



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Proyecto De Titulación

Carrera De Tecnologías De La Información

Tema: Diseño, fabricación y evaluación de un calentador de fibra ensamblado con tecnología de impresión 3D y control de temperatura mediante IOT.

Autor: Noboa Pincay, Bryan David

Tutor: Ing. Rodríguez Galán, German Eduardo, Mgtr.

Santo Domingo, 08 de marzo 2024

REPORTE DE VERIFICACIÓN DE CONTENIDO



Plagiarism and AI Content Detection Report

Tesis calefactor Bryan.docx

Scan details

Scan time:
March 10th, 2024 at 16:35 UTC

Total Pages:
72

Total Words:
17816

Plagiarism Detection



Types of plagiarism		Words
Identical	1.4%	244
Minor Changes	0.1%	24
Paraphrased	1.7%	309
Omitted Words	32.4%	5770

AI Content Detection



Text coverage		Words
AI text	11.1%	1426
Human text	88.9%	10620

[Learn more](#)

Alerts: (1)

Cross Language: Same Document Language

Submitted language and cross-language text are the same language. No credits were used.

2/5 Severity



Firma:



Firmado electrónicamente por:
GERMÁN EDUARDO
RODRIGUEZ GALÁN

.....

Ing. Rodríguez Galán, Germán Eduardo

C.C: 0603431685



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN**

CERTIFICACIÓN

Certifico que el trabajo de integración curricular: **“Diseño, fabricación y evaluación de un calentador de fibra ensamblado con tecnología de impresión 3D y control de temperatura mediante IOT.”** fue realizado por el señor **Noboa Pincay, Bryan David**, el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizada en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Santo Domingo de los Tsáchilas, 08 de marzo de 2024

Firma:



**GERMÁN EDUARDO
RODRIGUEZ GALAN**

.....

Ing. Rodríguez Galán, Germán Eduardo, Mgtr

C.C: 0603431685



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN**

RESPONSABILIDAD DE AUTORÍA

Yo, **Noboa Pincay, Bryan David**, con cédula de ciudadanía No. 0803856467, declaro que el contenido, ideas y criterios del trabajo de integración curricular: **“Diseño, fabricación y evaluación de un calentador de fibra ensamblado con tecnología de impresión 3D y control de temperatura mediante IOT”** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Santo Domingo de los Tsáchilas, 08 de marzo de 2024

Firma

.....
Noboa Pincay, Bryan David

C.C: 0803856467



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN**

AUTORIZACIÓN DE PUBLICACIÓN

Yo, **Noboa Pincay, Bryan David**, con cédula de ciudadanía No. 0803856467, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: **“Diseño, fabricación y evaluación de un calentador de fibra ensamblado con tecnología de impresión 3D y control de temperatura mediante IOT”** en el Repositorio institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Santo Domingo de los Tsáchilas, 08 de marzo de 2024

Firma

.....
Noboa Pincay, Bryan David

C.C: 0803856467

Dedicatoria

Este proyecto de titulación va dedicado a mis padres quienes se esforzaron y sacrificaron para que yo logre culminar mis estudios de grado superior y pusieron toda su confianza en mí. A mi pareja Noemi Chumanía, quien ha estado en los momentos buenos y malos, siempre haciéndome tener presente la importancia de tener alguien con quien contar y que con su carisma alegraba los días en los que estudiar se volvía algo agotador. A mi amiga Jhanira Aranda que siempre me apoyó con frases motivadoras para no rendirme. A mis amigos Maylon Cuzco e Ing. Jefferson Guamán, quienes fueron de las personas más cercanas con las que pase en esta etapa y siempre estuvieron para ayudar. De igual manera agradezco al ejercicio físico, que hizo que pudiera desestresarme cada tarde luego de mañanas agobiantes en la Universidad. Y sobre todo este proyecto va dedicado a mi porque en los días que no tenía ánimos para desarrollar alguna actividad, me levantaba y lo hacía.

Bryan David Noboa Pincay

Agradecimiento

Quiero expresar mi profundo agradecimiento a todas las personas que contribuyeron de manera significativa a la realización de este proyecto de titulación en la Universidad de las Fuerzas Armadas (ESPE) sede Santo Domingo.

Primero y principalmente, quiero agradecer a mi tutor, el Ing. Germán Eduardo Rodríguez Galán, Mgtr., por su orientación, apoyo y dedicación a lo largo de todo este proceso. Su experiencia y conocimientos fueron fundamentales para la consecución de este proyecto.

También deseo reconocer y agradecer a todos los docentes de la carrera de Tecnologías de la Información, cuyas enseñanzas y consejos han sido de gran valor en mi formación académica y en el desarrollo de este trabajo.

Agradezco a mi familia y amigos por su comprensión, paciencia y aliento durante este exigente período académico. Su apoyo ha sido mi mayor motivación.

Así mismo agradezco el cambio de mentalidad que tuve para lograr conseguir todas las metas que me proponía.

Finalmente, agradezco a la Universidad de las Fuerzas Armadas (ESPE) sede Santo Domingo por proporcionar el ambiente propicio para la investigación y el aprendizaje, y por fomentar el desarrollo de proyectos que buscan impactar de manera positiva en la comunidad académica y más allá.

A todos ustedes, ¡muchas gracias!

Bryan David Noboa Pincay

Índice de contenido

Dedicatoria	I
Agradecimiento	II
Resumen	1
Abstract	2
I. Introducción.....	3
A. Antecedentes.....	3
B. Problemática	3
C. Justificación	4
D. Marco referencial:.....	5
E. Objetivo general	5
F. Objetivos específicos.....	5
II. Marco Teórico	7
G. Características de los materiales de impresión	7
1) Características de los filamentos PLA:	7
2) Características de los filamentos PET:.....	7
3) Humedad en los filamentos, ¿Qué es la higroscopicidad?:	7
H. Impresión 3D	8
4) Fundamentos de la Impresión 3D:	8
5) Tecnologías de impresión 3D relevantes:	8
6) Tinkercad para diseño 3D:	8
I. Iot y control de dispositivos térmicos.....	9
7) Internet de las Cosas:.....	9
8) Control de dispositivos térmicos usando IoT:.....	9
III. Propuesta Metodológica y Diseño	12
J. Diseño 3D del calefactor de fibras PLA y PET	12

K.	Impresión 3D de las piezas del calefactor	16
L.	Montaje de piezas y componentes	19
M.	Análisis económico de componentes electrónicos utilizados para el control de temperatura y humedad	21
9)	Módulo ESP32:	21
10)	Módulo Relé 5V:.....	21
11)	Resistencia Niquelina:.....	21
12)	Ventilador:.....	21
13)	Módulo Sensor de Temperatura y Humedad (DTH11 y DTH22):	22
N.	Implementación del circuito de control de humedad y temperatura.....	24
O.	Desarrollo y evaluación del sistema de control y visualización de datos.....	25
15)	Etapa de configuración del entorno de trabajo en Blynk IOT:.....	25
16)	Etapa de configuración de sensores DHT11/DTH22 en Blynk IOT:.....	30
17)	Etapa de configuración del control de temperatura con Blynk IOT:	35
18)	Configuración de servidor MQTT y Node-RED:	40
19)	Desarrollo de nodos para configurar el flujo de trabajo en Node-RED: ..	43
20)	Configuración del programa para la gestión de dispositivos IoT mediante MQTT y su integración con Node-RED:	44
21)	Edición de nodos en Node-RED asignando los temas y la dirección IP del servidor MQTT:	46
22)	Vista de la interfaz de usuario (UI) de los nodos empleados en Node-RED: 49	
P.	Pruebas operativas y de integración	50
IV.	Resultados.....	52
V.	Conclusiones y recomendaciones	58
Q.	Conclusiones.....	58
R.	Recomendaciones	59
VI.	Referencias bibliográficas	60

Índice de tablas

TABLA I. ANÁLISIS ECONÓMICO DE LOS COMPONENTES ELECTRÓNICOS UTILIZADOS + PRECIO IMPRESIÓN/HORA DE LAS PARTES QUE COMPONEN EL CALEFACTOR.....	23
---	-----------

Índice de figuras

Fig. 1. Propuesta metodológica para el diseño y fabricación del calefactor de fibras PLA y PET.	12
Fig. 2. Diseño general del calefactor de fibras PLA y PET realizado con el software TinkerCAD.....	12
Fig. 3. Diseño de las caras exteriores del calefactor en forma de “rebanadas de pizza”.....	13
Fig. 4. Diseño de soporte interior para el rollo de fibra.	13
Fig. 5. Diseño del soporte del calefactor.....	14
Fig. 6. Diseño de las partes unidas.....	14
Fig. 7. Diseño de la cubierta del calefactor en 2 partes (base y cubierta).....	15
Fig. 8. Diseño del cilindro hueco para la parte interna del calefactor.....	15
Fig. 9. Piezas para armar las caras externas del calefactor (8 por cada cara).....	16
Fig. 10. Impresión de los soportes para el rollo de filamento PLA y PET.....	16
Fig. 11. Piezas de circunferencia para ampliar el tamaño del calefactor.	17
Fig. 12. Vista previa del acoplamiento de piezas fabricadas para las caras externas y soportes de fibra.	17
Fig. 13. Soportes para el calefactor.....	18
Fig. 14. Impresión de la cubierta del calefactor.	18
Fig. 15. Vista previa del cilindro hueco impreso para dar soporte al rollo fibra.....	19
Fig. 16. Ensamblaje de las caras externas.....	19
Fig. 17. Ensamblaje de los componentes internos y base del calefactor.....	20
Fig. 18. Ensamblaje total del Calefactor Impreso en 3D.	20
Fig. 19. Conexión de sensores, resistencia niquelina y ventiladores.	25
Fig. 20. Creación de un template en Blynk IOT.	26
Fig. 21. Creación de dispositivo en Blynk IOT.....	26
Fig. 22. Nombre de dispositivo en Blynk IOT.....	27
Fig. 23. Pin virtual para la temperatura interior.	27
Fig. 24. Pin virtual para la humedad interior.....	28

Fig. 25. Pines virtuales creados.....	28
Fig. 26. Uso de módulos tipo Gauge para la representación de datos en el Web Dashboard. .	29
Fig. 27. Configuración de los módulos Gauge para visualización de datos.....	29
Fig. 28. Cuatro Gauges para recibir datos de los sensores de humedad/temperatura DHT11/DTH22.	30
Fig. 29. Librerías necesarias para la comunicación con Blynk IOT.	30
Fig. 30. Configuración de conexión y sensores para la comunicación entre Blynk y el módulo ESP32.	31
Fig. 31. Función de envío de datos del sensor de temperatura/humedad al servidor de Blynk.	
32	
Fig. 32. Proceso de Inicialización y ejecución del sistema para monitoreo y envío de datos..	33
Fig. 33. Conexión física de las primeras pruebas.....	33
Fig. 34. Lectura de datos de temperatura y humedad.....	34
Fig. 35. Se reflejan los datos de la temperatura y humedad en el servidor Blynk.	34
Fig. 36. Creación del Pin virtual para el control de temperatura.....	35
Fig. 37. Configuración del slider para el control de temperatura.....	35
Fig. 38. Slider que permite el control de temperatura del calefactor.	36
Fig. 39. Variables y configuración para control de temperatura con resistencia niquelina.	36
Fig. 40. Control de temperatura con resistencia niquelina utilizando sensor DHT11/DHT22.	37
Fig. 41. Control interactivo de temperatura en Blynk.....	37
Fig. 42. Configuración de la función inicial agregando el control de los pines del control de temperatura.....	38
Fig. 43. Control automático de temperatura con ciclo de encendido del calentador.	38
Fig. 44. Control automático de temperatura con ciclo de apagado del calentador.....	39
Fig. 45. Conexión y pruebas de la resistencia niquelina en ESP32 para Blynk.....	39
Fig. 46. Actualización de la lista de paquetes en Ubuntu.....	40
Fig. 47. Instalación de Mosquitto en Ubuntu.	40
Fig. 48. Estado activo del mosquitto.	41

Fig. 49. Instalación de Node.js en Windows.....	41
Fig. 50. Instalación de Node-RED en Windows.	42
Fig. 51. Configuración inicial y puesta en marcha de Node-Red.....	42
Fig. 52. Muestra del panel de control de flujos de trabajo en el software Node-Red.	43
Fig. 53. Configuración de nodos MQTT IN y dashboards de tipo gauge en Node-RED.	43
Fig. 54. Flujo de trabajo con nodos MQTT OUT controlando la niquelina y los ventiladores. 44	
Fig. 55. Configuración del programa para la gestión de dispositivos IoT mediante MQTT. ..	45
Fig. 56. Configuración de clientes WiFi y MQTT en el código desarrollado.....	45
Fig. 57. Configuración de las propiedades del servidor MQTT en Node-RED.....	46
Fig. 58. Configuración de nodos MQTT-In.	47
Fig. 59. Configuración de nodos MQTT-Out.	47
Fig. 60. Configuración de nodo de slider.	48
Fig. 61. Configuración de nodo switch para controlar el encendido y apagado.	48
Fig. 62. Interfaz Gráfica que será controlado por el usuario del calefactor.	49
Fig. 63. Interfaz GUI de Node-RED mostrando datos recibidos desde el ESP32.	49
Fig. 64. Pruebas de integración del componente impreso con el circuito electrónico.	50
Fig. 65. Diseño Alternativo del Calefactor.....	52
Fig. 66. Ensamblaje final de las 37 piezas del calefactor.....	53
Fig. 67. Problemas de cristalización en la fibra PLA durante el proceso de impresión.....	54
Fig. 68. Control de temperatura con margen de error en Node-RED.	55
Fig. 69. Control de temperatura con margen de error en la consola de Arduino IDE.....	55
Fig. 70. Restauración de la fibra PLA cristalizada mediante calefacción.....	56
Fig. 71. Restauración de la fibra PLA cristalizada mediante calefacción.....	56
Fig. 72. Fibra PLA restaurada (Se vuelve maleable).	57

Resumen

Esta propuesta de titulación se enfoca en abordar los desafíos de almacenamiento y procesamiento de materiales clave para la impresión 3D, específicamente la fibra de Ácido Poliláctico (PLA) y la fibra de Polietileno Tereftalato (PET), en los laboratorios de la Universidad de las Fuerzas Armadas - ESPE sede Santo Domingo. Estos materiales son sensibles a las variaciones de temperatura y han experimentado cristalización y fracturas debido a las condiciones inadecuadas de almacenamiento.

Los objetivos del proyecto incluyen diseñar, fabricar y evaluar un calefactor de fibra ensamblado mediante tecnología de impresión 3D, junto con la implementación de un sistema de control de temperatura basado en Internet de las Cosas (IoT). Se realizó un diseño tridimensional detallado utilizando software de diseño asistido por computadora, y los componentes del calefactor se fabricaron mediante tecnologías de impresión 3D. El sistema de calefacción se diseñó específicamente para extraer eficientemente la humedad de 2 tipos de fibras de impresión: PLA y PET, con el fin de mejorar el proceso de impresión.

Durante el desarrollo del proyecto, inicialmente, se exploró el software en línea Blynk IOT para el control de temperatura, pero se encontró que la funcionalidad completa requería una suscripción paga, limitando su uso. En su lugar, se utilizó el protocolo de transporte de telemetría de colas de mensajes (MQTT) junto con el software editor de flujos Node-RED, herramientas más versátiles y adecuadas para el Internet de las Cosas (IoT).

Los resultados confirman la viabilidad del diseño y fabricación del calefactor, así como la exitosa implementación del sistema de control de temperatura basado en MQTT y Node-RED. Sin embargo, se reconoce la necesidad de explorar alternativas web de código abierto en futuras investigaciones, dada la limitación de acceso a ciertas funcionalidades de Blynk IOT debido a su modelo de suscripción paga.

Palabras clave: *Calefactor, PLA, PET, Impresión 3D, MQTT, Node-RED, Blynk IOT.*

Abstract

This degree proposal focuses on addressing the storage and processing challenges of critical materials for 3D printing, specifically Polylactic Acid (PLA) fiber and Polyethylene Terephthalate (PET) fiber, in the laboratories of the Universidad de las Fuerzas Armadas - ESPE Santo Domingo. These materials are sensitive to temperature variations and have experienced crystallization and fractures due to inadequate storage conditions.

The project's objectives include designing, fabricating, and evaluating a fiber heater assembled using 3D printing technology and implementing a temperature control system based on the Internet of Things (IoT). A detailed three-dimensional design was performed using computer-aided design software, and the heater components were fabricated using 3D printing technologies. The heating system was designed to efficiently extract moisture from 2 printing fibers, PLA, and PET, to improve the printing process.

During project development, initially, Blynk IOT online software for temperature control was explored, but it was found that full functionality required a paid subscription, limiting its use. Instead, the message queuing telemetry transport protocol (MQTT) was used along with the Node-RED flow editor software, which are more versatile tools suitable for the Internet of Things (IoT).

The results confirm the feasibility of the heater design and fabrication and the successful implementation of the temperature control system based on MQTT and Node-RED. However, the need to explore open-source web alternatives is recognized in future research, given the limited access to certain functionalities of Blynk IoT due to its paid subscription model.

Keywords: Heater, PLA, PET, 3D printing, MQTT, Node-RED, Blynk IOT.

I. Introducción

A. Antecedentes

En los laboratorios del Departamento de Ciencias de la Computación (DCCO-SS) en la Universidad de las Fuerzas Armadas - ESPE sede Santo Domingo, se ha presentado el desafío crítico de almacenamiento de materiales clave para la impresión 3D, como la fibra de Ácido Poliláctico (PLA) y la fibra de Polietileno Tereftalato (PET). Estos materiales son sensibles a las variaciones de temperatura, y han sufrido cristalización y fracturas en las condiciones actuales de almacenamiento.

Cuando la fibra PLA está cristalizada, restaurar su maleabilidad requiere establecer su temperatura entre 45 y 50 grados Celsius para evitar problemas durante la impresión 3D. De manera similar, la fibra PET cristalizada, más los problemas de dureza y quebrantamiento, dificultan su procesamiento. Calentar el PET a 65-70 grados Celsius lo ablanda, pero es crucial no sobrecalentarlo, ya que podría volverse quebradizo nuevamente.

Por lo tanto, es recomendado iniciar en 35 grados Celsius y ajustar gradualmente la temperatura hasta lograr la flexibilidad deseada para manipular las fibras para el proceso de impresión 3D, recordando siempre su requerimiento funcional de temperatura. En ese contexto, este enfoque integral busca mejorar el manejo de ambos materiales en los laboratorios del DCCO de la ESPE.

B. Problemática

La Universidad de las Fuerzas Armadas - ESPE sede Santo Domingo ha experimentado un aumento significativo en la adopción de tecnologías de impresión 3D, generando una fuerte dependencia de materiales esenciales como las fibras PLA y PET para la producción de piezas tridimensionales. Aunque estos materiales son cruciales para el funcionamiento adecuado de las impresoras 3D, su rendimiento está estrechamente vinculado a condiciones específicas de temperatura durante el proceso de impresión.

La falta de un control riguroso de las condiciones térmicas en el entorno de trabajo de las impresoras 3D ha dado lugar a problemas recurrentes. Entre ellos, se incluyen: deformaciones en las piezas impresas, obstrucción en los extrusores y, en situaciones más críticas, fallas completas en el funcionamiento de las impresoras. Estos problemas, en su

mayoría, se derivan directamente de variaciones no controladas en la temperatura ambiente y, en algunos casos, de la incapacidad para mantener las temperaturas ideales de fusión para el PLA y el PET.

A pesar de la existencia de soluciones convencionales para el control de temperatura en impresoras 3D, estas carecen de la capacidad de monitoreo y ajuste en tiempo real que brinda la tecnología de Internet de las Cosas (IoT). La ausencia de un sistema integral de control de temperatura mediante IoT limita la eficiencia y confiabilidad del proceso de impresión 3D en la Universidad, afectando directamente la calidad de los productos finales y generando una necesidad urgente de una solución más avanzada y adaptable.

En este contexto, la hipótesis que se ha planteado y que aportará al desarrollo de este proyecto es: *“Se espera que el diseño y fabricación de un calentador de fibra ensamblado utilizando tecnología de impresión 3D, junto con el control de temperatura mediante el Internet de las cosas (IoT), resulte en un dispositivo eficiente y preciso en la regulación térmica de fibras PLA y PET, ofreciendo una alternativa innovadora y viable para el proceso de impresión 3D”*.

C. Justificación

En la Universidad de las Fuerzas Armadas - ESPE hay una alta integración de tecnologías de impresión 3D que ha presentado desafíos críticos asociados al control térmico, especialmente en la manipulación de materiales como la fibra PLA y PET. La ausencia de una gestión precisa de las condiciones térmicas ha conducido a problemas operativos y limitaciones en la calidad de las impresiones 3D.

En este contexto, este trabajo tiene como objetivo diseñar y construir un sistema de control térmico basado en Internet de las Cosas (IoT) que se adapte específicamente a las condiciones de los laboratorios del Departamento de Ciencias de la Computación (DCCO-SS). Más allá de sus implicaciones académicas, este proyecto busca ofrecer una solución práctica y tecnológica que no solo optimice la eficiencia del proceso de impresión y reduzca las fallas operativas, sino que también eleve la calidad de las piezas fabricadas.

La implementación exitosa de este calefactor, que integre un sistema de control térmico basado en IoT, permitirá restablecer el uso del material que se ha descartado por causa de la humedad.

D. Marco referencial:

En el ámbito de la impresión 3D, se resalta la contribución significativa de Ibarra Villacreses y Ortega Pio [15], quienes han diseñado un sistema embobinador y secador de filamentos. Este innovador proyecto aborda de manera efectiva el desafío crítico de la humedad en los filamentos, especialmente en materiales como PLA, Acrilonitrilo-butadieno-estireno (ABS) y PLA con recubrimiento de carbono. Las pruebas exhaustivas realizadas con diversas variedades de filamentos validan la eficacia del secador al eliminar con éxito las partículas de agua. La capacidad de este sistema para gestionar la humedad de manera eficiente destaca su función esencial como solución práctica, elevando la calidad de las impresiones 3D y previniendo posibles problemas asociados con la absorción de humedad. Este trabajo representa un hito importante en la investigación, consolidando la importancia del control de la humedad para optimizar los resultados en la impresión 3D.

Los filamentos utilizados en la impresión 3D mediante modelado por deposición fundida (FDM) son higroscópicos, lo que significa que absorben la humedad del aire. Esta propiedad puede causar daños en los filamentos, ya que, al ser polímeros plásticos, están compuestos por cadenas de moléculas enlazadas que se ven afectadas por la humedad [9]. Una solución para revertir el daño causado por la humedad es secar los filamentos, ya sea utilizando un secador de filamentos específico o un horno doméstico a una temperatura adecuada.

E. Objetivo general

Diseñar, fabricar y evaluar un calentador de fibra ensamblado utilizando tecnología de impresión 3D y control de temperatura mediante IoT.

F. Objetivos específicos

1. Realizar el diseño tridimensional detallado del calentador de fibra, empleando software de diseño asistido por computadora.
2. Fabricar los componentes del calentador utilizando tecnologías de impresión 3D.

3. Diseñar y construir de manera integral el sistema de calefacción destinado a la extracción eficiente de humedad en fibras de impresión PLA y PET, optimizando la eficacia del proceso.

II. Marco Teórico

G. Características de los materiales de impresión

1) Características de los filamentos PLA:

El Ácido Poliláctico (PLA) destaca como un material altamente prometedor debido a su capacidad para ser producido en grandes cantidades sin perder sus cualidades biodegradables y biocompatibles. Se trata de un poliéster alifático de origen biológico derivado del ácido láctico, el cual puede obtenerse a partir de diversas fuentes naturales como la celulosa, el almidón, el maíz, los desechos de pescado y de cocina. En cuanto a sus propiedades térmicas, el PLA tiene un rango de temperatura de obtención y fusión que oscila entre los 190 y 220 grados Celsius [2].

2) Características de los filamentos PET:

El polietileno tereftalato (PET) es un polímero termoplástico derivado del petróleo, utilizado ampliamente por su suavidad, ligereza y transparencia. Aunque resistente a la degradación biológica debido a sus enlaces C-C, su baja área superficial y alto peso molecular lo hacen no biodegradable. Puede presentarse en forma amorfa, transparente, o semicristalina, opaca, según el proceso de enfriamiento durante su síntesis [10].

3) Humedad en los filamentos, ¿Qué es la higroscopicidad?:

Se aborda el fenómeno de la higroscopicidad en materiales poliméricos, destacando su impacto en la absorción de humedad [14]. Los polímeros se clasifican en dos grupos según su higroscopicidad: higroscópicos y no higroscópicos. Los polímeros higroscópicos, como el ácido poliláctico (PLA) y el tereftalato de polietileno (PET), presentan un desequilibrio en la carga de sus moléculas, lo que les permite formar enlaces de hidrógeno con las moléculas de agua, resultando en la penetración de la humedad en su estructura. Por otro lado, los polímeros no higroscópicos, como el polietileno (PE) y el polipropileno (PP), son considerados apolares y no absorben agua en su estructura, limitándose a la superficie de los materiales. Esta base conceptual proporciona una comprensión fundamental de cómo la higroscopicidad afecta la interacción de los polímeros con la humedad ambiental, lo que es crucial para el almacenamiento y procesamiento adecuado de materiales utilizados en impresión 3D, como el PLA y el PET.

H. Impresión 3D

4) Fundamentos de la Impresión 3D:

La impresión 3D, una técnica revolucionaria en la fabricación ha evolucionado desde su comercialización en 1980 para abarcar diversos campos industriales, como medicina, ingeniería aeroespacial, automovilismo e incluso la industria alimentaria [5]. Este proceso, que construye objetos capa por capa, inicia con el diseño en software, luego la configuración para impresión y finalmente la producción en el hardware de la impresora. Aunque, de forma simple, se deben considerar cuidadosamente las limitaciones y elecciones de materiales en cada etapa para garantizar el éxito del proyecto y la calidad del producto final.

5) Tecnologías de impresión 3D relevantes:

Las tecnologías de impresión 3D han experimentado un rápido desarrollo en las últimas décadas, lo que ha llevado a la diversificación de los métodos y procesos de impresión. Algunas de las tecnologías de impresión 3D más relevantes incluyen la estereolitografía (SLA), la fusión por deposición de material (FDM), la sinterización selectiva por láser (SLS), la litografía de proyección digital (DLP), entre otras [13].

Cada una de estas tecnologías tiene sus propias características, ventajas y limitaciones, lo que las hace adecuadas para diferentes aplicaciones y sectores industriales. Desde la fabricación rápida de prototipos hasta la producción de piezas finales, las tecnologías de impresión 3D están transformando la forma en que se diseñan y fabrican los objetos en diversos campos.

6) Tinkercad para diseño 3D:

Tinkercad es una plataforma en línea de modelado 3D que permite a los usuarios crear diseños digitales de manera intuitiva y fácil de usar, sin necesidad de tener experiencia previa en diseño asistido por computadora (CAD, por sus siglas en inglés). Fue desarrollada por Autodesk [1], una empresa líder en software de diseño y modelado.

Una de las características destacadas de Tinkercad es su facilidad de uso, lo que lo convierte en una herramienta accesible para estudiantes, entusiastas de las manualidades, y profesionales que buscan prototipar ideas rápidamente. Además, Tinkercad es compatible con la impresión 3D, lo que significa que los diseños creados en la plataforma pueden ser exportados y fabricados físicamente en una impresora 3D.

I. Iot y control de dispositivos térmicos

7) Internet de las Cosas:

El Internet de las Cosas (IoT) ha emergido como un concepto trascendental, impulsando la evolución tecnológica en diversas esferas [4]. Este aborda la influencia de IoT en sectores clave como salud, transporte, industria y energía, anticipando mejoras en procesos, reducción de costos y tiempos. El objetivo es responder interrogantes sobre la naturaleza y beneficios de IoT, así como superar barreras actuales para su desarrollo. Simultáneamente en [12] se ha proyectado la conexión de aproximadamente 28 mil millones de dispositivos para 2021, transformando la interacción tecnológica y revolucionando aplicaciones profesionales. Este panorama refleja una era digital en constante evolución, donde IoT redefine la experiencia cotidiana y profesional.

8) Control de dispositivos térmicos usando IoT:

a) Módulo microcontrolador ESP32:

El chip ESP32 ha ganado reconocimiento como una solución fundamental para aplicaciones de Internet de las cosas (IoT). Se destaca la versatilidad del módulo al implementar estaciones meteorológicas en tiempo real mediante el protocolo MQTT [18], logrando una comunicación eficiente entre dispositivos y servidores. Se presenta un estudio sobre la automatización de la medición de humedad, temperatura y pH del suelo en cultivos de espinacas, utilizando el ESP32 como microcontrolador en soluciones agrícolas [7]. Se detallan las potentes características del ESP32, incluyendo su procesador dual, conectividad Wi-Fi/Bluetooth integrada y eficiencia energética, respaldando su preferencia en dispositivos IoT multifuncionales [8]. En el ámbito de la IoT, el ESP32, desarrollado por Espressif Systems, destaca como un componente esencial para soluciones eficientes y versátiles.

b) Sensores de temperatura y humedad:

El sensor DHT11/DTH22 [6], utilizado en el proyecto, destaca por su salida de señal digital calibrada, lo que garantiza una alta confiabilidad y estabilidad a largo plazo. Integrando componentes de medición de humedad de tipo resistivo y de temperatura, este sensor se puede conectar a un Microcontrolador de 8 bits de alto rendimiento, ofreciendo una calidad excepcional y una respuesta rápida. Calibrado en laboratorio para una precisión extrema, los coeficientes de calibración se almacenan como programas en la memoria, utilizados por el proceso de detección de señales internas del sensor. Con una interfaz serial de un solo cable, su

pequeño tamaño y bajo consumo de energía, el DHT11/DHT22 se destaca como una opción óptima para diversas aplicaciones, brindando una transmisión de señal de hasta 20 metros y una integración rápida y sencilla en sistemas más amplios.

c) Sistema de calefacción usando resistencias térmicas:

En el informe presentado por los Bachilleres Roger Bernin Huilca Achahui y Romulo Gaspar Huilca Achahui [17], se ha detallado el funcionamiento de un sistema de calefacción y los sistemas de control de una máquina. El crisol de la máquina requiere una calefacción eléctrica mediante una niquelina colocada en su base, similar a un reverbero eléctrico, para alcanzar una temperatura de hasta 150°C, aunque el controlador de temperatura puede alcanzar hasta 400°C. La instalación eléctrica recomendable es de 110V, y la niquelina es un repuesto común disponible en ferreterías. Para el control neumático y térmico, se necesitan un regulador de presión con manómetro y válvula de desfogue, y un controlador de temperatura digital para la seguridad y precisión en la lectura de la temperatura. Se destaca la importancia del mantenimiento preventivo, con partes y accesorios fácilmente desmontables para realizar reparaciones o limpieza, evitando soldaduras y remaches. Se enfatiza la atención especial que se debe prestar a componentes críticos como el regulador de presión, el controlador de temperatura y la niquelina, siguiendo las instrucciones del manual de operación para su cuidado adecuado.

d) Aplicación Blynk IOT para el control de dispositivos IoT:

Karuppusamy ha propuesto un sistema de monitoreo y control basado en IoT para electrodomésticos, abordando el alto consumo energético y riesgos asociados [16]. Este sistema, compatible con diversos dispositivos en entornos domésticos e industriales, utiliza smartphones para control y monitoreo remoto mediante conexiones inalámbricas. Al integrar plataformas como Blynk y seleccionar hardware IoT, como Arduino o Raspberry Pi, se logra una solución eficiente y flexible para la gestión de dispositivos eléctricos, mejorando la eficiencia energética y la confiabilidad operativa.

e) Aplicación Node Red para el control de dispositivos IoT:

Node-RED es una plataforma de desarrollo que está basada en JavaScript y fue creada por expertos de IBM, diseñada para facilitar la conexión de dispositivos dentro de un sistema de Internet de las cosas (IoT). Permite conectar hardware y software para la creación de varios

flujos de datos desde los sensores hasta la nube de manera rápida, eficiente y controlada [11]. Con Node-RED, los usuarios pueden construir, de manera fácil, interfaces amigables sin la necesidad de desarrollar complejas páginas web. Además, ofrece una forma sencilla de procesamiento de datos y enviarlos a sistemas más avanzados, como bases de datos SQL o servicios en la nube, lo que lo convierte en una herramienta versátil y poderosa para el desarrollo de proyectos IoT.

f) Protocolo de comunicación MQTT:

El protocolo de comunicación MQTT (Message Queuing Telemetry Transport) se destaca como una herramienta vital para la interconexión de dispositivos inteligentes. MQTT ha sido objeto de investigación y análisis durante las últimas dos décadas [3], mostrando un crecimiento significativo en comparación con otros protocolos en el ámbito de la comunicación máquina a máquina (M2M). Este protocolo es utilizado en entornos IoT para facilitar la comunicación entre dispositivos distribuidos en diversas ubicaciones entre los que destacan, hogares, hospitales, fábricas y vehículos. Además, se conoce la importancia de MQTT en la búsqueda de soluciones para lograr una comunicación segura y bidireccional entre dispositivos interconectados, como sensores, actuadores, microcontroladores y electrodomésticos inteligentes, y los servicios en la nube correspondientes.

III. Propuesta Metodológica y Diseño

Se propone la siguiente metodología que abarca todo el proceso, desde el diseño hasta la construcción del calefactor de fibras PLA y PET, así como la implementación del circuito de control de los dispositivos y sensores IoT y las pruebas funcionales de todo el sistema. Esto se ilustra en la Fig. 1.

Calefactor de fibra PLA y PET usando tecnologías de impresión 3D

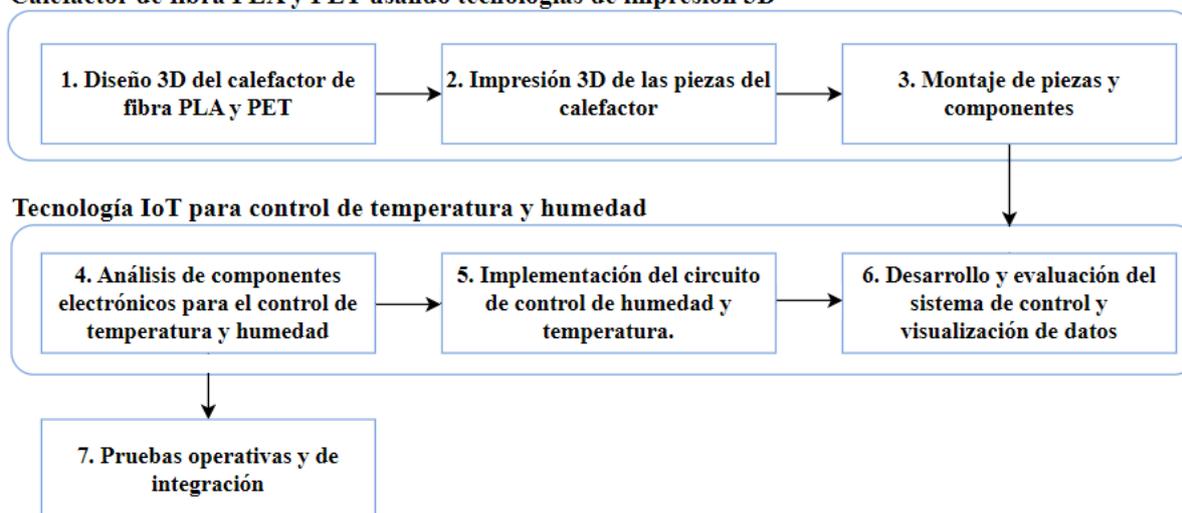


Fig. 1. Propuesta metodológica para el diseño y fabricación del calefactor de fibras PLA y PET.

J. Diseño 3D del calefactor de fibras PLA y PET

Se diseñó un sistema completamente desarmable para optimizar el ensamblaje, considerando las limitaciones del área de impresión disponible en la impresora del DCCO, la cual no permitía la impresión de piezas mayores a 30 x 30 cm. Esto garantizó la viabilidad del proceso de fabricación y ensamblaje dentro de las restricciones de espacio de la impresora. Esto se representa en la Fig. 2.

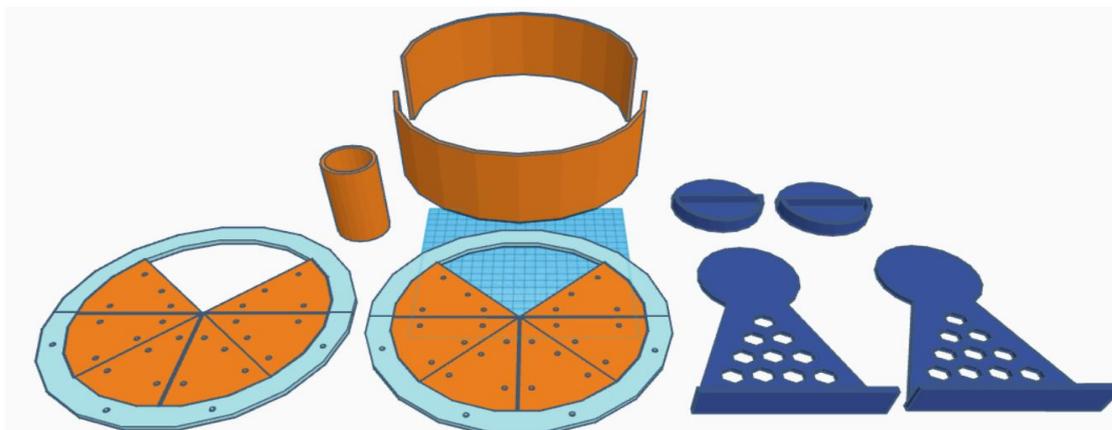


Fig. 2. Diseño general del calefactor de fibras PLA y PET realizado con el software TinkerCAD.

Se modelaron las caras externas del sistema de calefacción tomando como referencia una “pizza de 8 rebanadas”, de un tamaño que permita que un rollo común de fibra (8cm de ancho) encaje perfectamente. Este diseño se compuso de 8 piezas que se acoplaron usando tornillos de 1/8 de grosor por 1/2 pulgada de largo, tal como se muestra en la Fig. 3.

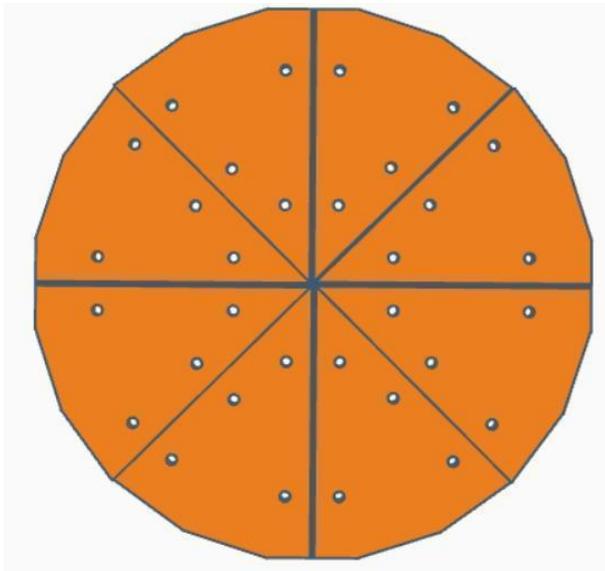


Fig. 3. Diseño de las caras exteriores del calefactor en forma de “rebanadas de pizza”.

Se modeló la parte interna del calefactor, cuyo diseño permitió sostener el rollo de fibra en el interior, como se muestra en la Fig. 4. Este diseño además permitió acoplar todas las partes de las caras exteriores del calefactor.

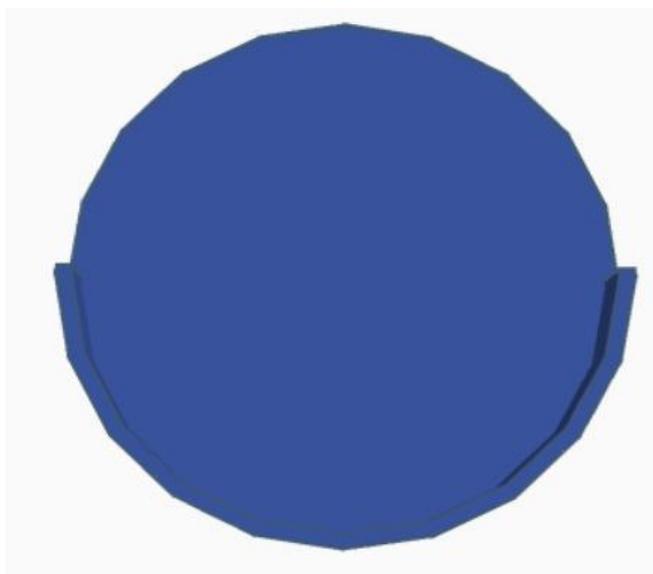


Fig. 4. Diseño de soporte interior para el rollo de fibra.

Se diseñaron los soportes que fueron la base para sostener todo el calefactor, como se muestra en la Fig. 5.

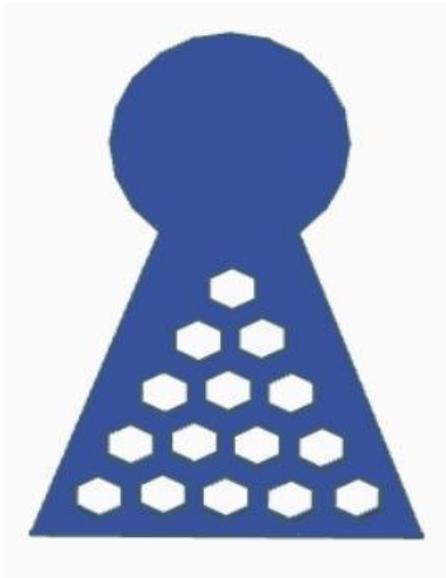


Fig. 5. Diseño del soporte del calefactor.

En la Figura 6 se muestra una vista previa del diseño en 3D con las partes acopladas (caras externas, soporte interno para el rollo de fibra, soporte externo).

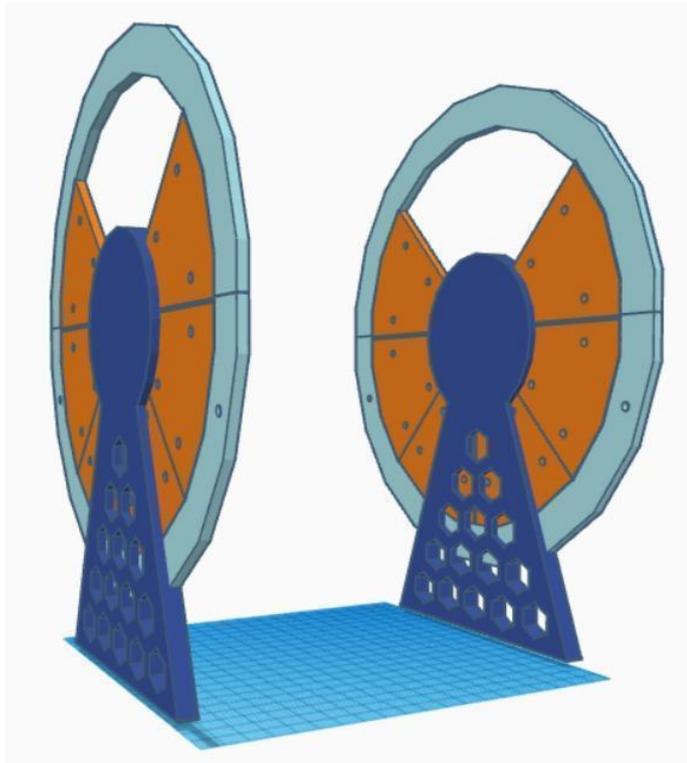


Fig. 6. Diseño de las partes unidas.

Por otro lado, se diseñaron las cubiertas del calefactor, como se observa en la Fig. 7, tomando como referencia una base (parte inferior) y una tapa (parte superior).

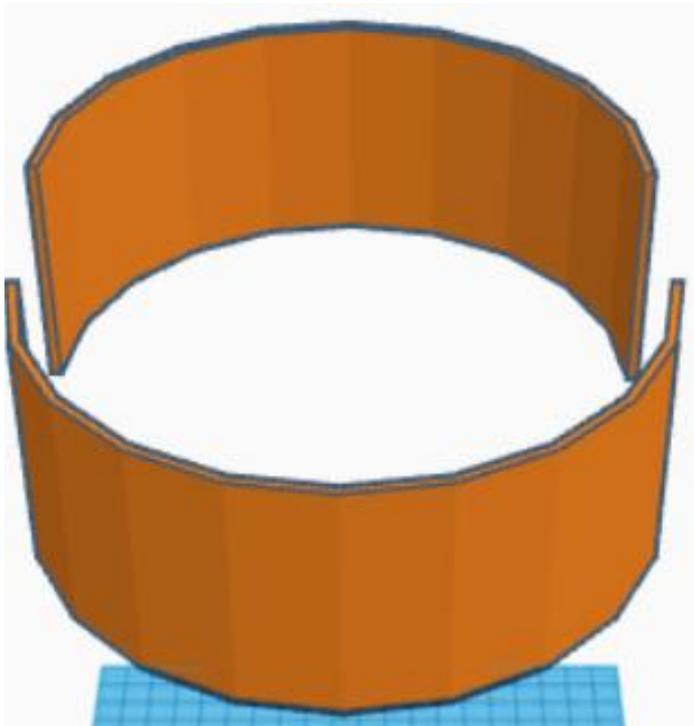


Fig. 7. Diseño de la cubierta del calefactor en 2 partes (base y cubierta).

Además, se diseñó un cilindro hueco para colocarlo en la parte interna del calefactor, como se muestra en la Fig. 8.

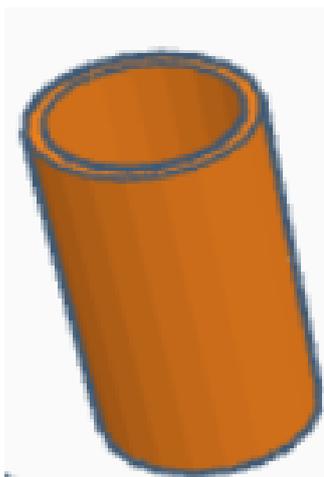


Fig. 8. Diseño del cilindro hueco para la parte interna del calefactor.

K. Impresión 3D de las piezas del calefactor

En total, se imprimieron 16 piezas individuales, que conformaron las 2 caras exteriores del calefactor (8 piezas por cara), como se muestra en la Fig. 9.

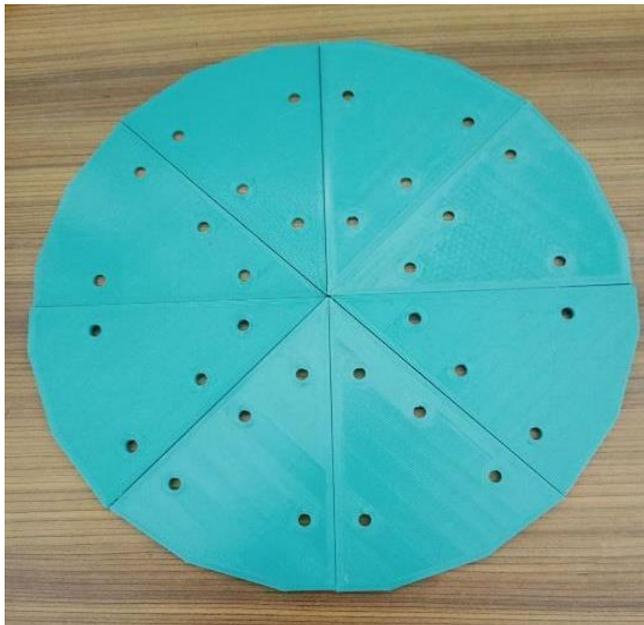


Fig. 9. Piezas para armar las caras externas del calefactor (8 por cada cara).

En la Fig. 10 se muestra la impresión de los dos soportes destinados a sostener el rollo de filamento de fibra PLA y PET. Estos soportes se instalaron en el interior de cada una de las caras impresas que se observan en la Fig. 9.



Fig. 10. Impresión de los soportes para el rollo de filamento PLA y PET.

La Fig. 11 presenta el diseño de la circunferencia que se imprimió y se acopló a las caras mostradas en la Fig. 9, con el fin de aumentar ligeramente el tamaño del calefactor.

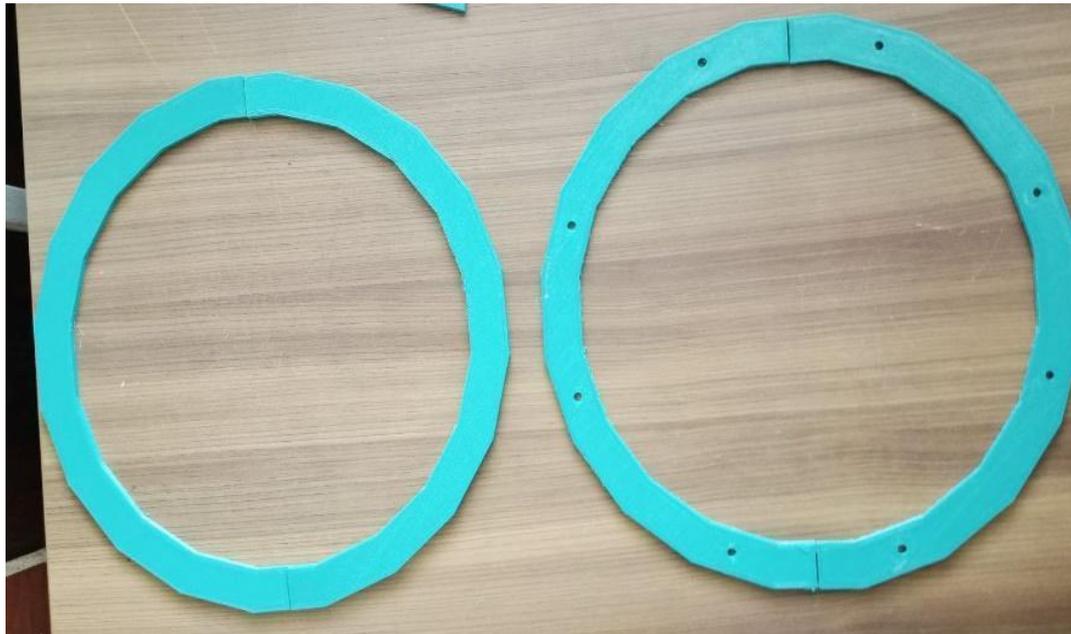


Fig. 11. Piezas de circunferencia para ampliar el tamaño del calefactor.

En la Fig. 12 se verifica el acople preciso de las piezas impresas en la Fig. 11 con las partes fabricadas en la Fig. 9.

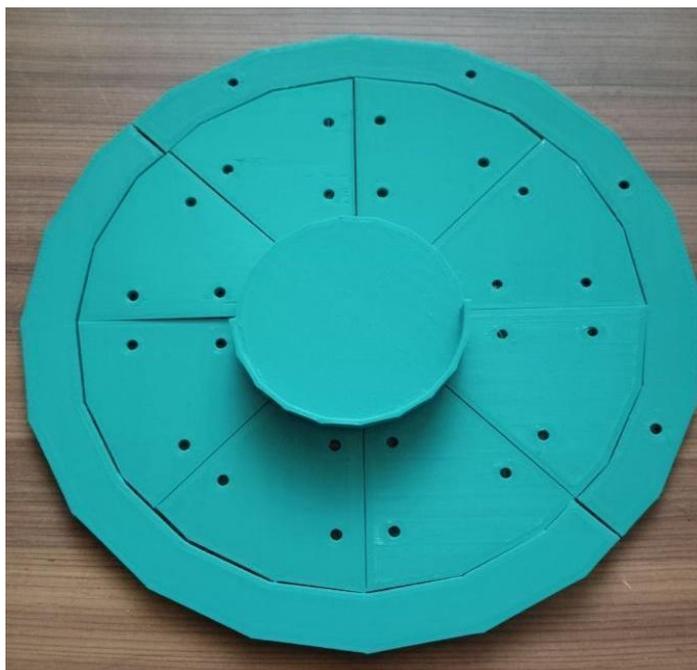


Fig. 12. Vista previa del acoplamiento de piezas fabricadas para las caras externas y soportes de fibra.

Se imprimieron dos piezas que sirvieron de soporte para todo el calefactor, se instalaron en cada cara externa de la Fig. 9. En la Fig. 13 se muestra la vista general de los soportes terminados, se propuso un diseño hueco para ahorrar el uso de fibra PLA.



Fig. 13. Soportes para el calefactor.

Además, se imprimió la cubierta del calefactor, en dos partes, como se muestra en la Fig. 14.



Fig. 14. Impresión de la cubierta del calefactor.

Se imprimió el cilindro hueco diseñado previamente para sostener el rollo de filamento de fibra PLA, asegurando medidas precisas para adaptarse perfectamente a su circunferencia, tal como se muestra en la Fig. 15.



Fig. 15. Vista previa del cilindro hueco impreso para dar soporte al rollo fibra.

L. Montaje de piezas y componentes

Se ensamblaron todas las piezas impresas para construir el calefactor. Para lograrlo, se imprimieron piezas pequeñas diseñadas para el acoplamiento de todas las partes, las cuales se unieron con tornillos para asegurar el acoplamiento de todas las partes. Esto se ilustra en la Fig. 16.



Fig. 16. Ensamblaje de las caras externas.

Se ensamblaron los soportes internos junto con las caras externas, los soportes externos que actuaron como base del calefactor y la cubierta externa, tal como se observa en la Fig. 17.

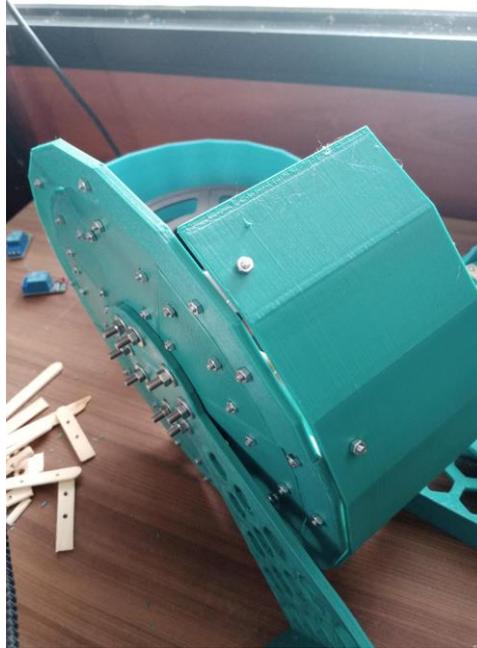


Fig. 17. Ensamblaje de los componentes internos y base del calefactor.

En la Fig. 18 se muestra el resultado final del ensamblaje de las piezas impresas para construir el calefactor.



Fig. 18. Ensamblaje total del Calefactor Impreso en 3D.

M. Análisis económico de componentes electrónicos utilizados para el control de temperatura y humedad

9) Módulo ESP32:

El Módulo ESP32 es un microcontrolador Wifi-Bluetooth que ofrece conectividad de red inalámbrica y capacidades de comunicación Bluetooth. Este componente fue esencial para la comunicación del sistema de control con otros dispositivos y para el acceso a Internet. Con un precio de \$12.00 y una unidad requerida para el proyecto, el Módulo ESP32 proporcionó una solución eficiente y económica para la integración de conectividad inalámbrica en el sistema.

10) Módulo Relé 5V:

Los Módulos Relé de 5V son interruptores electromagnéticos utilizados para controlar dispositivos de alto voltaje o corriente con señales de bajo voltaje. Con un precio de \$5.00 por unidad y dos unidades necesarias para el proyecto, estos relés fueron fundamentales para activar o desactivar los sistemas de calefacción, ventilación, y otros dispositivos eléctricos dentro del sistema de control de temperatura y humedad.

11) Resistencia Niquelina:

La resistencia niquelina es un componente resistivo que genera calor cuando se le aplica una corriente eléctrica, utilizada en aplicaciones de calefacción como el extrusor de una impresora 3D. Con un precio de \$5.50 y una unidad requerida para el proyecto, esta resistencia proporcionó una fuente confiable de calor para el control preciso de la temperatura dentro del sistema.

12) Ventilador:

Los ventiladores de 12V, con dimensiones de 4x4x1cm, son dispositivos de refrigeración que generan flujo de aire para enfriar componentes electrónicos. Con un precio de \$5.00 por unidad y dos unidades necesarias para el proyecto, estos ventiladores fueron fundamentales para proporcionar ventilación y recirculación de aire caliente dentro del sistema de control de temperatura y humedad, garantizando un funcionamiento eficiente y una temperatura estable.

13) Módulo Sensor de Temperatura y Humedad (DTH11 y DTH22):

Los Módulos Sensores de Temperatura y Humedad (DTH11 y DTH22) contienen sensores precisos que permiten medir y monitorear las condiciones ambientales. Con precios de \$4.50 y \$7.00 respectivamente, y dos unidades de cada tipo necesarias para el proyecto, estos sensores proporcionaron lecturas precisas de temperatura y humedad, esenciales para el control y regulación del ambiente dentro y fuera del sistema de control.

El análisis económico de los componentes electrónicos utilizados más el costo/tiempo de impresión de las piezas para el control de temperatura y humedad se los observa en la Tabla 1.

TABLA I.
ANÁLISIS ECONÓMICO DE LOS COMPONENTES ELECTRÓNICOS UTILIZADOS + PRECIO
IMPRESIÓN/HORA DE LAS PARTES QUE COMPONEN EL CALEFACTOR.

Componente	Tipo	Descripción	Precio (USD)	Cantidad
Módulo ESP32	Electrónico	Microcontrolador Wifi- Bluetooth ESP32 30 Pines Devkit	\$12.00	1
Módulo Relé 5V	Electrónico	Interruptor electromagnético	\$5.00	2
Resistencia níquelina	Resistivo	12V 40W Para extrusor de impresora 3D	\$5.50	1
Ventilador	Electrónico	12V 4X4X1cm 4010 3D	\$5.00	2
Modulo Sensor de temperatura y Humedad	Electrónico	DTH11 con cable Arduino	\$4.50	2
Modulo Sensor de temperatura y Humedad	Electrónico	DTH22 con cable Arduino	\$7.00	2
Precio x hora de impresión 3D	Impresora 3D	Tiempo requerido para imprimir todas las partes y piezas del calefactor	\$0.75 /hora	111 horas
Precio rollo de impresión PLA	Impresora 3D	Rollo de fibra PLA requerido para imprimir las piezas del calefactor	\$22.0	1
Sub Total	-	-	\$39.00	10
Total	-	-	\$89.50	-

Tomando como referencia que en el mercado el precio por hora para la impresión de partes y piezas en 3D es aproximadamente de 0.75 centavos, se considera como un ahorro por haber utilizado la impresora 3D del Departamento de Ciencias de la Computación. Sin embargo, si se toma en consideración ese precio, la factibilidad económica se puede reducir cambiando la configuración de los parámetros de velocidad de impresión, así el tiempo requerido se podría reducir a la mitad.

N. Implementación del circuito de control de humedad y temperatura

La conexión del circuito para controlar la temperatura y humedad del calefactor se implementó de forma meticulosa utilizando el microcontrolador ESP32 WROOM. Este dispositivo fue seleccionado por su capacidad para gestionar múltiples entradas y salidas, lo que lo hizo ideal para este proyecto de control de dispositivos IoT. Se conectaron dos sensores diferentes al microcontrolador (DHT22): uno diseñado para medir la temperatura exterior del calefactor y otro para medir la temperatura interior.

Para estas conexiones, se asignaron pines específicos en el microcontrolador. El sensor de temperatura exterior se conectó al pin D5, mientras que el sensor de temperatura interior se conectó al pin D15. Además de estas conexiones, se establecieron las conexiones necesarias para el suministro de energía y la referencia a tierra (GND y 3V3) para ambos sensores, garantizando así un funcionamiento estable y confiable del sistema.

Cada detalle de la conexión se llevó a cabo con precisión para asegurar un funcionamiento óptimo del circuito. Se prestó especial atención a la calidad de las conexiones y se realizaron pruebas exhaustivas para verificar la integridad de la señal y la precisión de las mediciones. Este enfoque meticoloso en la conexión del circuito garantizó un control preciso de la temperatura y la humedad en el calefactor, contribuyendo al éxito general del proyecto.

Para controlar el voltaje de los ventiladores y la resistencia níquelina, se implementaron módulos relé de 5V, asignando uno específico para cada dispositivo. Esto implicó la utilización de tres módulos relé en total: uno para la resistencia níquelina y uno para cada ventilador del calefactor.

Cada módulo relé se conectó a un pin específico en el microcontrolador ESP32 para activar o desactivar su respectivo dispositivo. El módulo relé del calefactor se conectó en el pin D12, mientras que el módulo relé de los ventiladores se conectó en el pin D14. Además, se establecieron las conexiones necesarias para el suministro de energía de los módulos relé. El lado positivo de cada módulo relé se conectó al positivo de una fuente de alimentación de 12V, mientras que el lado negativo de los módulos relé se conectó al componente electrónico correspondiente: la resistencia níquelina o los ventiladores. Esta configuración garantizó un control eficaz del voltaje de cada dispositivo, permitiendo su activación y desactivación según las necesidades del sistema (Fig. 19).

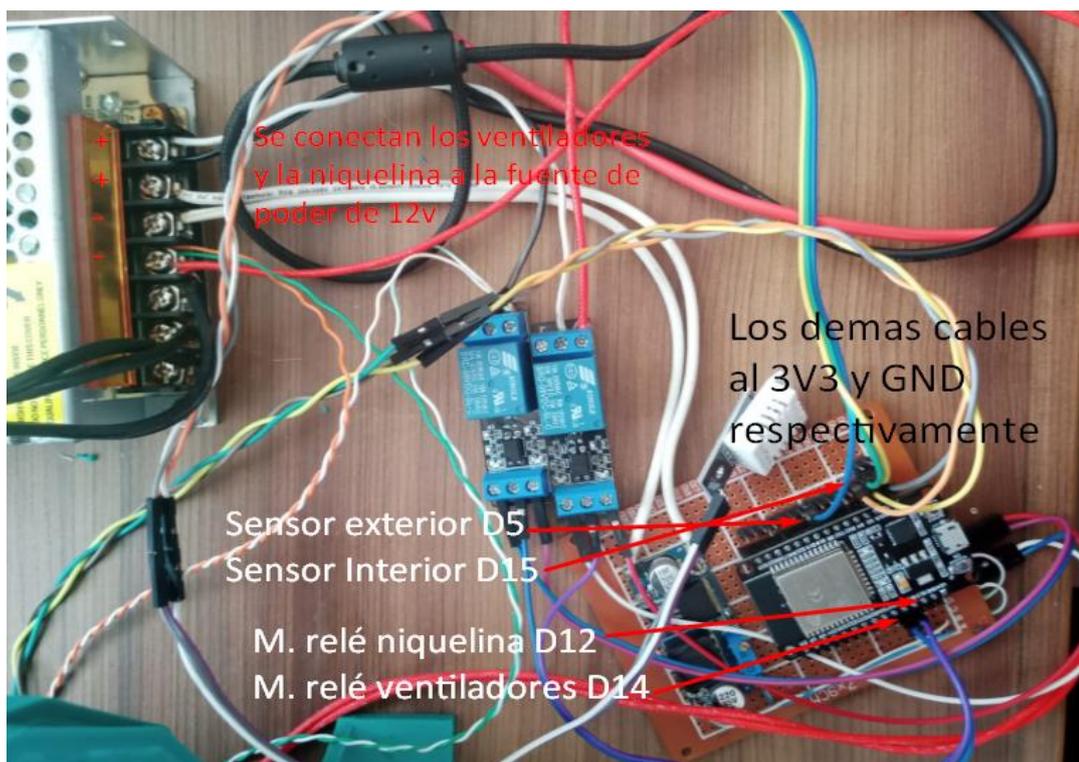


Fig. 19. Conexión de sensores, resistencia níquelina y ventiladores.

O. Desarrollo y evaluación del sistema de control y visualización de datos

15) Etapa de configuración del entorno de trabajo en Blynk IOT:

En la consola de Blynk se creó una nueva plantilla denominada "TESIS", donde se especificó el hardware a utilizar, que fue el módulo ESP32 con una conexión de tipo WiFi. Este proceso se muestra en la Fig. 20.

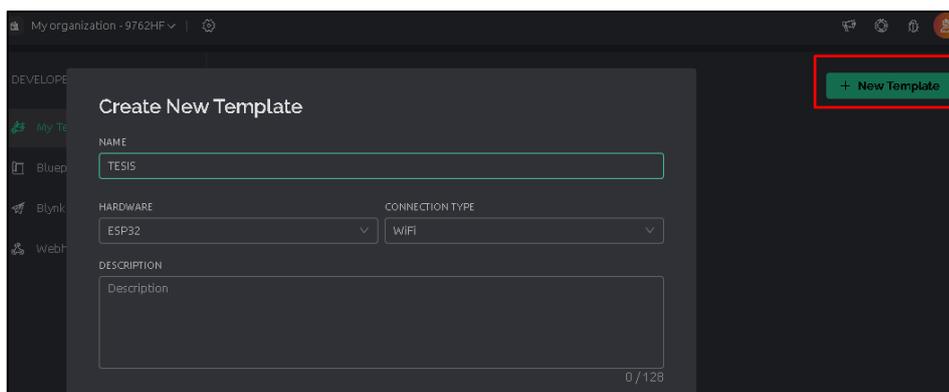


Fig. 20. Creación de un template en Blynk IOT.

Una vez creada la plantilla, se creó un nuevo dispositivo basado en la plantilla previamente configurada, tal como se observa en la Fig. 21.

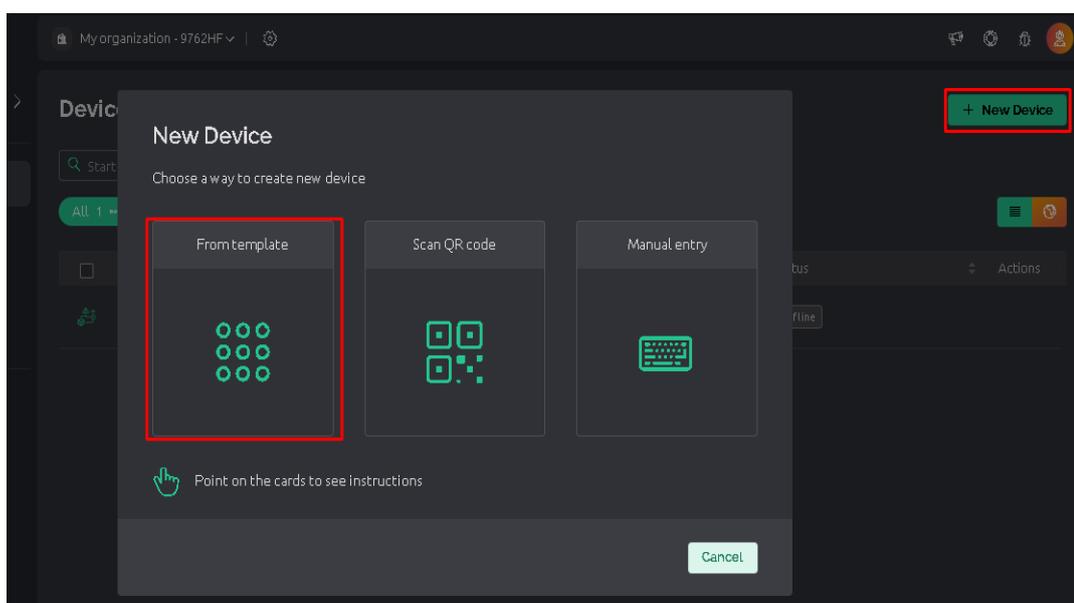
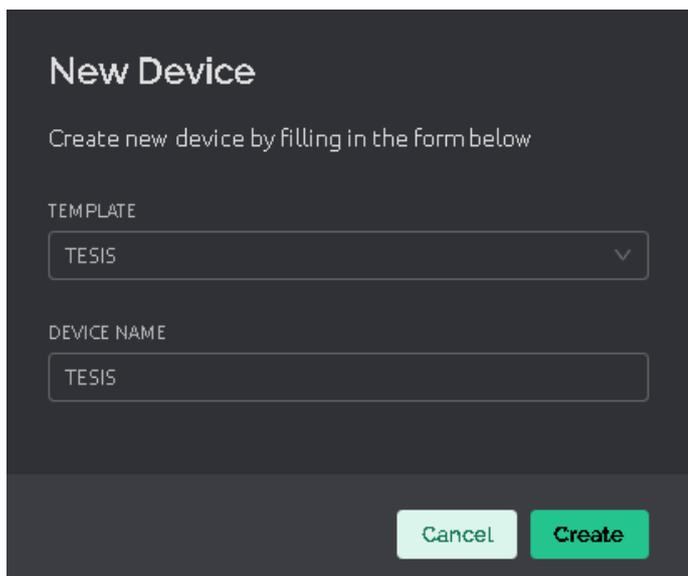


Fig. 21. Creación de dispositivo en Blynk IOT.

En la Fig. 22, se seleccionó la plantilla TESIS previamente creada y se estableció el nombre del dispositivo a crear.



New Device

Create new device by filling in the form below

TEMPLATE

TESIS

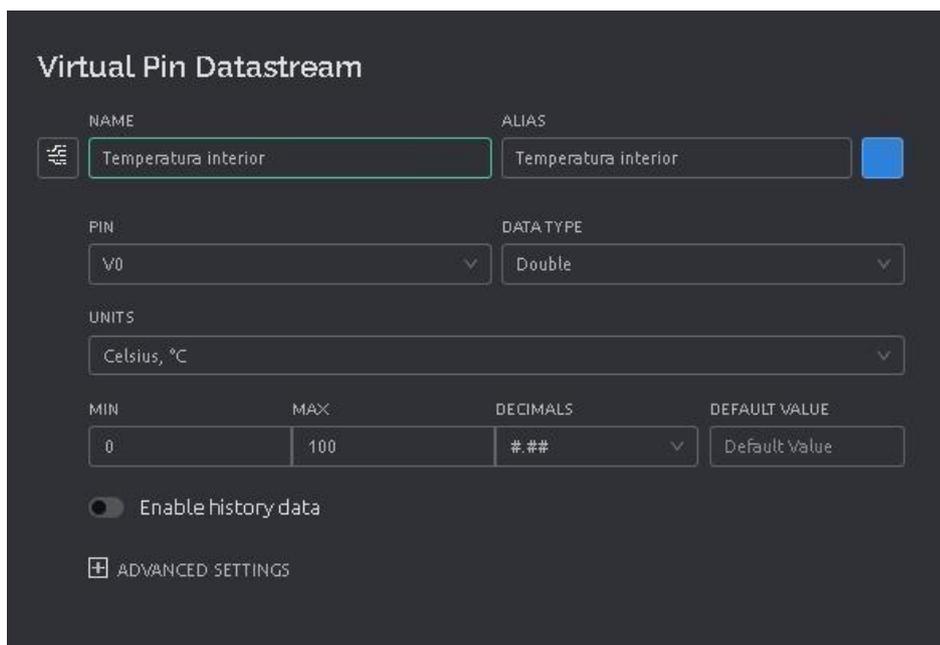
DEVICE NAME

TESIS

Cancel Create

Fig. 22. Nombre de dispositivo en Blynk IOT.

En la Fig. 23, se crearon pines virtuales en la sección Datastream para el dispositivo recién creado. El primer pin virtual configurado fue para la temperatura interior, designado como PIN V0. Este pin se estableció como tipo de datos Double, con unidades en Celsius, y se le asignó un rango de valores mínimo de 0 y máximo de 100.



Virtual Pin Datastream

NAME: Temperatura interior

ALIAS: Temperatura interior

PIN: V0

DATA TYPE: Double

UNITS: Celsius, °C

MIN: 0

MAX: 100

DECIMALS: ###

DEFAULT VALUE: Default Value

Enable history data

ADVANCED SETTINGS

Fig. 23. Pin virtual para la temperatura interior.

También, en la Fig. 24, se muestra la creación del pin virtual destinado a monitorear la humedad interior, designado como PIN V1. Este pin se configuró como tipo Double para representar valores decimales y permitir mediciones precisas de la humedad.

Virtual Pin Datastream

NAME: ALIAS:

PIN: DATATYPE:

UNITS:

MIN: MAX: DECIMALS: DEFAULT VALUE:

Enable history data

ADVANCED SETTINGS

Fig. 24. Pin virtual para la humedad interior.

En la Fig. 25, se presentan los pines virtuales creados, los cuales fueron fundamentales para su posterior uso en la programación y control del sistema.

Id	Name	Alias	Color	Pin	Data Type	Units	Is Raw	Min	Max	Actions
4	Humedad exterior	Humedad exterior	Orange	V5	Double		False	0	100	
2	Humedad interior	Humedad interior	Light Blue	V1	Double		False	0	100	
3	Temperatura exterior	Temperatura exterior	Light Blue	V2	Double	°C	False	0	100	
1	Temperatura interior	Temperatura interior	Blue	V0	Double	°C	False	0	100	
5	Control temperatura	Control temperatura	Green	V4	Double	°C	False	20	60	

Fig. 25. Pines virtuales creados.

Se configuraron los paneles web para permitir la visualización de los datos obtenidos de los pines virtuales. En este contexto, se utilizaron medidores (Gauge) para representar la temperatura y humedad, como se observa en la Fig. 26.

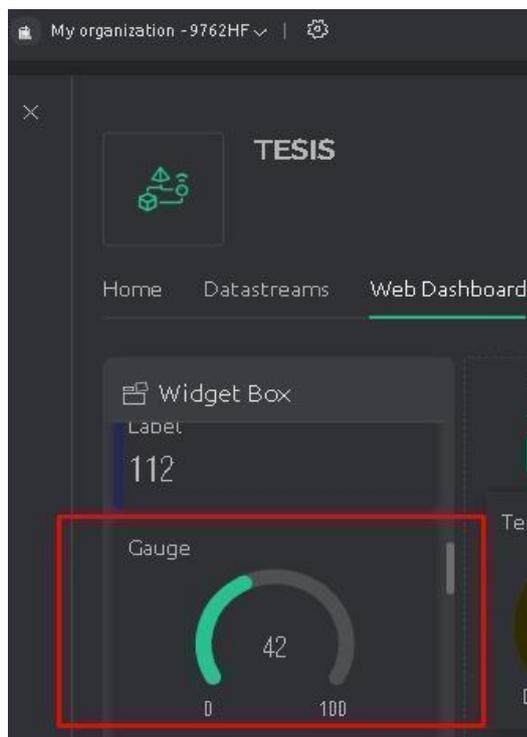


Fig. 26. Uso de módulos tipo Gauge para la representación de datos en el Web Dashboard.

Al situar el medidor (Gauge) en la interfaz de los paneles web, se realizó la configuración correspondiente para que pueda recibir los datos provenientes de cualquiera de los pines virtuales creados anteriormente. Este proceso se muestra en la Fig.27.

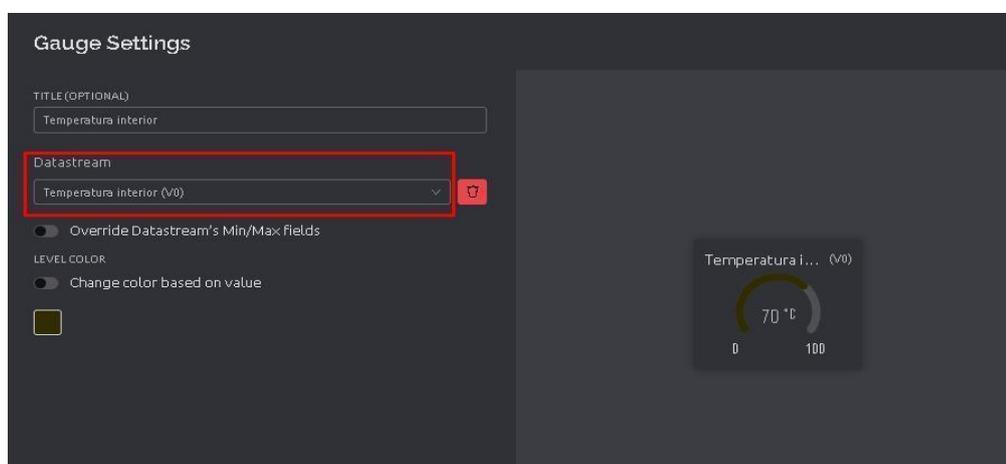


Fig. 27. Configuración de los módulos Gauge para visualización de datos.

Inicialmente, se dispusieron cuatro medidores (Gauges) destinados a recibir los datos de temperatura y humedad del entorno interno y externo del calefactor, a través de los sensores DTH11/DTH22. Esto se observa en la Fig. 28, en donde se puede apreciar las primeras pruebas de conexión realizadas.

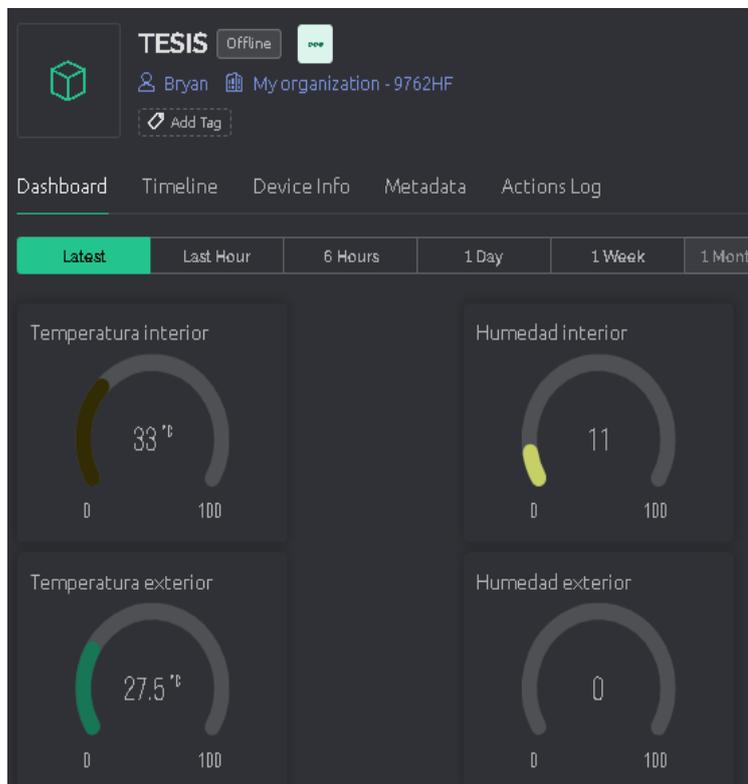


Fig. 28. Cuatro Gauges para recibir datos de los sensores de humedad/temperatura DHT11/DTH22.

16) Etapa de configuración de sensores DHT11/DTH22 en Blynk IOT:

Para obtener los datos de temperatura y humedad, fue necesario definir las librerías proporcionadas por Blynk IoT después de haber creado el dispositivo correspondiente. Esta configuración se muestra en la Fig. 29.

```
// Incluimos las librerías necesarias
#define BLYNK_TEMPLATE_ID "TMPL2hgUlpGK-"
#define BLYNK_TEMPLATE_NAME "TESIS"
#define BLYNK_AUTH_TOKEN "ei7AkgD8024Uxr32CoIMOc0Z4YWJ6eQO"
```

Fig. 29. Librerías necesarias para la comunicación con Blynk IOT.

Para establecer la conexión con el servidor de Blynk IoT a través de WiFi, se definieron las credenciales correspondientes. Esto implica establecer las directivas de preprocesador DHTPIN1 y DHTPIN2 como constantes con los valores 15 y 2 respectivamente, que representan los puertos digitales a los que fueron conectados los sensores DHT11/DHT22. A continuación, se crearon dos objetos de la clase DHT para cada sensor, inicializándolos con los puertos digitales correspondientes y el tipo de sensor (DHT22 en este caso).

Además, se declaró una variable auth como un array de caracteres (string) y se inicializaron con el valor de BLYNK_AUTH_TOKEN. Esta variable almacenó el token de autenticación de Blynk, necesario para conectar el dispositivo ESP32 al servidor de Blynk. Esta configuración se visualiza en la Fig. 30.

```
#define BLYNK_PRINT Serial // Configuración para imprimir mensajes de depuración en el
#include <WiFi.h> // Librería para la conexión WiFi en placas ESP32
#include <BlynkSimpleEsp32.h> // Librería para la comunicación con la plataforma Blynk
#include <DHT.h> // Librería para el sensor DHT

// Definimos las credenciales de conexión WiFi
char ssid[] = "XXXXXXXXXX"; // Nombre de la red WiFi
char pass[] = "XXXXXXXXXX"; // Contraseña de la red WiFi

// Creamos objetos DHT para interactuar con los sensores
#define DHTPIN1 15 // Puerto digital al que está conectado el primer sensor DHT
#define DHTPIN2 2 // Puerto digital al que está conectado el segundo sensor DHT
#define DHTTYPE DHT11 // Tipo de sensor DHT utilizado (DHT11 en este caso)
DHT dht1(DHTPIN1, DHTTYPE); // Primer sensor DHT
DHT dht2(DHTPIN2, DHTTYPE); // Segundo sensor DHT

// Configuración de la autenticación Blynk
char auth[] = BLYNK_AUTH_TOKEN;

BlynkTimer timer; // Creamos un objeto timer para realizar tareas con temporizadores
```

Fig. 30. Configuración de conexión y sensores para la comunicación entre Blynk y el módulo ESP32.

En la Fig. 31 se muestra la función encargada de enviar los valores de temperatura y humedad del primer sensor al servidor de Blynk. En primer lugar, la función lee los valores de humedad y temperatura del primer sensor DHT11. Luego, verifica si estos valores son válidos. En el caso de no cumplir esta condición inicial, imprime un mensaje de error en el puerto serial y sale de la función. Si los valores son válidos, los envía al servidor de Blynk utilizando los pines virtuales V0 y V1. Además, para fines de depuración, la función imprime los valores en el puerto serie, mostrando la temperatura y humedad del ambiente interior obtenidos por el primer sensor.

```

// Función para enviar los valores de temperatura y humedad de
void sendSensor1()
{
  float h = dht1.readHumidity(); // Leemos la humedad del primer sensor
  float t = dht1.readTemperature(); // Leemos la temperatura del primer sensor

  if (isnan(h) || isnan(t)) { // Verificamos si los valores son NaN
    Serial.println("Failed to read from DHT sensor 1!"); // Imprimimos el mensaje de error
    return; // Salimos de la función
  }

  // Enviamos los valores al servidor de Blynk
  Blynk.virtualWrite(V0, t); // Enviamos la temperatura del primer sensor
  Blynk.virtualWrite(V1, h); // Enviamos la humedad del primer sensor

  // Imprimimos los valores en el puerto serie para propósitos de depuración
  Serial.print("Sensor 1 - Temperatura interior : ");
  Serial.print(t);
  Serial.print("    Humedad interior : ");
  Serial.println(h);
}

```

Fig. 31. Función de envío de datos del sensor de temperatura/humedad al servidor de Blynk.

En su función de inicialización, el dispositivo puede establecer una comunicación serial para propósitos de depuración, asegurando que cualquier información relevante se pueda apreciar a través de este canal de comunicación. Luego, permite la conexión con Blynk utilizando las credenciales de autenticación previamente configuradas y la red WiFi proporcionada. Además, se inician ambos sensores DHT para recolectar los datos de temperatura y humedad. Para garantizar la actualización regular de los datos, se establecieron intervalos de tiempo para leer y enviar los datos de ambos sensores utilizando temporizadores. En cuanto a la función principal de ejecución, se aseguró de que el servidor de Blynk se ejecute correctamente y de que los temporizadores estén activos para mantener la lectura y envío de datos de manera continua. Esto se visualiza en la Fig. 32.

```

// Función de inicialización
void setup()
{
  Serial.begin(115200); // Iniciamos la comunicación s

  Blynk.begin(auth, ssid, pass); // Iniciamos la conex

  dht1.begin(); // Iniciamos el primer sensor DHT
  dht2.begin(); // Iniciamos el segundo sensor DHT

  timer.setInterval(10000, sendSensor1); // Configuram
  timer.setInterval(10000, sendSensor2); // Configuram
}

// Función principal de ejecución
void loop()
{
  Blynk.run(); // Ejecutamos el servidor de Blynk
  timer.run(); // Ejecutamos el temporizador
}

```

Fig. 32. Proceso de Inicialización y ejecución del sistema para monitoreo y envío de datos.

Se comprobó que la conexión de los sensores DHT11 con los pines del ESP32 sea correcta. Esto se puede observar en la Fig. 33.

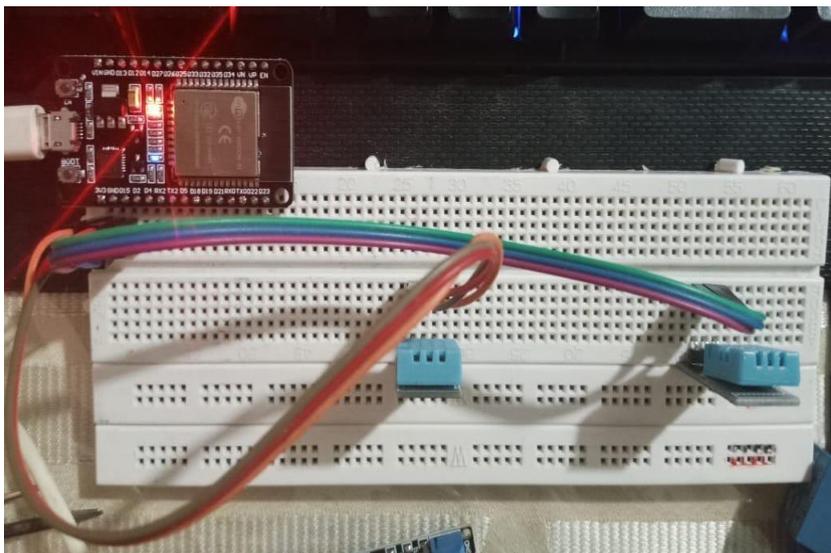


Fig. 33. Conexión física de las primeras pruebas.

Se cargó el código al módulo ESP32 y se esperó a que se establezca la conexión con el servidor Blynk mientras los sensores obtenían los datos. Esta operación se puede observar en la Fig. 34.

```
[619] IP: 192.168.137.160
[619]

  _ _ _ _ _
 / _ \ / _ \ / _ \ / _ \
/_ _ \/_ _ \/_ _ \/_ _ \
 / _ \ / _ \ / _ \ / _ \
/_ _ \/_ _ \/_ _ \/_ _ \
 / _ \ / _ \ / _ \ / _ \
/_ _ \/_ _ \/_ _ \/_ _ \

#StandWithUkraine   https://bit.ly/swua

[629] Connecting to blynk.cloud:80
[1165] Ready (ping: 203ms).
Sensor 1 - Temperatura interior : 27.30   Humedad interior : 81.00
Sensor 2 - Temperatura exterior: 27.00   Humedad exterior : 43.00
Sensor 1 - Temperatura interior : 27.30   Humedad interior : 81.00
Sensor 2 - Temperatura exterior: 27.00   Humedad exterior : 43.00
Sensor 1 - Temperatura interior : 27.30   Humedad interior : 81.00
Sensor 2 - Temperatura exterior: 27.00   Humedad exterior : 44.00
Sensor 1 - Temperatura interior : 27.30   Humedad interior : 81.00
Sensor 2 - Temperatura exterior: 27.00   Humedad exterior : 43.00
```

Fig. 34. Lectura de datos de temperatura y humedad.

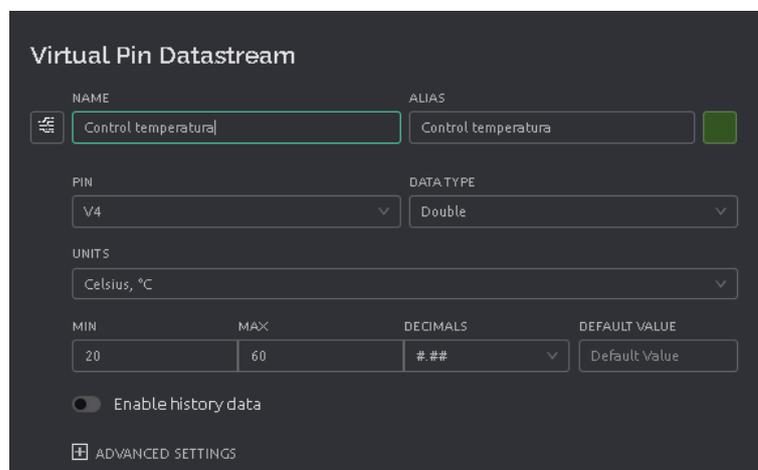
En la Fig. 28 se observa la ubicación de los módulos del dashboard principal creados para para verificar si el servidor Blynk está recibiendo los datos enviados desde el módulo ESP32. Esta verificación se puede observar en la Fig. 35.



Fig. 35. Se reflejan los datos de la temperatura y humedad en el servidor Blynk.

17) Etapa de configuración del control de temperatura con Blynk IOT:

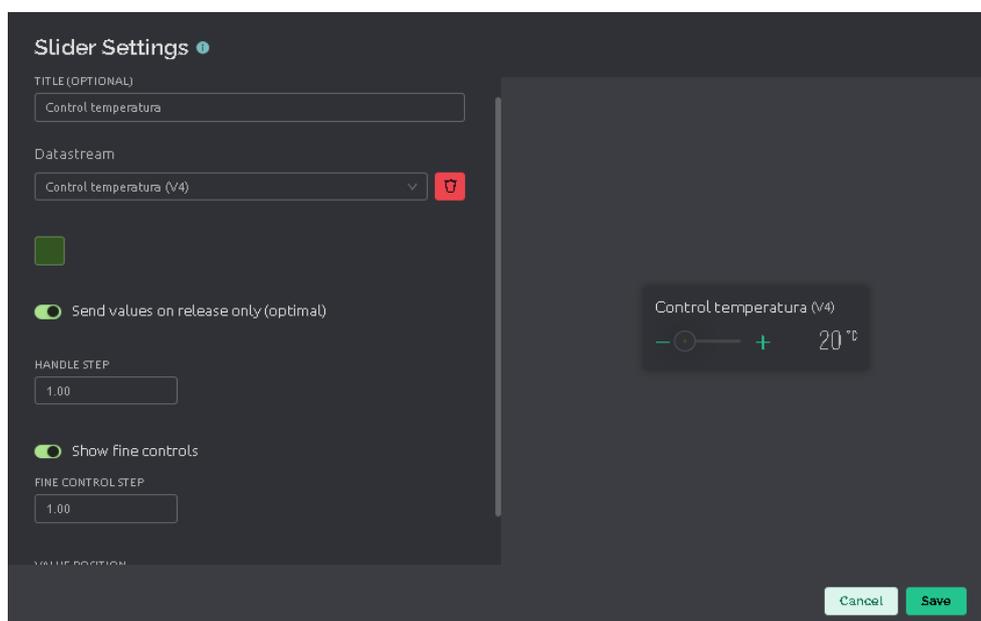
Una vez que los datos de temperatura y humedad de los sensores DHT22 fueron enviados exitosamente a Blynk, se instaló la resistencia níquelina de 12V, que cumplió la función de calefactor. Para este fin, se creó un nuevo PIN virtual en Blynk, ubicado específicamente en la vista de Datastream. Se establecieron valores de temperatura mínima de 20 grados y máxima de 60 grados, como se detalla en la Fig. 36.



The screenshot shows the 'Virtual Pin Datastream' configuration interface. It includes fields for NAME (Control temperatura), ALIAS (Control temperatura), PIN (V4), DATA TYPE (Double), and UNITS (Celsius, °C). There are also input fields for MIN (20), MAX (60), DECIMALS (###), and DEFAULT VALUE (Default Value). A toggle for 'Enable history data' is present, along with an 'ADVANCED SETTINGS' button.

Fig. 36. Creación del Pin virtual para el control de temperatura.

En la vista del tablero web de Blynk, se añadió un control deslizante (slider) para gestionar la temperatura, realizando las configuraciones necesarias para seleccionar el pin virtual creado previamente, como se aprecia en la Fig. 37.



The screenshot shows the 'Slider Settings' configuration interface. It includes fields for TITLE (OPTIONAL) (Control temperatura), Datastream (Control temperatura (V4)), HANDLE STEP (1.00), and FINE CONTROL STEP (1.00). There are also toggle switches for 'Send values on release only (optimal)' and 'Show fine controls'. A preview of the slider control is shown on the right, displaying 'Control temperatura (V4)' with a slider and a value of 20 °C. 'Cancel' and 'Save' buttons are at the bottom right.

Fig. 37. Configuración del slider para el control de temperatura.

En la Fig. 38 se observa la representación visual del slider, el cual permitió ajustar la temperatura de manera sencilla y precisa para el control del calefactor.

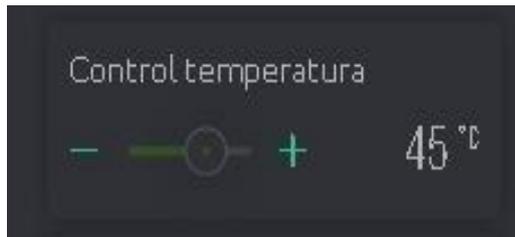


Fig. 38. Slider que permite el control de temperatura del calefactor.

En el código, se incluyeron variables para el control automático de la temperatura. La variable *TARGET_TEMP* representó la temperatura objetivo, fijada en 45 grados Celsius. La variable *heaterState* registró el estado actual del calentador, siendo FALSO cuando estaba apagado. *TEMP_THRESHOLD* indicó el margen de temperatura permitido para mantenerse cerca de la temperatura objetivo, establecido en 1 grado Celsius. El pin 13 del dispositivo ESP32 se utilizó para controlar físicamente el módulo de relé que activaba o desactivaba la resistencia níquelina, permitiendo así el control automático del calentador para mantener la temperatura dentro del rango deseado. Esta configuración se ilustra en la Fig. 39.

```
// Variables para el control de
int TARGET_TEMP = 45; // Tempera
bool heaterState = false; // Est
const int TEMP_THRESHOLD = 1; //

// Pin para controlar el módulo
const int RELAY_PIN = 13; // Pin
```

Fig. 39. Variables y configuración para control de temperatura con resistencia níquelina.

Se ajustó la función *sendSensor1* del sensor de temperatura y humedad interior para proporcionar un control adicional sobre la resistencia níquelina, que funcionó como el calentador. Además de leer los datos de temperatura y humedad para enviarlos al servidor de Blynk, esta función también supervisó y controló la temperatura ambiente (afuera del calefactor). Después de verificar la validez de los datos del sensor, se implementó un control de temperatura. Si la temperatura medida superaba la temperatura objetivo más el margen establecido y el calentador estaba apagado, se activaba automáticamente. Por el contrario, si la temperatura medida era inferior a la temperatura objetivo menos el margen y el calentador estaba encendido, se apagaba. Esta adaptación permitió que el sistema mantenga la temperatura

interior dentro de los límites deseados, regulando la activación y desactivación del calentador según sea necesario. Este proceso se detalla en la Fig. 40.

```
// Control de la temperatura
if (t > TARGET_TEMP + TEMP_THRESHOLD && !heaterState) {
  // Encender calentador si la temperatura es mayor que la temperatura objetivo más el margen
  heaterState = true;
  digitalWrite(RELAY_PIN, HIGH); // Activar módulo de relé para la resistencia niquelina
  Serial.println("Calentador encendido ");
} else if (t < TARGET_TEMP - TEMP_THRESHOLD && heaterState) {
  // Apagar calentador si la temperatura es menor que la temperatura objetivo menos el margen
  heaterState = false;
  digitalWrite(RELAY_PIN, LOW); // Desactivar módulo de relé para la resistencia niquelina
  Serial.println("Calentador apagado");
}
```

Fig. 40. Control de temperatura con resistencia niquelina utilizando sensor DHT11/DHT22.

Se desarrolló una función para actualizar dinámicamente la temperatura objetivo (*TARGET_TEMP*) con el valor del control deslizante ubicado en la plataforma Blynk (pin virtual V4). Esta función permitió ajustar la temperatura deseada de manera interactiva desde la aplicación Blynk. Este proceso se ilustra detalladamente en la Fig. 41.

```
// Función para manejar el cambio de temperatura
BLYNK_WRITE(V4)
{
  TARGET_TEMP = param.asInt();
}
```

Fig. 41. Control interactivo de temperatura en Blynk.

La Fig. 42 muestra la modificación de la función de inicialización, en la que se han añadido dos líneas de código significativas. La primera línea, *pinMode(RELAY_PIN, OUTPUT)*;, permitió la configuración del pin del módulo relé para la resistencia niquelina como salida. Esta configuración fue esencial para que el microcontrolador pueda enviar señales al módulo relé y controlar la activación o desactivación de la resistencia niquelina. La segunda línea, *digitalWrite(RELAY_PIN, LOW)*;, inicialmente desactivaba el módulo relé para la resistencia niquelina. Este paso garantizó que el calentador esté apagado al inicio, evitando un calentamiento innecesario hasta que sea requerido.

```

// Función de inicialización
void setup()
{
  Serial.begin(115200); // Iniciamos la c

  Blynk.begin(auth, ssid, pass); // Inici

  dht1.begin(); // Iniciamos el primer se
  dht2.begin(); // Iniciamos el segundo s

  pinMode(RELAY_PIN, OUTPUT); // Configu
  digitalWrite(RELAY_PIN, LOW); // Desact

  timer.setInterval(10000, sendSensor1);
  timer.setInterval(10000, sendSensor2);
}

```

Fig. 42. Configuración de la función inicial agregando el control de los pines del control de temperatura.

Se ejecutaron pruebas para verificar la efectividad del control, ajustando inicialmente el slider a 60 grados y luego colocándolo en la temperatura objetivo de 45 grados, como se muestra en la Fig. 43.

```

Calentador encendido
Sensor 2 - Temperatura exterior: 27.20      Humedad exterior : 79.00
Sensor 1 - Temperatura interior : 50.00     Humedad interior : 3.00
Sensor 2 - Temperatura exterior: 27.20      Humedad exterior : 79.00
Sensor 1 - Temperatura interior : 50.00     Humedad interior : 3.00
Sensor 2 - Temperatura exterior: 27.30      Humedad exterior : 79.00
Sensor 1 - Temperatura interior : 48.00     Humedad interior : 4.00
Sensor 2 - Temperatura exterior: 27.30      Humedad exterior : 79.00
Sensor 1 - Temperatura interior : 47.00     Humedad interior : 4.00
Sensor 2 - Temperatura exterior: 27.30      Humedad exterior : 79.00
Sensor 1 - Temperatura interior : 46.00     Humedad interior : 5.00
Sensor 2 - Temperatura exterior: 27.30      Humedad exterior : 79.00
Sensor 1 - Temperatura interior : 45.00     Humedad interior : 5.00
Sensor 2 - Temperatura exterior: 27.30      Humedad exterior : 79.00
Sensor 1 - Temperatura interior : 45.00     Humedad interior : 5.00

```

Fig. 43. Control automático de temperatura con ciclo de encendido del calentador.

Una vez que el calentador alcanzó la temperatura objetivo y se apagó, la temperatura comienza a disminuir. Sin embargo, debido a la configuración establecida, esta disminución es detectada y el calentador se enciende nuevamente hasta alcanzar nuevamente la temperatura objetivo, como se muestra en la Fig. 44.

```

Calentador apagado
Sensor 2 - Temperatura exterior: 27.30 Humedad exterior : 79.00
Sensor 1 - Temperatura interior : 42.00 Humedad interior : 7.00
Sensor 2 - Temperatura exterior: 27.30 Humedad exterior : 79.00
Sensor 1 - Temperatura interior : 42.00 Humedad interior : 7.00
Sensor 2 - Temperatura exterior: 27.30 Humedad exterior : 79.00
Sensor 1 - Temperatura interior : 42.00 Humedad interior : 7.00
Sensor 2 - Temperatura exterior: 27.30 Humedad exterior : 79.00
Sensor 1 - Temperatura interior : 42.00 Humedad interior : 7.00
Sensor 2 - Temperatura exterior: 27.30 Humedad exterior : 79.00
Sensor 1 - Temperatura interior : 43.00 Humedad interior : 6.00
Sensor 2 - Temperatura exterior: 27.30 Humedad exterior : 79.00
Sensor 1 - Temperatura interior : 43.00 Humedad interior : 6.00
Sensor 2 - Temperatura exterior: 27.30 Humedad exterior : 79.00
Sensor 1 - Temperatura interior : 43.00 Humedad interior : 6.00
Sensor 2 - Temperatura exterior: 27.30 Humedad exterior : 79.00

```

Fig. 44. Control automático de temperatura con ciclo de apagado del calentador.

La Fig. 45 muestra la conexión de la resistencia níquelina y el proceso de prueba para asegurar una correcta conexión con el módulo ESP32.

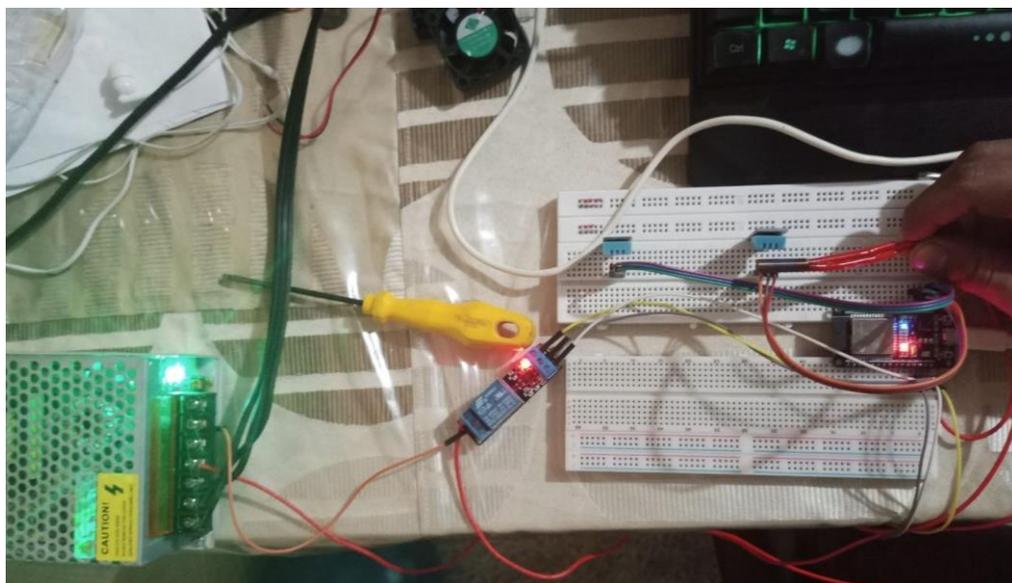


Fig. 45. Conexión y pruebas de la resistencia níquelina en ESP32 para Blynk.

El desarrollo del proceso de medición y control de temperatura mediante Blynk IoT resultó exitoso. Sin embargo, no se completó su implementación debido a limitaciones en la suscripción necesaria para agregar más dashboards y controlar más dispositivos. La suscripción era necesaria para acceder a todas las funcionalidades del servicio. Además, el servidor creado solo permitía mantener en línea el control de dispositivos IoT durante una semana, después de lo cual se requería una suscripción para continuar con el mismo nivel de funcionalidad. En vista

de estas limitaciones, se optó por utilizar MQTT y Node-RED para el proyecto, ya que ofrecían una solución más flexible y adaptable a largo plazo.

18) Configuración de servidor MQTT y Node-RED:

Se abrió una terminal en Ubuntu y se actualizó la lista de paquetes disponibles del sistema operativo utilizando el comando "sudo apt update". Esto se muestra en la Fig. 46.

```
bryan@bryan:~$ sudo apt update
[sudo] contraseña para bryan:
Des:1 https://dl.google.com/linux/chrome/deb stable InRelease [1.825 B]
Des:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Des:3 https://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1.081 B]
1
```

Fig. 46. Actualización de la lista de paquetes en Ubuntu.

Se configuró un servidor MQTT en una máquina virtual Ubuntu para actuar como el broker distribuidor de mensajes, además, sirvió como punto central de conexión entre el software Node-RED y otros dispositivos en la red. Este servidor se encargó de gestionar la comunicación entre los diferentes dispositivos IoT y el Node-RED, permitiendo el intercambio de mensajes y datos de forma eficiente y segura. La Fig. 47 lustra la instalación de Mosquitto.

```
bryan@bryan:~$ sudo apt install mosquitto
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
mosquitto ya está en su versión más reciente (2.0.11-1ubuntu1.1).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 20 no actualizados.
```

Fig. 47. Instalación de Mosquitto en Ubuntu.

En la Fig. 48 se muestra que con el comando "sudo systemctl status mosquitto" se muestra el estado del servidor MQTT configurado.

```

bryan@bryan:~$ sudo systemctl status mosquitto
[sudo] contraseña para bryan:
Lo siento, pruebe otra vez.
[sudo] contraseña para bryan:
● mosquitto.service - Mosquitto MQTT Broker
   Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; vendor pre
   Active: active (running) since Tue 2024-02-27 21:35:47 -05; 52min ago
     Docs: man:mosquitto.conf(5)
           man:mosquitto(8)
   Process: 690 ExecStartPre=/bin/mkdir -m 740 -p /var/log/mosquitto (code=exi
   Process: 704 ExecStartPre=/bin/chown mosquitto /var/log/mosquitto (code=exi
   Process: 708 ExecStartPre=/bin/mkdir -m 740 -p /run/mosquitto (code=exited,
   Process: 710 ExecStartPre=/bin/chown mosquitto /run/mosquitto (code=exited,
   Main PID: 711 (mosquitto)
      Tasks: 1 (limit: 4554)
     Memory: 2.8M
        CPU: 6.927s
    CGroup: /system.slice/mosquitto.service
            └─711 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf

feb 27 21:35:47 bryan systemd[1]: Starting Mosquitto MQTT Broker...
feb 27 21:35:47 bryan systemd[1]: Started Mosquitto MQTT Broker.
Lines 1-18/18 (END)

```

Fig. 48. Estado activo del mosquitto.

Para configurar Node-RED, primero fue necesario asegurarse de tener acceso a una máquina física compatible, preferiblemente con el sistema operativo Windows instalado. Luego, se procedió a descargar e instalar Node.js, que es un entorno de ejecución para JavaScript que permitió ejecutar aplicaciones de Node.js. Esto se visualiza en la Fig. 49.

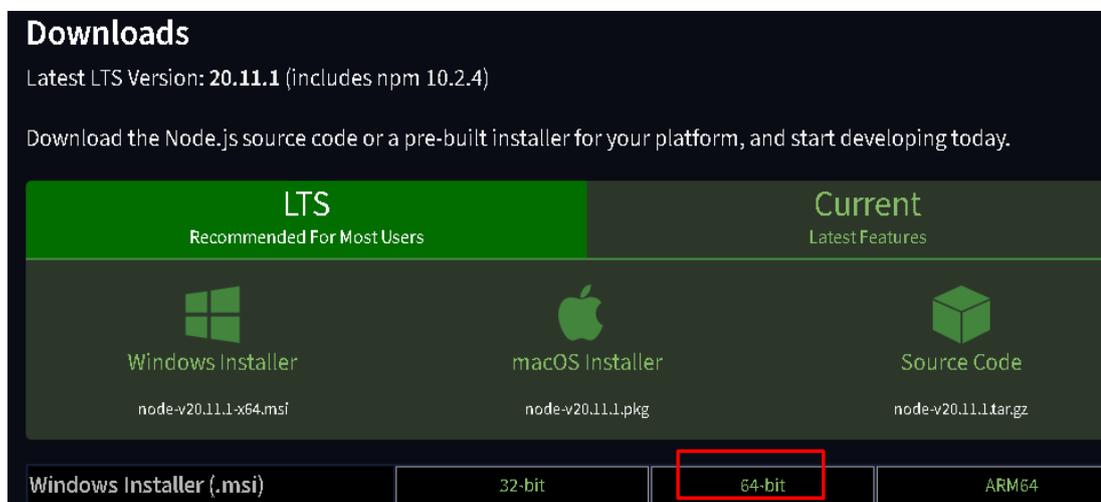
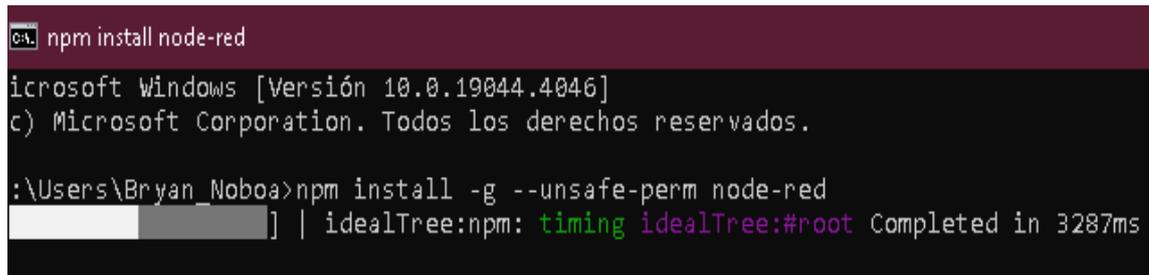


Fig. 49. Instalación de Node.js en Windows.

Una vez instalado Node.js, se instaló Node-RED mediante el administrador de paquetes npm (Node Package Manager), que se incluye con Node.js. Esto se representa en la Fig. 50.



```

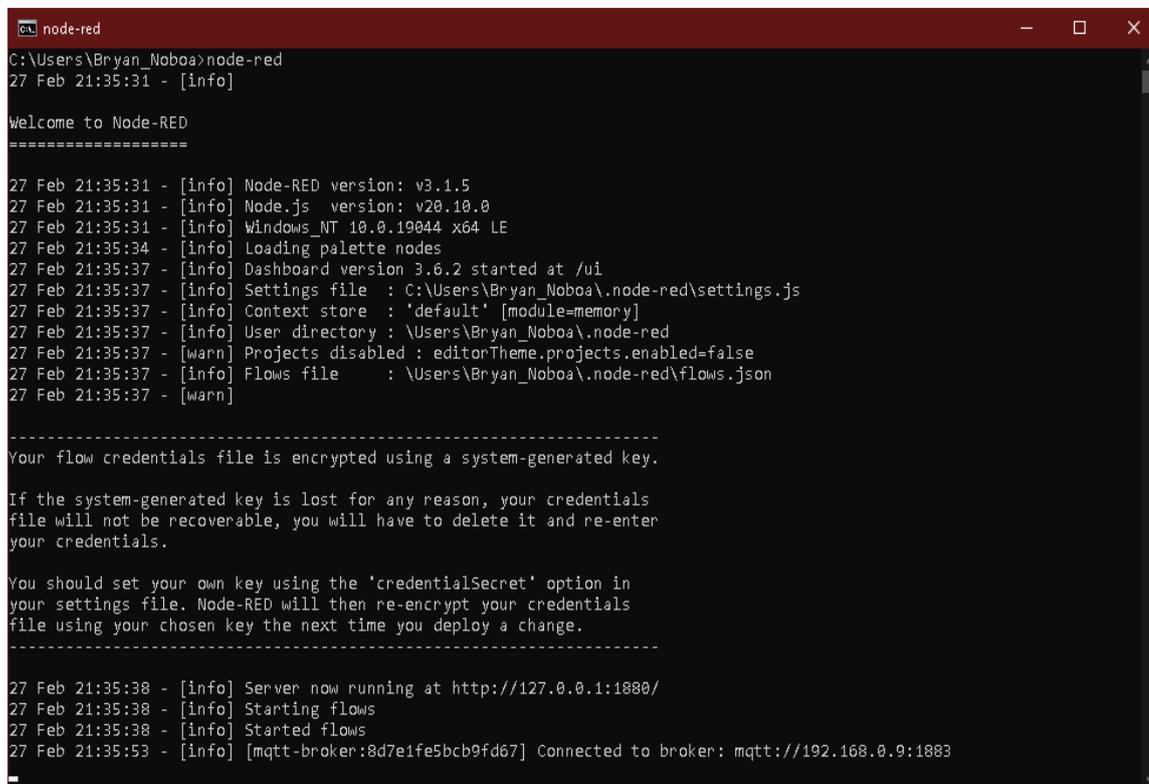
C:\> npm install node-red

Microsoft Windows [Versión 10.0.19044.4046]
(c) Microsoft Corporation. Todos los derechos reservados.

:\Users\Bryan_Noboa>npm install -g --unsafe-perm node-red
[Progress bar] | idealTree:npm: timing idealTree:#root Completed in 3287ms
  
```

Fig. 50. Instalación de Node-RED en Windows.

Luego de la instalación, se pudo acceder a Node-RED a través de un navegador web, utilizando la dirección IP de la máquina y el puerto predeterminado para el software (generalmente el puerto 1880). Una vez que Node-RED entró en funcionamiento, se configuraron los nodos y las conexiones para interactuar con otros servicios y dispositivos. La Fig. 51 ilustra el proceso de inicialización de Node-RED.



```

node-red
C:\Users\Bryan_Noboa>node-red
27 Feb 21:35:31 - [info]

Welcome to Node-RED
=====

27 Feb 21:35:31 - [info] Node-RED version: v3.1.5
27 Feb 21:35:31 - [info] Node.js version: v20.10.0
27 Feb 21:35:31 - [info] Windows_NT 10.0.19044 x64 LE
27 Feb 21:35:34 - [info] Loading palette nodes
27 Feb 21:35:37 - [info] Dashboard version 3.6.2 started at /ui
27 Feb 21:35:37 - [info] Settings file : C:\Users\Bryan_Noboa\.node-red\settings.js
27 Feb 21:35:37 - [info] Context store : 'default' [module=memory]
27 Feb 21:35:37 - [info] User directory : \Users\Bryan_Noboa\.node-red
27 Feb 21:35:37 - [warn] Projects disabled : editorTheme.projects.enabled=false
27 Feb 21:35:37 - [info] Flows file : \Users\Bryan_Noboa\.node-red\flows.json
27 Feb 21:35:37 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----

27 Feb 21:35:38 - [info] Server now running at http://127.0.0.1:1880/
27 Feb 21:35:38 - [info] Starting flows
27 Feb 21:35:38 - [info] Started flows
27 Feb 21:35:53 - [info] [mqtt-broker:8d7e1fe5bcb9fd67] Connected to broker: mqtt://192.168.0.9:1883
  
```

Fig. 51. Configuración inicial y puesta en marcha de Node-Red.

Node-RED es una plataforma visual que posibilita la creación de flujos de trabajo mediante la interconexión de varios nodos. Estos nodos representan diferentes componentes y funciones que interactúan entre sí para realizar tareas específicas. En el ámbito de IoT y la

configuración de un dashboard, estos nodos pueden abarcar elementos gráficos como medidores (gauge) que muestran datos de sensores, entre otras funciones. En la Fig. 52 se muestra la representación visual del software por acceso vía web.

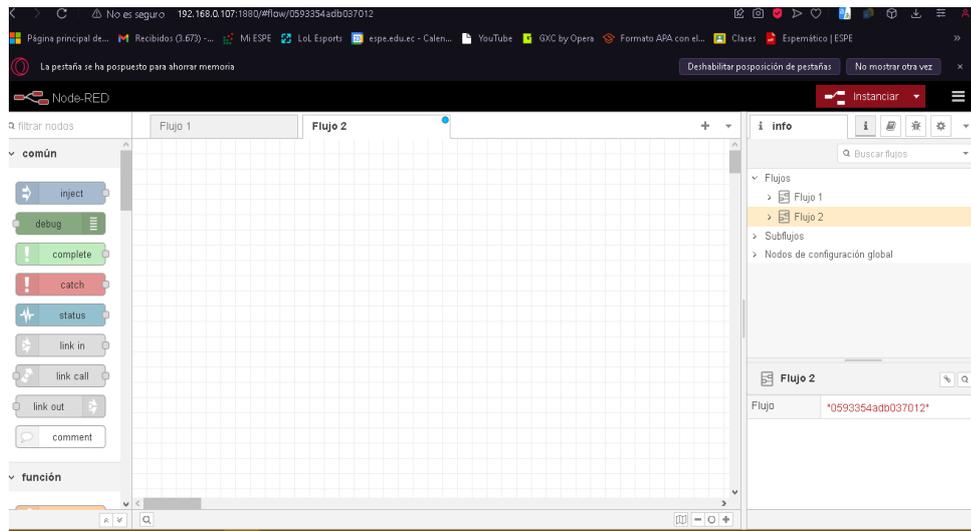


Fig. 52. Muestra del panel de control de flujos de trabajo en el software Node-Red.

19) Desarrollo de nodos para configurar el flujo de trabajo en Node-RED:

Se utilizaron los nodos MQTT IN, para establecer la conexión con el servidor y suscribirse a temas específicos para recibir mensajes generados por el microcontrolador ESP32. Los nodos mencionados se enlazaron a un dashboard configurado para representar la temperatura y humedad en forma de medidores (gauge), facilitando la visualización precisa de estos datos. El flujo de trabajo se ilustra en la Fig. 53.

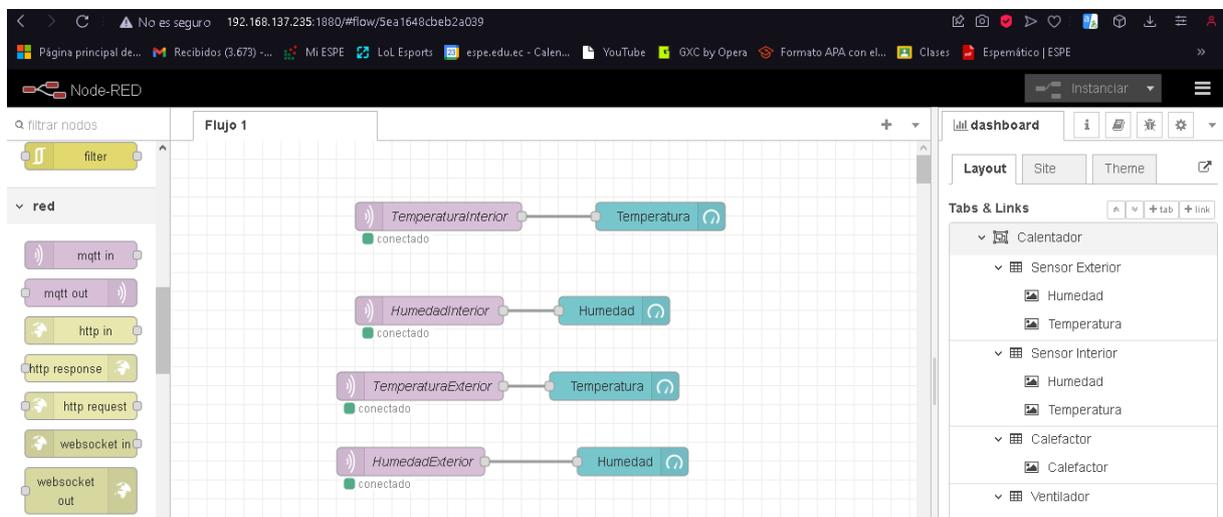


Fig. 53. Configuración de nodos MQTT IN y dashboards de tipo gauge en Node-RED.

También se utilizaron los nodos MQTT OUT para establecer la conexión con el servidor y enviar mensajes. Dichos nodos tuvieron la responsabilidad de publicar datos hacia el módulo ESP32 para controlar los dispositivos IoT, como el calefactor, que se vinculó a un nodo de tipo slider (resistencia niquelina) para ajustar la temperatura deseada, y la ventilación, que se conectó a un nodo de tipo switch (ventiladores) para activar o desactivar la función de ventilación. El flujo de trabajo correspondiente se detalla en la Fig. 54.

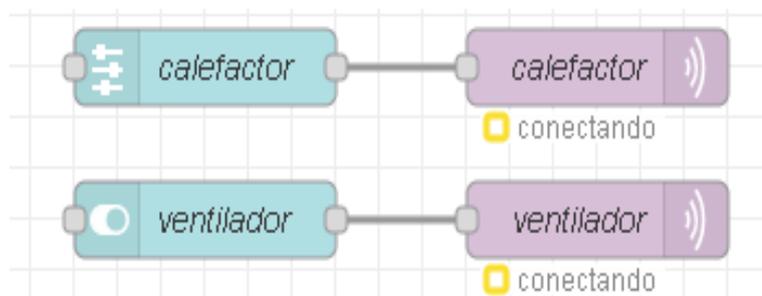


Fig. 54. Flujo de trabajo con nodos MQTT OUT controlando la niquelina y los ventiladores.

20) Configuración del programa para la gestión de dispositivos IoT mediante MQTT y su integración con Node-RED:

Se desarrolló un programa utilizando el entorno Arduino IDE que se encargó de la gestión de dispositivos mediante una conexión WiFi y el protocolo MQTT. Utilizando bibliotecas específicas, se estableció una comunicación bidireccional con el servidor MQTT. Se establecieron diversos "temas" o "topics" para la transmisión de datos (estos fueron enviados a Node-Red), los cuales abarcaron información sobre la temperatura y humedad tanto en el interior del calefactor como en el exterior, además de la temperatura deseada y el control de los ventiladores. Mediante la definición de constantes para los pines de sensores y dispositivos IoT, así como los parámetros de la red WiFi y la dirección IP del servidor MQTT, el programa permitió el monitoreo y control remoto de los dispositivos desde cualquier ubicación dentro de la red, como se observa en la Fig. 55.

```

#include <WiFi.h>
#include <PubSubClient.h>
#include <DHT.h>

// Definiciones de pines para los sensores DHT y los actuadores (calefactor y ventiladores)
#define DHTPIN1 15 // Pin de datos del primer sensor DHT
#define DHTPIN2 5 // Pin de datos del segundo sensor DHT
#define DHTTYPE DHT22 // Tipo de sensor DHT utilizado
#define pinCalefactor 12 // Pin de control del calefactor
#define pinVentilador1 14 // Pin de control del primer ventilador
// #define pinVentilador2 25 // Pin de control del segundo ventilador (cambiado)

// Credenciales de la red WiFi y configuración del servidor MQTT
const char* ssid = "BRYAN_NOBOA"; // Nombre de la red WiFi
const char* password = "XXXXXXXXXX"; // Contraseña de la red WiFi
const char* mqtt_server = "192.168.137.124"; // Dirección IP del servidor MQTT

// Temas MQTT para la publicación y suscripción de datos
const char* topic_temp_interior = "TemperaturaInterior"; // Temperatura interior
const char* topic_hum_interior = "HumedadInterior"; // Humedad interior
const char* topic_temp_exterior = "TemperaturaExterior"; // Temperatura exterior
const char* topic_hum_exterior = "HumedadExterior"; // Humedad exterior
const char* topic_temp_deseada = "temperatura_deseada"; // Temperatura deseada
const char* topic_ventilador = "ventilador"; // Control del ventilador

```

Fig. 55. Configuración del programa para la gestión de dispositivos IoT mediante MQTT.

Se inició un cliente WiFi y un cliente *PubSubClient*, que se emplearon para la comunicación dentro de la red. Además, se definieron variables para almacenar la temperatura deseada (la cual representaba la temperatura inicial) y para establecer un margen alrededor de esta temperatura, lo que permitió un control más preciso del entorno. También se controló el estado del ventilador, determinando si está activo o no según las condiciones de temperatura. Esto simplificó el monitoreo y el control remoto de los dispositivos, especialmente en condiciones en donde se requería un ajuste preciso de la temperatura, como se visualiza en la Fig. 56.

```

WiFiClient espClient;
PubSubClient client(espClient);

float temperatura_deseada = 0; //
float margen = 2.0; // Margen alre
bool ventiladorActivo = false; //

```

Fig. 56. Configuración de clientes WiFi y MQTT en el código desarrollado.

Para el resto del código, se consideró el uso del microcontrolador ESP32 para administrar todos los dispositivos mediante una conexión WiFi y el protocolo MQTT. Se

definieron constantes para los pines de los sensores y dispositivos, así como los parámetros de la red WiFi y la dirección IP del servidor MQTT. El programa estableció una comunicación bidireccional con el servidor MQTT y se suscribió a los temas (tópicos) específicos para la transmisión de datos relacionados con la temperatura y humedad tanto interior como exterior, la temperatura deseada y el control de los ventiladores.

La función de *devolución de llamada* se encargó de manejar los mensajes recibidos, activando o desactivando el ventilador según el estado del payload en el tema MQTT "ventilador", y ajustando la temperatura deseada en base al payload recibido en el tema MQTT "temperatura_deseada". El bucle principal del programa realizó la lectura de los sensores de temperatura y humedad interior y exterior, publicando los datos correspondientes en los temas MQTT definidos y controlando el calefactor según la temperatura deseada y un margen establecido. Esto permitió la gestión remota y eficiente de los dispositivos IoT, contribuyendo a la automatización y mejora de la eficiencia en el entorno propuesto para la calefacción de las fibras de impresión PLA y PET.

21) Edición de nodos en Node-RED asignando los temas y la dirección IP del servidor MQTT:

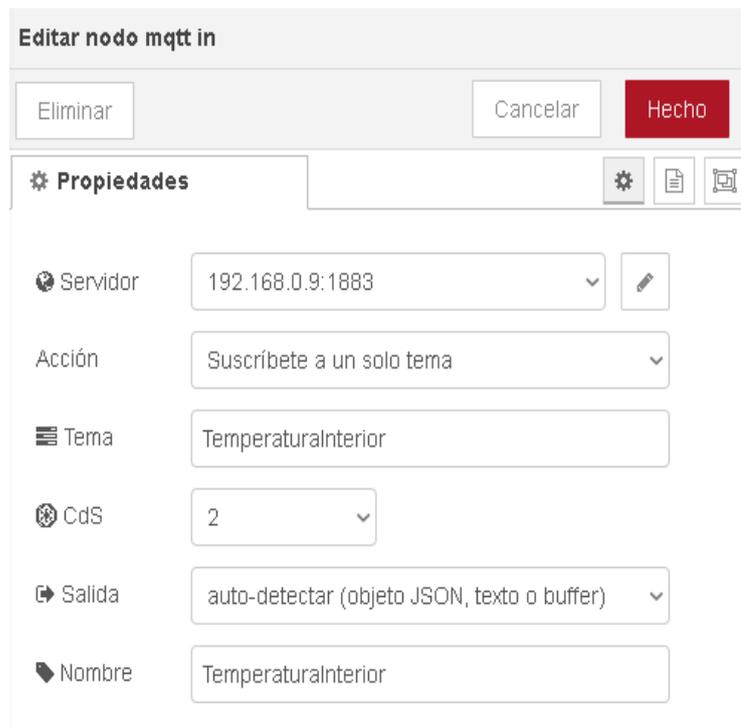
Se configuraron los nodos MQTT-In, los cuales representan el servidor, especificando la dirección IP de la máquina virtual que alojaba el servidor MQTT y el puerto correspondiente. Este proceso se detalla en la Fig. 57.

The image shows the configuration window for an MQTT broker node in Node-RED. The window title is 'Editar nodo mqtt in > Editar nodo mqtt-broker'. At the top, there are three buttons: 'Eliminar', 'Cancelar', and 'Actualizar'. Below this is a section titled 'Propiedades' with a gear icon and a document icon. Underneath, there are three tabs: 'Conexión', 'Seguridad', and 'Mensajes'. The 'Conexión' tab is selected and contains the following settings:

- Servidor:** 192.168.0.9
- Puerto:** 1883
- Conectar automáticamente
- Utilizar TLS
- Protocolo:** MQTT V3.1.1 (dropdown menu)
- ID Cliente:** Dejar en blanco para auto generado
- Mantener activo:** 60
- Sesión: Usar sesión limpia

Fig. 57. Configuración de las propiedades del servidor MQTT en Node-RED.

Una vez configurado el servidor, fue necesario suscribirse al tema definido en la Fig. 55 y asignarle un nombre. Este proceso se ilustra en la Fig. 58.



Editar nodo mqtt in

Eliminar Cancelar Hecho

Propiedades

Servidor 192.168.0.9:1883

Acción Suscríbete a un solo tema

Tema TemperaturaInterior

CdS 2

Salida auto-detectar (objeto JSON, texto o buffer)

Nombre TemperaturaInterior

Fig. 58. Configuración de nodos MQTT-In.

También se configuraron los nodos MQTT OUT para el calefactor y el ventilador, utilizando el servidor previamente configurado en la Fig. 57. Este proceso se muestra en la Fig. 59.



Editar nodo mqtt out

Eliminar Cancelar Hecho

Propiedades

Servidor 192.168.0.9:1883

Tema temperatura_deseada

CdS Retener

Nombre calefactor

Consejo: Deja el tema, CdS o manténgalo en blanco si quieres configurarlos a través de las propiedades del mensaje.

Fig. 59. Configuración de nodos MQTT-Out.

Para el nodo del slider, fué necesario ingresar el tema (topic) correspondiente, que en este caso era "*temperatura_deseada*", tal como se muestra en la Fig. 60.

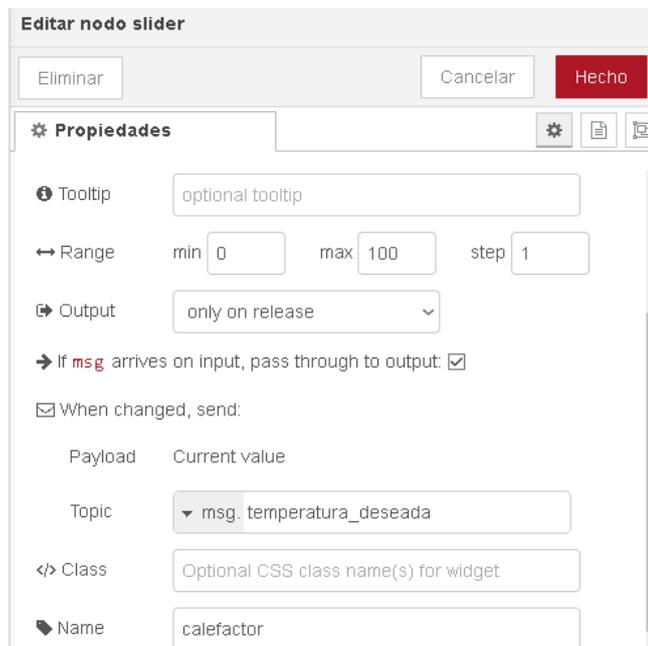


Fig. 60. Configuración de nodo de slider.

Para el nodo switch, se ingresó el tema (topic) correspondiente, que en este caso fue "*ventilador*". En el campo On Payload se registró el valor que activaría la función (encender), que fue el número 0, mientras que en la opción Off Payload se especificó el valor que desactivaría la función (apagar), que fue el número 1. Ambos valores debían ser de tipo texto, como se observa en la Fig. 61.

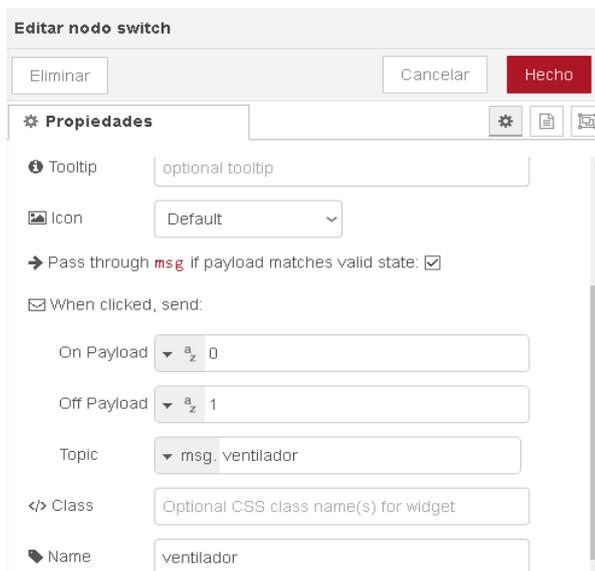


Fig. 61. Configuración de nodo switch para controlar el encendido y apagado.

22) Vista de la interfaz de usuario (UI) de los nodos empleados en Node-RED:

Una vez que se organizaron todos los elementos dentro del dashboard, se estableció la vista general del panel de control (Interfaz Gráfica de Usuario) principal que el usuario final podrá controlar, tal como se aprecia en la Fig. 62.

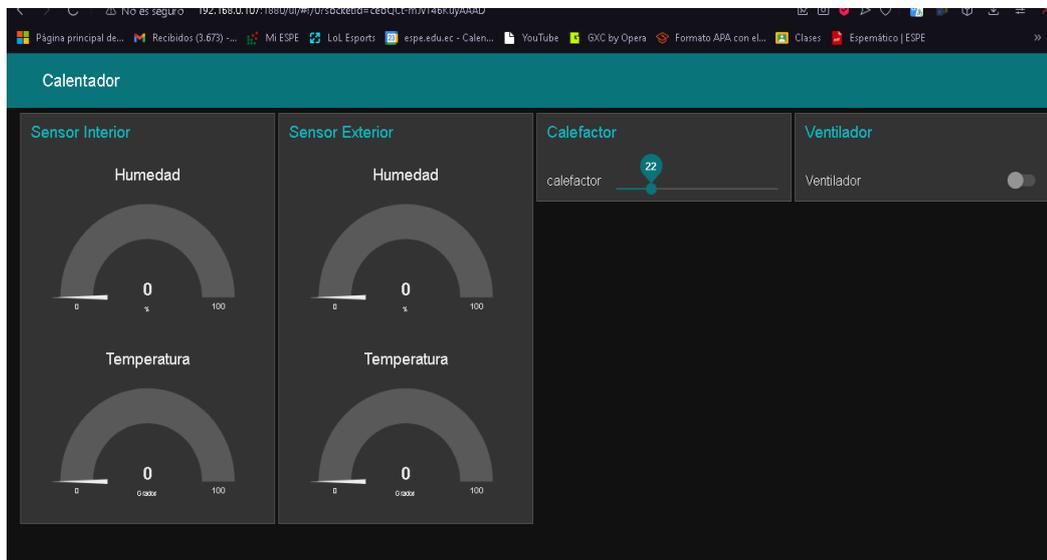


Fig. 62. Interfaz Gráfica que será controlado por el usuario del calefactor.

Se conectó el módulo ESP32 a una fuente de energía (5v) y se observó en la interfaz gráfica de usuario (GUI) de Node-RED cómo se recibían los datos publicados desde el módulo en el suscriptor, que fué Node-RED. Además, Node-RED también actuó como publicador tanto en la sección del slider para el control del calefactor como en la sección del switch para encender y apagar los ventiladores, como se visualiza en la Fig. 63.

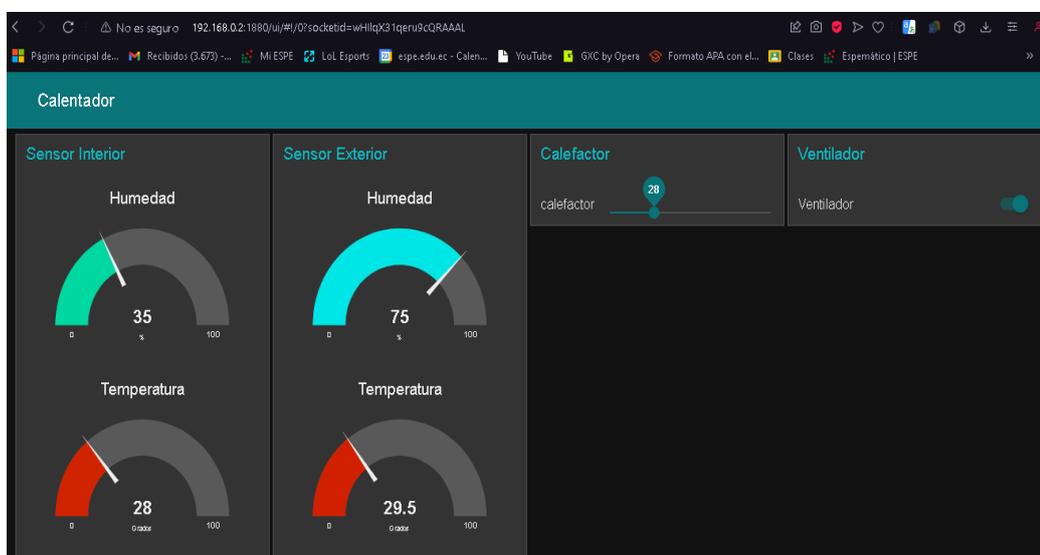


Fig. 63. Interfaz GUI de Node-RED mostrando datos recibidos desde el ESP32.

P. Pruebas operativas y de integración

Se ejecutaron pruebas de integración del calefactor ensamblado con la parte electrónica. El objetivo fue probar que en la parte interior del dispositivo se disponga de un ambiente caliente que recircule a través de los 2 ventiladores hasta alcanzar la temperatura objetivo. Se verificó la conectividad y la interoperabilidad entre los diferentes elementos del sistema, tal como se muestra en la Fig. 64.

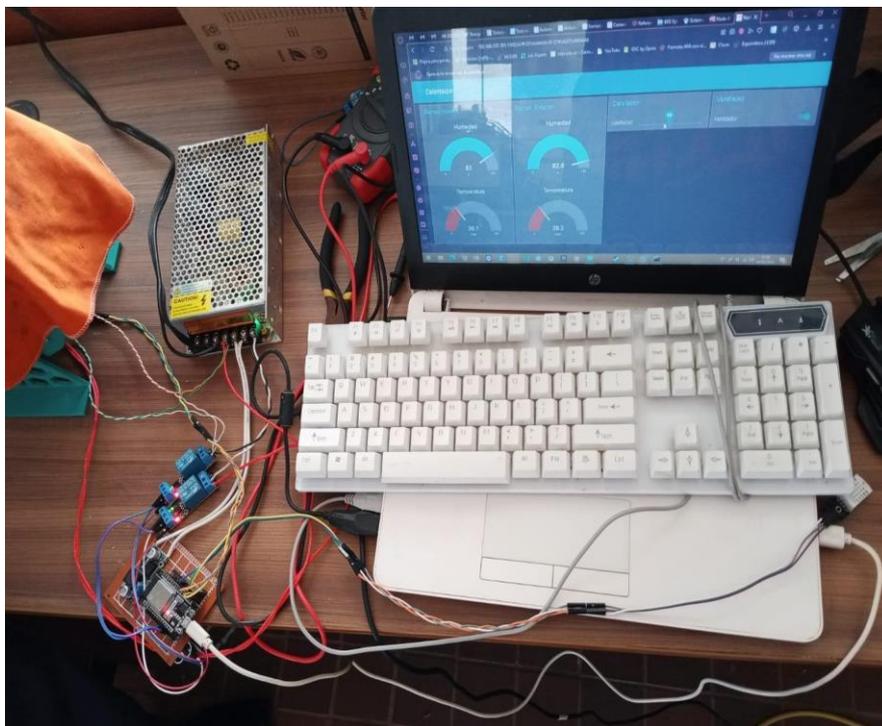


Fig. 64. Pruebas de integración del componente impreso con el circuito electrónico.

Durante estas pruebas, se evaluó específicamente la conexión del módulo ESP32 con punto de acceso Wi-Fi designado. Se verificó la estabilidad y consistencia de esta conexión, asegurando que el módulo se pueda comunicar de manera efectiva con la red Wi-Fi para recibir comandos y transmitir datos.

Además, se evaluó la conexión entre el módulo ESP32 y sus sensores/actuadores con el servidor MQTT a través de Node-RED. Se comprobó que Node-RED pudiera recibir datos del módulo a través del protocolo MQTT y también pueda enviar los comandos de control de forma confiable.

Estas pruebas operativas y de integración jugaron un papel importante en la validación del sistema en conjunto, garantizando que todos los componentes funcionaran de manera coordinada y que el sistema cumpliera con los requisitos de conectividad y comunicación establecidos.

La base operativa del sistema propuesto fue a través de la calefacción de aire dentro del dispositivo armado para que recircule por operación de los ventiladores. Este aire caliente se puede controlar para que alcance la temperatura requerida para sacar la humedad de las fibras PLA (45 grados) y PET (70 grados). Para tener una medida relativa, se implementó un sensor de temperatura y humedad en la parte exterior del dispositivo.

IV. Resultados

Anteriormente, se había desarrollado un diseño alternativo para el calefactor, que constaba de una parte inferior y una superior unidas por unas bisagras para simular una caja con apertura y cierre. Después de evaluar esta propuesta de diseño, se llegó a la conclusión de que no era óptima. Esto se debió a que el tamaño del diseño resultaba demasiado grande para la capacidad de la impresora 3D utilizada, y el tiempo de impresión requerido era excesivamente prolongado. Este diseño se lo puede visualizar en la Fig. 65.

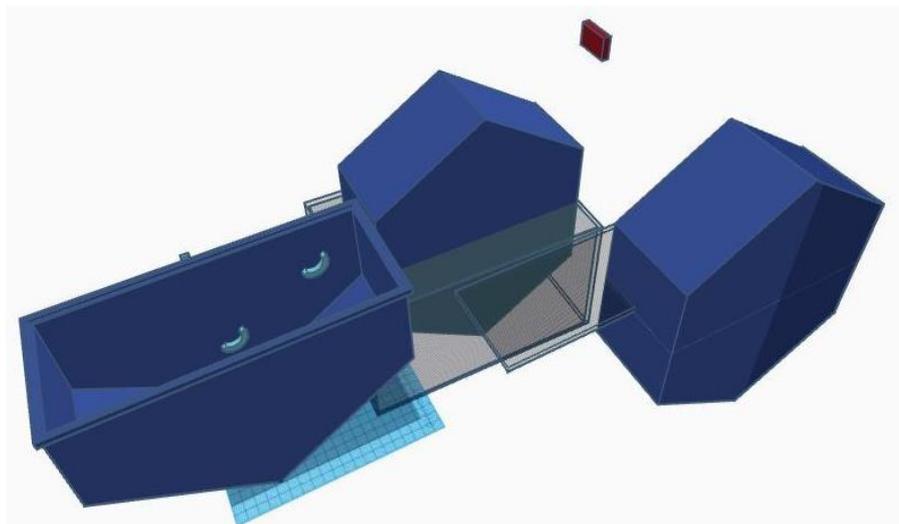


Fig. 65. Diseño Alternativo del Calefactor.

En total, se realizaron 37 impresiones, abarcando una variedad de componentes que conformaron el dispositivo calefactor. Se destacan las piezas de la cubierta, las cuales requerían un tiempo de impresión de 5 horas cada una (base y tapa), estas fueron las partes que más tiempo demandaron para completarse. Además, las piezas del soporte del calefactor, esenciales para su estabilidad, tomaban alrededor de 4 horas cada una. Las piezas triangulares destinadas a formar las caras del calefactor, por su parte, tuvieron un tiempo de impresión de 2 horas y 30 minutos cada una.

Un aspecto para considerar fue el complemento para las caras externas, compuesto por 4 piezas que se imprimieron en dos fases, con dos piezas en cada sesión de impresión. Cada una de estas impresiones tomaron en promedio 3 horas, contribuyendo significativamente al tiempo total de fabricación. Las piezas del soporte interno del calefactor también demandaron un tiempo considerable, aproximadamente 2 horas y 30 minutos cada una. El cilindro interno requería 3 horas para su impresión, mientras que los acoples necesarios consumieron en promedio 2 horas.

Sumando el tiempo de impresión de todas las piezas, se estima que el proceso completo tomó aproximadamente 111 horas y 30 minutos. Considerando que cada impresión se realizó en un día diferente, y teniendo en cuenta el tiempo adicional necesario para la configuración previa y preparación de la impresora, es razonable estimar que el proyecto llevó alrededor de 20 días para completarse, contando solamente los días laborables (aproximadamente 5 semanas). Este período incluyó no solo la impresión en sí misma, sino también la supervisión y el mantenimiento continuo de la impresora durante todo el proceso. Esto se ilustra en la Fig. 66.



Fig. 66. Ensamblaje final de las 37 piezas del calefactor.

El estado de algunos de los rollos de fibra PLA en la Universidad de las Fuerzas Armadas ESPE, sede Santo Domingo, se vio comprometido, ya que presentaron un fenómeno conocido como higroscopía (sensibilidad a la humedad). Al intentar imprimir con este material para realizar pruebas, se evidenció que la fibra estaba cristalizada (perdió su maleabilidad), debido a los cambios de temperatura y al entorno poco óptimo de almacenamiento. En condiciones ambientales normales, en el laboratorio en donde se ejecutaron las actividades de impresión, la temperatura ambiental fue de 25 a 27 grados centígrados con una humedad del 80 a 90%. Estas condiciones desfavorables ocasionaron problemas durante el proceso de impresión. Como resultado, el material se rompió con facilidad, tal como se puede apreciar en la Fig. 67.



Fig. 67. Problemas de cristalización en la fibra PLA durante el proceso de impresión.

La temperatura objetivo para restaurar la maleabilidad de la fibra PLA fue de 45 grados centígrados, sin embargo, en una prueba inicial se estableció a través del slider de su interfaz de Node-RED una temperatura deseada de 34 grados Celsius. El control de la temperatura se realizó conforme a la programación establecida, manteniendo un margen de error de 2.0 grados Celsius.

El proceso de control se desarrolló de la siguiente manera: cuando la temperatura medida por el sensor era inferior a la temperatura deseada más el margen de error (es decir, menor a 36 grados Celsius en este caso), el módulo relé conectado a la resistencia de niquelina permitía el paso de voltaje, encendiendo así la resistencia y elevando la temperatura. Una vez que el sensor detectaba que la temperatura alcanzaba el valor de prueba (34 grados Celsius) y se encontraba dentro del margen de error, el módulo relé bloqueaba el paso de voltaje hacia la niquelina, apagándola y permitiendo que la temperatura comenzará a descender.

Este ciclo de encendido y apagado del módulo relé se mantuvo de manera continua para mantener la temperatura en el rango de prueba, asegurando que oscilara alrededor de los 34 grados Celsius con un margen de error de ± 2.0 grados Celsius.

El sistema funcionó según lo previsto y como se observa en la Fig. 68, el sistema se comportó de acuerdo a lo programado.



Fig. 68. Control de temperatura con margen de error en Node-RED.

En la Fig. 69 se observa la consola de Arduino IDE y cómo, en el interior del dispositivo, los valores de temperatura (se eleva) y humedad (se reduce) hasta alcanzar el valor deseado, y luego, enviar la señal de apagado de la niquelina.

```

14
15 const char* topic_temp_interior = "TemperaturaInterior"; // Tema MQTT para la temperatura
16 const char* topic_hum_interior = "HumedadInterior"; // Tema MQTT para la humedad interior
17 const char* topic_temp_exterior = "TemperaturaExterior"; // Tema MQTT para la temperatura
18 const char* topic_hum_exterior = "HumedadExterior"; // Tema MQTT para la humedad exterior
19 const char* topic_temo_deseada = "temperatura deseada"; // Tema MQTT para la temperatura d

```

Serial Monitor X Output

```

Message (Enter to send message to 'ESP32 Dev Module' on 'COM4')
Ventilador encendido
Calefactor apagado
Temperatura interior: 34.00 °C - Humedad interior: 14.00 %
Temperatura exterior: 30.20 °C - Humedad exterior: 69.00 %
Calefactor encendido
Temperatura interior: 34.00 °C - Humedad interior: 14.00 %
Temperatura exterior: 30.20 °C - Humedad exterior: 69.00 %
Calefactor encendido

```

Fig. 69. Control de temperatura con margen de error en la consola de Arduino IDE.

Después de realizar varias pruebas, se llegó a la conclusión de que calentar el PLA cristalizado a una temperatura comprendida entre 45 y 50 grados Celsius resultaba adecuado para restaurar su maleabilidad y prevenir su rotura durante el proceso de impresión. Para este fin, se colocó el rollo de PLA dentro del calefactor durante aproximadamente 40 minutos para extraer la humedad y mejorar su calidad (para que los ventiladores hagan circular el aire calienten por el interior del calefactor). Esta temperatura permitió que el PLA recuperara su

flexibilidad y manejabilidad, lo que facilitó significativamente el proceso de impresión y disminuyó la probabilidad de que se rompiera durante su uso en la impresora 3D. Este proceso se ilustra en la Fig. [70, 71, 72].



Fig. 70. Restauración de la fibra PLA cristalizada mediante calefacción.

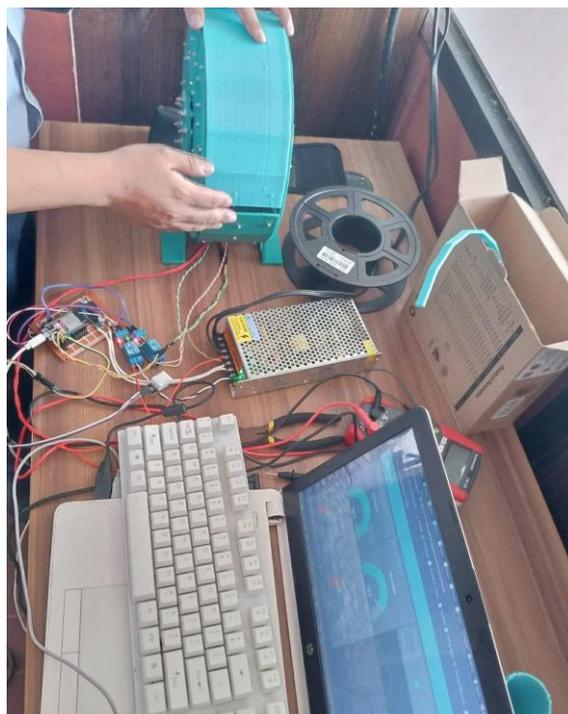


Fig. 71. Restauración de la fibra PLA cristalizada mediante calefacción.

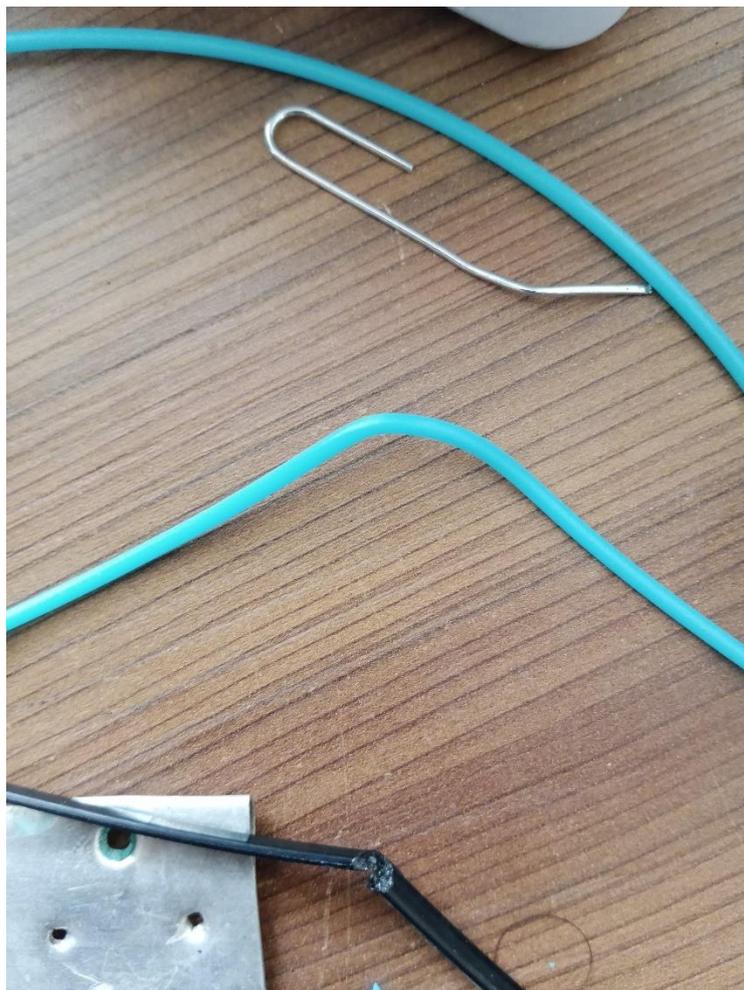


Fig. 72. Fibra PLA restaurada (Se vuelve maleable).

El mismo proceso es requerido para restaurar la maleabilidad de la fibra PET, que requiere una temperatura elevada para extraer su humedad (alrededor de 65 a 70 grados) en un tiempo de 40 a 60 minutos.

El código fuente de la aplicación web en GitHub se puede acceder desde el siguiente enlace: https://github.com/BryanDav1d/Calefactor_De_Fibra_PLA.git

V. Conclusiones y recomendaciones

Q. Conclusiones

En conclusión, todo el diseño del calefactor propuesto es original y modular, es decir, se puede armar y desarmar sin ningún problema. Estas piezas fueron diseñadas con el software de diseño asistido llamado Tinkercad, seleccionado por su facilidad para manipular elementos y figuras muy básicas, que, en conjunto, conforman un sistema completo de calefacción de fibras PLA y PET. Se utilizó la impresora 3D del Departamento de Ciencias de la Computación marca Creality modelo CR10S pro, cuyo trabajo y rendimiento en horas permitió que el dispositivo se concluya con éxito. Durante el proceso de impresión se enfrentó algunos inconvenientes debido a la fragilidad de ciertos rollos de fibra (fibra dañada por la humedad), lo que dificultó la impresión correcta de algunas piezas y prolongó el trabajo. Sin embargo, se logró concluir con el proceso de impresión y ensamblaje de manera adecuada.

Por otro lado, se exploró el uso del software en línea Blynk IoT, una plataforma que permitió la conexión y control de dispositivos IoT a través de dashboards. La programación y configuración en este entorno se realizó de manera efectiva debido a su facilidad de uso y rápida curva de aprendizaje. Sin embargo, es importante destacar que Blynk IoT no es completamente de acceso libre, ya que se requiere una suscripción para desbloquear funciones adicionales como la creación de más dashboards. A pesar de esta limitación, se pudo utilizar Blynk IoT como una primera prueba exitosa, pero con la limitante para añadir más dispositivos (sensores y actuadores).

Es importante destacar que se cumplió con el objetivo principal de este proyecto que fue el diseño y la construcción de un sistema de calefacción destinado a la extracción de humedad de fibras PLA y PET. Mediante la integración de tecnologías 3D e IoT se pudo restaurar, en una etapa inicial, la maleabilidad de una fibra de color Cyan que presentó cristalización. Esto permitió que, al momento de reutilizar esa fibra para imprimir cualquier diseño, ya no se rompiera, garantizando que los estudiantes que usarán la fibra posterior a este trabajo no tengan inconvenientes con una fibra frágil y quebradiza, y que, además, dificulte la impresión de diseños en 3D. Este logro añade valor al proyecto al solucionar un problema concreto relacionado con el almacenamiento en condiciones ambientales no adecuadas de las fibras de impresión.

Con los resultados expuestos, se confirma la hipótesis planteada: se diseñó y construyó un calefactor eficiente y preciso, diseñado y fabricado con tecnologías de impresión 3D y que ofrece un control de temperatura mediante sensores y actuadores que forman una red IoT, con

lo cual, fue posible la regulación térmica y la restauración de la maleabilidad de fibras de impresión de tipo PLA y PET.

R. Recomendaciones

Si se desea utilizar Blynk IoT para el manejo de dispositivos IoT a gran escala, es recomendable optar por la suscripción paga, ya que proporciona una mayor seguridad y funcionalidad. Además, para propósitos de prueba y aprendizaje, Blynk IoT es una excelente opción para estudiantes, ya que su interfaz intuitiva y de fácil uso, permite realizar una variedad de proyectos pequeños con eficacia.

Es crucial que todos los dispositivos estén conectados en la misma red para facilitar su control y permitir la comunicación entre ellos. Para asegurar una conexión estable, se recomienda verificar la conectividad entre la máquina física y virtual mediante el uso de comandos de ping a las direcciones IP correspondientes. Asimismo, el uso de protocolos de publicación y suscripción (pub/sub), como MQTT, es esencial para la transmisión eficiente de datos entre los dispositivos.

Es fundamental utilizar fibra PLA en buen estado al momento de imprimir los diseños en 3D, ya que una fibra deteriorada puede romperse durante el proceso de impresión, lo que conlleva dificultades en el trabajo. Para mantener la fibra PLA en óptimas condiciones, es recomendable almacenarla en lugares con una temperatura adecuada que garantice su integridad y evite posibles deformaciones.

VI. Referencias bibliográficas

- [1] Autodesk. "Tinkercad - Design your own 3D models". Autodesk, Inc. [Online]. Available: <https://www.tinkercad.com/>. [Accessed: February 28, 2024].
- [2] A. Guzmán Cortés, M. Cuatzo Itzcua, y G. A. Balderas Flores, «Diseño y construcción de una extrusora para la obtención de filamento PLA reciclado», *difu100cia*, vol. 16, n.º 3, pp. 126-131, dic. 2022.
- [3] A. J. Hintaw, S. Manickam, S. Karuppayah, M. A. Aladaileh, M. F. Aboalmaaly y S. U. Arfeen Laghari, «A Robust Security Scheme Based on Enhanced Symmetric Algorithm for MQTT in the Internet of Things,» *IEEE Access*, vol. 11, pp. 43019-43040, 2023.
- [4] A. Pisano, «Internet de las cosas,» Universidad de San Andrés. Escuela de Negocios, Buenos Aires, Argentina, 2018.
- [5] A. Pons Abenza, «Diseño de un taller de impresión 3D para la mejora de la inteligencia espacial en alumnos de secundaria y bachillerato,» Tesis de Maestría, 2020.
- [6] B. E. Sánchez Rinza, L. Cabrera Hernández y M. Rossainz López, «Desarrollo e implementación de un sistema de monitoreo de temperatura y humedad utilizando Internet de las Cosas IOT,» en *la Nueva Era*, p. 751, 2022.
- [7] D. Fortuna y H. Zakaria, «Sistem Otomatisasi Untuk Mengukur Kelembaban Suhu Dan pH Tanah Berbasis Android Menggunakan Arduino ESP32 Pada Tanaman Bayam (Studi Kasus : Syahmi Organik),» *OKTAL: Jurnal Ilmu Komputer Dan Sains*, vol. 1, n.º No. 09, p. 6, 01 10 2022.
- [8] D. Hercog, T. Lerher, M. Truntič y O. Težak, «Design and Implementation of ESP32-Based IoT Devices,» *Sensors*, vol. 23, n.º 15, p. 6739, 27 Julio 2023.
- [9] H. Kondo, «Cómo secar filamentos: PLA, ABS y nailon», *All3DP*, 4 de octubre de 2023. <https://all3dp.com/es/2/secar-filamento-pla-abs-nailon/>

- [10] I. HERRARTE LOPEZ y J. S. MARMOLEJO VILLAMIL, «PROTOTIPO DE UNA MÁQUINA EXTRUSORA DE FILAMENTO PARA IMPRESIÓN 3D A PARTIR DEL RECICLADO DE BOTELLAS PET,» 2024.
- [11] J. K. Adarsh, V. T. Sreedevi y D. Thangavelusamy, «Product Review System With BERT for Sentiment Analysis and Implementation of Administrative Privileges on Node-RED,» IEEE Access, vol. 11, pp. 65968-65976, 2023.
- [12] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi y M. Mustaqim, «Internet of Things (IoT) for Next-Generation Smart Systems: A Review of Current Challenges, Future Trends and Prospects for Emerging 5G-IoT Scenarios,» IEEE, vol. 8, pp. 23022 - 23040, 28 Enero 2020.
- [13] M. G. Alberto y U. De Valladolid Escuela de Ingenierías Industriales, «Análisis del proceso de obtención de filamento plástico para impresión 3D utilizando una máquina extrusora de bajo coste», Universidad de Valladolid, 2022.
<https://uvadoc.uva.es/handle/10324/54423>
- [14] M. R. Berman, "3-D Printing: The New Industrial Revolution," Business Horizons, vol. 55, no. 2, pp. 155-162, 2012.
- [15] P. Ibarra, J. Ortega y J. León, «Diseño de un sistema embobinador y secador de filamentos para impresión 3D,» 2022.
- [16] P. Karrupusamy, A Sensor based IoT Monitoring System for Electrical Devices using Blynk framework, Journal of Electronics and Informatics, vol. 2, nº 3, pp. 182-187, 27 Agosto 2020.
- [17] R. B. HUILLCA ACHAHUI y R. G. HUILLCA ACHAHUI, DISEÑO Y FABRICACIÓN DE UN MÓDULO DE CASTING PARA JOYERÍA (INNOVACION TECNOLÓGICA),2022

- [18] T. Belmonte Rios, «Diseño de una aplicación para la gestión de una estación meteorológica mediante protocolo de transporte de telemetría por mensaje de colas (MQTT) sobre módulo de radio frecuencia ESP32,» 2022.