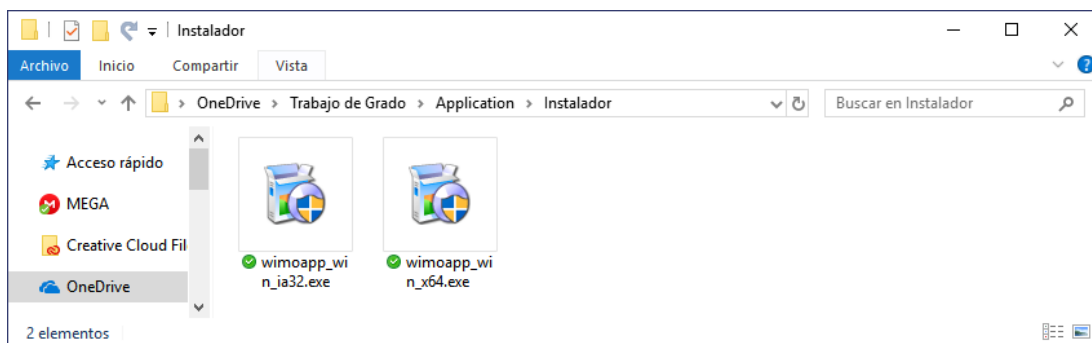
0.0469

La información anterior es obtenida y empleada para la corrección atendiendo a los modelos matemáticos definidos en los capítulos 2 y 5.

## G – WIMO APP

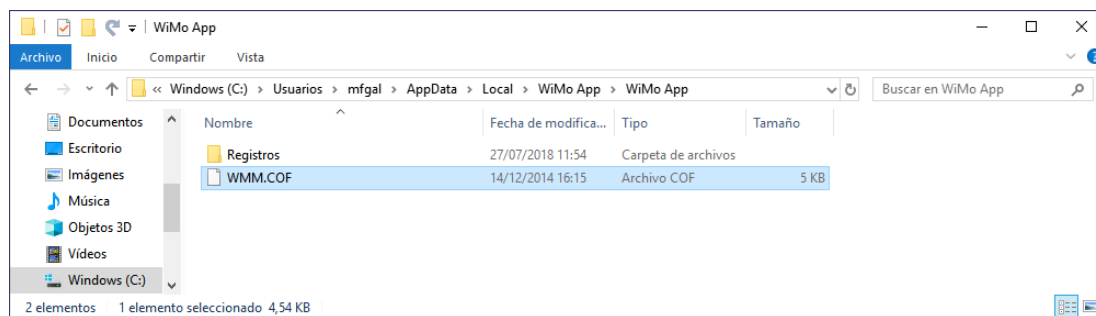
### 1. Instalación

Se dispone de dos paquetes de instalación compatibles con Windows 7, 8.1 y 10. Ejecute la instalación *wimoapp\_win\_ia32.exe* si su sistema es de 32bit, en el caso de que sea de 64bit (AMD64) emplee el instalador *wimoapp\_win\_x64.exe*.



Siga las instrucciones en pantalla, el instalador no permite cambiar la ruta de ubicación del programa que por defecto es **C:\Users\{nombre de usuario}\AppData\Local\WiMo App**. Automáticamente será creado un acceso directo a la aplicación y de la carpeta que almacena datos en su escritorio, puede que este último acceso directo su sistema no lo reconozca correctamente, en ese caso abra el explorador y diríjase a la ruta **%userprofile%\AppData\Local\WiMo App\WiMo App**. Cree un acceso directo de esta carpeta para comodidad del usuario, en esta ruta serán almacenados los datos generados durante las secciones de captura de datos.

Para utilizar el modo de cálculo y ajuste para la corrección de sensores, es necesario que el archivo de coeficientes del Modelo Magnético Mundial 2015 *WMM.COF* se encuentre también en esta carpeta, para obtener este archivo descargue *WMM2015COF.zip* de la página de NOAA <https://www.ngdc.noaa.gov/geomag/WMM/data/WMM2015/WMM2015COF.zip>



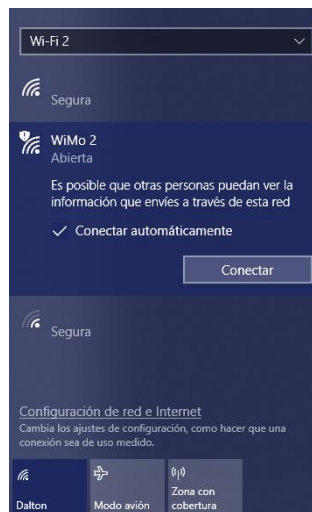
Si tiene un espacio limitado en su sistema deberá dirigir manualmente los datos de registros a otro directorio o descargar constantemente los ficheros de la carpeta Registros. Con esto completado ya podrá ejecutar el programa empleando los paneles descritos en los siguientes numerales.

## 2. Panel de datos

El panel de datos permite indicar la dirección del nodo con el que se establecerá conexión, capturar datos, visualizar parámetros de configuración, máximos y mínimos de cada sensor, etc.



Para iniciar, establezca una conexión a través de red con el nodo:



Una vez conectado, indique la dirección IP en el panel CONEXIÓN dentro de WiMo APP. Si el nodo no tiene configurada una dirección estática deberá utilizar el monitor serial conectado al nodo para observar que IP tiene asignada o recurrir a una herramienta de análisis de red para determinarla.

Escriba la dirección con puntos, los ceros a la izquierda son indiferentes, por ejemplo:

192.168.1.1 > Es una dirección válida

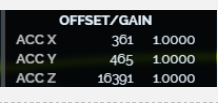



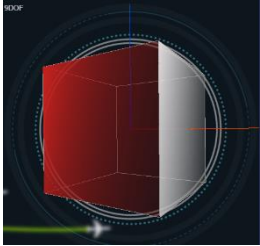
192.168.01.01 > Es una dirección válida  
 19216811 > No es una dirección válida

Al ingresar la dirección haga click en el botón Conectar o presione la tecla Enter, si todo es correcto debería escuchar un sonido y los datos del sensor ya deben ser visibles:



En la siguiente tabla se detalla cada bloque dentro del panel de datos:

Bloque	Detalle
	<p><b>CONEXIÓN:</b> Permite al usuario ingresar la dirección IP e iniciar la sesión con el botón Conectar. La última dirección IP utilizada es almacenada, por lo que no es necesario reescribirla cada vez que se inicie la aplicación.</p>
	<p><b>WIMO:</b> Muestra el estado básico del sensor. El primer indicador muestra el tiempo que el nodo lleva activado. Energía marca un estimado del voltaje que está entrando al nodo. Muestreo estima la frecuencia de recepción de paquetes, este valor depende del periodo de muestreo (TX PERIOD) y de la señal de red. Temperatura aproxima en grados Celsius la temperatura del nodo.</p>
	<p><b>PARÁMETROS:</b> Indica el estado de configuración según los registros asignados en el nodo. Para más detalle revise el capítulo 4 y el anexo C. Presione el botón con flechas en círculo para actualizar los datos en caso de mostrarse en ceros.</p>

	<p><b>FACTOR ALPHA:</b> Son los factores de corrección térmica lineal asignados a cada sensor.</p>
	<p><b>OFFSET / GAIN:</b> Muestra el valor asignado de factor de corrimiento (offset) en la columna izquierda y el factor de ganancia en la columna derecha.</p>
	<p><b>Ubicación:</b> Permite al usuario ingresar manualmente las coordenadas (WGS84) y altura en metros sobre el nivel del mar. La ubicación es necesaria para el cálculo de intensidad magnética en la zona de acuerdo al MMM.</p> <p>El botón actualizar ordena almacenar la información. No es soportado el punto como decimal, debe usarse una coma.</p>
	<p><b>SENSORES:</b> Muestra los valores de los tres sensores en su respectivo eje. En blanco y de mayor tamaño es mostrado el valor actual. Los números en tamaño más pequeño son los máximos en púrpura y mínimos en cian. El botón azul en la esquina superior izquierda reinicia los indicadores de máximos y mínimos.</p>
	<p><b>COMP:</b> Muestra el estado de los compensadores: térmico(thermal), de corrimiento (offset) y ganancia (gain).</p>
	<p><b>REGISTRO:</b> Controla el registro. Presione Buscar... para seleccionar una ruta de almacenamiento de datos. Y deslice “Crear registro” para iniciarlo. Recuerde que si cambia de pestaña (Configuración u Optimización) el registro se detiene.</p>
	<p><b>Cubo:</b> Representa en 3D el movimiento calculado con el algoritmo Madgwick. Está disponible el modo 6DOF (acelerómetro y giroscopio) y 9DOF (acelerómetro, giroscopio y magnetómetro) que se alterna haciendo click sobre el cubo. Los ejes de color representan X (rojo), Y(verde) y Z(azul).</p>
	<p><b>Posición de cubo:</b> Muestra el valor calculado de posición del cubo según el algoritmo Magwick. El formato de cuaternión responde a la referencia NASA-JPL. En la esquina inferior izquierda está el indicador Z-Offset, si se hace click sobre esta palabra se realiza una adición de 10° para ajustar la diferencia del eje Z entre el norte magnético y la posición de la pantalla.</p>

Para cambiar de panel seleccione uno de ellos haciendo click en el bloque superior, si hace click sobre el panel que está activo se cargará nuevamente el panel, esto puede ser útil en caso de presentarse algún error o inconsistencia en los datos o visualización.



**Nota:** Si no hay conexión activa no es posible cambiar de panel.

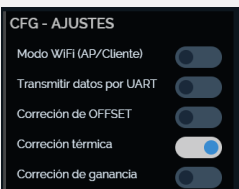
### 3. Panel de configuración

Este panel permite transmitir los parámetros de configuración a el nodo. Para acceder a este panel es necesario tener una sesión iniciada, en caso de que la conexión se interrumpa se intentará reconectar y no podrá cambiarse de pestaña.



#### Bloque

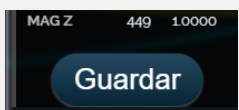
#### Detalle



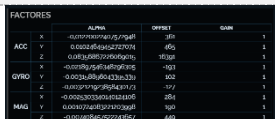
**CFG – AJUSTES:** Permite ajustar, mediante deslizadores, los valores del registro CFG del nodo, que configura los modos descritos en el capítulo 4.



**VALORES DE CONFIGURACIÓN:** Ajusta valores asociados para configurar las ganancias de los sensores, el filtrado y periodo de transmisión. Para aplicar estos valores de configuración y del bloque CFG presione el botón con el nombre Aplicar. **Los valores se mantienen en memoria durante la ejecución, pero no son almacenados.**



**PARÁMETROS:** Muestra los valores que se encuentran en memoria en el dispositivo. **Presione el botón Guardar para almacenar los valores modificados** en los bloques CFG, VALORES DE CONFIGURACIÓN y FACTORES.



**FACTORES:** Permite observar e ingresar los factores alpha, offset y gain. Una vez modificados presione el botón Aplicar y si es el caso, Guardar.



**SENSORES:** Muestra en tiempo real las respuestas de los sensores.

#### 4. Panel Optimización:

Este panel asiste para el cálculo de parámetros de corrección térmica, ganancia y compensación con el método integrado en el nodo. Si se va a realizar la obtención de parámetros de ganancia y compensación es necesario que la plataforma de rotación esté conectada y encendida.



##### 4.1 Corrección térmica

Para realizar este proceso es necesario que el sensor tenga la posibilidad de cambiar su temperatura de manera constante. La forma más fácil, consiste en realizar este procedimiento apenas el nodo sea encendido, así iniciará a temperatura ambiente que luego aumentará producto de la actividad de los circuitos.

Primero verifique la temperatura actual en el panel de sensores:

Temperatura: 41.41°C

Si la temperatura es muy elevada, puede guardar el nodo apagado durante un tiempo en una nevera, asegurándose de hacerlo con una bolsa o recipiente junto con gel de sílice para prevenir condensación de humedad, de no hacerlo es posible que se bloqueen los circuitos.

Luego, en el bloque AJUSTE TÉRMICO ingrese la temperatura de inicio y final, puede ser para aumento o disminución de temperatura, las dos opciones son válidas. Si ya hay parámetros de corrección en memoria puede utilizar el modo “mejorar factores actuales”, de esta manera se mantienen los valores de corrección en aplicación, lo que potencia la mejora de estos. En caso de no hacerlo, se ignoran los valores de corrección y se calcularán nuevos factores. Finalmente, deslice Iniciar para activar el registro de valores.





Cuando se inicia la captura de datos para obtener los valores de corrección térmica, se ajustarán primeramente los valores de OFFSET para que los sensores sean cercanos a cero, esto como ayuda para visualizar los cambios en las curvas del bloque SENSORES. Si se desea almacenar los datos utilizados en este proceso en un archivo CSV deslice REGISTRAR EN CSV en el bloque REGISTRO.

Una vez finalizado el proceso, los nuevos factores se mostrarán en el bloque FACTORES ALPHA, si el valor de corrección necesario es muy bajo la palomita o check no se marcará en la lista. Los valores que tengan el check marcado serán aplicados y guardados al presionar el botón "GUARDAR". Respecto a los valores de offset y ganancia serán restaurados a sus valores originales previos a la sesión.

FACTORES ALPHA	
<input checked="" type="checkbox"/> ACC X	-0.015341564585791712
<input checked="" type="checkbox"/> ACC Y	0.013435393847178649
<input checked="" type="checkbox"/> ACC Z	0.06935367890674898
<input checked="" type="checkbox"/> GYRO X	-0.02375127792434929
<input checked="" type="checkbox"/> GYRO Y	-0.004809903113050816
<input checked="" type="checkbox"/> GYRO Z	-0.0026782988311033597
<input checked="" type="checkbox"/> MAG X	-0.004080733893602495
<input checked="" type="checkbox"/> MAG Y	-0.001264137364135009
<input checked="" type="checkbox"/> MAG Z	-0.006273756988485761

**Guardar**

## 4.2 Ajuste de ganancia y corrimiento

Primero asegúrese de que la plataforma rotativa se encuentre conectada, es posible verificarlo abriendo nuevamente el panel Optimización, cuando esto sucede la plataforma emitirá un sonido para informar que está operativa y controlada por la aplicación.

Para iniciar la rutina haga click en el botón >> y siga las instrucciones indicadas en pantalla



Primero se realizará un trabajo estático, es necesario que el nodo sea ubicado sobre una superficie nivelada, esto permite obtener los valores de ajuste para corregir el acelerómetro:



En la segunda sección se utilizará la plataforma rotativa que también debe estar en una superficie nivelada, siga las instrucciones, cuando esté rotando debe asegurarse que la velocidad de rotación sea estable en la curva mostrada en el bloque SENSORES, con esto se ajustará el giroscopio:



En la última parte se realizarán las operaciones con el magnetómetro, por eso es necesario que el archivo *WMM.COF* se encuentre en la ruta mencionada anteriormente, además la rutina debe realizarse en un área libre de materiales que afecten el campo magnético de la tierra.

## 5. Sobre los archivos de configuración

Estos archivos son generados en la ruta de almacenamiento de registros, modificar algunos valores pueden resultar útiles.

### 5.1 WiMoApp.config

Este archivo contiene una estructura JSON que almacena datos de ubicación, configuración de red y el factor de ganancia de integración del algoritmo Madwick.

```
{
  "geolocation": {
    "lat": 2.6409517,
    "long": -76.5324257,
    "alt": 1810
  },
  "madgwick": {
    "beta": 0.5
  },
  "cnx": {
    "ip": "192.168.181.180",
    "port": 42100
  }
}
```

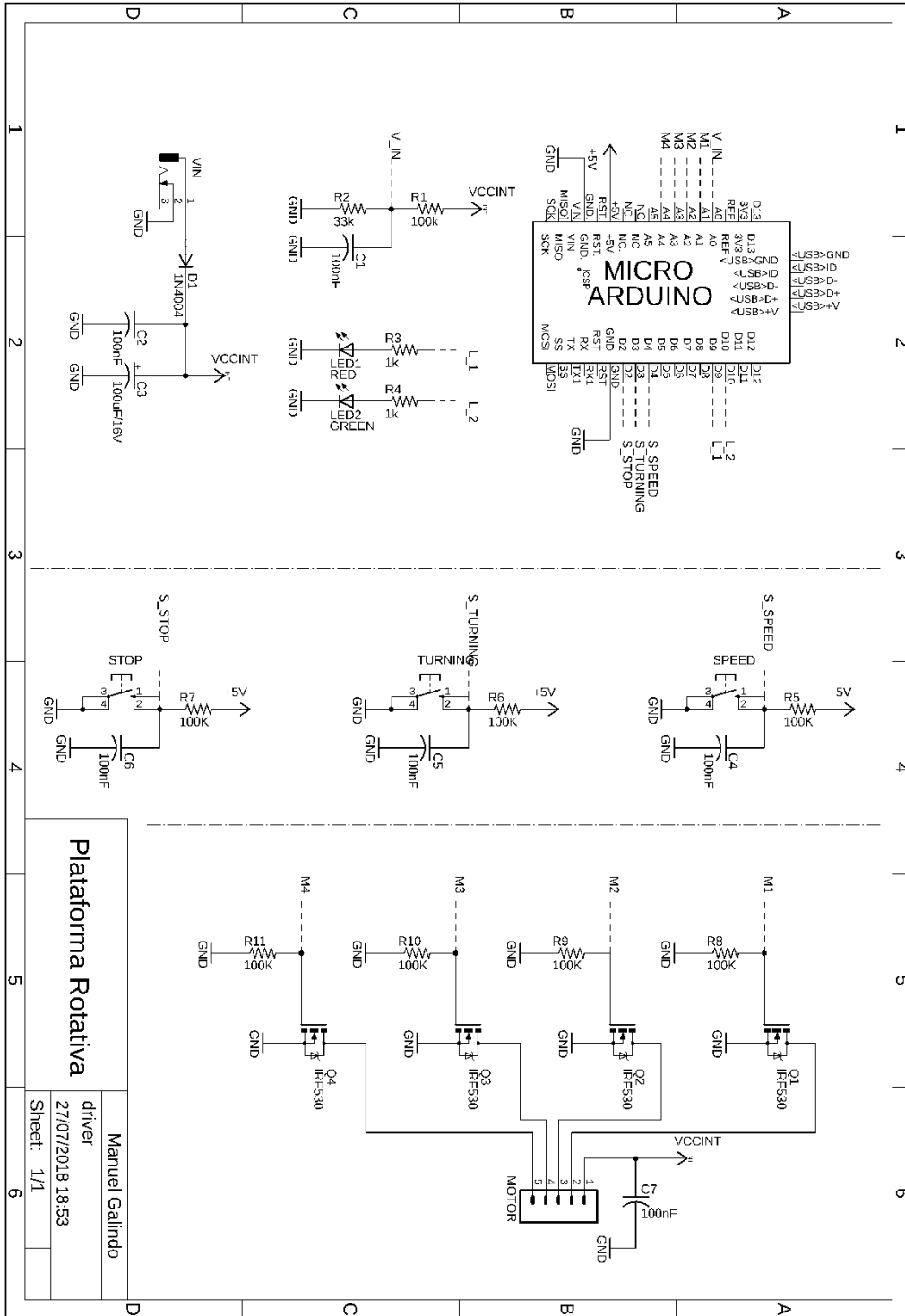
### 5.2 WMM\_Local.json

Al ejecutar la corrección de magnetómetro, este archivo se genera en un formato JSON para mostrar al usuario los valores calculados al aplicar el Modelo Magnético Mundial. Los nombres de las variables responden a los utilizados en la documentación oficial del WMM2015.

```
{
  "dec": -5.401554260893232,
  "dip": 25.29551359831603,
  "ü": 29900.585671693407,
  "bh": 27033.59821625097,
  "bx": 26913.553127205014,
  "by": -2544.8164151468814,
  "bz": 12776.133648040373,
  "lat": "2.6409517",
  "lon": "-76.5324257",
  "epoch": 2015
}
```

# H – PLATAFORMA ROTATIVA

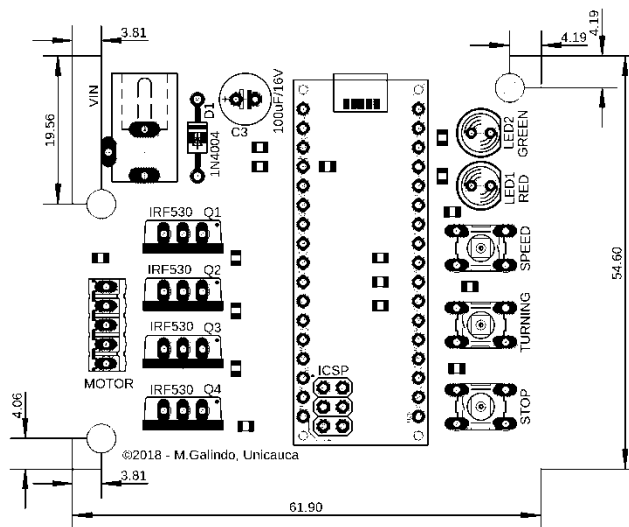
## 1. Esquema eléctrico



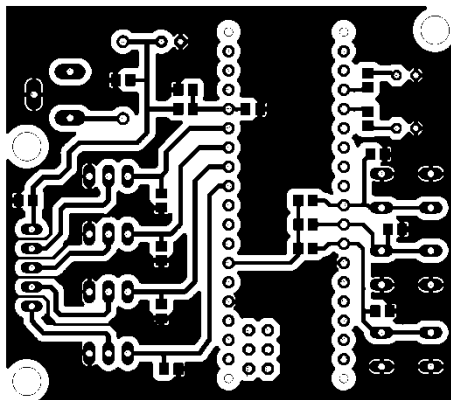
**Plataforma Rotativa**

Manuel Galindo  
 driver  
 27/07/2018 18:53  
 Sheet: 1/1

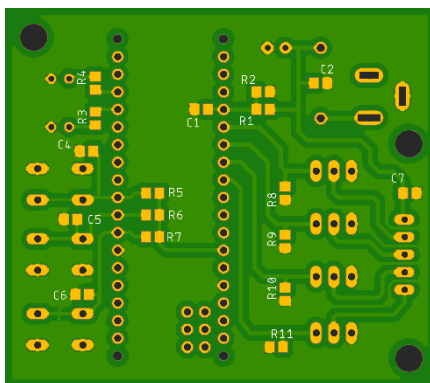
## 2. Esquema mecánico



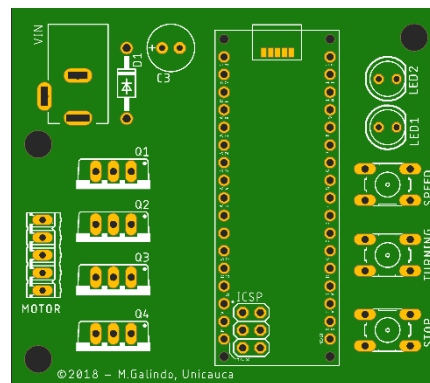
## 3. Esquema de circuito eléctrico



## 4. Esquema de ubicación de elementos



Vista inferior



Vista superior

## 5. Bill of Materials

Parte	Valor	Dispositivo	Paquete
<b>C1</b>	100nF	C-USC0805	C0805
<b>C2</b>	100nF	C-USC0805	C0805
<b>C3</b>	100uF/16V	CPOL-USE2.5-7	E2,5-7
<b>C4</b>	100nF	C-USC0805	C0805
<b>C5</b>	100nF	C-USC0805	C0805
<b>C6</b>	100nF	C-USC0805	C0805
<b>C7</b>	100nF	C-USC0805	C0805
<b>D1</b>	1N4004	1N4004	DO41-10
<b>LED1</b>	RED	LED-5MM	LED-5MM
<b>LED2</b>	GREEN	LED-5MM	LED-5MM
<b>MOTOR</b>		M05	05P
<b>PCB1</b>	ARDUINO_MICRO	ARDUINO_MICRO	ARDUINO_MICRO
<b>Q1</b>	IRF530	IRF530	TO220BV
<b>Q2</b>	IRF530	IRF530	TO220BV
<b>Q3</b>	IRF530	IRF530	TO220BV
<b>Q4</b>	IRF530	IRF530	TO220BV
<b>R1</b>	100k	R-US_R0805	R0805
<b>R2</b>	33k	R-US_R0805	R0805
<b>R3</b>	1k	R-US_R0805	R0805
<b>R4</b>	1k	R-US_R0805	R0805
<b>R5</b>	100K	R-US_R0805	R0805
<b>R6</b>	100K	R-US_R0805	R0805
<b>R7</b>	100K	R-US_R0805	R0805
<b>R8</b>	100K	R-US_R0805	R0805
<b>R9</b>	100K	R-US_R0805	R0805
<b>R10</b>	100K	R-US_R0805	R0805
<b>R11</b>	100K	R-US_R0805	R0805
<b>SPEED</b>		10-XX	B3F-10XX
<b>STOP</b>		10-XX	B3F-10XX
<b>TURNING</b>		10-XX	B3F-10XX
<b>VIN</b>		DCJ0303	DCJ0303

**NOTA:** El motor empleado es paso a paso unipolar de 200 pasos por vuelta (1,8°/paso) a 5V.