

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

### **IMPLEMENTACIÓN DE UN SISTEMA DE TELEMETRÍA PARA UN MÓDULO TERMOSOLAR DE AGUA CALIENTE**

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

**Angelo Leonardo Armas Rea**

angelo.armas@epn.edu.ec

**Emerson Saúl Viteri Chávez**

emerson.viteri@epn.edu.ec

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

alan.cuenca@epn.edu.ec

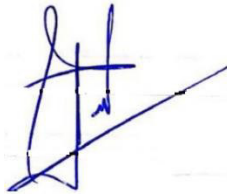
**CODIRECTOR: ING. FABIO MATÍAS GONZÁLEZ GONZÁLEZ, MSc.**

fabio.gonzalez@epn.edu.ec

**Quito, octubre de 2021**

# CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por los Srs. Armas Rea Angelo Leonardo y Viteri Chávez Emerson Saúl como requerimiento parcial a la obtención del título de TECNÓLOGO EN ELECTRÓNICA Y TELECOMUNICACIONES, bajo nuestra supervisión:



---

**Ing. Alan Cuenca**

DIRECTOR DEL PROYECTO



---

**Ing. Fabio Gorizález**

CODIRECTOR DEL PROYECTO

## DECLARACIÓN

Nosotros Armas Rea Angelo Leonardo con CI: 1725052425 y Viteri Chávez Emerson Saúl con CI: 1722520879, 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.



---

**Angelo Armas**

ESTUDIANTE



---

**Emerson Viteri**

ESTUDIANTE

## **DEDICATORIA**

Este proyecto está dedicado especialmente a mi familia por brindarme el apoyo necesario, y apoyarme con su conocimiento, habilidades, y valores. Tanto en mi vida de estudiante, profesional y social.

**Angelo Armas**

## **AGRADECIMIENTO**

Gracias a Dios por permitirme tener y disfrutar a mi familia, agradezco a mis padres por la paciencia, amor y ayuda en cada ciclo de mi vida.

No ha sido sencillo el camino, pero agradezco por los aportes de conocimiento por parte de los docentes de la Escuela Politécnica Nacional, especialmente al Ing. Alan Cuenca que ha sido de gran ayuda en la creación de nuevos profesionales tanto en ámbito profesional y personal.

**Angelo Armas**

# **DEDICATORIA**

Este proyecto está dedicado especialmente a todas las personas que me brindaron su apoyo y conocimiento para seguir adelante y jamás rendirme durante esta etapa de estudiante y en la vida en general.

**Emerson Viteri**

# **AGRADECIMIENTO**

Agradezco a mis padres por todo el amor, apoyo y comprensión que me han proporcionado a lo largo de mi vida, y en especial durante mis estudios universitarios.

A mis hermanos por motivarme a ser mejor y siempre sacarme una sonrisa en los momentos que más lo he necesitado.

A mi enamorada por brindarme todo su cariño, apoyarme y alentarme a lo largo de estos años que ha durado la carrera.

Al Ing. Alan Cuenca por sus observaciones y recomendaciones durante el desarrollo de este proyecto.

A los docentes de la Escuela de Formación de Tecnólogos por su dedicación al compartir sus conocimientos y experiencias, tanto del ámbito profesional como del personal.

**Emerson Viteri**

# ÍNDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Objetivo general.....	2
1.2	Objetivos específicos.....	2
1.3	Fundamentos.....	2
	Calentador solar.....	2
	Captador solar de tubos al vacío.....	3
	Efecto termosifón.....	3
	<i>NodeMCU</i> .....	4
	<i>Firebase</i> .....	5
	<i>RealTime Database</i> .....	5
	<i>MIT App Inventor</i> .....	5
2	Metodología.....	7
2.1	Descripción de la metodología usada.....	7
3	Resultados y Discusión.....	9
3.1	Requerimientos.....	9
	Dimensiones del módulo termosolar.....	9
	Tanque interno.....	12
	Tanque Externo.....	13
	Sensores.....	15
	<i>LM35</i> .....	15
	<i>ML8511</i> .....	15
3.2	Construcción del módulo termosolar.....	16
3.3	Algoritmos de medición y transmisión de datos.....	18
	Algoritmo de <i>Arduino Uno</i> .....	19
	Algoritmo de <i>NodeMCU</i> .....	21
	Sistema de telemetría.....	25
3.4	Comunicación del sistema.....	28



Desarrollo de la aplicación en <i>App Inventor</i> .....	28
3.5 Pruebas y Análisis de Resultados .....	33
Calentamiento de agua .....	33
Medición de datos de temperatura y radiación solar.....	34
Comunicación entre los microcontroladores y los sensores .....	35
Transmisión, monitoreo y registro de datos .....	37
3.6 Manual de Uso y Mantenimiento .....	38
4 Conclusiones y Recomendaciones.....	39
4.1 Conclusiones.....	39
4.2 Recomendaciones .....	40
5 Referencias Bibliográficas .....	41
ANEXOS .....	43
ANEXO 1: CERTIFICADO DE FUNCIONAMIENTO .....	i
ANEXO 2: CÓDIGOS DE PROGRAMACIÓN .....	iii
ANEXO 3: PLANOS Y ESQUEMAS .....	x

## ÍNDICE DE FIGURAS

<b>Figura 1.1</b> Componentes de un calentador solar.....	2
<b>Figura 1.2</b> Partes de un tubo al vacío .....	3
<b>Figura 1.3</b> Proceso del efecto termosifón .....	4
<b>Figura 1.4</b> <i>NodeMCU</i> .....	4
<b>Figura 1.5</b> Funcionamiento de <i>RealTime Database</i> .....	5
<b>Figura 1.6</b> Diagrama de funcionamiento de <i>MIT App Inventor</i> .....	6
<b>Figura 3.1</b> Lugar de instalación del sistema.....	9
<b>Figura 3.2</b> Irradiación directa de la ubicación (2007 – 2016) .....	10
<b>Figura 3.3</b> Irradiación directa normal e irradiación a 12° .....	11
<b>Figura 3.4</b> Estructura del termotanque.....	13
<b>Figura 3.5</b> Triángulo formado por la estructura.....	14
<b>Figura 3.6</b> Diseño 3D del módulo calentador solar.....	15
<b>Figura 3.7</b> LM35.....	15
<b>Figura 3.8</b> <i>ML8511</i> .....	16
<b>Figura 3.9</b> Tanque interno de Acero A36 .....	16
<b>Figura 3.10</b> Tanque externo de Acero galvanizado.....	17
<b>Figura 3.11</b> Termotanque y soporte .....	17
<b>Figura 3.12</b> Empaque de silicón.....	18
<b>Figura 3.13</b> Módulo termosolar de agua caliente terminado .....	18
<b>Figura 3.14</b> Librería y variables del programa ( <i>Arduino</i> ) .....	19
<b>Figura 3.15</b> Función <i>void setup</i> ( <i>Arduino</i> ) .....	19
<b>Figura 3.16</b> Acondicionamiento del sensor UV ( <i>Arduino</i> ).....	20
<b>Figura 3.17</b> Transmisión de datos al monitor serial y módulo <i>WiFi</i> ( <i>Arduino</i> ).....	20
<b>Figura 3.18</b> Diagrama de flujo del programa del <i>Arduino</i> Uno .....	21
<b>Figura 3.19</b> Librerías y pines para TX - RX ( <i>NodeMCU</i> ).....	22
<b>Figura 3.20</b> Credenciales <i>WiFi</i> y <i>Firestore</i> ( <i>NodeMCU</i> ) .....	22
<b>Figura 3.21</b> Variables del sensor de temperatura ( <i>NodeMCU</i> ) .....	22
<b>Figura 3.22</b> Función <i>void setup</i> ( <i>NodeMCU</i> ) .....	23
<b>Figura 3.23</b> Acondicionamiento del sensor de temperatura ( <i>NodeMCU</i> ) .....	23
<b>Figura 3.24</b> Cadena de caracteres en <i>Firestore</i> ( <i>NodeMCU</i> ).....	23
<b>Figura 3.25</b> Recepción y envío de datos ( <i>NodeMCU</i> ).....	24
<b>Figura 3.26</b> Diagrama de flujo del programa de <i>NodeMCU</i> .....	24
<b>Figura 3.27</b> Diagrama esquemático del sistema de telemetría .....	25

<b>Figura 3.28</b> Ubicación de los elementos del circuito .....	26
<b>Figura 3.29</b> Pistas de conexión .....	26
<b>Figura 3.30</b> Circuito Implementado .....	27
<b>Figura 3.31</b> Sistema de telemetría vista superior .....	27
<b>Figura 3.32</b> Conexión a la red <i>WiFi</i> .....	28
<b>Figura 3.33</b> Comunicación con la base de datos.....	28
<b>Figura 3.34</b> Pantalla de inicio de <i>SOLAR HELP</i> .....	29
<b>Figura 3.35</b> Código de bloques (Pantalla de inicio) .....	29
<b>Figura 3.36</b> Pantalla de monitoreo de <i>SOLAR HELP</i> .....	30
<b>Figura 3.37</b> Bloques iniciales (Pantalla de monitoreo) .....	30
<b>Figura 3.38</b> Bloques para presentación de datos (Pantalla de monitoreo) .....	31
<b>Figura 3.39</b> Bloques del botón de manual de usuario (Pantalla de monitoreo) .....	31
<b>Figura 3.40</b> Bloques del botón de mantenimiento (Pantalla de monitoreo) .....	31
<b>Figura 3.41</b> Conexión exitosa para el monitoreo de datos .....	32
<b>Figura 3.42</b> Diagrama de flujo de la aplicación.....	32
<b>Figura 3.43</b> Temperatura en la entrada de agua fría.....	33
<b>Figura 3.44</b> Temperatura en la salida de agua caliente .....	33
<b>Figura 3.45</b> Gráfica de temperatura en relación con el tiempo .....	35
<b>Figura 3.46</b> Gráfica de índice UV en relación con el tiempo .....	35
<b>Figura 3.47</b> Comunicación con el sensor de radiación.....	36
<b>Figura 3.48</b> Comunicación con la red <i>WiFi</i> .....	36
<b>Figura 3.49</b> Comunicación con el sensor de temperatura.....	36
<b>Figura 3.50</b> Datos en <i>Firebase</i> .....	37
<b>Figura 3.51</b> Datos en <i>SOLAR HELP</i> .....	37
<b>Figura 3.52</b> Código QR de las pruebas de funcionamiento.....	38
<b>Figura 3.53</b> Código QR del manual de uso.....	38
<b>Figura 3.54</b> Código QR del manual de mantenimiento.....	38

## ÍNDICE DE TABLAS

<b>Tabla 3.1</b> Ángulo de inclinación respecto a la latitud del lugar .....	10
<b>Tabla 3.2</b> Medición de temperatura y radiación UV .....	34

## RESUMEN

El presente proyecto de titulación referente a la implementación de un sistema de telemetría para un módulo termosolar de agua caliente, tiene como objetivo abastecer a la Escuela de Formación de Tecnólogos (ESFOT) con un módulo para agua caliente con su propio sistema de telemetría, que permita medir y visualizar datos de temperatura del agua y radiación solar en tiempo real. Así como también, conocer acerca del proceso de calentamiento de agua por efecto termosifón utilizando energía solar.

Este documento está formado por cinco secciones, tal como se detalla a continuación:

En la primera sección se describe la introducción, el objetivo general y los objetivos específicos, además de una corta explicación del funcionamiento del módulo termosolar y las distintas plataformas utilizadas para el desarrollo del sistema.

La segunda sección da a conocer la metodología aplicada en la elaboración de este proyecto, de acuerdo con cada objetivo planteado para el correcto dimensionamiento, construcción y operación del sistema.

La tercera sección explica el procedimiento y los resultados de la construcción del módulo, el desarrollo de los algoritmos de medición y transmisión de datos, la comunicación del sistema y sus respectivas pruebas de funcionamiento.

En la cuarta sección se presentan las conclusiones y recomendaciones obtenidas durante el desarrollo de este proyecto.

Finalmente, en la quinta sección se muestran las referencias bibliográficas en base a las cuales se realiza este proyecto de titulación.

**PALABRAS CLAVE:** Calentador termosolar, Telemetría, *NodeMCU*, *Firebase*, *ApplInventor*.

## **ABSTRACT**

*The present degree project with the topic: Implementation of a telemetry system for a thermosolar hot water module, aims to supply the “Escuela de Formación de Tecnólogos” (ESFOT) with a thermosolar hot water module with its own telemetry system, which allows to measure and visualize water temperature and solar radiation data in real time. As well as learn about the process of heating water by thermosyphon effect using solar energy.*

*This document consists of five sections, as detailed below:*

*The first section describes the introduction, the general objective and the specific objectives, as well as a brief explanation of the solar thermal module operation and the different platforms used for its development.*

*The second section presents the methodology applied in the elaboration of this project, according to each objective set for the correct dimensioning, construction and system operation.*

*The third section explains the procedure and the results obtained in relation to the objectives set for the module construction, development of measurement and data transmission algorithms, the system communication and its respective performance test.*

*The fourth section presents the conclusions and recommendations obtained during this project development.*

*Finally, the fifth section shows the bibliographic references on the basis of this degree project has been prepared.*

**KEYWORDS:** *Thermosolar heater, Telemetry, NodeMCU, Firebase, AppInventor.*

# 1 INTRODUCCIÓN

Uno de los mecanismos más usados para el calentamiento de agua, es la utilización de combustibles fósiles. Actualmente en Ecuador el 77% de los sistemas usa combustibles fósiles, más de la mitad de esta energía es consumida para generación de agua caliente, este costo es elevado y representa una gran inversión acumulada durante la vida útil que tiene un calefón tradicional [1].

Debido al alza constante de precios de los hidrocarburos el valor de combustibles fósiles va en aumento, siendo cada vez más evidente un costo significativo para el país. El agotamiento de reservas, dificultad de abastecimiento y dependencia energética son graves efectos del consumo de energías de origen fósil y una de las consecuencias más importantes es la severa contaminación ambiental [1].

La utilización de calentadores termosolares para producción de agua caliente representa un ahorro significativo en términos monetarios, a su vez, una medida de apoyo a un adecuado manejo medioambiental, disminución del efecto invernadero y la sobreexplotación de hidrocarburos, ya que utiliza energía limpia y sostenible [2].

Un módulo calentador solar es un tipo de dispositivo que a través del uso de energía solar permite calentar agua sin recurrir al gasto energético de energías tradicionales como gas o electricidad [3].

Actualmente las Energías Renovables y las Tecnologías de la Información y Comunicación (TIC) han alcanzado un alto desarrollo a nivel nacional, ya que han contribuido a la solución de múltiples problemas de la sociedad ecuatoriana a través de soluciones innovadoras. Por lo tanto, la combinación del módulo calentador solar y el sistema de telemetría permitirá conocer las diferentes condiciones operativas del módulo en tiempo real, y proporcionará al usuario accesibilidad, seguridad, servicio óptimo y satisfacción.

Teniendo en cuenta estas consideraciones, es muy importante que los estudiantes de la ESFOT puedan conocer el principio de funcionamiento de un módulo termosolar con un sistema de telemetría, que permita la medición y obtención de información relevante como temperatura del agua y niveles de radiación. Estos datos serán transmitidos a través de una red *WiFi* y presentados al usuario mediante una aplicación para monitoreo del sistema. En el ANEXO 1 se adjunta el certificado de funcionamiento que valida la correcta implementación de este proyecto.

## 1.1 Objetivo general

Implementar un sistema de telemetría para un módulo termosolar de agua caliente.

## 1.2 Objetivos específicos

Determinar los principales requerimientos para el sistema termosolar.

Construir la estructura del módulo calentador solar de agua.

Desarrollar el algoritmo de medición y transmisión de datos.

Establecer la comunicación del sistema a través de una red *WiFi* para monitoreo de datos.

Realizar pruebas de funcionamiento.

Realizar un manual de uso y mantenimiento.

## 1.3 Fundamentos

### Calentador solar

Es un dispositivo que convierte la energía solar en calor usado para calentar algún fluido como: agua, alcohol, aceite, etc. La energía solar se transforma en calor sobre una superficie oscura que calienta al fluido con el que está en contacto. Es comúnmente utilizado para calentar agua en regaderas, lavadoras, lavaplatos, etc [4]. Es decir, puede ser usado en sectores residenciales, comerciales e industriales sin ningún inconveniente. En la Figura 1.1 se puede apreciar un módulo calentador solar.



**Figura 1.1** Componentes de un calentador solar [5]



Un calentador solar de agua está compuesto por las siguientes partes principales:

- **Termotanque:** Es un depósito aislado térmicamente en el que se almacena el agua [4].
- **Aislamiento térmico:** Es un compuesto similar a una esponja rígida que mantiene el agua caliente dentro del termotanque. Regularmente se utiliza espuma de poliuretano gracias a su gran capacidad de aislamiento [4].
- **Colector solar:** Es la superficie donde se calienta el agua, existen de tipo plano o de tubos al vacío [4].
- **Estructura de soporte:** Es la base principal que se encarga de dar la inclinación necesaria al colector solar para obtener calentamiento por termosifón [4].

### Captador solar de tubos al vacío

El captador solar de tubos al vacío consta de un colector lineal (tubo de calor) que recoge la energía solar y se aloja en el interior de tubos de vidrio de dos capas entre las cuales está hecho el vacío (Figura 1.2), esto permite reducir las fugas térmicas en comparación a otros tipos de colectores o paneles solares [6].

Este tipo de colector proporciona un mejor rendimiento frente a los de tipo plano. Su tecnología de tubo de calor lo vuelve más eficiente e indicado para captar la energía solar en tiempos con menos radiación, además que mantiene mejor el calor respecto a otras soluciones [6].

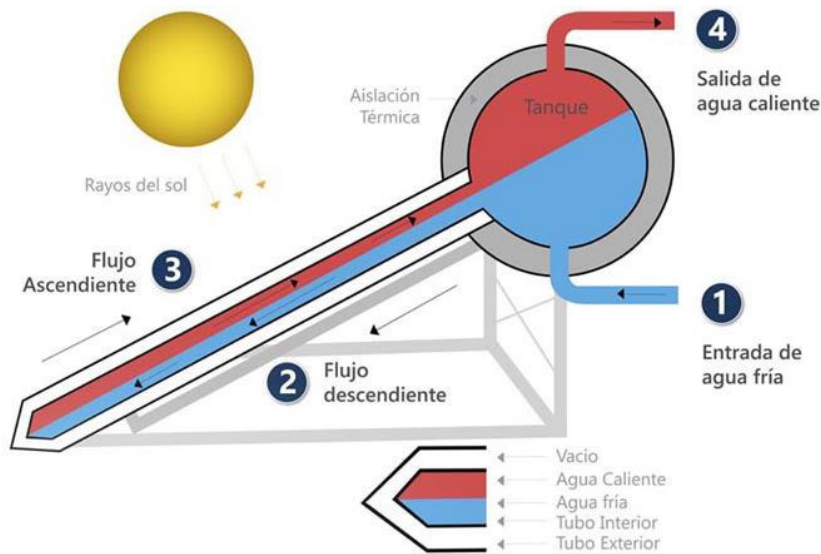


**Figura 1.2** Partes de un tubo al vacío [6]

### Efecto termosifón

El colector absorbe la energía solar, razón por la cual el agua dentro de los tubos comienza a calentarse. Cuando el agua se calienta pierde densidad, es decir, el agua caliente sube a la parte más alta del sistema, en este caso el termotanque [6]. Al mismo tiempo el agua fría se ubica en el colector solar que es la parte más baja del sistema y así se repetirá el ciclo. Luego de varias horas el agua dentro de todo el sistema estará

muy caliente y disponible para su uso, este proceso se encuentra detallado en la Figura 1.3.



**Figura 1.3** Proceso del efecto termosifón [6]

### **NodeMCU**

Es una placa de desarrollo similar a *Arduino* orientada a soluciones de *Internet of Things (IoT)*, en base al microcontrolador *ESP8266* que permite conectarse a través de la red *WiFi*. Esta plataforma (Figura 1.4) permite el desarrollo de aplicaciones en diferentes lenguajes como: *Arduino*, *Python*, *C/C++* [7]. A continuación, se detallan algunas características técnicas importantes:

- Voltaje de alimentación: 5 (V<sub>DC</sub>)
- Voltaje de Entradas/Salidas: 3,3 (V<sub>DC</sub>)
- Antena integrada
- Cuenta con dieciséis pines digitales y uno analógico
- Memoria *flash*: 4 (MB)
- Funcionalidades *WiFi*



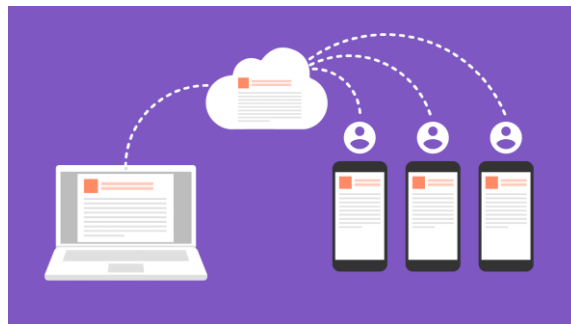
**Figura 1.4** NodeMCU [7]

## ***Firestore***

Es una plataforma en la nube para el desarrollo de aplicaciones *web* y móviles, es compatible con *iOS*, *Android* y *web*, lo que agiliza el trabajo manteniendo la calidad en el desarrollo. Comenzó siendo una base de datos en tiempo real, aunque actualmente cuenta con varias herramientas agrupadas simplificando la gestión de tareas a una misma plataforma [8].

## ***RealTime Database***

Es una de las herramientas principales en el desarrollo de este proyecto y de las más destacadas de *Firestore*, es una base de datos que almacena y dispone datos o información manteniéndolos actualizados en tiempo real [8]. Aunque algún usuario no se encuentre conectado, los datos seguirán disponibles para los demás usuarios y se sincronizan inmediatamente. En la Figura 1.5 se observa el esquema de funcionamiento de *RealTime Database*.



**Figura 1.5** Funcionamiento de *RealTime Database* [9]

## ***MIT App Inventor***

Es un entorno de desarrollo de *software* para elaborar aplicaciones destinadas al sistema operativo de *Android*. Esta plataforma tiene como objetivo facilitar la creación de aplicaciones para personas con o sin conocimientos, ya que está basada en la programación mediante códigos de bloque. *MIT App Inventor* cuenta con una interfaz gráfica e intuitiva además de un simulador de aplicaciones directa, mediante *WiFi* o a través de un emulador. Otra gran característica de esta plataforma es que permite al usuario descargar la aplicación en *.apk* o alojarla en *Google Play* [10]. En la Figura 1.6 se muestra Diagrama de funcionamiento de *MIT App Inventor*.



**Figura 1.6** Diagrama de funcionamiento de *MIT App Inventor* [11]

## 2 METODOLOGÍA

### 2.1 Descripción de la metodología usada

Se implementó un módulo calentador solar de agua, el cual tiene un sistema de calentamiento por tubos de vacío, además, cuenta con sensores para la obtención de datos de radiación solar y temperatura del agua, los cuales son transmitidos al usuario a través de una red *WiFi*.

Se determinaron los principales elementos para la construcción del módulo calentador termosolar tomando en cuenta el principio de funcionamiento de los sistemas solares para ACS (Agua Caliente Sanitaria). Además, en base a la utilización de herramientas *web* como el Sistema de Información Geográfica Fotovoltaica (*PVGIS*) se determinaron los niveles de radiación solar directa de la ESFOT para realizar el respectivo dimensionamiento del sistema.

Con los datos de radiación se procedió a seleccionar los respectivos componentes del calentador solar como captador termosolar (panel) y tanque reservorio para el agua caliente. Finalmente, se usaron los sensores para obtener los respectivos datos de temperatura y radiación, los cuales fueron transmitidos por la red *WiFi*.

Con base en los requerimientos establecidos y los elementos del calentador solar, se desarrollaron planos esquemáticos del módulo utilizando el *software AutoCAD*. Posteriormente se realizó la adquisición de los elementos y se procedió a la construcción del módulo, que consta de un tanque reservorio y un panel solar de tubos al vacío que están conectados entre sí, permitiendo la circulación de agua para su calentamiento por efecto de la energía solar.

Una vez construido el módulo calentador solar de agua, se procedió a desarrollar la debida programación para la obtención y transmisión de datos. Para el procesamiento de datos, se utilizó *Arduino Uno*, el cual recibe información procedente del sensor de radiación solar (*UV ML8511*), y *NodeMCU ESP8266* para recibir información del sensor de temperatura (*LM35*). Para la transmisión de datos, se utilizó el módulo *NodeMCU* que permite la conexión del sistema a la correspondiente red *WiFi*, el algoritmo permite el envío de datos obtenidos del calentador solar hacia una base de datos en tiempo real (*Firebase*).

Se estableció la conexión entre el sistema y la red *WiFi* correspondiente, de esta manera los usuarios pueden visualizar en tiempo real tanto temperatura como radiación solar en

dispositivos que estén conectados a esta red o a Internet, sean computadoras, tabletas o teléfonos celulares. El usuario debe ingresar en una aplicación donde se pueden observar los datos del sistema.

A continuación, se procedió a la comprobación del funcionamiento de todo el sistema, para ello se realizaron pruebas de calentamiento de agua, medición de datos de temperatura y radiación solar, comunicación entre el microcontrolador y los sensores y transmisión, monitoreo y registro de datos.

Luego de verificar el funcionamiento del sistema, se realizaron dos videos en los cuales se incluyen el manual de usuario y mantenimiento respectivamente. Dichos manuales incluyen aspectos importantes del módulo como: utilización, conexión a la red *WiFi*, visualización de datos y recomendaciones para un mantenimiento adecuado.

### 3 RESULTADOS Y DISCUSIÓN

Se implementó un módulo termosolar que calienta el agua por efecto termosifón utilizando la radiación solar absorbida por un captador de tubos al vacío, consecuentemente el agua caliente asciende a la parte superior del sistema correspondiente al termotanque, el cual está aislado térmicamente permitiendo conservar el agua caliente. A su vez, el sistema cuenta con sensores de radiación y temperatura, cuya información es almacenada en una base de datos (*Firestore*) y transmitida por *WiFi* a una interfaz desarrollada con *MIT App Inventor*. Ingresando a la aplicación es posible monitorear los datos registrados por los sensores.

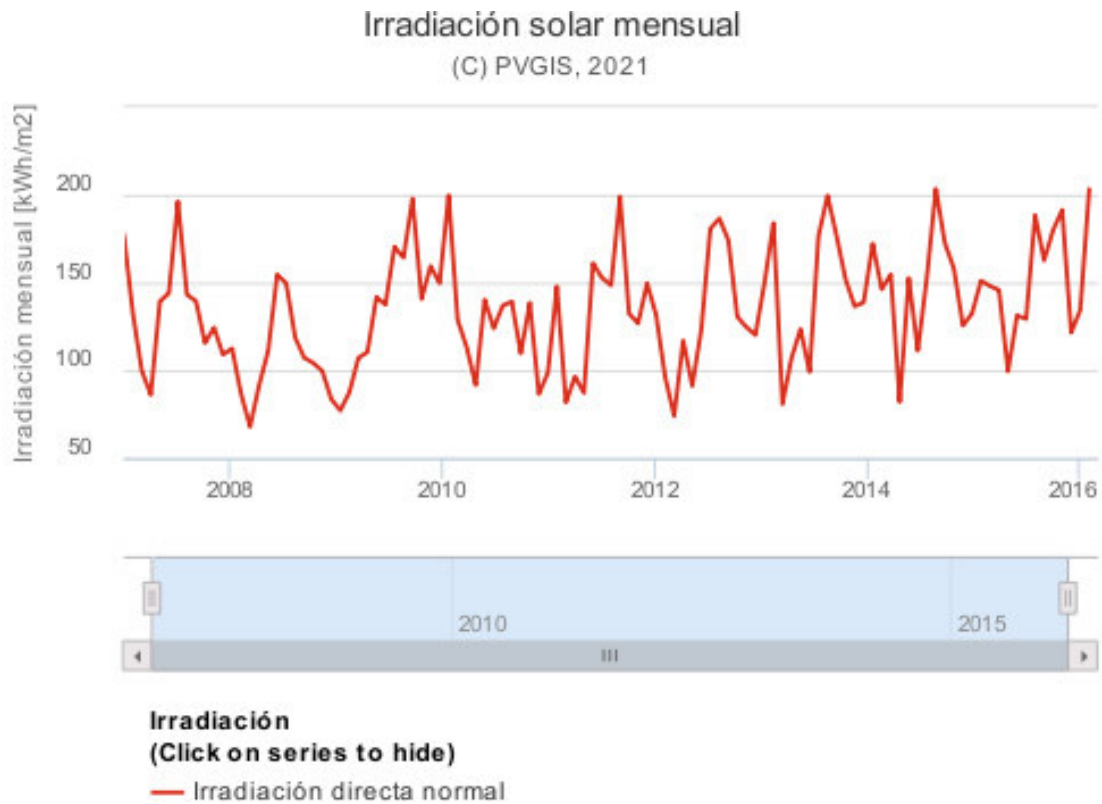
#### 3.1 Requerimientos

##### Dimensiones del módulo termosolar

Para el correcto dimensionamiento del sistema termosolar de agua caliente se determinó el lugar donde será ubicado el módulo y los niveles de radiación directa de la ESFOT empleando la herramienta *PVGIS*, tal como se muestra en la Figura 3.1 y la Figura 3.2 respectivamente.



**Figura 3.1** Lugar de instalación del sistema



**Figura 3.2** Irradiación directa de la ubicación (2007 – 2016)

Si bien los datos obtenidos con la herramienta *PVGIS* abarcan solo hasta inicios del año 2016, se tomaron en cuenta las mediciones desde el año 2007, lo cual muestra la tendencia general de radiación en dicho punto de instalación del módulo termosolar. A partir de estos datos se determinó el ángulo de inclinación del captador solar, en la Tabla 3.1 se muestra el ángulo de inclinación “óptimo” al cual se deben colocar los sistemas termosolares con respecto a la latitud del lugar de instalación.

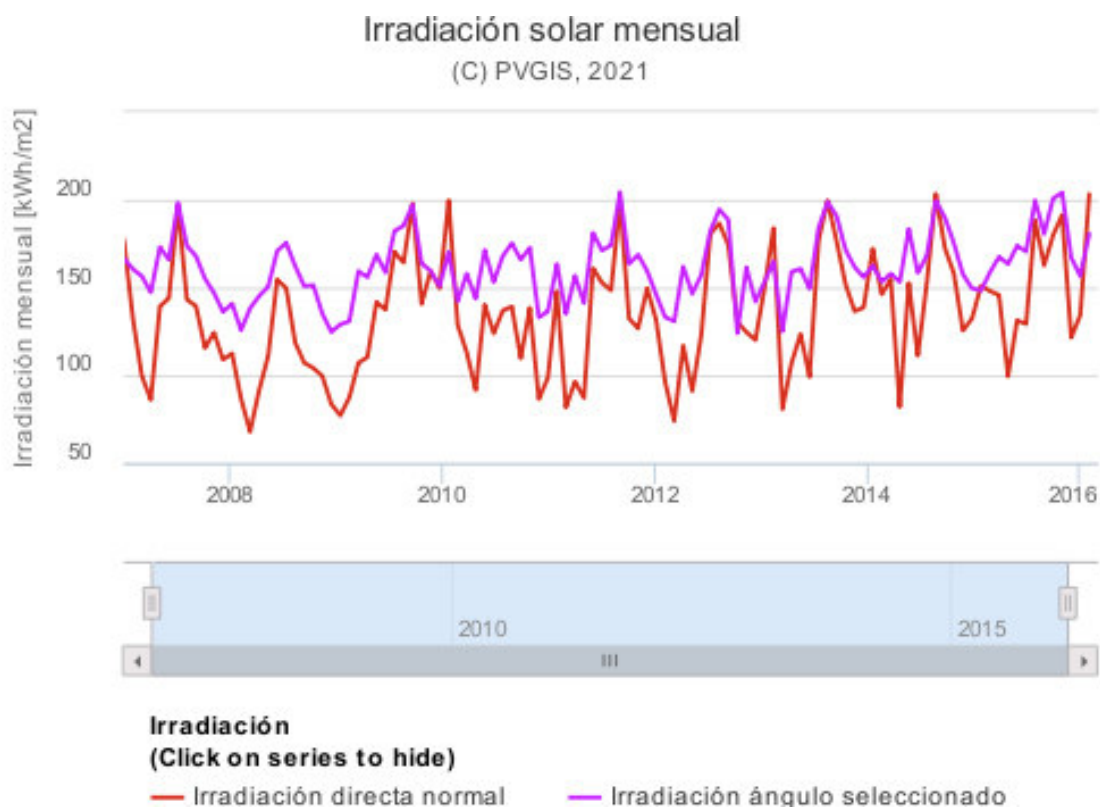
**Tabla 3.1** Ángulo de inclinación respecto a la latitud del lugar [12]

Latitud del lugar (°)	Ángulo de inclinación (°)
0° a 15°	15°
15° a 25°	La misma latitud
25° a 30°	Latitud más 5°
30° a 35°	Latitud más 10°
35° a 40°	Latitud más 15°
40° o más	Latitud más 20°

Para la construcción del captador solar se debería tomar como referencia un ángulo de inclinación de 15°, sin embargo, haciendo uso de *PVGIS* se determinó que el ángulo



óptimo para aprovechar al máximo la radiación solar en la ESFOT es de  $12^\circ$  tanto para los picos altos como para los bajos de la curva de irradiación. En la Figura 3.3 se muestran las gráficas de irradiación directa normal e irradiación con el ángulo de inclinación de  $12^\circ$ .



**Figura 3.3** Irradiación directa normal e irradiación a  $12^\circ$

A través de la gráfica obtenida se comprueba que con un ángulo de  $12^\circ$  de inclinación, se aprovecha de manera más eficiente la radiación solar, tanto que, a lo largo de casi toda la gráfica, la curva que representa la irradiación captada con el ángulo de  $12^\circ$  se mantiene por encima de la de irradiación directa.

Por otro lado, para determinar las dimensiones del termostanque es importante tener en cuenta que al tratarse de un módulo didáctico se procedió a utilizar cinco tubos al vacío de 1,80 (m) de largo, ya que corresponden a la única medida disponible de los tubos en el mercado. El termostanque está formado por dos depósitos: uno interno y otro externo, aislados por espuma de poliuretano para sellar térmicamente el tanque interno y conservar el agua caliente incluso durante la noche.

A continuación, se presentan los cálculos correspondientes al dimensionamiento del material utilizado para cada una de las láminas que conformarán los dos tanques.

## Tanque interno

Para determinar las dimensiones del tanque interno, se utilizó la Ecuación 3.1 para el cálculo del perímetro del mismo.

$$P = 2 \cdot \pi \cdot r$$

### Ecuación 3.1 Perímetro de un círculo

Donde:

$$\pi : 3,14159$$

$$r : 16,25 \text{ (cm)}$$

$$P : \text{(cm)}$$

Usando la Ecuación 3.1 se obtiene:

$$P = 102,10 \text{ (cm)}$$

En base al valor calculado, se estableció la utilización de una lámina de acero A36 de 102,10 x 70 (cm) para el tanque interno, el valor de 102,10 (cm) corresponde a la lámina que se enrolla para formar el cilindro. El acero A36 es una aleación de acero al carbono utilizado en la fabricación de estructuras soldadas y atornilladas para el sector de la construcción y la ingeniería.

A continuación, se presenta el cálculo del volumen de este tanque en función del diámetro de la circunferencia del cilindro y su altura en decímetros (dm). Para esto se utiliza la Ecuación 3.2 y la Ecuación 3.3 para realizar las respectivas conversiones de unidades.

$$32,5 \text{ (cm)} \cdot \frac{1 \text{ (dm)}}{10 \text{ (cm)}} = 3,25 \text{ (dm)}$$

### Ecuación 3.2 Conversión de (cm) a (dm) del diámetro

$$69 \text{ (cm)} \cdot \frac{1 \text{ (dm)}}{10 \text{ (cm)}} = 6,9 \text{ (dm)}$$

### Ecuación 3.3 Conversión de (cm) a (dm) de la altura

Para determinar el volumen del tanque interno, se utilizó la Ecuación 3.4:

$$V = \frac{\pi \cdot d^2 \cdot h}{4}$$

### Ecuación 3.4 Volumen de un cilindro

Donde:

$\pi$  : 3,14159

$d$  : 3,25 (dm)

$h$  : 6,9 (dm)

$V$  : ( $dm^3$ ) ó (litros)

Usando la Ecuación 3.4 se obtiene:

$$V = 57,24 (dm^3) \approx 57,3 \text{ (litros)}$$

### Tanque Externo

Para determinar las dimensiones del tanque externo, se utilizó la Ecuación 3.1 para el cálculo del perímetro del mismo.

Siendo:

$\pi$  : 3,14159

$r$  : 18,25 (cm) longitud

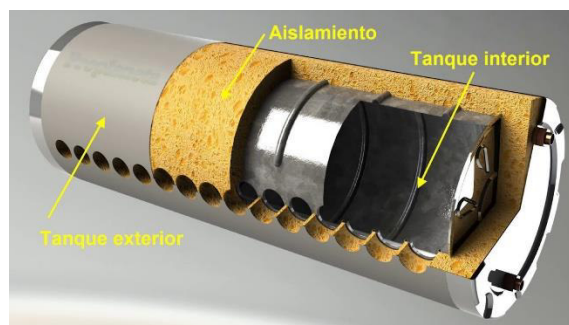
$P$  : (cm) longitud

Usando la Ecuación 3.1, se obtiene:

$$P = 114,67 \text{ (cm)}$$

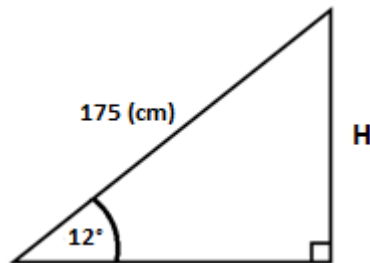
En base al valor calculado empleando, se estableció la utilización de una chaqueta de acero galvanizado de 114,67 x 75 (cm) para la construcción del tanque externo. Siendo similar al tanque interior, el valor de 114,67 (cm) corresponde a la lámina que se enrolla para formar el cilindro.

Para este cilindro no se realiza el cálculo del volumen debido a que solamente el tanque interno es el que almacena agua, la función del tanque externo es sellar el sistema con el aislante térmico de poliuretano en su interior, tal como se muestra en la Figura 3.4.



**Figura 3.4** Estructura del termostanque [13]

Para dimensionar el soporte tanto para el termotanque como para el captador solar de tubos al vacío, se tuvo en cuenta el ángulo de inclinación de  $12^\circ$  obtenido con la ayuda de *PVGIS*. En base a este ángulo se estableció el soporte de los tubos de vidrio con una altura de 31 (cm) desde el suelo, y para calcular la altura a la que se construirá el soporte del termotanque se formó un triángulo rectángulo con el largo de los tubos al vacío como hipotenusa (175 (cm)) y el ángulo de  $12^\circ$ , tal como se muestra en la Figura 3.5.



**Figura 3.5** Triángulo formado por la estructura

Empleando los datos de la Figura 3.5, y utilizando la función trigonométrica seno, se despeja la ecuación de modo que se obtenga el valor en centímetros de la altura a la que se tiene que construir el soporte del termotanque. A continuación, se muestran los cálculos correspondientes.

$$H = \text{Hipotenusa} \cdot \text{Sen } \theta$$

**Ecuación 3.5** Altura en función del seno

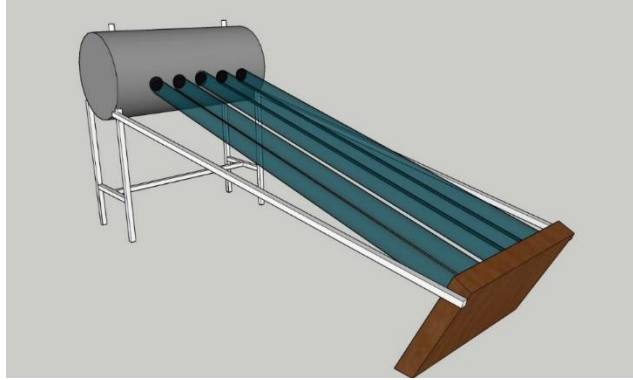
Donde:

- $\theta$  :  $12^\circ$
- Hipotenusa : 175 (cm)
- H : (cm)

Usando la Ecuación 3.5 se obtiene:

$$H = 36 \text{ (cm)}$$

Con todas las dimensiones establecidas, se realizó el diseño 3D del módulo termosolar (Figura 3.6), con ayuda de la herramienta *SketchUp*. Este diseño es adecuado ya que ha sido realizado tomando en cuenta de que se trata de un módulo termosolar didáctico que cumple los requisitos del Laboratorio de Tecnología Industrial LTI-ESFOT en cuanto a dimensiones y ubicación.



**Figura 3.6** Diseño 3D del módulo calentador solar

### **Sensores**

Tomando en cuenta que se trata de un módulo didáctico el cual permitirá comprender el principio de aprovechamiento de la energía solar para calentar agua a temperaturas no mayores a 65°C, se utilizará un sensor de temperatura *LM35* y otro de radiación UV *ML8511* encargados de realizar mediciones periódicas de ambas variables físicas para el posterior monitoreo del sistema. A continuación, se detallan las principales características de cada sensor.

#### ***LM35***

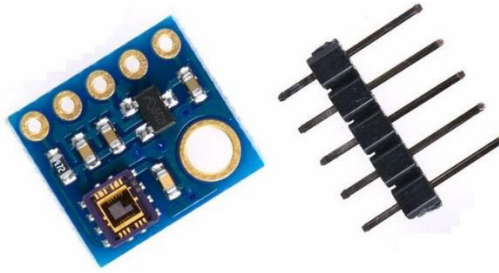
Es un dispositivo electrónico sensor que mide temperatura, su salida es analógica y tiene un rango desde -55°C a 150°C. Su resolución es de 10 (mV) por cada grado centígrado (°C), consta de 3 pines (GND,  $V_{CC}$ ,  $V_{out}$ ) y está calibrado para medir directamente en °C [14]. En la Figura 3.7 se muestra el sensor LM35.



**Figura 3.7** LM35 [14]

#### ***ML8511***

Es un sensor de luz ultravioleta (UV) que entrega una señal analógica dependiendo de la intensidad de radiación UV que percibe. Este sensor detecta luz con una longitud de onda entre 280 – 390 (nm) [15]. En la Figura 3.8 se muestra el sensor UV *ML8511*.



**Figura 3.8** *ML8511* [15]

Además de estos sensores, se necesita de las placas electrónicas *Arduino UNO* y *NodeMCU*, que son las encargadas de recibir la información transmitida por los sensores y alojarlos en la base de datos para la posterior visualización en la aplicación. La información relevante sobre la plataforma *NodeMCU* se encuentra en la sección de Fundamentos.

### **3.2 Construcción del módulo termosolar**

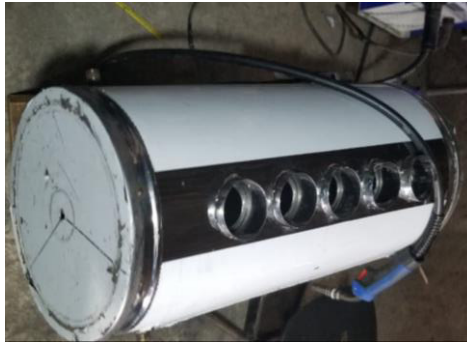
Una vez realizados los cálculos para el dimensionamiento del sistema, se diseñaron los planos esquemáticos del módulo utilizando *AutoCAD*. Se diseñaron los planos de los dos depósitos que forman el termotanque, la base de los tubos de vidrio y la base del termotanque. Para revisar los planos de cada componente dirigirse al ANEXO 3.

Previo a la construcción de los componentes del módulo termosolar de agua caliente se acudió a una empresa metálica que contaba con los materiales y experiencia previa en la construcción de termotanques para módulos de agua caliente. Se solicitó la construcción de cada parte del módulo de acuerdo con el diseño y medidas planteadas en los planos. En la Figura 3.9 se puede apreciar el tanque interno construido en Acero A36 con los agujeros correspondientes al punto de conexión entre este almacenador de agua y los tubos de vidrio.



**Figura 3.9** Tanque interno de Acero A36

En la Figura 3.10 se muestra el tanque externo construido en acero galvanizado correspondiente a la chaqueta para sellamiento térmico, ambos tanques cuentan con las respectivas entradas para los tubos al vacío, entrada de agua fría, salida de agua caliente y desfogue.



**Figura 3.10** Tanque externo de Acero galvanizado

Para el ensamblaje del termotanque se rellena el espacio entre ambos tanques con espuma de poliuretano con el fin de obtener la máxima retención de calor posible en el sistema, en la Figura 3.11 se presenta al termotanque y su base terminados y soldados, listos para unirse con el captador solar de tubos al vacío.



**Figura 3.11** Termotanque y soporte

A continuación, fue necesario adquirir empaques de silicón en forma de anillo hechos a la medida exacta del perímetro de los tubos al vacío, debido a que estos empaques tienen como función aislar la conexión entre el termotanque y los tubos, es decir, evitar posibles fugas de agua en el sistema. En la Figura 3.12 se muestra el diseño de los anillos.





**Figura 3.12** Empaque de silicón

Finalmente se ensambló el módulo por completo, los tubos al vacío se unieron correctamente al termotanque y por acción de los empaques se obtuvo un sistema sin fugas listo para unirse con la base del colector solar elaborado en tabla trípex. Después de colocar el colector en su base, fue necesario atornillar el sistema con barras laterales de aluminio para brindarle más estabilidad al módulo y evitar que accidentalmente se separen los componentes o se rompan los tubos. Adicionalmente se le adaptó un pequeño reservorio en la entrada de agua fría encargado de mantener el volumen de agua constante dentro del termotanque (sistema de autollenado). La Figura 3.13 muestra el módulo construido.



**Figura 3.13** Módulo termosolar de agua caliente terminado

### **3.3 Algoritmos de medición y transmisión de datos**

Se desarrollaron dos algoritmos en el entorno de trabajo *Arduino IDE*, el primero de ellos para *Arduino Uno* y el otro para el módulo *WiFi NodeMCU*. A cada microcontrolador se le asignó la tarea de recibir la información de un sensor, así como también de transmitirla, ya sea de *Arduino Uno* al módulo *WiFi* o hacia una base de datos alojada en la nube.



## Algoritmo de *Arduino Uno*

Este código de programación está desarrollado con el fin de obtener la información que detecta el sensor de radiación UV, acondicionar la señal y presentarla por niveles de radiación; además de mostrar las mediciones en el monitor serial y transmitir las hacia el módulo *WiFi*. En la Figura 3.14 se observa el uso de la librería *SoftwareSerial* que permite la comunicación serial en pines digitales, también se definen los pines de transmisión y recepción de datos, y la declaración de variables necesarias para el caso.

```
#include <SoftwareSerial.h>           //Librería para comunicación serie en pines digitales

#define rxPin 2                       //definir pin 2 como recepción serie
#define txPin 5                       //definir pin 5 como transmisión serie

SoftwareSerial Trans(rxPin, txPin); //Pines para Rx y Tx, y denominar Trans como monitor serial

//Declaración de variables del proceso
int uv=A3;
int uvIndice;
```

**Figura 3.14** Librería y variables del programa (*Arduino*)

La Figura 3.15 muestra la sección *void setup* donde se ha configurado el pin 5 como salida ya que dicho pin envía los datos del sensor UV. También se inicializa y establece la velocidad de transmisión para el monitor serial de *Arduino* y transmisión hacia el módulo *WiFi*.

```
void setup()
{
  pinMode(txPin, OUTPUT);           //Definir pin de transmisión como salida
  Serial.begin(9600);              //Iniciar monitor serial de Arduino, velocidad 9600
  Trans.begin(9600);               //Velocidad de envío para la transmisión al NodeMCU, e iniciar monitor serial
  pinMode(uv, INPUT);              //Pin de sensor ML8511 radiación como entrada
}
```

**Figura 3.15** Función *void setup* (*Arduino*)

Seguidamente se muestra el código desarrollado en la función *void loop* en la Figura 3.16, aquí primero se agregó una variable de tipo entero en la que se almacena la señal analógica del sensor UV para posteriormente ser acondicionada en función de las unidades de procesamiento (digitales) que se recibe de este. Luego se agregaron varios condicionales para que, según el voltaje de salida del sensor, se interprete la información como índice de radiación UV.

```

void loop()
{
    int uvLevel = analogRead(uv);           //lectura de sensor de radiación
    float outputVoltage = 3.3 * uvLevel/1024;//cálculo de la salida de voltaje del sensor uv

    if(outputVoltage<=0.7){                //condición de la salida de voltaje,entre 0 y 0.7
        uvIndice=0;                        //declaración del nivel 0 de índice de radiación
    }else if(outputVoltage>0.7 && outputVoltage<=0.83){ //condición de la salida de voltaje,entre 0.71 y 0.83
        uvIndice=1;                        //declaración del nivel 1 de índice de radiación
    }else if(outputVoltage>0.83 && outputVoltage<=0.96){ //condición de la salida de voltaje,entre 0.84 y 0.96
        uvIndice=2;                        //declaración del nivel 2 de índice de radiación
    }else if(outputVoltage>0.96 && outputVoltage<=1.09){ //condición de la salida de voltaje,entre 0.97 y 1.09
        uvIndice=3;                        //declaración del nivel 3 de índice de radiación
    }else if(outputVoltage>1.09 && outputVoltage<=1.22){ //condición de la salida de voltaje,entre 1.10 y 1.22
        uvIndice=4;                        //declaración del nivel 4 de índice de radiación
    }else if(outputVoltage>1.22 && outputVoltage<=1.35){ //condición de la salida de voltaje,entre 1.23 y 1.35
        uvIndice=5;                        //declaración del nivel 5 de índice de radiación
    }else if(outputVoltage>1.35 && outputVoltage<=1.48){ //condición de la salida de voltaje,entre 1.36 y 1.48
        uvIndice=6;                        //declaración del nivel 6 de índice de radiación
    }else if(outputVoltage>1.48 && outputVoltage<=1.61){ //condición de la salida de voltaje,entre 1.49 y 1.61
        uvIndice=7;                        //declaración del nivel 7 de índice de radiación
    }else if(outputVoltage>1.61 && outputVoltage<=1.74){ //condición de la salida de voltaje,entre 1.62 y 1.74
        uvIndice=8;                        //declaración del nivel 8 de índice de radiación
    }else if(outputVoltage>1.74 && outputVoltage<=1.87){ //condición de la salida de voltaje,entre 1.75 y 1.87
        uvIndice=9;                        //declaración del nivel 9 de índice de radiación
    }else if(outputVoltage>1.87 && outputVoltage<=2){ //condición de la salida de voltaje,entre 1.88 y 2
        uvIndice=10;                       //declaración del nivel 10 de índice de radiación
    }else if(outputVoltage>2 && outputVoltage<=2.13){ //condición de la salida de voltaje,entre 2.01 y 2.13
        uvIndice=11;                       //declaración del nivel 11 de índice de radiación
    }else if(outputVoltage>2.13 && outputVoltage<=2.26){ //condición de la salida de voltaje,entre 2.14 y 2.26
        uvIndice=12;                       //declaración del nivel 12 de índice de radiación
    }else if(outputVoltage>2.26 && outputVoltage<=2.39){ //condición de la salida de voltaje,entre 2.27 y 2.39
        uvIndice=13;                       //declaración del nivel 13 de índice de radiación
    }else if(outputVoltage>2.39 && outputVoltage<=2.52){ //condición de la salida de voltaje,entre 2.40 y 2.52
        uvIndice=14;                       //declaración del nivel 14 de índice de radiación
    }else if(outputVoltage>=2.53){ //condición de la salida de voltaje,entre 2.53 y 3.3
        uvIndice=15;                       //declaración del nivel 15 de índice de radiación
    }
}

```

**Figura 3.16** Acondicionamiento del sensor UV (*Arduino*)

Por último, se le asignó la tarea de imprimir en el monitor serial de *Arduino* la lectura del sensor UV, su salida de voltaje, y el respectivo índice de radiación. En la Figura 3.17 también se puede observar la línea de programación que indica al microcontrolador enviar el valor de índice UV a través del pin 5.

```

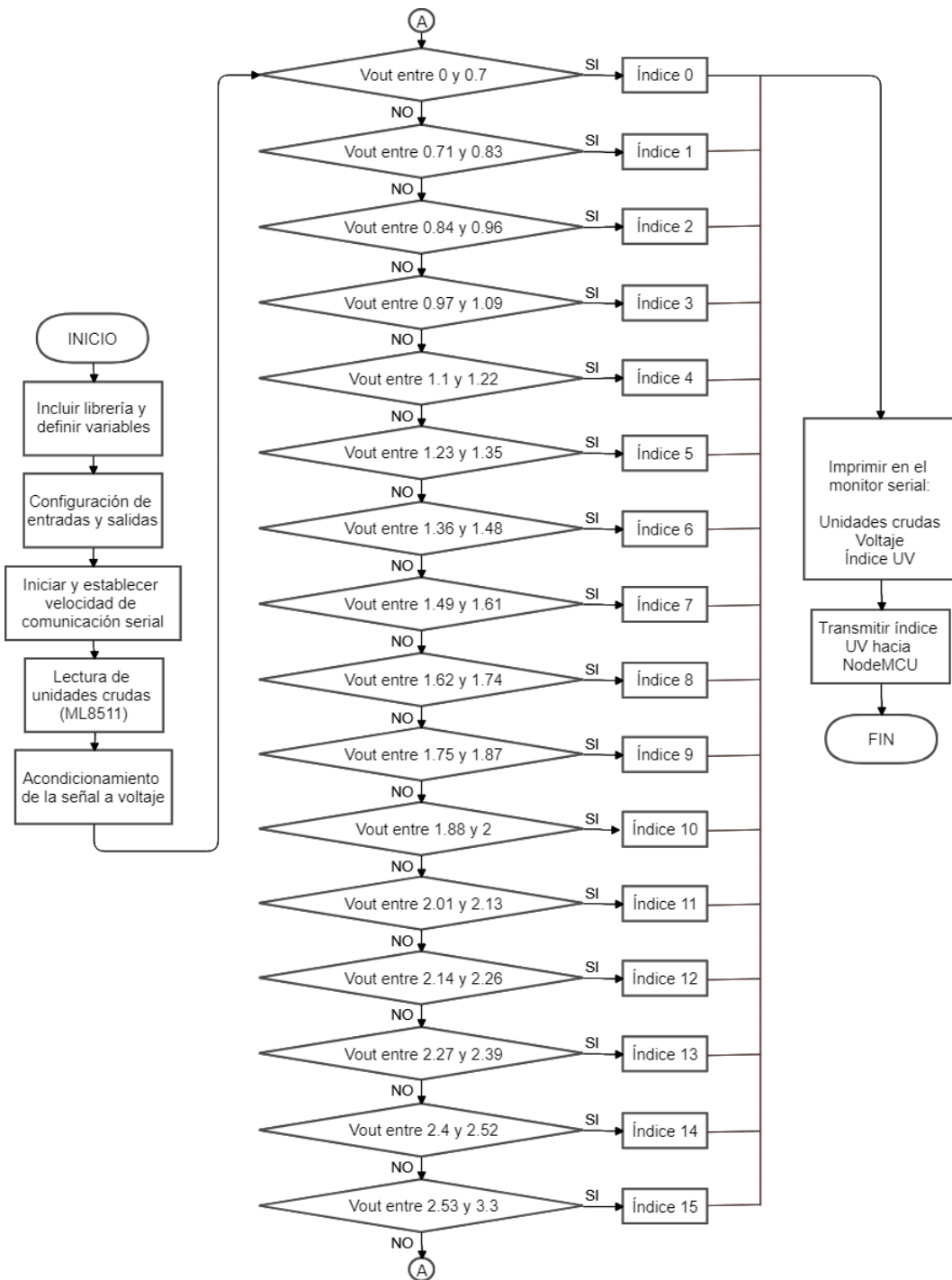
//Mostrar el valor en el monitor serie de arduino.
Serial.println(uvLevel);           //Imprime en monitor serial la lectura del sensor uv
Serial.println(outputVoltage);    //Imprime en monitor serial la salida del voltaje
Serial.println("Indice UV:"+String(uvIndice)); //Imprime en monitor serial el Indice UV
delay(1000);                       //1 seg de espera

Trans.println(uvIndice);           //Para enviar la variable de Indice UV por el pin 5 del arduino UNO
}

```

**Figura 3.17** Transmisión de datos al monitor serial y módulo *WiFi* (*Arduino*)

Para comprender de forma adecuada todo el código de programación desarrollado se muestra en la Figura 3.18 el respectivo diagrama de flujo.



**Figura 3.18** Diagrama de flujo del programa del *Arduino Uno*

### Algoritmo de *NodeMCU*

Este código de programación cumple con las tareas detalladas a continuación:

- Obtiene la información que detecta el sensor de temperatura, acondiciona la señal, promedia cinco valores recibidos y la presenta en grados centígrados.

- Se conecta a la red *WiFi* mediante código de programación utilizando las credenciales de esta red.
- Establece comunicación con la base de datos en *Firebase*, a través del enlace del proyecto y la llave de autorización.
- Recibe los datos transmitidos por *Arduino* Uno (índice UV), los presenta a través del monitor serial y envía ambas mediciones hacia la base de datos.

En la Figura 3.19 se muestran las librerías necesarias para establecer la comunicación con la red *WiFi* y el proyecto en *Firebase*, además de la comunicación serial en pines digitales y la asignación de sus pines de transmisión y recepción.

```
#include <ESP8266WiFi.h>           //Librería que permite establecer comunicación inalámbrica
#include <FirebaseESP8266.h>       //Librería que permite conexión de la base de datos en firebase a ESP8266
#include <SoftwareSerial.h>        //Librería para comunicación serie en pines digitales
SoftwareSerial mySerial(3, 15);    //Pines para Rx y Tx, y denominar mySerial como monitor serial
```

**Figura 3.19** Librerías y pines para TX - RX (*NodeMCU*)

A continuación, se asignan variables de carácter constante que contienen las credenciales tanto de la red *WiFi* a la que se conectará el sistema, como del proyecto en *Firebase* que aloja las mediciones realizadas por los sensores (Figura 3.20). También se encuentran definidas las variables necesarias para lectura, acondicionamiento y promedio de los datos de temperatura del agua (Figura 3.21).

```
// Credenciales wifi
const char *ssid = "Claro_REA0014342900"; // nombre de la red
const char *password = "CARMEN84616960328815"; //contraseña de la red
// Credenciales Proyecto Firebase
const char *FIREBASE_HOST = "calentadorsolaragua-default-rtdb.firebaseio.com"; //Nombre del Host Firebase
const char *FIREBASE_AUTH = "ZjHeUtrRH6AcIQ10XCmQMokEpUptm8jdF921yd7iD"; //LLave de Autorización
// Objeto de datos de Firebase en el ámbito global
FirebaseData firebaseData; //Nombre de la Base de Datos
```

**Figura 3.20** Credenciales *WiFi* y *Firebase* (*NodeMCU*)

```
float promedio; //Declaración de variable para lm35
float temp; //Declaración de variable para lm35
float sensorlm35; //Declaración de variable para lm35
```

**Figura 3.21** Variables del sensor de temperatura (*NodeMCU*)

La Figura 3.22 muestra la inicialización y velocidad establecida para la transmisión de datos al monitor serial y del módulo *NodeMCU*, después se realiza la conexión a la red *WiFi* con ayuda de sus credenciales, y se envía un mensaje de aviso cuando se ha conectado exitosamente; consecuentemente se realiza la conexión del módulo con el proyecto en *Firebase*.

```

void setup()
{
  Serial.begin(9600);           //Iniciar monitor serial de Arduino, velocidad 9600
  mySerial.begin(9600);        //Velocidad para la recepción al NodeMCU, e iniciar monitor serial

  WiFi.begin(ssid, password);  //Iniciar conexión con los datos de la red definidos arriba

  while (WiFi.status() != WL_CONNECTED) //Condición mientras se realiza la conexión
  {
    Serial.print(".");          //Imprime en monitor serial puntos
    delay(250);                 // 250ms de espera
  }
  Serial.print("\nConectado al Wi-Fi"); //Al salir del bucle while, se realizó la correcta conexión
  Serial.println();             //Salto de línea
  //Una vez realizada conexión a Internet, se realiza conexión con la base de datos
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); //Inicialización de la base de datos con las credenciales definidas
  Firebase.reconnectWiFi(true); //reconexión de AP
}

```

**Figura 3.22** Función *void setup* (*NodeMCU*)

Antes del acondicionamiento de la señal del sensor se inicia en cero las variables de promedio e índice UV para evitar que tomen valores incorrectos, enseguida se inserta un bucle en el que se realiza la lectura del sensor de temperatura, se acondiciona la señal y, después de 5 mediciones obtenidas se promedian para obtener un valor más exacto (Figura 3.23).

```

void loop()
{
  promedio=0;                    //Encerar variable de promedio de sensor lm35
  int uvIndice=0;                //Encerar variable de índice uv de sensor de radiación
  for(int i=0; i<5; i++){        //Ciclo que toma 5 veces el dato de temperatura, y saca promedio para mayor exactitud
    sensorlm35=analogRead(A0);   //lectura de lm35
    temp=((sensorlm35*3300.0)/1023)/10; //fórmula de sensor para temperatura lm35 con 3.3V
    promedio=promedio+temp;      //se suma el valor tomado 5 veces
    delay(200);                  //espera de 200 ms
  }                               //salida de 5 lecturas
  promedio=promedio/5;          //promedio definitivo de temperatura
}

```

**Figura 3.23** Acondicionamiento del sensor de temperatura (*NodeMCU*)

Las siguientes dos líneas de código se encargan de crear una cadena de caracteres en el proyecto de *Firestore* que es donde se almacena la información obtenida por los sensores (Figura 3.24).

```

String nodo = "valor1";          //Creación de cadena de caracteres con nombre valor1 a Firestore
String nodol = "valor2";        //Creación de cadena de caracteres con nombre valor2 a Firestore

```

**Figura 3.24** Cadena de caracteres en *Firestore* (*NodeMCU*)

En esta sección de la función *void loop*, se habilita al módulo *NodeMCU* el poder enviar datos únicamente cuando los recibe, luego este microcontrolador recibe la información transmitida por *Arduino Uno* e imprime los datos de temperatura y radiación UV en el monitor serial. Finalmente, con las dos últimas líneas del programa se envían ambas mediciones a la base de datos de *Firestore* tal como se muestra en la Figura 3.25.

```

if (mySerial.available()>0) //Envia datos solo cuando recibe datos
{
  int uvIndice = mySerial.parseInt(); //El parseInt funciona cuando en arduino..
                                     //se usa un println en vez de un print.
                                     //guarda el dato recibido del arduino uno indice uv

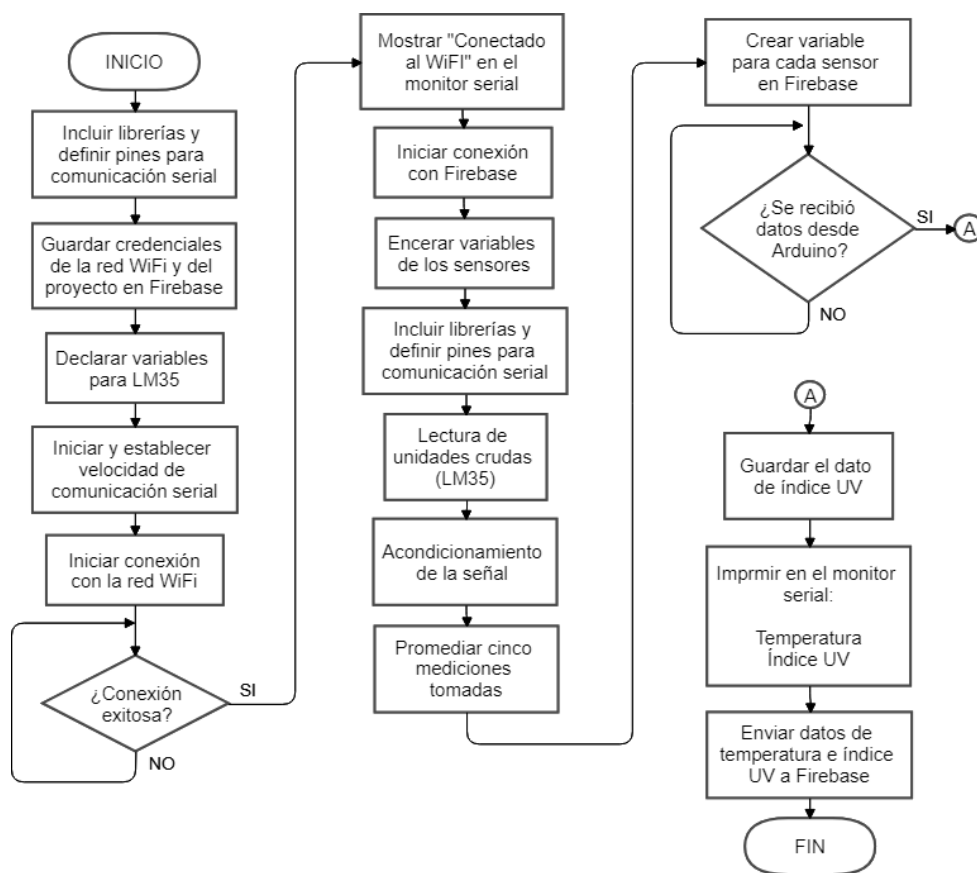
  Serial.println("Temperatura =" + String(promedio)); //Imprime en monitor serial dato de temperatura
  Serial.println("Indice UV =" + String(uvIndice)); //Imprime en monitor serial dato de indice uv

  Firebase.setInt(firebaseData, nodo, promedio); //Se envia dato de T a la base de datos en Firebase en valor1,
  Firebase.setInt(firebaseData, nodo1, uvIndice); //Se envia dato de Indice UV a la base de datos en Firebase, valor1
}
}

```

**Figura 3.25** Recepción y envío de datos (*NodeMCU*)

Para comprender de forma adecuada todo el código de programación desarrollado se muestra en la Figura 3.26 el respectivo diagrama de flujo.



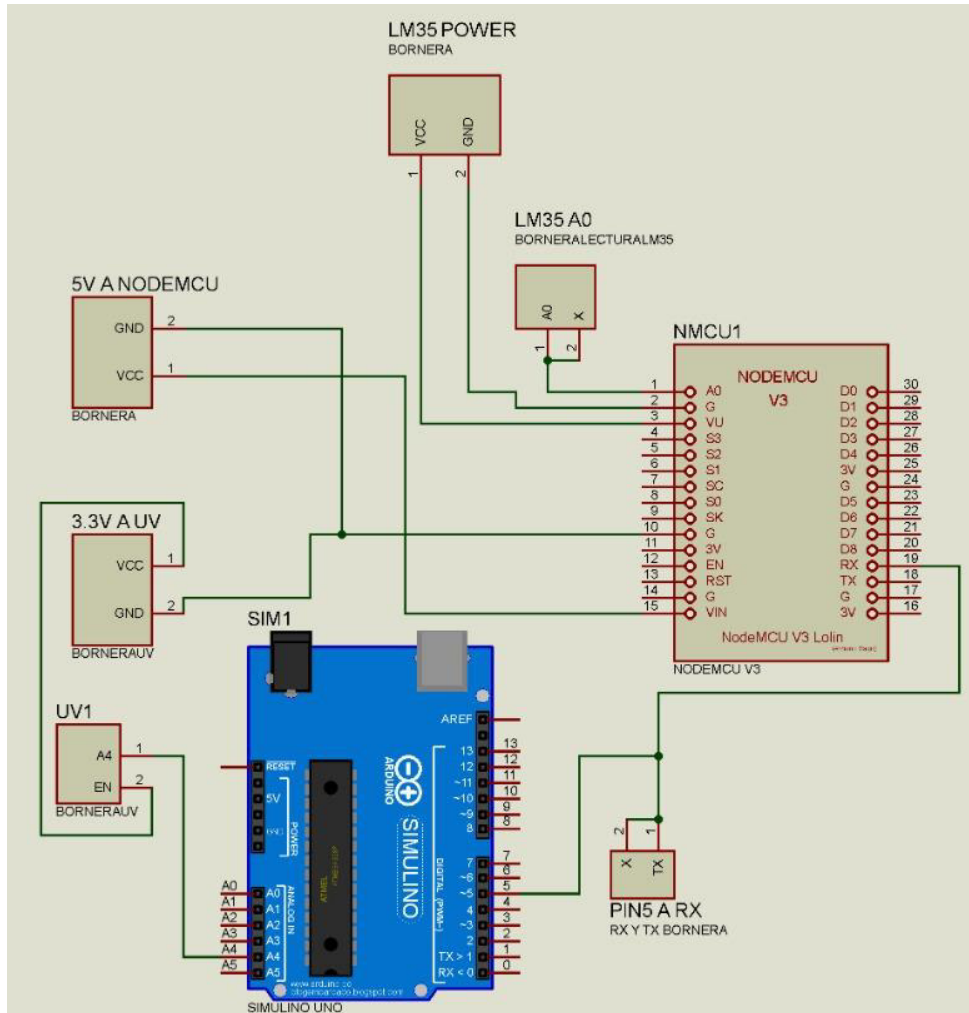
**Figura 3.26** Diagrama de flujo del programa de *NodeMCU*

Para la visualización de los códigos de programación debidamente comentados dirigirse al ANEXO 2.

Una vez desarrollados los códigos de programación, se diseñó el diagrama esquemático del circuito electrónico, para luego elaborar el correspondiente PCB de soporte y conexiones del sistema.

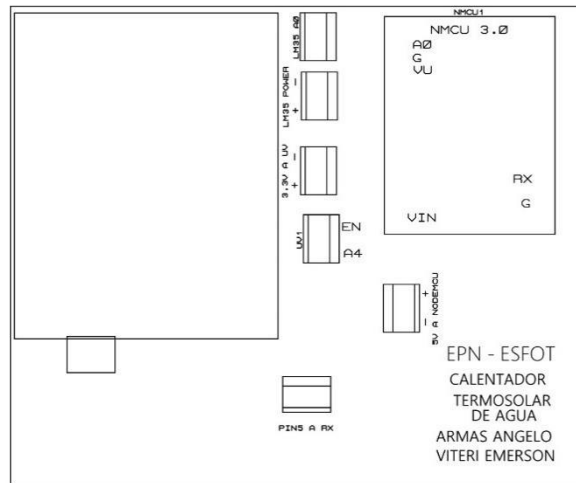
## Sistema de telemetría

Basándose en los elementos con los que cuenta el sistema, y con ayuda del *software Proteus* se diseñó el circuito electrónico para la conexión de los sensores y los microcontroladores. La Figura 3.27 muestra el diagrama esquemático del circuito donde se aprecia al sensor de temperatura conectado al módulo *WiFi* aprovechando su único pin analógico, mientras que, el sensor de radiación UV se conecta a *Arduino Uno*, además cada una de las conexiones fueron realizadas mediante borneras para una transmisión segura de señales y datos.

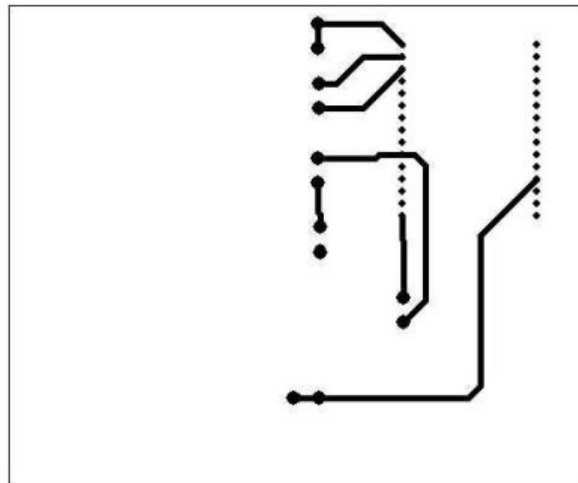


**Figura 3.27** Diagrama esquemático del sistema de telemetría

Posteriormente se ubicaron los elementos del circuito de manera eficiente para elaborar el diseño PCB tal como se muestra en la Figura 3.28 y Figura 3.29 respectivamente.



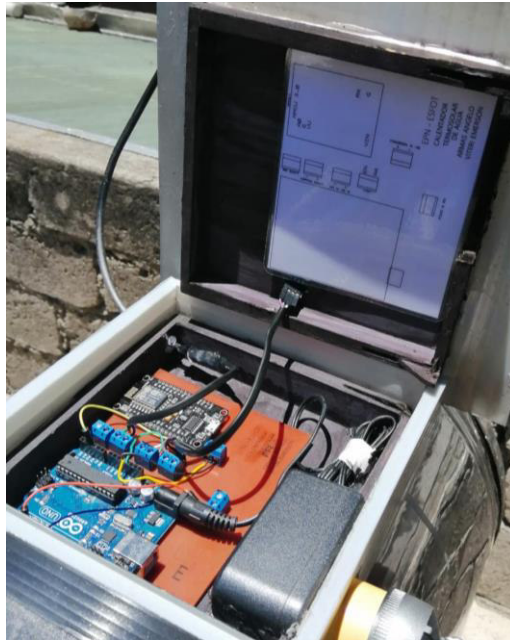
**Figura 3.28** Ubicación de los elementos del circuito



**Figura 3.29** Pistas de conexión

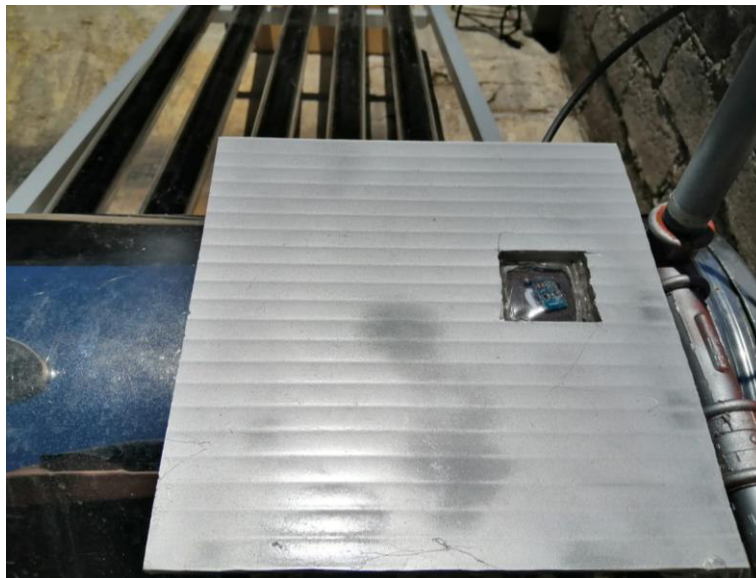
Previo al montaje de los elementos del circuito en el PCB, se elaboró una caja de madera posteriormente recubierta de policarbonato que contenga al circuito y lo proteja de las lluvias, de tal manera que solamente el sensor de radiación reciba luz del sol y cuente con las aberturas necesarias para su alimentación. En la Figura 3.30 se puede apreciar el circuito totalmente montado y colocado en su caja.





**Figura 3.30** Circuito Implementado

En la Figura 3.31 se muestra la caja recubierta del sistema de telemetría, donde se logra apreciar al sensor de radiación UV.



**Figura 3.31** Sistema de telemetría vista superior

### 3.4 Comunicación del sistema

Una vez implementado y alimentado el circuito de medida, se verificó si la conexión a la red *WiFi* fue exitosa a través del puerto serial en la plataforma de *Arduino IDE*, aquí se observa si las credenciales de la red colocadas en el código de programación coinciden y se establece la conexión (Figura 3.32). De igual manera se ingresa a la plataforma de *Firebase* para verificar si las mediciones obtenidas por los sensores están siendo transmitidas a la base de datos, en la parte superior de la Figura 3.33 se observa cuando los sensores no transmiten señales (valores igual a cero), mientras que, en la parte inferior se observan las mediciones recibidas por los sensores.

```
ssid = "Claro_REA0014342900";  
password = "CARMEN84616960328815";  
  
COM4  
.....  
Conectado al Wi-Fi
```

Figura 3.32 Conexión a la red *WiFi*

```
calentadorsolaragua-default-rtdb  
├─ valor1: "\"0\""  
└─ valor2: "\"0\""  
  
calentadorsolaragua-default-rtdb  
├─ valor1: 42  
└─ valor2: 0
```

Figura 3.33 Comunicación con la base de datos

A continuación, se desarrolló una aplicación móvil que corresponde a la interfaz de usuario, en donde se pueden observar los datos del sistema y encontrar los manuales de usuario y mantenimiento.

#### Desarrollo de la aplicación en *App Inventor*

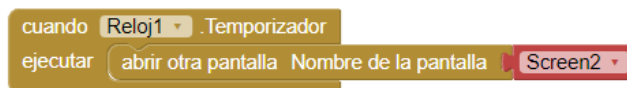
El objetivo principal de esta aplicación es acceder a la información obtenida de los sensores, que se encuentra alojada en *Firebase*, y mostrarla en dispositivos con *Android* para el monitoreo de temperatura y radiación en tiempo real. Esta aplicación está

formada por dos pantallas, la primera es la pantalla de inicio y la segunda es la pantalla de monitoreo de información.

En la Figura 3.34 se observa la pantalla de inicio en la que se han colocado los logotipos de la Escuela Politécnica Nacional (EPN), ESFOT y *SOLAR HELP* (nombre de la aplicación). Además, como componente no visible se le asignó un temporizador de espera de 5 segundos para ingresar a la segunda pantalla. Tal como se muestra en su código de bloques (Figura 3.35).

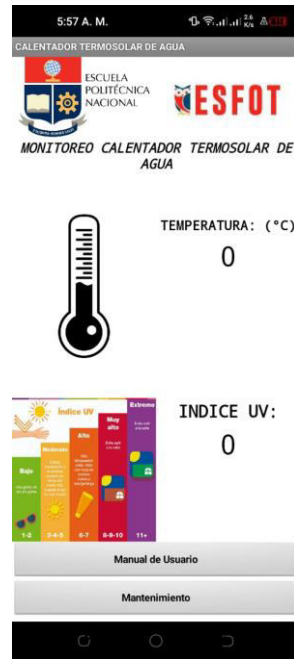


**Figura 3.34** Pantalla de inicio de *SOLAR HELP*



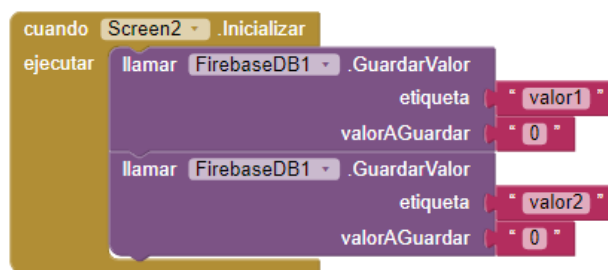
**Figura 3.35** Código de bloques (Pantalla de inicio)

En la Figura 3.36 se observa la pantalla de monitoreo de información.



**Figura 3.36** Pantalla de monitoreo de *SOLAR HELP*

Se agregan a la segunda pantalla los componentes gráficos, imágenes representativas para cada variable física, botones, cuadros de texto, etc. Al igual que en la pantalla de inicio se le agregaron componentes no visibles como temporizador, conexión a *Firebase* y redirección a una página externa para acceder a los manuales de usuario y mantenimiento. Cuando inicializa la segunda pantalla de la aplicación se procede a escribir cero en el valor de cada medición para evitar errores de lectura anteriores, en la Figura 3.37 se muestran los bloques de programación que cumplen con esta función.

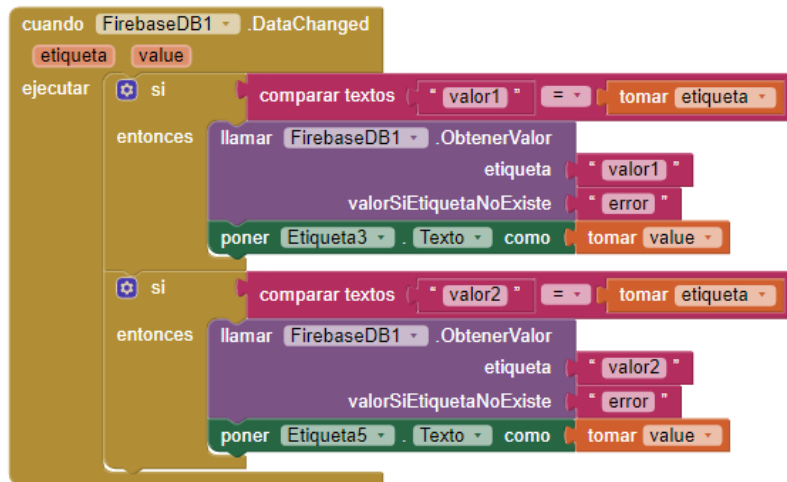


**Figura 3.37** Bloques iniciales (Pantalla de monitoreo)

Seguidamente en la Figura 3.38 se detalla el proceso mediante el cual se presentan en la aplicación, los datos alojados en *Firebase*. En primer lugar, se localiza la base de datos a utilizar, esto mediante la dirección *URL* y *Token* del proyecto en *Firebase*.

Luego se utiliza una comparación entre el valor de 0 que se escribe al iniciar la pantalla y el dato correspondiente a cada una de las variables, si los valores coinciden con el 0

es un indicativo que se encuentra operando correctamente y de inmediato se revisa si los sensores están transmitiendo mediciones, en caso de no hacerlo se muestra un mensaje de error. Por el contrario, si los sensores detectan nuevas mediciones diferentes a 0, se toma el valor alojado en *Firestore* y se presenta en el sitio respectivo de cada variable.

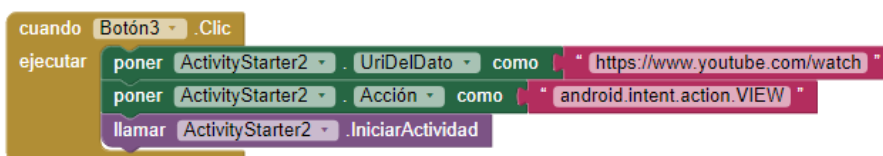


**Figura 3.38** Bloques para presentación de datos (Pantalla de monitoreo)

Finalmente se desarrollaron los bloques correspondientes a los botones de acceso a los manuales de usuario y mantenimiento para lo cual se ha establecido que al presionar alguno de los botones se redirige, en el navegador web, a una *URL* específica donde se alojan los videos correspondientes a cada uno los manuales, tal como se muestra en la Figura 3.39 y Figura 3.40 respectivamente.



**Figura 3.39** Bloques del botón de manual de usuario (Pantalla de monitoreo)



**Figura 3.40** Bloques del botón de mantenimiento (Pantalla de monitoreo)

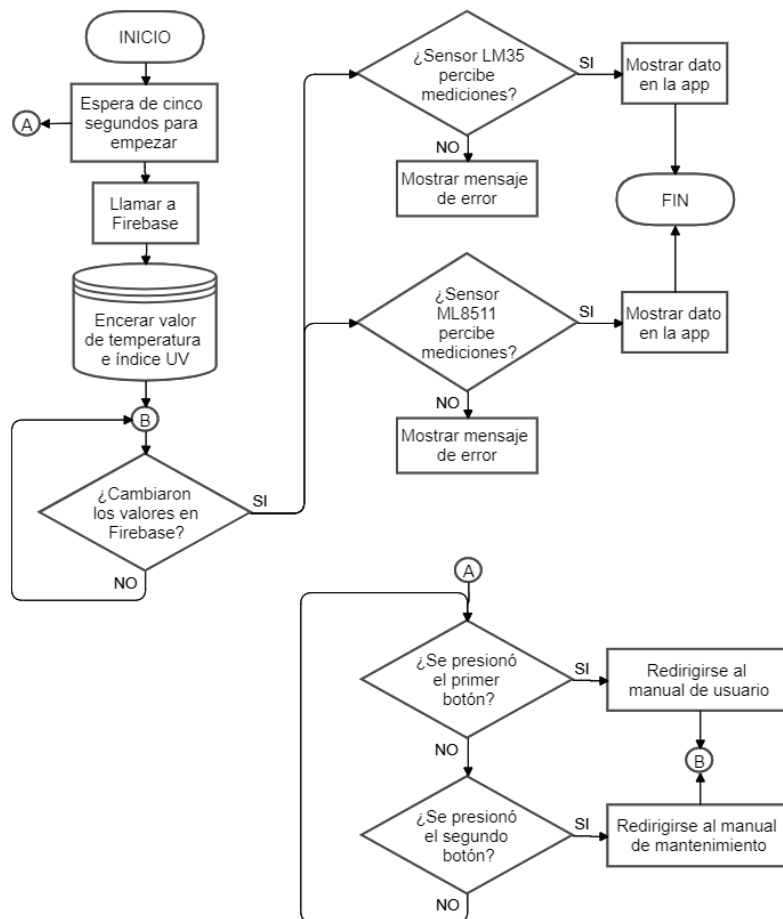
En la Figura 3.41 se muestra la aplicación funcionando correctamente puesto que no aparece el mensaje de error, es decir, el sistema ha establecido una exitosa conexión

con la base de datos y consecuentemente la aplicación ha podido acceder a la información obtenida por los sensores.



**Figura 3.41** Conexión exitosa para el monitoreo de datos

En la Figura 3.42 se muestra el diagrama de flujo del funcionamiento de la aplicación móvil.



**Figura 3.42** Diagrama de flujo de la aplicación

### 3.5 Pruebas y Análisis de Resultados

Una vez que el módulo termosolar y el sistema de telemetría fueron implementados, para evidenciar el correcto funcionamiento del sistema fue necesario realizar las siguientes pruebas:

#### Calentamiento de agua

El objetivo principal de esta prueba es evidenciar que el agua se calienta por acción de la energía solar, para llevar a cabo esta prueba se utilizó la termocupla del multímetro, la cual se introdujo primero en el tanque de autollenado de agua fría para registrar esta temperatura y luego en el termotanque (agua caliente). De esta manera se verifica la temperatura a la que se encuentra el agua directa de la red de distribución de agua potable (Figura 3.43), y la contenida dentro del termotanque (Figura 3.44).



Figura 3.43 Temperatura en la entrada de agua fría



Figura 3.44 Temperatura en la salida de agua caliente

### Medición de datos de temperatura y radiación solar

En esta prueba se busca registrar las mediciones obtenidas por los sensores de tal manera que sean claros los cambios que experimenta el agua en el termotanque a lo largo del día. Para realizar esta prueba se recopilaron datos cada hora a partir de las 07:00 hasta las 18:00 y a las 23:00, con el fin de evidenciar como se mantiene la temperatura del agua y el índice UV. En la Tabla 3.2 se muestran los datos recopilados.

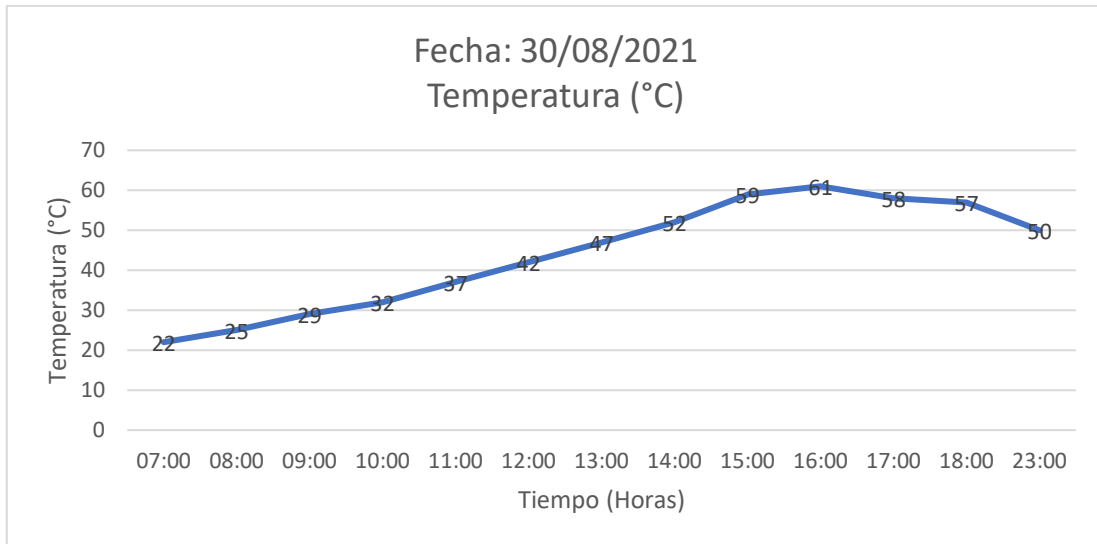
**Tabla 3.2** Medición de temperatura y radiación UV

Hora	Temperatura (°)	Índice UV	Cambio de temperatura (°)
7:00	22	2	
8:00	25	3	3
9:00	29	3	4
10:00	32	4	3
11:00	37	4	5
12:00	42	4	5
13:00	47	5	5
14:00	52	4	5
15:00	59	6	7
16:00	61	3	2
17:00	58	2	-3
18:00	57	2	-1
23:00	50	1	-7

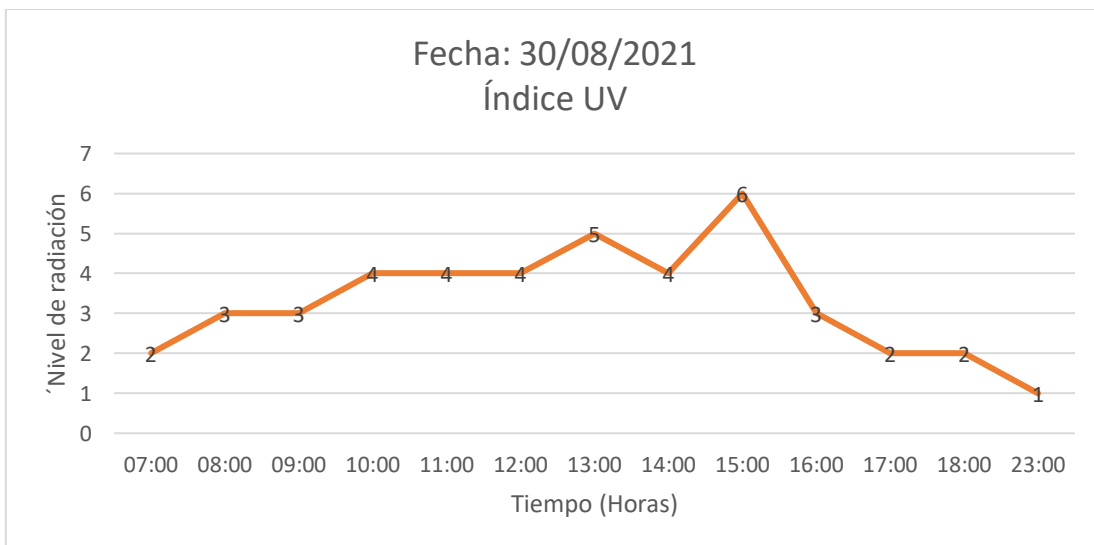
La recopilación de datos indica que a pesar de que el sistema se haya encontrado operando con climas nublados, el captador solar cumple eficientemente su objetivo pues ante las condiciones, el agua ha logrado alcanzar una temperatura máxima de 61 (°C).

A continuación, en la Figura 3.45 y Figura 3.46, se pueden visualizar las gráficas de temperatura e índice UV respectivamente.





**Figura 3.45** Gráfica de temperatura en relación con el tiempo



**Figura 3.46** Gráfica de índice UV en relación con el tiempo

### **Comunicación entre los microcontroladores y los sensores**

Esta prueba se llevó a cabo para corroborar que los sensores se encuentran funcionando de acuerdo con lo establecido en el código de programación desarrollado en la plataforma de trabajo *Arduino IDE*. Para realizar las pruebas de comunicación se conecta el sistema a la alimentación y haciendo uso del monitor serial de *Arduino* se procede a observar los datos obtenidos por los sensores.

La Figura 3.47 corresponde a las mediciones realizadas por el sensor de radiación UV en el siguiente orden: unidades digitales, voltaje de salida acondicionado y el nivel o índice de radiación al que pertenece.

```
COM3
313
1.01
Indice UV:3
483
1.56
Indice UV:7
525
1.69
Indice UV:8
313
1.01
Indice UV:3
475
1.53
Indice UV:7
```

**Figura 3.47** Comunicación con el sensor de radiación

Adicionalmente, en la Figura 3.48, se presenta el monitor serial que confirma el establecimiento de comunicación entre el módulo *WiFi* y la red doméstica utilizada en el momento de realizar esta prueba.

```
COM4
.....
Conectado al Wi-Fi
```

**Figura 3.48** Comunicación con la red *WiFi*

Debido a que el módulo *NodeMCU* es el encargado de recibir la información de índice UV, acondicionar la señal obtenida por el sensor de temperatura y enviar ambos datos a *Firebase*; en el monitor serie se observa el estado de la conexión inalámbrica y las mediciones captadas por ambos sensores, tal como se muestra en la Figura 3.49.

```
COM4
.....
Conectado al Wi-Fi
Temperatura =32.65
Indice UV =6
```

**Figura 3.49** Comunicación con el sensor de temperatura

## Transmisión, monitoreo y registro de datos

Esta prueba se realizó con el fin de constatar que la transmisión de información obtenida por parte de los sensores se transmite de manera correcta de acuerdo con el acondicionamiento de cada sensor.

En base a la prueba anterior que evidencia la comunicación del sistema, se ingresó al proyecto en *Firebase* y se pudo verificar el envío de datos desde el módulo *WiFi* hacia la base de datos alojada en la nube. Previo al monitoreo de datos, es necesario instalar la aplicación móvil *Solar Help* (interfaz de usuario) y, una vez instalada se ingresa para verificar si las mediciones coinciden en ambas plataformas. En la Figura 3.50 y Figura 3.51 se observan los datos en *Firebase* y la aplicación respectivamente.

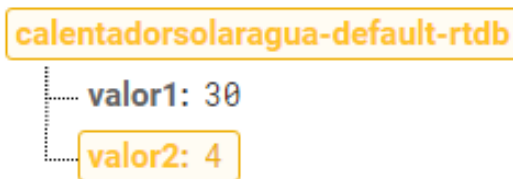


Figura 3.50 Datos en *Firebase*

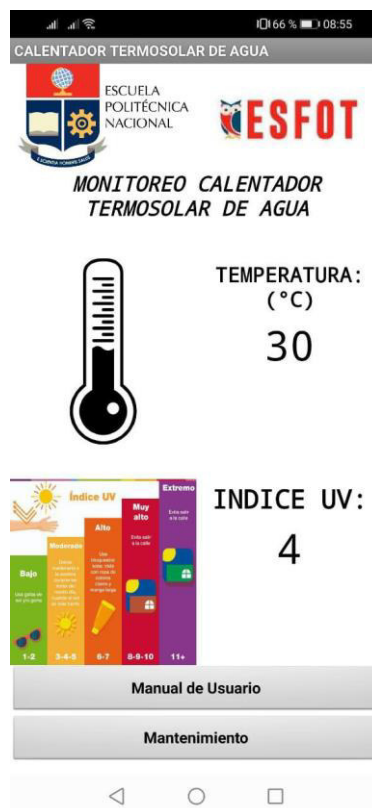


Figura 3.51 Datos en *SOLAR HELP*

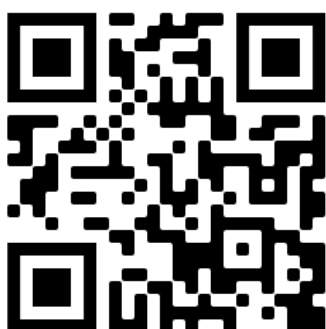
Finalmente, como constancia del funcionamiento del sistema, se realizó un video en el que se muestran todas las pruebas realizadas en el módulo termosolar. Se puede acceder al video mediante el código QR de la Figura 3.52.



**Figura 3.52** Código QR de las pruebas de funcionamiento

### **3.6 Manual de Uso y Mantenimiento**

Se realizaron dos videos explicativos para el correcto uso y mantenimiento del sistema de telemetría y el módulo termosolar de agua caliente, en la Figura 3.53 y Figura 3.54 se encuentran dos códigos QR para acceder a estos videos.



**Figura 3.53** Código QR del manual de uso



**Figura 3.54** Código QR del manual de mantenimiento

## 4 CONCLUSIONES Y RECOMENDACIONES

### 4.1 Conclusiones

- El uso de la herramienta *PVGIS* fue fundamental para conocer los datos de recurso solar para el diseño y construcción del módulo termosolar de agua caliente ya que permitieron comprender las características necesarias para instalar el sistema completo en ese punto geográfico.
- El diseño 3D del módulo termosolar fue desarrollado con la herramienta *SketchUp* por los conceptos básicos que proporciona para realizar el modelaje con precisión, además de la posibilidad de exportar el diseño en formato *DWG* lo cual resultó útil para los planos esquemáticos del sistema.
- El colector solar de tubos al vacío es más eficaz que los de plato plano gracias a la forma y separación de los tubos de vidrio, de esta manera se aprovechan los rayos del sol desde su salida hasta su puesta, por el contrario, los colectores planos alcanzan su máximo rendimiento al mediodía.
- La utilización de empaques de silicón en el punto de conexión del termotanque con el colector solar es sumamente importante ya que aseguran los tubos de vidrio a las entradas del termotanque evitando que salgan disparados por la presión que ejerce el agua en ellos.
- El envío de datos desde *Arduino Uno* hacia el módulo *WiFi* se realizó a través de los pines 2 y 5, utilizando la librería *SoftwareSerial* que permite la transmisión y recepción de datos en pines digitales, esto con el fin de mostrar otra alternativa a la comunicación *UART* (pines 0 y 1).
- La plataforma *Firebase* brindó grandes beneficios al desarrollar la aplicación móvil ya que sirvió de base para alojar los datos obtenidos por los sensores y dar accesibilidad al usuario para monitorearlos en cualquier momento.
- El entorno de trabajo de *MIT App Inventor* facilitó de gran manera el desarrollo de la aplicación de monitoreo gracias a su interfaz gráfica intuitiva para agregar elementos indispensables como la base de datos alojada en la nube y botones de acción, así como también la posibilidad de probarla previamente con un emulador de *Android* con el objetivo de corregir errores antes de ser instalada.
- La medición de temperatura se realizó mediante el promedio de cinco valores obtenidos por el sensor *LM35* cada 200 (ms), con el fin de disminuir las variaciones en las mediciones realizadas en el agua del termotanque y que la medida sea correcta.

- La tecnología termosolar permite obtener energía limpia y sostenible, la cual establece el ahorro energético en un sistema domiciliario en cuanto a agua caliente sanitaria, ya que a través del presente proyecto se obtuvieron temperaturas adecuadas, lo cual demuestra la importancia de este tipo de sistemas.
- Monitorear la temperatura y la radiación del módulo termosolar a través del sistema de telemetría permite obtener y verificar, en tiempo real y desde cualquier lugar, estos datos para su posterior análisis.

## 4.2 Recomendaciones

- Adaptar un tanque que sirva para el llenado automático de agua fría, instalándolo a mayor altura en comparación a la salida de agua caliente, con el objetivo de brindar la presión necesaria para expulsar el agua con mayor fuerza.
- Es importante tener en cuenta que al momento en que el sistema requiera conectarse a otra red *WiFi*, es necesario acceder al entorno de trabajo *Arduino IDE* y editar las credenciales correspondientes a la nueva red en el código de programación de *NodeMCU*.
- Es indispensable realizar el mantenimiento del sistema termosolar para limpiar las impurezas del agua que se almacenan con el pasar del tiempo, es decir, enfocarse en reducir los minerales como calcio, magnesio y potasio que pueden afectar a la eficiencia de los tubos al vacío.
- Se puede utilizar una resistencia sumergible para mantener el calor del agua dentro del termotanque durante las noches, ya que, en base a las pruebas de funcionamiento la temperatura del agua disminuyó 7 (°C).
- Para el desarrollo de proyectos que requieran enviar datos por comunicación serial y esté descartado el uso de los pines 0 y 1 de la comunicación *UART* debido a que ya están en uso o alguna otra razón, se recomienda tomar en cuenta la librería *SoftwareSerial* de *Arduino* que permite la comunicación serial en pines digitales.
- Se recomienda visualizar los videos correspondientes a los manuales de uso y mantenimiento para comprender de mejor manera el modo de funcionamiento del módulo, sus características, y como evitar fallas en el sistema.
- Para complementar el trabajo realizado, se recomienda el desarrollo de una página *web*, la cual mediante un servidor muestre en tiempo real los datos de temperatura y radiación.

## 5 REFERENCIAS BIBLIOGRÁFICAS

- [1] Gomelsky Roberto, "Sustainable Energy For All," *Ministerio de Electricidad y Energía Renovable*, 2013. [https://www.seforall.org/sites/default/files/Ecuador\\_RAGA\\_ES\\_Released.pdf](https://www.seforall.org/sites/default/files/Ecuador_RAGA_ES_Released.pdf) (accessed Sep. 01, 2020).
- [2] Blog Ecofener, "¿Qué son y cómo funcionan los captadores solares térmicos para calentar el agua?," 2018. <https://ecofener.com/blog/funcionan-los-captadores-solares-termicos-calentar-agua-piscinas/> (accessed Sep. 02, 2020).
- [3] Global solare, "¿Por qué invertir en un calentador solar?" <https://globalsolare.com/2018/04/16/por-que-invertir-en-un-calentador-solar/> (accessed Sep. 02, 2020).
- [4] "Que es y como funciona un calentador solar," *AQUA*, Nov. 06, 2015. <https://calentador.mx/blog/que-es-y-como-funciona-un-calentador-solar--n2> (accessed Sep. 08, 2021).
- [5] "SOL y AGUA CALIENTE. ¿Cómo funciona un calentador solar? | Diariodeciencias.com.ar," *Diario de Ciencias Aplicadas*, Jan. 10, 2019.
- [6] Global solare, "¿CÓMO ESTÁ COMPUESTO UN CALENTADOR SOLAR?" <https://globalsolare.com/2018/03/28/como-esta-compuesto-un-calentador-solar/> (accessed Sep. 02, 2020).
- [7] "NodeMCU v2 ESP8266 WiFi," *Naylamp Mechatronics*. <https://naylampmechatronics.com/espressif-esp/153-nodemcu-v2-esp8266-wifi.html> (accessed Sep. 09, 2021).
- [8] S. López, "Firebase: qué es, para qué sirve, funcionalidades y ventajas," May 17, 2020. <https://www.digital55.com/desarrollo-tecnologia/que-es-firebase-funcionalidades-ventajas-conclusiones/> (accessed Sep. 08, 2021).
- [9] Google, "Firebase Realtime Database | Store and sync data in real time." <https://firebase.google.com/products/realtime-database?hl=es> (accessed Sep. 08, 2021).
- [10] "¿Qué es AppInventor? - Curso de introducción a la programación de apps con AppInventor." <https://www.programoergosum.com/cursos-online/appinventor/27-curso-de-programacion-con-app-inventor/primeros-pasos> (accessed Sep. 08,

- 2021).
- [11] F. Posada Prieto, "Creando aplicaciones para móviles Android con MIT App Inventor 2 - INTEF," *INTEF*, 2019. [https://intef.es/observatorio\\_tecno/creando-aplicaciones-para-moviles-android-con-mit-app-inventor-2/](https://intef.es/observatorio_tecno/creando-aplicaciones-para-moviles-android-con-mit-app-inventor-2/) (accessed Sep. 08, 2021).
  - [12] *Como inclinar mi panel solar ? a que angulo ? - YouTube*. La tienda de Juank, 2017.
  - [13] "TERMOTANQUE / CALEFÓN SOLAR ," *Ferreteria San Luis*. <https://www.ferreterasanluis.com/p/termotanque-calefon-solar-200lts-lusqtoff-lc-sol200> (accessed Sep. 09, 2021).
  - [14] "Sensor de temperatura sumergible," *PatagonaTec*. <https://compras.patagoniatec.com/productos/sensor-de-temperatura-sumergible/> (accessed Sep. 09, 2021).
  - [15] "Sensor Luz Ultravioleta UV ML8511 | AV Electronics," *AV Electronics*. <https://avelectronics.cc/producto/sensor-luz-ultravioleta-uv-ml8511/> (accessed Sep. 09, 2021).



## **ANEXOS**

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



# ESCUELA POLITECNICA NACIONAL

Campus Politécnico "J. Rubén Orellana R

Quito, 04 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 del sistema de telemetría del módulo didáctico termosolar, el cual fue implementado por los estudiantes Armas Rea Angelo Leonardo y Viteri Chávez Emerson Saúl.

El proyecto cumple con los requerimientos de diseño y parámetros necesarios para que los usuarios puedan utilizar el módulo didáctico 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: CÓDIGOS DE PROGRAMACIÓN**

## Código de *Arduino UNO*

```
#include <SoftwareSerial.h>    //Librería para comunicación serie en pines digitales

#define rxPin 2                //definir pin 2 como recepción serie

#define txPin 5                //definir pin 5 como transmisión serie

SoftwareSerial Trans(rxPin, txPin); //Pines para Rx y Tx, y denominar Trans como
monitor serial

//Declaración de variables del proceso

int uv=A3;

int uvIndice;

void setup()

{

  pinMode (txPin , OUTPUT);    //Definir pin de transmisión como salida

  Serial.begin(9600);          //Iniciar monitor serial de Arduino, velocidad 9600

  Trans.begin(9600);           //Velocidad de envío para la transmisión al NodeMCU, e
iniciar monitor serial

  pinMode(uv, INPUT);          //Pin de sensor ML8511 radiación como entrada

}

void loop()

{

  int uvLevel = analogRead(uv);    //lectura de sensor de radiación

  float outputVoltage = 3.3 * uvLevel/1024;//cálculo de la salida de voltaje del sensor uv

  if(outputVoltage<=0.7){          //condición de la salida de voltaje,entre 0 y
0.7

    uvIndice=0;                    //declaración del nivel 0 de índice de radiación

  }else if(outputVoltage>0.7 && outputVoltage<=0.83){ //condición de la salida de
voltaje,entre 0.71 y 0.83
```

```

uvIndice=1; //declaración del nivel 1 de índice de radiación
}else if(outputVoltage>0.83 && outputVoltage<=0.96){ //condición de la salida de
voltaje,entre 0.84 y 0.96

uvIndice=2; //declaración del nivel 2 de índice de radiación
}else if(outputVoltage>0.96 && outputVoltage<=1.09){ //condición de la salida de
voltaje,entre 0.97 y 1.09

uvIndice=3; //declaración del nivel 3 de índice de radiación
}else if(outputVoltage>1.09 && outputVoltage<=1.22){ //condición de la salida de
voltaje,entre 1.10 y 1.22

uvIndice=4; //declaración del nivel 4 de índice de radiación
}else if(outputVoltage>1.22 && outputVoltage<=1.35){ //condición de la salida de
voltaje,entre 1.23 y 1.35

uvIndice=5; //declaración del nivel 5 de índice de radiación
}else if(outputVoltage>1.35 && outputVoltage<=1.48){ //condición de la salida de
voltaje,entre 1.36 y 1.48

uvIndice=6; //declaración del nivel 6 de índice de radiación
}else if(outputVoltage>1.48 && outputVoltage<=1.61){ //condición de la salida de
voltaje,entre 1.49 y 1.61

uvIndice=7; //declaración del nivel 7 de índice de radiación
}else if(outputVoltage>1.61 && outputVoltage<=1.74){ //condición de la salida de
voltaje,entre 1.62 y 1.74

uvIndice=8; //declaración del nivel 8 de índice de radiación
}else if(outputVoltage>1.74 && outputVoltage<=1.87){ //condición de la salida de
voltaje,entre 1.75 y 1.87

uvIndice=9; //declaración del nivel 9 de índice de radiación
}else if(outputVoltage>1.87 && outputVoltage<=2){ //condición de la salida de
voltaje,entre 1.88 y 2

uvIndice=10; //declaración del nivel 10 de índice de radiación

```

```

}else if(outputVoltage>2 && outputVoltage<=2.13){ //condición de la salida de
voltaje,entre 2.01 y 2.13

    uvIndice=11; //declaración del nivel 11 de índice de radiación

}else if(outputVoltage>2.13 && outputVoltage<=2.26){ //condición de la salida de
voltaje,entre 2.14 y 2.26

    uvIndice=12; //declaración del nivel 12 de índice de radiación

}else if(outputVoltage>2.26 && outputVoltage<=2.39){ //condición de la salida de
voltaje,entre 2.27 y 2.39

    uvIndice=13; //declaración del nivel 13 de índice de radiación

}else if(outputVoltage>2.39 && outputVoltage<=2.52){ //condición de la salida de
voltaje,entre 2.40 y 2.52

    uvIndice=14; //declaración del nivel 14 de índice de radiación

}else if(outputVoltage>=2.53){ //condición de la salida de voltaje,entre
2.53 y 3.3

    uvIndice=15; //declaración del nivel 15 de índice de radiación

}

//Mostrar el valor en el monitor serie de Arduino.

Serial.println(uvLevel); //Imprime en monitor serial la lectura del sensor
uv

Serial.println(outputVoltage); //Imprime en monitor serial la salida del voltaje

Serial.println("Indice UV:"+String(uvIndice)); //Imprime en monitor serial el Índice UV

delay(1000); //1 seg de espera

Trans.println(uvIndice); //Para enviar la variable de Índice UV por el pin 5 del
Arduino UNO

}

```

## **Código de *NodeMCU***

```
#include <ESP8266WiFi.h>           //Librería que permite establecer comunicación
inalámbrica

#include <FirebaseESP8266.h>       //Librería que permite conexión de la base de
datos en firebase a ESP8266

#include <SoftwareSerial.h>        //Librería para comunicación serie en pines digitales

SoftwareSerial mySerial(3, 15);    //Pines para Rx y Tx, y denominar mySerial como
monitor serial

// Credenciales WiFi

const char *ssid = "Claro_REA0014342900";    // nombre de la red

const char *password = "CARMEN84616960328815"; //contraseña de la red

// Credenciales Proyecto Firebase

const char *FIREBASE_HOST = "calentadorsolaragua-default-rtdb.firebaseio.com";
//Nombre del Host Firebase

const char *FIREBASE_AUTH = "ZjHeUtRH6AcIQlOXcmQMOkEpUptm8jdF92lyd7iD";
//LLave de Autorización

// Objeto de datos de Firebase en el ámbito global

FirebaseData firebaseData;        //Nombre de la Base de Datos

float promedio;                   //Declaración de variable para Im35

float temp;                        //Declaración de variable para Im35

float sensorIm35;                 //Declaración de variable para Im35

void setup()

{

    Serial.begin(9600);           //Iniciar monitor serial de Arduino, velocidad 9600

    mySerial.begin(9600);        //Velocidad para la recepción al NodeMCU, e iniciar
monitor serial

    WiFi.begin(ssid, password);  //Iniciar conexión con los datos de la red definidos
arriba
```



```

while (WiFi.status() != WL_CONNECTED) //Condición mientras se realiza la conexión
{
  Serial.print(".");          //Imprime en monitor serial puntos
  delay(250);                // 250ms de espera
}

Serial.print("\nConectado al Wi-Fi"); //Al salir del bucle while, se realizó la correcta
conexión

Serial.println();           //Salto de linea

//Una vez realizada conexión a Internet, se realiza conexión con la base de datos

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); //Inicialización de la base de
datos con las credenciales definidas

  Firebase.reconnectWiFi(true);           //reconexión de AP
}

void loop()

{

  promedio=0;          //Encerar variable de promedio de sensor lm35

  int uvIndice=0;      //Encerar variable de índice uv de sensor de radiación

  for(int i=0; i<5; i++){ //Ciclo que toma 5 veces el dato de temperatura, y saca
promedio para mayor exactitud

    sensorlm35=analogRead(A0); //lectura de lm35

    temp=((sensorlm35*3300.0)/1023)/10; //fórmula de sensor para temperatura lm35 con
3.3V

    promedio=promedio+temp; //se suma el valor tomado 5 veces

    delay(200);           //espera de 200 ms

  }                       //salida de 5 lecturas

  promedio=promedio/5;   //promedio definitivo de temperatura

```

```

String nodo = "valor1";          //Creación de cadena de caracteres con nombre valor1
a Firebase

String nodo1 = "valor2";        //Creación de cadena de caracteres con nombre valor2
a Firebase

if (mySerial.available()>0)     //Envía datos solo cuando recibe datos
{
    int uvIndice = mySerial.parseInt(); //El parseInt funciona cuando en Arduino..

        //se usa un println en vez de un print.

        //guarda el dato recibido del Arduino uno índice uv

    Serial.println("Temperatura =" + String(promedio)); //Imprime en monitor serial dato de
temperatura

    Serial.println("Índice UV =" + String(uvIndice)); //Imprime en monitor serial dato de
índice uv

    Firebase.setInt(firebaseData, nodo,promedio); //Se envía dato de T a la base de
datos en Firebase en valor1,

    Firebase.setInt(firebaseData, nodo1,uvIndice); //Se envía dato de Índice UV a la base
de datos en Firebase, valor1

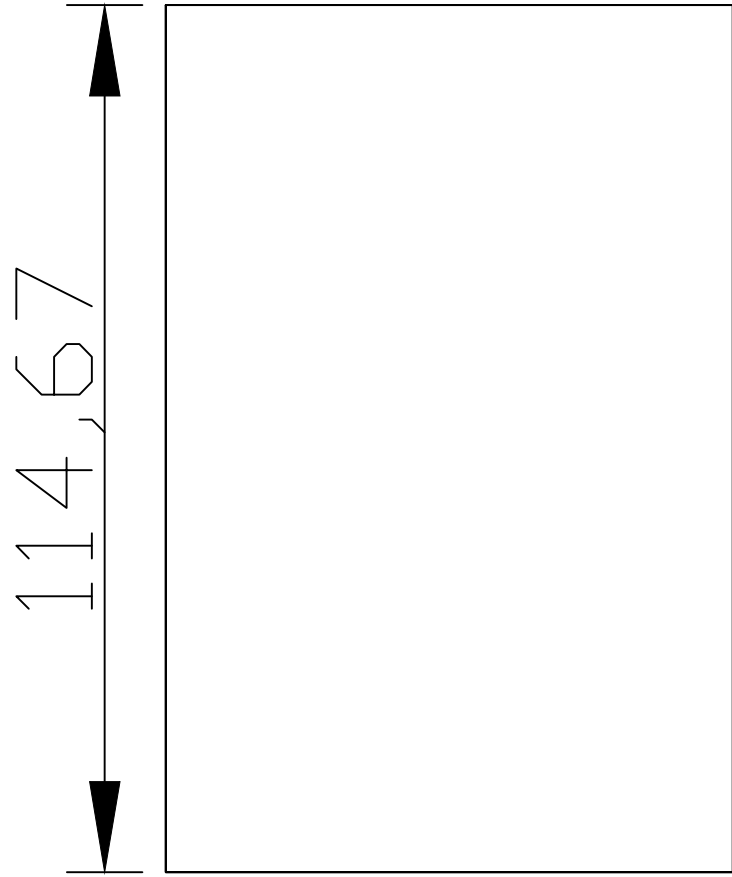
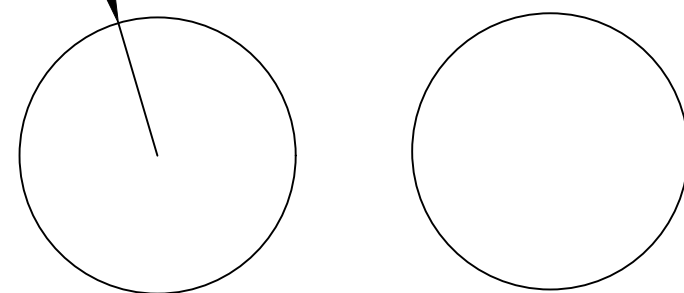
}
}

```

## **ANEXO 3: PLANOS Y ESQUEMAS**

R18,25

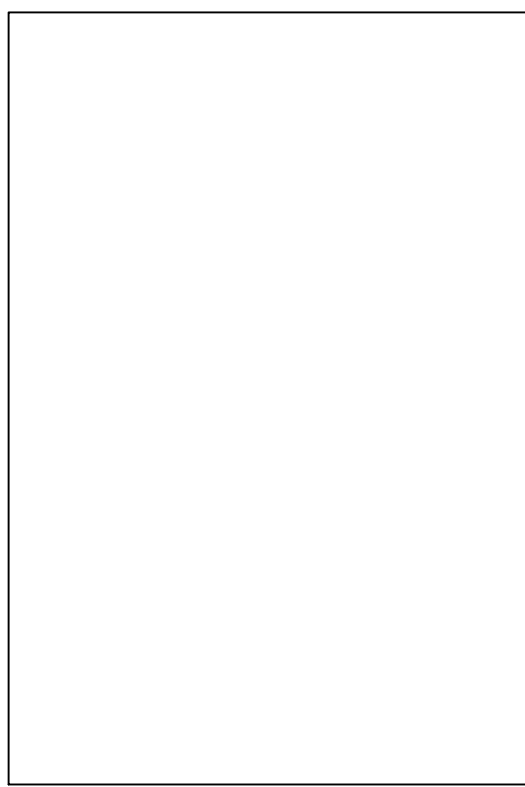
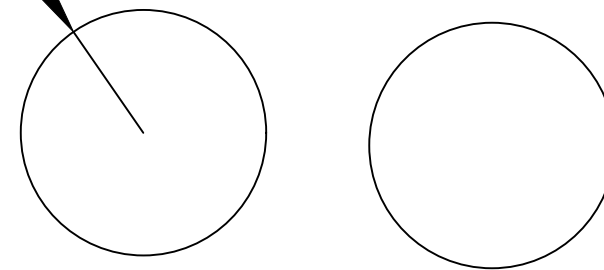
TANQUE EXTERNO



75

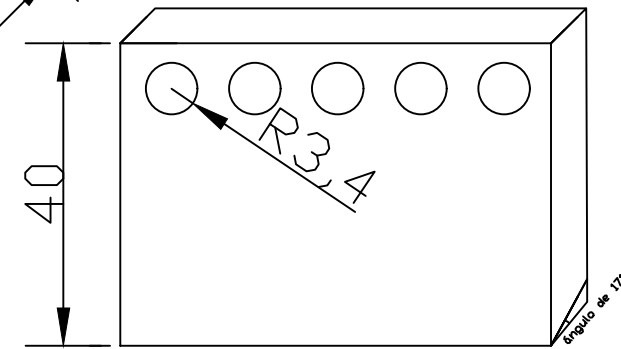
R16,25

TANQUE INTERNO



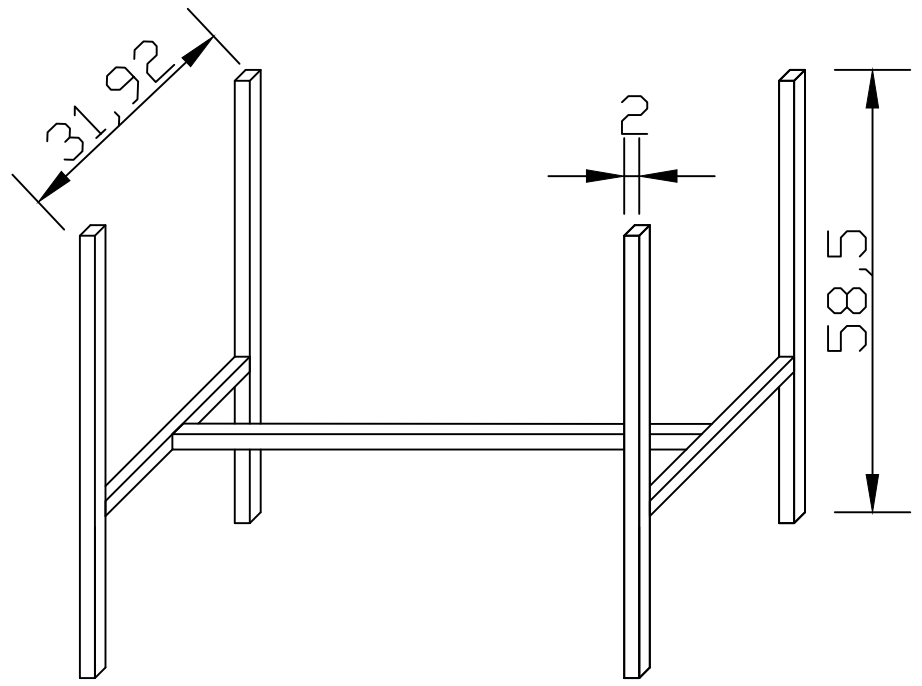
69

5/54



57

BASE TUBOS

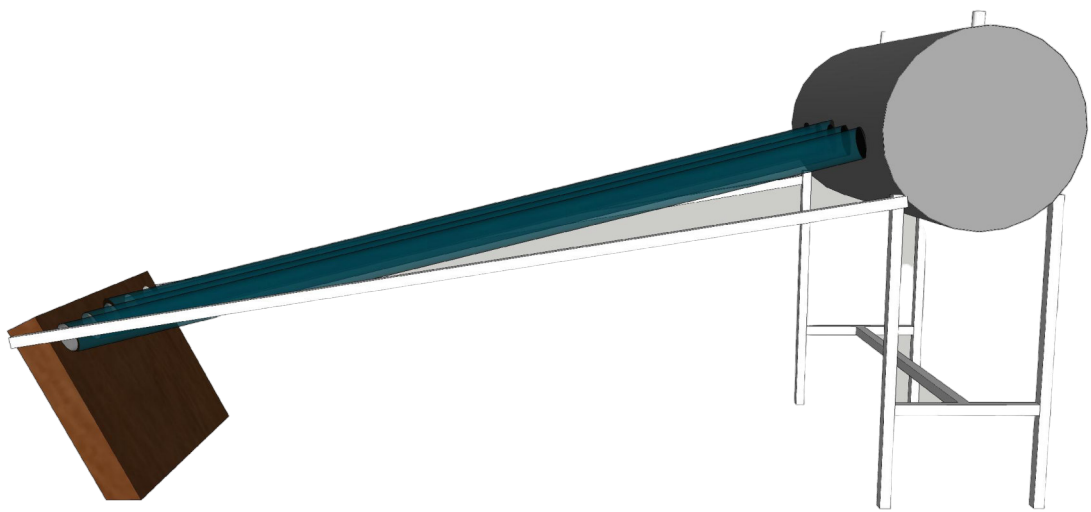
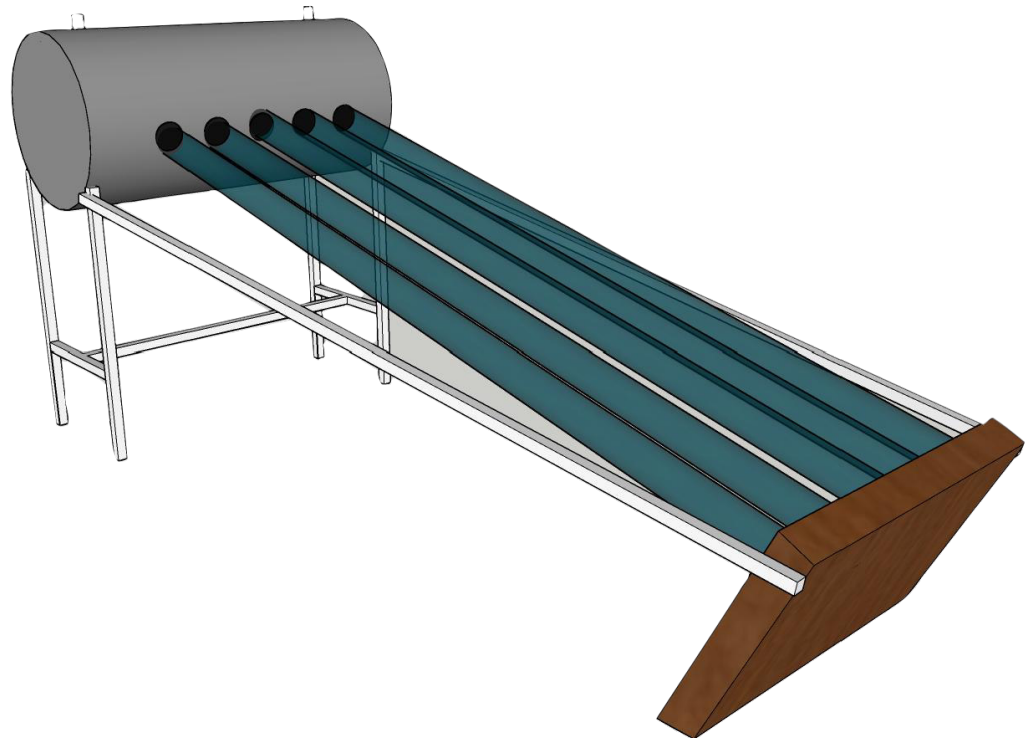


74

BASE TANQUE

ESCUELA POLITÉCNICA NACIONAL - ESFOT

<b>MÓDULO TERMOSOLAR</b>	TRATAMIENTO TÉRMICO: N/A	MATERIAL: ACERO A36, MADERA, ALUMINIO
	ESCALA: 1:10	FORMATO: A3
DISEÑO POR: ANGELO ARMAS EMERSON VITERI	FECHA: 04/10/2021	LÁMINA: 1



## ESCUELA POLITÉCNICA NACIONAL - ESFOT

**DISEÑO 3D MÓDULO  
TERMOSOLAR**

**TRATAMIENTO  
TÉRMICO: N/A**

**MATERIAL :MADERA,  
ALUMINIO, ACERO**

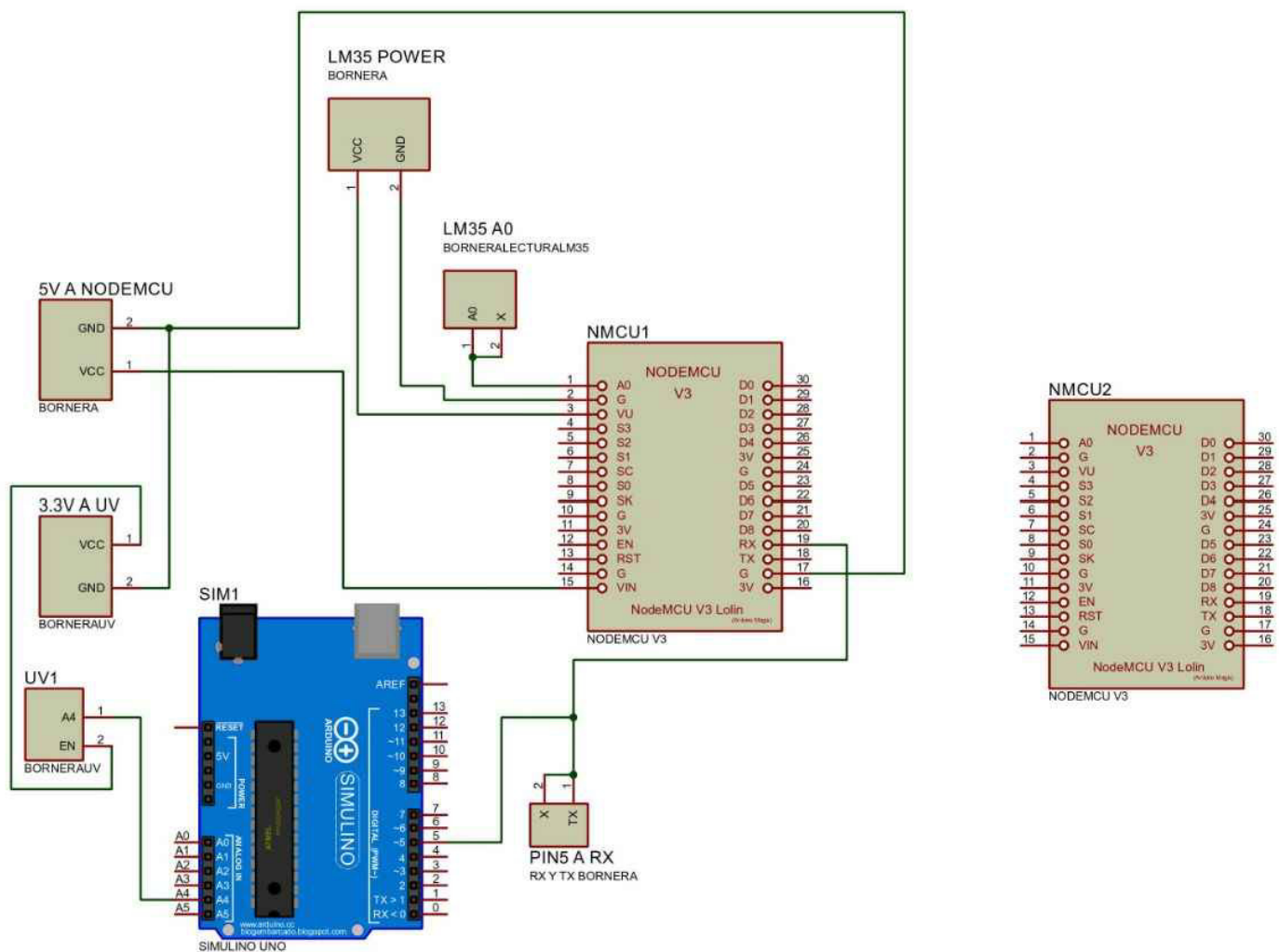
**ESCALA: N/A**

**FORMATO: A4**

**DISEÑADO POR:  
ANGELO ARMAS  
EMERSON VITERI**

**FECHA: 04/10/2021**

**LÁMINA:2**



## ESCUELA POLITÉCNICA NACIONAL - ESFOT

### ESQUEMA ELECTRÓNICO

TRATAMIENTO  
TÉRMICO: N/A

MATERIAL :ELÉCTRICO

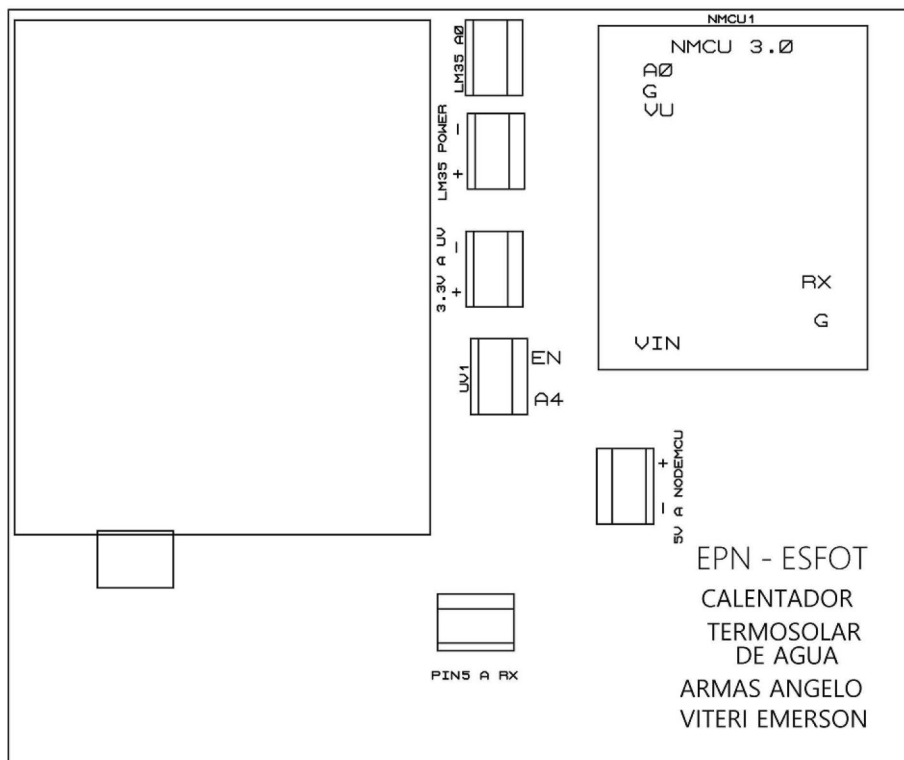
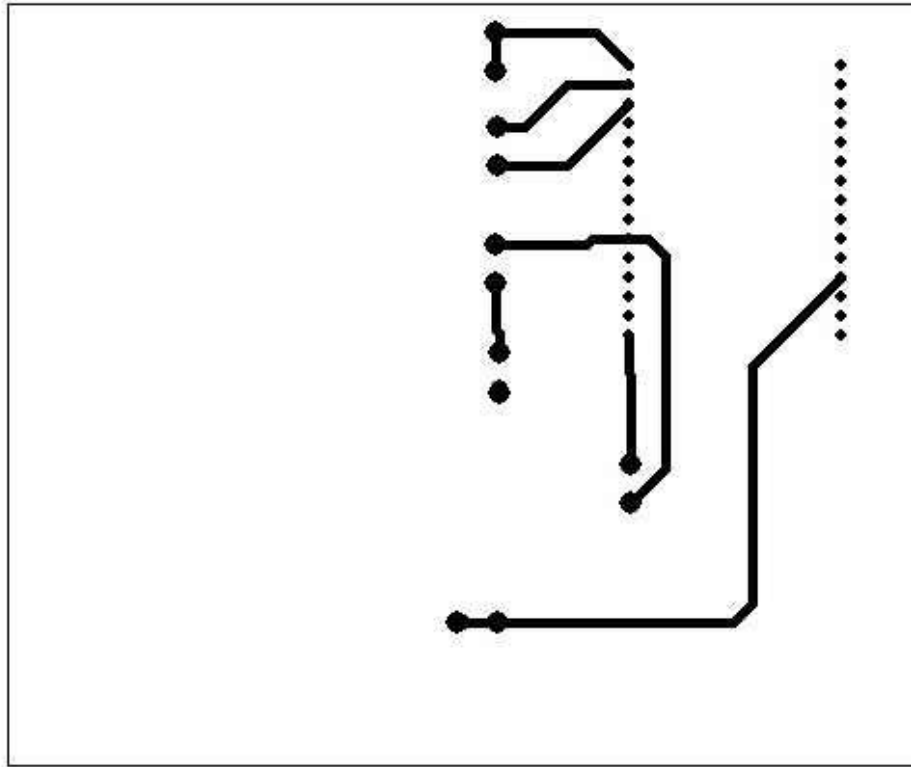
ESCALA: N/A

FORMATO: A4

DISEÑADO POR:  
ANGELO ARMAS  
EMERSON VITERI

FECHA: 04/10/2021

LÁMINA:3



## ESCUELA POLITÉCNICA NACIONAL - ESFOT

<b>VIAS CIRCUITO PCB Y TOP SILK PCB</b>	<b>TRATAMIENTO TÉRMICO: N/A</b>	<b>MATERIAL :ELÉCTRICO</b>
	<b>ESCALA: 1:1</b>	<b>FORMATO: A4</b>
<b>DISEÑADO POR: ANGELO ARMAS EMERSON VITERI</b>	<b>FECHA: 04/10/2021</b>	<b>LÁMINA:4</b>