



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Diseño de un sistema de control de flotabilidad para un
planeador submarino*

Grado en Ingeniería Mecánica

ALUMNO: Nuttawut Uttaros

DIRECTORES: José Antonio González Prieto

Carlos Pérez Collazo

CURSO ACADÉMICO: 2022-2023

Universida_{de}Vigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Diseño de un sistema de control de flotabilidad para un
planeador submarino*

Grado en Ingeniería Mecánica

Intensificación en Tecnología Naval

Cuerpo General / Infantería de Marina

UniversidadeVigo

RESUMEN

Este trabajo trata sobre el diseño y desarrollo de un sistema de control de flotabilidad para un planeador submarino. El objetivo es crear un sistema que permita al planeador mantener una profundidad deseada mediante el control de la cantidad de aceite que es transferido entre un depósito interno y una vejiga en contacto con el exterior. El diseño incluye la creación de un circuito hidráulico y electrónico, cuya función es controlar el flujo de aceite entre el depósito interior y la vejiga exterior, de forma que la unidad de control se realiza empleando el microcontrolador ESP32 Dev kit V1, lo que permitirá reducir el tamaño del sistema, optimizar el consumo y disponer de conectividad wifi-integrada. Para analizar el comportamiento de la bomba y las válvulas, se realizarán ensayos para obtener las curvas características y los datos de consumo. El soporte del sistema se diseñará de forma cilíndrica utilizando AutoCAD e Inventor. Además, se desarrolla un modelo matemático del sistema de control mediante los métodos PID y MPC. Este modelo se utilizará para realizar simulaciones con Python, lo que permitirá optimizar y ajustar el control del sistema.

En resumen, este trabajo implica el diseño y desarrollo de un sistema de control de flotabilidad para un planeador submarino, que incluye la creación de un circuito hidráulico y electrónico, el control mediante un microcontrolador, ensayos de análisis de componentes, el diseño de un soporte y la simulación del sistema.

PALABRAS CLAVE

Glider, AUV, Planeador submarino, Flotabilidad, Robótica

CONTENIDO

Contenido	1
Índice de Figuras	3
Índice de Tablas.....	7
1 Introducción y objetivos	9
1.1 Introducción	9
1.2 Objetivos	9
2 Estado del arte	11
2.1 Ingeniería 4.0	11
2.2 <i>Internet of Things</i> (IoT).....	12
2.3 Vehículos no tripulados	12
2.4 Planeador Submarino	14
2.5 Sistema de control de flotabilidad.....	16
2.6 Sistema de control de flotabilidad por transferencia de aceite.....	18
2.7 Microcontroladores	19
2.8 Movimiento del vehículo submarino	21
2.9 Investigación submarina	24
3 Materiales y métodos.....	27
3.1 Consideraciones desde modelo matemático	27
3.1.1 Modelo cinético	27
3.1.2 Modelo dinámico	28
3.1.3 Ecuaciones del espacio de estado	29
3.2 Sistema hidráulico.....	30
3.2.1 Factores a considerar	30
3.2.2 Componentes	30
3.2.3 Diseños propuestos	36
3.3 Sistema electrónico, integración de programa autónoma	37
3.3.1 Componentes	37
3.3.2 Diseño propuesto	41
3.4 Método de ensayo	41
3.5 Soporte del sistema	46
3.5.1 Software para el diseño.....	46
3.5.2 Materiales para la impresión 3D.....	47
3.6 Sistema de control.....	48
3.6.1 Controlador PID.....	48

3.6.2 Controlador MPC.....	49
4 Resultados y validación experimental	53
4.1 Resultados de los ensayos	53
4.1.1 Ensayo de la bomba en vacío.....	53
4.1.2 Ensayo de la válvula de 3 vías en vacío	54
4.1.3 Ensayo de la válvula de 2 vías en vacío	54
4.1.4 Curvas características de la bomba con agua.....	55
4.1.5 Curvas características de la bomba con agua por transferencia a una vejiga	56
4.2 Soporte del sistema	56
4.2.1 Módulos principales.....	56
4.2.2 Depósito del aceite.....	56
4.2.3 Conjunto de la placa electrónica y sistema de control.....	57
4.2.4 Soporte de la bomba	58
4.2.5 Soporte de válvulas y antirretornos	58
4.2.6 Conjunto de tuberías y sensores	59
4.2.7 Tapa final	60
4.3 Simulador de movimiento.....	60
4.3.1 Valores iniciales de control	60
4.3.2 Resultados gráficos.....	61
4.4 Discusión.....	66
5 Conclusiones y líneas futuras	69
5.1 Conclusión	69
5.2 líneas futuras	70
6 Bibliografía.....	71
ANEXO I Implicaciones Sociales, Económicas, y Ambientales.....	75
ANEXO II Reflexiones Éticas y Sociales	76
ANEXO III Código de ensayo.....	77
ANEXO IV Código de insertar datos a SQLite (MAIN FILE)	81
ANEXO V Código de insertar datos a SQLite (CLASS FILE)	83
ANEXO VI Código simulación PID	85
ANEXO VII Código simulación MPC (<i>MAIN File</i>)	90
ANEXO VIII Código simulación MPC (<i>Support File</i>).....	95
ANEXO IX Planos	100

ÍNDICE DE FIGURAS

Figura 2-1 Internet of Things [10].....	12
Figura 2-2 Vehículos submarinos [12].....	13
Figura 2-3 ROV modelo <i>Chasing M2</i> [14].	14
Figura 2-4 AUV modelo REMUS 600 en la cubierta del barco de NOAA [15].....	14
Figura 2-5 Uso de planeador submarino de NOAA [17].	15
Figura 2-6 el diagrama de los sistemas de un planeador submarino [18].	16
Figura 2-7 El UUV está enviando los datos a través del satélite de la estación en tierra [19].....	16
Figura 2-8 Sistema de flotabilidad con jeringuillas de <i>OSUG</i> [21].	17
Figura 2-9 Sistema de flotabilidad de <i>SLOCUM</i> [22].....	18
Figura 2-10 El sistema SMA [23].	18
Figura 2-11 Esquema hidráulico tipo <i>Oil Bladder</i> [24].	19
Figura 2-12 Esquema de control básico para el sistema de flotabilidad.	19
Figura 2-13 Arduino MEGA, UNO, NANO y Pro Mini [26].....	20
Figura 2-14 La familia de Raspberry Pi [29].....	20
Figura 2-15 Los modelos de ESP32 [31].	21
Figura 2-16 6DOF en un vehículo submarino [32].	22
Figura 2-17 Ala de perfil NACA [33].	23
Figura 2-18 El principio de Arquímedes en un vehículo submarino [34].....	24
Figura 2-19 Snell law [35].....	24
Figura 2-20 Perfil de la velocidad de sonido según la profundidad [37].	25
Figura 2-21 Rayo directo o Canal de sonoro superficial [38].	25
Figura 2-22 Canal de sonoro [38].....	26
Figura 2-23 Zona de Convergencia [38].	26
Figura 2-24 Rebote en el fondo [38].	26
Figura 3-1 Tanque combustible.....	30
Figura 3-2 Bomba hidráulica sin escobilla.....	31
Figura 3-3 - Sharplace Mini Bomba de Succión Autocebante	32
Figura 3-4 Válvula Solenoide 2 posiciones 3 vías.	32
Figura 3-5 Válvula de 2 vías.	33
Figura 3-6 Antirretorno [46].....	34
Figura 3-7 Membrana para la vejiga.	35
Figura 3-8 Tubería [48].	35
Figura 3-9 Conectores de tuberías [49].	36
Figura 3-10 El circuito del sistema hidráulico.	37

Figura 3-11 ESP32 Deokit V1.....	37
Figura 3-12 Relay 3DC.	38
Figura 3-13 Sensor INA219.	39
Figura 3-14 Transductor de presión, Akozon Sensor de presión de acero inoxidable.....	39
Figura 3-15 Sensor de Caudalímetro de Agua.	40
Figura 3-16 Soporte de expansión de batería.	41
Figura 3-17 Esquema de ensayo con Arduino UNO.	42
Figura 3-18 Ensayo de tres componentes juntos con Arduino UNO.	42
Figura 3-19 Esquema de ensayo con ESP32.	43
Figura 3-20 Ensayo de un componente individual con ESP32 Devkit V1.....	43
Figura 3-21 Esquema de ensayo para la bomba de motor sin escobilla con Arduino UNO.	43
Figura 3-22 Circuito de ensayo con agua.	44
Figura 3-23 Circuito de ensayo con agua por transferencia a una vejiga.....	44
Figura 3-24 Circuito de ensayo con aceite.	45
Figura 3-25 El flujo de trabajo y los servicios usados.	45
Figura 3-26 Hacer un perfil con AutoCAD 2023.....	46
Figura 3-27 Extrusión de un perfil con Inventor profesional 2023.	47
Figura 4-1 Ensayo de la bomba en vacío: la gráfica de intensidad-tiempo.....	53
Figura 4-2 Ensayo de la bomba en vacío: arranque.	54
Figura 4-3 Ensayo de la válvula de 3 vías en vacío.	54
Figura 4-4 Ensayo de la válvula de 2 vías en vacío.	55
Figura 4-5 Curvas características de la bomba con agua.	55
Figura 4-6 Curvas características de la bomba con agua por transferencia a una vejiga.	56
Figura 4-7 Soporte del sistema de flotabilidad con los componentes.	56
Figura 4-8 Depósito del aceite.....	57
Figura 4-9 Conjunto de la placa electrónica y sistema de control.....	57
Figura 4-10 Soporte de la bomba.	58
Figura 4-11 Soporte de válvulas y antirretornos.	59
Figura 4-12 Conjunto de tuberías y sensores.	59
Figura 4-13 Tapa final.....	60
Figura 4-14 Simulación PID: movimiento de bajada-tiempo.....	62
Figura 4-15 Simulación PID: movimiento de bajada-tiempo (zoom).....	62
Figura 4-16 Simulación MPC: movimiento de bajada-tiempo.	63
Figura 4-17 Simulación MPC: movimiento de bajada-tiempo (zoom).....	64
Figura 4-18 Simulación MPC: velocidad y aceleración.....	64
Figura 4-19 Simulación MPC: velocidad y aceleración (zoom).	65

Figura 4-20 Simulación MPC con la minimización de la velocidad: movimiento de bajada-tiempo.

.....66

Figura 4-21 Simulación MPC con la minimización de la velocidad: velocidad y aceleración.66

ÍNDICE DE TABLAS

Tabla 3-1 Tabla de propiedades de filamentos [58].	47
Tabla 3-2 Efecto de los constantes K_p , K_i y K_d en el lazo cerrado [59].	48
Tabla 4-1 Parámetros del planeador submarino simulado.	60
Tabla 4-2 Parámetros del ambiente simulado.	61
Tabla 4-3 Parámetros PID simulado.	61
Tabla 4-4 Parámetros MPC simulado.	63
Tabla 4-5 Parámetros MPC con la minimización de la velocidad simulado.....	65

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Introducción

Un sistema no tripulado es un sistema autónomo que no necesita un operador humano para realizar las misiones que le han sido asignadas [1]. De esta forma, existen sistemas autónomos con una posición fija, como puede ser un brazo robótico empleado en cirugía, o móviles, como por ejemplo los sistemas de ayuda a la conducción avanzada en los vehículos actuales más avanzados.

Dentro de la categoría de sistemas autónomos móviles podemos encontrar diferentes clasificaciones de sistemas en base al entorno operativo en el que deben ejecutar sus misiones. De esta forma podemos tener sistemas que operan en tierra (*Unmanned Ground Vehicle* - UGV), en el aire (*Unmanned Air Vehicle* - UAV), en la superficie del mar (*Unmanned Surface Vehicle* - USV), submarinos (*Unmanned Underwater Vehicle* - UUV), así como una serie de vehículos híbridos que cuya movilidad les permite desplazarse en diferentes entornos.

El trabajo desarrollado en este TFG se enmarca dentro del campo de los UUV, también conocidos como AUV (un acrónimo de *Autonomous Underwater Vehicle*), los cuales son vehículos tripulados sin cable de sujeción que, entre otras aplicaciones, suelen ser empleados como herramientas para la investigación submarina [2] o en el desarrollo de aplicaciones industriales avanzadas.

Un planeador submarino o *glider*, es un vehículo submarino no tripulado que no dispone de sistemas de propulsión activa, de forma que su consumo de energía es muy inferior al de vehículos que disponen de propulsión. Su desplazamiento se consigue mediante cambios en sus propiedades inerciales que le permiten controlar la flotabilidad del dispositivo, así como los cambios de rumbo que le permitan trazar las rutas planeadas en cada misión. La combinación de estos elementos permite crear sistemas autónomos submarinos que generan soluciones con una alta capacidad de inmersión a grandes profundidades, que ofrecen una alta robustez en cuanto a sus prestaciones y que pueden ser desarrollados con unos costes muy inferiores a los sistemas tradicionales de exploración submarina, como los buques de exploración [3].

De esta forma, la flotabilidad y la capacidad del sistema para controlarla, constituyen uno de los principales factores que determinan la operatividad de un *glider* submarino, permitiéndole controlar su profundidad de forma robusta y precisa con un coste energético muy inferior al que ofrecen otros sistemas submarinos no tripulados [4].

1.2 Objetivos

Este trabajo tiene como objetivo principal el estudio y construcción de un sistema de control de flotabilidad empleado en un planeador submarino desarrollado de forma modular, empleando tecnología

de impresión 3D y elementos de electrónica IoT. De esta forma, para lograr cumplir este objetivo, se definen los siguientes hitos:

- Estudio del movimiento 6DOF para planeador submarino: permitirá comprender y modelar el movimiento del planeador en todas las direcciones y en diferentes profundidades.
- Estudio del sistema de control: matemáticas y simulador: permitirá diseñar un simulador del sistema que facilite la evaluación y ajuste del control. Este simulador se utilizará para validar y optimizar el control del sistema de flotabilidad.
- Diseños del sistema de flotabilidad: estará enfocado en la creación de un sistema robusto y confiable. Se llevarán a cabo pruebas y ensayos para analizar el comportamiento de los componentes y garantizar el correcto funcionamiento del sistema en diferentes profundidades.
- Integración la programación para trabajo autónomo: permitirá que el sistema pueda operar de manera autónoma en el agua. Se utilizarán microcontroladores y sistemas de comunicación inalámbrica para lograr una interacción confiable y efectiva entre el sistema y el usuario.

La estructura de este trabajo de fin de grado es la siguiente. En el capítulo 2 se introduce el estado del arte referente a los tipos de submarinos planeadores similares que ya han sido desarrollados. A continuación, en el capítulo 3, se presenta la metodología de trabajo empleada en el diseño del *glider*, incluyendo los cálculos del modelado, métodos de análisis, materiales empleados y diseños del sistema de flotabilidad.

Capítulo 2: Estado del arte

- Se realiza una revisión bibliográfica de submarinos planeadores y de sistemas de control de flotabilidad.
- Se presentan ejemplos de modelos de submarinos planeadores existentes y se analizan las diferencias y similitudes entre ellos.
- Se describe el estado actual de la tecnología en cuanto a sistemas de control de flotabilidad para planeadores submarinos y se destacan las limitaciones y áreas de mejora.

Capítulo 3: Materiales y métodos

- Se describe detalladamente la metodología empleada para el diseño del planeador submarino, incluyendo los cálculos de modelado, métodos de análisis y materiales empleados.
- Se explica la selección de los componentes hidráulicos y electrónicos, incluyendo las bombas, válvulas y sensores.
- Se presentan los diseños del sistema de flotabilidad y se explican las consideraciones de diseño para garantizar un sistema confiable y seguro.
- Se describen las técnicas de ensayo y validación utilizadas para garantizar el correcto funcionamiento del sistema de control de flotabilidad.

Capítulo 4: Resultados validación experimental

- Se realizan pruebas y ensayos para evaluar el comportamiento del sistema de control.
- Se presentan los resultados de las pruebas y se discuten las limitaciones y áreas de mejora del sistema de control.
- Se resumen los resultados y logros obtenidos en el diseño y desarrollo del sistema de control de flotabilidad para el planeador submarino.
- Se discuten las implicaciones y aplicaciones potenciales de este trabajo.

Capítulo 5: Línea futuras

- Se presentan los trabajos futuros y las áreas de mejora para el sistema de control de flotabilidad, incluyendo la optimización del control, la integración de sensores adicionales y la mejora del sistema de comunicación inalámbrica.

2 ESTADO DEL ARTE

2.1 Ingeniería 4.0

La ingeniería 4.0 se basa en la digitalización de los procesos industriales conectando la inteligencia artificial (también denominada ingeniería digital) y la maquinaria [5]. Su objetivo es conseguir la producción industrial con una mayor eficiencia permitiendo, además, que los clientes configuren los productos deseados a partir de modelos intercambiables.

La ingeniería 4.0 tiene 5 pilares fundamentales en su desarrollo [6]:

1. *Big data*: Se refiere al análisis y procesamiento de datos en grandes cantidades, para después ordenarlo y presentarlo en forma sencilla y clara. Estos análisis de los datos pueden emplearse, por ejemplo, para la creación de sistemas automáticos de toma de decisiones.
2. Robótica: El uso de robots permite la automatización de los procesos industriales de forma que el desarrollo de sus capacidades sigue incrementándose con nuevos diseños que permiten integrarlos de forma óptima en las tareas logísticas y de producción.
3. Simulación: Se trata del uso de software que permite realizar análisis del comportamiento de los sistemas mediante la simulación empleando modelos matemáticos. Este tipo de forma de trabajo con un modelo digital permite, entre otras ventajas, realizar un análisis de las propiedades del sistema con un menor coste que con un modelo real, de forma más rápida y permitiendo obtener información de situaciones críticas en las que el modelo real no puede ser testeado. Herramientas de diseño por ordenador como CAD, CAE, CAM, juegan un papel crucial en la creación de estas simulaciones numéricas.
4. *Internet of things*: Conjunto de dispositivos, como sensores o herramientas de control que pueden conectarse al internet, de forma que empleando esta red los dispositivos pueden interactuar intercambiando información.
5. Manufactura aditiva: Tipo de producción que emplea herramientas como la impresión 3D para la producción de productos industriales. Esta forma de trabajo permite reducir las pérdidas de materiales en la construcción, así como realizar la fabricación de piezas a medida, optimizando de esta forma los procesos productivos en tiempo real.

Una de las principales características de la ingeniería 4.0 es la interconexión de los sistemas y dispositivos, lo que permite la recopilación y el análisis de datos en tiempo real. Esto permite una mejor toma de decisiones y una mayor eficiencia en la producción.

La ingeniería 4.0 permite una mayor flexibilidad en la producción, ya que los sistemas se pueden adaptar rápidamente a los cambios en la demanda. También se menciona que la interconexión de los sistemas permite una mayor colaboración entre los diferentes departamentos de una empresa, lo que mejora la eficacia en la toma de decisiones [7].

La ingeniería 4.0 también tiene implicaciones en la formación de los trabajadores, ya que se requiere una mayor habilidad en el manejo de las tecnologías de la información y la comunicación [8].

En resumen, la ingeniería 4.0 combina las tecnologías de la información y la comunicación con la automatización y la robótica para mejorar la eficiencia y la eficacia en la industria. Esto incluye el uso de tecnologías como el IoT, la IA y la automatización avanzada. La interconexión de los sistemas permite la recopilación y el análisis de datos en tiempo real, lo que permite una mejor toma de decisiones y una mayor eficiencia en la producción.

2.2 Internet of Things (IoT)

Internet of Things (IoT) es una red de dispositivos conectados a internet que pueden recolectar y compartir datos. Estos dispositivos incluyen desde dispositivos tradicionales como computadoras y teléfonos inteligentes, hasta objetos cotidianos como electrodomésticos, dispositivos de seguridad, vehículos, y otros. El objetivo de IoT es mejorar la eficiencia y automatizar procesos mediante la recolección y análisis de datos en tiempo real [9].

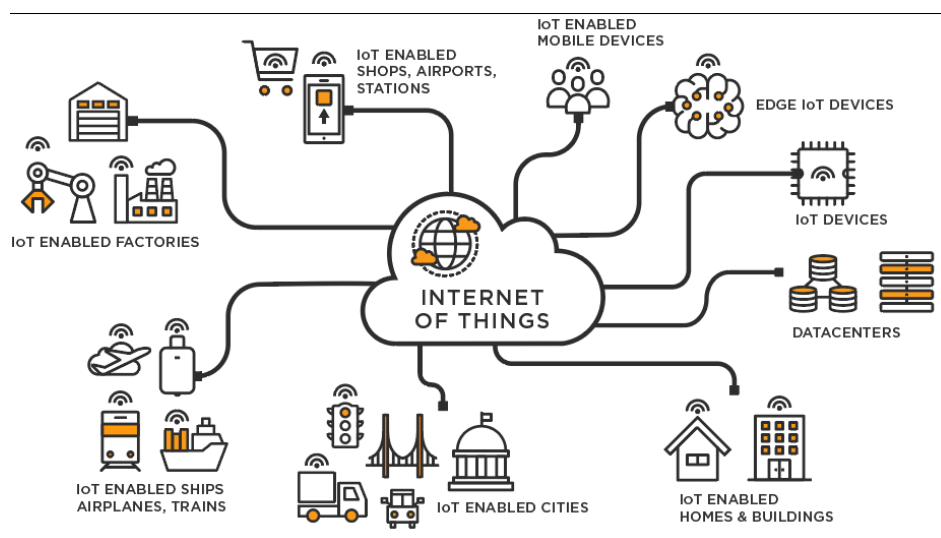


Figura 2-1 Internet of Things [10].

IoT permite la comunicación bidireccional entre dispositivos y sistemas, lo que permite una mayor interconexión y automatización como en Figura 2-1. Por ejemplo, los dispositivos IoT pueden ser utilizados para controlar sistemas de iluminación y climatización en edificios inteligentes, para monitorear el rendimiento de equipos en la industria, o para ayudar a las personas con discapacidad o mayores a vivir de manera independiente [11].

IoT también ofrece un gran potencial para mejorar la eficiencia en la toma de decisiones, ya que permite la recolección y análisis de una gran cantidad de datos en tiempo real. Esto puede ayudar a mejorar la seguridad, la eficiencia energética, la movilidad y la salud.

2.3 Vehículos no tripulados

Un vehículo no tripulado es un vehículo que se mueve sin la presencia física de un ser humano a bordo. También se conocen como drones o UAV (vehículos aéreos no tripulados). Estos vehículos pueden ser controlados de forma remota o programados para funcionar de forma autónoma. Se utilizan en una variedad de aplicaciones, incluyendo la vigilancia, la exploración, la cartografía, la agricultura, la logística, el transporte de carga, la fotografía y el cine, entre otros.

UUV es el acrónimo en inglés de "*Unmanned Underwater Vehicle*" (Vehículo submarino no tripulado) es un vehículo submarino que se mueve sin la presencia física de un ser humano a bordo. Los UUV se utilizan para una amplia variedad de tareas, incluyendo la inspección de submarinos, la

cartografía de fondos marinos, la recuperación de objetos sumergidos, la investigación oceanográfica, la detección de minas y la vigilancia submarina. Algunos UUVs son remotamente controlados, mientras que otros son programados para funcionar de forma autónoma.

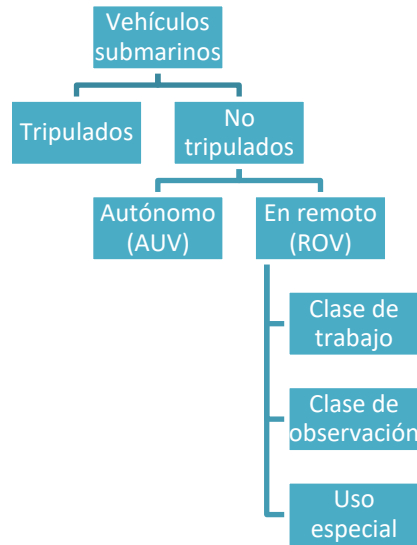


Figura 2-2 Vehículos submarinos [12].

Para los vehículos submarinos no tripulados o UUVs (*Unmanned Underwater Vehicles*) según *UUV Master Plan (2004 edition, section 1.3)* de *US Navy* [13], Se define como.

Self-propelled submersible whose operation is either fully autonomous (preprogrammed or real-time adaptive mission control) or under minimal supervisory control and is untethered except, possibly, for data links such as a fiber-optic.

De esta forma, tal y como se muestra en la Figura 2-2 , se considera que los tipos de UUV pueden ser clasificados de la siguiente forma:

- En remoto (ROV: *Remotely Operated Vehicle*). Son los que se supervisan a través de un sistema de enlace de datos como los cables fibra óptica. Es un vehículo submarino no tripulado que es controlado de forma remota desde un lugar seguro en la superficie. Los ROVs se utilizan en la industria petrolífera, la industria de la construcción naval, la investigación científica y en operaciones de salvamento. Ver Figura 2-3.

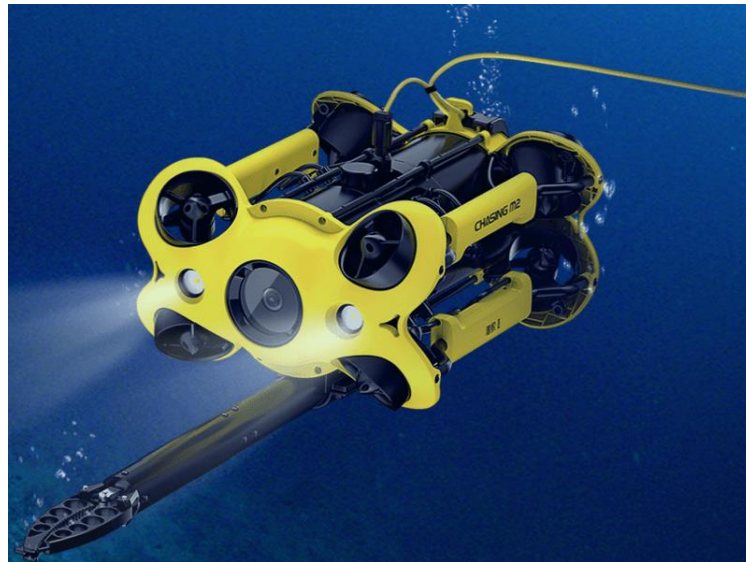


Figura 2-3 ROV modelo *Chasing M2* [14].

- Autónomo (AUV: *Autonomous Underwater Vehicle*). Se denominan así a aquellos vehículos no tripulados cuyos movimientos son controlados por programas instalados previamente. Es un vehículo submarino no tripulado que se mueve de forma autónoma, es decir, sin ser controlado directamente por un ser humano. Algunos AUVs son programados para seguir un curso predefinido, mientras que otros utilizan inteligencia artificial o aprendizaje automático para tomar decisiones y evitar obstáculos, pudiendo generar trayectorias inteligentes que amplían su capacidad para realizar tareas. Ver Figura 2-4.



Figura 2-4 AUV modelo REMUS 600 en la cubierta del barco de NOAA [15].

2.4 Planeador Submarino

Un planeador submarino es un tipo de AUV que tiene coste de desarrollo relativamente bajo y consume poca energía para realizar sus funciones. Resulta un tipo de dispositivo especialmente atractivo por su autonomía, su capacidad de largo alcance y por tener una duración de despliegue extensa [16]. Se usa para la ciencia oceanográfica, tal y como puede verse en Figura 2-5 , de manera pasiva, como por ejemplo con sensores acústicos empleados para obtener mediadas de los sonidos del mundo submarino. Además, puede ser utilizado para campo militar para la vigilancia en una zona determinada.

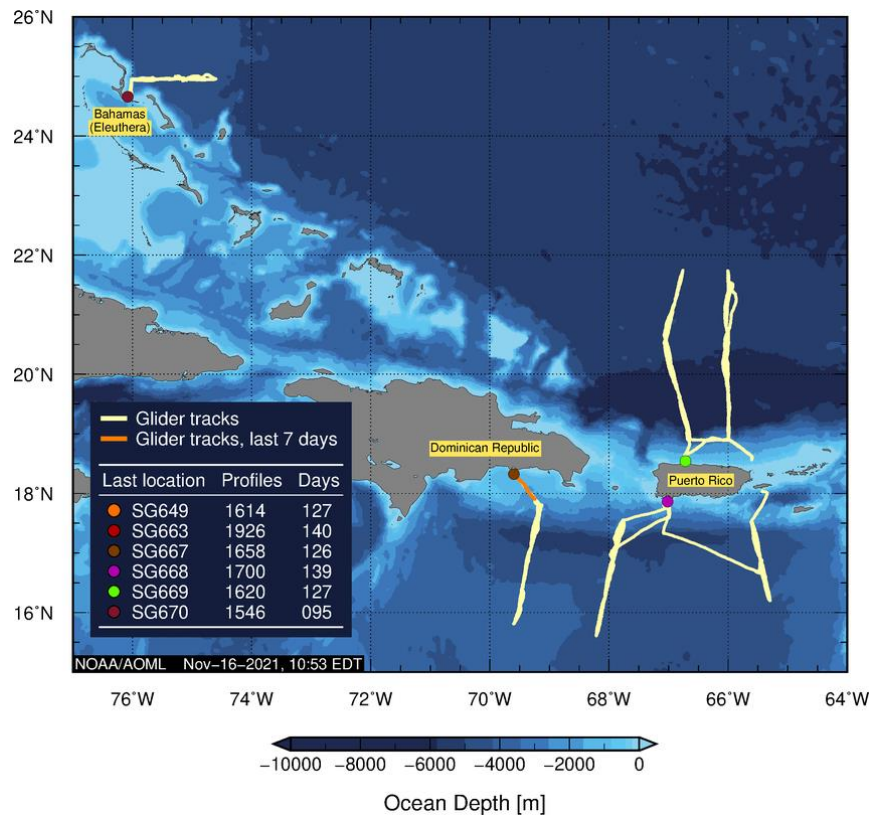


Figura 2-5 Uso de planeador submarino de NOAA [17].

Los componentes básicos de un planeador submarino como Figura 2-6 incluyen:

- Estructura o cuerpo: es la parte principal del planeador submarino, que contiene todos los componentes internos y los protege del agua.
- Sistema de flotabilidad variable: es el componente encargado de permitir al planeador submarino subir y bajar en el agua.
- Sistema de estabilidad: para controlar su dirección de movimiento. Contiene el lastre de cuerpo de masa que es un conjunto de batería, y altas de perfil naca.
- Baterías: proporcionan la energía eléctrica necesaria para los demás componentes del planeador submarino.
- Sistema de control: es el componente encargado de controlar la dirección y la velocidad del planeador submarino.
- Sensores: los planeadores submarinos están equipados con una variedad de sensores, como cámaras, sonar, magnetómetros y otros instrumentos para recopilar datos.

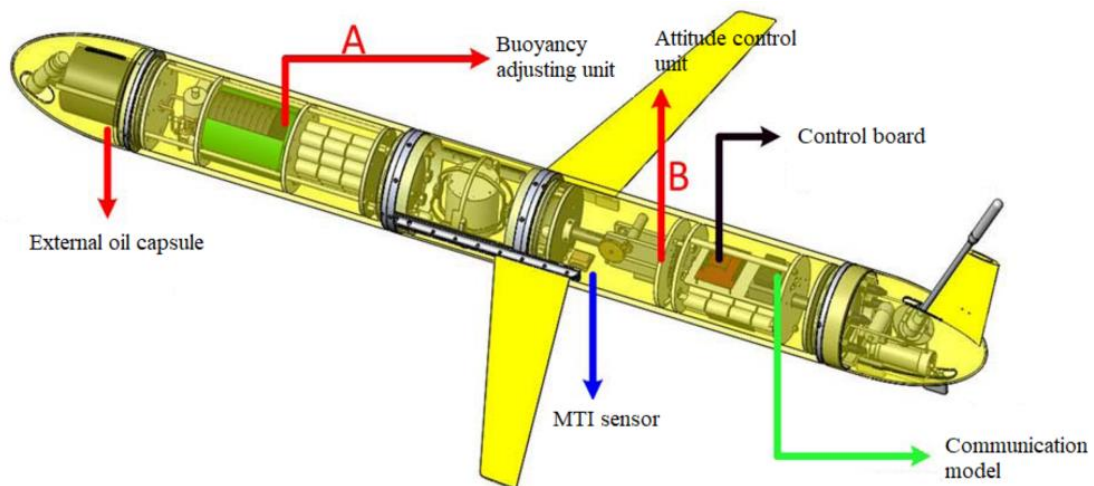


Figura 2-6 el diagrama de los sistemas de un planeador submarino [18].

Los planeadores submarinos también pueden incluir un sistema de comunicación, como muestra el ejemplo de la Figura 2-7, para transmitir los datos recopilados a una estación en la superficie o a una plataforma en tierra [19]. Sin embargo, es importante mencionar que este tipo de UUV sería limitado en cuanto a velocidad y distancia de recorrido debido a su dependencia de las corrientes y olas y a que su capacidad de maniobra es reducida al no disponer de mecanismo autónomos de propulsión.



Figura 2-7 El UUV está enviando los datos a través del satélite de la estación en tierra [19].

Los planeadores submarinos utilizan un sistema de flotabilidad variable para controlar su movimiento a través del agua. El principio básico es que el planeador puede cambiar su densidad para subir o bajar en el agua. La densidad se controla mediante la adición o eliminación de aceite, agua o aire en compartimentos internos del planeador. Además, los planeadores submarinos también pueden moverse horizontalmente mediante el uso de un sistema de propulsión, como un motor eléctrico o un sistema de lastre, así como aprovechando las corrientes naturales del agua.

Los planeadores submarinos pueden seguir trayectorias predefinidas o pueden ser programados para seguir una ruta específica, como una línea de costa o una línea de isobatas (líneas que conectan puntos del fondo del mar con la misma profundidad). También pueden seguir trayectorias controladas en tiempo real, como evitar obstáculos en el agua o seguir un patrón de corriente específico.

2.5 Sistema de control de flotabilidad

Dinámicamente, un planeador submarino tiene su movimiento característico completamente dependiente de su sistema de flotabilidad. El sistema de control de flotabilidad basa su principio de funcionamiento en desplazar un volumen determinado de agua, para aumentar su masa, consiguiendo así que el vehículo tenga una flotabilidad negativa, es decir, su peso sea mayor que la fuerza de empuje

del agua. Para ascender, se requiere el caso inverso, que la flotabilidad sea positiva. Los sistemas de control de flotabilidad más empleados en este tipo de vehículos puede ser clasificada según [20].

- Sistema de inyección de masa: también conocido como sistema de flotabilidad por inyección de agua. Este sistema funciona inyectando agua o un líquido similar en tanques específicos en el planeador submarino. Al inyectar agua en los tanques, se aumenta la masa del planeador submarino, lo que a su vez reduce su flotabilidad y permite que el planeador submarino se sumerja. Por otro lado, al bombear agua de los tanques, se reduce la masa del planeador submarino y aumenta su flotabilidad, permitiendo que el planeador submarino suba a la superficie.
- Sistema de embarca-desembarca de agua (reversible/no reversible): Un sistema de embarque y desembarque de agua reversible es aquel en el cual se puede controlar la entrada y salida de agua de los tanques de lastre del submarino. Esto permite al submarino cambiar su flotabilidad a voluntad y poder sumergirse y emerger a la superficie. Los sistemas de embarque y desembarque de agua reversible son utilizados en submarinos de investigación científica y militares. Por otro lado, un sistema de embarque y desembarque de agua irreversible es aquel en el cual una vez que el agua ingresa a los tanques de lastre no puede ser drenada. Esto significa que una vez que el submarino se sumerge, no puede volver a la superficie a menos que se recurra a otro sistema de flotabilidad como son los flotadores o botellas de aire. Estos sistemas son utilizados en submarinos turísticos o submarinos no tripulados ya que no requieren de un sistema de control de flotabilidad. Ver Figura 2-8.

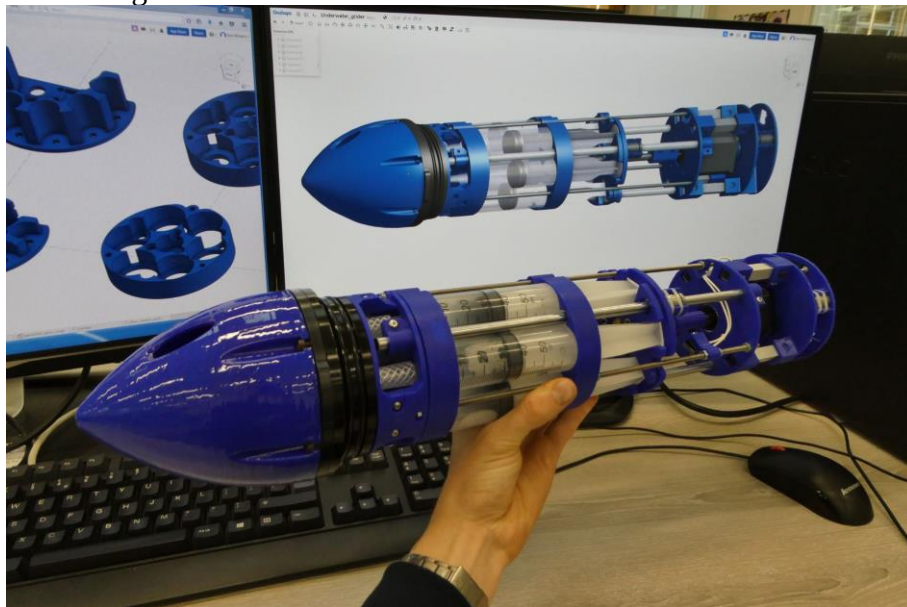


Figura 2-8 Sistema de flotabilidad con jeringuillas de OSUG [21].

- Sistema de transferencia de aceite (reversible/no reversible): se utilizan bolsas o "*bladders*" (vejigas) llenas de aceite o un lubricante similar para controlar la flotabilidad de un submarino. El aceite se emplea debido a su alta viscosidad y baja compresibilidad, ya que esto permite alcanzar altas presiones dentro de la vejiga, permitiendo, de esta forma, desplazar agua de mar dentro de un doble casco del submarino, incluso a grandes profundidades. Este sistema es menos común que otros sistemas de control de flotabilidad debido a su costo y complejidad. Sin embargo, tiene algunas ventajas, como una mayor precisión en el control de la flotabilidad y una mayor seguridad en caso de una falla en el sistema, ya que el aceite no se escaparía fuera del submarino al estar contenido en un circuito hidráulico cerrado. Un ejemplo de este tipo de soluciones es el que se presenta en el modelo SLOCUM que puede ser observado en la Figura 2-9

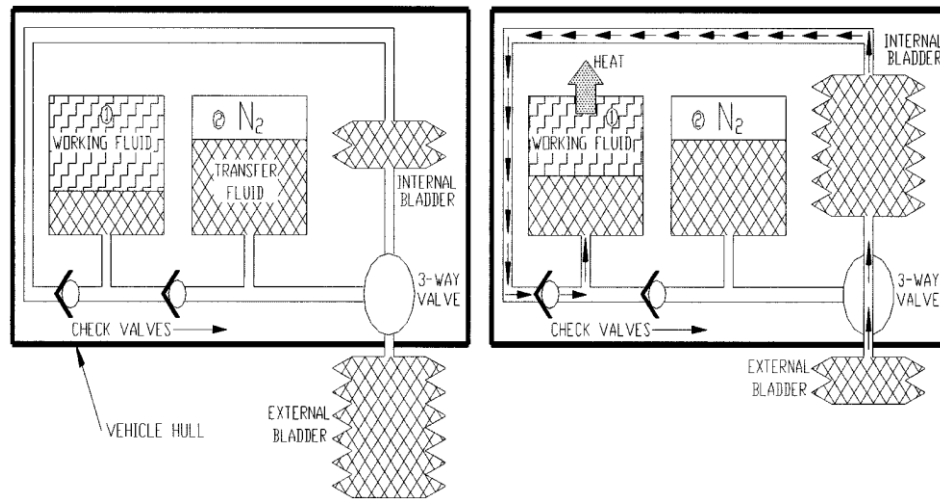


Figura 2-9 Sistema de flotabilidad de SLOCUM [22].

- Sistema híbrido usando el cambio de volumen de flotación (SMA, etc.): El sistema SMA (*Shape Memory Alloy*) es un tipo de material inteligente que cambia su forma cuando se calienta o se enfría. Estos materiales son capaces de "recordar" su forma original y volver a ella cuando se someten a cambios en la temperatura. Estos materiales están compuestos de aleaciones metálicas como el *Nitinol* (ni-ti) que tienen propiedades de memoria de forma. En el contexto de submarinos, el SMA se puede utilizar en un sistema de flotabilidad para controlar la flotabilidad del submarino mediante cambios en la temperatura. Por ejemplo, se pueden instalar actuadores SMA en los tanques de lastre del submarino que se activan cuando se quiere que el submarino se sumerja. Al activar los actuadores, el SMA se calienta y cambia de forma, aumentando la masa del submarino y disminuyendo su flotabilidad. Por otro lado, cuando se quiere que el submarino suba a la superficie, los actuadores se enfrían y vuelven a su forma original, reduciendo la masa del submarino y aumentando su flotabilidad. Ver Figura 2-10.

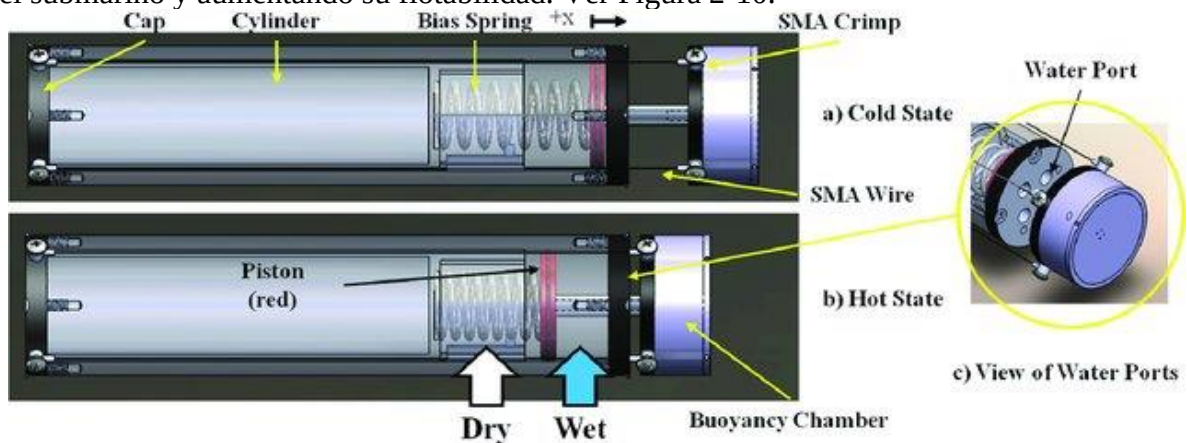


Figura 2-10 El sistema SMA [23].

2.6 Sistema de control de flotabilidad por transferencia de aceite

El sistema de control de flotabilidad variable por transferencia de aceite es un sistema que emplea el cambio de volumen de una vejiga situada en el exterior del casco para ajustar la flotabilidad del vehículo. La vejiga es esencial para el funcionamiento del sistema, ya que está en contacto directo con el fluido y, por lo tanto, puede variar su volumen para ajustar la flotabilidad.

El sistema incluye una serie de componentes clave que trabajan juntos para lograr el ajuste deseado, siendo el depósito de aceite uno de los componentes más importantes es el depósito de aceite, que se

utiliza en el sistema. La bomba hidráulica, con un esquema hidráulico como el mostrado en Figura 2-11, es otro elemento clave, ya que transfiere el aceite desde el depósito hasta la vejiga.

Las válvulas antirretornos son elementos cruciales en el sistema, ya que controlan la dirección del aceite y garantizan que fluya en la dirección correcta. Los sensores también son un elemento importante, ya que recopilan los datos necesarios para realizar el seguimiento y así poder asegurar el correcto funcionamiento del sistema, mientras que el controlador integra todas las funciones de los componentes anteriores para tomar las decisiones que afectan a la activación de los elementos de control.

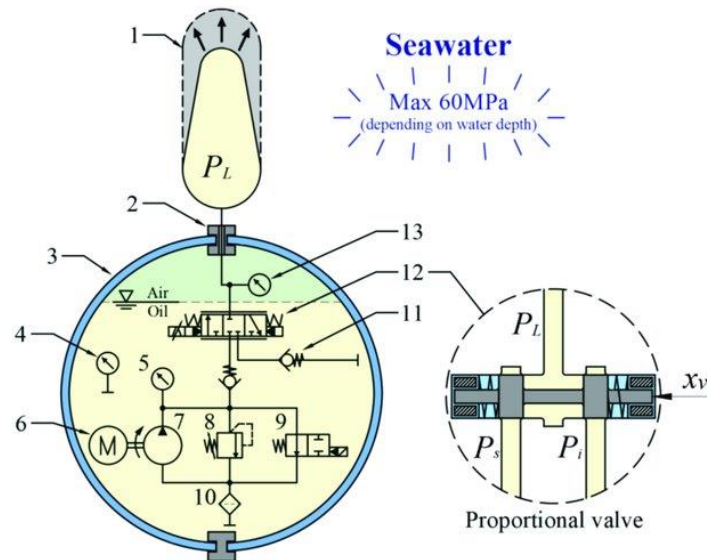


Figura 2-11 Esquema hidráulico tipo Oil Bladder [24].

En resumen, el sistema de control de flotabilidad variable por transferencia de aceite es un sistema altamente complejo que combina: la vejiga, el depósito de aceite, la bomba y componentes hidráulicos, los sensores y el controlador; para lograr un ajuste preciso de la flotabilidad. Este sistema es esencial para una variedad de aplicaciones, incluyendo la navegación marítima, la investigación oceanográfica y la tecnología submarina. La Figura 2-12 presenta un diagrama básico de su funcionamiento.

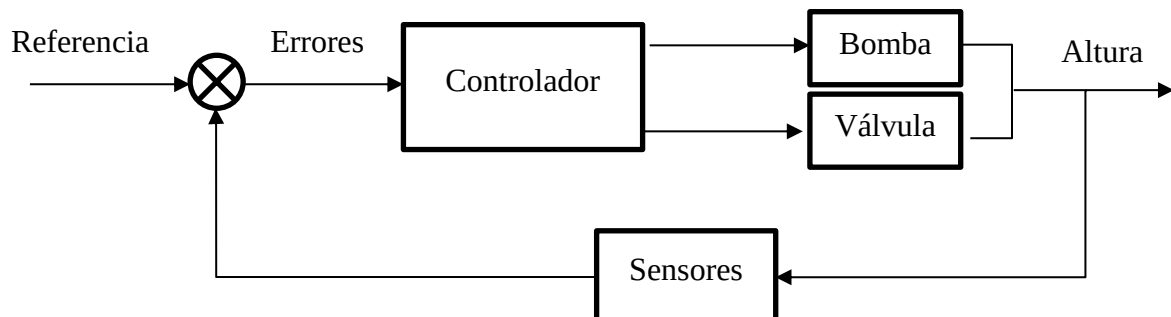


Figura 2-12 Esquema de control básico para el sistema de flotabilidad.

2.7 Microcontroladores

Un microcontrolador es un pequeño ordenador en un circuito integrado único que contiene un núcleo de procesador, memoria y periféricos programables de entrada/salida. Los microcontroladores se utilizan comúnmente en dispositivos IoT, ya que son pequeños, de bajo consumo y capaces de realizar una variedad de tareas, como la recopilación y procesamiento de datos, la comunicación y el control de otros dispositivos o sistemas. Los microcontroladores son el "cerebro" de los dispositivos IoT controlando sus funciones y comunicaciones con otros dispositivos o servidores. Los microcontroladores más conocidos son los siguientes:

- **Arduino:** es una plataforma de desarrollo de hardware y software libre basada en un microcontrolador. Está diseñada para facilitar el uso de los microcontroladores para proyectos de electrónica de bajo nivel, como la creación de dispositivos interactivos, robots, sensores, etc. Arduino proporciona una forma fácil de escribir código para el microcontrolador y una amplia variedad de librerías y herramientas de desarrollo para ayudar en la creación de proyectos [25]. Arduino es compatible con varios tipos de microcontroladores, como el AVR de Atmel y el SAM3X de Atmel, y se utiliza ampliamente en proyectos IoT debido a su facilidad de uso y bajo costo. Hay variedades como Figura 2-13.

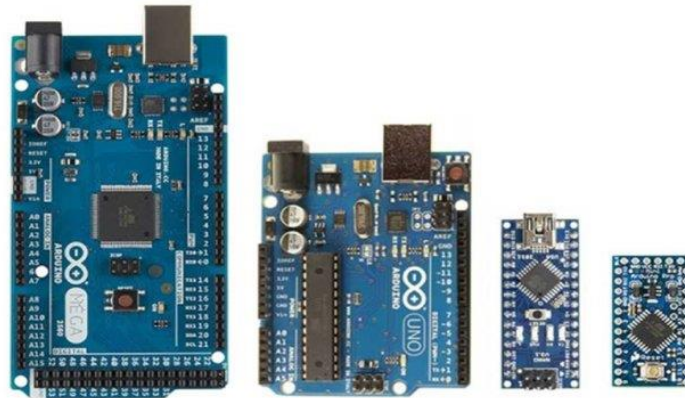


Figura 2-13 Arduino MEGA, UNO, NANO y Pro Mini [26].

- **Raspberry Pi:** Se trata de una computadora de placa única de bajo coste y pequeño tamaño desarrollada por la Fundación Raspberry Pi en el Reino Unido. Está diseñado para promover la enseñanza de la informática básica en escuelas y países en desarrollo. Raspberry Pi tiene una variedad de usos, incluyendo como una computadora básica, un centro de medios, una consola de juegos y una placa de desarrollo para proyectos de hardware [27]. En el caso del modelo Raspberry Pi 4, por ejemplo, dispone de un chip Broadcom BCM2835 (SoC), que incluye un procesador ARM1176JZF-S de 700 MHz, una GPU VideoCore IV y 256 megabytes a 4 gigabytes de RAM. Raspberry Pi tiene varios puertos USB, un puerto Ethernet, salida de audio, salida de vídeo RCA y un conector HDMI, así como un encabezado GPIO (*General Purpose Input/Output*) de 40 pines [28]. Funciona en una variedad de sistemas operativos Linux y es compatible con una amplia gama de lenguajes de programación. Hay variedades como Figura 2-14.

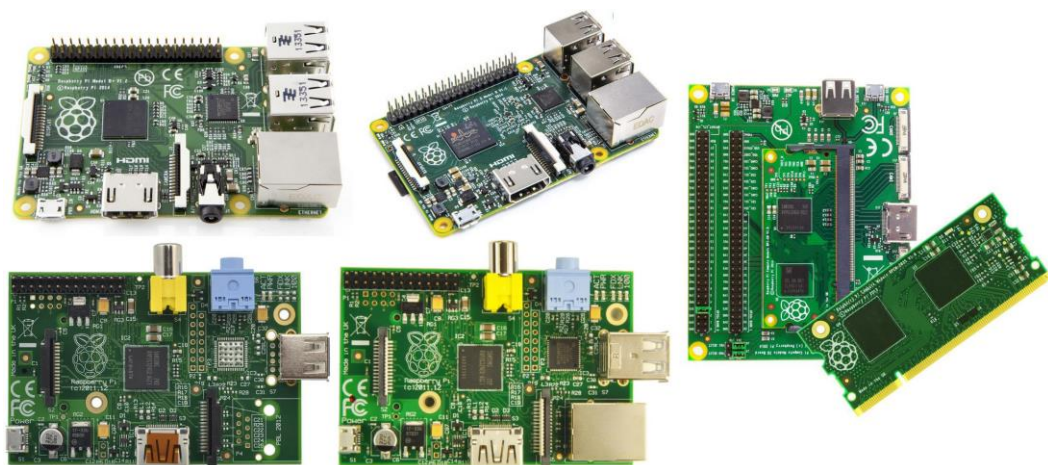


Figura 2-14 La familia de Raspberry Pi [29].

- **ESP32:** Se trata de un microcontrolador de bajo costo y de altas prestaciones desarrollado por *Espressif Systems*. Se trata de una solución integrada que combina Wi-Fi, Bluetooth, y capacidades de procesamiento de alto rendimiento en un solo chip [30]. Esto lo hace ideal para

aplicaciones de Internet de las cosas (IoT) ya que proporciona una manera eficiente de conectar dispositivos a la red y de controlarlos remotamente. ESP32 es conocido por su bajo consumo de energía y su capacidad para ejecutar aplicaciones de forma autónoma, lo que lo hace ideal para aplicaciones de IoT de baja potencia. Además, el ESP32 es compatible con una amplia gama de sistemas operativos y plataformas de desarrollo, lo que lo hace accesible para una gran cantidad de desarrolladores. El ESP32 también cuenta con características avanzadas, como una CPU dual-core, puertos de entrada/salida, comunicaciones serie, y una gran cantidad de periféricos disponibles. Esto lo hace adecuado para una amplia gama de aplicaciones, desde sensores de movimiento y control de iluminación hasta robots y sistemas de seguridad. Hay variedades como Figura 2-15.



Figura 2-15 Los modelos de ESP32 [31].

2.8 Movimiento del vehículo submarino

El movimiento de un vehículo submarino se realiza empleando un modelo dinámico que tiene 6 grados de libertad, y su análisis y comprensión resulta ser un concepto crucial para poder entender los movimientos que es posible generar en la navegación submarina. De esta forma, este tipo de modelos dinámicos establece la capacidad de un submarino de moverse en tres direcciones, así como su orientación determinada por tres ángulos, lo que le permite un alto grado de control y maniobrabilidad en el agua, tal y como se muestra en la Figura 2-16. Los estados o grados de libertad empleados para crear el modelo dinámico son los siguientes:

- Profundidad o pesantez (*Heave*): El submarino puede moverse hacia arriba o hacia abajo en el agua, ajustando su nivel de profundidad. Esto es importante para evitar obstáculos o para mantener una profundidad segura en situaciones de peligro. Coincide con variable “z” en la referencia.
- Avance (*Surge*): El submarino puede moverse hacia adelante o hacia atrás en el agua. Esto es fundamental para la navegación y el control de la velocidad del submarino. Coincide con variable “x” en la referencia.
- Deriva (*Sway*): El submarino puede moverse hacia la derecha o hacia la izquierda en el agua. Esto es importante para evadir obstáculos o para mantener un curso específico. Coincide con variable “y” en la referencia.
- Balance (*Roll*): El submarino puede rotar sobre su eje longitudinal, lo que significa que puede inclinarse hacia un lado o hacia el otro. Coincide con variable “ φ ” en la referencia.
- Cabeceo (*Pitch*): El submarino puede rotar sobre su eje transversal, lo que significa que puede inclinarse hacia adelante o hacia atrás. Coincide con variable “ θ ” en la referencia.

- Guiñada (Yaw): El submarino puede rotar sobre su eje vertical, lo que significa que puede orientarse con respecto a su orientación con respecto al norte terrestre (vista de pájaro). Coincide con variable “ ψ ” en la referencia.

Estos seis grados de libertad permiten al submarino navegar con precisión de forma que es posible generar algoritmos para, por ejemplo, generar trayectorias viables que eviten obstáculos localizados por los sensores en el agua. También le permiten al submarino realizar maniobras para evadir a enemigos o realizar tareas específicas, como la recolección de datos o la realización de misiones militares. En resumen, el movimiento submarino en 6 grados de libertad es fundamental para la navegación submarina y es un aspecto crucial de la tecnología submarina moderna.

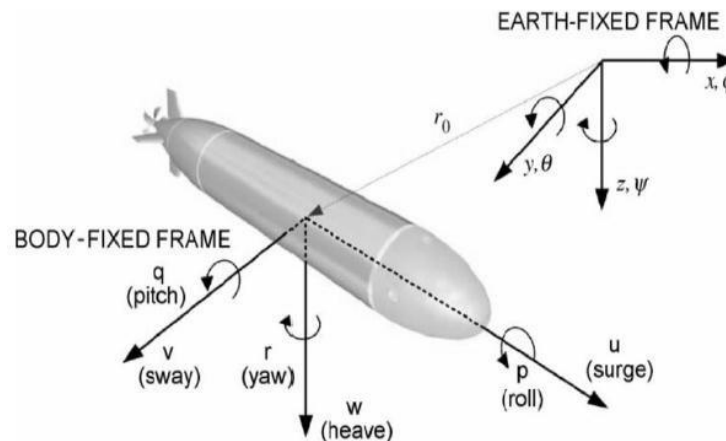


Figura 2-16 6DOF en un vehículo submarino [32].

El planeador submarino emplea un perfil alar para producir una fuerza de sustentación que le permite desplazarse por el agua. Este perfil alar se asemeja a una estructura de ala de avión, y es capaz de generar una fuerza de sustentación a través de la interacción entre la superficie del perfil y el agua. El diseño y la geometría de este perfil alar son críticos para lograr una sustentación eficiente y un desplazamiento óptimo del planeador submarino.

El perfil NACA es un diseño de ala utilizado en aviación, que se caracteriza por su forma curva y suave. Este perfil se utiliza para maximizar la eficiencia aerodinámica de las alas y proporcionar un buen rendimiento en términos de velocidad, maniobrabilidad y estabilidad. El movimiento que hace un ala de perfil NACA se basa en la circulación del aire alrededor de la superficie del ala. Cuando un avión se mueve a través del aire, el aire fluye sobre la superficie superior del ala y se separa en un flujo laminar en la parte posterior. Debido a la forma curva y suave del perfil NACA, la superficie inferior del ala genera una mayor presión, lo que ayuda a mantener el flujo de aire pegado a la superficie del ala.

Este flujo de aire se ve afectado por el ángulo de ataque del ala, que es el ángulo entre la dirección del viento y la superficie del ala, tal y como se presenta en la Figura 2-17. Cuando el ángulo de ataque es mayor, el flujo de aire se desvía hacia arriba y genera una mayor sustentación, lo que permite al avión elevarse. Además, el perfil NACA también es capaz de generar tanto sustentación como arrastre, lo que significa que puede mantener el avión en el aire y proporcionar la resistencia necesaria para moverse a través del aire. Esta combinación de sustentación y arrastre permite al avión volar de manera eficiente y maniobrar con precisión.

Sin embargo, el movimiento submarino es más complejo que el movimiento aéreo debido a las diferencias en las propiedades del agua y del aire. En el agua, la viscosidad es mucho mayor que en el aire, lo que significa que el flujo de agua alrededor de las alas del planeador submarino es más resistente. Esto significa que el planeador submarino requiere más potencia para mantener su velocidad y maniobrar que un avión.

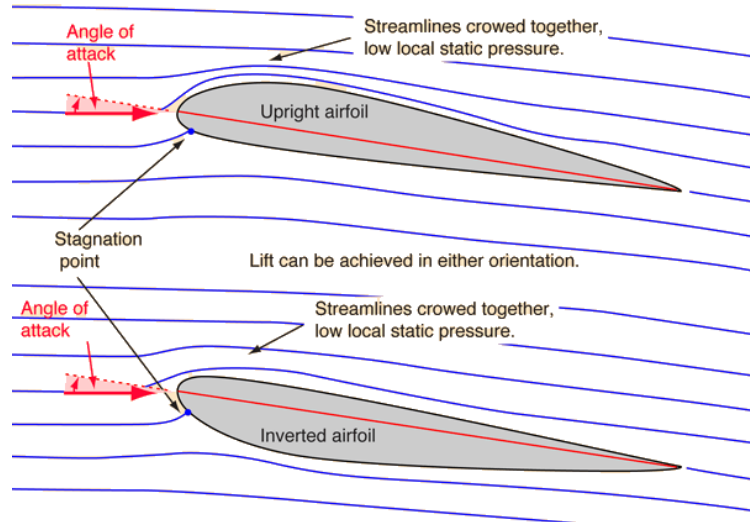


Figura 2-17 Ala de perfil NACA [33].

Bernoulli es un principio físico que describe cómo la velocidad de un fluido (ya sea un gas o un líquido) afecta la presión en ese fluido. Este principio se basa en la ley de conservación de la energía, que establece que la energía total en un sistema permanece constante. Según Bernoulli, a medida que la velocidad de un fluido aumenta, la presión disminuye y viceversa. Esto se debe a que la energía cinética del fluido, que depende de su velocidad, aumenta a medida que la presión disminuye.

Este principio es ampliamente utilizado en una variedad de aplicaciones, incluyendo aerodinámica, ingeniería naval y medicina. Por ejemplo, en la aerodinámica, el principio de Bernoulli se utiliza para explicar la sustentación que sostiene un avión en el aire. En un avión, el aire fluye sobre las superficies curvas de las alas y se desvía hacia arriba, generando una disminución en la velocidad y un aumento en la presión. Esta mayor presión en la parte inferior del ala empuja hacia arriba al avión, lo que permite que mantenga su posición en el aire.

$$P_i + \rho_i v_i^2 \frac{1}{2} + \rho_i g h_i = cte \quad (1)$$

El principio de Arquímedes es una ley fundamental de la física que describe el comportamiento de los cuerpos sumergidos en un fluido. Según este principio, un cuerpo sumergido en un fluido experimenta una fuerza hacia arriba, conocida como fuerza de flotación, igual al peso del volumen de fluido desplazado por el cuerpo, tal y como se muestra en la Figura 2-18. Este principio se basa en el hecho de que un objeto sumergido en un fluido desplaza un volumen de fluido igual a su propio volumen, y que la fuerza hacia arriba ejercida por el fluido es igual a la densidad del fluido multiplicada por el volumen desplazado y la gravedad.

$$F_{Bouyant} = \rho_{fluid} V_{Submarine} g \quad (2)$$

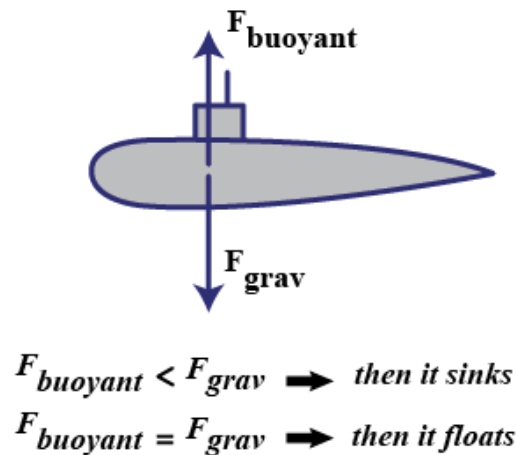


Figura 2-18 El principio de Arquímedes en un vehículo submarino [34].

2.9 Investigación submarina

La investigación submarina implica la necesidad de enfrentarse a un entorno que presenta ciertas propiedades particulares que dificultan el desarrollo de soluciones autónomas. Además de tener que actuar en un entorno hostil, teniendo que soportar altas presiones, que genera una alta corrosión de elementos por humedad, hay que tener en cuenta que las comunicaciones inalámbricas son más difíciles de generar que en un medio como el aire. De esta forma, las pérdidas generadas en la transmisión de datos en condiciones subacuáticas implican la necesidad de crear sistemas con un grado de autonomía suficiente que les permita recuperarse automáticamente en caso de fallo de comunicaciones.

Las comunicaciones subacuáticas suelen basarse en la transmisión de información por sonido, teniendo en cuenta que a mayores profundidades la velocidad del sonido es superior debido al aumento de presión. De esta forma, según la ley de Snell la trayectoria de la propagación sufre una refracción que genera una curva hacia la zona de menor velocidad, tal y como se muestra en la Figura 2-19. Otros factores que afectan a la calidad de la transmisión de comunicaciones son la temperatura, la salinidad del agua o el gradiente de densidad submarino.

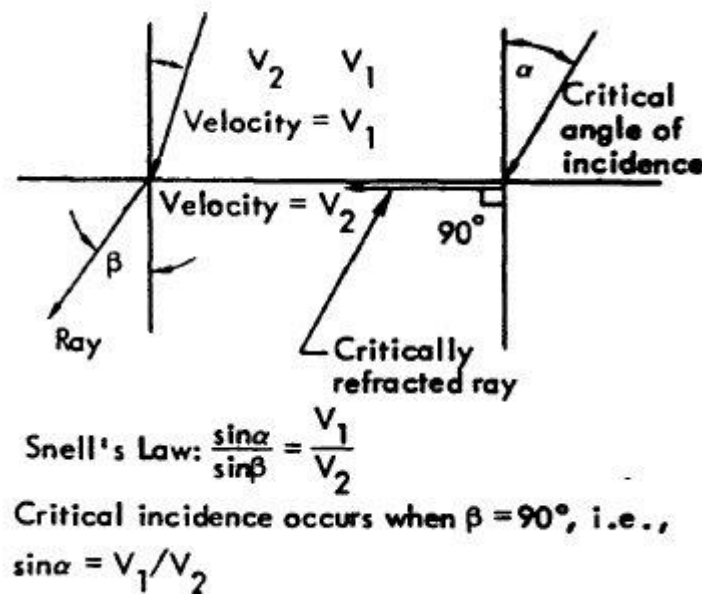


Figura 2-19 Snell law [35].

$$\text{Ley de Snell: } \frac{\sin \beta_1}{c_1} = \frac{\sin \beta_2}{c_2} \quad (3)$$

Este gradiente se refiere a la variación en la densidad de la columna de agua en el fondo del mar. Esta variación de densidad puede, además ser afectada por diferencias en la salinidad o en la temperatura del agua, tal y como se muestra en la Figura 2-20. El gradiente submarino puede tener un gran impacto en la vida marina, ya que puede afectar la distribución de las especies y las corrientes oceanográficas, y se puede medir utilizando un medidor CTD (Conductividad, Temperatura y Densidad) medidor[36]. Este instrumento mide la conductividad, la temperatura y la densidad del agua en diferentes profundidades, permitiendo a los oceanógrafos calcular el gradiente submarino.

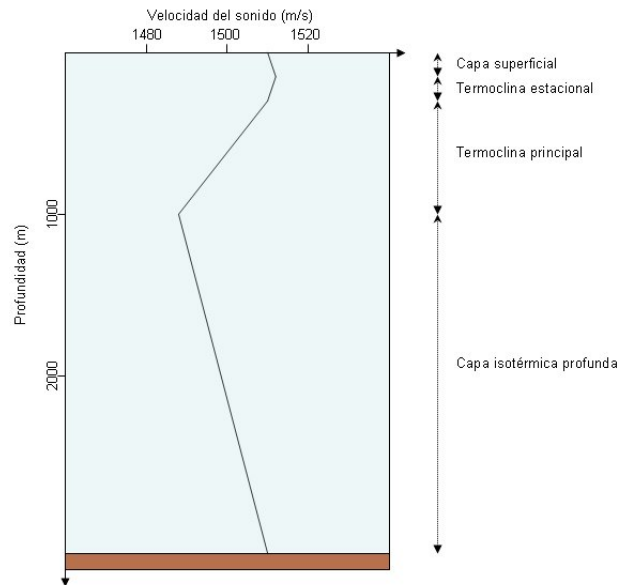


Figura 2-20 Perfil de la velocidad de sonido según la profundidad [37].

En la armada, para el desarrollo de la guerra antisubmarina (*Anti-submarine Warfare – ASW*) se usa como medio de detección sonares que utilizan rayos sonoros. Ya que la velocidad acústica está afectada por los factores mencionados, se puede dividir las capas de agua en cada zona donde tengan las características concretas. Por lo que para ASW, hay 4 métodos de propagación para detección, que son los siguientes.

1. Rayo directo (*Direct Path – DP*) o Canal de sonoro superficial (*Surface Duct – SD*): Es el método más simple de propagación en el que el sonido se envía directamente desde la fuente de emisión hasta el objeto y luego se refleja de regreso al receptor. Este método es efectivo para la detección de objetos cercanos y en aguas poco profundas. Ver Figura 2-21.

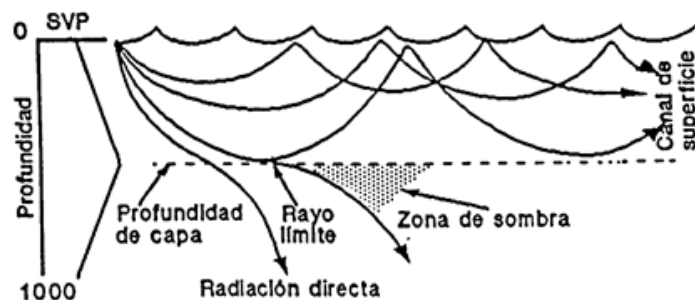


Figura 2-21 Rayo directo o Canal de sonoro superficial [38].

2. Canal de sonoro (*Sound Chanel – SC*): Transmite a través de un medio físico, como el agua, y se ve afectado por las características del medio y del entorno. El sonido viaja a través del agua y puede ser afectado por factores como la salinidad, la temperatura, la velocidad de la corriente, la profundidad y la presencia de objetos y obstáculos en el camino. Ver Figura 2-22.

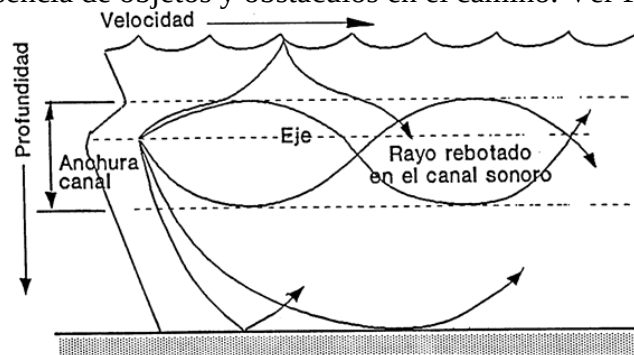


Figura 2-22 Canal de sonoro [38].

3. Zona de Convergencia (*Convergence Zone – CZ*): se refiere a un fenómeno que ocurre en el agua en el que dos o más masas de agua con características diferentes, como la salinidad y la temperatura, se encuentran y se mezclan. Esto puede ocurrir en el océano, en los ríos o en las bahías. Cuando ocurre este fenómeno, se produce una zona de mezcla en la que las características del agua cambian bruscamente. Ver Figura 2-23.

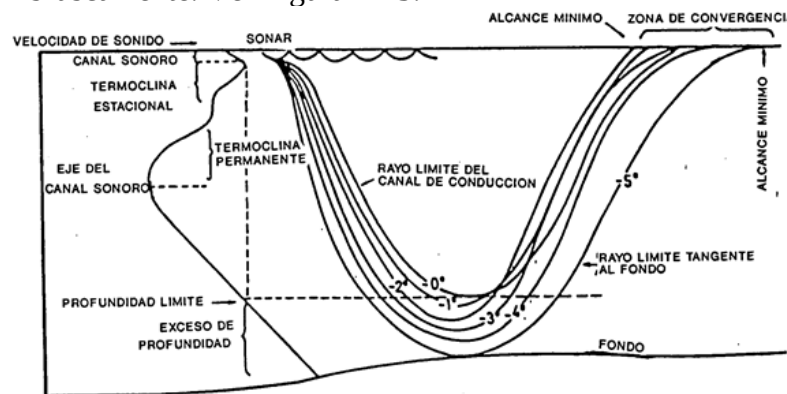


Figura 2-23 Zona de Convergencia [38].

4. Rebote en el fondo (*Bottom Bounce – BB*): El sonido se refleja en el fondo del mar antes de llegar al objeto y luego se refleja de regreso al receptor. Este método es útil para la detección de objetos en aguas profundas debido a la mayor propagación del sonido en el fondo del mar. Ver Figura 2-24.

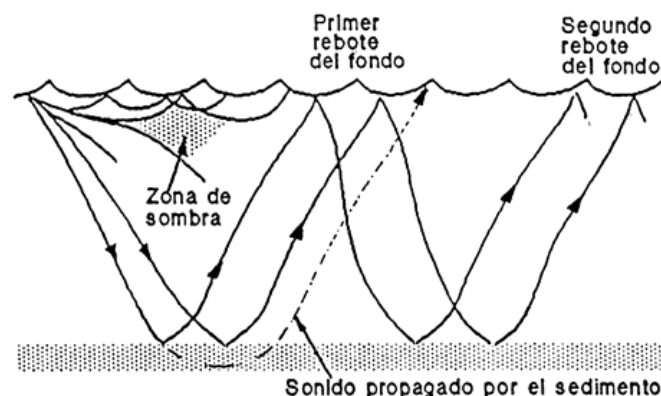


Figura 2-24 Rebote en el fondo [38].

3 MATERIALES Y MÉTODOS

3.1 Consideraciones desde modelo matemático

3.1.1 Modelo cinético

Los marcos de referencia fijos a la Tierra son aquellos que están anclados a la Tierra y no rotan con relación a la esfera celeste. Estos marcos de referencia se utilizan para describir la posición y orientación de objetos en la Tierra y en el espacio cercano a ella. Por otro lado, los marcos de referencia fijos al cuerpo son aquellos que están conectados a un cuerpo en movimiento y giran con él. Estos marcos de referencia se utilizan para describir la posición y orientación de objetos que se encuentran en movimiento. La orientación de un marco de referencia fijo al cuerpo se define con relación a la posición y orientación del cuerpo en el espacio. Por ejemplo, un marco de referencia fijo al cuerpo de un satélite girará con el satélite mientras se desplaza alrededor de la Tierra. Esto permite describir la posición y orientación del satélite en el espacio de una manera coherente y uniforme en todo momento, independientemente de su posición y orientación en cualquier momento dado.

Sistema de referencia:

$$\overrightarrow{\Gamma^E} = (X, Y, Z)^T \quad (4)$$

$$\overrightarrow{x^B} = (x, y, z)^T \quad (5)$$

Dónde X, Y, Z son los ejes de referencia inercia fija a la Tierra y x, y, z son los ejes de referencia del cuerpo.

Vectores de velocidad en la referencia fijo:

$$\overrightarrow{\dot{\Gamma}^E} = (\dot{X}, \dot{Y}, \dot{Z})^T \text{ m/s} \quad (6)$$

$$\overrightarrow{\dot{\theta}^E} = (\dot{\varphi}, \dot{\theta}, \dot{\psi})^T \text{ rad/s} \quad (7)$$

Dónde $\dot{X}, \dot{Y}, \dot{Z}$ son las velocidades traslacionales de referencia inercia y $\dot{\varphi}, \dot{\theta}, \dot{\psi}$ son las velocidades rotacionales del mismo.

Vectores de velocidad en la referencia de cuerpo:

$$\overrightarrow{V^B} = (u, v, w)^T \text{ m/s} \quad (8)$$

$$\overrightarrow{w^B} = (p, q, r)^T \text{ rad/s} \quad (9)$$

$$\overrightarrow{v^B} = (\overrightarrow{V^B}, \overrightarrow{w^B})^T = (u, v, w, p, q, r)^T \quad (10)$$

Dónde u, v, w son las velocidades traslacionales de referencia del cuerpo y p, q, r son las velocidades rotacionales del mismo.

Matriz de rotación:

$$R_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{bmatrix}, R_Y = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}, R_Z = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R = R_{XYZ} = R_{zyx} = R_Z R_Y R_X$$

$$R = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \varphi - \sin \psi \cos \varphi & \cos \psi \sin \theta \cos \varphi + \sin \psi \sin \varphi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \varphi + \cos \psi \cos \varphi & \sin \psi \sin \theta \cos \varphi - \cos \psi \sin \varphi \\ -\sin \theta & \cos \theta \sin \varphi & \cos \theta \cos \varphi \end{bmatrix} \quad (11)$$

Dónde ϕ es ángulo de giro respecto el eje x o ángulo balancee, θ es ángulo de giro respecto el eje y o ángulo cabecero y ψ es ángulo de giro respecto el eje z o ángulo guiñada.

Por lo que, para cambiar de sistema referencia inercia a sistema de referencia del cuerpo y viceversa.

$$\begin{bmatrix} \dot{\vec{r}}^E \\ \dot{\vec{\theta}}^E \end{bmatrix} = R \cdot \overrightarrow{v^B}$$

$$\overrightarrow{v^B} = R^{-1} \cdot \begin{bmatrix} \dot{\vec{r}}^E \\ \dot{\vec{\theta}}^E \end{bmatrix} \quad (12)$$

3.1.2 Modelo dinámico

En el sistema de flotabilidad se centra en el control de la posición vertical, es decir, en la gestión del movimiento ascendente y descendente en el agua. Considerando esta simplificación, el espacio que determina la posición del vehículo pasa de tener 3 dimensiones (X, Y, Z) a tener únicamente 1 dimensión (solo Z). Este apartado muestra las leyes físicas que se usan para crear un modelo dinámico no lineal de la planta que simula este compartimiento dinámico del movimiento vertical del elemento submarino.

Se define el peso total como:

$$W_{tot} = W_{DW} - B_{GB} - B_{BB} \quad (13)$$

Dónde W_{tot} es el peso total, W_{DW} es el peso del planeador submarino en el aire $W_{DW} = mg$, B_{GB} es la fuerza de empuje en el estado inicial respecto del volumen del planeador submarino y B_{BB} es la fuerza de empuje respecto del cambio de volumen en la vejiga.

Se define la fuerza de resistencia en el agua como:

$$F_D = \frac{1}{2} C_D \rho \dot{z}^2 A_S \quad (14)$$

Dónde F_D es la fuerza de resistencia en el agua, C_D es el coeficiente del flujo respecto a número de Reynolds y la geometría del cuerpo [39], ρ es la densidad del agua salada $\rho = 1025 \text{ kg/m}^3$, $\dot{z}$ es la velocidad traslacional en el eje Z, A_S es el área proyectada a la dirección de movimiento que en este

caso es lado del tubo por su diámetro $A_S = l \cdot D$. El coeficiente del flujo se aproxima a $C_D = 0,4$ para un vehículo submarino elipsoidal [40].

Se define la presión exterior/en la vejiga como:

$$P_{ex} = \rho g |Z| \quad (15)$$

, dónde P_{ex} es la presión exterior y $|Z|$ es la profundidad.

Caudal de descarga:

Se define el caudal de descarga a partir de la ecuación de Bernoulli

$$P_{ex} + \rho_{oil} v_{ex}^2 \frac{1}{2} + \rho_{oil} g h_{ex} = P_i + \rho_{oil} v_i^2 \frac{1}{2} + \rho_{oil} g h_i \quad (16)$$

$$v_i = \sqrt{\frac{2(P_{ex} - P_i)}{\rho_{oil}}} \quad (17)$$

de forma que se obtiene la expresión del caudal:

$$q_i = A_i v_i = A_i \sqrt{\frac{2(P_{ex} - P_i)}{\rho_{oil}}} \quad (18)$$

Dónde q_i es el caudal de descarga de la vejiga P_i es la presión en el tanque de aceite que será la presión atmosférica ($P_i = 1 \text{ bar}$), ρ_{oil} es la densidad de aceite usado para el trabajo y A_i es el área de la sección del tubo $A_i = \frac{\pi D^2}{4}$

Se define la fuerza de empuje como:

$$B_{BB} = \rho w_{BB} g \quad (19)$$

$$B_{BB} = \rho g \int q dt \quad (20)$$

Dónde w_{BB} es el volumen de la membrana en la vejiga, q es el caudal de entrada/salida de la vejiga.

$$q = \begin{cases} q_b, \text{ es el caudal de la bomba (constante)} \\ q_i(Z), \text{ es el caudal de descarga (variable)} \end{cases}$$

La aplicación de las Leyes de Newton lleva a obtener:

$$+\uparrow \sum F_z = m\ddot{z} = B_{BB} + B_{GB} - W_{DW} - F_D \quad (21)$$

$$\ddot{z} = \frac{\rho g}{m} \int q dt + \frac{B_{GB} - W_{DW} - F_D}{m} \quad (22)$$

Derivando la expresión anterior se tiene:

$$\ddot{z} = \frac{\rho g q}{m} - C_D \rho A_S \dot{z} \quad (23)$$

3.1.3 Ecuaciones del espacio de estado

En este trabajo, se elige las variables siguientes como estado $\vec{x} = [z \quad \dot{z} \quad \ddot{z}]^T$. La entrada del sistema será el caudal $\vec{u} = [q]$ y necesitamos disponer de la referencia de entrada como el valor de la profundidad deseada. A partir de los resultados indicados, las ecuaciones del espacio de estado, asociadas al modelo dinámico del sistema planteado, son las siguientes:

$$\vec{\dot{x}} = \begin{bmatrix} \dot{z} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} \dot{z} \\ \frac{\rho g q}{m} - C_D \rho A_S \ddot{z} \end{bmatrix} \quad (24)$$

$$\vec{y} = [1 \quad 0 \quad 0] \vec{x} \quad (25)$$

Las ecuaciones del espacio de estado pueden escribirse de la siguiente forma:

$$\vec{\dot{x}} = A\vec{x} + B\vec{u} \quad (26)$$

$$\vec{y} = C\vec{x} + D\vec{u} \quad (27)$$

, es decir:

$$\vec{\dot{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -C_D \rho A_S \end{bmatrix} \begin{bmatrix} Z \\ \dot{z} \\ \ddot{z} \end{bmatrix} + \begin{bmatrix} \rho g \\ m \end{bmatrix} [q] \quad (28)$$

$$\vec{y} = [1 \quad 0 \quad 0] \begin{bmatrix} Z \\ \dot{z} \\ \ddot{z} \end{bmatrix} + [0][q] \quad (29)$$

, en donde:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -C_D \rho A_S \end{bmatrix}, B = \begin{bmatrix} \rho g \\ m \end{bmatrix}, C = [1 \quad 0 \quad 0], D = [0]$$

3.2 Sistema hidráulico

3.2.1 Factores a considerar

En este trabajo, se considera un sistema de control de flotabilidad por transferencia de aceite a una vejiga de membrana flexible en el casco exterior. Los sistemas situados en el interior del cilindro de presión mantienen una buena estanqueidad, por lo que, la presión interior es la presión atmosférica que se asume constante. Por otro lado, la membrana, que está en el exterior del cilindro de presión, está sujeta a la presión ejercida por la profundidad de la mar, y por lo tanto a una presión variable.

3.2.2 Componentes

- **Depósito combustible de tamaño 1000 mL [41].**



Figura 3-1 Tanque combustible.

Especificación:

Condición: 100% nuevo

Material: Plástico

Color: Transparente

Entrada de aire / entrada de combustible: 4,5 mm Material Metal

Combustible Outlet: 5mm

Tamaño Cuerpo principal: 215 * 78 * 78mm / 8.46 * 3.07 * 3.07inch

Peso Neto: 120 g

- **Bomba hidráulica Brushless mini M3 con motor [42].**



Figura 3-2 Bomba hidráulica sin escobilla.

ESPECIFICACIONES:

Presión máxima: 20bar

Válvula de alivio: No

Diámetro de tubería: 4 mm

Racor: Recto

Peso: 83g

Dimensiones: 22 mm * 30,56 mm * 51,10 mm

- **Sharplace Mini Bomba de Succión Autocebante [43]**



Figura 3-3 -Sharpplace Mini Bomba de Succión Autocebante

Especificaciones:

Tensión nominal: DC 12V

Diámetro: Aprox. 27 mm (motor 370)

Ruido: < 60 db

Corriente clasificada: < 150 mA

Uso: uso como bomba de aire, bomba de agua. bomba de succión, vacío Bomba autocebante.

Material: Aluminio

Altura: aprox. 61 mm

Caudal de agua: 0.4-0.7L / min

Color plata

Rango de temperatura de funcionamiento: 5-45 °C

- **Válvula Solenoide 12 V de 2 posiciones 3 vías de puerto [44].**



Figura 3-4 Válvula Solenoide 2 posiciones 3 vías.

Especificaciones:

Modelo: 3V1-06

Modo de funcionamiento: NC, normalmente cerrado

Acción Directa

Tamaño del puerto: 1/8" BSP

Voltaje de la bobina: DC: 12V 3W

Frecuencia máxima: 10 Hz

Tipo de válvula solenoide: 3 puertos de 2 posiciones

Presión de funcionamiento: 0-10 kgf/cm²

Prueba de presión: 1.5 Mpa

Protección: IP65

Temperatura: 5-50 °C

Material del cuerpo: aleación de aluminio

Nota:

1: ① Puerto → ② Puerto, siempre está abierto.

2: ① puerto o ② Puerto → ③ Puerto,

Cuando se apaga, se cierra

Cuando se enciende, está abierto

- **Válvula Solenoide 12 V de 2 vías normalmente cerradas [45].**

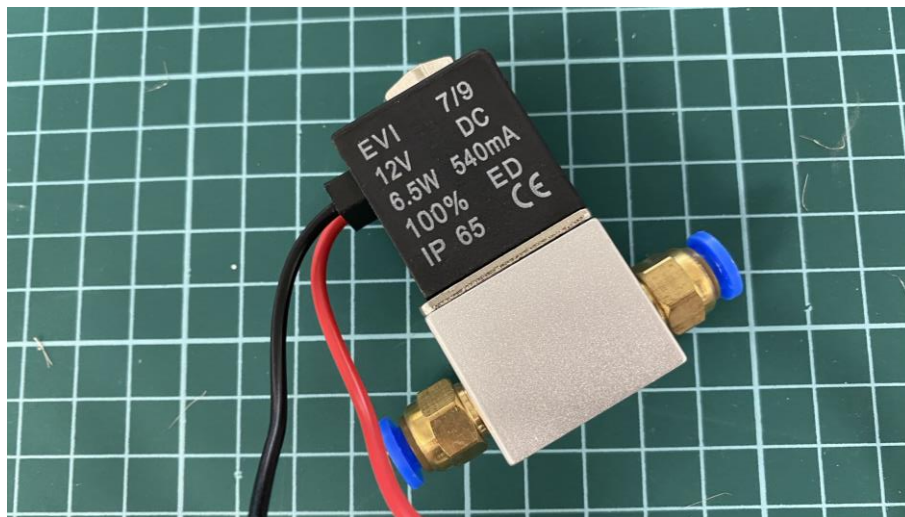


Figura 3-5 Válvula de 2 vías.

Especificaciones:

Modelo: 2V025-1/8-12VDC

Tipo de válvula: 2 vías normalmente cerradas (la válvula se abre con activación).

Tamaño del puerto: 1/4" BSP Puerto hembra NPT, Puerto hembra NPT de 1/8" BSP

Voltaje de la bobina: 12V DC

Medios de servicio: aire, gas, líquido, agua

Modelo de operación: Acción Directa

Modelo de flujo: unidireccional

Caudal: 0,23 cv (flujo de fluido)

Tiempo de respuesta: menos de 50 ms

Presión de funcionamiento: 1,2 MPa (se puede utilizar a 0 PSI (vacío))

Temperatura de funcionamiento: -10 a 80 °C

Potencia de la bobina: 6,5 W

Material del cuerpo: Aluminio

Material del sello: NBR (Buna N)

Material del tubo de la armadura: latón

Émbolo y Material del resorte: acero inoxidable

- **2 antirretornos de rosca 1/4" [46].**



Figura 3-6 Antirretorno [46].

Especificaciones:

Material: acero inoxidable.

Conexión: rosca 1/4" BSP

Rango de temperatura de funcionamiento: 0 a 70 °C.

Medio aplicable: aceite, agua, gas y otros medios no corrosivos o corrosivos.

- **Membrana de vejiga de 1 L [47].**



Figura 3-7 Membrana para la vejiga.

Especificaciones:

Material: NBR (*Nitrile rubber*)

Volumen: 1 L

Dimensiones: 180 mm x 44 mm x 49 mm

Rosca interna: M16 x 1,5

Rosca externa: M24 x 1,5

- **Tubo neumático 6 mm x 4 mm [48].**



Figura 3-8 Tubería [48].

Especificaciones:

Material: PU (*polyurethane*)

Tamaño: 6 mm x 4 mm

Presión máxima: 2,5 MPa

Presión de trabajo: 1 MPa

Temperatura de trabajo: -20 °C a 80 °C

Resistencia a la tracción: 400 kgf/cm²

Resistencia al desgarro: 110 kgf/cm²

Resistencia a la abrasión: 35 mg

Máxima elongación: 450 %

Dureza: 95 ± 2 A

- Juego de conectores de tubería tipo PC6 y adaptadores

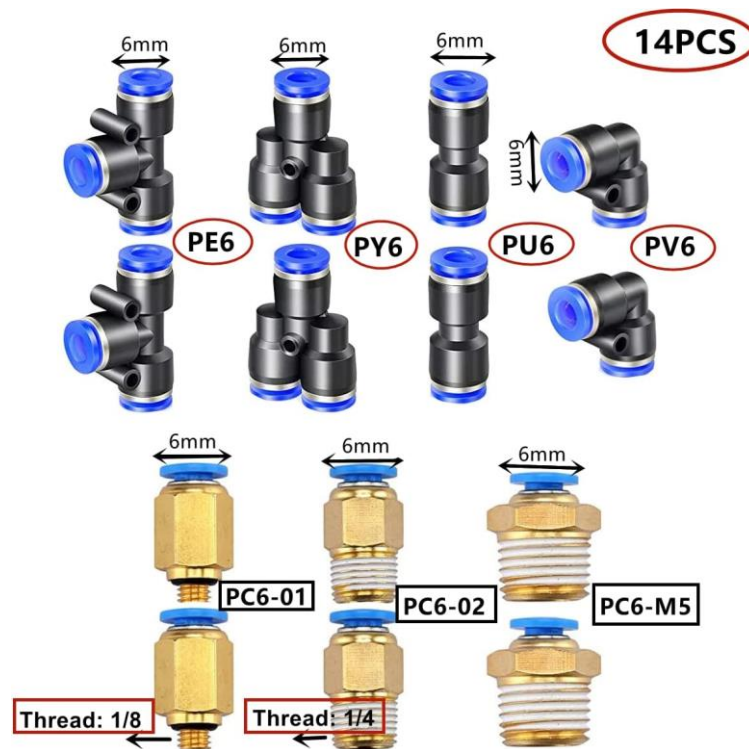


Figura 3-9 Conectores de tuberías [49].

3.2.3 Diseños propuestos

El circuito del sistema hidráulica está diseñado por el programa FluidSIM como puede ver en Figura 3-10. El circuito tiene característica siguiente: El tanque de aceite está conectado con una salida hacia la bomba y dos entradas, una de la válvula de 3 vías para evitar la fuerza de reacción en caso de que la bomba funciona con el paso de una válvula cerrada que puede dañar a la bomba, otra con la válvula de 2 vías para la descarga de la vejiga. Después de cada válvula está un antirretorno para dejar que el aceite fluya solo en un sentido. Se unen las dos vías de componentes interiores y se conecta con la membrana de la vejiga.

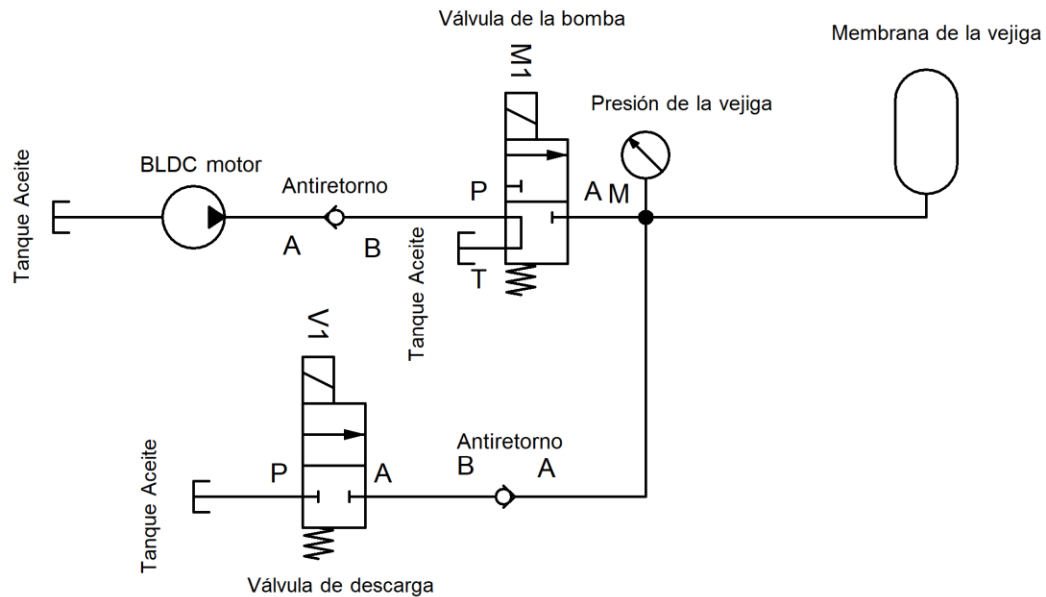


Figura 3-10 El circuito del sistema hidráulico.

3.3 Sistema electrónico, integración de programa autónoma

3.3.1 Componentes

En este trabajo, con este sistema de control de flotabilidad, tiene tres componentes principales para controlar electrónicamente que son: la bomba de motor sin escobillas con presión máxima 20 bares, la válvula de 3 vías y 2 posiciones AIDTAG 3V1-06 y la válvula de 2 vías y 2 posiciones 2V025-1/8-12VDC. Además, debe considerarse el tamaño del sistema electrónico de forma que este pueda integrarse dentro de las dimensiones del tubo que forma el recipiente a presión (120 mm de diámetro exterior). Son necesarios, además los siguientes elementos.

- **Microcontrolador ESP32 Devkit V1 [31].**

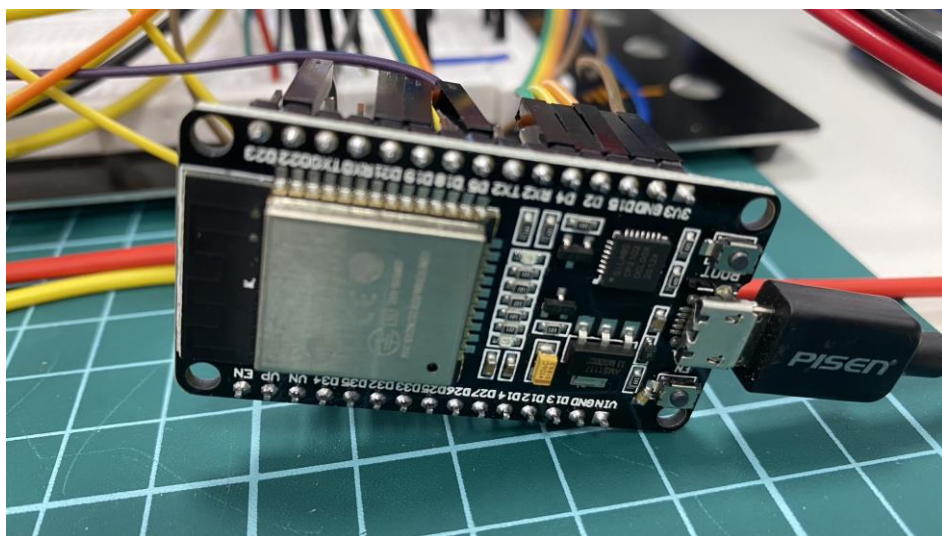


Figura 3-11 ESP32 Deckit V1.

Especificaciones:

Número de cores: 2 (Dual core)

Wi-Fi: 2.4 Ghz hasta 150 Mbps

Bluetooth: BLE (Bluetooth Low Energy) y Bluetooth legado

Arquitectura: 32 bits

Frecuencia de reloj: Hasta 240 Mhz

RAM: 512 kB

Pins: 30

Periférico: Toque capacitivo, ADC (*analog to digital converter*), DAC (*digital to analog converter*), I2C (*Inter-Integrated Circuit*), UART (*universal asynchronous receiver/transmitter*), CAN 2.0 (*Controller Area Network*), SPI (*Serial Peripheral Interface*), I2S (*Integrated Inter-IC Sound*), RMII (*Reduced Media-Independent Interface*), PWM (*pulse width modulation*), y más.

Botones incorporados: RESET y BOOT botones

LEDs incorporados: LED azul incorporado conectado a GPIO2, LED rojo incorporado demuestra que la placa está siendo alimentado.

Puerto USB a UART: CP2102

- 3 relés SRD-3VDC [50].

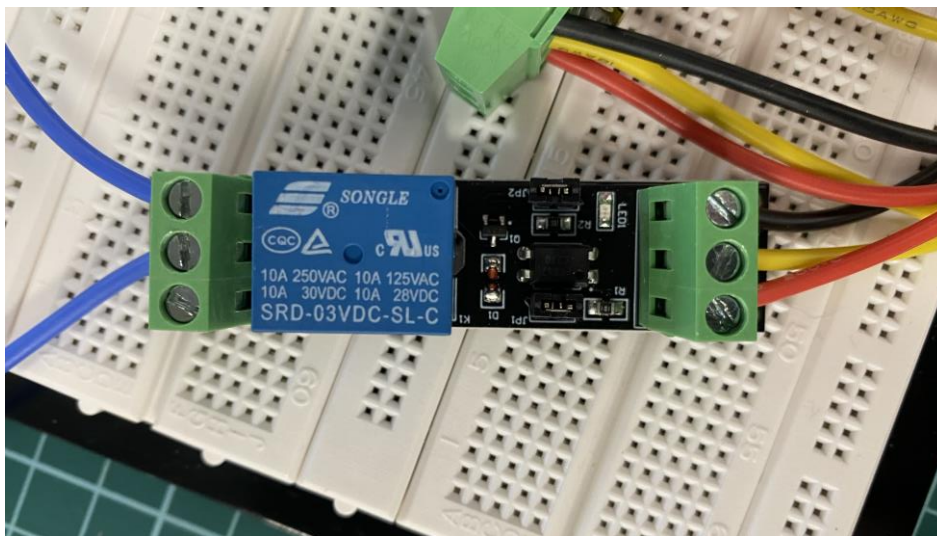


Figura 3-12 Relay 3DC.

Especificaciones:

Capacidad de cambio a 10A

UL, CUL, TUV reconocimiento

Material de plástico para alta temperatura y mejor rendimiento de la solución química

Voltajes nóminas en la bobina: 3V, 5V, 6V, 9V, 12V, 24V, 48V DC

Conexiones:

- DC+: 3V
- DC-: GND
- IN: *OUTPUT* señal
- COM: Puerto común

- NO: *Normally open*
- NC: *Normally close*

- **Sensor de corriente eléctrico INA219 [51].**

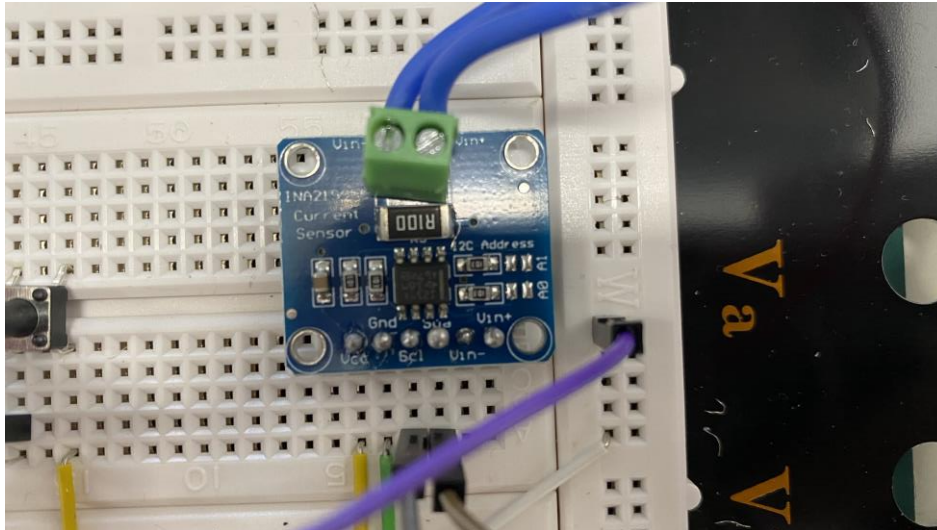


Figura 3-13 Sensor INA219.

Especificaciones:

Voltaje Bus: de 0 V a +26 V

Datos: Corriente, voltaje y potencia

16 direcciones programable (I2C)

Alto precisión: 0,5%

Voltaje de operación: de +3V a +5.5V

Temperatura de operación: -40 °C a +125 °C

- **Transductor de presión [52].**



Figura 3-14 Transductor de presión, Akozon Sensor de presión de acero inoxidable.

Especificaciones:

Material: Acero inoxidable

Voltaje de operación: 5V DC

Presión máxima: 100 psi (6,9 bar)

Entrada: 0-100 psi

Salida: 0,5V-4,5V lineal

Precisión: 2%

Rosca: 1/8" NPT

- **Sensor de Caudalímetro de Agua de 3,5-24 V [53]**

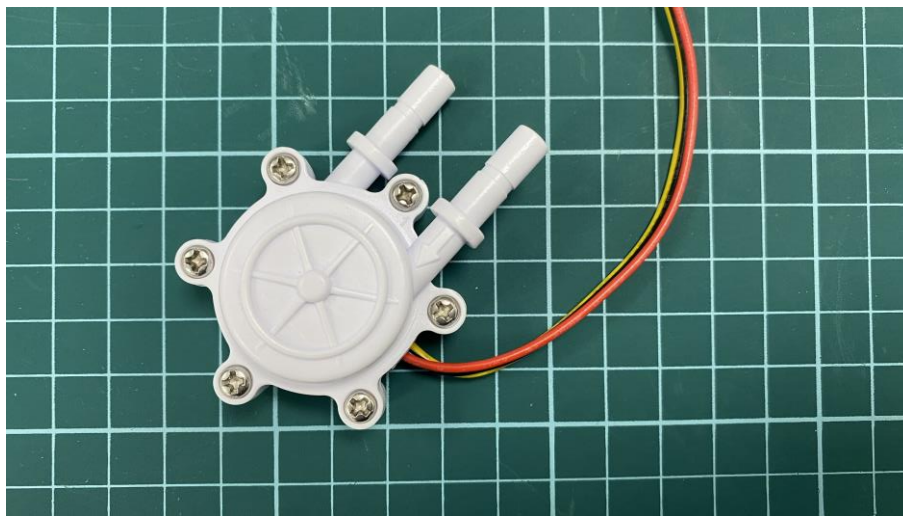


Figura 3-15 Sensor de Caudalímetro de Agua.

Especificaciones:

Material: *POM (Polyoxymethylene)*

Tamaño de interfaz: 1/4"

Rango de voltaje: DC 3,5-24 V

Rango de caudales: 0,2-8 l/min

Frecuencia de pulso: $F=(20 \times Q)\pm 5 \%$

Voltaje soportado: ≤ 1.75 mapa

Pulso de salida de alto nivel: $\geq$ DC 4.6 V (voltaje de entrada DC 5 V)

Nivel bajo de pulso de salida: $\leq$ DC 0.5 V (voltaje de entrada DC 5 V)

Corriente máxima de funcionamiento: 15 mA (DC 5 V)

Rango de temperatura de funcionamiento: ≤ 80 °C

- **Módulo de soporte de la alimentación de batería 18650 V3 [54].**

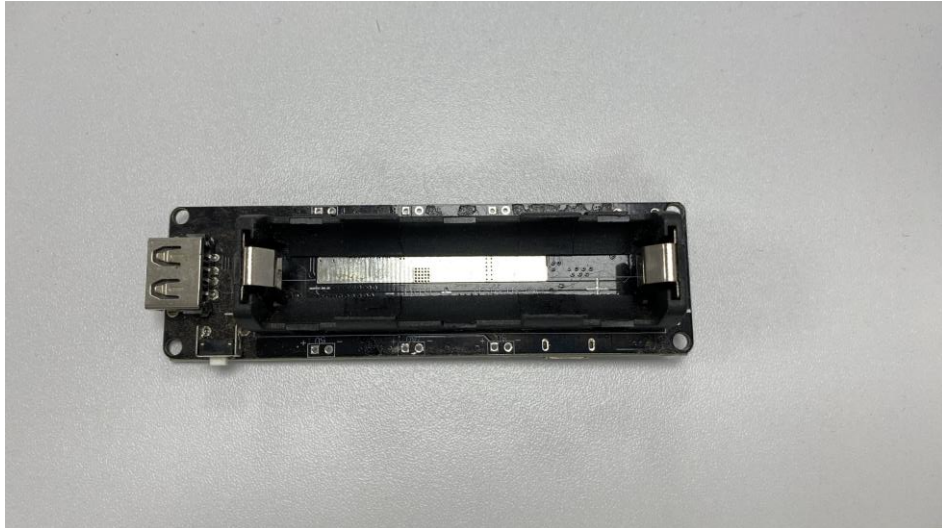


Figura 3-16 Soporte de expansión de batería.

Especificaciones:

Tamaño de la placa: 100x97x21 mm

Conector de entrada: Micro USB. Hasta 6.5V

Requerimiento de entrada: 5V constante, compatible para cargador 5V/1A o 3V/1A

Conector de salida: USB o conectores de expansión

Salida: 5V/1A o 3V/1A

Eficiencia de conversión: Hasta 95%

Temperatura de operación: -20 a 70 °C

3.3.2 Diseño propuesto

Para este diseño, se necesita conectar el microcontrolador ESP32 Devkit V1 con los tres relés correspondientes con la bomba sin escobillas, la válvula de 3 vías y la válvula de 2 vías a través del conversor de nivel lógico ya que el ESP32 proporciona la salida de 3,3 voltios y la mayoría de los componentes trabaja con la entrada de 5 voltios. El conversor de nivel lógico ayuda a convertir la tensión de 3,3 V de la placa ESP32 a la tensión de 5 V requerida por los relés SRD. El ESP32 proporciona la alimentación de 3,3 V de pin 3,3 V para el pin LV del conversor y también la alimentación de 5 V de pin Vin para el pin HV del conversor. Se alimenta el ESP32 por el módulo de escudo 18650 V3 a través del puerto mini USB.

3.4 Método de ensayo

Primero se comprueba que los componentes hidráulicos funcionan correctamente. Para ello, se usa un microcontrolador Arduino Uno para controlar los ensayos usando un botón para enviar la señal de empezar su función. El esquema se muestra en Figura 3-17 y el circuito real en Figura 3-18.

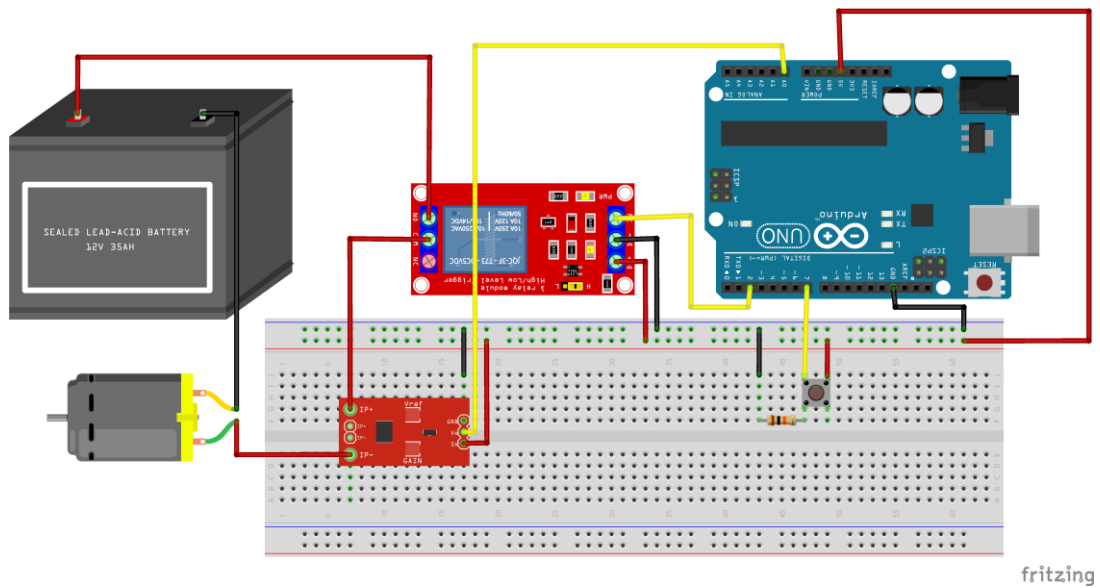


Figura 3-17 Esquema de ensayo con Arduino UNO.

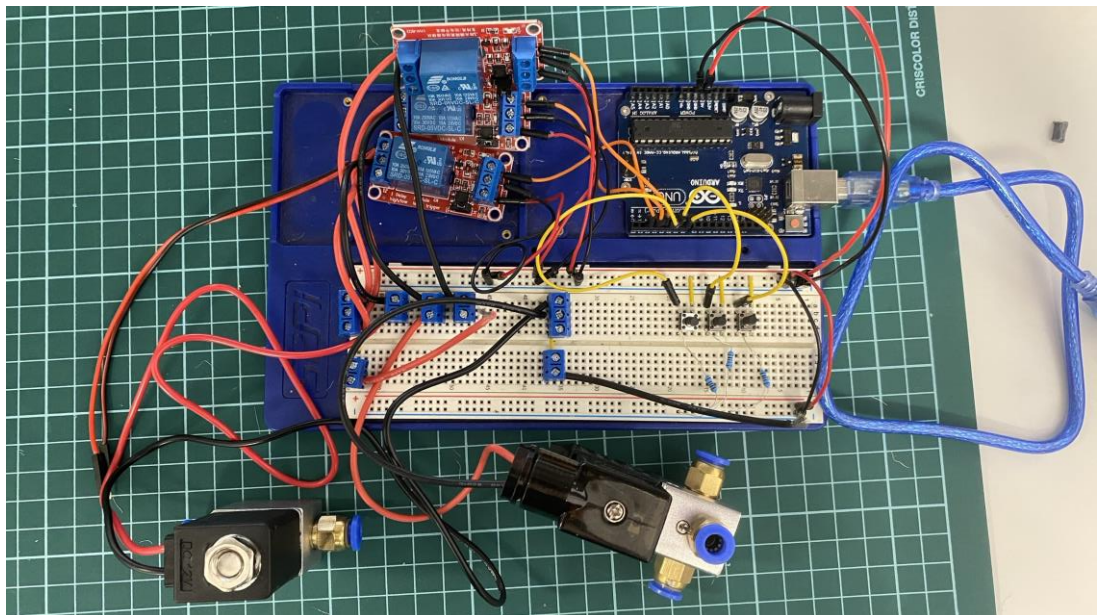


Figura 3-18 Ensayo de tres componentes juntos con Arduino UNO.

Observamos que el Arduino UNO ocupa mucho sitio, por lo que, se cambiar Arduino UNO por ESP32 Devkit V1 30 pins. En cambio, hay que usar un conversor de nivel lógico para que los componentes puedan funcionar correctamente en 5 voltios. Además, necesita la técnica de divisor de tensión para la señal analógica del sensor de corriente para que en la entrada se trabaje a máximo de 3,3 voltios. El esquema se muestra en Figura 3-19 y el circuito real en Figura 3-20.

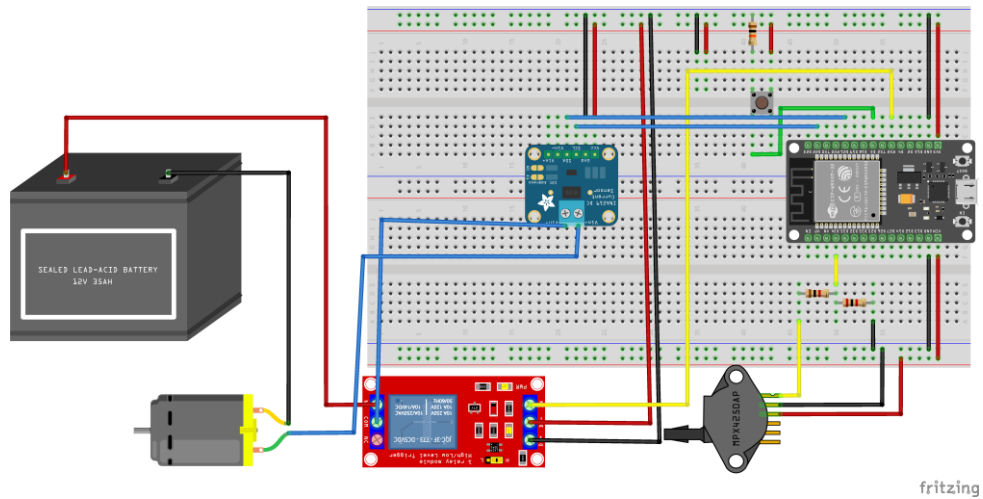


Figura 3-19 Esquema de ensayo con ESP32.

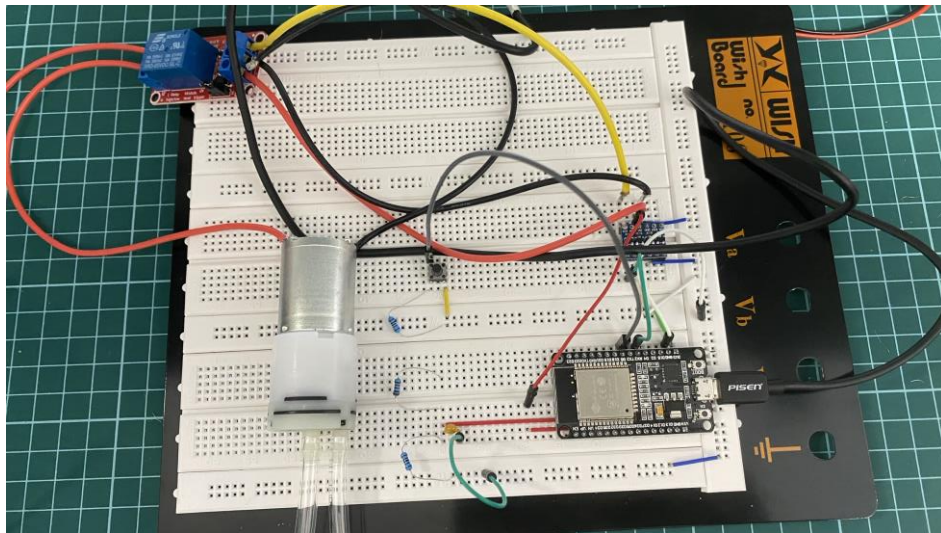


Figura 3-20 Ensayo de un componente individual con ESP32 Devkit V1.

Para la bomba de motor sin escobilla, se puede controlar su velocidad de giro a través de un ESC (*Electronic Speed Control*). Por lo que, en vez de usar un botón para enviar la señal de actuar, se usa un potenciómetro que puede mandar señal analógica para conseguir la velocidad que necesita. El esquema se demuestra en Figura 3-21.

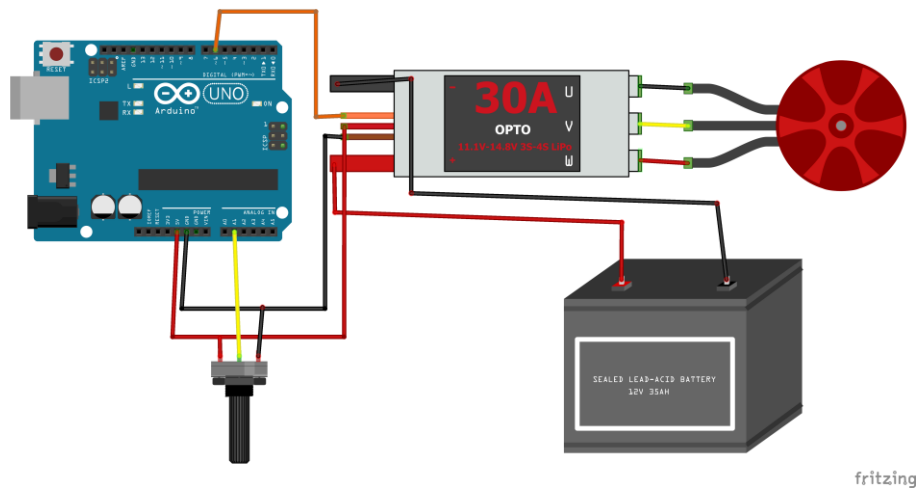


Figura 3-21 Esquema de ensayo para la bomba de motor sin escobilla con Arduino UNO.

En este trabajo se empleará un enfoque integrado para la obtención de datos relevantes. Se utilizará el base de datos SQLite, conectado a través de un servidor de mensajería MQTT (*Message Queuing Telemetry Transport*), el cual es un protocolo de transmisión de mensajes altamente eficiente para redes de IoT con ancho de banda limitado y es ampliamente empleado para redes con dispositivos remotos [55]. Por ello, se empleará la interfaz Wifi del ESP32 para conectar el dispositivo a un ordenador, con el objetivo de recopilar los datos necesarios a través del protocolo MQTT. De esta manera, se garantiza una recopilación de datos precisa y eficiente, que permita un análisis exhaustivo y detallado de la información relevante. Los experimentos se demuestran tal y como la Figura 3-22, Figura 3-23 y Figura 3-24.

Los datos que se necesitan son los siguientes:

- La relación entre consumo y caudal de la bomba usando agua pura
- La relación entre consumo y caudal de la bomba usando aceite
- La gráfica de la curva característica de la bomba para agua y aceite
- La relación entre consumo y la apertura de la válvula de 2 vías con la presión de agua
- La relación entre consumo y la apertura de la válvula de 2 vías con la presión de aceite
- La relación entre consumo y la apertura de la válvula de 3 vías con la presión de agua
- La relación entre consumo y la apertura de la válvula de 3 vías con la presión de aceite

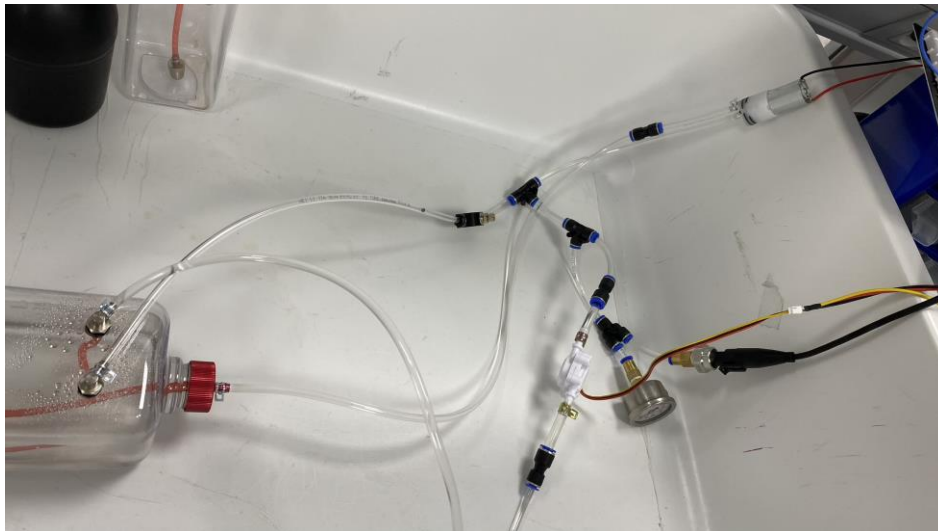


Figura 3-22 Circuito de ensayo con agua.

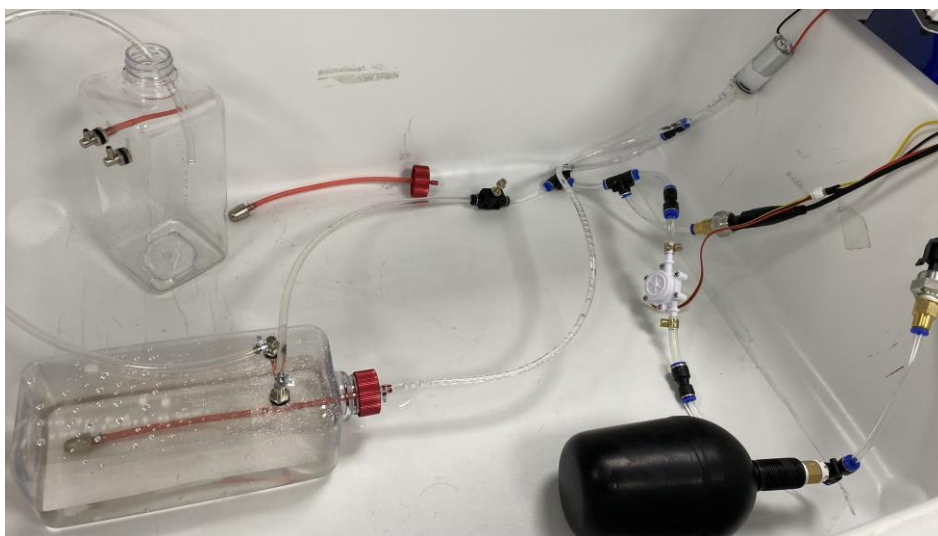


Figura 3-23 Circuito de ensayo con agua por trasferencia a una vejiga.

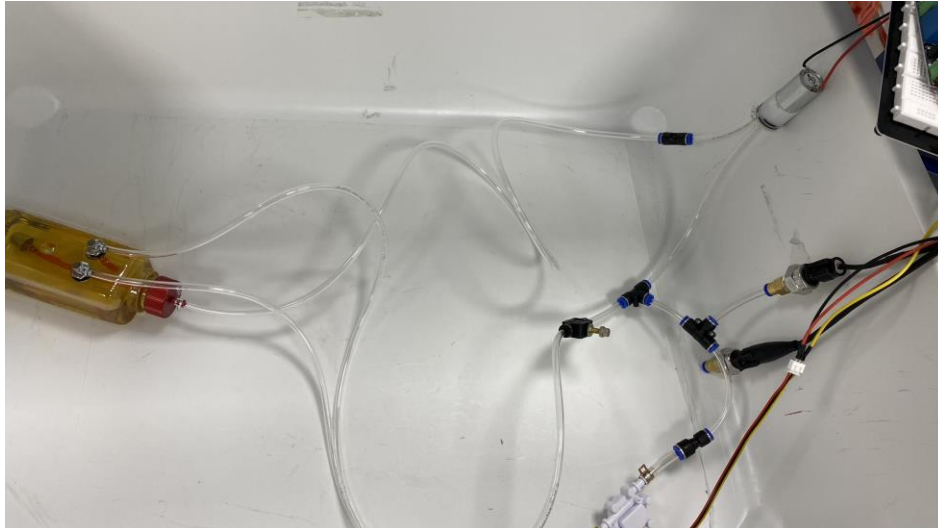


Figura 3-24 Circuito de ensayo con aceite.

El presente proyecto contempla la recopilación de datos mediante la utilización de una base de datos SQLite, la cual constituye una librería en lenguaje C diseñada para la gestión de proyectos de pequeño tamaño. La mencionada herramienta se caracteriza por su rapidez, autonomía, alta confiabilidad y funcionalidad integral. En la actualidad, SQLite es considerado el tipo de base de datos más utilizado a nivel mundial, lo cual confirma su eficacia y validez en el ámbito de la gestión de información [56]. Con la finalidad de proceder a la inserción de los datos recopilados en la mencionada base de datos, se emplea el código Python, el cual permite la adquisición de los datos enviados por servicio MQTT y su posterior almacenamiento en los campos preconfigurados de la base de datos. El código empleado para el ensayo puede encontrarse en el ANEXO III.

Posteriormente, se usa el servidor Grafana que es una plataforma de análisis y visualización de datos en tiempo real que permite a los usuarios crear paneles y gráficos interactivos para monitorear y analizar diversas métricas, datos y estadísticas provenientes de diferentes fuentes [57]. Esta herramienta de código abierto, compatible con múltiples bases de datos, facilita la representación gráfica y la interpretación de los datos a través de una interfaz intuitiva y altamente personalizable. Se demuestra los procesos y flujo de trabajo que pasa cada servicio como Figura 3-25. El código de inserción de los datos a la base de datos SQLite puede encontrarse en los ANEXO IV y en ANEXO V.

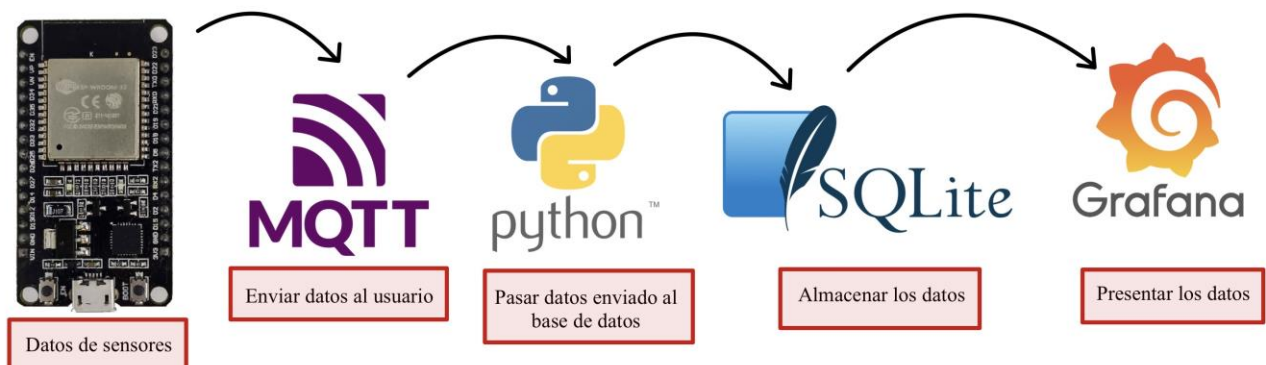


Figura 3-25 El flujo de trabajo y los servicios usados.

3.5 Soporte del sistema

3.5.1 Software para el diseño

En este proyecto, se han utilizado dos programas de Autodesk para el diseño del soporte del sistema de flotabilidad. Dado que el sistema tiene una forma general de cilindro, se utiliza AutoCAD para crear un perfil en 2D de cada capa como Figura 3-26, posteriormente fueron integrados en un modelo tridimensional a través de la herramienta de extrusión de Inventor para crear el modelo en 3D como Figura 3-27. De esta manera, se logra la alineación precisa de los agujeros para la conexión entre los componentes necesarios.

AutoCAD es un software de diseño asistido por computadora (*CAD: Computer Aid Design*) utilizado en ingeniería, arquitectura y diseño industrial. Permite crear dibujos en 2D y 3D, y ofrece varias herramientas para ayudar a los usuarios a diseñar, documentar y visualizar sus ideas. El programa incluye herramientas de dibujo, edición y modificación, así como herramientas de anotación y dimensionamiento que facilitan la creación de planos y diseños precisos. Además, AutoCAD cuenta con la capacidad de trabajar con capas, lo que permite a los usuarios organizar sus dibujos y diseños de manera eficiente.

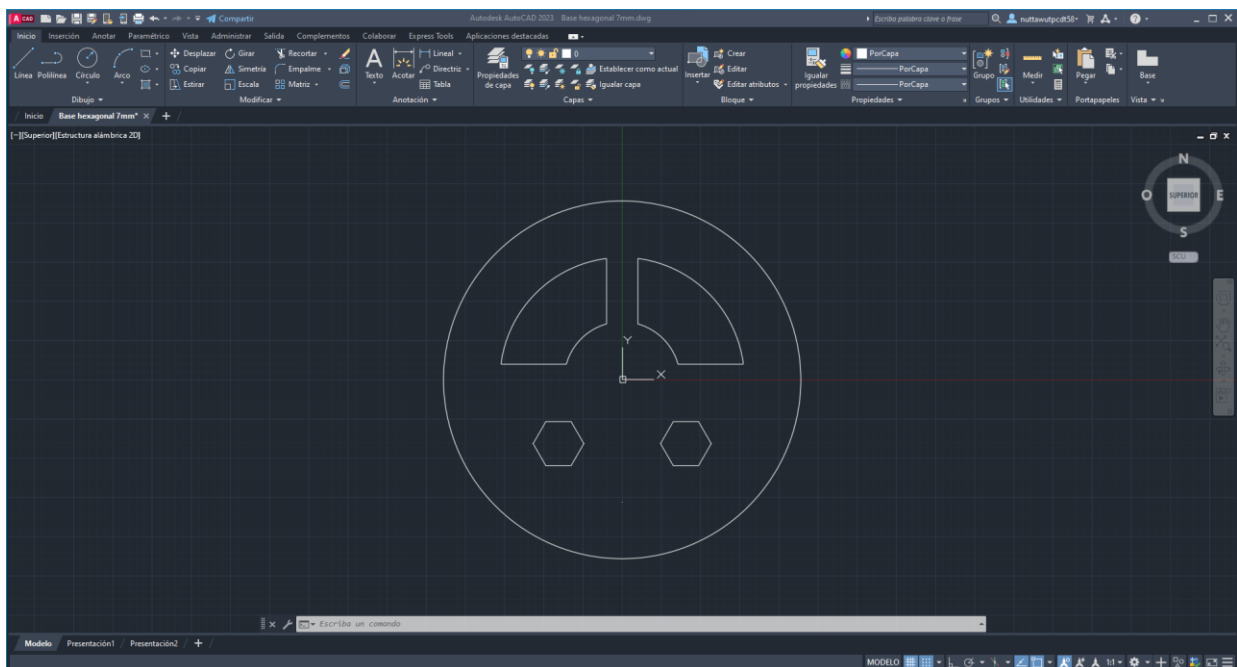


Figura 3-26 Hacer un perfil con AutoCAD 2023.

Inventor es una herramienta de modelado 3D que permite a los usuarios crear diseños detallados de piezas y ensamblajes de productos. Permite a los usuarios crear formas y geometrías complejas. También cuenta con herramientas de animación y simulación para ayudar a los usuarios a visualizar el funcionamiento de los diseños antes de la producción. Además, Inventor tiene una capacidad incorporada de análisis de elementos finitos (*FEM: Finite Element Method*), lo que significa que los usuarios pueden analizar y evaluar el comportamiento estructural de sus diseños de manera virtual antes de la fabricación del producto.

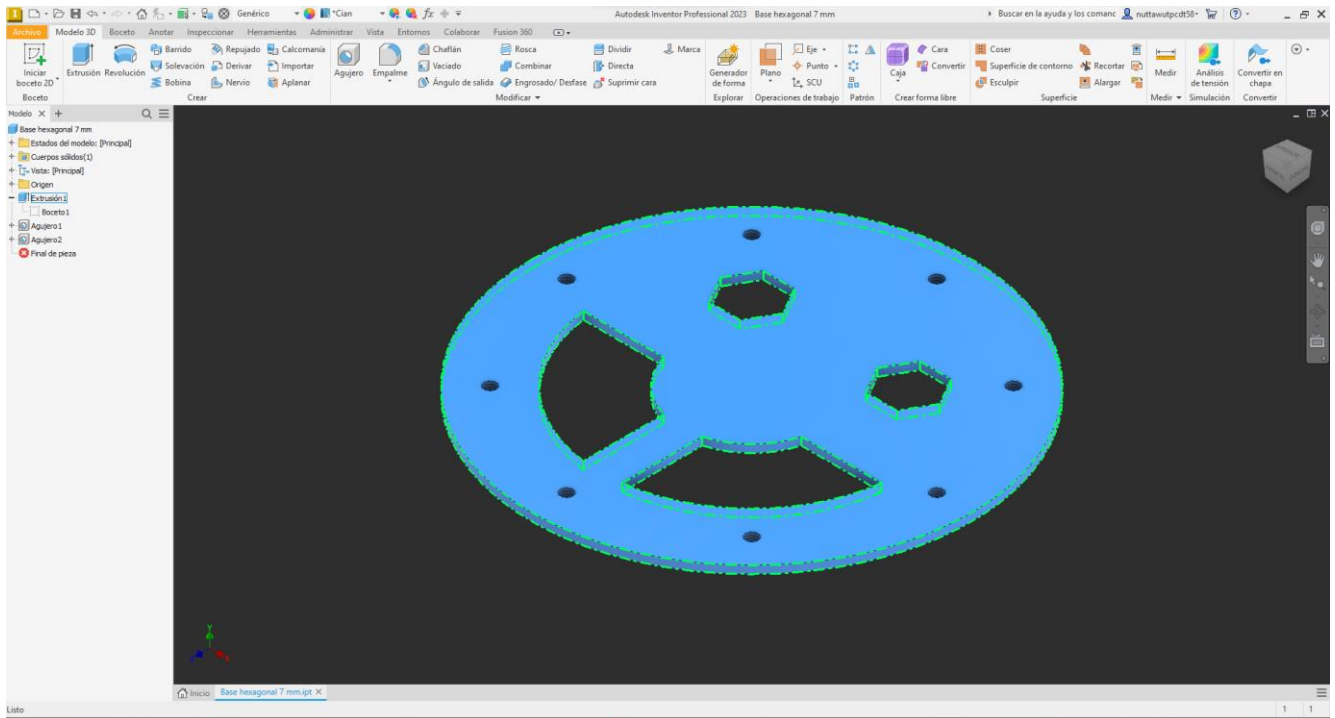


Figura 3-27 Extrusión de un perfil con Inventor profesional 2023.

3.5.2 Materiales para la impresión 3D

En el presente proyecto, el proceso de diseño se aborda considerando la necesidad de minimizar el uso de material de soporte para la impresión 3D, con el propósito de reducir su impacto ambiental y mejorar la eficiencia del proceso de producción. Por eso, se diseña la pieza que no necesita ninguno material de soporte. Sin embargo, se cuenta con una selección de materiales disponibles que permiten imprimir la pieza con las características requeridas, tal como se especifica en la Tabla 3-1.

	ABS	PLA	PETG	PP
Resistencia límite	40 MPa	65 MPa	53 MPa	32 MPa
Máxima temperatura de servicio	98 °C	52 °C	73 °C	100 °C
Densidad	1,04 g/cm ³	1,24 g/cm ³	1,23 g/cm ³	0,9 g/cm ³
Precio por kg	10-40 \$	10-40 \$	20-60 \$	60-120 \$
Flexible	No	No	No	Sí
Resistente de impacto	Sí	No	No	No
Resistente de agua	No	No	Sí	Sí

Tabla 3-1 Tabla de propiedades de filamentos [58].

Se las propiedades de los materiales disponibles, que el *Polypropylene* (PP) es el material óptimo para la fabricación de la pieza en cuestión, en términos de sus características físicas y químicas. No obstante, su elevado costo podría suponer un obstáculo en la implementación del proyecto. En consecuencia, se hace necesaria la búsqueda de alternativas.

En este sentido, es posible determinar que, en base a las propiedades de los cuatro materiales disponibles, tanto el *Acrylonitrile butadiene styrene* (ABS) como el *Polylactic acid* (PLA) cumplen con los requerimientos necesarios para el trabajo. Si bien es cierto que estos materiales presentan algunas limitaciones en comparación con el PP, su disponibilidad en el mercado y su menor costo resultan ventajosos para la implementación del proyecto.

Asimismo, es importante mencionar que el PLA es uno de los materiales más utilizados en el campo de la impresión 3D, lo cual puede representar una ventaja en términos de accesibilidad y disponibilidad en el mercado. En definitiva, se concluye que la utilización de materiales como el ABS o el PLA resulta una alternativa viable y económica sin comprometer la calidad y funcionalidad del producto final.

3.6 Sistema de control

3.6.1 Controlador PID

En el sistema dinámico planteado, se emplea una referencia de la profundidad deseada como entrada al sistema y la profundidad obtenida mediante sensores supone la salida obtenida. De esta forma, el algoritmo de control debe gestionar el caudal para poder generar un error cero entre la entrada deseada y la salida obtenida, para lo cual se puede aplicar un controlador clásico PID. El PID toma la desviación entre el valor deseado y el obtenido y le aplica operadores de integración y derivación, de forma que el algoritmo se puede expresar como.

$$u_i = K_p e + K_I \int_0^t e \, dt + K_D \dot{e} \quad (30)$$

Dónde, K_p , K_I y K_D son las ganancias proporcional, integral y diferencial respectivamente y el valor del error de la realimentación es $e = Z_R - Z$

La elección de los valores de K_p , K_D y K_I en un controlador PID se basa en el análisis de la dinámica del sistema a controlar y en la optimización de los criterios de desempeño. Ver la Tabla 3-2.

K_p es la ganancia proporcional que ajusta la magnitud de la corrección en función del error actual. Un valor alto de K_p puede hacer que el sistema sea rápido pero inestable, mientras que un valor bajo puede hacer que sea lento pero estable.

K_D es la ganancia derivativa que ajusta la magnitud de la corrección en función de la tasa de cambio del error. Un valor alto de K_D ayuda a reducir las oscilaciones y a estabilizar el sistema, pero también puede hacer que sea menos sensible a pequeños cambios en el error.

K_I es la ganancia integral que ajusta la magnitud de la corrección en función de la acumulación del error a lo largo del tiempo. Un valor alto de K_I puede hacer que el sistema sea más lento, pero más preciso, mientras que un valor bajo puede hacer que sea más rápido, pero menos preciso.

En general, los valores óptimos de K_p , K_D y K_I se determinan a través de un proceso de prueba y ajuste empírico, y dependen del sistema específico y de los objetivos de control deseados. El código empleado para el control del sistema de control puede encontrarse en el ANEXO VI.

CL RESPONSE	RISE TIME	OVERSHOOT	SETTLING TIME	S-S ERROR
K_p	Decrease	Increase	Small Change	Decrease
K_i	Decrease	Increase	Increase	Decrease
K_d	Small Change	Decrease	Decrease	No Change

Tabla 3-2 Efecto de los constantes K_p , K_i y K_d en el lazo cerrado [59].

3.6.2 Controlador MPC

MPC (*Model Predictive Control*) [60] es una técnica de control utilizada en sistemas dinámicos para alcanzar un objetivo deseado. La idea detrás de MPC es utilizar un modelo matemático del sistema para predecir cómo se comportará en el futuro, y luego utilizar esa información para calcular la mejor acción de control a tomar en el presente.

En MPC, se realiza una optimización en tiempo real para determinar la secuencia óptima de acciones de control a lo largo de un horizonte temporal futuro. Esto se hace mediante la minimización de una función objetivo que combina el desempeño deseado del sistema con restricciones en las acciones de control y en las variables del sistema.

MPC se ha utilizado en una variedad de aplicaciones [61], incluyendo control de procesos industriales, control de sistemas de transporte, control de energía y sistemas de control de robots. El código empleado para el control del sistema de control puede encontrarse en el ANEXO VII, y en el ANEXO VIII. En resumen, se debe definir un horizonte de simulación en base al parámetro $T_s = 0,02$ s y se reconstruye la estimación futura empleando un algoritmo simple de integración por el método de Euler, es decir:

$$\vec{x}_{k+1} = \vec{x}_k + \dot{\vec{x}}_{k+1} \cdot T_s \quad (31)$$

$$\vec{x}_{k+1} = (A + I) \cdot \vec{x}_k + B T_s \cdot \vec{u}_k \quad (32)$$

De esta forma puede escribirse:

$$A_d = A + I, B_d = B T_s, C_d = C$$

$$\vec{x}_{k+1} = A_d \vec{x}_k + B_d \vec{u}_k$$

$$\vec{y}_k = C_d \vec{x}_k$$

El vector de estados para el tiempo de predicción establecido se obtiene a partir de las siguientes expresiones:

$$\vec{x}_1 = A_d \vec{x}_0 + B_d \vec{u}_0$$

$$\vec{x}_2 = A_d \vec{x}_1 + B_d \vec{u}_1 = A_d^2 \vec{x}_0 + A_d B_d \vec{u}_0 + B_d \vec{u}_1$$

$$\vec{x}_3 = A_d \vec{x}_2 + B_d \vec{u}_2 = A_d^3 \vec{x}_0 + A_d^2 B_d \vec{u}_0 + A_d B_d \vec{u}_1 + B_d \vec{u}_2$$

·
·
·

$$\vec{x}_n = A_d^n \vec{x}_0 + [A_d^{n-1} B_d, A_d^{n-2} B_d, A_d^{n-3} B_d, \dots, A_d B_d, B_d] \begin{bmatrix} \vec{u}_0 \\ \vec{u}_1 \\ \vec{u}_2 \\ \vdots \\ \vec{u}_{n-2} \\ \vec{u}_{n-1} \end{bmatrix}$$

Se puede escribir en formato matricial como:

$$\begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vec{x}_3 \\ \vdots \\ \vec{x}_{n-1} \\ \vec{x}_n \end{bmatrix} = \begin{bmatrix} B_d & 0 & 0 & \dots & 0 & 0 \\ A_d B_d & B_d & 0 & \dots & 0 & 0 \\ A_d^2 B_d & A_d B_d & B_d & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ A_d^{n-2} B_d & A_d^{n-3} B_d & A_d^{n-4} B_d & \dots & B_d & 0 \\ A_d^{n-1} B_d & A_d^{n-2} B_d & A_d^{n-3} B_d & \dots & A_d B_d & B_d \end{bmatrix} \begin{bmatrix} \vec{u}_0 \\ \vec{u}_1 \\ \vec{u}_2 \\ \vdots \\ \vec{u}_{n-2} \\ \vec{u}_{n-1} \end{bmatrix} + \begin{bmatrix} A_d \\ A_d^2 \\ A_d^3 \\ \vdots \\ A_d^{n-1} \\ A_d^n \end{bmatrix} \vec{x}_0$$

$$\vec{x}_G = \bar{\bar{C}}\vec{u}_G + \hat{A}\vec{x}_0 \quad (33)$$

Dónde $\vec{x}_G$ es el vector de los valores de estados de la predicción, $\vec{u}_G$ es el vector de las entradas de cada momento, $\vec{x}_0$ es el estado inicial y $\bar{\bar{C}}, \hat{A}$ son matrices de coeficientes.

Se define la ecuación cuadrática del coste como:

$$J = \frac{1}{2} \vec{e}_{k+n}^T S \vec{e}_{k+n} + \frac{1}{2} \sum_{i=0}^{n-1} \vec{e}_{k+i}^T Q \vec{e}_{k+i} + \vec{u}_{k+i-1}^T R \vec{u}_{k+i-1} \quad (34)$$

Dónde J es la función cuadrática de los costes, S es la matriz de peso del último error, Q es la matriz de peso de los errores del resto, R es la matriz de peso de las entradas, $\vec{e}$ es el error de la referencia

$$\vec{e}_k = \vec{r}_k - \vec{y}_k \quad (35)$$

Dónde $\vec{r}_k$ es la referencia en k y $\vec{y}_k$ es la salida en k.

Se sustituye (17) en (16) considerando que el sistema no es afectado por valores de desfase (off-set).

$$J = \left(\frac{1}{2} \vec{r}_{k+n}^T S \vec{r}_{k+n} + \frac{1}{2} \sum_{i=0}^{n-1} \vec{r}_{k+i}^T Q \vec{r}_{k+i} \right) + \left(\frac{1}{2} \vec{x}_{k+n}^T C_d^T S C_d \vec{x}_{k+n} + \frac{1}{2} \sum_{i=0}^{n-1} \vec{x}_{k+i}^T C_d^T Q C_d \vec{x}_{k+i} - \vec{r}_{k+n}^T S C_d \vec{r}_{k+n} - \sum_{i=0}^{n-1} \vec{r}_{k+i}^T Q C_d \vec{x}_{k+i} + \frac{1}{2} \sum_{i=0}^{n-1} \vec{u}_{k+i-1}^T R \vec{u}_{k+i-1} \right)$$

Dónde $\frac{1}{2} \vec{r}_{k+n}^T S \vec{r}_{k+n} + \frac{1}{2} \sum_{i=0}^{n-1} \vec{r}_{k+i}^T Q \vec{r}_{k+i}$ es el desfase de la función, por lo que se extrae de la función

$$J' = \frac{1}{2} \begin{bmatrix} \vec{x}_k \\ \vdots \\ \vec{x}_{k+n} \end{bmatrix}^T \begin{bmatrix} C_d^T Q C_d & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C_d^T S C_d \end{bmatrix} \begin{bmatrix} \vec{x}_k \\ \vdots \\ \vec{x}_{k+n} \end{bmatrix} - \begin{bmatrix} \vec{r}_k \\ \vdots \\ \vec{r}_{k+n} \end{bmatrix}^T \begin{bmatrix} Q C_d & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & S C_d \end{bmatrix} \begin{bmatrix} \vec{x}_k \\ \vdots \\ \vec{x}_{k+n} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \vec{u}_{k-1} \\ \vdots \\ \vec{u}_{k+n-1} \end{bmatrix}^T \begin{bmatrix} R & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R \end{bmatrix} \begin{bmatrix} \vec{u}_{k-1} \\ \vdots \\ \vec{u}_{k+n-1} \end{bmatrix}$$

, dónde

$$\bar{\bar{Q}} = \begin{bmatrix} C_d^T Q C_d & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C_d^T S C_d \end{bmatrix}, \bar{\bar{T}} = \begin{bmatrix} Q C_d & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & S C_d \end{bmatrix}, \bar{\bar{R}} = \begin{bmatrix} R & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R \end{bmatrix}$$

Finalmente se obtiene:

$$J' = \frac{1}{2} \vec{x}_G^T \bar{\bar{Q}} \vec{x}_G - \vec{r}_G^T \bar{\bar{T}} \vec{x}_G + \frac{1}{2} \vec{u}_G^T \bar{\bar{R}} \vec{u}_G \quad (36)$$

Se sustituye (33) en (36) considerando que no le importa los valores de desfase.

$$J' = \frac{1}{2} \vec{u}_G^T (\bar{\bar{C}}^T \bar{\bar{Q}} \bar{\bar{C}} + \bar{\bar{R}}) \vec{u}_G + \vec{x}_0^T \hat{A}^T \bar{\bar{Q}} \bar{\bar{C}} \vec{u}_G - \vec{r}_G^T \bar{\bar{T}} \bar{\bar{C}} \vec{u}_G - \vec{r}_G^T \bar{\bar{T}} \hat{A} \vec{x}_0$$

Dónde $-\vec{r}_G^T \bar{\bar{T}} \hat{A} \vec{x}_0$ es el desfase y dado que $\bar{\bar{H}} = \bar{\bar{C}}^T \bar{\bar{Q}} \bar{\bar{C}} + \bar{\bar{R}}$, entonces.

$$J'' = \frac{1}{2} \vec{u}_G^T \bar{H} \vec{u}_G + [\vec{x}_0^T \quad \vec{r}_G^T] \begin{bmatrix} \hat{A}^T \bar{Q} \bar{C} \\ -\bar{T} \bar{C} \end{bmatrix} \vec{u}_G$$

Dado $\bar{F}^T = \begin{bmatrix} \hat{A}^T \bar{Q} \bar{C} \\ -\bar{T} \bar{C} \end{bmatrix}$, se cumple.

$$J'' = \frac{1}{2} \vec{u}_G^T \bar{H} \vec{u}_G + [\vec{x}_G^T \quad \vec{r}_G^T] \bar{F}^T \vec{u}_G \quad (37)$$

Se calcula el gradiente para obtener los valores de la entrada que permite, para la función de coste, obtener unos resultados mínimos.

$$\begin{aligned} \nabla \cdot J'' &= \bar{H} \vec{u}_G + \bar{F} \begin{bmatrix} \vec{x}_G^T \\ \vec{r}_G^T \end{bmatrix} = 0 \\ \vec{u}_G &= -\bar{H}^{-1} \bar{F} \begin{bmatrix} \vec{x}_G^T \\ \vec{r}_G^T \end{bmatrix} \end{aligned} \quad (38)$$

4 RESULTADOS Y VALIDACIÓN EXPERIMENTAL

4.1 Resultados de los ensayos

4.1.1 Ensayo de la bomba en vacío

Mediante el procedimiento del ensayo de la bomba en vacío, en el cual se enciende la bomba durante un periodo de tiempo de 20 segundos, es posible obtener una gráfica de intensidad-tiempo. Dicha gráfica es presentada utilizando el servicio Grafana, tal como se ilustra en la Figura 4-1

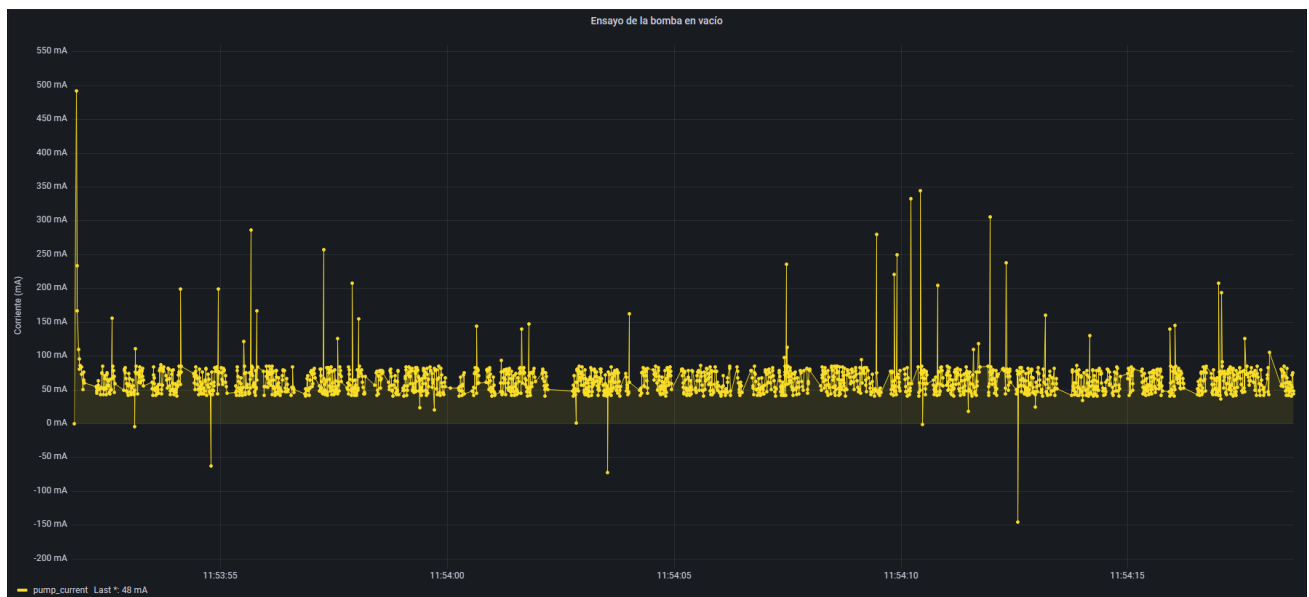


Figura 4-1 Ensayo de la bomba en vacío: la gráfica de intensidad-tiempo.

En particular, al centrarnos en el momento de arranque de la bomba, se puede observar un periodo de 1 segundo, como se muestra en la Figura 4-2. En este periodo, se observa que la bomba consume una intensidad pico de alrededor de 500 mA, para luego estabilizarse en la intensidad nominal tras 0,2 segundos aproximadamente. Además, destaca que el valor medio de intensidad durante los 20 segundos de operación es de 64 mA.

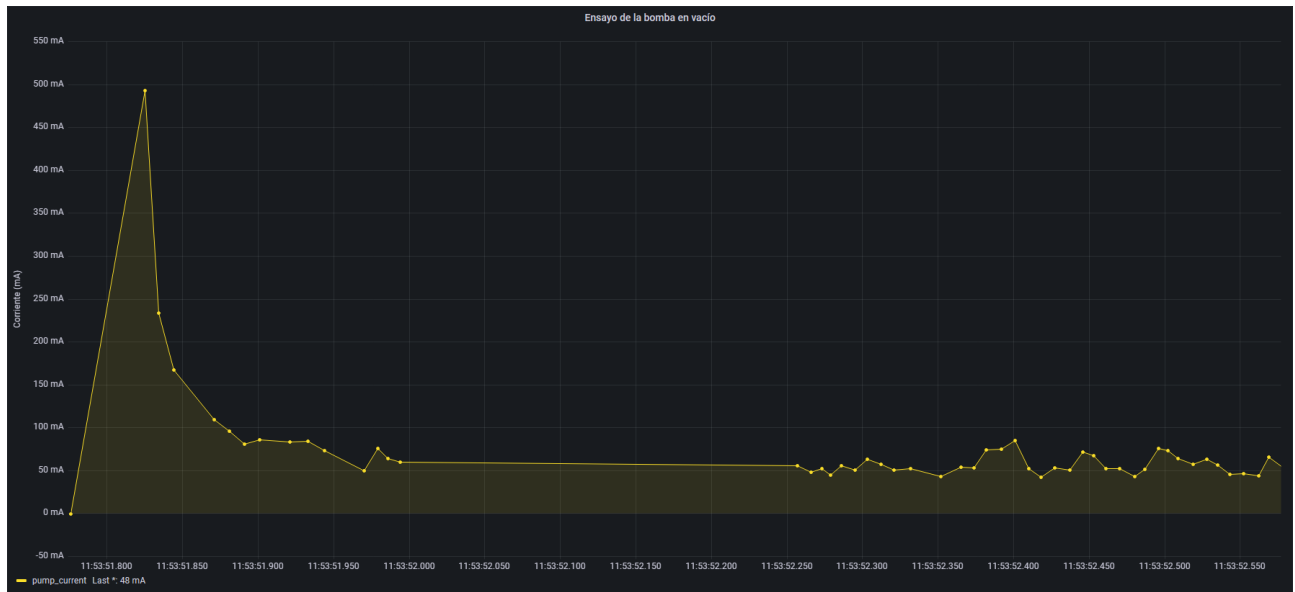


Figura 4-2 Ensayo de la bomba en vacío: arranque.

4.1.2 Ensayo de la válvula de 3 vías en vacío

Mediante el procedimiento del ensayo de la válvula de 3 vías en vacío, en el cual se enciende la válvula durante un periodo de tiempo de 20 segundos, se obtiene una gráfica de intensidad-tiempo tal y como se ilustra en la Figura 4 3. La válvula consume una intensidad en el tiempo de arranque de 470 mA y se sigue reduciendo ligeramente durante el momento de apertura.

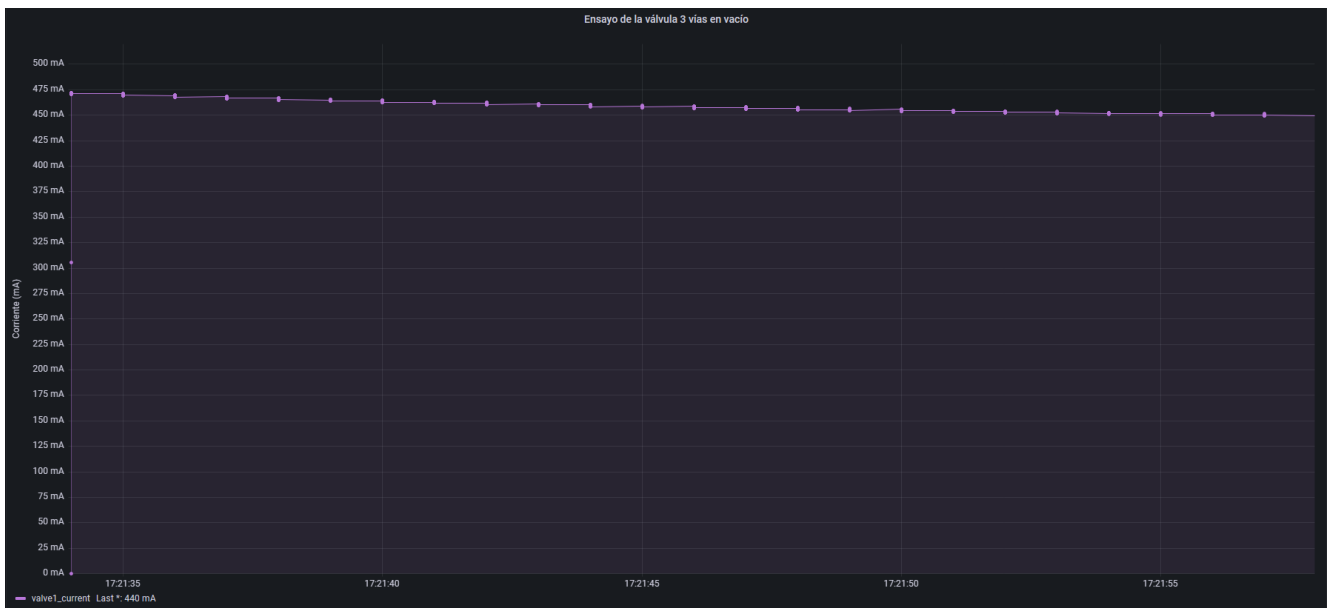


Figura 4-3 Ensayo de la válvula de 3 vías en vacío.

4.1.3 Ensayo de la válvula de 2 vías en vacío

Mediante el procedimiento del ensayo de la válvula de 2 vías en vacío, en el cual se enciende la válvula durante un periodo de tiempo de 20 segundos, se obtiene una gráfica de intensidad-tiempo tal y como se ilustra en la Figura 4-4. La válvula consume la intensidad en el tiempo de arranque 500 mA y se sigue reduciendo ligeramente durante el momento de apertura.

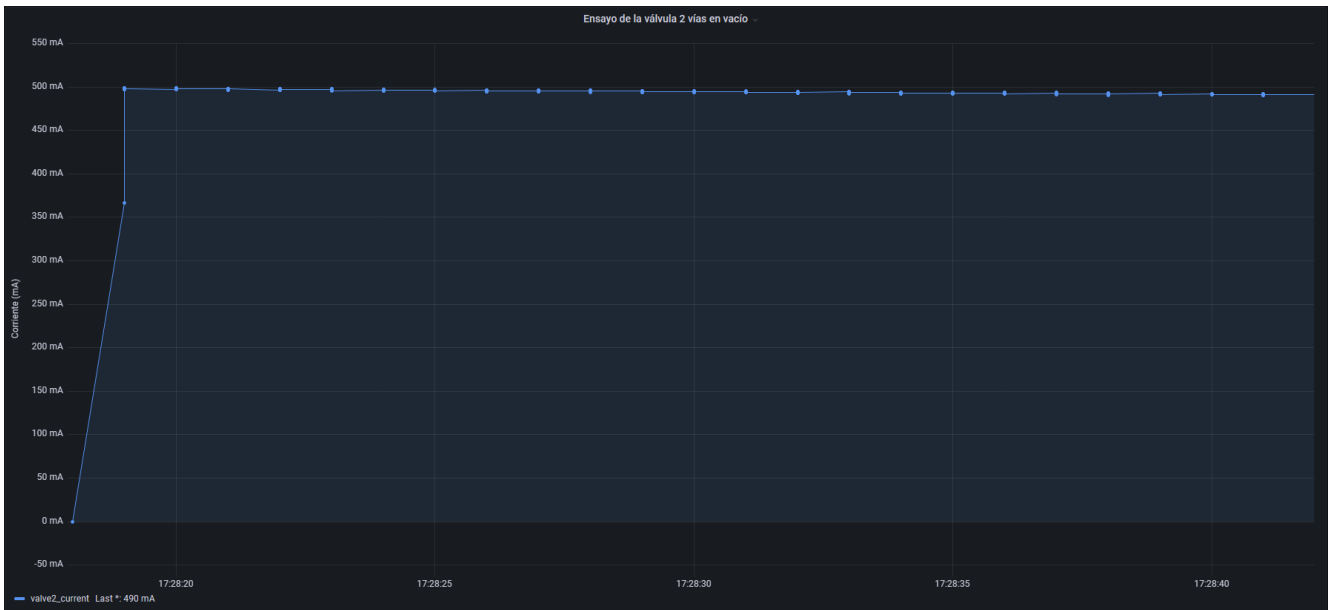


Figura 4-4 Ensayo de la válvula de 2 vías en vacío.

4.1.4 Curvas características de la bomba con agua

En el presente experimento, se han seleccionado datos de caudal en una serie de pruebas que implicaron la variación de la alimentación eléctrica de una bomba, junto con la regulación de la válvula reguladora para lograr distintos niveles de presión del sistema. Los resultados de estas pruebas fueron representados en forma de gráficas que exhibieron una curva de regresión de polinomio de grado 2 como muestra en Figura 4-5.

Se pudo observar que el caudal registrado aumentó a medida que se incrementaba la alimentación eléctrica suministrada a la bomba. Esto indica que la potencia de la bomba es un factor clave en la determinación del caudal del sistema. De esta manera, se ha demostrado empíricamente la relación directa entre la potencia suministrada y el caudal obtenido en el sistema.

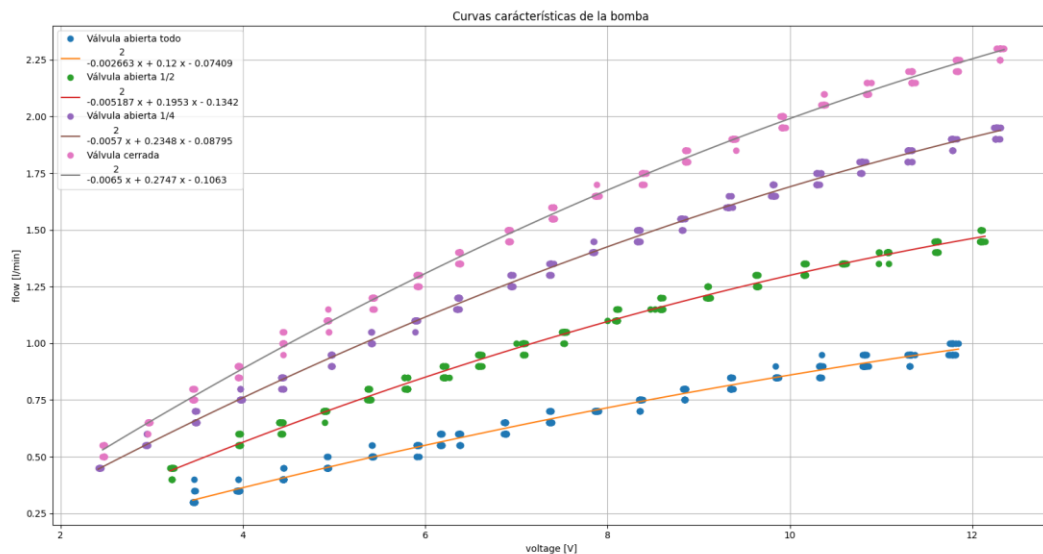


Figura 4-5 Curvas características de la bomba con agua.

4.1.5 Curvas características de la bomba con agua por trasferencia a una vejiga

En el presente experimento, se ha fijado la alimentación eléctrica de la bomba a una tensión constante de 12 V. Se toma datos de caudal y presión de la vejiga en una serie de pruebas que se realizaron hasta que se agotó el agua del depósito. Durante el transcurso de las pruebas, se pudo observar que la presión dentro de la vejiga aumentaba a medida que se utilizaba el sistema, lo que tuvo un efecto negativo sobre el caudal de agua que se registraba en el sistema. De hecho, se observó que el caudal disminuía gradualmente hasta que llegaba el momento en que se alcanzaba un caudal nulo. Este fenómeno se puede mostrar una curva de regresión polinómica de grado 2 que evidenciaba una disminución lineal en el caudal en función de la presión generada en la vejiga como muestra en Figura 4-6.

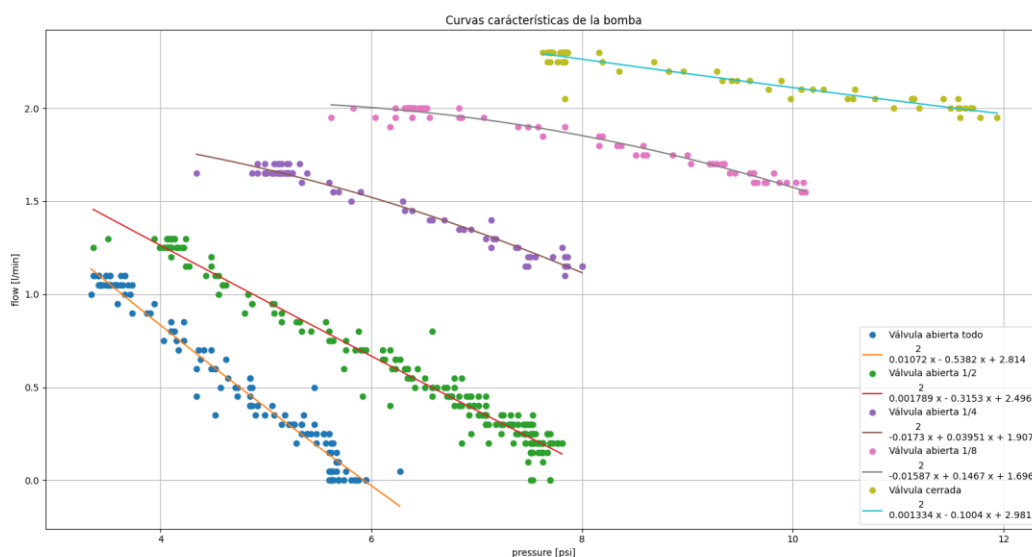


Figura 4-6 Curvas características de la bomba con agua por trasferencia a una vejiga.

4.2 Soporte del sistema

4.2.1 Módulos principales

Este soporte se encuentra dividido en seis módulos principales como muestra en la Figura 4-7, los cuales se han organizado de forma interna a externa como sigue: depósito de aceite, conjunto de la placa electrónica y sistema de control, soporte de la bomba, soporte de válvulas y antirretornos, conjunto de tuberías y sensores, y finalmente, la tapa de la estanqueidad, la cual cuenta con una conexión destinada a la vejiga. Los planos de los módulos se emplean en el ANEXO IX.

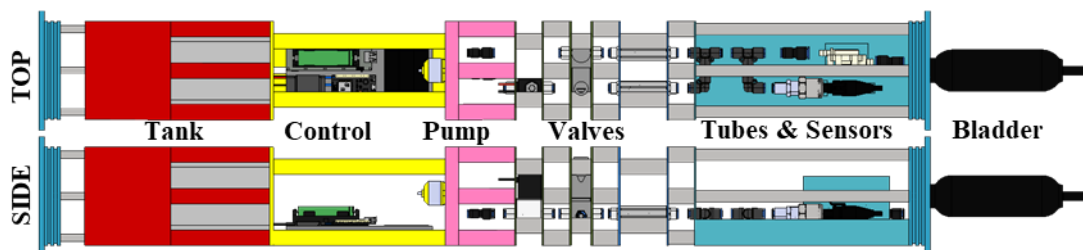


Figura 4-7 Soporte del sistema de flotabilidad con los componentes.

4.2.2 Depósito del aceite

El soporte del depósito se trata de un soporte para depósito con medidas de 78 mm x 78 mm y una profundidad adecuada para evitar su movimiento en el depósito. Además, cuenta con cuatro barras que

permiten fijarla a otro módulo mediante agujeros roscados tipo M4, lo que asegura su estabilidad y sujeción. Asimismo, cuenta con cuatro roscas de tipo BSP 1/8 para conectar con otros sistemas y dos agujeros para facilitar el paso de cables de conexión como muestra en la Figura 4-8.

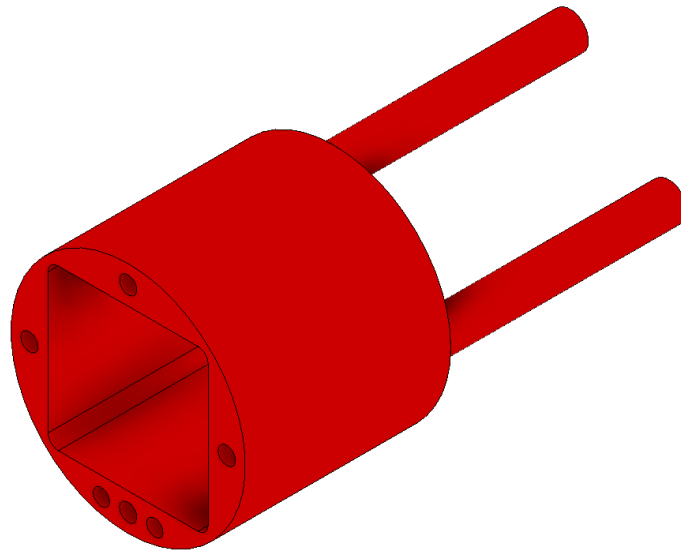


Figura 4-8 Depósito del aceite.

4.2.3 Conjunto de la placa electrónica y sistema de control

El soporte del conjunto de la placa electrónica y sistema de control es una base plana diseñada para soportar una placa electrónica. Cuenta con dos barras y cuatro agujeros roscados para su fijación a otro módulo mediante agujeros tipo M4, lo que asegura su estabilidad y sujeción. Además, su diseño incluye un hueco en la mitad de la base que permite enroscar y fijarla con otro módulo de forma fácil y sencilla, como muestra en la Figura 4-9.

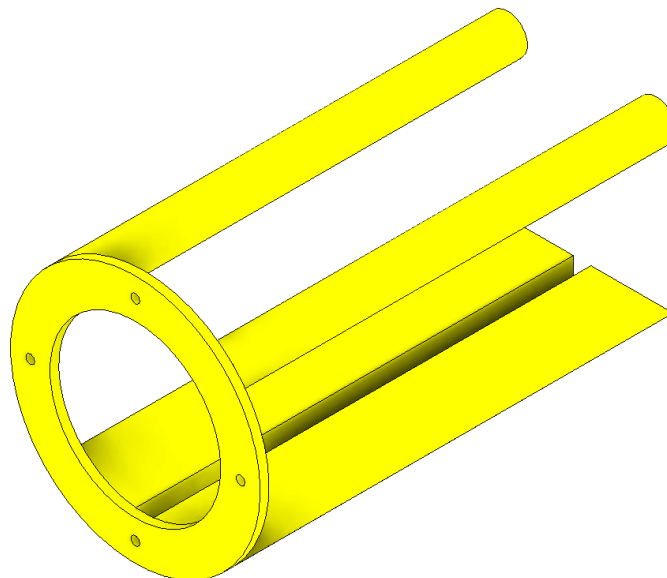


Figura 4-9 Conjunto de la placa electrónica y sistema de control.

4.2.4 Soporte de la bomba

El soporte de la bomba es especialmente diseñado para fijar la bomba de motor sin escobillas en el sistema. Su forma se adapta perfectamente a la estructura de la bomba, asegurando su sujeción y estabilidad. Además, tiene cuatro barras que permiten su fijación a otro módulo mediante agujeros tipo M4, lo que garantiza su correcta instalación en el sistema. Hay huecos para el paso de cables y tubos hidráulicos, lo que facilita su conexión y el correcto funcionamiento del sistema en su conjunto como muestra en la Figura 4-10.

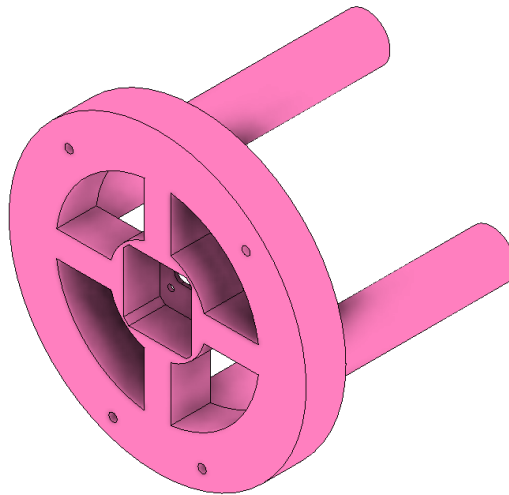


Figura 4-10 Soporte de la bomba.

4.2.5 Soporte de válvulas y antirretornos

Los soportes de válvulas y antirretornos son especialmente diseñados para conectar y fijar las válvulas del sistema hidráulico. Cuenta con dos agujeros hexagonales que permiten apoyar los conectores de las válvulas y el paso de los tubos hidráulicos, los cuales están perfectamente alineados para asegurar el correcto flujo de fluido. Además, dispone de ocho agujeros tipo M4 que permiten la fijación de barras en cruz y diagonal de manera alternativa, facilitando su montaje y desmontaje. También cuenta con huecos para el paso de cables y tubos hidráulicos, lo que simplifica su conexión y asegura su correcto funcionamiento en el sistema como muestra en la Figura 4-11.

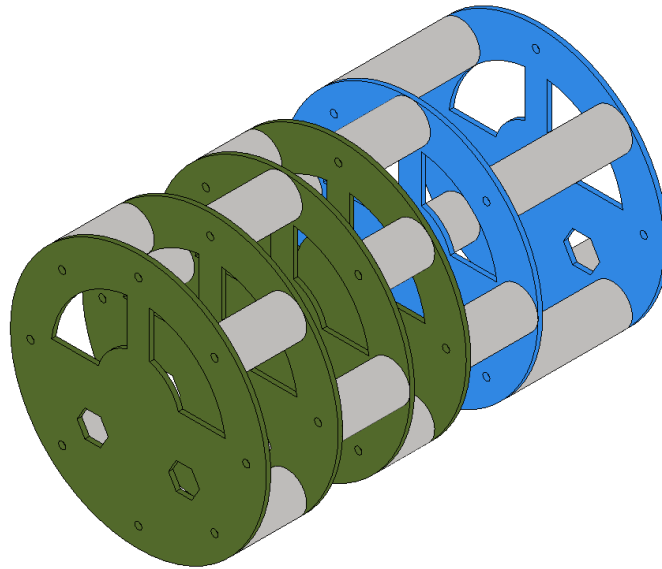


Figura 4-11 Soporte de válvulas y antirretornos.

4.2.6 Conjunto de tuberías y sensores

El soporte del conjunto de tuberías y sensores es una base plana especialmente diseñada para la fijación de sensores y tuberías en el sistema hidráulico. Dispone de dos agujeros hexagonales que permiten apoyar los conectores de las tuberías y el paso de los tubos hidráulicos, los cuales están perfectamente alineados para asegurar el correcto flujo de fluido. Además, cuenta con cinco agujeros tipo M4 y un agujero de M18 para la fijación de barras en cruz y diagonal de manera alternativa, lo que facilita su montaje y desmontaje. También dispone de huecos para el paso de cables y tubos hidráulicos, lo que simplifica su conexión y asegura su correcto funcionamiento en el sistema como muestra en la Figura 4-12.

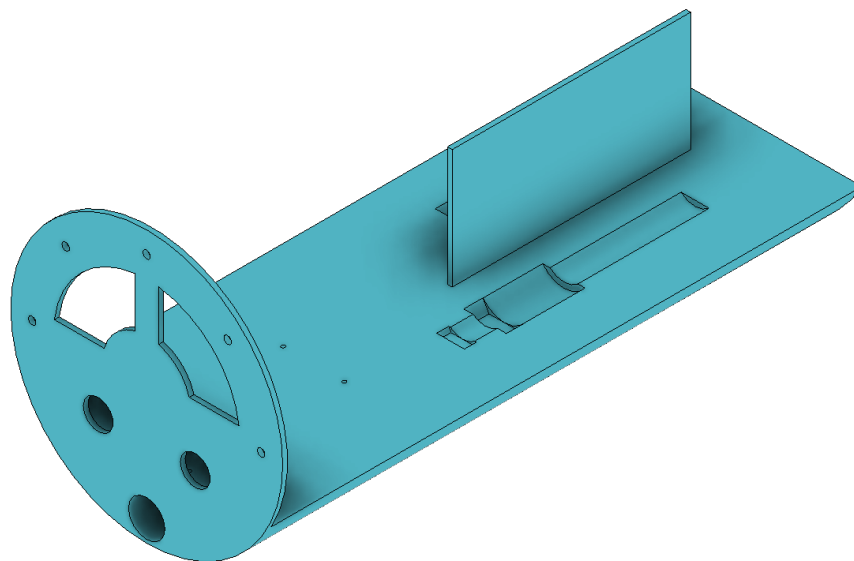


Figura 4-12 Conjunto de tuberías y sensores.

4.2.7 Tapa final

La tapa final es una tapa de estanqueidad con conexión para la vejiga. Dispone de tres agujeros tipo M15 para su fijación en el sistema, garantizando su estabilidad y seguridad. Además, cuenta con un hueco de M24, cuyo tamaño de rosca exterior permite la conexión de la vejiga al sistema hidráulico. La pieza también cuenta con huecos entre capas que aseguran su estanqueidad, evitando cualquier tipo de fuga de fluido en el sistema hidráulico como muestra en la Figura 4-13.

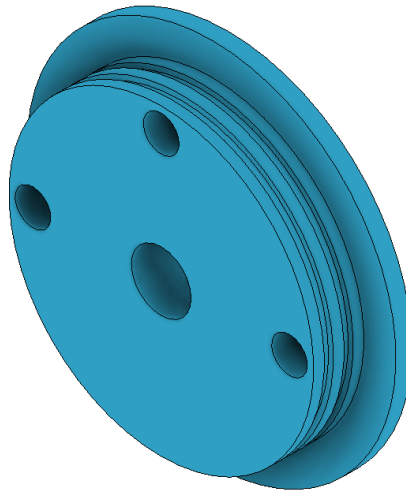


Figura 4-13 Tapa final.

4.3 Simulador de movimiento

4.3.1 Valores iniciales de control

Se simula el movimiento vertical de un planeador submarino de forma cilíndrica que tiene las características siguientes:

Parámetros	Valor	Unidad
Masa (m)	20	kg
Diámetro (D)	120	mm
Longitud (l)	1000	mm
Profundidad inicial (Z)	0	m
Velocidad inicial ($\dot{z}$)	0	m/s
Aceleración inicial ($\ddot{z}$)	0	m/s ²

Tabla 4-1 Parámetros del planeador submarino simulado.

Se considera un ambiente de simulación definido por los siguientes valores:

Parámetros	Valor	Unidad
Gravedad (g)	9.81	m/s ²
Densidad de agua (ρ_{fw})	1025	kg/m ³
Densidad de aceite (ρ_{oil})	885	kg/m ³
Coefficiente de resistencia en el agua (C_D)	0,4	-

Tabla 4-2 Parámetros del ambiente simulado.

Además, tiene las consideraciones siguientes:

- El flujo de aceite del sistema es controlable.
- No hay perturbación externa.
- Se ignora los cambios de las densidades de aceite y aire por el cambio de la temperatura.
- Solo considera movimiento pesantez de los 6DOF, los demás no serán considerados
- Se azume objeto como un punto.
- La presión en el agua solo varía por la profundidad
- Simulación de la bajada hasta la profundidad 150 metros.

4.3.2 Resultados gráficos

- Control PID

Parámetros de control

Parámetros	Valor
K_P	0,022
K_I	0,00005
K_D	0,02

Tabla 4-3 Parámetros PID simulado.

Dado que el controlador PID se fundamenta en la medición del error de profundidad, se puede observar que el planeador submarino desciende hasta sobrepasar ligeramente el valor de referencia establecido, para posteriormente iniciar el proceso de corrección de su posición, tal y como se detalla en la Figura 4-14. Es importante mencionar que, una vez que se inicia la corrección de la profundidad, se puede notar que la trayectoria del planeador submarino experimenta pequeñas oscilaciones en su desplazamiento, oscilaciones que oscilan alrededor de 1 metro, como resultado de la influencia combinada del efecto de la deriva y la integración del error, tal como se visualiza en la Figura 4-15.

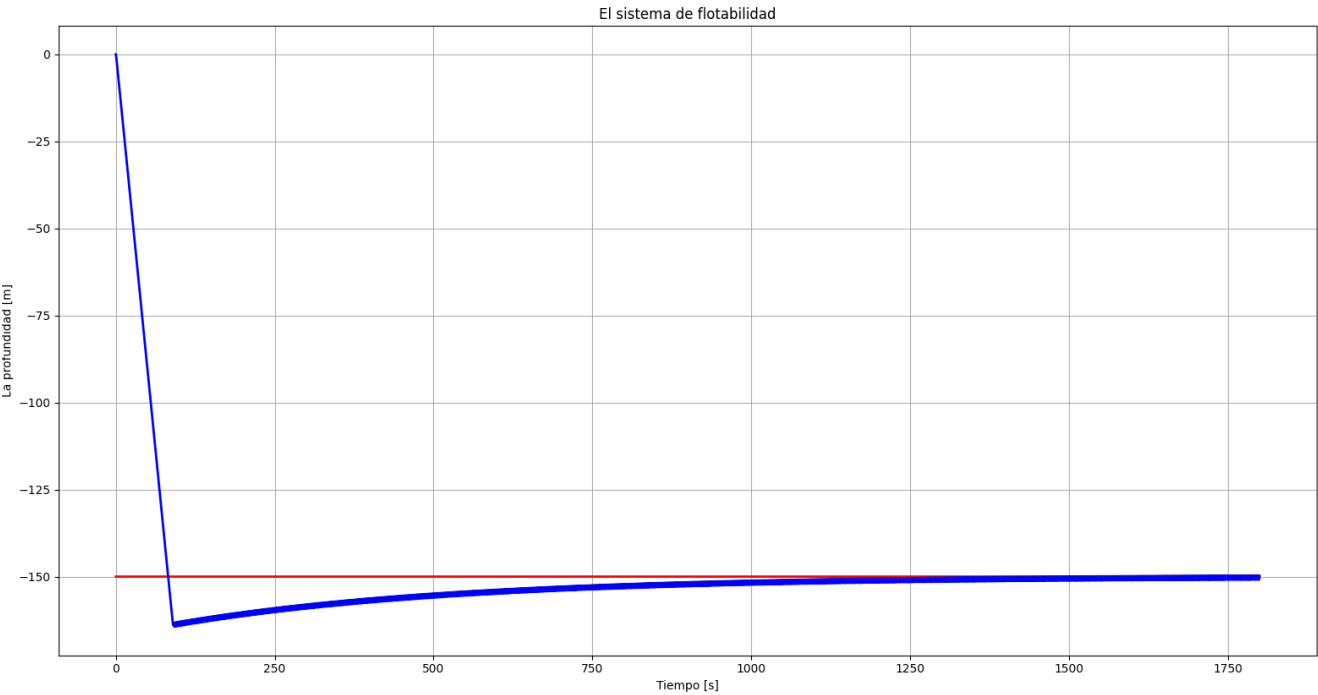


Figura 4-14 Simulación PID: movimiento de bajada-tiempo.

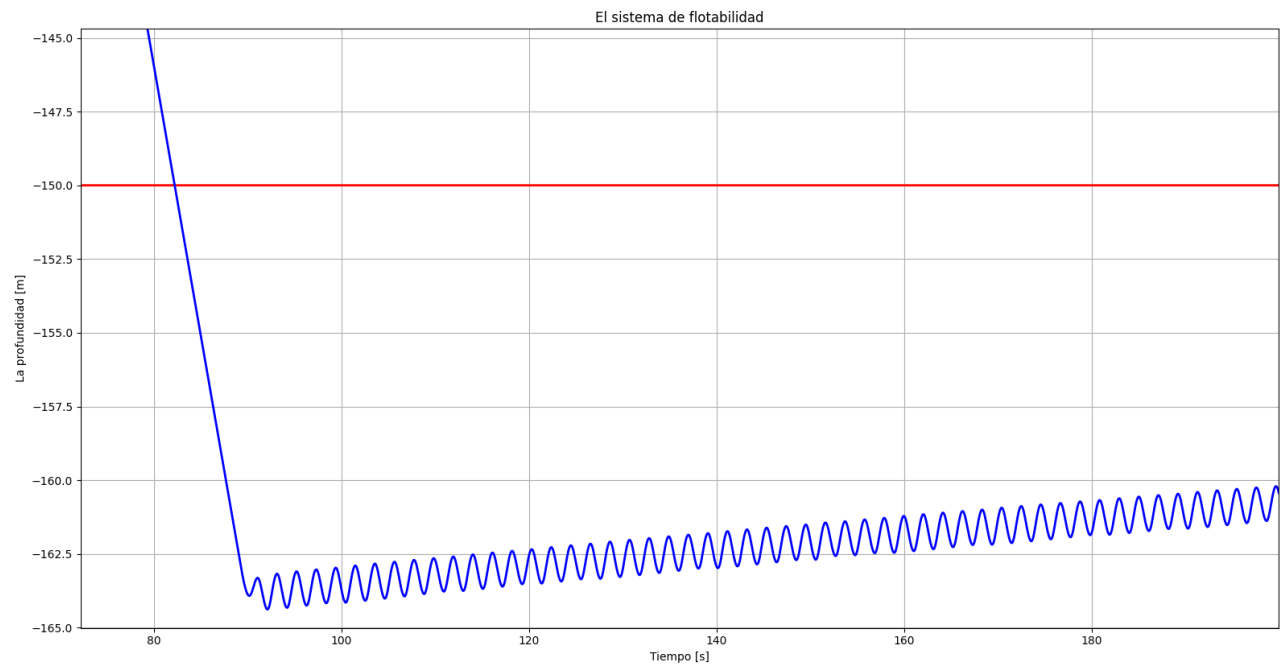


Figura 4-15 Simulación PID: movimiento de bajada-tiempo (zoom).

- Control MPD: sin minimizar parámetro

Parámetros de control

Parámetros	Valor
Q	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

S	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
R	$[1]$
hz	10

Tabla 4-4 Parámetros MPC simulado.

El controlador MPC se fundamenta en la predicción de los estados de control del sistema, tales como la profundidad, velocidad y aceleración, con la misma importancia a cada uno de ellos en este caso. Se puede observar que, en presencia de este tipo de control, el planeador submarino no sobrepasa el valor de referencia establecido y comienza a ajustar sus parámetros incluso antes de llegar a dicho punto, como se detalla en la Figura 4-16. Al enfocarse en el movimiento del planeador submarino, se puede apreciar que este tipo de control no presenta oscilaciones, como se visualiza en la Figura 4-17, ya que se logra alcanzar la referencia de forma directa, a partir de las predicciones de los estados de control. En este caso, son 10 pasos en el futuro.

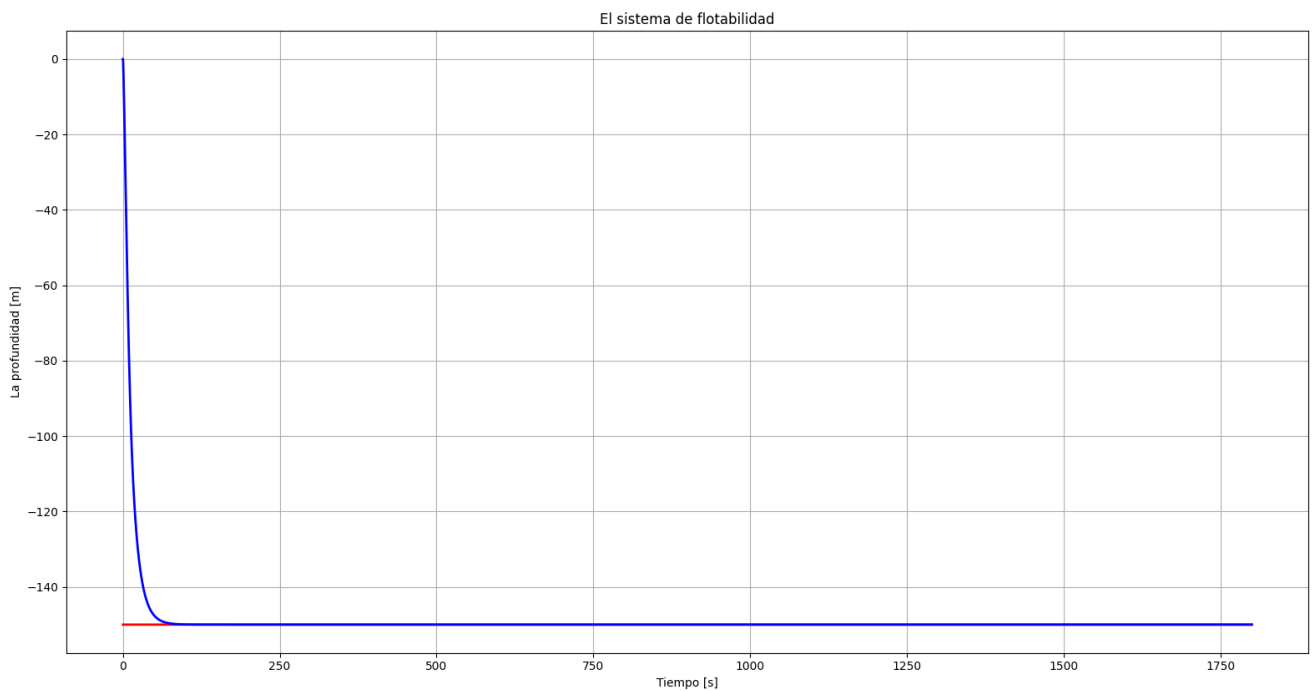


Figura 4-16 Simulación MPC: movimiento de bajada-tiempo.

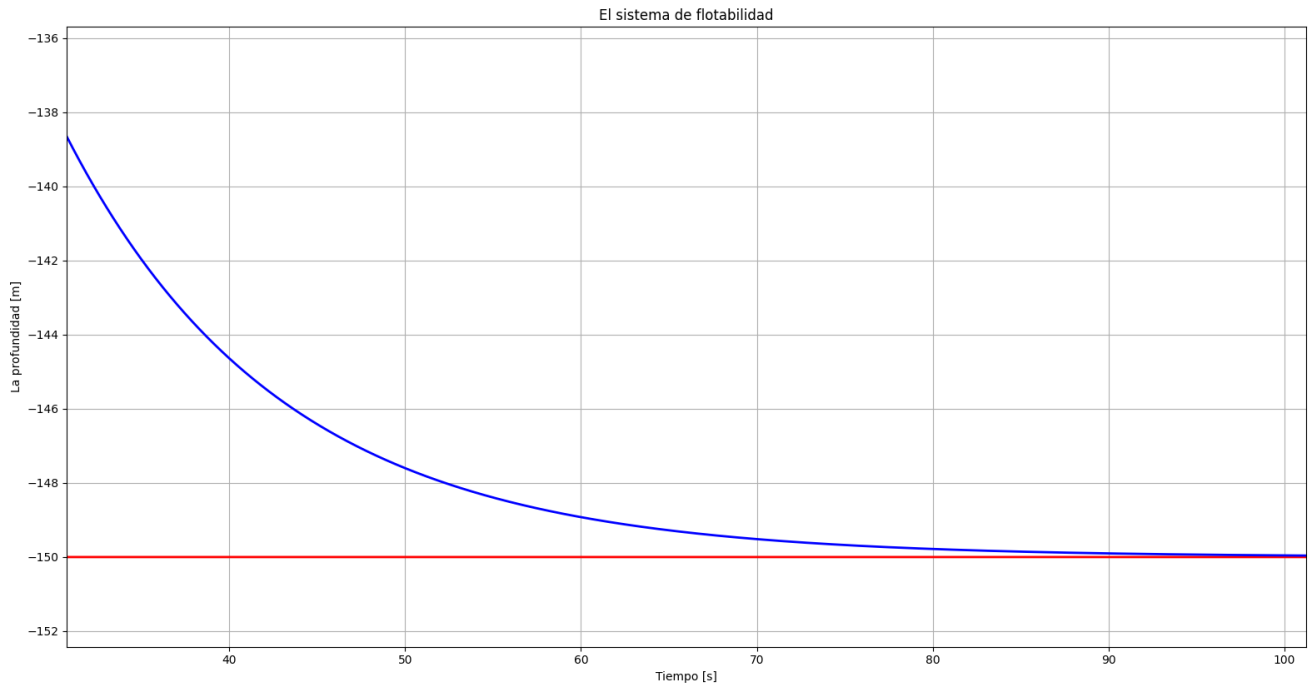


Figura 4-17 Simulación MPC: movimiento de bajada-tiempo (zoom).

Al analizar detalladamente las gráficas correspondientes a la velocidad y aceleración del sistema, se puede observar que existe un cambio brusco en la velocidad, tal como se visualiza en las Figura 4-18 y Figura 4-19. Este cambio en la velocidad puede ser indicativo de una variación repentina en la fuerza o en la dirección de la trayectoria, lo que podría generar una perturbación en el sistema y afectar su desempeño. Por ello, ante la presencia de este tipo de cambios abruptos, es necesario evaluar cuidadosamente la causa subyacente y realizar ajustes en el sistema de control para garantizar su correcto funcionamiento y evitar posibles fallas o daños en el sistema.

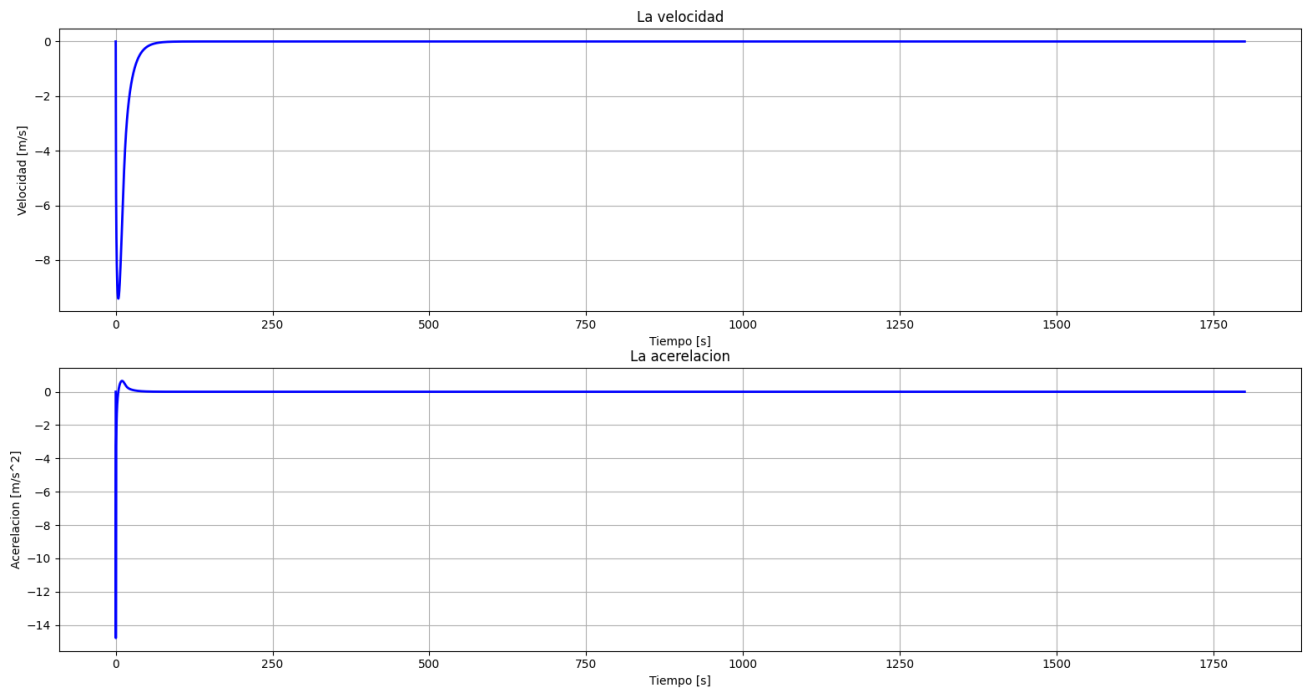


Figura 4-18 Simulación MPC: velocidad y aceleración.

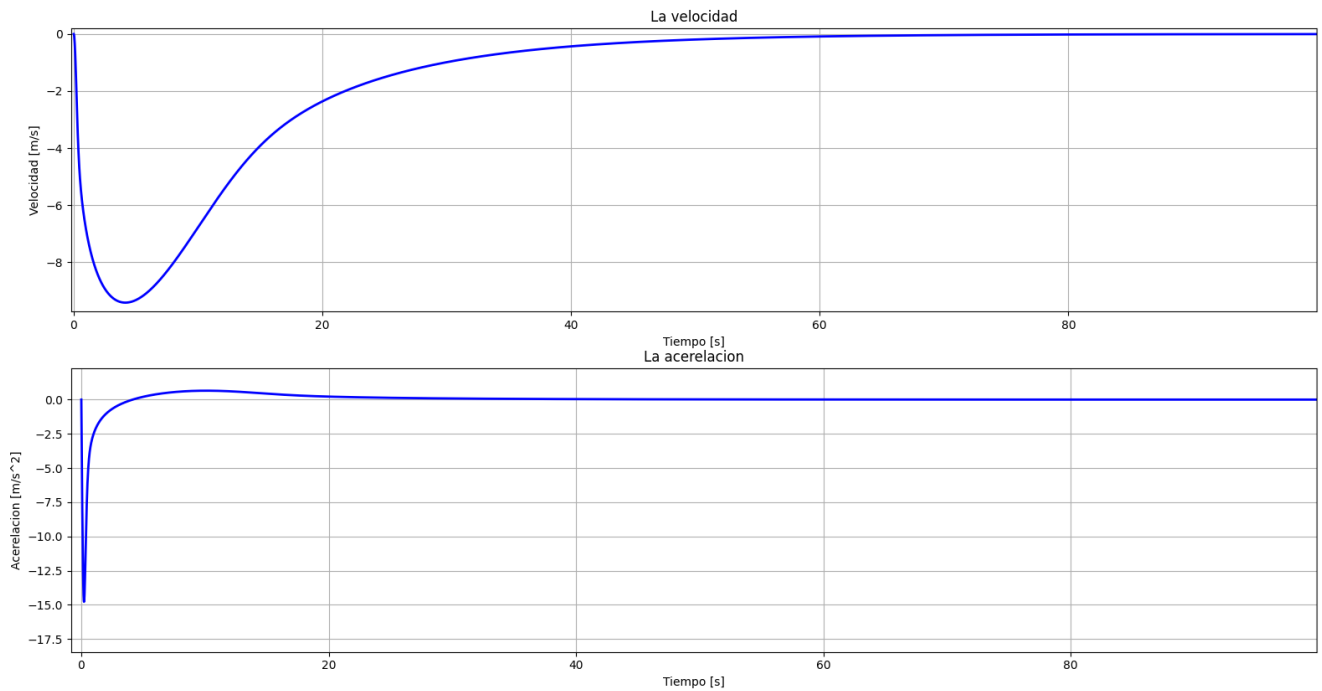


Figura 4-19 Simulación MPC: velocidad y aceleración (zoom).

- Control MPD: minimiza la velocidad

En este caso, y a partir de los resultados obtenidos antes, se decide optar por los siguientes parámetros de control para reducir s los picos de velocidades iniciales.

Parámetros	Valor
Q	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
S	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
R	[1]
hz	10

Tabla 4-5 Parámetros MPC con la minimización de la velocidad simulado.

Tal como se menciona en los resultados de otro ensayo con el controlador MPC, se ha identificado la necesidad de minimizar la velocidad empleada, que corresponde al segundo parámetro de los estados del sistema. Para lograr esto, se ha decidido asignar un valor de 100 en la posición de fila 2 y columna 2 de las matrices de peso Q y S. Como resultado de esta modificación en la configuración del controlador, se puede apreciar en la Figura 4-20 que la profundidad del sistema se ajusta de manera más gradual, lo que indica un descenso más lento en la velocidad de descenso del planeador submarino. Adicionalmente, en la Figura 4-21 se puede observar que el valor de aceleración es utilizado de manera menos intensa que en el caso anterior, lo que se traduce en una reducción en los cambios bruscos y, entonces, se consigue un aumento de la seguridad y estabilidad del sistema. Es importante resaltar que, al realizar este tipo de ajustes en las matrices de peso Q y S, se logra una mayor flexibilidad y capacidad de adaptación en el controlador MPC, lo que se traduce en un mejor desempeño y una mayor eficiencia en el sistema controlado.

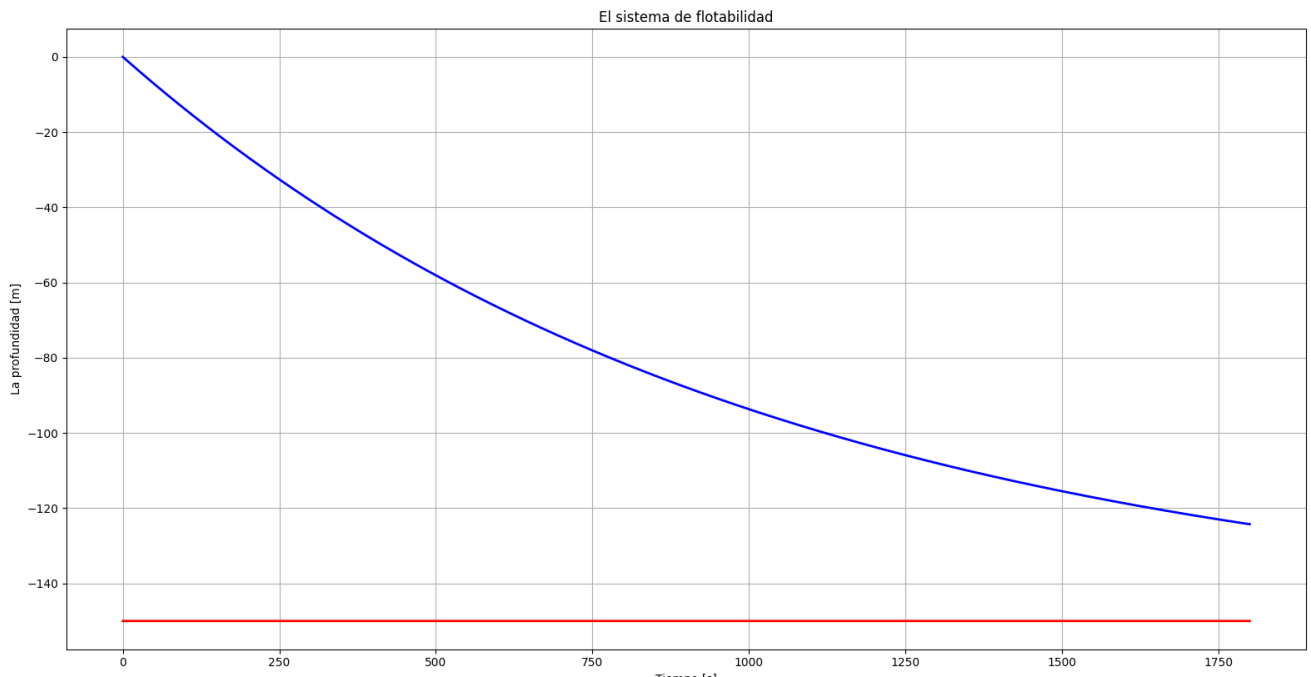


Figura 4-20 Simulación MPC con la minimización de la velocidad: movimiento de bajada-tiempo.

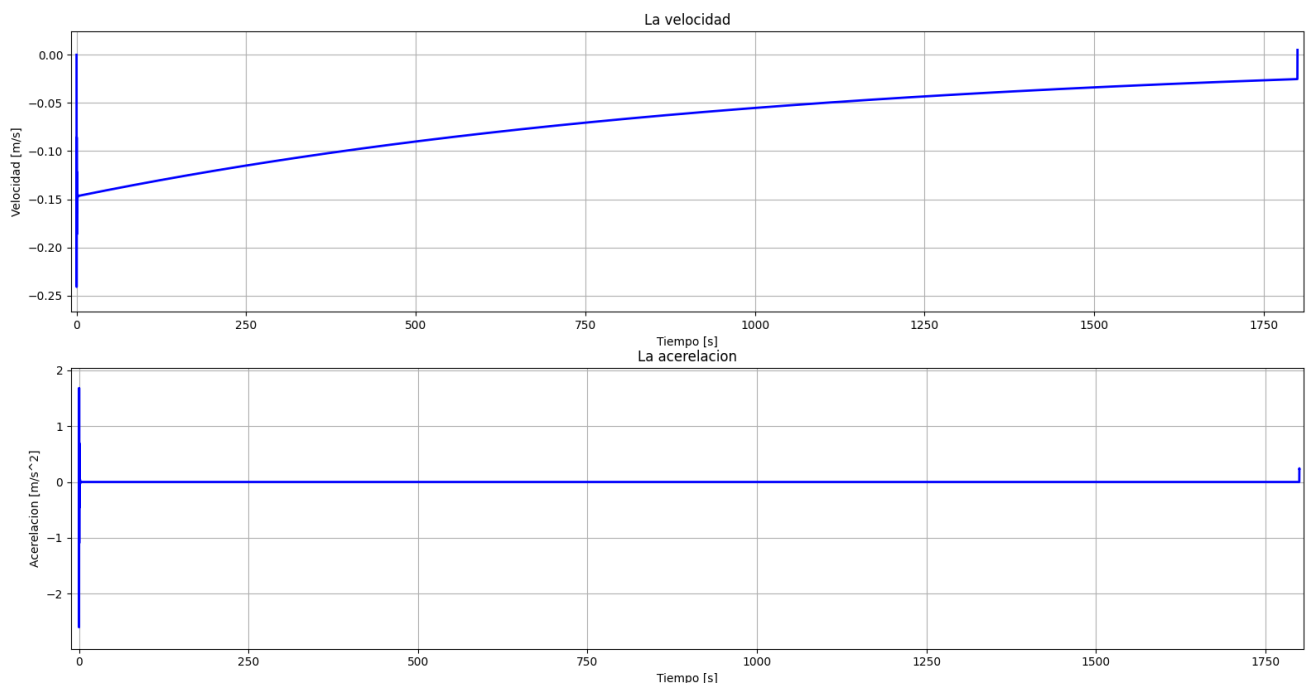


Figura 4-21 Simulación MPC con la minimización de la velocidad: velocidad y aceleración.

4.4 Discusión

En cuanto a la primera discusión, es importante considerar que el ESP32 es un microcontrolador de bajo costo y tamaño reducido, por lo que algunas limitaciones son inherentes a su diseño. En el caso de la lectura analógica, aunque puede ser un problema en algunos casos, es posible solucionarlo mediante la calibración del sensor de presión y la aplicación de técnicas de compensación de la no linealidad. Además, también es posible utilizar módulos de conversión analógico-digital externos para mejorar la precisión de las lecturas.

En relación con la segunda discusión, es cierto que la viscosidad del aceite puede afectar la precisión del caudalímetro, especialmente en casos en los que la viscosidad es alta. Sin embargo, existen diferentes

tipos de caudalímetros disponibles en el mercado, con diferentes características y especificaciones. Es importante evaluar cuidadosamente las necesidades del sistema y elegir el caudalímetro más adecuado para el trabajo en cuestión.

En cuanto a la tercera discusión, el diseño del sistema de control se ha basado en la suposición de que el valor de caudal puede ser controlado de manera precisa tanto en la entrada como en la salida del depósito. No obstante, se reconoce que esta suposición presenta ciertas complejidades, especialmente en lo que respecta a la implementación práctica del sistema. Es necesario considerar cuidadosamente los desafíos técnicos asociados con el control del caudal y evaluar las diferentes alternativas de diseño para lograr los resultados deseados de manera efectiva y eficiente. La aplicación de técnicas no lineal de control robustas, como el diseño en modos deslizantes, puede suponer una solución que permita obtener un sistema de control adecuado.

5 CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Conclusiones

En conclusión, el proyecto de diseño y desarrollo del sistema de control de flotabilidad para un planeador submarino con control de flotabilidad por transferencia de aceite ha sido exitoso. Se han diseñado y analizado las dos partes importantes del sistema, el sistema hidráulico y el sistema de control y electrónico. Se ha utilizado la tecnología adecuada para cada componente, como la bomba sin escobillas y las válvulas solenoides que funcionan con señal eléctrica, y el microcontrolador ESP32 devkit v1 para controlar el sistema. Además, se ha llevado a cabo ensayos para comprobar el comportamiento de los componentes y se ha logrado obtener las curvas características de la bomba y el consumo de los componentes. El soporte del sistema se ha diseñado utilizando tecnologías de modelado 3D y se han elegido los materiales adecuados para la producción. Se ha desarrollado un módulo de control matemático utilizando los algoritmos de control PID y MPC, y se ha simulado su comportamiento en Python.

Para alcanzar los objetivos mencionados, se ha llevado a cabo un enfoque metodológico que ha permitido cubrir cada aspecto del proyecto.

En primer lugar, para estudiar el movimiento 6DOF del planeador submarino, se ha analizado los principios físicos y las leyes de la mecánica de fluidos que rigen el comportamiento del planeador submarino. Se han utilizado herramientas matemáticas y de simulación para modelar y entender el movimiento del planeador.

En segundo lugar, para estudiar el sistema de control, se ha utilizado un enfoque teórico-práctico, que combina la teoría matemática con la simulación. Se han utilizado técnicas de control clásico y moderno, como los algoritmos de control PID y MPC, y se han simulado su comportamiento en un entorno de simulación para evaluar su desempeño. Además, se ha desarrollado un simulador de sistema de control para evaluar la eficiencia del sistema.

En tercer lugar, para diseñar el sistema de flotabilidad, se ha utilizado un enfoque de diseño basado en la simulación y el análisis de los requerimientos del sistema. Se ha llevado a cabo un análisis de los componentes hidráulicos, la electrónica y los materiales adecuados para el trabajo. También se ha utilizado herramientas de modelado 3D, como AutoCAD e Inventor, para diseñar y visualizar el sistema de flotabilidad.

Por último, para integrar la programación para el trabajo autónomo, se ha utilizado un enfoque basado en el uso de microcontroladores programables como el ESP32 devkit v1, junto con sensores de presión y caudalímetro. Se ha desarrollado un algoritmo de control para garantizar la operación autónoma del sistema de flotabilidad, y se han implementado diversas técnicas para garantizar la

eficiencia del sistema, como el uso de MQTT y SQLite para el almacenamiento de datos y la visualización de datos utilizando herramientas como Grafana o Matplotlib.

En resumen, este proyecto ha demostrado la capacidad del trabajo de diseño y desarrollo para crear un sistema eficiente y funcional para el control de flotabilidad en un planeador submarino de tipo transferencia de aceite.

5.2 Líneas futuras

A continuación, se presentan algunas consideraciones importantes para líneas futuras de trabajo en este proyecto:

- Sería recomendable considerar la posibilidad de utilizar controladores y sensores de mayor precisión para mejorar la calidad y precisión del sistema de control. No obstante, se debe tener en cuenta que esto implicaría costos adicionales significativos que deberán ser evaluados cuidadosamente.
- Es importante abordar la integración del sistema de control con el circuito completo de manera efectiva. Se requiere un análisis detallado de los desafíos técnicos asociados con la integración de todos los componentes y asegurar que el sistema funcione de manera óptima.
- Sería necesario llevar a cabo ensayos adicionales con aceite para obtener datos más precisos y completos. Se deben evaluar diferentes opciones de sensores para garantizar que se cumplan los requisitos necesarios.
- Se recomienda llevar a cabo ensayos del sistema en un banco de ensayo que pueda simular la presión externa a diferentes profundidades. Además, también se debe realizar un ensayo en agua real para evaluar el comportamiento del sistema en condiciones reales.
- Una vez que el sistema esté funcionando de manera adecuada, se puede considerar la integración con otros sistemas como el sistema de estabilidad de desplazamiento de masa [62]. También se pueden implementar sensores adicionales y capacidades de almacenamiento para investigaciones submarinas o aplicaciones militares, como por ejemplo la detección de minas y objetos submarinos.

6 BIBLIOGRAFÍA

- [1] H.-M. Huang, «Autonomy Levels for Unmanned Systems (ALFUS) Framework Volume I: Terminology Version 1.», *NIST Spec. Publ.*
- [2] Eelume, «Reshaping Underwater Operations», *Eelume*, 11 de febrero de 2023. <http://www.eelume.com> (accedido 11 de febrero de 2023).
- [3] «Seaglider: Autonomous Underwater Vehicle». https://apl.uw.edu/project/project.php?id=seaglider_auv (accedido 6 de enero de 2023).
- [4] P. Singh, S. Shankar, y Dr. R. Vijayakumar, «Review and Design of Buoyancy Engine for Underwater Glider operating at Shallow Depth», en *OCEANS 2022 - Chennai*, feb. 2022, pp. 1-7. doi: 10.1109/OCEANSSchennai45887.2022.9775521.
- [5] «¿En qué consiste la ingeniería 4.0? - NC Tech», 7 de septiembre de 2021. <https://nctech.com.mx/blog/ingenieria-digital/que-es-la-ingenieria-4-0/>, <https://nctech.com.mx/blog/ingenieria-digital/que-es-la-ingenieria-4-0/> (accedido 7 de enero de 2023).
- [6] «(25) Engineering 4.0: A Glimpse into the Future of Engineering | LinkedIn». <https://www.linkedin.com/pulse/engineering-40-how-get-from-here-carmit-berdugo-cohen/> (accedido 7 de enero de 2023).
- [7] E. Hofmann y M. Rüsch, «Industry 4.0 and the current status as well as future prospects on logistics», *Comput. Ind.*, vol. 89, pp. 23-34, ago. 2017, doi: 10.1016/j.compind.2017.04.002.
- [8] A. Görmüş, «Future of Work with the Industry 4.0», 2019, pp. 317-323.
- [9] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, y M. Ayyash, «Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications», *IEEE Commun. Surv. Tutor.*, vol. 17, n.º 4, pp. 2347-2376, 2015, doi: 10.1109/COMST.2015.2444095.
- [10] A. Garg, «How to Connect IoT Sensors Wirelessly With a Web Application?», *Analytics Vidhya*, 2 de septiembre de 2022. <https://www.analyticsvidhya.com/blog/2022/09/how-to-connect-iot-sensors-wirelessly-with-a-web-application/> (accedido 28 de enero de 2023).
- [11] B. Holfeld *et al.*, «Wireless Communication for Factory Automation: an opportunity for LTE and 5G systems», *IEEE Commun. Mag.*, vol. 54, n.º 6, pp. 36-43, jun. 2016, doi: 10.1109/MCOM.2016.7497764.
- [12] R. D. Christ y R. L. Wernli, *The ROV manual: a user guide to observation-class remotely operated vehicles*, 1st ed. Amsterdam ; Boston: Butterworth-Heinemann, 2007.
- [13] «UUVMPRelease.pdf». Accedido: 8 de enero de 2023. [En línea]. Disponible en: <https://higherlogicdownload.s3.amazonaws.com/AUVSI/958c920a-7f9b-4ad2-9807-f9a4e95d1ef1/UploadedImages/UUVMPRelease.pdf>
- [14] «Chasing M2 ROV submarino», *El vuelo del Drone*. <https://elvuelodeldrone.com/drones-profesionales/drones-acuaticos/chasing-m2-rov-submarino/> (accedido 28 de enero de 2023).

- [15] «What is an AUV?: Ocean Exploration Facts: NOAA Office of Ocean Exploration and Research». <https://oceanexplorer.noaa.gov/facts/auv.html> (accedido 6 de enero de 2023).
- [16] M. Javaid, M. Ovinis, T. Nagarajan, y F. Hashim, «Underwater Gliders: A Review», *MATEC Web Conf.*, vol. 13, p. 02020, jul. 2014, doi: 10.1051/mateconf/20141302020.
- [17] «Physical Oceanography Division - Global Ocean Observations - NOAA/AOML - CARICOOS Hurricane Underwater Gliders». <https://www.aoml.noaa.gov/phod/goos/gliders/2021/index.php?id=SG649#latestobs> (accedido 15 de enero de 2023).
- [18] W. Sun, W. Zang, C. Liu, T. Guo, Y. Nie, y D. Song, «Motion Pattern Optimization and Energy Analysis for Underwater Glider Based on the Multi-Objective Artificial Bee Colony Method», *J. Mar. Sci. Eng.*, vol. 9, n.º 3, Art. n.º 3, mar. 2021, doi: 10.3390/jmse9030327.
- [19] F. M. & Comunicatie, «Mapping The Norwegian Seafloor with Kongsberg Seaglider AUV (Autonomous Underwater Vehicle)». <https://geo-matching.com/content/mapping-the-norwegian-seafloor-with-kongsberg-seaglider-auv-autonomous-underwater-vehicle> (accedido 28 de enero de 2023).
- [20] P. Singh, S. Shankar, y Dr. R. Vijayakumar, «Review and Design of Buoyancy Engine for Underwater Glider operating at Shallow Depth», en *OCEANS 2022 - Chennai*, feb. 2022, pp. 1-7. doi: 10.1109/OCEANSCennai45887.2022.9775521.
- [21] «Gallery | OSUG: Open-Source Underwater Glider». <https://hackaday.io/project/20458-osug-open-source-underwater-glider> (accedido 28 de enero de 2023).
- [22] D. C. Webb, P. J. Simonetti, y C. P. Jones, «SLOCUM: an underwater glider propelled by environmental energy», *IEEE J. Ocean. Eng.*, vol. 26, n.º 4, pp. 447-452, oct. 2001, doi: 10.1109/48.972077.
- [23] T. Miller, F. Gandhi, y R. Rufino, «Morphing Hull Implementation for Unmanned Underwater Vehicles», *Smart Mater. Struct.*, vol. 22, p. 5014, nov. 2013, doi: 10.1088/0964-1726/22/11/115014.
- [24] F. Zhang, J. Hou, D. Ning, y Y. Gong, «Depth Control of an Oil Bladder Type Deep-Sea AUV Based on Fuzzy Adaptive Linear Active Disturbance Rejection Control», *Machines*, vol. 10, p. 163, feb. 2022, doi: 10.3390/machines10030163.
- [25] «About Arduino». <https://www.arduino.cc/en/about> (accedido 28 de enero de 2023).
- [26] E. S. Kumar, «Arduino Working Principle and It's Use in Education», *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, abr. 2022, doi: 10.22214/ijraset.2022.41784.
- [27] «Raspberry Pi - About us», *Raspberry Pi*. <https://www.raspberrypi.com/about/> (accedido 28 de enero de 2023).
- [28] R. P. Ltd, «Raspberry Pi 4 Model B specifications», *Raspberry Pi*. <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/> (accedido 28 de enero de 2023).
- [29] O. Media, «Raspberry Pi goes custom for industrial, commercial applications», *Embedded Computing Design*. <https://embeddedcomputing.com/technology/open-source/raspberry-pi-goes-custom-for-industrial-commercial-applications> (accedido 28 de enero de 2023).
- [30] «ESP32 Wi-Fi & Bluetooth MCU I Espressif Systems». <https://www.espressif.com/en/products/socs/esp32> (accedido 4 de febrero de 2023).
- [31] «Getting Started with the ESP32 Development Board | Random Nerd Tutorials», 2 de noviembre de 2022. <https://randomnerdtutorials.com/getting-started-with-esp32/> (accedido 4 de febrero de 2023).
- [32] N. Tin, E. Apriliani, y H. Nurhadi, *The Position Estimation of AUV Based on Non-Linear Ensemble Kalman Filter Method*. 2016.
- [33] «Airfoils, Bernoulli and Newton». <http://hyperphysics.phy-astr.gsu.edu/hbase/Fluids/airfoil.html#c3> (accedido 4 de febrero de 2023).
- [34] «On what principle does the submarines based?», *Toppr Ask*. <https://www.toppr.com/ask/question/on-what-principle-does-the-submarines-based/> (accedido 4 de febrero de 2023).

- [35] A. Khair, B. S. Biswas, y M. Ansary, *APPLICATION OF SEISMIC REFRACTION METHOD TO INVESTIGATE SOIL STRUCURE OF DHAKA CITY*. 2015.
- [36] R. Feistel, G. Nausch, y N. Wasmund, *State and Evolution of the Baltic Sea, 1952-2005: A Detailed 50-Year Survey of Meteorology and Climate, Physics, Chemistry, Biology, and Marine Environment*. John Wiley & Sons, 2008.
- [37] J. Aparicio, E. García Núñez, A. Jiménez Martín, F. Álvarez, y J. Ureña, «Modelos de Propagación de Señales Acústicas en Entornos Subacuáticos (II)», *CONECtrónica*, vol. 139, p. 40, jul. 2010.
- [38] Director Dep. Det. Cont. y T. y A. J. VIDAL ROJAS, «TEMA I ACÚSTICA», en *ACÚSTICA*, 2019.
- [39] «Drag (physics)», *Wikipedia*. 18 de enero de 2023. Accedido: 22 de enero de 2023. [En línea]. Disponible en: [https://en.wikipedia.org/w/index.php?title=Drag_\(physics\)&oldid=1134372089](https://en.wikipedia.org/w/index.php?title=Drag_(physics)&oldid=1134372089)
- [40] K.-L. Huang, Y.-J. Pang, Y.-M. Su, y J. Zhu, «Research on linearity hydrodynamic coefficients calculation method of submergible vehicle», *Chuan Bo Li Xue Journal Ship Mech.*, vol. 12, pp. 697-703, oct. 2008.
- [41] «Heitune Control Remoto Modelo De Accesorios Transparente 1000ml del Depósito De Combustible (Compatible con El Avion RC 85-120CC): Amazon.es: Hogar y cocina». https://www.amazon.es/gp/product/B08FJ3JYPM/ref=ppx_yo_dt_b_asin_title_o07_s00?ie=UTF8&psc=1 (accedido 22 de enero de 2023).
- [42] «Bomba hidráulica Brushless mini M3 con motor», *MAGOM HRC*. <https://magomhrc.com/es/bomba-brushless-sin-deposito/486-bomba-hidraulica-brushless-mini-m3-con-motor.html> (accedido 22 de enero de 2023).
- [43] «Sharplace Mini Bomba de Succión Autocebante Agua Aire DC 12V Relación <60db : Amazon.es: Coche y moto». https://www.amazon.es/dp/B076PRZSJ2/ref=redir_mobile_desktop?_encoding=UTF8&psc=1&ref=ppx_pop_mob_b_asin_title (accedido 19 de marzo de 2023).
- [44] «013-3V1-202107271100006686.pdf». Accedido: 22 de enero de 2023. [En línea]. Disponible en: <https://as-th.airtac.com/upload/013-3V1-202107271100006686.PDF>
- [45] «7.46€ 10% de DESCUENTO|Válvula Solenoide 24V DC 12V 110V 220V, 1/4 "1/8", 2 vías normalmente cerradas, válvulas neumáticas de acción directa para agua, aire, Gas caliente|Válvula - AliExpress», *aliexpress.com*. https://es.aliexpress.com/item/4001343774912.html?src=ibdm_d03p0558e02r02&sk=&aff_platform=&aff_trace_key=&af=&cv=&cn=&dp= (accedido 22 de enero de 2023).
- [46] «Válvula de retención neumática SUPERTOOL Válvula de retención sin retorno rosca hembra Válvula de retención de aire unidireccional de puerto completo para petróleo y gas de agua (2 unidades, 1/4") : Amazon.es: Bricolaje y herramientas». https://www.amazon.es/gp/product/B07XP4R3DQ/ref=ppx_yo_dt_b_asin_title_o08_s00?ie=UTF8&th=1 (accedido 22 de enero de 2023).
- [47] «Rubber Bladder Accumulator Pressure Vessel Nxq-1.6/31.5-l 1.6liter Hydraulic Bladder Accumulator Telescoping Hydraulic Cylinder - Buy Accumulator Pressure Calculation Inflatable Rubber Bladder, Accumulator Charge Pressure, Inflatable Bladders Manufacturer High Pressure Rubber Bladder Product on Alibaba.com». https://www.alibaba.com/product-detail/rubber-bladder-accumulator-pressure-vessel-NXQ_1600097524294.html?spm=a2700.shop_plgr.41413.61.54687699mk5LIN (accedido 22 de enero de 2023).
- [48] «Gexmile Manguera de Tubo Neumático PU 10 metros Manguera de Línea Aire Poliuretano Tubería Compresor Alta Presión Exterior 6mm x Interior 4mm Negro : Amazon.es: Bricolaje y herramientas». https://www.amazon.es/gp/product/B09BF8MZ2V/ref=ppx_yo_dt_b_asin_title_o07_s00?ie=UTF8&th=1 (accedido 22 de enero de 2023).

- [49] «HUAZIZ 10m Manguera Aire Poliuretano, 4mm Aire de PU Tubo Manguera transparente + 14Pcs 4mm Air Push Quick Conectores Fittings, Neumáticos de Aire Impresora, Conexión Rápida de Tuberías de Aire : Amazon.es: Bricolaje y herramientas». <https://www.amazon.es/HUAZIZ-Poliuretano-transparente-Conectores-Neum%C3%A1ticos/dp/B09FL2DJK3> (accedido 22 de enero de 2023).
- [50] «SRD-03VDC-SL-C pdf, SRD-03VDC-SL-C Description, SRD-03VDC-SL-C Datasheet, SRD-03VDC-SL-C view :: ALLDATASHEET »: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1132027/SONGLERELAY/SRD-03VDC-SL-C.html> (accedido 11 de febrero de 2023).
- [51] «ina219.pdf». Accedido: 11 de febrero de 2023. [En línea]. Disponible en: <https://cdn-shop.adafruit.com/datasheets/ina219.pdf>
- [52] «Transductor de presión, Akozon Sensor de presión de acero inoxidable, transmisor 1/8 pulgada NPT DC 0.5-4.5V para compresor de aire, aceite, agua, cerámica(100PSI) : Amazon.es: Coche y moto». https://www.amazon.es/dp/B07J6GZB4H/ref=redir_mobile_desktop?_encoding=UTF8&psc=1&ref=ppx_pop_mob_b_asin_title (accedido 11 de febrero de 2023).
- [53] «Sensor de Caudalímetro de Agua de 3,5-24V CC, Sensor de Caudalímetro de 0,2-8L/min, Caudalímetro de Efecto Hall con Cable de 25cm para Máquina de Café, Dispensador de Agua : Amazon.es: Industria, empresas y ciencia». https://www.amazon.es/dp/B0B1N9N3XC/ref=redir_mobile_desktop?_encoding=UTF8&psc=1&ref=ppx_pop_mob_b_asin_title (accedido 19 de marzo de 2023).
- [54] «Battery Expansion Shield 18650 V3 including USB cable», *AZ-Delivery*. <https://www.az-delivery.de/en/products/battery-expansion-shield-18650-v3-inkl-usb-kabel> (accedido 5 de febrero de 2023).
- [55] «MQTT - The Standard for IoT Messaging». <https://mqtt.org/> (accedido 12 de febrero de 2023).
- [56] «SQLite Home Page». <https://www.sqlite.org/index.html> (accedido 20 de febrero de 2023).
- [57] «Grafana OSS | Metrics, logs, traces, and more», *Grafana Labs*. <https://grafana.com/oss/grafana/> (accedido 20 de febrero de 2023).
- [58] «Ultimate 3D Printing Material Properties Table». <https://www.simplify3d.com/support/materials-guide/properties-table/> (accedido 22 de abril de 2022).
- [59] «Control Tutorials for MATLAB and Simulink - Introduction: PID Controller Design». <https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction§ion=ControlPID> (accedido 5 de febrero de 2023).
- [60] J. B. Rawlings, D. Q. Mayne, y M. Diehl, *Model predictive control: theory, computation, and design*, 2nd edition. Madison, Wisconsin: Nob Hill Publishing, 2017.
- [61] E. Camacho y C. Bordons, *Model Predictive Control*, vol. 13. 2004. doi: 10.1007/978-0-85729-398-5.
- [62] I. Vidal Medina y C. Pérez Collazo (advisor), «Diseño de un sistema de energía y control de la orientación de un planeador submarino», Centro Universitario de la Defensa de Marín, Working Paper, abr. 2022. Accedido: 19 de marzo de 2023. [En línea]. Disponible en: <http://calderon.cud.uvigo.es/handle/123456789/552>

ANEXO I IMPLICACIONES SOCIALES, ECONÓMICAS, Y AMBIENTALES

El proyecto de diseño y desarrollo del sistema de control de flotabilidad de un planeador submarino tiene implicaciones sociales, económicas y ambientales significativas.

- En términos sociales, el uso de planeadores submarinos para investigación y exploración submarina puede tener un gran impacto en la mejora del conocimiento sobre el ecosistema marino y la identificación de problemas ambientales que pueden afectar a la sociedad en general. También puede tener aplicaciones en la industria militar para la búsqueda de minas y objetos submarinos, lo que puede aumentar la seguridad de las poblaciones cercanas a la costa.
- En términos económicos, el uso de estos dispositivos para la exploración submarina y la identificación de recursos naturales puede tener un impacto importante en la economía, ya que puede llevar a la identificación de nuevos recursos y la mejora de la explotación de los ya existentes. Además, el desarrollo de sistemas de control de flotabilidad más precisos y eficientes puede tener un impacto positivo en la industria naval y marítima, al mejorar la navegación y la seguridad en el mar.
- En términos ambientales, el uso de planeadores submarinos para la investigación y exploración del ecosistema marino puede tener un impacto significativo en la mejora de la conservación y protección del medio ambiente marino. Además, el desarrollo de sistemas de control de flotabilidad más eficientes puede ayudar a reducir el consumo de energía y la emisión de gases de efecto invernadero, lo que puede tener un impacto positivo en el medio ambiente.

En conclusión, el proyecto de diseño y desarrollo del sistema de control de flotabilidad de un planeador submarino tiene implicaciones significativas en términos sociales, económicos y ambientales, lo que demuestra la importancia de tener en cuenta estos aspectos en la planificación y desarrollo de productos y sistemas en cualquier industria.

ANEXO II REFLEXIONES ÉTICAS Y SOCIALES

En cuanto a las reflexiones éticas y sociales, es importante considerar el impacto que tiene el desarrollo de este tipo de tecnología en el medio ambiente y en la sociedad. Por un lado, el uso de sistemas submarinos autónomos para la investigación y la exploración puede ser una herramienta valiosa para la protección y conservación del medio ambiente marino. Sin embargo, también es importante considerar los posibles efectos negativos en el ecosistema y en la vida marina que pueden resultar del uso de esta tecnología.

En cuanto a las implicaciones sociales, es importante considerar el acceso a la tecnología. Es posible que el costo de desarrollo y producción de este tipo de sistemas los hagan inaccesibles para ciertos grupos o comunidades. Además, es importante considerar cómo se puede utilizar esta tecnología de manera responsable y ética, evitando su uso para fines militares o para actividades ilegales como la exploración de recursos naturales en territorios de otros países sin su consentimiento.

En resumen, la reflexión ética y social es fundamental en el desarrollo de cualquier tecnología y es importante considerar no solo los beneficios y avances que se puedan obtener, sino también los posibles impactos negativos en el medio ambiente y en la sociedad, y trabajar para minimizarlos o evitarlos.

ANEXO III CÓDIGO DE ENSAYO

```

#include <Wire.h>
#include <Adafruit_INA219.h>
#include <WiFi.h>
#include <PubSubClient.h>

//Objeto del proyecto
Adafruit_INA219 ina219;
WiFiClient wifiClient;
PubSubClient MQTTClient(wifiClient);

//Configuracion Wifi
const char* wifi_ssid = "WIFI_SSID";
const char* wifi_password = "*****";

//Configuracion MQTT
char mqtt_hostname[] = "192.168.137.1"; // MQTT IP address of machine
const char* mqtt_data_topic = "TFG"; // MQTT token to publish IMU data
int mqtt_port = 1883; // Default MQTT port
String mqtt_message = ""; // MQTT message to be send o received

const int button = 19;

const int relay = 4;

//Estados
int buttonState = 0;
float pump_current = 0.0;
float pump_voltage = 0.0;
float pressure = 0.0;
float flow = 0.0;
float membrane_pressure = 0.0;

//Sensor de presión
const int pressureInput = 34;
const int membranepressureInput = 35;

//Sensor de caudal
#define Flow_Input 33
volatile unsigned int pulseCount = 0;
float flowRate;
unsigned long lastPulseTime = 0;

void setup() {
  Serial.begin(115200);
  while (!Serial) {
    delay(1);
  }
}

```

```
//Connect to INA219
if (! ina219.begin()) {
    Serial.println("Failed to find INA219 chip");
    while (1) { delay(10); }
}
//Calibration of INA219
ina219.setCalibration_32V_1A();
Serial.println("Measuring voltage and current with INA219 ...");

// Connect to MQTT server
MQTTClient.setServer(mqtt_hostname, mqtt_port);
mqttConnect();
delay(10);

pinMode(button, INPUT);
pinMode(relay, OUTPUT);
pinMode(pressureInput, INPUT);
pinMode(Flow_Input, INPUT_PULLUP);

Serial.begin(115200);
}

void loop() {
    buttonState = digitalRead(button);

    if (buttonState == HIGH) {
        digitalWrite(relay, HIGH);
    } else {
        digitalWrite(relay, LOW);
    }

    pump_current = MeasureCurrent();
    pump_voltage = MeasureVoltage();
    pressure = MeasurePresure();
    flow = MeasureFlow();
    membrane_pressure = MeasureMembranePresure();

    mqtt_message = String(pump_current) + "," + String(pump_voltage) + "," +
String(pressure) + "," + String(membrane_pressure)+ "," + String(flow);
    Serial.println(mqtt_message);
    mqttSendMessage();
    delay(1000);
}

void mqttSendMessage()
{
    char attributes[1000];
    mqtt_message.toCharArray( attributes, 200 );
    MQTTClient.publish(mqtt_data_topic, attributes);
}
```

```
void mqttConnect()
{
    const char* sensor_id = "your_sensor_id";
    while (!MQTTClient.connected()) {

        if ( WiFi.status() != WL_CONNECTED) {
            Serial.println("Connecting to Wifi");
            WiFi.begin(wifi_ssid, wifi_password);
            while (WiFi.status() != WL_CONNECTED) {
                delay(500);
                Serial.print(".");
            }
            Serial.println(".");
            Serial.println("Connected to Wifi !!");
        }

        Serial.println("Connecting to MQTT server");
        if (MQTTClient.connect(sensor_id, mqtt_data_topic, NULL) )
        {
            Serial.println("Connected to MQTT server !!" );
        }
        else {
            Serial.println( " ... retrying in 1 second ... " );
            delay( 1000 );
        }
    }
}

float MeasureCurrent()
{
    float current = 0;
    current=ina219.getCurrent_mA();
    return current;
}

float MeasureVoltage()
{
    float voltage = 0;
    voltage=ina219.getBusVoltage_V() + (ina219.getShuntVoltage_mV() / 1000);
    return voltage;
}

float MeasurePresure() //loop routine runs over and over again forever
{
    const int pressureZero = 256; //analog reading of pressure transducer at 0psi
    const int pressureMax = 3686.4; //analog reading of pressure transducer at 100psi
    const int pressuretransducermaxPSI = 80; //psi value of transducer being used
```

```
float pressureValue = 0; //variable to store the value coming from the pressure
transducer

    pressureValue = analogRead(pressureInput); //reads value from input pin and assigns
to variable
    pressureValue = ((pressureValue-
pressureZero)*pressuretransducermaxPSI)/(pressureMax-pressureZero); //conversion
equation to convert analog reading to psi
    return pressureValue;
}

float MeasureMembranePresure() //loop routine runs over and over again forever
{
const int pressureZero = 230; //analog reading of pressure transducer at 0psi
const int pressureMax = 3686.4; //analog reading of pressure transducer at 100psi
const int pressuretransducermaxPSI = 174; //psi value of transducer being used
float pressureValue = 0; //variable to store the value coming from the pressure
transducer

    pressureValue = analogRead(membranepressureInput); //reads value from input pin and
assigns to variable
    pressureValue = ((pressureValue-
pressureZero)*pressuretransducermaxPSI)/(pressureMax-pressureZero); //conversion
equation to convert analog reading to psi
    //Serial.print(pressureValue, 3); //prints value from previous line to serial
    //Serial.println("psi"); //prints label to serial
    return pressureValue;
}

void pulseCounter() {
    pulseCount++;
}

float MeasureFlow() {
// Constants
const float flowRateConversionFactor = 20; // Flow rate conversion factor from
pulses to liters per minute

    unsigned long currentTime = millis();
    if (currentTime - lastPulseTime >= 1000) { // Calculate flow rate every 1 second
        detachInterrupt(digitalPinToInterrupt(Flow_Input)); // Disable interrupts to
avoid conflicts
        flowRate = pulseCount / flowRateConversionFactor; // Calculate flow rate
        pulseCount = 0; // Reset pulse count
        attachInterrupt(digitalPinToInterrupt(Flow_Input), pulseCounter, FALLING); //
Re-enable interrupts
        lastPulseTime = currentTime; // Update last pulse time
    }
    return flowRate;
}
```

ANEXO IV CÓDIGO DE INSERTAR DATOS A SQLITE (MAIN FILE)

```

import sys, traceback, time
from classes import Logger
import paho.mqtt.client as paho

def on_message(client, userdata, message):
    global loggerObj, session_id

    # Leer los datos
    message_data = str(message.payload.decode("utf-8")).split(',')
    data = [session_id, str(time.ctime()), message_data]

    #Cambiar formato de tiempo a string
    time_txt = time.time()
    time_txt = time.strftime('%Y-%m-%d %H:%M:%S', time.localtime(time_txt))

    #Pasar datos a los variables
    pump_current = message_data[0]
    valve1_current = message_data[1]
    valve2_current = message_data[2]
    tank_pressure= message_data[3]
    membrane_pressure= message_data[4]
    valve1_state= message_data[5]
    valve2_state= message_data[6]
    pump_state= message_data[7]

    #Demostrar los datos recopilados
    data = [session_id, time_txt, valve1_current, valve2_current, pump_current,
    tank_pressure, membrane_pressure, valve1_state, valve2_state, pump_state]
    print('data=', data)

    #Insertar los datos al base de datos
    loggerObj.insertData(data)
    return

if __name__ == '__main__':
    try:
        # Session id
        session_id = 1
        steps = 20

#####

        # Database
        database_path = "./TFG.db"

```

```
# MQTT
mqtt_broker = '192.168.137.1'
mqtt_port = 1883
mqtt_topic = 'TFG'

# Crear Logger
loggerObj = Logger()
loggerObj.initialize(database_path)

# Crear un MQTT subscripores
mqtt_client = paho.Client("tfg_logger")
mqtt_client.on_message = on_message
mqtt_client.connect(mqtt_broker, mqtt_port)
mqtt_client.subscribe(mqtt_topic)

# Bucle eterna de Leer Los datos
while True:
    mqtt_client.loop_forever()

# Para interrumpir el programa
except KeyboardInterrupt:
    print('KEYBOARD INTERRUPT !!! ', str(traceback.format_exc()))
    sys.exit(0)
except:
    print('MAIN ERROR : ', str(traceback.format_exc()))
    sys.exit(0)
```

ANEXO V CÓDIGO DE INSERTAR DATOS A SQLITE (CLASS FILE)

```

import traceback, datetime, time
import sqlite3

class DBsqlite():
    def __init__(self, database_path):
        self.connection = sqlite3.connect(database_path)
        self.cursor = self.connection.cursor()

    def __del__(self):
        self.connection.close()

    def insert(self, sql, data):
        try:
            self.cursor.execute(sql, data)
            self.connection.commit()
            return self.cursor.lastrowid
        except:
            print("[DBSqlite] insert error !!!", str(traceback.format_exc()))
            return None

    def update(self, sql, data):
        try:
            self.cursor.execute(sql, data)
            self.connection.commit()
            return True
        except:
            print("[DBSqlite] insert error !!!", str(traceback.format_exc()))
            return False

    def select(self, sql, data):
        try:
            self.cursor.execute(sql, data)
            rows = self.cursor.fetchall()
            return rows
        except:
            print("[DBSqlite] insert error !!!", str(traceback.format_exc()))
            return None

class Logger():
    def __init__(self):
        self.dbObj = None
        self.insert_log_data_sql = ''' INSERT INTO LOG_TFG (session_id, time,
valve1_current, valve2_current, pump_current, tank_pressure, membrane_pressure,
valve1_state, valve2_state, pump_state) VALUES(?,?,?,?,?,?,?,?,?,?) '''

```

```
def initialize(self, database_path):
    try:
        self.dbObj = DBsqlite(database_path)
        return True
    except:
        print("[Logger] insertData error !!!", str(traceback.format_exc()))
        return False

def insertData(self, data):
    try:
        self.session_id = int(self.dbObj.insert(self.insert_log_data_sql, data))
        return self.session_id
    except:
        print("[Logger] insertData error !!!", str(traceback.format_exc()))
        return None
```

ANEXO VI CÓDIGO SIMULACIÓN PID

```

import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
import matplotlib.animation as animation
import numpy as np

D=120 # Diametro del tubo en mm
l=1000 # Longitud del tubo en mm
b=0 # profundidad inicial
m=20 # kg
g=9.81 # [m/s^2]
Wdb=m*g # Peso del planeador.
As=l*D*10**(-6) # Sección flotante
final_b=-100 # profundidad maxima
dVol=10 # The change of volume on the vertical scale.
densidad_water=1025 # [kg/m^3]
densidad_oil=885 # [kg/m^3]
Cd=0.4 # Coeficiente de resistencia en el agua
Bgb=densidad_water*g*np.pi*D**(2)/4*l*10**(-9)

dt=0.04 # intervalo de tiempo
t0=0 # Tiempo de inicio de la simulación
t_end=1800 # Tiempo final de la simulación
frame_amount=int(t_end/dt) # Valor de Los frame que usa
t=np.arange(t0,t_end+dt,dt) # Array de tiempo

Kp=0.022 # Proportional constant for the 1st tank
Ki=0.00005 # Proportional constant for the 2nd tank
Kd=0.02 # Proportional constant for the 3rd tank

Z_i=np.zeros(len(t)) # Crea array de valor verdadero de La profundidad
z_dot=np.zeros(len(t)) # Crea array de La velocidad
z_dot_dot=np.zeros(len(t)) # Crea array de La aceleracion
z_dot_dot_dot=np.zeros(len(t)) # Crea array del jerk
e=np.zeros(len(t)) # Crea array del error
e_dot=np.zeros(len(t)) # Crea array de La deriva del error
e_int=np.zeros(len(t)) # Crea array de La integral del error
Z_r=np.zeros(len(t)) # Crea array de referencia de La profundidad
V_i=np.zeros(len(t))

Z_o_i=0 # Profundidad inicial del planeador
Z_r_i=-100 # Profundidad de referencia del planeador
z_dot_i=0 # Velocidad inicial del planeador
z_dot_dot_i=0 # Aceleracion inicial del planeador
z_dot_dot_dot_i=0 # Jerk inicial del planeador

Z_r[0]=Z_r_i # Pone valor inicial de La referencia
Z_i[0]=Z_o_i # Pone valor inicial de La profundidad
z_dot[0]=z_dot_i # Pone valor inicial de velocidad

```

```
z_dot_dot[0]=z_dot_dot_i # Pone valor inicial de aceleracion
z_dot_dot_dot[0]=z_dot_dot_dot_i # Pone valor inicial de jerk
q=Kp*e+Ki*e_int+Kd*e_dot # Crea array para el caudal

# Determina la referencia
Z_r=1.003**(-t+1672)-150

# Empieza simulacion
for i in range(1,len(t)): # Itera el bucle de simulacion cada dt
    e[i-1]=Z_r[i-1]-Z_i[i-1]
    if i>1:
        e_dot[i-1]=(e[i-1]-e[i-2])/dt
        e_int[i-1]=e_int[i-2]+(e[i-2]+e[i-1])/2*dt
    if i==len(t)-1:
        e[-1]=e[-2]
        e_dot[-1]=e_dot[-2]
        e_int[-1]=e_int[-2]

    # Entrada del controlador
    q[i]=Kp*e[i-1]+Ki*e_int[i-1]+Kd*e_dot[i-1]
    if V_i[i-1]<=0 and q[i]<=0 :
        q[i]=0

    # Calcula la profundidad en el siguiente periodo (Ley de trapezoidal)
    V_i[i]=V_i[i-1]+q[i-1]*dt
    z_dot_dot_dot[i]=(densidad_water*g*q[i-1]/m)-((Cd*densidad_water*As*abs(z_dot[i-1])*z_dot_dot[i-1])/m)
    z_dot_dot[i]=(densidad_water*g*V_i[i-1]+Bgb-Wdb-Cd*densidad_water*abs(z_dot[i-1])*z_dot[i-1]*As/2)/m
    z_dot[i]=z_dot[i-1]+(z_dot_dot[i-1]+z_dot_dot[i])/2*dt
    Z_i[i]=Z_i[i-1]+(z_dot[i-1]+z_dot[i])/2*dt

# Empieza animacion
V_L=V_i*1000
q_L=q*1000*60
Z_r_2=Z_r

def update_plot(num):
    if num>=len(Z_i):
        num=len(Z_i)-1

    Z.set_data(t[0:num],Z_i[0:num])
    Z_r.set_data([-5,5],[Z_r_2[num],Z_r_2[num]])
    Z_i_line.set_data([-0.5,0.5],[Z_i[num],Z_i[num]])

    v.set_data(t[0:num],z_dot[0:num])
    a.set_data(t[0:num],z_dot_dot[0:num])
    V.set_data(t[0:num],V_L[0:num])
    Q.set_data(t[0:num],q_L[0:num])
```

```

    return Z_r,Z,Z_i_line,\
           v,a,V,Q

# Configura la demostración de datos
fig=plt.figure(figsize=(16,9),dpi=120,facecolor=(0.8,0.8,0.8))
gs=gridspec.GridSpec(4,3)

# El planeador
ax0=fig.add_subplot(gs[0:2,0],facecolor=(0.9,0.9,0.9))
Z_r,=ax0.plot([],[],'r',linewidth=2)
Z_i_line,=ax0.plot([],[],'black',linewidth=10)
plt.xlim(-5,5)
plt.ylim(-200,0)
plt.ylabel('La profundidad [m]')
plt.title('El sistema de flotabilidad')

# Velocidad
ax1=fig.add_subplot(gs[0,1],facecolor=(0.9,0.9,0.9))
v,=ax1.plot([],[],'blue',linewidth=2,label='z_dot')
plt.xlim(0,t_end)
plt.ylim(-6,6)
plt.ylabel('La velocidad [m/s]')
plt.grid(True)
plt.legend(loc='upper right',fontsize='small')

# Acercelacion
ax2=fig.add_subplot(gs[1,1],facecolor=(0.9,0.9,0.9))
a,=ax2.plot([],[],'blue',linewidth=2,label='z_dot_dot')
plt.xlim(0,t_end)
plt.ylim(-20,20)
plt.ylabel('La acercelacion [m/s^2]')
plt.grid(True)
plt.legend(loc='upper right',fontsize='small')

# Volumen vejiga
ax3=fig.add_subplot(gs[0,2],facecolor=(0.9,0.9,0.9))
V,=ax3.plot([],[],'blue',linewidth=2,label='Volumen')
plt.xlim(0,t_end)
plt.ylim(-0,1)
plt.ylabel('Volumen vejiga [l]')
plt.grid(True)
plt.legend(loc='upper right',fontsize='small')

# Caudal
ax4=fig.add_subplot(gs[1,2],facecolor=(0.9,0.9,0.9))
Q,=ax4.plot([],[],'blue',linewidth=2,label='Caudal')
plt.xlim(0,t_end)
plt.ylim(-10,10)
plt.ylabel('Caudal [l/min]')
plt.grid(True)

```

```
plt.legend(loc='upper right',fontsize='small')

# Como se actua el planeador
ax6=fig.add_subplot(gs[2:4,:], facecolor=(0.9,0.9,0.9))
Z,=ax6.plot([],[],'blue',linewidth=2,label='El planeador')
plt.plot(t,Z_r_2,'r',linewidth=2,label='Z ref')
plt.xlim(0,t_end)
plt.ylim(-200,0)
plt.ylabel('La profundidad [m]')
plt.grid(True)
plt.legend(loc='upper right',fontsize='small')

plane_ani=animation.FuncAnimation(fig,update_plot,
    frames=frame_amount,interval=0.001,repeat=True,blit=True)
plt.show()

#####
# Grafica fija
# El planeador
plt.plot(t,Z_r_2,'r',linewidth=2,label='Z ref')
plt.plot(t,Z_i,'b',linewidth=2,label='Z real')
plt.grid(True)
plt.xlabel('Tiempo [s]')
plt.ylabel('La profundidad [m]')
plt.title('El sistema de flotabilidad')
plt.show()

# Velocidad
plt.subplot(2,1,1)
plt.plot(t,z_dot,'b',linewidth=2,label='z dot')
plt.grid(True)
plt.xlabel('Tiempo [s]')
plt.ylabel('Velocidad [m/s]')
plt.title('La velocidad')

# Acercelacion
plt.subplot(2,1,2)
plt.plot(t,z_dot_dot,'b',linewidth=2,label='z dot dot')
plt.grid(True)
plt.xlabel('Tiempo [s]')
plt.ylabel('Acercelacion [m/s^2]')
plt.title('La acercelacion')
plt.show()

# Volumen vejiga
plt.subplot(2,1,1)
plt.plot(t,V_L,'b',linewidth=2,label='Volumen')
plt.grid(True)
plt.xlabel('Tiempo [s]')
plt.ylabel('Volumen [l]')
```

```
plt.title('El volumen')

# Caudal
plt.subplot(2,1,2)
plt.plot(t,q_L,'b',linewidth=2,label='Caudal')
plt.grid(True)
plt.xlabel('Tiempo [s]')
plt.ylabel('Caudal [l/min]')
plt.title('El caudal')
plt.show()
```

ANEXO VII CÓDIGO SIMULACIÓN MPC (MAIN FILE)

```
import numpy as np
import matplotlib.pyplot as plt
import support_files_buoyancy as sfb
import matplotlib.gridspec as gridspec
import matplotlib.animation as animation
np.set_printoptions(suppress=True)

# Crea el objeto de soporte
support=sfb.SupportFilesBuoyancy()
constants=support.constants

# Carga Los constante
Ts=constants['Ts']
outputs=constants['outputs'] # número de salidas (Z, z_dot, z_dot_dot)
hz = constants['hz'] # número de La prediccion
time_length=constants['time_length'] # Duracion de La simulacion
inputs=constants['inputs'] # Numero de entradas

# Generacion de señal de referencia
t=np.zeros((int(time_length/Ts+1)))

for i in range(1,len(t)):
    t[i]=np.round(t[i-1]+Ts,2)

# Crea Las referencias
Z_ref,z_dot_ref,z_dot_dot_ref=support.generador_ref(t)
sim_length=len(t) # Numero de bucles
refSignals=np.zeros(len(Z_ref)*outputs)

# Crea Los vectores de referencia:
# refSignal = [x_dot_ref_0, psi_ref_0, X_ref_0, Y_ref_0, x_dot_ref_1, psi_ref_1,
# X_ref_1, Y_ref_1, x_dot_ref_2, psi_ref_2, X_ref_2, Y_ref_2, ... etc.]
k=0
for i in range(0,len(refSignals),outputs):
    refSignals[i]=Z_ref[k]
    refSignals[i+1]=z_dot_ref[k]
    refSignals[i+2]=z_dot_dot_ref[k]
    k=k+1

# Crea estados iniciales
Z_i=0.
z_dot=0.
z_dot_dot=0.

states=np.array([Z_i,z_dot,z_dot_dot])
statesTotal=np.zeros((len(t),len(states))) # Traquear los valores de salidas durante
tiempos
statesTotal[0][0:len(states)]=states
```

```
#####

# Valor inicio de la entrada
Q_i=0. # Caudal 0 m^3/s
UTotal=np.zeros(len(t)) # Traquear Los valores de entradas durante tiempos
UTotal[0]=Q_i
V_i=0.
VTotal=np.zeros(len(t)) # Traquear Los volumenenes
VTotal[0]=V_i

# Empieza el controlador
k=0
du=np.zeros((inputs*hz,1))

# Arrays para animacion
t_ani=[]
Z_ani=[]
z_dot_ani=[]
z_dot_dot_ani=[]

for i in range(0,sim_length-1):
    # Genera Las matrices discretos
    Ad,Bd,Cd,Dd=support.state_space(states)

    # Genera el vector de estados
    x_g=np.transpose([(states)])

    # Sacar Las referencias necesarias
    k=k+outputs
    if k+outputs*hz<=len(refSignals):
        r=refSignals[k:k+outputs*hz]
    else:
        r=refSignals[k:len(refSignals)]
        hz=hz-1

    # Genera matrices para la funcion de Los costes
    Hdb,Fdbt,Cdb,Adc=support.mpc_simplification(Ad,Bd,Cd,Dd,hz)
    ft=np.matmul(np.concatenate((np.transpose(x_g)[0][0:len(x_g)]),r),axis=0),Fdbt)

#####

# Calcula Las entradas predictados
U1=-np.matmul(np.linalg.inv(Hdb),np.transpose([ft]))
U=U1[0]
UTotal[i+1]=U
VTotal[i+1]=VTotal[i]+U*Ts

states=support.open_loop_new_states(states,U)
statesTotal[i+1][0:len(states)]=states
```

```
##### ANIMATION LOOP #####
frame_amount=int(time_length/Ts)
def update_plot(num):
    if num>=len(t):
        num=len(t)-1

    Z.set_data(t[0:num],statesTotal[0:num,0])
    Z_r.set_data([-5,5],[Z_ref[num],Z_ref[num]])
    Z_i_line.set_data([-0.5,0.5],[statesTotal[num,0],statesTotal[num,0]])
    v.set_data(t[0:num],statesTotal[0:num,1])
    a.set_data(t[0:num],statesTotal[0:num,2])
    V.set_data(t[0:num],VTotal[0:num])
    Q.set_data(t[0:num],UTotal[0:num])

    return v,a,V,Q,Z,Z_r,Z_i_line

# Configura la demostración de datos
fig=plt.figure(figsize=(16,9),dpi=120,facecolor=(0.8,0.8,0.8))
gs=gridspec.GridSpec(4,3)

# El planeador
ax0=fig.add_subplot(gs[0:2,0],facecolor=(0.9,0.9,0.9))
Z_r,=ax0.plot([],[],'r',linewidth=2)
Z_i_line,=ax0.plot([],[],'black',linewidth=10)
plt.xlim(-5,5)
plt.ylim(-200,0)
plt.ylabel('La profundidad [m]')
plt.title('El sistema de flotabilidad')

# Velocidad
ax1=fig.add_subplot(gs[0,1],facecolor=(0.9,0.9,0.9))
v,=ax1.plot([],[],'blue',linewidth=2,label='z_dot')
plt.xlim(0,time_length)
plt.ylim(-6,6)
plt.ylabel('La velocidad [m/s]')
plt.grid(True)
plt.legend(loc='upper right',fontsize='small')

# Acercelacion
ax2=fig.add_subplot(gs[1,1],facecolor=(0.9,0.9,0.9))
a,=ax2.plot([],[],'blue',linewidth=2,label='z_dot_dot')
plt.xlim(0,time_length)
plt.ylim(-20,20)
plt.ylabel('La acercelacion [m/s^2]')
plt.grid(True)
plt.legend(loc='upper right',fontsize='small')

# Volumen vejiga
ax3=fig.add_subplot(gs[0,2],facecolor=(0.9,0.9,0.9))
V,=ax3.plot([],[],'blue',linewidth=2,label='Volumen')
```

```

plt.xlim(0,time_length)
plt.ylim(-0,1)
plt.ylabel('Volumen vejiga [l]')
plt.grid(True)
plt.legend(loc='upper right',fontsize='small')

# Caudal
ax4=fig.add_subplot(gs[1,2],facecolor=(0.9,0.9,0.9))
Q,=ax4.plot([],[],'blue',linewidth=2,label='Caudal')
plt.xlim(0,time_length)
plt.ylim(-10,10)
plt.ylabel('Caudal [l/min]')
plt.grid(True)
plt.legend(loc='upper right',fontsize='small')

# Como se actua el planeador
ax6=fig.add_subplot(gs[2:4,:], facecolor=(0.9,0.9,0.9))
Z,=ax6.plot([],[],'blue',linewidth=2,label='El planeador')
plt.plot(t,Z_ref,'r',linewidth=2,label='Z ref')
plt.xlim(0,time_length)
plt.ylim(-200,0)
plt.ylabel('La profundidad [m]')
plt.grid(True)
plt.legend(loc='upper right',fontsize='small')

plane_ani=animation.FuncAnimation(fig,update_plot,
    frames=frame_amount,interval=1,repeat=True,blit=True)
plt.show()
#####

# El planeador
plt.plot(t,Z_ref,'r',linewidth=2,label='Z ref')
plt.plot(t,statesTotal[0:,0],'b',linewidth=2,label='Z real')
plt.grid(True)
plt.xlabel('Tiempo [s]')
plt.ylabel('La profundidad [m]')
plt.title('El sistema de flotabilidad')
plt.show()

# Velocidad
plt.subplot(2,1,1)
plt.plot(t,statesTotal[0:,1],'b',linewidth=2,label='z dot')
plt.grid(True)
plt.xlabel('Tiempo [s]')
plt.ylabel('Velocidad [m/s]')
plt.title('La velocidad')

# Acercelacion
plt.subplot(2,1,2)
plt.plot(t,statesTotal[0:,2],'b',linewidth=2,label='z dot dot')

```

```
plt.grid(True)
plt.xlabel('Tiempo [s]')
plt.ylabel('Acerelacion [m/s^2]')
plt.title('La aceleracion')
plt.show()

# Volumen vejiga
plt.subplot(2,1,1)
plt.plot(t,VTotal,'b',linewidth=2,label='Volumen')
plt.grid(True)
plt.xlabel('Tiempo [s]')
plt.ylabel('Volumen [l]')
plt.title('El volumen')

# Caudal
plt.subplot(2,1,2)
plt.plot(t,UTotal,'b',linewidth=2,label='Caudal')
plt.grid(True)
plt.xlabel('Tiempo [s]')
plt.ylabel('Caudal [l/min]')
plt.title('El caudal')
plt.show()
```

ANEXO VIII CÓDIGO SIMULACIÓN MPC (SUPPORT FILE)

```

import numpy as np
import matplotlib.pyplot as plt

class SupportFilesBuoyancy:

    def __init__(self):

        # Constantes
        D=120 # Diametro del tubo en mm
        l=1000 # Longitud del tubo en mm
        m=20 # kg
        g=9.81 # [m/s^2]
        Wdb=m*g # Peso del planeador.
        As=l*D*10**(-6) # Sección flotante
        final_b=-100 # profundidad maxima
        dVol=10 # The change of volume on the vertical scale.
        densidad_water=1025 # [kg/m^3]
        densidad_oil=885 # [kg/m^3]
        Cd=0.4 # Coeficiente de resistencia en el agua
        Bgb=densidad_water*g*np.pi*D**(2)/4*l*10**(-9)
        Ts=0.02

        ##### Heave control #####

        outputs=3 # numero de salidas
        inputs=1 # numero de entradas
        hz = 10 # numero de las predicciones
        Q=np.matrix('1 0 0;0 1 0;0 0 1') # Peso de las salidas
        S=np.matrix('1 0 0;0 1 0;0 0 1') # Peso de la salida final de la prediccion
        R=np.matrix('1') # Peso de la entrada

        time_length=1800

        self.constants={'g':g,'m':m,'densidad_water':densidad_water,'Cd':Cd,'As':As,'
Ts':Ts,\
        'Q':Q,'S':S,'R':R,'outputs':outputs,'inputs':inputs,'hz':hz,'time_length':tim
e_length,}
        return None

    def generador_ref(self,t):

        Ts=self.constants['Ts']
        time_length=self.constants['time_length']

        # Define trayectoria
        Z_r=np.zeros(len(t)) # Crea array de referencia de la profundidad

```

```
# Z_r=1.003**(-t+1672)-150 # Crea trayectoria exponencial
# for i in range(0,len(t)):
#     if i < len(t)*4/10:
#         Z_r[i]=-150
#     elif i < len(t)*6/10:
#         Z_r[i]=Z_r[i-1]+140/len(t)*10/2
#     else:
#         Z_r[i]=-10
Z_r=0.000000086*(t-1000)**(3)-101
# Vectors de cambio
dZ=Z_r[1:len(Z_r)]-Z_r[0:len(Z_r)-1]

z_dot=dZ/Ts
z_dot=np.concatenate(([z_dot[0]],z_dot),axis=0)
dz_dot=z_dot[1:len(z_dot)]-z_dot[0:len(z_dot)-1]

z_dot_dot=dz_dot/Ts
z_dot_dot=np.concatenate(([z_dot_dot[0]],z_dot_dot),axis=0)

return Z_r,z_dot,z_dot_dot

def state_space(self,states):

    # Constantes necesarios
    m=self.constants['m']
    g=self.constants['g']
    densidad_water=self.constants['densidad_water']
    Cd=self.constants['Cd']
    As=self.constants['As']
    Ts=self.constants['Ts']

    # Estados necesarios
    A33=-Cd*densidad_water*As
    B13=densidad_water*g/m

    # Matrices de espacio de Los estados del control
    A=np.array([[0, 1, 0],[0, 0, 1],[0, 0, A33]])
    B=np.array([[0],[0],[B13]])
    C=np.array([[1, 0, 0],[0, 1, 0],[0, 0, 1]])
    D=0

    # Sistema discreto (forward Euler)
    Ad=np.identity(np.size(A,1))+Ts*A
    Bd=Ts*B
    Cd=C
    Dd=D

    return Ad, Bd, Cd, Dd

def augmented_matrices(self, Ad, Bd, Cd, Dd):
```

```

A_aug=np.concatenate((Ad,Bd),axis=1)
temp1=np.zeros((np.size(Bd,1),np.size(Ad,1)))
temp2=np.identity(np.size(Bd,1))
temp=np.concatenate((temp1,temp2),axis=1)

A_aug=np.concatenate((A_aug,temp),axis=0)
B_aug=np.concatenate((Bd,np.identity(np.size(Bd,1))),axis=0)
C_aug=np.concatenate((Cd,np.zeros((np.size(Cd,0),np.size(Bd,1)))),axis=1)
D_aug=Dd

return A_aug, B_aug, C_aug, D_aug

def mpc_simplification(self, Ad, Bd, Cd, Dd, hz):
    # db - double bar
    # dbt - double bar transpose
    # dc - double circumflex

    # A_aug, B_aug, C_aug, D_aug=self.augmented_matrices(Ad, Bd, Cd, Dd)
    A_aug, B_aug, C_aug, D_aug = Ad, Bd, Cd, Dd

    Q=self.constants['Q']
    S=self.constants['S']
    R=self.constants['R']
    #####

    CQC=np.matmul(np.transpose(C_aug),Q)
    CQC=np.matmul(CQC,C_aug)

    CSC=np.matmul(np.transpose(C_aug),S)
    CSC=np.matmul(CSC,C_aug)

    QC=np.matmul(Q,C_aug)
    SC=np.matmul(S,C_aug)

    Qdb=np.zeros((np.size(CQC,0)*hz,np.size(CQC,1)*hz))
    Tdb=np.zeros((np.size(QC,0)*hz,np.size(QC,1)*hz))
    Rdb=np.zeros((np.size(R,0)*hz,np.size(R,1)*hz))
    Cdb=np.zeros((np.size(B_aug,0)*hz,np.size(B_aug,1)*hz))
    Adc=np.zeros((np.size(A_aug,0)*hz,np.size(A_aug,1)))

    for i in range(0,hz):
        if i == hz-1:
            Qdb[np.size(CSC,0)*i:np.size(CSC,0)*i+CSC.shape[0],np.size(CSC,1)*i:n
p.size(CSC,1)*i+CSC.shape[1]]=CSC
            Tdb[np.size(SC,0)*i:np.size(SC,0)*i+SC.shape[0],np.size(SC,1)*i:np.si
ze(SC,1)*i+SC.shape[1]]=SC
        else:
            Qdb[np.size(CQC,0)*i:np.size(CQC,0)*i+CQC.shape[0],np.size(CQC,1)*i:n
p.size(CQC,1)*i+CQC.shape[1]]=CQC

```

```
Tdb[np.size(QC,0)*i:np.size(QC,0)*i+QC.shape[0],np.size(QC,1)*i:np.size(QC,1)*i+QC.shape[1]]=QC

Rdb[np.size(R,0)*i:np.size(R,0)*i+R.shape[0],np.size(R,1)*i:np.size(R,1)*i+R.shape[1]]=R

for j in range(0,hz):
    if j<=i:
        Cdb[np.size(B_aug,0)*i:np.size(B_aug,0)*i+B_aug.shape[0],np.size(B_aug,1)*j:np.size(B_aug,1)*j+B_aug.shape[1]]=np.matmul(np.linalg.matrix_power(A_aug,((i+1)-(j+1))),B_aug)

        Adc[np.size(A_aug,0)*i:np.size(A_aug,0)*i+A_aug.shape[0],0:0+A_aug.shape[1]]=np.linalg.matrix_power(A_aug,i+1)

        Hdb=np.matmul(np.transpose(Cdb),Qdb)
        Hdb=np.matmul(Hdb,Cdb)+Rdb

        temp=np.matmul(np.transpose(Adc),Qdb)
        temp=np.matmul(temp,Cdb)

        temp2=np.matmul(-Tdb,Cdb)
        Fdbt=np.concatenate((temp,temp2),axis=0)

    return Hdb,Fdbt,Cdb,Adc

def open_loop_new_states(self,states,U):

    # Constantes necesarios
    m=self.constants['m']
    g=self.constants['g']
    densidad_water=self.constants['densidad_water']
    Cd=self.constants['Cd']
    As=self.constants['As']
    Ts=self.constants['Ts']

    current_states=states
    new_states=current_states
    Z_i=current_states[0]
    z_dot=current_states[1]
    z_dot_dot=current_states[2]

    sub_loop=30 #Trozaer en 30 piezas
    for i in range(0,sub_loop):
        # Calcular el Jerk
        z_dot_dot_dot=(densidad_water*g*U/m)-
        ((Cd*densidad_water*As*abs(z_dot)*z_dot_dot)/m)

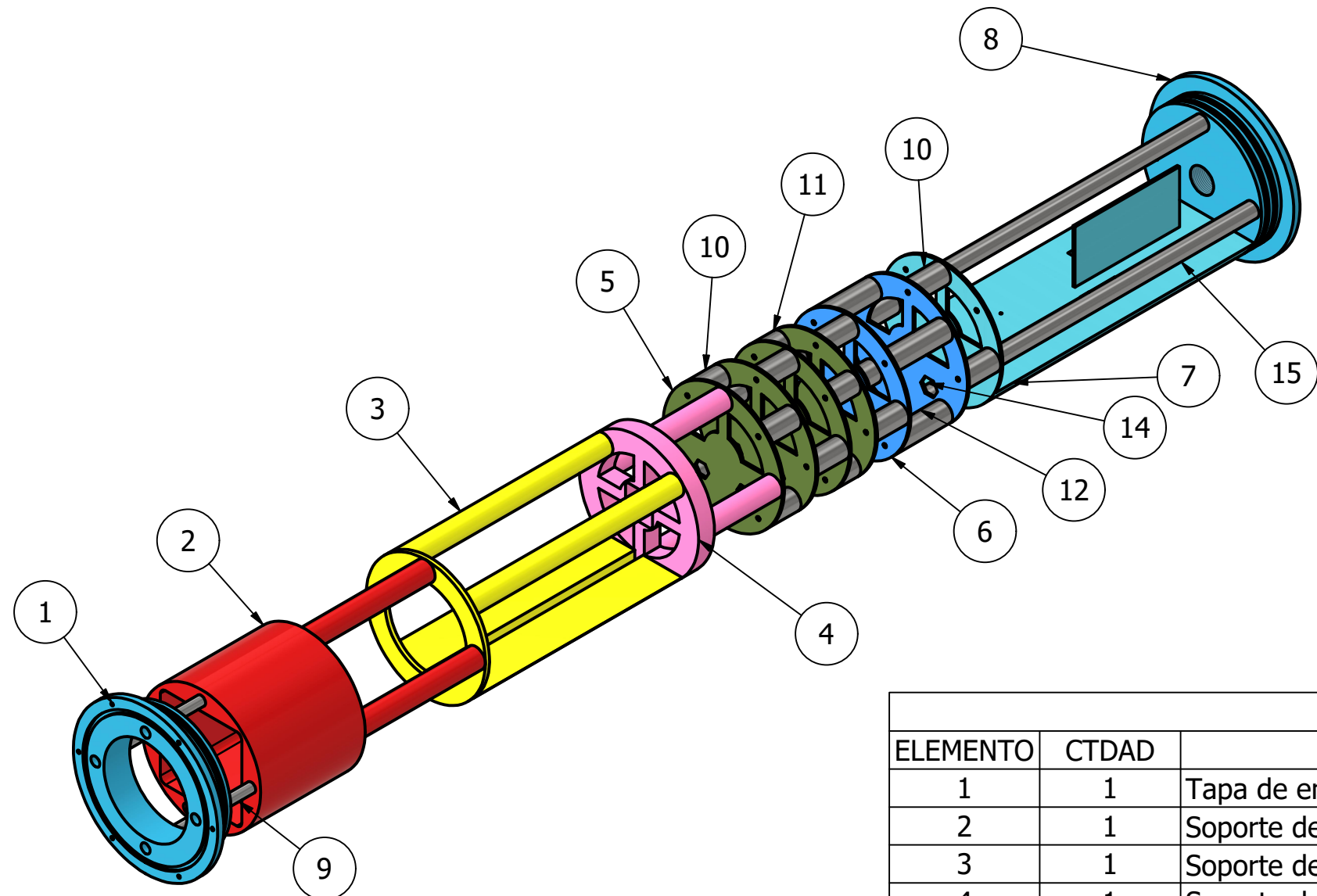
        # Actualizar los valores de estados
        z_dot_dot_dot=z_dot_dot_dot+z_dot_dot_dot*Ts/sub_loop
```

```
z_dot=z_dot+z_dot_dot*Ts/sub_loop
Z_i=Z_i+z_dot*Ts/sub_loop

# Toma los valores nuevos
new_states[0]=Z_i
new_states[1]=z_dot
new_states[2]=z_dot_dot

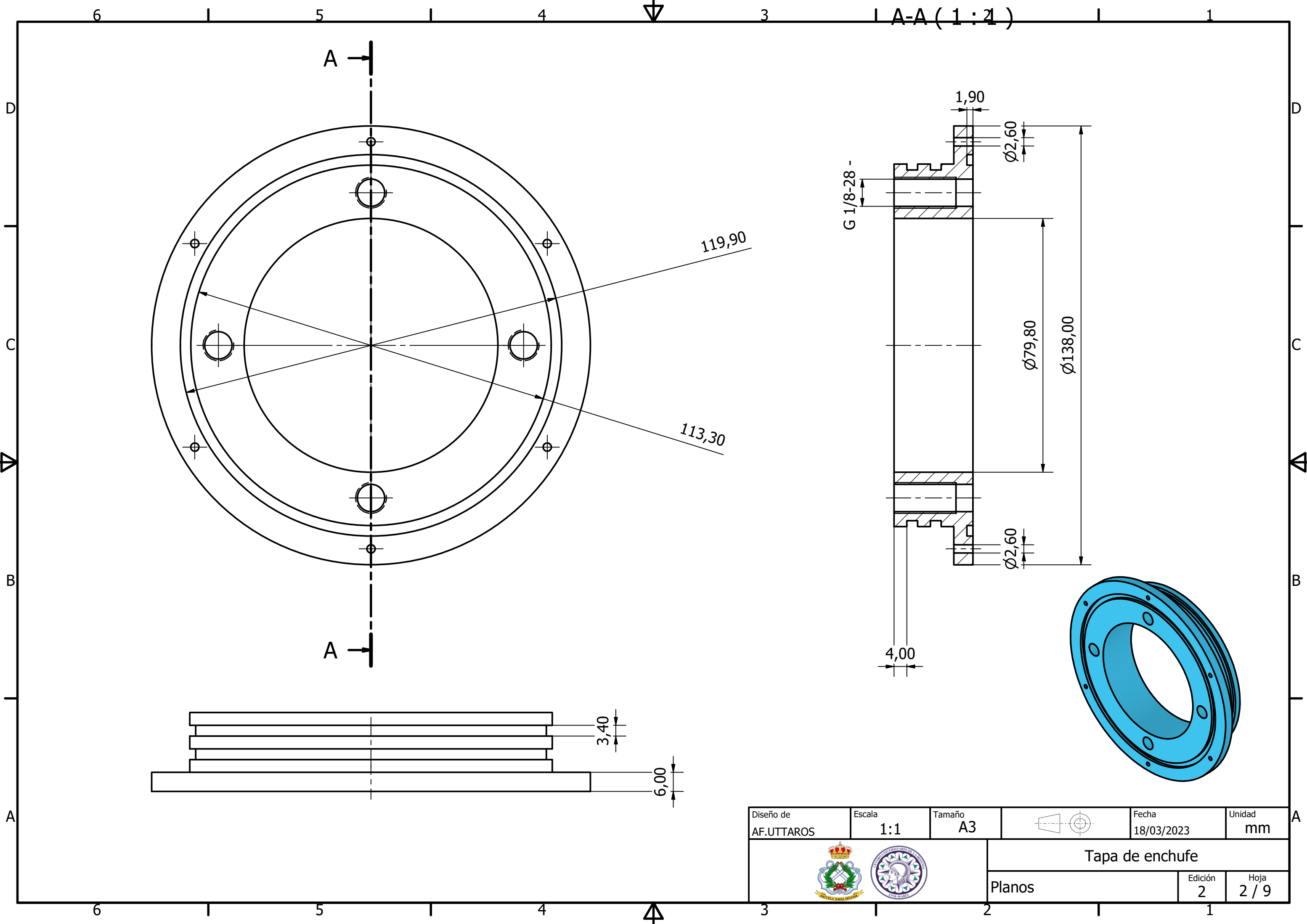
return new_states
```

ANEXO IX PLANOS

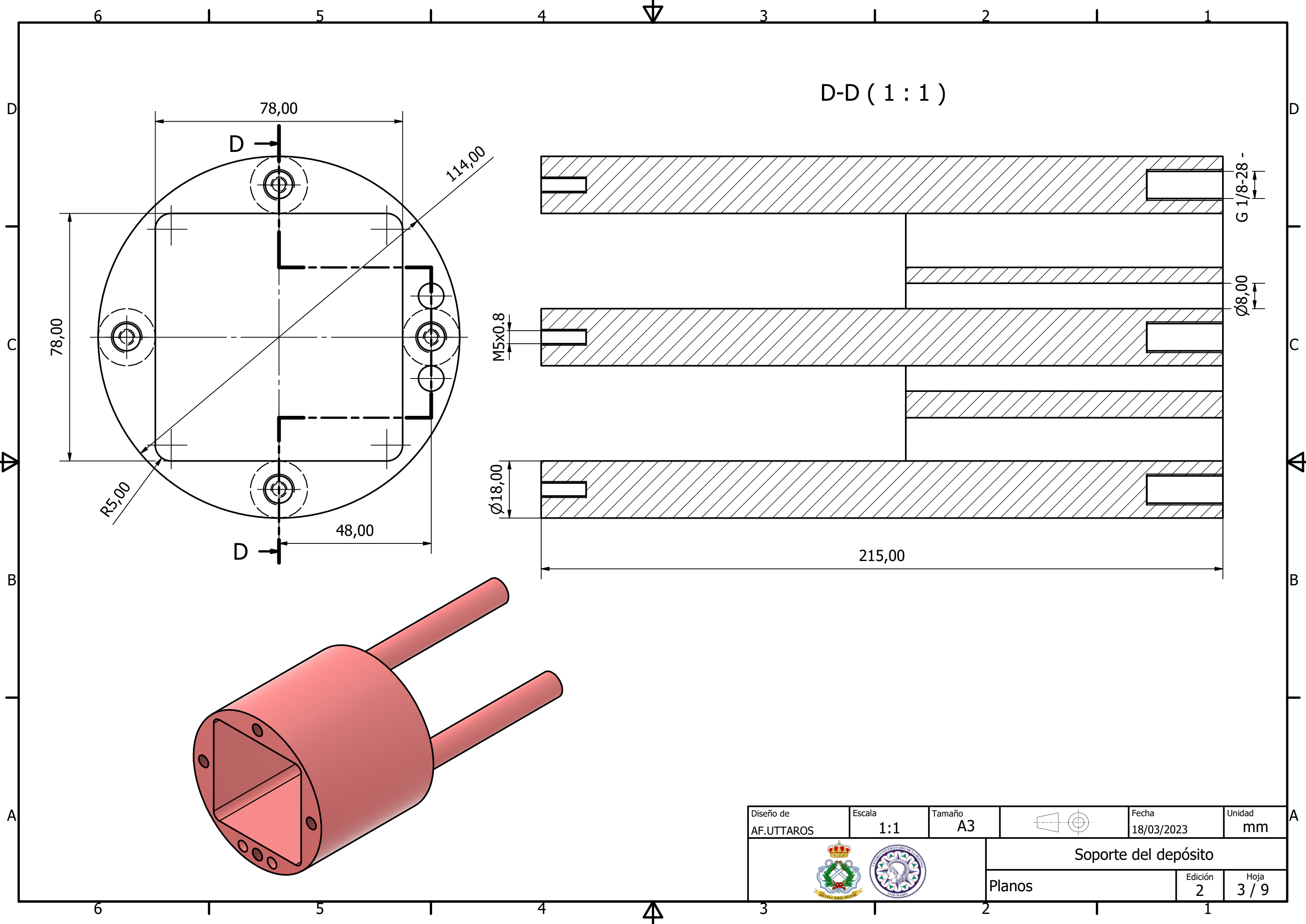


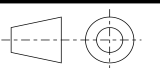
LISTA DE PIEZAS		
ELEMENTO	CTDAD	Nº DE PIEZA
1	1	Tapa de enchufe
2	1	Soporte del depósito
3	1	Soporte del conjunto de la placa electrónica y sistema de control
4	1	Soporte de la bomba
5	4	Soportes de válvulas
6	2	Soportes de antirretornos
7	1	Soporte del conjunto de tuberías y sensores
8	1	Tapa final
9	4	Barra 30 mm roscado doble
10	4	Barra 30 mm
11	4	Barra 23 mm
12	4	Barra 54 mm
14	1	Barra 30 mm roscado
15	3	Barra 247.5 mm roscado

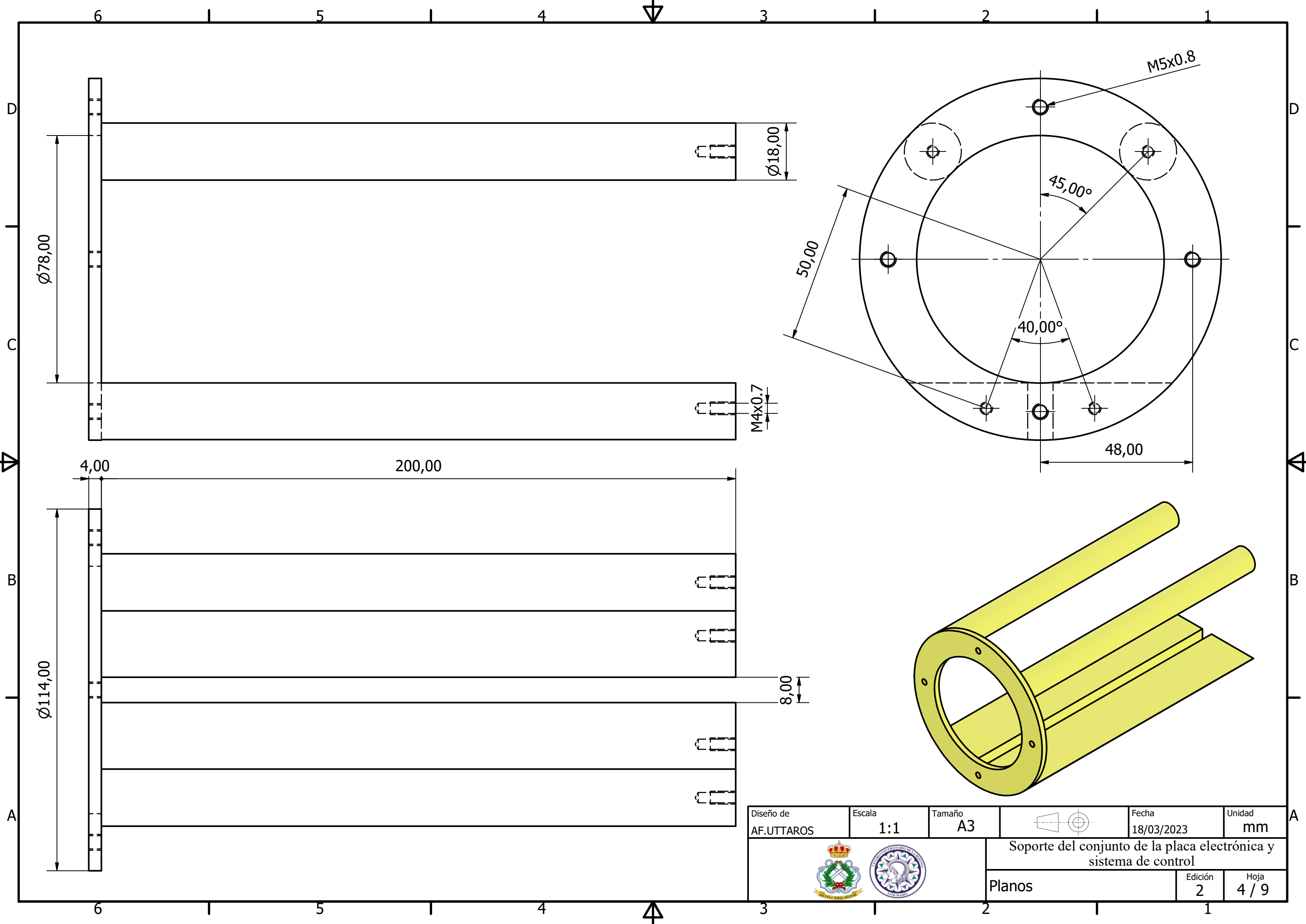
Diseño de AF.UTTAROS	Escala 1:4	Tamaño A3		Fecha 18/03/2023	Unidad mm
 			Ensamblaje		
			Planos	Edición 2	Hoja 1 / 9

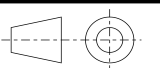


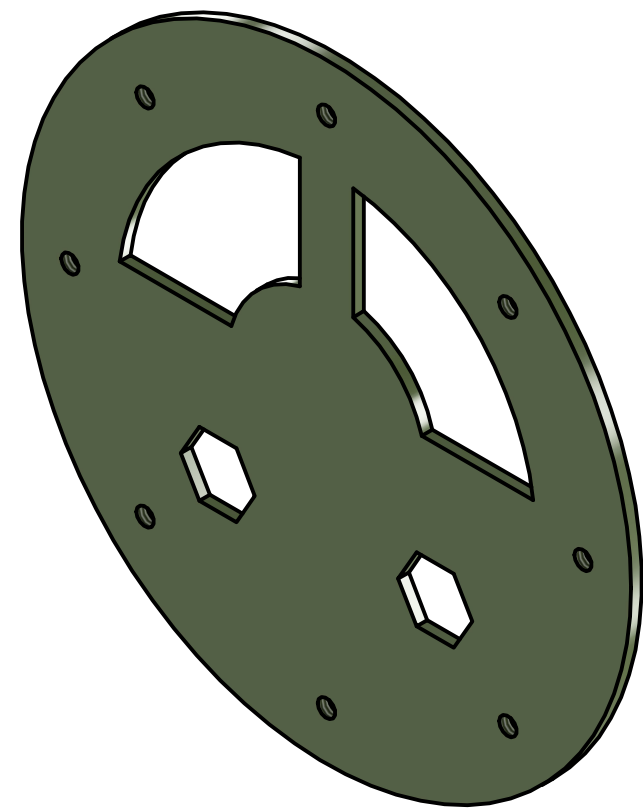
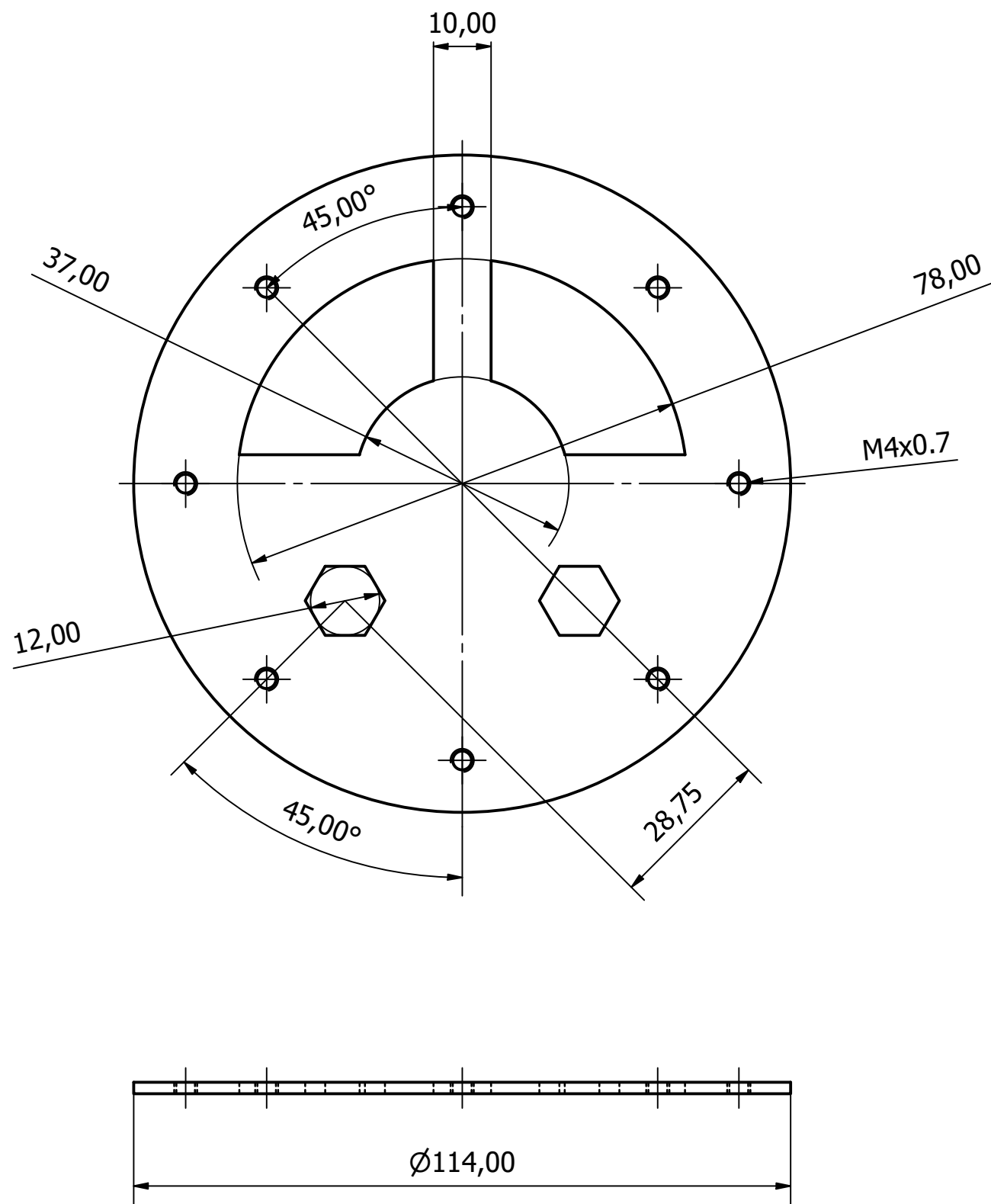
Diseño de AF.UTTAROS	Escala 1:1	Tamaño A3		Fecha 18/03/2023	Unidad mm
 			Tapa de enchufe		
			Planos	Edición 2	Hoja 2 / 9



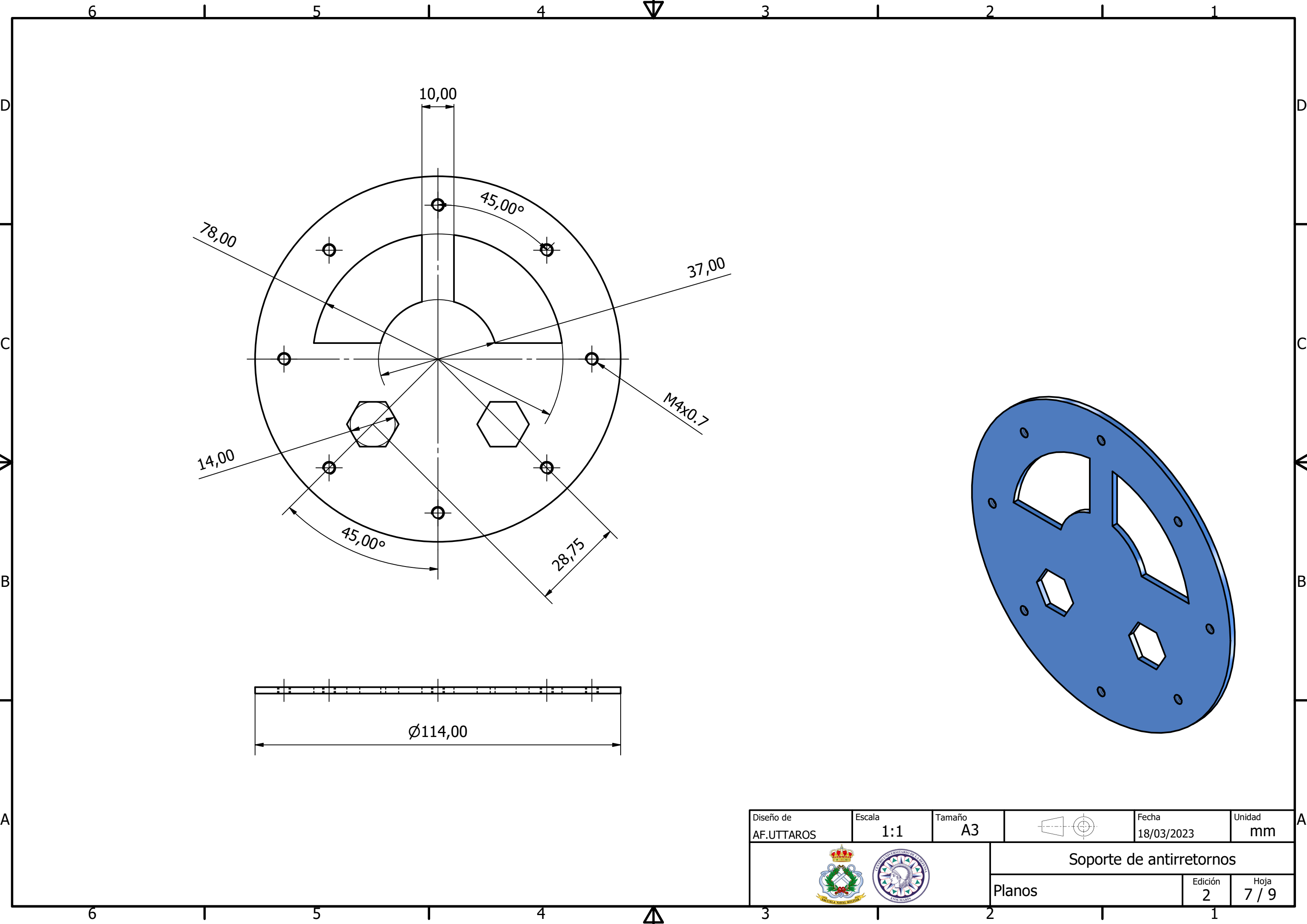
Diseño de AF.UTTAROS	Escala 1:1	Tamaño A3		Fecha 18/03/2023	Unidad mm
 			Soporte del depósito		
			Planos	Edición 2	Hoja 3 / 9

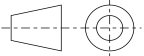


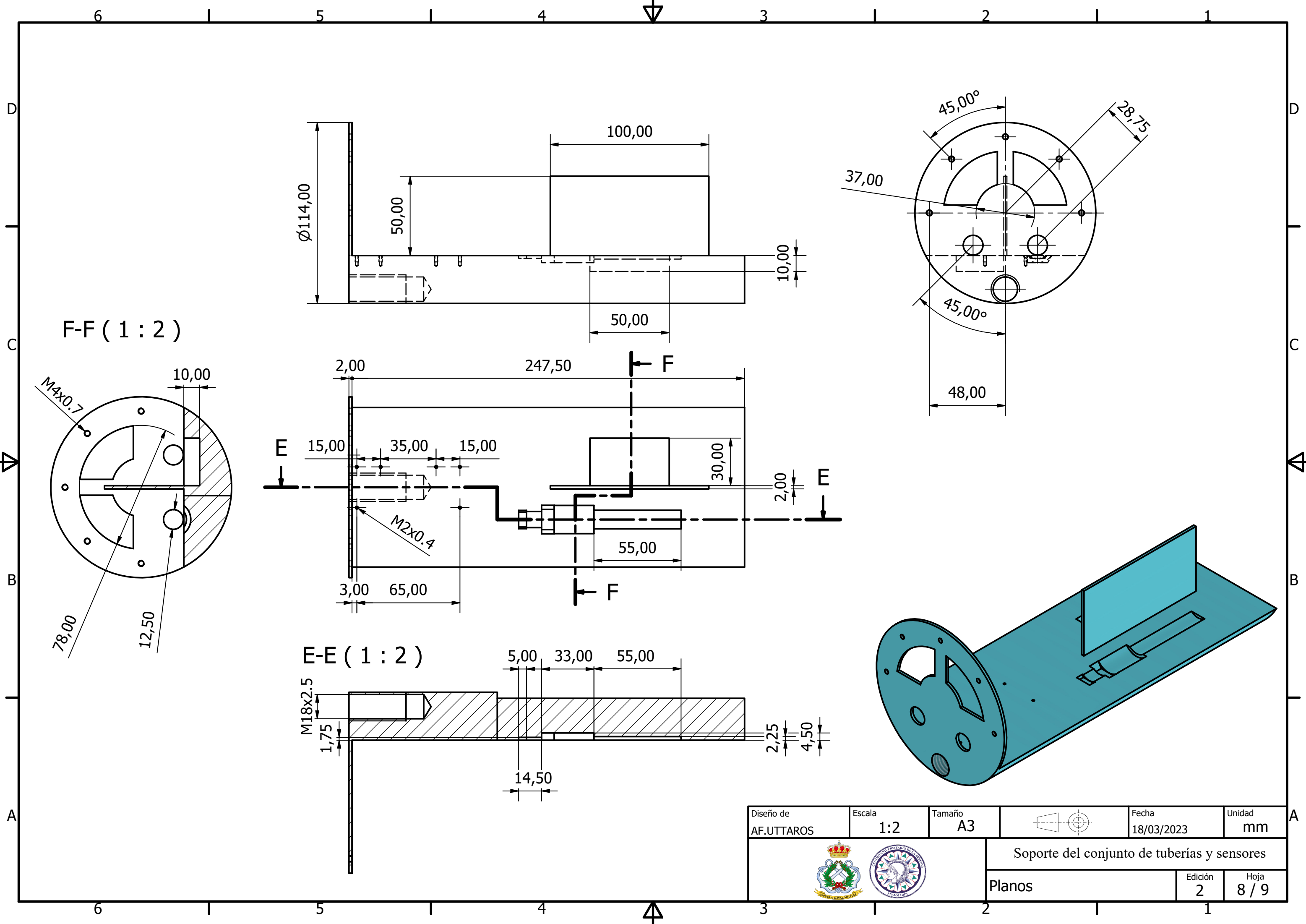
Diseño de AF.UTTAROS	Escala 1:1	Tamaño A3		Fecha 18/03/2023	Unidad mm
 			Soporte del conjunto de la placa electrónica y sistema de control		
			Planos	Edición 2	Hoja 4 / 9

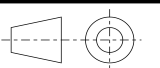


Diseño de AF.UTTAROS	Escala 1:1	Tamaño A3		Fecha 18/03/2023	Unidad mm
 			Soporte de válvulas		
			Planos	Edición 2	Hoja 6 / 9



Diseño de AF.UTTAROS	Escala 1:1	Tamaño A3		Fecha 18/03/2023	Unidad mm
 			Soporte de antirretornos		
			Planos	Edición 2	Hoja 7 / 9



Diseño de AF.UTTAROS	Escala 1:2	Tamaño A3		Fecha 18/03/2023	Unidad mm
 			Soporte del conjunto de tuberías y sensores		
			Planos	Edición 2	Hoja 8 / 9

