



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Industrial

Diseño e impresión 3D de un modelo de dron para
competición

Trabajo Fin de Grado

Grado en Ingeniería en Tecnologías Industriales

AUTOR/A: Zanón Sáiz, Raúl

Tutor/a: Baraza Calvo, Juan Carlos

Cotutor/a: Gracia Morán, Joaquín

CURSO ACADÉMICO: 2022/2023

AGRADECIMIENTOS

“A mis padres, por no rendirse conmigo

A mi novia, Lea, por su incansable apoyo

A mi hermano, por prácticamente todo”

RESUMEN

El presente Trabajo de Fin de Grado (TFG) consiste en el diseño de un dron de carreras, abordado desde la multidisciplinariedad de competencias que convergen en la producción de un prototipo funcional. El proyecto comprende desde la fase de estudio preliminar hasta la construcción y puesta en marcha del prototipo. Es imprescindible obtener un orden de magnitud a fin de establecer las necesidades a cumplir, así como el diseño detallado del dispositivo a fabricar.

Para desarrollar los contenidos, se profundiza en aspectos del detalle mediante la investigación y la consulta a profesionales con el fin de completar y/o complementar los conocimientos adquiridos. Además, el aprendizaje y manejo de SolidWorks como herramienta de diseño y ensayo, y Arduino como plataforma de desarrollo para el sistema de control, son competencias clave para la ejecución del proyecto.

En el documento se recoge el diseño y la fabricación del chasis de la aeronave mediante impresión 3D. También el montaje, el diseño de la electrónica y automática, el cableado y las pruebas necesarias para comprobar su correcto funcionamiento.

El prototipo es un cuadricóptero que cumple la normativa europea de drones de categoría abierta, subcategoría A3 sin marcado de clase, ya que se trata de una aeronave no tripulada (UAS, del inglés *Unnamed Aircraft System*) de construcción privada.

Palabras Clave: Dron, Impresión 3D, SolidWorks, Arduino, UAS, Aeronave, Cuadricóptero.

ABSTRACT

This Final Degree Project (TFG) covers the design of a racing drone, addressed from multidisciplinary of various competences that converge on the production of a functional prototype. The project includes from the preliminary study phase to the construction and commissioning of the prototype. The role of carrying out the preliminary project is essential to understand project requirements, as well as the detailed design of the device to be manufactured.

In order to carry out the contents, aspects of detail are delved into and consultation with professionals in pursuance of complete and/or complement the knowledge acquired. In addition, learning and operating SolidWorks as a design and testing tool, and Arduino as a development platform for the control system, are key skills for the execution of the project.

The document includes the design and manufacture of the chassis of the aircraft using 3D printing. Also, the assembly, the wiring and flight tests to check it functions properly.

The resulting prototype is a quadcopter that complies European laws concerning open category drones, subcategory A3 without type labelling, since it is a private construction Unmanned Aircraft System (UAS); the final product belongs to this same classification.

Keywords: Drone, 3D Print, SolidWorks, Arduino, UAS, Aircraft, Quadcopter.

RESUM

El present Treball de Fi de Grau (TFG) consisteix en el diseny d'un dron de carreres, abordat des de la multidisciplinarietat de competències que convergeixen en la producció d'un prototipus funcional. El projecte comprén des de la fase d'estudi preliminar fins a la construcció i posada en marxa del prototip. És imprescindible obtindre un ordre de magnitud a fi d'establir les necessitats a complir, així com el disseny detallat del dispositiu a fabricar.

Per a desenvolupar els continguts s'aprofundeix en aspectes del detall mitjançant la investigació i la consulta a professionals amb la finalitat de completar i/o complementar els coneixements adquirits. A més, l'aprenentatge i maneig de SolidWorks com a ferramenta de disseny i assaig, i Arduino com a plataforma de desenvolupament per al sistema de control, són competències clau per a l'execució del projecte.

En el document es recollix el disseny i la fabricació del xassís de l'aeronau mitjançant impressió 3D. També el muntatge, el cablejat i les proves necessàries per a comprovar el seu correcte funcionament.

El prototipus és un quadricopter que compleix la normativa europea de drons de categoria oberta, subcategoria A3 sense marcat de clase, ja que es tracta de un *Unmanned Aircraft System* (UAS) de construcció privada; el producte final també s'emmarca en aquesta categoria.

Paraules Clau: Dron, Impresió 3D, SolidWorks, Arduino, UAS, Aeronau, Quadricopter.

ÍNDICE

DOCUMENTOS CONTENIDOS EN EL TFG

- Memoria
- Presupuesto
- Planos
- Anejos

ÍNDICE DE LA MEMORIA

CAPÍTULO 1. INTRODUCCIÓN.....	1
1.1. Motivación	1
1.2. Justificación.....	1
1.3. Objetivo.....	2
1.4. Normas y referencias	3
CAPÍTULO 2. ESTUDIO PRELIMINAR.....	4
2.1. Principio de funcionamiento de un cuadricóptero	4
2.2. Introducción a la fabricación digital.....	5
2.3. Plan de Resolución	6
2.4. Análisis de productos similares.....	7
2.5. Orden de magnitud	9
2.5.1. Hélices	9
2.5.2. Motores.....	10
2.5.3. Configuración de los motores	11
2.5.4. Control de la aeronave.....	11
2.5.5. Control del movimiento	12
2.5.6. Sistema de visionado.....	13
2.5.7. Control remoto.....	13
2.5.8. Alimentación	14

CAPÍTULO 3. SELECCIÓN DE COMPONENTES.....	15
3.1. Aeronave.....	15
3.1.1. Controlador	15
3.1.2. Sensor de presión.....	17
3.1.3. Unidad de medición inercial.....	18
3.1.4. Transceptor de radiofrecuencia	19
3.1.5. Cámara	20
3.1.6. Motores y hélices	21
3.1.7. Batería	23
3.1.8. Circuito propuesto.....	25
3.2. Mando	26
3.2.1. Controlador	27
3.2.2. <i>Joysticks</i>	27
3.2.3. Circuito propuesto.....	28
CAPÍTULO 4. DISEÑO ESTRUCTURAL.....	30
4.1. Chasis de la aeronave.....	30
4.1.1. Estudio del arrastre y la sustentación	30
4.1.2. Velocidad terminal de la aeronave	35
4.1.3. Flotación.....	40
4.1.4. Material	40
4.2. Carcasa del mando.....	43
CAPÍTULO 5. FABRICACIÓN	44
5.1. Impresión 3D.....	44
5.1.1. Aeronave	45
5.1.2. Mando	48
5.2. Calibrado de las hélices.....	50
5.3. Electrónica.....	51
5.4. Montaje.....	52
CAPÍTULO 6. CONCLUSIONES.....	54
6.1. Adversidades del proyecto.....	54
6.1.1. Prototipo alfa	54
6.1.2. Incompatibilidad y errores en el <i>software</i> de diseño	54
6.1.3. Inviabilidad de la unidad de control seleccionada	55

6.2. Revisión de diseño	56
6.2.1. Cambios en la selección de componentes	58
6.2.2. Previsión del <i>software</i>	61
6.2.3. Ajuste del control de actitud.....	63
6.3. Reflexión personal	63
BIBLIOGRAFÍA.....	65

ÍNDICE DE FIGURAS

Figura 1.....	4
Figura 2.....	5
Figura 3.....	7
Figura 4.....	11
Figura 5.....	17
Figura 6.....	18
Figura 7.....	19
Figura 8.....	20
Figura 9.....	21
Figura 10.....	25
Figura 11.....	26
Figura 12.....	27
Figura 13.....	28
Figura 14.....	28
Figura 15.....	29
Figura 16.....	30
Figura 17.....	31
Figura 18.....	32
Figura 19.....	32
Figura 20.....	33
Figura 21.....	34
Figura 22.....	34
Figura 23.....	35
Figura 24.....	36

Figura 25.....	36
Figura 26.....	37
Figura 27.....	37
Figura 28.....	38
Figura 29.....	38
Figura 30.....	39
Figura 31.....	39
Figura 32.....	40
Figura 33.....	42
Figura 34.....	42
Figura 35.....	43
Figura 36.....	45
Figura 37.....	46
Figura 38.....	47
Figura 39.....	49
Figura 40.....	49
Figura 41.....	50
Figura 42.....	51
Figura 43.....	52
Figura 44.....	53
Figura 45.....	53
Figura 46.....	55
Figura 47.....	57
Figura 48.....	58
Figura 49.....	59

ÍNDICE DE TABLAS

Tabla 1.....	8
Tabla 2.....	16
Tabla 3.....	22
Tabla 4.....	24
Tabla 5.....	41

Tabla 6.....	46
Tabla 7.....	48
Tabla 8.....	56
Tabla 9.....	60
Tabla 10.....	61

ÍNDICE DE ECUACIONES

Ecuación 1.....	10
Ecuación 2.....	20
Ecuación 3.....	23
Ecuación 4.....	23
Ecuación 5.....	24
Ecuación 6.....	31
Ecuación 7.....	31
Ecuación 8.....	59

ÍNDICE DEL PRESUPUESTO

1. Cuadro de precios unitarios.....	1
2. Cuadro de precios número 1.....	2
3. Cuadro de precios número 2.....	6
4. Precio total del proyecto.....	11

MEMORIA

DISEÑO E IMPRESIÓN 3D DE UN MODELO DE
DRON PARA COMPETICIÓN



CAPÍTULO 1. INTRODUCCIÓN

1.1. MOTIVACIÓN

La elección del tema del presente Trabajo de Fin de Grado (TFG) responde al interés que este tiene desde el punto de vista industrial. Las aeronaves no tripuladas (UAS, del inglés *Unnamed Aircraft System*) son objeto de aplicación en cada vez más sectores (como la agricultura, la logística o el salvamento, entre otros) (CORTEVA, s.f.) (MECALUX, 2020) (Ministerio de Transportes, Movilidad y Agenda Urbana [Mitma], 2021); es de esperar que, en el futuro, su uso se vea incrementado como consecuencia de la investigación y el desarrollo de esta tecnología. Además, los drones ofrecen una alternativa de menor impacto medioambiental en lo que al transporte de mercancías y mensajería se refiere, así como una potente herramienta para combatir la contaminación (Magán, 2017).

Aunque todas las aplicaciones citadas anteriormente presentan cierto atractivo, la falta de maduración de una amplia legislación, y las limitaciones actuales por coste, carga y autonomía, dificultan el desarrollo de esta tecnología en el ámbito industrial. Aun así, su uso como elemento lúdico está muy extendido. Enfocar el proyecto en este aspecto ofrece la posibilidad de familiarizarse con el creciente sector aeronáutico, encaminado a un porvenir en el que la fabricación de drones tome una posición de mayor importancia.

A nivel académico, el desarrollo de este TFG pretende acercarse a la labor profesional de la ingeniería. Este aspecto permite ahondar en los conocimientos aprendidos desde un punto de vista académico, siempre imprescindible para el constante crecimiento del ingeniero, y en las vicisitudes de la vida laboral que comprenden la aplicación de la norma, la relación con personas externas, la eficiencia, etc.

Por último, personalmente he de destacar que me supone fascinante tener la oportunidad de poder aplicar los conceptos trabajados y asimilar nuevos a partir de la realización de este proyecto.

1.2. JUSTIFICACIÓN

La pluralidad de materias impartidas en el Grado en Ingeniería en Tecnologías Industriales ha permitido desarrollar ciertas competencias en las bases de los distintos ámbitos de la ingeniería. Durante la trayectoria académica, en lo que respecta al avance del aprendizaje, ha sido de gran importancia el dominio de nociones previas que no necesariamente tienen origen en asignaturas explícitamente relacionadas. De la misma forma que en las materias cursadas es necesario comprender de manera interdisciplinar los conocimientos propios a cada una de ellas, un proyecto de categoría industrial compromete diferentes ramas de la ingeniería a fin de obtener un producto acabado que cumpla con las especificaciones deseadas.

A la hora de plantear el orden de magnitud del proyecto, es especialmente relevante entender las distintas facetas que componen el problema; para hallar las soluciones individuales, se realiza un anteproyecto. Una vez comprendidas las soluciones a nivel de subsistemas, se realiza el proyecto haciéndolas converger en una solución definida. Este TFG consta de todas estas características, y culmina evaluando la viabilidad del proyecto mediante la producción de un prototipo.

Se plantea de esta manera el diseño de un dron cuadricóptero que implica numerosas asignaturas cursadas en el grado; que, en mayor o menor medida, son de aplicación para el proyecto. Al adentrarse en dicha aplicación, no solo se refuerzan los conceptos, sino que se profundiza en los mismos.

Por otra parte, el uso recreativo de drones ha experimentado un notable crecimiento en el sector del ocio: las ventas a particulares han aumentado en los últimos años, y ya existen muchas personas que enfocan el arte de volar como afición (Aguado, 2017). En lo referente a la cuestión lúdico-deportiva, existe una creciente propuesta de eventos de libre implicación, además de aquellos de participación profesional. La mayoría de los pilotos que compiten a nivel profesional, y algunos aficionados, lo hacen pilotando obras de su propia creación.

1.3. OBJETIVO

El fin último de este TFG es el diseño y puesta en marcha de una aeronave no tripulada controlada de forma remota. Para que el proyecto sea viable, se han de cumplir ciertos requisitos impuestos por la Agencia Estatal de Seguridad Aérea (AESA). La normativa establece distintas categorías según las especificaciones del producto y la función para la que son creados: el caso de este proyecto se adscribe a la categoría abierta por su objetivo último: el ocio; concretamente, sin marcado de clase, debido a sus características de construcción.

La ley establece con carácter general para las aeronaves que, si estas pierden la comunicación con el piloto, como mínimo han de aterrizar de manera segura para las personas.

El alcance del proyecto no contempla la comercialización real del producto; más bien se trata de una simulación de un proyecto real. Es por esto que el documento de la memoria no incluirá información al respecto de temas como distribución, trato con proveedores, etc.

En cuanto a las especificaciones de los distintos eventos de competición, los UAS se agrupan por categorías según sus características. Dependiendo de la organización, se exigen determinados requerimientos, siendo común la ausencia de los mismos; a pesar de todo, distintas configuraciones dan lugar a diferentes características de vuelo. Este proyecto se enfoca en las tipologías *Long Range* por su tamaño, más acordes para principiantes.

Por otra parte, al tratarse de un dron de carreras, la destreza en el manejo toma un rol de gran importancia; observar el movimiento en primera persona permite al piloto cumplir con mayor facilidad. Será necesario, por tanto, un sistema de visionado acorde a esta función. Obviamente, será imprescindible un método de comunicación entre el dispositivo de la aeronave y el de control, tanto para retransmitir la imagen como para operar.

El producto final, por su cometido, ha de presentar una cierta resistencia en el núcleo en caso de esfuerzos imprevistos en maniobra regular. Si bien el diseño de una carrocería queda fuera de los objetivos del proyecto, sí se estudia el comportamiento del chasis ante esfuerzos aerodinámicos.

1.4. NORMAS Y REFERENCIAS

Según la propia AESA, “La categoría operacional ‘abierta’ cubre todas las operaciones con UAS que sean de bajo riesgo operacional, como marca la normativa europea, y no requieren de ninguna autorización operacional ni de una declaración por parte del operador del UAS previo al vuelo.” (Agencia Estatal de Seguridad Aérea [AESA], s.f.). Dicha normativa se recoge en el artículo 4 del Reglamento de Ejecución (UE) 2019/947 (2019), que fija como condiciones para el dron un peso máximo de 25kg y una altitud máxima de vuelo de 120m desde el punto de despegue.

Su subcategoría no aporta mayor restricción, si bien fija ciertos requerimientos para el uso. En el caso de este proyecto, se ha optado por la subcategoría A3 (que permite operaciones en áreas alejadas en un mínimo de 150 metros de zonas residenciales, recreacionales, industriales o comerciales), con objeto de garantizar la seguridad de las personas. Con todo ello, y tratándose de una aeronave de construcción privada, no es necesario tener en cuenta el marcado de clase; sin embargo, el piloto deberá completar un curso *online* y superar un examen teórico, de igual manera que como se establece para operar con UAS con marcado de clase C1. Para pertenecer a esta subcategoría, además, la nave no debe disponer de modos de control automático excepto para la estabilización o para la asistencia en caso de pérdida del enlace.

Por último, la ley establece mediante el Cuadro Nacional de Atribución de Frecuencias (Orden ETD/666/2020, 2020), las bandas en las que está permitido emitir, así como la potencia máxima con la que se permite hacerlo. El caso del presente proyecto se enmarca en las subbandas de 430 a 440MHz, de 2.4GHz y/o de 5.8GHz; todas ellas reservadas para aplicaciones industriales, científicas y médicas.

CAPÍTULO 2. ESTUDIO PRELIMINAR

2.1. PRINCIPIO DE FUNCIONAMIENTO DE UN CUADRICÓPTERO

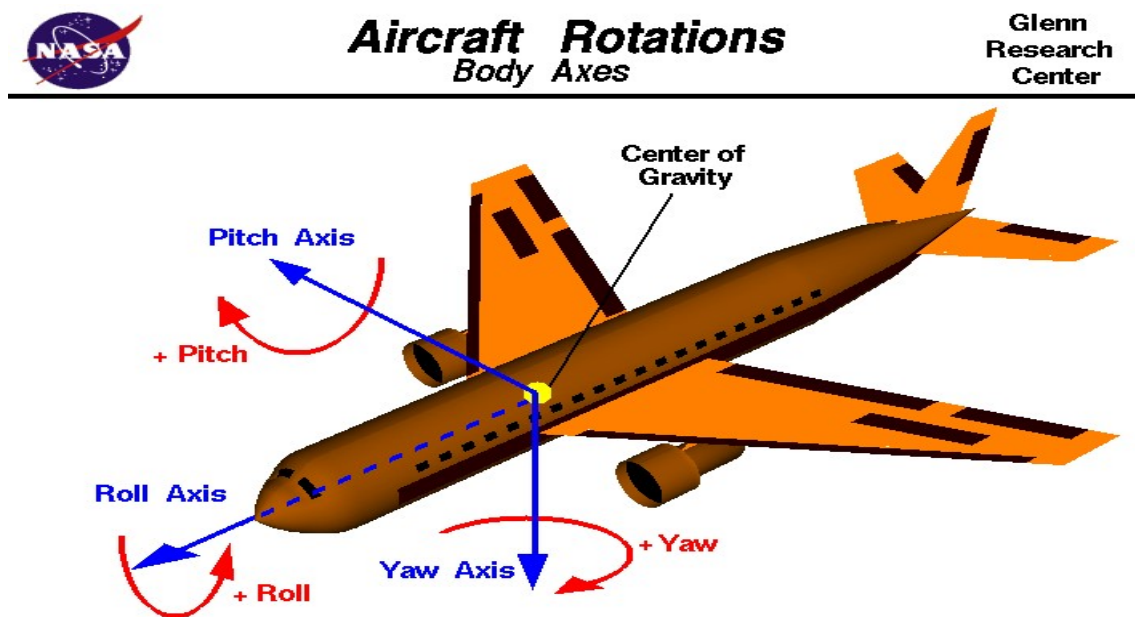
Con el propósito de comprender los distintos mecanismos que influyen en el funcionamiento normal de un dron cuadricóptero, se describen en esta parte de la memoria algunos conceptos básicos.

Las aeronaves disponen de seis grados de libertad; para denotarlos, es común recurrir a los ejes de navegación mostrados en la Figura 1. De esta manera, cada eje tiene asociado un giro:

- El eje vertical tiene su origen en el centro de gravedad y está dirigido hacia la parte inferior de la aeronave. Al giro en este eje se le denomina guiñada, o *yaw* (ψ).
- El eje longitudinal tiene su origen en el centro de gravedad y se dirige hacia delante, paralelo al fuselaje de la aeronave. El giro en este eje recibe el nombre de alabeo, o *roll* (ϕ).
- El eje lateral o transversal tiene su origen en el centro de gravedad y se dirige hacia la derecha respecto al eje longitudinal. Al giro en este eje se le llama cabeceo, o *pitch* (θ).

Figura 1

Ejes de navegación.

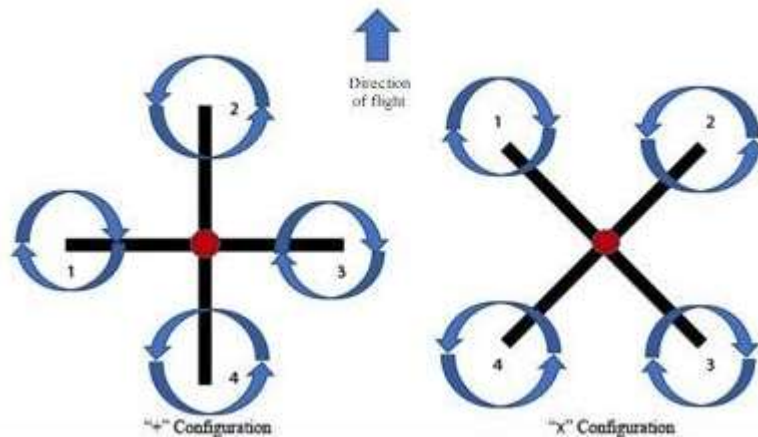


Nota. Adaptado de Aircraft Rotations, por National Aeronautics and Space Administration (NASA), <https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/aircraft-rotations/#>.

En cuanto a los cuadricópteros se refiere, las dos configuraciones más extendidas son en “+” y en “x”; si bien existen variaciones de las mismas, estas no difieren en cuanto a principios de funcionamiento con las citadas. En la Figura 2 se pueden apreciar ambas.

Figura 2

Configuración “+” y “x”.



Nota. Adaptado de “+” and “x” configuration for the quadcopter, de Nur Salma Mohd Mokhtar, 2020, Modelling and PID control system integration for quadcopter DJI F450 attitude stabilization.

El movimiento se produce por variaciones en la velocidad de giro de los motores, de manera que, cuando todos los motores aumentan o disminuyen su velocidad en igual magnitud, se produce el movimiento vertical (o *throttle*). Si los motores, consecutivos, modifican su velocidad por pares, el dron pivota sobre uno de los ejes de navegación; al modificarse el ángulo de cabeceo o alabeo, el dron se mueve en el plano horizontal. Por último, cuando los motores modifican su rotación de forma alterna, los momentos generados por las hélices horarias difieren de los de las antihorarias, produciéndose la rotación en el eje de guiñada. Cabe destacar que dependiendo del tipo de configuración, los motores implicados en un mismo movimiento difieren, siendo lo anteriormente expresado exclusivo de la configuración en “x”.

2.2. INTRODUCCIÓN A LA FABRICACIÓN DIGITAL

Cuando se habla de fabricación digital, se hace referencia al “proceso controlado por un computador que da forma a la materia a partir de un archivo digital” (Universidad Politécnica de Valencia [UPV], s.f., p. 5). Dentro de este grupo se encuentran dos estrategias de manufacturación, sustractiva y aditiva, que requieren de *Computer Numerical Control* (CNC) y, por tanto, de archivos digitales que puedan comprender las máquinas pertinentes.

La fabricación aditiva hace referencia al conjunto de tecnologías que permiten la fabricación de objetos de manera secuencial, habitualmente capa a capa. Se define como aditiva porque el material se añade sucesivamente, obteniéndose un objeto tridimensional. Según la Organización Internacional de Normalización, en la norma ISO/ASTM 52900-2022 (2022), se establecen siete categorías de procesos aditivos:

- Proyección de aglutinante: un agente líquido aglutinante se deposita de forma selectiva para unir materiales en polvo.
- Deposición de energía focalizada: utiliza energía térmica focalizada para unir materiales mediante fusión, a medida que se depositan.
- Extrusión de material: el material se dispensa de forma selectiva a través de una boquilla o un orificio.
- Proyección de material: se depositan de forma selectiva gotas de materia prima.
- Fusión de lecho de polvo: la energía térmica funde de forma selectiva ciertas zonas de un lecho de material en polvo.
- Laminado de hojas: el material en forma de láminas se une para formar una pieza.
- Fotopolimerización en tanque o cuba: un polímero fotosensible en estado líquido se cura de forma selectiva en una cuba al irradiarlo con una fuente de luz.

En el caso del proyecto se emplea la extrusión de material como método de fabricación. El robot cartesiano que ejecutará el proceso es una impresora de tecnología *Fused Filament Fabrication* (FFF), que imprime las capas del objeto mediante la extrusión de material fundido; el movimiento del extrusor o la plataforma de soporte es facilitado por el robot.

Esta tecnología ha ganado mucha importancia en los últimos años debido a las posibilidades de prototipado que ofrece; permitiendo materializar los diseños de manera rápida y barata en comparación con las técnicas sustractivas o formativas. Puesto que la fabricación aditiva permite el uso de materiales biodegradables, se estudia en primera instancia el uso de los mismos a fin de que el proyecto guarde cierto compromiso medioambiental; en cualquier caso, ya que los plásticos que se emplean en la impresión son termoplásticos, como mínimo serán reciclables.

2.3. PLAN DE RESOLUCIÓN

Una vez establecido el objeto del proyecto y algunos conceptos básicos, se definirá el camino a seguir para alcanzar la solución final. El proceso de resolución parte de un análisis de productos similares al que se pretende diseñar; mediante este método se trata de conocer de manera más cercana la realidad del objeto a proyectar, concretamente las funciones y elementos que lo componen. A partir de esta información, se plantea un primer orden de magnitud, donde se toman algunas decisiones de carácter general y se presenta información sobre las configuraciones y las tecnologías empleadas.

A partir de este punto, se comienza a profundizar en la solución al detalle. En primer lugar, se escoge la placa controladora y los sensores necesarios a fin de establecer el circuito electrónico, tanto para el caso de la aeronave como para el mando. Adicionalmente, se deciden los motores a emplear mediante una primera aproximación del objeto.

Con todo ello, se escogen sendas baterías, de manera que se asegure un tiempo de funcionamiento mínimo, basado en el análisis de productos similares. A partir del modelo de circuito se crean los diseños de las Placas de Circuito Impreso (PCB por sus siglas en inglés) necesarias para conectar los componentes entre sí. Dichos diseños determinarán las dimensiones finales de la aeronave.

Partiendo de un diseño inicial y las dimensiones consideradas, en base a las decisiones tomadas, así como de los requisitos impuestos por la electrónica, se diseña iterativamente una solución de mejores prestaciones aerodinámicas. El objetivo es que la aeronave sea capaz de alcanzar una velocidad entre los 120 y 150km/h, tal y como lo hacen los drones en el ámbito deportivo (Bayerische Motoren Werke AG [BMW], 2019). Se comprobará, además, que las alas del prototipo no flecten excesivamente, ya que esto podría comprometer la estabilidad.

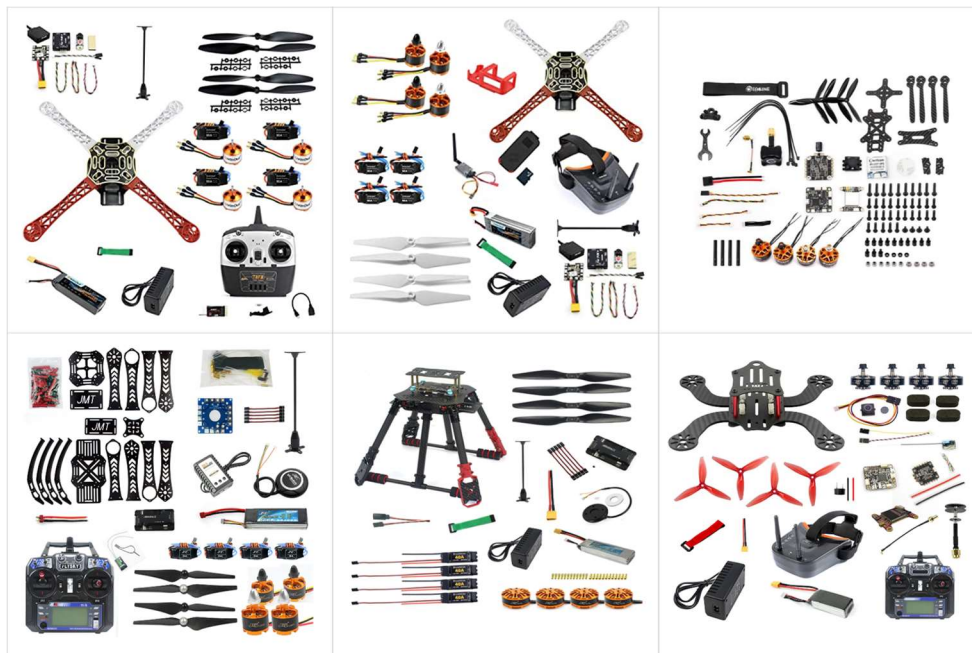
Finalmente, una vez comprobados los componentes de manera individual, se monta el prototipo y se realizan las pruebas pertinentes. Por último, se realizan las pruebas de vuelo, tanto con el dron sujeto para estudiar las funciones de transferencia y desarrollar un control en consecuencia, como para demostrar su viabilidad.

2.4. ANÁLISIS DE PRODUCTOS SIMILARES

Los modelos seleccionados para el estudio son: QWinOut F450, QWinOut F450-V2, Tyro119, QWinOut RTF X4M360L, Dron X4 plegable, Mini F4 Pro; que se corresponden con las imágenes mostradas en la Figura 3. Existen otros kits similares que no se han incluido debido a que no tienen un nombre como conjunto comercial, sino que se venden como un lote de piezas sin especificar.

Figura 3

Kits comerciales de cuadricópteros.



Nota. Fuente propia. Elaborada a partir de las imágenes ofrecidas por las empresas distribuidoras.

Así pues, se ha elaborado la Tabla 1, que incluye tanto especificaciones técnicas como elementos y funciones, a partir de la cual se evalúan las diferentes características. Cabe destacar que en este apartado no se busca una solución, sino orientación para plantear un orden de magnitud. Los vínculos a la web de los distribuidores se recogen en Amazon (s.f. a), Amazon (s.f. b), AliExpress (s.f. a), AliExpress (s.f. b), AliExpress (s.f. c) y AliExpress (s.f. d), respectivamente.

Tabla 1

Matriz de características de los kits de drones analizados.

Especificaciones	QWinOut F450	QWinOut F450-V2	Tyro119
Marco	Modular	Modular	Bloque
Motores	<i>Brushless</i> 1000KV	<i>Brushless</i> 920KV	<i>Brushless</i> 1800KV
Hélices	2 palas	2 palas	3 palas
Imax Electronic Speed Controler	30A	30A	40A
Control remoto	Mando. GFSK	Mando. GFSK	Receptor para la aeronave.
Acelerómetro/Giroscopio	MPU6500	MPU6500	MPU6000
Barómetro	LPS22HB	LPS22HB	BMP280
GPS	Brújula QMC5883L	Radiolink TS100	BN-220T
Controlador	Mini PIX	Mini PIX	BetaFlight F4
Cámara	No incluido	FIREFLY Q6 Action	Caddx.us Turbo f2
Sistema de visionado	No incluido	LST-009 FPV Goggles	No incluido
Distancia entre motores opuestos	450mm	450mm	250mm
Batería	11,1V 2200mAh LiPo	11,1V 4400mAh LiPo	No incluido
Tiempo de vuelo	8-12 min	10 min	No especificado
Precio	219,87	395,44	399,57

Especificaciones	QWinOut RTF X4M360L	Dron X4 plegable	Mini F4 Pro
Marco	Modular	Modular y plegable	Bloque
Motores	<i>Brushless</i> 920KV	<i>Brushless</i> 700KV	<i>Brushless</i> 2400KV
Hélices	2 palas	2 palas	3 palas
Imax Electronic Speed Controler	30A	40A	30A
Control remoto	Mando. GFSK	Mando. GFSK	Mando. GFSK
Acelerómetro/Giroscopio	MPU6000	MPU6000	MPU6000
Barómetro	MS5611	MS5611	BMP280
GPS	M6	LEA-6H	No incluido
Controlador	APM 2.8	APM 2.8	BetaFlight F4
Cámara	No incluido	No incluido	700TVL
Sistema de visionado	No incluido	No incluido	Mini FPV 3 pulgadas
Distancia entre motores opuestos	450mm	300mm	200mm
Batería	11,1V 3300mAh LiPo	11,1V 4400mAh LiPo	11,1 V 1500mAh LiPo
Tiempo de vuelo	No especificado	No especificado	10 min
Precio	433,6	406,49	459,37

Los motores incluidos en cada kit varían en el rango de 700-2400KV (rpm por voltio aplicado). Sin embargo, es necesario hacer hincapié en que aquellos que presentan mayores velocidades de giro son los que emplean hélices de tres palas. En el apartado 2.5.1 se estudia la diferencia entre ambas opciones. Los variadores *Electronic Speed Controller* (ESC) incluidos trabajan con intensidades máximas en el rango 30-40A. Es importante que dicho amperaje sea suficiente para soportar las necesidades del motor; por lo general, así como en los casos presentados, se sobredimensiona para evitar problemas.

Respecto al marco, no es de extrañar que en su mayoría no sean plegables, ya que esto implica dificultades para transmitir los esfuerzos. Cabe destacar que la relación entre la distancia entre motores opuestos y el radio de las hélices afecta al control de vuelo; para una misma hélice, mayor separación implica más estabilidad, pero menor velocidad de respuesta ante maniobras (Arslan y Ozbek, 2018). Todos los ejemplos analizados presentan dimensiones típicas de drones *Long Range*, a excepción del Mini F4 Pro.

Los kits incluyen, en su mayoría, un mando para comunicar al piloto con la aeronave; dicho mando emplea modulación *Gaussian Frequency Shift Keying* (GFSK) como método de comunicación. Respecto a la transmisión de imagen, solo dos de ellos ofrecen el conjunto de cámara y sistema de visionado. Uno de ellos incluye una cámara con transmisor, pero ningún útil para reproducir.

En referencia a la electrónica, las placas controladoras que se emplean son la Mini PIX, la BetaFlight F4 y la APM 2.8. Algunas de ellas integran sensores de la aeronave, mientras otras requieren de cableado adicional. Los sensores más importantes para el control son el acelerómetro/giroscopio y el barómetro, puesto que de ellos depende la estabilidad y el cumplimiento de la norma. En lo que respecta al primero se observa el uso de Unidades de Medición Inercial (IMU), concretamente las de la serie MPU; cabe destacar que la MPU6000 es de gama superior.

En cuanto a los barómetros, no presentan diferencias significativas, aunque el MS5611 es el de gama más alta. Se observa también el uso de *Global Positioning System* (GPS) en muchos modelos; uno solo incluye una brújula, y otro no ofrece este elemento. El GPS es útil en caso de pérdida de enlace, si se opta por un sistema de retorno.

Con todo ello, el conjunto se alimenta mediante baterías de Polímero de Litio (LiPo) de 3 celdas, de 3.7V de voltaje nominal cada una. La capacidad varía entre 1500mAh y 4400mAh; mientras que las relaciones de descarga no superan los 30C, que es una forma de expresar el amperaje máximo que es capaz de suministrar la batería en función de su capacidad. Los tiempos de vuelo no siempre son facilitados, pero se estiman en alrededor de 10 minutos.

2.5. ORDEN DE MAGNITUD

El paso previo a la materialización del proyecto es el estudio de las posibilidades en los distintos niveles que componen el dron. La finalidad de establecer un orden de magnitud es plantear los requerimientos de las facetas que afronta este proyecto, de manera que sea posible establecer los objetivos secundarios y aproximar así el principal a la solución.

2.5.1. Hélices

El número de palas es un factor importante a considerar en primer lugar. No solo orienta en la función final que ha de cumplir el dron, sino que también determina, en parte, la potencia de los motores y la eficiencia de la propulsión. A continuación, se presentan algunas definiciones necesarias para comprender esta relación.

- Carga de disco: es la *ratio* entre el peso y la superficie que generan las hélices del rotor en movimiento.
- Paso de hélice: distancia axial que recorre la hélice de manera ideal por cada revolución.
- *Ratio* de avance: cociente entre la velocidad del fluido a través de la hélice (si, de manera ideal, no le afectarían las palas) y la velocidad de giro en la punta exterior de la pala.

En lo referente al número de palas, como se ha comentado en el apartado 2.4, predominan en el mercado las hélices de 2 y 3. Dada la Ecuación (1) de la sustentación, y teniendo en cuenta que el área de interacción entre la hélice y el viento es mayor para las hélices de 3 palas, es posible confirmar que, para una misma velocidad de giro, las hélices de 3 palas ofrecen un empuje mayor, que resulta también en mayor estabilidad. Sin embargo, presentan una eficiencia menor para cualquier carga de disco (Theys et al. 2019) y para relaciones de avance menores al 85% (Rajeevalochanam et al. 2016).

$$F_L = \frac{1}{2} \cdot C_L \cdot \rho \cdot A \cdot V^2 \quad (1)$$

Por otro lado, el paso de hélice está directamente relacionado con el empuje, y determina la velocidad del aire que atraviesa la hélice. Mayores pasos de hélice conllevan mayor velocidad del aire trasegado, y mayor velocidad terminal; aunque esto también implica mayor turbulencia y consumo de la batería. Por otra parte, pasos de hélice bajos permiten una mayor aceleración.

2.5.2. Motores

Escoger unos motores apropiados es una decisión importante para el proyecto, ya que de ello dependerán muchos aspectos del movimiento de la nave. El hecho de que los kits analizados en el apartado 2.4 contaran con motores paso a paso, *brushless*, con funcionamiento en alterna, no sorprende si se tienen en cuenta las grandes ventajas que ofrecen en cuanto a control de velocidad. Si bien los servomotores también presentan esta característica, han sido descartados por su mayor coste.

Los motores *brushless* reciben su nombre de la propia ausencia de escobillas. En ellos, la conmutación se realiza de manera electrónica. Al no existir este rozamiento, la vida útil y la eficiencia del motor son mayores; y, puesto que la conmutación es electrónica, la respuesta dinámica es mayor. Esto último es de interés si se tiene en cuenta que se precisa de respuestas rápidas tanto para el manejo como para el control de estabilidad.

2.5.3. Configuración de los motores

Las composiciones más extendidas son de 3 o 4 grupos motor. La razón por la que se ha decidido no contemplar la posibilidad de usar 3 motores es la rotación en el eje *yaw*, que en este tipo de drones hay que compensar continuamente. Puesto que los motores son impares, uno de ellos tiene que compensar la torsión provocada por los otros dos; esto lleva a un control más exhaustivo por parte del controlador.

En cuanto a los drones de 4 motores, como ya se comentó en el apartado 2.1, existen dos configuraciones principales. La configuración en “+” presenta diversos problemas en lo que se refiere al manejo, principalmente debido a su hándicap en lo referente a la actitud. Al rotar en el eje de *pitch* o de *roll*, la configuración en “x” ofrece la intervención de todos los motores, por lo que el giro sobre los ejes presenta mayor torsión (Niemic y Gandhi, 2016); a pesar de que, en la otra, al existir mayor distancia desde los motores a los ejes, se genere mayor torsión por motor. Además, en la configuración en “+”, al rotar en dichos ejes se produce un momento en el eje *yaw* que ha de ser compensado (Niemic y Gandhi, 2016). Por todo ello, se ha decidido emplear la configuración en “x”.

Por otra parte, la dirección de los motores también admite dos variantes: si el sentido es hacia arriba (esto es, las hélices quedan por encima de los motores, como en la Figura 4.b), se habla de configuración *puller*; si es hacia abajo (como en la figura 4.a), se denomina *pusher*. Aunque la mayor parte de los drones emplea la configuración “*puller*”, la “*pusher*” presenta una eficiencia mayor para cualquier carga de disco (Theys et al. 2019); por lo que para el presente trabajo se ha optado por esta.

Figura 4

Configuraciones “*pusher*” y “*puller*”.



Nota. Adaptado de *Experimental Investigation of Tractor and Pusher Hexacopter Performance*, de Prashin Sharma y Ella Atkins, 2019, *Journal of Aircraft*, <https://arc.aiaa.org/doi/abs/10.2514/1.C035319?journalCode=ja>.

2.5.4. Control de la aeronave

En lo que se refiere al control de la aeronave, existen controladoras comerciales diseñadas específicamente para este fin, tal y como se ha visto en el apartado 2.4. Si bien es cierto que presentan buenas prestaciones, como se verá a continuación, el carácter académico del TFG inclina la decisión por el diseño del circuito de control.

Si el proyecto no tuviera un carácter académico, el uso de una controladora de vuelo comercial sería una decisión acertada. Principalmente, por los programas que las respectivas empresas proveedoras ofrecen para configurar las placas al gusto del cliente. Pero también porque integran algunos de los sensores necesarios para el control, de manera que se reduce el volumen y peso en la aeronave.

Por otro lado, el precio de estos componentes es relativamente alto y, dado que este proyecto contempla el prototipado, se opta por el diseño del circuito de control mediante lenguaje C++, concretamente empleando el entorno de desarrollo de Arduino. Si se tiene en cuenta que la placa APM 2.8 presentada en el punto 2.4 está basada en este lenguaje, se partirá de la hipótesis de que esta tecnología tiene potencial como solución.

Arduino es un lenguaje de programación basado en Wiring (escrito en C++). Wiring es un *framework* (una plantilla o estructura base para la organización y el desarrollo de *software*) de código abierto muy extendido para la programación de microcontroladores. Si bien este lenguaje fue creado para las placas electrónicas que comercializaba la propia empresa (también llamadas Arduino), su carácter de código abierto ha propulsado una gran cantidad de alternativas en este lenguaje. Así pues, no es de extrañar el uso de diferentes placas frente a distintos requerimientos.

2.5.5. Control del movimiento

Tal y como se explica en el apartado 2.1, la velocidad de la aeronave se obtiene a partir de las fuerzas generadas por la rotación de las hélices y su dirección, dada por el giro en los ejes de navegación. Para traducir la rotación en las hélices a movimiento del sistema se requiere de un modelo matemático complejo.

En dicho modelo, una vez linealizado, las variables manipuladas serían las velocidades de rotación de cada motor; las variables controladas serían las velocidades de giro en los ejes de navegación, los ángulos que forma el sistema no inercial con su homólogo inercial, y la altitud respecto al punto de despegue. Ya que la variación de cualquier velocidad de giro de las hélices provoca una respuesta en el movimiento, el conjunto de variables considerado implica la necesidad de un sistema de control *Multiple Input Multiple Output* (MIMO).

Puesto que el tiempo que contempla la elaboración del TFG es limitado, y dado que se ha otorgado mayor importancia a otros aspectos del trabajo, se emplea una simplificación de lo expuesto en el párrafo anterior. De esta manera, se trabaja con cuatro sistemas *Single Input Single Output* (SISO): la altitud de la aeronave se controla con un cambio en la velocidad de rotación de todos los motores al mismo tiempo; los movimientos horizontales mediante una reducción en la velocidad de los motores frontales al movimiento y un aumento proporcional en los otros dos; el giro en la guiñada como un aumento en la velocidad de dos motores en contraposición y una reducción proporcional en los otros dos.

El primero de los sistemas descritos y el último de ellos se puede regular con un controlador PID; para el caso de los otros dos se emplea un control en cascada de PID. Se emplea un control *override* para que, en caso de exceder la altitud máxima, la aeronave descienda. Para obtener el modelo de cada uno de estos sistemas independientes, se utiliza la herramienta System Identification Toolbox de Matlab. También se usa este programa para desarrollar los controladores; para su implementación, existen bibliotecas de Arduino específicas.

Para transmitir la información sobre la velocidad de giro a los motores, se emplean los ESC. Estos variadores transforman los pulsos de Modulación por Ancho de Pulsos (PWM, por sus siglas en inglés) en una corriente trifásica que genera en el motor una velocidad de giro proporcional al pulso recibido.

2.5.6. Sistema de visionado

Se plantean dos opciones principales: transmitir las imágenes a una pantalla o a unas gafas de realidad virtual. Mientras que la segunda opción ofrece un mejor control a altas velocidades, también plantea un coste mucho mayor; puesto que a nivel de implantación no supone un cambio significativo optar por una u otra, se ha decidido continuar con la primera opción empleando la pantalla de un teléfono móvil como plataforma de reproducción.

De nuevo, se plantea una nueva decisión sobre qué tecnología emplear para la transmisión de imagen; en cualquier caso, este sistema será independiente del de control. De lo contrario, el envío de imagen podría perjudicar el correcto funcionamiento, colapsando los canales de comunicación. La opción habitual dentro de la drónica es el uso de WiFi a 5.8GHz, por la baja latencia; aunque es verdad que existen modelos que trabajan a 2.4GHz. Teniendo en cuenta el entorno de operación establecido en el apartado 1.4, que limita las interferencias de posibles routers, se emplea la frecuencia de 2.4GHz por motivos de presupuesto. En cualquier caso, esto no es una solución aceptable dada la necesidad de respuesta rápida; por este motivo, en el apartado 3.1.5 se incluye no solo la solución adoptada para el prototipo, sino también una que garantice la viabilidad.

2.5.7. Control remoto

Es necesario establecer una forma en que comunicarse con la aeronave. Las distintas propuestas analizadas en el apartado 2.4 incluyen, en su mayoría, un mando comercial. Al analizar métodos alternativos, se observó que la tecnología de radio *Long Range* (LoRa) ofrece resultados a nivel de distancia muy superiores a las prestaciones de estos mandos; sin embargo, ofrece una velocidad de comunicación demasiado baja para las necesidades planteadas.

Cabe decir que algunos drones que se enmarcan como *Long Range* disponen de controles remotos capaces de transmitir a distancias similares a las que opera la tecnología LoRa. Sin embargo, puesto que la transmisión de la señal de video a 5.8GHz no suele alcanzar longitudes mayores de 500m, se ha decidido optar por la modulación GFSK; en el apartado 2.4 se puede apreciar que es una solución extendida.

También se planteó emplear modulación *Frequency Shift Keying*, puesto que requiere de menor potencia de procesador; pero, dada la poca cantidad de información que se envía, y las mejores prestaciones de la modulación GFSK, principalmente en cuanto a la limpieza de la señal se refiere, finalmente se ha decidido emplear esta última.

Por razones de reducción de coste en el prototipado, y con intención de analizar la viabilidad de crear un mando propio, se decide emplear la tecnología mencionada. Se plantea, además, la producción de un mando que, aunque simple, permite evaluar la eficacia de esta propuesta.

Con todo ello, son necesarios: *joysticks*, un controlador para el mando y su correspondiente programación, y el diseño de una carcasa funcional impresa en 3D. Dicha carcasa dispondrá de un módulo para sujetar el teléfono del cliente, lo que sería inviable en caso de emplear un mando comercial.

2.5.8. Alimentación

Evidentemente, se debe encontrar un mecanismo para alimentar los distintos componentes de la aeronave. Puesto que el peso es un factor importante, es de interés encontrar una buena relación del mismo respecto a la capacidad de energía de la batería; también se ha de tener en cuenta que será necesaria una capacidad de descarga de esta al menos igual a la máxima de los ESC. La batería, además, ha de proporcionar el voltaje suficiente como para alimentar los motores y la placa controladora.

En referencia al aeromodelismo a radiocontrol, las baterías de LiPo y las de Níquel Hidruro Metálico (NiMH) son las más extendidas. De entre estas opciones, se ha decidido emplear las de LiPo por la diferencia de potencial nominal de 3.7V que ofrecen por celda, frente a los 1.2V de las NiMH. El principal inconveniente de este tipo de baterías es que son muy sensibles tanto a las sobrecargas como a las sobre descargas; es por ello que se debe incluir un sistema que monitorice el estado de la misma. En caso de reducirse el voltaje a una cantidad menor a la establecida, la aeronave debe realizar un aterrizaje seguro y entrar en un modo de operación de mínimo consumo.

Puesto que la velocidad de giro de los motores depende del voltaje suministrado, a fin de obtener un rango de operación más elevado, se ha decidido emplear una batería de 3 celdas para la aeronave. Por otra parte, siendo que el mando no presenta estos requerimientos, se ha optado por emplear una de 2 celdas.

CAPÍTULO 3. SELECCIÓN DE COMPONENTES

En este capítulo se describen los distintos componentes electrónicos que forman parte de cada uno de los dos subsistemas en que se divide el sistema objeto del presente TFG: la aeronave y el mando.

Para cada uno de ellos, adicionalmente, se especifica el circuito electrónico que contiene estos mismos elementos a fin de tomar decisiones más adelante, tanto en este mismo capítulo como en posteriores. Además, en cada subapartado se detallan los motivos por los cuales se ha decidido emplear el componente correspondiente y se ofrece información del mismo en base a su catálogo.

3.1. AERONAVE

Para la selección de los componentes de la aeronave se ha decidido comenzar por aquellos que comprometen la selección de otros. De esta manera, se elige en primer lugar el microcontrolador y después los sensores; el conjunto hélice-motor se considera en base al peso de los componentes ya elegidos y, una vez establecido esto, se procede a la selección de una batería adecuada.

3.1.1. Controlador

Dada la decisión de programar el controlador de vuelo de la aeronave, la primera decisión a tomar es qué placa emplear. Dicha placa ha de cumplir con algunos requisitos mínimos:

1. Dimensiones aceptables para la magnitud del proyecto.
2. Conteo de pines coherente a las necesidades.
3. Memoria *flash* suficiente para almacenar el código.
4. Potencia de procesado capaz de gestionar los distintos sensores y actuadores en tiempos que no causen problemas de estabilidad.

Aplicando el primer requisito, las opciones decaen; por ello, de entre las diferentes opciones en el mercado, se consideran tres microcontroladores: ATmega328, SAMD21 y SAMD51. Las placas en que se integran son la Arduino Nano, la SAMD21 M0-mini y la Teensy 4.1 respectivamente. La Tabla 2 muestra una comparativa de sus especificaciones. Los precios se han obtenido desde la misma web de compra (Amazon) para facilitar la comparación, ya que existen numerosas empresas que venden estos productos por precios muy diferentes; es por ello que puede haber discrepancia con el precio final reflejado en el presupuesto.

Tabla 2

Comparación de los posibles microcontroladores candidatos.

Especificación	Teensy 4.1	SAMD21 M0-Mini	Arduino Nano
Arquitectura	ARM Cortex M7	ARM Cortex M0+	AVR
Frecuencia del reloj	600MHz	48MHz	20MHz
RAM	1024kB	32kB	2kB
Memoria <i>flash</i>	2048kB	256kB	32kB
PWM (pines y resolución)	34 pines de 9bits	12 pines de 10bits	6 pines de 8bits
Dimensiones	61x20mm	54x20mm	45x18mm
Precio	54.95€	23.95€	21.60€

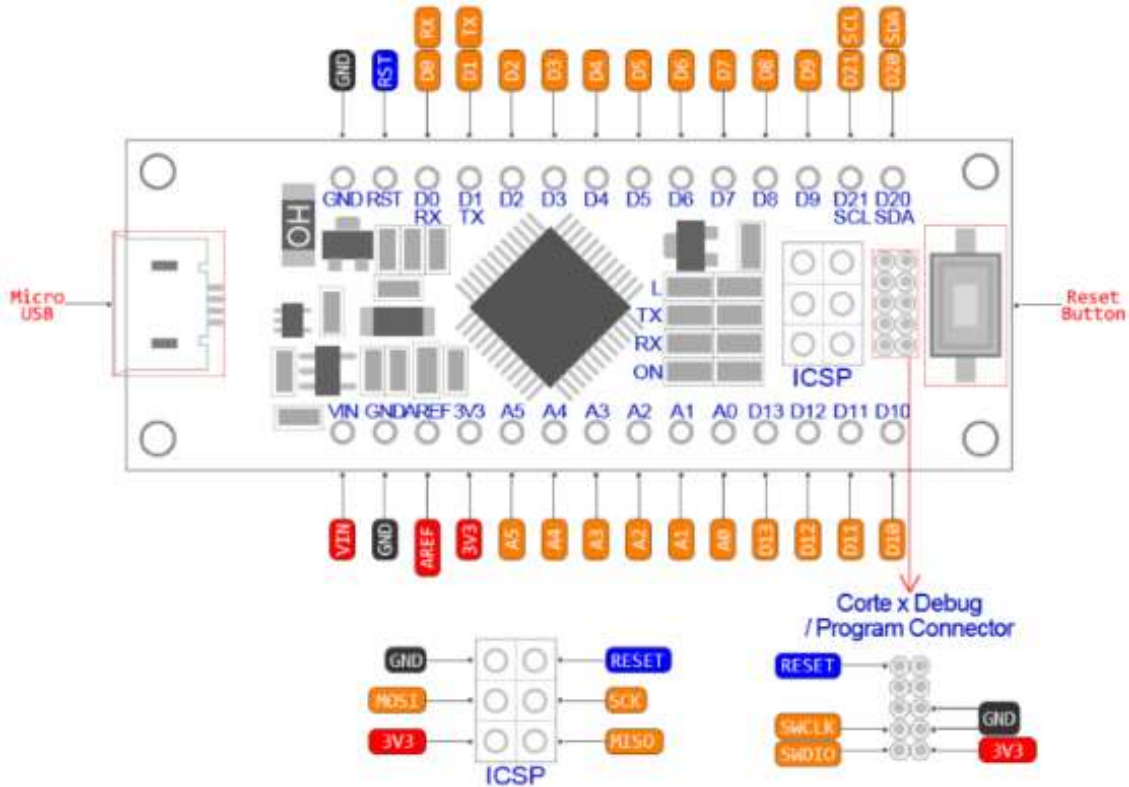
Puesto que muchos de los sensores se pueden escoger para que su comunicación sea mediante el protocolo *Inter Integrated Circuits* (I2C), común a todos los microcontroladores presentados, y por tanto en paralelo por los mismos pines, el segundo requisito no es restrictivo frente a la selección actual; tampoco lo es en cuanto al conteo de pines PWM, necesarios para el control de los ESC y por tanto de los motores.

En cuanto al tamaño de la memoria *flash*, se observa que, con solo 32kB, la Arduino Nano quizá no sea una buena opción. De entre las dos placas restantes, se puede observar que la Teensy 4.1 supera en muchos aspectos a la SAMD21 M0-mini; sin embargo, ya que ambas cumplen con los requisitos prefijados y dada la diferencia entre sus precios, se ha decidido emplear la SAMD21 M0-mini.

Por último, en la Figura 5 se muestra la distribución de pines de la placa seleccionada; cabe destacar dos datos más: en primer lugar, que no se emplean en el proyecto los pines del sistema de depuración; y, en segundo lugar, que todos los pines de las SAMD21 permiten interrupción. También es de interés mencionar que el fabricante original, RobotDyn, ya no comercializa estas placas controladoras; la única manera de obtenerlas es a través de otras empresas que fabrican la placa con la misma configuración.

Figura 5

Asignación de pines de la placa SAMD21 M0-Mini.



Nota. Adaptado de *PINOUT DIAGRAM SAMD21 M0-Mini*, de RobotDyn, 2017, <https://www.robotdyn.com>.

3.1.2. Sensor de presión

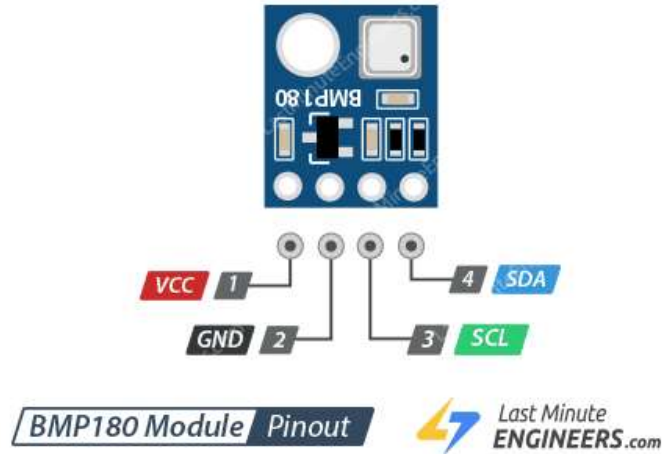
Aunque una buena opción habría sido el uso del barómetro BMP280 tal y como se ha indicado en el apartado 2.4, se disponía ya de su versión predecesora (BMP180) por proyectos anteriores. Además, entre ambos dispositivos se mantiene la precisión y el rango, por lo que serían fácilmente intercambiables.

Este dispositivo permite medir la presión y temperatura del aire, de manera que se corrige la primera mediante la segunda. Gracias a la diferencia entre la presión inicial y la medición se puede obtener la distancia respecto al suelo, y por tanto controlar que esta no exceda los 120m de altitud máxima permitida, según se referencia en el apartado 1.4.

El patillaje del dispositivo se muestra en la Figura 6, realizándose la conexión mediante el protocolo I2C.

Figura 6

Asignación de pines del sensor BMP180.



VCC is the power supply for the module which can be anywhere between 3.3V to 5V.

GND should be connected to the ground of Arduino.

SCL is a serial clock pin for I2C interface.

SDA is a serial data pin for I2C interface.

Nota. Adaptado de *BMP180 Module Pinout*, de Last Minute ENGINEERS, <https://lastminuteengineers.com/bmp180-arduino-tutorial/>.

3.1.3. Unidad de medición inercial

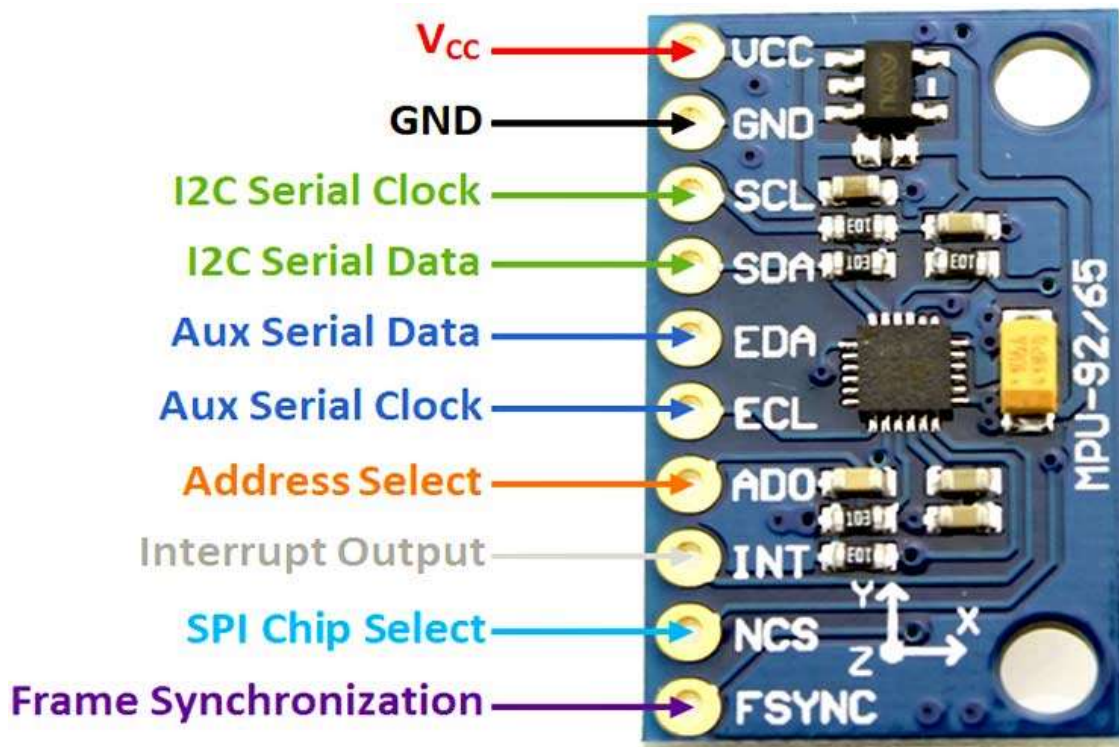
Existen en el mercado un gran número de dispositivos IMU con diferentes características; en lo referente al análisis realizado en el apartado 2.4, se observa el uso tanto de 6 ejes (3 ejes para controlar el acelerómetro y otros 3 para el giroscopio) como de 9 (con 3 pines más para el control del magnetómetro).

Aunque es posible determinar la dirección de la aeronave mediante los 6 primeros ejes especificados, se optará por una IMU de 9 grados de libertad, con el fin de obtener mediciones más precisas; para ello se empleará un filtro Mahony, frente a otros como Madwick o un filtro de Kalman extendido, por su rápida ejecución y su bajo error (Ludwig y Burnham, 2018). Se emplea, por tanto, la IMU MPU9250, que integra los 3 sensores y cuenta con una biblioteca específica para ser calibrado y aplicar el filtro Mahony.

En la Figura 7 se muestra el patillaje correspondiente. Es importante resaltar que el chip permite la conexión tanto mediante I2C como con Serial Peripheral Interface (SPI). Si bien es verdad que el segundo protocolo es más rápido, se decide emplear el primero; esto es debido a que la biblioteca que se utiliza para la fusión de las lecturas funciona mediante este protocolo de comunicación: La diferencia de velocidad entre ambos protocolos no es tal que pudiera afectar al correcto funcionamiento.

Figura 7

Asignación de pines del sensor MPU9250.



Nota. Adaptado de *MPU9250 Module Pinout*, de COMPONENTS101, <https://components101.com/sensors/MPU9250-9-dof-mems-sensor-module-datasheet-pinout-features-working>.

3.1.4. Transceptor de radiofrecuencia

Respecto a la comunicación por radio, se emplea el dispositivo RFM69HC; tanto por la amplia comunidad que lo sustenta, como por su alta potencia. Existen otros dispositivos con estas características como el RFM22B, pero presenta una menor *ratio* de envío de datos. Otra alternativa de especificaciones casi idénticas es el TI CC1101, pero el RFM69HC tiene un soporte mayor por parte de la comunidad de Arduino, lo que lo hace más atractivo de cara a posibles problemas.

Por otra parte, será necesario emplear una antena para aumentar el alcance del dispositivo; para ello se ha elegido una antena omnidireccional de polarización vertical. La ganancia de la antena seleccionada es de 5dBi (G_a). Teniendo en cuenta las características del dispositivo y la antena, y empleando la Ecuación (2), se tiene una potencia radiada aparente de 22.85dBm;

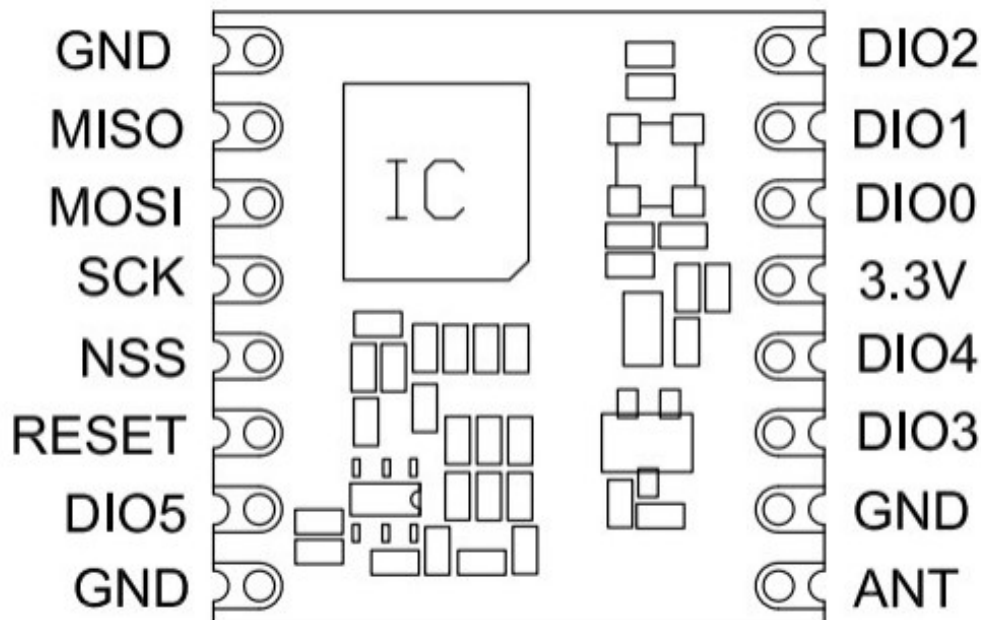
que equivale a 193mW. Si bien este valor se encuentra por encima de lo admisible según la norma referenciada en el apartado 1.4, no se han tenido en cuenta las pérdidas de potencia en el cable (L_c); en cualquier caso, el dispositivo puede ser regulado para funcionar a potencias menores a la máxima.

$$PRA = P_T - L_c + G_a - 2,15 \quad (2)$$

En la Figura 8 se puede observar el patillaje del dispositivo. Cabe destacar que el pin de antena no está preparado para ser conectado a un cable coaxial, por lo que será necesario incorporar un conector uFL SMT; que consta del patillaje necesario para conectar correctamente el cable a masa.

Figura 8

Asignación de pines del transceptor RFM69HC



Nota. Adaptado de *Marking Diagram*, de HopeRF, https://www.hoperf.cn/data/upload/portal/20190225/RFM69HC_DataSheet_v1.1.pdf.

3.1.5. Cámara

Como se ha comentado en el apartado 2.5.6, se proponen dos soluciones diferentes: una para el prototipo, que únicamente requiere de este sistema para comprobar otras características; y otra para ofrecer una solución viable y real respecto a las necesidades planteadas.

Para el prototipo, se ha escogido la cámara A9 HD 1080p que genera su propio *hotspot* WiFi a 2.4GHz; el móvil se conecta al mismo mediante contraseña y, a través de una aplicación, se retransmite. La cámara se alimenta desde una batería interna. Por supuesto, esta no es una solución aceptable para las velocidades de operación, pero es adecuada para las pruebas del prototipo.

La otra opción que se plantea es el uso de una cámara con transmisor de 5.8GHz integrado. En este caso, sería necesario comprar un receptor acorde; el Skydroid es una opción popular para este fin. Además, la cámara debería recibir alimentación por parte de la propia batería del dron. El conjunto asciende por encima de los 50 €, frente a los 10 € que se plantean en el uso de la cámara A9 HD 1080p.

Puesto que entre ambas no existe una diferencia de implementación significativa, el prototipo se diseña para la cámara A9 HD 1080p; pero de cara al cálculo de los requisitos de consumo de energía (es decir, para la elección de la batería) y de potencia de los motores se tendrá en cuenta la segunda opción, acorde al objeto del presente proyecto. Adicionalmente, se incluirán las modificaciones necesarias en el chasis de la aeronave para soportar la cámara de la segunda propuesta, que no dispone de soporte; los estudios aerodinámicos se realizarán para esta configuración.

En la Figura 9 se muestra el equipo necesario tanto para el prototipo (a) como para el objeto del proyecto (b). Notar que las imágenes no están a escala.

Figura 9

Equipos necesarios para la transmisión de video.



Nota. Fuente propia. Elaborada a partir de las imágenes ofrecidas por las empresas distribuidoras.

3.1.6. Motores y hélices

En base a la información presentada en los apartados 2.5.1 y 2.5.2, se desea encontrar un conjunto hélice-motor que proporcione el suficiente empuje para compensar el peso de la aeronave. En este apartado no se estudia con profundidad la aerodinámica del UAS; por el contrario, se analiza la potencia de motor necesaria para mantener la aeronave en el aire.

A la hora de elegir un motor, se debe tener en cuenta el diámetro de las hélices. También es necesario definir el paso de la hélice. Para aeronaves *Long Range* se emplean distancias entre motores, por lo general, mayores a 250mm, lo que implica un diámetro de la hélice de al menos 7" (17.78cm) (Drone Nodes, s.f.). Para asegurar que el motor es adecuado para la hélice, se decide comprar en conjunto; así pues, queda por determinar qué potencia de motor es adecuada.

En primer lugar, será necesario determinar el peso de la aeronave; se estima en base a los datos conocidos y a pesos máximos admisibles. Para ello, se ha elaborado la Tabla 3, que contiene los pesos de los diferentes módulos seleccionados. Se considera que el peso del marco, así como el resto de los elementos estructurales, no deberá sobrepasar 1kg. La batería, por su parte, no deberá exceder 0.5kg. La PCB, y algunos elementos más pequeños, presentados en el apartado 3.1.8, se estiman en 0.13kg

Tabla 3

Peso de las partes de la aeronave.

Parte	Peso (gramos fuerza)
Estructura de la aeronave	1000
Batería	500
PCB	130
RFM69HC	1.5
Antena	17
MPU9250	2
<i>Level Shifter</i>	1.5
Regulador de tensión	1.5
SAMD21 M0 mini	6
BMP180	1
Cámara	40
Total	1700.5

Además, para poder elevarse no solo será necesario que sea capaz de compensar su propio peso, sino que debe poder generar un empuje positivo. Se toma como referencia de cálculo, de forma conservadora, el 200% del peso final. De esta manera, de acuerdo con la teoría de momentos clásica (Sforza, 2017), y aceptando el supuesto de flujo estacionario en cuatro hélices abiertas sin interferencia, se llega a la Ecuación (3), donde la potencia de los motores (4·P), así como su respectivo rendimiento, se relacionan con el diámetro de la hélice (D) y el peso (T). Se seleccionará un conjunto que cumpla con este requisito.

$$4 \cdot P \cdot \eta_{mec \ y \ elec} = \sqrt{\frac{T^3}{2 \cdot \rho \cdot \pi \cdot D^2}} \quad (3)$$

El conjunto seleccionado consta de un motor (A2212/1400KV 10T) de 180W con una eficiencia del 75%, hélices de dos palas de 8'' (20.32cm) de diámetro y un ESC. Se debe tener en cuenta que la incorporación de estos motores al diseño implica añadir 300g más, no incluidos en la Tabla 3. Con todo ello (1.97kg), se mayor el peso según lo expuesto, resultando en 4.001kg (equivalente a 39.25N); la potencia requerida es inferior a la que es capaz de aportar el motor: 45.25N según (3).

Las hélices del conjunto presentan pasos de hélice de 4.5'' (11.43cm), que es un valor medio-alto en lo que se refiere a estas escalas. Dado que se pretende alcanzar altas velocidades sin sacrificar excesivamente la aceleración, se da por válido. Por su parte, los motores presentan un valor de 1400KV; esto, junto con el voltaje de la batería, determina la máxima velocidad de giro de las hélices. La corriente consumida por el motor que asegura la eficiencia del fabricante es toda aquella entre 4-10A. Para el cálculo de la batería se tomarán 7A como corriente nominal.

En lo referente a los ESC del conjunto, están montados sobre sistemas *Battery Eliminator Circuit* (BEC) lineales. Esto implica que, al conectarlos a una fuente, absorberán la diferencia de voltaje entre la batería y su punto de funcionamiento, 5 V, convirtiendo dicho exceso en calor. A pesar de todo, el fabricante especifica su compatibilidad con baterías LiPo de hasta 3 celdas (11,1 V); además, se especifica que la corriente máxima admitida por los mismos es de 30 A.

Los BEC permiten conectar los ESC desde la batería, dividiendo el circuito en dos potencias diferentes. En cualquier caso, la modulación de los ESC funciona a 5 V, mientras que la placa envía señales PWM a 3,3 V; a fin de evitar problemas, es necesario implementar un cambio lógico a nivel de voltajes. Para ello se emplean adaptadores de nivel. Adviértase que en la Tabla 3 solo se ha incluido uno, debido a que cada módulo es capaz de operar hasta 4 señales independientes.

3.1.7. Batería

En primer lugar, se ha elaborado la Tabla 4 donde se especifican los consumos de corriente de los distintos componentes. Con esta información se puede obtener el consumo nominal de corriente y, para un tiempo medio de vuelo de 10 minutos como se observa en el apartado 2.4, los mAh que debe tener la batería. Cabe destacar que: se ha considerado el consumo de la radio RFM69HC cuando actúa como receptor; el BMP no se ha incluido por consumir menos de 1mA, al igual que otros componentes mencionados en el apartado 3.1.8.

Tomando como referencia el consumo nominal y multiplicando dicha cifra por los 10 minutos de vuelo y pasando el resultado a mAh, se obtiene mediante la Ecuación (4) una carga de 4758mAh. En el apartado 2.4 se puede observar que, a pesar de presentar características similares, ningún dron comercial incorpora una batería con tanta capacidad de carga.

$$Capacidad = Consumo \cdot Tiempo \ de \ funcionamiento \quad (4)$$

Tabla 4

Consumos de corriente de los elementos de la aeronave.

Componente	Consumo (mA)
RFM69HC	16
MPU9250	4
SAMD21 M0-mini	220
I/O pin (máx.)	63
• SCL	(7)
• SDA	(7)
• MOSI	(7)
• MISO	(7)
• PWM motores (4)	(28)
• LED	(7)
Motores (4)	28000
Regulador de tensión	25
Cámara (caso del objeto del proyecto)	220
Total	28548

Por otra parte, se ha de asegurar una *ratio* de descarga de la batería que cumpla con las necesidades del proyecto; dicha capacidad de descarga viene determinada por la máxima corriente que es capaz de consumir el dron en operación. Este cálculo es necesario hacerlo en base al consumo de corriente máximo de los ESC para evitar posibles complicaciones, en este caso 120A sustituirán a los 28A requeridos por los motores.

Para conocer la descarga máxima de una batería LiPo, se ha de tener en cuenta la carga que es capaz de almacenar, así como el parámetro C (*ratio* de descarga); multiplicando uno por otro se pueden conocer los amperios que es capaz de suministrar la fuente de manera segura. Asumiendo una capacidad de 5000 mAh, la *ratio* de descarga debería ser al menos 25C según la Ecuación (5). La batería seleccionada es la OVONIC LiPo 3s 5000mAh 50C. El peso de la batería es de 320 g, cumpliendo con el requisito impuesto en el apartado 3.1.6.

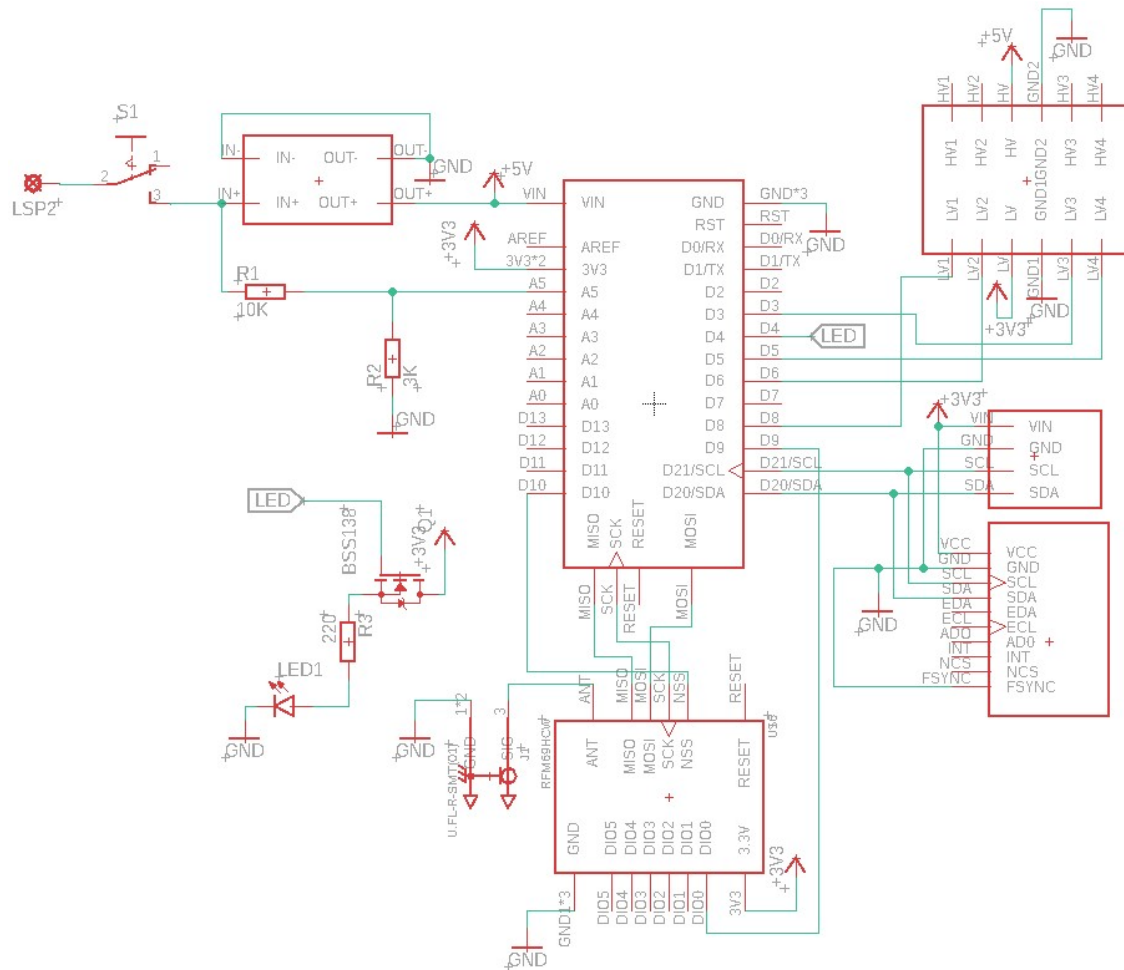
$$Ratio = \frac{Consumo\ máximo}{Capacidad} \quad (5)$$

3.1.8. Circuito propuesto

Una vez seleccionados todos los componentes, es necesario establecer el circuito de la aeronave para materializarlo en forma de PCB. Se han incorporado elementos secundarios, algunos ya mencionados, como se puede apreciar en la Figura 10; dichos elementos son: un regulador de tensión, un LED, un MOSFET, tres resistencias, un interruptor y un conector para la antena de la radio.

Figura 10

Circuito de la aeronave.



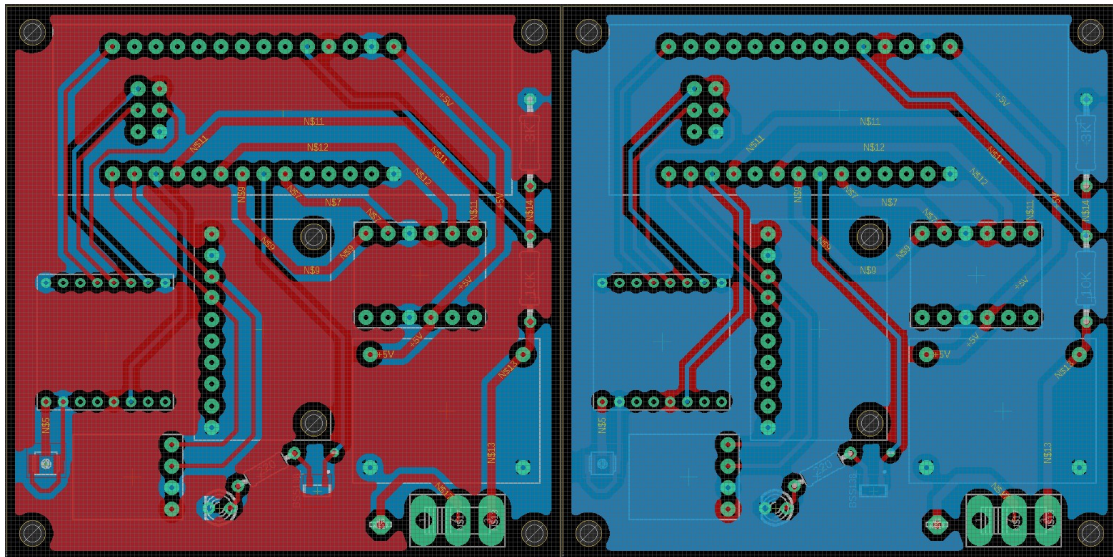
Nota. Fuente propia. Elaborada mediante el programa Eagle.

El MOSFET se emplea para controlar el conjunto resistencia-LED. Las otras dos resistencias generan un divisor de voltaje para leer la carga de la batería desde un pin analógico; el divisor de tensión se ha configurado mediante una resistencia de 10kΩ y otra de 3kΩ. De esta manera el voltaje máximo de salida, para carga completa de la batería (4.2V por celda, 12.6V en total), será de 2.91V; mientras que el mínimo, sin provocar daños a la batería (3.4V por celda, 10.2V en total), será de 2.35V. Con ello, se evita alcanzar el límite de 3.3V de entrada al convertidor analógico digital (ADC por sus siglas en inglés) de la SAMD21 M0-mini.

En la Figura 11, por otra parte, se aprecia la propuesta de la PCB; se incorporan dos imágenes para apreciar mejor las dos capas que la componen. Notar que, aunque el programa detecta la falta de conexión con un pin de masa, está conectado internamente en su respectivo dispositivo. Para la distribución del voltaje de alimentación a 3.3V y la masa se han dispuesto superficies equipotenciales en sendas caras.

Figura 11

Placa de circuito impreso de la aeronave.



Nota. Fuente propia. Elaborada mediante el programa Eagle

Para el desarrollo de las placas, se han creado bibliotecas personalizadas para la mayoría de los elementos; atendiendo a las dimensiones tomadas con un pie de rey y las referencias de los fabricantes. Los elementos descargados han sido modificados y/o comprobadas sus medidas para asegurar que los espacios asignados en la PCB se correspondan a la realidad.

Los elementos se conectan mediante pistas de 1mm y 0.5mm; de esta manera se asegura que la intensidad máxima que puedan soportar sea adecuada (Díaz, 2021), también se reduce el ruido de las señales y la resistencia propia a la conexión. Por otra parte, se ha habilitado algo de espacio libre en la parte inferior, donde sería posible conectar un par de pines: a GND y a +3V3, para alimentar la cámara.

3.2. MANDO

En lo que respecta al mando, el módulo de radio ha quedado establecido en el apartado 3.1.4, por lo que no se detalla aquí. A pesar de todo, sí forma parte del conjunto, como se aprecia en el desarrollo del circuito y la PCB correspondiente. Quedan pues, por establecer el microcontrolador y el método de interacción entre usuario y mando.

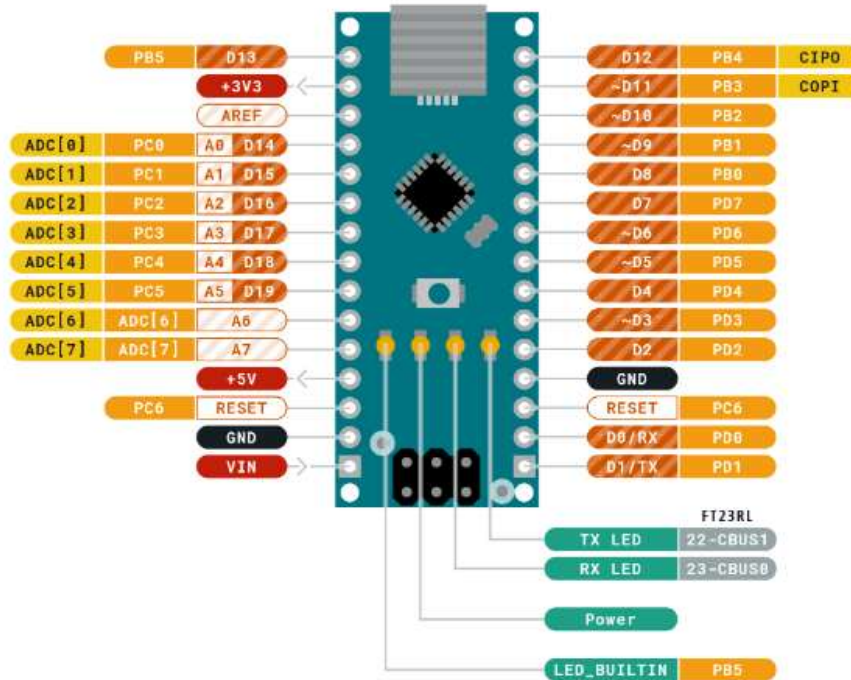
3.2.1. Controlador

Puesto que el mando no requiere ni de mucha potencia de procesamiento ni de mucha memoria *flash*, ya que su única función es retransmitir las órdenes del usuario, se ha escogido la Arduino Nano ya presentada en el apartado 3.1.1.

En la Figura 12 se muestra el patillaje de la Arduino Nano. Aunque es verdad que las placas de Arduino constan tanto de un pin de alimentación a 3.3V como de uno a 5V, los pines I/O solamente trabajan a 5V; esto significa que será necesario emplear un cambio de nivel lógico para comunicar el microcontrolador y la radio. Hay que destacar que la placa también permite comunicación SPI, aunque en la imagen se ha obviado el etiquetado de sus pines.

Figura 12

Asignación de pines de la placa Arduino Nano.



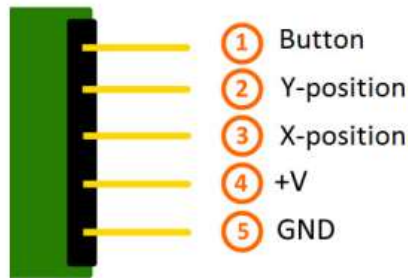
Nota. Adaptado de *PINOUT ARDUINO NANO*, de Arduino, <https://docs.arduino.cc/hardware/nano>.

3.2.2. Joysticks

Para interactuar con el mando se emplean dos *joysticks*: uno para controlar la altitud y la guiñada, y otro para el cabeceo y el alabeo. Se corresponden con los ejes Y y X de los *joysticks*, respectivamente. Estos dispositivos no son más que la unión entre dos potenciómetros accionados por palancas regulables. En la Figura 13 se puede observar el patillaje de un *joystick*, siendo igual para ambos.

Figura 13

Asignación de pines del joystick KY-023.



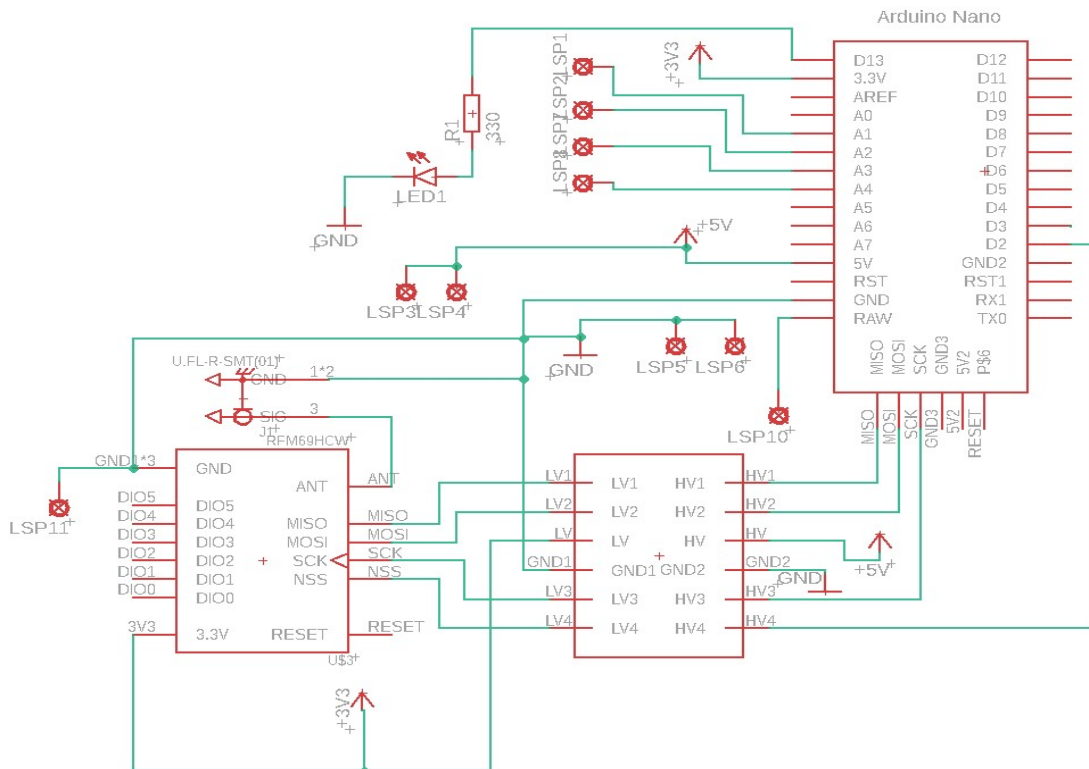
Nota. Adaptado de KY-023 Joystick module (XY-Axis), de JOY-IT, <https://datasheetspdf.com/pdf-file/1402034/Joy-IT/KY-023/1>.

3.2.3. Circuito propuesto

De igual forma que en el caso de la aeronave, se ha diseñado una PCB para alojar la electrónica del mando. Los *joysticks* se conectarán a la placa mediante cables, debido a la dirección de los pines de salida de los mismos. No se ha considerado necesario introducir un sistema para la lectura de la batería por no ser necesario para comprobar el buen funcionamiento del prototipo, pero sí sería necesario de cara a su producción final. En la Figura 14 se puede apreciar el circuito.

Figura 14

Circuito del mando.

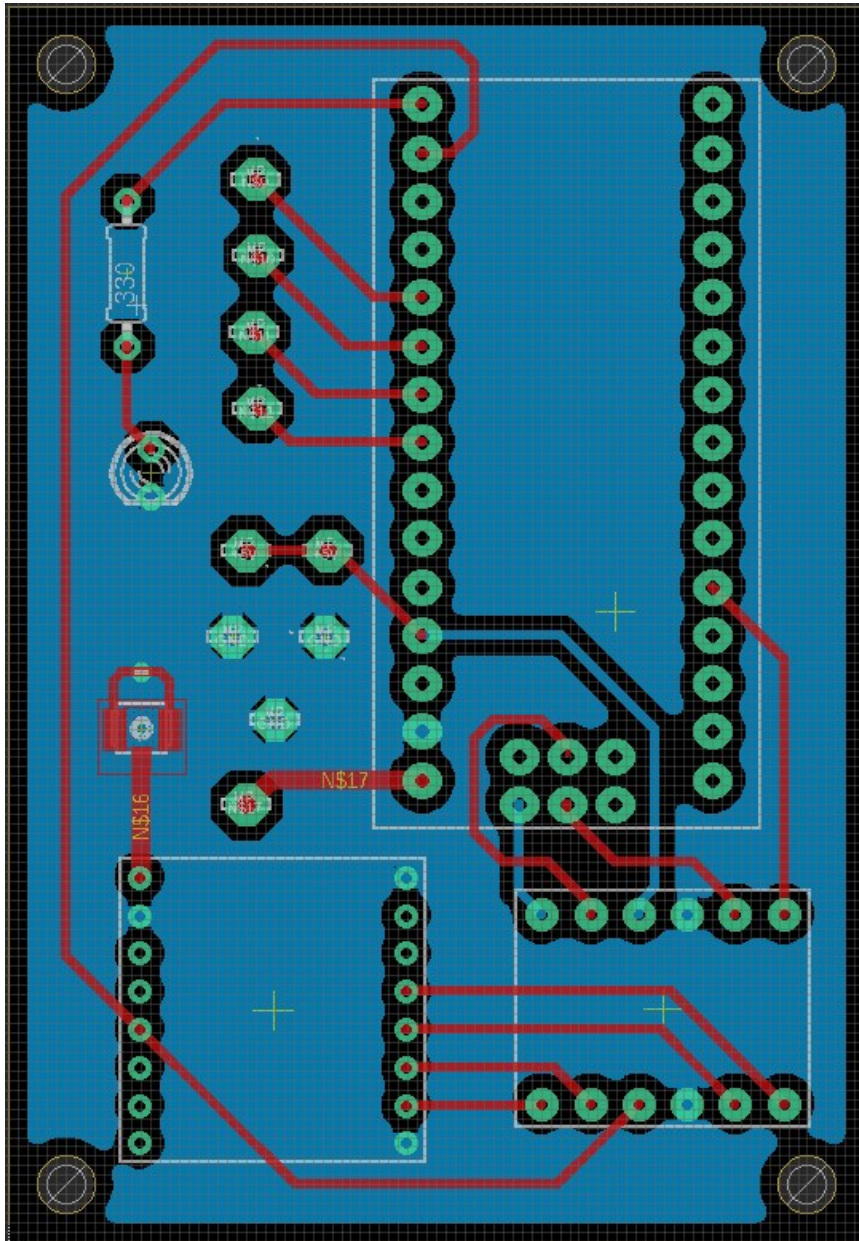


Nota. Fuente propia. Elaborada mediante el programa Eagle.

En la Figura 15 se muestra la PCB propuesta para el mando; obsérvese que en este caso el interruptor se acopla en la carcasa, conectado mediante cable soldado a la placa, en lugar de estar directamente sobre la placa. Los pines habilitados para las lecturas de los joysticks parten de los pines analógicos de la placa; también se han tenido en cuenta aquellos que se emplean para alimentarlos. Se aprecia, además, espacio suficiente en caso de necesitar añadir componentes para la lectura de la batería.

Figura 15

Placa de circuito impreso del mando.



Nota. Fuente propia. Elaborada mediante el programa Eagle.

Puesto que, en este caso, el suministro máximo de corriente por cada pin I/O es de 40mA, no se requiere el uso de un transistor MOSFET que actúe como interruptor, como sí era necesario en la aeronave. Tampoco se ha visto necesario crear una superficie equipotencial positiva por la simplicidad del circuito.

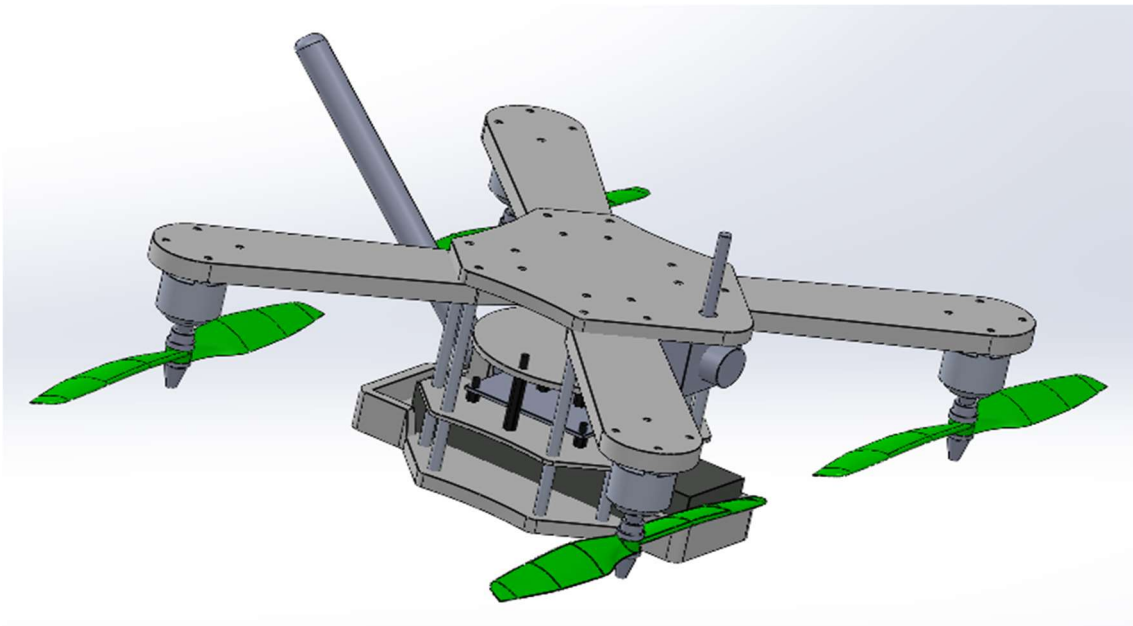
CAPÍTULO 4. DISEÑO ESTRUCTURAL

4.1. CHASIS DE LA AERONAVE

Como se ha indicado en el apartado 2.3, el proceso de elaboración del diseño del chasis de la aeronave se ha realizado de forma iterativa. La idea inicial de diseño se muestra en la Figura 16. Por medio de la herramienta Flow Simulation, un complemento de *Computational Fluid Dynamics* (CFD) de SolidWorks, se han analizado diferentes geometrías, optimizando el diseño en base a los resultados. Se toma una distancia entre motores (opuestos) de 290mm, cuya validez se verifica en el apartado 4.1.2. Se ha omitido la acción gravitatoria en todos los estudios, controlando el peso como resultado y no como dato.

Figura 16

Concepción inicial de la aeronave.



Nota. Fuente propia. Elaborada mediante el programa SolidWorks.

4.1.1. Estudio del arrastre y la sustentación

En primer lugar, se analiza la interacción de la geometría con el medio. En adelante, se trabaja con una inclinación de la aeronave de 45°; en el apartado 4.1.2 se estudia la viabilidad de este ángulo en operación. Puesto que 45° es el máximo ángulo de ataque que se va a considerar para evitar problemas en la estabilidad, se ha tomado como referencia para este estudio; el motivo de esto es que, en esta configuración, se ejerce la máxima fuerza en el eje X por parte de los motores.

Las fórmulas empleadas para el cálculo han sido las presentadas en las Ecuaciones (6) y (7), siendo la primera una extensión de (1) con el principio de Arquímedes; no es necesario contemplar el peso por lo expuesto en anteriormente. Para la obtención del área perpendicular a la dirección de avance se ha creado un plano frente al ensamblaje y se ha proyectado la geometría del mismo mediante la herramienta Silueta. Una vez obtenida la superficie cerrada, simplemente se obtiene la magnitud a través de la instrucción Medir.

$$C_L = \frac{2 \cdot (|F_Y| - V \cdot g \cdot \rho)}{\rho \cdot A \cdot V^2} \quad (6)$$

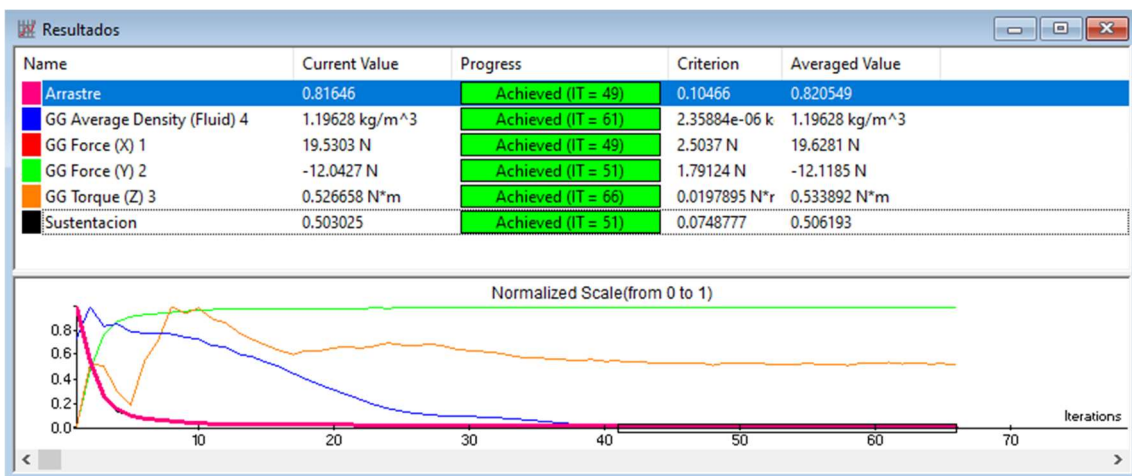
$$C_D = \frac{2 \cdot F_X}{\rho \cdot A \cdot V^2} \quad (7)$$

El volumen de la aeronave se obtiene al comprobar la geometría antes de proceder con el análisis. Las fuerzas horizontales y verticales sobre el sólido, así como la densidad del fluido, son datos calculados por el mismo programa; es por esto que es necesario tener en cuenta el empuje generado por la geometría, de lo contrario estaríamos computando, en la ecuación, una fuerza mayor a la realmente generada por la sustentación. La velocidad del fluido, por otra parte, es una de las condiciones de contorno; también lo es la humedad del aire, establecida en el 70% por ser el valor promedio en la costa mediterránea española (lugar de estudio).

A partir de la concepción inicial, se han modificado las partes a fin de minimizar ambos coeficientes. El motivo de minimizar el coeficiente de sustentación, así como el hecho de tomar el valor absoluto de la fuerza en el eje Y, es que dicha fuerza se produce en dirección al suelo en condiciones normales de operación. Una vez mallada la geometría inicial, se ejecuta el análisis; los resultados se muestran en la Figura 17. En la Figura 18 se pueden observar las presiones en el plano de simetría del chasis; por otra parte, en la Figura 19, se muestran las líneas de corriente de aire en este mismo plano. Clarificar que para los estudios de arrastre y sustentación no se han tenido en cuenta las hélices, que han sido excluidas del análisis.

Figura 17

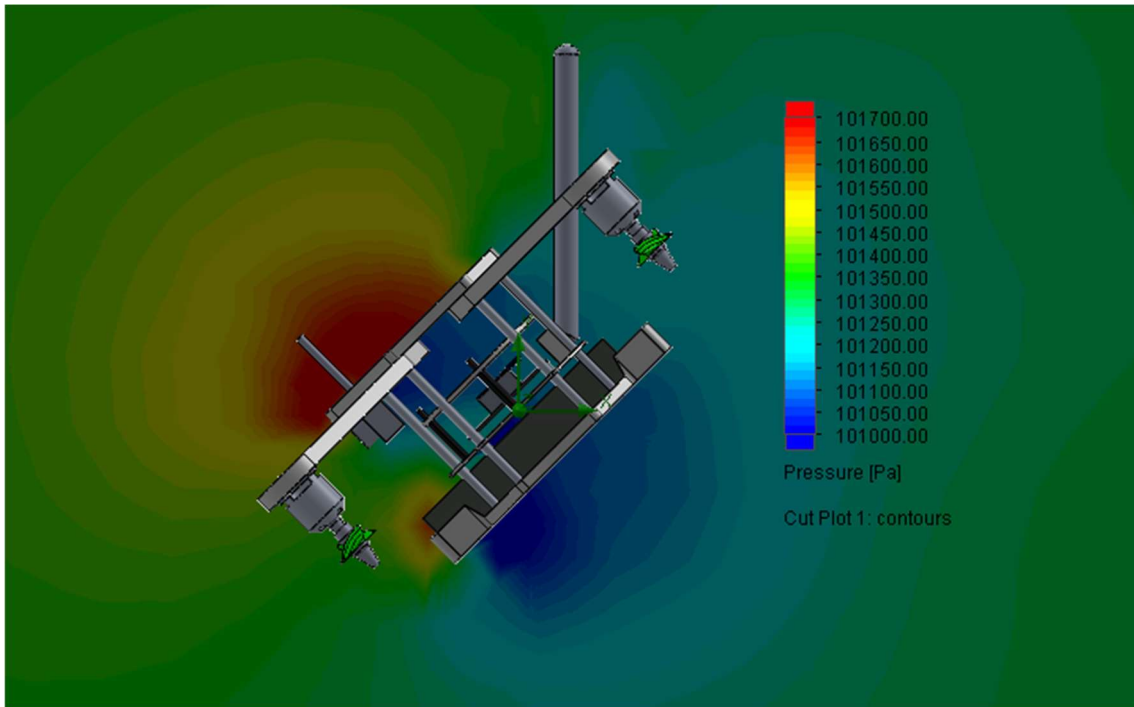
Arrastre y sustentación para la geometría inicial.



Nota. Fuente propia. Elaborada mediante el programa SolidWorks Flow Simulation. La imagen muestra los coeficientes de sustentación y arrastre, resultantes de aplicar las Ecuaciones (6) y (7) respectivamente. $C_L \approx 0.503$, $C_D \approx 0.816$; se genera un par a favor del ángulo de ataque.

Figura 18

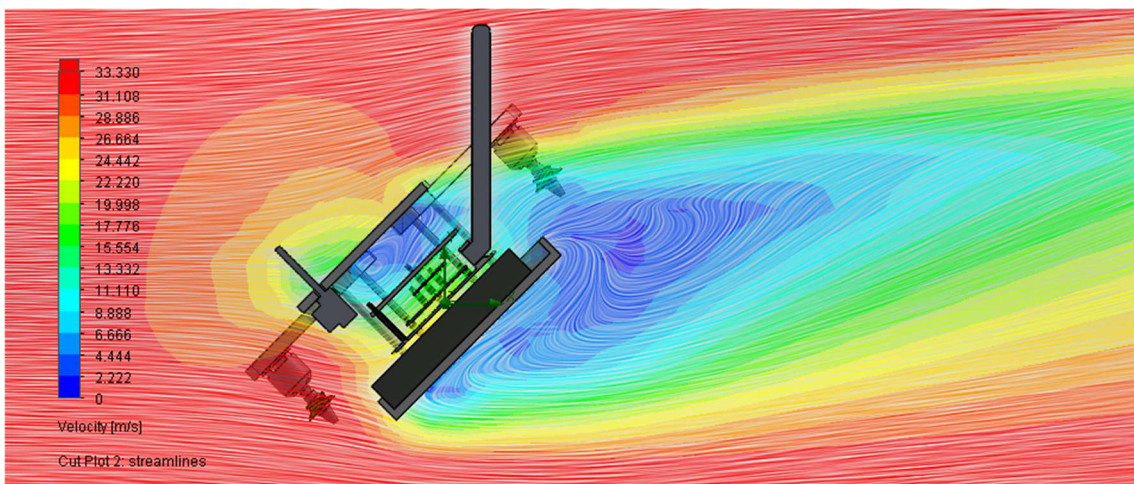
Presiones plano de simetría, arrastre y sustentación para la geometría inicial.



Nota. Fuente propia. Elaborada mediante el programa SolidWorks Flow Simulation. El gradiente de presiones generado es responsable de las fuerzas presentadas en la Figura 17, también explica el par que se genera.

Figura 19

Velocidades plano de simetría, arrastre y sustentación para



Nota. Fuente propia. Elaborada mediante el programa SolidWorks Flow Simulation. En la imagen se aprecia el vórtice generado a causa de la interposición del cuadricóptero con el flujo de aire; este fenómeno se da tanto aguas debajo de la aeronave como en su interior, la velocidad del fluido disminuye drásticamente en estas zonas.

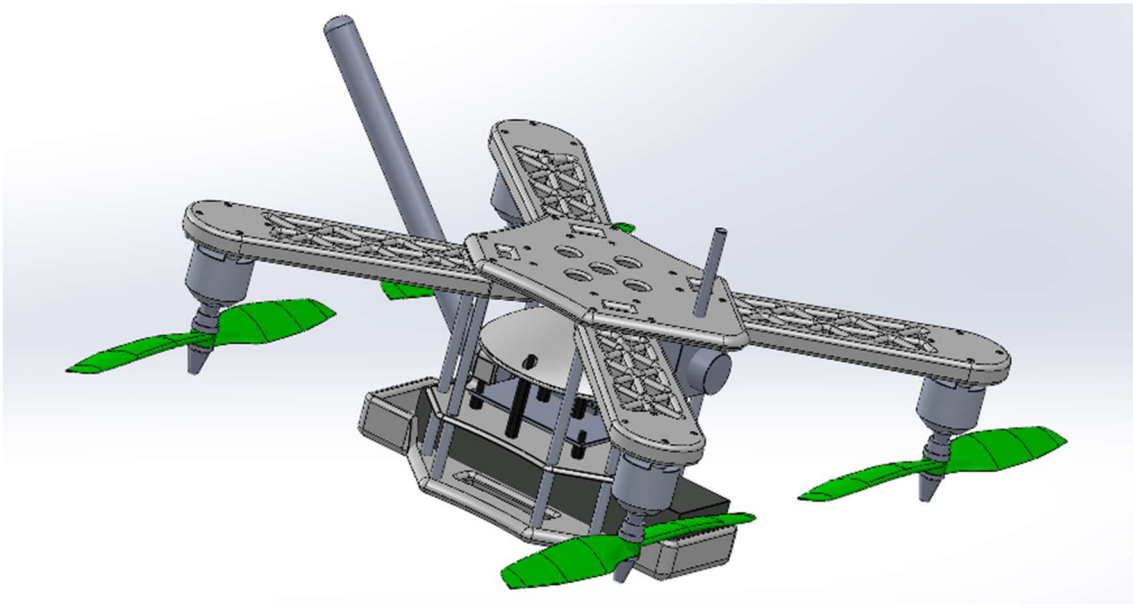
Para lograr estos objetivos, se han introducido cambios notables en el diseño de la aeronave; como se aprecia en la Figura 20. Sin entrar en el detalle de cada pieza, puesto que los estudios se realizan sobre el conjunto, se comentan a continuación los cambios más notables y las razones por las que han sido introducidos.

En primer lugar, se han creado agujeros en la parte superior del dron con la finalidad de reducir el choque del aire contra la misma; mientras que los de las alas no presentan ninguna geometría especial, los que se encuentran sobre el marco se han diseñado con forma de difusor (asumiendo velocidades a su través menores a 1 Mach) para aprovechar la velocidad relativa del aire respecto a la aeronave y conseguir un aumento de presión a la salida.

Por otra parte, la estructura que protege la electrónica ha sido modificada para reducir su impacto sobre los coeficientes de arrastre y sustentación; con este fin, se han probado diversas geometrías, comparándolas, hasta obtener resultados satisfactorios. También se han introducido redondeos para mantener mejor la adherencia de la capa límite; aunque esto no impide que se despegue del cuerpo, se reduce de esta forma el efecto vorticial en el flujo aguas abajo de la aeronave.

Figura 20

Diseño propuesto para la aeronave.

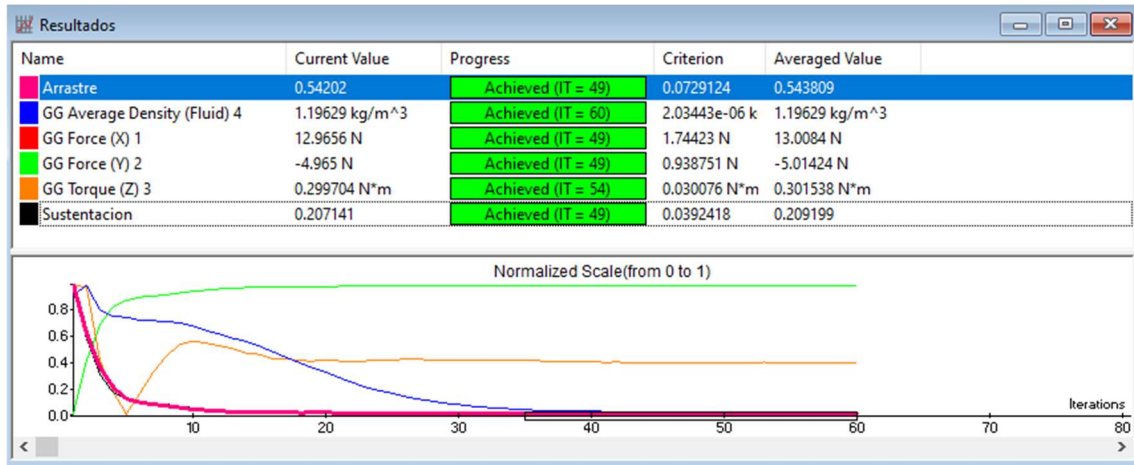


Nota. Fuente propia. Elaborada mediante el programa SolidWorks.

Con todo ello, se malla el conjunto y se ejecuta el análisis una vez más; los resultados se muestran en la Figura 21. En la Figura 22 se pueden observar las presiones en el plano de simetría del chasis; por otra parte, en la Figura 23, se muestran las líneas de corriente de aire en este mismo plano.

Figura 21

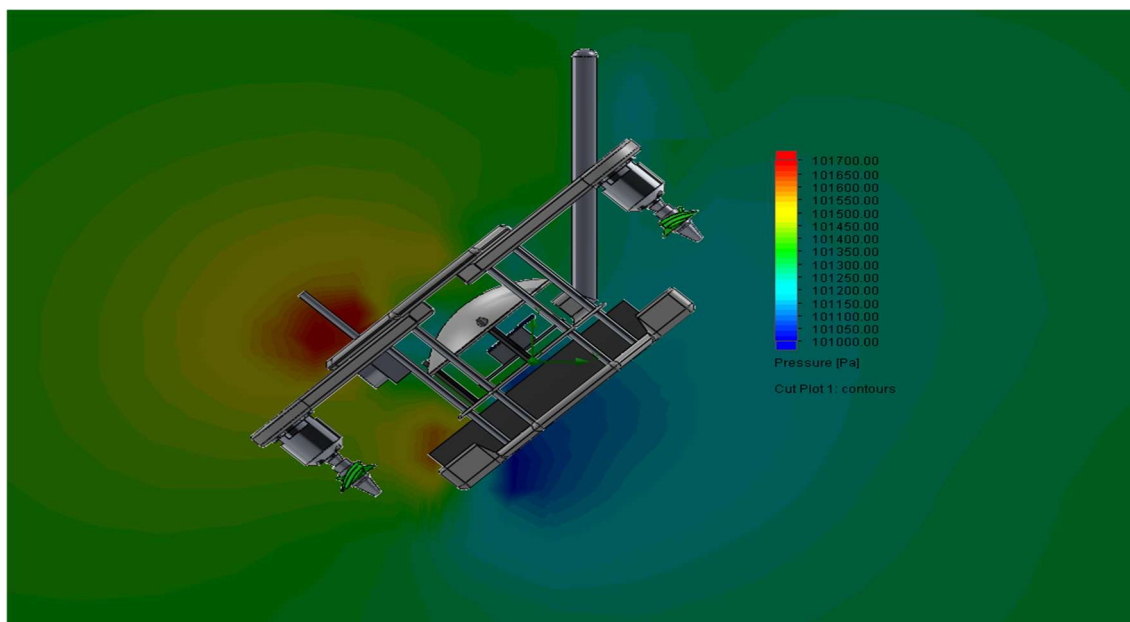
Arrastre y sustentación para un ángulo de ataque de 45°.



Nota. Fuente propia. Elaborada mediante el programa SolidWorks Flow Simulation. Los coeficientes presentan los valores $C_L \approx 0.207$ y $C_D \approx 0.542$, lo que supone una reducción del 58.85% y 33.58% respectivamente. También se observa que el momento mantiene la misma dirección, pero se ha reducido su magnitud.

Figura 22

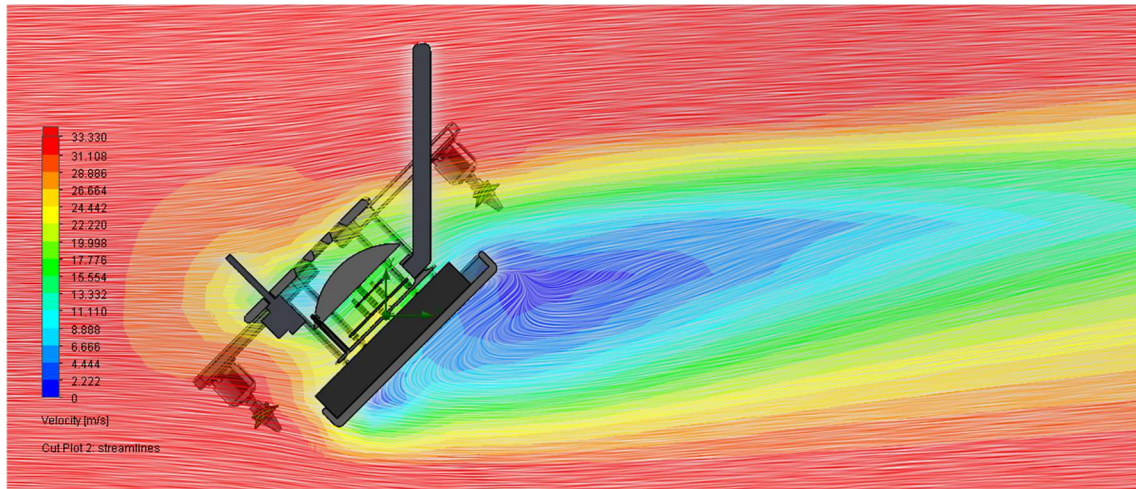
Presiones plano de simetría, arrastre y sustentación para un ángulo de ataque de 45°.



Nota. Fuente propia. Elaborada mediante el programa SolidWorks Flow Simulation. De nuevo, el gradiente de presiones justifica los resultados anteriores; visualmente, se puede apreciar la reducción del mismo con respecto al generado en la geometría inicial. El cambio más destacable es la presión que se observa ahora en el interior de la aeronave, que revela que la función de los difusores es satisfactoria.

Figura 23

Velocidades plano de simetría, arrastre y sustentación para un ángulo de ataque de 45°.



Nota. Fuente propia. Elaborada mediante el programa SolidWorks Flow Simulation. Comparándola con su homóloga para la geometría inicial, se puede observar que el vórtice aguas debajo de la aeronave es menos pronunciado.

4.1.2. Velocidad terminal de la aeronave

En el apartado anterior, las iteraciones se realizan en pos de reducir los coeficientes de arrastre y sustentación variando la geometría; en este apartado, la geometría se considera fija, y se itera en base a la velocidad relativa del fluido y el ángulo de inclinación. En primera instancia, no obstante, es necesario realizar de nuevo los estudios de arrastre y sustentación respectivos a la inclinación pertinente.

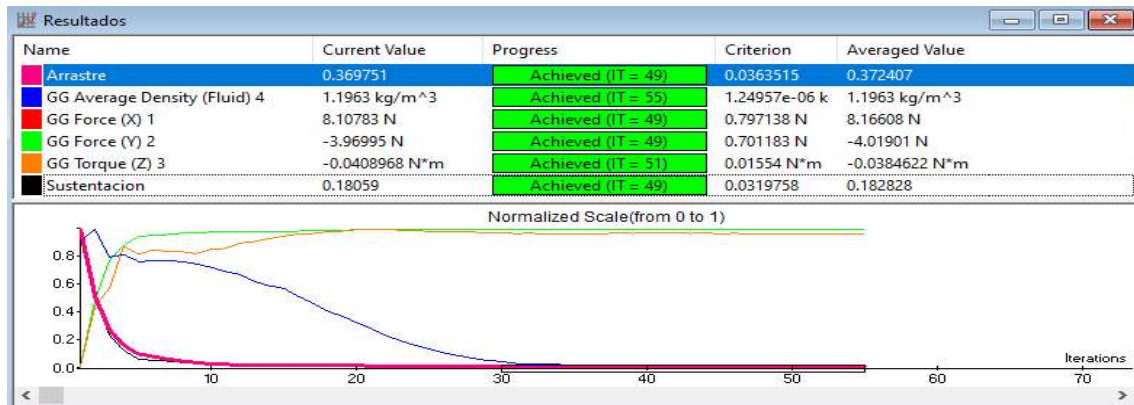
La filosofía de cálculo tomada consiste en obtener el empuje que producen las hélices en el eje X mediante el programa; para ello, se ha considerado la fuerza que se produce en la superficie de las mismas. Dicha fuerza en X se iguala a la fuerza de arrastre, de manera que la aceleración sea nula. En dicho caso, es posible conocer la velocidad del fluido, que ha de aproximarse a la velocidad introducida como dato para validar el resultado de la iteración. Esta velocidad se emplea para el cálculo de la fuerza de sustentación; para todo ello, se continúan empleando las ecuaciones (6) y (7). Calculando también la fuerza producida por las hélices en Y, y teniendo en cuenta de nuevo el empuje, se obtiene el peso que evita la aceleración vertical, que debe ser, al menos, mayor al peso de la aeronave.

Para simular la rotación de las hélices se crean volúmenes virtuales cilíndricos a su alrededor, dentro de los cuales, los elementos geométricos giran a la velocidad angular introducida. Teniendo en cuenta que los motores son de 1400KV y que el voltaje nominal de la batería es de 11.1V, se introducen velocidades de 15540rpm en sentido positivo y/o negativo según corresponda. En realidad, las hélices delanteras deberían girar ligeramente más rápido para compensar el torque generado por la sustentación; al menos para la configuración expuesta en el apartado 4.1.1. Pero si suponemos que este aumento de velocidad es proporcional a la disminución en las traseras y teniendo en cuenta la magnitud del momento, se acepta como válido considerar una velocidad de giro única.

Finalmente, el ángulo escogido es de 30°. Los coeficientes de arrastre y sustentación en esta situación se muestran en la Figura 24. A modo de comparación, se presentan también las Figuras 25 y 26, análogas a las Figuras 22 y 23; se ha tratado de mantener el rango de valores de la leyenda entre las Figuras 18, 22 y 25, a fin de presentar la información de manera más visual.

Figura 24

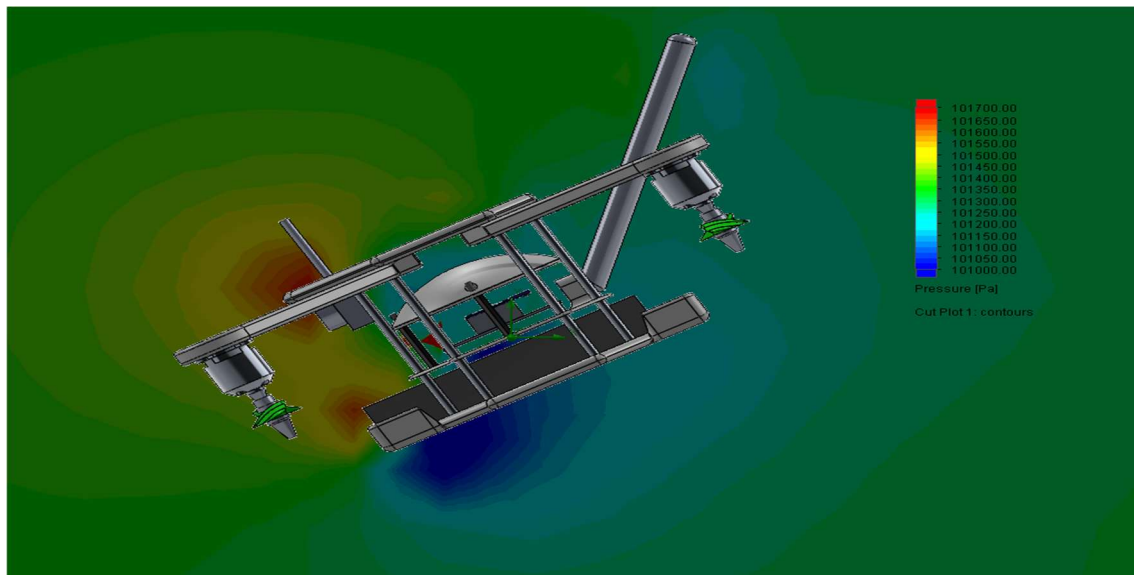
Arrastre y sustentación para un ángulo de ataque de 30°.



Nota. Fuente propia. Elaborada mediante el programa SolidWorks Flow Simulation. De nuevo se aprecia una disminución de los coeficientes de arrastre y sustentación, siendo $C_L \approx 0.181$ y $C_D \approx 0.37$; disminuciones porcentuales de 12.56% y 31.73% respectivamente, de 64.02% y 54.66% si se compara con los resultados iniciales. Se aprecia, además, que el par generado no solamente es mucho menor, sino de sentido contrario.

Figura 25

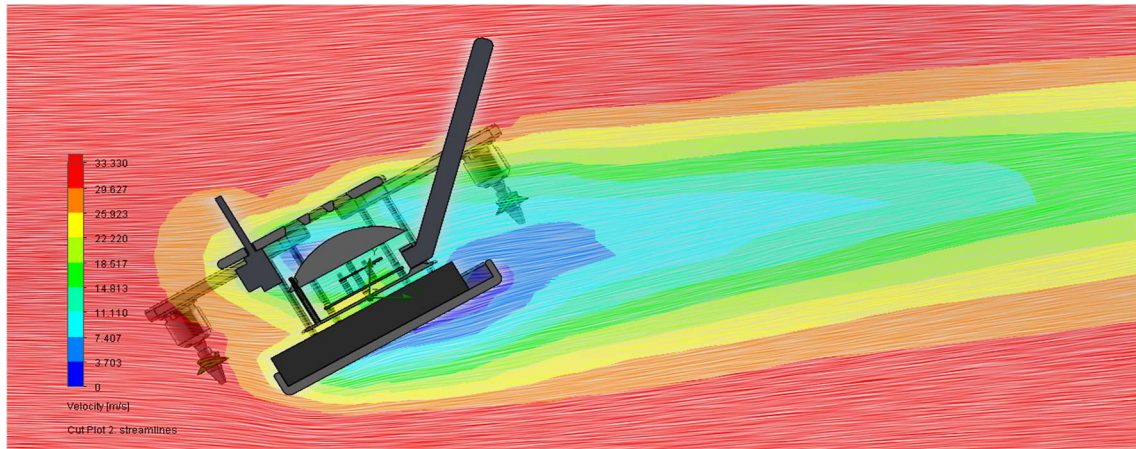
Presiones plano de simetría, arrastre y sustentación para un ángulo de ataque de 30°.



Nota. Fuente propia. Elaborada mediante el programa SolidWorks Flow Simulation. La distribución más uniforme de la presión superior podría explicar la disminución del momento producido sobre la aeronave. Se observa que los difusores presentan una eficiencia menor, esto puede ser debido al ángulo de entrada del aire (menos perpendicular) en los mismos.

Figura 26

Velocidades plano de simetría, arrastre y sustentación para un ángulo de ataque de 30°.

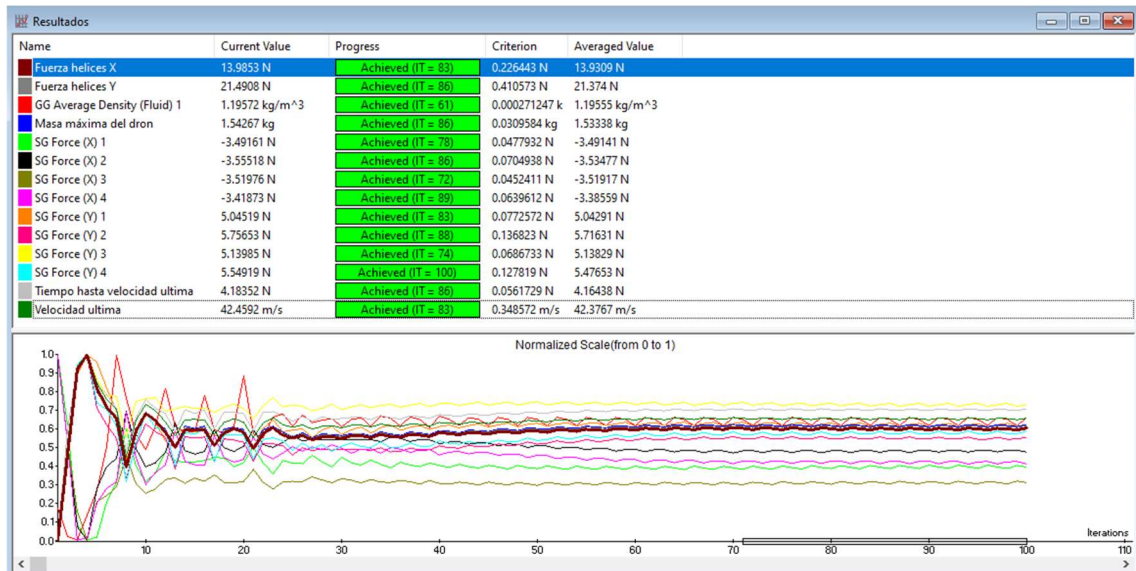


Nota. Fuente propia. Elaborada mediante el programa SolidWorks Flow Simulation. Se aprecia que el vórtice aguas debajo de la aeronave es prácticamente inexistente, la adherencia a la capa límite muy superior.

Para estas condiciones, se ejecuta un nuevo estudio para evaluar las variables de iteración. Los resultados finales se muestran en la Figura 27. Se muestran, nuevamente, las presiones y líneas de corriente en condiciones de funcionamiento en las Figuras 28 y 29. Adicionalmente, se incluye en la Figura 30 una vista de las líneas de corriente a través de las hélices, que justifican la no interferencia entre ambas.

Figura 27

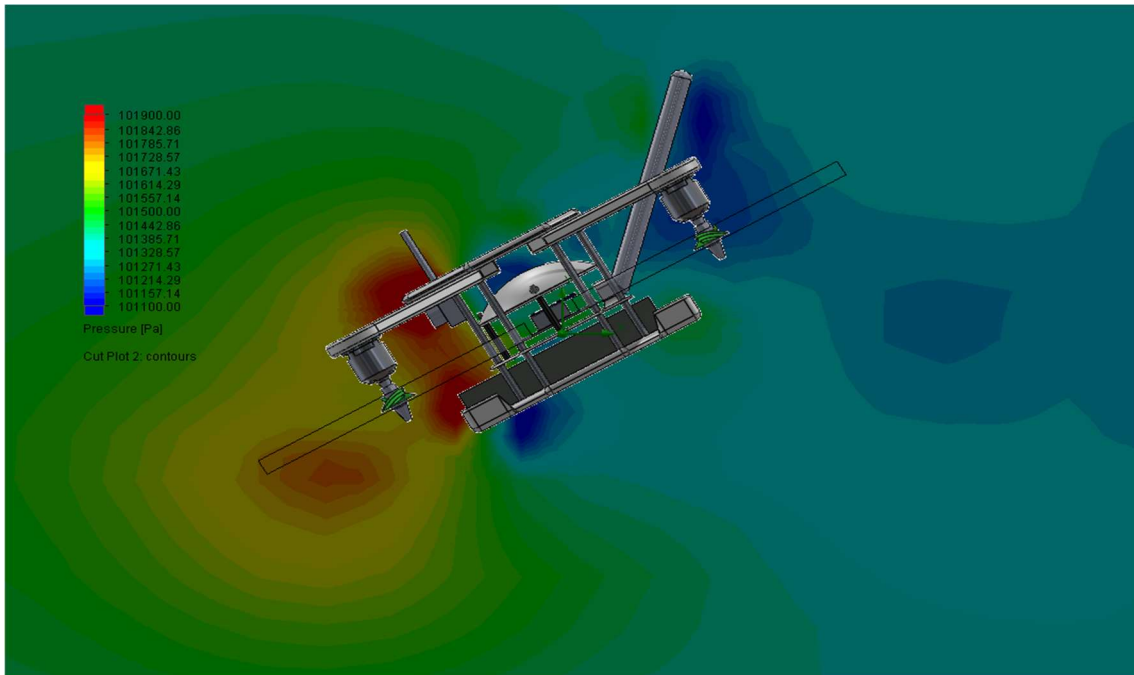
Velocidad terminal.



Nota. Fuente propia. Elaborada mediante el programa SolidWorks Flow Simulation. La velocidad terminal resultante es de 42.459m/s, frente a los 42.5m/s introducidos (error <0.1%), por lo que se considera válido el resultado; esta velocidad se alcanza, asumiendo aceleración constante, en 4.183s. En este supuesto, la masa de la aeronave puede alcanzar 1.543kg.

Figura 28

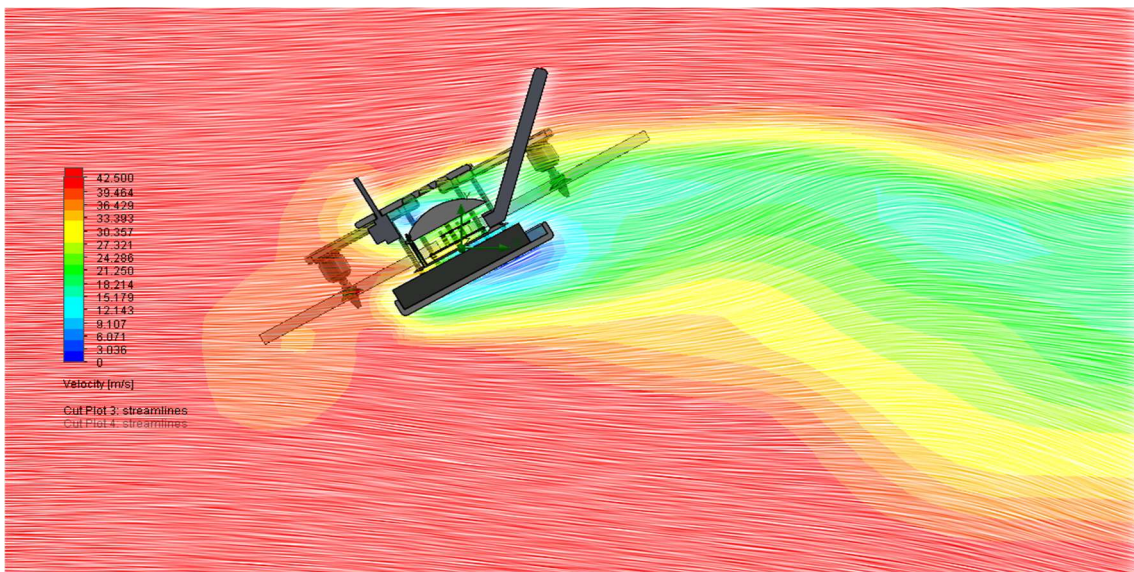
Presiones plano de simetría, velocidad terminal.



Nota. Fuente propia. Elaborada mediante el programa SolidWorks Flow Simulation. En contraposición a las figuras presentadas anteriormente, en esta imagen las presiones máximas se encuentran en la parte frontal de la aeronave, lo cual tiene sentido considerando la velocidad a la que se mueve. Las presiones mínimas se presentan de forma más dispersa, en general, aguas debajo del cuadricóptero.

Figura 29

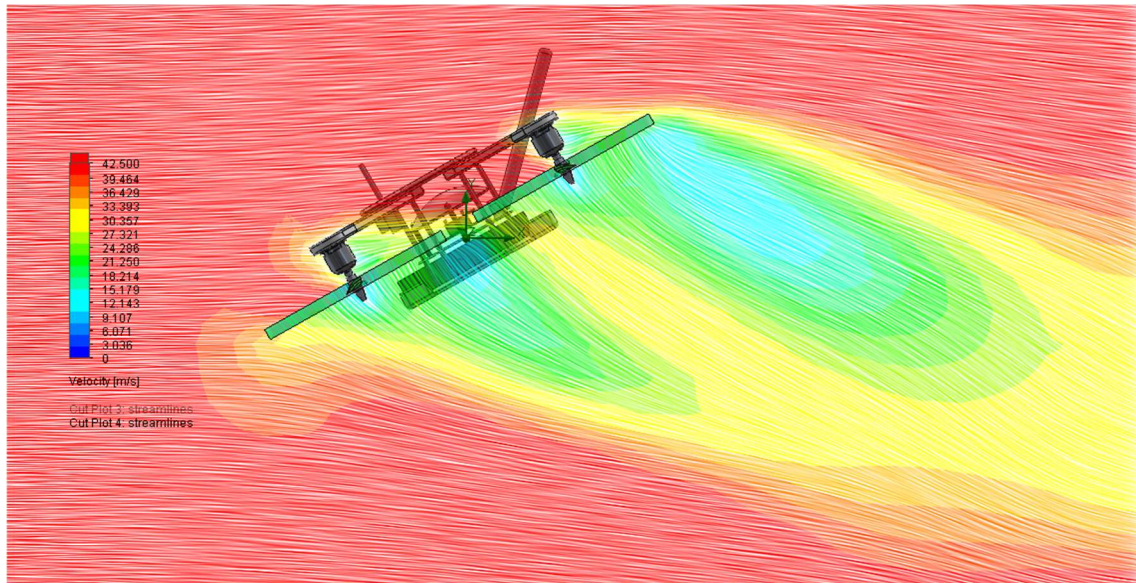
Velocidades plano de simetría, velocidad terminal.



Nota. Fuente propia. Elaborada mediante el programa SolidWorks Flow Simulation. Se aprecia la distorsión sobre la estela que producen las hélices; sin embargo, este hecho no parece afectar a la adherencia de la capa límite.

Figura 30

Líneas de corriente en las hélices.

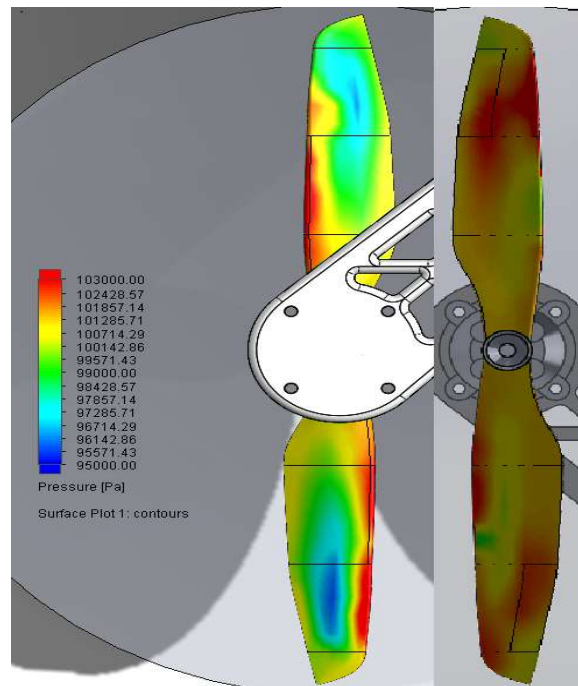


Nota. Fuente propia. Elaborada mediante el programa SolidWorks Flow Simulation.

También se ha considerado oportuno mostrar las presiones sobre la superficie de una de las hélices, a modo de justificación de las fuerzas que se generan en su seno, siguiendo el razonamiento explicado por la NASA en cuanto a propulsión con hélices se refiere (National Aeronautics and Space Administration [NASA], s.f.). En la Figura 31 se muestran dichas presiones.

Figura 31

Presiones sobre la superficie de las hélices.



Nota. Fuente propia. Elaborada mediante el programa SolidWorks.

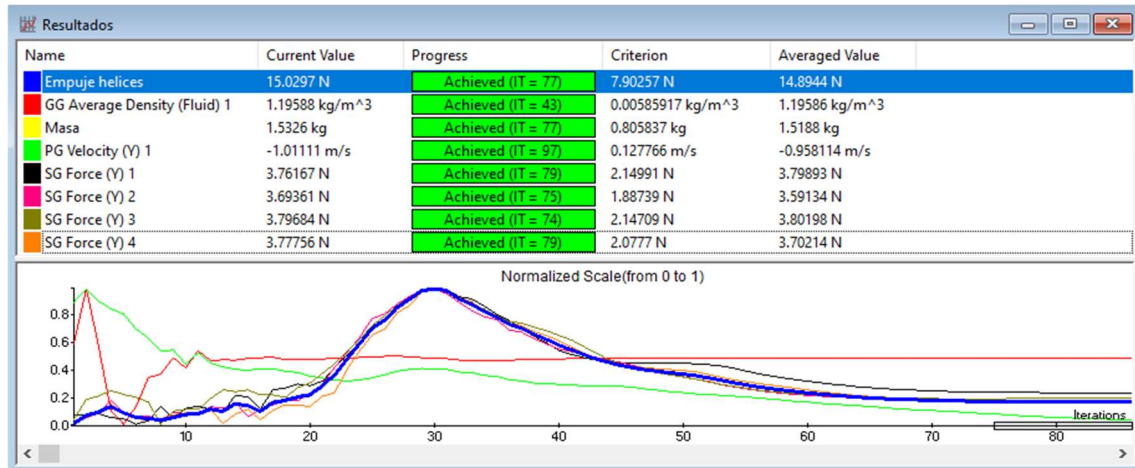
Obsérvese que la velocidad terminal se sitúa por encima de los 42m/s (unos 151 km/h), superando incluso los objetivos presentados en el apartado 2.3. A pesar de todo, para validar realmente estos estudios sería necesario un túnel de viento en el que probar el prototipo; esto, sin embargo, esta fuera de los límites del presupuesto.

4.1.3. Flotación

Conocida la masa máxima en condiciones de diseño es posible determinar la velocidad angular en las hélices que asegura la flotación. Para ello se ha de iterar de nuevo. En la Figura 32 se muestran los resultados para velocidades de giro, de 14000rpm en valor absoluto, lo que implica un voltaje de suministro de 10V en los motores. Puesto que este valor está por debajo del voltaje de seguridad de la batería (10.2V), se puede asegurar que el dron es capaz de mantenerse en el aire incluso con batería baja.

Figura 32

Flotación.



Nota. Fuente propia. Elaborada mediante el programa SolidWorks. En esta imagen se puede observar que, cuando la velocidad percibida por la cara superior de la aeronave es -1.011m/s, la masa máxima que eleva en estas condiciones es de 1.533kg; este valor es un 0.7% inferior a lo obtenido en el estudio anterior, sin embargo, en estas condiciones la nave aun asciende.

4.1.4. Material

Como ya se ha comentado en el apartado 2.2, se emplea la tecnología FFF para la conformación del prototipo. En este punto, cabe destacar que esto presenta dos inconvenientes principales: por una parte, las propiedades mecánicas, debido a que esta técnica de impresión genera materiales transversalmente isotrópicos; por otra, la rugosidad del acabado, mucho mayor a la considerada durante las simulaciones aerodinámicas. Por todo ello, si se deseara producir el objeto de proyecto, sería necesario buscar otras opciones en cuanto a material y/o proceso productivo. En este aspecto, una solución habitual es el uso de fibra de carbono.

En primer lugar, se estudia, como se ha dicho con anterioridad, el uso de materiales biodegradables: el Ácido Poliláctico (PLA) es un plástico con un uso muy extendido en la fabricación FFF por su facilidad de impresión. Su principal desventaja es su poca resistencia a condiciones ambientales y a la radiación solar. Es por ello que se considerarán otros termoplásticos no biodegradables.

El material escogido para el chasis de la aeronave es el Tereftalato de Polietileno Glicol (PETG), en la Tabla 5 se muestra la comparación respecto al PLA. Sin entrar en los detalles de la anisotropía, se puede comprobar que sus propiedades mecánicas son superiores a las del PLA. Su densidad es muy parecida a la fibra de carbono, lo que podría facilitar su adaptación en fases de madurez del proyecto posteriores; si bien es cierto que el PETG presenta especificaciones mecánicas muy inferiores a este, se realiza un estudio de esfuerzos para comprobar la viabilidad.

Tabla 5

Comparación entre los plásticos PLA y PETG.

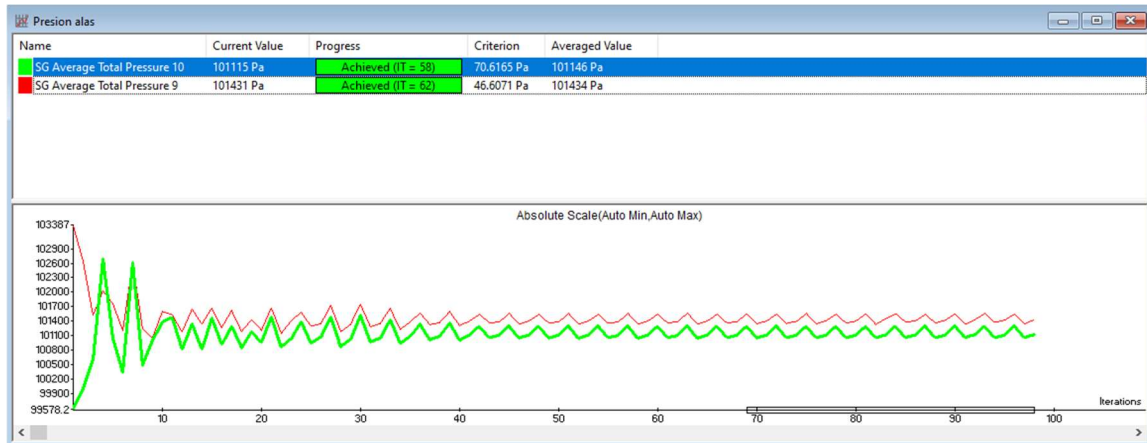
Propiedad	PLA	PETG
Modulo elástico (GPa)	0.05-13.8	1.1-20.3
Límite elástico (MPa)	8-103	28.3-101
Conductividad térmica (W/m-K)	0.032-0.17	0.162-0.225
Densidad (g/cm ²)	1.00-3.41	1.18-1.37
Precio (€/kg)	20.99	22.99

Los elementos del chasis sometidos a mayores esfuerzos mecánicos son las alas, por lo que serán el objeto de este estudio. El material se introduce en el programa como un plástico personalizado, ortotrópico de comportamiento lineal; los parámetros de referencia han sido extraídos de Szykiedansy et al. (2017) y Quevedo y Bonilla (2021). Con todo ello, falta por establecer las condiciones del entorno.

En condiciones de diseño, la resultante de la fuerza provocada por una hélice en la superficie del ala es, mediante la media de los módulos de las fuerzas mostradas en la Figura 24, del orden de 6.35N. Si a este valor se le resta el peso propio del motor, se obtiene algo menos de 5.5N; se toma esta fuerza como referencia para el cálculo. Además, en este supuesto, existe una diferencia de presión entre ambas caras del ala. Se ha obtenido la presión media sobre cada superficie como un resultado adicional del estudio en velocidad terminal; los resultados se muestran en la Figura 33.

Figura 33

Presiones medias sobre las superficies de las alas.

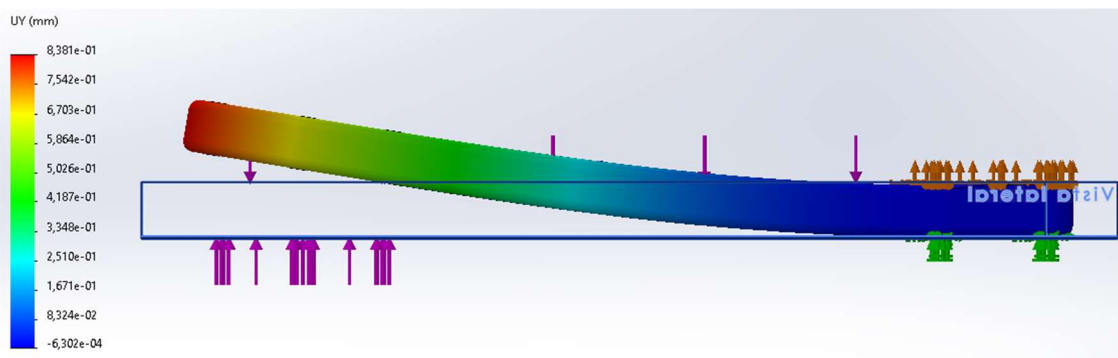


Nota. Fuente propia. Elaborada mediante el programa SolidWorks Flow Simulation. La presión media en la superficie superior es de 101431Pa, mientras que en la cara inferior es de 101115Pa.

Tomando el incremento de presiones y multiplicando por el área del ala se obtiene una fuerza algo mayor de 1 N operando perpendicularmente a su superficie, que también se incluye en el cálculo. La deformación elástica que experimenta el ala en tal situación se muestra en la Figura 34; sería necesario comprobar experimentalmente si este hecho tiene repercusiones en la estabilidad de vuelo, pero a falta de más datos y dado que no se ha tenido en cuenta el motor como elemento rigidizante, se va a considerar que es aceptable.

Figura 34

Desplazamientos en la dirección normal a la superficie del ala.



Nota. Fuente propia. Elaborada mediante el programa SolidWorks. La imagen muestra la deformada de forma exagerada para facilitar la comprensión del desplazamiento.

Considerando lo expuesto, el nuevo peso de la aeronave puede obtenerse de la misma forma que en el apartado 3.1.6; el peso de la batería queda expuesto en el apartado 3.1.7 y el de los motores en el apartado 3.1.6, el de la PCB y los componentes es ahora conocido: 20g.

Por último, en cuanto a los componentes estructurales, el conjunto de piezas diseñadas computa un total de 482.571cm^3 . Sin embargo, por ser un elemento que no va a ser sometido a esfuerzos mecánicos, la cobertura de la electrónica (85.354cm^3) se va a considerar que está hecha de Acrilonitrilo Butadieno Estireno (ABS por sus siglas en inglés). Empleando técnicas de moldeo rotacional podría producirse de forma hueca, y por tanto reducir el peso de la misma, pero se va a suponer que pesa 85g por ser la densidad del ABS cercana a la unidad.

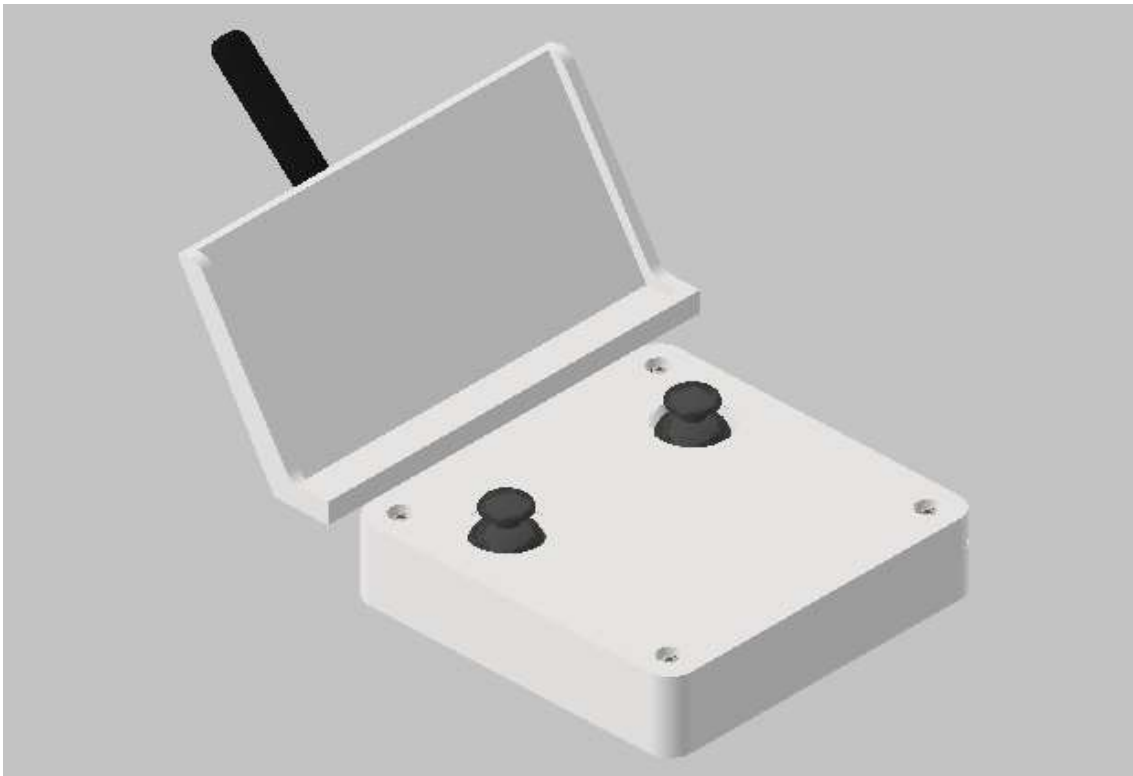
El nuevo volumen asciende a 397.217cm^3 . Multiplicando esta cantidad por la densidad de la fibra de carbono (1.75g/cm^3), por ser el material elegido para el producto final, se obtienen 695g. También se han empleado varillas de aluminio, que añaden otros 30g. Con todo ello, el peso de la aeronave, entendida como objeto del proyecto, presenta un peso de 1.52kg. Con respecto al peso nominal (el de velocidad terminal), esto deja un margen de 20g aproximadamente.

4.2. CARCASA DEL MANDO

Respecto al mando, a falta de requerimientos estructurales, se diseña de manera que sea cómodo a las manos, y lo suficientemente grande para albergar los componentes. En este caso, sí se emplea como material el PLA, una opción biodegradable. El mando incorporará también un elemento para sujetar el teléfono móvil; en la Figura 35 se muestra el ensamblaje, el motivo por el cual se ha desarrollado en Inventor se comenta en el apartado 6.1.2.

Figura 35

Diseño propuesto para la aeronave.



Nota. Fuente propia. Elaborada mediante el programa Inventor.

CAPÍTULO 5. FABRICACIÓN

5.1. IMPRESIÓN 3D

Para la impresión de las piezas se ha empleado un robot cartesiano basado en la máquina comercial Ender 3X a la que se le han realizado una serie de modificaciones a fin de mejorar la calidad de impresión. Dichas modificaciones incluyen:

- Sustitución de la placa controladora por una MKS Monster8 de 32 bits sobre la que se ha instalado el firmware Marlin 2.1.2.1.
- Drivers de los motores (incluyendo el extrusor) actualizados a TMC2290.
- Añadido un segundo motor en el eje Z, sincronizado mediante banda y tensor.
- Añadido a la cama caliente una lámina magnética metálica con superficie de Polieterimida (PEI) en forma de recubrimiento en polvo
- Sustitución del *hot-end* por una Micro-Swis.
- Sustitución del extrusor por un BMG Trianglelab.
- Sustitución del *nozzle* por uno de cobre con punta de rubí.
- Añadidos relés de estado sólido en cama caliente y en *hot-end*.
- Sustitución de la conexión entre el extrusor y el *hot-end* por un tubo de Politetrafluoroetileno (PTFE) de la marca Capricorn.
- Añadido un sensor BLTouch para la auto-nivelación.
- Añadido el sistema operativo Octoprint para controlar la impresora mediante una Raspberry Pi 3 B.

Cabe destacar, también, que la impresora se encuentra en una superficie atornillada directamente a la pared y sobre una alfombra de 1cm de espesor de caucho anti vibrante.

En los siguientes subapartados se especifican los filamentos empleados y los distintos parámetros de configuración necesarios para la impresión. Sin embargo, puesto que la boquilla del *nozzle* es de 0.4mm, la altura de capa no superará los 0.32mm para evitar problemas en la impresión; en adelante, se emplea una altura de capa de 0.2mm.

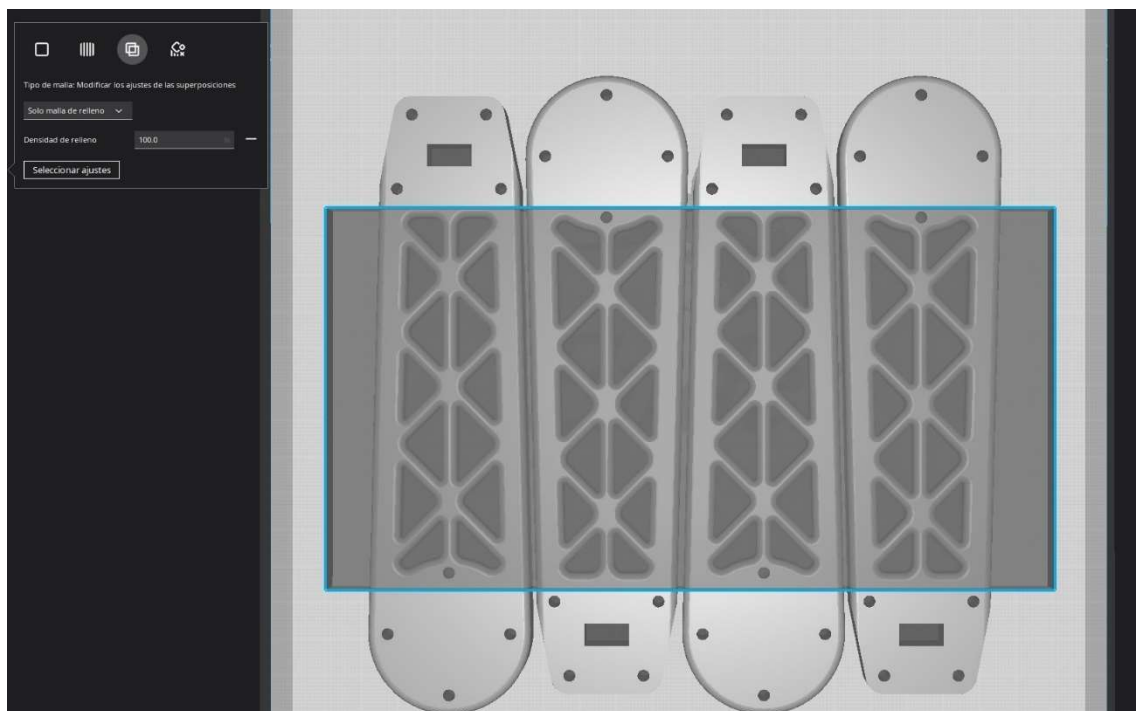
5.1.1. Aeronave

En lo referente a la impresión de las piezas del dron, se ha empleado el filamento PETG de Amazon Basics; se ha seleccionado el color negro por ser el más parecido al propio de la fibra de carbono. Dada la configuración de la impresora y el material, la temperatura de impresión ajustada, en base a la experiencia, es de 235°C. La cama caliente se configura para una temperatura de 75°C a fin de evitar tanto el fenómeno de *warping* (común para este material), como la transición vítrea (que se da a los 85°C).

Por parte del relleno, se ha escogido el patrón en giroide por presentar una ratio de resistencia mecánica por peso muy elevada. El relleno se ha seleccionado en un 20% general a fin de obtener un peso reducido, mientras que, para las piezas denominadas como alas, se ha creado un volumen auxiliar en el que la impresión se realiza de forma maciza. Dicho volumen se ha delimitado en función de los esfuerzos máximos observados en el estudio del material del apartado 4.1.4, esto es, en las zonas que presentan los huecos. En la Figura 36 se puede apreciar el volumen comentado, así como la característica del relleno modificada.

Figura 36

Configuración especial de la impresión de las alas.

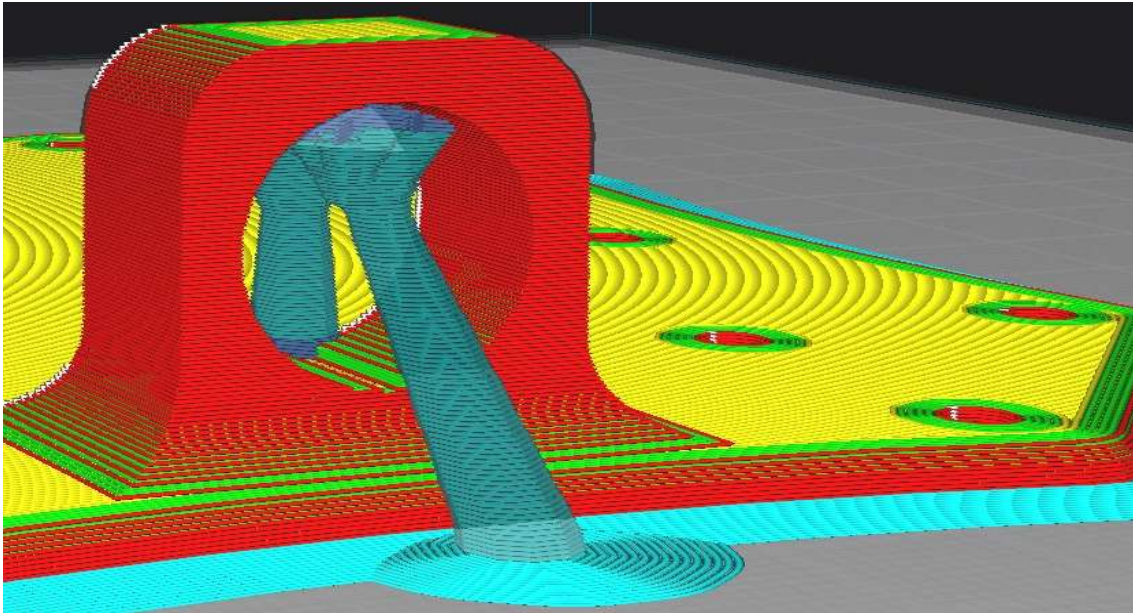


Nota. Fuente propia. Elaborada mediante el programa Ultimaker Cura 5.0.

Solo se han empleado soportes en la pieza correspondiente al soporte de la electrónica puesto que presenta un agujero transversal; el resto de las piezas no tienen puentes ni partes huérfanas; los soportes se han seleccionado en forma de árbol, como se observan en la Figura 37. Tampoco se ha empleado refrigeración por no considerarse necesaria; ya que las superficies de impresión son considerablemente grandes en cada remesa y a fin de evitar un endurecimiento innecesario del material en detrimento de su tenacidad, es conveniente evitar corrientes de aire directamente sobre las piezas. El PETG es propenso al fenómeno de delaminación entre capas, o *cracking*, producido por tensiones internas.

Figura 37

Soporte en árbol para agujero transversal.



Nota. Fuente propia. Elaborada mediante el programa Ultimaker Cura 5.0.

Respecto a las tolerancias, no existe mayor restricción que la dimensión de los agujeros. Puesto que todos ellos son pasantes, no roscados, se ha decidido aumentar su dimensión en 0.4mm para que presenten una holgura suficiente para el tornillo. En cualquier caso, una vez impresas las piezas se emplea una herramienta de desbarbado para asegurar que el tornillo no tenga problemas de ajuste.

Las paredes se seleccionan de manera que existan 4 filas de material; con una fila adicional alternante, que se superpone con el relleno, para aumentar la rigidez. El grosor de las superficies superior e inferior se establece en 1mm, múltiplo de la altura de capa, de manera que están conformadas por 5 capas macizas consecutivas.

Con todo ello, la velocidad de impresión se establece como un acuerdo entre el tiempo de impresión y la calidad; se deben tener en cuenta, además, las limitaciones propias de la impresora. En la Figura 38 se muestra la configuración general descrita; mientras que en la Tabla 6 se especifican los tiempos de impresión para cada remesa de piezas.

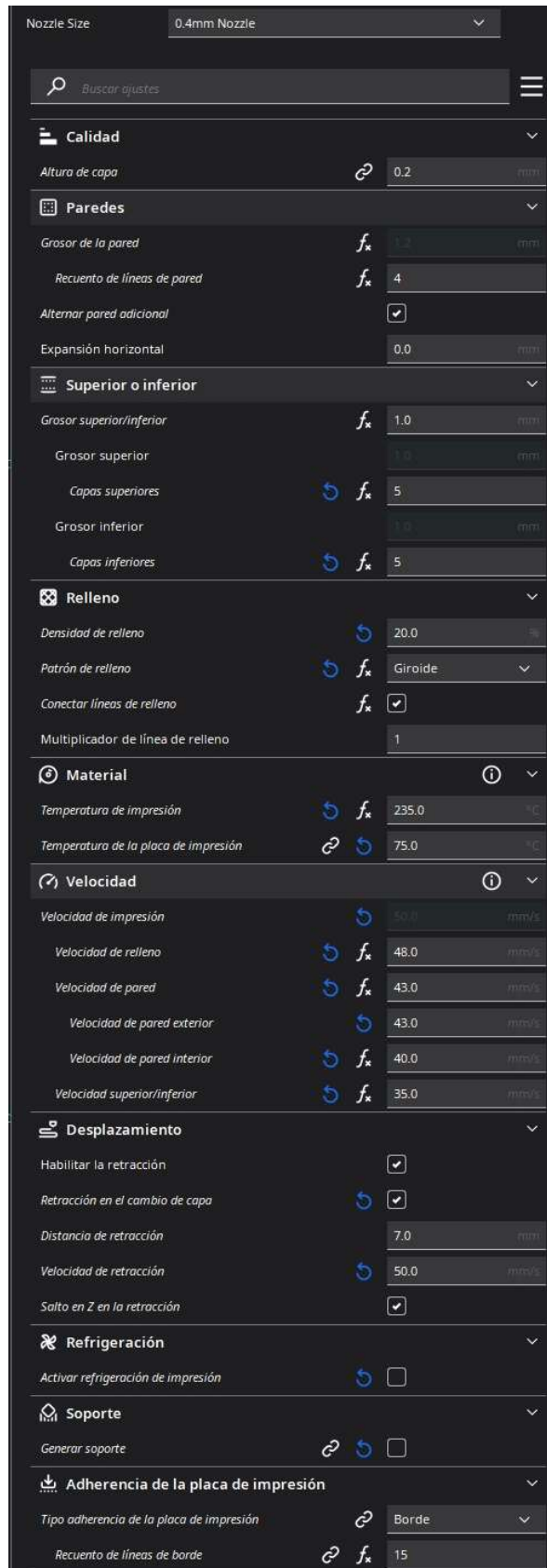
Tabla 6

Tiempos de impresión de la aeronave.

Piezas que componen la remesa	Tiempo de impresión
Cuatro alas	18h 40min
Plancha de soporte para la batería	6h 38min
Cobertura de la electrónica y marco para las alas	8h 54min
Plancha de soporte para la electrónica	3h 14min

Figura 38

Configuración general de la impresión de la aeronave.



Nota. Fuente propia. Elaborada mediante el programa Ultimaker Cura 5.0.

5.1.2. Mando

En cuanto a la impresión de las piezas del mando, el filamento empleado es el PLA de SUNLU; el color se ha seleccionado en negro para ser parejo a la aeronave. Para este material, la temperatura de impresión se ha establecido en 215°C. Ahora bien, la temperatura de la cama caliente se ha establecido en 65°C, la cual se encuentra por encima de la de transición vítrea del material (60°C). En el caso del PLA, es una buena práctica establecer la temperatura de la cama caliente entre 60 y 70°C para evitar el *warping* (Spoerk et al., 2017); concretamente, se recomienda aumentarla por encima de 60°C en caso de imprimir piezas grandes. A pesar de todo, se ha de encontrar una temperatura de acuerdo mediante la experimentación, ya que una temperatura demasiado alta puede provocar el fenómeno conocido como “pata de elefante”.

En lo referente al relleno se ha establecido con forma de rejilla con el fin de reducir los tiempos de impresión; si bien, no es el tipo de relleno más rápido de imprimir, presenta una buena relación velocidad-resistencia de la pieza. De nuevo, se han empleado paredes alternantes, de manera que estas están formadas por cuatro filas y una extra que se superpone con el relleno una capa sí, otra no. Las superficies superior e inferior se han establecido de la misma forma que en la impresión de las piezas de la aeronave.

En este caso sí se ha configurado el ventilador durante la impresión, ya que el PLA es menos propenso al *cracking* que el PETG. Respecto a la velocidad de impresión, se ha empleado la misma que para la aeronave, en la Tabla 7 se especifican los tiempos de impresión. La configuración empleada para la impresión de todas las piezas del mando se muestra en la Figura 39. Obsérvese que solamente se ha incluido aquella que difiere de la presentada en la Figura 38; también se ha omitido la información respectiva a los soportes, que se comenta ac continuación.

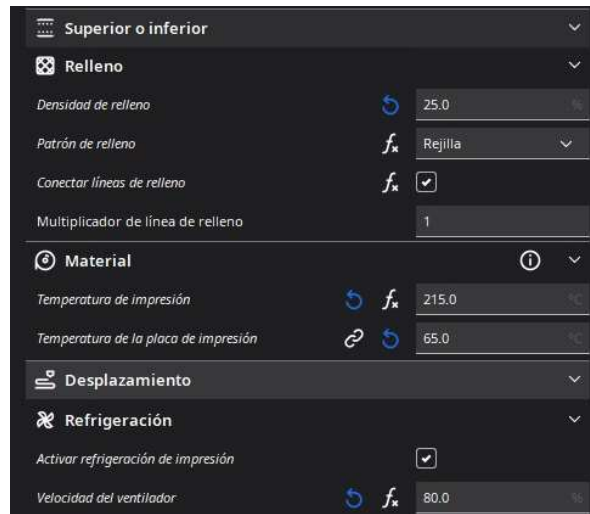
Tabla 7

Tiempos de impresión del mando.

Piezas que componen la remesa	Tiempo de impresión
Base del mando	13h 46min
Cobertura del mando	8h
Soporte para el teléfono móvil	14h 1min

Figura 39

Configuración de los soportes de la impresión del mando.



Nota. Fuente propia. Elaborada mediante el programa Ultimaker Cura 5.0.

Por parte de los soportes, son necesarios en todas las piezas que componen el mando. Las piezas realizadas con PLA son sencillas de separar de sus soportes, por lo que no se ha considerado necesario tomar precauciones adicionales. En cualquier caso, al estar presentes en todas las piezas (en algunas, como el soporte para el teléfono móvil, de manera abundante), se ha considerado conveniente presentar sus parámetros de configuración; en la Figura 40 se muestra esta información.

Figura 40

Configuración de la impresión del mando.



Nota. Fuente propia. Elaborada mediante el programa Ultimaker Cura 5.0.

5.2. CALIBRADO DE LAS HÉLICES

Se ha decidido incluir en el capítulo de fabricación los procesos manuales que se han de llevar a cabo para el calibrado de las hélices. Con objeto de reducir las vibraciones provocadas por el conjunto hélices-motor el método más extendido es la calibración mediante equilibrador magnético. Una vez montada la hélice, tal y como se muestra en la Figura 41, se procede a añadir o sustraer material de la misma hasta que se obtiene el equilibrio entre las palas.

Figura 41

Montaje para la calibración manual de las hélices.



Nota. Fuente propia. Hélice 8045 montada sobre equilibrador magnético.

En su forma sustractiva, se ha de tener cuidado de no eliminar material en exceso, ya que la única manera de compensarlo es repetir el proceso con la otra pala. En este caso, la resistencia de la hélice se deteriora de manera permanente haciéndola más susceptible a la rotura.

Por otra parte, en el método aditivo, se emplean adhesivos para añadir peso en una de las hélices; esta manera, aunque reversible, añade peso a la hélice y puede provocar un desbalance en la misma durante la operación de vuelo si el material añadido no queda lo suficientemente fijado. Se ha procedido con este último empleando pinta uñas y acetona de manera que se puedan mitigar sus inconvenientes.

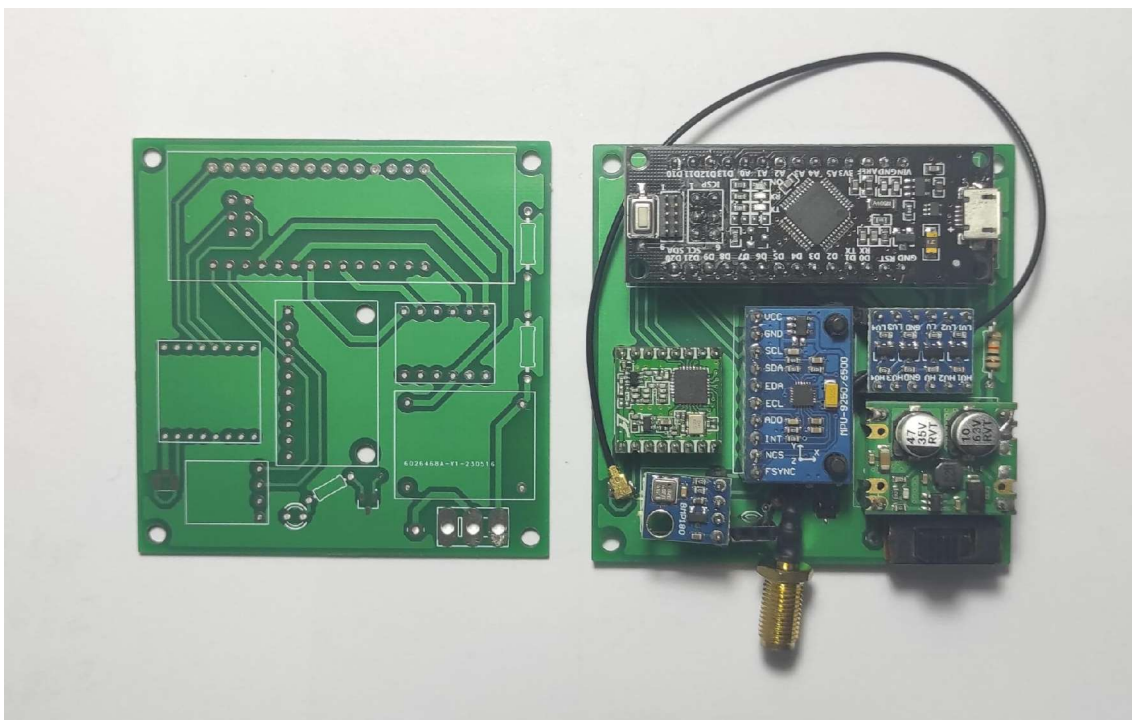
5.3. ELECTRÓNICA

La electrónica se ha soldado en primera instancia a conectores Dupont, macho y/o hembra, a fin de facilitar el reemplazo en caso de fallo de alguno de los componentes. Si bien este método de conexión ocupa un volumen superior, se ha considerado más coherente en cuanto a los objetivos que debe cumplir un prototipo. Existen, no obstante, algunos componentes soldados directamente a la placa, como las resistencias o los componentes *Surface-Mounted Device* (SMD).

El equipo empleado para realizar esta labor está compuesto por un soldador JBC de 25W con punta R05D y una estación de trabajo de soldadura de NEWACALOX. Se ha empleado alambre de soldadura (Sn99.3%, Colofonia 2% y Cu0.7%) para los elementos montados en agujero y pasta de soldadura (Sn42% y Bi58%) para los componentes SMD. En las Figuras 42 y 43 se muestran el antes y el después de las PCB de la aeronave y el mando respectivamente; las PCB han sido fabricadas por la empresa JLCPCB.

Figura 42

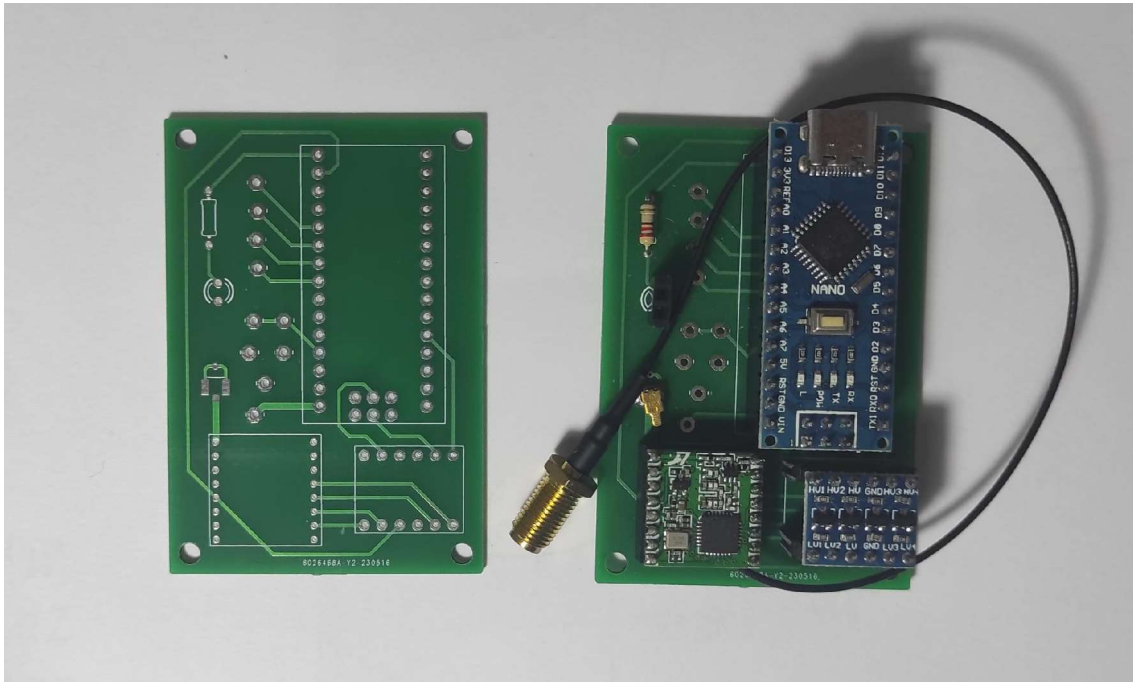
PCB de la aeronave antes y después de soldar.



Nota. Fuente propia. Aunque no se aprecia en la imagen, la unión de los componentes a la placa se realiza mediante conectores Dupont. La alimentación se conecta mediante el *pad* libre de la placa (positivo) y una de las conexiones del regulador de tensión (negativo).

Figura 43

PCB del mando antes y después de soldar.



Nota. Fuente propia. Los *pads* libres que se aprecian en la imagen se corresponden con conexiones por cable al LED, los *joysticks* y la alimentación.

5.4. MONTAJE

El montaje de los elementos se desarrolla en base a las referencias especificadas en los planos de los ensamblajes respectivos. Se describe el montaje del prototipo, por lo que es posible que exista disonancia entre las partes que lo componen y lo observado en los estudios aerodinámicos del apartado 4.1. Cabe destacar que los elementos de la electrónica acoplados a las PCB se presuponen parte intrínseca de las mismas, por lo que no se han tenido en consideración en las instrucciones. En el Anejo I se detallan dichas instrucciones. En las Figuras 44 y 45 presentadas a continuación, se muestran, respectivamente, el cuadricóptero y el mando completamente montados.

Figura 44

Aeronave completamente montada.



Nota. Fuente propia. Fotografía realizada tras el ensamblaje.

Figura 45

Mando completamente montado.



Nota. Fuente propia. Fotografía realizada tras el ensamblaje.

CAPÍTULO 6. CONCLUSIONES

6.1. ADVERSIDADES DEL PROYECTO

En el transcurso de cualquier proyecto es inevitable encontrarse con desafíos y obstáculos que ponen a prueba la planificación, la ejecución y la resiliencia del equipo involucrado. En este apartado, se abordarán los diversos problemas que se han presentado a lo largo del presente proyecto. En el apartado 6.2 se profundiza en las mejoras y soluciones para aquellos puntos que no se han podido resolver durante el transcurso del proyecto.

6.1.1. Prototipo alfa

En la evolución del proyecto, el prototipo alfa se erige como una etapa crucial que proporciona valiosas lecciones y perspectivas. A partir de este, se han llevado a cabo mejoras significativas en cuanto al diseño estructural, la selección de componentes y el diseño y producción de las PCB; dichas mejoras se recogen en el Anejo II.

Durante la producción de esta primera versión del prototipo, el microcontrolador (SAM D21 M0-mini) resultó dañado debido a un error humano. Se asumió incorrectamente que la placa controladora podía ser conectada directamente a la batería, provocando el fallo del regulador de tensión interno. A pesar de este hecho, la experiencia en proyectos anteriores había propiciado la compra de un microcontrolador extra con motivo, precisamente, de mitigar este tipo de inconvenientes.

Ambas placas controladoras habían sido testadas al momento de su recepción; sin embargo, en el momento de cargar el código en el microcontrolador sustituto, se observó que este se apagaba. Puesto que el mismo código ya había sido ejecutado en la otra unidad de control, la explicación más plausible ante este fenómeno fue el fallo del propio *hardware*.

Dada la situación y el tiempo que se tardaba en recibir de nuevo el componente, se estimó como inviable presentar el proyecto en la convocatoria planificada. Puesto que esto implicaba el pago de una tasa administrativa para poder defender el año siguiente, se decidió emplear el tiempo en investigar de forma más profunda los requisitos y soluciones; así nace la versión actual del prototipo, sobre la que se cimienta la memoria presentada.

6.1.2. Incompatibilidad y errores en el *software* de diseño

Tanto en la versión alfa como en la actual, el hecho de emplear el *software* Solidworks como herramienta de diseño ha supuesto un consumo extra del tiempo debido a los requerimientos de formato de los planos.

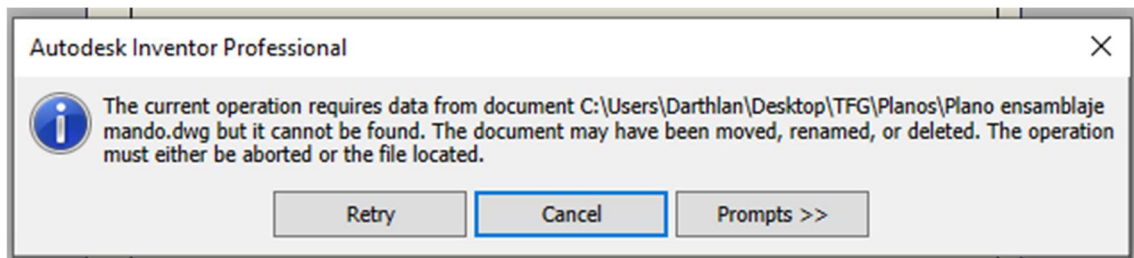
Debido a que la plantilla proporcionada para la creación de los planos estaba en formato dwg, para editarla era necesario el uso de AutoCAD. Sin embargo, SolidWorks no era capaz de reconocer las modificaciones realizadas sobre el plano, abriendo el archivo como un documento en blanco. Para poder presentar los planos en el formato correcto, ha sido necesario exportar tanto las partes como el ensamblaje de la aeronave a un formato step; a partir de este se han importado a Inventor, que sí leía correctamente las plantillas.

Durante la realización del prototipo alfa, el procedimiento seguido para el mando también fue este ya que se desconocía esta problemática. Se planteó usar un *software* de CFD diferente, concretamente ANSYS CFX, para la versión actual ya que en cualquier caso la geometría debía ser construida en Inventor; pero dadas las pocas diferencias a nivel de resultados (Ramlan et al., 2020) y el hecho de haber aprendido ya a configurar SolidWorks Flow Simulation, se acabó decidiendo mantener el mismo programa de simulación.

Finalmente, es necesario comentar otra adversidad relacionada con este tema: un error producido al guardar los planos de Inventor; concretamente, al guardar el segundo plano. Dicho error, presentado en la Figura 46, solo se ha podido salvaguardar desinstalando el programa y volviendo a instalarlo de nuevo cada vez que se completaba un plano. Esta situación, para la cual la comunidad de Autodesk no ha encontrado solución, ha llevado a aumentar significativamente las horas de un proceso por lo general sencillo.

Figura 46

Error de Inventor.



Nota. Fuente propia. Capturado del mensaje de error que ofrece Inventor al guardar el archivo, que se encuentra en la ruta mostrada.

6.1.3. Inviabilidad de la unidad de control seleccionada

Puesto que la primera placa controladora probada presentó un correcto funcionamiento, se asumió que el fallo de la segunda constituía un caso aislado. Debido a esto, al igual que durante el desarrollo del prototipo alfa, se han vuelto a comprar dos microcontroladores (SAMD21 M0-mini). Como buena práctica, ambos fueron comprobados en el momento de la recepción.

El problema reside en que, pasado el tiempo, ambas unidades de control presentan problemas con algunos de sus pines. Si bien esto no debería suponer un problema *per se* ya que ambas cuentan con pines no utilizados, este hecho presenta dos inconvenientes: uno a nivel de prototipo, otro a nivel de solución viable.

En el prototipo, supone que la PCB diseñada y soldada específicamente para esta configuración de conexión no se puede emplear. Si bien podría diseñarse una nueva PCB, con sus costes de manufactura asociados, esto solo resuelve el problema a corto plazo. De cara al objeto de proyecto, y teniendo en cuenta que tres de cuatro unidades eran defectuosas, no se puede garantizar la fiabilidad de la placa controladora; por lo cual, se ha descartado su viabilidad.

Debido a este inconveniente, es necesario realizar una revisión de la situación actual del proyecto. Se deberán realizar cambios en la selección de componentes que podrían comprometer el conjunto del diseño. Puesto que estos cambios requieren de estudio y trabajo para implementarse, así como de tiempo para recibir los nuevos componentes adoptados, se decide concluir esta versión del prototipo como una beta.

6.2. REVISIÓN DE DISEÑO

Debido a las limitaciones detectadas en el hardware existente, se presenta una propuesta alternativa para la placa controladora actual. Este proceso de revisión de diseño no solo implica la selección de un nuevo componente, sino que también conlleva una serie de cambios sobre las soluciones adoptadas y la oportunidad de optimizar algunos aspectos. En este sentido, algunas ideas que no se habían implantado por no ser suficientemente importantes y por haber sido concebidas tras la resolución de los apartados que comprometen, tendrán cabida en este proceso de adaptación.

El componente que se ha decidido emplear para sustituir el microcontrolador es el Tensilica Xtensa LX6, integrado en la placa ESP32-WROOM-32D. Esta unidad de control no se había considerado en un principio por desconocimiento de la misma. En la Tabla 8 se muestra una comparación con la SAMD21 M0-mini.

Tabla 8

Comparación entre microcontroladores

Especificación	ESP32-WROOM-32D	SAMD21 M0-Mini
Arquitectura	RISC-V	ARM Cortex M0+
Número de núcleos	2	1
Frecuencia del reloj	240MHz	48MHz
RAM	520kB	32kB
Memoria <i>flash</i>	448kB internos	256kB
PWM (pines y resolución)	28 pines 15/16bits	12 pines de 10bits
Dimensiones	54x28mm	54x20mm
Precio	11.99€	23.95€

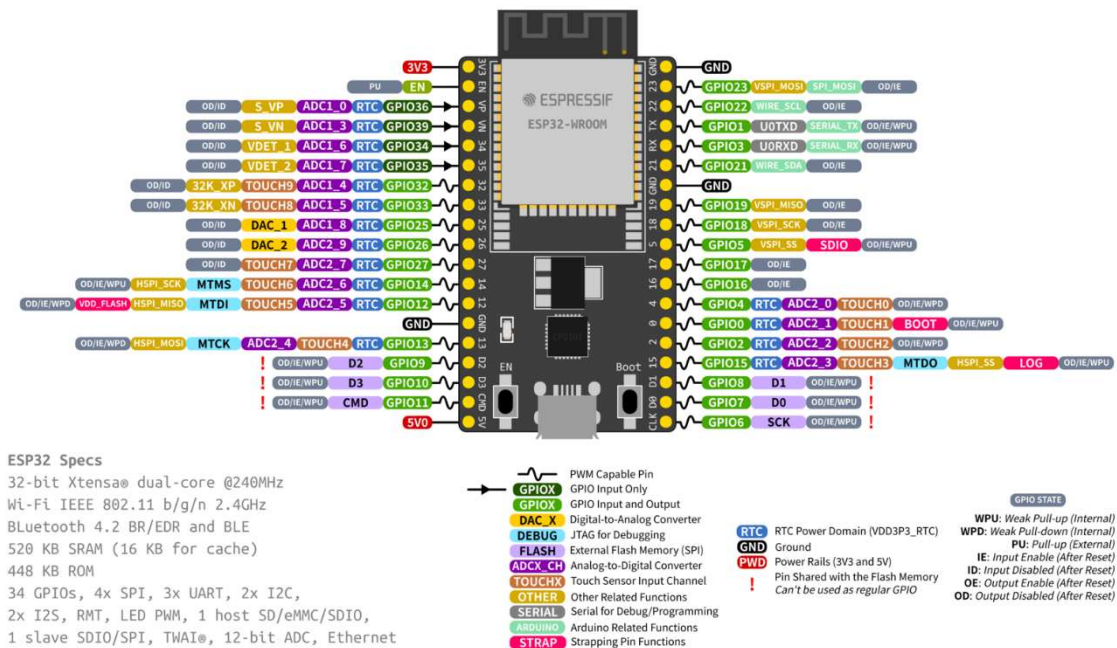
Es conveniente hacer algunas puntualizaciones sobre lo expuesto. Lo primero, que la ESP32-WROOM-32D cumple todos los requisitos planteados en el apartado 3.1.1; lo segundo, que para poder aprovechar completamente la ventaja que ofrece el uso de dos núcleos, es necesario emplear un *framework* para que el microcontrolador trabaje en tiempo real.

Por último, que el motivo de que casi todos los pines de la ESP32-WROOM-32D permitan el uso de PWM es debido a que sus pines son configurables, pudiendo ejercer varias funciones incluso un mismo pin. Además, el hecho de que se presenten dos resoluciones es debido a las posibilidades de configuración de las unidades PWM: dos unidades de 16bits cuando se trabaja expresamente con motores, pero solo dos motores al mismo tiempo como máximo; una unidad de 15bits para cualquier otra aplicación. En el caso del presente proyecto se emplea la segunda.

También cabe decir que el microcontrolador seleccionado muestra una velocidad del reloj y RAM muy superiores a las del anterior, todo ello por un precio menor. La Figura 47 se muestra la asignación de pines de la placa.

Figura 47

Asignación de pines de la placa ESP32-WROOM-32D.



Nota. Adaptado de *Pin Layout*, de ESPRESSIF, 2023, <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-devkitc.html>.

En adelante se presentan los cambios necesarios para poder implementar esta placa controladora al presente proyecto, así como la propuesta del software y otros cambios menores que, por un coste relativamente bajo, mejoran la calidad global del prototipo y del producto.

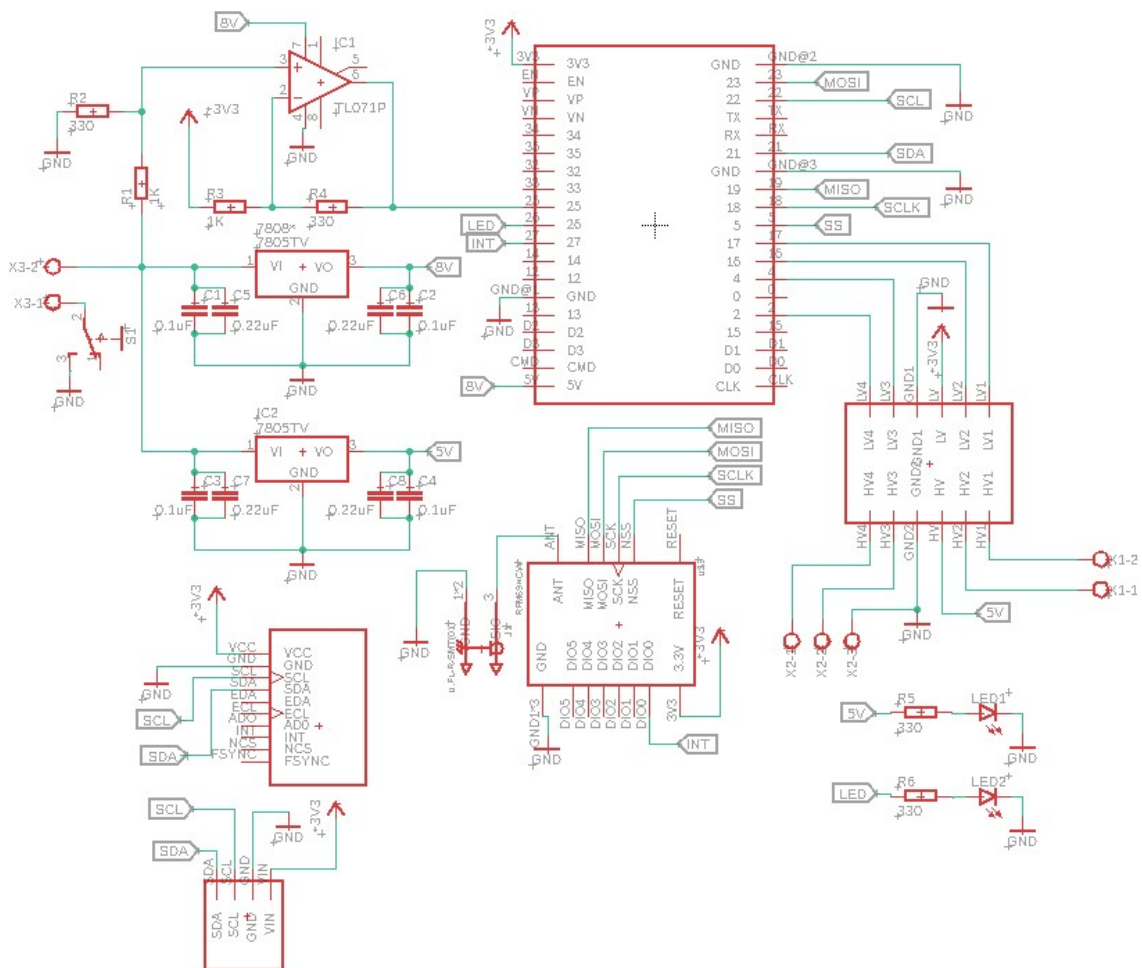
6.2.1. Cambios en la selección de componentes

Debido al uso de la nueva unidad de control, la alimentación mediante un regulador de tensión cuya salida es de 5V queda ahora en el límite del voltaje de alimentación recomendado (5-12V). Es por ello por lo que se ha decidido sustituir el antiguo regulador de voltaje por un 7808; ahora bien, aun es necesario suplir el componente de cambio de nivel con un voltaje de 5V, con este fin se emplea un 7805.

Dados estos cambios y sus implicaciones en el circuito, se ha decidido ir más allá y sustituir el divisor de tensión que se empleaba para la lectura de la batería por un Amplificador Operacional (AO) a fin de mejorar la precisión de las medidas. El nuevo circuito se presenta en la Figura 48, mientras que la nueva PCB se muestra en la Figura 49.

Figura 48

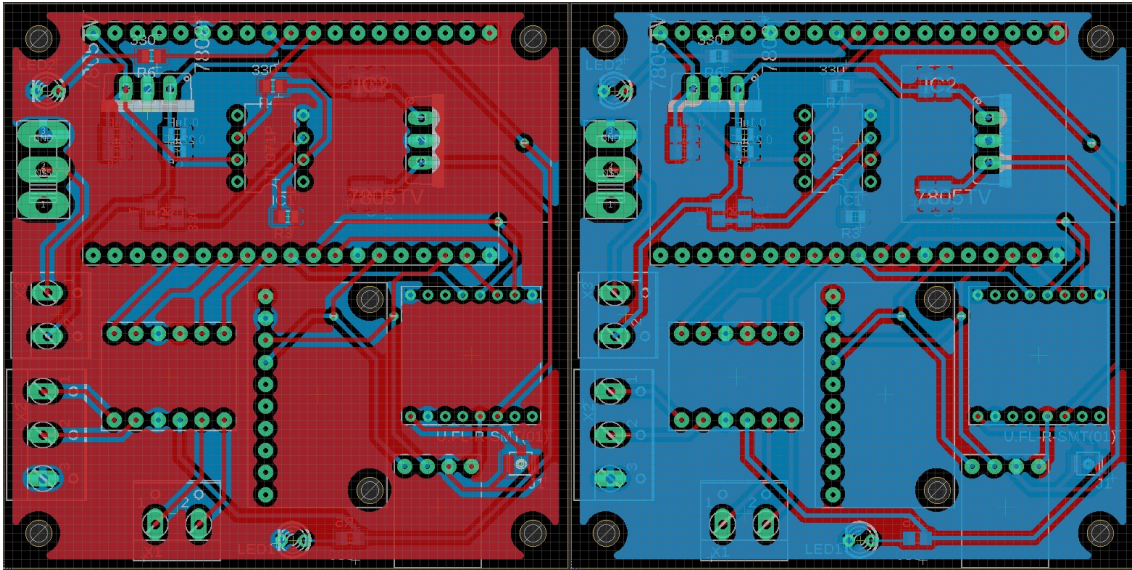
Circuito de la aeronave revisado.



Nota. Fuente propia. Elaborada mediante el programa Eagle.

Figura 49

Placa de circuito impreso de la aeronave revisada.



Nota. Fuente propia. Elaborada mediante el programa Eagle.

Obsérvese que el circuito incluye ahora condensadores en paralelo a fin de obtener un condensador equivalente de $0.33\mu\text{F}$, su función es la de reducir el posible ruido que se genere. El AO no consta de este sistema de desacoplo por estar relativamente cerca del regulador de tensión que le suministra, motivo por el cual se ha visto innecesario.

También en referencia al AO, la selección de las resistencias se ha realizado fijando la de $1\text{k}\Omega$ y aplicando el Teorema de Millman. De esta manera, conocido el voltaje de entrada máximo (12.6V) y estableciendo una salida para dicho voltaje de 3V , para evitar alcanzar el límite de voltaje del ADC de la placa (3.3V), se acaba eligiendo la resistencia normalizada de 330Ω . La función resultante que permite verificar la transformación de voltaje establecida se muestra en la Ecuación (8); según la cual, para la mínima batería (10.2V), se tiene un voltaje de 2.28V .

$$V_{OUT} = \frac{33 \cdot V_{IN} - 108.9}{100} \quad (8)$$

Tanto resistencias como condensadores se han tomado en formato SMD para reducir en la medida de lo posible el espacio que requiere su implementación. Además, se ha aprovechado el espacio que queda bajo la placa controladora para ubicar algunos componentes que no se conectan mediante pines Dupont. Por otra parte, se han instalado bloques terminales para facilitar el cableado con los motores y la alimentación y se ha incluido un led más para indicar que el sistema está encendido.

Propuestas todas estas modificaciones, es necesario comprobar que tanto la batería como los motores siguen resultando factibles para la nueva configuración. Por este motivo, se vuelven a plantear las tablas correspondientes a los apartados 3.1.6 y 3.1.7 adaptadas a la nueva situación. En la Tabla 9 se muestra cómo afectan los cambios al peso de la aeronave.

Tabla 9

Peso de las partes de la aeronave revisado.

Parte	Peso (gramos fuerza)
Estructura de la aeronave	810
Batería	320
PCB	20
RFM69HC	1.5
Antena	17
MPU9250	2
<i>Level Shifter</i>	1.5
Regulador de tensión (2)	3
ESP32-WROOM-32D	7
Amplificado operacional	2
Bloque terminal (3)	12
BMP180	1
Cámara	40
Total	1537

Como se puede apreciar, el peso, aunque ajustado, continua sin exceder el que es capaz de levantar el cuadricóptero en condiciones de operación nominales. Por lo que se puede afirmar que los motores cumplen los requisitos para que no sea necesario repetir los estudios aerodinámicos. En cualquier caso, para poder asegurarlo completamente, es necesario verificar que la batería sigue siendo válida, con este motivo se presenta la Tabla 10.

Empleando la Ecuación (4), se puede comprobar que, para el nuevo consumo de energía del sistema, la batería propuesta como solución continúa asegurando al menos 10 minutos de vuelo en condiciones nominales. Es por ello que se puede concluir que no es necesario realizar ningún cambio más en lo referente a la selección de componentes de la aeronave.

Tabla 10

Consumos de corriente de los elementos de la aeronave.

Componente	Consumo (mA)
RFM69HC	16
MPU9250	4
ESP32-WROOM-32D	240
I/O pin (máx.)	360
• SCL	(40)
• SDA	(40)
• MOSI	(40)
• MISO	(40)
• PWM motores (4)	(160)
• LED	(40)
Motores (4)	28000
Regulador de tensión (2)	50
Amplificado operacional	1
Cámara (caso del objeto del proyecto)	220
Total	28891

Por otra parte, y aunque no se contempla de cara a un prototipo 1.0, cabe destacar que sería necesario modificar algunos aspectos del mando a fin de proporcionar un producto de calidad. Estos aspectos incluyen: introducción de un mecanismo de lectura de la batería, adición de un interruptor y un led que muestre que está funcionando sin necesidad de abrir la carcasa y adición de bloques terminales para facilitar el cableado.

6.2.2. Previsión del *software*

Debido a la incapacidad de realizar las pruebas pertinentes, el *software* que se presenta no ha podido ser comprobado; sin embargo, ya ha sido adaptado a la nueva situación. El código se muestra en el Anejo III; también se incluye el *software* relativo al mando, aunque este no presenta modificaciones con relación al anterior. En este apartado se hacen algunas aclaraciones al respecto.

Con motivo del aprovechamiento de los dos núcleos, se ha decidido emplear el sistema operativo FreeRTOS, también conocido como *Real-Time Operating System* (RTOS), por su código abierto y por estar diseñado específicamente para sistemas embebidos y aplicaciones en tiempo real. FreeRTOS ofrece una amplia gama de características, incluyendo planificación de tareas basada en prioridades, semáforos y otros mecanismos de sincronización que permiten la comunicación entre tareas.

Las funciones necesarias para el funcionamiento del dron se agrupan en tareas que pueden ejecutarse de manera paralela. De esta forma, se ha establecido una tarea que ejecuta el bucle de control actualizando las mediciones, procesando dichas mediciones respecto a la referencia y codificando los resultados para poder enviar la señal PWM; esta tarea, que asegura la estabilidad, se ejecuta cada 4 segundos como máximo, tiempo que se corresponde con dos ciclos seguidos del PWM con un protocolo ESC estándar.

Se han implementado: una tarea para recibir por radio las señales del mando y procesarlas mediante un filtro exponencial, otra que lee la batería y corrige el voltaje necesario para mantener la flotación y otra que comprueba y asegura que no se alcanza la altitud máxima. Se configuran cuatro tareas más, una por cada motor, cuya única función es actualizar el ciclo de los PWM cada 4 segundos.

En caso de pérdida de señal o de batería baja, la aeronave realiza un aterrizaje de emergencia; pero solo puede salir de este proceso en el primer caso, al reestablecerse la conexión. El mando es capaz de avisar al usuario, mediante un led, que una de estas dos situaciones se ha producido.

Las bibliotecas necesarias para permitir el funcionamiento del código presentado se enumeran a continuación con su respectiva cita y una breve explicación:

- Arduino, Wire y SPI, propias del entorno de desarrollo de Arduino; están instaladas por defecto.
- Math, propia del lenguaje C++ sobre el que se construye Arduino; está instalada por defecto.
- RH_RH69 y RHReliableDatagram son sub-bibliotecas de RadioHead (Mike McCauley, 2022); permiten la lectura/escritura de registros de la radio de manera sencilla y establecen un protocolo para la comunicación entre este tipo de dispositivos.
- I2Cdev y MPU9250 son modificaciones de la propia biblioteca I2Cdev, han sido adaptadas a las necesidades del código por Remington (2022); se emplean para comunicarse con el periférico que ostenta este mismo nombre.
- SFE_BMP180 es una biblioteca desarrollada por Rusu et al. (2020) que facilita la comunicación el sensor barométrico y el microcontrolador.
- PIDController es una biblioteca basada en Arduino PID que facilita la creación y gestión de este tipo de controladores, fue creada por Llamas (2021) para dar continuidad a su predecesora.
- ledc y esp_timer son bibliotecas internas de la propia ESP32-WROOM-32D; la primera permite el uso de una unidad PWM de 15bits de resolución, la segunda obtener el tiempo que la unidad de control lleva encendida.
- FreeRTOS, task y semphr son parte del *framework* FreeRTOS, en la ESP32-WROOM-32D están instaladas por defecto.

6.2.3. Ajuste del control de actitud

En el *software* presentado, así como en el apartado 2.5.5, se menciona el uso de controladores PID para controlar la actitud de la aeronave. Para el proceso de ajuste, en primer lugar, sería necesario obtener datos tanto de la referencia como de los sensores; por desgracia, lo segundo es imposible dadas las circunstancias en las que se encuentra el proyecto. A continuación, se relata el proceso que sería necesario seguir para ajustar los parámetros que definen los PID, en caso de poder disponer de dicha información.

Antes que nada, sería necesario configurar la comunicación entre el ordenador y microcontrolador; el programa Matlab es capaz de obtener la información de la Arduino, mediante un objeto "serial" y las funciones "fopen" y "fscanf" a las que se les entrega dicho objeto, si ambos dispositivos comparten la misma *ratio* de baudios y la placa está imprimiendo los valores por el puerto serie. Obtenidos los valores, y preprocesados en un objeto "iddata", se importan a System Identification Toolbox.

Mediante la citada herramienta de Matlab se obtienen las funciones de transferencia. Es importante disponer de datos suficientes y diversos para asegurar una buena identificación, así como asegurar que los datos de entrada están libres de perturbaciones; para ello, sería una buena decisión realizar las pruebas en un entorno controlado. Además, a fin de evitar que la aeronave sufra daños, sería una buena práctica emplear un amarre que no impida ni entorpezca los movimientos.

Una vez hecho todo eso, es necesario validar las funciones de transferencia obtenidas. Para esto, se vuelven a recoger datos de las referencias y de los sensores; los primeros se introducen como dato de entrada en Simulink para obtener las salidas de las funciones de transferencia, los segundos se emplean para validar los resultados de la simulación. Las funciones que se emplean para esto último son "compare" y "resid"; de no ser satisfactorios los resultados, es necesario proceder con otra iteración.

Validados los modelos, es necesario ajustar los controladores para que el sistema trate de alcanzar las referencias especificadas. Para el ajuste de los parámetros de los PID es posible emplear un gran número de métodos; el que se propone es mediante el uso de la herramienta de ajuste de Tune PID Controller de Matlab, que no es sino un método experimental; pero podrían aplicarse procedimientos más complejos vistos en el grado, como el lugar de las raíces, si se analizaran en profundidad los requerimientos.

6.3. REFLEXIÓN PERSONAL

A lo largo del presente proyecto se han aplicado de forma práctica los conocimientos adquiridos durante el grado. Con cada etapa, se ha podido comprender de una manera más profunda los desafíos técnicos que enfrenta el diseño de un dron. Al mismo tiempo, se ha ganado experiencia en lo relativo a la gestión de proyectos y se han adquirido habilidades relativas a la planificación, la toma de decisiones y la gestión de riesgos; todas ellas transferibles a una amplia gama de contextos, lo que hace de esta una experiencia enriquecedora desde una perspectiva personal y profesional.

En la realidad, los proyectos a menudo implican enfrentarse a obstáculos inesperados, como en este caso el problema ocurrido con los microcontroladores. Al afrontarlos no como fracasos, sino como oportunidades para crecer, aprender y reflexionar, es posible enfocarse en la adaptabilidad de la ingeniería y proponer soluciones que no solo resuelven los problemas; los transforman en mejoras de la calidad. Por otra parte, se ha aprendido a valorar la resiliencia como una aptitud crucial en el desarrollo de proyectos. Esto se debe a que, en muchas ocasiones, la motivación es uno de los elementos que más puede frenar la búsqueda de soluciones ante la adversidad; analizar la situación con filosofía y mente abierta encauza el pensamiento al resultado y no al problema.

A pesar de que el proyecto no haya alcanzado la fase de validación del *software* y vuelo, la revisión de diseño y la propuesta de mejoras muestran el compromiso con la optimización y el aprendizaje. Personalmente, me habría gustado alcanzar los objetivos presentados al inicio de esta memoria; sin embargo, la existencia de una tasa administrativa tan cuantiosa actúa como barrera presupuestaria frente a la posibilidad de desarrollar una versión 1.0 del prototipo.

BIBLIOGRAFÍA

Bellini, A., & Güçeri, S. (2003). *Mechanical characterization of parts fabricated using fused deposition modeling*. Rapid Prototyping Journal.

Agencia Estatal de Seguridad Aérea. (s.f.). *Operaciones UAS/Drones - Categoría Abierta*. Obtenido de <https://www.seguridadaerea.gob.es/es/ambitos/drones/operaciones-uas-drones/operaciones-con-uas-drones---categoria-abierta-subcategorias-a1-a2-y-a3>

Aguado, C. (6 de 5 de 2017). *Red Bull*. Obtenido de <https://www.redbull.com/es-es/las-carreras-de-drones-el-deporte-del-futuro>

AliExpress. (s.f. a). *AliExpress*. Recuperado el 2022, de https://es.aliexpress.com/item/4000341305404.html?spm=a2g0o.search0304.0.0.41a7171eJido2p&algo_pvid=3c7f5058-1e55-4205-b288-d70ce7d2d7c0&algo_exp_id=3c7f5058-1e55-4205-b288-d70ce7d2d7c0-30&pdp_ext_f=%7B%22sku_id%22%3A%2212000028519586967%22%7D&pdp_npi=2%4

AliExpress. (s.f. b). *AliExpress*. Recuperado el 2022, de https://es.aliexpress.com/item/32912933373.html?spm=a2g0o.search0304.0.0.41a7171eJido2p&algo_pvid=68daf699-2656-4b6f-945a-bea36e6aa12a&algo_exp_id=68daf699-2656-4b6f-945a-bea36e6aa12a-47&pdp_ext_f=%7B%22sku_id%22%3A%2265953909461%22%7D&pdp_npi=2%40dis%21E

AliExpress. (s.f. c). *AliExpress*. Recuperado el 2022, de https://es.aliexpress.com/item/1005001374629450.html?spm=a2g0o.detail.0.0.23f94d48ez9mXx&gps-id=pcDetailBottomMoreThisSeller&scm=1007.13339.274681.0&scm_id=1007.13339.274681.0&scm-url=1007.13339.274681.0&pvid=1e9d564c-a075-41b8-a692-355124fc62bc&_t=gps-id

AliExpress. (s.f. d). *AliExpress*. Recuperado el 2022, de https://es.aliexpress.com/item/32971184721.html?spm=a2g0o.detail.0.0.23f94d48ez9mXx&gps-id=pcDetailBottomMoreThisSeller&scm=1007.13339.274681.0&scm_id=1007.13339.274681.0&scm-url=1007.13339.274681.0&pvid=1e9d564c-a075-41b8-a692-355124fc62bc&_t=gps-id%3Apc

Amazon. (s.f. a). *Amazon*. Recuperado el 2022, de <https://www.amazon.es/QWinOut-bricolaje-desmontaje-Quadcopter-versi%C3%B3n/dp/B08NPT94PM?th=1>

- Amazon. (s.f. b). *Amazon*. Recuperado el 2022, de https://www.amazon.es/QWinOut-F450-V2-Quadcopter-Goggles-Battery/dp/B081N9GRWZ/ref=sr_1_15?keywords=FPV%2BKit&qid=1652724995&sr=8-15&th=1
- Arslan, Z., & Ozbek, A. (2018). *Effect of Propeller Size and Motor Position on the Performance of a Quadrotor UAV*. Journal of Intelligent & Robotic Systems.
- Bayerische Motoren Werke AG. (27 de 11 de 2019). *Ni siquiera un deportivo puede superarlos en la aceleración*. Obtenido de <https://www.bmw.com/es/automotive-life/carreras-de-drones.html>
- CORTEVA. (s.f.). *El futuro de la agricultura*. Obtenido de <https://www.corteva.es/blog/drones-en-la-agricultura-sus-usos-y-beneficios.html>
- Díaz, M. J. (2021). *Normas Básicas y Recomendaciones en el Diseño de PCBs*. Universidad de Sevilla.
- Drone Nodes. (s.f.). *How to choose drone frame for racing or freestyle?* Obtenido de <https://dronenodes.com/drone-frame-racing-freestyle/>
- Llamas Binaburo, L. (2021). Obtenido de <https://github.com/luisllamasbinaburo/Arduino-PIDController-Library/tree/main>
- Ludwig, S. A., & Burnham, K. D. (2018). *Comparison of EulerEstimate using Extended Kalman Filter, Madgwick and Mahony on Quadcopter Flight Data*. Dallas, TX, USA: International Conference on Unmanned Aircraft Systems (ICUAS).
- Magán Parreño, S. (7 de 11 de 2017). *Universidad Camilo José Cela*. Obtenido de Blog de CC de Transporte y Logística: <https://blogs.ucjc.edu/cc-transporte-logistica/2017/11/uso-drones-reducir-los-niveles-contaminacion/>
- McCauley, M. (2022). Obtenido de <http://www.airspayce.com/mikem/arduino/RadioHead/>
- MECALUX. (16 de 6 de 2020). *Mecalux News*. Obtenido de <https://www.mecalux.es/blog/drones-logistica>
- MINISTERIO DE TRANSPORTES, MOVILIDAD Y AGENDA URBANA. (14 de 7 de 2021). *Sala de prensa*. Obtenido de <http://www.salvamentomaritimo.es/sala-de-comunicacion/sala-de-prensa/salvamento-maritimo-utiliza-drones-de-ems-a-para-labores-de-lucha-contra-la-contaminacion-y-control-de-traffic-maritimo>
- Mokhtar, N. S. (2020). *Modelling and PID control system integration for quadcopter DJI F450 attitude stabilization*. Indonesian Journal of Electrical Engineering and Computer Science.
- National Aeronautics and Space Administration. (s.f.). *Propeller Thrust*. Obtenido de <https://www.grc.nasa.gov/www/k-12/airplane/proph.html>
- Niemiec, R., & Gandhi, F. (2016). *A Comparison Between Quadrotor Flight Configurations*.
- Orden ETD/666/2020. (13 de 7 de 2020). *Aprueba el Cuadro Nacional de Atribución de Frecuencias*. *Boletín Oficial del Estado*. Obtenido de <https://www.boe.es/eli/es/o/2020/07/13/etd666>

- Organización Internacional de Normalización. (2021). ISO/ASTM 52900. *Fabricación aditiva*.
- Quevedo Supelano, Y. N., & Bonilla Dueñas, C. A. (2021). *Caracterización de las propiedades mecánicas del copolímero PETG mediante impresión FDM 3D*. Bogotá: Universidad Distrital Francisco José de Caldas.
- Rajeevalochanam, P., Kumar, V., Sahadevan, S., Sharma, N., & Janney, K. (2016). *Comparative Study of Two and Three Blade Mini Propellers Aerodynamic Performance*.
- Ramlan, I., Darlis, N., & Jet, L. J. (2020). *Comparison between Solidworks and Ansys Flow Simulation on Aerodynamic Studies*. Journal of Industry, Engineering and Innovation.
- Reglamento de Ejecución (UE) 2019/947. (24 de 5 de 2019). relativo a las normas y los procedimientos aplicables a la utilización de aeronaves no tripuladas. *Boletín Oficial del Estado*. Recuperado el 11 de 6 de 2019, de <https://www.boe.es/eli/es/o/2020/07/13/etd666>
- Remington, S. J. (2022). Obtenido de <https://github.com/jremington/MPU-9250-AHRS>
- Rusu, F., Lyons, G., & Peter, A. (2020). Obtenido de https://github.com/LowPowerLab/SFE_BMP180
- Sforza, P. M. (2017). *Theory of Aerospace Propulsion*. Butterworth-Heinemann.
- Spoerk, M., Sapkota, J., Gonzalez Gutierrez, J., Holzner, C., & Schuschnigg, S. (2017). *Effect of the printing bed temperature on the adhesion of parts produced by fused filament fabrication*. *Plastics Rubber and Composites*.
- Szykiedans, K., Credo, W., & Osiński, D. (2017). *Selected Mechanical Properties of PETG 3-D Prints*. *Procedia Engineering*.
- Theys, B., Dimitriadis, G., Hendrick, P., & Schutter, J. (2016). *Influence of propeller configuration on propulsion system efficiency of multi-rotor Unmanned Aerial Vehicles*.
- Universidad Politécnica de Valencia. (s.f.). *Impresión y Fabricación 3D, Tema 1 (p. 5)*.

PRESUPUESTO

DISEÑO E IMPRESIÓN 3D DE UN MODELO DE
DRON PARA COMPETICIÓN



PRESUPUESTO

Para la elaboración del presupuesto se han tenido en cuenta, únicamente, los recursos pertinentes a la versión beta de la aeronave; sin embargo, en referencia a los precios descompuestos, se ha decidido incluir también aquellos artículos asociados a la versión alfa y los adquiridos, pero no empleados. Esta decisión se sustenta en que, realmente, han sido parte de los costos; a pesar de todo, no se han empleado para producir la versión actual. No se han considerado los costes directos complementarios. En un proyecto real, algunos trabajos expuestos no serían realizados por un ingeniero.

1. CUADRO DE PRECIOS UNITARIOS

Referencia	Uds.	Descripción	Precio total (€)	Vida util / Cantidad	Precio por unidad (€)
MO.GITI	h	Graduado en Ingeniería en Tecnologías Industriales	-	-	15,00
PR.SWP	h	Licencia SolidWorks Premium	4999,00	8760	0,57
PR.INV	h	Licencia Inventor Professional	2886,00	8760	0,33
PR.F360	h	Licencia Eagle, Fusion360	581,00	8760	0,07
MQ.FFF	h	Impresora 3D FFF	500,00	26280	0,02
MQ.PC	h	Ordenador de torre	930,00	43800	0,02
MQ.MB	h	Balaceador magnético	12,99	26280	0,00
MQ.ESS	h	Estación de soldadura, estaño y soldador	132,95	8760	0,02
MP.KMH	ud.	Kit motor, hélices y ESC	97,96	4	24,49
MP.SAM	ud.	SAMD21 M0-Mini	36,80	4	9,20
MP.IMU9	ud.	Sensor MPU9250	3,79	1	3,79
MP.IMU6	ud.	Sensor MPU6050	12,00	3	4,00
MP.BAT1	ud.	Batería mando	16,87	1	16,87
MP.BAT2	ud.	Batería aeronave	39,99	1	39,99
MP.RFM	ud.	Radio RFM69HC	11,49	4	2,87
MP.LORA	ud.	Radio LoRa SX1278	6,72	2	3,36
MP.BMP	ud.	Barómetro BMP180	1,27	1	1,27
MP.CAM	ud.	Cámara WiFi	12,57	1	12,57
MP.PEN	ud.	Kit pilares espaciadores nylon	10,99	280	0,04
MP.PEL	ud.	Kit pilares espaciadores latón	8,99	120	0,07
MP.PAL	ud.	Kit pilares espaciadores aluminio	12,21	20	0,61

Referencia	Uds.	Descripción	Precio total (€)	Vida util / Cantidad	Precio por unidad (€)
MP.TOR	ud.	Kit de tornillos y tuercas M3	21,55	380	0,06
MP.PETG	g	Rollo PETG	24,19	1000	0,02
MP.PLA	g	Rollo PLA	16,49	1000	0,02
MP.PCB	ud.	Fabricación PCB	14,73	10	1,47
MP.RES	ud.	Juego de resistencias E24	5,09	300	0,02
MP.TBSS	ud.	Transistor BSS138	1,34	100	0,01
MP.BUCK	ud.	Regulador de tensión	3,89	2	1,95
MP.CCR	cm	Cable de cobre rígido	21,99	700	0,03
MP.CAWG	cm	Cable AWG 22	13,78	1100	0,01
MP.LED	ud.	Surtido de luces LED	1,62	100	0,02
MP.LVL	ud.	<i>Level shifter</i>	1,74	10	0,17
MP.INT	ud.	Interruptor	3,48	10	0,35
MP.ANT	ud.	Antena 433MHz 5dBi	5,93	2	2,97
MP.UFL	ud.	Conectores UFL	1,64	10	0,16
MP.ARD	ud.	Arduino Nano	7,34	2	3,67
MP.JOY	ud.	<i>Joysticks</i> KY-023	1,33	2	0,67
MP.ESM	ml	Esmalte de uñas negro	1,80	8	0,23
MP.ACE	ml	Acetona	2,19	200	0,01

2. CUADRO DE PRECIOS NÚMERO 1

Referencia	Ud.	Descripción de la partida	Rdto.	Precio (€)	Importe (€)
UO 01	ud.	Estudio preliminar		Subtotal	3778,02
UO 01.1	ud.	Análisis del estado del arte		3002,40	
MO.GITI	h	Graduado en Ingeniería en Tecnologías Industriales	120	25,00	3000,00
MQ.PC	h	Ordenador de torre	120	0,02	2,40
UO 01.2	ud.	Elaboración de un plan de resolución		100,08	
MO.GITI	h	Graduado en Ingeniería en Tecnologías Industriales	4	25,00	100,00
MQ.PC	h	Ordenador de torre	4	0,02	0,08
UO 01.3	ud.	Creación de una matriz de características y análisis de productos		325,26	
MO.GITI	h	Graduado en Ingeniería en Tecnologías Industriales	13	25,00	325,00
MQ.PC	h	Ordenador de torre	13	0,02	0,26
UO 01.4	ud.	Establecimiento de un orden de magnitud		350,28	
MO.GITI	h	Graduado en Ingeniería en Tecnologías Industriales	14	25,00	350,00
MQ.PC	h	Ordenador de torre	14	0,02	0,28

Referencia	Ud.	Descripción de la partida	Rdto.	Precio (€)	Importe (€)
UO 02	ud.	Selección de componentes		Subtotal	1579,98
UO 02.1	ud.	Estudio y comparativa de opciones		350,28	
MO.GITI	h	Graduado en Ingeniería en Tecnologías Industriales	14	25,00	350,00
MQ.PC	h	Ordenador de torre	14	0,02	0,28
UO 02.2	ud.	Compra		351,55	
MO.GITI	h	Graduado en Ingeniería en Tecnologías Industriales	6	25,00	150,00
MQ.PC	h	Ordenador de torre	6	0,02	0,12
MP.KMH	ud.	Kit motor, hélices y ESC	4	24,49	97,96
MP.SAM	ud.	SAMD21 M0-Mini	1	9,20	9,20
MP.ARD	ud.	Arduino Nano	1	3,67	3,67
MP.IMU9	ud.	Sensor MPU9250	1	3,79	3,79
MP.REG	ud.	Regulador de tensión	1	1,95	1,95
MP.LVL	ud.	<i>Level shifter</i>	2	0,17	0,34
MP.BMP	ud.	Barómetro BMP180	1	1,27	1,27
MP.RFM	ud.	Radio RFM69HC	2	2,87	5,74
MP.CAM	ud.	Cámara WiFi	1	12,57	12,57
MP.UFL	ud.	Conectores UFL	2	0,16	0,32
MP.ANT	ud.	Antena 433MHz 5dBi	2	2,97	5,94
MP.RES	ud.	Resistencias serie E24	4	0,02	0,08
MP.INT	ud.	Interruptor	1	0,35	0,35
MP.LED	ud.	LED rojo	2	0,02	0,04
MP.JOY	ud.	Joysticks KY-023	2	0,67	1,34
MP.TBSS	ud.	Transistor BSS138	1	0,01	0,01
MP.BAT1	ud.	Batería mando	1	16,87	16,87
MP.BAT2	ud.	Batería aeronave	1	39,99	39,99
UO 02.3	ud.	Diseño de los circuitos electrónicos y PCB		878,15	
MO.GITI	h	Graduado en Ingeniería en Tecnologías Industriales	35	25,00	875,00
MQ.PC	h	Ordenador de torre	35	0,02	0,70
PR.F360	h	Licencia Eagle, Fusion360	35	0,07	2,45

Diseño e impresión 3D de un modelo de dron para competición

Referencia	Ud.	Descripción de la partida	Rdto.	Precio (€)	Importe (€)
UO 03	ud.	Diseño estructural		Subtotal	2947,60
UO 03.1	ud.	Diseño inicial		460,62	
MO.GITI	h	Graduado en Ingeniería en Tecnologías Industriales	18	25,00	450,00
MQ.PC	h	Ordenador de torre	18	0,02	0,36
PR.SWP	h	Licencia SolidWorks Premium	18	0,57	10,26
UO 03.2	ud.	Estudios del arrastre y la sustentación y rediseño		898,60	
MO.GITI	h	Graduado en Ingeniería en Tecnologías Industriales	35	25,00	875,00
MQ.PC	h	Ordenador de torre	40	0,02	0,80
PR.SWP	h	Licencia SolidWorks Premium	40	0,57	22,80
UO 03.3	ud.	Análisis de la velocidad terminal		619,47	
MO.GITI	h	Graduado en Ingeniería en Tecnologías Industriales	24	25,00	600,00
MQ.PC	h	Ordenador de torre	33	0,02	0,66
PR.SWP	h	Licencia SolidWorks Premium	33	0,57	18,81
UO 03.4	ud.	Estudio de flotación		154,72	
MO.GITI	h	Graduado en Ingeniería en Tecnologías Industriales	6	25,00	150,00
MQ.PC	h	Ordenador de torre	8	0,02	0,16
PR.SWP	h	Licencia SolidWorks Premium	8	0,57	4,56
UO 03.5	ud.	Selección del material y análisis mecánico		256,49	
MO.GITI	h	Graduado en Ingeniería en Tecnologías Industriales	10	25,00	250,00
MQ.PC	h	Ordenador de torre	11	0,02	0,22
PR.SWP	h	Licencia SolidWorks Premium	11	0,57	6,27
UO 03.6	ud.	Diseño del mando		202,80	
MO.GITI	h	Graduado en Ingeniería en Tecnologías Industriales	8	25,00	200,00
MQ.PC	h	Ordenador de torre	8	0,02	0,16
PR.INV	h	Licencia Inventor Professional	8	0,33	2,64
UO 03.7	ud.	Diseño de los planos		354,90	
MO.GITI	h	Graduado en Ingeniería en Tecnologías Industriales	14	25,00	350,00
MQ.PC	h	Ordenador de torre	14	0,02	0,28
PR.INV	h	Licencia Inventor Professional	14	0,33	4,62

Diseño e impresión 3D de un modelo de dron para competición

Referencia	Ud.	Descripción de la partida	Rdto.	Precio (€)	Importe (€)
UO 04	ud.	Fabricación		Subtotal	670,78
UO 04.1	ud.	Impresión 3D		73,10	
MO.GITI	h	Graduado en Ingeniería en Tecnologías Industriales	2	25,00	50,00
MQ.PC	h	Ordenador de torre	2	0,02	0,04
MQ.FFF	h	Impresora 3D FFF	73,22	0,02	1,46
MP.PETG	g	Plástico PETG	620	0,02	15,00
MP.PLA	g	Plástico PLA	400	0,02	6,60
UO 04.2	ud.	Calibrado de las hélices		100,56	
MO.GITI	h	Graduado en Ingeniería en Tecnologías Industriales	4	25,00	100,00
MQ.MB	h	Balaceador magnético	4	0,00	0,00
MP.ESM	ml	Esmalte de uñas negro	2	0,23	0,46
MP.ACE	ml	Acetona	10	0,01	0,10
UO 04.3	ud.	Electrónica		378,32	
MO.GITI	h	Graduado en Ingeniería en Tecnologías Industriales	15	25,00	375,00
MQ.ESS	h	Estación de soldadura, estaño y soldador	15	0,02	0,30
MP.PEN	ud.	Kit pilares espaciadores nylon	2	0,04	0,08
MP.PCB	ud.	PCB fabricadas por JLCPCB	2	1,47	2,94
UO 04.4	ud.	Montaje		118,80	
MO.GITI	h	Graduado en Ingeniería en Tecnologías Industriales	4	25,00	100,00
MP.PEN	ud.	Kit pilares espaciadores nylon	22	0,04	0,88
MP.PAL	ud.	Kit pilares espaciadores aluminio	16	0,61	9,76
MP.TOR	ud.	Kit de tornillos y tuercas M3	136	0,06	8,16
UO 05	ud.	Revisión de diseño y propuesta de software		Subtotal	1102,00
MO.GITI	h	Graduado en Ingeniería en Tecnologías Industriales	44	25,00	1100,00
MQ.PC	h	Ordenador de torre	44	0,02	0,88
PR.F360	h	Licencia Eagle, Fusion360	16	0,07	1,12

3. CUADRO DE PRECIOS NÚMERO 2

Referencia	Ud.	Resumen	Precio (€)
UO 01		Estudio preliminar	Total
			3929,14
UO 01.1	ud.	Análisis del estado del arte Análisis de las necesidades del proyecto, de la norma y el ámbito de aplicación, búsqueda y lectura de bibliografía de interés, comprensión de los principios básicos.	
		Mano de obra.....	3000,00
		Maquinaria.....	2,40
		Materiales y otros.....	0,00
		Suma de la partida.....	3002,40
		Costes indirectos (4%).....	120,10
		Total de la partida.....	3122,50
UO 01.2	ud.	Elaboración de un plan de resolución Estipulación de las actividades y sus precedencias, establecimiento de un orden de ejecución.	
		Mano de obra.....	100,00
		Maquinaria.....	0,08
		Materiales y otros.....	0,00
		Suma de la partida.....	100,08
		Costes indirectos (4%).....	4,00
		Total de la partida.....	104,08
UO 01.3	ud.	Creación de una matriz de características y análisis de productos Búsqueda de productos similares, estudio y comparación de las características y sus implicaciones.	
		Mano de obra.....	325,00
		Maquinaria.....	0,26
		Materiales y otros.....	0,00
		Suma de la partida.....	325,26
		Costes indirectos (4%).....	13,01
		Total de la partida.....	338,27

Referencia	Ud.	Resumen	Precio (€)
UO 01.4	ud.	Establecimiento de un orden de magnitud Determinación de las necesidades técnicas y resolución general de las mismas.	
		Mano de obra.....	350,00
		Maquinaria.....	0,28
		Materiales y otros.....	0,00
		Suma de la partida.....	350,28
		Costes indirectos (4%).....	14,01
		Total de la partida.....	364,29
UO 02		Selección de componentes	Total 1643,18
UO 02.1	ud.	Estudio y comparativa de opciones Búsqueda de alternativas para la solución de detalle de las necesidades establecidas en el orden de magnitud, selección de los componentes que mejor se ajustan a dichas necesidades.	
		Mano de obra.....	350,00
		Maquinaria.....	0,28
		Materiales y otros.....	0,00
		Suma de la partida.....	350,28
		Costes indirectos (4%).....	14,01
		Total de la partida.....	364,29
UO 02.2	ud.	Compra Contacto con los distintos proveedores mediante páginas específicamente diseñadas para ello, compra del material necesario y verificación.	
		Mano de obra.....	150,00
		Maquinaria.....	0,12
		Materiales y otros.....	201,43
		Suma de la partida.....	351,55
		Costes indirectos (4%).....	14,06
		Total de la partida.....	365,61
UO 02.3	ud.	Diseño de los circuitos electrónicos y PCB Creación de bibliotecas personalizadas, diseño del circuito electrónico del mando y la aeronave, diseño de las PCB asociadas a dichos circuitos.	
		Mano de obra.....	875,00
		Maquinaria.....	0,70
		Materiales y otros.....	2,45
		Suma de la partida.....	878,15
		Costes indirectos (4%).....	35,13
		Total de la partida.....	913,28

Referencia	Ud.	Resumen	Precio (€)
UO 03		Diseño estructural	Total
			3121,34
UO 03.1	ud.	Diseño inicial Concepción inicial de la aeronave, diseño de las piezas y ensamblado.	
		Mano de obra.....	450,00
		Maquinaria.....	0,36
		Materiales y otros.....	10,26
		Suma de la partida.....	460,62
		Costes indirectos (4%)....	18,42
		Total de la partida.....	479,04
UO 03.2	ud.	Estudios del arrastre y la sustentación y rediseño Estudios aerodinámicos para determinar arrastre y sustentación de la aeronave y modificación del diseño en pos de soluciones óptimas.	
		Mano de obra.....	875,00
		Maquinaria.....	0,80
		Materiales y otros.....	22,80
		Suma de la partida.....	898,60
		Costes indirectos (4%)....	35,94
		Total de la partida.....	934,54
UO 03.3	ud.	Análisis de la velocidad terminal Estudios aerodinámicos para determinar el ángulo de ataque óptimo de forma iterativa, estudio del arrastre y sustentación para cada ángulo.	
		Mano de obra.....	600,00
		Maquinaria.....	0,66
		Materiales y otros.....	18,81
		Suma de la partida.....	619,47
		Costes indirectos (4%)....	24,78
		Total de la partida.....	644,25
UO 03.4	ud.	Estudio de flotación Estudios aerodinámicos para determinar el mínimo voltaje de funcionamiento.	
		Mano de obra.....	150,00
		Maquinaria.....	0,16
		Materiales y otros.....	4,56
		Suma de la partida.....	154,72
		Costes indirectos (4%)....	6,19
		Total de la partida.....	160,91

Referencia	Ud.	Resumen	Precio (€)
UO 03.5	ud.	Selección del material y análisis mecánico Análisis de materiales aptos para el prototipo y selección del mismo, selección de un material para el objeto de proyecto, estudio mecánico de la viabilidad del material.	
		Mano de obra.....	250,00
		Maquinaria.....	0,22
		Materiales y otros.....	6,27
		Suma de la partida.....	256,49
		Costes indirectos (4%).....	10,26
		Total de la partida.....	266,75
UO 03.6	ud.	Diseño del mando Diseño de las partes del mando y ensamblaje, selección del material del mando en el prototipo.	
		Mano de obra.....	200,00
		Maquinaria.....	0,16
		Materiales y otros.....	2,64
		Suma de la partida.....	202,80
		Costes indirectos (4%).....	8,11
		Total de la partida.....	210,91
UO 03.7	ud.	Diseño de los planos Diseño de los planos correspondientes a las piezas de la aeronave y el mando, así como de los respectivos ensamblajes.	
		Mano de obra.....	350,00
		Maquinaria.....	0,28
		Materiales y otros.....	4,62
		Suma de la partida.....	354,90
		Costes indirectos (4%).....	14,20
		Total de la partida.....	369,10
UO 04	Fabricación	Total	697,61
UO 04.1	ud.	Impresión 3D Procesamiento de las piezas en un <i>slicer</i> e impresión de las mismas.	
		Mano de obra.....	50,00
		Maquinaria.....	1,50
		Materiales y otros.....	21,60
		Suma de la partida.....	73,10
		Costes indirectos (4%).....	2,92
		Total de la partida.....	76,02

Referencia	Ud.	Resumen	Precio (€)
UO 04.2	ud.	Calibrado de las hélices Adición controlada de material sobre las hélices.	
		Mano de obra.....	100,00
		Maquinaria.....	0,00
		Materiales y otros.....	0,56
		Suma de la partida.....	100,56
		Costes indirectos (4%).....	4,02
		Total de la partida.....	104,58
UO 04.3	ud.	Electrónica Soldado de los elementos de la electrónica a pines Dupont y posterior ensamblado macho-hembra de los mismos, soldado de componentes SMD a las PCB.	
		Mano de obra.....	375,00
		Maquinaria.....	0,30
		Materiales y otros.....	3,02
		Suma de la partida.....	378,32
		Costes indirectos (4%).....	15,13
		Total de la partida.....	393,45
UO 04.4	ud.	Montaje Ensamblado de las piezas y tornillería de la aeronave y del mando	
		Mano de obra.....	100,00
		Maquinaria.....	0,00
		Materiales y otros.....	18,80
		Suma de la partida.....	118,80
		Costes indirectos (4%).....	4,75
		Total de la partida.....	123,55
UO 05		Revisión de diseño y propuesta de software	Total
			1146,08
UO 05	ud.	Revisión de diseño y propuesta de software Búsqueda de alternativas para el microcontrolador, comprobación del resto de elementos, aplicación de mejoras y propuesta de <i>software</i> para las nuevas condiciones.	
		Mano de obra.....	1100,00
		Maquinaria.....	0,88
		Materiales y otros.....	1,12
		Suma de la partida.....	1102,00
		Costes indirectos (4%).....	44,08
		Total de la partida.....	1146,08

4. PRECIO TOTAL DEL PROYECTO

Referencia	Descripción	Importe (€)
UO 01	Estudio preliminar	3929,14
UO01.1	Análisis del estado del arte	3122,50
UO01.2	Elaboración de un plan de resolución	104,08
UO01.3	Creación de una matriz de características y análisis de productos	338,27
UO01.4	Establecimiento de un orden de magnitud	364,29
UO 02	Selección de componentes	1643,18
UO02.1	Estudio y comparativa de opciones	364,29
UO02.2	Compra	365,61
UO02.3	Diseño de los circuitos electrónicos y PCB	913,28
UO 03	Diseño estructural	3065,50
UO03.1	Diseño inicial	479,04
UO03.2	Estudios del arrastre y la sustentación y rediseño	934,54
UO03.3	Análisis de la velocidad terminal	644,25
UO03.4	Estudio de flotación	160,91
UO03.5	Selección del material y análisis mecánico	266,75
UO03.6	Diseño del mando	210,91
UO03.7	Diseño de los planos	369,10
UO 04	Fabricación	697,61
UO04.1	Impresión 3D	76,02
UO04.2	Calibrado de las hélices	104,58
UO04.3	Electrónica	393,45
UO04.4	Montaje	123,55
UO 05	Revisión de diseño y propuesta de software	1146,08
	Presupuesto de ejecución material.....	10481,52
	Gastos generales (13%).....	1362,59
	Beneficio industrial (6%).....	628,89
	Presupuesto de ejecución por contrata.....	12473,00
	I.V.A. (21%).....	2619,33
	Presupuesto base.....	15092,33

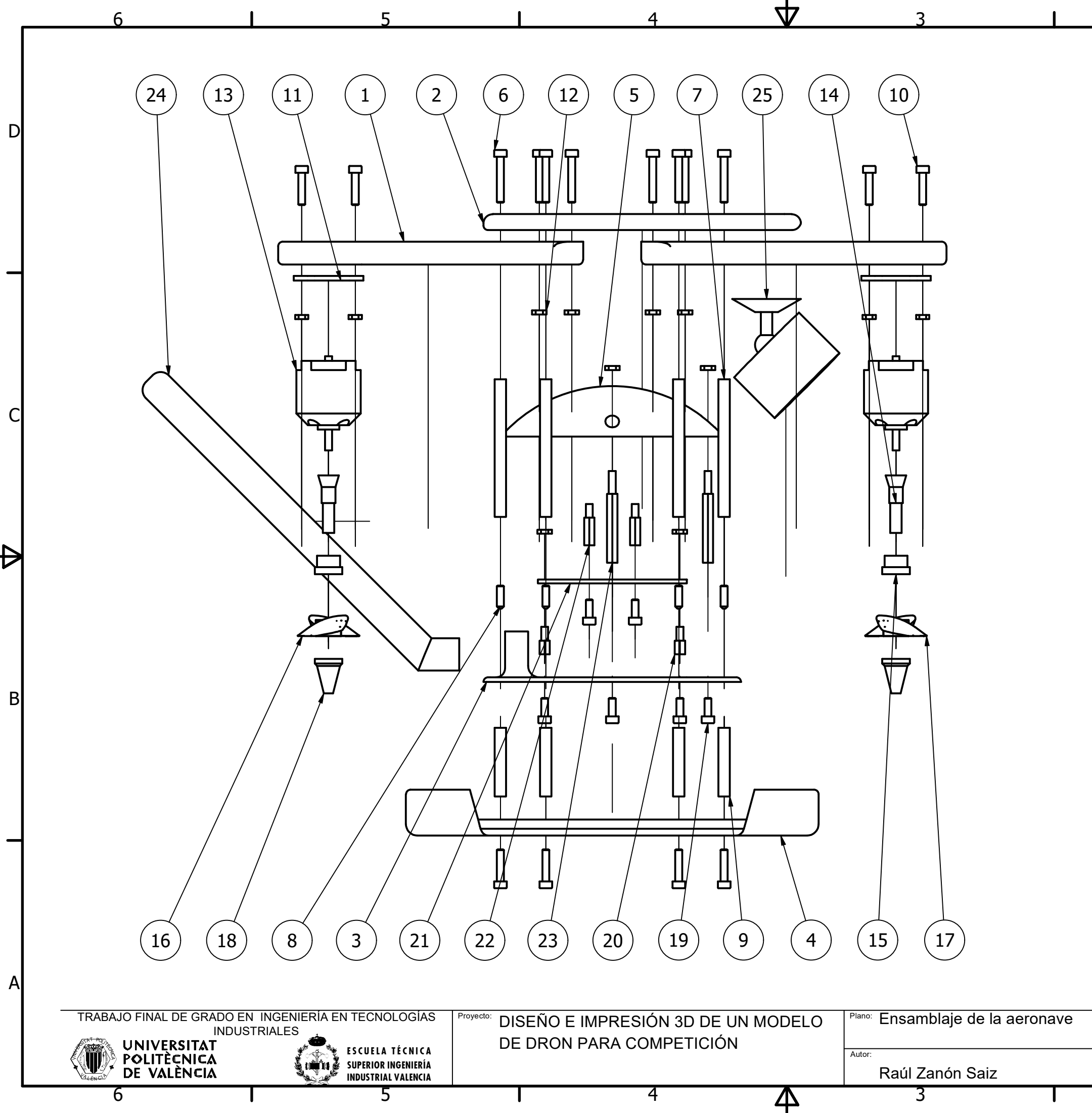
Asciende el presente presupuesto a la expresada cantidad de:

QUINCE MIL NOVENTA Y DOS EUROS CON TREINTA Y TRES CÉNTIMOS.

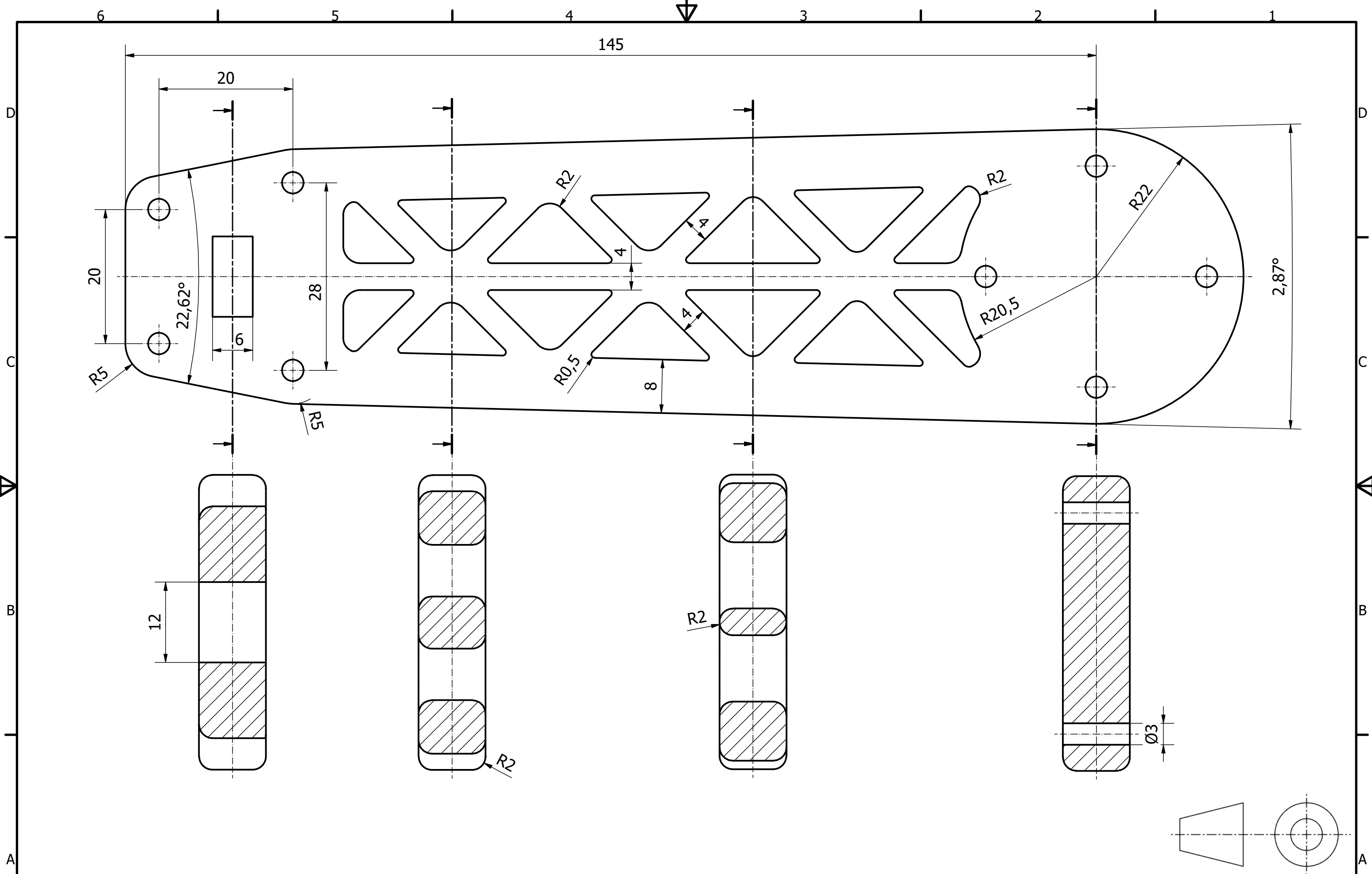
PLANOS

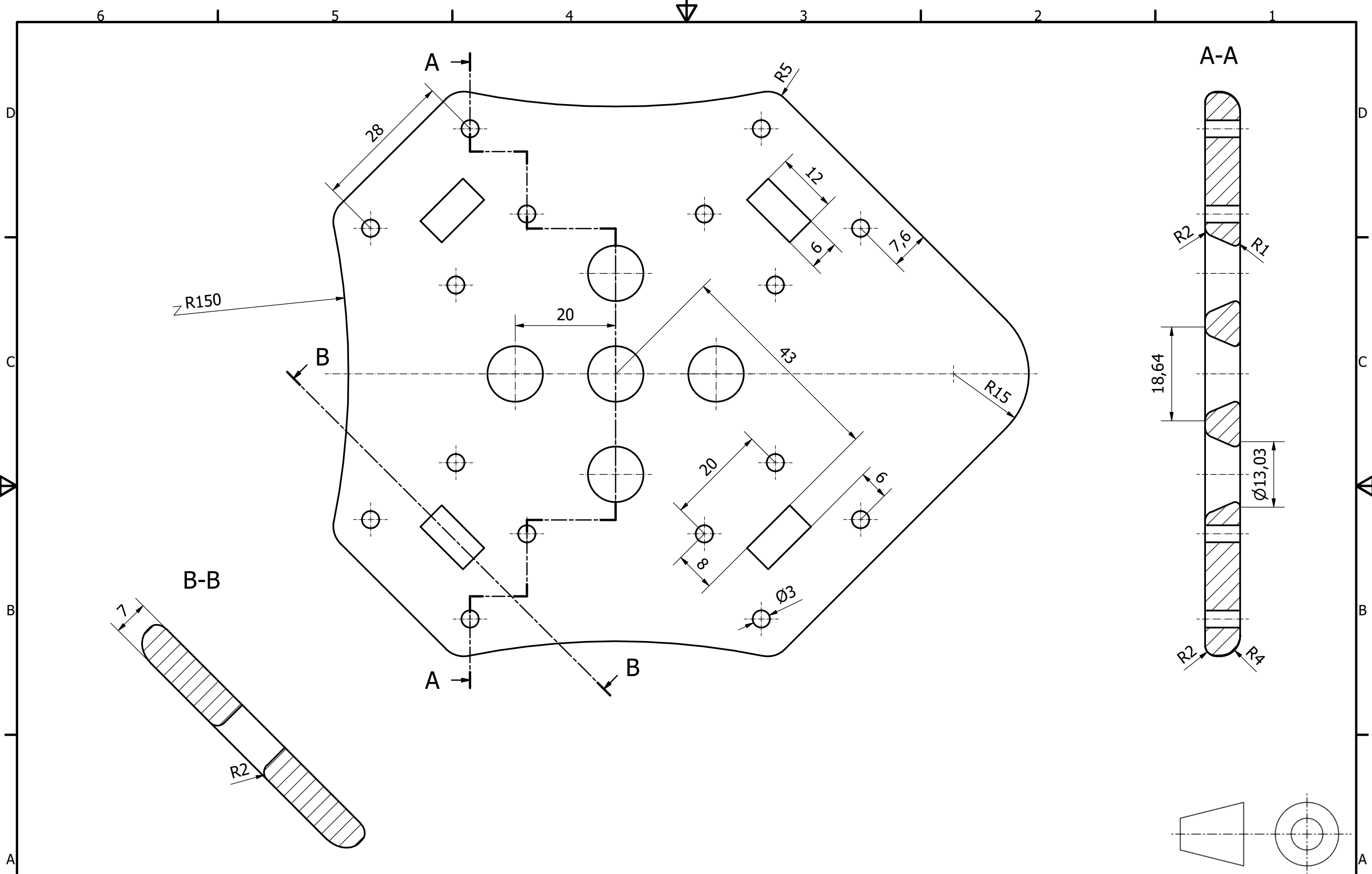
DISEÑO E IMPRESIÓN 3D DE UN MODELO DE
DRON PARA COMPETICIÓN





PARTS LIST		
ITEM	QTY	PART NUMBER
1	4	Ala
2	1	Marco
3	1	Soporte electrónica
4	1	Soporte batería
5	1	Cobertura electrónica
6	16	Tornillo M3 x 20
7	8	Pilar estructural superior
8	8	Tornillo prisionero M3 x 10
9	8	Pilar estructural inferior
10	24	Tornillo M3 x 14
11	4	Soporte motor
12	31	Tuerca M3
13	4	Motor
14	4	Soporte hélice base
15	4	Soporte hélice intermedio
16	2	Hélice anti-horaria
17	2	Hélice horaria
18	4	Soporte hélice punta
19	9	Tornillo M3 x 8
20	4	Pilar pequeño
21	1	PCB
22	2	Pilar mediano
23	3	Pilar grande
24	1	Antena
25	1	Cámara





TRABAJO FINAL DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES



Proyecto: DISEÑO E IMPRESIÓN 3D DE UN MODELO DE DRON PARA COMPETICIÓN

Plano: Marco

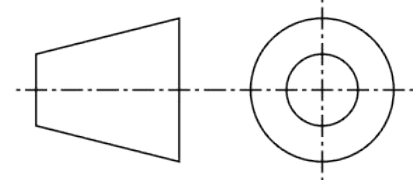
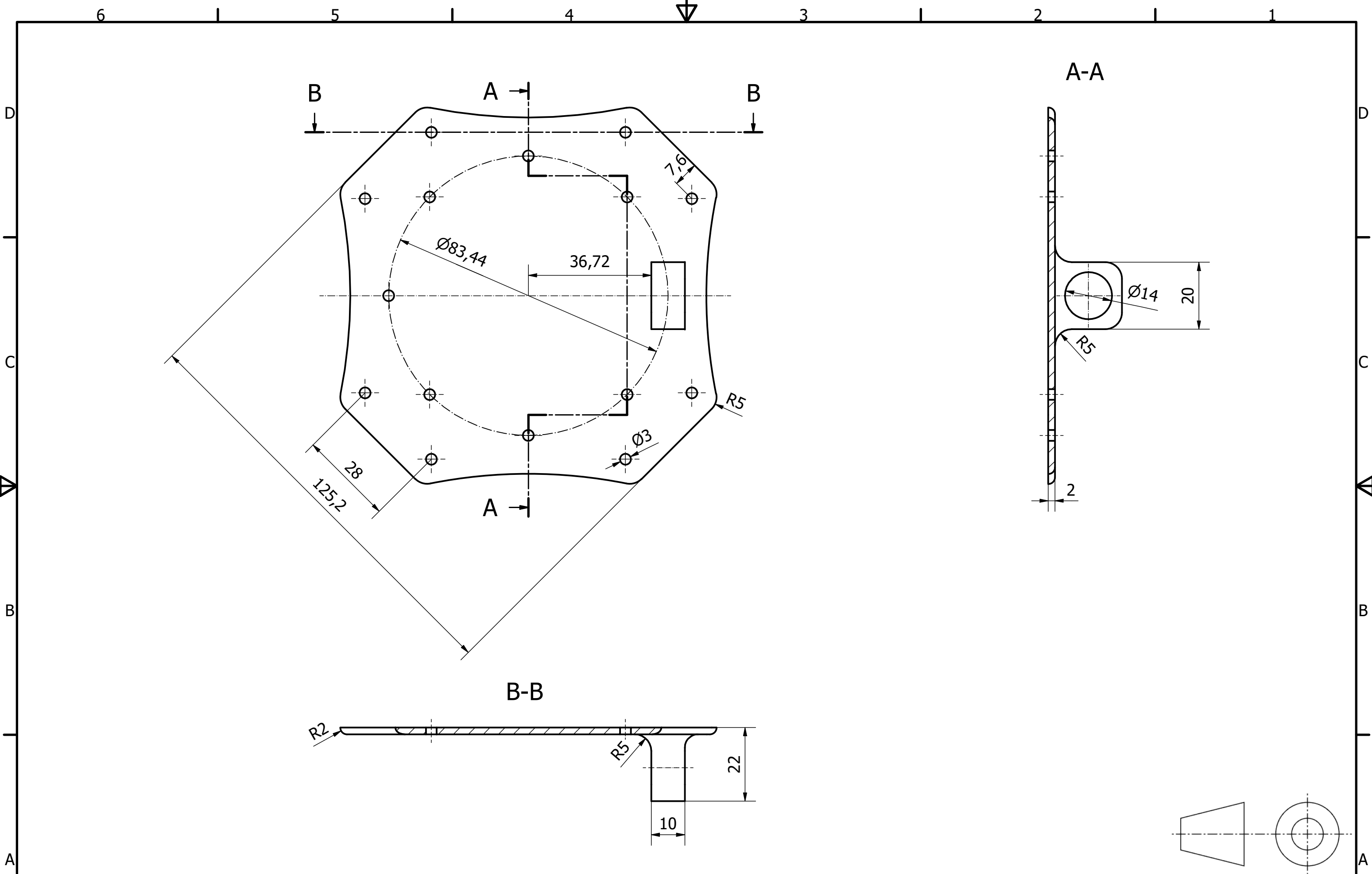
Autor: Raúl Zanón Saiz

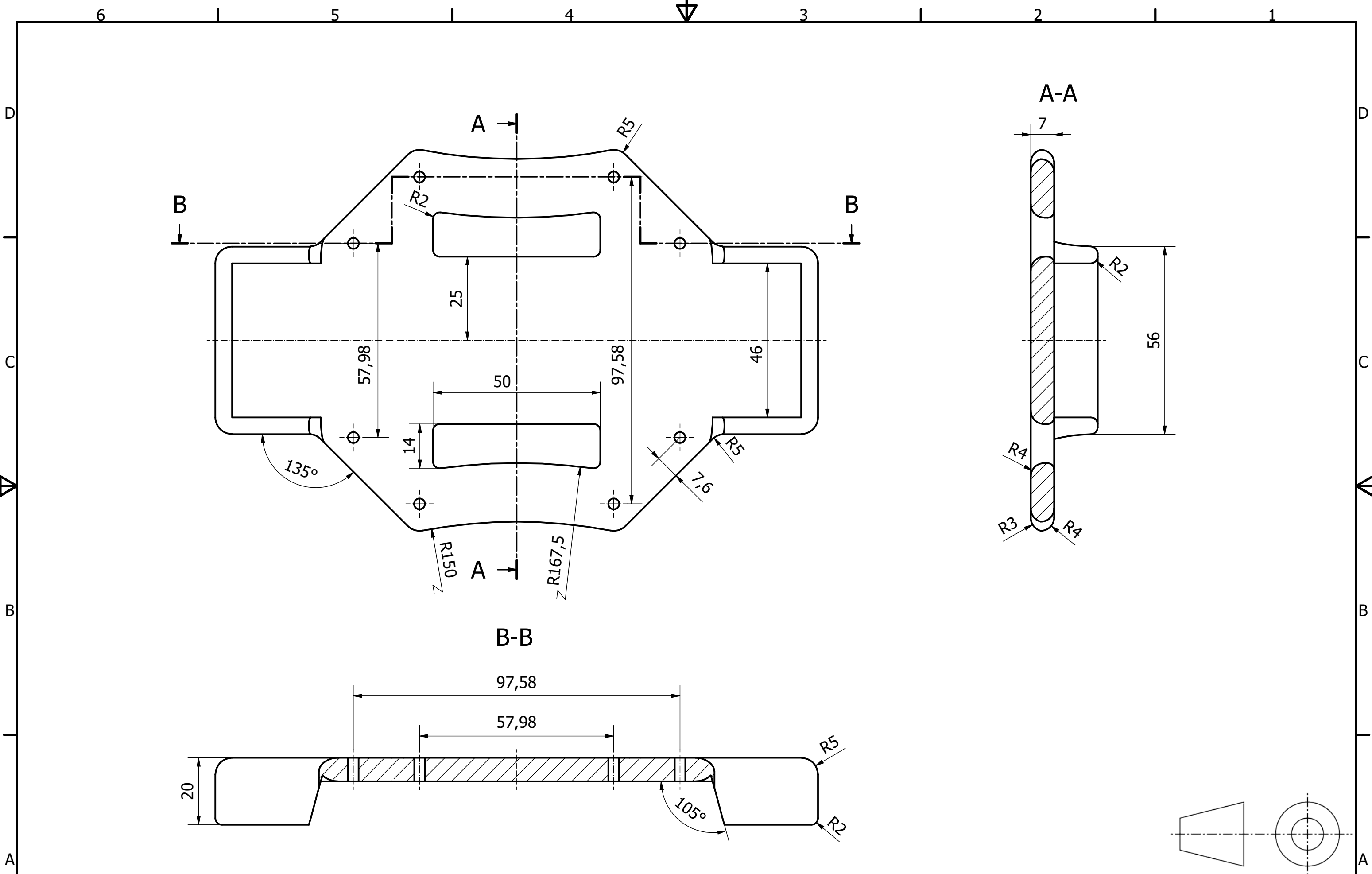
Fecha: Septiembre 2023

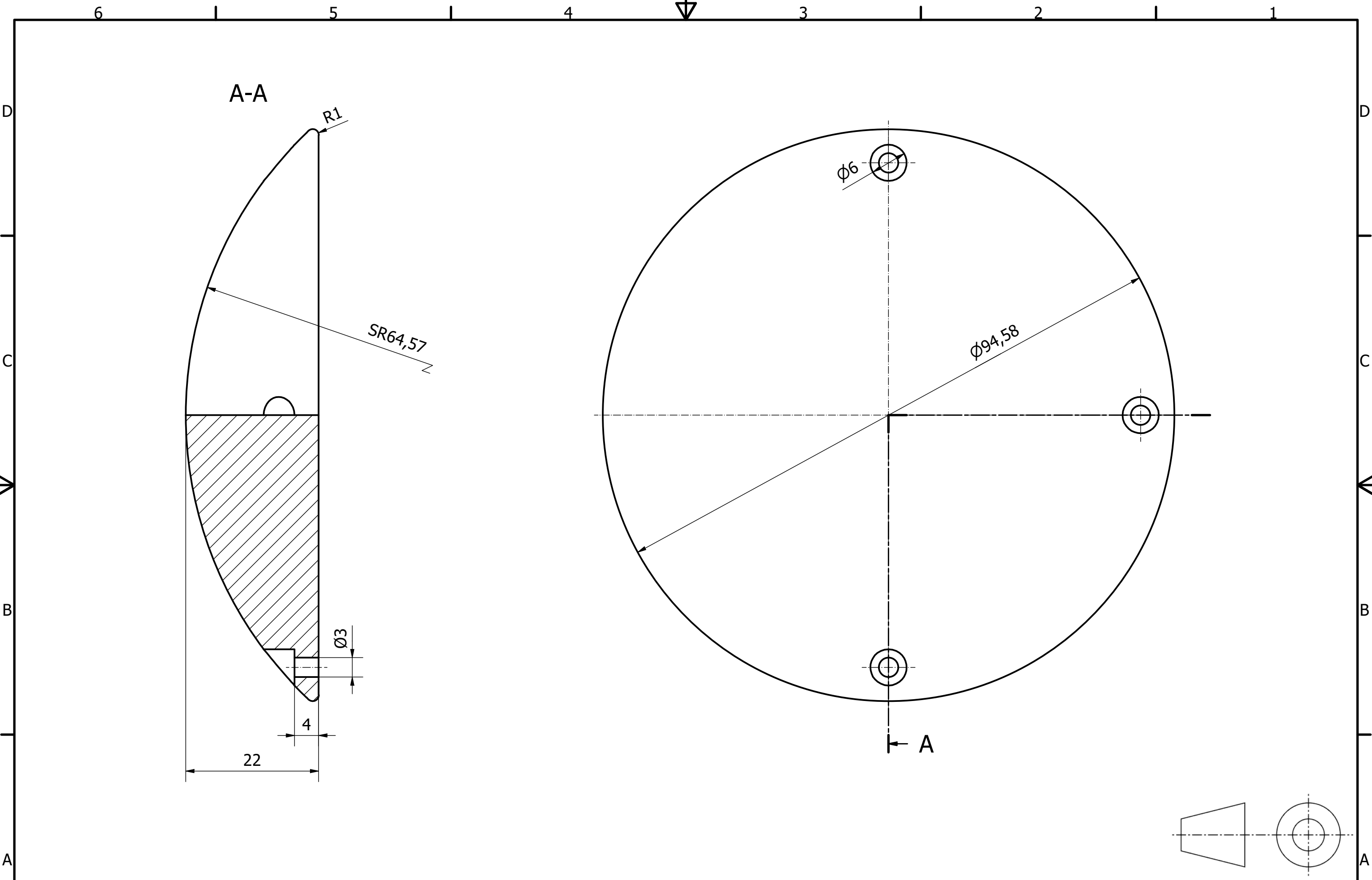
Escala: 3 : 2

Nº Plano:

3







TRABAJO FINAL DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES



Proyecto: DISEÑO E IMPRESIÓN 3D DE UN MODELO DE DRON PARA COMPETICIÓN

Plano: Cobertura electrónica

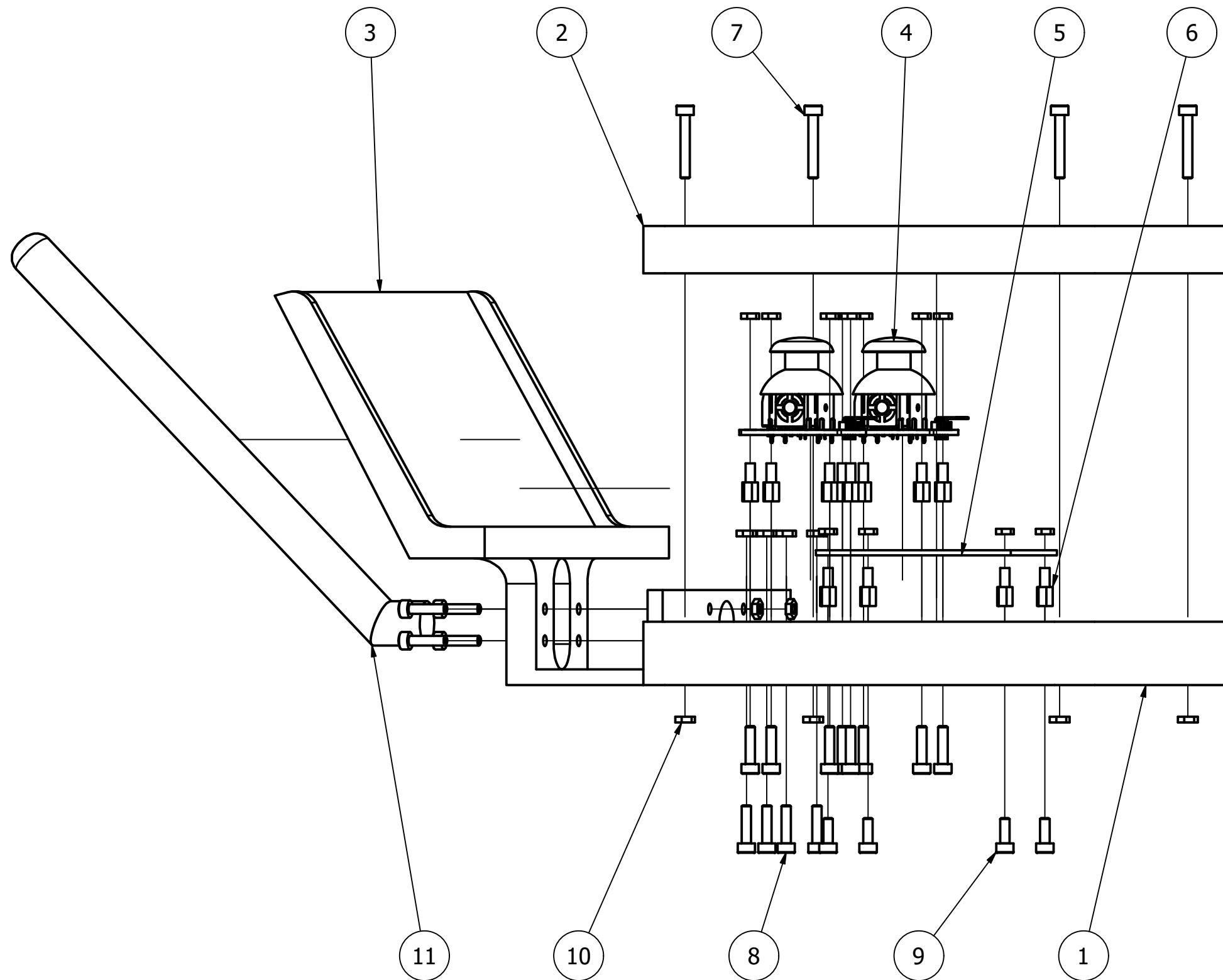
Autor: Raúl Zanón Saiz

Fecha: Septiembre 2023

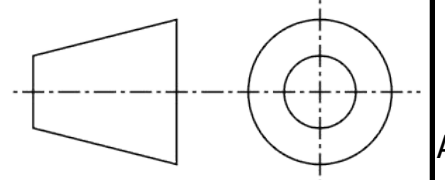
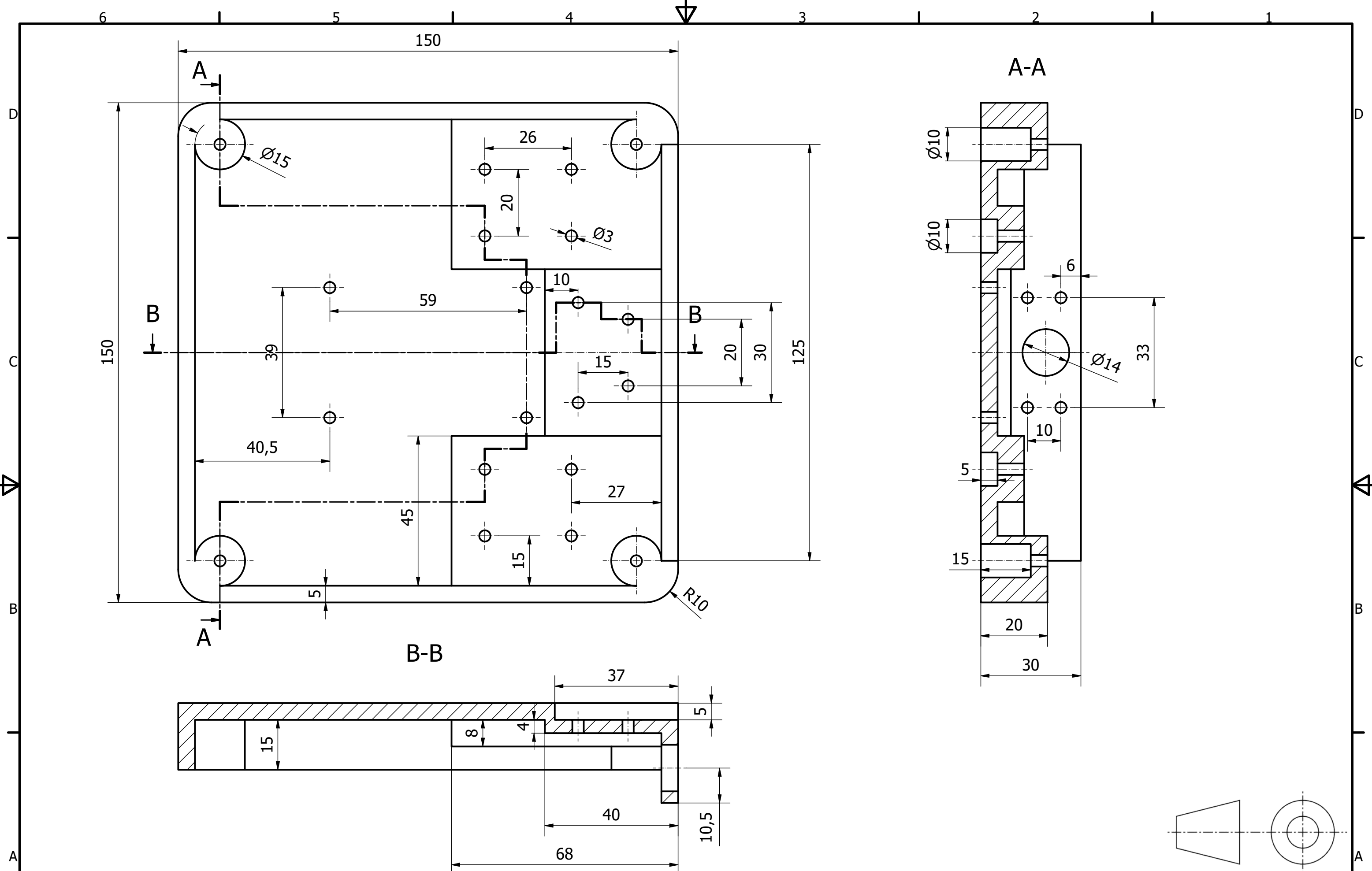
Escala: 9 : 5

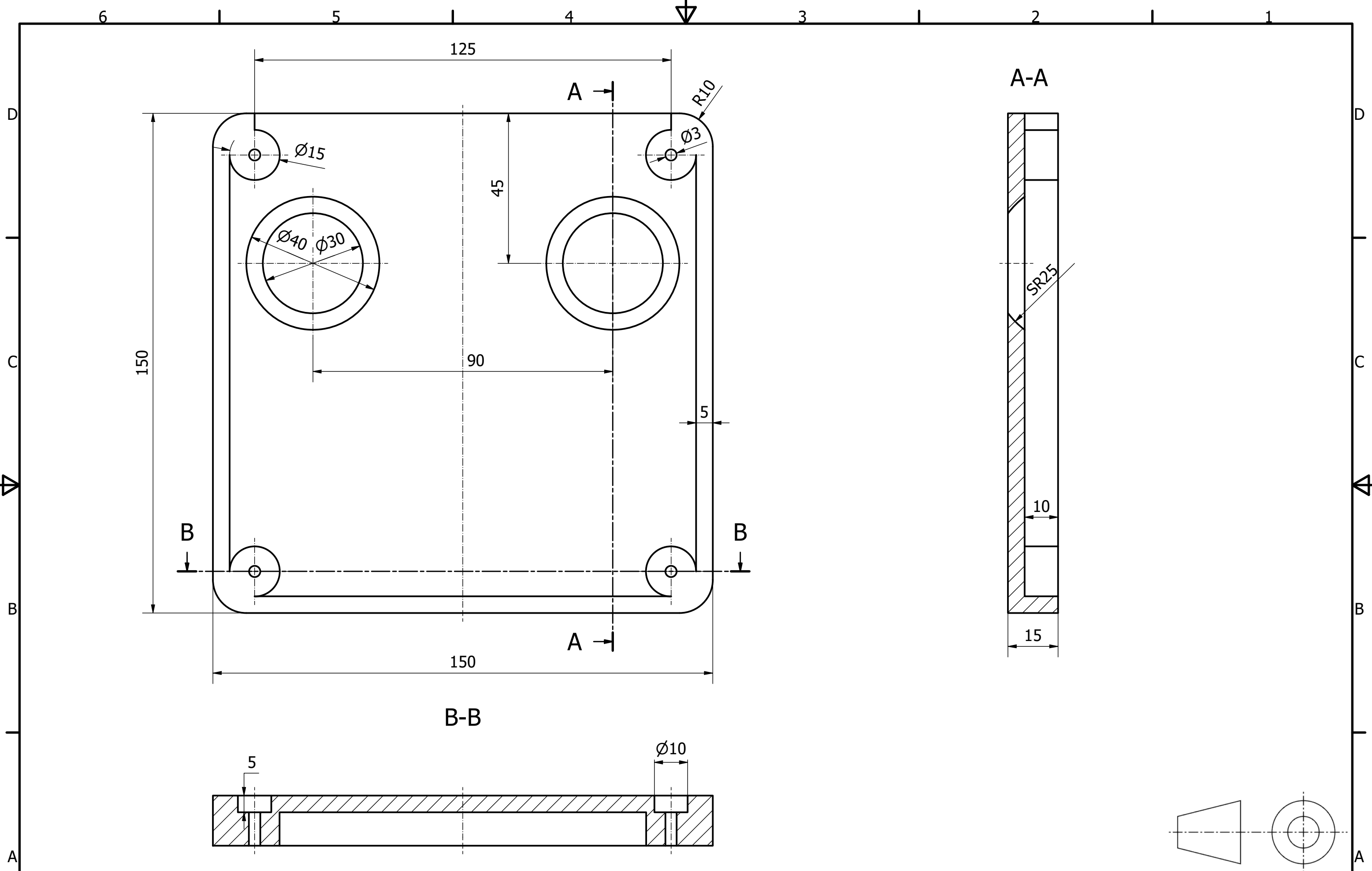
Nº Plano:

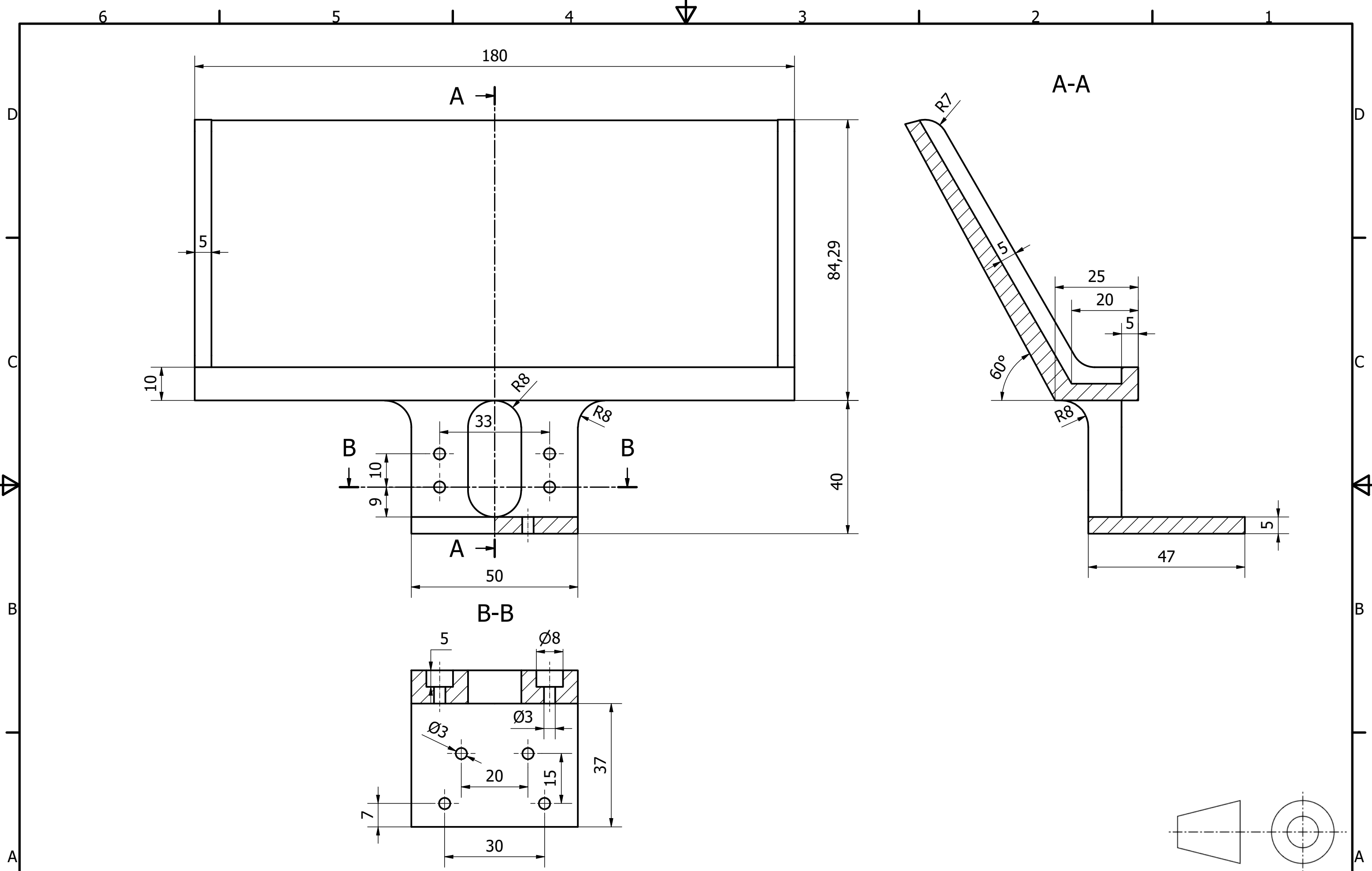
6



PARTS LIST		
ITEM	QTY	PART NUMBER
1	1	Base del mando
2	1	Cobertura del mando
3	1	Soporte para el teléfono móvil
4	2	Joystick
5	1	PCB
6	12	Pilar pequeño
7	4	Tornillo M3 x 20
8	16	Tornillo M3 x 12
9	4	Tornillo M3 x 8
10	24	Tuerca M3
11	1	Antena







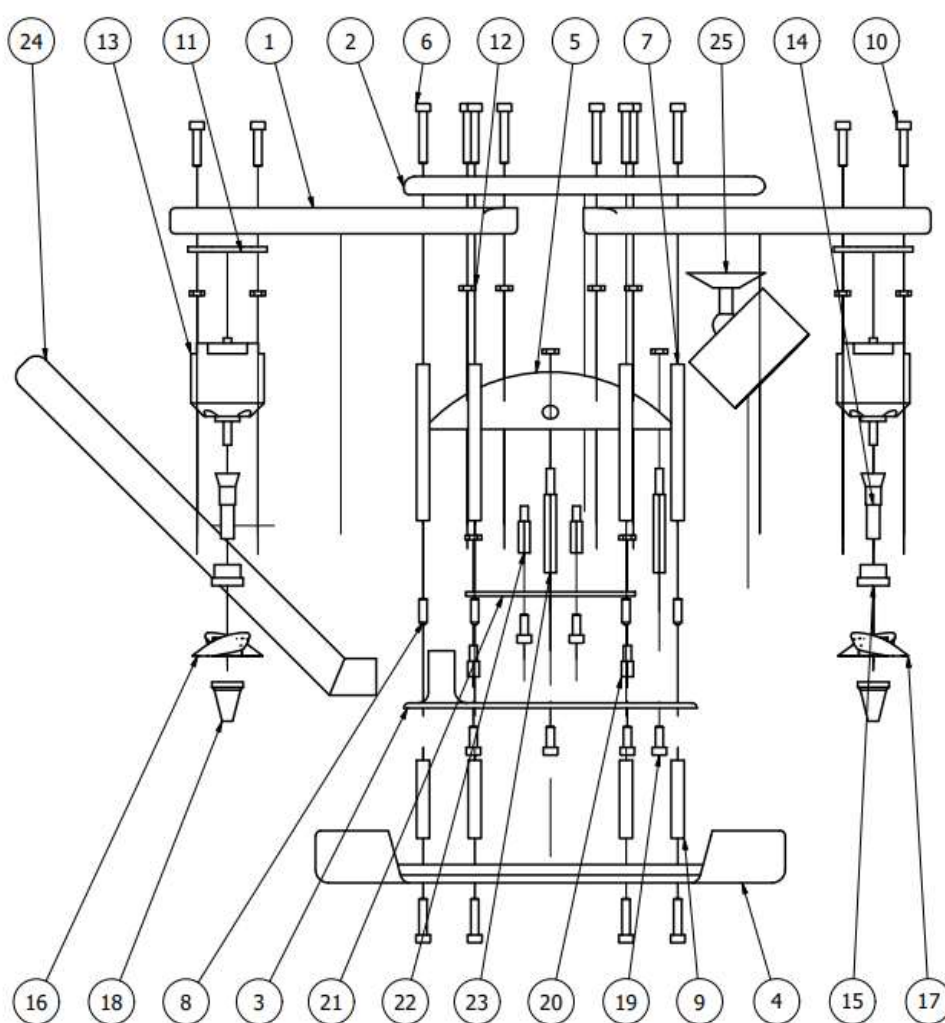
ANEJOS

DISEÑO E IMPRESIÓN 3D DE UN MODELO DE
DRON PARA COMPETICIÓN



ANEJO I

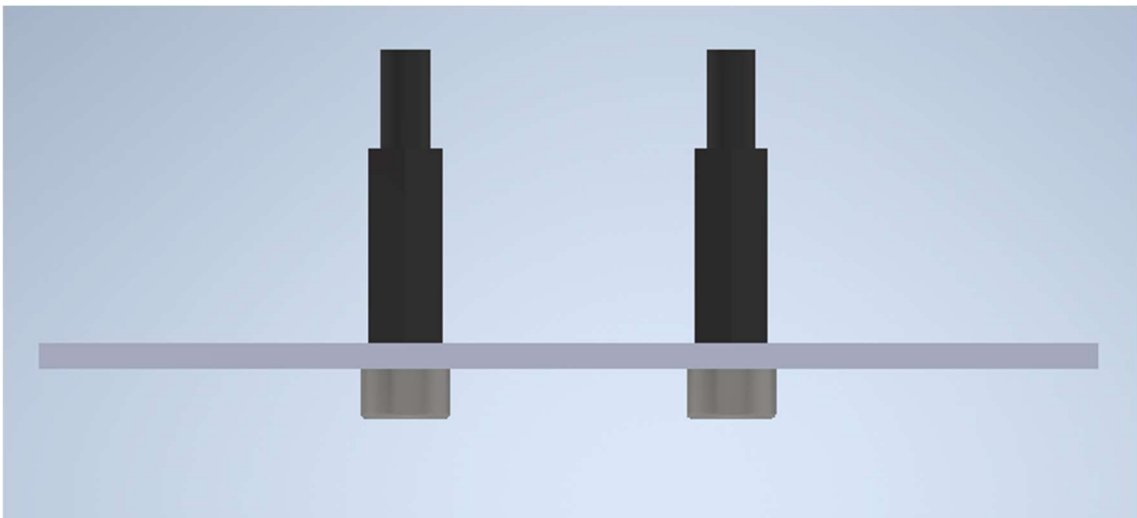
Se muestran, a continuación, las instrucciones del montaje del dron. Se facilita una captura del propio plano por motivos de comodidad para el lector. Así mismo, por lo complejo que supone seguir las instrucciones en base únicamente a su descripción, se incluye, con cada fase, una imagen explicativa.



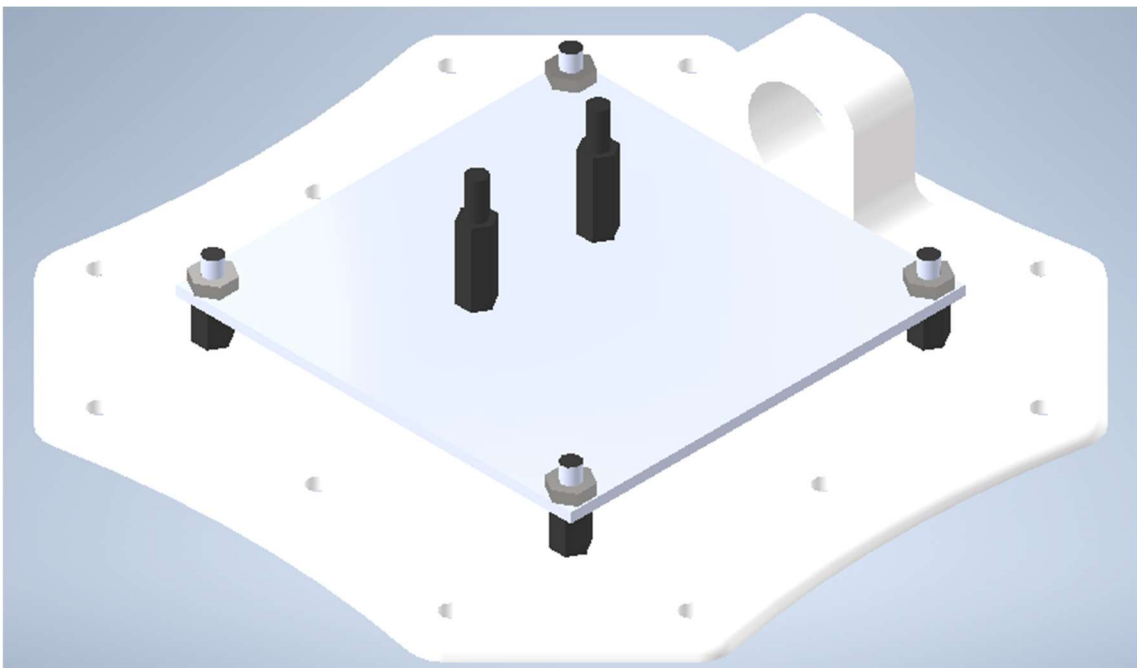
Fase 1: Se toman cuatro elementos 19 (Tornillo M3x8) y se introducen en los agujeros que forman un cuadrado en el elemento 3 (Soporte de la electrónica); entrando por la parte inferior, esto es, por la que no presenta el saliente. Para fijar los elementos 19, se enroscan en el lado contrario sendos elementos 20 (Pilar pequeño).



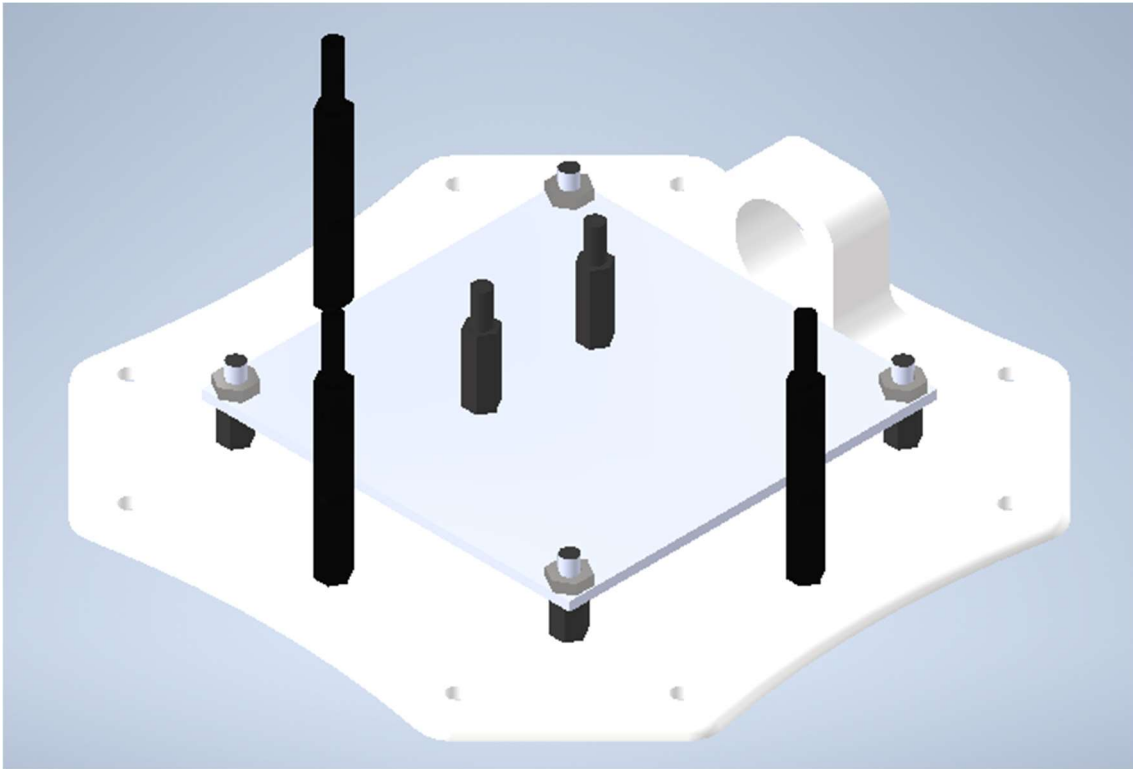
Fase 2: Se introducen, en los agujeros centrales del elemento 21 (PCB), dos elementos 19 que se atornillan a los elementos 22 (Pilar mediano).



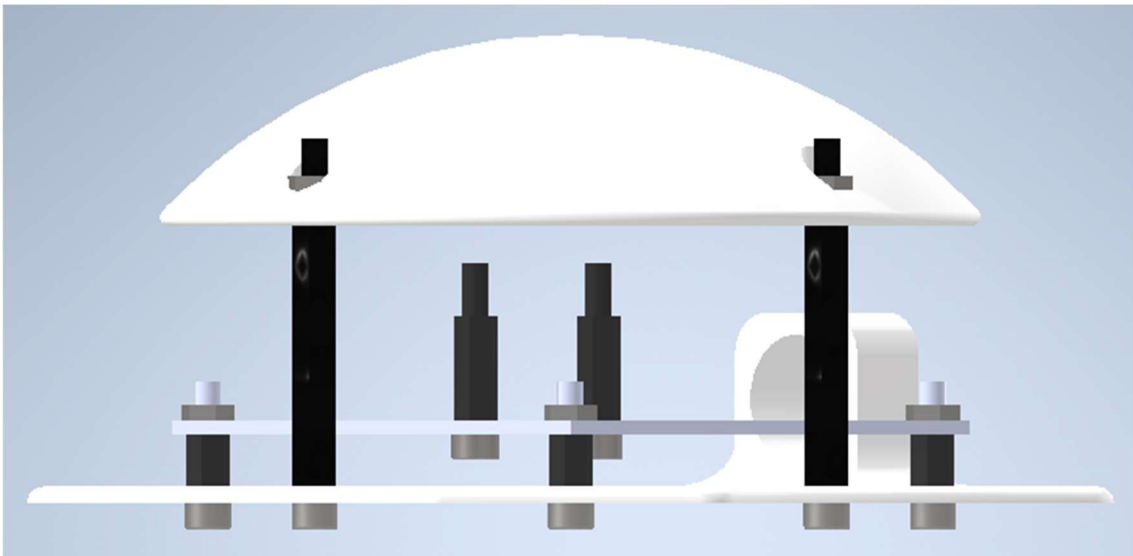
Fase 3: Sobre los elementos 20 mencionados en la Fase 1, se posiciona el elemento 21 de manera que los primeros se introducen en los agujeros de las esquinas del segundo. La unión se fija mediante elementos 12 (Tuerca M3).



Fase 4: Se introducen, en el elemento 3, tres elementos 19 adicionales; en los agujeros más próximos al elemento 21. Los tres se fijan mediante el uso de elementos 23 (Pilar grande).



Fase 5: Se coloca el elemento 5 (Cobertura de la electrónica) haciendo coincidir sus agujeros con las roscas de los elementos 23. La unión se consolida mediante el uso de tres elementos 12.



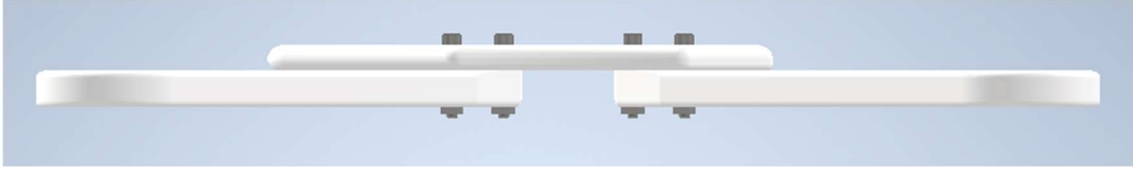
Fase 6: Se procede con la unión roscada entre los elementos 7 (Pilar estructural superior) y 9 (Pilar estructural inferior) mediante los elementos 8 (Tornillo prisionero M3x10) de manera que el elemento 3 se encuentre entre los dos primeros. Para ello, los elementos 8 han de situarse en los agujeros verticales que restan.



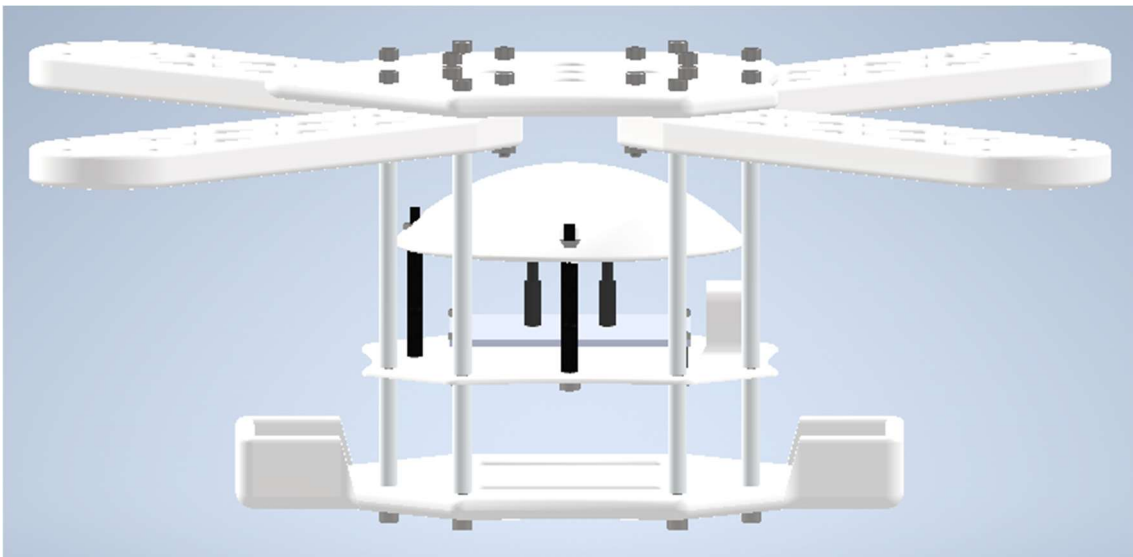
Fase 7: El elemento 4 (Soporte de la batería) se ensambla al conjunto; el elemento 4 debe quedar alineado con el segmento que forman los elementos 22 y con las paredes que sobresalen apuntando hacia el elemento 3. La unión se realiza mediante el roscado de los elementos 10 (Tornillo M3x14) en los elementos 9, pasando estos primeros a través de los agujeros presentes en el elemento 4.



Fase 8: Se introducen, en el elemento 2 (Marco), ocho elementos 6 (Tornillo M3x20) en los agujeros interiores respecto huecos rectangulares. Los cuatro elementos 1 (Ala) se posicionan en cada uno de estos grupos y se fijan mediante el roscado de elementos 12. Las cabezas de los elementos 6 han de quedar en la parte en que el diámetro de los agujeros cónicos sea menor.



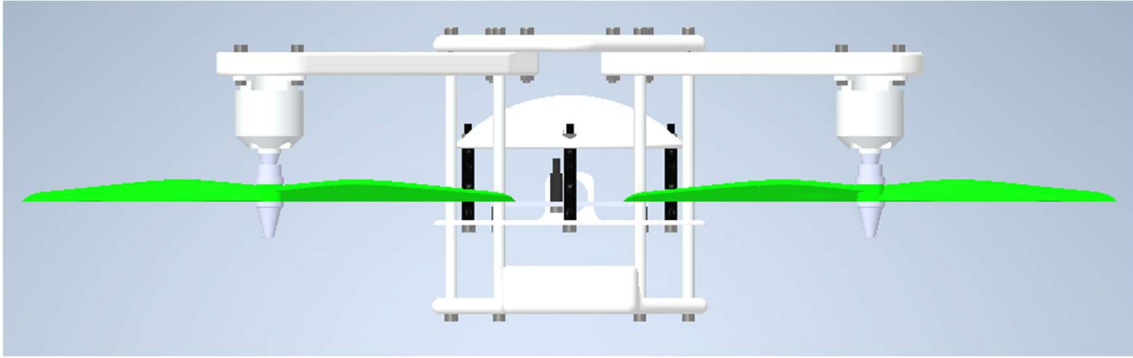
Fase 9: El conjunto formado en la Fase 8 se coloca sobre el ensamblaje de forma que la punta del elemento 2 se sitúe lo más alejada posible de la protuberancia del elemento 3. Haciendo coincidir los agujeros de los elementos 1 y los propios a los elementos 7, se procede al roscado mediante elementos 10.



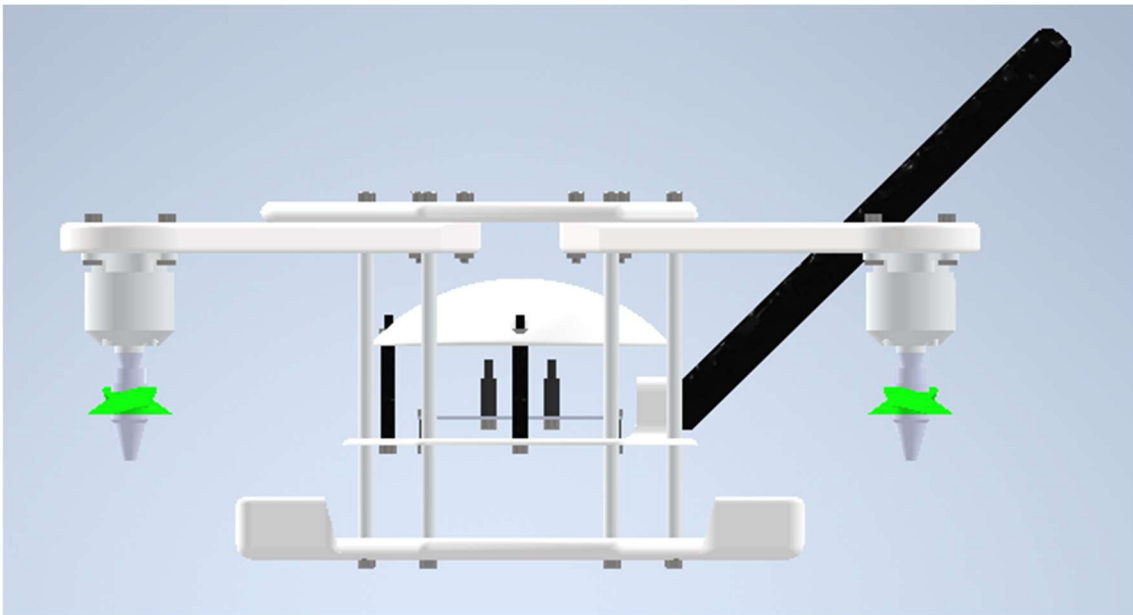
Fase 10: Se montan los conjunto hélice-motor tal y como especifica su manual; dos conjuntos deben montarse con elementos 17 (Hélice horaria) y los otros dos con elementos 16 (Hélices anti-horarias). El resto de los elementos que conforman el conjunto son: 11 (Soporte motor), 13 (Motor), 14 (Soporte hélice base), 15 (Soporte hélice intermedio) y 18 (Soporte hélice punta).



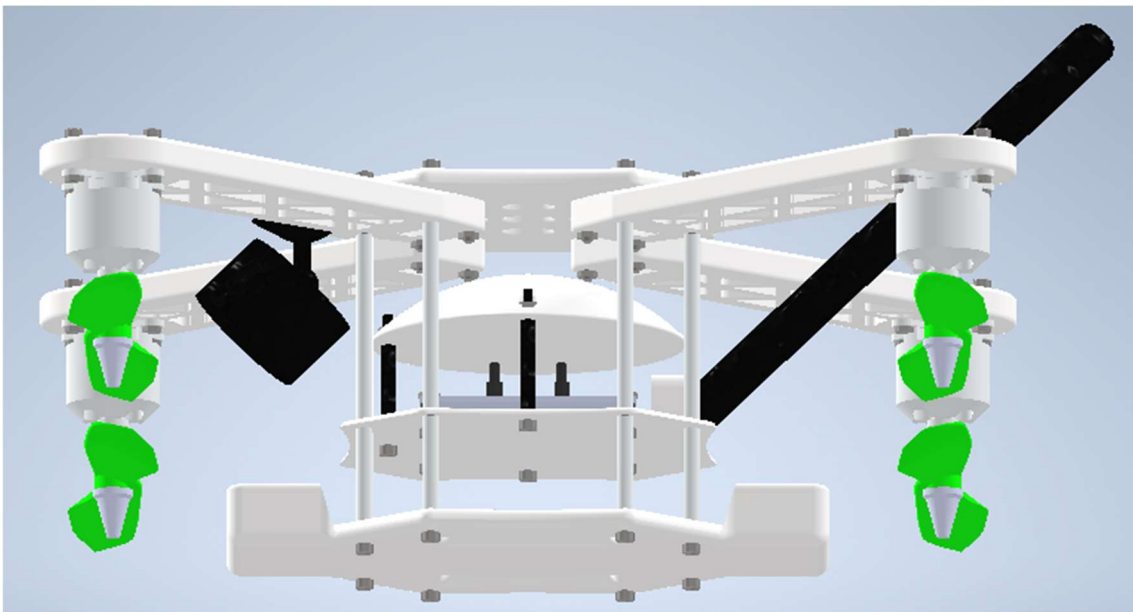
Fase 11: Cada conjunto hélice-motor se coloca bajo uno de los elementos 1 de manera que aquellos con hélices iguales se encuentren en contraposición. Se emplean elementos 10, con las cabezas en la parte superior, y elementos 12 roscados en estos para consolidar la unión.



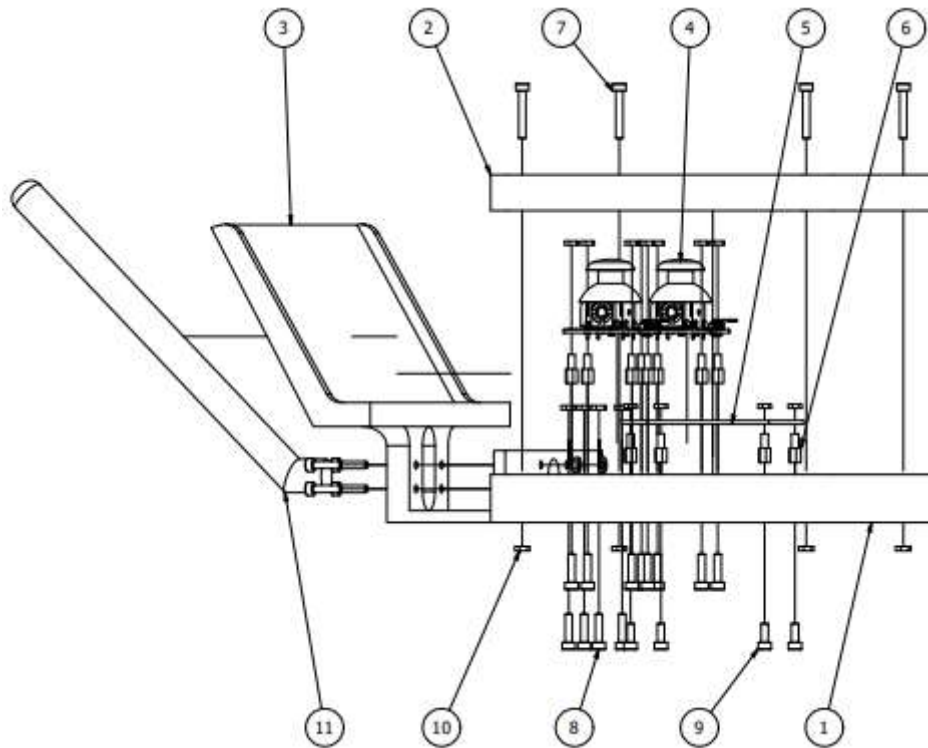
Fase 12: El elemento 24 (Antena) se inserta en el agujero transversal del elemento 3. En estas circunstancias, se asegura la unión mediante pegamento termofusible.



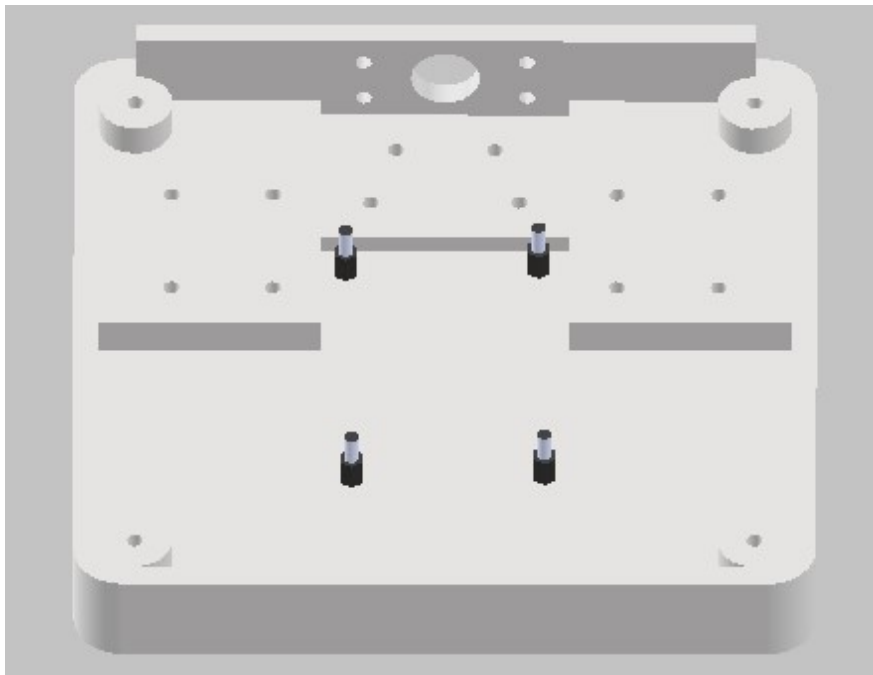
Fase 13: El elemento 25 (Cámara) se pega en la parte inferior del elemento 2, concretamente, en la punta que sobresale y evitando la colisión con el elemento 5.



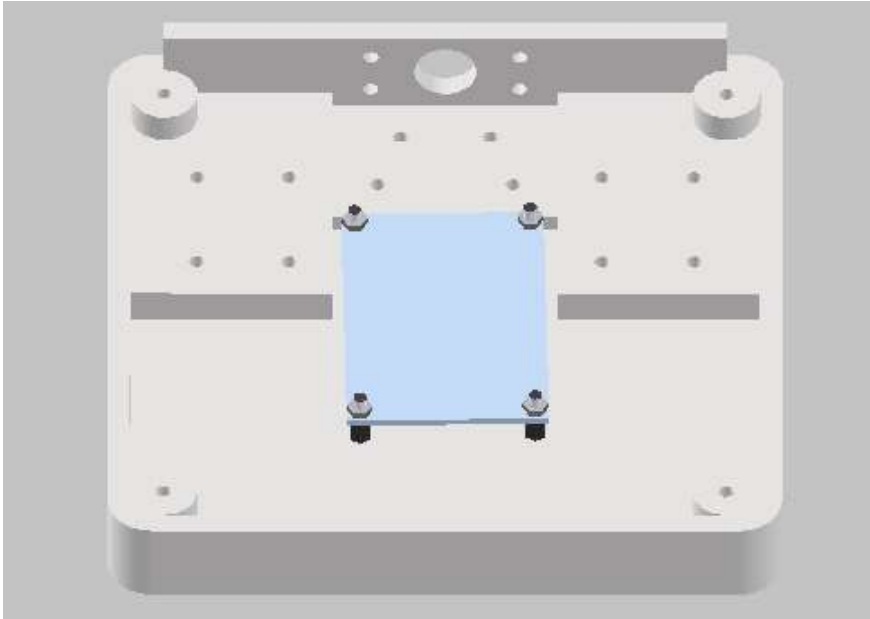
Respecto al mando, se procede de la misma manera que con la aeronave. En primer lugar, se muestra el diagrama obtenido del plano que muestra el ensamblaje completo y, después, se completa la información de cada fase con una imagen.



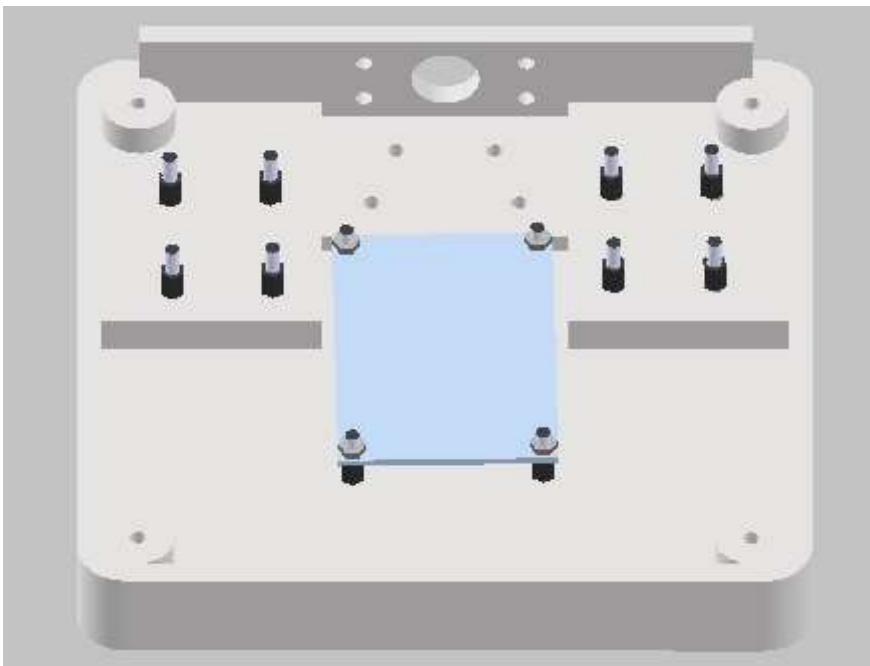
Fase 1: Al elemento 1 (Base del mando) se le introducen los cuatro elementos 9 (Tornillo M3x8) a través de los agujeros situados en la zona más baja de la superficie interior. Dentro de la caja se enroscan los elementos 6 (Pilar pequeño).



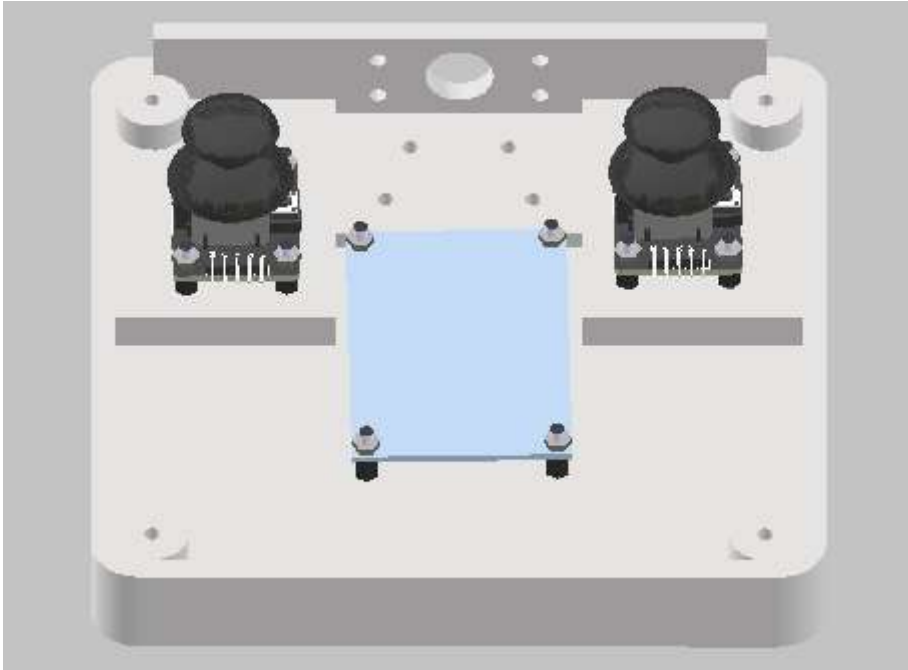
Fase 2: Se coloca el elemento 5 (PCB) sobre los elementos 6 ya colocados, de manera que estos se introduzcan por los agujeros habilitados para ello. El elemento 5 se fija mediante el uso de elementos 10 (Tuerca M3).



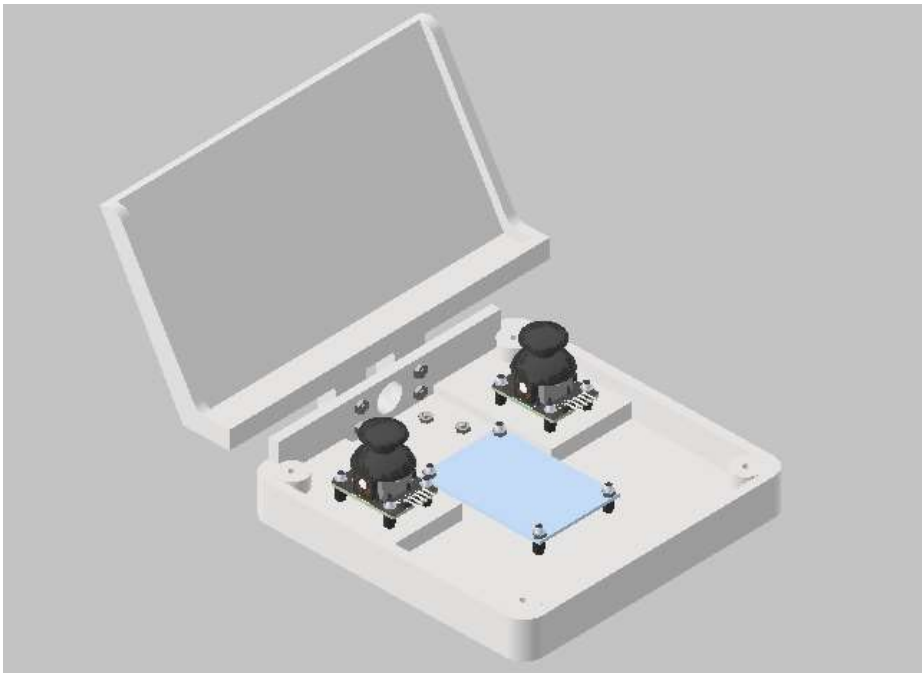
Fase 3: Se repite el proceso de la Fase 1 introduciendo, en esta ocasión, ocho elementos 8 (Tornillo M3x12) a través de los agujeros que se encuentran en la zona más alta del interior. Dentro de la caja se vuelven a enroscar los elementos 6.



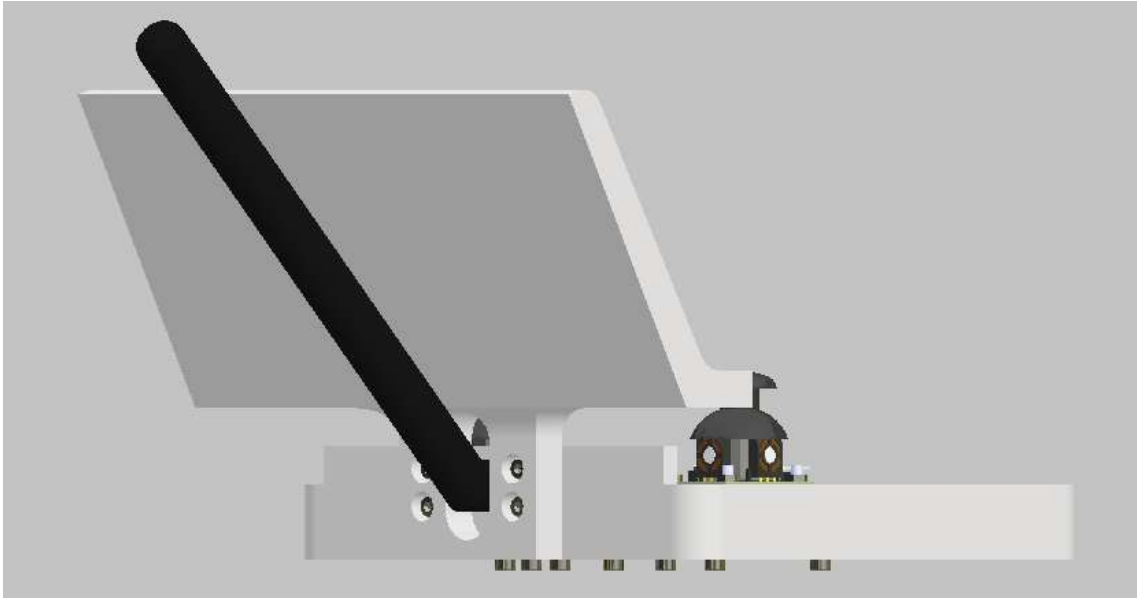
Fase 4: Sobre los elementos 6 fijados en la fase anterior se colocan los elementos 4 (Joystick) de manera que las patillas queden en dirección al escalón hacia la superficie más baja. Los elementos 6 han de quedar introducidos en los agujeros, asegurando la unión con elementos 10.



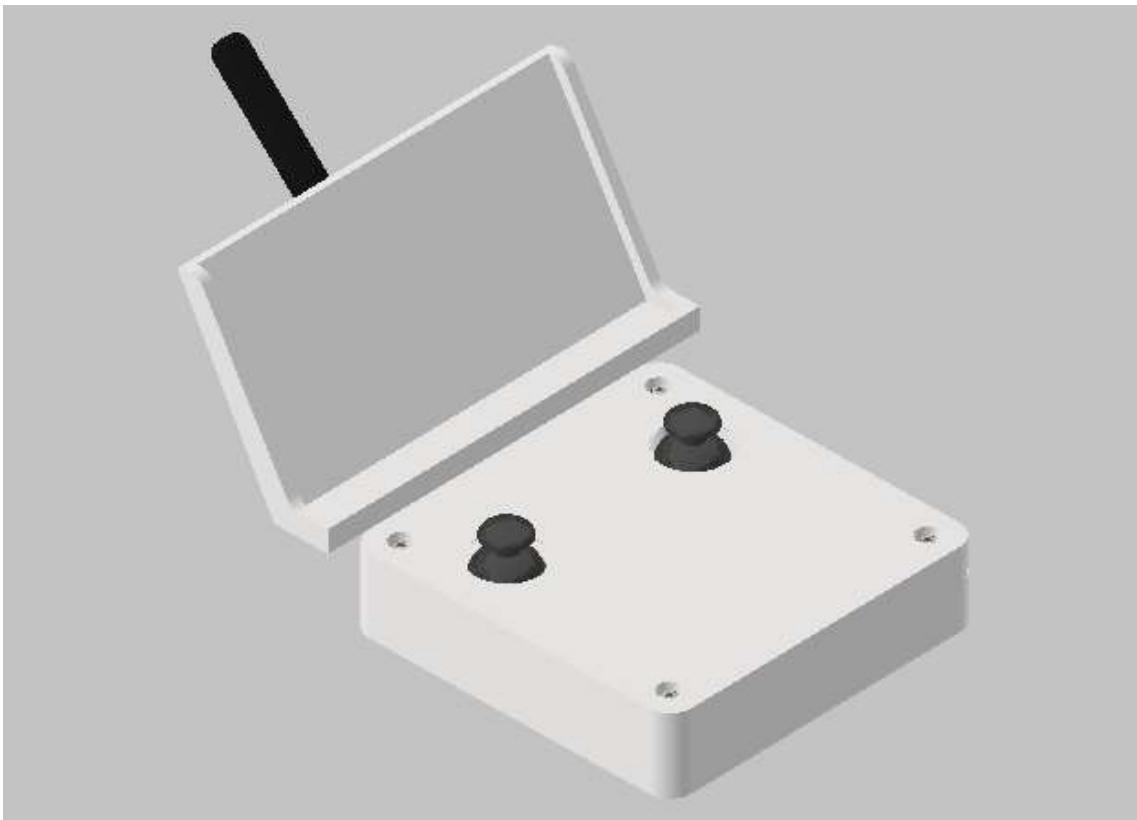
Fase 5: Se introduce la parte plana del elemento 3 (Soporte para el teléfono móvil) en la concavidad inferior del elemento 1 y se fija la unión mediante ocho elementos 8 y 10; la cabeza de los elementos 8 han de quedar sobre la superficie del elemento 3.



Fase 6: El elemento 11 (Antena) se inserta en el agujero transversal del elemento 1. En estas circunstancias, se asegura la unión mediante pegamento termofusible.



Fase 7: El elemento 2 (Cobertura del mando) se encaja sobre el elemento 1 y se fija mediante elementos 7 (Tornillo M3x20) y elementos 10.



ANEJO II

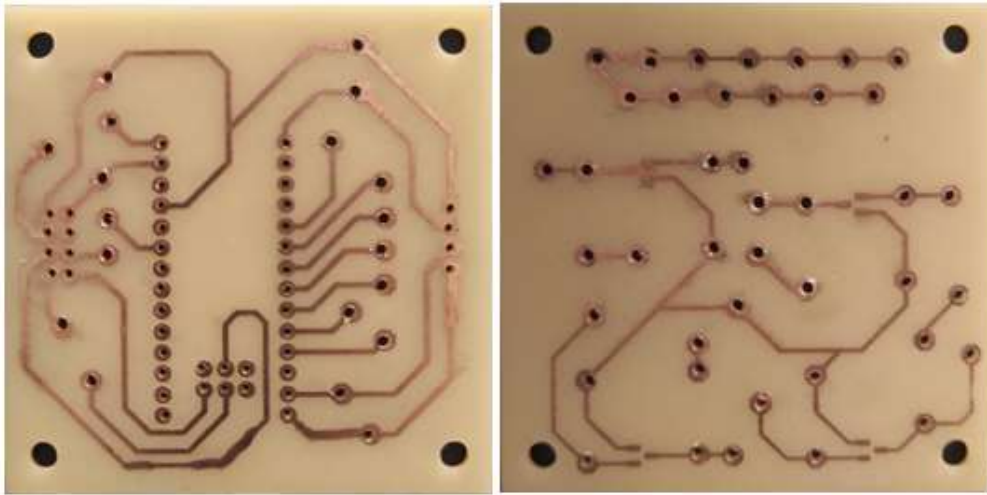
Con respecto a la versión alfa del prototipo, se han producido cambios tanto en lo referente a los componentes empleados como al diseño estructural. En primer lugar, se describen los dispositivos electrónicos que habían sido seleccionados para esta versión, así como la PCB diseñada; después, se comentan las principales modificaciones en la geometría y como estas afectan a la velocidad terminal.

Los componentes que han sido sustituidos han sido la radio, la IMU y los *level shifter*; para la primera, se había decidido emplear el módulo SX1278 ra-01 con una antena de espira para abaratar costes. El motivo de haber considerado la tecnología LoRa, en la que se adscribe el componente mencionado, eran las largas distancias que permitía cubrir. Finalmente, fue descartado porque la velocidad de comunicación podría haber comprometido la respuesta de la aeronave.

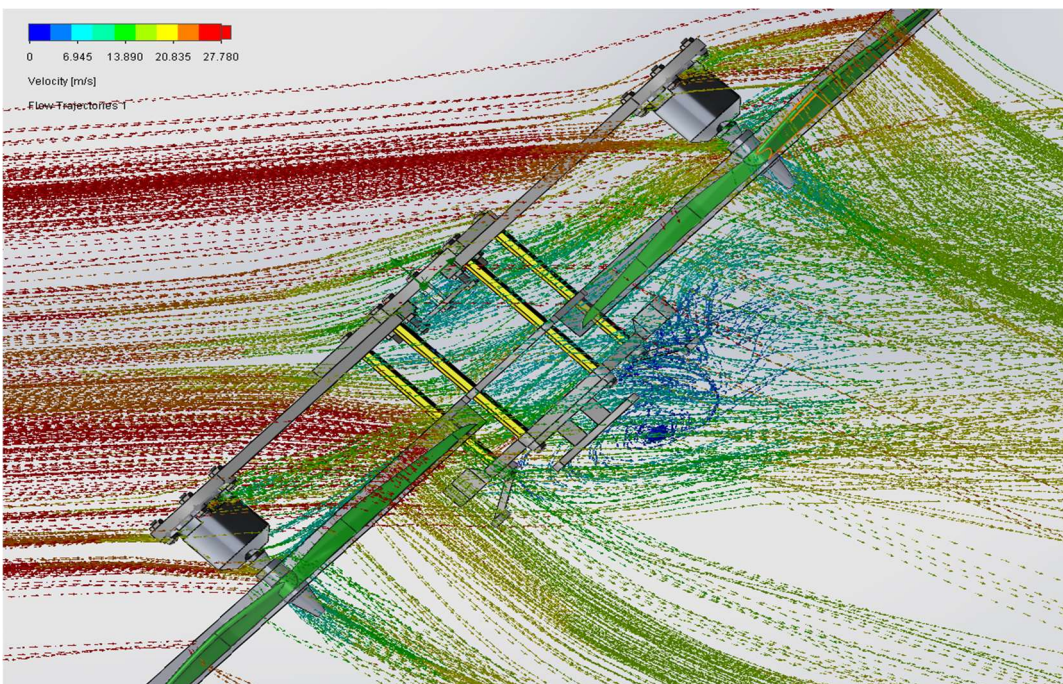
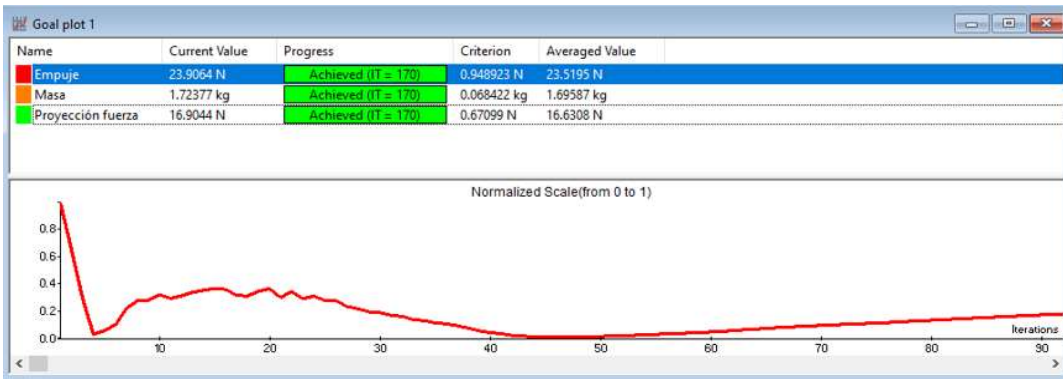
Por parte de la IMU, se había seleccionado el dispositivo MPU6050, que consta de 6 grados de libertad. Si bien en las pruebas realizadas de forma individual sobre el componente no revelaron incapacidad para cumplir su función, la mejora que supone una fusión de tres sensores en lugar de dos junto con el bajo coste que acarrea el cambio propiciaron la sustitución.

En lo que se refiere a la PCB, el cambio es notorio. En la versión alfa, el hecho de no contratar los servicios de una empresa para que las produjeran resultaba más barato, pero mucho más costoso; además, puesto que los medios de que se disponía eran limitados, no era posible crear pistas en ambas caras ni superficies equipotenciales. Esto, sumado a que los *level shifter* no estaban integrados en un componente único, sino que se habían configurado mediante resistencias y transistores, conllevaba la necesidad de emplear dos PCB para las conexiones de la aeronave.

Añadido a esto, estaba el hecho de que en un principio no se empleaba ningún elemento para controlar la tensión desde la batería; es por ello que el regulador de voltaje no se incluye en las PCB, sino que se conectó mediante cable rígido. A continuación se presentan las PCB empleadas, donde se puede apreciar el hecho de que son un poco rudimentarias. La de la izquierda corresponde al microcontrolador, el barómetro y la radio; también consta de *pads* libres para la conexión entre placas y con la IMU, que en esta versión estaba separada del resto de la electrónica. La de la derecha está preparada para alojar, principalmente, las resistencias y transistores que permitían los cambios de nivel; también constaba de un conjunto de *pads* cuyo objetivo era permitir la conexión a masa de forma sencilla.



En referencia al diseño estructural, la antigua distancia entre motores opuestos era de 350mm. Los estudios aerodinámicos fueron mucho más limitados y no se consideró modificar el ángulo de ataque; la velocidad terminal alcanzada fue de 100km/h. En las siguientes imágenes se muestran los resultados de los estudios realizados y una representación del flujo en dichas condiciones. Cabe destacar que el dron era más pesado en esta versión; también que las patas fueron eliminadas por su complejidad en cuanto a la impresión y poca utilidad.



ANEJO III

En referencia al *software*, se ha tratado de referenciar de manera clara la utilidad de la mayor parte del programa mediante comentarios. Se muestra, a continuación, el código propio a la aeronave:

```
//Librerías
#include <Arduino.h>
#include <Wire.h>
#include <Math.h>
#include <RH_RF69.h> //Librería específica del módulo de radio
#include <SPI.h> //Librería para comunicación SPI con periféricos
#include <RHReliableDatagram.h> //Librería para protocolo de comunicación
#include "I2Cdev.h" //Librería para comunicación I2C con periféricos
#include "MPU9250.h" //Librería para la lectura de la Unidad de Medición
Inercial
#include "SFE_BMP180.h" //Librería para la lectura del barómetro
#include "PIDController.hpp" //Librería para facilitar el uso de PID
//Framework FreeRTOS
#include <freertos/FreeRTOS.h>
#include <freertos/task.h>
#include <freertos/semphr.h>
#include <driver/ledc.h> //MCPWM sería la librería idónea pero solo
admite dos motores. Válida gracias a que los ESC actúan de intermediarios
#include "esp_timer.h" //Permite obtener el tiempo de ejecución en
microsegundos

//Definiciones
#define UASID 1 //Identificación de la radio de la aeronave
#define CS 5 //Pin de selección de dispositivo en comunicación SPI
#define IRQ 27 //Pin de interrupción de la radio
#define PM1 17 //Pin motor 1
#define PM2 16 //Pin motor 2
#define PM3 4 //Pin motor 3
#define PM4 2 //Pin motor 4
#define FREQ 500 //Frecuencia de la señal PWM
#define RES 15 //Resolución del PWM
#define BAT 25 //Pin batería
#define LED 26 //Pin LED
#define gconv PI/(131.0*180.0) //conversión de LSB por grado/segundo a
rad/s
#define MKp 30.0 //Definición del parámetro proporcional del filtro de
Mahony, valor recomendado por el creador
```

```
#define MKi 0.0 //Definición del parámetro integral del filtro de Mahony,
valor recomendado por el creador
//Valores máximos y mínimos que aceptan los ESC, vienen dados por el
protocolo estándar
#define PWM_min 1000
#define PWM_max 2000
#define ang_lim 30 //Definido en el estudio aerodinámico
#define w_lim 360 //Este dato se debería comprobar
#define w_yaw_lim 50 //Sería necesario comprobar si compromete la
comodidad del piloto
#define acc_lim 2 //Este dato se debería comprobar
#define min_throttle 1600
#define neutral_throttle 1780 //Ancho de pulso que mantiene el dron a
flote neutro a máxima batería
#define max_throttle 1900
#define sat 150
#define alt_max 120 //Límite legal
#define corr_lim 200
#define margen_acc 0.1 //Margen de error del acelerómetro. Por ajustar
#define exe_int 4 //Intervalo de ejecución de los PWM
#define MAX_TimeOut 5 //Tiempo en el que salta el Timeout en segundos.
Por ajustar

//Constructor de la radio y variables globales
RH_RF69 driver(CS, IRQ);
RHReliableDatagram manager(driver, UASID);
uint8_t buf[RH_RF69_MAX_MESSAGE_LEN];
uint8_t conf=0xFF; //1 lógico en byte
unsigned long Rahora, Rultimo;

//Constructor de los PID, actualmente controladores P de ganancia
unitaria a falta de ajustar mediante System Identification Toolbox
PID::PIDParameters<double> AngRoll(1,0,0);
PID::PIDController<double> PID_AngRoll(AngRoll);
PID::PIDParameters<double> AngPitch(1,0,0);
PID::PIDController<double> PID_AngPitch(AngPitch);
PID::PIDParameters<double> WRoll(1,0,0);
PID::PIDController<double> PID_WRoll(WRoll);
PID::PIDParameters<double> WPitch(1,0,0);
PID::PIDController<double> PID_WPitch(WPitch);
PID::PIDParameters<double> WYaw(1,0,0);
PID::PIDController<double> PID_WYaw(WYaw);
float output_pitch, output_roll, output_yaw, output_throttle;
PID::PIDParameters<double> Hover(1, 0, 0);
PID::PIDController<double> PID_Hover(Hover);
PID::PIDParameters<double> Emergencia(1, 0, 0);
PID::PIDController<double> PID_Emergencia(Emergencia);
float corr_alt;
```

```
//Constructor MPU9250 y variables globales. Falta calibrar
MPU9250 IMU;
I2Cdev I2C_M;
const float A_off[3]={1,1,1}; //Offset del acelerómetro
//Matriz de corrección del acelerómetro
const float A_inv[3][3]={
    {1,1,1},
    {1,1,1},
    {1,1,1}
};
const float G_off[3]={1,1,1}; //Offset del giroscopio
const float M_off[3]={1,1,1}; //Offset del magnetómetro
//Matriz de corrección del magnetómetro
const float M_inv[3][3]={
    {1,1,1},
    {1,1,1},
    {1,1,1}
};
/*const float M_dec=1.9;*/ //Declinación magnética en Valencia (positiva
este), en principio no es necesaria
int16_t ax, ay, az;
int16_t gx, gy, gz;
int16_t mx, my, mz;
float q[4]={1.0, 0.0, 0.0, 0.0}; //Vector que representa el cuaternión
float pitch, roll/*, yaw*/, pitch_rate, roll_rate, yaw_rate; //Ángulos de
Euler y velocidades angulares, yaw se omite por no ser necesario
float Axyz[3]; //Aceleración del sistema en ejes no inerciales
float Acc[3]; //Aceleración del sistema en ejes inerciales

//Constructor BMP180 y variables globales
SFE_BMP180 bmp180;
double Pi; //Presión inicial
char status;

//Otras variables globales
bool TO=false; //Variable para indicar que se está en TimeOut
bool first=true; //Variable que indica que se entra por primera vez al
bucle de aterrizaje
float bat, bat_corr; //Estado de la batería y corrección en us del
throttle para dicho estado
float ESC1, ESC2, ESC3, ESC4; //Micros a enviar a cada motor en canales
PWM
//Estructura puente entre datos de la radio y variables internas
typedef struct{
    uint8_t data1;
    uint8_t data2;
    uint8_t data3;
    uint8_t data4;
}RCref_struct;
```

```
RCref_struct *rc_ref;
uint8_t rc_ref_throttle=100, rc_ref_pitch, rc_ref_roll, rc_ref_yaw;
//Referencias del radio control
float ref_throttle, ref_pitch, ref_roll, ref_yaw; //Referencias
procesadas por el filtro EMA
bool alt_or=false; //Variable de override para la altitud máxima
alcanzada

//Declaración de funciones
void recibir();
void EMA();
void MPU();
void MahonyQuaternionUpdate(float ax, float ay, float az, float gx, float
gy, float gz, float mx, float my, float mz, float deltat);
void acel();
void PIDs();
void modular();
void BMP();
void bateria();
void aterrizaje();

//Declaración de semáforos
SemaphoreHandle_t sem_esc1;
SemaphoreHandle_t sem_esc2;
SemaphoreHandle_t sem_esc3;
SemaphoreHandle_t sem_esc4;
SemaphoreHandle_t sem_bat;
SemaphoreHandle_t sem_ref;
SemaphoreHandle_t sem_alt;
SemaphoreHandle_t sem_timeout;
SemaphoreHandle_t sem_TO;
SemaphoreHandle_t sem_first;

//Tarea de parpadeo del led para indicar al usuario que la inicialización
ha sido un éxito
void blink(void* pvParameters){
    for(int i=0; i<5; i++){
        digitalWrite(LED, HIGH);
        vTaskDelay(pdMS_TO_TICKS(1000));
        digitalWrite(LED, LOW);
        vTaskDelay(pdMS_TO_TICKS(1000));
    }
    vTaskDelete(nullptr);
}

//Tarea del loop de control principal
void control(void* pvParameters){
    while(1){
```

```
    TickType_t aux=xTaskGetTickCount(); //cuando empieza el loop
    if(xSemaphoreTake(sem_timeout, portMAX_DELAY)==pdTRUE) //Si se da el
    Timeout, bloquea esta tarea
        xSemaphoreGive(sem_timeout);
    MPU();
    acel();
    PIDs();
    modular();
    vTaskDelayUntil(&aux,pdMS_TO_TICKS(exe_int)); //Delay hasta que se
    alcanza el límite de tiempo desde el inicio del loop
}
}

//Tarea de lectura del radio control
void radio(void* pvParameters){
    while(1){
        recibir();
        if(TO==false){
            EMA();
            vTaskDelay(pdMS_TO_TICKS(200));
        } else
            vTaskDelay(pdMS_TO_TICKS(10));
    }
}

//Tarea para controlar la altura máxima y evitar rebasar el límite legal
void altitud(void* pvParameters){
    while(1){
        if(xSemaphoreTake(sem_timeout, portMAX_DELAY)==pdTRUE) //Si se da el
        Timeout, bloquea esta tarea
            xSemaphoreGive(sem_timeout); //Puesto que va a intentar aterrizar,
            no tiene sentido ejecutar este control
        BMP();
        vTaskDelay(pdMS_TO_TICKS(1000));
    }
}

//Tarea para controlar la carga de la batería
void carga(void* pvParameters){
    while(1){
        bateria();
        if(xSemaphoreTake(sem_TO, portMAX_DELAY)==pdTRUE &&
        xSemaphoreTake(sem_first, portMAX_DELAY)==pdTRUE){
            if(first==false && TO==false){
                xSemaphoreGive(sem_TO);
                xSemaphoreGive(sem_first);
                vTaskDelay(pdMS_TO_TICKS(10));
            } else{ //Si se cumplen estas condiciones, se esta en aterrizaje de
            emergencia, conviene que el ciclo se ejecute más rápido
```

```
xSemaphoreGive(sem_T0);
xSemaphoreGive(sem_first);
vTaskDelay(pdMS_TO_TICKS(1000));
}
}
}
}

//Tareas que escriben los PWM
void PWM1(void *pvParameter){
    if(xSemaphoreTake(sem_esc1, portMAX_DELAY)==pdTRUE){
        ledcWrite(1, ESC1);
        xSemaphoreGive(sem_esc1);
    }
    vTaskDelay(pdMS_TO_TICKS(4));
}
void PWM2(void *pvParameter){
    if(xSemaphoreTake(sem_esc2, portMAX_DELAY)==pdTRUE){
        ledcWrite(2, ESC2);
        xSemaphoreGive(sem_esc2);
    }
    vTaskDelay(pdMS_TO_TICKS(4));
}
void PWM3(void *pvParameter){
    if(xSemaphoreTake(sem_esc3, portMAX_DELAY)==pdTRUE){
        ledcWrite(3, ESC3);
        xSemaphoreGive(sem_esc3);
    }
    vTaskDelay(pdMS_TO_TICKS(4));
}
void PWM4(void *pvParameter){
    if(xSemaphoreTake(sem_esc4, portMAX_DELAY)==pdTRUE){
        ledcWrite(4, ESC4);
        xSemaphoreGive(sem_esc4);
    }
    vTaskDelay(pdMS_TO_TICKS(4));
}

void setup() {
    Wire.begin();
    pinMode(LED,OUTPUT);
    digitalWrite(LED,LOW);

    //Inicializar la radio y las interrupciones
    bool aux=false;
    if (!manager.init()){ //RH_RF69::GFSK_Rb250Fd250 es la configuración
por defecto
        digitalWrite(LED,HIGH);
        aux=true;
    }
}
```



```
}
driver.setTxPower(17,true);//PRA=19.85-Perdidas cable. Bastante
razonable teniendo en cuenta que el cable es de 10cm
driver.RHModeRx;

//Inicializar el MPU9250
IMU.initialize();
if(!IMU.testConnection()){
    digitalWrite(LED,HIGH);
    aux=true;
}
//Configurar filtro pasa bajo del giroscopio a 42Hz
Wire.beginTransmission(0x68);
Wire.write(0x1A);
Wire.write(0x03);
Wire.endTransmission();

//Inicializar barómetro
double T, P;
if(!bmp180.begin()){
    digitalWrite(LED,HIGH);
    aux=true;
} else{
    status=bmp180.startTemperature();
    if(status!=0){
        delay(status);
        status=bmp180.getTemperature(T);
        if(status!=0){
            status=bmp180.startPressure(3);
            if(status!=0){
                delay(status);
                status=bmp180.getPressure(P,T);
                if(status!=0)
                    Pi=P;
            }
        }
    }
}
}

//Declaración de los motores
ledcSetup(1, FREQ, RES);
ledcAttachPin(PM1, 1);
ledcSetup(2, FREQ, RES);
ledcAttachPin(PM2, 2);
ledcSetup(3, FREQ, RES);
ledcAttachPin(PM3, 3);
ledcSetup(4, FREQ, RES);
ledcAttachPin(PM4, 4);
```

```
//Creación de semáforos. Mutex los que interactúan con los ESC porque
//en caso de Timeout podrían intentar escribirse a la vez
//desde control y aterrizaje; también los de variables booleanas
sem_esc1=xSemaphoreCreateMutex();
if(sem_esc1==NULL){
    digitalWrite(LED,HIGH);
    aux=true;
}
sem_esc2=xSemaphoreCreateMutex();
if(sem_esc2==NULL){
    digitalWrite(LED,HIGH);
    aux=true;
}
sem_esc3=xSemaphoreCreateMutex();
if(sem_esc3==NULL){
    digitalWrite(LED,HIGH);
    aux=true;
}
sem_esc4=xSemaphoreCreateMutex();
if(sem_esc4==NULL){
    digitalWrite(LED,HIGH);
    aux=true;
}
sem_bat=xSemaphoreCreateBinary();
if(sem_bat==NULL){
    digitalWrite(LED,HIGH);
    aux=true;
}
sem_ref=xSemaphoreCreateBinary();
if(sem_ref==NULL){
    digitalWrite(LED,HIGH);
    aux=true;
}
sem_alt=xSemaphoreCreateBinary();
if(sem_alt==NULL){
    digitalWrite(LED,HIGH);
    aux=true;
}
sem_timeout=xSemaphoreCreateBinary();
if(sem_timeout==NULL){
    digitalWrite(LED,HIGH);
    aux=true;
}
sem_TO=xSemaphoreCreateMutex();
if(sem_TO==NULL){
    digitalWrite(LED,HIGH);
    aux=true;
}
sem_first=xSemaphoreCreateMutex();
```

```
if(sem_first==NULL){
    digitalWrite(LED,HIGH);
    aux=true;
}

while(aux==true); //Si ha habido problemas durante las
inicializaciones, no permite ejecutar el resto

//Generación de una tarea auxiliar para indicar que todo está correcto
xTaskCreate(blink, "Parpadeo de led", 512, NULL, 1, NULL);
vTaskSuspend(blink);
digitalWrite(LED,HIGH);

//Para iniciar el programa es necesario verificar que el mando se ha
conectado
while(rc_ref_throttle>25){
    if(manager.available()){
        uint8_t len=sizeof(buf);
        uint8_t from;
        if(manager.recvfromAck(buf,&len,&from)){
            rc_ref=(RCref_struct *)buf;
            conf=htons(conf);
            manager.sendtoWait((uint8_t*)&conf,sizeof(conf),from);
            rc_ref_throttle=ntohs(rc_ref->data1);
        }
    }
}
digitalWrite(LED,LOW);

//Activar PID
PID_AngPitch.SetOutputLimits(-w_lim, w_lim, -ang_lim, ang_lim);
PID_AngPitch.TurnOn();
PID_AngRoll.SetOutputLimits(-w_lim, w_lim, -ang_lim, ang_lim);
PID_AngRoll.TurnOn();
PID_WPitch.SetOutputLimits(-sat, sat, -w_lim, w_lim);
PID_WPitch.TurnOn();
PID_WRoll.SetOutputLimits(-sat, sat, -w_lim, w_lim);
PID_WRoll.TurnOn();
PID_WYaw.SetOutputLimits(-sat, sat, -w_yaw_lim, w_yaw_lim);
PID_WYaw.TurnOn();
PID_Hover.SetOutputLimits(-corr_lim, corr_lim, -acc_lim, acc_lim);
PID_Hover.TurnOn();
PID_Emergencia.Setpoint=0.1;
PID_Emergencia.SetOutputLimits(-corr_lim, corr_lim);

//Activar las tareas
xTaskCreatePinnedToCore(control,"Gestion del loop de control", 4096,
NULL, 19, NULL, 1); //Prioridad 19 para que no sea interrumpida ni por
interrupciones
```

```
xTaskCreate(radio,"Lectura de la radio", 1024, NULL, 2, NULL); //No se obliga a que se ejecute en un core por si tiene que aterrizar, lo mismo que la siguiente
xTaskCreate(carga,"Control de la carga de la batería", 512, NULL, 1, NULL); //En ambos casos el loop de control no se esta ejecutando y el core 1 esta libre
xTaskCreatePinnedToCore(altitud,"Control de altitud de la aeronave", 512, NULL, 1, NULL, 0);
xTaskCreate(PWM1, "Actualización motor 1", 512, NULL, 20, NULL);
xTaskCreate(PWM2, "Actualización motor 2", 512, NULL, 20, NULL);
xTaskCreate(PWM3, "Actualización motor 3", 512, NULL, 20, NULL);
xTaskCreate(PWM4, "Actualización motor 4", 512, NULL, 20, NULL);
}

void loop() {
    //No debe utilizarse en FreeRTOS.
    vTaskDelete(NULL); //Para liberar espacio, se elimina la tarea loopTask inherente al sistema operativo
    //Esto no es problemático ya que la función setup() ya se ha ejecutado. El espacio se libera durante la ejecución de la tarea IDLE
}

//Función para leer la radio
void recibir(){
    //Obtener valores desde el bufer
    if(manager.available()){
        uint8_t len=sizeof(buf);
        uint8_t from;
        if(manager.recvfromAck(buf,&len,&from)){
            rc_ref=(RCref_struct *)buf;
            conf=htons(conf);
            manager.sendtoWait((uint8_t*)&conf,sizeof(conf),from);
            rc_ref_throttle=ntohs(rc_ref->data1);
            rc_ref_pitch=ntohs(rc_ref->data2);
            rc_ref_roll=ntohs(rc_ref->data3);
            rc_ref_yaw=ntohs(rc_ref->data4);
            Rultimo=esp_timer_get_time();
        }
    }
    Rahora=pdTICKS_TO_MS(xTaskGetTickCount());
    if((Rahora-Rultimo)*1.0e-3>MAX_TimeOut){
        if(TO==false){
            if(xSemaphoreTake(sem_timeout, portMAX_DELAY)==pdTRUE)
                TO=true;
            PID_Emergencia.TurnOn();
            PID_Hover.TurnOff();
        }
        aterrizaje();
    }else{
```

```

    TO=false;
    xSemaphoreGive(sem_timeout);
    PID_Emergencia.TurnOff();
    PID_Hover.TurnOn();
    first=true;
  }
}

//Función para aplicar filtro exponencial y convertir a parámetros para
el PID
void EMA(){
  static uint8_t rc_filt_throttle, rc_filt_pitch, rc_filt_roll,
rc_filt_yaw;

  //Alfa del filtro elegido en 0.15 para una respuesta rápida pero suave,
necesario verificar.
  rc_filt_throttle=0.85*rc_filt_throttle+0.15*rc_ref_throttle;
  rc_filt_pitch=0.85*rc_filt_pitch+0.15*rc_ref_pitch;
  rc_filt_roll=0.85*rc_filt_roll+0.15*rc_ref_roll;
  rc_filt_yaw=0.85*rc_filt_yaw+0.15*rc_ref_yaw;

  //La entrada es un byte y la salida depende de la variable controlada
  if(xSemaphoreTake(sem_ref, portMAX_DELAY)==pdTRUE){
    //Como el throttle es función de la batería se debe separar en dos
para poder operarlo en el PID como una corrección en caso de no ajustarse
a la aceleración
    if(xSemaphoreTake(sem_bat, portMAX_DELAY)==pdTRUE){
      if(rc_ref_throttle>=115 && rc_ref_throttle<=140){ //Supuesto de
flotación neutra
        ref_throttle=neutral_throttle+bat_corr;
        PID_Hover.Setpoint=0;
      } else{
        ref_throttle=map(rc_filt_throttle, 255, 0,
max_throttle+bat_corr, min_throttle+bat_corr);
        PID_Hover.Setpoint=map(rc_filt_throttle, 255, 0, -acc_lim,
acc_lim); //en m/s2
      }
      xSemaphoreGive(sem_bat);
    }
    ref_pitch=map(rc_filt_pitch, 255, 0, ang_lim, -ang_lim); //en °
    ref_roll=map(rc_filt_roll, 255, 0, ang_lim, -ang_lim); //en °
    ref_yaw=map(rc_filt_yaw, 255, 0, w_yaw_lim, -w_yaw_lim); //en °/s
    xSemaphoreGive(sem_ref);
  }

  if(xSemaphoreTake(sem_alt, portMAX_DELAY)==pdTRUE){
    //Si la altitud de vuelo ha superado el límite, se ejecuta un
override para disminuirla
    if(alt_or==true)

```

```
        ref_throttle=min_throttle;
        xSemaphoreGive(sem_alt);
    }
}

//Función lectura IMU
void MPU(){
    float temp[3];
    int i;
    float norm;
    float Gxyz[3];
    float Mxyz[3];
    unsigned long Mahora=0, Multimo=0; //Variables para el control del
tiempo de loop
    float deltat=0; //Incremento temporal del último loop
    IMU.getMotion9(&ax, &ay, &az, &gx, &gy, &gz, &mx, &my, &mz);

    //Obtener medida corregida del giroscopio
    Gxyz[0]=((float) gx-G_off[0])*gconv;
    Gxyz[1]=((float) gy-G_off[1])*gconv;
    Gxyz[2]=((float) gz-G_off[2])*gconv;
    pitch_rate=(Gxyz[0]*180)/PI;
    roll_rate=(Gxyz[1]*180)/PI;
    yaw_rate=(Gxyz[2]*180)/PI;

    //Obtener medida corregida del acelerómetro
    Axyz[0]=(float) ax;
    Axyz[1]=(float) ay;
    Axyz[2]=(float) az;
    for (i = 0; i < 3; i++)
        temp[i] = (Axyz[i] - A_off[i]);
    Axyz[0] = A_inv[0][0] * temp[0] + A_inv[0][1] * temp[1] + A_inv[0][2] *
temp[2];
    Axyz[1] = A_inv[1][0] * temp[0] + A_inv[1][1] * temp[1] + A_inv[1][2] *
temp[2];
    Axyz[2] = A_inv[2][0] * temp[0] + A_inv[2][1] * temp[1] + A_inv[2][2] *
temp[2];
    norm=sqrt(Axyz[0] * Axyz[0] + Axyz[1] * Axyz[1] + Axyz[2] * Axyz[2]);
    Axyz[0]/=norm;
    Axyz[1]/=norm;
    Axyz[2]/=norm;

    //Obtener medida corregida del magnetómetro
    Mxyz[0]=(float) mx;
    Mxyz[1]=(float) my;
    Mxyz[2]=(float) mz;
    for (i = 0; i < 3; i++)
        temp[i] = (Mxyz[i] - M_off[i]);
```

```

    Mxyz[0] = M_inv[0][0] * temp[0] + M_inv[0][1] * temp[1] + M_inv[0][2] *
temp[2];
    Mxyz[1] = M_inv[1][0] * temp[0] + M_inv[1][1] * temp[1] + M_inv[1][2] *
temp[2];
    Mxyz[2] = M_inv[2][0] * temp[0] + M_inv[2][1] * temp[1] + M_inv[2][2] *
temp[2];
    norm=sqrt(Mxyz[0] * Mxyz[0] + Mxyz[1] * Mxyz[1] + Mxyz[2] * Mxyz[2]);
    Mxyz[0]/=norm;
    Mxyz[1]/=norm;
    Mxyz[2]/=norm;

    //Aplicación del filtro de Mahony
    Mahora = esp_timer_get_time();
    deltat = (Mahora - Multimo) * 1.0e-6;
    Multimo = Mahora;
    MahonyQuaternionUpdate(Axyz[0], Axyz[1], Axyz[2], Gxyz[0], Gxyz[1],
Gxyz[2], Mxyz[1], Mxyz[0], -Mxyz[2], deltat);

    //Conversión a ángulos de Euler. Para la configuración dada y las
posibilidades del cuadricóptero no debería darse bloqueo de cardán
    roll = atan2((q[0] * q[1] + q[2] * q[3]), 0.5 - (q[1] * q[1] + q[2] *
q[2]));
    pitch = asin(2.0 * (q[0] * q[2] - q[1] * q[3]));
    /*yaw = atan2((q[1] * q[2] + q[0] * q[3]), 0.5 - (q[2] * q[2] + q[3]
* q[3]));
    yaw *= 180.0 / PI;*/
    pitch *= 180.0 / PI;
    roll *= 180.0 / PI;
    /*yaw = -yaw + M_dec;
    if(yaw<0) yaw += 360.0;
    if(yaw>360.0) yaw -= 360.0;*/
}

//Función filtro de Mahony de S. James Remington
void MahonyQuaternionUpdate(float ax, float ay, float az, float gx, float
gy, float gz, float mx, float my, float mz, float deltat)
{
    // Vector to hold integral error for Mahony method
    static float eInt[3] = {0.0, 0.0, 0.0};
    // short name local variable for readability
    float q1 = q[0], q2 = q[1], q3 = q[2], q4 = q[3];
    float norm;
    float hx, hy, bx, bz;
    float vx, vy, vz, wx, wy, wz;
    float ex, ey, ez;
    float pa, pb, pc;

    // Auxiliary variables to avoid repeated arithmetic
    float q1q1 = q1 * q1;

```

```

float q1q2 = q1 * q2;
float q1q3 = q1 * q3;
float q1q4 = q1 * q4;
float q2q2 = q2 * q2;
float q2q3 = q2 * q3;
float q2q4 = q2 * q4;
float q3q3 = q3 * q3;
float q3q4 = q3 * q4;
float q4q4 = q4 * q4;

// Reference direction of Earth's magnetic field
hx = 2.0f * mx * (0.5f - q3q3 - q4q4) + 2.0f * my * (q2q3 - q1q4) +
2.0f * mz * (q2q4 + q1q3);
hy = 2.0f * mx * (q2q3 + q1q4) + 2.0f * my * (0.5f - q2q2 - q4q4) +
2.0f * mz * (q3q4 - q1q2);
bx = sqrt((hx * hx) + (hy * hy));
bz = 2.0f * mx * (q2q4 - q1q3) + 2.0f * my * (q3q4 + q1q2) + 2.0f * mz
* (0.5f - q2q2 - q3q3);

// Estimated direction of gravity and magnetic field
vx = 2.0f * (q2q4 - q1q3);
vy = 2.0f * (q1q2 + q3q4);
vz = q1q1 - q2q2 - q3q3 + q4q4;
wx = 2.0f * bx * (0.5f - q3q3 - q4q4) + 2.0f * bz * (q2q4 - q1q3);
wy = 2.0f * bx * (q2q3 - q1q4) + 2.0f * bz * (q1q2 + q3q4);
wz = 2.0f * bx * (q1q3 + q2q4) + 2.0f * bz * (0.5f - q2q2 - q3q3);

// Error is cross product between estimated direction and measured
direction of the reference vectors
ex = (ay * vz - az * vy) + (my * wz - mz * wy);
ey = (az * vx - ax * vz) + (mz * wx - mx * wz);
ez = (ax * vy - ay * vx) + (mx * wy - my * wx);
if (MKi > 0.0f)
{
    eInt[0] += ex;        // accumulate integral error
    eInt[1] += ey;
    eInt[2] += ez;
    // Apply I feedback
    gx += MKi*eInt[0];
    gy += MKi*eInt[1];
    gz += MKi*eInt[2];
}

// Apply P feedback
gx = gx + MKp * ex;
gy = gy + MKp * ey;
gz = gz + MKp * ez;

// Integrate rate of change of quaternion

```



```

// small correction 1/11/2022, see
https://github.com/kriswiner/MPU9250/issues/447
gx = gx * (0.5*deltat); // pre-multiply common factors
gy = gy * (0.5*deltat);
gz = gz * (0.5*deltat);
float qa = q1;
float qb = q2;
float qc = q3;
q1 += (-qb * gx - qc * gy - q4 * gz);
q2 += (qa * gx + qc * gz - q4 * gy);
q3 += (qa * gy - qb * gz + q4 * gx);
q4 += (qa * gz + qb * gy - qc * gx);

// Normalise quaternion
norm = sqrt(q1 * q1 + q2 * q2 + q3 * q3 + q4 * q4);
norm = 1.0f / norm;
q[0] = q1 * norm;
q[1] = q2 * norm;
q[2] = q3 * norm;
q[3] = q4 * norm;
}

//Función para obtener la aceleración en los ejes inerciales
void acel(){
    //Matriz de rotación
    float R[3][3]={
        {1.0-2.0*(sq(q[2])+sq(q[3])),2*(q[1]*q[2]-q[0]*q[3]),2*(q[1]*q[3]-
q[0]*q[2])},
        {2*(q[1]*q[2]-q[0]*q[3]),1.0-2.0*(sq(q[1])+sq(q[3])),2*(q[2]*q[3]-
q[0]*q[1])},
        {2*(q[1]*q[3]-q[0]*q[2]),2*(q[2]*q[3]-q[0]*q[1]),1.0-
2.0*(sq(q[1])+sq(q[2]))}
    };

    //Conversión de ejes no inerciales a inerciales
    for (int i=0; i<3; i++){
        Acc[i]=0.0;
        for (int j=0; j<3; j++){
            Acc[i]+=R[i][j]*Axyz[j];
        }
    }
}

//Función ejecución de los PID
void PIDs(){
    //Referencias del mando
    if(xSemaphoreTake(sem_ref, portMAX_DELAY)==pdTRUE){
        output_throttle=ref_throttle;
        PID_AngPitch.Setpoint=ref_pitch;
    }
}

```

```
PID_AngRoll.Setpoint=ref_roll;
PID_WYaw.Setpoint=ref_yaw;
xSemaphoreGive(sem_ref);
}

//Entradas de los primeros PID
PID_AngPitch.Input=pitch;
PID_AngRoll.Input=roll;
PID_WYaw.Input=yaw_rate;

//Actualizar primeros PID y obtener sus salidas
PID_AngPitch.Update();
PID_AngRoll.Update();
PID_WYaw.Update();
output_pitch=PID_AngPitch.Output;
output_roll=PID_AngRoll.Output;
output_yaw=PID_WYaw.Output;

//Entradas a los segundos PID
PID_WPitch.Input=pitch_rate;
PID_WPitch.Setpoint=output_pitch;
PID_WRoll.Input=roll_rate;
PID_WRoll.Setpoint=output_roll;

//Actualizar segundos PID y obtener sus salidas
PID_WPitch.Update();
PID_WRoll.Update();
output_pitch=PID_WPitch.Output;
output_roll=PID_WRoll.Output;

//PID del throttle
PID_Hover.Input=Acc[2];
PID_Hover.Update();
corr_alt=PID_Hover.Output;
}

//Función para modular la señal PWM
void modular(){
    float aux[4];
    //Establecemos la contribución de cada salida de los PID a la señal de
    cada motor
    aux[0]=ref_throttle+output_pitch-output_roll+output_yaw+corr_alt;
    aux[1]=ref_throttle+output_pitch+output_roll-output_yaw+corr_alt;
    aux[2]=ref_throttle-output_pitch+output_roll+output_yaw+corr_alt;
    aux[3]=ref_throttle-output_pitch-output_roll-output_yaw+corr_alt;

    //Limitamos la señal dentro de los márgenes de funcionamiento
    aux[0]=constrain(aux[0], PWM_min, PWM_max);
    aux[1]=constrain(aux[1], PWM_min, PWM_max);
```

```
aux[2]=constrain(aux[2], PWM_min, PWM_max);
aux[3]=constrain(aux[3], PWM_min, PWM_max);

//Convertimos a valores admisibles para una resolución de 15bit
if(xSemaphoreTake(sem_esc1, portMAX_DELAY)==pdTRUE){
    ESC1=aux[0]*32.767/2.0;
    xSemaphoreGive(sem_esc1);
}
if(xSemaphoreTake(sem_esc2, portMAX_DELAY)==pdTRUE){
    ESC2=aux[1]*32.767/2.0;
    xSemaphoreGive(sem_esc2);
}
if(xSemaphoreTake(sem_esc3, portMAX_DELAY)==pdTRUE){
    ESC3=aux[2]*32.767/2.0;
    xSemaphoreGive(sem_esc3);
}
if(xSemaphoreTake(sem_esc3, portMAX_DELAY)==pdTRUE){
    ESC4=aux[3]*32.767/2.0;
    xSemaphoreGive(sem_esc3);
}
}

//Función lectura de la altitud
void BMP(){
    double T, P, A;
    status=bmp180.startTemperature();
    if(status!=0){
        delay(status);
        status=bmp180.getTemperature(T);
        if(status!=0){
            status=bmp180.startPressure(3);
            if(status!=0){
                delay(status);
                status=bmp180.getPressure(P,T);
                if(status!=0)
                    A=bmp180.altitude(P,Pi);
            }
        }
    }
}
if(xSemaphoreTake(sem_alt, portMAX_DELAY)==pdTRUE){
    if(A>alt_max)
        alt_or=true;
    else if(alt_or==true && A<alt_max)
        alt_or=false;
    xSemaphoreGive(sem_alt);
}
}

//Función para leer la batería y configurar la corrección del throttle
```

```
void bateria(){
    if(xSemaphoreTake(sem_TO, portMAX_DELAY)==pdTRUE &&
xSemaphoreTake(sem_first, portMAX_DELAY)==pdTRUE){
        if(first==true || TO==true){
            xSemaphoreGive(sem_TO);
            xSemaphoreGive(sem_first);
            float rawBAT=analogRead(BAT);
            bat=rawBAT/4095.0*13.3;
        }
    }

    //Override en caso de batería baja
    if(bat<10.2){
        vTaskDelete(control);
        vTaskDelete(radio);
        vTaskDelete(altitud);
        PID_Hover.TurnOff();
        PID_Emergencia.TurnOn();
        aterrizaje();
    } else if(bat>11.1){ //Corrección del throttle neutro en función del
voltaje de la batería
        if(xSemaphoreTake(sem_bat, portMAX_DELAY)==pdTRUE){
            bat_corr=bat*(-73.3)+2703-neutral_throttle;
            xSemaphoreGive(sem_bat);
        }
    } else{ //Actualmente linearizado en dos segmentos, comprobar si es
viable
        if(xSemaphoreTake(sem_bat, portMAX_DELAY)==pdTRUE){
            bat_corr=bat*(-83.3)+2815-neutral_throttle;
            xSemaphoreGive(sem_bat);
        }
    }
}

//Función para realizar un aterrizaje de emergencia
void aterrizaje(){
    //Comprueba el AHRS de la aeronave
    MPU();
    acel();

    //Aprovecha los PIDs para mantenerse estable mientras aterriza
    PID_AngPitch.Input=pitch;
    PID_AngPitch.Setpoint=0;
    PID_AngRoll.Input=roll;
    PID_AngRoll.Setpoint=0;
    PID_WYaw.Input=yaw_rate;
    PID_WYaw.Setpoint=0;
    PID_AngPitch.Update();
    PID_AngRoll.Update();
}
```

```
PID_WYaw.Update();
output_pitch=PID_AngPitch.Output;
output_roll=PID_AngRoll.Output;
output_yaw=PID_WYaw.Output;
PID_WPitch.Input=pitch_rate;
PID_WPitch.Setpoint=output_pitch;
PID_WRoll.Input=roll_rate;
PID_WRoll.Setpoint=output_roll;
PID_WPitch.Update();
PID_WRoll.Update();
output_pitch=PID_WPitch.Output;
output_roll=PID_WRoll.Output;

//Se sigue corrigiendo el throttle según la batería pero se establece
al mínimo
if(xSemaphoreTake(sem_bat, portMAX_DELAY)==pdTRUE){
    ref_throttle=min_throttle+bat_corr;
    xSemaphoreGive(sem_bat);
}

//Corrección de la velocidad de bajada mediante fusión de los datos del
acelerómetro y los del barómetro
double T, P, Aahora, Aultima;
float Accultima, deltaAlt, deltaAcc, deltat, vel;
unsigned long nuevo, ultimo;

status=bmp180.startTemperature();
if(status!=0){
    delay(status);
    status=bmp180.getTemperature(T);
    if(status!=0){
        status=bmp180.startPressure(3);
        if(status!=0){
            delay(status);
            status=bmp180.getPressure(P,T);
            if (status != 0)
                Aahora=bmp180.altitude(P,Pi);
        }
    }
}
nuevo=esp_timer_get_time();
if(first==true){
    first=false;
    deltaAlt=Aahora-0;
    deltaAcc=Acc[2]-0;
    deltat=nuevo-0;
    vel=0.5*deltaAcc*deltat+0.5*deltaAlt/deltat;
}
else{
```

```

    deltaAlt=Aahora-Aultima;
    deltaAcc=Acc[2]-Accultima;
    deltat=nuevo-ultimo;
    vel=0.5*deltaAcc*deltat+0.5*deltaAlt/deltat+vel; //Filtro
complementario, alfa por determinar experimentalmente
}
Aultima=Aahora;
Accultima=Acc[2];
ultimo=nuevo;
PID_Emergencia.Input=vel;
PID_Emergencia.Update();
corr_alt=PID_Emergencia.Output;

//Modular la señal
if(Aahora<=0 || (-margen_acc<Acc[2]<margen_acc && -0.2<vel<0.2)){ //Se
asume que ha aterrizado. Validez por comprobar
    if(xSemaphoreTake(sem_esc1, portMAX_DELAY)==pdTRUE){
        ESC1=0;
        xSemaphoreGive(sem_esc1);
    }
    if(xSemaphoreTake(sem_esc2, portMAX_DELAY)==pdTRUE){
        ESC2=0;
        xSemaphoreGive(sem_esc2);
    }
    if(xSemaphoreTake(sem_esc3, portMAX_DELAY)==pdTRUE){
        ESC3=0;
        xSemaphoreGive(sem_esc3);
    }
    if(xSemaphoreTake(sem_esc3, portMAX_DELAY)==pdTRUE){
        ESC4=0;
        xSemaphoreGive(sem_esc3);
    }
}
}else
    modular();
}

```

A continuación, se muestra también el código del mando:

```

//Librerías
#include <Arduino.h>
#include <RH_RF69.h> //Librería específica del módulo de radio
#include <SPI.h> //Librería para comunicación SPI con periféricos
#include <RHReliableDatagram.h> //Librería para protocolo de comunicación

#define X1 A1
#define Y1 A2

```

```
#define X2 A3
#define Y2 A4
#define RCID 2
#define UASID 1
#define CS 2
#define LED 13

//Constructor de la radio y variables globales
RH_RF69 driver(CS);
RHReliableDatagram manager(driver, RCID);
uint8_t buf[RH_RF69_MAX_MESSAGE_LEN];
uint8_t conf=0xFF; //1 lógico en byte

//Estructura puente entre datos de la radio y variables internas
typedef struct{
    uint8_t data1;
    uint8_t data2;
    uint8_t data3;
    uint8_t data4;
}RCref_struct;
RCref_struct *rc_ref;

//Otras variables globales
int cont1=0, cont2=0;
const unsigned long intervalo=1000;
unsigned long Mantes, Mahora;
int estado=LOW;
bool ok=true;

void setup(){
    pinMode(LED,OUTPUT);
    digitalWrite(LED,HIGH);

    //Inicializar la radio y las interrupciones
```

```
bool aux=false;
if (!manager.init()){
    digitalWrite(LED,LOW);
    aux=true;
}
driver.setTxPower(17,true); //PRA=19.85-Perdidas cable. Bastante
razonable teniendo en cuenta que el cable es de 5cm
driver.RHModeTx;

while(aux==true); //Si han habido problemas durante la inicialización,
no permite ejecutar el resto
}

void loop(){
    //Leer los joysticks y transformar le información a un formato que
    pueda usar la radio
    int aux[4];
    aux[0]=map(analogRead(Y1), 1023, 0, 255, 0);
    aux[1]=map(analogRead(X1), 1023, 0, 255, 0);
    aux[2]=map(analogRead(Y2), 1023, 0, 255, 0);
    aux[3]=map(analogRead(X2), 1023, 0, 255, 0);
    rc_ref->data1=htons(aux[0]);
    rc_ref->data4=htons(aux[1]);
    rc_ref->data2=htons(aux[2]);
    rc_ref->data3=htons(aux[3]);

    //La radio manda la información
    if(manager.sendtoWait((uint8_t*)&rc_ref, sizeof(rc_ref), UASID)){
        uint8_t len = sizeof(buf);
        uint8_t from;
        if(manager.recvfromAckTimeout(buf, &len, 2000, &from)){ //¿Hay
respuesta?
            cont1=0;
            uint8_t rec=(uint8_t) buf;
            if(ntohs(rec)!=0xFF) //¿Contesta correctamente?
                cont2++;
        }
    }
}
```



```
    else
        cont2=0;
    } else
        cont1++;
} else
    ok=false;

//Avisar si la radio funciona mal. No debe retrasarse el código por el
parpadeo del led
if(cont1>5 || cont2>=10 || ok==false){
    Mahora=millis();
    if (Mahora-Mantes>=intervalo){
        Mantes=Mahora;
        if (estado==LOW) {
            estado=HIGH;
        } else {
            estado=LOW;
        }
        digitalWrite(LED, estado);
    }
} else
    digitalWrite(LED, HIGH);
delay(100);
}
```