

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
ESCUELA DE POSGRADO



**NAVEGACIÓN Y ORIENTACIÓN DE UNA PEQUEÑA
EMBARCACIÓN**

Tesis para optar el grado de Magíster en Ingeniería Mecatrónica que
presenta

LUIS SANTOS COAQUIRA PAREDES

ASESOR:

Julio Cesar Tafur Sotelo, PhD. Ing.

Lima, 2017



DEDICATORIA

A mis padres y hermanos por ser los pilares fundamentales en el desarrollo de mi vida personal y por su incondicional apoyo a lo largo de este tiempo.



AGRADECIMIENTOS

iv

Quiero agradecer a todos mis profesores y a mis padres ya que ellos me enseñaron valorar los estudios y superarme día a día.

Estoy seguro que la dedicación empleada en la elaboración de la presente tesis dará fruto en el futuro. Y como todo en la vida si siembras esfuerzos al final cosecharas lo sembrado.



RESUMEN

Actualmente el monitoreo y toma de datos en embalses, lagunas y presas se realizan con embarcaciones de capacidad de carga de 4 personas como mínimo, esto supone costos por movilización de embarcación, personal y equipos, porque estos están ubicados en el interior del Perú. A esto si le agregamos el riesgo que corre el personal al ingresar a estas lagunas y presas que por ser lugares alejados, no existe equipos de salvataje y no existe personal con experiencia marina. Es por ello que el desarrollo de pequeñas embarcaciones autónomas controladas es una necesidad que necesita ser cubierta de acuerdo a los tiempos modernos que necesitamos de estar actualizados con el avance de la ciencia.

Es por ello que la presente tesis consistió en la construcción de un modelo experimental de un vehículo autónomo superficial (ASV) pensado para navegar en lagunas, presas y en cualquier tipo de aguas calmas y sirva de plataforma para llevar equipos y sensores de medición acuática. Este modelo experimental tendrá la capacidad de seguir una trayectoria definida previamente por el usuario y realizar la navegación de la ruta definida. La dirección de la embarcación se realizara por un timón, el posicionamiento se realizara por GPS y la propulsión se realizara por medio de un motor eléctrico y el control se realizara mediante un microcontrolador Atmel contenida en una tarjeta de desarrollo denominada ArduPilot.

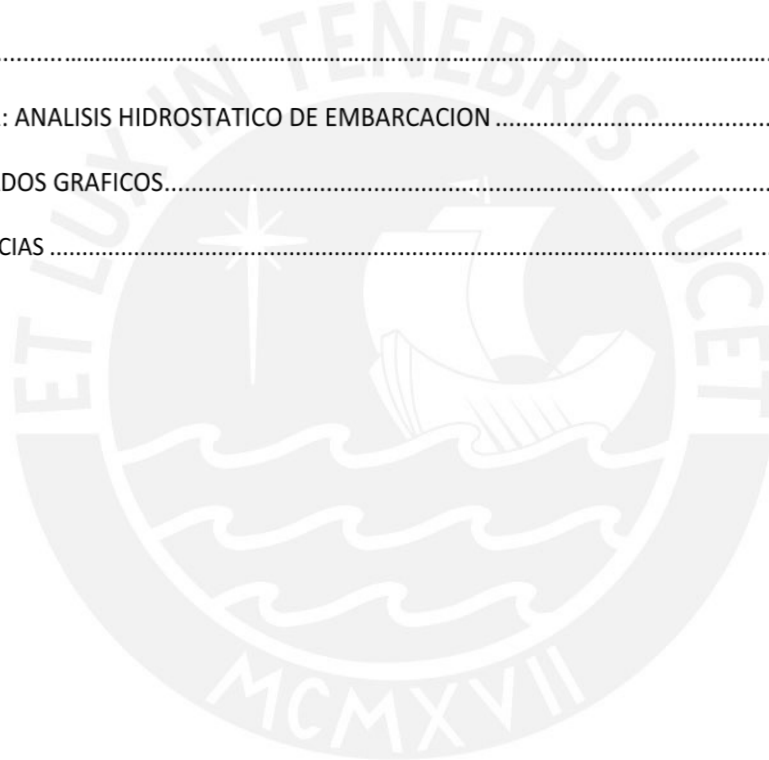
INDICE

vi

DEDICATORIA.....	III
AGRADECIMIENTOS.....	IV
RESUMEN.....	V
INDICE.....	VI
LISTA DE TABLAS.....	X
LISTA DE FIGURAS.....	XI
CAPÍTULO 1.....	1
1 INTRODUCCIÓN E INFORMACIÓN GENERAL.....	1
1.1 IDENTIFICACION DEL TEMA.....	1
1.2 ESTADO DEL ARTE.....	2
1.2.1 ESTADO DEL ARTE DE LOS <i>asv's</i> HIDROGRAFICOS.....	2
1.3 PLANTEAMIENTO DEL PROBLEMA.....	11
1.4 OBJETIVOS.....	12
1.4.1 OBJETIVOS GENERALES.....	12
1.4.2 OBJETIVOS ESPECIFICOS.....	12
1.5 LISTA DE EXIGENCIAS.....	13
2 CONCEPCION DE LA SOLUCION.....	15
2.1 ESTRUCTURA DE FUNCIONES.....	15
2.1.1 Estructura de funciones mecanicas.....	15
2.1.2 Estructura de funciones electronica.....	17
2.1.3 CALIFICACION DEL CONCEPTO DE SOLUCION electronica.....	19
3 PROYECTO PRELIMINAR.....	23
3.1 PROYECTO PRELIMINAR MECANICO.....	23
3.1.1 solucion nº1 DE EMBARCACION Nº1.....	26
3.1.2 solucion DE EMBARCACION nº 2.....	26
3.1.3 SOLUCION DE EMBARCACION Alternativa 3.....	26
3.1.4 selección DE ALTERNATIVA ÓPTIMA.....	27
3.2 PROYECTO PRELIMINAR ELECTRONICO.....	29
3.2.1 solucion nº 1 – electronica.....	29
3.2.2 solucion nº 2 - electronica.....	29

3.2.3	<i>solucion nº 3 – electronica</i>	30	vii
3.2.4	<i>selección DE ALTERNATIVA ÓPTIMA</i>	30	
4	PROYECTO DEFINITIVO.....	32	
4.1	ESTRUCTURA DE EMBARCACIÓN.....	32	
4.2	DESARROLLO DEL PROYECTO.....	34	
	4.2.1 <i>tarjeta de desarrollo</i>	34	
4.2.2	<i>SISTEMA DE POSICIONAMIENTO GLOBAL - GPS</i>	37	
4.2.3	<i>Motor Brushless y Helice de propulsion</i>	38	
4.2.4	<i>Regulador Electrónico de velocidad de 30Amp marca RC Timer y BEC 5v. Y2 A.</i>	41	
4.2.5	<i>Servomotor</i>	42	
4.2.6	<i>estructura de la embarcacion</i>	43	
5	MODELO DE EMBARCACION Y DISEÑO DE PILOTO AUTOMATICO.....	47	
5.1	MODELO LINEAL DE EMBARCACIÓN DE SUPERFICIE.....	47	
5.1.1	<i>Modelo newtoniano</i>	48	
	5.1.2 <i>MODELOS namoto</i>	50	
5.2	SISTEMA DE IDENTIFICACION.....	51	
	5.2.1 <i>Sistema de referencia</i>	51	
5.3	CINEMÁTICA DE LA EMBARCACION.....	52	
5.4	CONTROL DE DIRECCION.....	53	
5.4.1	<i>CONTROL DE EMBARCACION</i>	53	
5.4.2	<i>6.3.1 CÁLCULO DE LA DISTANCIA Y RUMBO</i>	53	
5.4.3	<i>CONTROLADOR PID</i>	54	
6	SOFTWARE.....	55	
6.1	SOFTWARE LIBRE.....	55	
6.2	SOFTWARE DEL PROYECTO.....	55	
6.2.1	<i>SOFTWARE ARDUINO</i>	55	
7	APLICACIONES PRINCIPALES.....	59	
	7.1 <i>BATIMETRIA</i>	59	
7.2	METODOLOGÍA DE MEDICION.....	59	
	<i>Generación de mapas</i>	59	
8	IMPLEMENTACION Y PRUEBAS.....	61	
8.1	EMBARCACION- FABRICACION.....	61	
8.2	DESARROLLO DEL SOFTWARE.....	63	
	8.2.1 <i>SOFTWARE ARDUINO</i>	63	

8.2.2	<i>Objetivo de Software</i>	63	viii
8.2.3	<i>Objetivo específico de Software</i>	63	
8.2.4	<i>ESTRUCTURA DEL PROGRAMA</i>	63	
8.2.5	<i>BUCLE PRINCIPAL DE CONTROL DE NAVEGACION</i>	64	
8.2.6	<i>ESTRUCTURA DE BLOQUES DEL PROGRAMA</i>	68	
		8.3 PRUEBAS	69
8.3.1	<i>INSTALACIÓN DE SOFTWARE</i>	69	
8.3.2	<i>INSTALACIÓN DEL ARDUINO IDE</i>	69	
8.3.3	<i>8.3 CONFIGURACIÓN DEL ARDUINO IDE</i>	69	
8.3.4	<i>PRUEBAS DE CAMPO</i>	70	
8.3.5	<i>PRUEBAS DE MODELO EXPERIMENTAL</i>	71	
ANEXOS.....			76
ANEXO 1: ANALISIS HIDROSTATICO DE EMBARCACION			76
RESULTADOS GRAFICOS.....			82
REFERENCIAS			103





LISTA DE TABLAS

TABLA Nº1. CUADRO DE INFORMACIÓN DE LA TESIS "NAVIGATION AND GUIDANCE OF AN AUTONOMOUS SURFACE".....	2
TABLA Nº2. CUADRO DE INFORMACIÓN DE LA TESIS "REMOTE DEPTH SURVEY OF THE CHARLES RIVER BASIN".....	3
TABLA Nº3. CUADRO DE INFORMACIÓN DE LA TESIS "SISTEMA DE TELEMETRÍA Y CONTROL DE UN BARCO AUTÓNOMO".....	4
TABLA Nº4. CUADRO DE INFORMACIÓN DE LA TESIS "PROJETO E CONSTRUÇÃO DE UM BARCO INTELIGENTE COM INTEGRAÇÃO INS/GPS E BÚSSOLA".....	5
TABLA Nº5 CUADRO DE INFORMACIÓN DE LA TESIS "AUTONOMOUS SURVEYING BOAT".....	6
TABLA Nº6. CUADRO DE INFORMACIÓN DE LA TESIS "CONTROLE AUTOMÁTICO DE RUMO DE UMA EMBARCAÇÃO DE SUPERFÍCIE NÃO TRIPULADA".....	7
TABLA Nº7. CUADRO DE INFORMACIÓN DE TESIS "PROJETO E CONSTRUÇÃO DE UMA EMBARCAÇÃO TELEOPERADA".....	8
TABLA Nº8. CUADRO DE INFORMACIÓN DE LA TESIS "NAVIGATION & CONTROL OF AN AUTOMATED SWATH SURFACE VESSEL FOR BATHYMETRIC MAPPING".....	9
TABLA Nº9. CUADRO DE INFORMACIÓN DE LA TESIS "GUIDANCE SYSTEM FOR AUTONOMOUS SURFACE VEHICLES".....	10
TABLA Nº10 MATRIZ MORFOLÓGICA MECÁNICA.....	16
TABLA Nº11 CONCEPTO SOLUCIÓN MECÁNICA.....	17
TABLA Nº12 MATRIZ MORFOLÓGICA ELECTRÓNICA.....	19
TABLA Nº13 CONCEPTO SOLUCIÓN ELECTRÓNICA.....	20
TABLA Nº14 MATRIZ MORFOLÓGICA SOFTWARE.....	21
TABLA Nº15 CONCEPTO SOLUCIÓN SOFTWARE.....	22
TABLA Nº16 COMPARACIÓN DE ESFUERZOS.....	25
TABLA Nº17 EVALUACIÓN TÉCNICA DE SOLUCIÓN MECÁNICA.....	27
TABLA Nº18 EVALUACIÓN ECONÓMICA DE SOLUCIÓN MECÁNICA.....	28
TABLA Nº19 EVALUACIÓN TÉCNICA DE SOLUCIÓN ELECTRÓNICA.....	30
TABLA Nº20 EVALUACIÓN ECONÓMICA DE SOLUCIÓN ELECTRÓNICA.....	31
TABLA Nº21 RPM IDEAL DE ESPECIFICACIONES DE MOTORES.....	40

LISTA DE FIGURAS

Figura N°1.	Embarcación desarrollada en estudio.....	3
Figura N°2.	Embarcación y hardware utilizado en el proyecto de tesis	4
Figura N°3.	Plataforma de pruebas (barco) del estudio	6
Figura N°4.	Embarcación desarrollada para medir la profundidad	7
Figura N°5.	Embarcación desarrollada en el proyecto	8
Figura N°6.	Embarcación tele operada	9
Figura N°7.	Embarcación en pleno trabajo.....	10
Figura N°8.	Caja Negra de funciones mecánicas.....	15
Figura N°9.	Estructura de funciones parciales mecánicas.	16
Figura N°10.	Caja Negra electrónica	17
Figura N°11.	Estructura de funciones parciales electrónicas.....	18
Figura N°12.	Caja Negra software.....	20
Figura N°13.	Estructura de funciones parciales software.....	21
Figura N°14.	Solución N°1-Electronica	29
Figura N°15.	Solución N°2-Electronica	29
Figura N°16.	Solución N°3-Electronica	30
Figura N°17.	Estructura de la embarcación.....	32
Figura N°18.	Diagrama de bloques de componentes electrónicos.	33
Figura N°19.	Board Ardupilot	35
Figura N°20.	Diagrama de bloques del Autopilot	36
Figura N°21.	Vista de Conexiones Ardupilot.....	36
Figura N°22.	Módulo de GPS EM-406 y distribución de pines	38
Figura N°23.	Vista de un módulo de potencia 3DR (PM).....	38
Figura N°24.	Fuerza de empuje, Potencia y Velocidad de hélice Aeronáutica clásica 12"x7"	39
Figura N°25.	ESC 30A Brushless Motor Speed Controller	41
Figura N°26.	Motor RCX A2830-11 1000kV	41
Figura N°27.	Servo Motor standard	42
Figura N°28.	Vista lateral de dimensiones de la embarcación y el calado de diseño.	43
Figura N°29.	Vista frontal y posterior y el calado de diseño.....	44
Figura N°30.	Grafica Velocidad vs Resistencia de la embarcación.	45
Figura N°31.	Vista de láminas generadas a partir de las dimensiones calculadas previamente.	46
Figura N°32.	Notación estándar y las convenciones de signo para la descripción del movimiento de una embarcación (SNAME, 1950).....	47
Figura N°33.	Las variables que describen el movimiento del modelo físico en el plano horizontal.	52
Figura N°34.	Vista del software Arduino.....	56
Figura N°35.	Software Google Earth	57
Figura N°36.	Obtención de las coordenadas de posición usando del Software Google Earth.	58
Figura N°37.	Vista de las coordenadas de posición de una posición usando del Software Google Earth	58
Figura N°38.	Vista de la Trayectoria seguida por una embarcación para realizar trabajos de medición de profundidad	60

Figura N°39.	Vista de Resultado del Trabajo Batimetrico	60	xii
Figura N°40.	Vista de Molde fabricado en yeso	61	
Figura N°41.	Fabricación de Modelo molde de resina de Poliéster	62	
Figura N°42.	Producto final – embarcación de resina de Poliéster	62	
Figura N°43.	Selección del hardware	69	
Figura N°44.	Conexión del ArduPilot con el adaptador de programación.	70	
Figura N°45.	Vista obtenida de Google Earth de la Laguna Rajulcota en Huaraz	71	
Figura N°46.	Vista de area presa de la Laguna Rajulcota en Huaraz	71	
Figura N°47.	Vista de prueba de flotabilidad del Modelo experimental N°1.	72	
Figura N°48.	Vista de prueba hidrodinamica del Modelo experimental N°1.	72	
Figura N°49.	Vista obtenida de Google Earth junto con la trayectoria de la prueba tipo C del Modelo experimental.	73	
Figura N°50.	Vista del Modelo experimental en trayectoria de la prueba tipo C del Modelo experimental.....	73	
Figura N°51.	Vista obtenida de Google Earth junto con la trayectoria de la prueba tipo Zic Zac del Modelo experimental.	74	
Figura N°52.	Vista del Modelo experimental en trayectoria Zic Zac del Modelo experimental.....	74	



CAPÍTULO 1

1 INTRODUCCIÓN E INFORMACIÓN GENERAL

1.1 IDENTIFICACION DEL TEMA

El cambio climático y los impactos realizados por el hombre con el medio ambiente, en particular en los mares, ríos y atmósfera (por ejemplo, las emisiones de contaminantes de gases y efluentes vertidos en ríos, lagos y océanos) son motivos para investigadores e ingenieros de todo el mundo para desarrollar proyectos de monitoreo y control ambiental más eficiente y sostenible.

Uno de estos estudios es el monitoreo del relieve bajo el agua en el puertos, presas, ríos y relaves mineros con diferentes fines en cada uno de ellos.

Aunque hoy en gran parte de estos estudios y monitoreo se realizan en el mar, hay también muchas aplicaciones en presas, ríos, lagos y embalses de agua, que son estudios similares.

Un ejemplo típico son las presas de centrales hidroeléctricas, estas presas deben ser monitoreados periódicamente con la finalidad de conocer el relieve de fondo del embalse de presa, para conocer el volumen total de agua disponible y la distribución del relieve de la cuenca con el fin de programar mantenimientos de presa y conocer la energía almacenada.

Aunque el proceso actual emplea una embarcación de tamaño de 5m a 10m, que alberga un motorista, un operador del sistema de medición y personal de apoyo, pero en todos los casos se puede utilizar este sistema. Porque por desgracia no todos los sitios a monitorear son lugares tranquilos y de fácil acceso, a veces hay lugares sin acceso vehicular por lo tanto imposible de transportar la embarcación y también se dan casos en que el agua a navegar son de profundidad limitada en donde una embarcación corre el riesgo de que las hélices y timón choquen con el fondo.

Una buena alternativa a esas dificultades que se ofrece hoy es la utilización de vehículos no tripulados de superficie (USV, por sus siglas en inglés), barcos robóticos que pueden reemplazar un equipo humano y automáticamente puedan lograr su objetivo.

Eliminar la necesidad de llevar a seres humanos a bordo reduce significativamente el tamaño y el peso del barco.

Es evidente cómo esto afecta positivamente todo el proceso, tanto en términos de eficiencia, facilidad de manejo, costo y por último pero no menos importante, la seguridad del personal.

1.2 ESTADO DEL ARTE

Este ítem se presenta los resultados de la reunión de información sobre el estado de desarrollo en el campo de vehículos de superficie autónoma (ASV) de medianas y pequeñas dimensiones desarrollados por universidades y centros de investigación.

Además se indicara su nivel de desarrollo alcanzado, las aplicaciones que tuvieron dichos desarrollos, la tecnología y las formas de inteligencia asociada.

1.2.1 ESTADO DEL ARTE DE LOS ASV'S HIDROGRAFICOS.

Los últimos años se realizaron varios estudios en universidades y recopilamos los relacionados al tema del presente trabajo.

TESIS 1: En el año 2007 la *University of Southern California* desarrollo la tesis titulada *“Navigation and Guidance of an Autonomous Surface”* en esa tesis se convirtió una pequeña embarcación en un vehículo autónomo de superficie capaz de navegar en lagos y mares, este vehículo para navegación y guiado usa un GPS + IMU. Además utiliza visión estéreo para detectar obstáculos en su ruta de navegación. Esta embarcación tiene como objetivo la recolección de datos sobre algas marinas [1].

Tabla N°1. Cuadro de información de la tesis *“Navigation and Guidance of an Autonomous Surface”*:


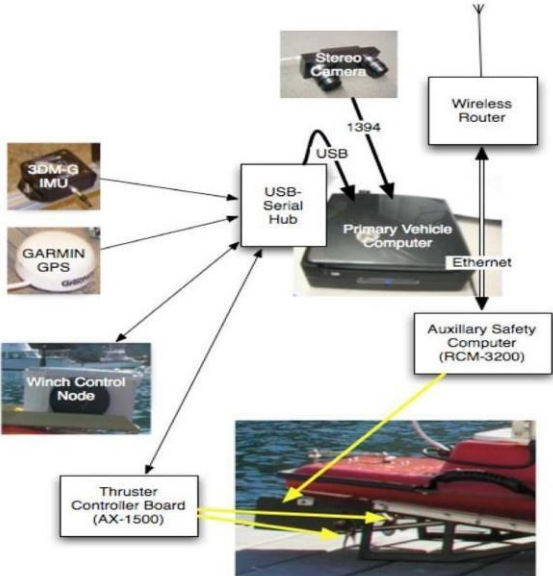
Tesis/ Universidad	Especificaciones	Hardware (Implementación)
 <p>“Navigation and Guidance of an Autonomous Surface” / University of Southern California [1]</p>	<ul style="list-style-type: none"> • Año: 2007 • Dimensiones: 0.45. x 0.2m. • Propulsión: Motor eléctrico. • Autonomía: 6 a 8 horas. • Navegación: GPS + IMU + Cámara Estéreo. • Aplicación: Recolección de datos Sobre las Algas marinas. 	



Figura N°1. Embarcación desarrollada en estudio [1].

TESIS 2: En el año 2007 en el *Massachusetts Institute of Technology* se desarrolló una tesis denominado “*Remote Depth Survey of the Charles River Basin*”, el propósito de este proyecto fue hacer un estudio de profundidad de la cuenca de un río con un prototipo USV con la finalidad de saber si este tipo de tecnología podría utilizarse con éxito en estudios hidrográficos a gran escala. Para lo cual se utilizó una PC portátil y un equipo de comunicaciones inalámbrica entre el operador y la embarcación, con la finalidad de obtener los datos en tiempo real [2].

Tabla N°2. Cuadro de información de la tesis “*Remote Depth Survey of the Charles River Basin*”.

Modelo/ fabricante	Especificaciones	Hardware (Implementación)
“Remote Depth Survey of the Charles River Basin” / Massachusetts Institute of Technology. [2]	<ul style="list-style-type: none"> • Año: 2007 • Dimensiones: 3.7 m. x 0.74m. • Propulsión: Motor eléctrico • Velocidad: 4.4kn • Navegación: GPS + Compas. • Aplicación: Medición de Fondo marino. 	

TESIS 3: En el Año 2010 la *Universidad Carlos III de Madrid* realizó la tesis denominada “*Sistema de telemetría y control de un barco autónomo*”. En este estudio se utilizó un bote a control remoto el cual se modificó y se implementó un control con la técnica de control PID. La navegación y el guiado se realizaron por medio de GPS y magnetómetro, esta embarcación podía seguir trayectorias previamente definidas por el usuario y además poseía comunicación entre la estación de tierra constituida por una PC y la embarcación mediante WiFi.

Tabla N°3. Cuadro de información de la tesis “*Sistema de telemetría y control de un barco autónomo*”.


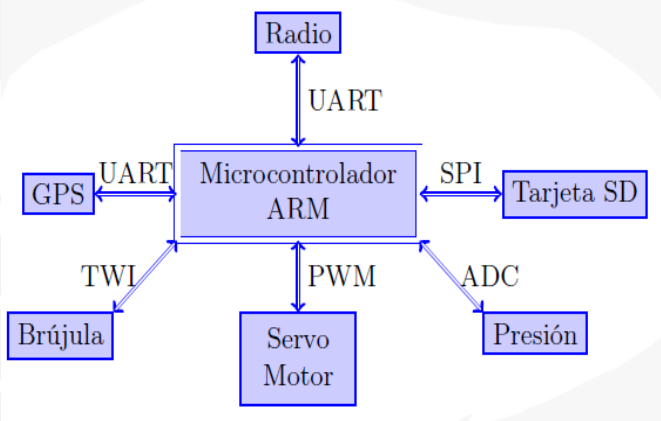
Modelo/ fabricante	Especificaciones	Hardware (Implementación)
 <p>“Sistema de telemetría y control de un barco autónomo” / Universidad Carlos III de Madrid. [3]</p>	<ul style="list-style-type: none"> • Año: 2010. • Dimensiones: 0.8 m. x 0.25m. • Propulsión: Motor eléctrico. • Velocidad: 1m/s. • Navegación: GPS + Compas. • Aplicación: Investigación. 	



Figura N°2. Embarcación y hardware utilizado en el proyecto de tesis [3].

TESIS 4: En Brasil el año 2011 *El Instituto Tecnológico de Aeronáutica São José dos Campos* se realizó un proyecto de tesis para optar el título de doctor en ciencias el estudio denominado "*Projeto e Construção de um Barco Inteligente com Integração INS/GPS e Bússola*". En este proyecto se desarrolló una plataforma de navegación capaz de seguir pequeños circuitos impulsada por 02 motores eléctricos. Para la Posición y navegación utilizo GPS, IMU (unidad de medida inercial), Magnetómetro y además aplico la técnica de filtro de Kalman para corregir errores en posición y navegación [4].

Tabla N°4. Cuadro de información de la tesis "*Projeto e Construção de um Barco Inteligente com Integração INS/GPS e Bússola*".


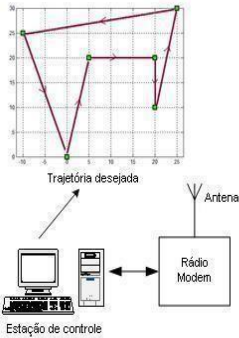

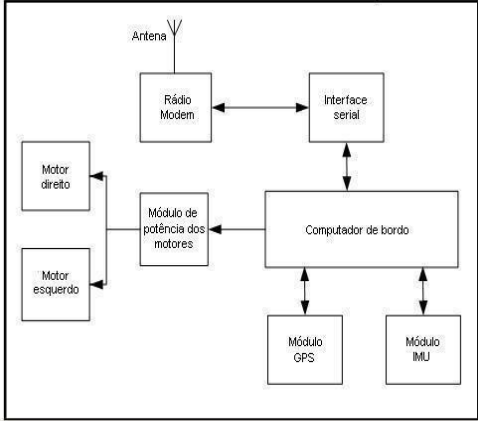
Modelo/ fabricante	Especificaciones	Hardware (Implementación)
 <p>Projeto e Construção de um Barco Inteligente com Integração INS/GPS e Bússola. / Instituto Tecnológico de Aeronáutica, São José dos Campos [4]</p>	<ul style="list-style-type: none"> • Año: 2011. • Dimensiones: 2.5m x 2.0. • Propulsión: Motor eléctrico. • Navegación: GPS + IMU. • Aplicación: Investigación. 	  

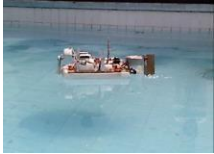
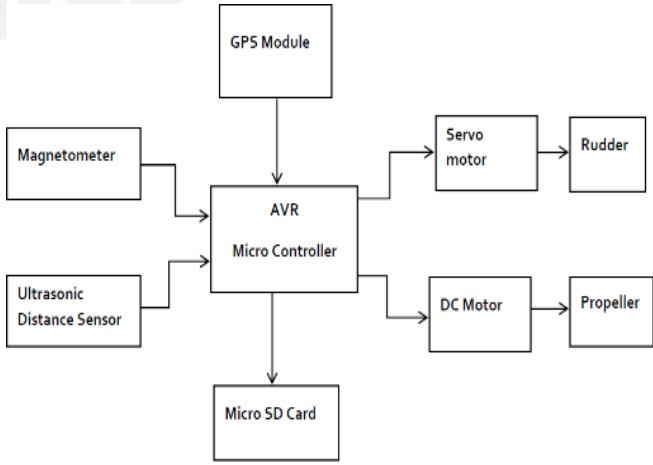


Figura N°3. Plataforma de pruebas (barco) del estudio [4].

TESIS 5: En el año 2012 la *University of Engineering and Technology Peshawar* – Pakistán desarrollo un proyecto de tesis denominado “*Autonomous surveying Boat*” con el objetivo de registrar profundidades del agua en una presa a lo largo de un conjunto de puntos predefinidos.

Obtener el nivel del agua de embalses permitirá determinar la presión y el flujo de agua en presas. Para guiado y navegación de la embarcación se emplea un GPS y un magnetómetro [5].

Tabla N°5. Cuadro de Información de la tesis “*Autonomous surveying Boat*”:

Modelo/ fabricante	Especificaciones	Hardware (Implementación)
 <p>“Autonomous surveying Boat”/ University of Engineering and Technology Peshawar. [5]</p>	<ul style="list-style-type: none"> • Año: 2012. • Dimensiones: 0.4m x 0.2. • Propulsión: Motor eléctrico. • Navegación: GPS + magnetómetro. • Aplicación: Investigación. 	 <pre> graph TD GPS[GPS Module] --> AVR[AVR Micro Controller] Magnetometer --> AVR US[Ultrasonic Distance Sensor] --> AVR AVR --> Servo[Servo motor] Servo --> Rudder[Rudder] AVR --> DC[DC Motor] DC --> Propeller[Propeller] AVR --> SD[Micro SD Card] </pre>

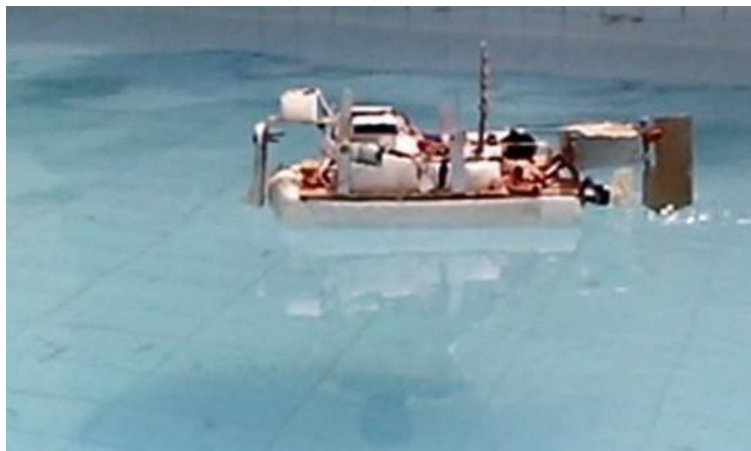


Figura N°4. Embarcación desarrollada para medir la profundidad [5].

TESIS 6: En Brasil en el año 2012 la *Universidade do Estado do Rio de Janeiro* desarrollo la tesis “*Controle Automático de Rumo de uma Embarcação de Superfície não Tripulada*” en este trabajo se realizó un sistema de control automático capaz de mantener una embarcación no tripulada en un rumbo definido, independientemente de las perturbaciones externas. Para lograr este objetivo se desarrolló un algoritmo capaz de corregir la orientación de la embarcación en base a la posición de referencia y la posición medida por el magnetómetro. Se realizó la simulación del sistema propuesto en MATLAB y se presentaron pruebas y resultados [6].

Tabla N°6. Cuadro de información de la tesis “*Controle Automático de Rumo de uma Embarcação de Superfície não Tripulada*”


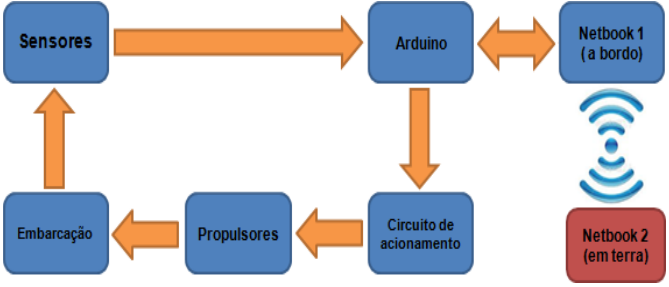
Modelo/ fabricante	Especificaciones	Hardware (Implementación)
 <p>“Controle Automático de Rumo de uma Embarcação de Superfície não Tripulada” / Universidade do Estado do Rio de Janeiro. [6]</p>	<ul style="list-style-type: none"> • Año: 2012. • Dimensiones: 3.15m x 0.80m. • Propulsión: 02 Motores eléctrico. • Navegación: Magnetómetro. • Aplicación: Investigación. 	



Figura N°5. Embarcación desarrollada en el proyecto [6].

TESIS 7: También en el mismo año 2012 en Brasil en la *Universidade do Estado do Rio de Janeiro* se desarrolló la tesis *“Projeto e construção de uma embarcação teleoperada”*. El proyecto desarrollo una embarcación teleoperada movida por 02 motores eléctricos de corriente continua. La embarcación se equipó con una Computadora equipada con WiFi y una Tarjeta de desarrollo Arduino que controla los motores [7].

Tabla N°7. Cuadro de información de tesis *“Projeto e construção de uma embarcação teleoperada”*:


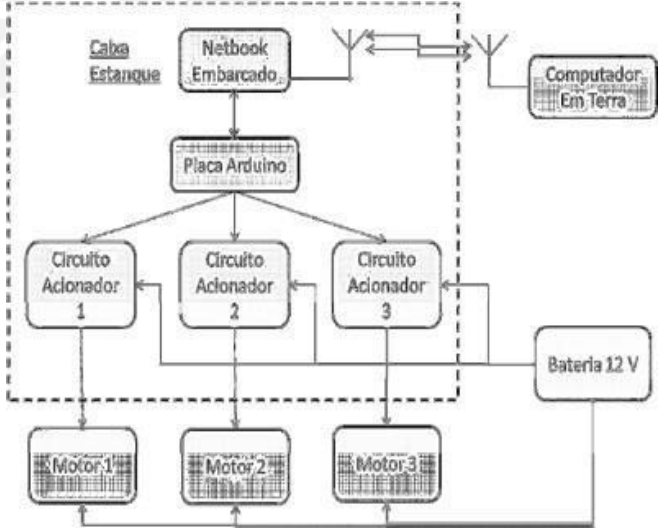
Modelo/ fabricante	Especificaciones	Hardware (Implementación)
 <p>Projeto e construção de uma embarcação teleoperada / Universidade do Estado do Rio de Janeiro.</p> <p>[7]</p>	<ul style="list-style-type: none"> • Año: 2012. • Dimensiones: 3.15m x 0.80m. • Propulsión: 02 Motores eléctrico. • Navegación: Magnetómetro. • Aplicación: Investigación. 	



Figura N°6. Embarcación tele operada [7].

TESIS 8: En el Año 2013 en la *Universidad de Santa Clara* – California se desarrolló la tesis titulada “*Navigation & Control of an Automated SWATH Surface Vessel for Bathymetric Mapping*” en este estudio se diseñó y construyó una embarcación autónoma con el objetivo de medir las profundidades y crear mapas batimétricos (relieve marino). Para generar los mapas usa un sonar multibeam y para la navegación se utilizó GPS [8].

Tabla N°8. Cuadro de información de la tesis “*Navigation & Control of an Automated SWATH Surface Vessel for Bathymetric Mapping*”


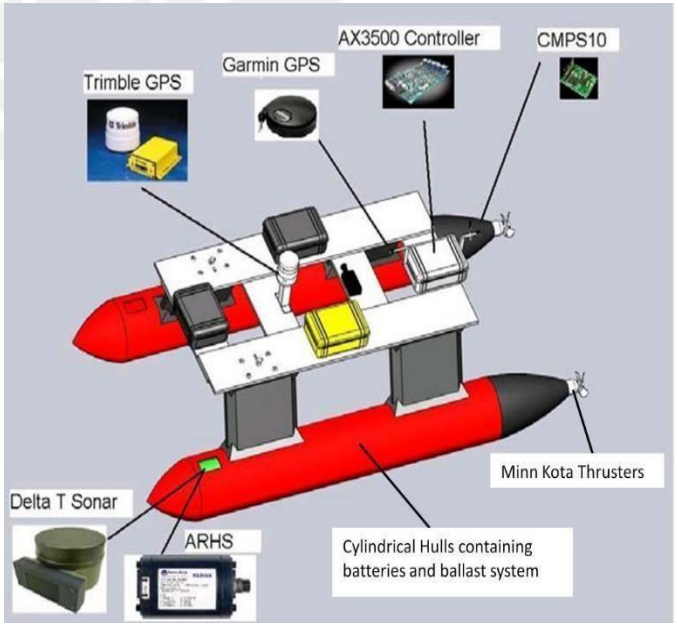
Modelo/ fabricante	Especificaciones	Hardware (Implementación)
 <p>“Navigation & Control of an Automated SWATH Surface Vessel for Bathymetric Mapping” / Santa Clara University. [8]</p>	<ul style="list-style-type: none"> • Año: 2013. • Dimensiones: 3.4m x 1.6m. • Propulsión: 02 Motores eléctrico. • Navegación: GPS. • Aplicación: Medición de Fondo Para obtener mapas del relieve marino. 	



Figura N°7. Embarcación en pleno trabajo [8].

TESIS 9: En el año 2015 en noruega en la *University of Science and Technology* se desarrolló una tesis denominada “*Guidance System for Autonomous Surface Vehicles*” en esta tesis se desarrolló un sistema anticolidión para una embarcación de superficie autónoma (ASV) pueda operar en el mar y en tráfico marítimo. Para validar el algoritmo se implementó un simulador para su funcionamiento [9].

Tabla N°9. Cuadro de información de la tesis “*Guidance System for Autonomous Surface Vehicles*”

Modelo/ fabricante	Especificaciones	Hardware (Implementación)
<p>“Guidance System for Autonomous Surface Vehicles /Norwegian University of Science and Technology [9]</p>	<ul style="list-style-type: none"> • Año: 2015. • Navegación: GPS+IMU. • Aplicación: simulación. 	

1.3 PLANTEAMIENTO DEL PROBLEMA

En las presas se necesita conocer el relieve del fondo del agua embalsada para determinar el relieve de los sedimentos y su monitoreo a lo largo del tiempo con el fin de programar la descarga y limpieza de los sedimentos, evitando así que los sedimentos tapen las compuertas y otras estructuras.

El cubrimiento de sedimentos ocasionaría pérdidas económicas por que se reduciría el volumen de embalse y el destrabe de la compuerta implica un alto costo económica por que se tendría que traer buzos e equipos para realizar esas tareas.

Una buena alternativa de solución para conocer el relieve del fondo del agua embalsada para determinar el relieve de los sedimentos y su monitoreo a lo largo del tiempo es la utilización de vehículos no tripulados de superficie (USV) que pueden ejecutar las mediciones y las ventajas que se obtendrían con respecto al sistema tradicional empleando una embarcación de 6 m. a 10 m de largo serán las siguientes:

- Mayor seguridad para el personal reduciendo el contacto con el agua y el tiempo de exposición a la radiación solar.
- Mayor seguridad para el personal al evitar que ingresen en aguas con obstáculos y en aguas bajas.
- Mayor control de costos de operación ya que es operado por una sola persona
- Disponibilidad de los datos de las mediciones en tiempo real.
- Menor peso y calado que permiten la obtención de mediciones precisas en aguas someras (aguas de baja profundidad).
- Reducción del nivel de ruido, y la contaminación ambiental, al utilizar motores eléctricos en lugar de los motores de combustión de las embarcaciones.
- Mayor frecuencia de monitoreo o vigilancia.

1.4 OBJETIVOS

1.4.1 OBJETIVOS GENERALES

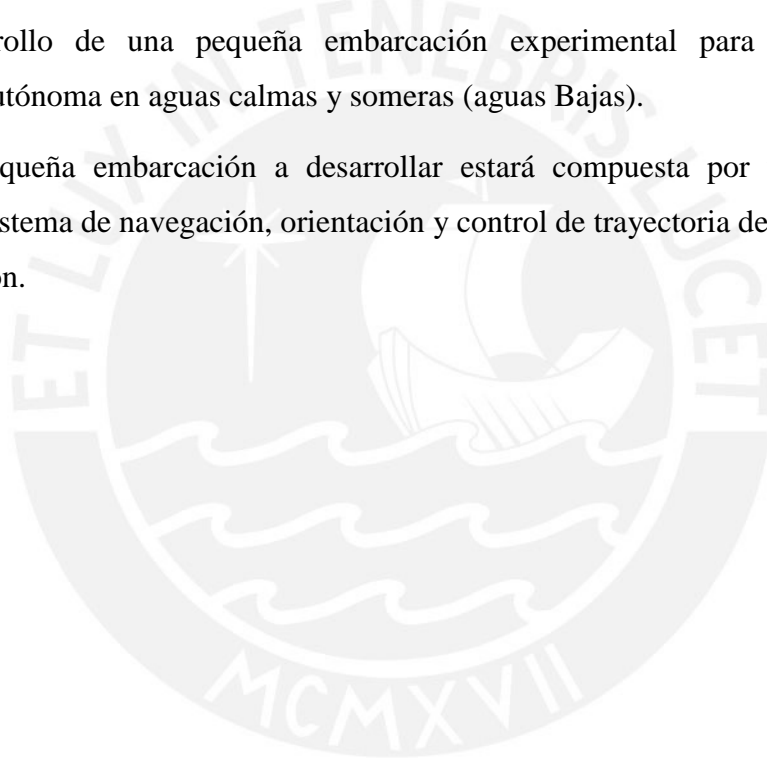
Los objetivos generales que persigue el presente estudio son:

- Investigar el avance y desarrollo de las investigaciones en control navegación marina de vehículos experimentales.
- Estudio y desarrollo de las ciencias mecánicas, electrónicas y de control para la implementación de una pequeña investigación experimental.
- Diseño mecánico de la de la embarcación experimental.

1.4.2 OBJETIVOS ESPECIFICOS

Desarrollo de una pequeña embarcación experimental para navegación y orientación autónoma en aguas calmas y someras (aguas Bajas).

La pequeña embarcación a desarrollar estará compuesta por un sistema de propulsión, sistema de navegación, orientación y control de trayectoria de navegación de la embarcación.



1.5 LISTA DE EXIGENCIAS

LISTAS DE EXIGENCIAS				
Proyecto:		Navegación y orientación de una pequeña embarcación autónoma para medición de profundidad.		Fecha: Mar 2016. Revisado: J.T.S.
Escuela de Posgrado		Pontificia Universidad católica del Perú.		
Cambios (fechas)	Características	Deseo o exigencia	Descripción	Realizado Por:
Marzo de 2016	Función	E	Navegación y orientación de la embarcación siguiendo una trayectoria trazado por el usuario	L.S.C.P.
		E	La embarcación deberá ser capaz de obtener su posición en el espacio.	L.S.C.P.
		E	La embarcación tendrá que albergar e instalar los equipos, baterías y equipos de medición de profundidad.	L.S.C.P.
		E	La embarcación operara en aguas calmas.	L.S.C.P.
Marzo de 2016	Ergonomía	E	El manejo de la embarcación debe ser amigable y sencillo.	L.S.C.P.
Marzo de 2016	Energía	E	Deberá ser suministrada por baterías de electricidad.	L.S.C.P.
Marzo de 2016	Geometría	E	La embarcación será un tamaño entre 0.9 a 1.2 m. y de superficie aerodinámica.	L.S.C.P.
Marzo de 2016	Velocidad	E	La velocidad de la embarcación será de 3 a 5 m/s.	L.S.C.P.
Marzo de 2016	Fuerzas	E	Las fuerzas de impulsión deberán propulsar la embarcación	L.S.C.P.
Marzo de 2016	Seguridad	E	La embarcación deberá ser estable y hermético para impedir que el agua ingrese y dañe los equipos electrónicos.	L.S.C.P.

Marzo de 2016	Ergonomía	E	La Embarcación deberá tener manijas para levantar y manipular la embarcación.	L.S.C.P.
Marzo de 2016	Señales	E	Uso de GPS para posición y Mi controladores y/o tarjetas de desarrollo para procesamiento de los datos.	L.S.C.P.
Marzo de 2016	Control	E	Empleo de algoritmo de control	L.S.C.P.
Marzo de 2016	Programación	E	Desarrollo Código de control.	L.S.C.P.
Marzo de 2016	Fabricación	E	La máquina deberá poder fabricarse en talleres locales.	L.S.C.P.
Marzo de 2016	Costo	E	El costo será económico.	L.S.C.P.
Marzo de 2016	Montaje	E	La Embarcación deberá ser de fácil montaje e implementación.	L.S.C.P.
Marzo de 2016	Plazo de Entrega	E	14 Semanas.	L.S.C.P.

Fuente: Curso metodológica de diseño PUCP.

2 CONCEPCION DE LA SOLUCION.

2.1 ESTRUCTURA DE FUNCIONES.

2.1.1 ESTRUCTURA DE FUNCIONES MECANICAS.

La función total Mecánica (flujo de energía) de la embarcación puede representarse en forma de una caja negra, donde se tienen en cuenta las magnitudes de entrada y de salida (ver figura 8).

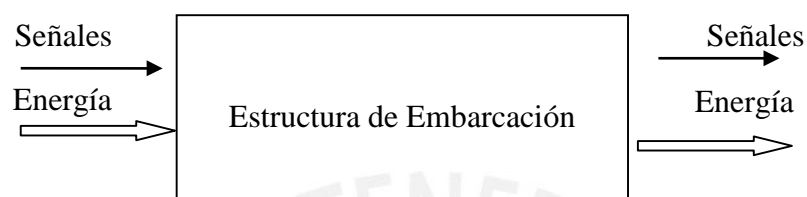


Figura N°8. Caja Negra de funciones mecánicas.

INGRESO

- Energía: Energía eléctrica como fuerza motriz.
- Señales: Señal para inicio del proceso.

SALIDA

- Energía: Desplazamiento, vibraciones, fricción, ruidos.
- Señales: Señal de finalización del proceso.

2.1.1.1 DESCRIPCIÓN DE FUNCIONES

- Encender: Suministro de energía.
- Conversión de energía: Consiste en la transformación de energía suministrada a energía mecánica.
- Impulsión: La anergia Mecánica debe convertirse en fuerza de impulsión.
- Dirección de impulsión: la fuerza de impulsión debe ser orientada para seguir un rumbo o dirección.
- Control del proceso: inicio, control de dirección y fin del proceso.
- Apagar: Desactivar suministro de energía.

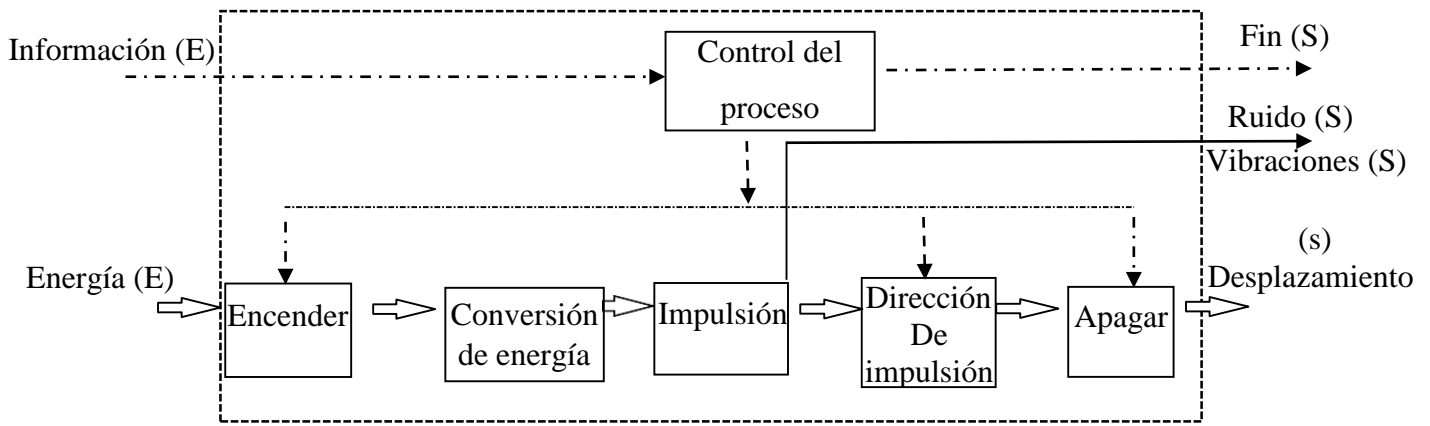
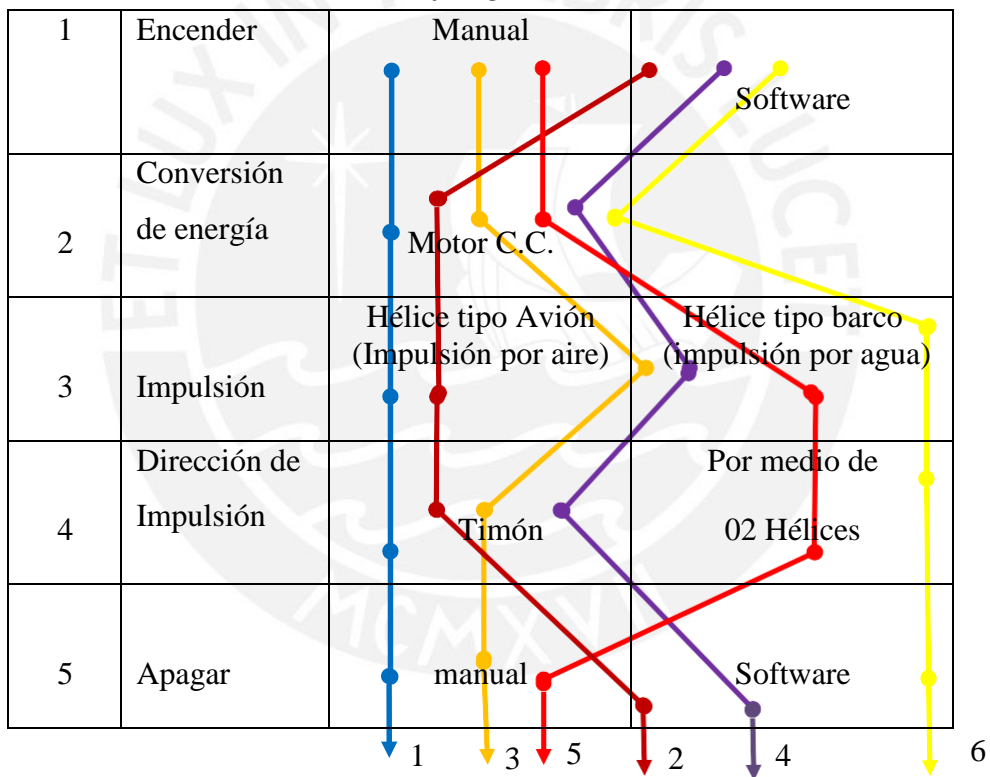


Figura N°9. Estructura de funciones parciales mecánicas.

2.1.1.2 **MATRIZ MORFOLOGICA MECANICA**

Tabla N°10. *Matriz morfológica mecánica.*



LEYENDA

Concepto solución	Tipo de línea
1	
2	
3	
4	
5	
6	

2.1.1.3 CALIFICACION DEL CONCEPTO DE SOLUCION MECANICA

En base a la matriz morfológica se realizara la evaluación del concepto solución mecánica. Se procede a hacer una evaluación tomando valores del 1 al 4, donde:

0: No aceptable 2: Suficiente 4: Muy satisfactorio

1: poco satisfactorio 3: Satisfactorio

Tabla N°11. *Concepto solución mecánica.*

N°1	Criterios Técnico-Económicos	Soluciones					
		1	2	3	4	5	6
1	Buen uso de Energía	2	2	2	2	2	2
2	Complejidad	3	3	3	3	2	3
3	Facilidad de montaje	3	3	3	3	2	2
4	Facilidad de manejo	2	3	2	3	2	3
5	Lista de exigencias	2	3	2	2	2	3
6	Facilidad de Adquisición	2	2	2	2	2	2
7	Costo de tecnología	3	3	3	3	3	3
8	Facilidad de mantenimiento	2	2	2	2	2	2
	Suma Total	19	21	19	20	17	20

El concepto solución que tuvo mayor puntaje fue el número 2.

2.1.2 ESTRUCTURA DE FUNCIONES ELECTRONICA

La función de electrónica (flujo de data - hardware) de la embarcación se puede representarse en forma de caja negra, donde se tienen en cuenta las magnitudes de entrada y de salida (ver figura N°10).

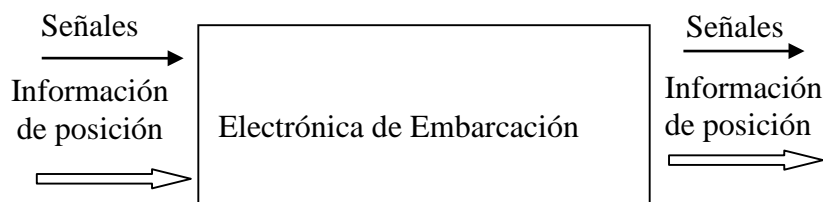


Figura N°10. Caja Negra electrónica.

INGRESO

- Información de posición: datos de posición en el espacio.
- Señales: Señal para inicio del proceso.

SALIDA

- Información de posición: posición de la embarcación.
- Señales: Señal de finalización del proceso.

2.1.2.1 DESCRIPCIÓN DE FUNCIONES

- Inicio: Activación del Inicio del proceso.
- Receptor de posición: Consiste en la adquisición de datos de posición en el espacio.
- Receptor de profundidad: Consiste en la adquisición de datos profundidad.
- Procesamiento: Procesamiento de la data recibida por el GPS.
- corrección de posición: Corrección de posición para seguir un rumbo o dirección deseada.
- Apagar: Desactivar el proceso.

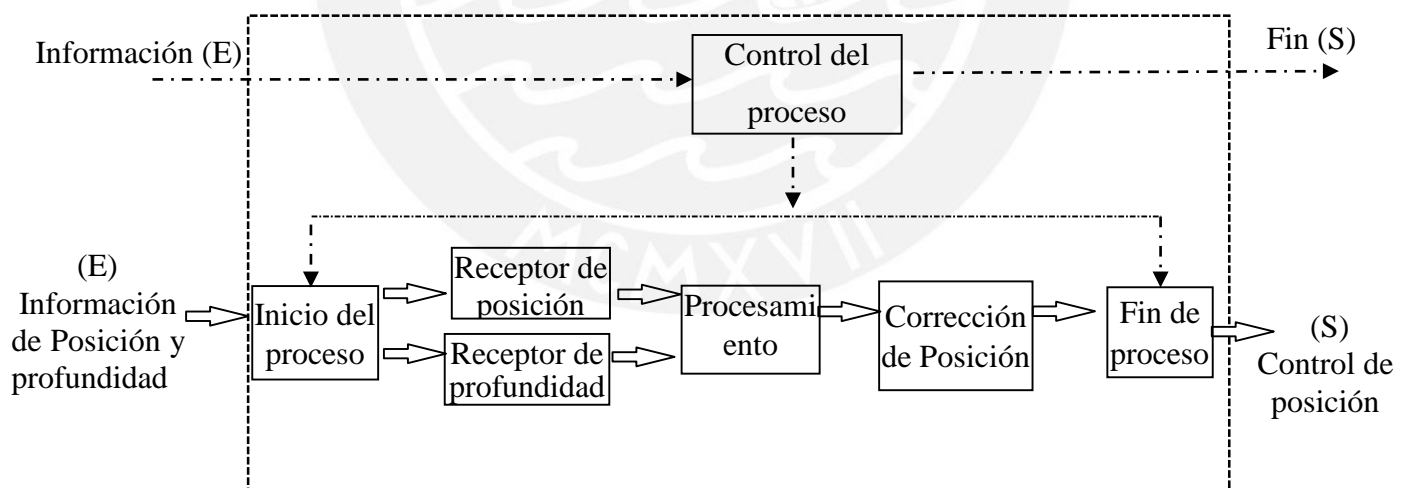
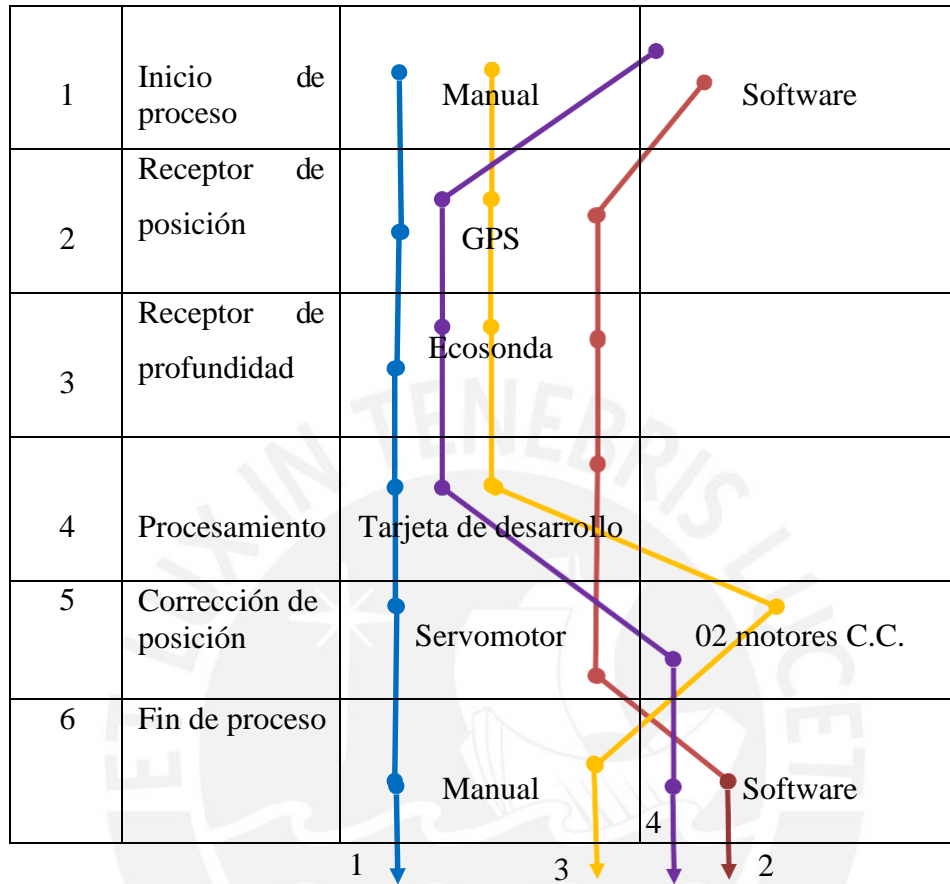


Figura N°11. Estructura de funciones parciales electrónicas.

2.1.2.2 **MATRIZ MORFOLOGICA ELECTRONICA**

Tabla N°12. *Matriz morfológica electrónica.*



LEYENDA

Concepto solución	Tipo de línea
1	
2	
3	
4	

2.1.3 **CALIFICACION DEL CONCEPTO DE SOLUCION ELECTRONICA**

En base a la matriz morfológica se realizara la evaluación del concepto solución electrónica. Se procede a hacer una evaluación tomando valores del 1 al 4, donde:

- 0: No aceptable
- 2: Suficiente
- 4: Muy satisfactorio

- 1: poco satisfactorio
- 3: Satisfactorio.

Tabla N°13. *Concepto solución electrónica.*

N°	Criterios Técnico-Económicos	Soluciones			
		1	2	3	4
1	Complejidad	2	3	2	3
2	Facilidad de montaje	3	3	2	2
3	Lista de exigencias	2	3	2	3
4	Facilidad de Adquisición	2	2	2	2
5	Costo de tecnología	2	2	2	2
6	Facilidad de mantenimiento	3	3	3	3
	Suma Total	14	16	13	15

El concepto solución que tuvo mayor puntaje fue el número 2.

2.1.3.1 *ESTRUCTURA DE FUNCIONES SOFTWARE*

La función de software de la embarcación se puede representarse en forma de caja negra, donde se tienen en cuenta las magnitudes de entrada y de salida (ver figura 12).

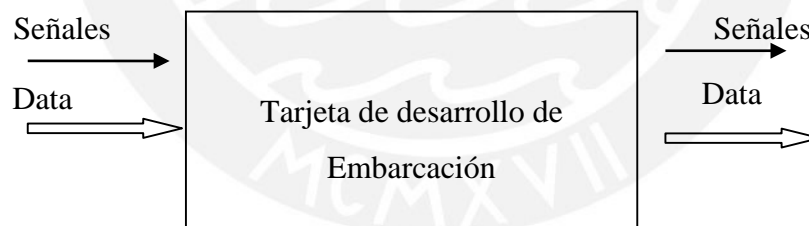


Figura N°12. Caja Negra software.

INGRESO

- Señales: Inicio del programa.
- Data: Ruta de navegación, datos de posición y profundidad.

SALIDA

- Señales: Finalización del programa.
- Data: Datos de posición, data para control de timón.

2.1.3.2 **DESCRIPCIÓN DE FUNCIONES**

- Inicio: Inicio del programa.
- Calculo de Rumbo y distancia: En base a los datos recibidos.
- Comparación de Ruta: comparación de rumbo real con rumbo calculado.
- Fin de programa: apagar el programa.

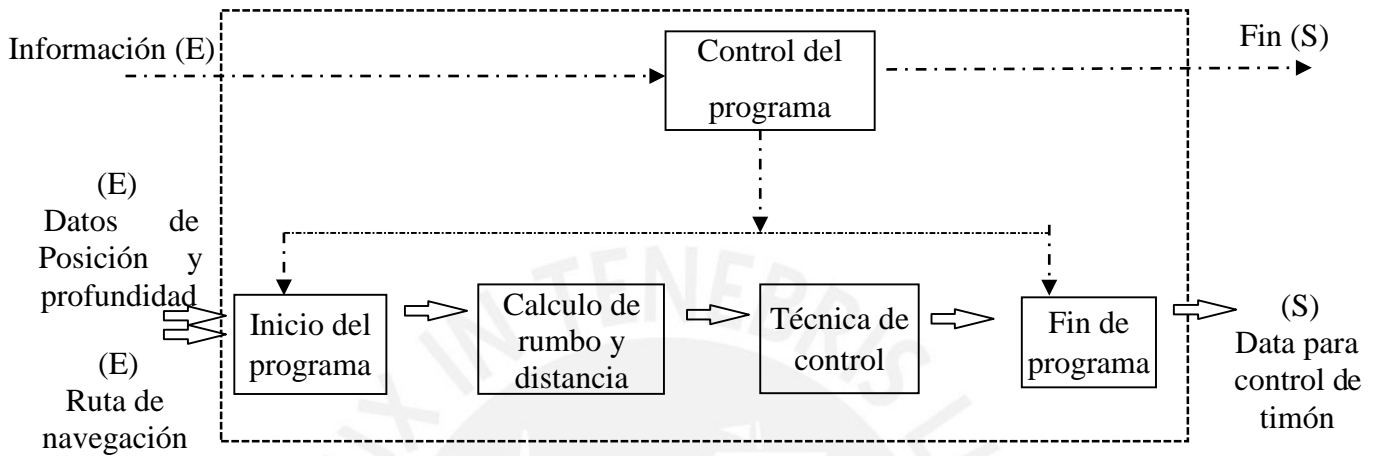


Figura N°13. Estructura de funciones parciales software.

2.1.3.3 **MATRIZ MORFOLOGICA SOFTWARE**

Tabla N°14. Matriz morfológica software.

1	Inicio de programa	Código	
2	Calculo de Rumbo y distancia	Calculos matematicos	
3	Técnica de control	Comparación de rumbos	Algoritmo de control
4	Fin de programa	Código	

1 2

LEYENDA

Concepto solución	Tipo de línea
1	
2	

2.1.3.4 CALIFICACION DEL CONCEPTO DE SOLUCION SOFTWARE

En base a la matriz morfológica se realizara la evaluación del concepto solución software. Se procede a hacer una evaluación tomando valores del 1 al 4, donde:

0: No aceptable 2: Suficiente 4: Muy satisfactorio

1: poco satisfactorio 3: Satisfactorio

Tabla N°15. *Concepto solución software.*

N°1	Criterios Técnico-Económicos	Soluciones	
		1	2
1	Complejidad	3	2
2	Confiabilidad	2	3
3	Lista de exigencias	2	4
4	Facilidad de programación	3	2
	Suma Total	10	11

El concepto solución que tuvo mayor puntaje fue el número 2.

3 PROYECTO PRELIMINAR

3.1 PROYECTO PRELIMINAR MECANICO

Una de las actividades principales en el desarrollo de proyecto preliminar es:

- Configuración y/o forma de la embarcación.
- Material de la embarcación.
- Tipo de propulsión.
- Dimensiones de la embarcación.

Consideraciones principales de material de la embarcación.

Las embarcaciones se pueden construir en los siguientes principales materiales:

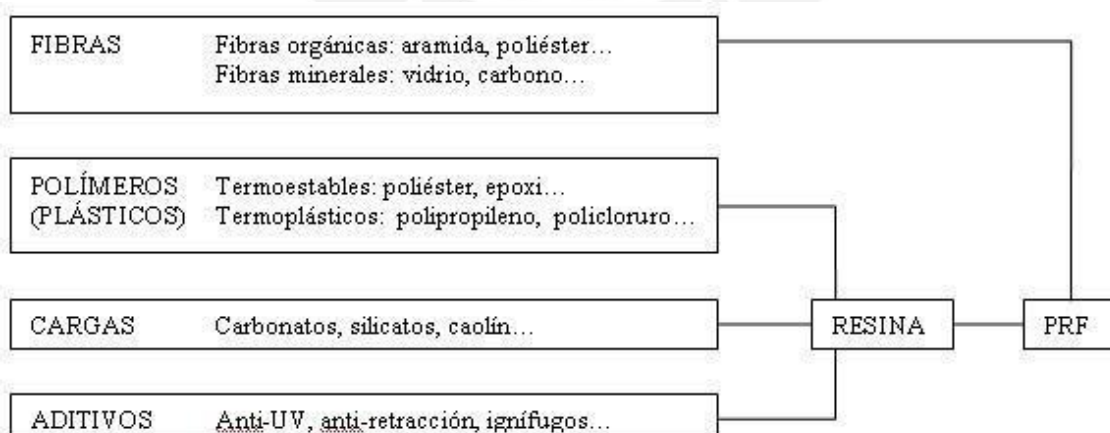
- Madera
- Acero.
- Aluminio.
- Plásticos reforzados de Fibra.

Que es Plastico Reforzado de fibra de Vidrio

El termino PRF se acepta generalmente como significado de plastico reforzado con fibra.

Composición

Los plásticos reforzados con fibras (PRF) están compuestos por un polímero (también llamado la matriz) que junto a las cargas y aditivos forman la resina, y unas fibras determinadas.



Fibra de vidrio

Esta es la fibra más empleada en los PRF, especialmente en aplicaciones industriales, debido a su gran disponibilidad, sus buenas características mecánicas y a su bajo coste.

Existe una gran variedad de fibras de vidrio disponibles en el mercado, en las que priman distintas características, entre las que destacan:

- aislamiento eléctrico
- resistencia química
- alta resistencia mecánica
- elevado módulo de elasticidad
- propiedades dieléctricas
- transparencia a los rayos X

Las principales características de la fibra de vidrio son:

- Alta adherencia fibra-matriz.
- Resistencia mecánica, con una resistencia Específica (tracción/densidad) superior a la del acero.
- Características eléctricas: aislante eléctrico, buena permeabilidad. dieléctrica, permeable a las ondas electromagnéticas.
- Incombustibilidad. No propaga la llama ni origina humos o toxicidad.
- Estabilidad dimensional (bajo coeficiente de dilatación).
- Compatibilidad con las materias orgánicas.
- Imputrescibilidad, insensible a roedores e insectos.
- Débil conductividad térmica (ahorro de calefacción).
- Excesiva flexibilidad.
- Bajo coste.

Fuente: https://es.wikipedia.org/wiki/Pl%C3%A1sticos_reforzados_con_fibras

Comparación con otros materiales de construcción.

Los principales materiales usados para la construcción de embarcaciones son: madera, acero, aluminio, Ferro cemento y FPR. Cada uno tiene un uso óptimo así como cada uno tiene ventajas y desventajas.

La madera es un material tradicional más conocido, pero depende de recursos forestales cada vez más escasos y de carpinteros altamente calificados.

El Aluminio de tipo marino es ligero, durable y requiere una fuerza laboral diestra, mientras que el ferro cemento usa materiales de bajo costo y una gran fuerza laboral, cada uno tiene sus aplicaciones.

El acero es más fácil de obtener que el aluminio, es más fuerte pero se corroe si no está protegido, es el principal material para la construcción de embarcaciones. El PRF es el nuevo en llegar y será comparado con los materiales clásicos en la siguiente tabla.

Tabla N°16. *Comparación de esfuerzos*

Material	Peso específico		Resistencia a la tensión	Resistencia a la compresión	Módulo de elasticidad
	lb/ft ³	ton/m ³	kN/m ² x 10	kN/m ² x 10	kN/m ² x 10
PRF (CMC)	94	1,5	100	100	6
PRF (TM)	105	1,7	240	170	14
Madera: Abeto	42	0,7	55	40	8
" Terciada	40	0,65	16	12	11
Aluminio	170	2,7	120	85	70
Acero	485	7,8	210	190	200

Fuente: Construcción de embarcaciones de fibra de vidrio (FAO).

3.1.1 SOLUCION N°1 DE EMBARCACION N°1

Descripción	Especificaciones	Forma de embarcación
Embarcación de fibra de vidrio propulsada por una hélice tipo avión.	<ul style="list-style-type: none"> • Dimensiones: 0.8 m. x 0.28m. • Propulsión: Motor eléctrico-Hélice tipo avión. • Velocidad: 25m/s. • Diámetro de Hélice: 300mm. 	

3.1.2 SOLUCION DE EMBARCACION N° 2

Descripción	Especificaciones	Forma de embarcación
Embarcación de fibra de vidrio propulsada por una hélice marina.	<ul style="list-style-type: none"> • Dimensiones: 1.0 m. x 0.35m. • Propulsión: 02 Motor eléctrico-Hélice marina. • Velocidad: 25m/s. • Diámetro de Hélice: 30m. 	

3.1.3 SOLUCION DE EMBARCACION ALTERNATIVA 3

Descripción	Especificaciones	Forma de embarcación
Embarcación de tubos de plástico propulsada por 02 hélices marinas.	<ul style="list-style-type: none"> • Dimensiones: 0.80 m. x 0.6m. • Propulsión: 02 Motor eléctrico-Hélice marina. • Velocidad: 25m/s. • Diámetro de Hélice: 30mm. 	

3.1.4 SELECCIÓN DE ALTERNATIVA ÓPTIMA

VALORACIÓN TÉCNICA

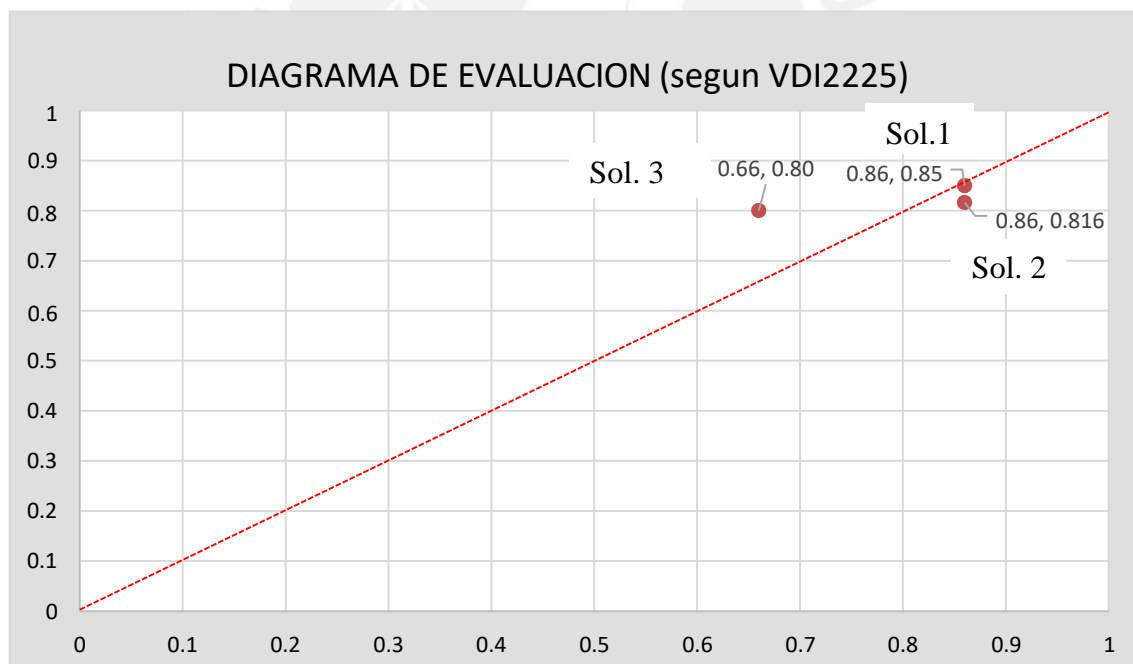
Tabla N°17. *Evaluación técnica de solución mecánica.*

Proyecto: Navegación y orientación de una pequeña embarcación autónoma para medición de profundidad.								
P: Puntaje de 0 a 4 (Escala de valores según VDI 2225) 0= No satisface, 1=Aceptable a las Justas, 2=Suficiente, 3=Bien								
G: El peso ponderado está en función de los criterios de evaluación.								
CRITERIOS DE EVALUACIÓN PARA DISEÑOS EN FASE DE CONCEPTOS O PROYECTOS								
Variantes de concepto			Solución 1		Solución 2		Solución 3	
Nº	Criterios de evaluación	g	P	GP	P	GP	P	GP
1	Función	20	3	0.60	3	0.60	3	0.60
2	Forma	5	3	0.15	3	0.15	2	0.10
3	Diseño	5	3	0.15	3	0.15	2	0.10
4	Seguridad	5	3	0.15	3	0.15	3	0.15
5	Fabricación	15	2	0.30	2	0.30	3	0.45
6	Montaje	5	3	0.15	3	0.15	2	0.10
7	Uso	5	3	0.15	3	0.15	2	0.10
8	Mantenimiento	10	3	0.30	2	0.20	2	0.20
9	Costo	15	2	0.30	2	0.30	2	0.30
10	Plazos	15	2	0.30	2	0.30	2	0.30
Puntaje máximo $\sum P \text{ ó } \sum GP$		100	25	2.55	23	2.45	24	2.40
Puntaje Unitario $PU = \sum GP / 3$				0.85		0.816		0.8
Orden				1		2		3

VALORACION ECONOMICA

Tabla N°18. Evaluación económica de solución mecánica.

Proyecto: Navegación y orientación de una pequeña embarcación autónoma para medición de profundidad.								
P: Puntaje de 0 a 4 (Escala de valores según VDI 2225) 0= No satisface, 1=Aceptable a las Justas, 2=Suficiente, 3=Bien								
G: El peso ponderado está en función de los criterios de evaluación.								
CRITERIOS DE EVALUACIÓN PARA DISEÑOS EN FASE DE CONCEPTOS O PROYECTOS								
Variantes de concepto			Solución 1		Solución 2		Solución 3	
N°	Criterios de evaluación	G	P	GP	P	GP	P	GP
1	Costo de material	30	3	0.90	3	0.90	2	0.6
2	Costo de fabricación	50	2	1.0	2	1.0	2	1.0
3	Costo de montaje	20	3	0.60	3	0.60	2	0.40
Puntaje máximo $\sum P$ ò $\sum GP$		100	25	3.0	23	2.60	24	2.00
Puntaje Unitario $PU = \sum GP / 3$				0.86		0.86		0.66
Orden				1		2		3



Según los resultados las soluciones óptimas son la solución 1 y solución 2.

3.2 PROYECTO PRELIMINAR ELECTRONICO

Una de las actividades en el desarrollo del presente proyecto es determinar la configuración de la electrónica de la embarcación.

3.2.1 SOLUCION N° 1 – ELECTRONICA

El control se realiza por la tarjeta de desarrollo, el control de dirección por el servomotor (timón), la propulsión por motor c.c. La adquisición de datos de posición y profundidad se realizaran por el GPS y la ecosonda.

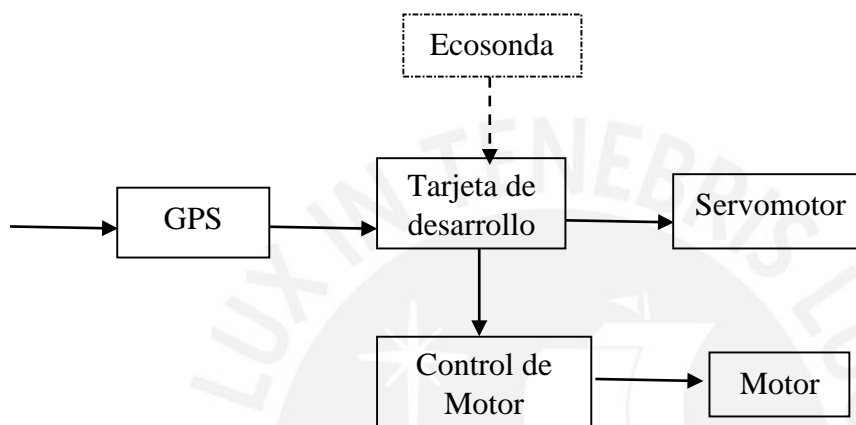


Figura N°14. Solución N°1-Electrónica

3.2.2 SOLUCION N° 2 - ELECTRONICA

El control se realiza por la tarjeta de desarrollo, el control de dirección y la propulsión por los dos motores de c.c. La adquisición de datos de posición y profundidad se realizaran por el GPS y la ecosonda.

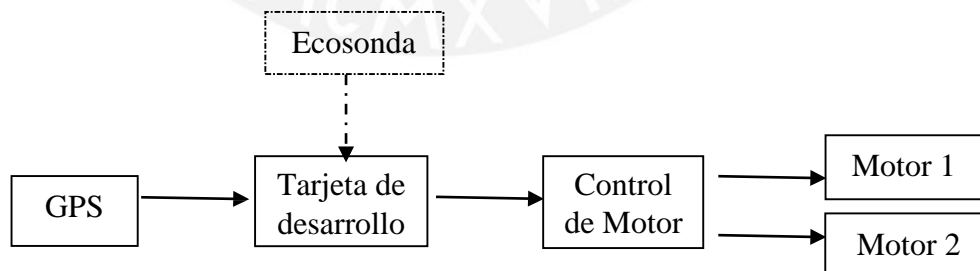


Figura N°15. Solución N°2-Electrónica

3.2.3 SOLUCION N° 3 – ELECTRONICA

El control se realiza por la tarjeta de desarrollo, el control de dirección por el servomotor (timón), la propulsión por motor c.c. La adquisición de datos de posición y profundidad se realizaran por el DGPS y la Ecosonda hidrográfica.

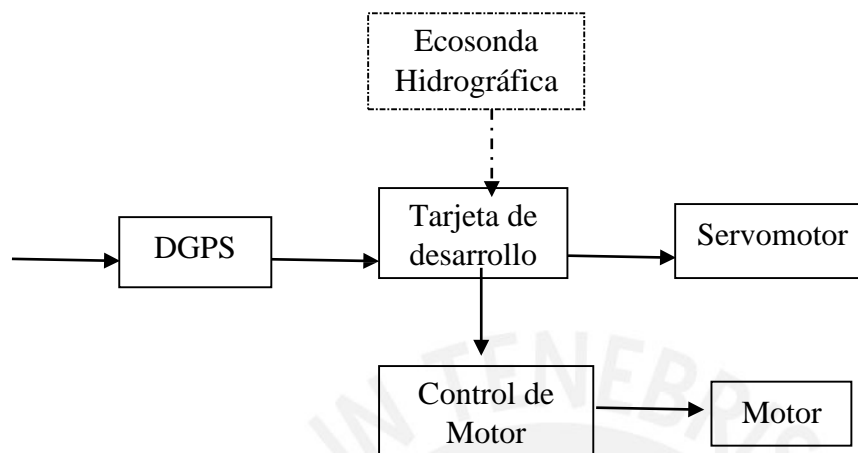


Figura N°16. Solución N°3-Electronica.

3.2.4 SELECCIÓN DE ALTERNATIVA ÓPTIMA.

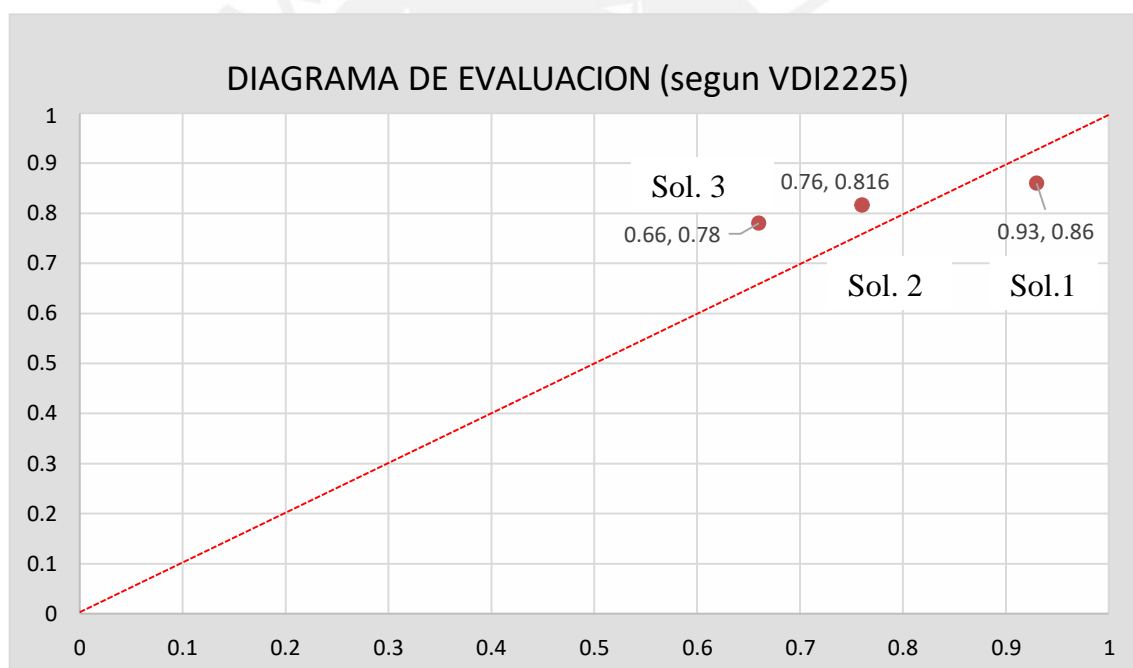
VALORACION TECNICA

Tabla N°19. Evaluación técnica de solución electrónica.

Proyecto: Navegación y orientación de una pequeña embarcación autónoma para medición de profundidad.								
P: Puntaje de 0 a 4 (Escala de valores según VDI 2225) 0= No satisface, 1=Aceptable a las Justas, 2=Suficiente, 3=Bien								
G: El peso ponderado está en función de los criterios de evaluación.								
CRITERIOS DE EVALUACIÓN PARA DISEÑOS EN FASE DE CONCEPTOS O PROYECTOS								
Variantes de concepto			Solución 1		Solución 2		Solución 3	
Nº	Criterios de evaluación	g	P	GP	P	GP	P	GP
1	Función	20	2	0.40	2	0.40	3	0.60
2	Forma	5	3	0.15	3	0.15	2	0.10
3	Diseño	5	2	0.10	2	0.10	3	0.15
4	Seguridad	5	2	0.10	2	0.10	3	0.15
5	Fabricación	15	3	0.45	2	0.30	2	0.30
6	Montaje	5	3	0.15	2	0.15	2	0.10
7	Uso	5	3	0.15	3	0.15	2	0.15
8	Mantenimiento	10	2	0.20	2	0.20	2	0.20
9	Costo	15	3	0.45	3	0.45	2	0.30
10	Plazos	15	3	0.45	2	0.45	2	0.30
Puntaje máximo $\sum P$ ó $\sum GP$		100	25	2.60	23	2.45	24	2.35
Puntaje Unitario $PU = \sum GP/3$				0.86		0.816		0.78
Orden				1		2		3

VALORACION ECONOMICATabla N°20. *Evaluación económica de solución electrónica.*

Proyecto: Navegación y orientación de una pequeña embarcación autónoma para medición de profundidad.								
P: Puntaje de 0 a 4 (Escala de valores según VDI 2225) 0= No satisface, 1=Aceptable a las Justas, 2=Suficiente, 3=Bien								
G: El peso ponderado está en función de los criterios de evaluación.								
CRITERIOS DE EVALUACIÓN PARA DISEÑOS EN FASE DE CONCEPTOS O PROYECTOS								
Variantes de concepto			Solución 1		Solución 2		Solución 3	
Nº	Criterios de evaluación	G	P	GP	P	GP	P	GP
1	Costo de material	30	3	0.90	3	0.90	2	0.6
2	Costo de fabricación	50	3	1.5	2	1.0	2	1.0
3	Costo de montaje	20	2	0.40	2	0.40	2	0.40
Puntaje máximo $\sum P \text{ ó } \sum GP$		100	25	2.8	23	2.30	24	2.00
Puntaje Unitario $PU = \sum GP / 3$				0.93		0.76		0.66
Orden				1		2		3



Según los resultados las soluciones optimas están entre la solución 1, seguido por la solución 2.

4 PROYECTO DEFINITIVO

En esta sección se presentara la solución obtenida, para cumplir con los objetivos propuestos.

4.1 ESTRUCTURA DE EMBARCACIÓN

➤ *Embarcación*

- Se desarrollara y fabricara una embarcación de 0.8 m. de largo y 0.30 m. de ancho.
- La estructura principal de la embarcación será de fibra de vidrio.
- En la parte interior albergara los componentes electrónicos y energía (baterías).
- La propulsión será por medio de una hélice tipo avión.

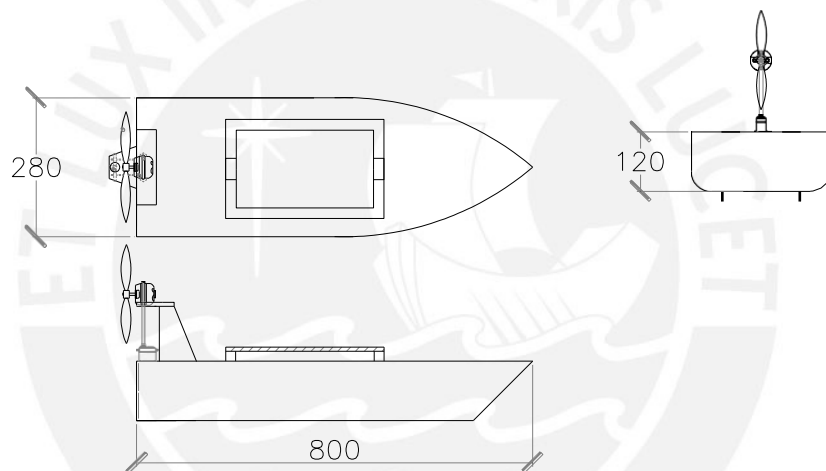


Figura N°17. Estructura de la embarcación.

➤ *Tarjeta de desarrollo*

La embarcación y el manejo de los periféricos se realizaran por medio de una placa especializada en control de vehículos autónomos denominado ArduPilot perteneciente a la familia de Arduino y el lenguaje de programación Arduino.

➤ *Posicionamiento*

La ubicación de la embarcación en la tierra (obtener las coordenadas de latitud y longitud) se utilizara un GPS.

➤ *Medición de profundidad*

La medición de profundidad se realizará por una ecosonda de pesca tipo Garmin o similar.

➤ **Propulsión de embarcación**

La propulsión de la embarcación se realizara por medio de una hélice tipo avión accionada por un motor.

➤ **Dirección de navegación**

La dirección de navegación de la embarcación se realizara por medio de un timón accionado por un servomotor.

➤ **Energía de propulsión**

La energía será proporcionada por una batería de 12 v. 4AH. (Batería del tipo usado en motocicletas).

➤ **Diagrama bloques de conexión**

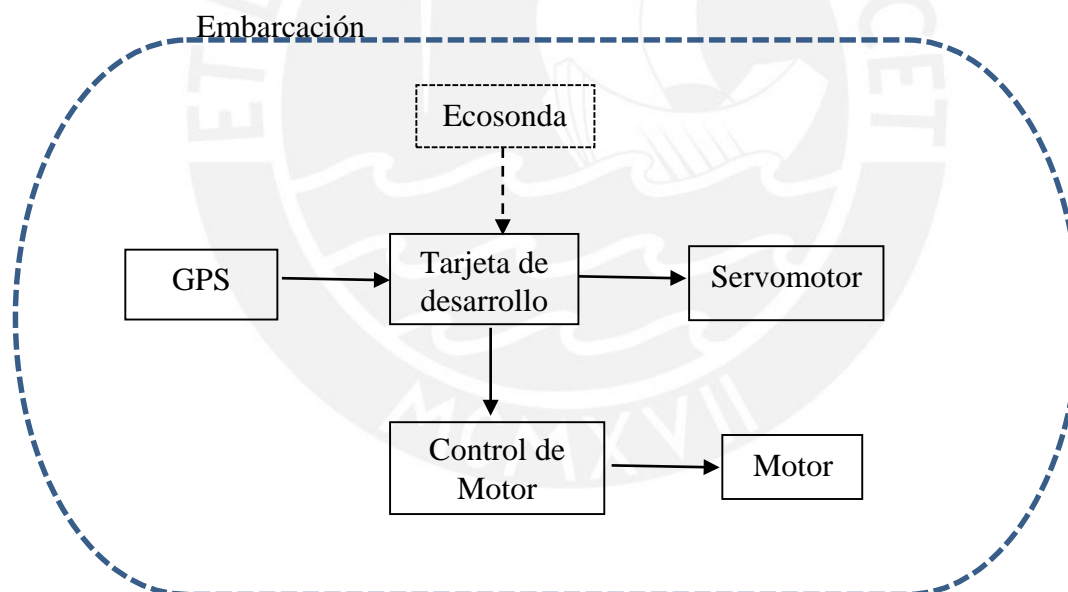


Figura N°18. Diagrama de bloques de componentes electrónicos.

4.2 DESARROLLO DEL PROYECTO

4.2.1 TARJETA DE DESARROLLO

4.2.1.1 ARDUINO

Arduino fue inventado en el año 2005 por el entonces estudiante del instituto IVRAE Massimo Banzi, quien, en un principio, pensaba en hacer Arduino por una necesidad de aprendizaje para los estudiantes de computación y electrónica del mismo instituto [10]. En la Actualidad Arduino es una plataforma de electrónica de código abierto (open – source) basada en hardware y software flexibles y fáciles de usar. [10].

4.2.1.2 SELECCIÓN

Para la selección se tendrá en cuenta las siguientes consideraciones:

- Compatible con el entorno Arduino.
- Piloto automático para el control de un vehículo no tripulado.
- Costo.
- Facilidad de adquisición.

Entre los principales proveedores de controladores tenemos los siguientes:.

- 3DR Pixhawk
- The Raspberry Pi.
- ArduPilot.

Por las consideraciones enunciadas anteriormente se seleccionó la tarjeta de desarrollo compatible con Arduino denominado ArduPilot.

4.2.1.3 ARDUINO ARDUPILOT BASICO

Hecho por Sparkfun Electronics y también producido por Shenzhen Shanghai Technology Ltd.

- Basada en el entorno Arduino.
- Piloto automático para el control de un vehículo no tripulado.
- Interactúa con otros dispositivos informáticos de entrada y salida
- Basado en el Microcontrolador ATmega328.

4.2.1.4 DESCRIPCION TECNICA DE ARDUPILOT BASICO

Es una plataforma que tiene todos los elementos necesarios para controlar un vehículo por GPS y es compatible con Arduino.

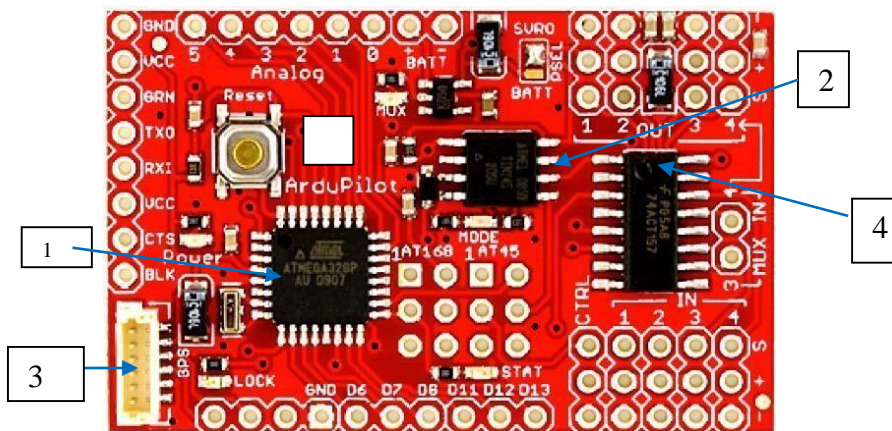


Figura N°19. Board Ardupilot [11].

La plataforma Ardupilot está basado en los tiene los siguientes componentes:

- Un microcontrolador AVR ATmega328 de 8 bit (1).
- Un microcontrolador ATtiny AVR para conmutación entre R/C y un control autónomo (2).
- Interface para GPS (3).
- Interface para RC servo y control electrónico de velocidad ESC (4).
- Un dispositivo de control de diferentes operaciones que puede cambiar el control entre el control RC y el microcontrolador.

PINOUT DE ARDUPILOT.

Port	Analog pin	Usage
PD 0		USART RX
PD 1		USART TX
PD 2		Ext Int 0
PD 3		PWM T2B, Ext Int 1
PD 4		
PD 5		PWM T0B
PD 6		PWM T0A
PD 7		
PB 0		Input Capture
PB 1		PWM T1A
PB 2		PWM T1B
PB 3		PWM T2A, MOSI
PB 4		MISO
PB 5		SCK
PC 0	0	
PC 1	1	
PC 2	2	
PC 3	3	
PC 4	4	I2C SDA
PC 5	5	I2C SCL

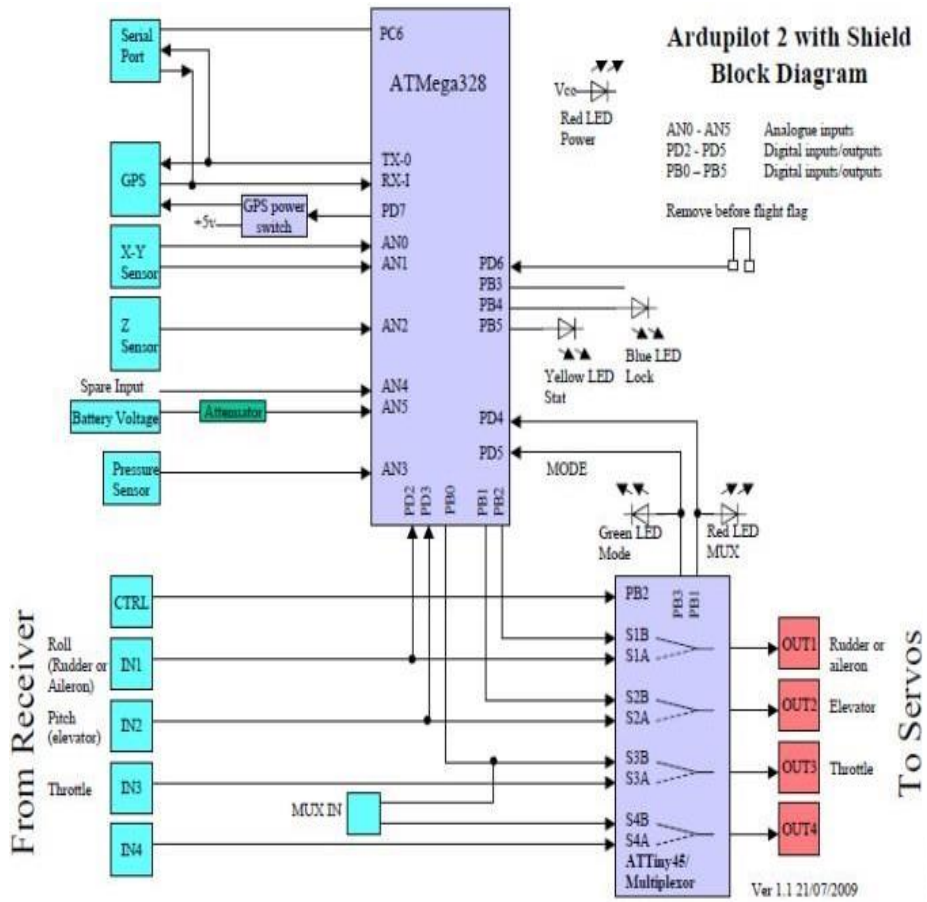


Figura N°20. Diagrama de bloques del Autopilot [12].

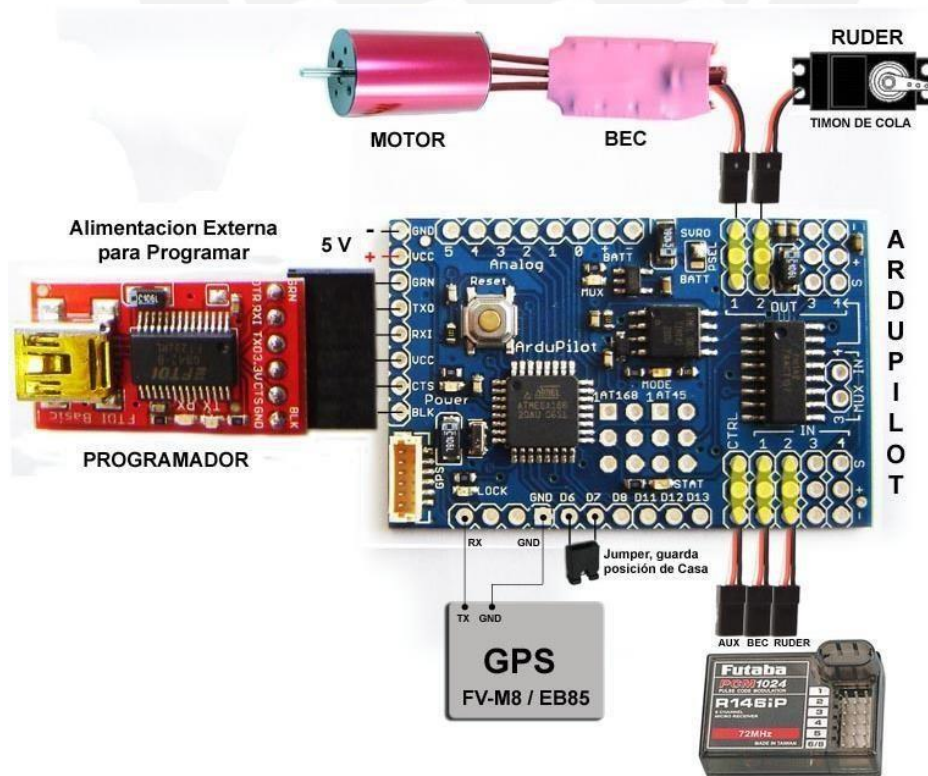


Figura N°21. Vista de Conexiones Ardupilot [14].

4.2.2 SISTEMA DE POSICIONAMIENTO GLOBAL - GPS

El sistema GPS está constituido por 24 satélites que orbitan la tierra y utilizan como mínimo 04 satélites para determinar la ubicación y altura en cualquier parte del globo terrestre. La posición tiene una precisión métrica y/o submétrica dependiendo de las características del equipo GPS receptor.

a.- SELECCIÓN GPS

Para la selección del GPS se tendrá en cuenta las siguientes consideraciones:

Precisión

El GPS debe tener una precisión métrica $\pm 1\text{m}$. a $\pm 2\text{m}$.

Compatibilidad con Ardupilot

Dado que el algoritmo de control se implementa en Ardupilot, debe ser compatible.

Voltaje de alimentación

El Arduino tiene dos tensiones de salida: 3,3 a 5 Vdc. que serán tomados en cuenta en la selección.

Costo de adquisición

Se tendrá en cuenta el coste de adquisición sin descuidar las características anteriores.

En base a las consideraciones anteriores se seleccionó el GPS EM-406.

b.- MODULO GPS EM-406A

Módulo GPS basado en el chip EM-406A de USGlobalSat. Éste módulo GPS es extremadamente pequeño ya que apenas mide unos 3x3cm y tan sólo pesa 16 gramos.

CARACTERÍSTICAS TÉCNICAS

- Peso: 16 g.
- Canales: 20.
- Extremadamente sensible: -159dBm.
- Margen de error posicional de 10 metros (5m con WAAS).
- Señal de salida: NMEA 0183 y protocolo SiRF binario.
- Dimensiones: 30mm x 30mm x 10.5mm.

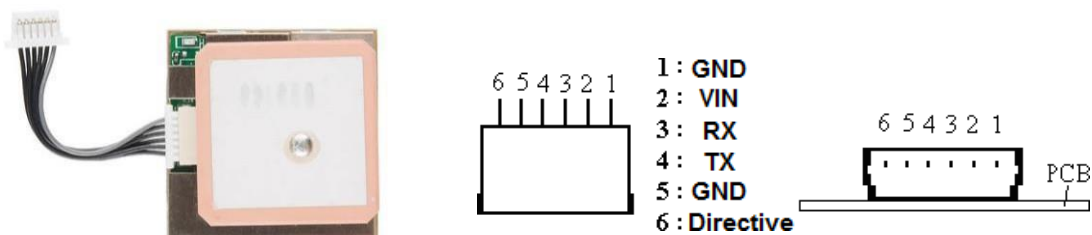


Figura N°22. Módulo de GPS EM-406 y distribución de pines [15].

DESCRIPCIÓN DE LOS PINES

- VIN (entrada de corriente continua): Este es el suministro principal de CC para un 4.5V ~ 6.5 de alimentación de CC.
- TX: Este es el principal canal transmite para la salida de la navegación y medición de los datos en el software o del usuario de software escrito navegación del usuario.
- RX: Este es el principal canal de recepción para recibir comandos de software para el motor, tablero de software SiRFdemo o desde el software escrito del usuario.
- GND: proporciona las bases para la junta del motor. Conecte todos los terrenos.
- PPS: Este pin provee un pulso por segundo de salida de la placa del motor que es sincronizado al tiempo GPS.

4.2.2.1 MÓDULO DE POTENCIA

Proporciona un módulo de alimentación de 2.25A a 5.37V.

El módulo está diseñado para convertir la energía de la batería principal (máximo hasta 18 voltios). A voltajes de 5.37 V.



Figura N°23. Vista de un módulo de potencia 3DR (PM) [16].

4.2.3 MOTOR BRUSHLESS Y HELICE DE PROPULSION

Se utilizara un motor BrushLess para realizar la propulsión de la embarcación por medio de una hélice.

a. -Dimensionamiento y Selección de Motor eléctrico BrushLess y hélice

Calculamos el empuje producido por la hélice.

La Ecuación (1) brinda el empuje basado en la teoría del momento [17].

$$T = \frac{\pi}{4} D^2 \rho v \Delta v \quad (\text{Ecu. 4.1})$$

Dónde: T = empuje [N].

D = diámetro de la hélice [m] = 0.30 m.

v = velocidad del aire a la hélice = 5 m/seg.

Δv = Velocidad del aire acelerado por la hélice = 30 m/seg.

ρ = Densidad del aire [1.225kg/m³].

T = 12.98 [N] fuerza de empuje.

El par motor puede ser calculado.

$$P = \frac{T \Delta v}{2} \quad (\text{Ecu. 4.2})$$

P = 194.7 N-m/seg = 194.7 watt.

Ahora elegimos la hélice del siguiente gráfico que nos muestra las características de la hélice Clásica de Aeronáutica 12" x 7".

Aeronaut Classic Glas 12" x 7"

10250rpm Profile E387 H/D0.7 0.51 Blade angle adjust 0°

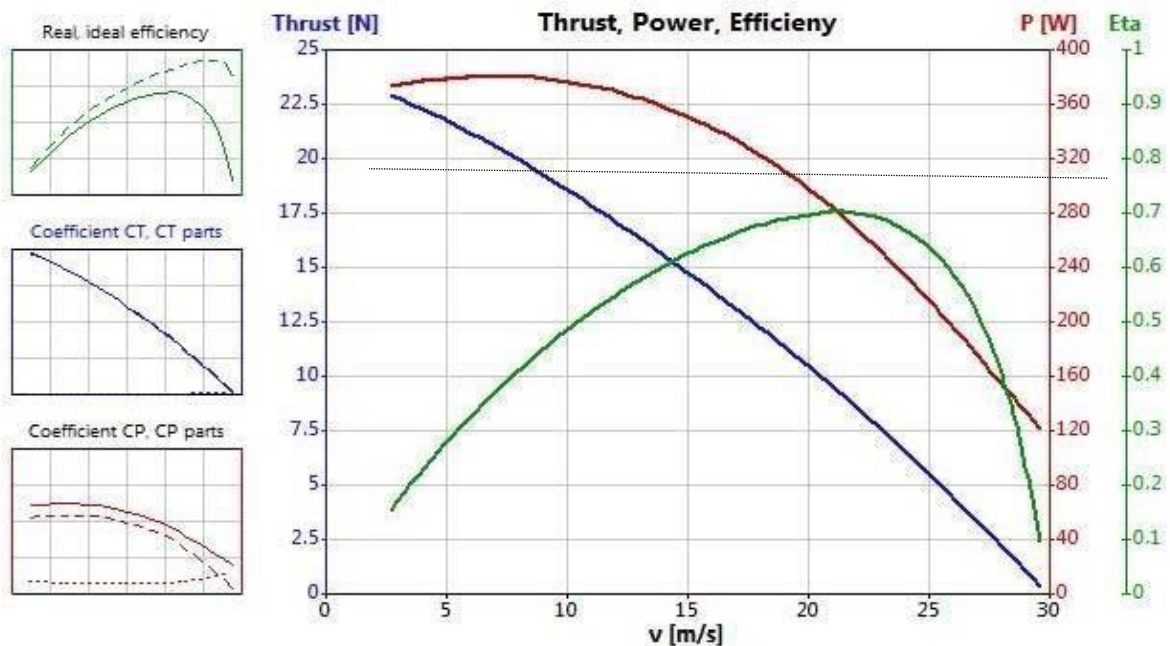


Figura N°24. Fuerza de empuje, Potencia y Velocidad de hélice Aeronáutica clásica 12"x7" [18].

En la figura anterior se observa que la Hélice clásica Aeronáutica 12" x 7" a una velocidad 10250 RPM satisface nuestro requerimientos de Fuerza de empuje (Thurst) y Potencia.

Los motores de corriente continua BrushLess se selecciona en función Kv (RPM/v) por lo cual calcularemos el Kv del motor para nuestra embarcación

Para 12V de tensión se tiene:

$$Kv \text{ motor} = 10250 \text{ RPM} / 12\text{V} = 854 \text{ kv}$$

Para 9V de tensión se tiene:

$$Kv \text{ motor} = 10250 \text{ RPM} / 9\text{V} = 1138 \text{ kv}$$

El motor BrushLess a utilizar será de 1000Kv (RPM/v).

Tabla N°21. *RPM ideal de especificaciones de motores*

Motor	Kv(rpm/v)	Max Rpm	Ideal Rpm
NTM Prop Drive Series 35-30 1100kv	1100	12210	3053
RCX A2830-11 1000kV	1000	11320	2830
NTM Prop Drive Series 3548 900kv	900	9990	2498
TowerPro Brushless Outrunner 2410-08T	890	9879	2470
D4023-850 OutRunner Motor	850	9435	2359
hexTronik DT750 Brushless Outrunner 750kv	750	8325	2081
Turnigy G15 Brushless Outrunner 710kv	710	7881	1970
Turnigy G25 Brushless Outrunner 610kv	610	6771	1693

De la *Tabla 21* se selecciona el motor RCX A2830-11 1000 kV.

Las principales características del motor serán:

- KV (rpm/v): 1000.
- Max Power: 210W.
- Max Current: 14.5A.
- Max Pull (Approx.): 890g.

- Suitable ESC (A): 30A

Ahora en función al motor seleccionado, amperaje y voltaje utilizado usaremos un regulador electrónico de velocidad recomendado para este motor.

4.2.4 REGULADOR ELECTRÓNICO DE VELOCIDAD DE 30AMP MARCA RC TIMER Y BEC 5V. Y 2 A.

NOTA: BEC, son las siglas de Battery Eliminator Circuit. Es un circuito electrónico diseñado para entregar energía eléctrica a otros circuitos sin la necesidad de múltiples baterías [19].



Figura N°25. ESC 30A Brushless Motor Speed Controller [20].



Figura N°26. Motor RCX A2830-11 1000kV [21].

Resumen de resultados de selección de Motor, hélice y controlador electrónico de velocidad

De los cálculos realizados se seleccionó el siguiente equipamiento:

Motor	:	RCX A2830-11 1000 Kv.
Helices	:	10 x 4.5 CF.
ESC	:	30A. Brushless Motor Speed Controller.

4.2.5 SERVOMOTOR

Con el fin de accionar el timón trabajo se utilizara un servomotor estándar que tendrá la función de hacer girar el timón de la embarcación.

Calculo de torque:

$$\text{Torque} = \text{Fuerza} \times \text{distancia.}$$

Primeramente tendremos que hallar que fuerza del flujo de aire actúa el timón:

Según el valor hallado en la fuerza que ejercerá la hélice es de 12.98 N (1.32 kg-f) y el timón tiene que vencer esta fuerza poder actuar sobre la dirección de la embarcación.

Calculando el torque para un brazo de palanca de 0.05m.

$$\text{Torque de timón} = 1.32 * 0.05$$

$$\text{Torque de timón} = 0.066 \text{ kg-f} \quad \text{m} = 6.6 \text{ kgf-cm.}$$



Figura N°27. Servo Motor standard [22].

Se selecciona un servomotor Estándar con un torque mínimo de 6.6 Kgf-cm.

4.2.6 ESTRUCTURA DE LA EMBARCACION

a.- Diseño de casco de la embarcación

Para obtener una menor resistencia del agua al avance de la embarcación es necesario que el casco de la embarcación tenga menor pérdida por fricción al contacto con el agua, Por lo cual el diseño consiste en lograr ese objetivo.

Software Delft Ship

El Software DELFTship se especializa en el desarrollo de software marino desarrollado por la Universidad de Tecnología de Delft.

El Software DELFTship es un modelador de casco de una embarcación, en la que se elige el tipo de embarcación, la principales dimensiones eslora x manga (largo Ancho) y calado (Altura de contacto con el agua de la embarcación) [23].

Este Programa se puede descargar gratuitamente de la página <http://www.delftship.net/>.

Introducción de datos principales

- Eslora de Diseño=0.80 m.
- Manga de diseño=0.28m.
- Calado de diseño=0.05 m.
- Espesor de material 0.002 m.
- Modelo de Embarcación: Tipo Bote.

Resultados:

El Software genera un informe hidrostático, Informe de cálculo de resistencia y dimensiones graficas de la embarcación.

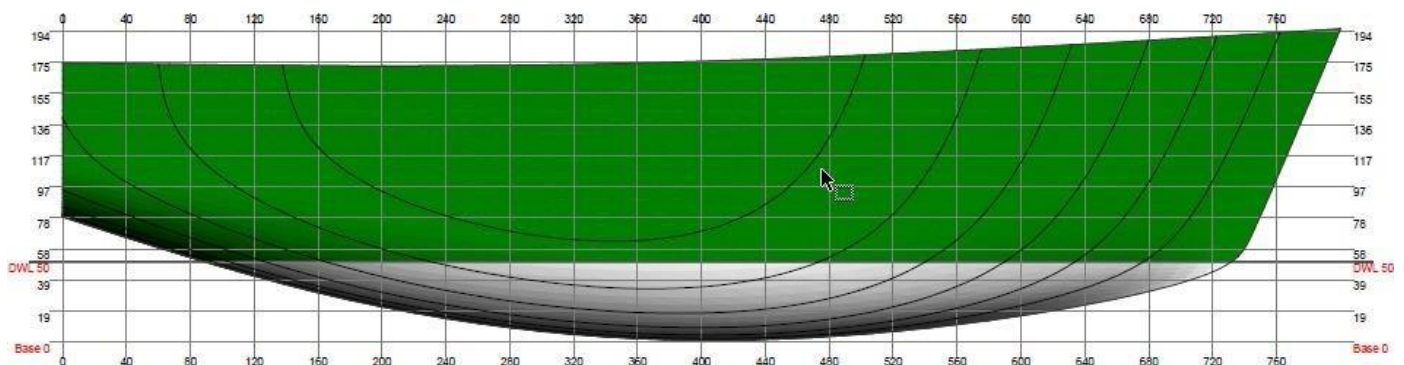


Figura N°28. Vista lateral de dimensiones de la embarcación y el calado de diseño.

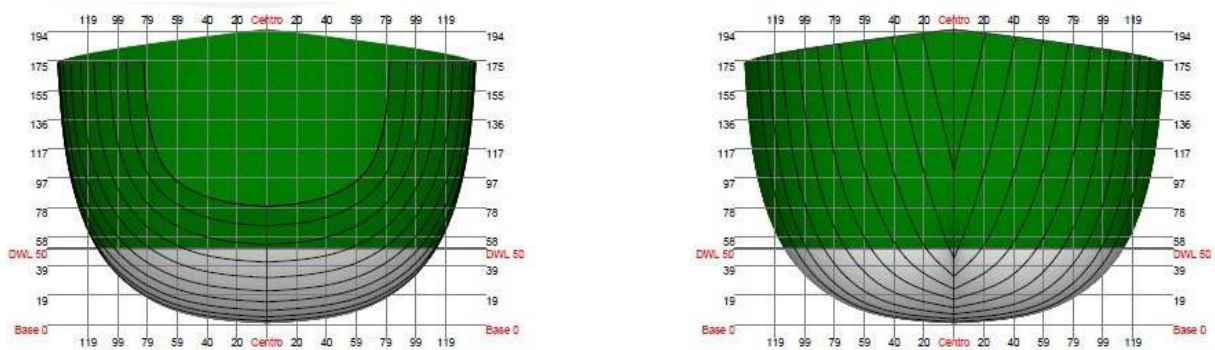


Figura N°29. Vista frontal y posterior y el calado de diseño.

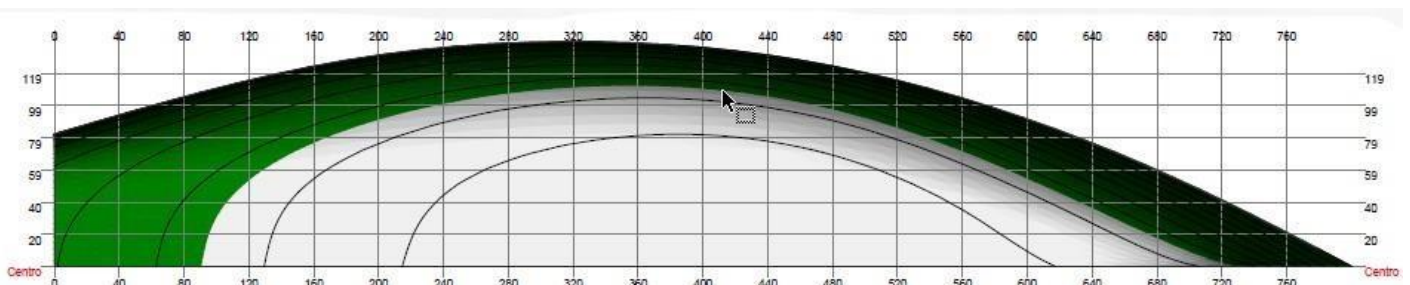


Figura N°30. Vista del fondo y el calado de diseño.

Informe hidrostático del diseño.

Diseñador

Creado por

Comentario

Nombre de archivo

Modelo 0.80x0.fbm

Eslera de diseño 0.800 (m)

Posición de la de sección media

0.400 (m)

Eslera total 0.800 (m)

Relative water density

1.025

Manga de diseño 0.280 (m)

Grosor medio del forro

0.0000 (m)

Manga total 0.277 (m)

Coefficiente de apéndice

1.0000

Calado de diseño 0.050 (m)

Propiedades del volumen		Propiedades del plano de flotación	
Moulded volume	0.003 (m ³)	Eslera en línea de agua	0.641 (m)
Volumen desplazado	0.003 (m ³)	Manga en la línea de agua	0.224 (m)
Desplazamiento	0.003 (tonnes)	Ángulo de entrada	11.213 (Degr.)
Coefficiente de bloque	0.2448	Área del plano de flotación	0.099 (m ²)
Coefficiente prismático	0.4248	Coefficiente del plano de flotación	0.4413
Coefficiente prismático vertical	0.5547	Centro del plano de flotación	0.371 (m)
Superficie del área mojada	0.114 (m ²)	Momento de inercia transversal	0.000 (m ⁴)
Centro de flotación longitudinal	0.387 (m)	Momento de inercia longitudinal	0.002 (m ⁴)
Centro de flotación longitudinal	-2.095 %		
Centro de flotación vertical	0.033 (m)		

Propiedades de la de sección media		Estabilidad inicial	
Área de la de sección media	0.008 (m ²)	Altura metacentro transversal	0.140 (m)
Coefficiente de la sección media	0.5762	Altura del metacentro longitudinal	0.801 (m)

Plano lateral	
Área lateral	0.021 (m ²)
Centro de esfuerzo longitudinal	0.414 (m)
Centro de esfuerzo vertical	0.030 (m)

Las siguientes propiedades de capa fueron calculadas para ambos lados del buque

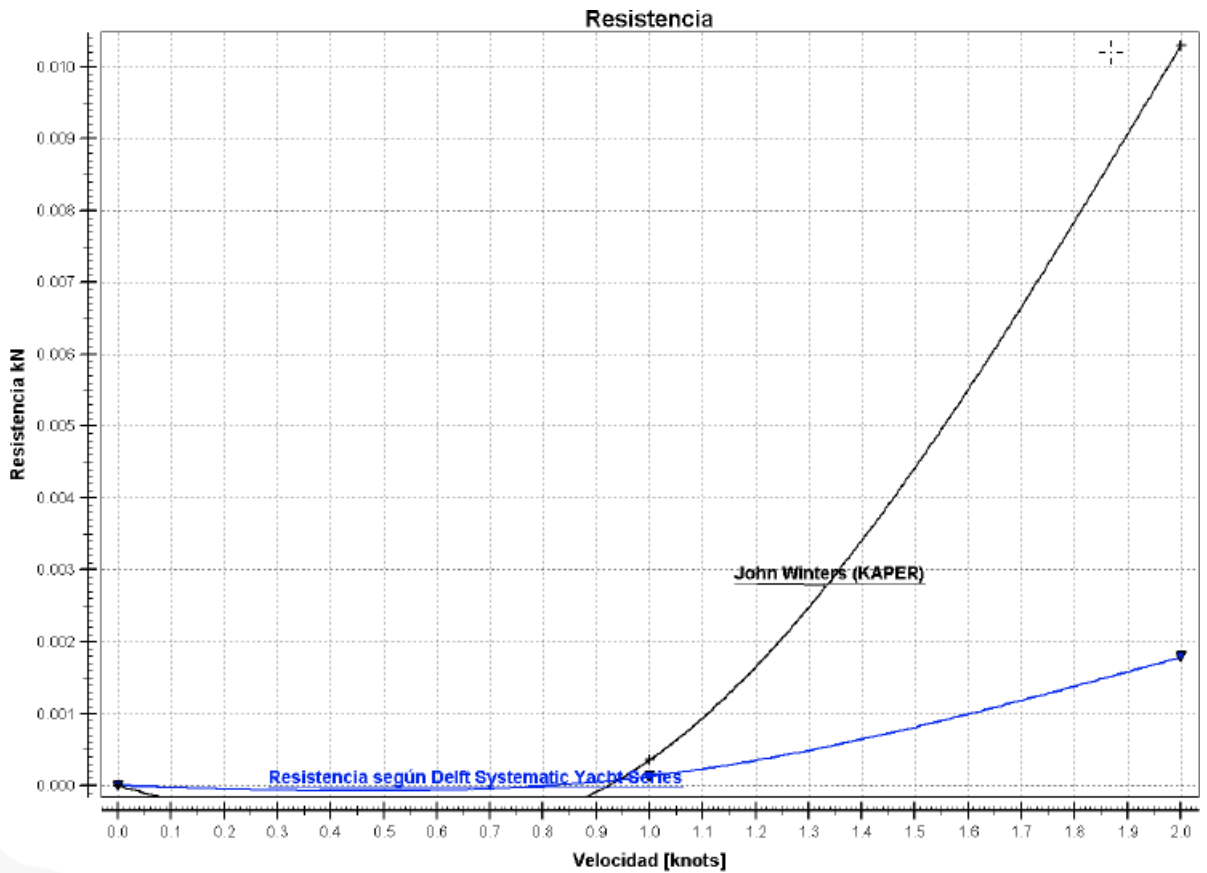


Figura N°31. Grafica Velocidad vs Resistencia de la embarcación.

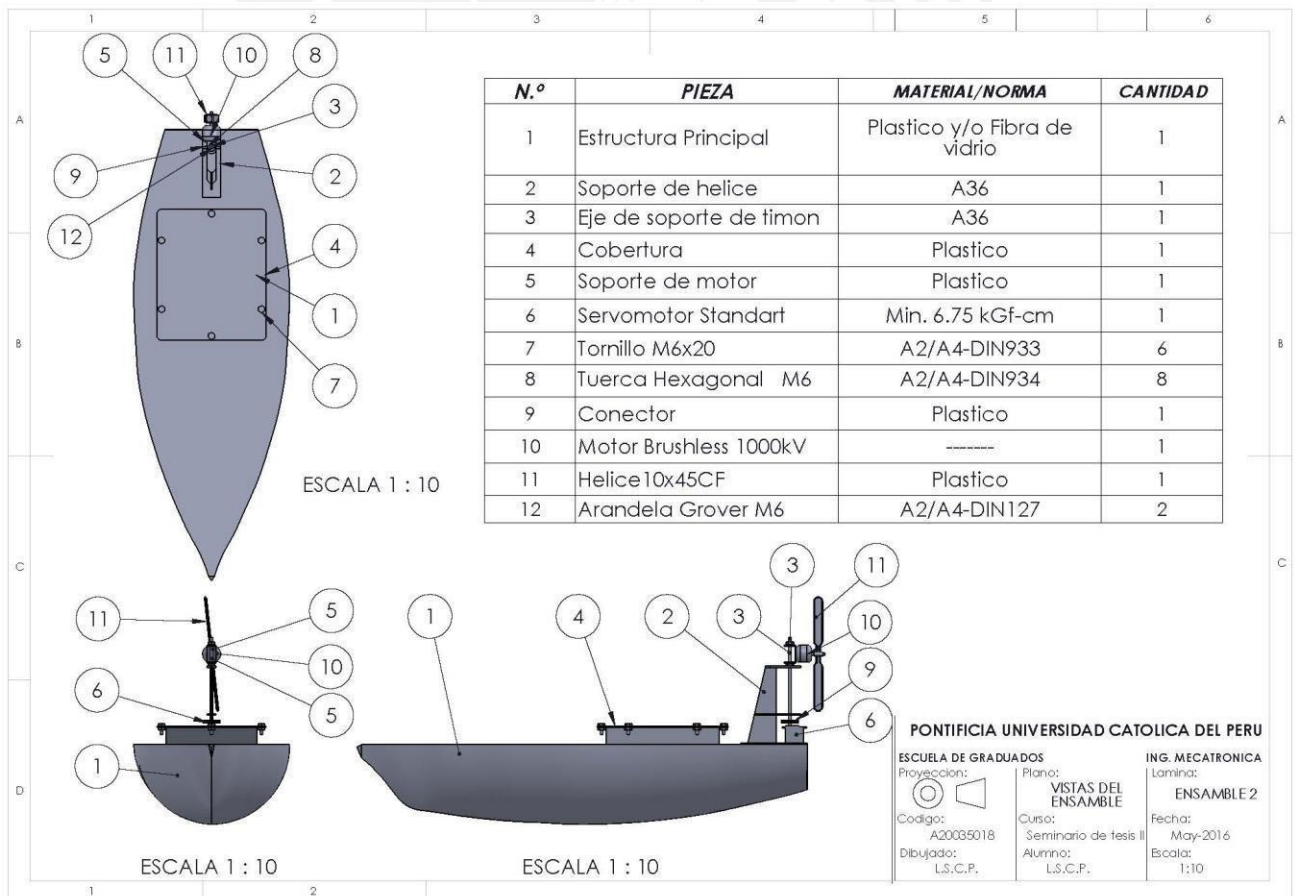


Figura N°32. Vista de láminas generadas a partir de las dimensiones calculadas previamente.

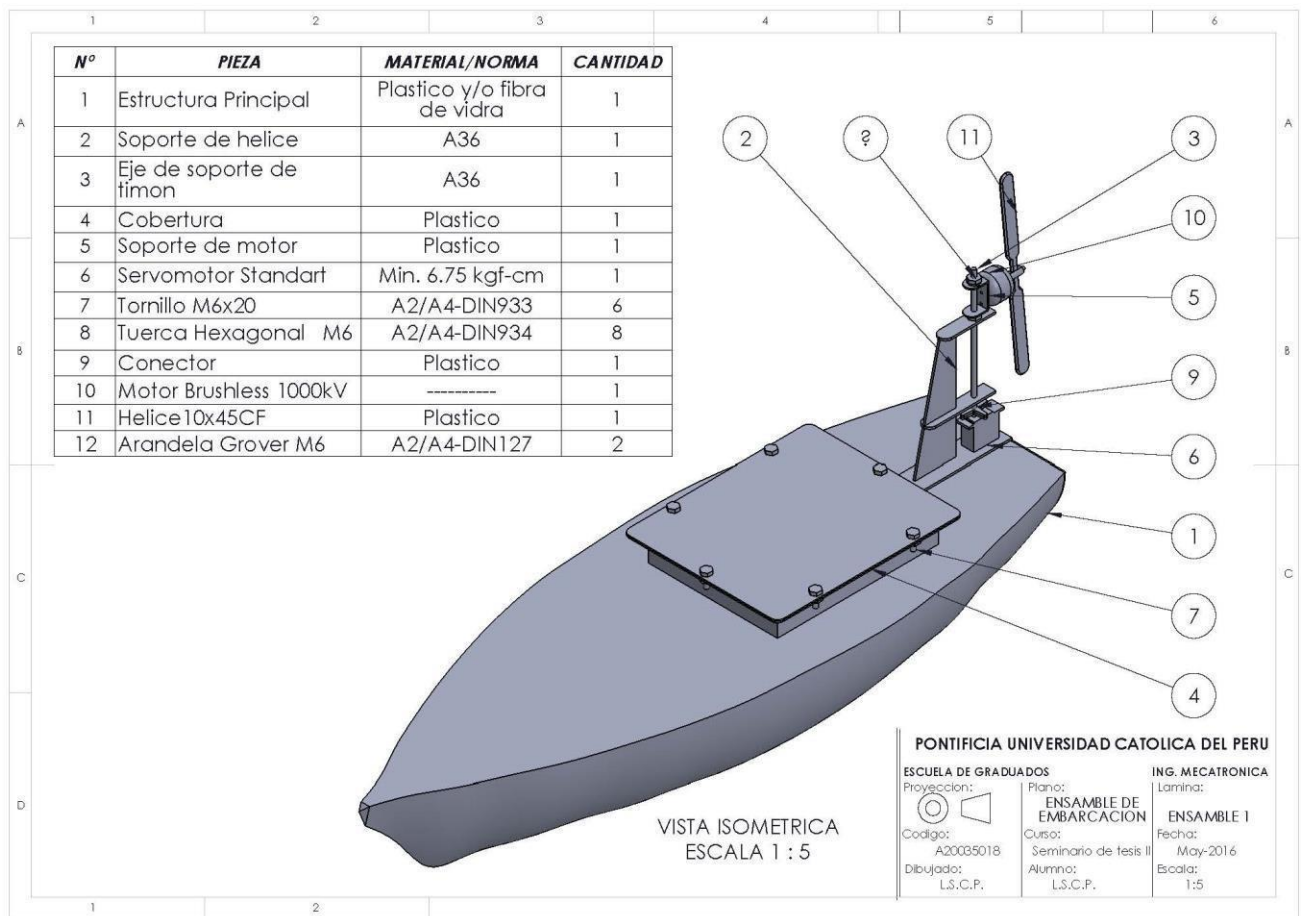


Figura N°33. Otra vista de láminas generadas a partir de las dimensiones calculadas previamente.

5 MODELO DE EMBARCACION Y DISEÑO DE PILOTO AUTOMATICO

5.1 MODELO LINEAL DE EMBARCACIÓN DE SUPERFICIE

El sistema de coordenadas en la que se enmarca una embarcación es el siguiente:

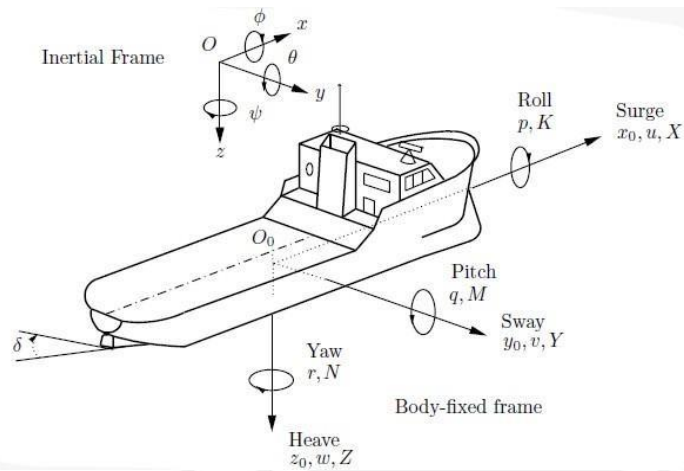


Figura N°34. Notación estándar y las convenciones de signo para la descripción del movimiento de una embarcación (SNAME, 1950) [24].

Tabla N°1. Notación de SNAME (1950) para embarcaciones [24].

DOF		Forces and moments	Linear and angular velocities	Positions and Euler angles
1	motions in the x direction (surge)	X	u	x
2	motions in the y direction (sway)	Y	v	y
3	motions in the z direction (heave)	Z	w	z
4	rotation about the x axis (roll, heel)	K	p	ϕ
5	rotation about the y axis (pitch, trim)	M	q	θ
6	rotation about the z axis (yaw)	N	r	ψ

Traslación	Fuerza	Velocidad lineal	Posición
Avance	X	u	x
Desp. lateral	Y	v	y
Arfada	Z	w	z
Rotación	Momento	Velocidad angular	Angulo
Balanceo	K	p	ϕ
Cabeceo	M	q	θ
Guiñada	N	r	ψ

5.1.1 MODELO NEWTONIANO

Se asume:

$\vec{v} = \{u, v, w\}$ Es el vector velocidad, donde:

$u =$ Velocidad a lo largo del eje de embarcación – eje x .

$v =$ Velocidad de oscilación en el eje y .

$w =$ Velocidad vertical al eje z .

Además:

$\vec{r}_G = \{x_G, y_G, z_G\}$ Ubicación del centro de masa en coordenadas.

$\vec{\omega} = \{p, q, r\}$ Es el vector rotación en coordenadas locales.

$\vec{F} = \{X, Y, Z\}$ Es el vector que representa las fuerzas externas actuantes en la embarcación.

Consideraremos movimientos sólo en el plano horizontal, que significa $\theta = \varphi = p = q = w = 0$. Puesto que el buque es simétrico sobre el plano $x - z$, $y_G = 0$; z_G es inconsecuente. Entonces tenemos [25]:

$$X = m \left(\frac{\partial u}{\partial t} - rv - x_G \frac{\partial r}{\partial t} \right) \quad (\text{Ecu. 5.1})$$

$$Y = m \left(\frac{\partial v}{\partial t} - ru - x_G \frac{\partial r}{\partial t} \right) \quad (\text{Ecu. 5.2})$$

$$N = I_{zz} \frac{\partial r}{\partial t} + mx_G \left(\frac{\partial u}{\partial t} + ru \right) \quad (\text{Ecu. 5.3})$$

Que $u = U + u$, donde $U \gg u$ ($U = \sqrt{u^2 + v^2}$) y la eliminación de términos de orden superior, este sistema es: [25]

$$X = m \left(\frac{\partial u}{\partial t} \right) \quad (\text{Ecu. 5.1})$$

$$Y = m \left(\frac{\partial v}{\partial t} - rU - x_G \frac{\partial r}{\partial t} \right) \quad (\text{Ecu. 5.2})$$

$$N = I_{zz} \frac{\partial r}{\partial t} + mx_G \left(\frac{\partial u}{\partial t} + rU \right) \quad (\text{Ecu. 5.3})$$

Un número de coeficientes puede ser descontado, como se señaló en el párrafo anterior, considerando un mar homogéneo, sin ninguna corriente, ola, o los efectos del viento, $\{X_x X_y X_\emptyset Y_x Y_y Y_\emptyset N_x N_y N_\emptyset\}$ son todos cero.

Considerando X_v : es una fuerza longitudinal tienen el mismo signo sin tener en cuenta el signo de la v (debido a la simetría del casco de lado a lado), debe tener cero de pendiente con v en el origen.

Se supondrá que ninguna fuerza hidrodinámica depende de la posición de la embarcación, en segundo lugar, considerar X_v : dado que esta fuerza longitudinal tienen el mismo signo sin tener en cuenta el signo de la v (debido a la simetría del casco de lado a lado), debe tener pendiente cero con el origen de v . Por lo tanto $X_v = 0$. Este mismo argumento demuestra que $\{X_r X_v X_r Y_u Y_u N_u N_u\} = 0$.

Por último, puesto que la aceleración de partículas de fluido se relaciona linealmente con la presión o fuerza, no consideramos términos de aceleración no lineal, o derivados de tiempo mayor. Cabe señalar que algunos términos no lineales relacionados con los que hemos eliminado anteriormente no son cero.

Por ejemplo, $Y_{uu} = 0$ debido a la simetría del casco, pero en general $X_{vv} = 0$ sólo si el vaso es proa-popa son simétricas. Teniendo en cuenta sólo las condiciones hidrodinámicas lineales tenemos [25]:

$$\langle m - X_u \rangle u = X_u u + X' \quad (\text{Ecu. 5.4})$$

$$\langle m - Y_v \rangle v + \langle mx_G - Y_r \rangle r = Y_v v + \langle Y_r - mU \rangle r + Y' \quad (\text{Ecu. 5.5})$$

$$\langle mx_G - N_v \rangle v + \langle I_{zz} - N_r \rangle r = N_v v + \langle N_r - mx_G U \rangle r + N' \quad (\text{Ecu. 5.6})$$

Los términos de la derecha son las fuerzas impuestas por la hélice (s) y el timón(s) $\{X', Y', N'\}$.

Con el vector estado $\bar{s} = \{v, r\}$ y fuerza/momento externos vector $\bar{F} = \{Y', N'\}$, es una representación de espacio de estado del sistema [25]:

$$\begin{bmatrix} m - Y_v & mx_G - Y_r \\ mx_G - N_v & I_{zz} - N_r \end{bmatrix} \frac{d\bar{s}}{dt} = \begin{bmatrix} Y_v & Y_r - mU \\ N_v & N_r - mx_G U \end{bmatrix} \bar{s} + \bar{F} \quad (\text{Ecu. 5.7})$$

$$\mathbf{M} \dot{\bar{s}} = \mathbf{N} \bar{s} + \mathbf{F}$$

$$\dot{\bar{s}} = \mathbf{M}^{-1} \mathbf{N} \bar{s} + \mathbf{M}^{-1} \mathbf{F}$$

$$\dot{\bar{s}} = \mathbf{A} \bar{s} + \mathbf{B} \bar{F} \quad (\text{Ecu. 5.8})$$

La matriz \mathbf{M} es matriz de masa o de inercia, que siempre es invertible. La última forma de la ecuación en la que \mathbf{A} representa la dinámica interna del sistema y \mathbf{B} es una matriz de ganancia para las entradas de control y disturbios.

Desafortunadamente, muchos de estos términos son muy difíciles de medir por lo cual una forma de uso frecuente en ingeniería de control para obtener un modelo de un sistema es analizar la respuesta de paso.

En nuestro caso deseamos conocer la variación del ángulo de guiñada de la embarcación (rotación sobre el eje z). Obtendremos la respuesta es decir la variación del Angulo de guiñada, cuando la entrada es una fuerza de propulsión.

5.1.2 MODELOS NAMOTO

Controladores de dirección basado en el modelo para embarcaciones marinas se basan generalmente en la representación del modelo de Nomoto, (1957). El modelo del piloto automático de Nomoto se puede derivar el modelo lineal de maniobras, tal como se explica a continuación.

5.1.2.1 *MODELO DE SEGUNDO ORDEN NOMOTO (SUBSISTEMA DE ORIENTACIÓN)*

En 1957 Nomoto propuesta funciones de la transferencia dos basado en modelos de simulación de dirección y control [26], es decir, el segundo modelo de Nomoto de orden y el primer modelo de Nomoto orden. El segundo modelo de Nomoto orden se muestra en la ecuación (5.9).

$$G_1(s) = \frac{r(s)}{\delta(s)} = \frac{K_0(sT_3+1)}{(1+sT_1)(1+sT_2)} \quad (\text{Ecu. 5.9})$$

Este modelo muestra el efecto dinámico de ángulo de desviación del timón en desvío tasa r. Se observa que casi uno de los polos es cancelada por su cero [17]. Con este hecho, ecuación (5.9) se simplifica aún más en la ecuación (5.10). Se llama el primer modelo de Nomoto orden.

$$G_2(s) = \frac{r(s)}{\delta(s)} = \frac{K}{(1+s\tau)} \quad (\text{Ecu. 5.10})$$

$$\text{Siendo: } \tau = \tau_1 + \tau_2 - \tau_3$$

Finalmente, $\psi = r$

$$G_2(s) = \frac{\psi(s)}{\delta(s)} = \frac{K}{s(1+s\tau)} \quad (\text{Ecu. 5.11})$$

Cuál es la función de transferencia que se utiliza en sistemas de piloto automático más comerciales.

Modelo de Nomoto es válido en el supuesto de que el barco se mueve a velocidad constante hacia delante, la fuerza propulsora es constante y el ángulo del timón es pequeño.

5.1.2.2 *ANALISIS DE LOS MODELOS*

Hasta ahora hemos introducido dos modelos: la derivación siguiendo la ley de movimiento de Newton y el modelo simplificado lineal de Nomoto. El primero es conocido más matemáticamente mientras que el segundo es más simple en su forma.

El primer modelo para obtener los parámetros se debe obtener en forma experimental realizados en países del primer mundo que poseen laboratorios en las que se puede simular la embarcación en modelo a escala, es un proceso costoso y de mucho tiempo antes de obtener un modelo útil.

El Modelo de Nomoto, por el contrario, es simple y sus pocos parámetros del modelo son fáciles de identificar en los experimentos. Esto hace atractivo en el diseño de control modelo de Nomoto. Sin embargo, el modelo de Nomoto es sólo para el movimiento de guiñada (rotación en el eje z).

5.2 SISTEMA DE IDENTIFICACION

5.2.1 SISTEMA DE REFERENCIA

Las variables que describen el movimiento del modelo físico en el plano horizontal son:

ψ : Guiñada.

v : Desplazamiento lateral.

u : Velocidad de Avance.

U : Velocidad de Barco.

δ : Angulo de impulsor.

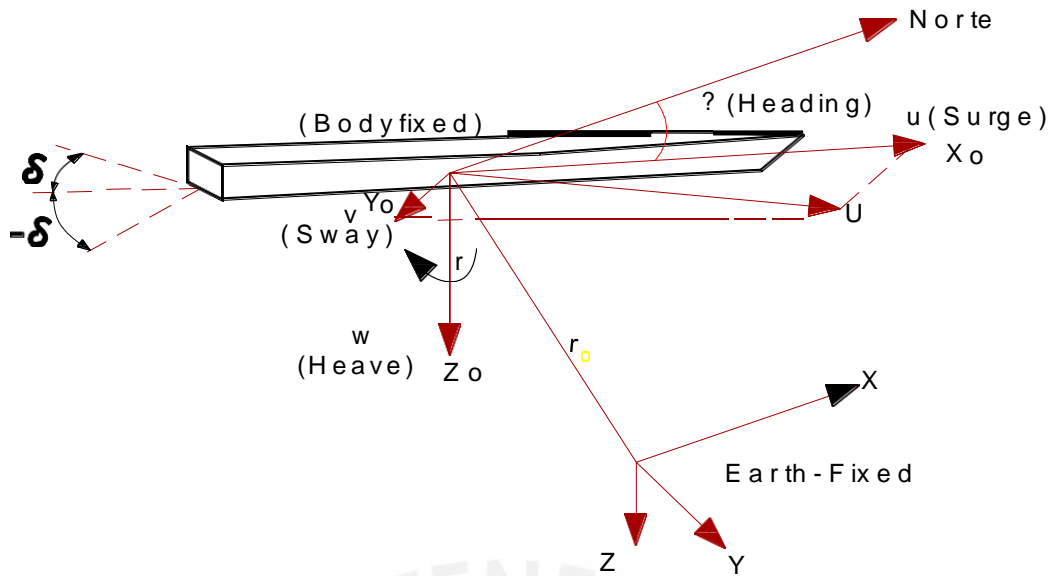
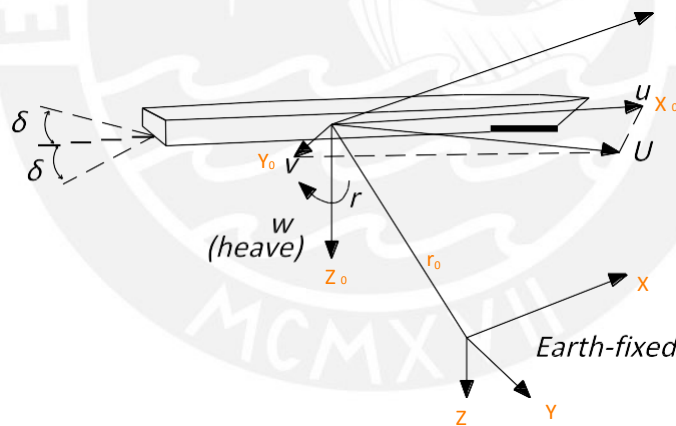


Figura N°35. Las variables que describen el movimiento del modelo físico en el plano horizontal [28].

5.3 CINEMÁTICA DE LA EMBARCACION

Trayectoria de desplazamiento de la embarcación en relación con el sistema de coordenadas fijo de la tierra y se da por una transformación de velocidad.



Modelo Cinemático de la embarcación [28].

$$\eta_1 = J_1(\eta_2) v_1 \tag{Ecu. 5.12}$$

Donde:

$$\eta_1 = [x, y, z]^T$$

$$\eta_2 = [\phi, \theta, \psi]^T$$

$$\eta_3 = [u, v, w]^T$$

$$J_1(\eta_2) = \begin{pmatrix} c\psi c\theta & -s\psi c\theta + c\psi s\theta s\phi & s\psi s\theta + c\psi c\theta s\phi \\ s\psi c\theta & c\psi c\theta + s\psi s\theta s\phi & -c\psi s\theta + s\psi c\theta s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{pmatrix} \tag{Ecu. 5.13}$$

Asumiendo la ausencia del balanceo y cabeceo ($\theta = \phi = 0$) las siguientes ecuaciones obtenidas son:

$$x = u \cos\Psi + v \operatorname{sen}\Psi \quad (\text{Ecu. 5.14})$$

$$y = u \operatorname{sen}\Psi - v \cos\Psi \quad (\text{Ecu. 5.15})$$

$$\Psi = r \quad (\text{Ecu. 5.16})$$

5.4 CONTROL DE DIRECCION

5.4.1 CONTROL DE EMBARCACION

El control de la embarcación consiste en mantener un rumbo ante la presencia de perturbaciones.

5.4.2 6.3.1 CÁLCULO DE LA DISTANCIA Y RUMBO

Para calcular la distancia entre dos puntos de posición geográfica debemos emplear un método que relacione las coordenadas de latitud y longitud de cada punto con el radio de la tierra.

5.4.2.1 *CÁLCULO DE DISTANCIA ENTRE PUNTOS*

El cálculo de la distancia se realiza por medio de la fórmula de Haversine [29].

R = radio de la Tierra (radio Medio=6,371 km)

$\Delta\text{lat} = \text{lat}2 - \text{lat}1$

$\Delta\text{long} = \text{long}2 - \text{long}1$

Fórmula de Haversine

$$a = \sin^2(\Delta\text{lat}/2) + \cos(\text{lat}1) \cdot \cos(\text{lat}2) \cdot \sin^2(\Delta\text{long}/2) \quad (\text{Ecu. 5.17})$$

$$c = 2 \cdot \operatorname{atan}2(\sqrt{a}, \sqrt{1-a})$$

$$d = R \cdot c \text{ (distancia)}$$

5.4.2.2 *CÁLCULO DE RUMBO*

Para el cálculo de rumbo se utilizó la siguiente fórmula:

$$\theta = \operatorname{atan}2(\sin(\Delta\text{long}) \cdot \cos(\text{lat}2), \cos(\text{lat}1) \cdot \sin(\text{lat}2) - \sin(\text{lat}1) \cdot \cos(\text{lat}2) \cdot \cos(\Delta\text{long}))$$

.....(Ecu. 5.18)

5.4.3 CONTROLADOR PID

Ecuación de Nomoto de primer orden (Ec. 5.19):

$$\frac{\psi(s)}{\delta(s)} = \frac{K}{s(1+\tau s)} \quad \tau\dot{\psi} + \psi = K\delta$$

Aplicando un controlador PID/PD para el piloto automático. Para asegurar la robustez [30].

Teniendo en cuenta una ley de control de tipo PD, en la forma:

$$\delta = K_p(\psi - \psi_d) - K_d\dot{\psi} \quad (\text{Ecu. 5.20})$$

Donde K_p y K_d son los parámetros de diseño del controlador.

La dinámica de lazo cerrado resultante de la dinámica de la nave y el controlador PD es [30]:

$$T\ddot{\psi} + (1 + Kk_d)\dot{\psi} + Kk_p\psi = Kk_p\psi_d \quad (\text{Ecu. 5.21})$$

Esta expresión corresponde a un sistema de orden 2 de la forma:

$$\ddot{\Psi}(t) + 2\zeta\omega_n\dot{\Psi}(t) + \omega_n^2\Psi(t) = \omega_n^2\Psi_d(t) \quad (\text{Ecu. 5.22})$$

Donde ω_n es la frecuencia natural y ζ es el coeficiente del sistema de amortiguación del sistema en lazo cerrado.

Amortiguación relativa se elige típicamente en el intervalo de $0,8 < \zeta < 1,0$, mientras que la elección de ω_n estará limitada por el ancho de banda resultante del timón ω_δ (rad/seg) y la dinámica de nave $1/T$ (rad/seg) según:

$$\frac{1}{T} < \omega_n \sqrt{1 - 2\zeta^2 + \sqrt{(4\zeta^4 - 4\zeta^2 + 2)}} < \omega_\delta \quad (\text{Ecu. 5.23})$$

Para una embarcación críticamente amortiguada ($\zeta = 1$), el ω_b de ancho de banda de lazo cerrado se relaciona con la frecuencia natural ω_n del sistema lazo cerrado por un factor de $0.64\omega_n$. Los valores de K_p y K_d son dependientes en ω_n y ζ (que son parámetros de diseño), por las siguientes relaciones [30]:

$$K_p = \frac{T\omega_n^2}{K} \quad (\text{Ecu. 5.24})$$

$$K_d = \frac{2T\zeta\omega_n - 1}{K} \quad (\text{Ecu. 5.25})$$

6 SOFTWARE

El presente capítulo se presentara el software utilizado en el desarrollo del presente trabajo.

6.1 SOFTWARE LIBRE

Según la Free Software Foundation (<http://www.fsf.org>), organización encargada de fomentar el uso y desarrollo del software libre a nivel mundial, un software para ser considerado libre ha de ofrecer a cualquier persona u organización cuatro libertades básicas e imprescindibles [31]:

- ✓ **Libertad 0:** la libertad de usar el programa con cualquier propósito y en cualquier sistema informático.
- ✓ **Libertad 1:** la libertad de estudiar cómo funciona internamente el programa, y adaptarlo a las necesidades particulares. El acceso al código fuente es un requisito previo para esto.
- ✓ **Libertad 2:** la libertad de distribuir copias.
- ✓ **Libertad 3:** la libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. El acceso al código fuente es un requisito previo para esto.

Un programa es software libre si los usuarios tienen todas estas libertades. Así pues, el software libre es aquel software que da a los usuarios la libertad de poder ejecutarlo, copiarlo y distribuirlo (a cualquiera y a cualquier lugar), estudiarlo, cambiarlo y mejorarlo, sin tener que pedir ni pagar permisos al desarrollador original ni a ninguna otra entidad específica. La distribución de las copias puede ser con o sin modificaciones propias, y atención, puede ser gratis ¡o no!: el "software libre" es un asunto de libertad, no de precio [31].

6.2 SOFTWARE DEL PROYECTO

6.2.1 SOFTWARE ARDUINO

6.2.1.1 *QUE ES UN IDE*

Las siglas IDE vienen del inglés “*Integrated Development Environment*”, lo que traducido al español, significa Entorno de Desarrollo Integrado. Esto es simplemente una forma de llamar al conjunto de herramientas software que permite a los programadores poder desarrollar (es decir, básicamente escribir y probar) sus propios programas con

comodidad. En el caso de Arduino, necesitamos un IDE que nos permita escribir y editar nuestro programa (también llamado “sketch” en el mundo de Arduino), que nos permita comprobar que no hayamos cometido ningún error y que nos permita, cuando ya estemos seguros de que el sketch es correcto, grabarlo en la memoria del microcontrolador de la placa Arduino para que este se convierta a partir de entonces en el ejecutor autónomo de dicho programa [32].

6.2.1.2 **SOFTWARE ARDUINO IDE**

Para obtener el software Arduino es necesario descargarse de la página web de Arduino el entorno de desarrollo (IDE). Existen versiones para Windows y para MAC, así como las fuentes para compilarlas en LINUX.

En la Figura 36 se muestra el aspecto del entorno de programación. El entorno de desarrollo Arduino está formado por un editor de texto, un área de mensajes, una consola de texto, una barra de accesos directos y un conjunto de herramientas:

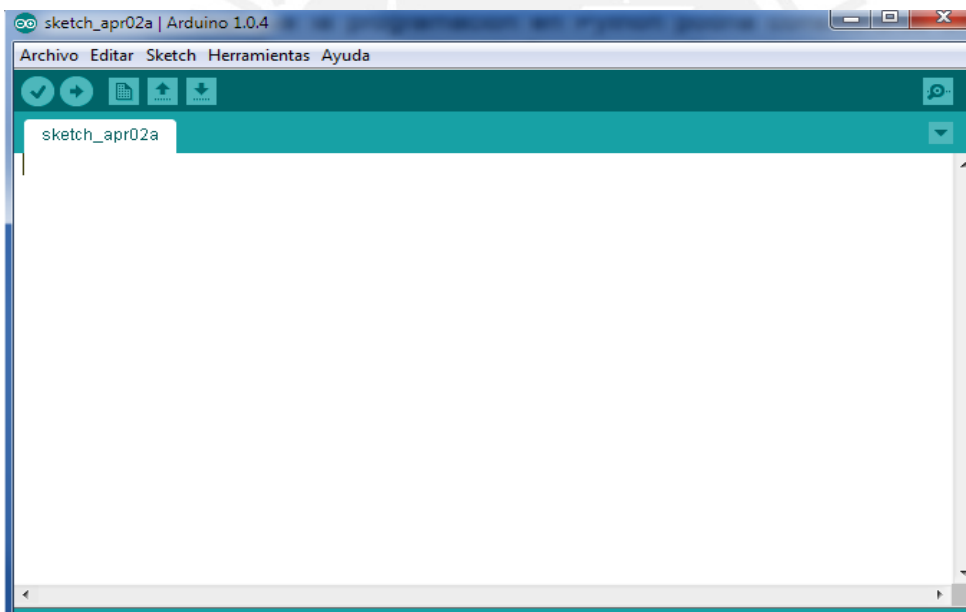


Figura N°36. Vista del software Arduino.

Los programas hechos con Arduino se dividen en tres partes principales: estructura, valores (variables y constantes), y funciones. El Lenguaje de programación Arduino se basa en C/C++.

6.2.1.3 **OBTENCION DE COORDENADAS**

El software utilizado para obtener las coordenadas de posición de la embarcación se utilizara el Google Earth, este programa nos permite ubicar un punto en la tierra en coordenadas geográficas en el sistema WGS84.

a.- GOOGLE EARTH.

Es un programa informático similar a un sistema de información geográfica (SIG), creado por la empresa Keyhole Inc., que permite visualizar imágenes del planeta, combinando imágenes de satélite, mapas y el motor de búsqueda de Google que permite ver imágenes a escala de un lugar específico del planeta.

Uso de Google Earth

En la siguiente imagen se muestra las funciones disponibles en la ventana principal de Google Earth.

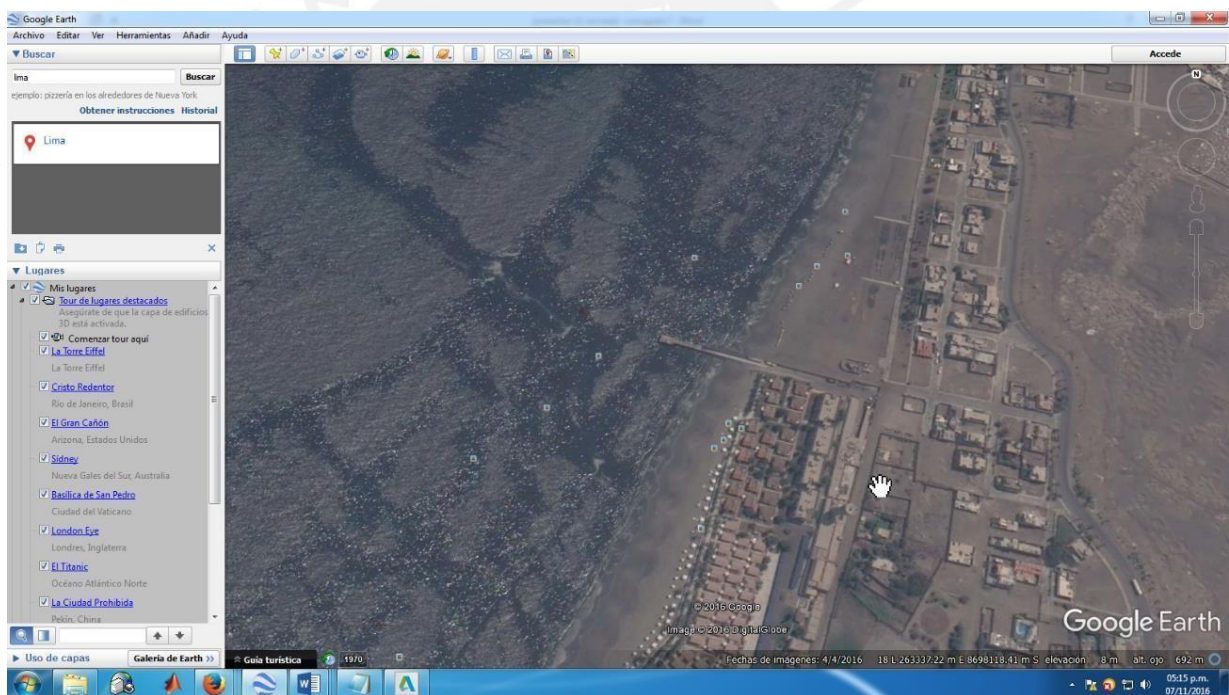



Figura N°37. Software Google Earth [33].

OBTENCIÓN DE COORDENADAS

Para obtener las coordenadas de un punto en el espacio terrestres usando el Google Earth se seguirá lo siguientes pasos.

- a.- abrir el programa Google Earth.
- b.- Usar el panel de búsqueda para buscar la zona del trabajo.

c.- Usar el icono  para marcar una posición y en propiedades de esta posición se indicara las coordenadas de la posición.

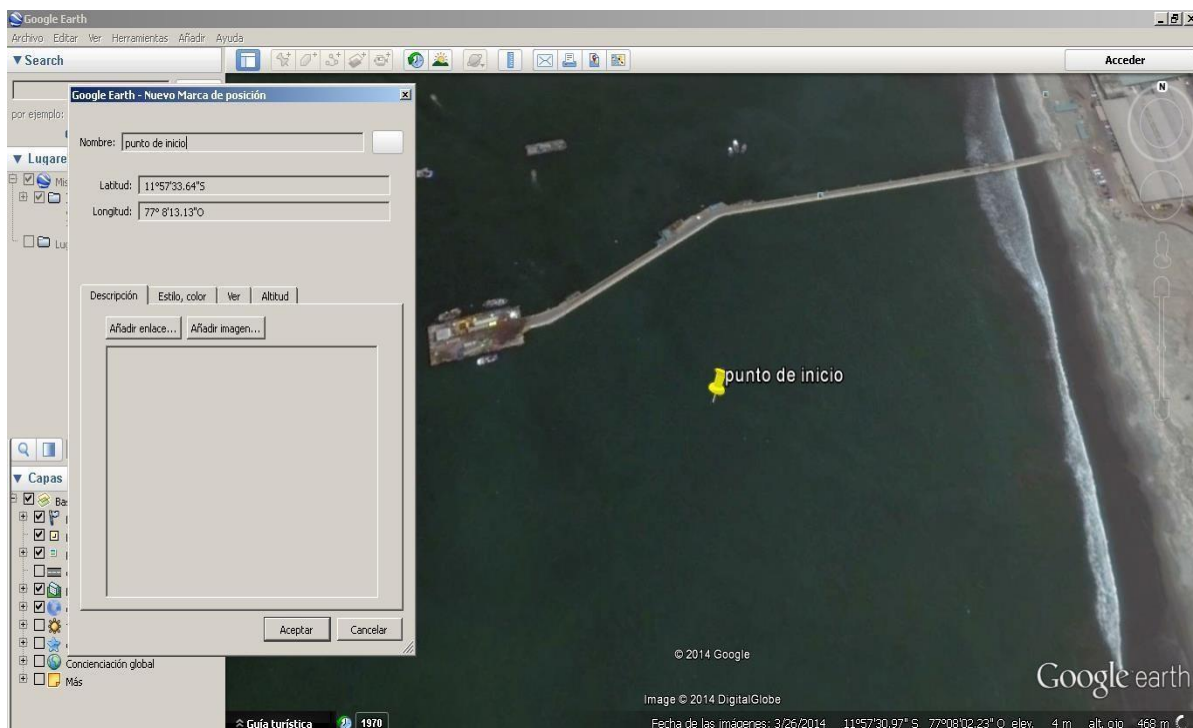


Figura N°38. Obtención de las coordenadas de posición usando del Software Google Earth.

Fuente propia Usando el Software Google Earth online <http://www/google-earth/>

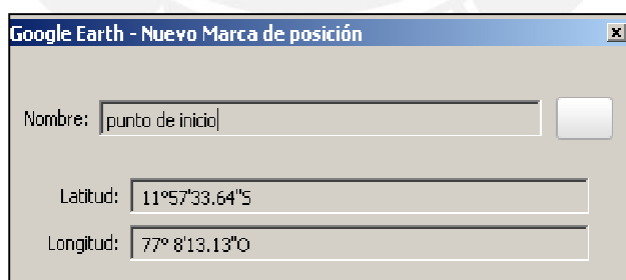


Figura N°39. Vista de las coordenadas de posición de una posición usando del Software Google Earth

Fuente propia Usando el Software Google Earth online <http://www/google-earth/>

7 APLICACIONES PRINCIPALES

7.1 BATIMETRIA

El término "batimetría" procede del griego, el Diccionario de la Real Academia de la Lengua lo define como "el arte de medir las profundidades".

Se entiende por batimetría el levantamiento del relieve de superficies subacuáticas, tanto los levantamientos del fondo de mar, como del fondo de cursos de agua, de embalses etc.

La batimetría tiene como fin la obtención de las coordenadas (X, Y, Z) de todos estos puntos. La parte más compleja y que caracteriza a los diversos métodos de levantamientos batimétricos es la determinación de la profundidad. Esta tarea se denomina operación de sondeo o simplemente sondar. La profundidad de un punto se obtendrá midiendo la distancia vertical entre el nivel del agua y la superficie del fondo [34].

7.2 METODOLOGÍA DE MEDICION

La metodología para obtener un mapa de un embalse o laguna es el siguiente:

Generación de mapas

Para la generación de mapas con curvas de nivel son necesario los datos del fondo, estos datos son la posición (Este, Norte) de cada punto y su correspondiente altura.

Obtención de Puntos georreferenciados.

Para Obtener Los puntos es necesario Obtener los siguientes datos.

- 1.- Referencia con el nivel del agua del embalse o laguna a medir (Z)
- 2.- Medición del calado (h).
- 3.- medición de la posición (x, y).
- 4.- análisis y tratamiento de datos.

NIVEL, Z: Regla graduada y/o mareógrafo.

CALADO, h: ecosonda.

POSICION (x,y) : Sistemas de GPS.

RESULTADOS: programas de generación de curvas (Civil 3D)

Traectoria para recolección de puntos

Si la medición se realiza en una laguna o embalse la trayectoria de medición será en líneas paralelas cada 5 y/o 10 metros entre líneas.

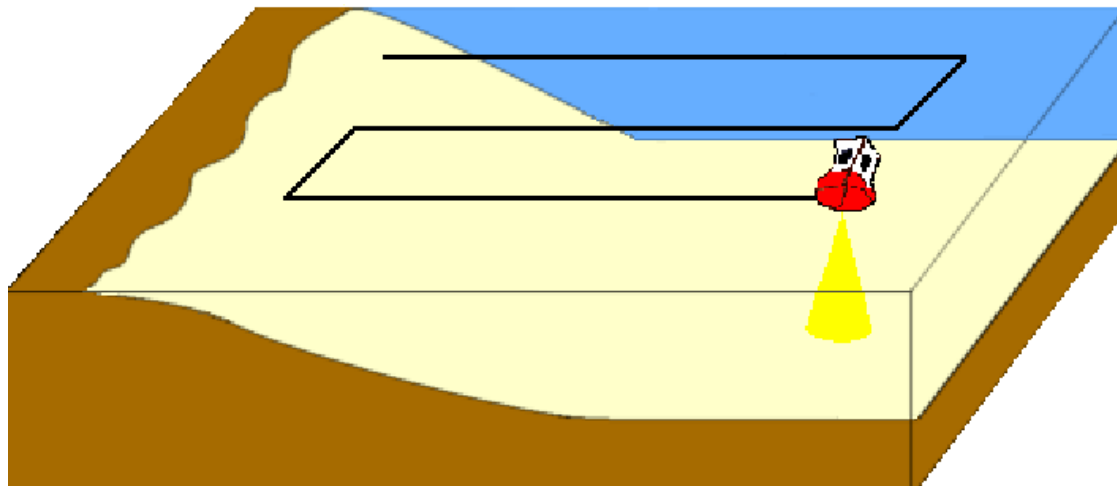


Figura N°40. Vista de la Trayectoria seguida por una embarcación para realizar trabajos de medición de profundidad.

Fuente: https://www.researchgate.net/figure/259592011_fig6_Fig-8-Trayectoria-para-una-batimetria-con-ecosonda-monohaz

Resultado final

El resultado del proceso es un mapa denominado batimétrico del fondo marino fondo del terreno bajo agua.

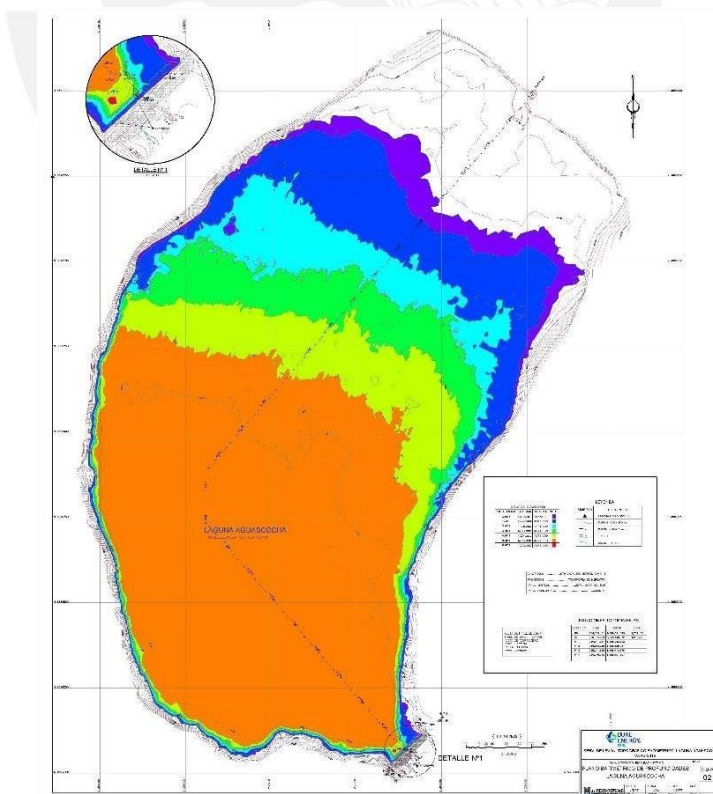


Figura N°41. Vista de Resultado del Trabajo Batimetrico

Fuente: www.hydroeval.com

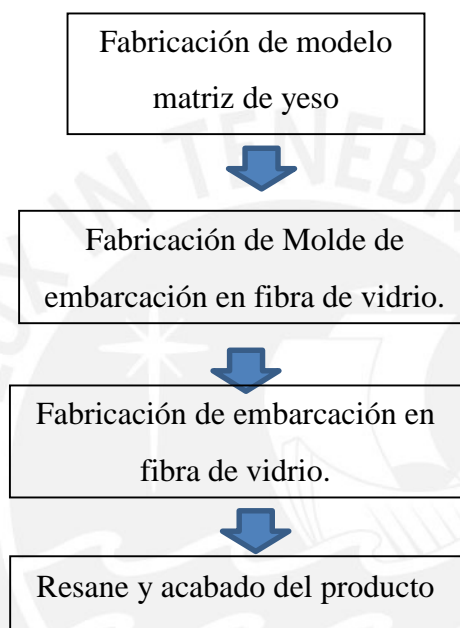
8 IMPLEMENTACION Y PRUEBAS

El presente capítulo se presentara el desarrollo de la fabricación e implementación con sus pruebas y ajuste correspondiente

8.1 EMBARCACION- FABRICACION

La embarcación casco principal será construido de base de plástico reforzado en fibra de vidrio comúnmente llamado fibra de vidrio.

El proceso se puede resumir en el siguiente grafico de proceso productivo



Fabricación de modelo

Para la Fabricación de la embarcación es necesario la fabricación de un modelo de yeso en la que se pueda replicar la forma diseñada.

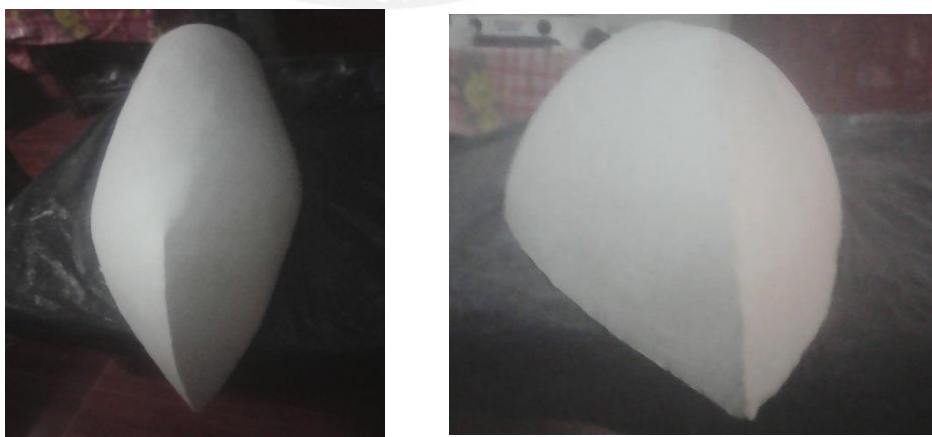


Figura N°42. Vista de Molde fabricado en yeso

Fabricación de Molde de Fibra de Vidrio.

En base Al modelo de yeso se fabrica un Molde de Fibra de Vidrio (Constituido Gelcoat que es una mezcla de resinas y fibras de refuerzo).

Gelcoat: Básicamente es una resina Poliéster diseñada para tener alta resistencia a la Abrasión.



Figura N°43. Fabricación de Modelo molde de resina de Poliéster.

Fabricación de embarcación en fibra de vidrio

En base al Molde de Fibra de vidrio se fabrica la pequeña embarcación usando una Fibra de vidrio que consiste en una combinación de resina Poliéster, telas de fibra de vidrio, Acelerador (Octato de Cobalto: Sirve para modificar la velocidad de reacción de las resinas,) y catalizador (Peróxido de Metil Etil Cetona Encargado de la polimerización).

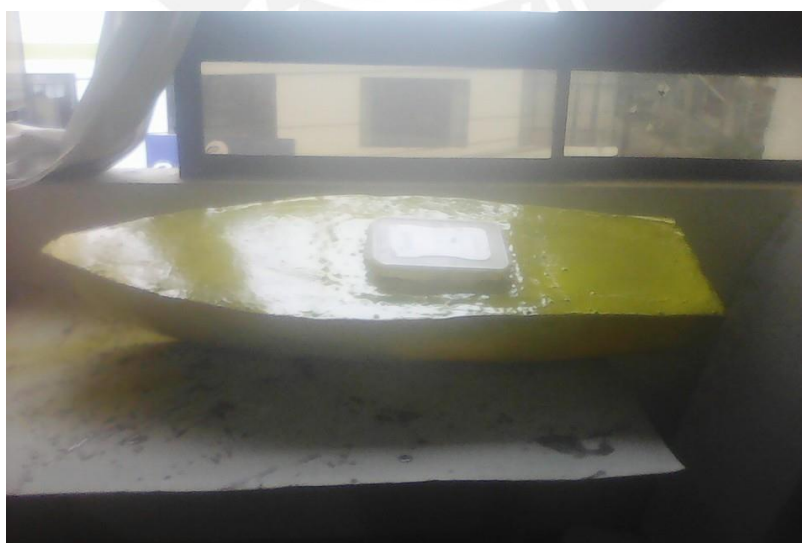


Figura N°44. Producto final – embarcación de resina de Poliéster.

8.2 DESARROLLO DEL SOFTWARE

8.2.1 SOFTWARE ARDUINO

El Arduino IDE es el software a utilizar para controlar el hardware y los sensores instalados en la embarcación.

8.2.2 OBJETIVO DE SOFTWARE

Los principales objetivos que persigue el software instalado en el hardware de la embarcación son los siguientes.

- a) Procesar la ruta de navegación ingresada.
- b) Obtener los datos de posición del GPS.
- c) Calcular el rumbo del bote.
- d) controlar el motor de impulsión y el servomotor de dirección.

8.2.3 OBJETIVO ESPECÍFICO DE SOFTWARE

El software en el microprocesador ArduPilot está diseñado para realizar las siguientes tareas:

1. Inicializar el hardware; espere hasta que el módulo GPS tenga un punto válido, y luego encender el motor.
2. Obtener la posición real y la información de rumbo desde el GPS.
3. Calcular la distancia y el rumbo al siguiente waypoint.
4. Comparar el rumbo real con la dirección y alimentar a la diferencia al servomotor del timón (PID).
5. Cuando la posición real está cerca dentro del rango del primer waypoint, interrumpir y seguir al siguiente waypoint.
6. Si cuando se alcanza el último punto de control, parar el motor; en caso contrario, proceder a Paso 2.

8.2.4 ESTRUCTURA DEL PROGRAMA

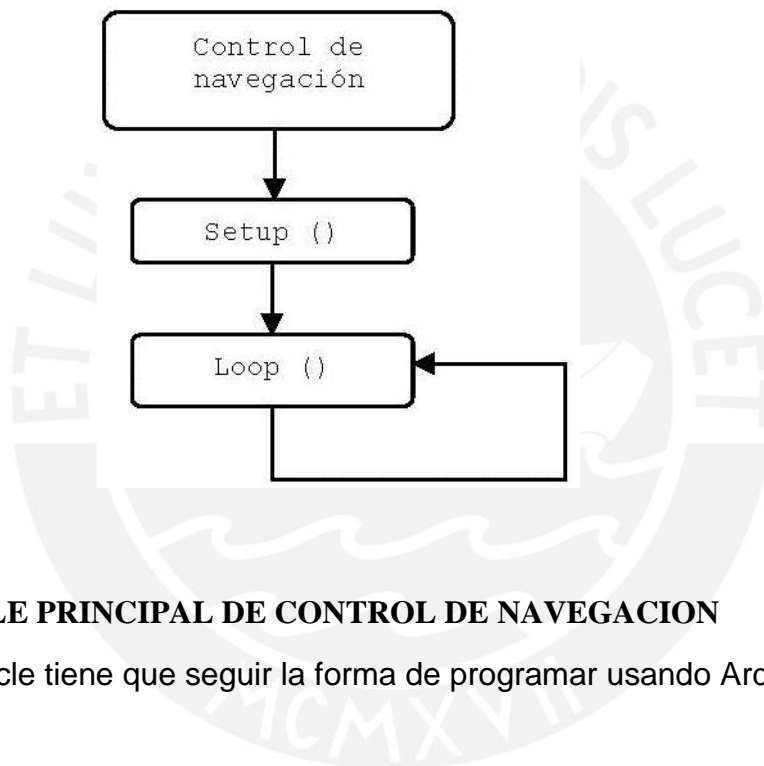
La estructura básica del lenguaje de programación de Arduino se compone de dos partes. Estas dos partes encierran bloques que contienen declaraciones, estamentos o instrucciones.

```

void setup()
    {
    Estamentos;
    }
void loop()
    {
    Estamentos;
    }

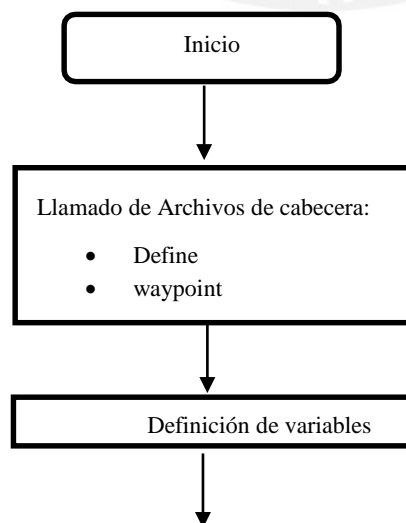
```

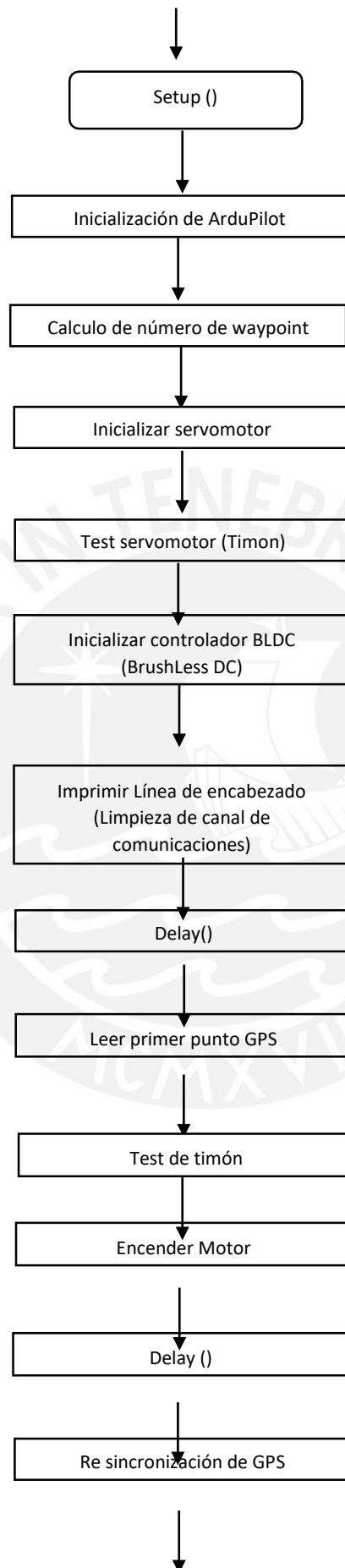
En donde *Setup* () es la parte encargada de recoger la configuración y *loop* () es la que contienen el programa que se ejecutará cíclicamente.

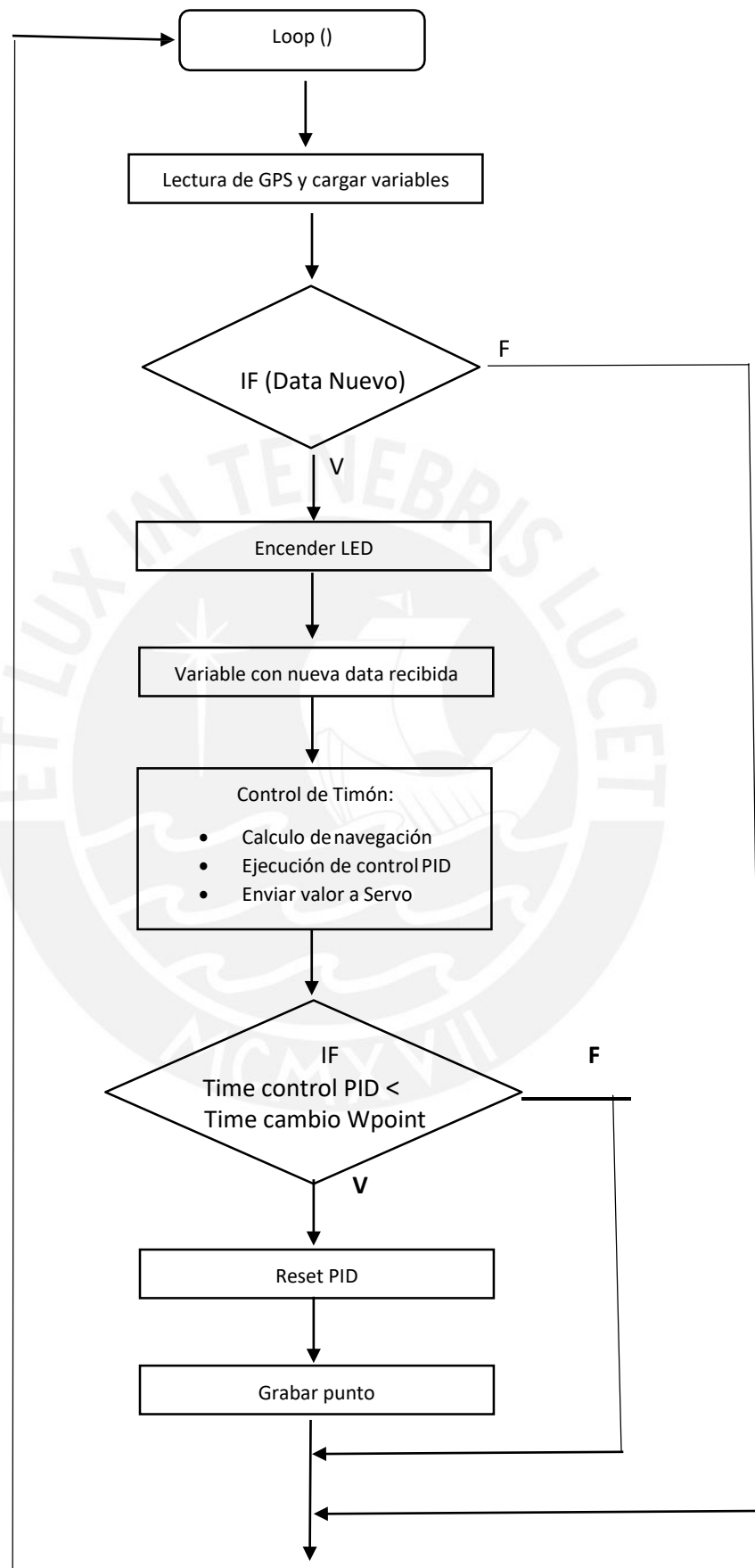


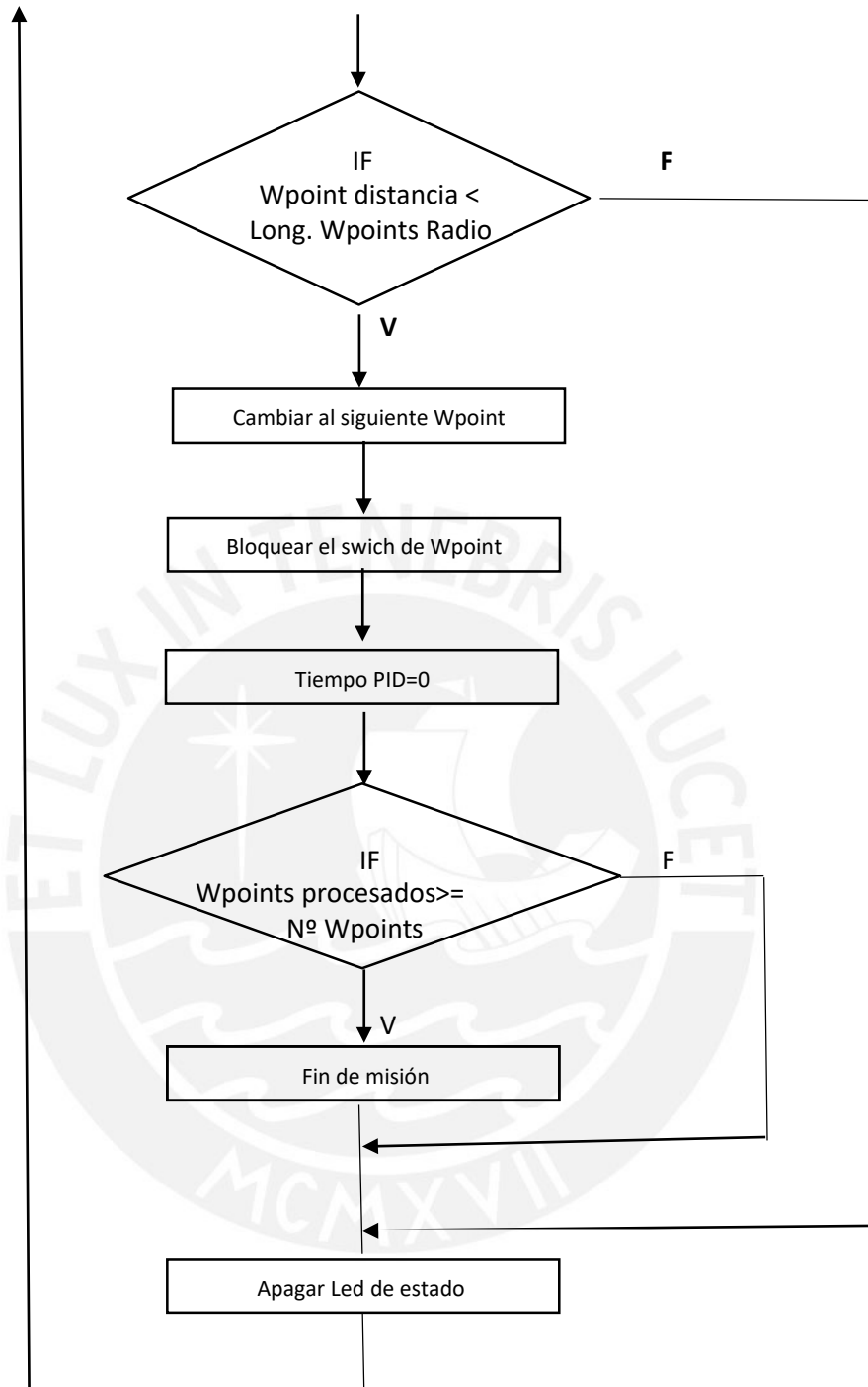
8.2.5 BUCLE PRINCIPAL DE CONTROL DE NAVEGACION

El bucle tiene que seguir la forma de programar usando Arduino IDE.





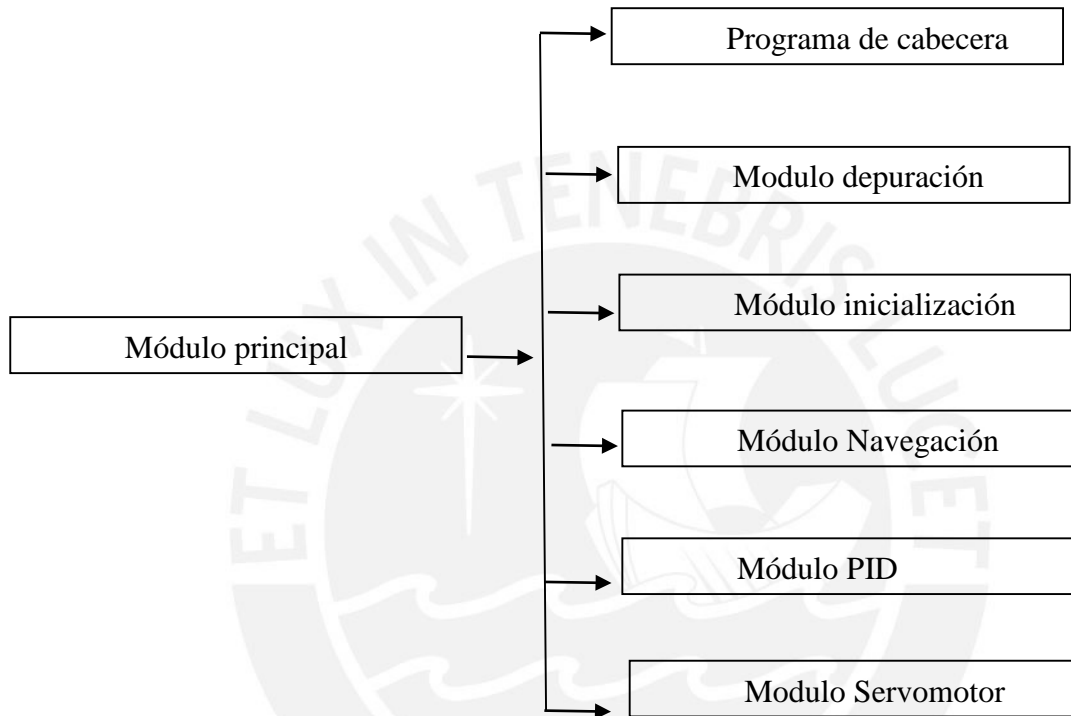




8.2.6 ESTRUCTURA DE BLOQUES DEL PROGRAMA

Para facilidad de manejo y depuración el programa se divide en módulos y consta de un módulo principal y módulos secundarios.

Los módulos secundarios son llamados por el modulo principal según necesidad.



8.3 PRUEBAS

Estas pruebas nos permitirán identificar las dificultades técnicas y proponer las mejoras en el modelo.

8.3.1 INSTALACIÓN DE SOFTWARE

Para cargar el software en el tablero ArduPilot, necesitará lo siguiente:

- Un Computadora para ejecutar el Arduino IDE.
- Un Breakout Board FTDI con el cable USB
- Código de embarcación autónoma.

Una vez que se tenga todos los requisitos anteriores listos se compilara y cargara el código en la tarjeta de desarrollo.

8.3.2 INSTALACIÓN DEL ARDUINO IDE

Antes de iniciar el Arduino IDE, se tiene que instalar los driver del convertidor de USB a serie y se deberá indicar al Arduino qué puerto serie se utilizará para comunicarse con el board.

8.3.3 8.3 CONFIGURACIÓN DEL ARDUINO IDE

Después de haber instalado correctamente el Arduino IDE, se tiene que indicarle al Arduino que tipo de tarjeta que está utilizando.

El ArduPilot utiliza un microprocesador ATmega328, y el board que es similar es el "Arduino Duemillanove Board. Por lo tanto se seleccionara el Arduino Duemilanove w/ATmega328 en el menú Herramientas Arduino IDE, como se muestra en la Figura inferior.

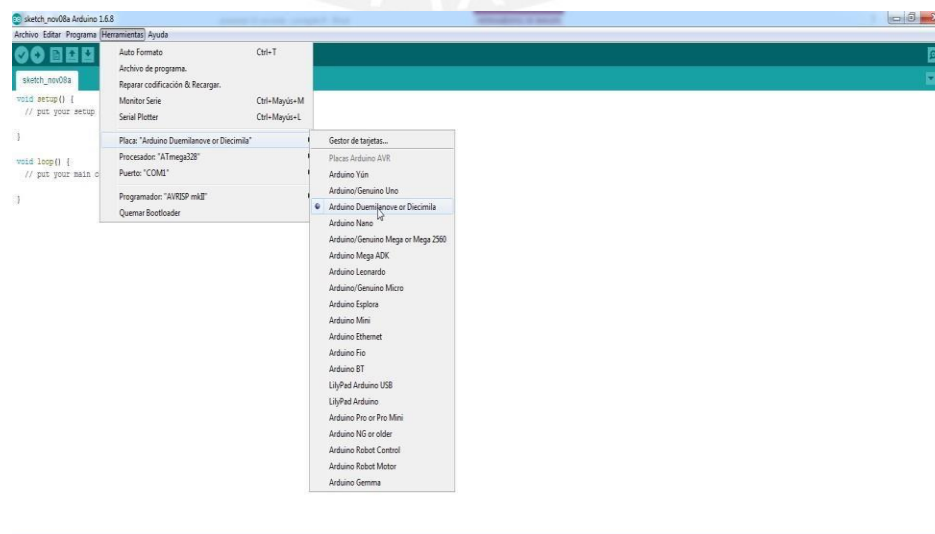


Figura N°45. Selección del hardware

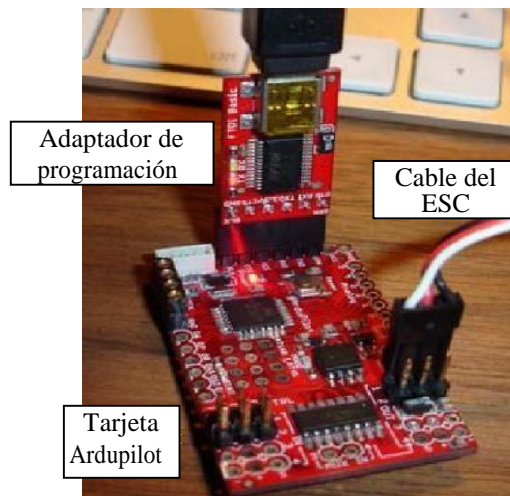


Figura N°46. Conexión del ArduPilot con el adaptador de programación.

Una vez cargado está listo para hacer las primeras pruebas con el bote.

8.3.4 PRUEBAS DE CAMPO

Condiciones para pruebas de campo:

- Las aguas tienen que ser calmas.
- Los vientos deben ser de baja intensidad.
- Temperatura de ambiente de 10 a 30 Grados centígrados.
- Las condiciones anteriormente descritas lo cumplen lagunas, presas, playas y piscinas de grandes dimensiones.

Dado al viaje permanente del tesista por razones laborales la prueba se realizó en una Laguna de la Ciudad de Huaraz.

Prueba en la Laguna Rajucolta

Para realizar las pruebas del Modelo experimental fue necesario ir a una laguna en este caso por razones laborales recurrí a la Laguna Rajucolta ubicado a 30 km de la ciudad de Huaraz en Ancash.



Figura N°47. Vista obtenida de Google Earth de la Laguna Rajulcota en Huaraz.



Figura N°48. Vista de area presa de la Laguna Rajulcota en Huaraz.

8.3.5 PRUEBAS DE MODELO EXPERIMENTAL

a) prueba de flotabilidad y la comportamiento hidrodinámico del Modelo experimental.

Se introdujo el Modelo experimental en el agua y se observó lo siguiente:

La flotabilidad del Modelo experimental es normal según lo esperado es decir floto correctamente.



Figura N°49. Vista de prueba de flotabilidad del Modelo experimental N°1.

Prueba hidrodinámica de la embarcación, consistente en mover el Modelo experimental de un punto a otro y ver el grado resistencia que ofrece el agua a la navegación.

En este punto los resultados fueron no los esperados pero lo suficiente para seguir con las pruebas.



Figura N°50. Vista de prueba hidrodinamica del Modelo experimental N°1.

b) Trayectoria de pruebas del Modelo experimental.

Se programó tres tipos de pruebas.

b.1 prueba tipo C, que consiste en soltar el bote y esperar que recorra la ruta tipo C.

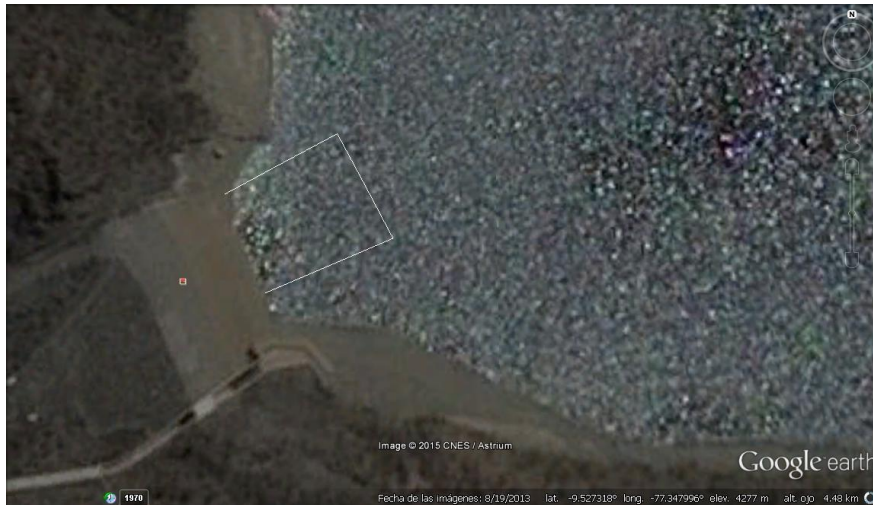


Figura N°51. Vista obtenida de Google Earth junto con la trayectoria de la prueba tipo C del Modelo experimental.

Coordenadas de prueba tipo C:

-77.34844003755818,-9.527409863307339,
 -77.34812674844577,-9.52727632805238,
 -77.34826775805662,-9.527017063778594,
 -77.34855262442439,-9.527165997081973,

Vista Fotográfica de la Prueba.



Figura N°52. Vista del Modelo experimental en trayectoria de la prueba tipo C del Modelo experimental.

Resultado 1: en esta prueba el Modelo experimental no recorrió la trayectoria en forma óptima, uno de los motivos el viento que era fuerte y sacaba de trayectoria el Modelo experimental.

b.2 prueba tipo Zic Zac, consiste en hacer una ruta en Zic Zac como se ve en la imagen.

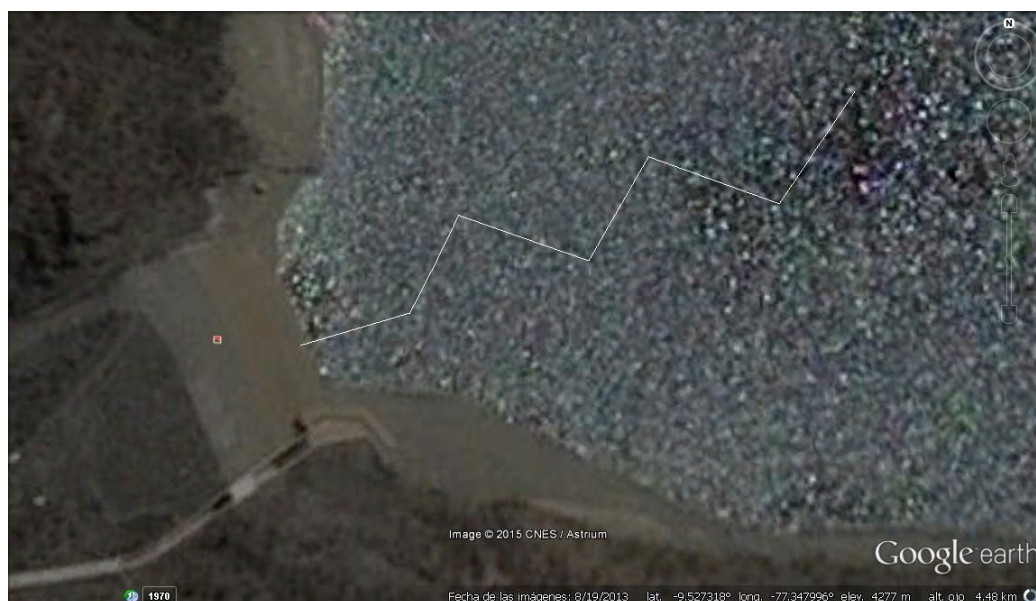


Figura N°53. Vista obtenida de Google Earth junto con la trayectoria de la prueba tipo Zic Zac del Modelo experimental.

Coordenadas de prueba tipo C:

-77.34847116680911,-9.527393413214393,
 -77.34824318939073,-9.527326766526381,
 -77.34814359107052,-9.52712267782201,
 -77.34786860122648,-9.527218103091848,
 -77.34774127789738,-9.527000190135601,
 -77.34747074654069,-9.527101331167215,
 -77.34730690209047,-9.526865046358914,

Vista Fotográfica de la Prueba.



Figura N°54. Vista del Modelo experimental en trayectoria Zic Zac del Modelo experimental.

Resultado 2: en la presenta prueba el Modelo experimental no recorrió la trayectoria en forma óptima, uno de los motivos es que la trayectoria del Modelo experimental coincidía la dirección del viento.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Después de desarrollar el presente informe podemos concluir lo siguiente:

- La tesis se basa en una embarcación que se fibra de vidrio y son de las siguientes medidas 1.0m x 0.40. ya que las embarcaciones de radio control que ofertan en el mercado local son pequeñas y no son adecuadas
- El controlador principal se está usando el controlador Ardupilot que es un controlador especializado para vehículos aéreos que tiene conexión para periféricos de GPS, servomotores y motores BrushLess.
- El protipo está diseñado para seguir una ruta previamente trazada y grabada en la memoria del controlador.
- El Gps utilizado es del tipo Navegador es decir presiones +-6 metros, por lo cual la distancia mínimas de los Waypoint debe ser al menos de 20 metros.
- Se recomienda utilizar el protipo en zonas abiertas y no en quebradas por que la precisión del GPS disminuye hasta +-15

Recomendaciones

- Mejorar la hidrodinámica de la embarcación por que opone mucha resistencia a la navegación del Modelo experimental en el agua.
- Empleo de un Gps geodésico de doble frecuencia que tiene precisión +-1 cm. Que consiste en usar 02 equipos GPS. Uno se utilizara como base y otro con GPS Rover este último iría en la embarcación. Aunque esto último escapa del presupuesto del tesista.
- Adicionando un Ecosonda este Modelo experimental se puede emplear para sondear y/o medir profundidades y confeccionar mapas.
- Se puede complementar el Modelo experimental adicionándole varios sensores según al objetivo que deseamos darle.

ANEXOS

ANEXO 1: ANALISIS HIDROSTATICO DE EMBARCACION

Informe hidrostático del diseño



Informe hidrostático del diseño.

Diseñador

Creado por

Comentario

Nombre de archivo

Modelo 0.80x0.fbm

Eslora de diseño	0.800 (m)	Posición de la de sección media	0.400 (m)
Eslora total	0.800 (m)	Relative water density	1.025
Manga de diseño	0.280 (m)	Grosor medio del forro	0.0000 (m)
Manga total	0.277 (m)	Coefficiente de apéndice	1.0000
Calado de diseño	0.050 (m)		

Propiedades del volumen		Propiedades del plano de flotación	
Moulded volume	0.003 (m ³)	Eslora en línea de agua	0.641 (m)
Volumen desplazado	0.003 (m ³)	Manga en la línea de agua	0.224 (m)
Desplazamiento	0.003 (tonnes)	Ángulo de entrada	11.213 (Degr.)
Coefficiente de bloque	0.2448	Área del plano de flotación	0.099 (m ²)
Coefficiente prismático	0.4248	Coefficiente del plano de flotación	0.4413
Coefficiente prismático vertical	0.5547	Centro del plano de flotación	0.371 (m)
Superficie del área mojada	0.114 (m ²)	Momento de inercia transversal	0.000 (m ⁴)
Centro de flotación longitudinal	0.387 (m)	Momento de inercia longitudinal	0.002 (m ⁴)
Centro de flotación longitudinal	-2.095 %		
Centro de flotación vertical	0.033 (m)		

Propiedades de la de sección media		Estabilidad inicial	
Área de la de sección media	0.008 (m ²)	Altura metacentro transversal	0.140 (m)
Coefficiente de la sección media	0.5762	Altura del metacentro longitudinal	0.801 (m)

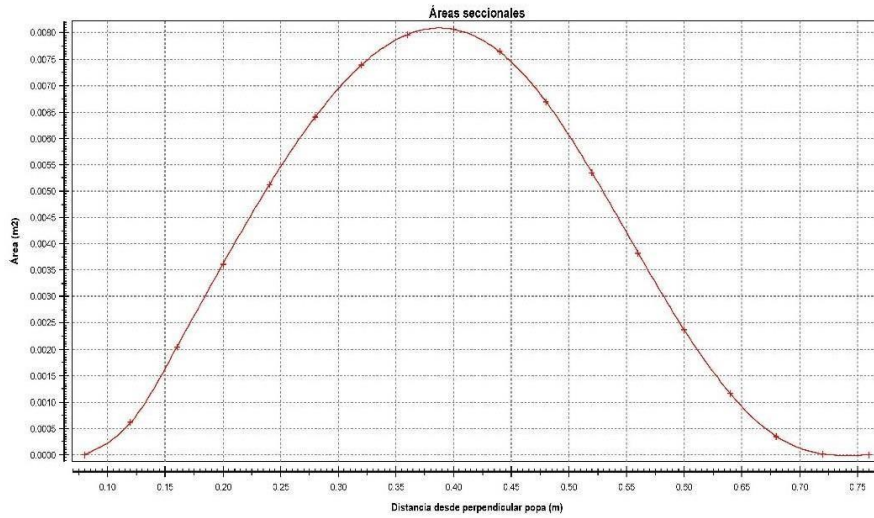
Plano lateral	
Área lateral	0.021 (m ²)
Centro de esfuerzo longitudinal	0.414 (m)
Centro de esfuerzo vertical	0.030 (m)

Las siguientes propiedades de capa fueron calculadas para ambos lados del buque



Posición	Área (m ²)	Grosor	Peso (tonnes)	LCG (m)	TCG (m)	VCG (m)
Capa 0	0.339	0.000	0.000	0.382	0.000 (CL)	0.082

Áreas seccionales									
Posición (m)	Área (m ²)	Posición (m)	Área (m ²)	Posición (m)	Área (m ²)	Posición (m)	Área (m ²)	Posición (m)	Área (m ²)
0.080	0.000	0.240	0.005	0.400	0.008	0.560	0.004	0.720	0.000
0.120	0.001	0.280	0.006	0.440	0.008	0.600	0.002	0.760	0.000
0.160	0.002	0.320	0.007	0.480	0.007	0.640	0.001		
0.200	0.004	0.360	0.008	0.520	0.005	0.680	0.000		



NOTA 1: ¡El calado (y todas las otras alturas) se han medido sobre la base Z=0.00!
 NOTA 2: Todos los coeficientes calculados se han basado en la eslora, calado y manga del proyecto.



Hydrostatics

Asiento: 0.000 (m)

Calado (m)	Volumen (m ³)	Despl. (tonnes)	Despl. (tonnes)	LCB (m)
0.000	0.000	0.000	0.000	0.000
0.010	0.000	0.000	0.000	0.405
0.020	0.000	0.000	0.000	0.401
0.030	0.001	0.001	0.001	0.397
0.040	0.002	0.002	0.002	0.392
0.050	0.003	0.003	0.003	0.387

NOTA 1: ¡El calado (y todas las otras alturas) se han medido sobre la base Z=0.00!0.000

NOTA 2: Todos los coeficientes calculados se han basado en la eslora, calado y manga del proyecto.

Nomenclature

Calado	<i>Moulded draught, measured from baseline</i>
Volumen	<i>Volumen desplazado</i>
Despl.	<i>Desplazamiento</i>
Despl.	<i>Desplazamiento</i>
LCB	<i>Centro de flotación longitudinal, medido desde la popa perpendicular a X=0.0</i>



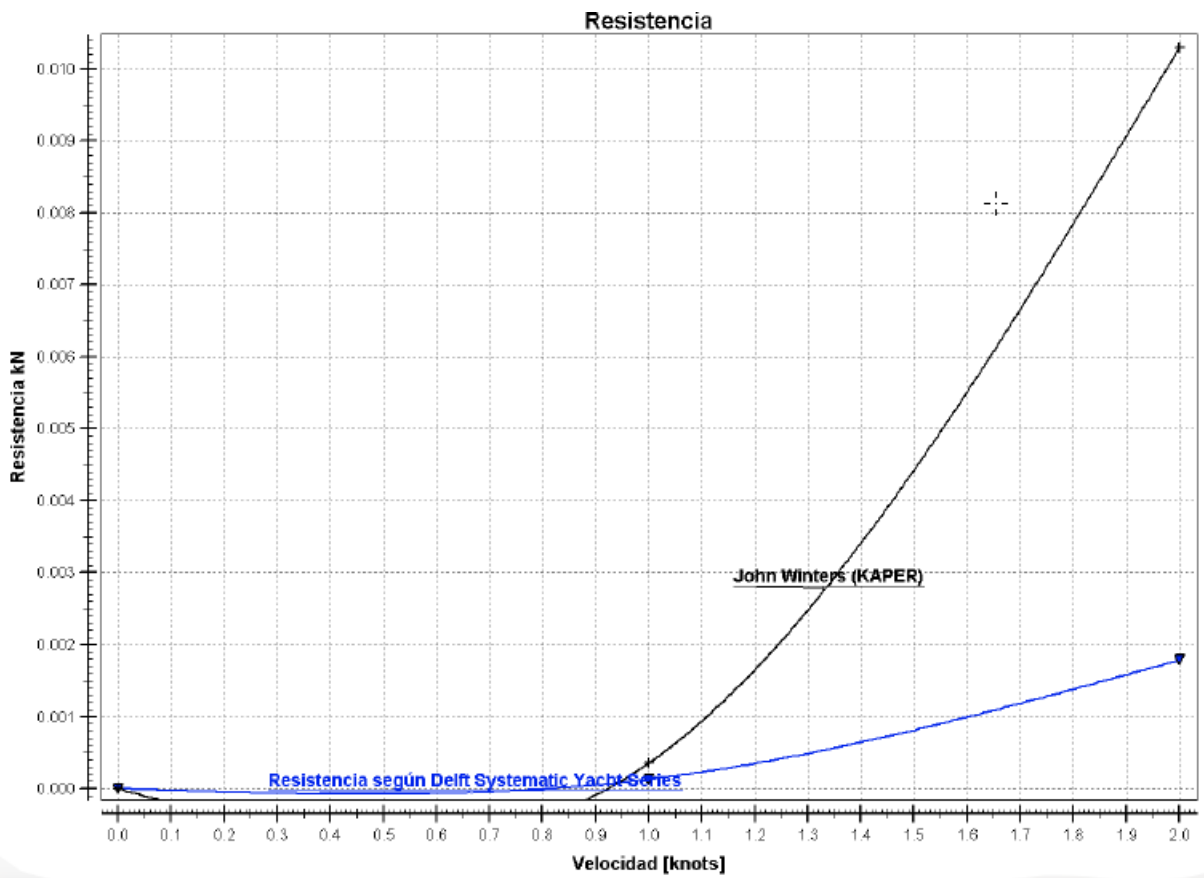
Resistance calculations.

Resistencia según Delft Systematic Yacht Series.

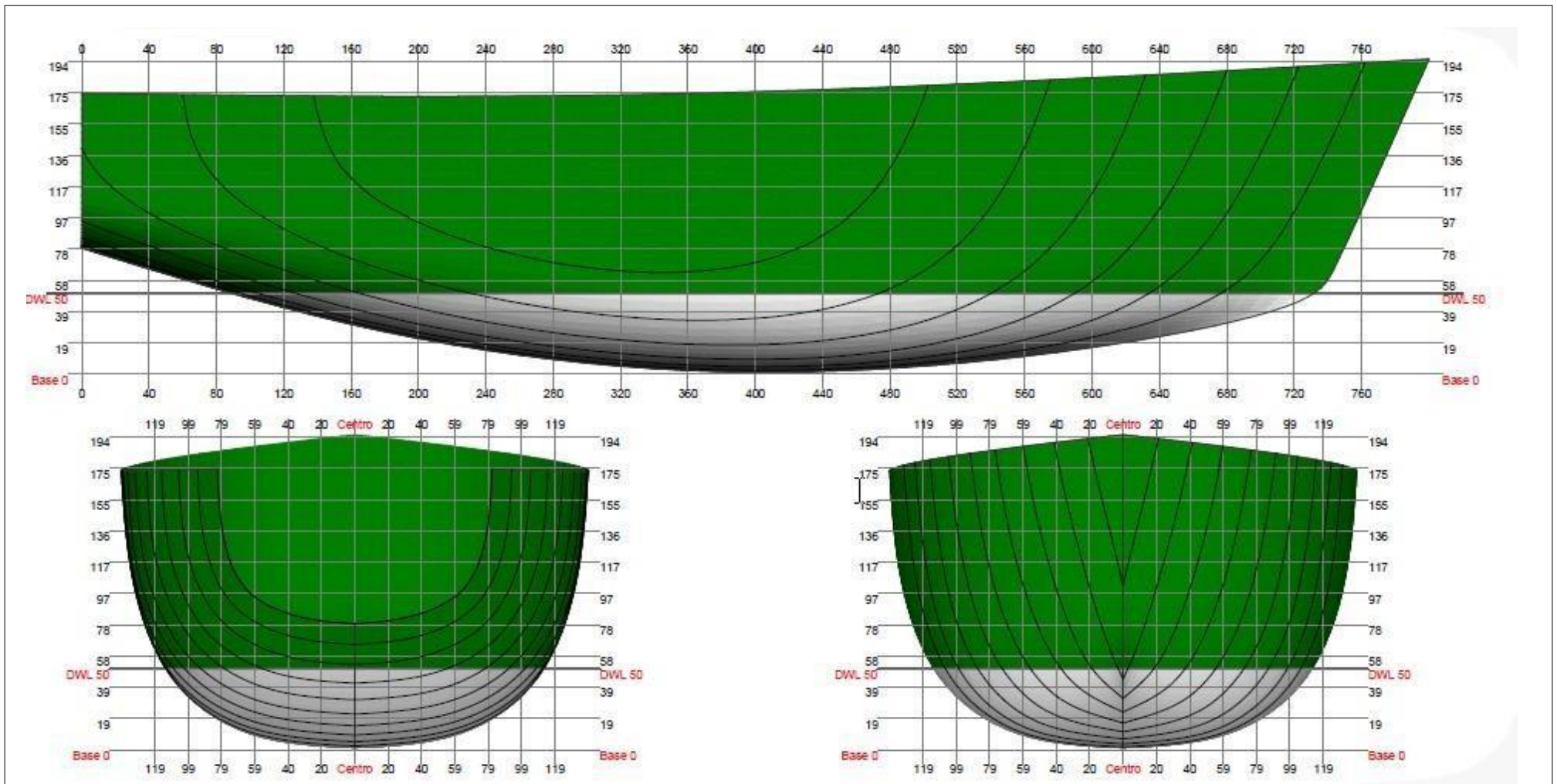
Resistance according to Resistencia según Delft Systematic Yacht Series									
V	v	Fn	Rf	Rr	Rtotal	Pe			
0.00	0.000	0.000	0.0000	0.0000	0.0000	0.0000			
1.00	0.514	0.245	0.0001	0.0000	0.0001	0.0000			
2.00	1.029	0.490	0.0003	0.0015	0.0018	0.0018			

John Winters (KAPER).

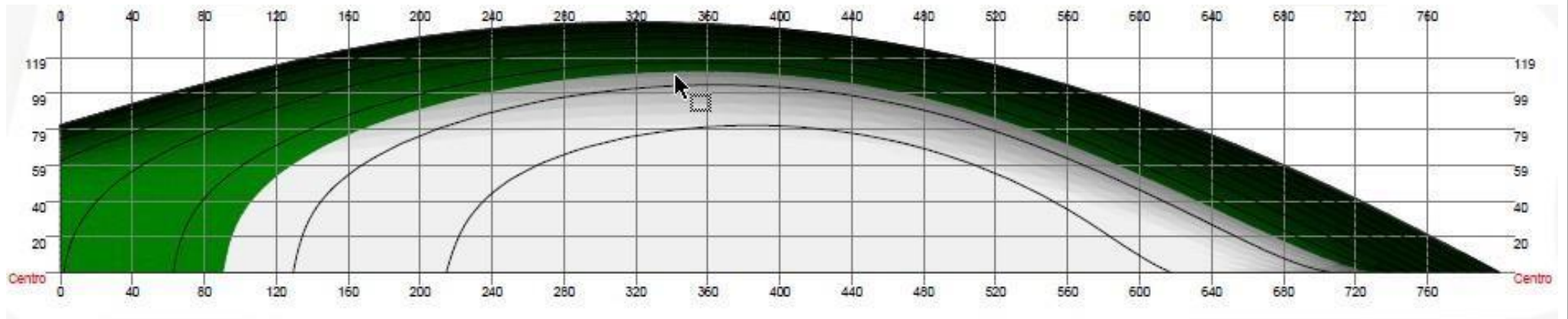
Resistance according to John Winters (KAPER)								
V	v	Fn	Rf	Rr	Rtotal			
kn.	m/s		[kN]	[kN]	[kN]			
0.00	0.000	0.000	0.000	0.0000	0.0000			
1.00	0.514	0.205	0.0001	0.0001	0.0003			
2.00	1.029	0.410	0.0003	0.0100	0.0103			



RESULTADOS GRAFICOS



Vista de la forma longitudinal, posterior y delantera de los resultados obtenidos de la modelación hidrostática para obtener la forma de embarcación óptima con menos pérdidas por fricción



Vista de la forma longitudinal inferior de los resultados obtenidos de la modelación hidrostática para obtener la forma de embarcación optima con menos pérdidas por fricción

ANEXO 2: DESARROLLO DEL PROGRAMA

El Ardupilot viene con programas propios de libre disposición para control de helicópteros y aviones pequeños, en nuestro caso adaptaremos estos programas básicos para usarlos en nuestra embarcación para una navegación autónoma usando solo el GPS para la ubicación de la embarcación, Motor Brushless para control de velocidad y servomotor para control de dirección.

a) **Programa principal:**

Está constituido por un programa que controlara el software de la embarcación y hará un llamado a las demás funciones según necesidad.

El programa principal contiene la declaración de las variables globales y la configuración de las funciones principales de Arduino *Setup ()* y *loop ()*.

```
principal.pde
#include "defines.h"
#include "waypoints.h"
//Definición de variables globales
int waypoints; //Contador de waypoint
unsigned int integrator_reset_counter = 0; //Contador variable (en segundos) para el Integrador tras un switch del
    waypoint.
byte current_wp = 0; // Esta variable almacena el waypoint real que estamos tratando de alcanzar
int wp_bearing = 0; // El rumbo para el waypoint actual (en grados)
unsigned int wp_distance = 0; //El distanciamiento para el waypoint actual (en metros)

//Variables de Información obtenida del GPS
float lat = 0; //Latitud actual
float lon = 0; //Longitud actual
unsigned long time; // Tiempo actual de la hora universal
float ground_speed = 0; //Velocidad
int course = 0; // Rumbo
int alt = 0; //Altitud

//Variables del indicador
byte jumplock_wp = 0; // Cuando se cambia waypoints este bloqueo permitirá sólo una transición
byte gps_new_data_flag = 0; //Un indicador simple para estar enterado cuando tenemos nuevos datos de
    GPS.

//Variable setpoint de timón, mantiene el valor calculado para el servo de timón
int rudder_setpoint = 0;
```



```

byte fix_position = 0; // El indicador variable para posición válida de GPS
//Inicio de Arduino, punto de entrada después de la puesta en marcha
void setup()
{
  init_ardupilot(); // Inicializar los accesorios periféricos del hardware (llamado al módulo de inicialización)
  waypoints = sizeof(wps) / sizeof(LONLAT); // cálculo de los numero de waypoints
  Init_servo(); //Inicializarlo servos, llamado al Módulo de Servomotor
  test_rudder(); //Simple movimiento del servo para ver si está activo
  bldc_arm_throttle(); //Inicialice el controlador BLDC (Brushless DC)
  print_header(); //Imprimir la línea del encabezado para limpiar el canal de comunicaciones
  delay(500); //Esperar que el UART Tx Buffer sea correctamente purgado
  init_startup_parameters(); // Esperar el primer Punto GPS
  test_rudder(); //Mover rudder-servo para ver el tiempo cuando el bote está cerca
  bldc_start_throttle(); //Empezar a mover el motor
  delay(5000); //Pasar los primeros 5seg. sin GPS para que la dirección se establezca
  init_startup_parameters(); //Re-sincronizar el GPS
}
// Lazo del programa principal
// Lazo Arduino principal

void loop()
{
  gps_parse_nmea(); //Análisis y lectura de datos del módulo GPS y almacenar en las variables globales
  if((gps_new_data_flag & 0x01) == 0x01) //Checking nuevo GPS "GPRMC" indicador de datos de posición
  {
    digitalWrite(YELLOW_LED, HIGH); //Pulsar LED amarillo para indicar frase GPS recibida
    gps_new_data_flag &= (~0x01); //Limpieza del indicador de new data...
    rudder_control(); //Función de control para dirigir el curso al siguiente waypoint
    if (integrator_reset_counter++ < WP_TIMEOUT) //Control I y D del cero para WP_TIMEOUT segundos después
      de cada cambio de waypoint
    reset_PIDs();
    send_to_ground(); /* Valores impresión en datalogger, sólo para la depuración */
  }
  // Terminar si gps_new_data...

// El Ardupilot saltará SOLO UN waypoint
  if((wp_distance < WP_RADIUS) && (jumplock_wp == 0x00)) //Verificando la distancia waypoint es menor que
    WP_RADIUS m, y comprobar si el bloqueo está abierto
  {
    current_wp++; //Cambie el waypoint
    jumplock_wp = 0x01; //Bloqueo del conmutador waypoint
    integrator_reset_counter = 0;
    if(current_wp >= waypoints) //Compruebe si hemos pasado todos los puntos de interés, si es así detener motor
    finish_mission();
  }
}

```

```
// Termine si wp_distance...

digitalWrite(YELLOW_LED,LOW); // Al apagar el estado de LED
}
//fin de loop ()
```

b) Programas de cabecera.

Está constituido programa por 02 archivos de programa que serán llamados por el programa principal, estas contienen datos de ruta, y los parámetros PID para el control de dirección.

b.1) waypoint.h

En este archivo se indicara los puntos (waypoint) o ruta que seguirá la embarcación.

```
// Definición de tipo de datos de waypoint

typedef struct { float lon;
                float lat;
                }LONLAT;

// A continuación se definen son los puntos de referencia codificadas para las ruta
#if 0
LONLAT wps[] =
{
-77.17706435232431,-11.77381093739521,
-77.17731021263153,-11.77359595297081,
-77.17706594285977,-11.77330522015979,
-77.17681972134602,-11.77353722956144,
};
#endif
```

b.2) define.h

En este archivo se indicara los parámetros PID, del motor, servomotor y definición de pinout.

```
#define REVERSE_RUDDER 1 // Normal = 0 y reversa = 1
#define MIDDLE_RUDDER 90 // Posición central del servo del timón en grados, ajuste para las correcciones de
ajuste si es necesario

// Conjunto de parámetros para la embarcación

#define KP_HEADING 2.0 // Parte proporcional de control PID
#define KI_HEADING 0.07 // Integrador parte de control PID
#define KD_HEADING 0.00001 // Derivador parte del control PID (no se utiliza)
#define MOTOR_SPEED 80 // Alrededor 5A
```

```
#define WP_TIMEOUT 15 // Valor del contador de tiempo de espera en segundos Waypoint
#define WP_RADIUS 15 // Radio de waypoint-hit en m
```

```
// PID max y min
```

```
#define HEADING_MAX 60 // Posición max del Servo en grados
```

```
#define HEADING_MIN -60 // Posición min del Servo en grados
```

```
#define INTEGRATOR_LIMIT 240 // acerca de + - 15 grados
```

```
// Definición de rangos de los servos (y ESC), Los valores son en segundos
```

```
#define MAX16_THROTTLE 2000 // ESC max posicion
```

```
#define MIN16_THROTTLE 1000 //ESC min posicion
```

```
#define MAX16_RUDDER 2000 // Posición máxima de (timón) Servo
```

```
#define MIN16_RUDDER 1000 // Posición mínima de (timón) Servo
```

```
// Definición de los pines I/O
```

```
#define MUX_PIN 4
```

```
#define SERVO1_IN_PIN 2 // Servomecanismo del Regulador del Receptor de RC introducido
```

```
#define SERVO2_IN_PIN 3 // Servomecanismo del Timón del Receptor de RC introducido
```

```
#define BLUE_LED 12 // Indicador de posición GPS LED
```

```
#define YELLOW_LED 13 // LED de estado
```

```
#define THROTTLE 9 // Acelerador
```

```
#define RUDDER 10 // Timón
```

```
#define MOTOR_OFF 5 // Ajuste para ESC, que con seguridad apaga el motor
```

c) Módulo de depuración y pruebas

Este archivo contiene funciones de depuración y pruebas. Además contiene programación para grabar los datos del bucle de control PID que se emite cada segundo.

```
// Variables de ajuste de PID
```

```
float pid_p;
```

```
float pid_i;
```

```
float pid_d;
```

```
float pid_dt;
```

```
int dbg_pid_error;
```

```
// Depuración de salida, envía el valor de las variables internas para el registrador de datos cada segundo
```

```
// Los valores del punto flotante se multiplican y se convierten a números enteros para obtenerla mediante la función
Serial.print
```

```
void send_to_ground(void)
```

```
{
```

```

Serial.print(course);
Serial.print("\t");

Serial.print((int)wp_bearing);
Serial.print("\t");

Serial.print(dbg_pid_error);
Serial.print("\t");

Serial.print(wp_distance);
Serial.print("\t");

Serial.print(time);
Serial.print("\t");

Serial.print((int)rudder_setpoint);
Serial.print("\t");

Serial.print((int)current_wp);
Serial.print("\t");

Serial.print((int)pid_p);
Serial.print("\t");

Serial.print((int)pid_i);
Serial.print("\t");

Serial.print((int)pid_d);
Serial.print("\t");

ground_speed *= 18.0; // Scale miles/h to km/h * 10
Serial.print((int)ground_speed);
Serial.print("\t");
Serial.print(alt);
Serial.println();
}

// Depuración de la salida, envía el valor de variables internas al datalogger una vez en el arranque
// Los valores del punto flotante se multiplican y se convierten a números enteros para conseguirlo a través de la
función de Serial.print

void print_header(void)

{
// Encabezado para constantes del sistema

```

```

Serial.println("KP_HEADING\t KI_HEADING\t KD_HEADING\t INTEGRATOR_MAX\t RAM");
delay(250);
Serial.print ((int)(KP_HEADING * 100));
Serial.print("\t");
Serial.print ((int)(KI_HEADING * 100));
Serial.print("\t");
Serial.print ((int)(KD_HEADING * 100));
Serial.print("\t");
Serial.print((int)(INTEGRATOR_LIMIT));
Serial.print("\t");
Serial.println( ram_info() );
delay(250);

```

// Cabecera para las variables de depuración

```

Serial.println("Act\t Setp\t err\t Dist\t Time\t Rudd\t WP\t pid_p\t pid_i\t pid_d\t speed\t alt");
delay (250);
}

```

// Función para calcular la cantidad restante de memoria RAM en Bytes

// Comprobacion, si ha cambiado la matriz Waypoint (ver el encabezado del Depuracion- output)

```

int ram_info()
{
uint8_t *heapptr;
uint8_t *stackptr;

stackptr = (uint8_t *)malloc(4); //Utilizar stackptr temporalmente
heapptr = stackptr; //Guardar el valor del puntero de almacenamiento
free(stackptr); // liberar la memoria otra vez (conjuntos stackptr a 0)
stackptr = (uint8_t *) (SP); // guardar el valor del puntero depila

return ((int) stackptr - (int) heapptr);
}

```

d) Módulo inicialización

Esta ficha contiene todas las funciones de inicialización para el hardware.

```

void init_ardupilot(void)
{
gps_init_baudrate();

Serial.begin(9600);

//Declaracion de pines
pinMode(5,INPUT);

```

```

// Pin de modo (no utilizado)

```

```

pinMode(11,OUTPUT); // Pin Simulador de salida (no se utiliza)
pinMode(MUX_PIN,OUTPUT); // Pin MUX, se aplica sólo a hardware modificado!
pinMode(BLUE_LED,OUTPUT); // bloquear pin LED de ArduPilot, que indican los datos GPS
pinMode(YELLOW_LED,OUTPUT); // LED de estado, parpadea al recibir datos del satélite válidos
pinMode(SERVO1_IN_PIN,INPUT); // Entrada de acelerador de RC Rx (solo para radio control RC)
pinMode(SERVO2_IN_PIN,INPUT); // Entrada del timón de RC Rx (solo para radio control de RC)

#ifdef RADIO_CONTROL
init_RC_control(); // Inicialice radiocontrol
#endif

switch_to_ardupilot(); // Control por defecto de servo por ArduPilot
}

void init_startup_parameters(void)
{
// Sí un do-while, comprueba una y otra vez hasta que tengamos una posición GPS válida y lat está diferentes de
// cero.
// Volver a verificar la Lat porque a veces falla y establecer el punto inicial lat como cero.
do
{
gps_parse_nmea(); // Lectura y análisis de los datos del GPS
}
while(((fix_position < 0x01) || (lat == 0)));

// otra verificación gps_new_data_flag=0;

do
{
gps_parse_nmea(); // Lectura y análisis de los datos del GPS
}
while((gps_new_data_flag&0x01 != 0x01) & (gps_new_data_flag&0x02 != 0x02));

rudder_control(); // calcular la distancia hasta el siguiente punto de referencia, de lo contrario, se iniciará a las
// waypoint 2.
}

```

e) Módulo Navegación.

Este módulo es uno de los más importantes del piloto automático contiene el llamado "NMEA - parser", que decodifica los datos desde el módulo GPS y lo almacena en las variables globales. El modulo también contiene las funciones que calculan el ángulo del azimut y la distancia hasta el siguiente punto de referencia.

// Las variables utilizadas por el Analizador

```

NMEA(NMEA Parser) char buffer[90]; // Búfer serial para coger los datos del GPS
/* GPS Punteros */
char *token;
char *search = ",";
char *brkb, *pEnd;

/*****
*Esta función analiza las cadenas NMEA...
*Bastante complejo, pero nunca falla y funciona bien con todos los módulos GPS y velocidades de transmisión.. :-
)
*Sólo cambiar el valor Serial.begin() en la primera pestaña para velocidades de transmisión más altas
*****/

void gps_parse_nmea(void)
{
const char head_rmc[]="GPRMC"; //Buscar encabezado GPS NMEA
const char head_gga[]="GPGGA"; // Buscar encabezado GPS NMEA

static byte unlock=1; // Indicador de evento
static byte checksum=0; //El Checksum generado
static byte checksum_received=0; //Checksum recibido
static byte counter=0; // Contador general

// Las variables temporales para algunas tareas, especialmente utilizados en la parte de análisis GPS

unsigned long temp=0;
unsigned long temp2=0;
unsigned long temp3=0;

while(Serial.available() > 0)
{
if(unlock==0)
{
buffer[0]=Serial.read();//coloca un byte en el búfer

if(buffer[0]=='$')// Verifique si es el preámbulo $
{
unlock=1;
}
}
/*****/ else
{
buffer[counter]=Serial.read();

```

```

if(buffer[counter]==0x0A)//Looks para \F
{
unlock=0;
if (strncmp (buffer, head_rmc, 5) == 0)    // $GPRMC análisis comienza aquí
{

/* Generación y análisis de checksum recibido, */

for(int x=0; x<100; x++)
{
if(buffer[x]=='*')
{

checksum_received=strtol(&buffer[x+1],NULL,16);// Analizar checksum ...
break;

else
{
checksum ^= buffer[x]; //XOR los datos recibidos...
}
}

if(checksum_received == checksum)// checksum Comprobación
{
/* De emergencia señalará los datos entre comas "", devuelve los datos en el orden recibido */
/* La orden GPRMC es: UTC, estado UTC, Lat, N / indicador S, Lon, E / W indicador, velocidad, rumbo, la fecha,
el modo, checksum */

token = strtok_r(buffer, search, &brkb); // Contiene la cabecera GPRMC, no se utiliza utiliza...

token = strtok_r(NULL, search, &brkb); // Hora UTC, no se utiliza
time = atol (token);
token = strtok_r(NULL, search, &brkb); // Datos UTC Válido? quizá no usado

// Longitud en grados, minutos decimales. (ej. 4750.1234 grados min. Dec.= 47.835390 grados decimales)
// Donde 47 son grados y 50 minutos y los 0.1234 los decimales de los minutos. decimales),
// Convertir en grados decimales, divida los minutos 60 (incluyendo
// Ejemplo: "50.1234 / 60 = 0,835390", a continuación, añadir los grados, ex: "47 + 0.835390 = 47.835390" grados
decimales
token = strtok_r(NULL, search, &brkb); // Contiene Latitud en grados decimales minutos...

// Serial.println(token);
// tomando sólo grados, minutos y sin decimales,
// strtol deja de analizar hasta alcanzar el punto decimal "." ejemplo resultado 4750, elimina 0.1234

```



```

temp = strtol (token, &pEnd, 10);

// Toma sólo los decimales de los minutos
// ejemplo resultado 1234.
temp2 = strtol (pEnd + 1, NULL, 10);

// unirse grados, minutos y los decimales de minutos, ahora sin el punto...
// Antes era 4750.1234, ahora el ejemplo de resultado es 47501234...

temp3 = (temp * 10000) + (temp2);

// Módulo para dejar sólo los minutos decimales, eliminando sólo los grados
// Antes era 47501234, el ejemplo resultado es 501.234.

temp3 = temp3 % 1000000;

// Dividir para obtener sólo los grados, antes era de 4750
// El resultado es 47 (4750/100=47)

temp /= 100;

// Juntar todo y convertir a variable flotante.....
// Primero convierto los minutos dec a grados dec almacenados en "TEMP3", ejemplo: 501234/600000 = 0,835390
// Los grados almacenados en "temp" y añadir el resultado de la 1ra etapa, el ejemplo 47 + 0.835390 = 47.835390
// El resultado se almacena en la variable "lat"...

lat=temp + ( (float)temp3 / 600000 );

token = strtok_r(NULL, search, &brkb); //lat, Norte o Sur?

// Si el Char es igual a S (sur), multiplicar el resultado por -1 ..
if(*token == 'S')
{
lat = lat * -1;
}

// Este mismo procedimiento en el uso lat, pero ahora para Lon ....

token = strtok_r(NULL, search, &brkb);

// Serial.println(token);

temp = strtol (token, &pEnd, 10);
temp2 = strtol (pEnd + 1, NULL, 10);
temp3 = (temp * 10000) + (temp2);
temp3 = temp3 % 1000000;

```

```

temp /= 100;
lon=temp + ((float)temp3 / 600000);

token = strtok_r(NULL, search, &brkb); //lon, Este o Oeste?
if(*token == 'W')
{
lon=lon * -1;
}
token = strtok_r(NULL, search, &brkb); //Velocidad de superficie?
ground_speed = atof(token);
token = strtok_r(NULL, search, &brkb); //Rumbo?
course= atoi(token);

gps_new_data_flag |= 0x01; // Actualización del indicador para señalarla llegada de los nuevos datos

jumplock_wp=0x00;// bloqueo waypoint de compensación..

}
checksum=0;
} // Fin del análisis GPRMC

if (strcmp (buffer,head_gga,5) == 0) // $GPGGA El analisis empieza aqui
{
/* Generación y análisis recibido checksum, */

for(int x=0; x<100; x++)
{
if(buffer[x] == '*')
{
checksum_received = strtol(&buffer[x+1], NULL, 16); //Analisis checksum...
break;
}
else
{
checksum ^= buffer[x]; //XOR de la Data recibida...
}
}

if(checksum_received == checksum)//Chequeo checksum
{
token = strtok_r(buffer, search, &brkb); //GPGGA header, no usado
token = strtok_r(NULL, search, &brkb); //UTC, no usado!!
token = strtok_r(NULL, search, &brkb); //lat, no usado!!
token = strtok_r(NULL, search, &brkb); //norte/sur, ...
token = strtok_r(NULL, search, &brkb); //lon, no usado!!
token = strtok_r(NULL, search, &brkb); //wets/east

```

```

token = strtok_r(NULL, search, &brkb); //Posicion fija, usado!!
fix_position = atoi(token);
token = strtok_r(NULL, search, &brkb); //satelites en uso!!
token = strtok_r(NULL, search, &brkb); //HDOP, No se necesita
token = strtok_r(NULL, search, &brkb); //ALTITUD, es el único significativo de esta cadena
alt = atoi(token); if(alt < 0)
{
alt = 0;
}

if(fix_position >= 0x01) digitalWrite(BLUE_LED,HIGH); //Estado de LED...
else
digitalWrite(BLUE_LED,LOW);

gps_new_data_flag |= 0x02; // Actualización de la indicador para señalar los nuevos datos ha llegado.
}
checksum=0; //Reseteando el checksum
} // final de análisis $ GPGGA

for(int a=0; a<=counter; a++)//Reseteando el buffer
{
buffer[a]=0;
}
counter=0; //Reseteando el contador
}
else
{
counter++; //Incremento contador
}
}
}

}

/*****
*           // Función para calcular el curso entre dos waypoints
*****/

int get_gps_course(float flat1, float flon1, float flat2, float flon2)
{
float calc; float bear_calc;

float x = 69.1 * (flat2 - flat1);
float y = 69.1 * (flon2 - flon1) * cos(flat1/57.3);
calc=atan2(y,x);
bear_calc= degrees(calc);

if(bear_calc<=1){ bear_calc=360+bear_calc;

```

```

}
return bear_calc;
}

/*****
*           // Función para calcular la distancia entre dos puntos dereferencia
*****/

unsigned int get_gps_dist(float flat1, float flon1, float flat2, float flon2)
{

float x = 69.1 * (flat2 - flat1);
float y = 69.1 * (flon2 - flon1) * cos(flat1/57.3);

return (float)sqrt((float)(x*x) + (float)(y*y))*1609.344;
}

/*****
// Calcula el error, y elegir el camino más corto para llegar al rumbo deseado
*****/

int compass_error(int PID_set_Point, int PID_current_Point)
{
float PID_error=0;//Temporary variable

if(fabs(PID_set_Point-PID_current_Point) > 180)
{

if(PID_set_Point-PID_current_Point < -180)
{
PID_error=(PID_set_Point+360)-PID_current_Point;
}
else
{

}

else
{
PID_error=(PID_set_Point-360)-PID_current_Point;
}
PID_error=PID_set_Point-PID_current_Point;
}
return PID_error;
}

// Esta función detiene toda la actividad
// Este es el final...

void finish_mission(void)

```

```

{
bldc_stop_throttle();

#ifdef RADIO_CONTROL
switch_to_radio(); // Dar control al Radio
#endif

while (1) // bucle para siempre, si alcanza el tiempo de espera (y empezar a navegar y recuperar el barco)

{
digitalWrite(YELLOW_LED,LOW); //Parpadeo rápido LED amarillo para indicar la llegada
delay(100);
digitalWrite(YELLOW_LED,HIGH);
delay(100);
}
}

/*****
* Control de timón, lee información GPS, calcula navegación, ejecuta PID y envía los valores al servo.
*****/
void rudder_control(void)
{
wp_bearing=get_gps_course(lat, lon, wps[current_wp].lat, wps[current_wp].lon);//Calculando el rumbo, esta
función está localizada en el GPS_Navigation tab..

wp_distance = get_gps_dist(lat, lon, wps[current_wp].lat, wps[current_wp].lon);// Cómo calcular la distancia, esta
función se encuentra en el GPS_Navigation tab..

rudder_setpoint = MIDDLE_RUDDER+PID_heading(compass_error(wp_bearing, course));

// Posición central + PID (compass_error (curso deseado, curso actual)).
pulse_servo_rudder((long)rudder_setpoint); // El envío de valores de servo, 90 grados es la posición central.

}
// Esta función apaga la EM406 en 9600 baudios
// Normalmente, los valores predeterminados EM406 a NMEA 4800 baudios y después de largos tiempos de
apagado

void gps_init_baudrate(void)
{
Serial.begin(4800); // Siempre pruebe en 4800 baudios primero.
delay(100);
Serial.println("$PSRF100,1,9600,8,1,0*0D"); // comando para cambiar SIRFIII a NMEA, 9600, 8, N, 1
delay(100);
Serial.begin(9600); // conmutar finalmente de vuelta a 9600 baudios

```

}

f) Módulo PID

Este módulo contiene la función que hace el lazo de control de la navegación en línea recta. Este es uno de los módulos más importantes, ya que utiliza un algoritmo de PID.

El comportamiento del bucle PID es controlada por las constantes que se describen en la sección de "Las constantes PID."

// Variables de bucle PID

```
int heading_previous_error;
float heading_I=0.0; // Almacena el resultado del integrador (I)

/*****
*           PID = P + I + D Esta función sólo funciona cuando se utiliza el GPS con una sola segunda
*           actualización.
*****/
int PID_heading(int PID_error)
{
    static float heading_D; // Almacena el resultado del derivador (D)
    static float heading_output; // Almacena el resultado del bucle PID

    dbg_pid_error = PID_error; //Depurar
    heading_I += (float)PID_error;
    heading_I = constrain(heading_I, -INTEGRATOR_LIMIT, INTEGRATOR_LIMIT); //Limite del PID integrador

//Parte de derivacion
    heading_D = ((float)PID_error - (float)heading_previous_error);
    heading_output = 0.0; // Borrado de la variable.
    heading_output = (KP_HEADING * (float)PID_error); // Parte proporcional, es sólo el error * constante KP .. y
        añadido a la salida

    pid_p = (KP_HEADING * (float)PID_error);

    heading_output += (KI_HEADING * heading_I); // Adición resultado integrador...
    pid_i = (KI_HEADING * heading_I);

    heading_output += (KD_HEADING * heading_D); // Adición resultado derivador...
    pid_d = (KD_HEADING * heading_D);

// Añade todos los resultados de PID y limita la salida ...
```

```
heading_output = constrain(heading_output, (float)HEADING_MIN, (float)HEADING_MAX); // limitando la
    salida....
```

```
heading_previous_error = PID_error;// Guardar el error actual y se usa más tarde (en parte Derivación) ...
```

```
// Ahora comprobar si el usuario ha seleccionado el modo normal o inversa (servo) ...
```

```
if(REVERSE_RUDDER == 1)
{
return (int)(-1 * heading_output);
}
else
{
return (int)(heading_output);
}
}
```

```

/*****
*           Reset todos los PID
*****/
void reset_PIDs(void)
{
heading_previous_error = 0.0;
heading_I = 0.0;
}

```

g) Módulo Servomotor

Este módulo contiene las funciones de los pulsos de salida al servo del timón y del ESC (acelerador) para el motor de impulsión.

Este módulo utiliza la unidad de modulación de ancho de pulso (PWM) del microcontrolador AVR que se utiliza en las plataformas Arduino.

```

/*****
*           Configurar el PWM hardware,
*           Leer la hoja de datos del atmega168.
*           Las funciones operativas están optimizados para velocidad. La biblioteca Arduino Servo
*           Puede que no funcione, porque consume más tiempo de procesamiento,
*****/

void Init_servo(void)// Esta parte va a configurar el PWM para controlar el amplificador 100% por hardware, y no
    perder tiempo de CPU.

{
digitalWrite(RUDDER,LOW);// Definir los pines de salida del servo

```

```
pinMode(RUDDER,OUTPUT);
digitalWrite(THROTTLE,LOW);
pinMode(THROTTLE,OUTPUT);
```

```
/* Configuración de temporizador PWM rápido */
```

```
//Nota: estas cadenas extrañas que siga, como OCRI,1A, son en realidad Atmega168 registros predefinidos.
```

```
// Cargamos los registros y el chip hace el resto.
```

```
// Recuerde los registros no declara aquí permanecen a cero por defecto...
```

```
TCCR1A = ((1<<WGM11)|(1<<COM1B1)|(1<<COM1A1)); // Segun la página 131 de hoja de datos, estamos
cambiando los registros de configuración WGM11,COM1B1,COM1A1 a 1 eso es todo..
```

```
TCCR1B = (1<<WGM13)|(1<<WGM12)|(1<<CS11); // Divisor en 8, que nos dan una resolución de 2, leer la
página 134 de hoja de datos
```

```
OCR1A = 2000; // El período de servo 1, recuerde 2us resolución 2000/2 = 1000 us el período de pulso de la servo...
```

```
OCR1B = 3000; // El período de servo 2, 3000/2 =1500, más o menos es la posición central...
```

```
ICR1 = 40000; //50Hz ... la Hoja datos dice (system_freq/prescaler)/target frequency. Por lo tanto (16000000hz/ 8)
/ 50hz=40000,
```

```
// Debe ser de 50hz porque es el servo estándar (cada 20 ms, y 1 hz = 1seg) 1000ms/ 20ms= 50hz
```

```
}
```

```
/******
```

```
* Función de pulso del servo del acelerador
```

```
/******
```

```
void pulse_servo_throttle(long angle) // Converte el ángulo equivalente a posición del servo del acelerador..
```

```
{
```

```
//angle=constrain(angle,180,0);
```

```
OCR1A = ((angle * (MAX16_THROTTLE - MIN16_THROTTLE)) / 180L + MIN16_THROTTLE) * 2L;
```

```
}
```

```
/******
```

```
* Función de pulso del servo de orientación / timón de dirección...
```

```
/******
```

```
void pulse_servo_rudder(long angle) // Convierte el ángulo equivalente a la posición del servo timon...
```

```
{
```

```
OCR1B = ((angle *(MAX16_RUDDER - MIN16_RUDDER)) / 180L + MIN16_RUDDER) * 2L;
```

```
}
```

```
void bldc_arm_throttle(void) // Inicializar controlador del BLDC (motor Brushless DC)
```

```
{
```

```
delay(2000);
```

```
bldc_stop_throttle(); // Luego cambiar a aprox. cero, el controlador del BLDC (motor Brushless DC).
```

```
delay(4000);
```

```
}
```

```
void bldc_start_throttle(void) // Encender Motor brushless (controlador de Múltiplex)
```



```

{
pulse_servo_throttle(MOTOR_SPEED); // Régimen del Motor
}

// function para detener el motor
// Motor brushless (controlador de Múltiplex)

void bldc_stop_throttle(void) // Detener Motor brushless (controlador de Múltiplex)

{
pulse_servo_throttle(MOTOR_OFF); // cambiar a aprox. cero
}

void test_rudder(void) // prueba de timon
{
pulse_servo_rudder(MIDDLE_RUDDER+ HEADING_MIN);
delay(1500);
pulse_servo_rudder(MIDDLE_RUDDER+ HEADING_MAX);
delay(1500);
pulse_servo_rudder(MIDDLE_RUDDER);
delay(1500);
}

// Módulo para controlar el ArduPilot a través de la Radio Control (RC)

// Return 0 if no pulse available (timeout > 25ms)

int check_radio(void)
{
return (int) pulseIn(SERVO2_IN_PIN, HIGH, 25000); // Compruebe, si hay pulsos en el timón de entrada Rx;
}

// Función para cambiar el multiplexor al ArduPilot

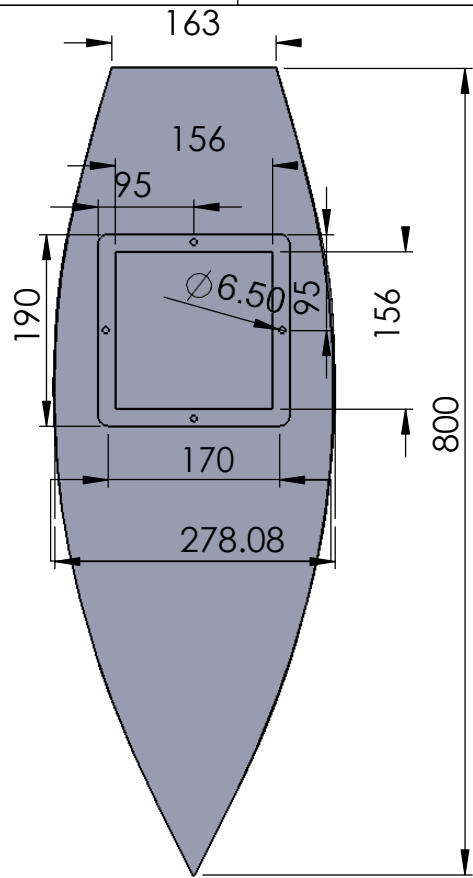
void switch_to_ardupilot (void)
{
digitalWrite(MUX_PIN,HIGH); // Servos controlados por ArduPilot
}

// Función para cambiar el multiplexor al Receptor RC
void switch_to_radio (void)
{
digitalWrite(MUX_PIN,LOW); // servos controlados por radio control
}

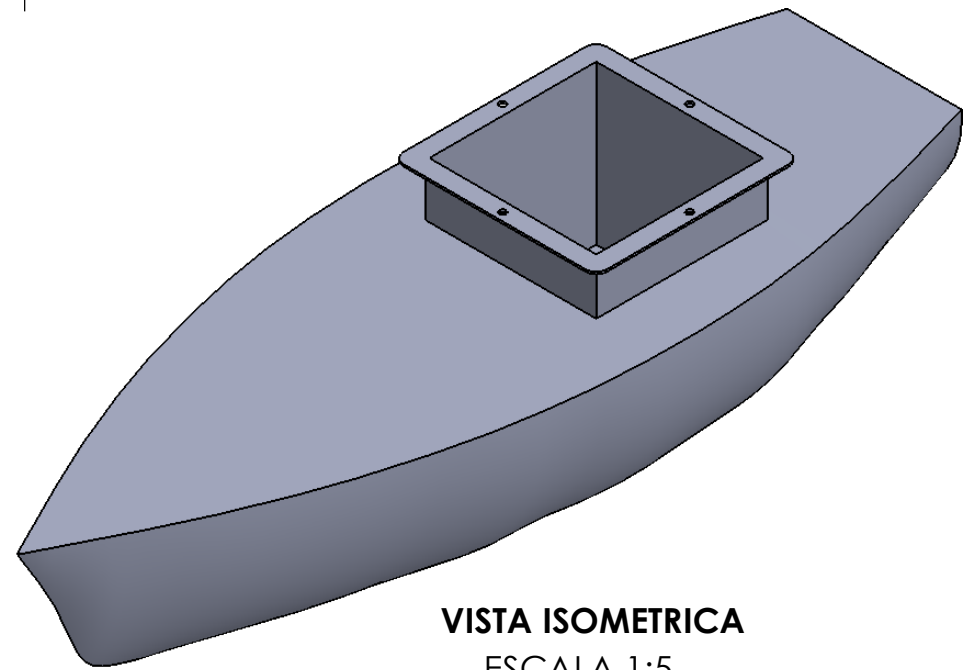
```

ANEXO 3: PLANOS

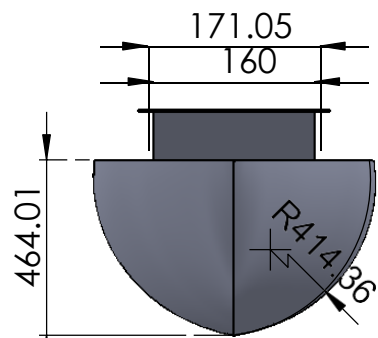




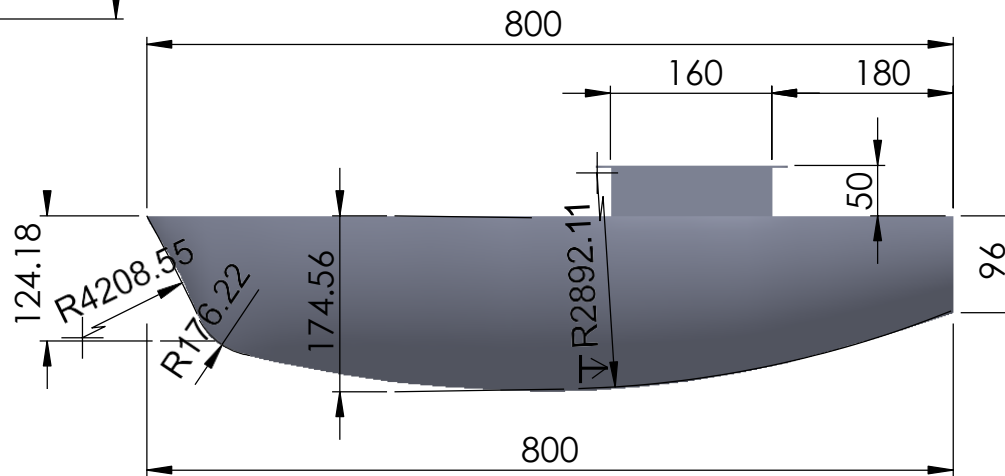
ESCALA 1:7.5



VISTA ISOMETRICA
ESCALA 1:5

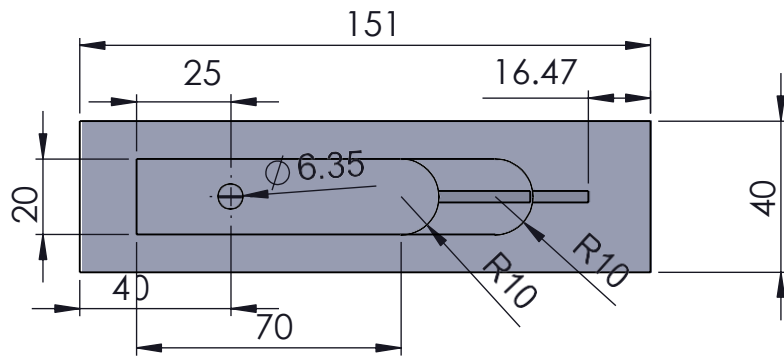


ESCALA 1:7.5

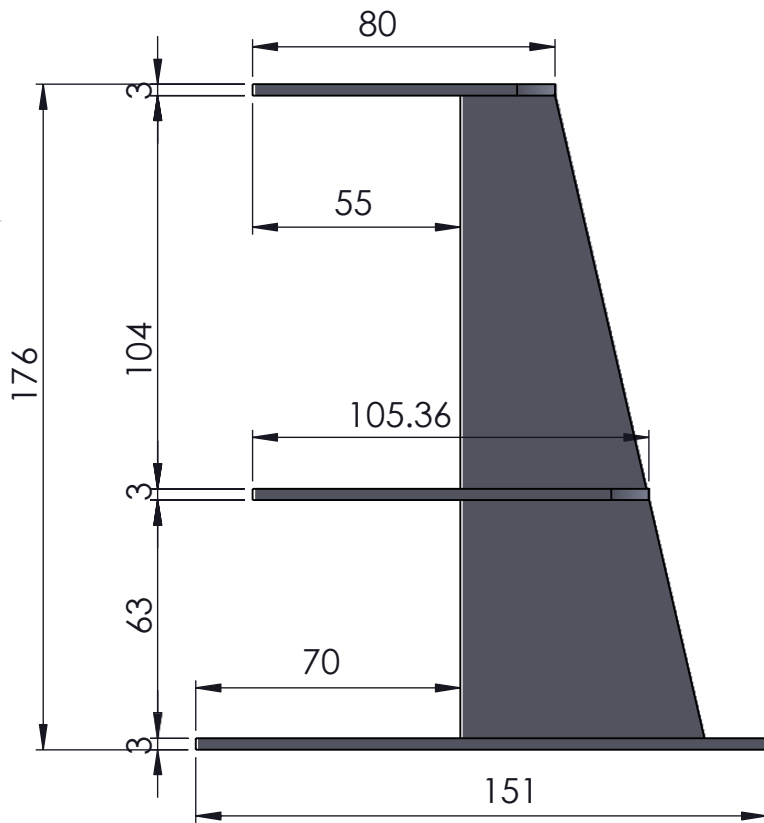


ESCALA 1:7.5

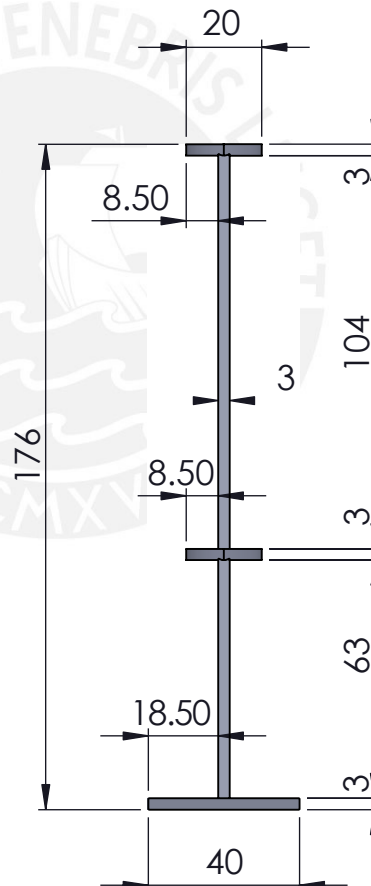
PONTIFICIA UNIVERSIDAD CATOLICA DEL PERU		
ESCUELA DE GRADUADOS		ING. MECATRONICA
Proyeccion: 	Plano: Estructura principal	Lamina: PIEZA 1
Codigo: A20035018	Curso: Seminario de tesis II	Fecha: May-2016
Dibujado: L.S.C.P.	Alumno: L.S.C.P.	Escala: 1:10



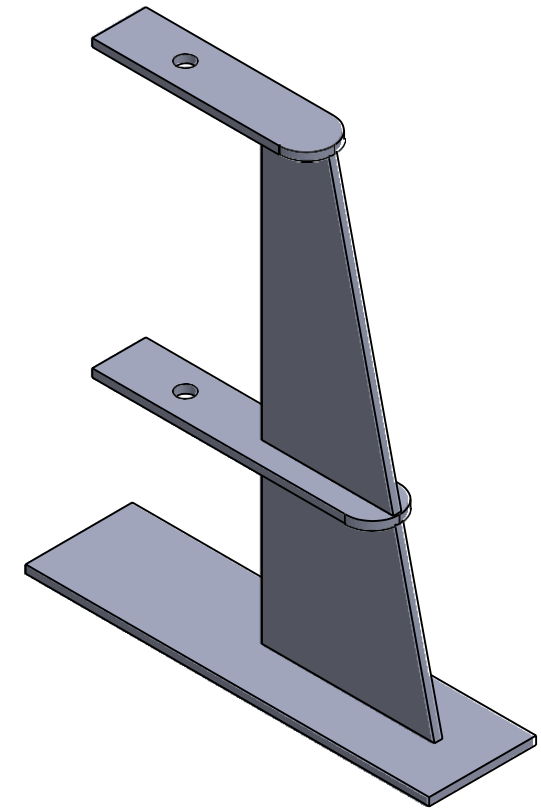
ESCALA 1 : 2



ESCALA 1 : 2



ESCALA 1 : 2



VISTA ISOMETRICA
ESCALA: 1 : 2

NOTA:
Los elementos sera unidos soldadura.

PONTIFICIA UNIVERSIDAD CATOLICA DEL PERU

ESCUELA DE GRADUADOS

ING. MECATRONICA

Proyeccion:

Plano:

Lamina:



Soporte de Helice

PIEZA 2

Codigo:

Curso:

Fecha:

A20035018

Seminario de tesis II

May-2016

Dibujado:

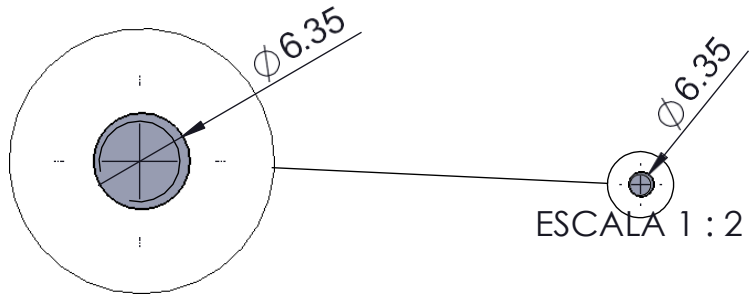
Alumno:

Escala:

L.S.C.P.

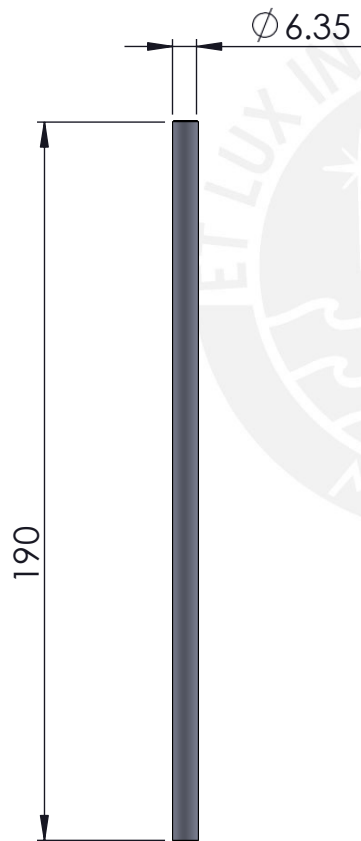
L.S.C.P.

1:2



DETALLE A
ESCALA 2 : 1

ESCALA 1 : 2



ESCALA 1 : 2



VISTA ISOMETRICA
ESCALA 1 : 2

PONTIFICIA UNIVERSIDAD CATOLICA DEL PERU

ESCUELA DE GRADUADOS

ING. MECATRONICA

Proyeccion:

Plano: EJE DE

Lamina:



SOPORTE DE
TIMON

PIEZA 1

Codigo:

Curso: Seminario de tesis II

Fecha:

A20035018

Alumno:

MAYO-2016

Dibujado:

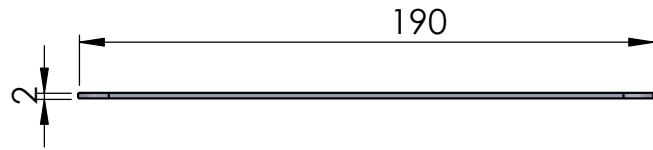
L.S.C.P.

Escala:

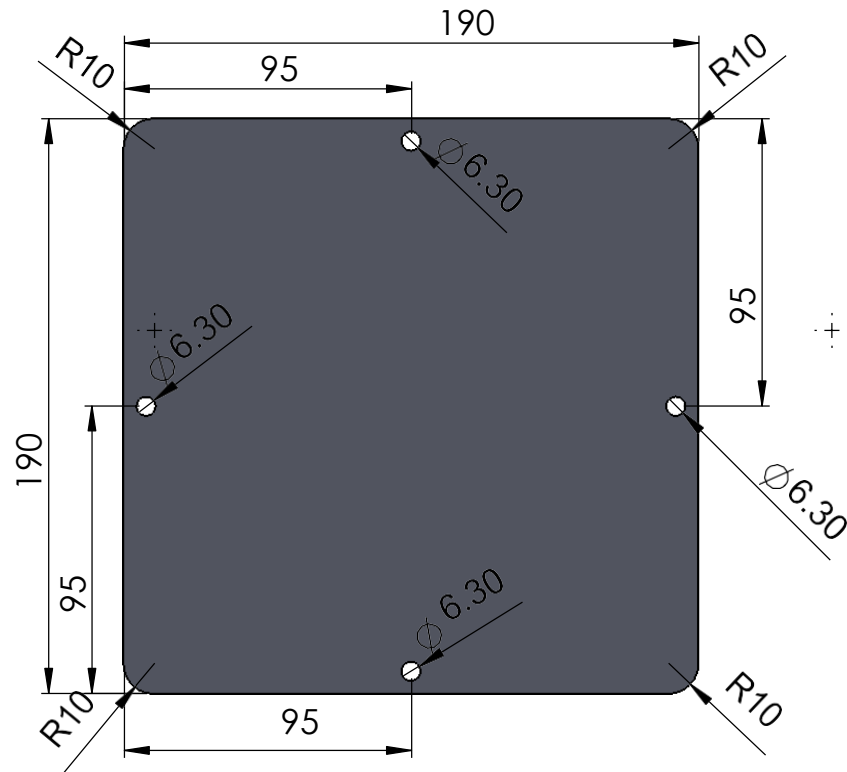
L.S.C.P.

L.S.C.P.

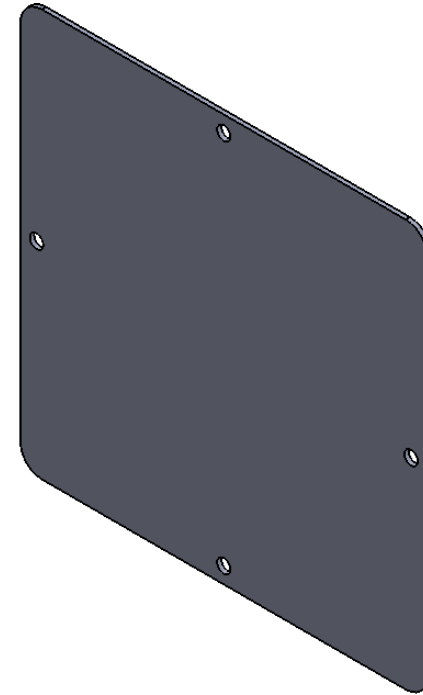
1:5



ESCALA 1:2.5



ESCALA 1:2.5



VISTA ISOMETRICA
ESCALA 1:2.5

PONTIFICIA UNIVERSIDAD CATOLICA DEL PERU

ESCUELA DE GRADUADOS

ING. MECATRONICA

Proyeccion:

Plano:

Lamina:



COBERTURA
- TAPA

PIEZA 4

Codigo:

Curso:

Fecha:

A20035018

Seminario de tesis II

May-2016

Dibujado:

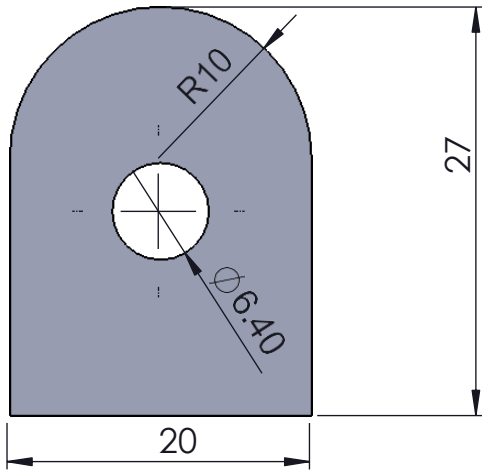
Alumno:

Escala:

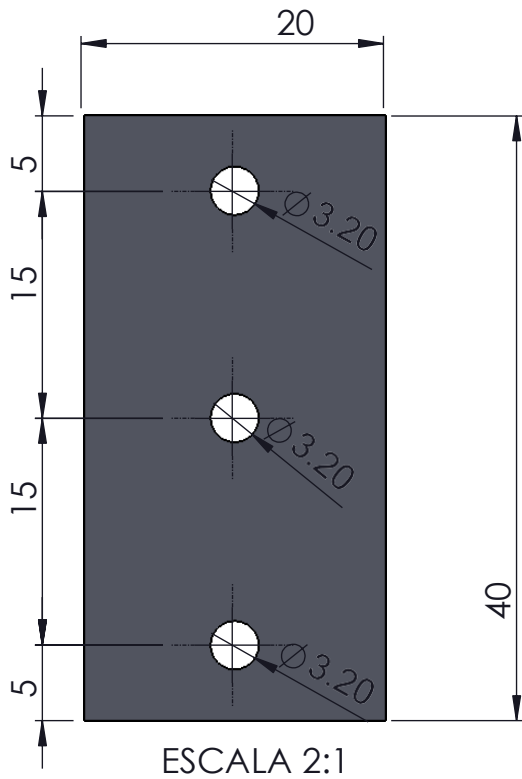
L.S.C.P.

L.S.C.P.

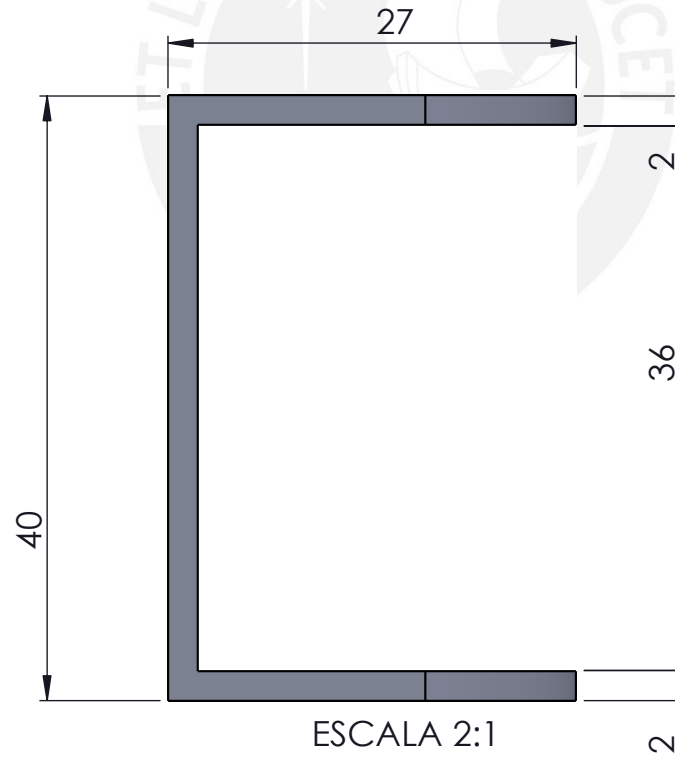
1: 2.5



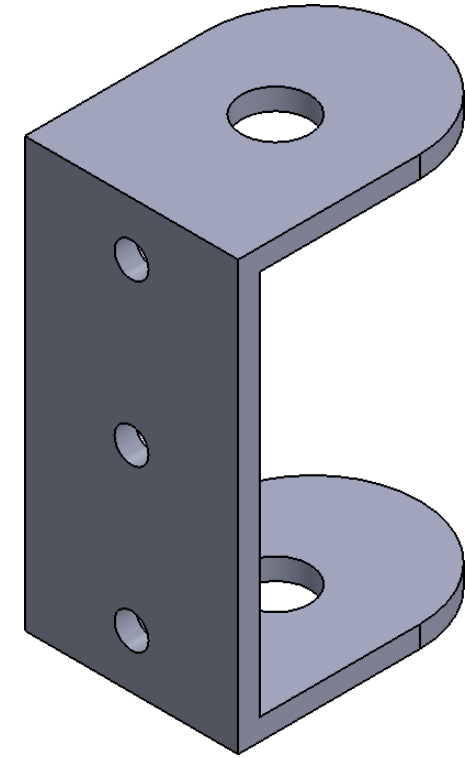
ESCALA 2:1



ESCALA 2:1



ESCALA 2:1



VISTA ISOMETRICA
ESCALA 2:1

PONTIFICIA UNIVERSIDAD CATOLICA DEL PERU

ESCUELA DE GRADUADOS

ING. MECATRONICA

Proyeccion:

Plano:

Lamina:



SOPORTE DE MOTOR

PIEZA 5

Codigo:

Curso:

Fecha:

A20035018

Seminario de tesis II

May-2016

Dibujado:

Alumno:

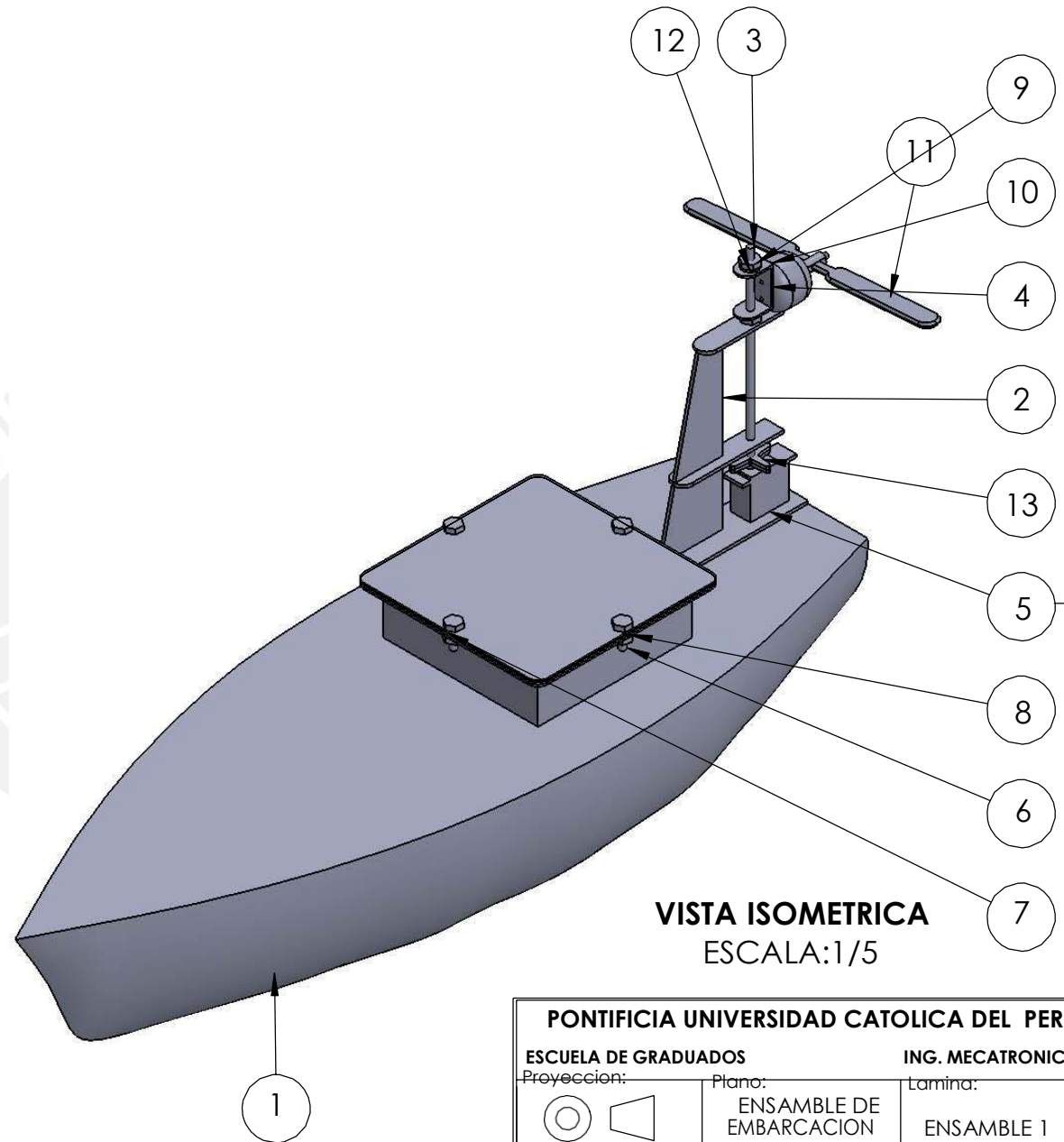
Escala:

L.S.C.P.

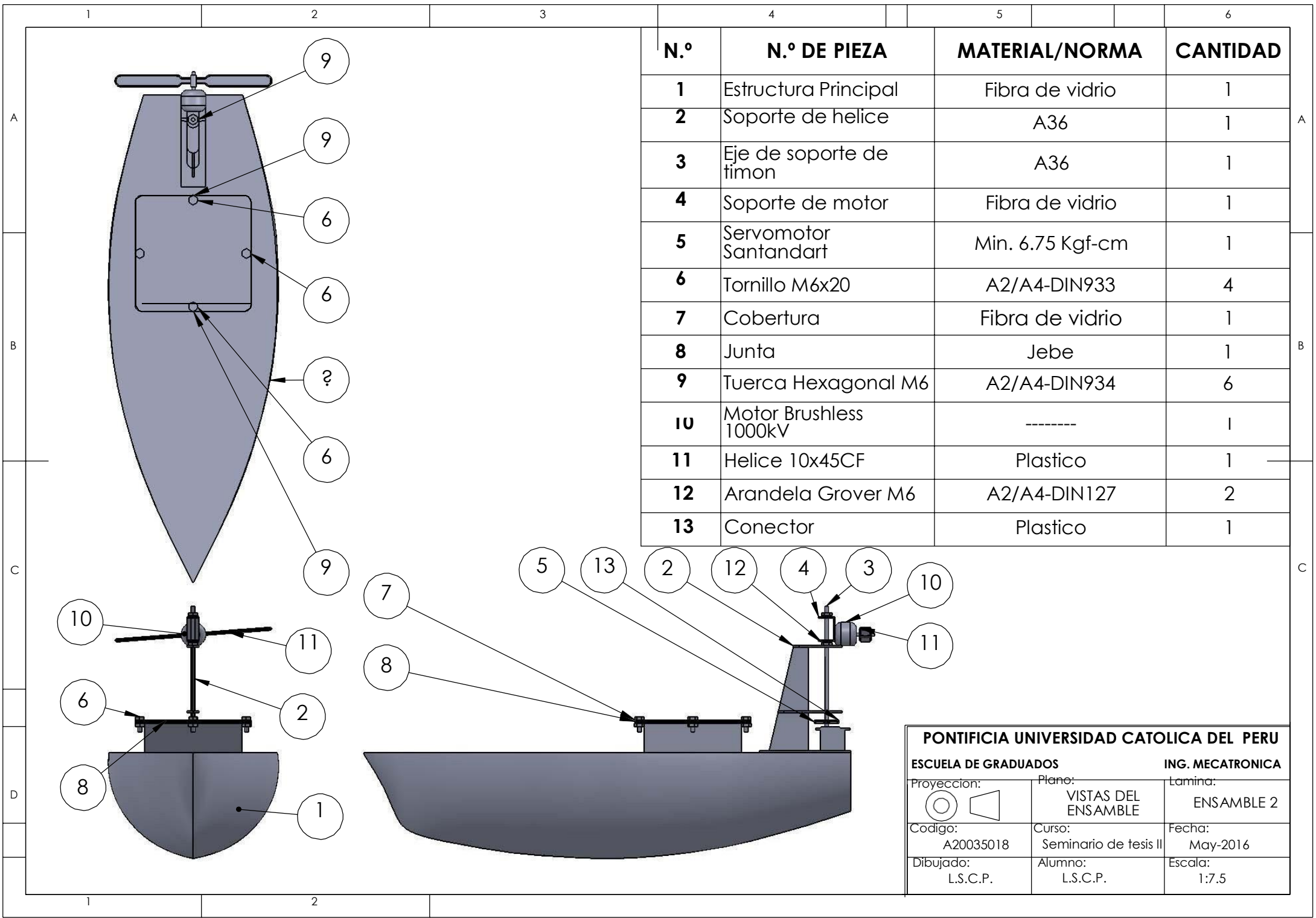
L.S.C.P.

2:1

N.º	PIEZA	MATERIAL/ NORMA	CANTIDAD
1	Estructura Principal	Fibra de vidrio	1
2	Soporte de helice	A36	1
3	Eje de soporte de timon	A36	1
4	Soporte de motor	Fibra de vidrio	1
5	Servomotor Santandart	Min. 6.75 Kgf-cm	1
6	Tornillo M6x20	A2/A4-DIN933	4
7	Cobertura	Fibra de vidrio	1
8	Junta	Jebe	1
9	Tuerca Hexagonal M6	A2/A4-DIN934	6
10	Motor Brushless 1000kV	-----	1
11	Helice 10x45CF	Plastico	1
12	Arandela Grover M6	A2/A4-DIN127	2
13	Conector	Plastico	1



PONTIFICIA UNIVERSIDAD CATOLICA DEL PERU		
ESCUELA DE GRADUADOS		ING. MECATRONICA
Proyeccion:	Plano:	Lamina:
	ENSAMBLE DE EMBARCACION	ENSAMBLE 1
Codigo: A20035018	Curso: Seminario de tesis II	Fecha: May-2016
Dibujado: L.S.C.P.	Alumno: L.S.C.P.	Escala: 1:5



N.º	N.º DE PIEZA	MATERIAL/NORMA	CANTIDAD
1	Estructura Principal	Fibra de vidrio	1
2	Soporte de helice	A36	1
3	Eje de soporte de timon	A36	1
4	Soporte de motor	Fibra de vidrio	1
5	Servomotor Santandart	Min. 6.75 Kgf-cm	1
6	Tornillo M6x20	A2/A4-DIN933	4
7	Cobertura	Fibra de vidrio	1
8	Junta	Jebe	1
9	Tuerca Hexagonal M6	A2/A4-DIN934	6
10	Motor Brushless 1000kV	-----	1
11	Helice 10x45CF	Plastico	1
12	Arandela Grover M6	A2/A4-DIN127	2
13	Conector	Plastico	1

PONTIFICIA UNIVERSIDAD CATOLICA DEL PERU
ESCUELA DE GRADUADOS **ING. MECATRONICA**

Proyeccion: 	Plano: VISTAS DEL ENSAMBLE	Lamina: ENSAMBLE 2
Codigo: A20035018	Curso: Seminario de tesis II	Fecha: May-2016
Dibujado: L.S.C.P.	Alumno: L.S.C.P.	Escala: 1:7.5

REFERENCIAS

- [1] Menezes Pereira, Arvind Antonio (2007) Navigation and Guidance of an Autonomous Surface vehicle [Navegación y orientación de un vehículo autónomo superficie], California: University of Southern California.
- [2] Karlik, Evan A. (2007) Remote Depth survey of the Charles river Basin [Medición de profundidad remota de la cuenca del río Charles], Massachusetts: Massachusetts Institute of Technology.
- [3] J. Pereda, Fernando. (2010) Sistema de telemetría y control de un barco autónomo, Madrid: Universidad Complutense de Madrid.
- [4] Soares dos Santos, Douglas (2011) Projeto e Construção de um Barco Inteligente com Integração INS/GPS e Bússola [Diseño y construcción de un barco inteligente Compuesto de INS/GPS y brújula], São José dos Campos: Instituto Tecnológico de Aeronáutica.
- [5] Ullah, Sajid (2012) Autonomous Surveying Boat [Barco de prospección autónoma], Islamabad Pakistan: Institute of Space and Technology.
- [6] Oliveira de Arruda, José Wilton (2012) Controle Automático de Rumo de uma Embarcação de Superfície não Tripulada [Control automático del rumbo de un buque de superficie no tripulado], Rio de Janeiro: Universidade do Estado do Rio de Janeiro.
- [7] Hendrik, Jürgen Schultze (2012) Projeto e Construção de uma Embarcação Teleoperada [Diseño y construcción de una embarcación Teleoperada], Rio de Janeiro: Universidade do Estado do Rio de Janeiro.
- [8] Rasal, Ketan (2013) Navigation & Control of an Automated SWATH Surface Vessel for Bathymetric Mapping [Navegación y control de un buque de superficie SWATH automatizada de Cartografía batimétrica], Santa Clara: Santa Clara University.
- [9] Stenersen, Thomas (2015) Guidance System for Autonomous Surface Vehicles [Sistema de dirección para vehículos autónomos de superficie], Trondheim: Norwegian University of Science and Technology.
- [10] <http://arduinodhtics.weebly.com/historia.html> [Online].

- [11] <http://www.electan.com/ardupilot-arduino-compatible-uav-controller-atmega328-p-3829.html> [Online].
- [12] ArduPilot 2.x manual.
<http://tech-insider.org/diy-drones/research/acrobat/090919.pdf> [Online].
- [13] http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf [Online].
- [14] <http://electronicarc.com/catalogo/images/coneccionArduPilot.jpg> [Online].
- [15] http://cdn.sparkfun.com/datasheets/GPS/EM506_um.pdf [Online].
- [16] <http://ardupilot.org/rover/docs/common-powermodule-landingpage.html> [Online].
- [17] Sánchez Carrión, Felipe (2012) Diseño y Construcción de un Modelo experimental de tricóptero controlado de forma remota mediante radiofrecuencia, Ecuador: Escuela Politécnica Nacional-facultad de Ingeniería Eléctrica y Electrónica.
- [18] <http://www.drivecalc.de/PropCalc/index.html> [Online].
- [19] <http://www.aerodelismoelcan.com/component/content/article/2-uncategorised/73-que-es-el-bec> [Online].
- [20] <http://rctimer.com/product-779.html> [Online].
- [21] <http://www.myrcmart.com/rcx-a283011-1000kv-outrunner-brushless-motor-airplane-free-mounts-p-3722.html> [Online].
- [22] <http://www.adafruit.com/product/155> [Online].
- [23] <http://www.delftship.net/> [Online].
- [24] Pérez T. y Blanke M. "Mathematical Ship Modeling for Control Applications", University of Newcastle, Australia.
- [25] [Triantafyllou et al., 2003]. Triantafyllou, Michael S., Hover, Franz S. (2003).
- [26] K. Nomoto, K. Taugchi, K. Honda and S. Hirano, 1957: "On the steering qualities of ships". International Shipbuilding Progress.
- [27] Vidar Ødegård, 2009: "Nonlinear Identification of Ship Autopilot Models" ,Norwegian University of Science and Technology Department of Engineering Cybernetics, Noruega.
- [28] Fossen, T. I. Handbook of Marine Craft Hydrodynamics and Motion Control. (2011). Wiley.
- [29] https://es.wikipedia.org/wiki/F%C3%B3rmula_del_Haversine [Online].

[30] Mridul Pande, K K Mangrulkar, Design of a Heading Autopilot for Mariner Class Ship with Wave Filtering Based on Passive Observer, 1,2Aerospace Engg Dept DIAT (DU), Pune – La India

[31] Óscar Torrente Artero, Arduino. Curso práctico de formación (2013) Alfaomega Grupo Editor, S.A. de C.V., México.

[32] <http://arduino.cc/en/Main/Software> [Online].

[33] <https://support.google.com/earth/answer/148176?hl=es&rd=1> [Online].

[34] <http://ocw.upm.es/ingenieria-cartografica-geodesica-y-fotogrametria/topografia-ii> [Online].

