



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS**  
**APLICADAS**  
**CARRERA DE INGENIERÍA MECATRÓNICA**

**TRABAJO DE INTEGRACIÓN CURRICULAR**

**TEMA:**

“SISTEMA DIGITAL PARA LECTURA DE CONSUMO DE AGUA  
POTABLE”

Trabajo de titulación previo a la obtención del título en ingeniero en  
Mecatrónica

**Línea de investigación:** Producción industrial y tecnológica  
sostenible.

**AUTOR:**

Franklin Javier Escobar Cacuango

**DIRECTOR:**

MSc. Cosme Damián Mejía Echeverría

**Ibarra – Ecuador 2025**



# UNIVERSIDAD TÉCNICA DEL NORTE

## BIBLIOTECA UNIVERSITARIA

### AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

#### 1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1728782861		
APELLIDOS Y NOMBRES:	Franklin Javier Escobar Cacuango		
DIRECCIÓN:	Cayambe, Juan Montalvo, Barrio Central, Calle Primero de Mayo		
EMAIL:	fescobar@utn.edu.ec		
TELÉFONO FIJO:		TELÉFONO MÓVIL:	0994252368

DATOS DE LA OBRA	
TÍTULO:	SISTEMA DIGITAL PARA LECTURA DE CONSUMO DE AGUA POTABLE
AUTOR (ES):	Franklin Javier Escobar Cacuango
FECHA: DD/MM/AAAA	06/09/2024
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TÍTULO POR EL QUE OPTA:	INGENIERO EN MECATRÓNICA
ASESOR /DIRECTOR:	MSc. Cosme Damián Mejía Echeverría Dr. Carlos Xavier Rosero

#### 2. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asumí (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 4 días del mes de Julio de 2025

EL AUTOR:

Nombre: Franklin Javier Escobar Cacuango



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**  
**CARRERA DE INGENIERÍA EN MECATRÓNICA**

**CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE  
INTEGRACIÓN CURRICULAR**

Ibarra, 2 de julio de 2025

MSc. Cosme Damián Mejía Echeverría

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del Trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.



Firmado electrónicamente por:  
**COSME DAMIAN MEJIA  
ECHEVERRIA**

Validar únicamente con FirmaRC

(f).....

MSc. Cosme Damián Mejía Echeverría

## **AGRADECIMIENTO**

En primer lugar, deseo expresar mi más sincero agradecimiento a la Universidad Técnica del Norte por la formación integral recibida durante mi trayectoria académica. A través de sus programas, docentes y recursos, la institución me ha dotado no solo de conocimientos teóricos y prácticos fundamentales, sino también de las competencias necesarias para desarrollarme como un profesional competitivo en el amplio y dinámico campo de la Ingeniería Mecatrónica. Cada aprendizaje adquirido ha sido y será un pilar esencial para mi crecimiento, permitiéndome enfrentar con solvencia los retos que demanda esta disciplina.

De manera muy especial, quiero extender mi gratitud al MSc. Cosme Mejía, director de este trabajo, por su invaluable guía, dedicación y conocimiento compartido durante todo este proceso. Su disposición para brindarme su tiempo y sus observaciones fueron determinantes para la culminación exitosa de este trabajo.

## **DEDICATORIA.**

A Dios, quien en estos últimos tiempos ha sido uno de mis pilares fundamentales, sembrando en mí una esperanza en los momentos de desánimo y siendo la luz que me guió cuando todo parecía oscuro.

A mis padres, que durante estos 24 años nunca se han apartado de mí. Supieron guiarme por el buen camino, a pesar de que los suyos no fueron fáciles ni sencillos, y aun así me dieron lo mejor que he podido conocer en este mundo. A ellos, mi eterno agradecimiento.

A mi hermana, una de las personas más bondadosas y cercanas que he conocido. Sin su apoyo emocional, sin la fuerza que transmitía en cada palabra y acción, esto no habría sido posible.

## **RESUMEN.**

Algunas de las principales dificultades identificadas en la lectura convencional de los medidores de agua son la poca visibilidad de los visores, la limitada accesibilidad a los medidores y los errores humanos en la toma manual de lecturas, los cuales pueden derivar en interpretaciones incorrectas y cobros imprecisos por consumo de agua. Por esta razón se planteó el desarrollo de un dispositivo digital de medición del agua potable. Para llevar a cabo este proyecto, se realizó una contextualización sobre el funcionamiento y las características de sensores de flujo de agua disponibles en el mercado, junto con un análisis de su compatibilidad con microcontroladores y memorias no volátiles. Posteriormente, se realizó la selección de los componentes del dispositivo usando una matriz morfológica, esto dio como resultado el uso de un microcontrolador ESP32, un diseño estructural tipo unidad montable y alimentación energética mixta. Por último, para la interacción del usuario con el dispositivo medidor, se desarrolló una Interfaz Humano Máquina (HMI), la cual fue programada usando lenguaje C# en Visual Studio. El dispositivo se conecta al HMI mediante un cable USB a micro USB para realizar una comunicación serial directa. Los resultados obtenidos muestran que el dispositivo es capaz de registrar el consumo de agua con un margen de error del 0,36 % en comparación con los medidores convencionales, garantizando así una alta precisión en las lecturas.

Palabras clave: Medidores de agua, ESP32, Agua potable, Comunicación serial

## **ABSTRACT**

Some of the main difficulties identified in the conventional reading of water meters are the poor visibility of the displays, the limited accessibility to the meters and human errors in the manual taking of readings, which can lead to incorrect interpretations and inaccurate charges for water consumption. For this reason, the development of a digital device for measuring drinking water was proposed. To carry out this project, a contextualization of the operation and characteristics of water flow sensors available in the market was carried out, together with an analysis of their compatibility with microcontrollers and non-volatile memories. Subsequently, the device components were selected using a morphological matrix, resulting in the use of an ESP32 microcontroller, a mountable unit type structural design and mixed power supply. Finally, for user interaction with the metering device, a Human Machine Interface (HMI) was developed, which was programmed using C# language in Visual Studio. The device is connected to

the HMI via a USB to micro USB cable for direct serial communication. The results obtained show that the device is capable of recording water consumption with a margin of error of 0,36 % compared to conventional meters, thus ensuring high accuracy in the readings.

Keywords: Water meters, ESP32, Drinking water, Serial Communication

## ÍNDICE DE CONTENIDOS

CAPÍTULO I. INTRODUCCIÓN.....	9
1.1 Planteamiento del Problema. ....	9
1.2 Objetivos.....	9
1.2.1 General. ....	9
1.2.2 Específicos. ....	9
1.3 Alcance y delimitación.....	9
1.4 Justificación .....	10
CAPITULO 2: MARCO TEÓRICO. ....	11
2.1 Antecedentes. ....	11
2.2 Bases Teóricas.....	12
2.2.2 Tipos de medidor de agua potable.....	13
2.2.3 Medidores inteligentes de agua potable. ....	14
2.2.3.4 Medidor Lain Holding.....	16
2.2.3.5 Medidor inteligente SHMETERS .....	17
2.2.4 Caudal del agua. ....	17
2.2.5 IOT .....	20
2.2.6 Placas de desarrollo o microcontrolador. ....	21
2.2.7 Memoria no volátil.....	25
2.2.8 Protocolos de Comunicación serial. ....	27
2.2.9. Modulo SD.....	30
2.9.2 Módulo de reloj RTC.....	30
CAPÍTULO 3: MARCO METODOLÓGICO. ....	32
3.1 Enfoque y tipo de investigación.....	32
3.2 Diseño de la investigación. ....	32
3.2.1 Fase 1: Contextualización de las características y el funcionamiento de los sensores de flujo disponibles.....	32

3.2.2 Fase 2: Revisión de las características de conexión y compatibilidad de los sensores con microcontroladores, memorias no volátiles. ....	33
3.2.3 Fase 3: Diseñar el sistema de medición digital. ....	33
3.2.4 Fase 4: Implementar el sistema de medición para su validación bajo condiciones reales de funcionamiento.....	34
CAPÍTULO 4: ANÁLISIS Y RESULTADOS .....	35
4.1 Criterios y restricciones .....	35
4.2 Alternativas de solución.....	35
4.3 Selección de la mejor solución .....	39
4.4 Solución propuesta para el sistema digital de medición .....	42
4.5 Construcción del sistema digital de medición. ....	45
4.6 Resultados de pruebas.....	60
4.7 Análisis de resultados.....	64
CONCLUSIONES.....	67
RECOMENDACIONES .....	68
REFERENCIAS BIBLIOGRÁFICAS .....	69
ANEXOS.....	72
ANEXO A. Código para obtener pulsos.....	72
ANEXO B. Código general del programa. ....	72
ANEXO C. Código del HMI. ....	86
ANEXO D. Planos.....	96
ANEXO E. Hojas de procesos.....	97

## ÍNDICE DE TABLAS

Tabla 1. Alternativas de solución del módulo de procesamiento de datos.....	36
Tabla 2. Alternativas de solución del módulo de rendimiento energético .....	36

Tabla 3.	Alternativas de solución del módulo de mecanismo de instalación. ....	37
Tabla 4.	Alternativas de solución .....	38
Tabla 5.	componentes de las alternativas .....	38
Tabla 6.	Evaluación de criterios .....	39
Tabla 7.	Comparación de alternativas con el criterio 1 .....	39
Tabla 8.	Comparación de alternativas con el criterio 2 .....	40
Tabla 9.	Comparación de alternativas con el criterio 3 .....	40
Tabla 10.	Comparación de alternativas con el criterio 4 .....	41
Tabla 11.	Comparación de alternativas con el criterio 5 .....	41
Tabla 12.	Evaluación de las alternativas frente a todos los criterios .....	42
Tabla 13.	Componentes de la solución propuesta. ....	45
Tabla 14.	Componentes electrónicos usados en el sistema .....	45
Tabla 15.	Consumo de voltaje y amperaje de componentes.....	46
Tabla 16.	Pulsos generados .....	51
Tabla 17.	Pruebas de resistencia estructural .....	61
Tabla 18.	Pruebas de carga de batería. ....	62
Tabla 19.	Pruebas de conmutación de fuentes.....	62
Tabla 20.	Resultados de pruebas de precisión .....	63
Tabla 21.	Resultados de pruebas a largo plazo .....	64
Tabla 22.	Error de medición del medidor digital.....	65

## ÍNDICE DE FIGURAS

Fig 1.	Mecanismo de medidor de disco nutativo .....	13
Fig 2.	Medidor de turbina .....	14
Fig 3.	Medidor de flujo electromagnético. ....	14
Fig 4.	Especificaciones del medidor Kamstrup. ....	15
Fig 5.	Medidor Gaoxiang Smart .....	16

Fig 6.	Medidor Gaoxiang Smart .....	16
Fig 7.	Medidor inteligente Lain Holding. ....	17
Fig 8.	Medidor inteligente Ain Holding.....	17
Fig 9.	Sensor YF-S201.....	19
Fig 10.	Sensor FS300A.....	19
Fig 11.	Sensor FS400A.....	20
Fig 12.	Arquitectura de un sistema IOT.....	21
Fig 13.	Placa de Desarrollo Arduino Uno .....	22
Fig 14.	Placa de Arduino Pro Mini .....	23
Fig 15.	Microcontrolador Arduino Nano .....	23
Fig 16.	ESP32-WROOM. ....	24
Fig 17.	ESP328266. ....	25
Fig 18.	Disco duro magnético.....	27
Fig 19.	Comunicación serial y comunicación paralela. ....	27
Fig 20.	Comunicación serial asíncrona.....	28
Fig 21.	Comunicación serial síncrona.....	28
Fig 22.	Comunicación serial SPI. ....	29
Fig 23.	Comunicación serial I2C. ....	30
Fig 24.	Módulo SD. ....	30
Fig 25.	Módulo RTC.....	31
Fig 26.	Módulos del dispositivo. ....	36
Fig 27.	Diagrama de bloques del sistema. ....	42
Fig 28.	Modelo de la solución propuesta.....	44
Fig 29.	Conexión del modelo propuesto al medidor convencional de agua doméstico. 44	
Fig 30.	Vista explosionada de la solución propuesta. ....	44
Fig 31.	Esquema lineal de conexión para conmutación mixta.....	48

Fig 32.	Esquema del circuito electrónico para conmutación mixta. ....	48
Fig 33.	Esquema lineal de conexión para unidad de procesamiento. ....	49
Fig 34.	Circuito electrónico para unidad de procesamiento. ....	49
Fig 35.	Pistas generadas de circuito de procesamiento y conmutador.....	49
Fig 36.	Transferencia térmica por planchado.....	50
Fig 37.	revelado de circuitos en acido férrico.....	50
Fig 38.	Montaje y soldadura de componentes electrónicos. ....	50
Fig 39.	Entorno de programación Arduino IDE.....	53
Fig 40.	Diagrama de flujo de procesamiento de datos.....	55
Fig 41.	Diagrama de flujo de HMI.....	55
Fig 42.	Interfaz HMI.....	56
Fig 43.	Interfaz HMI en funcionamiento. ....	56
Fig 44.	Interfaz HMI en funcionamiento, consumo litros acumulados. ....	57
Fig 45.	Modelo CAD de tapa superior ....	57
Fig 46.	Modelo CAD de carcasa superior.....	58
Fig 47.	Modelo CAD de carcasa inferior.....	58
Fig 48.	Modelo CAD de tapa inferior.....	59
Fig 49.	Carcasa impresa.....	60
Fig 50.	Ensamblaje de los componentes del dispositivo. ....	60
Fig 51.	Instalación del dispositivo in situ. ....	60
Fig 52.	Pruebas de precisión. ....	63
Fig 53.	Comparación de medidor convencional y medidor digital.....	66

# **CAPÍTULO I. INTRODUCCIÓN.**

## **1.1 Planteamiento del Problema.**

Los sistemas tradicionales de recolección de datos de consumo de agua, ampliamente utilizados en diversas zonas, presentan algunos desafíos y limitaciones. Uno de los problemas más significativos radica en el hecho de que el personal encargado debe realizar esta tarea manualmente.

Este enfoque manual comúnmente suele ir acompañado de errores humanos involuntarios que pueden producir fallas en la toma de lecturas, desencadenando en el desperdicio de recursos tecnológicos, económicos, de personal y de tiempo [1]. Además, en algunos casos la poca accesibilidad del medidor, suciedad, humedad o desgaste que sufren los visores del medidor dificultan la tarea del personal encargado de la toma de lectura. Por otro lado, la falta de recursos tecnológicos adecuados impide la implementación de sistemas de lectura avanzados, dando así una poca escalabilidad de este sistema tradicional de recolección de datos. En este sentido la implementación de un sistema de lectura digital de agua potable que permita una lectura fuera del error humano es una solución práctica para este problema.

## **1.2 Objetivos.**

### **1.2.1 General.**

Desarrollar un sistema digital para lectura de consumo de agua.

### **1.2.2 Específicos.**

- Contextualizar las características y el funcionamiento de los sensores de flujo disponibles.
- Diseñar el sistema de medición digital.
- Implementar el sistema de medición para su validación bajo condiciones reales de funcionamiento.

## **1.3 Alcance y delimitación**

Este trabajo se centra en el desarrollo de un medidor digital, diseñado para captar el consumo de agua potable y transmitir la información de manera serial. El sistema está concebido con la prospectiva de permitir el envío remoto de las lecturas a través de una red de largo alcance en posteriores implementaciones.

Con base en las características y el funcionamiento de los dispositivos disponibles en el mercado, se determinarán los componentes electrónicos y los dispositivos para procesamiento y envío de datos. En cuanto a su funcionalidad, el sistema en general podrá configurarse in situ, guardar las lecturas en memoria no volátil y enviar la información por comunicación serial.

La implementación se realizará en un domicilio con acceso al suministro público de agua potable y que cuente con un medidor convencional instalado. Las pruebas de validación compararán los datos captados por el sistema digital con los obtenidos del medidor convencional.

#### **1.4 Justificación**

La gestión eficiente del consumo de agua es un desafío significativo que afecta tanto la sostenibilidad de los recursos hídricos como la calidad de vida de los habitantes. La recolección manual de datos de consumo de agua presenta errores en la correcta interpretación de los datos arrojados por el medidor esto debido a los factores asociados al error humano. Por este motivo la integración de sistemas digitales que ayuden a la recolección e interpretación de los datos puede colaborar en el desarrollo de una mejor flexibilidad del sistema de gestión. Tal como se afirma en [2], los sistemas inteligentes de distribución de agua potable deben ser cada vez más eficientes y responder a múltiples restricciones.

Además, la transformación de los sistemas gestores de información a sistemas netamente tecnológicos es un campo en auge en la actualidad, tal como se discute en [3], el sector del agua debe centrarse en la gestión del cambio, ayudando a las personas y a las empresas a redefinir sus sistemas de procesos para afrontar la llegada de la industria 4.0.

En este sentido, el desarrollo de un sistema digital para lectura del consumo de agua basado en microcontroladores, sensores y tecnología de comunicación serial representa una solución práctica y relevante dentro de las estrategias de modernización y optimización actuales. La implementación de este sistema ofrece una oportunidad para explorar nuevas formas de gestionar los recursos hídricos de manera más eficiente y efectiva.

## **CAPITULO 2: MARCO TEÓRICO.**

### **2.1 Antecedentes.**

En la Universidad Técnica del Norte se llevaron a cabo dos proyectos enfocados a la gestión del agua potable utilizando recursos tecnológicos e informáticos, los mismo comparten el objetivo general de realizar la medición del consumo de agua potable de manera digital.

Se desarrolló un medidor de agua inteligente con el objetivo de gestionar y transmitir datos sobre el consumo de agua doméstico. Este proyecto logró migrar de un sistema rudimentario de gestión de datos a un sistema inteligente mediante la implementación de sensores de caudal, un microcontrolador ESP E12 y electroválvulas. Uno de los aspectos clave fue la transmisión de los datos de lectura, para lo cual se utilizaron módulos Wifi-conectados a la red domiciliaria del cliente, estos módulos se encargaron de enviar la información a la junta administradora de agua potable. Además, se empleó una base de datos alojada en un hosting gratuito para el almacenamiento de los datos, garantizando un acceso eficiente y continuo a la información recopilada [4].

Se realizó la construcción de un prototipo electrónico de medición de caudal para el registro de lecturas de medidores de agua potable, este prototipo permite obtener la información de consumo de agua potable de cada usuario y almacenar la información en un historial de consumo en un sistema de base datos de carácter local. Para la construcción de este prototipo se utilizaron placas de desarrollo de código abierto, sensores de caudal de efecto hall y cargadores solares [5].

Además, en la facultad de ingeniería y ciencias agropecuarias de la universidad UDLA, se desarrolló un prototipo basado en un sistema de recolección de información para medidores de agua potable, este tiene la funcionalidad de dar a conocer la medición del consumo de agua potable utilizando microcontroladores, sensores de caudal y fuentes de alimentación. Para el tratamiento y visualización de los datos se utilizaron lenguajes de programación HTML y PHP con una base de datos en MySQL [6].

En la Universidad de las Fuerzas Armadas ESPE se realizó un prototipo de un medidor digital para la adquisición de los datos del consumo de agua potable domiciliaria, el medidor consta de una gráfica de cristal líquido (GLCD) y se realiza el envío de las

lecturas de manera inalámbrica a través del protocolo GSM. Para el procesamiento del flujo de programación se utilizó un PIC18F452 y para la visualización de los datos en la estación principal se realizó un HMI utilizando la aplicación de LabView [7].

En la Universidad Superior Politécnica de Chimborazo se desarrolló un prototipo de un nodo de medidor inteligente compuesto por un ESP8266 y dos Arduino Nano para la parte de procesamiento de los datos del consumo de agua, además para las mediciones se utilizó un medidor de la marca Zenner MTKD-S, el cual tiene incorporado un dispositivo electrónico en su relojería mecánica, dicho dispositivo genera un impulso cada 10 litros, este impulso fue útil para ser captado por un sensor de efecto hall anexo en la estructura del nodo medidor. Para la fase de envío de datos se usó una infraestructura de red Wi-Fi, lo que permitió conectarse a la red de internet. Finalmente, los datos captados se almacenaron en una base de datos alojada en MySQL, permitiendo así ser visibles en una aplicación Web [8].

## **2.2 Bases Teóricas.**

### **2.2.1 Agua potable.**

El agua potable es aquel líquido vital dulce que tras una serie de tratamientos químicos para separar sedimentos o componentes residuales se habilita para distribución a los sectores de consumo doméstico. Dentro de los lineamientos químicos según [9] para que el agua sea considerada potable la medida de acidez o alcalinidad (pH) debe rondar entre los 6,5 y máximo 9,5. Además su importancia para la existencia humana la destaca en [10] donde define al agua en sí como un recurso indispensable para los seres vivos, encontrándose presente en toda forma de vida y en muchas actividades productivas como la agricultura, la ganadería y otros procesos derivados.

#### **2.2.1.1 Sistema de gestión del agua potable en Ecuador.**

Según el artículo 318 de la constitución la gestión del recurso hídrico para destino consumo potable y de riego es exclusivamente pública o comunitaria, además se prescribe que, el Estado a través de la Autoridad Única del Agua será el responsable directo de la planificación y gestión de los recursos hídricos que se destinaran a consumo humano y de riego que garantice la soberanía alimentaria, caudal ecológico y actividades productivas.

Bajo la misma Autoridad Única de Agua se realiza la coordinación de la gestión con los diferentes niveles de gobierno [11].

Los niveles de gobierno autónomos bajo la ordenanza superior del Estado y bajo convenio comunal de la zona, se encargan de gestionar el suministro del líquido vital a los diferentes moradores del poblado mediante redes de tuberías conectadas próximos a sus domicilios. Para la gestión de la recaudación monetaria en función al consumo domiciliario del agua, las juntas administradoras de agua potable (JAAPs) integran un medidor de agua convencional con unidades de medida en metros cúbicos generalmente.

### **2.2.2 Tipos de medidor de agua potable.**

Los sistemas de medición de agua potable son aquellos que se utilizan para la gestión eficiente de la cantidad de consumo del agua potable residencial. Estos sistemas ayudan a la correcta facturación, planificación y control del suministro del agua potable en cada hogar residencial. Independiente del tipo de medidor que se emplee este realiza el conteo el volumen de agua que se recibe de un sistema público y que pasa a través de él.

#### **2.2.2.1 Medidores de disco oscilante o nutativo.**

Tan pronto como el agua entra en el espacio entre el disco y las paredes del medidor este empuja el disco imprimiéndose un movimiento de oscilación rotativa similar al comportamiento de un trompo al final de su fase de giro



Fig 1. Mecanismo de medidor de disco nutativo [12].

#### **2.2.2.2 Medidores de turbina o velocidad.**

Se realiza la medición en función de la velocidad del flujo de agua que pasa a través de unos elementos móviles contenidos dentro de la cámara del medidor. Estos emplean partes mecánicas y que por acción de la velocidad el agua hace girar el mecanismo de traducción de la medida, en este caso suele ser una turbina o una hélice[12].

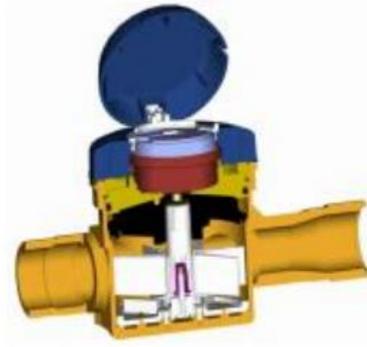


Fig 2. Medidor de turbina [12].

### 2.2.2.3 Medidor electromagnético.

La electrónica contenida en este medidor controla un campo magnético en el tubo del sensor, el cual se ve alterado por la velocidad del flujo de agua que pasa por el tubo, este cuantifica su intensidad que es proporcional al flujo y por consiguiente calcula el volumen y el caudal [13].



Fig 3. Medidor de flujo electromagnético [13].

### 2.2.3 Medidores inteligentes de agua potable.

Debido al gran auge de las tecnologías en el campo de la instrumentación, los aparatos encargados de registrar y traducir una cantidad física de medida ahora cuentan con el apoyo de las nuevas tecnologías, en este caso los medidores de agua potable ahora cuentan con mejoras en el sistema de captación del consumo de agua potable. Tales mejoras son:

Precisión en la lectura del flujo de agua.

Capacidad de gestionar el flujo de agua (habilitación o corte de suministro).

Envío y transmisión ágil de datos.

Escalabilidad.

Disminución de costes operativos.

### 2.2.3.1 Medidor Kamstrup.

Es un tipo de medidor de agua electrónico con una interfaz que incorpora un puerto COM, conexión USB, adaptador para ojo óptico y tecnología de comunicación Bluetooth. Este medidor cuenta con una interfaz de control que permite la interacción con las diferentes funcionalidades del medidor.

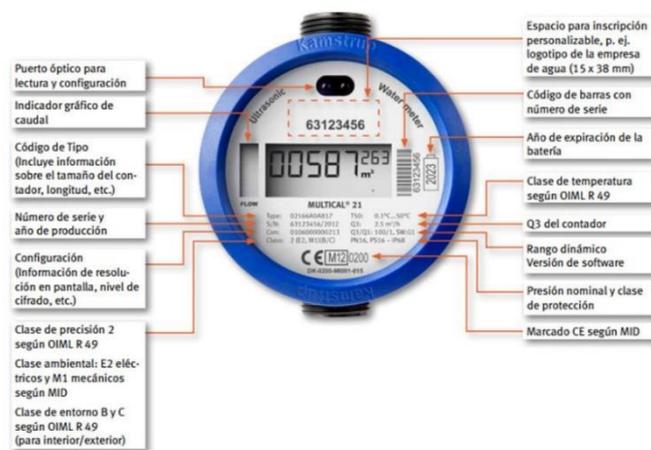


Fig 4. Especificaciones del medidor Kamstrup [14].

### 2.2.3.2 Medidor Gaoxiang Smart.

Este tipo de medidor está compuesto por un módulo de microchip, válvulas de control automático de apertura y cierre de suministro, pantalla LCD que permite la visualización del consumo de agua y un lector de tarjetas IC y RF que permiten el almacenamiento y gestión de contabilidad del servicio.



Fig 5. Medidor Gaoxiang Smart [15]

### 2.2.3.3 Medidor ultrasónico IOT

Este tipo de medidor ultrasónico funciona gracias a la detección de la diferencia de velocidad cuando el haz ultrasónico se propaga a lo largo y contra la corriente de agua para calcular el flujo, este medidor no cuenta con partes mecánicas móviles en el interior de la carcasa, no hay elementos de bloqueo en el paso del agua por lo que las impurezas no afectan a la vida útil. Cuenta con tecnología de comunicación remota inalámbrica LorA/LoRaWAN/NB-IOT/GPRS y tecnología remota con cable RS-485 y MBus.

Además, el modo de alimentación puede cambiar entre alimentación por batería interna y alimentación externa.



Fig 6. Medidor Gaoxiang Smart [16].

### 2.2.3.4 Medidor Lain Holding

Este es un tipo de medidor de agua de chorro único que tiene la característica de estar equipado con tecnología inductiva, dicha tecnología permite la instalación de dispositivos emisores de pulso como módulos emisores de radio frecuencia brindando así una lectura remota a través de SIGFOX, LORAWAN O Wireless MBUS. Posee una curva de medición capaz de capturar niveles de caudal altos y bajos. Posee un funcionamiento por batería, la durabilidad de esta es en función de la frecuencia de transmisión de datos alcanzado así una vida útil de entre 10 a 15 años. Para la gestión de los datos de consumo este medidor cuenta con la capacidad de enlazarse con una plataforma informática robusta [17].



Fig 7. Medidor inteligente Lain Holding.

#### 2.2.3.5 Medidor inteligente SHMETERS

Este medidor tiene mayor precisión de medida debido a que soporta una medición de tonelaje para unidades pequeñas debido a su conteo por goteo que es altamente sensible. Posee un sistema de lectura remota a través de protocolo LORA, el mismo que puede controlar el funcionamiento del medidor, no necesita de cableado externo por lo que lo hace fácil de instalar, La resistencia de este medidor es alta debido a que su fabricación es a base de una carcasa de cobre, resistiendo así a las condiciones de oxidación [18].



Fig 8. Medidor inteligente Ain Holding[18].

#### 2.2.4 Caudal del agua.

El caudal es la medida de una determinada cantidad de agua que pasa por un espacio (tubería, canal, maguera, etc) en un cierto espacio de tiempo, es decir, esta medida corresponde a un volumen de agua con unidades ya sea en litros, metros cúbicos, etc, multiplicado por una unidad de tiempo ya sean, segundos, minutos, horas [19].

Determinando la relación entre la velocidad y un área de una sección de espacio, se puede plantear la relación matemática con la siguiente ecuación de caudal del agua.

$$Q = V * A \quad (1)$$

Donde:

Q: caudal de agua.

V: velocidad del fluido.

A: área.

#### 2.2.4.1 Sensores de caudal.

Los sensores de caudal, también conocidos como detectores de flujo son aquellos dispositivos de instrumentación aplicados para la medición de magnitudes físicas, en este caso estos sensores son utilizado en las líneas de tuberías de una red de suministro de líquidos o gases con la finalidad de conocer la cantidad de fluido que pasa de un punto a otro en un determinado tiempo. Este conocimiento del flujo permite a las industrias la automatización, especialmente en sectores como el de la fabricación de las bebidas, donde es esencial controlar y asegurar la cantidad de líquido que se distribuye en cada producto.

- **Sensor YF-S201.**

Este es un tipo de sensor de flujo que funciona a partir del efecto hall. Construido dentro de una carcasa de plástico resistente y adaptable para tuberías de ½". Su funcionamiento se basa en un rotor de paletas, este rotor contiene elementos magnéticos de efecto hall, estos elementos permite captar una señal electrónica con cada movimiento del rotor de paletas, estas señales pueden ser interpretadas como pulsos los cuales pueden ser relacionados con la cantidad de líquido corriente en un determinado tiempo [20].



Fig 9. Sensor YF-S201 [20].

- **Sensor FS300A.**

El sensor de flujo FS300A es apto para medidas de caudales de agua en tuberías de 3/4 de pulgada. Es un tipo de caudalímetro de rotor de turbina compatible con sistemas digitales. El funcionamiento consta de un rotor de turbina de paletas que contiene imanes y que a su giro en presencia de movimiento de líquido lo hacen rotar, estas variaciones son captadas por un sensor de efecto hall que traduce cada variación como un pulso digital que puede reconocer un sistema embebido como placas de Arduino.



Fig 10. Sensor FS300A[20].

- **Sensor FS400A.**

Este caudalímetro electrónico es apto para colocarse en tuberías de 1 pulgada, es capaz de medir caudales de líquidos con baja viscosidad, internamente este sensor posee un mecanismo giratorio de hélice el cual es accionado por el paso de una corriente de líquido, esta hélice por medio de un conjunto de componentes basado en el efecto hall generan señales o pulsos electromagnéticos que pueden ser relacionados con la cantidad de líquido que pasa por una tubería. Su rango de alimentación es de 5 a 24 volts, tiene un factor de conversión de 4.8.



Fig 11. Sensor FS400A.

### **2.2.5 IOT**

El internet de las cosas es un concepto que se acuña por dos palabras “internet” y “cosas”. Internet es un sistema que conforma una red global informática, interconectados entre si por protocolos (TCP/IP) para posibilitar la comunicación entre millones de usuarios con redes empresariales, educativas, personales, públicas que están conectados a través de diferentes dispositivos electrónicos y redes inalámbricas.

Al llegar a definir “cosas”, puede tomar varios significados, este puede ser un tipo de objeto físico o persona que se distinga del mundo real. En es contexto de la tecnología el internet se combina con estas “cosas” para lograr una optimización y control de estas, esta unión se amplía a este compendio de cosas vivas y no vivas.

Una definición aproximada no generalista define al internet de las cosas puede entenderse como datos que se generan a través de las cosas, a diferencia del internet convencional que trata con datos creados por las personas, por lo que se podría concluir que el internet de las cosas (IOT) es un red abierta y completa de objetos inteligentes que pueden organizarse, compartir datos y recursos actuando bajo condiciones de un entorno definido [21].

#### **2.2.5.1 Arquitectura del IOT.**

Un requisito mínimo dentro de la infraestructura del IOT es que los objetos deben estar interconectados entre sí a la red de trabajo. La arquitectura de un sistema IOT garantiza esta interacción entre ambos elementos, esto permite que se cierra la brecha entre el mundo físico y el mundo digital. El diseño de una arquitectura basada en IOT implica algunos factores como las redes, la comunicación, los modelos y la seguridad por mencionar algunos. Además, el IOT debe poder interactuar con estos objetos en un entorno dinámico, lo que implica que la comunicación debe mantenerse en todo momento independiente del desplazamiento[22].

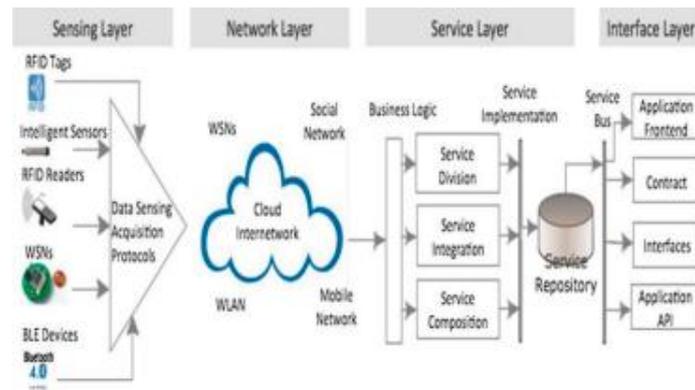


Fig 12. Arquitectura de un sistema IOT[22].

### 2.2.6 Placas de desarrollo o microcontrolador.

Un microcontrolador es un dispositivo de carácter computacional que es operado por un chip fabricado especialmente para aplicaciones de control de computadoras integradas. Estos dispositivos con características por su bajo costo y su sencillez de uso en mayoría de aplicaciones.

La mayoría de los microcontroladores tienen un circuito integrado que se encarga del manejo y procesamiento de por ejemplo convertidores A/D, que facilitan el muestreo de señales externas, también contienen puertos de entrada y salidas de datos digitales, algunos microcontroladores tienen convertidores D/A, además en su arquitectura de construcción contiene un circuito de temporización que permite tareas sincronizadas lo que permite la realización de programas de control mejor optimizados.

La programación para los microcontroladores originalmente es el lenguaje ensamblador, pero debido a la demanda de un entorno de programación de alto nivel los fabricantes han desarrollado microcontroladores capaces de ser programados mediante lenguaje de alto nivel como PASCAL, C o BASIC. Estos lenguajes ofrecen ciertas ventajas[23].

- Facilidad de desarrollo de programas en alto nivel.
- Mejor mantenimiento del programa.
- Las pruebas de un programa de alto nivel son más fáciles.
- Menos posibilidades de cometer errores durante la programación.
- El más fácil documentar un programa hecho en alto nivel.

### 2.2.6.1 Arduino UNO.

Es una placa de desarrollo que tanto su hardware y su software son de acceso libre y tiene muchas posibilidades de adaptación a los diversos entornos de aplicación. Arduino es un entorno que combina la electrónica y la programación basada en un lenguaje de alto nivel [24]. Esta placa de desarrollo utiliza el microcontrolador Atmega 328 IC, encargado de procesar la lógica de las instrucciones programadas, además a esto le acompañan ciertas especificaciones[25]:

- Voltaje de funcionamiento de 5v.
- Voltaje de entrada de 7-12v.
- 14 pines de entrada y salida digitales.
- 6 pines analógicos.
- Corriente continua de pin de entrada y salida de 50mA.
- Compilación de corriente continua 3.3v 50mA.
- Memoria flash 30KB.
- SRAM 2 KB.
- EEPROM 1KB.
- Velocidad de reloj de 16 MHz.



Fig 13. Placa de Desarrollo Arduino Uno[26].

### 2.2.6.2 Arduino pro mini.

El Arduino pro mini es una placa de microcontrolador basada en el chip de ATmega 328. Tiene en total 14 pines de entrada y salida de los cuales 6 pines se pueden usar como salidas para PWM, 6 entrada analógicas, un resonador, un botón de reinicio y cuenta con orificios para montar los pines de conexión. Es de destacar que existen dos versiones de Arduino Pro Mini, uno que funciona a 3.3v a 8Mhz y el otro modelo que funciona a 5v a 16Mhz, el puerto de entrada para la programación es de tipo micro USB[27].

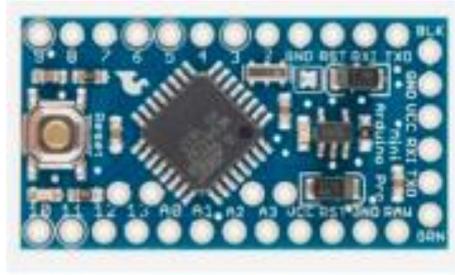


Fig 14. Placa de Arduino Pro Mini[27].

### 2.2.6.3 Arduino NANO.

LA PLACA Arduino Uno es una placa que se caracteriza por su tamaño reducido ideal para proyectos de un procesamiento medio y donde se busca priorizar el consumo de energía. El hardware tiene integrado pines de conexión lo que la hace ideal para conexión en placa de pruebas, esta placa está basada en el microcontrolador ATmega328 en la versión de Arduino Nano 3.x, por otro lado, está la placa basada en el microcontrolador ATmega168 en la versión de Arduino Nano 2.x. La versión basada en el ATmega 168 tiene 16KB de memoria flash, mientras que la placa basada en el ATmega 328 tiene 32KB [28].

Otras de las características que destacan son:

- Todas las versiones del Arduino Nano trabajan a 16 MHz.
- Tiene una tensión de alimentación de 5v, pero puede permitirse variar de 4 a 12v.
- Posee 14 pines digitales, 8 analógicos, 2 pines de reinicio y 6 pines de potencia (Vcc y GND).
- Su consumo de energía es de 19mA.
- Su tamaño es de 18x45 mm con un peso de 7 gramos[29].

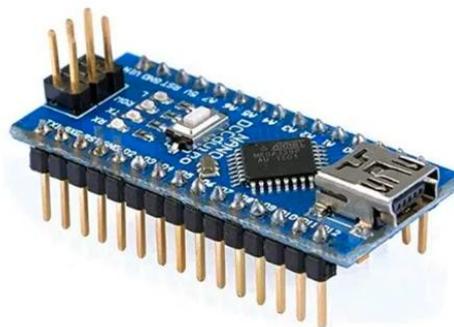


Fig 15. Microcontrolador Arduino Nano[29].

#### 2.2.6.4 ESP32 - WROOM.

El ESP32 desarrollado por ESPRESSIF actualmente es conocido por ser un tipo de microcontrolador capaz de realizar procesamiento de datos a una velocidad mayor, además debido a sus características de conectividad es muy utilizado en aplicaciones de monitoreo de sensores y sistemas que trabajan con IOT[30]. dentro de las características del hardware tenemos:

- Un procesador de 32 bits a 260 MHz máximo.
- 4 comunicaciones SPI, 2 de I2C, 2 de I2S, 3 de UART y Bus CAN2.0.
- 36 pines GPIO, de los cuales 16 pueden usarse como PWM.
- 18 pines que pueden usarse como entradas analógicas.
- Posee 2 conversores análogo digital de entrada múltiple.
- Convertidores de 12 bits.
- Posee una configuración de *Deep Sleep* que reduce su consumo a 2.5uA [31].



Fig 16. ESP32-WROOM.

#### 2.2.6.5 ESP 328266.

Esta placa de desarrollo posee grandes capacidades mejoradas a la del EsP32 WROOM, destaca su gran versatilidad en cuanto al consumo de energía debido a que posee 3 tipos de configuración energética en modo activo, modo de suspensión y modo de suspensión profunda, esta característica lo hace ideal para proyectos móviles alimentados por batería. Su alta durabilidad lo hace apto para trabajar en entornos industriales debido a que soporta un amplio rango de temperaturas de funcionamiento. El procesador que integra es un Tensilica de 32 bits que alcanza una velocidad de reloj máxima de 160 Mhz. El sistema operativo en tiempo real permite que la potencia de procesamiento esté disponible en hasta en un 80 % para las tareas de programación y desarrollo de aplicaciones del usuario.



Fig 17.ESP328266.

### **2.2.7 Memoria no volátil.**

La memoria no volátil (NVM) es un tipo de microchip que tiene la capacidad de realizar el almacenamiento de datos, escribirlos y guardarlos. Una de las características de esta memoria es que los datos escritos se mantienen incluso sin alimentación energética. Este tipo de memorias ampliamente utilizados en muchos dispositivos se pueden clasificar dependiendo de su direccionamiento eléctrico o mecánico.

#### **2.2.7.1 Direccionamiento eléctrico.**

- **Memoria PROM**

Las memorias PROM son un tipo de memorias programables que solo son de lectura. La arquitectura de la memoria PROM consiste en un conjunto de compuertas AND que alimentan una matriz programable OR. La memoria PROM son usadas para la decodificación de señales de entrada para obtener distintas combinaciones de salida. El proceso de grabación de datos se controla aplicando tensión a los transistores internos[32].

- **Memoria EPROM.**

La memoria EPROM es un tipo de memoria que aparte de permitir la lectura permite el borrado de datos, estas memorias cuentan con un interruptor que al activarse permite el borrado, con esto la información almacenada en los transistores del circuito integrado pueden ser borrados, además una vez borrado los datos se puede volver a escribir otros datos diferentes[33].

- **Memoria EEPROM y la memoria Flash.**

Otra clase de memorias borrables son las EEPROM (Electrically Erasable PROM), la EEPROM puede ser borrada con la aplicación de un pulso de voltaje. Algunas

versiones de esta memoria permiten el borrado del arreglo entero, mientras que otras versiones solo pueden borrar individualmente ciertos bytes. Las versiones más comunes de estas memorias son las de 32x16, 256x4, 1024x4 y 2048x8. Los tiempos para lectura son de 250ns a 10ns por palabra y los tiempos de escritura son de 50ms por palabra y el tiempo de borrado es de 10ms a 100 ms por palabra[33].

- **Memoria ferroeléctrica o F-RAM**

Este tipo de memoria no volátil se caracteriza por su velocidad de escritura, muy superior a diferencias de las EEPROM y las Flash. La construcción de esta memoria es a base de material ferroeléctrico polarizable en lugar de material dieléctrico y no tiene limitaciones en los ciclos de escritura y borrado. El método de escritura es de palabra a palabra y no en bloques[34][35].

- **Memoria RAM magnetorresistiva.**

Es un tipo de memoria no volátil similar la memoria RAM ferroeléctrica. Esta memoria en particular usa el magnetismo para almacenar datos su espacio. Actualmente se usa en pocos dispositivos.

#### **2.2.7.2 Direccionamiento mecánico.**

- **Disco duro magnético.**

Este tipo de memoria consta de varias partes magnéticas, el principal es una superficie magnética giratoria denominada plato y un brazo mecánico que se desplaza sobre este disco. El brazo mecánico se usa para leer y escribir datos en el disco magnético. Cabe destacar que la escritura y borrado de los datos se realiza de manera magnética. Los datos se escriben cuando se aplica una corriente al brazo, esto magnetiza puntos existentes en el disco que gira a gran velocidad. Los datos en este tipo de memoria se organizan en forma de pistas y sectores, aquellas pistas son divisiones circulares. Las pistas a su vez se dividen en bloques.



Fig 18. Disco duro magnético.

### 2.2.8 Protocolos de Comunicación serial.

Este tipo de comunicación transmite una trama de datos uno tras otro, en este caso los datos son exclusivamente digitales ya que debido a las características de conexión a la que le limita este tipo de comunicación los datos viajan por un solo medio de conexión[36]. Este beneficio de simplicidad de conexión hace que este tipo de comunicación sea lenta.

Por otro lado, hay que destacar la existencia una comunicación mucho más rápida, esta es la comunicación en paralelo, esta permite enviar 1 byte con un solo pulso de reloj, es decir cuando suceda el pulso de reloj se enviaran los 8 bits que contiene 1 byte al mismo tiempo, esto claramente se traduce en una rapidez de 8 veces más que la comunicación serial. La comunicación paralela tiene la desventaja de emplear muchas conexiones físicas algo que hoy en día no es común debido a las reducciones de hardware de varios microcontroladores.

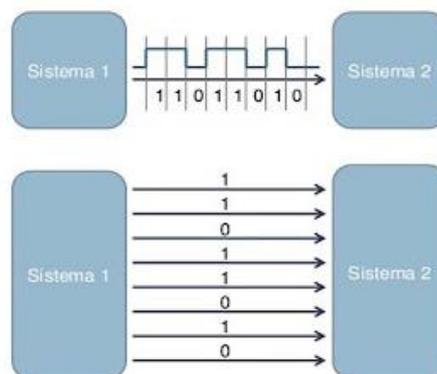


Fig 19. Comunicación serial y comunicación paralela[37].

#### 2.2.8.1 Modo asíncrono.

El modo de comunicación asíncrona no utiliza un reloj para saber el inicio y el final del byte enviado del Tx al Rx, del mismo modo el receptor (Rx) no sabe en qué momento llegara el inicio del byte, para resolver esto se utiliza un bit de inicio que va en el encabezado del byte y un bit de parada que define el final del byte entero. Este formato del byte se hace con la finalidad de sincronizar los relojes del Tx y el Rx[36].

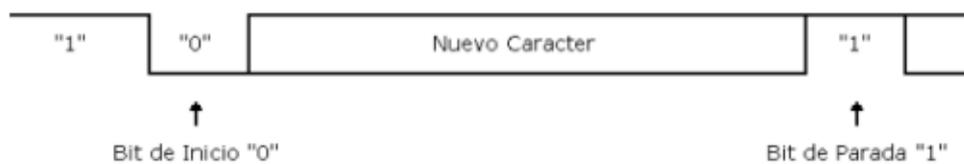


Fig 20. Comunicación serial asíncrona [36].

### 2.2.8.2 Modo síncrono.

Este modo de comunicación serial usa una señal de reloj para la transmisión de una trama de datos, el formato de organización del byte consta de un conjunto de bits de sincronismo y un conjunto de bits de final. El bit de sincronismo tiene la funcionalidad de sincronizar los relojes del transmisor (Tx) y del receptor (Rx) de tal manera que estos controlan la duración de envío de cada bit y byte.

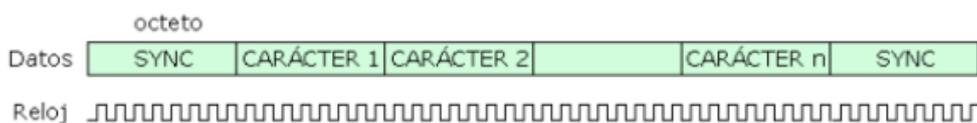


Fig 21. Comunicación serial síncrona.

### 2.2.8.3 Tipos de comunicación serial.

- **UART**

La comunicación Receptor y Trasmisor Asíncrono Universal (UART) por sus siglas en inglés es una especie de comunicación serial de tipo asíncrona. Este protocolo de comunicación seria al no tener presencia de una señal de reloj debe realizar la transmisión de datos con la configuración de un bit de inicio y un bit de parada, esto para ayudar a sincronizar los tiempos del emisor y del receptor. Para ayudar a mantener la fijación de sincronicidad de emisor y receptor se usa el concepto de velocidad en baudios que ayuda a sincronizar la velocidad de intercambio de información. La conexión para esta comunicación consta de 1 cable para el Tx y 1 cable para el Rx, además de una conexión a tierra común (GND).

- **SPI**

La comunicación SPI es un tipo de comunicación en serie de tipo síncrono, este funciona con dos líneas de comunicación denominadas MISO y MOSI, un reloj de sincronización denominado SCK y una línea para seleccionar el esclavo llamado (SS o CS)[36].

En este tipo de protocolo el maestro es el que controla la llamada de los esclavos y ningún esclavo puede enviar datos al maestro. El maestro envía los datos a través del MOSI (Master Output Slave Input), mientras que el esclavo puede responder a través del MISO (Master Input Slave Output). En todo momento el reloj (SCK) está sincronizando el envío de datos de maestro a esclavo. La función del selector de chip (SS o CS) es la de seleccionar el esclavo al cual se desea comunicar el maestro. Este protocolo permite conectar hasta dos dispositivos esclavo [38].

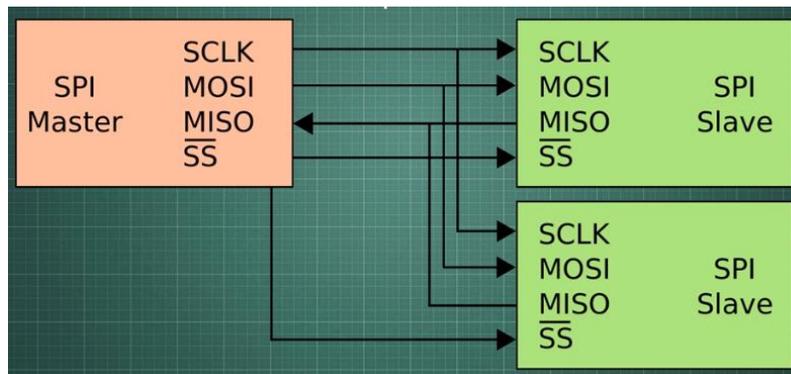


Fig 22. Comunicación serial SPI [38].

- **I2C**

Este es un tipo de comunicación síncrono, este protocolo es característico por tener la capacidad de conectarse con más de dos esclavos usando únicamente dos cables conocidos como el SDA (datos en serie) y el SCL (reloj en serie), a diferencia de otros protocolos este solo admite un solo maestro. Debido a que existe varios esclavos, es necesario crear bits de identificación para cada esclavo, de 7 a 10 bits específicamente. Cuando el maestro quiere enviar datos a un determinado esclavo, este debe crear una solicitud con los bits de identificación de ese esclavo en específico, luego cada esclavo hace coincidir su dirección y si corresponde a su identificador el esclavo responderá al maestro. Además de los bits de inicio y de parada, se emplea un bit de reconocimiento

antes mencionado denominado (ACK) y uno de reconocimiento negativo denominado (NACK)[38].

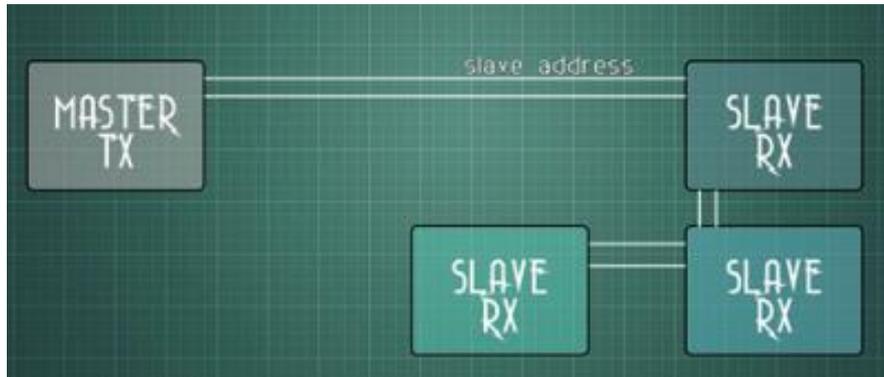


Fig 23. Comunicación serial I2C [38].

### 2.2.9. Módulo SD.

El módulo SD es un dispositivo que permite acoplar, leer y escribir una memoria microSD, el método que se emplea para realizar la comunicación con este módulo es a través del protocolo serial SPI. Debido a que maneja esta comunicación serial el módulo implementa pines ya definidos que facilitan la conexión del módulo, dichos pines están denominados como MOSI, MISO, SS o CS, SCK, VCC y GND. La tensión de alimentación que soporta es de máximo 5v. [39]

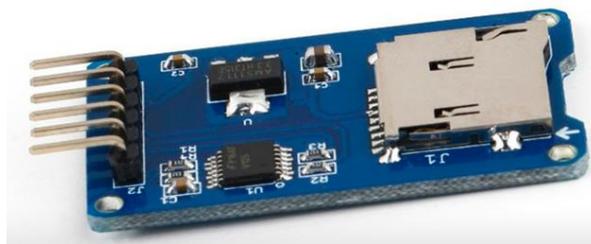


Fig 24. Módulo SD.

### 2.9.2 Módulo de reloj RTC.

El módulo RTC (Real Time Clock), conocido como reloj de tiempo real es un dispositivo que sirve para obtener el tiempo actual, comprendido en segundo, minuto, día, mes y año, esto dependerá de la configuración inicial. La medida del tiempo lo hace a partir del conteo de los pulsos de la señal de reloj en combinación con la frecuencia. El módulo integra un cristal de cuarzo integrado el cual es el responsable de emitir la frecuencia. Internamente implementa circuitería que permite el conteo y almacenamiento de tiempo en una memoria. El módulo integra una batería de 3v que sirve para la

alimentación de la circuitería de memoria, lo que impide la pérdida del tiempo una vez grabada. El rango de funcionamiento va desde 3.3v a 5v.

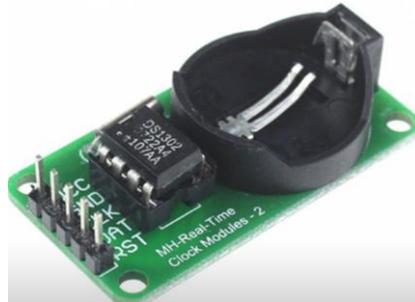


Fig 25. Módulo RTC.

## **CAPÍTULO 3: MARCO METODOLÓGICO.**

### **3.1 Enfoque y tipo de investigación**

Este trabajo de integración curricular se enmarca en una investigación aplicada, ya que en el desarrollo de este sistema digital para lectura de agua potable se aplican de manera integrada los conocimientos adquiridos a lo largo de la carrera de ingeniería mecatrónica, como la electrónica de sistema embebidos, la lógica de programación y el diseño. Estos elementos se conjuntan para brindar una solución práctica y tecnológica que permita mejorar la medición y gestión del consumo de agua potable.

Por otro lado, debido a la necesaria búsqueda de información pertinente a los elementos y datos a utilizarse para dar solución a la problemática que abarca la lectura de agua potable se aplica una investigación documental ya que es de vital importancia contar con recursos como libros, informes, revistas y artículos confiables que brinden información verídica en la cual basar este trabajo[40]. De este mismo modo se plantea una investigación descriptiva debido a que se pretende dar a conocer el objetivo de función contenido en los sensores, módulos, microcontroladores y demás elementos que conforman el sistema digital para lectura de agua potable, dando así un entendimiento general del propósito del dispositivo [41].

### **3.2 Diseño de la investigación.**

Para el desarrollo de este trabajo se plantean diferentes actividades que vayan dentro del marco de los tipos de investigación antes mencionados y que aporten al desarrollo de un sistema digital para lectura de agua potable.

#### **3.2.1 Fase 1: Contextualización de las características y el funcionamiento de los sensores de flujo disponibles.**

En esta fase se realiza una investigación y conocimiento previo sobre los distintos sensores de flujo existentes en el mercado, así como la compatibilidad con los microcontroladores.

*Actividad 1.1: Investigación de las características y funcionalidad de diferentes tipos de sensores de flujo disponibles en el mercado;* se plantea buscar información bibliográfica y técnica referente a las características de cada sensor disponible en el mercado.

***Actividad 1.2: Revisión de las características de conexión y compatibilidad de los sensores;*** esta actividad tiene como objetivo conocer las características de funcionamiento de los sensores de caudal.

***Actividad 1.3: Selección del sensor de flujo;*** esta actividad plantea la selección de un sensor de flujo específico de acuerdo a las características de compatibilidad, disponibilidad, funcionalidad y requerimientos del proyecto.

### **3.2.2 Fase 2: Revisión de las características de conexión y compatibilidad de los sensores con microcontroladores, memorias no volátiles.**

Esta fase de objetivo es realizar una investigación acerca de los posibles microcontroladores compatibles con sensores digitales, además de una investigación literaria y técnica de las unidades de almacenamiento no volátil.

***Actividad 2.1: Revisión literaria de microcontroladores disponibles en el mercado;*** con esta actividad se plantea hacer un sondeo de posibles microcontroladores que satisfagan las cuestiones de procesamiento y compatibilidad con sensores de caudal.

***Actividad 2.2: Selección de microcontrolador:*** Se busca la selección de un microcontrolador que satisfagan los criterios que demandan un sistema autónomo.

***Actividad 2.3: Investigación literaria y técnica de memorias no volátiles:*** La actividad busca realizar un sondeo de las diferentes opciones de almacenamiento de datos en memorias no volátiles, así como sus características de conexión, lectura y escritura de datos.

### **3.2.3 Fase 3: Diseñar el sistema de medición digital.**

Esta fase se enfoca en el diseño y montaje de todos los elementos seleccionados previamente, el diseño abarca el desarrollo del hardware necesario como placa PCB y piezas de ensamblaje, y dentro del desarrollo de software se abarca el diseño y programación del código de funcionamiento del sistema y de una interfaz de usuario.

***Actividad 3.1: Diseño del circuito;*** La actividad se centra en el desarrollo del circuito en base a los elementos seleccionados y sus condiciones de funcionamiento. El objetivo es obtener una PCB con conexiones hábiles para cada elemento del sistema.

***Actividad 3.2: montaje del hardware de sistema de medición;*** esta actividad se encarga de montar todos los elementos seleccionados en la PCB antes desarrollada.

***Actividad 3.3: desarrollo de la programación;*** Con esta actividad se busca realizar un diagrama de flujo de acuerdo con la función que debe realizar el medidor para finalmente realizar la programación, además se busca ubicar posibles puntos de falla en el comportamiento del dispositivo y proveerlos mediante programación.

***Actividad 3.4: Desarrollo y prueba del software necesario para la correcta operación del sistema de medición digital;*** en esta actividad se busca desarrollar un entorno de usuario para la interacción del humano con la maquina (HMI), el cual ofrezca acciones básicas para la operación del medidor.

#### **3.2.4 Fase 4: Implementar el sistema de medición para su validación bajo condiciones reales de funcionamiento.**

Esta fase última busca realizar las pruebas de funcionamiento del dispositivo del sistema digital en condiciones reales, tales condiciones abarcan la instalación a la red de suministro de agua potable de la vivienda y el funcionamiento continuado para coleccionar la lectura de agua en un periodo mensual.

***Actividad 4.1: Realización de pruebas de conexión y funcionamiento del sistema de medición en condiciones reales.:*** la finalidad de esta actividad es realizar la conexión del medidor desarrollado a la red de suministro de agua, así como la conexión al entorno de usuario.

***Actividad 4.2: Comparación de los datos medidos por el sistema digital con los datos de un medidor convencional para validar su precisión y eficacia:*** Con esta actividad buscamos realizar las comparaciones respectivas de los valores medidos por el medidor digital desarrollado y el medidor convencional para definir la eficacia y precisión del medidor digital.

***Actividad 4.3: elaboración del documento escrito:*** el objetivo de esta actividad es elaborar el documento escrito con los resultados y conclusiones obtenidos durante la investigación y presentarlos bajo las normativas que dispone el o la docente de titulación II.

## CAPÍTULO 4: ANÁLISIS Y RESULTADOS

### 4.1 Criterios y restricciones

Para definir los límites y requerimientos de funcionamiento del dispositivo, se ha elaborado una lista detallada de criterios y restricciones que considera múltiples aspectos técnicos, funcionales y operativos.

**Función (C<sub>1</sub>):** Debe realizar la medida y el guardado de la lectura de agua potable en una memoria no volátil y habilitarse para extraer la información almacenada mediante un computador por comunicación serial.

**Resistencia (C<sub>2</sub>):** El sistema debe ser capaz de resistir la corrosión por agua, humedad y temperatura presente en condiciones exteriores.

**Facilidad de instalación (C<sub>3</sub>):** El sistema debe permitir una rápida instalación, así como de desmontaje al momento de requerir mantenimiento.

**Rendimiento energético (C<sub>4</sub>):** El medidor debe ser capaz de optimizar el consumo de energía para poder funcionar de manera ininterrumpida.

**Interfaz de usuario (C<sub>5</sub>):** La interacción con la interfaz del dispositivo debe ser amigable para el usuario final, sin necesidad de conocimientos avanzados en software o computación.

**Conexión con tuberías comerciales:** El sistema debe ser fácil de instalar en las tuberías de ½ pulgada.

**Precisión:** El medidor debe tener una precisión de medición cercana a la de los medidores convencionales de agua.

### 4.2 Alternativas de solución

#### 4.2.1 Módulos de funcionamiento.

Para realizar el análisis del funcionamiento del dispositivo es conveniente dividir el dispositivo en tres diferentes módulos principales (ver Fig.26.) y analizar cada uno por separado, en este caso se identificó el módulo de procesamiento de datos, rendimiento energético y mecanismo de instalación.

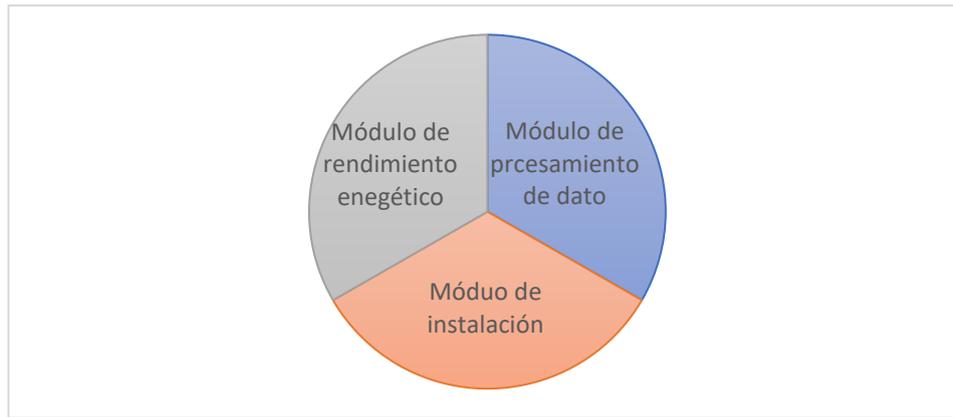


Fig 26. Módulos del dispositivo.

A continuación, en las Tablas 1-3 se presenta de manera detallada la descripción de cada una de las alternativas de los módulos. Para el módulo de procesamiento de datos se eligieron aquellas placas de microcontrolador que consumen menos energía eléctrica, viniendo a ser estas el ESP32 y Arduino Nano.

Tabla 1. Alternativas de solución del módulo de procesamiento de datos

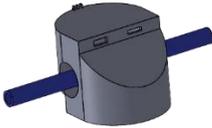
Soluciones	Descripción	Ventajas	Desventajas
<b>ESP 32</b>	Placa de microcontrolador con conexiones inalámbricas integradas	<ul style="list-style-type: none"> <li>-Económica</li> <li>-Tamaño reducido</li> <li>-Modo de Deep Sleep (10 - 150<math>\mu</math>A)</li> <li>-5 a 3.3V de funcionamiento</li> <li>-Mayor velocidad</li> </ul>	<ul style="list-style-type: none"> <li>-80 mA de consumo.</li> <li>-Requiere configuraciones extra de energía.</li> <li>-Librerías oficiales no disponibles</li> </ul>
<b>Arduino Nano</b>	Placa de tamaño reducido, comandos básicos de funcionamiento.	<ul style="list-style-type: none"> <li>-Económica</li> <li>-Tamaño reducido</li> <li>-3.3V y 5V con 19mA y 9 a 12mA de consumo respectivamente</li> <li>-Librerías básicas compatibles.</li> </ul>	<ul style="list-style-type: none"> <li>-Velocidad media</li> <li>-No requiere configuraciones extra de energía.</li> </ul>

Tabla 2. Alternativas de solución del módulo de rendimiento energético

Soluciones	Descripción	Ventajas	Desventajas
<b>Energía AC</b>	Mediante el uso de la red eléctrica del domicilio se energiza el medidor para su funcionamiento.	<ul style="list-style-type: none"> <li>-Fácil de implementar</li> <li>-No se requiere circuitos extra.</li> <li>-Económico</li> </ul>	<ul style="list-style-type: none"> <li>-Puede sufrir cortes de energía</li> </ul>

<b>Baterías</b>	Con el uso de baterías DC se alimenta la unidad de procesamiento por tiempo prolongados	-Periodo limitado de funcionamiento. -Fácil implementación. -No requiere fuentes externas	-Periodo limitado de suministro. -Circuito adicional para recargar. -puede elevar el costo y dimensiones del dispositivo - Precio elevado.
<b>Mixta</b>	Se plantea unir energía AC, Baterías y energía solar para garantizar un suministro continuo simulando un UPS.	-El dispositivo estará siempre alimentado. -No se pierden datos de medición.	-Circuito adicional. -Mayor número de componentes. -Precio moderadamente elevado.

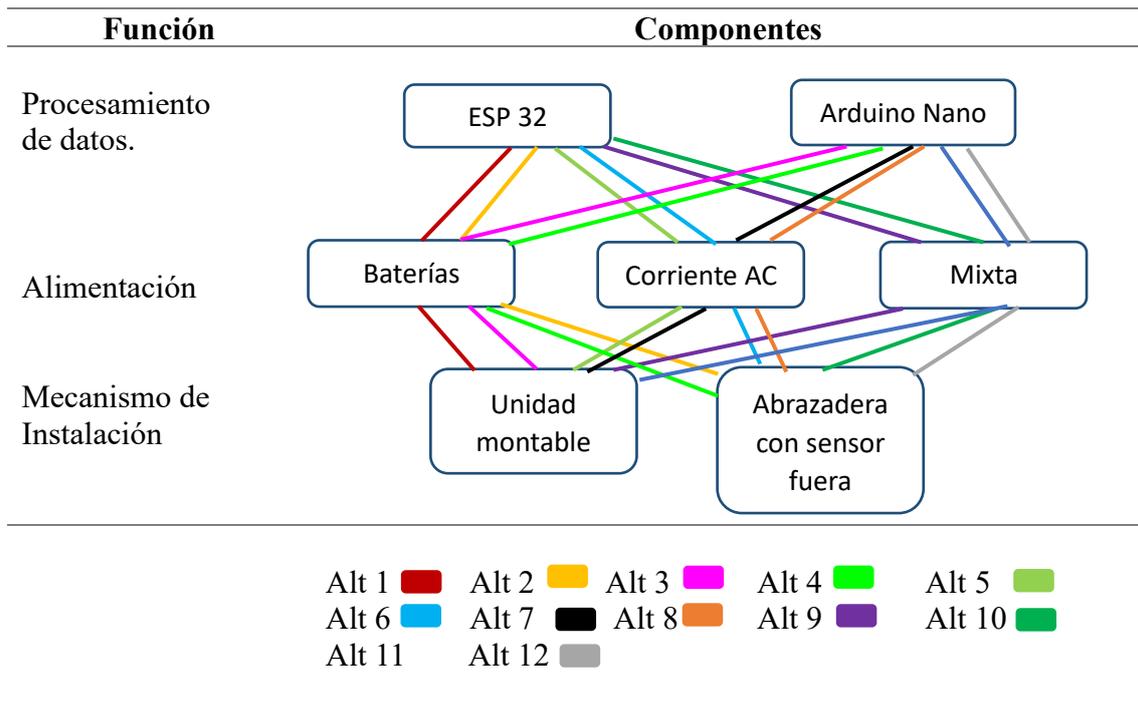
Tabla 3. Alternativas de solución del módulo de mecanismo de instalación.

<b>Soluciones</b>	<b>Descripción</b>	<b>Ventajas</b>	<b>Desventajas</b>
 <p><b>Unidad montable</b></p>	Se propone un dispositivo compuesto en una sola unidad, donde la instalación se realizará por sus extremos.	-Fácil de instalar. -Protege al sensor. -Ocupa menos espacio en tubería. -Fácil mantenimiento.	-Ligeramente Complejo de diseñar.
 <p><b>Abrazadera con sensor fuera</b></p>	Se propone una unidad tipo abrazadera para el circuito y el sensor ubicado externamente.	-Sencillo de diseñar. -Fácil mantenimiento.	-No protege al sensor. -Medianamente fácil de instalar -Ocupa mayor espacio en tubería. -Mayor número de piezas

#### 4.2.2 Matriz morfológica.

Después de establecer los criterios y restricciones del dispositivo se procede a realizar una matriz morfológica detallada en la Tabla 4 con el fin de obtener distintas alternativas de solución posibles considerando los componentes y funciones principales. Dentro de los aspectos principales a considerar para evaluar las alternativas están el procesamiento de los datos provenientes del sensor, la alimentación energética y la facilidad de instalación.

Tabla 4. Alternativas de solución



Los componentes que comprende cada una de las 12 soluciones antes obtenidas se los puede observar en la siguiente Tabla 5

Tabla 5. componentes de las alternativas

Alternativa	Procesamiento de datos.	Alimentación	Mecanismo de Instalación
1	ESP32	Baterías	Unidad montable
2	ESP32	Baterías	Abrazadera
3	Arduino Nano	Baterías	Unidad montable
4	Arduino Nano	Baterías	Abrazadera
5	ESP32	Corriente AC	Unidad montable
6	ESP32	Corriente AC	Abrazadera
7	Arduino Nano	Corriente AC	Unidad montable
8	Arduino Nano	Corriente AC	Abrazadera
9	ESP32	Mixta	Unidad montable
10	Arduino Nano	Mixta	Unidad montable
11	ESP32	Mixta	Abrazadera
12	Arduino Nano	Mixta	Abrazadera

### 4.3 Selección de la mejor solución

#### 4.3.1 Evaluación de criterios

A continuación, se realizó la evaluación de las alternativas frente a los criterios que destacan la funcionalidad del medidor, dichos criterios que se consideraron fueron la Función, la resistencia en general, la facilidad de la instalación, el rendimiento energético y por último la interfaz del usuario. A cada criterio se lo evalúa frente a otro distinto y se escribe en la casilla cuál de los dos tiene más importancia, para después realizar una sumatoria y finalmente una ponderación de importancia.

##### Lista de criterios

Función: (C1)

Resistencia: (C2)

Facilidad de instalación: (C3)

Rendimiento energético: (C4)

Interfaz de usuario: (C5)

Tabla 6. Evaluación de criterios

	C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>5</sub>	Nº veces	$\sum_{+1}$	Ponderación
C <sub>1</sub>		C <sub>1</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>1</sub>	2	3	0.13
C <sub>2</sub>			C <sub>3</sub>	C <sub>4</sub>	C <sub>2</sub>	1	2	0.15
C <sub>3</sub>				C <sub>4</sub>	C <sub>3</sub>	3	4	0.25
C <sub>4</sub>					C <sub>4</sub>	4	5	0.35
C <sub>5</sub>						0	1	0.12

#### 4.3.2 Ponderación de soluciones de acuerdo con cada criterio

Ahora teniendo claro los criterios y su ponderación se realiza una evaluación de cada criterio individual frente a todas las alternativas de solución previamente obtenidas tal como se puede observar en las Tablas 7-11.

Tabla 7. Comparación de alternativas con el criterio 1

C <sub>1</sub>	Alt 1	Alt 2	Alt 3	Alt 4	Alt 5	Alt 6	Alt 7	Alt 8	Alt 9	Alt 10	Alt 11	Alt 12	$\sum_{+1}$	Ponderación
Alt 1		0.5	1	1	0.5	0.5	1	1	0.5	0.5	1	1	9,5	0,14
Alt 2	0.5		1	1	0.5	0.5	1	1	0.5	0.5	1	1	9,5	0,14
Alt 3	0	0		0	0	0	0	0	0	0	0	0	1	0,01
Alt 4	0	0	0		0	0	0	0	0	0	0	0	1	0,01

<b>Alt 5</b>	0,5	0,5		1	1		1	1	1	1	1	1	11	0,16
<b>Alt 6</b>	0,5	0,5		1	1	1		1	1	1	1	1	11	0,16
<b>Alt 7</b>	0	0		0	0	0	0		0	0	0	0	1	0,01
<b>Alt 8</b>	0	0		0	0	0	0	0		0	0	0	1	0,01
<b>Alt 9</b>	0,5	0,5		1	1	1	1	1		1	1	1	11	0,16
<b>Alt 10</b>	0,5	0,5		1	1	1	1	1	1		1	1	11	0,16
<b>Alt 11</b>	0	0		0	0	0	0	0	0	0		0	1	0,01
<b>Alt 12</b>	0	0		0	0	0	0	0	0	0	0		1	0,01
<b>SUMA</b>													69	1

Alt 10=Alt 9=Alt 6 = Alt 5> Alt 1= Alt 2> Alt 3= Alt 4= Alt 7= Alt 8= Alt 11= Alt 12

Tabla 8. Comparación de alternativas con el criterio 2

<b>C<sub>2</sub></b>	<b>Alt 1</b>	<b>Alt 2</b>	<b>Alt 3</b>	<b>Alt 4</b>	<b>Alt 5</b>	<b>Alt 6</b>	<b>Alt 7</b>	<b>Alt 8</b>	<b>Alt 9</b>	<b>Alt 10</b>	<b>Alt 11</b>	<b>Alt 12</b>	$\sum^{+1}$	<b>Ponderación</b>
<b>Alt 1</b>		1	0,5	0	0,5	1	0,5	1	0,5	0,5	1	1	8,5	0,11
<b>Alt 2</b>	0		0	0,5	0	0,5	0	1	0	0,5	0	1	4,5	0,06
<b>Alt 3</b>	0,5	1		1	0	0	0,5	1	0	0,5	0,5	1	7	0,09
<b>Alt 4</b>	1	0,5	0		0	0	0	0,5	0	0	0	0,5	3,5	0,04
<b>Alt 5</b>	0,5	1	1	1		1	1	1	0,5	1	1	1	11	0,14
<b>Alt 6</b>	0	0,5	1	1	0		0,5	1	0	0,5	0,5	1	7	0,09
<b>Alt 7</b>	0,5	1	0,5	1	0	0,5		1	0	0	0,5	1	7	0,09
<b>Alt 8</b>	0	0	0	0,5	0	0	0		0	0	0	0,5	2	0,03
<b>Alt 9</b>	0,5	1	1	1	0,5	1	1	1		1	1	1	11	0,14
<b>Alt 10</b>	0,5	0,5	0,5	1	0	0,5	1	1	0		1	1	8	0,10
<b>Alt 11</b>	0	1	0,5	1	0	0,5	0,5	1	0	0		1	6,5	0,08
<b>Alt 12</b>	0	0	0	0,5	0	1	0	0,5	0	0	0		3	0,04
<b>SUMA</b>													79	1

Alt 5= Alt 9> Alt 1> Alt 10> Alt 3= Alt 6= Alt 7> Alt 11> Alt 2> Alt 4= Alt 12 > Alt 8

Tabla 9. Comparación de alternativas con el criterio 3

<b>C<sub>3</sub></b>	<b>Alt 1</b>	<b>Alt 2</b>	<b>Alt 3</b>	<b>Alt 4</b>	<b>Alt 5</b>	<b>Alt 6</b>	<b>Alt 7</b>	<b>Alt 8</b>	<b>Alt 9</b>	<b>Alt 10</b>	<b>Alt 11</b>	<b>Alt 12</b>	$\sum^{+1}$	<b>Ponderación</b>
<b>Alt 1</b>		1	0,5	1	0,5	1	0,5	1	0,5	1	0,5	1	9,5	0,12
<b>Alt 2</b>	0		0	0,5	0	0,5	0	0,5	0	0,5	0	0,5	3,5	0,04
<b>Alt 3</b>	0,5	1		1	0,5	1	0,5	1	0,5	1	0,5	1	9,5	0,12
<b>Alt 4</b>	0	0,5	0		0	0,5	0	0,5	0	0,5	0	0,5	3,5	0,04
<b>Alt 5</b>	0,5	1	0,5	1		1	0,5	1	0,5	1	0,5	1	9,5	0,12
<b>Alt 6</b>	0	0,5	0	0,5	0		0	0,5	0	0,5	0	0,5	3,5	0,04
<b>Alt 7</b>	0,5	1	0,5	1	0,5	1		1	0,5	1	0,5	1	9,5	0,12
<b>Alt 8</b>	0	0,5	0	0,5	0	0,5	0		0	0,5	0	0,5	3,5	0,04
<b>Alt 9</b>	0,5	1	0,5	1	0,5	1	0,5	1		1	0,5	1	9,5	0,12
<b>Alt 10</b>	0	0,5	0	0,5	0	0,5	0	0,5	0		0	0,5	3,5	0,04
<b>Alt 11</b>	0,5	1	0,5	1	0,5	1	0,5	1	0,5	1		1	9,5	0,12
<b>Alt 12</b>	0	0,5	0	0,5	0	0,5	0	0,5	0	0,5	0		3,5	0,04
<b>SUMA</b>													78	1

Alt 1= Alt 3= Alt 5= Alt 7= Alt 9= Alt 11> Alt 2= Alt 4= Alt 6= Alt 8= Alt 10= Alt 12

Tabla 10. Comparación de alternativas con el criterio 4

C <sub>4</sub>	Alt 1	Alt 2	Alt 3	Alt 4	Alt 5	Alt 6	Alt 7	Alt 8	Alt 9	Alt 10	Alt 11	Alt 12	$\sum^{+1}$	Ponderación
Alt 1		0.5	1	1	0.5	0.5	0.5	0.5	0	0	0	0	5,5	0,08
Alt 2	0.5		1	1	0	0.5	0.5	0.5	0	0	0	0	5	0,07
Alt 3	0	0		0.5	0.5	0.5	0	0	0	0	0	0	2,5	0,03
Alt 4	0	0	0.5		0	0	0	0	0	0	0	0	1,5	0,02
Alt 5	0.5	1	0.5	1		0.5	0.5	0.5	0	0	0	0	5,5	0,08
Alt 6	0.5	0.5	0.5	1	0.5		0.5	0.5	0	0	0	0	5	0,07
Alt 7	0.5	0.5	1	1	0.5	0.5		0.5	0	0	0	0	5,5	0,08
Alt 8	0.5	0.5	1	1	0.5	0.5	0.5		0	0	0	0	5,5	0,08
Alt 9	1	1	1	1	1	1	1	1		0.5	1	1	9	0,13
Alt 10	1	1	1	1	1	1	1	1	0.5		1	1	9	0,13
Alt 11	1	1	1	1	1	1	1	1	0	0		0.5	9	0,13
Alt 12	1	1	1	1	1	1	1	1	0	0	0.5		9	0,13
<b>SUMA</b>													72	1

**Alt 9= Alt 10= Alt 11= Alt 12> Alt 1= Alt 5= Alt 7= Alt 8> Alt 2= Alt 6> Alt 3> Alt 4**

Tabla 11. Comparación de alternativas con el criterio 5

C <sub>5</sub>	Alt 1	Alt 2	Alt 3	Alt 4	Alt 5	Alt 6	Alt 7	Alt 8	Alt 9	Alt 10	Alt 11	Alt 12	$\sum^{+1}$	Ponderación
Alt 1		0.5	1	1	0.5	0.5	1	1	0.5	0.5	1	1	9,5	0,12
Alt 2	0.5		1	1	0.5	0.5	1	1	0.5	0.5	1	1	9,5	0,12
Alt 3	0	0		0.5	0	0	0.5	0.5	0	0	0.5	0.5	3,5	0,04
Alt 4	0	0	0.5		0	0	0.5	0.5	0	0	0.5	0.5	3,5	0,04
Alt 5	0.5	0.5	1	1		0.5	1	1	0.5	0.5	1	1	9,5	0,12
Alt 6	0.5	0.5	1	1	0.5		1	1	0.5	0.5	1	1	9,5	0,12
Alt 7	0	0	0.5	0.5	0	0		0.5	0	0	0.5	0.5	3,5	0,04
Alt 8	0	0	0.5	0.5	0	0	0.5		0	0	0.5	0.5	3,5	0,04
Alt 9	0.5	0.5	1	1	0.5	0.5	1	1		0.5	1	1	9,5	0,12
Alt 10	0.5	0.5	1	1	0.5	0.5	1	1	0.5		1	1	9,5	0,12
Alt 11	0	0	0.5	0.5	0	0	0.5	0.5	0	0		0.5	3,5	0,04
Alt 12	0	0	0.5	0.5	0	0	0.5	0.5	0	0	0.5		3,5	0,04
<b>SUMA</b>													78	1

**Alt 1= Alt 2= Alt 5= Alt 6= Alt 9= Alt 10> Alt 3= Alt 4= Alt 7= Alt 8= Alt 11= Alt 12**

### 4.3.3 Ponderación final

Luego de obtener la ponderación de las alternativas con cada uno de los criterios se realiza una evaluación final, multiplicando cada valor obtenido por el valor de la ponderación de cada criterio y eligiendo la alternativa que más se ajuste a las necesidades del sistema.

Tabla 12. Evaluación de las alternativas frente a todos los criterios

	C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>5</sub>	Sum	Ponderación
Alt 1	0,14x0.13	0,11x0.15	0,12x0.25	0,08x0.35	0,12x0.12	0,1071	3
Alt 2	0,14x0.13	0,06x0.15	0,04x0.25	0,07x0.35	0,12x0.12	0,0761	8
Alt 3	0,01x0.13	0,09x0.15	0,12x0.25	0,03x0.35	0,04x0.12	0,0601	10
Alt 4	0,01x0.13	0,04x0.15	0,04x0.25	0,02x0.35	0,04x0.12	0,0291	12
Alt 5	0,16x0.13	0,14x0.15	0,12x0.25	0,08x0.35	0,12x0.12	0,1142	2
Alt 6	0,16x0.13	0,09x0.15	0,04x0.25	0,07x0.35	0,12x0.12	0,0832	6
Alt 7	0,01x0.13	0,09x0.15	0,12x0.25	0,08x0.35	0,04x0.12	0,0776	7
Alt 8	0,01x0.13	0,03x0.15	0,04x0.25	0,08x0.35	0,04x0.12	0,0486	11
Alt 9	0,16x0.13	0,14x0.15	0,12x0.25	0,13x0.35	0,12x0.12	0,1317	1
Alt 10	0,16x0.13	0,10x0.15	0,04x0.25	0,13x0.35	0,12x0.12	0,1057	4
Alt 11	0,01x0.13	0,08x0.15	0,12x0.25	0,13x0.35	0,04x0.12	0,0936	5
Alt 12	0,01x0.13	0,04x0.15	0,04x0.25	0,13x0.35	0,04x0.12	0,0676	9

Como se puede evidencia en la Tabla 12 La alternativa que mejor cumple con los requerimientos del sistema es la Alt 9, por lo que el sistema de medición se compondrá de una placa ESP32, una alimentación energética mixta y un tipo de mecanismo de instalación de tipo Unidad montable.

#### 4.4 Solución propuesta para el sistema digital de medición

##### 4.4.1 Funcionamiento general del sistema.

El dispositivo tiene como objetivo realizar la medida del consumo de agua potable mediante un sensor digital de caudal, para llevar un registro de consumo, estos datos son guardados periódicamente en una memoria microSD. Para la interacción y visualización de los datos el dispositivo se conecta por comunicación serial a una interfaz HMI instalada en un computador.

##### 4.4.2 Diagrama de bloques para el funcionamiento del sistema

A continuación, se presenta el diagrama de bloques que ilustra el funcionamiento del dispositivo medidor de agua.

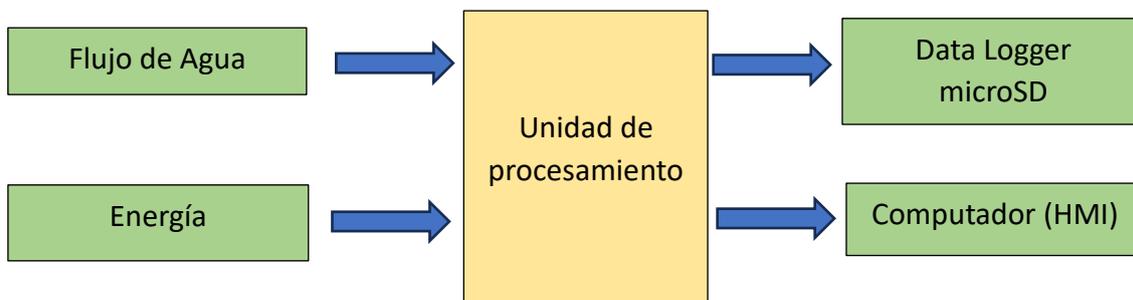


Fig 27. Diagrama de bloques del sistema.

##### 4.4.3 Descripción del modelo propuesto para el sistema digital.

La solución propuesta (ver Fig. 28) consiste en un dispositivo compuesto por una unidad central diseñada para proteger y organizar los componentes internos. La carcasa está dividida en dos partes principales: una carcasa superior (pieza N°3) y una carcasa inferior (pieza N°5).

La **carcasa superior** (pieza N°3) sostiene el sensor de caudal (pieza N°4), dejando sus entradas expuestas para facilitar la conexión a una tubería de ½ pulgada mediante uniones estándar. Además, esta carcasa alberga el circuito de procesamiento (pieza N°2), que está protegido del ambiente externo mediante una tapa superior (pieza N°1), fijada mediante tornillos milimétricos M3.

Por otra parte, la carcasa **inferior** (pieza N°5) contiene el circuito conmutador de energía (pieza N°6), responsable de alternar automáticamente entre la alimentación por corriente alterna (AC) y el suministro por baterías en caso de un corte de energía. Debajo del circuito conmutador se encuentran las baterías recargables (pieza N°7), conectadas parcialmente al circuito para garantizar su recarga.

La **carcasa inferior** y sus componentes están protegidos por una **tapa inferior** (pieza N°8), también fijada con tornillos M3. La conexión del dispositivo a la red de suministro de agua potable se realiza mediante dos uniones de ½ pulgada (pieza N°9).

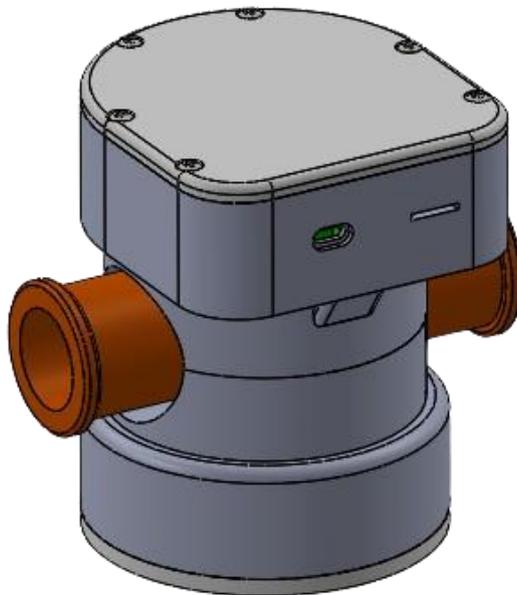


Fig 28. Modelo de la solución propuesta.

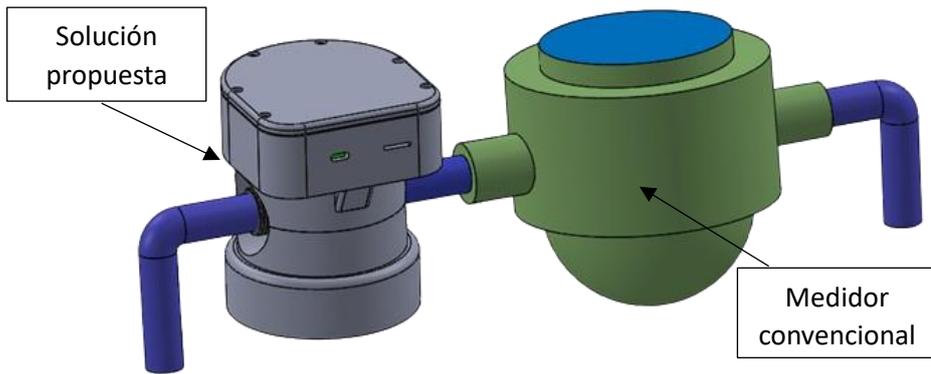


Fig 29. Conexión del modelo propuesto al medidor convencional de agua doméstico.

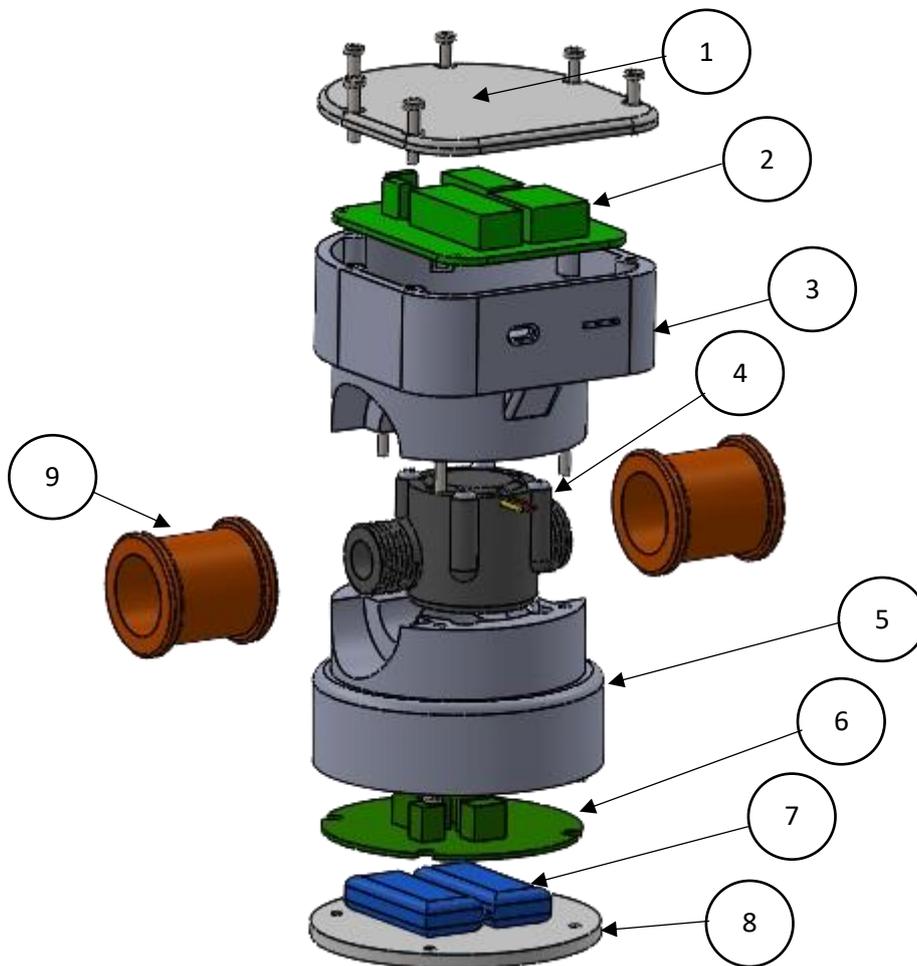


Fig 30. Vista explosionada de la solución propuesta.

Tabla 13. Componentes de la solución propuesta.

N° Pieza	Componente	Material	Cantidad
1	Tapa Superior	Filamento PLA	1
2	Circuito de procesamiento	Baquelita con grabado de cobre	1
3	Carcasa superior	Filamento PLA	1
4	Sensor de caudal	Plástico (Nylon reforzado)	1
5	Carcasa inferior	Filamento PLA	1
6	Circuito conmutador	Baquelita con grabado de cobre	1
7	Baterías	Litio	2
8	Tapa de inferior	Filamento PLA	1
9	Unión de ½	PVC	2

#### 4.5 Construcción del sistema digital de medición.

##### 4.5.1 Diseño del circuito.

Una vez identificados los componentes necesarios para la construcción electrónica del medidor, se elaboró una tabla que detalla las principales características de funcionamiento y los valores de consumo y aporte de energía correspondientes a cada elemento. Los resultados se presentan a continuación en la Tabla 14.

Tabla 14. Componentes electrónicos usados en el sistema

Componente	Precio \$	Cantidad	Características
<b>Batería LiPo 3.7</b>	3,50	2	Corriente: 800 mAh Voltaje: 3.7 V
<b>Relé 5 V DC</b>	0,75	1	Voltaje bobina: 5v DC Contactos: NO, NC Corriente consumo: 12.5mA
<b>Diodo 1N4007</b>	0,20	2	Voltaje consumo: 0,7V DC Corriente consumo: 5,0 µA
<b>Elevador Boost</b>	2,50	1	Voltaje de entrada: 2V a 24V DC Voltaje de salida: 5V a 28V DC V.Salida ajustable (Regulable) Corriente de salida: 2A máx. Potenciómetro: giro en sentido contrario al reloj Eficiencia: 96% Consumo corriente: 0.9mA sin carga

<b>Módulo TP4056</b>	1,75	1	Corriente entrada: 1A ajustable Voltaje de entrada: 4,5 V-5,5 V Voltaje de carga completa: 4,2 V
<b>Módulo RTC</b>	2,00	1	Voltaje de entrada: 3.3V-5V Corriente de consumo: 2 mA Batería integrada.
<b>Módulo SD</b>	2,00	1	Voltaje de entrada: 3.3V-5 V DC Corriente de consumo: 2 mA Comunicación: SPI
<b>ESP32</b>	14,00	1	Voltaje de entrada: 3.3V-5 V DC Corriente de consumo: 80 mA
<b>Sensor YF-S201</b>	9,00	1	Caudal: 1-30 L/min Presión: <= 1.75 MPa Voltaje de entrada: 3.3 -24 V DC Corriente de consumo: 15mA Salida: Onda cuadrada pulsante Factor de conversión: 7.5

Con las características de consumo de todos los componentes antes mencionados en la Tabla 14. se procede a realizar el cálculo de la suma de los voltajes máximos y sobre todo de corrientes de consumo para seleccionar una fuente de alimentación adecuada para el sistema, los resultados se los puede evidenciar en la Tabla 15.

Tabla 15. Consumo de voltaje y amperaje de componentes

Componente	Voltaje (V DC)	Corriente (mA)	Cantidad	Amperaje total (mA)
<b>Relé 5 V DC</b>	5v	12,5	1	12,5
<b>Diodo 1N4007</b>	0,7	5,0 $\mu$ A	2	0,00010
<b>Elevador Boost</b>	2 a 24	0,9 mA sin carga 45mA con carga	1	54
<b>Módulo TP4056</b>	4,5 -5,5	Soporta 1A	1	1.5

<b>Módulo RTC</b>	3.3-5.	2	1	2
<b>Módulo SD</b>	3.3-5	2	1	2
<b>ESP32</b>	3.3-5	80	1	80
<b>Sensor YF-S201</b>	3.3 24	15	1	15
			Total	167

Considerando el voltaje de consumo promedio más alto entre todos los componentes, se determinó que la fuente de alimentación debe proporcionar una conversión de 110 V AC a 5 V DC. En cuanto a la corriente de consumo, dado que esta supera 1 A, se optó por una fuente de 5 V DC con una capacidad máxima de 2 A, para garantizar un suministro adecuado.

Después de definir los componentes a usarse se diseñó el circuito de alimentación mixta, el mismo implementa un sistema de conmutación automática que permite alternar entre dos fuentes de energía: la red eléctrica del hogar y una batería. El sistema prioriza la alimentación desde la red eléctrica mientras esta esté disponible. En caso de una interrupción repentina del suministro eléctrico, el circuito conmuta automáticamente para que el sistema de medición continúe funcionando alimentado por la batería, sin interrupciones en el suministro de energía. La fuente de 5V es la encargada de recargar la batería en los momentos de presencia del suministro eléctrico.

- **Esquema lineal de conexión**

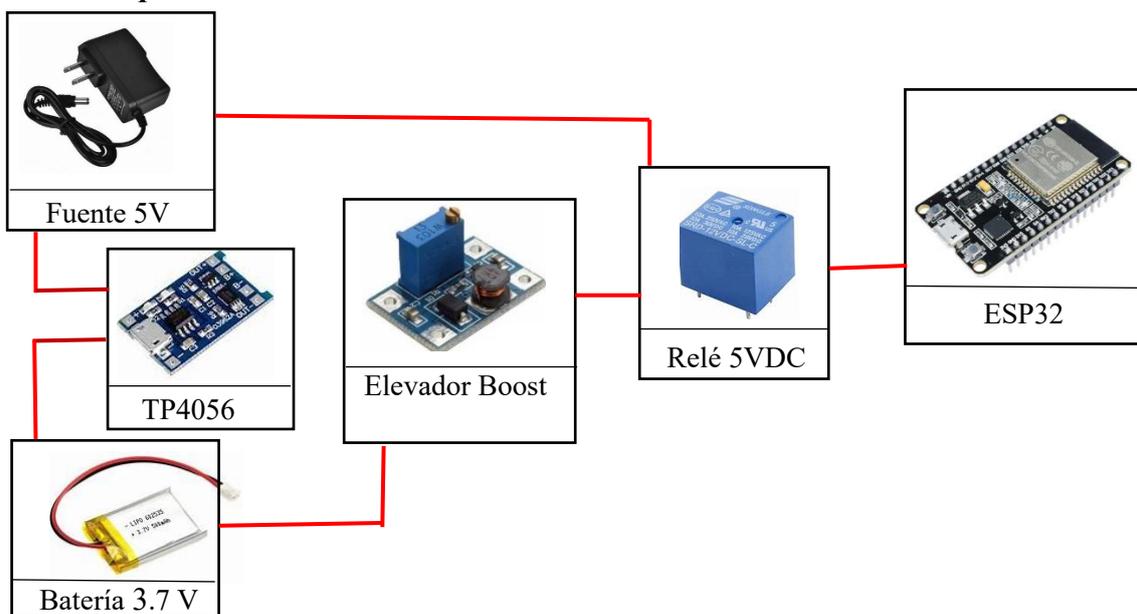


Fig 31. Esquema lineal de conexión para conmutación mixta

- **Esquema del circuito electrónico.**

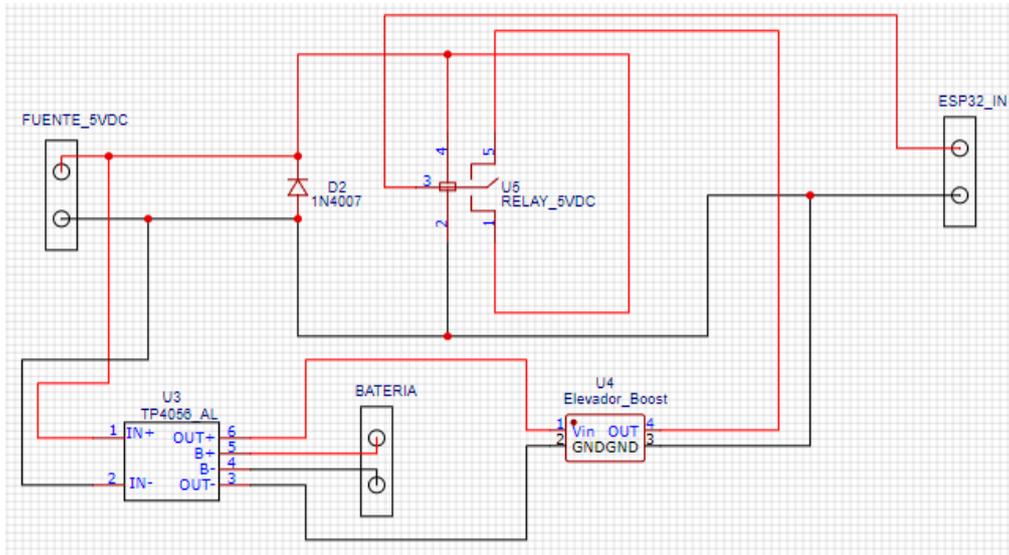


Fig 32. Esquema del circuito electrónico para conmutación mixta.

Del mismo modo se realizó el diseño del circuito para la unidad de procesamiento del medidor de agua, este incluye como componentes principales un ESP32, que se encarga del control y la gestión de las funciones principales del sistema, un módulo de reloj para proporcionar la referencia temporal necesaria, un módulo lector de tarjetas SD para el almacenamiento de datos y un sensor de caudal para medir el consumo de agua.

- **Esquema lineal de conexión**

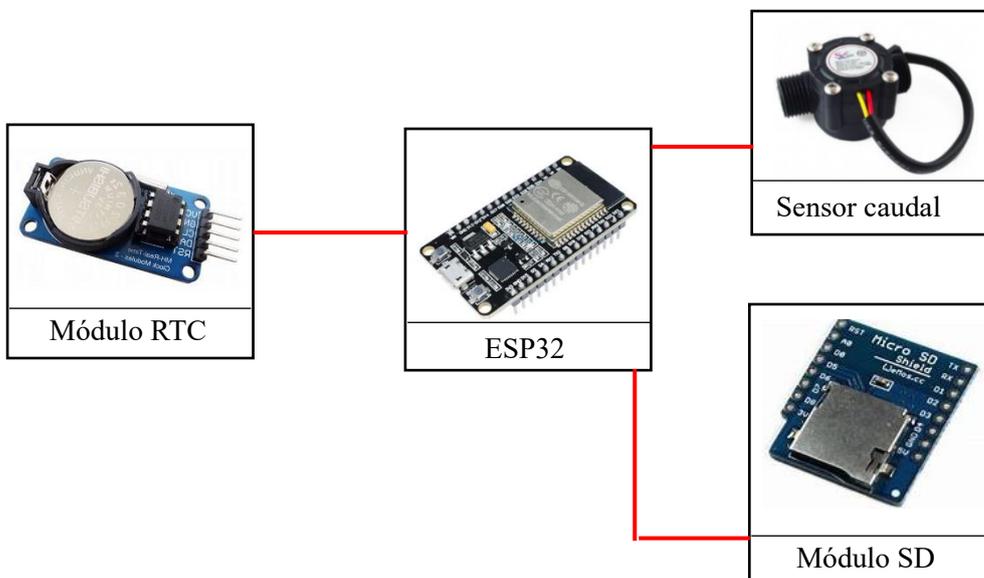


Fig 33. Esquema lineal de conexión para unidad de procesamiento.

- **Plano de circuito electrónico.**

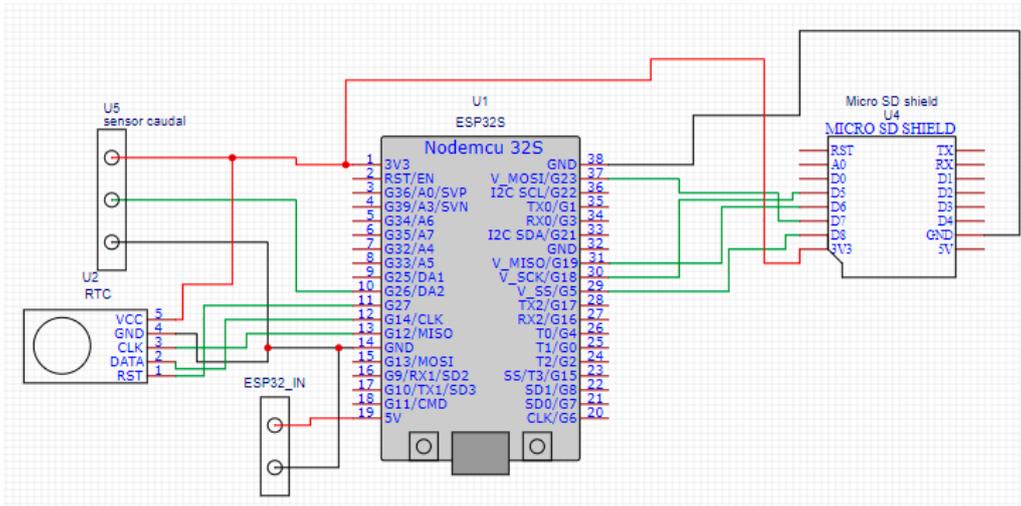


Fig 34. Circuito electrónico para unidad de procesamiento.

#### 4.5.2 Diseño de la placa física.

Una vez elaborados los esquemas de conexión correspondientes a los dos circuitos (procesamiento y conmutador), se procedió a la fabricación de la placa de circuito impreso (PCB). Para ello, se utilizó el software EasyEDA, en su versión estándar, con el fin de generar los archivos de diseño necesarios para la impresión de las pistas del circuito (ver Fig.35).

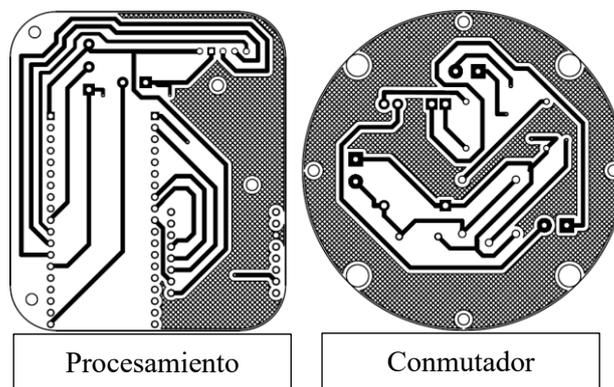


Fig 35. Pistas generadas de circuito de procesamiento y conmutador

Posteriormente, se empleó la técnica de transferencia térmica para plasmar las imágenes del diseño en una baquelita de una sola cara, utilizando papel fotográfico y un planchado controlado (ver **Fig.36.**).

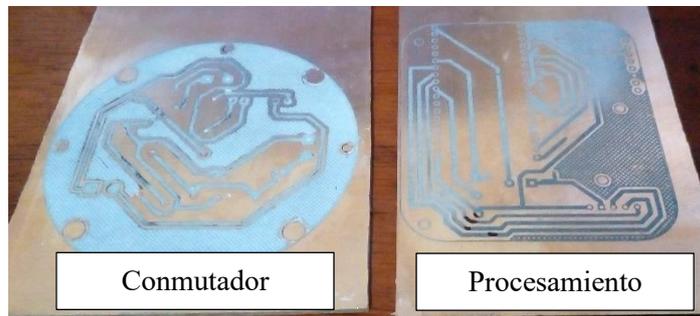


Fig 36. Transferencia térmica por planchado.

Una vez transferido el diseño, las pistas del circuito fueron reveladas mediante el uso de ácido férrico, eliminando el cobre sobrante y dejando únicamente las conexiones necesarias.

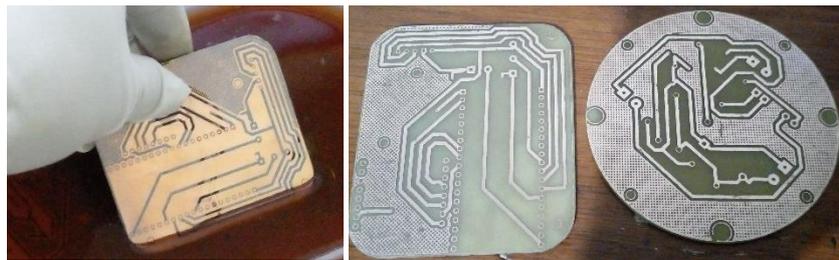


Fig 37. revelado de circuitos en acido férrico.

Finalmente, se realizó la perforación de la placa para insertar los componentes electrónicos, seguido del proceso de soldadura para asegurar las conexiones. Los resultados obtenidos se presentan en las ilustraciones siguientes (ver **Fig.38.**).

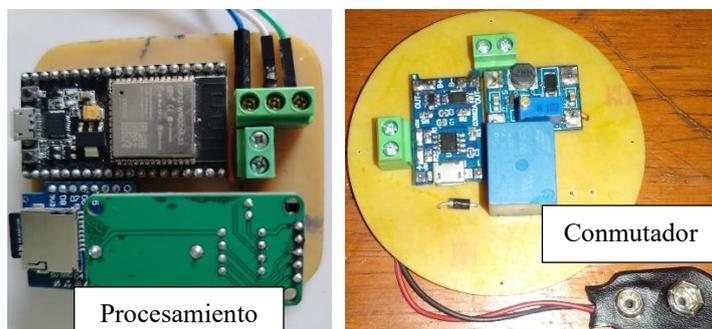


Fig 38. Montaje y soldadura de componentes electrónicos.

#### 4.5.3 Calibracion del sensor de caudal.

El sensor de flujo YF-S201 calcula el caudal mediante un factor de conversión predeterminado de 7,5. No obstante, para mejorar la precisión de las mediciones, es fundamental realizar una calibración manual que permita determinar un factor de conversión ajustado a las condiciones específicas del sistema.

El proceso de calibración consiste en registrar los pulsos generados por el sensor al medir un volumen conocido de agua. Para este caso, se ha utilizado un volumen de referencia de 2000 ml (2 litros). Los resultados obtenidos durante la calibración se presentan en la Tabla 16.

El código utilizado para obtener los pulsos generados por el sensor YF-S201 se encuentra detallado en el Anexo A.

Tabla 16. Pulsos generados

<b>Nº Prueba</b>	<b>ml</b>	<b>Pulsos</b>
<b>1</b>	2000	853
<b>2</b>	2000	854
<b>3</b>	2000	834
<b>4</b>	2000	847
<b>5</b>	2000	828
<b>6</b>	2000	861
<b>7</b>	2000	893
<b>8</b>	2000	845
<b>9</b>	2000	869
<b>10</b>	2000	868
<b>Promedio</b>		855,2

Una vez obtenido los registros de pulsos en función del volumen de agua se procede a calcular el nuevo factor de conversión con la siguiente fórmula:

$$K = \frac{n^\circ}{\text{volumen} * 60} \quad (2)$$

Donde:

- K: factor de conversión (pulsos/segundo)
- n°: promedio de pulsos
- volumen: volumen (agua) de referencia

- 60: segundos

$$K = \frac{855,2}{2 * 60} = 7,12$$

Ahora con este valor K podemos realizar el cálculo del caudal en L/h o L/min dependiendo del uso. Además, para obtener el volumen en ml hay que tomar en cuenta que 1 pulso de este modelo en específico equivale a 2,25 ml, entonces podemos calcular el equivalente de 1 litro en pulsos mediante una regla de 3.

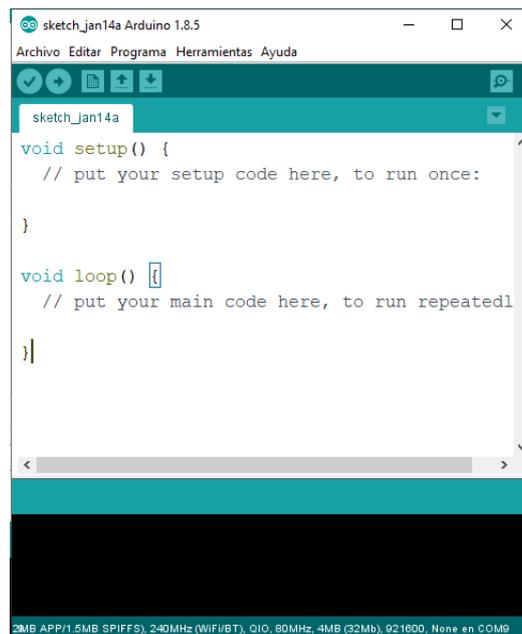
$$Pulsos = \frac{1000 \text{ ml} * 1 \text{ pulso}}{2,25 \text{ ml}} = 444,44 \quad (3)$$

Como se puede observar en la ecuación (3) 444,44 pulsos equivalen a 1000 ml o 1 litro. La implementación de estos cálculos en el código de programación se los puede observar en el Anexo B

#### 4.5.4 Programación

##### 4.5.4.1 Software de programación

La programación del microcontrolador ESP32 se llevó a cabo utilizando el entorno de desarrollo Arduino IDE, debido a su amplia uso en proyectos basados en microcontroladores. Este software emplea el lenguaje de programación C++, lo que facilita el acceso a una amplia variedad de bibliotecas que se actualizan constantemente gracias a que es de código abierto[42] .



```

sketch_jan14a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly
}
  
```

2MB APP/1.5MB SPIFFS, 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, None en COM9

Fig 39. Entorno de programación Arduino IDE.

#### 4.5.4.2 Librerías

Para programar la placa ESP32, es necesario contar con bibliotecas específicas que permitan la correcta compilación del código para este microcontrolador. Estas bibliotecas se pueden obtener a través de los recursos de código abierto disponibles en el entorno de desarrollo Arduino IDE, desde donde es posible descargarlas e instalarlas. A continuación se detallan las librerías usadas.

- esp32 by Espressif Systems
- SD.h
- SPI.h
- virtuabotixRTC.h

Del mismo modo para la programación del entorno HMI se usaron varios complementos visuales contenidos en los paquetes de NuGet de VisualStudio, estos son:

- AlphaGradientPanel
- AltoControls
- LiveCharts
- LiveCharts.WinForms
- LiveCharts.Wpf
- WinFormAnimation

#### 4.5.4.2 Diagramas de flujo

Para garantizar el correcto funcionamiento del dispositivo, se siguió una secuencia de pasos estructurada que permite el sensado, registro de los datos de consumo de agua y comunicación con la interfaz de usuario. Para visualizar el proceso de programación, se diseñó los diagramas de flujo correspondientes al procesamiento de los datos y el entorno HMI (ver Fig. 40 y Fig. 41).

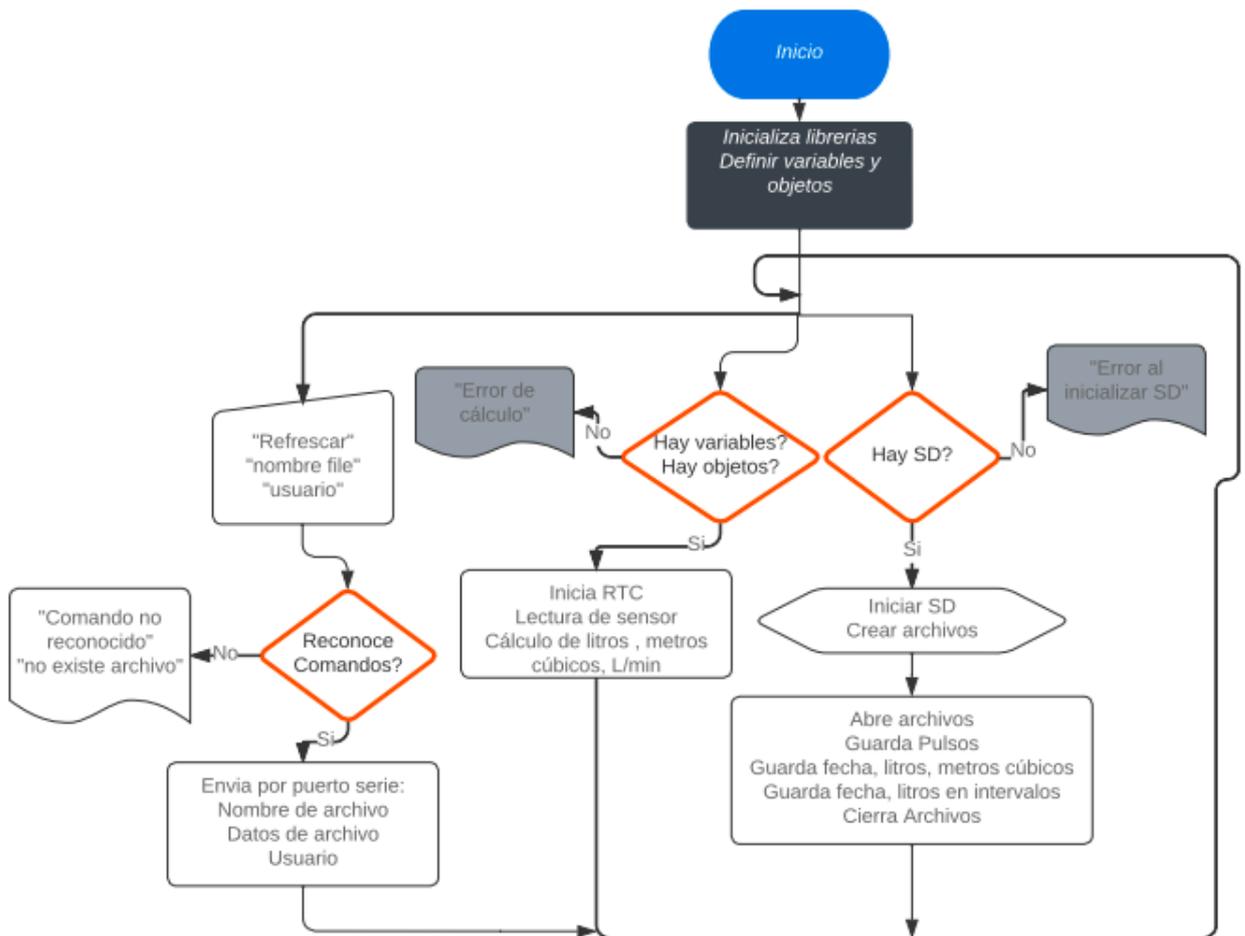


Fig 40. Diagrama de flujo de procesamiento de datos.

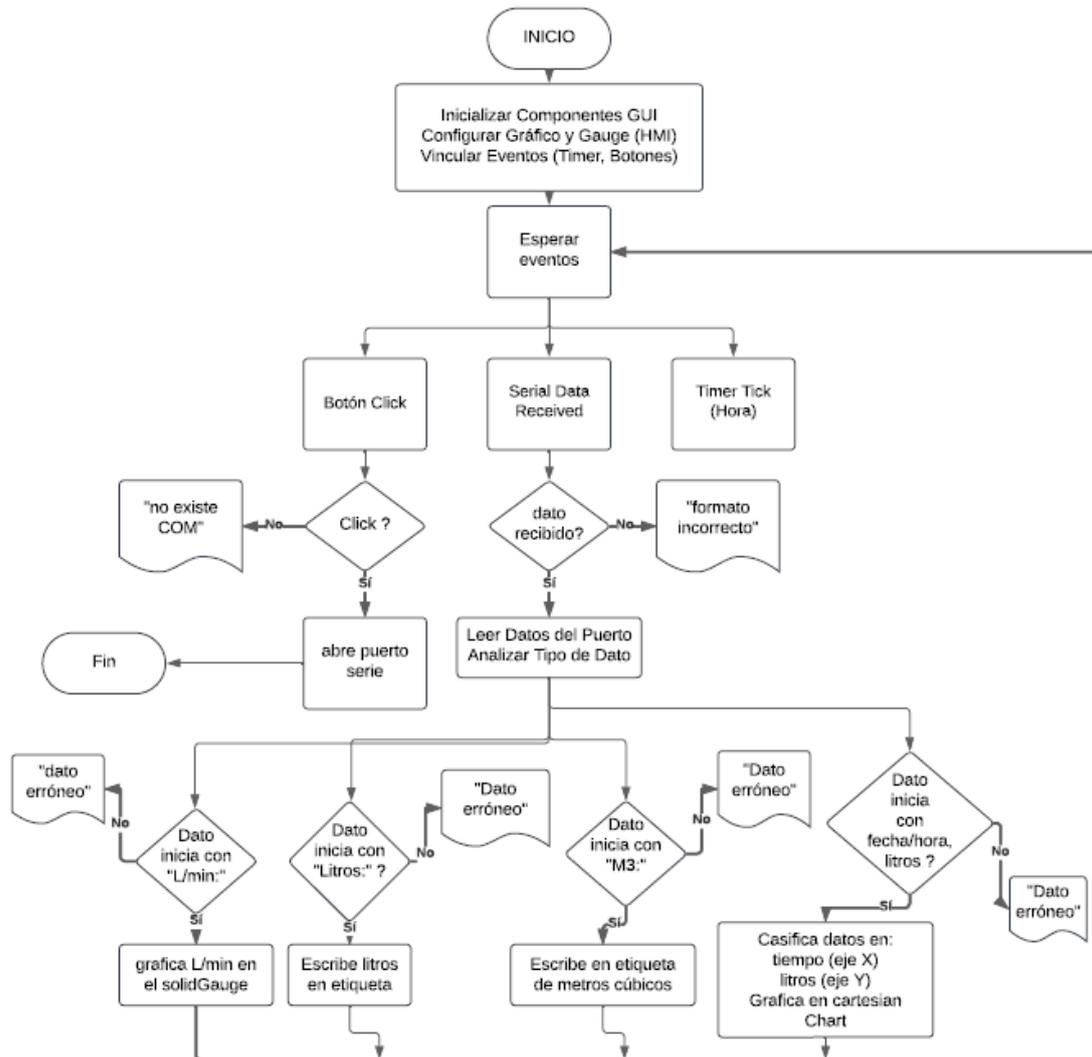


Fig 41. Diagrama de flujo de HMI.

#### 4.5.5 HMI

En cuanto al entorno de usuario, se decidió desarrollar una interfaz de usuario (HMI) utilizando Visual Studio y el lenguaje C# para una aplicación de escritorio compatible con sistemas operativos Windows. Esta interfaz se comunica de manera directa y serial con el puerto serie del ESP32 a través de un cable USB a micro USB. Mediante controles de botones integrados en el HMI, los cuales son activados por el usuario, se envían comandos específicos al puerto serie del microcontrolador para ejecutar diversas acciones, tales como conectar el microcontrolador, solicitar el nombre de usuario, obtener el contenido de archivos y recibir distintos datos enviados por el microcontrolador. El código de C# para el HMI se encuentra en el ANEXO C.

A continuación se muestran las ventanas del HMI al abrirse y luego se muestra el mismo entorno en funcionamiento (ver Fig. 42 y Fig. 43).



Fig 42. Interfaz HMI.

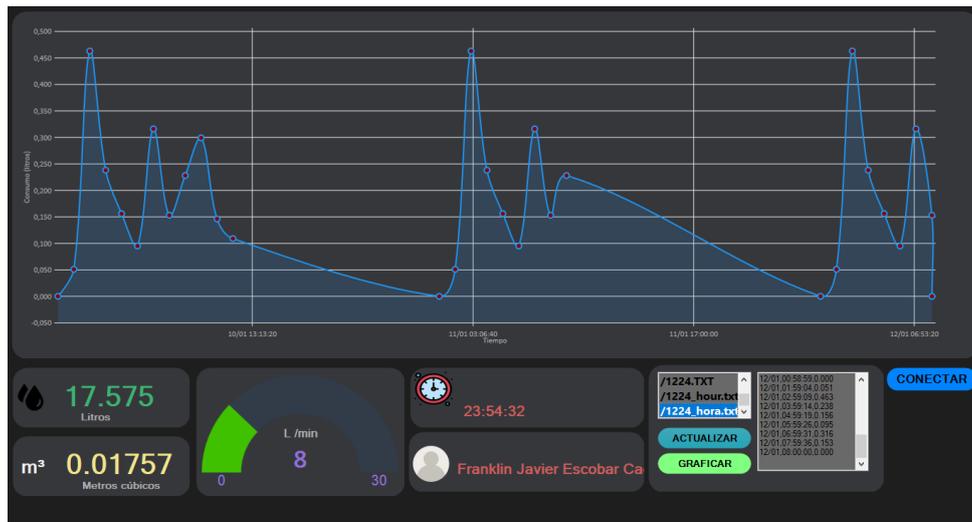


Fig 43. Interfaz HMI en funcionamiento.

Como se muestra en la Fig. 5.4, el HMI incluye un gráfico para visualizar los datos de consumo de agua, el cual varía según el archivo seleccionado en la sección de archivos. En la ilustración 5.4, se presenta un gráfico que muestra los picos de consumo de agua capturados cada hora, lo que permite identificar las horas de mayor consumo. Por otro lado, la siguiente ilustración (ver Fig. 44) muestra un gráfico con los consumos acumulados de manera gradual.

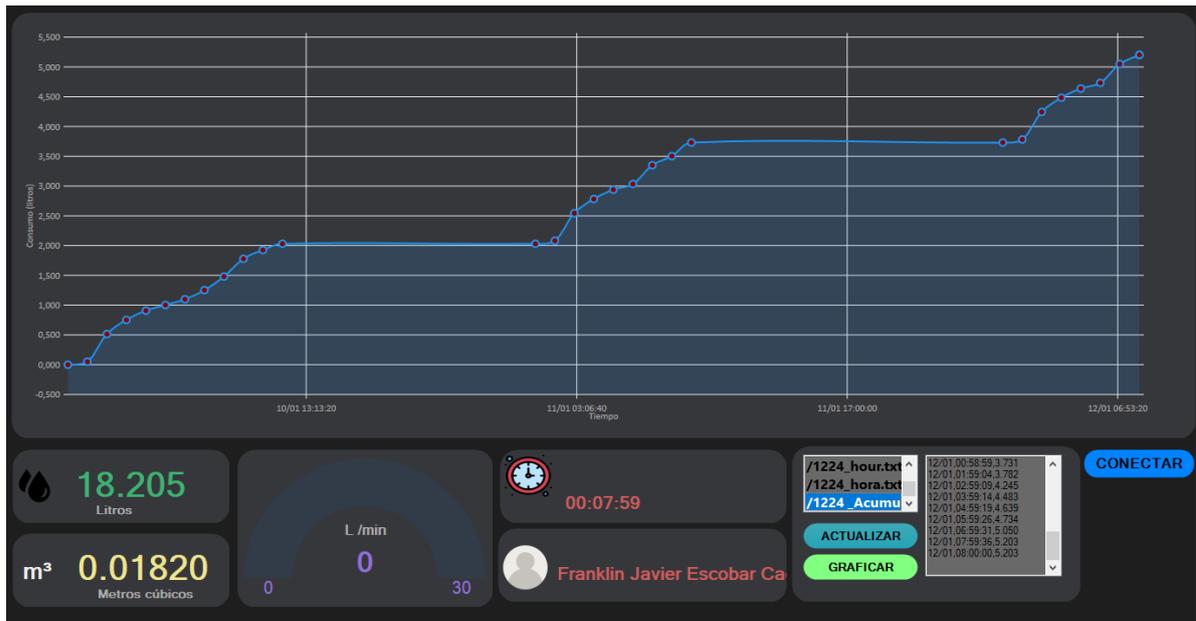


Fig 44. Interfaz HMI en funcionamiento, consumo litros acumulados.

#### 4.5.5 Diseño estructural.

Para el diseño estructural del dispositivo, se consideró la creación de una estructura unitaria que facilite su instalación. Por ello, se decidió diseñar una única unidad montable, compuesta por cuatro piezas principales, las cuales se detallan a continuación. Los planos correspondientes a estas piezas se encuentran en el Anexo D.

##### 4.5.5.1 Tapa Superior

Es la pieza encargada de cubrir el circuito de procesamiento de datos ante las condiciones externas, contiene 6 orificios para ajustarse con tornillos M3.

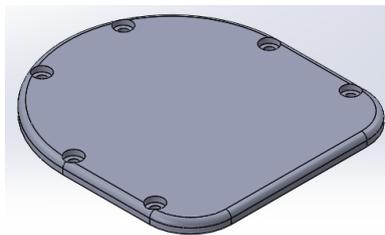


Fig 45. Modelo CAD de tapa superior

##### 4.5.5.2 Carcasa superior

Esta pieza está diseñada para sostener el circuito de procesamiento, contando con cuatro orificios para tornillos M2 que permiten fijar de forma segura dicho circuito. Además, incorpora seis orificios pasantes destinados a alojar tornillos de ajuste M3. Estos orificios incluyen un espacio específico para insertar tuercas compatibles, lo que garantiza

un ajuste firme y seguro con la tapa superior. Adicionalmente, la pieza está diseñada para alojar parcialmente el sensor de caudal, asegurando su correcta posición y funcionamiento dentro del sistema. Por último, esta pieza incorpora dos orificios pasantes en la parte frontal, diseñados para facilitar la conexión del cable micro USB y el acceso a la tarjeta microSD.

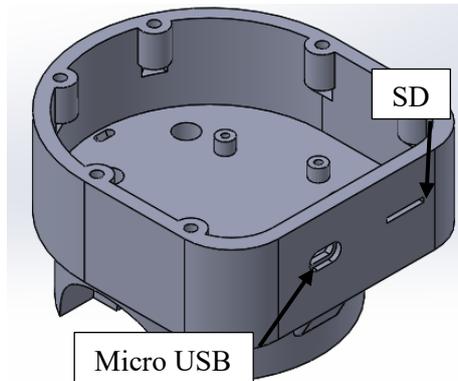


Fig 46. Modelo CAD de carcasa superior

#### 4.5.5.3 Carcasa inferior

Esta pieza se encarga de sostener parcialmente el sensor de caudal por su parte inferior y cuenta con un orificio destinado al paso de los cables de alimentación hacia el circuito de procesamiento. Su sujeción con la carcasa superior se realiza mediante tornillos y tuercas M3, asegurando un ajuste firme. Además, esta pieza está diseñada para alojar en su interior el circuito conmutador y las baterías de respaldo.

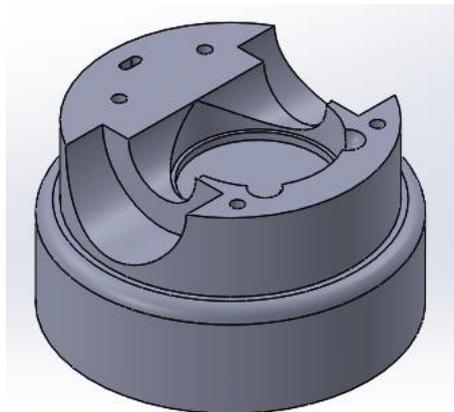


Fig 47. Modelo CAD de carcasa inferior

#### 4.5.5.4 Tapa inferior

Esta pieza tiene la función de proteger el circuito conmutador y las baterías de respaldo. Se fija a la carcasa inferior mediante tornillos y tuercas M3, para lo cual cuenta con orificios pasantes que facilitan su correcta sujeción.

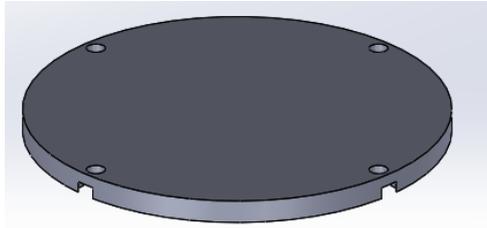


Fig 48. Modelo CAD de tapa inferior

#### 4.5.6 Ensamblaje e implementación del dispositivo en condiciones reales

La fabricación de las distintas piezas antes presentadas se lo realizaron mediante la técnica de impresión 3d, esto debido a que es una fabricación sencilla en comparación con otros métodos. El material usado para la impresión de las piezas es el PLA, este material es lo suficientemente resistente antes condiciones de esfuerzo considerables, y debido a que este dispositivo no mantiene contacto físico variable no necesita mayor resistencia a golpes, además este material ofrece una resistencia a temperaturas de hasta 60 C°.

En la Fig.49, Fig.50. y Fig. 51 se muestran el resultado de la impresión, ensamblaje e instalación in situ de todos los componentes del dispositivo. Las respectivas hojas de procesos de cada pieza se encuentran en el ANEXO E.

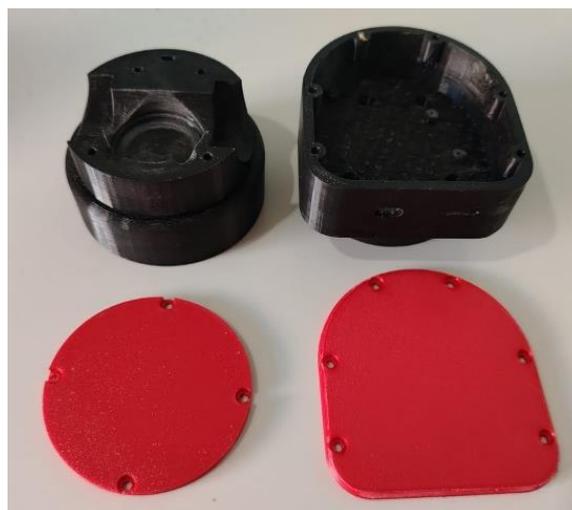


Fig 49. Carcasa impresa.



Fig 50. Ensamblaje de los componentes del dispositivo.



Fig 51. Instalación del dispositivo in situ.

#### **4.6 Resultados de pruebas**

Para validar el funcionamiento general del dispositivo, se realizaron diversas pruebas enfocadas en evaluar la resistencia de la estructura, el desempeño del circuito electrónico, la precisión de las lecturas, y finalmente, una prueba integral de funcionamiento.

##### **4.6.1 Pruebas de estructura**

Para evaluar la resistencia de la estructura frente a la humedad, se llevaron a cabo simulaciones de un entorno con presencia de chorro de agua en todas las direcciones, utilizando un aerosol. El objetivo fue verificar la integridad de los circuitos electrónicos alojados dentro de las carcasas y asegurar que no se produjeran infiltraciones de agua en su interior los resultados de las pruebas se encuentran en la Tabla 17.

Tabla 17. Pruebas de resistencia estructural

<b>N° Prueba</b>	<b>Tiempo</b>	<b>Observaciones</b>
1	1 min	Ninguna infiltración
2	2 min	Mínima infiltración
3	40 s	Ninguna infiltración
4	4 min	Considerable infiltración
5	3 min	Poca infiltración
6	10 s	Ninguna infiltración
7	30 s	Mínima infiltración
8	10s	Ninguna infiltración
9	5 s	Ninguna infiltración
10	1 min 30 s	Poca infiltración

#### 4.6.2 Pruebas del circuito

Para validar el funcionamiento general del circuito del sistema, se llevaron a cabo pruebas específicas en dos aspectos clave: la carga de la batería y el correcto funcionamiento de la conmutación automática del circuito entre las fuentes de energía. Estas pruebas incluyeron la evaluación del tiempo de carga de la batería, la estabilidad y rapidez de la conmutación automática en caso de pérdida de la fuente principal de energía. Los resultados se los puede visualizar en las Tablas 18 – 19.

Tabla 18. Pruebas de carga de batería.

<b>tiempo</b>	<b>V inicial</b>	<b>V final</b>	<b>Corriente de carga</b>	<b>% de carga</b>
30	3.13	3.72	1.01	55.14
60	3.72	3.89	0.87	71.03
90	3.89	3.96	0.61	77.57
120	3.96	4.03	0.48	84.11
150	4.03	4.07	0.36	87.85
180	4.07	4.09	0.30	89.72
210	4.09	4.11	0.25	91.59
240	4.11	4.15	0.22	95.33
270	4.15	4.2	0.09	100

Tabla 19. Pruebas de conmutación de fuentes.

<b>N° Prueba</b>	<b>Condición Inicial</b>	<b>Evento de Conmutación</b>	<b>Fuente Activa Tras Conmutación</b>	<b>Tiempo de Conmutación (ms)</b>	<b>Comportamiento</b>	<b>Observaciones</b>
1	Fuente	Desconexión fuente	Batería	50	Normal	Cambio exitoso
2	Batería	Reconexión de fuente	Fuente	30	Normal	Cambio sin interrupciones
3	Fuente	Desconexión de fuente	Batería	30	Normal	Breve reinicio del sistema
3	Batería	Reconexión de fuente	Fuente	20	Normal	Breve reinicio del sistema

### 4.6.3 Pruebas de precisión

Para las pruebas de precisión en la medición, el dispositivo se instaló parcialmente en tuberías de media pulgada. Posteriormente, se dejó fluir un volumen de agua referencial a través del sistema. Para estas pruebas, se utilizaron 1,5 litros de agua, los cuales fueron suministrados de manera controlada mediante una jarra que dirigía el líquido hacia las tuberías.

La ejecución de la prueba se encuentra documentada en la Fig.52, mientras que los resultados de las mediciones registradas por el dispositivo se presentan en la Tabla 20.

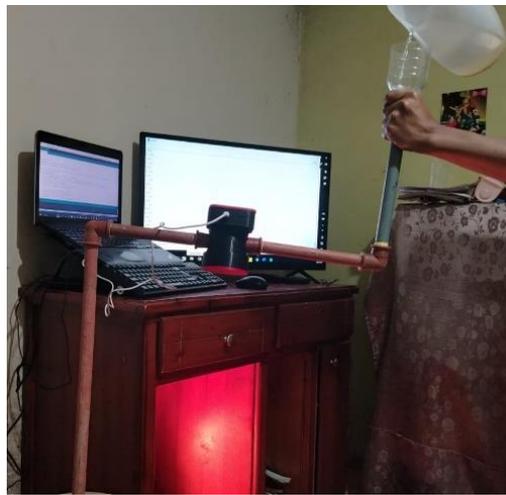


Fig 52. Pruebas de precisión.

Tabla 20. Resultados de pruebas de precisión

<b>N° Prueba</b>	<b>Agua (L)</b>	<b>Medidor digital (L)</b>
1	1,5	1.48
2	1,5	1.43
3	1,5	1.46
4	1,5	1.50
5	1,5	1.44
6	1,5	1.46
7	1,5	1.40
8	1,5	1.48
9	1,5	1.47
10	1,5	1.49

### 4.6.4 Pruebas de funcionamiento largo plazo

Para validar el dispositivo, se llevó a cabo una comparación entre las mediciones obtenidas por el dispositivo digital y las registradas por el medidor convencional. Las pruebas se realizaron durante un período de cinco días consecutivos, durante los cuales el dispositivo digital permaneció en funcionamiento de manera continua. Las lecturas de ambos dispositivos se tomaron en intervalos de 8 horas, los resultados obtenidos se presentan en la Tabla 21.

Tabla 21. Resultados de pruebas a largo plazo

<b>Día (dd / mm / aa)</b>	<b>Intervalo de 8 horas (hh:mm:ss)</b>	<b>Medidor convencional (L)</b>	<b>Medidor digital (L)</b>
18/2/2025	16:10:00	0,0	0,0
	0:10:09	144,5	143,988
19/2/2025	8:10:10	202,0	201,886
	16:10:10	466,5	466,070
20/2/2025	0:10:11	526,0	525,373
	8:10:12	744,0	743,446
	16:10:13	752,0	751,625
21/2/2025	0:10:00	842,5	842,003
	8:10:00	1023,5	1022,921
	16:10:00	1.106,5	1.106,032
22/2/2025	0:10:00	1.186,0	1.185,328
	8:10:00	1.276,5	1.275,961
	16:10:00	1.341,0	1.340,842
23/2/2025	0:10:00	1.435,5	1.435,022
	8:10:00	1.501,0	1.500,623
	16:10:00	1.664,0	1.664,474

## **4.7 Análisis de resultados**

### **4.7.1 Análisis de resultados de la estructura.**

A partir de los resultados obtenidos en las pruebas de impermeabilidad de la estructura se determinó la presencia de infiltraciones considerables de agua en el interior de las carcasas. Para mitigar este problema, se decidió emplear un material aislante, optando por cinta de teflón para sellar tanto las uniones de las carcasas como las de las tapas superior e inferior.

### **4.7.1 Análisis de resultados del circuito.**

A partir de los resultados obtenidos en las pruebas realizadas a los circuitos, especialmente al circuito de conmutación de fuentes de energía, se determinó que la conmutación entre fuentes funciona correctamente, aunque con breves retardos. Además, se evaluó el comportamiento del proceso de carga de la batería, obteniendo un tiempo total de carga de 270 minutos para un banco de baterías de 3.7 V y 800 mAh conectadas en serie. Por otro lado, el módulo de carga TP4056 mostró un funcionamiento adecuado en la regulación de la corriente de carga, como se detalla en la Tabla 2.7. Durante el proceso de carga, la corriente disminuyó progresivamente de 1.01 A a 0.09 A conforme la batería alcanzaba su capacidad máxima.

#### 4.7.1 Análisis de resultados de precisión.

Mediante el cálculo del error absoluto y relativo a partir de los datos obtenidos en las pruebas de precisión, se determinó que el dispositivo presenta un error de medición del 2.60%. Los resultados detallados de estos cálculos se encuentran disponibles en la Tabla 22.

Tabla 22. Error de medición del medidor digital.

<b>Medida (litros)</b>	<b>Error absoluto (litros)</b>	<b>Error relativo (%)</b>
1.48	0.02	1.33
1.43	0.07	4.67
1.46	0.04	2.67
1.50	0.00	0.00
1.44	0.06	4.00
1.46	0.04	2.67
1.40	0.10	6.67
1.48	0.02	1.33
1.47	0.03	2.00
1.49	0.01	0.67
<b>Error promedio</b>	0.039	2.60

#### 4.7.2 Análisis de resultados largo plazo.

El medidor digital demostró un error relativo promedio de 0.36% durante 5 días de medición en la tubería, como se muestra en la Fig.53. Esta alta precisión se debe a las condiciones estables de presión y caudal dentro de la tubería. En cambio, en las pruebas iniciales de precisión realizadas con una jarra con la referencia de 1,5 Litros, el error relativo fue de 2.6%, ya que la variación en el flujo de líquido al arrojar el agua afectó la medición. Estos resultados resaltan que el medidor digital opera con mayor precisión en entornos controlados y sellados.

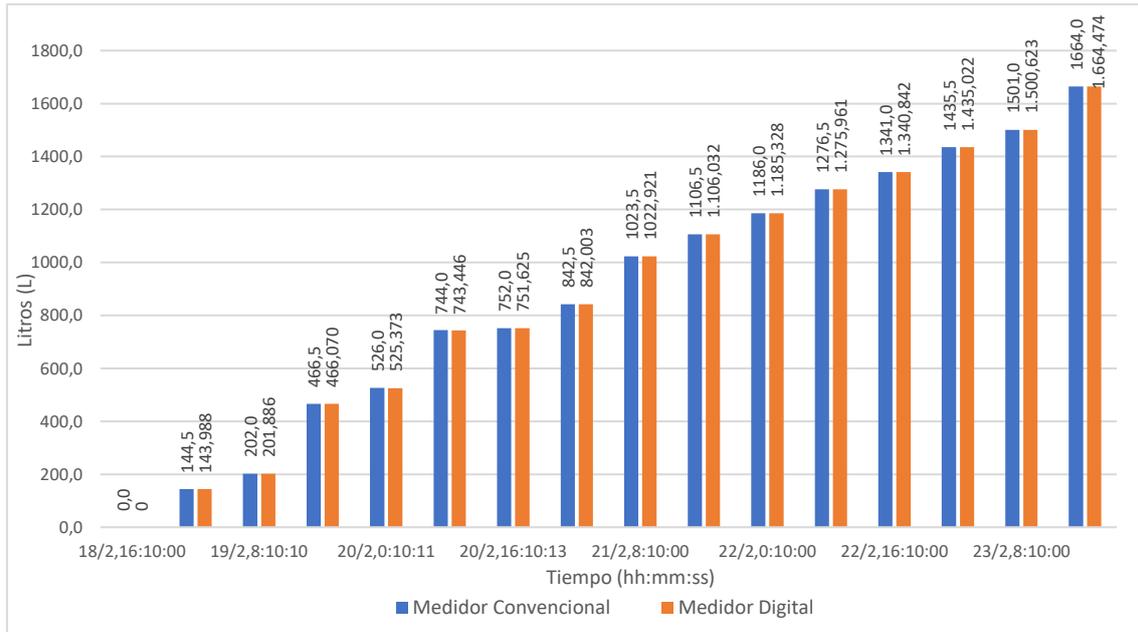


Fig 53. Comparación de medidor convencional y medidor digital.

## CONCLUSIONES

Se logró desarrollar un dispositivo digital para la medición del consumo de agua potable, abordando las principales dificultades de la toma de lecturas en la práctica convencional.

Para la selección del sensor de caudal, se llevó a cabo un análisis comparativo de los sensores electrónicos disponibles en el mercado, evaluando su compatibilidad con microcontroladores y sus condiciones de funcionamiento. Como resultado, se seleccionó el sensor de caudal YF-S201, debido a su precisión y adaptabilidad a las tuberías estándar de agua potable residencial.

Se diseñaron los diagramas de flujo tanto para el microcontrolador como para la interfaz humano-máquina (HMI), asegurando una secuencia lógica de funcionamiento. La programación del microcontrolador se realizó en el entorno de desarrollo Arduino IDE, mientras que la HMI se desarrolló en C# utilizando Visual Studio.

Considerando la facilidad de instalación como un criterio clave, se diseñó un modelo compacto y de fácil montaje en la línea de tuberías, similar a un medidor convencional. El modelado se llevó a cabo en software CAD y las piezas fueron fabricadas mediante impresión 3D en PLA. A partir de una revisión bibliográfica, se concluyó que este material puede soportar condiciones ambientales adversas.

Se realizaron pruebas experimentales para evaluar distintas áreas del dispositivo. En primer lugar, se analizó la resistencia estructural ante condiciones de humedad y chorros de agua, dando como resultado pequeñas infiltraciones de agua al interior de las carcasas, para lo cual se resolvió usando fómix como aislante entre las uniones del dispositivo. Posteriormente, se validó el correcto funcionamiento del circuito electrónico donde se concluyó que existe breves periodos de reinicio del sistema y se realizaron pruebas de precisión en la medición del caudal dando como resultado un error de medición del 2.6%.

Finalmente, se llevaron a cabo pruebas de medición en un entorno real con funcionamiento ininterrumpido durante un periodo de 5 días. Los resultados arrojaron un margen de error del 0,36 % en comparación con un medidor convencional, demostrando un desempeño aceptable para su aplicación.

## **RECOMENDACIONES**

Para mejorar el funcionamiento del circuito conmutador de energía, se recomienda reemplazar el relé de estado sólido por un MOSFET, ya que esto reducirá los tiempos de conmutación y evitará interrupciones prolongadas en el suministro de energía, previniendo así reinicios inesperados del sistema.

Para futuras investigaciones, se recomienda aprovechar al máximo las capacidades de conectividad del ESP32, como Bluetooth, Wi-Fi o ESP-NOW, con el fin de desarrollar un sistema completamente inalámbrico. Esto permitiría la transmisión de datos en tiempo real a una plataforma de monitoreo remoto, eliminando la necesidad de almacenamiento local o intervenciones manuales para la recolección de datos.

Para garantizar un envío remoto eficiente de datos, se recomienda implementar una red LoRaWAN (Long Range Wide Area Network) que centralice la comunicación de los medidores digitales en un único nodo de recepción.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] H. R. Mejía Mejía, "Lectura de medidores eléctricos analógicos mediante visión artificial," Universidad Católica de Cuenca, Cuenca, 2023. Accessed: Oct. 09, 2024. [Online]. Available: <https://dspace.ucacue.edu.ec/handle/ucacue/14061>
- [2] P. Daniel, A. Enrique, A. Villagra, and L. Guillermo, "Redes Inteligentes de Agua: Factores y métodos para la predicción del consumo residencial de agua potable," Apr. 2018.
- [3] Jorge Helmbrecht, "Transformación digital de redes inteligentes: retos y riesgos," Barcelona, May 2018.
- [4] JUAN CARLOS FARINANGO TUQUERRES, "SISTEMA DE MEDIDOR INTELIGENTE DE AGUA POTABLE DOMICILIARIO PARA LA JUNTA ADMINISTRADORA DE AGUA 'EI ABRA'.," UNIVERSIDAD TÉCNICA DEL NORTE, Ibarra, 2018. doi: <https://repositorio.utn.edu.ec/handle/123456789/8758>.
- [5] Pablo Andrés Gómez Cadena, "Prototipo electrónico de medición de caudal para el registro de lecturas de medidores de agua potable," Universidad Técnica del Norte, Ibarra, 2023. doi: <https://repositorio.utn.edu.ec/handle/123456789/13840>.
- [6] Padilla Cevallos Sabrina Nataly and Hernández Granda Paúl Guillermo, "Diseño de un prototipo de un sistema de recolección de información para medidores de agua potable en la ciudad de Quito," UDLA, Quito, 2015. Accessed: Oct. 09, 2024. [Online]. Available: <http://dspace.udla.edu.ec/handle/33000/4477>
- [7] E. Pruna, V. Andaluz, C. Molina, R. Lara, C. Naranjo, and I. Escobar, "Medidor digital de agua potable con comunicación inalámbrica," Jul. 2016. [Online]. Available: [www.cdc.org/sv/archivos/folleto-conozca-](http://www.cdc.org/sv/archivos/folleto-conozca-)
- [8] Paredes Paucar Héctor Alexis, "DISEÑO DE UN PROTOTIPO DE NODO INTELIGENTE INTEGRABLE A UNA RED INALÁMBRICA PARA TELEMEDICIÓN DEL CONSUMO DE AGUA POTABLE COMO PROPUESTA PARA EL GADM DE GUANO," ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO, Riobamba, 2AD.
- [9] "Agua potable | Qué es, Definición y Concepto." Accessed: Oct. 09, 2024. [Online]. Available: <https://enciclopediaonline.com/es/agua-potable/>
- [10] "CRITERIOS DE CALIDAD Y GESTIÓN DEL AGUA POTABLE - PRADANA PÉREZ Juan Ángel , GALLEGU PICÓ Alejandrina , GARCÍA AVILÉS Javier , BRAVO YAGÜE Juan Carlos , GARCÍA MAYOR María Asunción , PANIAGUA GONZÁLEZ Gema - Google Libros." Accessed: Oct. 09, 2024. [Online]. Available: [https://books.google.com.mx/books?hl=es&lr=&id=\\_bOWDwAAQBAJ&oi=fnd&pg=PP1&dq=gesti%C3%B3n+del+agua+potable.+&ots=WCJiXPv8SW&sig=KN20389TkLp4fGfiFtWgpilJk\\_w#v=onepage&q=gesti%C3%B3n%20del%20agua%20potable.&f=false](https://books.google.com.mx/books?hl=es&lr=&id=_bOWDwAAQBAJ&oi=fnd&pg=PP1&dq=gesti%C3%B3n+del+agua+potable.+&ots=WCJiXPv8SW&sig=KN20389TkLp4fGfiFtWgpilJk_w#v=onepage&q=gesti%C3%B3n%20del%20agua%20potable.&f=false)
- [11] Ley, "LEY ORGANICA DE RECURSOS HIDRICOS USOS Y APROVECHAMIENTO DEL AGUA", Accessed: Oct. 09, 2024. [Online]. Available: [www.lexis.com.ec](http://www.lexis.com.ec)
- [12] C. P. Br. TUESTA BARBOZA, "EVALUACIÓN DEL SUBREGISTRO DE LOS MEDIDORES DE ACUERDO A SU ANTIGÜEDAD EN LA CIUDAD DE CAJAMARCA PARA EL AÑO 2015," Universidad Privada del Norte, Perú, 2015.

- [13] “Medidores de flujo electromagnéticos | Badger Meter.” Accessed: Oct. 09, 2024. [Online]. Available: <https://www.badgermeter.com/es-us/productos/medidores/medidores-de-flujo-electromagneticos/>
- [14] Pereira Herrera María, “Propuesta de diseño del procedimiento para la toma de lectura con los nuevos medidores inteligente de Essbio S.A.” UNIVERSIDAD DEL BIO-BIO, 2017.
- [15] A. Fernando, “Medidor electrónico de agua residencial con comunicación LPWAN y aplicación IoT,” UNIVERSIDAD TECNOLÓGICA ISRAEL, 2022. Accessed: Oct. 09, 2024. [Online]. Available: <http://repositorio.uisrael.edu.ec/handle/47000/3379>
- [16] “Medidor Ultrasonico De Agua Inteligente 100% lot - LAIN HOLDINGS.” Accessed: Oct. 09, 2024. [Online]. Available: <https://lainholding.com/producto/medidor-ultrasonico-de-agua/>
- [17] AIN HOLDING, “Medidores de agua domésticos inteligentes IOT .” Accessed: May 12, 2025. [Online]. Available: <https://lainholding.com/producto/medidores-de-agua-domesticos-iot/>
- [18] Hebei Shang Hong Agua Metros Tecnología Co, “Medidor De agua Inteligente LORA.”
- [19] M. T. P. Q. Marco Antonio Bello U, *MEDICION DE PRESIÓN Y CAUDAL*, vol. 28. Chile, 2000.
- [20] Mechatronics, “Sensor de flujo de agua.” Accessed: Oct. 12, 2024. [Online]. Available: <https://naylampmechatronics.com/sensores-liquido/108-sensor-de-flujo-de-agua-12-yf-s201.html>
- [21] S. Madakam, R. Ramaswamy, S. Tripathi, S. Madakam, R. Ramaswamy, and S. Tripathi, “Internet of Things (IoT): A Literature Review,” *Journal of Computer and Communications*, vol. 3, no. 5, pp. 164–173, May 2015, doi: 10.4236/JCC.2015.35021.
- [22] O. Bhat, P. Gokhale, and S. Bhat, “Introduction to IOT,” *International Advanced Research Journal in Science, Engineering and Technology ISO*, vol. 3297, no. 1, 2007, doi: 10.17148/IARJSET.2018.517.
- [23] Dogan Ibrahim, “Microcontroller Based Applied Digital Control,” 2006, doi: 10.1002/0470863374.fmatter.
- [24] Jeriot Magallanes Hernandez, “Estudio comparativo de placas electrónicas Arduino en el mundo actual,” Perú, Feb. 23, 2023. doi: <https://doi.org/10.60091/ric.2023.v2n1.01>.
- [25] M. I. Hafidhin, A. Saputra, Y. Ramanto, and S. Samsugi, “ALAT PENJEMURAN IKAN ASIN BERBASIS MIKROKONTROLER ARDUINO UNO,” Dec. 2020.
- [26] “Placa Uno R3 compatible Arduino + Cabo USB - Usinainfo.” Accessed: Oct. 10, 2024. [Online]. Available: <https://www.usinainfo.com.br/placas-arduino-compatible/placa-uno-r3-compatible-arduino-cabo-usb-3513.html>
- [27] SARAGURO ERAS ANA LUCIA, “PROTOTIPO ELECTRONICO PARA PREVENIR ATAQUES DE FEMICIDIO MEDIANTE UNA ALERTA INMEDIATA DE GPS,” UNIVERSIDAD TECNOLÓGICA ISRAEL, 2020.
- [28] MANTECH, “Arduino Nano,” 2018.

- [29] Hardwarelibre, "Arduino Nano." Accessed: Oct. 11, 2024. [Online]. Available: <https://www.hwlibre.com/arduino-nano/>
- [30] H. J. El-Khozondar *et al.*, "A smart energy monitoring system using ESP32 microcontroller," *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 9, p. 100666, Sep. 2024, doi: 10.1016/J.PRIME.2024.100666.
- [31] J. Ikiss, "SISTEMA DE ADQUISICIÓN DE DATOS CON ESP32," Universidad Politécnica de Catalunya., 2020. doi: <http://hdl.handle.net/2117/344400>.
- [32] Horacio D. Vallejo, "PLC LOS CONTROLADORES LÓGICOS PROGRAMABLES," *Saber Electrónica*, Accessed: Oct. 12, 2024. [Online]. Available: [www.editorialquark.com.ar](http://www.editorialquark.com.ar)
- [33] Bolivar Díaz Chag, "Diseño y construcción de un equipo Emulador de Memorias Eprom - 2716-2732-2764 apoyado por un Computador," Escuela Superior Politécnica Del Litoral, Guayaquil, 1991. doi: <http://www.dspace.espol.edu.ec/handle/123456789/59983>.
- [34] Geenkel Coss Lara, Mildred González Hau, Carlos Brito Loeza, and Anabel Martin González, "COMPUTACIONAL LEARNING AND IMAGING RESEARCH," Nov. 2016.
- [35] José H. Moyano, "Clase 02: Hardware básico Sistemas Embebidos," 2019.
- [36] Alejandro Daniel Perez, "Protocolos de comunicación entre microcontroladores. Caso de estudio: Protocolo CAN," UNIVERSIDAD NACIONAL DE LA PLATA, 2016. doi: <http://sedici.unlp.edu.ar/handle/10915/72131>.
- [37] "¿Que es la Comunicación Serial?" Accessed: Oct. 12, 2024. [Online]. Available: [https://iarobotxyz.blogspot.com/2023/07/que-es-la-comunicacion-serial.html#google\\_vignette](https://iarobotxyz.blogspot.com/2023/07/que-es-la-comunicacion-serial.html#google_vignette)
- [38] "COMPARISON BETWEEN SERIAL COMMUNICATION PROTOCOLS (SPI, I2C & USART)." Accessed: Oct. 12, 2024. [Online]. Available: [https://electronoobs.com/eng\\_circuitos\\_tut36.php#google\\_vignette](https://electronoobs.com/eng_circuitos_tut36.php#google_vignette)
- [39] "Leer y escribir en una tarjeta SD o micro SD con Arduino." Accessed: Oct. 14, 2024. [Online]. Available: <https://www.luisllamas.es/tarjeta-micro-sd-arduino/>
- [40] Constantino Tancara Q., "Temas Sociales - LA INVESTIGACION DOCUMENTAL," Dec. 1993. Accessed: Oct. 12, 2024. [Online]. Available: [http://revistasbolivianas.umsa.bo/scielo.php?pid=S0040-29151993000100008&script=sci\\_arttext&tIng=en](http://revistasbolivianas.umsa.bo/scielo.php?pid=S0040-29151993000100008&script=sci_arttext&tIng=en)
- [41] M. Metodológico, "Capitulo III. Marco Metodológico".
- [42] "Arduino Documentation," 2015, Accessed: Jan. 13, 2025. [Online]. Available: [https://www.sparkfun.com/arduino\\_guide](https://www.sparkfun.com/arduino_guide)



```

//          BIBLIOTECAS

//*****
*****

#include <SD.h>

#include <SPI.h>

#include <virtuabotixRTC.h> // Biblioteca para el RTC

//*****
*****

//          TAREAS PARA SEGUNDO NÚCLEO DEL ESP32

//*****
*****

TaskHandle_t Tarea0; // Tarea0 Procesar Lecturas
TaskHandle_t Tarea2; // Tarea2 Procesar Lecturas1
TaskHandle_t Tarea1; // Tarea1 para manejar reconocerComandos y leerArchivo
TaskHandle_t Tarea3; // Tarea3 guardar Lecturas1
TaskHandle_t Tarea4; // Tarea4 guardar Lecturas
TaskHandle_t Tarea5; // Tarea5 para guardar pulsos
TaskHandle_t Tarea6; // Tarea6 para guardar pulsos1

//*****
*****

//          INICIALIZACIÓN DE OBJETOS Y VARIABLES

//*****
*****

virtuabotixRTC myRTC(12, 14, 27); // (CLK, DAT, RST) Configuración del RTC

const int sensor = 26; // Pin del sensor de flujo

volatile int pulsos = 0; // Contador de pulsos

volatile int pulsos1 = 0; // Contador de pulsos para grafica

unsigned long tiempoAnteriorLecturas = 0;

unsigned long pulsosAcumulados = 0; // Pulsos acumulados

unsigned long pulsosAcumulados1 = 0; // Pulsos acumulados

float litros = 0; // Litros calculados

float litros1 = 0; // Litros calculados

float metroc = 0; // Metros cúbicos calculados

unsigned int litrosPorHora; // variable del cálculo de litros/hora

```

```

unsigned int litrosPorMinuto;// variable del cálculo de litros/minuto

String respuesta = "";

const int chipSelect = 5; // Pin CS de la SD

char fileName[] = "/pulsos.txt"; // Nombre del archivo para guardar pulsos
char fileName1[] = "/pulsos1.txt"; // Nombre del archivo para guardar pulsos en intervalos
char NameMes[20]= "/0825_acumulado.txt"; // Nombre dinámico de archivos
char NameMes1[20]= "/0825_intervalo.txt"; // Nombre dinámico de archivos

File archivo;//objeto tipo File

void setup() {
    // Configuración inicial de fecha y hora si es necesario

    //myRTC.setDS1302Time(0, 56, 14, 0, 18, 2, 2025);//(0, 58(minutos), 23(hora), 0(segundos),
    31(dia), 12(mes), 2024(años))

    Serial.begin(115200); // Se inicializa al arrancar

    xTaskCreatePinnedToCore(loop0,"Tarea_0",4000,NULL,0,&Tarea0,0); // Core 0
    xTaskCreatePinnedToCore(loop2,"Tarea_2",4000,NULL,0,&Tarea2,0); // Core 0
    xTaskCreatePinnedToCore(loop3,"Tarea_3",5000,NULL,0,&Tarea3,0); // Core 0
    xTaskCreatePinnedToCore(loop4,"Tarea_4",5000,NULL,0,&Tarea4,0); // Core 0
    xTaskCreatePinnedToCore(loop5,"Tarea_5",5000,NULL,0,&Tarea5,0); // Core 0
    xTaskCreatePinnedToCore(loop6,"Tarea_6",5000,NULL,0,&Tarea6,0); // Core 0

    // Configuración del sensor
    pinMode(sensor, INPUT);
    attachInterrupt(digitalPinToInterrupt(sensor), flujo, RISING);
    iniciarSD();//inicializa micro SD
    recuperarPulsos();// recupera pulsos si se ha reiniciado el sistema
    recuperarPulsos1();// recupera pulsos de intervalos si se ha reiniciado el sistema
}

void loop() {
    myRTC.updateTime(); // Actualiza el tiempo del RTC
    printDateTime(); //imprime fecha y hora
    String archivoSeleccionado = reconocerComandos();

```

```

if (!archivoSeleccionado.isEmpty()) {
    leerArchivo(archivoSeleccionado); // Lee el archivo seleccionado
}

}

//*****
*****

//          FUNCIONES

//*****
*****

void flujo() {
    pulsos++; // Incrementar contador de pulsos incrementales
    pulsos1++; // Incrementar contador de pulsos por intervalos
}

//*****
*****

void procesarLecturas() {
    pulsosAcumulados += pulsos; // Sumar pulsos acumulados

    litros = pulsosAcumulados * 1.0 / 444.44; // 444.44; // Calcular litros (444.44 pulsos = 1 litro)

    litrosPorHora = (pulsos * 60 / 7.12); // (Pulsos de frecuencia x 60 min) / 7.12 = factor de
conversion calculado

    litrosPorMinuto = litrosPorHora/60;

    metroc = litros / 1000; // Convertir a metros cúbicos

    pulsos = 0; // Reiniciar el contador de pulsos

// Mostrar los datos en el monitor serial
Serial.println("Litros:"+String(litros, 3));
Serial.println("M3:"+String(metroc,5));
Serial.println("L/min:" + String(litrosPorMinuto));
}

```

```

//*****
*****

void procesarLecturas1() {
    pulsosAcumulados1 += pulsos1; // Sumar pulsos acumulados
    litros1 = pulsosAcumulados1 * 1.0 / 444.44; // Calcular litros (444.44 pulsos = 1 litro)
    pulsos1 = 0; // Reiniciar el contador de pulsos
}

//*****
*****

void guardarPulsosEnSD() {
    static unsigned long tiempoAnterior = 0; // Almacena el último tiempo de ejecución
    const unsigned long intervalo = 5000; // Tiempo de espera: 5 segundo
    if (millis() - tiempoAnterior >= intervalo) {
        tiempoAnterior = millis(); // Actualizar el tiempo de referencia
        archivo = SD.open(fileName, FILE_WRITE);
        if (archivo) {
            archivo.println(pulsosAcumulados); // Guardar el valor de los pulsos
            archivo.flush(); // Asegurarse de que los datos se escriben
            archivo.close(); //Cierra archivo
        } else {
            Serial.println("Error al guardar pulsos en la SD.");
        }
    }
}

//*****
*****

void guardarPulsosEnSD1() {
    static unsigned long tiempoAnterior = 0; // Almacena el último tiempo de ejecución
    const unsigned long intervalo = 5000; // Tiempo de espera: 5 segundo

    if (millis() - tiempoAnterior >= intervalo) {
        tiempoAnterior = millis(); // Actualizar el tiempo de referencia
    }
}

```

```

archivo = SD.open(fileName1, FILE_WRITE);
if (archivo) {
    archivo.println(pulsosAcumulados1); // Guardar el valor de los pulsos
    archivo.flush(); // Asegurarse de que los datos se escriben
    archivo.close(); // Cierra archivo

} else {
    Serial.println("Error al guardar pulsos(1) en la SD.");
}
}
}

//*****
*****

void guardarLecturas() {
    static unsigned long tiempoAnteriorLecturas = 0; // Tiempo anterior específico para esta
función

    const unsigned long intervalo = 7200000 ; // Intervalo de 2 horas

    if (millis() - tiempoAnteriorLecturas >= intervalo) {
        tiempoAnteriorLecturas = millis(); // Actualizar el tiempo anterior

        // Formatear hora, minuto y segundo con dos dígitos
        String hora = String(myRTC.hours);
        String minuto = String(myRTC.minutes);
        String segundo = String(myRTC.seconds);
        String mes = String(myRTC.month);
        String dia = String(myRTC.dayofmonth);

        if (hora.length() == 1) hora = "0" + hora; // Asegura que la hora tenga 2 dígitos
        if (minuto.length() == 1) minuto = "0" + minuto; // Asegura que el minuto tenga 2 dígitos

```

```
if (segundo.length() == 1) segundo = "0" + segundo; // Asegura que el segundo tenga 2
dígitos
```

```
if (dia.length() == 1) dia = "0" + dia; //asegura que el año tenga dos digitos
```

```
if (mes.length() == 1) mes = "0" + mes; //asegura que el mes tenga dos digitos
```

```
archivo = SD.open(NameMes, FILE_APPEND);
```

```
if (archivo) {
```

```
    archivo.print(dia + "/" + mes + ",");
```

```
    archivo.print(hora + ":" + minuto + ":" + segundo + ","); // Formato: 23:00:01
```

```
    archivo.println(String(litros, 3)); // Guardar el valor de los litros con 3 decimales
```

```
    archivo.flush(); // Asegurarse de que los datos se escriben
```

```
    archivo.close(); // Ciera archivo
```

```
} else {
```

```
    Serial.println("!!!!!!!!!!!!!!!!!! Error al guardar litros en la SD.");
```

```
}
```

```
}
```

```
}
```

```
void guardarLecturas1() {
```

```
    static unsigned long tiempoAnteriorLecturas = 0; // Tiempo anterior específico para esta
función
```

```
    const unsigned long intervalo = 3600000 ; // Intervalo de 1 hora
```

```
    if (millis() - tiempoAnteriorLecturas >= intervalo) {
```

```
        tiempoAnteriorLecturas = millis(); // Actualizar el tiempo anterior
```

```
        // Formatear hora, minuto y segundo con dos dígitos
```

```
        String hora = String(myRTC.hours);
```

```
        String minuto = String(myRTC.minutes);
```

```
        String segundo = String(myRTC.seconds);
```

```
        String mes = String(myRTC.month);
```

```
        String dia = String(myRTC.dayofmonth);
```

```

if (hora.length() == 1) hora = "0" + hora; // Asegura que la hora tenga 2 dígitos
if (minuto.length() == 1) minuto = "0" + minuto; // Asegura que el minuto tenga 2 dígitos
if (segundo.length() == 1) segundo = "0" + segundo; // Asegura que el segundo tenga 2
dígitos
if (dia.length() == 1) dia = "0" + dia; //asegura que el año tenga dos digitos
if (mes.length() == 1) mes = "0" + mes; //asegura que el mes tenga dos digitos

archivo = SD.open(NameMes1, FILE_APPEND);
if (archivo) {
    archivo.print(dia + "/" + mes + ",");
    archivo.print(hora + ":" + minuto + ":" + segundo + ","); // Formato: 23:00:01
    archivo.println(String(litros1, 3)); // Guardar el valor de los litros
    archivo.flush(); // Asegurarse de que los datos se escriben
    archivo.close();

    // Reiniciar pulsos y litros
    pulsosAcumulados1 = 0;
    litros1 = 0;

    Serial.println("reiniciando pulsos y litros");
} else {
    Serial.println(".....Error al guardar litros (1) en la SD");
}
}
}

//*****
*****

String reconocerComandos() {
    String command = "";
    if (Serial.available() > 0) {
        command = Serial.readStringUntil('\n'); // Lee el comando hasta el salto de línea
        command.trim(); // Elimina espacios o saltos de línea innecesarios
    }
}

```

```

if (command == "REFRESCAR") { //recibe comando del HMI
    // Comando REFRESCAR: Lista todos los archivos en la raíz de la SD
    File root = SD.open("/");
    while (true) {
        File entry = root.openNextFile();
        if (!entry) break; // No hay más archivos
        Serial.print("FL: "); // Etiqueta de archivo
        Serial.println(entry.name());
        entry.close();
    }
    root.close(); // Cierra el directorio raíz
}
else if (command.startsWith("/")) {
    Serial.print("Archivo seleccionado: ");
    Serial.println(command);
    return command;
}
else if (command == "Id"){
    Serial.println("&Franklin Javier Escobar Cacuango");
}
}
return ""; // Si no se recibe ningún comando válido, retorna cadena vacía
}

//*****
*****

void leerArchivo(String nombreArchivo) {
    if (nombreArchivo.startsWith("/")) {
        if (SD.exists(nombreArchivo)) { //Identifica nombre de archivo
            File archivo = SD.open(nombreArchivo, FILE_READ);
            if (archivo) {
                Serial.println("Contenido del archivo " + nombreArchivo + ":");
            }
        }
    }
}

```

```

String linea = "-"; // Inicia la línea con el identificador

while (archivo.available()) {
    char c = archivo.read(); // Leer un carácter del archivo
    if (c == '\n') { // Detectar fin de línea
        linea += '\n'; // Añadir salto de línea al final
        Serial.print(linea); // Usar Serial.print para enviar la línea
        linea = "-"; // Reiniciar para la siguiente línea con el identificador
    } else {
        linea += c; // Acumular caracteres en la línea
    }
}

// Si queda algo en la variable después del bucle, enviarlo
if (linea.length() > 1) { // Si la línea contiene algo más que solo el identificador
    Serial.print(linea); // Enviar la última línea acumulada
}

archivo.close();
} else {
    Serial.println("Error al abrir el archivo.");
}
} else {
    Serial.println("El archivo no existe.");
}
}
}

//*****
*****

void printDateTime() {
    static unsigned long tiempoAnterior = 0; // Almacena el último tiempo de ejecución
    const unsigned long intervalo = 1000; // Tiempo de espera: 1 segundo
    if (millis() - tiempoAnterior >= intervalo) {

```

```

        tiempoAnterior = millis(); // Actualizar el tiempo de referencia
// Imprime la fecha y hora actual
Serial.print("          ");
Serial.print(myRTC.dayofmonth);
Serial.print("/");
Serial.print(myRTC.month);
Serial.print("/");
Serial.print(myRTC.year);
Serial.print(" ");
Serial.print(myRTC.hours);
Serial.print(":");
Serial.print(myRTC.minutes);
Serial.print(":");
Serial.println(myRTC.seconds);

    }
}

//*****
//          TAREAS EN SEGUNDO NÚCLEO
//*****

void loop0(void *parameter){ // Tarea0 procesar lecturas en segundo nucleo
while(1==1){
    procesarLecturas();
    vTaskDelay(1000 / portTICK_PERIOD_MS); // Pausa de 100 ms (ajustable)
    // Serial.println("procesando lecturas en:" + String(xPortGetCoreID())+"núcleo");
}
}

void loop2(void *parameter){ // Tarea0 procesar lecturas1 en segundo nucleo
while(1==1){
    procesarLecturas1();
}
}

```

```
vTaskDelay(1000 / portTICK_PERIOD_MS); // Pausa de 100 ms (ajustable)

// Serial.println("procesando lecturas 11111111111111 en:" +
String(xPortGetCoreID())+"núcleo");

}

}
```

```
void loop3(void *parameter){ // Tarea3 guardar lecturas1 en segundo nucleo

while(1==1){

guardarLecturas1();

vTaskDelay(1000 / portTICK_PERIOD_MS); // Pausa de 100 ms (ajustable)

// Serial.println("procesando lecturas 11111111111111 en:" +
String(xPortGetCoreID())+"núcleo");

}

}
```

```
void loop4(void *parameter){ // Tarea4 guardar lecturas en segundo nucleo

while(1==1){

guardarLecturas();

vTaskDelay(1000 / portTICK_PERIOD_MS); // Pausa de 100 ms (ajustable)

// Serial.println("procesando lecturas 11111111111111 en:" +
String(xPortGetCoreID())+"núcleo");

}

}
```

```
void loop5(void *parameter){ // Tarea5 guardar pulsos

while(1==1){

guardarPulsosEnSD(); //almacena pulsos acumulados

vTaskDelay(1000 / portTICK_PERIOD_MS); // Pausa de 100 ms (ajustable)

// Serial.println("procesando lecturas 11111111111111 en:" +
String(xPortGetCoreID())+"núcleo");

}

}
```

```
void loop6(void *parameter){ // Tarea6 guardar pulsos1
```

```

while(1==1){
    guardarPulsosEnSD1();
    vTaskDelay(1000 / portTICK_PERIOD_MS); // Pausa de 100 ms (ajustable)
    // Serial.println("procesando lecturas 111111111111 en:" +
String(xPortGetCoreID()+"núcleo");
}
}

//*****
*****

//          FUNCIONES PARA VOID SETUP

//*****
*****

void recuperarPulsos(){
    // Leer los últimos pulsos acumulados desde la SD
    archivo = SD.open(fileName, FILE_READ);
    if (archivo) {
        String ultima_linea;
        while (archivo.available()) {
            ultima_linea = archivo.readStringUntil('\n'); // Leer hasta la última línea
        }
        archivo.close();

        // Convertir la última línea a un número entero
        if (ultima_linea.length() > 0) {
            pulsosAcumulados = ultima_linea.toInt(); // Pasa a entero
            Serial.print("Pulsos recuperados: ");
            Serial.println(pulsosAcumulados);
        } else {
            pulsosAcumulados = 0;
            Serial.println("No se encontraron datos previos. Iniciando en 0.");
        }
    } else {
        pulsosAcumulados = 0;
    }
}

```

```

    Serial.println("Archivo no encontrado. Iniciando en 0.");
}
}

void recuperarPulsos1(){
    // Leer los últimos pulsos acumulados desde la SD
    archivo = SD.open(fileName1, FILE_READ);
    if (archivo) {
        String ultima_linea;
        while (archivo.available()) {
            ultima_linea = archivo.readStringUntil('\n'); // Leer hasta la última línea
        }
        archivo.close();

        // Convertir la última línea a un número entero
        if (ultima_linea.length() > 0) {
            pulsosAcumulados1 = ultima_linea.toInt(); // Pasa a entero
            Serial.print("Pulsos recuperados: ");
            Serial.println(pulsosAcumulados1);
        } else {
            pulsosAcumulados1 = 0;
            Serial.println("No se encontraron datos previos. Iniciando en 0.");
        }
    } else {
        pulsosAcumulados1 = 0;
        Serial.println("Archivo no encontrado. Iniciando en 0.");
    }
}

void iniciarSD(){
    // Inicialización de la tarjeta SD
    if (!SD.begin(chipSelect)) {

```

```
Serial.println("Error inicializando la SD");  
return;  
}  
Serial.println("SD inicializada correctamente");  
}
```

### **ANEXO C. Código del HMI.**

```
using LiveCharts;  
using LiveCharts.Defaults;  
using LiveCharts.Definitions.Series;  
using LiveCharts.Wpf;  
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Globalization;  
using System.IO.Ports;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows;  
using System.Windows.Forms;  
using System.Windows.Media;  
using System.Xml;  
using static System.Windows.Forms.VisualStyles.VisualStyleElement;
```

```
namespace medidor_de_agua_2  
{
```

```

public partial class Form1 : Form
{
    private SeriesCollection seriesCollection;

    private List<string> tiempo = new List<string>(); // Lista para el tiempo
    private List<double> litros = new List<double>(); // Lista para litros

    public Form1()
    {
        InitializeComponent();

        label2.Text = "m³";

        //label1.Font = new Font(label1.Font.FontFamily, 20); // Tamaño de fuente: 20

        solidGauge1.ForeGround = new
System.Windows.Media.SolidColorBrush(System.Windows.Media.Colors.MediumPurple);

        solidGauge1.From = 0; //color de nivel mínimo
        solidGauge1.To = 30; //color de nivel máximo

        solidGauge1.FromColor = System.Windows.Media.Color.FromRgb(0, 255, 0); // Verde
        solidGauge1.ToColor = System.Windows.Media.Color.FromRgb(255, 0, 0); // Rojo
        alphaGradientPanel1.BackColor = System.Drawing.Color.Red; // Asigna un color sólido

        // Inicializar el gráfico

        InicializarGrafico();

        timer1.Tick += Timer1_Tick; // Vincular el evento Tick
    }

    private void Timer1_Tick(object sender, EventArgs e)
    {
        label8.Text = DateTime.Now.ToString("HH:mm:ss"); // Mostrar la hora en el Label
    }

    private void InicializarGrafico()
    {
        seriesCollection = new SeriesCollection

```

```

{
new LineSeries
{
    Values = new ChartValues<ObservablePoint>(),
    PointGeometrySize = 10,
    PointForeground = System.Windows.Media.Brushes.DarkRed, // Color de los puntos
    Title = string.Empty, // Elimina el título de la serie
    ToolTip = null, // Desactiva el tooltip predeterminado
    LabelPoint = point => $"Litros: {point.Y:F3}" // Formato personalizado
}
};

```

```

cartesianChart1.Series = seriesCollection;

```

```

cartesianChart1.AxisX.Add(new Axis

```

```

{
    Title = "Tiempo",
    LabelFormatter = value =>
    {
        // Convertir segundos de vuelta a fecha y hora para mostrar
        DateTime fechaHora = new DateTime(2024, 1, 1).AddSeconds(value);
        return fechaHora.ToString("dd/MM HH:mm:ss");
    }, //coma

```

```

    // MinValue = 0, // Valor mínimo inicial

```

```

    // MaxValue = 20, // Valor máximo inicial (ajústalo según tu caso)

```

```

});

```

```

cartesianChart1.AxisY.Add(new Axis

```

```

{

```

```
Title = "Consumo (litros)",
LabelFormatter = value => value.ToString("0.000"),

});

// Habilitar zoom y scroll
cartesianChart1.Zoom = ZoomingOptions.X; // Zoom en el eje X
cartesianChart1.Pan = PanningOptions.X; // Permitir desplazamiento horizontal

}

private void Form1_Load(object sender, EventArgs e)
{

}

private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    if(serialPort1.IsOpen)
    {
        serialPort1.Close();
    }
}

private void button1_Click(object sender, EventArgs e)
{
    try
    {
```

```

        serialPort1.Open();
        serialPort1.WriteLine("REFRESCAR");

    }
    catch(Exception ex)
    {
        System.Windows.MessageBox.Show(ex.Message);
    }
}

private void serialPort1_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    while(serialPort1.IsOpen && serialPort1.BytesToRead > 0)
    {

        string serialData = serialPort1.ReadLine(); // Lee todos los datos disponibles

        //Elementos graficos del HMI

        if (serialData.StartsWith("Litros:"))
        {
            string Litros = serialData.Replace("Litros:", " ").Trim();
            label_Litros.Invoke((MethodInvoker)() =>
            {
                label_Litros.Text = Litros;
            }));
        }
    }
}

```

```

else if (serialData.StartsWith("M3:"))
{
    string Metros = serialData.Replace("M3:", " ").Trim();
    label_Metros.Invoke((MethodInvoker)() =>
    {
        label_Metros.Text = Metros;
    }));
}
else if(serialData.StartsWith("L/min:"))
{
    string Lmin = serialData.Replace("L/min:", " ").Trim();
    int valor = Convert.ToInt32(Lmin);

    solidGauge1.Invoke((MethodInvoker)() =>
    {
        solidGauge1.Value = valor;
    }));
}
else if(serialData.StartsWith("FL:"))//para extraer nombres de archivos
{
    // Extrae el nombre del archivo eliminando "FL:" al inicio
    string nombreArchivo = serialData.Substring(3).Trim();
    // Actualiza el ListBox en el hilo principal
    this.Invoke(new Action() =>
    {
        // Verifica si el archivo ya está en la lista para evitar duplicados

```

```

        if (!listBoxArchivos.Items.Contains(nombreArchivo))
        {
            listBoxArchivos.Items.Add(nombreArchivo); // Agrega el archivo al ListBox
        }
    });
}
else if (serialData.StartsWith("-"))
{
    // Actualiza el TextBox en el hilo principal
    this.Invoke(new Action(() =>
    {
        string dato = serialData.Substring(1);
        Console.WriteLine($"Dato recibido: {dato}");
        ProcesarDato(dato); // Procesa y agrega el dato al gráfico
        textBoxContenido.AppendText(serialData.Substring(1) +
Environment.NewLine);

    }));
}
else if (serialData.StartsWith("&"))
{
    // Actualiza el TextBox en el hilo principal
    this.Invoke(new Action(() =>
    {
        string id = serialData.Substring(1);
        Console.WriteLine($"Dato recibido: {id}");
        IdLabel.Text = id;

    }));
}

```

```

    }

}

private void ProcesarDato(string dato) //procesar datos de fecha, hora y litros
{
    var partes = dato.Split(',');

    if (partes.Length == 3)
    {
        string fechaStr = partes[0];
        string tiempoStr = partes[1];

        if (double.TryParse(partes[2], NumberStyles.Float, CultureInfo.InvariantCulture, out
double litrosValue)) //conserva los decimales para el arreglo de litros
        {
            // Convertir la fecha (día) y la hora a DateTime

            string fechaHoraStr = fechaStr + " " + tiempoStr; // Se concatena la fecha y la hora
con un espacio

            // Asegurarse de que la cadena tiene el formato correcto

            try
            {
                DateTime fechaHora = DateTime.ParseExact(fechaHoraStr, "dd/MM
HH:mm:ss", CultureInfo.InvariantCulture);

                // Convertir la fecha y hora a segundos desde el 01/01/2000

                DateTime referencia = new DateTime(2000, 1, 1);
                double tiempoSegundos = (fechaHora - referencia).TotalSeconds;

                // Añadir punto al gráfico

                var serie = (LineSeries)cartesianChart1.Series[0];
                serie.Values.Add(new ObservablePoint(tiempoSegundos, litrosValue));
            }
        }
    }
}

```

```

    }
    catch (FormatException ex)
    {
        // Si no se puede parsear la fecha, mostrar un mensaje de error
        Console.WriteLine("Error al analizar la fecha: " + ex.Message);
    }
}
}
}

private void Graficar_Click(object sender, EventArgs e)
{

    // Verifica si hay un archivo seleccionado
    if (listBoxArchivos.SelectedItem != null)
    {
        string archivoSeleccionado = listBoxArchivos.SelectedItem.ToString();

        // Envía el nombre del archivo al ESP32
        serialPort1.WriteLine(archivoSeleccionado);

        // Limpia el TextBox antes de mostrar el nuevo contenido seleccionado
        textBoxContenido.Clear();

        // Limpia el gráfico antes de graficar nuevos datos
        var serie = (LineSeries)cartesianChart1.Series[0];
        serie.Values.Clear(); // Limpia los puntos anteriores

        // reinicia las listas de tiempo y litros
        tiempo.Clear();
        litros.Clear();
    }
}

```

```
else
{
    System.Windows.MessageBox.Show("Por favor, selecciona un archivo de la lista.");
}

}

private void Actualizar_Click(object sender, EventArgs e)
{
    while (serialPort1.IsOpen && serialPort1.BytesToRead > 0)
    {
        serialPort1.WriteLine("REFRESCAR");
    }
}

private void Conectar_Click(object sender, EventArgs e)
{
    try
    {
        serialPort1.Open();
        serialPort1.WriteLine("REFRESCAR");
        serialPort1.WriteLine("Id");
        serialPort1.WriteLine("REFRESCAR");
    }
    catch (Exception ex)
    {
        System.Windows.MessageBox.Show(ex.Message);
    }
}
```

```

}

private void timer1_Tick(object sender, EventArgs e)
{
    label8.Text = DateTime.Now.ToString("HH:mm:ss"); // Actualiza con la hora actual
}

private void C_Horario_Click(object sender, EventArgs e)
{
    while (serialPort1.IsOpen && serialPort1.BytesToRead > 0)
    {
        serialPort1.WriteLine("/0825_intervalo.txt");
    }
}

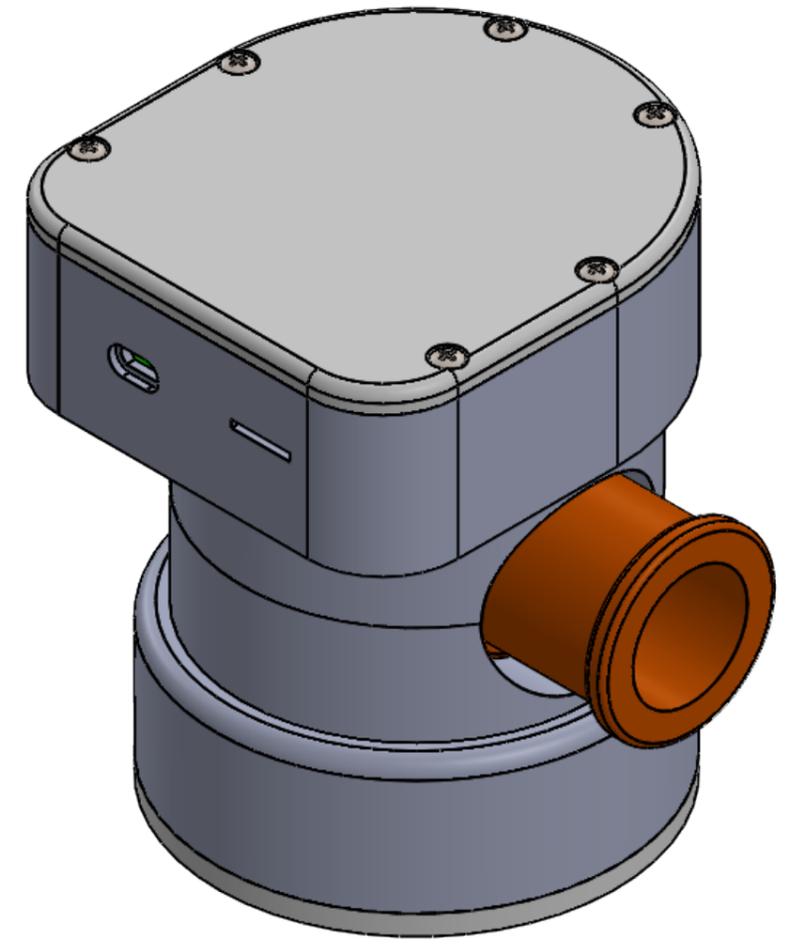
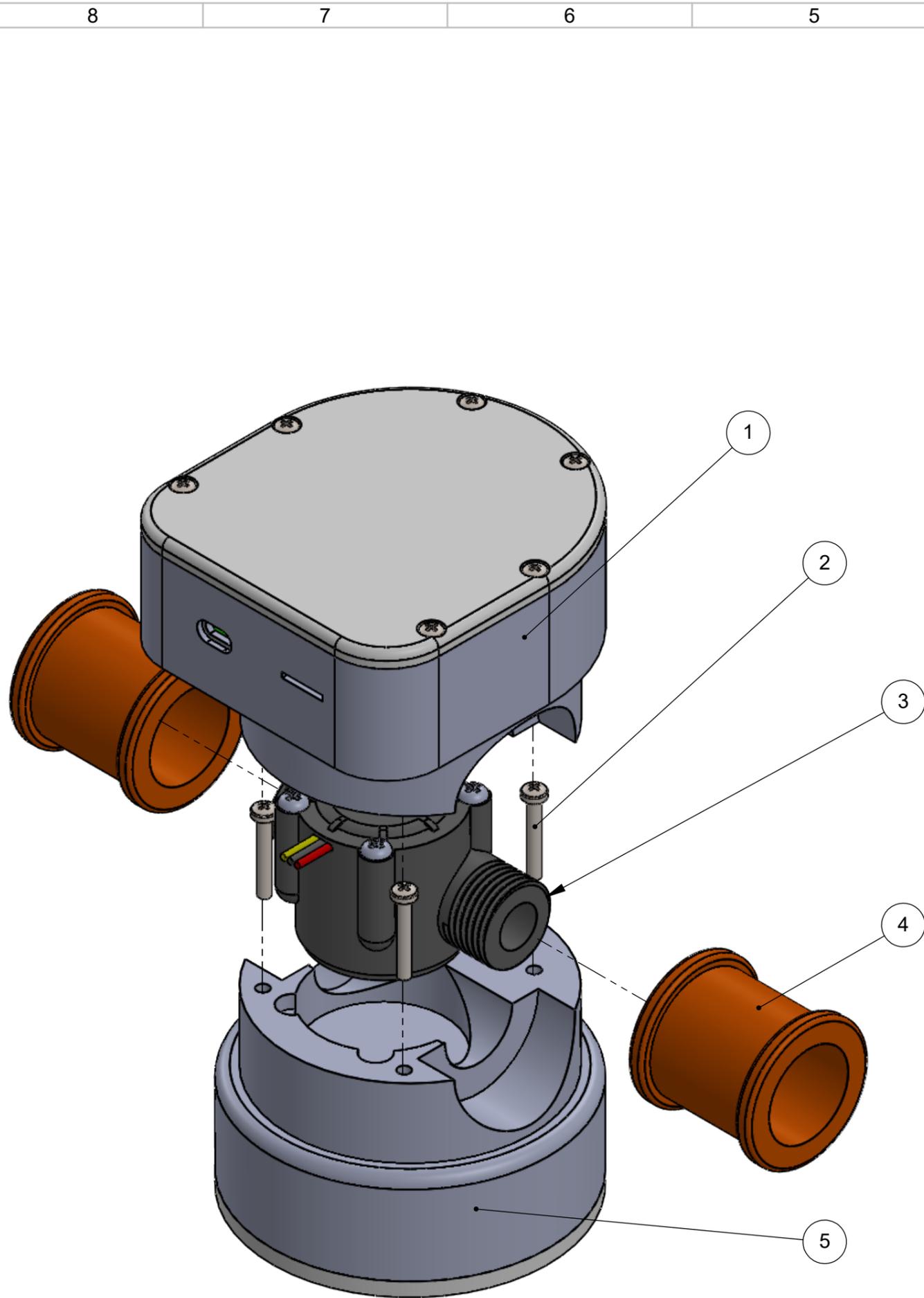
private void C_Acumulado_Click(object sender, EventArgs e)
{
    while (serialPort1.IsOpen && serialPort1.BytesToRead > 0)
    {
        serialPort1.WriteLine("/0825_acumulado.txt");
    }
}

}

}
}

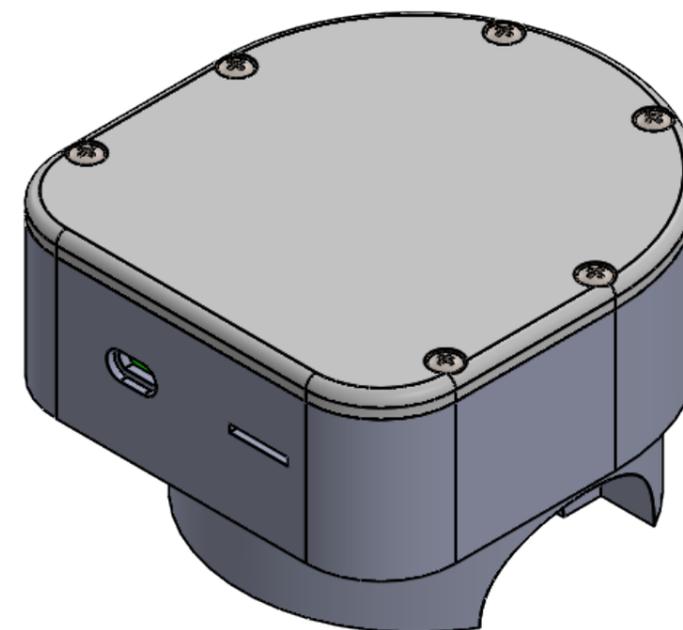
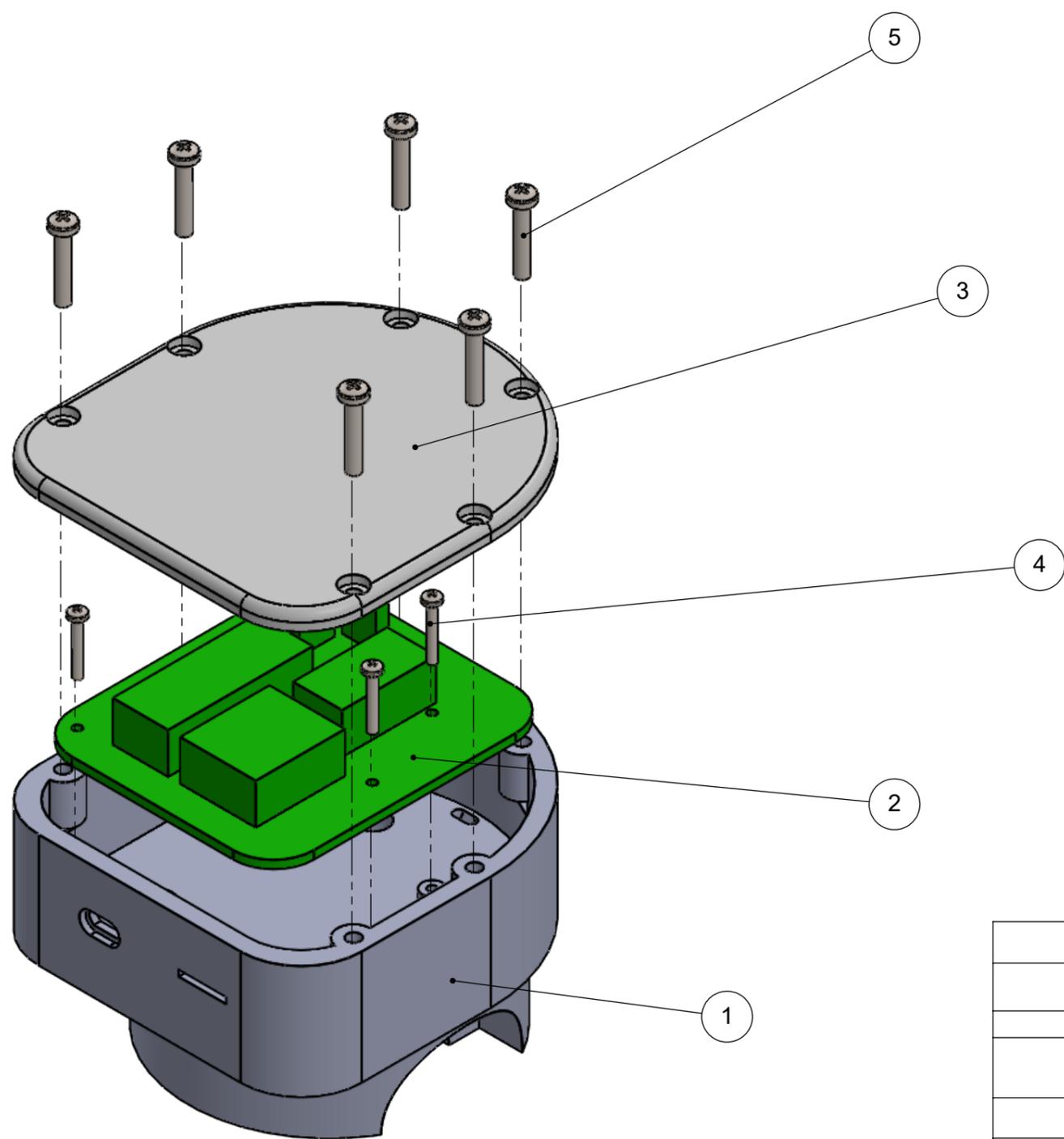
```

**ANEXO D. Planos.**



5	ENSAMBLAJE INFERIOR	----	C1	1	TMD-EI-002
4	UNION DE MEDIA	PVC	B1	2	----
3	SENSOR YF-S201	----	B1	1	----
2	PERNOS M3 x 20 mm	ALUMINIO	B1	4	----
1	ENSAMBLAJE SUPERIOR	----	A1	1	TMD-ES-001
N.º DE ELEMENTO	DENOMINACIÓN	MATERIAL	ZONA	CANTIDAD	CÓDIGO

	PROYECTO: MEDIDOR DIGITAL DE AGUA POTABLE				ESCALA 1:1
	CONJUNTO: ENSAMBLAJE GENERAL				
	CÓDIGO: TMD-EG-000				
	PESO:	297,6 g	DISEÑO:	ESCOBAR J.	5/1/2025
LARGO:	150	DIBUJÓ:	ESCOBAR J.	8/1/2025	
ANCHO:	85	REVISÓ:	MEGÍA C.	8/1/2025	
ALTO:	105.2	APROBÓ:	MEGÍA C.	8/1/2025	
<b>UTN</b>	<b>CIME</b>				



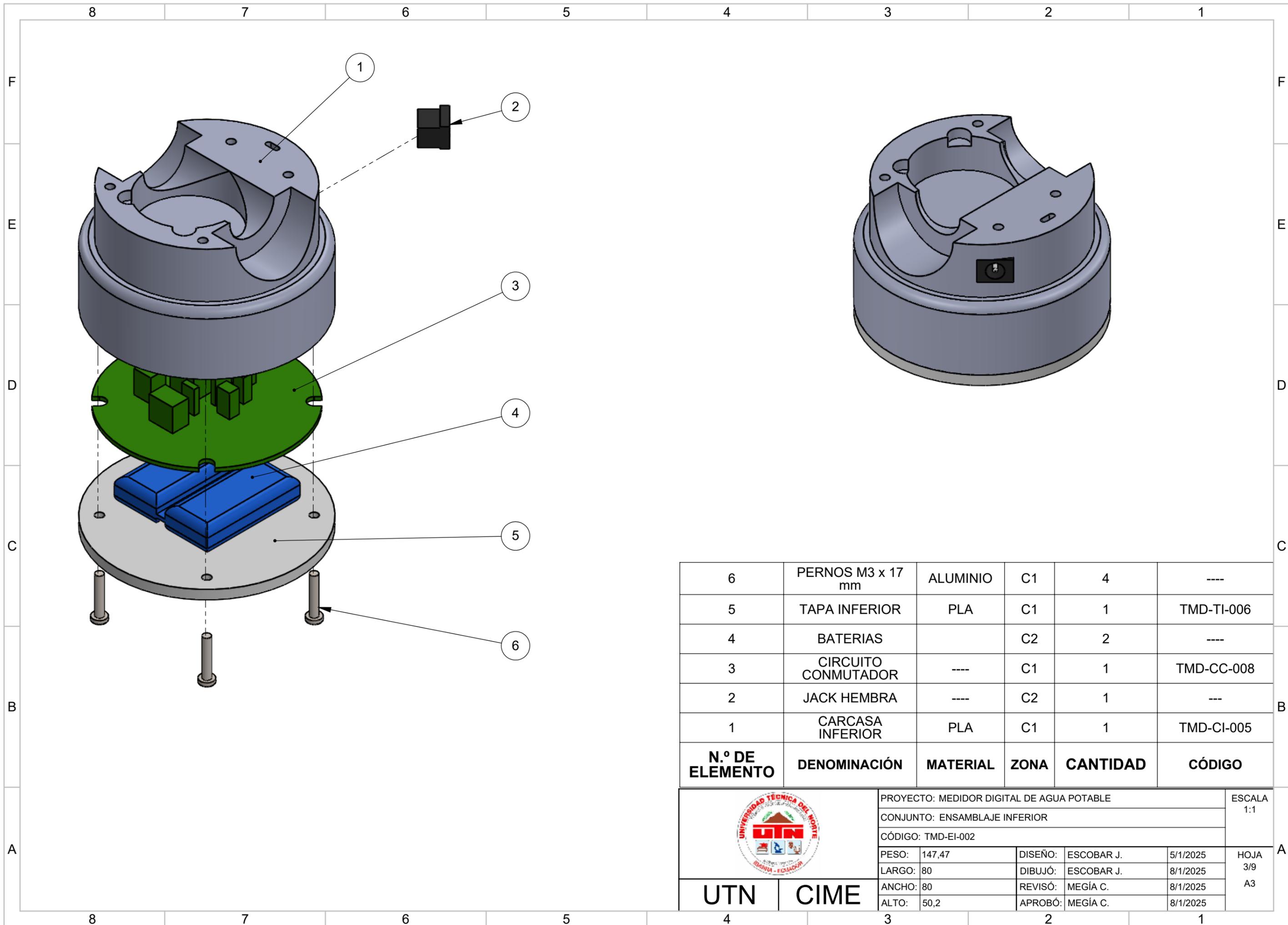
5	PERNOS M3 X 15 mm	ALUMINIO	A1	6	----
4	PERNOS M2 x 10 mm	ALUMINIO	A2	4	----
3	TAPA SUPERIOR	PLA	A1	1	TMD-TS-003
2	CIRCUITO PROCESAMIENTO	----	A2	1	TMD-CP-007
1	CARCASA SUPERIOR	PLA	A1	1	TMD-CS-004

N.º DE ELEMENTO	DENOMINACIÓN	MATERIAL	ZONA	CANTIDAD	CÓDIGO
-----------------	--------------	----------	------	----------	--------



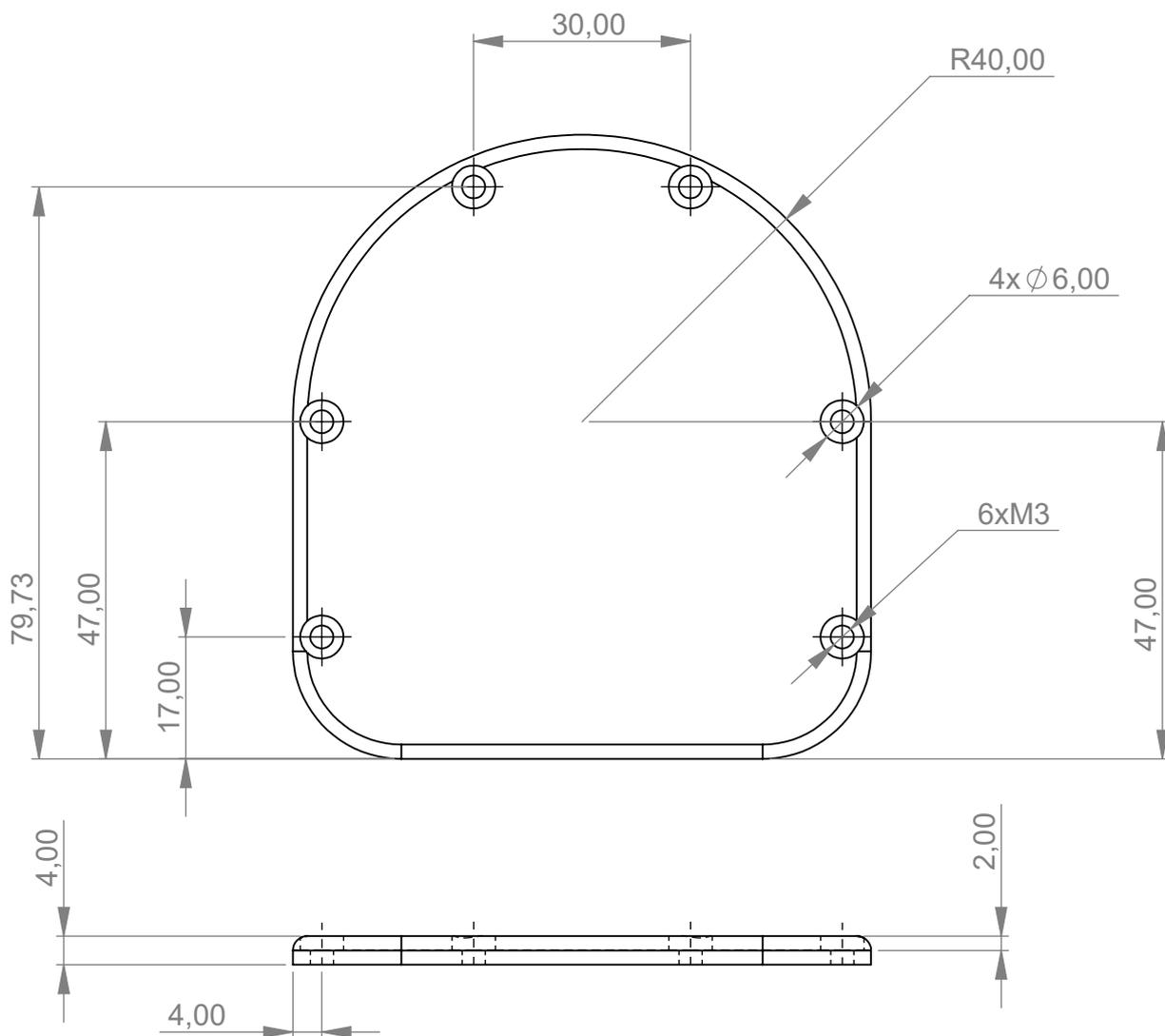
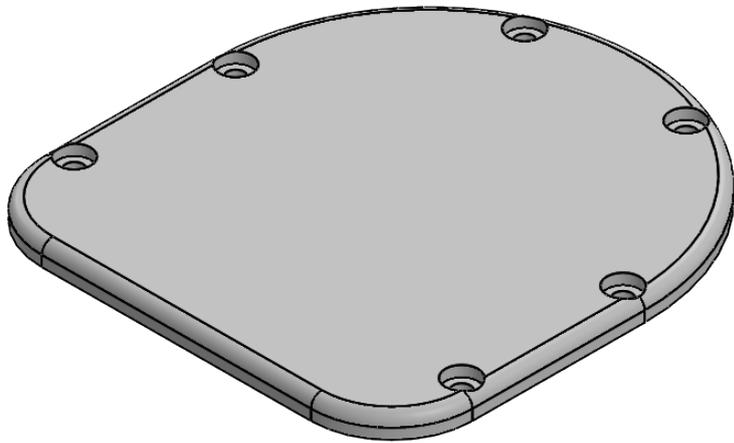
**UTN CIME**

PROYECTO: MEDIDOR DIGITAL DE AGUA POTABLE				ESCALA 1:1	
CONJUNTO: ENSAMBLAJE SUPERIOR					
CÓDIGO: TMD-ES-001					
PESO:	95,13 g	DISEÑO:	ESCOBAR J.	5/1/2025	HOJA 2/9 A3
LARGO:	80	DIBUJÓ:	ESCOBAR J.	8/1/2025	
ANCHO:	85	REVISÓ:	MEGÍA C.	8/1/2025	
ALTO:	55	APROBÓ:	MEGÍA C.	8/1/2025	



6	PERNOS M3 x 17 mm	ALUMINIO	C1	4	----
5	TAPA INFERIOR	PLA	C1	1	TMD-TI-006
4	BATERIAS		C2	2	----
3	CIRCUITO CONMUTADOR	----	C1	1	TMD-CC-008
2	JACK HEMBRA	----	C2	1	---
1	CARCASA INFERIOR	PLA	C1	1	TMD-CI-005
N.º DE ELEMENTO	DENOMINACIÓN	MATERIAL	ZONA	CANTIDAD	CÓDIGO

	PROYECTO: MEDIDOR DIGITAL DE AGUA POTABLE				ESCALA 1:1
	CONJUNTO: ENSAMBLAJE INFERIOR				
	CÓDIGO: TMD-EI-002				
	PESO:	147,47	DISEÑO:	ESCOBAR J.	5/1/2025
	LARGO:	80	DIBUJÓ:	ESCOBAR J.	8/1/2025
ANCHO:	80	REVISÓ:	MEGÍA C.	8/1/2025	
ALTO:	50,2	APROBÓ:	MEGÍA C.	8/1/2025	
<b>UTN</b>		<b>CIME</b>			HOJA 3/9 A3



PROYECTO: MEDIDOR DIGITAL DE AGUA POTABLE

ESCALA  
1:1



PARTE: TAPA SUPERIOR

CÓDIGO: TMD-TS-003

TOLERANCIA:  $\pm 0,2$

MATERIAL: PLA      DISEÑO: ESCOBAR J.      8/1/2025

TRATAMIENTO: NINGUNO      DIBUJO: ESCOBAR J.      11/1/2025

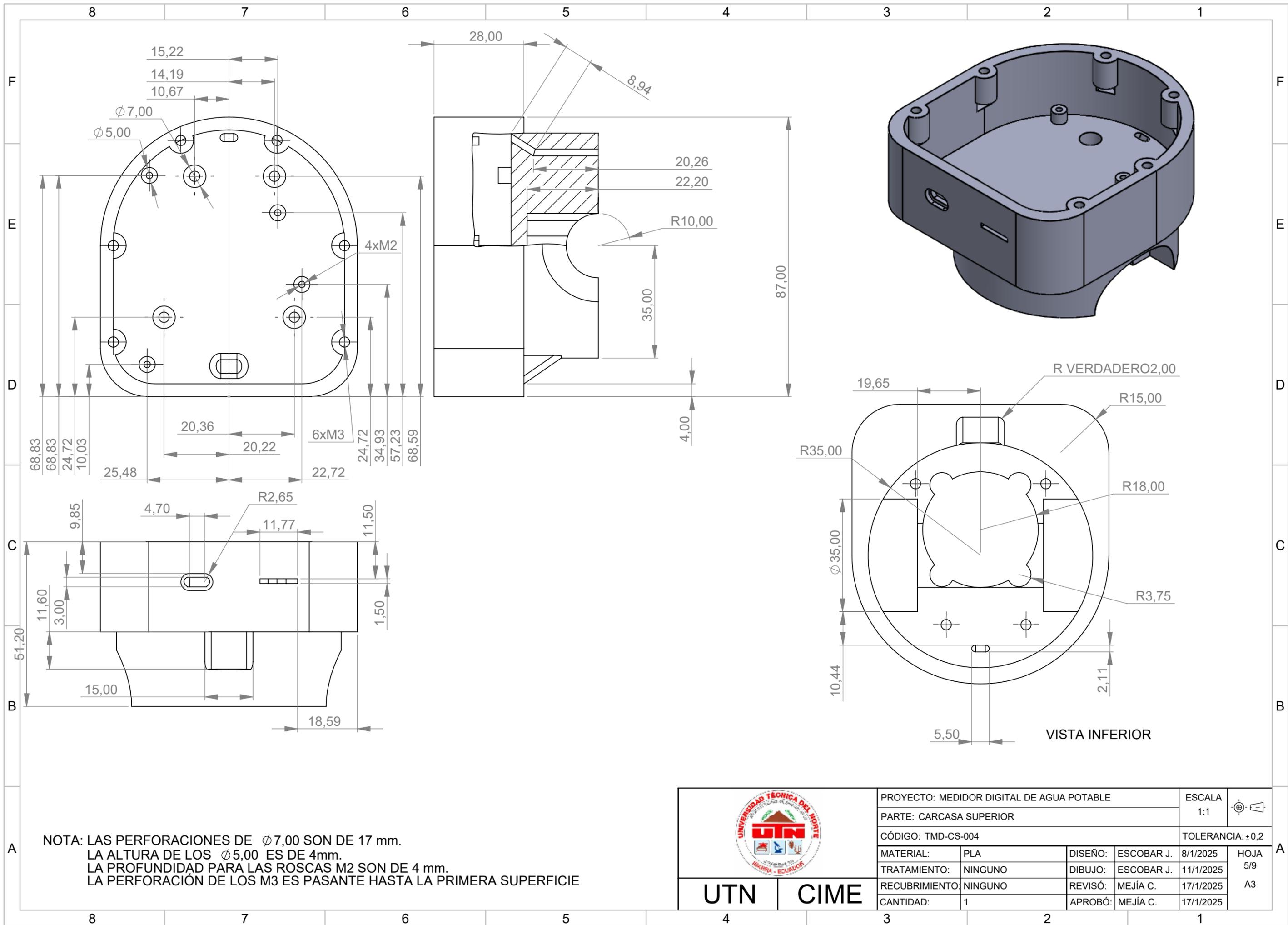
RECUBRIMIENTO: NINGUNO      REVISÓ: MEJÍA C.      17/1/2025

CANTIDAD: 1      APROBÓ: MEJÍA C.      17/1/2025

UTN

CIME

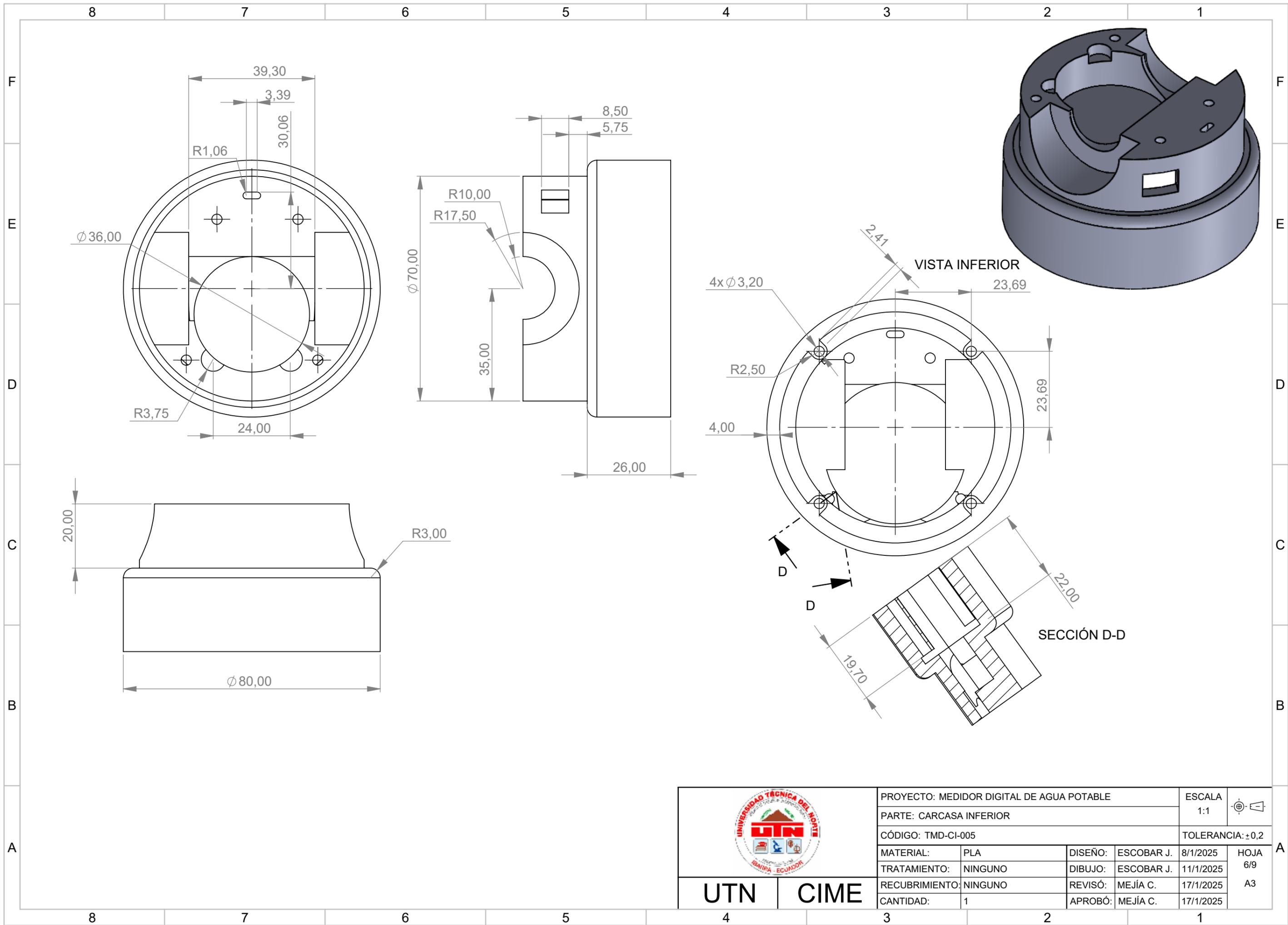
HOJA  
4/9  
A4



NOTA: LAS PERFORACIONES DE  $\phi 7,00$  SON DE 17 mm.  
 LA ALTURA DE LOS  $\phi 5,00$  ES DE 4mm.  
 LA PROFUNDIDAD PARA LAS ROSCAS M2 SON DE 4 mm.  
 LA PERFORACIÓN DE LOS M3 ES PASANTE HASTA LA PRIMERA SUPERFICIE

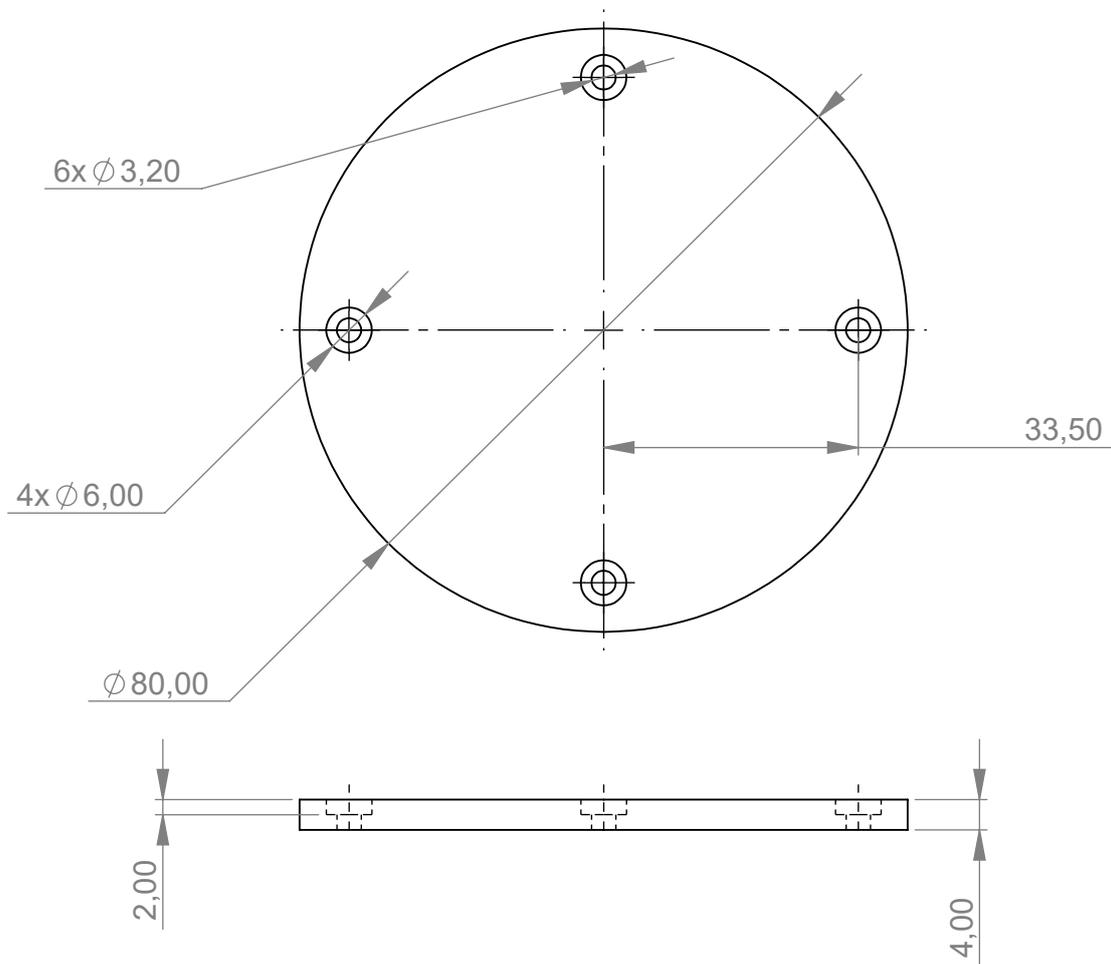
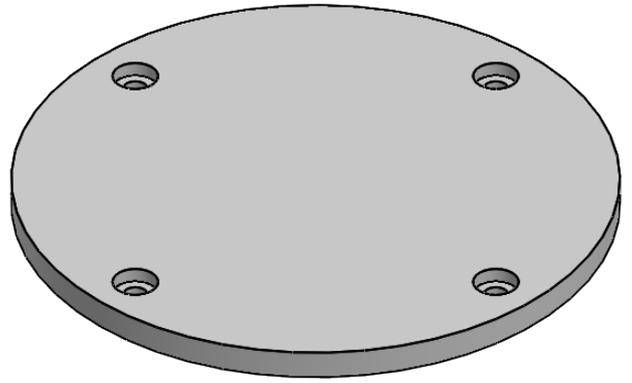
	PROYECTO: MEDIDOR DIGITAL DE AGUA POTABLE				ESCALA	
	PARTE: CARCASA SUPERIOR				1:1	
	CÓDIGO: TMD-CS-004				TOLERANCIA: $\pm 0,2$	
	MATERIAL:	PLA	DISEÑO:	ESCOBAR J.	8/1/2025	HOJA 5/9 A3
	TRATAMIENTO:	NINGUNO	DIBUJO:	ESCOBAR J.	11/1/2025	
	RECUBRIMIENTO:	NINGUNO	REVISÓ:	MEJÍA C.	17/1/2025	
CANTIDAD:	1	APROBÓ:	MEJÍA C.	17/1/2025		

UTN CIME



	PROYECTO: MEDIDOR DIGITAL DE AGUA POTABLE				ESCALA	
	PARTE: CARCASA INFERIOR				1:1	
	CÓDIGO: TMD-CI-005				TOLERANCIA: ±0,2	
	MATERIAL:	PLA	DISEÑO:	ESCOBAR J.	8/1/2025	HOJA
	TRATAMIENTO:	NINGUNO	DIBUJO:	ESCOBAR J.	11/1/2025	6/9
RECUBRIMIENTO:	NINGUNO	REVISÓ:	MEJÍA C.	17/1/2025	A3	
CANTIDAD:	1	APROBÓ:	MEJÍA C.	17/1/2025		

**UTN** | **CIME**



PROYECTO: MEDIDOR DIGITAL DE AGUA POTABLE

ESCALA  
1:1



PARTE: TAPA INFERIOR

CÓDIGO: TMD-TI-006

TOLERANCIA:  $\pm 0,2$

MATERIAL: PLA DISEÑO: ESCOBAR J. 8/1/2025

TRATAMIENTO: NINGUNO DIBUJO: ESCOBAR J. 11/1/2025

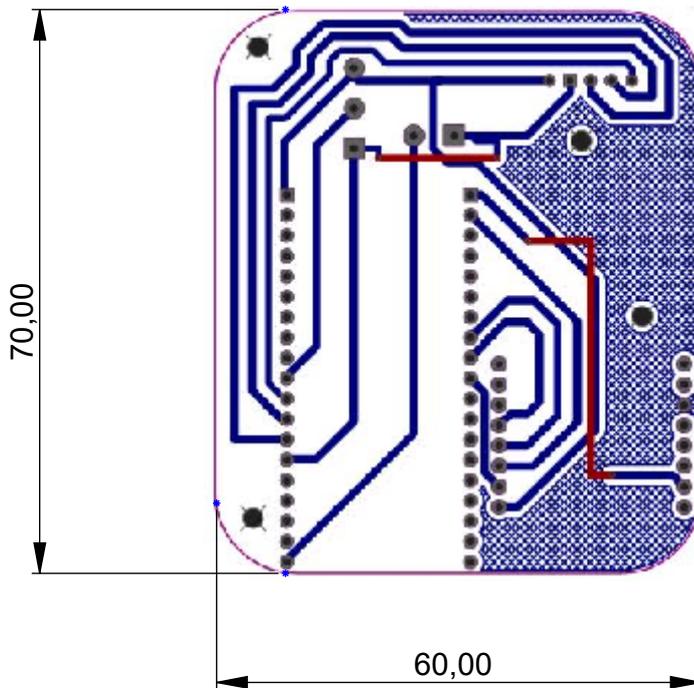
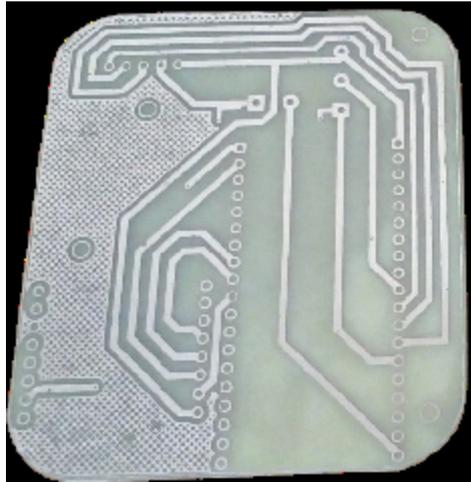
RECUBRIMIENTO: NINGUNO REVISÓ: MEJÍA C. 17/1/2025

CANTIDAD: 1 APROBÓ: MEJÍA C. 17/1/2025

HOJA  
7/9  
A4

UTN

CIME



PROYECTO: MEDIDOR DIGITAL DE AGUA POTABLE

ESCALA  
1:1



PARTE: CIRCUITO PROCESAMIENTO

CÓDIGO: TMD-CP-007

TOLERANCIA:  $\pm 0,1$

MATERIAL: BAQUELITA COBRE DISEÑO: ESCOBAR J. 8/1/2025

HOJA  
8/9

TRATAMIENTO: PLANCHADO, ÁCIDO DIBUJO: ESCOBAR J. 11/1/2025

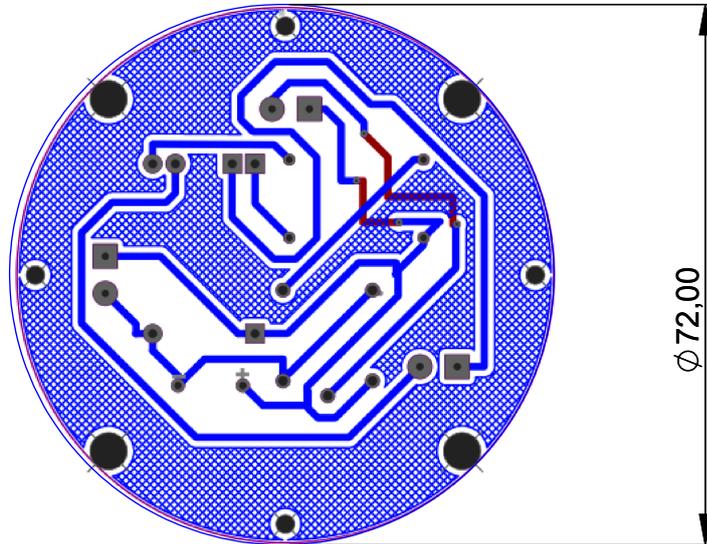
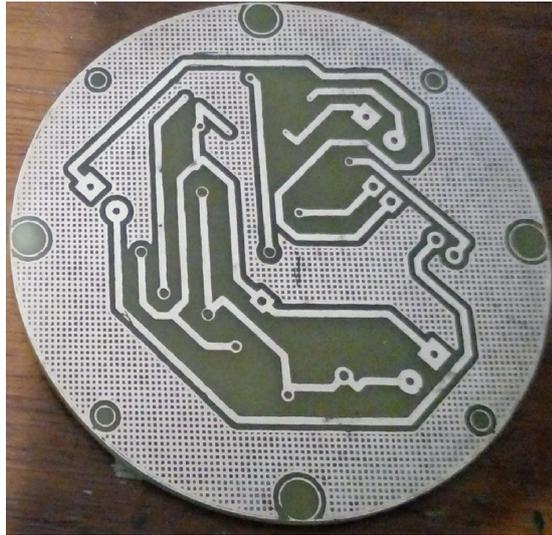
RECUBRIMIENTO: NINGUNO REVISÓ: MEJÍA C. 17/1/2025

A4

CANTIDAD: 1 APROBÓ: MEJÍA C. 17/1/2025

UTN

CIME

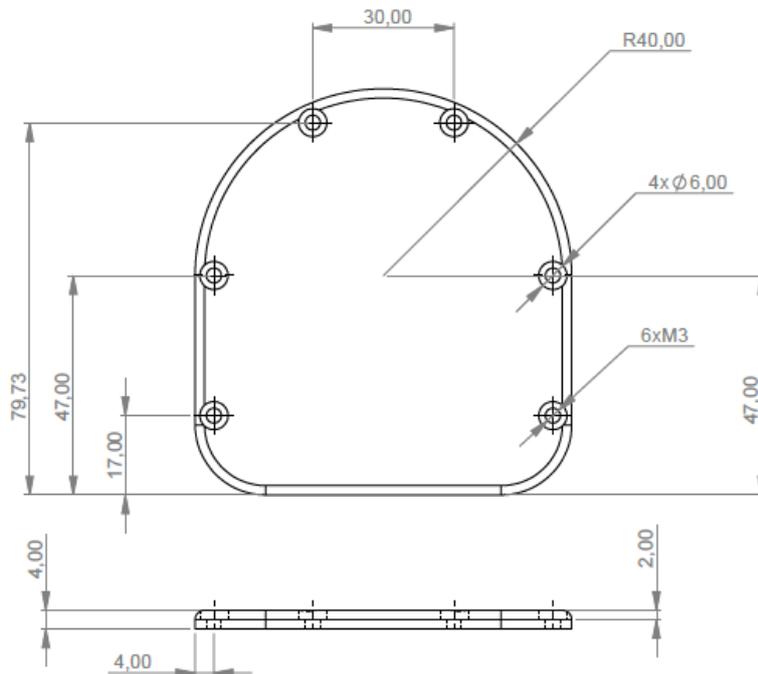
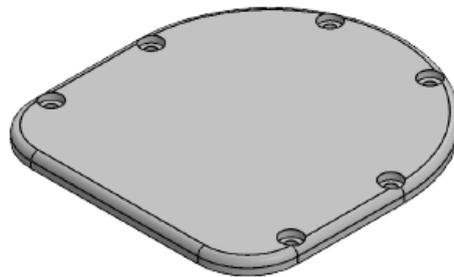


	PROYECTO: MEDIDOR DIGITAL DE AGUA POTABLE				ESCALA 1:1	
	PARTE: CIRCUITO CONMUTADOR					
	CÓDIGO: TMD-CC-008				TOLERANCIA: ±0,1	
	MATERIAL:	BAQUELITA COBRE	DISEÑO:	ESCOBAR J.	9/1/2025	HOJA 9/9
	TRATAMIENTO:	PLANCHADO, ÁCIDO	DIBUJO:	ESCOBAR J.	10/1/2025	
RECUBRIMIENTO:	NINGUNO	REVISÓ:	MEJÍA C.	17/1/2025	A4	
CANTIDAD:	1	APROBÓ:	MEJÍA C.	17/1/2025		
UTN	CIME					

**ANEXO E.** Hojas de procesos.

ANEXO E. Hojas de proceso de piezas

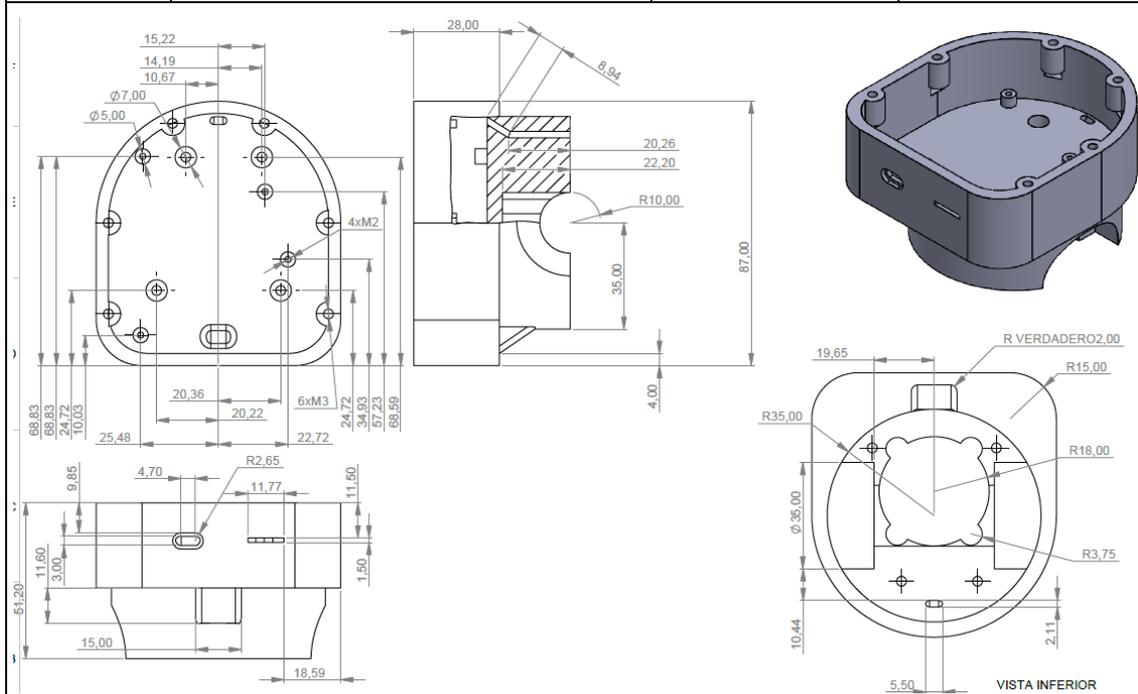
Hoja de Procesos Tapa Superior					
<b>Proyecto</b>	Medido digital de agua potable		<b>Máquina</b>	Diy Adventurer	<b>Fecha:</b> 30/1/2025
<b>Pieza</b>	Tapa Superior	<b>Plano Nro.</b>	TMD-TS-003	<b>Hoja</b>	
<b>Material</b>	PLA		<b>Dimensiones en bruto</b>	88x80x4 mm	
<b>Realizado por:</b>	Escobar Cacuango Franklin Javier		<b>Observaciones</b>	-----	



Fase	Tiempo de operación	Parámetros de impresión			
		Densidad de relleno	15%	Tipo de Soporte	Árbol
Impresión 3D	45 min	Velocidad de impresión	250 mm/s	Diámetro del extrusor	4 mm

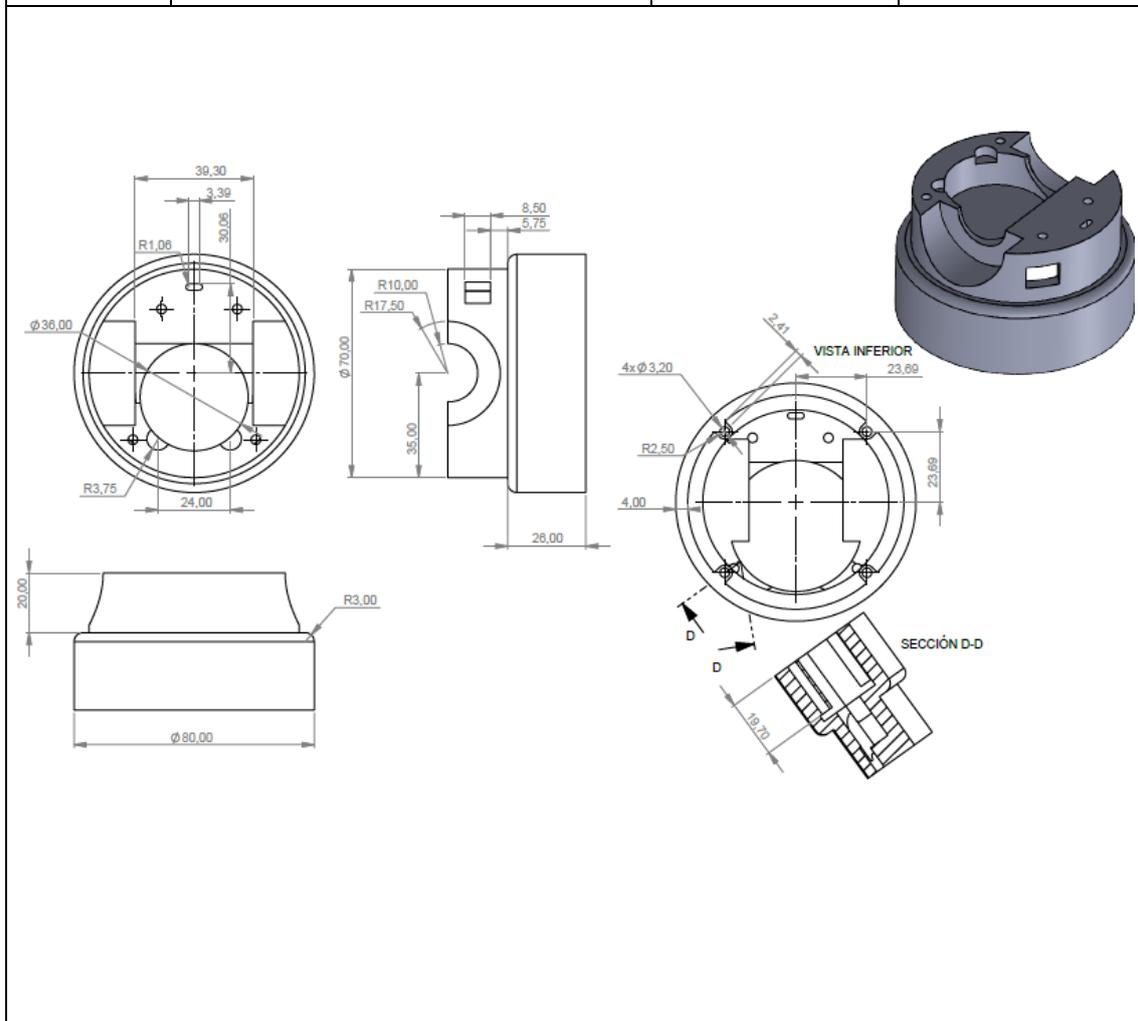
## Hoja de Procesos Carcasa Superior

<b>Proyecto</b>	Medido digital de agua potable		<b>Máquina</b>	Diy Adventurer	<b>Fecha:</b> 30/1/2025
<b>Pieza</b>	Carcasa Superior	<b>Plano Nro.</b>	TMD-CS-004	<b>Hoja</b>	
<b>Material</b>	PLA		<b>Dimensiones en bruto</b>	51,5x80x86x4 mm	
<b>Realizado por:</b>	Escobar Cacuangó Franklin Javier		<b>Observaciones</b>	-----	



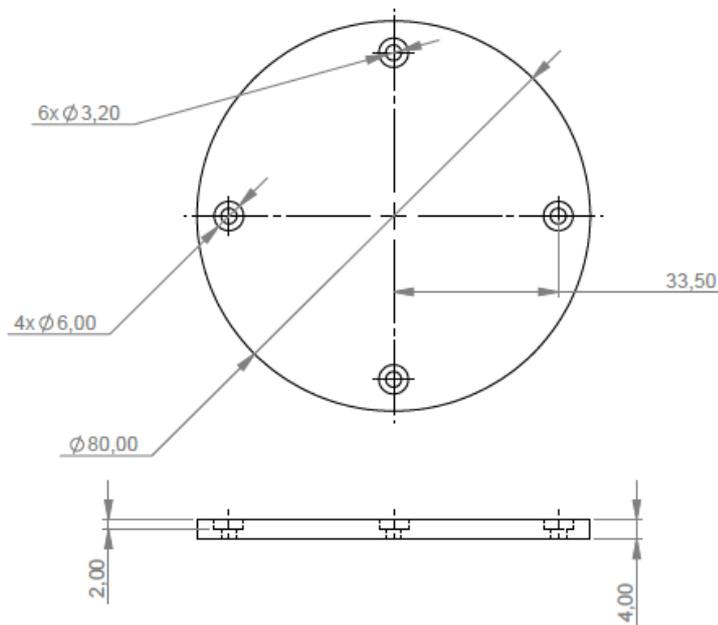
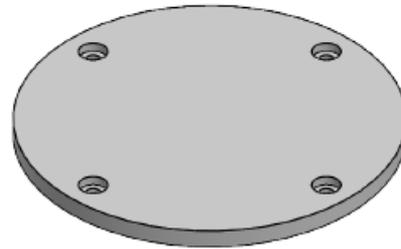
Fase	Tiempo de operación	Parámetros de impresión			
Impresión 3D	4h10min	<b>Densidad de relleno</b>	15%	<b>Tipo de Soporte</b>	Árbol
		<b>Velocidad de impresión</b>	250 mm/s	<b>Diámetro del extrusor</b>	4 mm

Hoja de Procesos Carcasa Inferior					
<b>Proyecto</b>	Medido digital de agua potable		<b>Máquina</b>	Diy Adventurer	<b>Fecha:</b> 30/1/2025
<b>Pieza</b>	Tapa Superior	<b>Plano Nro.</b>	TMD-CI-005	<b>Hoja</b>	
<b>Material</b>	PLA		<b>Dimensiones en bruto</b>	80x47x4 mm	
<b>Realizado por:</b>	Escobar Cacuangó Franklin Javier		<b>Observaciones</b>	-----	



Fase	Tiempo de operación	Parámetros de impresión			
		Densidad de relleno	15%	Tipo de Soporte	Árbol
Impresión 3D	3h17min	Velocidad de impresión	250 mm/s	Diámetro del extrusor	4 mm

Hoja de Procesos Tapa Inferior						
<b>Proyecto</b>	Medido digital de agua potable			<b>Máquina</b>	Diy Adventurer	<b>Fecha:</b> 30/1/2025
<b>Pieza</b>	Tapa Inferior	<b>Plano Nro.</b>	TMD-TI-006	<b>Hoja</b>	1/1	
<b>Material</b>	PLA			<b>Dimensiones en bruto</b>	80x4 mm	
<b>Realizado por:</b>	Escobar Cacuangó Franklin Javier			<b>Observaciones</b>	-----	



Fase	Tiempo de operación	Parámetros de impresión			
		Impresión 3D	30 min	<b>Densidad de relleno</b>	15%
<b>Velocidad de impresión</b>	250 mm/s			<b>Diámetro del extrusor</b>	4 mm