

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

### **IMPLEMENTACIÓN DE MÓDULOS DIDÁCTICOS PARA MEDICIÓN REMOTA DE PRESIÓN Y CAUDAL EN EL LTI DE LA ESFOT**

#### **TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN ELECTRÓNICA Y TELECOMUNICACIONES**

**Roger Paúl Morejón Herrera**

roger.morejon@epn.edu.ec

**Jonathan Patricio Villagómez Villamarín**

jonathan.villagomez@epn.edu.ec

**DIRECTOR: ING. ALAN DANIEL CUENCA SÁNCHEZ MSc.**

alan.cuenca@epn.edu.ec

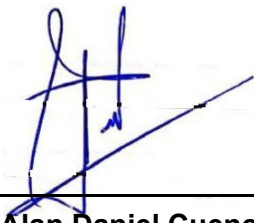
**CODIRECTOR: ING. FABIO MATÍAS GONZÁLEZ GONZÁLEZ MSc.**

fabio.gonzalez@epn.edu.ec

**Quito, febrero 2022**

# CERTIFICACIÓN

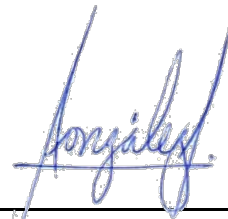
Certificamos que el presente trabajo fue desarrollado por los señores Roger Paúl Morejón Herrera y Jonathan Patricio Villagómez Villamarín como requerimiento parcial a la obtención del título de TECNÓLOGO EN ELECTRÓNICA Y TELECOMUNICACIONES, bajo nuestra supervisión:



---

**Ing. Alan Daniel Cuenca  
Sánchez**

DIRECTOR DEL PROYECTO



---

**Ing. Fabio Matías González  
González**

CODIRECTOR DEL PROYECTO

## DECLARACIÓN

Nosotros Morejón Herrera Roger Paúl con CI: 1718267295 y Villagómez Villamarín Jonathan Patricio con CI: 1723565527 declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

Sin perjuicio de los derechos reconocidos en el primer párrafo del artículo 144 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación – COESC-, somos titulares de la obra en mención y otorgamos una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional.

Entregamos toda la información técnica pertinente, en caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.



---

Roger Morejón

ESTUDIANTE



---

Jonathan Villagómez

ESTUDIANTE

## **DEDICATORIA**

Dedico con todo mi corazón y cariño el presente trabajo a mi madre, pues sin ella no lo habría logrado, su oración y bendición a lo largo de mi vida me han protegido y guiado por el camino del bien, en ofrenda a esto, por su paciencia y amor de madre, le doy este y los futuros logros de mi vida.

**Roger Morejón**

# **AGRADECIMIENTO**

Todo mi agradecimiento para mi madre y mi tío, fueron y siguen siendo los ejes fundamentales en los que hoy se asientan los logros de mi vida, su sabiduría, cariño, comprensión me han permitido alcanzar destrezas que hoy las pongo a prueba en el largo camino a recorrer que me espera, siempre estaré agradecido y los guardo en lo más profundo de mi corazón y mente.

Agradezco también a mi institución en donde he desarrollado habilidades y proyectos de los cuales no sabía podía realizar, mi universidad me ha dado tanto que espero algún día devolverle lo atribuido.

**Roger Morejón**

# **DEDICATORIA**

Dedico el presente trabajo a las presentes y futuras generaciones de la Escuela de Formación de Tecnólogos para que con mi aporte puedan instruirse y así ampliar sus conocimientos y habilidades en telemetría.

**Jonathan Villagómez**

# **AGRADECIMIENTO**

En primer lugar, agradezco a mis padres por brindarme el apoyo y la fuerza necesaria para poder culminar mis estudios.

A mi tío por ser como mi segundo padre y siempre creer en mí a pesar de los fallos cometidos en el largo camino.

A mis amigos por ser unos de los pilares importantes y haberme dejado contar con su ayuda incondicional.

A mis maestros por los conocimientos que fueron impartidos a través de los años.

Y, por último, a la Escuela Politécnica Nacional por tantos recuerdos acogidos en sus instalaciones.

**Jonathan Villagómez**

# ÍNDICE DE CONTENIDOS

1	INTRODUCCIÓN .....	15
1.1	Objetivo general.....	16
1.2	Objetivos específicos.....	16
1.3	Fundamentos .....	16
	<i>WeMos D1 mini ESP8266</i> .....	16
	<i>Cayenne myDevices</i> .....	17
2	METODOLOGÍA.....	18
2.1	Descripción de la metodología usada .....	18
3	RESULTADOS Y DISCUSIÓN .....	20
3.1	Requerimientos.....	20
	Instrumentos de medición de caudal y presión.....	20
	Diseño de los módulos.....	23
3.2	Algoritmos para registro de datos.....	27
	Lectura de datos Arduino NANO .....	27
	Envío de datos al servidor <i>Cayenne</i> .....	32
	Interfaz de <i>Cayenne myDevices</i> .....	37
3.3	Construcción .....	39
	Diagrama esquemático .....	39
	Adquisición de materiales .....	42
	Implementación .....	46
3.4	Pruebas y Análisis de Resultados.....	53
	Pruebas de acondicionamiento.....	53
	Pruebas de transmisión de datos .....	58
	Pruebas de almacenamiento de datos .....	59
3.5	Manual de Uso.....	60
4	CONCLUSIONES Y RECOMENDACIONES .....	61
4.1	Conclusiones.....	61



4.2	Recomendaciones .....	62
5	REFERENCIAS BIBLIOGRÁFICAS .....	63
	ANEXOS .....	65
	Anexo 1: Certificado de Funcionamiento .....	i
	Anexo 2: Planos y Esquemas .....	iii
	Anexo 3: Especificaciones Técnicas .....	iv

# ÍNDICE DE FIGURAS

<b>Figura 1.1</b> WeMos D1 mini .....	17
<b>Figura 1.2</b> Plataforma <i>Cayenne myDevices</i> .....	17
<b>Figura 3.1</b> Transmisor de presión .....	20
<b>Figura 3.2</b> Transmisores de presión LTI-ESFOT .....	21
<b>Figura 3.3</b> Circuito de conversión de corriente a voltaje.....	21
<b>Figura 3.4</b> Módulo de caudal .....	22
<b>Figura 3.5</b> Módulos de caudal del LTI-ESFOT .....	22
<b>Figura 3.6</b> Esquema de la caja (módulo de presión).....	24
<b>Figura 3.7</b> Esquema de la caja (módulo de caudal) .....	24
<b>Figura 3.8</b> Disposición de los dispositivos vista superior (módulo de presión).....	25
<b>Figura 3.9</b> Disposición de los dispositivos vista superior (módulo de caudal) .....	25
<b>Figura 3.10</b> Vista superior de uno de los módulos de presión .....	26
<b>Figura 3.11</b> Vista frontal de uno de los módulos de presión .....	26
<b>Figura 3.12</b> Vista posterior de uno de los módulos de presión .....	26
<b>Figura 3.13</b> Vista superior de uno de los módulos de caudal.....	27
<b>Figura 3.14</b> Vista frontal de uno de los módulos de caudal.....	27
<b>Figura 3.15</b> Declaración de variables (módulo de caudal) .....	28
<b>Figura 3.16</b> Declaración de variables (módulo de presión).....	28
<b>Figura 3.17</b> Parte “void setup()” .....	28
<b>Figura 3.18</b> Inicio de lectura de datos y acondicionamiento parte “void loop()”.....	29
<b>Figura 3.19</b> Acondicionamiento de los valores de presión .....	30
<b>Figura 3.20</b> Acondicionamiento de los valores de caudal .....	31
<b>Figura 3.21</b> Función “Serial.println()”, datos de presión .....	31
<b>Figura 3.22</b> Función “Serial.println()”, datos de caudal.....	31
<b>Figura 3.23</b> Diagrama de flujo del código en Arduino NANO .....	32
<b>Figura 3.24</b> Declaración constantes y variables, librerías a usar (variable caudal)....	33
<b>Figura 3.25</b> Declaración constantes y variables, librerías a usar (variable presión)....	33
<b>Figura 3.26</b> Autenticación de la red y conexión con el servidor <i>Cayenne</i> .....	34
<b>Figura 3.27</b> Ejecución de <i>Cayenne</i> (datos del transmisor de presión 1) .....	34
<b>Figura 3.28</b> Ejecución de <i>Cayenne</i> (datos del transmisor de presión 2) .....	34
<b>Figura 3.29</b> Ejecución de <i>Cayenne</i> (datos del transmisor de presión 3) .....	35
<b>Figura 3.30</b> Ejecución de <i>Cayenne</i> (datos del transmisor de presión 4) .....	35

<b>Figura 3.31</b> Ejecución de <i>Cayenne</i> (datos del módulo de caudal A) .....	35
<b>Figura 3.32</b> Ejecución de <i>Cayenne</i> (datos del módulo de caudal B) .....	36
<b>Figura 3.33</b> Ejecución de <i>Cayenne</i> (datos del módulo de caudal C) .....	36
<b>Figura 3.34</b> Ejecución de <i>Cayenne</i> (datos del módulo de caudal D) .....	36
<b>Figura 3.35</b> Diagrama de flujo del programa en el WeMos D1 mini ESP8266 .....	37
<b>Figura 3.36</b> Formulario de registro en <i>Cayenne</i> .....	38
<b>Figura 3.37</b> Interfaz web de <i>Cayenne</i> .....	38
<b>Figura 3.38</b> Interfaz de aplicación móvil de <i>Cayenne</i> .....	39
<b>Figura 3.39</b> Inicio de sesión en <i>Cayenne</i> .....	39
<b>Figura 3.40</b> Diagrama electrónico (módulo de presión).....	40
<b>Figura 3.41</b> Diagrama electrónico (módulo de caudal).....	41
<b>Figura 3.42</b> Arduino Nano .....	42
<b>Figura 3.43</b> WeMos D1 mini ESP8266 .....	43
<b>Figura 3.44</b> Potenciómetros de precisión.....	43
<b>Figura 3.45</b> Condensadores de 1 ( $\mu$ F).....	44
<b>Figura 3.46</b> Borneras tipo banana.....	44
<b>Figura 3.47</b> Conector Power Jack .....	45
<b>Figura 3.48</b> Fuente de 24 ( $V_{DC}$ ).....	45
<b>Figura 3.49</b> Caja de acrílico para el módulo de caudal .....	46
<b>Figura 3.50</b> Caja de acrílico para el módulo de presión .....	46
<b>Figura 3.51</b> Diseño de la PCB para módulo de caudal.....	47
<b>Figura 3.52</b> Placas para los módulos de caudal.....	47
<b>Figura 3.53</b> Vista frontal y posterior de la placa del módulo de caudal.....	48
<b>Figura 3.54</b> Caja de acrílico con el módulo de caudal.....	48
<b>Figura 3.55</b> Módulo de caudal etiquetado.....	49
<b>Figura 3.56</b> Módulos de medición remota de caudal finalizados .....	49
<b>Figura 3.57</b> Diseño de PCB para módulo de presión .....	50
<b>Figura 3.58</b> Placas para los módulos de presión.....	50
<b>Figura 3.59</b> Vista frontal y posterior de la placa del módulo de presión .....	51
<b>Figura 3.60</b> Caja de acrílico con el módulo de presión.....	51
<b>Figura 3.61</b> Módulo de presión etiquetado .....	52
<b>Figura 3.62</b> Módulos de medición remota de presión finalizados .....	52
<b>Figura 3.63</b> Datos del módulo de caudal A.....	53
<b>Figura 3.64</b> Datos del módulo de caudal B.....	53
<b>Figura 3.65</b> Datos del módulo de caudal C.....	54
<b>Figura 3.66</b> Datos del módulo de caudal D.....	54

<b>Figura 3.67</b> Datos del módulo de presión 1 .....	56
<b>Figura 3.68</b> Datos del módulo de presión 2 .....	56
<b>Figura 3.69</b> Datos del módulo de presión 3 .....	56
<b>Figura 3.70</b> Datos del módulo de presión 4 .....	57
<b>Figura 3.71</b> Código QR de las pruebas de funcionamiento.....	58
<b>Figura 3.72</b> Almacenamiento de datos de presión .....	59
<b>Figura 3.73</b> Almacenamiento de datos de caudal.....	60
<b>Figura 3.74</b> Código QR del manual de uso de módulo de caudal.....	60
<b>Figura 3.75</b> Código QR del manual de uso de módulo de presión .....	60

## ÍNDICE DE TABLAS

<b>Tabla 3.1</b> Datos obtenidos para el módulo A de caudal .....	54
<b>Tabla 3.2</b> Datos obtenidos para el módulo B de caudal .....	55
<b>Tabla 3.3</b> Datos obtenidos para el módulo C de caudal .....	55
<b>Tabla 3.4</b> Datos obtenidos para el módulo D de caudal .....	55
<b>Tabla 3.5</b> Datos obtenidos para el módulo 1 de presión .....	57
<b>Tabla 3.6</b> Datos obtenidos para el módulo 2 de presión .....	57
<b>Tabla 3.7</b> Datos obtenidos para el módulo 3 de presión .....	58
<b>Tabla 3.8</b> Datos obtenidos para el módulo 4 de presión .....	58
<b>Tabla 3.9</b> Resultados de la prueba de transmisión y recepción de datos .....	59

## RESUMEN

El presente proyecto de titulación, Implementación de módulos didácticos para la medición remota de presión y caudal realizado en la Escuela de Formación de Tecnólogos (ESFOT). Tiene como finalidad equipar al Laboratorio de Tecnología Industrial (LTI) de un sistema de medición, monitoreo y registro de variables físicas, que permita a los estudiantes visualizar y registrar datos en tiempo real mediante una página *web* y una aplicación móvil, para reforzar los conocimientos en las áreas de instrumentación y telemetría.

El presente documento está conformado por cinco secciones que se detallan de la siguiente manera:

En la primera sección se describe la introducción, el objetivo general y los objetivos específicos, también se incluye una descripción del dispositivo *WeMos D1 mini ESP8266* y de la plataforma *Cayenne myDevices*.

La segunda sección contiene la descripción de la metodología empleada, de acuerdo con los objetivos planteados.

La tercera sección detalla el proceso de construcción de los módulos, además se describe el desarrollo de los códigos de programación, selección de materiales, diseño de diagramas esquemáticos y pruebas de funcionamiento.

La cuarta sección expone las conclusiones y recomendaciones obtenidas durante la realización del proyecto.

Para finalizar, la quinta sección contiene las referencias bibliográficas utilizadas para la realización del proyecto.

**PALABRAS CLAVE:** Caudal, Presión, Monitoreo, ESP8266, *Cayenne myDevices*.

## **ABSTRACT**

*The present degree project, Implementation of training modules for the remote measurement of pressure and flow deployed in the “Escuela de Formación de Tecnólogos” (ESFOT). It has as purpose, to equip the “Laboratorio de Tecnología Industrial” (LTI) of a system of measurement, monitoring and recording physical variables, that allow students visualize and records data of these variables in real time through a website and mobile application, to reinforce the knowledge in the area of instrumentation and telemetry.*

*The current document is made up of five sections that are detailed as follows:*

*The first part describes the introduction, the general objective, and the specific objectives, also includes a brief description of the WeMos D1 mini ESP8266 device and the Cayenne myDevices platform.*

*The second part contains the description of the methodology used, in accordance with the objectives set.*

*The third section details the process of building the modules, also describes the development of programming codes, selection of materials, design of schematic diagrams and functional tests.*

*The fourth part presents the conclusions and recommendations obtained during the implementation of the project.*

*Finally, the fifth part contains the bibliographic references used for the realization of the project.*

**KEYWORDS:** *Flow, Pressure, Monitoring, ESP8266, Cayenne myDevices.*

# 1 INTRODUCCIÓN

El Laboratorio de Tecnología Industrial (LTI) está ubicado en la Escuela de Formación de Tecnólogos, el cual cuenta con módulos industriales didácticos, utilizados por los estudiantes para ampliar sus habilidades en el manejo de equipos eléctricos y electrónicos para la medición, control, monitoreo y supervisión de distintas variables físicas. El laboratorio es utilizado por estudiantes de las carreras de Tecnología en Electrónica y Telecomunicaciones, Electromecánica y Agua y Saneamiento Ambiental con un aforo de 40 personas [1].

Sin embargo, dicho laboratorio no cuenta con un sistema de medición y registro de las magnitudes de presión y caudal de forma remota, el cual es muy importante ya que la telemetría permite medir magnitudes tanto físicas como químicas en sitios remotos y de difícil acceso, además registra estas variables mediante sistemas de adquisición de datos y las envía al operador del sistema [2]. En las prácticas realizadas en el laboratorio, en cuanto a la medición de presión y caudal, únicamente se utilizan dispositivos analógicos y digitales como los multímetros, los cuales en muchas de las ocasiones debido a su resolución y características provocan errores en las mediciones, además por el número limitado de equipos no existen los suficientes para todos los grupos de estudiantes. Esto ha provocado problemas en cuanto a tiempo de realización y presentación de prácticas de laboratorio. Otro de los problemas principales radica en la falta de un sistema de adquisición y registro de datos en el LTI, a través de redes de comunicación, para la posible implementación de un sistema de Supervisión, Control y Adquisición de Datos (SCADA) que permita la gestión no solo de los datos de caudal y presión sino de todos los módulos didácticos industriales presentes en el Laboratorio.

Teniendo en cuenta estos problemas, el presente proyecto constituye un aporte inicial para la implementación de sistemas basados en telemetría, además es muy importante que los estudiantes que utilicen el laboratorio de la ESFOT tengan conocimientos relacionados con la medición, registro y monitoreo de variables físicas a través de telemetría para posteriores aplicaciones en el campo laboral.

Para ello, a través de este proyecto se realizó la implementación de ocho módulos didácticos que permiten la medición y registro de forma remota de presión y caudal utilizando una red Wi-Fi. Además, los datos se pueden visualizar en tiempo real en una página *web* para su posterior análisis.



En el Anexo 1 se adjunta el certificado de funcionamiento el cual avala que este proyecto se implementó correctamente.

## 1.1 Objetivo general

Implementar módulos didácticos para medición remota de presión y caudal en el LTI de la ESFOT.

## 1.2 Objetivos específicos

Determinar los requerimientos necesarios para el desarrollo de los módulos didácticos para la medición de caudal y presión.

Desarrollar algoritmos para registro de datos.

Construir ocho módulos para medición remota.

Realizar pruebas de funcionamiento.

Elaborar un manual de usuario.

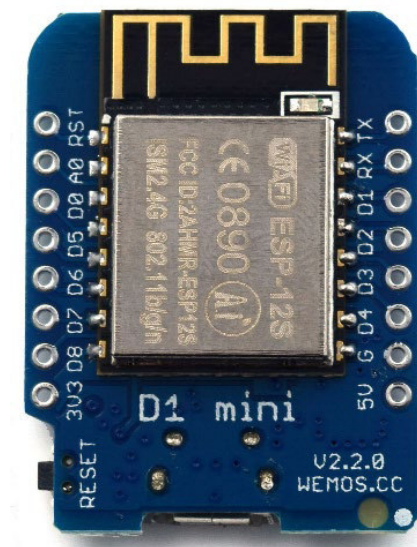
## 1.3 Fundamentos

### **WeMos D1 mini ESP8266**

Es una tarjeta basada en el microcontrolador ESP8266, la cual es utilizada para aplicaciones de *Internet of Things (IoT)*, esta placa posee una comunicación IEEE 802.11n o Wi-Fi para realizar el envío de información a plataformas en la nube, además de otras utilidades. Esta tarjeta tiene la ventaja de ser compatible con Arduino IDE, agregando algunas librerías para su programación [3]. En la Figura 1.1 se observa el dispositivo.

Posee algunas características que son importantes [4]:

- Voltaje de operación de 3,3 ( $V_{DC}$ ).
- Frecuencia de 80 (MHz).
- 11 puertos digitales de I/O.
- 1 puerto analógico.
- Comunicación UART, I2C.

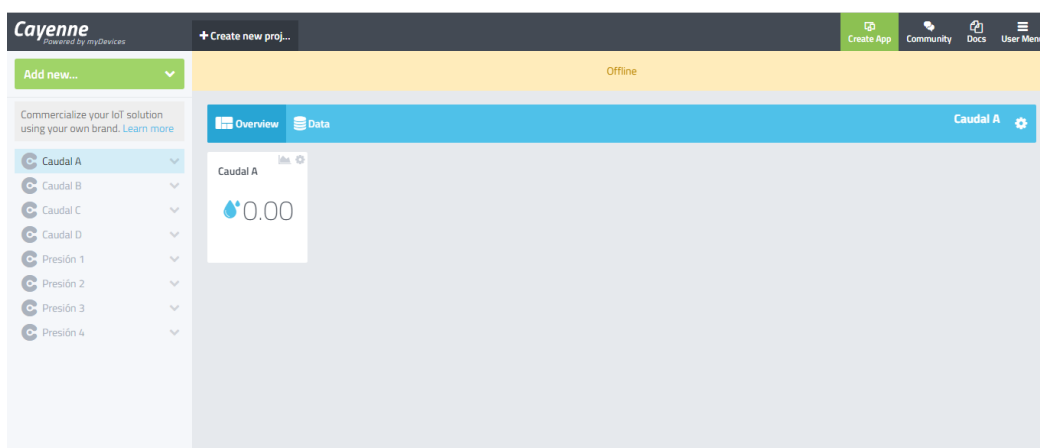


**Figura 1.1** WeMos D1 mini [4]

El Anexo 3 dispone de información complementaria sobre las características de *WeMos D1 mini*.

### ***Cayenne myDevices***

*Cayenne myDevices* es una plataforma gratuita para la creación de prototipos basados en IoT. Una de sus mayores ventajas es que es compatible con varios dispositivos, entre los más destacados Arduino y Raspberry Pi. Además, cuenta con un sistema fácil de configurar sin necesidad de programación, su sistema es de *Drag and Drop* (arrastrar y soltar). *Cayenne* también permite monitorear y controlar remotamente proyectos de E/S desde su plataforma *web* y móvil [5]. En la Figura 1.2 se observa la plataforma.



**Figura 1.2** Plataforma *Cayenne myDevices*

## 2 METODOLOGÍA

### 2.1 Descripción de la metodología usada

El diseño e implementación de los módulos didácticos para el LTI de la ESFOT se realizó mediante la aplicación de los conocimientos adquiridos a lo largo de la formación académica y con el apoyo de la respectiva investigación en temas relacionados a base de datos, telemetría y páginas *web*.

Durante el transcurso del proyecto se llevó a cabo la visita al LTI, donde se encontraban 4 módulos de medición de caudal y 4 transmisores de presión. Posteriormente, se efectuaron pruebas para comprobar su correcto funcionamiento y verificar las respectivas señales eléctricas de salida. En base a esto, se procedió a calibrar y acondicionar los transmisores para obtener los valores de voltaje y corriente normalizados.

Una vez realizada la visita al laboratorio y mediante una investigación en algunas fuentes de Internet, *blogs* de Arduino y video tutoriales, se logró desarrollar un código de programación en Arduino IDE que consigue leer las variables de los instrumentos de transmisión y ser enviadas al servidor. Para la transmisión de datos vía Wi-Fi se utilizó el puerto serial del microcontrolador.

Para la configuración de la base de datos se utilizó una plataforma de prototipado de dispositivos *IoT* llamada *Cayenne myDevices*, que es una página *web* dinámica con acceso a la nube, que permite presentar el sistema de una forma gráfica.

En la página *web* se registra un usuario y las variables provenientes de los instrumentos de medición se muestran a través de herramientas gráficas, las cuales permiten observar los valores medidos de presión y caudal en tiempo real.

Una vez elaborados los algoritmos de control, en el programa *Proteus Design Suite* se realizaron los diagramas esquemáticos de los circuitos donde constan todas las conexiones realizadas entre el Arduino Nano, *WeMos D1 mini ESP8266* y los respectivos instrumentos de medición. Se utilizó el módulo *WeMos D1 mini ESP 8266* para la transmisión de datos, ya que este tiene integrado el protocolo de comunicación TCP/IP, lo que permite un fácil acceso a una red Wi-Fi.

Luego, se obtuvieron los materiales necesarios para la construcción de los módulos de medición remota. Se diseñó la placa *Printed Circuit Board (PCB)* en *Proteus Design Suite* en base a los esquemas electrónicos.

Se ejecutaron las respectivas pruebas eléctricas y de comunicación para comprobar el correcto funcionamiento de los módulos de presión, caudal y el microcontrolador, así como pruebas técnicas para comprobar el correcto funcionamiento de la página *web* y la base de datos.

Por último, se realizaron dos videos que contienen el manual de uso de los módulos de presión y caudal, respectivamente. En los mismos se incluyen sus características principales para que el usuario pueda realizar una manipulación correcta, además se mencionan algunas soluciones a posibles problemas que se puedan presentar.

### 3 RESULTADOS Y DISCUSIÓN

Se elaboraron ocho módulos didácticos para el registro de variables físicas para el LTI-ESFOT, los cuales permiten obtener en tiempo real los valores medidos por cuatro módulos de caudal y cuatro módulos de presión empleando ocho Arduinos Nano y ocho placas *WeMos* D1 mini ESP8266, con sus respectivos algoritmos para registrar la información en un servidor *web*. A continuación, se describen los requerimientos, diseño de los módulos, estado de los sensores, diagramas esquemáticos, códigos de programación, implementación, pruebas de funcionamiento y el manual de usuario.

#### 3.1 Requerimientos

##### Instrumentos de medición de caudal y presión

Se revisaron los instrumentos de caudal y presión, disponibles en el LTI, con el propósito de constatar su correcto funcionamiento. A continuación, se detallan las principales características de los instrumentos utilizados en este proyecto.

##### Presión

Los transmisores de presión miden de 0 (bar) a 10 (bar) con una señal de salida de corriente de 4 (mA) a 20 (mA); cabe mencionar que es necesaria una alimentación de 24 (V<sub>DC</sub>) para su energización. Dicho transmisor se observa en la Figura 3.1.



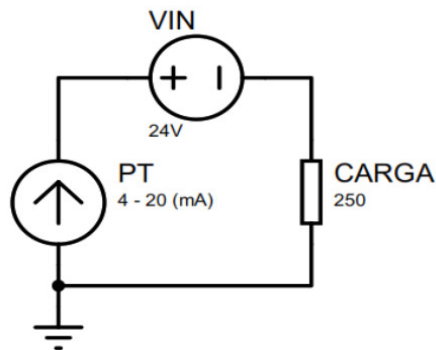
**Figura 3.1** Transmisor de presión

En el LTI-ESFOT se disponen de cuatro transmisores de presión, los cuales se observan en la Figura 3.2.



**Figura 3.2** Transmisores de presión LTI-ESFOT

Para transformar la señal de corriente de salida que brindan estos transmisores, en una señal de voltaje normalizado de 1 (V<sub>DC</sub>) a 5 (V<sub>DC</sub>), es necesario realizar un circuito en serie con una única resistencia de valor de 250 (Ω) para trabajar con los valores adecuados de voltaje. En la Figura 3.3 se presenta el respectivo circuito diseñado.



**Figura 3.3** Circuito de conversión de corriente a voltaje

Para determinar el valor de la resistencia fue necesario emplear la ley de Ohm que se muestra en la Ecuación 3.1:

$$R = \frac{V}{I}$$

**Ecuación 3.1** Ley de Ohm [6]

Donde:

$V$  : 5 (V<sub>DC</sub>) voltaje

$I$  : 20 (mA) corriente

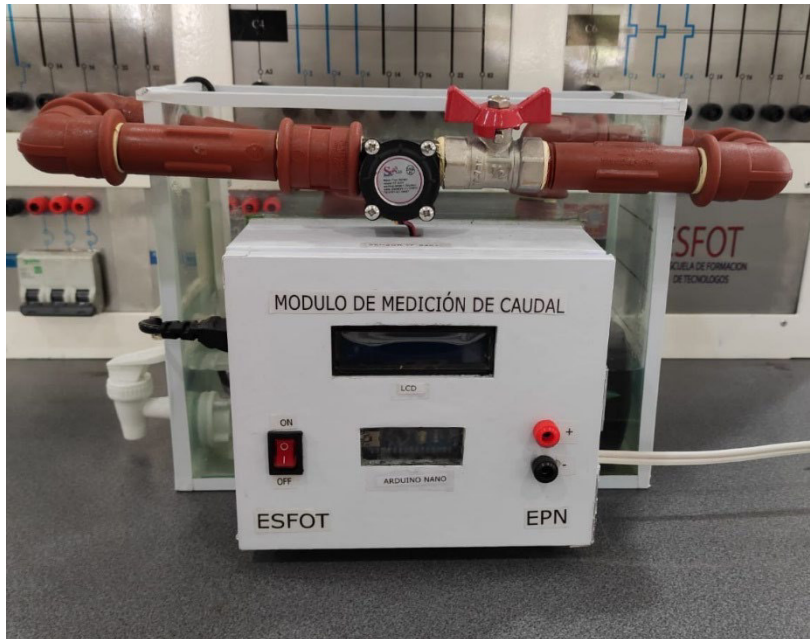
$R$  : (Ω) resistencia

Usando la Ecuación 3.1 se obtiene:

$$R = 250 (\Omega)$$

### Caudal

Se trata de un módulo para medición de caudal a partir de un sensor de flujo YF-S201, el cual mide de 1 (l/min) a 7 (l/min). A la salida entrega una señal normalizada de 1 (V<sub>DC</sub>) a 4 (V<sub>DC</sub>). Dicho módulo se observa en la Figura 3.4.



**Figura 3.4** Módulo de caudal

En el LTI-ESFOT se disponen de cuatro módulos de caudal, los cuales se observan en la Figura 3.5.



**Figura 3.5** Módulos de caudal del LTI-ESFOT

Tomando en cuenta las características de los instrumentos de medición de presión y caudal, se determinaron los principales requerimientos que deben cumplir los módulos de medición remota:

- Dimensiones adecuadas en función de los instrumentos de medición y de las mesas de trabajo del LTI-ESFOT.
- Compatibilidad con las señales de salida de los módulos de caudal y de los transmisores de presión.
- Manipulación, transporte y conexión sencillos.
- Espacio reducido de almacenamiento.

## **Diseño de los módulos**

### **Dimensiones**

Para precisar las dimensiones de los módulos de presión, se tomó en cuenta el tamaño de los dispositivos utilizados (Arduino Nano, *WeMos* D1 mini ESP8266 y circuito acondicionador), dado que en las mesas de trabajo del LTI existe un espacio limitado de trabajo. Se determinó un área de 53,04 (cm<sup>2</sup>), con las siguientes dimensiones: 7,8 (cm) de largo y 6,8 (cm) de ancho.

Al igual que en el módulo de presión, para los módulos de caudal se tomaron en cuenta los dispositivos utilizados (Arduino Nano, *WeMos* D1 mini ESP8266 y filtros RC pasa bajos), así mismo, por el tamaño reducido de los dispositivos empleados, se prioriza el espacio en las mesas de trabajo del LTI, otorgando más comodidad a los estudiantes para una adecuada manipulación. Se determinó un área de 53,72 (cm<sup>2</sup>), con las siguientes dimensiones: 7,9 (cm) de largo, 6,8 (cm) de ancho.

Tanto los módulos de presión, así como los módulos de caudal cumplen con las siguientes características:

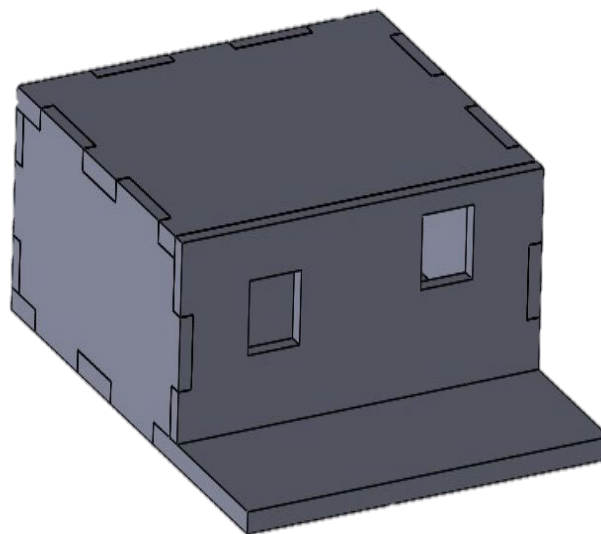
- Etiquetado de los pines más relevantes.
- Orificios en lugares adecuados para energizar a los dispositivos.
- Espacio reducido.
- Alimentación de voltaje de 9 (V<sub>DC</sub>) mediante el *Jack* de Arduino Nano.

### **Estructuras de las cajas**

Se esquematizaron las cajas en el *software* de diseño 3D *SolidWorks* teniendo en cuenta las dimensiones de los diferentes dispositivos, así como la de sus respectivos conectores. El material utilizado es acrílico de 2,5 (mm) de espesor, que les brinda mayor seguridad ante manipulaciones, ya que este permite visualizar el interior de la caja gracias a que es transparente. Las cajas, tienen el propósito de proteger a los dispositivos internos y organizarlos de forma adecuada.

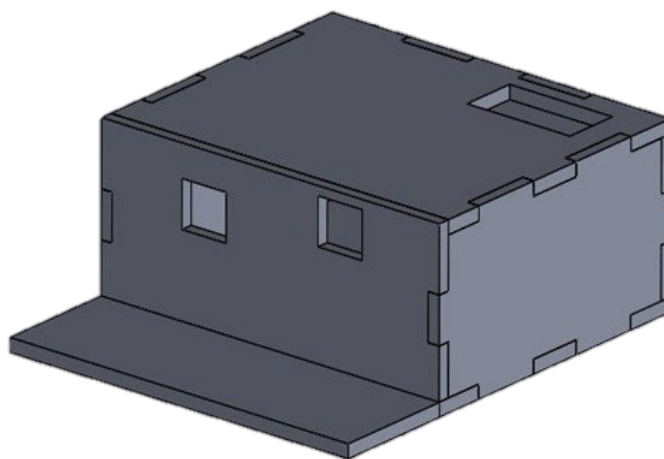


En la Figura 3.6 se presenta el esquema de una caja armada del módulo de presión con los orificios para sus respectivos conectores. Los planos diseñados se observan en el Anexo 2, lámina 1.



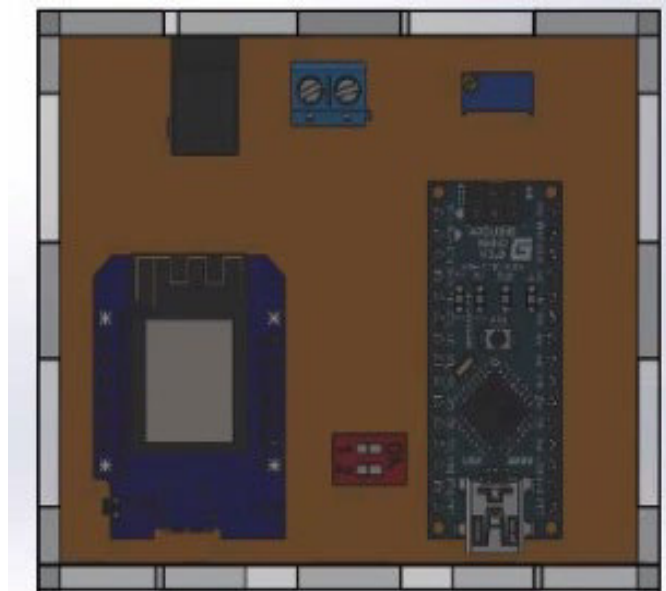
**Figura 3.6** Esquema de la caja (módulo de presión)

La Figura 3.7 presenta el esquema de una caja armada del módulo de caudal con los orificios para sus respectivos conectores. Los planos diseñados se observan en el Anexo 2, lámina 2.

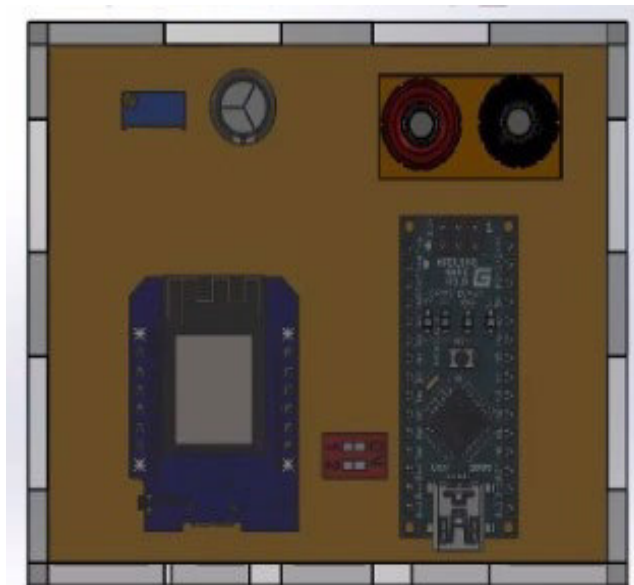


**Figura 3.7** Esquema de la caja (módulo de caudal)

La Figura 3.8 y Figura 3.9 muestran la disposición de los dispositivos dentro de las cajas de presión y caudal, respectivamente.



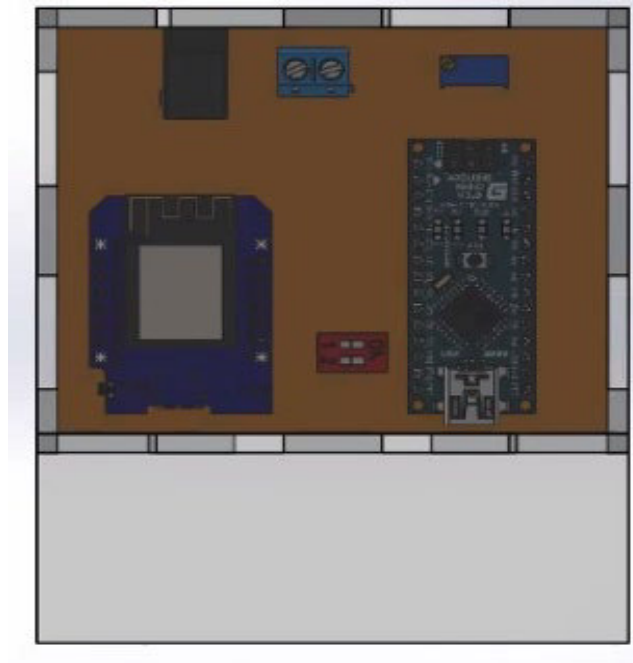
**Figura 3.8** Disposición de los dispositivos vista superior (módulo de presión)



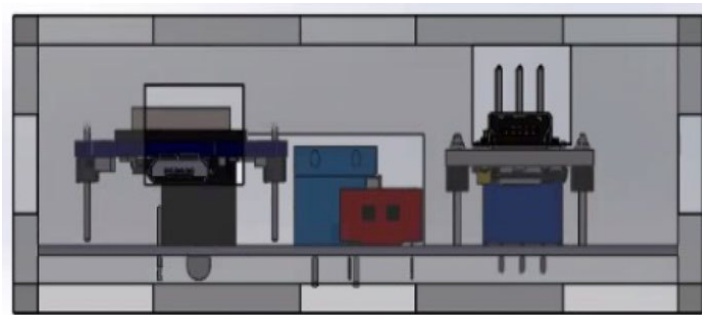
**Figura 3.9** Disposición de los dispositivos vista superior (módulo de caudal)

### **Estructura de los módulos**

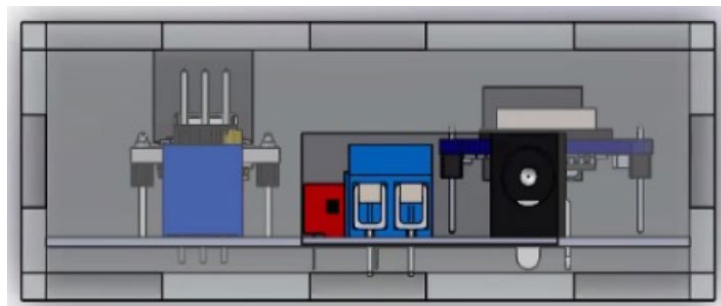
Tanto para la estructura de los módulos de presión, así como para la estructura de los módulos de caudal se estableció una única base de acrílico de 2,5 (mm) de espesor y dimensiones de largo = 106 (cm) y ancho = 92,1 (cm). Sobre esta se asientan las cajas diseñadas anteriormente para los módulos de presión y caudal según corresponda, los circuitos de conexión de los sensores, sus respectivos filtros y acondicionamiento. En las Figuras 3.10, 3.11 y 3.12, se muestra la vista superior, la vista frontal y la vista posterior del módulo de presión, respectivamente.



**Figura 3.10** Vista superior de uno de los módulos de presión



**Figura 3.11** Vista frontal de uno de los módulos de presión

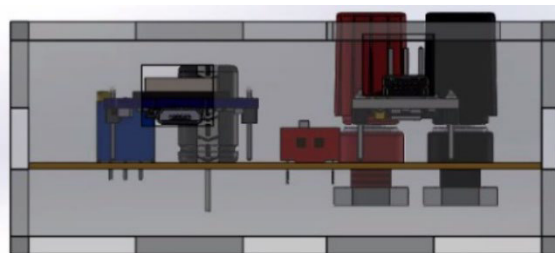


**Figura 3.12** Vista posterior de uno de los módulos de presión

Así mismo, en la Figuras 3.13 y 3.14 se muestra la vista superior y la vista frontal del módulo de caudal, respectivamente.



**Figura 3.13** Vista superior de uno de los módulos de caudal



**Figura 3.14** Vista frontal de uno de los módulos de caudal

## 3.2 Algoritmos para registro de datos

Los algoritmos que permiten la lectura de datos en los pines analógicos, el acondicionamiento de los sensores mediante *software*, el envío de los datos leídos al *WeMos D1 mini ESP8266* y a la vez el envío posterior al servidor *Cayenne* fueron desarrollados en el *software* Arduino IDE.

### Lectura de datos Arduino Nano

Los datos provenientes de los módulos de caudal y de los transmisores de presión fueron receptados por los pines A0 y A1 de los Arduinos Nano respectivamente. En base a esto, se realizó el respectivo acondicionamiento a través de programación. Una vez procesadas las señales, se procedió a enviarlas al *WeMos D1 mini ESP8266* por comunicación serial, a través de los pines D10 y D11 del Arduino Nano. A continuación, se explica el código desarrollado.

Para leer los datos de los módulos de caudal y de los transmisores de presión se desarrolló un código de programación que aplica para cada módulo y transmisor según

sea el caso. Teniendo en cuenta que se leen los datos de un solo módulo o de un solo transmisor, no es necesario diferenciar los pines de entrada en el código; este se repite en todos los códigos.

En la Figuras 3.15 y 3.16 se muestran las variables empleadas en el código para el registro de los datos de caudal y presión, respectivamente. Se utilizó una variable tipo flotante (*float*) para incluir todos los decimales en la lectura de datos y así no perder información. Se incluyó la librería “*SoftwareSerial*” para recibir y transmitir los datos mediante los pines 10 y 11 del Arduino Nano, respectivamente.

```
//LECTURA DE DATOS|  
  
#include <SoftwareSerial.h> //Libreria para emular un puerto serial fisico  
  
SoftwareSerial mySerial(10, 11); //Pines RX, TX, pines digitales para emular un puerto de comunicaci3n  
  
int sensor;  
float caudal;  
float suma;  
float variable;
```

**Figura 3.15** Declaraci3n de variables (m3dulo de caudal)

```
//LECTURA DE DATOS  
  
#include <SoftwareSerial.h> //Libreria para emular un puerto serial fisico  
  
SoftwareSerial mySerial(10, 11); //Pines RX, TX, pines digitales para emular un puerto de comunicacion  
  
float sensor1;  
float presion;  
int suma;  
float variable1;
```

**Figura 3.16** Declaraci3n de variables (m3dulo de presi3n)

En la Figura 3.17 se muestra la funci3n “void setup()”, en la cual da inicio la Comunicaci3n Serial; se aplica la misma funci3n para los c3digos de presi3n y caudal.

```
void setup() {  
  Serial.begin(9600); //Comunicaci3n Serial  
  mySerial.begin(9600); //Puerto serial emulado igualmente inicia en 9600 baudios  
}
```

**Figura 3.17** Parte “void setup()”

En la Figura 3.18 se muestra la funci3n “void loop()”, en la cual se lee y acondiciona los datos provenientes de presi3n y caudal, respectivamente.

```

void loop() {
  suma=0;

  for (int i=0; i<5; i++){

```

**Figura 3.18** Inicio de lectura de datos y acondicionamiento parte “void loop()”

A continuación, se presentan las ecuaciones de la recta obtenidas para cada variable física.

### Presión

El acondicionamiento de la señal de presión se realizó obteniendo la ecuación de la recta con respecto a la variable de entrada y la variable de salida. Se consiguieron los siguientes puntos a partir del instrumento de medición:  $P_1$  (0;205),  $P_2$  (10 ;1023).

$$UP - UP_1 = m(P - P_1)$$

**Ecuación 3.2** Ecuación de la recta: presión en función de unidades de procesamiento.

$$m = \frac{UP_2 - UP_1}{P_2 - P_1}$$

**Ecuación 3.3** Ecuación de la pendiente en función de la presión.

Donde:

UP : Variable de entrada

$UP_1$  : 1023 (Up) unidad de procesamiento

$UP_2$  : 205 (Up) unidad de procesamiento

$P_1$  : 0 (bar) bares

$P_2$  : 10 (bar) bares

m : Pendiente

Reemplazando los valores y despejando P de las Ecuaciones 3.2 y 3.3 se obtiene:

$$P = 0,01209(UP) - 2,37$$

**Ecuación 3.4** Ecuación de la señal de presión

En la Figura 3.19 se muestra la lectura de los datos a través del pin A1 y su respectivo acondicionamiento.

```

sensor1=analogRead(A1);    // Lee entrada analoga (A1) entre 0-4v
variable1=0.01028*sensor1-2.51; //Ecuación de la recta entrada vs salida(ACONDICIONAMIENTO)
suma=variable1+suma;      //Promedio de lecturas

}

```

**Figura 3.19** Acondicionamiento de los valores de presión

### Caudal

El acondicionamiento de la señal de caudal se realizó obteniendo la ecuación de la recta con respecto a la variable de entrada y la variable de salida. Se obtuvieron los siguientes puntos a partir del instrumento de medición: P<sub>1</sub> (1,08; 174), P<sub>2</sub> (6,89; 1023).

$$UP - UP_1 = m(C - C_1)$$

**Ecuación 3.5** Ecuación de la recta: caudal en función de unidades de procesamiento

$$m = \frac{UP_2 - UP_1}{C_2 - C_1}$$

**Ecuación 3.6** Ecuación de la pendiente en función del caudal

Donde:

UP : Variable de entrada

UP<sub>1</sub> : 174 (Up) unidad de procesamiento

UP<sub>2</sub> : 1023 (Up) unidad de procesamiento

C<sub>1</sub> : 1,08 (l/min) litros sobre minuto

C<sub>2</sub> : 6,89 (l/min) litros sobre minuto

m : Pendiente

Reemplazando los valores y despejando C de la Ecuaciones 3.5 y 3.6 se obtiene:

$$C = 0,0068(UP) - 0,1107$$

**Ecuación 3.7** Ecuación de la señal de caudal

En la Figura 3.20 se muestra la lectura de los datos a través del pin A0 y su respectivo acondicionamiento.

```
sensor=analogRead(A0);           //Lee entrada analoga 0-4v
caudal=map(sensor,0,818,0,1023); //Transforma de voltaje a unidades crudas
variable=0.0070*caudal-0.471; //Ecuacion de la recta entrada vs salida (ACONDICIONAMIENTO)
suma=variable+suma;             //Promedio de lecturas
delay(200);
}
```

**Figura 3.20** Acondicionamiento de los valores de caudal

### **Función Serial.println**

Se envían los datos ya acondicionados anteriormente mediante Comunicación Serial al WeMos mini ESP8266. En la Figura 3.21 se aprecia la función “Serial.println()” que permite imprimir en el monitor serial los datos procesados de presión.

```
Serial.println(variable1);       //Publica en el monitor serial la PRESIÓN
mySerial.println(variable1);     //Usa el puerto serial emulado para enviar LA PRESION
delay(500);                      //Toma la medicion cada medio segundo
}
```

**Figura 3.21** Función “Serial.println()”, datos de presión

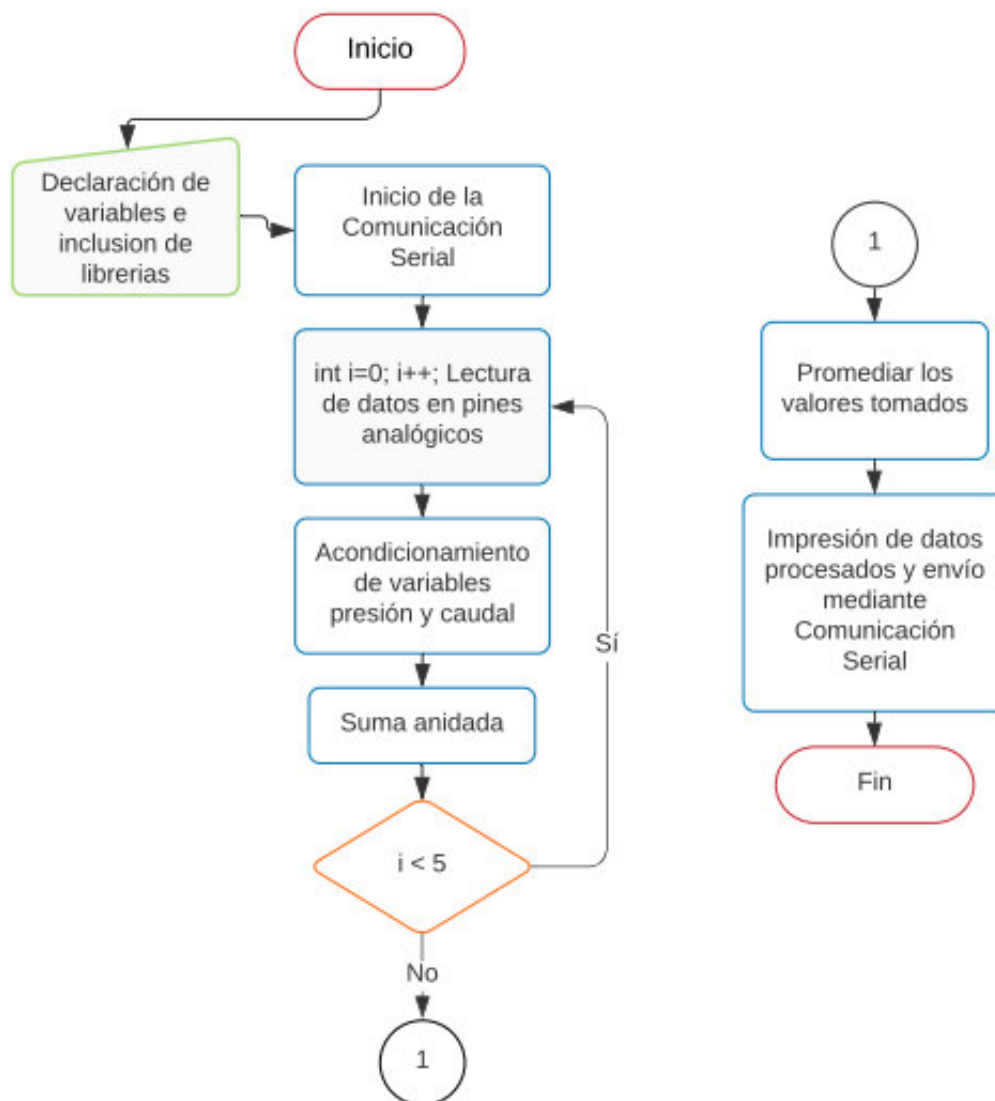
En la Figura 3.22 se aprecia la función “Serial.println()” que permite imprimir en el monitor serial los datos procesados de caudal.

```
Serial.println(variable, 2);     //Publica en el monitor serial el CAUDAL
mySerial.println(variable);     //Usa el puerto serial emulado para enviar el caudal
delay(500);                     //Toma la medicion cada medio segundo.
}
```

**Figura 3.22** Función “Serial.println()”, datos de caudal

En la Figura 3.23 se aprecia el diagrama de flujo correspondiente a la programación en los Arduino Nano.





**Figura 3.23** Diagrama de flujo del código en Arduino Nano

### **Envío de datos al servidor *Cayenne***

Para el envío de datos desde los Arduinos Nano se diseñó un algoritmo que permitió receptor dichos datos en el *WeMos D1 mini ESP8266* y establecer una conexión adecuada con el servidor *Cayenne*.

En la Figura 3.24 se muestra la parte de código en donde se añaden las librerías a usar específicamente para los modelos de placas Wi-Fi “ESP8266” añadidas únicamente con carácter tipo “#define”, se declaran las constantes y variables necesarias incluyendo la información de la cuenta registrada en *Cayenne*.

```

////////SELECCIONAR LA PLACA WIFI A UTILIZAR////////
//-----
//#define CAYENNE_DEBUG
#define CAYENNE_PRINT Serial
//-----
//#define IOT_MOD_WIZNET_COMPILE
#define IOT_MOD_8266_COMPILE
//#define IOT_MOD_ESP32_COMPILE
//-----
#include "Comandos.h"
//-----

// Informacion de la red WIFI
char ssid[] = "Control2019A"; //NOMBRE DE LA RED
char wifiPassword[] = "LTI-CI-2019"; //CLAVE DE LA RED

// Datos de la cuenta registrada en CAYENNE
char username[] = "aa39b380-ec27-11eb-b767-3fla8f1211ba";
char password[] = "e949014213733487ead9268efb7eaf571c2933f7";
char clientID[] = "b7874f70-ec27-11eb-a2e4-b32ea624e442";

float caudal1;

```

**Figura 3.24** Declaración constantes y variables, librerías a usar (variable caudal)

La parte del código inicial es la misma para presión y caudal. En la Figura 3.25 se puede apreciar la única diferencia que es la declaración de la variable para el dato de presión a leer, en vez de la variable de caudal.

```

////////SELECCIONAR LA PLACA WIFI A UTILIZAR////////
//-----
//#define CAYENNE_DEBUG
#define CAYENNE_PRINT Serial
//-----
//#define IOT_MOD_WIZNET_COMPILE
#define IOT_MOD_8266_COMPILE
//#define IOT_MOD_ESP32_COMPILE
//-----
#include "Comandos.h"
//-----

// Informacion de la red WIFI
char ssid[] = "Pato"; //NOMBRE DE LA RED
char wifiPassword[] = "jv123456"; //CLAVE DE LA RED

// Datos de la cuenta registrada en CAYENNE
char username[] = "aa39b380-ec27-11eb-b767-3fla8f1211ba";
char password[] = "e949014213733487ead9268efb7eaf571c2933f7";
char clientID[] = "b7874f70-ec27-11eb-a2e4-b32ea624e442";

float presion1;

```

**Figura 3.25** Declaración constantes y variables, librerías a usar (variable presión)

En la Figura 3.26 se aprecia la función “void setup()”, en la cual se establece la conexión con la red Wi-Fi señalada anteriormente; la función “Cayenne.begin” admite 5 parámetros, los 3 primeros son los necesarios para la comunicación con *Cayenne*, se establece la conexión con *Cayenne* para la recepción de datos y se elige una salida para mostrar los datos.

```

void setup() {
  Serial.begin(9600);    //Inicia el puerto serial en 9600 baudios

  ///Envia los parametros ingresados de Cayenne
  #ifdef IOT_MOD_WIZNET_COMPILE
  Cayenne.begin(username, password, clientID);
  #endif

  #ifdef IOT_MOD_8266_COMPILE
  Cayenne.begin(username, password, clientID, ssid, wifiPassword); // Inicia la comunicación con Cayenne
  #endif

  #ifdef IOT_MOD_ESP32_COMPILE
  Cayenne.begin(username, password, clientID, ssid, wifiPassword);
  #endif

  CAYENNE_OUT(1); //Selecciona la salida 1 en Cayenne
}

```

**Figura 3.26** Autenticación de la red y conexión con el servidor *Cayenne*

En cada una de las Figuras 3.27, 3.28, 3.29 y 3.30 se muestra la función “void loop()” para los datos procesados de presión y listos para ser mostrados en el servidor *Cayenne*, identificando claramente el transmisor del cual provienen los datos.

```

void loop() {
  if (Serial.available() > 0) {
    presion1 = Serial.parseFloat();
  }

  delay(1);
  Cayenne.loop(); //Selecciona el método loop de Cayenne para ejecutar información.
}

CAYENNE_OUT_DEFAULT(){

  Cayenne.virtualWrite(CANAL_CYN_00, presion1); //Envía el dato del transmisor de presión 1 por el canal 0.

}

```

**Figura 3.27** Ejecución de *Cayenne* (datos del transmisor de presión 1)

```

void loop() {
  if (Serial.available() > 0) {
    presion = Serial.parseFloat();
  }

  delay(1);
  Cayenne.loop(); //Selecciona el método loop de Cayenne para ejecutar información
}

CAYENNE_OUT_DEFAULT(){

  Cayenne.virtualWrite(CANAL_CYN_00, presion); //Envía el dato procesado del transmisor de presión 2 por el canal 0.

}

```

**Figura 3.28** Ejecución de *Cayenne* (datos del transmisor de presión 2)

```

void loop() {
  if (Serial.available() > 0) {
    presion1 = Serial.parseFloat();
  }

  delay(1);
  Cayenne.loop(); //Selecciona el método loop de Cayenne para ejecutar información
}

CAYENNE_OUT_DEFAULT(){

  Cayenne.virtualWrite(CANAL_CYN_00, presion1); //Envía el dato procesado del transmisor de presión 3 por el CANAL 0.
}

```

**Figura 3.29** Ejecución de *Cayenne* (datos del transmisor de presión 3)

```

void loop() {
  if (Serial.available() > 0) {
    presion1 = Serial.parseFloat();
  }

  delay(1);
  Cayenne.loop(); //Selecciona el método loop de Cayenne para ejecutar información
}

CAYENNE_OUT_DEFAULT(){

  Cayenne.virtualWrite(CANAL_CYN_08, presion1); //Envía el dato procesado del transmisor de presión 4 por el CANAL 0.
}

```

**Figura 3.30** Ejecución de *Cayenne* (datos del transmisor de presión 4)

En cada una de las Figuras 3.31, 3.32, 3.33 y 3.34 se muestra la función “void loop()” para los datos procesados de caudal y listos para ser mostrados en el servidor *Cayenne*, identificando claramente el módulo del cual provienen los datos.

```

void loop() {
  if (Serial.available() > 0) {
    caudal1 = Serial.parseFloat();
  }

  delay(1);
  Cayenne.loop(); //Selecciona el método loop de Cayenne para ejecutar información
}

CAYENNE_OUT_DEFAULT(){

  Cayenne.virtualWrite(CANAL_CYN_00, caudal1); //Envía el dato procesado del módulo de caudal A por el CANAL 0.
}

```

**Figura 3.31** Ejecución de *Cayenne* (datos del módulo de caudal A)

```

void loop() {
  if (Serial.available() > 0) {
    caudal1 = Serial.parseFloat();
  }

  delay(1);
  Cayenne.loop(); //Selecciona el método loop de Cayenne para ejecutar información
}

CAYENNE_OUT_DEFAULT(){

  Cayenne.virtualWrite(CANAL_CYN_00, caudal1); //Envía el dato procesado del módulo de caudal B por el CANAL 0.
}

```

**Figura 3.32** Ejecución de *Cayenne* (datos del módulo de caudal B)

```

void loop() {
  if (Serial.available() > 0) {
    caudal1 = Serial.parseFloat();
  }

  delay(1);
  Cayenne.loop(); //Selecciona el método loop de Cayenne para ejecutar información
}

CAYENNE_OUT_DEFAULT(){

  Cayenne.virtualWrite(CANAL_CYN_03, caudal1); //Envía el dato procesado del módulo de caudal C por el CANAL 0.
}

```

**Figura 3.33** Ejecución de *Cayenne* (datos del módulo de caudal C)

```

void loop() {
  if (Serial.available() > 0) {
    caudal1 = Serial.parseFloat();
  }

  delay(1);
  Cayenne.loop(); //Selecciona el método loop de Cayenne para ejecutar información
}

CAYENNE_OUT_DEFAULT(){

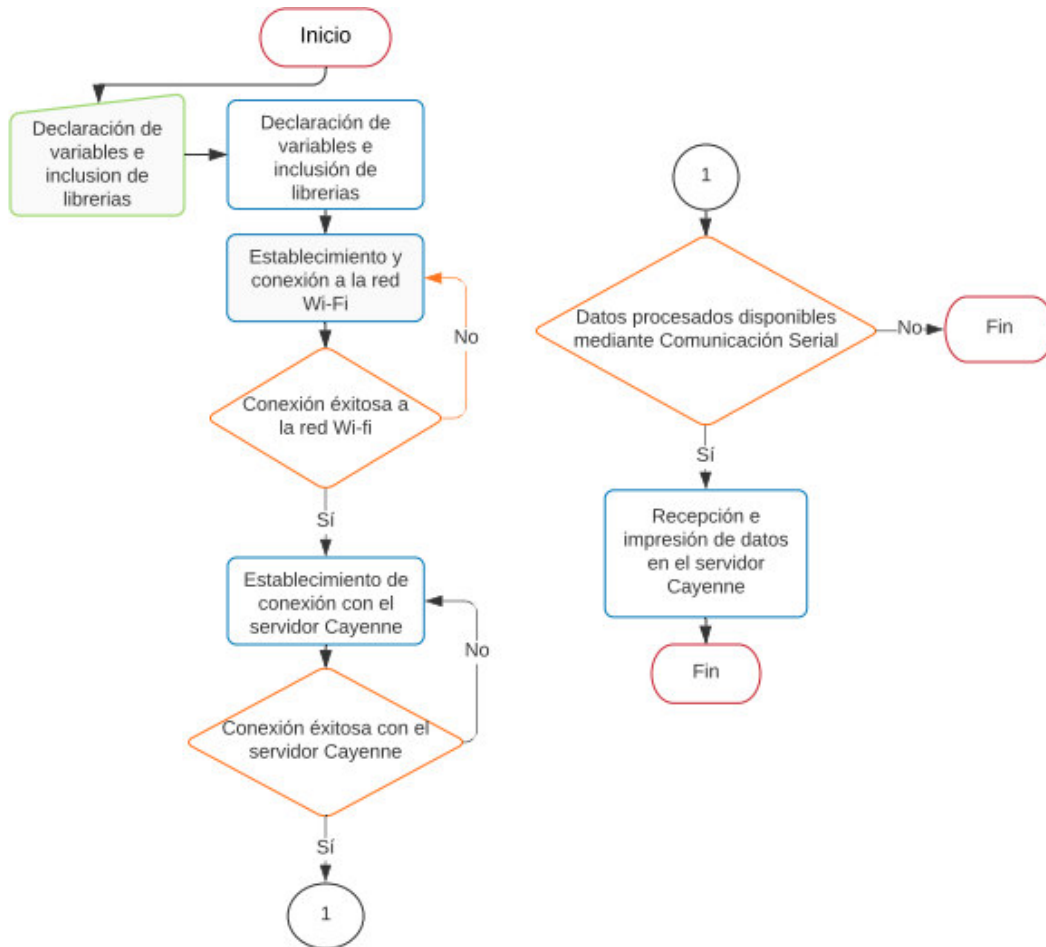
  Cayenne.virtualWrite(CANAL_CYN_04, caudal1); //Envía el dato procesado del módulo de caudal D por el CANAL 0.
}

```

**Figura 3.34** Ejecución de *Cayenne* (datos del módulo de caudal D)

Finalmente, se llama a la función “Cayenne.loop()” que ejecuta los datos en intervalos regulares ya definidos y muestra esos datos en los canales predeterminados con ayuda de la función “CAYENNE\_OUT\_DEFAULT ()” [7].

La Figura 3.35 muestra el diagrama de flujo empleado para el algoritmo del *WeMos D1 mini ESP8266*.



**Figura 3.35** Diagrama de flujo del programa en el *WeMos D1 mini ESP8266*

### Interfaz de *Cayenne myDevices*

La plataforma *Cayenne myDevices* fue desarrollada por *myDevices*, que es una empresa dedicada a la venta de sensores y equipos electrónicos y el propósito de la plataforma es el monitoreo y control de sensores mediante su interfaz amigable con el usuario y sin necesidad de programación; muestra de forma gráfica la información de cada uno de los sensores que se conecten. De este modo, toda la información se almacena en la nube de *Cayenne* de forma gratuita para futuros análisis.

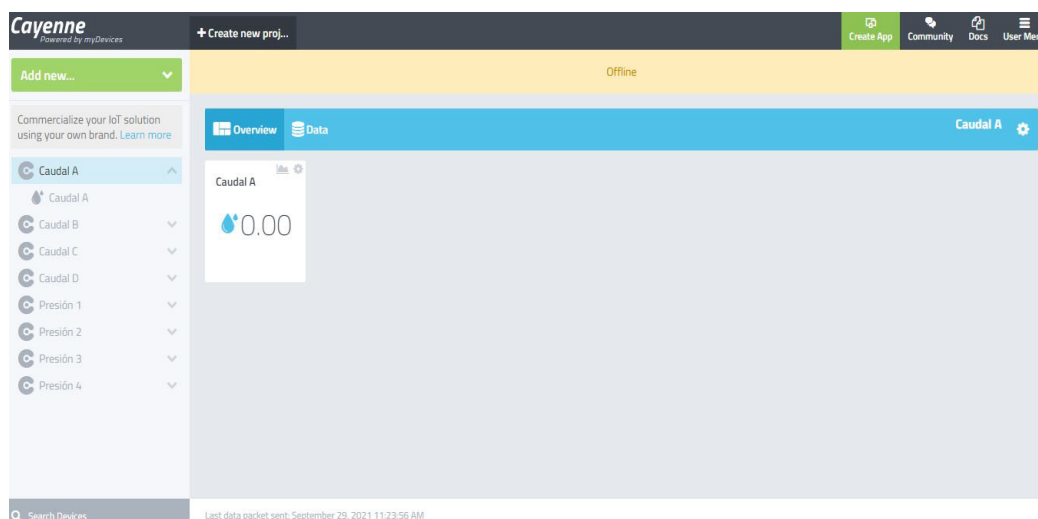
El registro en la plataforma es muy sencillo, se necesita llenar un formulario de registro (Figura 3.36) con 5 datos y ya se puede acceder a la plataforma.

The image shows a registration form for Cayenne myDevices. The form is titled "Register" and is set against a dark blue background. It contains the following fields: "First name", "Last name", "Email", "Password", and "Confirm password". Each field is represented by a white rectangular input box. Below the "Confirm password" field, there is a blue link that says "← Back to Login". At the bottom of the form, there is a prominent blue button labeled "Register". The Cayenne myDevices logo is positioned at the top left of the form area.

**Figura 3.36** Formulario de registro en *Cayenne*

Una vez que se realizó el registro, la programación y la configuración de los dispositivos, se puede acceder a la plataforma *Cayenne* que posee una página *web* y una aplicación móvil, la interfaz de la página *web*, donde se encuentran los dispositivos configurados se muestra en la Figura 3.37, igualmente en la Figura 3.38 se muestra la interfaz de la aplicación móvil.

Luego de la configuración, se ingresa a la plataforma con el correo y la contraseña que se colocó en el registro inicial, el proceso se muestra en la Figura 3.39.



**Figura 3.37** Interfaz *web* de *Cayenne*

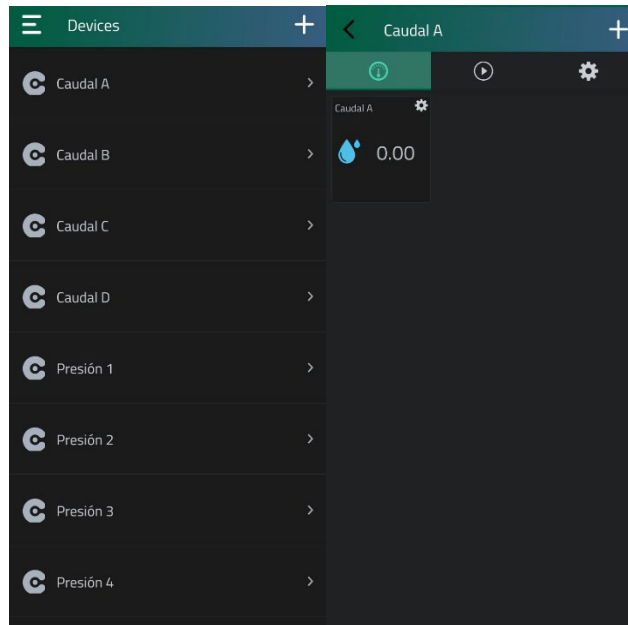


Figura 3.38 Interfaz de aplicación móvil de Cayenne

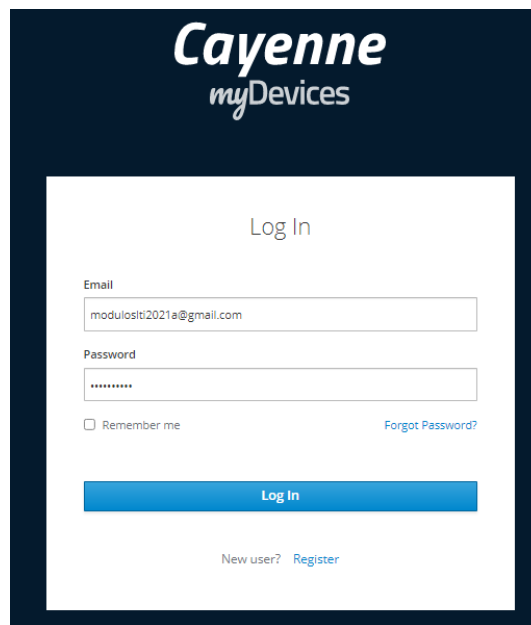


Figura 3.39 Inicio de sesión en Cayenne

### 3.3 Construcción

#### Diagrama esquemático

Se utilizó el *software Proteus* para el diseño del diagrama electrónico que establece las conexiones entre dispositivos. En la Figura 3.40 y 3.41 se observan todos los enlaces identificados con un número y detallados tanto para el módulo de presión y el módulo de caudal, respectivamente.



### Módulo de Presión

- **1 y 2:** Estos enlaces permiten el envío de los datos desde el Arduino Nano hasta el *WeMos D1 mini ESP8266*. Se dispone de un *DIP Switch* para controlar la conexión o desconexión, según se requiera. El envío de datos se logra mediante Comunicación Serial, es así como se tiene:
  - **1** = Conexiones desde el Arduino Nano al *DIP Switch* mediante los pines D10 y D11 para RX y TX, respectivamente.
  - **2** = Conexiones desde los pines 1 y 2 del *DIP Switch* a los pines del *WeMos D1 mini ESP8266* TX y RX, respectivamente.
- **3:** Muestra todas las conexiones a tierra (GND).
- **4:** Permite unir la salida del circuito acondicionador del transmisor de presión al pin A1 del Arduino Nano.
- **5:** Alimentación para el correcto funcionamiento del circuito de acondicionamiento del transmisor de presión.

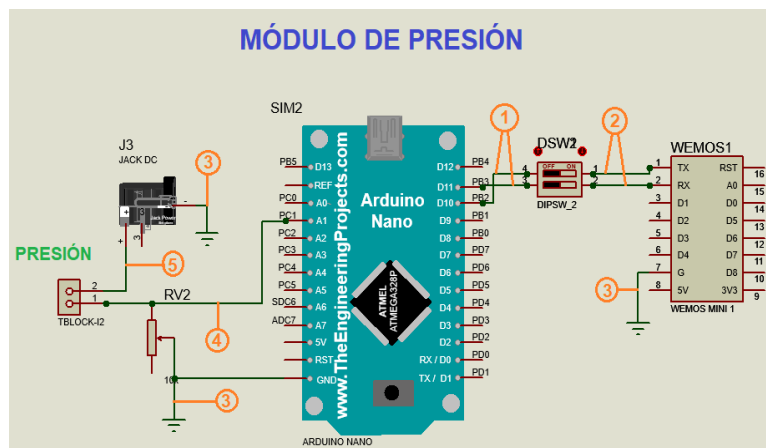


Figura 3.40 Diagrama electrónico (módulo de presión)

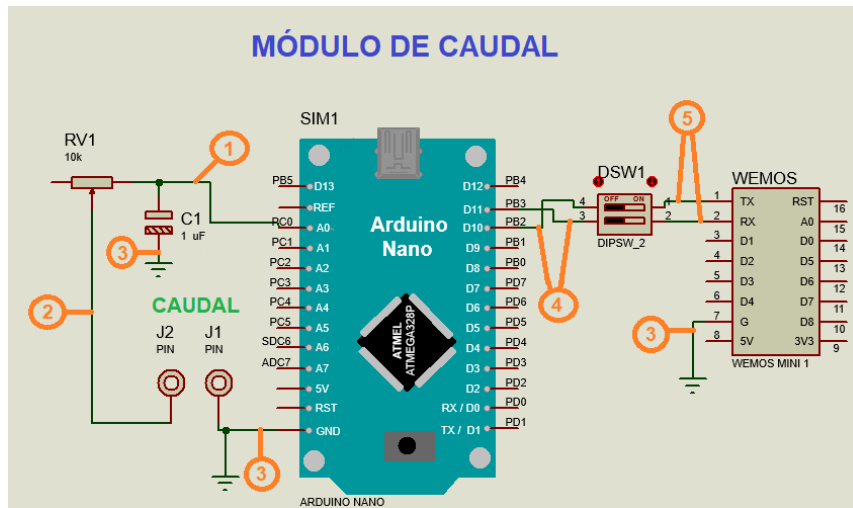
En el Anexo 2 lámina 3 se aprecia el diagrama esquemático con la disposición de los enlaces, tal cual están plasmados en las placas realizadas para los módulos de presión.

### Módulo de Caudal

- **1:** Este enlace conecta un extremo del filtro RC al pin A0 del Arduino Nano.
- **2:** Es el otro extremo del filtro RC que une a este con la bornera hembra banana del módulo de caudal.
- **3:** Muestra todas las conexiones a tierra (GND).
- **4 y 5:** Estos enlaces permiten el envío de los datos desde el Arduino Nano hasta el *WeMos D1 mini ESP8266*. Se dispone un *DIP Switch* para controlar la

conexión o desconexión según se requiera, el envío de datos se logra mediante Comunicación Serial, es así que se tiene:

- **4** = Conexiones desde el Arduino Nano al *DIP Switch* mediante los pines D10 y D11 para RX y TX, respectivamente.
- **5** = Conexiones desde los pines 1 y 2 del *DIP Switch* a los pines del *WeMos D1 mini ESP8266* TX y RX, respectivamente.



**Figura 3.41** Diagrama electrónico (módulo de caudal)

En el Anexo 2 lámina 3 se puede apreciar el diagrama esquemático con la disposición de los enlaces, tal cual están plasmados en las placas realizadas para los módulos de caudal.

### Diseño del filtro RC pasa bajos

Debido a que en las entradas analógicas de Arduino se necesita una señal estable con valores de bajos de frecuencia y lo que se dispone son señales de Modulación por Ancho de Pulso (PWM), que corresponde a un voltaje promedio según la señal de caudal medida, se utilizó un filtro RC pasa bajos para atenuar la onda PWM y obtener a la salida una señal analógica estable.

Para el cálculo de los valores de los dispositivos RC fue necesario probar valores de capacitancia y resistencia, con los cuales se obtenga un valor de frecuencia de corte necesaria para tener un rizado lo bastante bajo para la correcta atenuación de la señal, es así como se obtuvo un valor de resistencia de 10 (k $\Omega$ ) y un valor de capacitancia de 1 ( $\mu$ F). Mediante la Ecuación 3.8 se determinó el valor de la frecuencia de corte.

$$f_c = \frac{1}{(2\pi)(R)(C)}$$

**Ecuación 3.8** Ecuación para el cálculo de la frecuencia de corte del filtro RC [8]

Donde:

$f_c$  : (Hz) frecuencia de corte

R : 10 (k $\Omega$ ) resistencia

C : 1 ( $\mu$ F) capacitancia

Con la Ecuación 3.8 se obtiene el valor de la frecuencia de corte:

$$f_c = 15,9 \text{ (Hz)}$$

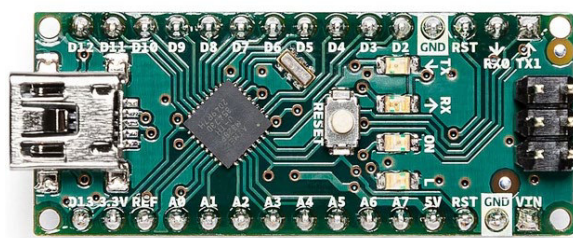
En base a los diseños esquemáticos realizados en *Proteus*, se elaboraron cuatro módulos didácticos para la medición de caudal y cuatro para la medición de presión. A continuación, se detallan las actividades llevadas a cabo.

### Adquisición de materiales

Se adquirieron los componentes, en base a los requerimientos establecidos para la implementación de los módulos. Estos componentes son presentados a continuación:

#### Arduino Nano

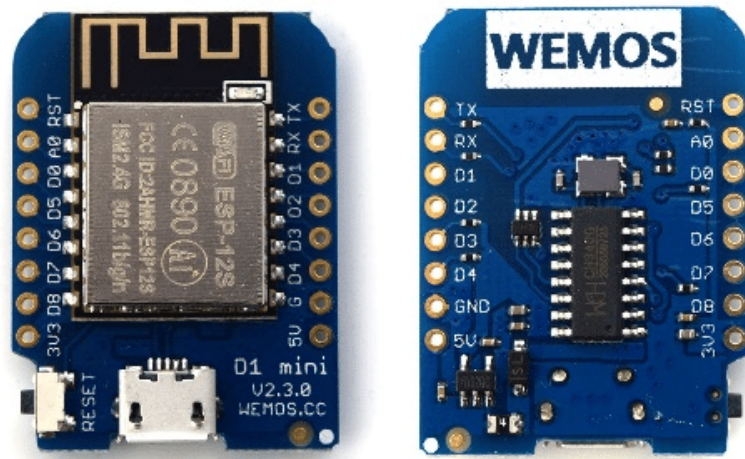
Se usa un Arduino Nano (Figura 3.42) debido a que cuenta con las entradas y salidas necesarias para la implementación de los módulos. Se utiliza para leer las señales de las variables físicas, luego las almacena para posteriormente enviarlas al módulo ESP8266.



**Figura 3.42** Arduino Nano [9]

#### WeMos D1 mini ESP8266

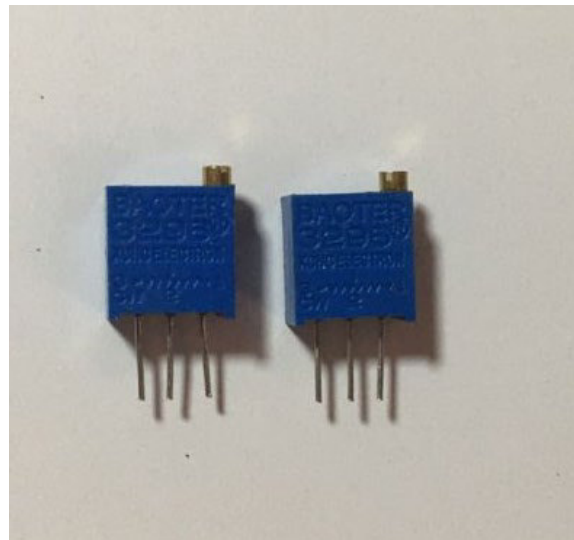
Este módulo ESP8266 (Figura 3.43) posee tecnología Wi-Fi y se encarga de recibir la información proveniente del Arduino Nano mediante Comunicación Serial entre los dos dispositivos, luego envía la información al servidor de *Cayenne myDevices*.



**Figura 3.43** WeMos D1 mini ESP8266 [10]

### **Potenciómetros de precisión**

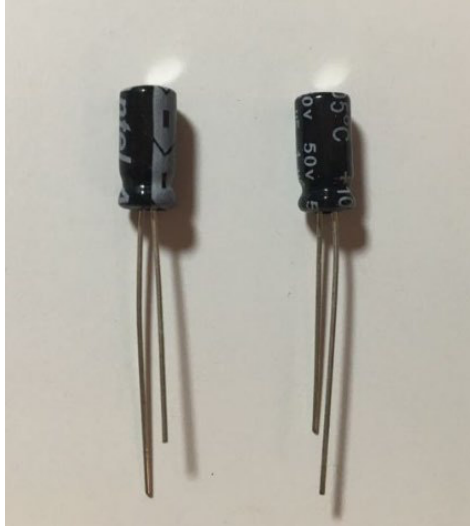
Se emplean para la realización de un circuito de conversión de corriente a voltaje en el caso de los módulos de presión, también forman parte de los filtros RC para convertir la señal PWM en una señal continua que ingresa al Arduino. En la Figura 3.44 se visualizan los potenciómetros de precisión.



**Figura 3.44** Potenciómetros de precisión

### **Condensadores**

Los capacitores (Figura 3.45) son componentes utilizados en los módulos de caudal para realizar un filtro RC, el cual ayuda a reducir el ruido de la señal que ingresa al Arduino. Se requirieron 4 condensadores de 1 ( $\mu\text{F}$ ) para los cuatro módulos de caudal.



**Figura 3.45** Condensadores de 1 ( $\mu\text{F}$ )

#### **Borneras tipo banana**

Estas borneras (Figura 3.46) reciben las señales de los módulos de caudal del laboratorio, se emplearon 4 pares de borneras tipo banana. Por las borneras de color rojo ingresan las señales de los instrumentos de medición, mientras que en las de color negro se conectan las tierras (GND).



**Figura 3.46** Borneras tipo banana [11]

#### **Conector *Power Jack***

Este conector se usa para alimentar el circuito de conversión de corriente a voltaje para el transmisor de presión, el cual funciona con una fuente de 24 ( $V_{\text{DC}}$ ). El conector se observa en la Figura 3.47.



**Figura 3.47** Conector *Power Jack* [12]

### **Fuente de 24 (V<sub>DC</sub>)**

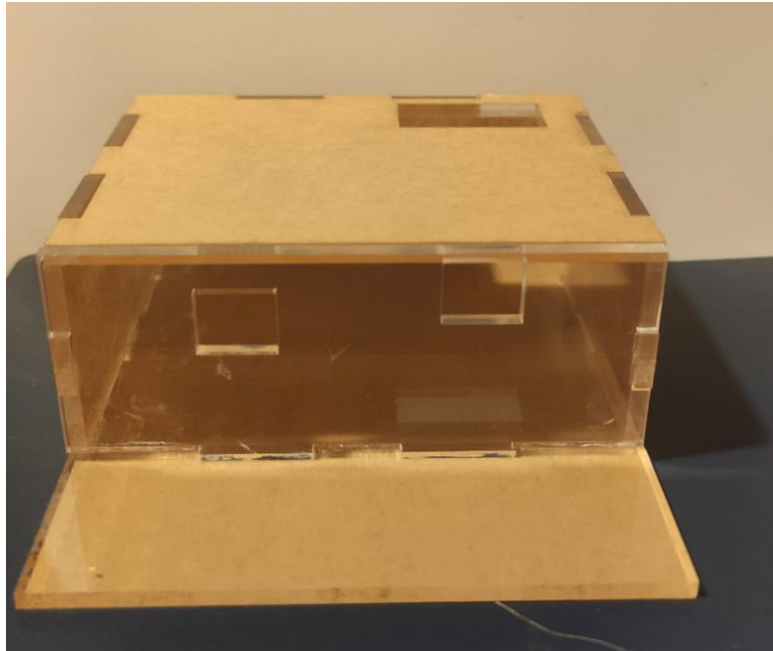
Esta fuente alimenta al circuito electrónico de conversión de corriente a voltaje para el transmisor de presión; este dispositivo realiza una conversión de 110 (V<sub>AC</sub>) a 24 (V<sub>DC</sub>). La fuente de alimentación se observa en la Figura 3.48.



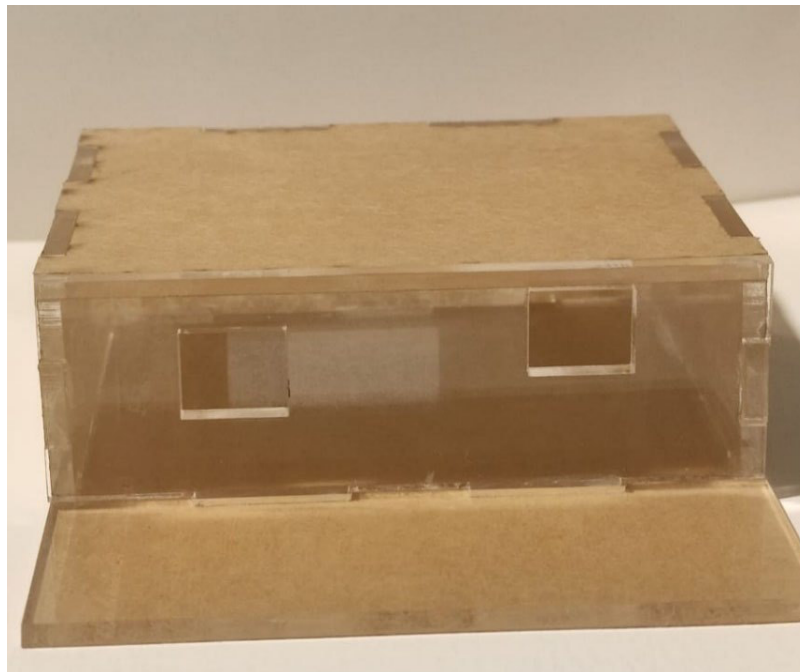
**Figura 3.48** Fuente de 24 (V<sub>DC</sub>)

### **Caja de acrílico**

Después de diseñar las piezas de la caja de acrílico en el programa *SolidWorks*, se enviaron a cortar en una empresa especializada en cortes a láser; posteriormente, se armaron y pegaron cada una de ellas. En la Figura 3.49 y 3.50 se aprecian las cajas terminadas.



**Figura 3.49** Caja de acrílico para el módulo de caudal



**Figura 3.50** Caja de acrílico para el módulo de presión

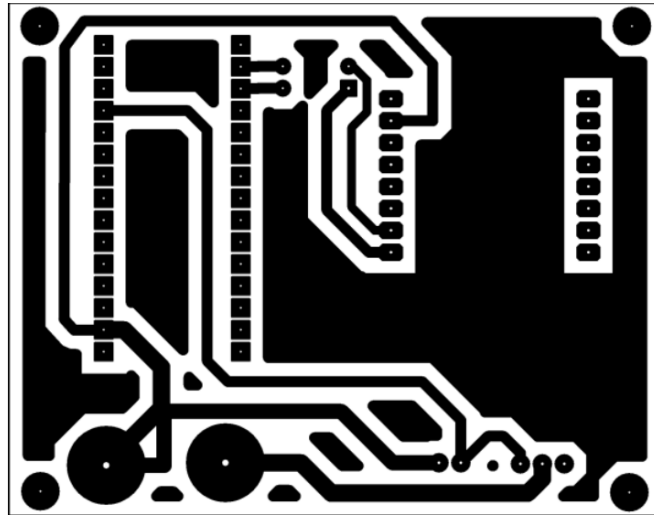
### **Implementación**

La implementación de los módulos de medición remota se dividió en dos partes; primero, la construcción de los módulos de caudal y luego se procedió a la construcción de los módulos de presión. El procedimiento se detalla a continuación:



### Módulos de caudal

Una vez diseñado el diagrama esquemático en *Proteus*, se procedió al diseño de la PCB (Figura 3.51). Se elaboraron los circuitos impresos para poder conectar los diferentes componentes, evitando usar cableado.



**Figura 3.51** Diseño de la PCB para módulo de caudal

El esquema de la PCB se observa en el Anexo 2, lámina 4.

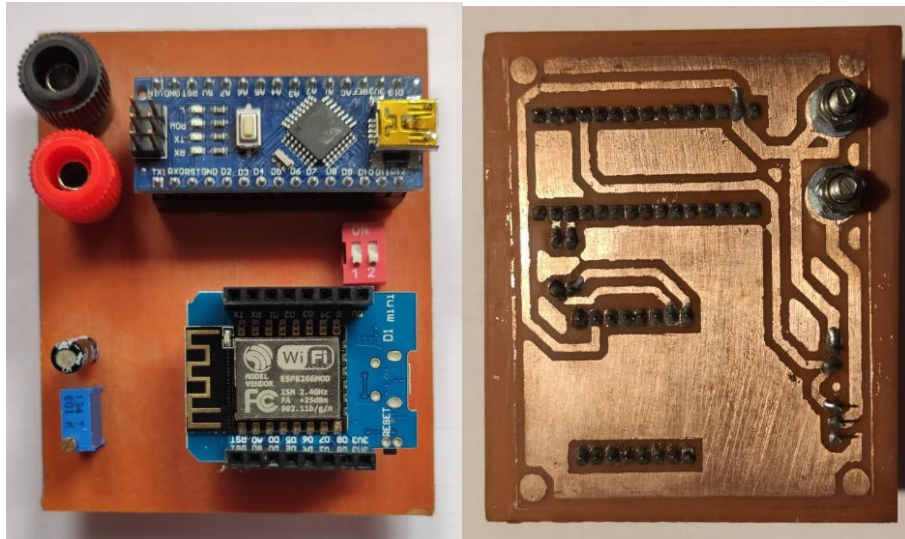
Con el diseño terminado, se procedió al planchado de las placas y a grabar cada una de ellas (Figura 3.52), para luego realizar los agujeros respectivos. Más adelante, se integraron los respectivos elementos electrónicos, tales como: Arduino Nano, *WeMos* D1 mini ESP8266, capacitores, resistencias, borneras tipo banana, entre otros.



**Figura 3.52** Placas para los módulos de caudal

En la Figura 3.53 se observa el montaje de los elementos (parte frontal y parte posterior).

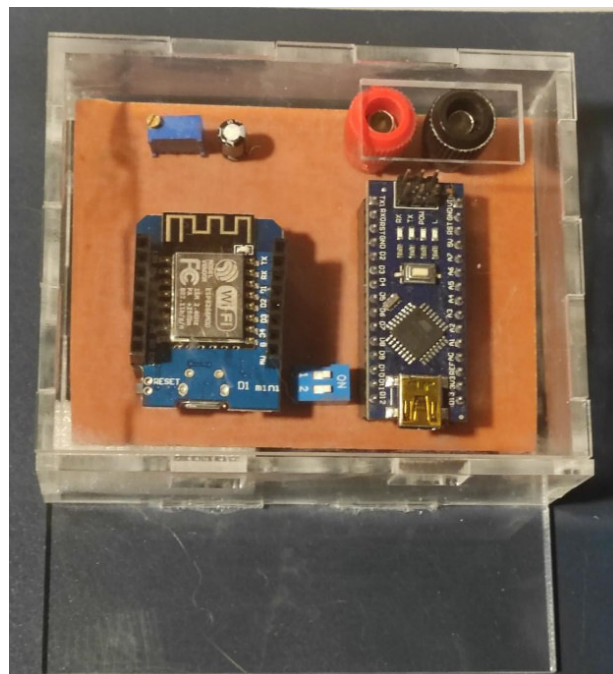




**Figura 3.53** Vista frontal y posterior de la placa del módulo de caudal

Luego, se ensambló la caja uniendo cada pieza con pegamento y asegurando la estructura de acrílico para luego comprobar que la placa encaja correctamente en el armazón.

En la Figura 3.54 se visualiza la caja de acrílico con su respectiva placa.



**Figura 3.54** Caja de acrílico con el módulo de caudal

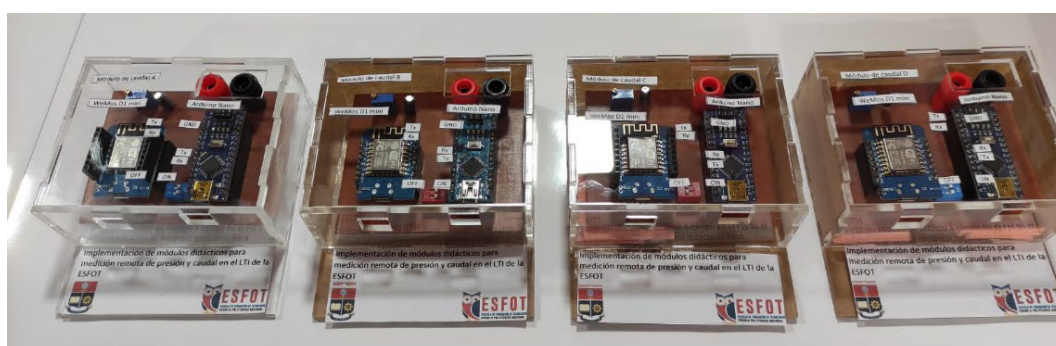
Para finalizar el proceso de construcción, se colocaron etiquetas (Figura 3.55) para identificar los pines más importantes, las cuales se colocaron en la caja, acorde a sus funcionalidades.



**Figura 3.55** Módulo de caudal etiquetado

Además, en la parte inferior se incluyeron el título del proyecto y dos códigos QR, uno para el manual de usuario y otro para descargar la aplicación móvil exclusiva de *Cayenne myDevices*.

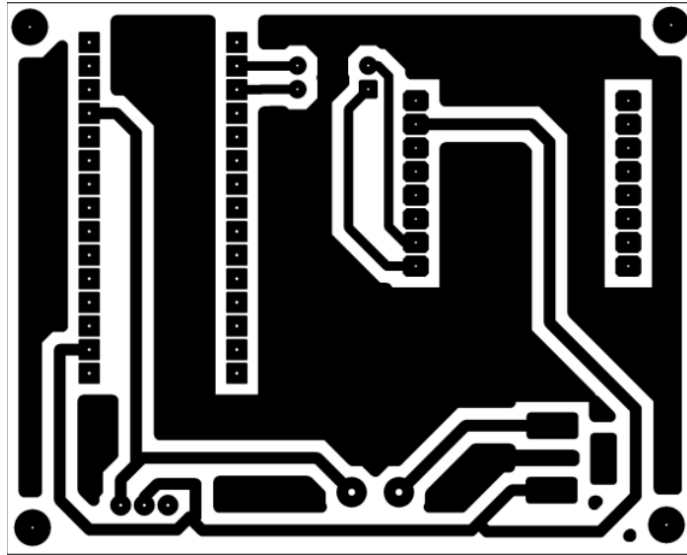
Los cuatro módulos de medición remota de caudal construidos se observan en la Figura 3.56.



**Figura 3.56** Módulos de medición remota de caudal finalizados

### Módulos de presión

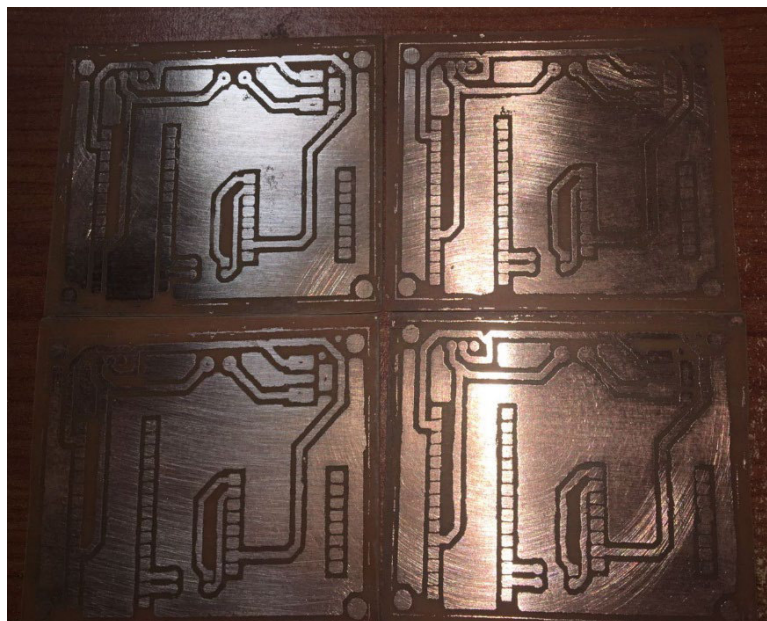
Después del diseño del diagrama esquemático en *Proteus*, se procedió al diseño de la PCB (Figura 3.57). Se elaboraron los circuitos impresos para poder conectar los diferentes componentes, evitando usar cableado.



**Figura 3.57** Diseño de PCB para módulo de presión

El esquema de la PCB se observa en el Anexo 2, lámina 5.

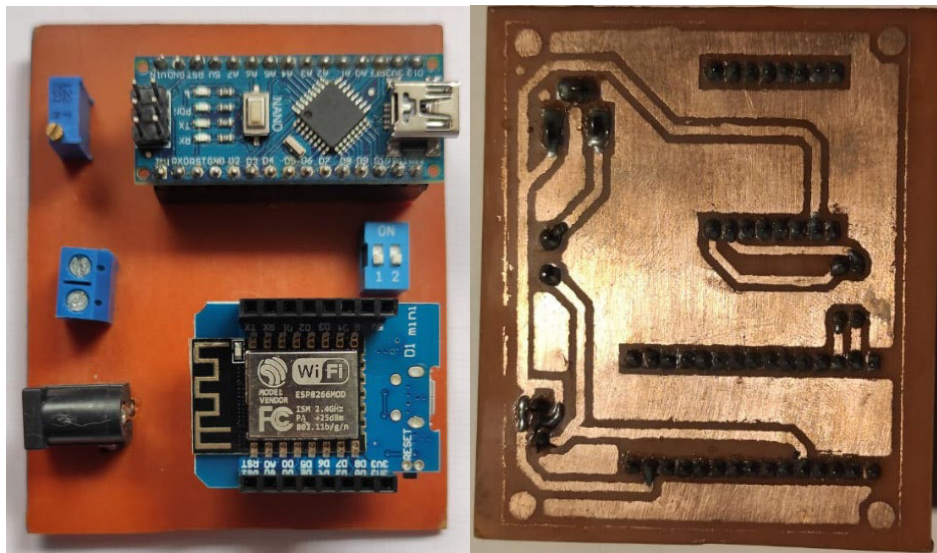
Con el diseño culminado, se procedió al planchado de las placas y a grabar cada una de ellas (Figura 3.58) para luego realizar los agujeros respectivos. Más adelante, se integraron los respectivos elementos electrónicos, tales como: Arduino Nano, *WeMos* D1 mini ESP8266, resistencias, *power jacks*, borneras, entre otros.



**Figura 3.58** Placas para los módulos de presión

En la Figura 3.59 se observa el montaje de los elementos (parte frontal y parte posterior).

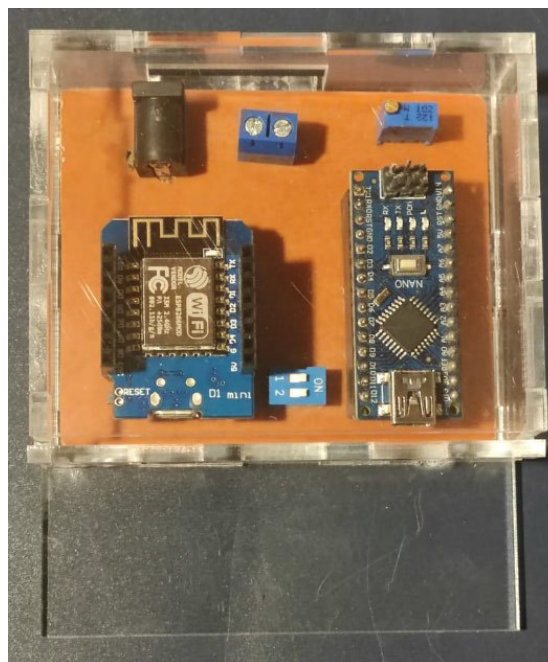




**Figura 3.59** Vista frontal y posterior de la placa del módulo de presión

Luego, se ensambló la caja uniendo cada pieza con pegamento y asegurando la estructura de acrílico para luego comprobar que la placa encaja correctamente en el armazón.

En la Figura 3.60 se visualiza la caja de acrílico con su respectiva placa.



**Figura 3.60** Caja de acrílico con el módulo de presión

Para finalizar el proceso de construcción, se colocaron etiquetas (Figura 3.61) para identificar los pines más importantes, las cuales se colocaron en la caja, acorde a sus funcionalidades.



**Figura 3.61** Módulo de presión etiquetado

Además, en la parte inferior se incluyeron el título del proyecto y dos códigos QR, uno para el manual de usuario y otro para descargar la aplicación móvil exclusiva de *Cayenne myDevices*.

Los cuatro módulos de medición remota de presión construidos se observan en la Figura 3.62.



**Figura 3.62** Módulos de medición remota de presión finalizados

### 3.4 Pruebas y Análisis de Resultados

Después de haber desarrollado los algoritmos de programación y la implementación, se realizaron las siguientes pruebas para verificar el correcto funcionamiento de los módulos didácticos:

#### Pruebas de acondicionamiento

Esta prueba consiste en validar que los instrumentos de medición estén acondicionados correctamente en Arduino IDE; también se comprueba que los datos están siendo transmitidos al módulo *WeMos D1 mini ESP8266* mediante comunicación serial y del módulo Wi-Fi al servidor de *Cayenne myDevices*, obteniendo errores que no sean mayores al 5%.

Los valores que se tomaron en consideración estaban dentro del rango de medida de cada instrumento de medición. En el caso de los módulos de caudal se comparó lo obtenido en la pantalla LCD y la página *web* de *Cayenne myDevices*. Estas pruebas se observan en las Figuras 3.63, 3.64, 3.65 y 3.66.

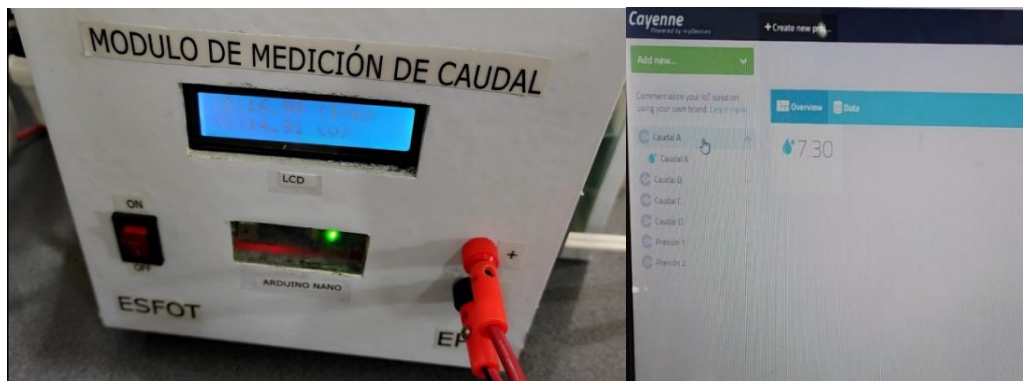


Figura 3.63 Datos del módulo de caudal A

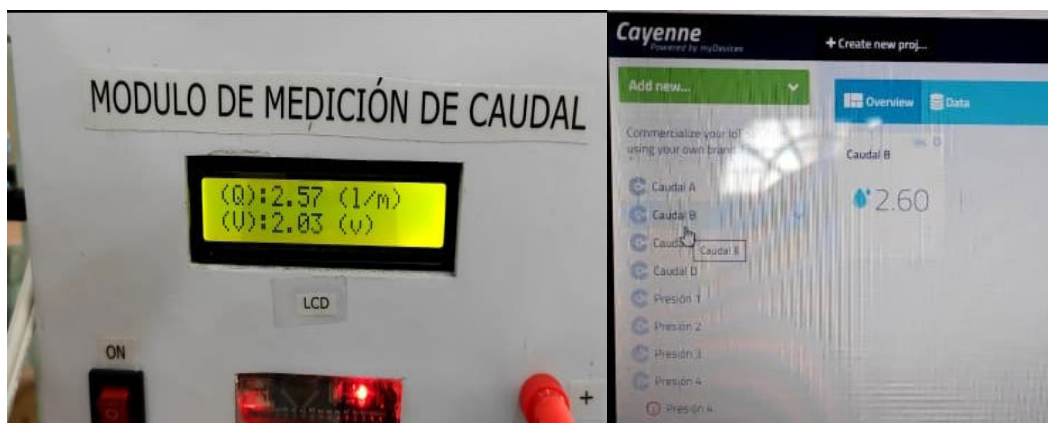
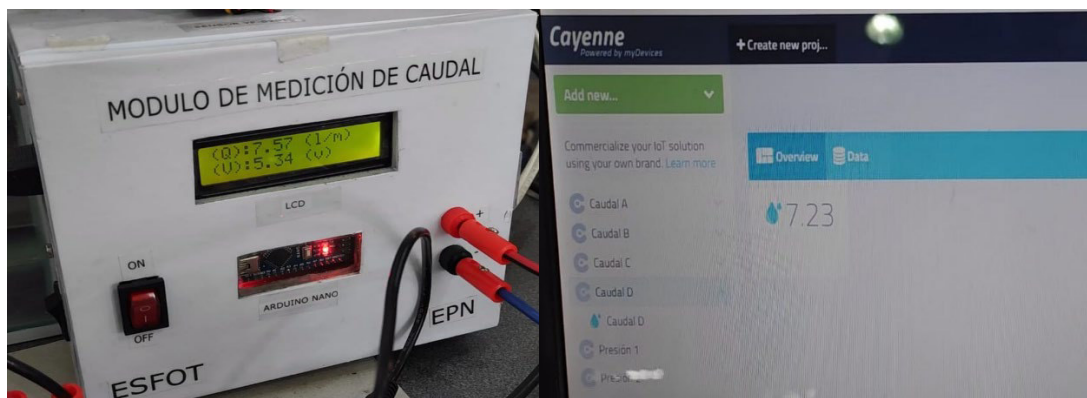


Figura 3.64 Datos del módulo de caudal B





**Figura 3.65** Datos del módulo de caudal C



**Figura 3.66** Datos del módulo de caudal D

Los datos obtenidos en la prueba de acondicionamiento de los módulos de caudal se presentan en las Tablas 3.1, 3.2, 3.3 y 3.4, respectivamente.

**Tabla 3.1** Datos obtenidos para el módulo A de caudal

Caudal medido en el LCD (L/min)	Caudal medido en Cayenne myDevices (L/min)	Error (%)
1,35	1,37	-1,48
2,16	2,14	-0,92
4,73	4,78	1,05
5,54	5,64	-1,80
6,35	6,46	-2.04
7,57	7,24	4,35

**Tabla 3.2** Datos obtenidos para el módulo B de caudal

Caudal medido en el LCD (L/min)	Caudal medido en <i>Cayenne</i> <i>myDevices</i> (L/min)	Error (%)
1,24	1,22	1,61
2,03	2,00	1,47
4,32	4,26	1,38
5,81	5,72	1,54
6,89	6,77	1,74
7,57	7,31	4,35

**Tabla 3.3** Datos obtenidos para el módulo C de caudal

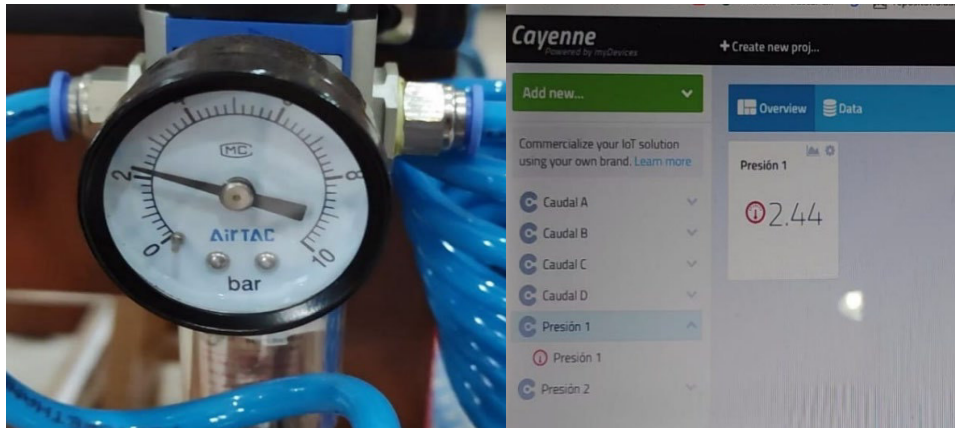
Caudal medido en el LCD (L/min)	Caudal medido en <i>Cayenne</i> <i>myDevices</i> (L/min)	Error (%)
1,49	1,51	-1,34
2,30	2,33	-1,30
3,24	3,27	-0,92
5,13	5,21	-1,55
6,89	7,01	-1,74
7,30	7,47	-2,32

**Tabla 3.4** Datos obtenidos para el módulo D de caudal

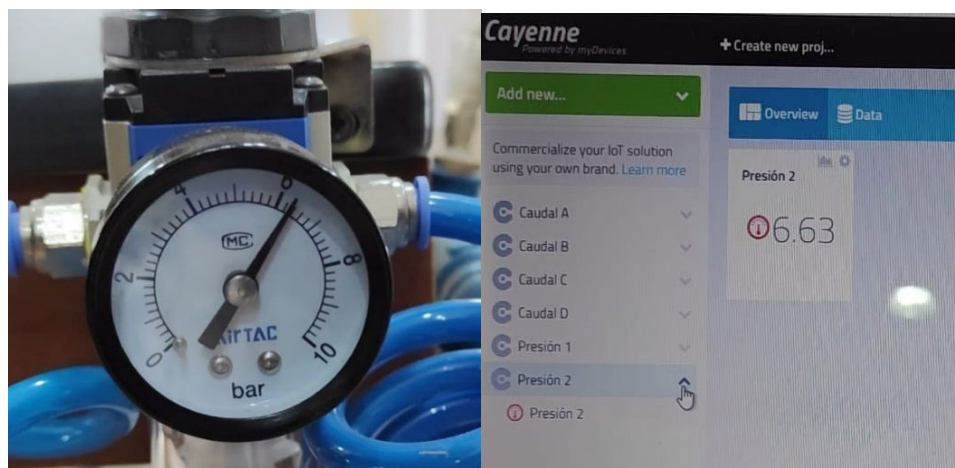
Caudal medido en el LCD (L/min)	Caudal medido en <i>Cayenne</i> <i>myDevices</i> (L/min)	Error (%)
1,22	1,20	1,63
3,11	3,08	0,96
4,46	4,39	1,56
5,95	5,88	1,17
6,35	6,46	-2,04
7,57	7,23	4,49



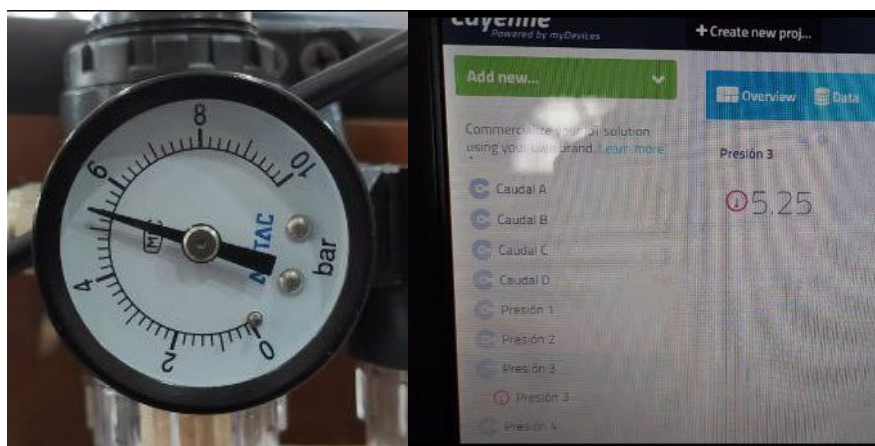
En los transmisores de presión se contrastaron los valores obtenidos en el barómetro que se encuentra dentro del LTI con la página web de *Cayenne myDevices*. Estas pruebas se observan en las Figuras 3.67, 3.68, 3.69 y 3.70.



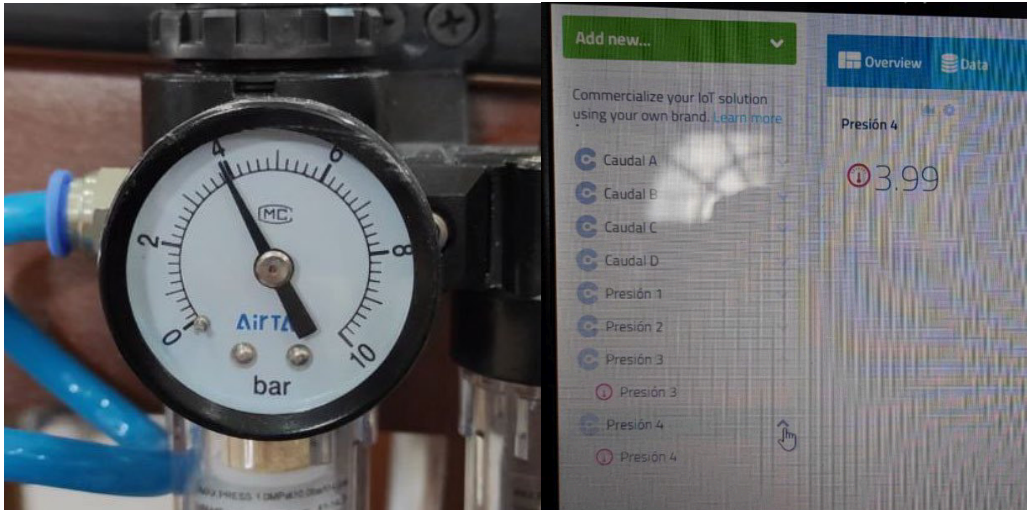
**Figura 3.67** Datos del módulo de presión 1



**Figura 3.68** Datos del módulo de presión 2



**Figura 3.69** Datos del módulo de presión 3



**Figura 3.70** Datos del módulo de presión 4

Los datos obtenidos en la prueba de acondicionamiento de los módulos de presión se presentan en las Tablas 3.5, 3.6, 3.7 y 3.8 respectivamente.

**Tabla 3.5** Datos obtenidos para el módulo 1 de presión

Presión medida en el Barómetro (bar)	Presión medida en Cayenne <i>myDevices</i> (bar)	Error (%)
1,00	1,02	-2,00
2,00	2,06	-3,00
4,20	4,31	-2,55
5,60	5,62	-1,80
6,40	6,46	-0,93
8,00	8,02	-0,25

**Tabla 3.6** Datos obtenidos para el módulo 2 de presión

Presión medida en el Barómetro (bar)	Presión medida en Cayenne <i>myDevices</i> (bar)	Error (%)
1,20	1,22	-1,48
2,80	2,81	-0,35
4,00	4,08	-2,00
5,00	5,06	-1,20
6,40	6,46	-0,93
8,00	8,04	-0,50

**Tabla 3.7** Datos obtenidos para el módulo 3 de presión

Presión medida en el Barómetro (bar)	Presión medida en <i>Cayenne</i> <i>myDevices</i> (bar)	Error (%)
1,40	1,42	-1,42
2,00	2,06	-3,00
4,40	4,51	-2,50
5,20	5,28	-1,53
6,00	6,05	-0,83
8,00	8,01	-0,125

**Tabla 3.8** Datos obtenidos para el módulo 4 de presión

Presión medida en el Barómetro (bar)	Presión medida en <i>Cayenne</i> <i>myDevices</i> (bar)	Error (%)
1,00	1,01	-1,00
2,40	2,46	-2,50
4,00	4,08	-2,00
5,20	5,24	-0,76
6,80	6,91	-1,61
7,60	7,64	-0,52

### Pruebas de transmisión de datos

Para esta prueba se realizó un video como constancia del funcionamiento del sistema, en el cual se muestran las pruebas donde se verifica la correcta transmisión y recepción de datos entre el microcontrolador y el servidor de *Cayenne myDevices*. Se puede visualizar el video mediante el código QR de la Figura 3.71.



**Figura 3.71** Código QR de las pruebas de funcionamiento

En la Tabla 3.9 se presentan los resultados de la prueba de transmisión y recepción de datos para cada módulo de caudal y presión.

**Tabla 3.9** Resultados de la prueba de transmisión y recepción de datos

Sensor	Datos enviados a Cayenne myDevices correctamente	Datos recibidos a Cayenne myDevices correctamente
Módulo caudal A	SI	SI
Módulo caudal B	SI	SI
Módulo caudal C	SI	SI
Módulo caudal D	SI	SI
Módulo presión 1	SI	SI
Módulo presión 2	SI	SI
Módulo presión 3	SI	SI
Módulo presión 4	SI	SI

### Pruebas de almacenamiento de datos

La siguiente prueba se efectuó para validar que los datos mostrados en la página web de *Cayenne* pueden ser registrados y almacenados.

La prueba se ha realizado con todos los dispositivos encendidos; una vez conectados a la red Wi-Fi, *Cayenne* comienza a almacenar inmediatamente los datos. También es posible descargar los datos mediante el botón *Download*, el cual permite obtener un archivo .CSV con la información requerida de cada instrumento de medición. Cabe recalcar que para una mejor visualización de los datos se recomienda la misma página web de *Cayenne*. En las Figuras 3.72 y 3.73 se observan los valores obtenidos para un posterior análisis.

The screenshot shows the Cayenne web interface for a device named 'Presión 1'. The interface includes a navigation bar with 'Overview' and 'Data' tabs, and a 'Download' button. Below the navigation bar, there are filters for 'Live', 'm h d w', '1mo', 'Custom', and 'Query'. The main data table has the following columns: Timestamp, Device, Channel, Sensor Name, Sensor ID, Data T., Unit, and Values. The table contains four rows of data, all showing a value of 0 for the 'Values' column. The data is as follows:

Timestamp	Device	Channel	Sensor Name	Sensor ID	Data T.	Unit	Values
2021-09-28 11:08:42	Presión 1	0	Presión 1	cd8df920-10c8-11ec-8da...			0
2021-09-28 11:08:27	Presión 1	0	Presión 1	cd8df920-10c8-11ec-8da...			0
2021-09-28 11:08:12	Presión 1	0	Presión 1	cd8df920-10c8-11ec-8da...			0
2021-09-28 11:07:57	Presión 1	0	Presión 1	cd8df920-10c8-11ec-8da...			0

At the bottom of the table, there is a link that says 'End of list'.

**Figura 3.72** Almacenamiento de datos de presión

Overview		Data		Caudal A			
Live		m h d w		1mo		Custom Query	
Timestamp	Device...	Channel	Sensor Name	Sensor ID	Data T...	Unit	Values
2021-09-29 11:23:56	Caudal A	0	Caudal A	fac66840-20e1-11ec-ad9...			
2021-09-29 11:23:41	Caudal A	0	Caudal A	fac66840-20e1-11ec-ad9...			6
2021-09-29 11:23:26	Caudal A	0	Caudal A	fac66840-20e1-11ec-ad9...			0.0099999997764826
2021-09-29 11:23:10	Caudal A	0	Caudal A	fac66840-20e1-11ec-ad9...			8
2021-09-29 11:22:55	Caudal A	0	Caudal A	fac66840-20e1-11ec-ad9...			0.050000000745058
2021-09-29 11:22:40	Caudal A	0	Caudal A	fac66840-20e1-11ec-ad9...			0.0099999997764826
2021-09-29 11:22:25	Caudal A	0	Caudal A	fac66840-20e1-11ec-ad9...			0.079999998211861
2021-09-29 11:22:11	Caudal A	0	Caudal A	fac66840-20e1-11ec-ad9...			4
2021-09-29 11:21:55	Caudal A	0	Caudal A	fac66840-20e1-11ec-ad9...			8
2021-09-29 11:21:40	Caudal A	0	Caudal A	fac66840-20e1-11ec-ad9...			0.070000000298023
2021-09-29 11:21:25	Caudal A	0	Caudal A	fac66840-20e1-11ec-ad9...			3
2021-09-29 11:21:10	Caudal A	0	Caudal A	fac66840-20e1-11ec-ad9...			0.10000000149012
2021-09-29 11:20:55	Caudal A	0	Caudal A	fac66840-20e1-11ec-ad9...			0.050000000745058
2021-09-29 11:20:39	Caudal A	0	Caudal A	fac66840-20e1-11ec-ad9...			0.0099999997764826

**Figura 3.73** Almacenamiento de datos de caudal

### 3.5 Manual de Uso

Para mostrar el uso de los módulos didácticos se realizaron dos videos explicativos en los cuales se detallan sus principales características. Se puede acceder a los videos tutoriales mediante los respectivos códigos QR de las Figuras 3.74 y 3.75.



**Figura 3.74** Código QR del manual de uso del módulo de caudal



**Figura 3.75** Código QR del manual de uso del módulo de presión

## 4 CONCLUSIONES Y RECOMENDACIONES

### 4.1 Conclusiones

- Al utilizar la comunicación serial basada en el protocolo UART, se pueden disponer tantos pines para la transmisión y recepción de datos como sea posible configurarlos, lo cual amplía las aplicaciones de los sistemas de comunicación.
- El valor de la frecuencia de corte utilizada se obtuvo al emplear valores ya establecidos de resistencia y capacitancia, en ocasiones es necesario probar con varios valores hasta obtener un filtro pasa bajos que permita una señal de salida continua y estable en función de la PWM generada.
- El *software Proteus* resultó bastante útil para esquematizar los diagramas electrónicos, así como para la simulación del funcionamiento de los módulos, además se plasmaron las conexiones a ser utilizadas en las PCB.
- Se tiene un error de carga del código en el dispositivo *WeMos D1 mini ESP8266* debido que al estar energizado mediante el cable USB y no con la alimentación propia de este dispositivo se crea un conflicto con el envío de datos en los pines TX y RX.
- En la construcción de los módulos se optó por utilizar el Arduino Nano, ya que, por su tamaño y funcionalidades, resultó ser de gran utilidad a la hora de realizar las placas PCB, asimismo cuenta con las entradas analógicas necesarias para el presente proyecto.
- Se prefirió utilizar el módulo Wi-Fi *WeMos D1 mini* que facilitó la conexión a Internet, ya que es una placa que cuenta con una velocidad de conexión bastante rápida y sencilla.
- *Cayenne myDevices*, es una plataforma fácil de usar y amigable con el usuario, ya que, es posible programar varios dispositivos de forma sencilla, debido a que cuenta con los códigos de ejemplo y tutoriales que hacen más fácil la conexión al servidor, además permite conectar hasta 32 dispositivos de manera gratuita.
- Las placas PCB realizadas permitieron un diseño adecuado de los módulos de medición remota, además evitan cualquier cortocircuito o interferencia de algún elemento eléctrico.
- Con las pruebas de acondicionamiento de las señales recibidas desde los instrumentos de medición fue necesario realizar una calibración mediante *software* para obtener un error menor al 5%.

- Al realizar las pruebas de transmisión y recepción de datos al servidor de *Cayenne* se identificó que los datos enviados son correctos; sin embargo, el tiempo de recepción es de 10 segundos desde que los datos ingresan al microcontrolador hasta la nube de *Cayenne myDevices*.

## 4.2 Recomendaciones

- Al momento de usar los módulos de medición remota, es necesario verificar que exista una excelente conexión a Internet; en caso de que en el laboratorio no funcione el Internet, se pueden usar datos móviles para conectar los módulos.
- Para el correcto uso de los módulos, es recomendable seguir a detalle los respectivos manuales de usuario y evitar así posibles fallos a futuro.
- Para ingresar a la plataforma de *Cayenne myDevices*, es preciso tener los módulos de medición remota encendidos y conectados según corresponda, ya que, si no se encuentran bien conectados aparecerá el mensaje que indica el estado del dispositivo.
- Se debe tener un especial cuidado en la alimentación suministrada a los transmisores de presión ya que son dispositivos herméticamente sellados, lo que resulta casi imposible repararlos si son expuestos a una sobretensión.
- Es apropiado informarse acerca de la utilización de la librería “*SoftwareSerial*”, ya que existen varias funciones para iniciar la Comunicación Serial, las cuales no son compatibles con el código a emplear.
- Para solucionar el error en la carga del código al *WeMos D1 mini ESP8266*, resulta útil colocar un *DIP Switch* entre los pines de TX y RX del Arduino Nano y los pines de TX y RX del *WeMos D1 mini ESP8266*, ya que así se puede desconectar temporalmente un pin del *WeMos D1 mini ESP8266* hasta que se pueda cargar el código empleado.
- Es recomendable que para el análisis de los datos que se registran y almacenan en la página *web* de *Cayenne* sean visualizados en la propia página, ya que tienen una mejor organización y distribución de los datos.
- En proyectos futuros, en donde existan más variables físicas que se necesiten incorporar al sistema, se recomienda el uso de un microcontrolador con un mayor número de entradas y salidas, ya que el Arduino Nano es muy limitado en este sentido.
- Si es necesario se puede lograr una comunicación *Duplex* entre el Arduino y la plataforma *Cayenne* utilizando las librerías de las que dispone Arduino IDE para *Cayenne*.

## 5 REFERENCIAS BIBLIOGRÁFICAS

- [1] Escuela de Formación de Tecnólogos, [En línea]. Available: <https://esfot.epn.edu.ec/index.php/oferta-academica/laboratorios>.
- [2] J. H. S. Alonso, «Universidad Militar Nueva Granada,» 23 noviembre 2018. [En línea]. Available: <https://repository.unimilitar.edu.co/bitstream/handle/10654/20935/SanchezAlonsojaimehumberto2018.pdf?sequence=1&isAllowed=y>. [Último acceso: 24 julio 2020].
- [3] J. F. C. Angel Rodriguez, «Revista ESPACIOS,» 03 03 2018. [En línea]. Available: <https://www.revistaespacios.com/a18v39n22/a18v39n22p30.pdf>. [Último acceso: 14 05 2021].
- [4] Aliexpress, «Aliexpress,» [En línea]. Available: <https://www.aliexpress.com/item/32820069575.html>. [Último acceso: 15 06 2021].
- [5] F. Martínez, «Universidad Politecnica de Cartagena,» 30 08 2019. [En línea]. Available: <https://repositorio.upct.es/bitstream/handle/10317/8000/tfm-mardis.pdf?sequence=1&isAllowed=y>. [Último acceso: 15 06 2021].
- [6] «Inventable.eu,» [En línea]. Available: <https://www.inventable.eu/ley-de-ohm/>. [Último acceso: 24 08 2021].
- [7] A. Vu, «Az delivery,» 28 11 2018. [En línea]. Available: <https://www.az-delivery.de/es/blogs/azdelivery-blog-fur-arduino-und-raspberry-pi/arbeiten-mit-dem-cayenne-dashboard-teil-1>. [Último acceso: 24 08 2021].
- [8] E. Gómez, «Ricón Ingenieril,» 27 06 2017. [En línea]. Available: <https://www.rinconingenieril.es/filtros-paso-bajos-y-paso-altos/>. [Último acceso: 24 08 2021].
- [9] Arduino Store, «Arduino Store,» [En línea]. Available: <https://store.arduino.cc/products/arduino-nano>. [Último acceso: 23 06 2021].



- [10] ResearchGate, «ResearchGate,» [En línea]. Available: [https://www.researchgate.net/figure/Wemos-D1-mini-module-ESP-8266\\_fig2\\_333686151](https://www.researchgate.net/figure/Wemos-D1-mini-module-ESP-8266_fig2_333686151). [Último acceso: 12 07 2021].
- [11] Amazon, «Amazon,» [En línea]. Available: <https://www.amazon.com/-/es/conexi%C3%B3n-conector-amplificador-potencia-pulgadas/dp/B0818VZX4Z>. [Último acceso: 14 07 2021].
- [12] Ag Electrónica, «Ag Electrónica,» [En línea]. Available: <https://www.agelectronica.com/detalle.php?p=DC-005-2.1R>. [Último acceso: 20 07 2021].
- [13] WEMOS, «WEMOS,» [En línea]. Available: [https://www.wemos.cc/en/latest/d1/d1\\_mini.html](https://www.wemos.cc/en/latest/d1/d1_mini.html). [Último acceso: 28 06 2021].
- [14] L. Llamas, «luisllamas.es,» 17 Septiembre 2020. [En línea]. Available: <https://www.luisllamas.es/programar-esp8266-con-el-ide-de-arduino/>. [Último acceso: 2016].
- [15] Á. Robledano, «OpenWebinars,» 17 Septiembre 2020. [En línea]. Available: <https://openwebinars.net/blog/que-es-tcpip/>. [Último acceso: 18 Junio 2019].
- [16] J. L. Martinez Valencia, «Diseño y desarrollo de un sistema de medición y monitoreo remoto,» 30 Agosto 2020. [En línea]. Available: <https://core.ac.uk/download/pdf/71398882.pdf>. [Último acceso: 15 Octubre 2016].

## **ANEXOS**

## **ANEXO 1: CERTIFICADO DE FUNCIONAMIENTO**



# ESCUELA POLITECNICA NACIONAL

Campus Politécnico "J. Rubén Orellana R

Quito, 12 de octubre de 2021

## CERTIFICADO DE FUNCIONAMIENTO DE PROYECTO DE TITULACIÓN

Yo, *Alan Daniel Cuenca Sánchez*, docente a tiempo completo de la Escuela Politécnica Nacional y como director de este trabajo de titulación, certifico que he constatado el correcto funcionamiento de los módulos didácticos para la medida remota de caudal y presión en el LTI - ESFOT, el cual fue implementado por los estudiantes Roger Paúl Morejón Herrera y Jonathan Patricio Villagómez Villamarín.

El proyecto cumple con los requerimientos de diseño y parámetros necesarios para que los usuarios puedan utilizar los módulos didácticos en las instalaciones de la ESFOT con seguridad para los equipos y las personas.

**DIRECTOR**

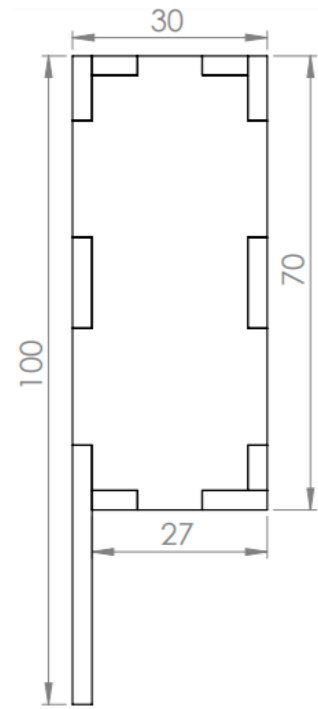
Ing. Alan Daniel Cuenca Sánchez, Msc.

---

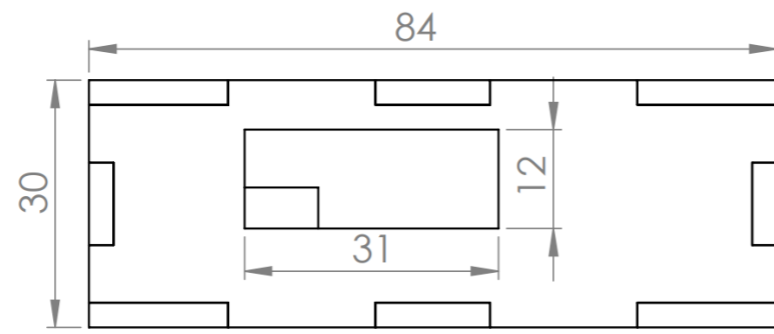
Ladrón de Guevara E11-253, Escuela de Formación de Tecnólogos, Oficina 22. EXT: 2736  
email: alan.cuenca@epn.edu.ec Quito-Ecuador

## **ANEXO 2: PLANOS Y ESQUEMAS**

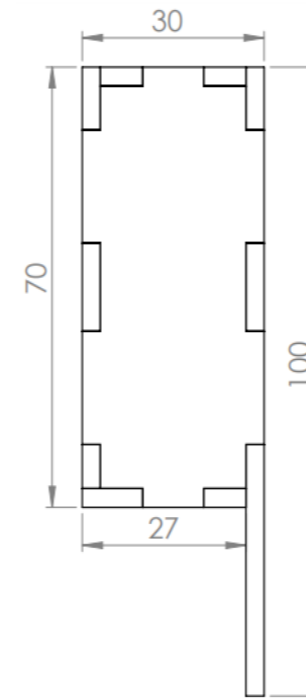
Vista lateral izquierda



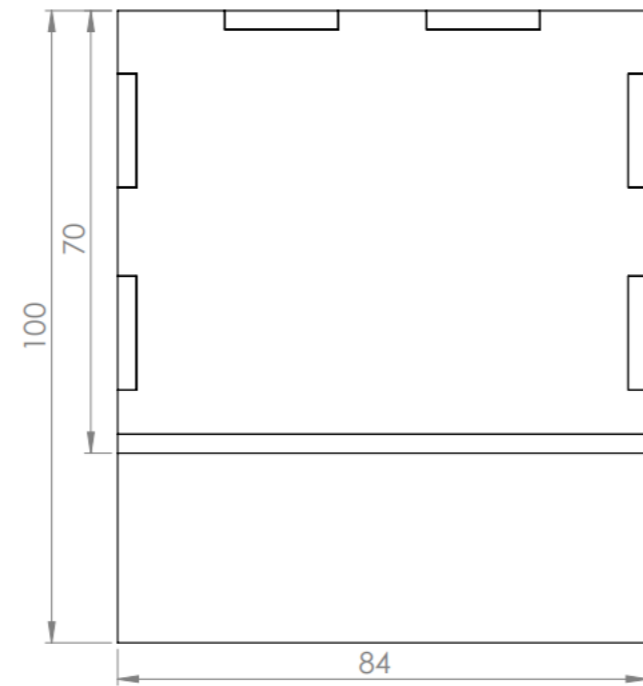
Vista Posterior



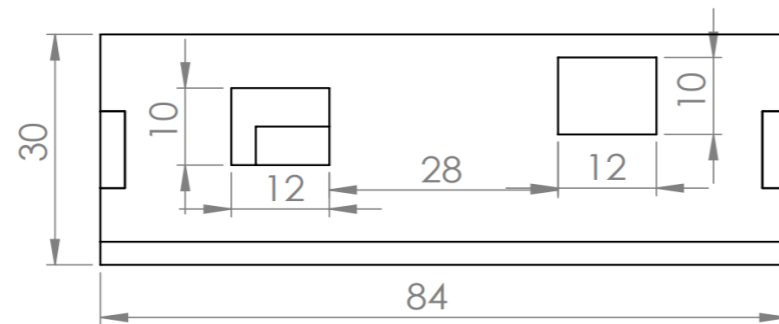
Vista lateral derecha



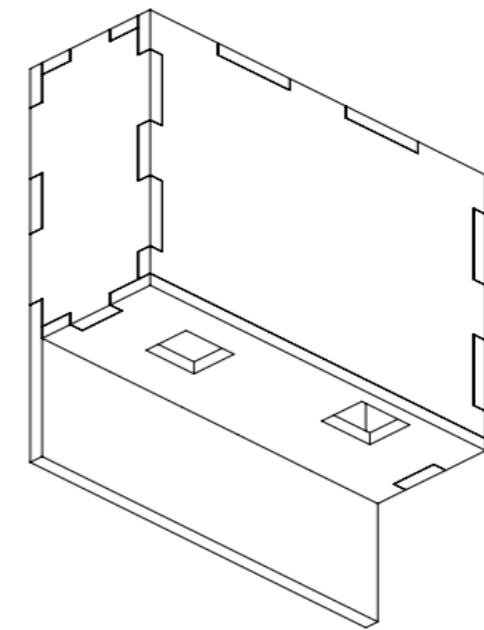
Vista Superior



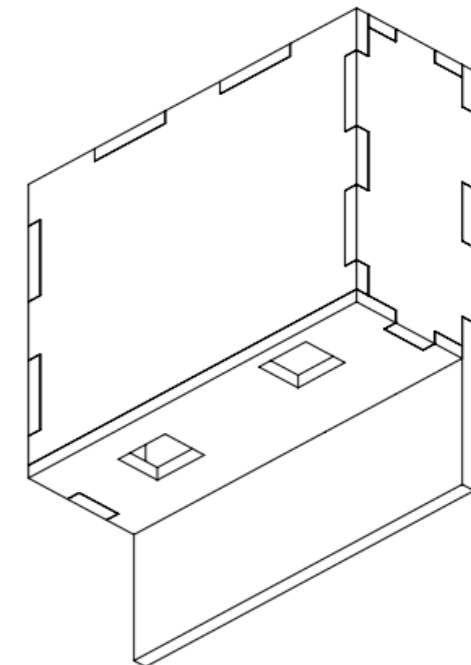
Vista Frontal



Vista isométrica izquierda



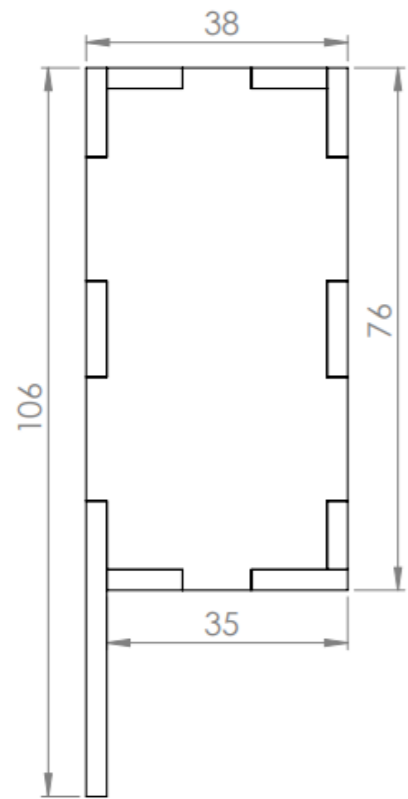
Vista isométrica derecha



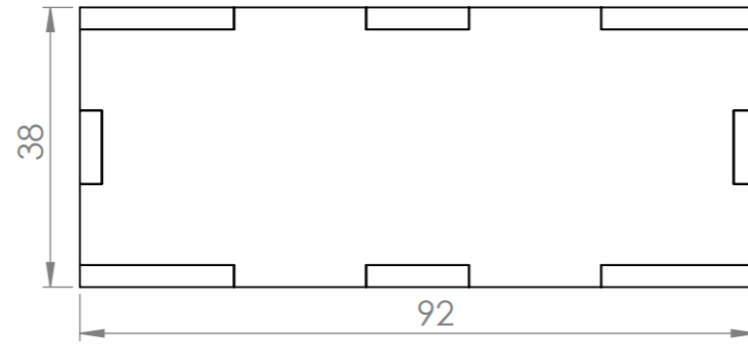
**ESCUELA POLITÉCNICA NACIONAL - ESFOT**

MÓDULO DE PRESIÓN (CAJA)	TRATAMIENTO TÉRMICO: N/A	MATERIAL: Acrílico
	ESCALA: 1:1	FORMATO: A3
DISEÑADO POR: Morejón Roger Villagómez Jonathan	FECHA: 12/10/2021	LÁMINA: 1

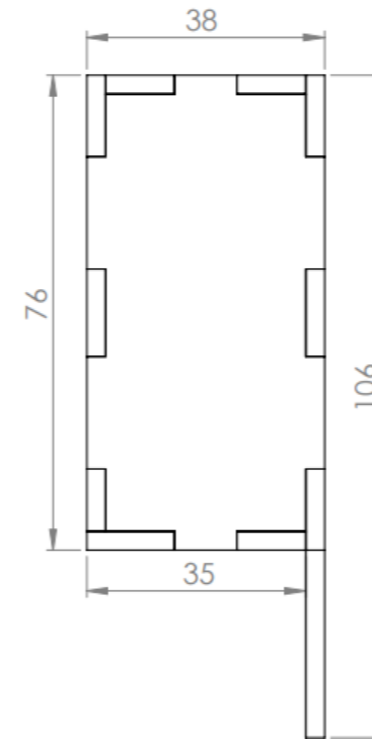
Vista lateral izquierda



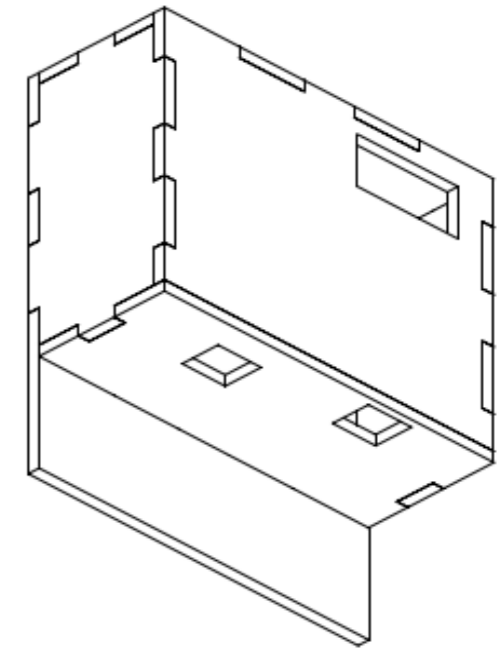
Vista Posterior



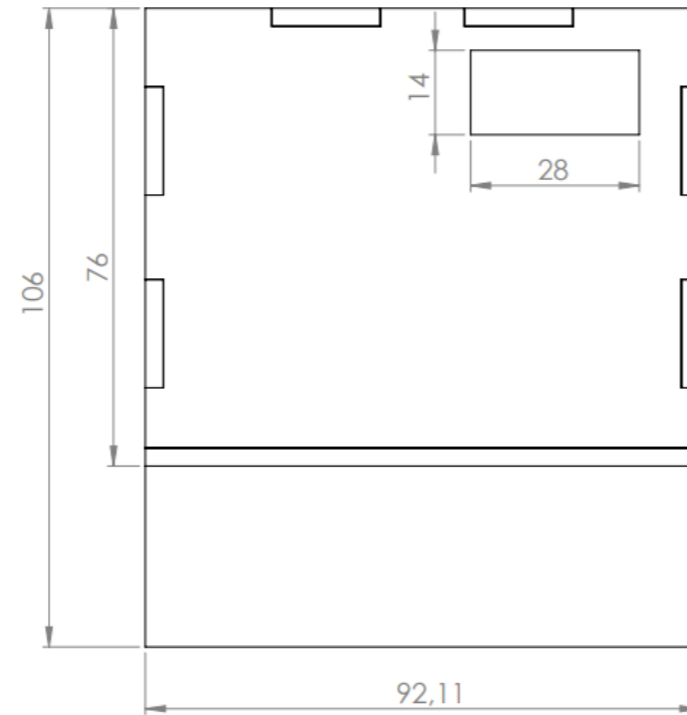
Vista lateral derecha



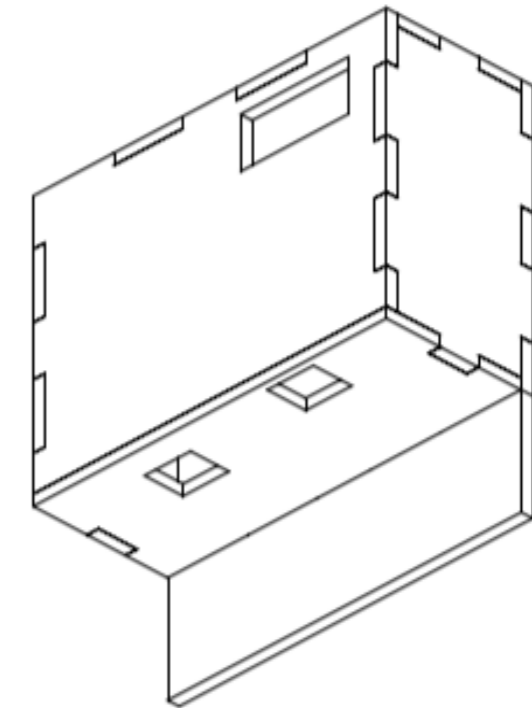
Vista isométrica izquierda



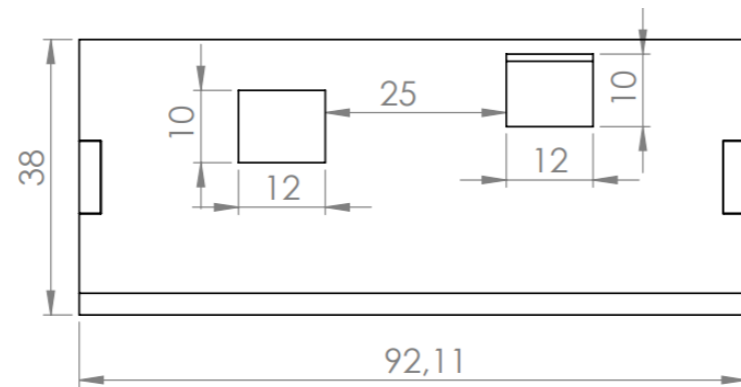
Vista Superior



Vista isométrica derecha



Vista Frontal



**ESCUELA POLITÉCNICA NACIONAL - ESFOT**

MÓDULO DE CAUDAL (CAJA)	TRATAMIENTO TÉRMICO: N/A	MATERIAL: Acrílico
	ESCALA: 1:1	FORMATO: A3
DISEÑADO POR: Morejón Roger Villagómez Jonathan	FECHA: 12/10/2021	LÁMINA: 2

Diagrama esquemático de presión

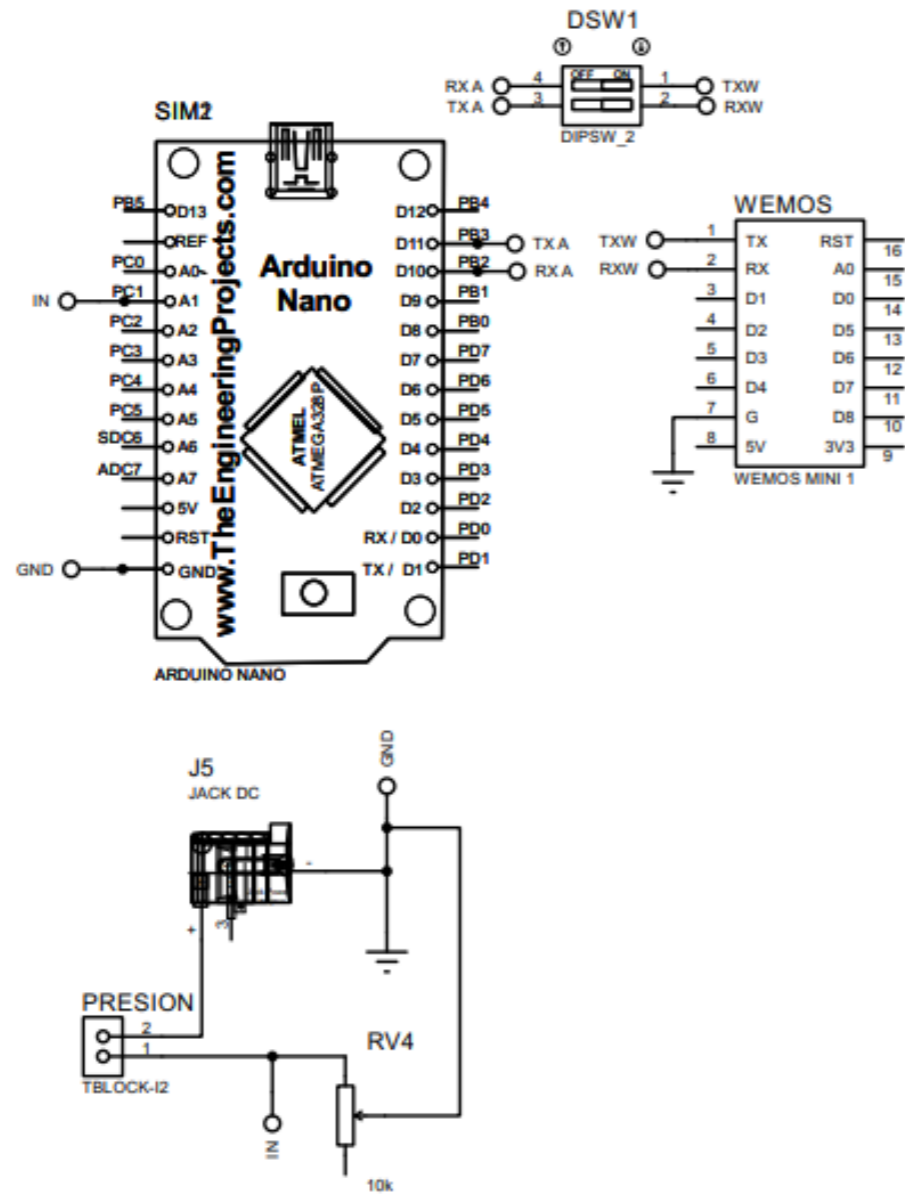
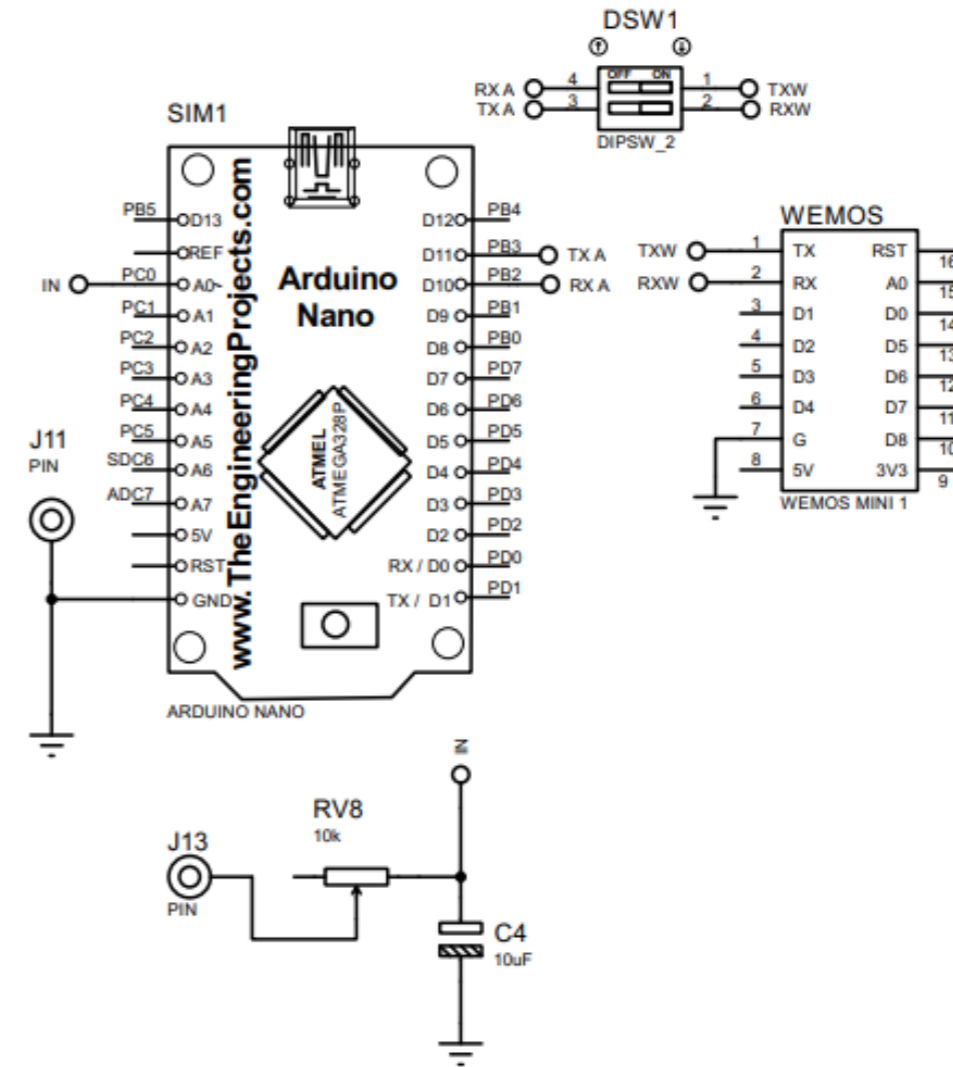


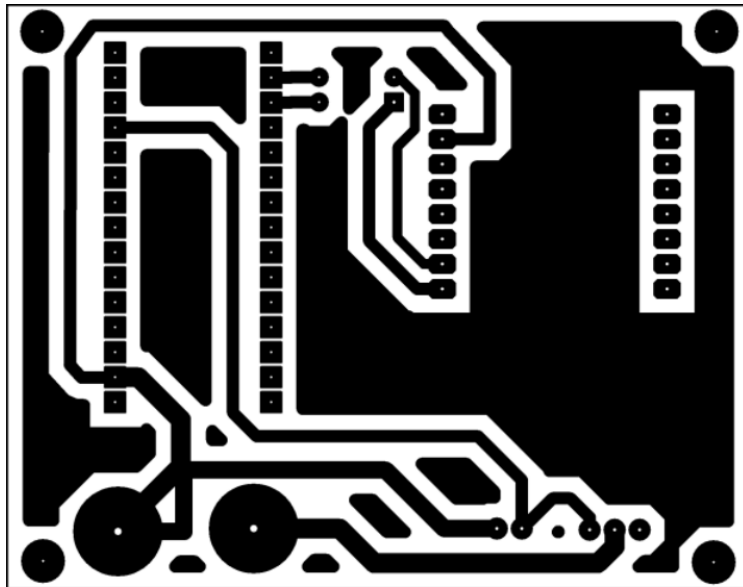
Diagrama esquemático de caudal



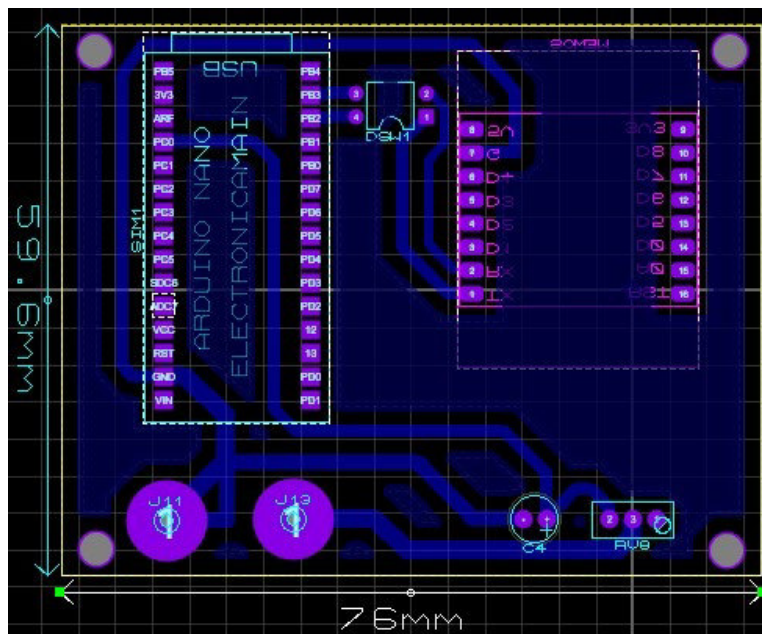
**ESCUELA POLITÉCNICA NACIONAL - ESFOT**

DIAGRAMAS ESQUEMÁTICOS DE PRESIÓN Y CAUDAL	TRATAMIENTO TÉRMICO: N/A	MATERIAL: Acrílico
	ESCALA: 1:1	FORMATO: A3
DISEÑADO POR: Morejón Roger Villagómez Jonathan	FECHA: 12/10/2021	LÁMINA: 3





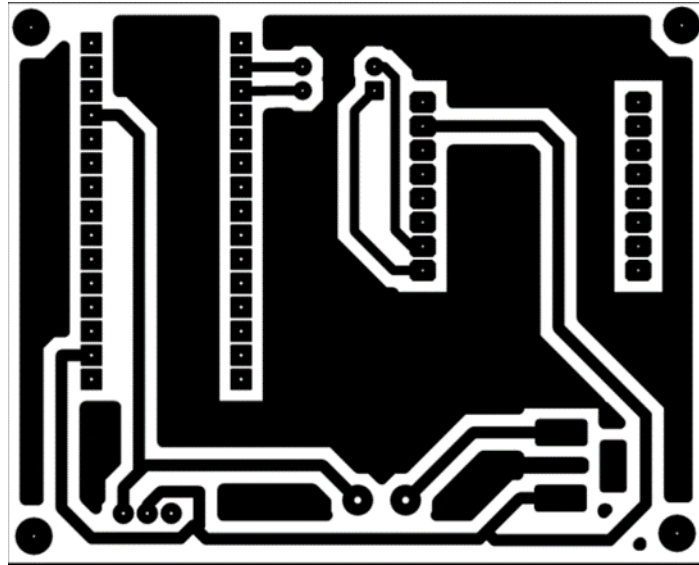
Diseño PCB de caudal para impresión



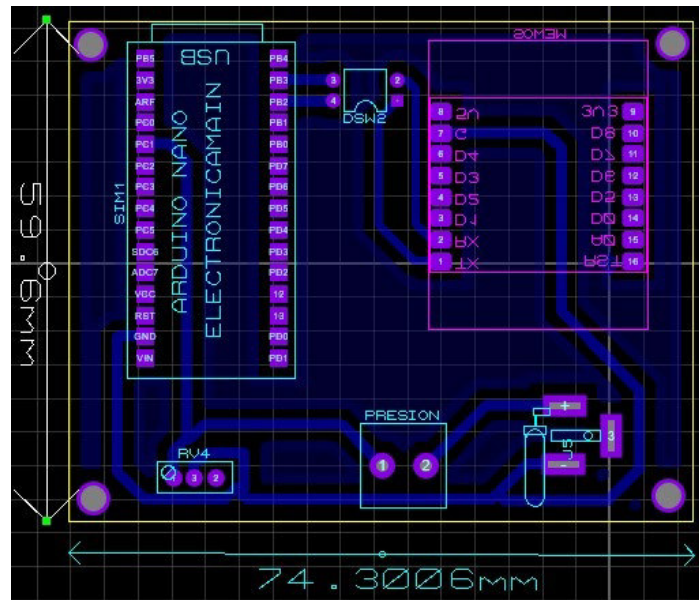
Diseño PCB de caudal en *Proteus*

## ESCUELA POLITÉCNICA NACIONAL - ESFOT

DIAGRAMA PCB PARA MÓDULO DE CAUDAL	TRATAMIENTO TÉRMICO: N/A	MATERIAL: Eléctrico
	ESCALA: 1:1	FORMATO: A4
DISEÑADO POR: Morejón Roger Villagómez Jonathan	FECHA: 12/10/2021	LÁMINA: 4



Diseño PCB de presión para impresión



Diseño PCB de presión en *Proteus*

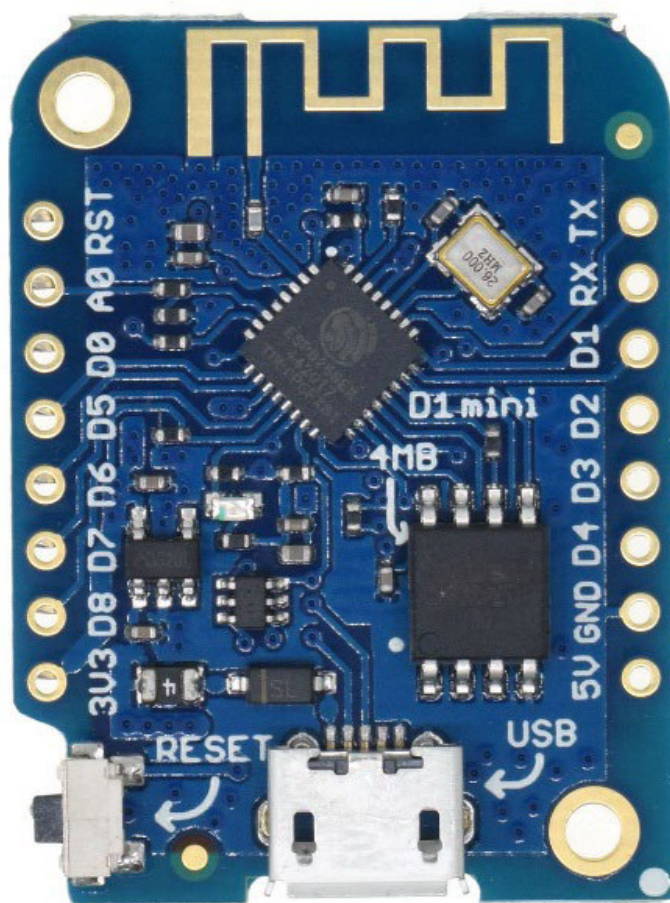
## ESCUELA POLITÉCNICA NACIONAL - ESFOT

DIAGRAMA PCB PARA MÓDULO DE PRESIÓN	TRATAMIENTO TÉRMICO: N/A	MATERIAL: Eléctrico
	ESCALA: 1:1	FORMATO: A4
DISEÑADO POR: Morejón Roger Villagómez Jonathan	FECHA: 12/10/2021	LÁMINA: 5

## **ANEXO 3: ESPECIFICACIONES TÉCNICAS**

**Tabla A.1** Especificaciones técnicas *WeMos D1 mini* [13]

Datos	Valores
Voltaje de operación	3,3 (V <sub>DC</sub> )
Pines digitas E/S	11
Pines de entrada analógica	1
Frecuencia	80/160 (Mhz)
Memoria Flash	4 (Mbytes)
Tamaño	34,2*25,6 (mm)
Peso	3 (g)



**Figura A.1** Pines de *WeMos D1 mini* [13]

**Tabla A.2** Distribución de pines *WeMos D1 mini* [13]

Pin	Función	Pin ESP8266
TX	TXD	TXD
RX	RXD	RXD
A0	Entrada analógica	A0
D0	ES	GPIO16
D1	ES, SCL	GPIO5
D2	ES, SDA	GPIO4
D3	ES, 10k <i>Pull-up</i>	GPIO0
D4	ES, 10k <i>Pull-up</i>	GPIO2
D5	ES, SCK	GPIO14
D6	ES, MISO	GPIO12
D7	ES, MOSI	GPIO13
D8	ES, 10k <i>Pull-up</i> , SS	GPIO15
G	Tierra	GND
5(V <sub>DC</sub> )	5(V <sub>DC</sub> )	-
3,3(V <sub>DC</sub> )	3,3(V <sub>DC</sub> )	3,3(V <sub>DC</sub> )
RST	Reinicio	RST

### **Código de programación de Arduino para los módulos de caudal**

```
#include <SoftwareSerial.h> // Librería para emular un puerto serial físico

SoftwareSerial mySerial(10, 11); // Pines RX, TX, pines digitales para emular un puerto
                                de comunicación

int sensor;

float caudal;

float suma;

float variable;

void setup() {

  Serial.begin(9600);

  mySerial.begin(9600); // Puerto serial emulado igualmente inicia en 9600 baudios
```

```

}

void loop() {

  suma=0;

  for (int i=0; i<5; i++){

    sensor=analogRead(A0); // lee entrada analoga 0-4v

    caudal=map(sensor,0,819,0,1023); // transforma de voltaje a unidades crudas

    variable=(0.00689*caudal)-0.4654; //ecuacion de la recta entrada vs salida

    suma=variable+suma; //promedio de lecturas

    delay(300);

  }

  //Serial.println(variable);    // Publica en el monitor serial el CAUDAL

  mySerial.println(variable);    // Usa el puerto serial emulado para enviar el caudal

  delay(500);                    //Toma la medición cada medio segundo

}

```

### **Código de programación de Arduino para los módulos de presión**

```

#include <SoftwareSerial.h> // Librería para emular un puerto serial físico

SoftwareSerial mySerial(10, 11); // Pines RX, TX, pines digitales para emular un puerto
                                de comunicación

float sensor1;

float presion;

int suma;

float variable1;

void setup() {

  Serial.begin(9600);

  mySerial.begin(9600); //Puerto serial emulado igualmente inicia en 9600 baudios

}

```

```

void loop() {

suma=0;

for (int i=0; i<5; i++){

sensor1=analogRead(A1); // lee entrada análoga 0-4v

variable1=0.01281*sensor1-2.88; //ecuación de la recta entrada vs salida

suma=variable1+suma;//promedio de lecturas

delay(300);

}

//Serial.println(variable1);    //Publica en el monitor serial el CAUDAL

mySerial.println(variable1);    //Usa el puerto serial emulado para enviar el caudal

delay(300);    //Toma la medición cada medio segundo

```

### **Código de programación de WeMos ESP8266 para los módulos de caudal**

```

////////SELECCIONAR LA PLACA WIFI A UTILIZAR////////

//-----

#define CAYENNE_DEBUG

#define CAYENNE_PRINT Serial

//-----

#define IOT_MOD_WIZNET_COMPILE

#define IOT_MOD_8266_COMPILE

#define IOT_MOD_ESP32_COMPILE

//-----

#include "Comandos.h"

//-----

```

```
// Informacion de la red WIFI

char ssid[] = "laboratorio_LTI"; //NOMBRE DE LA RED

char wifiPassword[] = "12345678a"; //CLAVE DE LA RED

// Datos de la cuenta registrada en CAYENNE

char username[] = "aa39b380-ec27-11eb-b767-3f1a8f1211ba";

char password[] = "e949014213733487ead9268efb7eaf571c2933f7";

char clientID[] = "d7ee7730-20e2-11ec-ad90-75ec5e25c7a4";

//"01b140e0-d591-11eb-883c-638d8ce4c23d";

float caudal1;

void setup() {

    Serial.begin(9600); //Inicia el puerto serial en 9600 baudios

    ///ENvia los parametros ingresados de Cayenne

    #ifdef IOT_MOD_WIZNET_COMPILE

    Cayenne.begin(username, password, clientID);

    #endif

    #ifdef IOT_MOD_8266_COMPILE

    Cayenne.begin(username, password, clientID, ssid, wifiPassword);

    #endif

    #ifdef IOT_MOD_ESP32_COMPILE

    Cayenne.begin(username, password, clientID, ssid, wifiPassword);
```



```
#endif

CAYENNE_OUT(1); //Selecciona la salida 1
}

void loop() {
    if (Serial.available() > 0) {
        caudal1 = Serial.parseFloat();
    }

    delay(1);

    Cayenne.loop(); //Selecciona el metodo loop de Cayenne para ejecutar
informacion
}

CAYENNE_OUT_DEFAULT(){

    ////Cayenne permite hasta 31 canales para guardar informacion////

    Cayenne.virtualWrite(CANAL_CYN_00, caudal1); //Envia la temperatura por el
CANAL 0

    //Cayenne.virtualWrite(CANAL_CYN_01, Humedad); //Envia la temperatura
por el CANAL 1

    //CAYENNE_PRINT.println("Enviando Datos a Cayenne"); //Imprime el texto de
datos enviados a Cayenne
}
```

## Código de programación de WeMos ESP8266 para los módulos de presión

```
////////SELECCIONAR LA PLACA WIFI A UTILIZAR////////  
  
//-----  
  
//#define CAYENNE_DEBUG  
  
#define CAYENNE_PRINT Serial  
  
//-----  
  
//#define IOT_MOD_WIZNET_COMPILE  
  
#define IOT_MOD_8266_COMPILE  
  
//#define IOT_MOD_ESP32_COMPILE  
  
//-----  
  
#include "Comandos.h"  
  
//-----  
  
  
  
// Informacion de la red WIFI  
  
char ssid[] = "laboratorio_LTI"; //NOMBRE DE LA RED  
  
char wifiPassword[] = "12345678a"; //CLAVE DE LA RED  
  
  
  
// Datos de la cuenta registrada en CAYENNE  
  
char username[] = "aa39b380-ec27-11eb-b767-3f1a8f1211ba";  
  
char password[] = "e949014213733487ead9268efb7eaf571c2933f7";  
  
char clientID[] = "638b9290-20df-11ec-9f5b-45181495093e";  
  
float presion1;  
  
  
void setup() {  
  
    Serial.begin(9600); //Inicia el puerto serial en 9600 baudios
```

```
///  
//ENvia los parametros ingresados de Cayenne  
  
#ifndef IOT_MOD_WIZNET_COMPILE  
  
Cayenne.begin(username, password, clientID);  
  
#endif  
  
  
#ifndef IOT_MOD_8266_COMPILE  
  
Cayenne.begin(username, password, clientID, ssid, wifiPassword);  
  
#endif  
  
  
#ifndef IOT_MOD_ESP32_COMPILE  
  
Cayenne.begin(username, password, clientID, ssid, wifiPassword);  
  
#endif  
  
  
CAYENNE_OUT(1); //Selecciona la salida 1  
  
}  
  
  
void loop() {  
  
    if (Serial.available() > 0) {  
  
        presion1 = Serial.parseFloat();  
  
    }  
  
  
    delay(1);  
  
    Cayenne.loop(); //Selecciona el metodo loop de Cayenne para ejecutar  
informacion  
  
}
```

```
CAYENNE_OUT_DEFAULT(){
```

```
    /////Cayenne permite hasta 31 canales para guardar informacion/////
```

```
    Cayenne.virtualWrite(CANAL_CYN_00, presion1);    //Envia la temperatura por el  
CANAL 0
```

```
    //Cayenne.virtualWrite(CANAL_CYN_01, Humedad);    //Envia la temperatura  
por el CANAL 1
```

```
    //CAYENNE_PRINT.println("Enviando Datos a Cayenne"); //Imprime el texto de  
datos enviados a Cayenne
```

```
}
```