

Universidad Católica de Santa María
Facultad de Ciencias e Ingenieras Físicas y
Formales
Escuela Profesional de Ingeniería Mecánica,
Mecánica Eléctrica y Mecatrónica



**DISEÑO Y EVALUACION DE UN SISTEMA DE CONTROL PID Y UN
SISTEMA DE CONTROL DIFUSO DE UN CUADRICOPTERO EN MODULOS
EXPERIMENTALES**

Tesis presentada por la Bachiller:

Gutierrez Sucla, Andree

para optar el Título Profesional de

Ingeniero Mecatrónico

Asesor:

**Ing. Mestas Ramos, Sergio
Orlando**

Arequipa- Perú

2021

UCSM-ERP

UNIVERSIDAD CATÓLICA DE SANTA MARÍA
INGENIERIA MECANICA, MECANICA-ELECTRICA Y MECATRONICA
DICTAMEN APROBACIÓN DE BORRADOR DE TESIS

Arequipa, 31 de Octubre del 2020

Dictamen: 000932-C-EPIMMEM-2020

Visto el borrador de tesis del expediente 000932, presentado por:

2011802251 - GUTIERREZ SUCLLA ANDREE

Titulado:

**DISEÑO Y EVALUACION DE UN SISTEMA DE CONTROL PID Y UN SISTEMA DE CONTROL DIFUSO
DE UN CUADRICOPTERO EN MODULOS EXPERIMENTALES**

Nuestro dictamen es:

APROBADO

**1936 - MESTAS RAMOS SERGIO ORLANDO
DICTAMINADOR**



**2213 - QUISPE CCACHUCO MARCELO JAIME
DICTAMINADOR**



Dedicatorias

Dedico este trabajo a mis padres, por su incondicional apoyo a lo largo de mi carrera profesional y en todos mis proyectos personales a lo largo de mi vida. A mi familia por estar conmigo, apoyarme siempre y en especial a mi hermano Eduardo. A mi enamorada por su apoyo constante y presión para terminar este trabajo por darme los ánimos para culminar este proyecto.



Agradecimiento



Agradezco a los docentes e ingenieros de mi universidad por las lecciones y apoyo constante para mi formación profesional, gracias a ellos me encuentro en el lugar donde estoy

A las personas que son importantes y muy especiales para mí, porque me ayudaron de una manera desinteresada además que siempre estuvieron conmigo dándome ánimos y apoyándome en todo momento a seguir avanzando y crecer como profesional y persona.

RESUMEN

En esta tesis se propone el desarrollo de distintos sistemas de control PID como FUZZY para el funcionamiento de un cuadricoptero, y evaluar dichos sistemas desarrollados en la plataforma real mediante el uso de módulos experimentales.

Un cuadricoptero es un helicóptero con cuatro rotores, lo cual permite tener una mayor estabilidad que un helicóptero normal. El cuadricoptero posee seis grados de libertad, tres que determina su posición: desplazamiento en el X, desplazamiento en el Y y desplazamiento en el Z; y otros tres que determinan en la orientación del cuerpo: Roll, Pitch y Yaw. Al desarrollar el módulo matemático nos damos cuenta que es un sistema MIMO, en cual sus principales variables de entrada son: Z, Roll, Pitch y Yaw, dando entender que nos encontramos frente a un sistema de control en cascada.

Los sistemas desarrollados deben ser capaz de estabilizarse en los siguientes movimientos: la elevación vertical y movimientos angulares. Para la evaluación se desarrolló una plataforma GUIDE para una mayor interacción con la persona, en dicha plataforma se establecieron trenes de pulsos, los cuales están en una base de datos y de ser necesario se pueden agregar otras nuevas.

Para una mayor seguridad en las pruebas de experimentación, se implementó el uso de dos módulos para las distintas entradas que posee el equipo. Se implementó un módulo HOVER para el desplazamiento en Z y el módulo STABILITY para los movimientos angulares ROLL, PITCH y YAW.

Finalmente se comparará el desempeño utilizando los diferentes tipos de control, encontrando que cual sistema es el más estable y se acerca más a los set points deseados.

Palabras claves:

PID, FUZZY, cuadricoptero, MIMO, GUIDE, set points

ABSTRACT

This thesis proposes the development of different PID control systems such as FUZZY for the operation of a quadcopter, and evaluate these systems developed on the real platform through the use of experimental modules.

A quadcopter is a helicopter with four rotors, which allows for greater stability than a normal helicopter. The quadcopter has six degrees of freedom, three that determines its position: displacement in the X, displacement in the Y and displacement in the Z; and three others that determine the orientation of the body: Roll, Pitch and Yaw. When developing the mathematical module, we realize that it is a MIMO system, in which its main input variables are: Z, Roll, Pitch and Yaw, implying that we are facing a cascade control system.

The developed systems must be able to stabilize in the following movements: vertical elevation and angular movements. For the evaluation, a GUIDE platform was developed for greater interaction with the person, on this platform pulse trains were established, which are in a database and, if necessary, new ones can be added.

For greater security in the experimentation tests, the use of two modules for the different inputs that the team has is implemented. A HOVER module for Z displacement and the STABILITY module for ROLL, PITCH and YAW angular movements were implemented.

Finally, performance will be compared using the different types of control, finding which system is the most stable and closer to the desired set points.

Key words:

PID, FUZZY, quadcopter, MIMO, GUIDE, set points

INTRODUCCIÓN

El presente trabajo de investigación tiene por finalidad, diseñar distintos sistema de control de un Cuadricoptero (UAV) utilizando métodos de control PID y métodos de control DIFUSO, al hacer una comparativa entre ambos, podremos decidir qué control es el más conveniente para este tipo de sistema MIMO

El proyecto consta de cinco capítulos, los mismos que serán descritos a continuación:

El **Capítulo I Marco Metodológico** describe el planteamiento y justificación del problema a investigar, así como los objetivos generales y específicos que se esperan obtener de este estudio.

El **Capítulo II Marco Teórico** describe toda la información necesaria para abordar el problema planteado. Se describe temas como conceptos básicos a considerar, conceptos sobre los cuadricópteros, conceptos sobre sistemas de control y modelamiento matemáticos

El **Capítulo III Diseño del Sistema de Control** contiene a detalle todo lo desarrollado para obtener los distintos sistemas de control. Se divide en 4 puntos principales a considerar, los cuales son los bloques de implementación, el modelo de planeación, el modelo de estabilidad y la interfaz gráfica

El **Capítulo IV Pruebas** contiene los resultados obtenidos de las pruebas planteadas para poder verificar la eficacia de los sistemas de control planteados

El **Capítulo V Aspectos Administrativos** nos indica el plan de gestión de cronograma y costos que se realizó para poder alcanzar los objetivos planteados

ÍNDICE

DICTAMEN APROBATORIO.....	i
DEDICATORIAS	ii
AGRADECIMIENTO	iii
RESUMEN	iv
ABSTRACT.....	v
INTRODUCCIÓN	vi
ÍNDICE	vii
ÍNDICE FIGURAS	x
ÍNDICE ECUACIONES	xiv
ÍNDICE TABLAS.....	xvi
CAPITULO I MARCO METODOLÓGICO.....	1
1 Planteamiento del problema.....	4
2 Metodología de la investigación	5
3 Justificación del problema.....	6
4 Antecedentes	6
5 Objetivos:.....	8
5.1 Objetivo general.....	8
5.2 Objetivos específicos	8
6 Hipótesis	9
7 Alcances y limitaciones	9
7.1 Alcances	9
7.2 Limitaciones	9
8 Contribuciones de la tesis:.....	9
9 Requerimientos de trabajo	10
CAPITULO II MARCO TEÓRICO	11
1 Conceptos básicos.....	12

1.1	Chasis o frame	12
1.2	Sensores.....	13
1.3	Actuador	15
1.4	Controlador	16
2	Cuadricoptero:.....	18
2.1	Denominaciones	19
2.2	Clasificación.....	19
2.3	Características del cuadricóptero:	21
3	Modelo Matemático	35
4	Módulos experimentales.....	36
5	Sistema de control a implementar en el proyecto	39
5.1	Control PID	39
5.2	Control Difuso	43
6	Modelo matemático:.....	45
6.1	Modelo cinemático.....	45
6.2	Modelo dinámico.....	48
CAPITULO III DISEÑO DEL SISTEMA DE CONTROL.....		70
1	Bloques de implementacion	72
1.1	Bloques de entradas:.....	72
1.2	Bloque de sensores:.....	73
1.3	Bloque de planta:	80
1.4	Bloque de control	81
2	Modelo de planeación (hover):.....	85
3	Modelo de estabilización:	87
4	Interfaz grafica	88
4.1	Módulo automático	89
4.2	Modulo manual.....	90

CAPITULO IV PRUEBAS	92
1 Planeación (HOVER)	93
1.1 Control PID	95
1.2 Control DIFUSO.....	96
2 Estabilidad (STABILITY).....	97
2.1 Control PID	100
2.2 Control DIFUSO.....	102
CAPITULO V ASPECTOS ADMINISTRATIVOS.....	107
1 Diagrama de Gantt:	108
2 Costos	109
CONCLUSIONES	111
RECOMENDACIONES	113
Recomendaciones	114
Trabajos futuros	114
REFERENCIAS BIBLIOGRÁFICAS	116
ANEXOS	121
Anexo 01: PARROT AR DRONE 2.0 - Características generales	122
Anexo 02: Planos de soporte de prueba vertical (HOVER).....	127
Anexo 03: Modulo de estabilidad (STABILITY)	156
Anexo 04: AR Drone Developer Guide 3-12p.	188
Anexo 05: SIMULINK	198
Modulo Control.....	219
Anexo 06: Guide	227
Guide.....	227
Automático	231
Manual.....	237

ÍNDICE FIGURAS

Figura 1 Ramas que conforman la ingeniería mecatrónica	2
Figura 2 Brazo robótico	3
Figura 3 Partes principales de un robot	4
Figura 4 Cuadricoptero Ar. Drone 2.0.....	5
Figura 5 Frame de un robot	12
Figura 6 Sensor de temperatura	15
Figura 7 Actuador neumático.....	15
Figura 8 Programador lógico controlable	17
Figura 9 MATLAB	18
Figura 10 Aplicaciones de los Micro-Vehículos Aéreos	18
Figura 11 Clasificación de Robots Móviles.....	20
Figura 12 Parrot AR Drone 2.0	21
Figura 13 Características Mecánicas.....	22
Figura 14 Batería LiPo	23
Figura 15 Sensor ultrasónico	23
Figura 16 Acelerómetro	24
Figura 17 Sensor Giroscopio	24
Figura 18 Motor Brushless	25
Figura 19 Ejes de vuelo	26
Figura 20 Rotación en el eje Yaw.....	27
Figura 21 Rotación en el eje Pitch	27
Figura 22 Rotación en el eje Roll	28
Figura 23 Fuerza de empuje vs Peso.....	28
Figura 24 Configuración outdoor e indoor.....	30
Figura 25 Movimiento de Elevación.....	31
Figura 26 Giro Anti-Horario	32
Figura 27 Giro Horario	32
Figura 28 Giro Hacia Adelante	33
Figura 29 Giro Hacia Atrás	33
Figura 30 Giro a la Derecha.....	34
Figura 31 Giro a la Izquierda	34
Figura 32 Modelo matemático	35

Figura 33 Módulo experimental 1	38
Figura 34 Modulo experimental 2	39
Figura 35 Bloques de control PID	41
Figura 36 Grados de verdad	44
Figura 37 Cuadro de control de reglas difuso	45
Figura 38 Sistemas de referencias del cuadricóptero	46
Figura 39 Modelo de Fuerzas y Momentos	49
Figura 40 Comportamiento del motor	50
Figura 41 Modelo de Fuerzas y Momentos con sistema de coordenadas relativo	51
Figura 42 Modelo de elevación vertical	59
Figura 43 Modelo de rotación en Roll	60
Figura 44 Modelo de rotación en Pitch	60
Figura 45 Modelo de rotación en Yaw	61
Figura 46 Consideración geométrica para modelo no lineal	62
Figura 47 Matriz A obtenida del Matlab	67
Figura 48 Matriz B obtenida del Matlab	68
Figura 49 Matriz C obtenida del Matlab	68
Figura 50 Matrices de Taylor desarrolladas	69
Figura 51 Diagrama de bloques control cascada	71
Figura 52 Programa Simulink	71
Figura 53 Bloque de entradas	72
Figura 54 Entradas deseadas	73
Figura 55 Bloque de sensores	73
Figura 56 Procesamiento de señal	74
Figura 57 Rotación de un cuerpo rígido	78
Figura 58 Señal Raw del giroscopio	79
Figura 59 Señal calibrada del giroscopio	80
Figura 60 Bloque de Planta	80
Figura 61 Bloques de accionamiento	81
Figura 62 Bloque de control	81
Figura 63 Control de altura, raw, pitch y yaw	82
Figura 64 Bloque de control PID y bloque de control difuso	82
Figura 65 Funciones de membresía	84

Figura 66 Grafico de reglas intrínsecas y superficies	84
Figura 67 Modelo de planeación	86
Figura 68 Modulo de experimentación HOVER	86
Figura 69 Modulo de experimentación elevación	87
Figura 70 Modulo de experimentación STABILITY	87
Figura 71 Modulo de experimentación de estabilidad	88
Figura 72 GUI del proyecto	89
Figura 73 GUI automático	89
Figura 74 GUI manual	90
Figura 75 Señal de entrada para Z	93
Figura 76 Planeación a una altura media y altura máxima	94
Figura 77 Planeación entre una altura media y altura máxima en modulo experimental	94
Figura 78 Resultados con controlador PID en EJE Z	95
Figura 79 Control Difuso de altura en el eje Z	96
Figura 80 Resultados con controlador Difuso en EJE Z	97
Figura 81 Señal de entrada para ROLL	98
Figura 82 Señal de entrada para PITCH	98
Figura 83 Señal de entrada para YAW	99
Figura 84 Posicionamiento angular en Roll, Pitch y Yaw	99
Figura 85 Posicionamiento angular en Roll, Pitch y Yaw en modulo experimental	100
Figura 86 Resultados con controlador PID en Roll	101
Figura 87 Resultados con controlador PID en	101
Figura 88 Resultados con controlador PID en Yaw	102
Figura 89 Control Difuso de Roll	103
Figura 90 Resultados con controlador Difuso en Roll	103
Figura 91 Control Difuso de Pitch	104
Figura 92 Resultados con controlador Difuso en Pitch	104
Figura 93 Control Difuso de Yaw	105
Figura 94 Resultados con controlador Difuso en Yaw	106
Figura 95 Diagrama de Gantt	108
Figura 96 Diagrama de bloques desarrollado	198
Figura 97 Modulo entradas desarrollado	199

Figura 98 Modulo SP	200
Figura 99 Modulo SP Establecido	201
Figura 100 Tren de pulsos ROLL	202
Figura 101 Modulo de Sensores	203
Figura 102 Acondicionamiento de entrada	204
Figura 103 Adquisición de sensores	205
Figura 104 Calibración de sensores	206
Figura 105 Calibración acelerómetro	207
Figura 106 Parámetros acelerómetro	207
Figura 107 Calibración giroscopio.....	208
Figura 108 Parámetros de giroscopio	208
Figura 109 Modulo cálculos	209
Figura 110 Cálculos sensor ultrasónico	210
Figura 111 Cálculos ángulos de Eurler	213
Figura 112 Cálculos acelerómetro	214
Figura 113 Cálculos giroscopio	215
Figura 114 Calibración de ángulo Yaw giroscopio.....	218
Figura 115 Modulo de control	219
Figura 116 Bloques de control altura	220
Figura 117 Bloques de control Difuso altura	220
Figura 118 Bloques de control roll	221
Figura 119 Bloques de control Difuso roll	221
Figura 120 Bloques de control pitch	222
Figura 121 Bloques de control Difuso pitch	222
Figura 122 Bloques de control yaw	223
Figura 123 Bloques de control Difuso yaw	223
Figura 124 Bloques acondicionamiento de salidas	224
Figura 125 Bloques de monitoreo.....	225
Figura 126 Bloques de scope.....	226

ÍNDICE ECUACIONES

Ecuación 1 Relación peso – fuerzas.....	28
Ecuación 2 Control PID.....	39
Ecuación 3 Vector de posición.	46
Ecuación 4 Vector de orientación.....	47
Ecuación 5 Matriz de Rotación Roll – Pitch – Yaw	47
Ecuación 6 Raw-Pitch-Yaw.....	47
Ecuación 7 Ecuación de Euler ZYX	48
Ecuación 8 Matriz de Rotación ZYX Simplificada.....	48
Ecuación 9 Empuje aerodinámico	49
Ecuación 10 Empuje individual aerodinámico.....	49
Ecuación 11 Momento aerodinámico	50
Ecuación 12 Fuerzas en equilibrio (HOVER).....	50
Ecuación 13 Momentos En Equilibrio	51
Ecuación 14 Coordenada generalizada del cuadricóptero.....	51
Ecuación 15 Euler-Lagrange	52
Ecuación 16 Función Lagrangiana	52
Ecuación 17 Energía potencial.....	52
Ecuación 18 Energía cinética en traslación	52
Ecuación 19 Energía cinética en rotación.....	52
Ecuación 20 Jacobiana Global.....	52
Ecuación 21 Jacobiana Local.....	53
Ecuación 22 Matriz de inercias	53
Ecuación 23 Función Lagrangiana de traslación	53
Ecuación 24 Fuerza Global.....	54
Ecuación 25 Ecuaciones dinámicas de traslación	54
Ecuación 26 Función Lagrangiana de rotación	54
Ecuación 27 Matriz de Coriolis.....	55
Ecuación 28 Euler-Lagrange de rotación.....	55
Ecuación 29 Matriz de Coriolis.....	56
Ecuación 30 Linealización.....	56
Ecuación 31 Matriz de Coriolis lineal	56
Ecuación 32 Matriz Jacobiana lineal	56

Ecuación 33 Ecuaciones dinámicas de rotación.....	57
Ecuación 34 Newton-Euler	57
Ecuación 35 Ecuación de newton para las fuerzas	57
Ecuación 36 Ecuaciones dinámicas de traslación	58
Ecuación 37 Ecuación de newton para los torques	58
Ecuación 38 Ecuaciones dinámicas de rotación.....	59
Ecuación 39 Ecuaciones dinámica de elevación.....	59
Ecuación 40 Ecuaciones dinámicas de rotación en Roll	60
Ecuación 41 Ecuaciones dinámicas de rotación en Pitch	60
Ecuación 42 Ecuaciones dinámicas de rotación en Yaw	61
Ecuación 43 Matriz de velocidades angulares.....	61
Ecuación 44 Sumatoria de velocidades angulares de las hélices	61
Ecuación 45 Efectos producidos por la rotación de las hélices	62
Ecuación 46 Ecuaciones de rotación no lineales.....	63
Ecuación 47 Momento de inercia rotacional alrededor del eje	63
Ecuación 48 Matriz de traslación no lineal con perturbación	63
Ecuación 49 Matriz de rotación no lineal con perturbación	63
Ecuación 50 Funciones de traslación y rotación para liberalización	64
Ecuación 51 Serie de Taylor general	64
Ecuación 52 Condición de derivadas igual a 0.....	65
Ecuación 53 Condición de derivadas igual a 0 – Punto de operación respecto a u_1	65
Ecuación 54 Condición de derivadas igual a 0 – Punto de operación respecto a u_2	65
Ecuación 55 Condición de derivadas igual a 0 – Punto de operación respecto a u_3 y u_4	66
Ecuación 56 Condición generalizada	66
Ecuación 57 Ecuaciones de Taylor en base al objetivo.....	66
Ecuación 58 Matrices de Taylor	67
Ecuación 59 Ecuaciones de calibración del ultrasonido	75
Ecuación 60 Ecuaciones de calibración del acelerómetro	75
Ecuación 61 Obtención de ROLL PITCH YAW	78

ÍNDICE TABLAS

Tabla 1 Características de los sensores	14
Tabla 2 Tipos comunes de sensores.....	14
Tabla 3 Tipos de actuadores	16
Tabla 4 Clasificación de Cuadricoptero	19
Tabla 5 Fases para la construcción de un modelo.....	36
Tabla 6 Sensores implementados en el Ar. Drone 2.0	74
Tabla 7 Calibración del sensor ultrasónico	75
Tabla 8 Calibración del acelerómetro en el eje X	76
Tabla 9 Calibración del acelerómetro en el eje Y	76
Tabla 10 Calibración del acelerómetro en el eje Z.....	77
Tabla 11 Calibración del acelerómetro.....	77
Tabla 12 Reglas Fuzzy	85
Tabla 13 Valores PID en el eje Z.....	95
Tabla 14 Valores PID en Roll, Pitch y Yaw	100
Tabla 15 Costos realizados	109



CAPITULO I

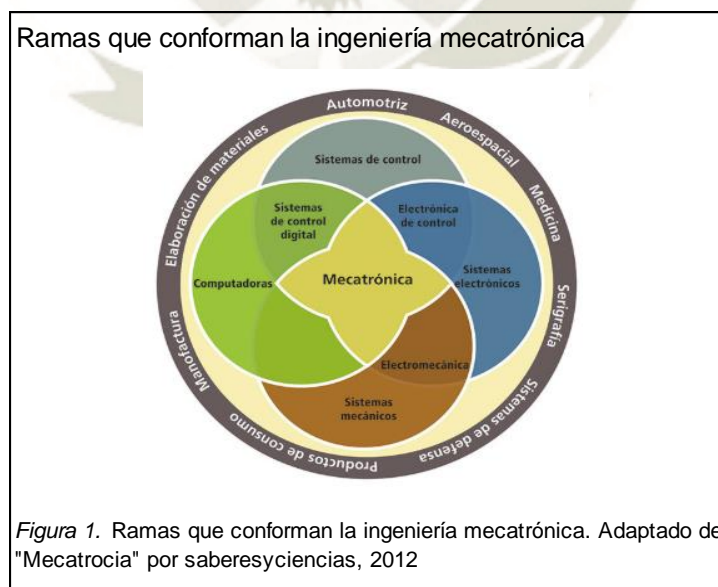
MARCO METODOLÓGICO

Según (Mechanical and Mechatronics Engineering Department, 2020) refiere que:

La mecatrónica - palabra acrónimo de mecánica y electrónica -, inunda casi todos los aspectos de la sociedad. Actualmente las máquinas, equipos, electrodomésticos y unidades informáticas son concebidos desde una perspectiva mecatrónica. Es decir, son sistemas que mezclan en su funcionamiento, componentes mecánicos y electrónicos. La sinergia entre dichas áreas busca crear productos inteligentes, con mejores cualidades respecto a los demás, capaces de procesar paralelamente diversas informaciones para optimizar el funcionamiento, mejorar la productividad y el desempeño (p. 46)

La ingeniería mecatrónica es considerada una ingeniería multidisciplinaria, debido a que es la integración de distintas áreas de conocimiento. Las principales áreas de conocimiento son:

- Conocimiento de computación
- Conocimiento de mecánica industrial,
- Conocimiento de electricidad y electrónica
- Conocimiento de robótica.



Desde los inicios de la robótica siempre se buscó obtener maquinas autónomas con capacidades que ayuden a la humanidad en situaciones adversas (situaciones de peligro hacia las personas). La mayoría de personas, considera los brazos robóticos de ensamblaje como la cúspide de la robótica, pero en verdad puede ser cualquier sistema que se pueda imaginar con algunos de los elementos dichos

Brazo robotico



Figura 2. Brazo robotico indutrail. Adaptacion propia en base a la investigacion realizada, 2019

Según la Federación Internacional de Robótica (IFR, 2019) define que un robot es un mecanismo de acción programable que realiza tareas útiles para humanos o equipos, con un grado de autonomía, es decir, realizar tareas sin la intervención humana.

De manera más general, un robot, o sistema robótico, se resume en un conjunto de 4 partes principales:

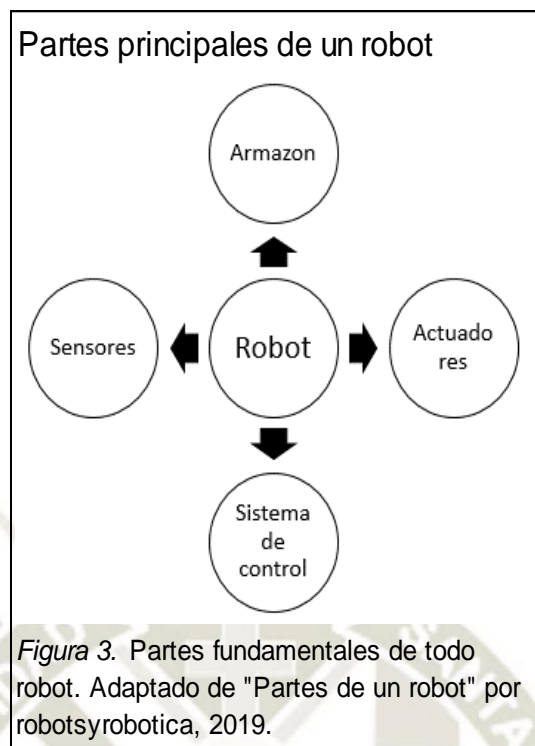


Figura 3. Partes principales de un robot

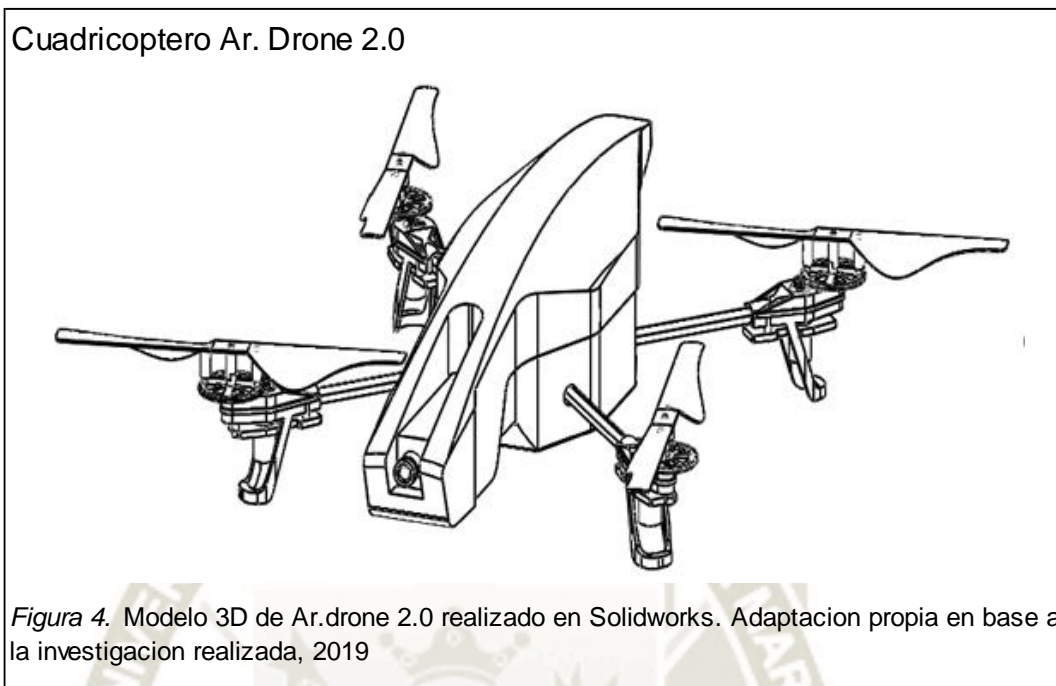
Últimamente, debido al incremento de la tecnología, conseguir las herramientas necesarias para la construcción de robots aéreos es relativamente fácil, por ello se ha generado un gran interés en esta área de la robótica aérea entre ellos hay una rama denominada cuadricópteros, quadcopter o quadrotor

1 Planteamiento del problema

En la actualidad la industria de los vehículos aéreos no tripulados (UAV por sus siglas en inglés) tiene una gran acogida por la población, esto es debido a que los sistemas tecnológicos son de fácil alcance y son de gran ayuda para la sociedad en la realización de distintas tareas, como: entretenimiento, investigación, transporte, monitoreo y localización, entre otras.

Debido a la gran practicidad y la gran cantidad de tareas que se pueden realizar con un cuadricoptero surge la necesidad de crear sistemas de control, para eliminar el factor humano, el cual necesita ser capacitado y tener un área de trabajo adecuado para poder emplear un cuadricoptero de forma óptima, estos errores generan una disminución de la autonomía del cuadricoptero, debido a que el tiempo de respuesta para alcanzar la estabilidad es alto y depende principalmente del criterio y la

capacidad del usuario. Para ello, los UAV necesitan de técnicas de control, además de una autonomía para su funcionamiento y el buen desarrollo de las tareas.



Es así como, mediante el análisis de estas necesidades y las diferentes áreas de aplicación para los robots móviles, en este proyecto se diseñó y evaluó distintos sistemas de control para un cuadricoptero o quadrotor, por la complejidad que posee, debido a que el cuadricoptero es un sistema subactuado.

2 Metodología de la investigación

- **Diseño de investigación:** Investigación aplicada
- **Hipótesis:** Tipo causalidad
- **Arena de conocimiento:** Robótica
- **Sujeto de prueba:** Cuadricoptero - Drone (Parrot AR Drone 2.0)
- **Instrumentos:** Computadora, programa Matlab
- **Técnicas:** Control PID y Control DIFUSO
- **Procedimiento:** Pruebas en módulos experimentales
- **Análisis de datos:** En tiempo real y gráficos de respuesta

3 Justificación del problema.

Actualmente existe un gran avance en tecnologías nuevas, en especial, existen muchos centros de investigación que realizan proyectos relacionados con los cuadricópteros, tanto en el entorno nacional como el internacionalmente.

En el Perú no se encontraron muchos trabajos previos sobre esta investigación que hayan diseñado sistemas de control para cuadricópteros o drones. Por lo cual se vio la oportunidad para realizar un trabajo de tesis con este tipo de dispositivos. Buscando como meta el incremento de la autonomía del cuadricóptero, minimizando el tiempo de respuesta de los actuadores del cuadricóptero y alcanzar un modo de estabilidad más rápido al eliminar el control manual (errores humanos)

Se buscará las diferentes maneras de controlar un cuadricóptero, debido a que es un sistema MIMO (Múltiple Entrada Múltiple Salida), las ecuaciones de movimiento que rigen la dinámica del sistema son no lineales y posee una gran variedad de sensores que podrían realizar un feedback adecuado.

Por otro lado, el desarrollo de este proyecto mediante el uso de diversos módulos permitirá a estudiantes y profesores de la universidad, llevar a la práctica algunos conceptos de Ingeniería en la rama de Mecatrónica. Pudiendo utilizar este módulo en materias como: control I, II y III, Proyecto Mecatrónico, Sistemas digitales, Procesamiento de señales, Robótica I y Robótica II. Considerando que en dichas materias el presente trabajo se podría mostrar como un ejemplo práctico de la aplicación de los conocimientos teóricos.

4 Antecedentes

- Gil (2018) “Diseño y Construcción de un Cuadricóptero para el Monitoreo de Niveles de Contaminación Acústica y Parámetros Físicos Ambientales en la Ciudad de Arequipa.” Universidad Católica de Santa María, Arequipa–Perú.

La investigación tuvo como objetivo el diseño de un cuadricoptero para un sistema de monitoreo en el cual se encargaba de detectar los niveles de contaminación acústica, parámetros ambientales (temperatura, humedad, nivel de radiación UV, presión barométrica, niveles de luminosidad), niveles de monóxido de carbono y nivel de dióxido de carbono en la ciudad de Arequipa

Para la realización de esta investigación, Gil realizó distintas pruebas del sistema integrado, consiguiendo un cuadricoptero controlado por radio frecuencia que puede llegar a partes de la ciudad Arequipa. El incremento del parque automotor ha hecho que aumente indiscriminadamente la contaminación acústica y la variación exagerada de las condiciones ambientales.

Gil llegó a las siguientes conclusiones:

- Debido a diferentes problemáticas se vio en la necesidad de hacer un cambio en la estructura
- El uso de equipos combinados con el sistema de monitoreo que se construyó se permitió conocer cuáles son los puntos existentes en la ciudad de Arequipa con un mayor índice de contaminación acústica, contaminación del aire y el incremento de parámetros ambientales debido a diversos factores.

➤ Paiva (2016) “MODELADO Y CONTROL DE UN CUADRICÓPTERO.”
Universidad de Piura, Piura-Perú.

El trabajo realizado consiste en el desarrollo del modelo matemático, la estimación de parámetros físicos, el diseño de controles PID para el cuadricóptero, la validación del modelo matemático y la implementación experimental del algoritmo de control en un sistema embebido para estabilizar la orientación. La tesis contempla realizar el modelo matemático de la aerodinámica del cuadricóptero agregando la dinámica de los propulsores

Paiva llegó a las siguientes conclusiones:

- Debido a diferentes problemáticas se vio en la necesidad de hacer un cambio en la estructura

- El uso de equipos combinados con el sistema de monitoreo que se construyó se permitió conocer cuáles son los puntos existentes en la ciudad de Arequipa con un mayor índice de contaminación acústica, contaminación del aire y el incremento de parámetros ambientales debido a diversos factores.

5 Objetivos:

Para la realización del presente trabajo, se plantearon los objetivos que se muestran a continuación:

5.1 Objetivo general

Diseñar un sistema de control de un Cuadróptero (UAV) aplicando métodos de control PID y métodos de control DIFUSO

5.2 Objetivos específicos

Diseñar un sistema de control de un Cuadróptero (UAV) aplicando métodos de control PID y métodos de control DIFUSO

- Obtener el modelo matemático de un Cuadróptero (UAV)
- Diseñar una interfaz que permita al usuario la manipulación y control del Cuadróptero.
- Diseñar sistemas de control para las variables de entrada: desplazamiento Z, Roll, Pitch y Yaw
- Hacer una comparación entre control PID y control DIFUSO.
- Emplear módulos experimentales para una mayor facilidad de pruebas reales
- Validación del diseño del sistema de control

6 Hipótesis

Con el diseño y evaluación de los sistemas de control podemos obtener una autonomía mayor, emplear eficientemente los recursos del cuadricoptero, reduciendo el factor humano en el control, debido a que este es manual

7 Alcances y limitaciones

7.1 Alcances

El presente proyecto compara dos tipos de control (Control PID y Control Difuso) y se determinara cual es mejor para este tipo de proyectos

7.2 Limitaciones

Una de las limitaciones del presente proyecto es el lead time del cuadricoptero, debido a que es un producto de Europa

8 Contribuciones de la tesis:

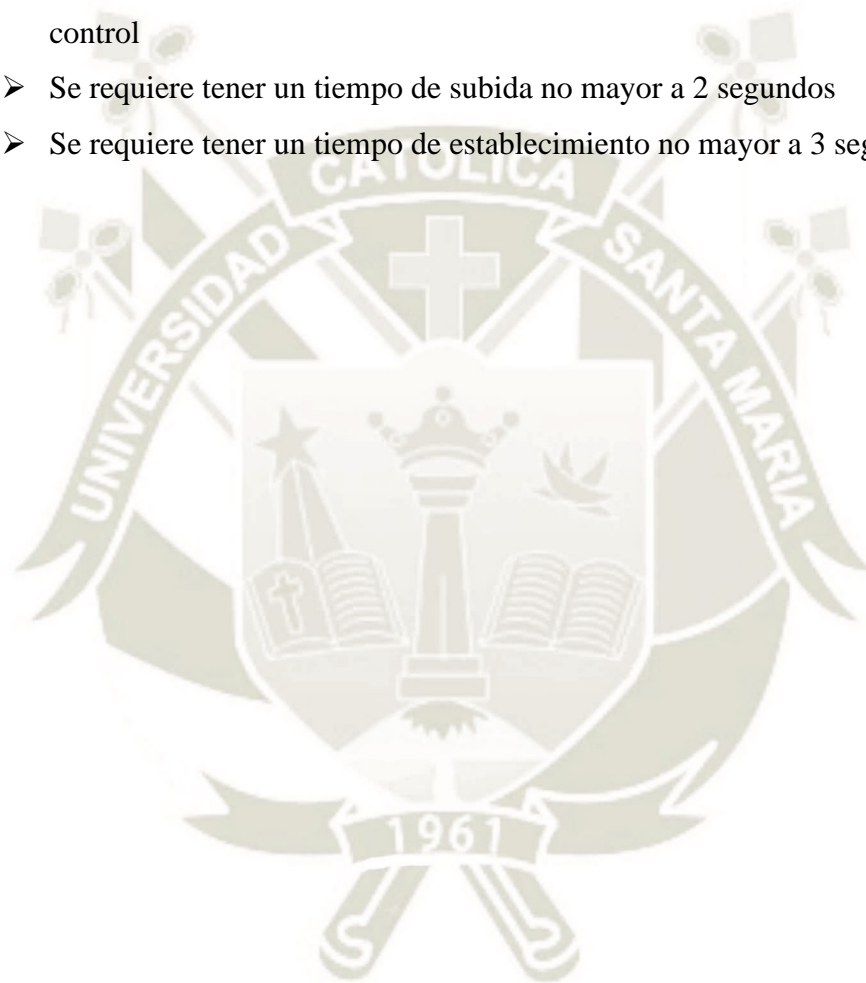
A continuación, se presenta las contribuciones del presente proyecto, luego de su desarrollo:

- Obtención y simplificación del modelo dinámico de un cuadricoptero, para poder así implementar las leyes de control.
- Desarrollo una plataforma de simulación para el seguimiento de un cuadricoptero

9 Requerimientos de trabajo

A continuación, se presenta los requerimientos para la evaluación del presente proyecto con los set points ya establecidos, las cuales son:

- Se requiere tener un error en estado estacionario los más aproximado a 0
- Se requiere tener un máximo sobre impulso no mayor del 20% para cada control
- Se requiere tener un tiempo de subida no mayor a 2 segundos
- Se requiere tener un tiempo de establecimiento no mayor a 3 segundos





CAPITULO II

MARCO TEÓRICO

1 Conceptos básicos

El proyecto está basado en la implementación de un robot, por lo tanto, es necesario entender las parte que lo conforman:

1.1 Chasis o frame

Según Rocha, (2020) nos explica que el chasis es:

Es la parte del robot que se considera la base para soportar las otras partes de un robot; de forma similar a un vehículo, un robot necesita una estructura rígida donde se puedan ubicar las diferentes piezas, como sensores, servomotores, controlador, baterías de alimentación y demás componentes. Dependiendo de la utilidad del robot cambia la forma de su esqueleto. También dependiendo del uso del robot, el chasis debe contar con ciertas características especiales (párr.1-3)

Las principales características para seleccionar un Frame son la resistencia de materiales, el peso y el tamaño



1.2 Sensores

Serna Ruiz, Ros Garcia, & Rico Noruega, (2010) nos indican que un sensor:

Son dispositivos electrónicos que imitan la capacidad de percepción de los seres humanos, por ellos se incorporan en cualquier área tecnológica. Los sensores nos permiten interactuar con el entorno, de forma que nos proporcionan datos de ciertas variables que nos rodean para poder procesarlas y así poder generar órdenes a los actuadores. (pág. 3)

Hay que tener en claro la definición de un sensor, ya que generalmente la gente suele confundirlo con un transductor, según (Pallàs Areny, 2005)

Un sensor es un dispositivo que mide una magnitud física en una señal fácilmente legible, y un transductor es un dispositivo capaz de transformar o convertir una determinada magnitud física de entrada, en otra diferente, especialmente en una magnitud eléctrica

Por esta explicación, podemos diferenciar un transductor de un sensor, por ello se tiene claro que todo transductor es considerado un sensor, pero no todo sensor es considerado un transductor.

1.2.1 Características principales de los sensores

Al momento de seleccionar un sensor se debe de tener en consideración las siguientes características técnicas y aspectos para obtener el rendimiento necesario esperado según el sistema que se desee implementar

Tabla 1
Características sensores

Característica	Descripcion
Resolucion	Es la minima variacion de la magnitud de entrada que se puede apreciar a la salida
Sensibilidad	Es la relacion entre la variacion de la magnitud de salida y la variacion de la magnitud de entrada
Error	Es la desviacion de medida del sensor respecto a lo real (suele expresarse en%)
Precision	Es el error de medida maximo esperado
Repetibilidad	Es el error esperado al repetir varias veces

Nota. Característica de los sensores. Adaptado de "Guía Práctica de Sensores" por A.Serna,F.Ros y J.Rico, 2010, p. 7.

Tabla 2
Tipos comunes de sensores

Magnitud	Sensores
Iluminacion	Fotorresistivos (LDR)
Temperatura	Termistores
Humedad	Resistivos
Posicion	Mecanicos, Acelerometros
Presencia	Ultrasonidos, Infrarrojos
Distancia	Resistivos
Presion	Piezoelectricos
Caudal	Piezoelectricos
Calor	Celulas Termoelectricas
Quimicos	Detectores de gas y humos

Nota. Característica de los sensores. Adaptado de "Guía Práctica de Sensores" por A.Serna,F.Ros y J.Rico, 2010, p. 7.

Sensor de temperatura



Figura 6. Sensor de temperatura pt100. Adaptado de "Pt100 3 wire class B Resistance Thermometer, KNE Head with Transmitter" por labfacility, 2020.

Figura 6 Sensor de temperatura

1.3 Actuador

Los actuadores son dispositivos usados para manejar aparatos mecatrónicos, según Vildósola C, (Sin fecha) nos indica que un actuador es

Es un dispositivo inherentemente mecánico cuya función es proporcionar fuerza para mover o “actuar” otro dispositivo mecánico. La fuerza que provoca el actuador proviene de tres fuentes posibles: Presión neumática, presión hidráulica, y fuerza motriz eléctrica (motor eléctrico o solenoide). Dependiendo de el origen de la fuerza el actuador se denomina “neumático”, “hidráulico” o “eléctrico”.

Actuador neumatico



Figura 7 Piston neumatico. Adaptado de "Actuadores neumaticos" por FESTO, 2019.

Tabla 3
Tipos de actuadores

Tipos	Descripción
Mecánicos	Los actuadores mecánicos trabajan mediante la conversión de un tipo de movimiento a otro tipo de movimiento (Un movimiento lineal o rotativo).
Neumáticos	Su fuente de energía es el aire comprimido, el cual se utiliza para mover un pistón cuando el aire se libera o descomprime.
Hidráulicos	Los actuadores hidráulicos utilizan líquidos como aceite para generar movimientos lineales, rotativos u oscilatorios, se utilizan en aplicaciones donde se requiere de una fuerza inmensa.
Eléctricos	Los actuadores eléctricos son de los más limpios, fáciles de usar y de disponibilidad inmediata, debido a que no usan aceite ni requieren aire comprimido para operar.
Térmicos	Utilizan energía térmica o magnética para producir el movimiento deseado

Nota. Característica de los sensores. Adaptado de "Guía Práctica de Sensores" por A.Serna, F.Ros y J.Rico, 2010, p. 7.

1.4 Controlador

La parte primordial de un robot es su controlador, Rocha, (2020) nos indica que

Los controladores, son los cerebros que recogen toda la información del exterior recogida por los sensores y dependiendo del programa, realizan diferentes tareas, controlando el movimiento de los motores y definiendo el funcionamiento de los diferentes dispositivos de salida.

Programador Logico Controlable (PLC)



Figura 8. Programador Logico Controlable de la marca SIEMENS. Adaptado de "Controladores SIMTAC" por SIEMENS, 2019.

Figura 8 Programador lógico controlable

1.4.1 Programa

El programa es la lógica que se determina en el controlador, y este debe de seguir su proceso según lo programado, Rocha, (2020) nos indica que

El programa es el conjunto de instrucciones que permite que el equipo realice diferentes tareas o labores de forma autónoma. Para lograr el funcionamiento un equipo tiene algo similar a un cerebro, esto corresponde al controlador, el cual en su memoria contiene un programa, elaborado dependiendo de la función del equipo. Cada programa se diseña dependiendo de la utilidad y funcionalidad del proyecto o del equipo específico.

Para el desarrollo del programa de esta tesis, se utiliza el software **MATLAB**, ya que es de las principales herramientas para universidades y centros de investigación y desarrollo. Goering, (2014) Entre las principales características está el tratamiento de datos, uso de matrices, representación de datos y funciones, implementación de algoritmos, creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware.

MATLAB posee de dos herramientas primordiales para el desarrollo de esta investigación, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI).

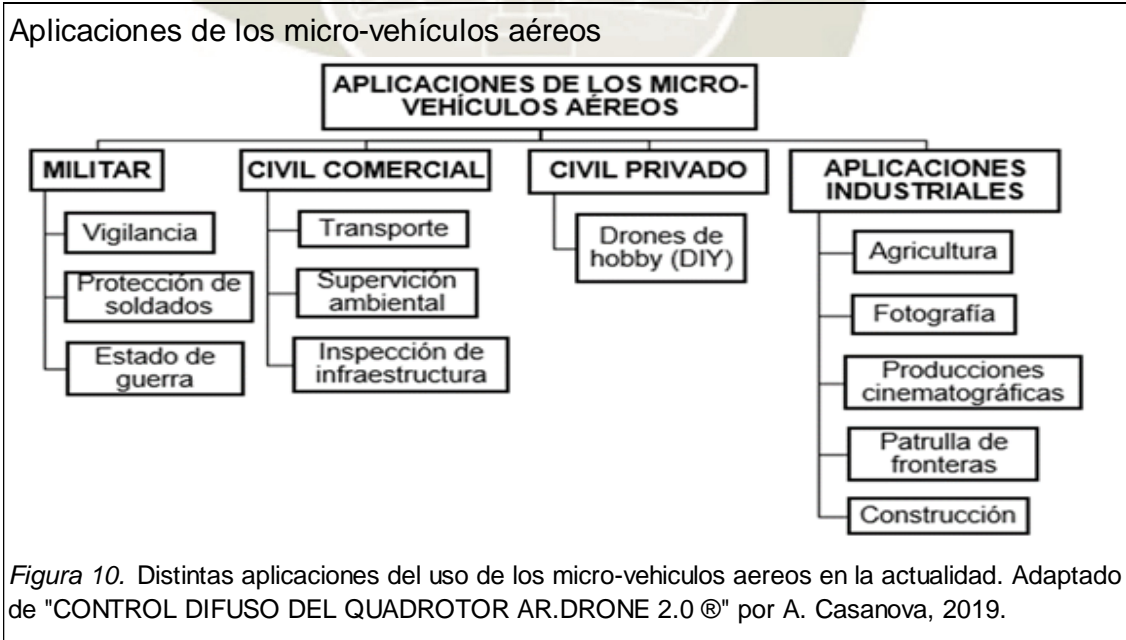
MATLAB



Figura 7 Logo de software MATLAB. Adaptado de "Matlab" por MATLAB, 2019.

2 Cuadróptero:

En la industria del micro vehículo aéreo, “The Economic Times” (2018) nos indica que se lograra alcanzar un estimado de \$ 21,47 billones para el final del 2021. Es por ello, que su estudio y aplicación se ha extendido a varias aplicaciones que pueden resumirse, ver figura 3.



Como se observa en la figura 10 existe una gran variedad de aplicaciones de vehículos aéreos, dentro de las aplicaciones industriales, podemos resaltar tres áreas:

- **Agricultura:** Inspección de los campos de forma remota y simplificada, sembrado de precisión, reconocimiento de frutos, hidratación de sembríos, fumigaciones de sembrío, etc.
- **Construcción:** Creación de mapas en 3 dimensiones, inspección de edificios, etc
- **Media:** Fotografía, filmación, etc.

En el caso de la utilización de cuadricópteros en las aplicaciones civiles comerciales, podemos resaltar un área:

- **Inspección de infraestructura:** Progreso de las construcciones, colocación de suministros y logística.

2.1 Denominaciones

Dentro de las denominaciones de los Micro vehículos aéreos, podemos encontrar 3 importantes:

- Vehículos aéreos sin piloto (UAV)
- Vehículos aéreos pilotados remotamente (RPV)
- Drones

2.2 Clasificación

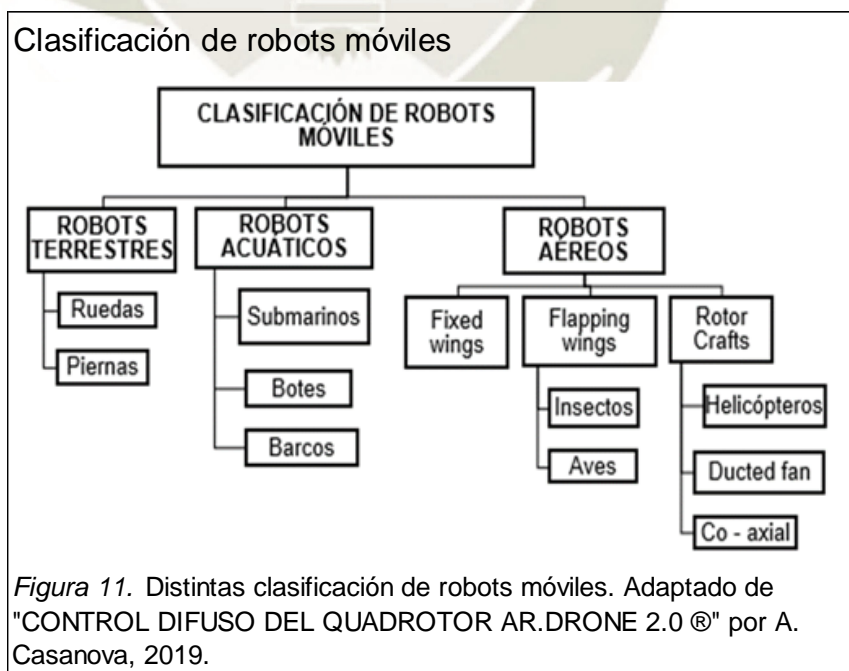
Según, Casanova de los Santos (2015) para precisar una clasificación correcta de la ubicación de los cuadricópteros dentro del mundo de la robótica, se puede usar como referencia la estructura.

Tabla 4
Clasificación de cuadricopteros

Clasificación	Tipo	Descripción
Estructura	Polis articulados	Son fijos y pueden mover sus elementos terminales en un determinado espacio de trabajo con un número limitado de grados de libertad.
	Moviles	Tienen la capacidad de desplazarse, basados en ruedas o alguna plataforma de propulsión.
	Androides	Reproducen total o parcialmente la forma y el comportamiento del ser humano.
	Zoomorfos	Imitan los sistemas de locomoción de los seres vivos, los cuales pueden ser caminadores o no caminadores.
	Hibridos	aquellos de difícil clasificación cuya estructura se sitúa en la combinación de algunas de las anteriores.

Nota. Clasificación de cuadricopteros por su estructura. Adaptación propia en base a la investigación realizada, 2019

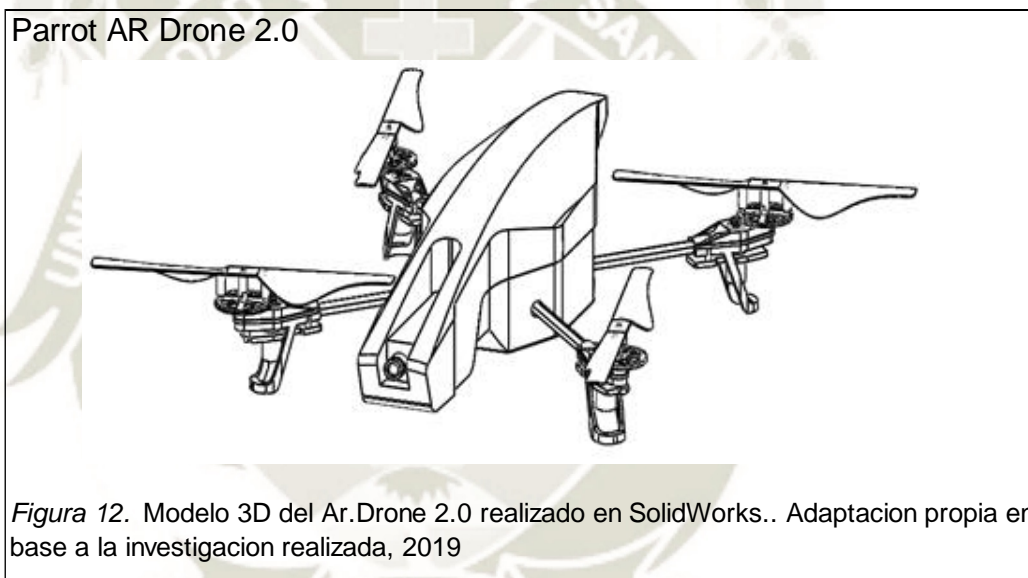
Nos centraremos en los robots móviles y cuya clasificación hasta llegar al cuadricoptero, podemos observarlo en el siguiente mapa conceptual, ver figura 11. En cual, los cuadricópteros pueden entrar dentro de los Rotor Crafts en el subgénero de Helicópteros.



Los cuadricópteros o drones se pueden concebir como un robot desarrollado por medio de 4 o más rotores fijados en una estructura o frame. Estos vehículos poseen hasta 6 grados de libertad por lo que podemos planear, hacer movimientos de traslación, rotación en los distintos ejes al igual que elevarnos y descender el vehículo.

2.3 Características del cuadricóptero:

El cuadricóptero usado para desarrollar el estudio es el modelo comercial “Parrot AR Drone 2.0” de la compañía Parrot.



2.3.1 Características mecánicas:

La estructura de un cuadricóptero se compone mecánicamente de 3 partes principales

➤ **Estructura o frame:** el cual es confeccionado según las necesidades que se requiera el drone. Los puntos principales que se requieren al seleccionar un frame son:

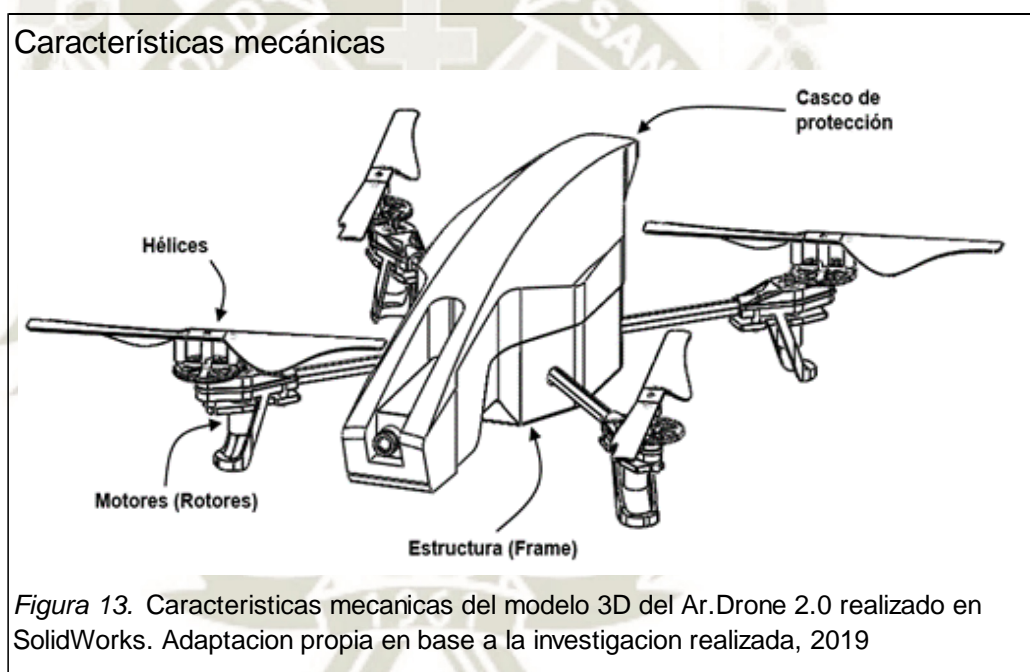
- Materiales de confección
- Peso del frame
- Dimensiones

- Cantidad de rotores

En el caso del Parrot AR Drone 2.0 podemos aumentar otras partes móviles como:

- **Casco de protección:** Casco de recubrimiento de la parte electrónica
- **Hélices:** Hélices de plástico de giro horario y antihorario

Según la figura 13, se puede apreciar los componentes mecánicos y partes móviles del Ar. Drone 2.0



Para una información más detallada, puede revisar Anexo 01 o Anexo 04

2.3.2 Características eléctricas:

Las partes eléctricas principales del Parrot AR Drone 2.0 pueden ser resumidas en las siguientes:

- **Batería LiPo:** Así Mobus, (2016) refiere “Una batería LiPo (Litio y polímero) es un tipo de batería recargable muy habitual en el mundo de los

multirrotores. Nacen como una opción aceptable a la utilización de combustibles para realizar vuelos” (“¿Qué es una batería LiPo?”, párr. 1).

Bateria Lipo



Figura 14. Bateria Lipo del Ar.Drone 2.0 . Adaptacion propia en base a la investigacion realizada, 2019

Las baterías que se utiliza son baterías de 1000mAh de 11.1V (Operación a toda carga de 12.5V hasta carga baja de 9V)

- **Sensor de Ultrasonido:** Los sensores de ultrasonidos o sensores ultrasónicos según KEYENCE CORPORATION, (2020) “miden la distancia mediante el uso de ondas ultrasónicas. El cabezal emite una onda ultrasónica y recibe la onda reflejada que retorna desde el objeto.” (“¿Qué es un sensor ultrasónico?”, párr. 1).

Sensor ultrasonico



Figura 15. Sensor ultrasonico. Adaptado de "Sensor De Ultrasonido HC-SR04 rango de sensado de 3-300cm" por electropro, 2019.

- **Acelerómetro** Omega Engineering, (2020) nos indica que un acelerómetro es un dispositivo que “mide la vibración o la aceleración del movimiento de una estructura. La fuerza generada por vibraciones o cambios en el movimiento (aceleración) hace que la masa "comprima" el material

piezoeléctrico, generando carga eléctrica que es proporcional a la fuerza.”
(¿“Acelerómetro?”, párr. 1).

Acelerometro



Figura 16 Acelerometro . Adaptado de "Acelerómetro / Giroscopio [MPU6050]" por nomada-e, 2020.

- **Giroscopio** Ro-botica Global, (2017) nos indica que “es un dispositivo que mide las velocidades angulares en los 3 ejes, este sensor calcula hacia que lado se produce la inclinación, su principal uso es mantener equilibrado y controlado el balanceo del equipo” (“Sensor giróscopo - Premium”, párr. 1).

Giroscopio



Figura 17 Sensor giroscopio . Adaptado de "Sensor giróscopo - Premium" por ro-botica, 2017.

Figura 17 Sensor Giroscopio

Motores: Mobus, (2016) nos indica que “un motor Brushless o motor sin escobilla, es el tipo de motor utilizado por los multirrotores para mover sus hélices. Su principal ventaja contra motores eléctricos de escobillas es que no se produce tanto desgaste en su funcionamiento” (“Motores Brushless para drones, todo lo que necesitas saber”, párr. 1).

Motor Brushless



Figura 18 Motor Brushless . Adaptado de "Motores Brushless para drones, todo lo que necesitas saber" por mobus, 2017.

Para una información más detallada, puede revisar Anexo 01 o Anexo 04

2.3.3 Teoría de vuelo:

Dentro de las configuraciones de vuelo de los cuadricópteros solo se pueden tener dos configuraciones:

- Cuadricópteros Plus
- Cuadricópteros X

La configuración usada en caso del Parrot AR Drone 2.0 es la correspondiente al Cuadricóptero X. Para dicho modelo, podemos determinar tres ejes principales de vuela que se muestra en la siguiente figura:

Ejes de vuelo

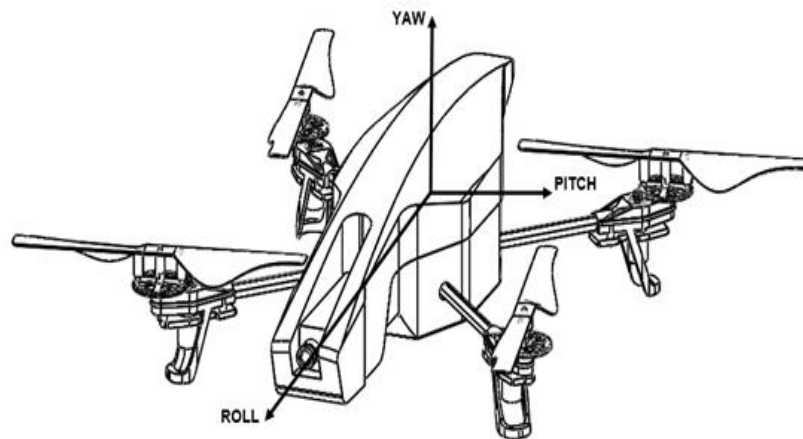


Figura 19. Ejes de vuelo en el sistema cartesiano. Adaptación propia en base a la investigación realizada, 2019

Ejes de vuelo

Cada eje nos permite un tipo de rotación para la orientación espacial del cuadricóptero. Estos movimientos son representados en las siguientes figuras.

El **eje vertical** es un:

Eje imaginario que, es perpendicular a los ejes transversal y longitudinal. Este eje está contenido en un plano que pasa por el centro de gravedad desde arriba hacia abajo.

El movimiento que realiza el avión alrededor de este eje se denomina **guiñada o Yaw**. (Ejes del avión, 2010, párr.9)

Rotacion en YAW

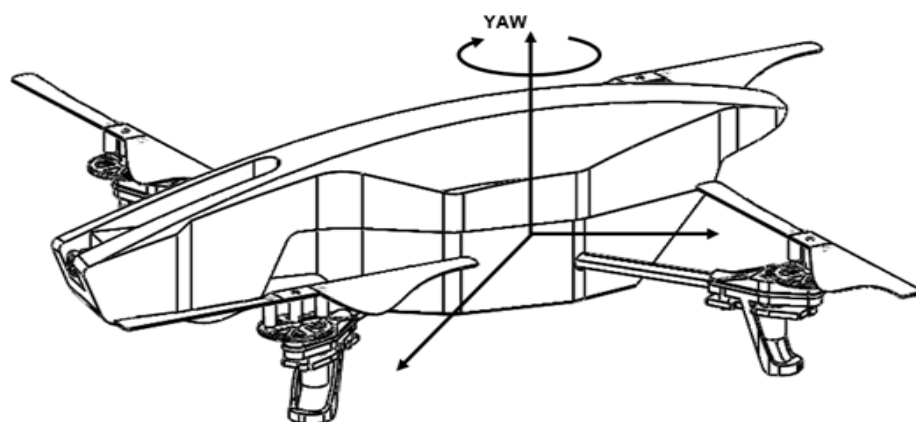
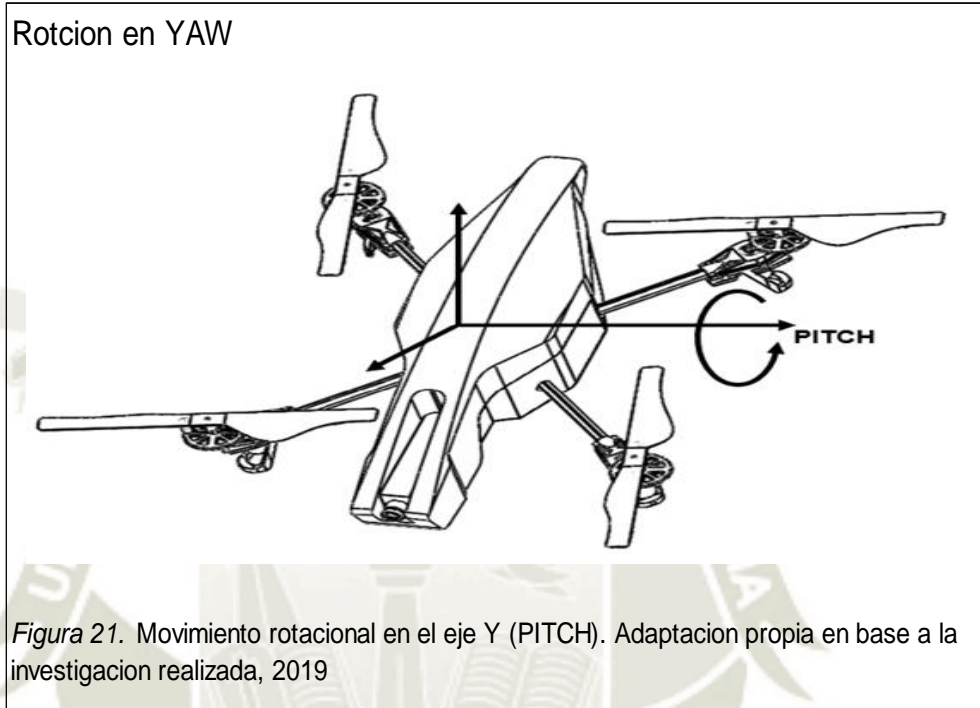


Figura 20. Movimiento rotacional en el eje Z (YAW). Adaptación propia en base a la investigación realizada, 2019

El eje **lateral** o **transversal** es un

Eje imaginario que se extiende de punta a punta en los lados laterales. El movimiento que realiza el equipo alrededor de este eje se denomina **cabeceo** o **Pitch**. (Ejes del avión, 2010, párr.2)



El **eje longitudinal** es un:

Eje imaginario que se extiende desde la punta a la cola del avión. El movimiento que realiza el avión alrededor de este eje se denomina **alabeo**, **balanceo** o **Roll**. (Ejes del avión, 2010, párr.5)

Rotcion en YAW

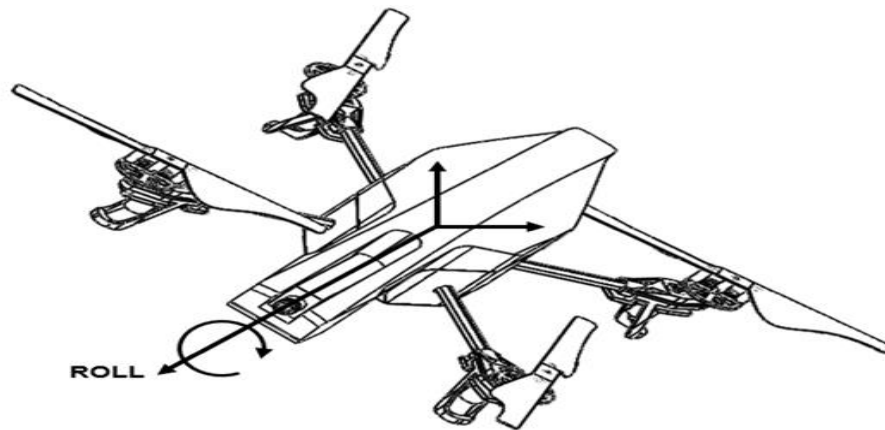


Figura 22. Movimiento rotacional en el eje x (ROLL). Adaptacion propia en base a la investigacion realizada, 2019

Rotacion en el eje Roll

De la misma manera, existe otra variable que actúa en el sistema de vuelo que es la fuerza de empuje que debe ser mayor a la masa del sistema y la gravedad juntas para poder despegar.

Fuerza de empuje vs peso

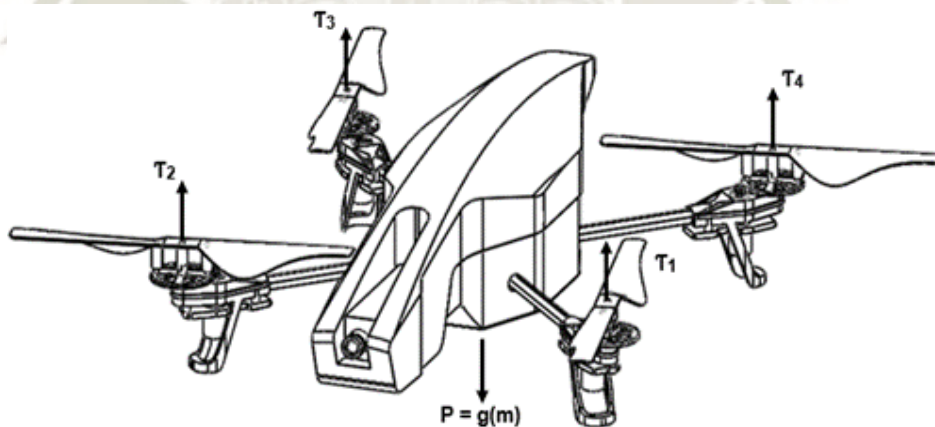


Figura 23. Fuerza de empuje vs peso. Adaptacion propia en base a la investigacion realizada, 2019

Por lo que podemos obtener que:

$$P = m(g) < T_1 + T_2 + T_3 + T_4$$

Ecuación 1 Relación peso – fuerzas

Donde:

- P: Peso del cuadricóptero (N)
- g: Aceleración de la gravedad (m/s^2)
- m: Masa del cuadricóptero (kg)
- T_i : Fuerza de empuje del motor (N)

Configuraciones de vuelo:

El Ar. Drone 2.0 tiene dos formas de uso o configuraciones de vuelo, cada configuración posee su propia carcasa de vuelo:

- **Outdoor:** Para vuelos en exterior con una estructura más ligera recomendado para corrientes de aire pequeñas o de baja intensidad
- **Indoor 6:** Equipado con una estructura externa más pesada, permite al cuadricóptero proteger al modelo de colisiones con objetos de un entorno cerrado.

Estos modelos se muestran en las siguientes figuras:

Configuración outdoor e indoor

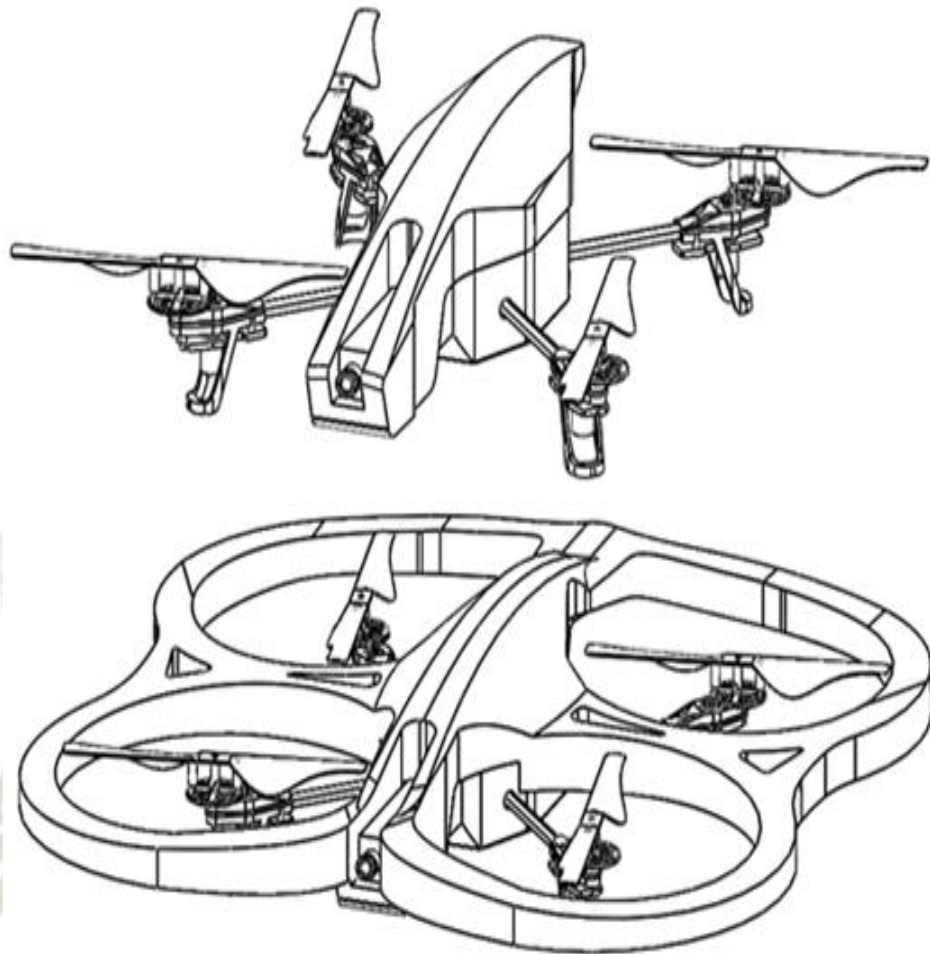


Figura 24. Modelo empleado para exteriores. Adaptación propia en base a la investigación realizada, 2019

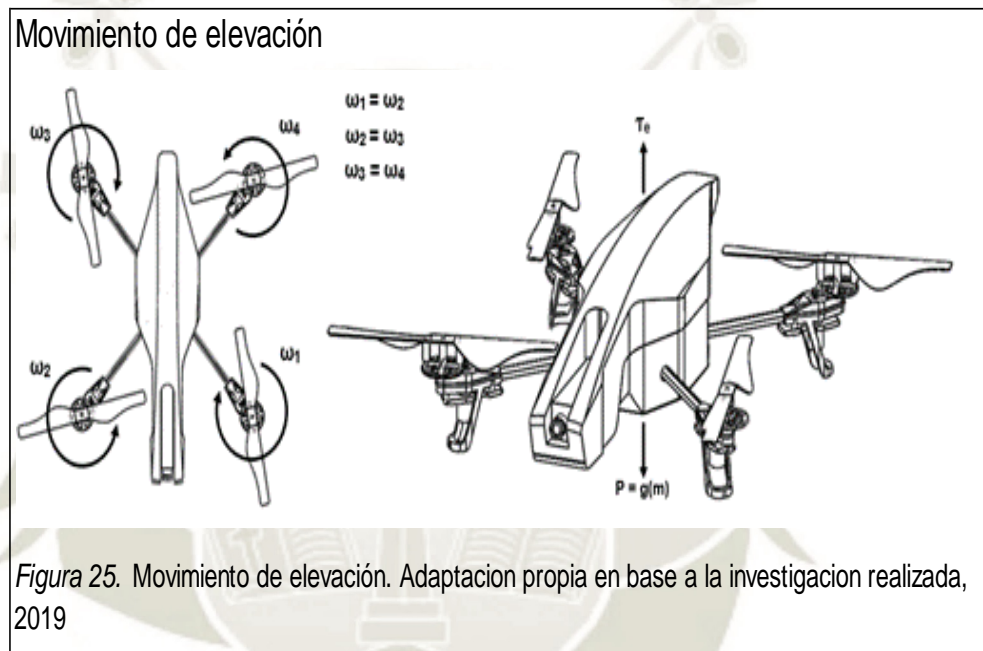
Movimientos:

En este punto se explicara los movimiento basicos que puede generar el cuadricoptero en el entorno. Estos movimientos tienen relacion a la cantidad de entradas que pueden ser controladas

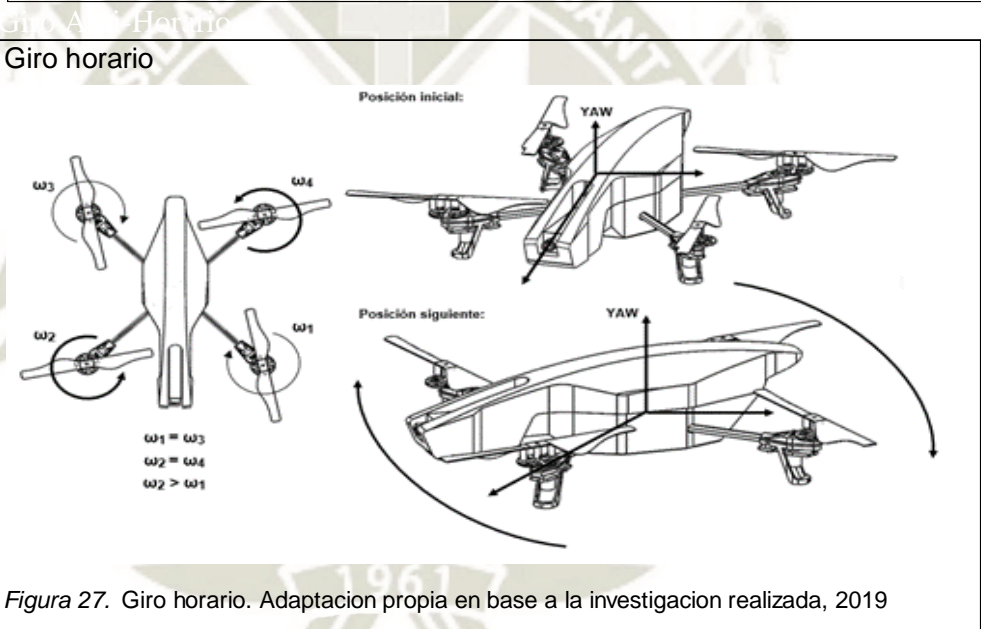
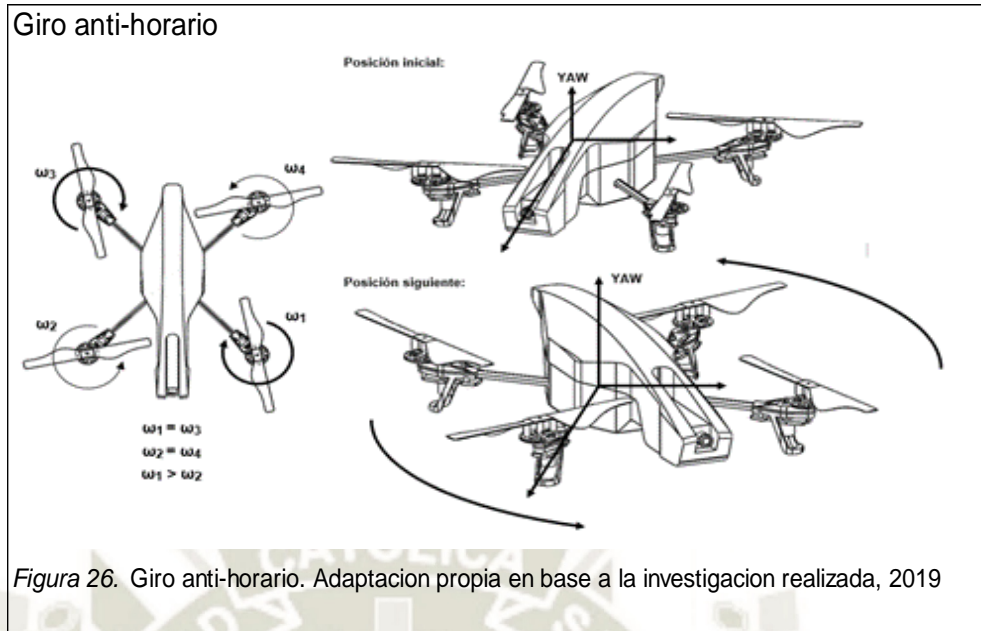
- Elevación en el eje Z
- Rotación en el eje Z (Yaw)

- Rotación en el eje Y (Pitch)
- Rotación en el eje X (Roll)

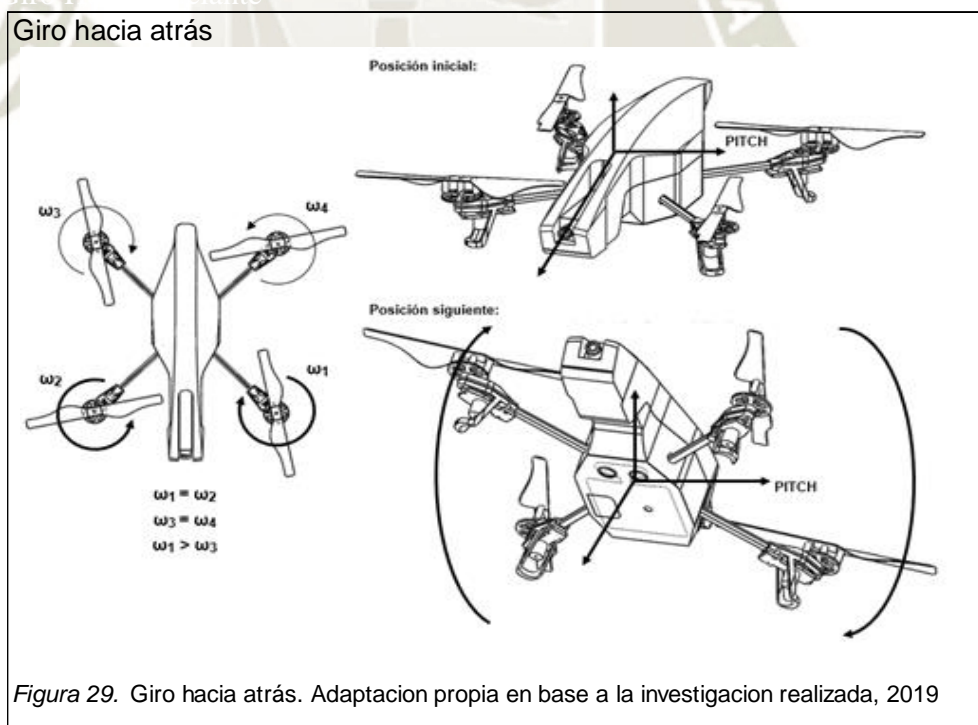
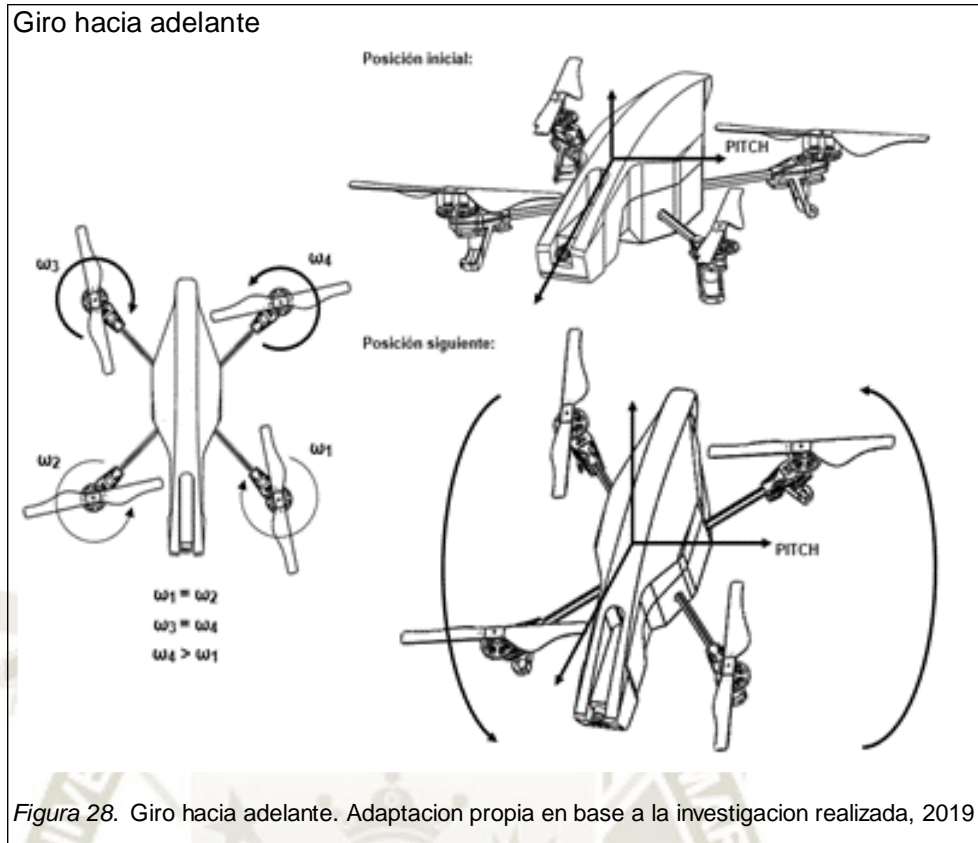
Para poder tener un movimiento de elevación, la fuerza de empuje generada deberá ser mayor al peso del sistema y la carga que esta lleve. Además, a mayor velocidad angular de los motores, será también mayor el empuje generado.



Para el caso del giro en el eje Yaw, el par de motores contrarios deberán tener la misma velocidad angular, a la vez que la velocidad angular del segundo par debe ser menor al anterior. Dependiendo del par de motores con mayor velocidad angular permitirá tener un giro horario o anti-horario.



Para el caso del giro en el eje Pitch, el par de motores delanteros deberán tener la misma velocidad angular, a la vez que la velocidad del par de motores traseros deberá ser menor al anterior. Dependiendo del par de motores con mayor velocidad angular permitirá girar hacia adelante o atrás.



Para el caso del giro en el eje Roll, el par de motores adyacentes deberán tener la misma velocidad angular, a la vez que la velocidad del segundo par de

motores deberá ser menor al anterior. Dependiendo del par de motores con mayor velocidad angular permitirá girar hacia la derecha o izquierda.

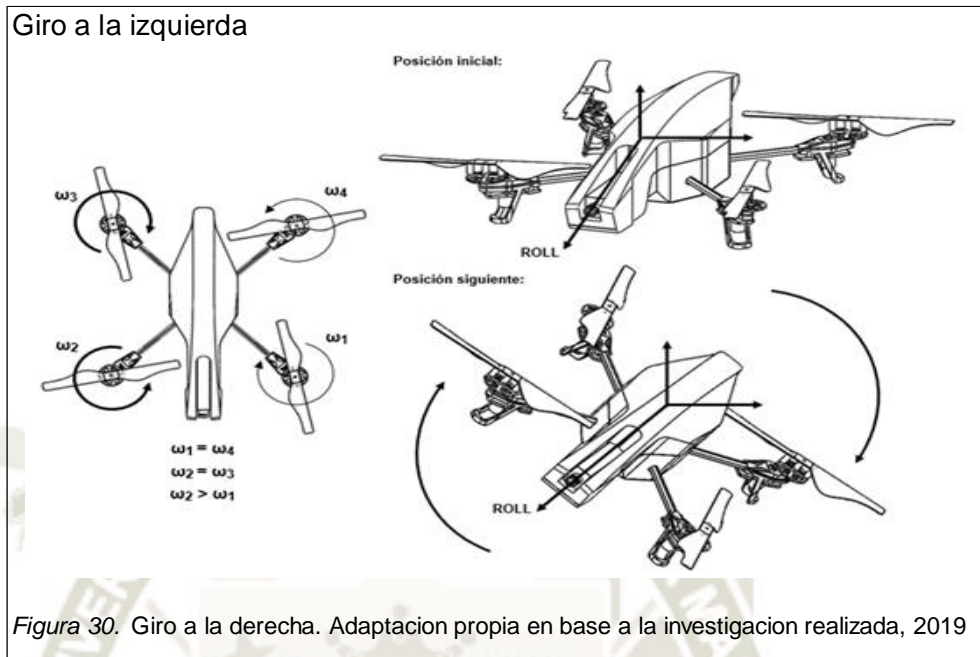


Figura 30. Giro a la derecha. Adaptacion propia en base a la investigacion realizada, 2019

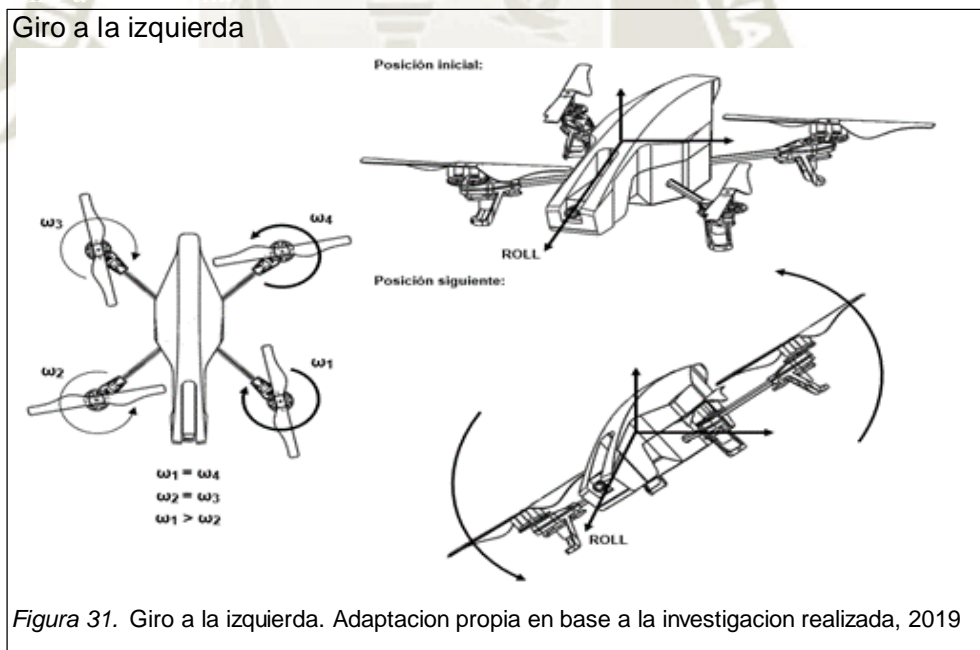
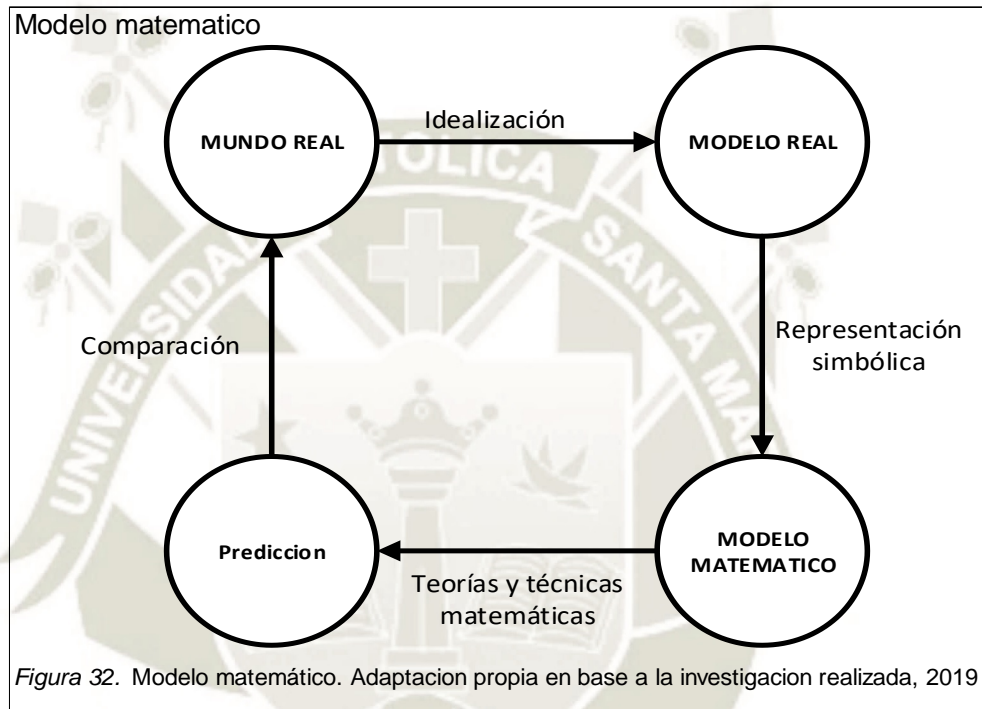


Figura 31. Giro a la izquierda. Adaptacion propia en base a la investigacion realizada, 2019

Para una información más detallada, puede revisar el plano “CUAD – CGEN - 01” o el “AR Drone Developer Guide”.

3 Modelo Matemático

Como parte inicial del desarrollo de esta tesis se necesita obtener el modelo matemático del cuadricoptero. Guts, Frolova, y Páutova (2013) refiere que mediante el uso de diversos instrumentos (símbolos y operaciones) se obtiene un esquema idealizado y simplificado del mismo, constituido en la ciencia matemática



Para el desarrollo de modelos matemáticos se debe de tener en consideración los aspectos mencionados en la tabla 5:

Tabla 5
Fases para la construcción de un modelo

Fases	Descripcion
Identificación	De la planta que necesita ser simulada y controlada
Tipo de modelo	Donde se determina que tipo de sistema se esta analizando, tomando en cuenta los datos de entradas, salidas y factores externos que afectan a la planta
Formulacion	Del modelo donde se detalla que algoritmos y herramientas matematicas a usar.
Comparacion	De los resultados obtenidos con el mundo real

Nota. Fases para la construcción de un modelo. Adaptado de "Métodos matemáticos en la sociología" por A. K. Guts, Yu. V. Frolova, L. A. Páutova, 2013, Editorial URSS Moscú

No siempre una modelización a detalle es la más práctica, ya que en algunas ocasiones puede ser demasiado complicado el tratar de resolverlo, a pesar del alto grado de idealización y simplificación

4 Módulos experimentales

Los módulos experimentales son plataforma diseñadas e implementadas para la experimentación de proyectos de desarrollo, los cuales brindan una forma de comprobación de teorías científicas, elementos computacionales, y otras nuevas tecnologías.

Es un método para probar un módulo particular en forma aislada. Puede ser implementado como un entorno de pruebas, pero no necesariamente con el propósito de verificar seguridad.

En el desarrollo de este trabajo, se utilizará dos módulos experimentales para realizar la comprobación del diseño de los distintos módulos de control (PID y FUZZY). Los módulos experimentales están implementados dependiendo de las variables a controlar, y es por eso las distintas características de cada módulo.

El principal motivo de la utilización de módulos experimentales en el desarrollo de este proyecto, fue por la seguridad y preservación del cuadricoptero, ya que, al no restringir los movimientos de este, no se podría tener varias interacción sin generar alguna colisión con el cuadricoptero y generar algún daño en su estructura o en sus componentes, por este motivo se desarrollaron 2 módulos experimentales (HOVER-ESTABILIZACION), que nos aseguran las pruebas sin ninguna

Modulo experimental 1

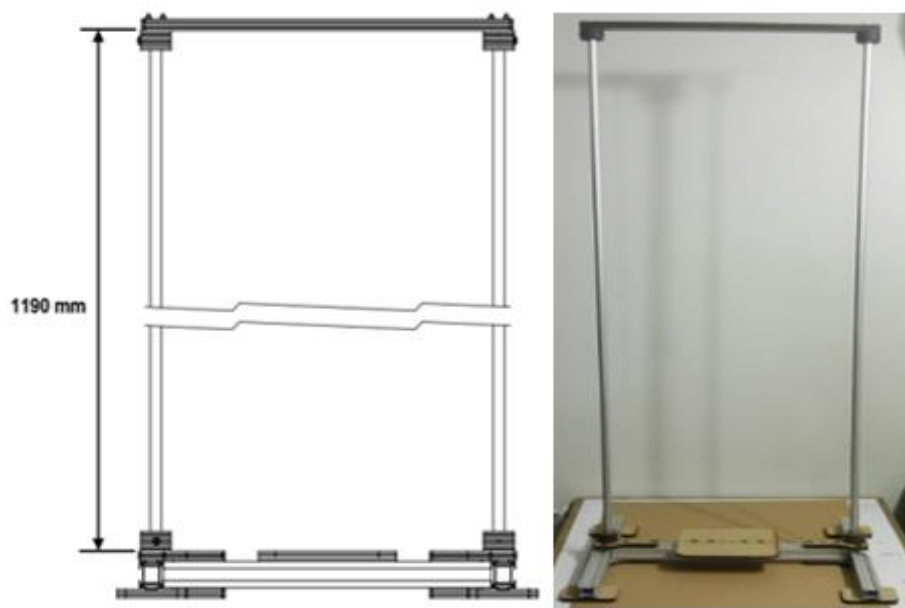


Figura 33. Modulo experimental utilizado en el desarrollo del proyecto . Adaptacion propia en base a la investigacion realizada, 2019

Modulo experimental 2

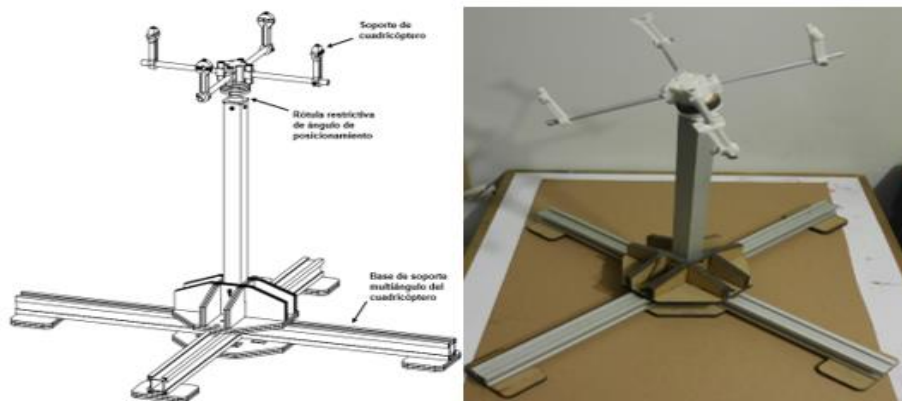


Figura 34. Modulo experimental utilizado en el desarrollo del proyecto . Adaptacion propia en base a la investigacion realizada, 2019

5 Sistema de control a implementar en el proyecto

5.1 Control PID

El controlador PID es la técnica de control más usada en las industrias. Es un controlador sencillo y con un buen rendimiento en una amplia gama de procesos.

$$y(t) = K_p * e(t) + K_i * \int e(t) dt + K_d * \frac{d e(t)}{dt}$$

Ecuación 2 Control PID

La señal de control es la suma de tres factores:

- KP: proporcional, según Pardo, (Sin fecha) el valor es proporcional al error de control (error en el presente). La principal característica del control proporcional es la reducción del error del sistema.

Aumentar la acción proporcional K_p tiene los siguientes efectos:

- Aumenta la velocidad de respuesta del sistema.
- Disminuye el error del sistema en régimen permanente.
- Aumenta la inestabilidad del sistema.

Al emplear un elevado K_p Ogata, (2010) nos dice que:

Los dos primeros efectos son positivos y deseables. El último efecto es negativo y hay que intentar minimizarle. Por lo tanto al aumentar la acción proporcional existe un punto de equilibrio en el que se consigue suficiente rapidez de respuesta del sistema y reducción del error, sin que el sistema sea demasiado inestable. Aumentar la acción proporcional más allá de este punto producirá una inestabilidad indeseable. Reducir la acción proporcional, reducirá la velocidad de respuesta del sistema y aumentará su error permanente.(p. 567).

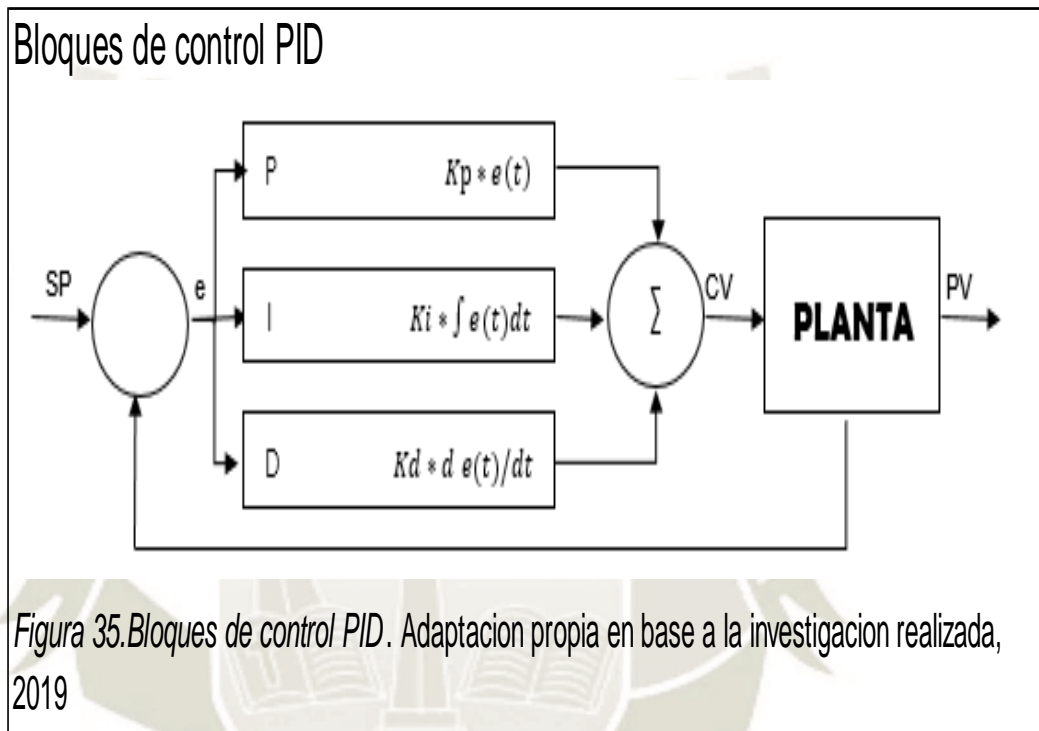
- K_D : derivativa, según Pardo, (Sin fecha) es proporcional a la derivada del error, La derivada del error es otra forma de llamar a la "velocidad" del error . Esta acción de control servirá por lo tanto para estabilizar una respuesta que oscile demasiado..

Aumentar la constante de control derivativa K_d tiene los siguientes efectos:

- Aumenta la estabilidad del sistema controlado.
 - Disminuye un poco la velocidad del sistema.
 - El error en régimen permanente permanecerá igual.
- K_I : integral, según Pardo, (Sin fecha) es proporcional a la integral del error de control. La integral se puede ver como la acumulación de la señal de error. Esta acción de control servirá para disminuir el error en régimen permanente.

Aumentar la acción integral K_i tiene los siguientes efectos:

- Disminuye el error del sistema en régimen permanente.
- Aumenta la inestabilidad del sistema.
- Aumenta un poco la velocidad del sistema.



5.1.1 Sintonía de controladores PID.

Para realizar una sintonización del controlador, hay que tener un mínimo de conocimiento sobre los efectos de las tres variables y, de ser posible, tener experiencia sobre el proceso a controlar. Con esta experiencia se puede seleccionar el control para la planta, ya sea proporcional (P), proporcional-integral (PI), proporcional-derivativo (PD), o PID.

La sintonización de los parámetros PID es frecuente en muchos procesos industriales. Si es posible obtener un modelo matemático de ella, se pueden aplicar diversas técnicas de diseño de controladores con el fin de seleccionar los mejores parámetros de K_p , K_d y K_i .

El empleo de técnicas experimentales es de gran ayuda debido a que, en la mayoría de las industrias, conseguir un modelo de la planta es demasiado compleja para extraer un modelo matemático exacto.

Estas son:

- Ensayos de prueba y error.
- Sintonía experimental en lazo abierto
- Sintonía experimental en lazo cerrado.
- Sintonía experimental con especificaciones frecuenciales.

Para una sintonía PID experimental se debe de considerar los tres tipos de acciones de control (proporcional, integral y derivativa). La opción que proporcionará mejores respuesta es la que se utilizara para implementar el control. Para desarrollar la sintonización experimental es necesario realizar los siguientes procedimientos:

➤ **Acción Proporcional.**

Ir incrementando poco a poco el valor de la acción proporcional para disminuir el error (diferencia entre el SP y el PV) Si se alcanza la respuesta deseada en velocidad y error, el valor proporcional del control PID ya está sintonizado.

Si el sistema se vuelve inestable antes de conseguir la respuesta deseada, se debe aumentar la acción derivativa.

➤ **Acción Derivativa.**

Si el sistema es demasiado inestable, se aumentará poco a poco el valor derivativo K_d para conseguir de nuevo estabilidad en la respuesta.

➤ **Acción Integral.**

Cuando el error del sistema es mayor que el deseado, se aumentará el valor integral K_i hasta que el error se reduzca con la rapidez deseada.

Si se vuelve inestable, se debe aumentar la acción derivativa.

Con estas sencillas reglas es sencillo afinar poco a poco el controlador PID hasta conseguir la respuesta deseada.

5.2 Control Difuso

Así, Wang, (2006) nos indica que la lógica difusa es:

El concepto de lógica difusa es muy común, está asociado con la manera en que las personas perciben el medio, por ejemplo, ideas relacionadas con la altura de una persona, velocidad con la que se mueve un objeto, la temperatura dominante en una habitación, cotidianamente se formulan de manera ambigua y depende de quien percibe el efecto físico o químico, será su enunciado acerca de tal fenómeno. Una persona puede ser alta o baja, algo puede moverse rápido o lento, una temperatura puede ser baja o moderada o alta, se dice que estas afirmaciones acerca de una variable son ambiguas porque rápido, bajo, alto son afirmaciones del observador, y estas pueden variar de un observador a otro. Uno se puede preguntar cuándo algo es frío o caliente, que tan baja es la temperatura cuando decimos frío, o que tan alta es cuando decimos caliente.

Los conjuntos difusos definen justamente estas ambigüedades, y son una extensión de la teoría clásica de conjuntos, donde un elemento pertenece o no a un conjunto, tal elemento tiene solo 2 posibilidades, pertenecer o no, un elemento es bi-valuado y no se definen ambigüedades. Con conjuntos difusos se intenta modelar la ambigüedad con la que se percibe una variable. Los conjuntos

Cuadro de control de reglas difuso

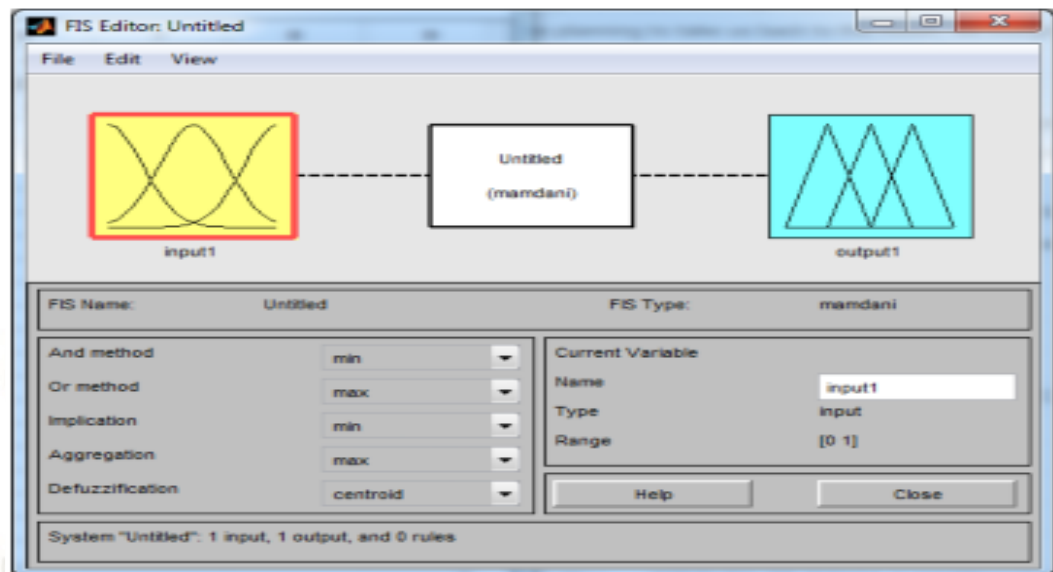


Figura 37. Cuadro de control de reglas difuso. Adaptación propia en base a la investigación realizada, 2019

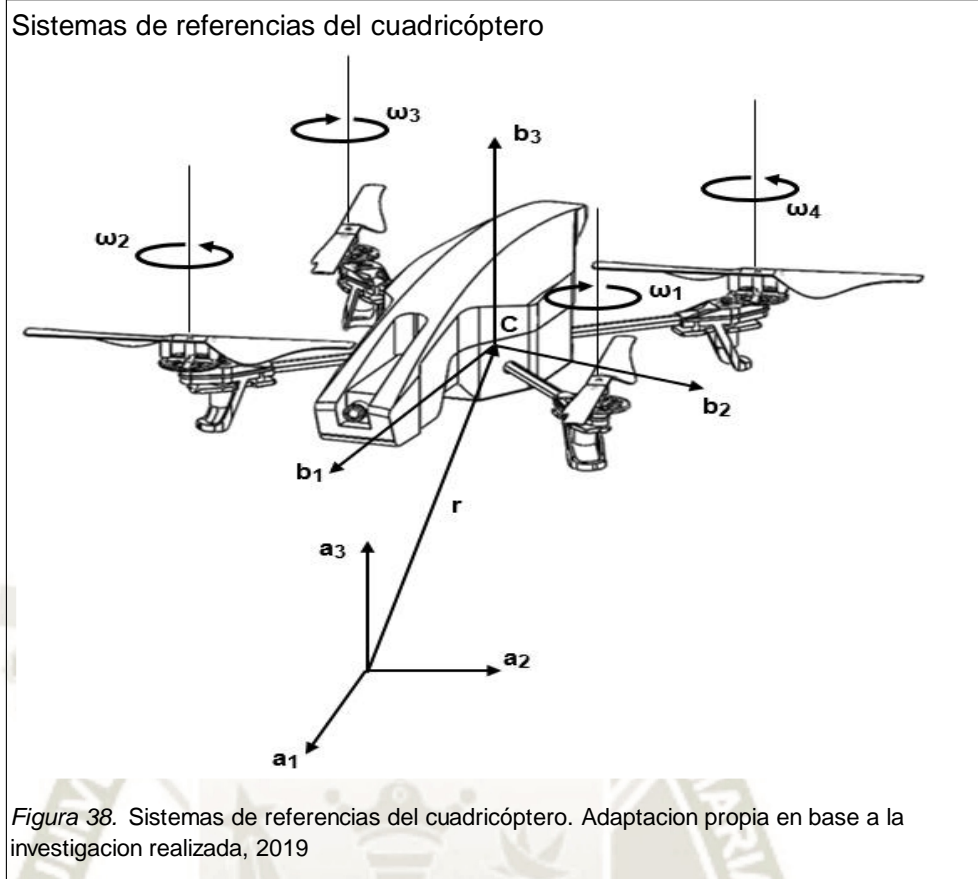
Cuadro de control de reglas difuso

6 Modelo matemático:

Para obtener un modelo matemático correspondiente se debe de analizar tanto la cinemática como la dinámica del dron. En el proyecto se utilizó el equipo Ar. Drone 2.0, debido a que es un equipo de uso comercial, con un costo no tan elevado y es un equipo práctico para la implementación de distintos tipos de control. **Ver Anexo 01** para mayor detalle

6.1 Modelo cinemático

Para el modelamiento y obtención de las ecuaciones del sistema, se definirán sistemas de referencia al que estará sujeto el cuadricoptero. Se establecerá un sistema fijo **A** y un marco móvil inercial **B** con origen en el centro de masas del cuadricoptero. Para obtener la ubicación del cuadricoptero en el espacio, es necesario conocer su posición y orientación respecto al sistema fijo.



3.8 Sistemas de referencias del cuadricóptero

- El eje X hace referencia al eje **a1** en el sistema **a**.
- El eje Y hace referencia al eje **a2** en el sistema **a**.
- El eje Z hace referencia al eje **a3** en el sistema **a**.

En coordenadas de traslación, la posición será representada por el vector **r**.

$$r = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^3$$

Ecuación 3 Vector de posición.

La orientación es obtenida mediante los ángulos de Euler respecto al sistema **b**, la cual indica la rotación que experimento el cuadricoptero en el espacio. Los ángulos de Euler serán representados por el vector η

$$\eta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \in \mathbb{R}^3$$

Ecuación 4 Vector de orientación

- El Angulo ϕ es la rotación respecto al eje X y se denomina Roll o cabeceo
- El Angulo θ es la rotación respecto al eje Y y se denomina Pitch o inclinación
- El Angulo ψ es la rotación respecto al eje Z y se denomina Yaw o vuelco

Para describir la orientación en robótica, se emplean las matrices de rotación, debido a la facilidad de la utilización del algebra matricial. Se emplea la matriz de rotación de Z – Y – X, como se muestra en la ecuación 5. (Barrientos, 1996)

$$R_A^B = R(z, \psi) R(y, \theta) R(x, \phi).$$

Ecuación 5 Matriz de Rotación Roll – Pitch – Yaw

Para obtener dicha matriz, primero expresaremos las matrices de rotación de Roll – Pitch – Yaw.

$$R(x, \phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\text{sen } \phi \\ 0 & \text{sen } \phi & \cos \phi \end{bmatrix}$$

$$R(y, \theta) = \begin{bmatrix} \cos \theta & 0 & \text{sen } \theta \\ 0 & 1 & 0 \\ -\text{sen } \theta & 0 & \cos \theta \end{bmatrix}$$

$$R(z, \psi) = \begin{bmatrix} \cos \psi & -\text{sen } \psi & 0 \\ \text{sen } \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ecuación 6 Raw-Pitch-Yaw

$$R_A^B = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix}$$

$$R_A^B = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix}$$

Ecuación 7 Ecuación de Euler ZYX

Para reducir la ecuación, se representará desde ahora $\sin \psi$ como $S\psi$ y $\cos \psi$ como $C\psi$ y sus equivalentes en los otros ejes correspondientes. De esta manera la matriz de rotación será representado como:

$$R_A^B = \begin{bmatrix} C\psi C\theta & C\psi S\theta S\phi - S\psi C\phi & C\psi S\theta C\phi + S\psi S\phi \\ S\psi C\theta & S\psi S\theta S\phi + C\psi C\phi & S\psi S\theta C\phi - C\psi S\phi \\ -S\theta & C\theta S\phi & C\theta C\phi \end{bmatrix}$$

Ecuación 8 Matriz de Rotación ZYX Simplificada

6.2 Modelo dinámico

El modelo dinámico de un sistema define la relación que existe entre las fuerzas y el movimiento que generan al sistema (Barrientos, 1996).

Se tomó las siguientes consideraciones para obtener un modelado del sistema adecuado

- Se considera una estructura rígida
- El centro de gravedad coincide con el centro del sistema de referencia inercial
- Las hélices se consideran rígidas

Comenzaremos usando un modelo de referencia de Fuerzas y Momentos ver figura 39.

Modelo de fuerzas y momentos

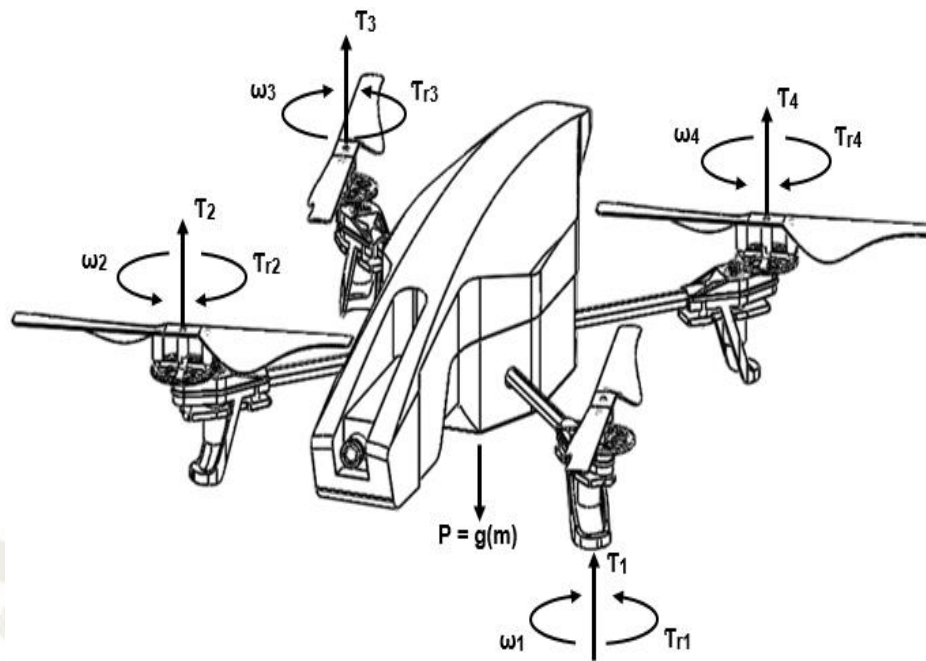


Figura 39. Modelo de fuerzas y momentos. Adaptación propia en base a la investigación realizada, 2019

Modelo de Fuerzas y Momentos

Podemos definir la fuerza de empuje como

$$T_i = K_f(\omega^2)$$

Ecuación 9 Empuje aerodinámico

Donde:

- T_i : Fuerza de empuje (N)
- K_f : Constante de fuerza contra-electromotriz
- ω : Velocidad angular (RPM)

En caso de la velocidad del motor, podemos formular lo siguiente:

$$K_f(\omega^2) = \frac{1}{4} m(g)$$

Ecuación 10 Empuje individual aerodinámico

Donde:

- K_f : Constante de fuerza contra-electromotriz
- ω : Velocidad angular (RPM)
- g : Aceleración de la gravedad (m/s^2)
- m : Masa del cuadricóptero (kg)

Para el torque del motor podemos formular:

$$M_i = K_m(\omega^2)$$

Ecuación 11 Momento aerodinámico

Donde:

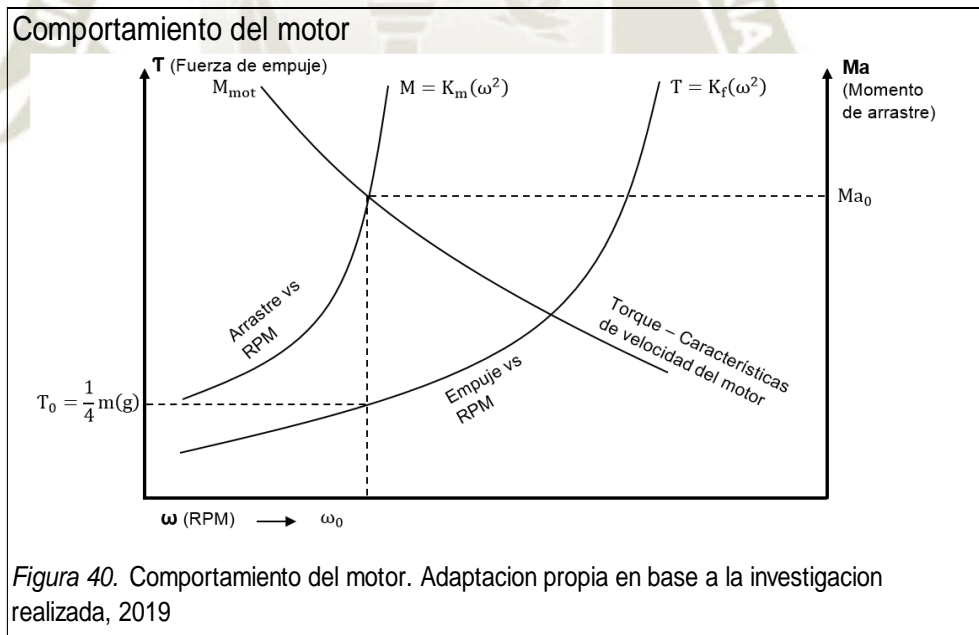
- K_m : Constante de torque electromagnético
- ω : Velocidad angular (RPM)
- M_i : Torque del motor (N-m)

Debemos recordar que el modelo al estar en equilibrio se puede plantear de la siguiente manera:

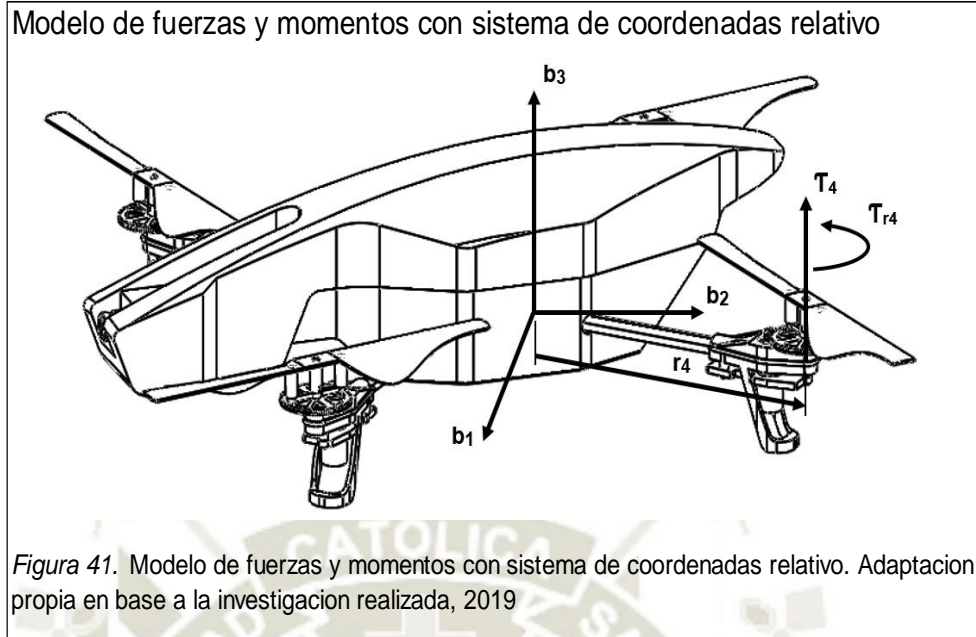
$$T_e = T_1 + T_2 + T_3 + T_4 + m(g)$$

Ecuación 12 Fuerzas en equilibrio (HOVER)

Podemos resumir el comportamiento del motor usando la siguiente gráfica



Podemos plantear un nuevo sistema de coordenadas relativo para un modelo de Fuerzas y Momentos como el que se observa en la figura 27.



Modelo de Fuerzas y Momentos con sistema de coordenadas relativo

Al estar en equilibrio el modelo de momentos se puede plantear de la siguiente manera:

$$M = T_1(r_1) + T_2(r_2) + T_3(r_3) + T_4(r_4) + T_{r1} + T_{r2} + T_{r3} + T_{r4}$$

Ecuación 13 Momentos En Equilibrio

Donde:

- T_i : Fuerza de empuje (N)
- T_{ri} : Momento de arrastre (N-m)
- r_i : Distancia entre eje de coordenadas y eje de giro del motor (m)

6.2.1 Ecuaciones de Euler-Lagrange

Para obtener la dinámica del sistema se representarán las coordinas generales del quadricoptero por q

$$q = \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix} \in \mathbb{R}^6$$

Ecuación 14 Coordenada generalizada del quadricóptero

La ecuación de Euler-Lagrange se basa en la suma de la energía cinética y de la energía potencial (Barrientos, 1996)

$$T_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i}$$

Ecuación 15 Euler-Lagrange

Donde:

- L es la función lagrangiana $L = K - U$
- K es la energía cinética
- U es la energía potencial
- q_i es la coordenada generalizada
- \dot{q}_i es la primera derivada de la coordenada generalizada respecto al tiempo
- T_i es el vector de fuerzas y pares aplicados en q_i

Primero debemos de obtener la función Lagrangiana:

$$L = K_{\text{tras}} + K_{\text{rot}} - U$$

Ecuación 16 Función Lagrangiana

Donde:

$$U = m(g)(Z)$$

Ecuación 17 Energía potencial

$$K_{\text{tras}} = \frac{m}{2} \dot{\mathbf{v}}^T \dot{\mathbf{v}}$$

Ecuación 18 Energía cinética en traslación

$$K_{\text{rot}} = \frac{J(\eta)}{2} \dot{\eta}^T \dot{\eta}$$

Ecuación 19 Energía cinética en rotación

Donde $J(\eta)$ es el Jacobiano en las coordenadas globales del sistema de referencias fijo

$$J(\eta) = W_{\eta}^T (I) (W_{\eta})$$

Ecuación 20 Jacobiana Global

Donde W_{η} es el Jacobiano respecto al sistema de referencias local, que relaciona la velocidad angular con el vector de velocidades de Euler; (Reinosos, 2014)

$$W_{\eta} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \cos \theta \sin \phi \\ 0 & -\sin \phi & \cos \theta \cos \phi \end{bmatrix}$$

Ecuación 21 Jacobiana Local

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$$

Ecuación 22 Matriz de inercias

Sustituyendo:

$$J(\eta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \cos \theta \sin \phi \\ 0 & -\sin \phi & \cos \theta \cos \phi \end{bmatrix}$$

$$J(\eta) = \begin{bmatrix} I_{xx} & 0 & -I_{xx} S\theta \\ 0 & I_{yy} C^2\phi + I_{zz} S^2\phi & (I_{yy} - I_{xx})C\phi S\phi C\theta \\ -I_{xx} S\theta & (I_{yy} - I_{xx})C\phi S\phi C\theta & I_{xx} S^2\theta + I_{yy} S^2\phi C^2\theta + I_{zz} C^2\phi C^2\theta \end{bmatrix}$$

La función Lagrangiana se puede dividir en dinámica para la traslación y en dinámica para la rotación. Resolviendo la ecuación de Euler-Lagrange para obtener la fuerza global:

$$L_{\text{tras}} = K_{\text{tras}} - U$$

$$L_{\text{tras}} = \frac{m}{2} \dot{v}^T \dot{v} - m(g)(Z)$$

Ecuación 23 Función Lagrangiana de traslación

$$T_e = \frac{d}{dt} \left(\frac{\partial L_{\text{tras}}}{\partial \dot{v}} \right) - \frac{\partial L_{\text{tras}}}{\partial v}$$

Donde:

$$T_e = R_a^b(T) \in \mathbb{R}^3$$

Ecuación 24 Fuerza Global

$$\frac{\partial L_{tras}}{\partial v} = -m(g) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\frac{\partial L_{tras}}{\partial \dot{v}} = m(\dot{v})$$

$$\frac{d}{dt} \left(\frac{\partial L_{tras}}{\partial \dot{v}} \right) = m\ddot{v}$$

Sustituyendo:

$$T_e = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} + m(g) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} C\psi C\theta & C\psi S\theta S\phi - S\psi C\phi & C\psi S\theta C\phi + S\psi S\phi \\ S\psi C\theta & S\psi S\theta S\phi + C\psi C\phi & S\psi S\theta C\phi - C\psi S\phi \\ -S\theta & C\theta S\phi & C\theta C\phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ u_1 \end{bmatrix} = \begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} + m(g) \end{bmatrix}$$

Finalmente, las ecuaciones dinámicas de traslación son:

$$u_1 = T_1 + T_2 + T_3 + T_4$$

$$\ddot{x} = (C\psi S\theta C\phi + S\psi S\phi) \frac{u_1}{m}$$

$$\ddot{y} = (S\psi S\theta C\phi - C\psi S\phi) \frac{u_1}{m}$$

$$\ddot{z} = (C\theta C\phi) \frac{u_1}{m} - g$$

Ecuación 25 Ecuaciones dinámicas de traslación

Resolviendo la ecuación de Euler-Lagrange para obtener el par global:

$$L_{rot} = K_{rot}$$

$$L_{rot} = \frac{J(\eta)}{2} \dot{\eta}^T \dot{\eta}$$

Ecuación 26 Función Lagrangiana de rotación

$$T_{\eta} = \frac{d}{dt} \left(\frac{\partial L_{\text{rot}}}{\partial \dot{\eta}} \right) - \frac{\partial L_{\text{rot}}}{\partial \eta}$$

Donde:

$$\frac{\partial L_{\text{rot}}}{\partial \eta} = \frac{1}{2} \frac{\partial J(\eta) \dot{\eta}^T \dot{\eta}}{\partial \eta}$$

$$\frac{\partial L_{\text{rot}}}{\partial \dot{\eta}} = J(\eta) \dot{\eta}$$

$$\frac{d}{dt} \left(\frac{\partial L_{\text{rot}}}{\partial \dot{\eta}} \right) = J(\eta) \ddot{\eta} + \dot{J}(\eta) \dot{\eta}$$

Definiendo la matriz de Coriolis, la cual hace referencia a los efectos giroscópicos y centrífugos del sistema; (Catillo, Lozano, y Dzul, 2005)

$$C(\eta, \dot{\eta}) = \left(\dot{J}(\eta) - \frac{1}{2} \frac{\partial J(\eta) \dot{\eta}^T}{\partial \eta} \right)$$

Ecuación 27 Matriz de Coriolis

Sustituyendo:

$$T_{\eta} = J(\eta) \ddot{\eta} + \dot{J}(\eta) \dot{\eta} - \frac{1}{2} \frac{\partial J(\eta) \dot{\eta}^T}{\partial \eta}$$

$$T_{\eta} = J(\eta) \ddot{\eta} + C(\eta, \dot{\eta}) \dot{\eta}$$

$$\ddot{\eta} = J(\eta)^{-1} (T_{\eta} - C(\eta, \dot{\eta}) \dot{\eta})$$

Ecuación 28 Euler-Lagrange de rotación

Obteniendo los términos no lineales de la matriz de Coriolis (RAFFO, 2007)

$$C(\eta, \dot{\eta}) = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$$

$$c_{11} = 0$$

$$c_{12} = (I_{yy} - I_{zz})(\dot{\theta} C \phi S \phi + \dot{\psi} S^2 \phi C \theta) + (I_{zz} - I_{yy}) \dot{\psi} C^2 \phi C \theta - I_{xx} \dot{\psi} C \theta$$

$$c13 = (I_{zz} - I_{yy}) \dot{\psi} C \phi S \phi C^2 \theta$$

$$c21 = (I_{zz} - I_{yy})(\dot{\theta} C \phi S \phi + \dot{\psi} S^2 \phi C \theta) + (I_{yy} - I_{zz}) \dot{\psi} C^2 \phi C \theta + I_{xx} \dot{\psi} C \theta$$

$$c22 = (I_{zz} - I_{yy}) \dot{\phi} C \phi S \phi$$

$$c23 = -I_{xx} \dot{\psi} S \theta C \theta + I_{yy} \dot{\psi} S^2 \phi C \theta S \theta + I_{zz} \dot{\psi} C^2 \phi S \theta C \theta$$

$$c31 = (I_{yy} - I_{zz})(\dot{\psi} C \phi S \phi C^2 \theta) - I_{xx} \dot{\theta} C \theta$$

$$c32 = (I_{zz} - I_{yy})(\dot{\theta} C \phi S \phi S \theta + \dot{\phi} S^2 \phi C \theta) + (I_{yy} - I_{zz}) \dot{\phi} C^2 \phi C \theta + I_{xx} \dot{\psi} C \theta S \theta - I_{yy} \dot{\psi} S^2 \phi C \theta S \theta - I_{zz} \dot{\psi} C^2 \phi C \theta S \theta$$

$$c33 = (I_{yy} - I_{zz})(\dot{\phi} C \phi S \phi C^2 \theta) - I_{yy} \dot{\psi} S^2 \phi C \theta S \theta - I_{zz} \dot{\theta} C^2 \phi C \theta S \theta + I_{xx} \dot{\theta} C \theta S \theta$$

Ecuación 29 Matriz de Coriolis

La dinámica presentada hasta este punto es compleja (no lineal), por lo tanto, se considerarán aproximaciones respecto a los ángulos de Roll-Pitch-Yaw según (Reinoso, 2014)

$$\cos \phi \approx \cos \theta \approx \cos \psi \approx 1$$

$$\text{sen } \phi \approx 0$$

$$\text{sen } \theta \approx 0$$

$$\text{sen } \psi \approx 0$$

Ecuación 30 Linalización

Al linealizar las siguientes ecuaciones serán expresadas como:

$$C(\eta, \dot{\eta})\dot{\eta} = \begin{bmatrix} 0 & (I_{zz} - I_{yy})\dot{\psi} - I_{xx}\dot{\psi} & 0 \\ (I_{yy} - I_{zz})\dot{\psi} + I_{xx}\dot{\psi} & 0 & 0 \\ -I_{xx}\dot{\theta} & (I_{yy} - I_{zz})\dot{\phi} & 0 \end{bmatrix}$$

Ecuación 31 Matriz de Coriolis lineal

$$J(\eta) = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$$

Ecuación 32 Matriz Jacobiana lineal

Reescribiendo la ecuación 26 obtenemos:

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}^{-1} \left(\begin{bmatrix} T_{\phi} \\ T_{\theta} \\ T_{\psi} \end{bmatrix} - \begin{bmatrix} 0 & (I_{zz} - I_{yy})\dot{\psi} & 0 \\ (I_{yy} - I_{zz})\dot{\psi} & 0 & 0 \\ -I_{xx}\dot{\theta} & (I_{yy})\dot{\phi} & 0 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \right)$$

Despejando y agrupando, se obtienen las siguientes ecuaciones

$$\ddot{\phi} = \frac{T_{\phi}}{I_{xx}} + \frac{(I_{yy} - I_{zz})\dot{\psi}\dot{\theta}}{I_{xx}}$$

$$\ddot{\theta} = \frac{T_{\theta}}{I_{yy}} + \frac{(-I_{xx} + I_{zz})\dot{\psi}\dot{\phi}}{I_{yy}}$$

$$\ddot{\psi} = \frac{T_{\psi}}{I_{zz}} + \frac{(I_{xx} - I_{yy})\dot{\phi}\dot{\theta}}{I_{zz}}$$

Ecuación 33 Ecuaciones dinámicas de rotación

6.2.2 Ecuaciones de Newton-Euler

La obtención del modelo dinámico de un mecanismo, y en particular de un robot, se basa en el equilibrio de fuerzas establecidos en la segunda ley de Newton, la denominada ley de Euler; (Barrientos, 1996).

$$\sum F = m\dot{v} \qquad \sum T = I\dot{\omega} + \omega \times (I\omega)$$

Ecuación 34 Newton-Euler

Obteniendo las ecuaciones

$$T_e = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ m(g) \end{bmatrix}$$

Ecuación 35 Ecuación de newton para las fuerzas

$$\begin{bmatrix} C\psi C\theta & C\psi S\theta S\phi - S\psi C\phi & C\psi S\theta C\phi + S\psi S\phi \\ S\psi C\theta & S\psi S\theta S\phi + C\psi C\phi & S\psi S\theta C\phi - C\psi S\phi \\ -S\theta & C\theta S\phi & C\theta C\phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ u_1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ m(g) \end{bmatrix} = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}$$

Finalmente, las ecuaciones dinámicas en relación a las fuerzas al cuadricoptero son:

$$u_1 = T_1 + T_2 + T_3 + T_4$$

$$\ddot{x} = (C\psi S\theta C\phi + S\psi S\phi) \frac{u_1}{m}$$

$$\ddot{y} = (S\psi S\theta C\phi - C\psi S\phi) \frac{u_1}{m}$$

$$\ddot{z} = (C\theta C\phi) \frac{u_1}{m} - g$$

Ecuación 36 Ecuaciones dinámicas de traslación

Como se observa, las ecuaciones dinámicas de traslación son las misma que con el método Euler-Lagrange. Para el caso de momentos:

$$T_\eta = I\dot{\omega} + \omega \times (I\omega)$$

Ecuación 37 Ecuación de newton para los torques

$$I\dot{\omega} = \begin{bmatrix} T_\phi \\ T_\theta \\ T_\psi \end{bmatrix} - \omega \times (I\omega)$$

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}^{-1} * \left(\begin{bmatrix} T_\phi \\ T_\theta \\ T_\psi \end{bmatrix} + \begin{bmatrix} \dot{\theta}\dot{\psi}(I_{yy} - I_{zz}) \\ \dot{\phi}\dot{\psi}(I_{zz} - I_{xx}) \\ \dot{\phi}\dot{\theta}(I_{xx} - I_{yy}) \end{bmatrix} \right)$$

Despejando y agrupando, se obtienen las siguientes ecuaciones

$$\ddot{\phi} = \frac{T_{\phi}}{I_{xx}} + \frac{(I_{yy} - I_{zz})\psi\dot{\theta}}{I_{xx}}$$

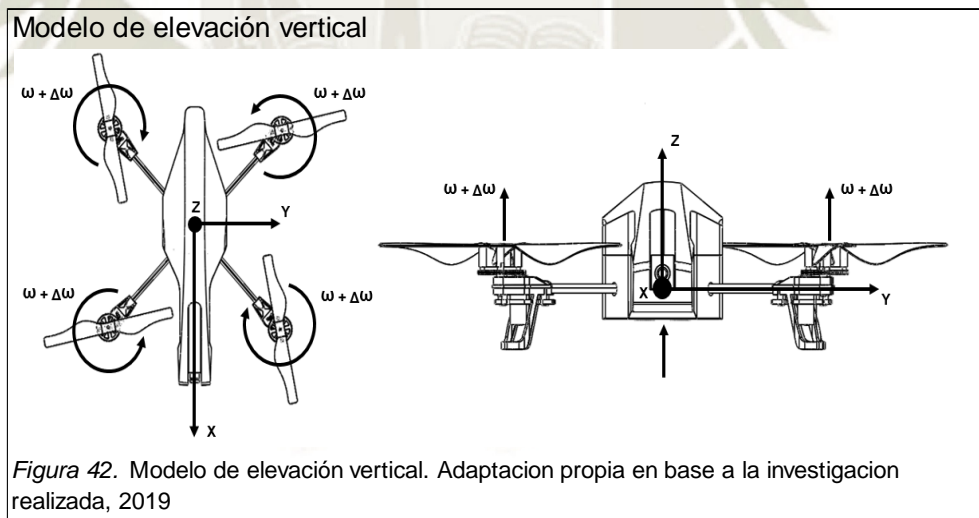
$$\ddot{\theta} = \frac{T_{\theta}}{I_{yy}} + \frac{(-I_{xx} + I_{zz})\psi\dot{\phi}}{I_{yy}}$$

$$\ddot{\psi} = \frac{T_{\psi}}{I_{zz}} + \frac{(I_{xx} - I_{yy})\phi\dot{\theta}}{I_{zz}}$$

Ecuación 38 Ecuaciones dinámicas de rotación

6.2.3 Modelamiento general del cuadricóptero

Basado en las ecuaciones podemos modelar los diferentes movimientos propios del cuadricóptero y los empujes correspondientes. Comenzaremos con el movimiento de elevación vertical.

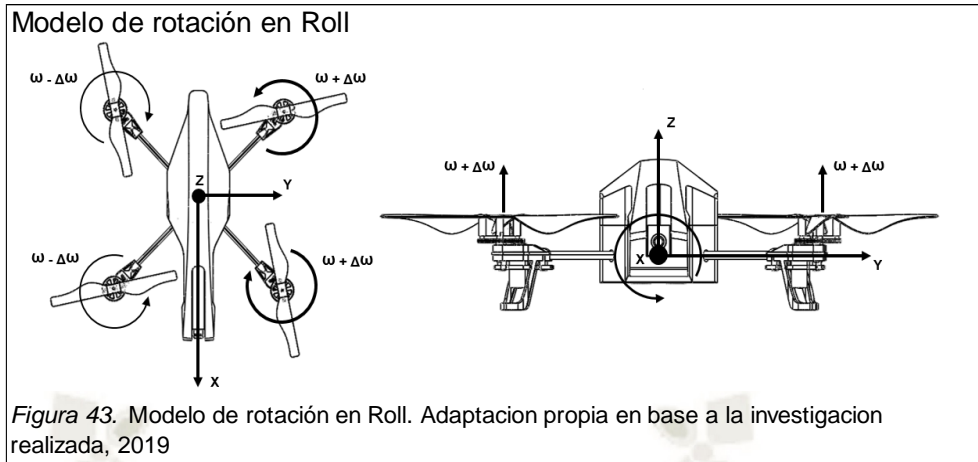


La suma de fuerzas de empuje para el movimiento de elevación está dado por la siguiente ecuación

$$u_1 = T_1 + T_2 + T_3 + T_4$$

Ecuación 39 Ecuaciones dinámica de elevación

Para la rotación del cuadricóptero en el eje Roll tomaremos en cuenta la siguiente figura.

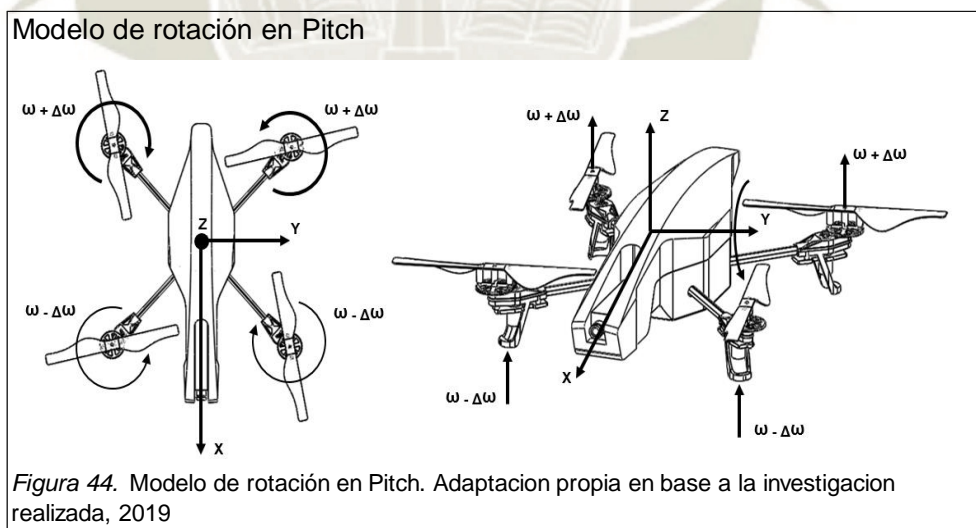


La suma de fuerzas de empuje para la rotación en Roll está dado por la siguiente ecuación

$$u_2 = T_1 - T_2 - T_3 + T_4$$

Ecuación 40 Ecuaciones dinámicas de rotación en Roll

Para la rotación del cuadricóptero en el eje Pitch tomaremos en cuenta la siguiente figura.



La suma de fuerzas de empuje para la rotación en Pitch está dado por la siguiente ecuación

$$u_3 = -T_1 - T_2 + T_3 + T_4$$

Ecuación 41 Ecuaciones dinámicas de rotación en Pitch

La suma de fuerzas de empuje para la rotación en Yaw está dado por la siguiente ecuación

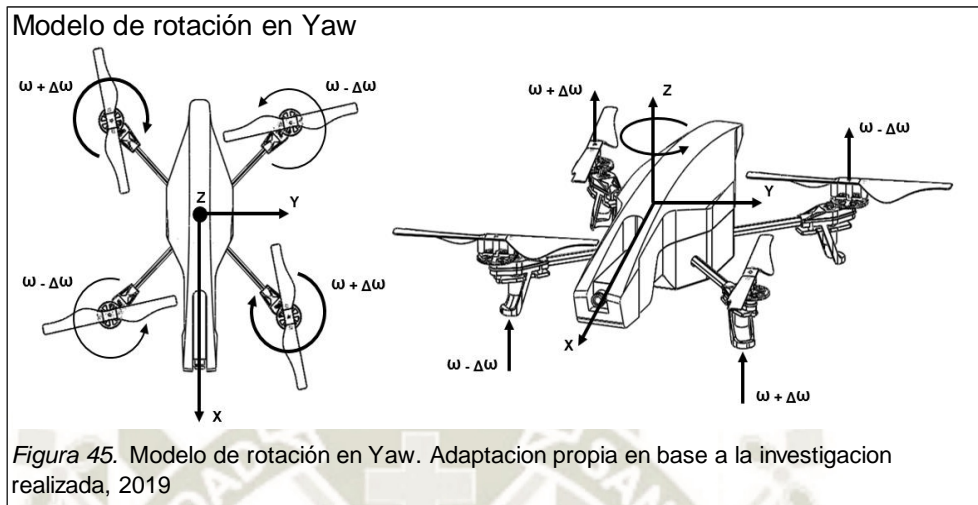


Figura 45. Modelo de rotación en Yaw. Adaptación propia en base a la investigación realizada, 2019

La suma de fuerzas de empuje para la rotación en Yaw está dado por la siguiente ecuación

$$u_4 = T_1 - T_2 + T_3 - T_4$$

Ecuación 42 Ecuaciones dinámicas de rotación en Yaw

Las velocidades de las hélices se colocan en la matriz mostrada.

$$\omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix}$$

Ecuación 43 Matriz de velocidades angulares

La suma de las velocidades angulares de los motores es.

$$\omega_r = \omega_1 - \omega_2 + \omega_3 - \omega_4$$

Ecuación 44 Sumatoria de velocidades angulares de las hélices

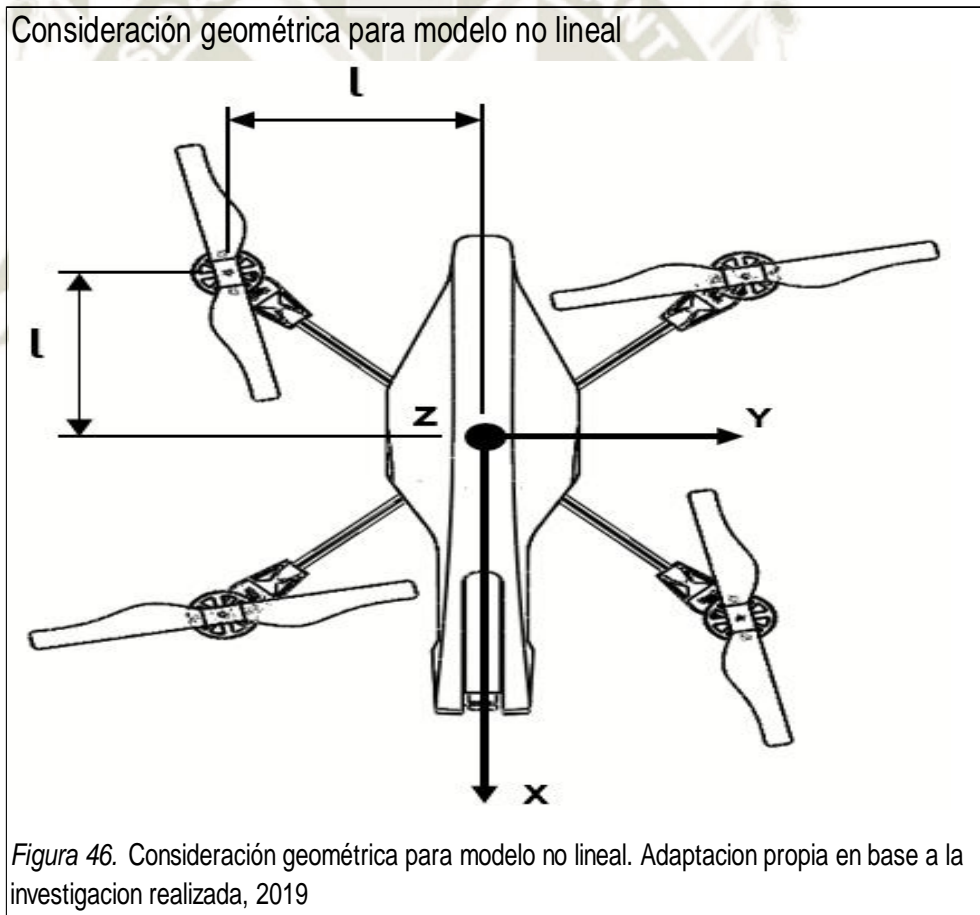
Para las ecuaciones de rotación tenemos que tomar en consideración los efectos giroscópicos producidos por la rotación de la hélice. Dos de los rotores giran en el sentido horario ($\omega_1 + \omega_3$) y los restantes en sentido antihorario ($\omega_2 + \omega_4$), esto produce un desequilibrio general cuando la sumatoria del las

velocidades de los rotores no son iguales a cero. El cuadricóptero experimenta un torque giroscópico de acuerdo a la siguiente ecuación.

$$HL = - \sum_{i=1}^4 J_r \left(\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) \omega_r$$

Ecuación 45 Efectos producidos por la rotación de las hélices

Debemos realizar una consideración geométrica para establecer el modelo no lineal en el cual se toma la distancia entre la hélice y el eje central del cuadricóptero que se observa en la figura.



Es fácil ver que los efectos giroscópicos producidos por la hélice rotación están relacionados con las ecuaciones angulares y no lineales. A su vez, se debe

de considerar las fuerzas y pares aerodinámicos que actúan sobre el cuadricoptero, y son calculados a partir de los coeficientes aerodinámicos

$$\begin{aligned}\ddot{\phi} &= \frac{I_{yy} - I_{zz}}{I_{xx}} \dot{\theta}(\psi) - \frac{J_r}{I_{xx}} \dot{\theta}(\omega_r) + l \left(\frac{u_2}{I_{xx}} \right) \\ \ddot{\theta} &= \frac{I_{zz} - I_{xx}}{I_{yy}} \dot{\phi}(\psi) + \frac{J_r}{I_{yy}} \dot{\phi}(\omega_r) + l \left(\frac{u_3}{I_{yy}} \right) \\ \ddot{\psi} &= \frac{I_{xx} - I_{yy}}{I_{zz}} \dot{\phi}(\dot{\theta}) + \frac{u_4}{I_{zz}}\end{aligned}$$

Ecuación 46 Ecuaciones de rotación no lineales

Donde

$$J_r = I_{xx} + I_{yy} + I_{zz}$$

Ecuación 47 Momento de inercia rotacional alrededor del eje

Podemos ordenar las ecuaciones de posición en un arreglo matricial como se muestra

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ \cos \phi \cos \theta \end{bmatrix} \frac{u_1}{m} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} \frac{1}{m}$$

Ecuación 48 Matriz de traslación no lineal con perturbación

Haremos lo mismo con las ecuaciones de traslación en un arreglo matricial como se muestra

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{\theta}(\psi) \left(\frac{I_{yy} - I_{zz}}{I_{xx}} \right) - \dot{\theta}(\omega_r) \left(\frac{J_r}{I_{xx}} \right) \\ \dot{\phi}(\psi) \left(\frac{I_{zz} - I_{xx}}{I_{yy}} \right) + \dot{\phi}(\omega_r) \left(\frac{J_r}{I_{xx}} \right) \\ \dot{\phi}(\dot{\theta}) \left(\frac{I_{xx} - I_{yy}}{I_{zz}} \right) \end{bmatrix} + \begin{bmatrix} l \left(\frac{u_2}{I_{xx}} \right) \\ l \left(\frac{u_3}{I_{yy}} \right) \\ \frac{u_4}{I_{zz}} \end{bmatrix} + \begin{bmatrix} A_\phi \\ A_\theta \\ A_\psi \end{bmatrix}$$

Ecuación 49 Matriz de rotación no lineal con perturbación

Convertiremos a funciones las ecuaciones para ser linealizadas

$$\dot{x} = f_1$$

$$\dot{y} = f_2$$

$$\dot{z} = f_3$$

$$\ddot{x} = f_4 = \frac{1}{m} (\cos \phi \sen \theta \cos \psi + \sen \phi \sen \psi) \sum_{i=0}^4 T_i + \frac{A_x}{m}$$

$$\ddot{y} = f_5 = \frac{1}{m} (\cos \phi \sen \theta \sen \psi - \sen \phi \cos \psi) \sum_{i=0}^4 T_i + \frac{A_y}{m}$$

$$\ddot{z} = f_6 = \frac{1}{m} (\cos \phi \cos \theta) \sum_{i=0}^4 T_i - g + \frac{A_z}{m}$$

$$\dot{\phi} = f_7$$

$$\dot{\theta} = f_8$$

$$\dot{\psi} = f_9$$

$$\ddot{\phi} = f_{10} = \dot{\theta}(\psi) \left(\frac{I_{yy} - I_{zz}}{I_{xx}} \right) - \dot{\theta}(\omega_r) \left(\frac{J_r}{I_{xx}} \right) + l \left(\frac{u_2}{I_{xx}} \right) + \frac{\dot{\phi}}{I_{xx}}$$

$$\ddot{\theta} = f_{11} = \dot{\phi}(\psi) \left(\frac{I_{zz} - I_{xx}}{I_{yy}} \right) + \dot{\phi}(\omega_r) \left(\frac{J_r}{I_{xx}} \right) + l \left(\frac{u_3}{I_{yy}} \right) + \frac{\dot{\theta}}{I_{yy}}$$

$$\ddot{\psi} = f_{12} = \dot{\phi}(\theta) \left(\frac{I_{xx} - I_{yy}}{I_{zz}} \right) + \frac{u_4}{I_{zz}} + \frac{\dot{\psi}}{I_{zz}}$$

Ecuación 50 Funciones de traslación y rotación para liberalización

Usaremos las series de Taylor por lo que debemos tener en cuenta que.

$$f_i(x, u) \approx f_i(x_0, u_0) + \left. \frac{\partial f}{\partial x} \right|_{x_0, u_0} \Delta x + \left. \frac{\partial f}{\partial u} \right|_{x_0, u_0} \Delta u$$

Ecuación 51 Serie de Taylor general

Donde:

- $f_i(x, u)$: Función original

- $f_i(x_0, u_0)$: Función evaluada en punto de equilibrio
- $\left. \frac{\partial f}{\partial u} \right|_{x_0, u_0} \Delta u$: Derivada parcial evaluada en el punto de equilibrio

Usando puntos de equilibrio (derivadas = 0).

$$\dot{x} = 0, \quad \dot{y} = 0, \quad \dot{z} = 0$$

$$\ddot{x} = 0, \quad \ddot{y} = 0, \quad \ddot{z} = 0$$

$$\dot{\phi} = 0, \quad \dot{\theta} = 0, \quad \dot{\psi} = 0$$

$$\ddot{\phi} = 0, \quad \ddot{\theta} = 0, \quad \ddot{\psi} = 0$$

Ecuación 52 Condición de derivadas igual a 0

Desarrollando las funciones para una condición de equilibrio en base a puntos de operación

$$f_6 = \frac{u_1}{m} (\cos \phi \cos \theta) - g + \frac{A_z}{m}$$

$$0 = \frac{u_1}{m} (1 * 1) - g + \frac{A_z}{m}$$

$$\frac{u_1}{m} = g - \frac{A_z}{m}$$

$$u_1 = (m)g - A_z$$

Ecuación 53 Condición de derivadas igual a 0 – Punto de operación respecto a u_1

$$f_{10} = \dot{\theta}(\dot{\psi}) \left(\frac{I_{yy} - I_{zz}}{I_{xx}} \right) - \dot{\theta}(\omega_r) \left(\frac{J_r}{I_{xx}} \right) + l \left(\frac{u_2}{I_{xx}} \right) + \frac{A_{\dot{\phi}}}{I_{xx}}$$

$$0 = 0 - 0 + l \left(\frac{u_2}{I_{xx}} \right) + \frac{A_{\dot{\phi}}}{I_{xx}}$$

$$l \left(\frac{u_2}{I_{xx}} \right) = - \frac{A_{\dot{\phi}}}{I_{xx}}$$

$$u_2(l) = -A_{\dot{\phi}}$$

Ecuación 54 Condición de derivadas igual a 0 – Punto de operación respecto a u_2

$$u_3(l) = -A_{\dot{\theta}}$$

$$u_4 = -A_{\dot{\psi}}$$

Ecuación 55 Condición de derivadas igual a 0 – Punto de operación respecto a u_3 y u_4

Asumiremos

$$A_{\phi} = A_{\theta} = A_{\psi} \cong 0$$

Ecuación 56 Condición generalizada

Ahora construiremos las matrices de Taylor en base a objetivos donde se considera de forma general

$$\dot{x} = F(x) + G(u)$$

$$y = H(x)$$

Ecuación 57 Ecuaciones de Taylor en base al objetivo

Construiremos las matrices correspondientes

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \vdots \\ \psi \end{bmatrix}$$

$$A = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \frac{\partial F_1}{\partial x_3} & \dots & \frac{\partial F_1}{\partial x_m} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \frac{\partial F_2}{\partial x_3} & \dots & \frac{\partial F_2}{\partial x_m} \\ \frac{\partial F_m}{\partial x_1} & \frac{\partial F_m}{\partial x_2} & \frac{\partial F_m}{\partial x_3} & \dots & \frac{\partial F_m}{\partial x_m} \end{bmatrix}_{x_0, u_0}$$

$$B = \begin{bmatrix} \frac{\partial F_1}{\partial u_1} & \frac{\partial F_1}{\partial u_2} & \frac{\partial F_1}{\partial u_3} & \dots & \frac{\partial F_1}{\partial u_m} \\ \frac{\partial F_2}{\partial u_1} & \frac{\partial F_2}{\partial u_2} & \frac{\partial F_2}{\partial u_3} & \dots & \frac{\partial F_2}{\partial u_m} \\ \frac{\partial F_m}{\partial u_1} & \frac{\partial F_m}{\partial u_2} & \frac{\partial F_m}{\partial u_3} & \dots & \frac{\partial F_m}{\partial u_m} \end{bmatrix}_{x_0, u_0}$$

$$C = \begin{bmatrix} \frac{\partial H_1}{\partial x_1} & \frac{\partial H_1}{\partial x_2} & \frac{\partial H_1}{\partial x_3} & \dots & \frac{\partial H_1}{\partial x_m} \\ \frac{\partial H_2}{\partial x_1} & \frac{\partial H_2}{\partial x_2} & \frac{\partial H_2}{\partial x_3} & \dots & \frac{\partial H_2}{\partial x_m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial H_m}{\partial x_1} & \frac{\partial H_m}{\partial x_2} & \frac{\partial H_m}{\partial x_3} & \dots & \frac{\partial H_m}{\partial x_m} \end{bmatrix}_{x_0, u_0}$$

Ecuación 58 Matrices de Taylor

Las matrices linealizadas y desarrolladas se muestran a continuación. La matriz es desarrollada con la asistencia de software **MATLAB**

Matriz A obtenida del Matlab

```

Matriz A:
[ 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[ 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[ 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
[ 0, 0, 0, 0, 0, 0, 0, 0, -(Az - g*m)/m, 0, 0, 0, 0, 0]
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, (Az - g*m)/m, 0, 0, 0, 0]
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

```

Figura 47. Matriz A obtenida del Matlab. Adaptacion propia en base a la investigacion realizada, 2019

Matriz B obtenida del Matlab

Matriz B:

```
[ 0,      0,      0,      0]
[ 0,      0,      0,      0]
[ 0,      0,      0,      0]
[ 0,      0,      0,      0]
[ 0,      0,      0,      0]
[ 1/m,    0,      0,      0]
[ 0,      0,      0,      0]
[ 0,      0,      0,      0]
[ 0,      0,      0,      0]
[ 0,      1/Ixx,  0,      0]
[ 0,      0,      1/Iyy,  0]
[ 0,      0,      0,      1/Izz]
```

Figura 48. Matriz B obtenida del Matlab. Adaptación propia en base a la investigación realizada, 2019

Matriz C obtenida del Matlab

Matriz C:

1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0

Figura 49. Matriz C obtenida del Matlab. Adaptación propia en base a la investigación realizada, 2019

Por lo tanto, las matrices obtenidas son


Matrices de Taylor desarrolladas

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g - A_z/m & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_z/m - g & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1/m & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & l/I_{xx} & 0 & 0 \\ 0 & 0 & l/I_{yy} & 0 \\ 0 & 0 & 0 & 1/I_{zz} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Figura 50. Matrices de Taylor desarrolladas. Adaptación propia en base a la investigación realizada, 2019



CAPITULO III
DISEÑO DEL SISTEMA DE CONTROL

Según el capítulo anterior, nos encontramos en el análisis de un sistema MIMO, **ver figura**

51.

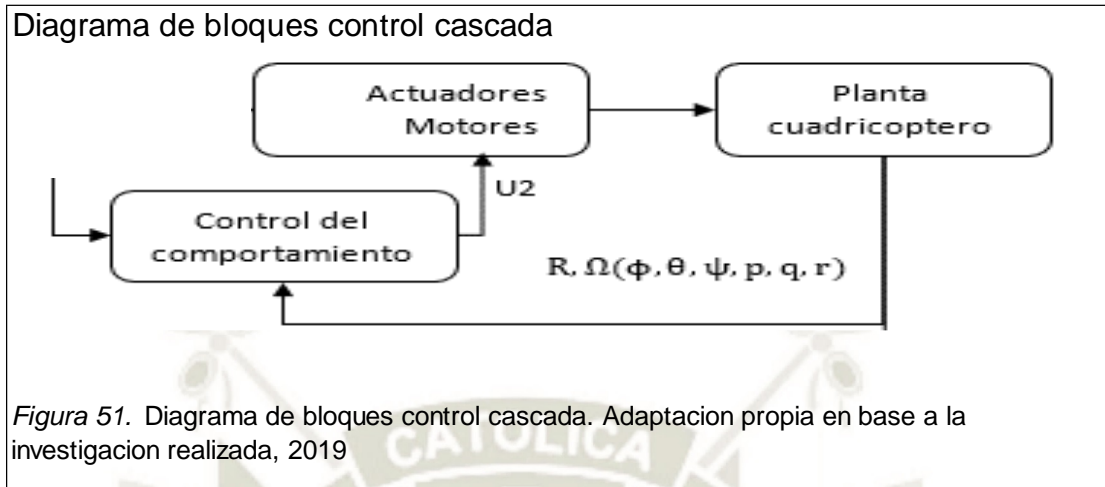
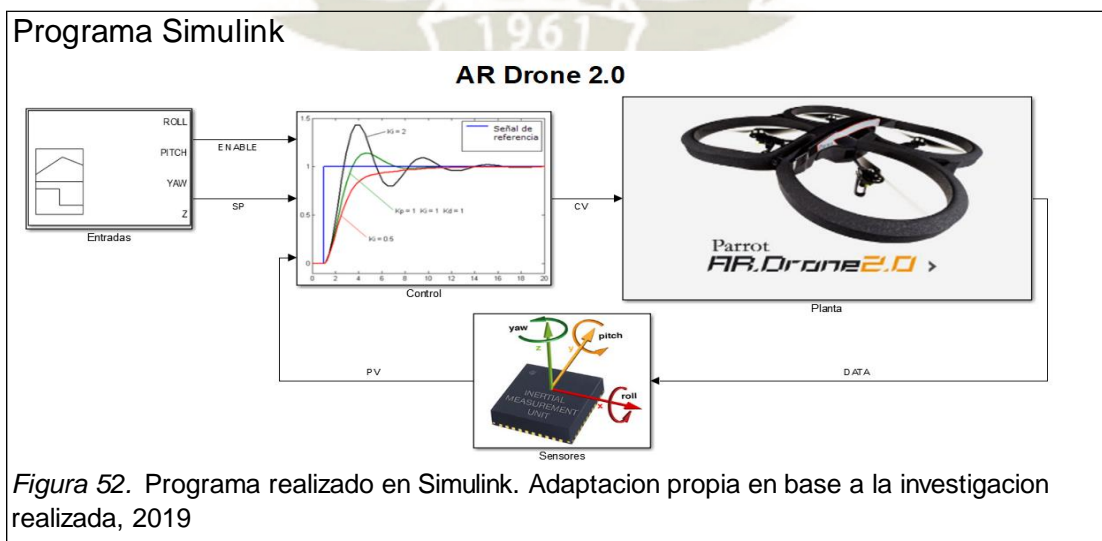


Diagrama de bloques control cascada

La gran ventaja que proporciona este tipo de control es el acceso a una variable interna medible que tenga una dinámica más rápida que la del proceso.

Como se observa, existen dos controles importantes, el control del comportamiento, el cual se analizan los ángulos; y el control de posición, el cual usa los movimientos cartesianos como datos obtenidos del control de comportamiento del sistema.

Para el desarrollo de la experiencia, usaremos el programa MATLAB y SIMULINK para la implementación del control, nos apoyaremos del hardware **Ar. Drone 2.0 Support from Embedded Coder**

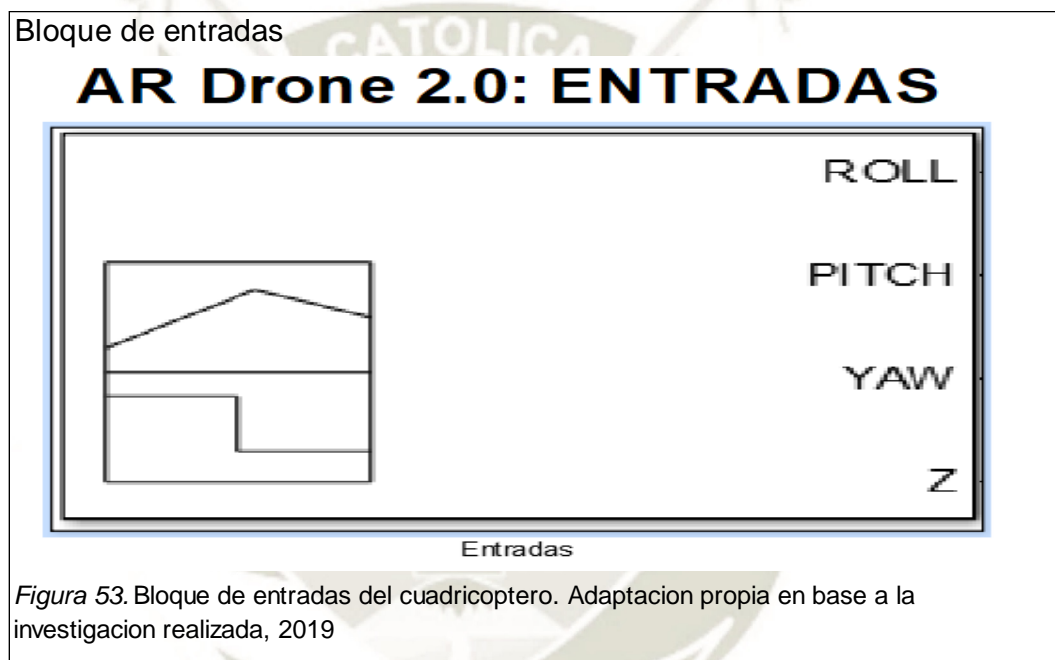


El programa realizado se divide en distintas etapas, las cuales desarrollaremos de la siguiente manera:

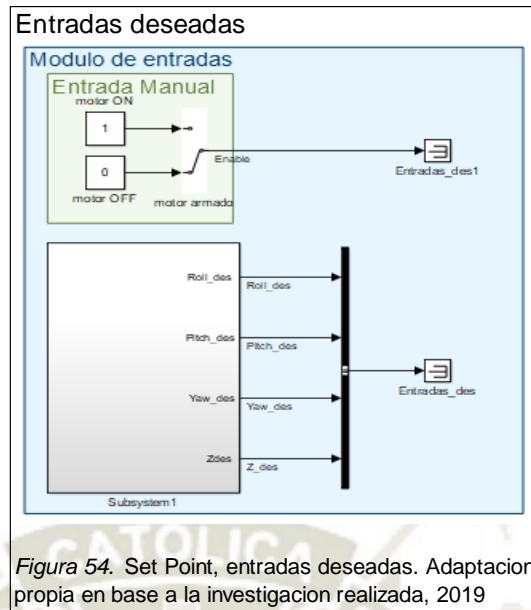
1 Bloques de implementacion

1.1 Bloques de entradas:

Según las ecuaciones dinámicas (Ecuación 33) obtenemos las 4 variables de entradas del sistema MIMO.

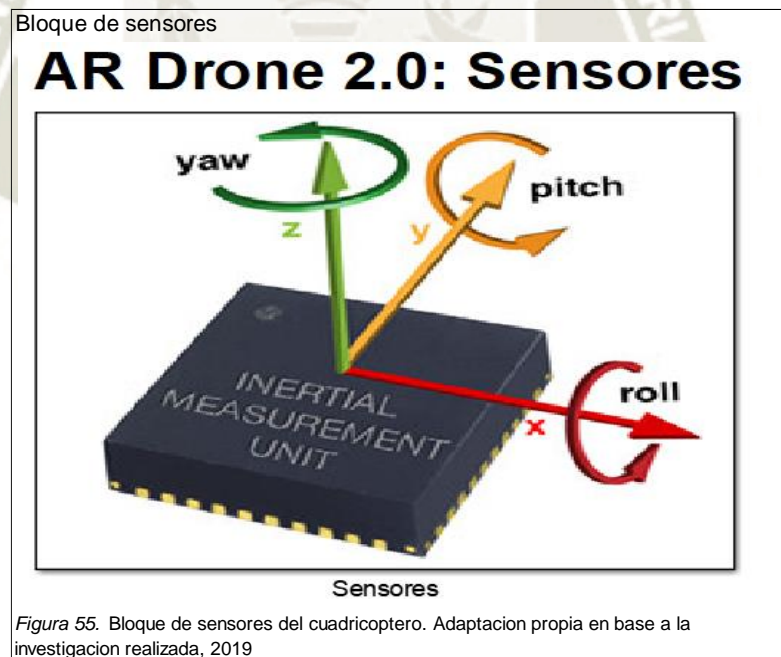


Como se observa en la figa 54, se tiene un bloque de **Entrada Manual**, en el cual se habilita el funcionamiento de los motores, una vez se de a inicio al programa y se estén cumpliendo el adecuado ingreso de entradas. También se desarrolló un bloque de entradas deseadas, en cual existen dos opciones, una para el ingreso manual de los set points, y otra donde se puede configurar set points preestablecidos



Entradas deseadas
Ver Anexo 05 para mayor detalle

1.2 Bloque de sensores:



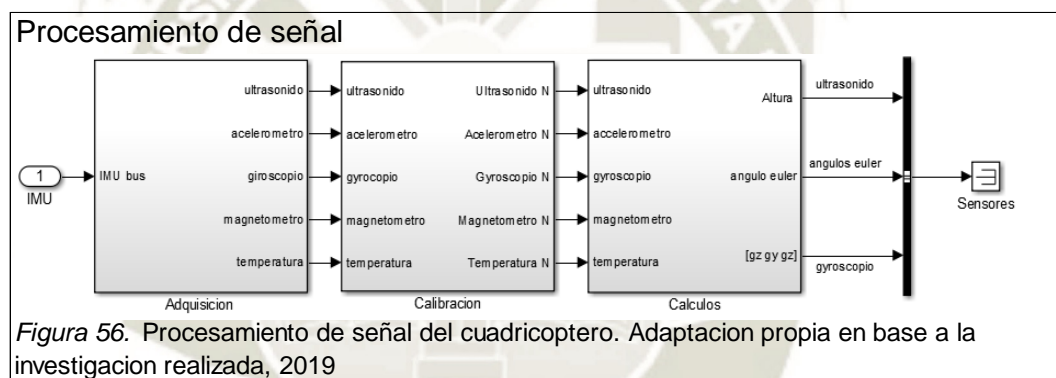
En este bloque se emplean todos los sensores que están incorporados en el Ar. Drone 2.0, los cuales son:

Tabla 6
Sensores implementados en el Ar.Drone 2.0

Sensor	Características
Giroscopio	3 ejes, precisión de 2000°/segundo
Acelerometro	3 ejes, precisión de ± 50 mg
Magnetometro	3 ejes, precisión de 6°
Ultrasonido	Medición de la altitud

Nota. Características técnicas eléctricas de los sensores del cuadricoptero. Adaptación propia en base a la investigación realizada, 2019

Estos sensores nos dan la información necesaria para la retroalimentación y así poder tener un lazo cerrado de control.



Para el uso correcto de los sensores, es necesario hacer una calibración previa para su correcto funcionamiento.

1.2.1 Calibración

Ultrasonido

Para la calibración del ultrasonido, usaremos el método de escalamiento, para ello es necesario obtener la lectura de dos puntos del sensor y reemplazarlo en la ecuación:

$$Y_{out} = (Gain * X_{out}) + Offset$$

$$Gain = \frac{y_2 - y_1}{x_2 - x_1}$$

$$Offset = y_2 - (Gain * x_2)$$

Ecuación 59 Ecuaciones de calibración del ultrasonido

Tabla 7
Calibración del sensor ultrasonico

Dato	Valor
Y1	100
Y2	24
X1	3300
X2	884
Gain	0.03141
Offset	3.6364

Nota. Datos para la calibración del sensor ultrasonico. Adaptación propia en base a la investigación realizada, 2019

Se observa que, a partir de los 24 cm, el sensor empieza a percibir cambios, es debido a esto que se toma como punto inicial los 24 cm (final de la zona muerta). De esta forma se obtendría el valor aproximado de la entrada recibida.

Acelerómetro

Para la calibración usaremos el método de calibración de dos puntos para una mejor precisión. Para el sensor acelerómetro es necesario orientar el cuadricoptero a una determinada dirección de modo que el sensor del acelerómetro sea paralelo a la gravedad (es decir, su dirección se enfrenta a la misma gravedad) o anti paralelo.

$$A_{+g}[g] = A_{OFF} + (1g \times Gain)$$

$$A_{-g}[g] = A_{OFF} - (1g \times Gain)$$

$$A_{OFF}[g] = 0.5 \times (A_{+g} + A_{-g})$$

$$Gain = 0.5 \times (A_{+g} - A_{-g})$$

$$A_{ACTUAL}[g] = \frac{A_{OUT} - A_{OFF}}{Gain}$$

Ecuación 60 Ecuaciones de calibración del acelerómetro

Eje-x

Para la calibración del acelerómetro en el eje X, se debe posicionar el cuadricoptero en dos posiciones, paralelos al eje X y en posición positiva y negativa en relación al eje.

Tabla 8
Calibración del acelerómetro en el eje X

Eje-X	Toma 1	Toma 2	Toma 3	Promedio
A+g	2532	2536	2532	2533.3333
A-g	1512	1516	1516	1514.6667

Nota. Datos para la calibración del acelerómetro en el eje X.
Adaptación propia en base a la investigación realizada, 2019

Eje-Y

Para la calibración del acelerómetro en el eje Y, se debe posicionar el cuadricoptero en dos posiciones, paralelos al eje Y y en posición positiva y negativa en relación al eje.

Tabla 9
Calibración del acelerómetro en el eje Y

Eje-Y	Toma 1	Toma 2	Toma 3	Promedio
A+g	2564	2568	2564	2565.3333
A-g	1528	1532	1528	1529.3333

Nota. Datos para la calibración del acelerómetro en el eje Y.
Adaptación propia en base a la investigación realizada, 2019

Eje-Z

Para la calibración del acelerómetro en el eje Z, se debe posicionar el cuadricoptero en dos posiciones, paralelos al eje Z y en posición positiva y negativa en relación al eje.

Tabla 10
Calibracion del acelerometro en el eje Z

Eje-Z	Toma 1	Toma 2	Toma 3	Promedio
A+g	2556	2552	2552	2553.3333
A-g	1548	1548	1548	1548

Nota. Datos para la calibracion del acelerometro en el eje Z.
Adaptacion propia en base a la investigacion realizada, 2019

Tabla 11
Calibracion del acelerometro

Eje	A+	A-	Gain	Offset
X	2533.3333	1514.6667	509.3333	2024
Y	2565.3333	1529.3333	527.3333	2040
Z	2553.3333	1548	518	2047.3333

Nota. Datos para la calibracion del acelerometro. Adaptacion propia
en base a la investigacion realizada, 2019

Obtención de Angulo

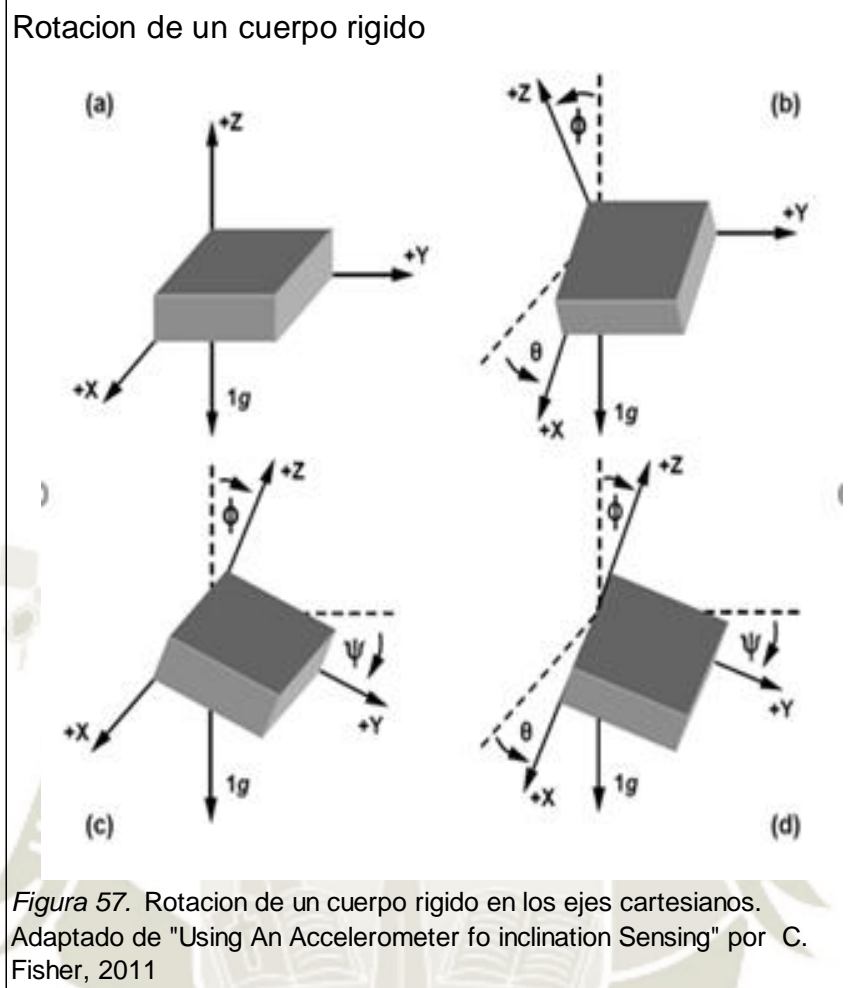


Figura 57 Rotación de un cuerpo rígido

La trigonometría básica se puede usar para mostrar que los ángulos de inclinación se pueden calcular usando la Ecuación 61

$$\phi = \tan^{-1} \left(\frac{A_x}{\sqrt{A_y^2 + A_z^2}} \right)$$

$$\theta = \tan^{-1} \left(\frac{A_y}{\sqrt{A_x^2 + A_z^2}} \right)$$

$$\psi = \tan^{-1} \left(\frac{\sqrt{A_x^2 + A_y^2}}{A_z} \right)$$

Ecuación 61 Obtención de ROLL PITCH YAW

Giroscopio

La integración de la medida del giroscopio (velocidad) nos dirá cuánto desplazamiento angular ha tenido lugar. Incluso estando en reposo el cuadricoptero, el valor sin calibración respecto al desplazamiento presenta una variación proporcional. Obviamente, este es un problema que debe abordarse.

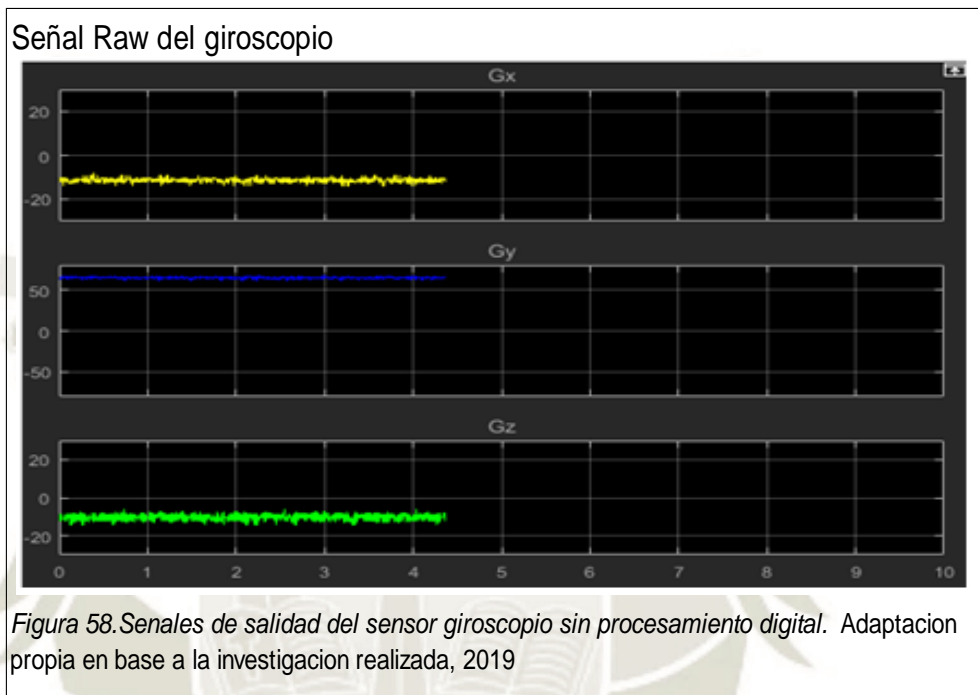


Figura 58 Señal Raw del giroscopio

El objetivo de esta calibración es eliminar la variación del giroscopio de velocidad de modo que, debería estar cerca de cero, lo que lleva a una deriva más baja. El mayor desafío aquí es que la variación del giroscopio no es constante y cambiará en función de numerosos factores. La variación depende principalmente de la temperatura (ruido térmico).

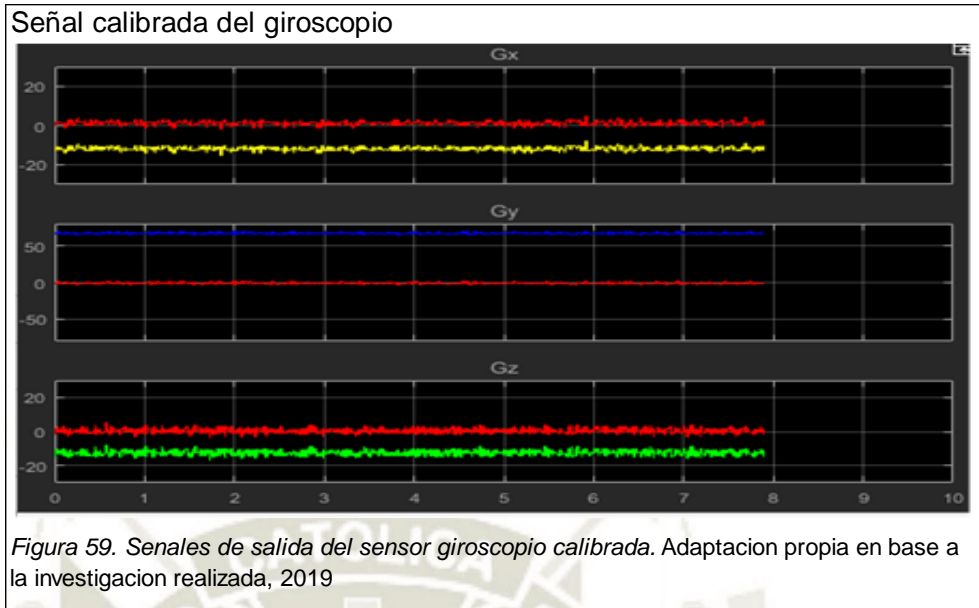
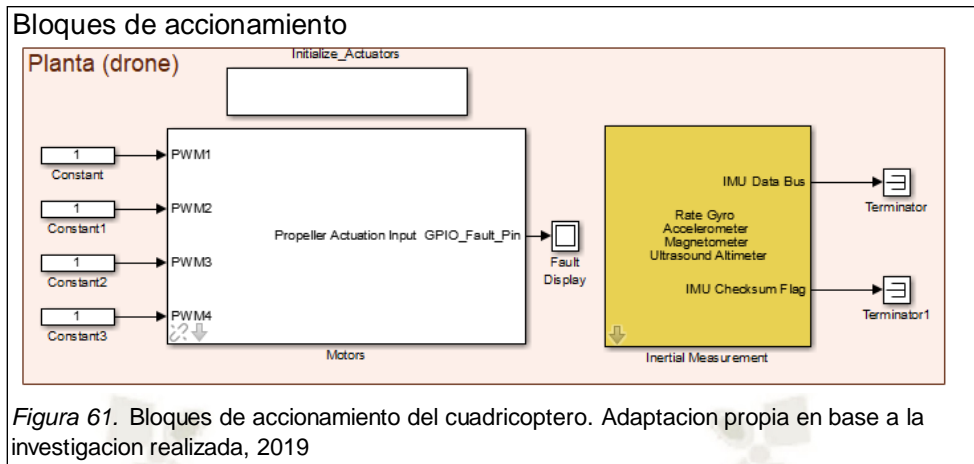


Figura 59. Señal calibrada del giroscopio.
Ver Anexo 05 para mayor detalle

1.3 Bloque de planta:

En este bloque se determina el modelo dinámico del cuadricoptero.



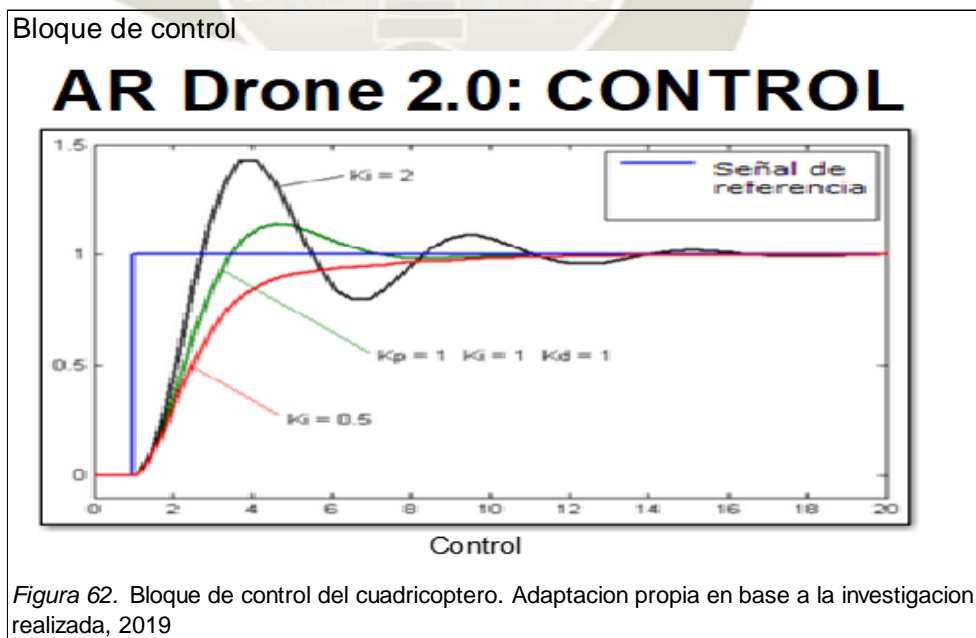


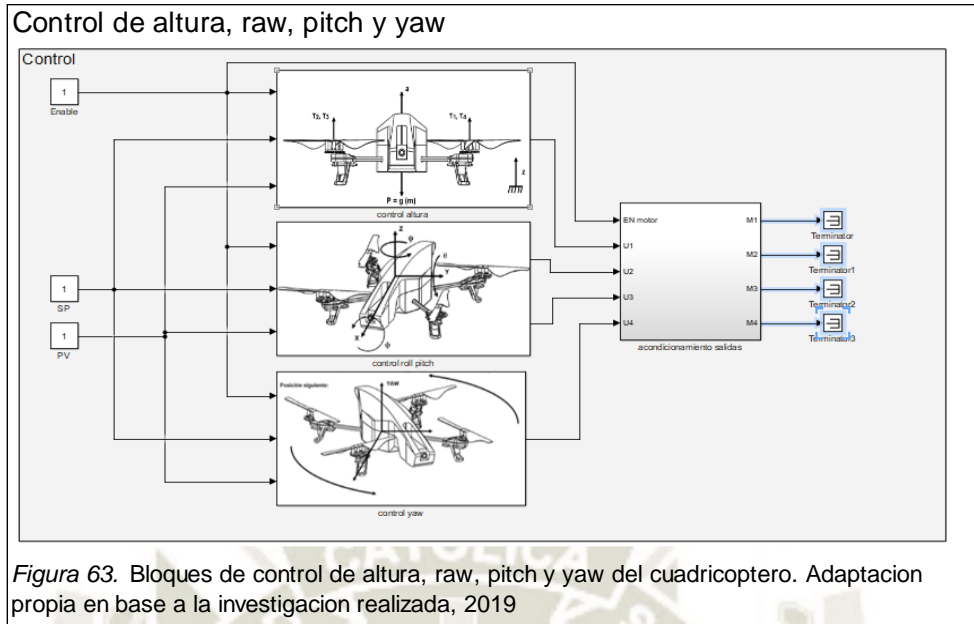
Ver Anexo 05 para mayor detalle

1.4 Bloque de control

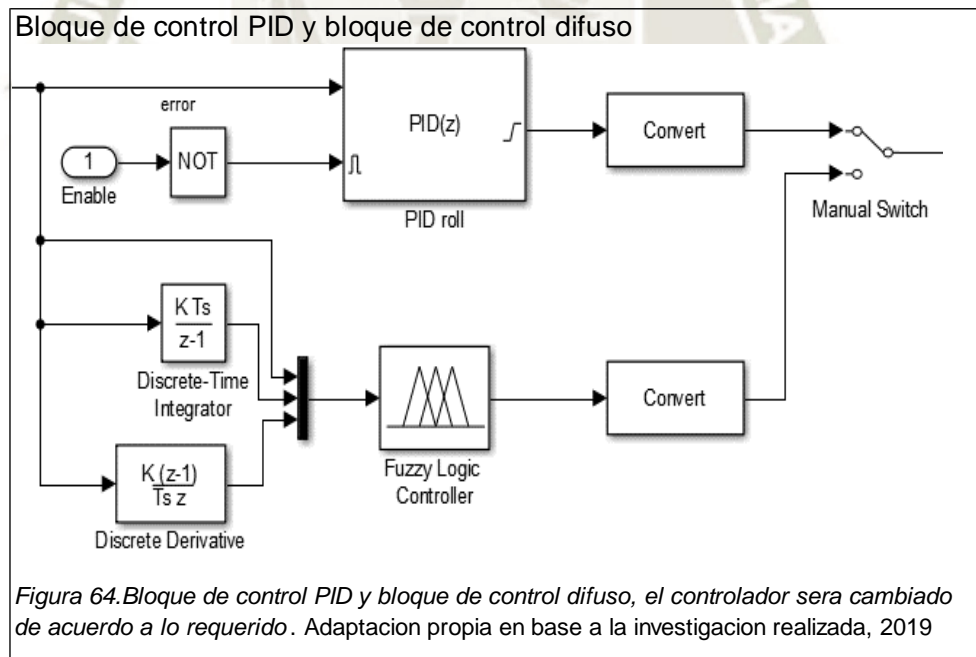
En el bloque de control, se determinó los 4 controles existentes del cuadricoptero, los cuales son:

- Control Altura
- Control Yaw
- Control Pitch
- Control Roll





Para la experimentación se compara dos tipos de control, PID y difuso; los cuales se pusieron a prueba para determinar cuál de los dos es más óptimo en la implementación.



1.4.1 Control PID

Para la implementación del controlador PID, se utilizó la sintonización de parámetros PID de forma experimental ya que es la opción que proporcionará mejores resultados.

Se siguió el procedimiento expuesto en el capítulo anterior

Se hizo el mismo proceso para cada una de las variables de entradas, teniendo en cuenta que, para la entrada de Z, se utiliza el módulo experimental de HOVER, ya que restringe el movimiento del dron a solo movimientos en el eje Z

Para las demás entradas, se utiliza el módulo experimental de STABILITY, el cual restringe todos los movimientos a expresión de los movimientos rotaciones (roll-pitch-yaw).

1.4.2 Control FUZZY

Para la experimentación del sistema de control difuso, se utiliza el sistema de inferencia de Mamdani, debido a que es más intuitivo y se utiliza principalmente en sistema donde se involucra una alta participación humana.

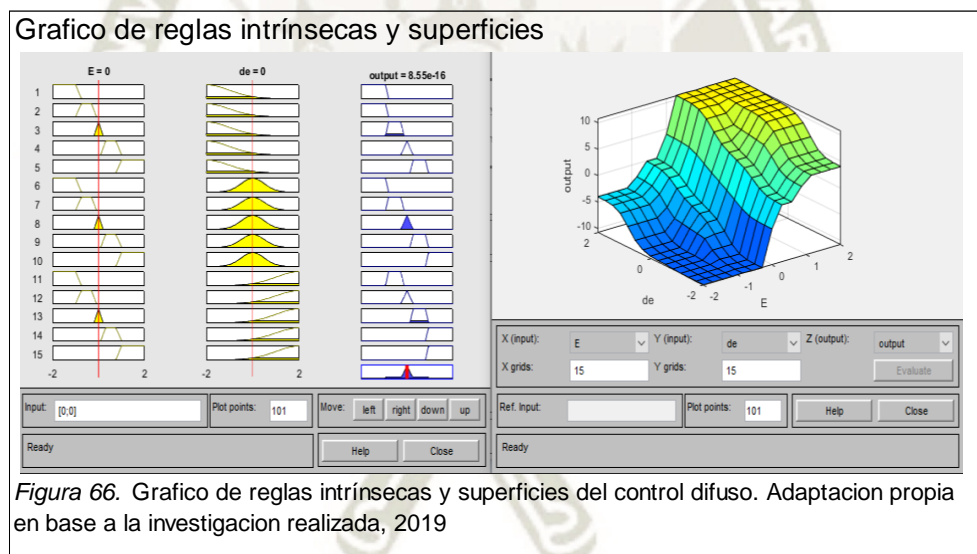
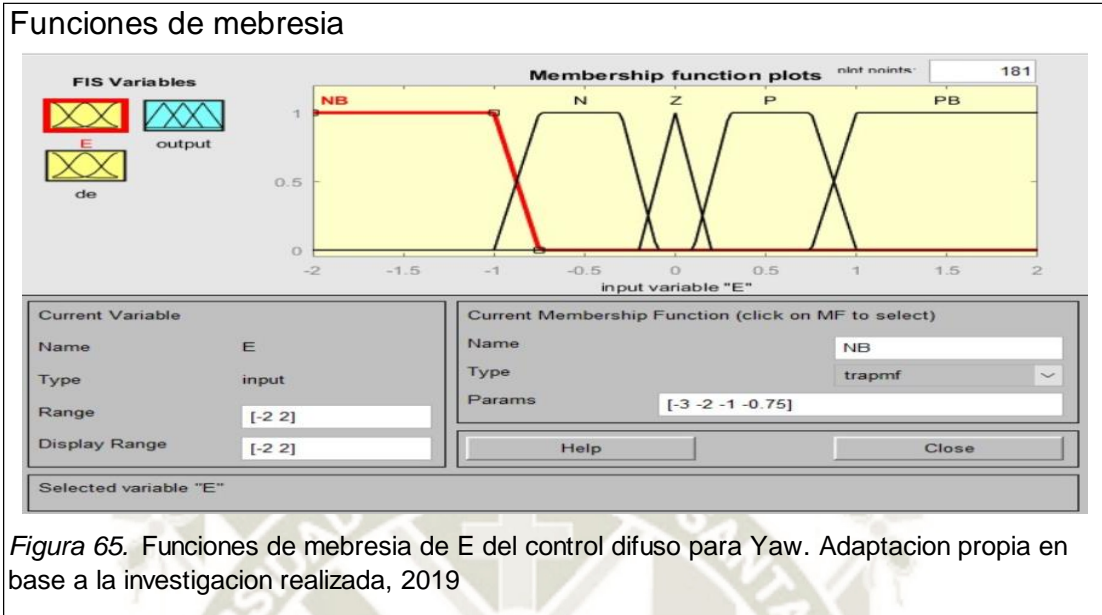
Se determinaron dos enunciados de entrada principales, los cuales son el error y la derivada del error debido a su practicidad, después se adjunta la corrección de la integral del error.

Dentro de cada entrada se consideraron 5 funciones de membresía las cuales son:

- MN: Mínimo Negativo
- N: Negativo
- Z: Zero
- P: Positivo
- MP: Máximo Positivo

Para las funciones de membresía, se están utilizando el tipo trapezoidal por que se quiere abarcar una área más grande para una reacción más pronta. A su vez se

está utilizando una función de membresía de tipo triangular para determinar la cercanía al set point, para que esta sea lo más precisa.



En total se tiene un registro de 25 reglas, las cuales serán detallas en la tabla.

Tabla 12
Reglas Fuzzy

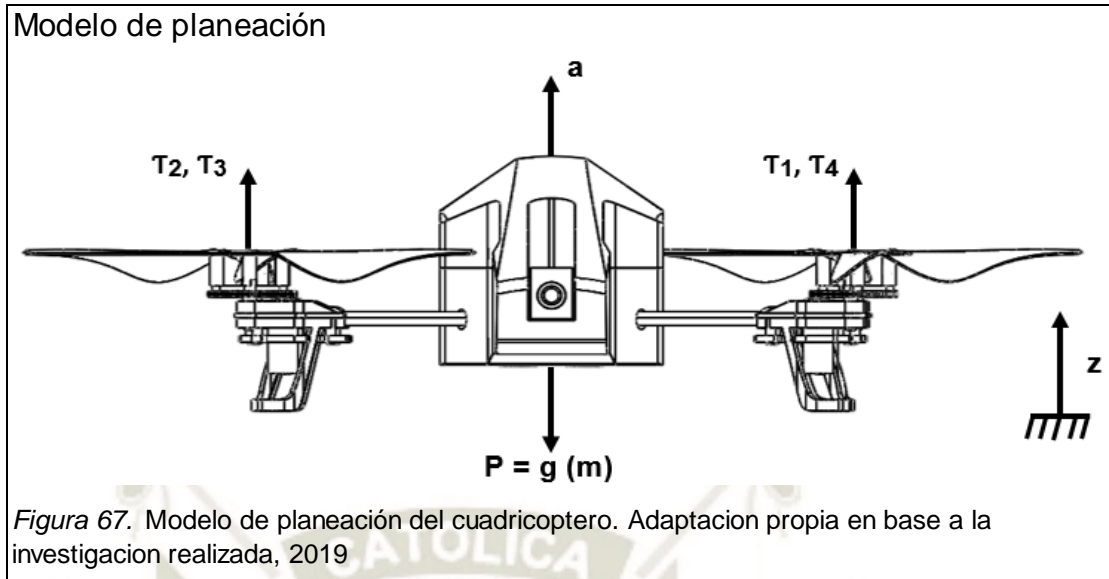
#	Regla
1	If (e is MN) and (de is MN) then (cv is F7)
2	If (e is MN) and (de is N) then (cv is F7)
3	If (e is MN) and (de is Z) then (cv is F6)
4	If (e is MN) and (de is P) then (cv is F5)
5	If (e is MN) and (de is MP) then (cv is F4)
6	If (e is N) and (de is MN) then (cv is F7)
7	If (e is N) and (de is N) then (cv is F6)
8	If (e is N) and (de is Z) then (cv is F5)
9	If (e is N) and (de is P) then (cv is F4)
10	If (e is N) and (de is MP) then (cv is F3)
11	If (e is Z) and (de is MN) then (cv is F6)
12	If (e is Z) and (de is N) then (cv is F5)
13	If (e is Z) and (de is Z) then (cv is F4)
14	If (e is Z) and (de is P) then (cv is F3)
15	If (e is Z) and (de is MP) then (cv is F2)
16	If (e is P) and (de is MN) then (cv is F5)
17	If (e is P) and (de is N) then (cv is F4)
18	If (e is P) and (de is Z) then (cv is F3)
19	If (e is P) and (de is P) then (cv is F2)
20	If (e is P) and (de is MP) then (cv is F1)
21	If (e is MN) and (de is MN) then (cv is F4)
22	If (e is MN) and (de is N) then (cv is F3)
23	If (e is MN) and (de is Z) then (cv is F2)
24	If (e is MN) and (de is P) then (cv is F1)
25	If (e is MN) and (de is MP) then (cv is F1)

Nota. Reglas para el control Fuzzy. Adaptacion propia en base a la investigacion realizada, 2019

Ver Anexo 05 para mayor detalle

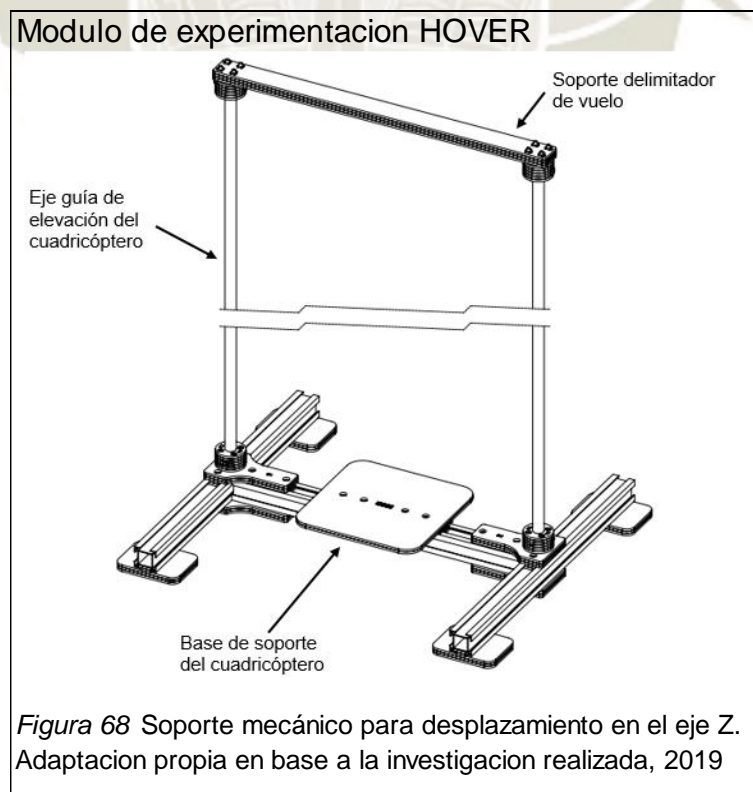
2 Modelo de planeación (hover):

La primera experiencia que se planteó a partir de un modelo dinámico, es el de planeación. Para ello, utilizamos como referencia la figura 67

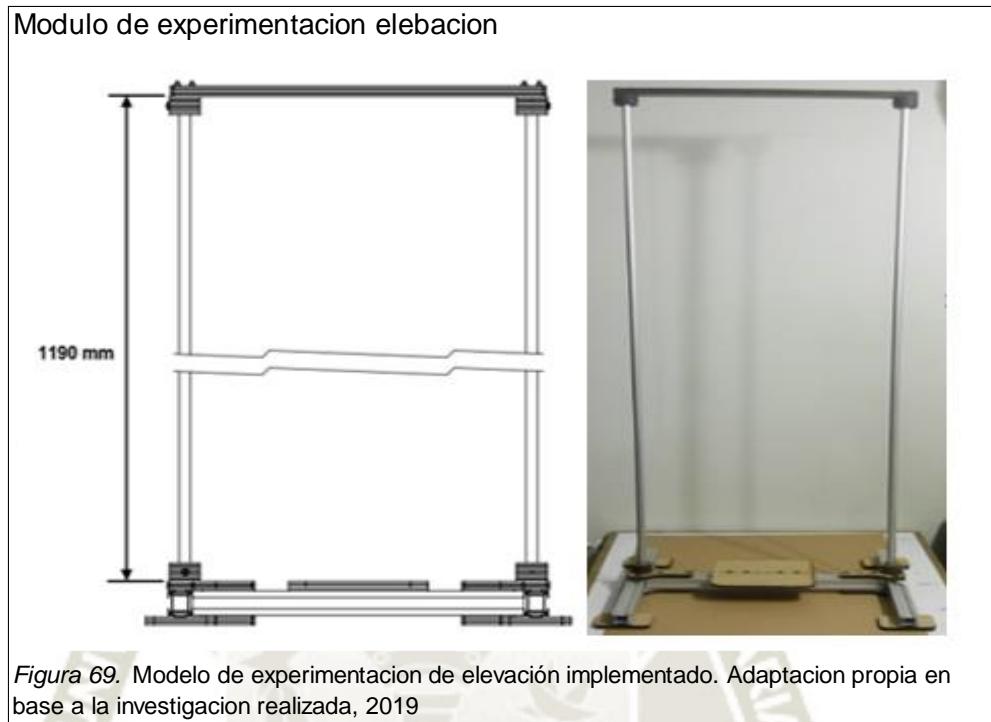


Modelo de planeación

En esta prueba solo se comprueba el control en 1D, la cual se basa en el eje Z, una de las entradas del sistema MIMO, por ende, las demás entradas se considerarán 0, así evitando los movimientos roll, pitch, yaw. Para las pruebas de planeación se realizó el diseño y la fabricación de un soporte mecánico capaz de restringir el vuelo en un solo eje, en este caso en el eje de elevación.



El módulo de experimentación de elevación permite al cuadricóptero planear de forma vertical hasta una altura relativa a la base de soporte de 1190 mm.

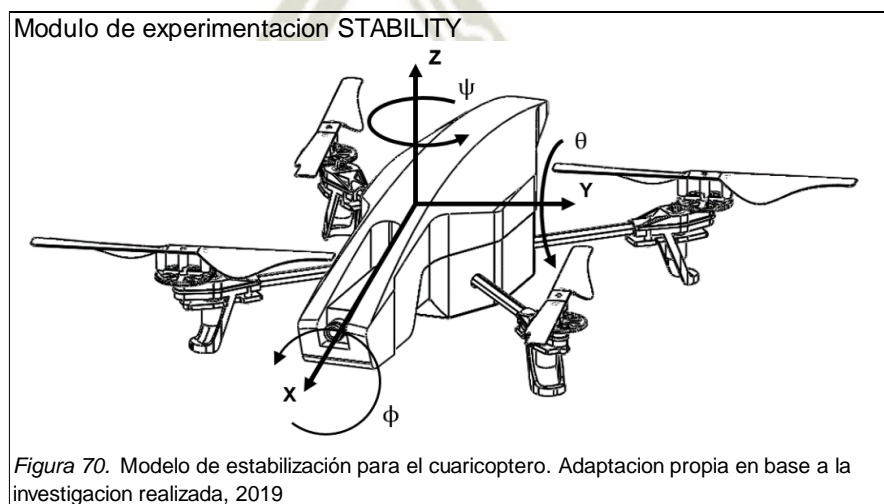


Modulo de experimentación elevación

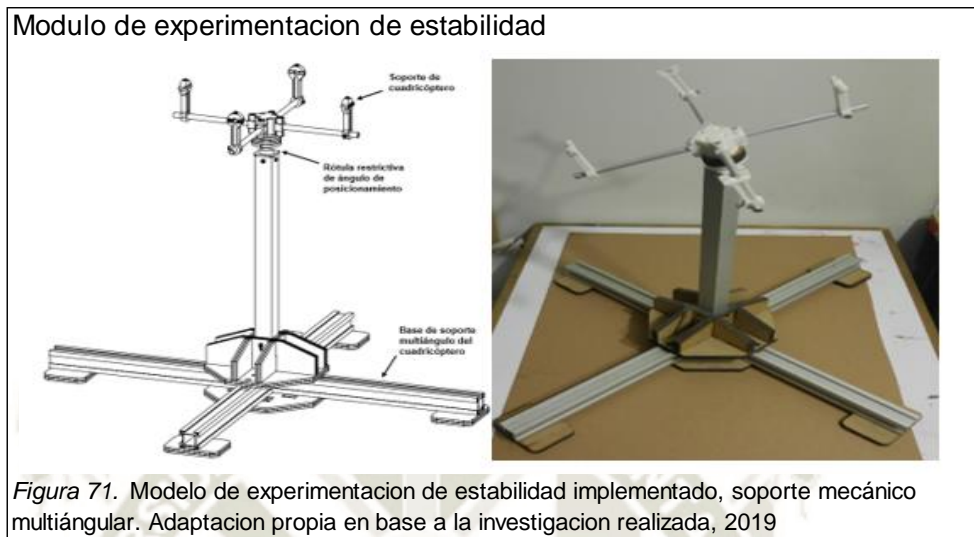
Para la implementación, se plante al cuadricóptero planear de forma vertical hasta una altura relativa a la base de soporte de 1190 mm.

3 Modelo de estabilización:

La segunda experiencia planteada a partir del modelo dinámico, es el de estabilización. Para ello, utilizamos como referencia la figura 70.



Para las pruebas de estabilización se realizó el diseño y la fabricación de un soporte mecánico capaz de permitir el vuelo en los tres ejes de rotación, y restringir la traslación en los mismos ejes.



4 Interfaz grafica

Para el desarrollo de la investigación, se vio por conveniente utilizar una interfaz gráfica para dar una mayor facilidad e interacción al momento de realizar las pruebas de verificación del programa, por ello, la interfaz gráfica fue creada mediante GUIDE de Matlab. Gracias a esta interfaz se realiza el control de todo el proceso de implementación. Esta interfaz abre el programa de Simulink correspondiente y mediante comandos se hace uso de los bloques necesarios.

Se desarrollaron dos opciones en la pantalla principal del GUI del proyecto



Figura 72. GUI principal desarrollado para la implementación del proyecto. Adaptación propia en base a la investigación realizada, 2019

Figura 72 GUI del proyecto

4.1 Módulo automático

Se desarrolló el GUI automático, en el cual el usuario puede seleccionar los módulos experimentales ya preestablecidos y seleccionar el método de control con el cual se registrará el experimento. Tener en cuenta que se debe de utilizar el módulo de experimentación adecuado para cada experimento.



Figura 73. GUI desarrollado para las pruebas pre establecidas del proyecto. Adaptación propia en base a la investigación realizada, 2019

4.2 Modulo manual

Se desarrolló el GUI manual, en el cual el usuario puede colocar los sets points deseados para la experimentación, en este módulo se utiliza el método de control más óptimo obtenido en las experimentaciones.



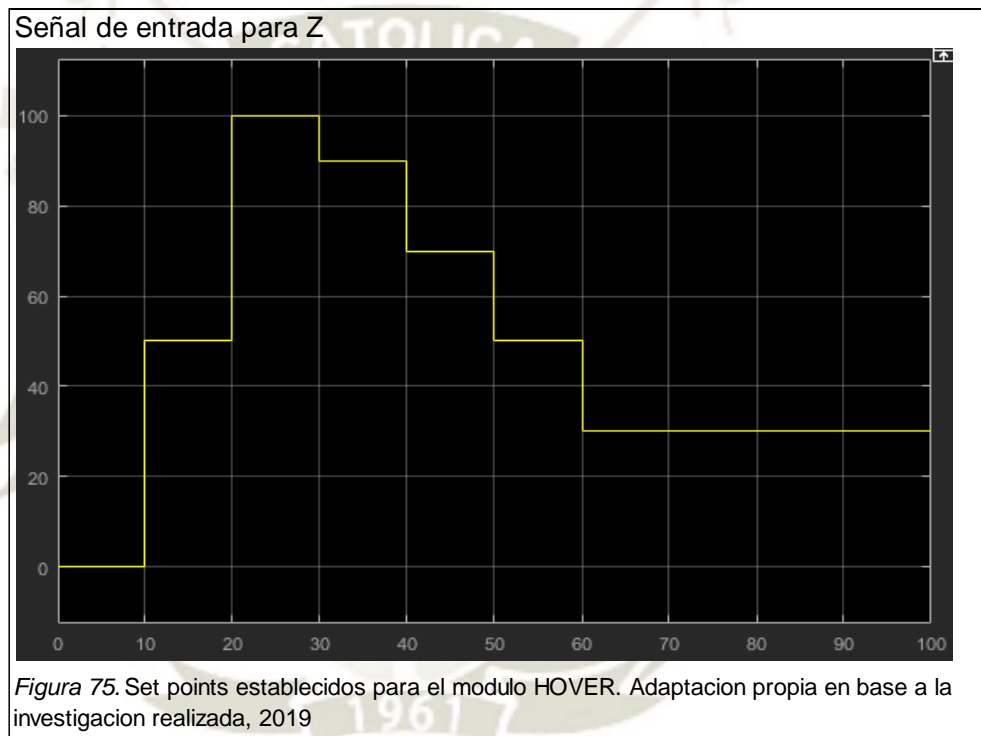




En este capítulo se explicará todas las pruebas planteadas para conseguir el mejor control posible del cuadricoptero, para ello usamos la interfaz gráfica implementada, la cual nos ayuda a entender y usar de mejor manera la experimentación.

1 Planeación (HOVER)

Para las pruebas en el módulo de planeación realizaremos un vuelo restringido entre dos alturas distintas. Una altura baja (0.3 m) y una altura máxima (1 m). Ver figura 75



Para el desarrollo de la experimentación se tomó en cuenta los efectos de inercia, el aire, la temperatura y la intensidad luminosa

Planeación entre una altura media y altura máxima

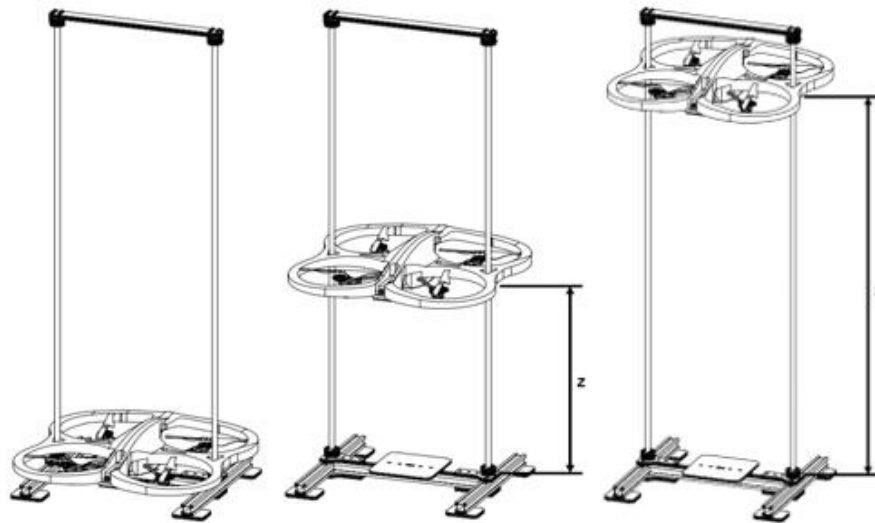


Figura 76. Planeación entre una altura media y altura máxima . Adaptacion propia en base a la investigacion realizada, 2019

Figura 76 Planeación a una altura media y altura máxima

Planeación entre una altura media y altura máxima



Figura 75. Modulo experimental utilizado para la experimentacion. Adaptacion propia en base a la investigacion realizada, 2019

1.1 Control PID

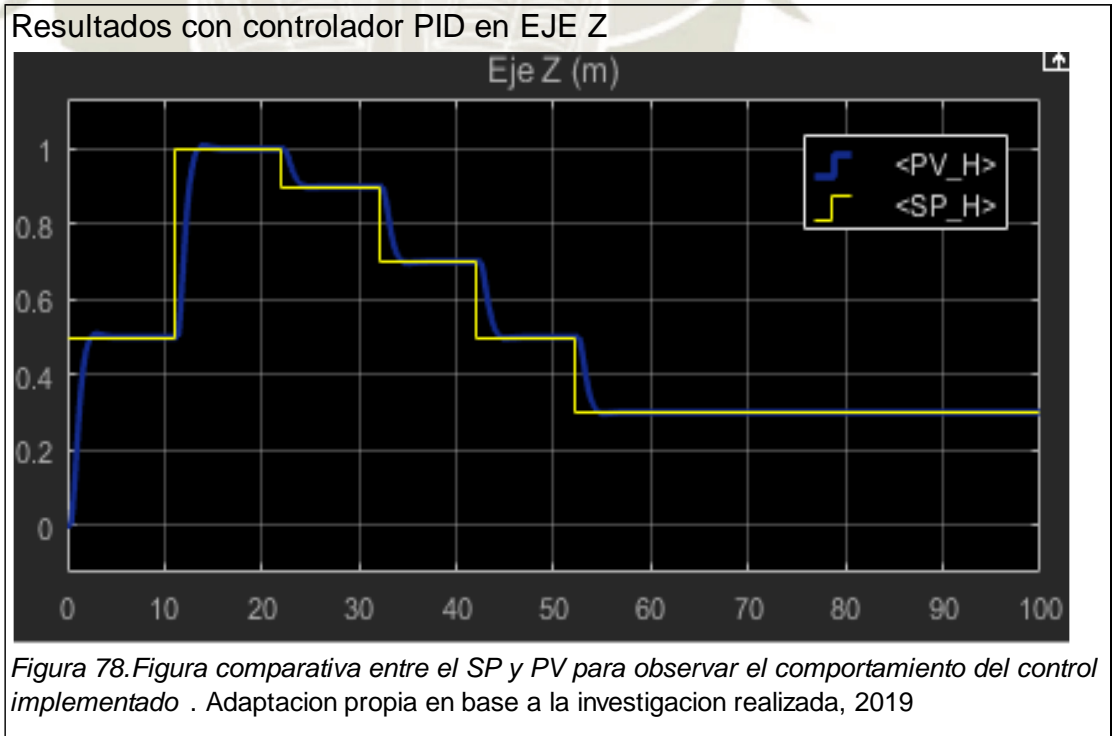
Para esta experimentación se decidió hacer la sintonización del control PID mediante la sintonización experimental, del cual se obtuvo los siguientes parámetros. Ver tabla 13

Tabla 13
Valores PID en el eje Z

Dato	Valor 1
Kp	0.08
Kd	0.0012
Ki	0.0005

Nota. Datos del controlador PID del eje Z por metodo experimental. Adaptacion propia en base a la investigacion realizada, 2019

Como resultado de los movimientos básicos en el eje Z como set points, se obtuvieron los siguientes gráficos comparativos

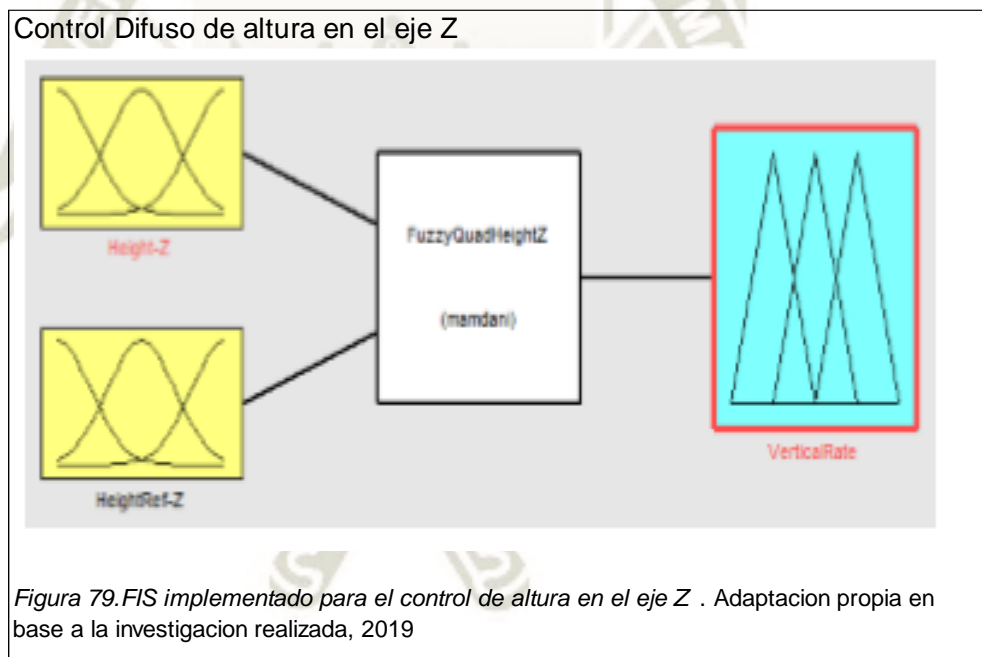


En la experimentación de planeación, se observa:

- El sistema logra llegar a los puntos establecidos y que el error en estado estacionario 0 mínimo
- La respuesta tiene un sobre impulsó del 5%
- El tiempo de establecimiento es de 2s
- El tiempo de subida es de 1s

1.2 Control DIFUSO

Para la experimentación se tiene que en la entrada del controlador se encuentran los valores medidos y los valores de referencia, en total en este controlador se tiene 5 funciones de membresía para cada entrada y 5 funciones para la salida. Rango de entrada [0 3] y rango de salida [-1 1]. Cada función de membresía es de tipo triangular y posee 25 Reglas



Como resultado de los movimientos básicos en el eje Z como set points, se obtuvieron los siguientes gráficos comparativos

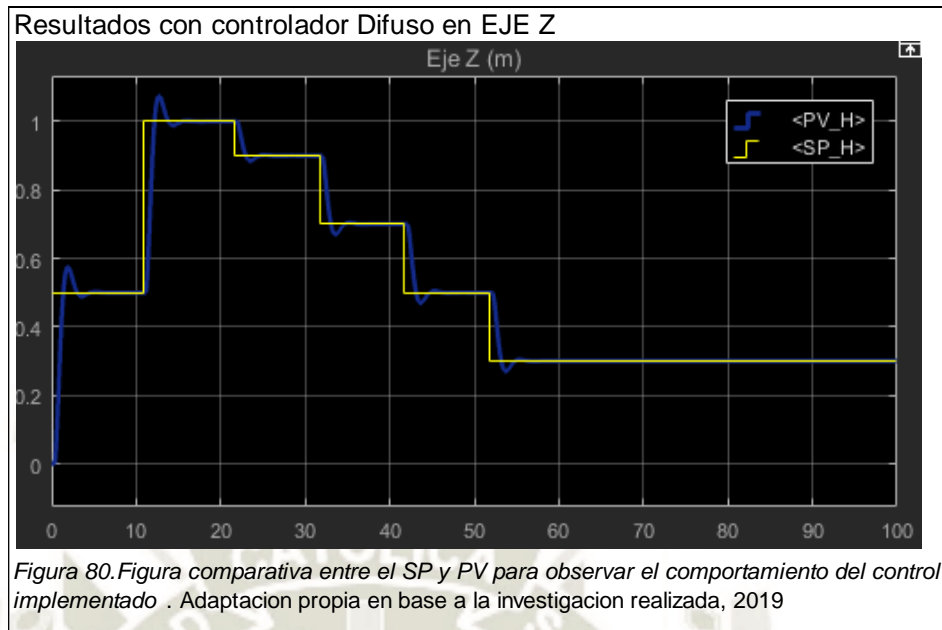


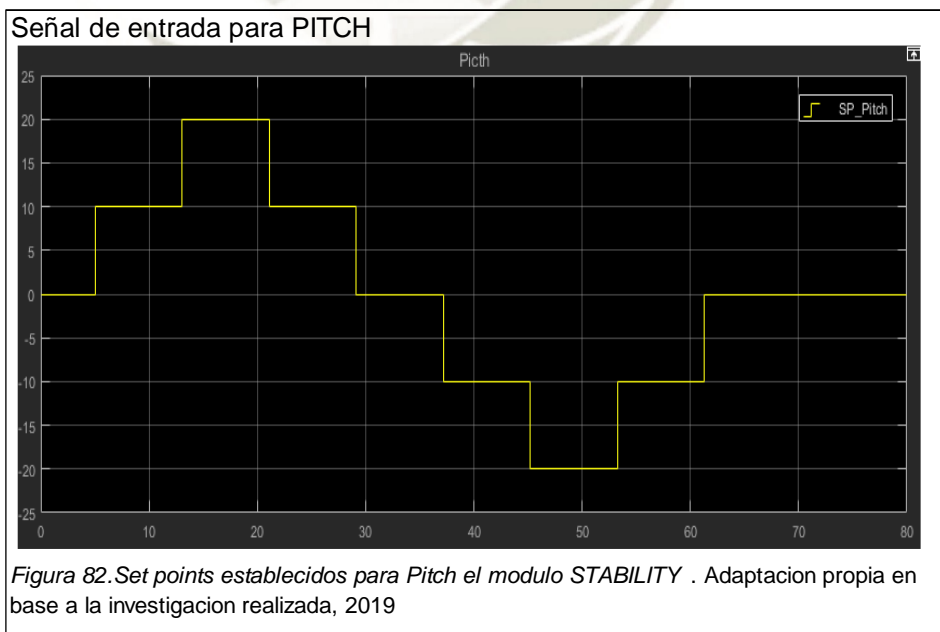
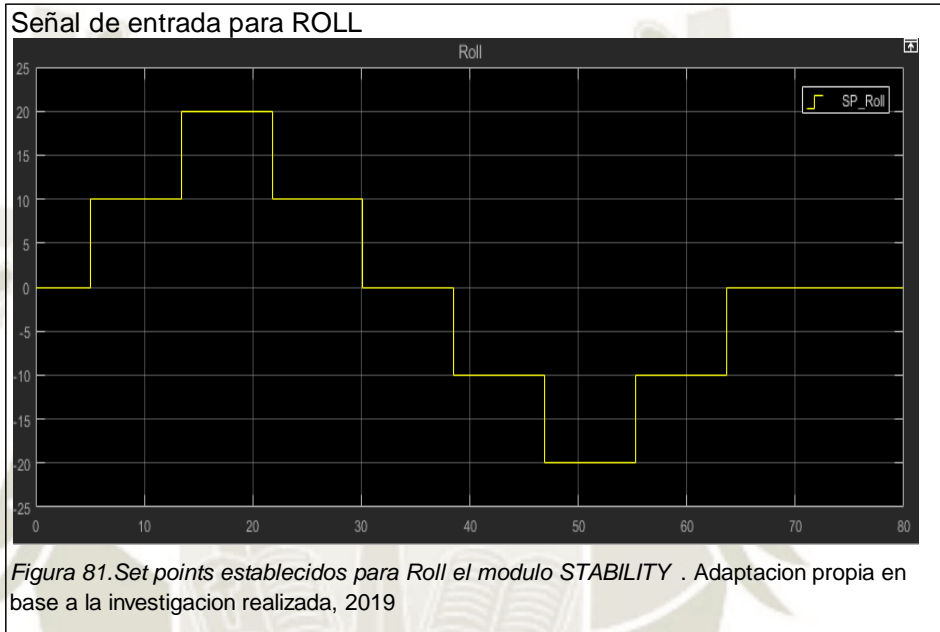
Figura 80. Resultados con controlador Difuso en EJE Z
En la experimentación de planeación, se observa:

- El sistema logra llegar a los puntos establecidos y que el error en estado estacionario 0 mínimo
- La respuesta tiene un sobre impulsó del 10%
- El tiempo de establecimiento es de 4s
- El tiempo de subida es de 2s

Según los datos obtenidos, se obtiene que el control PID realizado tiene mejor respuesta que el controlador difuso, se observa que posee menos sobre pulsos y que el tiempo de reacción es menor, pero ambos controles cumplen con la función principal de llegar al punto establecido y reducir el error de la variable medida con la variable deseada

2 Estabilidad (STABILITY)

Para las pruebas realizamos un vuelo restringido de estabilización. Debemos alcanzar un ángulo de posición estable para los 3 ángulos de posicionamiento con la mínima distorsión de los mismos. Ver figuras 81-83



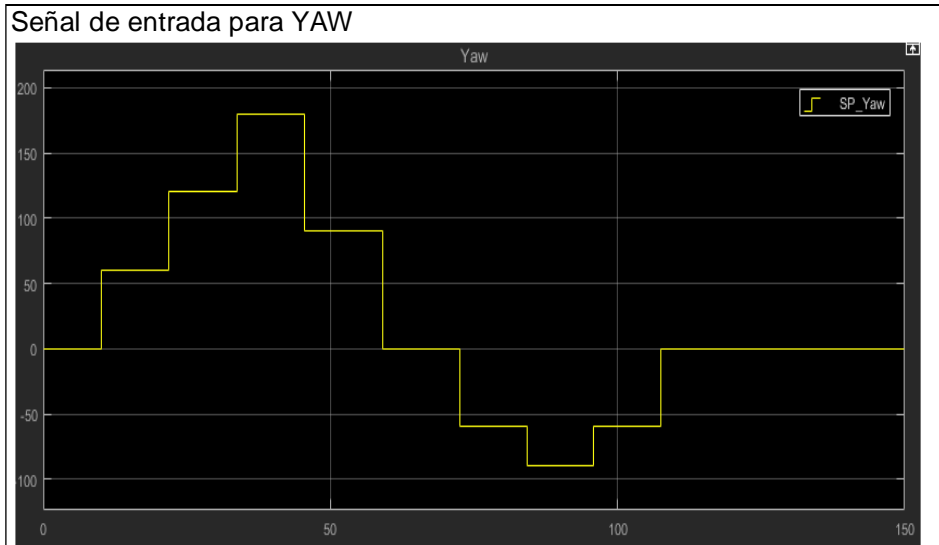


Figura 83. Set points establecidos para Yaw el modulo STABILITY . Adaptacion propia en base a la investigacion realizada, 2019

Señal de entrada para YAW

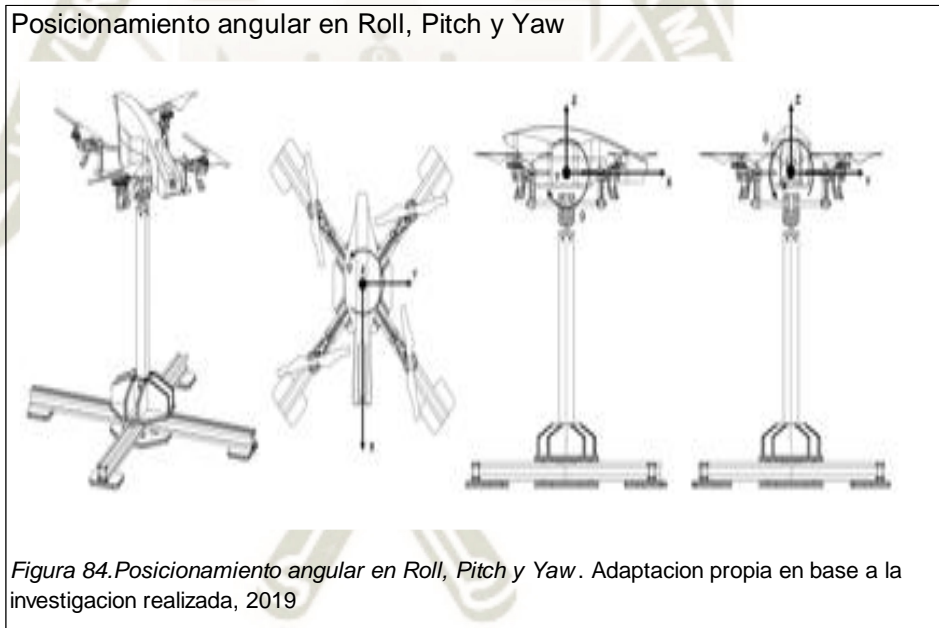
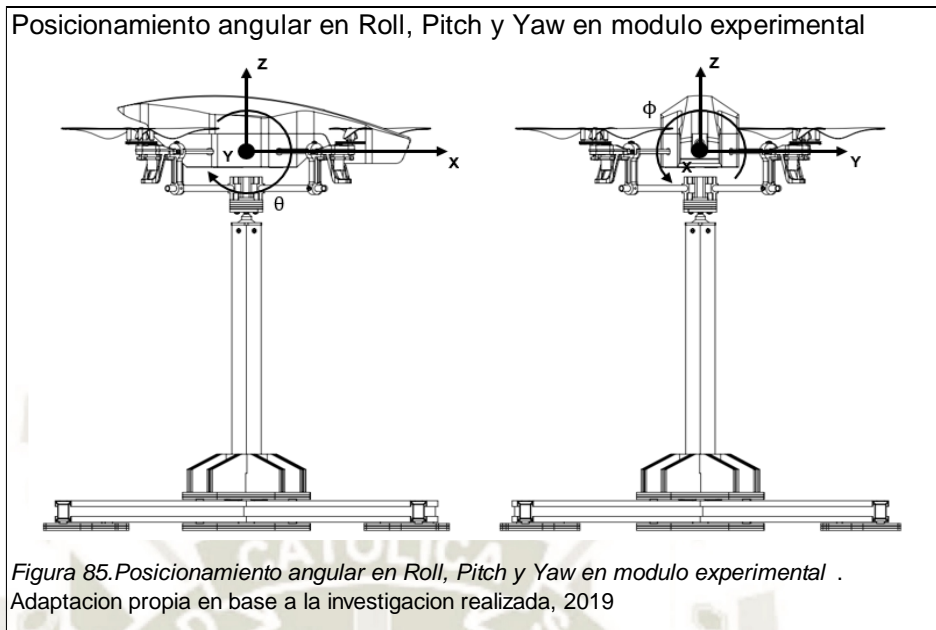


Figura 84. Posicionamiento angular en Roll, Pitch y Yaw . Adaptacion propia en base a la investigacion realizada, 2019



2.1 Control PID

Para esta experimentación se decidió hacer la sintonización del control PID mediante la sintonización experimental, del cual se obtuvo los siguientes parámetros. Ver tabla 14

Tabla 14 Valores PID en Roll, Pitch y Yaw

Tabla 14
Valores PID en Roll, Pitch y Yaw

Angulo	Dato	Valor
Roll	Kp	0.25
	Kd	0.001
	Ki	0.21
Pitch	Kp	0.25
	Kd	0.001
	Ki	0.21
Yaw	Kp	2
	Kd	0.1
	Ki	2

Nota. Datos del controlador PID en Roll, Pitch y Yaw por metodo experimental. Adaptación propia en base a la investigación realizada, 2019

Como resultado de los movimientos básicos en Roll, Pitch, Yaw en los set points, se obtuvieron los siguientes gráficos comparativos

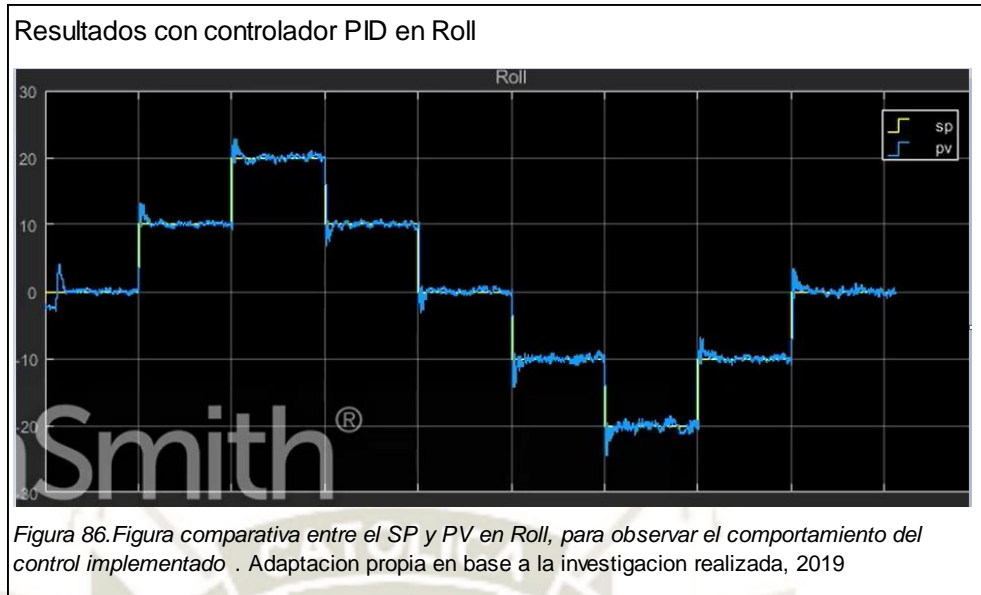
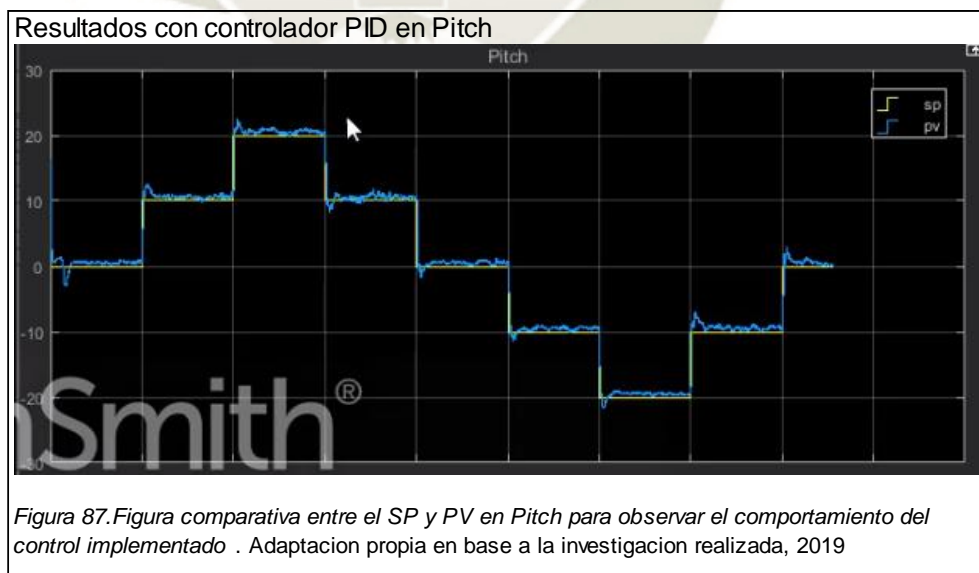


Figura 86 Resultados con controlador PID en Roll

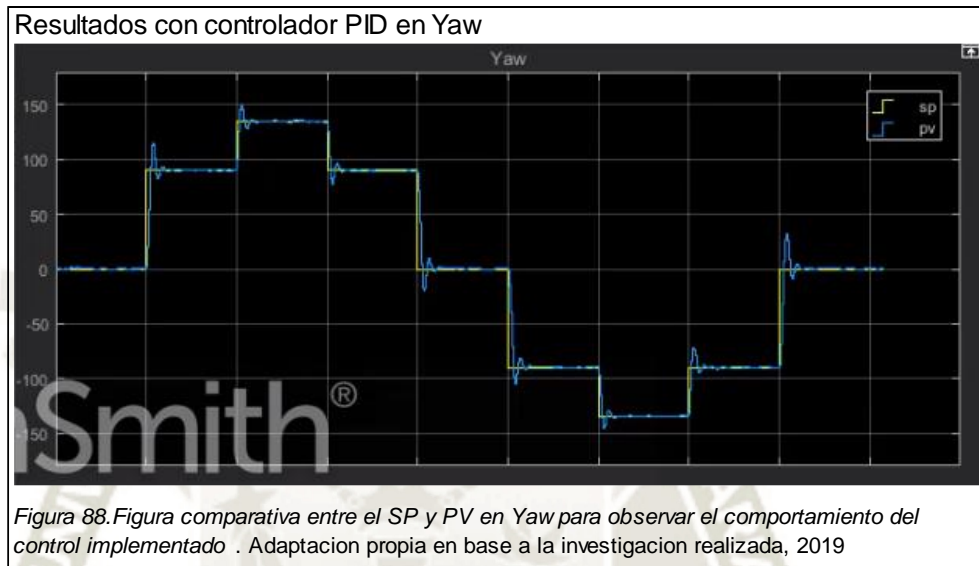
En la experimentación de roll, se observa:

- El sistema logra llegar a los puntos establecidos y que el error en estado estacionario 0 mínimo
- La respuesta tiene un sobre impulsó del 15%
- El tiempo de establecimiento es de 2s
- El tiempo de subida es de 1s



En la experimentación de pitch, se observa:

- El sistema logra llegar a los puntos establecidos y que el error en estado estacionario 0 mínimo
- La respuesta tiene un sobre impulsó del 10%
- El tiempo de establecimiento es de 2s
- El tiempo de subida es de 1s



3 Resultados con controlador PID en Yaw

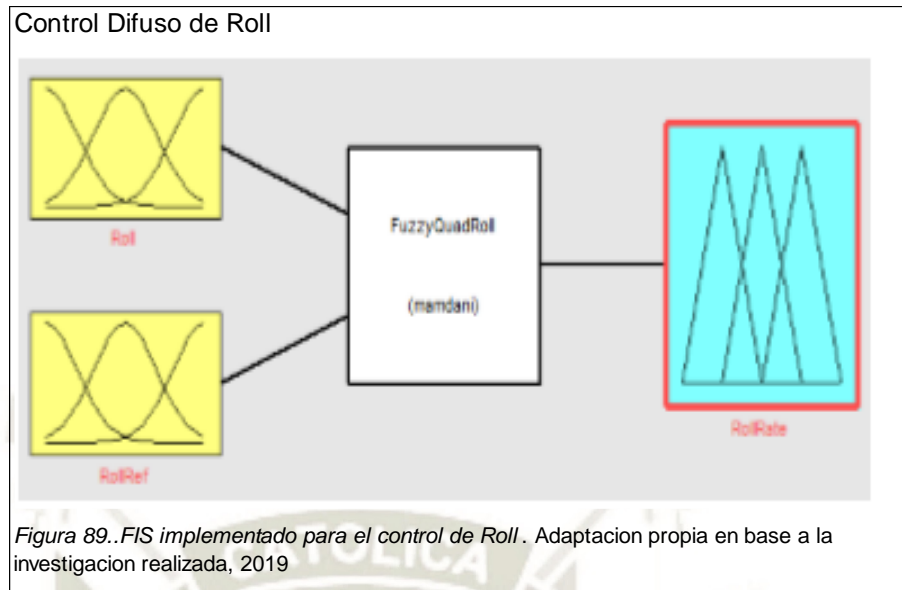
En la experimentación de yaw, se observa:

- El sistema logra llegar a los puntos establecidos y que el error en estado estacionario 0 mínimo
- La respuesta tiene un sobre impulsó del 20%
- El tiempo de establecimiento es de 2s
- El tiempo de subida es de 1s

2.2 Control DIFUSO

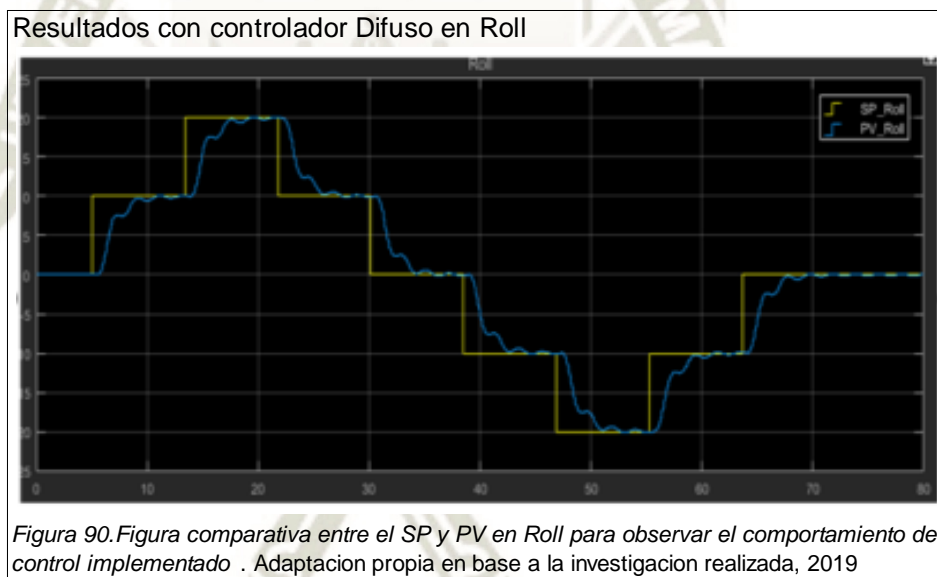
Para la experimentación se tiene que en la entrada del controlador se encuentran los valores medidos y los valores de referencia.

Para el desarrollo del control difuso en la entrada de roll se consideró para cada entrada y la salida 5 funciones de membresía de tipo triangular. Para el control se determinaron un total de 25 reglas y el rango de trabajo de las entradas es de [-30 30] y rango de trabajo de la salida es [-1 1].



Control Difuso de Roll

Como resultado de los movimientos básicos en Roll, Pitch, Yaw como set points, se obtuvieron los siguientes gráficos comparativos



Para el desarrollo del control difuso en la entrada de pitch se consideró para cada entrada y la salida 5 funciones de membresía de tipo triangular. Para el control se determinaron un total de 25 reglas y el rango de trabajo de las entradas es de $[-30 \ 30]$ y rango de trabajo de la salida es $[-1 \ 1]$.

En la experimentación de roll, se observa:

- El sistema logra llegar a los puntos establecidos y que el error en estado estacionario 0 mínimo

- El tiempo de establecimiento es de 6s
- El tiempo de subida es de 4s

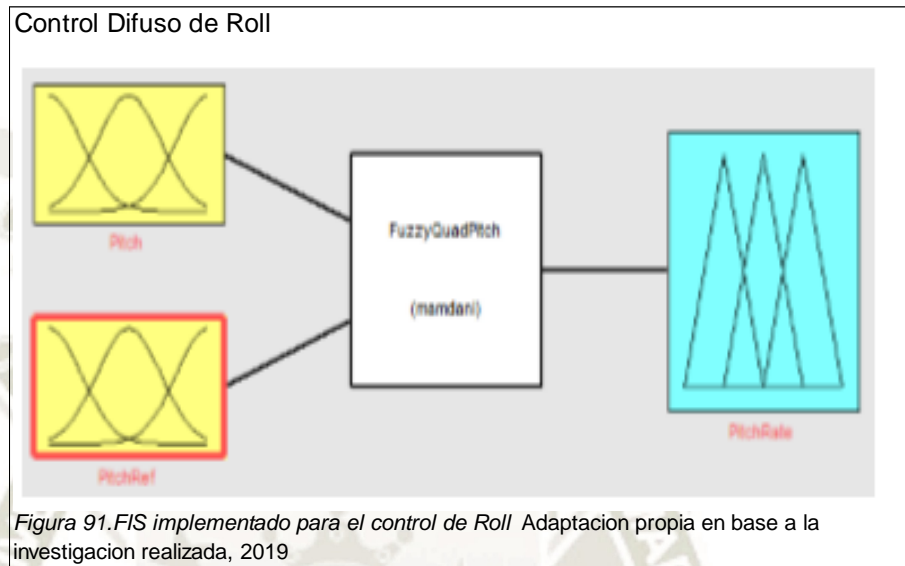
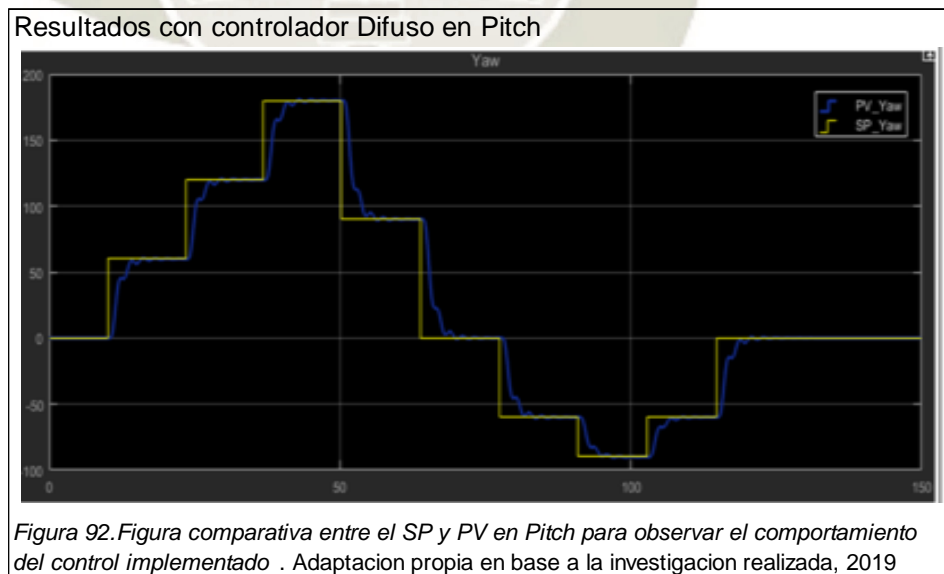


Figura 91 Control Difuso de Pitch

Como resultado de los movimientos básicos en Roll, Pitch, Yaw como set points, se obtuvieron los siguientes gráficos comparativos

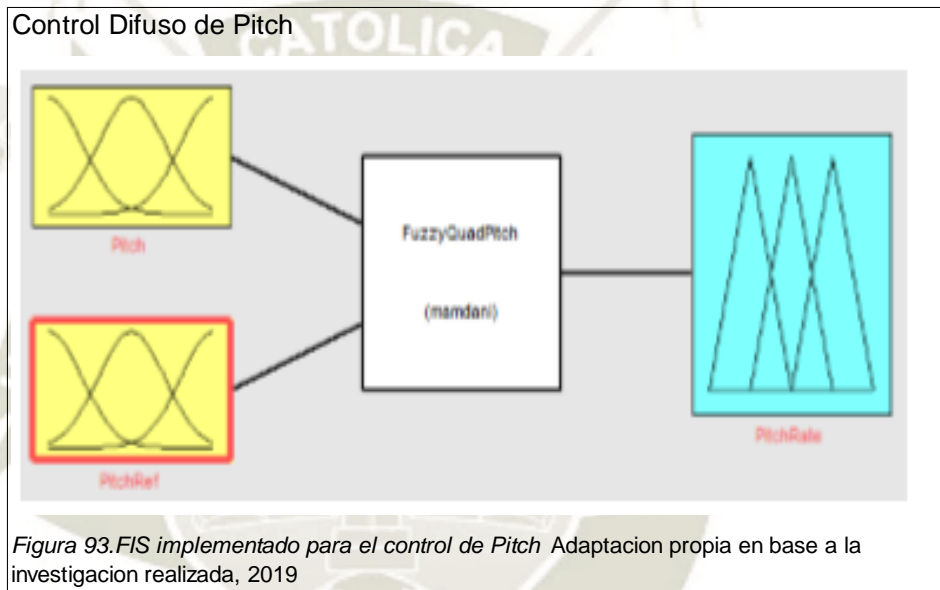


Para el desarrollo del control difuso en la entrada de roll se consideró para cada entrada y la salida 5 funciones de membresía de tipo triangular. Para el

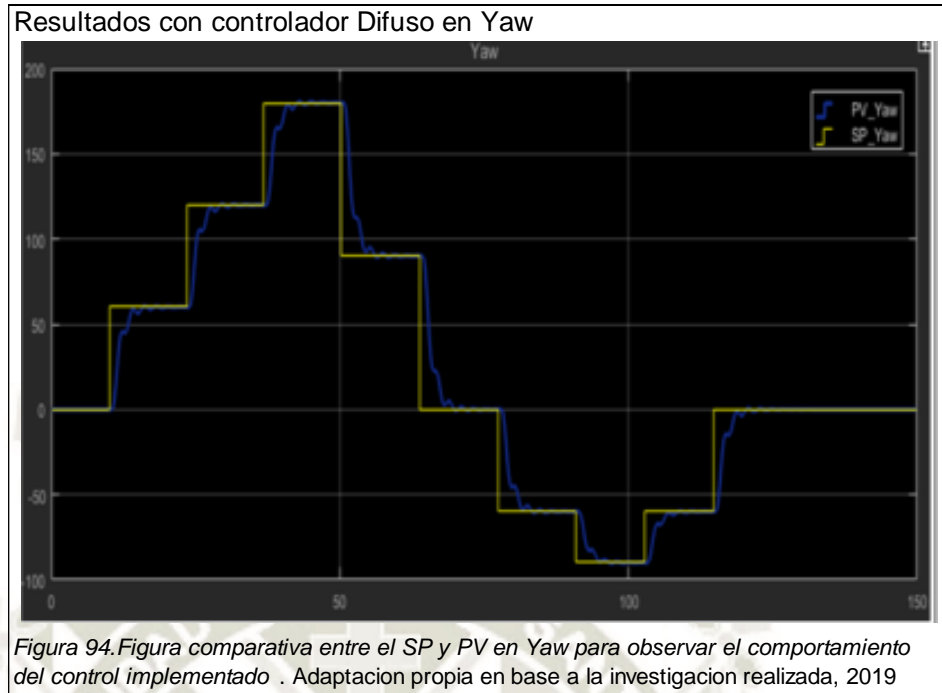
control se determinaron un total de 25 reglas y el rango de trabajo de las entradas es de $[0 \pi]$ y rango de trabajo de la salida es $[-1 \ 1]$.

En la experimentación de pitch, se observa:

- El sistema logra llegar a los puntos establecidos y que el error en estado estacionario 0 mínimo
- El tiempo de establecimiento es de 5s
- El tiempo de subida es de 3.5s



Como resultado de los movimientos básicos en Roll, Pitch, Yaw como set points, se obtuvieron los siguientes gráficos comparativos



Resultados con controlador Difuso en yaw

En la experimentación de planeación, se observa que el sistema logra llegar a los puntos establecidos. y que el error es el mínimo

En la experimentación de yaw, se observa:

- El sistema logra llegar a los puntos establecidos y que el error en estado estacionario 0 mínimo
- El tiempo de establecimiento es de 2s
- El tiempo de subida es de 1.5s

Según los datos obtenidos, se obtiene que el control PID realizado tiene mejor respuesta que el controlador difuso, se observa que posee menos sobre pulsos y que el tiempo de reacción es menor, pero ambos controles cumplen con la función principal de llegar al punto establecido y reducir el error de la variable medida con la variable deseada



CAPITULO V
ASPECTOS ADMINISTRATIVOS

1 Diagrama de Gantt:

Diagrama de Gantt

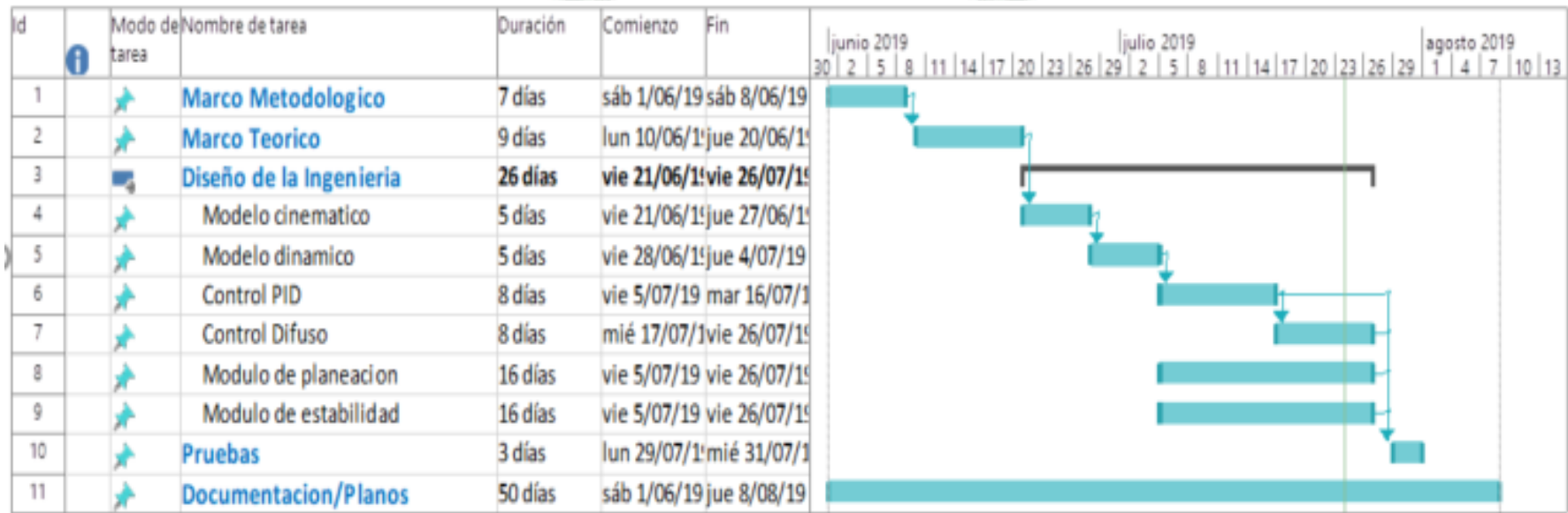


Figura 95. Diagrama de tiempos de las actividades. Adaptacion propia en base a la investigacion realizada, 2019

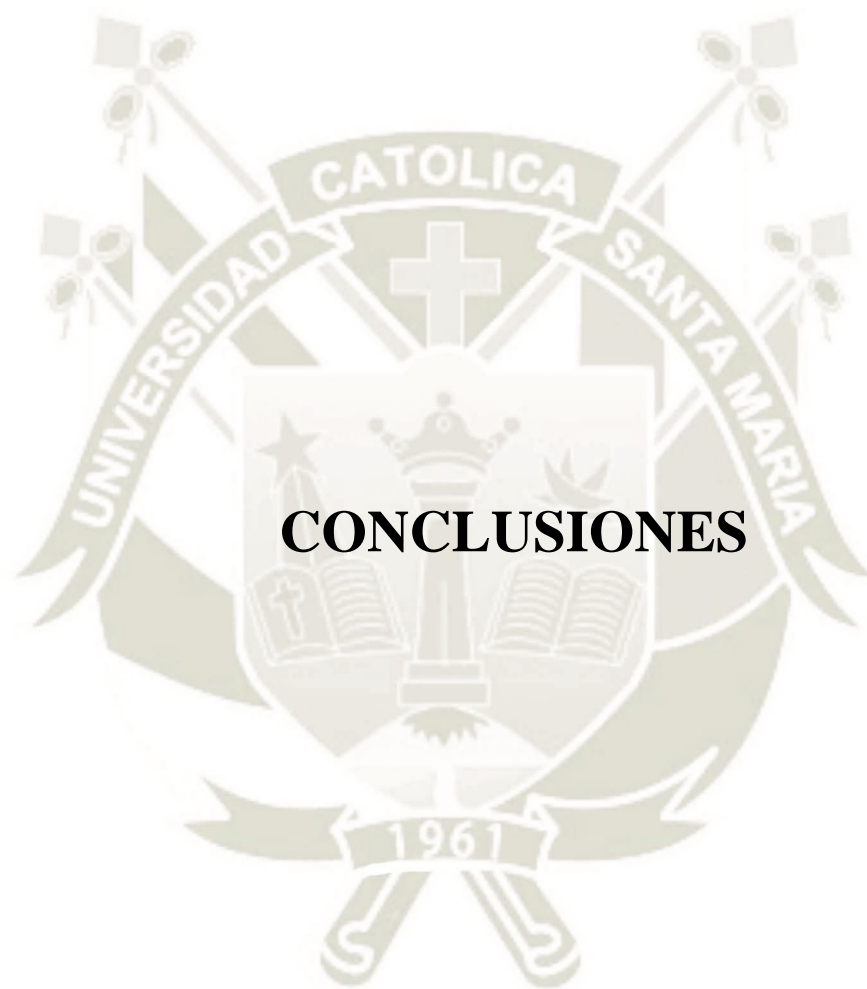
2 Costos

Tabla 15
Costos realizados

ítem	Cantidad	Costo
Ar.Drone 2.0	2	\$1,600.00
Modulo HOVER	1	\$ 250.00
Modulo STABILY	1	\$ 250.00
Laptop HP	1	\$1,500.00
Matlab	1	\$ 950.00
Totales	6	\$4,550.00

Nota. Costos realizados para la implementación de la tesis. Adaptación propia en base a la investigación realizada, 2019





CONCLUSIONES

El objetivo principal de este proyecto de investigación fue el desarrollo de un sistema de control para la obtención de un vuelo estable y seguimiento de trayectorias, el cual se alcanzó pese a las distintas dificultades que se presentaron desde el inicio del proyecto

- Se desarrolló la implementación de sistemas de control para un cuadricoptero (UAV) utilizando controladores PID como controladores difusos (FUZZY) mediante Matlab-Simulink.
- Se obtuvo el modelo matemático con distintos métodos, comprobando que el resultado es el mismo.
- Se desarrolló una interfaz para poder usar de una manera más interactiva, la cual fue realizada por un GUI de Matlab
- En total se desarrollaron 8 controladores en total, 1 control PID y 1 control Difuso por cada variable estudiada (Z, Roll, Pitch y Yaw)
- Según la experimentación se concluyó que el control PID era más preciso que el control Difuso, pero no por mucho por ende se pueden utilizar distintos tipos de control para el sistema no lineal
- En la implementación se presentaron varios problemas, por ende, se realizaron los distintos módulos de experimentación para poder así realizar las distintas pruebas de forma segura y menos riesgosas para el equipo
- Se validó los sistemas de control diseñados, mediante el seguimiento de set points establecidos, alcanzando una buena respuesta del controlador.

RECOMENDACIONES



Recomendaciones

- La medida de los sensores del cuadricoptero son mediciones estimadas por calculo debido a que no se mide directamente el desplazamiento, sino los ángulos con el giroscopio, esto podría mejorarse con el uso de las cámaras que presenta o una cámara externa
- El tiempo de autonomía por las pilas es un aspecto importante a tomar, ya que a falta de energía el sistema no responde como adecuadamente
- Debido a factores externos, el cuadricoptero podría tener fallas debido a los golpes caídas de alturas, por ello se debe de realizar un mantenimiento, con principal énfasis a las hélices de los motores
- Se recomienda utilizar el tipo de funciones de membresía tipo gaussiana para disminuir el ruido y tener cambios menos bruscos

Trabajos futuros

El avance tecnológico en el ámbito de los cuadricópteros, y de todos los vehículos autónomos está en pleno desarrollo por lo que se espera un gran campo de investigación, en este trabajo se determinaron como principales puntos a tratar a un futuro:

- Implementación del sistema de cámaras, para obtener así un control más preciso y con menor error
- El uso de una cámara como sensor para realizar seguimiento de objetos
- Desarrollar control de multidrones (sistemas de rescate)
- Generar practicas donde se permita a los alumnos de la Universidad Católica Santa María aprovechar de estos módulos para entender e implementar sus propios controles en sistemas MIMO





REFERENCIAS BIBLIOGRÁFICAS

- Barrientos, A. P. (1996). *Fundamentos de Robotica*. Madrid.
- Casanova de los Santos, A. A. (2015). *CONTROL DIFUSO DEL QUADROTOR AR.DRONE 2.0* ®. Mexico.
- Catillo, P., Lozano, R., & Dzul, A. (2005). Stabilization of a mini rotorcraft with four rotors.
- Ejes del avión*. (2010). Retrieved from Esacademic: https://esacademic.com/dic.nsf/eswiki/54301#Eje_vertical
- Fisher, C. J. (2011, 05 06). *Using An Accelerometer for Inclination Sensing* . Retrieved from Digi-key: <https://www.digikey.com/en/articles/techzone/2011/may/using-an-accelerometer-for-inclination-sensing>
- Gil, E. M. (2018). *Diseño y Construcción de un Cuadricoptero para el Monitoreo de Niveles de Contaminación Acústica y Parámetros Físicos Ambientales en la Ciudad de Arequipa*. Arequipa.
- Goering, R. (2014). Matlab edges closer to electronic design automation world. *EE Times*.
- Guts, A. K., Frolova, Y. V., & Páutova, L. A. (2013). *Métodos matemáticos en la sociología*. Moscú: URSS.
- IFR, I. F. (2019, Agosto 08). *Service Robots*. Retrieved from Internacional Federation of Robotics: <http://www.ifr.org/service-robots/>
- KEYENCE CORPORATION. (2020). *KEYENCE*. Retrieved from KEYENCE: <https://www.keyence.com.mx/ss/products/sensor/sensorbasics/ultrasonic/info/>
- Mechanical and Mechatronics Engineering Department, U. o. (2020, Setiembre 15). *WayBack Machine*. Retrieved from https://web.archive.org/web/20130209210608/http://www.metalactual.com/revista/8/tecnologia_mecatronica.pdf#

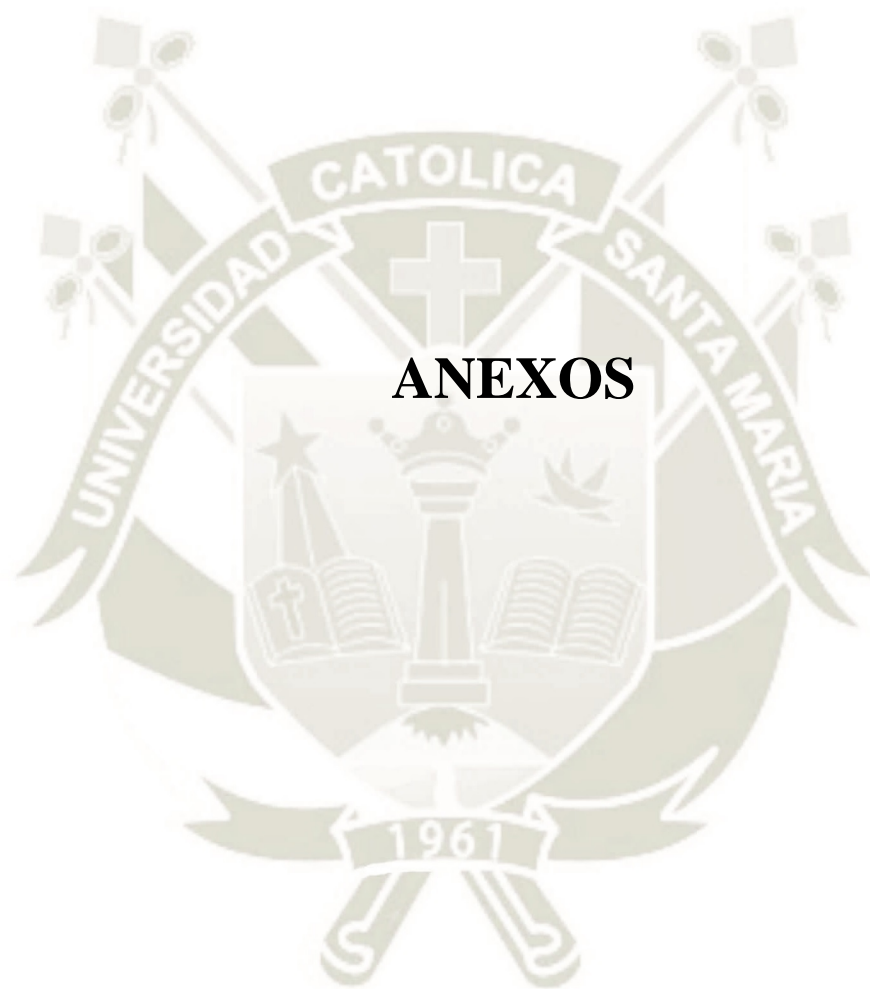
- Mobus. (2016, Marzo 18). *Mobus*. Retrieved from Mobus: <https://mobus.es/blog/que-es-una-bateria-lipo/>
- Morales, G. (2002). Introducción a la lógica difusa. *Centro de Investigación y Estudios Avanzados del IPN*.
- Ogata, . (2010). *Ingeniería de Control Moderna*. Pearson.
- Omega Engineering. (2020). *Omega Engineering*. Retrieved from Omega Engineering: <https://es.omega.com/prodinfo/acelerometro.html>
- Paiva, E. A. (2016). *Modelado y control de un cuadricóptero*. Piura.
- Pallàs Areny, R. (2005). Sensores y acondicionadores de señal. In R. Pallàs Areny, *sensores y acondicionadores de señal*. S.A. MARCOMBO.
- Pardo, C. (Sin fecha). *Controlador PID*. Retrieved from Picuino: <https://www.picuino.com/es/arduprog/control-pid.html>
- RAFFO, G. V. (2007). *Modelado y Control de un Helicóptero Quad-Rotor*. Sevilla.
- Reinoso Avecillas, M. J. (2014). *Diseño de un sistema de control por régimen deslizante para el seguimiento de trayectoria lineal de un quadrotor*. Cuenca.
- Ro-botica Global. (2017). *ro-botica*. Retrieved from ro-botica: <https://www.ro-botica.com/Producto/Sensor-giroscopo-Bioloid/>
- Rocha, J. R. (2020). *Robots y Robotica*. Retrieved from Robots y Robotica: www.robotsyrobotica.com
- Serna Ruiz, A., Ros Garcia, F., & Rico Noruega, J. c. (2010). *Guía practica de sensores*.
- The Economic Times. (2018, Marzo 26). *India fastest growing market for unmanned aerial vehicles*. Retrieved from The Economic Times: <https://economictimes.indiatimes.com/news/defence/india-fastest-growing-market-for-unmanned-aerial-vehicles/articleshow/63466658.cms>

Vildósola C, E. (n.d.). *Actuadores*. Chile: Soltex Chile S.A. .

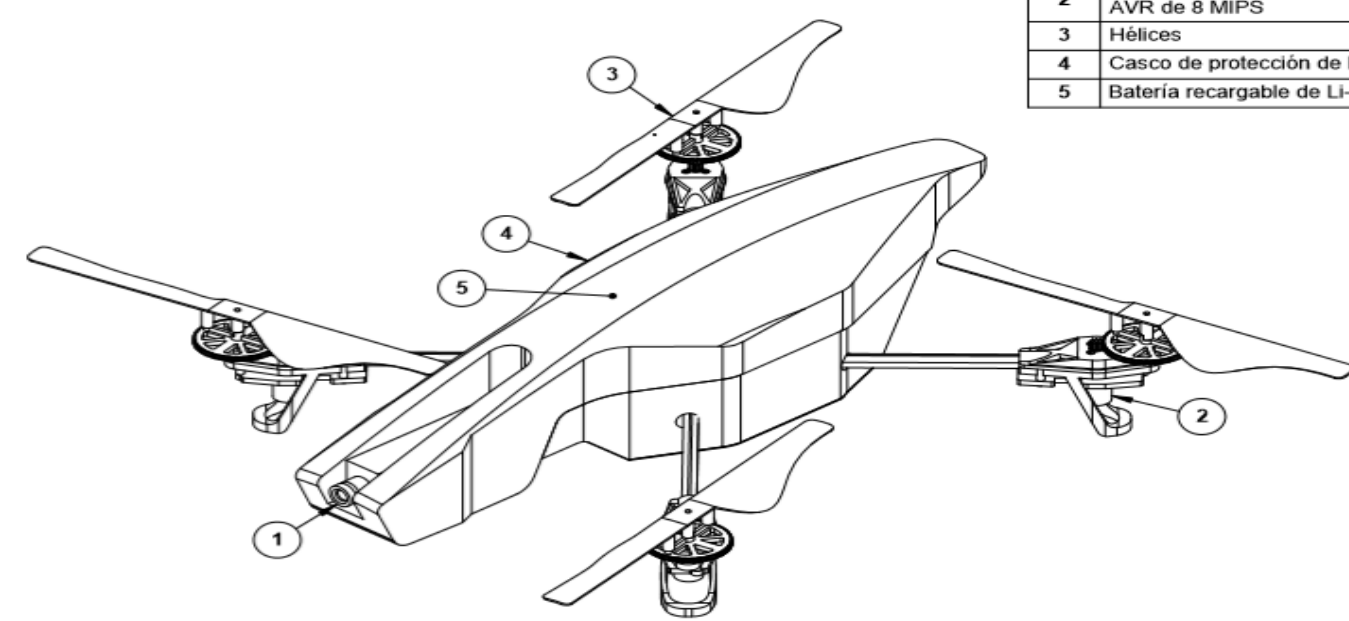

Wang, Y. B. (2006). *Advanced Fuzzy Logic Technologies in Industrial Applications*. London: Springer, .

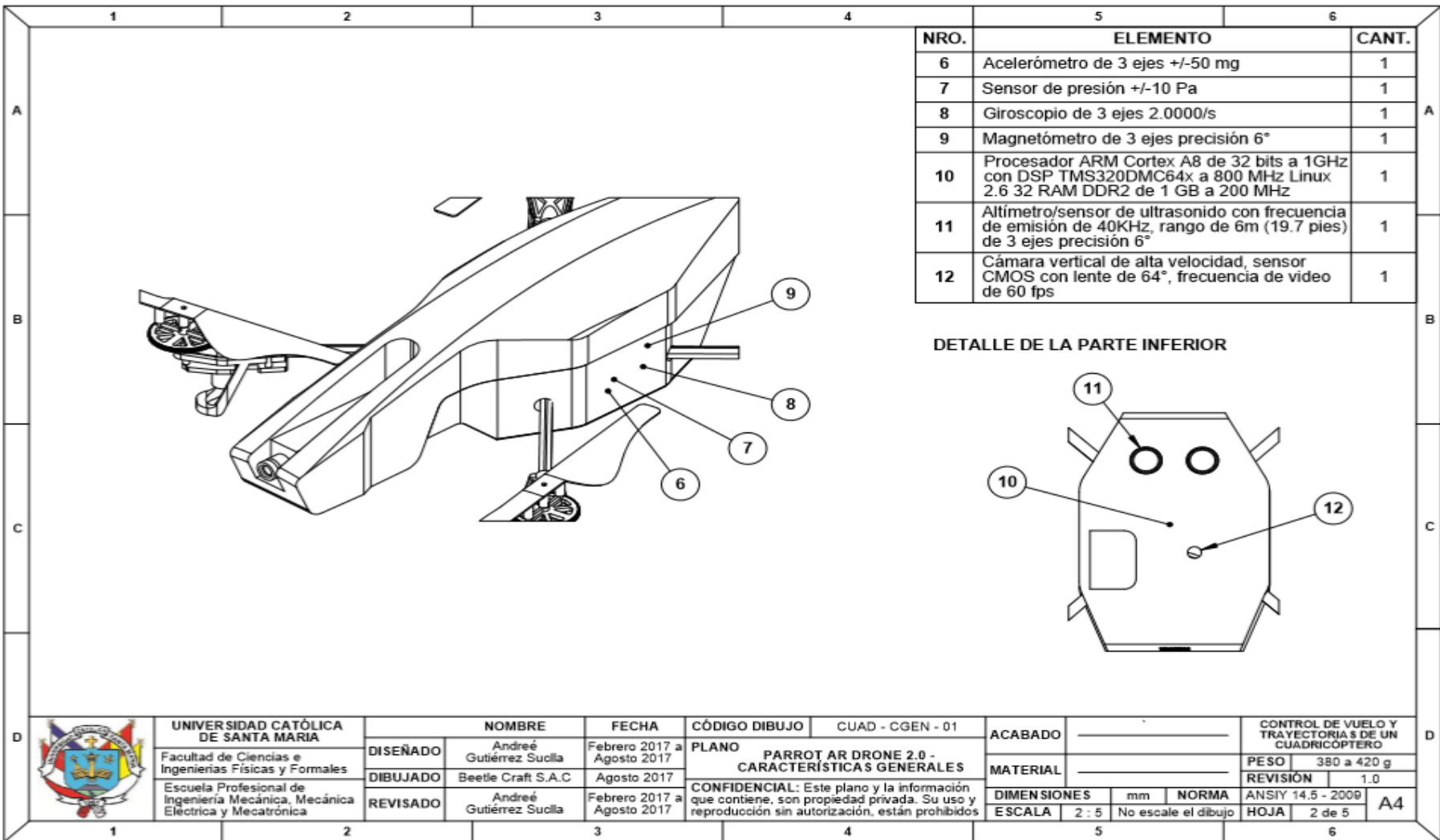


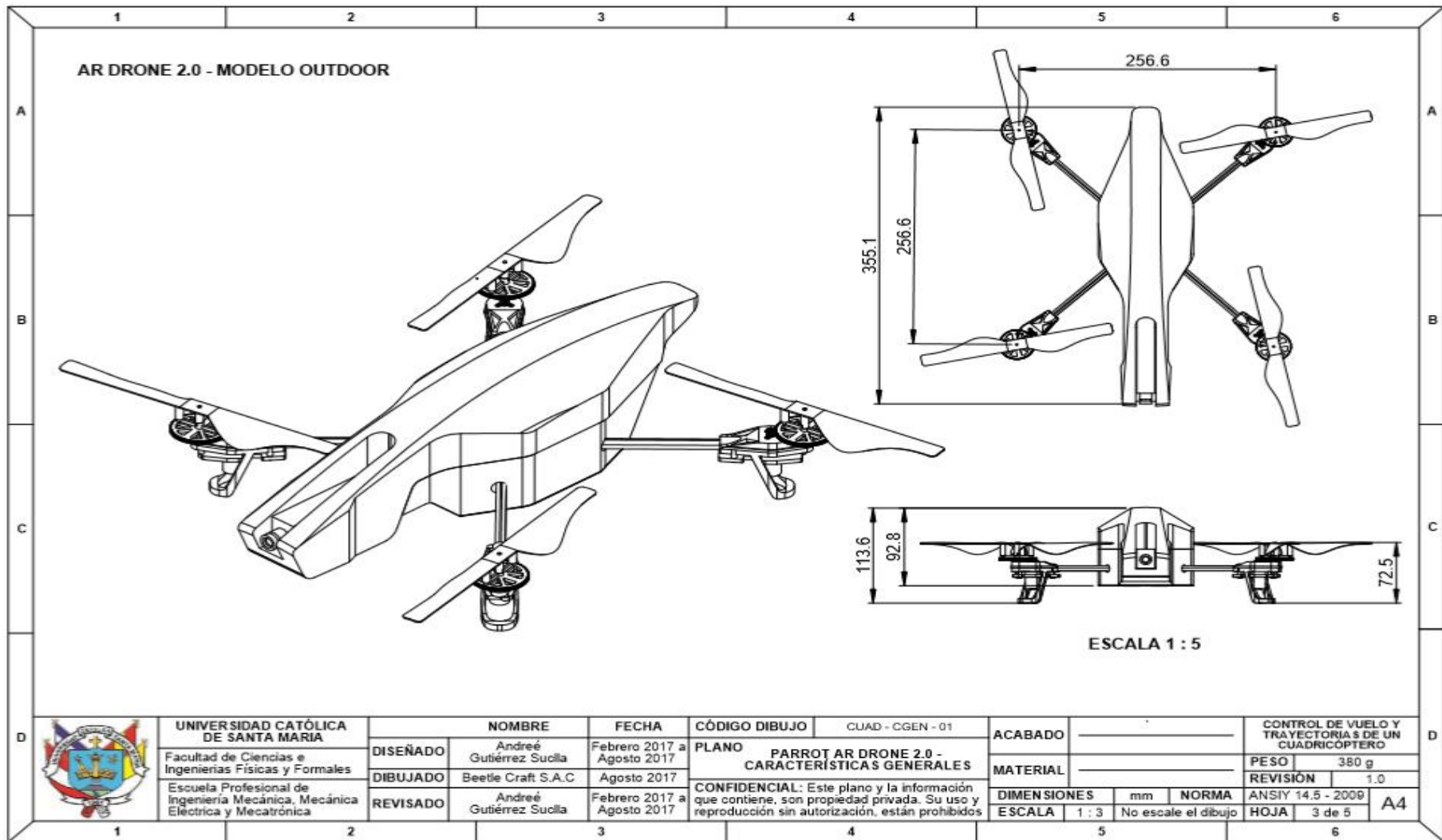




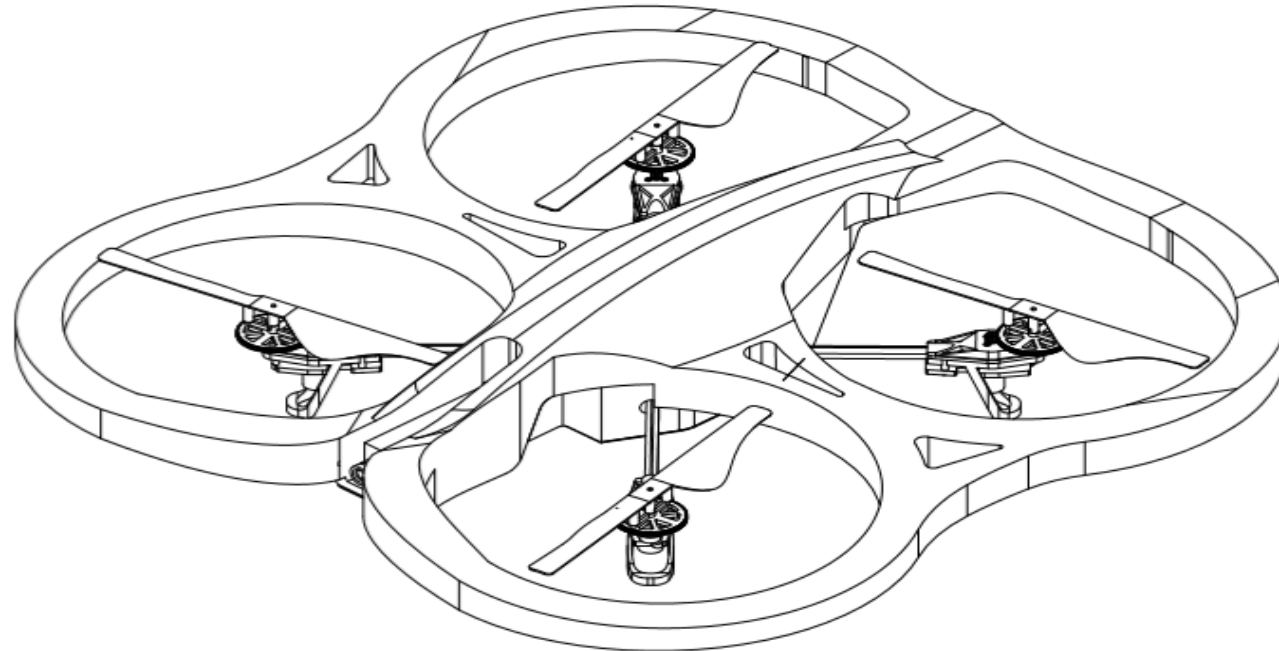
Anexo 01: PARROT AR DRONE 2.0 - Características generales

1	2	3	4	5	6					
				NRO.	ELEMENTO	CANT.				
				1	Cámara HD de 720 p de 30 fps	1				
				2	Motores brushless de 14.5W con controlador AVR de 8 MIPS	4				
				3	Hélices	4				
				4	Casco de protección de PPE	2				
				5	Batería recargable de LI-Po de 1000 mAh	1				
	UNIVERSIDAD CATÓLICA DE SANTA MARÍA		NOMBRE	FECHA	CÓDIGO DIBUJO	CUAD - CGEN - 01	ACABADO	CONTROL DE VUELO Y TRAYECTORIAS DE UN CUADRICOPTERO		
	Facultad de Ciencias e Ingenierías Físicas y Formales		André Gutiérrez Suella	Febrero 2017 a Agosto 2017	PLANO PARROT AR DRONE 2.0 - CARACTERÍSTICAS GENERALES		MATERIAL	PESO		380 a 420 g
	Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica		Beetle Craft S.A.C	Agosto 2017	CONFIDENCIAL: Este plano y la información que contiene, son propiedad privada. Su uso y reproducción sin autorización, están prohibidos		DIMENSIONES	mm	NORMA	ANSI Y 14.5 - 2009
	DISEÑADO	DIBUJADO	REVISADO	André Gutiérrez Suella	Febrero 2017 a Agosto 2017	ESCALA	2 : 5	No escale el dibujo	REVISIÓN	1.0
							HOJA	1 de 5	A4	
1	2	3	4	5	6					





AR DRONE 2.0 - MODELO INDOOR



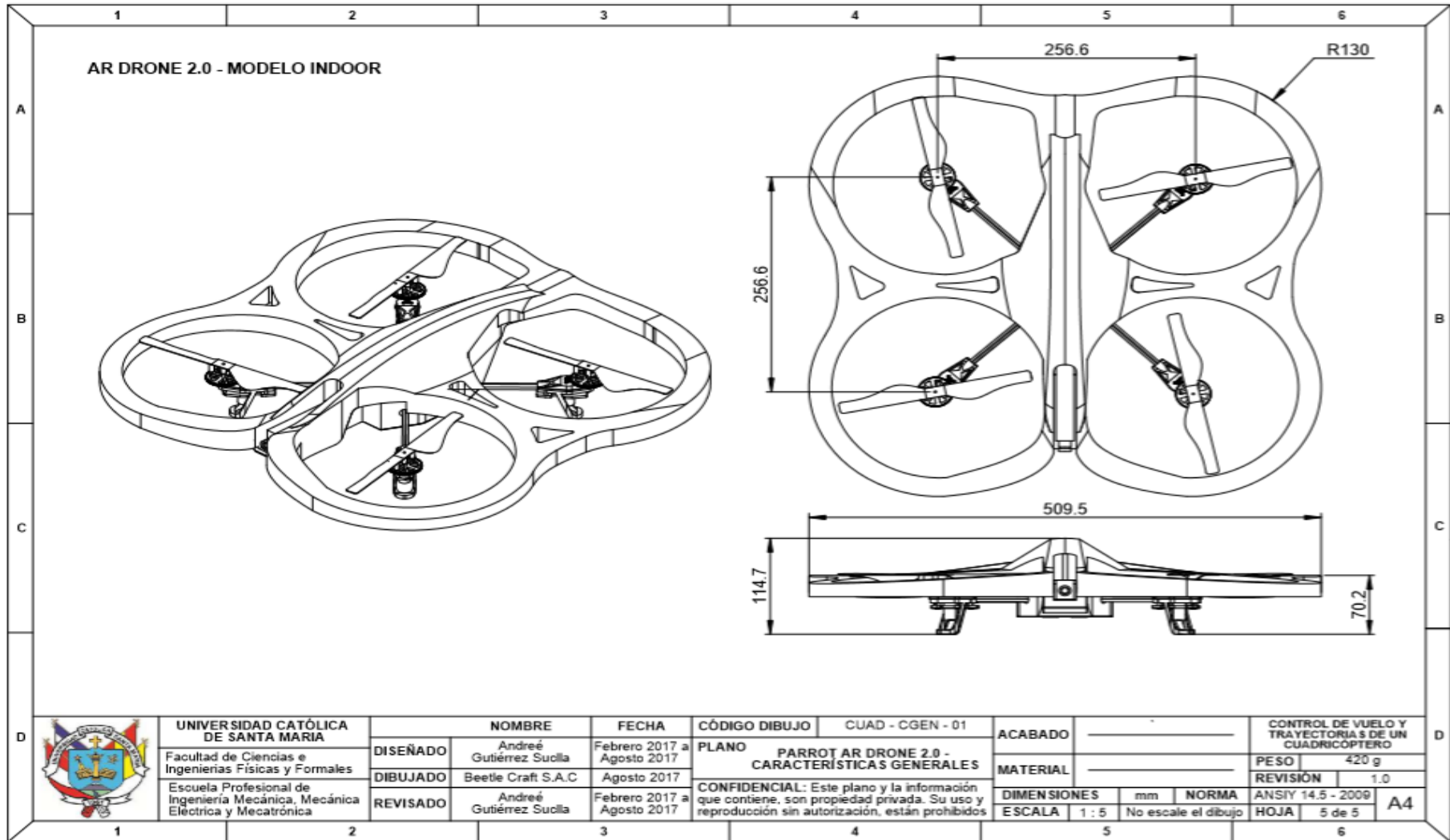
UNIVERSIDAD CATÓLICA DE SANTA MARÍA
 Facultad de Ciencias e Ingenierías Físicas y Formales
 Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica

	NOMBRE	FECHA
DISEÑADO	André Gutiérrez Suclla	Febrero 2017 a Agosto 2017
DIBUJADO	Beetle Craft S.A.C	Agosto 2017
REVISADO	André Gutiérrez Suclla	Febrero 2017 a Agosto 2017

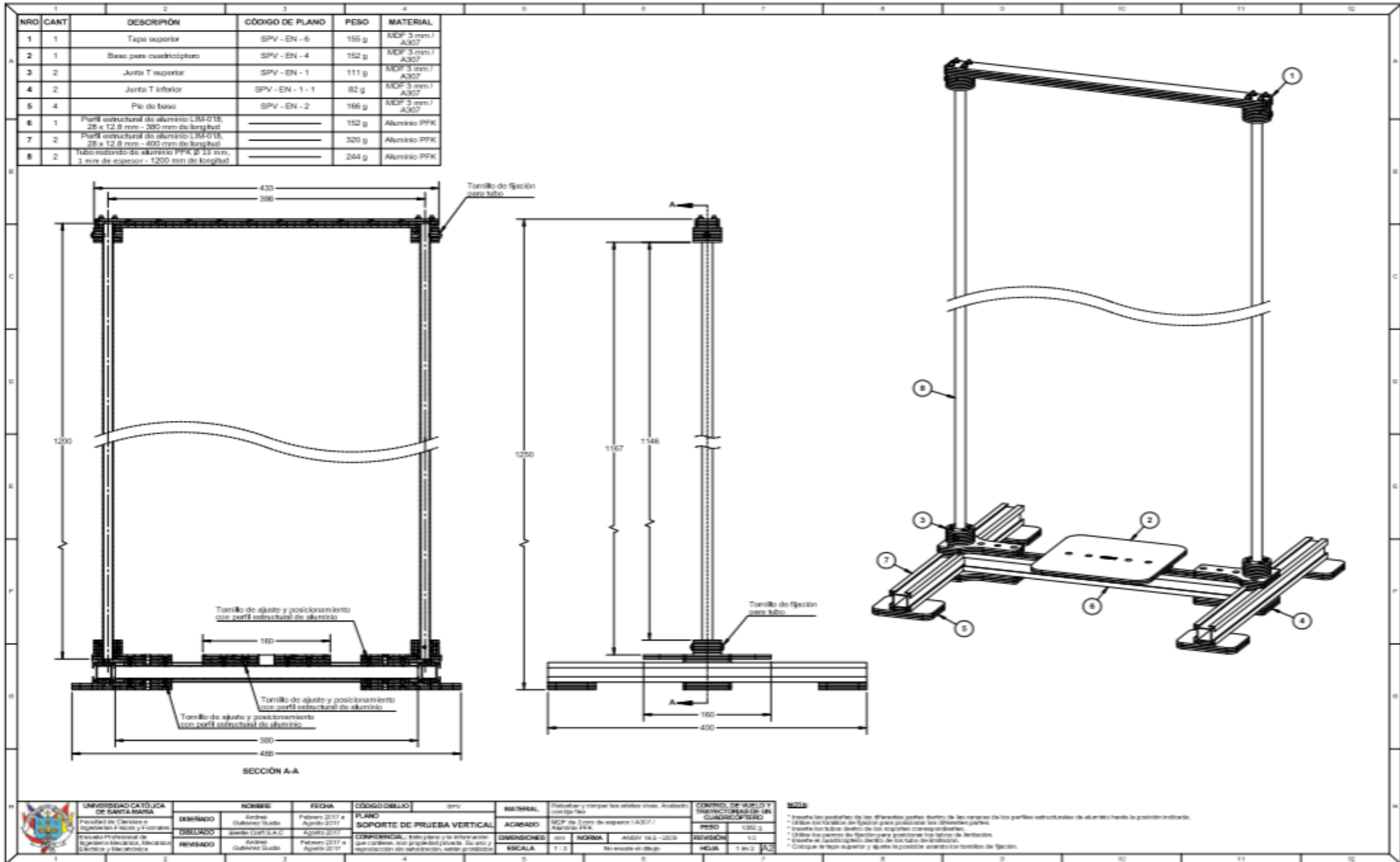
CÓDIGO DIBUJO	CUAD - CGEN - 01
PLANO PARROT AR DRONE 2.0 - CARACTERÍSTICAS GENERALES	
CONFIDENCIAL: Este plano y la información que contiene, son propiedad privada. Su uso y reproducción sin autorización, están prohibidos	

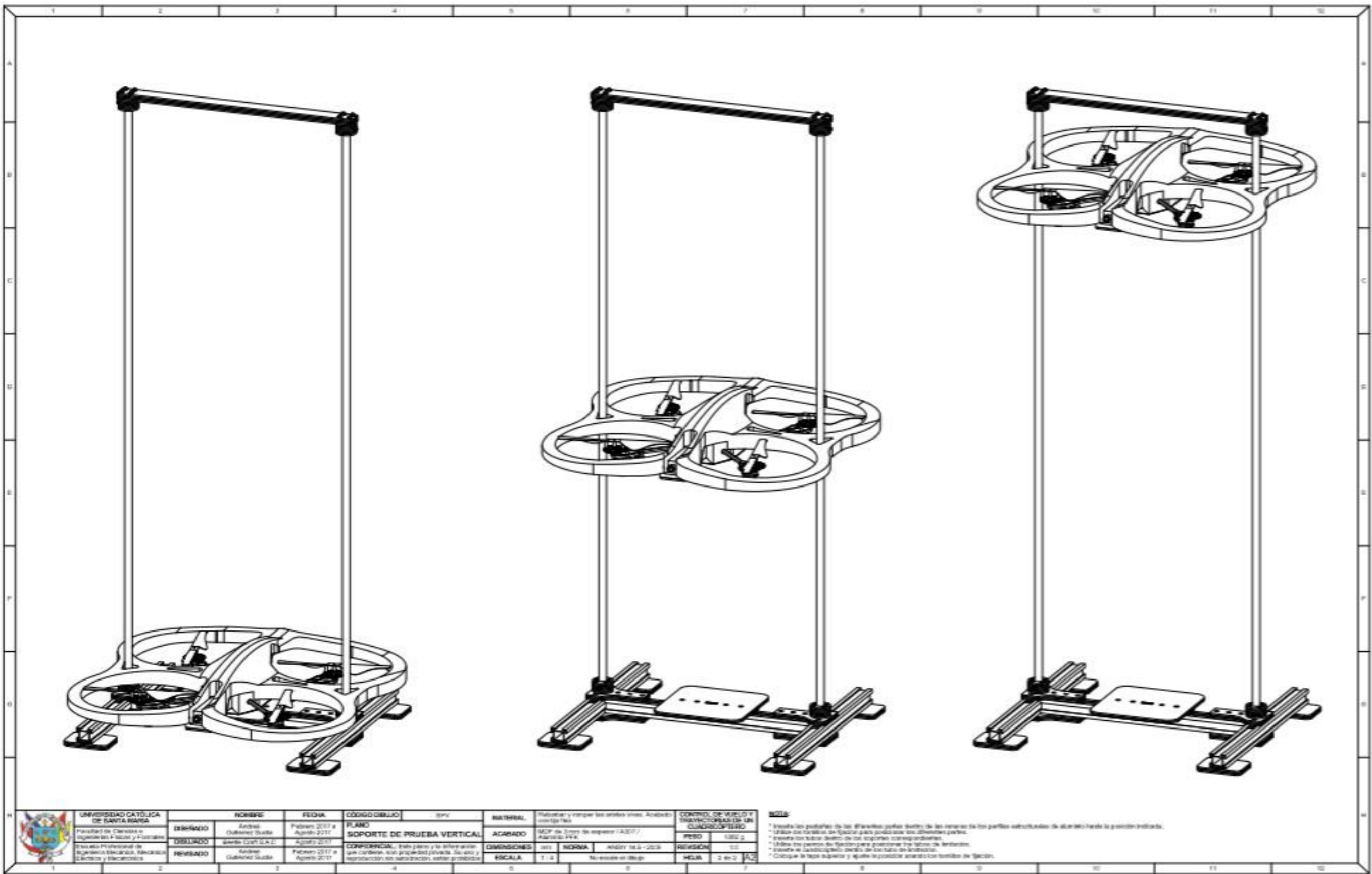
ACABADO			
MATERIAL			
DIMENSIONES	mm	NORMA	
ESCALA	1 : 3	No escale el dibujo	

CONTROL DE VUELO Y TRAYECTORIAS DE UN CUADRICÓPTERO		
PESO	420 g	
REVISION	1.0	
ANSI Y 14.5 - 2009	A4	
HOJA	4 de 5	



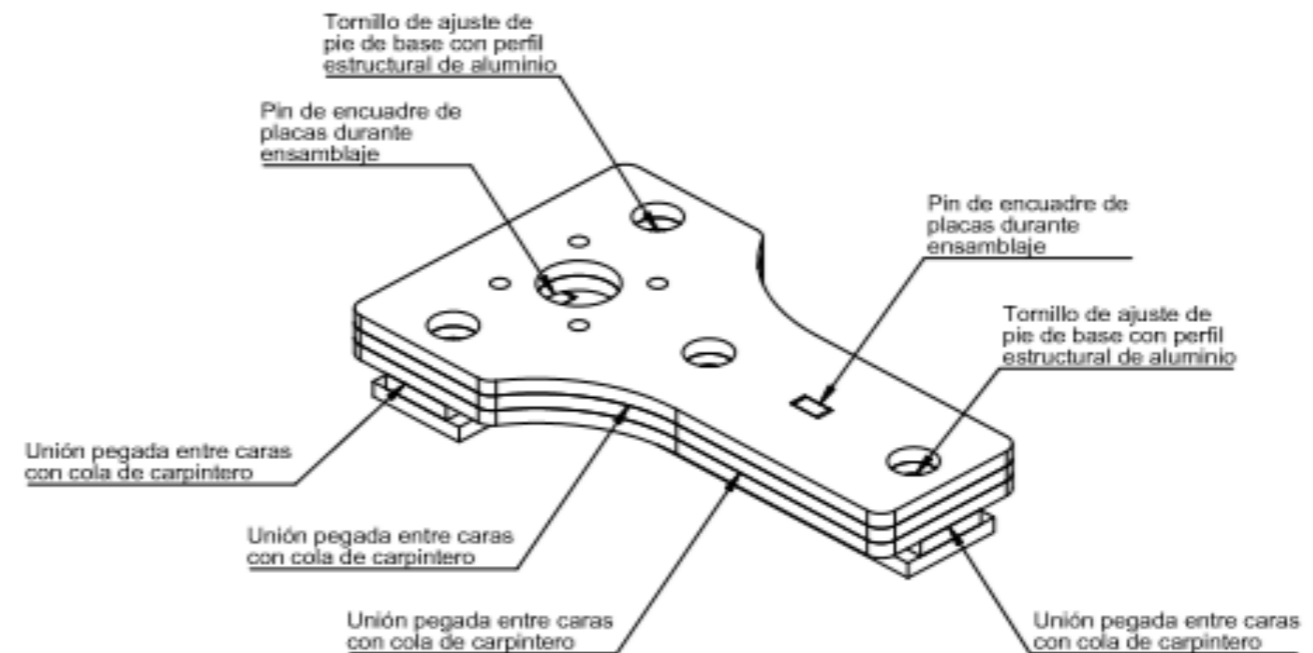
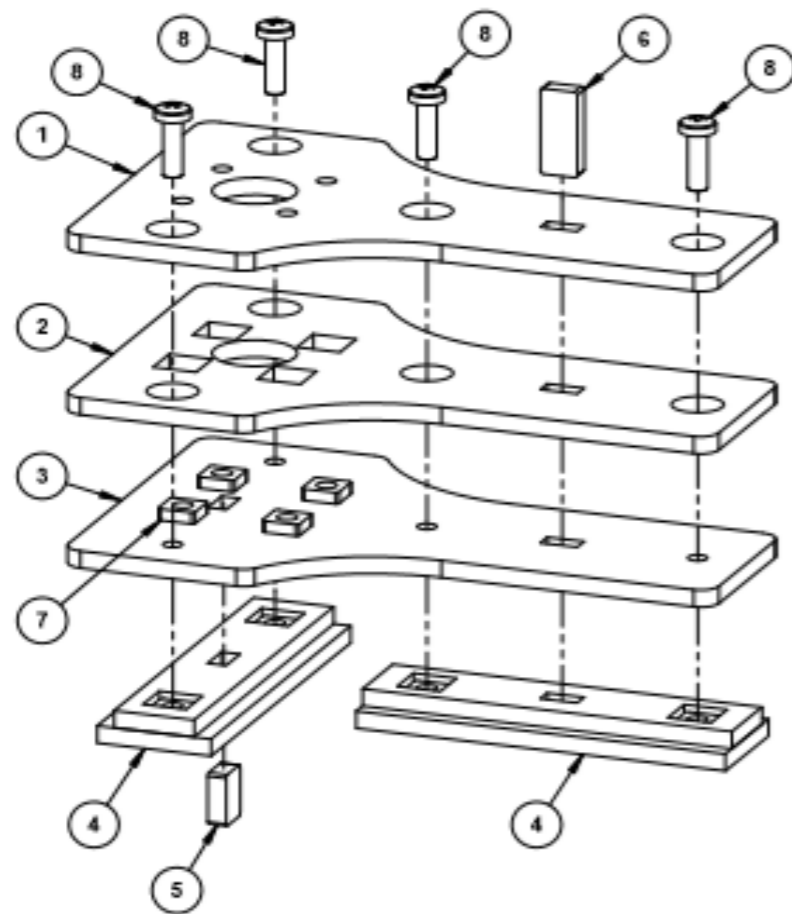
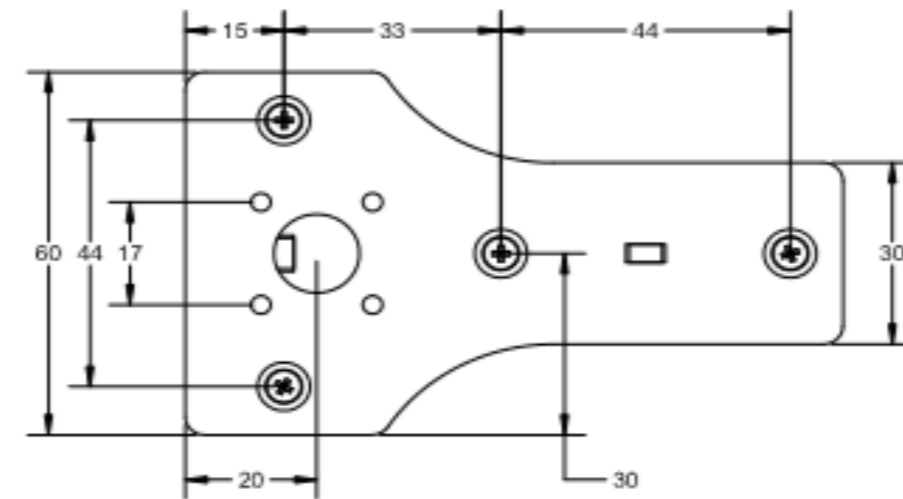
Anexo 02: Planos de soporte de prueba vertical (HOVER)





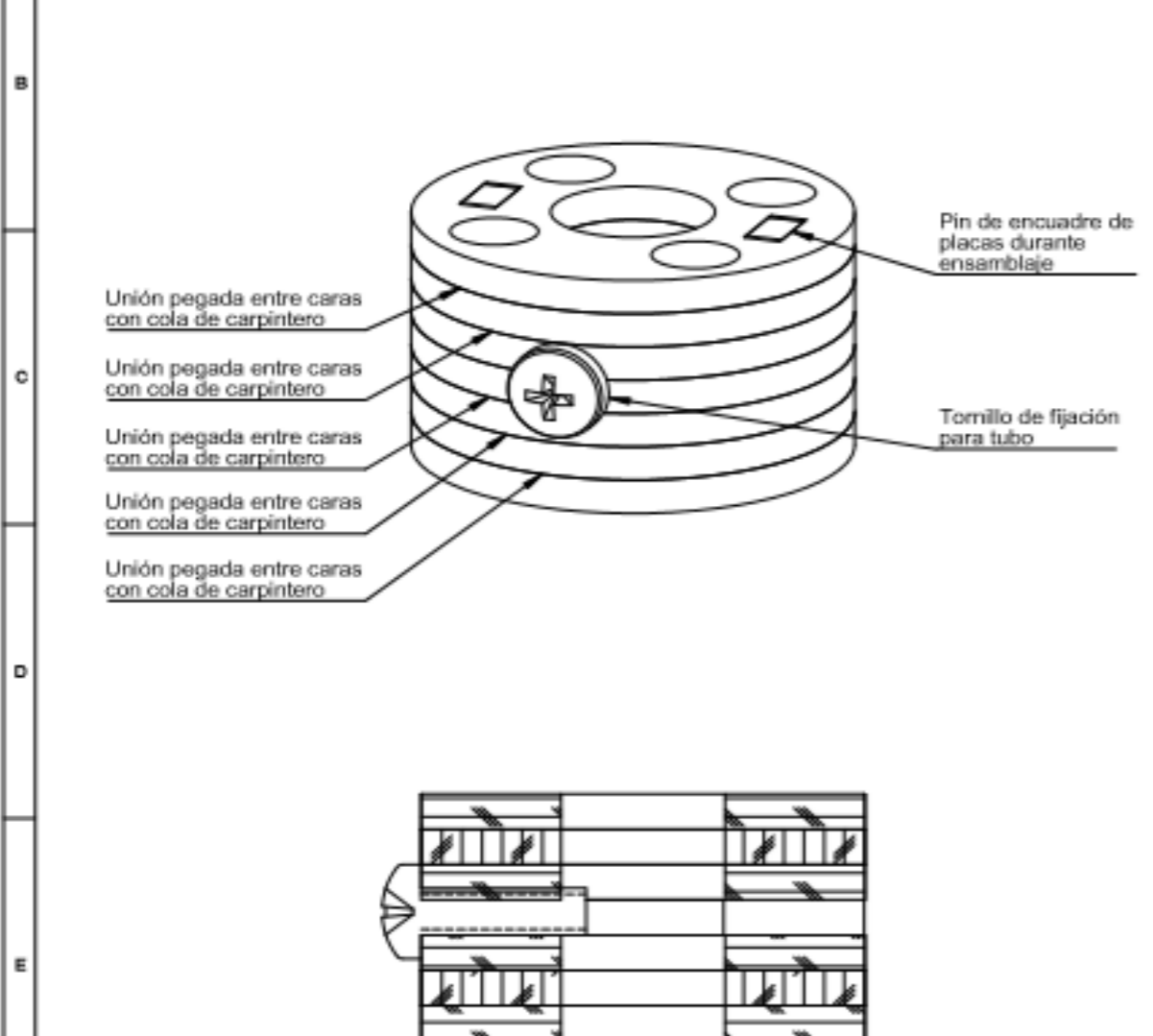
	UNIVERSIDAD CATÓLICA DE SANTA MARÍA Facultad de Ciencias e Ingeniería Escuela de Ingeniería y Arquitectura	NOMBRE Andrés Gutiérrez Suárez	FECHA Febrero 2017 a Agosto 2017	CÓDIGO DIBUJO 001	TÍTULO SOPORTE DE PRUEBA VERTICAL	MATERIAL Aluminio y acero inoxidable	ACABADO MDP de 3 mm de espesor (A307 / 6061 T6)	CONTROL DE VUELO Y TRANSMISORES DE UN CUADROPTERO
	DISEÑADO Andrés Gutiérrez Suárez	DIBUJADO Andrés Gutiérrez Suárez	REVISADO Andrés Gutiérrez Suárez	CONFERENCIA En el plano y se tiene en cuenta que con esta, sus propiedades físicas, su uso y representación en esta imagen, serán prohibidos.	DIMENSIONES mm	NORMA ASNT 10.3 - 2013	ESCALA 1:1	REVISIÓN 0.0
	PROYECTO Ingeniería de Mecánica	ASIGNATURA Mecánica de Estructuras	PROFESOR Andrés Gutiérrez Suárez	FECHA Febrero 2017 a Agosto 2017	ESCALA No existe en dibujo	NORMA ASNT 10.3 - 2013	REVISIÓN 0.0	FECHA 2 de 2 / 2017
	NOTAS * Montar los pedales de los diferentes ejes dentro de los soportes de los pedales estructurados de aluminio hacia la posición indicada. * Utilizar los tornillos de fijación para poder unir las diferentes partes. * Montar los tubos dentro de los soportes correspondientes. * Utilizar los tornillos de fijación para poder unir los tubos de aluminio. * Montar el quadcopter dentro de los tubos de aluminio. * Colocar la tapa superior y ajustar la posición de aluminio con tornillos de fijación.							

NRO	CANT	DESCRIPCIÓN	CÓDIGO DE PLANO	PE SO	MATERIAL
1	1	Junta T 3	SPV - PW - 3 - 3	10 g	MDF 3 mm
2	1	Junta T 1	SPV - PW - 3 - 1	9 g	MDF 3 mm
3	1	Junta T 2	SPV - PW - 3 - 2	11 g	MDF 3 mm
4	2	Base de apoyo	SPV - EN - 5	10 g	MDF 3 mm / A307
5	1	Pin de anclaje 4	SPV - PW - 6 - 4	0.5 g	MDF 3 mm
6	1	Pin de anclaje 1	SPV - PW - 6 - 1	0.5 g	MDF 3 mm
7	4	M3 TUERCA CUADRADA			A307
8	4	M3x10 TORNF CABCIL PHILLIPS			A307

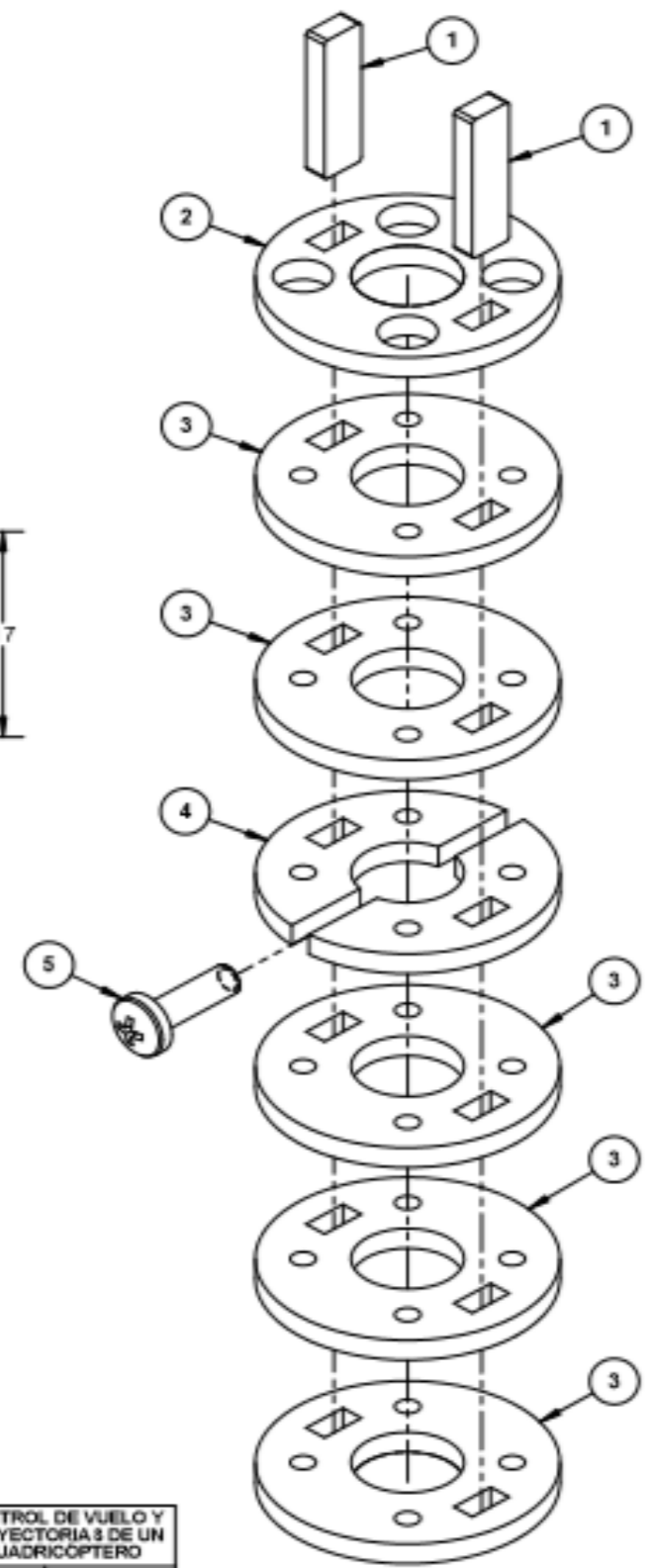
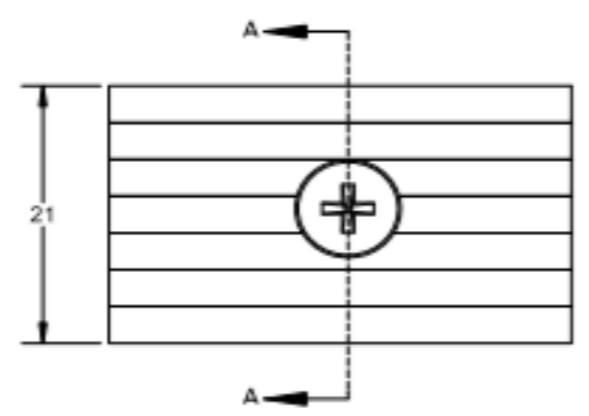
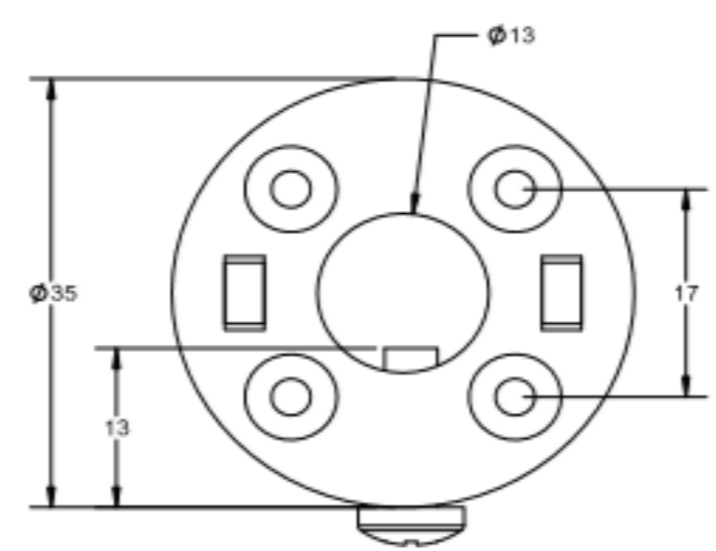


	UNIVERSIDAD CATÓLICA DE SANTA MARÍA		NOMBRE	FECHA	CÓDIGO DIBUJO	SPV - EN - 1 - 1	MATERIAL	Rebarbar y romper las aristas vivas. Acabado con lija fina		CONTROL DE VUELO Y TRAYECTORIAS DE UN CUADRICOPTERO			
	Facultad de Ciencias e Ingenierías Físicas y Formales		DISEÑADO	André Gutiérrez Suñá	Febrero 2017 a Agosto 2017	PLANO	JUNTA T INFERIOR	ACABADO	MDF de 3 mm de espesor / A307		PE SO	41 g	
	Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica		DIBUJADO	Bestie Craft S.A.C	Agosto 2017	CONFIDENCIAL: Este plano y la información que contiene, son propiedad privada. Su uso y reproducción sin autorización, están prohibidos		DIMENSIONES	mm	NORMA	ANSI Y 14.5 - 2009	REVISIÓN	1.0
			REVISADO	André Gutiérrez Suñá	Febrero 2017 a Agosto 2017			ESCALA	1 : 1	No escale el dibujo		HOJA	1 de 1 A3

NRO	CANT	DESCRIPCIÓN	CÓDIGO DE PLANO	PE SO	MATERIAL
1	1	Pin de anclaje 2	SPV - PW - 6 - 2	1 g	MDF 3 mm
2	1	Soporte de tubo 3	SPV - PW - 5 - 3	1.5 g	MDF 3 mm
3	5	Soporte de tubo 1	SPV - PW - 5 - 1	10 g	MDF 3 mm
4	1	Soporte de tubo 2	SPV - PW - 5 - 2	2 g	MDF 3 mm
5	1	M4x13 TORNIF CABCIL PHILLIPS			A307



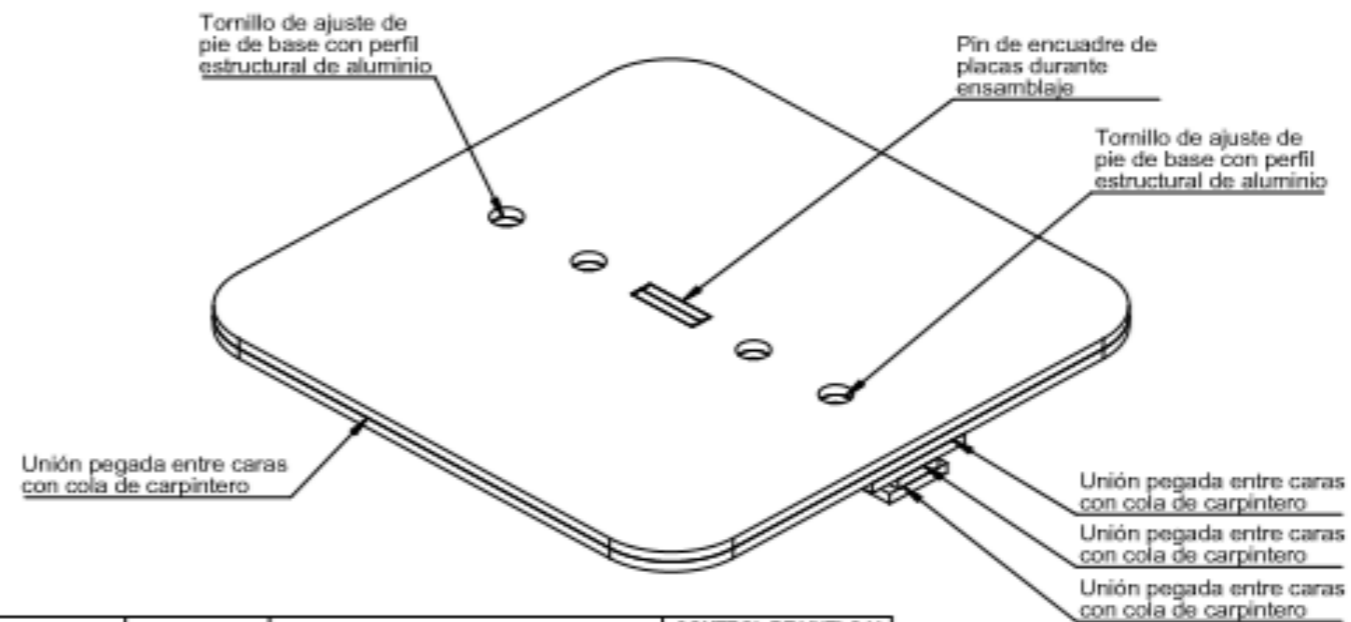
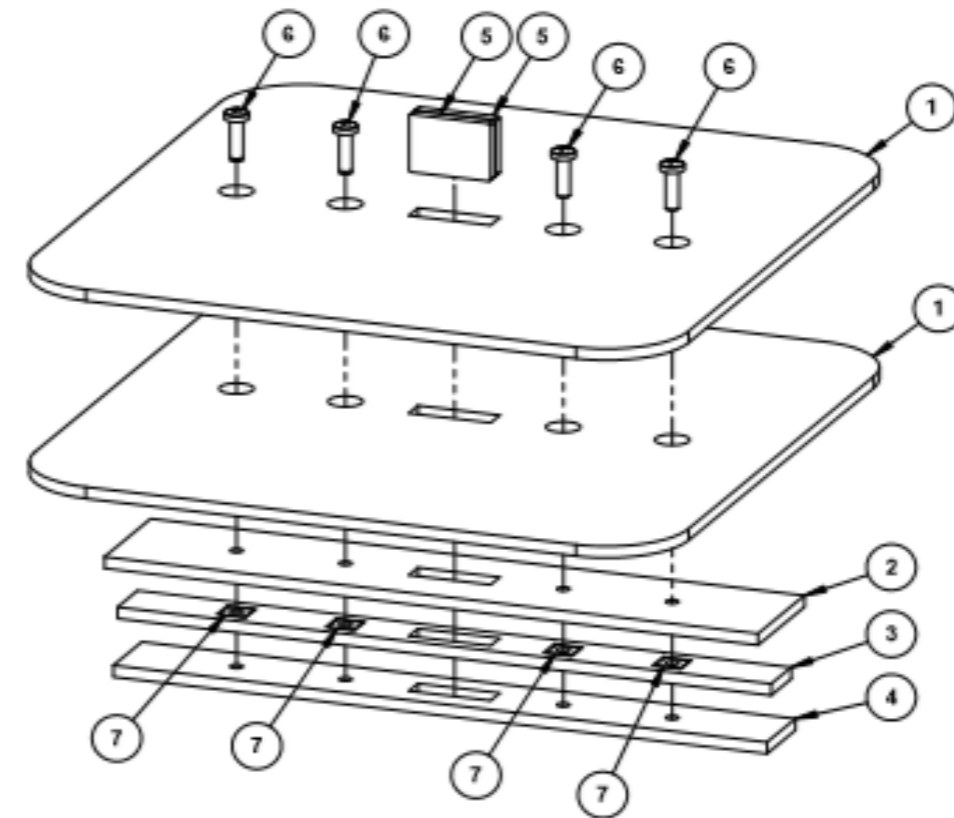
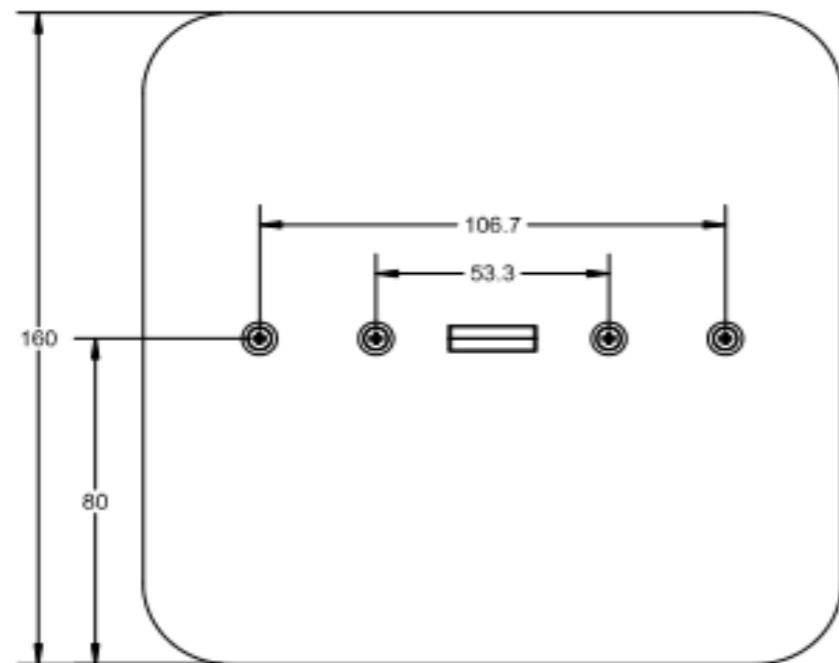
SECCIÓN A-A
ESCALA 2 : 1



ESCALA 4 : 3

	UNIVERSIDAD CATÓLICA DE SANTA MARÍA	NOMBRE	FECHA	CÓDIGO DIBUJO	SPV - EN - 1 - 2	MATERIAL	Rebarbar y romper las aristas vivas. Acabado con lija fina			CONTROL DE VUELO Y TRAYECTORIAS DE UN CUADRICOPTERO		
	Facultad de Ciencias e Ingenierías Físicas y Formales	DISEÑADO	André Gutiérrez Suñla	Febrero 2017 a Agosto 2017	PLANO	ACABADO	MDF de 3 mm de espesor / A307			PESO	14.5 g	
	Escola Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica	DIBUJADO	Beetle Craft S.A.C	Agosto 2017		DIMENSIONES	mm	NCRMA	ANSI 14.5 - 2009		REVISIÓN	1.0
		REVISADO	André Gutiérrez Suñla	Febrero 2017 a Agosto 2017	CONFIDENCIAL: Este plano y la información que contiene, son propiedad privada. Su uso y reproducción sin autorización, están prohibidos.	ESCALA	2 : 1			No escale el dibujo	HOJA	1 de 1 A3

NRO	CANT	DESCRIPCIÓN	CÓDIGO DE PLANO	PESO	MATERIAL
1	2	Base de cuadricóptero	SPV - PW - 8	126 g	MDF 3 mm
2	1	Soporte para base de cuadricóptero 1	SPV - PW - 9	12 g	MDF 3 mm
3	1	Soporte para base de cuadricóptero 3	SPV - PW - 11	5 g	MDF 3 mm
4	1	Soporte para base de cuadricóptero 2	SPV - PW - 10	7 g	MDF 3 mm
5	2	Pin de anclaje 3	SPV - PW - 6 - 3	2 g	MDF 3 mm
6	4	M3x10 TORNILLO CABCIL PHILLIPS			A307
7	4	M3 TUERCA CUADRADA			A307



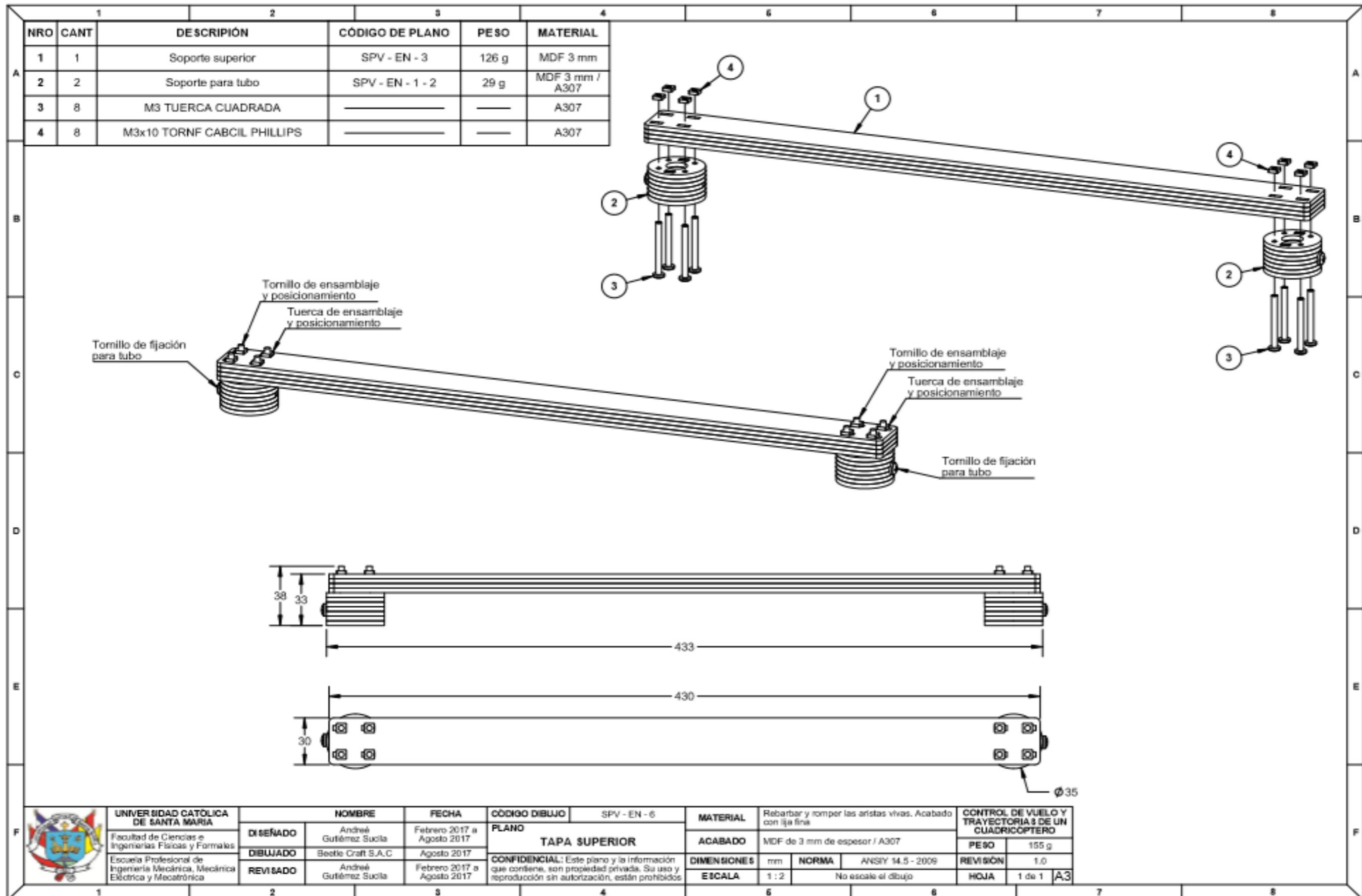
UNIVERSIDAD CATÓLICA DE SANTA MARÍA
Facultad de Ciencias e Ingenierías Físicas y Formales
Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica

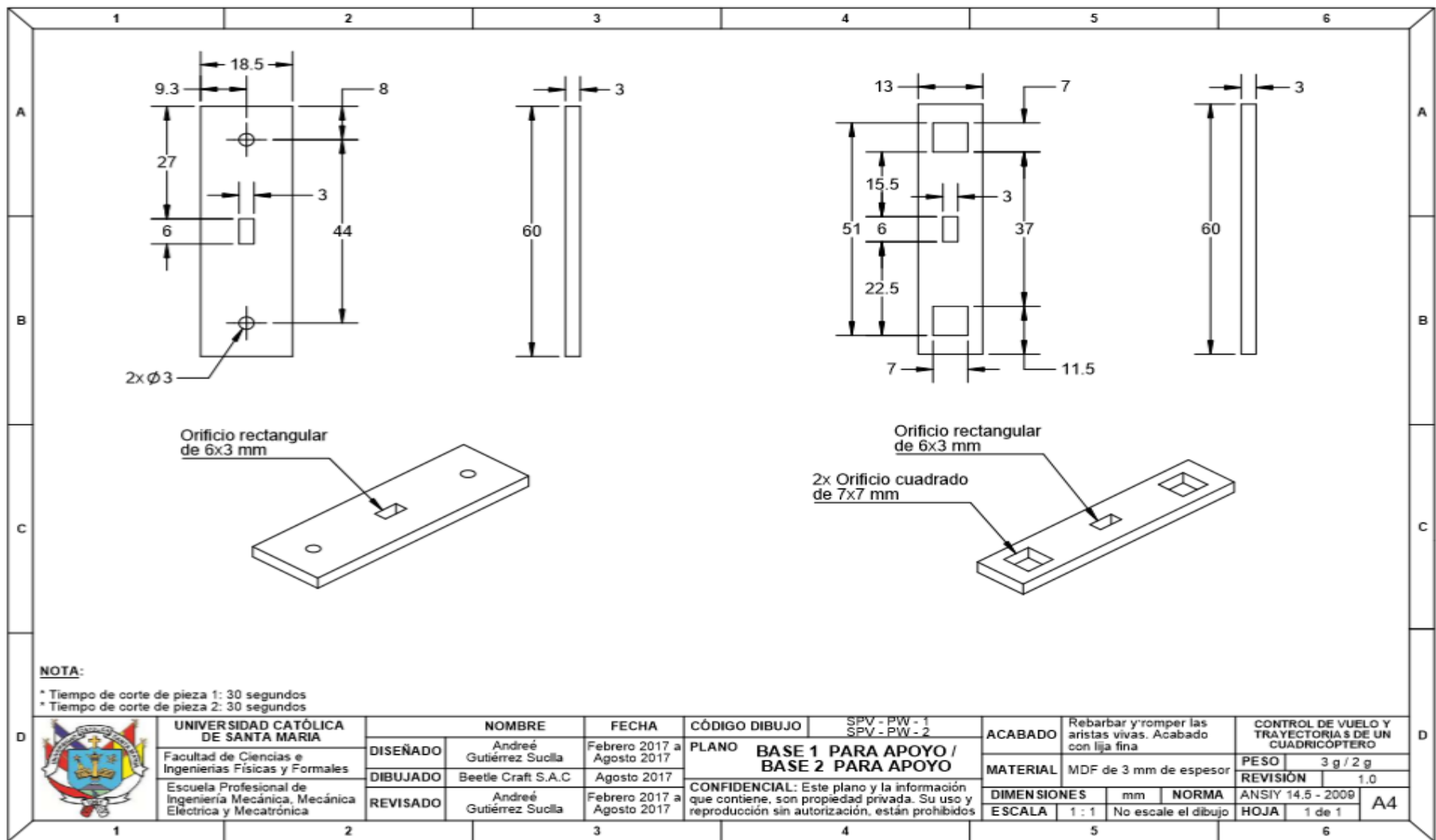
NOMBRE	FECHA	CÓDIGO DIBUJO
André Gutiérrez Suñla	Febrero 2017 a Agosto 2017	SPV - EN - 4
Beetle Craft S.A.C	Agosto 2017	
André Gutiérrez Suñla	Febrero 2017 a Agosto 2017	

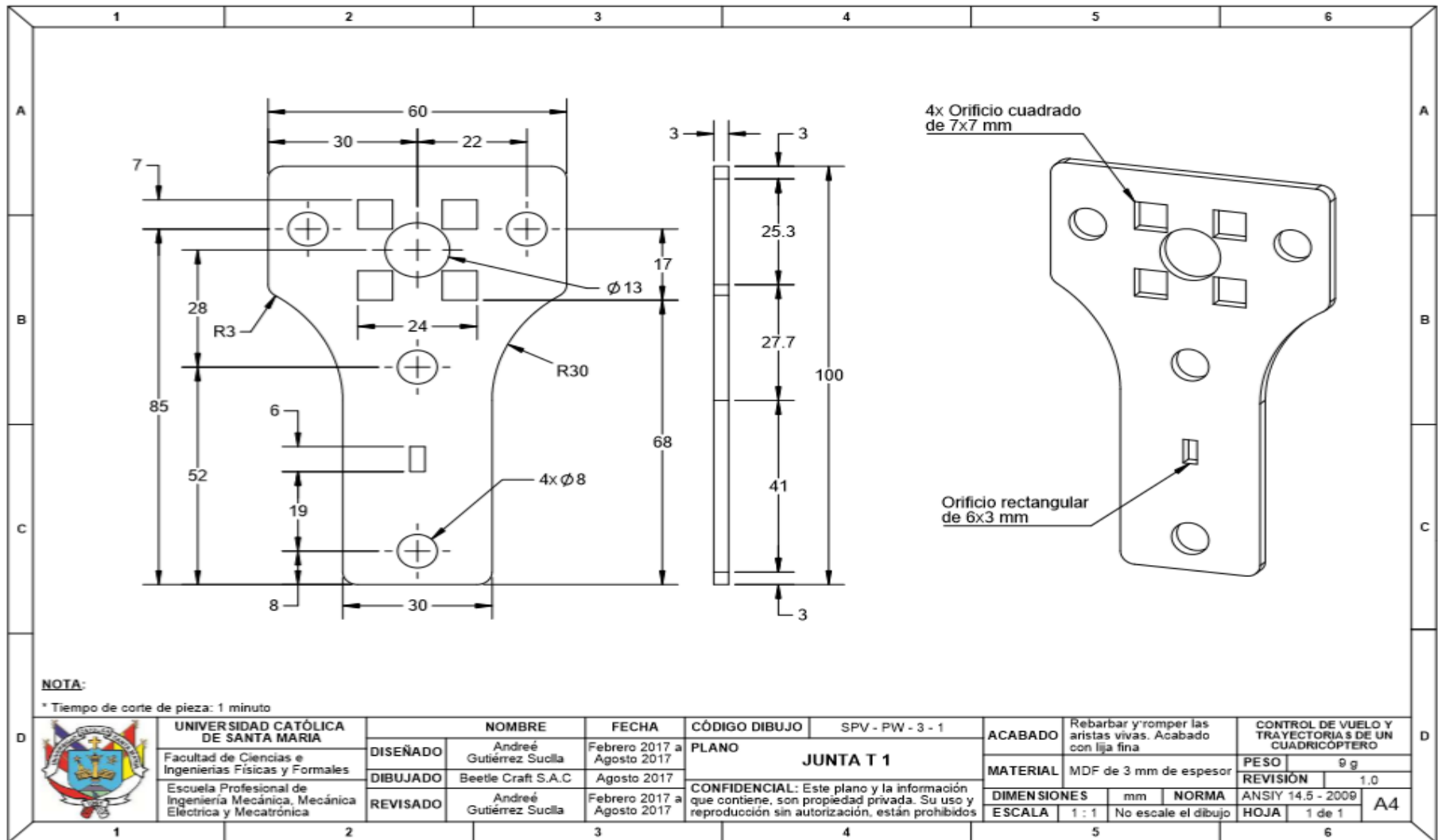
PLANO
BASE PARA CUADRICÓPTERO
CONFIDENCIAL: Este plano y la información que contiene, son propiedad privada. Su uso y reproducción sin autorización, están prohibidos

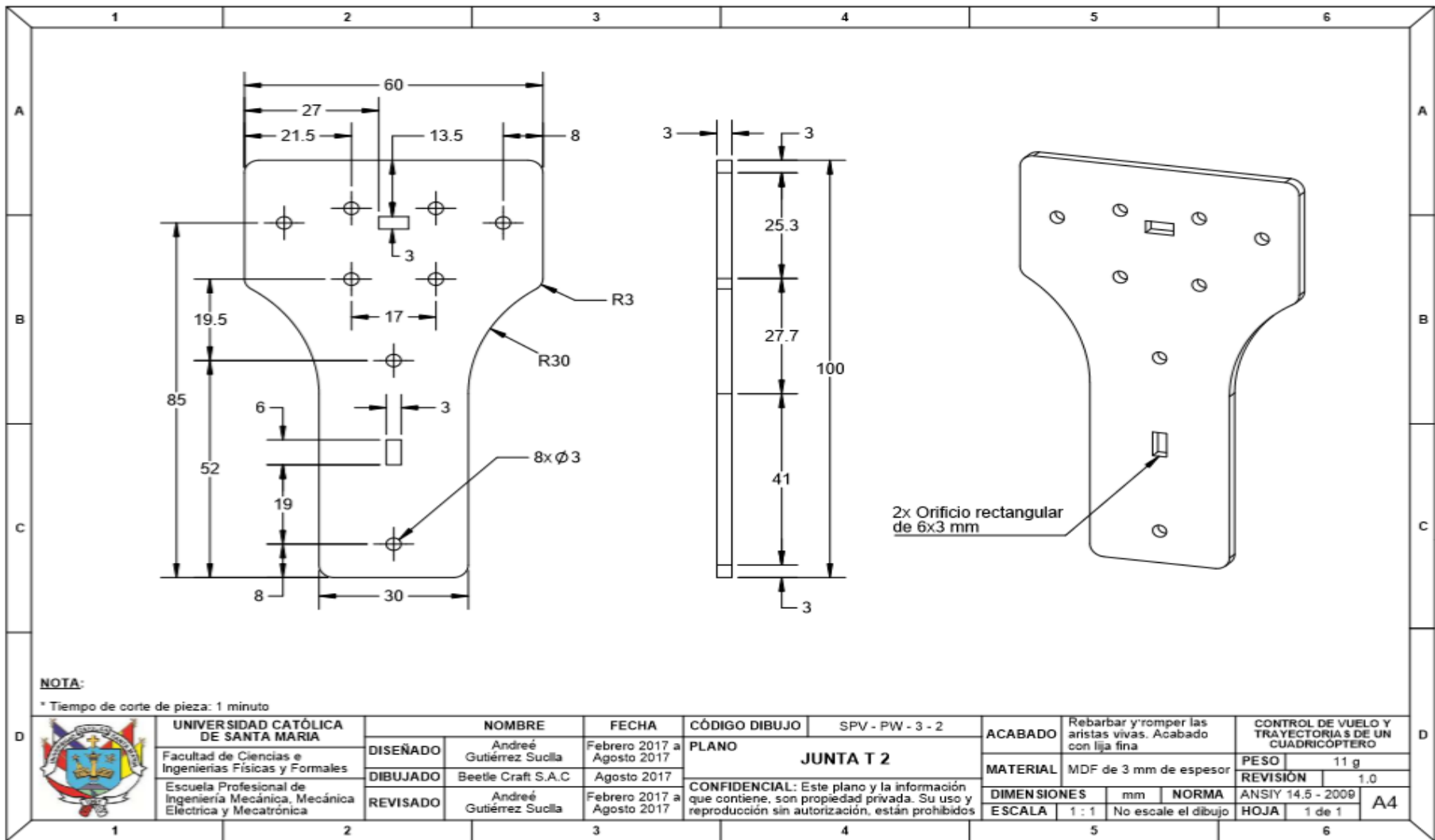
MATERIAL	ACABADO	DIMENSIONES	Escala
Rebarbar y romper las aristas vivas. Acabado con lija fina	MDF de 3 mm de espesor / A307	mm	2 : 3
		NORMA	No escale el dibujo
		ANSI Y 14.5 - 2009	

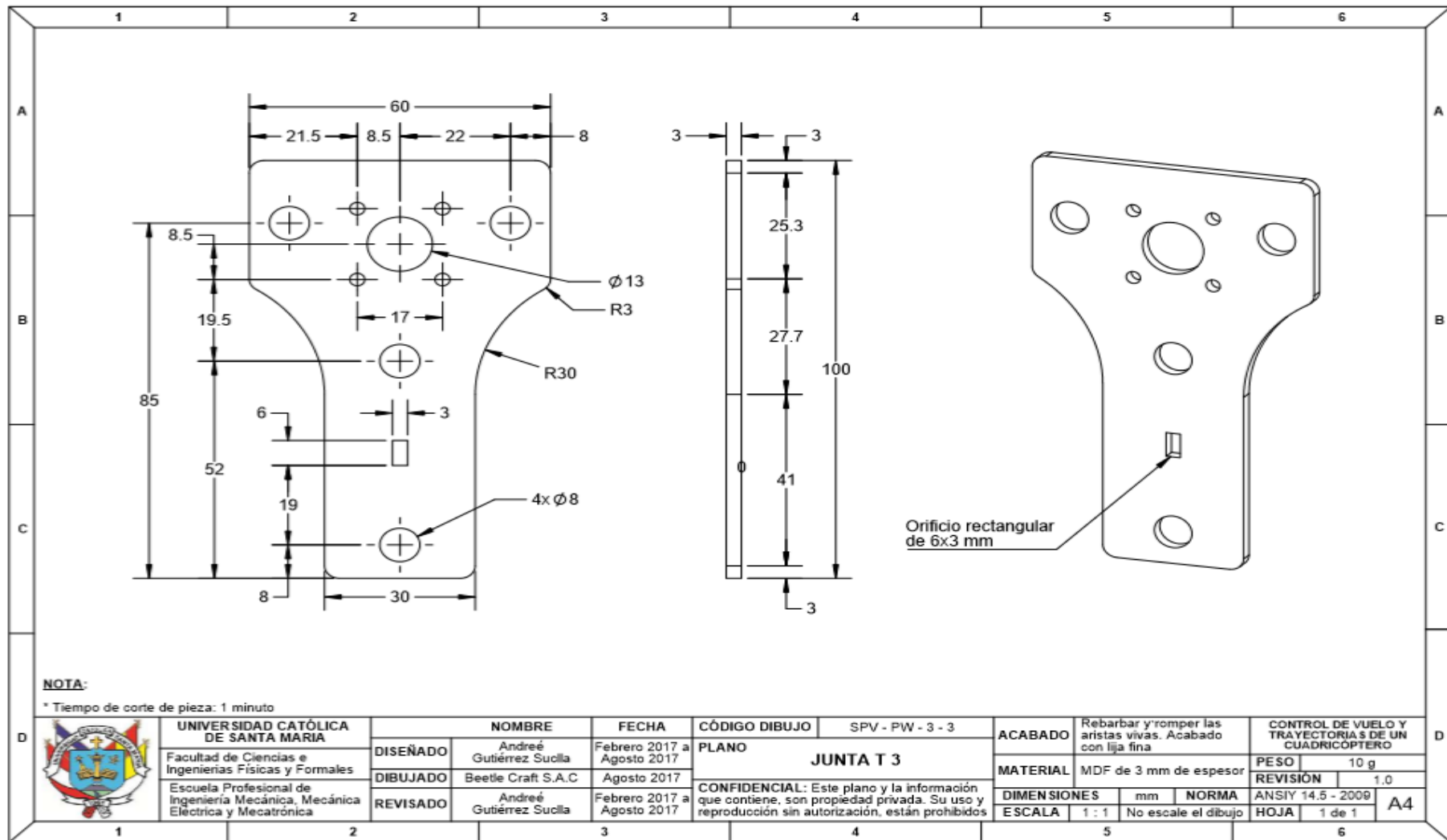
CONTROL DE VUELO Y TRAYECTORIA 3 DE UN CUADRICÓPTERO	
PESO	152 g
REVISIÓN	1.0
HOJA	1 de 1
	A3

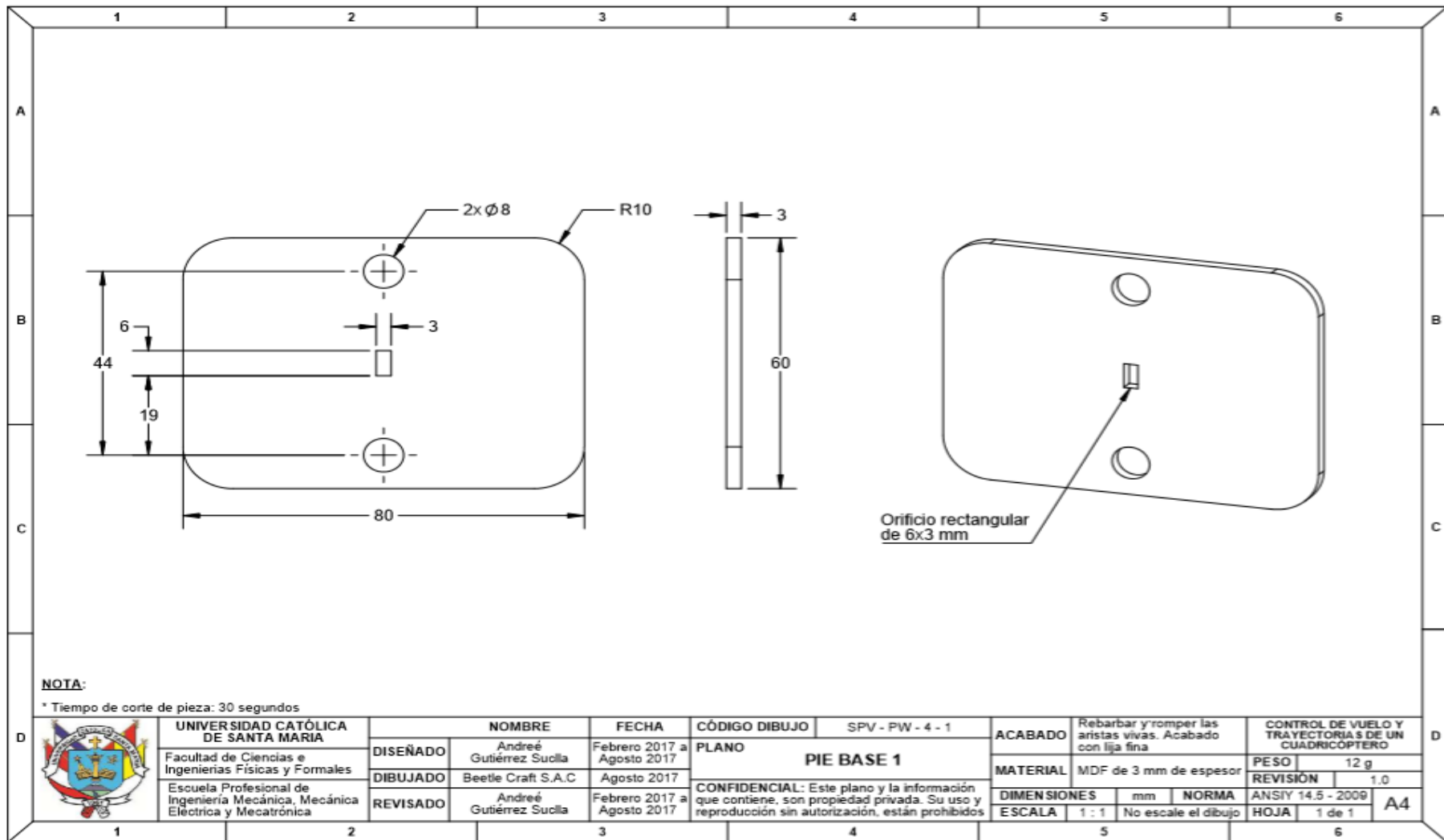


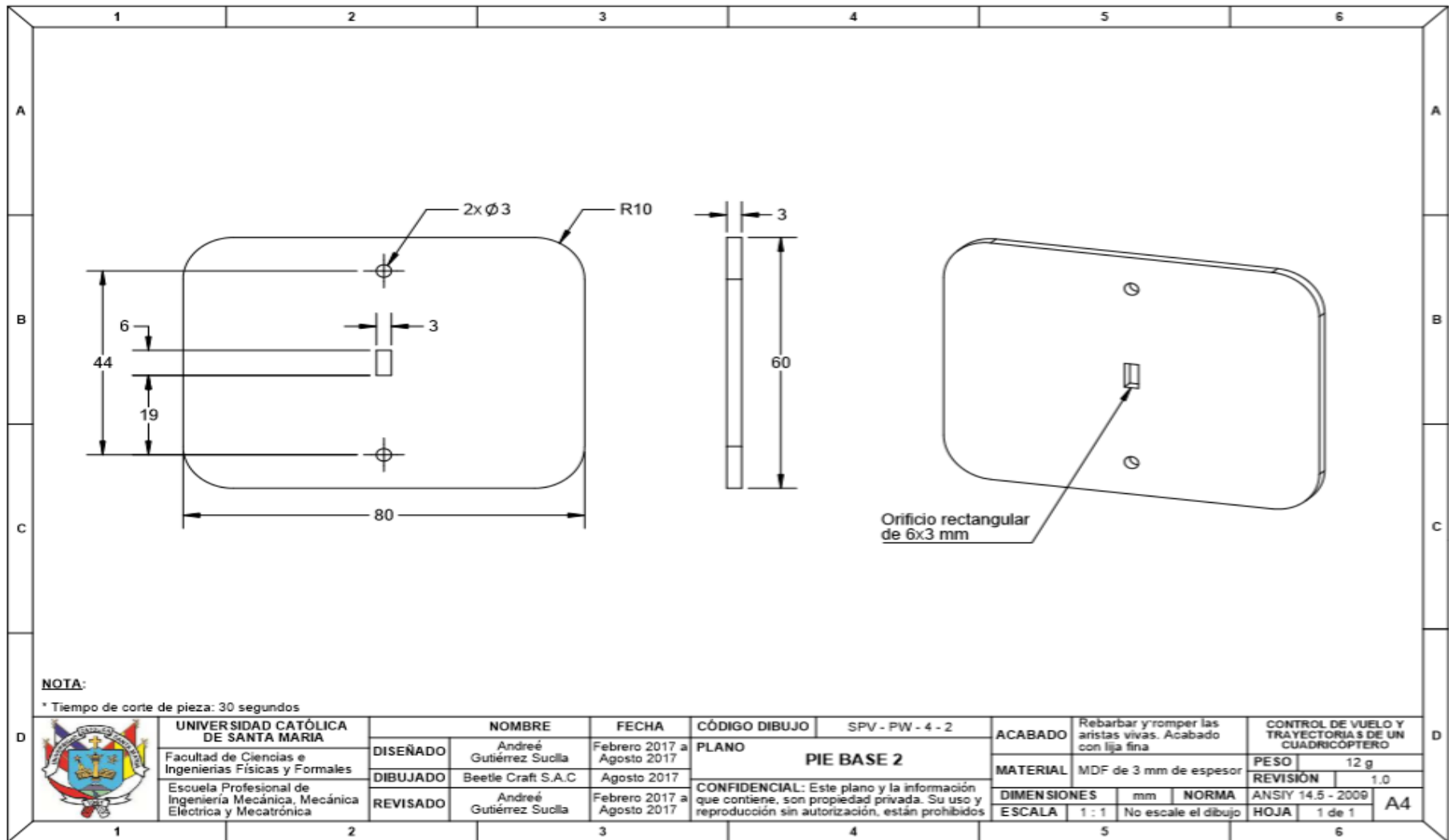


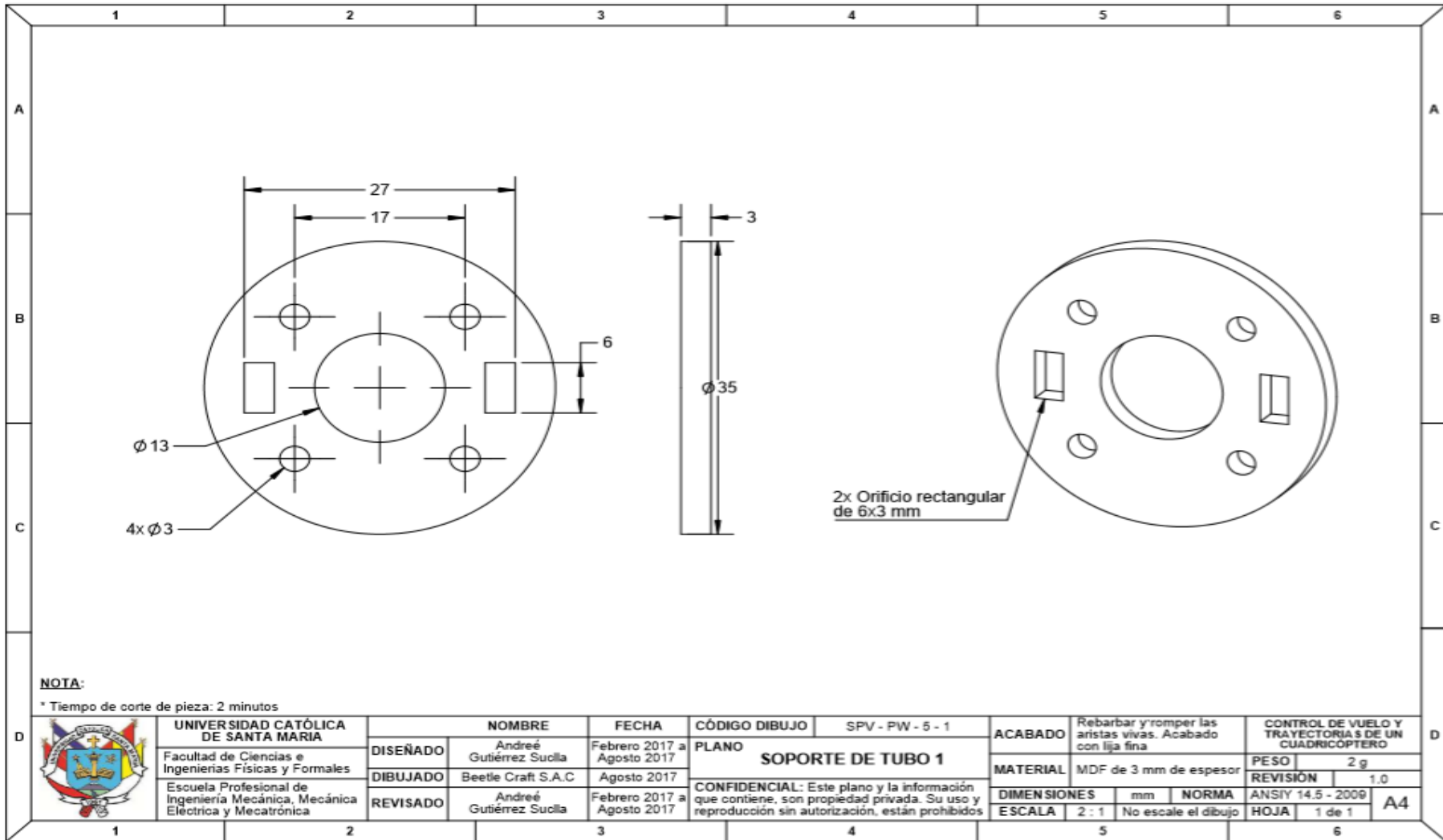


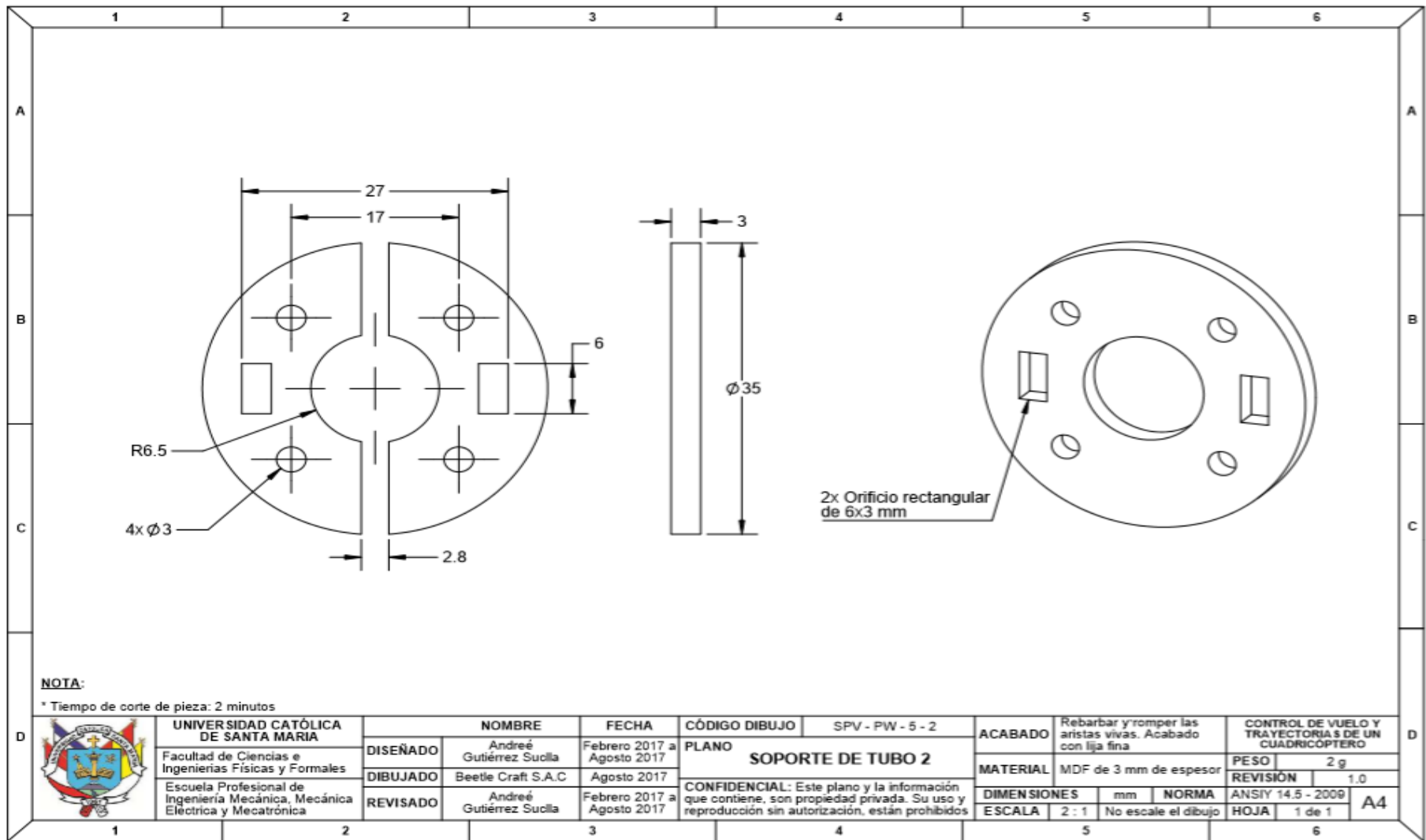


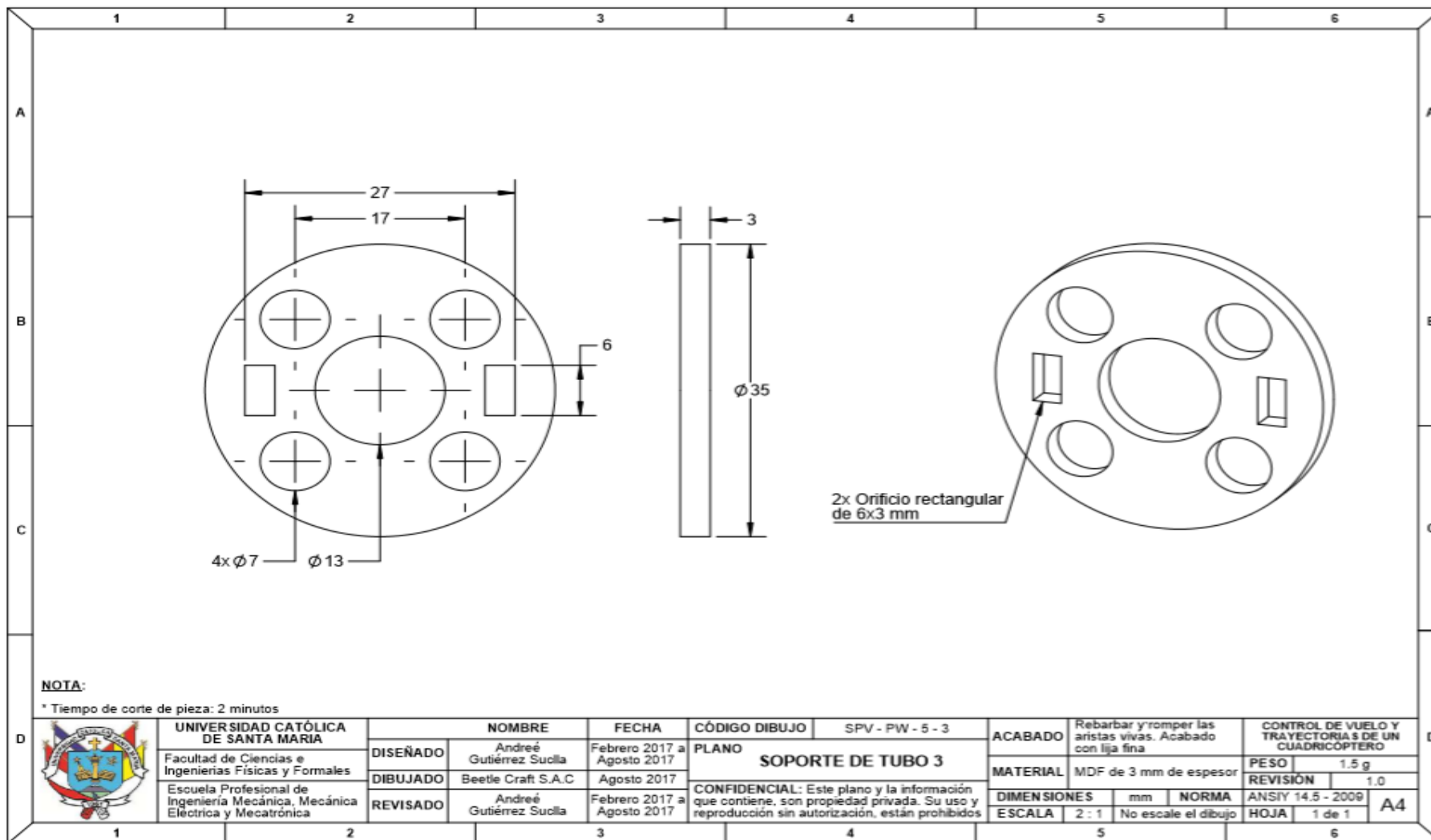


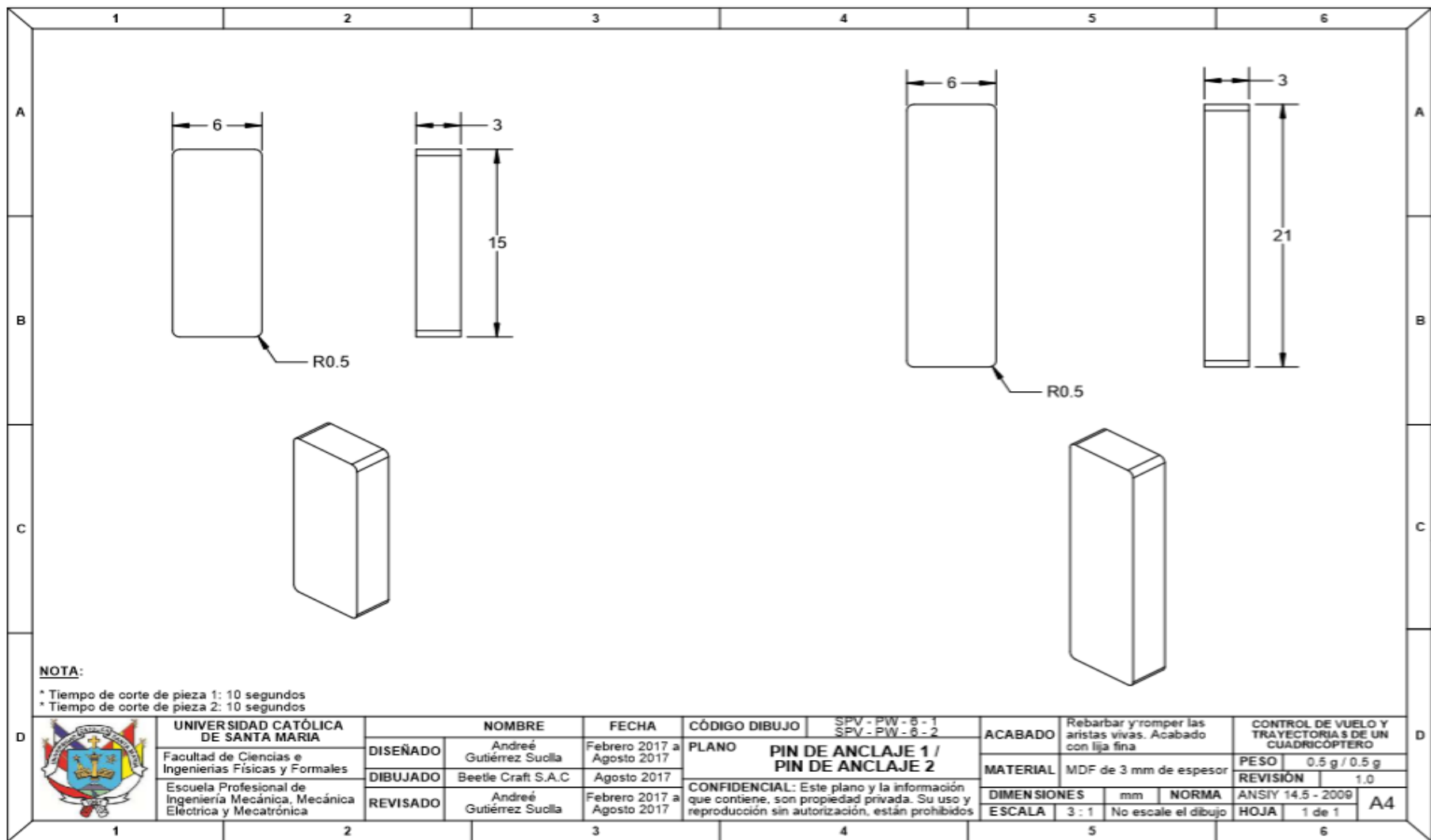


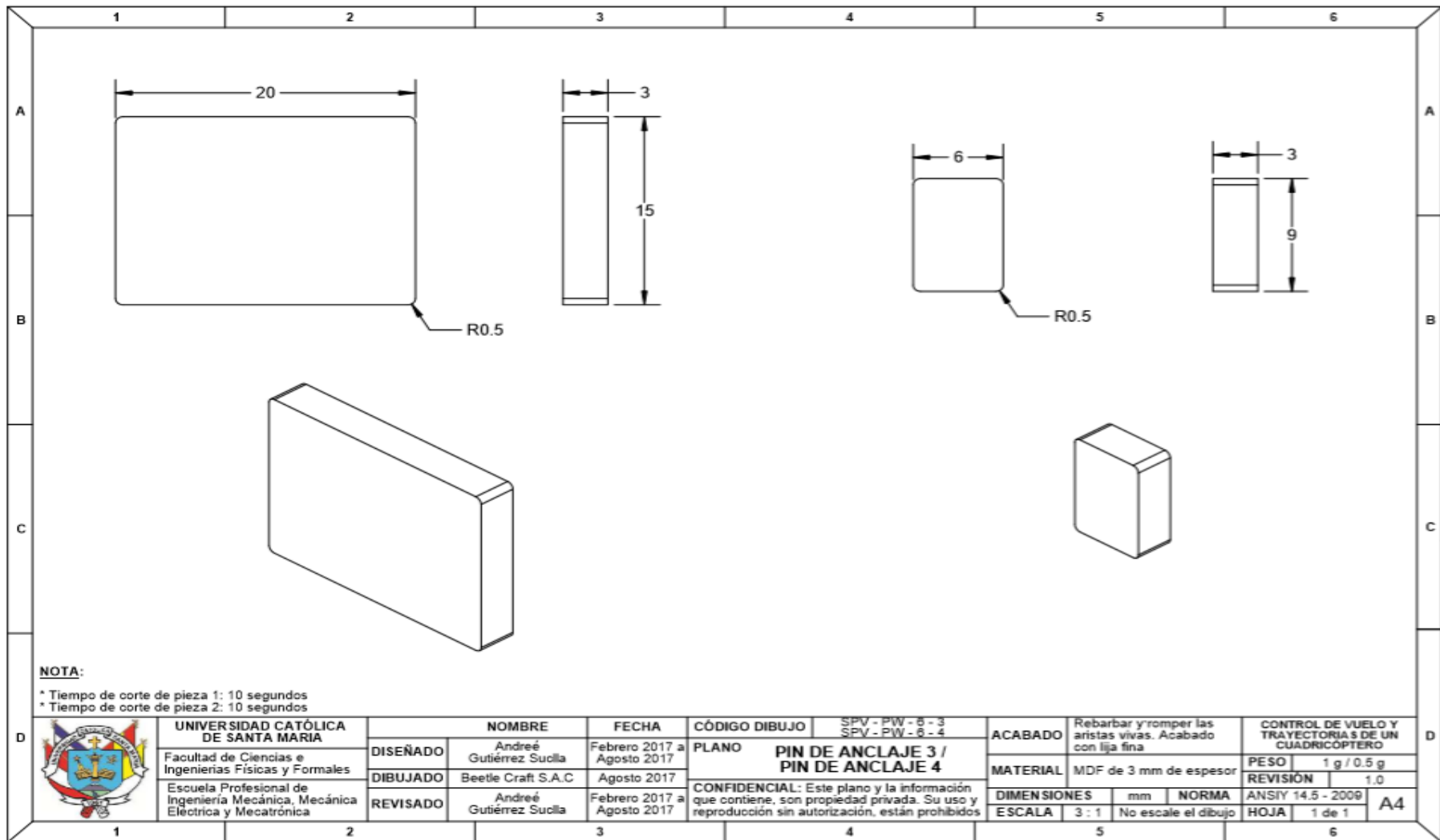


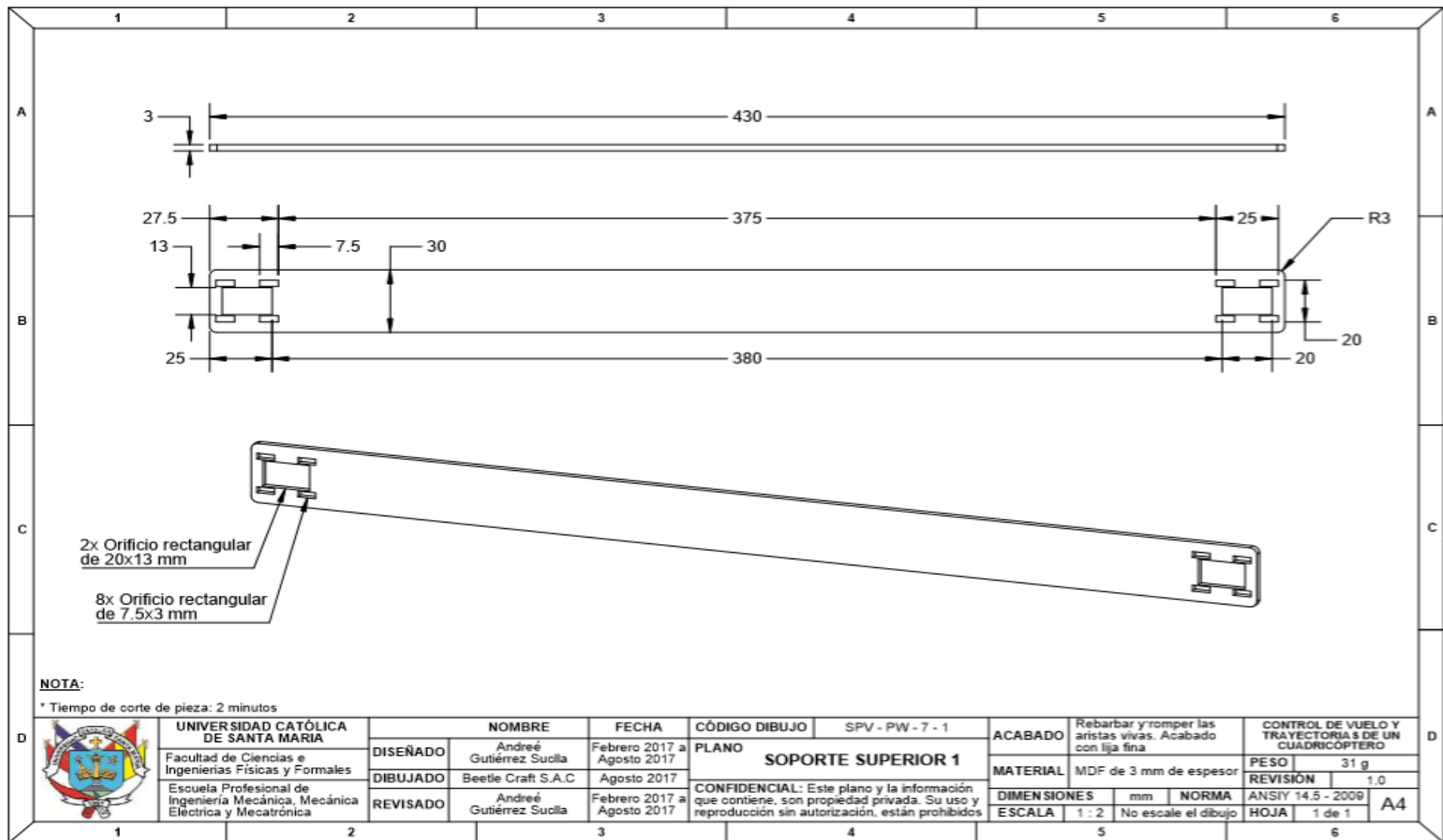


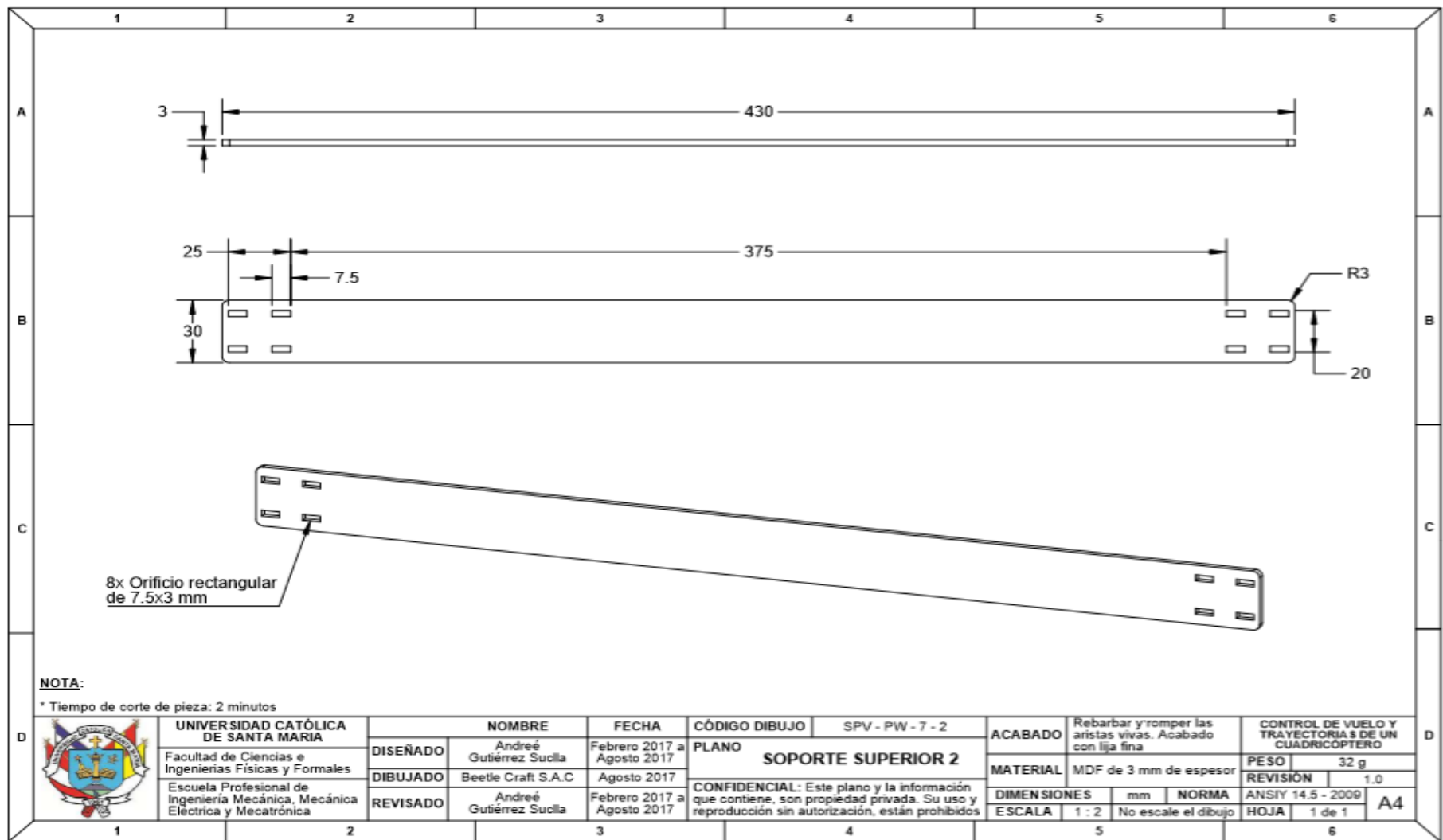


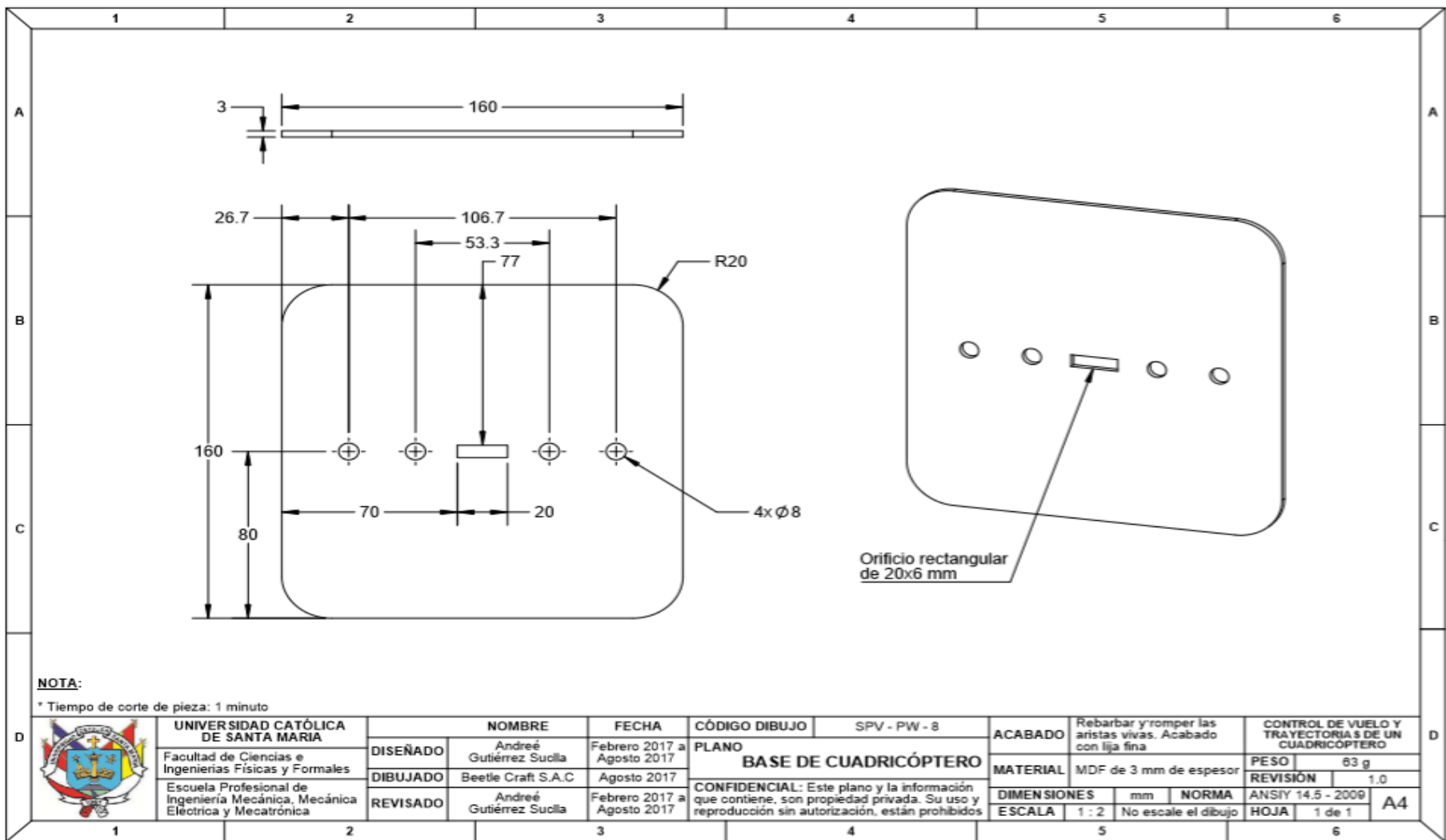


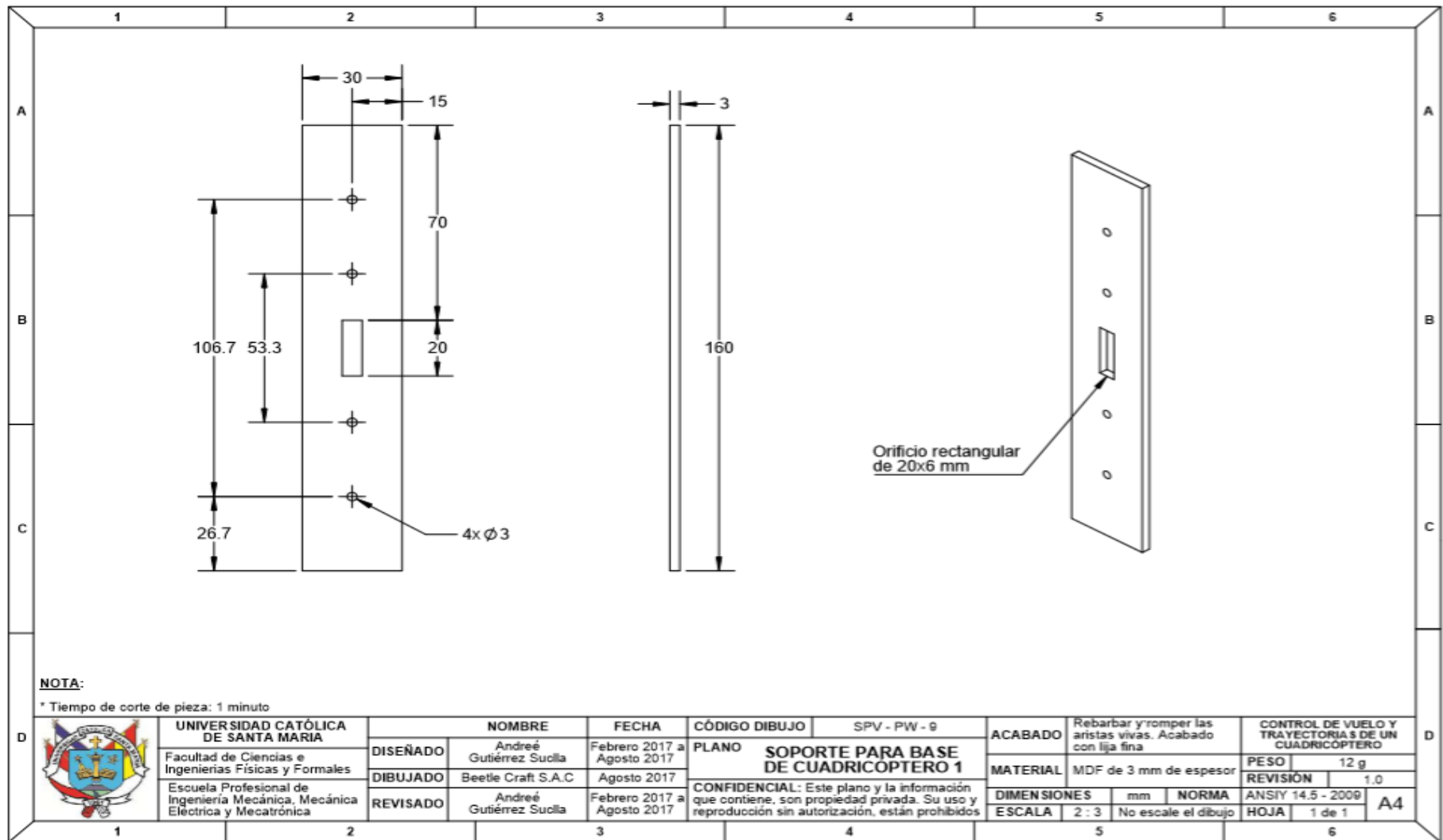


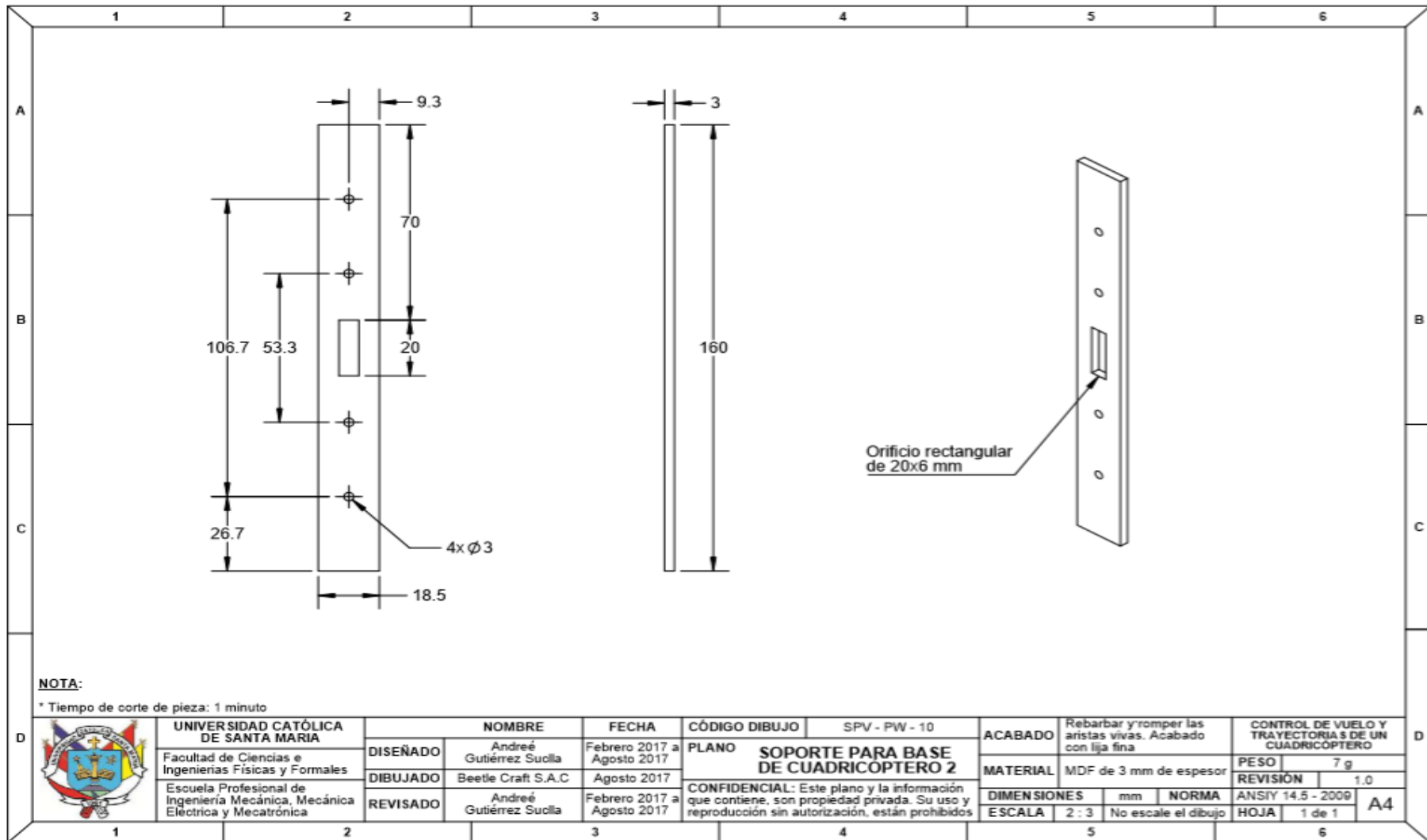








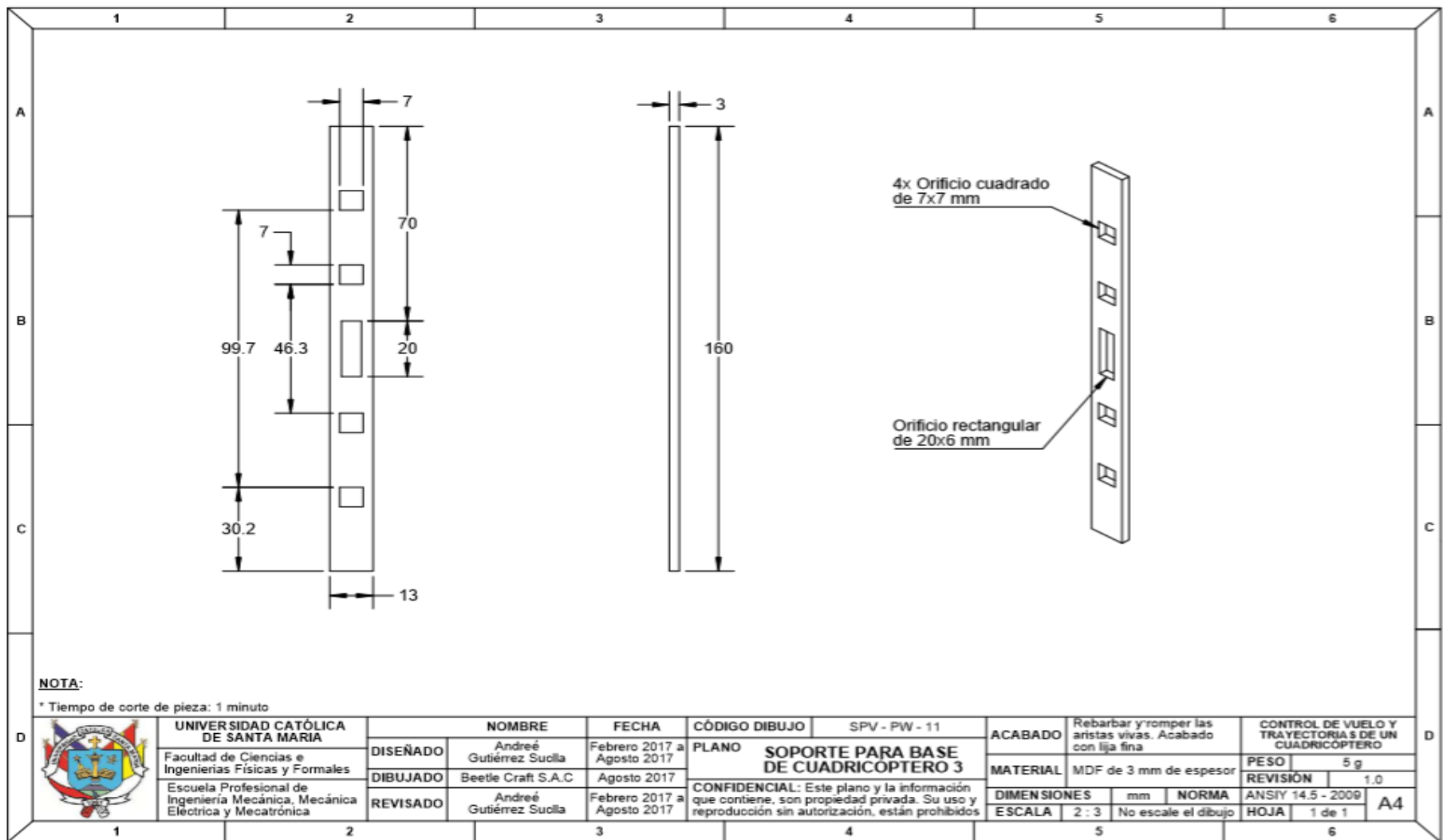




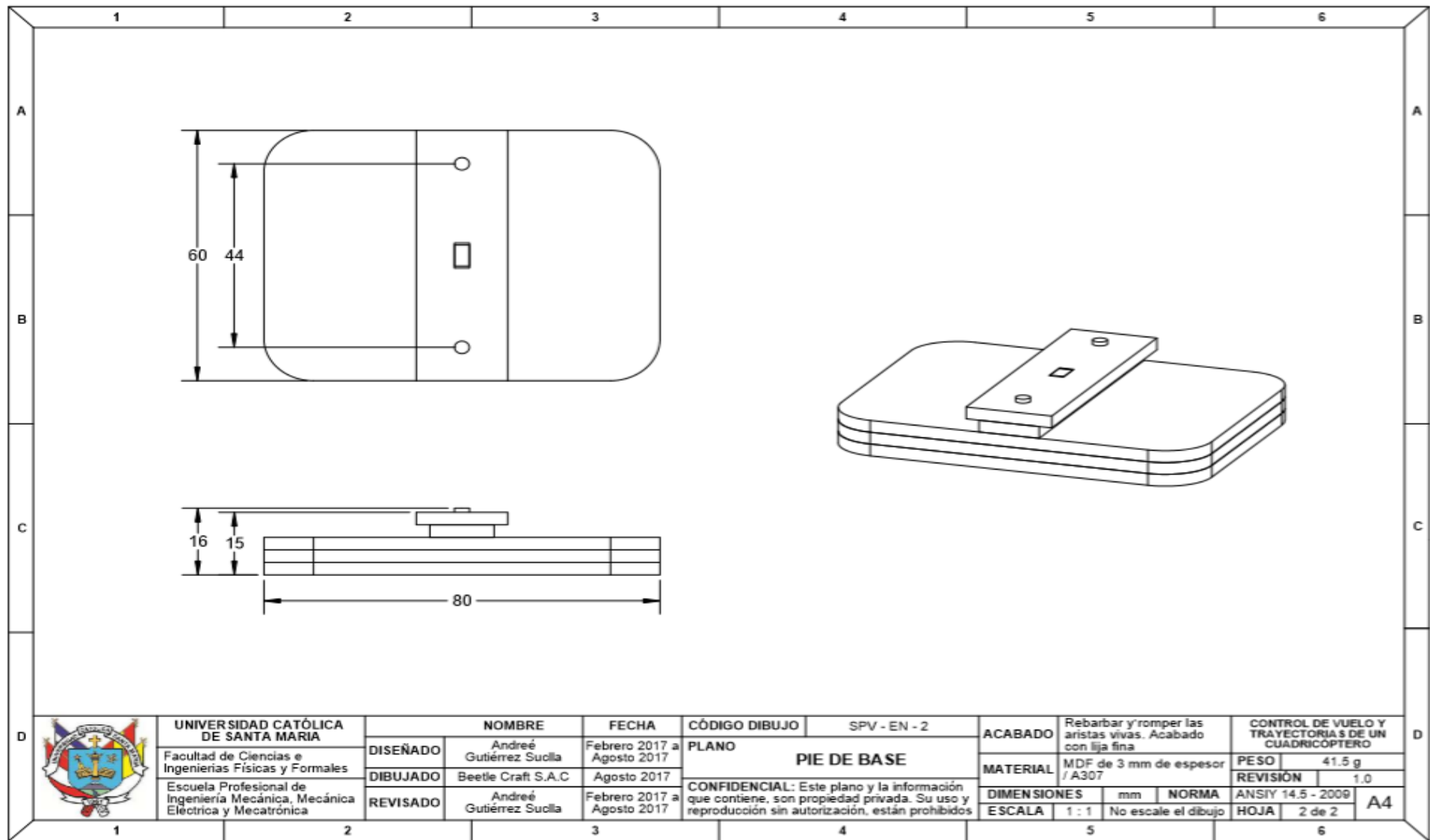
NOTA:

* Tiempo de corte de pieza: 1 minuto

	UNIVERSIDAD CATÓLICA DE SANTA MARÍA	NOMBRE		FECHA	CÓDIGO DIBUJO	SPV - PW - 10		ACABADO		CONTROL DE VUELO Y TRAYECTORIAS DE UN CUADRICÓPTERO	
	Facultad de Ciencias e Ingenierías Físicas y Formales	DISEÑADO	André Gutiérrez Suella	Febrero 2017 a Agosto 2017	PLANO SOPORTE PARA BASE DE CUADRICÓPTERO 2		MATERIAL		Rebarbar y romper las aristas vivas. Acabado con lija fina		PESO
	Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica	DIBUJADO	Beetle Craft S.A.C	Agosto 2017	CONFIDENCIAL: Este plano y la información que contiene, son propiedad privada. Su uso y reproducción sin autorización, están prohibidos		MDF de 3 mm de espesor		NORMA		7 g
		REVISADO	André Gutiérrez Suella	Febrero 2017 a Agosto 2017			2 : 3		ANSI Y 14.5 - 2009		REVISIÓN
						ESCALA		HOJA		A4	



1		2		3		4		5		6			
NRO	CANT	DESCRIPCIÓN	CÓDIGO DE PLANO	PESO	MATERIAL								
1	1	Pin de anclaje 1	SPV - PW - 6 - 1	0.5 g	MDF 3 mm								
2	1	Base de apoyo	SPV - EN - 5	5 g	MDF 3mm / A307								
3	1	Pie base 2	SPV - PW - 4 - 2	12 g	MDF 3 mm								
4	2	Pie base 1	SPV - PW - 4 - 1	24 g	MDF 3 mm								
5	2	M3x10 TORNF CABCIL PHILLIPS	_____	_____	A307								
<p>Tornillo de ajuste de pie de base con perfil estructural de aluminio</p> <p>Pin de encuadre de placas durante ensamble</p> <p>Unión pegada entre caras con cola de carpintero</p> <p>Unión pegada entre caras con cola de carpintero</p> <p>Unión pegada entre caras con cola de carpintero</p>													
		UNIVERSIDAD CATÓLICA DE SANTA MARÍA Facultad de Ciencias e Ingenierías Físicas y Formales Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica		NOMBRE Andréé Gutiérrez Suella		FECHA Febrero 2017 a Agosto 2017		CÓDIGO DIBUJO SPV - EN - 2		ACABADO Rebarbar y romper las aristas vivas. Acabado con lija fina		CONTROL DE VUELO Y TRAYECTORIAS DE UN CUADRICÓPTERO	
		DISEÑADO Andréé Gutiérrez Suella		FECHA Febrero 2017 a Agosto 2017		PLANO PIE DE BASE		MATERIAL MDF de 3 mm de espesor / A307		PESO 41.5 g		REVISIÓN 1.0	
		DIBUJADO Beetle Craft S.A.C		FECHA Agosto 2017		CONFIDENCIAL: Este plano y la información que contiene, son propiedad privada. Su uso y reproducción sin autorización, están prohibidos		DIMENSIONES mm		NORMA ANSII 14.5 - 2009		HOJA 1 de 2	
		REVISADO Andréé Gutiérrez Suella		FECHA Febrero 2017 a Agosto 2017				ESCALA 1 : 1		No escale el dibujo		A4	
1		2		3		4		5		6			



UNIVERSIDAD CATÓLICA DE SANTA MARÍA
 Facultad de Ciencias e Ingenierías Físicas y Formales
 Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica

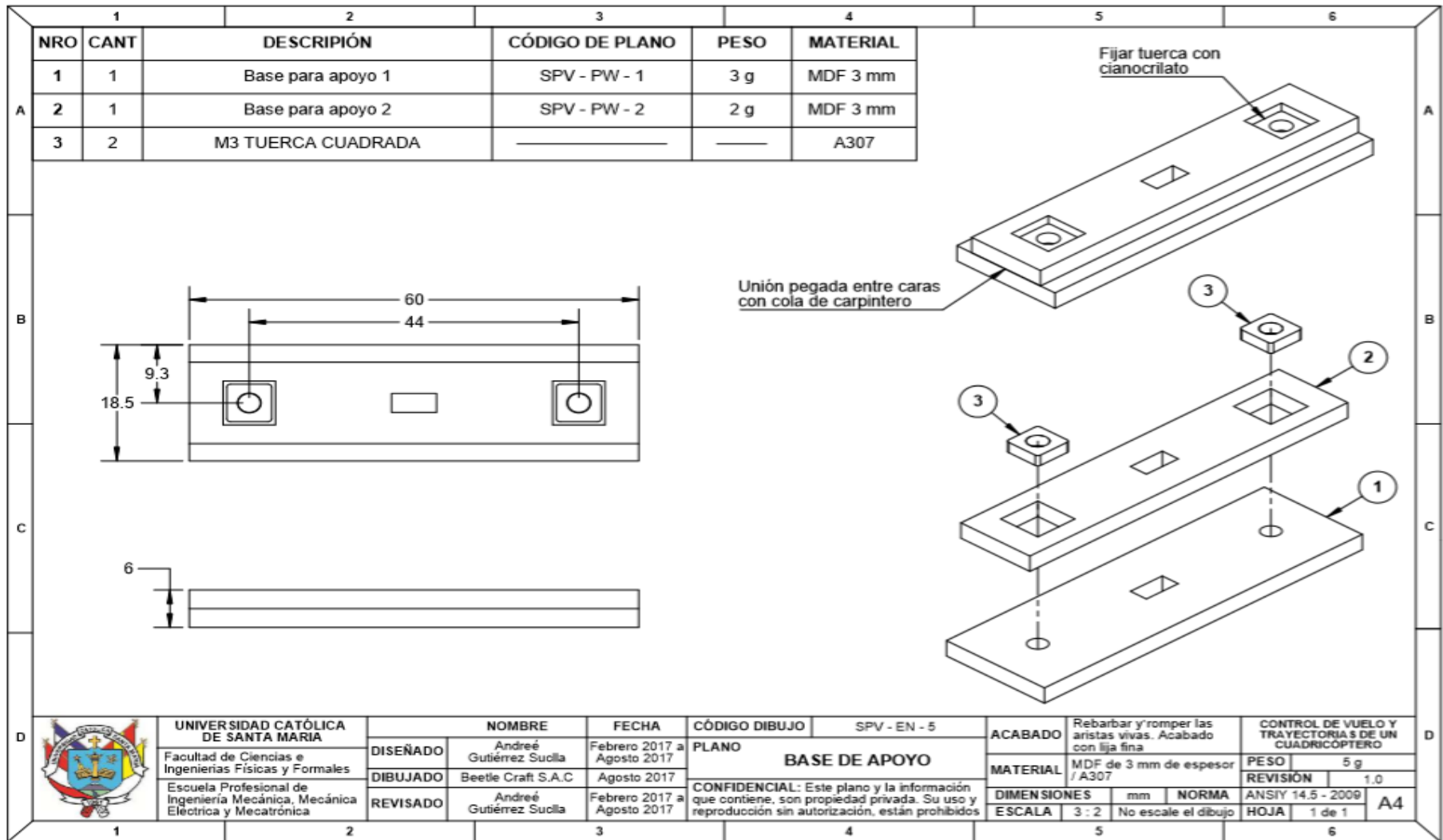
	NOMBRE	FECHA
DISEÑADO	André Gutiérrez Sualla	Febrero 2017 a Agosto 2017
DIBUJADO	Beetle Craft S.A.C	Agosto 2017
REVISADO	André Gutiérrez Sualla	Febrero 2017 a Agosto 2017

CÓDIGO DIBUJO: SPV - EN - 2
PLANO
PIE DE BASE
 CONFIDENCIAL: Este plano y la información que contiene, son propiedad privada. Su uso y reproducción sin autorización, están prohibidos

ACABADO	Rebarbar y romper las aristas vivas. Acabado con lija fina		
MATERIAL	MDF de 3 mm de espesor / A307		
DIMENSIONES	mm	NORMA	ANSI Y 14.5 - 2009
ESCALA	1 : 1	No escale el dibujo	

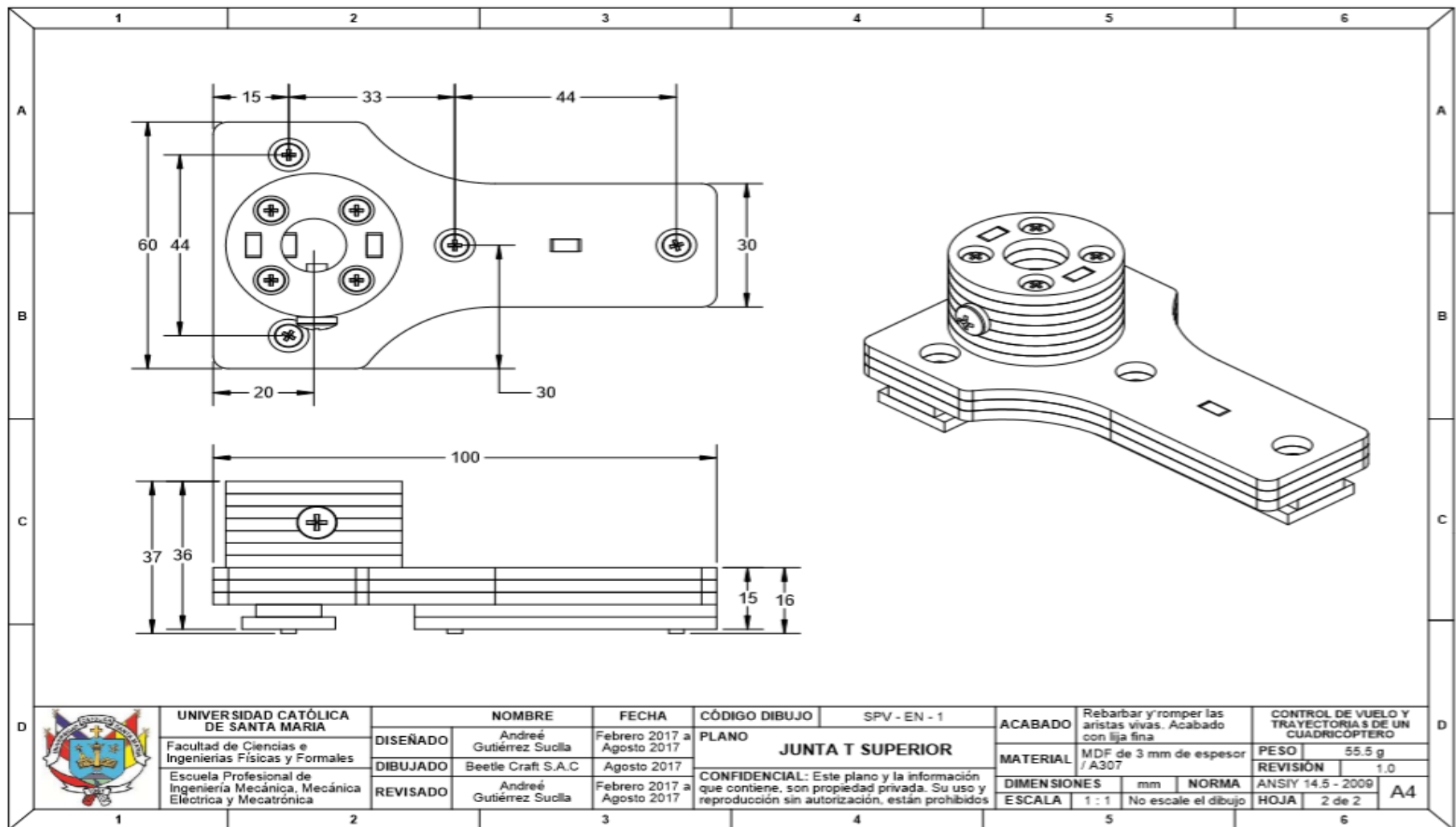
CONTROL DE VUELO Y TRAYECTORIAS DE UN CUADRICÓPTERO	
PESO	41.5 g
REVISIÓN	1.0
HOJA	2 de 2

A4



1	2	3	4	5	6
NRO	CANT	DESCRIPCIÓN	CÓDIGO DE PLANO	PESO	MATERIAL
1	1	Junta T inferior	SPV - EN - 1 - 1	41 g	MDF 3 mm / A307
2	1	Soporte para tubo	SPV - EN - 1 - 2	14.5 g	MDF 3 mm / A307
3	2	M3x25 TORNIF CABCIL PHILLIPS	_____	_____	A307

	UNIVERSIDAD CATÓLICA DE SANTA MARÍA	NOMBRE	FECHA	CÓDIGO DIBUJO	SPV - EN - 1	ACABADO	Rebarbar y romper las aristas vivas. Acabado con lija fina		CONTROL DE VUELO Y TRAYECTORIAS DE UN CUADRICÓPTERO	
	Facultad de Ciencias e Ingenierías Físicas y Formales	DISEÑADO	Andréé Gutiérrez Suclla	Febrero 2017 a Agosto 2017	PLANO	JUNTA T SUPERIOR	MATERIAL	MDF de 3 mm de espesor / A307	PESO	55.5 g
	Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica	DIBUJADO	Beetle Craft S.A.C	Agosto 2017				REVISIÓN	1.0	
		REVISADO	Andréé Gutiérrez Suclla	Febrero 2017 a Agosto 2017	CONFIDENCIAL: Este plano y la información que contiene, son propiedad privada. Su uso y reproducción sin autorización, están prohibidos		DIMENSIONES	mm	NORMA	ANSI Y 14.5 - 2009
						ESCALA	1 : 1	No escale el dibujo	HOJA	1 de 2

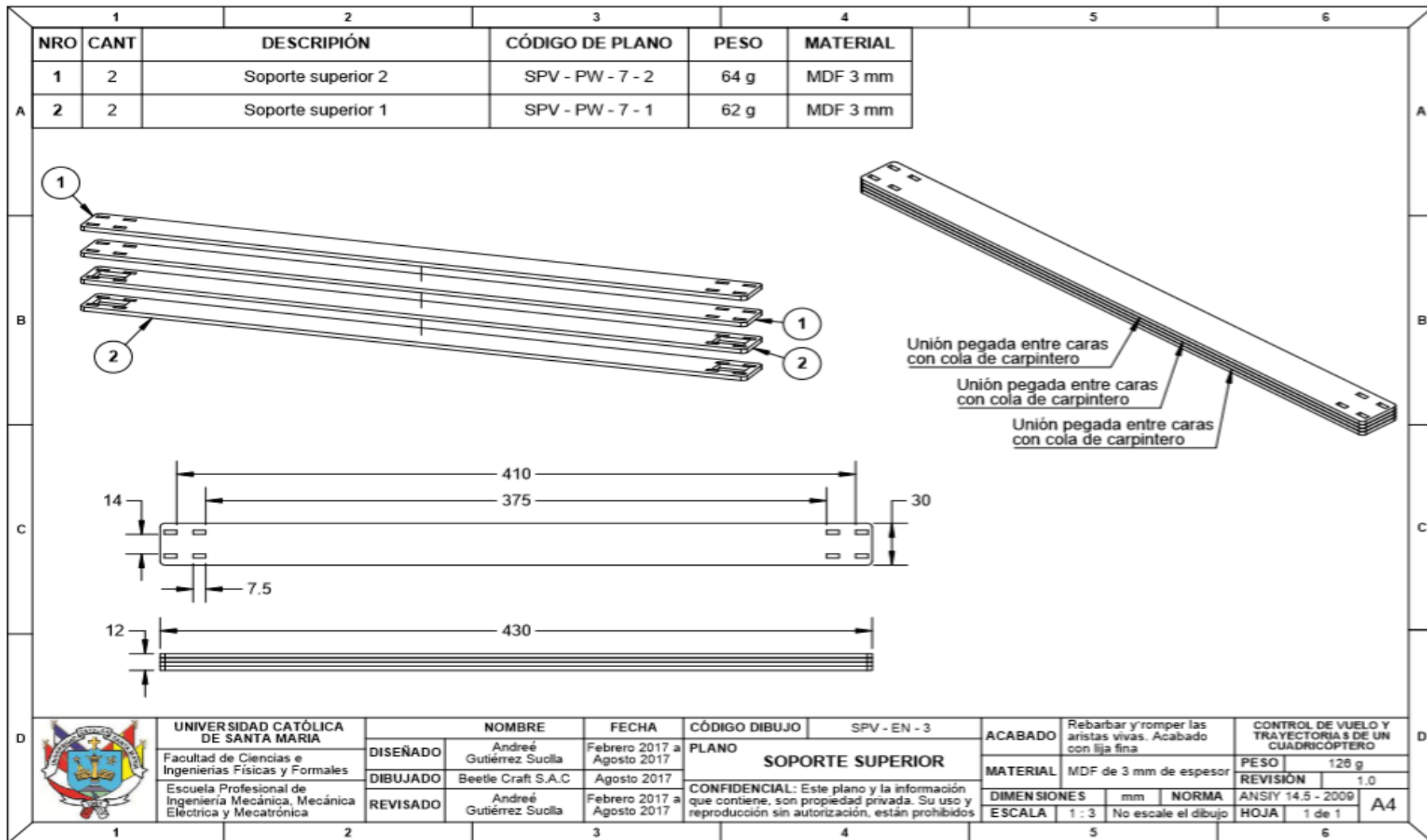


UNIVERSIDAD CATÓLICA DE SANTA MARÍA
 Facultad de Ciencias e Ingenierías Físicas y Formales
 Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica

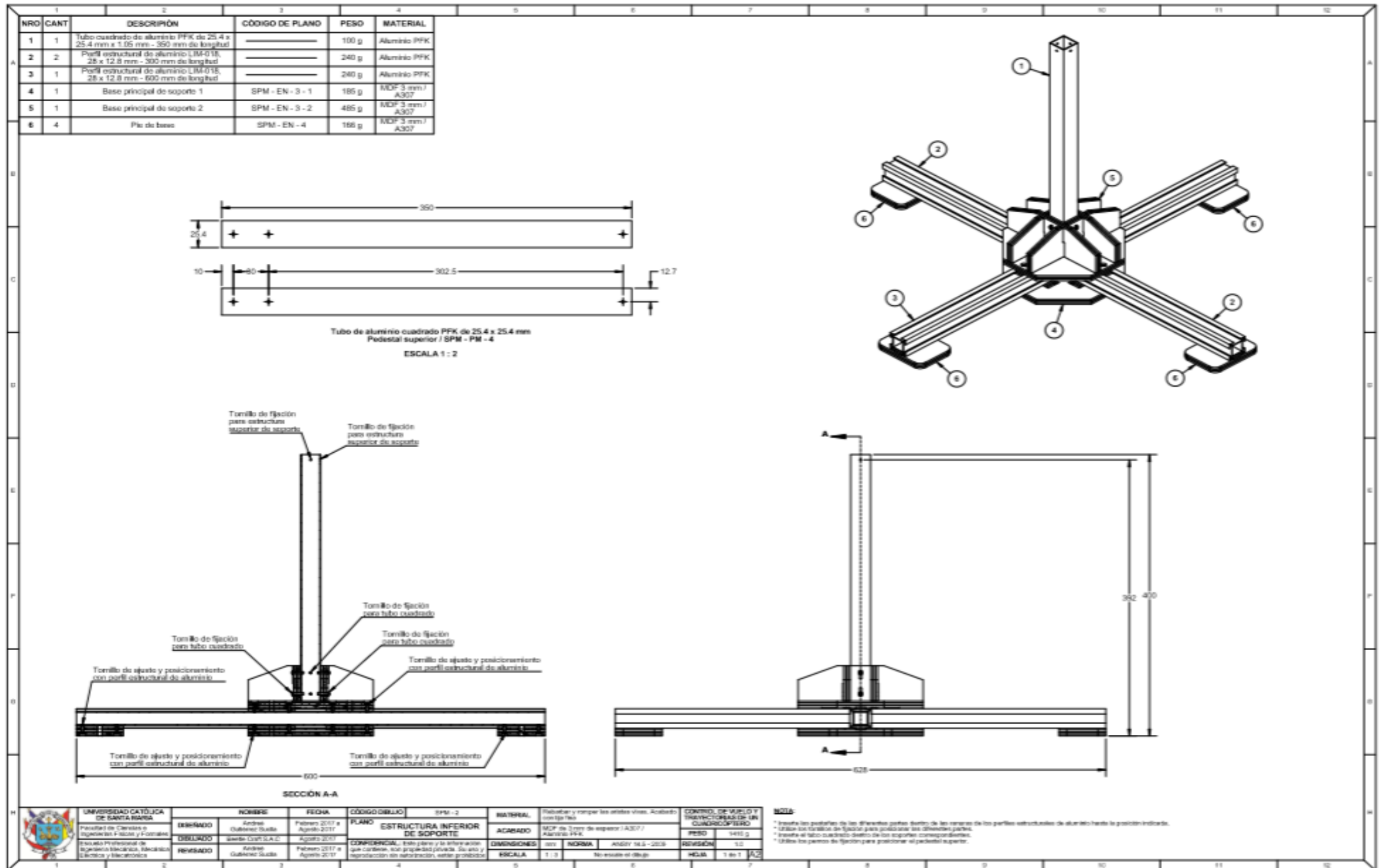
	NOMBRE	FECHA	CÓDIGO DIBUJO	SPV - EN - 1	
DISEÑADO	André Gutiérrez Suella	Febrero 2017 a Agosto 2017	PLANO	JUNTA T SUPERIOR	
DIBUJADO	Beetle Craft S.A.C	Agosto 2017	CONFIDENCIAL: Este plano y la información que contiene, son propiedad privada. Su uso y reproducción sin autorización, están prohibidos		
REVISADO	André Gutiérrez Suella	Febrero 2017 a Agosto 2017			

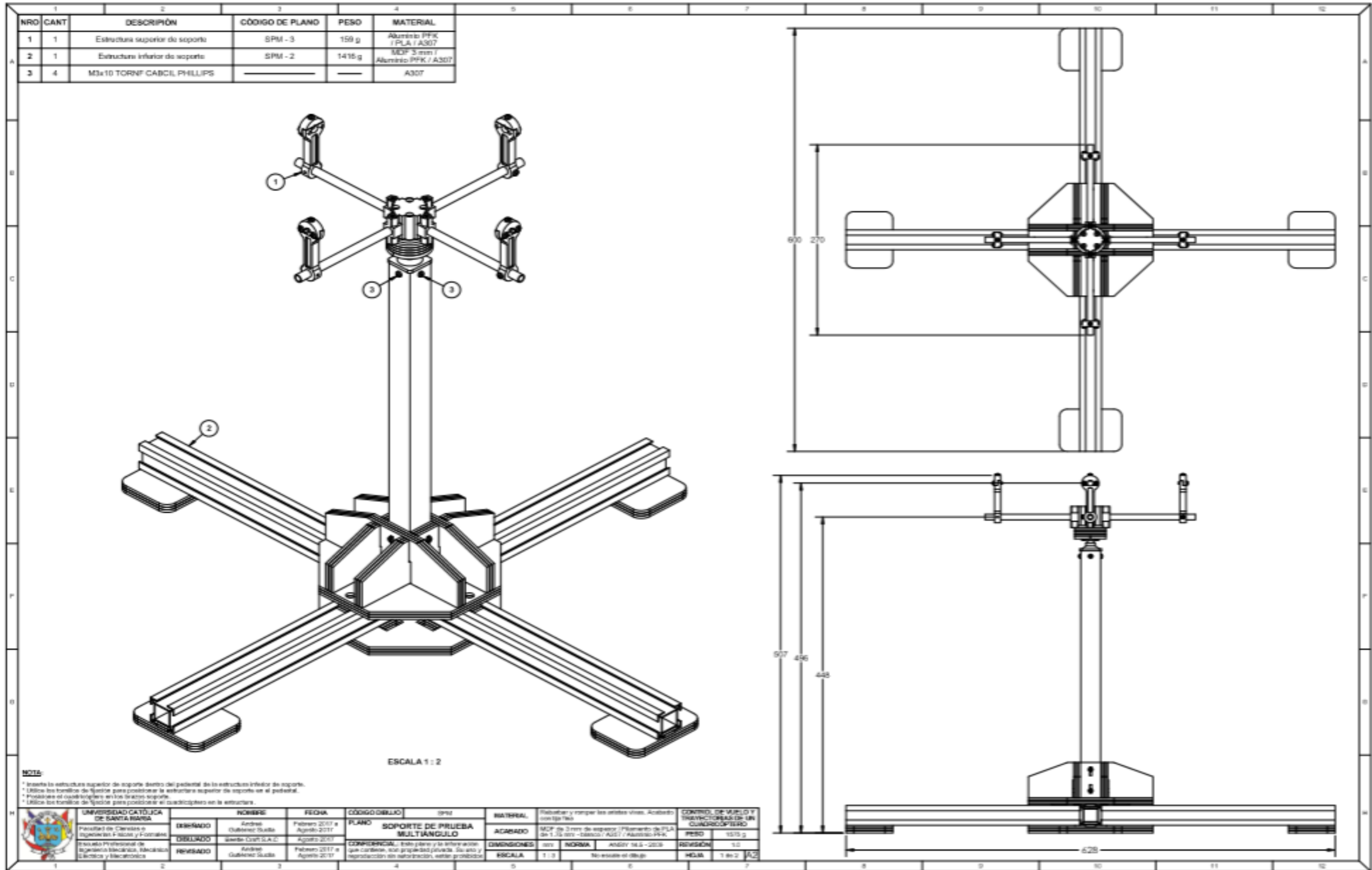
ACABADO	MATERIAL	DIMENSIONES	ESCALA	NORMA
Rebarbar y romper las aristas vivas. Acabado con lija fina	MDF de 3 mm de espesor / A307	mm	1 : 1	No escale el dibujo

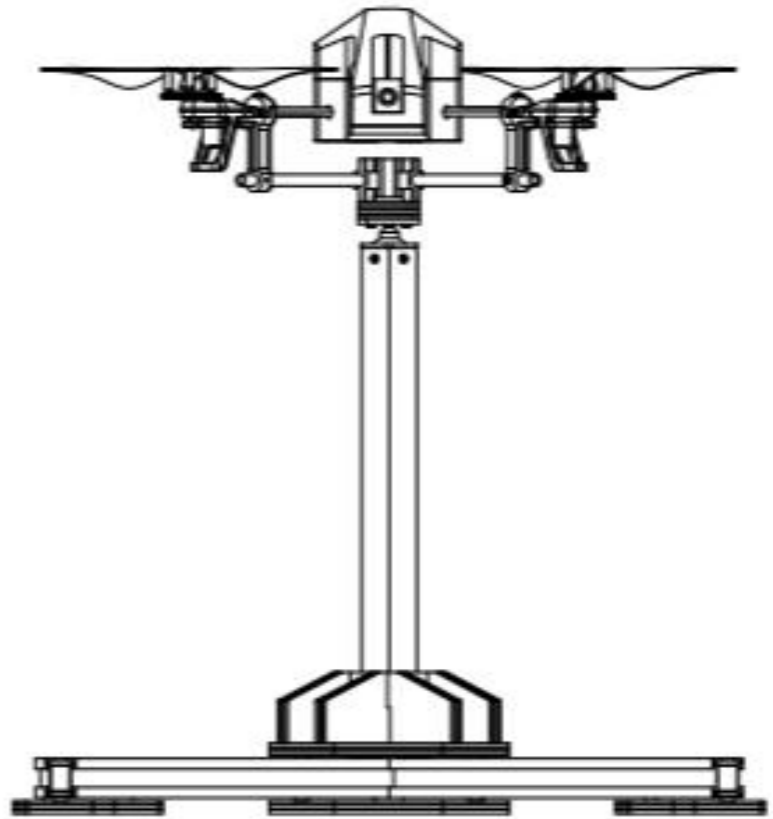
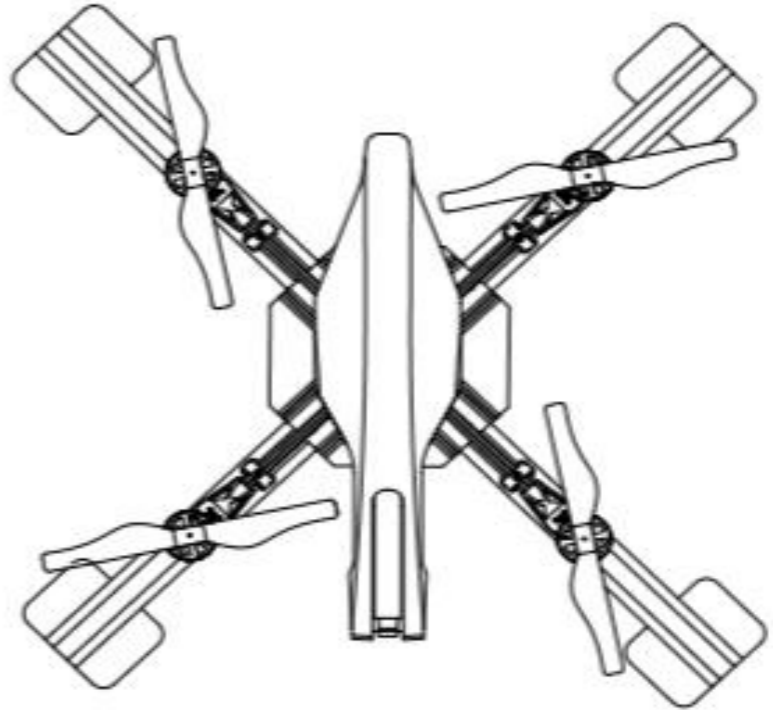
CONTROL DE VUELO Y TRAYECTORIAS DE UN CUADRICÓPTERO	
PESO	55.5 g
REVISIÓN	1.0
ANSI Y 14.5 - 2009	
HOJA	2 de 2
A4	



Anexo 03: Modulo de estabilidad (STABILITY)





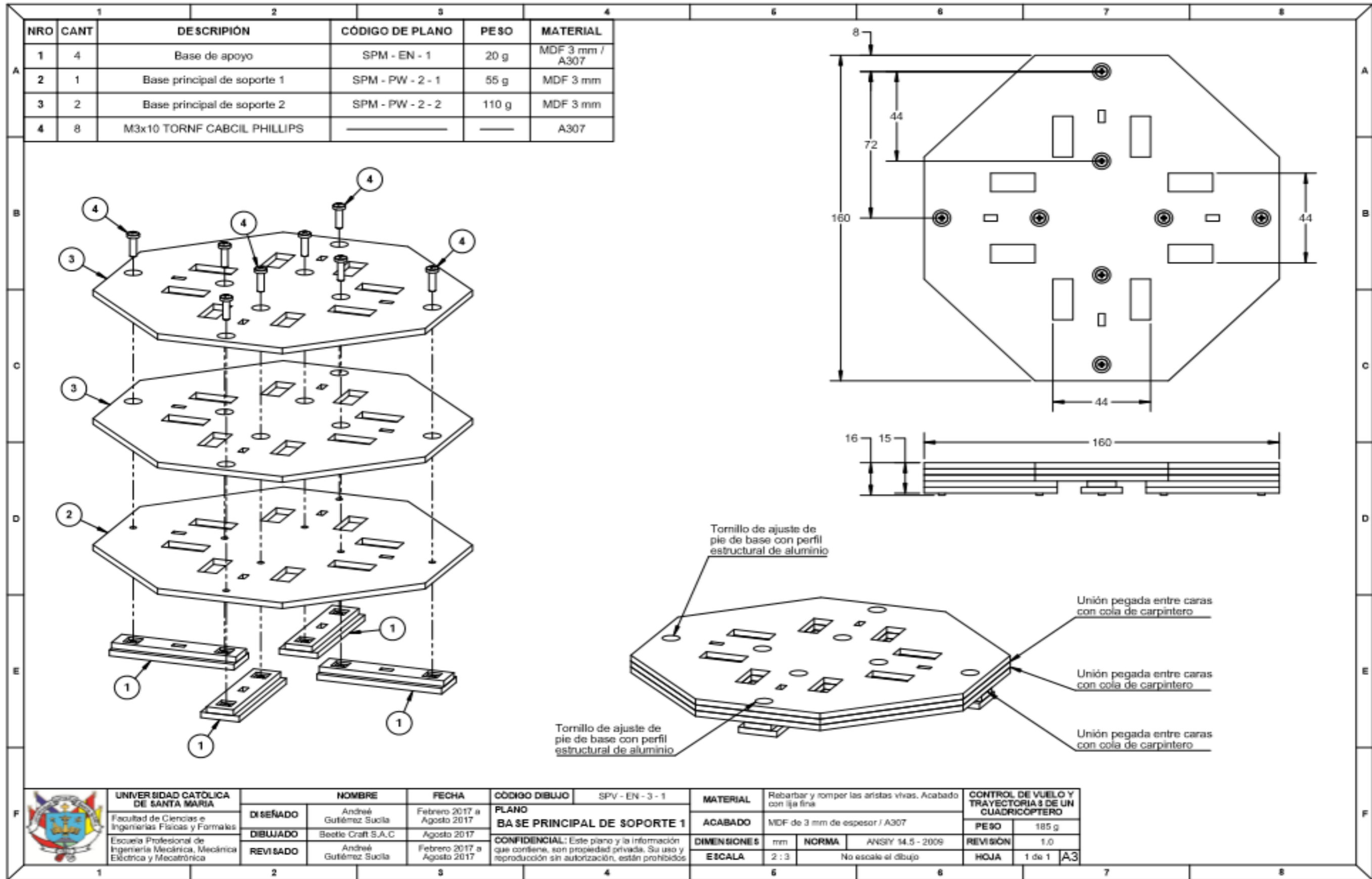


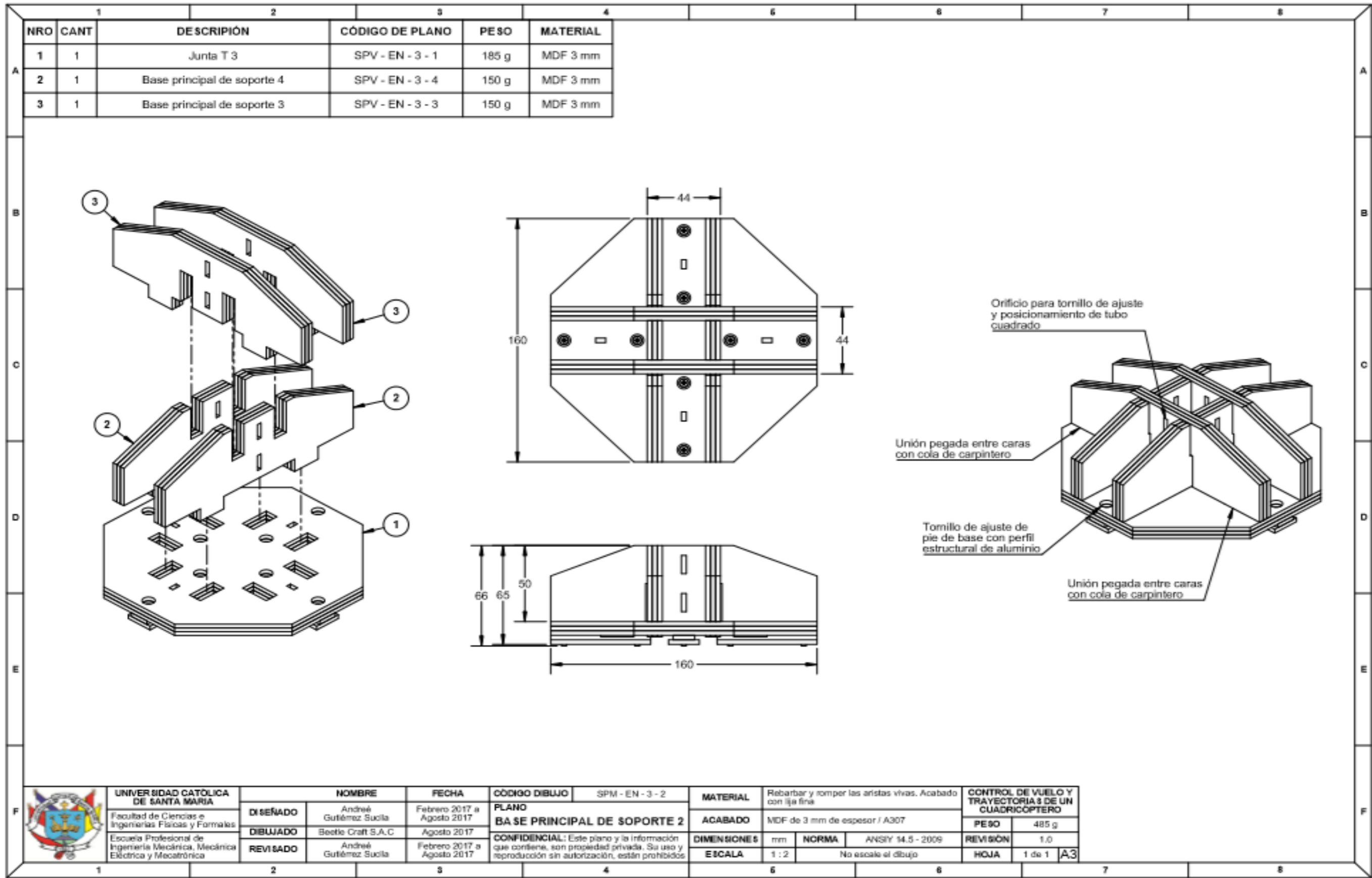
ESCALA 1 : 2

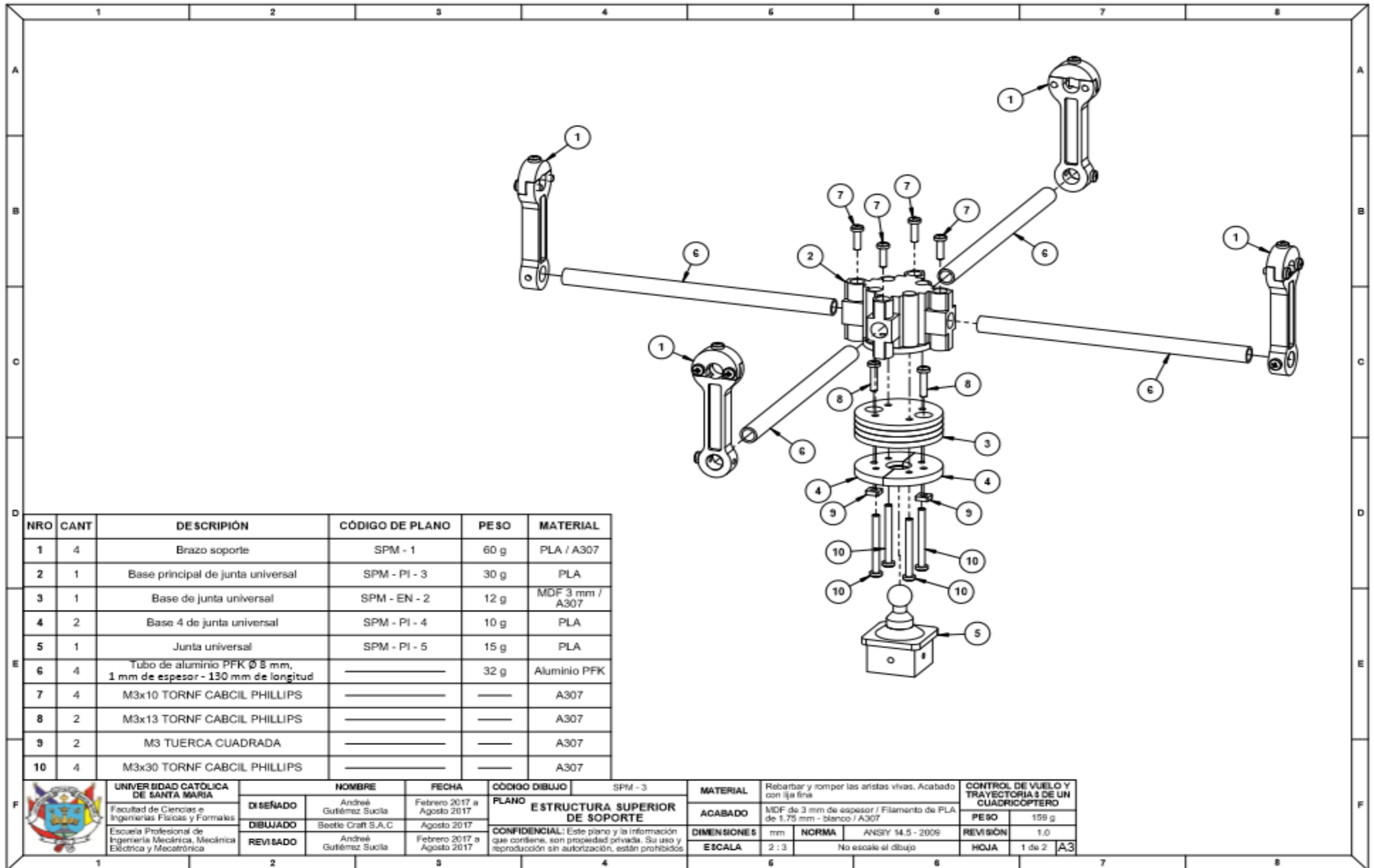
NOTAS

- * Inserte la estructura superior de soporte dentro del pedestal de la estructura inferior de soporte.
- * Coloque los tornillos de fijación para conectar la estructura superior de soporte en el pedestal.
- * Posicione el cuadrúptero en los brazos soporte.
- * Marque los tornillos de fijación para posicionar el cuadrúptero en la estructura.

	UNIVERSIDAD CATÓLICA DE SANTA BARBARA	NOMBRE	FECHA	CÓDIGO DISEÑO	TIPO	MATERIAL	Plástico y metal en partes vivas. Acabado con pintura.	CONTROL DE VUELO Y INSTRUMENTACIÓN DE UN CUADRÓPTERO
	Facultad de Ciencias e Ingeniería F. Física y Química	DISEÑADO	FECHA	PLANO	SOPOORTE DE PRUEBA MULTIANGULO	ACABADO	MCP de 3mm de espesor / Plástico de PLA de 1.75 mm / ABS / Acabado: PVA.	PERO
	Instituto Politécnico de Ingeniería Mecánica, Mecánica Eléctrica y Electrónica	DISEÑADO	FECHA	FECHA	COMPROBACION	Este plano y la lista de materiales que contiene, son propiedad privada. Su uso y reproducción sin autorización, están prohibidos.	DIMENSIONES	REVISION
		REVISADO	FECHA	FECHA	ESCALA	1 : 2	NORMA	ANSI 18.2 - 2018

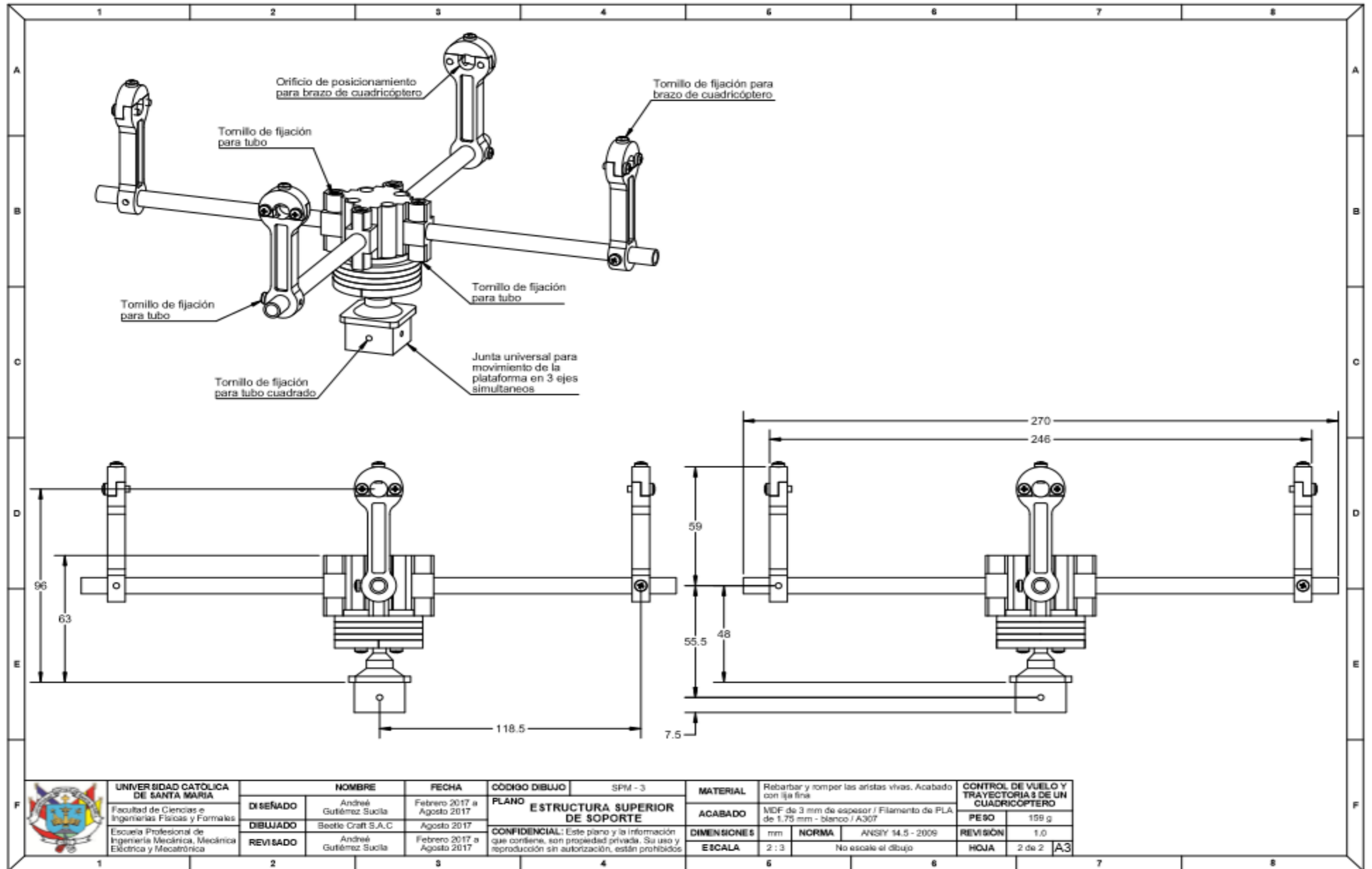






NRO	CANT	DESCRIPCIÓN	CÓDIGO DE PLANO	PE SO	MATERIAL
1	4	Brazo soporte	SPM - 1	60 g	PLA / A307
2	1	Base principal de junta universal	SPM - PI - 3	30 g	PLA
3	1	Base de junta universal	SPM - EN - 2	12 g	MDF 3 mm / A307
4	2	Base 4 de junta universal	SPM - PI - 4	10 g	PLA
5	1	Junta universal	SPM - PI - 5	15 g	PLA
6	4	Tubo de aluminio PFK Ø 8 mm, 1 mm de espesor - 130 mm de longitud	_____	32 g	Aluminio PFK
7	4	M3x10 TORNILLO CABCIL PHILLIPS	_____	_____	A307
8	2	M3x13 TORNILLO CABCIL PHILLIPS	_____	_____	A307
9	2	M3 TUERCA CUADRADA	_____	_____	A307
10	4	M3x30 TORNILLO CABCIL PHILLIPS	_____	_____	A307

	UNIVERSIDAD CATÓLICA DE SANTA MARÍA Facultad de Ciencias e Ingenierías Físicas y Formales Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica	NOMBRE André Gutiérrez Suñá	FECHA Febrero 2017 a Agosto 2017	CÓDIGO DIBUJO SPM - 3	MATERIAL Rebarbar y romper las aristas vivas. Acabado con lija fina.	CONTROL DE VUELO Y TRAYECTORIA 3 DE UN CUADRICÓPTERO		
	DISEÑADO André Gutiérrez Suñá	FECHA Agosto 2017	PLANO ESTRUCTURA SUPERIOR DE SOPORTE	ACABADO MDF de 3 mm de espesor / Filamento de PLA de 1.75 mm - blanco / A307	PE SO 159 g			
	DIBUJADO Beetle Craft S.A.C	FECHA Agosto 2017	CONFIDENCIAL: Este plano y la información que contiene, son propiedad privada. Su uso y reproducción sin autorización, están prohibidos.	DIMENSIONES mm	NORMA ANSI Y 14.5 - 2009	REVISIÓN 1.0		
	REVISADO André Gutiérrez Suñá	FECHA Agosto 2017	ESCALA 2 : 3	NORMA No escale el dibujo		HOJA 1 de 2	A3	



UNIVERSIDAD CATÓLICA DE SANTA MARÍA
Facultad de Ciencias e Ingenierías Físicas y Formales
Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica

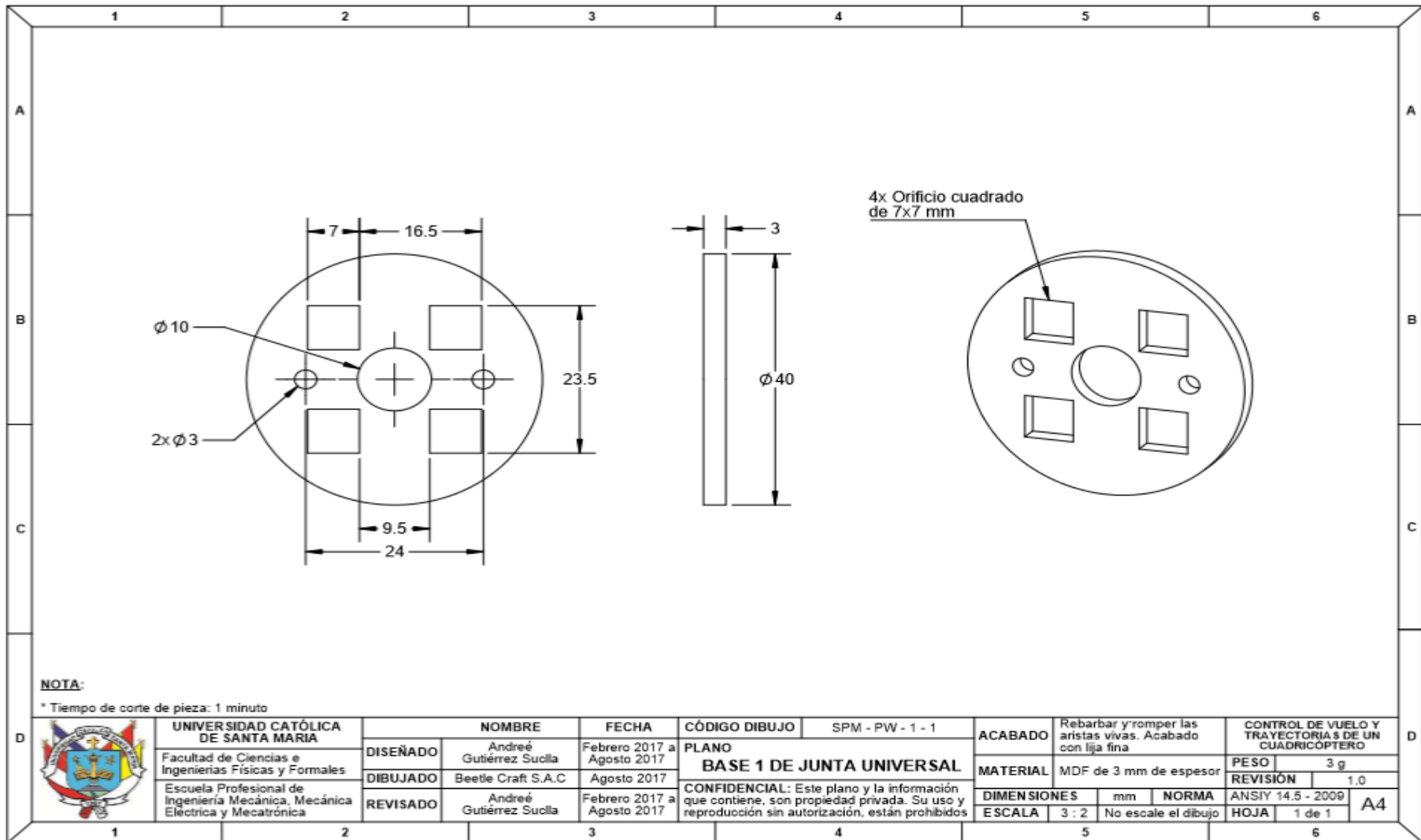
	NOMBRE	FECHA
DISEÑADO	André Gutiérrez Suñla	Febrero 2017 a Agosto 2017
DIBUJADO	Beetle Craft S.A.C	Agosto 2017
REVISADO	André Gutiérrez Suñla	Febrero 2017 a Agosto 2017

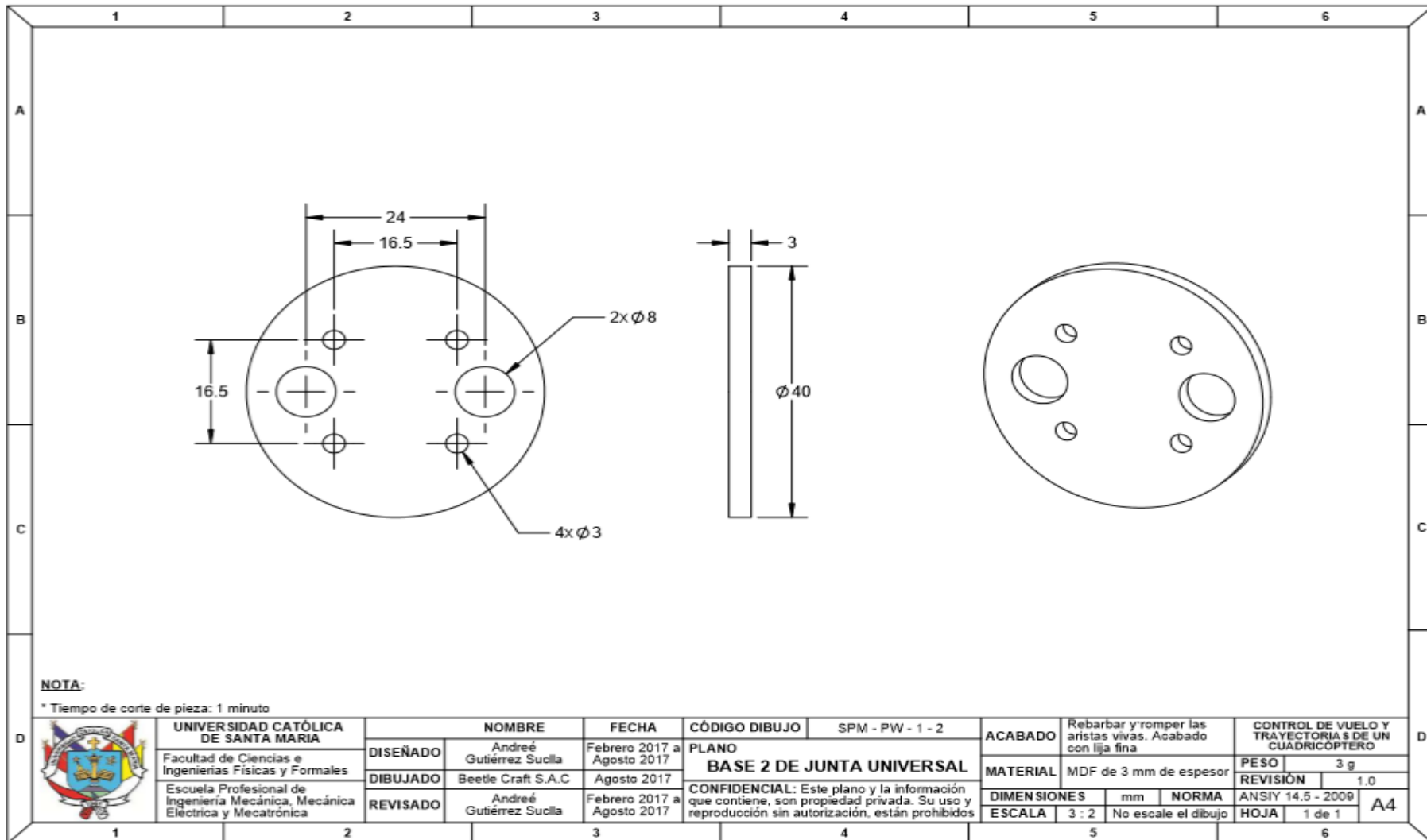
CÓDIGO DIBUJO: SPM - 3
PLANO: ESTRUCTURA SUPERIOR DE SOPORTE
CONFIDENCIAL: Este plano y la información que contiene, son propiedad privada. Su uso y reproducción sin autorización, están prohibidos

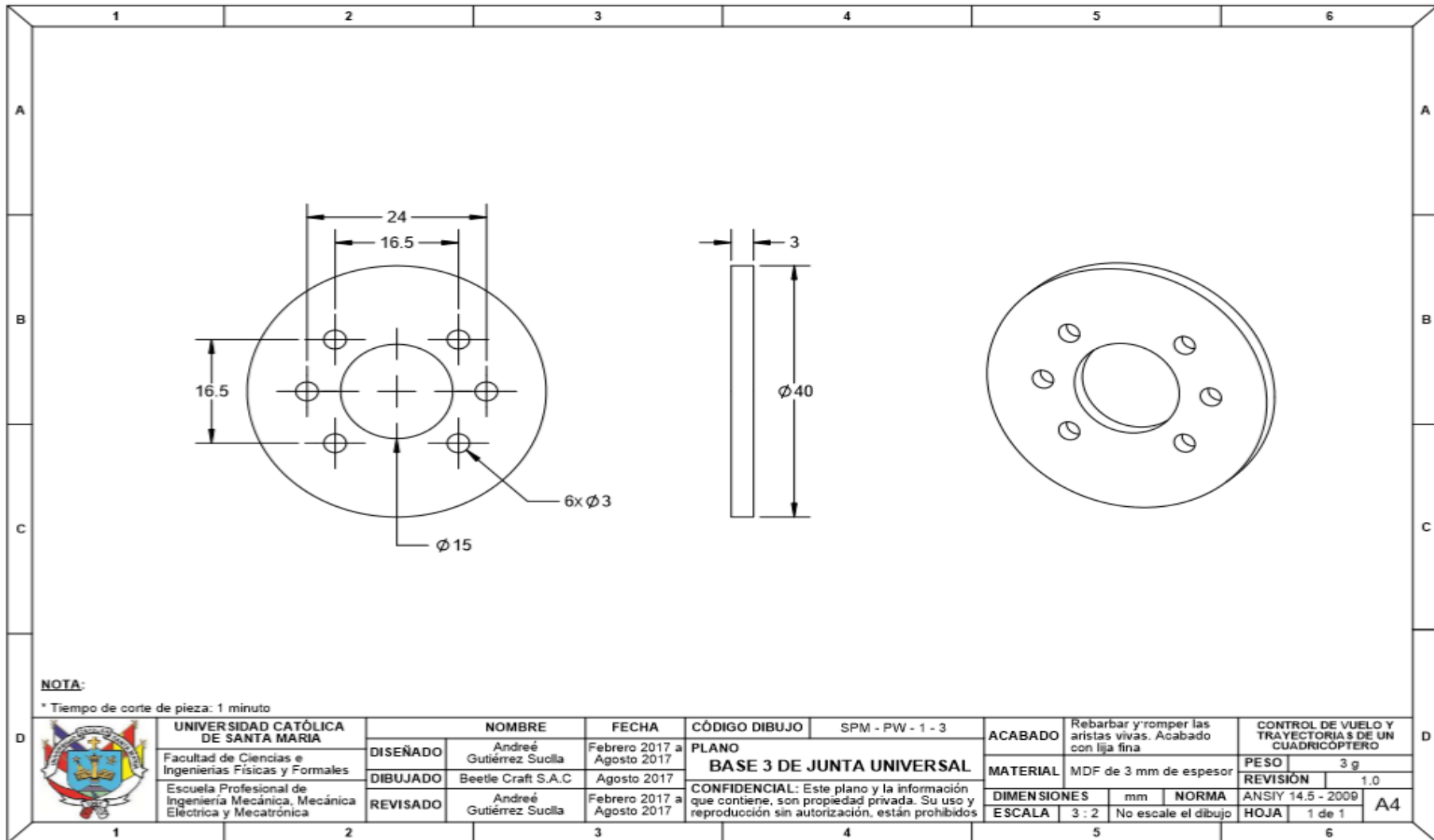
MATERIAL	Rebarbar y romper las aristas vivas. Acabado con lija fina		
ACABADO	MDF de 3 mm de espesor / Filamento de PLA de 1.75 mm - blanco / A307		
DIMENSIONES	mm	NORMA	ANSI Y 14.5 - 2009
ESCALA	2 : 3	No escale el dibujo	

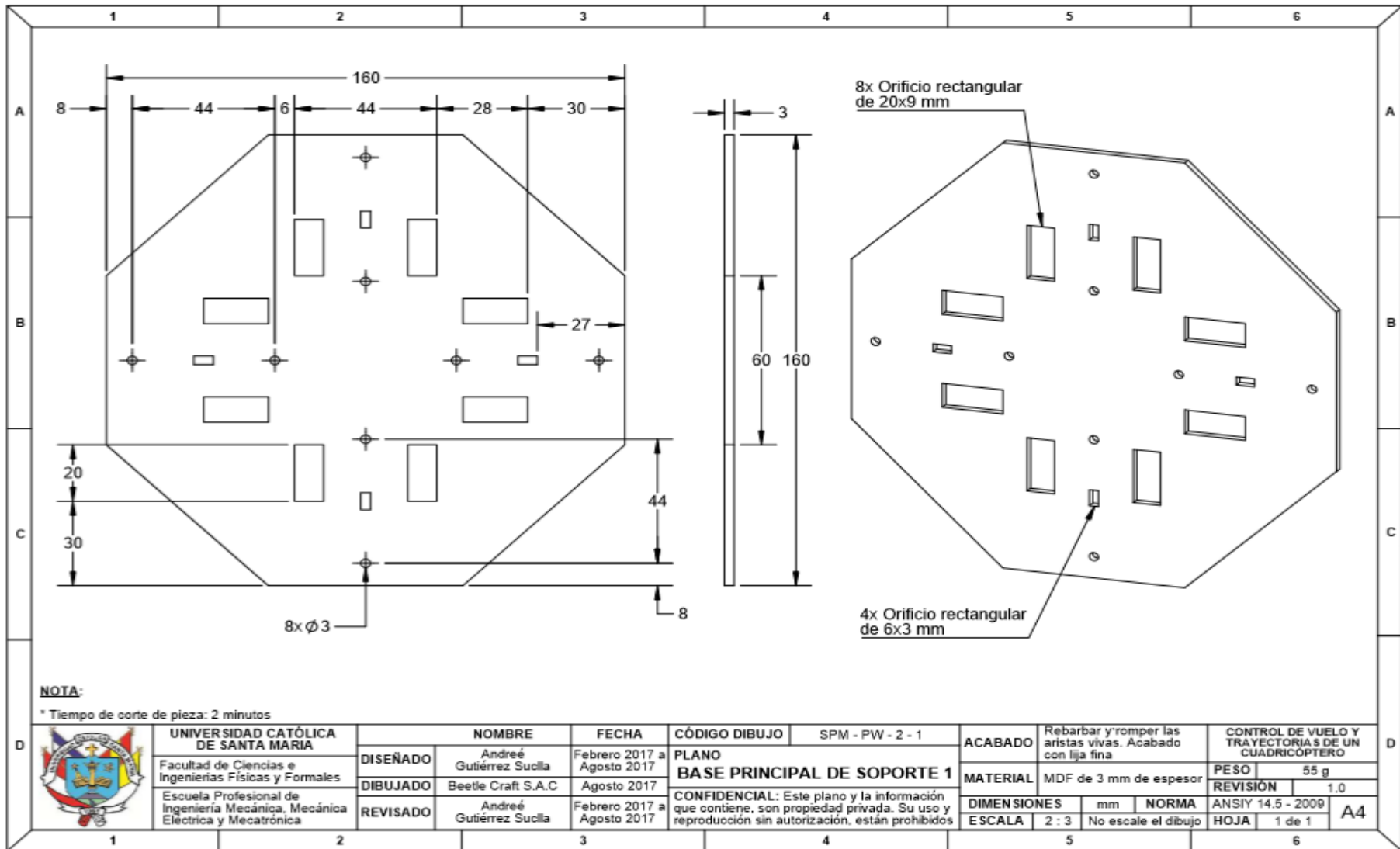
CONTROL DE VUELO Y TRAYECTORIAS DE UN CUADRICÓPTERO	
PESO	159 g
REVISIÓN	1,0
HOJA	2 de 2

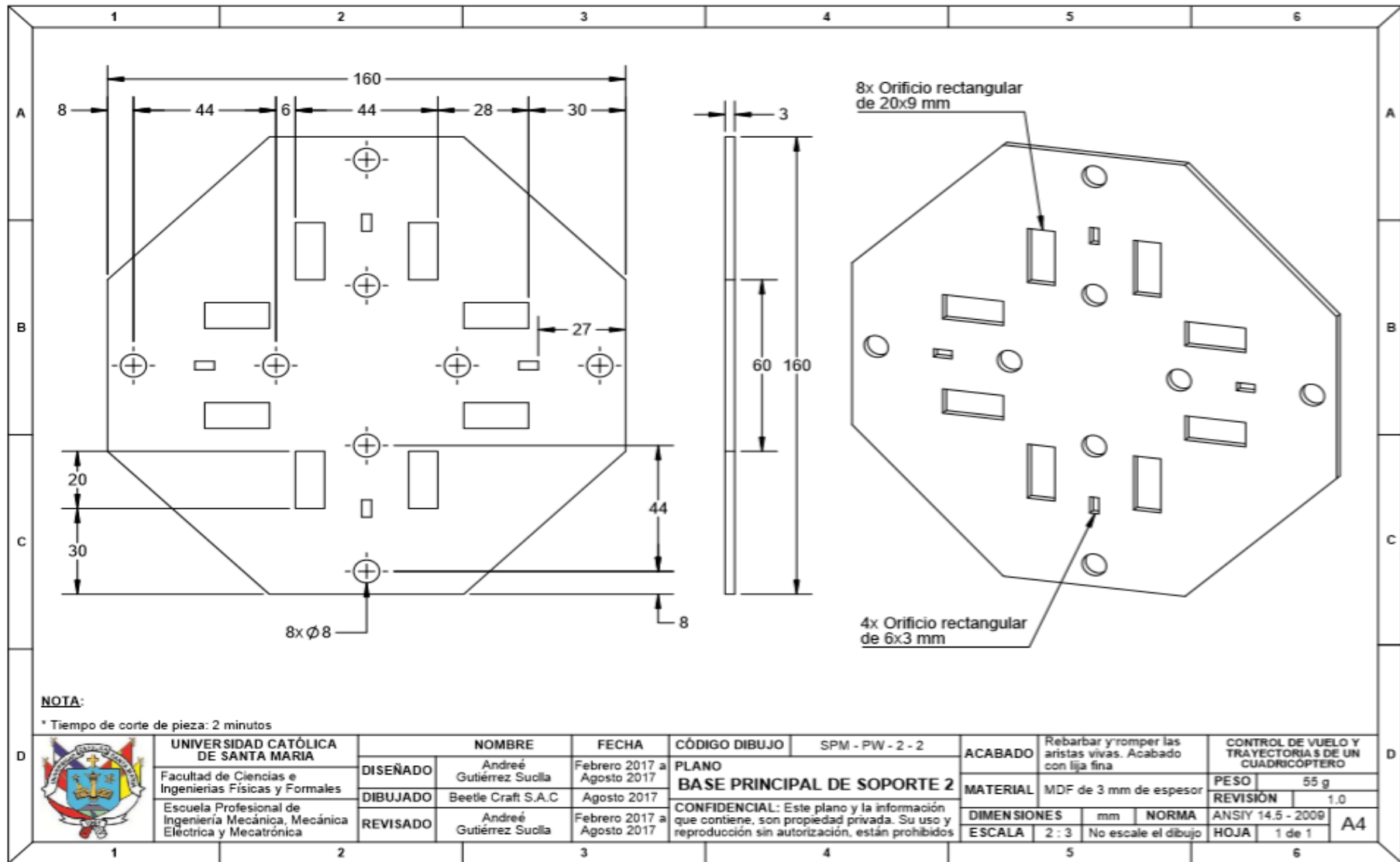
A3

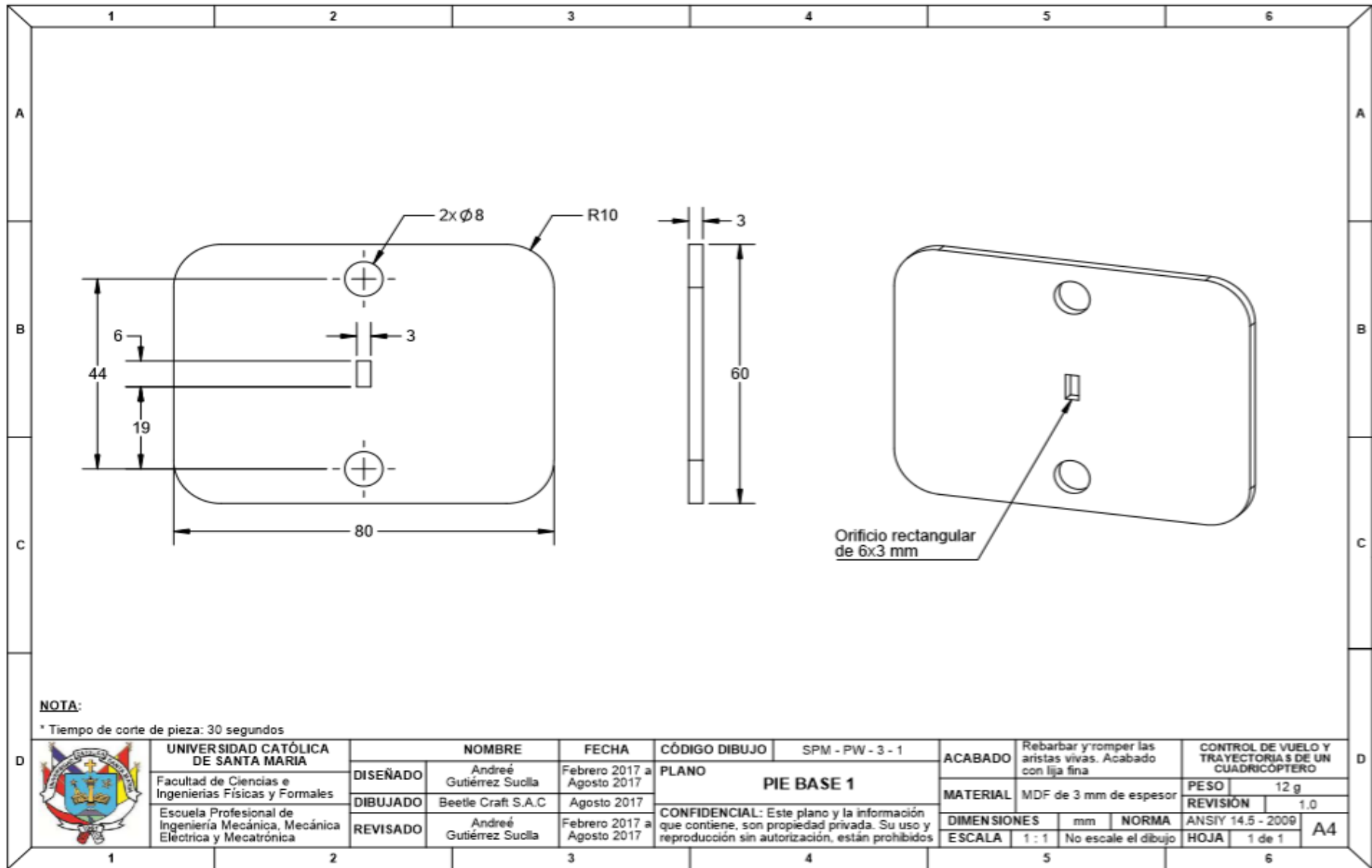


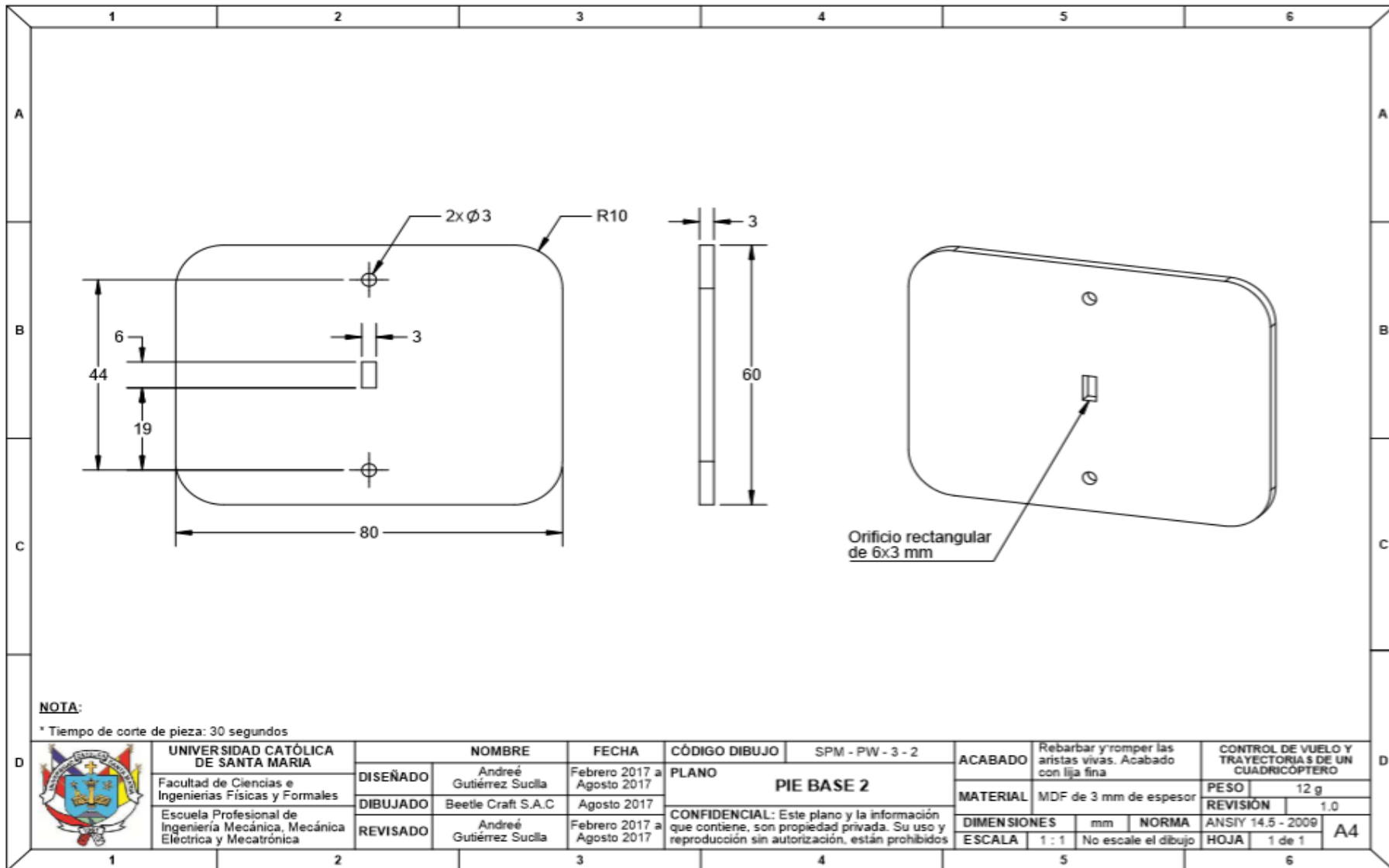


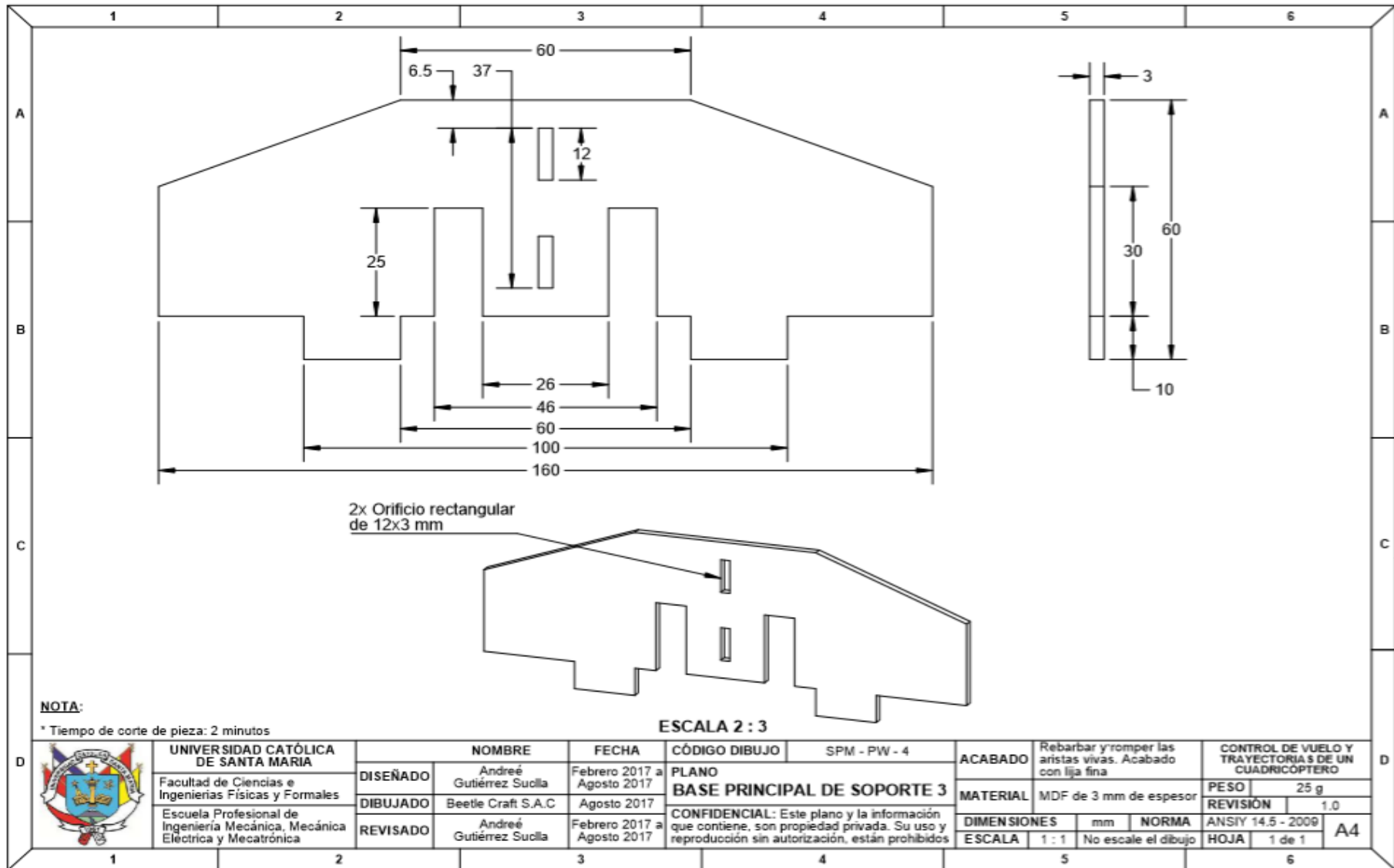


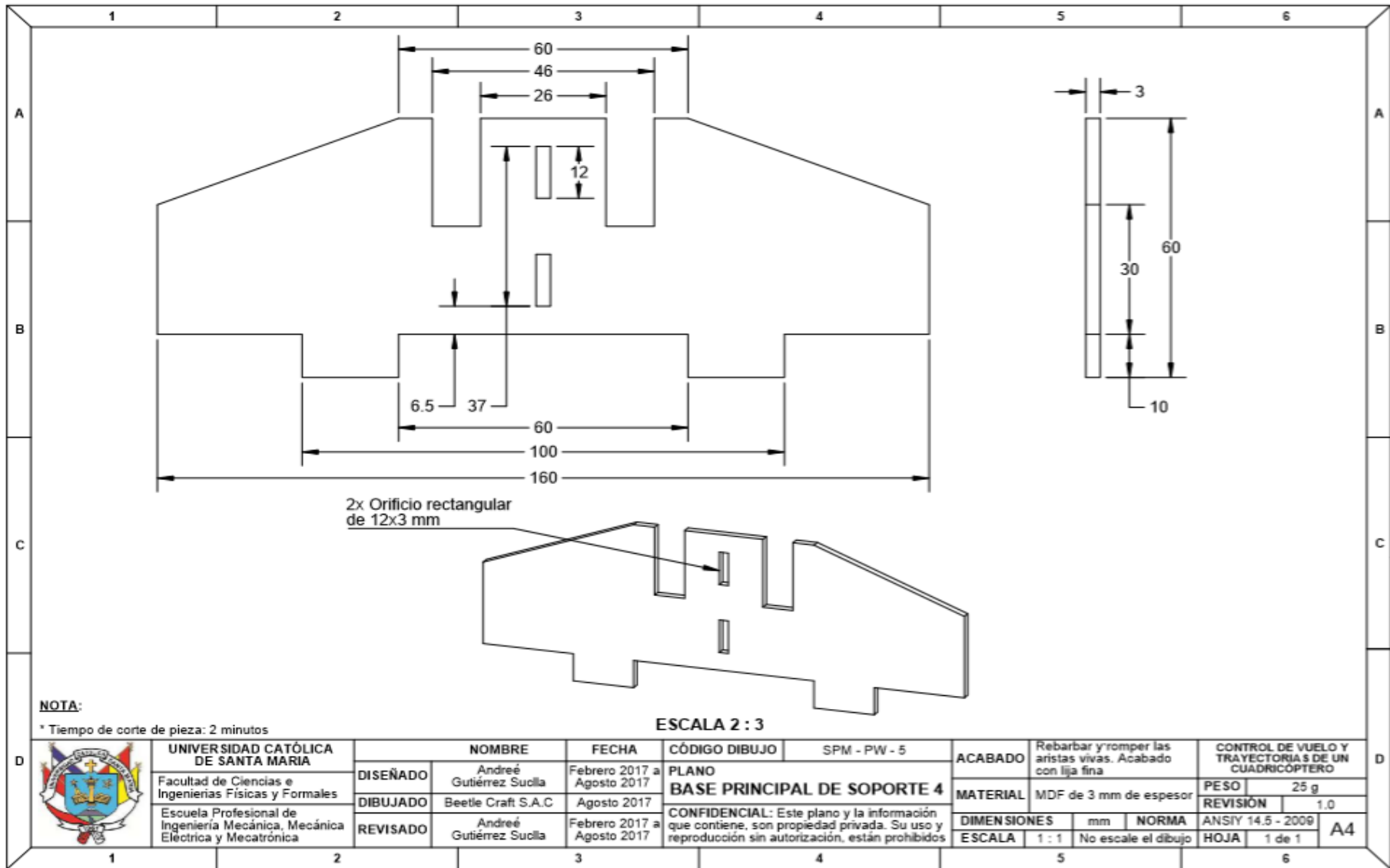


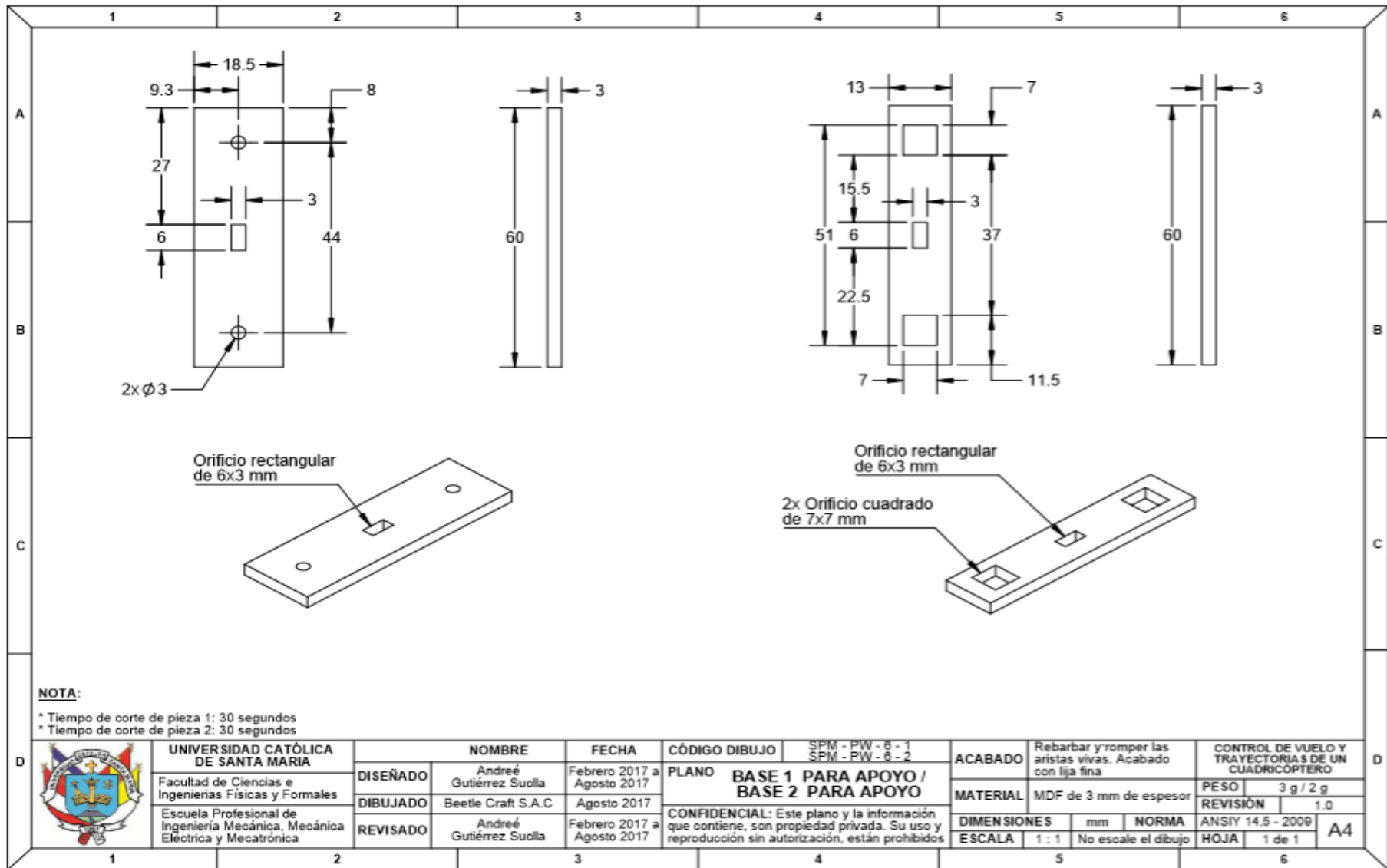


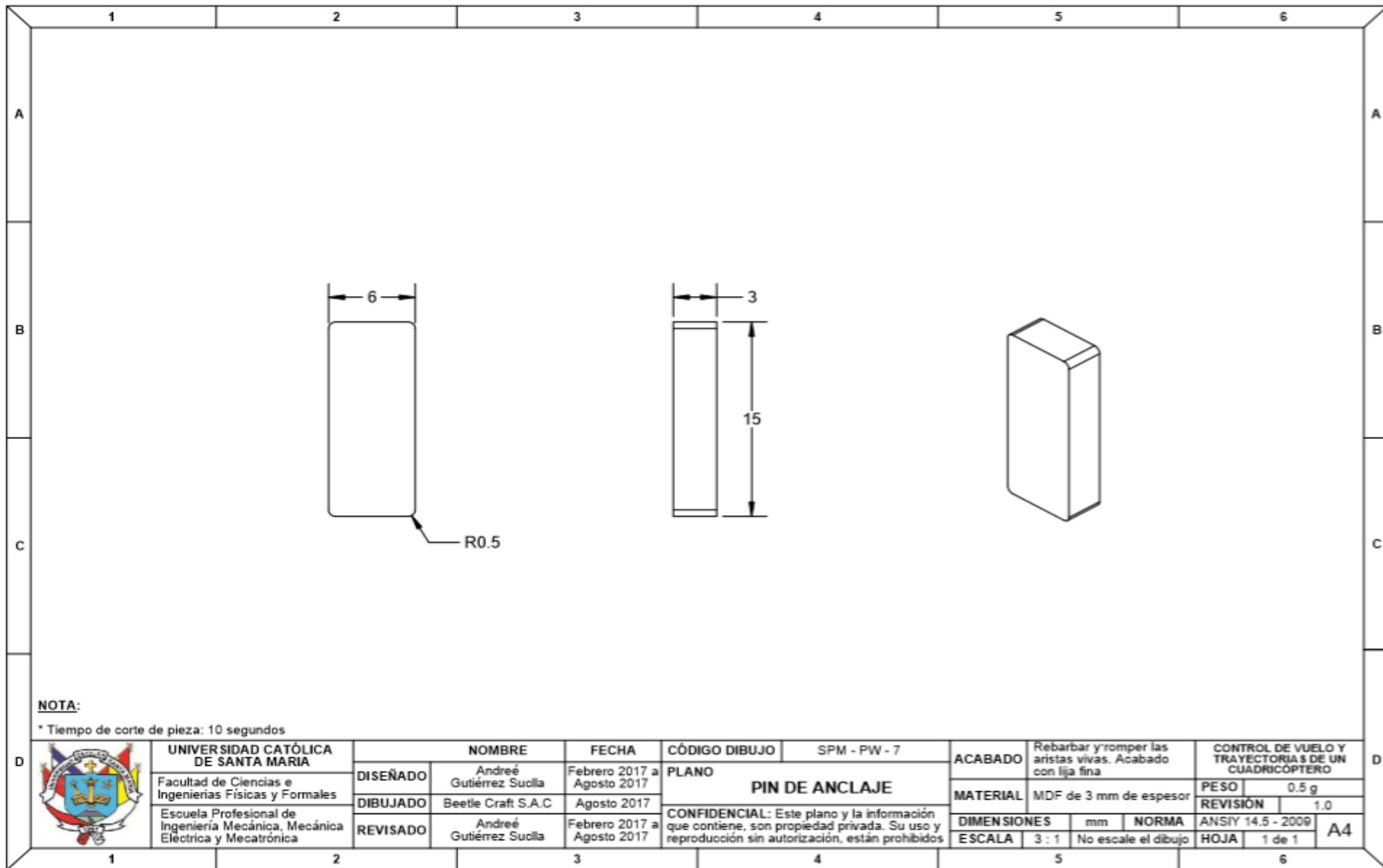









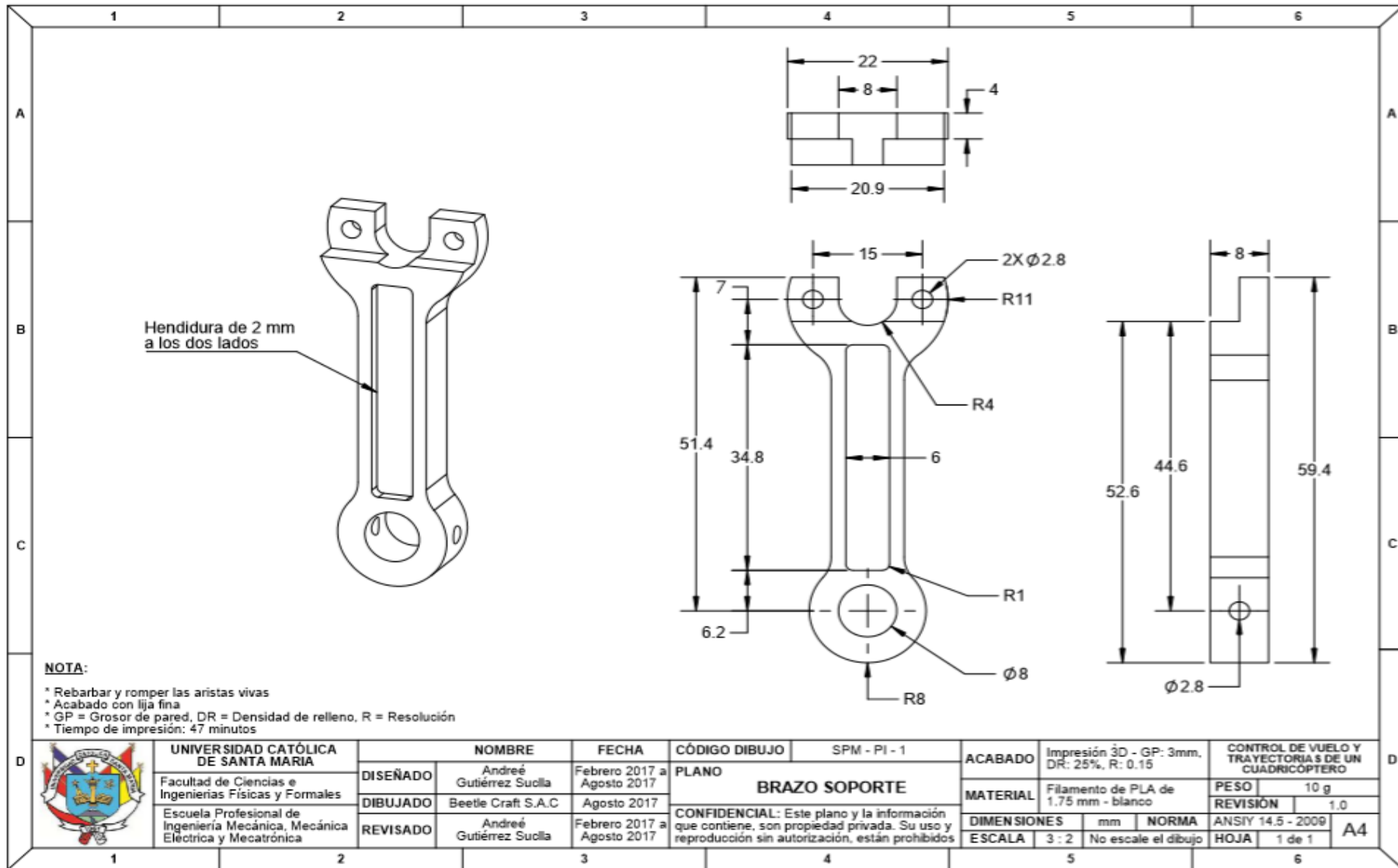


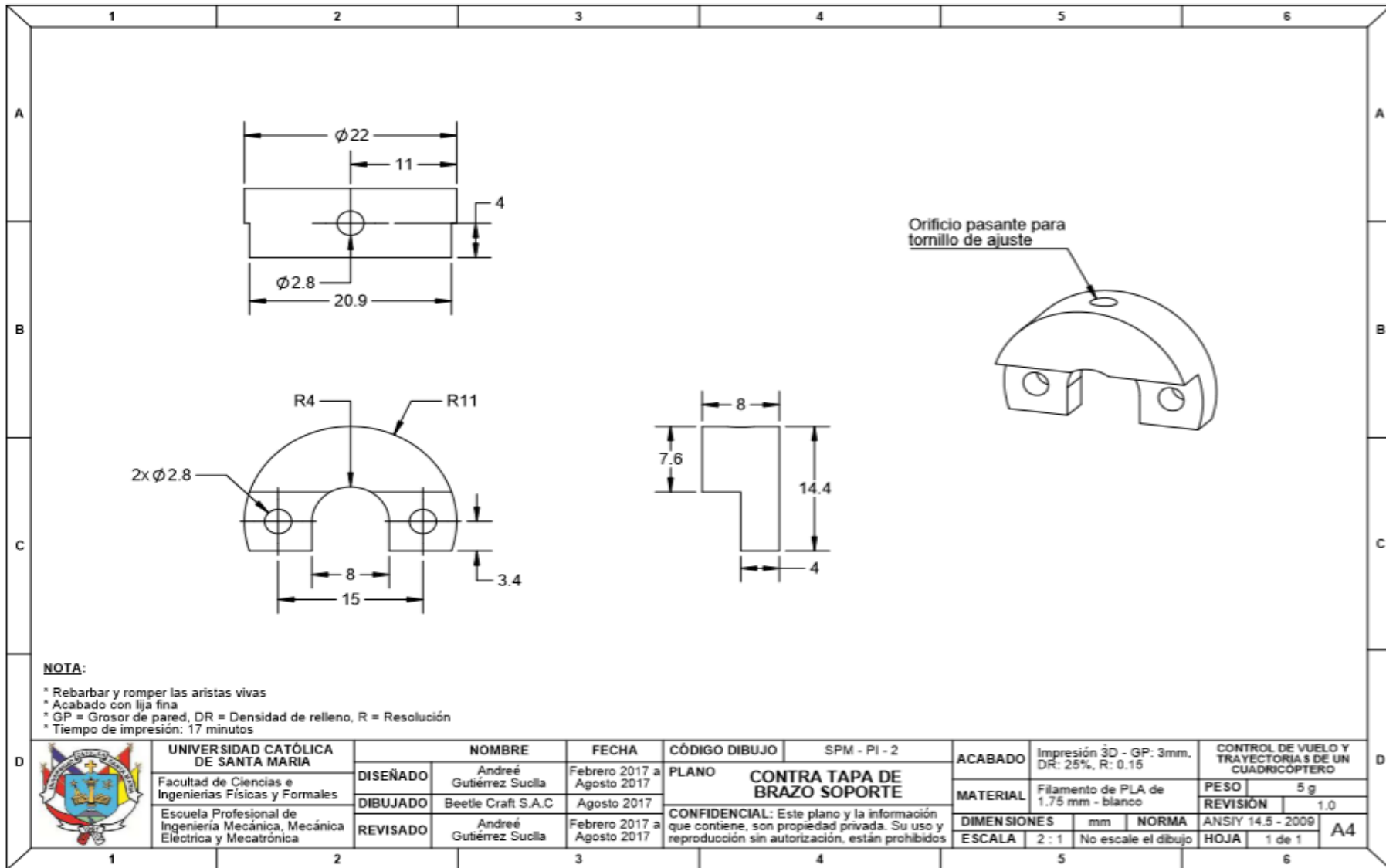


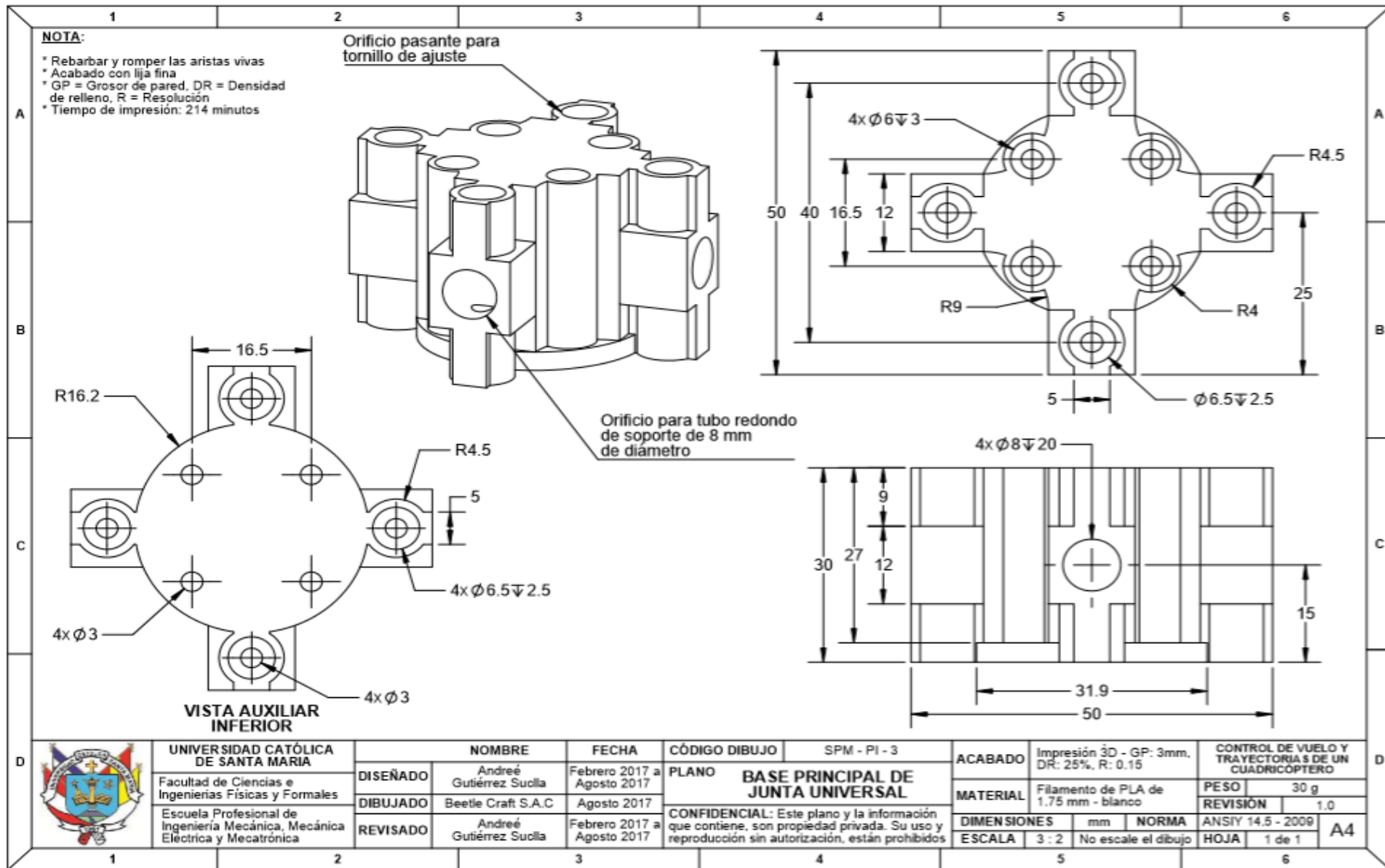
NOTA:

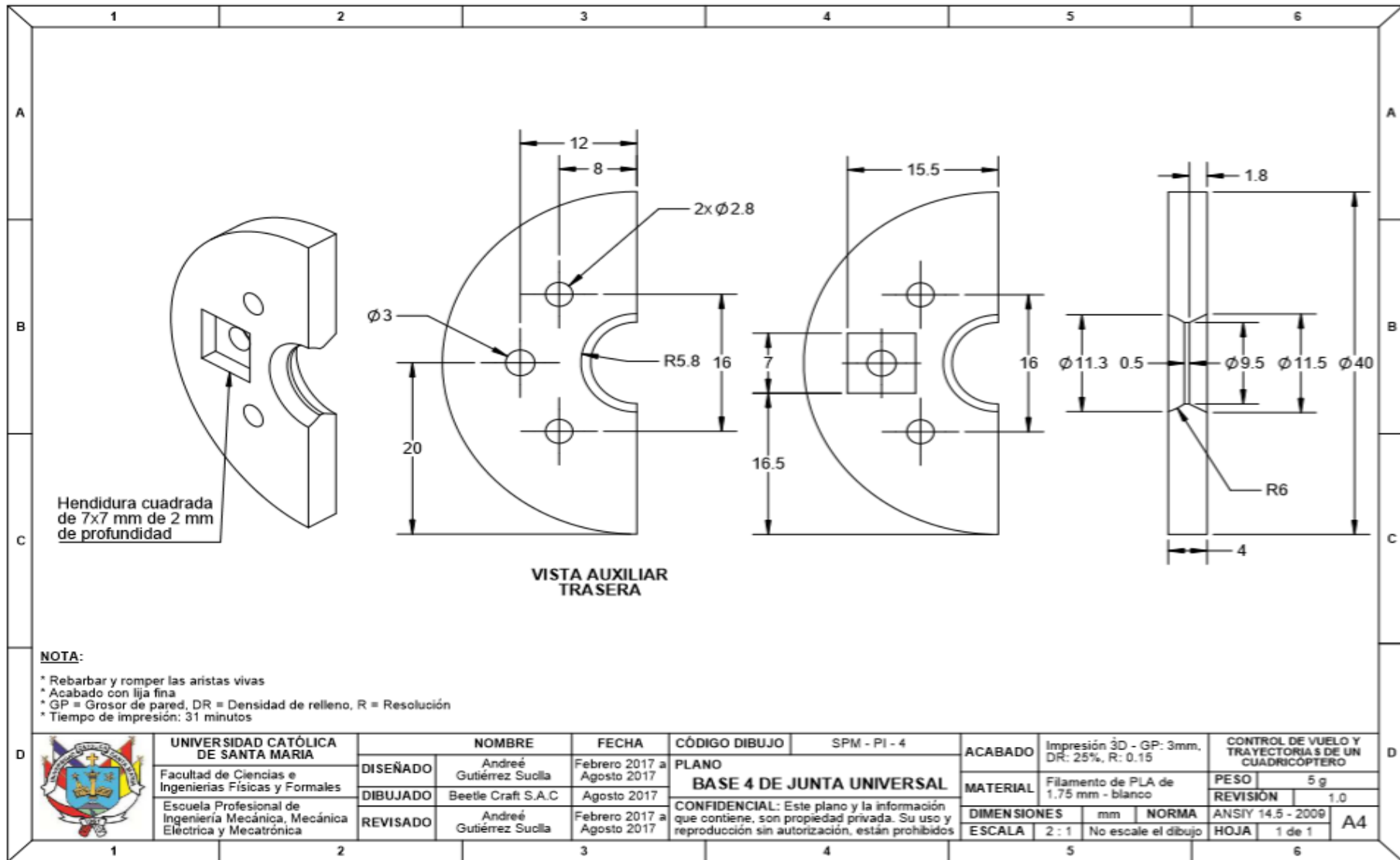
* Tiempo de corte de pieza: 10 segundos

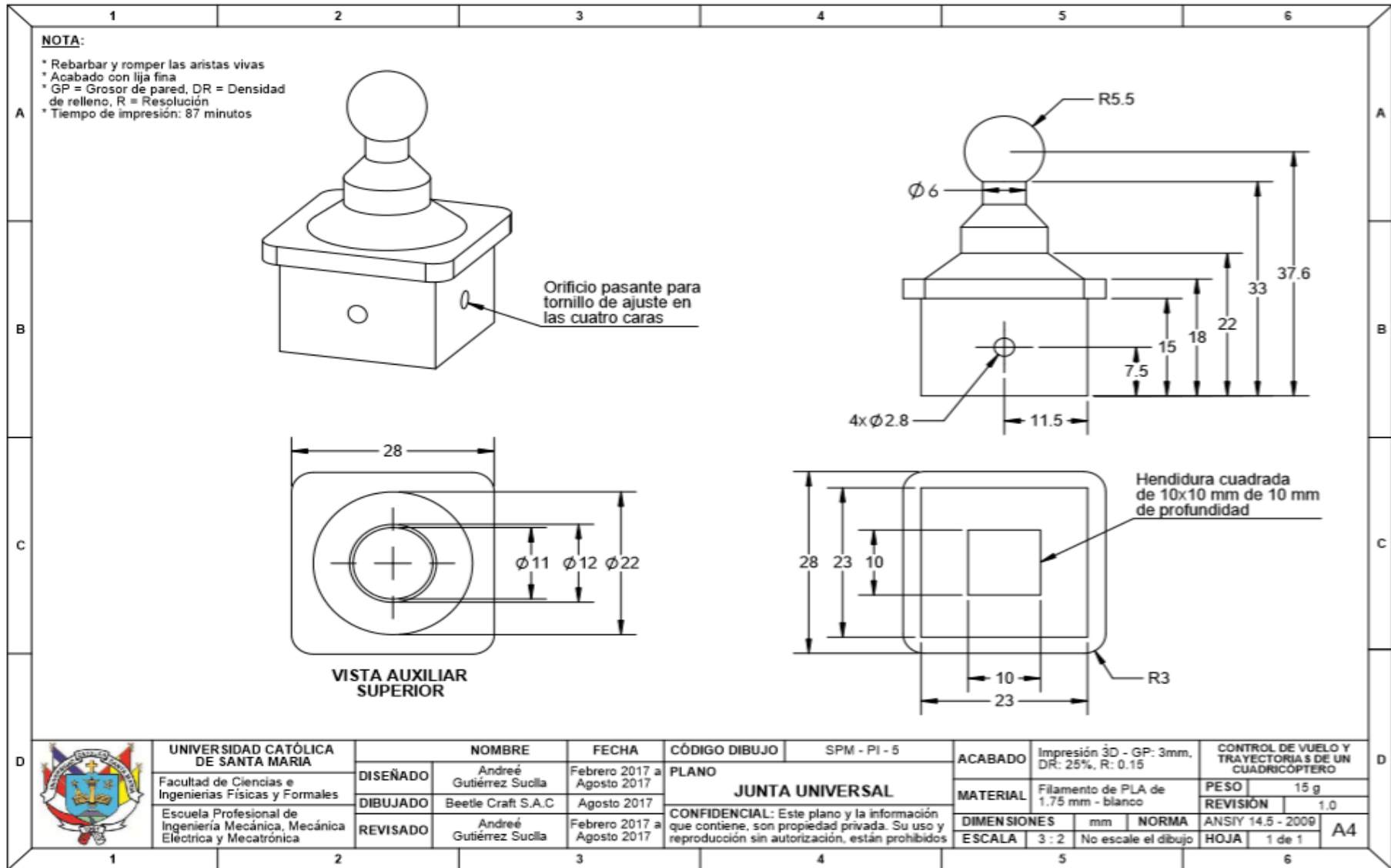
	UNIVERSIDAD CATÓLICA DE SANTA MARÍA Facultad de Ciencias e Ingenierías Físicas y Formales Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica	NOMBRE	André Gutiérrez Suclla	FECHA	Febrero 2017 a Agosto 2017	CÓDIGO DIBUJO	SPM - PW - 7	ACABADO	Rebarbar y romper las aristas vivas. Acabado con lija fina	CONTROL DE VUELO Y TRAYECTORIAS DE UN CUADRICÓPTERO		
		DISEÑADO	André Gutiérrez Suclla			PLANO		MATERIAL	MDF de 3 mm de espesor	PESO	0,5 g	
		DIBUJADO	Beetle Craft S.A.C		Agosto 2017	PIN DE ANCLAJE			DIMENSIONES	mm	NORMA	ANSI Y 14.5 - 2009
		REVISADO	André Gutiérrez Suclla	Febrero 2017 a Agosto 2017	CONFIDENCIAL: Este plano y la información que contiene, son propiedad privada. Su uso y reproducción sin autorización, están prohibidos			ESCALA	3 : 1	No escale el dibujo	REVISIÓN	1,0
										HOJA	1 de 1	A4

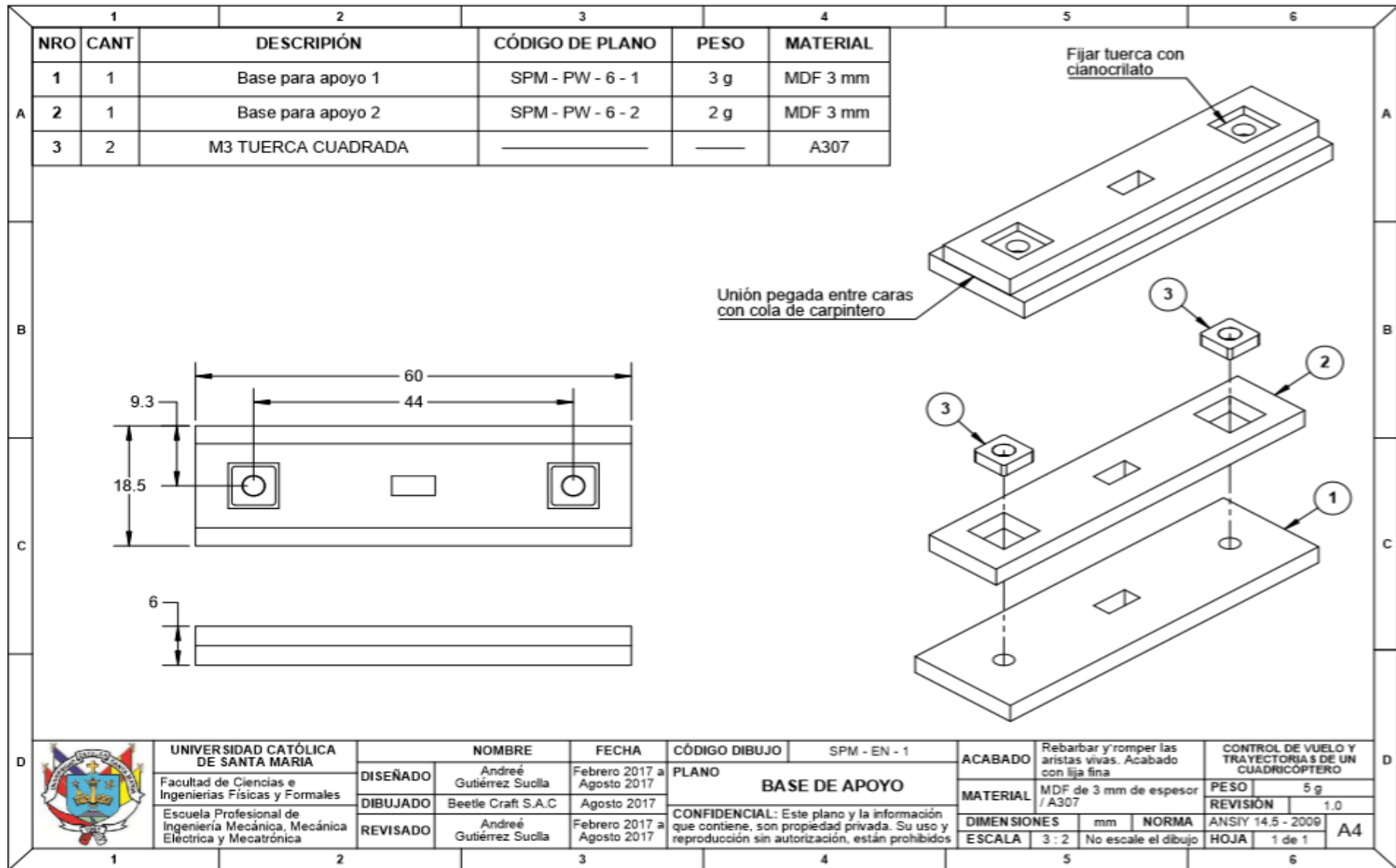


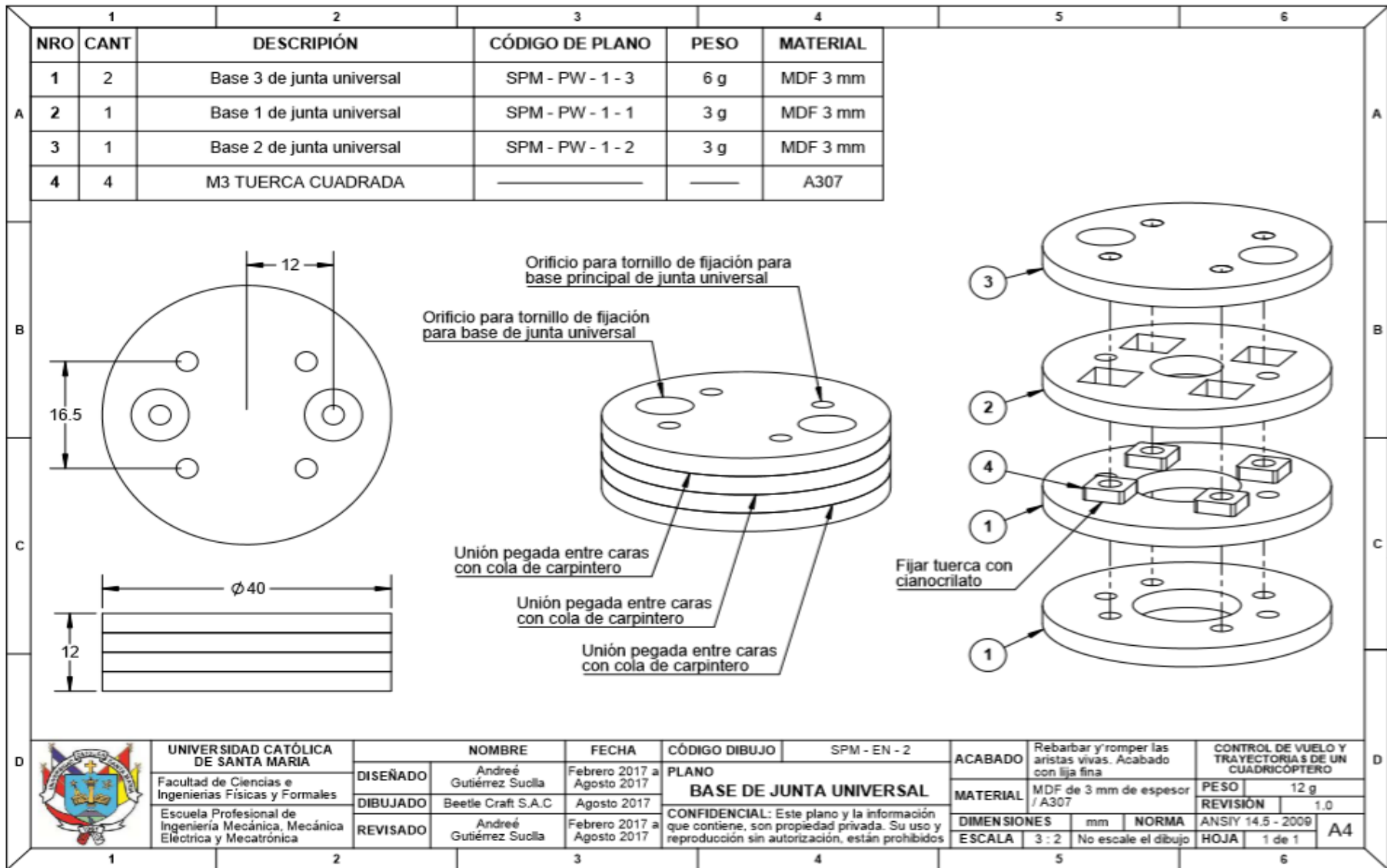






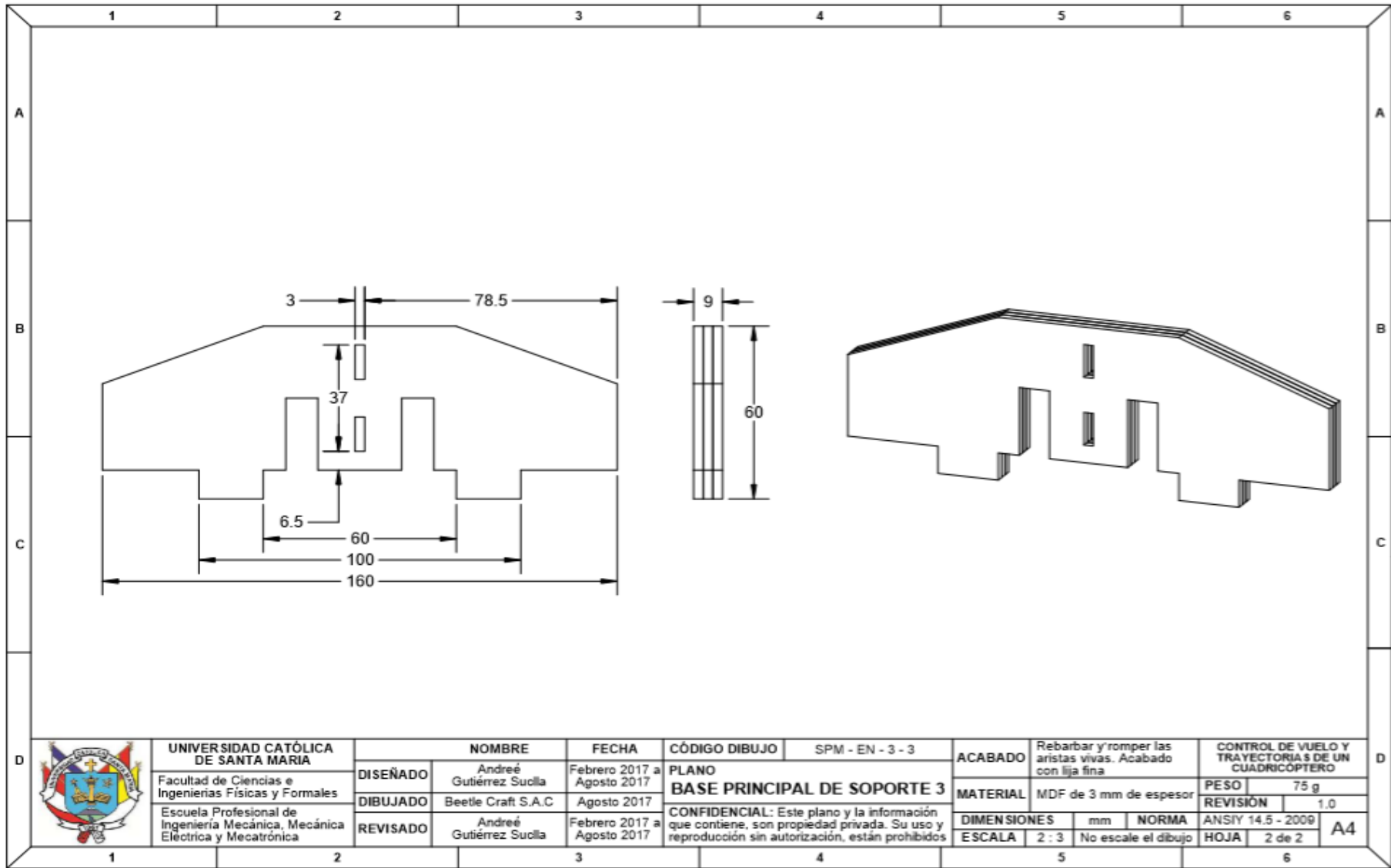


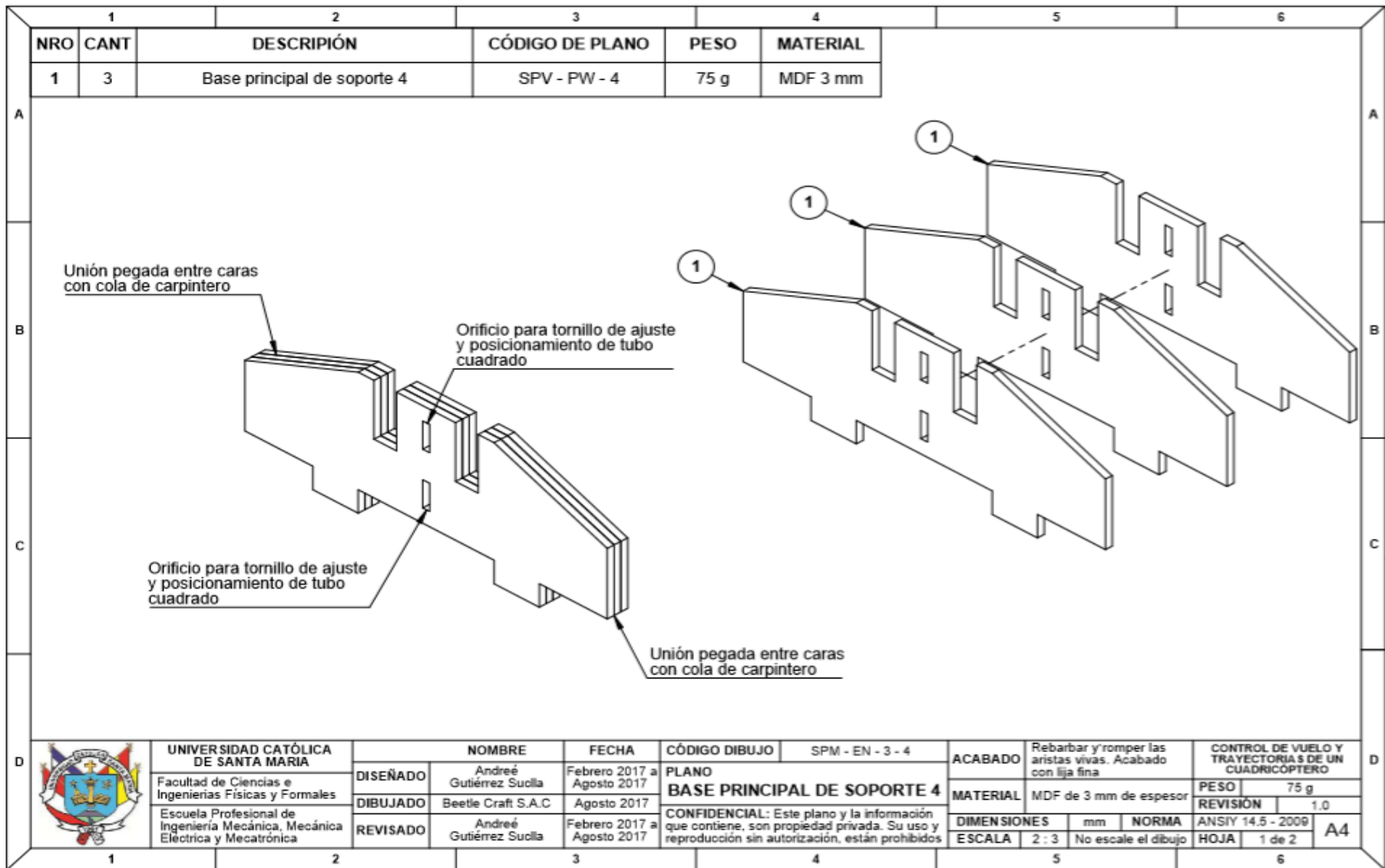


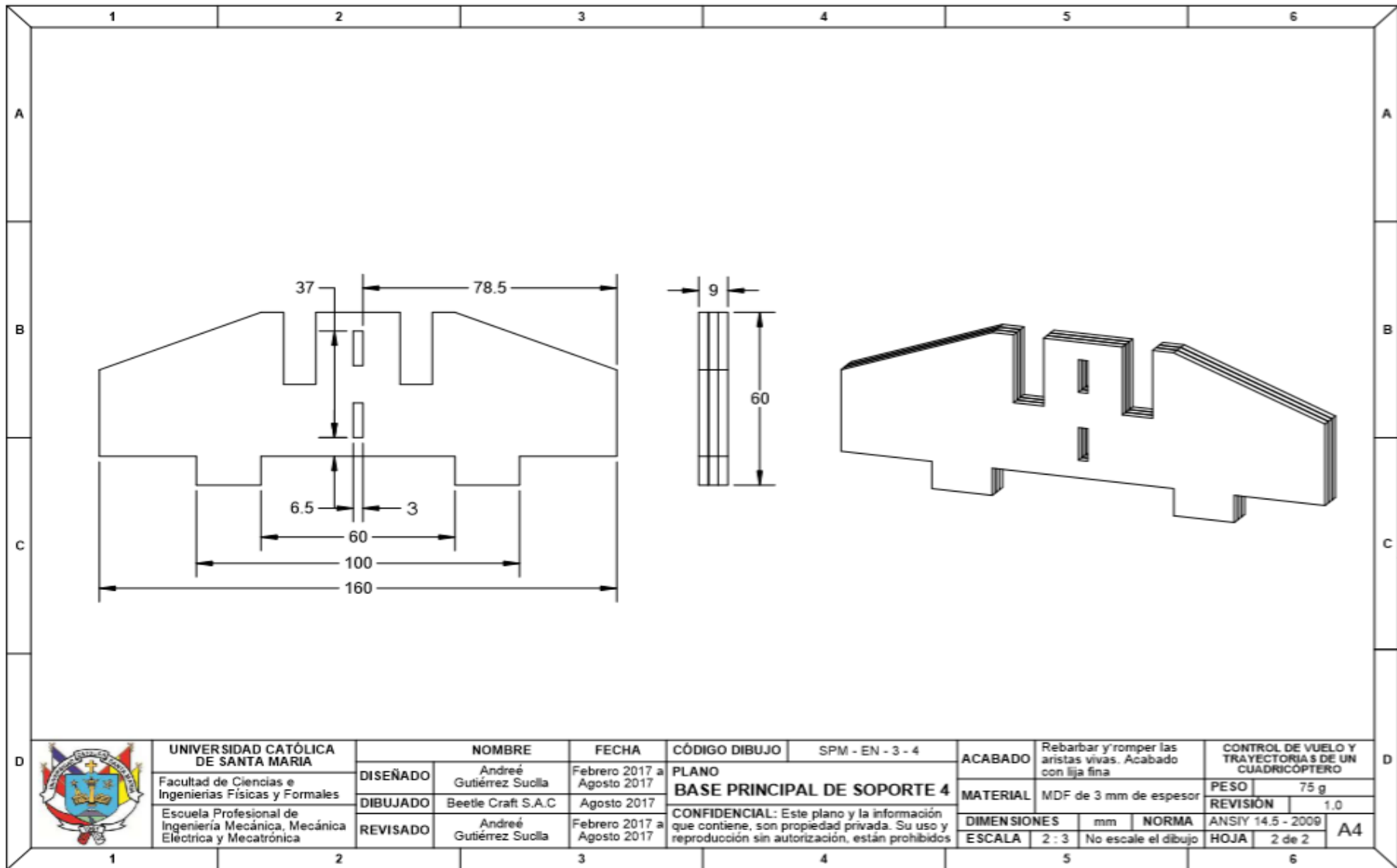


1	2		3	4	5	6
NRO	CANT	DESCRIPCIÓN	CÓDIGO DE PLANO	PESO	MATERIAL	
1	3	Base principal de soporte 3	SPV - PW - 4	75 g	MDF 3 mm	

	UNIVERSIDAD CATÓLICA DE SANTA MARÍA		NOMBRE		FECHA		CÓDIGO DIBUJO		SPM - EN - 3 - 3		ACABADO		CONTROL DE VUELO Y TRAYECTORIAS DE UN CUADRICÓPTERO	
	Facultad de Ciencias e Ingenierías Físicas y Formales		André Gutiérrez Suclla		Febrero 2017 a Agosto 2017		PLANO		Rebarbar y romper las aristas vivas. Acabado con lija fina		MDF de 3 mm de espesor		PESO 75 g	
	Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica		Beetle Craft S.A.C		Agosto 2017		BASE PRINCIPAL DE SOPORTE 3		MATERIAL		REVISIÓN 1.0		ANSI 14.5 - 2009	
			André Gutiérrez Suclla		Febrero 2017 a Agosto 2017		CONFIDENCIAL: Este plano y la información que contiene, son propiedad privada. Su uso y reproducción sin autorización, están prohibidos		DIMENSIONES mm NORMA		ESCALA 2 : 3 No escale el dibujo		HOJA 1 de 2	
												A4		







UNIVERSIDAD CATÓLICA DE SANTA MARÍA
 Facultad de Ciencias e Ingenierías Físicas y Formales
 Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica

	NOMBRE	FECHA
DISEÑADO	André Gutiérrez Suella	Febrero 2017 a Agosto 2017
DIBUJADO	Beetle Craft S.A.C	Agosto 2017
REVISADO	André Gutiérrez Suella	Febrero 2017 a Agosto 2017

CÓDIGO DIBUJO SPM - EN - 3 - 4

PLANO
BASE PRINCIPAL DE SOPORTE 4

ACABADO

Rebarbar y romper las aristas vivas. Acabado con lija fina

CONTROL DE VUELO Y TRAYECTORIAS DE UN CUADRICÓPTERO

MATERIAL

MDF de 3 mm de espesor

PESO 75 g
 REVISIÓN 1.0

DIMENSIONES

mm

NORMA ANSIY 14.5 - 2009

ESCALA

2 : 3 No escale el dibujo

HOJA 2 de 2

A4

1		2		3		4		5		6	
NRO	CANT	DESCRIPCIÓN		CÓDIGO DE PLANO		PESO	MATERIAL				
1	1	Pin de anclaje		SPM - PW - 7		0.5 g	MDF 3 mm				
2	1	Base de apoyo		SPM - EN - 1		5 g	MDF 3mm / A307				
3	1	Pie base 2		SPM - PW - 3 - 2		12 g	MDF 3 mm				
4	2	Pie base 1		SPM - PW - 3 - 1		24 g	MDF 3 mm				
5	2	M3x10 TORNF CABCIL PHILLIPS		_____		_____	A307				

Tornillo de ajuste de pie de base con perfil estructural de aluminio

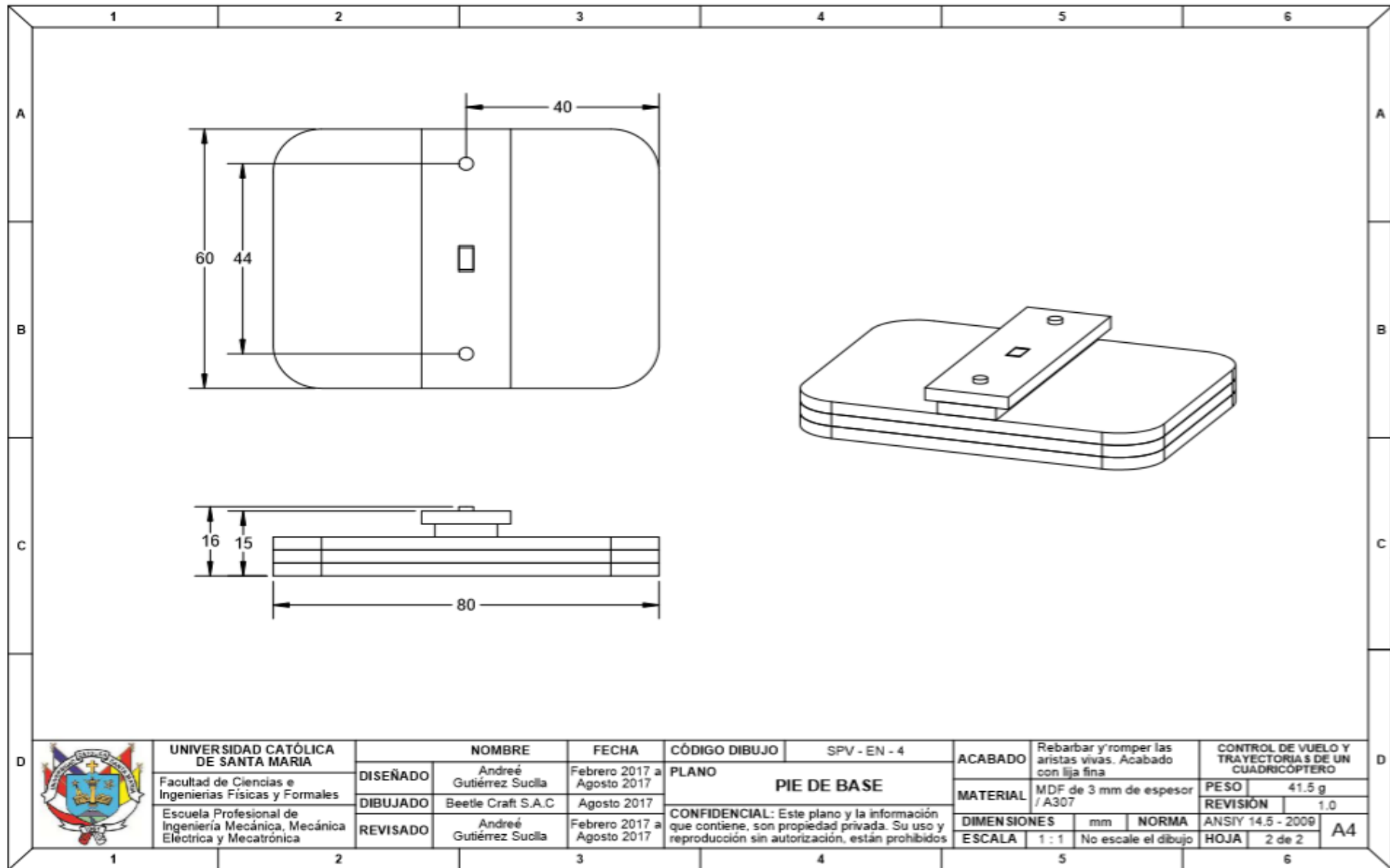
Pin de encuadre de placas durante ensamblaje

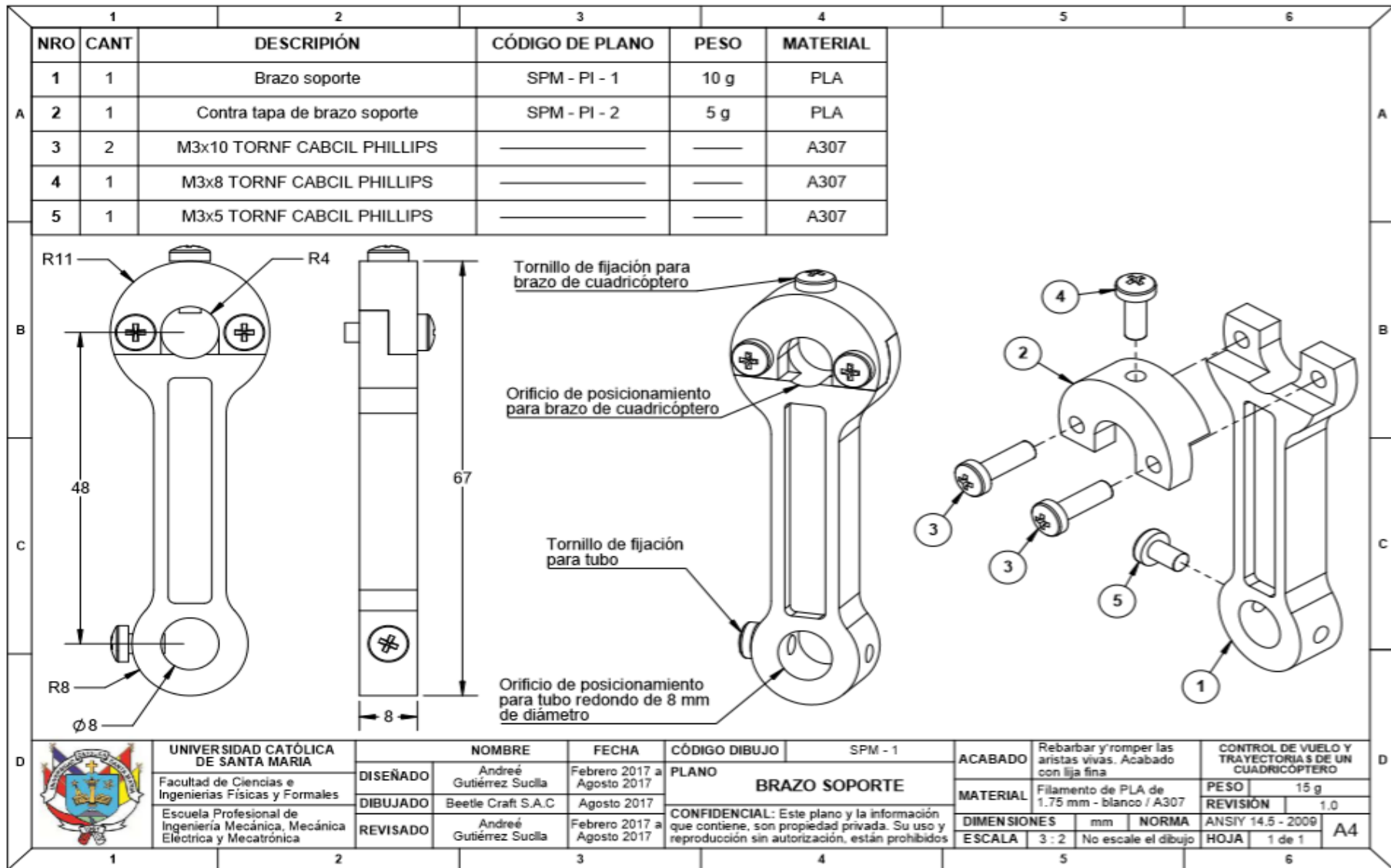
Unión pegada entre caras con cola de carpintero

Unión pegada entre caras con cola de carpintero

Unión pegada entre caras con cola de carpintero

	UNIVERSIDAD CATÓLICA DE SANTA MARÍA		NOMBRE		FECHA		CÓDIGO DIBUJO		ACABADO Rebarbar y romper las aristas vivas. Acabado con lija fina		CONTROL DE VUELO Y TRAYECTORIAS DE UN CUADRICOPTERO	
	Facultad de Ciencias e Ingenierías Físicas y Formales		André Gutiérrez Suolla		Febrero 2017 a Agosto 2017		SPM - EN - 4				PLANO PIE DE BASE	
	Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica		Beetle Craft S.A.C		Agosto 2017		CONFIDENCIAL: Este plano y la información que contiene, son propiedad privada. Su uso y reproducción sin autorización, están prohibidos		MATERIAL MDF de 3 mm de espesor / A307		REVISIÓN 1.0	
			André Gutiérrez Suolla		Febrero 2017 a Agosto 2017				DIMENSIONES mm		NORMA ANSI Y 14.5 - 2009	
								ESCALA 1 : 1		No escale el dibujo		A4





Anexo 04: AR Drone Developer Guide 3-12p.



1 |

Introduction

Welcome to the AR.Drone 2.0 Software Development Kit!

The AR.Drone 2.0 product and the provided host interface example have innovative and exciting features such as:

- intuitive touch and tilt flight controls
- live video streaming
- video recording and photo shooting
- updated Euler angles of the AR Drone
- embedded tag detection for augmented reality games

The AR.Drone 2.0 SDK allows third party developers to develop and distribute new games based on AR.Drone 2.0 product for Wifi, motion sensing mobile devices like the Apple iPhone, iPad, iPod touch, personal computers or Android devices.

To download the AR.Drone 2.0 SDK, third party developers will have to register and accept the AR.Drone 2.0 SDK License Agreement terms and conditions. Upon final approval from Parrot, they will have access to the AR.Drone 2.0 SDK download web page.

4

This SDK includes :

- this document explaining how to use the SDK, and describes the drone communications protocols;
- the AR.Drone 2.0 Library (**ARDroneLIB**), which provides the APIs needed to easily communicate and configure an AR.Drone 2.0 product;
- the AR.Drone 2.0 Tool (**ARDroneTool**) library, which provides a fully functional drone client where developers only have to insert their custom application specific code;
- the AR.Drone 2.0 Control Engine library which provides an intuitive control interface developed by Parrot for remotely controlling the AR.Drone 2.0 product from an iOS Device;
- several code examples that show how to control the drone from a Linux personal computer.
- source code for iOS and Android¹ AR.FreeFlight 2.0 applications

Where should I start ?

Please first read chapter 2 to get an overview of the drone abilities and a bit of vocabulary.

You then have the choice between :

- using the provided library 5 and modifying the provided examples (9, 10) to suit your needs
- trying to write your own software from scratch by following the specifications given in 6 and 7.

¹May not be available as part of the first release of the AR.Drone 2.0 SDK



2 |

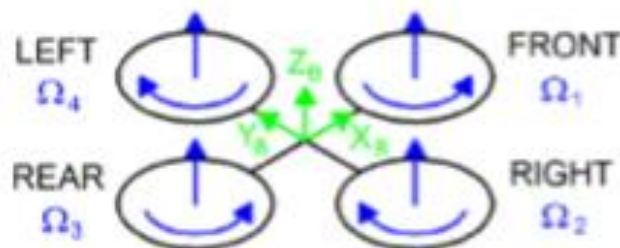
AR.Drone 2.0 Overview



2.1 Introduction to quadrotor UAV

AR.Drone 2.0 is a quadrotor. The mechanical structure comprises four rotors attached to the four ends of a crossing to which the battery and the RF hardware are attached.

Each pair of opposite rotors is turning the same way. One pair is turning clockwise and the other anti-clockwise.



5

6

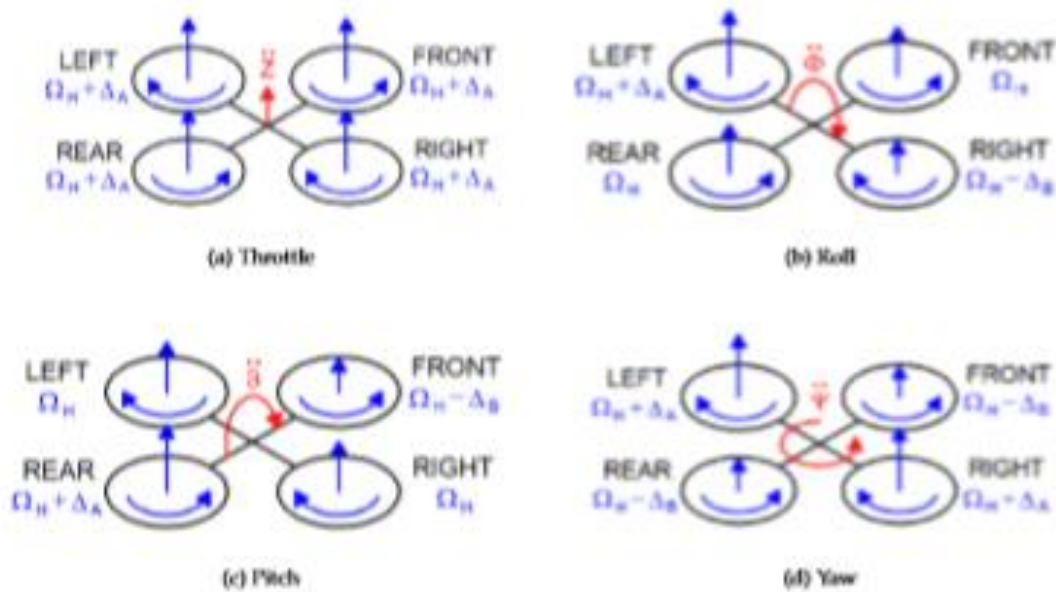
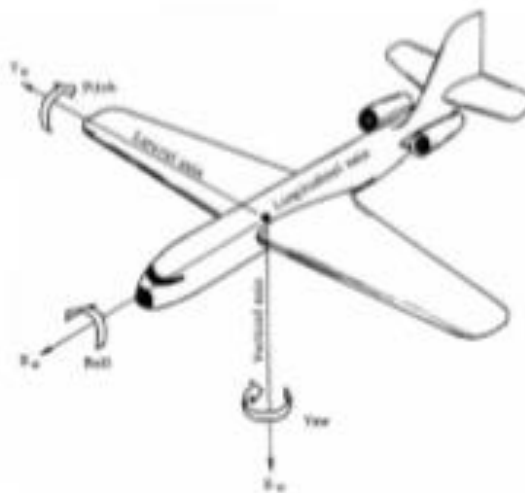


Figure 2.1: Drone movements

Manoeuvres are obtained by changing pitch, roll and yaw angles of the AR.Drone 2.0 .



Varying left and right rotors speeds the opposite way yields roll movement. This allows to go forth and back.

Varying front and rear rotors speeds the opposite way yields pitch movement.

Varying each rotor pair speed the opposite way yields yaw movement. This allows turning left and right.



Figure 2.2: Drone hulls

2.2 Indoor and outdoor design configurations

When flying outdoor the AR.Drone 2.0 can be set in a light and low wind drag configuration (2.2b). Flying indoor requires the drone to be protected by external bumpers (2.2a).

When flying indoor, tags can be added on the external hull to allow several drones to easily detect each others via their cameras.

2.3 Engines

The AR.Drone 2.0 is powered with brushless engines with three phases current controlled by a micro-controller

The AR.Drone 2.0 automatically detects the type of engines that are plugged and automatically adjusts engine controls. The AR.Drone 2.0 detects if all the engines are turning or are stopped. In case a rotating propeller encounters any obstacle, the AR.Drone 2.0 detects if any of the propeller is blocked and in such case stops all engines immediately. This protection system prevents repeated shocks.

2.4 LiPo batteries

The AR.Drone 2.0 uses a charged 1000mAh, 11.1V LiPo batteries to fly. While flying the battery voltage decreases from full charge (12.5 Volts) to low charge (9 Volts). The AR.Drone 2.0 monitors battery voltage and converts this voltage into a battery life percentage (100% if battery is full, 0% if battery is low). When the drone detects a low battery voltage, it first sends a warning message to the user, then automatically lands. If the voltage reaches a critical level, the whole system is shut down to prevent any unexpected behaviour.

8

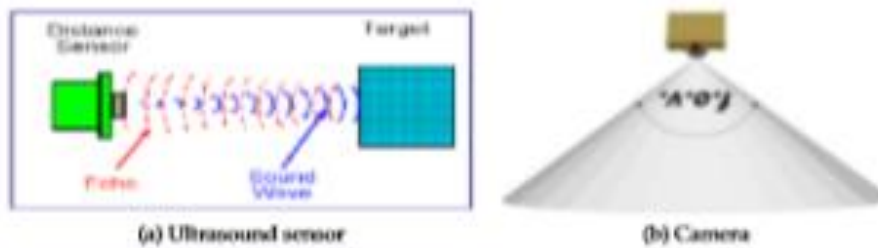


Figure 2.3: Drone Sensors

2.5 Motion sensors

The AR.Drone has many motions sensors. They are located below the central hull.

The AR.Drone 1.0 features a 6 DOF, MEMS-based, miniaturized inertial measurement unit. It provides the software with pitch, roll and yaw measurements.

Inertial measurements are used for automatic pitch, roll and yaw stabilization and assisted tilting control. They are needed for generating realistic augmented reality effects.

An ultrasound telemeter provides with altitude measures for automatic altitude stabilization and assisted vertical speed control.

A camera aiming towards the ground provides with ground speed measures for automatic hovering and trimming.

The AR.Drone 2.0 Add 3 DOF to the IMU with a 3 axis magnetometer (mandatory for Absolute Control mode). It also adds a pressure sensor to allow altitude measurements at any height.

2.6 Assisted control of basic manoeuvres

Usually quadrotor remote controls feature levers and trims for controlling UAV pitch, roll, yaw and throttle. Basic manoeuvres include take-off, trimming, hovering with constant altitude, and landing. It generally takes hours to a beginner and many UAV crashes before executing safely these basic manoeuvres.

Thanks to the AR.Drone 2.0 onboard sensors take-off, hovering, trimming and landing are now completely automatic and all manoeuvres are completely assisted.

User interface for basics controls on host can now be greatly simplified :

- When landed push *take-off* button to automatically start engines, take-off and hover at a pre-determined altitude.
- When flying push *landing* button to automatically land and stop engines.
- Press *turn left* button to turn the AR Drone automatically to the left at a predetermined speed. Otherwise the AR Drone automatically keeps the same orientation.
- Press *turn right* button to turn the AR Drone automatically to the right. Otherwise the AR Drone automatically keeps the same orientation.
- Push *up* button to go upward automatically at a predetermined speed. Otherwise the AR Drone automatically stays at the same altitude.
- Push *down* to go downward automatically at a predetermined speed. Otherwise the AR Drone automatically stays at the same altitude.

A number of flight control parameters can be tuned:

- altitude limit
- yaw speed limit
- vertical speed limit
- AR.Drone 2.0 tilt angle limit
- host tilt angle limit

2.7 Advanced manoeuvres using host tilt sensors

Many hosts now include tilt motion sensors. Their output values can be sent to the AR.Drone 2.0 as the AR.Drone 2.0 tilting commands.

One *tilting* button on the host activates the sending of tilt sensor values to the AR.Drone 2.0 . Otherwise hovering is a default command when the user does not input any manoeuvre command. This dramatically simplifies the AR.Drone 2.0 control by the user.

The host tilt angle limit and trim parameters can be tuned.

10

2.8 Video streaming, tags and roundel detection

The frontal camera is a CMOS sensor with a 90 degrees angle lens.

The AR.Drone automatically encodes and streams the incoming images to the host device.

The AR.Drone 1.0 use QCIF (176x144, bottom facing camera) or QVGA (320x240, front facing camera) image resolutions. The video stream frame rate is set to 15 FPS.

The AR.Drone 2.0 use 360p (640x360) or 720p (1280x720) image resolutions for both cameras (with upscaling from bottom facing camera). The video stream frame rate can be adjusted between 15 and 30 FPS.

The AR.Drone provides detection of three different tags types shown below.

After update of the AR.Drone 1.0 , cross detections with AR.Drone 2.0 is possible.

Users can download a printable version of the Oriented Roundel on [Parrot website](#).

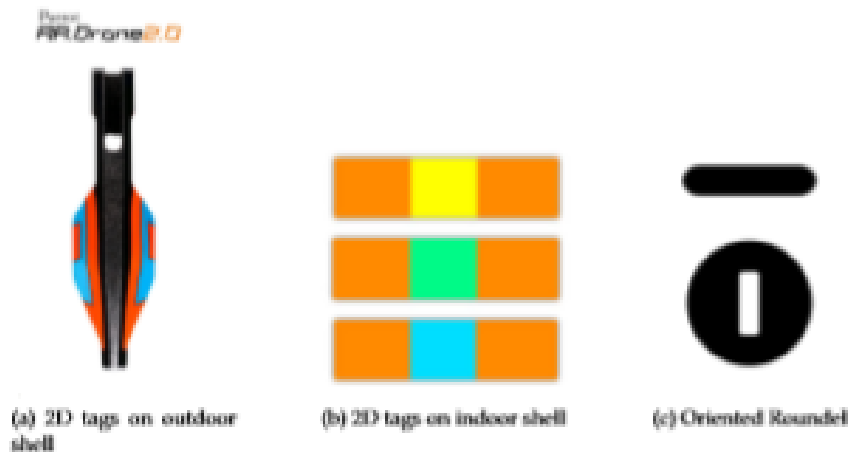


Figure 2.4: Detection tags

2.9 Wifi network and connection

The AR.Drone 2.0 can be controlled from any client device supporting Wifi. The following process is followed :

1. the AR.Drone creates a WIFI network with an ESSID usually called *ardrone2_xxx* (*ardrone_xxx* for AR.Drone 1.0) and self allocates a free, odd IP address (typically 192.168.1.1).
2. the user connects the client device to this ESSID network.
3. the client device requests an IP address from the drone DHCP server.
4. the AR.Drone DHCP server grants the client with an IP address which is :
 - the drone own IP address plus 1 (for AR.Drone 1.0 prior to version 1.1.3)

- the drone own IP address plus a number between 1 and 4 (for AR.Drone 2.0 and AR.Drone 1.0 after 1.1.3 version)
5. the client device can start sending requests the AR.Drone IP address and its services ports.

2.10 Communication services between the AR.Drone 2.0 and a client device

Controlling the AR.Drone is done through 3 main communication services.

Controlling and configuring the drone is done by sending *AT commands* on UDP port 5556. The transmission latency of the control commands is critical to the user experience. Those commands are to be sent on a regular basis (usually 30 times per second). The list of available commands and their syntax is discussed in chapter 6.

Information about the drone (like its status, its position, speed, engine rotation speed, etc.), called *navdata*, are sent by the drone to its client on UDP port 5554. These *navdata* also include tags detection information that can be used to create augmented reality games. They are sent approximatively 15 times per second in demo mode, and 200 times per second in full (debug) mode.

A video stream is sent by the AR.Drone to the client device on port 5555 (UDP for AR.Drone 1.0 , TCP for AR.Drone 2.0). Images from this video stream can be decoded using the codec included in this SDK. Its encoding format is discussed in section 7.2.

A fourth communication channel, called *control port*, can be established on TCP port 5559 to transfer critical data, by opposition to the other data that can be lost with no dangerous effect. It is used to retrieve configuration data, and to acknowledge important information such as the sending of configuration information.

2.11 Differences between AR.Drone 2.0 and AR.Drone 1.0

Please note that this list is not an exhaustive list, but rather a reminder for developpers that want to support both generations of AR.Drone

Sensors

AR.Drone 2.0 includes new hardware sensors : a 3 axis magnetometer, and a pressure sensor. Other sensors were also changed, as the navigation boards are not the same. These magnetometer is mandatory for the Absolute Control feature. The pressure sensor allows the AR.Drone 2.0 to know its height regardless of the ultrasound performance (after 6 meters, the ultrasound can't measure the height)

12

Cameras

AR.Drone 2.0 use a HD (720p-30fps) front facing camera. This camera can be configured to stream both 360p (640*360) or 720p (1280*720) images.

AR.Drone 1.0 use a VGA (640*480) camera, which can only stream QVGA (320*240) pictures. Full resolution pictures are only available to detection algorithms, and photo shooting.

AR.Drone 2.0 bottom facing camera is a QVGA (320*240) 60fps camera. This camera pictures will be upscaled to 360p or 720p for video streaming.

AR.Drone 1.0 use a QCIF (176*144) 60fps camera, which is streamed at full resolution.

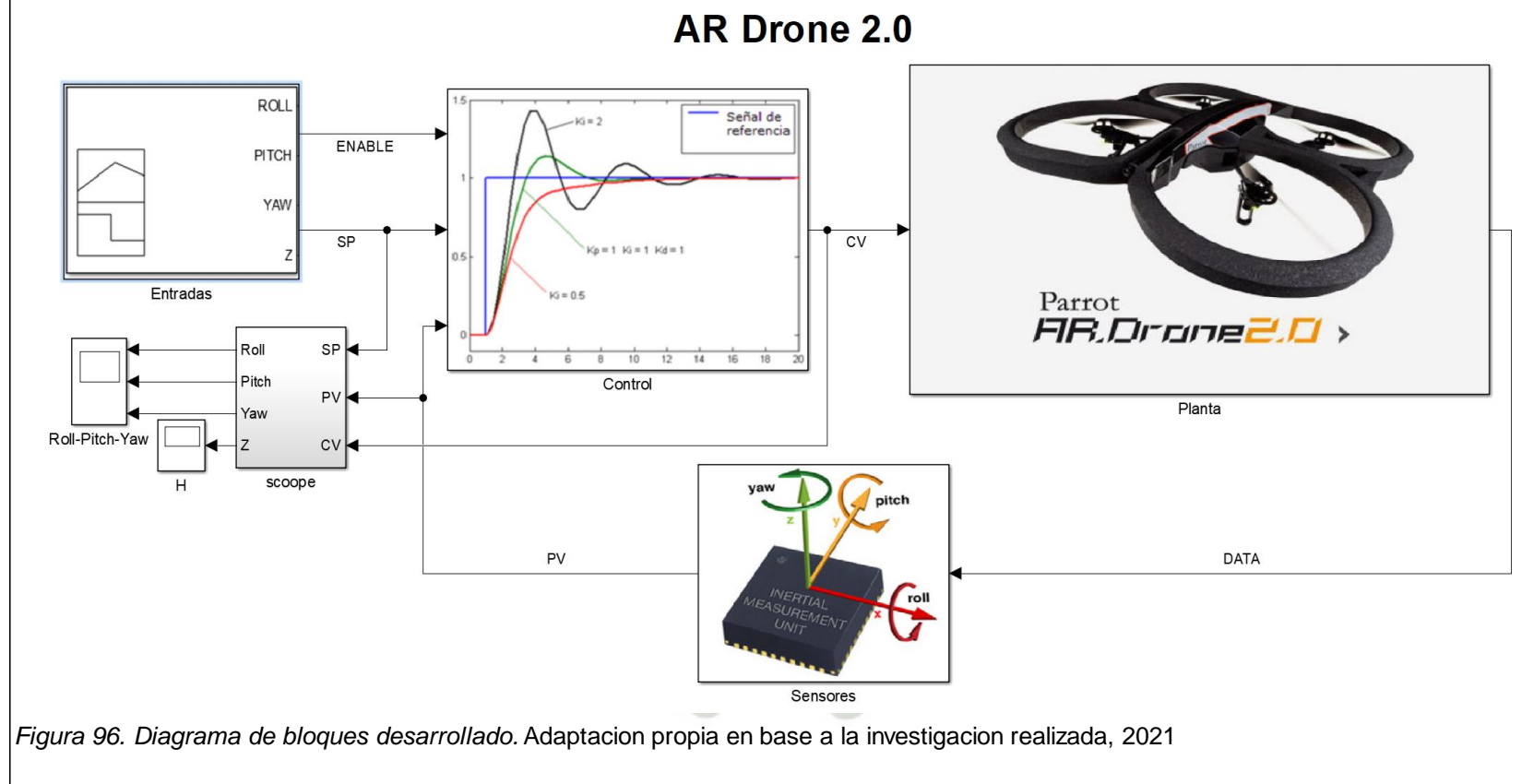
USB port

AR.Drone 2.0 has a master USB port, with a standard USB-A connector. This USB port is currently used for USB Key video recording.

Please note that the AR.Drone 2.0 only supports USB keys with a grounded USB connector casing, and formatted in FAT32 file format.

Anexo 05: SIMULINK

Diagrama de bloques



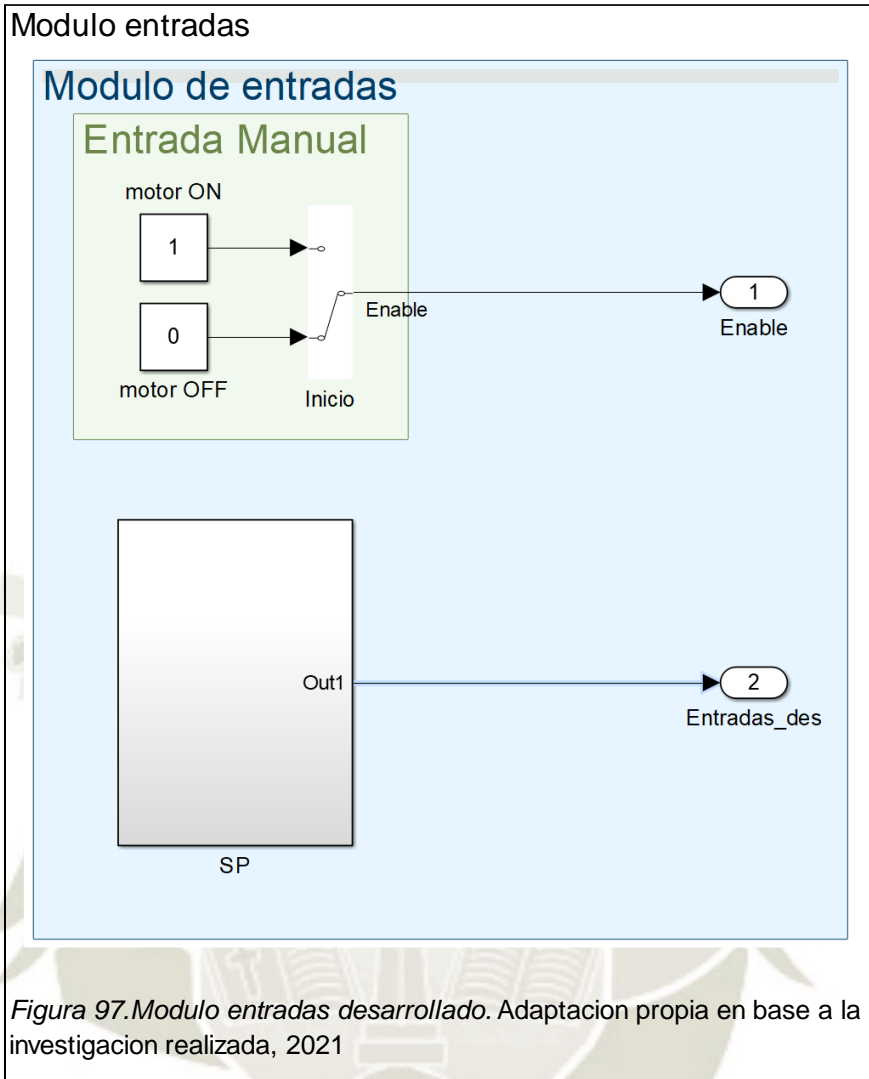


Figura 97. Modulo entradas desarrollado. Adaptacion propia en base a la investigacion realizada, 2021

Figura 97. Modulo entradas desarrollado

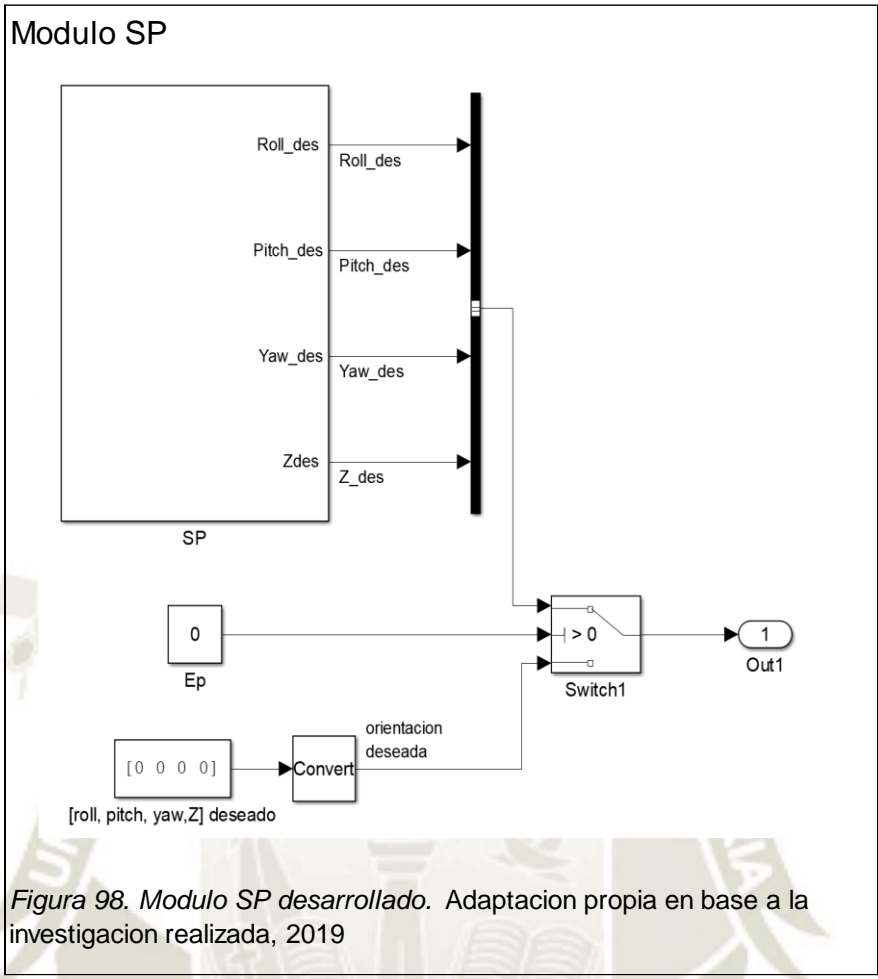


Figura 98 Modulo SP

Modulo SP Establecido

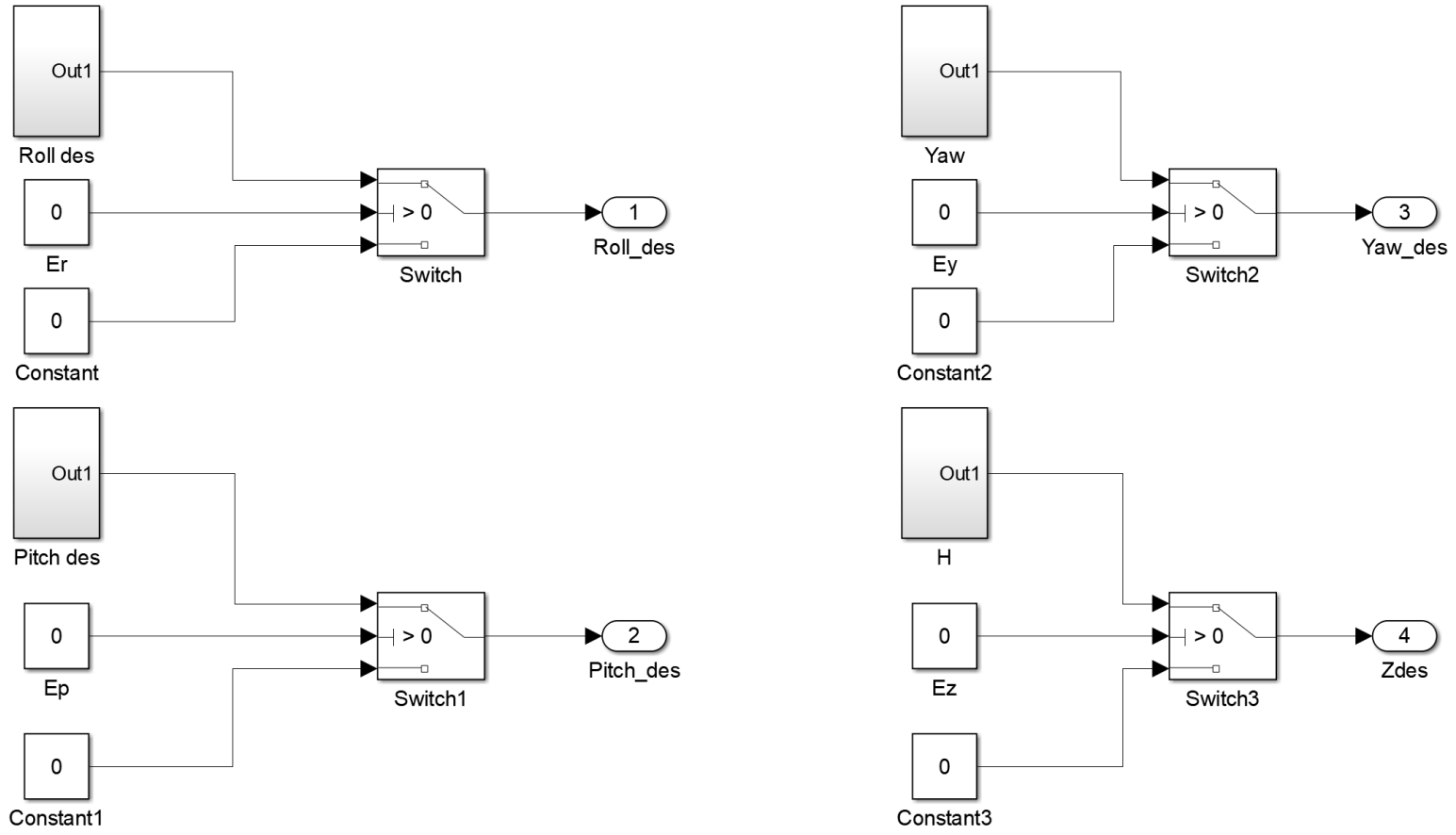


Figura 99. Modulo SP Establecido desarrollado . Adaptacion propia en base a la investigacion realizada, 2021

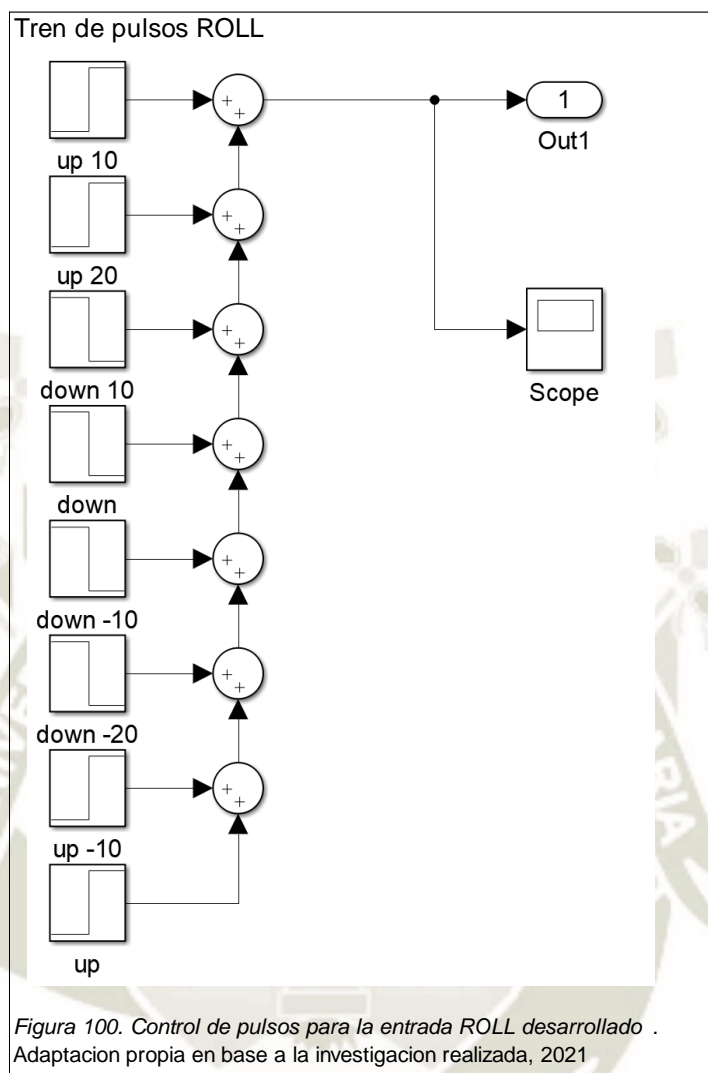
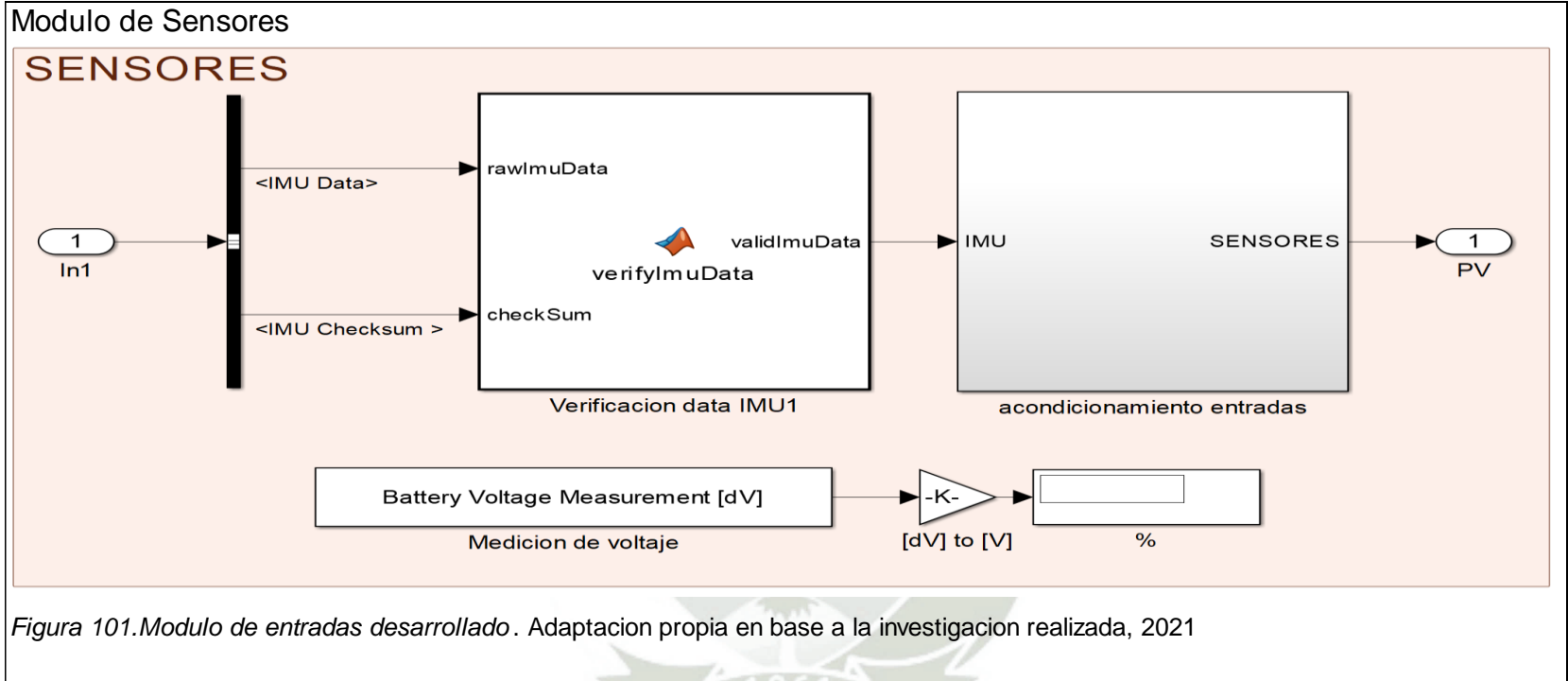


Figura 100. Tren de pulsos ROLL



Acondicionamiento de entrada

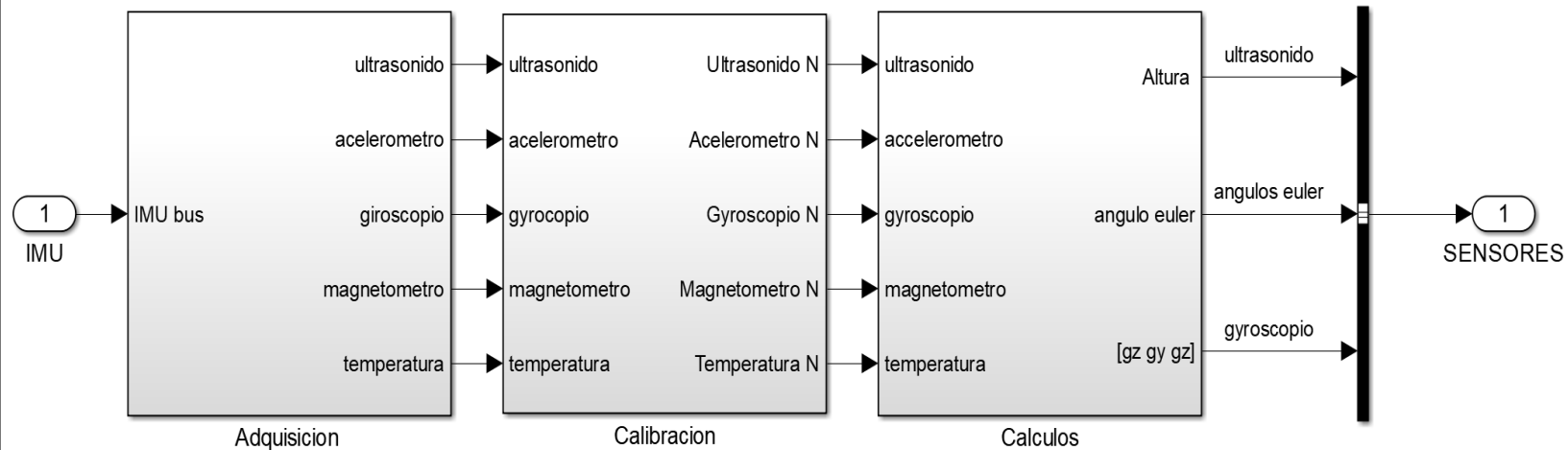


Figura 102. Acondicionamiento de entrada desarrollado . Adaptacion propia en base a la investigacion realizada, 2021

Adquisicion de sensores

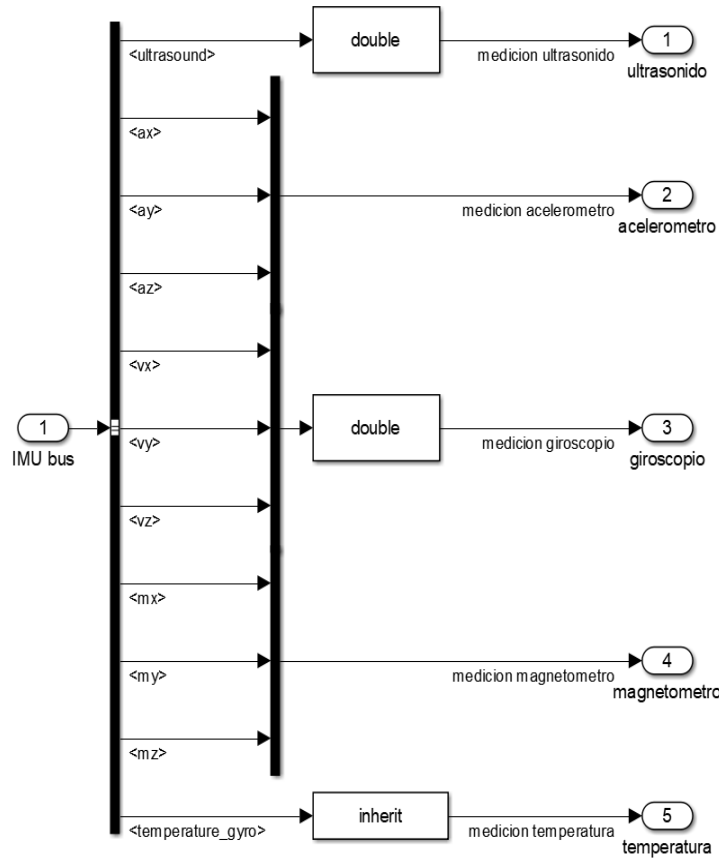
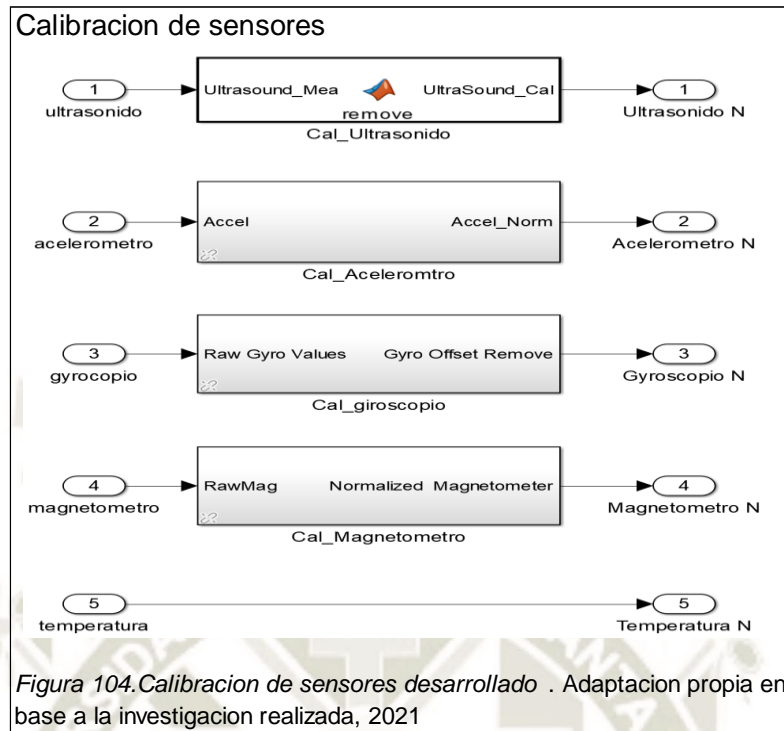


Figura 103. Adquisicion de sensores desarrollado. Adaptacion propia en base a la investigacion realizada, 2021

Figura 103. Adquisición de sensores



Bloque de calibración ultrasonido

CAL_ULTRASONIDO

```
function UltraSound_Cal = remove(Ultrasound_Mea)
```

```
 %#codegen
```

```
peakThreshold = 10000;
```

```
persistent validHeight
```

```
if isempty(validHeight)
```

```
    validHeight = 0;
```

```
end
```

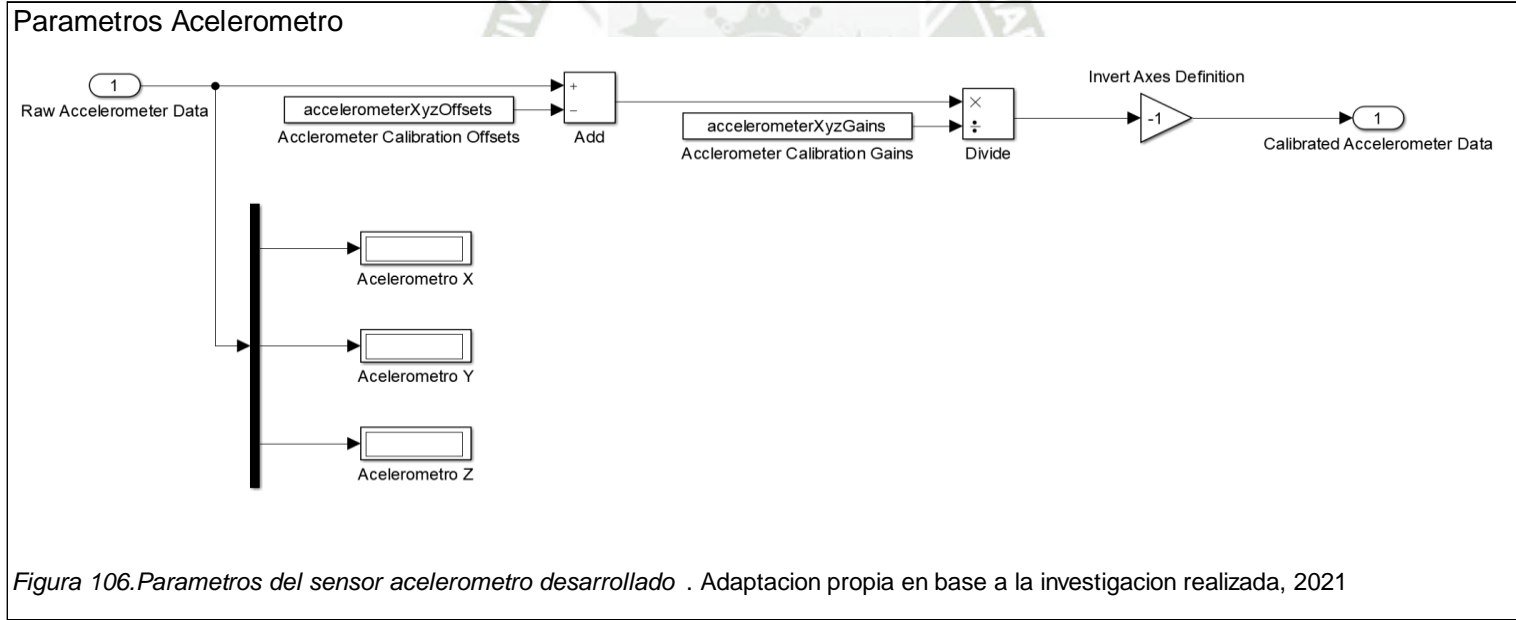
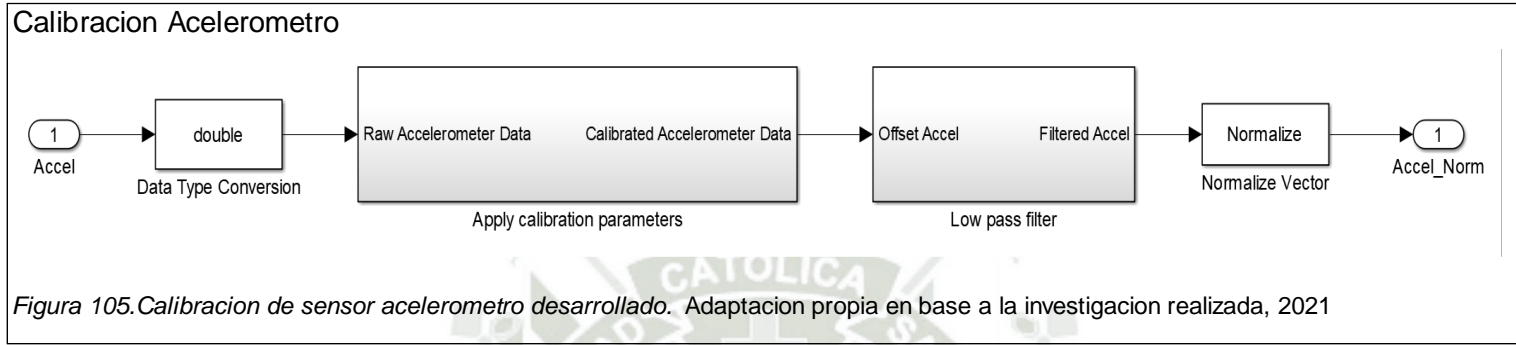
```
if Ultrasound_Mea < peakThreshold
```

```
    validHeight = Ultrasound_Mea*0.0277-0.5299;
```

```
    % validHeight = (Ultrasound_Mea - 880) / 26.533;
```

```
end
```

```
UltraSound_Cal = validHeight;
```



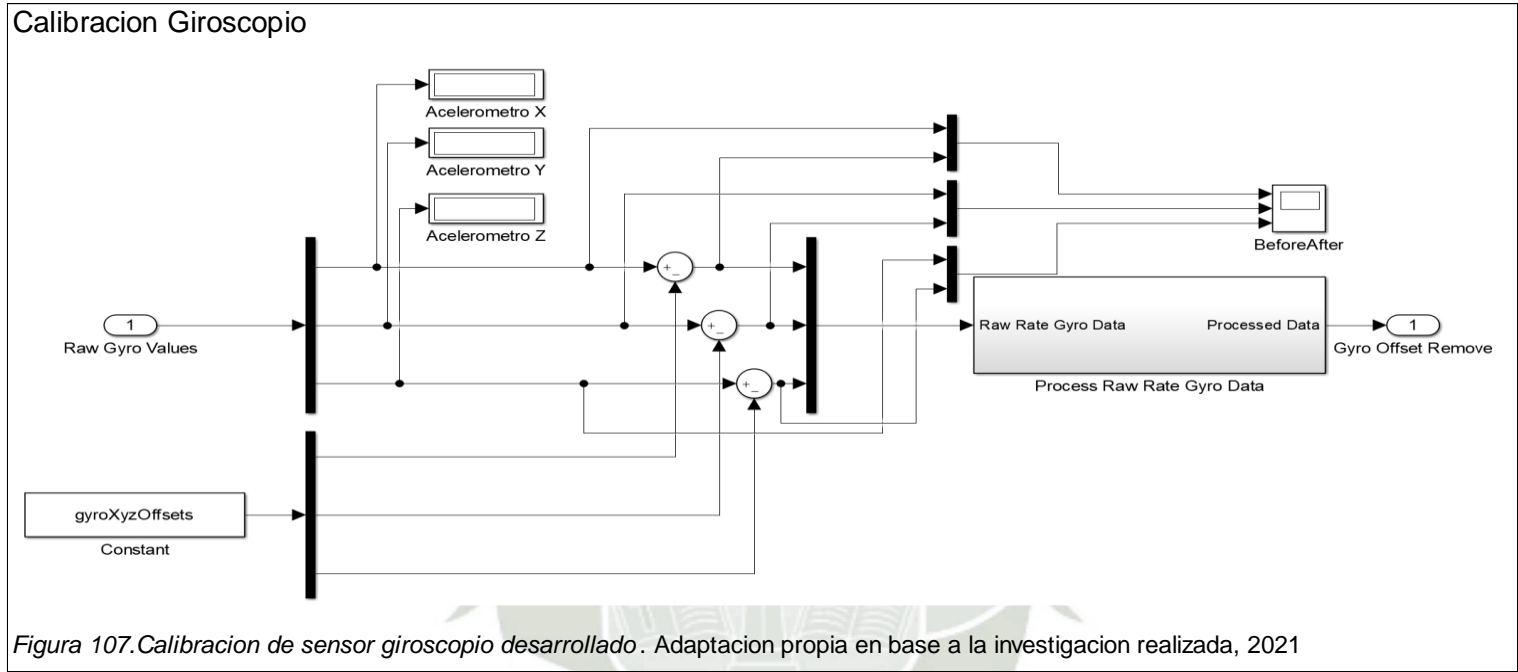


Figura 107. Calibracion de sensor giroscopio desarrollado. Adaptacion propia en base a la investigacion realizada, 2021

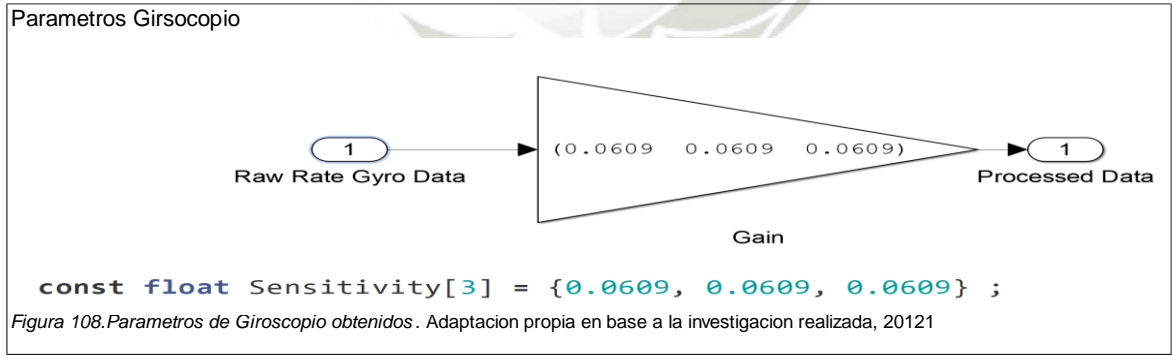


Figura 108. Parametros de Giroscopio obtenidos. Adaptacion propia en base a la investigacion realizada, 20121

Modulo calculos

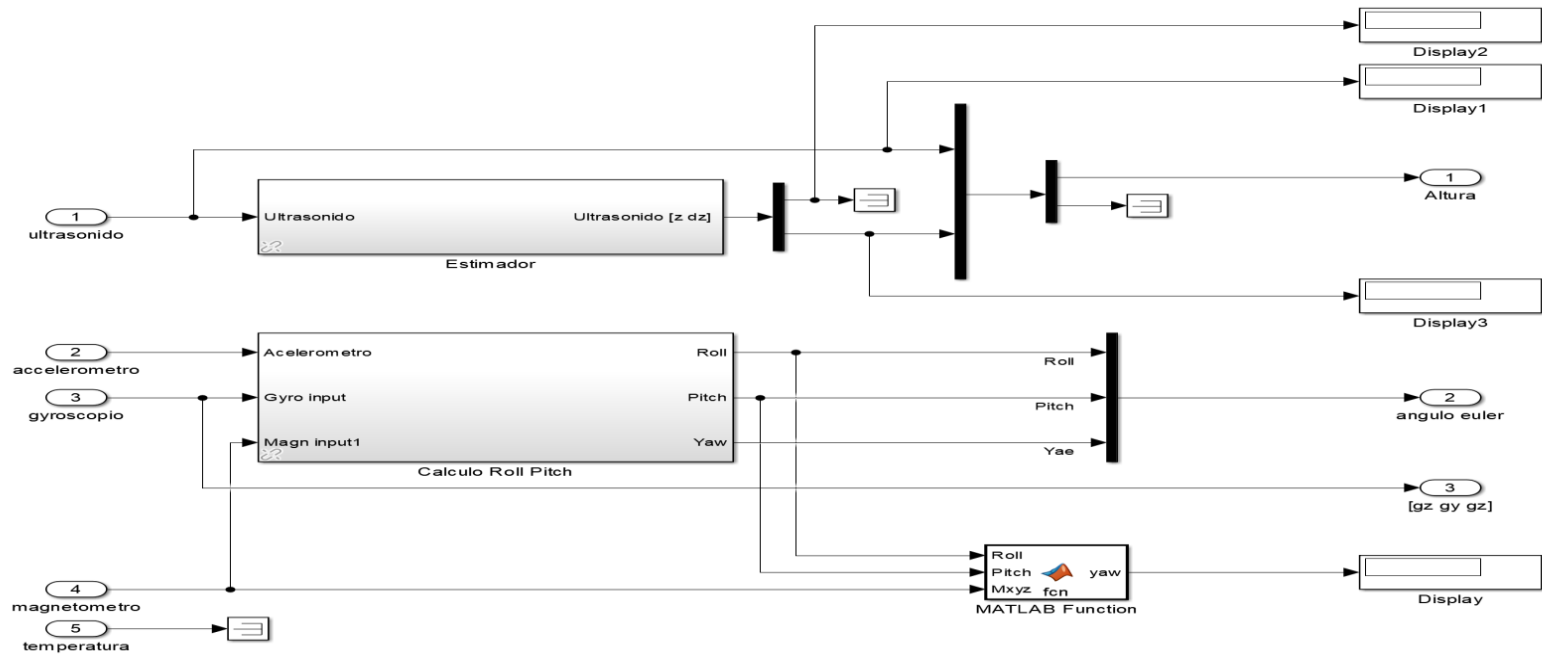
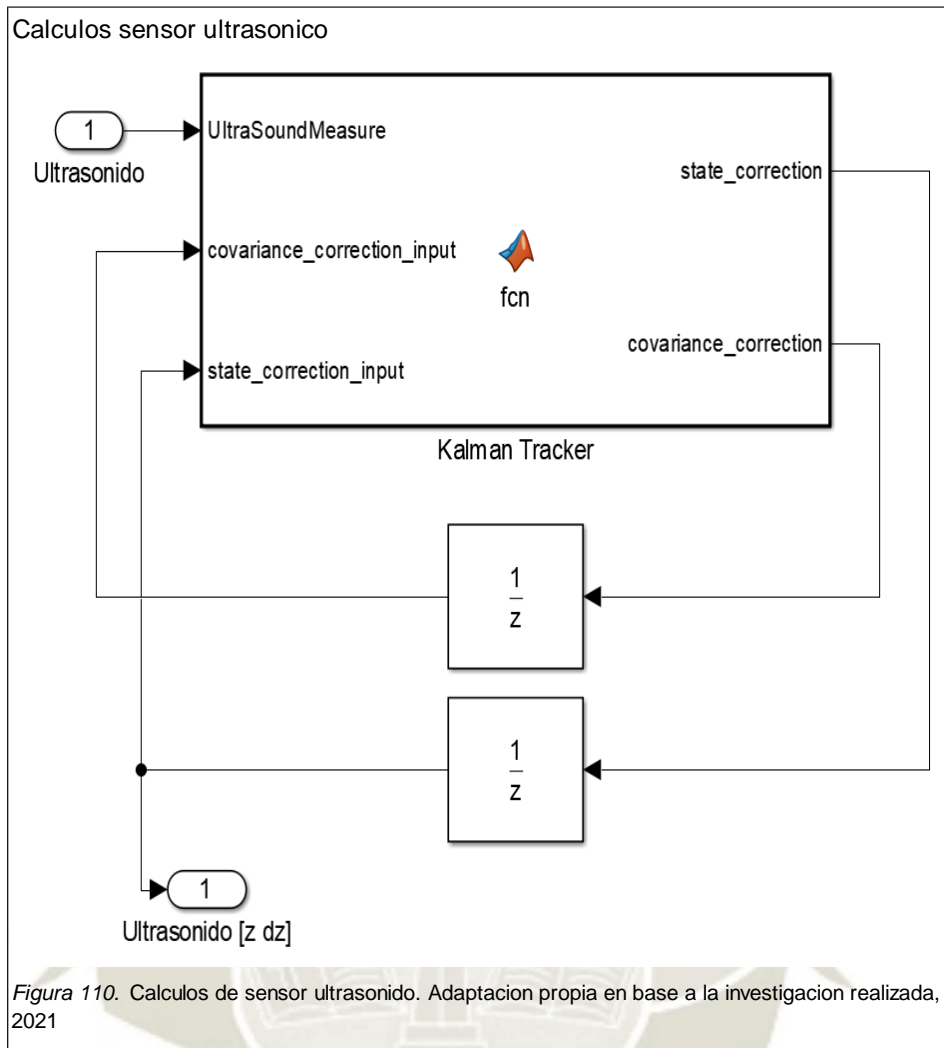


Figura 109. Modulo calculos desarrollado. Adaptacion propia en base a la investigacion realizada, 2021



Filtro Kalman

```
function [state_correction, covariance_correction] = fcn(UltraSoundMeasure,
covariance_correction_input,state_correction_input)
```

```
%-----
%-----
%---- Predict Stage -----
%-----
%-----
```

```
%our states:
% [Wx Wy Wz Bx By Bz ]
```

```

%

p_var = 1e-2;
Ts = 1/400;
%Process Noise

Q = [p_var^2    0
      0    p_var^2 ];

%A matrix
A = [1  Ts
      0  1];

%odoutput these values with a unit delay (for algebraic loop), apply initial conditions
state_prediction = A*state_correction_input;
covariance_prediction_ = A*covariance_correction_input*A' + Q;

%-----
%-----
%---- Estimate Stage -----
%-----
%-----

%H matrix
H = [1 0];

%Measurement noise
m_var = 1;
%   m_var_r = 1;

R = [m_var^2];

%Measurement Z
Z = UltraSoundMeasure;

```

```
%Compute Kalman Gain
Kalman_Gain = covariance_prediction_*H'*inv(H*covariance_prediction_*H'+R);
%2x2 matrix

%compute state space correction based on measurement
state_correction = state_prediction + Kalman_Gain*(Z - H*state_prediction); %2x1
vector of states

%compute covariance correction
covariance_correction = (eye(2) - Kalman_Gain*H)*covariance_prediction_; %2x2
covariance matrix

end
```

Calculo angulos de Euler

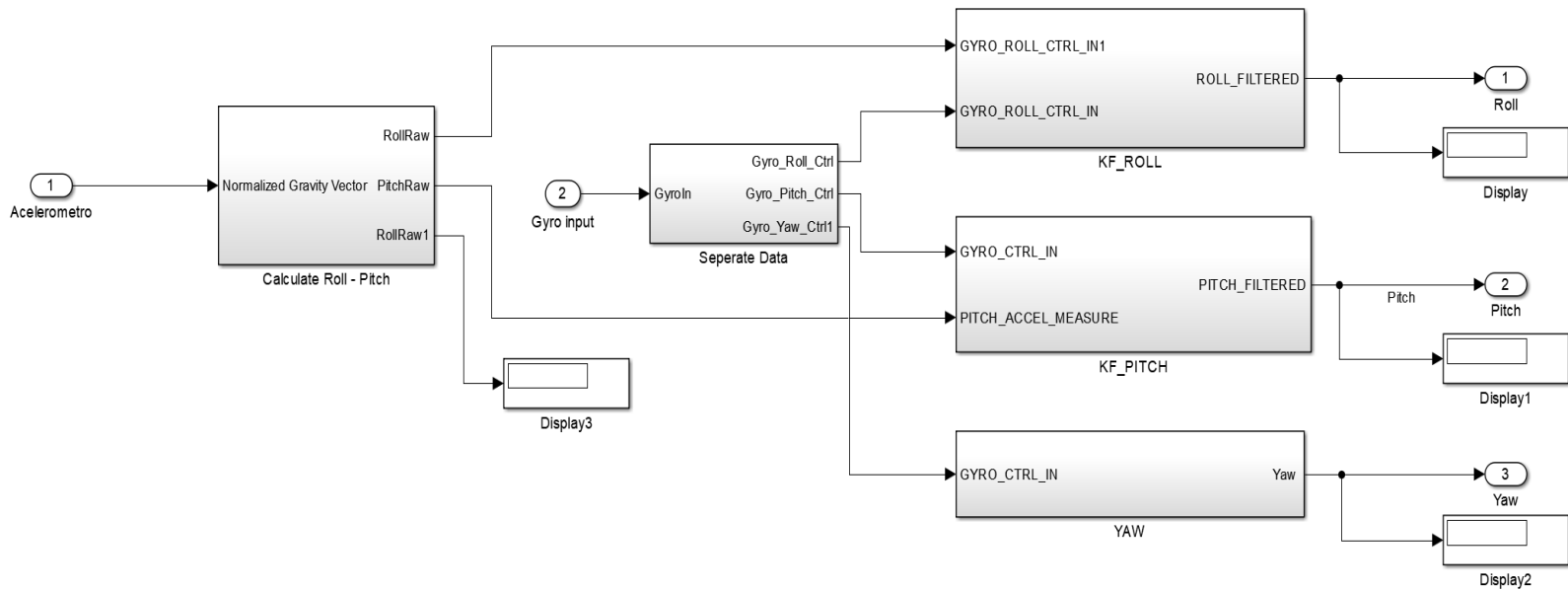
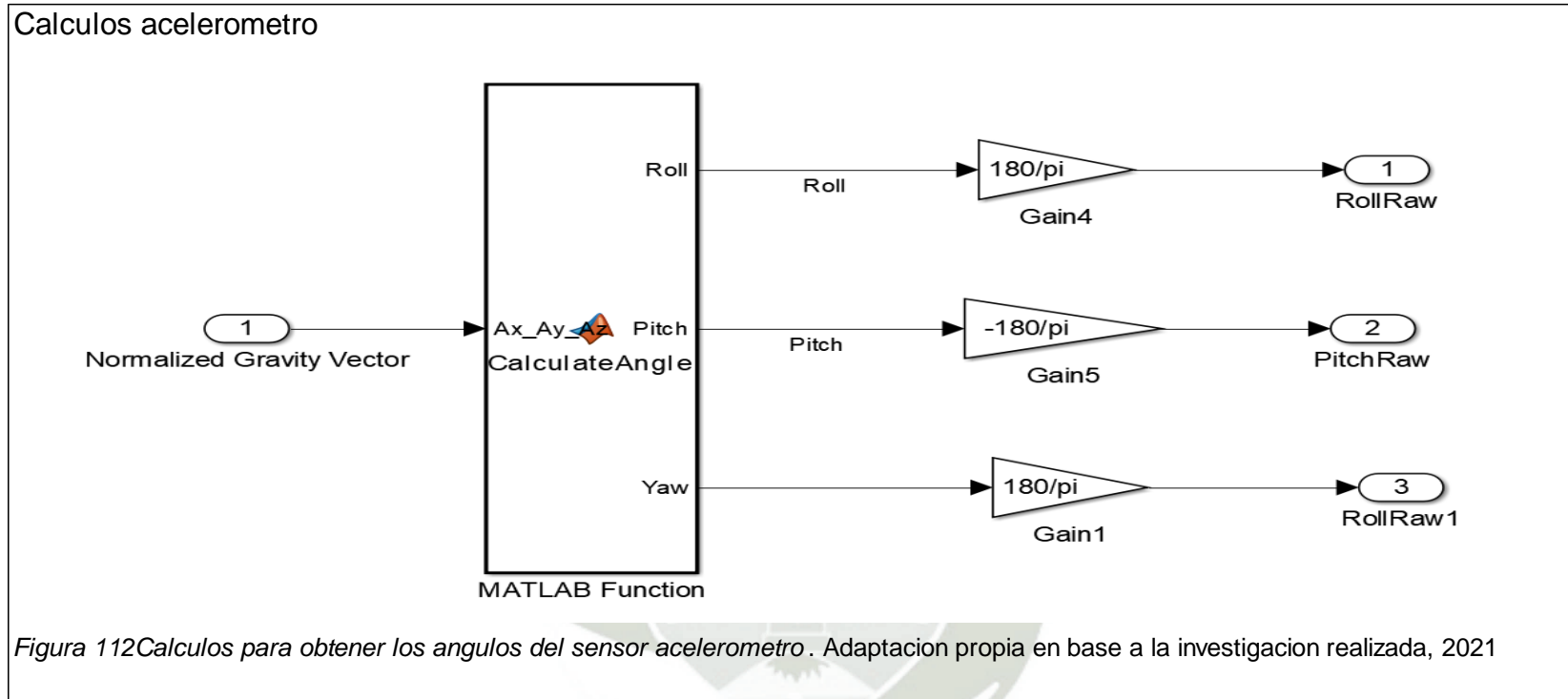


Figura 111. Calculo para obtener los angulos de Euler. Adaptacion propia en base a la investigacion realizada, 2021



Ecuaciones geométricas para obtener ángulos

```
function [Roll,Pitch,Yaw] = CalculateAngle(Ax_Ay_Az)
```

```
%#codegen
```

```
Ax = -Ax_Ay_Az(1);
```

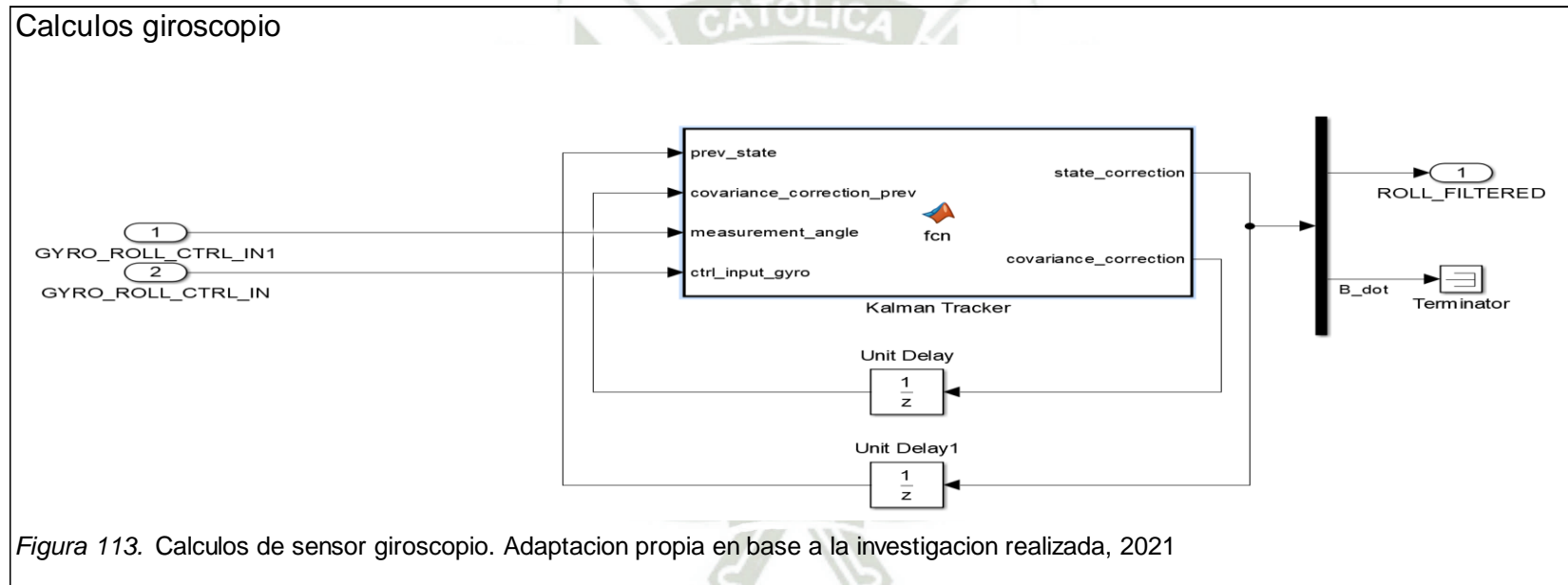
```
Ay = -Ax_Ay_Az(2);
```

```
Az = Ax_Ay_Az(3);
```

$$\text{Pitch} = \text{atan}\left(\frac{A_x}{\sqrt{A_y^2 + A_z^2}}\right);$$

$$\text{Roll} = \text{atan}\left(\frac{A_y}{\sqrt{A_x^2 + A_z^2}}\right);$$

$$\text{Yaw} = \text{atan}\left(\frac{\sqrt{A_x^2 + A_y^2}}{A_z}\right);$$



Filtro Kalman

function [state_correction, covariance_correction] = fcn(prev_state,
covariance_correction_prev, ctrl_input_gyro, measurement_angle)

%-----

%-----

%---- Predict Stage -----

%-----

%-----

% pix_value = 5;

Ts = 1/400;

%Process Noise

p_var = 0.01;

Q = [p_var^2 0
0 p_var^2];

%A matrix

A = [1 -Ts
0 1];

%output these values with a unit delay (for algebraic loop), apply initial conditions

state_prediction = A*prev_state + [Ts 0; 0 0]*[ctrl_input_gyro; 0];

covariance_prediction = A*covariance_correction_prev*A' + Q;

%-----

%-----

%---- Estimate Stage -----

%-----

%-----

%H matrix

H = [1 0];

```
%Measurement noise from accelerometer angle measurement
m_var = 1;
%   m_var_r = 1;

R = [m_var^2 ];

%Measurement Z
Z = [measurement_angle];

%Compute Kalman Gain
Kalman_Gain = covariance_prediction*H'*inv(H*covariance_prediction*H'+R);
%2x2 matrix

%compute state space correction based on measurement
state_correction = state_prediction + Kalman_Gain*(Z - H*state_prediction); %4x1
vector of states

%compute covariance correction
covariance_correction = (eye(2) - Kalman_Gain*H)*covariance_prediction; %2x2
covariance matrix

%   corrected_angle_measure = state_correction(1) - Ts*state_correction(2);
End
```

Calibracion de angulo Yaw giroscopio

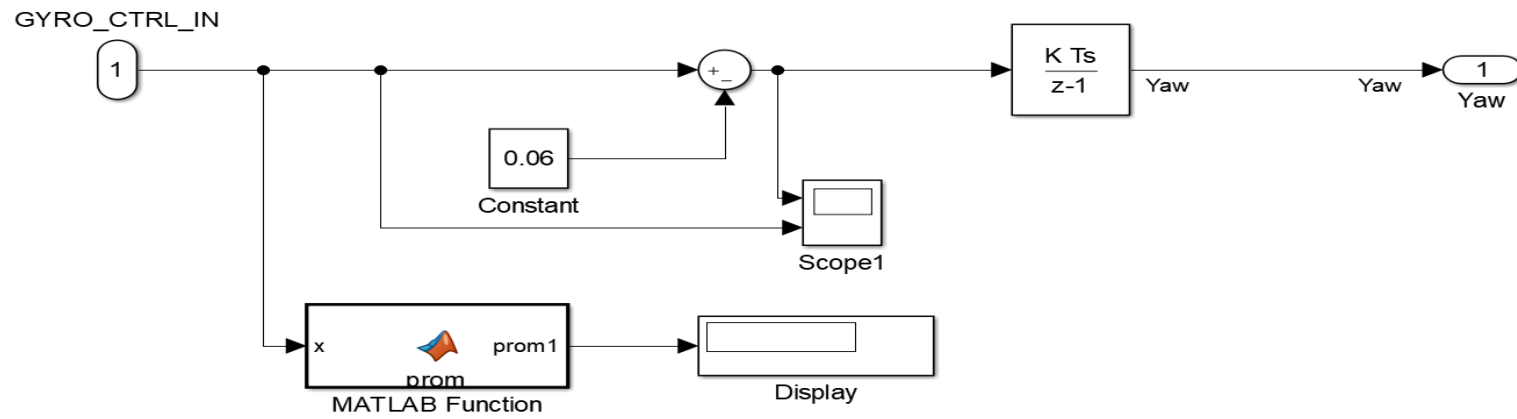
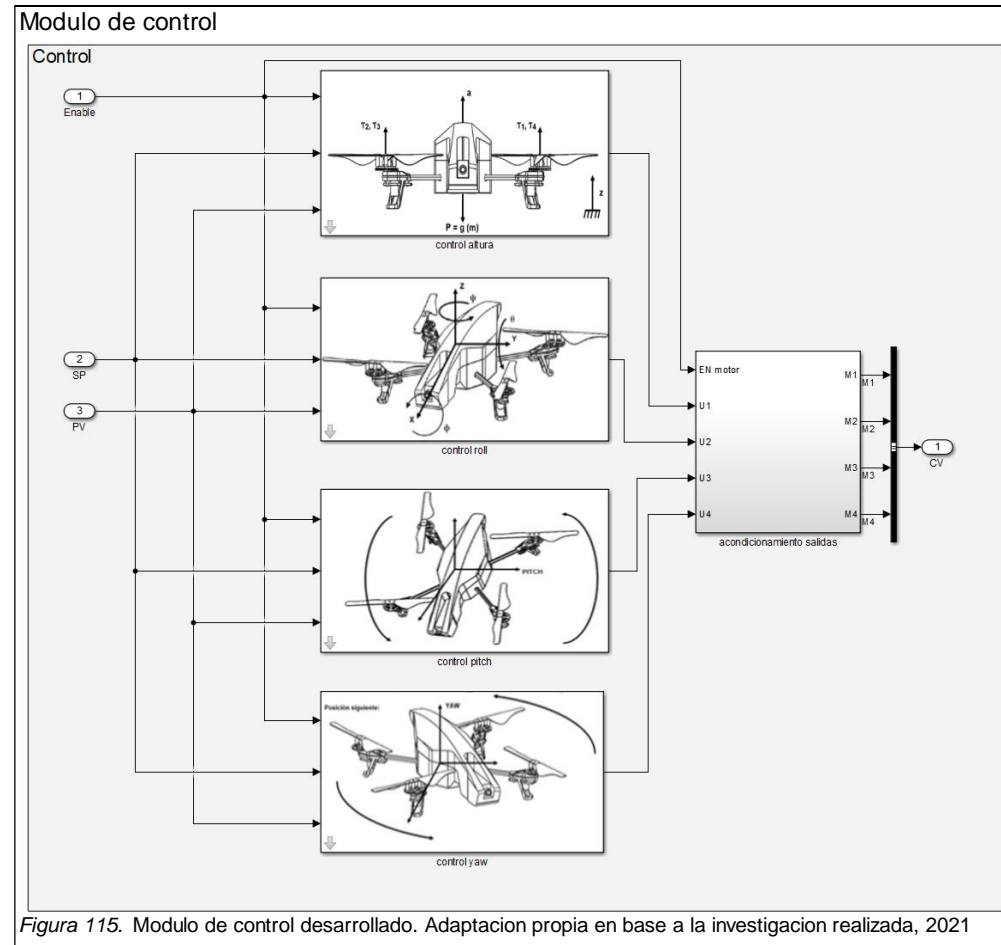


Figura 114. Calibracion de angulo Yaw de Girsocopio. Adaptacion propia en base a la investigacion realizada, 2021

Modulo Control



Bloques de control altura

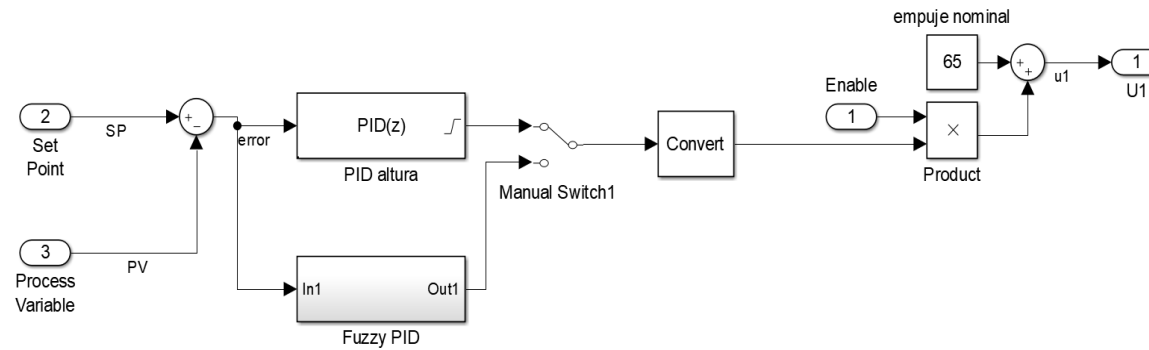


Figura 116. Bloques de control altura desarrollado. Adaptacion propia en base a la investigacion realizada, 2021

Bloque de control Difuso altura

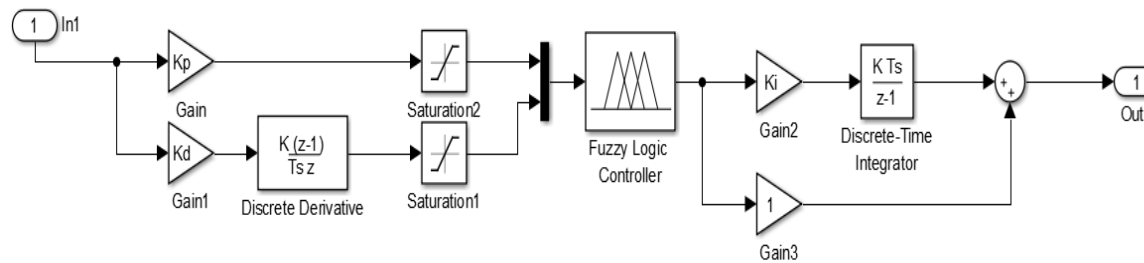
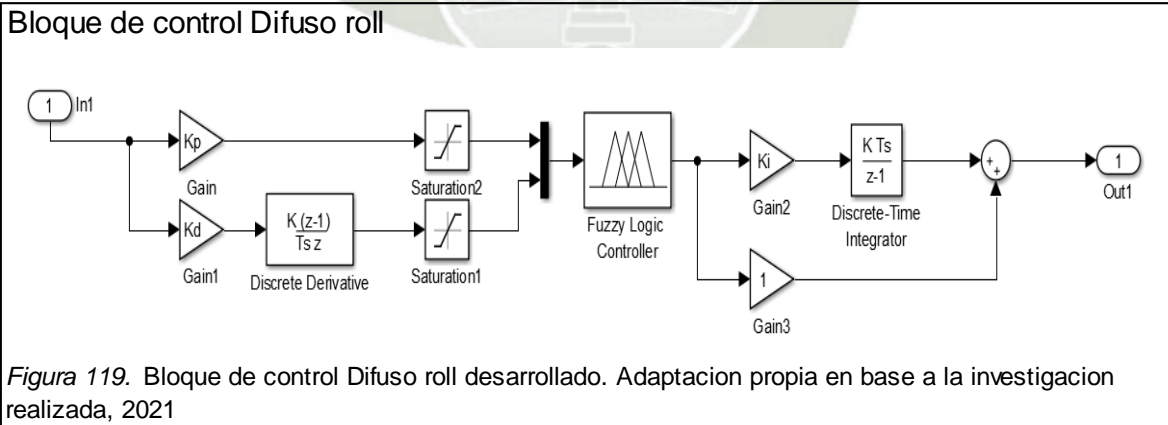
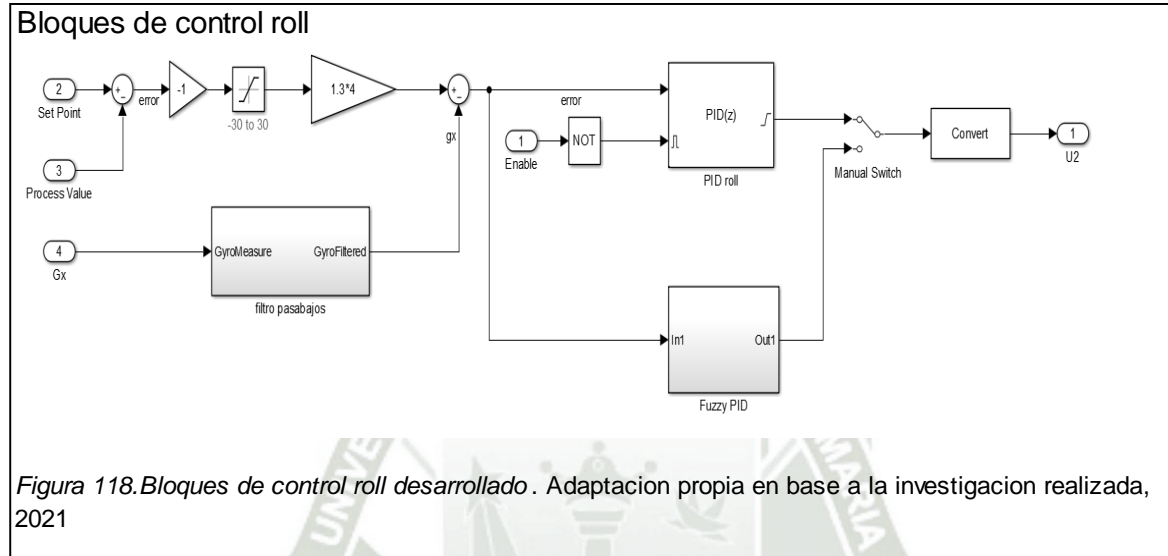
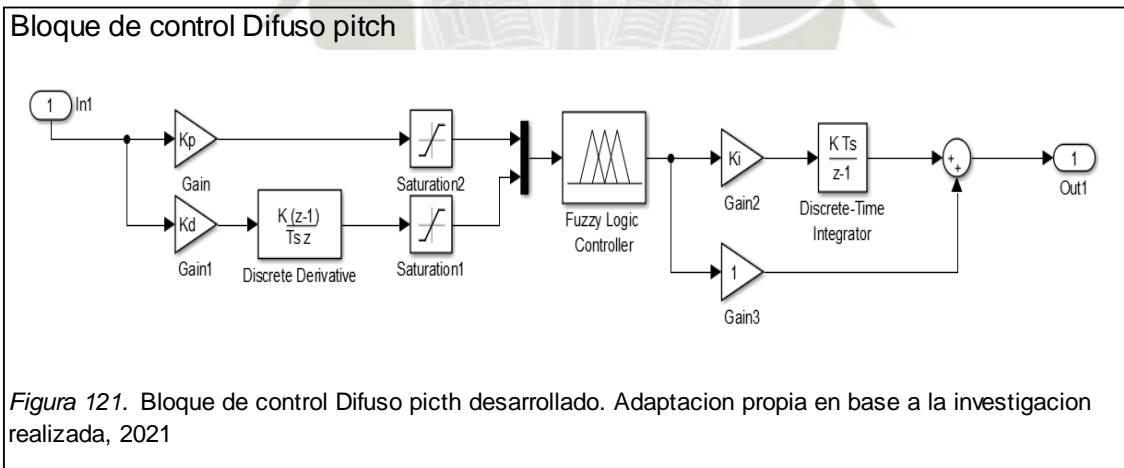
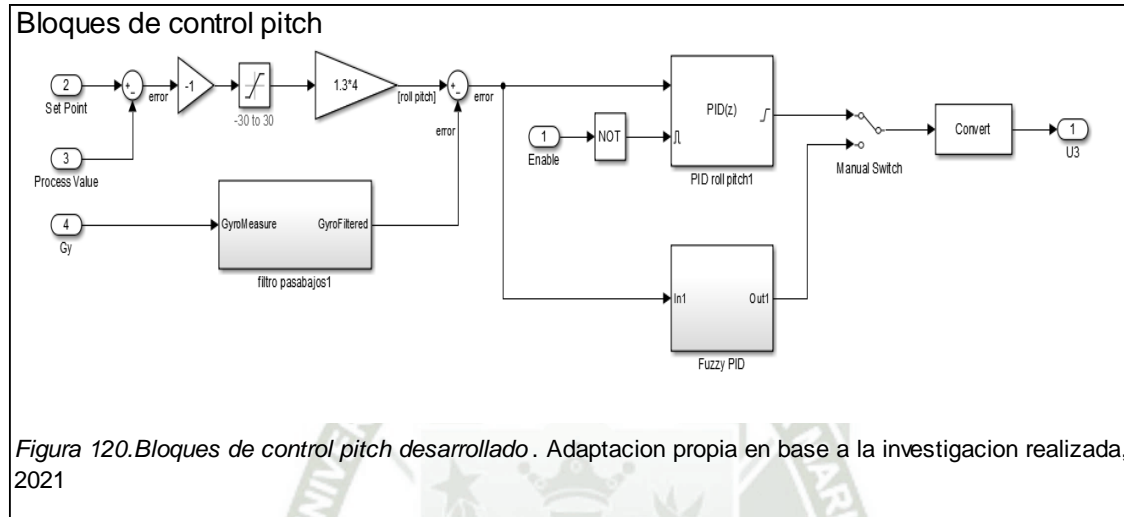


Figura 117. Bloque de control Difuso altura desarrollado. Adaptacion propia en base a la investigacion realizada, 2021





Bloques de control Yaw

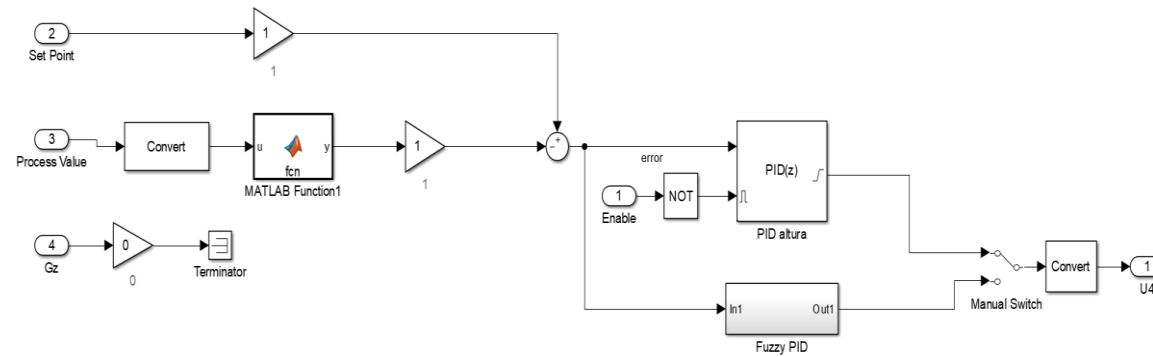


Figura 122. Bloques de control Yaw desarrollado. Adaptacion propia en base a la investigacion realizada, 2021

Bloque de control Difuso Yaw

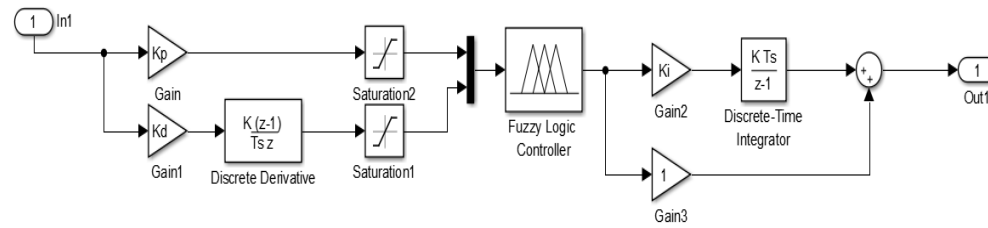


Figura 123. Bloque de control Difuso Yaw desarrollado. Adaptacion propia en base a la investigacion realizada, 2021

Función de límites de ángulos yaw

```
function y = fcn(u)
```

```
    p=rem(u,180);
```

```
    if u<-180
```

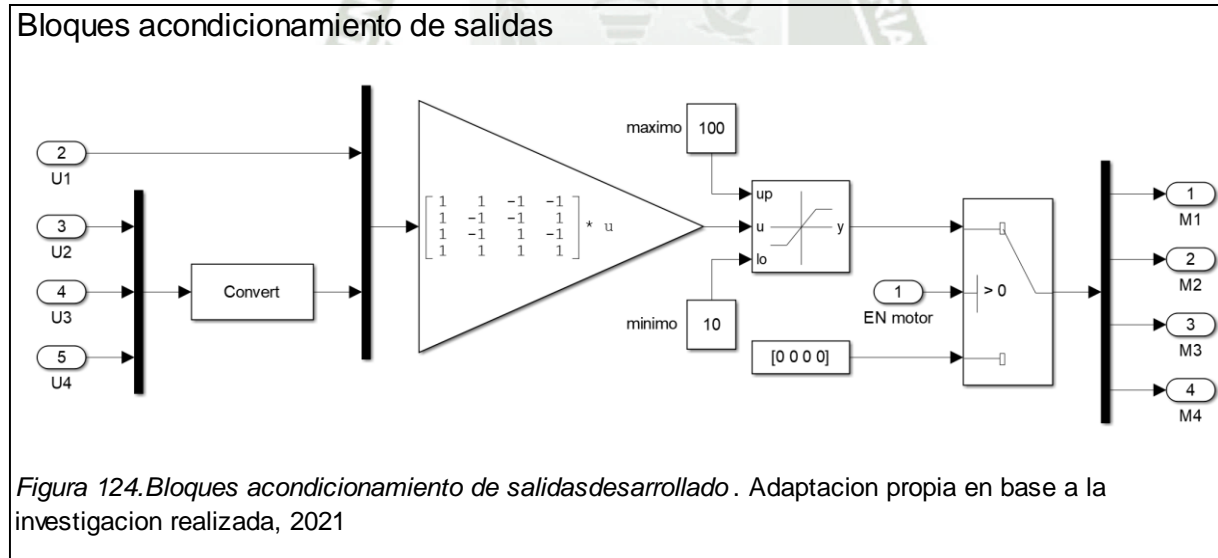
```
        u=180+p;
```

```
    elseif u>180
```

```
        u=-180+p;
```

```
    end
```

```
    y=u;
```



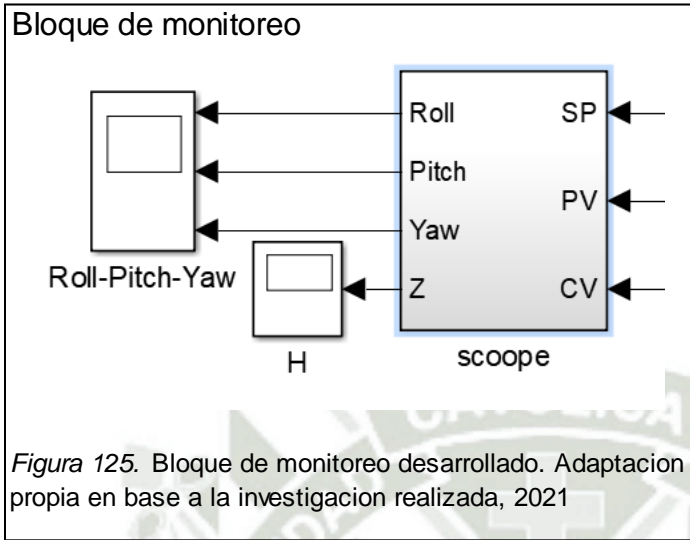
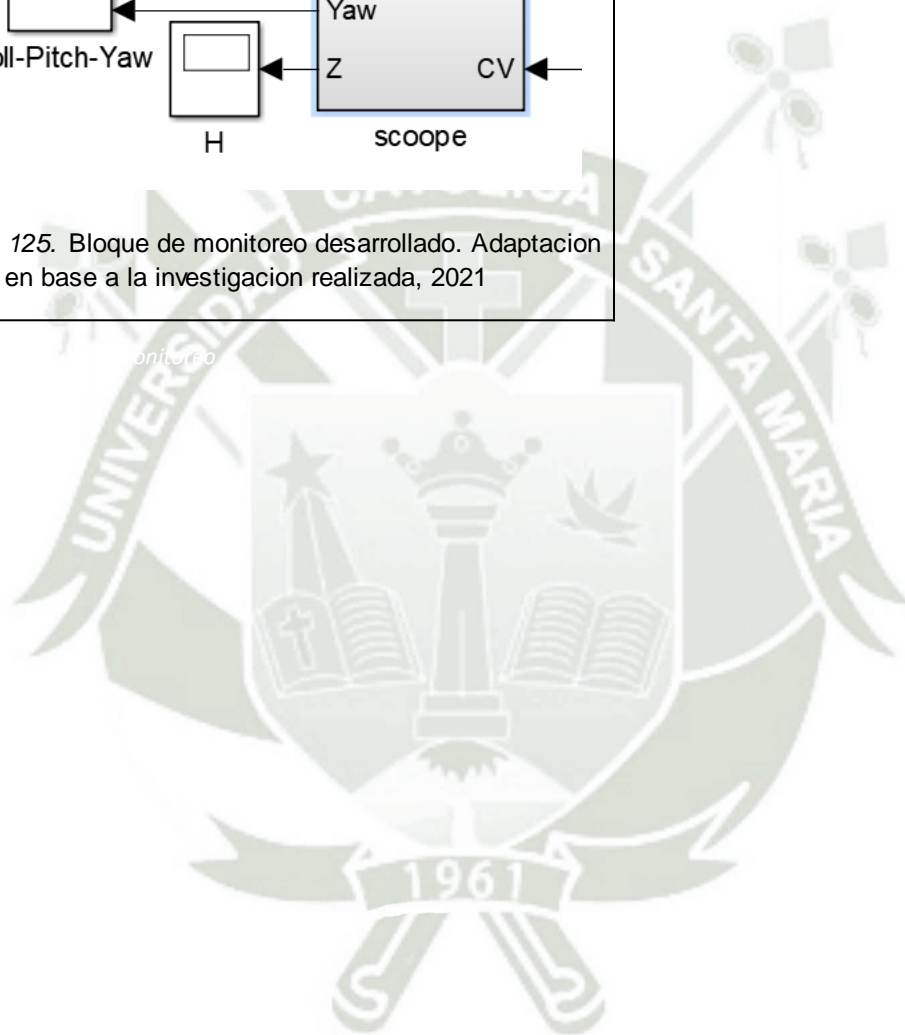
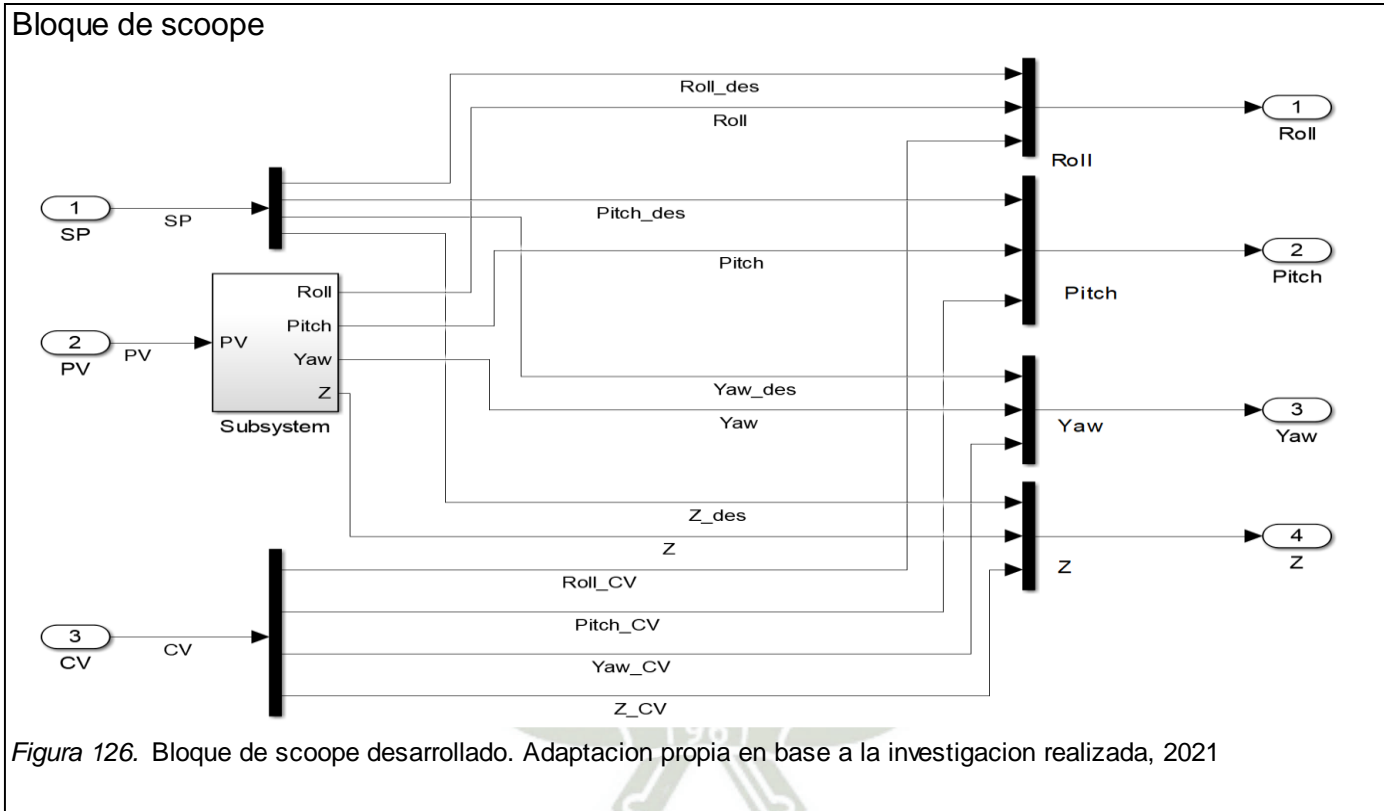


Figura 125. Bloque de monitoreo desarrollado. Adaptacion propia en base a la investigacion realizada, 2021





Anexo 06: Guide

Guide

```
function varargout = GUIDE(varargin)
% GUIDE MATLAB code for GUIDE.fig
%   GUIDE, by itself, creates a new GUIDE or raises the existing
%   singleton*.
%
%   H = GUIDE returns the handle to a new GUIDE or the handle to
%   the existing singleton*.
%
%   GUIDE('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in GUIDE.M with the given input arguments.
%
%   GUIDE('Property','Value',...) creates a new GUIDE or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before GUIDE_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to GUIDE_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUIDE

% Last Modified by GUIDE v2.5 12-Mar-2020 17:23:03

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
```

```
'gui_OpeningFcn', @GUIDE_OpeningFcn, ...
'gui_OutputFcn', @GUIDE_OutputFcn, ...
'gui_LayoutFcn', [], ...
'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if narginout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before GUIDE is made visible.
function GUIDE_OpeningFcn(hObject, eventdata, handles, varargin)
%find_system('Name','PID_2019');
%open_system('PID_2019');
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUIDE (see VARARGIN)

% Choose default command line output for GUIDE
fondo=imread('background.jpg');
axes(handles.axes1);
%axis off;
imshow(fondo);
handles.output = hObject;
```

```
%PID_2019;
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUIDE wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = GUIDE_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in Manual.
function Manual_Callback(hObject, eventdata, handles)
% hObject handle to Manual (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
close(GUIDE);
manual
%set_param('Pid_2019/Entradas/SP/EP','Value','-1');

% --- Executes on button press in Auto.
function Auto_Callback(hObject, eventdata, handles)
% hObject handle to Auto (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
close(GUIDE);
```

```
auto
```

```
%set_param('Pid_2019/Entradas/SP/Ep','Value','1');
```

```
% --- Executes during object creation, after setting all properties.
```

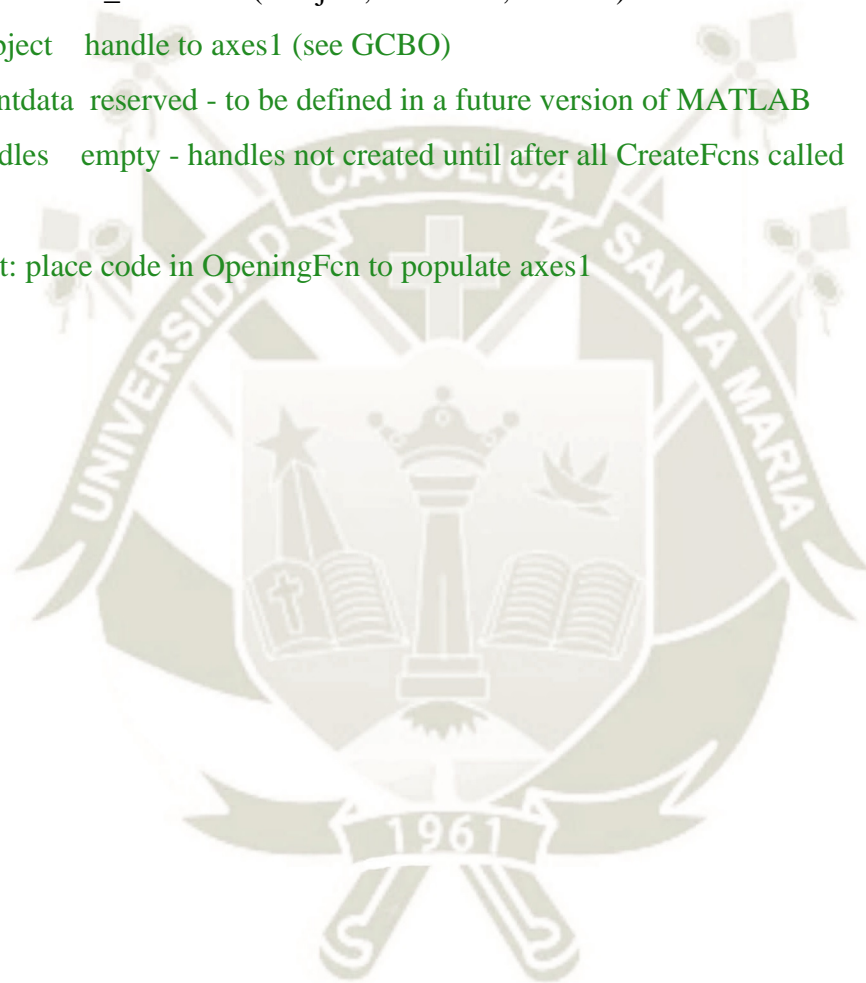
```
function axes1_CreateFcn(hObject, eventdata, handles)
```

```
% hObject handle to axes1 (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles empty - handles not created until after all CreateFcns called
```

```
% Hint: place code in OpeningFcn to populate axes1
```



Automático

```
function varargout = auto(varargin)
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @auto_OpeningFcn, ...
                  'gui_OutputFcn', @auto_OutputFcn, ...
                  'gui_LayoutFcn', [], ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before auto is made visible.

function auto_OpeningFcn(hObject, ~, handles, varargin)
fondo=imread('background.jpg');
axes(handles.axes1);
%axis off;
imshow(fondo);
handles.output = hObject;
handles.modelo = 'KAZU_v01';
guidata(hObject, handles);
function varargout = auto_OutputFcn(~, ~, handles)
varargout{1} = handles.output;
```

```
% --- Executes on button press in build_b.  
function build_b_Callback(~, ~, handles)  
modelo=handles.modelo;  
if bdIsLoaded(modelo)==0  
    open(modelo);  
end  
rtwbuild(modelo); % default behaviour is to run the program  
set_param(modelo,'SimulationCommand','connect'); % so connect to the drone  
%set_param(modelo,'SimulationCommand','start'); % and stop the program  
  
% --- Executes on button press in start_b.  
function start_b_Callback(~, eventdata, handles)  
modelo=handles.modelo;  
set_param(modelo,'SimulationCommand','start');  
  
% --- Executes on button press in stop_b.  
function stop_b_Callback(~, ~, handles)  
modelo=handles.modelo;  
set_param(modelo,'SimulationCommand','stop');  
  
% --- Executes on button press in Hover.  
function Hover_Callback(hObject, eventdata, handles)  
% hObject    handle to Hover (see GCBO)  
% eventdata reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
fondo=imread('ALTURA.jpg');  
axes(handles.axes1);
```

```
%axis off;
imshow(fondo)

% --- Executes on button press in EX.
set_param('Pid_2019/Entradas/SP/Er','Value','0');
set_param('Pid_2019/Entradas/SP/Ep','Value','0');
set_param('Pid_2019/Entradas/SP/Ey','Value','0');
set_param('Pid_2019/Entradas/SP/Ez','Value','1');
set_param('Pid_2019/Control/control altura/SWH','sw','1');

function EX_Callback(hObject, eventdata, handles)
% hObject handle to EX (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton17.

fondo=imread('ROLL.jpg');
axes(handles.axes1);
%axis off;
imshow(fondo)

% --- Executes on button press in pushbutton18.
set_param('Pid_2019/Entradas/SP/Er','Value','1');
set_param('Pid_2019/Entradas/SP/Ep','Value','0');
set_param('Pid_2019/Entradas/SP/Ey','Value','0');
set_param('Pid_2019/Entradas/SP/Ez','Value','0');
set_param('Pid_2019/Control/control altura/SWH','sw','0');
```

```
function EY_Callback(hObject, eventdata, handles)
% hObject handle to EY (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
fondo=imread('PITCH.jpg');
axes(handles.axes1);
%axis off;
imshow(fondo)
% --- Executes on button press in EZ.
set_param('Pid_2019/Entradas/SP/Er','Value','0');
set_param('Pid_2019/Entradas/SP/Ep','Value','1');
set_param('Pid_2019/Entradas/SP/Ey','Value','0');
set_param('Pid_2019/Entradas/SP/Ez','Value','0');
set_param('Pid_2019/Control/control altura/SWH','sw','0');
```

```
function EZ_Callback(~, eventdata, handles)
% hObject handle to EZ (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
fondo=imread('YAW.jpg');
axes(handles.axes1);
%axis off;
imshow(fondo)
```

```
% --- Executes on button press in pushbutton23.
```

```
set_param('Pid_2019/Entradas/SP/Er','Value','0');
set_param('Pid_2019/Entradas/SP/Ep','Value','0');
set_param('Pid_2019/Entradas/SP/Ey','Value','1');
set_param('Pid_2019/Entradas/SP/Ez','Value','0');
```

```
set_param('Pid_2019/Control/control altura/SWH','sw','0');
```

```
function Regresar_Callback(~, ~, ~)
```

```
% hObject handle to Regresar (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
close(auto);
```

```
GUIDE
```

```
% --- Executes on selection change in menu.
```

```
function menu_Callback(hObject, eventdata, handles)
```

```
control =get(handles.menu,'Value');
```

```
switch control
```

```
case 1
```

```
set_param('Pid_2019/Control/control altura/Controlador/P-F_Z','sw','1');
```

```
set_param('Pid_2019/Control/control_roll/Controlador/P-F_R','sw','1');
```

```
set_param('Pid_2019/Control/control_pitch/Controlador/P-F_P','sw','1');
```

```
set_param('Pid_2019/Control/control_yaw/Controlador/P-F_Y','sw','1');
```

```
case 2
```

```
set_param('Pid_2019/Control/control altura/Controlador/P-F_Z','sw','0');
```

```
set_param('Pid_2019/Control/control_roll/Controlador/P-F_R','sw','0');
```

```
set_param('Pid_2019/Control/control_pitch/Controlador/P-F_P','sw','0');
```

```
set_param('Pid_2019/Control/control_yaw/Controlador/P-F_Y','sw','0');
```

```
end
```

```
guidata(hObject, handles);
```

```
function menu_CreateFcn(hObject, eventdata, handles)
```

```
% hObject handle to menu (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.
```



Manual

```
function varargout = manual(varargin)
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @manual_OpeningFcn, ...
    'gui_OutputFcn', @manual_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before manual is made visible.
function manual_OpeningFcn(hObject, eventdata, handles, varargin)
fondo=imread('background.jpg');
axes(handles.axes1);
%axis off;
imshow(fondo);
handles.output = hObject;
handles.modelo = 'Pid_2019';
handles.vel=1;
    % set_param('Pid_2019/Control/control_altura/Controlador/P-F_Z','Value','0');
    % set_param('Pid_2019/Control/control_roll_pitch/Controlador/P-F_R-
P','Value','0');
```

```
% set_param('Pid_2019/Control/control_yaw/Controlador/P-F_Y','Value','0');

guidata(hObject, handles);

function varargout = manual_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

% --- Roll --- %
function Roll_Callback(hObject, eventdata, handles)
NewVal=str2num(get(hObject,'String'));
if (NewVal>=-100)&(NewVal<=100)
    set(handles.error_trayectoria,'String','');
else
    set(handles.error_trayectoria,'String','angulo incorrecta fuera de rango');
end
function Roll_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Pitch --- %
function Pitch_Callback(hObject, eventdata, handles)
NewVal=str2num(get(hObject,'String'));
if (NewVal>=-100)&&(NewVal<=100)
    set(handles.error_trayectoria,'String','');
else
    set(handles.error_trayectoria,'String','angulo incorrecta fuera de rango');
end
function Pitch_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
end
%--Yaw--%
function Yaw_Callback(hObject, eventdata, handles)
NewVal=str2num(get(hObject,'String'));
if (NewVal>=-181)&(NewVal<=181)
    set(handles.error_trayectoria,'String','');
else
    set(handles.error_trayectoria,'String','angulo incorrecta fuera de rango');
end
function Yaw_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Pz --- %
function Pz_Callback(hObject, eventdata, handles)
NewVal=str2num(get(hObject,'String'));
if (NewVal>=0)&(NewVal<=5)
    set(handles.error_trayectoria,'String','');
else
    set(handles.error_trayectoria,'String','altura incorrecta fuera de rango');
end
function Pz_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in build_b.
function build_b_Callback(hObject, eventdata, handles)
modelo=handles.modelo;
if bdIsLoaded(modelo)==0
    open(modelo);
```

```
end

rtwbuild(modelo); % default behaviour is to run the program
set_param(modelo,'SimulationCommand','connect'); % so connect to the drone
%set_param(modelo,'SimulationCommand','start'); % and stop the program

% find_system('name',modelo);
% open_system(modelo);
% set_param('sys','SimulationCommand','update')
% rtwbuild(ad.modelName);
Roll=str2num(get(handles.Roll,'String'));
Pitch=str2num(get(handles.Pitch,'String'));
Yaw=str2num(get(handles.Yaw,'String'));
Pz=str2num(get(handles.Pz,'String'));
if(length(Roll)==length(Pitch)&&length(Pitch)==length(Yaw)&&length(Pitch)==length(Yaw))
h(Pz)
    set(handles.error_trayectoria,'String','entradas correcta');
else
    set(handles.error_trayectoria,'String','dimensiones de vectores incorrectas');
end

% --- Executes on button press in start_b.
function start_b_Callback(hObject, eventdata, handles)
modelo=handles.modelo;
set_param(modelo,'SimulationCommand','start');

% --- Executes on button press in stop_b.
function stop_b_Callback(hObject, eventdata, handles)
modelo=handles.modelo;
set_param(modelo,'SimulationCommand','stop');

% --- Executes on button press in volar_b.
function volar_b_Callback(hObject, eventdata, handles)

% ENTRADAS
```

```
modelo=get(handles.modelo,'String');
Roll=str2num(get(handles.Roll,'String'));
Pitch=str2num(get(handles.Pitch,'String'));
Yaw=str2num(get(handles.Yaw,'String'));
Pz=str2num(get(handles.Pz,'String'));
% SETUP
modelo=handles.modelo;
if bdIsLoaded(modelo)==0
    open_system(modelo);
end
set_param('prueba/volar','value','1')

% LOOP
axes(handles.axes1);
% plot3(handles.axes1,X,Y,Z,':');
% hold on;
% axis manual

% SALIDAS
for i=1:length(Roll)
    plot3(handles.axes1,Roll,Pitch,Yaw,':');
    hold on;
    plot3(handles.axes1,Roll(i),Pitch(i),Yaw(i),'square');
    hold off;
    pause(1);
end
hold off;

% axis equal;

% --- Executes on button press in parar_b.
```

```
function parar_b_Callback(hObject, eventdata, handles)
if(get_param(strcat(modelo,'/pause'),'value')==1)
    set_param(strcat(modelo,'/pause'),'value','0');
elseif(get_param(strcat(modelo,'/pause'),'value')==0)
    set_param(strcat(modelo,'/pause'),'value','1');
end
```

% --- Executes on button press in regresar_b.

```
function regresar_b_Callback(hObject, eventdata, handles)
set_param('prueba/volar','value','1')
```

% --- Executes on button press in Regresar.

```
function Regresar_Callback(hObject, eventdata, handles)
% hObject    handle to Regresar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close(manual);
```