



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL**

CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN

**“SISTEMA DE MONITOREO Y ANÁLISIS DE
PARÁMETROS DE MÁQUINAS DE ACONDICIONAMIENTO
DE AIRE”**

Trabajo de titulación previo a la obtención del
Título de Ingeniería Electrónica

AUTORES: JOSÉ GUILLERMO NÚÑEZ RAMÍREZ
ROSENDO REYNALDO LÓPEZ PINCAY

TUTOR: ING. VICENTE PEÑARANDA

GUAYAQUIL – ECUADOR

2024 - 2025


CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, José Guillermo Núñez Ramírez con documento de identificación N° 0941476772 y Rosendo Reynaldo López Pincay con documento de identificación N° 0918661240; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

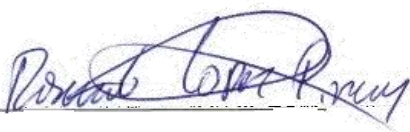
Guayaquil, 14 de febrero del año 2025

Atentamente



(f) José Guillermo Núñez Ramírez

C.I: N° 0941476772



(f) Rosendo Reynaldo López Pincay

C.I: N° 0918661240

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo ING.VICENTE PEÑARANDA con documento de identificación N° 0916113426, docente de la Universidad Politécnica Salesiana , declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: “SISTEMA DE MONITOREO Y ANÁLISIS DE PARÁMETROS DE MÁQUINAS DE ACONDICIONAMIENTO DE AIRE”, realizado por los señores José Guillermo Núñez Ramírez con documento de identificación N° 0941476772 , y Rosendo Reynaldo López Pincay con documento de identificación N° 0918661240 , obteniendo como resultado final el trabajo de titulación bajo la opción Ingeniero Electrónico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 14 de febrero del año 2025

Atentamente,



(f) Ing. Vicente Avelino Peñaranda

C.I 0916113426

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

José Guillermo Núñez Ramírez con documento de identificación N° 0941476772 y Rosendo Reynaldo López Pincay con documento de identificación N° 0918661240, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación: “SISTEMA DE MONITOREO Y ANÁLISIS DE PARÁMETROS DE MÁQUINAS DE ACONDICIONAMIENTO DE AIRE”, el cual ha sido desarrollado para optar por el título de: Título de Ingeniería Electrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 14 de febrero del año 2025

Atentamente,



(f) José Guillermo Núñez Ramírez

C.I: N° 0941476772



(f) Rosendo Reynaldo López Pincay

C.I: N° 0918661240

DEDICATORIA

José Guillermo Núñez Ramírez

Dedico este trabajo a Dios, a mis padres, a mi hermano, a mi familia y a todas las personas que estuvieron allí apoyándome en el camino dándome su sabiduría y consejos para poder avanzar y no desistir.

Rosendo Reynaldo López Pincay

Dedico este trabajo a mi madre, mi esposa y mi hija las cuales siempre estuvieron apoyándome a lo largo de camino de manera incondicional, ayudándome a no desistir de mis sueños.

AGRADECIMIENTO

José Guillermo Núñez Ramírez

En primer lugar, agradezco a Dios por permitirme este logro académico y darme siempre capacidad para avanzar, a mis padres y hermanos por ser y estar en cada momento de mi vida, a mis abuelos y a mi tío por su apoyo moral constante, a mi enamorada Alejandra por sus ánimos cuando los necesite, a mis docentes por su guía académica compartiendo sus conocimientos siempre de forma acertada y a mi gato rayito por acompañarme en mis largas jornadas de estudio.

Rosendo Reynaldo López Pincay

A mi padre celestial Dios, a mi madre por ser mi guía y apoyo moral, a mi hija por ser el pilar de mis decisiones y de mi vida, a mi esposa por su constante respaldo y cariño, les agradezco por todo lo que han hecho y todo el tiempo que han estado a mi lado dándome todo su apoyo y cariño para no rendirme y lograr salir adelante.

Resumen

Este trabajo de titulación presenta el desarrollo de un prototipo de sistema de monitoreo para equipos de aire acondicionado, con el objetivo de ofrecer información precisa en tiempo real sobre su funcionamiento, esta iniciativa busca favorecer a la comunidad de la Universidad Politécnica Salesiana. El sistema permite analizar y supervisar variaciones en presión, humedad y temperatura, facilitando la toma de decisiones para optimizar el rendimiento, reduciendo así los riesgos en los equipos. Además, se busca que este sistema sirva como recurso académico para futuras investigaciones, mejorando la comprensión del funcionamiento de las unidades de aire mediante la integración de tecnologías avanzadas.

El prototipo está basado en una plataforma IoT, una tecnología cada vez más popular en los hogares y empresas por su capacidad para ofrecer soluciones eficientes. La Raspberry Pi, que es el núcleo del sistema, permite una programación eficaz y una gestión dinámica del consumo energético. La conexión Wi-Fi, permite al sistema transmitir datos precisos a tiempo real. El sistema HVAC, que incluye ventilación, refrigeración, calefacción y control de humedad, es monitoreado mediante sensores avanzados que permiten medir variables como: temperatura, humedad interna y externa, sin la necesidad de una exposición directa de los trabajadores.

El proceso de configuración incluye la instalación de una tarjeta microSD con el sistema operativo en la Raspberry Pi y la activación de los sensores mediante el bus I2C. Además, se utilizan plataformas como Thingspeak y Thingview para visualizar y analizar las variables monitoreadas, lo que mejora la supervisión remota de los equipos de climatización.

PALABRAS CLAVES: Raspberry Pi, Sensor, Thing Speak.

Abstract

This degree work presents the development of a prototype monitoring system for air conditioning equipment, with the aim of offering precise information in real time about its operation. This initiative seeks to benefit the community of the Salesiana Polytechnic University. The system allows you to analyze and monitor pressure, humidity and temperature variations, facilitating decision-making to optimize performance, thus reducing equipment risks. In addition, it is intended that this system serve as an academic resource for future research, improving the understanding of the operation of air units through the integration of advanced technologies.

The prototype is based on an IoT platform, a technology that is increasingly popular in homes and businesses due to its ability to offer efficient solutions. The Raspberry Pi, which is the core of the system, allows for efficient programming and dynamic management of power consumption. The Wi-Fi connection allows the system to transmit precise data in real time. The HVAC system, which includes ventilation, refrigeration, heating and humidity control, is monitored using advanced sensors that allow variables such as: temperature, and internal and external humidity to be measured, without the need for direct exposure of workers.

The setup process includes installing a microSD card with the operating system on the Raspberry Pi and activating the sensors via the 12C bus. In addition, platforms such as Thingspeak and Thingview are used to visualize and analyze the monitored variables, which improves the remote supervision of air conditioning equipment.

KEY WORDS: Raspberry Pi, Sensor, Thing Speak.

Índice de contenido

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN	i
CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN	ii
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
Resumen	vi
I. INTRODUCCIÓN	1
II. PROBLEMA	2
2.1 Justificación	3
III. OBJETIVOS	4
3.1 Objetivo General.....	4
3.2 Objetivos Específicos.....	4
IV. FUNDAMENTO TEÓRICO	4
4.1 Sistemas HVAC	4
4.1.1 <i>Dispositivos con uso IoT</i>	5
4.1.2 <i>Monitoreo y control remoto del sistema</i>	5
4.1.3 <i>Redes inteligentes habilitadas para IoT</i>	6
4.2 Sistemas de climatización	6
4.2.1 <i>Sensores de IoT</i>	6
4.2.2 <i>Elementos necesarios para la climatización</i>	8
4.2.3 <i>Plataforma IoT software y conectividad</i>	9
4.3 Control climático de precisión	10
4.4 Programador automático de aire acondicionado	10
4.5 Arquitectura del sistema de control de aire acondicionado	10
4.6 Sincronización de los dispositivos	12
4.6.1 <i>ThingSpeak</i>	12
4.6.2 <i>Raspberry Pi</i>	12
4.6.3 <i>Phyton</i>	13
4.7 Módulo Wi-Fi protocolo de comunicación.....	13
4.8 Redes de comunicación.....	13
V. MARCO METODOLÓGICO	15
5.1. Descripción del diseño del sistema de monitoreo.....	15
5.2. CODIGO SENSOR DHT22, BMP280 Y DE USO SIMULTANEO CON THINGSPEAK	16
5.3. Pasos para utilizar la Raspberry	19

5.4. Pasos para utilizar los sensores	19
5.5. Pasos para usar Thingspeak y Thingview	20
5.6. Comunicación entre los dispositivos	20
5.7. Presentación de datos de la plataforma ThingSpeak.....	21
VI. RESULTADOS	31
VII. CRONOGRAMA	41
VIII. PRESUPUESTO	45
IX. CONCLUSIONES.....	46
X. RECOMENDACIONES	47
XI. REFERENCIAS.....	48
XII. ANEXOS.....	50

Índice de figuras

Figura 1 Sistema HVAC	5
Figura 2 Ciclo de refrigeración	8
Figura 3 Diagrama de conexión de los sensores a la placa Raspberry Pi.....	15
Figura 4 Conexiones de los dispositivos usados en las pruebas	22
Figura 5 Lectura de temperatura del sensor DHT22.....	23
Figura 6 Lectura de humedad del sensor DHT22	23
Figura 7 Consola del programa thonny usando el sensor DHT22.....	24
Figura 8 Lectura de temperatura del sensor BMP280	25
Figura 9 Lectura de presión en bares del sensor BMP280	25
Figura 10 Lectura de presión en PSI del sensor BMP280	26
Figura 11 Consola del programa thonny usando el sensor BMP280.....	27
Figura 12 Lectura de temperatura de ambos sensores en simultaneo.....	28
Figura 13 Lectura de presión en bares de ambos sensores en simultaneo	28
Figura 14 Lectura de presión en PSI de ambos sensores en simultaneo.....	29
Figura 15 Lectura de humedad de ambos sensores en simultaneo	29
Figura 16 Consola del programa thonny usando ambos sensores a la vez	30
Figura 17 Valores de temperatura mostrados en ThingView Free por el sensor BMP280	31
Figura 18 Valores de presión mostrados en ThingView Free por el sensor BMP280 en Bares	32
Figura 19 Valores de presión mostrados en ThingView Free por el sensor BMP280 en PSI	33
Figura 20 Valores de temperatura mostrados en ThingView Free por el sensor DHT22	34
Figura 21 Valores de humedad mostrados en ThingView Free por el sensor DHT22.....	35
Figura 22 Valores de temperatura mostrados en ThingView Free usando ambos sensores...36	
Figura 23 Valores de presión mostrados en ThingView Free usando ambos sensores en Bares	37
Figura 24 Valores de presión mostrados en ThingView Free usando ambos sensores en PSI	38
Figura 25 Valores de humedad mostrados en ThingView Free usando ambos sensores	39
Figura 26 Raspberry Pi 4 Modelo B en funcionamiento	50
Figura 27 Contenedor de nitrógeno, nanómetro y tubo de cobre utilizados para simular el ambiente de un compresor	51
Figura 28 Especificaciones técnicas de la unidad de aire acondicionado.....	52
Figura 29 Estado de la unidad de aire acondicionado	53

Índice de tablas

Tabla 1 Tipos de sensores IoT	7
Tabla 2 Partes del ciclo de refrigeración.....	9
Tabla 3 Componentes y sus interacciones	11
Tabla 4 Tabla de presupuestos para el proyecto	45

I. INTRODUCCIÓN

La variación climática a nivel global, provoca que muchas personas requieran de sistemas de climatización en sus edificaciones, esto hace que exista un mayor consumo del recurso energético en comparación con años anteriores, esto a su vez impulsa la creación de propuestas tecnológicas que disminuyan el consumo de la energía, como es el caso del diseño e implementación de sistemas de monitoreo basados en las herramientas IoT, los cuales permiten la conectividad entre dispositivos de máquinas de aire acondicionado ya sea para el sector residencial o industrial.

Es absolutamente necesario mejorar el desempeño de las unidades de climatización, una forma de hacerlo es a través de la tecnología IoT, donde el usuario puede realizar una gestión inteligente del consumo de la energía y evitar daños de los equipos durante su uso, además de poder prevenir daños provocados por las variaciones de voltajes y brindar un mantenimiento efectivo a los mismos.

En tanto, las condiciones climáticas de la ciudad de Guayaquil no son la mejores debido a la ola de calor que se presenta a diario, esto motivó a la realización del presente trabajo de titulación, el mismo que consiste en diseñar un prototipo para el monitoreo de los equipos de aire acondicionado mediante una tarjeta Raspberry Pi, el cual se acopla a la funcionalidad de las plataformas IoT cuyo propósito es permitir el análisis de la información recogida y almacenada en la nube mediante los sensores de medición y temperatura, lo cual brinda un margen de soluciones, al tener un rápido procesamiento de algoritmos de análisis y al estar conectados a la red wifi, se puede intercambiar la información durante el funcionamiento de las unidades de aire acondicionado.

II. PROBLEMA

A nivel internacional el consumo de recursos no renovables, es un tema de suma prioridad para instituciones públicas y del sector empresarial que cuentan con sistemas de climatización debido a que se utilizan 85 millones de KW/h al año lo cual implica que existe una tendencia marcada de crecimiento de un mayor uso de equipos de aire acondicionado y extractores de aire, los mismo que se pueden acoplar a los sistemas de monitoreo durante su funcionamiento con el fin de mantener de forma fiable su uso de acuerdo a los requerimientos de temperatura, humedad y calidad del aire, independientemente de las fluctuaciones de estos parámetros en el ambiente (zonas adyacentes, exteriores) (Danahe, 2020).

De igual forma a nivel nacional entre el 2010 hasta el 2023 se incrementó de 1.105KWh a 1.470KWh por habitante esto significa que el 73% de este recurso se destina principalmente a los sistemas de climatización de los hogares y lugares del sector ejecutivo, empresarial. Por lo cual es muy importante que estos equipamientos de aires acondicionados deben contar con un sistema de monitoreo que contribuyan con el medio ambiente de acuerdo a las normas internacionales, deben operar a una temperatura de 22.5 °C (Fernandez, 2019).

De igual forma en Guayaquil en tiempos actuales el cambio climático elevó considerablemente las temperaturas ambientes de las residencias y lugares de trabajo esto ocasionó un incremento del uso de unidades de aire acondicionado por tiempos prolongados, ocasionando un mayor consumo de energía, por lo tanto, al no contar con un sistemas de monitoreo de los parámetros de funcionamiento de los equipos no se puede hacer un seguimiento de la operación de las unidades de aire y así evitar posibles daños inesperados.

Para dar forma a la presente propuesta, se diseña un prototipo de sistema de monitoreo que monitoree los sistemas de climatización residenciales utilizando tecnología IoT, donde los

usuarios puedan conectar sus dispositivos móviles con la red que están vinculados las unidades de acondicionamiento de aire.

Por lo cual es necesario desarrollar un software que se acople a la tecnología IoT, a la conectividad de los dispositivos tales como laptop, ordenadores, iPad, tablet y smartphones, con los equipos de aire acondicionado, siguiendo las etapas de diseño, construcción e implementación del prototipo, a fin de lograr un gran alcance de control y monitorización al haber una capa de procesamiento de datos, los cuales son cifrados y autenticados.

En el repositorio de la Universidad Politécnica Salesiana no se ha observado ningún trabajo de titulación el cual cubra el monitoreo y análisis de las distintas variables y parámetros presentes en el funcionamiento de las unidades de acondicionamiento de aire doméstico. Lo cual motivó a la búsqueda de información, en medios académicos nacionales e internacionales para adoptar una postura clara con fines prácticos y profundizar criterios que permitan el diseño y la implementación del sistema de monitoreo.

2.1 Justificación

El presente trabajo de titulación pretende ser un insumo académico que permita conocer el funcionamiento de sistemas de monitoreo en equipos de aire acondicionado para futuros trabajos de investigación. Partiendo del conocimiento académico, que brinde oportunidad de mejorar el entendimiento sobre el funcionamiento de las unidades de acondicionamiento de aire al hacer una supervisión efectiva del sistema de monitoreo en los equipos de climatización.

III. OBJETIVOS

3.1 Objetivo General

- ❖ Desarrollar un prototipo de sistema para supervisar los parámetros operativos de temperatura, humedad y presión de una unidad de acondicionamiento de aire doméstica.

3.2 Objetivos Específicos

- ❖ Realizar la programación en el Raspberry Pi para la lectura de las variables.
- ❖ Realizar la conexión de los sensores con el Raspberry Pi.
- ❖ Diseñar un programa de monitoreo que pueda integrarse con una aplicación IoT.

IV. FUNDAMENTO TEÓRICO

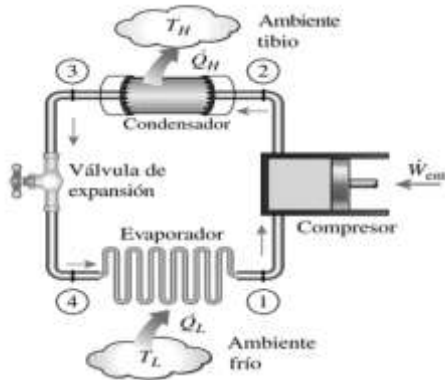
4.1 Sistemas HVAC

Este sistema HVAC (Heating, Ventilation, and Air Conditioning) se encuentra compuesto por los siguientes elementos: ventilación, refrigeración, calefacción y control de humedad. Se utiliza esencialmente para el monitoreo y los respectivos análisis de cada sistema, adicional a esto su implementación de sensores, con tecnología avanzada en la recopilación de datos, los mismos que permiten estudiar los parámetros de los equipos de aire.

Se utiliza un sistema HVAC de refrigeración que posee cuatro componentes básicos: líquido refrigerante, compresor de vapor del refrigerante, condensador de vapor, válvula de expansión y una evaporadora (COMAS, SIMANCAS, VÉLEZ, BERNAL, & PERCIA, 2018), tal como se observa en la figura 1.

Figura 1

Sistema HVAC



Nota. Funcionamiento de un sistema HVAC (COMAS-GONZÁLEZ, SIMANCAS-GARCÍA, VÉLEZ-ZAPATA, BERNAL Rueda, & PERCIA Vásquez, 2018)

4.1.1 Dispositivos con uso IoT

Estos dispositivos con uso IoT (Internet de las cosas) adaptados a los sistemas de monitoreo y análisis de parámetros de máquinas de acondicionamiento de aire, tienen una tecnología inteligente que se conecta al internet, tales como: wifi, Lorawan, ethernet o redes celulares (4G/5G), permitiendo monitorear en tiempo real el funcionamiento de los equipos de climatización.

Por ello, al instalar estos sensores en los aires acondicionados se pueden medir los siguientes parámetros como: temperatura y humedad (DHT11, DHT22), presión, flujo de aire, consumo energético, vibración y nivel de refrigerante.

4.1.2 Monitoreo y control remoto del sistema

En el monitoreo y control con los sistemas de IoT permite el análisis de los valores de temperatura y humedad, brindando parámetros que permite una identificación de fallos y aumentando la vida útil de los equipos de climatización.

Al configurar la plataforma del sistema de monitoreo, el software que reciba y visualice los datos de cada sensor, transmite alertas al acceso remoto de la interfaz y los operadores realizan ajustes en tiempo real a las maquinas sin necesidad de estar físicamente en múltiples unidades de aire acondicionado.

4.1.3 Redes inteligentes habilitadas para IoT

Las redes de inteligentes habilitadas para IoT (Smart Grids o Smaert Networks) son sistemas de comunicación y control alcanzados que permiten la integración eficiente de dispositivos conectados a internet en una infraestructura de red (Saleem, Crespi, Husain, & Copeland, 2019). Los mismos cumplen con la función de interconectarse con dispositivos, sensores y actuadores en una red centralizada o distribuida.

4.2 Sistemas de climatización

La tecnología que utiliza el sistema de climatización es la encargada de regular, controlar y mantener las condiciones ambientales dentro de un espacio cerrado, por ello, uno de los componentes clave de este sistema son las unidades de tratamiento de aire (UTA). Estas unidades procesan el aire y lo acondicionan antes de enviarlo, suelen incluir filtros e intercambiadores de calor (enfriadores o calefaccionadores), humificadores y deshumidificadores.

Este sistema es fundamental en varios entornos, manteniendo un nivel de confort termiónico y calidad del aire constante.

4.2.1 Sensores de IoT

Los sensores de IoT recogen datos físicos y lo envían de forma digital, los mismos son procesados a través de redes como wifi. La información es recopilada de forma continua en el sistema de procesamiento para ser analizada, permitiendo el aprendizaje automático para

identificar patrones y comportamientos inusuales en los equipos, por ello, se mencionan los siguientes:

Tabla 1
Tipos de sensores IoT

Sensores de temperatura	Sensores de humedad	Sensores de presión	Sensores de flujo de aire
Termistores	Sensores capacitivos	Sensores piezorresistivos	Sensores de anemómetro
Termopares	Sensores resistivos	Sensores de presión capacitivos	Sensores de presión diferencial
Sensores infrarrojos			

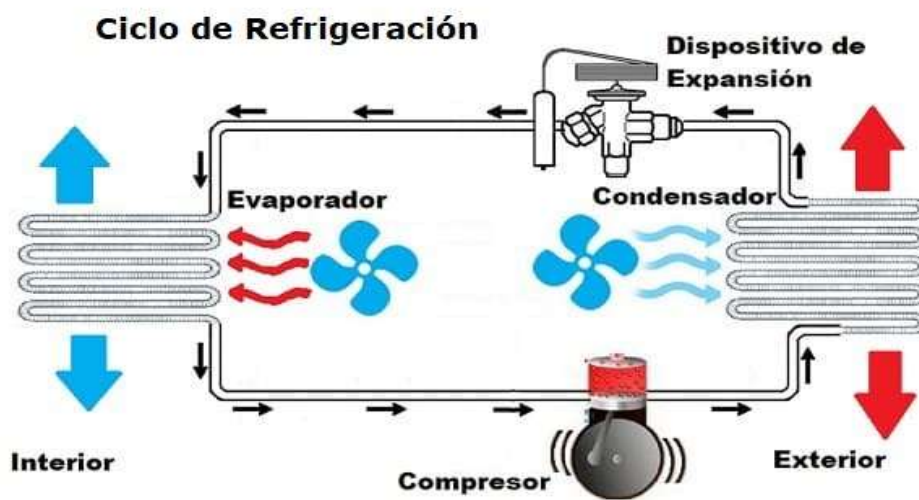
Sensores de vibración	Sensores de calidad de aire	Sensores de consumo energético	Aplicación de sensores Iot
Acelerómetros	Sensores de CO2	Medidores de corrientes	Monitoreo remoto
Sensores piezoeléctricos	Sensores de partículas	Sensores de potencia	Mantenimiento predictivo
			Optimización y mejora de los equipos de climatización

Nota. Se muestran todos los tipos de sensores que funcionan con tecnología IoT (AlfaIoT, 2023)

4.2.2 Elementos necesarios para la climatización

La climatización tiene procesos termodinámicos utilizados para transferir de un espacio a otro con el fin de mantener una temperatura deseada. Uno de los elementos esenciales es la transferencia del refrigerante a través una serie de etapas, incluyendo la evaporación, compresión, condensación y expansión (PULLA, 2024). Observar figura 2:

Figura 2
Ciclo de refrigeración



Nota. Se muestra el funcionamiento de un sistema de refrigeración con todas sus partes (Refri-Climas, 2024)

Tabla 2
Partes del ciclo de refrigeración

Compresor	Condensador	Válvula de expansión	Evaporador
Aspira el refrigerante en estado gaseoso.	Extrae el calor del refrigerante gaseoso y lo condensa a estado líquido.	Regula el paso del refrigerante líquido del condensador hacia el evaporador.	Evapora el calor y crea el efecto de refrigerante.

Nota: Funciones de cada parte en el ciclo de refrigeración (Academy, 2023)

Este ciclo es fundamental en los sistemas de climatización, solo así se puede completar el proceso de extraer el calor de una habitación y mantener en óptimas condiciones la temperatura y la humedad.

4.2.3 *Plataforma IoT software y conectividad*

Como tecnologías y herramientas de la capa de percepción se revisaron: SBC como Raspberry pi, Arduino, Node-MCU, Dev-Kit, placas genéricas basadas en SOC ESP8266 y ESP32, los lenguajes de programación de dichas placas como el utilizado por el IDE de Arduino basado en el lenguaje de alto nivel, el cual es muy similar a C++, además el IDE de Arduino es compatible y admite muchas de las placas mencionadas. Otro lenguaje de programación pertinente para este ámbito es Python.

En relación con los componentes necesarios para establecer la conectividad a internet y con otros dispositivos cercanos, se revisaron tecnologías de redes de área personal inalámbricas, redes área local inalámbricas, las redes inalámbricas de área amplia y de baja potencia, entre otras. Aunque como ejemplo solo se han mencionado tecnologías inalámbricas,

no quiere decir que las tecnologías cableadas sean excluidas del IoT (D. Navarro Pino, J. S. Badillo Rincón, M. D. Portillo Padilla, & S. E. Pineda Aguilera, 2024).

Los estándares inalámbricos existente en el mercado son abundantes, aunque dirigidos a usuarios con necesidades específicas, y que son seleccionadas basadas en criterios como cobertura, alcance, tasa de transferencia o ancho de banda, consumo energético, banda de frecuencia, costo y disponibilidad embebida en SBC, entre los cuales se identifican en la revisión los estándares Wi-Fi (802.11n/ac/ax/ad), Bluetooth y otros estándares 802.15 como ZBee, LoRaWAN, Sigfox, 4G, LTE, 5G, entre otros.

4.3 Control climático de precisión

El control climático de precisión (PICC) es una innovación que permite un control extremadamente detallado de las condiciones ambientales dentro de un edificio. Utilizando sensores distribuidos y algoritmos avanzados, los sistemas PICC pueden mantener temperaturas y niveles de humedad constantes en diferentes zonas del edificio. Esto es especialmente beneficioso para entornos comerciales donde las condiciones precisas son críticas, como laboratorios, hospitales y centros de datos (Koolnova, 2024).

4.4 Programador automático de aire acondicionado

Este programador automático es un sistema inteligente para la gestión y control remoto de los equipos de climatización a través de la IoT, basados en parámetros del entorno o las preferencias del usuario. Los actuadores IoT son dispositivos que permites modificar los parámetros de forma automática, utilizando los siguientes sensores: termostato inteligente, controlador de velocidad del ventilador y el control de encendido y apagado.

4.5 Arquitectura del sistema de control de aire acondicionado

La arquitectura del sistema de control de aire acondicionado se basa en seis capas y se presenta como un modelo integral para la gestión de los sistemas de climatización; a través de

la primera capa de sensores y dispositivos encargados de recopilar datos del entorno, la segunda capa es la de red, que maneja la transmisión de datos que van a las aplicaciones o plataformas basadas en la nube, la tercera capa es la gestión y análisis de datos, procesados en tiempo real, la cuarta capa es donde se desarrolla la aplicación, la cual utiliza los datos para ofrecer servicios a los usuarios finales, la quinta capa de seguridad con sistema IoT, protege todos los datos recopilados, por último la sexta capa de servicios, proporciona funcionalidades adicionales de mejora y soporte. A continuación, observar la tabla 3.

Tabla 3
Componentes y sus interacciones

Capa #1 Sensores y dispositivos	Capa #2 Red	Capa #3 Gestión y análisis de datos
Monitoreas el ambiente. Monitorear el estado de los componentes del a/c.	Trasmisión de datos. Redes de comunicación.	Procesamiento de datos en tiempo real. Análisis predictivo. Optimización del rendimiento.
Capa #4 Desarrollo de la aplicación	Capa #5 Seguridad IoT	Capa #6 Servicios
Interfaz de usuario Control remoto. Notificaciones y alertas.	Protección de datos Autenticidad y autorización. Seguridad contra ciberataques.	Mantenimiento predictivo. Actualizaciones remotas. Optimización continua.

Nota: Distintas capas y aportes en la arquitectura de un sistema de control (UNIR, 2023)

Este modelo de arquitectura de seis capas proporciona una estructura flexible y escalable para implementar un sistema IoT eficiente para el control y monitorio de los equipos de aire acondicionado.

4.6 Sincronización de los dispositivos

Los sistemas distribuidos se emplean ampliamente en entornos contemporáneos como Internet, en la nube y bases de datos. Un sistema distribuido se refiere a un sistema que comprende múltiples componentes informáticos que operan cohesivamente como una entidad unificada para sus usuarios. Estos dispositivos están dispersos geográficamente, funcionan simultáneamente y mantienen la sincronización a través del paso de mensajes (Ehtisham UI Haque, et al., 2024). Los sistemas distribuidos en el contexto de IoT, deben interactuar de manera autónoma. Necesitan un tercero de confianza para el mantenimiento del sistema. Sin embargo, si el operador del sistema se viera comprometido, la confiabilidad del sistema también se vería comprometida.

4.6.1 *ThingSpeak*

ThingSpeak es una plataforma diseñada para el Internet de las Cosas (IoT) que facilita la recopilación y almacenamiento de datos de sensores en la nube, permitiendo el desarrollo aplicaciones inteligentes que aprovechan esta información. Usando ThinkSpeak, también puedes analizar y visualizar tus datos mediante MATLAB, e incluso tomar decisiones automatizadas en función de ellos. Es compatible con dispositivos como Arduino y Raspberry Pi. (Arduino, 2018)

4.6.2 *Raspberry Pi*

Es una tarjeta de desarrollo con una similitud a una computadora completamente funcional en un paquete pequeño y de bajo costo. Las aplicaciones que se encuentran son para navegación por la web, interés sobre su programación o crear sus propios circuitos. Además de

dispositivos físicos, una Raspberry Pi puede hacer cualquier cosa que pueda hacer una computadora más grande, pero consume menos energía.

4.6.3 Python

Python es un lenguaje de programación muy versátil y sencillo de usar, ideal tanto para quienes están empezando como para desarrolladores con experiencia. Su sintaxis clara y sencilla hace que escribir y entender código sea mucho más fácil e intuitivo.

Gracias a esta facilidad, Python se usa en una gran variedad de proyectos, desde pequeñas automatizaciones hasta aplicaciones avanzadas como inteligencia artificial, análisis de datos y desarrollo web. (Frisoli, 2024)

4.7 Módulo Wi-Fi protocolo de comunicación

El Internet de las Cosas (IoT) es la conexión de dispositivos físicos que pueden enviar y recibir datos a través de redes inalámbricas con mínima intervención humana. Esto es posible porque se han integrado sistemas informáticos en todo tipo de objetos, desde electrodomésticos hasta maquinaria industrial.

En esencia, los dispositivos IoT están en constante comunicación, recopilando, enviando y analizando información en un ciclo continuo. Dependiendo de la tecnología utilizada, estos datos pueden ser procesados de inmediato por personas o por sistemas de inteligencia artificial y aprendizaje automático (IA/ML), o almacenados para su análisis posterior. (Hat, 2023)

4.8 Redes de comunicación

Estos dispositivos IoT generan grandes cantidades de datos y la infraestructura de red tradicional puede verse abrumada. Para proporcionar una solución escalable para administrar y controlar el flujo de datos en la red, se ha introducido la integración de redes definidas por software (SDN). Con SDN, los administradores de red pueden administrar y configurar toda la

red desde una única ubicación, lo que reduce la complejidad de administrar varios dispositivos. (Shafiq, Rahman, Ahmed, Mahmud, & Hosen, 2024)

Por lo tanto, la arquitectura SDN para IoT (SDN-IoT) se ha desarrollado para una administración eficiente debido al crecimiento global de IoT y su complejidad administrativa. Al establecer el método de implementación de software en la red IoT, SDN regula, controla y actualiza los comportamientos de la red de forma dinámica.

En el mundo digital moderno, las redes de comunicación son el tejido que mantiene todo unido. La ley de Metcalfe afirma que el valor de una red de comunicación es proporcional a la cantidad de dispositivos conectados. Como resultado, SDN ha ganado prominencia como una tecnología de servicio de red debido a su programabilidad y mantenimiento flexible. SDN ofrece una estructura en capas que involucra tres planos, a saber, los planos de datos, control y aplicación, donde cada plano opera de forma independiente. La separación de los planos de datos y control permite la administración de la red en tiempo de ejecución, el control del tráfico, la expansión de la red y la programabilidad flexible del sistema. (Shafiq, Rahman, Ahmed, Mahmud, & Hosen, 2024)

V. MARCO METODOLÓGICO

5.1. Descripción del diseño del sistema de monitoreo

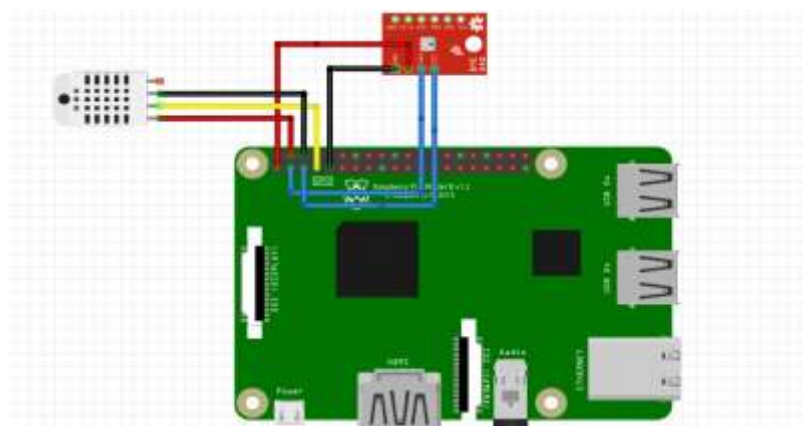
En el diseño del sistema de monitoreo se expone la descripción del proceso en su realización partiendo de la integración del hardware apropiado como es el caso de la instalación de una placa Raspberry Pi para brindar una óptima funcionalidad a la aplicación web durante el desarrollo del prototipo el cual se programa un código de lenguaje, para obtener datos de comunicación en los dispositivos conectados. Para el uso de el sensor DHT22 se instaló una librería con el comando: `sudo pip install Adafruit_DHT`, mientras que para el sensor BMP280 se usó `sudo pip install pimoroni-bme280 smbus`.

Arquitectura del sistema de monitoreo

Esquema de conexión de ambos sensores, se usan un DHT22 y un BMP280, en una Raspberry PI 4. El cual permite integrarse al sistema del aire acondicionado con el módulo wifi del lugar de forma inalámbrica, y así accede a la interfaz de la nube que a su vez es monitoreada de forma continua en base a protocolos serial de funcionamiento que hagan fluir eficazmente los datos de los sensores y librerías en un tiempo determinado a los diversos dispositivos en la que estén vinculados.

Figura 3

Diagrama de conexión de los sensores a la placa Raspberry Pi



Nota. Esto asegura el correcto funcionamiento de los sensores, en especial el BMP280

Por ello se usa el sensor de DHT 22 para la medición de temperatura y humedad y el sensor BMP280 se lo utiliza para medir temperatura y presión, como ambos sensores miden temperatura en el código se realizó una suma y división entre dos de ambas temperaturas para llegar a un promedio entre la recepción de ambos, aunque ambos sensores pueden trabajar de manera independiente o simultánea dependiendo del caso se puede comparar las medidas de uno u otro en términos de temperatura y utilizar las demás mediciones como una guía, para la conexión del sensor de DHT 22, se utilizaron la salida de 5 voltios una Tierra y cualquiera de los pines de la Raspberry, mientras que para el sensor BMP280, se utilizaron una salida de 3,3 voltios una Tierra y los pines 2 y 3 respectivamente debido a que estos son serial data y serial clock los cual son necesarios a la hora de procesar los datos de entrada del sensor.

5.2. CODIGO SENSOR DHT22, BMP280 Y DE USO SIMULTANEO CON THINGSPEAK

Programación de los códigos usados para comunicar los sensores con la plataforma Thingspeak el cual permite realizar las configuraciones respectivas para conectar los datos obtenidos y con la ayuda de la nube se puede respaldar la información, siendo esta accesible mediante la página oficial de Mathworks o mediante la aplicación móvil ThingView Free, la cual permite enlazar las gráficas y datos de Thingspeak mediante un código proporcionado por esta misma plataforma, a continuación se mostraran las partes del código necesarias para la lectura de los valores

Este es el código del sensor DHT22 encargado de tomar las lecturas de temperatura y humedad.

```
THINGSPEAK_API_KEY = "777A7GU55HAZ5QGU" # Reemplaza con tu clave de escritura
```

```
THINGSPEAK_URL = "https://api.thingspeak.com/update"
```

```
while True:
```

```
    # Leer datos del sensor
```

```

humedad, temperatura = Adafruit_DHT.read(SENSOR, PIN)

if humedad is not None and temperatura is not None:

    print("Temp={0:0.1f}C Hum={1:0.1f}%".format(temperatura, humedad))

# Enviar datos a ThingSpeak

payload = {

    "api_key": THINGSPEAK_API_KEY,

    "field1": temperatura,

    "field2": humedad

}

response = requests.get(THINGSPEAK_URL, params=payload)

if response.status_code == 200:

    print("Datos enviados correctamente a ThingSpeak.")

```

Este es el código del sensor BMP280 encargado de tomar las lecturas de temperatura y presión.

```

# Bucle infinito para leer datos del sensor y enviarlos a ThingSpeak

try:

    print("Presiona Ctrl+C para detener el programa.")

    while True:

        try:

            # Leer datos del sensor

            temperatura = bmp280.get_temperature()

            presion_hpa = bmp280.get_pressure()

```

A continuación se presenta el código para el funcionamiento de ambos sensores en simultaneo toma lecturas de temperatura, humedad y presión.

```
THINGSPEAK_API_KEY = "OXNF0BGW2MAH9HMY" # Reemplaza con tu API Key
```

```
THINGSPEAK_URL = "https://api.thingspeak.com/update"
```

```
# Bucle infinito para leer datos de los sensores y enviarlos a ThingSpeak
```

```
try:
```

```
    print("Presiona Ctrl+C para detener el programa.")
```

```
    while True:
```

```
        try:
```

```
            # Leer datos del sensor BMP280
```

```
            temperatura = bmp280.get_temperature()
```

```
            presion_hpa = bmp280.get_pressure()
```

```
            presion_bar = presion_hpa / 1000 # 1 hPa = 0.001 Bar
```

```
            presion_psi = presion_hpa * 0.0145038 # 1 hPa = 0.0145038 PSI
```

```
            # Leer datos del sensor DHT22
```

```
            humedad, temp_dht = Adafruit_DHT.read(DHT_SENSOR, DHT_PIN)
```

```
# Enviar datos a ThingSpeak
```

```
    payload = {
```

```
        "api_key": THINGSPEAK_API_KEY,
```

```
        "field1": temperatura,
```

```
        "field2": presion_bar,
```

```
        "field3": presion_psi
```

```
    }
```

```
response = requests.get(THINGSPEAK_URL, params=payload)
```

```
if response.status_code == 200:
```

```
    print("Datos enviados correctamente a ThingSpeak.")
```

5.3. Pasos para utilizar la Raspberry

Se procede a conseguir una carcasa o cobertor es recomendable para evitar exponer la Raspberry a los elementos, así como un difusor o ventilador para evitar sobrecalentamientos. Adquirir una tarjeta micro SD de 32 GB que permita grabar la imagen del sistema operativo de su preferencia, en este caso el Rpi_Bullseye_64_DNN, mediante el programa Raspberry Pi imager.

De allí se instala la tarjeta en el puerto de la Raspberry y conectar la pantalla mediante el puerto micro-hdmi, así como un mouse y un teclado para poder hacer las primeras configuraciones y así trabajar directamente o conectarla mediante el programa Real VNC, el cual viene integrado en la imagen, pero se debe descargar e instalar en teléfono o computador para acceder de manera remota, siempre y cuando ambos dispositivos estén en la misma red.

5.4. Pasos para utilizar los sensores

Para el sensor BMP280 se necesitan activar el bus I2C el cual puede ser mediante un comando en la consola o dándole al botón de inicio de la Raspberry, seleccionar preferencias, configuración de Raspberry pi, y en el menú de interfaces se debe activar el bus de datos I2C, después de reiniciarlo este estará activo al usar el comando de instalación de la librería en la consola. Al realizar las conexiones se efectuarán según el sensor lo indique en este caso el pin, da una salida de voltaje de 3.3V y los pines 2 y 3 que son serial data (SDA) y serial clock (SCL) respectivamente y el pin de tierra aunque este puede ser reemplazado por cualquier otro pin que cumpla la misma función.

Para el sensor DHT22 se necesita su código de funcionamiento y la Librería necesaria, para este caso utiliza la librería Adafruit_DHT. Para las conexiones se puede usar cualquier pin que de salida de 5V, así mismo con la tierra mientras que el pin de recepción de datos debe ser especificado en el código, aunque cualquiera de los pines que no sea voltaje o tierra puede cumplir esa función.

5.5. Pasos para usar Thingspeak y Thingview

Para usar la nube de Thingspeak se necesita una cuenta en Matlab, en el apartado de canales se debe crear uno o varios, dependiendo de las necesidades del operador. Después de realizada esta acción se nombra las gráficas para cada una de las variables a manejar, las cuales se mostrarán el cambio de las variables al pasar el tiempo. Para poder modificar esos valores se necesitará usar el código de escritura la cual permite acceder a las gráficas y modificarlas, por ello es integrada en el código o en los sensores que van a utilizar.

En el uso de Thingview Free se necesitará el código de usuario, la cual se puede encontrar en el apartado de perfil. De allí se procede a dar un clic al botón de "+" de la aplicación y se puede agregar solo un canal utilizando su código respectivo o todos los canales que se tiene en esa cuenta si se usa el código de usuario, estos aparecen en la pantalla y pueden ser accedidos sin ninguna restricción.

5.6. Comunicación entre los dispositivos

En lo que respecta a la comunicación entre los dispositivos con el sistema de monitoreo los sensores permiten una conexión eficiente para dar acceso a la información captada en tiempo real el cual al estar vinculados cada uno puede realizar una tarea en específico como almacenar y administrar datos. Al contar con una Raspberry Pi el cual es muy similar a las capacidades de una tarjeta Arduino lo que permite al sistema un mejor rendimiento una vez conectado con el dispositivo de aire acondicionado. Es decir, una vez que el sistema este instalado y listo para sus operaciones, los componentes integrados realizaran cada cual su

función donde el interfaz de conexión de la placa Raspberry Pi verificara que todo fluya en orden.

El manejo de la plataforma Thing Speak a más recopilar datos visualizar gráficas de consumo en el equipo de aire acondicionado, donde los canales creados tienen la función de almacenar todos los datos obtenidos de los sensores y proyectar los en una gráfica se forma instantánea mediante un protocolo de uso privado, aunque con la versión gratuita se puede crear hasta 5 canales de monitoreo donde la configuración otorga mayor lectura a los sensores dentro del perímetro del lugar o residencia.

5.7. Presentación de datos de la plataforma ThingSpeak

Se hace la presentación de datos al entrar en funcionamiento el prototipo del sistema de monitoreo en una sala donde se encontraban dos unidades de acondicionamiento de aire, se enmarcan diversos valores durante el tiempo que estuvieron encendidas de forma manual y automática, empezaron a mostrar información de temperatura, presión y humedad al estar operando en un ambiente controlado donde los sensores proporcionaron los datos suficientes para ser procesados por Raspberry Pi donde con la ayuda de la plataforma mencionada se observó el acoplamiento de los equipos de aire acondicionado al estar vinculados con el sistema mediante la conexión wifi se verificó en tiempo real el estado actual de funcionamiento o detectar posibles anomalías que existan, mientras que la temperatura y humedad fueron tomadas del ambiente, la presión fue simulada mediante el uso de una tubería de cobre conectada a un manómetro y este a un tanque de gas refrigerante.

Figura 4
Conexiones de los dispositivos usados en las pruebas

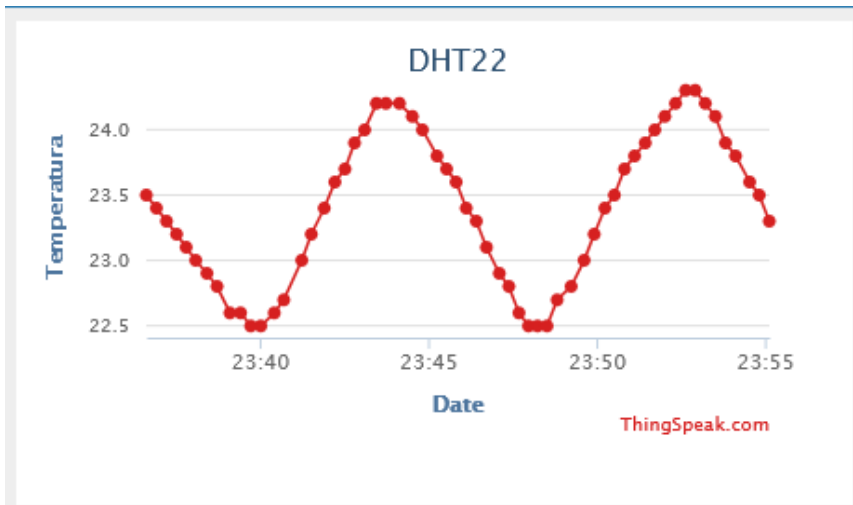


Nota. Se observan las conexiones de todos los sensores a la Raspberry Pi

GRÁFICAS DE LECTURAS DE TEMPERATURA Y HUMEDAD POR EL SENSOR DHT22

En la figura 5 se observa la respuesta del sensor DHT22 y sus lecturas a temperatura ambiente son comparadas con los valores de 26°C y 70% respectivamente, la unidad utilizada marcaba 26 °C y se puede observar una variación en temperaturas alrededor de 22.5°C y 24.3°C en general, estos valores se pueden ver afectados por las condiciones ambientales al momento de las mediciones

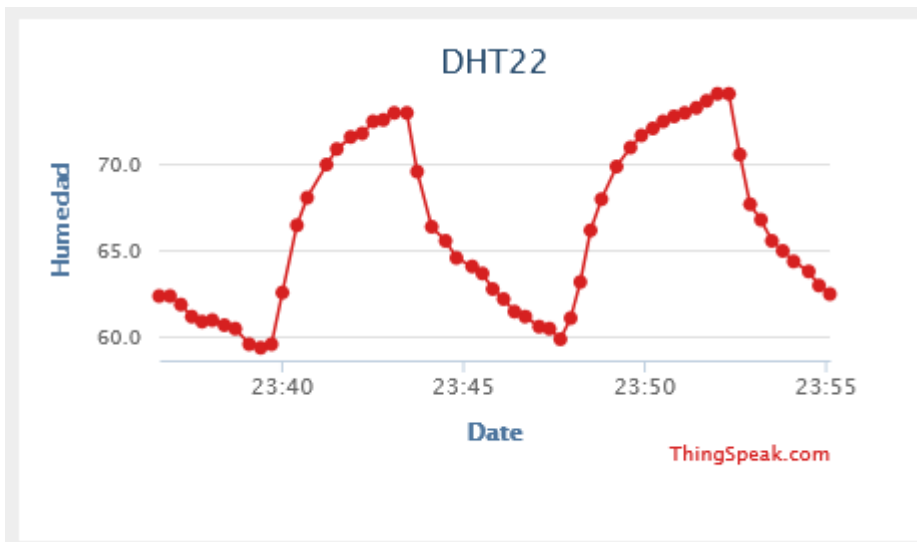
Figura 5
Lectura de temperatura del sensor DHT22



Nota. Grafico donde se pueden apreciar las lecturas de temperatura ambiente

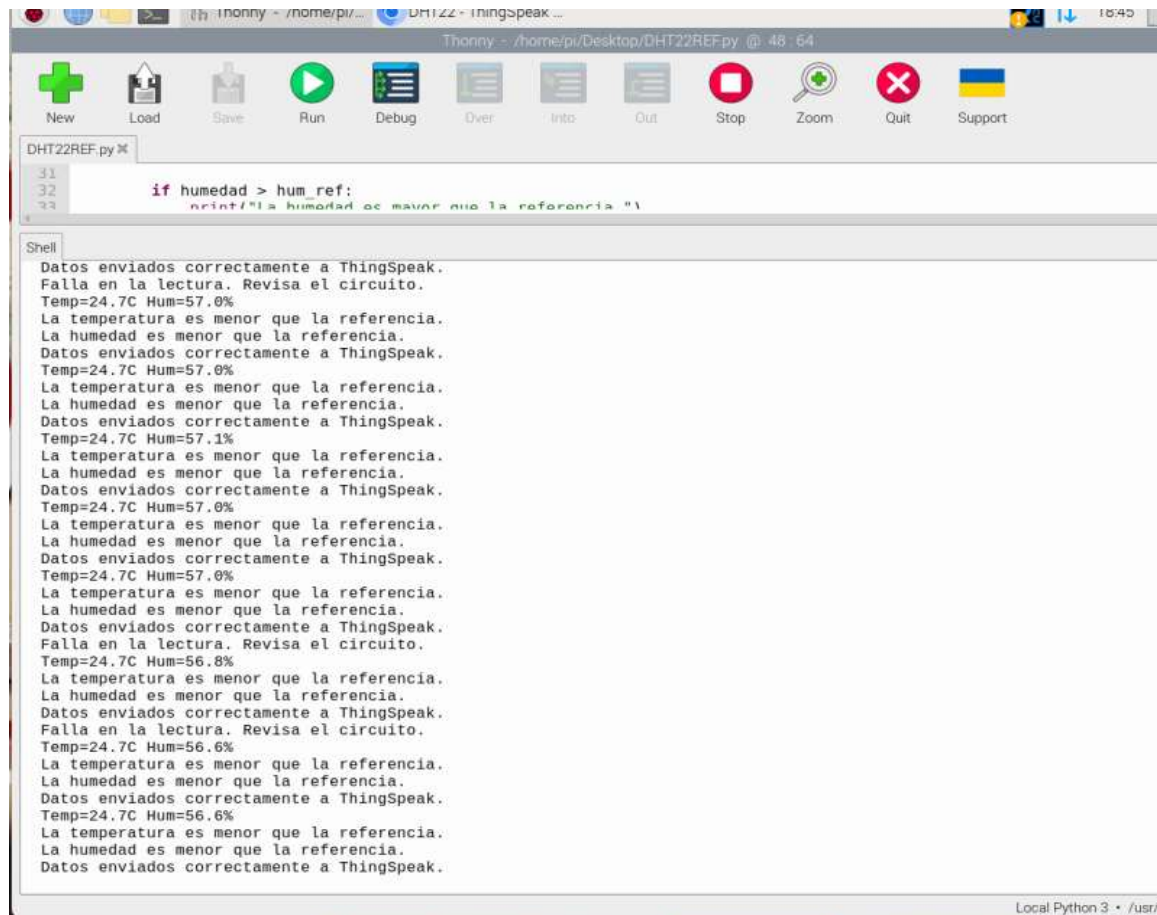
En la siguiente figura se puede observar la variación en la humedad que se mantiene entre valores de 70% y 59.4% como su pico más bajo mientras la unidad se encontraba encendida y funcionando.

Figura 6
Lectura de humedad del sensor DHT22



Nota. Se pueden visualizar los datos de humedad medidos por el sensor DHT22

Figura 7
Consola del programa thonny usando el sensor DHT22

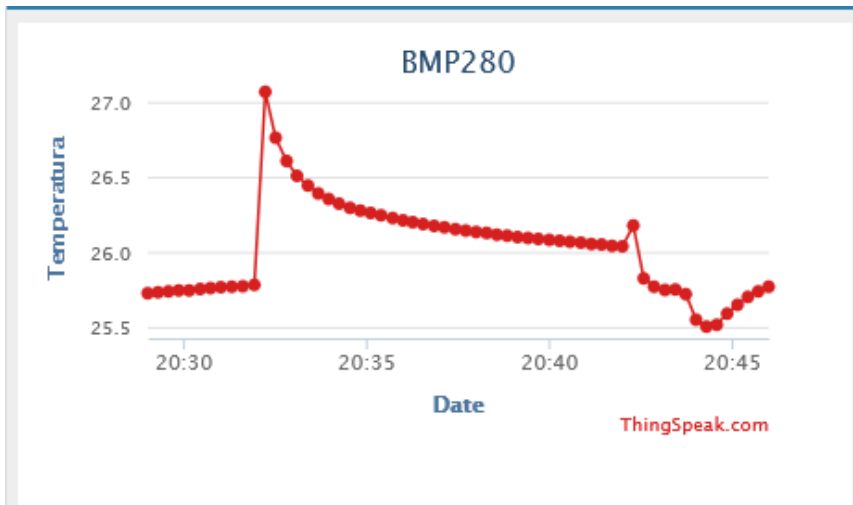


Nota. Muestra tanto datos como los mensajes de comparación.

GRÁFICAS DE LECTURA A TEMPERATURA Y PRESIÓN POR EL SENSOR BMP280

En la siguiente figura se aprecia la respuesta del sensor BMP280 y sus lecturas de temperatura y presión, estas son comparadas con los valores de 26°C, en este caso la temperatura que se mide es la temperatura interna dentro de nuestra simulación de compresor por lo cual la temperatura se eleva en un inicio con la entrada del gas refrigerante pero luego esta desciende hasta alrededor de los 25°C una vez que corte el flujo del gas.

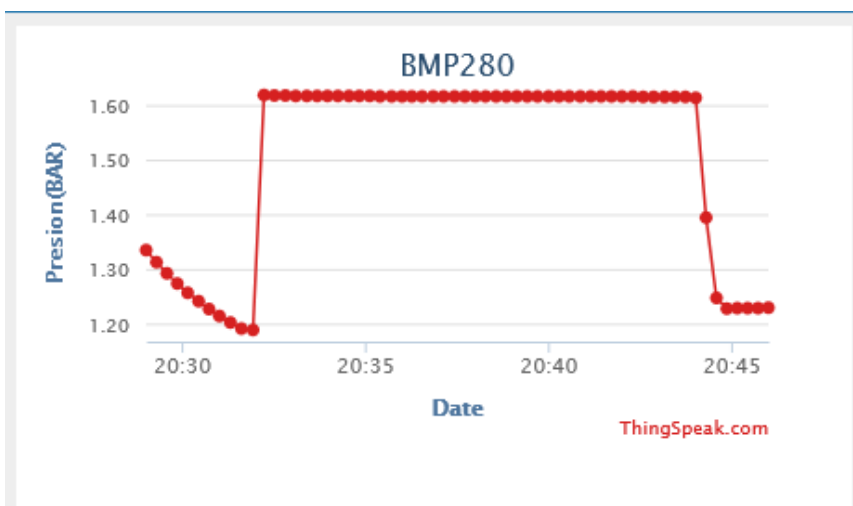
Figura 8
Lectura de temperatura del sensor BMP280



Nota. Muestra las variaciones de en la temperatura, que aumenta con la subida inicial de presión.

Con el flujo del gas refrigerante el pico más alto en la presión que se consiguió censar fue alrededor de 1.61 bares y se ve como esta descendiende paulatinamente una vez se abre la llave de paso del gas para que este escape al exterior, como se observa en la figura 9.

Figura 9
Lectura de presión en bares del sensor BMP280

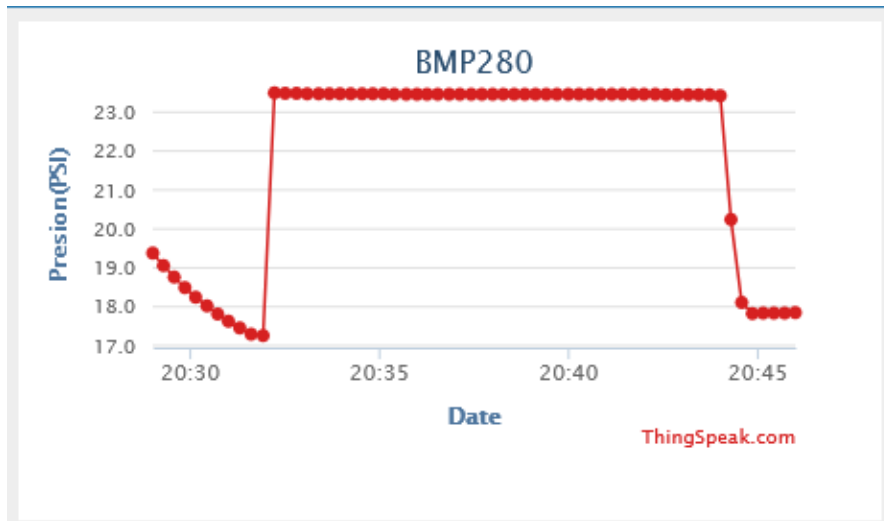


Nota. Se visualizan los datos de presión recibidos por el sensor BMP280.

En la figura 10 se puede observar la lectura de presiones en PSI del sensor BMP280, manteniéndose, en su pico más alto con 23.4 PSI y disminuyendo mientras se permite una fuga de aire para reducir la presión.

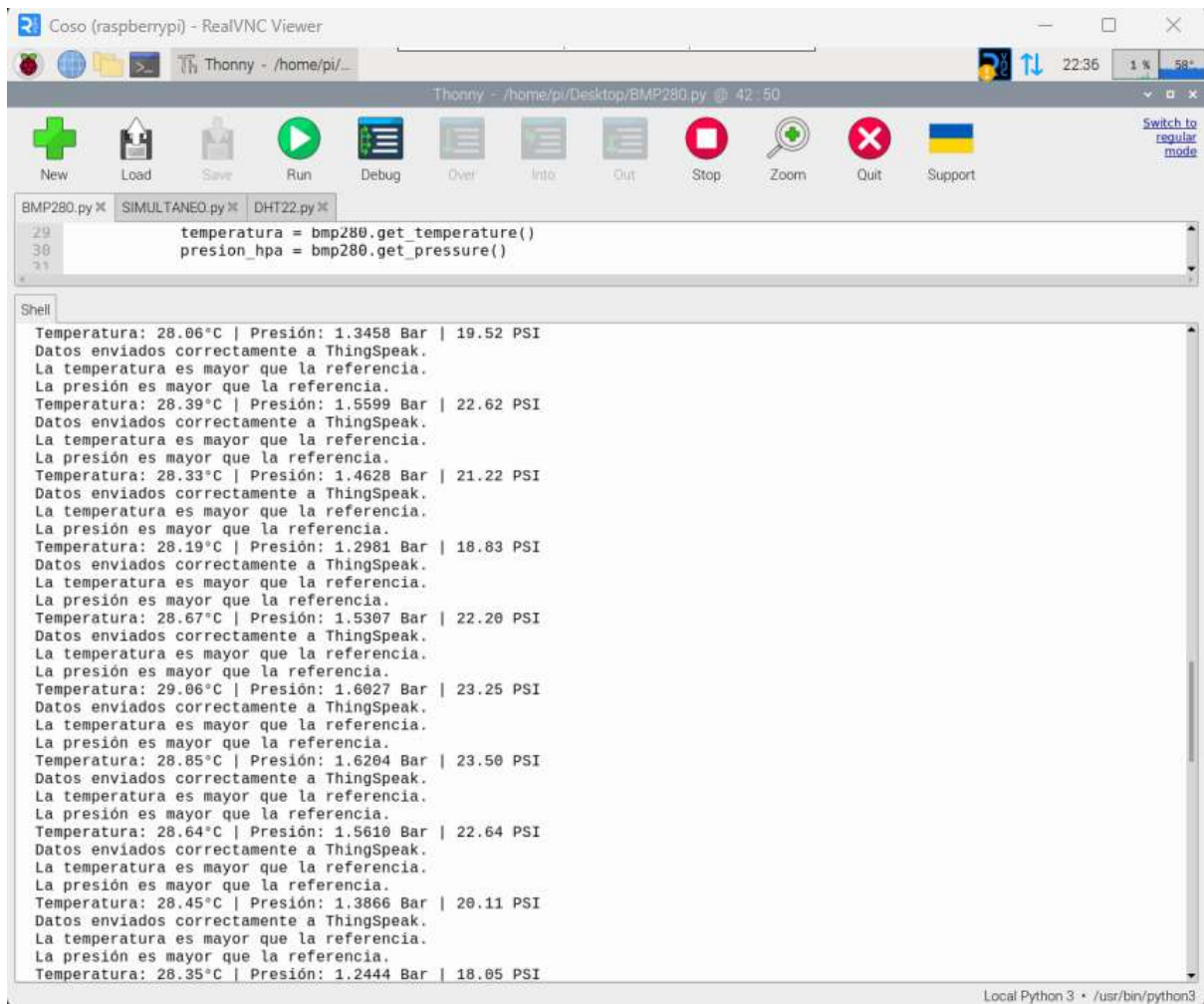
Figura 10

Lectura de presión en PSI del sensor BMP280



Nota. Se observa la traducción de Bares a PSI de las presiones censadas.

Figura 11
Consola del programa thonny usando el sensor BMP280



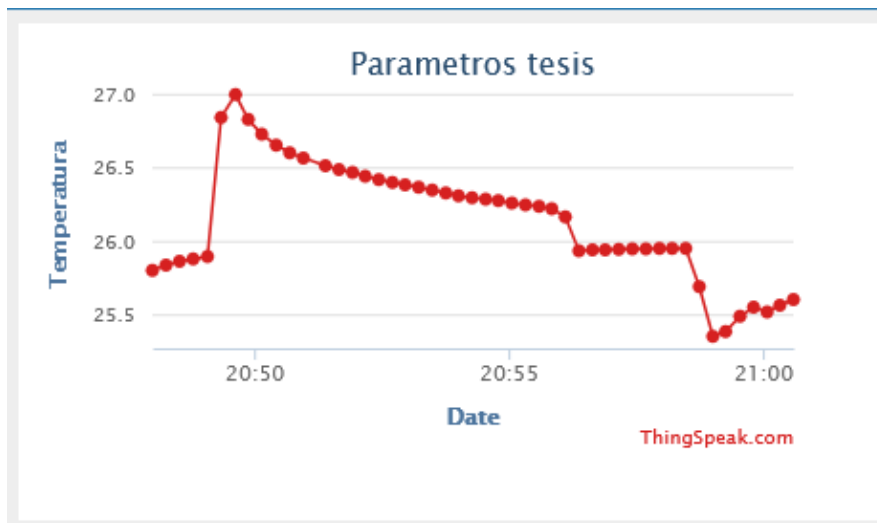
Nota. Se muestra la consola del thonny que proyecta los valores en tiempo real

GRÁFICAS DE LECTURAS DE TEMPERATURA, HUMEDAD Y PRESIÓN

En la figura 12 se puede observar la variación en las lecturas de temperatura, si bien la lectura puede parecer algo errática, esto se debe a que con el código de uso simultaneo la temperatura de ambos sensores es promediada, por lo que si la diferencia es muy amplia puede afectar la linealidad de la gráfica.

Figura 12

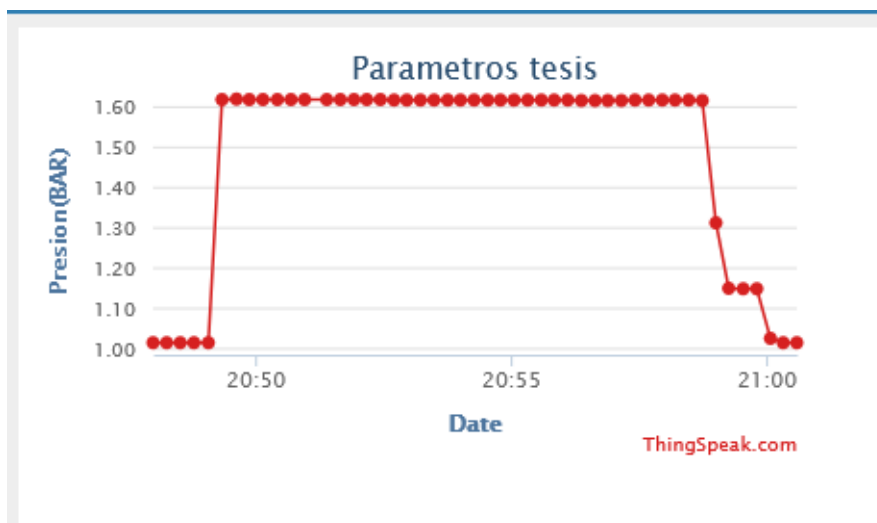
Lectura de temperatura de ambos sensores en simultaneo



Nota. Se visualiza el promedio de temperaturas entre ambos sensores.

Figura 13

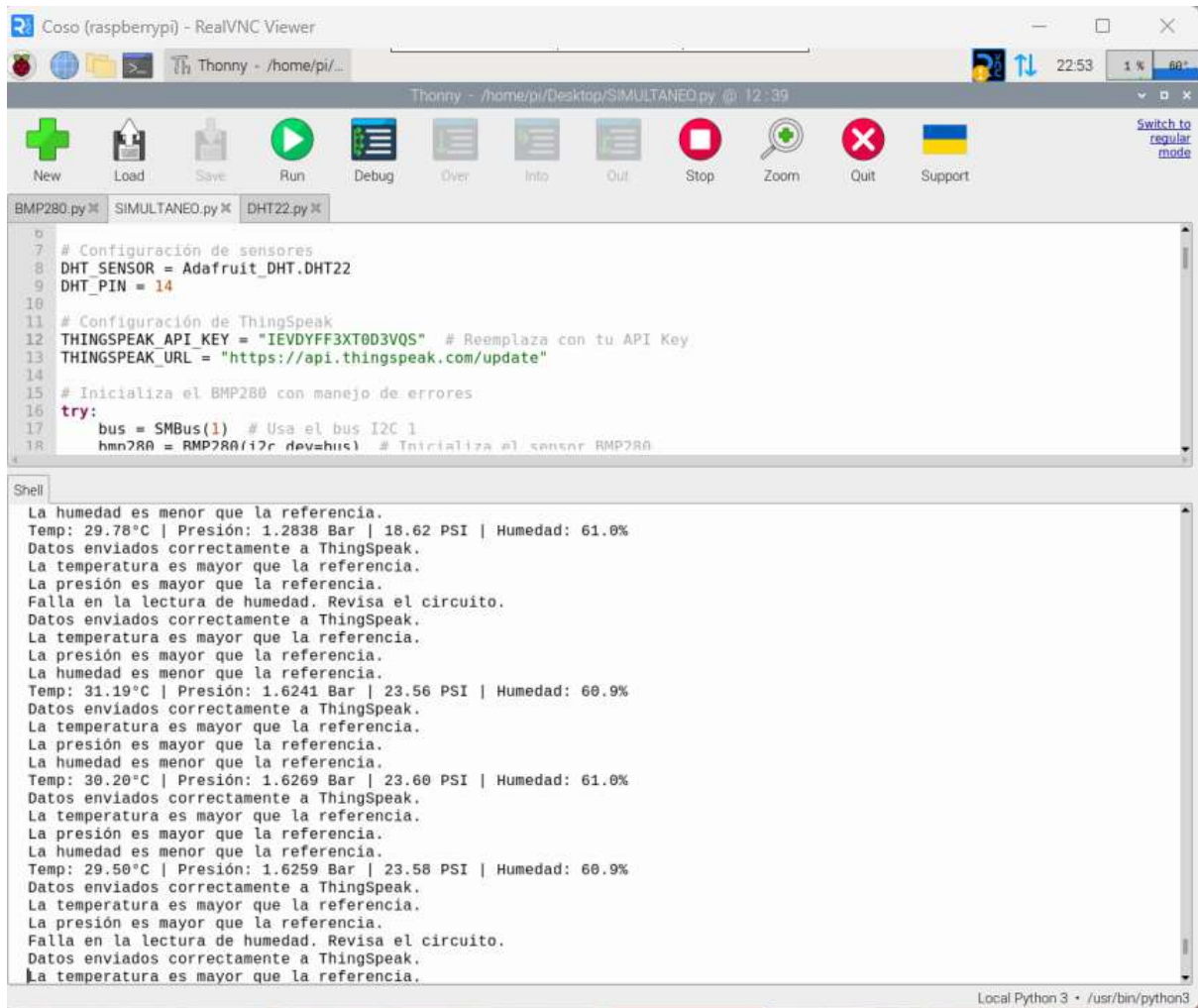
Lectura de presión en bares de ambos sensores en simultaneo



Nota. Se observa el valor de presión en Bares medido de manera constante

Figura 16

Consola del programa thonny usando ambos sensores a la vez



Nota. Se visualiza los mensajes que la consola del programa thonny muestra conforme receipta datos

VI. RESULTADOS

En este apartado se expone las pruebas realizadas con el prototipo de sistema de monitoreo con las dos unidades de aire acondicionado trabajando en simultaneo donde se verifico y evaluó el desempeño del mismo al realizarse la lectura de temperatura, presión y humedad bajo parámetros de tesis donde están operando en un escenario adecuado para la experimentación del cual se obtuvo la información lograda por los sensores y la precisión de la plataforma ThingSpeak de mostrar la información, donde fue necesario calibrar el sensor para conectar lo a los equipos de climatización y así comenzar obtener valores que son expresados en las siguientes gráficas.

6.1 resultados de lectura de los sensores BMP280 Y DHT22

Figura 17

Valores de temperatura mostrados en ThingView Free por el sensor BMP280



Nota. Se visualiza más a detalle las lecturas de temperatura del sensor BMP280

En las pruebas realizadas al sistema de monitoreo la lectura del BMP280, se obtuvo información acerca de la temperatura cuyo último valor fue 25.78, con un pico máximo de 27.07 a las 20:32, este pico de temperatura fue debido a los altos cambios de presión que sufrió el sensor en la simulación de compresor.

Figura 18

Valores de presión mostrados en ThingView Free por el sensor BMP280 en bares



Nota. Se visualiza una lectura constante de los valores de presión medida en Bares

En la gráfica se observa la lectura de presión en bares con el sensor BMP280 mostrando valores constantes desde las 20:32 hasta las 20:44 en los que las mediciones se mantuvieron constantes alrededor de los 1.6 bares, las bajadas de presión que se pueden observar se deben a la liberación de la misma para aumentar el rango de mediciones, alcanzando su punto más bajo en los 1.19 bares.

Figura 19

Valores de presión mostrados en ThingView Free por el sensor BMP280 en PSI



Nota. Se visualizan las lecturas de presión traducidas de Bares a PSI

Esta lectura es simplemente una traducción de Bares a PSI para mejor entendimiento de las presiones que se están receptando, siendo la conversión $1\text{Bar} = 14.50\text{PSI}$, se puede entender que tan alta es la presión atmosférica que se recepta en el exterior y que tan alta es la presión interna de nuestra simulación de compresor.

Figura 20

Valores de temperatura mostrados en ThingView Free por el sensor DHT22

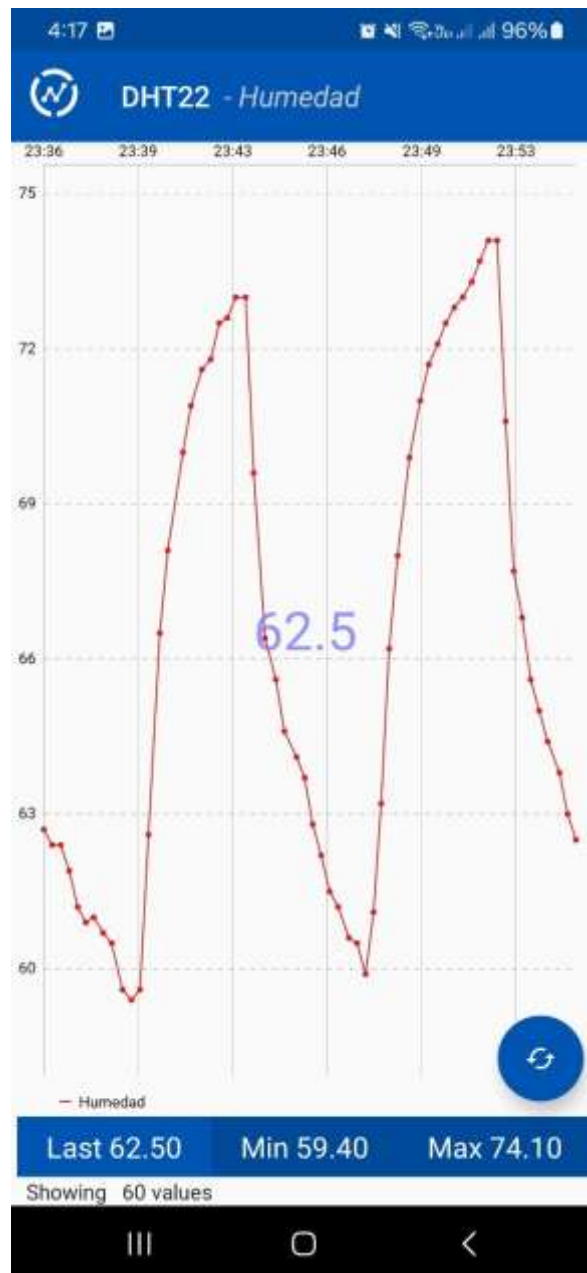


Nota. Se visualiza la gráfica de temperaturas captadas por el sensor DHT22

En las pruebas de la temperatura con el sensor DHT22 se posee un rango amplio de muestras con un pico máximo de 24.3 °C a las 23:52 y 23:43, con puntos muy bajos de 22.5°C a las 23:40 y 23:48, la unidad que fue utilizada para las mediciones marcaba alrededor de los 26°C cuando estos valores fueron registrados.

Figura 21

Valores de humedad mostrados en ThingView Free por el sensor DHT22



Nota. Se aprecian los valores de humedad receptados por el Thingview

En la siguiente grafica se puede observar la medición de humedad ente las 23:36 y las 23:55, manteniendo cambios casi constantes tanto en las bajadas como subidas de humedad, alcanzando picos de hasta 74% y puntos muy bajos para esta época del año como lo pueden ser 59.4%, teniendo en cuenta que la humedad suele estar entre 70% y 90% en estas épocas del año

6.2 resultados de lectura simultanea de los sensores

Figura 22

Valores de temperatura mostrados en ThingView Free usando ambos sensores



Nota. Se pueden observar los valores de temperatura promediados según la lectura de ambos sensores

Para la recepción de datos con el uso simultaneo de los sensores la temperatura se promedia entre la lectura del sensor DHT22 y el sensor BMP280, esto se hace para tener solo una lectura de temperatura y poder facilitar el análisis de datos, como se puede observar en la figura 21 se encuentran picos de subida y bajada irregulares debido a el promedio de las temperaturas ya que el sensor DHT22 se encontraba en un ambiente más climatizado que el BMP280, con valores variando entre 27°C y 25. °C.

Figura 23

Valores de presión mostrados en ThingView Free usando ambos sensores en Bares



Nota. Se observan los valores de presión en Bares captados por el sensor BMP280 en uso simultaneo

Así como en la figura 17 se puede observar las medidas de presión en Bares, en la figura 22 se puede observar un patrón muy similar con valores altos de 1.6 Bares que se mantiene prácticamente sin cambios durante casi 9 minutos seguidos.

Figura 24

Valores de presión mostrados en ThingView Free usando ambos sensores en PSI



Nota. Se visualiza la traducción de los valores de Bares a PSI de la figura 22

Partiendo de la base de que la presión promedio del ambiente esta alrededor de 14.50 PSI, se puede notar la subida en los valores debido a la presión realizada por el gas refrigerante en el tubo de cobre que utilizados para simular un compresor, elevándola hasta el punto de casi duplicarla con un 23.4 de PSI.

Figura 25

Valores de humedad mostrados en ThingView Free usando ambos sensores



Nota. Se aprecian los valores de humedad captados por el sensor DHT22 en uso simultaneo

Se puede apreciar en las mediciones que la humedad se eleva hasta aproximadamente los 80.3% debido a los cambios de clima presentados al momento de la medición, si bien la unidad de aire acondicionado del sitio donde se realizo la prueba lograba disminuir los valores en ciertos puntos como se puede apreciar en la figura 24, no fue suficiente para detenerla.

En las pruebas realizadas al sistema de monitoreo con dos equipos de aire acondicionado la plataforma ThingSpeak arrojó datos de las pruebas de resultados de lectura BMP280 y de lectura simultánea en cuestión de temperatura, presión y humedad cuyos resultados demuestran que el sistema de monitoreo funciona de manera óptima de donde se puede decir; que apenas se presentaron errores en la captación y procesamiento de la información proporcionada por los sensores cuando a momento de la experimentación los valores tienden a cambiar cuando se produce una variación en la temperatura ambiente lo que hace mayor el trabajo de la unidad de aire acondicionado reflejando un aumento de presión, humedad por ello el prototipo de monitoreo al estar conectado bajo la tecnología IoT puede efectuar un mejor optimización durante sus operaciones.

VII. CRONOGRAMA

Se estableció un cronograma de actividades para desarrollar paso a paso el sistema de monitoreo en un tiempo definido por la unidad de titulación de la Universidad politécnica Salesiana

Cronograma de Actividades																
Actividades	Noviembre				Diciembre				Enero				Febrero			
	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
Planificación y diseño inicial																
Definición de objetivos y requisitos del sistema	X															
Investigación de tecnologías IOT y sensores		X														
Estudio de la viabilidad técnica y desarrollo del prototipo			X													

Desarrollo de hardware y selección de sensores				
Selección de sensores y componentes.		X		
Desarrollo del circuito de conexión.		X		
Integración de sensores con microcontrolador.		X		
Implementación de pruebas de hardware.			X	
Desarrollo de Software y Plataforma IOT				

Desarrollo del software de recolección de datos.	X			
Implementación de plataforma IOT (nube).		X		
Configuración y pruebas del sistema.			X	
Pruebas, Optimización y Validación				
Realización de pruebas de campo, análisis de rendimiento y ajustes.				X

Optimización de alertas.	X		
Validación final del prototipo		X	
Pruebas y mantenimiento			
Implementación en entornos reales.		X	
Mantenimiento y soporte.			X
Documentación y Presentación			
Preparación de informes técnicos y manuales de usuario			X

VIII. PRESUPUESTO

A continuación, en la tabla 4 se detalla el presupuesto utilizado en el proyecto de titulación a realizar en el Periodo 65.

Tabla 4

Tabla de presupuestos para el proyecto

Cantidad	Componentes	Costos Estimados	
1	Raspberry Pi (Modelo Básico)	\$60	\$60
1	Sensores de Temperaturas y Humedad (DHT22)	\$10	\$10
1	Sensores de Corriente (ACS712)	\$8	\$8
1	Sensores de Presión (BMP180)	\$15	\$15
1	Módulo Wi-Fi ESP8266	\$10	\$10
1	Fuente de alimentación	\$10	\$10
1	Cables y Conectores	\$5	\$5
1	Placa de prueba (Breadboard)	\$5	\$5
1	Caja de plástico	\$5	\$5
1	Licencia de Firebase	Gratis uso básico	\$0
1	Servidor o plataforma en la nube	Gratis uso básico	\$0
1	Herramientas de medición	\$15	\$15
1	Costos adicionales (montaje, etc.)	\$4	\$4
60	Horas de Ingeniería	\$2.87	\$172.2
Total			\$319.2

Nota. Se detallan los gastos que se realizaron para el desarrollo del prototipo.

Estos valores fueron tomados de precios estándar de locales de venta de artículos tecnológicos para así estructurar el presupuesto adecuado que dio inicio al proceso de diseño y armado del sistema de monitoreo el cual está destinado a funcionar como un prototipo que controle el funcionamiento de una unidad de aire acondicionado.

IX. CONCLUSIONES

Una vez realizadas las pruebas del sistema de monitoreo se evidencia la efectividad del mismo durante sus operaciones al acoplarse con el equipo de aire acondicionado y la tecnología IoT, al constar una placa integrada Raspberry Pi el permitió un mayor rendimiento en los sensores y en la funcionalidad de la plataforma ThingSpeak a fin de lograr resultados óptimos al tener la convicción de haber realizado un prototipo tecnológico que cumplió con los objetivos tratados en la investigación.

Se concluye que el análisis y las pruebas efectuadas dieron el paso que consolido validar el diseño e implementación del sistema de monitoreo por cuanto fue previamente estudiado los tipos de componentes que integrarían tal obra tecnológica a beneficio y desarrollo de nuevos dispositivos que contribuyan con el bienestar general de la sociedad moderna.

Asimismo, se concluye que la obtención de datos en tiempo real permitió al usuario observar graficas de la plataforma ThingSpeak donde se brinda la potestad optimizar la funcionalidad no solo de la unidad de aire acondicionado sino también de los sensores y del sistema de monitoreo a fin de se realice mantenimiento al equipo si fuese necesario.

De igual manera el estudio permitió comprobar el estado actual de los dispositivos de aire acondicionado el cual previamente se le realizo el mantenimiento respectivo y preparación para acoplar el sistema de monitoreo y obtener en si la mejor capacidad de funcionamiento lo cual se comprobó en los resultados mostrados por la plataforma ThingSpeak demostraron que a pesar de que la temperatura ambiente influya el sistema en general correspondió en mostrar los datos procesados en tiempo real.

Además, se puede concluir que durante las pruebas los valores tienden a variar debido a situaciones externas es decir; cuando se produce una variación en la temperatura ambiente lo que hace mayor el trabajo de la unidad de aire acondicionado reflejando un aumento de presión, humedad por ello es necesario aprender optimizar el uso del equipo para evitar el sobrecalentamiento durante las horas que permanece encendido.

X. RECOMENDACIONES

Las pruebas realizadas permitieron comprobar la efectividad del sistema de monitoreo lo cual lo hace muy factible de implementación al tener componentes de fácil acceso en el mercado tecnológico local por lo cual es recomendable utilizar el prototipo que cuenta con acoplamiento a la tecnología Iot donde los datos se procesan inmediatamente para brindar al usuario un manejo adecuado a los equipos de aire acondicionado un óptimo funcionamiento.

Además, con las pruebas realizadas se permitió contrastar los resultados mostrados en las gráficas de la plataforma del sistema de monitoreo donde se pudo verificar que a temperatura ambiente la unidad de acondicionado suele aumentar o sobrecargar el trabajo del mismo por lo que es recomendable que los equipos cuenten con un sistema que envíe y procese datos en tiempo real que brinden información al usuario para que realice las acciones pertinentes en la optimización de la unidad de aire acondicionado.

Se recomienda efectuar revisiones mensuales a los equipos de aire acondicionado por parte de los usuarios para así evitar alteraciones en su funcionamiento al adquirir un sistema de monitoreo que brinde cobertura inmediata durante el tiempo de encendido del equipo y determina si existe sobrecalentamiento, y que los niveles de temperatura no sobrepasen los rangos establecidos en las gráficas de la plataforma.

XI. REFERENCIAS

- Academy, A. (10 de Enero de 2023). *Area Academy*. Obtenido de <https://areacooling.com/areacademy/es/ciclo-basico-de-refrigeracion/>
- AlfaIoT. (17 de Mayo de 2023). *AlfaIoT*. Obtenido de <https://alfaiot.com/iot/sensores-iot-populares-tipos-y-funciones/>
- COMAS, Z. G., SIMANCAS, J. G., VÉLEZ, J. Z., BERNAL, V. R., & PERCIA, I. V. (2018). *Redes de sensores inalámbricos para la monitorización de sistemas de calefacción, ventilación y aire acondicionado*. Obtenido de Revista Espacios: <https://www.revistaespacios.com/a18v39n45/18394513.html>
- COMAS-GONZÁLEZ, Z., SIMANCAS-GARCÍA, J., VÉLEZ-ZAPATA, J., BERNAL Rueda, V., & PERCIA Vásquez, I. (10 de Julio de 2018). *Revista Espacios*. Obtenido de <https://www.revistaespacios.com/a18v39n45/18394513.html>
- D. Navarro Pino, J. S. Badillo Rincón, M. D. Portillo Padilla, & S. E. Pineda Aguilera. (2024). *Tecnologías y herramientas del internet de las cosas (IoT) para el desarrollo de prototipos de entornos cotidianos*. Obtenido de Revista Colombiana de Tecnologías de Avanzada (RCTA): <https://ojs.unipamplona.edu.co/index.php/rcta/article/view/3020>
- Danahe, J. (12 de 06 de 2020). *Monitoreo y control de aires acondicionados*. Obtenido de Monitoreo y control de aires acondicionados: <https://0grados.com/monitoreo-y-control-de-aires-acondicionados/>
- DeweSoft. (2026). *ILOLITE Modular: Sistema de adquisición de datos industrial*. Obtenido de DeweSoft: <https://dewesoft.com/es/productos/iolite-modular>
- Ehtisham Ul Haque, Waseem Abbasi , Ahmad Almogren, Ayman Altameem, Ateeq Ur Rehman, & Habib Hamam . (2024). *Mejora del rendimiento en el intercambio de datos de IoT basado en blockchain mediante un algoritmo de consenso ligero*. Obtenido de Informes científicos: https://www-nature-com.translate.google/articles/s41598-024-77706-x?error=cookies_not_supported&code=f95b10cb-0e5a-456f-b458-79714ce42495&_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc
- Fernandez, J. (05 de 11 de 2019). *Universidad Salesiana de Guayaquil*. Obtenido de Módulo de control y monitoreo de sistemas de climatización mediante comunicación BACnet para ahorro energético: <https://dspace.ups.edu.ec/bitstream/123456789/17716/1/UPS-GT002744.pdf>
- Frisoli, C. (11 de Octubre de 2024). *HubSpot*. Obtenido de <https://blog.hubspot.es/website/que-es-python#que-es>
- Hat, R. (20 de Enero de 2023). *Red Hat*. Obtenido de <https://www.redhat.com/es/topics/internet-of-things/what-is-iot>
- IoT Analytics. (2024). *IoT Analytics: Los 10 principales casos de uso de IoT en 2024*. Obtenido de Generación Iot: <https://internetdelascosas.xyz/articulo.php?id=6996&titulo=IoT-Analytics-Los-10-principales-casos-de-uso-de-IoT-en-2024>
- Koolnova. (2024). *Innovaciones en HVAC para 2024: Cómo la tecnología está transformando la climatización comercial*. Obtenido de KOOLNOVA: <https://koolnova.com/innovaciones-en-hvac-para-2024/>
- PULLA, D. M. (2024). *DISEÑO DE UN SISTEMA IOT DE SUPERVISIÓN Y CONTROL DE MICROCLIMAS MEDIANTE LÓGICA DIFUSA APLICADO A UN LOTE DE VERMICOMPOST DENTRO DE LA ESPOCH*. Obtenido de ESCUELA SUPERIOR

- POLITÉCNICA DE CHIMBORAZO :
<https://182833.o7eteavr.asia/bitstream/123456789/22212/1/108T0539.pdf>
Refri-Climas. (28 de Abril de 2024). *Refri-Climas*. Obtenido de <https://www.refri-climas.com/ciclo-de-refrigeracion/>
- Saleem, Y., Crespi, N., Husain, M. R., & Copeland, R. (2019). *Internet of Things-Aided Smart Grid: Technologies, Architectures, Applications, Prototypes, and Future Research Directions*. Obtenido de ResearchGate:
https://www.researchgate.net/publication/316598969_Internet_of_Things-Aided_Smart_Grid_Technologies_Architectures_Applications_Prototypes_and_Future_Research_Directions
- Shafiq, S., Rahman, S., Ahmed, S. S., Mahmud, I., & Hosen, S. (2024). *Una revisión sobre redes definidas por software para la Internet de las cosas que incluye computación distribuida, blockchain y tecnología de redes móviles: conceptos básicos, tendencias, desafíos y potenciales de investigación futuros*. Obtenido de International Journal of Distributed Sensor Networks: <https://doi.org/10.1155/2024/9006405>
- UNIR. (17 de Mayo de 2023). *UNIR*. Obtenido de <https://www.unir.net/revista/ingenieria/arquitectura-iot/>

XII. ANEXOS

Figura 26

Raspberry Pi 4 Modelo B en funcionamiento



Nota. Se pueden observar las conexiones necesarias para el funcionamiento del prototipo

Figura 27

Contenedor de nitrógeno, nanómetro y tubo de cobre utilizados para simular el ambiente de un compresor



Nota. Sistema de pruebas para la mediciones de presión del sensor BMP280, antes de realizar el cambio a un tanque refrigerante.

Figura 28

Especificaciones técnicas de la unidad de aire acondicionado



Nota. Se observan las especificaciones técnicas de la unidad de acondicionamiento de aire utilizada para las pruebas del sensor DHT22

Figura 29

Estado de la unidad de aire acondicionado



Nota. Se observa el estado de la unidad utilizada para las mediciones del sensor DHT22

Programación utilizada para el uso de los sensores DHT22, BM280 y su uso simultaneo junto con la plataforma ThingSpeak

Programación sensor DHT22

```
import Adafruit_DHT
```

```
import time
```

```
import requests # Para enviar datos a ThingSpeak
```

```
# Configuración del sensor DHT22
```

```
SENSOR = Adafruit_DHT.DHT22
```

```
PIN = 14
```

```
# Configuración de ThingSpeak
```

```
THINGSPEAK_API_KEY = "777A7GU55HAZ5QGU" # Reemplaza con tu clave de escritura
```

```
THINGSPEAK_URL = "https://api.thingspeak.com/update"
```

```

# Solicitar valores de referencia al usuario

temp_ref = float(input("Ingrese el valor de temperatura de referencia (°C): "))

hum_ref = float(input("Ingrese el valor de humedad de referencia (%): "))

while True:

    # Leer datos del sensor

    humedad, temperatura = Adafruit_DHT.read(SENSOR, PIN)

    if humedad is not None and temperatura is not None:

        print("Temp={0:0.1f}C Hum={1:0.1f}%".format(temperatura, humedad))

    # Verificación de valores peligrosos

    if temperatura > 45 or humedad > 90:

        print(" ⚠ ¡ALERTA! Valores extremadamente altos y peligrosos ⚠ ")

        print("Revise las condiciones del ambiente y tome medidas preventivas.")

    # Comparar con los valores ingresados

    if temperatura > temp_ref:

        print("La temperatura es mayor que la referencia.")

    elif temperatura < temp_ref:

        print("La temperatura es menor que la referencia.")

    else:

        print("La temperatura es igual a la referencia.")

```

```

if humedad > hum_ref:

    print("La humedad es mayor que la referencia.")

elif humedad < hum_ref:

    print("La humedad es menor que la referencia.")

else:

    print("La humedad es igual a la referencia.")

# Enviar datos a ThingSpeak

payload = {

    "api_key": THINGSPEAK_API_KEY,

    "field1": temperatura,

    "field2": humedad

}

response = requests.get(THINGSPEAK_URL, params=payload)

if response.status_code == 200:

    print("Datos enviados correctamente a ThingSpeak.")

else:

    print("Error al enviar datos a ThingSpeak.")

else:

    print("Falla en la lectura. Revisa el circuito.")

time.sleep(5) # ThingSpeak permite actualizaciones cada 5 segundos

```

Programación sensor BMP280

```

import time

import requests

from smbus2 import SMBus

from bmp280 import BMP280

# Configuración de ThingSpeak

THINGSPEAK_API_KEY = "OXNF0BGW2MAH9HMY" # Reemplaza con tu API Key

THINGSPEAK_URL = "https://api.thingspeak.com/update"

# Inicializa el BMP280 con manejo de errores

try:

    bus = SMBus(1) # Usa el bus I2C 1

    bmp280 = BMP280(i2c_dev=bus) # Inicializa el sensor BMP280

    print("Sensor BMP280 inicializado correctamente.")

except Exception as e:

    print(f"Error al inicializar el sensor BMP280: {e}")

    exit(1)

# Solicitar valores de referencia al usuario

temp_ref = float(input("Ingrese el valor de temperatura de referencia (°C): "))

presion_ref_bar = float(input("Ingrese el valor de presión de referencia (Bar): "))

# Bucle infinito para leer datos del sensor y enviarlos a ThingSpeak

try:

    print("Presiona Ctrl+C para detener el programa.")

```

```

while True:

    try:

        # Leer datos del sensor

        temperatura = bmp280.get_temperature()

        presion_hpa = bmp280.get_pressure()

        # Convertir presión

        presion_bar = presion_hpa / 1000 # 1 hPa = 0.001 Bar

        presion_psi = presion_hpa * 0.0145038 # 1 hPa = 0.0145038 PSI

        # Verificación de valores peligrosos

        if temperatura > 45 or presion_bar > 7 or presion_psi > 110:

            print(" ⚠ ¡ALERTA! Valores extremadamente altos y peligrosos ⚠ ")

            print(f"Temperatura: {temperatura:.2f} °C | Presión: {presion_bar:.4f} Bar |
{presion_psi:.2f} PSI")

            print("¡Revise las condiciones del ambiente y tome medidas de seguridad!")

        # Comparar con los valores ingresados

        if temperatura > temp_ref:

            print("La temperatura es mayor que la referencia.")

        elif temperatura < temp_ref:

            print("La temperatura es menor que la referencia.")

        else:

            print("La temperatura es igual a la referencia.")

```

```

if presion_bar > presion_ref_bar:

    print("La presión es mayor que la referencia.")

elif presion_bar < presion_ref_bar:

    print("La presión es menor que la referencia.")

else:

    print("La presión es igual a la referencia.")

# Imprimir los datos en consola

print(f"Temperatura: {temperatura:.2f}°C | Presión: {presion_bar:.4f} Bar |
{presion_psi:.2f} PSI")

# Enviar datos a ThingSpeak

payload = {

    "api_key": THINGSPEAK_API_KEY,

    "field1": temperatura,

    "field2": presion_bar,

    "field3": presion_psi

}

response = requests.get(THINGSPEAK_URL, params=payload)

if response.status_code == 200:

    print("Datos enviados correctamente a ThingSpeak.")

else:

    print(f"Error al enviar datos: {response.status_code}")

```

```

except Exception as sensor_error:

    print(f"Error al leer datos del sensor: {sensor_error}")

# ThingSpeak permite actualizaciones cada 15 segundos (mínimo)
time.sleep(3)

except KeyboardInterrupt:

    print("\nPrograma finalizado por el usuario.")

finally:

    try:

        bus.close()

        print("Bus I2C cerrado correctamente.")

    except Exception as close_error:

        print(f"Error al cerrar el bus I2C: {close_error}")

```

Programación para el uso de ambos sensores en simultaneo

```

import time

import requests

import Adafruit_DHT

from smbus2 import SMBus

from bmp280 import BMP280

# Configuración de sensores

DHT_SENSOR = Adafruit_DHT.DHT22

DHT_PIN = 14

```

```

# Configuración de ThingSpeak

THINGSPEAK_API_KEY = "IEVDYFF3XT0D3VQS" # Reemplaza con tu API Key
THINGSPEAK_URL = "https://api.thingspeak.com/update"

# Inicializa el BMP280 con manejo de errores

try:

    bus = SMBus(1) # Usa el bus I2C 1

    bmp280 = BMP280(i2c_dev=bus) # Inicializa el sensor BMP280

    print("Sensor BMP280 inicializado correctamente.")

except Exception as e:

    print(f"Error al inicializar el sensor BMP280: {e}")

    exit(1)

# Solicitar valores de referencia al usuario

temp_ref = float(input("Ingrese el valor de temperatura de referencia (°C): "))

presion_ref_bar = float(input("Ingrese el valor de presión de referencia (Bar): "))

hum_ref = float(input("Ingrese el valor de humedad de referencia (%): "))

# Bucle infinito para leer datos de los sensores y enviarlos a ThingSpeak

try:

    print("Presiona Ctrl+C para detener el programa.")

    while True:

        try:

            # Leer datos del sensor BMP280

```

```

temperatura = bmp280.get_temperature()

presion_hpa = bmp280.get_pressure()

presion_bar = presion_hpa / 1000 # 1 hPa = 0.001 Bar

presion_psi = presion_hpa * 0.0145038 # 1 hPa = 0.0145038 PSI

# Leer datos del sensor DHT22

humedad, temp_dht = Adafruit_DHT.read(DHT_SENSOR, DHT_PIN)

# Verificación de valores peligrosos

if temperatura > 45 or presion_bar > 7 or presion_psi > 110 or (humedad is not None
and humedad > 90):

    print("\n ⚠ ¡ALERTA! Valores extremadamente altos y peligrosos ⚠")

    print(f"Temperatura: {temperatura:.2f}°C | Presión: {presion_bar:.4f} Bar |
{presion_psi:.2f} PSI | Humedad: {humedad:.1f}%")

    print("¡Revise las condiciones del ambiente y tome medidas de seguridad!\n")

# Comparar con los valores ingresados

if temperatura > temp_ref:

    print("La temperatura es mayor que la referencia.")

elif temperatura < temp_ref:

    print("La temperatura es menor que la referencia.")

else:

    print("La temperatura es igual a la referencia.")

if presion_bar > presion_ref_bar:

```

```

    print("La presión es mayor que la referencia.")

elif presion_bar < presion_ref_bar:

    print("La presión es menor que la referencia.")

else:

    print("La presión es igual a la referencia.")

if humedad is not None:

    if humedad > hum_ref:

        print("La humedad es mayor que la referencia.")

    elif humedad < hum_ref:

        print("La humedad es menor que la referencia.")

    else:

        print("La humedad es igual a la referencia.")

    print(f"Temp: {temperatura:.2f}°C | Presión: {presion_bar:.4f} Bar |
{presion_psi:.2f} PSI | Humedad: {humedad:.1f}%")

else:

    print("Falla en la lectura de humedad. Revisa el circuito.")

# Enviar datos a ThingSpeak

payload = {

    "api_key": THINGSPEAK_API_KEY,

    "field1": temperatura,

    "field2": presion_bar,

    "field3": presion_psi,

```

```

        "field4": humedad if humedad is not None else 0
    }

    response = requests.get(THINGSPEAK_URL, params=payload)

    if response.status_code == 200:

        print("Datos enviados correctamente a ThingSpeak.")
    else:

        print(f"Error al enviar datos: {response.status_code}")

except Exception as sensor_error:

    print(f"Error al leer datos del sensor: {sensor_error}")

time.sleep(3)

except KeyboardInterrupt:

    print("\nPrograma finalizado por el usuario.")

finally:

    try:

        bus.close()

        print("Bus I2C cerrado correctamente.")
    except Exception as close_error:

        print(f"Error al cerrar el bus I2C: {close_error}")

```