

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



**SISTEMA DE ASISTENCIA PARA ATERRIZAJE DE UN DRON
MULTIROTOR SOBRE UNA PLATAFORMA EN MOVIMIENTO
MEDIANTE EL USO DE PROCESAMIENTO DE IMÁGENES**

Tesis para obtener el título profesional de Ingeniero Mecatrónico

AUTOR:

Paolo Bryan Salazar Ayllón

ASESORES:

Mag. Ing. Carlos Saito Villanueva

Mag. Ing. Diego Martín Arce Cigüeñas

Lima, Enero, 2025


Informe de Similitud

Yo, Carlos Saito Villanueva, docente de la Facultad de Ciencias e Ingeniería de la Pontificia Universidad Católica del Perú, asesor de la tesis titulada Sistema de Asistencia para Aterrizaje de Un Dron Multirrotor sobre una Plataforma en Movimiento Mediante el Uso de Procesamiento de Imágenes, del autor Paolo Bryan Salazar Ayllón

dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 21%. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 10 febrero 2025.
- He revisado con detalle dicho reporte y la Tesis o Trabajo de Suficiencia Profesional, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha: San José, Costa Rica / 21 de febrero 2025.

Apellidos y nombres del asesor: <u>Saito Villanueva Carlos</u>	
DNI: 43657069	Firma: 
ORCID: 0000-0002-3884-7149	

RESUMEN

Este estudio resume el diseño de un módulo mecatrónico capaz de darle a un dron multirrotor la habilidad de realizar aterrizajes precisos y autónomos sobre plataformas que se encuentran en movimiento lineal y a velocidad constante.

Basado en el trabajo de investigación “Aterrizaje de precisión de un UAV sobre una plataforma móvil” (Salazar, 2020), se realizó el diseño electrónico y mecánico del sistema con los componentes y materiales seleccionados, la integración de estos; así como el diagrama de conexiones y los planos de fabricación y ensamblaje correspondientes, y una maqueta de la estructura.

También se realizó la lógica del procesamiento de imágenes con el código y pruebas correspondientes, además del diseño del control del dron. Se comprueba que es posible determinar el lugar del aterrizaje usando un arreglo de marcadores ArUco y de enviar controles de movimiento al dron con librerías disponibles para Python.

Finalmente, ya que los materiales, componentes y procesos de fabricación están disponibles en el mercado local e internacional, se logró detallar los costos de diseño y fabricación del sistema mecatrónico.

AGRADECIMIENTO

Quiero expresar mi más sincero agradecimiento a mi familia por su amor y apoyo incondicional, y a mis asesores por su invaluable guía y paciencia durante este proceso. Su dedicación y confianza fueron fundamentales para la realización de esta tesis.



ÍNDICE

RESUMEN	i
ÍNDICE DE TABLAS	v
ÍNDICE DE FIGURAS	vi
ÍNDICE DE ECUACIONES.....	viii
INTRODUCCIÓN	1
ANTECEDENTES	2
1.1 Marco teórico	2
1.2 Problemática.....	3
1.3 Alcance de la investigación y necesidades del usuario	3
1.4 Objetivos	4
1.4.1 Objetivo principal.....	4
1.4.2 Objetivos específicos.....	5
1.5 Metodología	5
DISEÑO MECÁNICO.....	6
2.1 Requerimientos mecánicos	6
2.2 Dimensionamiento mecánico.....	6
2.3 Presentación de componentes y materiales.....	8
2.3.1 Estructura del estabilizador.....	8
2.3.2 Amortiguadores de vibración.....	18
DISEÑO ELECTRÓNICO	19
3.1 Requerimientos electrónicos.....	19
3.2 Presentación de sensores y actuadores.....	19
3.2.1 Motores	20
3.2.2 Controlador del estabilizador de la cámara.....	25
3.2.3 Sensor de orientación.....	31
3.2.4 Controlador de vuelo.....	33
3.2.5 Cámara y controlador principal	33
3.2.6 Arquitectura del hardware.....	36
DISEÑO DEL CONTROL Y PROCESAMIENTO DE IMÁGENES.....	38
4.1 Requerimientos de control y procesamiento de imágenes.....	38
4.2 Operaciones del proceso del sistema.....	38
4.3 Lógica de control del sistema.....	39
4.4 Procesamiento de imágenes y control del dron.....	42
4.5 Integración del módulo mecatrónico	45
COSTOS DEL SISTEMA.....	48
5.1 Costos de los componentes electrónicos.....	48
5.2 Costos de materiales y fabricación.....	49
5.3 Costos totales	50
CONCLUSIONES	51
RECOMENDACIONES.....	53
REFERENCIAS.....	54

Anexo A: Norma Técnica Complementaria NTC 001-2015

Anexo B: OPAC tornillos según ISO 4762 (DIN 912)

Anexo C: Guía de usuario para tuneado de los motores y calibración del sensor IMU

Anexo D: Código de generación de marcadores ArUco

Anexo E: Código de procesamiento de imágenes y control del dron

Anexo F: Desarrollo de una maqueta de la estructura del estabilizador de la cámara

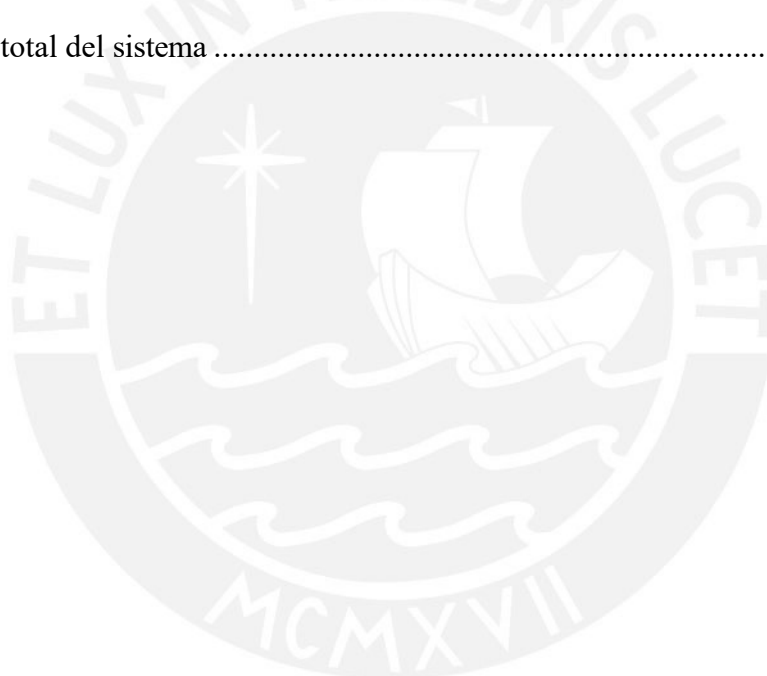
Anexo G: Costos

Anexo H: Planos



ÍNDICE DE TABLAS

Tabla 2.1 Lista de planos mecánicos.....	17
Tabla 3.1 Especificaciones del motor... ..	22
Tabla 3.2 Especificaciones del controlador AlexMos.....	25
Tabla 3.3 Especificaciones de OpenVM Cam RT1062.....	34
Tabla 4.1 Características más importantes del módulo mecatrónico... ..	47
Tabla 5.1 Costos de componentes electrónicos.....	48
Tabla 5.2 Costos de fabricación y diseño.....	49
Tabla 5.3 Costo total del sistema	50



ÍNDICE DE FIGURAS

Figura 1.1 Aterrizaje de un dron en un vehículo en movimiento.....	3
Figura 2.1 Dimensiones generales aproximadas del módulo mecatrónico	7
Figura 2.2 Modelado del módulo mecatrónico	8
Figura 2.3 Estructura de aluminio	8
Figura 2.4 Diagrama de cuerpo libre de la pieza 1	9
Figura 2.5 Tensión de Von Mises en la pieza 1	11
Figura 2.6 Deformaciones equivalente en la pieza 1	11
Figura 2.7 Desplazamientos en la pieza 1	12
Figura 2.8 Factor de seguridad en la pieza 1.....	12
Figura 2.9 Diagrama de cuerpo libre de la pieza 2	13
Figura 2.10 Tensión de Von Mises en la pieza 2.....	14
Figura 2.11 Deformaciones equivalente en la pieza 2	15
Figura 2.12 Desplazamientos en la pieza 2.....	15
Figura 2.13 Factor de seguridad en la pieza 2.....	16
Figura 2.14 Amortiguadores de caucho	18
Figura 3.1 Motores en el módulo mecatrónico	20
Figura 3.2 Motor T-MOTOR GB36-1	22
Figura 3.3 Dimensiones del motor	23
Figura 3.4 Unión atornillada de Brazo y Motor.....	24
Figura 3.5 Controlador AlexMos DV8313 Driver.....	25
Figura 3.6 Lazo de control del estabilizador de la cámara.....	26
Figura 3.7 Lazo de control del estabilizador de la cámara en SIMULINK.....	27
Figura 3.8 Respuesta en el tiempo del motor utilizando un controlador PID	27
Figura 3.9 Parámetros del controlador PID	27

Figura 3.10 Interfaz de computadora del firmware.....	29
Figura 3.11 Conexión del controlador con la batería.....	30
Figura 3.12 Conexión del controlador con los motores	30
Figura 3.13 Unión atornillada entre el controlador y el soporte mecánico del controlador	31
Figura 3.14 Sensor MPU 6050.....	31
Figura 3.15 Conexión del controlador con sensor IMU.....	32
Figura 3.16 Integración IMU y soporte de cámara.....	32
Figura 3.17 Controlador de vuelo PIXHAWK	33
Figura 3.18 OpenVM Cam RT1062	34
Figura 3.19 Integración de cámara a la estructura mecánica	35
Figura 3.20 Diagrama de la estructura del hardware	37
Figura 4.1 Diagrama de operaciones del sistema.....	39
Figura 4.2 Diagrama de flujo del sistema	40
Figura 4.3 Detalle del subproceso ‘Aterrizaje autónomo y preciso’	41
Figura 4.4 Ejemplos de reconocimiento de marcadores ArUco	42
Figura 4.5 Arreglo de marcadores en la plataforma de aterrizaje	43
Figura 4.6 Lógica del procesamiento de imágenes	44
Figura 4.7 Dron con el módulo mecatrónico integrado... ..	45
Figura 4.8 Módulo mecatrónico integrado... ..	46
Figura 4.9 Módulo mecatrónico integrado... ..	46

ÍNDICE DE ECUACIONES

Ecuación 2.3 Esfuerzo de flexión... ..	9
Ecuación 2.2 Esfuerzo de corte... ..	10
Ecuación 3.1 Momento de inercia... ..	21
Ecuación 3.2 Aceleración angular... ..	23



INTRODUCCIÓN

Los drones multirotor son naves aéreas no tripuladas que cuentan con 3, 4, 6 u 8 motores dispuestos de cierta forma que le brindan al dron ciertas habilidades en cuanto a estabilidad, velocidad, tiempo de vuelo, etc. Actualmente, no están diseñados para realizar aterrizajes autónomos y precisos sobre plataformas que se encuentran en movimiento horizontal, debido a la limitada precisión que ofrece el GPS y a que el modo autónomo de aterrizaje que poseen sólo controla la posición y velocidad vertical del dron, lo que limita el desempeño de estos en misiones cada vez más complejas.

El objetivo principal de este trabajo es diseñar un módulo mecatrónico que permita utilizar procesamiento de imágenes para suplir la limitación del GPS para permitir al dron realizar aterrizajes precisos y autónomos, considerando la posibilidad de que la plataforma de aterrizaje se encuentre en movimiento lineal y a velocidad constante. Esto a su vez, ampliaría el rango de tareas y complejidad de misiones que los drones pueden realizar; por ejemplo, realizar ciertas maniobras al detectar un código QR, al identificar el tipo de superficie sobre la que se encuentra volando o al detectar algún objeto en específico.

En el presente trabajo se diseña la parte mecánica, electrónica y de control, que son los 3 principales aspectos que comprende el diseño mecatrónico.

Para el desarrollo de la tesis se emplea la siguiente metodología que consta de 4 pasos:

1. El primer paso es buscar información de los componentes y materiales seleccionados en el trabajo de investigación previo, y realizar las actualizaciones y modificaciones necesarias.
2. El segundo paso es realizar el diseño mecánico, electrónico, de control y del procesamiento de imágenes.
3. El tercer paso es diseñar la integración total del sistema con el dron.
4. El cuarto paso es elaborar los planos correspondientes de cada diseño.

CAPÍTULO I

ANTECEDENTES

En este capítulo se presenta la problemática que incentiva el desarrollo del presente proyecto. Además, se define el alcance, objetivo principal, objetivos específicos y la metodología a seguir.

1.1 Marco teórico

En los últimos años, con el avance de la tecnología, se ha incrementado el desarrollo de drones tanto en el ámbito militar y civil dotando a estos de ciertas habilidades que les permiten realizar tareas específicas y cada vez más complejas. Un punto importante a mencionar es que desde hace varios años existe una mayor inversión en investigación en drones militares que en drones de uso civil (Gamarra, 2015). A pesar de esto los drones de uso civil tienen un amplio campo de actividades en las que pueden ser usados, tales como vigilancia (Torres, 2015), fotografía, búsqueda de personas, detección de defectos o fallas en estructuras; todo esto mediante la integración de diferentes tipos de cámaras y sensores.

En el aspecto marítimo las embarcaciones no pueden controlar el aterrizaje de este tipo de drones en una posición específica sin emplear recursos humanos, lo que es una fuente de riesgo (Raja, 2016; Parker, 2015). Y la propuesta de poder aterrizar el dron sobre un vehículo terrestre en movimiento abre nuevas posibilidades de implementar el uso de drones en cierto tipo de tareas. Este proyecto puede combinarse con algunos otros; por ejemplo, el delivery de paquetes por parte de Flytex que ya opera en Islandia haciendo entrega de paquetes de hasta 3 kg y de tamaño no mayor al de una caja de zapatos en zonas urbanas (FLYTREX). También está el gigante Google con su proyecto Wing que ya obtuvo permiso para operar en Australia (Porter, 2019). Y con los drones gigantes de Elroy Air que son capaces de transportar cargas de hasta 300 kg (Palou, 2017).

1.2 Problemática

Actualmente los drones multirrotor no están diseñados para realizar aterrizajes autónomos y precisos sobre plataformas que se encuentran en movimiento horizontal, debido a la limitada precisión que ofrece el GPS y a que el modo autónomo de aterrizaje que poseen sólo controla la posición y velocidad vertical del dron, lo que limita el desempeño de estos en misiones cada vez más complejas.

1.3 Alcance de la investigación y necesidades del usuario

Las principales necesidades del usuario son, en primer lugar, que los drones puedan aterrizar de manera completamente autónoma en el lugar específico que se le designa, y segundo, que este aterrizaje pueda realizarse con la plataforma en movimiento y así aprovechar de mejor manera la utilidad de los drones.



Figura 1.1 Aterrizaje de un dron en un vehículo en movimiento

Fuente: https://www.rinspeed.eu/en/Sigmatos_22_concept-car.html

Entonces el módulo mecatrónico a diseñar será capaz de aterrizar un dron de forma autónoma y precisa de forma segura. Además que, al ser un aterrizaje preciso, aumenta la seguridad de las personas que se encuentran en la cercanía durante el aterrizaje y esto permitiría realizar misiones en zonas urbanas o en las que el tiempo y el cuidado son puntos muy importantes, como los drones que transportarían medicina a ciertas zonas del país donde el acceso por otros medios es complicado, por ejemplo la Amazonía peruana como se propone en la tesis “diseño de un cuadricóptero para transporte de medicina en la amazonía peruana” (SALAZAR, 2017), lo que ayudaría a la población en momentos de urgencia. Finalmente, el diseño de este sistema podría facilitar la autorización para volar drones sobre zonas urbanas para ciertas tareas como se especifica en el párrafo 8, subpárrafo (f) y (g) de la Norma Técnica Complementaria NTC 001-2015, la cual se presenta en el anexo A, emitida por el Ministerio de Transporte y Comunicaciones (MTC), expuesta en la Ley 30740, que regula el uso de naves pilotadas a distancia (GUTIERREZ, 2015).

La investigación se limita al diseño mecatrónico de un módulo, que montado en el dron multirrotor, le dé a este la capacidad de aterrizar de forma precisa y autónoma sobre una plataforma en movimiento. Cumpliendo con los requisitos mecatrónicos, se presenta el diseño mecánico, análisis y planos correspondientes; el diseño electrónico con la selección de componentes, su integración y esquemáticos de conexiones; el diseño de la lógica del procesamiento de imágenes y control del dron, y la realización de pruebas y simulaciones para cada caso; y finalmente, se presenta la integración del propio sistema y su integración con el dron.

1.4 Objetivos

1.4.1 Objetivo principal

El objetivo principal es diseñar el módulo mecatrónico, basado en el concepto de solución óptimo presentado en el trabajo de investigación “Aterrizaje de precisión de un UAV sobre una plataforma móvil”, que permita el aterrizaje preciso y autónomo de un dron multirrotor sobre una plataforma que se encuentra en movimiento lineal y a velocidad constante mediante el procesamiento de imágenes adquiridas por una cámara estabilizada por un mecanismo diseñado especialmente para esta tarea.

1.4.2 Objetivos específicos

1. Diseñar el sistema mecánico y electrónico del módulo a montar en el dron, y elaborar los planos y diagramas correspondientes.
2. Diseñar la lógica y código del procesamiento de imágenes y control del dron, y realizar las pruebas y simulaciones respectivas.
3. Realizar la integración del propio sistema y su integración con el dron.
4. Detallar los costos del sistema.

1.5 Metodología

El desarrollo de la presente tesis se llevará a cabo de acuerdo con la siguiente metodología, basada en el Trabajo de investigación publicado en el repositorio de la universidad PUCP “Aterrizaje de precisión de un UAV sobre una plataforma móvil”, que consta de cuatro (04) pasos.

1. El primer paso es definir la lista de requerimientos generales, mecánicos, electrónicos, de control y económicos que debe cumplir el módulo mecatrónico a diseñarse.
2. Definir los procesos y diseñar la lógica de funcionamiento del módulo mecatrónico.
3. El tercer paso es seleccionar los componentes electrónicos y materiales mecánicos, así como diseñar la arquitectura del hardware.
4. El cuarto paso es diseñar la lógica del procesamiento de imágenes y conexión del módulo mecatrónico con el dron.

CAPÍTULO II

DISEÑO MECÁNICO

En este capítulo se trata el diseño mecánico del sistema; se presenta los requerimientos, dimensionamiento, componentes y materiales seleccionados y los análisis correspondientes.

2.1 Requerimientos mecánicos

En este segmento se definen los requerimientos que debe cumplir el módulo mecatrónico respecto al diseño mecánico:

- El módulo mecatrónico no deberá sobrepasar los 600 gramos de peso.
- El módulo mecatrónico no deberá exceder las siguientes dimensiones:
15 cm de alto, 12 cm de ancho y 12 cm de largo.
- El montaje y desmontaje del módulo no debe tardar más de 15 min.
- El módulo contará con un mecanismo de 2 grados de libertad.
- El mantenimiento del módulo se realizará cada 3 meses.
- El módulo debe poder instalarse en la parte inferior de la estructura del dron.

2.2 Dimensionamiento mecánico

Como se señala en la lista de requerimientos, el módulo mecatrónico no debe exceder las siguientes medidas: 15 cm de alto x 12 cm de ancho x 12 cm de largo, espacio disponible en drones multirrotor que poseen 6 u 8 motores. Los motores que se emplearán en este proyecto tienen aproximadamente las siguientes medidas: 3 cm de diámetro x 3 cm de largo. La cámara que se empleará posee aproximadamente las siguientes dimensiones: 6.23 cm de largo x 4.50 cm de alto x 3.3 cm de ancho. No es necesario considerar los demás componentes electrónicos ya que son piezas mucho más pequeñas, por lo tanto, no serán tomadas en cuenta para el dimensionamiento del módulo.

El presente módulo mecatrónico consta de 8 piezas mecánicas, de las cuales 6 son las más importantes al momento de considerar las dimensiones adecuadas para el módulo. Estas 6 piezas constan de 4 brazos y 2 soportes rectangulares. Los 4 brazos van conectados a los motores y no se requieren que sean muy grandes, ya que una de las funciones del presente módulo mecatrónico es mantener estable una cámara en posición horizontal y también se desea que trabajen con los menores esfuerzos de flexión y corte. Entonces, las piezas más grandes y en las cuales se basa el diseño y el deseo de cumplir con los requerimientos de dimensión son los 2 soportes rectangulares, los cuales estarán unidos a su vez por almohadillas anti-vibraciones; uno de estos soportes irá unido al dron y en el otro se colocará la tarjeta electrónica que servirá para controlar el funcionamiento de los motores.

Las dimensiones generales aproximadas del módulo a implementar en este proyecto y el modelado 3D realizado con el software INVENTOR se muestran a continuación en la figura 2.1 y figura 2.2 respectivamente.

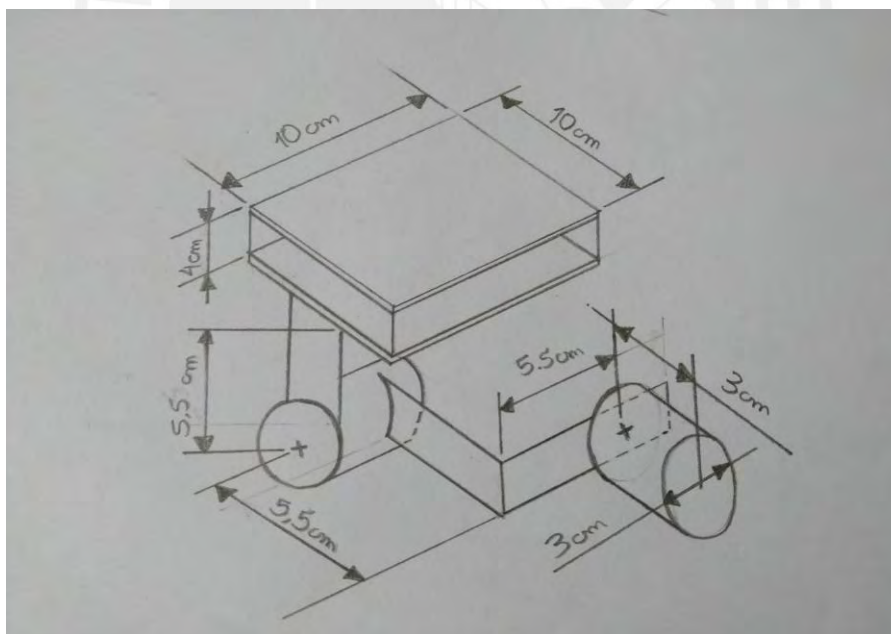


Figura 2.1 Dimensiones generales aproximadas del módulo mecatrónico

Fuente: elaboración propia

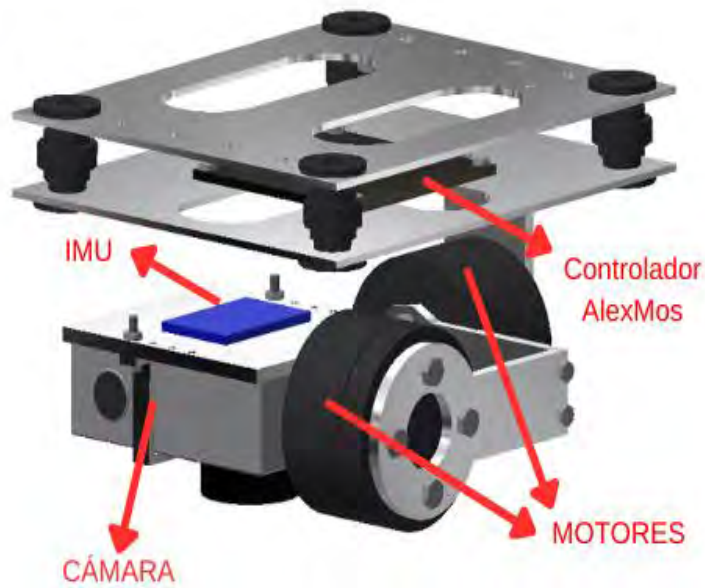


Figura 2.2 Modelado del módulo mecatrónico

Fuente: elaboración propia

2.3 Presentación de componentes y materiales

Se presenta los materiales y elementos mecánicos, así como los análisis correspondientes.

2.3.1 Estructura del estabilizador

La estructura del estabilizador para la cámara será hecha de aluminio, tendrá un peso aproximado de 320 g, todas las piezas están unidas por tornillos y será como se muestra en la figura 2.3.

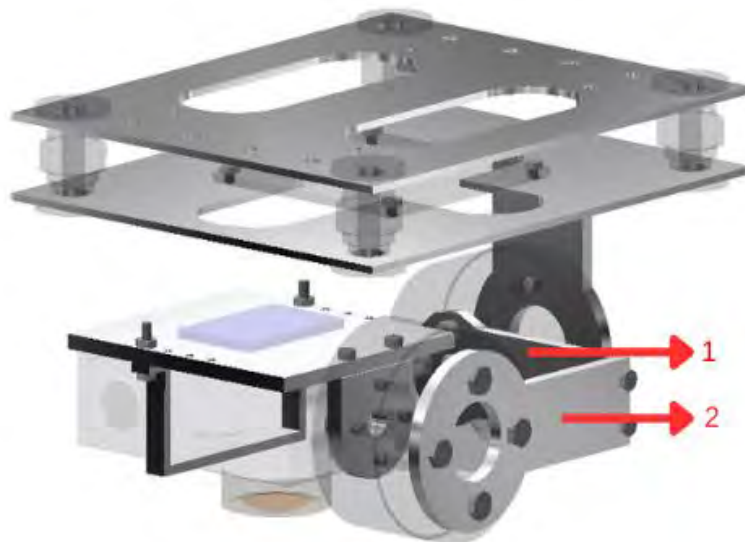


Figura 2.3 Estructura de aluminio

Fuente: elaboración propia

Las piezas 1 y 2 señaladas en la anterior figura son móviles y están sujetas a un torque de un motor y el peso de otros componentes, por lo que deben ser analizadas a flexión y a corte. El diagrama de cuerpo libre de la pieza 1 se muestra en la figura 2.4.

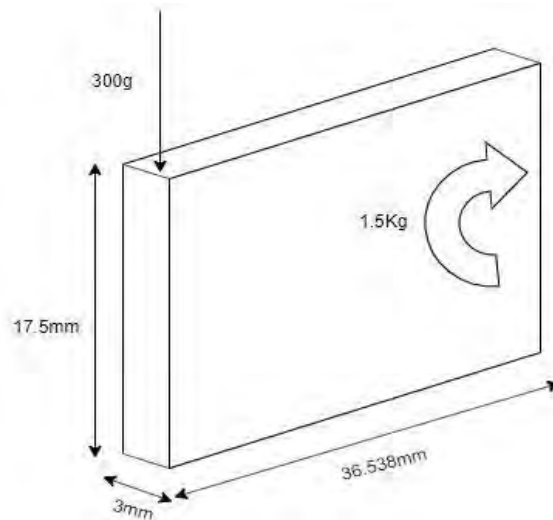


Figura 2.4 Diagrama de cuerpo libre de la pieza 1

Fuente: elaboración propia

Un extremo está sujetado al motor y el otro al resto de piezas, este último extremo soporta el peso de otras 4 piezas de aluminio, un motor y la cámara que aproximadamente suman un peso de 300 gramos y el motor ejerce un torque de 1.5Kg.

El esfuerzo de flexión a la que se encuentra sometida la pieza se calcula con la siguiente ecuación:

$$\sigma = \frac{T x l x \frac{h}{2}}{b x h^3 x \frac{1}{12}} \dots (2.1)$$

Donde:

Variable	Valor
T (torque)	17.64 Nmm
l (largo)	36.54 mm
h (altura)	17.5 mm
b (ancho)	3 mm
Resultado (σ)	4.21 MPa

El aluminio tiene un esfuerzo de fluencia de 55 MPa, entonces no fallará.

El esfuerzo de corte al que está sometida la pieza se calcula con la siguiente ecuación:

$$\tau = \frac{V \times Q}{I \times t} \dots (2.2)$$

Donde:

Variable	Valor
V (fuerza)	2.97 N
Q	114.84 mm ³
I (inerencia)	1339.84 mm ⁴
t	3 mm
Resultado (τ)	0.084 MPa

El aluminio tiene un esfuerzo máximo de corte de 26.3 MPa, entonces no fallará.

Estos cálculos se comprueban mediante la realización de una simulación de esfuerzos con ayuda del software INVENTOR, en la figura 2.5 se muestra lo resultados de esfuerzos tipo Von Mises, donde se puede apreciar que el esfuerzo máximo presente en la pieza es de 2.627 MPa.

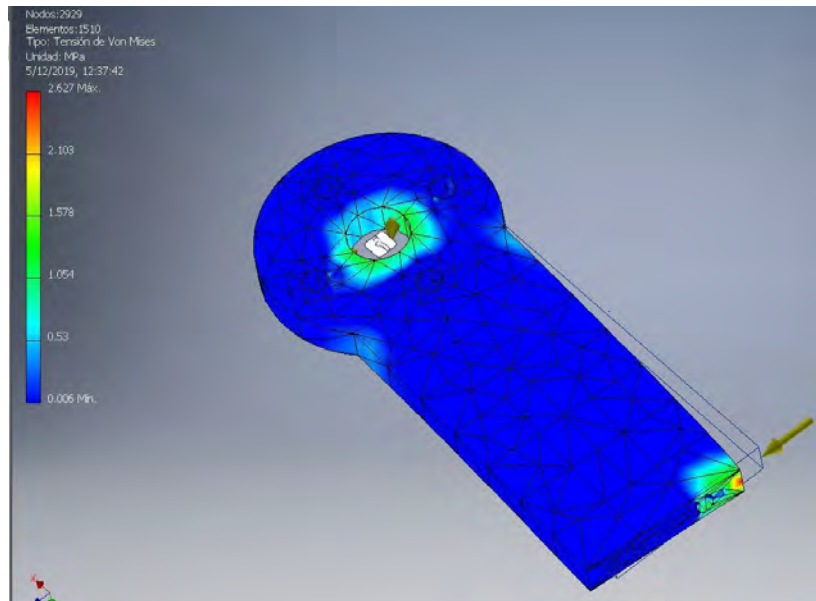


Figura 2.5 Tensión de Von Mises en la pieza 1

Fuente: elaboración propia

En la figura 2.6 y figura 2.7 se muestran los resultados de deformaciones y desplazamientos, respectivamente, para la misma simulación. Donde se puede apreciar que la deformación máxima que podría sufrir la pieza es de 10^{-5} ul y un desplazamiento máximo de 0.0016 mm.

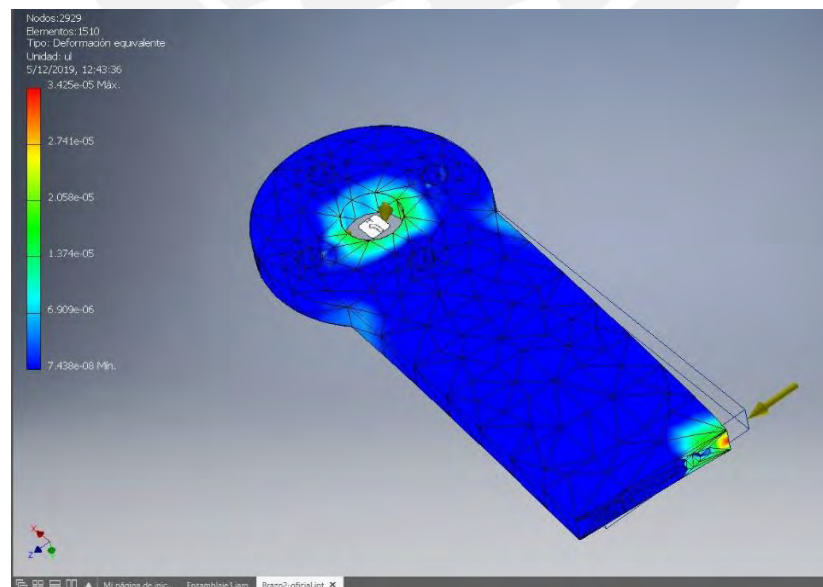


Figura 2.6 Deformaciones equivalente en la pieza 1

Fuente: elaboración propia

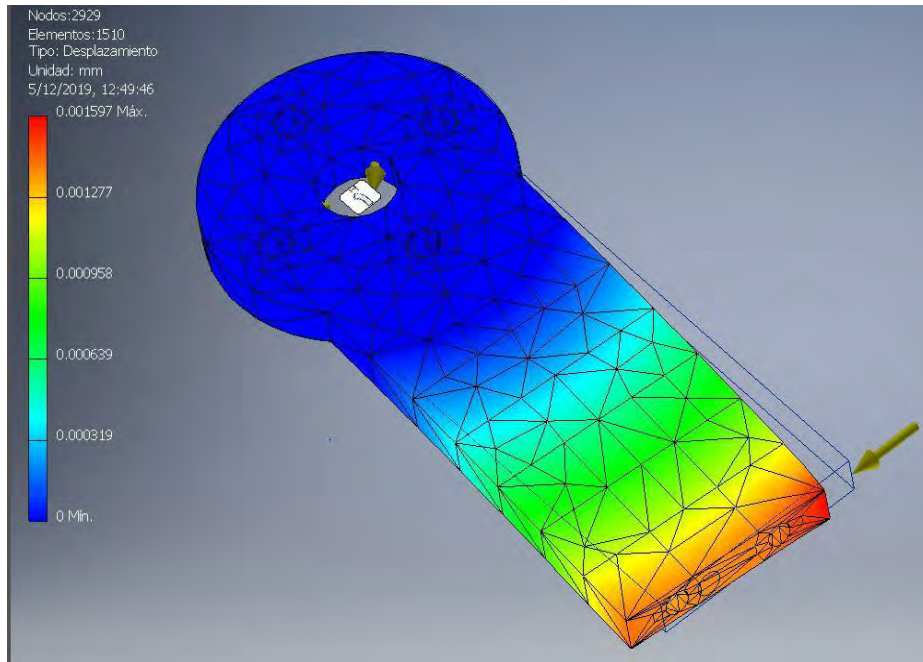


Figura 2.7 Desplazamientos en la pieza 1

Fuente: Elaboracion propia

Finalmente, en la figura 2.8 se muestra el factor de seguridad igual a 15 que presenta la pieza, para el material de Aluminio 6061.

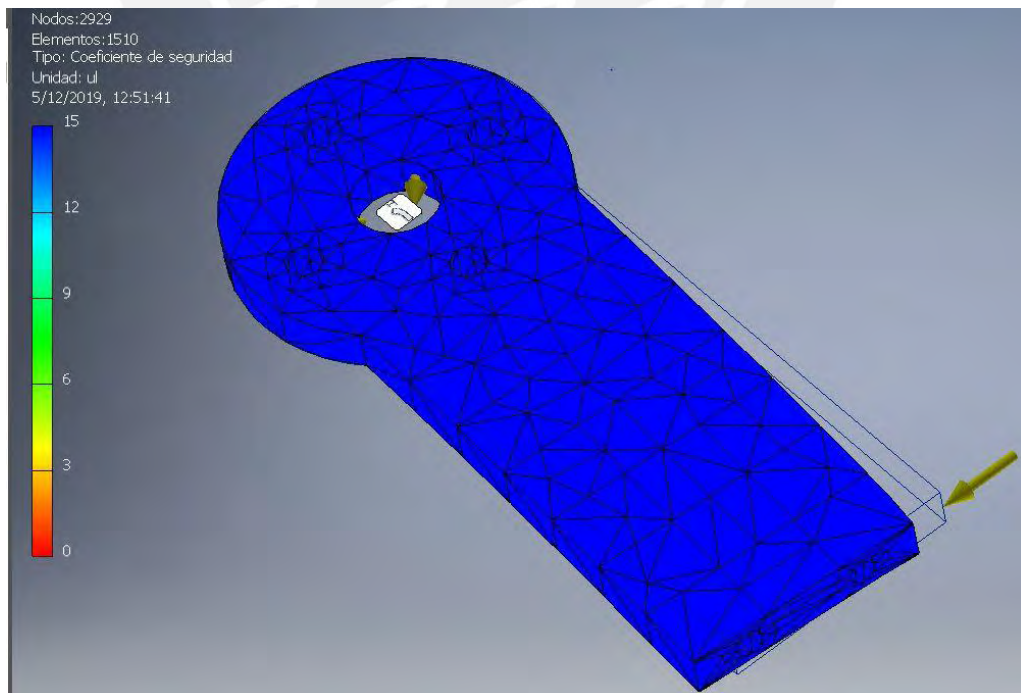


Figura 2.8 Factor de seguridad de la pieza 1

Fuente: elaboracion propia

El diagrama de cuerpo libre de la pieza 2 se muestra en la figura 2.9.

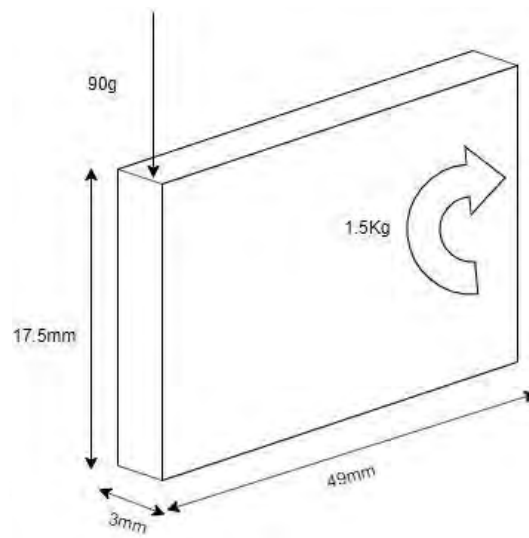


Figura 2.9 Diagrama de cuerpo libre de la pieza 2

Fuente: elaboración propia

Un extremo está sujeto al motor y el otro al resto de piezas, este último extremo soporta el peso de otras piezas de aluminio y la cámara que aproximadamente suman un peso de 90 gramos y el motor ejerce un torque de 1.5Kg.

El esfuerzo de flexión al que se encuentra sometida la pieza se calcula con la ecuación 2.1 mencionada anteriormente.

Reemplazando los siguientes valores:

Variable	Valor
T (torque)	15.58 Nmm
l (largo)	49 mm
h (altura)	17.5 mm
b (ancho)	3 mm
Resultado (σ)	4.99 MPa

El aluminio tiene un esfuerzo de fluencia de 55 MPa, entonces no fallará.

El esfuerzo de corte al que se encuentra sometida la pieza se calcula con la ecuación 2.2 mencionada anteriormente.

Reemplazando los siguientes valores:

Variable	Valor
V (fuerza)	0.882 N
Q	114.84 mm ³
I (ineria)	1339.84 mm ⁴
t	3 mm
Resultado (τ)	0.025 MPa

El aluminio tiene un esfuerzo máximo de corte de 26.3 MPa, entonces no fallará.

De la misma manera que para la pieza 1, se realiza una simulación de esfuerzos para comprobar que la pieza no falle. A continuación, en la figura 2.10 se muestra el resultado obtenido de la simulación y se observa que presenta un esfuerzo máximo de 0.8 Mpa.

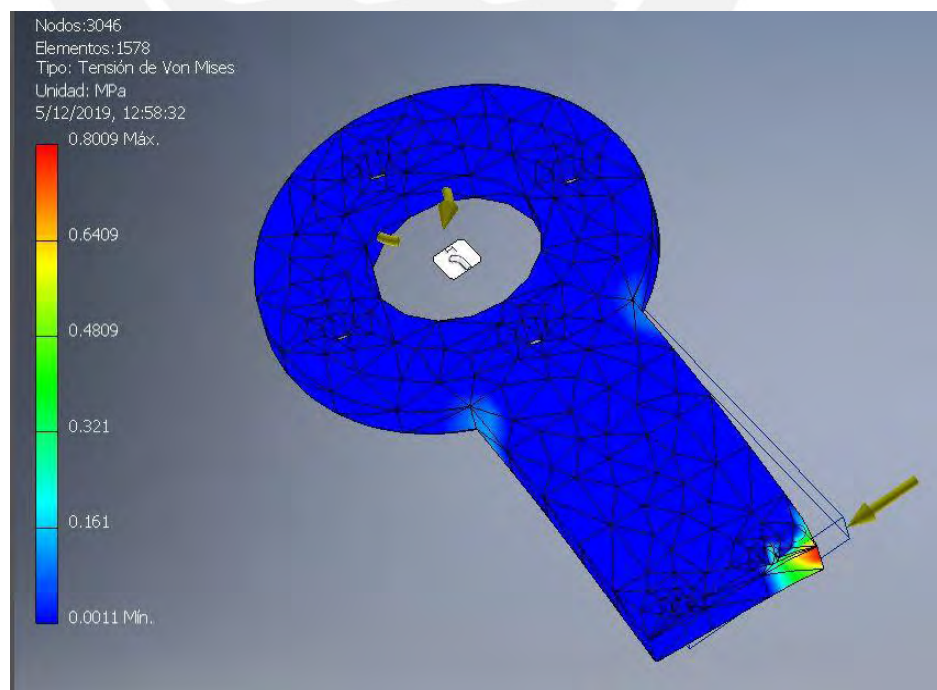


Figura 2.10 Tensiones de Von Mises en la pieza 2

Fuente: elaboración propia

En la figura 2.11 y figura 2.12 se muestran los resultados de deformaciones y desplazamientos, respectivamente, para la simulación realizada con ayuda del software INVENTOR. Donde se puede apreciar que la deformación máxima que podría sufrir la pieza es de $1.04 \times 10^{-5} ul$. Y sufre un desplazamiento máximo de $2.77 \times 10^{-4} mm$.

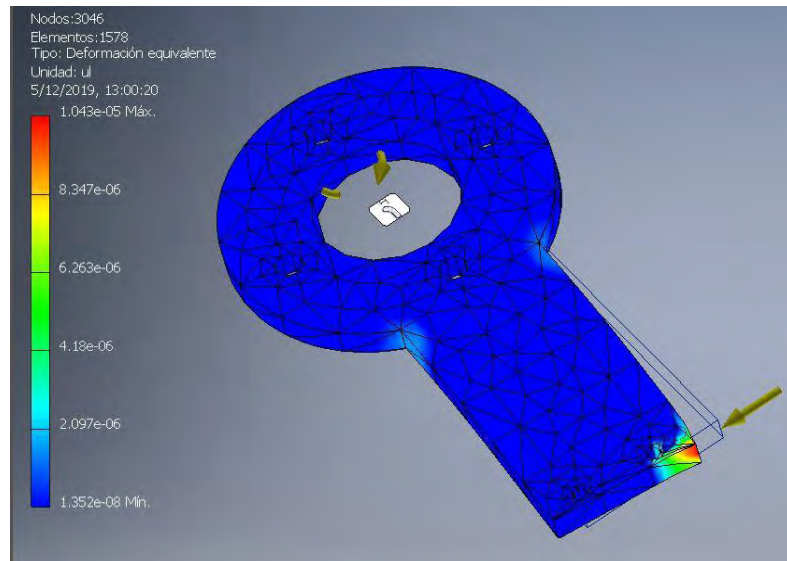


Figura 2.11 Deformaciones en la pieza 2

Fuente: elaboración propia

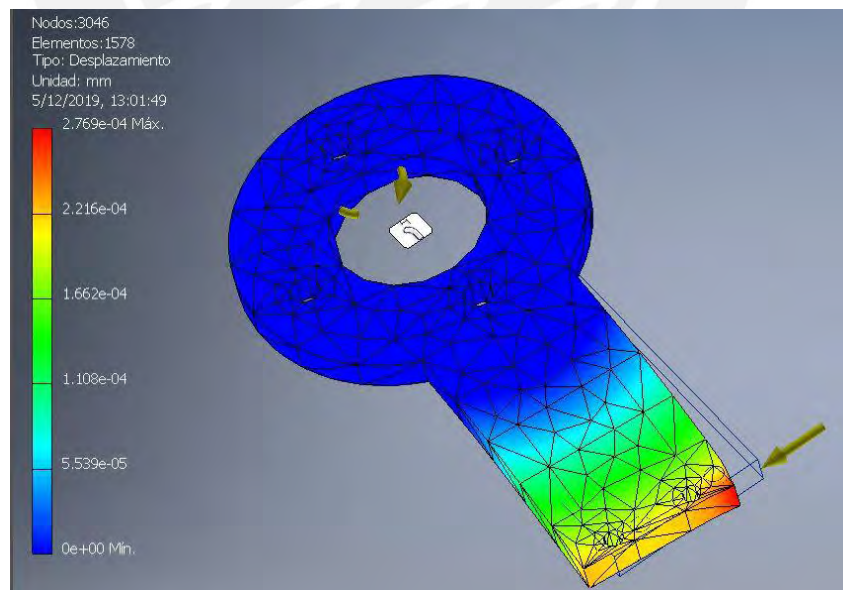


Figura 2.12 Desplazamientos en la pieza 2

Fuente: elaboración propia

Finalmente, en la figura 2.13 se muestra el factor de seguridad igual a 15 que presenta la pieza, también para el material de Aluminio 6061.

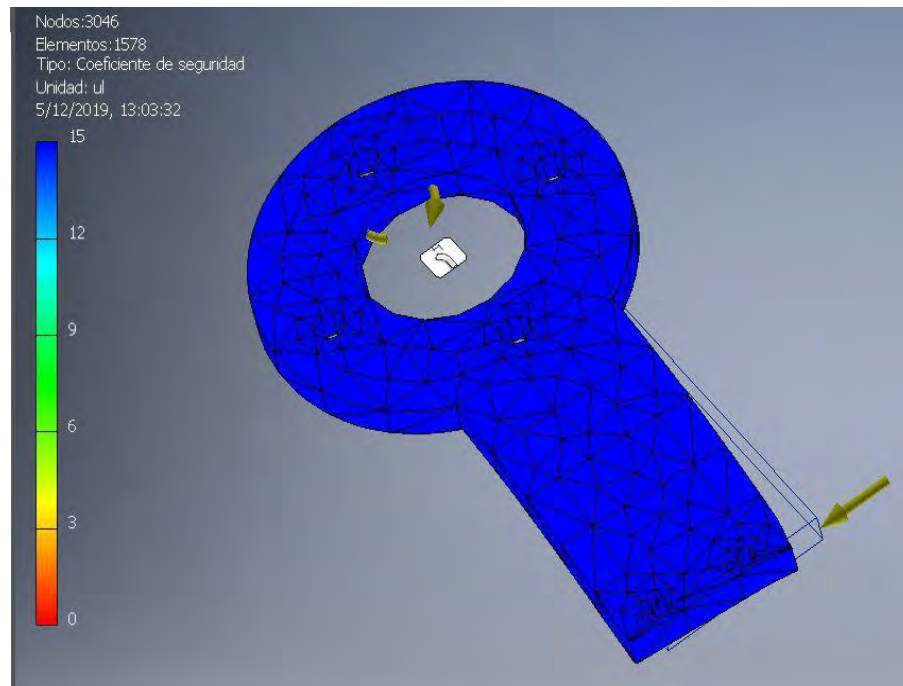


Figura 2.13 Factor de seguridad de la pieza 2

Fuente: elaboración propia

Con los análisis realizados, se muestra que los esfuerzos a los que encuentran sometidos ambas piezas están muy por debajo de los valores de falla, además de presentar un factor de seguridad mayor o igual a 15. Con esto se comprueba que las piezas no fallarán.

Además, correspondiente al diseño mecánico del presente módulo mecatrónico se realizan los planos correspondientes. Se presenta un plano de fabricación para cada una de las 8 piezas que conforman el presente; adicionalmente se realiza un plano de ensamblaje, haciendo así un total de 9 planos. Esta lista se presenta a continuación en la tabla 2.1 y los planos se anexan a este documento.

Tabla 2.1 Lista de planos mecánicos

Fuente: elaboración propia

N°	Detalle	Tipo
1	Soporte de Alexmos	fabricación
2	Soporte de unión al dron	fabricación
3	Sujetador de cámara	fabricación
4	Soporte de cámara	fabricación
5	Brazo N°1	fabricación
6	Brazo N°2	fabricación
7	Brazo N°3	fabricación
8	Brazo N°4	fabricación
9	Estabilizador de cámara	ensamblaje

Todas las partes son unidas por tornillos allen según la norma DIN 912 (OPAC), que se adjunta en el anexo B.

2.3.2 Amortiguadores de vibración

Las vibraciones que el dron genera pueden causar un mal funcionamiento en la cámara y en el controlador del estabilizador (AlexMos), por lo que se utilizarán amortiguadores de caucho para reducir estas perturbaciones, los cuales se muestran en la figura 2.14 con sus medidas respectivas.



Figura 2.14 Amortiguadores de caucho

Fuente: https://spanish.alibaba.com/p-detail/de-silicona-de-caucho-amortiguador-de-vibraciones-bolas-3_00004238542.html?spm=a2700.8699010.normalList.28.653d552bT1adtg

CAPÍTULO III

DISEÑO ELECTRÓNICO

En este capítulo se trata el diseño electrónico del sistema; se presenta la arquitectura del hardware, los sensores y actuadores seleccionados, y las características y análisis correspondientes.

3.1 Requerimientos electrónicos

- El módulo mecatrónico deberá contar con sensores que permitan la estabilización de la cámara.
- El módulo deberá energizarse utilizando la batería del dron y deberá tener un consumo de corriente menor a 5A.
- El módulo deberá contar con máximo 2 actuadores que permitan mantener la cámara horizontalmente estable a pesar del movimiento del dron.
- El módulo deberá contar con una cámara para realizar la adquisición de imágenes hasta una altura de 12 metros.
- El módulo deberá contar con un microcontrolador que realice la estabilización de la cámara, el procesamiento de imágenes y el control del dron durante el aterrizaje.

3.2 Presentación de sensores y actuadores

En esta sección se presentan los sensores y actuadores seleccionados, así como las características y análisis correspondientes.

3.2.1 Motores

Los motores seleccionados son de tipo “brushless” y deben ser capaces de mover una masa máxima de aproximadamente 300 gramos. Estos motores son diseñados especialmente para estabilizadores de cámara.

En la figura 3.1 se muestra los motores colocados en el módulo mecatrónico y se les identifica como MOTOR1 y MOTOR2 para facilitar su identificación al desarrollar los cálculos respecto a su funcionamiento.

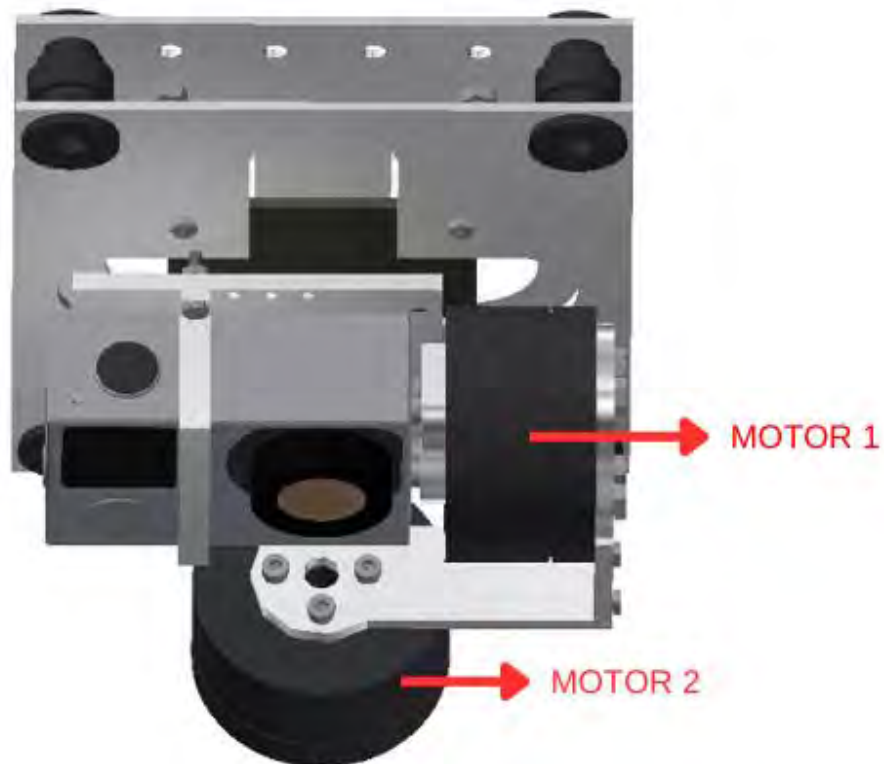


Figura 3.1 Motores en el módulo mecatrónico

Fuente: elaboración propia

El MOTOR1 es el encargado de mover sólo la cámara, cuyo centro de masa está alineado con el eje del motor.

El momento de inercia de la cámara en el eje del motor se calcula con la siguiente ecuación:

$$I = \frac{(h^2+b^2)w}{12} \dots (3.1)$$

Donde:

Variable	Valor
h (altura de la cámara)	42 mm
b (ancho de la cámara)	21.2 mm
w (masa de la cámara)	72 g

Entonces, reemplazando los valores en la ecuación (3.1) se obtiene para el MOTOR 1:

$$I = 1.328 \times 10^{-5} \text{ Kg. } m^2$$

Utilizando la ecuación (3.1) se calcula que la inercia que el MOTOR 2 debe ser capaz de mover es:

$$I = 2.789 \times 10^{-5} \text{ Kg. } m^2$$

Se espera que el sistema tenga un tiempo de respuesta máximo de 10 ms.

Del catálogo de T-MOTOR se elige el modelo “GB36-1”, el cual se muestra en la figura 3.2.



Figura 3.2 Motor T-MOTOR GB36-1

Fuente: <http://store-en.tmotor.com/goods.php?id=441>

Se muestra a continuación en la tabla 3.1 que las características del motor (T-MOTOR, 2019) son adecuadas.

Tabla 3.1 Especificaciones del motor

Fuente: elaboración propia

Dimensiones	41.8 (diámetro) x 24.3 mm
Torque	1.5Kg (con batería LiPo 4 celdas)
Velocidad de giro	600 RPM
Peso	88 gramos
bluetooth	4.0
Voltaje	11.1V - 22.2V

A continuación, en la figura 3.3 se muestra las dimensiones más importantes de estos motores.

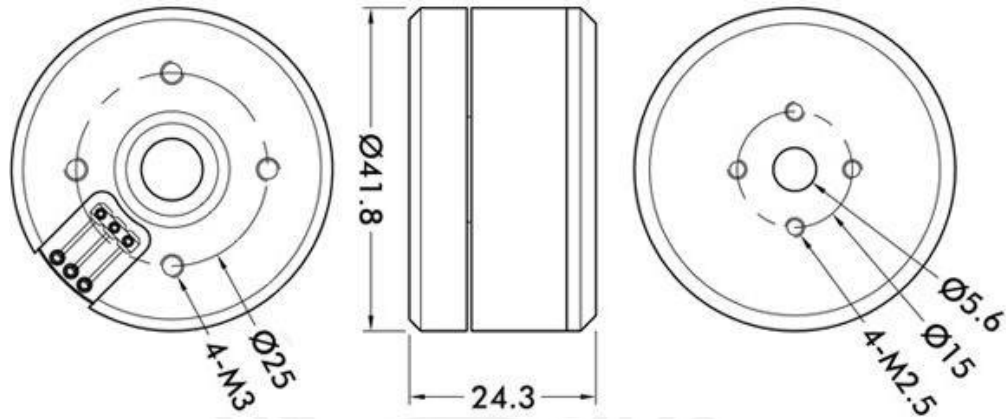


Figura 3.3 Dimensiones del motor

Fuente: <http://store-en.tmotor.com/goods.php?id=441>

Con los datos del catálogo para el modelo seleccionado, la aceleración angular del motor se calcula con la siguiente ecuación:

$$T = I \times \alpha \dots (3.2)$$

Donde, para el MOTOR 1:

T (Torque según catálogo) = 1.5 Kg

α = Aceleración angular

Entonces: $\alpha = 1.13 \times 10^5 \text{ rad/s}^2$

Con esta aceleración angular, el MOTOR 1 llega a su velocidad máxima de 600 RPM en 0.5 ms.

Utilizando la ecuación (3.2), se calcula la aceleración angular para el MOTOR 2:

Y se obtiene: $\alpha = 5.37 \times 10^4 \text{ rad/s}^2$

Con esta aceleración angular, el MOTOR 2 llega a su velocidad máxima de 600 RPM en 1 ms.

Se comprueba que los motores son capaces de tener un tiempo de respuesta que no supera el máximo esperado.

El MOTOR 1 y el MOTOR 2 son atornillados a los BRAZOS 1 y 2 respectivamente, esta unión atornillada se encuentra detallada en el plano de ensamblaje presentado en el anexo H, sin embargo también se muestra la integración en la siguiente Figura 3.4.

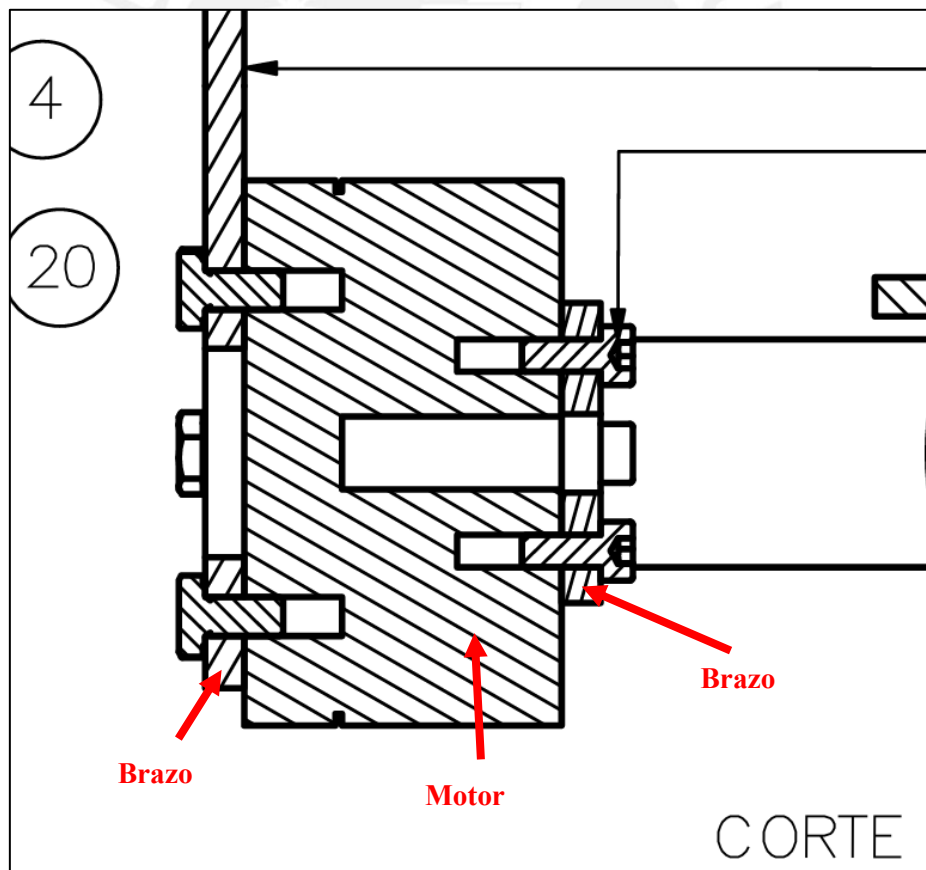


Figura 3.4 Unión atornillada de Brazo y Motor

Fuente: Elaboración propia

3.2.2 Controlador del estabilizador de la cámara

El controlador seleccionado es el controlador “DRV8313 Driver” (AlexMos), controlador diseñado especialmente para estabilizadores de hasta 3 ejes que trabajan con motores tipo brushless y posee su propio código, por lo que no será elaborado en el presente desarrollo, solo es necesario realizar la calibración ya con el módulo armado para obtener los parámetros que garanticen un correcto funcionamiento. En la figura 3.5 se muestra el controlador, seguido de sus especificaciones (BaseCam, 2019) presentadas en la tabla 3.2.



Figura 3.5 Controlador Alexmos DV8313 Driver

Fuente: <https://www.basecamelectronics.com/simplebgc/>

Tabla 3.2 Especificaciones del controlador AlexMos

Fuente: elaboración propia

Dimensiones	50 mm x 50 mm
Peso	10 gramos
Fuente de alimentación	9-18V (3 - 4S)
Corriente de accionamiento	Max. 1.5A
Protocolo de comunicación	I2C

Este controlador trabaja con un firmware que, conectándolo a una computadora a través de un puerto USB, permite variar los parámetros de trabajo, como las ganancias del controlador PID para obtener un error igual a cero (0) en estado estable, tiempo de establecimiento y sobre impulsos deseados. El lazo de control se muestra en la figura 3.6 donde se ve el controlador PID y la realimentación del sistema a través del sensor IMU que se coloca junto a la cámara, esto permite mantener la cámara de manera horizontal durante el aterrizaje del dron.

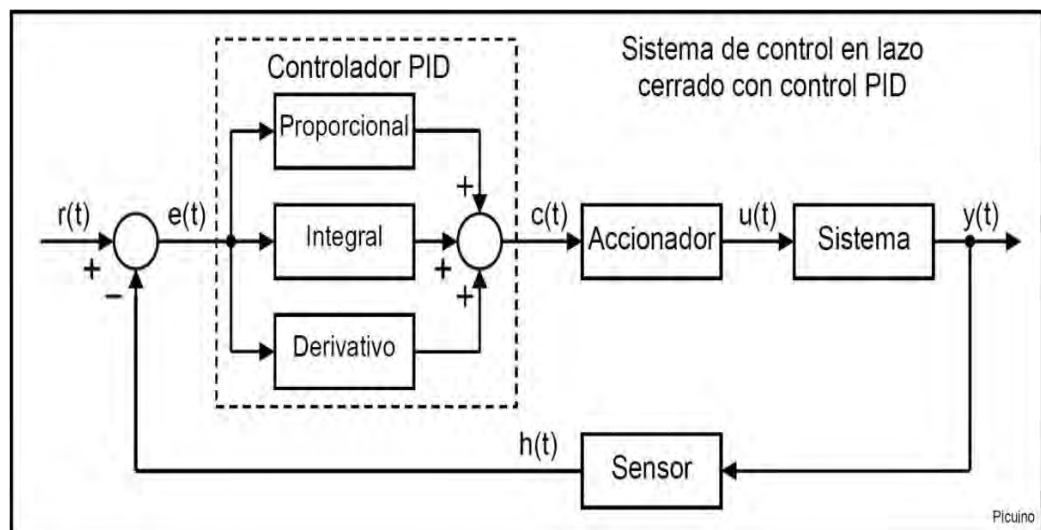


Figura 3.6 Lazo de control del estabilizador de la cámara

Fuente: <https://www.picuino.com/es/arduprog/control-pid.html>

Los parámetros del controlador PID de los motores se pueden hallar utilizando la herramienta Simulink del software MATLAB en el que se genera el lazo de control utilizando 4 bloques principales, El bloque “STEP” que es la referencia que recibe el sistema, el bloque PID que es el controlador, el bloque “TRANSFER FUNCTION” que simula el motor y finalmente el bloque “SCOPE” que permite ver la salida del sistema. La simulación de la respuesta de un motor del estabilizador se muestra a continuación en la figura 3.7.

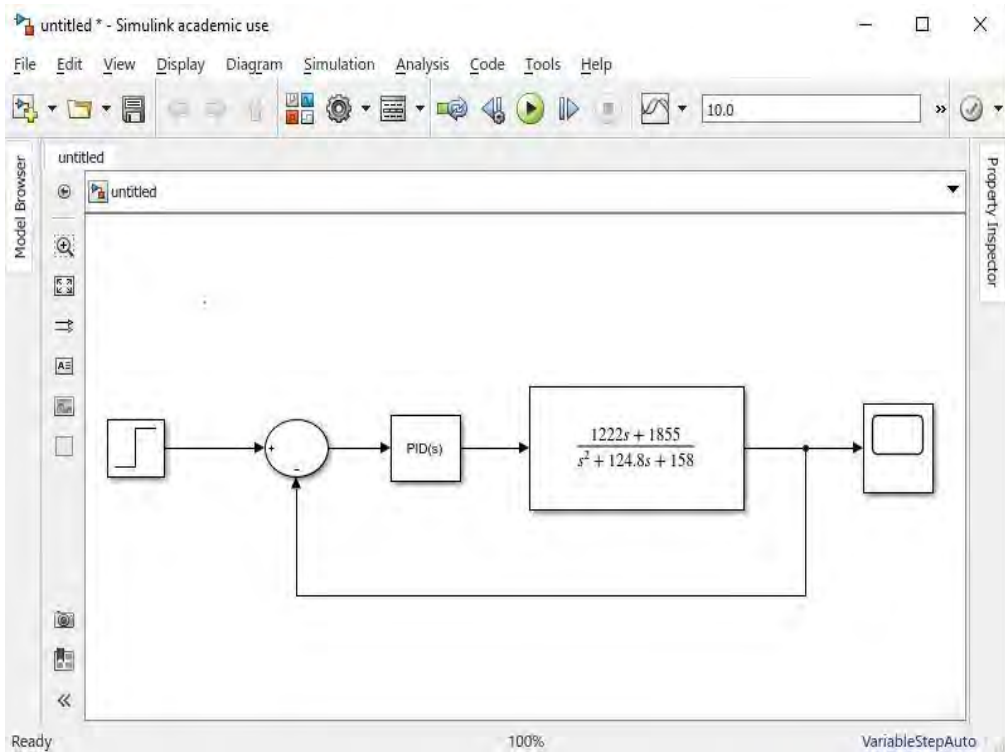


Figura 3.7 Lazo de control de un motor del estabilizador de la cámara Fuente: elaboración propia

El bloque PID de Simulink permite tuncar el controlador según el tiempo de respuesta deseado, anteriormente se mostró que este tiempo debe ser de 0.5 ms. A continuación, en la figura 3.8 se muestra que el sistema respeta el tiempo de establecimiento planteado.

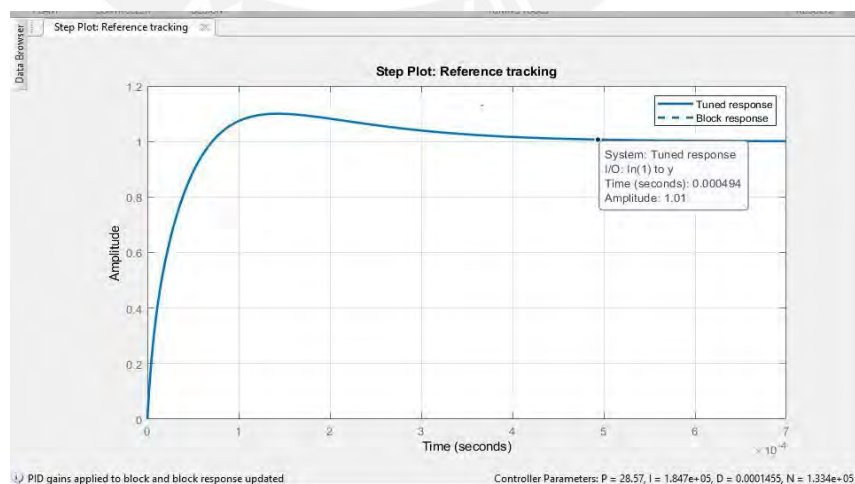


Figura 3.8 Respuesta en el tiempo del motor utilizando un controlador PID

Fuente: elaboración propia

Y los parámetros del controlador PID se muestran en la figura 3.9.

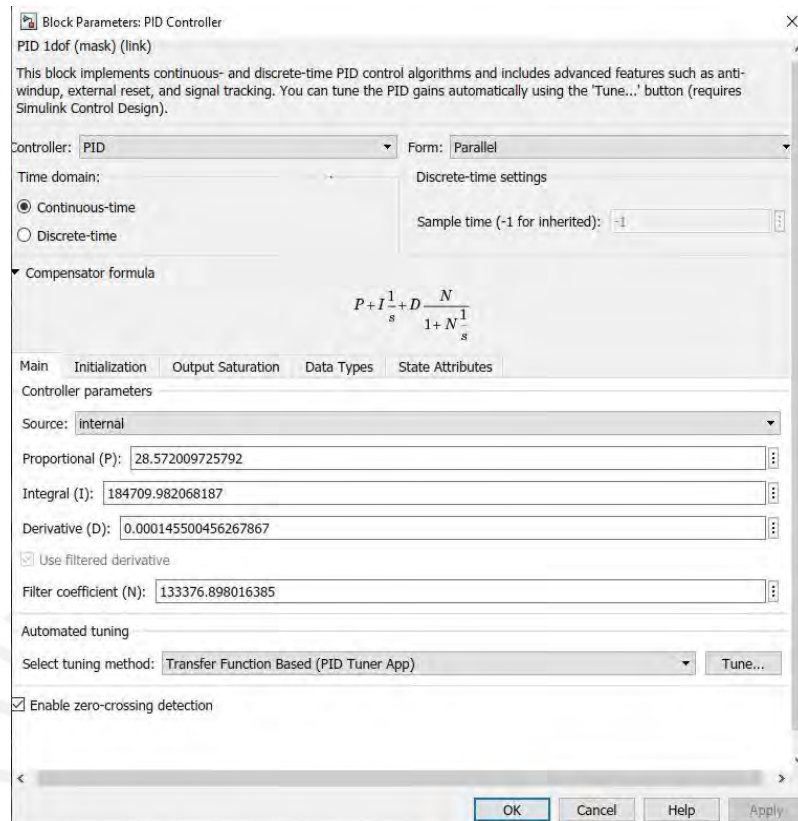


Figura 3.9 Parámetros del controlador PID

Fuente: elaboración propia

Como se observa en la figura 3.8 el sistema presenta un pequeño sobre impulso, esto se puede solucionar empleando el firmware presentado en la figura 3.10, donde se puede limitar el voltaje máximo que recibe cada motor. La siguiente imagen muestra como ejemplo una vista del firmware, mas no los parámetros reales para este desarrollo.

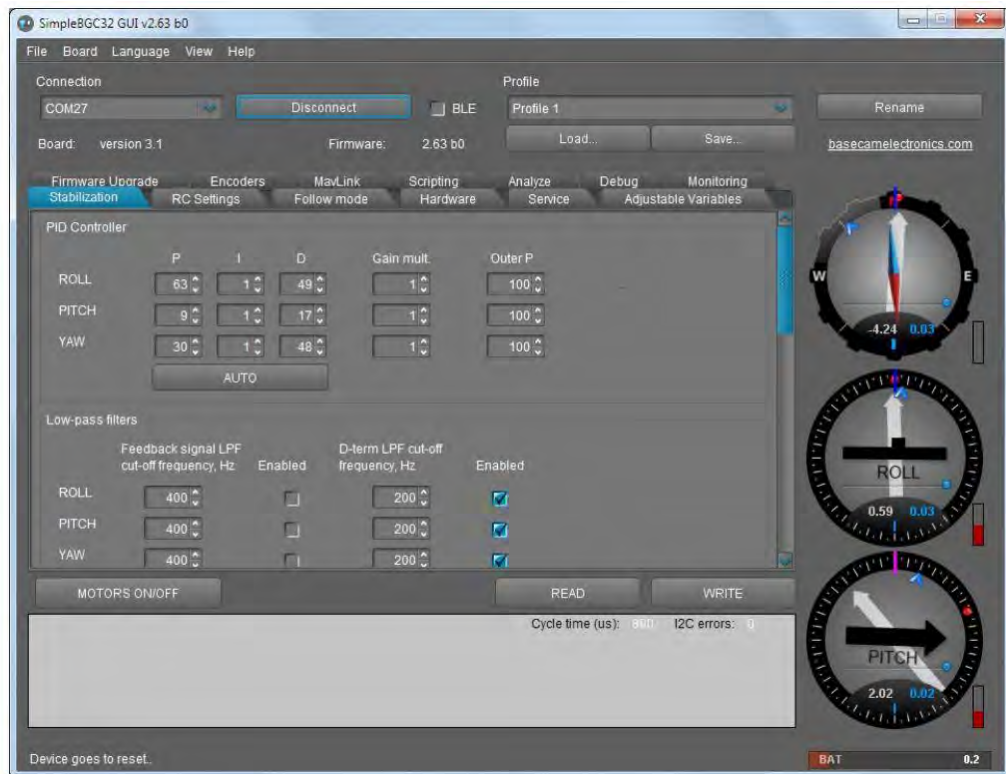


Figura 3.10 Interfaz de computadora del firmware

Fuente: <https://aerialpixels.com/>

Existe otra manera de tunear el controlador PID y es a través del firmware, en la ventana PID CONTROLLER se encuentra el botón AUTO, este botón permite calcular de manera automática las ganancias PID del controlador para cada motor (MDI, 2014). Se adjunta en el anexo C el manual de usuario del API de BaseCam en el que se explica con más detalle el proceso de tuneado del estabilizador de la cámara. Este API también permite calibrar el sensor IMU, los pasos más detallados para lograr esto también se encuentran en la guía de usuario de BaseCam en el mismo anexo.

El controlador se conecta a la batería LiPo del dron, como se muestra en la figura 3.11.

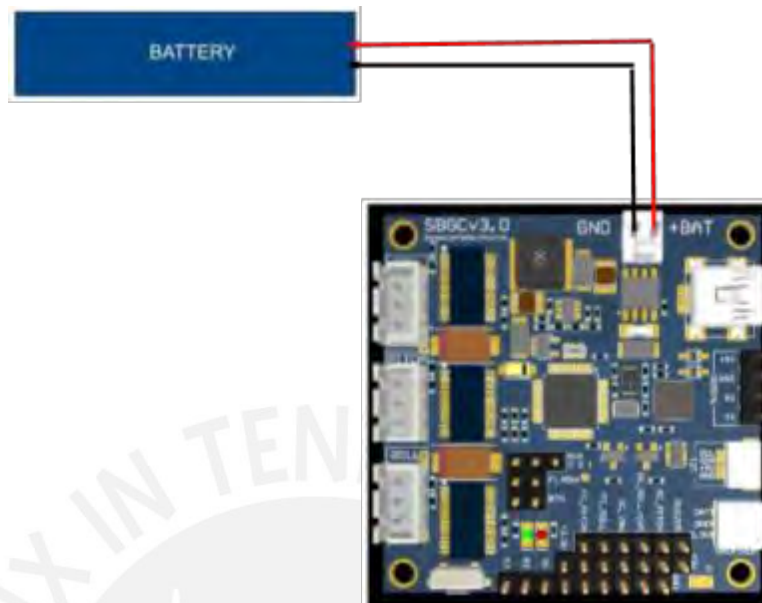


Figura 3.11 Conexión del controlador con la batería

Fuente: <https://aerialpixels.com/>

El controlador a su vez se conecta a los motores, en la figura 3.12 se muestra la conexión entre estos.

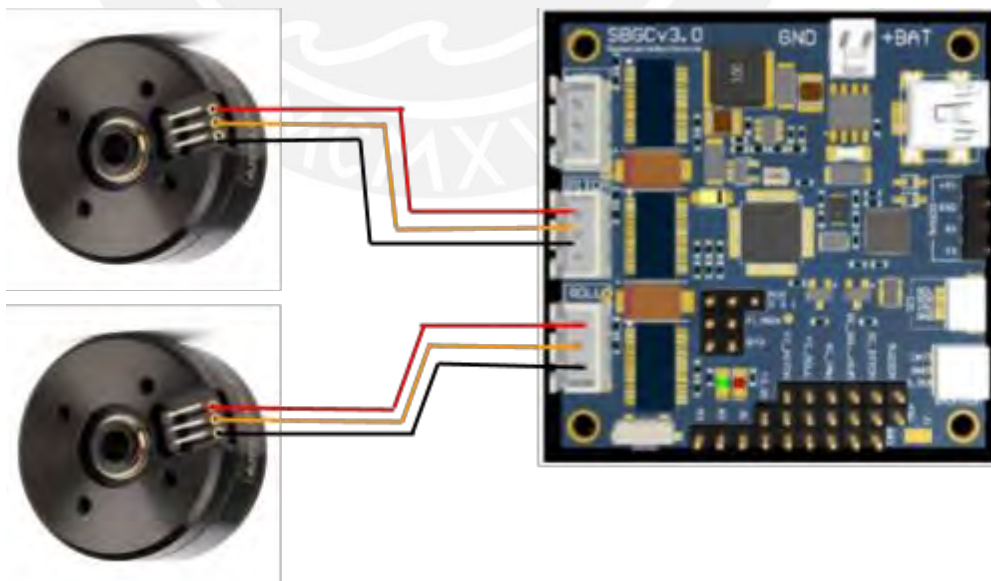


Figura 3.12 Conexión del controlador con los motores

Fuente: elaboración propia

La unión atornillada entre el controlador y la pieza mecánica correspondiente se muestra a detalle en el plano de ensamblaje presentado en el anexo H, también se muestra a continuación en la figura 3.13.

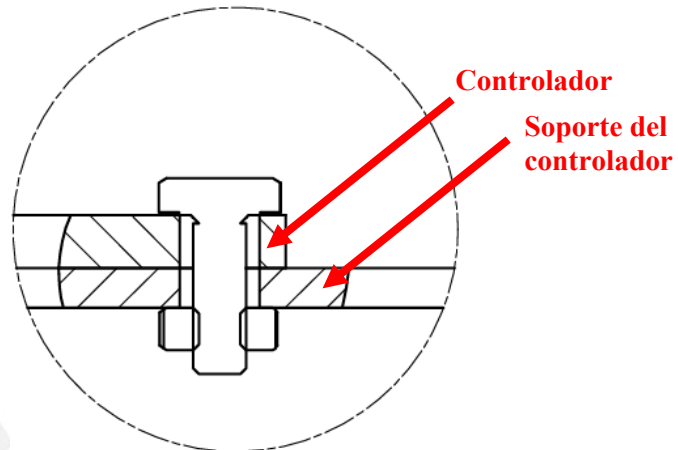


Figura 3.13 Unión atornillada entre el controlador y el soporte mecánico del controlador

Fuente: elaboración propia

3.2.3 Sensor de orientación

El sensor MPU6050 es un integrado que ofrece un giroscopio y un acelerómetro, ambos de 3 ejes. Este sensor es compatible con el controlador DRV8313 pues funciona con el protocolo de comunicación I2C. Este sensor y sus 6 ejes se muestran en la figura 3.14.

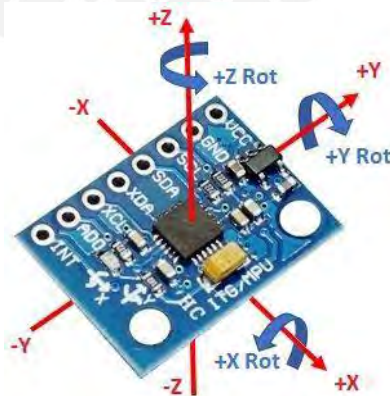


Figura 3.14 Sensor MPU6050

Fuente: <https://naylorlampmechatronics.com/sensores-posicion-inerciales-gps/33-modulo-mpu6050-acelerometro-giroscopio-i2c.html>

El sensor se conecta al controlador Alexmos a través del puerto I2C, como se muestra en la figura 3.15.

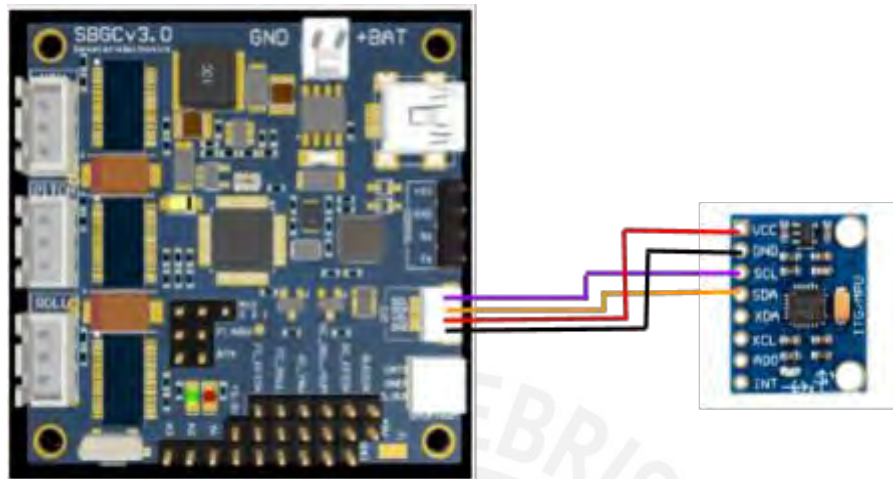


Figura 3.15 Conexión del controlador con sensor IMU

Fuente: elaboración propia

El sensor se sujeta al soporte de la cámara con cinta de doble contacto en la posición que se muestra a continuación en la figura 3.16.



Figura 3.16 integración IMU y soporte de cámara

Fuente: elaboración propia

3.2.4 Controlador de vuelo

El controlador de vuelo elegido es un PIXHAWK, pues al ser uno de los más comunes en la industria, existe un gran soporte para estos y trabaja con un código abierto de ArduPilot. En la figura 3.17 se muestra el Pixhawk.



Figura 3.17 Controlador de vuelo PIXHAWK

Fuente: <http://pixhawk.org/>

3.2.5 Cámara y controlador principal

Se utiliza la OpenMV CamVRT1062. Este es un integrado que posee una tarjeta y un lente. Está diseñado especialmente para realizar visión por computadora y la tarjeta funciona con MicroPython, además utiliza un IDE que se ejecuta en una computadora para editar y subir el código (OPENMV, 2019). Se puede conectar directamente al controlador de vuelo a través del puerto de telemetría (ARDUPILOT, 2024), lo que se muestra en la figura 3.18.

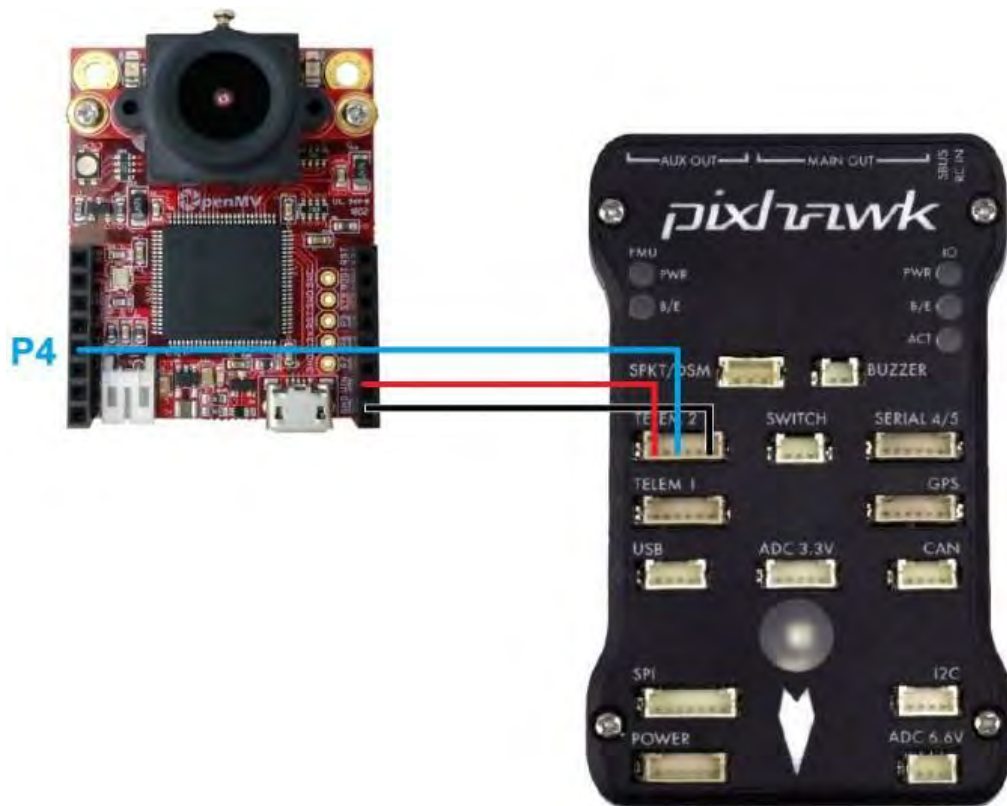


Figura 3.18 OpenVM Cam RT1062

Fuente: <https://openmv.io/collections/cams/products/openmv-cam-h7>

En la tabla 3.3 se presentan algunas de las especificaciones más importantes de la OpenVM Cam RT1062.

Tabla 3.3 Especificaciones de OpenVM Cam RT1062

Fuente: elaboración propia

Resolución máxima	2592x1944 (5MP)
Alimentación	3.3V 170ma
Dimensiones	4.5 cm x 2.9 cm x 3.6 cm
Peso	20 gramos

Este integrado al funcionar con MicroPython, no solo puede realizar procesamiento de imágenes, sino que con la librería MavRos y DroneKit puede comunicarse con el controlador de vuelo para realizar los movimientos del dron.

La cámara se sujeta al ‘Soporte de cámara’ con una segunda pieza denominada ‘Sujetador de cámara’, a través de uniones atornilladas. Esta unión se muestra a detalle en el plano de ensamblaje presentado en el anexo H, también se muestra a continuación en la Figura 3.19.

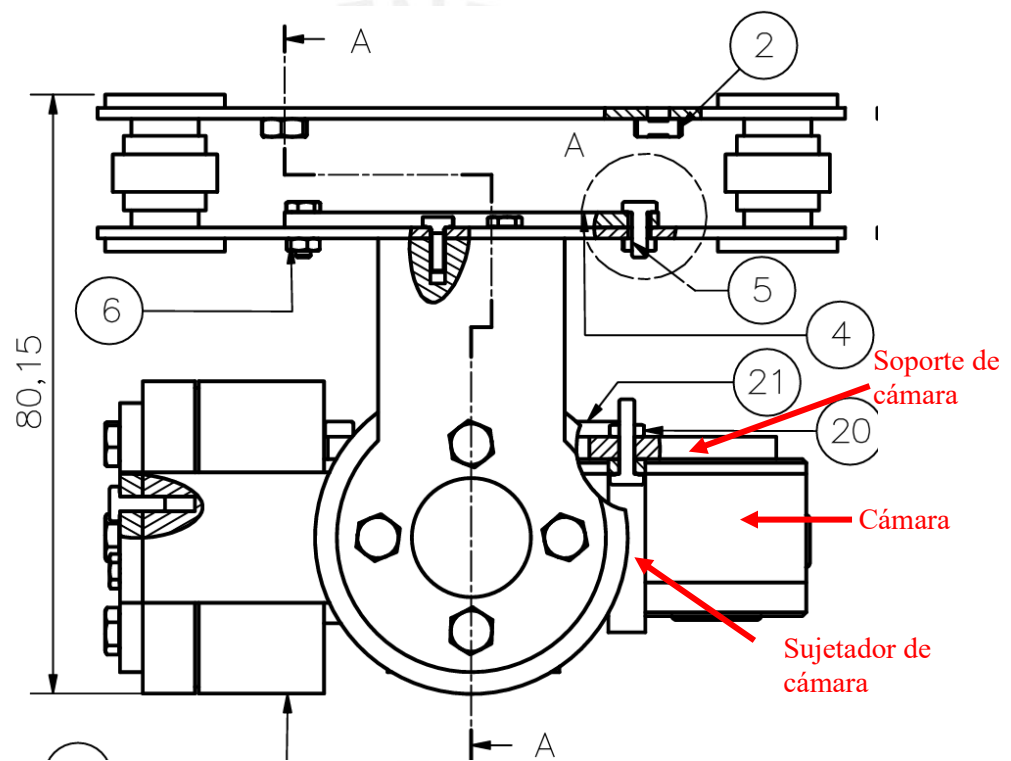


Figura 3.19 Integración de cámara a la estructura mecánica

Fuente: Elaboración propia

3.2.6 Arquitectura del hardware

El propósito de este proyecto es lograr que un dron multirrotor sea capaz de aterrizar de manera autónoma y precisa sobre una plataforma que se encuentra en movimiento. Entonces para explicar el hardware a utilizar, las conexiones y el funcionamiento de estos, como se muestra en la figura 3.20, se asume que el dron ya está volando.

Primero, el controlador principal (OpenVM CamRT1062) espera a recibir la señal del controlador de vuelo que le indique que el dron ya llegó al punto final de la misión (mediante la lectura del GPS) para activar el sistema de aterrizaje autónomo y preciso. Una vez activado el sistema de aterrizaje el controlador principal (OpenVM CamRT1062) recibe información del IMU del estabilizador de la cámara y activa los motores de este para compensar los movimientos del dron y así obtener imágenes sin ruido y estables, esto lo realiza durante todo el proceso de aterrizaje.

Segundo, el sistema intentará localizar la plataforma de aterrizaje mediante el procesamiento de las imágenes capturadas por la cámara (OpenVM CamRT1062), y una vez localizada el controlador principal establecerá conexión con el controlador de vuelo (PIXHAWK).

Tercero, el controlador principal (OpenVM CamRT1062) envía señales al controlador de vuelo (PIXHAWK), mediante el uso de la librería MAVlink, disponible para Python; para mantener al dron en una posición colineal con el centro de la plataforma de aterrizaje y realizar el descenso.

Cuarto, y último paso, una vez haya culminado el aterrizaje el controlador principal procederá a guardar la data obtenida durante el aterrizaje.

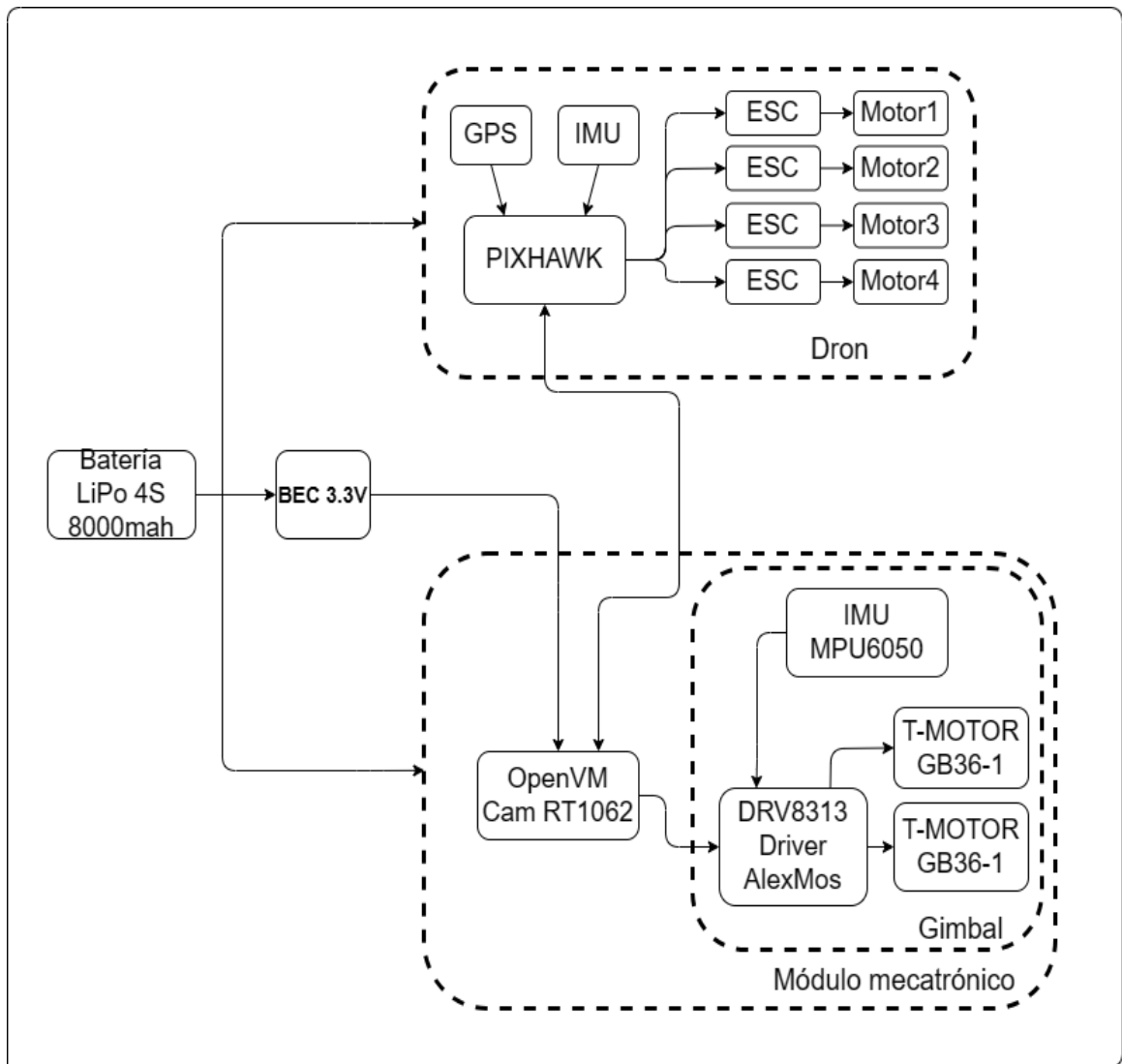


Figura 3.20. Diagrama de la estructura del hardware

Fuente: elaboración propia

CAPÍTULO IV

DISEÑO DEL CONTROL Y PROCESAMIENTO DE IMÁGENES

En este capítulo se trata el diseño de control y procesamiento de imágenes, desarrollando y explicando las operaciones y el funcionamiento del sistema.

4.1 Requerimientos de control y procesamiento de imágenes

- El módulo contará con un controlador especializado en la estabilización de la cámara en 2 ejes que posee su propio algoritmo.
- El módulo contará con un algoritmo de procesamiento de imágenes que permita el reconocimiento y seguimiento de la plataforma de aterrizaje.
- El módulo contará con un algoritmo que controle los movimientos del dron teniendo como entrada la posición relativa de la plataforma de aterrizaje.
- La data obtenida en cada aterrizaje deberá ser guardada para futuras revisiones y mejoras.

4.2 Operaciones del proceso del sistema

El módulo diseñado en este proyecto es totalmente autónomo una vez que es implementado en el dron, pues el usuario sólo debe designar un marcador específico para que el dron lo reconozca como la plataforma de aterrizaje. El diagrama de operaciones del sistema desde el punto de vista del usuario se explica con el siguiente diagrama, en la figura 4.1.

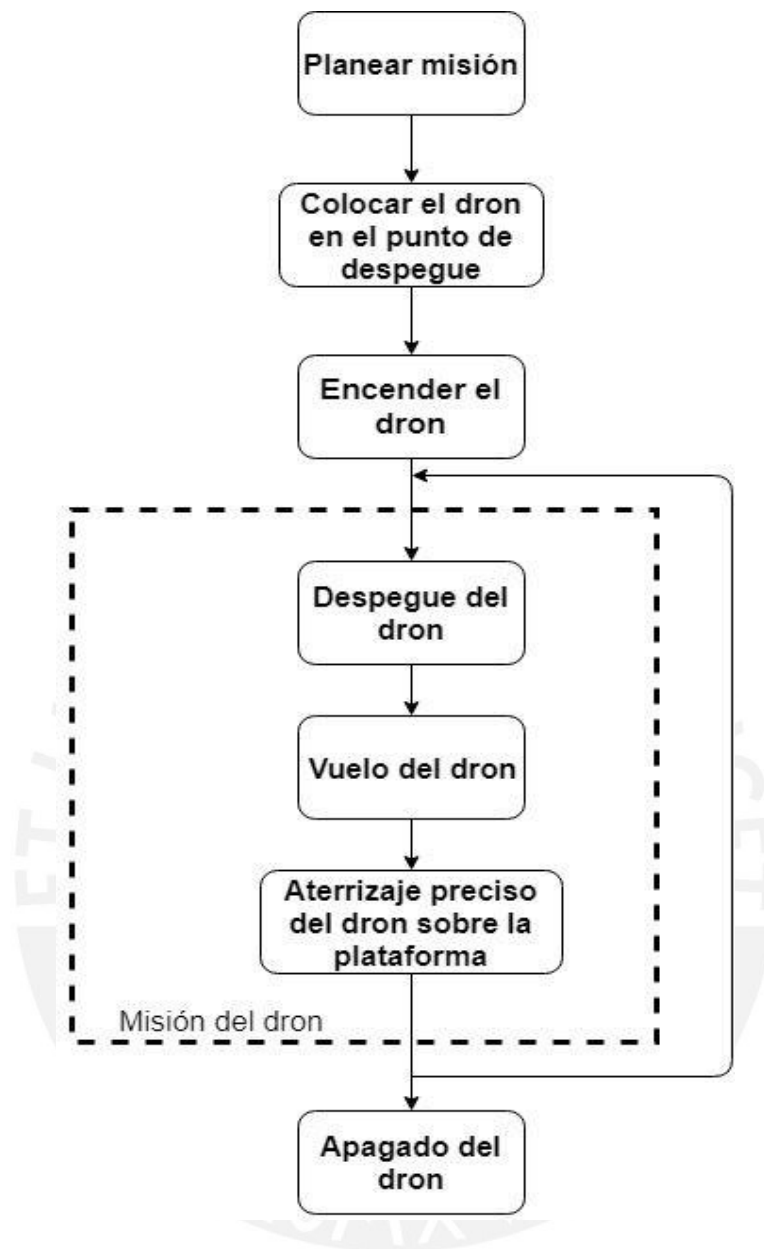


Figura 4.1 Diagrama de operaciones del sistema

Fuente: elaboración propia

4.3 Lógica de control del sistema

El aterrizaje autónomo y preciso se inicia con la lectura del GPS, al detectar el final de la misión; se estabiliza la cámara y se empieza a procesar las imágenes. Una vez la plataforma es detectada, se mueve al dron en dirección contraria a la posición relativa del dron respecto a la plataforma de aterrizaje, con el objetivo de que estos 2 se encuentren colineales, hasta culminar con el aterrizaje.

A continuación, se muestra la figura 4.2 explicando la lógica de control del sistema.

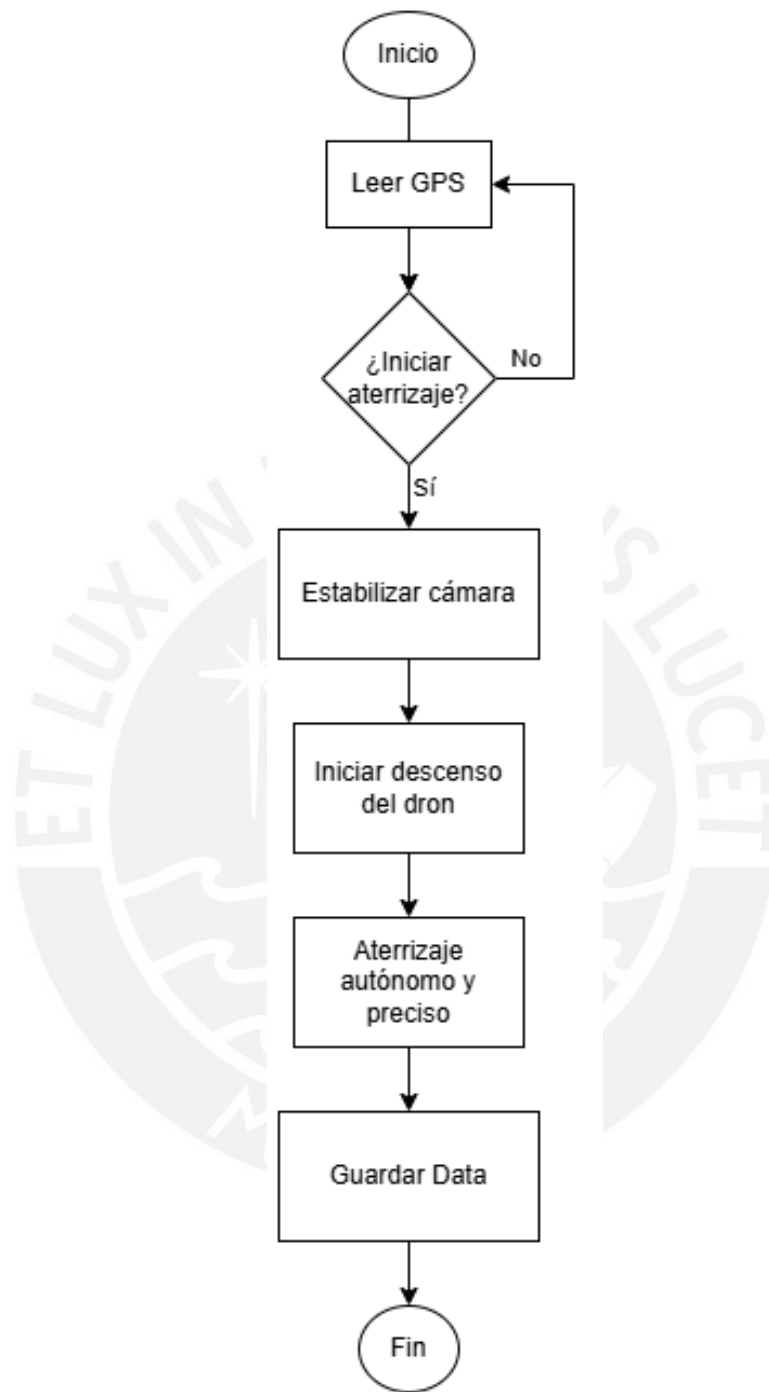


Figura 4.2 Diagrama de flujo del sistema

Fuente: elaboración propia

También se muestra a detalle la lógica del subproceso ‘Aterrizaje autónomo y preciso’ en la figura 4.3.

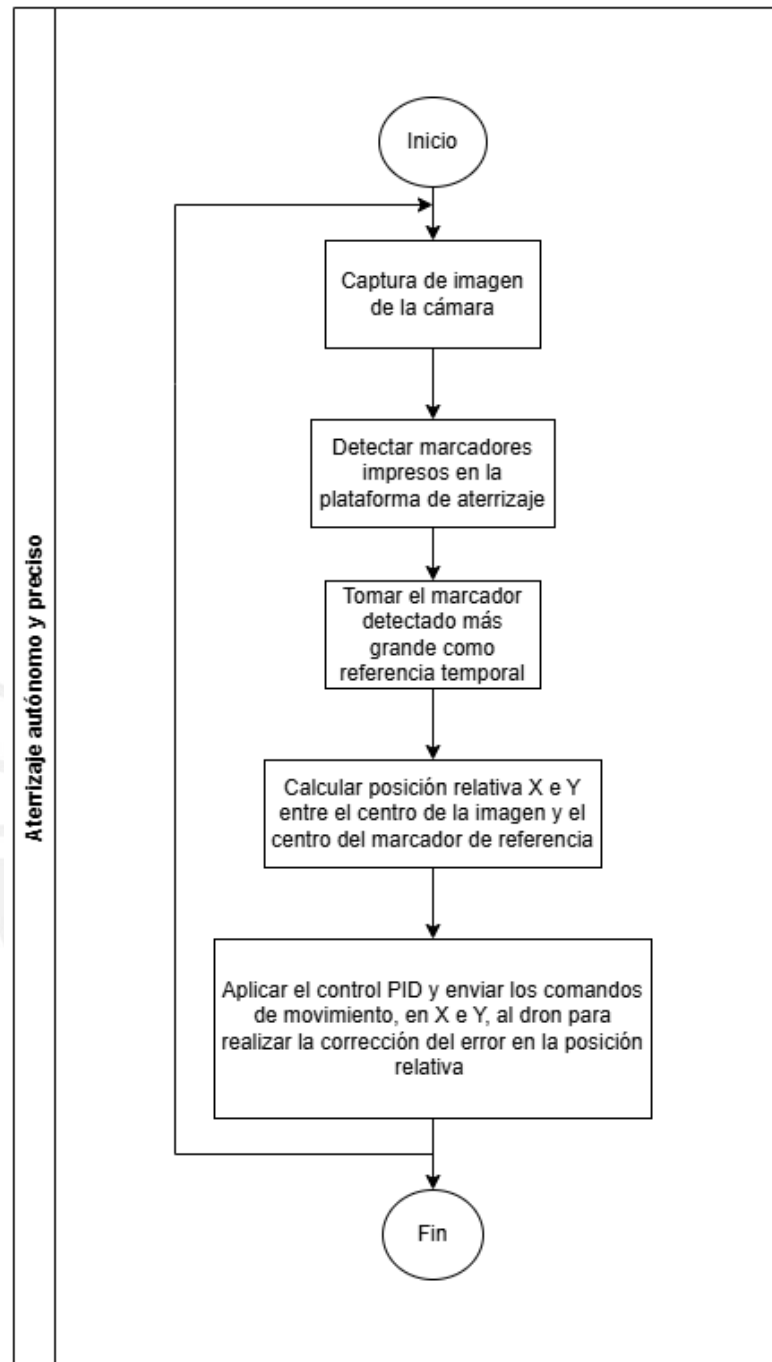


Figura 4.3 Detalle del subproceso ‘Aterrizaje autónomo y preciso’

Fuente: elaboración propia

4.4 Procesamiento de imágenes y control del dron

La librería que posee OpenCV permite la generación de marcadores disponibles de un extenso diccionario. El mayor beneficio de trabajar con marcadores ArUco es que proveen de suficiente información para detectar la posición relativa de la cámara, además de la codificación binaria interna los hacen especialmente robustos, permitiendo la posibilidad de manipular errores de detección y técnicas de corrección (OpenCV, 2024); además que todos los marcadores pueden ser procesados en paralelo, como se muestra en la figura 4.4.

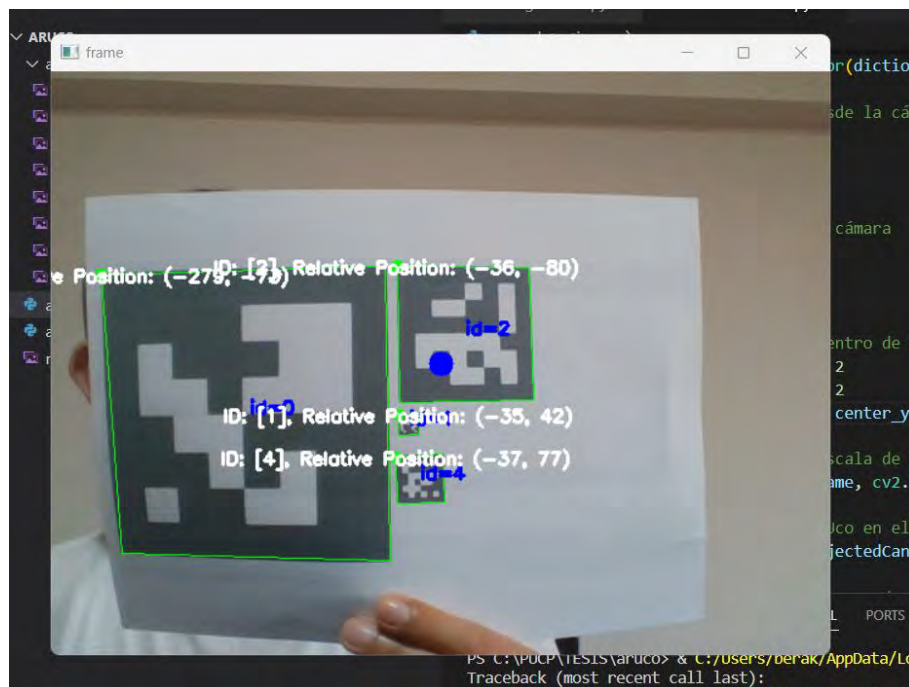


Figura 4.4 Ejemplos de reconocimiento de marcadores ArUco

Fuente: elaboración propia

Los 4 marcadores son todos de diferente tamaño para que puedan ser detectados a diferentes alturas, incluso cuando el dron se encuentre cerca de la plataforma. El marcador N°1 es el de mayor tamaño y tiene una medida de 80cm x 80cm y puede ser detectado desde una altura de aproximadamente 12m, el marcador N°2 tiene una medida de 40cm x 40 cm, el marcador N°3 tiene una medida de 15cm x 15cm y el marcador N°4 es el de menor tamaño y tiene una medida de 7cm x 7 cm y puede ser detectado hasta una altura mínima de 12cm; y se disponen como se muestra en la figura 4.5. El código en Python para crear estos marcadores se presenta en el anexo D de este documento.

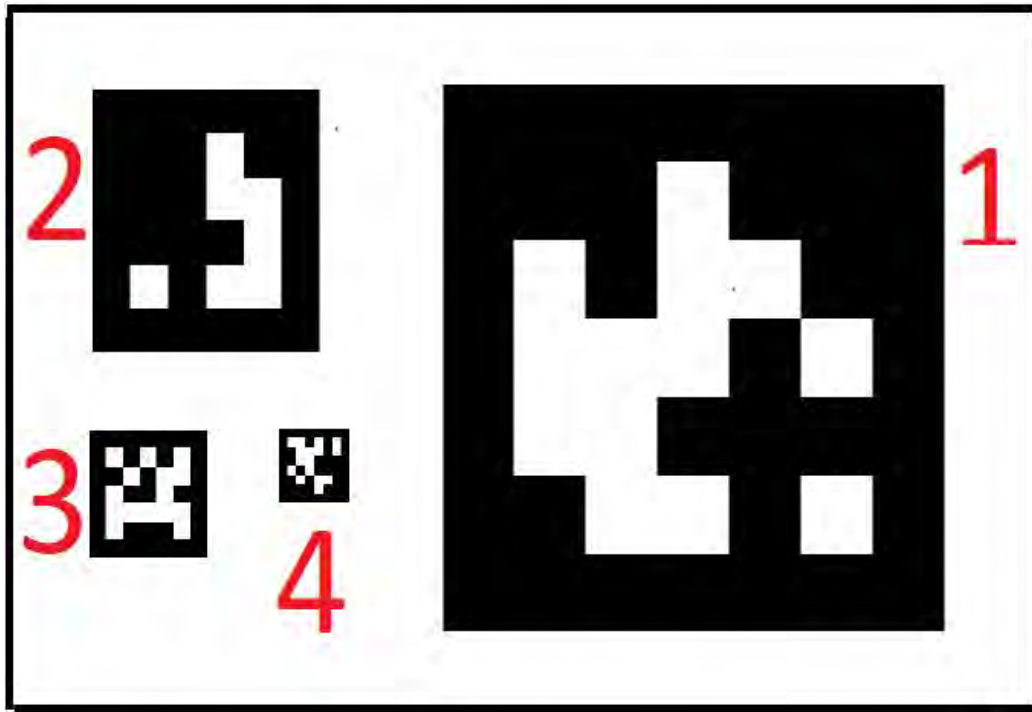


Figura 4.5 Arreglo de marcadores en la plataforma de aterrizaje

Fuente: Elaboración propia

A una determinada altura se detecta 1 o varios marcadores, y en caso de que suceda esto último, se escoge el marcador de menor tamaño y será tomado de referencia para realizar el control del dron.

El algoritmo prosigue de la siguiente manera, cuando el marcador es detectado, se extrae la posición en el eje X y en el eje Y, ambos en píxeles, y la orientación en grados sexagesimales. Luego de calcular el error entre las posiciones relativas del dron y la plataforma de aterrizaje, la cámara envía de manera serial los controles al Pixhawk, y este último es el encargado de interactuar directamente con el dron (VISION_LANDING, 2019).

Este proceso se presenta en la figura 4.6 y el código se presenta en el anexo E.

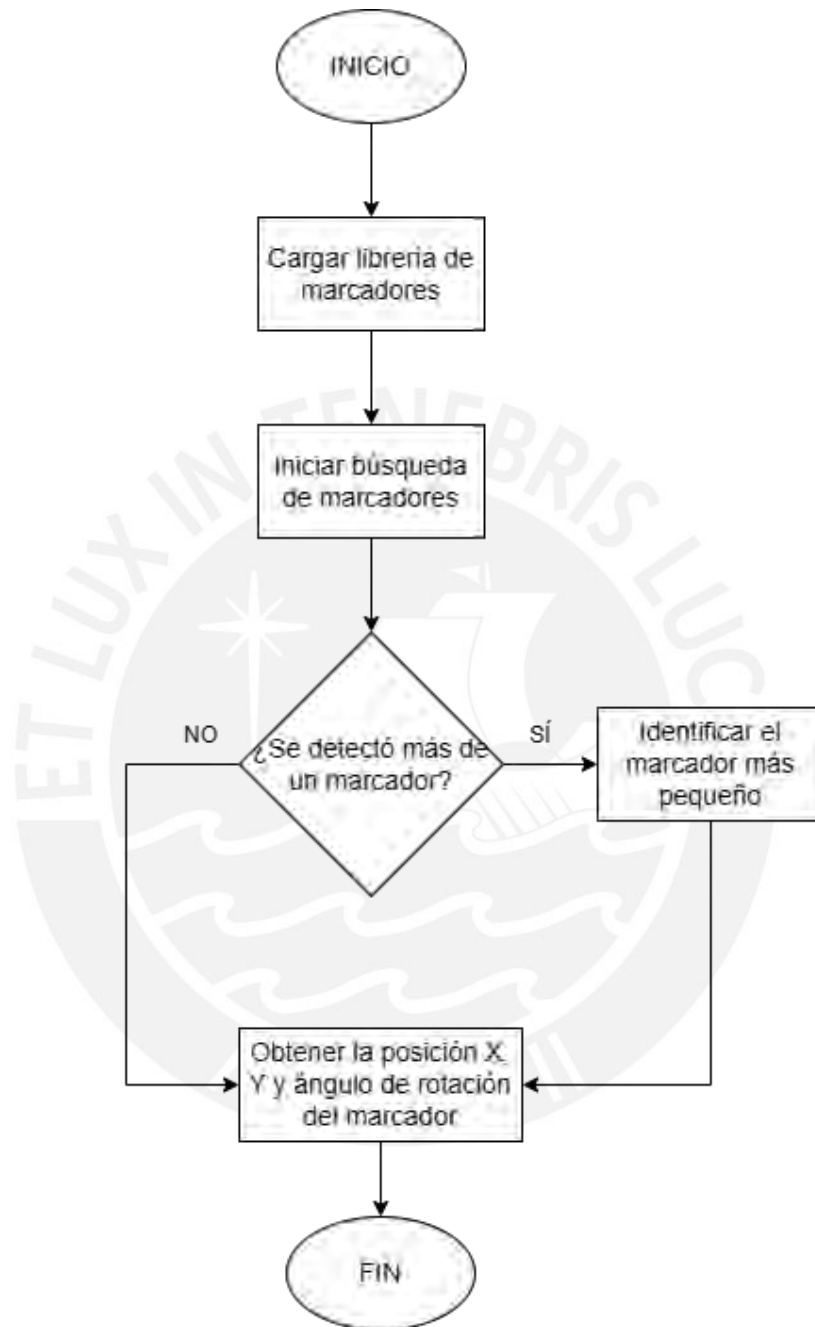


Figura 4.6 Lógica del procesamiento de imágenes

Fuente: elaboración propia

4.5 Integración del módulo mecatrónico

El módulo mecatrónico ya integrado con todos los componentes, tanto mecánicos como electrónicos se coloca en la parte inferior del dron, entre los pies de aterrizaje. Esto se muestra a continuación en las figuras 4.7, 4.8 y 4.9.

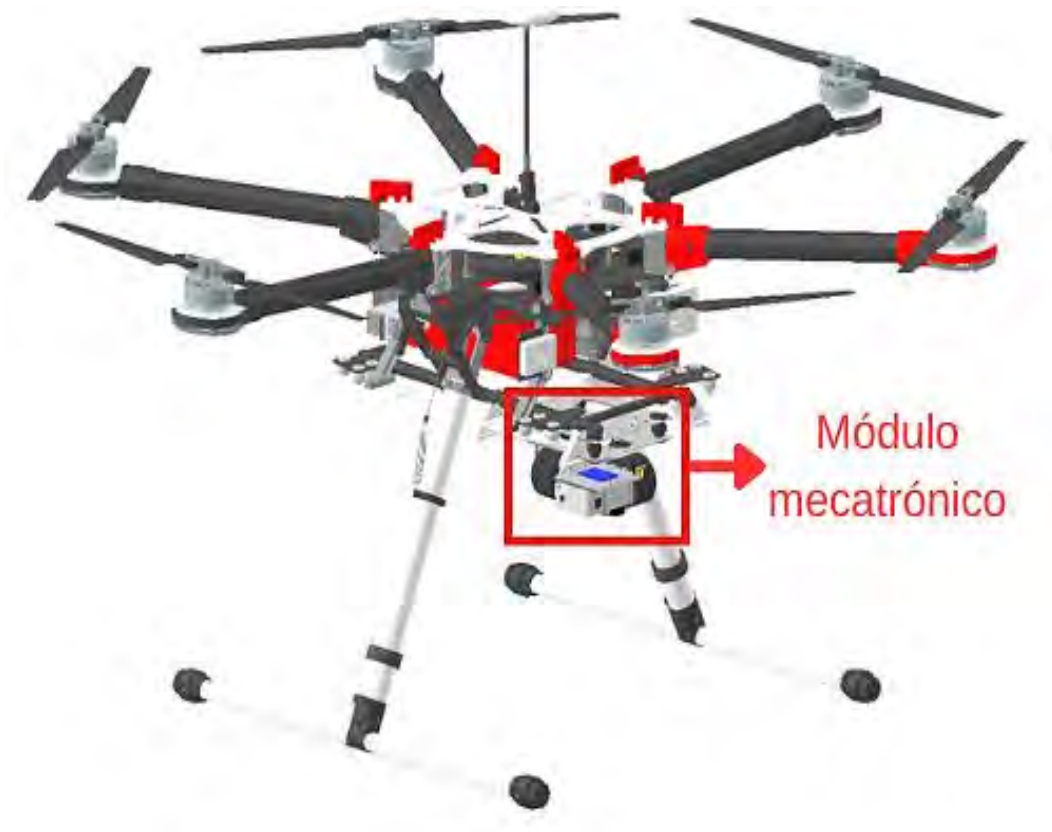


Figura 4.7 Dron con el módulo mecatrónico integrado

Fuente: elaboración propia

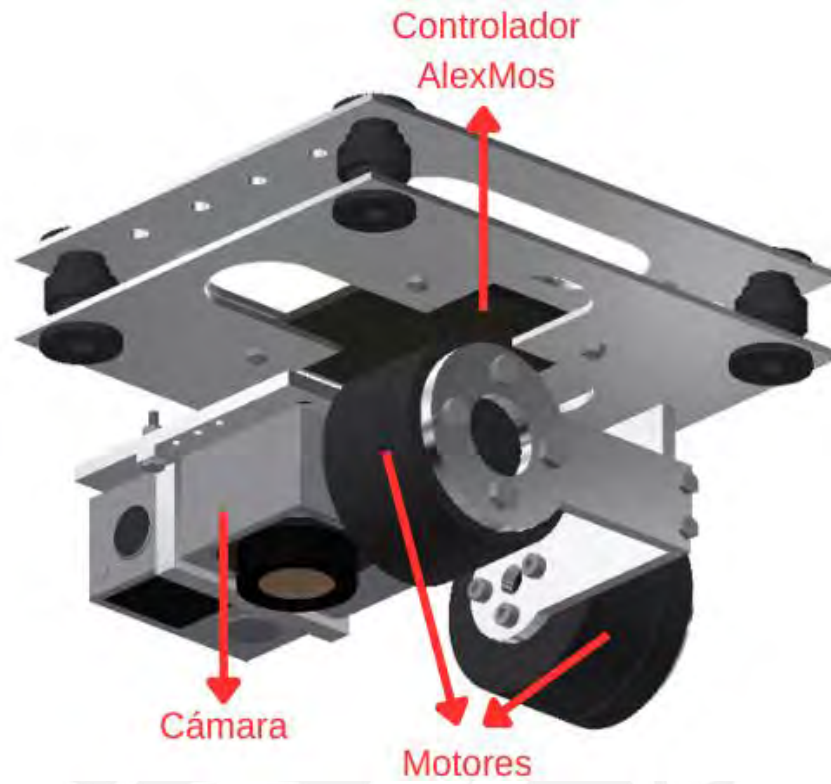


Figura 4.8 Módulo mecatrónico integrado

Fuente: elaboración propia

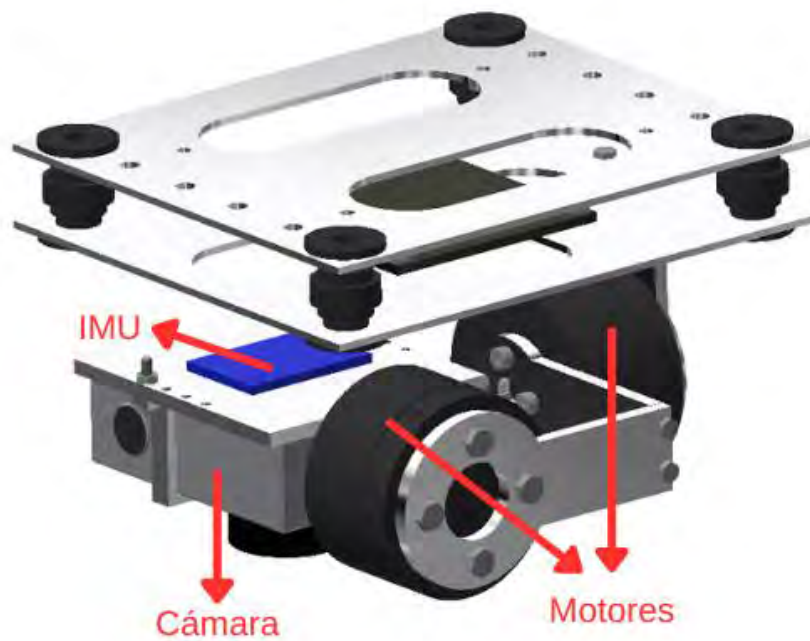


Figura 4.9. Módulo mecatrónico integrado

Fuente: elaboración propia

El controlador AlexMos es la tarjeta encargada de controlar el movimiento de los 2 motores, que se conectan en los puertos de Roll y Pitch, el sistema es realimentado con la medición de posición por parte del sensor IMU que se encuentra junto a la cámara para un mejor control de estabilidad. La cámara es la encargada también de realizar el procesamiento de imágenes, ya que al contar con microPython posee librerías que ofrecen soporte para realizar visión por computadora y el control del dron durante el aterrizaje, esto se logra ya que el microPython también posee soporte para enviar mensajes al controlador de vuelo Pixhawk vía protocolo Mavlink.

En la tabla 4.1 se presenta características generales del módulo mecatrónico integrado.

Tabla 4.1. características más importantes del módulo mecatrónico.

Fuente: elaboración propia

Peso Total	555 g
Dimensión Total	97mm x 120mm x 100mm
Consumo Total de corriente	4.52 A

CAPÍTULO V

COSTOS DEL SISTEMA

En este capítulo, se analizan los costos que implican la implementación del sistema mecatrónico. Se presentan los costos de los componentes electrónicos, componentes mecánicos, costos de la fabricación de las piezas requeridas para el sistema y finalmente el costo total.

5.1 Costos de los componentes electrónicos

En la tabla 5.1 se muestran los componentes electrónicos, así como las respectivas cantidades y precios, con el tipo de cambio 3.730 establecido por el BCR a la fecha del 03 de enero de 2024, en los cuales está incluido el precio de envío.

Tabla 5.1: Costos de componentes electrónicos.

Fuente: Elaboración Propia

Componente	Precio unitario (US\$)	Cantidad	Precio con envío incluido (US\$)	Precio (S/.)
Motor T-MOTOR GB36-1	49.90	2	149.70	558.38
DRV8313 Driver e IMU	220.40	1	330.60	1233.14
OpenMV Cam RT1062	120.00	1	180.00	671.40
			TOTAL	2462.92

Los componentes vienen con los respectivos cables y conectores, por lo que no es necesario realizar compras extras. En conclusión, el costo total de los componentes electrónicos es de S/. 2462.92.

5.2 Costos de materiales y fabricación

En la tabla 5.2 se presenta el costo de fabricación de la parte mecánica. El costo presentado fue cotizado en el laboratorio de manufactura de la sección de Ingeniería Mecánica de la Pontificia Universidad Católica del Perú.

Tabla 5.2: Costos de fabricación y diseño.

Fuente: Propia

Componente	Costo (S/.)
Material (Aluminio)	105.00
Procesos de fabricación	170.00
Tornillos (x24)	7.00
Diseño	360.00
TOTAL	642.00

El costo presentado en la anterior tabla incluye el precio del material, fabricación y de diseño, en el que se considera S/.90.00 por hora y 4 horas de trabajo, lo que resulta en un costo total de S/. 642.00.

5.3 Costos totales

En la tabla 5.3 se puede observar el costo total del sistema. Para poder determinar el costo total se sumaron los costos totales de componentes electrónicos y costos totales de fabricación y diseño.

Tabla 5.3: Costo total del sistema.

Fuente: Elaboración Propia

Concepto	Costo (S/.)
Costo total de componentes electrónicos	2462.92
Costo total de fabricación y diseño	642.00
Total	3104.92

El costo total de todo el sistema es S/. 3104.92.

CONCLUSIONES

1. Se logró realizar el diseño mecánico del módulo, cumpliendo con los requerimientos correspondientes, principalmente de peso y tamaño. La estructura está hecha de aluminio, con un peso total de 320 g y una medida de 12cm x 10cm x 8.015cm. La estructura consta de 8 piezas en total, para las cuales se presenta sus respectivos planos de fabricación y uno adicional de ensamblaje; también se presentan los análisis correspondientes para esfuerzos de corte y flexión y se comprueba que los componentes mecánicos no fallarán. Finalmente, se realizó una maqueta de la estructura, la cual se adjunta en el anexo F.
2. Se logró realizar el diseño electrónico del sistema, cumpliendo con los requerimientos correspondiente. El sistema consta de 5 componentes electrónicos: 2 motores brushless, 1 controlador de motores, 1 sensor IMU y la cámara OpenVM. Los cuales con completamente compatibles como se demuestra en el diagrama de conexiones y en la arquitectura de hardware. Se cumple con los requerimientos planteados, el marcador más grande (12cm x 12cm) es detectado hasta una altura de 12m y el tiempo de respuesta del sistema de control que estabiliza la cámara es de 0.0005 segundos.
3. Se logró diseñar la lógica y elaborar el código del procesamiento de imágenes usando una plataforma de aterrizaje con un arreglo de 4 marcadores ArUco de distintos tamaños, dispuestos de tal manera que el más pequeño se encuentre en el centro de la plataforma de aterrizaje, ya que será el que se tome como última guía cuando el dron se encuentre muy próximo a la plataforma y concluya el aterrizaje, todo este procesamiento lo realiza la cámara OpenCam con el uso de openCV, disponible para python. Así como el diseño de la lógica de control del dron, el cual consiste en enviar señales de movimiento del dron desde la cámara OpenCam hacia el controlador de vuelo Pixhawk a través de la librería MavLink, disponible para python. Se realizaron las simulaciones y pruebas respectivas y se comprobó que puede realizarse un procesamiento correcto.

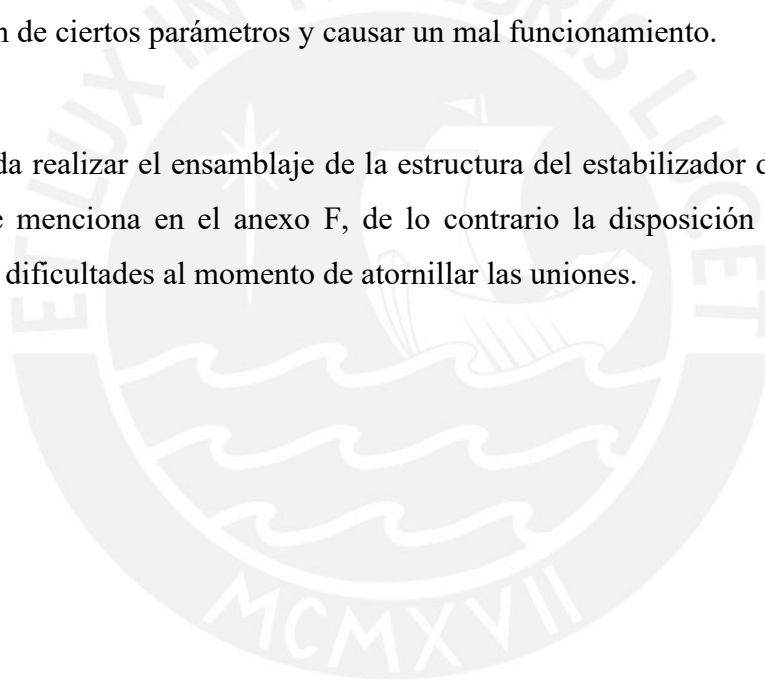
4. Se realizó el detalle económico del desarrollo de la presente solución propuesta, considerando el costo de componentes, materiales, fabricación y diseño, siendo un total de 3063.72 soles, cumpliendo con los requerimientos económicos establecidos. El detalle y la sustentación de la cotización se adjunta en el anexo G.

5. Con la pruebas y simulaciones realizadas se comprueba que es posible controlar los movimientos del dron hasta lograr el aterrizaje a través del procesamiento de imágenes adquiridas por una cámara estabilizada por un módulo mecatrónico.



RECOMENDACIONES

1. Se recomienda realizar la identificación de función de transferencia de cada motor antes de tunear los controladores PID, pues a pesar de ser motores de la misma marca y modelo, ningún componente es exactamente igual a otro.
2. Se recomienda realizar el tuneado del controlador de los motores cuando el módulo se encuentra totalmente armado con todos los componentes y partes.
3. Se recomienda revisar las actualizaciones en el firmware tanto del controlador AlexMos como de la cámara OpenVM Cam RT1062, pues en versiones nuevas podría cambiar la configuración de ciertos parámetros y causar un mal funcionamiento.
4. Se recomienda realizar el ensamblaje de la estructura del estabilizador de la cámara en el orden que se menciona en el anexo F, de lo contrario la disposición de ciertas piezas puede causar dificultades al momento de atornillar las uniones.



REFERENCIAS

- Salazar, P. (2020, 03 de marzo). Aterrizaje de precisión de un UAV sobre una plataforma móvil. Recuperado de:
<https://tesis.pucp.edu.pe/items/22c8eda0-5385-4738-9020-52231e481d5c>
- Gamarra, L. (2015). Un negocio de alto vuelo que va creciendo. Recuperado de:
<https://gestion.pe/especial/50-ideas-de-negocios/noticias/negocio-alto-vuelo-que-va-creciendo-noticia-1992621>
- Torres, A. (2015, 16 de mayo). Uso de drones en seguridad. Recuperado de:
<http://dronespain.pro/uso-drones-en-seguridad/>
- Raja That. (2016, 15 de agosto). \$3000 DJI Inspire 1 Drone crash. Recuperado de:
<https://www.youtube.com/watch?v=UOAYTy44uIE>
- Parker, S. (2015). Drone Landing on a Boat [videograbación]. Recuperado de:
<https://www.youtube.com/watch?v=cdwbjVYFges>
- FLYTREX. Making backyard deliveries a reality. Recuperado de:
<https://www.flytrex.com/>
- Porter, J. (2019, 9 de abril). Google's Wing drones approved to make public deliveries in Australia. Recuperado de:
<https://www.theverge.com/2019/4/9/18301782/wing-drone-delivery-google-alphabet-canberra-australia-public-launch>
- Palou, N. (2017, 26 de enero). Un megadron capaz de levantar 300kg. Recuperado de:
<https://www.microsiervos.com/archivo/drones/megadron-levantar-300-kg.html>

SALAZAR, J. (2017). Diseño de un cuadricóptero para transporte de medicina en la amazonía peruana. Recuperado de:
<http://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/8120>

GUTIERREZ, S (2015). Publican Ley 30740, que regula el uso de aeronaves pilotadas a distancia. Recuperado de:
<https://legis.pe/ley-30740-regula-uso-aeronaves-pilotadas-distancia/>

OPAC COMPONENTS. Tornillería 12.9 de alta resistencia. Recuperado de:
<https://www.opac.net/pdf/H912.pdf>

T-MOTOR (2019). GB36-1. Recuperado de:
<http://store-en.tmotor.com/goods.php?id=441>

BASECAM (2019). BaseCam SimpleBGC 32-bit. Recuperado de:
<https://www.basecamelectronics.com/simplebgc32bit/>

MDI (2014). About the 32-bit Alexmos Controller board. Recuperado de:
<https://mdifilm.com/2016/2014/09/14/8bit-alexmos-controller-board/>

OPENMV (2019). OpenMV Cam H7. Recuperado de:
<https://openmv.io/collections/cams/products/openmv-cam-h7>

OPENCV (2024). Detection of ArUco markers. Recuperado de:
https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html

Anexo A: Norma Técnica complementaria NTC 001-2015



PERÚ

Ministerio
de Transportes
y Comunicaciones

Viceministerio
de Transportes

Dirección General
de Aeronáutica Civil

- (13) En condiciones nocturnas (después de la puesta o antes de la salida del sol)
 - (14) Durante más de una hora continuada.
 - (15) Durante más del 80% de la autonomía establecida por el fabricante.
 - (16) Con un piloto que no haya sido acreditado y/o que no haya sido autorizado.
 - (17) Descuidando la atención exclusiva al control de la operación del RPAS. Está prohibido entregar el control a otro piloto/operador mientras el RPA está en vuelo y comandar dos RPA simultáneamente.
 - (18) Operar un RPAS bajo la influencia de las drogas o el alcohol.
 - (19) A menos de 4 km de un aeródromo, excepto para fines de prevención de impactos con aves, conforme a lo estipulado en el inciso (g) de este párrafo.
 - (20) Sobre vías de comunicación, incluyendo toda infraestructura vial (viaductos, carreteras, caminos, senderos, puentes), infraestructura de transmisión eléctrica y de telecomunicaciones (postes, torres, cables y antenas), cursos de agua navegables y ductos para transporte de hidrocarburos. Sin perjuicio de ello, los RPA podrán volar próximos a estas vías, manteniendo una separación horizontal mayor de 30 metros, del borde o extremo de las mismas.
 - (21) No se podrá dejar caer ni lanzar objetos, material o fluidos desde el RPAS salvo en los casos en que cuente con autorización expresa de la DGAC sobre la base de informes técnicos de impacto ambiental emitidos por la autoridad competente.
 - (22) "En zonas peligrosas, zonas prohibidas y zonas restringidas publicadas en la AIP-PERÚ o en los NOTAM incluidos en el sitio web de CORPAC". Una desviación a esta restricción se sujetará a la autorización escrita de parte de la Entidad responsable de la reserva de la zona en cuestión.
 - (23) En una navegación aérea internacional o en alta mar, salvo que cuente con una autorización apropiada de la DGAC y de acuerdo a lo establecido en el Apéndice M de la RAP 91.
- (f) *Operación en zonas urbanas:* La operación de un RPAS en zonas urbanas podrá autorizarse excepcionalmente a los RPA con un peso máximo de despegue de hasta 6 kg equipados con paracaídas de emergencia, sólo en los siguientes casos y según el procedimiento señalado más adelante en el sub párrafo (i):
- (1) Entidades gubernamentales por razones de seguridad ciudadana y otras actividades calificadas de interés público por la propia entidad.
 - (2) Entidades privadas contratadas por entidades gubernamentales para actividades calificadas de interés público por la entidad contratante.
 - (3) Entidades privadas que prestan servicios declarados de interés público por Resolución de la Entidad competente. Tales servicios pueden incluir la protección del patrimonio arqueológico, investigación científica, prevención de desastres, etc.
 - (4) Cuerpo General de Bomberos Voluntarios del Perú y el Instituto Nacional de Defensa Civil (INDECI), en actividades de búsqueda y salvamento, en el combate de incendios y otras emergencias.
- (g) *Operación de RPAS para control de peligro aviario:* La DGAC podrá autorizar el uso de RPAS dentro de los límites de un aeródromo y en la proximidad de este, bajo un "Protocolo de Seguridad de Uso de RPAS para el Control de peligro Aviario", el mismo que deberá estar aprobado por el explotador del aeródromo y que deberá contener como mínimo las siguientes previsiones:
- (1) Delimitación del espacio aéreo del aeródromo, características del aeródromo y zonas de riesgo
 - (2) Características de la fauna aviar vinculada a incidentes en el aeródromo y localización del peligro aviar




- (3) Mantenimiento del equipo RPAS
 - (4) Comprobaciones de seguridad de operación de los equipos
 - (5) Seguridad en las maniobras de vuelo
 - (6) Climatología del aeródromo
 - (7) Presentación de plan de vuelo y aceptación por el Servicio de Tránsito Aéreo
 - (8) Comunicaciones
 - (9) Registros
- (h) *Análisis de riesgo.* A excepción de las operaciones de emergencia realizadas por las Entidades establecidas en sub párrafo (f) inciso (4) precedente, toda operación en zonas urbanas y en la vecindad de aeródromos requiere que se efectúe una evaluación de riesgo según el Apéndice D (identificando los peligros y las acciones de mitigación y gestión del riesgo) a cargo del explotador o funcionario responsable designado por la entidad que opera el RPAS. Dicha evaluación fechada y firmada por el mencionado funcionario, antes de la operación deberá remitirse en copia escaneada por correo electrónico a la DGAC y ser registrada en un archivo disponible a la DGAC en el domicilio declarado por la entidad, conjuntamente con la copia de la póliza de seguros.
- (i) *Procedimiento de autorización para operación de RPAS en zonas urbanas y en la proximidad de aeródromos:* Un explotador autorizado a operar un RPAS conforme lo estipulado en el párrafo 8 sub párrafo (d) de esta NTC, podrá ser autorizado excepcionalmente por la DGAC, mediante Oficio de la Dirección de Certificaciones y Autorizaciones, a operar en zonas urbanas o en la proximidad de aeródromos, sujeto a cumplir los requisitos detallados en los sub párrafos (f) (g) y (h), precedentes y presentando los documentos siguientes:
- (1) Solicitud de autorización y análisis de riesgo conforme al formato que se indica en el sub párrafo (h) precedente.
 - (2) Constancia de pago de los derechos de tramitación correspondientes establecidos en el Texto Único de Procedimientos Administrativos – TUPA del Ministerio de Transportes y Comunicaciones.
- La DGAC resuelve las solicitudes de autorización en el plazo máximo de sesenta (60) días hábiles contados a partir del día siguiente de la fecha de presentación de la solicitud acompañada de la documentación completa o desde que esta se complete, de conformidad con el Artículo 95.2 de la Ley N° 27261, Ley de Aeronáutica Civil del Perú.
- (j) *Operaciones de los RPAS en recintos cerrados.*
- Los recintos completamente cerrados (un coliseo cerrado, un hangar cerrado, pabellón industrial o deportivo, un centro de convenciones, un domicilio particular, etc.) no están sujetos a la jurisdicción de la DGAC, al no formar parte del espacio aéreo. Los titulares de esos recintos son responsables de autorizar el vuelo de RPAs en su interior.

9. CONTACTOS PARA MAYOR INFORMACIÓN


Para cualquier consulta adicional referida a esta NTC, diríjirla a la Coordinación Técnica de Autorizaciones, teléfono: +51-1-6157800, anexos 1523 y 1156, ó al correo electrónico: rpas-dgac@mtc.gob.pe

Anexo B: OPAC tornillos según ISO-4762 (DIN 912)




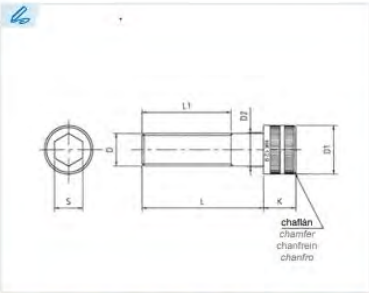
TORNILLERÍA 12.9 DE ALTA RESISTENCIA

12.9 HIGH STRENGTH FASTENERS • VISSERIE 12.9 À HAUTE RÉSISTANCE • PARAFUSOS 12.9 DE ALTA RESISTÊNCIA



ISO 4762 (DIN 912)





Tornillo de cabeza cilíndrica con hexágono interior
Socket head cap screw
Vis six pans creux tête cylindrique
Parafuso de cabeça cilíndrica com sextavado interior

Acero aleado de alta resistencia
Heat treated alloy steel
Acier allié haute résistance
Liga de aço de alta resistência

Calidad: 12.9
Property class: 12.9
Classe: 12.9
Classe de resistência: 12.9

Normas ISO 898-1
Standards ISO 898-1
Normes ISO 898-1
Normas ISO 898-1

Holo-Code: M5 - M36


D	M2 (x0,40)	M2,5 (x0,45)	M3 (x0,50)	M4 (x0,70)	M5 (x0,80)	M6 (x1,00)	M8 (x1,25)	M10 (x1,50)	M12 (x1,75)
D1 (max.)	3,8	4,5	5,68	7,22	8,72	10,22	13,27	16,27	18,27
D2 (max.)	2,6	3,1	3,6	4,7	5,7	6,8	9,2	11,2	13,7
K (max.)	2	2,5	3	4	5	6	8	10	12
S nom.	1,5	2	2,5	3	4	5	6	8	10
L1* (min.)	16	17	18	20	22	24	28	32	36
Sección útil mm ²	2,07	3,39	5,03	8,78	14,2	20,1	36,6	58	84,3
N_{min}	0,69	1,43	2,48	5,85	12,0	20,3	48,8	97,5	165
	4	5	5	5	6	8	10	10	12
	5	6	6	6	8	10	12	12	15
	6	8	8	8	10	12	14	15	16
	8	10	10	10	12	14	15	16	20
	10	12	12	12	14	15	16	18	25
	12		15	14	15	16	18	20	30
			16	15	16	18	20	22	35
			20	16	18	20	22	25	40
			25	18	20	22	25	30	45
			30	20	22	25	30	35	50
			35	22	25	30	35	40	55
				30	35	40	45	50	65
				35	40	45	50	55	70
				40	45	50	55	60	75
				45	50	55	60	65	80
				50	55	60	65	70	90
				60	65	70	75	80	100
				65	70	75	80	85	110
				70	75	80	85	90	120
				80	80	90	90	100	130
				90	90	100	100	110	140
				100	100	110	110	120	150
					110	120	120	130	160
						120	130	140	170
							140	150	180
							150	160	190
							160	180	200
								200	220
								220	240
								240	260
								260	280
								280	300
								300	

Ref. ?

ejemplo Ref. = H912-XX-XXX
example
exemple
exemplo

DIN 912 D L
M6x20

Ref. = H912-6-20



www.opac.net



Resistencia a la tracción 1220 N/mm² min.
Tensile strength
 Résistance à la rupture
 Resistência à tração

Temperaturas de uso -29° a +204°C
Working temperatures
 Températures d'emploi
 Temperaturas de utilização

Dureza 39 HRC min.
Hardness
 Dureté
 Dureza

Alargamiento mínimo a la ruptura 8%
Minimum elongation at fracture
 Allongement minimum à la rupture
 Elongação mínima à ruptura

ES EN FR PT

Sección útil mm²
 Usable Section mm²
 Section utile mm²
 Seção útil mm²

Par de apriete recomendado
Recommended tightening torque
 Couple de serrage recommandé
 Binário de aperto recomendado

* Los tornillos por encima de la línea azul están totalmente roscados, por debajo de la línea disponen de roscá parcial según L1.
 * According to L1, fasteners above the blue line are completely threaded while those below are partially threaded.
 * Vis entièrement filetés pour les dimensions au dessus de la ligne, vis partiellement filetés pour les dimensions en dessous de la ligne suivant L1.
 * Os parafusos acima da linha azul estão totalmente roscados, abaixo da linha dispõem de roscá parcial conforme L1.

Cantidad en caja
 Box qty:
 Boilage
 Quantidade na caixa

■ x1 ■ x25
■ x5 ■ x50
■ x10 ■ x100

M14 (x2,00)	M16 (x2,00)	M20 (x2,50)	M24 (x3,00)	M27 (x3,00)	M30 (x3,50)	M36 (x4,00)	M42 (x4,50)	M48 (x5,00)	D
21,33	24,33	30,33	36,39	40	45,39	54,46	63,46	72,46	D1 (max.)
15,7	17,7	22,4	26,4	30,4	33,4	39,4	45,6	52,6	D2 (max.)
14	16	20	24	27	30	36	42	48	K (max.)
12	14	17	19	19	22	27	32	36	S nom.
40	44	52	60	66	72	84	96	108	L1* (min.)
115	157	245	353	459	561	817	1.120	1.470	Sección útil mm ²
265	413	825	1.425	2.008	2.775	4.875	7.536	11.545	

20	25	30	40	50	50	80	60	70	
25	30	35	45	60	60	90	65	80	
30	35	40	50	70	70	100	70	90	
35	40	45	55	80	80	110	80	100	
40	45	50	60	90	90	120	90	110	
45	50	55	65	100	100	130	100	120	
50	55	60	70	110	110	140	110	130	
55	60	65	75	120	120	150	120	140	
60	65	70	80	130	130	160	130	150	
65	70	75	90	140	140	170	140	160	
70	75	80	100	150	150	180	150	180	
75	80	90	110	160	160	200	160	200	
80	90	100	120	180	180	220	180	220	
85	100	110	130	200	200	240	200	240	
90	110	120	140	220	220	260	220	260	
100	120	130	150	240	240	280	240	280	
110	130	140	160	260	260	300	260	300	
120	140	150	180	280	280	340	280	320	
	150	160	200	300	300	360	300	340	
	160	180	220	320	320	380	320	360	
	180	200	240	340	340	400	340	380	
	200	220	260	360	360	420	360	400	
	210	240	280	380	380	440	380	420	
	220	260	300	420	420	460	400	440	
	230	280	320	480	480	500	420	460	
	240	300	340	520	520	540	440	480	
	250	320	360	540	540	560	460	500	
	260	340	380	560	560	580	480	520	
	280	400		580	580	600	500	540	
	300			600	600	640			
	320			640	640	650			
	340			650	650				

M42 y M48 bajo consulta.
 M42 & M48 upon request.
 M42 et M48 sur demande.
 M42 e M48, disponíveis sob consulta.

Anexo C: Guía de usuario para tuneado de los motores y calibración del sensor IMU

2. Calibrating the sensor

Make sure IMU sensor is connected and recognized by the system: arrows on the gauge panel reflects a rotation of a sensor.

Configure sensor orientation by setting **Axis TOP** and **Axis RIGHT** parameters. The fastest way is to use the auto-detection utility: press **AUTO** button and follow the instructions (at first step, level sensor, then tilt it on the right side). More details you can find in the [Main IMU sensor](#) section.

Calibrating Gyroscope

The Gyro sensor is calibrated every time you turn the controller on, and it takes about 4 seconds to complete. Try to immobilize the camera sensor as hard as you can in first seconds after powering on while signal LED is blinking. After powering on you have 1 seconds to freeze the gimbal before calibration starts.

If you activated option "Skip gyro calibration at startup" then the gyro is not calibrated each time and the controller begins operating immediately after powering up. Be careful and recalibrate the gyro manually if you notice anything wrong with IMU angles.

Calibrating Accelerometer

You must perform ACC calibration only once, but it's recommended to recalibrate it from time to time or when the temperature significantly changes. Alternatively you can make a temperature calibration through a full range of possible working temperatures (see [Temperature Sensor Calibrating](#)).

IMPORTANT: Before processing any kind of calibration, you need to reset old values by pressing "RESET" button in the "IMU Calibration helper" window!

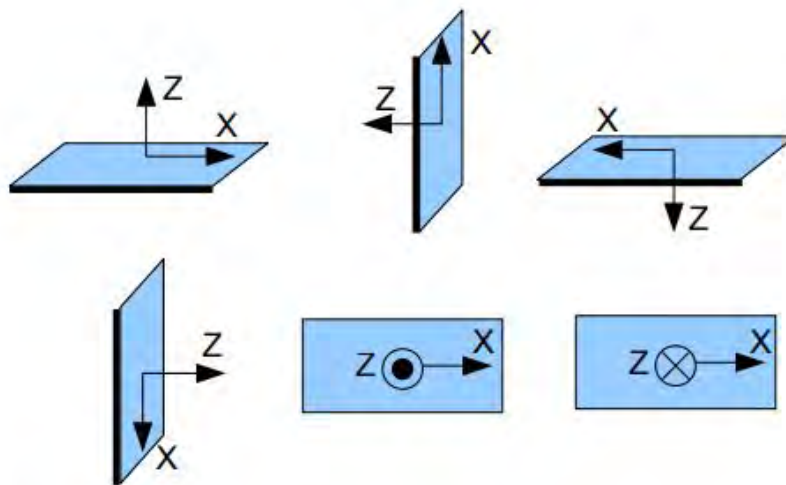
- **Simple calibration mode:** set the sensor horizontally, and press **CALIBRATE.ACC** button in the GUI (or the menu button, if it's assigned to "Calibrate ACC" action). The LED will blink for 2 seconds. Be sure not to allow the sensor to move during calibration.
- **Advanced 6-point mode (recommended):** turn sensor in order such that each side of the sensor

looks up (6 positions at all, including base one). To do this fix the sensor in each position, then press **CALIB.ACC** button in the GUI, and wait about 2-3 seconds (until the LED is stops flashing). *You do not have to press the WRITE button at each step, calibration data is written automatically (the data is written when the LED stops flashing for each orientation performed).*

To calibrate second sensor placed on the frame (if present), select it by the toggle buttons "Camera IMU/Frame IMU". All raw sensor data, IMU angles and all calibration commands now relates to selected sensor.

To simplify the 6-point calibration, use the "IMU Calibration helper" tool. It will show a currently selected position and positions that are already calibrated.

NOTE: Precise accelerometer and gyro calibration is a very important for horizon holding during dynamic flying or YAW rotation. it's advised to use a temperature compensation to keep precise operation in a wide range of environmental temperatures (see [Temperature Sensor Calibrating](#)).



3. Setting up basic parameters

- For 1- or 2-axis system, disable unused motor outputs in the "Hardware" tab - "Motor outputs" group. It is important to tell system how many active motors are connected, to select proper stabilization algorithm.
- Set **POWER** according to the motor configuration (see recommendations below)
- Connect the main power supply. Motors should start to spin and if all is okay at this moment – will stabilize camera. If motors spin randomly – check that sensor orientation is correct and motor outputs are assigned to proper motors.
- Auto-detect number of poles and motors direction in the "Hardware" – "Motor configuration" – "AUTO". Do not proceed to next step until proper direction is detected! It is normal if number of poles is detected with small error – in such case enter the correct value manually.

- Set parameters "Stabilization settings" – '**Gain multiplier**' to 1.0, '**Outer P**' to 100 for all axes (default values).
- Run auto-tuning for PID-controller, using default settings the first time.
- Adjust PID controller settings if required. To check stabilization quality use the peak indicator in the control panel (shown by the blue traces and blue numbers). Incline the frame by small angles and try to minimize peak values by increasing P, I and D to its maximum. You may use gyro data from the Monitoring tab to estimate stabilization quality too.

It is better to tune PID with the "Follow Mode" turned OFF for all axes.

Suggested algorithm for manual PID tuning:

1. Set I=0.01, P=10, D=10 for all axes. Gimbal should be stable at this moment. If not, decrease P and D a bit. Then start to tune each axis sequentially:
2. Gradually increase P until motor starts to oscillate (you may knock the camera and see on the gyro graph, how fast oscillation decays). Increase D a little – it should dampen oscillations, and decay time decreases. The lower is decay time, the better.
3. Repeat step 2 until D reaches its maximum which is when high-frequency vibration begins to appear (you may hear it or feel it in your hands and see noisy lines on the gyro graph). When this begins current P and D values are at maximums for your setup. At this point decrease them a little and go to step 4.
4. Increase I until low-frequency oscillation starts. Decrease I a little to keep gimbal stable. Now you have found a maximum for all PID values for selected axis. Repeat from step 1 for other axes.
5. When all axes are tuned in static, try to move gimbal's frame, emulating a real working environment. You may notice that cross-influence of axes may make gimbal unstable. In this case, decrease a little PID values from their maximum for axes that are animating.

Good tuning results in stabilization error of less than 1 degree when you slightly rock the gimbal's frame.

Further steps to improve the precision of stabilization:

- Connect, setup and calibrate second (frame) IMU (see [Second IMU sensor](#)).

Anexo D: Código de generación de marcadores Aruco

```
import numpy as np
import cv2
ARUCO_DICT = {
    "DICT_4X4_50": cv2.aruco.DICT_4X4_50,
    "DICT_4X4_100": cv2.aruco.DICT_4X4_100,
    "DICT_4X4_250": cv2.aruco.DICT_4X4_250,
    "DICT_4X4_1000": cv2.aruco.DICT_4X4_1000,
    "DICT_5X5_50": cv2.aruco.DICT_5X5_50,
    "DICT_5X5_100": cv2.aruco.DICT_5X5_100,
    "DICT_5X5_250": cv2.aruco.DICT_5X5_250,
    "DICT_5X5_1000": cv2.aruco.DICT_5X5_1000,
    "DICT_6X6_50": cv2.aruco.DICT_6X6_50,
    "DICT_6X6_100": cv2.aruco.DICT_6X6_100,
    "DICT_6X6_250": cv2.aruco.DICT_6X6_250,
    "DICT_6X6_1000": cv2.aruco.DICT_6X6_1000,
    "DICT_7X7_50": cv2.aruco.DICT_7X7_50,
    "DICT_7X7_100": cv2.aruco.DICT_7X7_100,
    "DICT_7X7_250": cv2.aruco.DICT_7X7_250,
    "DICT_7X7_1000": cv2.aruco.DICT_7X7_1000,
    "DICT_ARUCO_ORIGINAL": cv2.aruco.DICT_ARUCO_ORIGINAL,
    "DICT_APRILTAG_16h5": cv2.aruco.DICT_APRILTAG_16h5,
    "DICT_APRILTAG_25h9": cv2.aruco.DICT_APRILTAG_25h9,
    "DICT_APRILTAG_36h10": cv2.aruco.DICT_APRILTAG_36h10,
    "DICT_APRILTAG_36h11": cv2.aruco.DICT_APRILTAG_36h11
}

aruco_type = "DICT_6X6_1000"

for id in range(5):
    arucoDict = cv2.aruco.getPredefinedDictionary(ARUCO_DICT[aruco_type])
    print("ArUCo type '{}' with ID '{}'".format(aruco_type, id))
    tag_size = 250
    tag = np.zeros((tag_size, tag_size, 1), dtype="uint8")
    cv2.aruco.generateImageMarker(arucoDict, id, tag_size, tag, 1)
    tag_name = "arucoMarkers/" + aruco_type + "_" + str(id) + ".png"
    cv2.imwrite(tag_name, tag)
    cv2.imshow("ArUCo Tag", tag)

cv2.waitKey(0)

cv2.destroyAllWindows()
```

Anexo E: Código de procesamiento de imágenes y control del dron

```
import numpy as np
import cv2
from dronekit import connect, VehicleMode

# Establish a connection to the vehicle
vehicle = connect('/dev/ttyUSB0', baud=57600, wait_ready=True) # Update
the connection parameters based on your setup

class PID:
    def __init__(self, kp, ki, kd, setpoint=0):
        self.kp = kp
        self.ki = ki
        self.kd = kd
        self.setpoint = setpoint
        self.prev_error = 0
        self.integral = 0

    def update(self, current_value):
        error = self.setpoint - current_value
        self.integral += error
        derivative = error - self.prev_error

        output = self.kp * error + self.ki * self.integral + self.kd *
derivative

        self.prev_error = error
        return output

# Definición de un diccionario de códigos ArUco disponibles
ARUCO_DICT = {
    "DICT_4X4_50": cv2.aruco.DICT_4X4_50,
    "DICT_4X4_100": cv2.aruco.DICT_4X4_100,
    "DICT_4X4_250": cv2.aruco.DICT_4X4_250,
    "DICT_4X4_1000": cv2.aruco.DICT_4X4_1000,
    "DICT_5X5_50": cv2.aruco.DICT_5X5_50,
    "DICT_5X5_100": cv2.aruco.DICT_5X5_100,
    "DICT_5X5_250": cv2.aruco.DICT_5X5_250,
    "DICT_5X5_1000": cv2.aruco.DICT_5X5_1000,
    "DICT_6X6_50": cv2.aruco.DICT_6X6_50,
    "DICT_6X6_100": cv2.aruco.DICT_6X6_100,
    "DICT_6X6_250": cv2.aruco.DICT_6X6_250,
    "DICT_6X6_1000": cv2.aruco.DICT_6X6_1000,
    "DICT_7X7_50": cv2.aruco.DICT_7X7_50,
    "DICT_7X7_100": cv2.aruco.DICT_7X7_100,
    "DICT_7X7_250": cv2.aruco.DICT_7X7_250,
    "DICT_7X7_1000": cv2.aruco.DICT_7X7_1000,
    "DICT_ARUCO_ORIGINAL": cv2.aruco.DICT_ARUCO_ORIGINAL,
```

```

    "DICT_APRILTAG_16h5": cv2.aruco.DICT_APRILTAG_16h5,
    "DICT_APRILTAG_25h9": cv2.aruco.DICT_APRILTAG_25h9,
    "DICT_APRILTAG_36h10": cv2.aruco.DICT_APRILTAG_36h10,
    "DICT_APRILTAG_36h11": cv2.aruco.DICT_APRILTAG_36h11
}

# Seleccionar el tipo de código ArUco y el ID del marcador
aruco_type = "DICT_6X6_1000"
id = 1

# Obtener el diccionario ArUco y configurar el detector
dictionary = cv2.aruco.getPredefinedDictionary(ARUCO_DICT[aruco_type])
detector_params = cv2.aruco.DetectorParameters()
detector = cv2.aruco.ArucoDetector(dictionary, detector_params)

# Parámetros del controlador PID para el eje X
kp_x = 0.1
ki_x = 0.01
kd_x = 0.1

# Inicializar el controlador PID para el eje X
pid_controller_x = PID(kp_x, ki_x, kd_x, setpoint=0)

# Iniciar la captura de video desde la cámara
cap = cv2.VideoCapture(0)

posicion_rel_camaraguide_guia = None

while True:

    # Leer un fotograma desde la cámara
    ret, frame = cap.read()
    if not ret:
        break

    # Dibujar un círculo en el centro de la cámara
    center_x = frame.shape[1] // 2
    center_y = frame.shape[0] // 2
    cv2.circle(frame, (center_x, center_y), 10, (255, 0, 0), -1)

    # Convertir el fotograma a escala de grises
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Detectar los marcadores ArUco en el fotograma
    markerCorners, markerIds, rejectedCandidates =
detector.detectMarkers(frame)

    # Verificar si se detectaron marcadores y dibujarlos en la imagen
    if markerIds is not None and len(markerIds) > 0:

```

```

ids_en_fotograma = [id[0] for id in markerIds]

# Inicializar variables para el seguimiento de los marcadores en
el fotograma actual
tamano_marcadores_actual = float('inf')
marcador_guia_actual = None

for i, marcador_id in enumerate(ids_en_fotograma):
    # Calcular el área del marcador (usamos el área como medida
de tamaño)
    area = cv2.contourArea(markerCorners[i][0])

    # Actualizar el marcador guía actual si encontramos uno más
pequeño
    if area < tamano_marcadores_actual:
        tamano_marcadores_actual = area
        marcador_guia_actual = marcador_id
        # Calcular la posición relativa del centro del marcador
guía al centro de la cámara
        if marcador_guia_actual is not None and
marcador_guia_actual in markerIds:
            indice_guia = np.where(markerIds ==
marcador_guia_actual)[0][0]
            centro_guia =
np.intp(np.mean(markerCorners[indice_guia], axis=1)[0])
            centro_camara = (frame.shape[1] // 2, frame.shape[0]
// 2)
            posicion_rel_camaraguide_guia = (centro_guia[0] -
centro_camara[0], centro_guia[1] - centro_camara[1])
            # Obtener el tamaño del texto
            text_size = cv2.getTextSize(f"ID: {markerIds[i]},
Relative Position: {posicion_rel_camaraguide_guia}",
cv2.FONT_HERSHEY_SIMPLEX, 0.5, 2)[0]
            # Mostrar la información como texto en el centro del
marcador
            x_centro_guia, y_centro_guia = centro_guia

            # Calcular las diferencias en las coordenadas X e Y
de las dos esquinas continuas
            x_diff = markerCorners[i][0][1][0] -
markerCorners[i][0][0][0]
            y_diff = markerCorners[i][0][1][1] -
markerCorners[i][0][0][1]

            # Calcular el ángulo de rotación basado en las
diferencias de coordenadas
            angle_of_rotation = np.degrees(np.arctan2(y_diff,
x_diff))

```

```

        if marcador_id == marcador_guia_actual:
            cv2.putText(frame, f"ID: {markerIds[i]}, Angle:
{angle_of_rotation}, Relative Position: {posicion_rel_camaraguide_guia}",
(x_centro_guia, y_centro_guia), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255,
255), 2)

            print(f"Marcador guía ID: {marcador_guia_actual},
Posición relativa: {posicion_rel_camaraguide_guia}")

            # Calcular la salida del controlador PID para el eje
X

            output_x =
pid_controller_x(posicion_rel_camaraguide_guia[0])

            # Ajusta el valor de la salida según tus necesidades
            # Puedes utilizar otros parámetros de comandos según
la documentación de DroneKit
            vehicle.channels.overrides['RCIN_C2'] = int(1500 +
output_x * 5)

            # Mostrar información de depuración
            print(f"PID Output X: {output_x}")

    cv2.aruco.drawDetectedMarkers(frame, markerCorners, markerIds)

# Mostrar la imagen con los marcadores en una ventana
cv2.imshow("frame", frame)

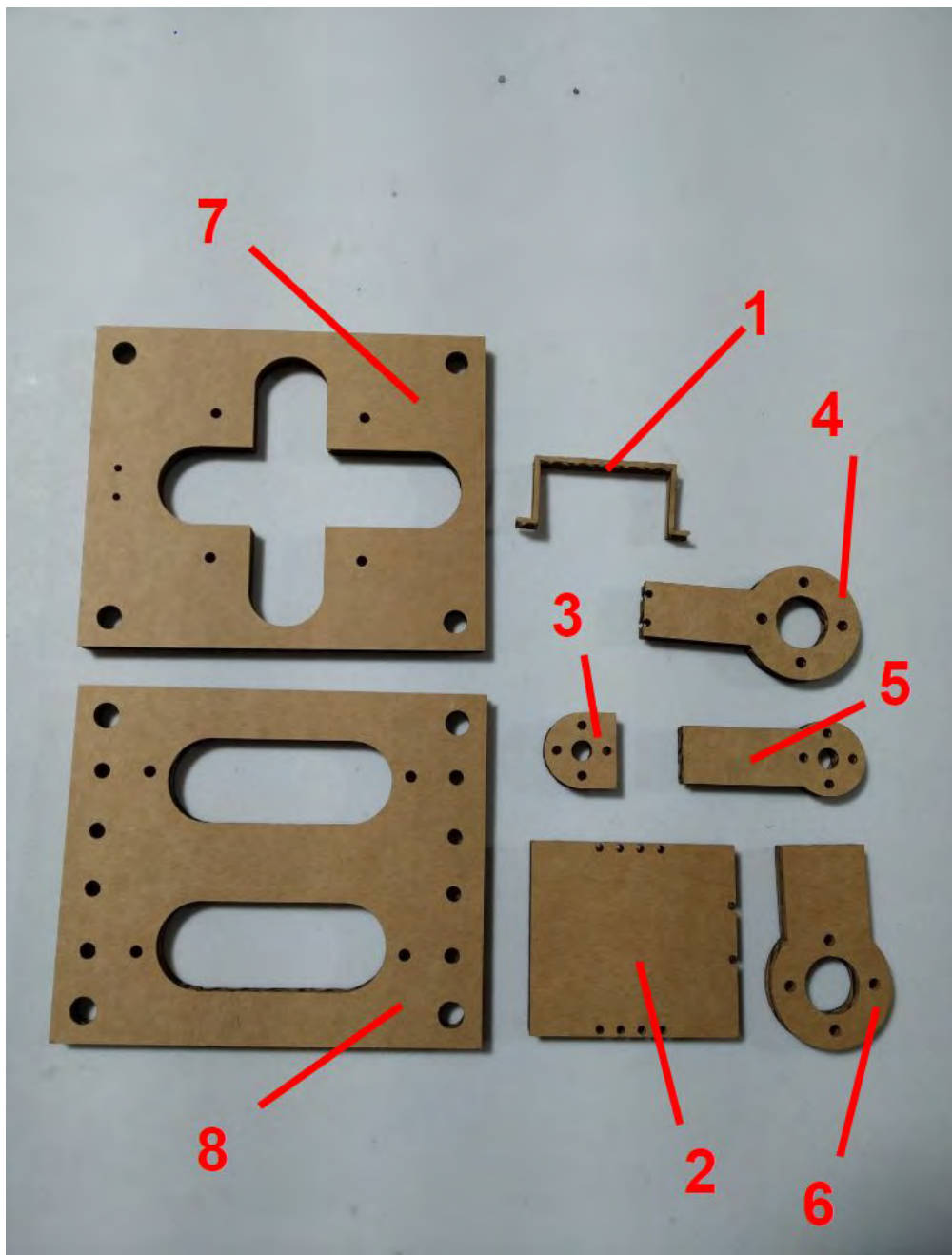
# Salir del bucle si se presiona la tecla 'q'
key = cv2.waitKey(1)
if key == ord('q'):
    break

# Liberar los recursos y cerrar las ventanas
cap.release()
cv2.destroyAllWindows()

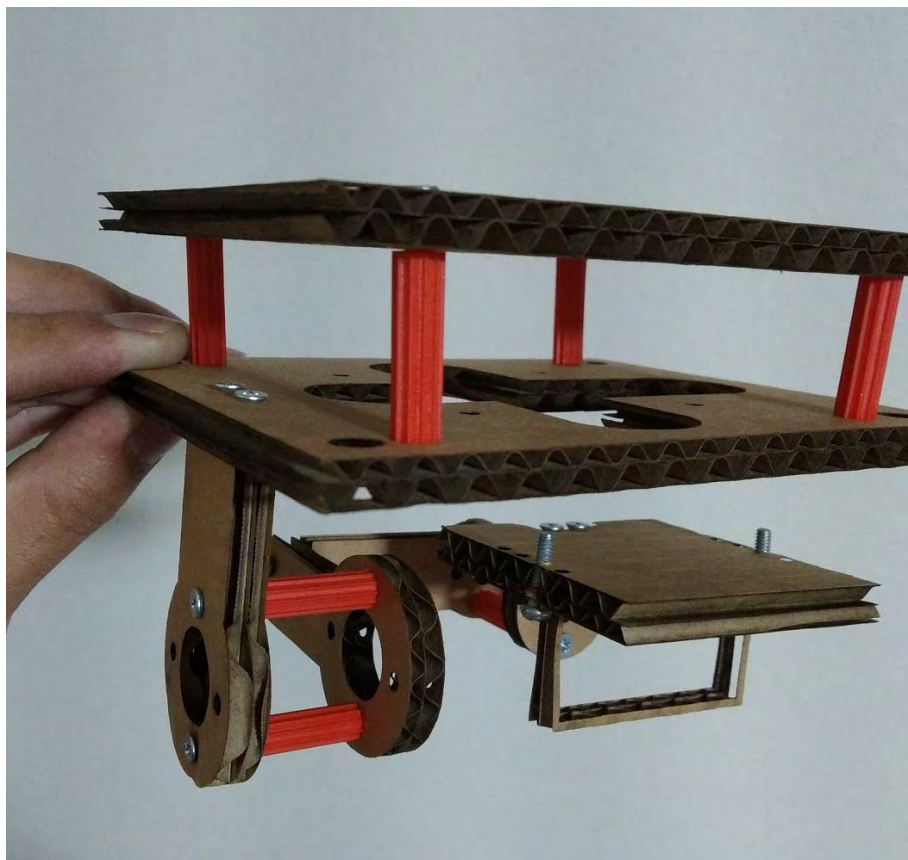
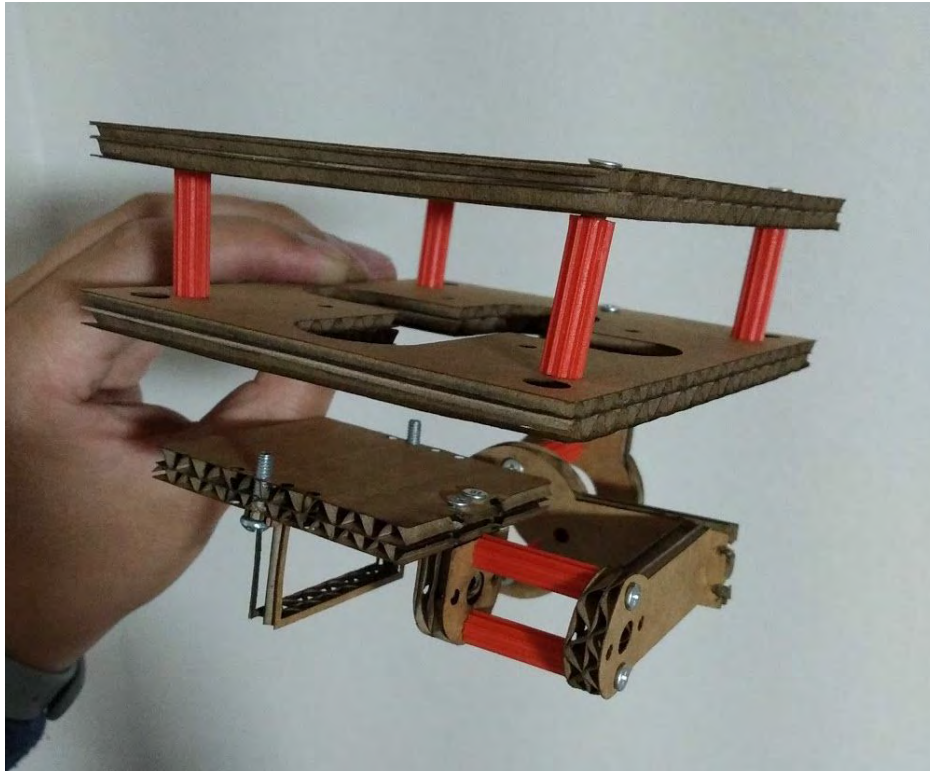
```

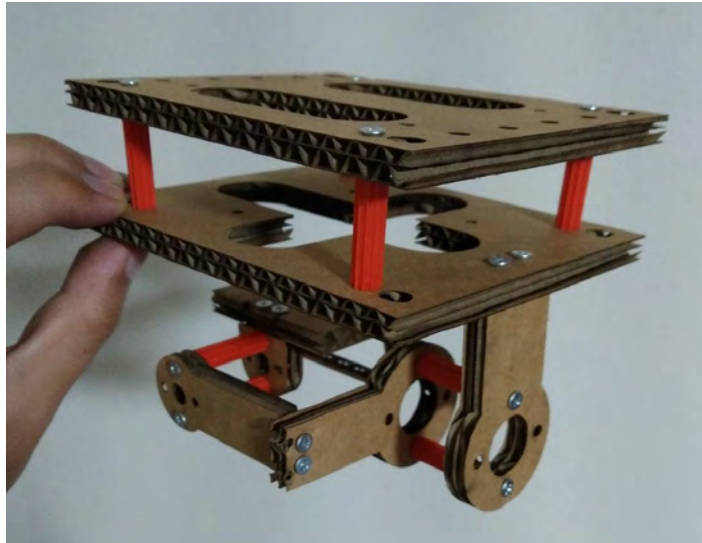
Anexo F: Desarrollo de una maqueta de la estructura del estabilizador de la cámara

La estructura del estabilizador de la cámara consta de 8 piezas, las cuales se cortan con láser para obtener una mayor precisión de las medidas establecidas en los planos de despiece. Las piezas ya cortadas se muestran en la siguiente imagen y la numeración representa el orden recomendado a seguir para armar la estructura, empezando por la pieza N°1 y terminando con la pieza N°8.



A continuación se presentan diferentes vistas de la estructura ya ensamblada.





Como se ve en las anteriores imágenes, todas las uniones son atornilladas, y se utilizaron separadores de plástico (piezas color naranja) para indicar los lugares en los que se colocará los motores y las almohadillas antivibraciones. La estructura efectivamente cumple con el requisito de tener medidas inferiores a 15 mm de alto x 12 mm de ancho y 12 mm de largo, además que se comprobó que es posible realizar el ensamblaje utilizando solamente uniones atornilladas y que es posible hacerlo en un tiempo inferior a 15 minutos, como se indicó en la lista de requerimientos.

Anexo G: Costos

Motor T-MOTOR GB36-1

The screenshot shows the product page for the T-MOTOR GB36-1 Gimbal Type 3-6S UAV Drone Motor KV50. The main image is a black cylindrical motor with 'GB' and 'T-MOTOR' branding. Below it are smaller images of the motor's components. The price is listed as \$49.90. The page includes a navigation menu with categories like UAV Power, VTOL/Fixed Wing Power, Combo Pack, Fixed Wing Model, FPV Model, Robot Dynamics, and T-DRONES. A shopping cart icon is visible in the top right. A chat bubble is present in the bottom right corner.

GB36-1 Gimbal Type 3-6S UAV Drone Motor KV50

\$49.90

Available points: 2500 Points | Points reward: 49 Points

- Gimbal system module
- Surveillance camera motor
- Small Cogging Torque

1 [Add to cart](#) [Buy now](#)

USD200 Consumption Free Shipping by DHL Express (Excl. Remote Areas and Special Items).

Chat with us

DRV8313 Driver e IMU

The screenshot shows the product page for the SimpleBGC 32bit Regular I2C 1-IMU Set, Revision B on the BASECAM website. The page features a navigation menu with links for About us, Products, Store, Software, Manuals, Support, and Forum. A shopping cart icon shows 2 items for a total of €408.98 (incl. VAT). A green notification bar at the top states that the product has been added to the cart. The main product image is a blue printed circuit board (PCB) with various components. The price is €204.49. A list of categories is provided on the right, including New and Discounted Products, Controllers, Brushless Motors, Pro Controller and Components, Components, Cables, and Gimbals. The 'The kit includes:' section lists the contents: SimpleBGC 32bit Regular main board (1 pc), Motor 30 cm wires (3 pcs), and Battery 20 cm wire (1 pc).

BASECAM

About us Products Store Software Manuals Support Forum

EN CN

Shipping and payments Warranty Checkout Cart My Account €408.98 (incl. VAT) 2 items

Home / Controllers / SimpleBGC 32bit Regular I2C 1-IMU Set, Revision B

"SimpleBGC 32bit Regular I2C 1-IMU Set, Revision B" has been added to your cart. [View cart](#)

SimpleBGC 32bit Regular I2C 1-IMU Set, Revision B

€204.49


The kit includes:

- SimpleBGC 32bit Regular main board – 1 pc
- Motor 30 cm wires – 3 pcs
- Battery 20 cm wire – 1 pc

Categories

- New and Discounted Products
- Controllers
- Brushless Motors
- Pro Controller and Components
- Components
- Cables
- Gimbals

OpenMV Cam RT1062



OpenMV Cam RT1062

\$120.00 USD

Shipping calculated at checkout.

Pay in 4 interest-free installments of \$30.00 with [shop Pay](#) [Learn more](#)

- This product is available

− 1 +

Add to cart

Buy with [shop Pay](#)

More payment options

Chat

penMV Search

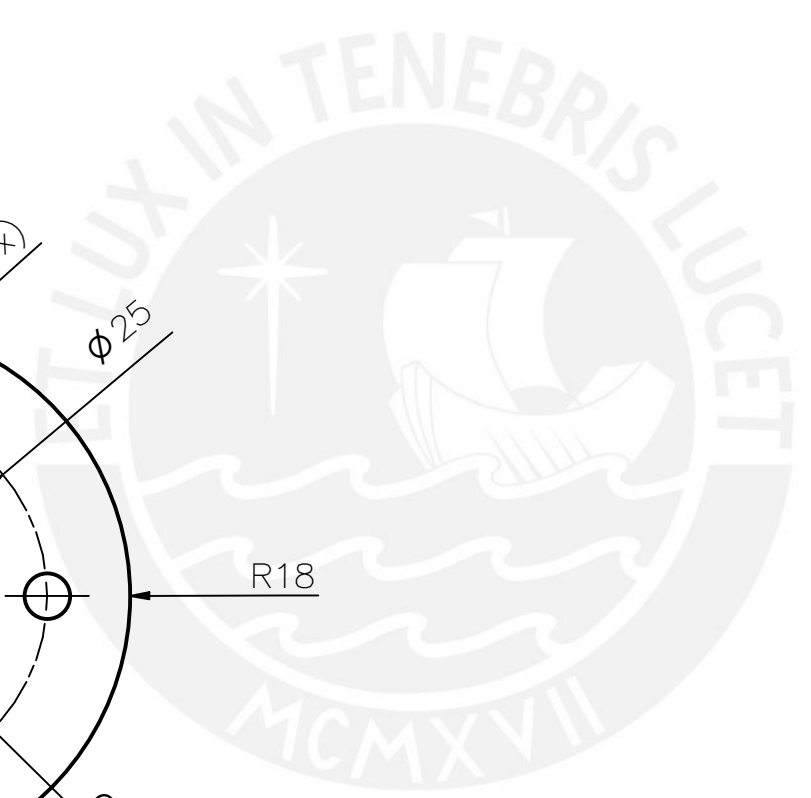
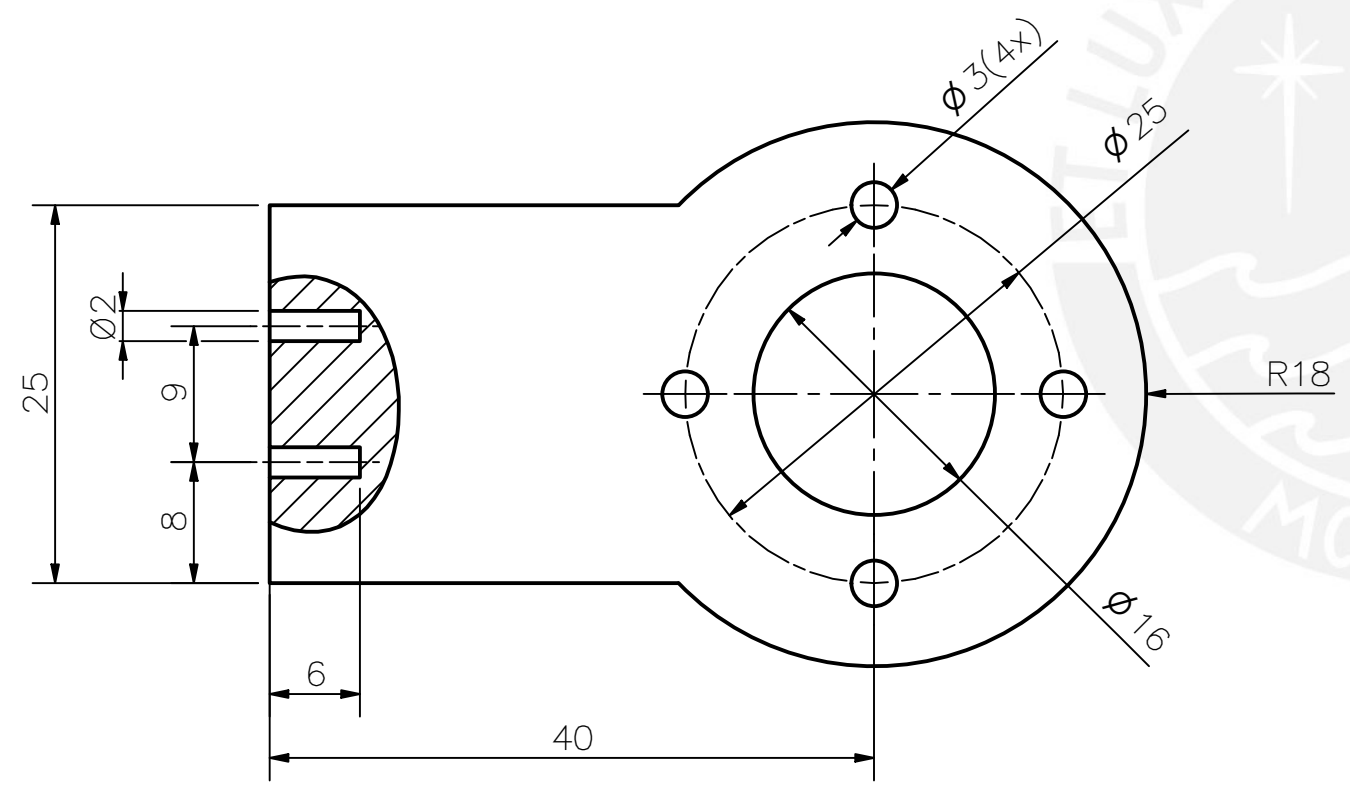
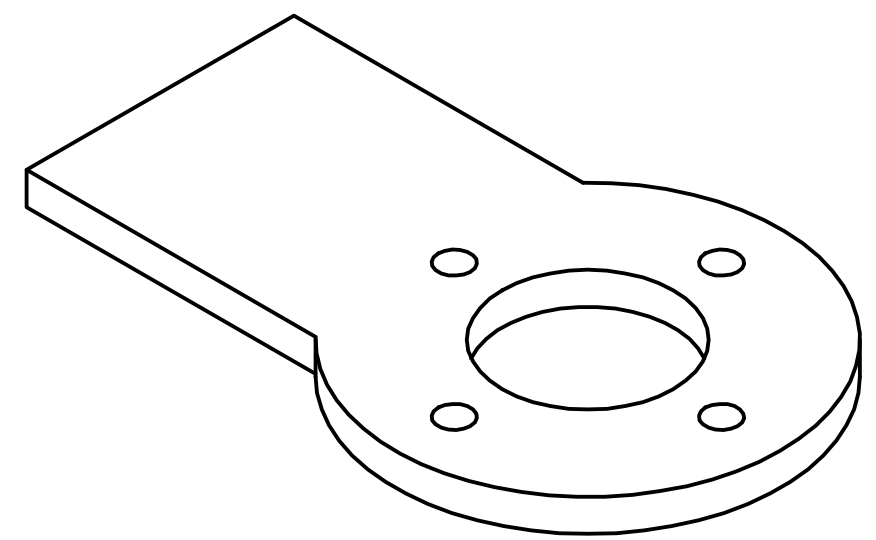
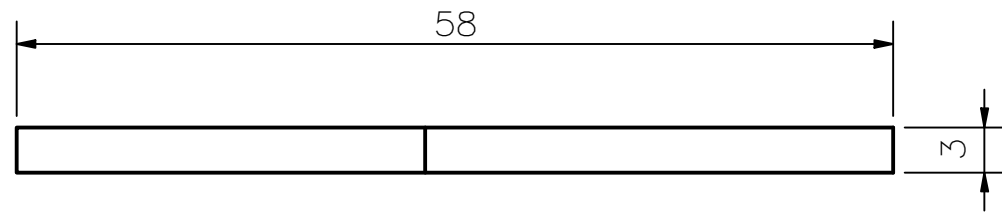
Home Products Applications Download Docs Forums About Blog GitHub

\$240.00 USD ³ Login

ET LU UCET MCMXVII

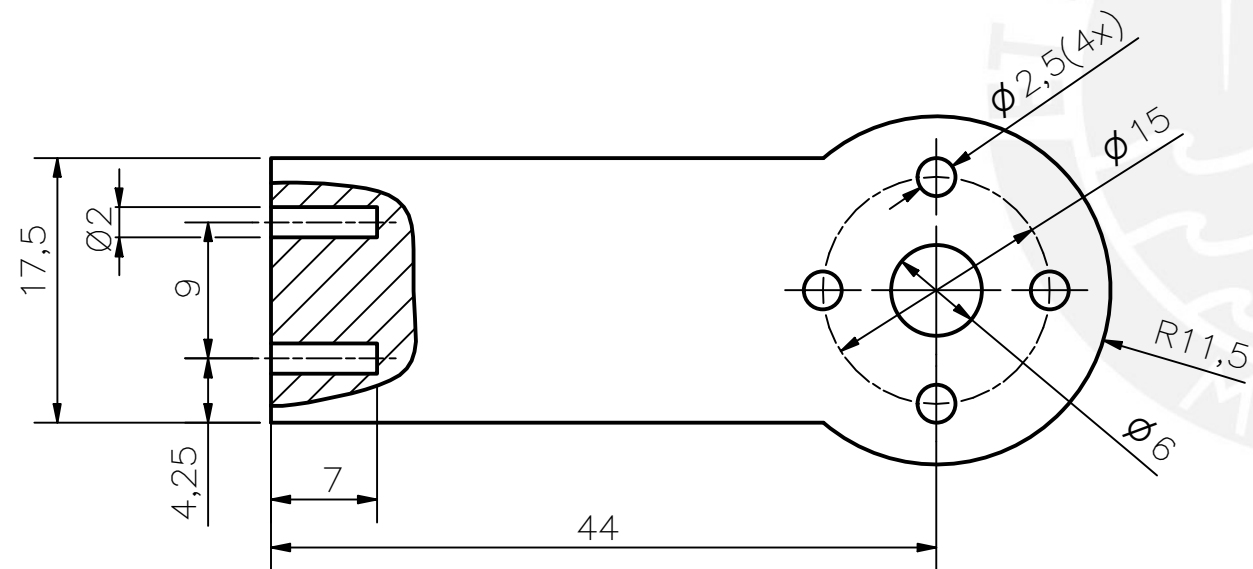
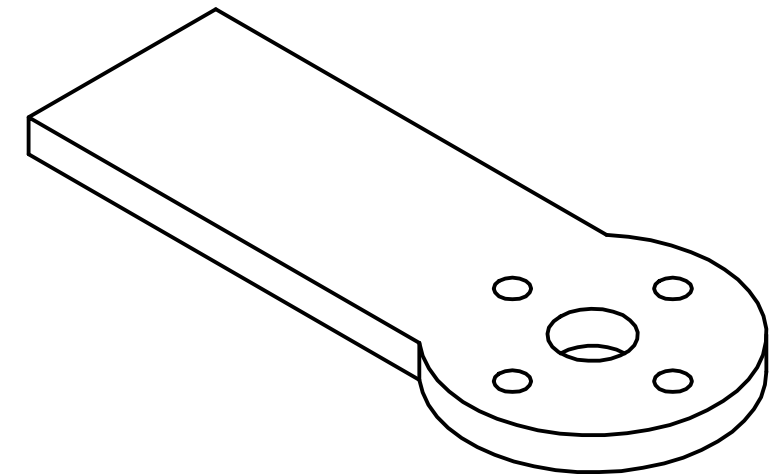
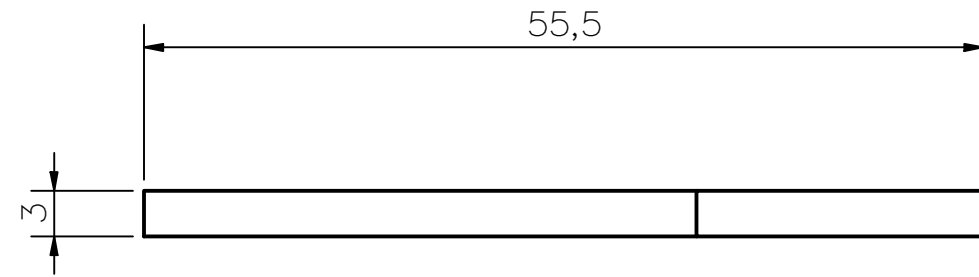
Anexo H: Planos

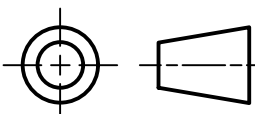




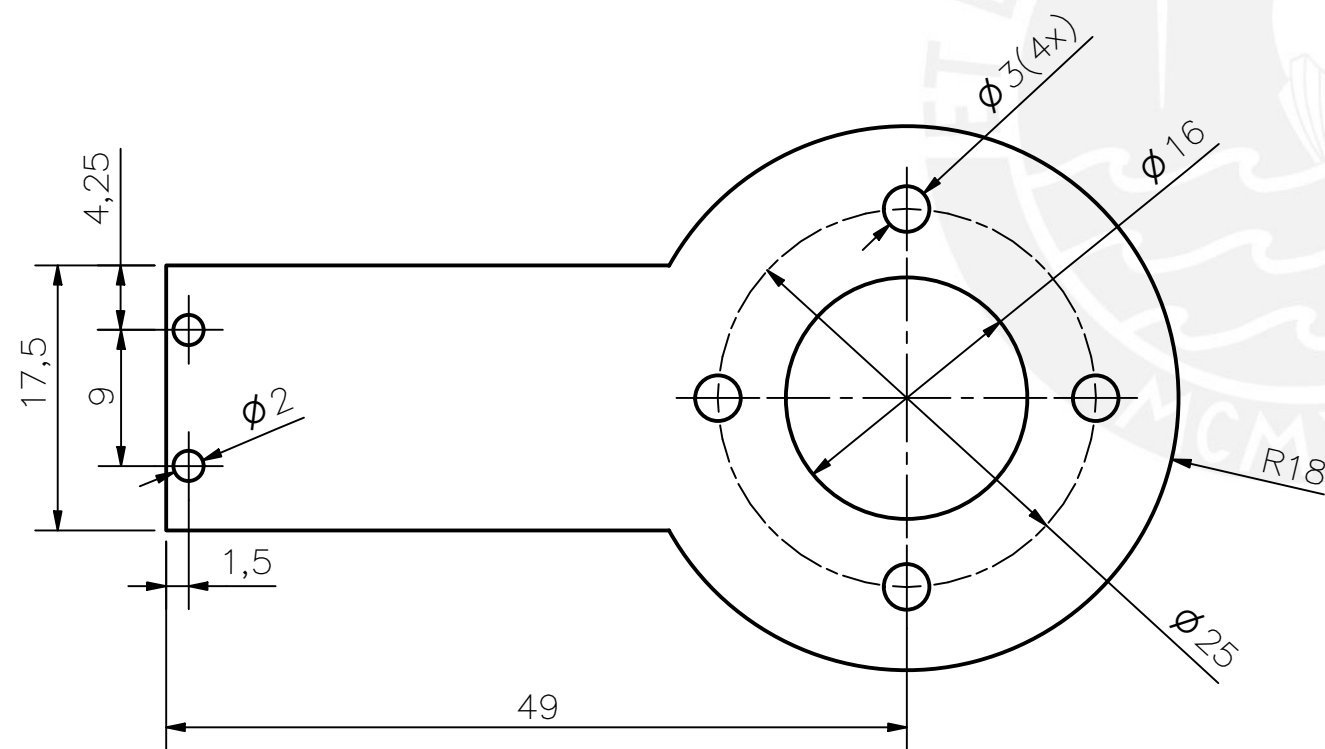
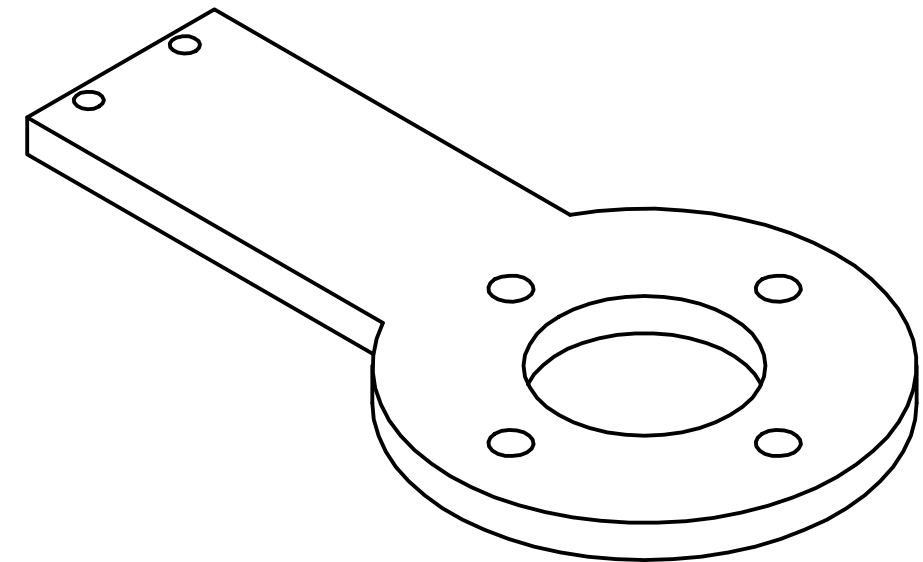
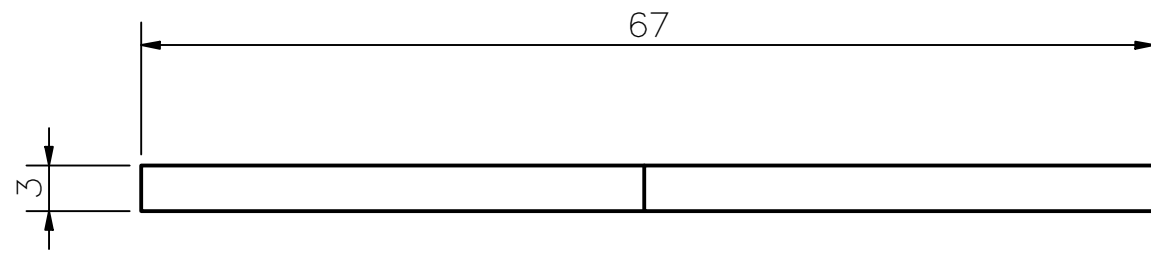
ACABADO SUPERFICIAL	TOLERANCIA GENERAL	MATERIAL
		al-6061
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ CIENCIAS E INGENIERÍA-ING. MECATRÓNICA		
MÉTODO DE PROYECCIÓN	1MTR02	ESCALA
	Brazo N°1	2:1
20150561	SALAZAR AYLLON, PAOLO BRYAN	FECHA: 2019.09.02
		LÁMINA: A4

COTA NOMINAL	COTA MÁXIMA	COTA MÍNIMA



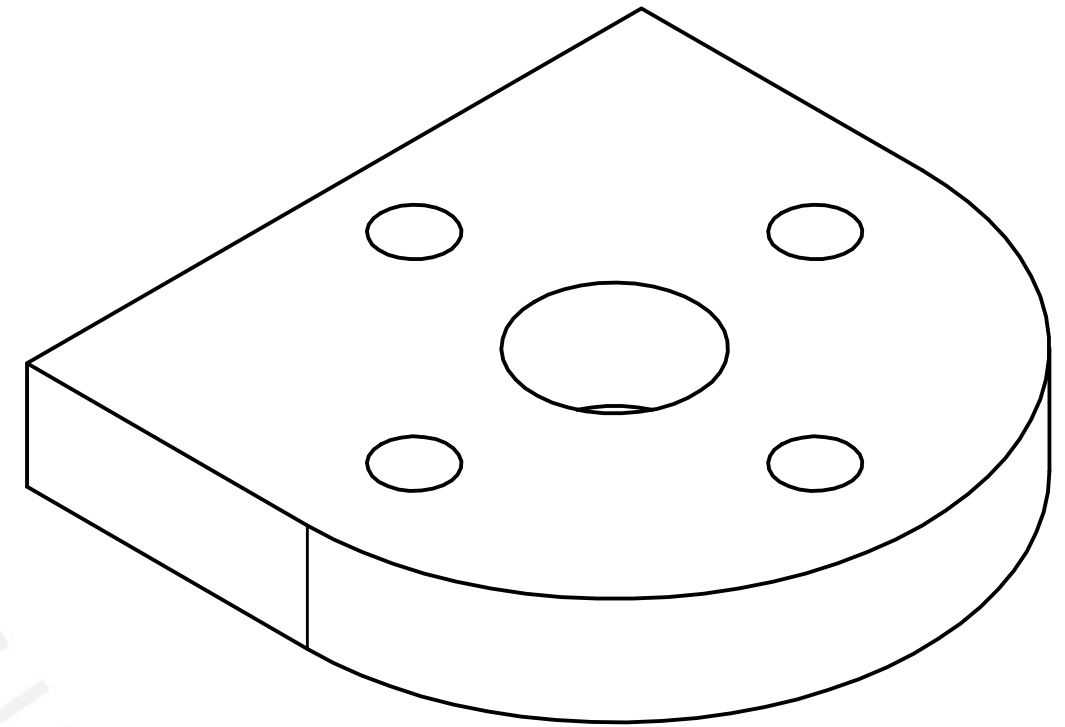
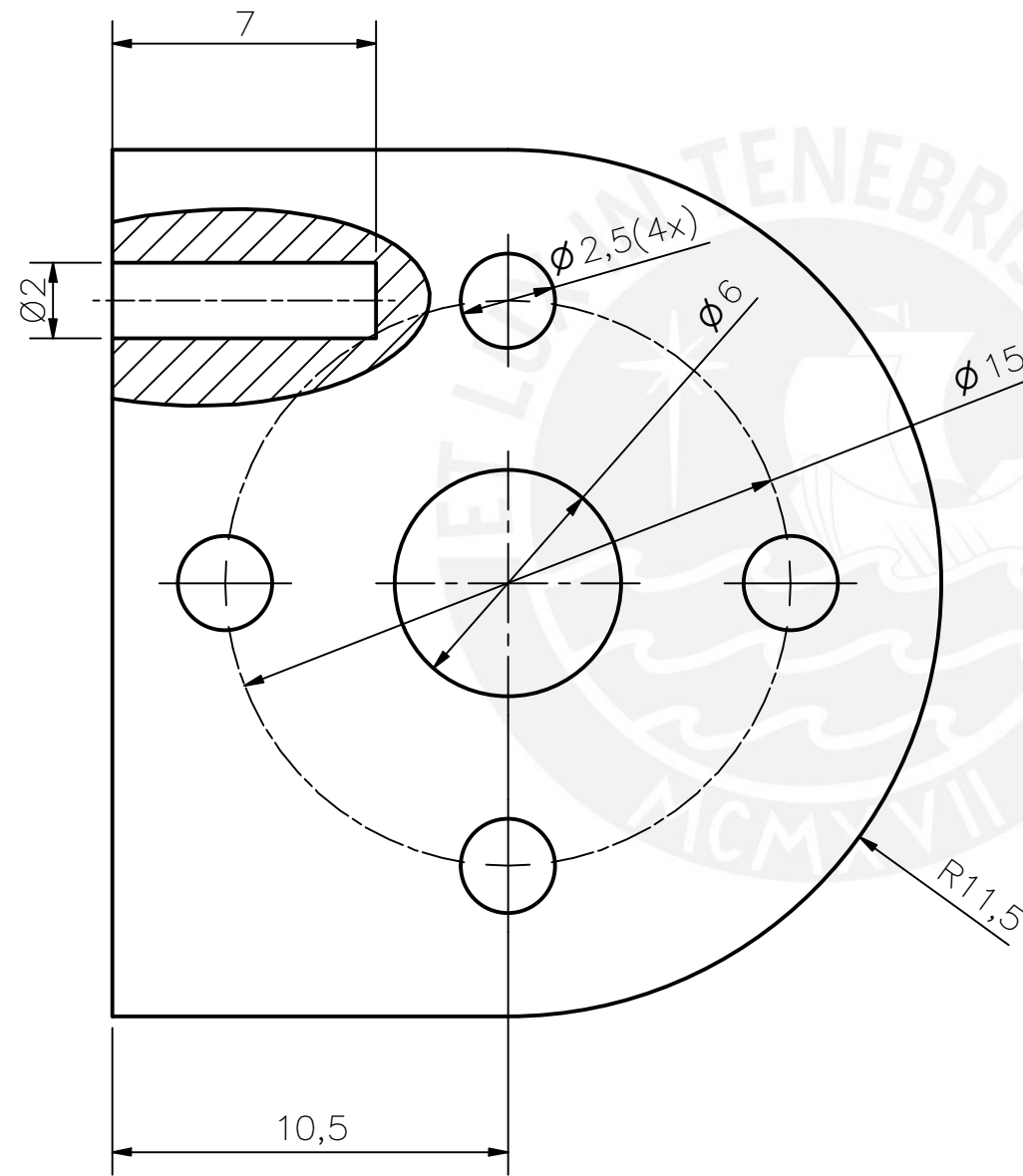
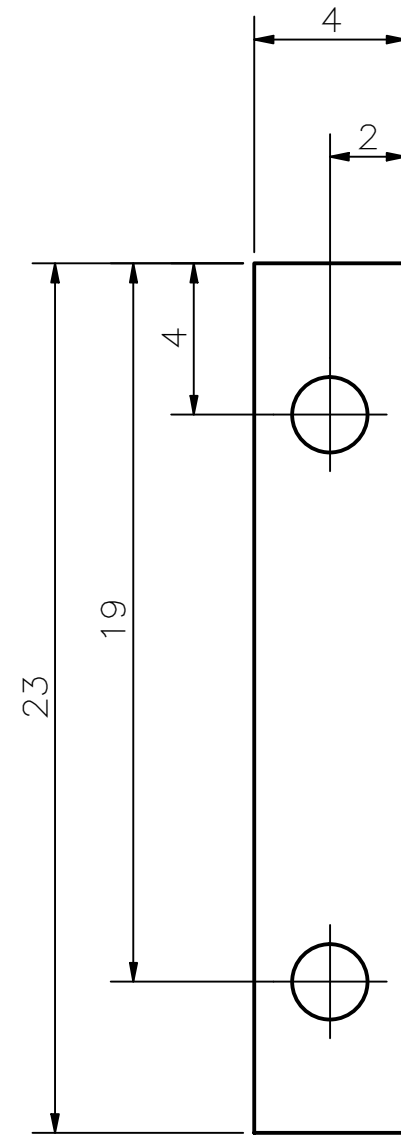
ACABADO SUPERFICIAL	TOLERANCIA GENERAL	MATERIAL al-6061
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ CIENCIAS E INGENIERÍA-ING. MECATRÓNICA		
MÉTODO DE PROYECCIÓN 	1MTR02 BRAZO N°2	ESCALA 2:1
20150561	SALAZAR AYLLÓN, PAOLO BYAN	FECHA: 2019.09.02
		LÁMINA: A4

COTA NOMINAL	COTA MÁXIMA	COTA MÍNIMA
--------------	-------------	-------------



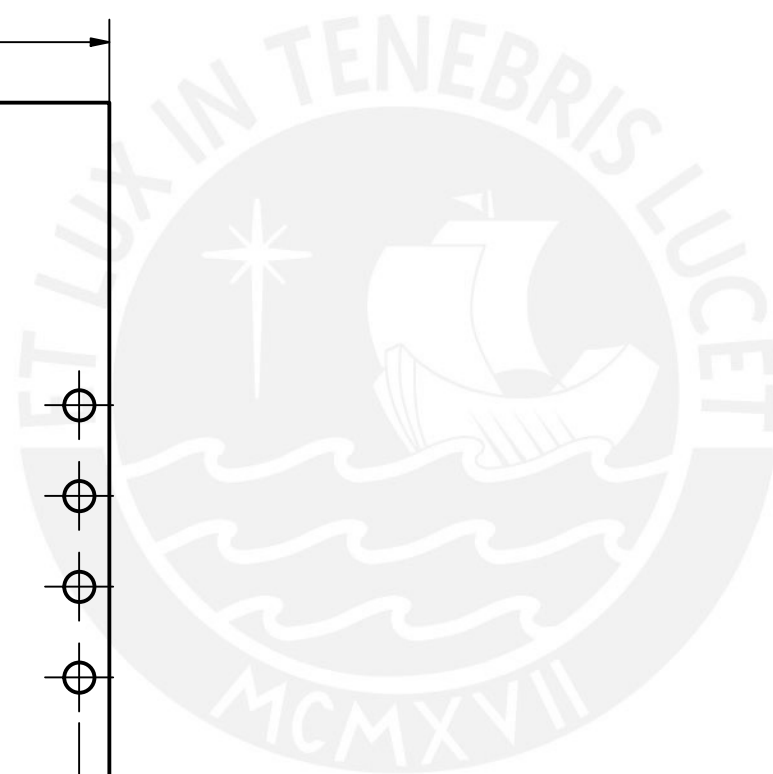
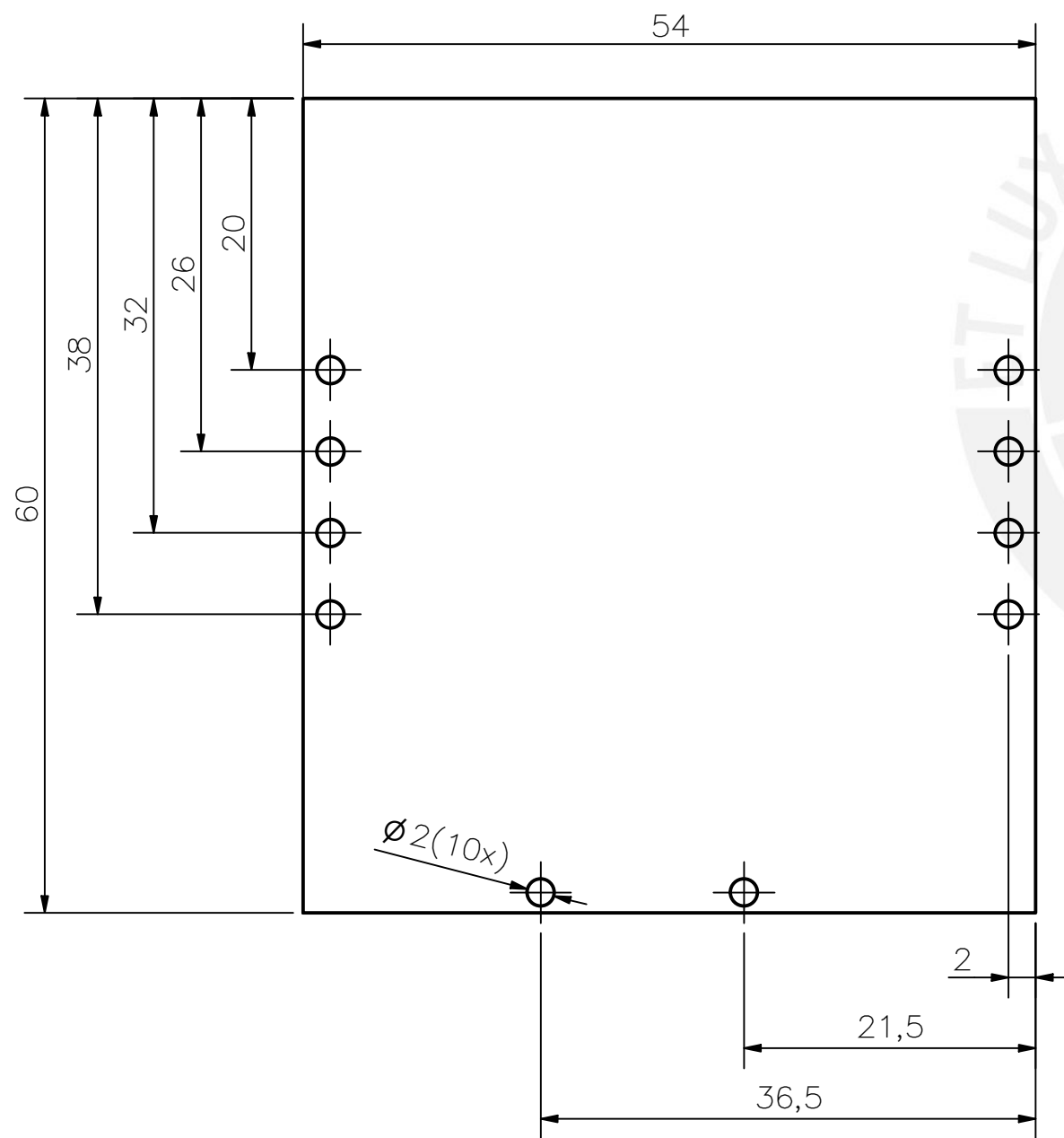
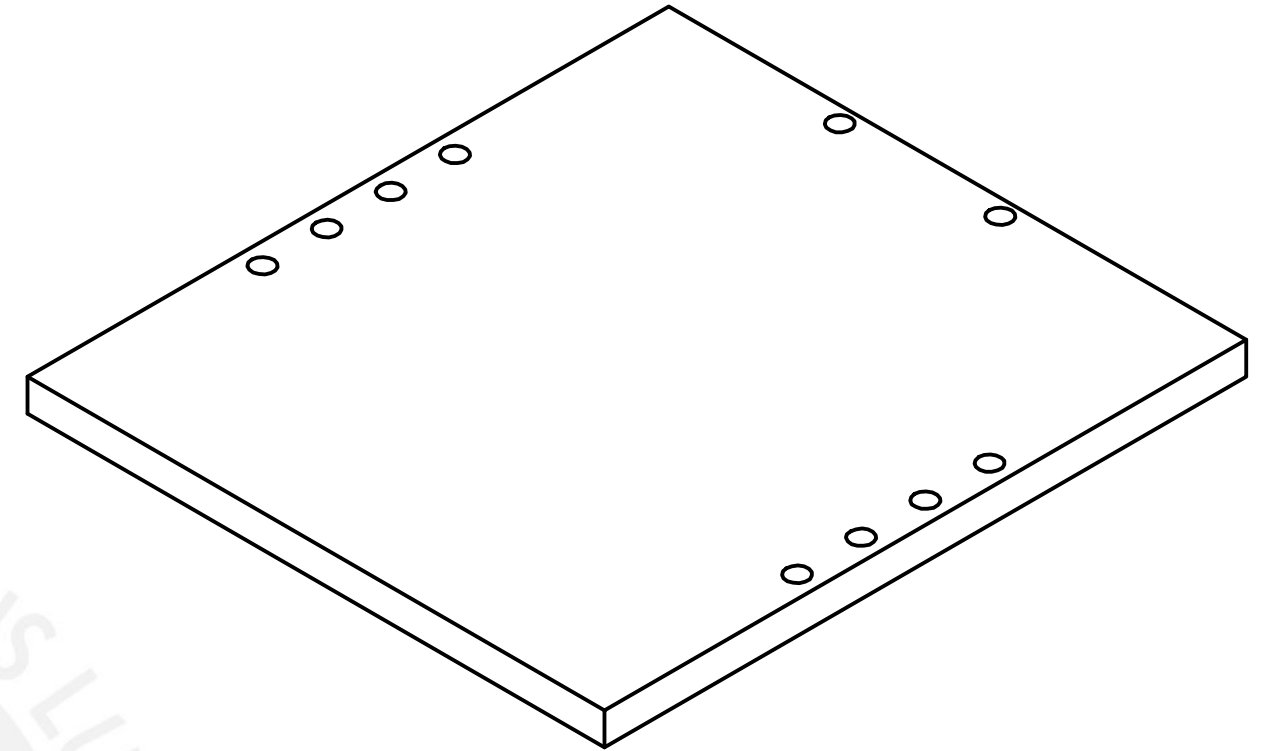
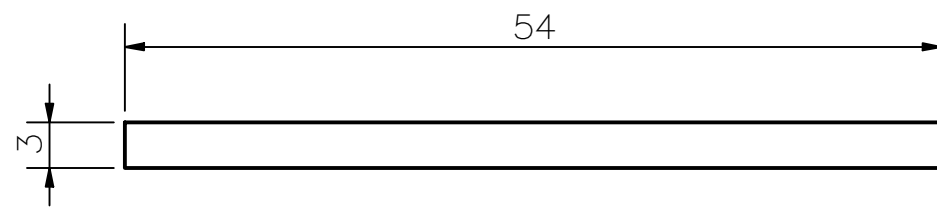
ACABADO SUPERFICIAL	TOLERANCIA GENERAL	MATERIAL
		al-6061
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ CIENCIAS E INGENIERÍA-ING. MECATRÓNICA		
MÉTODO DE PROYECCIÓN	1MTR02	ESCALA
	BRAZO N°3	2:1
20150561	SALAZAR AYLLÓN, PAOLO BRYAN	FECHA: 2019.09.02
		LÁMINA: A4

COTA NOMINAL	COTA MÁXIMA	COTA MÍNIMA



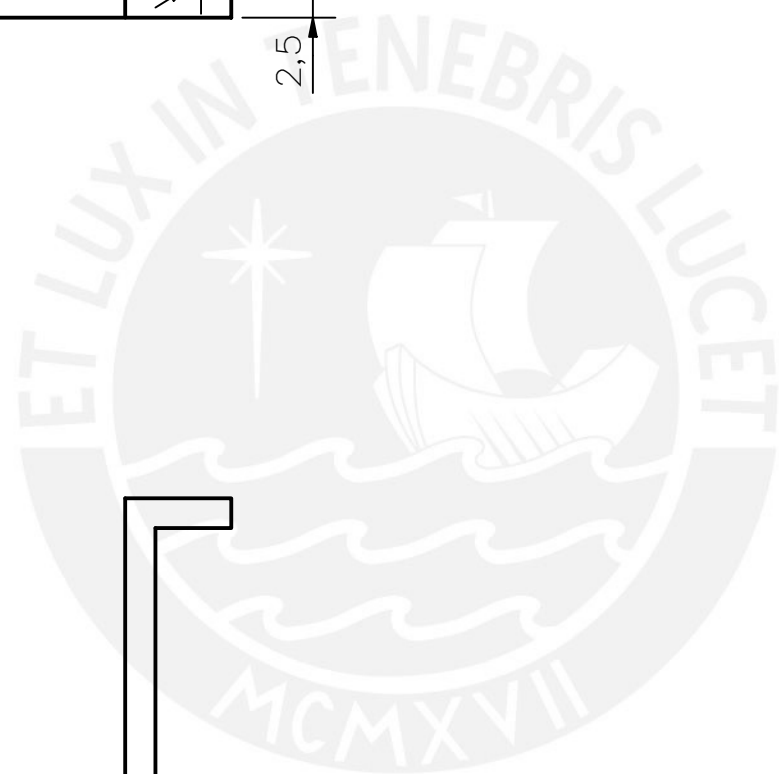
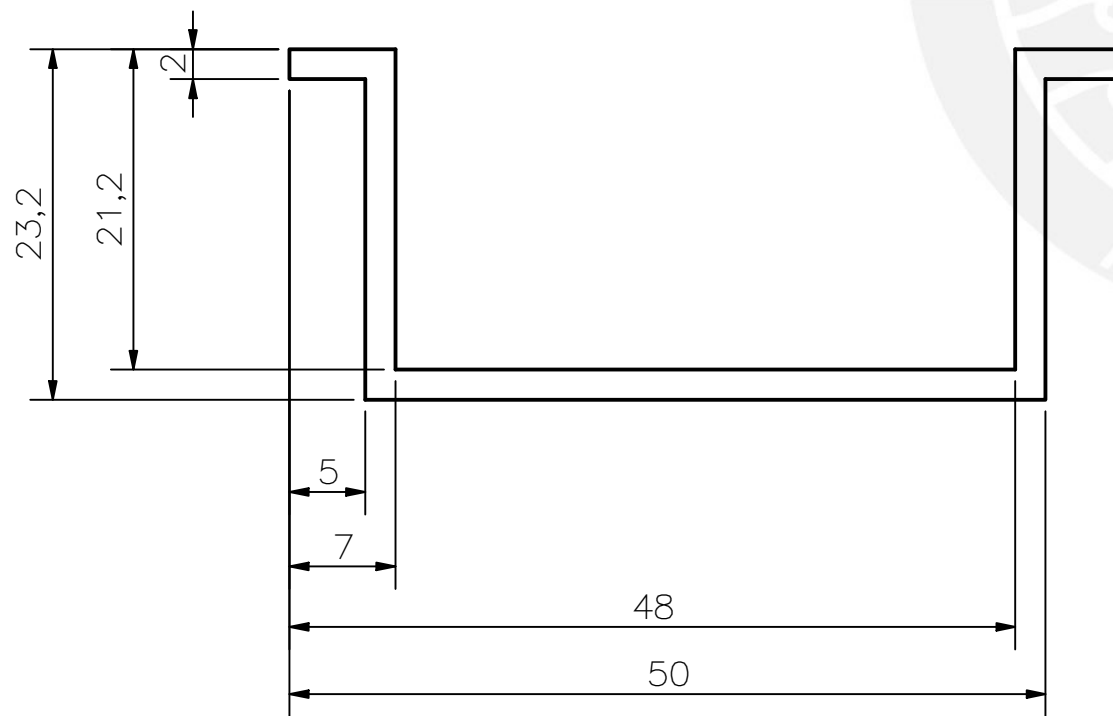
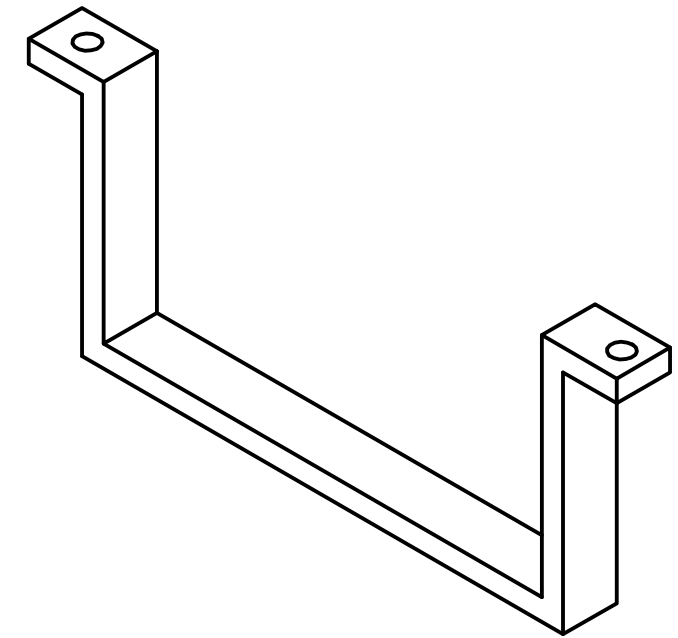
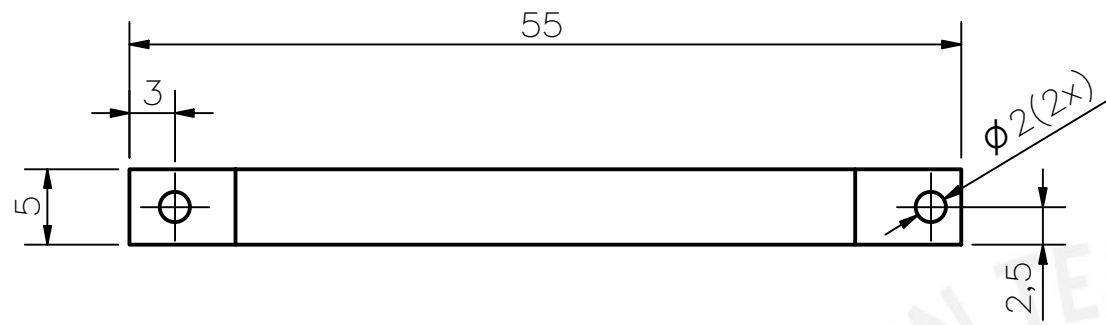
ACABADO SUPERFICIAL	TOLERANCIA GENERAL	MATERIAL
		al-6061
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ CIENCIAS E INGENIERÍA-ING. MECATRÓNICA		
MÉTODO DE PROYECCIÓN	1MTR02	ESCALA
	BRAZO N°4	5:1
20150561	SALAZAR AYLLÓN, PAOLO BRYAN	FECHA: 2019.09.02
		LÁMINA: A4

COTA NOMINAL	COTA MÁXIMA	COTA MÍNIMA

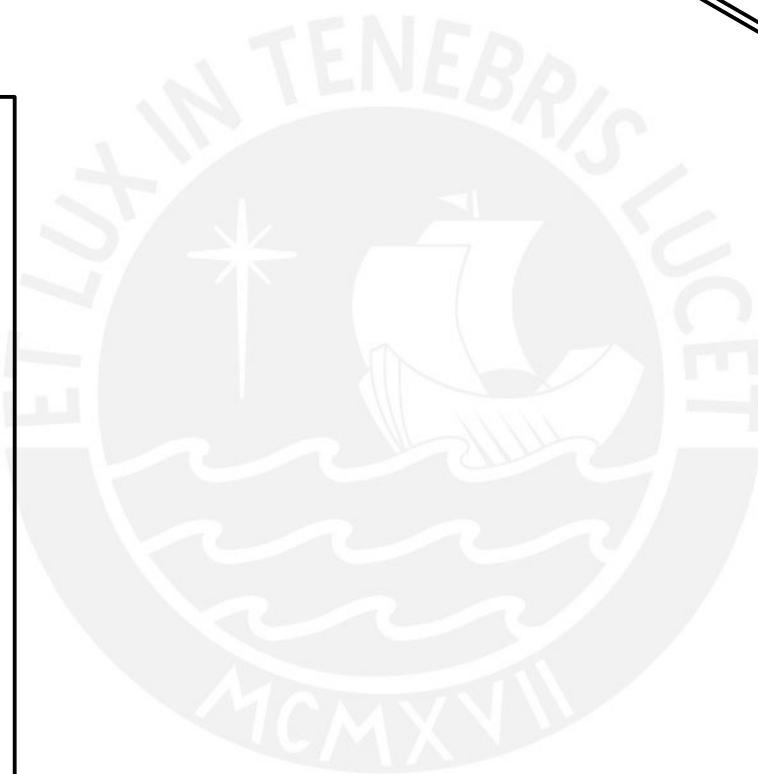
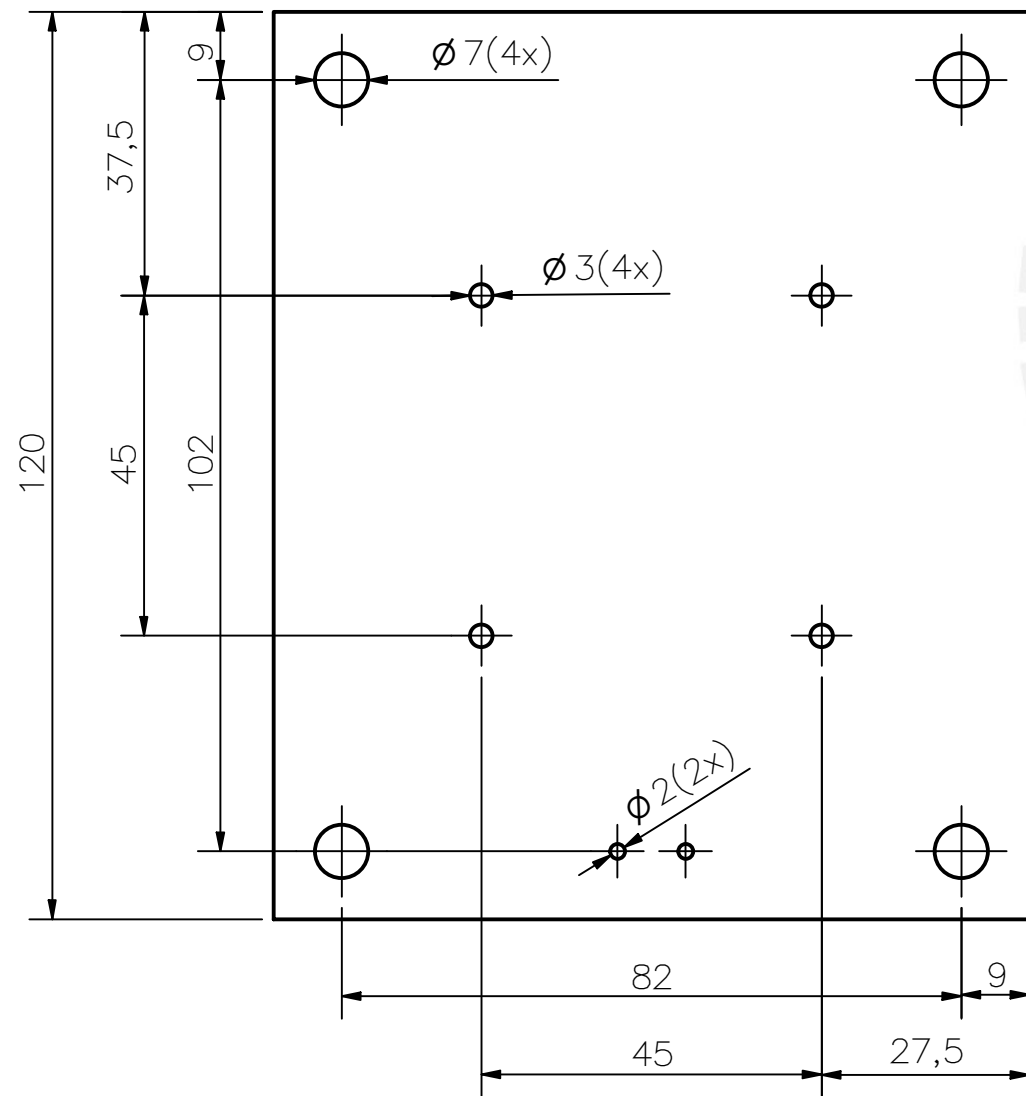
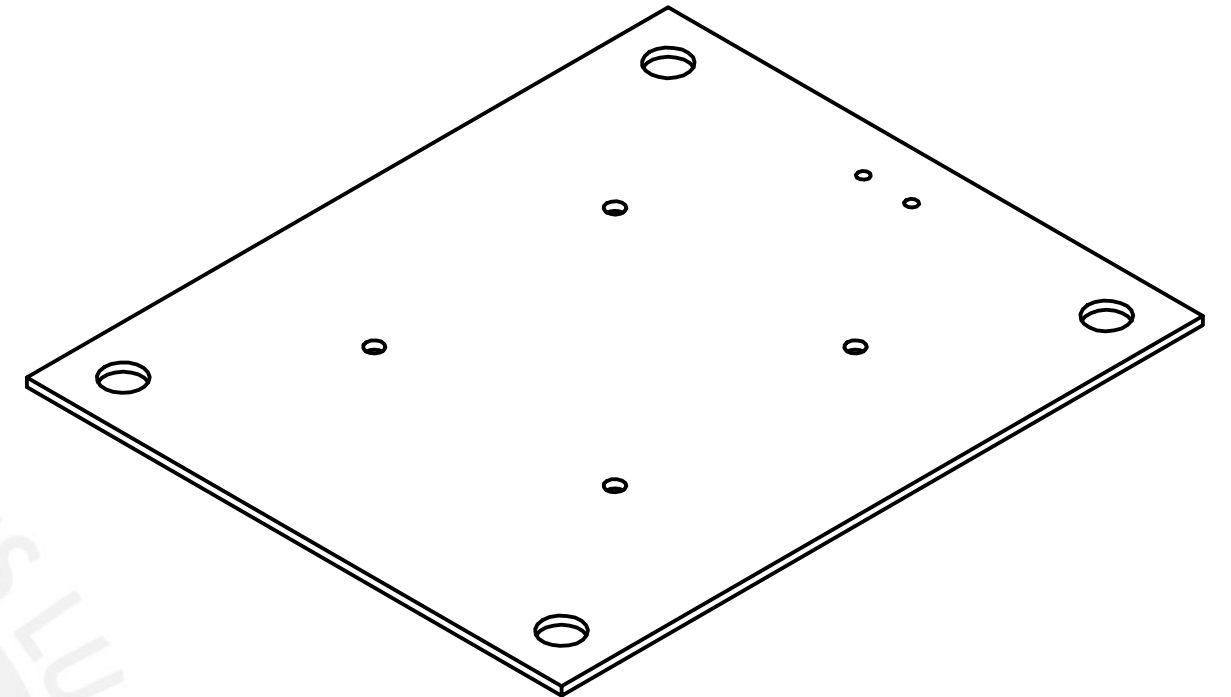
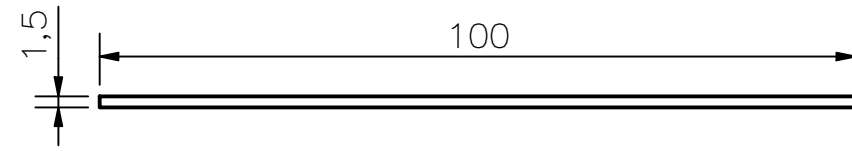


ACABADO SUPERFICIAL	TOLERANCIA GENERAL	MATERIAL
		al-6061
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ CIENCIAS E INGENIERÍA.ING. MECATRÓNICA		
MÉTODO DE PROYECCIÓN	1MTR02	ESCALA
	SOPORTE DE CÁMARA	2:1
20150561	SALAZAR AYLLÓN, PAOLO	FECHA: 2019.09.08
		LÁMINA: A4

COTA NOMINAL	COTA MÁXIMA	COTA MÍNIMA

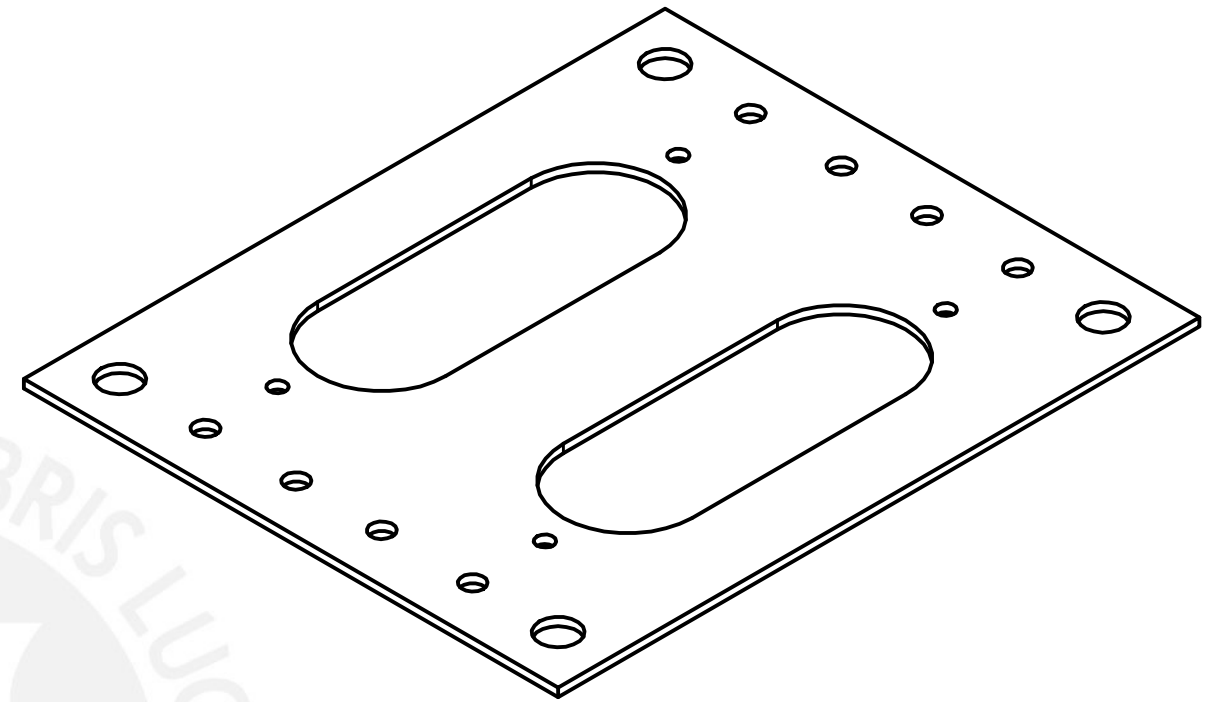
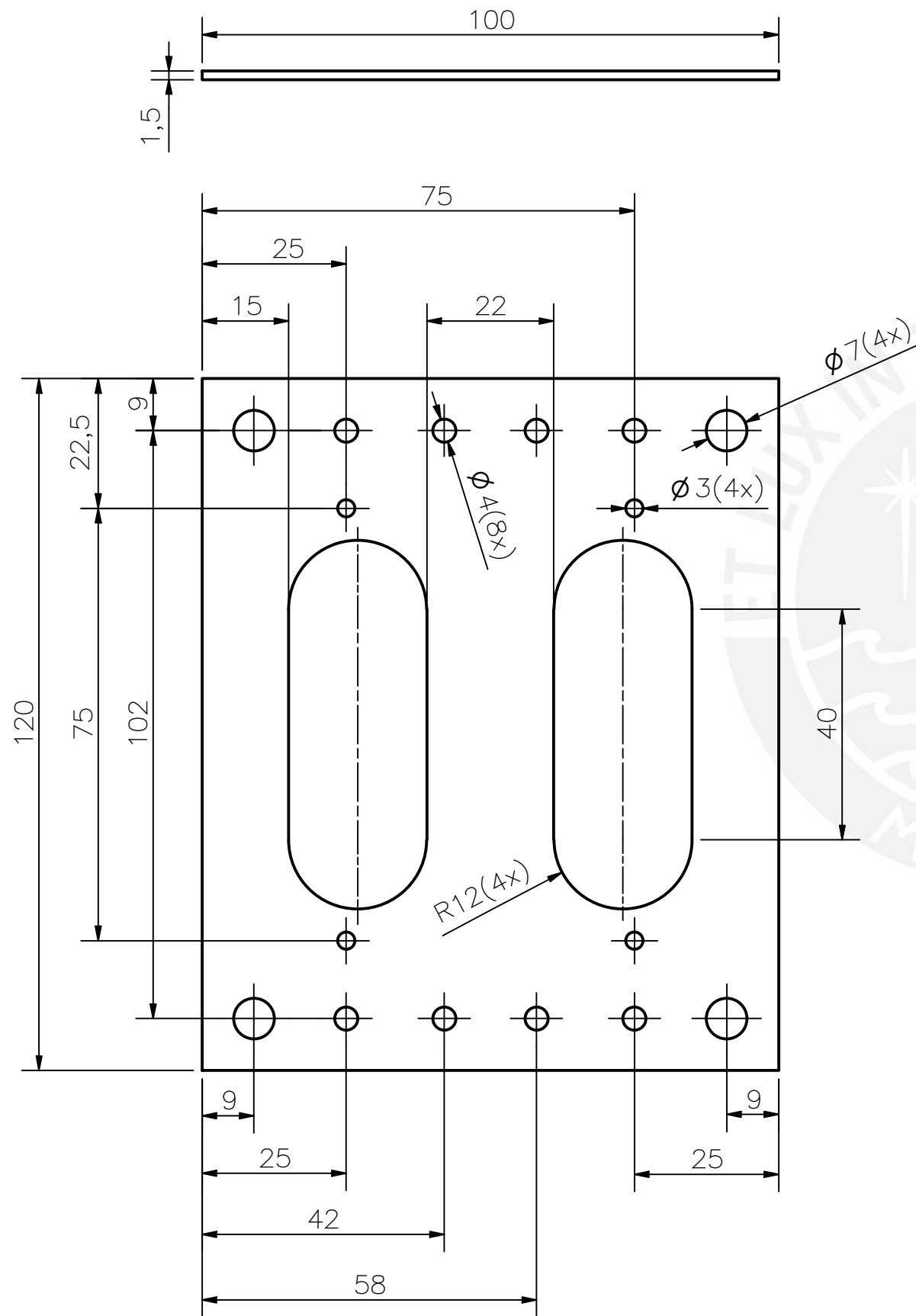


ACABADO SUPERFICIAL	TOLERANCIA GENERAL	MATERIAL
		al-6061
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ CIENCIAS E INGENIERÍA-ING. MECATRÓNICA		
MÉTODO DE PROYECCIÓN	1MTR02	ESCALA
	SUJETADOR DE CAMARA	2:1
20150561	SALAZAR AYLLÓN, PAOLO	FECHA: 2019.09.08
COTA NOMINAL	COTA MÁXIMA	COTA MÍNIMA
		LÁMINA: A4



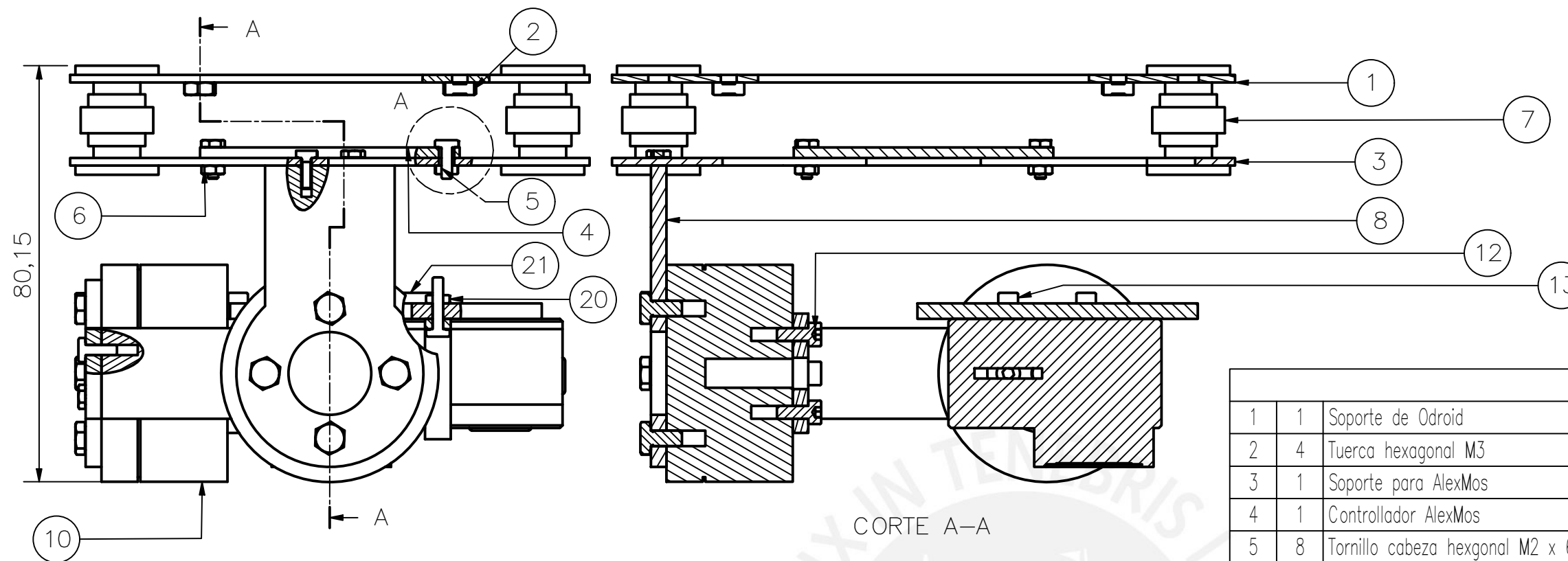
ACABADO SUPERFICIAL	TOLERANCIA GENERAL	MATERIAL
		al-6061
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ CIENCIAS E INGENIERÍA-ING. MECATRÓNICA		
MÉTODO DE PROYECCIÓN	1MTR02	ESCALA
	SOPORTE DE ALEXMOS	1:1
20150561	SALAZAR AYLLÓN, PAOLO	FECHA: 2019.09.08
		LÁMINA: A4

COTA NOMINAL	COTA MÁXIMA	COTA MÍNIMA

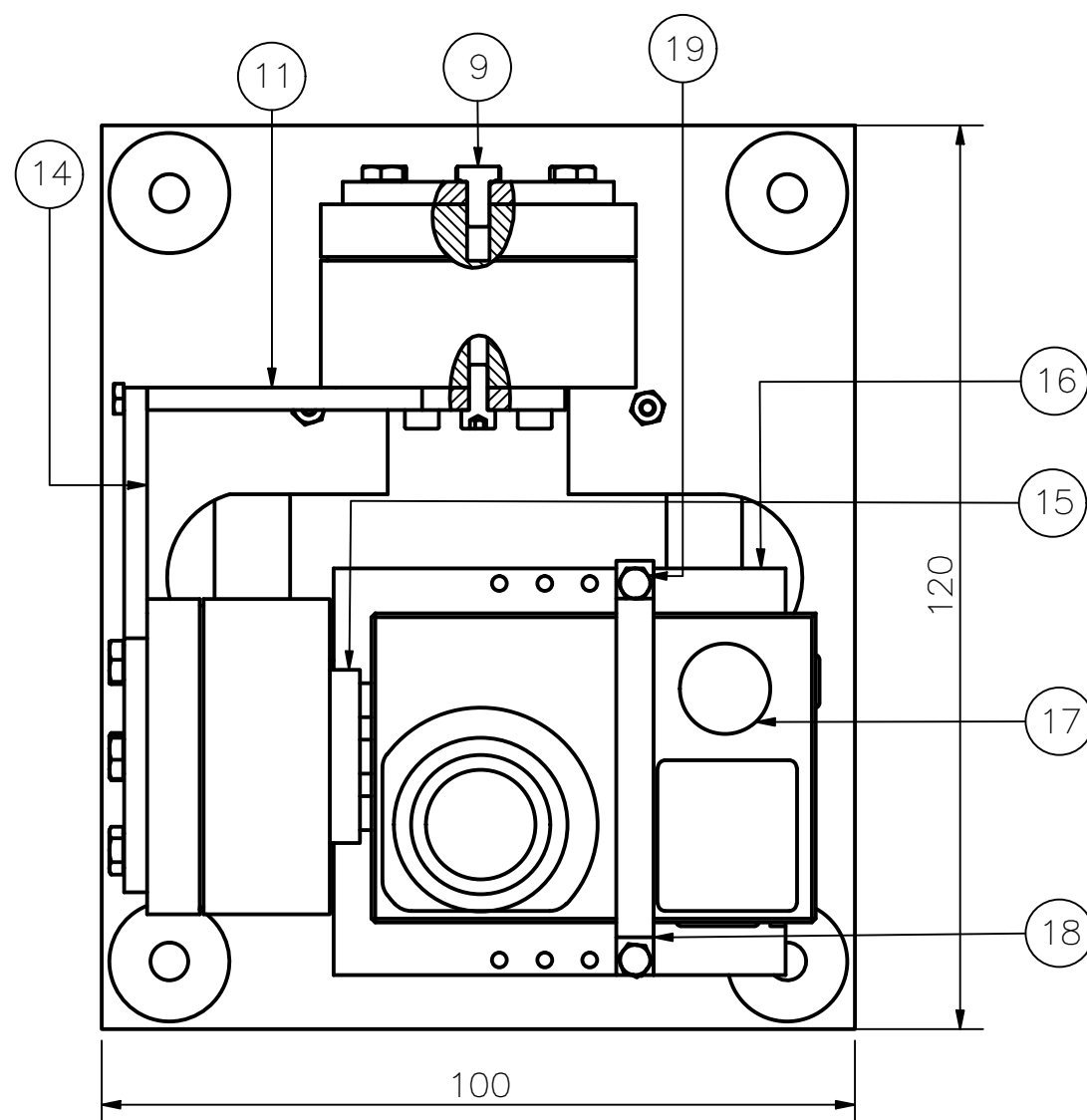


ACABADO SUPERFICIAL	TOLERANCIA GENERAL	MATERIAL
		al-6061
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ CIENCIAS E INGENIERÍA-ING.MECATRÓNICA		
MÉTODO DE PROYECCIÓN	1MTR02	ESCALA
	SOPORTE PARA ODRROID	1:1
20150561	SALAZAR AYLLÓN, PAOLO	FECHA: 2019.09.08
		LÁMINA: A4

COTA NOMINAL	COTA MÁXIMA	COTA MÍNIMA



CORTE A-A



DETALLE A
ESCALA 5 : 1

LISTA DE MATERIALES

POS.	CANT.	DESCRIPCIÓN	NORMA	MATERIAL	OBSERVACIONES
1	1	Soporte de Odroid		Aluminio 6061	
2	4	Tuerca hexagonal M3		8.8	
3	1	Soporte para AlexMos		Aluminio 6061	
4	1	Controlador AlexMos		Plástico	
5	8	Tornillo cabeza hexagonal M2 x 6		8.8	
6	4	Tuerca hexagonal M2		8.8	
7	4	Amortiguador		Caucho	
8	1	Brazo1		Aluminio 6061	
9	8	Tornillo cabeza hexagonal M3 x 6		8.8	
10	2	Motor		Plástico	
11	1	Brazo2		Aluminio 6061	
12	8	Tornillo cabeza hexagonal M2.5 x 6		8.8	
13	2	Tornillo cabeza hexagonal M2 x 8		8.8	
14	1	Brazo3		Aluminio 6061	
15	1	Brazo4		Aluminio 6061	
16	1	Soporte de cámara		Aluminio 6061	
17	1	Cámara		Plástico	
18	1	Sujetador de cámara		Aluminio 6061	
19	2	Tornillo cabeza hexagonal M2 x 10		8.8	
20	2	Tuerca hexagonal M2		8.8	
21	1	IMU		Plastico	

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
CIENCIAS E INGENIERÍA – ING. MECATRÓNICA

MÉTODO DE PROYECCIÓN	TRABAJO DE FIN DE CARRERA	ESCALA
	ESTABILIZADOR DE CÁMARA	1:1
20150561	SALAZAR AYLLON, PAOLO	FECHA: 2019.09.2
		LÁMINA: A3