

UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Diseño de controlador de vuelo para cuadricóptero con  
capacidad de integrarse a ecosistema Robotat vía inalámbrica  
Wi-Fi**

Trabajo de graduación presentado por Hans Alexander Burmester  
Campos para optar al grado académico de Licenciado en Ingeniería  
Mecatrónica

Guatemala,

2022







UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Diseño de controlador de vuelo para cuadricóptero con  
capacidad de integrarse a ecosistema Robotat vía inalámbrica  
Wi-Fi**


Trabajo de graduación presentado por Hans Alexander Burmester  
Campos para optar al grado académico de Licenciado en Ingeniería  
Mecatrónica

Guatemala,


2022




Vo.Bo.:

(f)   
MAEB Ing. Pablo Mazariegos

Tribunal Examinador:

(f)   
MAEB Ing. Pablo Mazariegos

(f)   
MSc. Miguel Enrique Zea Arenales

(f)   
MSc. Luis José Pinillos

Fecha de aprobación: Guatemala, 7 de enero de 2022 .





Quiero darle gracias a mi familia, quienes me han apoyado de principio a fin en este cansado, pero especial y único proceso. En el caso de mis padres y también modelos a seguir, Manuel y Lesly, agradecerles se queda corto por todo lo que han hecho por mí. Sin ustedes yo no sería ni la sombra de la persona que soy hoy en día, solo les puedo decir que este triunfo es más suyo que mío. Lo que los tres empezamos como un reto, cinco años después podemos decir que lo logramos. Para mis hermanos, Gustavo y Leslie, gracias por aguantar el estrés y apoyarme en los momentos difíciles. Cuando necesitaba una mano ustedes siempre estuvieron de manera incondicional.

De la misma manera agradezco a mis asesor Pablo Mazariegos, y al catedrático Miguel Zea, los cuales me asistieron y alentaron durante toda la ejecución del presente trabajo de graduación. Finalmente quiero agradecerles a mis amigos y compañeros de carrera, que más que eso se convirtieron en mi segunda familia. Ellos lograron que no me sintiera tan alejado de casa cuando empezamos este viaje que ahora termina. Gracias por hacer de esta etapa algo inolvidable y espero cumplan con todas las metas que se propongan de aquí en adelante.



<b>Prefacio</b>	<b>v</b>
<b>Lista de figuras</b>	<b>xii</b>
<b>Lista de cuadros</b>	<b>xiv</b>
<b>Resumen</b>	<b>xv</b>
<b>Abstract</b>	<b>xvii</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Antecedentes</b>	<b>3</b>
2.1. Controladores de vuelo en el mercado . . . . .	3
2.2. Robotarium by Georgia Tech . . . . .	4
2.3. Megaproyecto SEEQ ROBOHELICÓPTERO . . . . .	5
2.4. Diseño e implementación de una red de comunicación Wi-Fi e interfaz gráfica para una mesa de pruebas de robótica de enjambre . . . . .	5
<b>3. Justificación</b>	<b>7</b>
<b>4. Objetivos</b>	<b>9</b>
4.1. Objetivo general . . . . .	9
4.2. Objetivos específicos . . . . .	9
<b>5. Alcance</b>	<b>11</b>
<b>6. Marco teórico</b>	<b>13</b>
6.1. <i>Firmware</i> . . . . .	13
6.2. Componentes de un controlador de vuelo . . . . .	14
6.2.1. Motor BLDC . . . . .	14
6.2.2. Controladores de velocidad electrónicos . . . . .	15
6.2.3. Unidad de medición inercial . . . . .	16
6.2.4. Giroscopio . . . . .	17

6.2.5.	Acelerómetro . . . . .	18
6.2.6.	Magnetómetro . . . . .	19
6.2.7.	ESP32 . . . . .	19
6.2.8.	Regulador de voltaje . . . . .	20
6.3.	Baterías . . . . .	21
6.3.1.	Polímero de Litio (LiPo) . . . . .	21
6.3.2.	Ion de Litio (Li-Ion) . . . . .	23
6.4.	Selección de componentes . . . . .	24
6.4.1.	<i>Power Budget</i> . . . . .	24
6.4.2.	<i>Trade Study</i> . . . . .	24
6.5.	Algoritmo de Madgwick . . . . .	27
6.5.1.	Integración de velocidades angulares . . . . .	27
6.5.2.	Corrección de cuaterniones . . . . .	28
6.5.3.	Fusión de estimaciones . . . . .	28
6.5.4.	Normalización cuaternión . . . . .	28
6.6.	<i>Software</i> . . . . .	29
6.6.1.	Altium Designer . . . . .	29
6.6.2.	PlatformIO . . . . .	29
29	section.82	
6.7.1.	Wi-Fi . . . . .	29
6.7.2.	Medición de pose . . . . .	31
<b>7.</b>	<b>Desarrollo de <i>Firmware</i></b>	<b>33</b>
7.1.	Motores BLDC . . . . .	33
7.1.1.	Relación RPM-ancho de pulso PWM . . . . .	36
7.1.2.	Modelo experimental . . . . .	37
7.1.3.	Resultados . . . . .	37
7.2.	MPU-9250 . . . . .	42
7.2.1.	Librería I2C . . . . .	42
7.2.2.	Librería MPU-9250 . . . . .	43
7.2.3.	Librería magnetómetro . . . . .	50
7.2.4.	Modelo experimental . . . . .	54
7.2.5.	Resultados . . . . .	54
<b>8.</b>	<b><i>Hardware</i></b>	<b>57</b>
8.1.	Batería . . . . .	57
8.1.1.	Consumo de corriente motores . . . . .	58
8.1.2.	<i>Trade Study</i> . . . . .	59
8.1.3.	Modelo experimental . . . . .	78
8.1.4.	Resultados . . . . .	78
8.2.	PCB . . . . .	79
8.2.1.	Componentes . . . . .	79
8.2.2.	Esquemático . . . . .	81
8.2.3.	<i>Wire routing</i> . . . . .	82
8.2.4.	Manufactura y validación . . . . .	84

<b>9. Ecosistema Robotat</b>	<b>87</b>
9.1. Red de comunicación . . . . .	88
9.1.1. Aplicación Python . . . . .	88
9.1.2. Librería Robotat . . . . .	89
9.2. Implementación del cuadricóptero al ecosistema Robotat . . . . .	90
9.2.1. Colocación de marcadores . . . . .	90
9.2.2. Código . . . . .	92
9.2.3. Modelo experimental . . . . .	94
9.2.4. Resultados . . . . .	94
<b>10. Conclusiones</b>	<b>97</b>
<b>11. Recomendaciones</b>	<b>99</b>
<b>12. Bibliografía</b>	<b>101</b>
<b>13. Anexos</b>	<b>105</b>
13.1. Repositorio GitHub . . . . .	105
13.2. Componentes cuadricóptero . . . . .	105
13.2.1. PCB . . . . .	106



---

## Lista de figuras

---

1.	Componentes motor BLDC [12]. . . . .	14
2.	Funcionamiento motor BLDC [12]. . . . .	15
3.	Circuito ESC [14]. . . . .	16
4.	Calibración acelerómetro. . . . .	18
5.	Curva de descarga para baterías LiPo [32]. . . . .	22
6.	Química de batería Li-Ion y detalle del cátodo [34]. . . . .	23
7.	Diagrama de flujo para el análisis de decisiones [36]. . . . .	25
8.	Ancho de pulso para velocidad de 0 RPM. . . . .	37
9.	Ancho de pulso para velocidad de 12000 RPM. . . . .	38
10.	Colocación de cinta reflectiva para la medición con tacómetro. . . . .	38
11.	Regresión lineal de velocidades obtenidas con tacómetro para motor 1. . . . .	39
12.	Regresión lineal de velocidades obtenidas con tacómetro para motor 2. . . . .	40
13.	Comparación de velocidades medidas para dos motores diferentes. . . . .	41
14.	Comunicación I2C entre los 3 periféricos. . . . .	50
15.	<i>Interface Bypass Mux</i> para la interconexión de los buses I2C. . . . .	51
16.	Resultado de variación de ángulo <i>roll</i> . . . . .	54
17.	Resultado de variación de ángulo <i>pitch</i> . . . . .	55
18.	Resultado de variación de ángulo <i>yaw</i> . . . . .	55
19.	Relación RPM-Consumo de corriente. . . . .	58
20.	Opción 1 batería tipo Li-Po. . . . .	59
21.	Opción 2 batería tipo Li-Po. . . . .	60
22.	Opción 3 batería tipo Li-Po. . . . .	60
23.	Opción 4 batería tipo Li-Po. . . . .	61
24.	Opción 5 batería tipo Li-Po. . . . .	61
25.	Opción 1 batería tipo Li-Ion 18650. . . . .	62
26.	Opción 2 batería tipo Li-Ion 18650. . . . .	62
27.	Opción 3 batería tipo Li-Ion 18650. . . . .	63
28.	Opción 1 batería tipo NiMH. . . . .	63
29.	Opción 2 batería tipo NiMH. . . . .	64
30.	Opción 3 batería tipo NiMH. . . . .	64
31.	Opción 4 batería tipo NiMH. . . . .	65

32.	Opción 1 batería tipo NiCd. . . . .	65
33.	Opción 2 batería tipo NiCd. . . . .	66
34.	Gráfica de ponderación para criterios <i>trade study</i> . . . . .	68
35.	Rendimiento de cada alternativa para el criterio de peso <i>trade study</i> . . . . .	69
36.	Rendimiento de cada alternativa para el criterio de tiempo de vuelo <i>trade study</i> . . . . .	70
37.	Rendimiento de cada alternativa para el criterio de disponibilidad <i>trade study</i> . . . . .	71
38.	Rendimiento de cada alternativa para el criterio de corriente <i>trade study</i> . . . . .	72
39.	Rendimiento de cada alternativa para el criterio de cargador <i>trade study</i> . . . . .	73
40.	Rendimiento de cada alternativa para el criterio de voltaje <i>trade study</i> . . . . .	74
41.	Rendimiento de cada alternativa para el criterio de tiempo de vida <i>trade study</i> . . . . .	75
42.	Rendimiento de cada alternativa para el criterio de costo <i>trade study</i> . . . . .	76
43.	Resultados finales <i>trade study</i> . . . . .	77
44.	Relación RPM-Consumo de corriente real. . . . .	79
45.	Esquemático de la PCB. . . . .	82
46.	Cálculo de ancho de pista y espacio de vía para los nodos VCC, GND y SWITCH. . . . .	82
47.	Cálculo de ancho de pista y espacio de vía para el nodo 5V. . . . .	83
48.	Cálculo de ancho de pista y espacio de vía para los nodos SCL, SDA, PWM y 3V3. . . . .	83
49.	Vista de la capa inferior de la PCB desde Altium Designer. . . . .	84
50.	Vista 3D de la PCB desde Altium Designer . . . . .	84
51.	Vista superior placa manufacturada. . . . .	85
52.	Vista inferior placa manufacturada. . . . .	85
53.	Ecosistema Robotat CIT 116. . . . .	87
54.	Interconexión entre el sistema de captura de movimiento OptiTrack y el cuadricóptero. . . . .	90
55.	Cuadricóptero con marcadores para sistema OptiTrack. . . . .	90
56.	Primera prueba lectura de marcadores OptiTrack en cuadricóptero. . . . .	91
57.	Segunda prueba lectura de marcadores OptiTrack en cuadricóptero. . . . .	91
58.	Cuadricóptero con orientación $yaw = 0$ . . . . .	94
59.	Cuadricóptero en rotación sobre ángulo $yaw$ . . . . .	95
60.	Cuadricóptero con orientación $yaw = 90^\circ$ . . . . .	95
61.	Motor BLDC A2212/13T 1000kv. . . . .	105
62.	30A BLDC ESC. . . . .	105
63.	Módulo Wi-Fi + Bluetooth ESP32. . . . .	106
64.	Módulo MPU-9250. . . . .	106
65.	Bornera de 3 pines, 300V a 15A. . . . .	106
66.	Bornera de 2 pines, 300V a 15A. . . . .	106
67.	Header tipo hembra. . . . .	107
68.	Conector macho a placa de 3 pines. . . . .	107
69.	Switch 3 pines ON/OFF, 20A 12DC. . . . .	107



---

## Lista de cuadros

---

1.	Voltaje batería Li-Ion según el material del cátodo. . . . .	24
2.	Comparación de métodos para el análisis de decisión [36]. . . . .	25
3.	Atributos del objeto <i>bldc_esc_t</i> . . . . .	34
4.	Funciones del objeto <i>bldc_esc_t</i> . . . . .	35
5.	Comparación de velocidad en código contra la velocidad real medida para motor 1. . . . .	39
6.	Comparación de velocidad en código contra la velocidad real medida para motor 2. . . . .	40
7.	Funciones comunicación I2C. . . . .	43
8.	Atributos del objeto <i>vector_t</i> . . . . .	43
9.	Atributos del objeto <i>calibration_t</i> . . . . .	44
10.	Funciones inicialización MPU-9250 . . . . .	46
11.	Funciones para lectura de datos del giroscopio y acelerómetro. . . . .	48
12.	Funciones para impresión en consola de configuraciones realizadas sobre la MPU-9250. . . . .	50
13.	Funciones para inicialización, lectura de datos e impresión de configuraciones de sensor AK8963. . . . .	53
14.	Mediciones consumo de corriente de motores BLDC con fuentes en paralelo. . . . .	58
15.	Peso asignado para cada criterio según métodos de ponderación. . . . .	68
16.	Rendimiento de cada alternativa para el criterio de peso. . . . .	69
17.	Rendimiento de cada alternativa para el criterio de tiempo de vuelo. . . . .	70
18.	Rendimiento de cada alternativa para el criterio de disponibilidad. . . . .	71
19.	Rendimiento de cada alternativa para el criterio de corriente. . . . .	72
20.	Rendimiento de cada alternativa para el criterio de cargador. . . . .	73
21.	Rendimiento para cada alternativa para el criterio de voltaje. . . . .	74
22.	Rendimiento para cada alternativa para el criterio de tiempo de vida. . . . .	75
23.	Rendimiento para cada alternativa para el criterio de costo. . . . .	76
24.	Resultados finales <i>trade study</i> . . . . .	77
25.	Mediciones consumo de corriente de motores BLDC con batería seleccionada. . . . .	78
26.	Componentes PCB. . . . .	80
27.	Costos componentes PCB. . . . .	80

28. Anchos y separación de pistas utilizadas para el proceso de <i>wire routing</i> de la PCB. . . . .	83
29. Funciones librería Robotat utilizadas en el cuadricóptero. . . . .	89

El diseño de un controlador de vuelo involucra el desarrollo en conjunto de *software* y *hardware*. En este trabajo se busca que el controlador permita la interacción entre motores BLDC, los módulos ESP32, MPU-9250 y el sistema de captura de movimiento OptiTrack. Estos dos últimos permiten realizar la medición de la orientación del cuadricóptero y así cerrar el lazo de control, el cual fue desarrollado por el estudiante Carlos Alonzo. El proceso de comunicación entre los diferentes componentes se logra a través de los protocolos I2C y MQTT. Adicionalmente, se realiza el diseño de una placa electrónica, la cual se encontrará montada sobre la plataforma y será la encargada de distribuir la alimentación y señales de control respectivas.

Se realizaron diferentes mediciones con el módulo MPU-9250 con las cuales se lograron obtener resultados positivos en el cálculo de los ángulos *roll* y *pitch* y *yaw*, confirmando el correcto funcionamiento de la librería. Del mismo modo, se obtuvieron los resultados esperados con respecto a los anchos de pulso de las señales PWM generadas por el código para el control de los motores BLDC. Se producen los anchos de pulso mínimo y máximo requerido para sus correspondientes velocidades.

Entre los elementos involucrados en un plataforma de vuelo se encuentra la batería, la cual en este caso fue de tipo Li-Ion. Esta batería fue seleccionada considerando varios criterios y los diferentes niveles de importancia de cada uno. Entre las características más relevantes de la alternativa seleccionada son un voltaje nominal de 12V, capacidad de 6800 mAh y un peso de 287 gramos, entre otros. Luego de hacer pruebas con los motores a diversas velocidades se logró comprobar que la capacidad de la batería es más que suficiente para cubrir con la demanda de energía que se requiere para esta aplicación.



Flight controller design involves the joint development of *software* and *hardware*. In this work it is sought that the controller allows the interaction between BLDC motors, the ESP32, MPU-9250 modules and the OptiTrack motion capture system. These last two allow the measurement of the orientation of the quadcopter and thus close the control loop, which was developed by the student Carlos Alonzo. The communication process between the different components is achieved through the I2C and MQTT protocols. In addition, an electronic board was designed. This board will be mounted on the platform and will be in charge of distributing the power supply and the respective control signals.

Different measurements were carried out with the MPU-9250 module with which positive results were obtained in the calculation of the angles *roll* and *pitch* and *yaw*, confirming the correct functioning of the library. In the same way, the expected results were obtained with respect to the pulse widths of the PWM signals generated by the code for the control of BLDC motors. The minimum and maximum pulse widths required for their corresponding speeds are produced.

Among the elements involved in a flight platform is the battery, which in this case was of the Li-Ion type. This battery was selected considering several criteria and the different levels of importance of each one. Among the most relevant characteristics of the selected alternative are a nominal voltage of 12V, a capacity of 6800 mAh and a weight of 287 grams, among others. After testing the motors at different speeds, it was possible to verify that the battery capacity is more than enough to meet the energy demand required for this application.



En los últimos años el uso de los vehículos aéreos no tripulados se ha extendido a diversas áreas, como lo es la industria o el entretenimiento por ejemplo. Esto ha provocado un aumento en el interés y desarrollo de estas plataformas. Sin embargo, dicho desarrollo debe ser complementado por la academia e investigación, la cual en Guatemala es inexistente hoy en día. Anteriormente se realizó en la Universidad del Valle de Guatemala el megaproyecto SEEQ ROBOHELICÓPTERO, cuyo propósito era iniciar con esta línea de investigación. Por ciertos limitantes logísticos y económicos, el proyecto fue abandonado luego de cierto tiempo.

Considerado como la continuación del megaproyecto mencionado anteriormente, se plantea la elaboración de un cuadricóptero de bajo costo. Debido a la complejidad que esto implica y la cantidad de tiempo disponible, este proyecto se realiza en conjunto con el estudiante Carlos Alonzo. Por lo tanto, el presente trabajo de graduación tiene como objetivo general el diseño y manufactura del controlador de vuelo, el cual debe ser capaz de integrarse inalámbricamente con el sistema de captura de movimiento disponible en el laboratorio del Centro de Innovación y Tecnología (CIT). El desarrollo de este objetivo se detalla en los capítulos 7, 8 y 9 respectivamente.

Los capítulos 7 y 9 contemplan la parte de *software*, mientras que el capítulo 8 trata sobre la sección de *hardware*. En los dos primeros capítulos mencionados anteriormente se explica todo el *firmware* involucrado y la implementación del protocolo MQTT. Esto incluye el control e inicialización de los motores BLDC, la lectura de datos de una MPU-9250 y la comunicación de la plataforma con el sistema de captura de movimiento. En el capítulo restante se detalla todo el proceso de selección, *trade study*, para la batería que alimentará a toda la plataforma. También se especifica el proceso de diseño y manufactura de la placa electrónica encargada de la distribución de potencia y señales de control hacia todos los componentes. En dicho proceso de diseño fue necesario realizar un análisis detallado sobre el ancho y separación de pistas mínimos requeridos para algunos nodos debido a la cantidad de corriente que llegan a consumir los motores a ciertas velocidades.





Una de las partes fundamentales en el diseño y control de un *drone* es su controlador de vuelo, el cual viene siendo su placa madre y es el responsable de todos los comandos que el piloto emite al sistema [1]. Es decir, es una placa de circuitos integrados con sensores para la detectar cambios de orientación. Al mismo tiempo se encarga de comunicarse con los controladores electrónicos de velocidad (ESC) para la implementación y manejo de los motores correspondientes. Dependiendo del tipo del controlador, este puede permitir la conexión de diferentes periféricos como GPS, sensores ultrasónicos, etc.

## 2.1. Controladores de vuelo en el mercado

En el mercado existe una gran variedad de controladores de vuelo para el diseño de *drones*, los cuales varían en sus características y precio según la aplicación deseada para el sistema. Algunos ejemplos de estos controladores de vuelo comerciales son el Pixhawk PX4, DJI NAZA-M V2, LHI Pro Racing F3, TauLabs Sparky 2.0, Hobbypower Mini KK2.15, ArduPilot Mega, entre otros [2]. A pesar de que estos controladores ofrecen grandes ventajas ya que son plataformas dedicadas, presentan algunos inconvenientes al momento de realizar proyectos de investigación, principalmente que no permiten efectuar una modificación profunda del *firmware*.

En el año 2019, se publica *Big Unmanned Box-wing Operation & Bigger Unmanned Box-wing Operation (BUBO BUBO), Conceptual Design Document* por parte de *University of Colorado, Department of Aerospace Engineering Sciences*. En este documento se explica el proceso de selección de componentes para una aeronave controlada por radio. Los desarrolladores plantean los criterios más relevantes para una correcta selección del controlador de vuelo, y validan este proceso utilizando la herramienta *trade study*. Estos criterios son la optimización, soporte disponible, complejidad de configuración, tiempo de respuesta de los controladores, peso y precio de los mismos [3].

El nivel de optimización de un controlador de vuelo variará según el tipo de sistema a implementar (cuadricóptero, hexacóptero, etc) ya que los controladores optimizados deberán tener las suficientes entradas/salidas para comunicarse con el sistema de propulsión y control del sistema. Al mismo tiempo, el *software* requerirá poca modificación para implementar los algoritmos de control deseados. El soporte y documentación otorgada por los desarrolladores/comunidad es vital ya que con esto será posible comprender correctamente la configuración del controlador o resolver cualquier complicación que pueda surgir. Un nivel de soporte más alto implica un menor consumo de tiempo para la configuración. De igual forma, el controlador de vuelo debe tener una configuración simple para las salidas e implementación de otros sistemas que conforman al *drone*. Posiblemente el criterio con mayor relevancia es el tiempo de respuesta de control ya que esto afectará directamente el desempeño del sistema completo. Caso contrario al anterior, el criterio del peso es el que menor relevancia tiene. Es recomendable que el peso del controlador de vuelo no afecte en gran medida a otros parámetros de diseño, sin embargo, no es un factor que pueda llegar a disminuir el rendimiento general. Finalmente se tiene el precio, el cual es bastante relevante, especialmente cuando el proyecto a ejecutar está enfocado a la academia. Si el controlador de vuelo es costoso y/o debe importarse del extranjero, será muy difícil llegarlo a reemplazar si en caso este presenta algún problema o se daña durante la ejecución de pruebas [3].

Debido a que en investigación no se desea una plataforma tan completa y general como lo son los controladores de vuelo comerciales, se opta por el proceso de diseñar propiamente dicho controlador. Esto puede evidenciarse en proyectos desarrollados por *The General Robotics, Automation, Sensing and Perception (GRASP) Laboratory* de *University of Pennsylvania, School of the Engineering and Applied Sciences*. Uno de estos proyectos publicado como *Flight Performance of a Swashplateless Micro Air Vehicle* [4] se especifica que los controladores personalizados que se utilizaron tienen la capacidad de comunicación y control con los motores y de extraer las mediciones inerciales, todo esto a través de un microcontrolador. Plataformas de desarrollo como los *Crazyflie* [5] también cuentan con controladores de vuelo con una estructura similar a la mencionada anteriormente.

## 2.2. Robotarium by Georgia Tech

El Robotarium de Georgia Institute of Technology es una plataforma que se encuentra accesible para cualquier persona y permite la implementación de forma remota de algoritmos para robótica de enjambre. Esta área de investigación es considerablemente costosa y por ende en la mayoría de los casos se opta por dejar los resultados en simulación. Por lo tanto, el enfoque principal de esta plataforma es permitir a cualquier persona con interés en la robótica ver la implementación de sus algoritmos en múltiples robots de última generación. En el desarrollo de este proyecto se tienen factores relevantes que fueron tomadas en cuenta como por ejemplo la recolección de información de forma eficiente y sencilla, minimización de costo individual por cada agente que se encuentra en la plataforma y realizar un sistema escalable según la cantidad de agentes [6].

## **2.3. Megaproyecto SEEQ ROBOHELICÓPTERO**

El Megaproyecto realizado por Mario Búrbano, José Morales, Melisa Sandoval, Roberto Saravia y Daniel Suazo en el año 2011 consiste en el diseño y control de movimiento de un cuadricóptero. Para lograr esto el proyecto se dividió en cinco grandes módulos: estructura, estabilización y control, sensor e interfaz RS-232, mapeo y exploración y visualización en 3D. [7]. Como resultado final se obtuvo una sistema con vuelo estable y con capacidad de movimiento considerablemente buena, lo suficiente incluso para evadir obstáculos. A pesar de que no se obtuvieron resultados completamente favorables en algunos de los cinco módulos mencionados anteriormente y los derivados de los mismos, este megaproyecto tiene un gran impacto ya que surgió por la motivación de abrir una nueva línea de investigación en la Universidad del Valle de Guatemala.

## **2.4. Diseño e implementación de una red de comunicación Wi-Fi e interfaz gráfica para una mesa de pruebas de robótica de enjambre**

El trabajo de graduación realizado por Castañeda [8] en el año 2020 plantea y desarrolla de una forma eficiente la implementación de una red de comunicación inalámbrica a través del protocolo MQTT. Este protocolo de comunicación, además de permitir una gran flexibilidad en el diseño de topología de red, otorgó resultados óptimos para envíos de paquetes pequeños de información. Esta flexibilidad concede la capacidad de que los agentes dentro del sistema puedan comunicarse entre ellos o a una unidad de cómputo con mejores recursos, lo cual permite realizar simulaciones y rutinas complejas para la investigación y desarrollo de robótica de enjambre.

Además de establecer las bases para lograr un proceso de comunicación eficiente, el trabajo de Castañeda desarrolló cierta logística para el funcionamiento de los diversos agentes dentro del sistema. Ya que el presente trabajo de graduación se enfoca principalmente en el diseño del controlador de vuelo, es relevante considerar el flujo de trabajo planteado para la comunicación y ejecución de controladores por parte de los agentes [8]. Con todo esto se tienen los principios de implementación del cuadricóptero a través de comunicación inalámbrica.



---

### Justificación

---

Uno de los desafíos planteados para este trabajo de graduación es la integración del cuadricóptero en el ecosistema Robotat a través del sistema de captura de movimiento OptiTrack. El Robotat es la denominación dada al ecosistema que se encuentra en desarrollo actualmente en el laboratorio de robótica en el CIT, el cual está inspirado en otras plataformas presentes en universidades de primer mundo como lo es el Robotarium. Este ecosistema plantea la integración de diferentes plataformas robóticas que tengan la capacidad de poder interactuar entre sí de forma eficiente con comunicación inalámbrica.

Actualmente el rango de aplicaciones que puede tener un *drone* pueden ser de tipo industrial (fotografía e inspección), seguridad (detección y vigilancia), logística (envíos de paquetería), medios de comunicación (eventos y noticias), desastres naturales (búsqueda de personas en áreas de difícil acceso) e investigación (arqueológica, geológica y biológica), entre otros. En los últimos años en Guatemala ha brotado una fascinación y entusiasmo por estas plataformas, ya sea como entretenimiento o para la industria. Algunas empresas han optado utilizarlos para obtener videos y/o fotografías enfocados a *marketing* o para verificación de cultivos en grandes extensiones de tierra. Sin embargo, esta creciente fascinación no va acompañada de un componente de investigación formal en la academia. Tal y como se menciona en los antecedentes, años atrás la Universidad del Valle de Guatemala empezó a tratar este tema de investigación con el megaproyecto SEEQ ROBOHELICÓPTERO, el cual fue bastante innovador en su época ya que estas plataformas eran relativamente nuevas en el país.

A pesar de que el megaproyecto mencionado anteriormente abrió una nueva línea de investigación, esta se dejó abandonada a través de los años. Por ende este trabajo de graduación, el cual trata sobre el diseño y manufactura de un controlador de vuelo para un cuadricóptero de bajo costo, puede considerarse la primera continuación formal de este megaproyecto donde con uso de tecnología moderna será posible acoplarse a las necesidades del mundo actual. Es por ello que como universidad y país se tenga un desarrollo más constante y formal en esta área, lo cual se facilitará con el nuevo ecosistema Robotat. Ya que este cuadricóptero será destinado a la academia e investigación, los controladores de vuelo

comerciales no son la mejor opción tal como se mencionó en los antecedentes.

En investigación se requiere una plataforma que sea de propósito específico y no tan general como lo son los controladores de vuelo disponibles en el mercado. Integrar estos últimos al cuadricóptero deseado para el Robotat añadiría cierto grado de complejidad en varios aspectos y terminaría ralentizando todo el proceso de desarrollo. Sería necesario conocer y familiarizarse con el proceso de configuración del controlador antes de lograr implementar lo necesario para cumplir con el objetivo de investigación. Al mismo tiempo, ya que el proceso de investigación se haría en un ambiente controlado, muchas de las funcionalidades de estos controladores se verían desperdiciadas. Añadido a esto es necesario tomar en cuenta el aspecto económico. Al tratarse de dispositivos que se verán sometidos a diversas pruebas es preferible no optar por este tipo de controlador para que el impacto en la inversión realizada no sea tan alto al momento que se produzcan fallas.

Partiendo de lo mencionado anteriormente, este trabajo de graduación presenta el diseño e implementación de un controlador de vuelo de bajo costo que se acopla a la visión de investigación y desarrollo deseado para el ecosistema Robotat. Este controlador de vuelo se fabricará con componentes disponibles en el mercado local para que sea fácil de reemplazar y manufacturar en caso sea necesario. La importancia de esta investigación radica en que se potenciará el desarrollo de estas plataformas y abrirá campo para seguir realizando investigaciones más complejas en la universidad en un futuro cercano.

### 4.1. Objetivo general

Diseñar y manufacturar un controlador de vuelo para un cuadricóptero capaz de integrarse inalámbricamente al sistema de captura de movimiento del ecosistema Robotat.

### 4.2. Objetivos específicos

- Desarrollo e implementación del *firmware* para el manejo de motores BLDC mediante controladores electrónicos de velocidad y la obtención de variables de orientación de una unidad de medición inercial.
- Diseño y manufactura de una *Printed Circuit Board* (PCB) dedicada a la electrónica de potencia y control para un cuadricóptero capaz de integrarse en el ecosistema Robotat.
- Integración del sistema a un entorno de captura de movimiento a través de comunicación inalámbrica Wi-Fi en el microcontrolador ESP32 basada en el protocolo MQTT.





Este trabajo se enfoca en la elaboración de tres áreas del cuadricóptero para el ecosistema Robotat. La primera se encarga en el desarrollo del *firmware* necesario para ejercer control sobre la plataforma. Esto se logra obteniendo la lectura de datos de una MPU-9250 y controlando motores BLDC a través de ESC. Esto se implementa a través de programación orientada a objetos en lenguaje C. La segunda se centra en la selección de la batería para la alimentación de la plataforma bajo ciertos criterios específicos, además del diseño y manufactura de una placa electrónica. Esta placa esta diseñada para una corriente máxima de 11.5A y voltaje pico de 12V. Finalmente, la tercera permite la integración del sistema al entorno de captura de movimiento OptiTrack vía Wi-Fi a través del protocolo de comunicación MQTT.

Estas tres áreas se encuentran complementadas con el trabajo del estudiante Carlos Alonzo. En este otro trabajo se diseño y manufacturo el marco estructural de la plataforma, así como la programación del controlador de vuelo que permitirá al sistema alcanzar el estado de flotación o *hovering* en inglés. Cabe resaltar que el *drone* desarrollado no podrá ser teleoperado y requiere que se encuentre dentro del ecosistema Robotat para efectuar tareas útiles. Dicho en otras palabras, implica que no es autónomo. Al mismo tiempo se cuenta con un rango limitado de velocidades permitidas para su operación debido principalmente por el excesivo consumo de corriente que presentan los motores A2212/13T 1000KV utilizados para la implementación.

A causa del de la pandemia SARS-CoV-2, y el aumento repentino de casos positivos en los últimos meses del presente año, el progreso respectivo para el tercer objetivo específico planteado se atrasó. El acceso al campus para realizar trabajo en el laboratorio esta en función del semáforo de nivel de alerta para el municipio de Guatemala. Añadido a esto, la plataforma de pruebas en el CIT 116 tuvo varios atrasos durante todo el año. Es por estas razones que las pruebas con el ecosistema Robotat se vieron limitadas.



### 6.1. *Firmware*

El *firmware* se puede definir como la capa de comunicación entre la electrónica que conforma un dispositivo y los agentes/elementos externos del mismo, es decir, controla y envía instrucciones externas del equipo y garantiza que estas se ejecuten de forma correcta. Específicamente consiste en un bloque de instrucciones de código el cual está dirigido para propósitos específicos. Este código es guardado en una memoria no volátil como lo es la FLASH, ROM o EEPROM y fija una lógica del más bajo nivel para el control de circuitos electrónicos [9]. Este concepto tiene relacionada una parte de *hardware* y otra de *software*. Es parte de *hardware* ya que se integra en los circuitos que conforman al dispositivo, sin embargo también es *software* ya que la lógica se encuentra programada en un lenguaje de programación [10].

El *firmware* cumple con cuatro funciones específicas:

- Control y gestión la inicialización del dispositivo.
- Ejercer como intermediario entre la parte electrónica y los elementos de un dispositivo.
- Determinar rutinas para el funcionamiento.
- Determinar rutinas de respuesta.

Otra de las características principales del *firmware* es que este permite realizarle actualizaciones con el objetivo de añadirle al sistema nuevas funciones o bien optimizar las que ya se encuentran presentes. La actualización depende de las mejoras o cambios que se aporten al sistema, como por ejemplo si influye directamente en el rendimiento en el sistema [10].

## 6.2. Componentes de un controlador de vuelo

### 6.2.1. Motor BLDC

El motor DC sin escobilla o *Brushless Direct Current* (BLDC) es un tipo de motor eléctrico que ha ganado popularidad ya que, como su nombre lo indica, al no necesitar escobillas para la conmutación como los motores tradicionales se reduce significativamente el desgaste. Estas escobillas son reemplazadas por un controlador electrónico que optimiza la durabilidad y rendimiento. Presenta diversas ventajas sobre los motores de inducción o los motores DC con escobillas [11], algunas de estas son las siguientes:

- Alta eficiencia.
- Funcionamiento silencioso.
- Mayor vida útil.
- Rangos de velocidad mayores.
- Mejores características de velocidad-torque.
- Presentan la misma potencia en la salida, pero siendo más pequeños y livianos, lo que los hace ideales para aplicaciones donde el peso y tamaño son críticos.

Son considerados como motores de tipo síncrono, es decir, que los campos magnéticos del rotor y estator rotan a la misma frecuencia. A pesar de que pueden presentarse en diversas configuraciones, el más común y utilizado es el de tres fases, lo cual implica que son tres bobinados independientes en el estator. El rotor de estos motores está conformado por un conjunto de imanes permanentes con pares de polos N-S (North-South). El campo magnético constante creado por los imanes es lo que evita la utilización de escobillas y bobinados en el rotor para la conducción de corriente. El estator es la parte fija y se encuentra internamente, mientras que el rotor hace referencia a la parte móvil y es la carcasa unida al eje del motor [12].

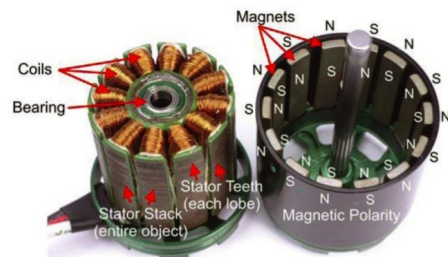


Figura 1: Componentes motor BLDC [12].

### Funcionamiento

Al momento de pasar corriente por una bobina en el estator esta creará un campo magnético. El rotor, el cual tiene un campo magnético fijo debido a sus imanes, tiende a alinearse

ambos campos magnéticos (el creado por el estator y el propio) a través de una rotación. Sin embargo, eso solo genera un ligero movimiento. Para lograr que el rotor continúe girando, antes que se dé la alineación completa entre los campos magnéticos, se deja de alimentar la bobina y se energiza la siguiente, creando así un campo magnético que nuevamente se encuentra desfasado respecto al del rotor [12]. Por lo tanto, se puede decir que a través del proceso de energizado de las bobinas del estator se crea un campo magnético variable en el tiempo el cual provoca que el rotor gire intentando lograr la sincronización. Esto implica que el control de estos motores se logra a través de la transmisión de series de pulsos en las fases, como se muestra en la Figura 2.

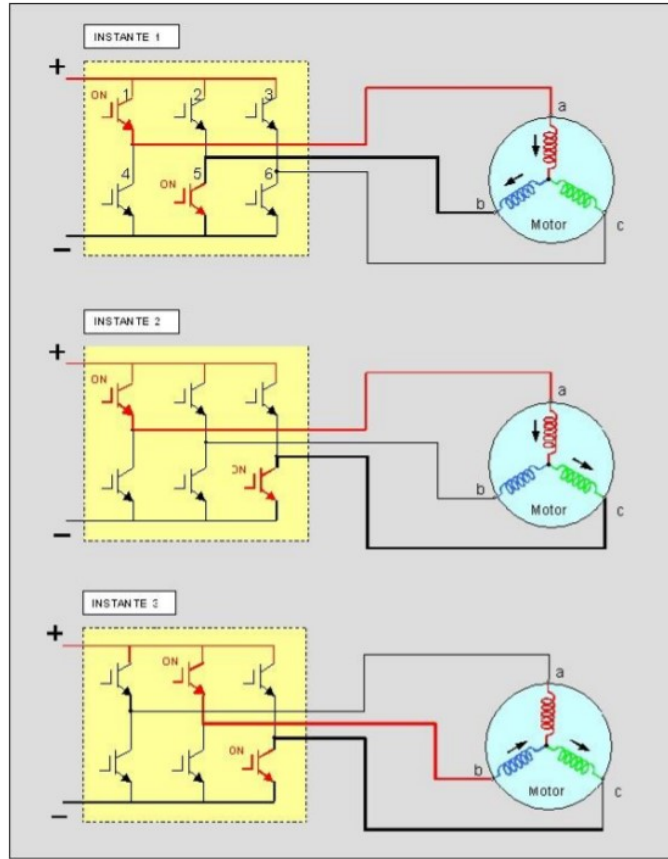


Figura 2: Funcionamiento motor BLDC [12].

Como se puede observar, en cada etapa de conmutación se energiza una de las bobinas de forma positiva, la segunda negativa y la tercera permanece abierta [13]. La velocidad del campo magnético variable está dictada por la frecuencia con que se envían los pulsos de alimentación por fase.

### 6.2.2. Controladores de velocidad electrónicos

Los controladores de velocidad electrónicos, o, por sus siglas en inglés (ESC), son la conexión entre la fuente de alimentación y el motor. Se encargan del control de la potencia a la salida y la velocidad de rotación del motor en respuesta a los comandos del operador.

En un alto porcentaje de aplicaciones donde son utilizados esos elementos, los ESC cumplen la función de convertir electricidad de corriente continua de una fuente o batería a corriente alterna trifásica, lo cual se logra mediante una conmutación electrónica de alta frecuencia. Internamente los ESC están conformados por seis transistores MOSFET, tal y como se muestra en la Figura 3. Estos transistores son controlados electrónicamente y actúan como interruptores digitales que hacen la conversión a electricidad trifásica modulada por anchos de pulso (PWM) para alimentar a los motores BLDC [14].

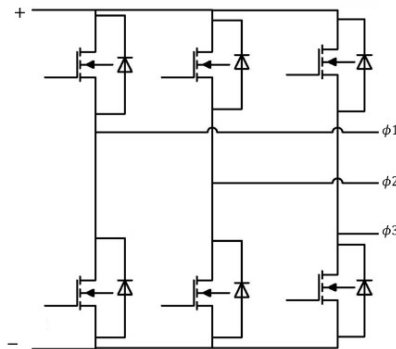


Figura 3: Circuito ESC [14].

Debido a la naturaleza de conmutación de este dispositivo, se tienen pulsos de alta frecuencia en donde el voltaje y la corriente aumentan al momento que se incrementa el ancho de pulso. Por lo tanto, el ESC esta controlado directamente por una señal de PWM en donde el ancho de pulso de la misma modifica la velocidad del motor BLDC respectivamente, al mismo tiempo que la corriente directa en la entrada es proporcional al par de carga del motor [15].

### 6.2.3. Unidad de medición inercial

Una unidad de medición inercial, o, por sus siglas en inglés (IMU), es un dispositivo conformado por magnetómetros, giroscopios y acelerómetros para la medición de la intensidad de campo magnético, velocidad angular, y aceleración lineal respectivamente. Estos dos últimos registran las mediciones en los tres ejes  $x$ ,  $y$ ,  $z$ . Integrando estos datos es posible obtener la pose (posición y orientación) de un objeto en el espacio a corto plazo [16]. Añadido a esto, estos sensores permiten obtener los ángulos de inclinación y balanceo, lo cual es de suma importancia para tareas de control. Es por ello que son bastante utilizados en aplicaciones como la navegación de drones.

Cabe mencionar que una limitante importante con este tipo de dispositivos es el error entre el valor medido y el real de la posición. Esta divergencia es causada principalmente por el ruido en las señales integradas. Para la reducción de este error se ha optado en ocasiones a utilizar múltiples sensores al mismo tiempo o en técnicas basadas en modelos matemáticos. Es por ello que en la práctica generalmente la combinación de los datos inerciales con la medición de la pose se realiza utilizando un filtro de Kalman [17]. Otro tipo de error común es que los acelerómetros y giroscopios tienen una salida diferente de cero aún cuando el sistema se encuentra en reposo. Debido a este conjunto de errores es complicado obtener

resultados precisos utilizando únicamente las mediciones otorgados por la IMU, por lo tanto es necesario realizar un proceso de procesamiento de las señales para obtener resultados confiables y que sean útiles en las aplicaciones [18].

## MPU-9250

Corresponde a una unidad de medición inercial desarrollada por InvenSense. Cuenta con giroscopios para la medición de la velocidad angular en los tres ejes. La salida digital de estos giroscopios tiene un rango de escala programable por el usuario de  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$  y  $\pm 2000$  °/seg, además de un filtro pasa bajas digital igualmente programable. Al mismo tiempo cuenta con un acelerómetro de triple eje con salida digital con un rango de escala programable de  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$  y  $\pm 16g$  y un ADC de 16 bits de resolución integrado. El magnetómetro integrado es un sensor monolítico de efecto Hall de triple eje cuyo rango de medición de escala completa es  $\pm 4800\mu T$ . Para lograr comunicación con otros dispositivos, esta unidad de medición inercial cuenta con comunicación I2C la cual tiene modo rápido de 400 kHz para lograr interactuar con todos los registros [19].

### 6.2.4. Giroscopio

Dispositivo que permite la medición de velocidad angular alrededor de un eje. Los giroscopios se basan en la denominada inercia giroscópica, la cual es la tendencia de un cuerpo que gira a mantener su plano de rotación. Dicho en otras palabras, el giroscopio conserva la orientación del eje de rotación cuando se le aplican fuerzas externas que cambie su pose [20]. Con el paso del tiempo los giroscopios han ido evolucionando; actualmente los más utilizados son los sensores giroscópicos de vibración MEMS, los cuales son los que posee la MPU-9250. Su forma de funcionamiento es al momento de que estos giren alrededor de uno de los ejes el efecto Coriolis provoca una vibración la cual es detectada por un *pickoff* capacitivo. Al amplificar y filtrar la señal obtenida se produce un voltaje el cual es proporcional a la velocidad angular de interés. Finalmente se utilizan convertidores ADC para digitalizar dicho voltaje [19].

## Calibración

La precisión es un factor clave para medir el rendimiento en un instrumento de medición. Una variación significativa en la precisión puede afectar negativamente los resultados finales, que en este caso es la orientación del sistema. Es común que se presenten errores de sesgo en este dispositivo, los cuales agregan una variación fija a la respuesta de salida del sensor. Dicha variación puede provocar que el sistema parezca que esta girando cuando realidad se encuentra estático. El resultado final es una acumulación constante de error de medición de ángulo igual al producto entre el error de sesgo y el tiempo acumulado [21]. Cabe mencionar que el proceso de calibración debe realizarse en un periodo corto de tiempo para considerar que las medidas no presenten *drifting*. Este fenómeno consiste en que el sensor incrementa su error conforme avanza el tiempo [22].

### 6.2.5. Acelerómetro

Dispositivo que permite la medición de aceleración lineal sobre un eje. La convención para los acelerómetros es medir la aceleración en unidades de gravedad (g), la cual corresponde a  $1 g = 9.81 m/s^2$ . Estos producen una salida en forma de voltaje el cual es proporcional a la aceleración sobre la cual esta sometido el objeto [23]. Al igual que los giroscopios, estos instrumentos han evolucionado a los denominados MEMS, sin embargo, estos siguen funcionando bajo el mismo principio de una masa y resorte. Cuando el objeto se acelera, la masa por su inercia desea permanecer estacionaria, lo que provoca que el resorte se estire o comprima creando así una fuerza la cual es medida y corresponde a la aceleración aplicada. Para el caso de los sensores MEMS se tiene un tipo de electrodo para realizar este proceso. El desplazamiento de la masa respecto al electrodo induce una capacitancia diferencial la cual es proporcional a la aceleración [24].

### Calibración

El principio de calibración de un acelerómetro se basa en que la magnitud de la aceleración cuando este se encuentre en reposo debe de ser igual a la gravedad terrestre. El proceso de calibración consiste en aprovechar la aceleración gravitatoria, la cual puede utilizarse en orientación positiva y negativa. Esto permite estimar el error de sesgo y la escala mínima y máxima por eje. Por lo tanto, el procedimiento de calibración es como se muestra en la Figura 5.

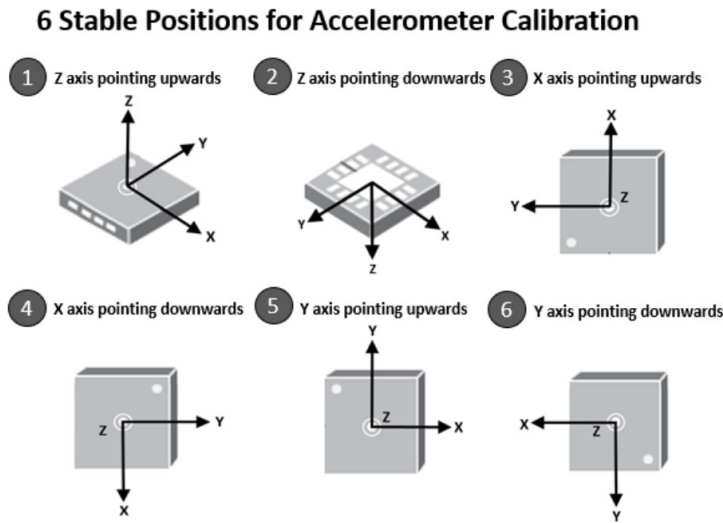


Figura 4: Calibración acelerómetro.



### 6.2.6. Magnetómetro

Dispositivo que cuantifica la dirección e intensidad de un campo magnético. En los sistemas de navegación estos dispositivos, a menos que no tengan de forma cercana una fuente de campo mucho mayor, miden la intensidad de campo magnético de la Tierra. El principio de transducción en este tipo de sensores pueden ser bastante variados dependiendo del fabricante, por ejemplo efecto Hall (principio utilizado en la MPU-9250), fuerza de Lorentz o piezoresistivo [25]. El efecto Hall se produce por la aplicación de un campo magnético transversal sobre un conductor. Debido a que la fuerza magnética es perpendicular al campo y a su velocidad, las cargas son hacia un extremo de conductor y se genera un voltaje.

Una de las desventajas de estos dispositivos es que son considerablemente sensibles a los campos magnéticos generados por los circuitos de lectura o cualquier dispositivo electrónico cercano. Es por ellos que es fundamental que se encuentran correctamente aislados cuando se utilicen.

### Calibración

Debido a que estos dispositivos son sumamente sensibles, es vital realizar un correcto proceso de calibración. En este caso lo que se quiere corregir son los *hard iron errors* y *soft iron errors*. En un diagrama de dispersión de lecturas del magnetómetro, la gráfica ideal debería ser circular y centrada. Sin embargo, los *hard iron errors* provocan que las mediciones se encuentren descentradas y los *soft iron errors* hacen que la gráfica sea elíptica en vez de circular. La forma de eliminar estos errores es encontrando los máximos y mínimos para cada eje del sensor, lo cual se logra recopilando una cantidad considerablemente alta de datos. Por lo tanto, el proceso de calibración consiste en mover el magnetómetro en todas las direcciones para obtener diferentes muestras en todas las orientaciones posibles [26].

Una vez encontrados los máximos y mínimos por eje, es necesario promediar estas medidas. Con esto se obtiene la cantidad de compensación por eje. Para lecturas posteriores es necesario restar el desplazamiento por cada eje y con esto se eliminan los *hard iron errors*. Para los *soft iron errors* existen técnicas avanzadas de corrección que implican el cálculo de matrices de rotación, sin embargo, esto es computacionalmente costoso, por lo tanto, lo que se hace es aplicar un factor de escala en cada eje para regresar la forma elíptica a la circular deseada.

### 6.2.7. ESP32

Es un microcontrolador diseñado por Espressif Systems de bajo consumo energético y bajo costo que cuenta con diversos módulos de comunicación, entre los que destacan el Wi-Fi y Bluetooth. Entre sus principales características de operación se puede mencionar que cuenta con 32 GPIO, 4 módulos SPI, 2 módulos I2C, 18 ADC con doce bits de resolución, 2 DAC de ocho bits, 3 puertos UART, 8 PWM, entre otros. Su voltaje de funcionamiento debe de ser entre 2.2V y 3.6V. Puede ser implementado en diferentes lenguajes de programación, siendo el más popular C y además es de código abierto. Sumado a esto cabe destacar que es un dispositivo de tamaño pequeño pero de alto rendimiento, lo cual lo hace ideal para la

ejecución de proyectos complejos [27].

### 6.2.8. Regulador de voltaje

Consiste en un circuito integrado que tiene como finalidad proporcionar un voltaje constante a su salida libre de ruidos y perturbaciones a pesar de que se presente un cambio en el voltaje de entrada o en la carga. Los reguladores de voltaje mantienen voltajes de una fuente de alimentación en un rango de operación donde es compatible con los demás componentes eléctricos. Generalmente se utilizan en un tipo de conversión DC/DC, sin embargo algunos tienen la capacidad de hacer conversiones AC/DC o AC/AC [28]. En el diseño de dispositivos electrónicos complejos es común que los diversos componentes que lo conforman operen a diferentes voltajes. Es por ello que los reguladores de voltaje sean tan comunes, ya que se encargan de regular el voltaje en una alimentación principal a lo requerido por dichos componentes que conforman al sistema. Hay dos tipos de reguladores de voltaje: lineal y de conmutación, donde los primeros operan a baja eficiencia y los últimos a alta eficiencia.

#### Reguladores lineales

En este tipo de regulador se utiliza un dispositivo de paso activo (transistores BJT o MOSFET) para lograr la regulación del voltaje en la salida. La conexión de la carga, la variación de la misma o de la señal de entrada produce un cambio en la corriente a través del transistor para seguir obteniendo la salida constante. Para el caso de transistores BJT esta corriente hace referencia a la de colector-emisor y debe de estar operando en la región activa/lineal. Cabe mencionar que presenta una gran desventaja, la cual consiste en que desperdicia energía como forma de voltaje, es decir, el voltaje a la salida siempre es menor al voltaje de entrada y esta diferencia es disipada en forma de calor. Esto provoca que su eficiencia sea baja, teniendo un rango entre el 20% y 60%. Sin embargo tiene diversas ventajas que lo hacen bastante atractivo, entre las cuales se puede mencionar que son de bajo costo y cuenta con una amplia gama de selección para corriente y voltaje, son robustos contra fallas térmicas o de sobrecorriente y son fáciles de implementar [29].

#### Reguladores de conmutación

Este tipo de reguladores funcionan diferente a los de tipo lineal. El transistor ahora opera en las regiones de corte y saturación, es decir que su función es la de un interruptor en donde la región de corte representa el estado apagado y la región de saturación el estado de encendido. [29]. A diferencia de los reguladores lineales que son convertidores reductores, los de conmutación pueden ser también convertidores elevadores o una combinación de ambos, lo cual los hace más versátiles [30]. Entre las desventajas de este regulador se encuentra la presencia de ruido e interferencia debido a la conmutación constante del transistor. Sin embargo, presenta ventajas relevantes como lo es su alta eficiencia, la cual puede alcanzar hasta el 95%. Esta alta eficiencia se debe a que el transistor se mantiene cambiando entre los dos estados de encendido y apagado, lo cual implica que permanece poco tiempo en la región activa y por ende disminuye las pérdidas de energía.

## 6.3. Baterías

### 6.3.1. Polímero de Litio (LiPo)

La característica principal de esta batería es la utilización de un electrolito. Un electrolito es una sustancia que esta compuesto por iones libres, lo cual implica que se comporte como un conductor eléctrico. En la batería tipo LiPo la sal del electrolito no se encuentra disuelta, en su lugar, se encuentra depositada en un polímero sólido el cual es generalmente polióxido de etileno [31]. Estas baterías están conformadas por diversas celdas, las cuales tienen un cátodo, un ánodo y el electrolito.

- **Ánodo.** Lámina de litio metálico ultra delgada que puede cumplir la función de colector (durante la carga) o de fuente (durante la descarga) de iones de litio [32].
- **Cátodo.** Está compuesto por diversas capas, las cuales están intercaladas de óxido de vanadio, sal de litio y polímeros. Dichas capas se encuentran laminadas sobre aluminio que funciona como colector [32].

Para estas baterías se utiliza un electrolito el cual se confecciona con una membrana la cual hace de separador entre las láminas de ánodo y cátodo. Este electrolito es un sólido con textura gomosa cuyo material puede variar según el fabricante. Se aplica un gel especial para disminuir impedancias internas (mejor capacidad de descargar) y mejorar capacidades térmicas ya que en ocasiones la temperatura podía llegar a aumentar hasta los 60°C [32].

#### Especificaciones eléctricas

- Bajo porcentaje de de auto-descarga.
- No presentan efecto memoria.
- Alta densidad de energía. Es la propiedad que permite determinar la cantidad de energía que se puede almacenar por kg de batería [32].
- Voltaje nominal por celda de 3.7 V. Están diseñadas para funcionar en un rango de 3V y 4.2 V. Si se llega a descargar por debajo del límite inferior se puede producir una pérdida de rendimiento. Lo más aconsejable es no permitir que que baje de 3.5 V. Caso contrario, cargarla por encima del límite superior puede provocar un aumento de temperatura hasta generar fuego.

#### Velocidad de descarga

Los fabricantes de estas baterías presentan un dato denominado como *C Rating*. Utilizando este valor en conjunto con la capacidad de la batería, es posible determinar la descarga máxima de forma continúa sin comprometer la vida útil de la batería [33].

$$Descarga_{maxima} = CRating \times Capacidad. \quad (1)$$

El problema surge cuando se tiene un valor bajo de *C Rating*. En ese caso la batería no tiene la capacidad para entregar la corriente necesaria a los motores y, por ende, no tendrá la potencia requerida para elevarse.

### Curva de descarga

Conforme se presenta un aumento en la descarga, se disminuye el voltaje por celda, tal y como se muestra en la Figura 5. Es posible observar que todas las curvas presentan el mismo comportamiento descrito anteriormente.

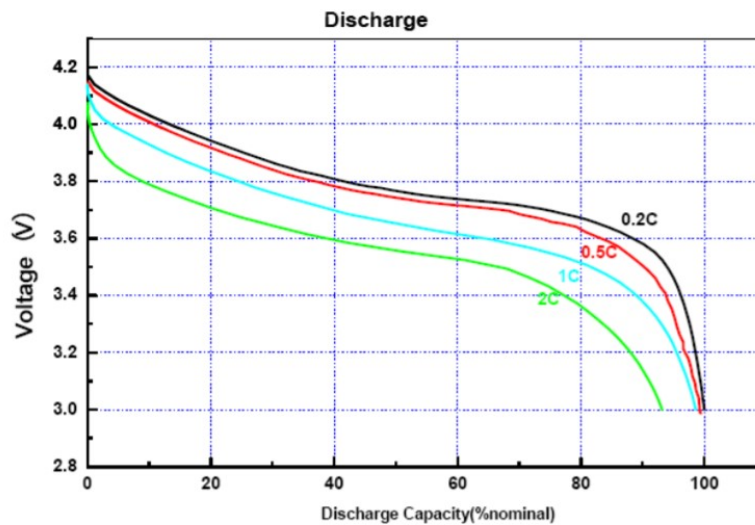


Figura 5: Curva de descarga para baterías LiPo [32].

### 6.3.2. Ion de Litio (Li-Ion)

Las celdas de esta batería, al igual que las tipo LiPo, están conformadas por un ánodo, cátodo y electrolito. El ánodo es el electrodo electropositivo donde se generan los electrones. En este tipo de batería el ánodo contiene litio sustentado en carbono para lograr una reacción fácil a temperatura ambiente y así generar el compuesto de intercalación  $LiC_6$ . [34].

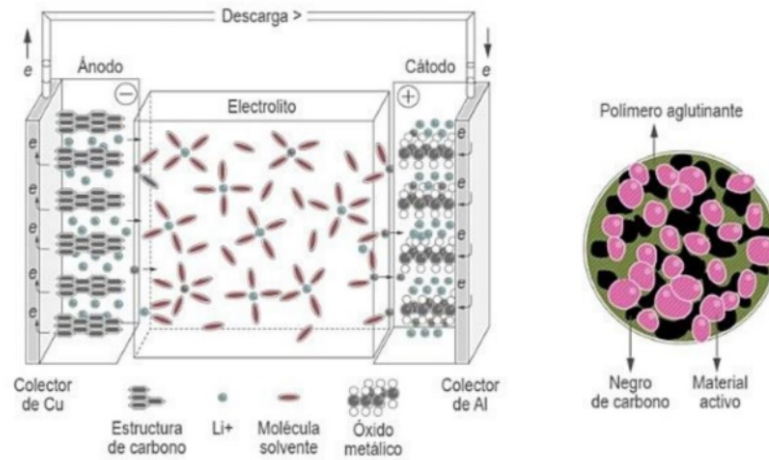


Figura 6: Química de batería Li-Ion y detalle del cátodo [34].

Como se logra apreciar en la Figura 6, el ánodo se encarga de la liberación de iones de litio durante el proceso de descarga. El material de este puede ser C-grafito, silicio, aluminio, estaño, germanio antimonio o indio. El cátodo es el electrodo electronegativo donde migran los iones positivos de la celda y los electrones durante la descarga. Los iones se desplazan a través del electrolito y los electrones vía un circuito externo. Por lo tanto el voltaje de la celda se determina por la energía de la reacción química.

A diferencia de las baterías LiPo, el electrolito utilizado es una solución líquida la cual contiene una sal disuelta en un solvente. Generalmente, el electrolito permite únicamente el flujo de iones y no electrones. Sin embargo, durante el proceso de carga estos fluyen en direcciones contrarias [34].

#### Especificaciones eléctricas

- Bajo porcentaje de auto-descarga (menor al 10%).
- No presentan efecto memoria.
- Voltaje nominal por celda de 3.6-3.7 V.
- Mayor densidad de energía disponible en el mercado.
- Eficiencia entre 85 % y 90 % [32].

A pesar de no presentar efecto memoria, es recomendable no cargar estas baterías cuando presenten más del 50 % de carga. Ya que existen diversas variantes de este tipo de baterías, en el Cuadro 1 se presenta el voltaje nominal según el material del cátodo.

Material cátodo	Voltaje medio (V)
$LiCoO_2$	3.7
$LiMn_2O_4$	4.0
$LiFePO_4$	3.3
$Li_2FePO_4F$	3.6

Cuadro 1: Voltaje batería Li-Ion según el material del cátodo.

## 6.4. Selección de componentes

### 6.4.1. *Power Budget*

Consiste en el cálculo de consumo de energía para un conjunto de dispositivos que estarán conectados a una misma fuente de alimentación [35]. Esto permite tener claras nociones relevantes durante el diseño de sistemas eléctricos como lo es la disponibilidad de energía, duración de la batería en caso se utilice esta como fuente de alimentación, costos ya que a mayor potencia implica un mayor precio, entre otros. Este proceso puede realizarse fácilmente en una hoja de cálculo en Excel. En su forma más general, se deben considerar las siguientes columnas: identificación del componente, la cantidad del componente a utilizar en la placa, el consumo de corriente individual, el consumo de corriente para la cantidad total del componente y el voltaje de suministro. Se realiza el cálculo de potencia a partir de (1).

$$P = IV, \tag{2}$$

donde  $I$  representa la corriente y  $V$  el voltaje. Se repite el mismo proceso para los  $n$  componentes que conforman a la placa. Posterior a esto se suman para obtener la potencia total requerida a la entrada del sistema. A partir de esto se pueden considerar otros factores como la eficiencia de la fuente de alimentación para encontrar la potencia real requerida. Cabe mencionar que también es recomendable diseñar con cierto margen de error ya que en ocasiones las especificaciones de los componentes no son completamente fiables. Es por ello que se debe mantener un equilibrio entre las especificaciones de diseño y los costos. Es recomendable realizar este análisis y cálculo en las fases tempranas del proyecto.

### 6.4.2. *Trade Study*

Los *trade studies* consisten en actividades para la toma de decisiones utilizados principalmente para la identificación de la solución técnica más eficiente y aceptable entre un conjunto de soluciones propuestas. Las decisiones en ocasiones son tomadas con cierto grado de subjetividad, lo cual implica riesgos. Es por ello que los *trade studies* proporcionan un proceso eficaz y objetivo para evitar este problema. Al realizar un proceso de documentación

en la toma de decisiones, tal como sucede en estos casos, se permite marcar una trazabilidad y repetibilidad para la revisión y ejecución de proyectos similares en el futuro. La evaluación de las soluciones propuestas se basa por el nivel de cumplimiento de una serie de características deseables [36]. Este tipo de proceso de decisión es útil cuando se presenta alguno de estos casos:

- Cuando variables como el rendimiento, calendario y costo se deben ponderar.
- Existe más de una solución posible y cada una cuenta con cierto grado de consecuencias o riesgos que son necesarios de evaluar.

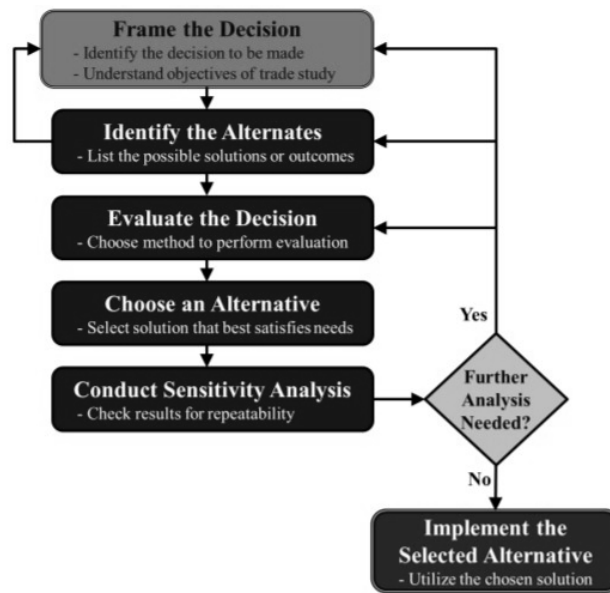


Figura 7: Diagrama de flujo para el análisis de decisiones [36].

Se encuentran disponibles diversas técnicas o métodos para llevar a cabo los *trade studies*. Cada método presenta sus ventajas y desventajas, por lo tanto, la selección de un método depende de factores que deben ser tomados en consideración por el diseñador. Entre estos factores se encuentran la cantidad de datos disponibles, la precisión deseada para la selección de una solución y el tiempo con el que se cuenta para realizar el proceso. Por lo tanto, se presentan a continuación tres de los métodos más comunes, los cuales pueden ser comparados según los criterios mencionados anteriormente.

Método	Tiempo	Precisión	Datos
<b>Pugh</b>	Menor tiempo	Menor precisión	Menos datos
<b>Análítico jerárquico</b>	Mayor tiempo	Mayor precisión	Más datos
<b>Kepner-Tregoe</b>	Mayor tiempo	Mayor precisión	Más datos

Cuadro 2: Comparación de métodos para el análisis de decisión [36].

## **Método Pugh**

También conocido como Método de Decisión Matricial. No es recomendable utilizarlo para la comparación de únicamente 2 opciones; en la práctica se llegan a comparar generalmente entre 3 a 7 opciones diferentes. En este método se colocan inicialmente los criterios de comparación, los cuales deben ser una medida objetiva y cuantificable. Posteriormente se establece la ponderación de cada uno de estos criterios. Se pueden utilizar diferentes escalas o rangos, sin embargo, es necesario tomar en cuenta que entre mayor sea dicha escala mayor debe ser la experiencia del diseñador sobre el tema. Luego se realiza un análisis de cada opción propuesta para así rellenar la matriz en cada criterio según el rango de ponderación dado anteriormente. Finalmente se calcula el punteo de cada opción propuesta, el cual se logra multiplicando el peso de cada criterio por el valor de la celda de dicha opción y sumando todos los valores. Por lo tanto, la solución con mayor punteo será la que representa a la más eficiente según los criterios planteados [37].

## **Proceso analítico jerárquico (AHP)**

Desde su desarrollo en la década de 1970 ha sido bastante utilizado para la toma de decisiones para casos de alta complejidad. Inicialmente se realiza una descomposición del problema en una jerarquía de criterios para realizar un análisis independiente de cada uno. Con la jerarquía lógica ya hecha se procede a realizar una evaluación sistemática de las alternativas propuestas a través de una comparación de los criterios. Esta comparación puede apoyarse utilizando datos concretos para cada alternativa. Este método transforma las comparaciones en valores numéricos, los cuales son procesados para un análisis más profundo. El peso de cada criterio permite la evaluación de los elementos en la jerarquía previamente definida. Ya hechas las comparaciones y establecidos los pesos de los criterios, se hace un cálculo de probabilidad para cada alternativa propuesta. Esta probabilidad determina el grado en que alguna alternativa pueda cumplir con el objetivo deseado. La principal diferencia de este método es que logra convertir datos (en su mayoría empíricos) en modelos matemáticos [38].

## **Kepner-Tregoe (KT)**

La secuencia de pasos a realizar en este método es aclarar el propósito, evaluar alternativas, evaluar los riesgos y beneficios de cada una y luego finalmente tomar la decisión. Es un método muy detallado y complejo que es aplicable en diferentes áreas, las cuales pueden ser más amplias que solo para la selección de ideas. Inicialmente se debe de tener claro qué es lo que se está decidiendo. Posteriormente se deben establecer y clasificar los objetivos en principales y secundarios. Luego se separan nuevamente los objetivos en dos categorías: necesario y deseado. A estos objetivos se les asignan factores de importancia en el rango de 1 al 10. Se evalúan las alternativas colocando una puntuación en función de lo deseado contra el objetivo principal. Se revisan las consecuencias negativas y finalmente se hace la toma de decisión [39].



## 6.5. Algoritmo de Madgwick

Los algoritmos de fusión de sensores permiten mejorar considerablemente la calidad de los datos obtenidos ya que aumenta su precisión y descarta gran parte del ruido adjunto a las lecturas. Para la mejora de medición inercial (como es el caso de la IMU), los algoritmos implican un costo computacional alto debido a que requieren cálculos matemáticos de álgebra lineal, álgebra de cuaterniones, procesos estocásticos y estadística inferencial.

El algoritmo de Madgwick, llamado así en honor a su desarrollador Sebastián Madgwick, representa la orientación a través de cuaterniones en lugar de matrices de cosenos directores. Esto es una ventaja significativa ya que al utilizar esta representación se evitan problemas de singularidades matriciales. Este algoritmo utiliza el método del gradiente descendente para realizar el cálculo de dirección del error en la medición del giroscopio [40]. Cabe mencionar dos aspectos importantes:

- Utiliza 109 operaciones por iteración para sensores IMU y 277 operaciones para sensores MARG (Conjunto IMU-magnetómetro).
- Presenta un buen rendimiento incluso cuando se presenta alguna frecuencia baja de muestreo.

El algoritmo puede dividirse en cuatro segmentos principales: integración de velocidades medidas con el giroscopio; cálculo de corrección de mediciones a través de los vectores campo magnético y del campo gravitacional; combinación de las estimaciones precedentes y la normalización de cuaterniones [40].

### 6.5.1. Integración de velocidades angulares

Los datos leídos del giroscopio se expresan de la siguiente manera:

$$S_\omega = [0, \omega_x, \omega_y, \omega_z] \quad (3)$$

La rotación del sistema de referencia inercial (Tierra) con respecto al sistema de referencia del sensor es posible obtenerla integrando la derivada de su cuaternión con respecto al tiempo. Inicialmente se calcula dicha derivada multiplicando el cuaternión por el vector de velocidades angulares expresada en (2).

$$\dot{q}_{\omega,k} = \frac{1}{2} \hat{q}_{k-1} \otimes S_\omega \quad (4)$$

En donde  $\hat{q}_{k-1}$  es el mejor cuaternión de pose de unidad obtenido en la última iteración. Posteriormente conociendo el intervalo de muestreo se realiza la integración del cuaternión.

$$\hat{q}_k = \hat{q}_{k-1} + \dot{q}_{\omega,k} \Delta t \quad (5)$$

### 6.5.2. Corrección de cuaterniones

El siguiente paso es calcular, a través de los vectores de campo magnético y del campo gravitacional, una aproximación del cuaternión actual.

$$f(\hat{q}_{k-1}, {}^E d, {}^B s) = \hat{q}_{k-1} \otimes {}^E d \otimes \hat{q}_{k-1} - {}^B s \quad (6)$$

Donde  $d$  representa el vector del campo magnético o campo gravitacional y  $f$  es la función objetivo.

$$\hat{q}_{\nabla, k-1} = \hat{q}_{k-1} - \mu \frac{\nabla f(\hat{q}_{k-1}, {}^E d, {}^B s)}{\|\nabla f(\hat{q}_{k-1}, {}^E d, {}^B s)\|} \quad (7)$$

Donde  $\mu$  hace referencia a la dirección del gradiente.

### 6.5.3. Fusión de estimaciones

El penúltimo paso es combinar las estimaciones realizadas en las subsecciones 6.4.1 y 6.4.2. Las expresiones matemáticas para realizar este proceso se describen a continuación.

$$\hat{q}_k = \gamma \hat{q}_{\nabla, k} + (1 - \gamma) \hat{q}_{\omega, k} \quad 0 \leq \gamma \leq 1 \quad (8)$$

Para el cálculo del coeficiente  $\gamma$  es necesario encontrar previamente una constante  $\beta$  la cual depende de la media del error de las mediciones realizadas con el giroscopio. Esta media se representa de la siguiente manera  $\omega_\beta$ .

$$\gamma_t = \frac{\beta}{\frac{\mu}{\Delta t} + \beta} \quad (9)$$

$$\beta = \sqrt{\frac{3}{4}} \omega_\beta \quad (10)$$

### 6.5.4. Normalización cuaternión

Finalmente el último paso es normalizar el cuaternión, lo cual se hace dividiéndolo sobre su norma.

$$\hat{q}_{\omega, k} = \frac{\hat{q}_{\omega, k}}{\|\hat{q}_{\omega, k}\|} \quad (11)$$

## 6.6. *Software*

### 6.6.1. Altium Designer

Corresponde a un paquete informático utilizado principalmente para el diseño de productos electrónicos uniendo el diseño PCB, gestión de datos y trabajo cooperativo. Altium Designer es un entorno de diseñador el cual ofrece una solución para el desarrollo de *hardware* y *software*. Ayuda al diseñador a tomar diseños para su posterior fabricación en un entorno unificado, al mismo tiempo que le permite explorar los dispositivos (microcontroladores, circuitos integrados, etc.) de una amplia gama de proveedores [41]. Una de las características más importantes de este programa es que ofrece la opción de visualizar e interactuar en 3D con el diseño de la placa en tiempo real. Esto quiere decir que mientras el diseñador trabaja desde un nivel de 2D puede observar los resultados y la distribución de los componentes en la placa de una manera sencilla y rápida. Es posible concluir que Altium Designer tiene capacidades de diseño, fabricación y edición de PCB sumamente poderosas, las cuales siguen en continúa innovación para lograr un desarrollo óptimo en el diseño a nivel de placa.

### 6.6.2. PlatformIO

PlatformIO es un entorno colaborativo profesional para el desarrollo de sistemas embebidos. Cuenta con un editor de código fuente multiplataforma con completaciones inteligentes basadas en definiciones de funciones, dependencias de librerías y tipos de variables declaradas. Al mismo tiempo tiene un flujo de trabajo de múltiples proyectos con fácil navegación por la base del código del proyecto. Presenta extensiones para 46 plataformas, 25 *frameworks*, 1007 dispositivos para desarrollo y 11148 librerías. Pone a la disposición del diseñador una gran cantidad de proyectos como ejemplos y un *debugger* integrado para la interpretación y detección de errores [42].

## 6.7. Ecosistema Robotat <sup>1</sup>

### 6.7.1. Wi-Fi

Wi-Fi es un acrónimo de *Wireless Fidelity*, la cual hace referencia a un tipo de tecnología de red inalámbrica la cual permite que diversos dispositivos informáticos puedan interconectarse. Una de sus ventajas más llamativas es que utiliza el mismo protocolo que internet (TCP/IP), por tal razón es fácil interconectar una red Wi-Fi con internet. Esta es la razón por la cual dicha tecnología esta teniendo un auge en el desarrollo e implementación en áreas como *Internet of Things* (IoT) [43].

---

<sup>1</sup>Ecosistema robótico definido en la sección de Justificación.

## Componentes de una red Wi-Fi

Una red Wi-Fi puede estar conformada por dos o más dispositivos. Para que sea posible la comunicación inalámbrica entre estos dispositivos deben de contar con lo que se conoce como adaptador de red. Este consiste en un equipo de radio con transmisor, antena y receptor el cual es el que permite establecer la comunicación. Estos adaptadores de red en su mayoría se encuentran integrados en el sistema. De igual manera, este tipo de redes pueden contar con equipos denominados como puntos de acceso (AP), los cuales son los encargados de gestionar la comunicación entre los diferentes equipos (también llamados terminales) que conforman la red. Los puntos de acceso funcionan de manera autónoma.

Al conjunto de equipos conformados por los puntos de acceso y terminales se les denomina como estación. Las estaciones de igual manera pueden comunicarse entre sí debido a que tienen la misma banda de frecuencias y el mismo conjunto de protocolos instalados. Estos protocolos pueden dividirse en dos grupos, cada uno con una función específica: uno son los protocolos Wi-Fi, los cuales deben garantizar la comunicación inalámbrica entre las estaciones que conforman la red, mientras que el otro protocolo es el encargado del transporte de información entre las terminales [43] Entre estos últimos se pueden encontrar los protocolos TCP/IP, SOCKETS y MQTT.

## Protocolo MQTT

MQTT (*Message Queue Telemetry Transport*) es un protocolo de transporte de mensajes Cliente/Servidor el cual se basa en las suscripciones y publicaciones a los denominados "tópicos". Cuando un mensaje es publicado este será recibido por todos los dispositivos enlazados a un tópico del protocolo. El protocolo MQTT funciona sobre otros protocolos con soporte bidireccional y sin pérdida de datos, como es el caso de TCP/IP [44]. Las principales características de este protocolo son las presentadas a continuación:

- Mecanismo de notificación por desconexiones inesperadas por parte de los dispositivos.
- El transporte de información es transparente y con un flujo de datos optimizados. Esto último reduce el tráfico en la red.
- Utiliza mensajes *broadcast* para la suscripción y publicación de datos con independencia de la publicación. Estos mensajes *broadcast* son los que se transmiten a todos los dispositivos de una red y que no necesita alguna acción de retroalimentación por parte del receptor.
- Pueden definirse tres calidades del servicio [44]:
  1. **“A lo sumo una vez”**. Esto consiste prácticamente en el protocolo TCP pero con pérdida de mensajes. Entre las aplicaciones se puede encontrar los casos donde se están leyendo datos por parte de sensores ya que, aunque haya cierta pérdida en algunas lecturas, eventualmente llegará otra con información más actualizada.
  2. **“Exactamente una vez”**. Se garantiza que los mensajes llegan exactamente una vez. Se utiliza con frecuencia en aplicaciones donde la duplicidad o la pérdida de mensajes es crítica.

3. “**Al menos una**”. Se garantiza la llegada de los mensajes pero tomando en cuenta algunos casos de duplicidad.

Debido a las características mencionadas anteriormente, este protocolo de transporte de información es ideal para las aplicaciones IoT. En estas aplicaciones el envío de información se da en pequeñas cantidades, lo cual implica que no hay necesidad de un gran ancho de banda.

### 6.7.2. Medición de pose

#### Captura de movimiento

La captura de movimiento es el proceso mediante el cual se registra, como su nombre lo dice, el movimiento de persona u objetos. Esto implica la medición de posición y orientación de estos elementos en el espacio físico. Actualmente esta tecnología es utilizada en diversos campos, como lo son la robótica, animación, gráficas de computadora, entre otros. La captura de movimiento puede tener diferentes enfoques [45]:

- **Pasivo óptico.** Es el método más práctico y utilizado en la industria. Los objetos utilizan marcadores retro reflectores, los cuales son rastreados por un conjunto de cámaras infrarrojas colocadas al rededor del espacio de análisis.
- **Activo óptico.** En este método los marcadores LED infrarrojos son rastreados nuevamente por un conjunto de cámaras infrarrojas. A diferencia del método de pasivo óptico, es necesario una fuente de alimentación externa para alimentar a los marcadores activos.
- **Video.** Se utiliza un programa para rastrear el movimiento del objeto en cada fotograma o *frame*. La precisión de los resultados puede variar según el algoritmo de seguimiento utilizado. Es por esto último que no es muy recomendable usar esta técnica para casos donde se desean resultados certeros.
- **Inercial.** A diferencia de las técnicas mencionadas anteriormente, esta no utiliza cámaras. En su lugar utiliza sensores de medición inercial colocados sobre el objeto.

Es de suma importancia seleccionar un correcto sistema de captura de movimiento según la aplicación deseada. Los sistemas óptico-pasivo son ampliamente utilizados en el área de robótica para el control y validación de la visión de computadora en un laboratorio con entorno controlado, especialmente para aplicaciones de robótica móvil [45].

## OptiTrack

OptiTrack es uno de los proveedores de sistemas de captura de movimiento más grande a nivel internacional. Sus sistemas ofrecen rastreo óptico de alto rendimiento a precios accesibles. Los productos OptiTrack conforman el *software* para realizar la captura de movimiento y las cámaras de seguimiento de alta velocidad [46]. A pesar de que se encuentran disponibles una gran variedad de cámaras de seguimiento en el catálogo, cabe mencionar que un sólo sistema de captura de movimiento OptiTrack con cámaras estándar tiene la capacidad de rastrear la pose de seis grados de libertad de uno o más objetos que se encuentren en el espacio de trabajo. Estos objetos deben de definirse como cuerpos rígidos. Cada cuerpo rígido es un conjunto de marcadores retro reflectores colocados en una configuración específica. El tamaño de estos reflectores son seleccionados según el tamaño y las limitaciones que presente la geometría del cuerpo rígido.

El *firmware* para la plataforma fue desarrollado a través de librerías para el control y manejo de los motores BLDC colocados en las hélices y para la extracción de datos de la MPU-9250, es decir, giroscopio, acelerómetro y magnetómetro. Estas librerías fueron programadas en lenguaje de programación C utilizando el SDK de Espressif e implementadas a través del microcontrolador ESP32. En estas librerías se tomo como base la programación orientada a objetos a través de estructuras. Se utilizó este tipo de programación principalmente por su ventaja de reusabilidad, es decir, las clases pueden ser utilizadas en diferentes secciones de código lo cual lo hace más eficiente y fácil de interpretar. También se logra desarrollar un código más generalizado, lo cual permite implementar de manera sencilla otros motores BLDC o controladores ESC en caso fuera necesario más adelante en la línea de investigación.

La necesidad del desarrollo de estas librerías se encuentra estrechamente relacionada con el controlador para la estabilización de la plataforma diseñado por Carlos Alonzo. El lazo de control se cierra a través de la fusión de datos de la MPU-9250 y los obtenidos a través del sistema de captura de movimiento OptiTrack para obtener la pose de la plataforma. Por otra parte, la implementación física del controlador se logra por medio de los motores BLDC modificando la velocidad de los mismos a través de los ESC.

## 7.1. Motores BLDC

La plataforma cuenta con cuatro motores BLDC y, por ende, cuatro ESC para el control de los mismos. Esta librería crea un objeto por cada conjunto motor-ESC, la cual presenta los atributos presentados a continuación.

Atributo	Descripción
<i>KV2RPM</i>	Variable tipo <i>int</i> . Especifica la relación RPM/V del motor que se este utilizando. Este dato se proporciona en la <i>datasheet</i> correspondiente.
<i>V_battery</i>	Variable tipo <i>float</i> . Voltaje nominal de la batería utilizada para la alimentación de los ESC.
<i>pulse_width_max</i>	Variable tipo <i>float</i> . Ancho de pulso máximo permitido por el ESC en ms. Este dato se proporciona en la <i>datasheet</i> correspondiente.
<i>pulse_width_min</i>	Variable tipo <i>float</i> . Ancho de pulso mínimo permitido por el ESC en ms. Este dato se proporciona en la <i>datasheet</i> correspondiente.

Cuadro 3: Atributos del objeto *bldc\_esc\_t*.

Estos atributos son utilizados principalmente para el cálculo de la relación del ancho de pulso necesario según la velocidad en RPM que requiere el controlador. Realizar esta programación como un objeto permite generalizar el código de manera que, al momento de cambiar los motores a utilizar, su implementación sea relativamente sencilla únicamente ingresando los datos descritos en el Cuadro 3.

Esta librería cuenta con seis funciones, las cuales cumplen con diferentes propósitos como la inicialización de los objetos, puertos para las señales PWM o de los ESC, entre otros. En el Cuadro 4 se hace el detalle de cada unas de estas funciones.

Función	Parámetros	Descripción
<code>bldc_esc_init</code>	Puntero a <i>bldc_esc_t</i> Relación RPM/V Voltaje de batería Ancho de pulso máximo ESC Ancho de pulso mínimo ESC	Inicializa a cada objeto con los parámetros ingresados por el usuario.
<code>config_gpio</code>	GPIO	Configura como salida los GPIO's colocados en el parámetro de la función. Se deshabilita las interrupciones, el <i>pull-down mode</i> y el <i>pull-up mode</i> .

*Continúa en la siguiente página*



Cuadro 4 – Continuación de la página previa

Función	Parámetros	Descripción
PWM_Init	GPIO LEDC <i>channel</i> LEDC <i>timer</i>	Configuración del <i>timer</i> y del canal deseado para la generación de la señal PWM a través del periférico LED <i>control</i> con el que cuenta el ESP32. El parámetro GPIO define el puerto de salida para la señal. Se define una resolución de 12 bits, frecuencia de señal de 50 Hz y el <i>timer</i> en <i>High Speed Mode</i> .
PWM_output	Puntero a <i>bldc_esc_t</i> LEDC <i>channel</i> Salida controlador	Genera la señal de PWM en el canal especificado. Se plantea una relación lineal para obtener el ancho de pulso deseado según la velocidad requerida por el controlador.
inicialización_ESC	LEDC <i>channel</i>	Cada ESC al momento de energizarlos es necesario enviarle una señal PWM con el ancho mínimo, el cual es de 1 ms. Una vez hecho esto la función realiza un <i>delay</i> de 2 segundos mientras el ESC termina de configurarse.
inicialización_ESC_- drone	void <sup>1</sup>	Realiza el mismo proceso que la función <i>inicialización_ESC</i> pero para los cuatro motores de forma simultánea.

Cuadro 4: Funciones del objeto *bldc\_esc\_t*.

En la función PWM\_Init se especificó una frecuencia de 50 Hz para la señal de PWM. Esto implica que el período del pulso es igual a  $1/50 = 0.02$  segundos = 20 ms. Según las especificaciones del ESC específico utilizado para la plataforma, el ancho de pulso de la señal debe de estar en el rango de 1 - 2 ms. La resolución es de 12 bits, lo cual es equivalente a 4095 en decimal ( $2^n - 1$ ), y considerando que el ancho de la señal es de 20 ms, es necesario actuar como máximo a una décima parte del dato anterior. Esto último dicho en otras palabras indica que la señal de PWM expresada en decimal debe de ser como máximo  $4095/10 = 409.5$ , pero se aproxima a 410 ya que el dato debe de ser un número entero. Lo equivalente en decimal para el caso del ancho mínimo de 1 ms es igual a  $409.5/2 = 204.75$ , lo cual nuevamente se aproxima al número entero más cercano el cual es 205.

<sup>1</sup>Indica que la función no necesita parámetros.

### 7.1.1. Relación RPM-ancho de pulso PWM

A la salida del controlador diseñado por Carlos Alonzo se obtiene un valor de velocidad expresado en RPM. Tomando esto en cuenta, se utilizó una regresión lineal para el planteamiento de la función `PWM_output`, en la cual se relaciona la velocidad en RPM con el ancho de pulso de la señal PWM. El ancho de pulso debe expresarse en forma decimal ya que de esta forma es requerido por la función `ledc_set_duty` en la librería de `LED control`.

Para el planteamiento de la relación lineal se partió del conocimiento de dos puntos: a 0 RPM se quiere el ancho de pulso mínimo permitido por el ESC, mientras que para la velocidad máxima se requiere el ancho de pulso máximo permitido. La velocidad máxima depende del voltaje de la batería y de la relación RPM/V que presente el motor utilizado. Generalmente se tiene acceso al ancho mínimo y máximo del ESC expresado en milisegundos, por lo tanto, por cuestiones de optimización se plantea una relación previa la cual se muestra a continuación. El planteamiento de ecuaciones se hará utilizando los atributos de los objetos para esta librería ya que de esta manera fueron implementadas en código.

$$PW_1 = \frac{\text{pulse\_width\_min}}{20}4095, \quad (12)$$

$$PW_2 = \frac{\text{pulse\_width\_max}}{20}4095. \quad (13)$$

Estas dos ecuaciones realizan la conversión a valores decimales del ancho de pulso máximo y mínimo. Dicha conversión es la misma que la realizada al final de la sección anterior, con la diferencia que en este caso se plantea de una forma general para el uso de ESC con anchos de pulso diferentes a los utilizados para la implementación de la plataforma. Con esto ya se plantean los dos puntos mencionados anteriormente para obtener la ecuación de la recta.

$$P_1(0, PW_1), \quad (14)$$

$$P_2(V\_battery * KV2RPM, PW_2). \quad (15)$$

Luego de cierta manipulación algebraica, se obtuvo el siguiente resultado:

$$PWM = \frac{4095}{20 \times V\_battery \times KV2RPM}(\text{pulse\_width\_max} - \text{pulse\_width\_min})x + \frac{4095 \times \text{pulse\_width\_min}}{20} \quad (16)$$

### 7.1.2. Modelo experimental

Como se mencionó en la introducción del capítulo, la importancia de esta librería recae en que es la conexión entre el controlador (*software*) y la implementación física del mismo (*hardware*). Por tal razón, para la validación del código se procedió a realizar dos pruebas diferentes, las cuales se describen a continuación.

- **Medición del ancho de pulso.** Para que el ESC logre energizar y controlar de manera correcta a un motor BLDC, es necesario que el ancho de pulso de la señal PWM generada se encuentra en el rango requerido por el fabricante. Esto es posible verificarlo utilizando el equipo de laboratorio disponible en el campus de la universidad.
- **Velocidad real durante implementación.** Verificar la relación RPM-ancho de pulso PWM descrita en la subsección anterior. Esto ya que parte del éxito de la implementación del controlador recae en que las velocidades solicitadas sean coherentes con las ejercidas por los motores. Para ello se utilizó un tacómetro láser disponible en la universidad para un rango de once velocidades diferentes.

### 7.1.3. Resultados

#### Ancho de pulso

Previo a conectar los motores con los ESC para realizar las pruebas del funcionamiento del mismo, se verificó que el ancho de pulso generado por la función `PWM_output` fuera el correcto. Para ello se utilizó un osciloscopio para documentar estos datos. La manera más eficiente fue observar el ancho de pulso para las velocidades 0 RPM y 12000 RPM, las cuales serán la mínima y máxima respectivamente. Se considera a 12000 RPM como velocidad máxima debido a que en las pruebas posteriores se alimentaría a la plataforma con 12V directamente de la fuente.

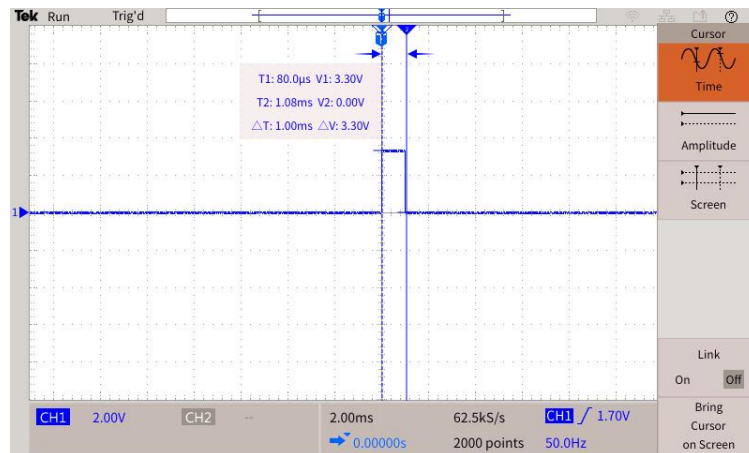


Figura 8: Ancho de pulso para velocidad de 0 RPM.

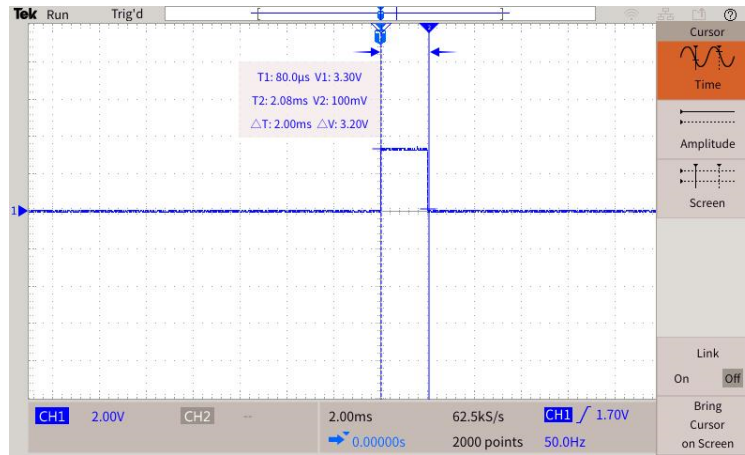


Figura 9: Ancho de pulso para velocidad de 12000 RPM.

Tal y como se observa en las Figuras 8 y 9, las mediciones coinciden con el ancho de pulso mínimo y máximo permitido por el ESC. Con esto es posible validar de forma parcial el correcto funcionamiento de la librería desarrollada.

### Velocidad real durante implementación

La primera medición se realizó a una velocidad de 2500 RPM para luego realizar incrementos de 200 RPM hasta finalmente llegar a 4500 RPM. El tacómetro utilizado fue el FLUKE 810 VIBRATION TESTER disponible en el campus. Según las recomendaciones del instructivo del dispositivo, para realizar mediciones precisas es necesario colocar la cinta reflectiva sobre la cual se apuntará el láser en el eje del motor. En este caso para se colocó sobre la copa de la hélice tal y como se muestra en la Figura 10.



Figura 10: Colocación de cinta reflectiva para la medición con tacómetro.

El tacómetro marca automáticamente la velocidad una vez obtenga una cantidad suficiente de muestras. Para obtener resultados más confiables, en cada velocidad se realizó una medición en el voltaje de la batería para cambiar este parámetro en el código fuente (este afecta directamente la relación RPM-ancho de pulso). Los resultados para un motor se muestran en el Cuadro 5.

Motor 1	
Velocidad código (RPM)	Velocidad real (RPM)
2500	2791.00
2700	3064.25
2900	3114.33
3100	3337.66
3300	3478.33
3500	3669.50
3700	3703.25
3900	3876.66
4100	3984.25
4300	4066.50
4500	4133.50

Cuadro 5: Comparación de velocidad en código contra la velocidad real medida para motor 1.

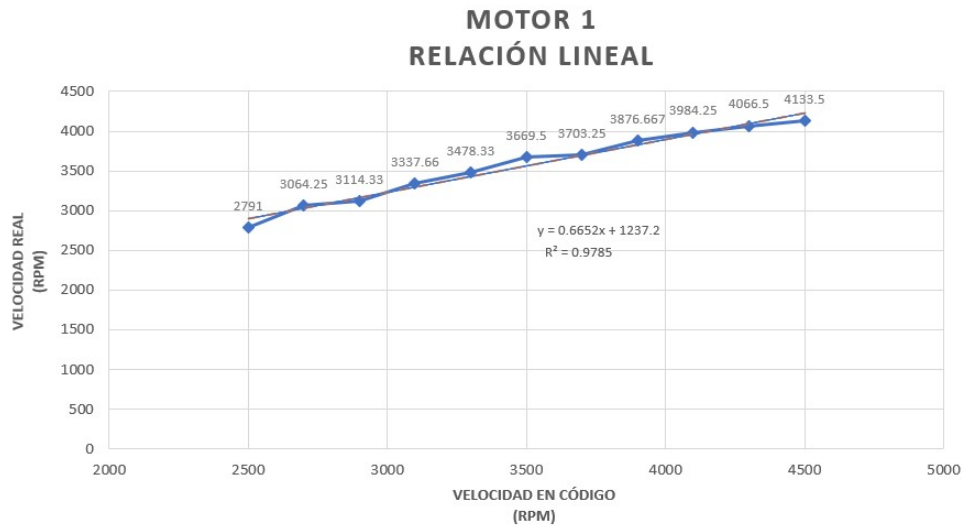


Figura 11: Regresión lineal de velocidades obtenidas con tacómetro para motor 1.

Tal y como se observa en la Figura 11, con los datos medidos se obtiene un coeficiente de determinación  $R^2$  de 0.9785, lo cual es un valor considerablemente alto para concluir que el modelo se ajusta correctamente a los datos.

Durante la realización de diversas de pruebas se logró percibir que algunos motores presentaban mayor fuerza a pesar de utilizar la misma relación y velocidad en código. Por tal razón, se procedió a realizar la misma prueba con el tacómetro para uno de estos motores con mayor empuje. Los resultados se muestran en el Cuadro 6.

Motor 2	
Velocidad código (RPM)	Velocidad real (RPM)
2500	3023.50
2700	3166.00
2900	3242.67
3100	3404.00
3300	3524.00
3500	3672.10
3700	3851.25
3900	3891.00
4100	4086.25
4300	4133.67
4500	4154.00

Cuadro 6: Comparación de velocidad en código contra la velocidad real medida para motor 2.

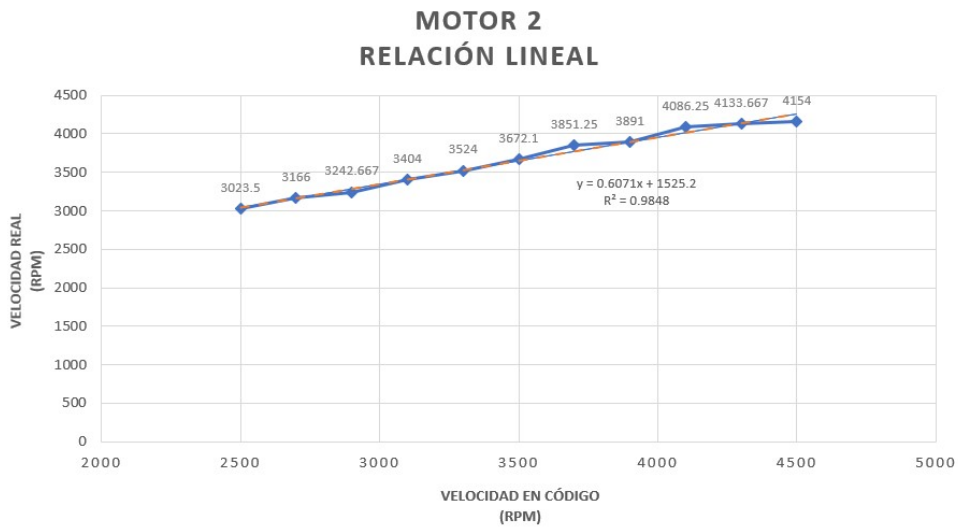


Figura 12: Regresión lineal de velocidades obtenidas con tacómetro para motor 2.

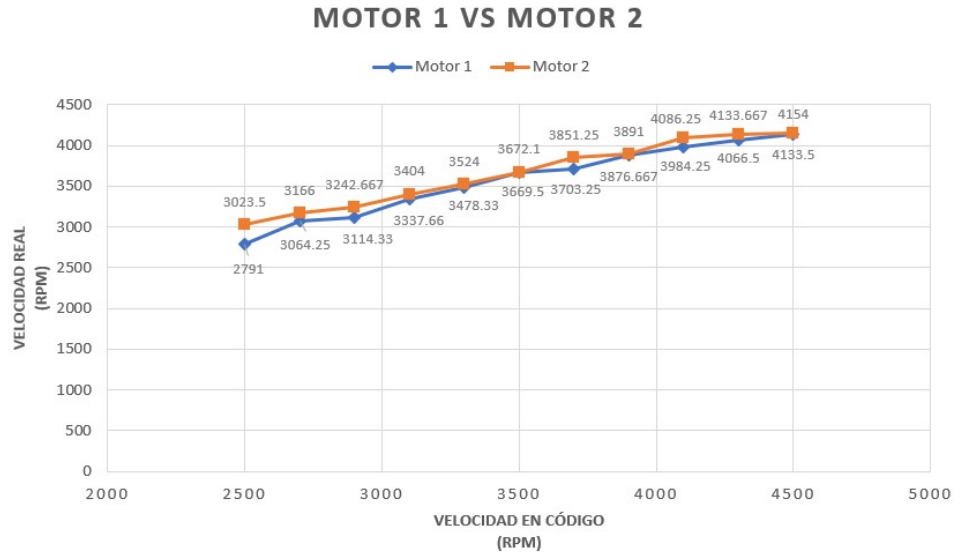


Figura 13: Comparación de velocidades medidas para dos motores diferentes.

A diferencia del motor 1, para este segundo caso se tiene un coeficiente de determinación  $R^2$  de 0.9848 tal y como se observa en la Figura 12. Esto indica que el modelo se ajusta aún mejor a los datos, sin embargo también refleja cierta descompensación entre los motores. Dicha descompensación se muestra en la Figura 13, en donde se observa que para todas las velocidades medidas el motor 2 presenta un valor más alto o, en su defecto ligeramente parecido a los presentados por el motor 1. Esto puede deberse a que son motores de baja calidad, y por ende, puede tener efectos bastante negativos al momento de la implementación del controlador.

## 7.2. MPU-9250

El objetivo principal de utilizar una unidad de medición inercial fue para obtener la orientación de la plataforma a través de las rotaciones en los ejes  $x$ ,  $y$ ,  $z$ . Estas rotaciones son representadas por los ángulos *roll*, *pitch* y *yaw* respectivamente. Dichos ángulos se obtienen combinando los datos del giroscopio, acelerómetro y magnetómetro. Para esta ocasión, la combinación de estos datos se realizó utilizando el filtro de Madgwick. Sin embargo, todo este proceso involucró el trabajo en conjunto de varias librerías las cuales se detallan en las siguientes subsecciones. Estas librerías fueron tomadas del siguiente repositorio: <https://github.com/psiphi75/esp-mpu9250>, por lo tanto, más que desarrollo de código esta parte consistió en validación de cada función con la ayuda del mapa de registro de la MPU-9250 y el manual de usuario del ESP32.

### 7.2.1. Librería I2C

La necesidad de esta librería surge para lograr la comunicación entre el microcontrolador ESP32, el cual hace de dispositivo maestro, y la MPU-9250, la cual es el esclavo en el protocolo de comunicación. Espressif Systems proporciona una librería para la implementación de este protocolo en el ESP32, por lo tanto, se toma como base estas funciones para desarrollar unas nuevas que se ajusten a las necesidades en este caso específico.

Función	Parámetros	Descripción
<code>i2c_master_init</code>	Número de puerto I2C GPIO para señal SDA GPIO para señal SCL	Realiza las configuraciones para establecer al dispositivo como maestro para el protocolo de comunicación. En la librería se define un macro con una velocidad de reloj de 40000 Hz, la cual es permitida por la MPU-9250.
<code>i2c_write_bytes</code>	Número de puerto I2C Dirección de periférico Dirección de registro Datos a escribir Cantidad de <i>bytes</i> a escribir	Permite la escritura de varios <i>bytes</i> de datos a partir de un registro específico del periférico esclavo.
<code>i2c_write_byte</code>	Número de puerto I2C Dirección de periférico Dirección de registro Datos a escribir	Realiza la escritura de un solo <i>byte</i> de datos en un registro específico del periférico esclavo.
<code>i2c_read_bytes</code>	Número de puerto I2C Dirección de periférico Dirección de registro Almacenamiento de datos leídos Cantidad de <i>bytes</i> a leer	Realiza la lectura de varios <i>bytes</i> de datos provenientes del periférico esclavo a partir del registro especificado por el usuario.

*Continúa en la siguiente página*



Cuadro 7 – Continuación de la página previa

Función	Parámetros	Descripción
<code>i2c_read_byte</code>	Número de puerto I2C Dirección de periférico Dirección de registro Almacenamiento de datos leídos	Lectura de un <i>byte</i> de datos provenientes del registro especificado del periférico esclavo.
<code>i2c_write_bits</code>	Número de puerto I2C Dirección de periférico Dirección de registro Bit a comenzar escritura Longitud de bits a escribir Valores a escribir	Escritura sobre varios bits de un registro específico del periférico esclavo. La escritura comienza a partir del número de bit del registro especificado por el usuario.
<code>i2c_write_bit</code>	Número de puerto I2C Dirección de periférico Dirección de registro Bit a comenzar escritura Valores a escribir	Escritura sobre un solo bit de un registro específico del periférico esclavo.
<code>i2c_read_bit</code>	Número de puerto I2C Dirección de periférico Dirección de registro Bit a leer Almacenamiento de dato leído	Permite la lectura de un bit en el registro especificado del periférico esclavo.

Cuadro 7: Funciones comunicación I2C.

### 7.2.2. Librería MPU-9250

Una vez establecida la librería para la comunicación, se procede a la escritura y lectura de los registros en la MPU-9250. Para la realización de estas funciones se apoyó en el mapa de registros del periférico que comparte el fabricante. La MPU-9250 tiene la dirección I2C 0x68 según su hoja de datos. En esta librería se establecieron dos objetos cuyos atributos se presentan a continuación.

Atributo	Descripción
<i>x</i>	Variable tipo <i>float</i> . Representa algún dato que describe una medición sobre el eje <i>x</i> .
<i>y</i>	Variable tipo <i>float</i> . Representa algún dato que describe una medición sobre el eje <i>y</i> .
<i>z</i>	Variable tipo <i>float</i> . Representa algún dato que describe una medición sobre el eje <i>z</i> .

Cuadro 8: Atributos del objeto *vector\_t*.

Atributo	Descripción
<i>mag_offset</i>	Variable tipo <i>vector_t</i> . Representa la medición de ruido del magnetómetro por eje, la cual es utilizada como una cantidad de sesgo.
<i>mag_scale</i>	Variable tipo <i>vector_t</i> . Representa el factor de escala para cada eje para eliminar la forma elíptica del rango de lectura. La forma ideal debería ser de tipo circular.
<i>gyro_bias_offset</i>	Variable tipo <i>vector_t</i> . Representa la medición de ruido del giroscopio por eje, la cual es utilizada como una cantidad de sesgo.
<i>accel_offset</i>	Variable tipo <i>vector_t</i> . Representa la medición de ruido del acelerómetro por eje, la cual es utilizada como una cantidad de sesgo.
<i>accel_scale_lo</i>	Variable tipo <i>vector_t</i> . Representa el rango mínimo de lectura del acelerómetro para cada eje.
<i>accel_scale_hi</i>	Variable tipo <i>vector_t</i> . Representa el rango máximo de lectura del acelerómetro para cada eje.

Cuadro 9: Atributos del objeto *calibration\_t*.

Los atributos descritos en los Cuadros 8 y 9 son utilizados para representar las mediciones del giroscopio, acelerómetro y magnetómetro en cada eje, tomando en cuenta cierto error el cual se intenta disminuir a través del proceso de calibración. La librería y la descripción de dicho proceso se detalla en la siguiente subsección del capítulo. Esta librería cuenta con una cantidad considerable de funciones, por lo tanto, se dividirán según el objetivo por el cual se desarrolló cada una de ellas.

## Inicialización MPU-9250

Función	Parámetros	Descripción
<code>i2c_master_init</code>	Puntero a <code>calibration_t</code>	Establece diversas configuraciones necesarias para utilizar el periférico: define el reloj, el rango de lectura del giroscopio y acelerómetro, habilitar el <i>sleep mode</i> y el magnetómetro. Realiza el proceso de impresión de todas las configuraciones realizadas. La función para el magnetómetro se explicará más adelante en el capítulo a pesar de que se utilice en esta sección. Establece el puerto I2C número cero y los GPIO 22 y 21 del ESP32 para las señales SCL y SDA respectivamente para la comunicación. Adicionalmente hace un reinicio a los registros a la configuración por defecto del dispositivo.
<code>set_clock_source</code>	Código de reloj deseado	Establece la fuente de reloj según el parámetro ingresado por el usuario. Este parámetro puede ser un valor de 0 a 7, donde cada uno representa una fuente de reloj especificada en el mapa de registros del dispositivo.
<code>set_full_scale_accel_range</code>	Código de escala deseada	Establece el rango de lectura para el acelerómetro según el parámetro ingresado por el usuario. Este parámetro puede ser un valor de 0 a 3 (o su equivalente en binario), donde cada uno representa las siguientes escalas: $\pm 2g$ (00), $\pm 4g$ (01), $\pm 8g$ (10) y $\pm 16g$ (11).

*Continúa en la siguiente página*

Cuadro 10 – Continuación de la página previa

Función	Parámetros	Descripción
<code>set_full_- scale_gyro_- range</code>	Código de escala deseada	Establece el rango de lectura para el giroscopio según el parámetro ingresado por el usuario. Este parámetro puede ser un valor de 0 a 3 (o su equivalente en binario), donde cada uno representa las siguientes escalas: $\pm 250^\circ/\text{sec}$ (00), $\pm 500^\circ/\text{sec}$ (01), $\pm 1000^\circ/\text{sec}$ (10) y $\pm 2000^\circ/\text{sec}$ (11).
<code>set_i2c_- master_mode</code>	Variable <i>bool</i>	Escritura sobre el bit 5 del registro <code>USER_CTRL</code> para la habilitación del modo maestro de la MPU-9250 en el segundo protocolo de comunicación I2C mostrado en la Figura 14.
<code>set_bypass_- mux_enabled</code>	Variable <i>bool</i>	Escritura sobre el bit 1 del registro <code>INT_PIN_CFG</code> para la habilitación del interfaz <i>bypass mux</i> para lograr comunicación con el magnetómetro.
<code>set_sleep_- enabled</code>	Variable <i>bool</i>	Habilita o no el <i>sleep mode</i> del dispositivo según el parámetro <i>booleano</i> ingresado por el usuario.

Cuadro 10: Funciones inicialización MPU-9250

## Lectura valores de giroscopio y acelerómetro

Las funciones desarrolladas y explicadas a continuación toman en cuenta que la resolución de lectura de estos instrumentos es de 16 bits por cada eje. Esto implica 6 *bytes* de datos para el giroscopio y el equivalente para el acelerómetro.

Función	Parámetros	Descripción
<code>scale_- accel</code>	Lectura de acelerómetro Atributo <i>accel_offset</i> Atributo <i>accel_scale_lo</i> Atributo <i>accel_scale_hi</i>	Esta función recibe el valor del acelerómetro y los datos obtenidos de la calibración del acelerómetro por cada eje. Devuelve la medición del acelerómetro pero con el ajuste necesario para la disminución de error.

*Continúa en la siguiente página*

Cuadro 11 – Continuación de la página previa

Función	Parámetros	Descripción
<code>accel</code>	6 <i>bytes</i> Puntero a <i>vector_t</i>	Se recibe el arreglo de 6 <i>bytes</i> , el cual contiene toda la información leída por el acelerómetro y se descomponen los datos correspondientes para cada eje. Una vez hecho esto, se utiliza la función <code>scale_accel</code> , nuevamente por eje, con los atributos respectivos del objeto <code>calibration_t</code> con el cual se inicializó la MPU-9250. Finalmente los resultados se almacenan en cada uno de los atributos del objeto <code>vector_t</code> ingresado por el usuario.
<code>get_accel</code>	Puntero a <i>vector_t</i>	Realiza la lectura de los registros para los datos del acelerómetro con la función <code>i2c_read_bytes</code> explicada en el Cuadro 7. Luego utiliza la función <code>accel</code> para finalmente almacenar el resultado final en los atributos del objeto <code>vector_t</code> ingresado por el usuario como parámetro.
<code>gyro</code>	6 <i>bytes</i> Puntero a <i>vector_t</i>	Se recibe un arreglo de 6 <i>bytes</i> , el cual contiene toda la información leída por el giroscopio y se descomponen los datos correspondientes para cada eje. Luego a estos datos se les suma el atributo <code>gyro_bias_offset</code> por cada eje del objeto <code>calibration_t</code> con el cual se inicializó la MPU-9250. Los resultados son almacenados en cada uno de los atributos del objeto <code>vector_t</code> ingresado por el usuario.

*Continúa en la siguiente página*

Cuadro 11 – Continuación de la página previa

Función	Parámetros	Descripción
<code>get_gyro</code>	Puntero a <code>vector_t</code>	Realiza la lectura de los registros para los datos del giroscopio con la función <code>i2c_read_bytes</code> explicada en el Cuadro 7. Luego utiliza la función <code>gyro</code> para finalmente almacenar el resultado final en los atributos del objeto <code>vector_t</code> ingresado por el usuario como parámetro.
<code>get_- accel_gyro</code>	Puntero a <code>vector_t</code> giroscopio Puntero a <code>vector_t</code> acelerómetro	Es la combinación de las funciones <code>get_gyro</code> y <code>get_accel</code> . Realiza una lectura de 14 <code>bytes</code> partiendo del registro <code>ACCEL_-XOUT_H</code> con dirección <code>0x3B</code> . Los 2 <code>bytes</code> restantes de estos 14 pertenecen a la medición de temperatura, sin embargo en este caso se omiten ya que no son de interés para la aplicación deseada.

Cuadro 11: Funciones para lectura de datos del giroscopio y acelerómetro.

## Impresión de configuraciones

La necesidad de estas funciones surge ya que se tiene una gran variedad de configuraciones para el giroscopio, acelerómetro y la propia MPU-9250. Por lo tanto, imprimir estos datos en consola al momento de encender el dispositivo le otorga al usuario toda la información para verificar si todo se encuentra de forma correcta antes de empezar la lectura de datos.

Función	Parámetros	Descripción
<code>get_device_id</code>	Puntero a variable <code>uint8_t</code>	Lectura del registro <code>WHO_AM_I</code> con dirección <code>0x75</code> . Es utilizada principalmente para verificar que la comunicación I2C entre el ESP32 y la MPU-9250 se encuentre correcta.
<code>get_sleep_- enabled</code>	Puntero a variable <code>bool</code>	Lectura del bit 6 del registro <code>PWR_MGMT_1</code> con dirección <code>0x6B</code> . Permite conocer si se encuentra habilitado el <code>sleep mode</code> .

*Continúa en la siguiente página*

Cuadro 12 – Continuación de la página previa

Función	Parámetros	Descripción
<code>get_i2c_- master_mode</code>	Puntero a variable <i>bool</i>	Permite conocer si los pines ES_DA y ES_SCL se encuentran aislados o controlados lógicamente por los pines SDA y SCL.
<code>get_clock_- source</code>	Puntero a variable <i>uint8_t</i>	Lectura sobre el registro PWR_-MGMT_1 con dirección 0x6B, específicamente los bits 0 al 2 para conocer el código de la fuente de reloj seleccionada.
<code>get_gyro_- settings</code>	3 <i>bytes</i>	Lectura sobre el registro PWR_-MGMT_2 sobre los bits del 0 al 2 para identificar si el giroscopio se encuentra habilitado para cada uno de los 3 ejes.
<code>get_full_- scale_gryo_- range</code>	Puntero a variable <i>uint8_t</i>	Permite conocer el rango de lectura del giroscopio seleccionada durante la inicialización de la MPU-9250 con la función <i>i2c_master_init</i> .
<code>get_accel_- settings</code>	3 <i>bytes</i>	Lectura sobre el registro PWR_-MGMT_2 sobre los bits del 3 al 5 para identificar si el acelerómetro se encuentra habilitado para cada uno de los 3 ejes.
<code>get_full_- scale_accel_- range</code>	Puntero a variable <i>uint8_t</i>	Permite conocer el rango de lectura del acelerómetro seleccionada durante la inicialización de la MPU-9250 con la función <i>i2c_master_init</i> .
<code>get_bypass_- mux_enabled</code>	Puntero a variable <i>bool</i>	Permite identificar si la interfaz <i>bypass mux</i> , la cual se muestra en la Figura 15, se encuentra habilitada para establecer comunicación con el magnetómetro.

*Continúa en la siguiente página*

Cuadro 12 – Continuación de la página previa

Función	Parámetros	Descripción
<code>mpu9250_print_settings</code>	<code>void</code>	Imprime los siguientes datos: número de puerto I2C, dirección de la MPU-9250, lectura del registro WHO_AM_I, el reloj y la habilitación del <i>bypass mux</i> , <i>sleep mode</i> , acelerómetro, giroscopio y del I2C <i>master mode</i> .
<code>print_accel_settings</code>	<code>void</code>	Imprime los siguientes datos: rango de lectura del acelerómetro y los atributos del objeto <i>calibration_t</i> correspondientes para este instrumento.
<code>print_gyro_settings</code>	<code>void</code>	Imprime en consola los siguientes datos: rango de lectura del giroscopio y los atributos del objeto <i>calibration_t</i> correspondientes para este instrumento, el cual para este caso únicamente es <i>gyro_bias_offset</i> .

Cuadro 12: Funciones para impresión en consola de configuraciones realizadas sobre la MPU-9250.

### 7.2.3. Librería magnetómetro

La MPU-9250 trae incorporada un AK8963, el cual es un circuito integrado de brújula electrónica de 3 ejes. Incorpora sensores magnéticos para la detección de magnetismo terrestre en dichos ejes. Presenta una resolución de salida de datos de 14 o 16 bits. Este dispositivo es considerado como un tercer sensor, con el cual se logra comunicación con la MPU-9250 por medio del protocolo I2C. Esto quiere decir que para obtener la medición de todos los sensores disponibles es necesaria una doble comunicación I2C, tal y como se presenta en la Figura 14.

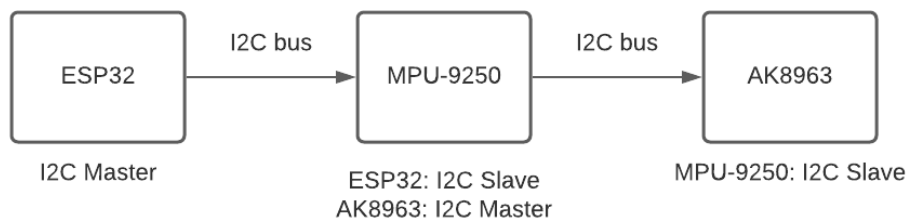


Figura 14: Comunicación I2C entre los 3 periféricos.



Por lo tanto, el ESP32 es el periférico maestro y la MPU-9250 el esclavo en el primer bus de comunicación. Al mismo tiempo, la MPU-9250 hace de maestro y el tercer sensor de esclavo, en este caso el AK8963, para el segundo bus de comunicación. Para lograr comunicación directa entre el ESP32 y el AK8963, la MPU-9250 cuenta con el modo de operación *Pass-Through Mode* para el bus I2C. En este modo se conectan los buses I2C primario y auxiliar juntos, lo que permite al ESP32 se comunique directamente con cualquier sensor externo colocado en el bus auxiliar. Esta interconexión entre dispositivos es posible ya que la MPU-9250 cuenta un multiplexor de derivación de interfaz (en inglés *interface bypass multiplexer*), el cual realiza la conexión entre buses de datos como se explicó anteriormente. Esto último puede observarse en la Figura 15.

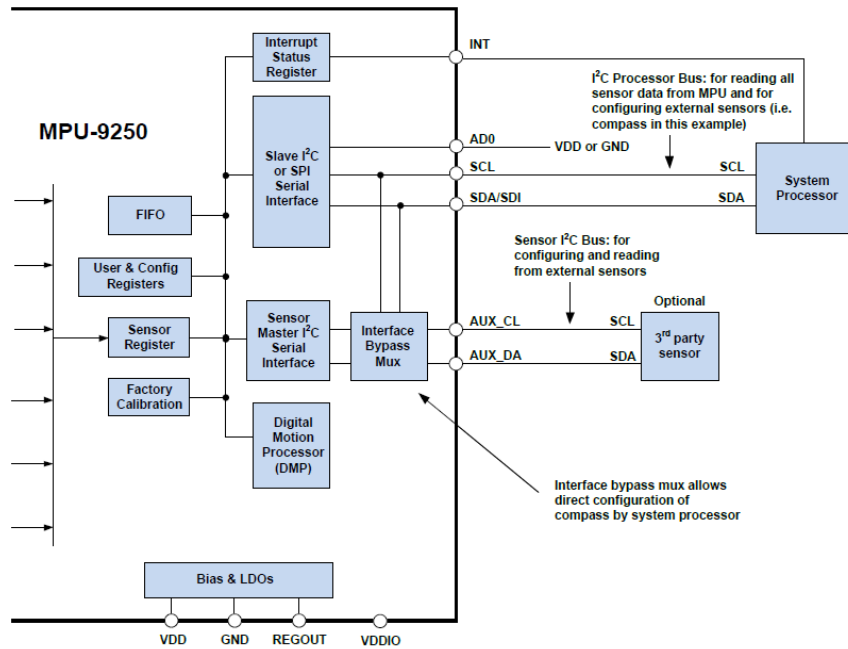


Figura 15: *Interface Bypass Mux* para la interconexión de los buses I2C.

Cabe mencionar que la MPU-9250 presenta capacidades limitadas como periférico maestro en el protocolo de comunicación, por ende depende directamente del procesador del sistema (ESP32 en este caso) para realizar la configuración inicial de cualquier sensor auxiliar. Una vez hecha dicha configuración, el multiplexor de derivación de interfaz debe desactivarse para que el maestro I2C auxiliar MPU-9250 tome el control del bus de datos y pueda recopilar los datos de los sensores auxiliares. Según datos del fabricante, en esta configuración la dirección I2C para AK8963 es 0x0C. Con esto tomado en cuenta, las funciones utilizadas para obtener los datos e inicializar el AK8963 se detallan en el Cuadro 13.

Función	Parámetros	Descripción
AK8963_init	Número de puerto I2C Puntero a <i>calibration_t</i>	Esta función obtiene inicialmente el ID del sensor, y si este es igual a 0x0C, se procede a realizar los siguientes pasos: obtener los valores de ajuste de sensibilidad para cada eje y luego establecer el modo de operación <i>Continuous measurement mode 2</i> con una escritura sobre el registro CNTL1 con dirección 0x0A.
ak8963_get_- device_id	Puntero a variable <i>uint8_t</i>	Lectura del registro WIM con dirección 0x00. Es utilizada principalmente para comprobar que si exista comunicación con el sensor.
ak8963_set_cntl1	Variable <i>uint8_t</i>	Escritura sobre el registro CNTL1 con dirección 0x0A. Según el parámetro ingresado por el usuario se escoge el modo de operación de los bits 0 al 3 y la resolución de la data de salida en el bit 4.
ak8963_get_cntl1	Puntero a variable <i>uint8_t</i>	Lectura sobre el registro CNTL1 con dirección 0x0A para conocer el modo de operación en el que se encuentra el sensor y la resolución que tiene la data de salida.
ak8963_get_- sensitivity_- adjustment_- values	void	Función para obtener los valores de ajuste de sensibilidad para cada eje. Estos datos se encuentran en los registros ASAX, ASAY y ASAZ con direcciones 0x10, 0x11 y 0x12 respectivamente. Sin embargo, para realizar la lectura de estos registros es necesario colocar <i>Fuse ROM access</i> en el modo de operación del sensor. Para esto se coloca el código '1111' en los bits 0 al 3 del registro CNTL1. Una vez realizada la lectura, se regresa al sensor al modo de operación al que se encontraba previo a llamar a esta función.

*Continúa en la siguiente página*

Cuadro 13 – Continuación de la página previa

Función	Parámetros	Descripción
ak8963_get_- mag_raw	6 <i>bytes</i>	Lectura de los registros de los datos medidos para cada eje. Por especificaciones de la hoja de datos, luego de obtener los datos es necesario realizar una lectura en el registro ST2 con dirección 0x09.
ak8963_get_mag	Puntero a <i>vector_t</i>	Utiliza la función ak8963_get_-mag_raw para obtener la lectura de los datos para luego dividir el par de <i>bytes</i> correspondientes para cada eje. Una vez hecho esto se hacen los ajustes con los valores de sensibilidad y los atributos de <i>calibration_t</i> correspondientes para el magnetómetro. Los resultados finales se almacenan finalmente en los atributos del objeto <i>vector_t</i> ingresado por el usuario.
ak8963_print_- settings	void	Imprime en consola los siguientes datos: dirección I2C, ID, modo de operación actual, valores de ajuste de sensibilidad para cada eje y los atributos del objeto <i>calibration_t</i> correspondientes para este sensor, los cuales son <i>mag_offset</i> y <i>mag_scale</i> .

Cuadro 13: Funciones para inicialización, lectura de datos e impresión de configuraciones de sensor AK8963.

## 7.2.4. Modelo experimental

Realizando la medición de los ángulos *roll*, *pitch* y *yaw* fue posible validar, en un solo experimento, todas las librerías descritas en esta sección del capítulo. La descripción del experimento se presenta a continuación.

- **Mediciones de orientación.** Para comprobar que los datos impresos en consola sí son confiables, se procede a colocar la MPU-9250 directamente en la plataforma. Por tal razón es necesario que el armazón se encuentre diseñado y manufacturado al momento de realizar esta prueba de validación. Una vez hecho el montaje en conjunto con el ESP32, se realiza de forma manual el cambio de orientación en los tres ejes respectivos para luego graficar los datos resultantes.

## 7.2.5. Resultados

En los resultados esperados tras la realización de esta prueba se espera observar una variación en la curva del ángulo sobre el eje en la cual se está modificando la orientación. Añadido a esto, los otros dos ángulos deben de mantenerse constantes o, en todo caso, presentar ligeras variaciones que no sean lo suficientemente significativas. Para ello se tomaron 1000 muestras de datos por cada ángulo, para luego ser almacenados en un archivo con terminación .csv y posteriormente graficarlos con las herramientas disponibles en Excel.

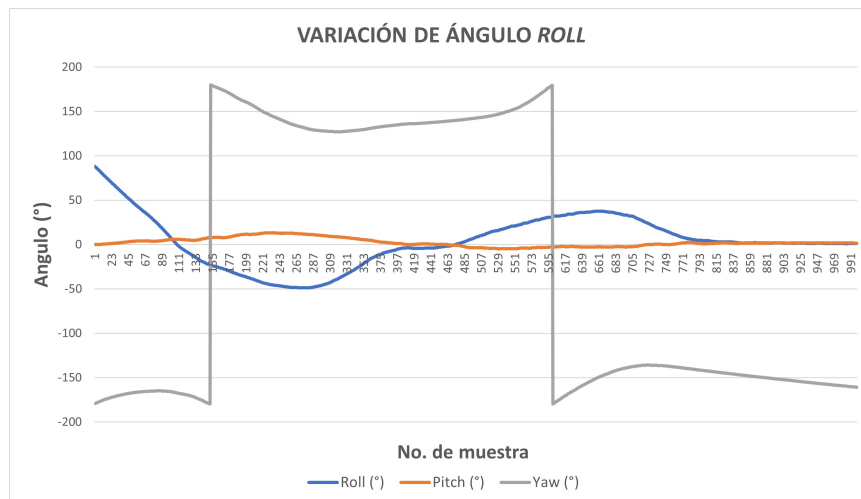


Figura 16: Resultado de variación de ángulo *roll*.

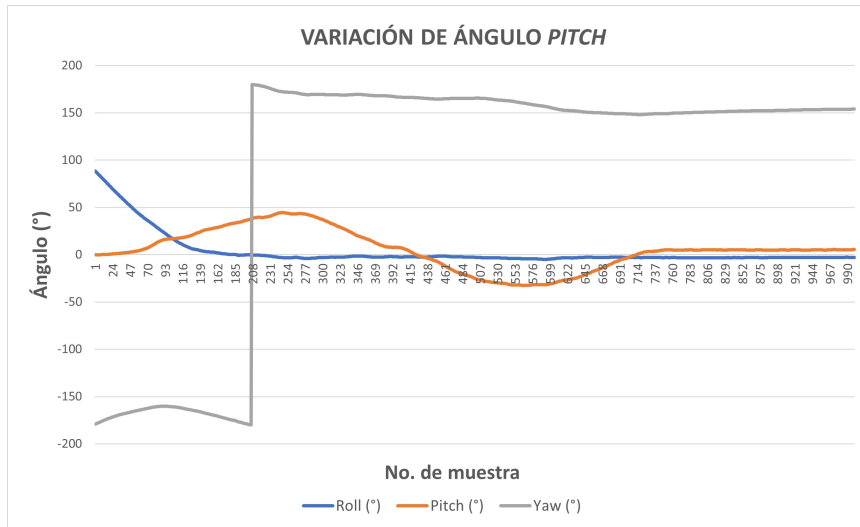


Figura 17: Resultado de variación de ángulo *pitch*

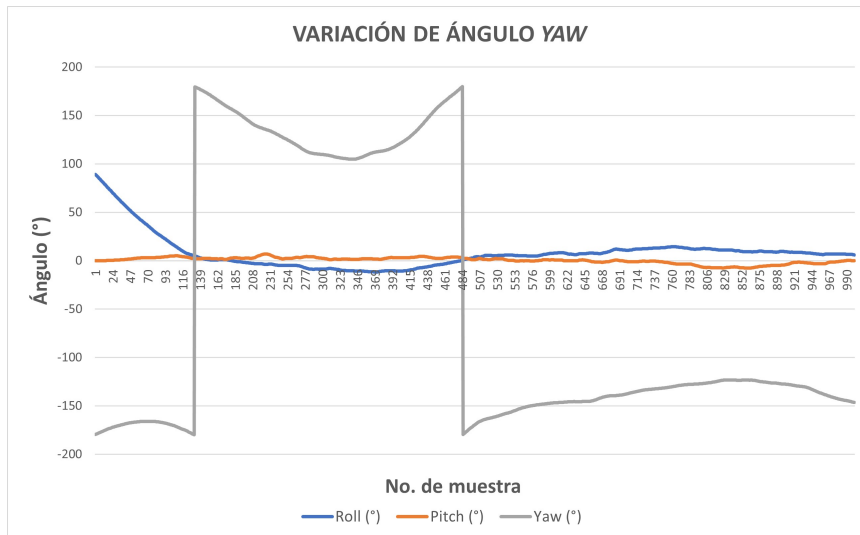


Figura 18: Resultado de variación de ángulo *yaw*

Con respecto a la Figura 16, se observa claramente cómo la curva que representa al ángulo *roll* va variando de forma suave conforme aumenta el número de muestras. El cambio de signo representa el punto en el cual la plataforma se volvió a colocar en su posición inicial y se comenzó a orientar hacia la dirección contraria. Cabe mencionar que el ángulo *pitch* se mantiene aproximadamente en cero durante todo el proceso de validación de este ángulo. Con respecto al ángulo *yaw* existen dos variaciones de signo bruscas. Sin embargo, independientemente del signo, los valores se mantienen relativamente constantes. Esto permite validar la correcta medición del ángulo *roll*.

La segunda prueba realizada, cuyos resultados se muestran en la Figura 17, representa la validación de la lectura de datos para el ángulo *pitch*. Al igual que con los resultados obtenidos para el ángulo *roll*, se logra observar la variación de forma suave de la curva respectiva conforme aumenta el número de muestras. Se aprecia nuevamente el cambio de

signo que representa la variación en la dirección de orientación. En esta ocasión los otros dos ángulos presentaron menor variación que en la prueba realizada anteriormente. El ángulo *roll* se mantuvo casi en su totalidad en cero, mientras que *yaw* presentó únicamente un cambio de signo abrupto para luego mantenerse aproximadamente constante hasta el final.

La tercera y última prueba realizada representó la validación de la lectura de datos para obtener el ángulo *yaw* y sus resultados se muestran en la Figura 18. A pesar de que los otros dos ángulos se mantienen aproximadamente constantes, *yaw* no presenta el comportamiento suave característico de las otras dos gráficas. Al realizar el cálculo de la derivada puede notarse que la pendiente para esta curva se mantiene en la misma dirección (positiva o negativa) antes y después de cada discontinuidad. Esto permite concluir que estos cambios abruptos se deben por el *map around* de los ángulos y, por ende, se valida su correcta medición.

Adicional a la estructura diseñada por Carlos Alonzo, la plataforma también requiere otros componentes físicos para lograr un correcto funcionamiento. Este capítulo trata sobre el proceso de selección de la batería y el diseño y manufactura de la placa electrónica. Para el caso de la batería, la importancia de su correcta selección recae en que impacta directamente en la fuerza de empuje de los motores y en el tiempo de vuelo disponible. Al mismo tiempo, el diseño y manufactura de la placa afecta en la electrónica de potencia y el control del cuadricóptero. Debido a que el proceso de diseño de la placa requiere conocer datos como el consumo de corriente o el voltaje nominal cuando la plataforma este en funcionamiento, el primer paso para la metodología de este capítulo consistió en enfocarse en el análisis de la batería.

## 8.1. Batería

La batería, además de cumplir con ciertas especificaciones técnicas, también es necesario que satisfaga requerimientos dados por el tipo de proyecto al cual pertenece. Esto último hace referencia a que esta plataforma de vuelo fue diseñada con fines didácticos y de investigación. Por tal razón, debe cumplir con ciertos rangos de costos o disponibilidad en el mercado para que una opción sea considerada como viable de implementar. Debido a que son varios los factores a considerar y al mismo tiempo se cuenta con diversas alternativas, el proceso de selección se realizó a través de un *trade study*, el cual se detalla más adelante en el capítulo.

### 8.1.1. Consumo de corriente motores

Uno de los principales requerimientos para la batería debe ser proporcionar la corriente requerida por los motores BLDC cuando estos se encuentren en el rango de velocidad necesaria para levantar la plataforma y ejercer control sobre ella. Conociendo este dato es posible determinar la capacidad mínima de la batería a utilizar.

#### Mediciones de corriente

El proceso para obtener una estimación de consumo de corriente por parte de los motores se realizó utilizando un multímetro Fluke 381 [47] y dos fuentes 2231A-30-3 marca KEITHLEY [48] colocadas en paralelo las cuales se encuentran disponibles en la universidad. La razón por la cual se colocaron estas fuentes en paralelo fue debido a la limitación de corriente que estas proporcionan, la cual es de 3A. La metodología utilizada fue ir aumentando la velocidad de un motor mientras se documentaban las corrientes picos y nominal para cada una. Posteriormente estos resultados se multiplicaron por un factor de cuatro para conocer el consumo total cuando todos los motores se encuentren funcionando de manera simultánea. Los resultados obtenidos se muestran en el Cuadro 14 utilizando tres cifras significativas.

Corriente (A)					
		Motores: 1		Motores: 4	
RPM	Nominal	Pico	Nominal	Pico	
3000	0.583	1.22	2.33	4.88	
4000	1.18	1.25	4.73	5.02	
5000	2.08	2.16	8.34	8.63	
6000	2.95	3.18	11.8	12.7	
7000	4.13	4.70	16.5	18.8	

Cuadro 14: Mediciones consumo de corriente de motores BLDC con fuentes en paralelo.

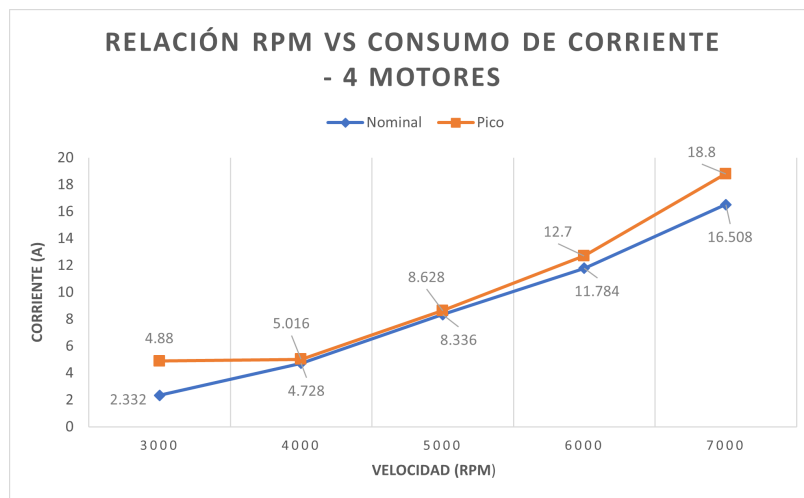


Figura 19: Relación RPM-Consumo de corriente.



Tal y como se observa en la Figura 19, el consumo de corriente presenta un crecimiento aproximadamente lineal conforme se aumenta la velocidad. Teniendo esto en consideración, se realizó el proceso de *trade study* para el análisis de las alternativas considerando ciertos criterios de selección.

### 8.1.2. *Trade Study*

Para lograr resultados confiables, el proceso se realizó a través de la calculadora en línea ofrecida por MechaniCalc [49].

## Identificación de alternativas

En el mercado se encuentran disponibles varios tipos de batería, sin embargo, para identificar cuáles eran las opciones recomendadas se procedió a revisar la hoja de datos del tipo de ESC utilizado. Según este documento, el voltaje de entrada puede ser suministrado por 2 a 4 celdas de Polímero de Litio (Li-Po)/Iones de Litio (Li-Ion) o bien por 5 a 12 celdas de Níquel-Hidruro Metálico (NiMH)/Níquel-Cadmio (NiCd) [50]. Partiendo de esto se realizó un estudio de mercado sencillo con el cual se lograron establecer las siguientes alternativas.

### 1. Baterías Li-Po

Los precios mostrados a continuación no toman en consideración el cargador, este tiene un costo aparte.

#### a) Opción 1

- 1) **Voltaje:** 11.1V (3 celdas).
- 2) **Capacidad:** 2200 mAh.
- 3) **Peso:** 166 gramos.
- 4) **Costo:** Q275.00
- 5) **Distribuidor:** La Electronica.



Figura 20: Opción 1 batería tipo Li-Po.

#### b) Opción 2

- 1) **Voltaje:** 11.1V (3 celdas).
- 2) **Capacidad:** 3300 mAh.

- 3) **Peso:** 320 gramos.
- 4) **Costo:** Q320.00
- 5) **Distribuidor:** Electrónica R&CH.



Figura 21: Opción 2 batería tipo Li-Po.

c) **Opción 3**

- 1) **Voltaje:** 11.1V (3 celdas).
- 2) **Capacidad:** 2200 mAh.
- 3) **Peso:** 175 gramos.
- 4) **Costo:** Q250.00
- 5) **Distribuidor:** Electrónica R&CH.



Figura 22: Opción 3 batería tipo Li-Po.

d) **Opción 4**

- 1) **Voltaje:** 11.1V (3 celdas).
- 2) **Capacidad:** 5200 mAh.
- 3) **Peso:** 400 gramos.
- 4) **Costo:** Q496.00
- 5) **Distribuidor:** Electrónica R&CH.



Figura 23: Opción 4 batería tipo Li-Po.

e) **Opción 5**

- 1) **Voltaje:** 14.8V (4 celdas).
- 2) **Capacidad:** 2800 mAh.
- 3) **Peso:** 295 gramos.
- 4) **Costo:** Q455.00
- 5) **Distribuidor:** Tettsa.



Figura 24: Opción 5 batería tipo Li-Po.

## 2. Baterías Li-Ion

Los precios mostrados a continuación toman en consideración el cargador correspondiente para este tipo de batería.

a) **Opción 1**

- 1) **Voltaje:** 12V.
- 2) **Capacidad:** 6800 mAh.
- 3) **Peso:** 287 gramos.
- 4) **Costo:** Q451.00
- 5) **Distribuidor:** OXDEA.



Figura 25: Opción 1 batería tipo Li-Ion 18650.

**b) Opción 2**

- 1) **Voltaje:** 12V.
- 2) **Capacidad:** 4400 mAh.
- 3) **Peso:** 330 gramos.
- 4) **Costo:** Q412.25
- 5) **Distribuidor:** OXDEA.



Figura 26: Opción 2 batería tipo Li-Ion 18650.

**c) Opción 3**

- 1) **Voltaje:** 12V.
- 2) **Capacidad:** 2600 mAh.
- 3) **Peso:** 200 gramos.
- 4) **Costo:** Q324.95
- 5) **Distribuidor:** OXDEA.



Figura 27: Opción 3 batería tipo Li-Ion 18650.

### 3. Baterías NiMH

Los precios mostrados a continuación no toman en consideración el cargador, exceptuando la opción 4.

#### a) Opción 1

- 1) **Voltaje:** 1.2V.
- 2) **Capacidad:** 1800 mAh.
- 3) **Peso:** 20 gramos.
- 4) **Costo:** Q62.00 (par)
- 5) **Distribuidor:** KEMIK.



Figura 28: Opción 1 batería tipo NiMH.

b) **Opción 2**

- 1) **Voltaje:** 1.2V.
- 2) **Capacidad:** 2500 mAh.
- 3) **Peso:** 17 gramos.
- 4) **Costo:** Q99.00 (kit de 4 baterías)
- 5) **Distribuidor:** Steren.



Figura 29: Opción 2 batería tipo NiMH.

c) **Opción 3**

- 1) **Voltaje:** 1.2V.
- 2) **Capacidad:** 2700 mAh.
- 3) **Peso:** 15 gramos.
- 4) **Costo:** Q75.00 (par)
- 5) **Distribuidor:** La Electrónica.



Figura 30: Opción 3 batería tipo NiMH.

d) **Opción 4**

- 1) **Voltaje:** 1.5V.
- 2) **Capacidad:** 2100 mAh.
- 3) **Peso:** 16 gramos.
- 4) **Costo:** Q179.00 (kit de 4 baterías con cargador).
- 5) **Distribuidor:** La Electrónica.



Figura 31: Opción 4 batería tipo NiMH.

4. **Baterías NiCd**

Los precios mostrados a continuación toman en consideración el cargador correspondiente para este tipo de batería.

a) **Opción 1**

- 1) **Voltaje:** 3.6V.
- 2) **Capacidad:** 650 mAh.
- 3) **Peso:** 35 gramos.
- 4) **Costo:** Q44.00.
- 5) **Distribuidor:** STEREN



Figura 32: Opción 1 batería tipo NiCd.

b) **Opción 2**

- 1) **Voltaje:** 3.6V.
- 2) **Capacidad:** 550 mAh.
- 3) **Peso:** 32 gramos.
- 4) **Costo:** Q35.00.
- 5) **Distribuidor:** RadioShack.



Figura 33: Opción 2 batería tipo NiCd.

### Criterios de selección

El *trade study* toma en cuenta los atributos críticos con los cuales debe cumplir la batería ideal que debe implementarse en la plataforma. La correcta identificación de dichos criterios permitió realizar un proceso de toma de decisión más rápido y efectivo.

- **Peso.** Debido a que los motores no son de gran capacidad, el peso juega un papel importante en la selección de la batería. Esto último se debe a que representa el elemento más pesado en el montaje de la plataforma. Según el análisis realizado por Carlos Alonzo en el diseño del armazón y también por recomendaciones de expertos en el área, para que sea posible ejercer control es necesario que cada motor se encuentre a no más de 1/3 de su capacidad al momento de que el sistema se eleve. Tomando en cuenta el peso de todos los elementos que lo conforman, incluyendo el armazón por sí mismo, la batería no debe pasarse de los 650 gramos para cumplir con lo mencionado anteriormente.
- **Duración de vuelo.** La plataforma será utilizada para fines didácticos y de investigación en un futuro. Por tal razón, se requirió por parte del personal encargado de laboratorio que se contara con un tiempo de vuelo aproximado de 25 a 30 minutos con una carga completa de la batería.



- **Disponibilidad.** Uno de los principales objetivos de este proyecto es lograr una reproducibilidad fácil de la plataforma. Adicionalmente, es posible que se desee adquirir varias unidades de la batería seleccionada para que al momento de hacer pruebas o demostraciones no se tenga que perder tiempo en esperar que se cargue una unidad para poder reanudar el proceso. Es por estas razones que la batería seleccionada debe poder adquirirse de manera sencilla y rápida, preferentemente en el mercado local.
- **Corriente.** Este criterio se encuentra relacionado con los resultados mostrados en el Cuadro 14. Considerando el caso crítico, los motores se mantendrán funcionando en el rango de 6000 a 7000 RPM durante el tiempo de vuelo de la plataforma. Esto indica que como mínimo la batería debe tener la capacidad de suministrar 12.7A continuos.
- **Cargador.** Debe ser posible adquirir el cargador específico según el tipo de batería seleccionada en el mercado local para que la solución sea considerada como viable.
- **Voltaje-Cantidad de celdas.** El voltaje de operación es un factor importante a tomar en cuenta debido a la relación 1000 RPM/V que presentan los motores BLDC utilizados. Cabe mencionar que entre mayor cantidad de celdas tenga una batería, más grandes serán las dimensiones de esta. Se espera que el voltaje de operación de la batería seleccionada sea de aproximadamente 10-15V. A pesar que en esta plataforma no se llegará a velocidad demasiado altas, se está considerando desde este punto la posibilidad de modificar los motores a unos con mayor capacidad. Es por ello que se desea dicho rango de voltaje.
- **Tiempo de vida.** Se desea que la batería seleccionada tenga un tiempo de vida lo más extendida posible. Esto ya que no es viable estar realizando una inversión de manera continua para adquirir este componente.
- **Costo.** La cuestión económica es un factor relevante al momento de diseñar un producto, especialmente cuando se desea una reproducibilidad fácil. Es por ello que la batería debe ser de bajo costo pero lo suficientemente buena para cumplir con todos los requisitos anteriores. Luego de una ligera investigación en el mercado, se espera que la batería seleccionada se encuentre en el rango de Q400.00 a Q550.00, incluyendo en este precio también su cargador correspondiente.

### Asignación de pesos a criterios

Los factores de ponderación son asignados según el grado de importancia relativa de cada criterio. Estas ponderaciones determinan en qué medida contribuye un criterio a la puntuación general. Una vez determinado dicho grado de importancia se realiza el cálculo de las ponderaciones a través de varios métodos. La calculadora utilizada en esta ocasión ofrece los métodos de ponderación lineal, por pares y el proceso de jerarquía analítica. La importancia de un criterio se califica en una escala de 1 al 9, siendo 1 la importancia más baja y 9 la más alta.

Criterio	Importancia	Ponderación AHP (%)	Ponderación por Pares (%)	Ponderación Lineal (%)
Peso	8	13.27	13.34	13.79
Duración de vuelo	7	8.08	7.99	12.07
Disponibilidad	8	13.27	13.34	13.79
Corriente	9	22.78	22.76	15.52
Cargador	4	2.74	2.69	6.90
Voltaje-Celdas	9	22.78	22.76	15.52
Tiempo de vida	8	13.27	13.34	13.79
Costo	5	3.82	3.78	8.62

Cuadro 15: Peso asignado para cada criterio según métodos de ponderación.

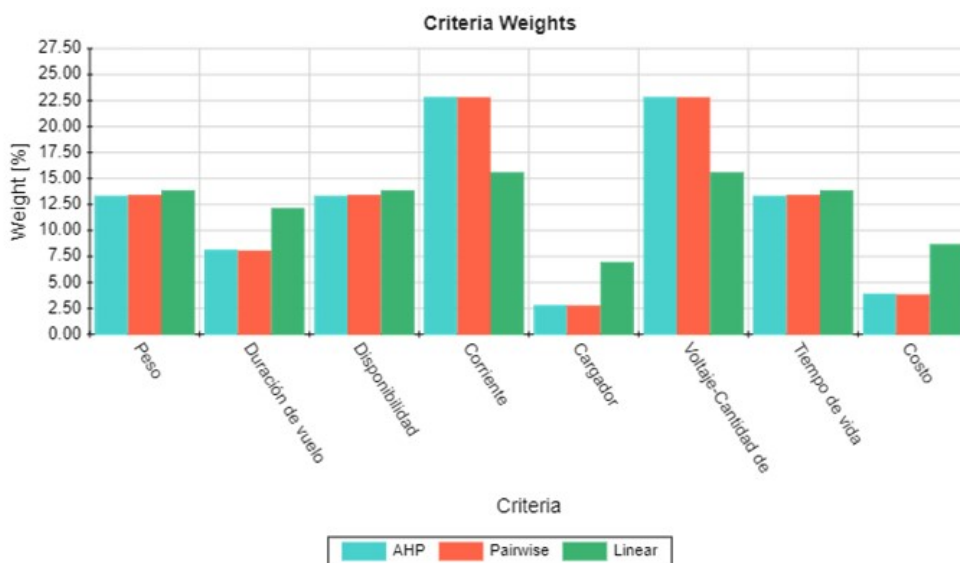


Figura 34: Gráfica de ponderación para criterios *trade study*.

Tal y como se observa en el Cuadro 15 y la Figura 34, los criterios con mayor peso para la toma de la decisión son la corriente y el voltaje, seguidos del peso, disponibilidad y tiempo de vida. Los criterios de cargador y costo son los menos significativos. A pesar que el costo sí es un dato a tomar en cuenta, los precios no son considerablemente elevados para este tipo de componentes, por ende no representa un problema cubrir estos gastos cuando se trata de fines académicos.

### Tasa de rendimiento

Cada alternativa fue calificada según su desempeño para cada criterio. La calificación puede ser según un valor numérico o subjetivo. Las tres posibles selecciones que permitió esta calculadora son las siguientes:

- **Cualitativo.** Únicamente se conoce una sensación subjetiva general de qué tan bien se desempeña cada alternativa. Se maneja en un rango de 1 al 5, donde el 1 es muy deficiente y 5 es excelente.

- **Cuantitativo - Valor alto mejor.** Se conocen valores numéricos para el rendimiento y un valor alto es mejor.
- **Cuantitativo- Valor bajo mejor.** Se conocen valores numéricos para el rendimiento y un valor bajo es mejor.

Una vez los valores se establecen para un criterio específico, estos valores son normalizados para que sumen 1. Los valores normalizados son utilizados posteriormente para los resultados finales. La tasa de rendimiento de las alternativas propuestas para cada criterio se muestran a continuación.

Peso Cuantitativo - Valor bajo mejor		
Alternativa	Valor (gr)	Normalizado
Opción 1 Li-Po	166.00	0.074
Opción 2 Li-Po	320.00	0.031
Opción 3 Li-Po	175.00	0.072
Opción 4 Li-Po	400.00	0.008
Opción 5 Li-Po	295.00	0.038
Opción 1 Li-Ion	287.00	0.040
Opción 2 Li-Ion	330.00	0.028
Opción 3 Li-Ion	200.00	0.065
Opción 1 NiMH	30.00	0.112
Opción 2 NiMH	78.00	0.099
Opción 3 NiMH	40.00	0.110
Opción 4 NiMH	74.00	0.100
Opción 1 NiCd	35.00	0.111
Opción 2 NiCd	32.00	0.112

Cuadro 16: Rendimiento de cada alternativa para el criterio de peso.

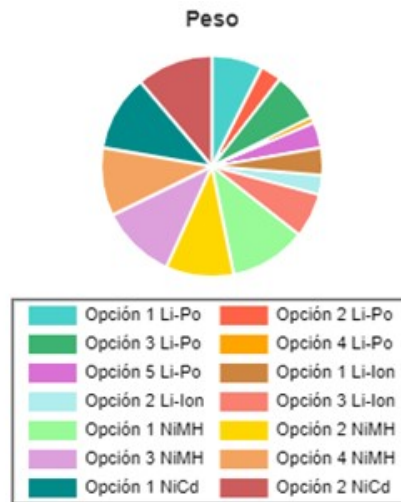


Figura 35: Rendimiento de cada alternativa para el criterio de peso *trade study*.

Para las opciones de las baterías NiMH se consideró el peso total del empaquetado, más un aproximado de 10 gramos el cual representa la pieza que permite conectarlas a la placa de la plataforma.

Duración de vuelo Cuantitativo - Valor alto mejor		
Alternativa	Valor (min)	Normalizado
Opción 1 Li-Po	10.39	0.038
Opción 2 Li-Po	15.59	0.057
Opción 3 Li-Po	10.39	0.038
Opción 4 Li-Po	24.57	0.090
Opción 5 Li-Po	13.23	0.048
Opción 1 Li-Ion	32.13	0.117
Opción 2 Li-Ion	20.79	0.076
Opción 3 Li-Ion	12.28	0.045
Opción 1 NiMH	17.01	0.062
Opción 2 NiMH	47.24	0.172
Opción 3 NiMH	25.51	0.093
Opción 4 NiMH	39.69	0.145
Opción 1 NiCd	3.07	0.011
Opción 2 NiCd	2.60	0.009

Cuadro 17: Rendimiento de cada alternativa para el criterio de tiempo de vuelo.

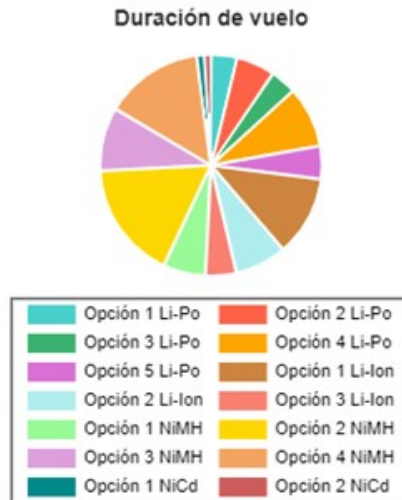


Figura 36: Rendimiento de cada alternativa para el criterio de tiempo de vuelo *trade study*.

Para obtener los valores del Cuadro 17 se convirtió inicialmente la capacidad de cada alternativa a unidad [Ah]. Posteriormente se dividió este dato con la corriente mínima requerida la cual era de 12.7A. Finalmente este resultado, expresado en horas, se traslada a unidades de minutos.

Disponibilidad Cualitativo		
Alternativa	Valor	Normalizado
Opción 1 Li-Po	3.00	0.053
Opción 2 Li-Po	5.00	0.088
Opción 3 Li-Po	5.00	0.088
Opción 4 Li-Po	5.00	0.088
Opción 5 Li-Po	5.00	0.088
Opción 1 Li-Ion	5.00	0.088
Opción 2 Li-Ion	4.00	0.070
Opción 3 Li-Ion	4.00	0.070
Opción 1 NiMH	3.00	0.053
Opción 2 NiMH	2.00	0.035
Opción 3 NiMH	4.00	0.070
Opción 4 NiMH	2.00	0.035
Opción 1 NiCd	5.00	0.088
Opción 2 NiCd	5.00	0.088

Cuadro 18: Rendimiento de cada alternativa para el criterio de disponibilidad.

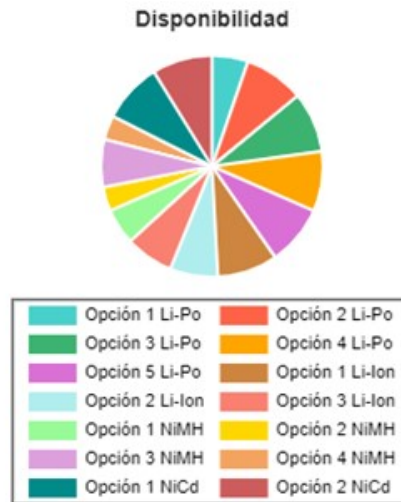


Figura 37: Rendimiento de cada alternativa para el criterio de disponibilidad *trade study*.

Los valores mostrados en el Cuadro 18 consideran si el distribuidor contaba con la alternativa respectiva en el inventario al momento de realizar este estudio. En caso no se encontrara disponible, también se evaluó el rango de tiempo de espera para que ingrese nuevamente en la mercadería.

Corriente		
Cuantitativo - Valor alto mejor		
Alternativa	Valor (mAh)	Normalizado
Opción 1 Li-Po	2200.0	0.038
Opción 2 Li-Po	3300.0	0.057
Opción 3 Li-Po	2200.0	0.038
Opción 4 Li-Po	5200.0	0.090
Opción 5 Li-Po	2800.0	0.048
Opción 1 Li-Ion	6800.0	0.117
Opción 2 Li-Ion	4400.0	0.076
Opción 3 Li-Ion	2600.0	0.045
Opción 1 NiMH	3600.0	0.062
Opción 2 NiMH	10000.0	0.172
Opción 3 NiMH	5400.0	0.093
Opción 4 NiMH	8400.0	0.145
Opción 1 NiCd	650.0	0.011
Opción 2 NiCd	550.0	0.009

Cuadro 19: Rendimiento de cada alternativa para el criterio de corriente.

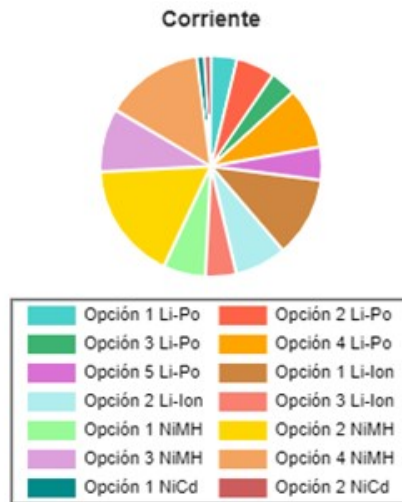


Figura 38: Rendimiento de cada alternativa para el criterio de corriente *trade study*.

Para las opciones de las baterías NiMH se consideró la corriente total del paquete respectivo. Es decir, se multiplicó la corriente disponible por pila por la cantidad de baterías que incluía el empaquetado. En la Figura 38 se logra apreciar una clara diferencia en el desempeño de este criterio entre algunas de las alternativas propuestas.

Cargador Cualitativo		
Alternativa	Valor	Normalizado
Opción 1 Li-Po	5.00	0.076
Opción 2 Li-Po	5.00	0.076
Opción 3 Li-Po	5.00	0.076
Opción 4 Li-Po	5.00	0.076
Opción 5 Li-Po	5.00	0.076
Opción 1 Li-Ion	5.00	0.076
Opción 2 Li-Ion	5.00	0.076
Opción 3 Li-Ion	5.00	0.076
Opción 1 NiMH	4.00	0.061
Opción 2 NiMH	4.00	0.061
Opción 3 NiMH	4.00	0.061
Opción 4 NiMH	4.00	0.061
Opción 1 NiCd	5.00	0.076
Opción 2 NiCd	5.00	0.076

Cuadro 20: Rendimiento de cada alternativa para el criterio de cargador.

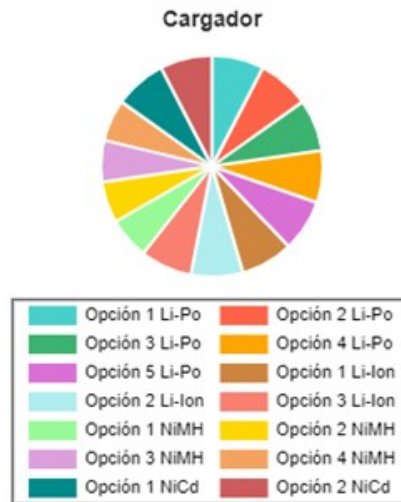


Figura 39: Rendimiento de cada alternativa para el criterio de cargador *trade study*.

Al igual que en el criterio de disponibilidad, el desempeño de cada alternativa es función si el elemento se encuentra disponible en inventario o, caso contrario, el tiempo aproximado para que se ingrese nuevamente en la mercadería. El desempeño para este criterio es casi idéntico para todas las alternativas.

Voltaje-Cantidad de Celdas Cuantitativo - Valor alto mejor		
Alternativa	Valor (V)	Normalizado
Opción 1 Li-Po	11.10	0.094
Opción 2 Li-Po	11.10	0.094
Opción 3 Li-Po	11.10	0.094
Opción 4 Li-Po	11.10	0.094
Opción 5 Li-Po	14.80	0.125
Opción 1 Li-Ion	12.00	0.102
Opción 2 Li-Ion	12.00	0.102
Opción 3 Li-Ion	12.00	0.102
Opción 1 NiMH	2.40	0.020
Opción 2 NiMH	4.80	0.041
Opción 3 NiMH	2.40	0.020
Opción 4 NiMH	6.00	0.051
Opción 1 NiCd	3.60	0.031
Opción 2 NiCd	3.60	0.031

Cuadro 21: Rendimiento para cada alternativa para el criterio de voltaje.

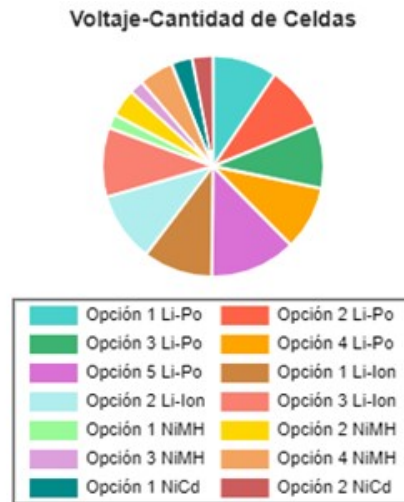


Figura 40: Rendimiento de cada alternativa para el criterio de voltaje *trade study*.

Para este criterio las baterías tipo Li-Po y Li-Ion presentaron una ventaja competitiva clara en comparación con las NiMH y las NiCd. Esta clara diferencia de valores en conjunto con la importancia asignada, y por ende el peso calculado, para este criterio, pueden afectar de manera significativa en los resultados finales.



Tiempo de Vida Cualitativo		
Alternativa	Valor	Normalizado
Opción 1 Li-Po	5.00	0.102
Opción 2 Li-Po	5.00	0.102
Opción 3 Li-Po	5.00	0.102
Opción 4 Li-Po	5.00	0.102
Opción 5 Li-Po	5.00	0.102
Opción 1 Li-Ion	5.00	0.102
Opción 2 Li-Ion	5.00	0.102
Opción 3 Li-Ion	2.00	0.041
Opción 1 NiMH	2.00	0.041
Opción 2 NiMH	2.00	0.041
Opción 3 NiMH	2.00	0.041
Opción 4 NiMH	2.00	0.041
Opción 1 NiCd	2.00	0.041
Opción 2 NiCd	2.00	0.041

Cuadro 22: Rendimiento para cada alternativa para el criterio de tiempo de vida.

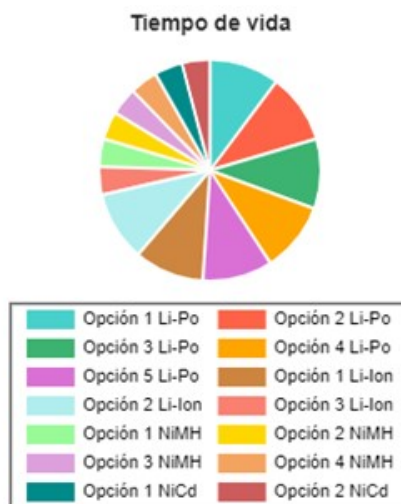


Figura 41: Rendimiento de cada alternativa para el criterio de tiempo de vida *trade study*.

En este criterio nuevamente se marcó una diferencia significativa entre las baterías Li-Po/Li-Ion y las NiMH/NiCd. Esto se debe a que las últimas, al estar compuestas de Níquel, sufren lo que se conoce como el efecto memoria. Este se produce cuando se cargan las baterías sin haberlas descargado previamente de forma completa. Como consecuencia se crean cristales en el interior de estas baterías que provocan que no se puedan volver a cargar a su capacidad máxima por el resto de su vida útil [51].

Costo Cuantitativo - Valor bajo mejor		
Alternativa	Valor (Q)	Normalizado
Opción 1 Li-Po	410.00	0.056
Opción 2 Li-Po	455.00	0.048
Opción 3 Li-Po	385.00	0.060
Opción 4 Li-Po	631.00	0.017
Opción 5 Li-Po	590.00	0.024
Opción 1 Li-Ion	451.00	0.049
Opción 2 Li-Ion	412.00	0.055
Opción 3 Li-Ion	324.00	0.071
Opción 1 NiMH	137.00	0.104
Opción 2 NiMH	174.00	0.097
Opción 3 NiMH	150.00	0.102
Opción 4 NiMH	179.00	0.097
Opción 1 NiCd	104.00	0.110
Opción 2 NiCd	95.00	0.111

Cuadro 23: Rendimiento para cada alternativa para el criterio de costo.

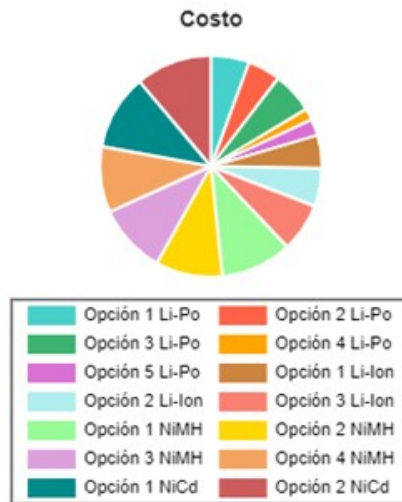


Figura 42: Rendimiento de cada alternativa para el criterio de costo *trade study*.

En los valores presentados en el Cuadro 23 ya se encuentra incluido el costo del cargador o un aproximado de este. Para el caso del cargador de las baterías tipo Li-Po se considera un precio de Q135.00 cotizado en la electrónica R&CH. En el caso de las baterías de Níquel se aproximan precios de Q75.00 para las de tipo NiMH y Q60.00 para las NiCd. Estos precios fueron considerados según consultas a diversas electrónicas a nivel local.

## Resultados

Debido a que se consideraron una cantidad considerable de alternativas y criterios de selección, se optó por utilizar el proceso de jerarquía analítica como método de ponderación. Esto ya que, como se mencionó en el marco teórico, se suele utilizar para casos de alta complejidad. Los resultados finales se calcularon tomando el desempeño de cada alternativa para cada criterio y multiplicando el valor del desempeño por la ponderación de los criterios. Las puntuaciones ponderadas se suman para cada alternativa y luego se escalan para que la alternativa con la puntuación más alta tenga una puntuación de 100. Los resultados finales se muestran en el Cuadro 24 y Figura 43, los cuales dan como elección ideal a la opción 1 de las baterías tipo Li-Ion.

Resultados		
Alternativa	Suma de puntuación ponderada	Puntuación final
Opción 1 Li-Po	0.068	72
Opción 2 Li-Po	0.072	77
Opción 3 Li-Po	0.072	77
Opción 4 Li-Po	0.078	83
Opción 5 Li-Po	0.077	82
Opción 1 Li-Ion	0.094	100
Opción 2 Li-Ion	0.077	83
Opción 3 Li-Ion	0.065	70
Opción 1 NiMH	0.057	60
Opción 2 NiMH	0.091	97
Opción 3 NiMH	0.068	73
Opción 4 NiMH	0.085	91
Opción 1 NiCd	0.049	52
Opción 2 NiCd	0.048	51

Cuadro 24: Resultados finales *trade study*.

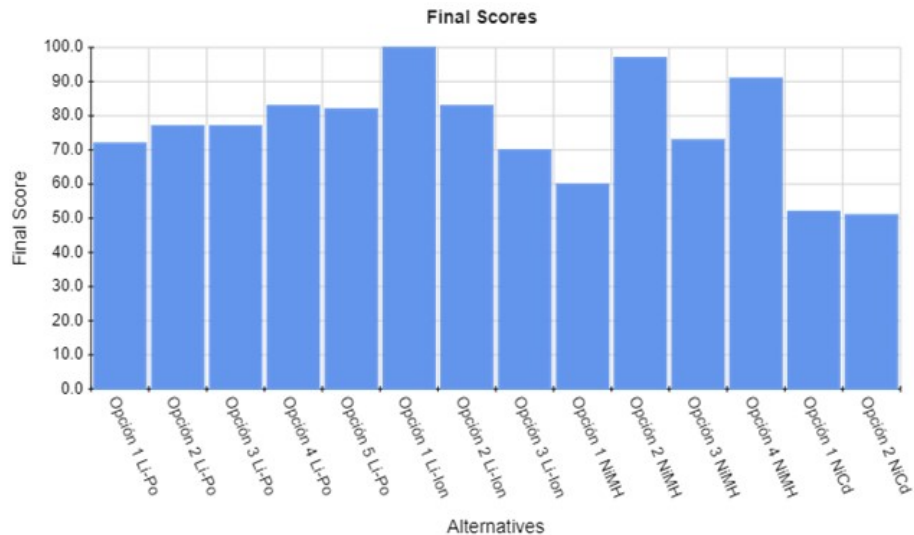


Figura 43: Resultados finales *trade study*.

### 8.1.3. Modelo experimental

Para la validación del *trade study* es necesario comprobar los criterios de corriente proporcionada y duración de vuelo. La razón por la cual únicamente estos dos son necesarios de verificar se debe a que son los requisitos técnicos planteados. Los demás ya fueron examinados y solucionados al momento de seleccionar el modelo de la batería. La prueba realizada se describen a continuación.

- **Mediciones de corriente.** En esta prueba se realiza el mismo proceso descrito en la sección 8.1.1, con las diferencias de que en esta ocasión se utilizará la batería en lugar de las dos fuentes 2231A-30-3 marca KEITHLEY en paralelo y las mediciones se harán únicamente para los 4 motores funcionando simultáneamente.

### 8.1.4. Resultados

#### Corriente

En esta ocasión la medición de corriente se realizó en más rangos de velocidades, a diferencia de los datos presentados en el Cuadro 14, debido a que se quería conocer a mayor detalle la tendencia de consumo que mostraba la plataforma con la batería. Cabe mencionar que al llegar a una velocidad de 8000 RPM la batería presentaba un ligero calentamiento. Los resultados obtenidos se muestran en el Cuadro 25 utilizando 3 cifras significativas.

Corriente (A)		
Motores: 4		
RPM	Nominal	Pico
3000	1.50	2.20
3500	2.50	2.60
4000	3.60	3.90
4500	4.60	5.00
5000	5.90	6.20
5500	7.10	7.50
6000	8.20	8.70
6500	9.90	10.3
7000	11.0	11.1
7500	11.2	11.4

Cuadro 25: Mediciones consumo de corriente de motores BLDC con batería seleccionada.

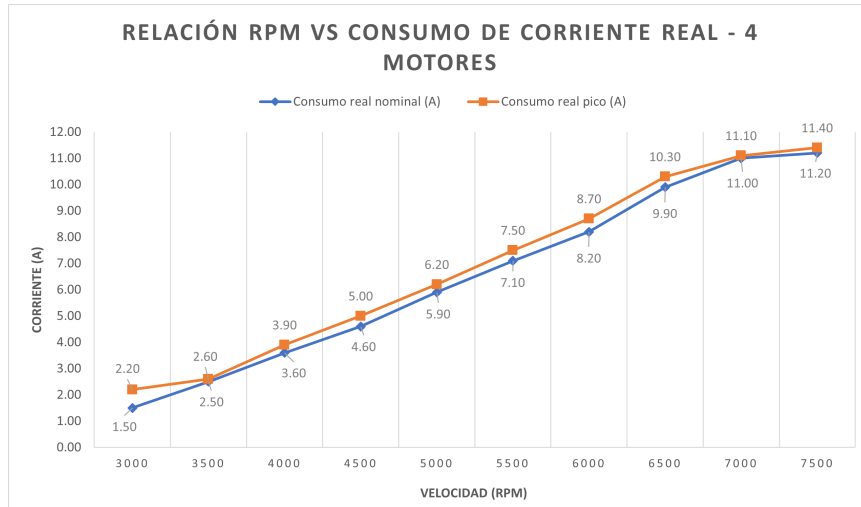


Figura 44: Relación RPM-Consumo de corriente real.

Al igual que en la Figura 19, el consumo de corriente aumenta linealmente respecto a la velocidad tal y como se muestra en la Figura 69. Sin embargo, a pesar que se presenta la misma tendencia, el consumo en este caso disminuyó de manera considerable. Esto último es beneficioso para el proyecto ya que permite aumentar el tiempo de vuelo pronosticado inicialmente o bien aumentar el rango de velocidades permitidas y aún manteniendo los 20-30 minutos de vuelo requeridos.

## 8.2. PCB

La fabricación de una placa electrónica fue vital para este proyecto por cuestiones de orden, estética, ahorro de componentes y optimización de espacio. Esto último toma relevancia ya que se cuenta con un espacio físico limitado en la estructura de la plataforma para la conexión de componentes. Adicional a esto, una correcta distribución de potencia hacia los motores sería difícil de realizar sin este elemento. Partiendo de los datos de corriente presentados en el Cuadro 25, y que el voltaje de alimentación era de 12V, fue posible comenzar el proceso de diseño. Esta placa, con dimensiones 75 mm × 110 mm, fue diseñada en el *software* Altium Designer bajo la norma IPC-2221A y manufacturada en la fresadora LPKF S103 disponible en la universidad.

### 8.2.1. Componentes

Debido a que la plataforma de vuelo es de bajo costo para lograr una fácil reproducibilidad, la placa consta de pocos componentes, sin embargo, cada uno de estos cumple con una aplicación fundamental. La principal limitante en la selección de los componentes utilizados para la conexión de los ESC es que cumplan con el rango de corriente requerido. En el Cuadro 26 se presenta a detalle la información de todos los elementos.

Componentes placa electrónica				
No.	Componente	Cantidad	Proveedor	Descripción
1	ESP32	1	La Electronica	Microcontrolador
2	MPU-9250	1	Universidad	Medición de orientación
3	Bornera 2 pines	5	La Electronica	Bornera 300V a 15A
4	Switch ON/OFF	1	Tettsa	20A, 12 DC, 3 pines
5	Bornera 3 pines	1	La Electronica	Bornera 300V a 15A
6	Conector	4	La Electronica	Conector macho a placa de 3 pines
7	Header tipo hembra	48	Tettsa	Header 2.54 mm

Cuadro 26: Componentes PCB.

## Costos

La totalidad de los componentes pueden obtenerse a nivel local, lo cual disminuye considerablemente el costo de manufactura. Los precios presentados en el Cuadro 27 son válidos para el mes de septiembre del año 2021, por lo que estos pueden variar ligeramente cuando se requiera la manufactura de otra plataforma igual en el futuro.

Precios componentes				
No.	Componente	Cantidad	Precio unitario	Total
1	ESP32	1	Q 120.00	Q 120.00
2	MPU-9250 <sup>1</sup>	1	Q 79.00	Q 9.00
3	Bornera 2 pines	5	Q 2.00	Q 10.00
4	Switch ON/OFF	1	Q 10.00	Q 10.00
5	Bornera 3 pines	1	Q 3.00	Q 3.00
6	Conector	4	Q 1.50	Q 6.00
7	Header tipo hembra <sup>2</sup>	48	Q 6.50	Q 13.00
<b>Total</b>				<b>Q 241.00</b>

Cuadro 27: Costos componentes PCB.

## Función por componente

- **ESP32:** Este componente es el cerebro de toda la plataforma. Es el encargado de establecer los protocolos de comunicación I2C y MQTT con la MPU-9250 y el sistema OptiTrack respectivamente. Adicionalmente implementa el controlador en conjunto con los motores BLDC.
- **MPU-9250:** Proporciona datos de orientación de la plataforma a través de sus 3 sensores: giroscopio, acelerómetro y magnetómetro. Estos datos son convertidos posteriormente a valores de ángulos *roll*, *pitch* y *yaw*.
- **Bornera 2 pines:** Una de estas borneras se encarga de recibir la alimentación directamente de la batería, mientras que las otras cuatro permiten a los ESC conectarse a la placa y, por ende, ser energizados.

- **Switch ON/OFF:** Permite conectar o desconectar la alimentación proveniente de la batería, y por lo tanto, encender o apagar a toda la plataforma. Este elemento se complementa con la bornera de 3 pines.
- **Bornera 3 pines:** El *switch* seleccionado presenta dimensiones considerablemente grandes: pines de tamaño 6.60 mm y separación entre ellos de 6.35 mm. Esto se debe principalmente por la capacidad de voltaje y corriente que este presenta. Las dimensiones representan un problema al momento de fabricar la placa, principalmente por el tamaño de agujeros que se deben realizar. Por ende se optó por utilizar una bornera de 3 pines y de esta sacar cables que conecten con el *switch*.
- **Conector macho de 3 pines:** Logra establecer la comunicación de los conectores servo del ESC con la placa. Con esto se logra de que el ESP32 pueda enviar las señales de PWM correspondientes a cada motor, al mismo tiempo de interconectar tierras.
- **Header tipo hembra:** Se utilizan con el propósito de no soldar el ESP32 y la MPU-9250 directamente a la placa. En el caso de que sea necesario remover uno de estos componentes para reemplazarlo por alguna falla el proceso será mucho más sencillo y rápido en tener que desoldar pin por pin. Se utilizarán 38 *headers* para la conexión del ESP32 y 10 para la MPU-9250.

### 8.2.2. Esquemático

El esquemático mostrado en la Figura 45 representa de forma gráfica las conexiones entre los elementos con las que cuenta la placa. Por la misma razón, este fue el primer paso en el proceso de diseño en Altium Designer. Dentro de dichas conexiones se considera la forma de alimentación hacia los módulos MPU-9250 y ESP32. Para el primero la alimentación debe de ser de 3.3V, los cuales son obtenidos directamente del pin 1 del ESP32. En el caso del microcontrolador, la alimentación puede obtenerse de tres formas diferentes según el fabricante: puerto micro USB, 5V/GND o 3V3/GND *header pins*. Ya que la plataforma se encontrará alejada de una computadora, la alimentación por el puerto micro USB queda descartada. La solución para este problema son los conectores servo con los que cuentan los ESC. Estos conectores tienen un pin que proporciona 5V y 2A. Esto evita utilizar un regulador de voltaje externo que disminuya el voltaje de la batería al rango requerido.

El esquemático puede dividirse en dos secciones: la parte izquierda representa la parte de potencia y la derecha el control. En la parte de control se utilizan los pines 26, 17, 4 y 2 para las señales PWM y los pines 22 y 21 para las señales SCL y SDA, respectivamente, del protocolo de comunicación I2C con la MPU-9250. A pesar de que en realidad la placa contará con una bornera de 3 pines, en la Figura 45 este elemento se representa como un *switch* ya que este será su verdadero propósito.

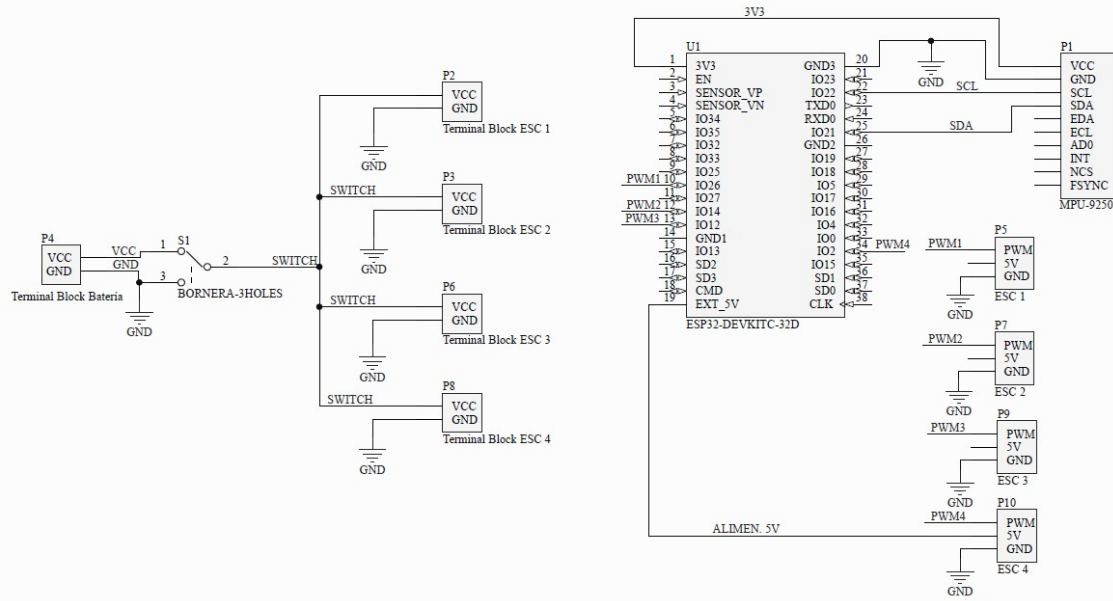


Figura 45: Esquemático de la PCB.

### 8.2.3. Wire routing

Una vez realizado el esquemático el siguiente paso fue realizar el proceso de *wire routing* de la PCB. Para ello fue necesario hacer el cálculo del ancho de pista (*trace width*) y espacio de vía requerido (*clearance*) bajo la norma ANSI IPC-2221A para cada nodo del circuito mostrado en la Figura 45. Este cálculo toma los siguientes parámetros de entrada: corriente, aumento de temperatura, espesor de cobre, temperatura ambiente, longitud del conductor y voltaje pico. Para todos los cálculos realizados se consideró un aumento máximo de temperatura de  $10^{\circ}\text{C}$ , espesor de  $1\text{ oz}/\text{ft}^2$  según las especificaciones de las placas disponibles en la universidad, temperatura ambiente de  $25^{\circ}\text{C}$  y una longitud promedio del conductor de 3 in. Los datos que varían para cada nodo son la corriente y el voltaje pico. Dichos cálculos se realizaron en [52] y los resultados se muestran en las Figuras 46, 47 y 48.

ANSI PCB TRACE WIDTH CALCULATOR							
Input Data			Results Data				
Field	Value	Units	Trace Data	Internal Traces		External Traces	
				Value	Units	Value	Units
Current (max. 35A)	11.5	Amps	Required Trace Width	916.2	mil	352.19	mil
Temperature Rise (max. 100°C)	10	°C	Cross-section Area	1231.19	mil <sup>2</sup>	473.27	mil <sup>2</sup>
Cu thickness	1	oz/ft <sup>2</sup>	Resistance	0	Ω Ohms	0	Ω Ohms
Ambient Temperature	25	°C	Voltage Drop	0.02	Volts	0.05	Volts
Conductor Length	3	inches	Loss	0.23	Watts	0.6	Watts
Peak Voltage	12	Volts	Required Track Clearance	25.4	mil		

Figura 46: Cálculo de ancho de pista y espacio de vía para los nodos VCC, GND y SWITCH.



ANSI PCB TRACE WIDTH CALCULATOR							
Input Data			Results Data				
Field	Value	Units	Trace Data	Internal Traces		External Traces	
				Value	Units	Value	Units
Current (max. 35A)	2	Amps	Required Trace Width	82.07	mil	31.55	mil
Temperature Rise (max. 100°C)	10	°C	Cross-section Area	110.28	mil <sup>2</sup>	42.39	mil <sup>2</sup>
Cu thickness	1	oz/ft <sup>2</sup>	Resistance	0.02	Ω Ohms	0.05	Ω Ohms
Ambient Temperature	25	°C	Voltage Drop	0.04	Volts	0.1	Volts
Conductor Length	3	inches	Loss	0.08	Watts	0.2	Watts
Peak Voltage	5	Volts	Required Track Clearance	24	mil		

Figura 47: Cálculo de ancho de pista y espacio de vía para el nodo 5V.

ANSI PCB TRACE WIDTH CALCULATOR							
Input Data			Results Data				
Field	Value	Units	Trace Data	Internal Traces		External Traces	
				Value	Units	Value	Units
Current (max. 35A)	40	mA	Required Trace Width	0.37	mil	0.14	mil
Temperature Rise (max. 100°C)	10	°C	Cross-section Area	0.5	mil <sup>2</sup>	0.19	mil <sup>2</sup>
Cu thickness	1	oz/ft <sup>2</sup>	Resistance	4.31	Ω Ohms	11.22	Ω Ohms
Ambient Temperature	25	°C	Voltage Drop	0.17	Volts	0.45	Volts
Conductor Length	3	inches	Loss	0.01	Watts	0.02	Watts
Peak Voltage	3.3	Volts	Required Track Clearance	23.66	mil		

Figura 48: Cálculo de ancho de pista y espacio de vía para los nodos SCL, SDA, PWM y 3V3.

Como es posible observar en los resultados, existe una diferencia significativa entre el ancho de pista de los nodos VCC, GND y SWITCH con el resto. Esto se debe principalmente por las magnitudes de corriente, ya que para los nodos mencionados anteriormente se tiene una corriente máxima de 11.5A. Este último dato se obtuvo de los resultados mostrados en el Cuadro 25. Con respecto al espacio de vía los resultados variaron de forma mínima. Considerando que el ancho de pista y el espacio mínimo entre ellas recomendado debe de ser de 20 mil en ambos casos, se procedió a utilizar los datos presentados en el Cuadro 28.

Nodo	Ancho de pista			Espacio de vía		
	Calculadora (mil)	Mínimo (mil)	Utilizado (mil)	Calculadora (mil)	Mínimo (mil)	Utilizado (mil)
VCC	352.19	20.00	<b>355.00</b>	25.40	20.00	<b>26.00</b>
GND	352.19	20.00	<b>355.00</b>	25.40	20.00	<b>26.00</b>
SWITCH	352.19	20.00	<b>355.00</b>	25.40	20.00	<b>26.00</b>
5V	31.550	20.00	<b>35.00</b>	24.00	20.00	<b>26.00</b>
PWM	0.140	20.00	<b>20.00</b>	23.66	20.00	<b>26.00</b>
SCL	0.140	20.00	<b>20.00</b>	23.66	20.00	<b>26.00</b>
SDA	0.140	20.00	<b>20.00</b>	23.66	20.00	<b>26.00</b>
3V3	0.140	20.00	<b>20.00</b>	23.66	20.00	<b>26.00</b>

Cuadro 28: Anchos y separación de pistas utilizadas para el proceso de *wire routing* de la PCB.

El ancho de pista necesario para los primeros tres nodos del Cuadro 28 representó un problema ya que las dimensiones de la placa no son tan grandes. Por esta razón se optó como estrategia utilizar un polígono de GND que cubriera la mayor parte de la capa inferior de la placa. Esto no solo optimiza el proceso de diseño sino que también permite hacer la fabricación en una sola cara, lo cual presenta múltiples ventajas como facilidad de soldadura y descarte del uso de vías. El resultado se puede observar en las Figuras 49 y 50.

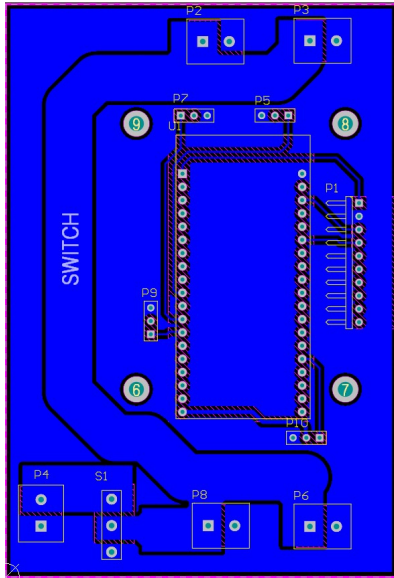


Figura 49: Vista de la capa inferior de la PCB desde Altium Designer.

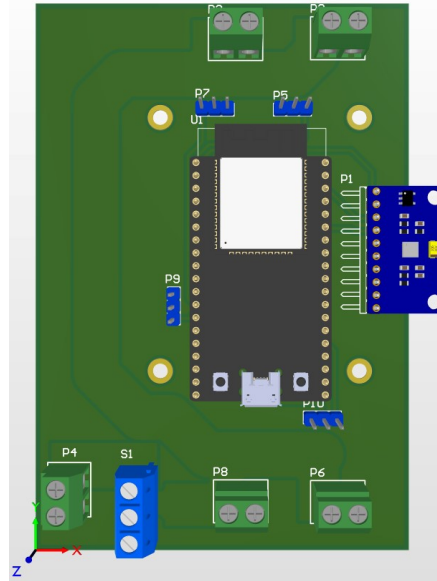


Figura 50: Vista 3D de la PCB desde Altium Designer .

#### 8.2.4. Manufactura y validación

Como se había mencionado al inicio de esta sección, la placa fue manufacturada en la fresadora LPKF S103. Cabe mencionar que, tal y como se observa en las Figuras 49 y 50, se colocaron cuatro agujeros adicionales por motivos de montaje de la placa en el cuadricóptero. Se dejó un pequeño *pad* en cada uno de estos agujeros por precaución de evitar continuidad entre los tornillos y la placa. Los tornillos utilizados fueron M3 × 30. Posterior a soldar los elementos respectivos, el resultado final de la placa se presentan en las Figuras 51 y 52.

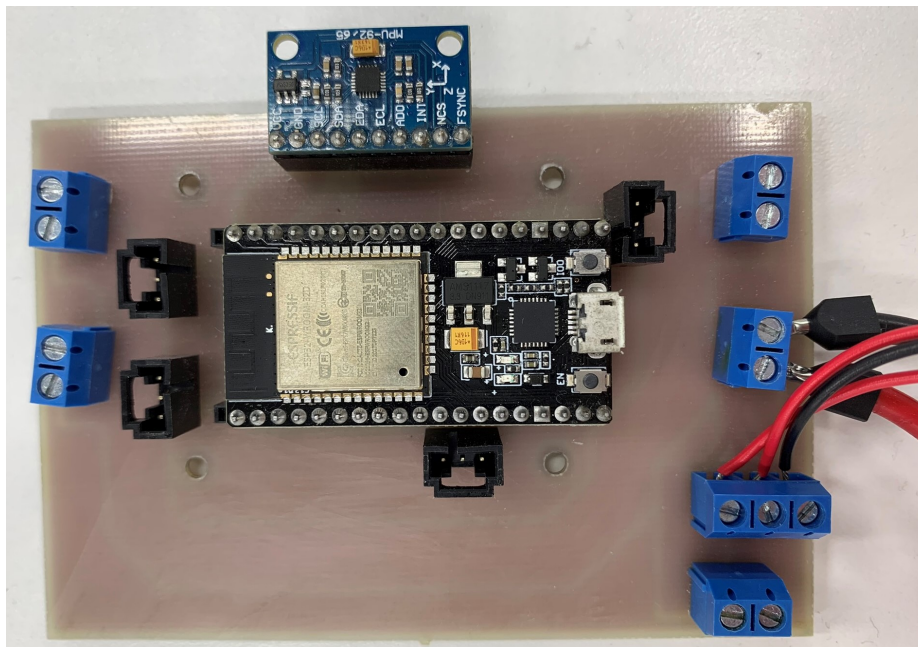


Figura 51: Vista superior placa manufacturada.

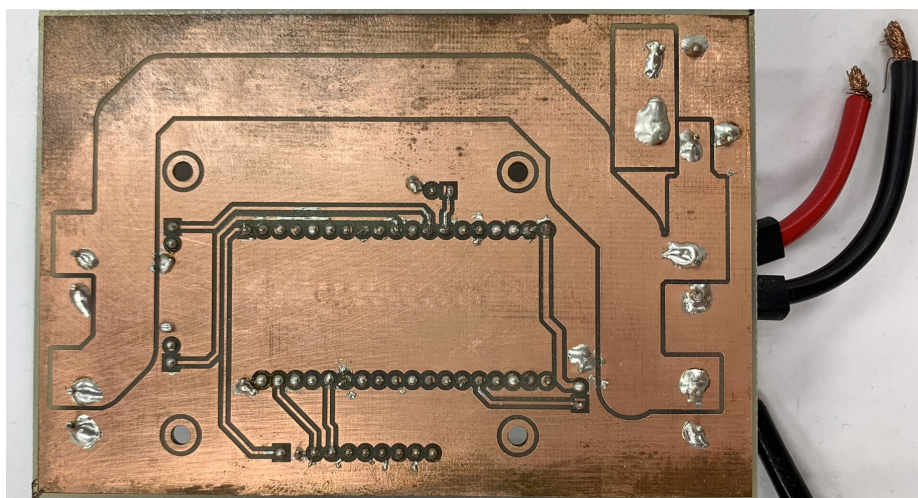


Figura 52: Vista inferior placa manufacturada.

Debido a que casi toda la parte inferior es cobre (polígono de GND), se presentaron ciertos problemas al realizar algunos puntos de soldadura. Esto ya que no se tenía la temperatura suficiente en el cautín y provocaba que el estaño no hiciera un correcto contacto entre el pin y la placa. Es por esta razón que en algunos pines fue necesario colocar una cantidad excesiva de estaño ya que en las primeras iteraciones de soldadura se presentaron algunos problemas de continuidad.



---

### Ecosistema Robotat

---

El cuadricóptero debe de tener la capacidad de integrarse al ecosistema Robotat, el cual está conformado por una plataforma instalada en el CIT 116 y un sistema de captura de movimiento OptiTrack con seis cámaras Prime X 41, tal y como se muestra en la Figura 53. La instalación y montaje del equipo OptiTrack fue realizado por los estudiantes José Pellecer, Rony Schumann y Camilo Perafan y descrito en el trabajo de graduación de este último [53].



Figura 53: Ecosistema Robotat CIT 116.

La implementación y fabricación del ecosistema Robotat se realizó en paralelo con este trabajo de graduación. Durante este proceso se noto que la obtención de datos leídos por el sistema de captura de movimiento no requería un grado de complejidad alto. Por lo tanto, la red de comunicación fue desarrollada en su totalidad por el estudiante Camilo Perafan, esto ya que era parte de los objetivos planteados en su trabajo de graduación. Debido a que el cuadricóptero corresponde a un agente del ecosistema, este trabajo de graduación se limitó a implementar y validar la librería para la red de comunicación respectivamente, la cual se explica de forma resumida en la sección 9.1.

## 9.1. Red de comunicación

La red de comunicación WiFi permite la transmisión bi-direccional de información entre los agentes del ecosistema Robotat y la computadora dedicada al uso del OptiTrack. Dicha red utiliza el router NETGEAR MBR624GU como punto de acceso, el cual opera en la banda de frecuencia 2.4 GHz, y una topología de bus. Esto último implica la existencia de un solo tópico en donde la computadora envía en un paquete de información la pose de todos los agentes del ecosistema. Para lograr que cada agente se ocupe únicamente de su propia información se utilizaron diferentes *ids*.

El desarrollo de la red esta dividida en dos partes: aplicación en Python y librería para el ESP32 en lenguaje de programación C. Cabe mencionar que, como se están utilizando dos lenguajes de programación diferentes, la red envía los datos en formato *string* y posteriormente en el ESP32 se decodifica este mensaje.

### 9.1.1. Aplicación Python

Tiene como objetivo principal tomar los datos proporcionados por el OptiTrack y enviarlos a cada agente del ecosistema por vía del protocolo MQTT. El código para esta aplicación cumple con diversas funciones, las cuales se describen brevemente a continuación:

- Valores de poses de cada agente del ecosistema proporcionados por el sistema de captura de movimiento.
- Conectarse como cliente de un servidor en el cual se reciben los mensajes publicados vía MQTT y donde los clientes suscritos pueden recibir los mensajes dependiendo del tópico.
- Diccionario de Python con los *ids* utilizados en una experimentación. Esto permite que dos agentes no pueden utilizar el mismo identificador.
- La publicación de información al tópico general se realiza inmediatamente después que se recibieran datos del sistema de captura de movimiento.

### 9.1.2. Librería Robotat

Esta librería permite la conexión del ESP32 al servidor a través del protocolo MQTT, la decodificación de datos provenientes de la aplicación de Python y las publicaciones/suscripciones a los tópicos activos durante las pruebas. Aquí se refleja una de las ventajas de utilizar el ESP32 como el microcontrolador por defecto para todos los agentes y aplicaciones del ecosistema Robotat. Esto último ya que este dispositivo tiene incorporado un módulo WiFi y el fabricante ya ofrece facilidades para la implementación de diferentes protocolos de comunicación, entre ellos MQTT.

Una parte vital es la decodificación de datos, la cual se realiza utilizando un vector que puede variar su tamaño en donde se sabe a priori las posiciones de los datos enviados. Al mismo tiempo la librería cuenta con una función encargada de separar los datos de la carga útil. Esta misma función revisa si el *id* recibido no está siendo usado, y si este es el caso, ningún otro agente puede tener el mismo identificador. El *id* se almacena en la memoria flash para convertirlo en un dato de tipo no volátil [53].

La librería cuenta con diferentes funciones, sin embargo, en el Cuadro 29 se describen únicamente las utilizadas para la implementación del cuadricóptero como agente del ecosistema Robotat.

Función	Parámetros	Descripción
<code>robotat_connect</code>	void	Realiza las configuraciones para que el ESP32 se conecte inicialmente a la red WiFi y posteriormente al servidor vía protocolo MQTT. Los parámetros de configuración se encuentran definidos por macros.
<code>robotat_get_data</code>	void	Se obtiene el vector de datos enviados vía MQTT.
<code>robotat_get_status</code>	void	Devuelve un valor numérico con el cual permite conocer el estado del agente dentro de la red de comunicación.

Cuadro 29: Funciones librería Robotat utilizadas en el cuadricóptero.

En el caso de la función `robotat_get_data` se obtiene la información de todos los agentes suscritos al tópico. Por lo tanto, es necesario separar los datos deseados según los índices específicos del *array*, los cuales ya son conocidos a priori.

## 9.2. Implementación del cuadricóptero al ecosistema Robotat

Considerando lo explicado en la sección 9.1, la implementación del cuadricóptero al ecosistema Robotat a través de la red de comunicación desarrollada por Camilo Perafán se ejemplifica en la Figura 54.

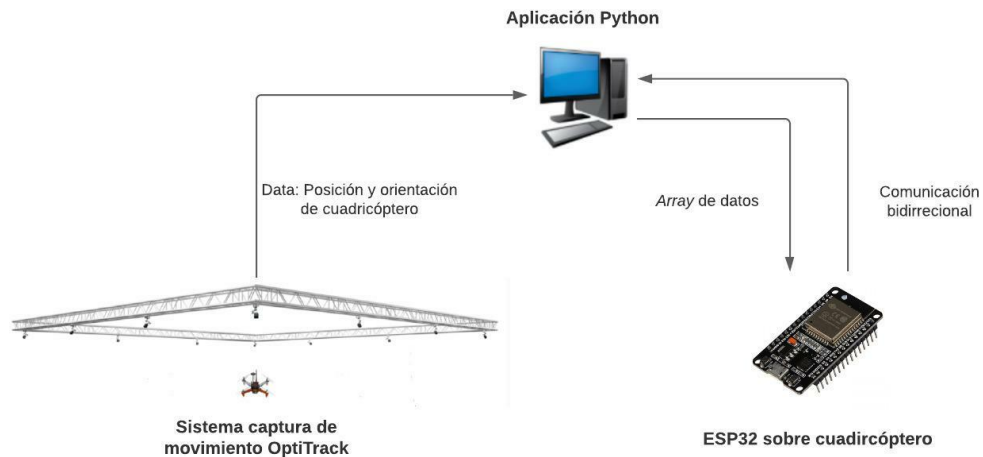


Figura 54: Interconexión entre el sistema de captura de movimiento OptiTrack y el cuadricóptero.

### 9.2.1. Colocación de marcadores

El sistema de captura de movimiento OptiTrack obtiene la posición y orientación de un objeto con ayuda de diferentes marcadores. Estos marcadores no deben de colocarse al azar ya que esto último provocaría una falta de precisión en los resultados obtenidos. En el caso del cuadricóptero estos marcadores se colocaron en las cuatro extremidades que unen la parte central con las hélices, tal y como se observa en la Figura 55.

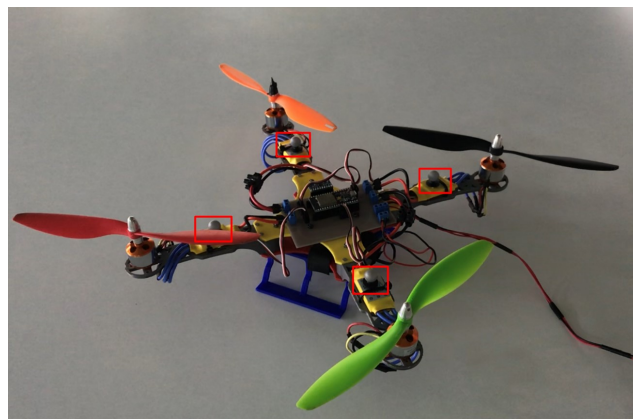


Figura 55: Cuadricóptero con marcadores para sistema OptiTrack.



Estos marcadores fueron colocados con cinta adhesiva de doble cara. Debido a que los marcadores están justo abajo de las hélices, se procedió a revisar desde la computadora del OptiTrack si se presentaba algún problema con esto último. Los resultados se muestran en las Figuras 56 y 57.

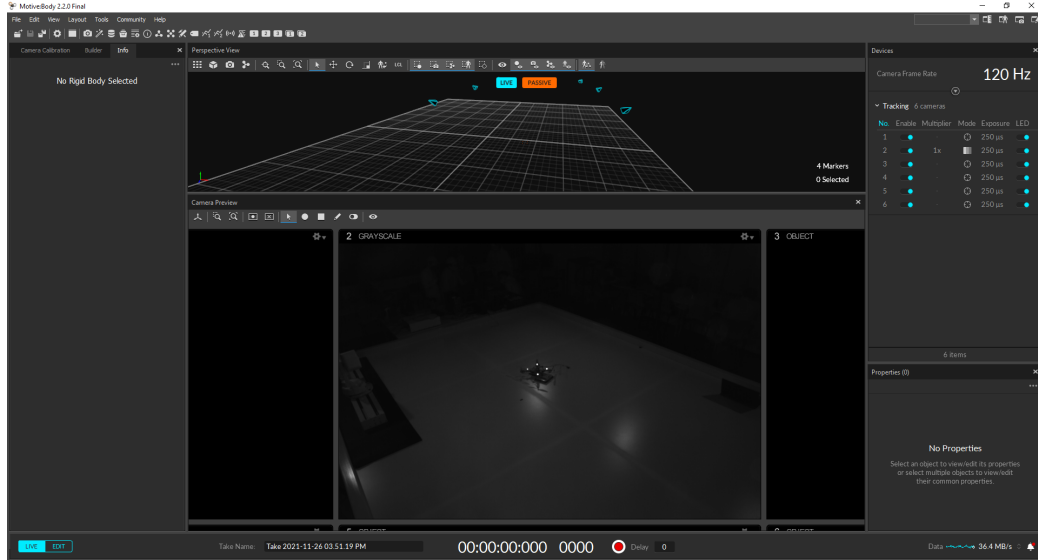


Figura 56: Primera prueba lectura de marcadores OptiTrack en cuadricóptero.

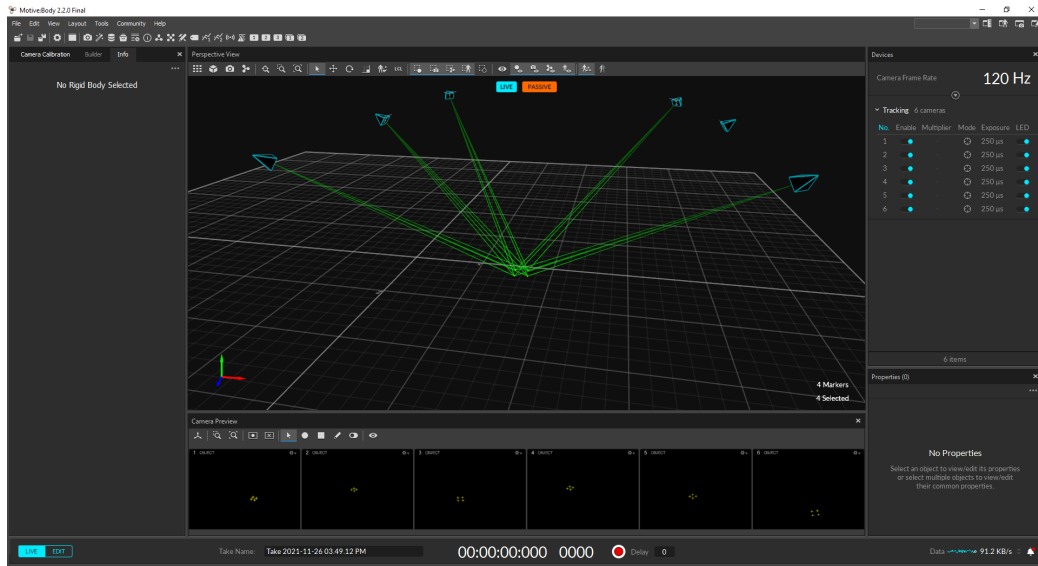


Figura 57: Segunda prueba lectura de marcadores OptiTrack en cuadricóptero.

Como se logra observar, la colocación de los marcadores en las posiciones correspondientes no presentan ningún problema para la lectura de datos del sistema de captura de movimiento.

### 9.2.2. Código

Una vez colocadas los marcadores en el cuadricóptero, el siguiente paso fue la implementación de las funciones de la librería Robotat en el *main.c* de la plataforma de vuelo. El código se divide en cuatro sección principalmente:

- Definiciones de macros y variables a utilizar en la lectura de datos e implementación del controlador.
- *void app\_main*. En esta sección se inicializa la comunicación I2C, filtro de Madgwick, motores, matrices para el controlador y la creación de un semáforo con sus respectivas *tasks*, las cuales una es para la lectura de ángulos desde la IMU y otra para ejecutar el controlador.
- *task\_LecturaAngulos*. Como su nombre lo indica, en esta *task* se realiza todo el proceso para obtener la orientación del cuadricóptero a través de los datos leídos desde la IMU.
- *task\_Controlador*. Utilizando los ángulos *roll*, *pitch* y *yaw*, se implementa el controlador desarrollado por Carlos Alonzo. Al mismo tiempo se ejecuta un arranque suave en los cuatro motores con la finalidad de que no se presente un pico de consumo de corriente cuando estos se enciendan.

El primer paso es configurar el módulo WiFi y el protocolo MQTT con el servidor, lo cual realiza la función `robotat_connect`. Dicha función debe ejecutarse únicamente una vez, por lo tanto se coloca en el *app\_main*.

Debido a que el controlador implementado únicamente permite que el cuadricóptero se estabilice a cierta altura, es decir, sin control de posición, los datos de interés del OptiTrack son los que determinan la orientación respectivamente. La sección de código mostrada a continuación fue colocada justamente en la *task* para la lectura de los ángulos.

```
1 //Obtencion de angulos OptiTrack
2 estado_wifi = robotat_get_status();
3 if(estado_wifi == 8){
4     data = robotat_get_data();
5     quat0 = data[4];
6     quat1 = data[5];
7     quat2 = data[6];
8     quat3 = data[7];
9     yaw_OT_deseado = data[8];
10    yaw_OT_deseado = yaw_OT_deseado*M_PI/180.0;
11    OptiTrackGetEulerAngles(quat0, quat1, quat2, quat3);
12    ROLL = ROLL*(180.0/M_PI);
13    PITCH = PITCH*(180.0/M_PI);
14    YAW = YAW*(180.0/M_PI);
15 }
16 }
```

Inicialmente se obtiene el estado del cuadricóptero dentro de la red de comunicación a través de la función `robotat_get_status()`. Si el valor que devuelve dicha función es igual a ocho, implica que todas las configuraciones e inicializaciones de módulos y protocolos han sido realizadas correctamente. Por lo tanto, es posible empezar a obtener el vector de datos

enviados por la aplicación en Python que almacena la información leída por el OptiTrack a través de la función `robotat_get_data()`. Para el caso del cuadricóptero, los datos enviados en este vector son los siguientes según cada posición del mismo:

- `data[0]` = Identificador del cuadricóptero en la red de comunicación.
- `data[1]` = Posición sobre el eje  $x$  del cuadricóptero dentro del ecosistema Robotat.
- `data[2]` = Posición sobre el eje  $y$  del cuadricóptero dentro del ecosistema Robotat.
- `data[3]` = Posición sobre el eje  $z$  del cuadricóptero dentro del ecosistema Robotat.
- `data[4]` = Primer componente del cuaternión que define la orientación del cuadricóptero.
- `data[5]` = Segundo componente del cuaternión que define la orientación del cuadricóptero.
- `data[6]` = Tercer componente del cuaternión que define la orientación del cuadricóptero.
- `data[7]` = Cuarto componente del cuaternión que define la orientación del cuadricóptero.

Como se explicó anteriormente, la separación de datos se realiza según las posiciones del *array*, las cuales son conocidas con anticipación. Es por ello que en esta ocasión se almacenaron en variables locales únicamente los componentes del cuaternión. Sin embargo, si más adelante se desea implementar un controlador que también considere la posición de la plataforma únicamente es de añadir en esta misma sección de código el almacenamiento de estos tres datos en variables locales.

Cabe mencionar que el *array* de datos permite enviar datos extras aparte de la posición y orientación de la plataforma de vuelo. Estos datos extras se encuentra a partir de la octava posición del vector de datos. Finalmente con la función `OptiTrackEulerAngles` se realiza la conversión de cuaterniones a ángulos de Euler con convención XYZ y normalizando dichos ángulos de 0 a  $2\pi$ . Por cuestiones de implementación del controlador, es necesario convertir los ángulos a radianes, lo cual se realiza en las últimas tres líneas del código mostrado anteriormente. La función `OptiTrackEulerAngles` es bastante similar a la que se encuentra en la librería del filtro de Madgwick ya que este también trabaja todo en cuaterniones para evitar singularidades matriciales. Para mayor información se puede revisar el repositorio en GitHub que se comparte en la sección de anexos.

### 9.2.3. Modelo experimental

Es necesario validar una correcta conexión y obtención de datos entre el sistema captura de movimiento OptiTrack y el cuadricóptero, independientemente que no se haya podido estabilizar a este último en el aire. Tomando esto en cuenta, la prueba realizada es la que se describe a continuación.

- **Modificación ángulo yaw.** En el controlador implementado se comparan los ángulos actuales que tiene la plataforma con los deseados por el usuario. Por lo tanto, la prueba consiste en encender la plataforma de vuelo con un ángulo  $yaw_{deseado}$  inicial y posteriormente, utilizando uno de los datos extras que se pueden enviar en el *array* desde Python, modificar dicho ángulo deseado y que esto se refleje físicamente en el cuadricóptero. La orientación se obtiene a través de la IMU y no con los datos provenientes del sistema de captura de movimiento.

### 9.2.4. Resultados

Durante esta prueba el ángulo  $yaw_{deseado}$  inicial fue de  $0^\circ$  y se modificó a un valor de  $90^\circ$  a través de la conexión con la aplicación en Python del ecosistema Robotat. Durante la ejecución de esta prueba el ángulo  $yaw = 0$  para la IMU es con el cuadricóptero en la orientación como se muestra en la Figura 58.

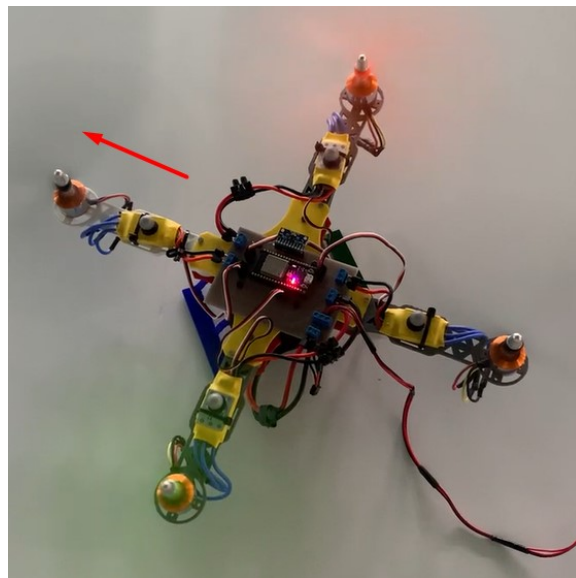


Figura 58: Cuadricóptero con orientación  $yaw = 0$ .

Ya que el segundo ángulo deseado es  $90^\circ$ , el cuadricóptero efectivamente roto en sentido contra horario por la regla de la mano derecha. Los resultados se muestran en las Figuras 59 y 60.

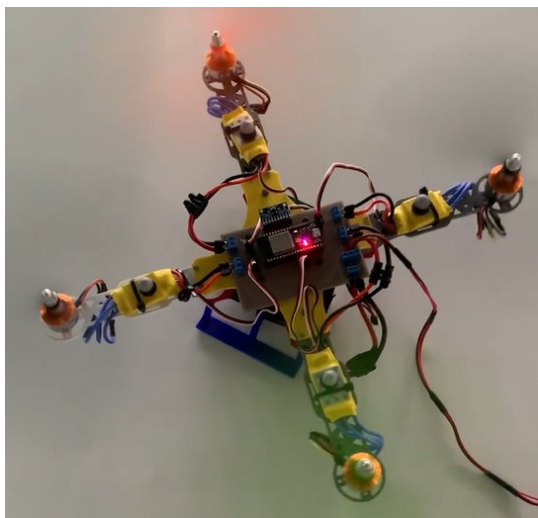


Figura 59: Cuadricóptero en rotación sobre ángulo *yaw*.

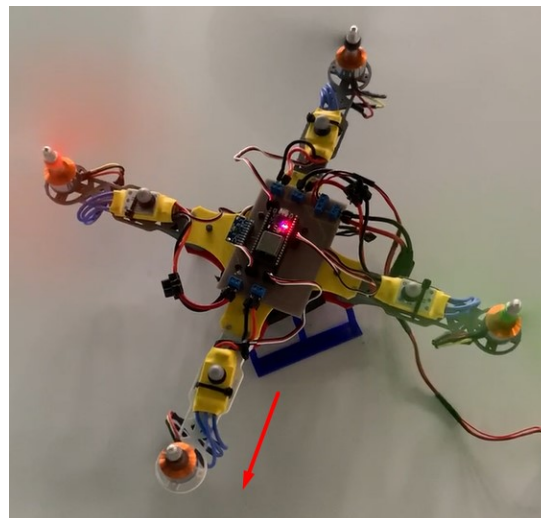


Figura 60: Cuadricóptero con orientación *yaw* = 90°.

Con esto se logra demostrar una correcta comunicación del cuadricóptero como agente del ecosistema Robotat. Uno de los principales problemas que surgieron durante esta prueba fue el GPIO 2 del ESP32. Durante el diseño de la placa se asignó este pin para la salida de la señal PWM del cuarto motor. Sin embargo, en la librería para la red de comunicación también se utiliza este pin para controlar el RGB LED con el que cuenta el ES32 para indicar el momento cuando la conexión ha sido finalizada. Es por ello que fue necesario deshabilitar este pin en la librería *robotat.h*.

Cabe mencionar que también se realizó una prueba adicional donde se deshabilitó la IMU y la orientación se obtuvo directamente con el sistema de captura de movimiento para observar el rendimiento de la plataforma con esta metodología. Los resultados permitieron observar que utilizando los datos del OptiTrack se tiene una mayor precisión en la orientación actual que tiene la plataforma. Sin embargo, el tiempo de reacción del cuadricóptero era mucho más lento. Esto debido a que la frecuencia de lectura del sistema de captura de movimiento es de 120 Hz, mientras que la IMU funciona a 1000 Hz, es decir, casi diez veces más rápido. Por lo tanto, sería recomendable utilizar un algoritmo de fusión de sensores para obtener mejores resultados en futuras experimentaciones.



- El código desarrollado para el control de los motores BLDC permitió además de inicializar correctamente los controladores ESC, generar la señal de PWM con el ancho de pulso requerido para su implementación. Esta librería es lo suficientemente general para su posterior uso con otro tipo de motores BLDC con ESC o controladores de vuelo.
- Mediante el proceso de selección *trade study*, el cual utilizó el método de ponderación AHP, se descartó el uso de baterías tipo Li-Po, NiMH y NiCD para la alimentación de la plataforma. La batería seleccionada de tipo Li-Ion fue la opción más ideal tomando en cuenta los siguientes criterios: peso, duración de vuelo, disponibilidad, corriente, cargador, voltaje, tiempo de vida y costo. La batería presenta un voltaje nominal de 12V, una capacidad de 6800 mAh y un peso de 287 gramos.
- Debido al consumo de corriente que presentan los motores a velocidades altas, se presenta un problema de apagado del cuadricóptero por auto protección de la batería tipo Li-Ion seleccionada.
- La correcta medición de los ángulos *roll*, *pitch* y *yaw* permitió validar una correcta comunicación y lectura de registros entre los módulos ESP32 y MPU-9250. Es necesario realizar ajustes en el cálculo del ángulo *yaw* para que esté presente un comportamiento más suave y continuo.
- Implementando la librería *robotat.h*, se logró integrar al cuadricóptero al ecosistema Robotat recibiendo datos provenientes de la red de comunicación de manera exitosa. A pesar de no utilizarse en la experimentación de este trabajo, se dejó listo el código para obtener información completa de la pose de la plataforma y así implementar controladores más sofisticados más adelante.





- Encontrar e implementar otro tipo de *switch* que permita su implementación directamente a la placa. Esto además de evitar el uso de una bornera de 3 pines, otorgará un resultado estéticamente más agradable y profesional.
- Conocer la carga disponible de la batería permite tener una noción del tiempo aproximado de vuelo con el que aún se cuenta al momento de realizar alguna prueba. Es por ello que se recomienda implementar en la placa un indicador para batería tipo Li-Ion con rango de lectura no menor a 12V.
- Cambiar los motores A2212/13T utilizados para la implementación de la plataforma. La descompensación que presentan provoca que el proceso de tuneo del controlador sea extremadamente complicado.
- Modificar el diseño de la placa de tal manera que esta tenga la forma poligonal de la parte central del cuadricóptero. Esto con la finalidad de aprovechar los agujeros con los que ya cuenta esta pieza y colocar ahí los tornillos de sujeción, evitando de esta manera debilitar estructuralmente la pieza. Adicionalmente ayuda a mejorar estéticamente la plataforma. Al mismo tiempo es necesario modificar el GPIO 2 para la señal PWM del motor 4. Esto último ya que este pin es utilizado por la librería *robotat.h* para el control del RGB LED incorporado en el ESP32.
- Implementar algún algoritmo de fusión de sensores entre el sistema de captura de movimiento y la unidad de medición inercial incorporada en la placa del cuadricóptero.
- Modificar la batería Li-Ion por una batería tipo Li-Po que tenga capacidades energéticas similares a la primera mencionada.



- [1] I. Tabachnik, *Drones*, ép. Informes Users. RedUsers, 2020. dirección: <https://books.google.com.gt/books?id=YdPKDwAAQBAJ>.
- [2] J. Brown, *Quadcopter flight controllers: The process behind every quadcopter*, <https://www.mydronelab.com/accessories/quadcopter-flight-controller.html>, Accessed: 2021-18-04, 2019.
- [3] C. Bellerjeau, K. Fitton, K. Krarti, K. Olson, J. Williams, A. Fendel, E. Gabrielle, N. Mororo, S. Said y J. Yanowitz, “Big Unmanned Box-wing Operation & Bigger Unmanned Box-wing Operation (BUBO BUBO) Conceptual Design Document,” University of Colorado, Department of Aerospace Engineering Sciences Senior Projects, Boulder, CO 80303, inf. téc., sep. de 2019.
- [4] J. Paulos y M. Yim, “Flight performance of a swashplateless micro air vehicle,” en *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, págs. 5284-5289.
- [5] Bitcraze, *Crazyflie 2.0*, <https://store.bitcraze.io/products/crazyflie-2-0>, Accessed: 2021-16-04, 2016.
- [6] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron y M. Egerstedt, “The robotarium: A remotely accessible swarm robotics research testbed,” en *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, págs. 1699-1706.
- [7] M. Búrbano, J. Morales, M. Sandoval, R. Saravia y D. Suazo, “MEGAPROYECTO SEEQ ROBOHELICÓPTERO,” Universidad del Valle de Guatemala, 18 Av. 11-95 zona 15, Vista Hermosa III. Guatemala, Guatemala, inf. téc., dic. de 2011.
- [8] J. Castañeda, “Diseño e implementación de una red de comunicación wifi e interfaz gráfica para una mesa de pruebas de robótica de enjambre,” Universidad del Valle de Guatemala, 18 Av. 11-95 zona 15, Vista Hermosa III. Guatemala, Guatemala, inf. téc., dic. de 2020.

- [9] C. Tedesco, *Ascensores electrónicos y Variadores de velocidad*. TECNIBOOK EDICIONES, ISBN: 9789505531899. dirección: <https://books.google.com.gt/books?id=38SWDwAAQBAJ>.
- [10] S. Arantes, *UF1880 - Gestión de redes telemáticas*. Editorial Elearning, S.L., 2016. dirección: <https://books.google.com.gt/books?id=V7hWDwAAQBAJ>.
- [11] P. Yedamale, “Brushless DC (BLDC) motor fundamentals,” *Microchip Technology Inc*, vol. 20, págs. 3-15, 2003.
- [12] Gonzalo Pérez y Maria Corres, *Control motor brushless sensorless*, [https://academic.e.unavarra.es/xmlui/bitstream/handle/2454/19222/TFG\\_Gonzalo%20Solchaga.pdf;jsessionid=991FCB58F6BB87771CA08131A46251AD?sequence=1](https://academic.e.unavarra.es/xmlui/bitstream/handle/2454/19222/TFG_Gonzalo%20Solchaga.pdf;jsessionid=991FCB58F6BB87771CA08131A46251AD?sequence=1), Accessed: 2021-05-11, 2015.
- [13] Steven Keeping, *Control de los motores BLDC sin sensores a través de la fuerza contra-electromotriz*, <https://www.digikey.com/es/articles/controlling-sensorless-bl-dc-motors-via-back-emf>, Accessed: 2021-05-11, 2013.
- [14] A. Gong y D. Verstraete, “Experimental testing of electronic speed controllers for UAVs,” en *53rd AIAA/SAE/ASEE Joint Propulsion Conference*, 2017, pág. 4955.
- [15] A. Gong, R. MacNeill y D. Verstraete, “Performance testing and modeling of a brushless dc motor, electronic speed controller and propeller for a small uav application,” en *2018 Joint Propulsion Conference*, 2018, pág. 4584.
- [16] L. E. Osborn, M. M. Iskarous y N. V. Thakor, “Sensing and control for prosthetic hands in clinical and research applications,” en *Wearable Robotics*, Elsevier, 2020, págs. 445-468.
- [17] L. von Stumberg, V. Usenko y D. Cremers, “A Review and Quantitative Evaluation of Direct Visual–Inertial Odometry,” *Multimodal Scene Understanding*, págs. 159-198, 2019.
- [18] H. Liu, *Robot Systems for Rail Transit Applications*. Elsevier, 2020.
- [19] InvenSense, *MPU-9250 Product Specification Revision 1.1*, <https://invensense.tdk.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf>, Accessed: 2021-05-20, 2016.
- [20] M. Serra, *Topografía para minería subterránea*, ép. Aula Politècnica. Edicions de la UPC, S.L., 2002, ISBN: 9788483015919. dirección: <https://books.google.com.gt/books?id=ktfCgsvIOUAC>.
- [21] M. Looney, “A simple calibration for MEMS gyroscopes,” *EDN (Electrical Design News)*, vol. 55, n.º 9, pág. 21, 2010.
- [22] L. E. Ardila Pérez y J. S. Eslava Garzón, “Diseño, calibración y prueba de una unidad de medida inercial (IMU) inalámbrica para cohetes,” *Ingenium Revista de la facultad de ingeniería*, vol. 13, n.º 25, págs. 36-43, 2012. DOI: 10.21500/01247492.1305. dirección: <https://revistas.usb.edu.co/index.php/Ingenium/article/view/1305>.
- [23] M. Cabanas, *Técnicas para el mantenimiento diagnóstico de máquinas eléctricas rotativas*, ép. ACCESO RÁPIDO. Marcombo, 1998, ISBN: 9788426711663. dirección: <https://books.google.com.gt/books?id=X3p4bZfoqgEC>.
- [24] *MEMS Accelerometers*, <https://www.siliconsensing.com/technology/mems-accelerometers/>, (Accessed on 11/25/2021).

- [25] L. Ramirez, G. Jiménez y J. Carreño, *Sensores y Actuadores*, ép. Libro Catedra. Grupo Editorial Patria, 2014, ISBN: 9786074389364. dirección: <https://books.google.com.gt/books?id=wMm3BgAAQBAJ>.
- [26] *Magnetometer Calibration*, <http://www.camelsoftware.com/2016/03/13/imu-maths-calculate-orientation-pt3/>, (Accessed on 11/28/2021).
- [27] A. Maier, A. Sharp e Y. Vagapov, “Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things,” en *2017 Internet Technologies and Applications (ITA)*, IEEE, 2017, págs. 143-148.
- [28] JAMECO ELECTRONICS, *Electronics Fundamentals: Voltage Regulator*, <https://www.jameco.com/Jameco/workshop/learning-center/voltage-regulator.html>, Accessed: 2021-05-12, 2014.
- [29] Electronics Hub, *Types of Voltage Regulators*, <https://www.electronicshub.org/types-of-voltage-regulators/>, Accessed: 2021-05-12, 2019.
- [30] Monolithic Power Systems, *Voltage Regulator Types and Working Principles*, <https://www.monolithicpower.com/en/voltage-regulator-types>, Accessed: 2021-05-12, 2020.
- [31] J. de Kuyper, *Principios y aplicaciones de la energía fotovoltaica y de las baterías*. Ediciones UC, 2018, ISBN: 9789561423565. dirección: <https://books.google.com.gt/books?id=EmqPDwAAQBAJ>.
- [32] F. A. P. Jabib, “MANUAL DE BATERÍAS Y ACUMULADORES,”
- [33] PROMETEC, *Lo que hay que saber para elegir una batería LIPO*, <https://www.prometec.net/elegir-bateria-lipo/>, (Accessed on 11/30/2021).
- [34] M. Álvarez-Láinez, H. Martínez-Tejada y F. Isaza, *Nanotecnología: Fundamentos y aplicaciones*. Universidad de Antioquia, 2020, ISBN: 9789587148985. dirección: <https://books.google.com.gt/books?id=0RbeDwAAQBAJ>.
- [35] Schneider Electric, *Power Budget*, <https://ecostruxure-building-help.se.com/bms/topics/show.castle?id=6562&locale=en-US&productversion=1.9>, Accessed: 2021-05-12, 2017.
- [36] Paula Baker and James Whalen, *SURVEY OF TRADE STUDY METHODS FOR PRACTICAL DECISION-MAKING*, [https://www.nasa.gov/sites/default/files/survey\\_of\\_trade\\_study\\_methods\\_-\\_baker.pdf](https://www.nasa.gov/sites/default/files/survey_of_trade_study_methods_-_baker.pdf), Accessed: 2021-05-15, 2012.
- [37] José Lugo, *Pugh Method: How to decide between different designs?* <https://sites.nd.edu/jlugo/2012/09/24/pugh-method-how-to-decide-between-different-designs/>, Accessed: 2021-05-15, 2012.
- [38] R. V. Vargas y P. IPMA-B, “Using the analytic hierarchy process (AHP) to select and prioritize projects in a portfolio,” en *PMI global congress*, vol. 32, 2010, págs. 1-22.
- [39] Kepner-Tregoe, *Problem Solving and Decision Making Workshop*, <https://www.kepner-tregoe.com/lp/problem-solving-and-decision-making-workshop/#:~:text=Kepner%2DTregoe%20Decision%20Analysis%20is,benefits%2C%20and%20making%20the%20decision.>, Accessed: 2021-05-15, 2017.

- [40] H. Rodríguez y C. Chérigo, “Evaluación de Algoritmos de Fusión de Datos de Medición Inercial para Vehículos Aéreos no Tripulados,” en *Global Partnerships for Development and Engineering Education: Proceedings of the 15th LACCEI International Multi-Conference for Engineering, Education and Technology, July 19-21, 2017, Boca Raton, FL, United States*, Latin American y Caribbean Consortium of Engineering Institutions, 2017, pág. 455.
- [41] S. Mehta, *Electronic Product Design*. S Chand & Company Limited, 2011, ISBN: 9788121936156. dirección: <https://books.google.com.gt/books?id=iqAbEAAAQBAJ>.
- [42] PlatformIO, *Professional collaborative platform for embedded development*, <https://platformio.org/>, Accessed: 2021-05-18, 2020.
- [43] J. Carballar y J. Falcón, *WI-FI. Lo que se necesita conocer*. RC Libros, 2010, ISBN: 9788493776909. dirección: <https://books.google.com.gt/books?id=rQmH6IKyvigC>.
- [44] TST, *MQTT*, <http://www.tst-sistemas.es/mqtt/>., Accessed: 2021-05-18, 2015.
- [45] F. Janabi-Sharifi y W. Melek, *Advances in Motion Sensing and Control for Robotic Applications: Selected Papers from the Symposium on Mechatronics, Robotics, and Control (SMRC'18)- CSME International Congress 2018, May 27-30, 2018 Toronto, Canada*, ép. Lecture Notes in Mechanical Engineering. Springer International Publishing, 2019, ISBN: 9783030173692. dirección: <https://books.google.com.gt/books?id=hZmdDwAAQBAJ>.
- [46] OptiTrack, *About OptiTrack*, <https://optitrack.com/about/>, Accessed: 2021-05-18, 2021.
- [47] *Fluke 381*, [https://dam-assets.fluke.com/s3fs-public/381\\_\\_\\_\\_\\_umspa0000.pdf](https://dam-assets.fluke.com/s3fs-public/381_____umspa0000.pdf), (Accessed on 09/30/2021).
- [48] *2231A-30-3*, [https://download.tek.com/datasheet/2231A-30-3\\_Datasheet\\_1KW-61078-0.pdf](https://download.tek.com/datasheet/2231A-30-3_Datasheet_1KW-61078-0.pdf), (Accessed on 09/30/2021).
- [49] MechaniCalc, *Trade Study*, <https://mechanicalcalc.com/reference/trade-study>, (Accessed on 09/26/2021).
- [50] *30A BLDC ESC Product Manual*, [https://www.optimusdigital.ro/index.php?controller=attachment&id\\_attachment=451](https://www.optimusdigital.ro/index.php?controller=attachment&id_attachment=451), (Accessed on 09/25/2021), 2012.
- [51] C. DRON, *Baterías de Drones*, <https://www.controldron.com/baterias-de-drones-2/>, (Accessed on 09/26/2021).
- [52] N. Smith, *Calculadora de ancho de vía ANSI PCB*, <https://www.desmith.net/NMds/Electronics/TraceWidth.html>, (Accessed on 09/24/2021), 2021.
- [53] C. Perafan, “Robotat: un ecosistema robótico de captura de movimiento y comunicación inalámbrica,” Universidad del Valle de Guatemala, 18 Av. 11-95 zona 15, Vista Hermosa III. Guatemala, Guatemala, inf. téc., dic. de 2021.

## 13.1. Repositorio GitHub

El código desarrollado y verificado se encuentra disponible en el *branch 17022 - Dev* correspondiente al repositorio del ecosistema Robotat: <https://github.com/mezea-uvg/robotat/tree/master>

## 13.2. Componentes cuadricóptero

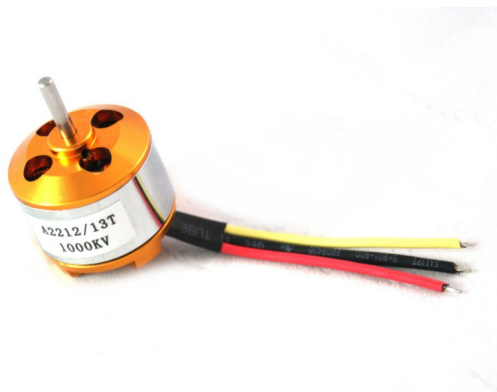


Figura 61: Motor BLDC A2212/13T 1000kv.



Figura 62: 30A BLDC ESC.

### 13.2.1. PCB

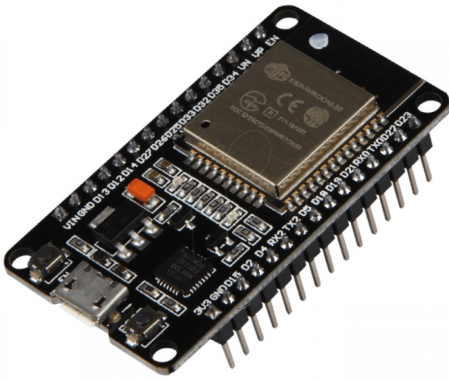


Figura 63: Módulo Wi-Fi + Bluetooth ESP32.

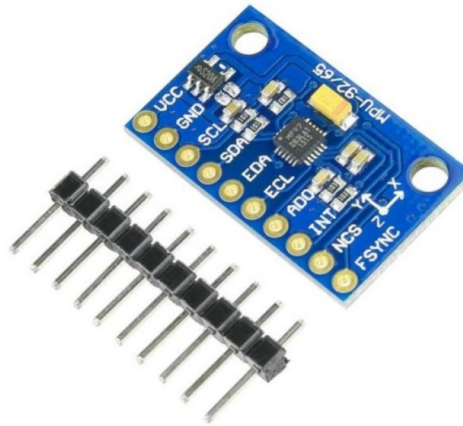


Figura 64: Módulo MPU-9250.



Figura 65: Bornera de 3 pines, 300V a 15A.



Figura 66: Bornera de 2 pines, 300V a 15A.



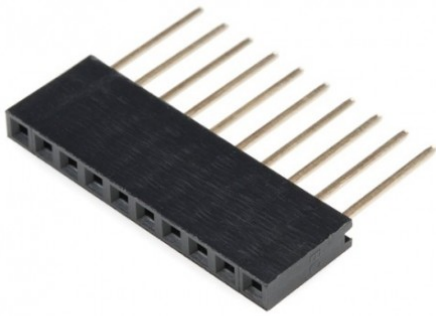


Figura 67: *Header* tipo hembra.

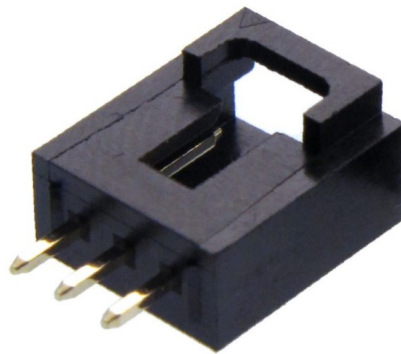


Figura 68: Conector macho a placa de 3 pines.



Figura 69: *Switch* 3 pines ON/OFF, 20A 12DC.

