



SEP

SECRETARÍA DE  
EDUCACIÓN PÚBLICA

SECRETARÍA DE EDUCACIÓN PÚBLICA  
TECNOLÓGICO NACIONAL DE MÉXICO  
INSTITUTO TECNOLÓGICO DE VERACRUZ.



*Reporte de proyecto: Medidor de Flujo IoT*

**Nombre del alumno:**

Ferat González José Manuel

**Número de control:**

E18020309

**Clave de la materia:**

AEF - 1035

**Clave de grupo:**

EJ21 6F2B

**Carrera:**

Ingeniería Mecatrónica

**Profesor Titular:**

M.C. Paulina Gutiérrez León

H. Veracruz, Ver.

junio de 2021

# CONTENIDO

## MEDIDOR DE FLUJO CON MONITOREO IOT ..... I

### Resumen..... I

### Objetivo ..... I

### Marco teórico .....II

    Arduino ..... II

    Blynk..... II

    Ecuación de la Continuidad..... II

    Open Source ..... III

    Internet de las cosas..... III

    Medidores de flujo ..... IV

    NodeMCU ..... IV

    Programación en lenguaje C..... V

    Visual Studio Code..... V

### Metodología ..... VI

### Cronograma metodológico .....VII

### Materiales y Presupuesto ..... VIII

### Desarrollo ..... 1

    Implementación de Blynk ..... 3

    Pruebas ..... 4

### Resultados.....5

### Conclusiones .....7

### Referencias .....A

### Anexos ..... B

    Anexo 1.0 - Código en VSCode ..... B

    Anexo 1.1 – Video de Youtube ..... C

# MEDIDOR DE FLUJO CON MONITOREO IOT

## RESUMEN

**Problemática:** Para una gran parte de los llamados "países desarrollados", abrir el grifo para beber un poco de agua es una acción tan sencilla que apenas se aprecia. Sin embargo, alrededor de una quinta parte de la población mundial enfrenta escasez de agua. Los informes recopilados por las Naciones Unidas (ONU) durante los últimos diez años así lo demuestran.

Es por eso que surge este proyecto, **medidor de flujo con monitoreo IoT**, que consiste en el monitoreo del consumo de agua en un sistema a través de la tarjeta NodeMCU, con la cual, a través de integración WiFi, podremos monitorear incluso fuera de casa.

Los resultados esperados del proyecto son generar un sistema **open-source**, de bajo costo y funcional, diseñado específicamente con el objetivo de concientizar al usuario sobre su consumo promedio de agua, reduciendo así el gasto neto en los consumidores.

La intención de abrir este proyecto como **open-source** es que quien guste pueda tomar este proyecto, modificarlo e implementarlo, sin limitaciones, simplemente dando crédito al autor.

## OBJETIVO

Implementar la tarjeta de desarrollo NodeMCU y el medidor de flujo YF-S201 en conjunto con las tecnologías móviles para el monitoreo de sistemas independiente a su posición, para la limitación y concientización del cuidado del agua.

## MARCO TEÓRICO

### *Arduino*

Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso.

### *Blynk*

Blynk es un proyecto realizado por personas en Nueva York, Kiev y Varsovia, es una herramienta de trabajo que permite diseñar una interface gráfica para proyectos dibujando y poniendo controles en la misma. Este sistema no está limitado a ningún hardware específico. Solo se requiere que el proyecto con Arduino o Raspberry Pi esté conectado a internet por medio de Wifi, Ethernet o un ESP8266. De esta manera, podrá conectarse con el servidor de Blynk y controlar los sistemas. Los costos en el modo de desarrollo son nulos, permitiendo esto tener un prototipo a un costo reducido.

### *Ecuación de la Continuidad*

La ecuación de continuidad es una variación del principio de conservación de la masa. Las ecuaciones de continuidad general también se pueden expresar en forma diferencial.

Este principio se conoce a menudo como el principio de conservación de la materia y establece que la masa de un objeto o un grupo de objetos nunca cambiará con el tiempo, sin importar cómo se reorganicen los componentes. Este principio se suele utilizar en análisis de fluidos en movimiento.

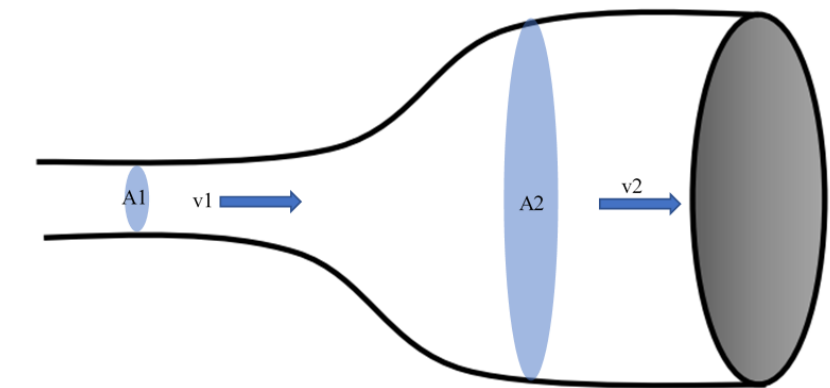
$$Q_1 = Q_2$$

$$A_1 v_1 = A_2 v_2$$

Donde:

A=Área

v=Velocidad



También se puede expresar como:

$$\frac{V_1}{t_1} = \frac{V_2}{t_2}$$

Donde:

V=Volumen

t=Tiempo

### ***Open Source***

Open-Source es un esquema de proyectos y distribución que se compone de dos sectores open-source y open-hardware:

El hardware libre son los dispositivos cuyas especificaciones y diagramas son de acceso público, de manera que cualquiera puede replicarlos. Esto quiere decir que un proyecto ofrece las bases para que cualquier otra persona o empresa pueda crear sus propios diseños, pudiendo ser diferentes entre ellos, pero igualmente funcionales al partir de la misma base.

El software libre son los programas informáticos cuyo código es accesible por cualquiera para que quien quiera pueda utilizarlo y modificarlo.

### ***Internet de las cosas***

EL internet de las cosas o Internet of Things (IoT), es el concepto de conectar cualquier dispositivo a Internet y a otros dispositivos conectados. El IoT es una red gigante de cosas y personas conectadas, todas las cuales recopilan y comparten datos sobre la forma en que se usan y sobre el entorno que las rodea.

Eso incluye un número extraordinario de objetos de todas las formas y tamaños, desde microondas inteligentes, que cocinan automáticamente sus alimentos durante el período de tiempo adecuado, hasta automóviles autónomos, cuyos complejos sensores detectan objetos en su camino, hasta dispositivos de acondicionamiento físico portátiles que miden su ritmo cardíaco y el número de pasos que ha dado ese día, y luego use esa información para sugerir planes de ejercicios. Incluso hay balones de fútbol conectados que pueden rastrear qué tan lejos y rápido se lanzan y registrar esas estadísticas a través de una aplicación para futuros fines de entrenamiento.

Los dispositivos y objetos con sensores incorporados están conectados a una plataforma de Internet de las cosas, que integra datos de los diferentes dispositivos y aplica análisis para compartir la información más valiosa con aplicaciones creadas para abordar necesidades específicas.

### ***Medidores de flujo***

Un medidor de flujo es un componente que mide el flujo de un fluido como un gas o líquido. Los sensores de flujo utilizan subsistemas mecánicos y eléctricos para medir los cambios en los atributos físicos del fluido y calcular su flujo. La medición de estos atributos físicos depende de los atributos físicos del fluido. Los fluidos gaseosos, líquidos y no newtonianos se comportan muy diferentes entre sí, por lo que los métodos que utilizamos para medir su flujo también deben ser diferentes.

Existen varias clasificaciones para los medidores de flujo:

1. Sensores a partir de Presión Diferencial
2. Sensores a partir de Masa Térmica (principalmente usado en gases)
3. Sensores por Desplazamiento Positivo
4. Sensores por Área Variable
5. Sensores por Fuerza Inducida
6. Etc...

### ***NodeMCU***

NodeMCU es un módulo para implementar una plataforma de IoT (Internet of Things), o del Internet de las cosas, de código abierto. Usa un firmware ejecutándose sobre el SoC ESP8266 de Espressif Systems y el hardware que se basa en el módulo ESP-12.

El término NodeMCU se refiere al firmware y no a los kits de desarrollo, aunque últimamente se usa como sinónimo de toda la plataforma completa.

Las aplicaciones para la placa NodeMCU sobre IoT, pueden variar en gran medida, y el límite es la imaginación del programador.

Ejemplos:

- Automatismos conectados, haciendo que ante un impulso se desencadene una acción, como el control de luces LED, conmutación de relés, activación de cualquier tipo de actuador, etc.
- Sistemas de posicionamiento para interiores de viviendas o edificios usando GPS.
- Juguetes de todo tipo, domótica de la casa, etc.

### ***Programación en lenguaje C***

El lenguaje C es un lenguaje para programadores en el sentido de que proporciona una gran flexibilidad de programación y una muy baja comprobación de incorrecciones, de forma que el lenguaje deja bajo la responsabilidad del programador acciones que otros lenguajes realizan por sí mismos.

Características del lenguaje C:

- Programación de nivel medio (beneficiándose de las ventajas de la programación de alto y bajo nivel).
- No depende del hardware, por lo que se puede migrar a otros sistemas.
- Ofrece un control absoluto de todo lo que sucede en el sistema.
- Organización del trabajo con total libertad.
- Los programas son producidos de forma rápida y son bastante potentes.

### ***Visual Studio Code***

Visual Studio Code es un editor de código libre que ayuda al programador a escribir código, ayuda en la depuración y corrige el código. En términos normales, apoya a los usuarios escribir el código de una manera fácil. Podemos describirlo como un entorno de desarrollo y un editor. Cualquier programa/software que veamos o usemos funciona en el código que se ejecuta en segundo plano.

## METODOLOGÍA

- I. **Diseñar el** circuito en el programa de diseño electrónico, el diseño de la placa base se compone, a su vez de subetapas:
  - a. Definir el objetivo del circuito.
  - b. Elegir los componentes adecuados para el objetivo.
  - c. Organizar los componentes en orden lógico de operación.
  - d. Trazado y optimización de la placa.
- II. **Desarrollo de la programación** del sistema en lenguaje C
  - a. Identificar el objetivo.
  - b. Dar solución y propósito al circuito mediante código.
  - c. Compilación del código y depuración de errores.
  - d. Pruebas del código en práctica.
- III. **Unificar** el código, el modelo y el circuito para obtener el producto final.
- IV. **Desarrollar, grabar y editar** el video del proceso de ensamblaje y explicación del funcionamiento.



## CRONOGRAMA METODOLÓGICO

ETAPA	DESCRIPCION	SEMANA 24 – 30 MAYO							SEMANA 31 – 6 JUNIO						
<b>Etapa 1: Diseño de placa</b>	Diseñar la placa de circuito impreso (PCB)														
<b>Etapa 2: Desarrollo del programa y funcionamiento</b>	Desarrollo de la programación del sistema en lenguaje C														
<b>Etapa 3: Unificación del proyecto</b>	Unificar el proyecto entre lo electrónico, lo mecánico y el código.														
<b>Etapa 4: Grabación y edición</b>	Modelado del encapsulado del producto.														

*Figura 1.1 Cronograma de trabajo*

## MATERIALES Y PRESUPUESTO

RECURSO	DESCRIPCIÓN	CANTIDAD	COSTO UNITARIO	COSTO TOTAL
<b>Node MCU</b>	Versión original o genérica	1 unidad	\$200	\$200
<b>Powerbank</b>	-	1 unidad	\$200	\$200
<b>Placa perforada</b>	-	1 unidad	\$20	\$20
<b>Estaño</b>	Infusionado con Flux	1 rollo	\$80	\$80
<b>Cautín</b>	40W	1 unidad	\$120	\$120
<b>Multímetro digital</b>	-	1 unidad	\$300	\$300
<b>Cinta doble cara o velcro</b>	Para anclaje a paredes o ventanas	1 unidad	\$50	\$50
<b>Tubería plástica transparente</b>	Tubería ¾"	30 cm	\$25	\$25
<b>Abrazaderas metálicas</b>	Tubería ¾"	2 unidades	\$12	\$24
<b>Retazo de madera</b>	20 x 20	1 unidad	\$50	\$50
<b>Medidor de flujo</b>	YF-S201	1 unidad	\$150	\$150
<b>Cople para tubería ¾"</b>	Macho-hembra o hembra-macho	1 unidad	\$10	\$10

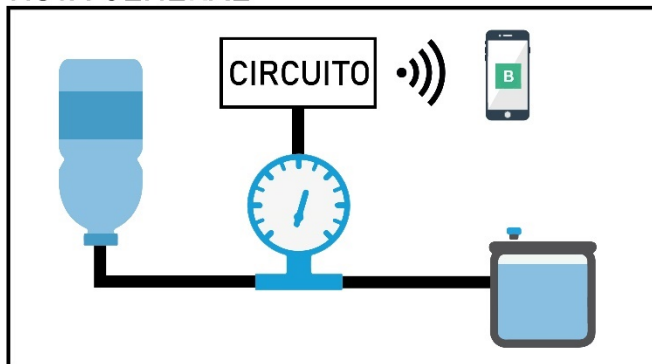
*Figura 1.2 Tabla de costos y materiales*

## DESARROLLO

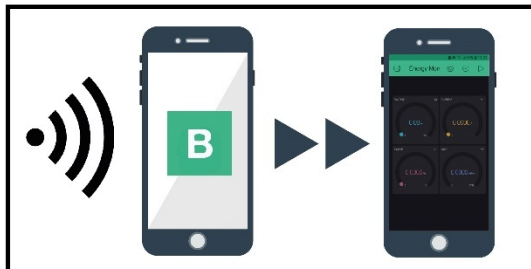
Para desarrollar el proyecto es importante considerar el *objetivo*, **Implementar la tarjeta de desarrollo NodeMCU y el medidor de flujo YF-S201 en conjunto con las tecnologías móviles para el monitoreo de sistemas independiente a su posición, para la limitación y concientización del cuidado del agua**, a partir de aquí es de donde nos basamos para dar el primer paso al proyecto.

Con la idea fresca en nuestras mentes, mi metodología de facto consiste en delimitar punto por punto los requerimientos del proyecto, esto se logra realizando un esquema grosso modo del funcionamiento del sistema:

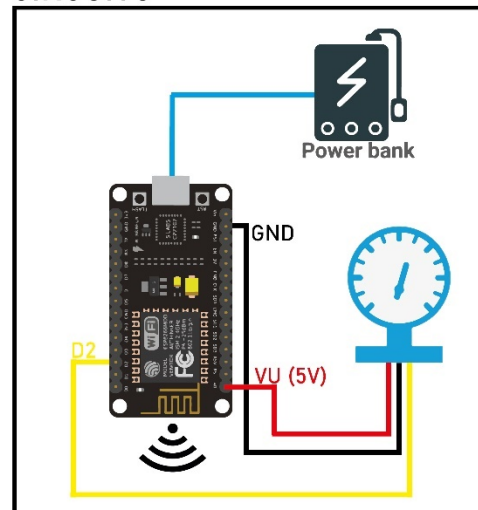
### VISTA GENERAL



### SALIDA MONITOREABLE



### CIRCUITO



*Figura 1.3 Mapa general del proyecto*

Este diagrama lo realicé de manera rápida en Adobe Illustrator y se puede interpretar de la siguiente manera:

**Vista General:** Representa a términos generales el proyecto, contando con una fuente de agua, que puede ser un garrafón, una manguera, una bomba o incluso un sistema de tubería (recordemos que este proyecto se realiza a escala micro, pero está dirigido y es escalable para la escala **macro**, posteriormente el fluido pasará por el medidor de flujo, el cual está conectado al circuito y a un depósito de agua, o a escala macro, el otro punto de la tubería.

**Circuito:** Se compone de una fuente de voltaje, powerbank o batería, según sea la escala (en este proyecto se utiliza una powerbank como fuente), la cual alimenta la NodeMCU, quien se encuentra cargada con el código, dicho código detecta la señal PWM de nuestro medidor de flujo, y a través de WiFi, lo transmite para su indicación y monitoreo en la nube.

**Salida Monitoreable:** Como fines prácticos y de accesibilidad se decidió utilizar un celular, pues es un dispositivo del que todos disponemos y tenemos siempre a la mano, por ello, a través de una red WiFi o Red Móvil podremos monitorear el consumo, flujo y análisis de datos de nuestro sistema de agua, como receptor se utilizará la aplicación Blynk, la cual nos presta sus servidores para la visualización de los datos.

La siguiente etapa es la realización del código, el cual es bastante sencillo, sin embargo, es importante revisar la Datasheet del medidor de flujo, pues suelen venir con ciertos parámetros a considerar, en el caso de este proyecto se utilizó el sensor YF-S201, por lo cual nos dirigimos al datasheet para revisar sus especificaciones:

**Features:**

- Model: YF-S201
- Working Voltage: 5 to 18V DC (min tested working voltage 4.5V)
- Max current draw: 15mA @ 5V
- Output Type: 5V TTL
- Working Flow Rate: 1 to 30 Liters/Minute
- Working Temperature range: -25 to +80?
- Working Humidity Range: 35%-80% RH
- Accuracy:  $\pm 10\%$
- Maximum water pressure: 2.0 MPa
- Output duty cycle: 50%  $\pm 10\%$
- Output rise time: 0.04us
- Output fall time: 0.18us
- Flow rate pulse characteristics:  $\text{Frequency (Hz)} = 7.5 * \text{Flow rate (L/min)}$
- Pulses per Liter: 450
- Durability: minimum 300,000 cycles
- Cable length: 15cm
- 1/2" nominal pipe connections, 0.78" outer diameter, 1/2" of thread
- Size: 2.5" x 1.4" x 1.4"

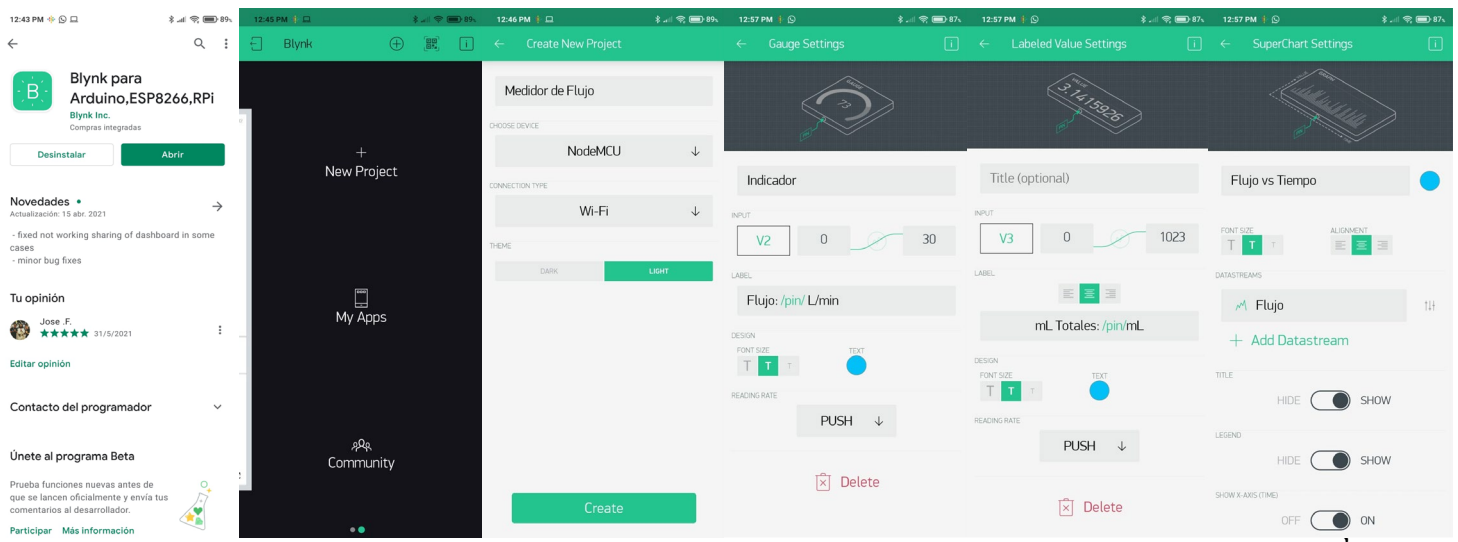
***Figura 1.4 Datasheet resumida YF-S201***

Como se puede observar, se han resaltado las especificaciones más importantes del sensor, especialmente se debe de prestar atención a las características del pulso que emite, pues éste nos proporciona información de campo interpretarlo.

## *Implementación de Blynk*

Blynk es una aplicación que facilita el acceso al IoT a entusiastas de la electrónica, mecatrónica y sistemas, su implementación es realmente sencilla y es una de mis aplicaciones imprescindibles para recomendar a cualquier estudiante de ingeniería, a continuación, explicaré paso a paso como implementarla:

1. Nos dirigimos a la Playstore y descargamos la app de Blynk.
2. La ejecutamos y creamos una cuenta.
3. Creamos un nuevo proyecto.
4. Asignamos un nombre, configuramos para NodeMCU y aceptar.
5. Tocamos en el símbolo + en la esquina superior
6. Insertamos un Gauge, un Label Value y un SuperChart.
7. Los escalamos al tamaño de nuestro agrado.
8. Tocamos sobre el Gauge y configuramos para que vaya de 0 a 30, pues 30 L/min es el límite que soporta nuestro sensor (revisar datasheet), asignamos los sufijos de la variable (L/min) y configuramos el pin como virtual en V2.
9. Tocamos sobre el Labeled Value , asignamos los sufijos de la variable (mL) y configuramos el pin como virtual en V3.
10. Tocamos sobre el Super Chart , asignamos la variable a desplegar y la nombramos Flujo y configuramos el pin como virtual en V2 y lo ponemos en LIVE en la parte inferior.



*Figura 1.5 Secuencia de creación de app*

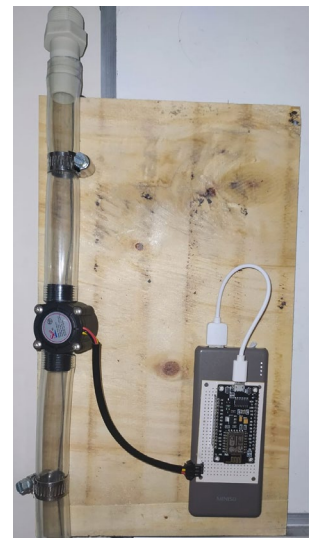


**Figura 1.6** Resultado final de la aplicación en funcionamiento

### ***Pruebas***

Una vez realizado el desarrollo electrónico y la programación del sistema, así como la aplicación celular, podemos armar el proyecto, debido a la sencillez del proyecto únicamente colocamos el medidor entre dos pedazos de tubería, verificando que la dirección del flujo sea la especificada en la bomba (incluye una flecha de flujo en la parte trasera), conectamos el medidor a la NodeMCU y conectamos la Node a la powerbank (ya con el código subido).

La **Figura 1.6** muestra el resultado de la prueba realizada durante 10 segundos, verificando que, en efecto, funciona adecuadamente, y podemos continuar con el proyecto.

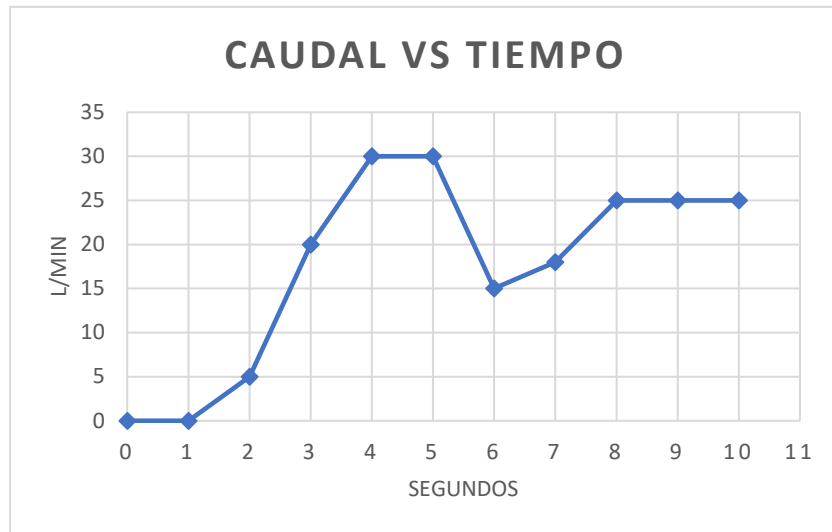


**Figura 1.7** Prototipo no final

## RESULTADOS

A través de un análisis de datos se consiguió exportar a Excel las mediciones del sensor, permitiéndonos observar con detenimiento los datos, y modelar una gráfica que nos transmite claramente toda la información recabada, los datos son los siguientes:

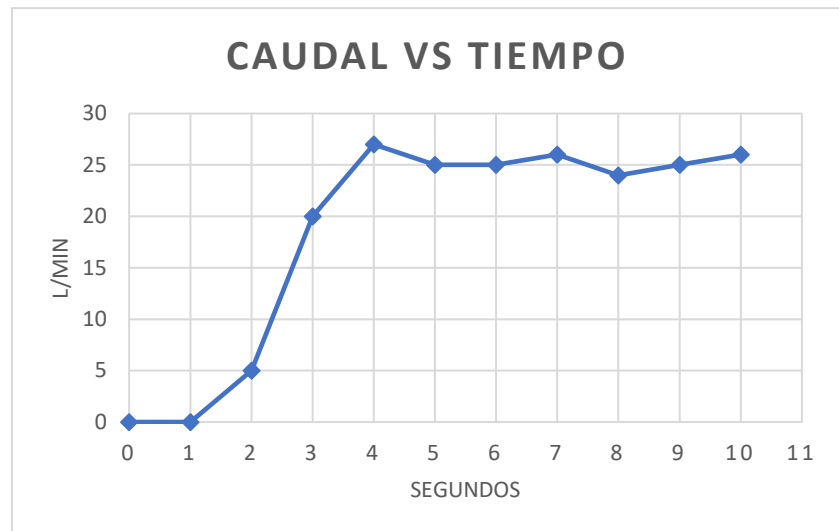
<i>Tiempo (s)</i>	<i>L/min</i>
0	0
1	0
2	5
3	20
4	30
5	30
6	15
7	18
8	25
9	25
10	25



**Figura 1.8** Tabla y gráfica del sistema con entrada variable

Como se puede observar, en este caso las mediciones se encuentran algo dispersa, usualmente este no es el caso para un medidor de flujo, pues usualmente el flujo es constante, sin embargo, en este caso se le proporcionó un flujo variable, repitámoslo con un flujo constante:

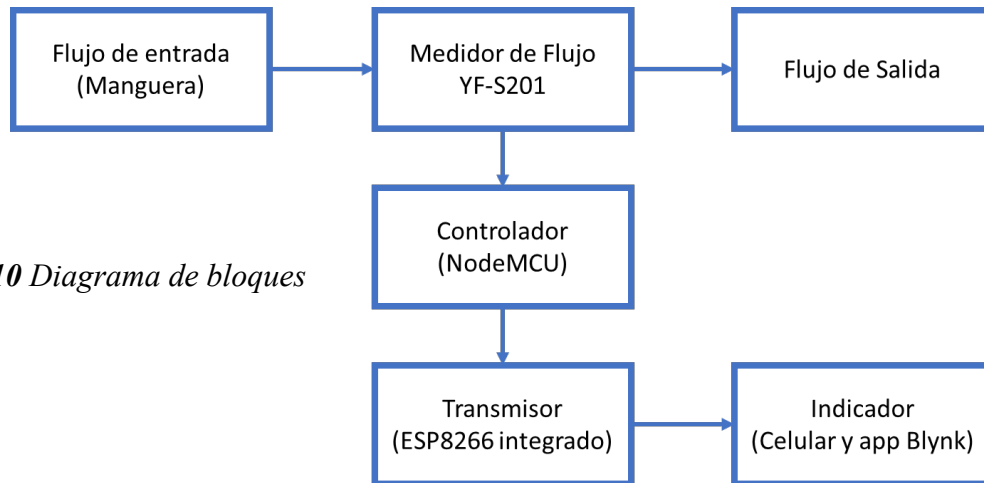
<i>Tiempo (s)</i>	<i>L/min</i>
0	0
1	0
2	5
3	20
4	27
5	25
6	25
7	26
8	24
9	25
10	26



**Figura 1.9** Tabla y gráfica del sistema con entrada constante

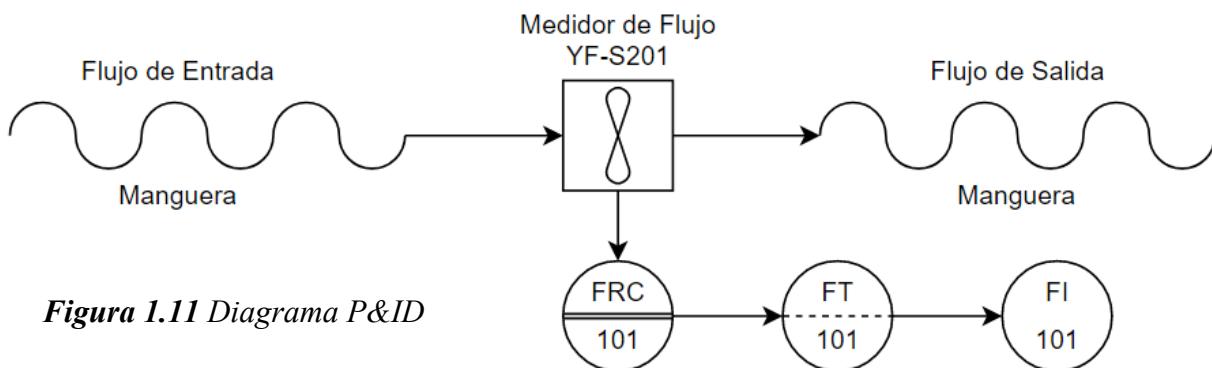
De esta manera podemos apreciar claramente que teniendo un flujo medianamente constante la señal se aplana mucho, recordemos que existen ciertos porcentajes de error, por lo que es normal que existen pequeñas fluctuaciones en la medición, esto aunado a que el flujo de entrada no es perfecto culmina en la gráfica anterior.

Podemos resumir el funcionamiento Entrada → Proceso → Salida en un diagrama de bloques,



**Figura 1.10** Diagrama de bloques

Como se puede observar, el diagrama de la **figura 1.10** se asemeja mucho a un diagrama P&ID, podemos realizarlo de la siguiente manera:



**Figura 1.11** Diagrama P&ID

Resumiendo la **figura 1.11**, tenemos una entrada de flujo en la manguera principal, la manguera lleva el fluido hasta nuestro medidor, el cual, a través de una turbine y un sensor magnético, mide las revoluciones y genera una señal PWM, la cual se envía al NodeMCU (FRC-101), la cual interpreta la señal y la registra para manipularla e interpretarla en datos matemáticos, una vez procesados, los datos se envían a través del ESP8266 (FT-101) por WiFi, llegando los datos al FI-10, es decir, nuestro celular.



## CONCLUSIONES

A través de este proyecto se logró concluir exitosamente con el diseño y operación de un sistema monitor de flujo IoT, este proyecto puede ser replicado a diversas magnitudes que abarcan desde pequeños proyectos hasta aplicaciones industriales o sanitarias.

El sensor YF-S201 nos permite trabajar con muchos tipos de líquidos, en general se puede trabajar con líquidos viscosos y no viscosos, sin embargo la datasheet nos advierte que existirá cierto grado de error es importante tenerlo en cuenta, con base en pruebas realizadas recomendando únicamente trabajar líquidos no abrasivos o corrosivos, pues la construcción del sensor es plástica y podría dañarlo, también, recomendando trabajar principalmente con líquidos no viscosos, , pues recordemos que al ser un método mecánico de medición (a través de aspas) la viscosidad podría presentar problemas.

Conocer los distintos tipos de sensores que existen es muy importante para un ingeniero mecatrónico, no es necesario dominarlos todos, sin embargo, es importante conocerlos, pues de esta manera sabremos cuando es momento de implementarlos de manera eficaz.

Es importante recordar que el proyecto pretende concientizar al usuario sobre su consumo, sin embargo no se implementó ningún sistema de control para que el mismo sistema regule el flujo de agua basado en el consumo promedio, siendo esta un área de oportunidad para una futura revisión del proyecto, implementando inteligencia artificial para que con base en el comportamiento habitual del usuario, regule a lo largo del día el flujo y consumo del agua, entre más gente y más tiempo se use, más inteligente se hará el sistema, por consiguiente ser á más eficiente.



*Figura 1.12 Producto Final*

## REFERENCIAS

- All Data Sheet. (s.f.). *alldatasheet.com*. Obtenido de <https://pdf1.alldatasheet.com/datasheet-pdf/view/1221259/ETC1/YF-S201.html>
- Blynk. (s.f.). *Blynk Documentation*. Obtenido de <https://docs.blynk.cc/>
- Campillo, S. C. (09 de Abril de 2018). *Xataka*. Obtenido de <https://www.xataka.com/medicina-y-salud/el-gran-problema-del-agua>
- Clark, J. (17 de Noviembre de 2016). *IBM*. Obtenido de <https://www.ibm.com/blogs/internet-of-things/what-is-the-iot/#:~:text=In%20a%20nutshell%2C%20the%20Internet%20of%20Things%20is,are%20used%20and%20about%20the%20environment%20around%20them.>
- Instituto Politecnico Nacional. (s.f.). *IPN*. Obtenido de <https://www.cecyl7.ipn.mx/assets/files/cecyl7/docs/estudiantes/apoyo-estudiantil/material-de-apoyo/guias/basicas/fisica/tm/Ecuacion%20Continuidad.pdf>
- UN Water. (2020). *UNWater.org*. Obtenido de <https://www.unwater.org/water-facts/scarcity/>

## ANEXOS

### *Anexo 1.0 - Código en VSCode*

```
//Definición de funciones y librerías
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// Credenciales WiFi
char auth[] = "Inserte el código de Blynk";
char ssid[] = "Inserte el SSID de su red"; //
char pass[] = "Inserte la clave de su red";

//Declaración de variables y funciones
#define SENSOR D2
long Millis = 0;
long ant_Millis = 0;
int intervalo = 1000;
boolean ledState = LOW;
float FactorDeFlujo = 7.5;
volatile byte Contador;
byte pulse1Sec = 0;
float Flujo;
unsigned int ml_fluidos;
unsigned long total_mL;

//Función reloj
void IRAM_ATTR Reloj()
{
  Contador++;
}

void setup()
{
  //Inicializa las variables del programa
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  pinMode(SENSOR, INPUT_PULLUP);
  Contador = 0;
  Flujo = 0.0;
  ml_fluidos = 0;
  total_mL = 0;
  ant_Millis = 0;
  attachInterrupt(digitalPinToInterrupt(SENSOR), Reloj, FALLING);
}
```

```

}

void loop()
{
    Millis = millis();
    if (Millis - ant_Millis > intervalo) {
        pulse1Sec = Contador;
        Contador = 0;
        Flujo = ((1000.0 / (millis() - ant_Millis)) * pulse1Sec) / FactorDeFlujo
;

        ant_Millis = millis();
        ml_fluidos = (Flujo / 60) * 1000;
        total_mL += ml_fluidos;

        //Despliega los valores calculados a los servidores de Blynk
        Serial.print("Flow Rate: ");
        Serial.print(int(Flujo));
        Serial.print("L/min");
        Serial.print("\t");
        Serial.print("Output Liquid Quantity: ");
        Serial.print(total_mL);
        Serial.print("mL / ");
        Serial.print(total_mL / 1000);
        Serial.println("L");
        //Escribe en los pines virtuales de Blynk para su interpretación en gráficos
        Blynk.virtualWrite(V2, Flujo);
        Blynk.virtualWrite(V3, total_mL);
        Blynk.virtualWrite(V4, Millis*1000);
        Blynk.run();
    }
}

```

### *Anexo 1.1 – Video de Youtube*

[Hipervínculo](#)

