



**UNIVERSIDAD ESTATAL  
PENÍNSULA DE SANTA ELENA**

**FACULTAD DE SISTEMAS Y  
TELECOMUNICACIONES**

**CARRERA DE ELECTRÓNICA Y  
TELECOMUNICACIONES**

**TRABAJO DE TITULACIÓN**

Propuesta Tecnológica, previo a la obtención del Título de:

**INGENIERO EN ELECTRÓNICA Y  
TELECOMUNICACIONES**

“Prototipo de robot submarino autónomo tipo torpedo para  
recopilación de datos oceanográficos”

**AUTOR**

MONTOYA VAQUE CARLOS VINICIO

**PROFESOR TUTOR**

ING. SENDEY AGUSTÍN VERA GONZÁLEZ, MACI

LA LIBERTAD – ECUADOR

2019

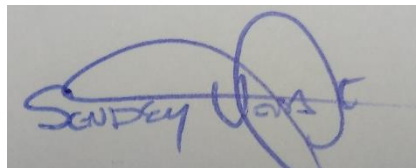
## **AGRADECIMIENTO**

Agradezco a Dios por permitirme vivir este momento anhelado, a mi familia que han sido un pilar fundamental para mi formación académica, a mi mamá por formarme desde pequeño a ser una persona de bien y a luchar por mis objetivos, a mi amada esposa que ha estado apoyándome y dándome sus palabras de aliento para seguir en el camino y conseguir llegar a la meta. Al Ingeniero Sendey Vera G. que junto al Ingeniero Arturo Cadena me guiaron académicamente aportando ideas y siendo los guías para la finalización del proyecto de titulación. A todos ustedes Muchas Gracias.

## APROBACIÓN DEL TUTOR

En mi calidad de Tutor del trabajo de titulación denominado **“Prototipo De Robot Submarino Autónomo Tipo Torpedo Para Recopilación De Datos Oceanográficos”**, elaborado por el estudiante **Montoya Vaque Carlos Vinicio**, de la carrera de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, me permito declarar que luego de haber orientado, estudiado y revisado, lo apruebo en todas sus partes y autorizo al estudiante para que inicie los trámites legales correspondientes.

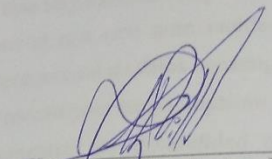
La Libertad, Enero del 2019



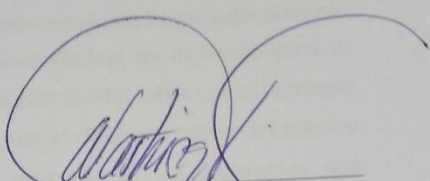
---

Ing. Sendey Vera González, MACI

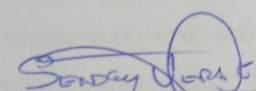
## TRIBUNAL DE GRADO



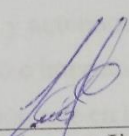
Ing. Freddy Villao Santos, MsC  
**DECANO DE FACULTAD**



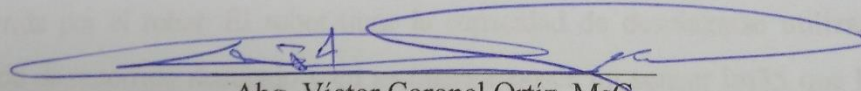
Ing. Washington Torres Guin, Msig  
**DIRECTOR DE CARRERA**



Ing. Sendey Vera González, MACI  
**PROFESOR TUTOR**



Ing. Luis Chuquimarca Jiménez, M.E.  
**PROFESOR DE ÁREA**



Abg. Víctor Coronel Ortiz, MsC  
**SECRETARIO GENERAL**

## RESUMEN

El presente documento presenta el diseño y construcción de un robot submarino tipo torpedo este equipo está diseñado de tal manera que pueda llegar a desplazarse en el mar con gran rapidez por su diseño hidrodinámico, es utilizado para la recolección de datos oceanográficos, se inicia el diseño 3D del robot con el software Freecad para luego imprimir las piezas más importantes de la estructura del robot y demostrar su flotabilidad e hidrodinámica, la mayoría de los elementos que componen la estructura se encuentran a nivel local como son tubos PVC, tapones PVC y ejes acerados. El diseño realizado posee 2 pares de timones que darán motricidad al robot, el dispositivo encargado de accionarlos son los servomotores que son dispositivos que dependiendo del ancho de pulso enviado se colocaran en posiciones diferentes que van desde los  $0^\circ$  a los  $180^\circ$ , los propulsores son de diseño profesional adquiridos para este trabajo, la parte de control y autonomía del robot está basada en el uso de software y hardware libre. Se diseñó e imprimió un mando inalámbrico para el control del robot para que se utilice cuando esté en la superficie, el robot submarino y el mando inalámbrico tendrán un alcance de máximo 200 metros, en la PCB de la tarjeta de control se realizó una etapa para cargar inalámbricamente un firmware, este proceso se lo realiza para evitar desmontar la parte electrónica de los recipientes herméticos cada vez que se necesite hacer algún cambio en el firmware de la tarjeta principal, además el mando inalámbrico se comunica con la aplicación móvil para presentar en mapas virtuales la trayectoria recorrida por el robot. El robot tiene la capacidad de desplazarse utilizando el método de recorrido deducido y en su interior posee un sensor lm35 que ha sido hermetizado en la parte de los pines para evitar que tengan contactos con el agua, con la finalidad de recolectar datos oceanográficos como temperatura y mostrar en la aplicación.

## **ABSTRACT**

This paper presents the design and construction of a torpedo underwater robot type, this equipment is designed so you can get to move into the sea very quickly on their hydrodynamic design, is used to collect oceanographic data, begins the 3D robot design in the FreeCad software and then print the most important pieces of the robot structure and demonstrate their buoyancy and hydrodynamics, most of the elements of the structure are found locally and are PVC pipes, plugs PVC and shafts steely. The designed has two pairs of rudders given the robot motor, the device manager to actuate the servomotors are devices that are depending on the width of pulse is sent placed in different positions ranging from  $0^\circ$  to  $180^\circ$ , propellants are professionally designed purchased for this work, the control part and robot autonomy is based on the use of free software and hardware. It was designed and printed a wireless controller for controlling the robot to be used when on the surface, the submarine robot and wireless controller will have a range of up to 200 meters on the PCB control board stage was conducted to wirelessly upload a firmware, this process is done to avoid removing the electronic part of the sealed containers whenever you need to make any changes to the firmware on the main board, plus wireless controller communicates with the mobile application to present maps virtual path traveled by the robot. The robot has the ability to navigate using the method of travel deducted and inside has a LM35 sensor which has been sealed on the part of the pins to prevent them from having contact with water, and in order to collect oceanographic data such as temperature and display in the application.

## DECLARACIÓN

El contenido del presente Trabajo de Graduación es de mi responsabilidad; el patrimonio intelectual del mismo pertenece a la Universidad Estatal Península de Santa Elena



.....  
Carlos Vinicio Montoya Vaque

## TABLA DE CONTENIDOS

ÍTEM	PÁGINA
AGRADECIMIENTO	I
APROBACIÓN DEL TUTOR	II
TRIBUNAL DE GRADO	III
RESUMEN	IV
ABSTRACT	V
DECLARACIÓN	VI
TABLA DE CONTENIDOS	VII
INTRODUCCIÓN	1
CAPÍTULO I	2
1 FUNDAMENTACION	2
1.1 ANTECEDENTES	2
1.2 DESCRIPCION DEL PROYECTO	3
1.3 OBJETIVOS	6
1.3.1 OBJETIVO GENERAL	6
1.3.2 OBJETIVOS ESPECIFICOS	6
1.4 RESULTADOS ESPERADOS	6
1.5 JUSTIFICACIÓN	7
1.6 ALCANCE DEL PROYECTO	8
1.7 METODOLOGÍA	8
CAPÍTULO II	9
2 LA PROPUESTA	9
2.1 MARCO CONTEXTUAL	9
2.2 MARCO CONCEPTUAL	11
2.2.1 ROBOT	11
	VII



2.2.2 SUBMARINO	12
2.2.3 ROBOT SUBMARINO	12
2.2.5 ACTUADORES	15
2.2.6 DATOS OCEANOGRÁFICOS	15
2.2.7 COMUNICACIÓN INALÁMBRICA	16
2.2.8 HERMETICIDAD	16
2.2.9 HIDRODINÁMICA	16
2.2.10 DISEÑO 3D	17
2.2.11 IMPRESIÓN 3D	17
2.3 MARCO TEÓRICO	18
2.4 COMPONENTES DE LA PROPUESTA	19
2.4.1 COMPONENTES FÍSICOS	19
2.4.2 COMPONENTE LÓGICOS	26
2.5 DISEÑO DE LA PROPUESTA ESQUEMA	32
2.5.1 DISEÑO ELECTRÓNICO	34
2.5.1.1 DISEÑO ELECTRONICO DEL ROBOT SUBMARINO	34
2.5.1.2 DISEÑO ELECTRÓNICO DEL CONTROL INALÁMBRICO	39
2.5.1.3 DISEÑO PLACA DE ROBOT SUBMARINO Y CONTROL	40
2.5.1.4 SISTEMA DE CONTROL	45
2.5.1.5 SENSORES	46
2.5.2 DISEÑO DE FIRMWARE	50
2.5.2.1 FIRMWARE DEL ROBOT SUBMARINO	51
2.5.2.2 FIRMWARE DEL CONTROL INALÁMBRICO	59
2.5.3 DISEÑO DE APLICACIÓN ANDROID	65
2.5.4 DISEÑO FÍSICO DEL ROBOT SUBMARINO	70
2.5.5 DISEÑO FÍSICO DEL CONTROL INALÁMBRICO	79

2.6 COSTO DEL PROYECTO	79
2.7 PRUEBAS	82
2.8 RESULTADOS	97
CONCLUSIONES	99
RECOMENDACIONES	101
BIBLIOGRAFÍA	102
ANEXOS	105

## ÍNDICE DE FIGURAS

ÍTEM	DESCRIPCIÓN	PÁGINA
Figura 1	Robot Submarino tipo torpedo	3
Figura 2	Diagrama de bloques propuesto para el Robot Submarino	5
Figura 3	Robot Submarino UPSE	9
Figura 4	Ubicación Ayangue en Google Maps	10
Figura 5	Ubicación de Salinas playa de Chipipe en Google Maps	11
Figura 6	ROV Hércules	14
Figura 7	AUV HUGIN	15
Figura 8	Programa FreeCAD	17
Figura 9	Impresora Creality Ender 3	18
Figura 10	Arduino Mega 2560 Mini	20
Figura 11	Modulo GPS NEO 6M	21
Figura 12	Módulo Xbee Pro S1	22
Figura 13	Módulo Bluetooth	23
Figura 14	Sensor 10DOF	24
Figura 15	Motor Brushless A2212/10T	24
Figura 16	Motor T200 Truster	25
Figura 17	Controlador ESC 30A	26
Figura 18	Servomotor DS3218	26
Figura 19	Arduino IDE	27
Figura 20	Página de diseño App Inventor 2	28
Figura 21	Página Programación Por Bloques	28
Figura 22	Código QR de Aplicación	29
Figura 23	Programa XCTU	30
Figura 24	PCB Wizard	30
Figura 25	Diseño hecho en FreeCAD	31
Figura 26	Ultimaker Cura	32
Figura 27	Estructura Control Inalámbrico	33
Figura 28	Estructura Robot Submarino	33
Figura 29	Robot Submarino UPSE	34
Figura 30	Diagrama eléctrico protección de los motores	35

Figura 31 Circuito Reductor de voltaje 7,4v a 5v	36
Figura 32 Circuito reductor de voltaje incluido alimentación a Arduino	37
Figura 33 DB15 Hembra y macho respectivamente	37
Figura 34 Conexión Bluetooth Para Carga de Sketch Inalámbricamente	39
Figura 35 Diseño Pistas para PCB del Robot Submarino	40
Figura 36 Diseño de la PCB del robot	40
Figura 37 Diseño pista PCB Control Inalámbrico	41
Figura 38 Diseño del PCB del Control	41
Figura 39 Diseño Placa Robot Submarino	42
Figura 40 Diseño Placa Control	42
Figura 41 Baquelita Limpia Luego de Pasarla por Cloruro Férrico	43
Figura 42 Placa Robot terminada de soldar	43
Figura 43 Placa Robot submarino	44
Figura 44 Placa Control Inalámbrico	44
Figura 45 Diodos Led Torre de Comunicaciones	45
Figura 46 Esquema De Conexiones De Los Sensores Y Actuadores Del Robot Submarino	45
Figura 47 Diagrama Conexión Bluetooth En Fritzing	46
Figura 48 Diagrama Conexión del GPS en el Robot	47
Figura 49 Diagrama Conexión del GPS en el Control	47
Figura 50 Diagrama Conexión del LSM303 en el Robot	48
Figura 51 Diagrama Conexión del Xbee en el Robot	49
Figura 52 Diagrama Conexión del Xbee en el Control	49
Figura 53 Diagrama Conexión del Sensor LM35 en el Robot	50
Figura 54 Diagrama de Flujo Robot Submarino	52
Figura 55 Robot sumergiendo	58
Figura 56 Robot emergiendo	58
Figura 57 Diagrama de Flujo Control Remoto	60
Figura 58 Cálculo Distancia y Dirección	64
Figura 59 Pantalla N°1 de la Aplicación Carátula	65
Figura 60 Pantalla N°2 de la Aplicación Página Principal con Menú Desplegable	65

Figura 61	Página de Ubicación en la Aplicación Móvil	66
Figura 62	Página del Rumbo en la Aplicación Móvil	67
Figura 63	Programación Pagina Carátula de la Aplicación Móvil	67
Figura 64	Programación Recepción de datos Aplicación Móvil	68
Figura 65	Programación Para el Menú desplegable	68
Figura 66	Programación para pantalla Ubicación	69
Figura 67	Corte longitudinal del Tubo de 6 pulgadas	70
Figura 68	Ojiva Delantera y Ojiva Trasera	71
Figura 69	Relación de Profundidad con Presión	71
Figura 70	Niple con Tapón	72
Figura 71	Tapa de Envases Herméticos Diseñada en 3D	73
Figura 72	Diseño de conexiones entre elementos	73
Figura 73	Motor Brushless A2212 recubierto con Pegatanque	74
Figura 74	Motores Impresos en 3D	74
Figura 75	Motor T200 Thruster y Controlados ECS	75
Figura 76	Recipiente para servomotores	75
Figura 77	Elementos de Rotación para Timones	76
Figura 78	Timones	76
Figura 79	Robot submarino sin pintar y Anaranjado	77
Figura 80	Interior Torre de Comunicación	77
Figura 81	Tarjeta de Fusibles y Tarjeta de control de Robot Submarino	78
Figura 82	Ensamble de robot Submarino	78
Figura 83	Control Inalámbrico	79
Figura 84	Motores a velocidad 0	89
Figura 85	Motores a velocidad media	89
Figura 86	Motores a velocidad completa	89
Figura 87	Servomotor direccional a 0°	90
Figura 88	Servomotor direccional a 90°	90
Figura 89	Servomotor direccional a 180°	91
Figura 90	Servomotor inmersión a 0°	91
Figura 91	Servomotor inmersión a 90°	92
Figura 92	Servomotor inmersión a 180°	92

Figura 93 Robot Submarino Boyanza Neutral En Piscina UPSE	93
Figura 94 Peso Lastre Del Robot	94
Figura 95 Peso Del Robot sin Lastre	94
Figura 96 Puntos GPS tomados durante el recorrido	94
Figura 97 Robot Submarino Desplazándose	96
Figura 98 Perfil 3D Timones	106
Figura 99 Toberas para Motor 3D	106
Figura 100 Propela Diseño 3D	106
Figura 101 Ojiva Delantera Diseño 3D	106
Figura 102 Ojiva Posterior	107
Figura 103 Abrazadera Recipiente Hermético Diseño 3D	107
Figura 104 Soporte Tarjeta control	107
Figura 105 Tapa Recipientes Herméticos	108
Figura 106 Engranajes Servos Diseño 3D	108
Figura 107 Control Inalámbrico Diseño 3D	108
Figura 108 Fotos pruebas de flotabilidad en piscina UPSE	109

## ÍNDICE DE TABLAS

ÍTEM	DESCRIPCIÓN	PÁGINA
Tabla 1	Tabla de consumo total de corriente del Robot Submarino	35
Tabla 2	Tabla de Comandos Configuración Bluetooth para la carga inalámbrica	38
Tabla 3	Parametrización Servomotor Cabeceo	54
Tabla 4	Parametrización Servomotor direccional	55
Tabla 5	Costos del Hardware Robot Submarino y Control	79
Tabla 6	Costos del Software Robot Submarino y Control	80
Tabla 7	Gastos Material Recipientes Herméticos, Casco Y Control	80
Tabla 8	Costos de Módulos y Actuadores para Robot Submarino y Control	81
Tabla 9	Costo Elementos Electrónicos	81
Tabla 10	Costos de Viáticos	82
Tabla 11	Total de inversión	82
Tabla 12	Comparación coordenadas GPS Submarino Parte 1	83
Tabla 13	Comparación coordenadas GPS Submarino Parte 2	84
Tabla 14	Comparación coordenadas GPS Submarino Parte (1)	84
Tabla 15	Comparación coordenadas GPS Submarino Parte (2)	85
Tabla 16	Diferencia de Grados entre Sensor y Teléfono Móvil	87
Tabla 17	Comparación Sensor LM35 con Termómetro Digital	88
Tabla 18	Cuadro de Velocidad Respecto a Ancho de Pulso	89
Tabla 19	Cuadro Cambio Posición Dirección Respecto a Ancho de Pulso	90
Tabla 20	Cuadro Cambio Posición Inmersión Respecto a Ancho de Pulso	91
Tabla 21	Cuadro de pesos del Robot Submarino	93
Tabla 22	Distancia recorrida en forma Automática	95
Tabla 23	Distancia entre Punto Inicial y Final	95
Tabla 24	Velocidad Aparente entre puntos tomados	96
Tabla 25	Datos de Temperatura recibidos	96

## **ANEXOS**

### **Nº. Descripción**

Anexo 1 Diseños 3D

Anexo 2 Fotos Varias de pruebas realizadas en la Universidad

Anexo 3 Código del Firmware para el Robot Submarino

Anexo 4 Código del Firmware para el Mando Inalámbrico

Anexo 5 Características del Robot Submarino

Anexo 6 Informe Urkund



## INTRODUCCIÓN

El presente proyecto muestra el diseño, construcción y programación de un robot submarino tipo torpedo para recolección de datos oceanográficos, las pruebas se realizaron en la piscina de la Universidad Estatal Península De Santa Elena y en la ensenada de la playa de la comuna Ayangue.

El robot submarino posee comunicación inalámbrica hasta un máximo de 200 metros con el mando inalámbrico, la comunicación que existe es bidireccional, las maniobras que realiza el prototipo son enviadas desde el mando, con esta se puede maniobrar el robot submarino cuando este se encuentre en la superficie, en el mismo mando mediante una combinación de teclas procede a la inmersión y entrará a trabajar de manera autónoma.

Las pruebas experimentales utilizadas demuestran el buen funcionamiento del robot tanto en la superficie como en la inmersión, estas van desde la prueba con el servomotor, temperatura, GPS, telecomunicaciones y potencia, estas pruebas de validación buscan demostrar el porcentaje de efectividad de cada dispositivo utilizado en el robot, un caso es la utilización del sensor de temperatura con pruebas de calibración y validación del valor según la muestra recogida y comparada con un termómetro comercial.

Como resultado de este trabajo, se obtuvo un robot submarino tipo torpedo funcional, capaz de desplazarse a una velocidad de aproximadamente 2 nudos y orientarse con sistemas de posicionamientos georreferencial GPS, desarrolla una inmersión y desplazamiento que tiene como guía un magnetómetro, a medida que se desplaza almacena datos de temperatura según la ubicación en que se encuentre.

# CAPÍTULO I

## FUNDAMENTACIÓN

### 1.1 ANTECEDENTES

Solo el 5% de los océanos ha sido explorado, alimenta a más de 2000 millones de personas y provee más del 50% de nuestro oxígeno. El cambio climático juega un papel esencial en los procesos biológicos en el medio marino que puede incluso afectar la soberanía alimentaria de las naciones. La acidificación de los mares puede tener un impacto negativo en la flora y fauna marina. Para una planificación sustentable y amigable con nuestros mares se requiere información oceanográfica que no es fácil acceder debido a los desafíos tecnológicos y económicos de la exploración oceánica. El costo de operación de un buque oceanográfico puede llegar a costar USD 50.000,00 por día. Una herramienta esencial para la exploración de los océanos es el robot submarino tipo autónomo. Estos dispositivos pueden realizar parte de las funciones de un buque oceanográfico, pero a una fracción del costo.

En el Ecuador han existido expediciones con vehículos autónomos submarinos Gliders de fabricación extranjera, al Noroeste y Sureste de Isla Isabella en la provincia de Galápagos quienes son los encargados de recabar la información de conductividad, temperatura, profundidad, clorofila y oxígeno mediante sus sensores durante meses, estos datos se transmiten hasta el INOCAR (Instituto Oceanográfico de la Armada) en tiempo casi real a través de telecomunicación satelital con Iridium y procesado por los técnicos del Instituto (INOCAR, 2013).

Un robot submarino propietario de origen extranjero tiene la desventaja que es prácticamente una “caja negra” que solo puede ser modificada a discreción del fabricante privando a nuestros investigadores de Ciencias del Mar explotar esta tecnología de la manera más eficiente posible. Por ese motivo es conveniente contar con tecnología de robótica submarina de origen nacional basada en hardware y

software libre que se ajuste a las necesidades de la comunidad local de investigadores de Ciencias del Mar.

Es difícil determinar la primera persona que creó un vehículo submarino no tripulado, pero en los libros de historia se menciona a la empresa Luppis-Whitehead Automobile al crear el torpedo en Austria en 1864 quien desarrolló un PUV (vehículo subacuático programado) y a Dimitri Rebikoff en 1953 por crear al primer ROV (vehículo operado remotamente) de nombre puddle (Carlos Carvalho Witte, 2011), Desde ese entonces varios robots submarinos tele operados y autónomos han sido desarrollados.

## 1.2 DESCRIPCIÓN DEL PROYECTO

El proyecto consiste en construir un vehículo autónomo subacuático AUV por sus siglas en inglés de *Autonomous Underwater Vehicle*, nuestra versión tendrá una parte de control inalámbrico solo en la superficie y una vez dada la señal de automático procederá a recorrer la trayectoria subacuática previamente programada, el AUV tiene la arquitectura tipo torpedo.

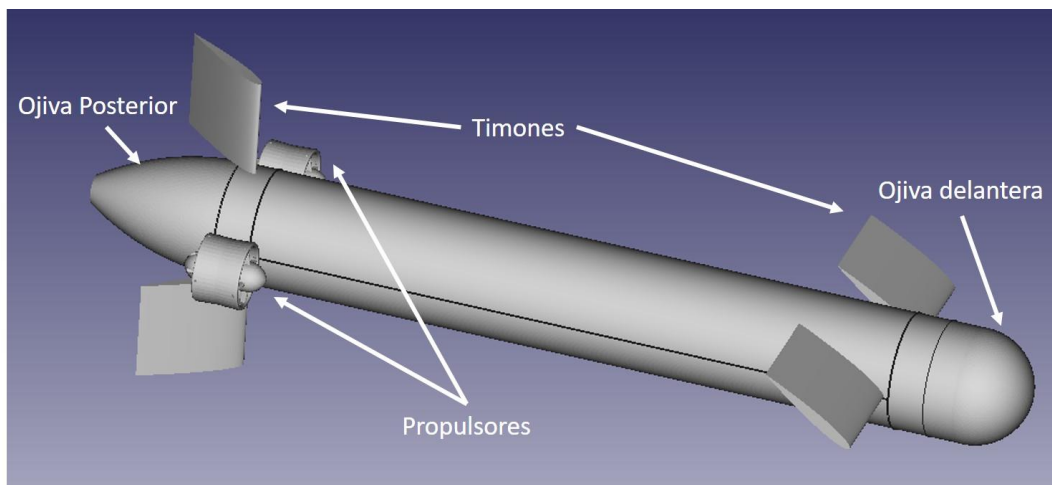


Figura 1 Robot Submarino tipo torpedo

Fuente: Elaborado por el Autor

El control remoto cuenta con un joystick que controla el giro de los servomotores y con ello la dirección que tome el AUV en la superficie, un potenciómetro lineal que controla el encendido, apagado y la velocidad con la que girarán los motores sin escobillas colocados en la parte trasera, posee también el NEO GPS 6m para poder ubicarnos según los datos geográficos de latitud y longitud, y para la comunicación inalámbrica se usaran 2 equipos de radiofrecuencia de bajo costo XBee emparejados, uno ubicado en el AUV y otro en el control remoto.

Se desarrolló una aplicación para teléfonos Android para monitorear los datos de posición, temperatura y del magnetómetro (brújula digital) que envíe el AUV cuando se encuentre en la superficie, se muestra en Google Maps la posición georreferenciada del robot y el control remoto trazando así una trayectoria hacia su posición mediante un cálculo aproximado de la distancia que se encuentra el submarino del control remoto, cada dato que se guarde desde la aplicación será almacenado en una tabla dinámica en Google Drive y automáticamente aparecerá la posición en nuestra aplicación, esto garantiza el manejo de la información que también podrá ser revisada de forma simultánea desde otro lugar en donde se tenga acceso a internet y a la misma cuenta registrada en Google Drive.

Cuando se dé la orden de inmersión el AUV comenzará a sumergirse, operando de forma totalmente autónoma sin comunicación con la superficie, cabe recalcar que tendrá la capacidad para bajar hasta 50 m y se desplazará por una ruta previamente programada, el robot se desplazara mediante recorrido deducido el cual consiste en darle una ruta previamente programada para que se desplace en los sentidos cardinales con determinado tiempo.

El robot constara de tres motores que le darán propulsión, un par de aletas unificadas para darle dirección y otro par de aletas para inmersión, tiene la capacidad de transportar sensores para recabar información submarina, una tarjeta Arduino mega encargada de cumplir los algoritmos para la locomoción, un sensor de 10 grados de libertad que incluye un giroscopio, un acelerómetro, un magnetómetro y un sensor de presión, para la comunicación inalámbrica se usará

un XBee con alcance de hasta 100 metros de distancia, será alimentado con 2 baterías tipo lipo de 11.1 v de 5000 mAh, para la inmersión y los giros las aletas serán movidas por dos servomotores que están en una capsula hermética cada uno, los servomotores giraran en 3 posiciones para lograr rotar el robot o incluso sumergirlo, los valores en grados son de 55, 90 y 125 en el caso del servomotor de dirección con 55° girará hacia la izquierda, 90° estará centrado y 125° girará hacia la derecha . La Figura 2 demuestra el diagrama de bloques propuesto para el submarino y el control

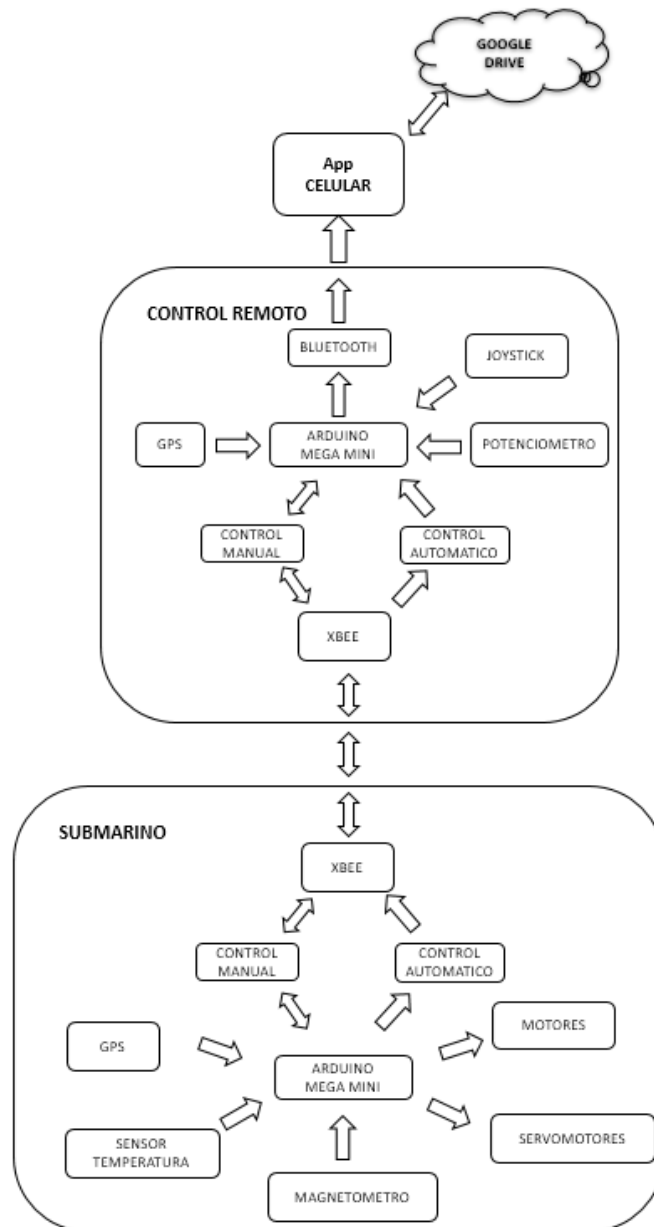


Figura 2 Diagrama de bloques propuesto para el Robot Submarino

Fuente: Elaborado por el Autor

## **1.3 OBJETIVOS**

### **1.3.1 OBJETIVO GENERAL**

Implementar un robot submarino tipo torpedo autónomo utilizando tecnología open source para recopilar datos oceanográficos de temperatura.

### **1.3.2 OBJETIVOS ESPECÍFICOS**

- Estudiar sistemas de navegación y equipos electrónicos utilizados en robots autónomos para la implementación en el robot submarino.
- Diseñar el sistema electrónico y el algoritmo de control para la autonomía de la trayectoria del robot submarino.
- Desarrollar una aplicación móvil para la presentación de los datos oceanográfico de temperatura
- Diseñar la estructura mecánico electrónico del robot submarino para cumplir trayectorias programadas.

## **1.4 RESULTADOS ESPERADOS**

La propuesta consiste en construir un prototipo de robot submarino tipo torpedo mediante tecnología open source, capaz de ser controlado en la superficie mediante comunicación inalámbrica con un control remoto y este a su vez conectado con una aplicación Android instalada en el teléfono celular, una vez que se ordene el descenso el robot deberá realizar un recorrido submarino mediante una trayectoria propuesta en el código realizando un recorrido deducido.

El robot se desplazará demostrando un eficiente desplazamiento por su diseño hidrodinámico tanto en la superficie como en estado sumergido, la trayectoria consistirá en el descenso de manera espiral por un tiempo menor a 10 segundos y luego hará un recorrido formando un cuadrado, mediante este tiempo se colecta el

dato de temperatura y se lo almacena en una dirección en la memoria eeprom del Arduino, una vez que el robot emerge enviará por trama de datos los valores de temperatura al control, este al celular y por ende será almacenado en tabla dinámica en Google drive.

Para garantizar la integridad de los equipos electrónicos se diseñarán y modificarán cápsulas herméticas con materiales de bajo costo e incluso complementar con la elaboración de piezas diseñadas e impresas en 3d.

## **1.5 JUSTIFICACIÓN**

Este proyecto es un subcomponente del proyecto “Desarrollo de Robots Submarinos para exploraciones en la zona marino-costera de la Península de Santa Elena y en la Antártida” aprobado en el 2017 con código 00382451 del INCYT (Instituto de Investigación Científica y Desarrollo de Tecnologías) de la Universidad Estatal Península de Santa Elena.

La Universidad Estatal Península de Santa Elena se ha destacado a nivel nacional en proyectos enfocados en aplicaciones de robótica submarina que son un sistema importante en la recolección de información oceanográfica, esto con el fin de profundizar estudios de factores que afecten la flora y fauna marina, es posible la elaboración de proyectos de robots submarinos para cubrir la necesidad de explorar los océanos y recolectar datos.

En el estudio sobre las aguas marinas territoriales, se necesita de muestra y análisis del agua de mar realizada por los biólogos marinos que utilizan estas muestras para generar información acertada de los cambios de comportamiento y características del mar, para lograr esto se requiere de un equipo autónomo que se desplace sobre y debajo del mar, siendo la mejor opción un robot submarino, como herramienta para cumplir este propósito, de tal manera que en la elaboración y ejecución de este proyecto se aplica conocimientos en las áreas de electrónica, informática y robótica que es requerida de forma imprescindible en este trabajo.

## **1.6 ALCANCE DEL PROYECTO**

Este proyecto deberá de concluir con un robot submarino terminado capaz de sumergirse hasta un máximo de 50 m de profundidad en aguas costeras y tendrá que tener autonomía, sumergirse y emerger después de cumplir la misión de recorrido previamente programada, teniendo un control adecuado de sus motores y alerones, las pruebas en mar abierto se las realizaran en playas del cantón Salinas (San Lorenzo) y en la comuna de Ayangué, el robot será un instrumento para la recolección de datos para su posterior procesamiento y análisis por parte de docentes y estudiantes de la carrera de Electrónica y Telecomunicaciones y Biología Marina de la Universidad Estatal Península de Santa Elena.

Al finalizar este proyecto se busca un impacto positivo en la población peninsular mediante el uso de electrónica aplicada dirigida a la recolección de los datos oceanográficos para su posterior estudio, esto implica que muchas personas de la provincia se ven involucradas directa o indirectamente y deja una puerta abierta a nuevos proyectos de tecnologías aplicados al estudio del mar.

## **1.7 METODOLOGÍA**

En este proyecto se aplica el método experimental mediante técnicas de observación y análisis del comportamiento en base a la programación que se realice para completar lo que es el recorrido deducido en base a los datos obtenidos por parte de los sensores, también se experimentará en la elaboración de recipientes y de más elementos que ayudaran a la sujeción de piezas en el robot y la hermetización de los envases que tendrán la parte electrónica en el submarino.



## CAPITULO II LA PROPUESTA

### 2.1 MARCO CONTEXTUAL

En la facultad de Electrónica y Telecomunicaciones se desarrolló un robot submarino hecho con partes diseñadas y otras que se adquirieron de manera local, la finalidad del robot es desplegarlo en aguas costeras para la recolección de datos oceanográficos, dichos datos pueden ser temperatura, salinidad, conductividad, muestras de fitoplancton, en nuestro caso usaremos solo un sensor de temperatura, los datos serán procesados por estudiantes de biología marina para su respectiva interpretación y así conocer los cambios que se van dando en el ambiente marítimo peninsular (Figura 3).



Figura 3 Robot Submarino UPSE

Fuente: Elaborado por el Autor

El lugar de operatividad del robot submarino podrá ser todo el perfil costanero pero las pruebas experimentales de este proyecto se las realizaron en la comuna Ayangué debido a la seguridad que presta sus aguas tranquilas para el mejor despliegue del equipo, al ser un lugar turístico existen embarcaciones que prestan los servicios de zarpe para actividades marítimas, gracias a este proyecto se podrá analizar en primera instancia los valores de temperaturas.

Ayangue se encuentra ubicada en la latitud  $-1.980687$  y longitud  $-80.748175$  con una distancia de 44,8 km y 53 minutos de viaje por vía terrestre desde las instalaciones de la Universidad Estatal Península de Santa Elena, la figura 4 tomada de la página de Google Maps nos indica dicho dato y la ruta a tomar para llegar a la comuna Ayangue.

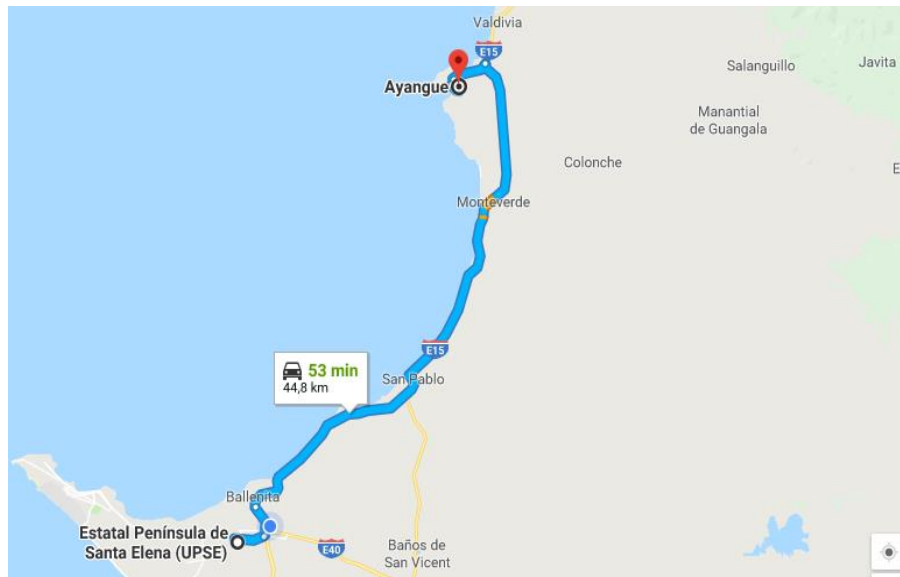


Figura 4 Ubicación Ayangue en Google Maps

Fuente: Google Maps

Otra localidad visitada por el robot submarino fueron las playas del Cantón Salinas ya que tiene un movimiento moderado del oleaje el robot submarino se puede desplegar sin ningún inconveniente y así comprobar su flotabilidad y motricidad que están siendo de lo más efectiva, Salinas está ubicado geográficamente en la latitud  $-2.196560$  y longitud  $-80.986869$  con una distancia de 13,8 Km y 30 minutos de viaje dependiendo del tráfico desde la Universidad Estatal Península de Santa Elena.

En la figura 5 tomada de Google Maps podemos ver la ruta trazada para el recorrido terrestre.

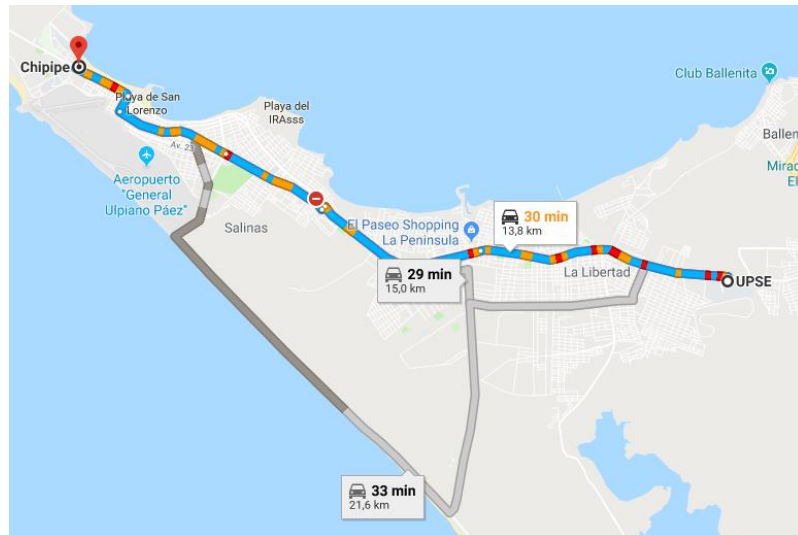


Figura 5 Ubicación de Salinas playa de Chipipe en Google Maps

Fuente: Google Maps

## 2.2 MARCO CONCEPTUAL

A continuación, detallaremos los conceptos e ideas pertinentes a los componentes del robot submarino.

### 2.2.1 ROBOT

Como concepto de Robot podemos encontrar varias definiciones dependiendo de la simplicidad, por ejemplo:

Un robot es una máquina que luce y actúa como ser humano.

Un robot es un equipo que realiza tareas repetitivas; o incluso

Un robot es algo guiado por mandos automáticos.

¿Pero en sí que es un robot? en una definición que le podemos dar a esta terminología un Robot es un equipo mecánico-eléctrico programable capaz de realizar tareas repetitivas dependiendo de la programación que el usuario le da para cumplir determinados propósitos, estos equipos pueden clasificarse de diferentes formas tales como su apariencia, funcionalidad, propósito, etc., pero es considerado

un robot como Agente Autónomo Inteligente (AAI) cuando llega a cumplir los requisitos de Autonomía e Inteligencia. (García, 1999).

**Autonomía** se hace referencia al sistema de navegación que posee el robot sin necesidad de estar conectado a algún equipo externo por ningún medio sea físico o inalámbrico.

**Inteligencia** el robot va a ser capaz de poder tomar sus propias decisiones basadas en un cuadro de probabilidades y alimentadas por las lecturas de los diferentes sensores.

### **2.2.2 SUBMARINO**

El submarino es un equipo de gran calado pensado para realizar sus operaciones bajo el agua a diferentes profundidades dependiendo del diseño y fin que tenga, ya que pueden ser varios, desde fines con actividad militar, transporte o incluso para recopilación e investigación de vida submarina. Hay que diferenciar dos equipos que tienen semejanza como es el submarino y el sumergible; el sumergible solo podrá estar bajo el agua por determinados momentos no así el submarino que puede pasar tiempos prolongados.

La manera que opera un submarino para lograr sumergirse es el tener compartimientos de aire que una vez dada la orden de inmersión estos se llenaran de agua para conseguir que el submarino se hunda, pero necesitaran sin duda del movimiento de sus timones y la fuerza de la o las propelas que este posea para alcanzar las velocidades necesarias para conseguir que se sumerja.

### **2.2.3 ROBOT SUBMARINO**

Un robot submarino es un equipo no tripulado de dimensiones relativamente pequeñas en comparación con los submarinos de gran calado, está diseñado para realizar trabajos bajo el agua mientras se encuentra navegando, un robot submarino

puede ser utilizado para dos tipos de misiones sean de manipulación o de inspección.

**Misiones de manipulación**, consiste en equipos que se desplazan hasta ciertas profundidades y están dotados de brazos o herramientas que recopilan muestras, comúnmente este tipo de robots son más manejables en espacios pequeños y pueden rotar en su propio eje, llevan un número mayor de motores que le dan la estabilidad y la maniobrabilidad (Moreno, 2013).

**Misiones de Inspección** en este tipo de trabajos los robots sirven para la recopilación de datos oceanográficos e imágenes mientras se están desplazando, no necesitan tener un mayor grado de maniobrabilidad ya que para cubrir el entorno donde se desplaza se le pueden ubicar varias cámaras (Moreno, 2013).

Los robots submarinos no tripulados pueden ser teleoperados o autónomos.

**Robots Submarinos Teleoperados** o ROV por sus siglas en inglés de Remote Operated Vehicle son equipos que necesariamente llevan una conexión física con un mando para su manejo, esa conexión física hace referencia a un cable de gran longitud que une el submarino al control que comúnmente es por computador para el proceso de las imágenes (Zilic, 2015).

Entre los equipos ROV que existen en el mercado como equipos comerciales podemos encontrar el siguiente:

### **ROV HERCULES**

El robot hércules posee seis propulsores para un mayor control, parecido a un helicóptero en el aire teniendo la capacidad para rotar en su propio eje, posee sensores de presión, temperatura, medidor de concentración de oxígeno, medidor de salinidad y cámaras, tiene implementado un par de brazos articulados para la manipulación y recolección de objetos están siendo utilizados para temas de carácter arqueológico, biológico y geológico. (Live, 2019)

## Especificaciones

**Peso en el aire** 5400 lb (2450 Kg)

**Peso en Agua de Mar** neutralmente flotante

**Dimensiones** 3,4m 1.8m x 2,3m

**Potencia** 2400 VCA, 3 fases, < 20kW

**Clasificación de profundidad** 4000m (12,124 pies)

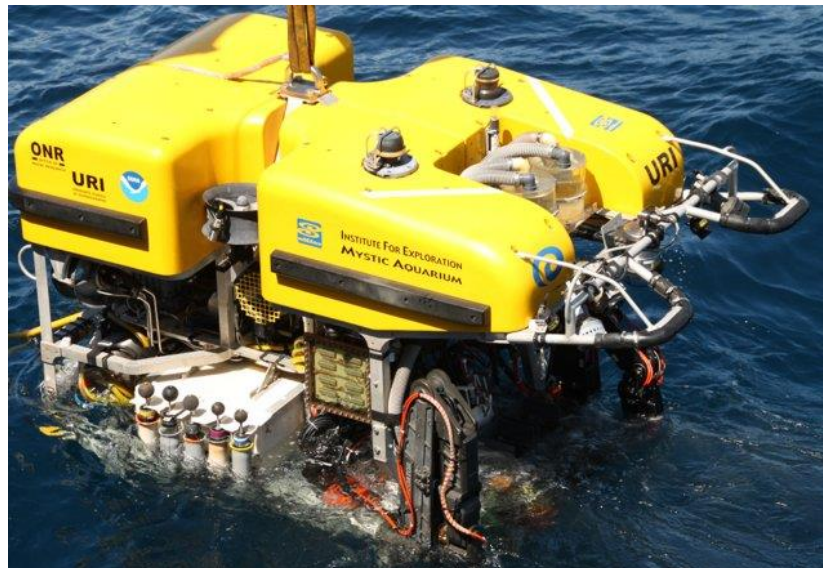


Figura 6 ROV Hércules

Fuente: Nautilus Live

**Robots Submarinos Autónomos** o AUV por sus siglas en Inglés de Autonomus Underwater Vehicule son equipos autopropulsados que no necesitan de ninguna conexión física, su desplazamiento será cumpliendo recorridos previamente programados o incluso usando módulos de inteligencia artificial para que puedan tomar sus propias decisiones al cumplir la tarea encomendada. (Zilic, 2015) entre los equipos de este tipo podemos referirnos al siguiente:

### Hugin

Es una plataforma hidrodinámica muy estable y de bajo nivel de ruido para los sensores de carga útil, tiene una maniobrabilidad de giro de 15 metros, puede llegar a profundidades máximas de 6000 metros posee un sistema de navegación inercial

asistido de última generación, entre las últimas implementaciones de Hugin es una batería que tendrá una duración de 100 horas a una velocidad de 4 nudos. (Maritime, 2019)

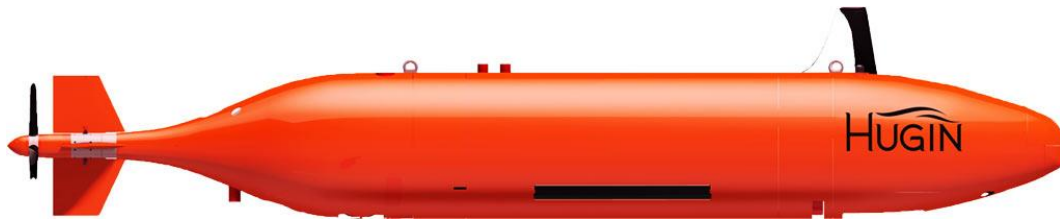


Figura 7 AUV HUGIN

Fuente: Kongsberg

### 2.2.5 ACTUADORES

Los actuadores son las partes que le dará la movilidad al robot submarino

#### **Propulsor**

Los propulsores en el robot son motores sin escobillas con un recubrimiento especial que impedirá que la salinidad llegue a dañar de manera inmediata las partes metálicas.

#### **Aletas**

El robot posee dos pares de aletas uno que es para sumergir y emerger y el otro que le dará la dirección, dichas aletas también llamadas timones deben de tener buena hidrodinámica para poder conducir el agua a su alrededor y no exista arrastre hidrodinámico.

### 2.2.6 DATOS OCEANOGRÁFICOS

Son los valores recopilados mediante sensores que indican los diferentes cambios que hay tanto en la superficie como en las diferentes profundidades del océano, entre los valores que se pueden recopilar con un robot submarino son:

Conductividad, Temperatura, Oxígeno disuelto, Presión, Turbidez, Muestras de fitoplancton (Zambrano, 2009).

### **2.2.7 COMUNICACIÓN INALÁMBRICA**

La comunicación inalámbrica o sin cables es en la que los extremos no están unidos por un medio físico sino por las señales electromagnéticas que se propagan por el espacio, esta comunicación se da en base a protocolos, los que usaremos para la comunicación entre el robot y el control, esta basados en componentes físicos denominados XBee que utilizan el protocolo IEEE 802.15.4 y para la comunicación inalámbrica de 2.4 GHz entre el control y la aplicación móvil se utiliza el protocolo RS232.

### **2.2.8 HERMETICIDAD**

La hermeticidad es una cualidad que tienen los envases de no dejar pasar ningún fluido a su interior incluyendo sustancias como aire o agua, los recipientes herméticos deben de tener la suficiente resistencia a las diferentes presiones a las que están inmersas en varios niveles del mar y evitar de esta manera que imploten por la poca resistencia de sus paredes.

### **2.2.9 HIDRODINÁMICA**

La hidrodinámica estudia el comportamiento de los fluidos en un cuerpo verificando cambios de velocidad, presión, flujo, en los submarinos es muy importante la hidrodinámica de su exterior ya que esto derivará en la velocidad resultante, maniobrabilidad que tenga en el entorno acuífero.

Al tener buena hidrodinámica se aprovecharán al máximo la velocidad de los motores, los timones también tienen una parte importante para el desplazamiento ya que ellos son los que dirigirán al robot en el entorno.



## 2.2.10 DISEÑO 3D

El diseño 3D consta en proporcionar a un objeto hecho en un programa o software especial las componentes en las tres dimensiones, tanto que podemos crear o modificar cada una de sus caras rotando el elemento, entre los softwares más usados podemos encontrar al programa FreeCAD (Figura 8), Sketchup entre otros.

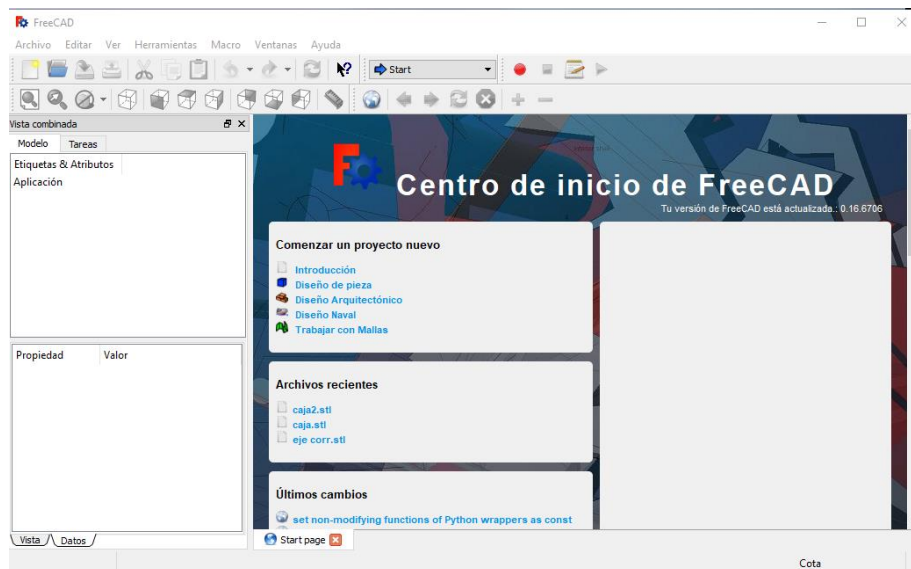


Figura 8 Programa FreeCAD

Fuente: El Autor

## 2.2.11 IMPRESIÓN 3D

Entre los avances de la humanidad en el siglo XXI podemos encontrar la aparición de la impresión 3D, que consiste en colocar capa a capa filamento extruido dando forma a algo previamente diseñado. Existen varios modelos entre ellos los siguientes:

### **Impresora Anet A6**

Impresora 3D de código abierto imprime varios tipos de filamentos entre ellos PLA, ABS, FLEXIBLE entre otros, al imprimir es silenciosa, el volumen de impresión está por encima del estándar ya que su cama es de 220mm x 220mm x 250 mm y es fácil de nivelar (IMPRESORAS3D.com, 2019).

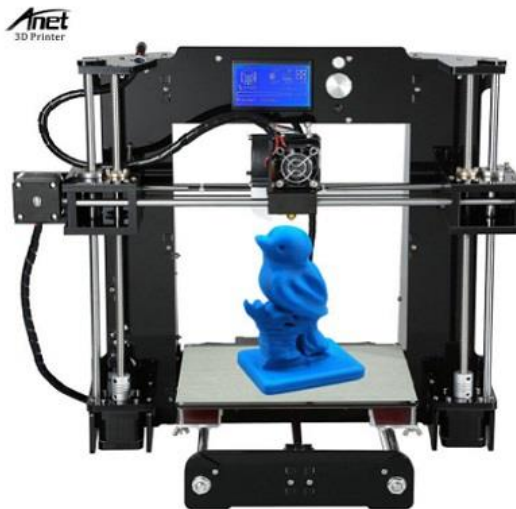


Figura 9 Impresora Creality Ender 3

Fuente: Impresoras3d.com

### **2.3 MARCO TEÓRICO**

En 2012 en la ciudad de México se llevó a cabo el “DISEÑO Y MANUFACTURA DE UN PROTOTIPO DE ROBOT SUBMARINO AUTÓNOMO CON CONTROL DE ESTABILIDAD DE INMERSIÓN” el proyecto consistía en conseguir que el equipo submarino logre la estabilidad en descensos a diferentes profundidades en un ambiente sin perturbaciones gracias al uso de tanque de oxígeno colocados en la parte inferior del robot, el control de estabilidad sería autónomo, no estaba considerado el desplazamiento de manera horizontal (Salamanca, 2012).

En 2004 en la ciudad de Bogotá Colombia se realizó la tesis “CONTROL DE INMERSION PARA PLATAFORMA ROBÓTICA SUBMARINA” la cual consistía en el diseño e implementación de una plataforma submarina capaz de recibir orden externa para la inmersión a cierta profundidad (Cuellar, 2004).

En 2014 en la Universidad San Francisco de Quito se presentó la propuesta “DISEÑO Y CONSTRUCCIÓN DEL SUBMARINO PARA EXPLORACIÓN ORCA I” que se trata de un equipo operado remotamente y con la finalidad de ser

utilizado para la exploración subacuática para la recolección de datos de lagunas y lagos de la ciudad, es un robot omnidireccional para estudio mediante imágenes (Carrillo, 2014).

## **2.4 COMPONENTES DE LA PROPUESTA**

A continuación, se detalla las características de cada uno de los componentes físicos y lógicos utilizados para el desarrollo del proyecto, mencionando las características individuales de cada elemento y así dar una noción más clara respecto a la selección de dicho material para el perfecto funcionamiento del robot submarino.

### **2.4.1 COMPONENTES FÍSICOS**

#### **TARJETA ARDUINO MEGA 2560**

Esta tarjeta posee un microcontrolador de gran capacidad como lo es el Atmega 2560, la tarjeta está compuesta de 54 pines digitales que funcionan tanto como pines de entrada como de salida, 16 entradas analógicas y un cristal oscilador de 16MHz, posee un puerto USB y un puerto de alimentación, 14 de sus pines digitales también pueden ser utilizados para la modulación de ancho de pulso (PWM), tiene 4 puertos seriales (UART) (Arduino, 2018)

#### **Especificaciones técnicas**

Microcontrolador	ATmega 2560
Tensión de funcionamiento	7-12V
Pines digitales de Entrada Salida	54 de los cuales 15 proporcionan salida PWM
Pines de entrada Analógica	16
Corriente DC por pin de Entrada Salida	20mA
Corriente DC para pin 3.3V	50mA
Memoria Flash	256 KB
SRAM	8 KB

EEPROM	4KB
Velocidad de reloj	16 MHz
Dimensión	53,3mm x 101.52mm

Una adaptación existente que facilita el uso de la tarjeta Arduino Mega pero con un tamaño más reducido es la hecha por RobotDyn que ha conseguido poner en una placa de solo 38mm x 55mm toda la funcionalidad de esta gran placa.

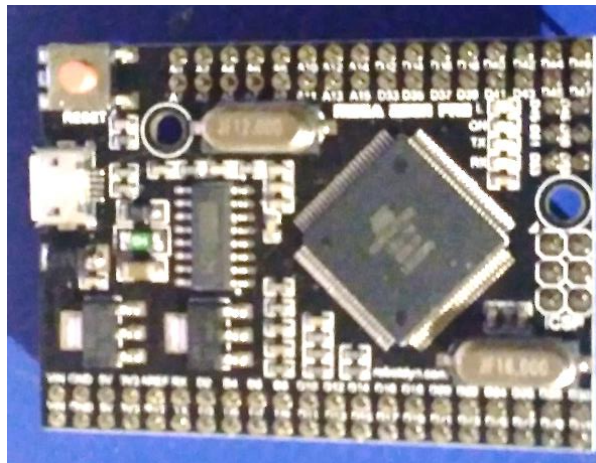


Figura 10 Arduino Mega 2560 Mini

Fuente: El Autor

### **GPS Ublox NEO 6M**

La serie de los módulos NEO-6M son receptores GPS autónomos que poseen alto rendimiento con el módulo u-blox 6, el motor de posicionamiento ublox posee un motor de posicionamiento con tecnología TTFF que no es más que un tiempo de primera corrección que es inferior a 1 minuto, su tamaño compacto es pensado para el uso de diversas aplicaciones inclusive da la facilidad de conectarlos con diferentes fuentes de poder, tiene un motor de adquisición dedicado, posee gran capacidad para búsquedas masivas de satélites y encontrarlos casi al instante de ser encendido siempre y cuando se encuentre en un área abierta. (Ublox, 2019)

### **Especificaciones Técnicas**

- Voltaje de alimentación: 3v-5v

- Memoria EEPROM para guardar datos de configuración, se comunica por comandos AT
- Tamaño de la tarjeta 23mm x 30mm
- Tamaño de la antena 22mm x 22mm
- Tipo de receptor de 50 canales, GPS L1 (1575MHz) C/A code, SBAS: WAAS/EGNOS/MSAS
- Precisión en posición horizontal 2.5 mCEP
- La Tasa actualización de Navegación es 5Hz máximo (1Hz por defecto)
- Sensibilidad de seguimiento y navegación -161dBm
- Protocolo de comunicación NMEA (Waveshare, 2019)

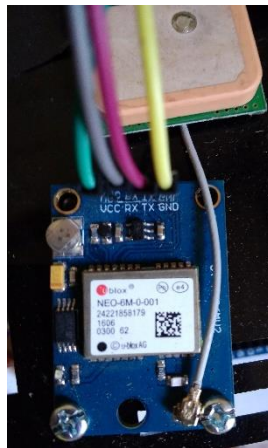


Figura 11 Modulo GPS NEO 6M

Fuente: El Autor

## MÓDULO XBEE PRO S1

Creado de Zigbee Alliance, el Xbee es un dispositivo de bajo consumo de energía basado en el protocolo de comunicación IEEE\_802.15.4, se comunica inalámbricamente con otros dispositivos de su misma categoría realizando una verificación previa de la serie haciendo así más confiable la comunicación sea unidireccional o bidireccional entre los dispositivos, la configuración de los equipos se los realiza en un software llamado XCTU, mediante una configuración básica podremos emparejar nuestros dispositivos.

Son usados comúnmente en la industria para el envío de datos de equipos distantes a la computadora central y así disminuir recursos, además de su tamaño compacto y bajo consumo este equipo al trabajar en bandas que no necesitan licencia puede llegar a abarcar distancias de más de 2 Km dependiendo del dispositivo adquirido (Oyarce, 2008)



Figura 12 Módulo Xbee Pro S1

Fuente: El Autos

## MÓDULO BLUETOOTH HC05

El módulo bluetooth es un dispositivo que reemplaza las conexiones eléctricas y la comunicación se hace inalámbrica en una frecuencia de 2.4 GHz, es un equipo capaz de conectarse a diferentes dispositivos, posee una memoria Eeprom y la configuración se la puede realizar por medio de los comandos AT, dependiendo de la configuración este dispositivo puede ser utilizado solo como un emisor - receptor de señales, trabaja de modo esclavo - servidor e incluso puede servir para poder cargar de manera inalámbrica el firmware a nuestra tarjeta Arduino, solo habrá que hacer cambios en su firmware por medio de los comandos para habilitar el pin

STATE que se conectara al pin reset de la tarjeta Arduino, la alimentación puede ser de 3.3v – 5 v con una sensibilidad en la recepción de -85 dBm con un alcance aproximado de 15 metros, el rango de operación según la temperatura presente esta entre -40 a 105 °C.



Figura 13 Módulo Bluetooth

Fuente: El Autor

### **SENSOR 10 GRADOS DE LIBERTAD GY 89**

Este sensor es un módulo compacto que contiene acelerómetro, giroscopio, brújula digital y sensor de presión barométrica, todos estos sensores están conexionados en paralelo y la manera de acceder a ellos es mediante una conexión I2C por eso solo consta de 4 pines SDA, SCL, Vin y GND, el sensor que utilizaremos que está dentro del chip es:

### **LSM303D**

Es una brújula digital de 3 ejes y acelerómetro de 3 ejes, las seis lecturas que da son totalmente independientes, la salida es de 16 bits, tiene interfaz serial SPI/I2C con una alimentación de 2,16 V a 3,6 V (STMicroelectronics, 2012).

La brújula digital se la utiliza en escala azimutal resultando valores que van desde los 0° a 360° indicando 0° y 360° Norte, 90° Este 180° Sur 270° Oeste.



Figura 14 Sensor 10DOF

Fuente: El Autor

### **MOTOR BRUSHLESS A2212/10T**

Motor sin escobillas potente para drones, consumo de 140 vatios con un pico de 180 vatios en lapsos cortos, soporta baterías de 2 a 3 celdas con una eficiencia máxima del 80% y consumo de corriente de 4 – 10 amperios.



Figura 15 Motor Brushless A2212/10T

Fuente: El Autor



## **MOTOR T200 THRUSTER**

El T200 es un propulsor diseñado específicamente para robótica marina incluso se pueden usar en el transporte de seres humanos por su gran potencia se los coloca en kayak, remo a pie, o incluso en pequeños botes de pesca, es un motor sin escobillas con sus partes protegidas para disminuir la corrosión. El diseño es propiedad de la compañía Blue robotics dedicados a equipamiento en general para robots submarinos (Robotics, 2019).



Figura 16 Motor T200 Truster

Fuente: El Autor

## **CONTROLADOR ESC 30A**

Controlador de motores sin escobillas programable, convierte voltaje monofásico en trifásico con una corriente máxima de 30 A posee un regulador interno de 5v para alimentar la parte lógica y dar una salida de alimentación para otros dispositivos, posee entrada de alimentación directa de la batería y 2 salidas una trifásica a los motores capaz de entregar en voltaje de entrada y otra con salida de 5v, adicional la entrada de señal de modulación de ancho de pulso.



Figura 17 Controlador ESC 30A

Fuente: El Autor

## SERVOMOTOR DS3218

Servomotor digital con engranaje metálico con giro de  $0^\circ$  a  $180^\circ$  con ancho de pulso q va desde los 1000 a 2000 microsegundos tiende un porcentaje de impermeabilidad alto, funciona con un voltaje que va desde 4.8v a 6.8v con un torque de 21.5 Kg



Figura 18 Servomotor DS3218

Fuente: El Autor

## 2.4.2 COMPONENTE LÓGICOS

### Arduino IDE

Arduino Integrated Development Environment o en español Entorno de desarrollo integrado de Arduino es una plataforma desarrollada en lenguaje C++, es un entorno

amigable y de fácil comprensión donde podremos realizar las diferentes rutinas para la ejecución de los proyectos, los archivos realizados en esta plataforma se guardan con la extensión de archivo .ino, Arduino es una plataforma programable de código abierto, posee una lista grande de posibles placas a las que se les puede cargar códigos, trabajan con placas que poseen incorporado microcontroladores ATMEGA en varias presentaciones como la 2560, 128, 8, 168 de la marca Atmel. La plataforma contiene un puerto serial en donde podremos visualizar lo que está pasando en tiempo real con el proyecto y en versiones actualizadas incluso un plotter para visualizar un eje de coordenadas (Banzi, 2014).

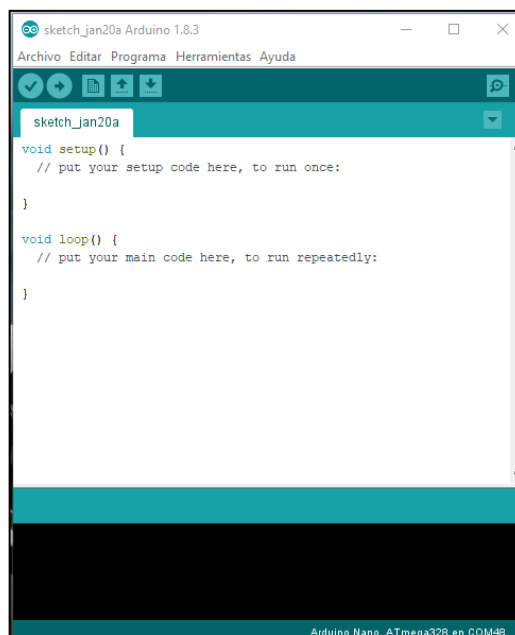


Figura 19 Arduino IDE

Fuente: El Autor

## MIT APP INVENTOR 2

El instituto tecnológico de Massachusetts fue el creador de una primera versión de la herramienta de programación adoptada por Google quien la mejoró para que sus usuarios puedan crear aplicaciones para dispositivos móviles de una manera más sencilla (FPP, 2015), consta de 3 partes:

## Diseño

En este apartado se muestra un display de un móvil donde se colocarán las diferentes componentes como botones, imágenes, audios, textos, etc. Se configura también su aspecto dependiendo del uso (FPP, 2015).

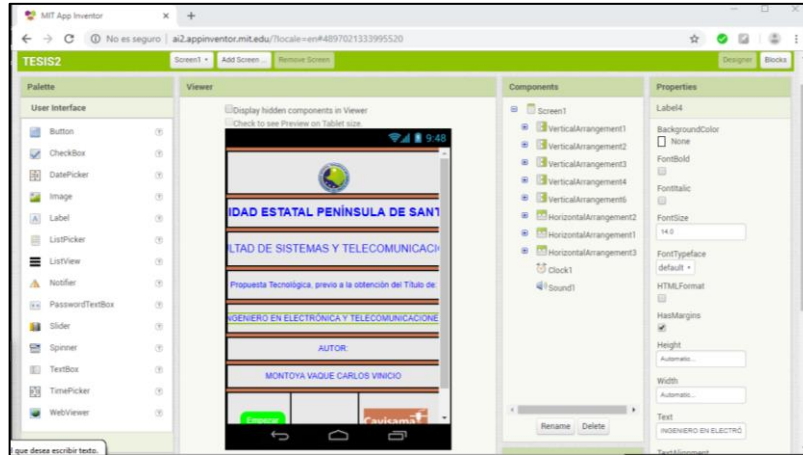


Figura 20 Página de diseño App Inventor 2

Fuente: El Autor

## Editor de bloques

La programación de las aplicaciones se da en este apartado, la programación es de una manera gráfica y visualmente atractiva para ayudar al diseñador a resolver las secuencias de una manera más versátil (FPP, 2015).



Figura 21 Pagina Programación Por Bloques

Fuente: El Autor

## Generador de aplicaciones

Una vez que esté lista la aplicación se puede generar el instalador APK y puede ser obtenido por un código QR para su descarga desde un móvil o incluso descargar la APK como tal al ordenador, otra función de este programa es el uso de emuladores en los cuales se podrá apreciar los cambios que se dan en la aplicación una vez terminada (FPP, 2015).

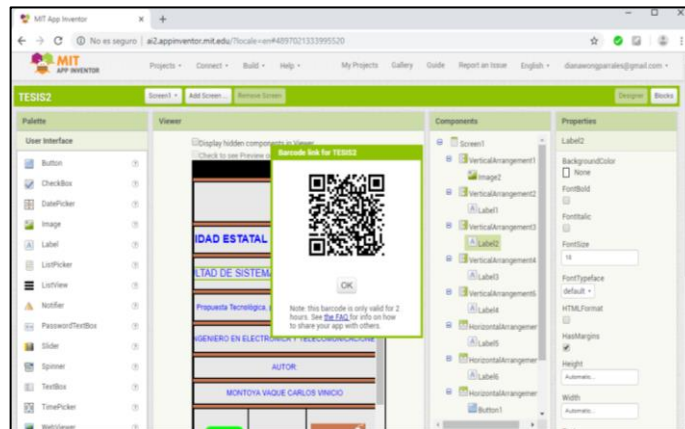


Figura 22 Código QR de Aplicación

Fuente: El Autor

## XCTU

XBee Configuration and Test Utility (XCTU) es un software multiplataforma que permite configurar los módulos XBee mediante la interacción de un interfaz gráfico, existen herramientas que hacen muy sencillo el configurar este tipo de dispositivos.

Las acciones que podemos llevar a cabo en esta plataforma son de configuración, inicialización, actualización de firmware y testeo de los módulos XBee mediante la comunicación por puerto serial, por su funcionalidad se puede ver rápidamente un resumen de los parámetros del módulo con la descripción respectiva (jcrepom, 2018).

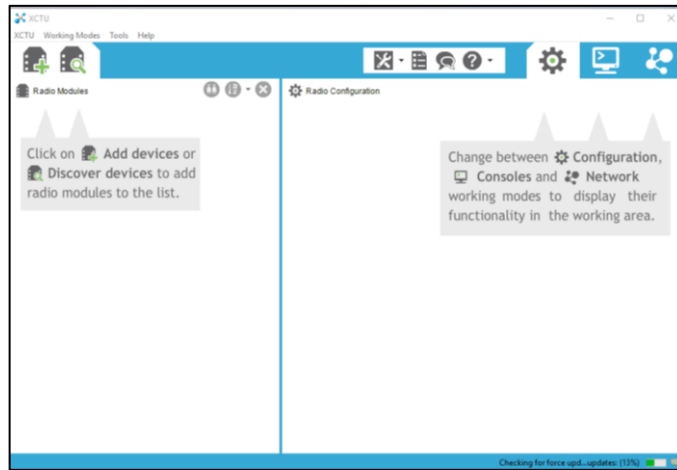


Figura 23 Programa XCTU

Fuente: El Autor

## PCB WIZARD

Es un programa diseñado para estudiantes y profesionales que trabajen con diseños de placas, contiene una lista de elementos completos que ayudan a realizar los circuitos que luego serán transferidos a las placas electrónicas (De La Rosa, 2018).

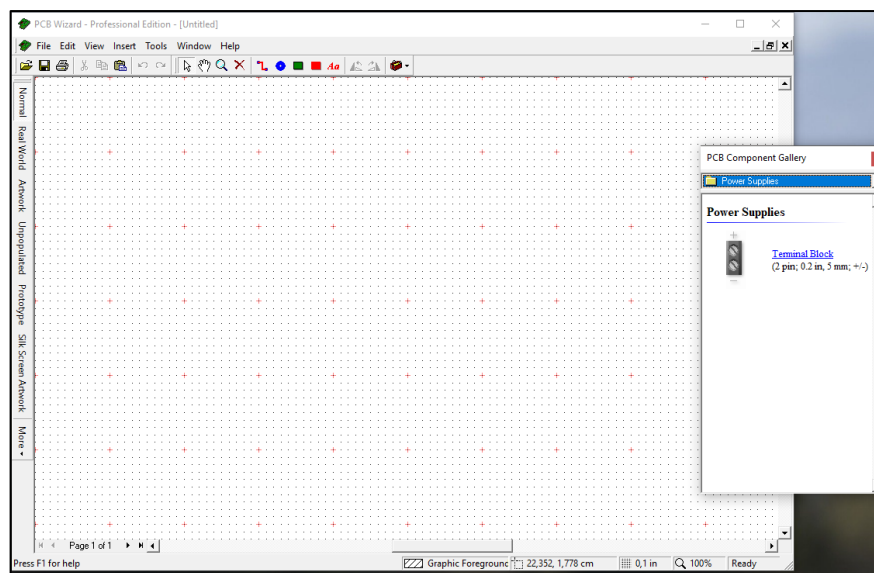


Figura 24 PCB Wizard

Fuente: El Autor

## **FreeCAD**

Es un programa open source (código abierto) fácil de utilizar y sirve para el diseño de elementos en 3D, el uso de este programa es extenso, de fácil aprendizaje y gran utilidad, se lo puede utilizar en varias plataformas como Windows, Linux y Mac; y está disponible en varios idiomas (Toloka, 2019).

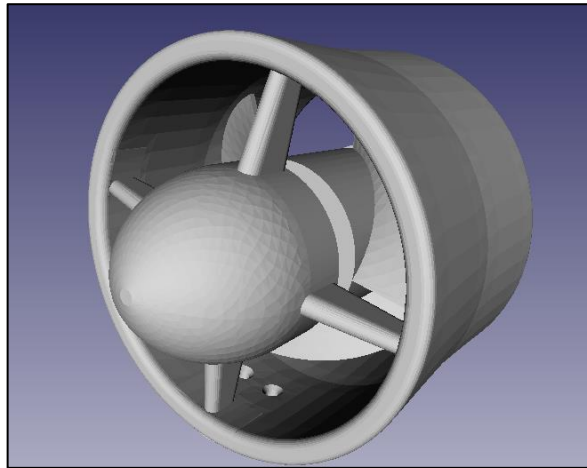


Figura 25 Diseño hecho en FreeCAD

Fuente: El Autor

## **ULTIMAKER CURA**

El software preferido para las impresoras Ultimaker sin duda es CURA, un programa donde se configuran las impresoras 3D y en base a esa configuración se convierten las imágenes 3D a Código G que es el archivo que las impresoras 3D utilizan para realizar los desplazamientos en los 3 ejes, este programa no solo sirve para las impresoras de marca ULTIMAKER que son de software propio sino también para las de construcción open source como la Anet, Tevo, Prusa, etc., eso depende de la configuración de las impresoras en dicho programa.

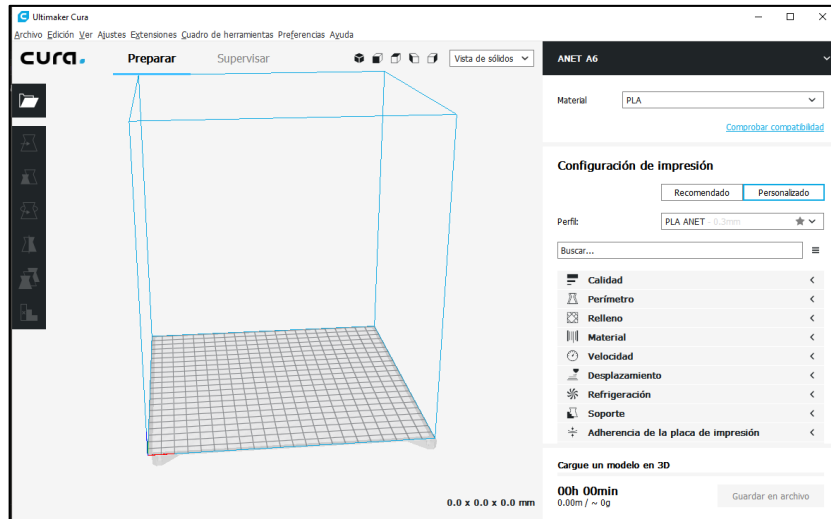


Figura 26 Ultimaker Cura

Fuente: El Autor

## 2.5 DISEÑO DE LA PROPUESTA ESQUEMA

A continuación, se muestra el esquema del robot submarino el cual lo dividiremos en 3 partes como es el hardware, firmware y diseño electrónico. El hardware es la parte externa del robot donde encontraremos la carcasa, los timones, propulsores, torre de comunicación, y envases herméticos que en su mayoría son de material plástico.

El firmware es el programa que comandará el movimiento que tenga el robot sea en estado manual o autónomo, en los procesos realizará la adquisición de los datos de los sensores y establece la comunicación con el control.

El diseño electrónico será la placa impresa donde está ubicado el conexionado de cada uno de los dispositivos electrónicos.



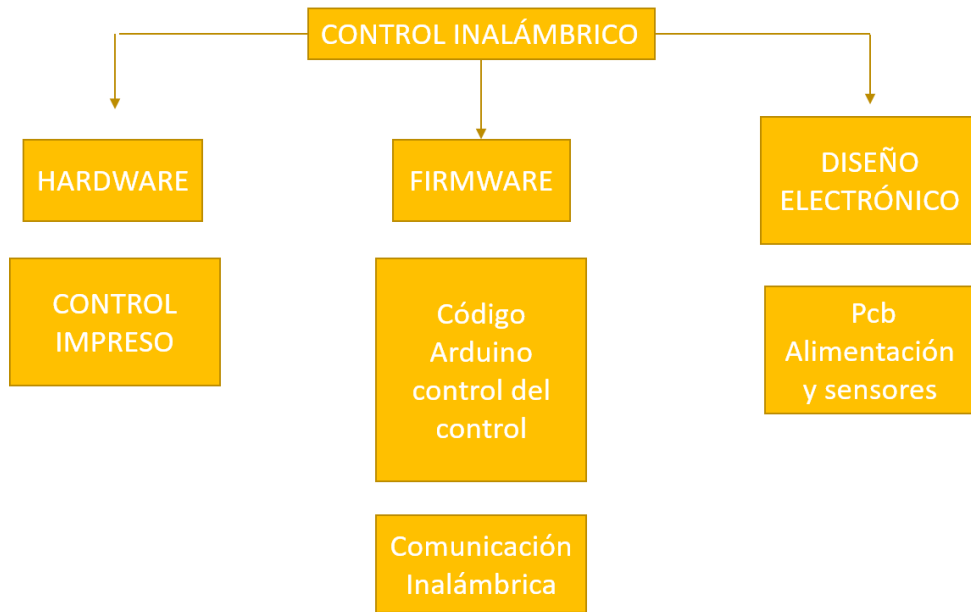


Figura 27 Estructura Control Inalámbrico

Fuente: El Autor

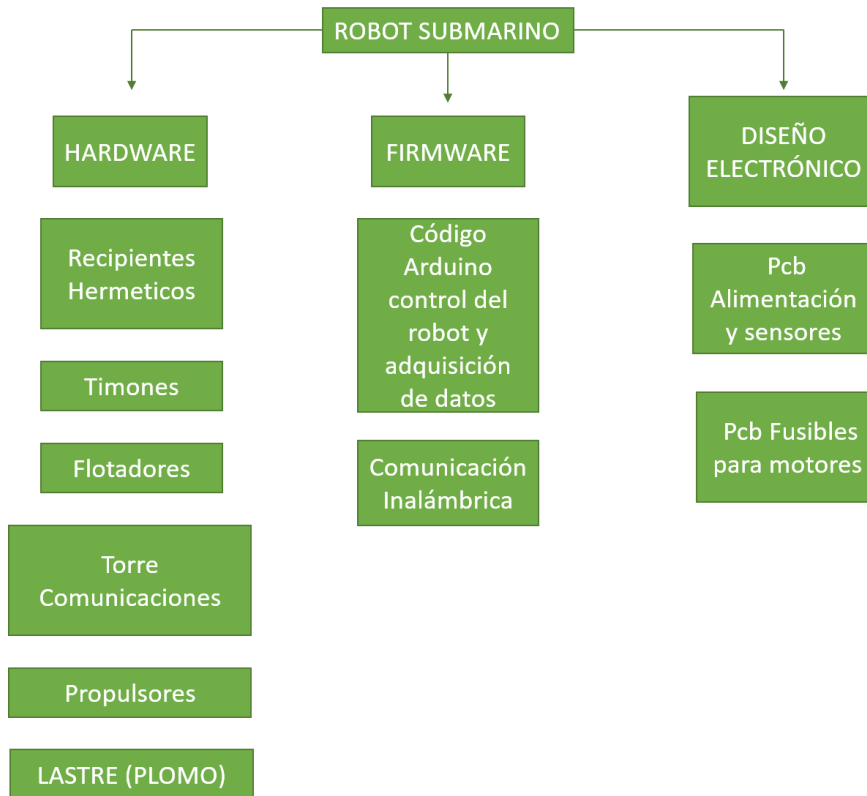


Figura 28 Estructura Robot Submarino

Fuente: El Autor

## 2.5.1 DISEÑO ELECTRÓNICO

### 2.5.1.1 DISEÑO ELECTRONICO DEL ROBOT SUBMARINO

El robot submarino está basado en la estructura tipo torpedo utilizado comúnmente para misiones de recolección de datos oceanográficos consta de dos compartimientos herméticos donde se distribuirán las cargas necesarias para los propulsores, servomotores y electrónica en general, la parte de los motores será alimentada por una batería de 14,1V de 4000 mAh, se utilizó inicialmente tres motores brushless “ A2212” para la propulsión utilizados en drones (Figura 15), pero por el esfuerzo que realizaban estos los ESC utilizados se estropearon, estas pruebas se las hicieron experimentalmente, finalmente en otra prueba se utilizan dos propulsores T200 Truster de la marca Blue Robotics garantizando el empuje necesario para que el prototipo pueda navegar debido a que son elementos fabricados para tal fin.



Figura 29 Robot Submarino UPSE

Fuente: El Autor

Para distribuir la energía de la batería hacia los propulsores y evitar que estos se estropeen por algún excedente de corriente se instalaron elementos limitadores como son los fusibles de uña con capacidad de 20 A (Figura 30).

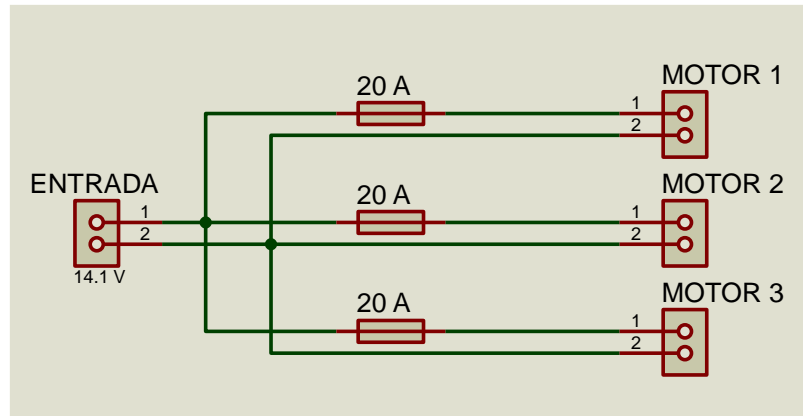


Figura 30 Diagrama eléctrico protección de los motores

Fuente: El Autor

La parte de alimentación de la tarjeta, los sensores y servomotores serán alimentados por una batería de 7,4v de 4000 mAh, la carga referente a los sensores es la siguiente:

1	Arduino Mega 2560	93,00 mA
1	GPS Ublox NEO 6M	67,00 mA
1	Xbee S1	500,00 mA
1	Sensor GY89	200,00 mA
2	Servomotor DS3218	3000,00 mA
1	Lm35	0,06 mA
	<b>Total</b>	<b>3860,06 mA</b>

Tabla 1 Tabla de consumo total de corriente del Robot Submarino

Fuente: Elaborado por el autor

El consumo total de corriente es de 3860,06 mA, para cubrir esta demanda usamos una batería de 7,4 V a 4000 mAh 30 C – 40 C el valor de 30C nos indica que la batería tendrá una entrega total de corriente nominal de 30 veces los 4000mA que serían 120 Amperios, con una entrega de 4000 mA por hora, la potencia de la batería es de 0,03 KWh [1].

$$P = V * I \quad P = 7,4 * 4Ah \quad P = 29,6 Wh \approx 0,03kWh \quad [1]$$

Para alimentar a la tarjeta controladora se realizó un circuito reductor de voltaje con el circuito integrado LM7805 el problema con este dispositivo es que solo puede entregar como máximo 800 mA, para solucionar la entrega de corriente en otros elementos se utilizó también el TIP41C que es un transistor de potencia usado en sistemas de amplificación y conmutación, la polaridad es tipo NPN, el voltaje colector emisor máximo de 100V, tiene la capacidad de disipar potencia hasta 65W con una corriente de entrega en su colector hasta de 6A (Figura 31).

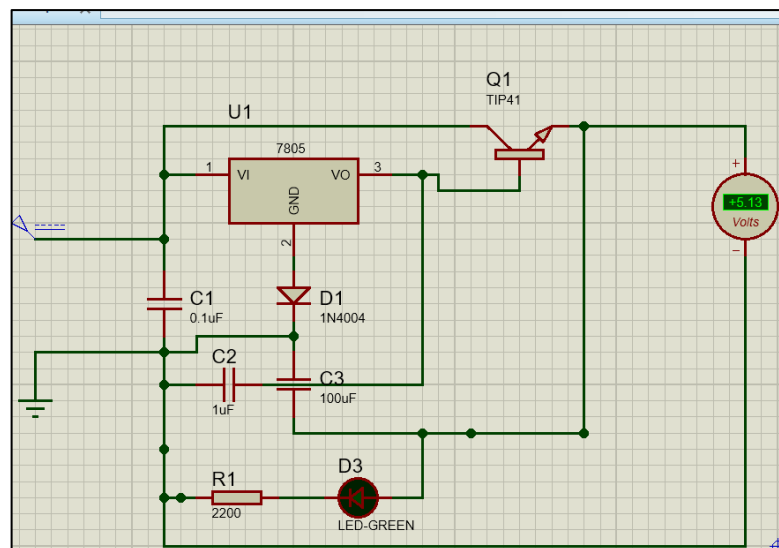


Figura 31 Circuito Reductor de voltaje 7,4v a 5v

Fuente: El Autor

En la Figura 32 se presenta el circuito reductor de voltaje que alimentará a los 2 servomotores, a la torre de comunicación que consta de un GPS Neo 6m y un XBee,

con otro circuito regulador LM7805 (Figura 45) alimentamos a la tarjeta Arduino en paralelo con el sensor GY89 y el sensor de temperatura LM35.

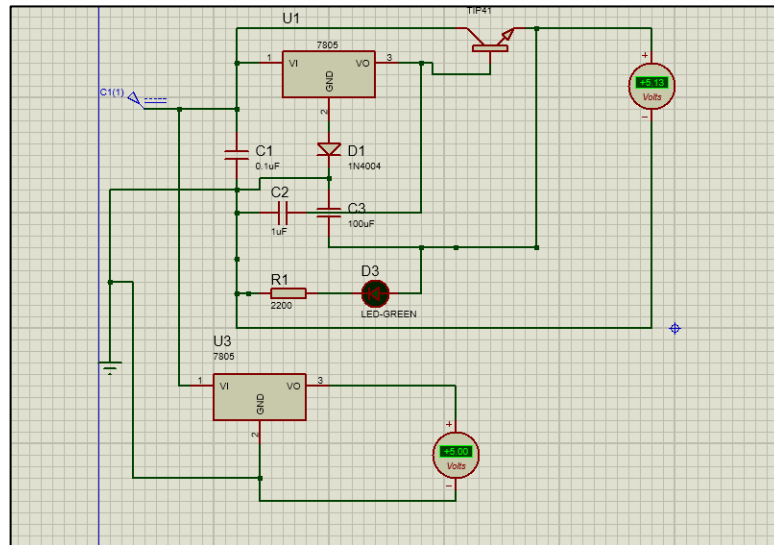


Figura 32 Circuito reductor de voltaje incluido alimentación a Arduino

Fuente: El Autor

La siguiente etapa en el diseño explica sobre las conexiones varias que van a la tarjeta Arduino para controlar y recibir información de todos los elementos electrónicos que componen el robot submarino, como varios de los actuadores no se encuentran en el mismo envase se colocó un conector DB15 hembra en la placa (figura 33) para realizar una sola conexión y evitar problemas de desconexiones, para el db15 macho (figura 33) se soldaron todos los cables de los actuadores y de los sensores.



Figura 33 DB15 Hembra y macho respectivamente

Fuente: Jaycar.com.au (Jaycar, 2018)

En el robot submarino se decidió hacer un circuito para conseguir cargar el firmware hacia el Arduino de manera inalámbrica ya que en muchos de los casos

los envases herméticos del robot estarían sellado y si se necesitara hacer una mejora en la programación, tendríamos que estar abriendo los envases, lo que no lo hace muy práctico, ya que consume tiempo y recursos, es por eso que se optó por esta solución, la cual consiste en un módulo bluetooth HC05 debidamente configurado para esta acción.

La configuración consiste en utilizar el pin 32 del módulo Bluetooth llamado comúnmente STATE, el proceso para cargar el firmware al Arduino consiste en ejecutarle un reset, para conseguir esto se utiliza un divisor capacitivo conectado como se muestra en la figura 37, en el punto de conexión se tienen las siguientes medidas de voltaje 3,4 a 3,6 V, cuando se carga el archivo y con ayuda de la descarga del capacitor el Arduino procederá a resetear y a recibir el nuevo código, para poder emplear este método hay que utilizar un software terminal como Putty para configurar el módulo bluetooth por comandos AT, las líneas que hay que sobrescribir son las siguientes:

AT	Debemos recibir el Ok para saber que si estamos conectados con el módulo
AT + ORGL	Para regresar datos de fabrica
AT+ROLE=0	Pasar a modo esclavo
AT+ POLAR =1	Se solicita que el PIN32 este en nivel bajo que es necesario para el reset
AT+UART=115200	Luego se configura el módulo para que trabaje a una velocidad de 115200

Tabla 2 Tabla de Comandos Configuración Bluetooth para la carga inalámbrica

Fuente: El Autor

Y con esos pasos a conseguimos cargar el firmware hacia el Arduino de manera inalámbrica.

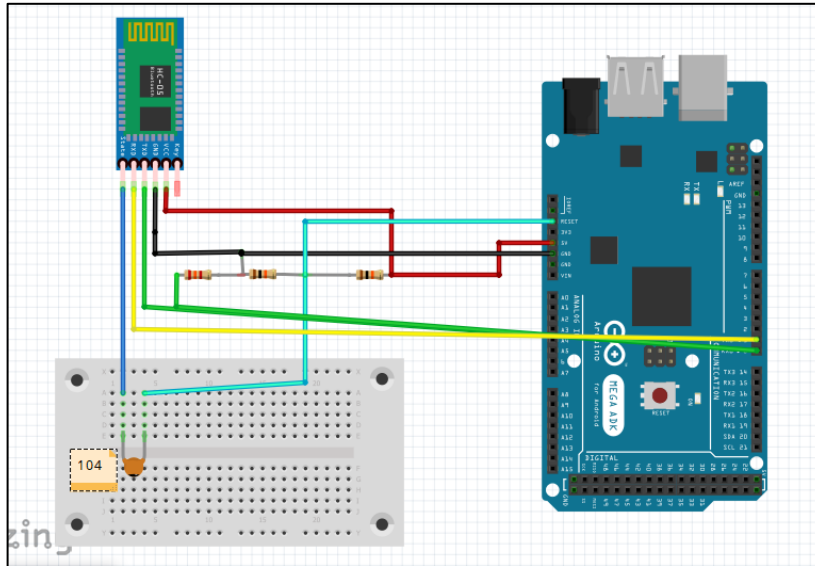


Figura 34 Conexión Bluetooth Para Carga de Sketch Inalámbricamente

Fuente: El Autor

### 2.5.1.2 DISEÑO ELECTRÓNICO DEL CONTROL INALÁMBRICO

Al igual que la alimentación del robot submarino se usó la misma configuración de reducción de voltaje (Figura 31), en el control inalámbrico existen varios dispositivos como es la tarjeta Arduino mega mini pro, bluetooth HC05, GPS Neo 6m, potenciómetro lineal, joystick, leds y pulsadores que están siendo alimentado con una batería de 7,4v 2200 mAh 30C – 40C con una potencia de 0,016 kWh [2] gracias al dispositivo electrónico inalámbrico XBee podremos comunicarnos con el robot submarino a una distancia hasta de 200 metros y con un dispositivo móvil hasta 15 metros.

$$P = V * I \quad P = 7,4 * 2,2Ah \quad P = 16,28 Wh \approx 0,016kWh \quad [2]$$

### 2.5.1.3 DISEÑO PLACA DE ROBOT SUBMARINO Y CONTROL

El diseño de la PCB para la placa de control del Robot Submarino se la hizo en el programa PCBWizard, dicho programa es libre y brinda muchas comodidades al momento de trabajar incluso trabaja a la par con el programa Livewire que es en donde se hacen los diseños electrónicos y dicho diseño puede ser convertido directamente a PCBWizard, el programa ubicara de la mejor manera los elementos con su respectivo conexionado, si no es esto lo que se desea se lo puede hacer de forma manual, de esta manera se realizó el diseño ya que elementos como la Arduino mega mini pro no existen aún diseñados.

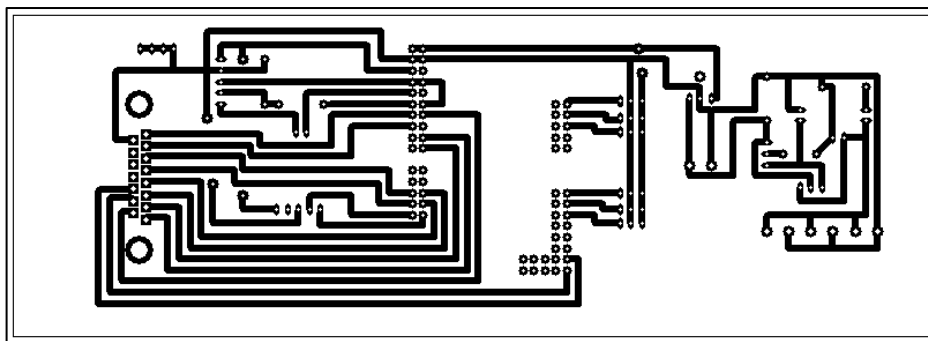


Figura 35 Diseño Pistas para PCB del Robot Submarino

Fuente: El Autor

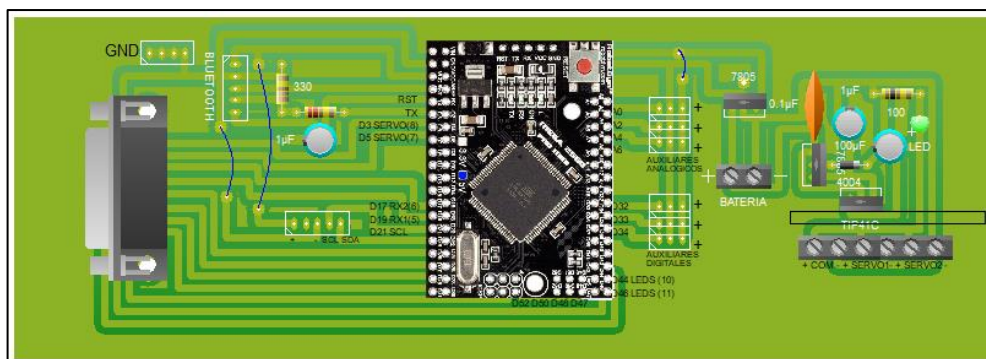


Figura 36 Diseño de la PCB del robot

Fuente: El Autor



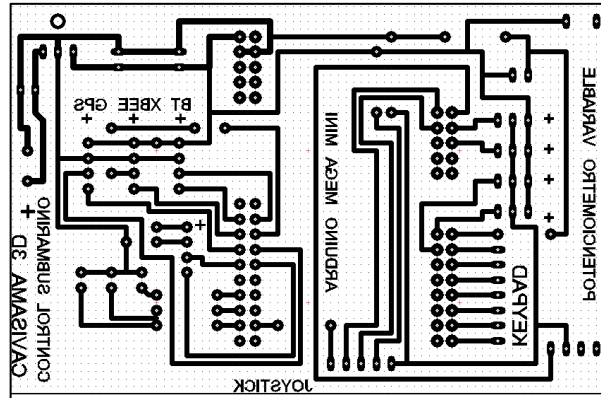


Figura 37 Diseño pista PCB Control Inalámbrico

Fuente: El Autor

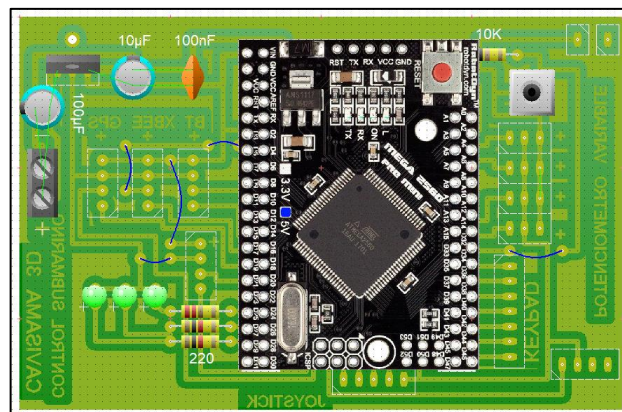


Figura 38 Diseño del PCB del Control

Fuente: El Autor

Para pasar los diseño (Figura 35 y 37) a la baquelita se usó el método de la plancha, en el programa PCBWizard para poder imprimir los diseño de las pistas se selecciona todo el rectángulo se copia y pega en el programa Word, una vez ahí se lo puede imprimir directamente en una hoja termotransferible en una impresora láser, en mi caso se imprimió con la mejor calidad en una hoja A4 y se sacó una copia en lámina de acetato, una vez transferida la imagen a la placa se repasó con marcador permanente la pista (Figura 49 y 40) para conseguir mejor acabado y que las pistas no se terminen cortando por la acción del ácido.

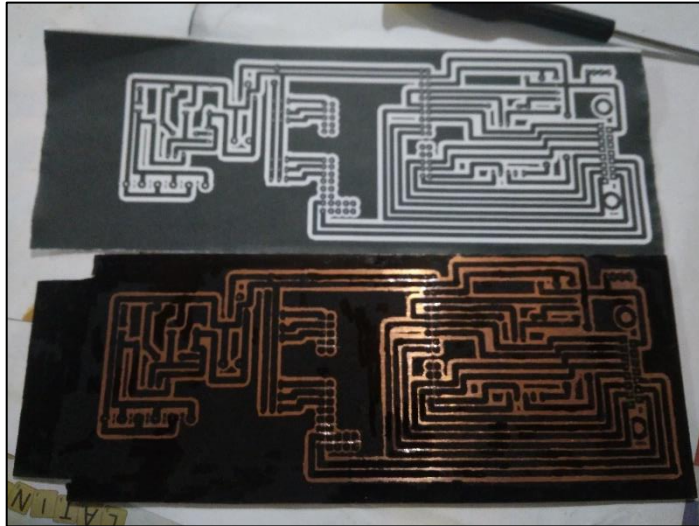


Figura 39 Diseño Placa Robot Submarino

Fuente: El Autor

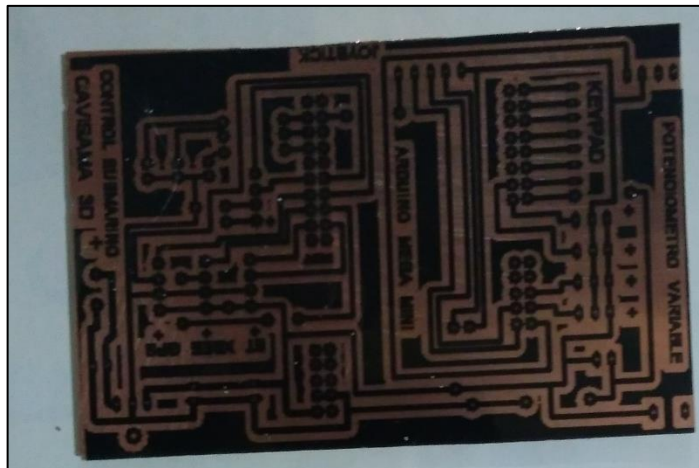


Figura 40 Diseño Placa Control

Fuente: El Autor

El siguiente paso es colocarlos en un recipiente plástico con suficiente Cloruro Férrico para conseguir quitar las partes q no se cubrieron y queden las pistas definidas (Figura 41).



Figura 41 Baquelita Limpia Luego de Pasarla por Cloruro Férrico

Fuente: El Autor

Para cerciorar que no haya errores en las pistas se mide la continuidad entre ellas definiendo cuales deberían y cuales no deberían estar conectadas (Figura 42), al terminar la revisión las placas estaban listas para ser perforadas, colocar los elementos, luego soldar conforme el diseño previamente realizado (Figura 43).

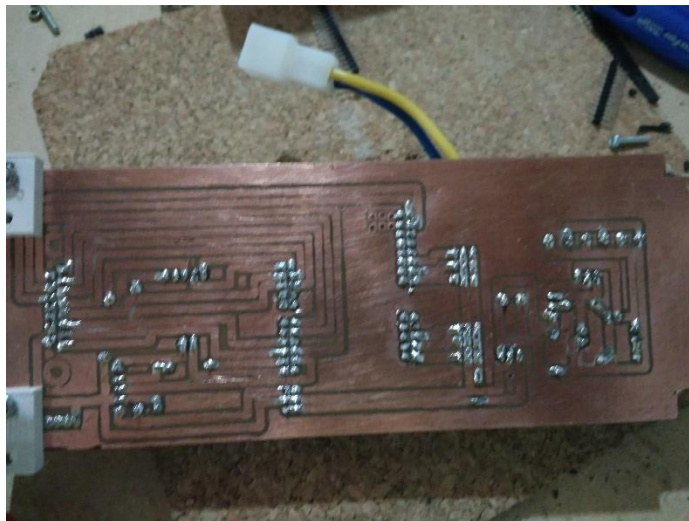


Figura 42 Placa Robot terminada de soldar

Fuente: El Autor

El siguiente paso es la verificación del funcionamiento de cada uno de los elementos correctamente



Figura 43 Placa Robot submarino

Fuente: El Autor

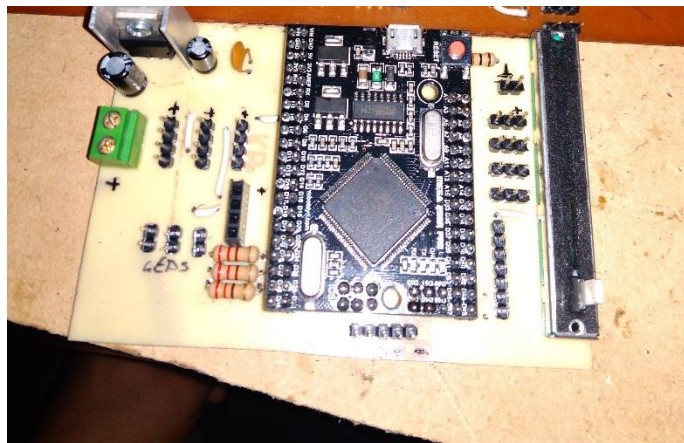


Figura 44 Placa Control Inalámbrico

Fuente: El Autor

Para la torre de comunicaciones se realizó una placa con diodos leds colocados en la torre de comunicación que sirve como indicador de que el Arduino se encuentra operativo (Figura 45), dependiendo del estado de la programación estos diodos van a parpadear de diferente modo.

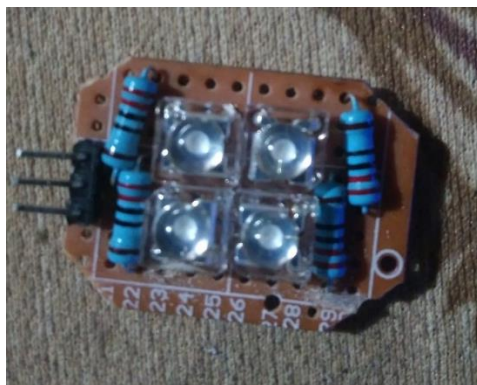


Figura 45 Diodos Led Torre de Comunicaciones

Fuente: El Autor

#### 2.5.1.4 SISTEMA DE CONTROL

La tarjeta Arduino en el proyecto es el cerebro del robot ya que controla cada una de las acciones que esta realiza dependiendo del modo en el que se encuentre, controla los actuadores en estado automático y depende de los datos que reciba por parte del magnetómetro para poder tomar la decisión de giro o de inmersión, todo controlado para un tiempo de desplazamiento del robot, en la Figura 46 se aprecian las conexiones entre el Arduino, los sensores y las salidas PWM (Modulación Por Ancho de Pulso) para los propulsores.

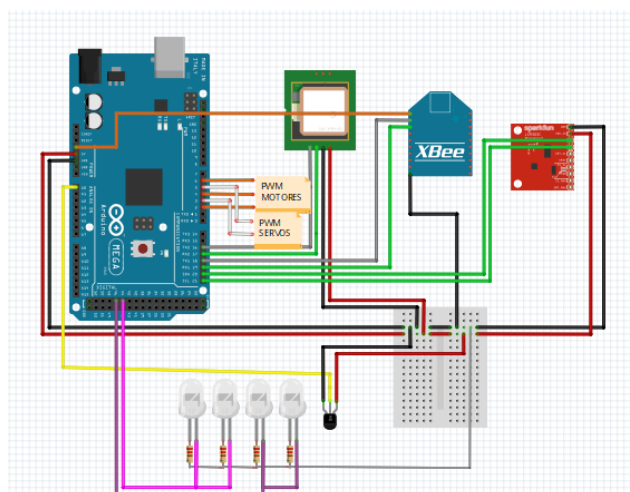


Figura 46 Esquema De Conexiones De Los Sensores Y Actuadores Del Robot

Submarino

Fuente: El Autor

### 2.5.1.5 SENSORES

#### Módulo Bluetooth HC05

Este dispositivo se encuentra ubicado en el control remoto y sirve para conectar al Arduino con la aplicación móvil a una distancia no mayor a 15 metros está conectado a los pines D14 y D15, que representan al puerto serial 3 con Tx3 y Rx3 respectivamente, y se alimenta directamente del regulador de voltaje.

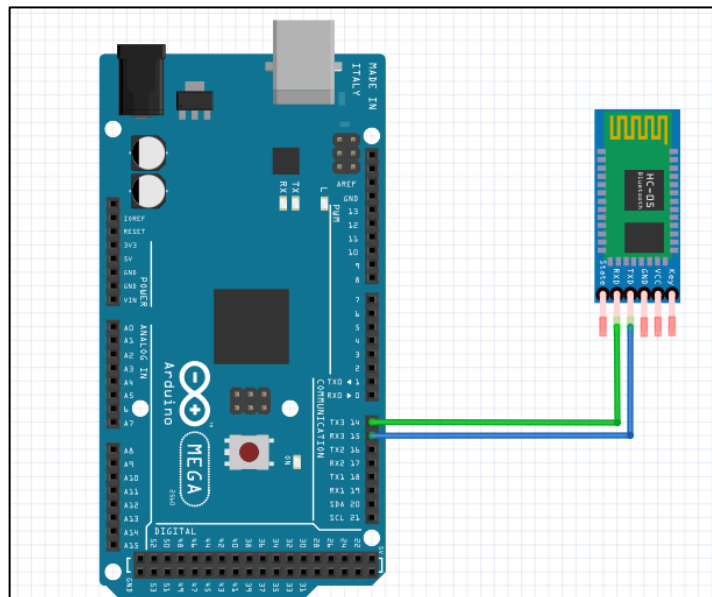


Figura 47 Diagrama Conexión Bluetooth En Fritzing

Fuente: El Autor

#### Gps Neo 6m

De estos dispositivos necesitamos dos, uno para el submarino y otro para el control inalámbrico, la utilidad en ambos es similar, consiste en recopilar el dato de posición de latitud y longitud para que la Arduino lo procese, en el robot Submarino el dato será reenviado vía inalámbrica hacia el control, el Arduino que hay en el control se encargará de recibir el dato enviado por el robot submarino y el dato que recibe de su propio GPS, mediante fórmula matemática se obtiene los valores de distancia y la trayectoria que hay entre los dos dispositivos. En el robot submarino el módulo está conectado al puerto serial dos y en el control inalámbrico al puerto serial uno.

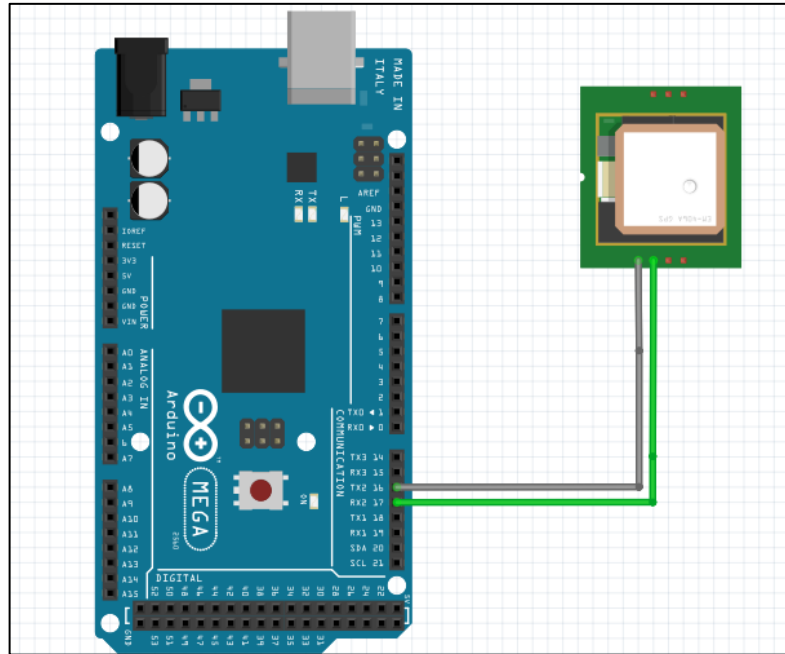


Figura 48 Diagrama Conexión del GPS en el Robot

Fuente: El Autor

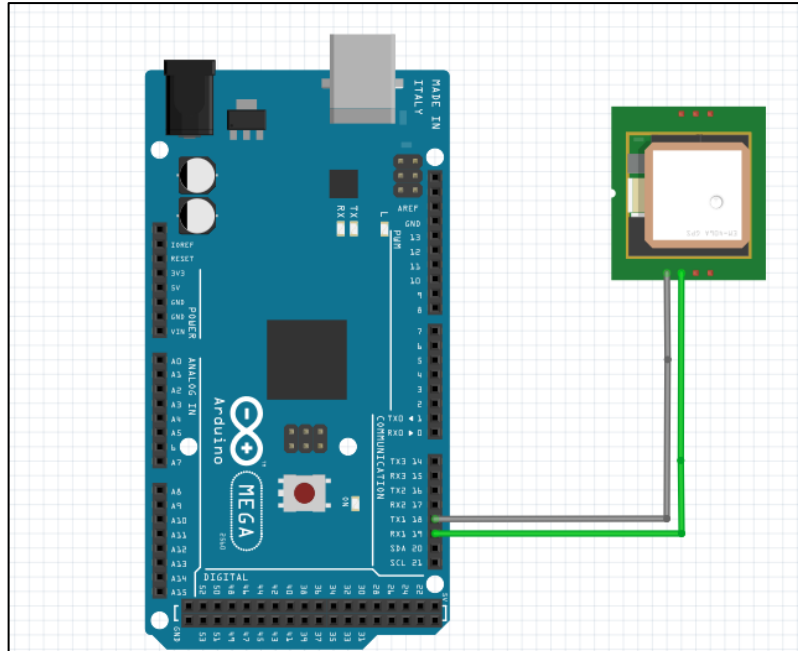


Figura 49 Diagrama Conexión del GPS en el Control

Fuente: El Autor

### GY89 Sensor LSM303

El sensor LSM303 es la brújula digital o giroscopio encargada de proporcionar los datos del rumbo en valores azimutales que van desde los 0° a los 360°, la comunicación que hay entre este dispositivo y el Arduino es por el protocolo I2C conectados a los pines SDA y SCL (Figura 50).

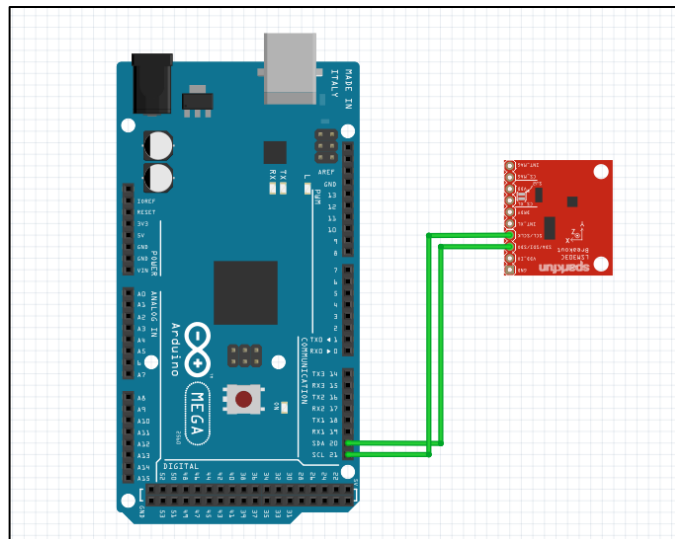


Figura 50 Diagrama Conexión del LSM303 en el Robot

Fuente: El Autor

En el modo automático, el robot dependiendo de la programación se dirigirá hacia los diferentes puntos cardinales, como por ejemplo cumplir un rectángulo dirigirse primero al norte luego seguir en línea recta por 10 segundos, buscar el este, seguir en línea recta por 10s, buscar el sur, seguir en línea recta por 10s, buscar el oeste y por último avanzar por 10 segundos más, la distancia recorrida variará porque depende mucho de la perturbación del movimiento del agua la velocidad aproximada del prototipo es de 1,5m/s a 2m/s eso quiere decir que recorrerá aproximadamente 15 a 20 metros en cada recorrido.

### XBee

Los dispositivos inalámbricos XBee son equipos de gran alcance llegando hasta máximo de 200 metros para su comunicación, en el robot submarino está conectado



en los pines 18 y 19 representando el puerto serial 1 del Arduino (Figura 51), y en el control inalámbrico el pin 16 y 17 que son el puerto serial 2 (Figura 52).

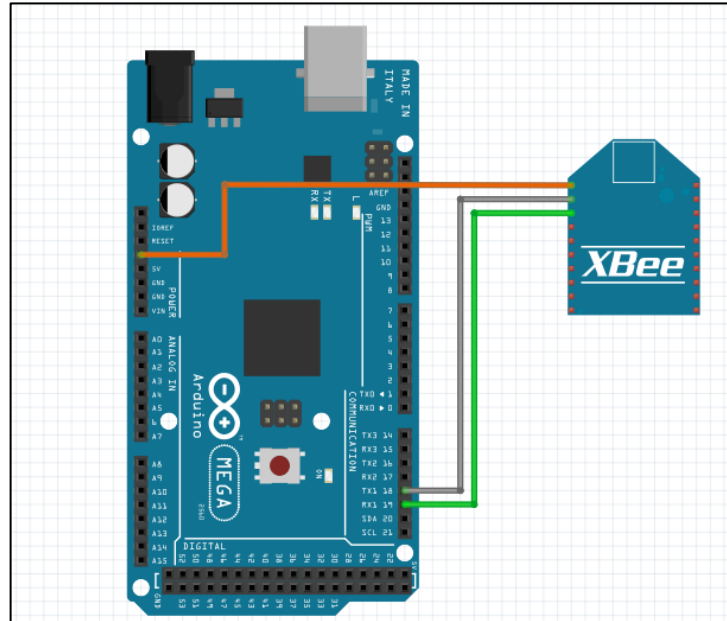


Figura 51 Diagrama Conexión del Xbee en el Robot

Fuente: El Autor

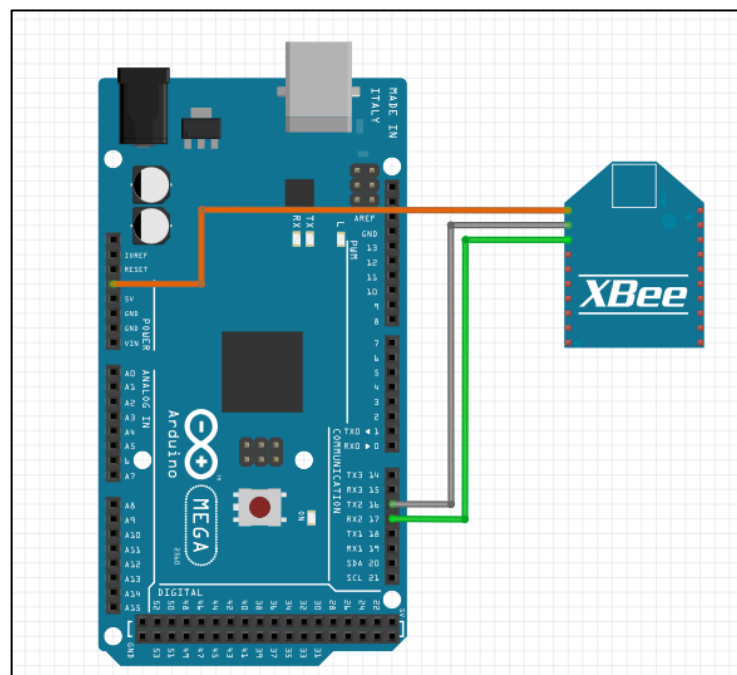


Figura 52 Diagrama Conexión del Xbee en el Control

Fuente: El Autor

## Sensor LM35

Es un sensor de temperatura de alta precisión, los datos entregados son en grados centígrados, tiene una precisión calibrada de  $1^{\circ}\text{C}$  y el rango es desde los  $-55^{\circ}$  a  $+150^{\circ}\text{C}$ , la salida lineal equivale a  $10\text{mV}/^{\circ}\text{C}$  eso significa que  $+1500\text{mV} = 150^{\circ}\text{C}$ ,  $+250\text{mV} = 25^{\circ}\text{C}$  o incluso  $-55\text{mV} = -55^{\circ}\text{C}$ .

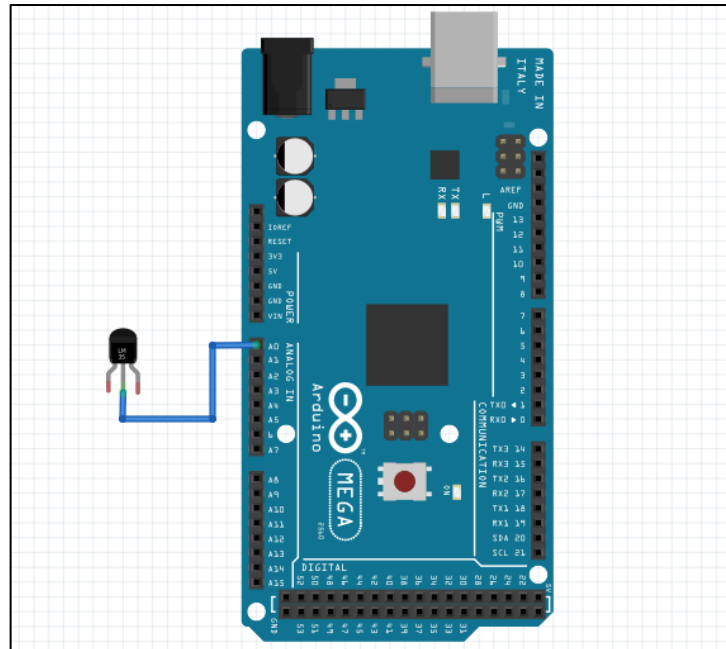


Figura 53 Diagrama Conexión del Sensor LM35 en el Robot

Fuente: El Autor

## 2.5.2 DISEÑO DE FIRMWARE

El firmware del robot y del control inalámbrico fueron realizados en el programa Arduino IDE con lenguaje C++ (Figura 10), al ser Arduino un dispositivo con Hardware y Software libre, tenemos un espacio más amplio para poder trabajar y encontrar herramientas que nos ayuden a desarrollar las secuencias de la programación como son las librerías, la mayoría de los dispositivos como sensores y actuadores tienen librerías versátiles para poder optimizar el uso del elemento.

### **2.5.2.1 FIRMWARE DEL ROBOT SUBMARINO**

Para poder comprender mejor el firmware se explicará primero lo que hace el diagrama de flujo (Figura 54) que es la base para el desarrollo del firmware, el robot al encenderse la primera acción que hará es verificar si hay datos de entrada, si no existieran es porque no hay comunicación con el control o simplemente el control está apagado, en este caso el robot tendrá que mantener los motores de propulsión apagados y las aletas o timones en una dirección central.

Los servos de los timones están ubicado a 90°, los servos utilizados son servos digitales por lo que se requiere que los valores sean enviados en términos de ancho de pulso en este caso el valor es de 1195 microsegundos, los motores T200 en su hoja de especificaciones indican que los valores en microsegundos van desde 1000 a 2000 microsegundos, con valores inferiores a los 1500 microsegundos el robot aumentara su velocidad pero en dirección opuesta o en reversa y con valores superiores el robot girará hacia adelante, el robot estará en reposo cuando el valor sea 1500 microsegundos.

Entre las opciones se maneja que si el control inalámbrico está en comunicación con el robot, este enviará los valores a los motores y servomotores respectivamente, claro está que solo podrá tener control sobre el robot cuando esté sobre la superficie del mar, en el momento que se da la orden de iniciar la rutina automática del robot, este comenzará a sumergirse y cumplir la trayectoria programada, una vez terminada la rutina este emergerá enviando una trama de datos respecto a su ubicación geográfica, en cada paso de la programación de la rutina existe la opción de abortar siempre y cuando exista comunicación con el control inalámbrico.

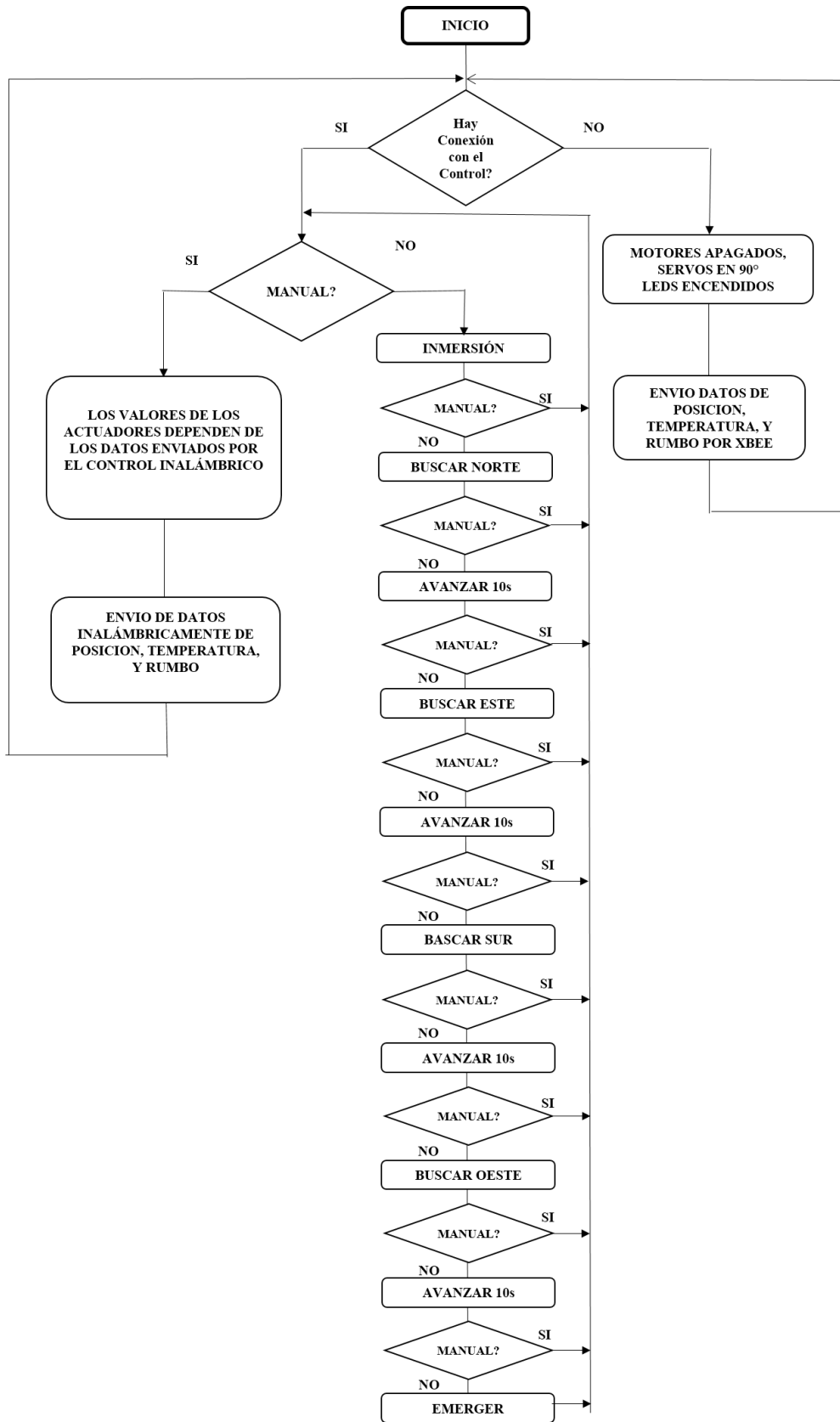


Figura 54 Diagrama de Flujo Robot Submarino

Fuente: El Autor

Para iniciar con el tema del firmware en el encabezado se colocaron las librerías usadas como son:

```
#include <TinyGPS.h>      Librería para uso del GPS NEO 6m
#include <Wire.h>          Librería para la comunicación I2C
#include <LSM303.h>,      Librería del sensor magnetómetro
#include <L3G.h>,         Librería del sensor giroscopio
#include<Servo.h>,        Librería para el control de los servos y motores
#include<EEPROM.h>        Librería para el uso de la memoria eeprom.
```

Seguido a las librerías se hace la declaración de variables que utilizadas para cada uno de los elementos del robot submarino, entre las variables declaradas se usan ESC1 Y ESC2 creadas como clase servo para controlar los motores brushless, dichos motores usan la librería de servos porque con ella se envía valores en microsegundos controlando el ancho de pulso requerido para accionarlos, entre otras variables también tenemos lat\_sub, lon\_sub, estas variables tipo float se utilizan para almacenar los datos de latitud y longitud recogidos por el GPS.

Dentro de la programación se separó la recolección de los datos de los diferentes sensores y el control de los actuadores, estas secciones fueron separadas en funciones para llevar un mejor orden, dichas funciones se utilizan dentro de la función principal o “void loop()” solo cuando sean requeridas.

Entre las diferentes funciones tenemos “void datos\_locomoción” donde se reciben los datos que llegan al Serial1 del Arduino, los datos que se reciben vienen desde el control inalámbrico como cadena de texto separada por comas, la función lo que hace es recibir los datos, separarlos cada vez que haya una coma y almacenarlos en una posición diferente, como por ejemplo el dato que llega en la posición cero es “verificación” que es un dato aleatorio entre cero y nueve, este se usará para identificar si estamos conectados con el control, cuando el dato que llega a esta ubicación es igual al anterior recibido significa que no hay conexión, mientras los datos recibidos y en comparación con el anterior son diferentes significa que si está

conectado con el control inalámbrico. Otros de los datos que recibe la función son velocidad, yaw, pitch y autónomo.

La siguiente función es “void motores (int velocidad)”, esta función depende de una variable que ingresa a ella, en este caso la variable de velocidad, que es la que se almacena en la función anterior “void datos\_locomoción”, la variable velocidad que son datos numéricos entre 0 y 1024 hay que mapearlos y hacer que se conviertan en datos que van desde 1500 hasta 1900, con 1500 indica que el motor está apagado y con 1900 indica el máximo de potencia.

Para la dirección e inmersión de los motores se usan las funciones “void direccional(int eje\_x)” y “void cabeceo(int eje\_z)” respectivamente, en la función direccional ingresan los datos de la variable pitch y en la función cabeceo los datos de la variable yaw, en vista que estos valores que llegan están entre 0 y 1023 en cada caso podríamos hacer un mapeo como en el caso de la velocidad, estos valores son enviados por el control mediante el uso del joystick, por cuestiones de calibración el dato enviado por el joystick estando en la parte central debería de estar en 512, en pruebas realizadas no es así enviando valores de 800 para el eje ‘x’ y 760 para el eje ‘y’ la solución fue usar la ecuación de la recta para poder así distribuir los valores que nos ayuden a posicionar bien los timones, en los servos digitales cuando se desea giren a 55° el valor en microsegundos que se necesita es de 847.5, para 90° es de 1195 y para 125° es de 1572,5.

### **Explicación de la parametrización de la Ecuación de la Recta utilizada en el Servomotor de Cabeceo**

<b>Datos recibidos</b>	<b>Tiempo en alto de la señal</b>	<b>Giro</b>
1023	1572,5 microsegundos	125°
760	1195 microsegundos	90°
0	847,5 microsegundos	55°

Tabla 3 Parametrización Servomotor Cabeceo

Fuente: El Autor

Condiciones

$$\text{eje\_z} \geq 0 \quad \text{eje\_z} \leq 760$$

Parámetro obtenido

$$z = 0.45723684 * \text{eje\_z} + 847.5 \quad [3]$$

Condiciones

$$\text{eje\_z} > 760 \quad \text{eje\_z} \leq 1023$$

Parámetro obtenido

$$z = 1.43536122 * \text{eje\_z} + 104.1254753 \quad [4]$$

### **Explicación de la parametrización de la Ecuación de la Recta utilizada en el Servomotor de dirección**

<b>Datos recibidos</b>	<b>Tiempo en alto de la señal</b>	<b>Giro</b>
1023	847,5 microsegundos	55°
800	1195 microsegundos	90°
0	1572,5 microsegundos	125°

Tabla 4 Parametrización Servomotor direccional

Fuente: El Autor

Condiciones

$$\text{eje\_x} \geq 0 \quad \text{eje\_x} \leq 800$$

Parámetro obtenido

$$x = -0.47187500 * \text{eje\_x} + 1572.5; \quad [5]$$

Condiciones

$$\text{eje\_x} > 800 \quad \text{eje\_z} \leq 1023$$

Parámetro obtenido

$$x = -1.55829596 * eje_x + 2441.636771; \quad [6]$$

Para poder controlar el robot mientras no hay comunicación el desplazamiento debe ser cero, eso significa que los propulsores deben de estar apagados y los timones centrados, por eso en la función “void desplazamiento (int a, int b, int c)” se maneja este caso, las variables de ‘a’ para motores, ‘b’ para timón de cabeceo y ‘c’ para el timón de la direccional.

La función “void gps\_submarino()” es donde recibimos los datos de la posición georreferenciada del robot, aquí se hace el llamado a la librería TinyGPS.h conectado al puerto serial dos, con la instrucción restringida que usa la librería denominada gps.f\_get\_position podemos obtener los valores para latitud y longitud que usaremos para reenviar hacia el control inalámbrico.

Para los sensores como el magnetómetro usamos la función “void magnetometro()”, de la librería LSM303.h hacemos la lectura del sensor con la instrucción compass.heading() que es una función propia de la librería, obtenemos el valor azimutal del sensor magnetómetro que usaremos para dirigir al robot en modo automático.

En la función “void\_temperatura” recibimos un valor analógico que mediante una formula dada por el fabricante obtenemos el valor en grados Celsius [7]. Para recibir mejor el dato del sensor se realiza un filtro, que consiste en sumar los veinte valores entrantes y obtener un promedio que significaría que el resultado de la suma se divide para 20.

$$A = (5.0 * (LM35) * 100.0) / 1023.0 - 16 \quad [7]$$

Una parte muy importante es la comunicación bidireccional que tenemos, eso consiste en poder enviar datos y recibirlos, como ya se mencionó recibimos datos del control inalámbrico y ahora en este caso vamos a enviar datos como el de posición, que consiste en datos de latitud y longitud, valores de la temperatura y del



magnetómetro, la función que está encargada de realizar este envío es “void envio\_datos()” que mediante el puerto serial1 se transmite, cabe indicar que al recibir los datos el control inalámbrico en la cadena de texto solo recibirá datos enteros, por ello se multiplican por un valor entero grande, como por ejemplo el valor de latitud -2.224543, para poder enviar y recibir todo el dato lo multiplicamos por 1000000 y como resultado de esta operación se obtendrá un valor entero, el valor entero que llega al control inalámbrico se hace el proceso inverso y lo dividiremos por 1000000.

En la torre de comunicaciones está ubicado los indicadores led que son un conjunto de 4 sensores que trabajan de par en par, las funciones “void parpadeo\_manual()” y “void parpadeo\_auto()” hacen que los leds parpadeen a diferentes frecuencias dependiendo del caso.

Para la parte autónoma existen varias funciones dependiendo de la orientación, estas funciones son: “void norte”, “void sur”, “void este”, “void oeste”, en cada función se controlan los actuadores y dependiendo del dato que se reciban del magnetómetro el giro va a ser diferente, por ejemplo, si necesitamos ir hacia el sur el valor es de  $180^\circ$  y si el robot inicialmente se encontrara al este con valor aproximado a los  $90^\circ$  el proceso que debe realizar es girar hacia la derecha hasta que la brújula indique  $170^\circ$ , si la posición del robot es diferente y se encontrara inicialmente apuntando hacia el oeste con valor aproximado de  $270^\circ$  deberá de girar hacia su lado izquierdo, el margen de grados para dirigirse a una dirección es de  $20^\circ$ ; es decir si vamos hacia el sur y el valor del magnetómetro está entre  $170^\circ$  y  $190^\circ$  el robot debe de ir hacia adelante con valores de 1900 microsegundos para los propulsores y 1195 microsegundos para los actuadores.

Para sumergir y emerger el robot tenemos las funciones “void sumergir” y “void emerger”, en ambos casos el robot cumple la trayectoria en forma de espiral, para sumergirse el timón de inmersión-emersión se inclina hacia abajo y el de dirección gira hacia la izquierda (Figura 55), esto hace que al desplazarse a más de sumergirse también gire, al emerger hace lo mismo, la diferencia es que el timón de inmersión-

emersión se inclinara hacia arriba y el timón de dirección girará hacia la derecha (Figura 56).



Figura 55 Robot sumergiendo

Fuente: El Autor



Figura 56 Robot emergiendo

Fuente: El Autor

En la función principal “void loop ()” se establecen las condiciones para el uso de las diferente funciones, para empezar se hace el llamado a la función “datos\_locomoción()” que es la que recibe información que es enviada por el control inalámbrico, en la parte final del “void loop()” tenemos que el valor de la variable ‘verificación’ se sobrescribirá siempre en la dirección cero de la memoria eeprom y luego se procede a leer esa misma dirección y ese valor lo toma la variable ‘verificación\_pasada’, esto se hace para luego verificar si el valor que está llegando es igual o diferente al anterior.

El dato que tiene la variable ‘autónomo’ se guardará en la dirección de memoria uno, pero con una restricción, que ese dato debe de ser cero o uno, si el dato es

diferente se descarta, el dato que esta almacenado en la memoria se atribuye a la variable 'auto1', la restricción radica que cuando existe una desconexión suelen llegar datos erróneos y podrían ser mal interpretados.

Para iniciar las acciones con los actuadores contamos con una sentencia if anidada por otra sentencia if en su parte verdadera y en la parte falsa un case anidado, la pregunta principal es si 'auto1' es igual a cero, si esta pregunta es verdadera pasaremos al if anidado, donde la siguiente pregunta es si el valor de verificación es diferente al de 'verificación\_pasada' y si la respuesta es verdadera significa que estamos en control manual y recibimos los valores que envía el control inalámbrico que es quien gobernaría la motricidad del robot, caso contrario quiere decir que los valores son iguales y por ende no existe comunicación con el control inalámbrico, las razones pueden ser porque está apagado o el robot esta fuera del rango que serían 200 metros, en esta situación el robot debe de mantener sus propulsores apagados y los timones en posición central.

Si la condición de 'auto1' es igual a cero por falso entra a modo automático y a la sentencia 'case' donde se ha escrito la rutina que el robot debe de seguir, dicha rutina están separadas en casos, por ejemplo en el caso uno el robot empezara a sumergirse, cumplido este caso pasa al siguiente que es buscar el norte, en total son seis casos y en cada uno existe la opción de abortar por si sucede algún problema mecánico y amerite hacerlo, claro está que solo se podrá abortar cuando exista comunicación directa con el control inalámbrico, en el Anexo 1 tenemos la programación del Robot Submarino.

### **2.5.2.2 FIRMWARE DEL CONTROL INALÁMBRICO**

Al igual que con el diseño del robot empezaremos con el diagrama de flujo (Figura 57), el control inalámbrico consiste en recibir los datos enviados por el submarino, utiliza los valores de latitud y de longitud para calcular la distancia entre los dos puntos, dirección y rumbo, mientras el control inalámbrico está en modo manual envía hacia el robot los parámetros para que los actuadores ejecuten alguna acción

y también es el que muestra al operador si desea comenzar la secuencia automática, otra función que realiza el control es el envío de información hacia la aplicación móvil.

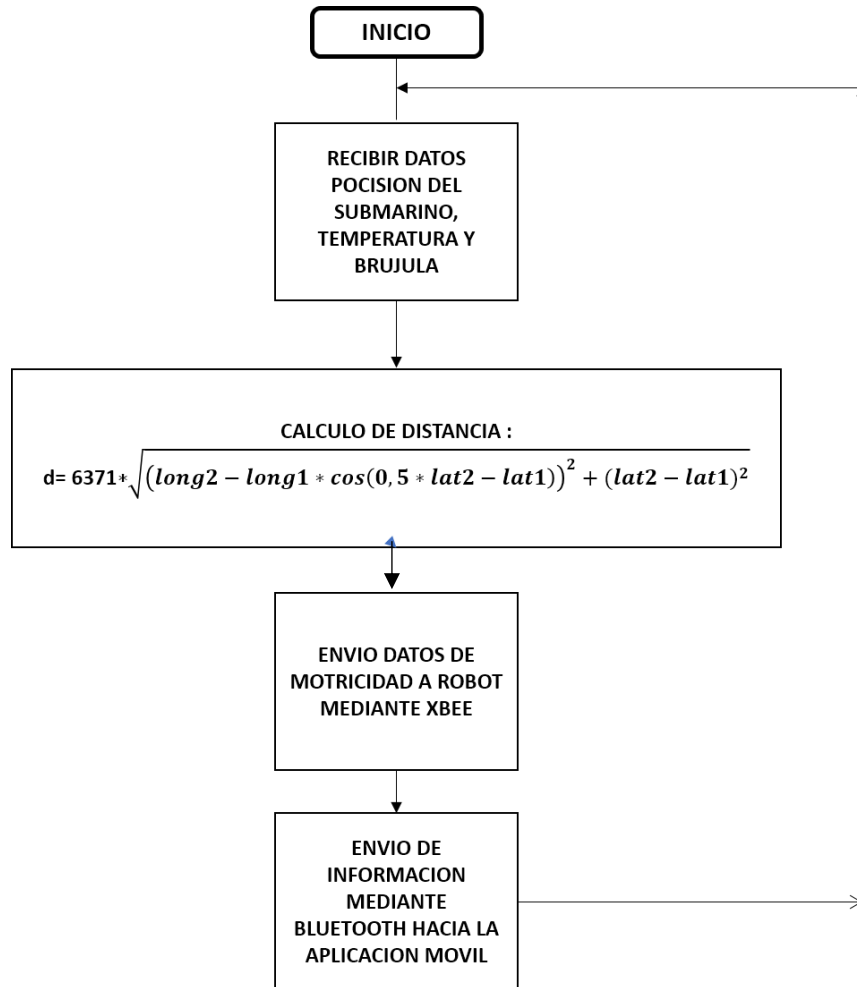


Figura 57 Diagrama de Flujo Control Remoto

Fuente: El Autor

El firmware del control empieza con el llamado de las librerías en este caso dos:

```

#include <TinyGPS.h>    Librería del módulo GPS
#include <EEPROM.h>    Librería para trabajar con la memoria eeprom
  
```

En este programa también usamos las funciones que son las encargadas de resolver los procesos, tenemos “void pulsador ()” cuya función es la de seleccionar en el robot submarino el modo automático o el modo manual.

Para poder seleccionar el modo automático en el robot submarino, la función se encargaría de verificar tres condiciones de forma simultánea, una variable llamada ‘estado’, una variable llamada ‘estadoAnterior’ y una variable de entrada analógica, la variable estado está conectada al pin 32, cada vez que se pulsa el dato es asignado a la variable ‘estadoAnterior’ y como seguridad se utiliza también el eje direccional del joystick ubicado en el pin analógico uno, la variable ‘automatico’ tomará valores de cero y uno.

El control inalámbrico posee también un sensor GPS Neo 6m al igual que el robot submarino recibimos los datos georreferenciales de latitud y longitud, estos valores se almacenan en las variables ‘lat1’ y ‘lon1’ respectivamente, el proceso de recolección de datos sucede en la función “void gps\_a\_bordo()”.

En la parte física del control inalámbrico se encuentra el potenciómetro lineal y el joystick conectados a pines analógicos, la función “void control\_manual ()” almacena los valores recibidos de los elementos, el potenciómetro lineal esta conectado al pin A0 con la variable ‘velocidad’, el desplazamiento horizontal del joystick al pin A1 con la variable ‘pitch’ y el desplazamiento vertical al pin A3 con la variable ‘yaw’, la lectura analógica en el Arduino consiste en valores que varían desde 0 a 1023 y estos valores son los que se enviarán inalámbricamente al robot por medio del puerto serial 2 del Arduino mega, otro parámetro que se realiza en esta función es la verificación de conectividad que consiste en enviar números ordenados que van del cero al nueve de una manera constante, el valor numérico es almacenado en la variable ‘i’. Junto a las variables ‘i’, ‘velocidad’, ‘yaw’ y ‘pitch’ también se envía la variable ‘val’ que se recibe de la función “void pulsador ()”.

El robot submarino envía cuatro parámetros que son datos de latitud, longitud, temperatura y del magnetómetro estos valores se los reciben en la función “void gps\_recibido ()”, los datos de recepción son recibidos en el puerto serial 2 donde

físicamente se encuentra conectado el dispositivo XBee, los datos llegan en una cadena de texto separando cada dato con una coma, la cadena se almacena en la variable 'data', cada dato de entrada se guarda en una posición diferente en la variable para luego ser recuperado y asignado a una nueva variable, en este caso a la variable 'lat\_sub' el valor de la posición cero, 'lon\_sub' el valor de la posición uno, 'temperatura' el valor de la posición dos y a 'magnetómetro' el valor de la posición tres, a las variables 'lat\_sub' y 'lon\_sub' se les realiza el proceso inverso a lo que se realizó en la programación del robot submarino donde se los multiplica por un entero para eliminar los decimales, en la programación del control inalámbrico se usa el mismo valor para una división.

Una de las partes importantes para la búsqueda del robot es conocer la distancia a la que nos encontramos de él y para aquello se debe calcular dicha distancia en base a datos de posicionamiento georreferencial de cada uno de los equipos involucrados, el método más conocido para calcular distancias entre dos puntos es el Teorema de Pitágoras [8] inclusive si la distancia que se requiere medir es una distancia terrestre el resultado del Teorema se multiplica por el diámetro de la tierra, se realiza la multiplicación porque el resultado del teorema es en radianes y será un valor muy pequeño, al multiplicar este valor la respuesta de la distancia quedara en kilómetros (Km) que hay entre el control inalámbrico y el robot, para que la distancia sea en metros se hace la conversión de unidades, este método solo sirve si la distancia es teóricamente pequeña no para distancias que sobrepasen los 20 km aproximadamente, en el caso del robot submarino la comunicación con el control inalámbrico no sobrepasa los 200 metros y no habría problema en utilizar esta fórmula (Straigh-Line Distance, s.f).

$$c^2 = a^2 + b^2 \quad [8]$$

Como ejemplo de la exactitud de los resultados de la formula se hizo la comparación con conversiones presentados en páginas web que dan el servicio de calculo de distancia terrestre como "<https://www.sunearthtools.com/es/tools/distance.php>", un ejemplo claro es el siguiente:

- Los valores de latitud y longitud son convertidos de grados a radianes

$$\text{Latitud1: } -2,22523^\circ = -0,038837681 \quad \text{Longitud1: } -80,85856^\circ = -1,411251401$$

$$\text{Latitud2: } -2,22501^\circ = -0,038833841 \quad \text{Longitud2: } -80,85854^\circ = -1,411251051$$

- Se calcula la diferencia entre latitudes y longitudes

$$\Delta\text{lat} = \text{latitud2} - \text{latitud1} \quad [9]$$

$$\Delta\text{lat} = -0,038833841 - (-0,038837681)$$

$$\Delta\text{lat} = \mathbf{0,00000383973}$$

$$\Delta\text{lon} = \text{longitud2} - \text{longitud1} \quad [10]$$

$$\Delta\text{lon} = -1,411251051 - (-1,411251401)$$

$$\Delta\text{lon} = \mathbf{0,0000003491}$$

- Se procede al cálculo con el teorema de Pitágoras

$$c = \sqrt{\Delta\text{lat}^2 + \Delta\text{lon}^2} \quad [11]$$

$$c = \sqrt{0,00000383973^2 + 0,0000003491^2}$$

$$c = 3,85557 \times 10^{-6}$$

- El valor obtenido está dado en radianes que representa a la fracción de una circunferencia por lo que se multiplica por el diámetro de la tierra que es 6371Km

$$D = R * c \quad [12]$$

$$D = 6371 * 3,85557 \times 10^{-6}$$

$$D = 0,02456 \text{ Km}$$

$$D = 24,56 \text{ m}$$

El resultado de la aproximación calculada y basada al teorema de Pitágoras es de 24,56 metros, comparada con la aplicación web tenemos 24.6 m, también podemos calcular la dirección del vector para dirigirnos hacia el punto.

$$\theta = \tan^{-1} \frac{\Delta lon}{\Delta lat} \quad [13]$$

$$\theta = \tan^{-1} \frac{0,0000003491}{0,00000383973}$$

$$\theta = 5,194$$

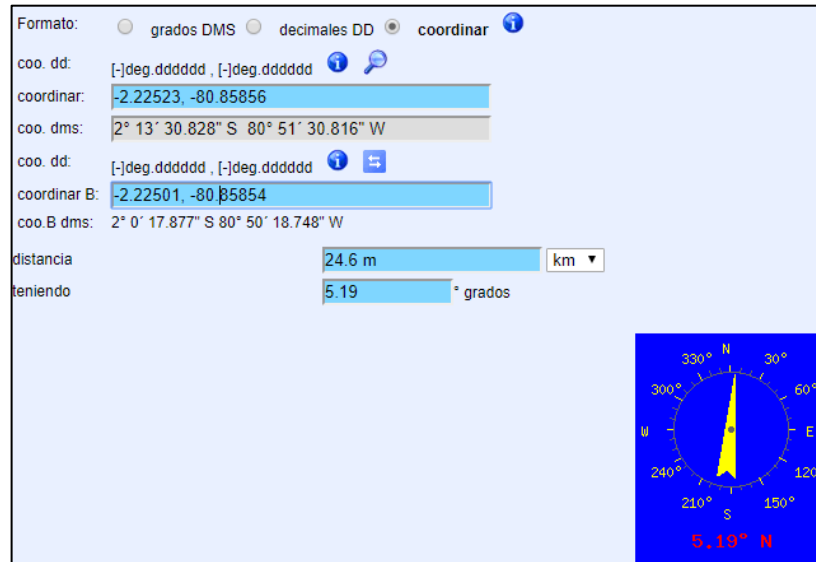


Figura 58 Cálculo Distancia y Dirección

Fuente: <https://www.sunearthtools.com/es/tools/distance.php>

Los cálculos referentes a la distancia entre el robot submarino y el control remoto se realizan en la función “void medir\_distancia ()”, en la función se recopilan los datos de la posición de latitud y longitud tanto del control inalámbrico como del robot submarino y se aplican las fórmulas [9], [10], [11], [12].

En el programa principal como es el “void loop()” está el llamado de cada una de las funciones empezando con la recopilación de los datos enviados al control inalámbrico, la recepción de los valores del gps del control, el cálculo de la distancia y los valores de pulsador, el void loop solo consta de una sentencia if que es la encargada de realizar el envío de los datos de manera inalámbrica por el puerto serial tres, donde físicamente se encuentra conectado a el módulo bluetooth, la información enviada por este medio es la que se receptorá en la aplicación móvil, en el Anexo 2 tenemos la programación del control inalámbrico.



### 2.5.3 DISEÑO DE APLICACIÓN ANDROID

La aplicación se la realizó en el MIT App Inventor 2 (Figura 20) consta de 2 pantallas una que es la carátula de presentación (Figura 59) y otra donde se reciben todos los datos, por la configuración que se realizó en la segunda pantalla existen paginas ocultas que aparecerán cuando se pulse en el icono respectivo en la parte superior izquierda de la pantalla que es donde se encuentra en el menú desplegable (Figura 60), cada página tiene un botón de regresar y redirecciona hacia la página principal, el diseño contempla Página Principal, Ubicación, Rumbo y Temperatura.



Figura 59 Pantalla N°1 de la Aplicación Carátula

Fuente: El Autor

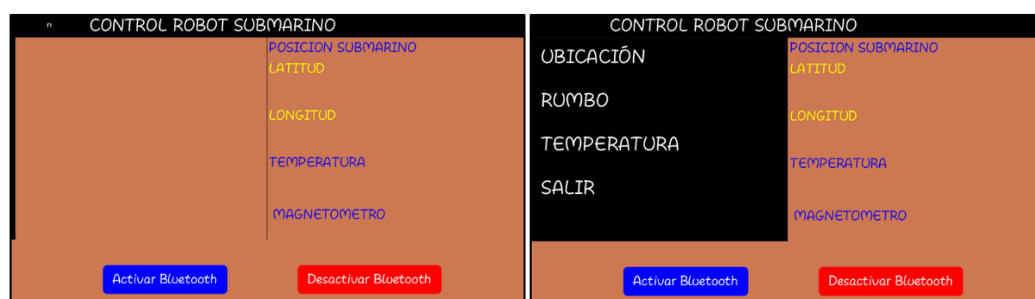


Figura 60 Pantalla N°2 de la Aplicación Página Principal con Menú Desplegable

Fuente: El Autor

En la página principal (Figura 60) se verá la información que le llega desde el Robot Submarino y que son reenviados hacia la aplicación móvil, la información que llega son Latitud, Longitud, Temperatura y datos del Magnetómetro, en esta página tenemos la opción de activar y desactivar el bluetooth, adicional a esto se ha incluido iconos con sonidos para que alerte si se ha realizado la acción por ejemplo al presionar el botón de Activar Bluetooth se escuchará “BLUETOOTH ACTIVADO” y si se lo desactiva “BLUETOOTH DESACTIVADO”

Al desplegar el menú podemos seleccionar entre las siguientes opciones Ubicación, Rumbo, Temperatura y Salir, al presionar el botón Ubicación se abrirá una ventana con la apariencia de Google Maps esto se debe a que se utilizan tablas dinámicas y cada vez que se presiona el botón grabar, la información del gps del control y del submarino se guardan en las tablas, después del guardado hay una actualización en el mapa mostrado y aparecen las coordenadas guardadas últimamente, la información se muestra en mapas diferentes y para intercambiar pantallas se debe presionar el botón visualizar, la manera de interpretar que mapa estamos visualizando se destaca en la parte superior izquierda donde se indica el mensaje “Mi Posición” que significa la posición del control inalámbrico y “Posición Submarino” que significa la posición georreferenciada del robot.

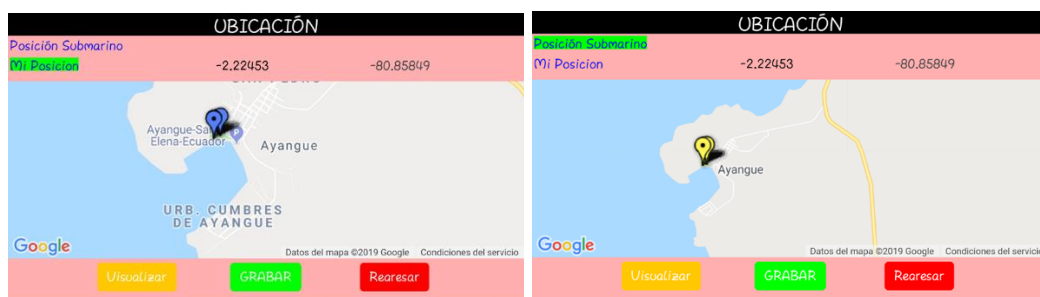


Figura 61 Página de Ubicación en la Aplicación Móvil

Fuente: El Autor

Para poder visualizar otra página debemos presionar el botón de Regresar que nos llevará a la Página Principal para nuevamente desplegar el menú y hacer una nueva selección, ahora en este caso RUMBO (Figura 62) en esta página se muestra la ubicación y la distancia aproximada que hay entre el control inalámbrico y el robot.



Figura 62 Pagina del Rumbo en la Aplicación Móvil

Fuente: El Autor

El programa MIT App Inventor 2 utiliza sistema de bloques para realizar la programación, la página de la caratula consiste en que si se presiona el botón de empezar nos envíe a la siguiente pantalla (Figura 63).

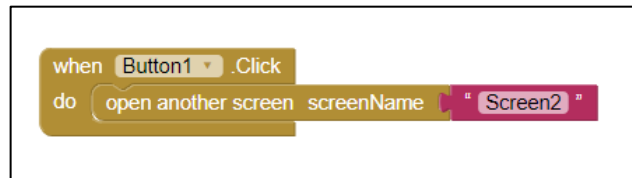


Figura 63 Programación Pagina Carátula de la Aplicación Móvil

Fuente: El Autor

En la sección de presentación de información del robot submarino, el control inalámbrico envía los datos en una sola trama compuesta por cantidad y separado por comas, la programación que se muestra en el bloque de la Figura 64 crea una variable global donde se almacena la cadena de texto que proviene del control inalámbrico y selecciona el dato según la posición, en la cadena se reciben los datos de latitud del submarino, longitud del submarino, temperatura del submarino, valor del magnetómetro, rumbo y la distancia entre el submarino y el control inalámbrico, lo que hacemos es direccionar y presentar en los casilleros correspondientes dichos valores.

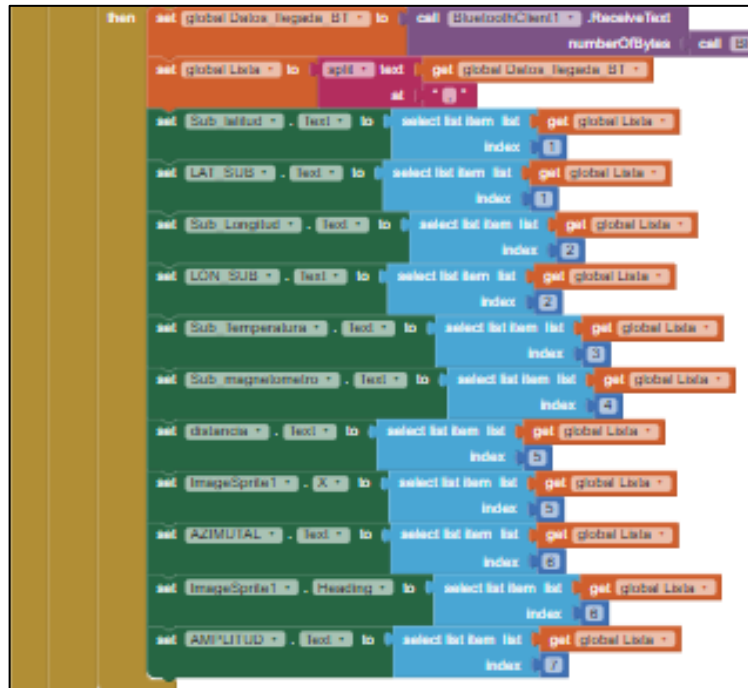


Figura 64 Programación Recepción de datos Aplicación Móvil

Fuente: El Autor

Para realizar el menú desplegable lo que hacemos es ocultar los botones en una barra de diseño que se deja con el valor de ancho mínimo, pero al presionar el botón de menú el programa altera este ancho y lo amplifica dejando visualizar los botones escondidos, si se presiona por segunda ocasión el botón de menú este vuelve a su estado reducido (Figura 65).

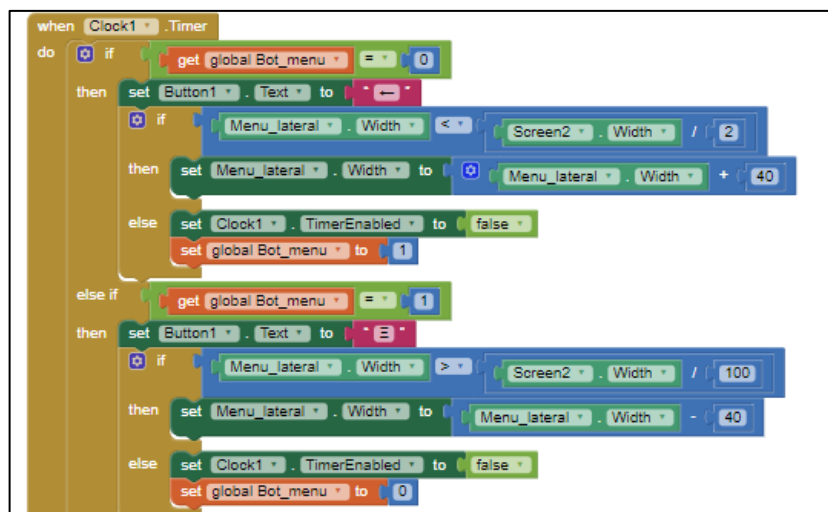


Figura 65 Programación Para el Menú desplegable

Fuente: El Autor

La pantalla de ubicación muestra los datos recibidos de la Posición del Submarino y Mi Posición que viene a ser la posición del dispositivo móvil (Figura 66), para almacenar los datos en necesario que el dispositivo móvil tenga conexión a Internet y es decisión del operador grabar los puntos referenciales pulsando el botón ‘GRABAR’, una vez hecha la acción los datos de Posición Submarino y Mi posición son grabadas en una hoja dinámica ubicada en Google Drive, existen 2 mapas en la pantalla pero se puede ver solo uno a la vez, cada mapa muestra la ubicación de uno de los dos puntos almacenado, cada ocasión que se guarde información los mapas se actualizarán y para poder revisarlos se pulsa el botón ‘VISUALIZAR’, la pantalla también cuenta con el botón de ‘REGRESAR’ que nos conduce a la pantalla principal. La programación de la pantalla rumbo se encuentra inmersa en las programaciones anteriores donde se trata de direccionamiento.

```

when LocationSensor1 LocationChanged
do
  set [Mi_latitude] .Text to [get latitude]
  set [Mi_longitude] .Text to [get longitude]

  to [Poner_Comillas] Cadena
  result [join]
  [get Cadena]

when GRABAR Click
do
  call [FusiontablesControl1] .InsertRow
  tableId [15TeQLLhCuNrq8sfUQNuoXDrxRYkdHMvqvidis5L]
  columns [FECHA_HORA MI_LATITUD MI_LONGITUD SUBMARINO L...]
  values [join]
  call [Poner_Comillas] Cadena
  call [Clock_Georeferencia] .FormatDateTime
  instant [call [Clock_Georeferencia] .Now]
  pattern [GGMM/yyyy HH:mm]
  call [Poner_Comillas] Cadena [LocationSensor1] .Latitude
  call [Poner_Comillas] Cadena [LocationSensor1] .Longitude
  call [Poner_Comillas] Cadena [Sub_latitude] .Text
  call [Poner_Comillas] Cadena [Sub_longitude] .Text

when FusionTablesControl1 GotResult
do
  call [Notifir_georeferencia] .ShowMessageDialog
  message [Georeferencia Registrada]
  title [Gracias]
  buttonText [Aceptar]

  call [WebView1] .GoHome
  call [WebView2] .GoHome

when Visualizar Click
do
  set [WebView1] .Visible to [true]
  set [WebView2] .Visible to [false]
  set [LABEL_MI_POSICION] .BackgroundColor to [red]
  set [LABEL_POSICION_SUBMARINO] .BackgroundColor to [green]

when Visualizar LongClick
do
  set [WebView1] .Visible to [false]
  set [WebView2] .Visible to [true]
  set [LABEL_POSICION_SUBMARINO] .BackgroundColor to [red]
  set [LABEL_MI_POSICION] .BackgroundColor to [green]
  
```

Figura 66 Programación para pantalla Ubicación

Fuente: El Autor

#### 2.5.4 DISEÑO FÍSICO DEL ROBOT SUBMARINO

El diseño 3D del robot submarino se lo realizó en la aplicación freecad donde se dio forma a cada una de las partes que lo componen desde los soportes de las dos tapas del cuerpo del robot hasta el diseño de motores, propelas, ojivas. Entre las características importantes que tenemos en el robot submarino es la hidrodinámica ya que de ella depende mucho el desplazamiento y la velocidad que llega el robot, evitar tener cuerpos extraños en su exterior para reducir el arrastre hidrodinámico.

El cuerpo central del robot submarino es un tubo pvc de seis pulgadas de diámetro con una longitud de 111 centímetros, para la colocación de los recipientes herméticos se realizó un corte longitudinal de 80 centímetros quedando así con una tapa que facilita la colocación y extracción de los recipientes herméticos (Figura 67).



Figura 67 Corte longitudinal del Tubo de 6 pulgadas

Fuente: El Autor

Entre los diseños 3D que se realizaron está la ojiva delantera que es la encargada de separar la columna de agua para conseguir el avance del robot, su forma es como de un domo, la ojiva trasera en cambio no es tan redondeada sino un poco alargada para conseguir que el agua recorra el cuerpo sin provocar turbulencias.

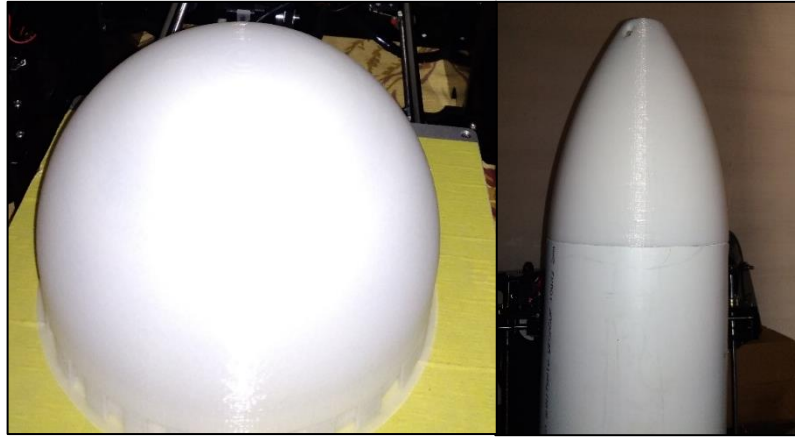


Figura 68 Ojiva Delantera y Ojiva Trasera

Fuente: El Autor

Los recipientes herméticos son uniones y tapones de pvc capaz de resistir hasta 1,6 MPa (mega pascales) lo que se aproximaría a 16 atmosferas de presión, tomando en cuenta que cada 10 metros de profundidad en el mar equivale a una atmosfera de presión los recipientes soportarían más de 100 metros de profundidad (Lanzarote, 2015).

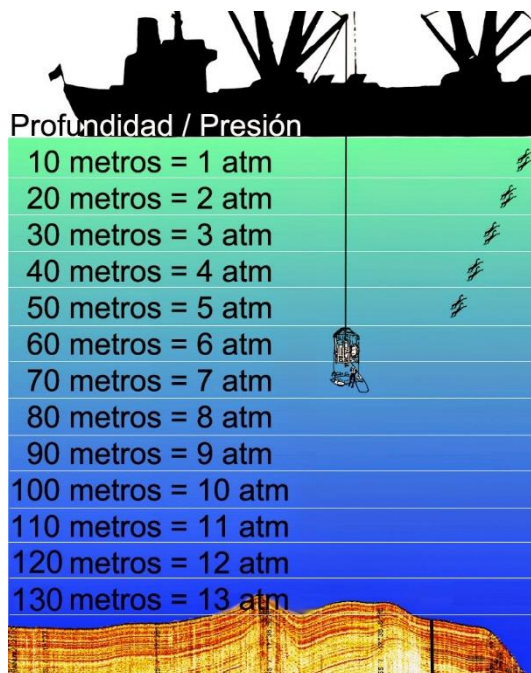


Figura 69 Relación de Profundidad con Presión

Fuente: Lanzarote

Se utilizaron niples y tapones ciegos de 110mm para conexiones de alta presión que se utilizan para sembríos de gran extensión, los tapones sirven para sellar la parte más ancha de recipiente y es por donde salen los cables que se interconectan (Figura 70), se hicieron los agujeros necesarios para que los cables puedan atravesar y posteriormente se sellaron utilizando pegamento epóxico de nombre Pegatanque. Existen cuatro envases, dos para las tarjetas electrónicas y dos para los servomotores y las conexiones necesarias para los elementos que hay en el interior.



Figura 70 Niple con Tapón

Fuente: El Autor

En la parte más angosta del niple será por donde se pueda manipular los elementos electrónicos que estarán dentro, la tapa está diseñada e impresa en 3D, entre el envase y la tapa se coloca un anillo de empaque plano cortado a medida y se asegura con 6 tornillos de 3mm de diámetro como se muestra en la figura 71.



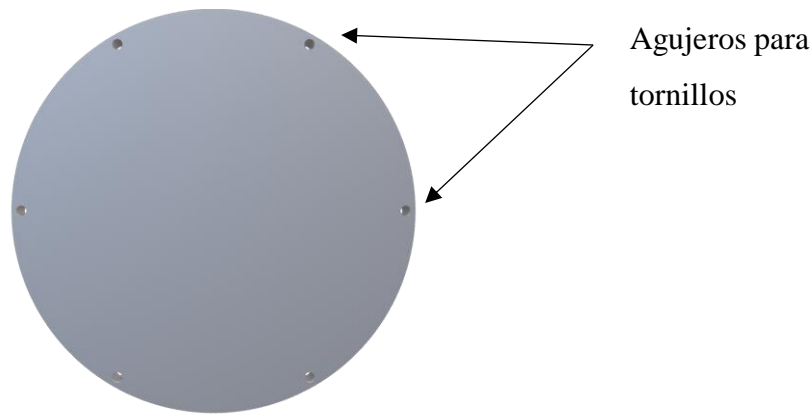


Figura 71 Tapa de Envases Herméticos Diseñada en 3D

Fuente: El Autor

En la Figura 72 se muestra las conexiones necesarias para cada uno de los recipientes que tienen 400mm de largo, un recipiente es el que llevará la electrónica, Arduino mega, sensores reguladores de voltajes y batería de 7,4v 4000 mAh, el otro recipiente de las mismas características almacena la parte de potencia de los motores que conlleva a los controladores ESC y batería de 14,8v 4000 mAh, dos contenedores pequeños de 90mm de largo que albergarán cada uno un servomotor que son los encargados de hacer girar los timones, en la parte superior el robot submarino posee una torre de comunicaciones donde se colocan los sensores de comunicación como es el XBee y GPS.

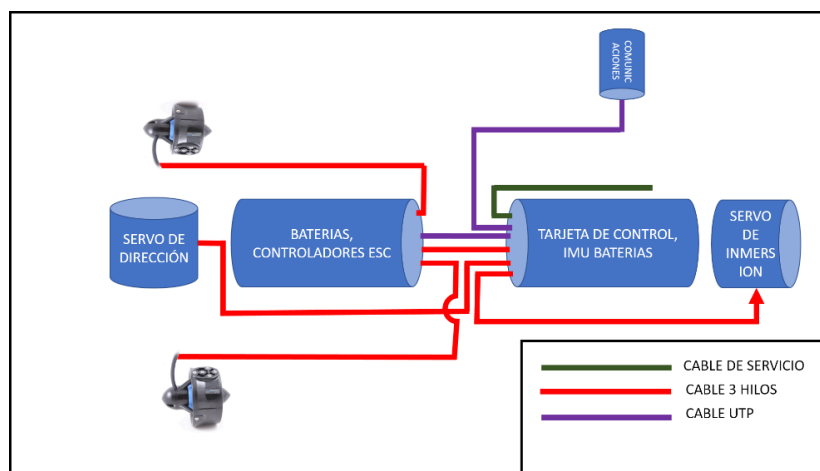


Figura 72 Diseño de conexiones entre elementos

Fuente: El Autor

Los motores brushless A2212/10T no son diseñados para trabajar en agua de mar, se aplica un método de impermeabilización que consiste en cubrir la parte del embobinado con el pegamento epóxico Pegatanque, una vez seco dar un tratamiento de lijado para evitar fricción entre los imanes y el recubrimiento (Figura 73).



Figura 73 Motor Brushless A2212 recubierto con Pegatanque

Fuente: El Autor

Adicional al método, se realizó el diseño e impresión en 3D de los protectores externos del motor como son las toberas y la propela los cuales funcionaban muy bien al hacer las pruebas (Figura 74), como ya mencionamos en apartados anteriores los motores A2212/10T fueron reemplazados por los motores T200 Thruster para crear mayor robustez sobre el prototipo ya que estos son tipos de turbinas dedicados para este tipo de trabajo y arrastre de robot submarino (Figura 75).

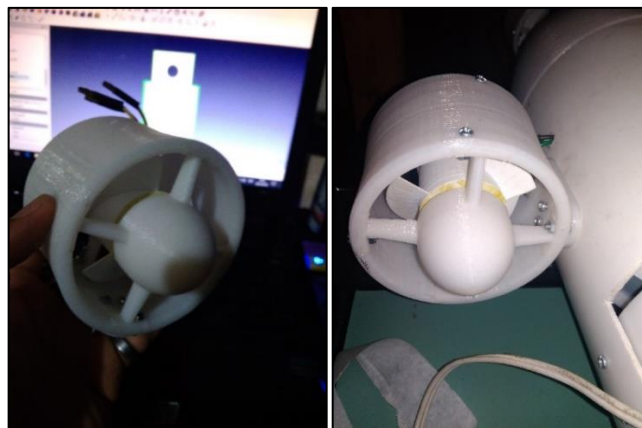


Figura 74 Motores Impresos en 3D

Fuente: El Autor



Figura 75 Motor T200 Thruster y Controlados ECS

Fuente: El Autor

En la Figura 76 se muestra la ubicación del servomotor, para cubrir el recipiente se realizó el diseño e impresión de las dos tapas una en la parte inferior con un orificio donde sale en cable de conexión del servomotor y otra que está ubicada en la parte superior también con un orificio donde sale el eje de 4mm de diámetro unido al servomotor por un acople diseñado e impreso.

Para evitar que el agua ingrese se coloca un retenedor y para mayor protección una vez cerrado el envase se coloca vaselina de manera generosa ya que al ser viscosa y aceitosa se mantiene en su lugar e impide el paso del agua.

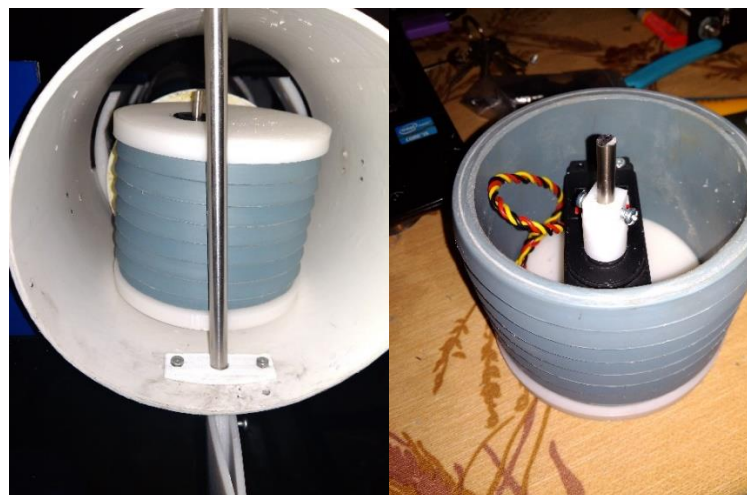


Figura 76 Recipiente para servomotores

Fuente: El Autor

La manera que los servomotores transmiten el movimiento a los timones es por medio de engranajes diseñados e impresos en 3D, para los dos servomotores se realiza en mismo diseño (Figura 77).

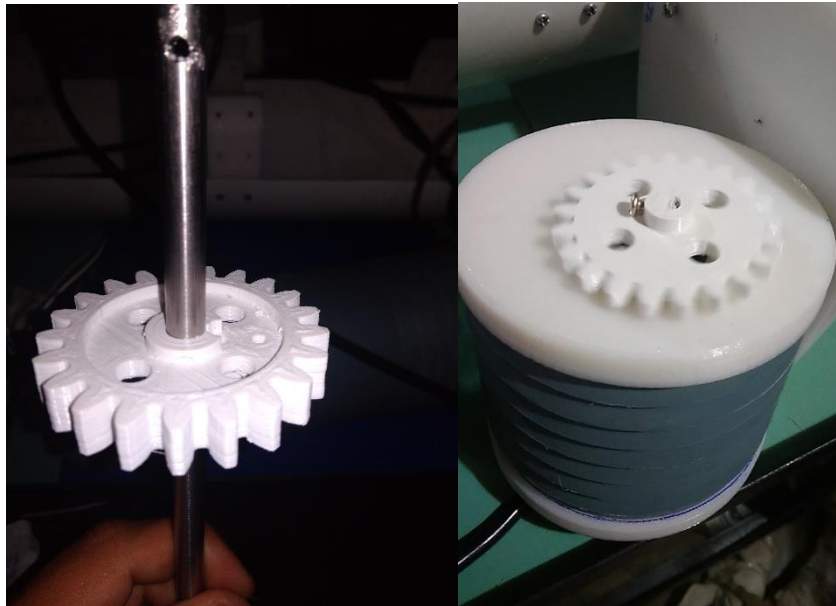


Figura 77 Elementos de Rotación para Timones

Fuente: El Autor

En la figura 78 se muestran la impresión de los timones delanteros y traseros que van a ser dirigidos por servomotores, el timón delantero que hará que el robot se sumerja y emerja este mide 140 mm de largo y el timón de dirección mide 110mm, se colocan dos pares de timones por cada lado y por ellos pasa un eje de 8mm.

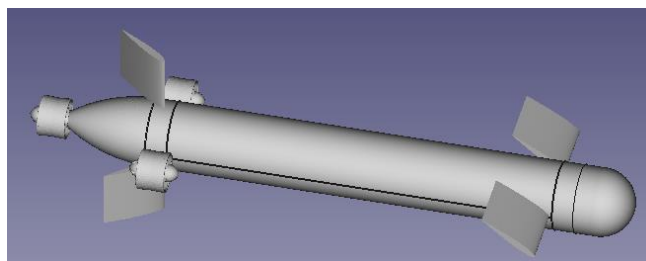


Figura 78 Timones

Fuente: El Autor

Las partes externas del robot se pintaron de color naranja anticorrosivo para que sea visible cuando se encuentre en el mar y el operador lo pueda localizar a mayor distancia (Figura 79).

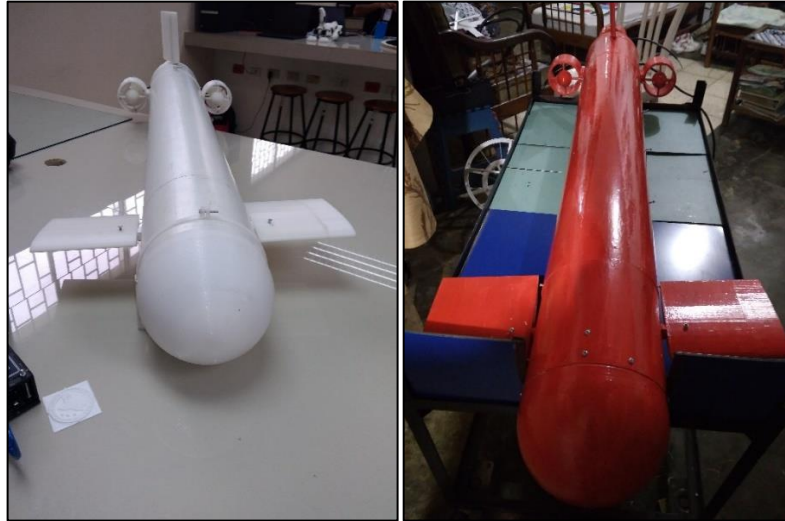


Figura 79 Robot submarino sin pintar y Anaranjado

Fuente: El Autor

Una vez que estuvieron impresas todas las partes del robot se realizó el ensamblaje de la parte electrónica, iniciando con la torre de comunicaciones donde se instalaron en el interior de la capsula impresa en 3D los módulos GPS y XBee, en la parte de superior están colocados los diodos leds color azul para identificar si el robot está comunicándose (Figura 80).

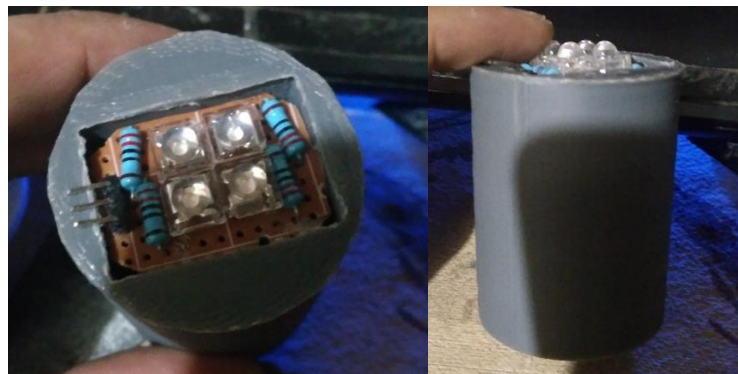


Figura 80 Interior Torre de Comunicación

Fuente: El Autor

Para protección de los motores y los controladores la corriente fue limitada por medio de fusibles de uña de 20 amperios, el circuito se lo realizó en baquelita agujerada y se reforzó las conexiones con cobre de calibre 10 AWG.

La tarjeta de control no presento falla alguna en su diseño y trabaja muy bien, posee sistema para cargar el firmware de manera inalámbrica, un regulador de voltaje con salida de hasta 5 amperios y un conector DB15 para las conexiones de los sensores y actuadores (Figura 81).

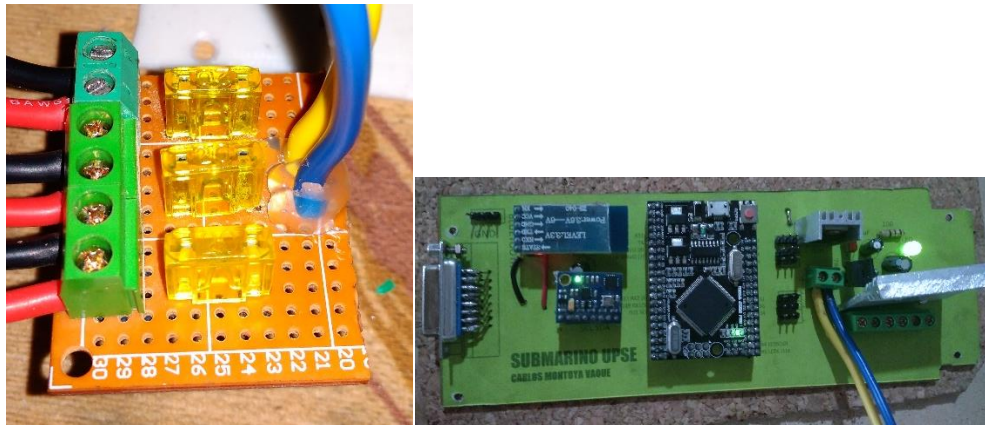


Figura 81 Tarjeta de Fusibles y Tarjeta de control de Robot Submarino

Fuente El Autor



Figura 82 Ensamble de robot Submarino

Fuente: El Autor

### 2.5.5 DISEÑO FÍSICO DEL CONTROL INALÁMBRICO

El control inalámbrico consiste en dos partes que es la base y la tapa diseñadas e impresas en 3D, la base se diseñó de tal manera que haya espacio para ventilación y la tapa con los orificios para colocar el joystick, un potenciómetro lineal, varios indicadores e incluso el soporte para colocar un teléfono móvil (Figura 83).



Figura 83 Control Inalámbrico

Fuente: El Autor

### 2.6 COSTO DEL PROYECTO

DETALLES DEL HARDWARE	CANT.	P/UNIT	TOTAL
HP Pavilion 1000	1	\$ 800,00	\$ 800,00
Impresora 3D Anet A6	1	\$ 480,00	\$ 480,00
Teléfono celular Xiamomi Redmi A4	1	\$ 150,00	\$ 150,00
<b>Total Hardware</b>			<b>\$ 1430,00</b>

Tabla 5 Costos del Hardware Robot Submarino y Control

Fuente: El Autor

<b>DETALLES DEL SOFTWARE</b>	<b>CANT.</b>	<b>P/UNIT</b>	<b>TOTAL</b>
Sistema Operativo Windows 10	1	\$ 220,00	\$ 220,00
Microsoft Office 2013 Profesional	1	\$ 50,00	\$ 50,00
Software PCB Wizard	1	\$ 0,00	\$ 0,00
Software Fritzing	1	\$ 0,00	\$ 0,00
Software Arduino IDE	1	\$ 0,00	\$ 0,00
Software MIT App Inventor2	1	\$ 0,00	\$ 0,00
Sistema Operativo Android	1	\$ 0,00	\$ 0,00
Software FreeCad	1	\$ 0,00	\$ 0,00
Software Cura	1	\$ 0,00	\$ 0,00
<b>Total Software</b>			<b>\$ 270,00</b>

Tabla 6 Costos del Software Robot Submarino y Control

Fuente: El Autor

<b>DETALLES DEL MATERIALES RECIPIENTE HERMÉTICO</b>	<b>CANT.</b>	<b>P/UNIT</b>	<b>TOTAL</b>
Tubo PVC 6" * 3m	1	\$ 20,00	\$ 20,00
Unión Flex 110mm Alta Presión	3	\$ 25,00	\$ 75,00
Tapón 110 mm Alta presión	2	\$ 10,00	\$ 20,00
Pegamento epóxico Pegatanke	8	\$ 6,00	\$ 48,00
Plomo por libra	6	\$ 3,00	\$ 18,00
Filamento ABS rollo	4	\$ 30,00	\$ 120,00
Filamento PLA rollo	2	\$ 30,00	\$ 60,00
<b>Total Materiales para Herméticidad</b>			<b>\$ 361,00</b>

Tabla 7 Gastos Material Recipientes Herméticos, Casco Y Control

Fuente: El Autor



<b>MÓDULOS Y ACTUADORES PARA ROBOT SUBMARINO y CONTROL</b>	<b>CANT.</b>	<b>P/UNIT</b>	<b>TOTAL</b>
Arduino Mega Mini Pro	2	\$ 20,00	\$ 20,00
Servomotor DS3218	2	\$ 40,00	\$ 80,00
Motor T200 Thruster	2	\$ 210,00	\$ 420,00
Sensor GY89	1	\$ 18,00	\$ 18,00
GPS Neo 6m	2	\$ 18,00	\$ 36,00
Módulo XBee	2	\$ 30,00	\$ 60,00
Módulo Bluetooth	2	\$ 8,00	\$ 16,00
Controlador ESC	2	\$ 35,00	\$ 70,00
Batería 7,4v 4000 mAh 30C-40C	1	\$ 60,00	\$ 60,00
Batería 7,4v 2200 mAh 30C-40C	1	\$ 30,00	\$ 30,00
Batería 14,8v 4000 mAh 30C-40C	1	\$ 96,00	\$ 96,00
Joystick	1	\$ 10,00	\$ 10,00
<b>Total Módulos electrónicos y actuadores</b>			<b>\$ 916,00</b>

Tabla 8 Costos de Módulos y Actuadores para Robot Submarino y Control

Fuente: El Autor

<b>ELEMENTOS ELECTRÓNICOS PARA ROBOT SUBMARINO y CONTROL</b>	<b>CANT.</b>	<b>P/UNIT</b>	<b>TOTAL</b>
TIP 41C	2	\$ 5,00	\$ 10,00
LM 7805	2	\$ 3,00	\$ 6,00
Capacitores	8	\$ 1,40	\$ 1,40
Resistencias	15	\$ 0,10	\$ 1,50
Diodo rectificador 1N4004	2	\$ 1,00	\$ 2,00
Conector DB15 macho y hembra	1	\$ 9,00	\$ 9,00
<b>Total Elementos Electrónicos</b>			<b>\$ 29,90</b>

Tabla 9 Costo Elementos Electrónicos

Fuente: El Autor

<b>VIÁTICOS</b>	<b>CANTIDAD</b>	<b>P/UNIT</b>	<b>TOTAL</b>
Viáticos Ayangue pasaje y alimentación	1	\$ 30,00	\$ 30,00
<b>Total Viáticos</b>			<b>\$ 30,00</b>

Tabla 10 Costos de Viáticos

Fuente: El Autor

<b>DETALLE DE INVERSIÓN</b>	<b>CANTIDAD</b>	<b>P/UNIT</b>	<b>TOTAL</b>
Total del software	1	\$ 270,00	\$ 270,00
Total del hardware	1	\$ 1430,00	\$ 1430,00
Total materiales para hermeticidad	1	\$ 361,00	\$ 361,00
Total módulos electrónicos y actuadores	1	\$ 916,00	\$ 916,00
Total elementos electrónicos	1	\$ 29,90	\$ 29,90
Total viáticos	1	\$ 30,00	\$ 30,00
<b>Total Inversión</b>			<b>\$ 3036,90</b>

Tabla 11 Total de inversión

Fuente: El Autor

## 2.7 PRUEBAS

### Experimentación 1.

Para comprobar el porcentaje de aciertos de la ubicación georreferencial se realizó la siguiente prueba, la comparación de datos entre el sensor GPS NEO 6M y la lectura del GPS integrado en el teléfono celular (Figura 84), esta comparación es válida cuando los datos son iguales de lo contrario es un desacierto.

<b>Prueba</b>	<b>Coordenada GPS Robot Submarino</b>	<b>Coordenada GPS Aplicación Móvil Diseñada</b>	<b>VÁLIDO</b>
1	Latitud: -2,225629 Longitud: -80,873309	Latitud: -2,225629 Longitud: -80,873309	SI
2	Latitud: -2,226704 Longitud: -80,872680	Latitud: -2,226704 Longitud: -80,872680	SI
3	Latitud: -2,226687 Longitud: -80,871917	Latitud: -2,226687 Longitud: -80,871917	SI
4	Latitud: -2,226516 Longitud: -80,872605	Latitud: -2,226516 Longitud: -80,872605	SI
5	Latitud: -2,226830 Longitud: -80,871809	Latitud: -2,226825 Longitud: -80,871869	NO
6	Latitud: -2,226511 Longitud: -80,871859	Latitud: -2,226511 Longitud: -80,871859	SI
7	Latitud: -2,226665 Longitud: -80,871367	Latitud: -2,226665 Longitud: -80,871367	SI
8	Latitud: -2,226650 Longitud: -80,870869	Latitud: -2,226650 Longitud: -80,870869	SI
9	Latitud: -2,226667 Longitud: -80,870193	Latitud: -2,226667 Longitud: -80,870193	SI
10	Latitud: -2,226716 Longitud: -80,868967	Latitud: -2,226716 Longitud: -80,868967	SI
11	Latitud: -2,226793 Longitud: -80,867392	Latitud: -2,226793 Longitud: -80,867392	SI
12	Latitud: -2,226758 Longitud: -80,866307	Latitud: -2,226708 Longitud: -80,866397	NO
13	Latitud: -2,226995 Longitud: -80,864709	Latitud: -2,226995 Longitud: -80,864709	SI
14	Latitud: -2,226469 Longitud: -80,864584	Latitud: -2,226469 Longitud: -80,864584	SI
15	Latitud: -2,226469 Longitud: -80,863875	Latitud: -2,226471 Longitud: -80,863877	NO
16	Latitud: -2,226511 Longitud: -80,862249	Latitud: -2,226511 Longitud: -80,862249	SI
17	Latitud: -2,226517 Longitud: -80,861653	Latitud: -2,226517 Longitud: -80,861653	SI
18	Latitud: -2,226492 Longitud: -80,859179	Latitud: -2,226492 Longitud: -80,859179	SI
19	Latitud: -2,226490 Longitud: -80,858060	Latitud: -2,226490 Longitud: -80,858060	SI
20	Latitud: -2,224581 Longitud: -80,857796	Latitud: -2,224581 Longitud: -80,857796	SI
21	Latitud: -2,224138 Longitud: -80,857759	Latitud: -2,224138 Longitud: -80,857759	SI
22	Latitud: -2,223838 Longitud: -80,858112	Latitud: -2,223838 Longitud: -80,858112	SI
23	Latitud: -2,222974 Longitud: -80,859662	Latitud: -2,222974 Longitud: -80,859662	SI

Tabla 12 Comparación coordenadas GPS Submarino Parte 1

Fuente: El Autor

<b>Prueba</b>	<b>Coordenada GPS Robot Submarino</b>	<b>Coordenada GPS Aplicación Móvil Diseñada</b>	<b>VÁLIDO</b>
24	Latitud: -2,221428 Longitud: -80,862144	Latitud: -2,221428 Longitud: -80,862144	SI
25	Latitud: -2,217640 Longitud: -80,864748	Latitud: -2,217640 Longitud: -80,864748	SI
26	Latitud: -2,217180 Longitud: -80,864051	Latitud: -2,217180 Longitud: -80,864051	SI
27	Latitud: -2,216209 Longitud: -80,862697	Latitud: -2,216209 Longitud: -80,862697	SI
28	Latitud: -2,215568 Longitud: -80,861762	Latitud: -2,215568 Longitud: -80,861762	SI
29	Latitud: -2,215456 Longitud: -80,866284	Latitud: -2,215456 Longitud: -80,866284	SI
30	Latitud: -2,214034 Longitud: -80,867312	Latitud: -2,214034 Longitud: -80,867312	SI

Tabla 13 Comparación coordenadas GPS Submarino Parte 2

Fuente: El Autor

<b>Prueba</b>	<b>Coordenada GPS Control Inalámbrico</b>	<b>Coordenada GPS Aplicación Móvil Diseñada</b>	<b>VÁLIDO</b>
1	Latitud: -2,225628 Longitud: -80,873309	Latitud: -2,225629 Longitud: -80,873309	NO
2	Latitud: -2,226704 Longitud: -80,872680	Latitud: -2,226704 Longitud: -80,872680	SI
3	Latitud: -2,226687 Longitud: -80,871917	Latitud: -2,226687 Longitud: -80,871917	SI
4	Latitud: -2,226516 Longitud: -80,872605	Latitud: -2,226516 Longitud: -80,872605	SI
5	Latitud: -2,226825 Longitud: -80,871869	Latitud: -2,226825 Longitud: -80,871869	SI
6	Latitud: -2,226511 Longitud: -80,871859	Latitud: -2,226511 Longitud: -80,871859	SI
7	Latitud: -2,226605 Longitud: -80,871366	Latitud: -2,226665 Longitud: -80,871367	NO
8	Latitud: -2,226650 Longitud: -80,870869	Latitud: -2,226650 Longitud: -80,870869	SI
9	Latitud: -2,226667 Longitud: -80,870193	Latitud: -2,226667 Longitud: -80,870193	SI
10	Latitud: -2,226716 Longitud: -80,868967	Latitud: -2,226716 Longitud: -80,868967	SI
11	Latitud: -2,226793 Longitud: -80,867392	Latitud: -2,226793 Longitud: -80,867392	SI
12	Latitud: -2,226708 Longitud: -80,866397	Latitud: -2,226708 Longitud: -80,866397	SI

Tabla 14 Comparación coordenadas GPS Submarino Parte (1)

Fuente: EL Autor

<b>Prueba</b>	<b>Coordenada GPS Control Inalámbrico</b>	<b>Coordenada GPS Aplicación Móvil Diseñada</b>	<b>VÁLIDO</b>
13	Latitud: -2,226995 Longitud: -80,864709	Latitud: -2,226995 Longitud: -80,864709	SI
14	Latitud: -2,226469 Longitud: -80,864585	Latitud: -2,226469 Longitud: -80,864584	SI
15	Latitud: -2,226469 Longitud: -80,863875	Latitud: -2,226471 Longitud: -80,863877	NO
16	Latitud: -2,226511 Longitud: -80,862249	Latitud: -2,226511 Longitud: -80,862249	SI
17	Latitud: -2,226517 Longitud: -80,861653	Latitud: -2,226517 Longitud: -80,861653	SI
18	Latitud: -2,226492 Longitud: -80,859179	Latitud: -2,226492 Longitud: -80,859179	SI
19	Latitud: -2,226490 Longitud: -80,858060	Latitud: -2,226490 Longitud: -80,858060	SI
20	Latitud: -2,224581 Longitud: -80,857796	Latitud: -2,224581 Longitud: -80,857796	SI
21	Latitud: -2,224138 Longitud: -80,857759	Latitud: -2,224138 Longitud: -80,857759	SI
22	Latitud: -2,223838 Longitud: -80,858112	Latitud: -2,223838 Longitud: -80,858112	SI
23	Latitud: -2,222974 Longitud: -80,859662	Latitud: -2,222974 Longitud: -80,859662	SI
24	Latitud: -2,221428 Longitud: -80,862144	Latitud: -2,221428 Longitud: -80,862144	SI
25	Latitud: -2,217640 Longitud: -80,864748	Latitud: -2,217640 Longitud: -80,864748	SI
26	Latitud: -2,217180 Longitud: -80,864051	Latitud: -2,217180 Longitud: -80,864051	SI
27	Latitud: -2,216209 Longitud: -80,862697	Latitud: -2,216209 Longitud: -80,862697	SI
28	Latitud: -2,215568 Longitud: -80,861762	Latitud: -2,215568 Longitud: -80,861762	SI
29	Latitud: -2,215456 Longitud: -80,866284	Latitud: -2,215456 Longitud: -80,866284	SI
30	Latitud: -2,214034 Longitud: -80,867312	Latitud: -2,214034 Longitud: -80,867312	SI

Tabla 15 Comparación coordenadas GPS Submarino Parte (2)

Fuente: EL Autor

En las Tabla 12, Table 13, Tabla 14 y Tabla 15 se verificaron los valores recibidos por los dispositivos GPS que luego se ubicaron en el robot submarino y control inalámbrico, se obtuvieron 30 muestras tomadas en diferentes sectores del cantón Santa Elena, en cada caso se verificaron los valores recibidos con los mostrados en

la aplicación desarrollada para el proyecto, donde también se verifica el posicionamiento georreferencial con el sensor GPS del dispositivo móvil.

Para calcular el porcentaje de eficiencia de los dispositivos hay que dividir para cantidad de aciertos con la cantidad de pruebas realizadas

$$\%Eficiencia = \frac{Cantidad\ de\ aciertos}{Cantidad\ de\ muestras} * 100\% \quad [14]$$

$$\%Eficiencia = \frac{27}{30} * 100\% \quad [15]$$

$$\%Eficiencia = 90\% \quad [16]$$

## **Experimentación 2.**

Para la orientación del robot se utiliza el sensor LSM303 que se encuentra en el módulo GY-89, antes de usar el sensor se debe de calibrar, la librería “LSM303.h” contiene tres ejemplos, el ejemplo de nombre “Calibrate” sirve para calibrar el sensor, la calibración consiste en cargar el sketch al Arduino conectado con el sensor, se debe de abrir el Monitor Serial y observar los valores que se presentan que son cantidades máximos y mínimos respecto a cada uno de los ejes ‘x’, ‘y’, ‘z’, para conseguir la calibración hay que rotar el sensor en cada uno de los ejes como si rotara en una esfera, cuando se realiza la rotación los valores presentes en el Monitor Serial variarán y llegará un momento que se hagan constantes lo que nos indica que ya se ha calibrado.

Los valores obtenidos en la calibración como mínimo son  $x = -2438$ ,  $y = -1785$ ,  $z = -1102$  y máximo  $x = +928$ ,  $y = +1242$ ,  $z = +1391$ , estos valores se sobrescriben en la programación del robot submarino en la parte del “void setup ()” como `compass.m_min = (LSM303::vector<int16_t>){-2438, -1785, -1102};` y `compass.m_max = (LSM303::vector<int16_t>){+928, +1242, +1391}`, pese a que el sensor ha sido calibrado se realizó la respectiva comparación con datos de la brújula digital del dispositivo móvil donde notamos ligera diferencia mostrada en la Tabla 16 teniendo en consideración una tolerancia de  $\pm 3^\circ$ .

<b>PUNTO CARDINAL</b>	<b>Brújula Digital Teléfono Móvil</b>	<b>Brújula Digital LSM303</b>	<b>Desfase</b>	<b>Acierto</b>
NORTE	0°	3°	3°	No
ESTE	90°	92°	2°	Si
SUR	180°	182°	2°	Si
OESTE	270°	268°	-2°	SI

Tabla 16 Diferencia de Grados entre Sensor y Teléfono Móvil

Fuente: El Autor

Uno de los dispositivos bluetooth se configuró para comunicar de manera inalámbrica al Arduino Mega y así cargar el firmware, el otro dispositivo no se le hizo cambio a su firmware, pero se conectó al Arduino Mega para verificar el funcionamiento enviando y recibiendo datos inalámbricos a las aplicaciones creadas con anterioridad.

Los módulos XBee una vez configurados en el software XCTU, se le realizó las pruebas correspondientes a comunicación inalámbrica a gran distancia cumpliendo con lo especificado en la hoja de datos llegando a tener comunicación hasta los 200 metros aproximadamente.

### **Experimentación 3**

El sensor de temperatura LM35 se hermetizó para que los pines de conexión no tengan contacto con el agua, una vez realizado este trabajo se hicieron las pruebas necesarias y la comparación de la lectura entre el sensor LM35 y un termómetro digital, se colocaron en un recipiente con agua y se comenzó a calentarla, en primera instancia el cambio de temperatura que mostraban cada dispositivo era progresiva,

la condición de acierto entre las medidas del sensor lm35 y el termómetro digital no debe excederse de  $\pm 1^\circ$  (Tabla 17).

Prueba	Sensor LM35	Termómetro Digital	DEFASE	VÁLIDO
1	23°	23°	0	SI
2	24°	24°	0	SI
3	25°	25°	0	SI
4	26°	26°	0	SI
5	27°	27°	0	SI
6	28°	28°	0	SI
7	29°	30°	1	NO
8	30°	30°	0	SI
9	33°	33°	0	SI
10	35°	35°	0	SI
11	38°	38°	0	SI
12	40°	40°	0	SI
13	45°	45°	0	SI
14	50°	50°	0	SI
15	56°	55°	-1	NO

Tabla 17 Comparación Sensor LM35 con Termómetro Digital

Fuente: El Autor

$$\%Eficiencia = \frac{Cantidad\ de\ aciertos}{Cantidad\ de\ muestras} * 100\% \quad [17]$$

$$\%Eficiencia = \frac{13}{15} * 100\% \quad [18]$$

$$\%Eficiencia = 86,67\% \quad [19]$$

Una vez ensamblado los actuadores a el robot submarino se comprobó su movilidad dando diferentes posiciones a los servomotores. Los motores y servomotores se activan mediante el uso de modulación por ancho de pulso y para verificar los cambios de la modulación por ancho de pulso en relación con los grados del movimiento del servomotor se utilizó un osciloscopio portátil MARCA OWON HDS1022M-N. el grado de acierto entre la señal PWM enviada al Servomotor y la señal PWM monitoreada en el dispositivo es de  $\pm 50$  microsegundos. El valor PWM



es enviado mediante el mando inalámbrico y recibido en la etapa de control del robot submarino generando la señal PWM necesaria para el control de los motores y servomotores.

### PROPULSORES

Velocidad	Microsegundos Requeridos	Microsegundos Recibidos	DESFASE	Válido	Referencia
0	1500	1500	0	SI	Figura 86
Media	1600	1600	0	SI	Figura 87
Completa	1900	1900	0	SI	Figura 88

Tabla 18 Cuadro de Velocidad Respecto a Ancho de Pulso

Fuente: El Autor



Figura 84 Motores a velocidad 0

Fuente: El Autor

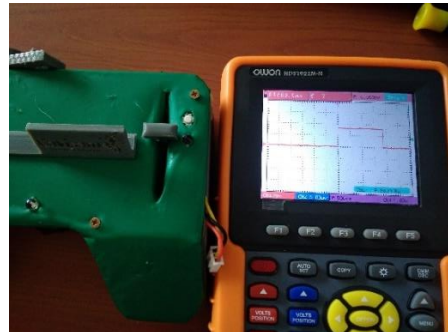


Figura 85 Motores a velocidad media

Fuente: El Autor



Figura 86 Motores a velocidad completa

Fuente: El Autor

## SERVOMOTOR DIRECCIONAL

Posición Grados	Microsegundos Requeridos	Microsegundos Recibidos	Desfase	Válido	Referencia
Izquierda 0°	500	500	0	SI	Figura 89
Centro 90°	1500	1500	0	SI	Figura 90
Derecha 180°	2400	2400	0	SI	Figura 91

Tabla 19 Cuadro Cambio Posición Dirección Respecto a Ancho de Pulso

Fuente: El Autor



Figura 87 Servomotor direccional a 0°

Fuente: El Autor



Figura 88 Servomotor direccional a 90°

Fuente: El Autor



Figura 89 Servomotor direccional a 180°

Fuente: El Autor

### SERVOMOTOR INMERSIÓN-EMERSIÓN

Posición Grados	Microsegundos Requeridos	Microsegundos Recibidos	Desfase	Válido	Referencia
Abajo 0°	900	900	0	SI	Figura 92
Centro 90°	1200	1200	0	SI	Figura 93
Arriba 180°	1500	1500	0	SI	Figura 94

Tabla 20 Cuadro Cambio Posición Inmersión Respecto a Ancho de Pulso

Fuente: El Autor



Figura 90 Servomotor inmersión a 0°

Fuente: El Autor



Figura 91 Servomotor inmersión a 90°

Fuente: El Autor



Figura 92 Servomotor inmersión a 180°

Fuente: El Autor

Las pruebas en ambiente acuático se las realizaron en la piscina de la Universidad Estatal Península de Santa Elena y en la ensenada de la Comuna Ayangue.

Las pruebas realizadas en la piscina de la universidad consistieron en conseguir la “Boyanza Neutral” que consiste en que el robot submarino debe de flotar dejando al descubierto solo una parte de su cuerpo y por sobre todo los motores estar sumergido o cebado para que pueda propulsar al robot, en esta parte experimental se decide la cantidad de lastre o flotadores que se utilizan. Al robot submarino se le dotó de flotadores con la forma de semiluna para que pueda encajar encima de los

recipientes herméticos, una vez colocado el robot en la piscina se notó que flotaba mucho, esto debido a su compuesto principal que es plástico y se decidió colocar plomo para conseguir elevar el peso del equipo, no se quitaron los flotadores porque se desea concentrar el peso en la parte baja del robot y así bajar su centro de masa consiguiendo tener un péndulo invertido el cual reducirá la posibilidad que el submarino sea volteado por alguna corriente marina.



Figura 93 Robot Submarino Boyanza Neutral En Piscina UPSE

Fuente: El Autor

Con esta experimentación se obtuvo el peso del lastre distribuido en el interior del robot submarino para obtener una boyanza neutral, las características generales del robot submarino se encuentran en el anexo 5.

Descripción	Peso	Referencia
Robot Submarino	13,270 Kg	Figura 97
Lastre	1,365 Kg	Figura 98
<b>Total</b>	<b>14,635 Kg</b>	

Tabla 21 Cuadro de pesos del Robot Submarino

Fuente: El Autor



Figura 94 Peso Lastre Del Robot

Fuente: El Autor



Figura 95 Peso Del Robot sin Lastre

Fuente: El Autor

Las pruebas en mar abierto que se realizaron en la ensenada de la Comuna Ayangué consistieron en controlar el desplazamiento del robot submarino con el mando inalámbrico mientras se guardaban las coordenadas de ubicación de los dos equipos verificando que no haya problemas con la comunicación, una vez dada la orden para que empiece la rutina automática el robot comenzó a girar y a buscar el norte según la programación, el siguiente paso fue desplazarse hacia delante usando su timón de dirección para no perder el rumbo por un lapso de 30 segundos, paso seguido comenzó a girar en busca del este, al orientarse avanzó de la misma manera hacia al frente por el tiempo estimado de 30 segundos, los pasos hacia el sur y oeste fueron similares, una vez terminada la rutina el robot se detuvo y siguió enviando la información de la posición



Figura 96 Puntos GPS tomados durante el recorrido

Fuente: El Autor

En la figura 96 se muestran los puntos geográficos tomados en cada giro que dio el robot submarino mientras se desplazaba por la superficie, en la gráfica notamos que el punto inicial y final no coinciden por asunto de la programación en modo automático que maneja una ventana de 20°, entre el punto inicial y final queda una distancia de 4,41m (Tabla 23),

<b>Posición Inicial</b>	<b>Posición Final</b>	<b>Distancia de punto a punto</b>
Latitud -1,980906 Longitud -80,754952	Latitud -1,980589 Longitud -80,754938	34,38 m
Latitud -1,980589 Longitud -80,754938	Latitud -1.980555 Longitud -80.754607	37,01 m
Latitud -1,980555 Longitud -80,754607	Latitud -1.980892 Longitud -80.754573	37,26 m
Latitud -1,980892 Longitud -80,754573	Latitud -1.980914 Longitud -80.754918	38,43 m
Distancia recorrida		147,08 m

Tabla 22 Distancia recorrida en forma Automática

Fuente: El Autor

<b>DISTANCIA APROXIMADA ENTRE PUNTOS</b>		
<b>PUNTO INICIAL</b>	<b>PUNTO FINAL</b>	<b>DISTANCIA</b>
Latitud -1,980914 Longitud -80,754918	Latitud -1.980906 Longitud -80.754952	4,41 m

Tabla 23 Distancia entre Punto Inicial y Final

Fuente: El Autor

El robot de forma autónoma ha recorrido un total de 147,08 metros según los datos recogidos con el sensor GPS (Tabla 22), en cada desplazamiento de 30 segundos se calculó el valor de la velocidad tomando como referencia la distancia recorrida, notando una gran similitud en cada desplazamiento, en términos marinos la velocidad está dada en nudos que equivale a 0,5144 m/s con lo que se pudo calcular el valor medio de velocidad (Tabla 24), el experimento se lo realiza para obtener la velocidad de desplazamiento del robot.

Posición Inicial	Posición Final	Velocidad Aparente	Nudos
Latitud -1,980906 Longitud -80,754952	Latitud -1,980589 Longitud -80,754938	1,15 m/s	2,23
Latitud -1,980589 Longitud -80,754938	Latitud -1.980555 Longitud -80.754607	1,23 m/s	2,39
Latitud -1,980555 Longitud -80,754607	Latitud -1.980892 Longitud -80.754573	1,24 m/s	2,41
Latitud -1,980892 Longitud -80,754573	Latitud -1.980914 Longitud -80.754918	1,28 m/s	2,48
<b>Velocidad Media</b>		<b>1,225 m/s</b>	<b>2,37</b>

Tabla 24 Velocidad Aparente entre puntos tomados

Fuente: EL Autor

En el recorrido que realizó el robot submarino recopiló la información del sensor magnetómetro dando como resultado la información expuesta en la Tabla 25, una vez sumergido el robot la lectura que almacenó fu la de 33,5°.

Posición Inicial	Temperatura
Latitud -1,980906 Longitud -80,754952	33,5
Latitud -1,980589 Longitud -80,754938	34,1
Latitud -1,980555 Longitud -80,754607	33,8
Latitud -1,980892 Longitud -80,754573	33,6

Tabla 25 Datos de Temperatura recibidos

Fuente: El Autor



Figura 97 Robot Submarino Desplazándose

Fuente: El Autor



## 2.8 RESULTADOS

En cada caso se encontró que la señal mostrada en el osciloscopio coincidía con los valores enviados, en el caso del controlador de velocidad se verificó que el dato mínimo y máximo sean los indicados en la programación y la variación que se le da es progresiva a medida que se realizaba el desplazamiento de la palanca de velocidad. Con estas pruebas de laboratorio se puede aseverar el correcto funcionamiento en el envío de la señal hacia los actuadores.

El diseño físico del robot submarino con estructura tipo torpedo necesitó de formas hidrodinámicas para su correcto desplazamiento, formas que se pudieron dar por el diseño en 3D que se realizó, el material que fueron filamento PLA y ABS soportaron el ambiente salino y no presentaron algún tipo de deformidad por esfuerzo mecánico o por simple exposición.

La parte de la electrónica resultó ser un pilar fundamental al momento de controlar el robot, los motores T200 no provocaron ruido que altere la lectura de los sensores, la comunicación inalámbrica tuvo respuestas rápidas al dar las ordenes de desplazamiento en el modo manual.

En las tablas 11 y 12 de la experimentación uno se muestra que las diferencias entre los desaciertos radicaban en las milésimas, para cada sensor existieron 3 desaciertos mientras que el resto de los datos tuvieron una completa similitud con lo que podemos aseverar que en base a las pruebas realizadas tenemos un porcentaje de eficiencia del 90% en cada dispositivo GPS NEO 6M

En la tabla 13 de la experimentación dos se muestran los valores de medidas en grados a comparar incluyendo la columna que muestra el desfase en cada comparación, debido a la tolerancia de  $\pm 3$  grados notamos que existieron 3 de 4 aciertos, en el firmware del robot submarino para la orientación se permite usar una ventana de  $20^\circ$  para lo que el desfase de 2 a 3 grados existentes no son causal para

provocar mayores errores por que se concluye que los sensores están dentro del rango permitido en el firmware.

En la experimentación tres respecto a la verificación del sensor de temperatura se verificó en base a las lecturas tomadas por un sensor de temperatura digital dando como resultado que existe un 86,67% de eficiencia.

En el caso de los actuadores como son los motores T200 y los servomotores DS3218 se encontró que la señal mostrada en el osciloscopio coincidía con los valores enviados, en el caso del controlador de velocidad se verificó que el dato mínimo y máximo sean los indicados en la programación y la variación que se le da es progresiva a medida que se realizaba el desplazamiento de la palanca de velocidad. Con estas pruebas de laboratorio se puede aseverar el correcto funcionamiento en el envío de la señal hacia los actuadores.

## CONCLUSIONES

Entre los elementos escogidos para la propuesta tenemos el GPS NEO 6M que nos da la posición geográfica del robot cuando se encuentre en la superficie, una vez sumergido el robot el dispositivo GPS no tendrá conexión con satélites. El método utilizado para el desplazamiento en modo autónomo es el recorrido deducido, en el algoritmo se utiliza máquinas de estado para programar los pasos a seguir, el desplazamiento lo hará en base a un tiempo establecido y para las pruebas fueron de 30 segundos demostrando un buen desempeño y realizando la recolección del dato de temperatura.

Para el empuje del robot se utilizan los motores T200 Thruster, estos motores son usados netamente para trabajos de inmersión teniendo un gran rendimiento, cada motor posee un empuje de cinco kilogramos fuerza siendo capaces de mover al robot sin la necesidad de consumir el máximo de la corriente especificada por el fabricante que es de 25 Amperios, se utilizan 2 motores teniendo un empuje total de 10 kgf, el robot en la superficie tiene un peso de 14,6 kg y una vez en el agua el peso baja a menos de la mitad por lo que los motores no tienen problema al momento de realizar el empuje.

El diseño electrónico del sistema de regulación de voltaje posee la capacidad de entregar hasta cinco amperios de corriente garantizando el buen desempeño de los servomotores, la electrónica de control consumirá un total de 3800 mAh, y la parte de potencia 40 Ah máximo. Los motores tendrán una autonomía máxima de 10 horas teniendo en cuenta que la batería tiene 14.1 voltios, 4000mAh con 25C de descarga continua. Con la implementación de la etapa de carga inalámbrica del firmware hacia la tarjeta de control del submarino se exime el uso de medios físicos (cables), esto garantiza una reprogramación inalámbrica y la optimización de tiempo al realizar modificaciones en el firmware.

La aplicación desarrollada para el proyecto tiene la cualidad de recibir información de posición del control y la del robot submarino, almacenarla en la nube de Google

drive para presentarla en mapas en el apartado denominado “Ubicación” que se encuentra dentro de la misma aplicación móvil, también en ella podremos ver el valor de distancia entre el submarino y el mando inalámbrico.

Los materiales utilizados para la parte física del robot son de PVC que se adquieren en el mercado local, fueron complementados con piezas diseñadas e impresas en 3D y se diseñaron de tal manera que la estructura sea hidrodinámicamente estable, en las pruebas realizadas se demostró que no existe arrastre hidrodinámico ya que con poco impulso el robot logra realizar un desplazamiento efectivo llegando a tener buena velocidad de desplazamiento.

## **RECOMENDACIONES**

En el diseño físico el Robot Submarino evitar todo elemento que provoque arrastre hidrodinámico porque con esa característica el prototipo perderá empuje y se estaría desperdiciando recursos.

Mediante una abrazadera sujetar los cables que salen de los recipientes herméticos para que la posible manipulación no provoque fisuras en la base de los recipientes.

Usar un domo transparente en reemplazo de la ojiva frontal para colocar una cámara tipo ojo de pez para capturar imágenes submarinas y aparte de la recolección de los datos oceanográficos también se puedan tomar muestras de agua para análisis en laboratorio.

Los resultados del desplazamiento del robot en la ensenada de la playa de Ayangué se notó poco error en las distancias de punto a punto, esto debido al movimiento del mar que de alguna manera ayudaban a adelantar o retrasar el avance, para minimizar el error se recomienda usar sistemas de medición inercial.

## BIBLIOGRAFÍA

- Sabitha A. Wadoo, Pushkin Kachroo. (2011). *Autonomous Underwater Vehicles*. New York: CRC Press.
- Carlos Carvalho Witte, D. Z. (Julio de 2011). *Rio Datacentro*. Obtenido de [http://meggi.usuarios.rdc.puc-rio.br/teses/TFC11\\_Daniel\\_Freitas\\_e\\_Carlos\\_Witte.pdf](http://meggi.usuarios.rdc.puc-rio.br/teses/TFC11_Daniel_Freitas_e_Carlos_Witte.pdf)
- Cruz, N. A. (Octubre de 2011). *Autonomous Undersea Vehicles Applications Center*. Obtenido de [http://auvac.org/uploads/publication\\_pdf/Autonomous\\_Underwater\\_Vehicles%20Intech.pdf](http://auvac.org/uploads/publication_pdf/Autonomous_Underwater_Vehicles%20Intech.pdf)
- Arduino. (2018). *Arduino*. Obtenido de <https://store.arduino.cc/usa/arduino-mega-2560-rev3>
- Banzi, M. (2014). *Make: Getting Started with Arduino*. Maker Media.
- Canadian, S. S. (2019). *Ropos*. Obtenido de <https://www.ropos.com/index.php/ropos-rov>
- Carrillo, P. (2014). *Diseño y Construcción Del Submarino para Exploración ORCA I*. Quito.
- Cuellar, B. (2004). *Control de inmersión para plataforma robótica submarina*. Bogotá.
- De La Rosa, C. (2018). Obtenido de <http://www.institutomarconi.net/descargas/TUTORIAL%20PCB%20WIZARD.pdf>
- FPP. (2015). Obtenido de <http://repositorio.puce.edu.ec/bitstream/handle/22000/13113/Anexo%20%20MIT%20App%20Inventor%202.pdf?sequence=4&isAllowed=y>
- García, C. U. (1999). *Grupo Isis*. Obtenido de <http://www.grupoisis.uma.es/microbot/public/robots.pdf>
- IMPRESORAS3D. (2018). Obtenido de <http://impresorad3d.com/?s=ULTIMAKER+2%2B>
- IMPRESORAS3D.com. (2019). *Impresoras 3D*. Obtenido de <http://impresorad3d.com/anet-a6-analisis/>

INOCAR. (Octubre de 2013). *Instituto Oceanográfico de la Armada INOCAR*.  
 Obtenido de <https://www.inocar.mil.ec/web/index.php/comunicamos/350-cinco-vehiculos-submarinos-autonomos-monitorean-el-oceano-en-las-islas-galapagos>

International Submarine Engineering Ltd. (2001). *Autonomous Undersea Vehicle Applications Center*. Obtenido de <http://auvac.org/platforms/view/163>

Jaycar. (2018). Obtenido de <https://www.jaycar.com.au/db15-female-connector-solder/p/PS0824>

jcrespom. (2018). Obtenido de <https://aprendiendoarduino.wordpress.com/tag/xctu/>

Lanzarote. (25 de 3 de 2015). Obtenido de <https://lanzarotenonstopdivers.blogspot.com/2015/03/presion-y-profundidad.html>

Live, N. (12 de 01 de 2019). Obtenido de <https://nautiluslive.org/tech/rov-hercules>

Maritime, K. (13 de 01 de 2019). Obtenido de <https://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/B3F87A63D8E419E5C1256A68004E946C?OpenDocument>

Moreno, H. A. (2013). Obtenido de [http://oa.upm.es/16634/1/04\\_HECTOR\\_ALONSO\\_MORENO\\_AVALOS.pdf](http://oa.upm.es/16634/1/04_HECTOR_ALONSO_MORENO_AVALOS.pdf)

Nicolas Soria, D. C. (23 de Marzo de 2014). *Universidad San Francisco de Quito*. Obtenido de [revistas.usfq.edu.ec/index.php/avances/article/download/166/168](http://revistas.usfq.edu.ec/index.php/avances/article/download/166/168)

Oyarce, A. (2008). Obtenido de [https://www.mcielectronics.cl/website\\_MCI/static/documents/XBee\\_Guia\\_Usuario.pdf](https://www.mcielectronics.cl/website_MCI/static/documents/XBee_Guia_Usuario.pdf)

Robotics, B. (2019). *Blue Robotics*. Obtenido de <https://www.bluerobotics.com/store/thrusters/t200-thruster-blueesc/>

Salamanca, B. (2012). *Diseño y manufactura de un prototipo de robot submarino autonomo con control de estabilidad de inmersión*. México DF. Obtenido de <https://tesis.ipn.mx/bitstream/handle/123456789/18569/Dise%C3%B1o%20y%20manufactura%20de%20un%20prototipo%20de%20robot%20submarino%20autonomo%20con%20control%20de%20estabilidad%20de%20inmersi%C3%B3n>

- 0y%20manufactura%20de%20un%20prototipo%20de%20robot%20subm  
arino%20aut%C3%B3nomo%20con%20control%20de%20estabilidad%2  
0de%20inmersi%C3%B3n.pdf?sequence=1&isAllowed=y
- STMicroelectronics. (2012). Obtenido de  
<https://www.pololu.com/file/0J703/LSM303D.pdf>
- Straigh-Line Distance, C. (s.f). Obtenido de  
<https://engineering.purdue.edu/ece477/Archive/2008/Spring/S08-Grp03/nb/datasheets/haversine.pdf>
- Toloka, P. (2019). Obtenido de  
<http://www.profetolocka.com.ar/2014/12/12/completo-tutorial-de-diseno-3d-con-freecad/>
- Ublox. (2019). Obtenido de [https://www.u-blox.com/sites/default/files/products/documents/NEO-6\\_DataSheet\\_%28GPS.G6-HW-09005%29.pdf?utm\\_source=en%2Fimagenes%2Fdownloads%2FProduct\\_Docs%2FNEO-6\\_DataSheet\\_%28GPS.G6-HW-09005%29.pdf](https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf?utm_source=en%2Fimagenes%2Fdownloads%2FProduct_Docs%2FNEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf)
- UVAC. (2019). *AUVAC* . Obtenido de <http://auvac.org/configurations/view/2>
- Waveshare. (2019). Obtenido de  
[https://www.waveshare.com/wiki/UART\\_GPS\\_NEO-6M\\_\(B\)](https://www.waveshare.com/wiki/UART_GPS_NEO-6M_(B))
- Zambrano, L. (2009). *Inocar*. Obtenido de  
[https://www.inocar.mil.ec/web/phocadownloadpap/actas\\_oceanograficas/acta15/OCE1501\\_16.pdf](https://www.inocar.mil.ec/web/phocadownloadpap/actas_oceanograficas/acta15/OCE1501_16.pdf)
- Zilic, F. S. (2015). *Researchgate*. Obtenido de  
[https://www.researchgate.net/publication/282866527\\_Desarrollo\\_de\\_Vehiculo\\_Submarino\\_Autonomo\\_AUV\\_para\\_Exploracion\\_Oceanografica\\_DID\\_NS-2014-34](https://www.researchgate.net/publication/282866527_Desarrollo_de_Vehiculo_Submarino_Autonomo_AUV_para_Exploracion_Oceanografica_DID_NS-2014-34)



# **ANEXOS**

Anexo 1 Diseños 3D



Figura 98 Perfil 3D Timones

Fuente: El Autor

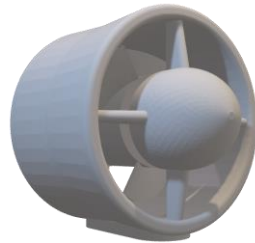


Figura 99 Toberas para Motor 3D

Fuente: El Autor

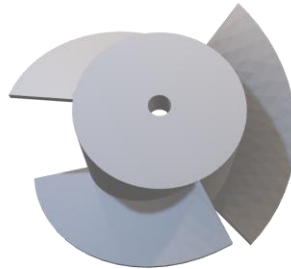


Figura 100 Propela Diseño 3D

Fuente: El Autor

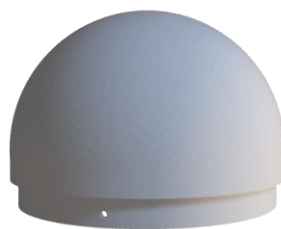


Figura 101 Ojiva Delantera Diseño 3D

Fuente: El Autor



Figura 102 Ojiva Posterior

Fuente: El Autor



Figura 103 Abrazadera Recipiente Hermético Diseño 3D

Fuente: El Autor



Figura 104 Soporte Tarjeta control

Fuente: El Autor

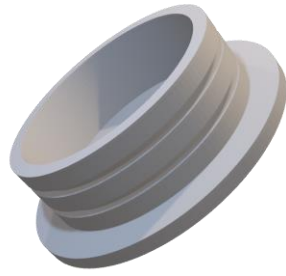


Figura 105 Tapa Recipientes Herméticos

Fuente: El Autor

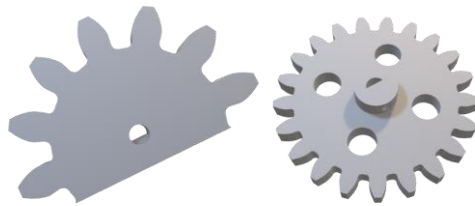


Figura 106 Engranajes Servos Diseño 3D

Fuente: El Autor

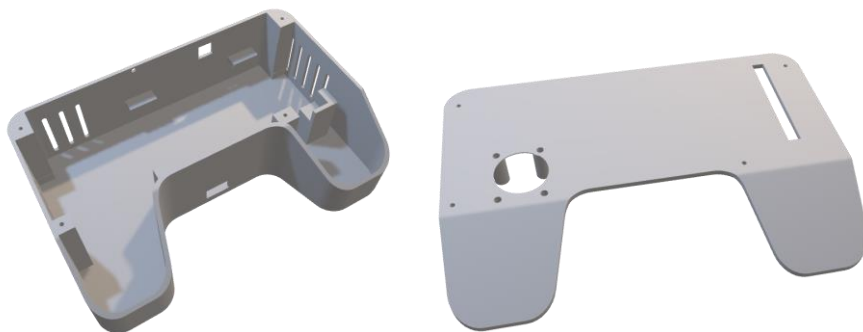


Figura 107 Control Inalámbrico Diseño 3D

Fuente: El Autor

Anexo 2 Fotos Varias de pruebas realizadas en la Universidad



Figura 108 Fotos pruebas de flotabilidad en piscina UPSE

Fuente: El Autor

## Anexo 3 Código del Firmware para el Robot Submarino

```

#include <TinyGPS.h>
#include <Wire.h>
#include <LSM303.h>
#include <L3G.h>
#include <Servo.h>
#include <EEPROM.h>

Servo ESC1;
Servo ESC2;
Servo ejez;
Servo ejax;
//-----Motores-----
int vel = 1500; //amplitud del pulso
//-----Temperatura-----
float tempC, tempC1;
int pinLM35 = A0;
//-----GPS SUBMARINO-----
TinyGPS gps; //Declaramos el objeto gps
float lat_sub, lon_sub;
int lat, lon, lat_en, lon_en;
int lat_dec, lon_dec;
int recepcion;
//----DATOS PARA CONTROL REMOTO----
String str = "";
const char separator = ',';
const int dataLength = 5;
float data[dataLength];
char buf[10];
int verificacion;
int verificacion_pasada = 0;

int yaw;
int pitch;
int z = 1195;

int y = 1195;
//-----Magnetometro-----
LSM303 compass;
float heading, mag, mag1;
char report[80];
//-----Giroscopio-----
L3G gyro;
char buffer[70];
unsigned long interval = 500;
unsigned long previousMillis = 0;
//-----AUTOMATICO-----int
autonomo, auto1;
unsigned long tic, toc;
int estado = 1;

void setup() {
  Serial1.begin(9600); //xbee
  Serial2.begin(9600); // gps
  Serial.begin(9600);
  Wire.begin();
  //Programacion para el magnetómetro
  compass.init();
  compass.enableDefault();
  compass.m_min = (LSM303::vector<int16_t>) {
    -32767, -32767, -32767
  };
  compass.m_max = (LSM303::vector<int16_t>) {
    +32767, +32767, +32767
  };
  //-----Giroscopio-----
  //----
  if (!gyro.init())
  {

```

```

    Serial.println("Failed to autodetect gyro
type!");
    while (1);
}
gyro.enableDefault();
}
//-----Declaracion de pines-----
pinMode(2, OUTPUT);
pinMode(6, OUTPUT);
pinMode(13, OUTPUT);

//-----Motores-----
ESC1.attach(2); //2 6
ESC2.attach(6); //2 6
ESC1.writeMicroseconds(1500); //1500 = 1.5ms
ESC2.writeMicroseconds(1500); //1500 = 1.5ms
//-----
ejey.attach(3);
ejez.attach(5);

ejey.writeMicroseconds(1195); //1195 =
1.195ms
ejez.writeMicroseconds(1195); //1195 =
1.195ms
delay(500);
Serial.setTimeout(10);
pinMode(44, OUTPUT);
pinMode(46, OUTPUT);
}

//-----Recepcion de datos para
locomocion Manual-----//
void datos_locomocion() {
    while (Serial1.available()){
        str = Serial1.readStringUntil('\n');
        for (int i = 0; i < dataLength ; i++){
            int index = str.indexOf(separator);
            data[i] = str.substring(0, index).toInt();
            str = str.substring(index + 1);
            digitalWrite(32, HIGH);
        }
        verificacion = data[0];
        velocidad = data[1];
        yaw = data[2];
        pitch = data[3];
        autonomo = data[4];
    }
}

void motores(int velocidad) {
    vel = map(velocidad, 0, 1023, 1500, 1900);
    ESC1.writeMicroseconds(vel);
    ESC2.writeMicroseconds(vel);
}

void direccional(int eje_x) {
    if ((eje_x >= 0) && (eje_x <= 800)) x = -
0.47187500 * eje_x + 1572.5;
    if ((eje_x > 800) && (eje_x <= 1023)) x = -
1.55829596 * eje_x + 2441.636771;
    ejex.writeMicroseconds(x);
}

void cabeceo(int eje_z) {
    if ((eje_z >= 0) && (eje_z <= 760)) z =
0.45723684 * eje_z + 847.5;
    if ((eje_z > 760) && (eje_z <= 1023)) z =
1.43536122 * eje_z + 104.1254753;
    ejez.writeMicroseconds(z);
}

void desplazamiento(int a, int b, int c) {
    ESC1.writeMicroseconds(a);
    ESC2.writeMicroseconds(a);
    ejez.writeMicroseconds(b);
    ejex.writeMicroseconds(c);
}

//-----Recepcion datos GPS del
submarino-----//
void gps_submarino() {
    recepcion = 0;
    while (Serial2.available())

```

```

{
  int c = Serial2.read();
  if (gps.encode(c))
  {
    recepcion = 1;
    gps.f_get_position(&lat_sub, &lon_sub);
  }
}

//-----Funcion para recibir datos del
magnetometro-----//

void magnetometro() {
  compass.read();
  heading = compass.heading();
}

//-----Temperatura-----

void temperatura() {
  tempC = 0;
  for (int e = 0; e < 20; e++)
  {
    tempC = tempC + analogRead(pinLM35);
  }
  tempC1 = ((5.0 * (tempC / 20) * 100.0) /
1023.0) - 16;}

void envio_datos() {
  unsigned long currentMillis = millis();
  if ((unsigned long)(currentMillis -
previousMillis) >= interval) {
    Serial.print(lat_sub * 1000000);
    Serial.print(",");
    Serial.print(lon_sub * 1000000);
    Serial.print(",");
    Serial.print(tempC1);
    Serial.print(",");
    Serial.println(heading * 100);

    Serial1.print(lat_sub * 1000000);
    Serial1.print(", ");
    Serial1.print(lon_sub * 1000000);
    Serial1.print(", ");

    Serial1.print(tempC1);
    Serial1.print(", ");
    Serial1.println(heading * 100);
  }
}

//-----Funcion loop interaccion de
los elementos-----//

void loop() {
  datos_locomocion();
  if (auto1 == 0) {
    if (verificacion != verificacion_pasada) {
      Serial.println("MANUAL");
      parpadeo_manual();
      gps_submarino(); //Llamamos funcion
recepcion datos GPS Submarino
      magnetometro();
      motores(velocidad);
      direccional(pitch);
      cabeceo(yaw);
      temperatura();
      envio_datos();
      estado = 1;
    } else {
      Serial.println("CONTROL APAGADO");
      digitalWrite(46, HIGH);
      digitalWrite(44, HIGH);
      desplazamiento(1500, 1195, 1195);
      magnetometro();
      temperatura();
      gps_submarino(); //Llamamos funcion
recepcion datos GPS Submarino
      envio_datos();
      estado = 1;
    }
  }
}
else {

```



```

Serial.println("AUTOMATICO");
gps_submarino();
parpadeo_auto();
envio_datos();
magnetometro();
EEPROM.write(4, tempC1);

switch (estado) {
  case 0:
    desplazamiento(1500, 1195,
1195);//velocidad, aleta inmersión, aleta giro
    break;

  case 1: tic = toc = millis();
  while (toc - tic < 10000) {
    toc = millis();
    parpadeo_auto();
    envio_datos();
    gps_submarino();
    magnetometro();
    sumergir();
    datos_locomocion();
    if (autonomo == 0) {
      estado = 0;
      break;
    }
  }
  estado = 2;
  break;

  case 2: tic = toc = millis();
  while (toc - tic < 30000) {
    toc = millis();
    parpadeo_auto();
    envio_datos();
    gps_submarino();
    magnetometro();
    norte();

    datos_locomocion();
    if (autonomo == 0) {
      estado = 0;
      break;
    }
  }
  estado = 3;
  break;

  case 3: tic = toc = millis();
  while (toc - tic < 30000) {
    toc = millis();
    parpadeo_auto();
    envio_datos();
    gps_submarino();
    magnetometro();
    este();
    datos_locomocion();
    if (autonomo == 0) {
      estado = 0;
      break;
    }
  }
  estado = 4;
  break;

  case 4: tic = toc = millis();
  while (toc - tic < 30000) {
    toc = millis();
    parpadeo_auto();
    envio_datos();
    gps_submarino();
    magnetometro();
    sur();
    datos_locomocion();
    if (autonomo == 0) {
      estado = 0;
      break;
    }
  }
}

```

```

    }
}
estado = 5;
break;
case 5: tic = toc = millis();
while (toc - tic < 30000) {
    toc = millis();
    parpadeo_auto();
    envio_datos();
    gps_submarino();
    magnetometro();
    oeste();
    datos_locomocion();
    EEPROM.write(5, tempC1);
    if (autonomo == 0) {
        estado = 0;
        break;
    }
}
estado = 6;
break;
case 6: tic = toc = millis();
while (toc - tic < 10000) {
    toc = millis();
    parpadeo_auto();
    gps_submarino();
    envio_datos();
    magnetometro();
    emerger();
    datos_locomocion();
    if (autonomo == 0) {
        estado = 0;
        break;
    }
}
estado = 0;
break;
}
}
EEPROM.write(0, verificacion);
verificacion_pasada = EEPROM.read(0);
if ((autonomo <= 1))
    EEPROM.write(1, autonomo);
auto1 = EEPROM.read(1);
}
void parpadeo_manual() {
    int w = millis() >> 9 & 1;
    digitalWrite(46, w);
    digitalWrite(44, !w);
}
void parpadeo_auto() {
    int x = millis() >> 7 & 1;
    digitalWrite(46, x);
    digitalWrite(44, !x);
}
////////////////////AUTOMATICO////////////////////////////////////
////////////////////
void norte() {
    // recto
    if ((heading >= 0) && (heading <= 10)) {
        desplazamiento(1800, 1195, 1195);
    }
    if ((heading >= 350) && (heading <= 360)) {
        desplazamiento(1800, 1195, 1195);
    }
    //giro derecha
    if ((heading > 180) && (heading <= 350)) {
        desplazamiento(1800, 1195, 847.5);
    }
    //giro izquierda
    if ((heading > 10) && (heading <= 180)) {
        desplazamiento(1800, 1195, 1572.5);
    }
}
}

```

```

}
void este() {
    //giro derecha
    if ((heading >= 0) && (heading < 80)) {
        desplazamiento(1800, 1195, 847.5);
    }
    // recto
    if ((heading >= 80) && (heading <= 100)) {
        desplazamiento(1800, 1195, 1195);
    }
    //giro izquierda
    if ((heading > 100) && (heading <= 270)) {
        desplazamiento(1800, 1195, 1572.5);
    }
    //giro derecha
    if ((heading > 270) && (heading <= 360)) {
        desplazamiento(1800, 1195, 847.5);
    }
}
void sur() {
    //giro derecha
    if ((heading >= 0) && (heading < 170)) {
        desplazamiento(1800, 1195, 847.5);
    }
    // recto
    if ((heading >= 170) && (heading <= 190)) {
        desplazamiento(1800, 1195, 1572.5);
    }
    //giro izquierda
    if ((heading > 190) && (heading <= 360)) {
        desplazamiento(1800, 1195, 1572.5);
    }
}
void oeste() {
    //giro izquierda
    if ((heading > 280) && (heading <= 360)) {
        desplazamiento(1800, 1195, 1572.5);
    }
    // recto
    if ((heading >= 260) && (heading <= 280)) {
        desplazamiento(1800, 1195, 1572.5);
    }
    //giro izquierda
    if ((heading >= 0) && (heading <= 90)) {
        desplazamiento(1800, 1195, 1572.5);
    }
    //giro derecha
    if ((heading > 90) && (heading < 260)) {
        desplazamiento(1800, 1195, 847.5);
    }
}
void sumergir() {
    desplazamiento(1800, 1572.5, 847.5);
}
void emerger() {
    desplazamiento(1800, 847.5, 1572.5);
}

```

## Anexo 4 Código del Firmware para el Mando Inalámbrico

```

#include <TinyGPS.h> //incluimos TinyGPS
Libreria para GPS
#include <EEPROM.h>
//GPS A BORDO
TinyGPS gps;//Declaramos el objeto gps
float lat1, lon1;

// Variables control manual TIPO ROV (Vehiculo
Operado Remotamente)
int velocidad;
int yaw; //rotacion eje z giros a los lados
int pitch; // rotacion eje x inclinacion arriba abajo
del ROV
int i;

String str = "";
const char separator = ',';
const int dataLength = 5;
float data[dataLength];
char buf[10];
double lat_sub;
double lon_sub;
int recepcion, a;
float magnetometro;
int temperatura;

unsigned long interval = 500; // the time we need
to wait
unsigned long previousMillis = 0; // millis()
returns an unsigned long.

String referencia, sentido;
int amplitud;
float g,d;
//-----PULSADOR-----
int direccion=0;
int estadoAnterior = 0;
int estado = 0;
int automatico = 0;
int val;

void setup(void) {
Serial.begin(9600);
Serial1.begin(9600); // 18, 19 GPS
Serial2.begin(9600); // 16, 17 xbee
Serial3.begin(9600); // 14, 15 BT
pinMode(25, OUTPUT); // MANUAL
AUTOMATICO COLOR ROJO
pinMode(27, OUTPUT); //GPS A BORDO
derecho
pinMode(29, OUTPUT); //GPS RECIBIDO
}

void pulsador() {
int estado = digitalRead(32);

if ((estado == HIGH) && (estadoAnterior ==
LOW) && (analogRead(A1) == 0)) //detecta un
flanco de subida
{ automatico = 1 - val;
EEPROM.write(direccion,automatico);
delay(20);
}
estadoAnterior = estado;
val=EEPROM.read(direccion);
//Serial.println(val);
if (val == 1) digitalWrite(25, HIGH);
else digitalWrite(25, LOW);
}

// FUNCIÓN RECOLECTAR DATOS GPS
void gps_a_bordo(void) {
digitalWrite(27, LOW);
while (Serial1.available() > 0)
{
int c = Serial1.read();
if (gps.encode(c))
{
digitalWrite(27, HIGH);
gps.f_get_position(&lat1, &lon1);
}
}
}

```

```

}
//-----Envio datos para control
manual-----//
void control_manual() {
    velocidad = analogRead(A0);
    yaw = analogRead(A3); //giros en x ARRIBA
    ABAJO
    pitch = analogRead(A1); // giros en z DERECHA
    IZQUIERDA
    for (i = 0 ; i <= 9 ; i++) {
        char buffer[70];
        sprintf(buffer, "%d,%d,%d,%d,%d", i,
        velocidad, yaw, pitch, val);
        Serial.println(buffer);
        Serial2.println(buffer);
    }//delay(100);
}

//-----Datos GPS recibidos del
submarino-----//
void gps_recibido(void) {
    a = 0;
    digitalWrite(29, LOW);
    while (Serial2.available())
    {
        a = 1;
        if (a == 1) {
            digitalWrite(29, HIGH);
            str = Serial2.readStringUntil('\n');

            for (int i = 0; i < dataLength ; i++)
            {
                int index = str.indexOf(separator);
                data[i] = str.substring(0, index) .toInt();
                str = str.substring(index + 1);
                //repcion = int(data[0]);
                lat_sub = (data[0]/1000000);
                lon_sub = (data[1]/1000000); //dividimos por
                1000000 para convertir el entero en decimales, y
                sumamos 300 para separarlos por rango
                temperatura = data[2];
                magnetometro = data[3];
            }
        }
    }
}

}
}

void loop() {
    // digitalWrite(27, LOW);
    // digitalWrite(29, LOW);
    int b = 0;
    control_manual();
    gps_a_bordo();
    gps_recibido();
    medir_distancia(lat1, lon1, lat_sub, lon_sub);
    pulsador();

    unsigned long currentMillis = millis();
    if ((unsigned long)(currentMillis -
    previousMillis) >= interval) {
        Serial3.print(lat_sub*100);
        Serial3.print(",");
        Serial3.print(lon_sub*100);
        Serial3.print(",");
        Serial3.print(temperatura);
        Serial3.print(",");
        Serial3.print(magnetometro / 100);
        Serial3.print(",");
        Serial3.print(d); //distancia
        Serial3.print(",");
        Serial3.print(g); //azimutal
        Serial3.print(",");
        Serial3.print(referencia) ;
        Serial3.print(amplitud) ;
        Serial3.println(sentido) ;
        previousMillis = millis();
    }
}

void medir_distancia(float latitud1, float
longitud1, float latitud2, float longitud2) {
    //latitud2 = latitud2; longitud2 = longitud2;
    //Serial.print(latitud1, 6); Serial.print(",");
    Serial.print(longitud1, 6); Serial.print(",");
    Serial.print(latitud2, 6); Serial.print(",");
    Serial.println(longitud2, 6);
}

```

```

float          pi          =
3.1415926535897932384626433832795;
double dlat = ((latitud2 + latitud1) * pi) / 180;
double dlong = ((longitud2 - longitud1) * pi) /
180;
float y = (((latitud2 - latitud1) * pi) / 180);
float x = dlong * cos( 0.5 * (dlat) );

acos (3);
d = 6371 * sqrt( x * x + y * y)*1000;
//d=d+880;

// Azimut
float alon = longitud2 - longitud1;
float blat = latitud2 - latitud1;
float alpha = atan(alon / blat ) * 180 / pi;
if (alpha < 0)
    alpha = alpha * (-1);

if ((blat > 0) and (alon > 0))
    g = 0 + alpha;
if ((blat > 0) and (alon < 0))//
    g = 360 - alpha;
if ((blat < 0) and (alon < 0))//
    g = 180 + alpha;
if ((blat < 0) and (alon > 0))//
    g = 180 - alpha;

//Rumbo

if ((g >= 0) && (g <= 90)) {
    referencia = "N";
    amplitud = g;
    sentido = "E";
}
if ((g > 90) && (g <= 180)) {
    referencia = "S";
    amplitud = 180 - g;
    sentido = "E";
}
if ((g > 180) && (g <= 270)) {
    referencia = "S";
    amplitud = g - 180;
    sentido = "O";
}
if ((g > 270) && (g <= 360)) {
    referencia = "N";
    amplitud = 360 - g;
    sentido = "O";
}
//g=g+1300;
//Serial.print(blat, DEC); Serial.print(",");
Serial.print(alon, DEC); Serial.print(",");
Serial.println(alpha, DEC);
//Serial.print(d, DEC); Serial.print(",");
}

```

## Anexo 5 Características del Robot Submarino

### **Características**

**Peso en el aire** 32,26 lb (14,635 Kg)

**Peso en Agua de Mar** neutralmente flotante

**Dimensiones** 1,11m x 0,43m x 0,47m

**Profundidad** 50 metros máximo

**Desplazamiento en mar** calmado

## Anexo 6 Informe Urkund

La Libertad, 5 de Febrero del 2019

Señor

Freddy Villao Santos, Mgt

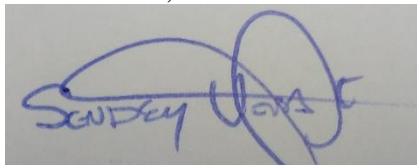
DECANO(e) DE LA FACULTAD DE SISTEMAS Y TELECOMUNICACIONES  
UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA  
Ciudad.

De mis consideraciones:

En mi calidad de Tutor del trabajo de tesis “PROTOTIPO DE ROBOT SUBMARINO AUTÓNOMO TIPO TORPEDO PARA RECOPIACIÓN DE DATOS OCEANOGRÁFICOS.”, realizado por Carlos Vinicio Montoya Vaque, egresado(a) de la Carrera de Electrónica y Telecomunicaciones, Facultad de Sistemas y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, previo a la obtención del Título de Ingeniero en Electrónica y Telecomunicaciones, me permito declarar que luego de haber orientado, estudiado y revisado, lo Apruebo en todas sus partes, además informo que el resultado obtenido en el sistema de coincidencias URKUND es de 6 %.

URKUND	
Dokument	<a href="#">tesis_robot_submarino_Montoya.docx</a> (D47590788)
Inskickat	2019-02-04 15:59 (-05:00)
Inskickad av	VERA GONZALEZ SENDEY AGUSTIN (svera@upse.edu.ec)
Mottagare	vera.upse@analysis.orkund.com
	6% av det här c:a 50 sidor stora dokumentet består av text som också förekommer i 7 st källor.

Atentamente,



Ing. Sendey Vera G, MACI

Tutor de Tesis