

# **UNIVERSIDAD TÉCNICA DEL NORTE**



## **FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

### **CARRERA DE INGENIERÍA EN MECATRÓNICA**

#### **TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN MECATRÓNICA**

##### **TEMA**

Prototipo electrónico de medición de caudal para el registro de lecturas de  
medidores de agua potable

##### **AUTOR**

PABLO ANDRÉS GÓMEZ CADENA

##### **DIRECTOR**

Milton Gavilánez

**Ibarra-Ecuador**

2023



**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			
<b>CÉDULA DE IDENTIDAD:</b>	1003673678		
<b>APELLIDOS Y NOMBRES:</b>	Gómez Cadena Pablo Andrés		
<b>DIRECCIÓN:</b>	Yacucalle		
<b>EMAIL:</b>	Pagomez@utn.edu.ec		
<b>TELÉFONO FIJO:</b>	062585320	<b>TELÉFONO MÓVIL:</b>	0987920226


DATOS DE LA OBRA	
<b>TÍTULO:</b>	Prototipo electrónico de medición de caudal para el registro de lecturas de medidores de agua potable.
<b>AUTOR (ES):</b>	Pablo Andrés Gómez Cadena
<b>FECHA: DD/MM/AAAA</b>	23/03/2023
SOLO PARA TRABAJOS DE GRADO	
<b>PROGRAMA:</b>	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
<b>TÍTULO POR EL QUE OPTA:</b>	Ingeniería Mecatronica
<b>ASESOR /DIRECTOR:</b>	MSc. Milton Gavilanes

**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 asume (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 23 días del mes de Marzo del 2023

**EL AUTOR:**

(Firma)   
Nombre: ..... Pablo Gómez



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERIA EN CIENCIAS APLICADAS**

**CERTIFICACIÓN**

En calidad de director del trabajo de grado “PROTOTIPO ELECTRÓNICO DE MEDICION DE CAUDAL PARA EL REGISTRO DE LECTURAS DE MEDIDORES DE AGUA POTABLE”, presentado por el egresado Pablo Andrés Gómez Cadena, para optar por el título de Ingeniero en Mecatrónica, certifico que el mencionado proyecto fue realizado bajo mi dirección.

Ibarra, 27 de Febrero 2023

Ing. Milton Gavilanes Mg.

TUTOR DEL TRABAJO DE AGRADO

## Resumen

El presente proyecto muestra el diseño e implementación de un prototipo inteligente de medición del consumo de agua potable para la Junta Administradora de Agua Potable de la Parroquia de Eugenio Espejo, cantón Otavalo, este se desarrolla con el objetivo de automatizar el proceso de recolección y almacenamiento de la información del consumo de agua potable.

El prototipo permite obtener la información del consumo de agua de cada usuario y almacenar todo el historial de consumo en un sistema de base de datos de forma local, la cual respalda la información en caso de que exista pérdida de datos.

Este prototipo de medición de agua cuenta con un sensor de flujo el cual sirve para medir caudal de agua, por medio de su sensor de efecto Hall que a su vez emite un pulso electrónico que puede ser leído de forma digital. El proceso conversión de pulsos electrónicos es efectuado por un modulo Wi-Fi Esp32, que por medio de comunicación inalámbrica a través de un punto de acceso (AP) publica los datos en el servidor PhpMyAdmin. Los datos generados por el prototipo de medición son almacenados en la base de datos con el código del medidor, fecha de registro y los valores del consumo; esta información es tomada por el sistema de facturación con el que cuenta la JAAP.

Los resultados de las pruebas de funcionamiento determinan la fiabilidad, y eficiencia del prototipo. Las pruebas son efectuadas a diferentes partes del sistema como son: lectura

de datos, transmisión – recepción y almacenamiento del consumo de agua en la base de datos, estas pruebas muestran la efectividad del prototipo. Concluyendo de esta forma que el prototipo automatiza el proceso con el que la Junta de Agua Potable de Eugenio Espejo maneja actualmente.

## **Abstract**

*This project shows the design and implementation of an intelligent prototype for measuring drinking water consumption for the Drinking Water Administration Board of the Eugenio Espejo Parish, Otavalo canton, this is developed with the aim of automating the collection and storage process. information on drinking water consumption.*

*The prototype allows obtaining information on the water consumption of each user and storing the entire consumption history in a database system locally, which supports the information in case of data loss.*

*This water measurement prototype has a flow sensor which is used to measure water flow through its Hall effect sensor which in turn emits an electronic pulse that can be read digitally. The electronic pulse conversion process is carried out by a Wi-Fi Esp32 module, which through wireless communication through an access point (AP) publishes the data on the PhpMyAdmin server. The data generated by the measurement prototype is stored in the database with the meter code, registration date and consumption values; this information is taken by the billing system that the JAAP has.*

*The results of the performance tests determine the reliability and efficiency of the prototype. The tests are carried out on different parts of the system such as: data reading, transmission - reception and storage of water consumption in the database, these tests show*

*the effectiveness of the prototype. Concluding in this way that the prototype automates the process with which the Eugenio Espejo Potable Water Board currently manages.*

## 1 Contenido

IDENTIFICACIÓN DE LA OBRA .....	¡Error! Marcador no definido.
CONSTANCIAS .....	¡Error! Marcador no definido.
CERTIFICACIÓN.....	¡Error! Marcador no definido.
DECLARACIÓN .....	¡Error! Marcador no definido.
Resumen .....	III
Abstract .....	V
1 Contenido .....	VII
Lista de figuras .....	XIII
1 Introducción.....	1
1.1 Problema.....	1
1.2 Objetivos. ....	2
1.2.1 Objetivo General. ....	2
1.2.2 Objetivos Específicos. ....	2
1.3 Justificación.....	3
2 Marco Teórico .....	4
2.1 Agua como recurso natural.....	4



2.2	Tipo de consumo de agua potable .....	4
2.3	Consumo de Agua Potable en Ecuador .....	5
2.4	Procedimiento de lectura del consumo de agua potable.....	6
2.5	Medición de consumo del agua en tuberías.....	7
2.6	.....	7
2.6.1	Métodos e instrumentos de medición de caudal.....	7
2.6	Medición Inteligente.....	13
2.7	Tecnologías de transmisión de datos inalámbricos .....	14
2.7.1	Bluetooth .....	14
2.8	Tarjetas de adquisición y procesamiento de señales con tecnología Wi-Fi .....	15
2.8.1	Raspberry Pi .....	15
2.8.2	Arduino.....	16
2.8.3	ESP32s.....	18
2.9	Base de datos .....	19
2.9.1	Firestore .....	19
2.9.2	PhPMYADMIN .....	20
2.10	Plataformas de software libre para el desarrollo .....	21

2.10.1	IDE de Arduino .....	21
2.11	Internet de las cosas (IoT) .....	22
3	Metodología.....	25
3.1	Especificaciones de diseño .....	25
3.2	Diseño conceptual del sistema .....	26
3.3	Requerimientos del prototipo .....	26
3.2	3.4 Elección del sensor de flujo.....	28
3.5	Elección de la tarjeta de procesamiento de datos. ....	30
3.6	Elección de la base de datos .....	31
3.7	Diseño.....	32
3.7.1	Estructura General .....	32
3.7.2	Nodo cliente. ....	33
3.7.3	Comunicación con base de datos.....	34
3.7.4	Base de datos .....	35
3.8	Diagrama de conexión entre la tarjeta ESP32 y el sensor FS400A.....	36
3.8.1	Medición y registro de datos del sensor a través de ESP32 .....	37
3.8.2	Diseño de circuito para baquelita .....	40

3.2.2	Creación de base de datos .....	43
3.2.3	Comunicación del módulo ESP con el servidor .....	44
4	Implementación y pruebas de funcionamiento.....	47
4.1	Pruebas de medición del sensor de flujo .....	47
4.1.1	Determinación del error.....	48
4.1.2	Datos recopilados en la tarjeta de adquisición .....	51
4.2	Pruebas de conectividad .....	51
4.2.1	Conexión en hogar.....	51
4.2.2	Conexión con servidor local.....	52
4.3	Pruebas de funcionamiento de servidor local.....	53
4.4	Prueba de autonomía eléctrica.....	54
4.5	Ubicación de sistema de medición .....	56
5	Conclusiones y recomendaciones.....	57
5.1	Conclusiones .....	57
5.2	Recomendaciones.....	58
6	Bibliografía.....	60



Tabla 2.1. Niveles de potencia de transmisores Bluetooth [17]. .....	14
<b>Tabla 3.1</b> Requerimientos de Hardware y Software (Autoría propia).....	26
<b>Tabla 3.2</b> Comparación de sensores de flujo de acuerdo con los requerimientos establecidos (Autoría propia) .....	28
Tabla 3.3 Comparación de tarjetas de desarrollo con tecnología Wi-Fi (Autoría propia) ....	30
<b>Tabla 3.4</b> Comparación de base de datos (Autoría Propia) .....	31
Tabla 4.1 Registro de datos con el valor promedio de pulsos .....	50

## Lista de figuras

Figura 2.1 Consumo de agua potable registrado por provincias .....	5
<b>Figura 2.2</b> Indicador numérico [14]. .....	10
<b>Figura 2.3</b> Indicadores de aguja [14]. .....	11
<b>Figura 2.4</b> Caudalímetro electrónico a turbina Modelo FS300A [16]. .....	12
Figura 2.5 Raspberry Pi 3B [18]. .....	16
<b>Figura 2.6</b> Arduino MKR1000 [19]. .....	17
<b>Figura 2.8</b> Esquema de entradas y salidas ESP32s [16]. .....	19
<b>Figura 2.9</b> Base de Datos Firebase [20]. .....	20
<b>Figura 2.10</b> Base de datos phpMyAdmin.....	20
<b>Figura 2.11</b> IDE de Arduino.....	22
<b>Figura 2.12</b> Ejemplo de un modelo de comunicación dispositivo a dispositivo [21]. .....	23
<b>Figura 2.13</b> Modelo de comunicación dispositivo a la nube [21]. .....	24

<b>Figura 3.1</b> Diagrama de bloques .....	33
Figura 3.2 Diagrama de flujo de nodo cliente.....	34
Figura 3.3 Diagrama de flujo de la base de datos.....	35
Figura 3.4 Diagrama de FS400A.....	36
Figura 3.5 Diagrama de conexión entre FS400A y ESP32 .....	37
Figura 3.6 Código de prueba de funcionamiento del sensor de flujo.....	38
Figura 3.7 Datos interpretados por la tarjeta ESP32 .....	40
Figura 3.8 Código de conexión a red wifi local .....	44
Figura 3.9 Base de datos local en la Junta de aguas .....	45
Figura 3.10 Herramientas de plataforma PhPMyAdmin.....	<b>¡Error! Marcador no definido.</b>
Figura 3.11 Tablas con datos registrados .....	46

# 1 Introducción

## 1.1 Problema

En la actualidad las juntas administradoras de Agua Potable realizan el registro del consumo de agua mediante la lectura de un medidor de caudal mecánico, anotado este valor en una libreta de lectura que posteriormente son utilizadas para el cálculo del consumo de agua potable de la comunidad [1]. Este tipo de registro que se realiza en la actualidad presenta un importante problema ya que puede fallar por causa de un error humano, tanto en la medición de lectura del medidor como el cálculo del consumo de agua debido a que este trabajo es realizado de manera manual, dando como resultado un pago erróneo con un valor inferior o superior al que realmente debe cancelar el usuario consumidor, ocasionando insatisfacción, reclamos y pérdidas económicas para la Junta Administradora de Agua Potable.

Esto tomado como referencia el análisis realizado al Sistema Informático de la Junta Administradora de Agua Potable de la Parroquia Eugenio Espejo donde se encontró inconsistencias en las lecturas ingresadas al sistema que no eran las mismas que el medidor registraba [2]



Después del contexto se llega a determinar el siguiente problema “Error en la medición y registro de lecturas de medidores en procesos de cobro de consumo de agua potable”.

La finalidad del prototipo que se presentara en este proyecto es automatizar el sistema de registro del consumo de agua potable y así evitar los errores que se comenten al realizar este registro de manera manual, por ello es viable el uso de una tarjeta de adquisición inalámbrica, fomentando el uso de nuevas tecnologías IoT.

## **1.2 Objetivos.**

### **1.2.1 Objetivo General.**

Desarrollar un prototipo electrónico de medidor para el registro de lecturas, aplicado al cálculo de consumo de agua potable en instalaciones residenciales.

### **1.2.2 Objetivos Específicos.**

- Identificar los elementos y el método más apropiado para desarrollar el prototipo.
- Diseñar el prototipo para el cálculo y registro del consumo de agua potable.
- Implementar el prototipo de medición del caudal de agua potable.
- Evaluar los resultados obtenidos mediante pruebas de validación del prototipo.

### **1.3 Alcance**

El proyecto está enfocado a desarrollar un prototipo para la medición de consumo de agua potable, este prototipo constará de un sensor de caudal y una tarjeta interactiva. La adquisición del registro de consumo se realiza de forma inalámbrica utilizando las tecnologías IoT. El registro de consumo de agua potable se presentará en un aplicativo. Se finalizará realizando pruebas de funcionamiento verificando la eficacia del prototipo.

### **1.3 Justificación**

El prototipo de medición de caudal proveerá un registro del caudal medido mensualmente. De esta forma se evita errores humanos al tomar dichas medidas dadas por los medidores de agua potable existentes. A este registro mensual tendrá acceso la institución administradora de este para monitorear su consumo.

## **2 Marco Teórico**

### **2.1 Agua como recurso natural**

El agua es el elemento más frecuente en el planeta, del cual únicamente el 2.53% es agua dulce y el resto es agua salada, el agua dulce disponible se encuentra en lagos, ríos y acuíferos almacenados en embalses [3].

Históricamente los ríos han constituido fuente de riqueza al proporcionar el agua imprescindible para la subsistencia y posterior desarrollo, proporcionando fertilidad a los suelos y facilitando la comunicación entre pueblos [4].

### **2.2 Tipo de consumo de agua potable**

Existen diferentes usos en los que el agua potable es empleada, tales como [5]:

- Consumo doméstico: se utiliza en viviendas para actividades de consumo, sanitarios entre otros, este varía entre la costumbre de la ciudadanía.
- Consumo público: hace referencia al consumo en; escuelas, mercados, hospitales, cuarteles, agua de riego, servicio contra incendios, este consumo es muy cambiante debido a desperdicios en el sistema de distribución [6].
- Consumo industrial: representa el agua utilizada por fábricas, empresas y hoteles, el consumo varía dependiendo del tamaño de la industria, en algunos casos las industrias cuentan con su propia planta de agua potables [7]

- Consumo comercial: el consumo de agua potable realizado por centro comerciales, tiendas, locales que están enfocados al comercio o brindan algún tipo de servicio

### 2.3 Consumo de Agua Potable en Ecuador

En el censo de información ambiental en hogares realizado en Junio del 2014 dio a conocer que el 76,51% de los hogares tienen acceso al agua potable, existiendo 31 Sistemas Hidrográficos conformados por 79 cuencas. El consumo de los hogares en el área urbana es de alrededor de 26,73 m<sup>3</sup> y en el área rural 27,74 m<sup>3</sup>, el gasto promedio mensual en el área urbana es de 23,64 dólares (figura 1.1) [8].

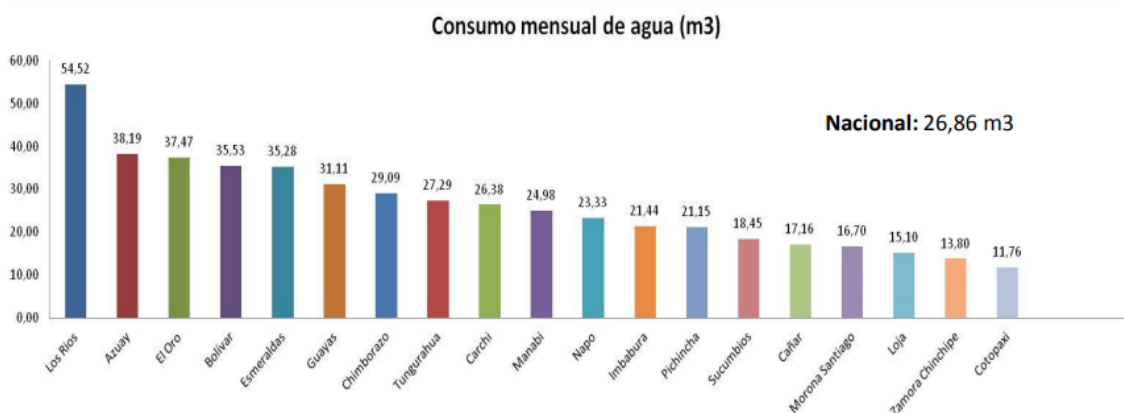


Figura 2.1 Consumo de agua potable registrado por provincias [8]

La regulación y control de los recursos hidrológicos lo ejerce el ARCA la Agencia de Regulación y Control del Agua, este realiza la gestión de los servicios públicos relacionados

al sector del agua y sus usos, de esta manera se identificó que 128 municipios realizan campañas para el ahorro del agua, de los cuales 91 municipios realizaron proyectos para proteger las fuentes de agua potable [8].

#### **2.4 Procedimiento de lectura del consumo de agua potable**

Las juntas de Agua Potable y Alcantarillado tienen como responsabilidad brindar un buen servicio a todos los usuarios que habitan en área de atención de la parroquia. Tomando como ejemplo la Junta Administradora de Agua Potable de la Parroquia Eugenio Espejo ubicada en el cantón Otavalo, el proceso actual de medición del consumo de agua en los domicilios se realiza por una toma de lectura, con la intervención de la mano humana, para lograr una buena obtención de la lectura se tiene la información de cada cliente (código único del medidor mecánico) y su ubicación geográfica (sector y número de casa), para posteriormente dirigirse rápida y ordenadamente a cada domicilio establecido en su ruta. [1]

La obtención de la lectura del consumo de agua potable y posterior registro de datos se lo anota en un cuaderno de registro o en una libreta de propiedad de la junta de Eugenio Espejo. Una vez que se a obtenido los datos del consumo de agua, por la persona encargada, estos son digitalizados manualmente en un programa de facturación que se encuentra en el computador de la JAAP de la parroquia. Ya que todo este proceso se lo realiza de forma manual, existen errores humanos, como pueden ser:

- Mala interpretación de la lectura del medidor.
- Escribir incorrectamente el consumo en el programa de facturación.

En conclusión, por los errores que se presentan al momento de adquirir o ingresar el consumo de agua, existen reclamos de miembros de la junta, como altos consumos de sus medidores teniendo así insatisfacción del servicio brindado por la junta de agua potable.

## **2.5 Medición de consumo del agua en tuberías**

La medición es la mejor forma de racionalizar el consumo de agua, acompañada de un buen pliego tarifario incentiva al consumidor a utilizar sólo el líquido necesario. La medida del caudal en una tubería consiste en la determinación de la cantidad de masa o volumen que circula por unidad de tiempo, este fluido es impulsado principalmente por una diferencia de presión. Como resultado existen diferentes métodos de medición de flujo y diferentes instrumentos de medición, dentro de estos métodos existen tres grupos, medidores de presión diferencial, medidores de desplazamiento positivo y medidores electrónicos [9].

### **2.6.1 Métodos e instrumentos de medición de caudal.**

Un medidor es un aparato que sirve para medir y registrar el volumen de agua que lo atraviesa, los medidores se componen de dos partes principales, una parte mide el consumo de agua, otra registra lo medido. De acuerdo con su funcionamiento los medidores se

clasifican en medidores de presión, medidores de desplazamiento positivo, medidores electrónicos a turbina [10].

Para uso doméstico estos medidores deben cumplir con propiedades que están dadas por la empresa que suministra el agua potable. Estas propiedades son:

- Caudal máximo: 3.125 m<sup>3</sup>/h
- Caudal permanente: 2.5 m<sup>3</sup>/h
- Caudal mínimo: 65 lt/h
- Presión permisible: 10 bar
- Sensibilidad permisible: +- 5%
- Lectura mínima: 0.05lt
- Lectura máxima: 99999 m<sup>3</sup>

### ***2.5.1.1 Medidores de desplazamiento positivo.***

Son medidores de flujo que mide directamente el volumen del fluido que pasa a través de una tubería. Esto se logra atrapando una cantidad específica del fluido con cada rotación del medidor. La velocidad rotativa del rotor es directamente proporcional a la tasa del flujo, pues el caudal del fluido es la causa de la rotación, siendo así medidores muy precisos independientemente de la viscosidad, densidad, velocidad o temperatura del fluido [11].

Su funcionamiento se deriva en dos tipos:

- **Chorro Único**

Es un medidor cuyo funcionamiento se basa en el movimiento del agua sobre el rotor con una única entrada tangencialmente dirigido, este movimiento se transmite magnéticamente a las agujas y tambores numerados, transformando a una lectura de litro/metros cúbicos, este tipo de medidor se caracteriza por su calibración externa y por tener una esfera seca que impide el empañamiento y el ensuciamiento de las piezas internas por aguas duras o ferruginosas [12].

- **Chorro Múltiple**

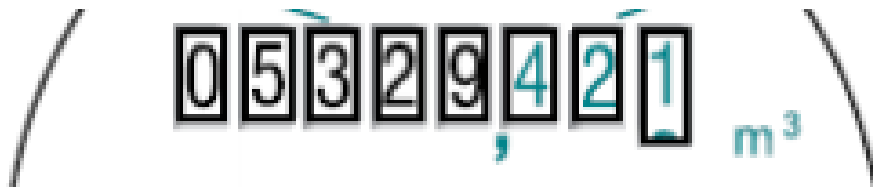
Es un medidor que al pasar el agua por la cámara de medición hace girar una turbina cuyo movimiento se transmite magnéticamente al registro y este lo convierte en la cantidad de agua que paso por la turbina. La entrada del agua a la cámara de medición es realizada por varias entradas que producen el efecto de chorro múltiple, esto permite que el medidor se conserve a lo largo del tiempo sus características hidráulicas [13].

Estos medidores tanto los de chorro único y chorro múltiple tiene un mecanismo interno el cual es accionado por el flujo de agua, este mecanismo mueve de manera continua las agujas y los tambores numerados, ubicados en la caratula del medidor donde se ubica el registro para realizar la lectura del consumo de agua [14].

Los componentes de la caratula del medidor son las siguientes:

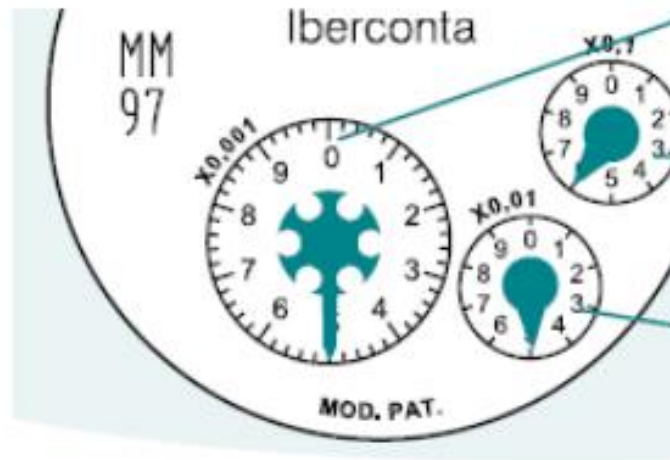


- Tambor numerado: Los medidores convencionales que se encuentran en cada hogar de Ecuador tienen una numeración en la parte superior, esta numeración esta hecha con un código de colores, en el cual el color negro indica los metros cúbicos, y el color azul o rojo indica los litros consumidos.



**Figura 2.2** Indicador numérico [14].

- Agujas: Estos elementos indican la cantidad de agua que se a consumido, primero se debe observar los números junto al reloj denominados factores, según su factor será la cantidad de agua que indica la aguja, se debe tomar en cuenta que estos indicadores trabajan en conjunto ya que, si uno de ellos marca cualquier número, este valor no está completo, si el indicador anterior no ha pasado de 9 a 0.



**Figura 2.3** Indicadores de aguja [14].

- Factor de X0,00001: Indica que la aguja está marcando una décima parte de un litro (decilitro) de consumo.
- Factor de X0,001: Indica que la aguja está marcando un litro de consumo.
- Factor de X0,01: Indica que la aguja está marcando las decenas de litros.
- Factor de X0,1: Indica que la aguja está marcando los cientos de litros.

### 2.5.1.2 Medidores electrónicos a turbina

Una forma muy común de medir el caudal de un fluido es por medio de una turbina que gira al pasar el fluido a través de ella, cuya velocidad es proporcional al caudal del fluido

que lo atraviesa. Este tipo de medidor utiliza un imán que activa un sensor de efecto Hall generando un pulso positivo con cada revolución, este imán se encuentra unido a la turbina del caudalímetro. Como el volumen del agua por cada pulso es fijo, se puede contar la cantidad de pulsos por unidad de tiempo y así multiplicarlo por el valor de volumen/pulso así se obtiene el caudal del fluido. El sensor de efecto Hall se encuentra aislado de los fluidos de manera que se mantiene seco y seguro [15].



**Figura 2.4** Caudalímetro electrónico a turbina Modelo FS300A [16].

## **2.6 Medición Inteligente**

La medición inteligente de agua potable deriva en potenciar la administración del consumo de agua potable ya que permite determinar en tiempo real o casi en tiempo real, el consumo de agua, ofreciendo la posibilidad de leer este consumo de forma local o remota, siendo una solución que permite leer, procesar y enviar la información del consumo mediante el uso de equipos electrónicos programados con funciones específicas, optimizando el funcionamiento de estas, mejorando la calidad del servicio y su seguridad.

El crecimiento de la población a nivel nacional y la reducción de los recursos naturales han generado el correcto aprovechamiento y ahorro del patrimonio natural especialmente del recurso hídrico (agua potable), lo cual a permitido que se desarrollen sistemas o dispositivos que permitan contabilizar y gestionar el uso de este recurso.

Los sistemas de medición inteligente deben cumplir diferentes características como:

- Lectura local o remota.
- Posibilidad de eventual desconexión remota del usuario.
- Capacidad de interactuar con otros dispositivos.

## 2.7 Tecnologías de transmisión de datos inalámbricos

La tendencia en el campo del Internet Of Things, es un concepto que se basa en una mayor conectividad, debido a la interconectividad de varios dispositivos [5]. Debido a la tendencia en las tecnologías existes distintas tecnologías de transmisión de datos.

### 2.7.1 Bluetooth

Esta tecnología opera en frecuencias entre 2.4 GHz y 2.485 GHz, la 4ta generación de esta tecnología alcanza una transmisión de datos de 24Mbit/s [17], el alcance de esta especificación depende de los niveles de potencia que estén utilizando como se puede apreciar en la tabla 2.1.

**Tabla 2.1.** Niveles de potencia de transmisores Bluetooth [17].

Niveles de potencia	Potencia	Alcance
Clase 1	100mW	100m+
Clase 2	2.5 mW	10m
Clase 3	1mW	1m

Como características principales de esta tecnología tenemos:

- Trabaja en la banda libre de 2.4 GHz
- Presenta mecanismos de ahorro de energía debido a que no siempre consume la misma potencia.
- No tiene un costo elevado.
- Facilita la transmisión de información entre dispositivos personales.

## **2.8 Tarjetas de adquisición y procesamiento de señales con tecnología Wi-Fi**

Este tipo de tecnología es primordial para el procesamiento y envío de datos como el consumo de agua, registrándose en una base de datos para posteriormente realizar el cálculo de la tarifa consumida por el usuario mensualmente.

### **2.8.1 Raspberry Pi**

La tarjeta Raspberry Pi es un ordenador completamente funcional construida en una sola placa de circuito de arquitectura abierta. Esta tarjeta cuenta con diferentes puertos como USB, HDMI, Ethernet, entre otros, pero esta placa se caracteriza por los pines que componen el conector con entradas-salidas de propósito general (GPIO) que se utilizan para interactuar con otros hardware por medio de diferentes tipos de comunicación. Comúnmente es utilizado para aprender a programar, crear diferentes circuitos o dispositivos físicos, con el fin de fomentar más ordenadores prácticos para la educación en todo el mundo [18].

El modelo 3B que es el más común en el mercado, cuenta con un CPU Cortex-A53 con arquitectura de 64-bit a 1.4GHz, memoria SDRAM de 1GB y sus puertos de conexión antes mencionados.

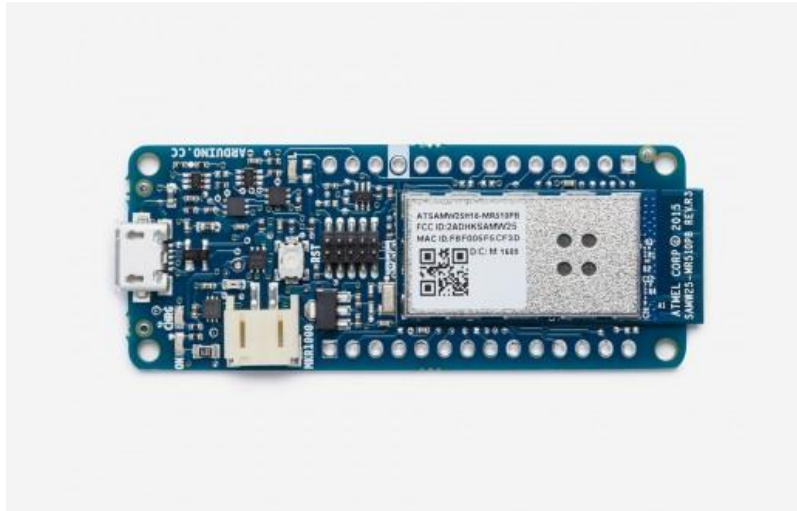


Figura 2.5 Raspberry Pi 3B [18].

## 2.8.2 Arduino

Arduino es una plataforma de código abierto diseñada para la creación de electrónica con entradas y salidas(E/S), analógicas y digitales y un microcontrolador reprogramable. Posee un entorno de desarrollo integrado IDE, que implementa un lenguaje de programación Processing/Wiring basado en Java.

Arduino no es un modelo concreto de placa ya que cuenta con una variedad de placas con diferentes características, una de ellas es la implementación de la tecnología Wi-Fi teniendo como ejemplo la placa ARDUINO MKR1000 WIFI.



**Figura 2.6** Arduino MKR1000 [19].

Arduino MKR1000 se ha creado con una solución practica para los desarrolladores integrando la conectividad Wi-Fi basada en el módulo Atmel ATSAMW25 Soc diseñado específicamente para proyectos y dispositivos de IoT. Esta placa cuenta con un microcontrolador SAMD21 de 32bits, 8 entradas/salidas digitales, 12 salidas PWM, 7 entradas analógicas, una salida analógica y el modulo Wi-Fi antes mencionado [19].

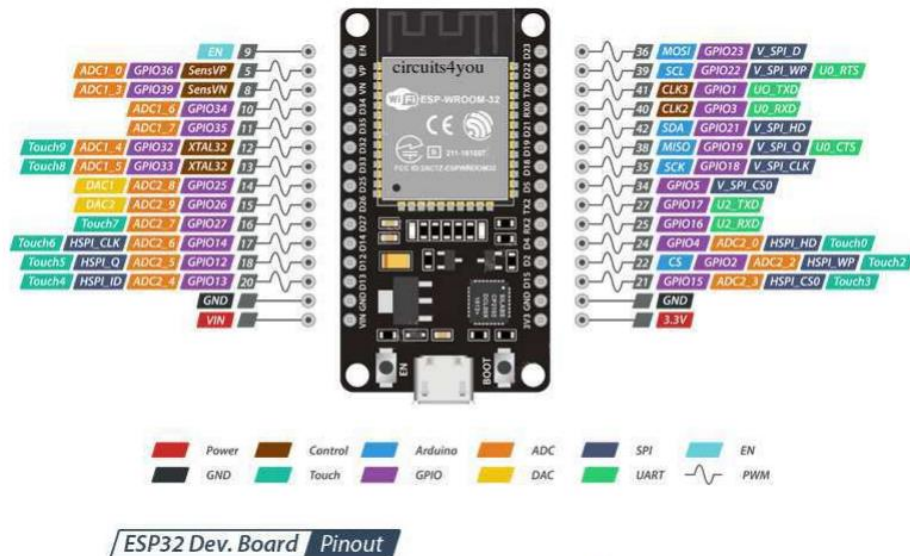


### **2.8.3 ESP32s**

Es una placa de código abierto que integra diferentes características como una solución Wi-Fi/Bluetooth, una interfaz periférica serial (SPI), un procesador de doble núcleo, regulador de voltaje de 4v~10v a 3.3v.

La tarjeta ESP32 que combina algunas características de la placa Arduino ordinaria, sus múltiples puertos GPIO poseen un API avanzada para el control de entradas y salidas que permiten conectar a la placa otros periféricos capaces de generar PWM, I2C, SPI y UART en comunicación serial.

Esta placa es utilizada para proyectos enfocados en la tecnología IoT, lo cual tiene un perfecto rendimiento para poder realizar múltiples tareas en enviar información o recoger información a través de una conexión inalámbrica [16].



**Figura 2.8** Esquema de entradas y salidas ESP32s [16].

## 2.9 Base de datos

### 2.9.1 Firebase

Firebase es un servicio computacional en la nube (Backend as a Service) proporcionada por Google que facilita el desarrollo apps. Esta herramienta ofrece diferentes servicios como una base de datos en tiempo real, autenticación de usuario, sistema de notificaciones Push utilizando una API destinada para ello, almacenamiento de archivos, funciones backend permitiendo ejecutar código de Javascript en el servidor, hosting y analítica [20].



**Figura 2.9** Base de Datos Firebase [20].

### 2.9.2 PhPMyADMIN

Es una herramienta de software libre escrita en un entorno php destinada a manejar una administración MySQL de forma local la cual es instalada en un computador mediante diferentes programas como xampp, mamp o wamp. Esta aplicación permite crear y eliminar bases de datos, ejecutar sentencias SQL, administrar términos privilegios y exportar datos en varios formatos.



**Figura 2.10** Base de datos phpMyAdmin

## **2.10 Plataformas de software libre para el desarrollo**

Estas plataformas pueden ser usadas para cualquier propósito siendo adaptadas a las necesidades del usuario, dando acceso al código fuente mejorándolo y adaptándolo para el beneficio de toda la comunidad, de esta forma ofrecen un gran potencial para realizar desarrollos gracias a la libertad que para efectuar los cambios en el diseño o en los códigos dependiendo de la necesidad del usuario.

### **2.10.1 IDE de Arduino**

El entorno de desarrollo integrado (IDE), es un entorno de programación empaquetado como un programa de aplicación, este consiste en un editor de código, un compilador, un depurador y un constructor de interfaz grafica. Los programas desarrollados en Arduino están compuestos por un solo fichero con extensión “ino” aunque es posible organizarlo en varios ficheros. Además, para su facilidad de uso este incorpora herramientas para poder cargar el programa ya compilado en una memoria flash integrada en el hardware.

Ventajas del IDE de Arduino

- Lenguaje de programación basado en C/C++
- Tiene una comunidad de desarrollo amplia que permite el acceso a diferentes librerías, ejemplos y referencias de gran ayuda



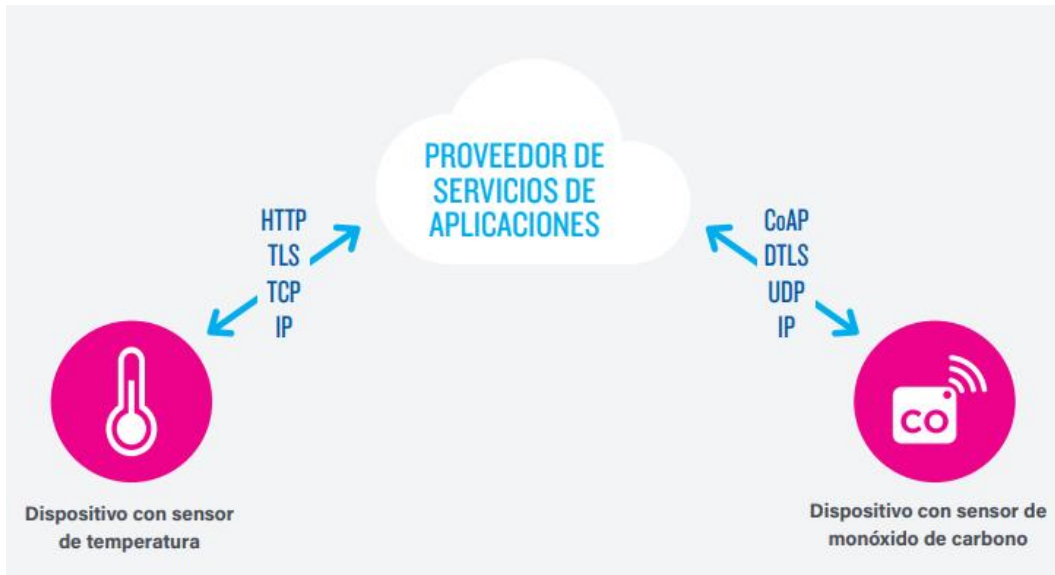
conecta y se comunican directamente entre sí y no a través de un intermediario como un servidor, el tipo de conexión que se aplica es por redes IP, por medio de la internet o en ciertos casos puntuales se aplica los protocolos Bluetooth, Z-Wave o Zigbee, de esta forma los dispositivos pueden comunicarse o intercambiar mensajes. En general, este modelo de comunicación se utiliza en varias aplicaciones como la automatización del hogar, ya que se utiliza pequeños paquetes de datos para la comunicación entre dispositivos con requerimientos básicos en la tasa de transmisión [21].



**Figura 2.12** Ejemplo de un modelo de comunicación dispositivo a dispositivo [21].

La comunicación de los dispositivos a la nube, se relacionan en forma directa, de esta forma en general tienen mecanismos de seguridad integrados, además se utilizan modelos de datos específicos para cada dispositivo si se requiere una comunicación directa hacia un servidor situado en la nube para aprovechar los mecanismos de comunicación existentes. Utilizando

una conexión a internet para transmitir información para su análisis, sin embargo intentar integrar dispositivos de diferentes fabricantes pueden surgir problemas de interoperabilidad.



**Figura 2.13** Modelo de comunicación dispositivo a la nube [21].

### **3 Metodología**

El capítulo analiza los diferentes componentes que se utilizaran en el desarrollo del prototipo como la descripción de sus características antes mencionadas y su modo de operación, tomando en cuenta cada uno de estos parámetros para alcanzar el objetivo planeado.

#### **3.1 Especificaciones de diseño**

El presente proyecto está diseñado y construido para soportar años en pleno funcionamiento ya que cuenta con un Case de protección y elementos que trabajan con una tensión de 5v para un funcionamiento óptimo, así se garantiza el envío y almacenamiento correcto de los datos obtenidos por el sensor de caudal, cumpliendo así con el propósito principal de este proyecto.

El prototipo está construido de la siguiente manera:

- Sensor de flujo (Utilizado para obtener el consumo exacto de agua potable)
- Tarjeta de adquisición de datos con tecnología Wi-Fi
- Timer en tiempo real
- Sistema de carga solar
- Case de protección



### 3.2 Diseño conceptual del sistema

En la actualidad existen diferentes maneras inteligentes de tomar los datos del consumo de agua potable registrado por los medidores, una de las más empleadas en la utilización de las redes Wi-Fi, ya que se aprovecha el acceso a una red ya existente teniendo una propia red de transmisión de datos. El desarrollo del prototipo se enfoca en el diseño e implementación de un sistema de medidores de agua digitales trabajando con la infraestructura de red que cuenta la junta de agua potable de la parroquia Eugenio Espejo para la transmisión de datos en tiempo real.

### 3.3 Requerimientos del prototipo

Por medio del levantamiento de información se estableció los parámetros que debe cumplir el prototipo y se establece las limitaciones funcionales de esta manera se especifican los requerimientos de: uso, interfaces y estados físicos. Se emplea la abreviatura SySR para hacer referencia al número de requerimientos. En la tabla 3.1 se muestran todos los requerimientos del proyecto.

**Tabla 3.1** Requerimientos de Hardware y Software (Autoría propia).

Requerimientos de arquitectura				
Requerimientos de Software		Prioridad		
SySR1		Alta	Media	Baja

	Una base de datos compatible con la placa de procesamiento.	X		
SySR2	Se precisa de un software que permita programar la placa de procesamiento.	X		
SySR3	Se requiere una base de datos local que permita almacenar el consumo de agua potable.	X		
Requerimientos de la tarjeta de procesamiento				
SySR4	La placa de control debe contar con un módulo wifi con un alcance de 10m como mínimo.	X		
SySR5	La tarjeta de procesamiento debe contar con interfaces periféricas como I2C y SPI.	X		
SySR6	La placa debe permitir reaccionar a eventos externos a la placa de forma rápida (Interrupciones)	X		
SySR7	Se requiere una placa regida a la relación calidad/precio.	X		
Requerimientos eléctricos				
SySR8	Se requiere una fuente de alimentación de 5v DC para la tarjeta de procesamiento y el sensor de flujo.	X		
SySR9	La fuente de energía debe contar con un módulo de carga.	X		

SySR10	Se requiere una fuente de alimentación regida a la relación calidad/precio.	X		
Requerimientos del sensor de caudal				
SySR11	El sensor debe tener una precisión de máximo +-5%	X		
SySR12	El caudal máximo debe ser hasta 3 m <sup>3</sup> /h y el caudal mínimo de 65 l/h.	X		
SySR13	Soportar una presión máxima de 10bar			

### 3.2 3.4 Elección del sensor de flujo.

El sensor de flujo genera pulsos de voltaje por medio de su sensor de efecto hall integrado en su turbina. Siendo la parte mas importante dentro del proyecto se busca el cumplimiento de diferentes características. La tabla 3.2 define si los modelos que existen en el mercado cumplen con los requerimientos establecidos.

**Tabla 3.2** Comparación de sensores de flujo de acuerdo con los requerimientos establecidos

(Autoría propia)

Sensor de Flujo	Requerimientos			Valoración
	SySR11	SySR12	SySR13	

YF-S201	1	0	0	1
FS300A	1	1	0	2
FS400A	1	1	1	3

Elección: El sensor FS400A cumple con todos los requerimientos planteados, y se puede adecuar a la implementación del proyecto.

Características del sensor de flujo FS400A [22].

- Voltaje de operación 5v-24v DC
- Consumo de corriente 15mA
- Salida: Onda cuadrada pulsante
- Rango de flujo: 1-60 l/min
- Presión de trabajo máxima: 12 bar

### 3.5 Elección de la tarjeta de procesamiento de datos.

La tarjeta de procesamiento se realiza la recolección y procesamiento de datos que genera el sensor de flujo, los datos que el sensor arroja son pulsos de voltaje generados por el sensor de efecto hall que el sensor tiene integrado.

La selección de la tecnología para transmitir los datos del consumo de agua, esta regida a la relación de calidad/precio. En la tabla 3.3 se compara los tipos de tarjeta con tecnología wifi que existen en el mercado.

**Tabla 3.3** Comparación de tarjetas de desarrollo con tecnología Wi-Fi (Autoría propia)

Placa de procesamiento	Requerimientos				Valoración
	SySR4	SySR5	SySR6	SySR7	
NodeMCU ESP8266	1	1	0	1	3
Arduino MKR1000	1	1	1	0	3
Raspberry Pi Zero	0	1	1	0	2
ESP32s	1	1	1	1	4

Elección: se selecciona la placa de procesamiento ESP32s para procesar los datos recogidos por el sensor de flujo y envío a la base de datos. Ya que cumple con los requerimientos especificados anteriormente, también se escoge esta placa por su bajo costo en el mercado.

Características de la tarjeta de desarrollo ESP32 [16]

- Procesador: Tensilica Xtensa LX6 de 32bits
- Wi-Fi: 2.4GHz hasta 150 Mbit/s
- Bluetooth Low Energy
- Frecuencia de Clock: Programable, hasta 240MHz

### 3.6 Elección de la base de datos

La base de datos es el entorno donde se almacenará el registro del consumo de agua que suministra el medidor, para este proyecto se requiere de una plataforma libre que se adapte a la tarjeta de procesamiento y al software que posee la JAAP de Eugenio Espejo.

**Tabla 3.4** Comparación de base de datos (Autoría Propia)

Base de datos	Requerimientos			Valoración
	SySR1	SySR3	SySR7	
PhpMyAdmin	1	1	1	3

Oracle	1	0	0	1
FireBase	0	1	1	2

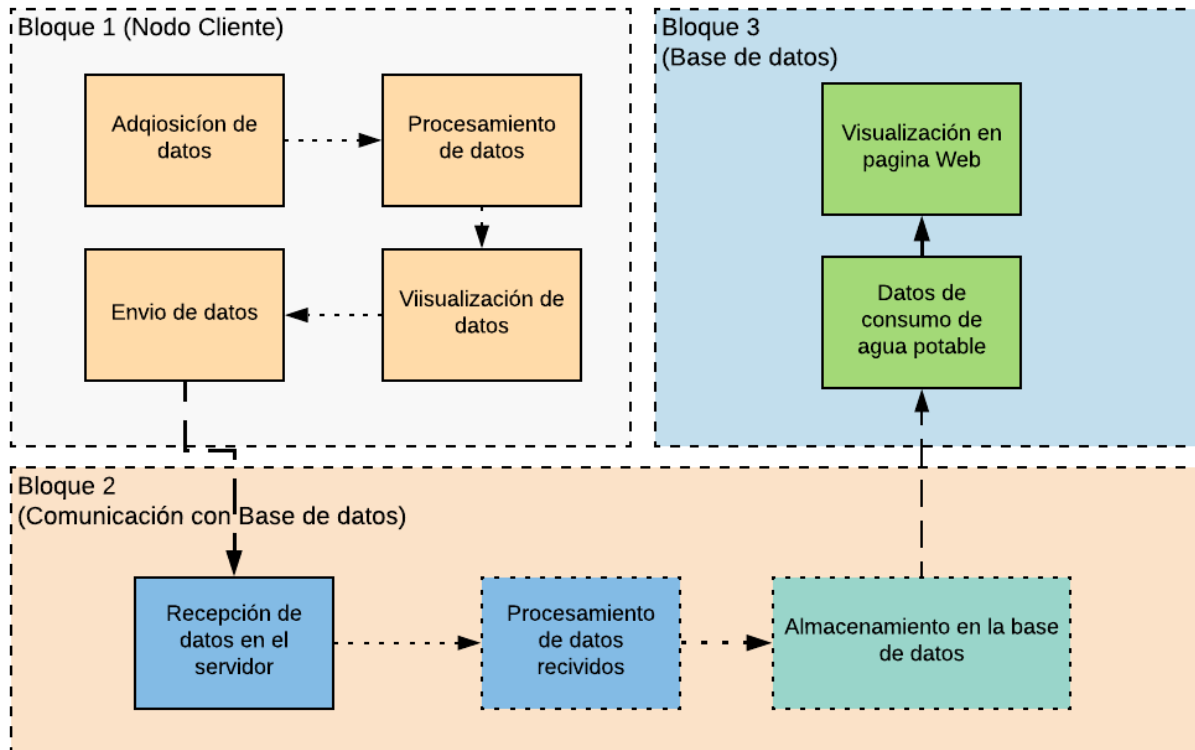
Elección: se selección la base de datos para almacenar el consumo de cada vivienda se selecciona la base de datos PhPMyAdmin ya que cuenta con una transmisión de datos instantánea a un servidor local y por la gratuidad que presenta en el mercado.

### **3.7 Diseño**

Para la construcción del prototipo de medición de agua se utiliza la información que se obtuvo anterior mente ya que se considera las limitaciones del sistema, la mas importante el sistema de carga del prototipo ya que depende exclusivamente del usuario.

#### **3.7.1 Estructura General**

La estructura genera que se implementara el funcionamiento y conexión del sistema el cual esta conformado por tres bloques que permiten obtener el resultado final como muestra la figura 3.1.



**Figura 3.1** Diagrama de bloques

### 3.7.2. Nodo cliente.

En el bloque 1 se realiza la adquisición de datos mediante la lectura por medio del medidor de flujo FS400A, la lectura de datos es recopilada por medio de la tarjeta de adquisición ESP32, esta procesa los datos adquiridos y se encarga de enviar la información recopilada a una base de datos a través de una conexión de red Wifi de hogar del usuario.



En la figura 3.2 se muestra un diagrama de flujo en el cual se determina el funcionamiento de medición y transmisión de datos que se realiza en los medidores de cada hogar.

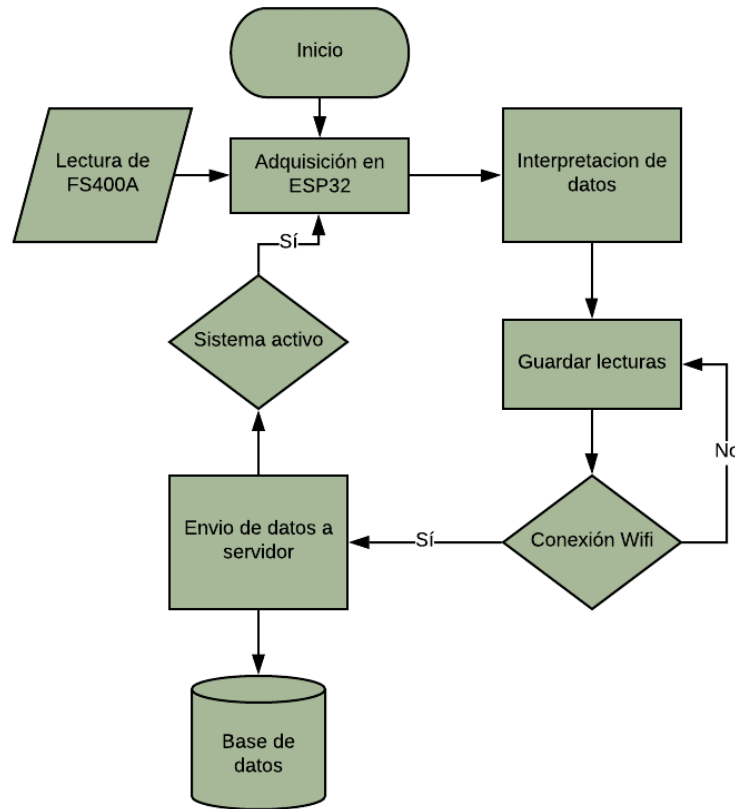


Figura 3.2 Diagrama de flujo de nodo cliente.

### 3.7.3 Comunicación con base de datos.

El bloque 2 visto en la figura 3.1 describe la entrada de datos en la base de datos programada con PhPMyAdmin, en un servidor local estos datos son procesados y

guardados, esta información es enviada desde el hogar por medio de la tarjeta ESP32 a través de un rúter en cada casa, la tarjeta se encuentra programada con la dirección de la base de datos.

### 3.7.4 Base de datos

El diagrama de flujo de la figura 3.3 contiene diferentes tipos de columnas las cuales se encuentra relacionadas de acuerdo con el consumo de cada usuario, la cual permite realizar la facturación del servicio por consumo de cada mes

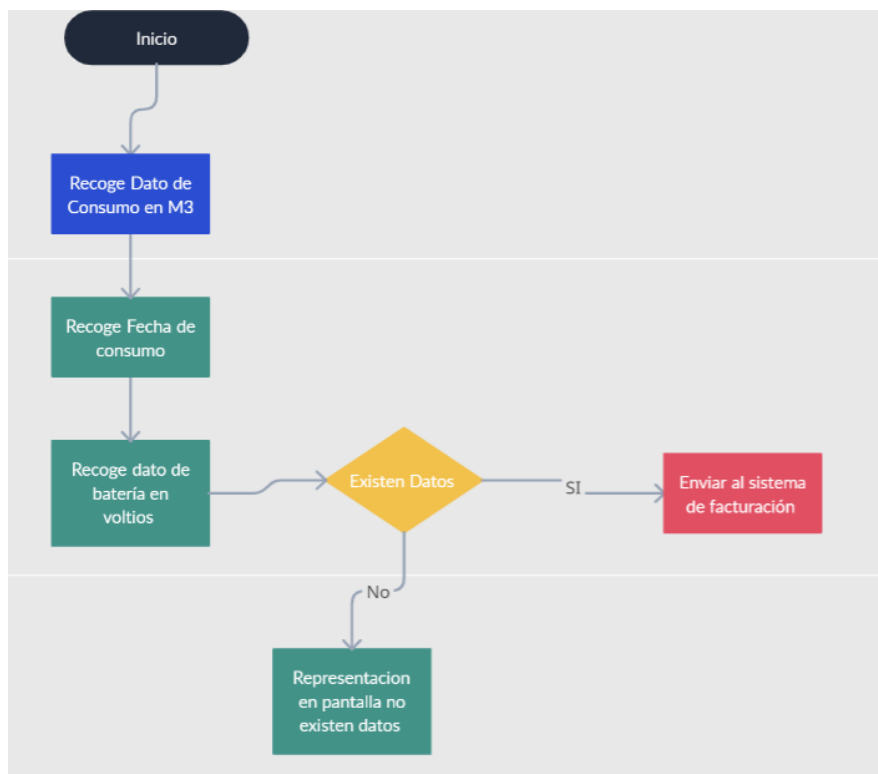


Figura 3.3 Diagrama de flujo de la base de datos

### 3.8 Diagrama de conexión entre la tarjeta ESP32 y el sensor FS400A

El sensor FS400A se comunica a la tarjeta de adquisición a través de tres cables para su funcionamiento y comunicación, en la figura 3.4 se puede observar la función de cada cable que utiliza el sensor [23].

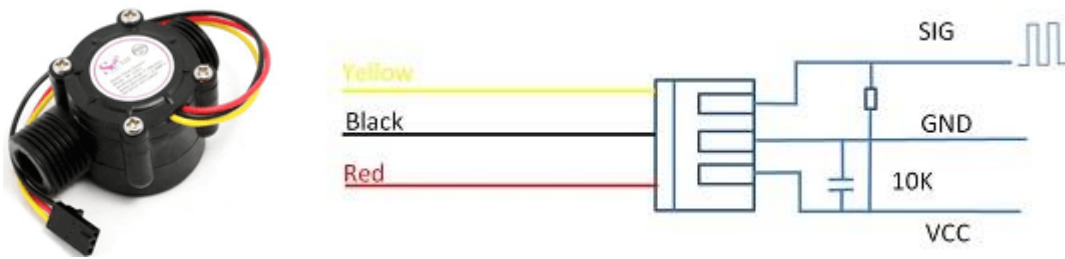


Figura 3.4 Diagrama de FS400A

Según sus características este se conecta a la ESP32 para funcionar por interrupciones, por lo que se conecta como se muestra en la figura 3.5, de igual manera se aprecia todas las conexiones con las que cuenta para poder funcionar, como la alimentación y entradas para usar un timer externo.

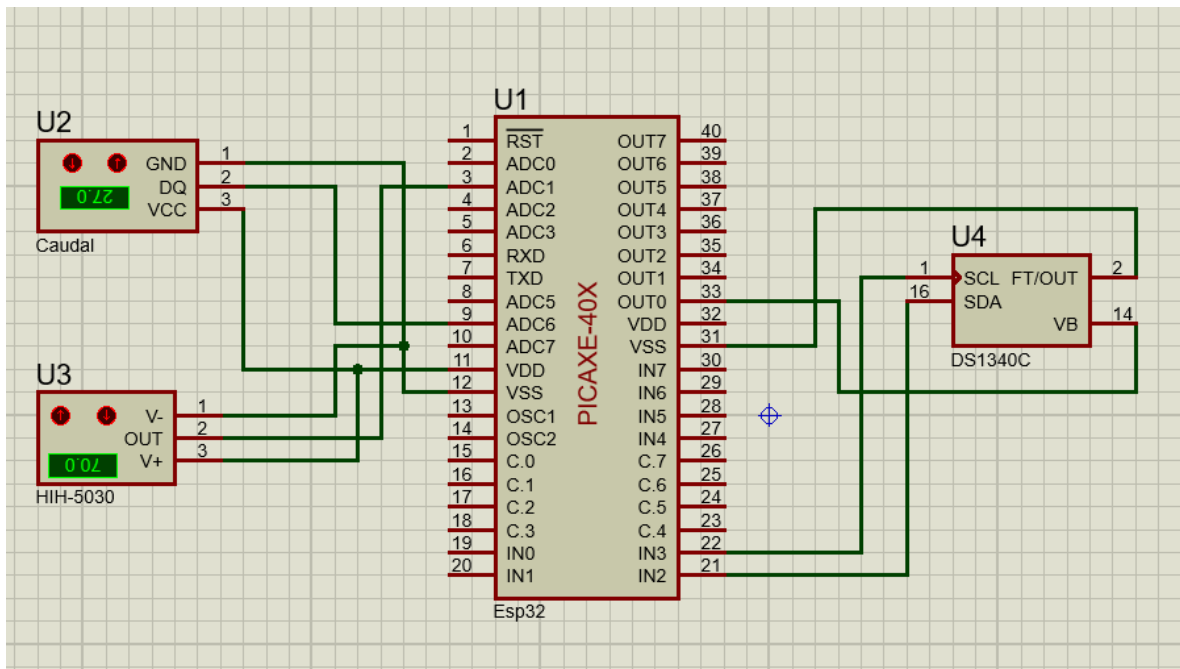


Figura 3.5 Diagrama de conexión entre FS400A y ESP32

### 3.8.1 Medición y registro de datos del sensor a través de ESP32

El sensor FS400A, de acuerdo con el fabricante [24] la fórmula para determinar el flujo en L/min es la siguiente:

$$Caudal = 3.5 * \text{numero de pulsos} \left( \frac{L}{min} \right) * tiempo(s)$$

Formula 1. Determinación de caudal de sensor

De tal manera que el código realizado en la IDE de Arduino que se realizó para determinar el consumo de agua se presenta la figura 3.6, cada pulso que genera el sensor es medido a

través de interrupciones, esto se realizó con ayuda de un tener externo para el cálculo de caudal, de igual manera se determina el ciclo con el que la medición va a ser enviada al servidor.

```
const float factorK = 3.5;
float volume = 0;
long t0 = 0;
float volumen = 0;

void ISRCountPulse()
{
    pulseConter++;
}

float GetFrequency()
{
    pulseConter = 0;

    interrupts();
    delay(measureInterval);
    noInterrupts();

    return (float)pulseConter * 1000 / measureInterval;
}

void SumVolume(float dV)
{
    volume += dV / 60 * (millis() - t0) / 1000.0;
    t0 = millis();
}
}
```

Figura 3.6 Código de prueba de funcionamiento del sensor de flujo.

Para utilizar el timer se necesita la librería RTCLib, una vez importada la librería se debe inicializar el timer, cada vez que se vuelva a poner en marcha la medición como se muestra en la figura 3.7.

```
Serial.begin(115200);
attachInterrupt(digitalPinToInterrupt(sensorPin), ISRCountPulse, RISING);
t0 = millis();
//Inicia el RTC
if (!rtc.begin()) {
    Serial.println(F("Couldn't find RTC"));
    while (1);
}

// Si se ha perdido la corriente, fijar fecha y hora
rtc.begin(); //Inicializamos el RTC
Serial.println("Estableciendo Hora y fecha...");
// rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
// Fijar a fecha y hora específica.
rtc.adjust(DateTime(2021, 1, 21, 4, 57, 10));
```

Figura 3.7 Inicialización del timer RTC

El código presentado en la figura 3.6 genera datos que son transformados de la frecuencia de los pulsos a valor del caudal según el factor de conversión que nos brinda el fabricante la conversión, estos datos son enviados al servidor cada hora para esta aplicación, la implementación de este medidor esta pensada para la comunidad de espejo que cuenta con servicio de internet comunitario, se tiene en cuenta la posibilidad de que ocurran cortes de energía eléctrica de manera inesperada, de tal manera que se implementó una batería de 12 V y 5 A, que garantice el funcionamiento por largos periodos sin una conexión eléctrica, esta batería se encuentra conectada a una celda de carga solar como se muestra en la figura 3.8, la cual ha probado tener autonomía para funcionar durante semanas sin necesidad de una fuente de energía eléctrica doméstica, para soportar las inclemencias del clima de igual manera la

tarjeta de adquisición y el sensor se encuentran protegidos por una carcasa diseñada para esta investigación.



Figura 3.7 Pruebas de campo, (1) cargador solar, (2) Batería, (3) carcasa de protección.

### 3.8.2 Diseño de circuito para baquelita

Debido a que en los programas de diseño de circuitos no se encontraba el modelo de Pin out de la tarjeta ESP32, se optó por diseñar el circuito a mano, para lo que se dibujó los caminos que sigue el circuito para la conexión entre la fuente de energía, la tarjeta, el timer y el sensor de flujo, tal como se muestra en la figura 3.8.



Figura 3.8 Baquelita soldada

### 3.8.2.1 Diseño de carcasa para el circuito impreso

El diseño fue hecho de tal manera que una vez montado los componentes estos quedan de manera fija y segura en una posición dentro de la carcasa tal como se muestra en la figura 3.9, estos se componen por una base donde se asienta la placa y una tapa que cubre todo el circuito y componentes electrónicos.



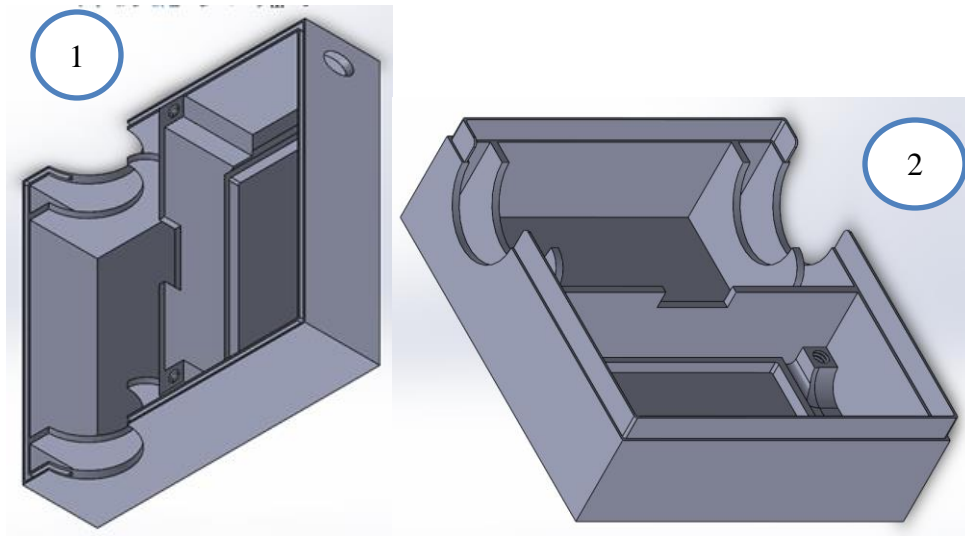


Figura 3.9 CAD de carcasa, (1) parte superior, (2) parte inferior.

Al estar protegidos del exterior se garantiza el correcto funcionamiento ya que esta carcasa se fija de manera segura a la tubería por donde ingresa el agua potable hacia los hogares (figura 3.10).

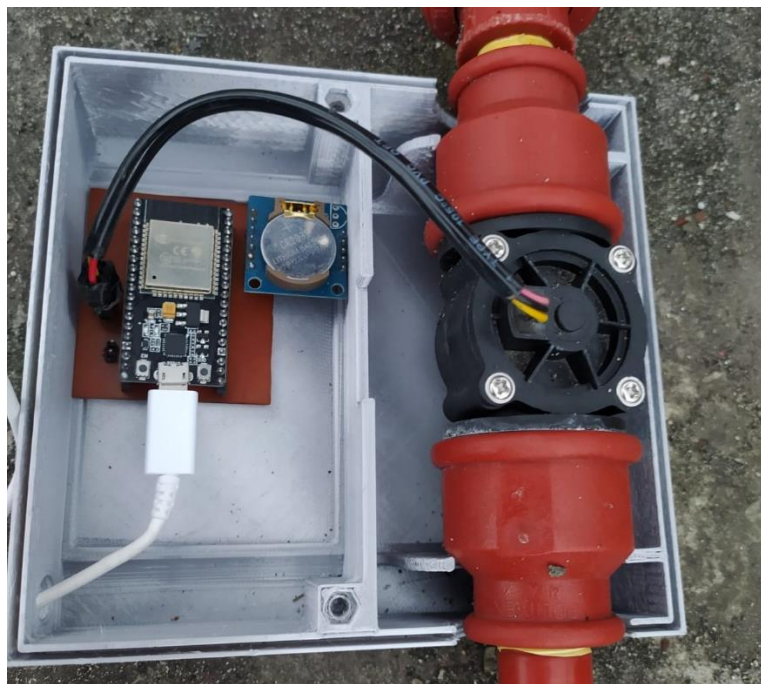


Figura 3.10 Montaje de circuito para pruebas de campo.

### 3.2.2 Creación de base de datos

Se utilizó la herramienta de software Xampp, la cual es de código abierto y brinda los medios para poner en línea una base de datos en MySQL y un servidor de manera local en la comunidad donde se realiza la medición, para el ingreso al servidor se debe activar los servidores como se muestra en la figura 3.11.

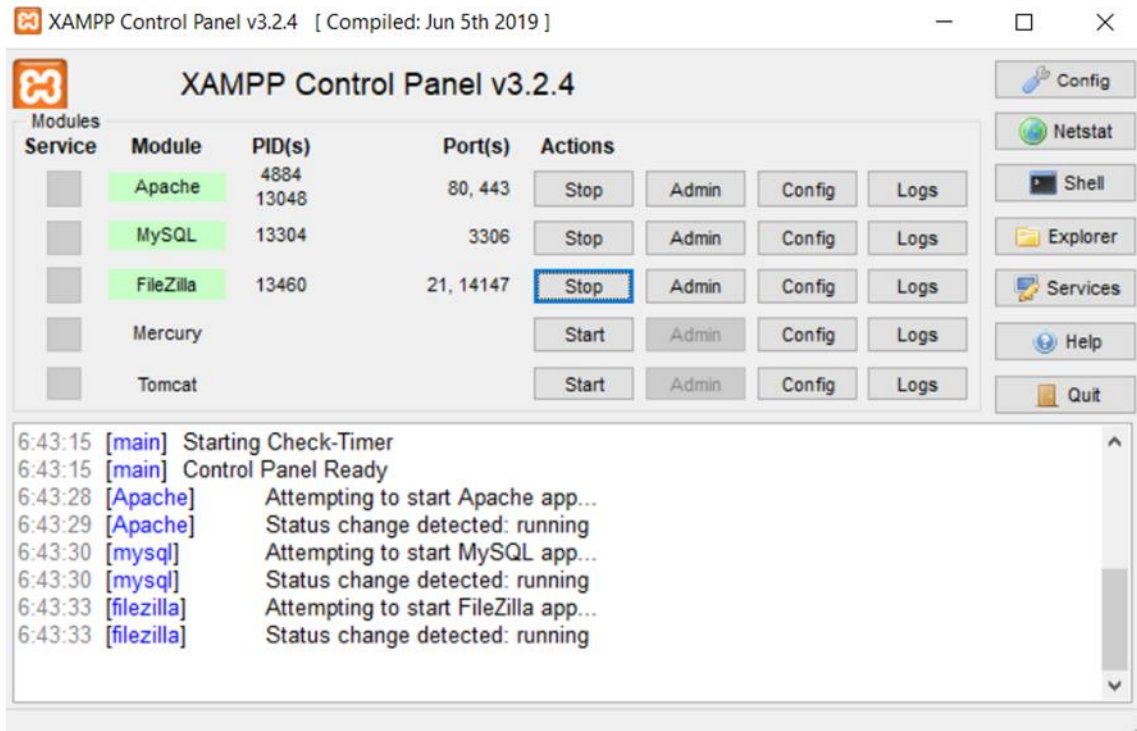


Figura 3.11 Panel de control para activación de servidores

### 3.2.3 Comunicación del módulo ESP con el servidor.

Posterior a la comprobación de funcionamiento se desarrolló el programa (figura 3.12) que realiza la conexión con la red wifi del domicilio donde se realiza la instalación.

```

//Wifi y MySql
const char* ssid = "Gomez2.4G_Ext";
const char* password = "19570911";
byte mac_addr[] = {0xDE, 0xDA, 0xBE, 0xEF, 0xFE, 0xED};
  
```

Figura 3.12 Código de conexión a red wifi local

Una vez ya realizada la conexión con la red wifi esta se conecta a una base de datos creada en PhPMyAdmin el cual permite crear una base de datos de manera local (figura 3.13) que se encuentra en la Junta de Aguas de la comunidad de Espejo, donde los datos sobre el consumo de agua potable son registrados a partir del consumo que se realice en el hogar.

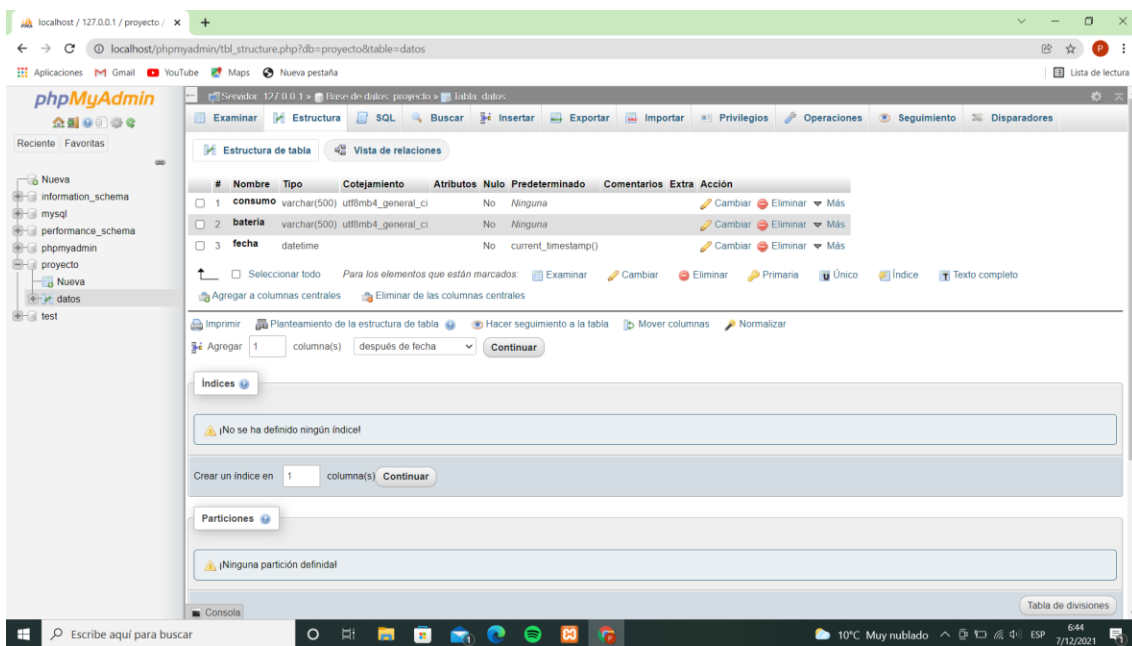


Figura 3.13 Base de datos local de la comunidad Espejo.

Una vez realizado todo el procedimiento para guardar los datos sobre el consumo de agua potable se puede visualizar varias herramientas que ofrece la plataforma PhpMyAdmin, en esta lista se seleccionó la opción phpMyAdmin, en la cual la base de datos fue desarrollada.

Al acceder en la opción seleccionada, se ingresa a la interfaz que ofrece la base de datos, en la figura 3.14 se puede visualizar las diferentes tablas que necesita el proyecto en la base de datos anteriormente creada

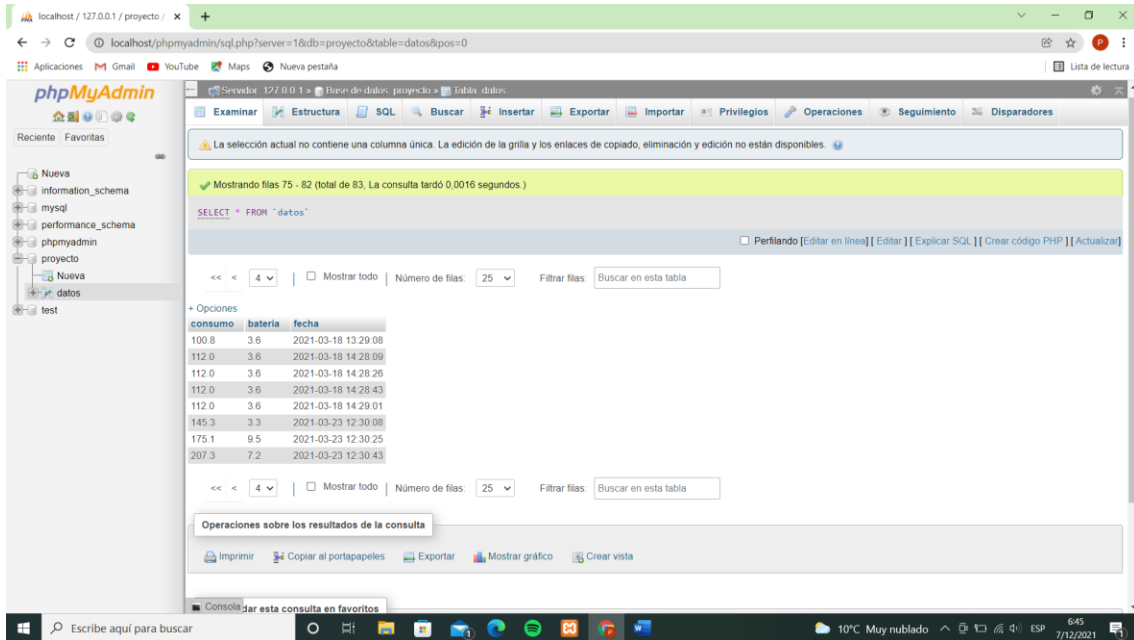


Figura 3.14 Tablas con datos registrados

## **4 Implementación y pruebas de funcionamiento**

Este capítulo contempla la fase de pruebas de funcionamiento donde se documenta todos los datos recopilados, el desempeño de los materiales utilizados, ya que este prototipo se encuentra dirigido para implementación en la comunidad de Espejo, se expondrán los datos recopilados para posteriormente detallar una serie de conclusiones y recomendaciones acerca de la medición realizada con el método que presenta el proyecto. Las pruebas de funcionamiento del prototipo fueron realizadas durante la emergencia sanitaria del Covid-19, debido a las restricciones presentaras por las autoridades, no fue viable realizar las pruebas en la comunidad, por lo que se las realizo en un entorno parecido, con condiciones climáticas similares a las que tiene la comunidad de Espejo, el sistema se mantuvo a la intemperie durante todo el ciclo de pruebas, para garantizar el funcionamiento ya que las condiciones fueron reales y de acuerdo a lo que se sometería el aparato en la comunidad.

### **4.1 Pruebas de medición del sensor de flujo**

El sistema de adquisición de datos fue implementado en el sistema de agua potable de una casa, por lo que es necesario realizar el monje como se muestra en el anexo 1. Ya instalado se verifica el funcionamiento del sensor a través de las lecturas que se pueden apreciar en el puerto serial a través de las herramientas que ofrece el idle de Arduino (figura4.1), para tener las lecturas se utilizó el código mostrado en la figura 4.2

Figura 4.1 Datos obtenidos mostrados a través de visualizador de puerto serial.

```
float frequency = GetFrequency();

// calcular caudal L/min
float flow_Lmin = frequency / factorK;
SumVolume(flow_Lmin);

Serial.print(" Caudal: ");
Serial.print(flow_Lmin, 3);
Serial.print(" (L/min)\tConsumo:");
Serial.print(volume, 1);
Serial.println(" (L)");
```

Figura 4.2 Obtención del volumen o consumo de agua en m<sup>3</sup>

#### 4.1.1 Determinación del error

Para determinar el error podemos comparar lo que el medidor habitual marca en el consumo medido durante la prueba, con el número de pulsos registrados y el volumen calculado por el sistema de adquisición de datos. Para determinar el volumen el sistema de adquisición trabaja conjunto un timer que nos ayuda a la toma de datos y calculo con el numero de pulsos, para determinar el volumen el fabricante del sensor brinda una fórmula con factor de conversión necesario para obtener la información del caudal que mide el sensor, esto se implementa en las líneas de código como muestra la figura 4.3.

```

const float factorK = 3.5;
float volume = 0;
long t0 = 0;
float volumen = 0;

void ISRCountPulse()
{
    pulseConter++;
}

float GetFrequency()
{
    pulseConter = 0;

    interrupts();
    delay(measureInterval);
    noInterrupts();

    return (float)pulseConter * 1000 / measureInterval;
}

void SumVolume(float dV)
{
    volume += dV / 60 * (millis() - t0) / 1000.0;
    t0 = millis();
}

```

Figura 4.3 Conversión de pulsos del sensor a una variable contable

Una vez obtenido las mediciones se compara los registros de las lecturas de flujo como se muestra en la tabla 4.1



**Tabla 4.1** Registro de datos con el valor promedio de pulsos

Metros cúbicos de agua	Valor medido por el sensor	Margen de error
4189	5.0	0
4195.3	6.3	0
4201.9	6.6	0
4207.3	5.4	0
4215.1	7.8	0
4221.8	6.7	0
<b>Error medio</b>		<b>0,0</b>

De la tabla 4.1 se puede concluir con un error medio de 0.0% gráficamente se pueden observar la variación que arrojan las mediciones en la figura 4.4.

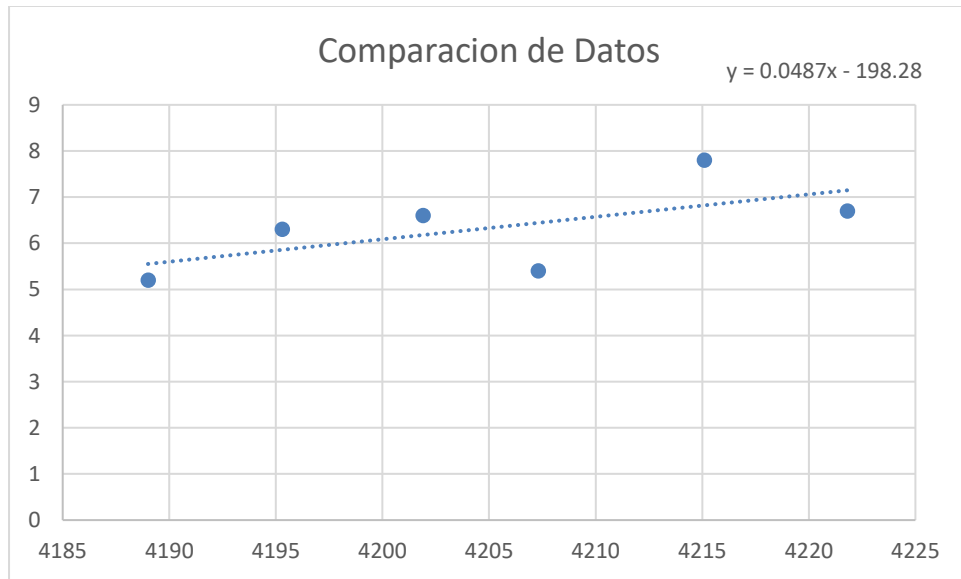


Figura 4.4 Comparación de datos reales con datos medidos con el sensor

### **4.1.2 Datos recopilados en la tarjeta de adquisición**

Una vez que se verifico que la medición se realice de manera correcta, el sistema envía la información de litros totales consumidos por lo que, si existe una hora donde no se logra realizar la verificación de medición este en el próximo envío llega la información de lectura, de esta manera no es imprescindible que todos los datos enviados sean de suma importancia, ya que lo que manda cada hora es la lectura de lo consumido de manera histórica, de manera similar a como funciona en los medidores convencionales

## **4.2 Pruebas de conectividad**

### **4.2.1 Conexión en hogar**

El módulo ESP32 cuenta con varias funcionalidades, es practica al momento de realizar la conexión por Wi-fi, para poder utilizar se necesita la librería en Arduino “Wifi.h”, esta tarjeta puede ser usada como estación o punto de acceso, para esta aplicación se utilizo el modo de estación, este modo asigna una dirección IP a la ESP32 al momento de conectarse a un ruter en cualquier lugar, para poder ingresar a la red wifi se debe guardar en una variable en nombre de la red y la contraseña de la misma como se muestra en la figura 4.5.

```
//Wifi y MySql
const char* ssid = "Gomez2.4G_Ext";
const char* password = "19570911";
byte mac_addr[] = {0xDE, 0xDA, 0xBE, 0xEF, 0xFE, 0xED};
```

Figura 4.5 Variables de ingreso a la red

Se confirmo la conectividad a la red, una vez obtenido la dirección IP que designa el ruter al que se encuentra conectado, se obtuvo la dirección IP a través del monitor serial, el código que se realizó para esta verificación se observa en la figura 4.6.

```
WiFi.begin(ssid, password);
Serial.print("Conectando...");
while (WiFi.status() != WL_CONNECTED){
  delay(500);
  Serial.print(".");
}
Serial.print("Conectado con exito, mi IP es: ");
Serial.println(WiFi.localIP());
```

Figura 4.6 Verificación de asignación de dirección IP

#### 4.2.2 Conexión con servidor local

La base de datos es una terminal que se encuentra de manera local en la comunidad donde se pretende poner a prueba el sistema de medición, por lo que la manera de conectarse a la base de datos es conociendo la dirección IP de la terminal donde se encuentra la base de datos,

para conectarse se realizó las siguientes líneas de código en la idle de Arduino, estas líneas de código guardan los datos del estado de la batería y el consumo del usuario (figura 4.7).

```
IPAddress server_addr(192, 168, 1, 3);
char user[] = "arduino";
char pass[] = "1234";

char INSERT_DATA[] = "INSERT INTO proyecto.datos (consumo, bateria) VALUES (%s, %s)";
char query[128];
char bateria[10];
char consumo[10];

WiFiClient client;
MySQL_Connection conn((Client *)&client);
```

Figura 4.7 Conexión con base de datos a través de dirección IP

### 4.3 Pruebas de funcionamiento de servidor local

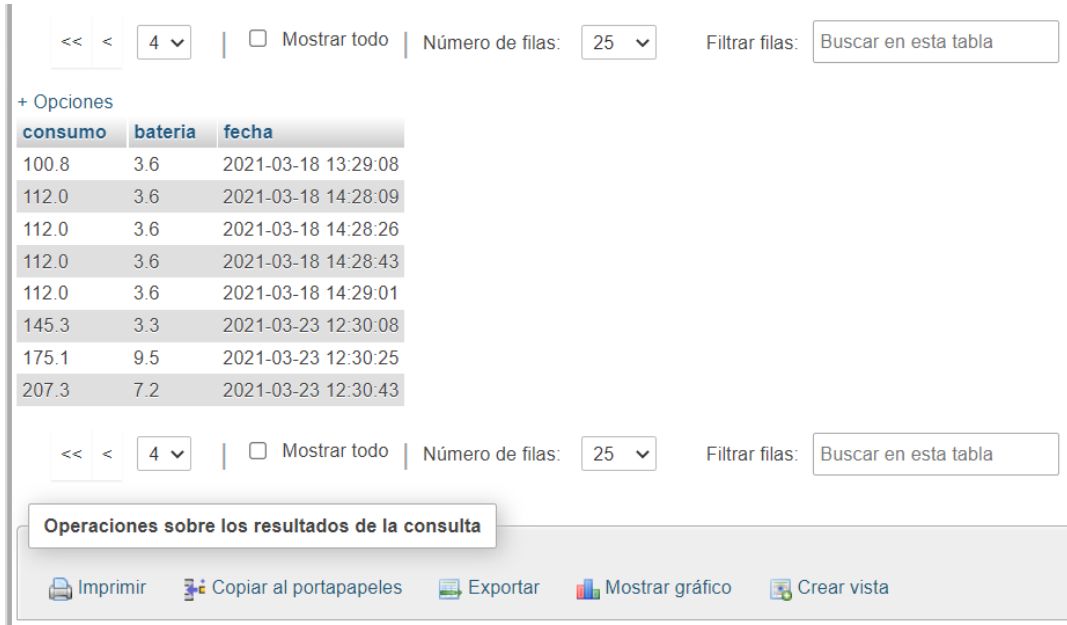
Por seguridad se necesita de un usuario y contraseña al momento de guardar los datos sobre el consumo medido, estas condicionales se programan en la base de datos y en la tarjeta de adquisición de datos se guardan las credenciales necesarias para guardar los datos de lectura (figura 4.8), una vez accedido a la base de datos se escribe los nuevos datos.

```
if (conn.connect(server_addr, 3306, user, pass)){
    delay (10000);

    MySQL_Cursor *cur_men = new MySQL_Cursor(&conn);
```

Figura 4.8 Credenciales para acceder a la base de datos en la prueba de campo.

Una vez validada la conexión con el servidor los datos comienzan a guardarse en la base de datos como se muestra en la figura 4.9, de esta manera se verifica que la conexión con el servidor sea exitosa, los resultados de estas pruebas se pueden mirar en el anexo 2.



The screenshot shows a data table with the following columns: consumo, batería, and fecha. The table contains 8 rows of data. Above the table, there are navigation controls including a dropdown menu set to '4', a checkbox for 'Mostrar todo', a 'Número de filas' dropdown set to '25', and a search box labeled 'Filtrar filas: Buscar en esta tabla'. Below the table, there is another set of navigation controls identical to the one above. At the bottom, there is a toolbar with icons for 'Imprimir', 'Copiar al portapapeles', 'Exportar', 'Mostrar gráfico', and 'Crear vista'.

consumo	batería	fecha
100.8	3.6	2021-03-18 13:29:08
112.0	3.6	2021-03-18 14:28:09
112.0	3.6	2021-03-18 14:28:26
112.0	3.6	2021-03-18 14:28:43
112.0	3.6	2021-03-18 14:29:01
145.3	3.3	2021-03-23 12:30:08
175.1	9.5	2021-03-23 12:30:25
207.3	7.2	2021-03-23 12:30:43

Figura 4.9 datos registrados en la base de datos.

#### 4.4 Prueba de autonomía eléctrica

Para esta prueba se dejó el sistema de adquisición funcionando de manera autónoma conectada a la celda de carga solar y una batería, este sistema en días nublados producía la energía suficiente para mantener la batería al 100% todo el día entre las 8 am y 5 pm (figura 4.10).



Figura 4.10 Controlador de cargador solar.

El nivel de la batería se envía como otra variable que guarda la base de datos, esta se obtiene mediante la tarjeta ESP32, para esto se existe en el código una sección (figura 4.11), que se encarga de medir el nivel de carga de la batería. Esta variable se puede mirar en la base de datos del anexo 2 y la figura 4.9

```
int volt = analogRead(36);  
double voltaje = map(volt, 0, 4096, 0, 1900) + offset;  
voltaje /= 100;  
Serial.print("Voltaje: ");  
Serial.println(voltaje);  
delay(1000);
```

Figura 4.11 Variable de voltaje

#### 4.5 Ubicación de sistema de medición

El servidor local para el proyecto pretende ser instalado en el parque central de la comunidad espejo de Otavalo como se observa en la figura 4.12. Las condiciones climáticas de Ibarra ciudad donde fue probado el dispositivo tiene condiciones climáticas muy parecidas a las que se encuentran en la comunidad de espejo, dada su cercanía y altitud a las que se encuentran. Los dispositivos de medición serían colocados en las casas de los moradores del sector de espejo.

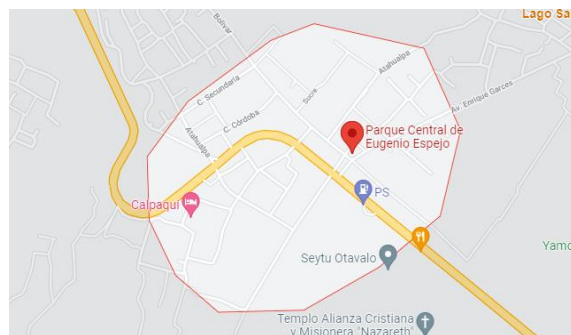


Figura 4.12. Ubicación geográfica de la comunidad de espejo y servidor de base de datos

## **5 Conclusiones y recomendaciones.**

### **5.1 Conclusiones**

El dispositivo desarrollado permite realizar un monitoreo de manera remota en residencias, ya que este se comunica con un servidor local donde se almacenan todos los datos de consumo realizado.

La construcción del dispositivo demostró resistir al exterior y sus factores climáticos, ya que su aplicación esta pensada para pasar largos periodos al exterior de la residencia donde se vaya a implementar el dispositivo, pasando varios meses sin demostrar señales de deterioro en su construcción, permitiendo que el dispositivo realice su medición de manera continua sin problemas.

El prototipo cuenta con un sistema de alimentación de energía solar y una batería que le permite seguir funcionando con una autonomía para poder funcionar por un periodo de 3 semanas sin una fuente que brinde energía de manera continua.

Los datos que arroja el sistema de medición es el volumen consumido ( $m^3$ ) desde el momento que se pone en marcha, por lo que si en la transmisión se pierde alguna medición esta no es significativa ya que en el próximo envío esta va a enviar el consumo realizado hasta ese momento, similar a como se realiza la medición en un medidor de consumo de agua que son utilizados en el cantón Otavalo.



La implementación de estos dispositivos en la comunidad de Espejo brindará una mejora en la automatización en el proceso de medición para la facturación del servicio de agua potable.

## **5.2 Recomendaciones.**

El dispositivo debe ser instalado en exterior, ya que esta pensado en trabajar a la par con el medidor análogo con el que cuenta la empresa publica de agua potable, también cuenta con protecciones que protegen de las inclemencias climáticas a las que podría estar sometido.

Verificar la conexión a través del router del hogar o comunidad en el lugar donde se pretende realizar la instalación del dispositivo para asegurar la comunicación con el servidor.

Se debe instalar el panel solar en el lugar mas alto posible para aprovechar la energía solar y mantener la batería del dispositivo con carga.

Cerrar el paso de agua antes de realizar la instalación del dispositivo de medición, una vez instalado verificar el funcionamiento y conexión con la base de datos.

Una vez instalado el dispositivo asegurarse de que el case encaje para asegurar el circuito y los sensores del exterior.

El proyecto debe ser complementado con un programa que genere la orden de pago y la orden de corte en caso de acumular deudas.

Se debe evitar lugares donde el dispositivo pueda estar sumergido bajo agua, ya que esta protegido contra la lluvia.

## 6 Bibliografía

- [1] P. Cadena, Manual de procedimientos administrativos, contables y financieros para la junta administradora de agua potable y alcantarillado de la cabecera parroquial de Eugenio Espejo, canctón Otavalo, Ibarra, 2015.
- [2] J. Gomez, Base de datos Sistema de factiracion de Eugenio Espejo, Espejo-Ecuador, 2019.
- [3] UNESCO, «Informe de las Naciones Unidas sobre el desarrollo de los recursos hidricos en el mundo,» de *Edisiones Unesco*, Paris, 2015.
- [4] L. S. I. K. E. & M. G. C. Quiroz Fernández, «Aplicación del índice de calidad de agua en el río Portoviejo,» *Ingeniería Hidráulica y Ambiental*, vol. 38, nº 3, pp. 41-51, 2017.
- [5] B. S. G. Teran, Diseño de un sistema inteligente de medicion del consumo de agua potable para la junta de agua "EL DIQUE" de la parroquia San Antonio De Ibarra, Ibarra, 2021.
- [6] P. Rodriguez, «SAAE,» 2010. [En línea]. Available: <http://www.saaeitapetinga.com.br/conhecendo-o-hidrometro/>. [Último acceso: 1 Diciembre 2021].
- [7] J. C. Jinde, ESTUDIO DEL CONSUMO DE AGUA POTABLE EN SECTORES RESIDENCIALES DE LA ZONA CENTRO DE LA CIUDAD DE AMBATO Y SU INCIDENCIA EN LA CURVA DE CONSUMO DIARIO., AMBATO: Universidad Técnica de Ambato, 2017.
- [8] INEC, «Ecuador-Estadistica de Informacion ambiental Economica en Gobiernos Autonomos Descentralizados Minicipales 2016, Agua Potable y Alcantarillado,» 2019. [En línea]. Available: <https://anda.inec.gob.ec/anda/index.php/catalog/693>.
- [9] L. G. Gutiérrez, «Teoría de la medición de caudales y volúmenes de agua e instrumental necesario disponible en el mercado,» Madrid, Tres Cantos.
- [10] M. d. Loja, «www.loja.gob.ec,» [En línea]. Available: [https://www.loja.gob.ec/files/image/dependencias/RegeneraionUrbana/especificaciones\\_tecnicas\\_aapp\\_aass\\_aall\\_ru\\_pc.pdf](https://www.loja.gob.ec/files/image/dependencias/RegeneraionUrbana/especificaciones_tecnicas_aapp_aass_aall_ru_pc.pdf).

- [11 O. a. S. Company, «Lecciones sobre Medidores de Flujo de Desplazamiento Positivo,» *Omega Engineering*.  
]
- [12 B. METERS, «www.medidores.com,» 01 2011. [En línea]. Available:  
] [http://www.medidores.com/pdfs/es\\_88.pdf](http://www.medidores.com/pdfs/es_88.pdf).
- [13 BERMAD, «www.grisa.com.mx,» [En línea]. Available:  
] <http://www.grisa.com.mx/web/images/pdf/BERMAD/Catalogo%20Medidor%20Plastico.pdf>.
- [14 C. Ardilla, «www.carlosardilla.net,» 18 Febrero 2019. [En línea]. Available:  
] <https://www.carlosardilla.net/medidores-de-agua/>.
- [15 D. Semiconductor, «Application Note,» *Dialog Semiconductor*.  
]
- [16 A. R. B. Saravia, «www.microelectrôcash.com,» 2019. [En línea]. Available:  
] [https://www.microelectronicash.com/downloads/ESP32\\_MANUAL.pdf](https://www.microelectronicash.com/downloads/ESP32_MANUAL.pdf).
- [17 H. B. Gonzalez, Diseño y Análisis de un Sistema Basado en un Transceptor Inalámbrico Embebido en una  
] Tabla de Surf, Pannplona: Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación, 2014.
- [18 G. Halfacree, «Raspberry Pi Guía de Usuario,» Inglarerra, Eben Upton.  
]
- [19 Arduino, «Arduino.cc,» 2018. [En línea]. Available: <https://store.arduino.cc/usa/arduino-mkr1000>.  
]
- [20 G. Developers, «Firebase,» Google, [En línea]. Available: <https://firebase.google.com/>.  
]
- [21 S. E. Karen Rosem, «La internet de las cosas - una breve reseña,» Internet Society, 2015.  
]
- [22 M. SAC, «naylampmechatronics,» 2021. [En línea]. Available: <https://naylampmechatronics.com/sensores-liquido/233-sensor-de-flujo-de-agua-1-fs400a.html>.  
]

[23 THAUYO ELECTRONICS , «hotmcu,» [En línea]. Available: <https://www.hotmcu.com/g1-water-flow-sensor-p-312.html>. [Último acceso: 2 12 2021].

[24 MICRO ROBOTICS, «ROBOTICS,» ROBOTICS, [En línea]. Available: <https://www.robotics.org.za/FS400A-G1>. [Último acceso: 2 Diciembre 2021].

[25 J. Cazes, Analytical instrumentation handbook, Marcel Dekker Inc., 2005.  
]

[26 A. Z. Z. D. D. & S. B. Ali, "Open Source Building Science Sensors (OSBSS): A low-cost," *Building and Environment*, pp. 114-126, 2016.

[27 HetPro, «Hetpro Store,» 20 Mayo 2014. [En línea]. Available: <https://hetpro-store.com/TUTORIALES/sensor-de-flujo/>.

[28 T. C. Blanco, «Sistema de medida de temperatura,» Madrid, 2018.  
]

[29 G. Halley, Introducción a Arduino, Bogotá: Complejo Astronómico.  
]

## 7. ANEXOS

### Codigo General del Programa

```
//librerias del RTC

#include <Wire.h>

#include "RTCLib.h"

RTC_DS1307 rtc;

//librerias para conexion wifi y Mysql

#include <MySQL_Connection.h>

#include <MYSQL_Cursor.h>

#include <WiFi.h>

//variables para hora y fecha

int segundo,minuto,hora,dia,mes;

long anio; //variable año

DateTime HoraFecha;

//factor de error para medir el voltaje

int offset = 20;
```

```
//Wifi y MySql

const char* ssid = "Gomez2.4G_Ext";

const char* password = "19570911";

byte mac_addr[] = {0xDE, 0xDA, 0xBE, 0xEF, 0xFE, 0xED};

//sensor de caudal

const int sensorPin = 27;

const int measureInterval = 2500;

volatile int pulseConter;

// FS400A

const float factorK = 3.5;

float volume = 0;

long t0 = 0;

float volumen = 0;

void ISRCountPulse()

{ pulseConter++;}
```

```
float GetFrequency()

{pulseConter = 0

interrupts();

delay(measureInterval);

noInterrupts();

return (float)pulseConter * 1000 / measureInterval;

}

void SumVolume(float dV)

{

volume += dV / 60 * (millis() - t0) / 1000.0;

t0 = millis();

}

IPAddress server_addr(192, 168, 1, 3);

char user[] = "arduino";

char pass[] = "1234";
```



```
char INSERT_DATA[] = "INSERT INTO proyecto.datos (consumo, bateria) VALUES (%s, %s)";

char query[128];

char bateria[10];

char consumo[10];

WiFiClient client;

MySQL_Connection conn((Client *)&client);

void setup() {

    delay(10);

    Serial.begin(115200);

    attachInterrupt(digitalPinToInterrupt(sensorPin), ISRCountPulse, RISING);

    t0 = millis();

    //Inicia el RTC

    if (!rtc.begin()) {

        Serial.println(F("Couldn't find RTC"));

        while (1);

    }

}
```

```
// Si se ha perdido la corriente, fijar fecha y hora

rtc.begin(); //Inicializamos el RTC

Serial.println("Estableciendo Hora y fecha...");

// rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));

// Fijar a fecha y hora específica. En el ejemplo, 21 de Enero de 2016 a las 03:00:00

rtc.adjust(DateTime(2016, 1, 21, 4, 57, 10));

WiFi.begin(ssid, password);

Serial.print("Conectando...");

while (WiFi.status() != WL_CONNECTED){

  delay(500);

  Serial.print(".");

}

Serial.print("Conectado con éxito, mi IP es: ");

Serial.println(WiFi.localIP());

}
```

```
void loop() {  
  
    //Recoje hora y fecha actual  
  
    HoraFecha = rtc.now(); //obtenemos la hora y fecha actual  
  
    //Se divide todo el formato string en caracteres int  
  
    segundo=HoraFecha.second();  
  
    minuto=HoraFecha.minute();  
  
    hora=HoraFecha.hour();  
  
    dia=HoraFecha.day();  
  
    mes=HoraFecha.month();  
  
    anio=HoraFecha.year();  
  
    Serial.print(hora);  
  
    Serial.print(":");  
  
    Serial.println(minuto);  
  
    delay(1000);  
  
    // obtener frecuencia en Hz  
  
    float frequency = GetFrequency();
```

```
// calcular caudal L/min

float flow_Lmin = frequency / factorK;

SumVolume(flow_Lmin);

Serial.print(" Caudal: ");

Serial.print(flow_Lmin, 3);

Serial.print(" (L/min)\tConsumo:");

Serial.print(volume, 1);

Serial.println(" (L)");

int volt = analogRead(36);

double voltaje = map(volt,0,4096, 0, 1900) + offset;

voltaje /=100;

Serial.print("Voltaje: ");

Serial.println(voltaje);

delay(1000);
```

```
if (conn.connect(server_addr, 3306, user, pass)){  
  
    delay (10000);  
  
    MySQL_Cursor *cur_men = new MySQL_Cursor(&conn);  
  
    if (minuto == 59 ){  
  
        if (volume < 1000){  
  
            dtostrf(voltaje, 1, 1, bateria);  
  
            dtostrf(volume, 1, 1, consumo);  
  
            sprintf(query,INSERT_DATA, consumo, bateria);  
  
            cur_men->execute(query);  
  
            delete cur_men;  
  
            Serial.println("Data recorder");  
  
            delay (1800);  
  
        }  
  
        else if (volume > 1000){  
  
            volumen = volume / 100;  
  
            dtostrf(voltaje, 1, 1, bateria);
```

```
dtostrf(volumen, 1, 1, consumo);

sprintf(query,INSERT_DATA, consumo, bateria);

cur_men->execute(query);

delete cur_men;

Serial.println("Data recorder");

delay (1800);

}

}

}

else

Serial.println("connection failed");

}
```

## Imágenes del Prototipo

### Primera instalación



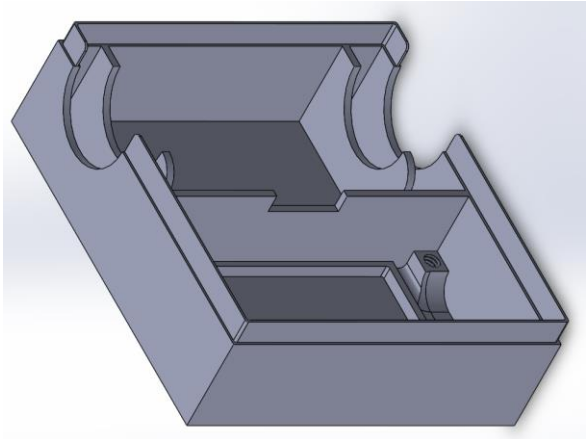
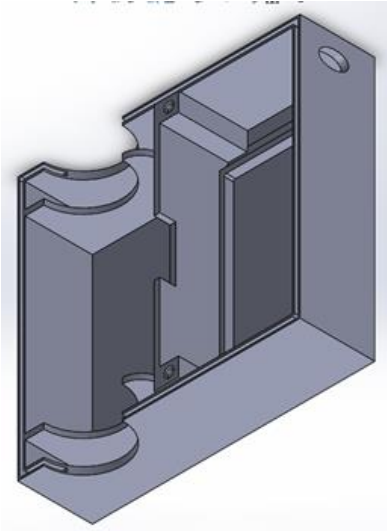








# Carcasa del prototipo



# Base de Datos

The screenshot shows the phpMyAdmin interface for a database named 'proyecto'. The 'datos' table is selected, and its structure is displayed in the 'Estructura de tabla' tab. The table has three columns: 'consumo', 'bateria', and 'fecha'. The 'consumo' and 'bateria' columns are of type 'varchar(500)' with 'utf8mb4\_general\_ci' collation. The 'fecha' column is of type 'datetime' with 'current\_timestamp()' as the default value. The interface also shows options for adding columns, creating indexes, and partitions.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1	consumo	varchar(500)	utf8mb4_general_ci	No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	2	bateria	varchar(500)	utf8mb4_general_ci	No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	3	fecha	datetime		No	current_timestamp()			Cambiar Eliminar Más

localhost / 127.0.0.1 / proyecto / x

localhost/phpmyadmin/sql.php?server=1&db=proyecto&table=datos&pos=0

Aplicaciones Gmail YouTube Maps Nueva pestaña Lista de lectura

phpMyAdmin

Reciente Favoritas

- Nueva
- information\_schema
- mysql
- performance\_schema
- phpmyadmin
- proyecto
  - Nueva
  - datos
- test

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Seguimiento Disparadores

La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copiado, eliminación y edición no están disponibles.

Mostrando filas 75 - 82 (total de 83, La consulta tardó 0.0016 segundos)

```
SELECT * FROM `datos`
```

Perfileando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

<< < 4 |  Mostrar todo | Número de filas: 25 | Filtrar filas:

+ Opciones

consumo	bateria	fecha
100.8	3.6	2021-03-18 13:29:08
112.0	3.6	2021-03-18 14:28:09
112.0	3.6	2021-03-18 14:28:26
112.0	3.6	2021-03-18 14:28:43
112.0	3.6	2021-03-18 14:29:01
145.3	3.3	2021-03-23 12:30:08
175.1	9.5	2021-03-23 12:30:25
207.3	7.2	2021-03-23 12:30:43

<< < 4 |  Mostrar todo | Número de filas: 25 | Filtrar filas:

Operaciones sobre los resultados de la consulta

Consolidar esta consulta en favoritos

Escribe aquí para buscar

10°C Muy nublado 6:45 7/12/2021