

# **Propuesta de arquitectura en nube de un sistema IoT para ventilación inteligente en las aulas de la Universidad de Cundinamarca sede Fusagasugá**

## **Autores:**

Giovanny Andrés Sabogal Romero  
Laura Daniela Díaz Arias

**Universidad de Cundinamarca**  
Facultad de Ingeniería  
Ingeniería Electrónica  
Fusagasugá, Colombia  
2025

# **Propuesta de arquitectura en nube de un sistema IoT para ventilación inteligente en las aulas de la Universidad de Cundinamarca sede Fusagasugá**

Trabajo de grado presentado como requisito parcial para optar por el título de ingeniero electrónico

## **Autores:**

Giovanny Andrés Sabogal Romero  
Laura Daniela Díaz Arias

## **Director:**

Ing. Alejandro Patiño Carrillo, MSc.

## **Codirector:**

Ing. Mónica Espinosa Buitrago, PhD.

Línea de investigación:  
Telemática y telecomunicaciones

**Universidad de Cundinamarca**  
Facultad de Ingeniería  
Ingeniería Electrónica  
Fusagasugá, Colombia  
2025

## **Dedicatoria**

Dedicamos este trabajo primeramente a Dios, que nos ha dado discernimiento y fortaleza en todos los retos y procesos de la vida, a nuestras familias que han sido nuestro apoyo y sostén en cada etapa de crecimiento académico y personal.

Especialmente a nuestros padres que han sacrificado tanto de ellos para vernos como profesionales y a nuestros abuelos maternos que nos han alentado y creído en nosotros.

También a los docentes que nos inspiraron y cada persona que conocimos en la trayectoria universitaria, con los que compartimos y llamamos amigos, cada uno de ustedes nos enseñaron cosas valiosas.

# Agradecimientos

A Dios quien sostiene nuestro futuro y ha sido nuestra esperanza en todo momento.  
A nuestro director el ingeniero Alejandro Patiño Carrillo, quien con su dedicación y conocimientos ha hecho posible la mejor redacción de este trabajo.

A nuestra codirectora, la doctora Mónica Espinosa Buitrago, quien nos han orientado en el proceso y nos han inquietado a ir más allá de lo que planeábamos en un principio. El compromiso, involucración y conocimientos que nos han compartido nos ha guiado e inspiraron en cada etapa de aprendizaje.  
Al cuerpo de docentes del programa de Ingeniería electrónica que nos permitieron usar el espacio de la sala de docentes como experimento, en la construcción de nuestro trabajo.

A la Universidad de Cundinamarca sede Fusagasugá por el respaldo de recursos educativos, el ambiente de estudio, los espacios de conocimiento y la formación que hemos recibido.

También a nuestros compañeros de carrera y amigos que estuvieron para asesorarnos y acompañarnos.

## Resumen

El presente trabajo contiene el desarrollo de una propuesta de arquitectura en nube para un sistema de Internet de las Cosas (IoT, por sus siglas en inglés), que tiene por objetivo ofrecer un servicio de ventilación inteligente de acuerdo con un caso de uso (este término se refiere a la interacción de un usuario con herramientas de software para lograr un objetivo), en este caso es: recolección, recepción, almacenamiento, análisis y procesamiento de datos. El servicio de ventilación está en función de tres variables ambientales que evalúan la calidad de aire y confort térmico dentro de un espacio cerrado: temperatura, humedad y concentración de dióxido de carbono (CO<sub>2</sub>, por sus siglas en inglés). Para lograrlo, primero se selecciona la sala de docentes del programa de ingeniería electrónica en la Universidad de Cundinamarca sede Fusagasugá, como prueba de concepto (este término se usa para comprobar una tecnología a través de un experimento o prototipo básico). Por consiguiente, se hace una caracterización de la ventilación presente en el aula seleccionada por medio de un software de modelamiento computacional de fluidos. Seguidamente, se instala el sistema IoT encargado de la obtención de las variables ambientales dentro del aula. Luego, se diseña e implementa la propuesta de arquitectura con los servicios de la nube de Amazon, Amazon Web Services (AWS, por sus siglas en inglés), donde se realiza una recepción, almacén, análisis y procesamiento de los datos obtenidos con el sistema IoT para evaluar la calidad de aire, confort térmico y posteriormente visualizar los datos procesados a través de unas métricas. Con el modelado computacional de fluidos dentro del software Ansys Fluent en su versión estudiantil, se observó el flujo que tiene el aire y la distribución de calor en el aula de prueba de concepto de acuerdo con las aperturas de ventilación presentes. Para la obtención de los datos, el sistema IoT instalado está conformado por una Raspberry Pi 4 modelo B y dos sensores que toman las variables ambientales. En el diseño de la arquitectura se trabajaron servicios dentro de la capa gratuita de AWS, como Amazon IoT Core para establecer comunicación con la Raspberry Pi, y así, tener recepción y enrutamiento de los datos a otros servicios y Amazon Simple Storage Service (S3) que almacena los datos por medio de objetos tipo JSON o CSV.

**Palabras claves:** Ventilación inteligente, Internet de las Cosas (IoT), modelamiento de fluidos, calidad de aire, arquitectura en nube.

## Abstract

This work presents the development of a cloud architecture proposal for an Internet of Things (IoT) system, aimed at providing an intelligent ventilation service based on a specific use case (this term refers to the interaction of a user with software tools to achieve a goal). In this case, the use case involves the collection, reception, storage, analysis, and processing of data. The ventilation service operates based on three environmental variables that evaluate air quality and thermal comfort within an enclosed space: temperature, humidity, and carbon dioxide (CO<sub>2</sub>) concentration. To achieve this, the faculty lounge of the Electronic Engineering program at the University of Cundinamarca, Fusagasugá campus, was selected as a proof of concept (this term refers to testing a technology through an experiment or basic prototype). Consequently, the ventilation conditions of the selected room were characterized using computational fluid dynamics modeling software. Next, the IoT system responsible for collecting environmental variables in the room was installed. Then, a cloud architecture proposal was designed and implemented using Amazon Web Services (AWS). This architecture handles the reception, storage, and analysis of data collected by the IoT system to evaluate air quality and thermal comfort and finally visualizes the proof-of-concept data through a set of metrics. Using computational fluid dynamics modeling in the student version of Ansys software, the airflow and heat distribution in the proof-of-concept classroom were analyzed based on the existing ventilation openings. The data collection system consists of a Raspberry Pi 4 Model B and two sensors that capture environmental variables. In the architecture design, AWS free tier services were used and adapted, taking into account their resource limitations for instance, using Amazon IoT Core to establish communication with the Raspberry Pi, enabling data reception and routing to other services, and Amazon Simple Storage Service (S3) to store the data in JSON or CSV format.

**Keywords:** smart ventilation, Internet of Things (IoT), fluid modeling, air quality, cloud architecture.

# Contenido

Resumen .....	5
Abstract.....	6
INTRODUCCIÓN .....	15
<b>PLANTEAMIENTO DEL PROBLEMA</b> .....	17
<b>OBJETIVO GENERAL</b> .....	18
<b>OBJETIVOS ESPECÍFICOS</b> .....	18
<b>JUSTIFICACIÓN</b> .....	19
<b>ALCANCES Y LIMITACIONES</b> .....	20
<b>Alcances</b> .....	20
<b>Limitaciones</b> .....	20
1. <b>MARCO REFERENCIAL</b> .....	21
1.1. <b>ESTADO DEL ARTE</b> .....	21
1.2. <b>FUNDAMENTOS TEÓRICOS</b> .....	22
1.2.1.    IoT (Internet of Things) .....	22
1.2.2.    Computación en la Nube .....	22
1.2.3.    Amazon Web Services (AWS).....	22
1.2.4.    Tipos de datos en la nube .....	23
1.2.5.    Tipos de almacenamiento en la nube .....	24
1.2.6.    Tipos de procesamiento .....	24
1.2.7.    Ventilación Inteligente.....	25
1.2.8.    Mecánica de Fluidos.....	25
1.2.9.    Calidad del aire y cómo se mide.....	26
1.2.10.    Estándares de Calidad del Aire.....	27
2. <b>DISEÑO METODOLÓGICO</b> .....	29
2.1. <b>METODOLOGÍA</b> .....	29
3. <b>DESARROLLO DEL TRABAJO</b> .....	31
3.1. <b>CARACTERIZACIÓN</b> .....	31

3.1.1.	Caracterización de la ventilación existente.....	32
3.1.1.1	Escenario de ventilación 1 (Puerta y ventanas abiertas). ....	34
3.1.1.2	Escenario de ventilación 2 (Puerta cerrada y ventanas abiertas): .....	36
3.1.1.3	Escenario de ventilación 3 (Puerta abierta y ventanas cerradas) .....	38
3.1.1.1	Parámetros meteorológicos .....	40
3.1.2.	Análisis de resultados de simulación.....	41
3.1.3.	Caracterización de un servicio de ventilación inteligente.....	41
3.1.2.1	Variables del sistema .....	41
3.1.2.2	Intervalos de medición de parámetros (Temperatura, humedad y CO2) 43	
3.1.3	Caracterización de hardware.....	44
<b>3.2</b>	<b>DISEÑO</b> .....	44
3.2.1	Escenario IoT .....	44
3.2.2	Amazon Web Services (AWS).....	45
3.2.3	Diseño de arquitectura en nube.....	45
3.2.4	Seguridad dentro de AWS .....	47
3.2.5	Protocolo compatible con la autenticación mutua:.....	48
<b>3.3</b>	<b>IMPLEMENTACIÓN</b> .....	48
3.3.1	Instalación del sistema IoT .....	48
3.3.2	Configuración de Node – Red .....	49
3.3.2.1	Configuración remota.....	52
3.3.3	Configuración de los servicios dentro de AWS .....	53
3.3.4.	Paso 1: Capa de recepción y almacenamiento de datos.....	53
3.3.5.	Paso 2: Capa de análisis .....	55
3.3.6.	Paso 3: Procesamiento de datos.....	57
3.3.7.	Visualización de los datos .....	60
3.3.8.	Extracción de datos del Bucket S3 .....	61
<b>3.4</b>	<b>PRUEBAS DE FUNCIONAMIENTO Y/O DESEMPEÑO</b> .....	62
3.4.1	Obtención de datos de monitoreo IoT.....	62
3.4.2	Preparación de los datos para el procesamiento mediante Lambda .....	62

3.4.3	Activación del trabajo de procesamiento por medio de Lambda .....	63
3.4.4	Implementación de bucket S3 .....	63
3.4.5	Procesamiento de los datos ambientales .....	64
3.4.6	Visualización mediante Tableau. ....	64
3.4.7	Validación de la arquitectura .....	68
3.4.8	Validación de recepción y almacenamiento de los datos .....	69
3.4.9	Validación de sistema IoT.....	71
3.4.10	Resumen de la validación.....	72
<b>4.</b>	<b>ANÁLISIS DE RESULTADOS Y DISCUSIÓN.....</b>	<b>73</b>
<b>4</b>	<b>. CONCLUSIONES Y TRABAJOS FUTUROS .....</b>	<b>76</b>
4.1.	CONCLUSIONES .....	76
<b>4.2.</b>	<b>PERSPECTIVAS DE TRABAJO FUTURO .....</b>	<b>76</b>
<b>5.</b>	<b>BIBLIOGRAFIA .....</b>	<b>78</b>

# Índice de ilustraciones

Ilustración 1. Fases metodológicas. <i>Fuente: elaboración propia.</i> .....	29
Ilustración 2. Plano 2D de aula de profesores, bloque C. <i>Fuente: elaboración propia.</i> .....	32
Ilustración 3. Plano 2D de aula de profesores, bloque C. <i>Fuente: elaboración propia.</i> .....	33
Ilustración 4. Modelado 3D de aula de profesores, bloque C. <i>Fuente: elaboración propia.</i> .....	33
Ilustración 5. Modelado de aula de profesores bloque C. <i>Fuente: elaboración propia.</i> .....	34
Ilustración 6. Vista lateral escenario de ventilación 1 (Puerta y ventanas abiertas). <i>Fuente: elaboración propia.</i> .....	34
Ilustración 7. Vista superior escenario de ventilación 1 (Puerta y ventanas abiertas). <i>Fuente: elaboración propia.</i> .....	35
Ilustración 8. Vista trasera escenario de ventilación 1 (Puerta y ventanas abiertas). <i>Fuente: elaboración propia.</i> .....	35
Ilustración 9. Vista frontal escenario de ventilación 1 (Puerta y ventanas abiertas). <i>Fuente: elaboración propia.</i> .....	36
Ilustración 10. Vista lateral escenario de ventilación 2 (Puerta cerrada y ventanas abiertas). <i>Fuente: elaboración propia.</i> .....	36
Ilustración 11. Vista superior escenario de ventilación 2 (Puerta cerrada y ventanas abiertas). <i>Fuente: elaboración propia.</i> .....	37
Ilustración 12. Vista trasera escenario de ventilación 2 (Puerta cerrada y ventanas abiertas). <i>Fuente: elaboración propia.</i> .....	37
Ilustración 13. Vista frontal escenario de ventilación 2 (Puerta cerrada y ventanas abiertas). <i>Fuente: elaboración propia.</i> .....	38
Ilustración 14. Vista lateral escenario de ventilación 3 (Puerta abierta y ventanas cerradas). <i>Fuente: elaboración propia.</i> .....	38
Ilustración 15. Vista superior escenario de ventilación 3 (Puerta abierta y ventanas cerradas). <i>Fuente: elaboración propia.</i> .....	39
Ilustración 16. Vista trasera escenario de ventilación 3 (Puerta abierta y ventanas cerradas). <i>Fuente: elaboración propia.</i> .....	39
Ilustración 17. Vista frontal escenario de ventilación 3 (Puerta abierta y ventanas cerradas).....	40
Ilustración 18. Resultados históricos del clima en Fusagasugá año 2024. <i>Fuente: [30]</i> .....	40
Ilustración 19. Ubicación del sistema IoT. <i>Fuente: elaboración propia.</i> .....	41
Ilustración 20. Rangos de temperatura y humedad ideal. <i>Fuente: [9]</i> .....	42

Ilustración 21. Efectos de la humedad relativa. <i>Fuente: [9]</i> .....	42
Ilustración 22. Efectos de la concentración de CO2 en ppm (partes por millón). <i>Fuente: [9]</i> .....	42
Ilustración 23. Diagrama de actores del sistema y el flujo de información. <i>Fuente: elaboración propia</i> .....	44
Ilustración 24. Propuesta de arquitectura. <i>Fuente: elaboración propia</i> .....	46
Ilustración 25. Montaje hardware IoT en la sala de prueba de concepto. <i>Fuente: elaboración propia</i> .....	49
Ilustración 26. Montaje hardware IoT. <i>Fuente: elaboración propia</i> .....	49
Ilustración 27. Configuración de módulos en la herramienta Node-RED. <i>Fuente: elaboración propia</i> .....	50
Ilustración 28. Debug de Node-RED. <i>Fuente: elaboración propia</i> .....	50
Ilustración 29. Notificación de alerta mediante correo electrónico. ....	51
Ilustración 30. Notificación de alerta mediante correo electrónico (Alerta de datos faltantes). <i>Fuente: elaboración propia</i> .....	51
Ilustración 31. Diagrama de secuencia funcionamiento de sensor MQ-135. <i>Fuente: elaboración propia</i> .....	51
Ilustración 32. Diagrama de flujo recopilación de datos en Node-RED. <i>Fuente: elaboración propia</i> .....	52
Ilustración 33. Creación de VPN mediante Tailscale. <i>Fuente: elaboración propia</i> .....	52
Ilustración 34. Creación de políticas AWS. <i>Fuente: elaboración propia</i> .....	53
Ilustración 35. Creación de certificados AWS. <i>Fuente: elaboración propia</i> .....	53
Ilustración 36. Conexión del Topic para la recepción de datos a través de MQTT. <i>Fuente: elaboración propia</i> .....	54
Ilustración 37. Diagrama de flujo de función lambda para enrutar. <i>Fuente: elaboración propia</i> .....	54
Ilustración 38. Comprobación de la transferencia de datos desde el IoT Core mediante la función Lambda a tabla en DynamoDB. <i>Fuente: elaboración propia</i> ..	55
Ilustración 39. Servicios de Capa de recepción y almacenamiento de datos del paso 1. <i>Fuente: elaboración propia</i> .....	55
Ilustración 40. Diagrama de flujo de función lambda para ordenar los datos a partir de DynamoDB. <i>Fuente: elaboración propia</i> .....	56
Ilustración 41. Creación de Bucket S3 para almacenamiento de datos por días, a partir de la tabla de DynamoDB. <i>Fuente: elaboración propia</i> .....	56
Ilustración 42. Creación de archivo script.py dentro de Bucket S3 para el uso de Sagemaker para procesamiento de los datos obtenidos. <i>Fuente: elaboración propia</i> .....	57
Ilustración 43. Servicios Capa de análisis del paso 2. <i>Fuente: elaboración propia</i> .....	57

Ilustración 44. Diagrama de flujo de la activación de Sagemaker con la función Lambda. <i>Fuente: elaboración propia</i> .....	58
Ilustración 45. Configuración de EventBridge para ejecución automática de función Lambda para activación de procesamiento en Sagemaker. <i>Fuente: elaboración propia</i> .....	58
Ilustración 46. Diagrama de flujo procesamiento mediante Sagemaker. <i>Fuente: elaboración propia</i> .....	59
Ilustración 47. Verificación de procesamiento mediante Sagemaker. <i>Fuente: elaboración propia</i> .....	59
Ilustración 48. Almacenamiento de resultados a partir de Sagemaker. <i>Fuente: elaboración propia</i> .....	60
Ilustración 49. Servicios de procesamiento de datos del paso 3. <i>Fuente: elaboración propia</i> .....	60
Ilustración 50. Resultados de script de extracción de datos de S3. <i>Fuente: elaboración propia</i> .....	61
Ilustración 51. Almacenamiento de los resultados en carpetas por día. <i>Fuente: elaboración propia</i> .....	61
Ilustración 52. Diagrama de flujo script extracción de datos de S3. <i>Fuente: elaboración propia</i> .....	62
Ilustración 53. Administración y control de facturación dentro de AWS. <i>Fuente: elaboración propia</i> .....	64
Ilustración 54. Resultados de archivos JSON de procesamiento de datos. <i>Fuente: elaboración propia</i> .....	64
Ilustración 55. Selección de fuente de datos dentro de la aplicación de Tableau. <i>Fuente: elaboración propia</i> .....	65
Ilustración 56. Selección de gráficas y datos relevantes dentro de Tableau. <i>Fuente: elaboración propia</i> .....	65
Ilustración 57. Historial para visualización de las gráficas obtenidas. <i>Fuente: elaboración propia</i> .....	65
Ilustración 58. Uso de interpolación lineal para relleno de datos faltantes. <i>Fuente: elaboración propia</i> .....	66
Ilustración 59. Terminal de Python programa de relleno de datos. <i>Fuente: elaboración propia</i> .....	67
Ilustración 60. Visualización comparativa de datos "procesados crudos" y datos "procesados rellenos". .....	67
Ilustración 61. Diagrama de flujo script relleno de datos faltantes. <i>Fuente: elaboración propia</i> .....	68
Ilustración 62. Datos almacenados en Bucket (buckdiarias), para análisis de los datos recopilados frente a los estimados. <i>Fuente: elaboración propia</i> .....	70

Ilustración 63. Grafica de horas completas frente a horas faltantes. *Fuente: elaboración propia* .....70

Ilustración 64. Grafica de diferencia de resultados AWS - Recolectados. *Fuente: elaboración propia* .....71

Ilustración 65. Resultados de script de horas críticas. *Fuente: elaboración propia*73

Ilustración 66. Detección de horas críticas. *Fuente: elaboración propia* .....73

Ilustración 67. Valores altos día 29 de abril de 2025. *Fuente: elaboración propia* .75

## Índice de tablas

Tabla 1. Protocolos de red. <i>Fuente: elaboración propia</i> .....	28
Tabla 2. Protocolos de IoT. (la información está basada en (Amazon AWS, 2024)). .....	28
Tabla 3. Parámetros de simulación de Ansys Fluent. <i>Fuente: elaboración propia</i> .	31
Tabla 4. Conteo de iteraciones para simulación de escenarios. <i>Fuente: elaboración propia</i> .....	32
Tabla 5. Caracterización de hardware empleado. <i>Fuente: elaboración propia</i> . ....	44
Tabla 6. Análisis de datos recopilados. <i>Fuente: elaboración propia</i> .....	69
Tabla 7. Datos de AWS frente a datos aproximados por día (horas validas). <i>Fuente: elaboración propia</i> .....	70
Tabla 8. Promedio de temperatura y humedad registrados por el sistema IoT. <i>Fuente: elaboración propia</i> . .....	72
Tabla 9. Resumen de validación. <i>Fuente: elaboración propia</i> . .....	72
Tabla 10. Horas más frecuentes de valores máximos. <i>Fuente: elaboración propia</i>	74

# INTRODUCCIÓN

En el contexto actual, la calidad del aire en ambientes académicos es un factor clave que influye en la salud, el confort y el rendimiento de estudiantes y profesores. En aulas donde se concentran varias personas durante largos períodos la ventilación adecuada es esencial para mantener niveles seguros de CO<sub>2</sub>, una temperatura óptima y una humedad relativa confortable. Estudios muestran que un ambiente con mala calidad del aire puede provocar fatiga, distracción y problemas de salud, lo cual afecta directamente el rendimiento y el bienestar general de los ocupantes.

Este proyecto propone una arquitectura en la nube para gestionar un sistema IoT de ventilación inteligente en las aulas académicas de la Universidad de Cundinamarca sede Fusagasugá. Esto se consigue mediante el monitoreo de variables para recolectar datos y mediante servicios en nube, almacenar, procesar y analizar la información ofreciendo un servicio de ventilación inteligente dentro de espacios académicos donde se analice el confort térmico y la Calidad de Aire Interno (IAQ, por sus siglas en inglés). Los objetivos específicos de este estudio comprenden la caracterización del servicio de ventilación, el monitoreo de un sistema IoT de variables dentro de una prueba de concepto, definir los servicios en nube de la arquitectura de acuerdo con el caso de uso: obtención, almacenamiento, análisis y procesamiento de datos.

Uno de los alcances más destacables que abarca el proyecto es el modelado computacional de fluidos donde se simula a través de un software especializado el comportamiento del aire como un fluido, esto es importante en diseños arquitectónicos para evaluar la distribución de los ductos y aperturas de ventilación en edificaciones de distintos fines.

La estructura del documento está seccionada por apartados. Inicia desde el resumen, que expone brevemente el tema en estudio, seguidamente la introducción y el planteamiento del problema, donde está descrito el contexto y la necesidad de este trabajo. Luego, se describen los objetivos de estudio, justificación, alcances y limitaciones. Después se tiene el marco referencial, con el estado del arte y los fundamentos teóricos, que explican las bases teóricas y conceptos como el modelado computacional de fluidos y la calidad de aire. Continúa la sección de diseño metodológico y el desarrollo del trabajo, que se extiende detalladamente en la caracterización de la ventilación y el servicio de ventilación propuesto, los componentes del sistema IoT y la propuesta de arquitectura en nube planteada. Posteriormente, en el apartado de implementación y pruebas de funcionamiento, se evalúan las funciones de obtención, almacenamiento y procesamiento de los datos de acuerdo con los servicios de Amazon Web Services (AWS, por sus siglas en inglés) para el caso de uso. Para finalizar, se encuentran los resultados, discusión, conclusiones y trabajos futuros, donde se exponen las observaciones, aportes y futuras líneas de estudio para mejorar o complementar el proyecto, sucesivamente

en las páginas finales se presenta la bibliografía, de todo el proceso de investigación y consulta.

## PLANTEAMIENTO DEL PROBLEMA

La ventilación es la renovación del aire de un espacio cerrado, donde se reemplaza el aire interno por aire del exterior que ha sido filtrado o tratado. Existen estrategias y métodos de ventilación que varían dependiendo del área, ocupación y actividades que se desarrollen. Estos métodos de ventilación pueden ser naturales, mecánicos accionados por el hombre o Ventilación Mecánica Controlada (VMC), automatizados e inteligentes. Cada uno se distingue del otro por la incorporación o ausencia de tecnología.

Después del comienzo de la pandemia de COVID-19 y las altas temperaturas desatadas climáticamente por el calentamiento global, los sistemas de ventilación inteligente en edificios se han centrado en mantener la calidad de aire interno para evitar enfermedades respiratorias y ofrecer confort en cada espacio en función de la ocupación, horario de uso y actividades que se desarrollen [1], [2]. Para garantizar la IAQ, los modelos de ventilación inteligente monitorean y controlan variables recomendadas por la Agencia de Protección Ambiental de los Estados Unidos (EPA, por sus siglas en inglés), como temperatura, humedad, concentración de CO<sub>2</sub>, partículas en suspensión o aerosoles, Compuestos Orgánicos Volátiles (COV), entre otros. Estos sistemas también implementan estrategias de ventilación como la descrita según la guía de la Escuela de Salud Pública de la Universidad de Harvard, donde hay ciertas recomendaciones y estrategias de ventilación en edificios de uso público. Un espacio cerrado debe contar con la Renovación de Aire por Hora (ACH, por sus siglas en inglés) que mantenga una mezcla continua de aire dependiendo de varios factores como la ocupación y el área. La ventilación comienza a ser efectiva a partir de 4 a 5 renovaciones de aire por hora para un salón de clases con 25 estudiantes y un área de cien metros cuadrados. Otra forma de evaluar la ventilación son los litros de aire por persona por segundo (la unidad de medida expresa cuántos litros de aire se debe entregar por persona cada segundo), que ingresa del exterior, donde sugieren que es necesario 14 litros por persona por segundo para evitar contagios de enfermedades respiratorias, teniendo en cuenta que el riesgo cero de contagio no existe [3], [4]. La mala ventilación está relacionada con el estado de los ocupantes, así lo describe la Organización Mundial de la Salud (OMS), como el síndrome de edificio enfermo, refiriéndose a la relación que hay entre la influencia de la calidad de aire interno y el estado de los ocupantes. Esto involucra el desempeño en el desarrollo de actividades y también su salud respiratoria. Un estudio realizado por investigadores de la Universidad de Harvard y UCLA analizaron los efectos de la temperatura en aulas educativas con el rendimiento académico, encontrando que por cada aumento en grados centígrados se pierde cerca del 1% del rendimiento en el aprendizaje [5], [6]. Por otro lado, un estudio desarrollado en Nueva Gales revela que un cambio brusco en la caída de la temperatura del aula afecta en 1.2% el rendimiento en los exámenes a los estudiantes [7], [8]. La ventilación inteligente se encarga de monitorear y controlar

variables para la captación de aire externo, filtrarlo, para posteriormente inyectarlo al espacio cerrado [9].

En el caso del campus universitario de la Universidad de Cundinamarca sede Fusagasugá, este no cuenta con la ventilación suficiente en sus distintos espacios teniendo en cuenta observaciones de su infraestructura. En general cada aula cuenta con ventanas y una puerta donde no hay ventilación suficiente. El comportamiento del aire caliente es ascendente, es decir que se concentra en las partes altas de un espacio cerrado, mientras que el frío se concentra en las partes bajas. Las edificaciones de cuatro pisos (bloques F y E) cuentan con laboratorios, salas de docentes, oficinas de las facultades, algunos auditorios y salones, donde la ventilación está dada únicamente por ventanas orientadas al centro del edificio, que dan hacia otro salón y no al exterior. Por otro lado, las edificaciones de dos pisos (bloques A, B, C y D), destinados a salas de administrativos, salas de docentes, salones de clases y laboratorios cuentan con ventanas donde difícilmente ingresa aire por su rango de apertura. A su vez, el tejado de estos bloques es muy antiguo, tratándose de tejas de fibrocemento que absorben el calor. Los laboratorios cuentan con extractores y muy pocos con aire acondicionado, ambos están delimitados a funcionar por acción del personal. Los auditorios y salas interactivas presentan la misma condición.

La ventilación inteligente puede propiciar y mantener las condiciones óptimas para mejorar el confort y estancia en función de variables. Plantear un servicio en nube de ventilación inteligente en la Universidad de Cundinamarca sede Fusagasugá, es relevante para evaluar la ventilación de aulas teniendo en cuenta variables ambientales que definen la calidad del aire interno y el confort térmico. El monitoreo de variables que inciden en el síndrome de edificio proporcionan información del estado de ventilación en un espacio cerrado. Es aquí cuando surge la interrogante de ¿cómo puede la tecnología IoT y servicios en nube gestionar un servicio de ventilación inteligente en el campus universitario para analizar una efectiva ventilación?

## **OBJETIVO GENERAL**

Desarrollar una arquitectura en nube para un sistema IoT de ventilación inteligente con el fin de monitorear la calidad de aire y confort térmico, en el caso de estudio definido en la Universidad de Cundinamarca sede Fusagasugá.

## **OBJETIVOS ESPECÍFICOS**

- Caracterizar el servicio IoT de ventilación inteligente teniendo en cuenta funcionamiento y restricciones.
- Realizar una prueba de concepto de un sistema IoT en el aula de docentes del programa de ingeniería electrónica del bloque C con el fin de realizar la medición de la calidad del aire.
- Implementar la arquitectura en un servidor en la nube teniendo en cuenta las variables de monitoreo, procesamiento y almacenamiento.

- Validar la arquitectura teniendo en cuenta la prueba de concepto de un sistema IoT y sus componentes definidos para el servicio en nube del sistema de ventilación inteligente.

## **JUSTIFICACIÓN**

La mala ventilación de instalaciones académicas es un problema que tiene un impacto directo en la salud, bienestar y rendimiento de los ocupantes [7]. Está demostrado que uno de los causantes del estrés en el trabajo de docencia está dado por las condiciones del entorno como el síndrome de edificio enfermo, el cual según la OMS existe de dos tipos: el síndrome de edificio enfermo temporal, presente en circunstancias de reparaciones, mantenimientos o construcción y el permanente, que con el pasar de los años los efectos siguen presente, como dolor de cabeza, fatiga mental, alergias respiratorias e hipersensibilidad [10]. El segundo tipo de síndrome de edificio enfermo se presenta en edificios herméticos como oficinas y aulas y está asociado con la falta de aire fresco. En respuesta los ocupantes reaccionan fisiológica y psicológicamente en áreas cognitivas, comportamentales y hasta emocionales. En el documento de “Cómo afecta el estrés del docente en el desempeño académico del estudiante de UPIBI-IPN” [11], resalta que una de las barreras físicas en el desempeño laboral y académico son la temperatura y la ventilación [9].

La propuesta de arquitectura en nube para un sistema IoT de ventilación inteligente aplicado a una prueba concepto involucra un análisis e identificación de las condiciones de ventilación en el aula de experimentación, apoyándose de la tecnología IoT, herramientas en nube y software para modelar el comportamiento del aire en escenarios de ventilación natural cruzada, monitorear y procesar variables. La implementación de estas herramientas tecnológicas entra en los Objetivos de Desarrollo Sostenibles (ODS) de las Naciones Unidas, en la categoría del objetivo 9 correspondiente a Industria, innovación e infraestructura, que busca promover la industrialización inclusiva y sostenible y el uso de las TIC para incrementar la capacidad productiva de los países, cada uno de los servicios planteados en el proyecto marco se enmarca estos lineamientos.

Una mala calidad de aire interno en las aulas afecta la salud y rendimiento de los ocupantes, la mayoría llegan a superar las concentraciones aceptables de CO<sub>2</sub>, según el estudio titulado “Sistemas y estrategias de ventilación en edificios educativos”.

# ALCANCES Y LIMITACIONES

## Alcances

- El sistema IoT monitorea variables ambientales consideradas en la calidad de aire y confort térmico: temperatura, humedad y concentración de CO2.
- Acceso a recursos en la nube de Amazon con la versión free tier que incluyen servicios de publicación, almacenamiento y procesamiento de datos por un año.
- Implementar una prueba de concepto en el aula de docentes de ingeniería electrónica de la Universidad de Cundinamarca sede Fusagasugá, donde se cuenta con distintos horarios de ocupación y condiciones ambientales variables.
- El modelamiento computacional del aire por medio de simulación 3D de la sala de prueba concepto para analizar el flujo del aire y distribución de calor del aula dependiendo de escenarios de ventilación.
- Manejo de la herramienta Node-RED para configurar y programar nodos que establecen la comunicación de dispositivos IoT a la nube de Amazon.
- Manejo de la herramienta Tailscale para configura una red virtual dentro de la red de la universidad para acceder a la Raspberry Pi.
- Uso de herramientas telemáticas open source como VNC viewer y Putty para acceder al eterno de la Raspberry Pi y poder manejarla de manera remota.
- Manejo de la visualización de los datos procesados en el software Tableau.

## Limitaciones

- Pérdidas de los datos en el sistema IoT al enviarlos a AWS por la red pública de Internet.
- La alimentación del sistema IoT depende de la red eléctrica del aula.
- La versión free tier de la nube de Amazon tiene un límite en el uso recursos para todos los servicios que incluye la capa gratuita. Si el proyecto se quiere escalar a muchos dispositivos o con envíos de grandes volúmenes de datos la versión gratuita no aplica y se comienzan a facturar costos por servicio mensualmente.
- El proyecto no cuenta con la licencia del software de modelado computacional en Ansys, la versión estudiantil se limita a pocas simulaciones y no permite importar o exportar datos de análisis realizados.

# 1. MARCO REFERENCIAL

---

## 1.1. ESTADO DEL ARTE

En esta sección se encuentra la revisión y análisis de los proyectos e investigaciones previas enfocadas en el uso de las tecnologías IoT para la implementación de sistemas de ventilación inteligente, además del uso de los servicios en nube o servidores locales para el monitoreo de variables ambientales como temperatura, humedad y la medición de CO<sub>2</sub>.

### 1.1.1. Sistema de ventilación inteligente para laboratorios

La integración de tecnologías IoT, modelos predictivos y control adaptativo para optimizar la calidad del aire y la eficiencia energética en espacios cerrados como aulas y laboratorios, en lugar de limitarse a inyectar aire frío o caliente según umbrales fijos, permiten la recirculación del aire dentro de rangos preestablecidos, mejorando el confort térmico y reduciendo el consumo de energía en comparación con sistemas ON/OFF. Este trabajo evalúa mediante el uso de sensores empleados para monitorear temperatura, humedad y concentración de CO<sub>2</sub>, junto con la programación de actuadores como ventiladores, extractores y fuentes de calor, facilita una gestión eficiente de los recursos. Además, la integración de algoritmos de aprendizaje automático y control predictivo basado en modelos (MPC, por sus siglas en inglés) permite ajustar dinámicamente la ventilación según la ocupación y condiciones ambientales, reduciendo hasta un 16% el consumo energético en comparación con sistemas tradicionales [12]. Esto permite establecer la base para sistemas que puedan tomar decisiones por sí mismos, mediante la implementación de tecnologías como los servicios de nubes y el procesamiento de la información recopilada por los sensores.

### 1.1.2. Control inteligente de temperatura de un departamento modelo

La implementación de la ventilación natural junto con un sistema que permite determinar si se requiere la apertura de las ventanas del entorno acorde a la presencia de una persona o dependiendo de la temperatura en el ambiente. A diferencia de los sistemas tradicionales, que dependen de ventilación mecánica constante, algunos sistemas combinan ventilación natural y automatizada, regulando el flujo de aire mediante sensores de temperatura, presencia y calidad del aire, junto con la gestión de actuadores como ventiladores y calefactores. Estos sistemas permiten ajustar dinámicamente la ventilación, manteniendo el confort térmico y reduciendo el consumo energético. Además, la orientación arquitectónica y el diseño de aperturas influyen en la eficiencia de la ventilación, considerando variables como velocidad y temperatura del viento [13].

### 1.1.3. Sistemas de acondicionamiento inteligente en edificios verdes

La evolución de los sistemas de climatización en edificios ecológicos e inteligentes, destacan por su impacto en la eficiencia energética y la reducción de la huella de carbono. El aumento de la urbanización ha incrementado la demanda de aire acondicionado, lo que hace fundamental mejorar su consumo mediante tecnologías como ventilación inteligente con sensores IoT, redes inteligentes y estrategias de Respuesta a la Demanda (DR, por sus siglas en inglés). Se revisan soluciones basadas en energías renovables, almacenamiento térmico y diseño arquitectónico eficiente, como techos verdes y materiales aislantes. Estudios de caso en diferentes países han demostrado que la combinación de IA, control en la nube y ventilación pasiva no solo optimiza el confort térmico, sino que también reduce el consumo energético y las emisiones contaminantes [14].

## **1.2. FUNDAMENTOS TEÓRICOS**

### **1.2.1. IoT (Internet of Things)**

La Internet de las Cosas (IoT) es un concepto que ha evolucionado para representar la interconexión de una amplia gama de dispositivos a través de redes, permitiendo la comunicación entre electrodomésticos, objetos y estructuras mediante interfaces simples. Esta tecnología, basada en la recopilación y análisis de datos, busca mejorar la comodidad y eficiencia en la vida cotidiana, posibilitando la monitorización en tiempo real y la creación de nuevas experiencias y oportunidades para individuos, empresas y ciudades. El término "Internet de las Cosas" fue acuñado en 1999 por el profesor Kevin Ashton del Massachusetts Institute of Technology (MIT, por sus siglas en inglés), y desde entonces ha experimentado un crecimiento significativo. La tecnología IoT permite la comunicación automática entre objetos y la transmisión de datos sin intervención humana, lo que posibilita una mayor eficiencia operativa y capacidades de procesamiento propio en los dispositivos [15].

### **1.2.2. Computación en la Nube**

La computación en la nube es un enfoque de procesamiento de datos basado en Internet que proporciona recursos compartidos de manera flexible a los usuarios según sus necesidades. Esto permite a las organizaciones centrarse en sus actividades principales sin tener que preocuparse por la infraestructura informática. Por lo tanto, la nube ofrece acceso y almacenamiento de datos y programas a través de Internet, eliminando la necesidad de depender exclusivamente del disco duro de una computadora [16].

### **1.2.3. Amazon Web Services (AWS)**

Amazon Web Services (AWS, por sus siglas en inglés) es una plataforma de computación en la nube líder que ofrece una amplia gama de servicios personalizables para satisfacer las necesidades individuales de los clientes. Con centros de datos distribuidos globalmente, garantiza una disponibilidad del servicio cercana al 100% y reduce las latencias, especialmente en aplicaciones que requieren ejecución en tiempo real. Ofrece servicios que abarcan desde infraestructura básica hasta tecnologías avanzadas como IoT, Big Data y AI. La seguridad es una prioridad, con un conjunto robusto de herramientas y estándares de cifrado para proteger la integridad de los datos y cumplir con los requisitos de confidencialidad más estrictos [17].

#### 1.2.4. Tipos de datos en la nube

- **Datos Estructurados:** Se organiza en un formato específico y predefinido, lo que facilita su almacenamiento y manipulación. Por ejemplo: incluyen bases de datos relacionales y hojas de cálculo.
- **Datos No Estructurados:** No siguen un formato predefinido y pueden incluir texto, imágenes, videos, archivos de audio, etc. Debido a su naturaleza diversa, a menudo son más difíciles de gestionar y analizar que los datos estructurados.
- **Datos Semi-Estructurados:** No cumplen completamente con la estructura rígida de los datos estructurados, pero tienen cierto grado de organización, como datos XML o JSON.
- **Datos de Series Temporales:** Se recopilan y organizan en función del tiempo, lo que los hace ideales para análisis predictivos y tendencias. Estos incluyen datos meteorológicos, registros de actividad del usuario y datos financieros.
- **Datos Geoespaciales:** Están asociados con ubicaciones geográficas específicas y se utilizan comúnmente en aplicaciones de mapeo, navegación y logística. Pueden incluir mapas digitales, imágenes satelitales y datos de GPS.
- **Datos Biométricos:** Se refieren a características físicas o comportamentales únicas de individuos, como huellas dactilares, reconocimiento facial, patrones de voz, etc. Son utilizados en sistemas de seguridad y autenticación.
- **Datos Multimedia:** Incluyen contenido multimedia como imágenes, videos y audio. Se utilizan en aplicaciones de almacenamiento y transmisión de medios, redes sociales, plataformas de streaming y mucho más.
- **Datos Transaccionales:** Generados como resultado de transacciones comerciales o financieras. Incluyen registros de ventas, historiales de transacciones bancarias y registros de inventario. Son fundamentales para la gestión empresarial y el análisis financiero.

- **Datos de Registro:** Incluyen registros de actividad del sistema, como registros de eventos, registros de acceso y registros de auditoría. Son esenciales para el monitoreo de seguridad, la resolución de problemas y la conformidad con las regulaciones [17].

### 1.2.5. Tipos de almacenamiento en la nube

- **Almacenamiento de Objetos:** Está diseñado para almacenar y recuperar grandes cantidades de datos no estructurados, como imágenes, videos, archivos de audio y documentos. Utiliza una arquitectura de almacenamiento distribuido y se accede a los datos a través de interfaces de programación de aplicaciones (API) estándar.
- **Almacenamiento de Archivos:** Permite almacenar y organizar archivos de datos de manera jerárquica, similar a la estructura de los sistemas de archivos locales. Proporciona acceso compartido a archivos y directorios a través de protocolos como NFS (Network File System) y SMB (Server Message Block).
- **Almacenamiento de Bloques:** Los datos se almacenan en bloques individuales, y cada bloque se trata como un dispositivo de almacenamiento independiente. Es ideal para aplicaciones que requieren un alto rendimiento y baja latencia, como bases de datos y sistemas de archivos de alto rendimiento [17].

### 1.2.6. Tipos de procesamiento

- **Procesamiento en Tiempo Real:** Este tipo de procesamiento implica analizar datos a medida que se generan, sin demoras perceptibles. Es esencial para aplicaciones que requieren respuestas instantáneas, como sistemas de detección de fraudes, monitoreo de redes y análisis de datos de sensores IoT.
- **Procesamiento Cuasirreal:** El procesamiento cuasirreal se sitúa entre el procesamiento de datos en tiempo real y el procesamiento por lotes. Se refiere a la capacidad de procesar datos con una latencia mínima y una alta frecuencia de actualización, aunque no necesariamente en tiempo real absoluto.
- **Procesamiento Predictivo:** Implica el uso de modelos y algoritmos para predecir eventos futuros basados en datos históricos. Es esencial para aplicaciones de análisis predictivo, pronóstico de ventas, mantenimiento predictivo y optimización de inventario.
- **Procesamiento Prescriptivo:** Se refiere al uso de algoritmos para recomendar acciones óptimas basadas en datos y objetivos específicos. Es útil para aplicaciones de toma de decisiones automatizadas, recomendación de productos y optimización de procesos empresariales.

- **Procesamiento por Lotes:** Este tipo de procesamiento implica la ejecución de tareas en lotes o grupos, sin necesidad de interacción en tiempo real. Es ideal para aplicaciones que pueden tolerar cierta demora en el procesamiento, como análisis de grandes conjuntos de datos históricos y procesamiento de informes.
- **Procesamiento de Aprendizaje Automático (Machine Learning):** Se refiere al uso de algoritmos para aprender de datos y realizar predicciones o tomar decisiones sin intervención humana. Es esencial para aplicaciones de inteligencia artificial, análisis predictivo y automatización de procesos basados en datos.
- **Procesamiento Basado en Servicios (Serverless):** Este enfoque permite ejecutar código en la nube sin preocuparse por la administración de servidores subyacentes. Los proveedores de servicios en la nube gestionan automáticamente la infraestructura, lo que permite a los desarrolladores centrarse en la lógica de la aplicación [17].

### 1.2.7. Ventilación Inteligente.

La ventilación inteligente se refiere a sistemas de ventilación que utilizan tecnología avanzada, como sensores, algoritmos de control y conectividad a Internet, para mejorar la calidad del aire interior y aumentar la eficiencia energética. Estos sistemas pueden monitorear constantemente la calidad del aire, ajustar automáticamente el flujo de aire según sea necesario y proporcionar retroalimentación en tiempo real sobre las condiciones ambientales. también conocida como ventilación de demanda controlada (DCV, por sus siglas en inglés).

A diferencia de los sistemas de ventilación convencionales que suministran aire exterior a un ritmo fijo, la ventilación inteligente monitorea continuamente parámetros como los niveles de CO<sub>2</sub>, humedad, compuestos orgánicos volátiles (COVs) y la ocupación del edificio. Luego, regula el caudal de aire exterior para mantener una calidad de aire interior óptima, al mismo tiempo que evita una ventilación excesiva innecesaria [18].

### 1.2.8. Mecánica de Fluidos

La mecánica de fluidos es una rama de la física que estudia el comportamiento de los fluidos (líquidos y gases) en reposo y en movimiento. Este campo se divide principalmente en dos áreas: la estática de fluidos, que analiza los fluidos en equilibrio o en reposo, y la dinámica de fluidos, que se centra en los fluidos en movimiento. La dinámica de fluidos, en particular, es fundamental para estudiar fenómenos como el flujo de aire en sistemas de ventilación, la transferencia de calor y masa, y el comportamiento de gases en espacios cerrados, como el caso de las aulas de una institución educativa [19].

Dentro de la dinámica de fluidos, existen ecuaciones matemáticas como las ecuaciones de Navier- Stokes, que describen el movimiento de los fluidos y consideran factores como la viscosidad, densidad y presión. Estas ecuaciones son esenciales en la simulación computacional del flujo de aire, ya que permiten predecir la velocidad, la dirección y la distribución de variables como la temperatura y la humedad dentro de un espacio cerrado[19]. En el contexto del proyecto, la mecánica de fluidos proporciona el marco teórico para modelar y analizar el flujo de aire en aulas mediante herramientas de dinámica de fluidos computacional (CFD, por sus siglas en inglés), como ANSYS Fluent. Esto permite identificar cómo se comporta el aire en diferentes condiciones, optimizando la colocación de sensores para la medición de las variables de calidad del aire.

### **1.2.9. Calidad del aire y cómo se mide**

La calidad del aire se refiere a la medida de pureza o contaminación del aire en un entorno específico, que puede ser tanto en exteriores como en interiores. Esta calidad depende de la presencia y concentración de diversos contaminantes que afectan la salud humana, el confort y la productividad en ambientes cerrados, así como el equilibrio ecológico en exteriores. En interiores, la calidad del aire se evalúa en función de contaminantes como el dióxido de carbono (CO<sub>2</sub>), el monóxido de carbono (CO), los compuestos orgánicos volátiles (COVs), partículas en suspensión y niveles de humedad [20].

La medición de la calidad del aire se realiza mediante sensores especializados que detectan la concentración de estos contaminantes en partes por millón (ppm), microgramos por metro cúbico ( $\mu\text{g}/\text{m}^3$ ), o porcentajes en el caso de la humedad relativa. Los sensores de CO<sub>2</sub>, por ejemplo, miden la cantidad de dióxido de carbono en el aire, y cuando esta concentración supera un umbral determinado (normalmente 1,000 ppm en interiores), puede indicar una ventilación insuficiente [21]. Otros parámetros, como la humedad y la temperatura, también son importantes ya que afectan la sensación de confort y la propagación de ciertos contaminantes.

La calidad del aire en interiores se evalúa a menudo utilizando índices de calidad del aire (ICA), que combinan varias lecturas de contaminantes en una escala para facilitar su interpretación. Estos índices ayudan a clasificar la calidad del aire en categorías como "Buena", "Moderada", "No saludable" o "Peligrosa", alertando a los ocupantes sobre la necesidad de ventilación o el uso de sistemas de filtración. En el contexto de un sistema de ventilación inteligente, estas mediciones son esenciales para que el sistema IoT responda automáticamente a cambios en la calidad del aire y mantenga el ambiente del aula dentro de los estándares de salud y confort [21].

### 1.2.10. Estándares de Calidad del Aire

Los estándares de calidad del aire son un conjunto de normas y directrices establecidas por organismos de salud y medioambientales que determinan los niveles máximos permitidos de contaminantes para proteger la salud humana y el medio ambiente. Estos estándares establecen límites específicos para una variedad de contaminantes, tales como el dióxido de carbono (CO<sub>2</sub>), monóxido de carbono (CO), partículas en suspensión (PM<sub>10</sub> y PM<sub>2.5</sub>), dióxido de azufre (SO<sub>2</sub>), óxidos de nitrógeno (NO<sub>x</sub>) y ozono (O<sub>3</sub>), entre otros.

Organismos como la Organización Mundial de la Salud (OMS), la Agencia de Protección Ambiental de Estados Unidos (EPA, por sus siglas en inglés) y en Colombia, el Ministerio de Ambiente y Desarrollo Sostenible, establecen los límites permitidos para cada contaminante en función de sus efectos sobre la salud humana. Por ejemplo, la OMS recomienda que los niveles de CO<sub>2</sub> en interiores no superen las 1,000 partes por millón (ppm) para evitar síntomas como la fatiga y la reducción de la concentración en ocupantes de espacios cerrados [6], [9], [22].

- **Emisiones de contaminantes:** La cantidad de contaminantes emitidos, ya sea de fuentes industriales, transporte vehicular, o combustión de combustibles fósiles, impacta directamente en la concentración de contaminantes en el aire.
- **Condiciones meteorológicas:** Factores como la presencia y dirección del viento, la temperatura, la radiación solar, y las precipitaciones afectan cómo se dispersan y reaccionan los contaminantes en la atmósfera, además de la altura de la capa de mezcla atmosférica.
- **Interacciones químicas en la atmósfera:** Reacciones químicas, como la oxidación, hidrólisis y reacciones fotoquímicas, pueden transformar contaminantes primarios en contaminantes secundarios, como el ozono o el trióxido de azufre (SO<sub>3</sub>), afectando la calidad del aire.
- **Topografía:** La geografía y disposición del terreno pueden facilitar o dificultar la dispersión de contaminantes, acumulándose en ciertas áreas y afectando localmente la calidad del aire [22], [23].

### 1.2.11. Prueba de concepto

Una prueba de concepto (POC, por sus siglas en inglés) es un proceso crucial en el desarrollo y la implementación de ideas innovadoras. Se trata de una investigación preliminar que busca validar la viabilidad técnica y practicidad de un concepto o proyecto específico antes de su implementación a gran escala. A través de una POC, se realizan pruebas limitadas en un entorno controlado para demostrar que una idea puede funcionar según lo previsto, identificando posibles desafíos y refinando el enfoque. Esta etapa permite a los creadores y desarrolladores tomar decisiones informadas sobre la continuación del proyecto, minimizando riesgos y maximizando la eficiencia en el proceso de desarrollo [24], [25].

Las siguientes tablas muestran los diferentes protocolos IoT y de red que se pueden implementar en sistemas IoT.

**Tabla 1. Protocolos de red. Fuente: elaboración propia**

Tecnología	Seguridad	Rango	Latencia	Velocidad
<b>Bluetooth</b>	(AES-128)	10 - 100 m	(~100 ms)	1 - 3 Mbps
<b>ZigBee</b>	(AES-128)	10 - 100 m	(~30 ms)	250 kbps
<b>Z-Wave</b>	(AES-128)	30 - 100 m	(~100 ms)	100 kbps
<b>6LoWPAN</b>	Alta (AES-128)	10 - 100 m	(~20-50 ms)	250 kbps
<b>SigFox</b>	(cifrado propio)	30-50 km (rural), 3-10 km (urbano)	(~1s)	100 bps
<b>Wi-Fi</b>	(WPA2/WPA3)	50 - 100 m	(~10-50 ms)	54 Mbps - 9.6 Gbps
<b>5G</b>	(AES-256, IPsec)	0 - 10 km	(~1 ms)	1 - 10 Gbps

**Tabla 2. Protocolos de IoT. (la información está basada en (Amazon AWS, 2024)).**

<b>HTTP</b>	El Hypertext Transfer Protocol, es un protocolo estándar para la comunicación entre cliente y servidor, su funcionamiento se basa en la interacción de solicitud y respuesta cuando un navegador o sensor IoT manda una solicitud al servidor de la web y este envía datos de respuesta.
<b>MQTT</b>	El Message Queuing Telemetry Transport, es un protocolo de mensajería a bajo consumo para sistemas IoT con recursos limitados. En este se debe suscribir un cliente IoT para publicar mensajes en un bróker IoT que los demás dispositivos del sistema reciben.
<b>DDS</b>	El Data Distribution Service, es un protocolo de transmisión de datos en tiempo real parecido a MQTT, orientado a aplicaciones en tiempo real que requieren alta confiabilidad, como automatización, conducción de vehículos autónomos, etc.
<b>XMPP</b>	El Extensible Messaging and Presence Protocol, es un protocolo basado en XML para comunicaciones en tiempo real por medio del modelo cliente – servidor, junto con encriptación de extremo a extremo y control de la disponibilidad de los dispositivos IoT dentro del sistema.
<b>AMQP</b>	El Advanced Message Queuing Protocol, es un protocolo de mensajería para comunicaciones robustas no entre dispositivos sino entre sistemas distribuidos en una red. Ofrece mensajería de una forma más compleja por medio del enrutamiento de la información para mayor seguridad y fiabilidad.
<b>CoAp</b>	El Constrained Application Protocol, es un protocolo de capa de aplicación basado en el modelo de HTTP, con optimizaciones para dispositivos IoT de poca capacidad de procesamiento y bajos recursos. Utiliza UDP para la transmisión de información más eficiente, rápida y ligera.

## 2. DISEÑO METODOLÓGICO

### 2.1. METODOLOGÍA

La metodología por seguir es de tipo cascadas, está dividida por fases que contienen las actividades respectivas. Se plantean 3 fases: caracterización de la ventilación, implementación y validación. En la Ilustración 1, se describe el aporte de este trabajo en el proyecto marco.

**Fase 1 - Planificación del sistema:** Contiene la caracterización de ventilación de la sala de prueba de concepto, por medio de observaciones de la infraestructura y modelado con las medidas reales por medio del software Ansys Fluent. Se evalúan tres escenarios de ventilación para obtener conclusiones preliminares. También contiene la caracterización de un servicio de ventilación inteligente de acuerdo con un objetivo y la definición de variables del sistema. Finalmente presenta el escenario IoT.

**Fase 2 – Análisis de infraestructura TIC:** Está fase es de implementación, abarca la selección del hardware e instalación del sistema IoT y diseño de la arquitectura en nube para el servicio de ventilación inteligente.

**Fase 3 – Análisis de la implementación de servicios TIC:** Corresponde a la fase de validación. Aquí se valida la arquitectura de acuerdo con los requerimientos del caso de uso: recolección, almacenamiento, análisis y procesamiento de los datos.

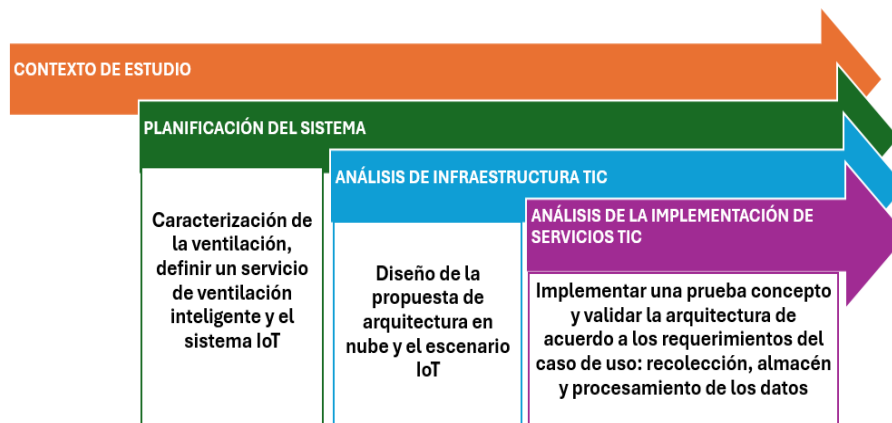


Ilustración 1. Fases metodológicas. Fuente: elaboración propia.

**Fase 1:** Caracterización de la ventilación existente en el aula de la prueba de concepto.

Actividad 1: Tomar medidas de las dimensiones del aula de prueba de concepto y la herramienta de modelado computacional de fluidos: el software Ansys en su versión estudiantil.

Actividad 2: Introducir en la herramienta las dimensiones y materiales del aula y simular distintos escenarios de ventilación: ventanas y puerta abiertas, ventanas abiertas y puertas cerradas y finalmente puertas y ventanas cerradas.

Actividad 3: Analizar los renderizados para identificar el flujo del viento, la distribución de calor en el aula y definir la mejor ubicación del sistema IoT encargado de la obtención de los datos.

Actividad 4: Caracterizar un servicio de ventilación inteligente para el aula de prueba concepto teniendo en cuenta: escenario IoT, variables en torno al objetivo de este.

**Fase 2:** Esta fase contiene la implementación del sistema IoT y el diseño de la arquitectura en nube.

Actividad 1: Configuración de la placa y los sensores que van a medir variables de calidad de aire (concentración de CO<sub>2</sub>) y confort térmico (temperatura y humedad), de acuerdo con el tipo de dato, interfaz de comunicación, rangos de medición dentro del aula y consumo de energía.

Actividad 2: Instalación del sistema IoT en el aula de prueba de concepto.

Actividad 3: Esquematizar el diseño de la arquitectura en nube de acuerdo con los servicios seleccionados para el caso de uso.

Actividad 5: Conectar el sistema IoT a la arquitectura en nube.

**Fase 3:** Validación. En esta fase final, se van a realizar las pruebas y observaciones operativas de la arquitectura en nube diseñada de acuerdo con los requerimientos del caso de uso.

Actividad 1: Protocolo de envío de datos a la nube y recepción de estos.

Actividad 2: Tipos de almacenamiento de los datos de acuerdo con el uso que tengan.

Actividad 3: Procesamiento en nube y preparación para la visualización.

Actividad 4: Definir el alcance de la arquitectura, posibles problemas y limitaciones de la arquitectura en nube del sistema, de acuerdo con los servicios seleccionados.

### 3. DESARROLLO DEL TRABAJO

#### 3.1. CARACTERIZACIÓN

En esta sección se encuentra la caracterización de ventilación que presenta el aula de la prueba de concepto (sala de docentes de ingeniería electrónica de la Universidad de Cundinamarca sede Fusagasugá) por medio de la herramienta de software Ansys. También se expone el planteamiento de un servicio de ventilación inteligente.

En primera instancia se realizó la caracterización de los materiales del aula (materiales de las paredes, techo y ventanas), ya que en el momento de realizar la simulación dentro de Ansys Fluent se debe tener en cuenta algunas características importantes para que se realice de manera correcta el modelado, como se observa en la

Tabla 3. Parámetros de simulación de Ansys Fluent. Fuente: elaboración propia.

Parámetros de simulación		
Material	Propiedades	Descripción
Ladrillo rojo	Densidad	1700 kg / m <sup>3</sup>
	Punto de fusión	1727 ° C
	Conductividad térmica	1,31 W / mK
	Capacidad calorífica	800 J / g K
Teja fibrocemento	Densidad	1.200 kg/m <sup>3</sup>
	Punto de fusión	No aplica
	Conductividad térmica	0,46 W / mK
	Capacidad calorífica	1 J / g K
Vidrio	Densidad	2,9 g / cm <sup>3</sup>
	Punto de fusión	1040 ° C
	Conductividad térmica	0,24 W / mK
	Capacidad calorífica	0,84 J / kg K
		Un ladrillo rojo es un bloque de construcción hecho principalmente de arcilla cocida a altas temperaturas, lo que le da su característico color rojizo [26].
		El fibrocemento es un material compuesto de cemento y fibras que ofrece resistencia, durabilidad y buen aislamiento térmico, usado comúnmente en techos y revestimientos [27].
		El vidrio es un material sólido, transparente y frágil, compuesto principalmente por sílice (arena), carbonato de sodio y caliza, fundidos a alta temperatura [28].

Estos datos son relevantes para la simulación del escenario, ya que en el entorno de ANSYS Fluent establecen variables fundamentales para la creación de las simulaciones. Además de esto, se elaboró una tabla comparativa de los tiempos de ejecución de acuerdo con las partículas generadas, estas partículas son pequeños elementos (sólido, líquido o gaseoso) suspendido en el aire usadas para simular el comportamiento de la dispersión de contaminantes dentro del espacio simulado, para este caso se establecieron 20.000 partículas por escenarios, esto se determinó, mediante el documento de guía de usuario de Ansys en la cual se establece que para espacios pequeños la cantidad de partículas necesarias varía entre 5.000 a 20.000 para una resolución media en los resultados [29], lo que permitió determinar que el número óptimo de iteraciones necesarias para obtener resultados estables, un resultado estable se considera cuando las variables de interés; concentración de partículas, flujo de aire, temperatura, se mantienen

constantes con más iteraciones o más partículas simuladas [29]. Como se evidencia en la Tabla 4, esta información se obtuvo a partir de la simulación de diferentes escenarios en los cuales se realizó la variación de las iteraciones y partículas necesarias.

Tabla 4. Conteo de iteraciones para simulación de escenarios. Fuente: elaboración propia

Simulación			
# de iteraciones	Partículas	Tiempo (min)	Tiempo total (20000 partículas)
100	99	3	10 horas
200	198	4	7 horas
300	296	5	5 horas 42 min
500	495	6	4 horas
1000	985	20	6 horas 45 min

### 3.1.1. Caracterización de la ventilación existente.

Se plantean diferentes escenarios de ventilación para evaluar el comportamiento del flujo de las corrientes de aire y la distribución de calor dentro del aula de prueba de concepto haciendo uso de la herramienta Ansys Fluent. Dentro del software de simulación Ansys, primero se realizó el modelado de las dimensiones del aula de profesores del programa de Ingeniería electrónica, ubicada en el bloque C, en la Universidad de Cundinamarca, sede Fusagasugá, como se observa en la Ilustración 2,6,7,8, a partir de esto se plantearon 3 escenarios:

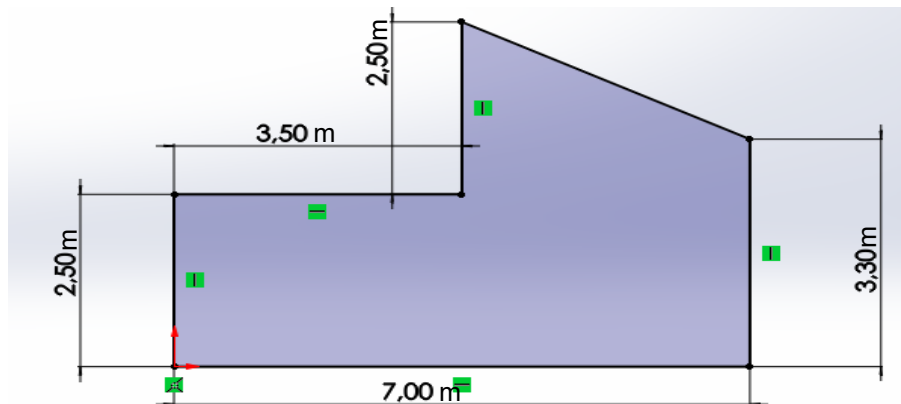


Ilustración 2. Plano 2D de aula de profesores, bloque C. Fuente: elaboración propia

En la Ilustración 2 se evidencia el modelado en plano 2D de la sala para la prueba de concepto, junto con la Ilustración 3, en las cuales se realiza la implementación de las medidas aproximadas del aula.

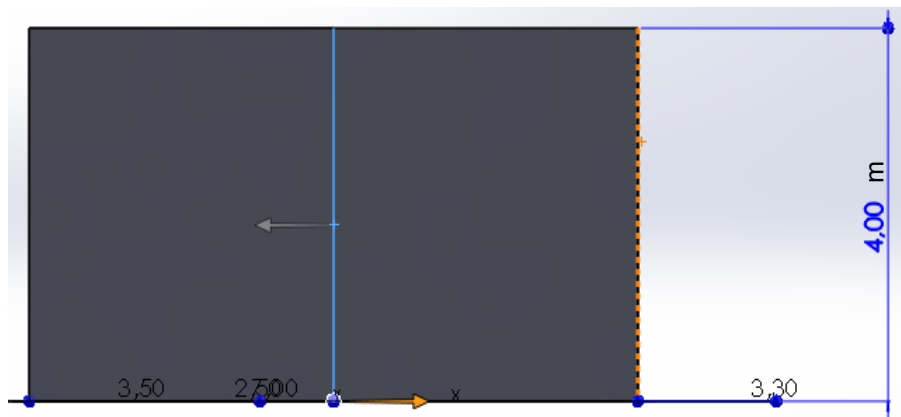


Ilustración 3. Plano 2D de aula de profesores, bloque C. Fuente: elaboración propia.

Después de hecho el modelado 2D del aula se procede a realizar el modelado 3D con las medidas establecidas anteriormente, además de esto se agregan las diferentes entradas y salidas de aire, en este caso las puertas y ventanas presentes en el aula, las cuales se tendrán en cuenta para la posterior simulación, en la Ilustración 4 e Ilustración 5 se puede observar el escenario final en el que se realizara la simulación del comportamiento de las partículas.

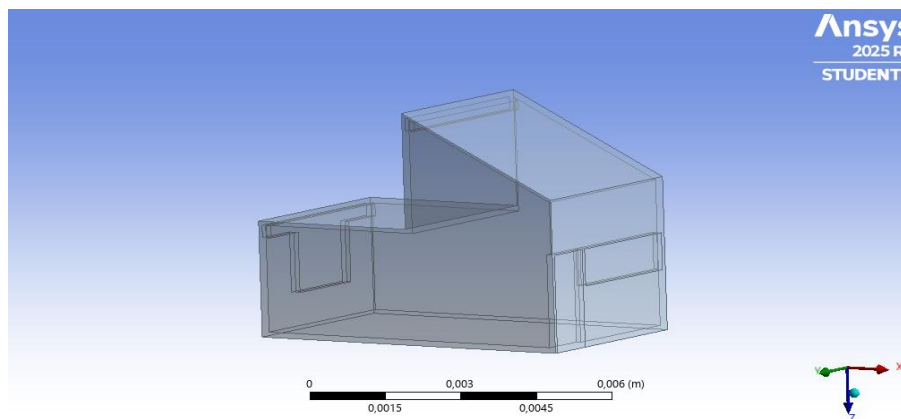
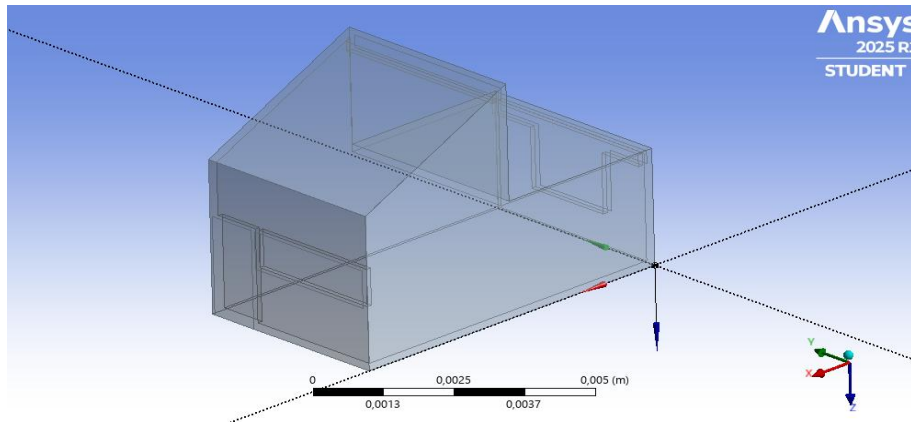


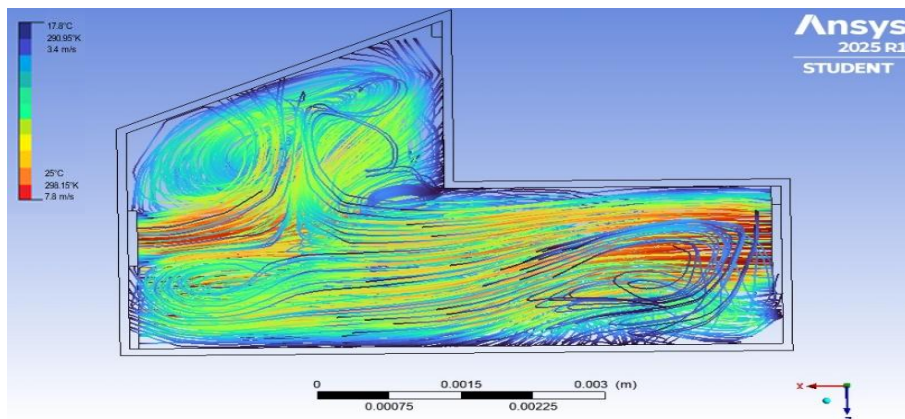
Ilustración 4. Modelado 3D de aula de profesores, bloque C. Fuente: elaboración propia.



**Ilustración 5. Modelado de aula de profesores bloque C. Fuente: elaboración propia.**

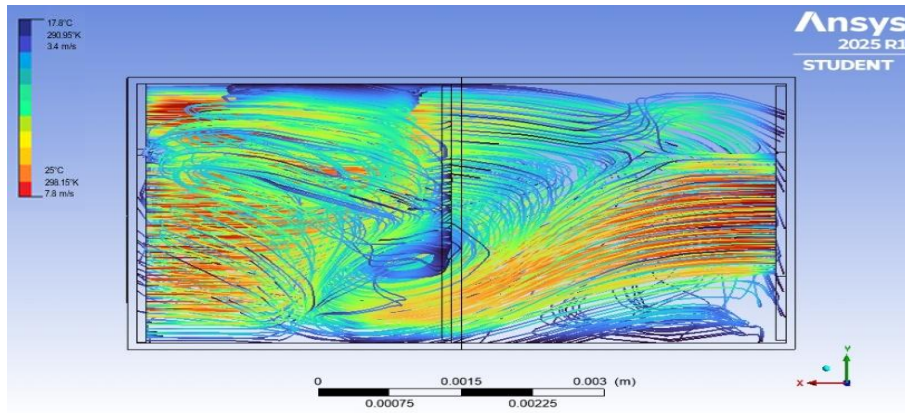
Una vez realizado el modelado 3D del aula de la prueba de concepto, se realizó la simulación de la circulación de viento interna. Estas simulaciones se realizaron como se menciona anteriormente en 3 escenarios, los cuales son escenario 1 (puerta y ventanas abiertas), como evidencia en la Ilustración 6-9, escenario 2 (puerta cerrada y ventanas abiertas) como se observa en la Ilustración 10-13 y escenario 3 (puerta abierta ventanas cerradas) como se observa en la Ilustración 14-17.

### 3.1.1.1 Escenario de ventilación 1 (Puerta y ventanas abiertas).



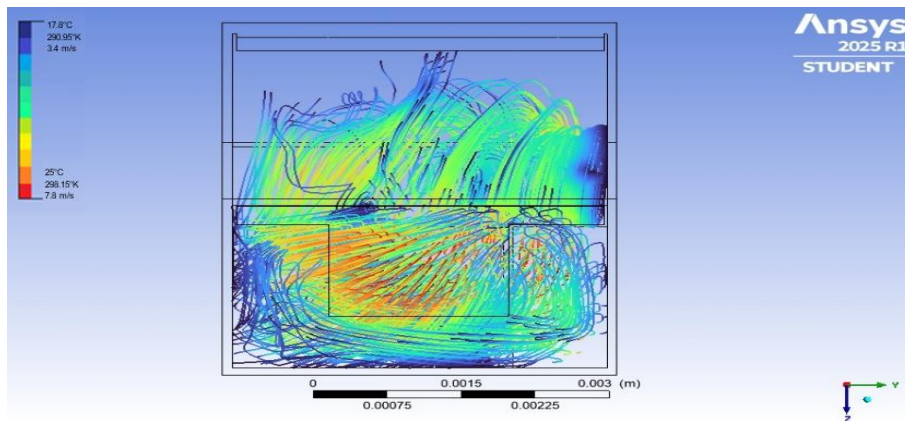
**Ilustración 6. Vista lateral escenario de ventilación 1 (Puerta y ventanas abiertas). Fuente: elaboración propia.**

El comportamiento del flujo de aire permitió observar que la mayor parte del aire caliente y de mayor velocidad proviene de las ventanas ubicadas en la parte trasera del aula, así como de las que se encuentran en la parte frontal, junto a la puerta.



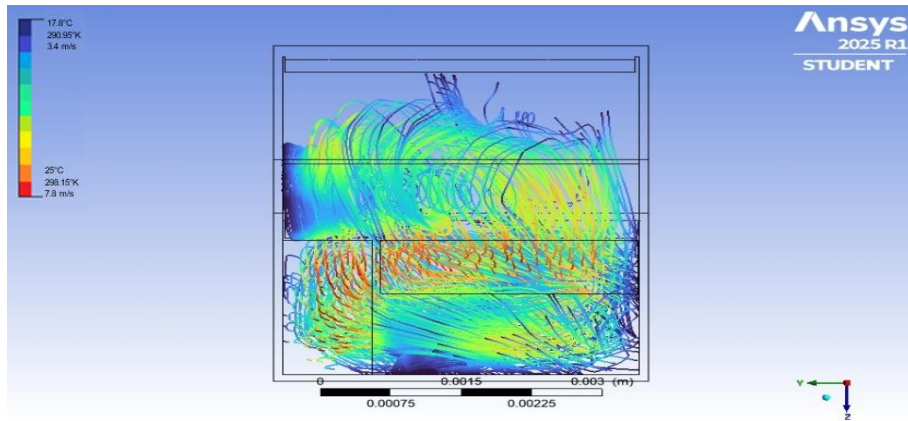
**Ilustración 7. Vista superior escenario de ventilación 1 (Puerta y ventanas abiertas). Fuente: elaboración propia.**

El comportamiento del flujo de aire desde una vista desde la parte superior permite corroborar lo anteriormente mencionado en la vista lateral del aula.



**Ilustración 8. Vista trasera escenario de ventilación 1 (Puerta y ventanas abiertas). Fuente: elaboración propia.**

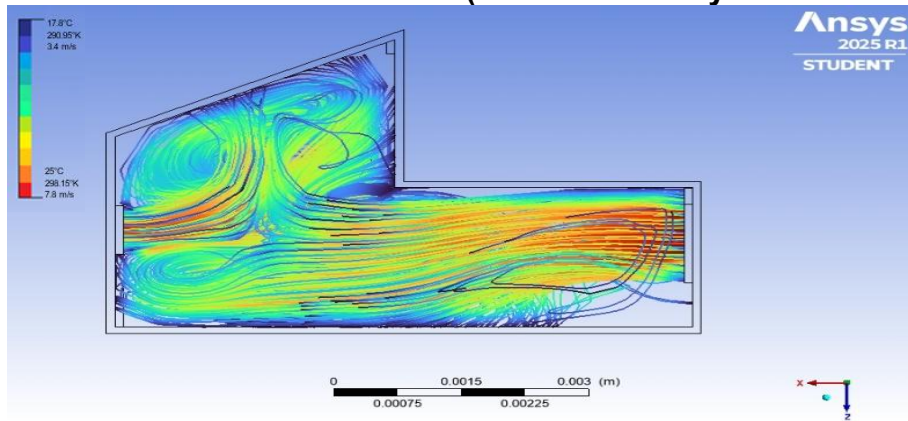
El mayor flujo de partículas de aire se concentra en la zona central del aula y en los alrededores de las fuentes de ingreso del aire, en la parte superior no se observa un flujo significativo, lo que podría indicar una circulación inadecuada del aire dentro del aula.



**Ilustración 9. Vista frontal escenario de ventilación 1 (Puerta y ventanas abiertas).** Fuente: *elaboración propia.*

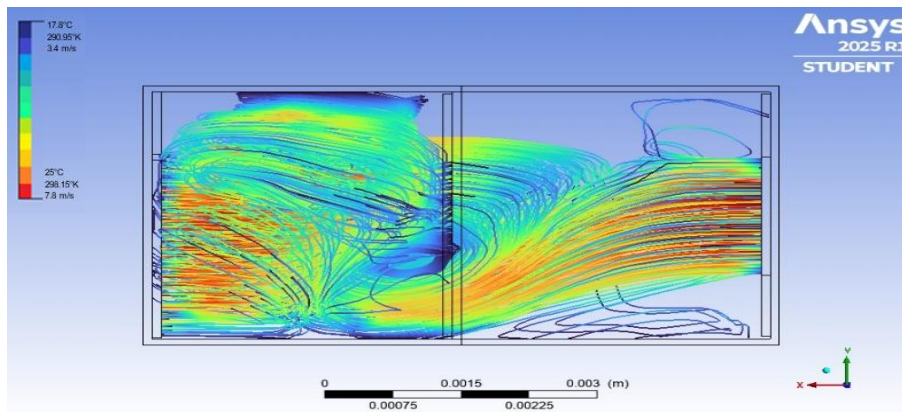
Se puede evidenciar lo anteriormente mencionado, se carece de un flujo adecuado en la parte superior del aula.

### 3.1.1.2 Escenario de ventilación 2 (Puerta cerrada y ventanas abiertas):



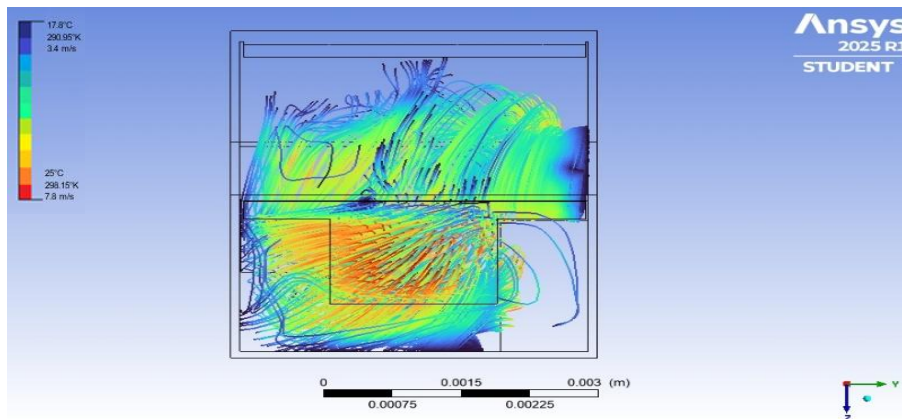
**Ilustración 10. Vista lateral escenario de ventilación 2 (Puerta cerrada y ventanas abiertas).** Fuente: *elaboración propia.*

Al igual que en el escenario 1, se observa que la mayor parte del flujo de aire proviene de las ventanas ubicadas en la parte trasera y frontal del aula. Sin embargo, en este caso, se observa una zona en la parte posterior donde no se detecta flujo de partículas frías, lo que indica la presencia de un área con baja o nula circulación de aire.



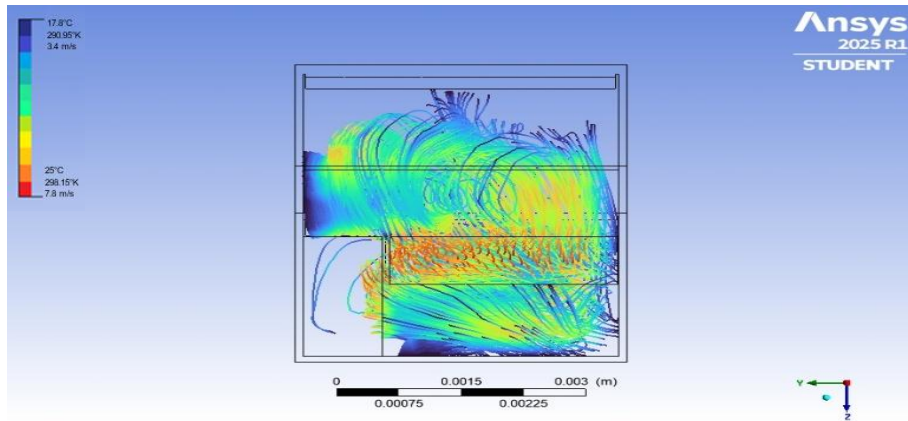
**Ilustración 11. Vista superior escenario de ventilación 2 (Puerta cerrada y ventanas abiertas). Fuente: elaboración propia.**

Contrario a lo observado en el escenario 1, este caso presenta una disminución notable en la circulación de partículas en la parte posterior del aula. La vista superior revela una concentración elevada de flujo en las zonas frontal y central, lo que indica una distribución no homogénea del aire.



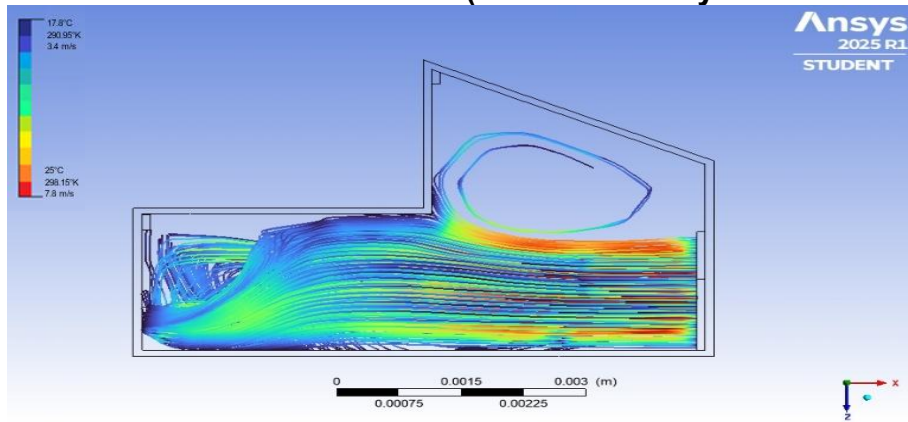
**Ilustración 12. Vista trasera escenario de ventilación 2 (Puerta cerrada y ventanas abiertas). Fuente: elaboración propia.**

Se mantiene una deficiente circulación de aire en la parte superior del aula. Asimismo, se identifica una escasa o nula ventilación en el área de la puerta frontal, lo que podría indicar una acumulación del flujo en las zonas central y frontal (en la ventana), según lo observado en la vista superior, esto mismo se observa en la vista frontal Ilustración 13.



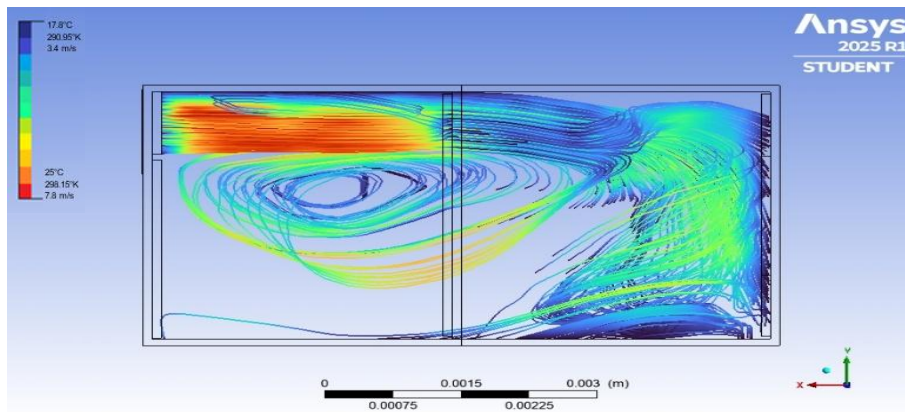
**Ilustración 13. Vista frontal escenario de ventilación 2 (Puerta cerrada y ventanas abiertas).** Fuente: elaboración propia.

### 3.1.1.3 Escenario de ventilación 3 (Puerta abierta y ventanas cerradas)



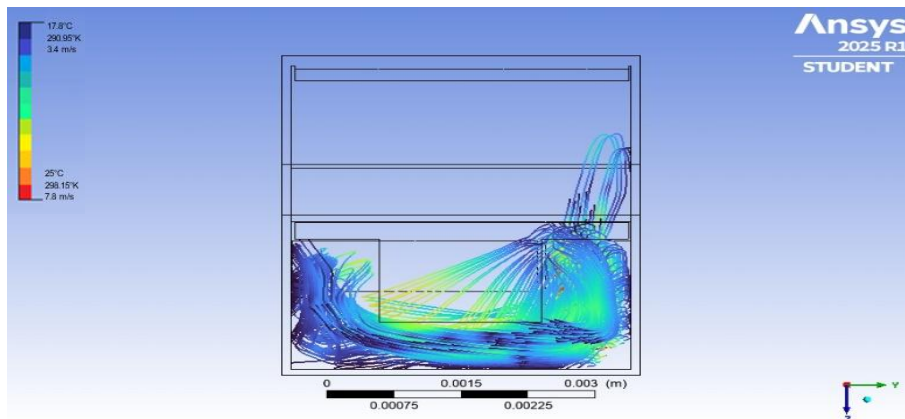
**Ilustración 14. Vista lateral escenario de ventilación 3 (Puerta abierta y ventanas cerradas).** Fuente: elaboración propia.

El comportamiento del flujo de partículas muestra una distribución diferente respecto a los casos anteriores, en este escenario, las partículas frías se concentran principalmente en las áreas posterior y central del aula, mientras que en la zona frontal se observa una mayor presencia de partículas cálidas, la ausencia de movimiento en la parte superior confirma una circulación ineficiente del aire en el aula.



**Ilustración 15. Vista superior escenario de ventilación 3 (Puerta abierta y ventanas cerradas). Fuente: elaboración propia.**

Se observa una acumulación de partículas en la parte posterior del aula, mientras que las regiones frontal y central del lado derecho presentan una circulación de aire mínima. Por otra parte, el lado izquierdo muestra un flujo elevado de partículas cálidas, que alcanza hasta la mitad del aula, evidenciando una distribución asimétrica del aire en el espacio.



**Ilustración 16. Vista trasera escenario de ventilación 3 (Puerta abierta y ventanas cerradas). Fuente: elaboración propia.**

Se evidencia una circulación de aire baja en la zona posterior del aula, y prácticamente nula en la parte superior. Asimismo, las partículas presentes en estas áreas presentan temperaturas bajas, lo que refuerza la hipótesis de una ventilación ineficiente en dichos sectores.

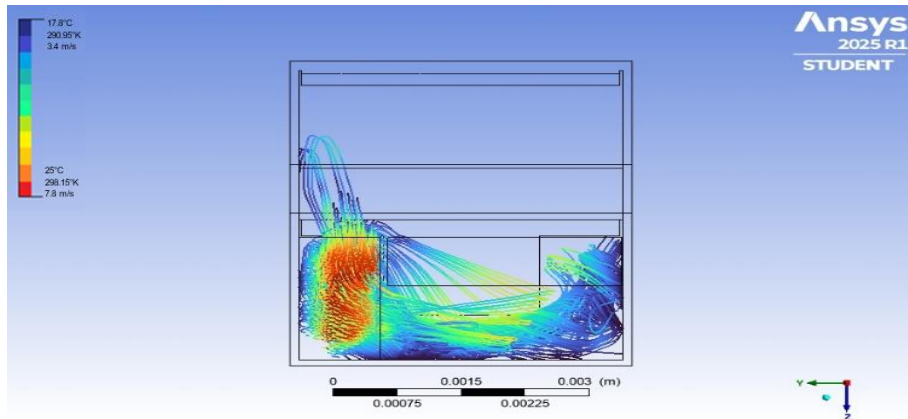


Ilustración 17. Vista frontal escenario de ventilación 3 (Puerta abierta y ventanas cerradas).

Mediante la simulación como se muestra en las Ilustración 6,11,15 el aire caliente también ingresa por la puerta de entrada, pero no abarca una gran área como el que se evidencia en las ventanas y se observa una acumulación de aire frío en la esquina superior trasera, pero con una circulación de aire baja.

### 3.1.1.1 Parámetros meteorológicos

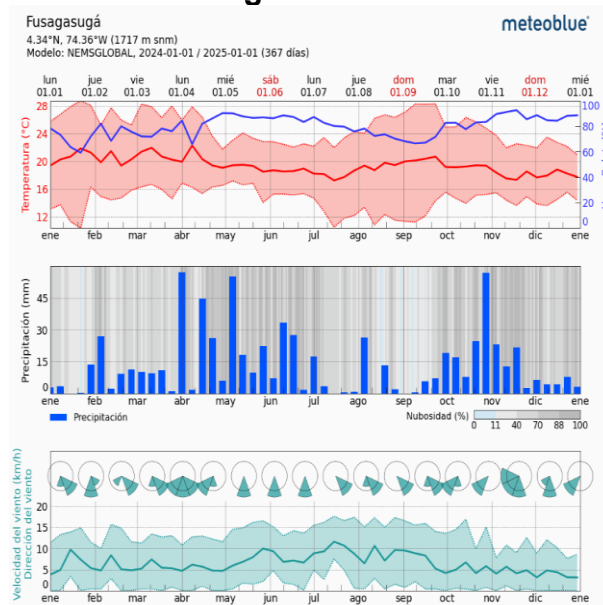


Ilustración 18. Resultados históricos del clima en Fusagasugá año 2024. Fuente: [30]

Para estas simulaciones, se consideró una temperatura ambiente variable basada en los registros meteorológicos del municipio de Fusagasugá, los cuales oscilan entre los 18 °C y 22 °C, con velocidades del viento entre 3.8 m/s y 7.3 m/s [30]. Estos valores fueron empleados como condiciones en la simulación de los distintos escenarios. Tal como se observa en las ilustraciones correspondientes, el color rojo representa las zonas con mayor temperatura y velocidad del flujo de aire dentro del

rango establecido, mientras que el azul oscuro indica las regiones con menor circulación y temperaturas más bajas. Estos parámetros son fundamentales para la adecuada representación del comportamiento del aire en los escenarios simulados.

### 3.1.2. Análisis de resultados de simulación.

Mediante la simulación de los diversos escenarios, se determinó que la ubicación óptima para la instalación del sistema IoT de medición es la zona central superior del aula, debido a que en esta área se presenta un mayor flujo de partículas de aire. Esta ubicación permite la recolección precisa de las variables necesarias para el análisis de la calidad del aire y la evaluación del confort térmico en el entorno.

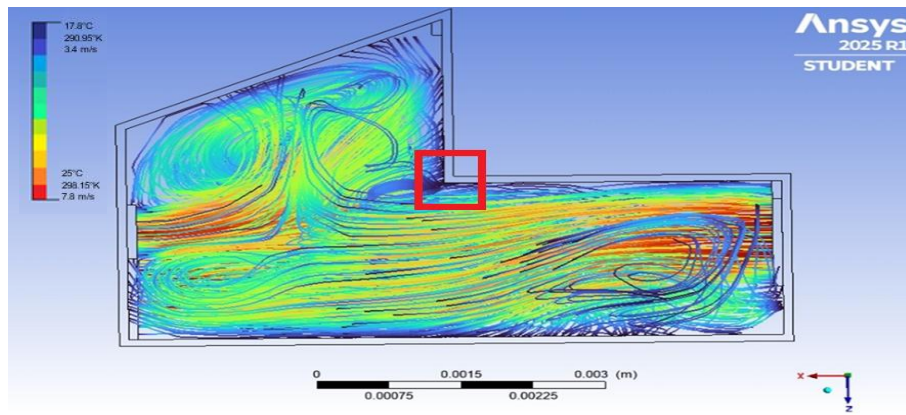


Ilustración 19. Ubicación del sistema IoT. Fuente: elaboración propia.

### 3.1.3. Caracterización de un servicio de ventilación inteligente.

El servicio de ventilación inteligente para aulas académicas tiene por objetivo monitorear y analizar tres variables mediante la prueba de concepto. Las variables seleccionadas son: temperatura y humedad para analizar confort térmico y concentración de CO2 para analizar calidad de aire interna.

- **Objetivo del servicio:** Analizar el confort térmico y la calidad de aire del aula de prueba de concepto por medio de una arquitectura en nube que cumpla con los requerimientos del caso de uso: recepción, almacenamiento, procesamiento y visualización de los datos.

#### 3.1.2.1 Variables del sistema

En esta sección se encuentra la descripción y rangos de las variables del servicio de ventilación inteligente para su respectivo monitoreo y análisis:

**Temperatura:** Esta variable influye en el nivel de ventilación que tenga un espacio cerrado y el uso de aire acondicionado, como se puede observar en la Ilustración

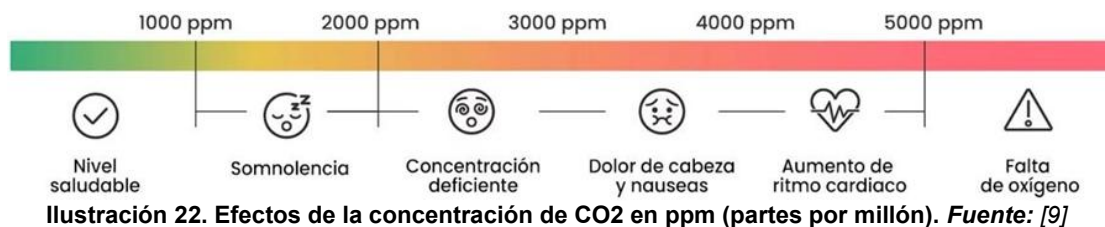
20 se definen los rangos de temperatura y humedad ideales dentro de un aula o habitación [9].



**Humedad relativa:** Niveles ideales de porcentaje de humedad en el ambiente cerrado con el fin de evitar al máximo la proliferación de bacterias y enfermedades respiratorias. Con base a la guía de ventilación planteada de un estudio entre las Universidades de Harvard y UCLA, la humedad interior debe estar en el 40%. También observaron que bajar la humedad hasta el 20% o subirla al 60% afecta la experiencia, con síntomas como dificultad para respirar y dolor en el pecho. En la Ilustración 21 se pueden evidenciar los efectos de los rangos de la humedad relativa [9].



**Concentración de CO<sub>2</sub>:** Límite de niveles seguros de presencia de CO<sub>2</sub> en un espacio cerrado que no sea perjudicial para el bienestar. Esta variable es clave en los sistemas de ventilación para identificar el momento en que se necesita una renovación del aire interno. Los umbrales de CO<sub>2</sub> se deben tener en cuenta ya que estos determinan no solo el rendimiento de los ocupantes, sino que también compromete su salud. Con base a la guía mencionada, un salón de clases típico (de 25 estudiantes y 100 metros cuadrados) debe tener una concentración de entre 400 a 1000 ppm. Al superar el umbral los ocupantes presentan somnolencia y falta de concentración. En niveles más altos pueden presentar aumento del ritmo cardiaco, náuseas y dificultad para respirar, como se observa en la Ilustración 22.



### 3.1.2.2 Intervalos de medición de parámetros (Temperatura, humedad y CO2)

En esta sección se tendrán en cuenta los intervalos de trabajo para la medición de variables de los sensores seleccionados para el sistema IoT, para más adelante evaluar el rendimiento de los datos recibidos con los estimados.

#### Temperatura y humedad

**Sensor SHTC3:** utiliza un principio capacitivo para medir humedad y un termistor de silicio para medir la temperatura. Funciona internamente con un convertidor ADC de alta resolución, que digitaliza las señales analógicas de los sensores.

La humedad se mide a través de un material dieléctrico cuya capacitancia varía en función de la cantidad de agua absorbida del aire, lo que se traduce en un valor digital correspondiente a la humedad relativa.

Por otro lado, la temperatura se determina mediante un termistor de silicio cuya resistencia eléctrica cambia en respuesta a las variaciones térmicas. Esta variación de resistencia es detectada y convertida en un valor digital que representa la temperatura.

Este sensor realiza una medición de las variables de temperatura y humedad en un rango de medición de 10.8ms en un modo de uso normal, se estima que para la correcta medición de estas dos variables se debe tomar muestras en un intervalo aproximado de 30s a 5 minutos, lo que permitiría un aproximado de 300-200 datos medidos en 5 minutos [31].

#### CO2

**Sensor MQ -135 de Winsen Electronics:** Emplea el principio de conductividad de semiconductores detecta gases como CO<sub>2</sub>, amoníaco, benceno, alcohol y humo mediante un material semiconductor sensible, generalmente óxido de estaño (SnO<sub>2</sub>), cuya conductividad eléctrica varía en presencia de ciertos gases. Cuando el gas entra en contacto con el sensor, reacciona químicamente en la superficie del material, alterando su resistencia. Este cambio es medido por un circuito interno y convertido en una señal analógica proporcional a la concentración del gas detectado.

Este sensor realiza una medición mínima cada 20-30s en un rango de 0-1000 ppm de partículas de CO<sub>2</sub> disueltas en el aire, se estima que la correcta medición de esta variable se deben tomar muestras en un intervalo mínimo de 5-10 minutos, lo cual permite un aproximado 10-15 de datos dependiendo del intervalo de medición [32].

Esta información se recopiló de los datasheet de los sensores.

### 3.1.3 Caracterización de hardware.

Esta sección contiene la respectiva caracterización de los elementos empleados en el sistema IoT. Comprende los sensores y la placa de desarrollo seleccionada.

Tabla 5. Caracterización de hardware empleado. Fuente: elaboración propia.

Sensor/A DC	Tipo de Dispositivo	Variables que mide	Rango de operación	Alimentación	Salida	Protocolo/Interfaz
<b>SENSE HAT B (SHTC3)</b>	Sensor digital de temperatura y humedad	Temperatura, Humedad relativa	Temp: -40°C a +125°C, Humedad: 0–100% RH	1.62V a 3.6V	Digital (I <sup>2</sup> C)	I <sup>2</sup> C
<b>MQ135</b>	Sensor analógico de gas	Calidad del aire (CO <sub>2</sub> , NH <sub>3</sub> , alcohol, humo, etc.)	Sensible en ppm (partes por millón), varía por gas	5V	Analógica	N/A (requiere ADC)
<b>MCP3008</b>	Convertor analógico a digital (ADC)	No mide directamente, convierte señales analógicas en digitales	0–Vref (Vref suele ser 3.3V o 5V)	2.7V a 5.5V	Digital (SPI)	SPI
<b>Raspberry Pi 4 Model B</b>	Microcomputadora de propósito general	controla y procesa datos de sensores	0–50°C (temperatura de operación)	5V (USB-C)	Digital	GPIO, I <sup>2</sup> C, SPI, UART

## 3.2 DISEÑO

En esta sección se encuentra el diseño del escenario IoT y la arquitectura en nube planteada con los servicios de AWS.

### 3.2.1 Escenario IoT

A continuación, está la representación del sistema IoT y el flujo de la información en el sistema.

La Ilustración 23 representa los actores del sistema y explica el flujo de la información.

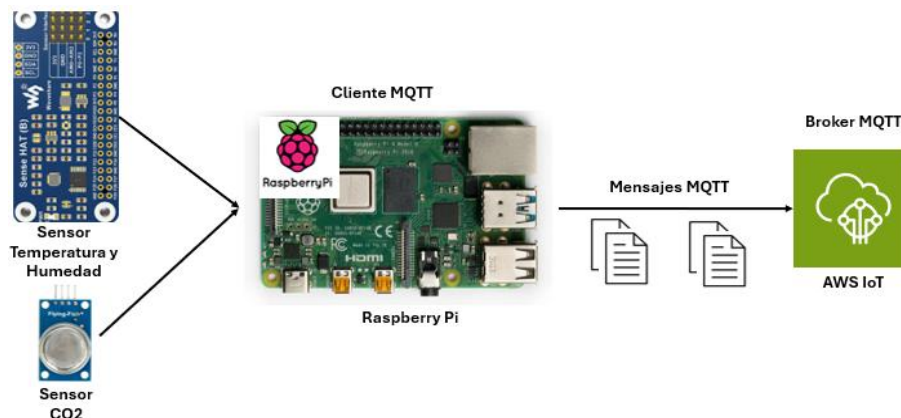


Ilustración 23. Diagrama de actores del sistema y el flujo de información. Fuente: elaboración propia.

- Sensores para las variables: temperatura, humedad y concentración de CO2.
- Una computadora de una sola placa (Raspberry Pi 4 modelo B), encargada de recolectar los datos de los sensores, empaquetarlos en forma de mensaje IoT y compartirlos por medio del protocolo de suscripción/publicación de MQTT.
- El bróker MQTT, está dentro de la nube de Amazon y se encarga de descifrar los mensajes publicados a un Topic MQTT.

### 3.2.2 Amazon Web Services (AWS)

En este apartado se tiene una breve descripción de cada servicio de AWS seleccionado para la arquitectura en nube de acuerdo con el caso de uso.

**AWS IoT Core:** Servicio para conectar y comunicar la nube de AWS con dispositivos IoT por medio de los protocolos de comunicación: MQTT, HTTP y WebSocket. Los dispositivos como Raspberry Pi pueden enviar datos a AWS IoT Core por medio de un protocolo ligero, confiable y seguro.

**AWS Lambda:** Este servicio permite realizar funciones en nube sin contar con servidores físicos. Dentro de las funciones que se pueden definir están: activación automática en respuesta a un evento, recepción de datos de sensores, análisis de datos de sensores.

**Amazon Simple Storage Service (S3):** Amazon Simple Storage Service es un servicio de almacenamiento por objetos para guardar archivos y datos de diferentes tipos. Se usa para almacenar datos históricos, archivos de registro y archivos generados tras análisis de datos.

**DynamoDB:** Base de datos NoSQL que almacena de forma rápida, escalable y se ajusta automáticamente para los requerimientos de mayor capacidad.

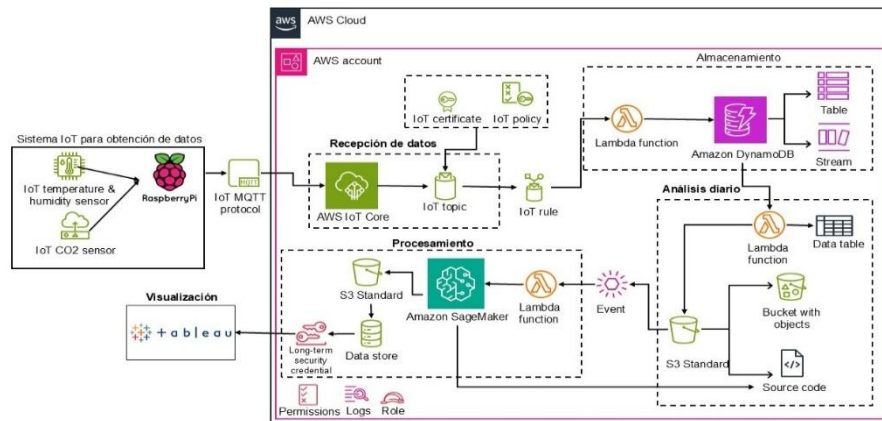
**Amazon EventBridge:** Servicio sin servidor para crear eventos con otros servicios o aplicaciones.

**Amazon CloudWatch:** Servicio que supervisa el funcionamiento de otros servicios dentro de AWS, notifica cambios, rendimiento e información del estado operativo. También permite visualizar gráficamente el rendimiento de las aplicaciones, configurar alarmas y reaccionar automáticamente ante cambios.

**Amazon SageMaker:** Servicio para el análisis de datos e Inteligencia Artificial (AI, por sus siglas en inglés). Usa Aprendizaje automático (ML, por sus siglas en inglés). Permite análisis, crear modelos, usar la IA generativa, trabajos de procesamiento y muchas funcionalidades para optimizar procesos autónomos y gobernanza.

### 3.2.3 Diseño de arquitectura en nube

A continuación, se expone el diseño de arquitectura en nube para el caso de uso que requiere recepción, almacenamiento, análisis y procesamiento de los datos por medio de los servicios de AWS previamente descritos.



**Ilustración 24. Propuesta de arquitectura. Fuente: elaboración propia.**

**Obtención de datos:** Por medio del sistema IoT conformado por la placa Raspberry Pi y sensores de temperatura, humedad y CO2.

**Envío de los datos a la nube:** La Raspberry Pi es el cliente MQTT que se encarga de empaquetar los datos de los sensores en formato de mensaje IoT y por medio del protocolo MQTT los envía al Bróker en AWS.

**Recepción de los datos:** Dentro de AWS IoT Core se crea un Topic IoT que contiene una política y certificados para establecer conexión con Raspberry Pi. De este modo se reciben los datos del sistema IoT y se publica en el Topic IoT.

**Almacenamiento de los datos:** Se crea una regla IoT para enrutar los datos del servicio AWS IoT Core a una función Lambda que se encarga de organizar los datos por columnas en una tabla de DynamoDB donde se almacenan.

**Análisis de los datos:** Se utiliza una función Lambda que se encarga de invocar automáticamente los datos almacenados en DynamoDB y por medio de código realiza un análisis previo de los datos en crudo. Toma dichos datos y los organiza por orden cronológico teniendo en cuenta el timestamp conformado por fecha y hora (ejemplo: 21-04-2025 12:44:20) y los pasa a un Bucket S3 en Amazon Simple Storage Service (S3) para almacenarlos por carpetas diarias como archivos de tipo JSON.

**Procesamiento de los datos:** En otro Bucket S3 se guarda un archivo.py que contiene el código para activar en el servicio SageMaker Studio un trabajo de procesamiento que se encarga de tomar las carpetas diarias, analizar calidad de aire, confort térmico y sacar estadísticas de las tres variables ambientales (temperatura, humedad y CO2). Para esto, se crea un evento con la herramienta Amazon EventBridge que active a medianoche con cinco minutos (00:05 am) una función Lambda encargada de disparar el trabajo de procesamiento en SageMaker. Cuando Lambda activa SageMaker, este invoca el archivo.py almacenado en el

Bucket S3 y lo corre para procesar los datos diariamente. Al terminar el trabajo de procesamiento almacena un archivo diario tipo JSON de resultados en un nuevo Bucket S3. Este proceso se repite automáticamente todos los días a la misma hora gracia a el evento y la función Lambda.

**Visualización de los datos:** Para la visualización, se utilizó la herramienta Tableau, que es externa a la nube de Amazon. Para eso se debe crear un usuario con credenciales que permitan exportar automáticamente los datos del S3 de resultados por medio de un script de Python y los almacene localmente en el computador. De esta forma los archivos con los resultados del procesamiento en nube son almacenados automáticamente a medianoche y se introducen manualmente en Tableau para visualizar los resultados por medio de métricas.

### 3.2.4 Seguridad dentro de AWS

AWS maneja la seguridad de forma granular, es decir que se puede configurar detalladamente los servicios y accesos que un usuario o dispositivo pueden tener.

- **Política (Policy):** Determina los permisos sobre un recurso, llamado "Resource based Policy".
- **IAM Role (IAM rol):** Entidad con permisos específicos.
- **Certificate (Certificado):** Se trata de un certificado X509 para autenticar dispositivos y conexiones.
- **IAM User (Usuario):** Entidad que representa una persona o aplicación en AWS que consiste en ingresar credenciales como nombre y contraseña.

**Autenticación mutua:** También llamada autenticación bidireccional, es cuando dos partes de un canal de comunicación se verifican la identidad mutuamente. Un ejemplo es cuando en una aplicación de Uber, el cliente verifica placa y credenciales del conductor, y este a su vez verifica la identidad de su pasajero. Del mismo modo en las comunicaciones digitales como cliente y un servidor o dispositivos IoT interconectados, con esta autenticación verifican de forma independiente la identidad del otro. Esta autenticación usualmente la asocian con el protocolo Transport Layer Security (TLS, por sus siglas en inglés), pero también es usada con otros protocolos [17].

A continuación, están los tres métodos principales de autenticación mutua los extremos en un canal de comunicación (emisor/receptor):

1. Autenticación de clave pública: Método basado en la encriptación de clave pública donde se codifica los datos mediante una clave. Esta encriptación utiliza dos claves: la clave pública y la privada, que se desencriptan la una con la otra.

En la autenticación mutua de clave de ambas partes anuncian una clave pública y tienen que demostrar que poseen la clave privada. Cada parte envía un fragmento

de datos llamado “firma” que encripta con su clave privada, la otra parte de la comunicación desencripta la firma usando su clave pública.

2. Autenticación por certificado: Es similar a la autenticación de clave pública, en estas ambas partes de la comunicación poseen un certificado de clave pública. El certificado contiene información adicional para verificar la identidad de ambas partes, esto al incluir quien emite el certificado junto con la clave pública a quien se le aplica dicho certificado.
3. Nombre de usuario y contraseña: Este método de autenticación mutua sigue usando un certificado en el servidor para que se lo presente al cliente, con el fin de que este ingrese el usuario y contraseña para que sean verificadas las credenciales.

En el Internet de las cosas se necesitan conectar los dispositivos a través de una red no segura que es el Internet. La autenticación mutua ayuda a garantizar que los datos recibidos sean precisos y de una fuente legítima [17].

### **3.2.5 Protocolo compatible con la autenticación mutua:**

- **TLS:** El TLS mutuo es un tipo de autenticación mutua donde ambos extremos tienen un certificado. Este se usa en la seguridad de las API, seguridad del IoT y aplicaciones de seguridad de Zero Trust.

## **3.3 IMPLEMENTACIÓN**

En esta sección se encuentra la configuración de una herramienta llamada Node-RED para comunicar el sistema IoT con AWS. También las configuraciones de los servicios dentro de AWS para conformar la arquitectura en nube, la instalación del sistema IoT en el aula de prueba de concepto y la comunicación del sistema IoT con la nube de Amazon.

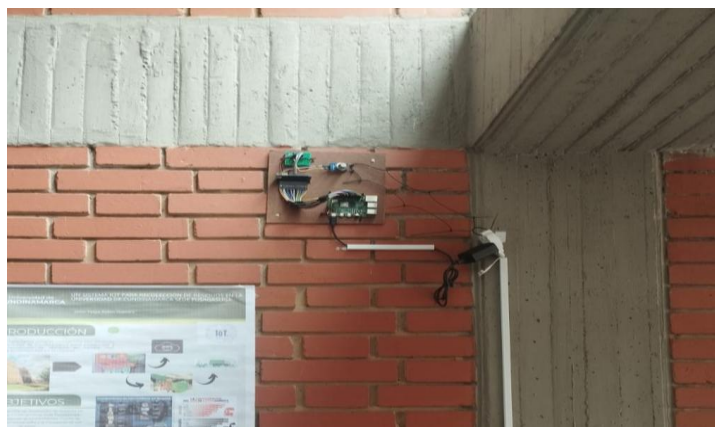
### **3.3.1 Instalación del sistema IoT**

La instalación del sistema IoT encargado de monitorear las variables ambientales fue teniendo en cuenta la caracterización de la ventilación que presenta el aula de prueba de concepto, con la cual se identificó en qué zona el monitoreo podría medir cualquier cambio en el ambiente interno.



**Ilustración 25. Montaje hardware IoT en la sala de prueba de concepto. Fuente: elaboración propia.**

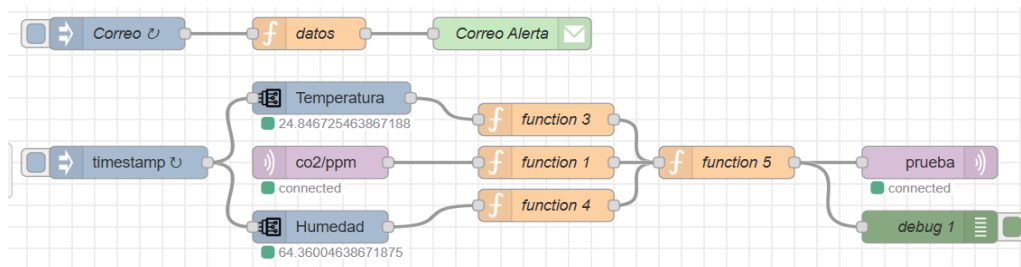
La recolección de los datos se realizó mediante el uso de dos sensores: un sensor de CO2 MQ-135, el cual mide la concentración de CO2 en partículas por millón (ppm) dentro de un espacio determinado, y un sensor de temperatura y humedad SHT3C integrado en una placa SENSE HAT B de Waveshare. Todo esto conectado a una placa Raspberry Pi 4 modelo B de 2GB de RAM, como se puede observar en la Ilustración 25. Dentro del entorno de la Raspberry se realizó la configuración respectiva de los sensores por medio de Python, y posteriormente se integró una herramienta que mediante módulos denominados Nodos, permite la conexión de diferentes sensores y aplicativos en entornos de Raspberry Pi.



**Ilustración 26. Montaje hardware IoT. Fuente: elaboración propia.**

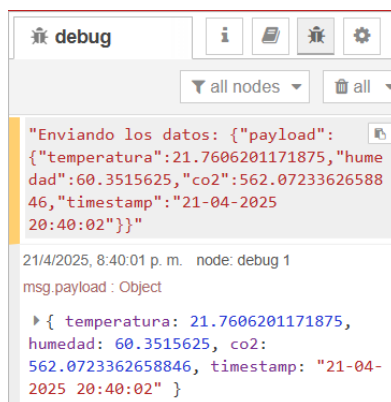
### **3.3.2 Configuración de Node – Red**

La herramienta Node-RED permite la configuración rápida y efectiva de diferentes sensores que se conectan a la Raspberry Pi por medio de nodos preconfigurados que ya existen dentro del entorno. Se deben descargar los paquetes en la Raspberry Pi y se accede en el buscador por medio de la IP que la red le asigna a la placa y podemos acceder mediante el puerto 1880 para allí realizar la configuración de los nodos requeridos, como se observa en la Ilustración 27.



**Ilustración 27. Configuración de módulos en la herramienta Node-RED. Fuente: elaboración propia.**

Este flujo permite la recolección de datos provenientes de sensores de temperatura, humedad y CO2, los cuales son procesados mediante funciones programadas en JavaScript o Node.js (nodo *function*). Posteriormente, los datos se unifican en un único mensaje (en la función 5) y se envían a un *broker* MQTT de prueba. Adicionalmente, se configuró una salida de depuración (*debug 1*) para visualizar los datos en consola, lo cual facilita la verificación del envío correcto de los mensajes a través del protocolo MQTT, como se muestra en la Ilustración 28. Esta funcionalidad permite identificar posibles fallos o anomalías en el sistema y activar alertas por correo electrónico en caso de que no se estén transmitiendo los datos como se espera.



**Ilustración 28. Debug de Node-RED. Fuente: elaboración propia**

Asimismo, se configuró un sistema de alerta por correo electrónico con el propósito de notificar si se están recibiendo correctamente los datos o si alguno de ellos falta. En caso de que no se obtengan los tres datos requeridos (temperatura, humedad y CO2), el mensaje combinado no se envía al *broker* MQTT, y se genera una notificación para alertar sobre la interrupción en el flujo de datos, o el correcto envío de los datos como se observa en la Ilustración 29 e Ilustración 30.

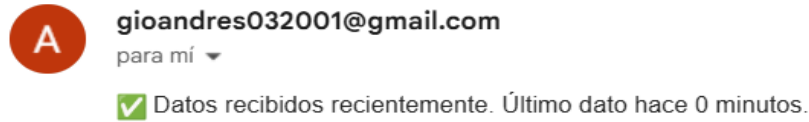


Ilustración 29. Notificación de alerta mediante correo electrónico.



Ilustración 30. Notificación de alerta mediante correo electrónico (Alerta de datos faltantes). Fuente: elaboración propia

El sensor de CO2 se configuro mediante un script empleando el lenguaje Python para crear un bróker local de MQTT y enviar los datos a la interfaz de Node-RED, como se puede observar en la Ilustración 31.

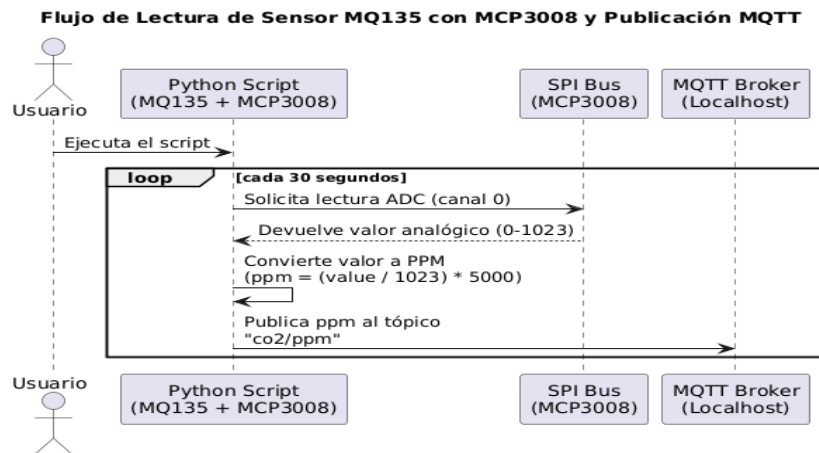


Ilustración 31. Diagrama de secuencia funcionamiento de sensor MQ-135. Fuente: elaboración propia

*Nota: debido a que la raspberry no cuenta con puertos análogos se empleó un conversor análogo digital MCP3008 conectado a un puerto digital mediante el protocolo SPI de la raspberry ya que el sensor mq-135 es un sensor análogo, mientras que el SHT3C es un sensor digital que trabaja mediante I2C.*

Dentro de las funciones implementadas, se configura la asignación de un identificador a cada dato utilizando la propiedad `msg.topic`. Esto resulta útil cuando se combinan múltiples flujos, porque permite distinguir a qué variable corresponde cada mensaje (por ejemplo, `msg.topic = "temperatura"`, `"co2"` o `"humedad"`). Posteriormente, otra función se encarga de recopilar los datos de temperatura, humedad y CO2, identificándolos por su valor de `msg.topic` y almacenándolos temporalmente en la memoria de flujo (flow). Una vez se han recibido las tres variables, se genera un mensaje combinado que incluye una marca de tiempo

(timestamp) en el formato día-mes-año hora:minutos:segundos. Este mensaje se emite como salida para su posterior uso, siendo finalmente enviado al bróker MQTT en AWS. Este proceso se representa en la Ilustración 32 mediante un diagrama de flujo.

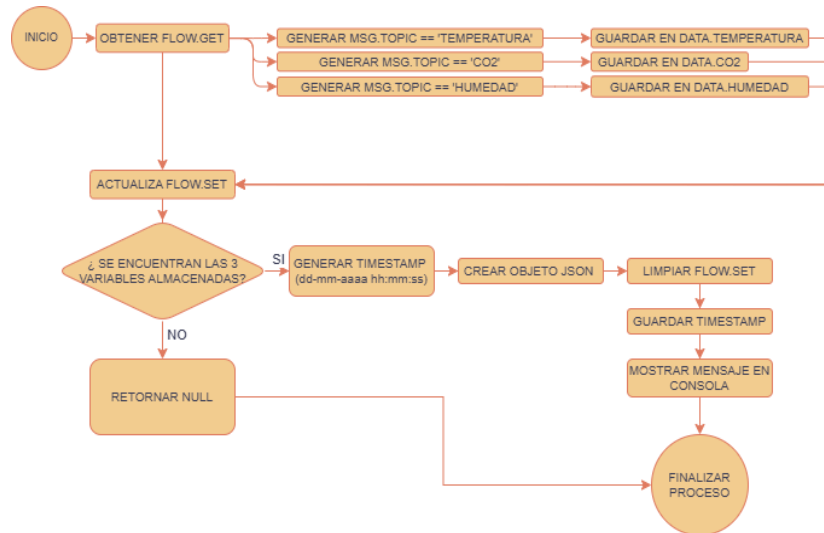


Ilustración 32. Diagrama de flujo recopilación de datos en Node-RED. Fuente: elaboración propia

### 3.3.2.1 Configuración remota

Para habilitar la configuración remota de la Raspberry Pi, se utilizó la herramienta Tailscale, la cual establece una red privada virtual (VPN) entre dispositivos mediante el protocolo WireGuard. Esto permite la interconexión de dispositivos como si pertenecieran a una misma red local, sin requerir la apertura de puertos ni configuraciones adicionales en el router, como se muestra en la Ilustración 33. De esta manera, es posible acceder de forma remota a la Raspberry Pi independientemente de la red a la que esté conectada, ya que, incluso tras un reinicio, el dispositivo se reconecta automáticamente a la dirección IP asignada, permitiendo su acceso remoto mediante protocolos como SSH o VNC.

MACHINE	ADDRESSES	VERSION	LAST SEEN
andres gloandres032001@gmail.com	100.84.191.13	1.82.0 Windows 11 24H2	Connected
maquinn gloandres032001@gmail.com	100.71.246.14	1.82.5 Windows 11 24H2	Connected
raspberrypi gloandres032001@gmail.com	100.110.132.86	1.82.0 Linux 6.12.20+rpt-rpi-v8	Connected

Ilustración 33. Creación de VPN mediante Tailscale. Fuente: elaboración propia

### 3.3.3 Configuración de los servicios dentro de AWS

A continuación, se encuentra la configuración de los servicios descritos y mencionados en el diseño propuesto de arquitectura en nube. Por consiguiente, se resume en tres pasos:

#### 3.3.4. Paso 1: Capa de recepción y almacenamiento de datos

Dentro de AWS IoT Core se realiza la creación de una política de IoT llamada “Acceso”, a la cual se le adjunta un certificado y al certificado se le asigna un dispositivo, Ilustración 34 y 35.

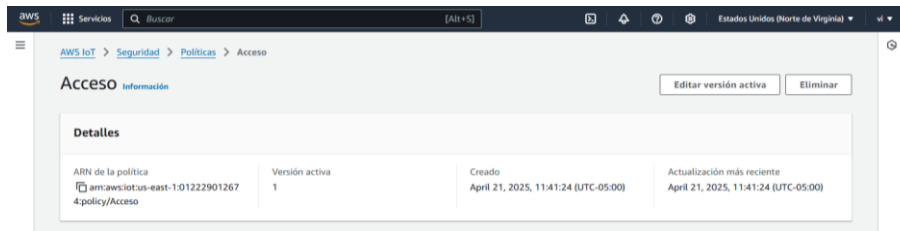


Ilustración 34. Creación de políticas AWS. Fuente: elaboración propia

Creación del certificado:

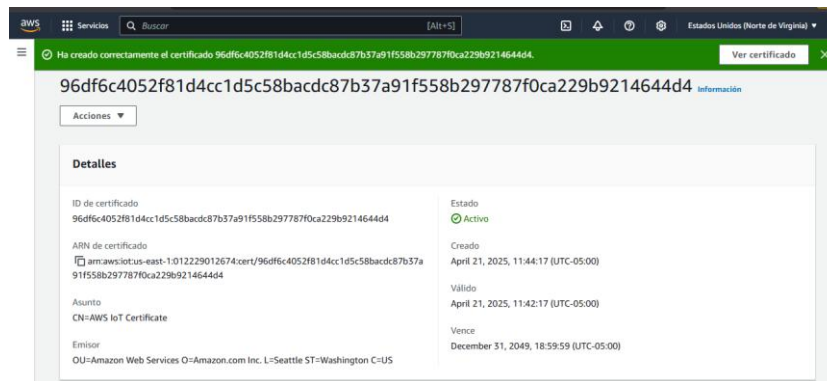


Ilustración 35. Creación de certificados AWS. Fuente: elaboración propia

Los archivos descargables del certificado creado se usan para enlazar Node – RED con AWS IoT Core y de esta forma la Raspberry Pi puede enviar los datos de los sensores a un Topic MQTT.

Se crea un objeto (Thing) correspondiente al cliente MQTT que es la placa Raspberry Pi del sistema IoT, bajo el nombre “Raspi”. A dicho objeto se le asigna el certificado de la Ilustración 35, que contiene la política de la Ilustración 34. El cliente MQTT (dispositivo Raspberry Pi), debe contener copia de los archivos del certificado para el correcto enlace a AWS. La correcta conexión, se evidencia por medio de la prueba de cliente MQTT de la Ilustración 36, donde dentro de AWS se debe suscribir al Topic MQTT creado (llamado “prueba”) y se observa que el mensaje llegue completo al bróker MQTT: datos de las 3 variables y la marca temporal “timestamp”.

```

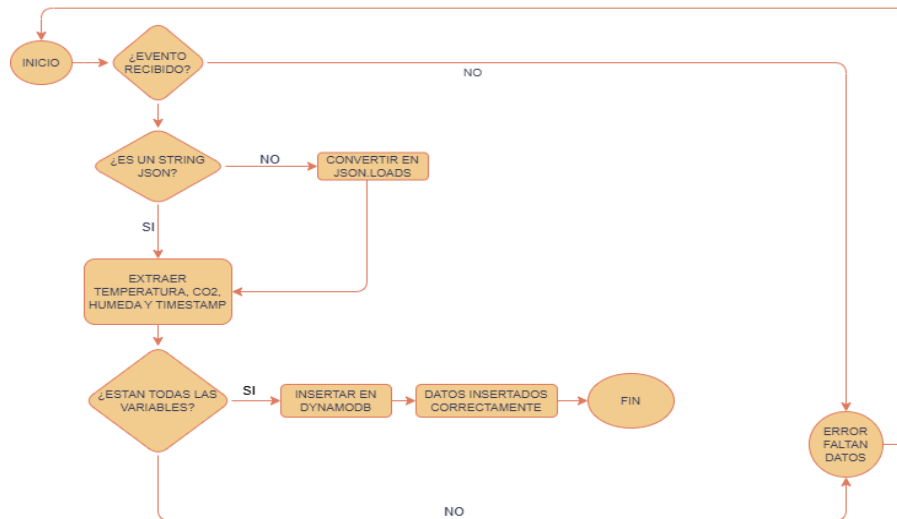
▼ prueba April 21, 2025, 12:20:31 (UTC-0500)
{
  "temperatura": 24.583114624023438,
  "humedad": 56.84814453125,
  "co2": 522.9716520039101,
  "timestamp": "21-04-2025 12:20:31"
}
► Propiedades

▼ prueba April 21, 2025, 12:20:01 (UTC-0500)
{
  "temperatura": 24.567092895507812,
  "humedad": 56.86187744140625,
  "co2": 527.8592375366569,
  "timestamp": "21-04-2025 12:20:01"
}

```

**Ilustración 36. Conexión del Topic para la recepción de datos a través de MQTT. Fuente: elaboración propia**

Para enrutar los datos a otros servicios de AWS, se crea una regla IoT llamada "enrutar\_datos", que va a pasarlos a una función Lambda llamada "enruta" de la Ilustración 37. Esta función los ordene por columnas y envía a la tabla en la base de almacenamiento DynamoDB.



**Ilustración 37. Diagrama de flujo de función lambda para enrutar. Fuente: elaboración propia**

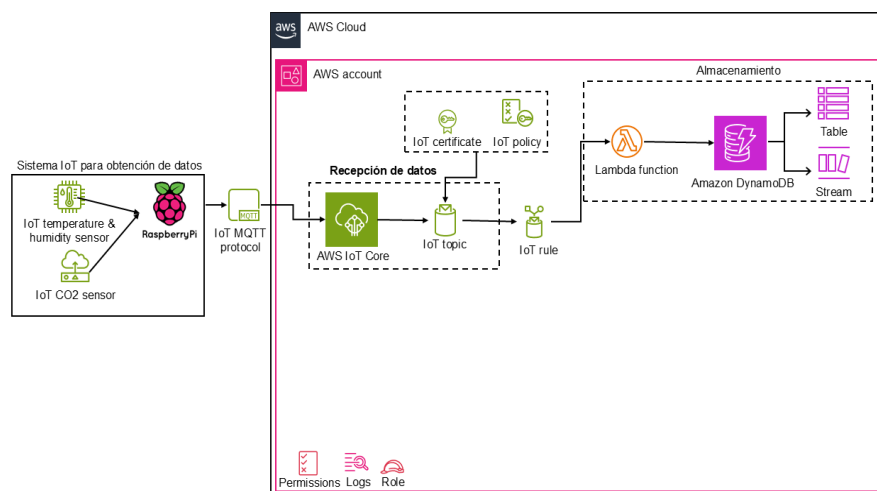
El diagrama de flujo de la Ilustración 37, resume la lógica del código de la función Lambda "enrutar", la cuál por medio de un trigger o desencadenador se enlaza a AWS IoT Core para que cada vez que llegue un dato nuevo la función lo detecte y el dato sea insertado en la tabla en DynamoDB Ilustración 38.

Tabla: Raspi\_datos: elementos devueltos (300)

El análisis inició el abril 23, 2025, 17:21:40

timestamp (Cadena)	co2	humedad	temperatura
<a href="#">23-04-2025 15:40:40</a>	542.52199...	55.804445...	26.1185302/343/3
<a href="#">22-04-2025 10:29:05</a>	527.85923...	68.331909...	21.02362060546875
<a href="#">23-04-2025 05:12:38</a>	508.30889...	67.109680...	18.526901245117188
<a href="#">21-04-2025 21:27:02</a>	552.29716...	60.647583...	21.394790649414062
<a href="#">22-04-2025 11:33:35</a>	537.63440...	66.636657...	21.9742431640625
<a href="#">21-04-2025 16:05:01</a>	513.19648...	57.124328...	22.64715576171875
<a href="#">22-04-2025 00:52:03</a>	508.30889...	63.766479...	20.2652587890625
<a href="#">22-04-2025 00:33:03</a>	508.30889...	63.607788...	20.390762329101562
<a href="#">21-04-2025 13:34:31</a>	522.97165...	55.352783...	26.0758056640625
<a href="#">21-04-2025 19:09:02</a>	566.95992...	61.750793...	22.43353271484375

**Ilustración 38. Comprobación de la transferencia de datos desde el IoT Core mediante la función Lambda a tabla en DynamoDB. Fuente: elaboración propia**

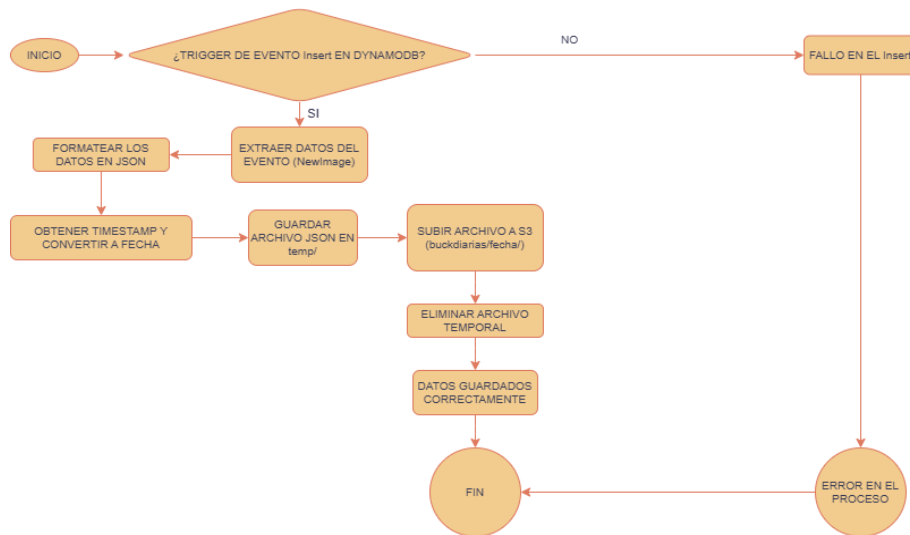


**Ilustración 39. Servicios de Capa de recepción y almacenamiento de datos del paso 1. Fuente: elaboración propia**

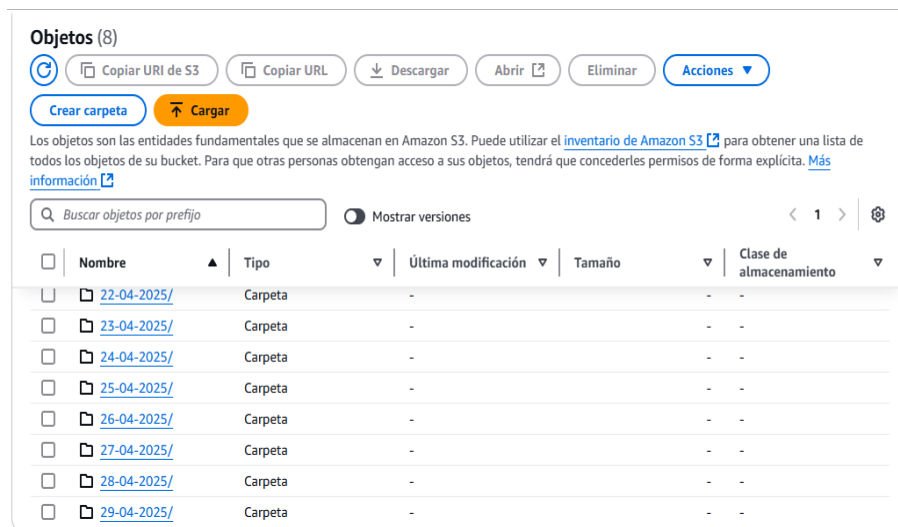
### 3.3.5. Paso 2: Capa de análisis

Luego con otra función Lambda, denotada como “diarias” Ilustración 40, se toman los datos de DynamoDB para clasificarlos y almacenarlos en carpetas de datos diarios en un Bucket S3 Ilustración 41, con el fin de procesarlos diariamente.

En la Ilustración 40, se encuentra el diagrama del funcionamiento del código de la función “diarias”. Esta función se encarga de tomar los datos que van llegando a la tabla de DynamoDB al agregarlo como un desencadenador y enrutarlos a un almacenamiento en un bucket S3 llamado “buckdiarias” de la Ilustración 41.

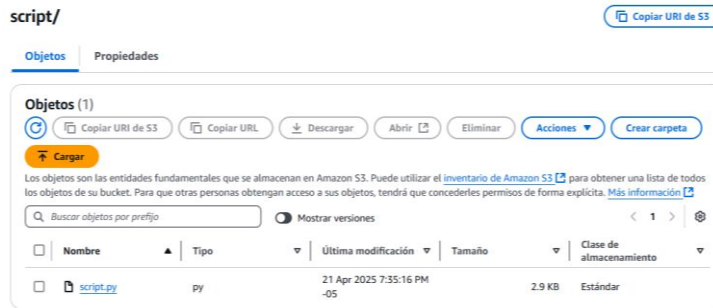


**Ilustración 40. Diagrama de flujo de función lambda para ordenar los datos a partir de DynamoDB. Fuente: elaboración propia**

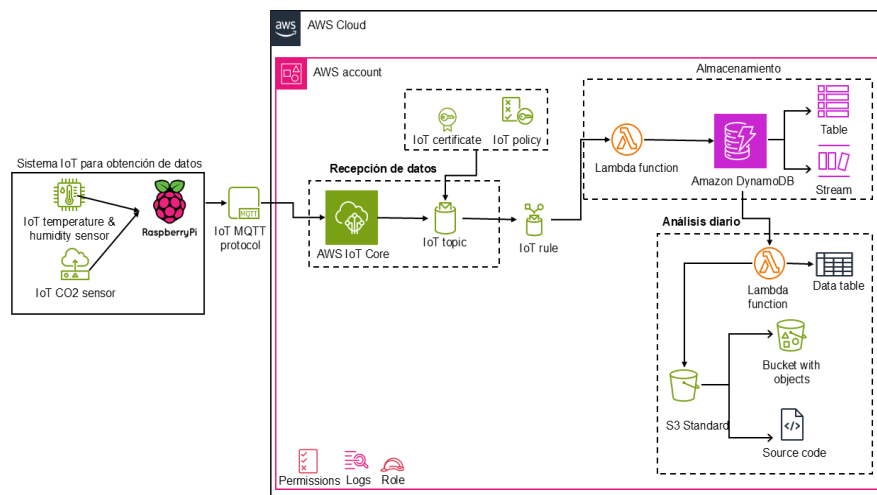


**Ilustración 41. Creación de Bucket S3 para almacenamiento de datos por días, a partir de la tabla de DynamoDB. Fuente: elaboración propia**

Para el procesamiento se creó otro Bucket S3 llamado “buckcodigos” que va a contener un archivo.py Ilustración 42, que activa un trabajo de procesamiento en el servicio de Amazon Sagemaker. Ese archivo.py lo invoca SageMaker Studio a través de una activación diaria con una función Lambda que tiene un evento programado por medio de EventBridge, para que se ejecute el procesamiento de los datos diarios a las 12:05 am.



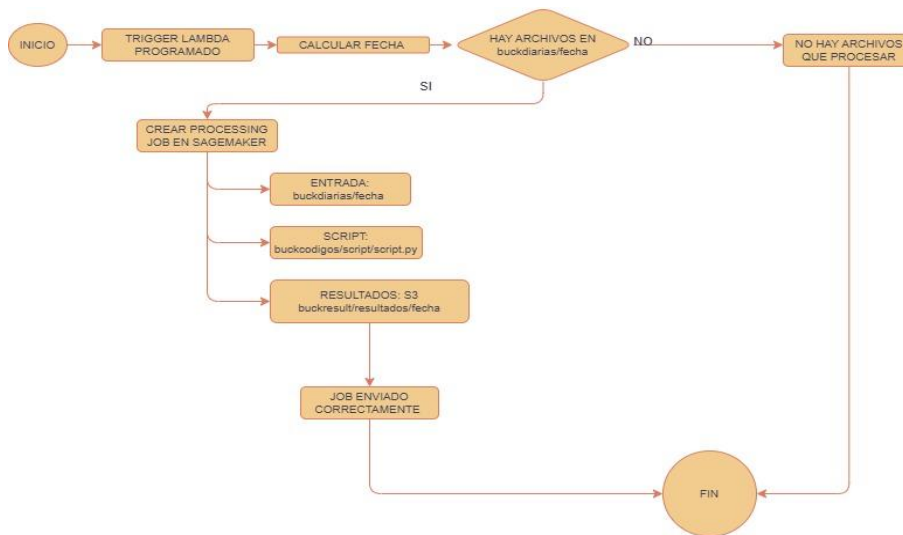
**Ilustración 42. Creación de archivo script.py dentro de Bucket S3 para el uso de Sagemaker para procesamiento de los datos obtenidos. Fuente: elaboración propia**



**Ilustración 43. Servicios Capa de análisis del paso 2. Fuente: elaboración propia**

### 3.3.6. Paso 3: Procesamiento de datos

Se crea una función Lambda llamada “procesa” Ilustración 44, encargada de lanzar el trabajo de procesamiento a Sagemaker.



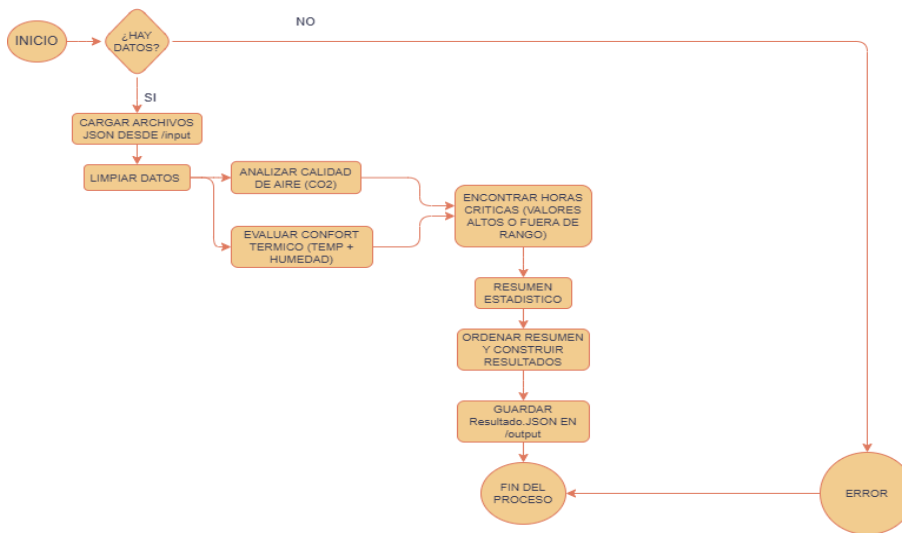
**Ilustración 44. Diagrama de flujo de la activación de Sagemaker con la función Lambda. Fuente: elaboración propia**

La Ilustración 44, representa el flujo del funcionamiento del código de la función Lambda “procesa”, que se activa por medio de un evento en Amazon EventBridge por el nombre de “activarLambda” Ilustración 45, que se ejecuta a las 12:05 am. Al activarse ejecuta un Trabajo de procesamiento dentro de Amazon Sagemaker Studio el cuál se lleva a cabo con el código de funcionamiento contenido en el archivo.py dentro del bucket S3 “buckcodigos” Ilustración 42. SageMaker Studio invoca desde S3 el archivo.py que contiene el código de funcionamiento para realizar análisis de calidad de aire, confort térmico y sacar estadísticas de las tres variables ambientales Ilustración 46.

<input type="checkbox"/>	Nombre de la programación	Grupo de programación	Estado	Destino	Tipo de destino	Última modificación
<input type="checkbox"/>	<a href="#">activarLambda</a>	<a href="#">default</a>	<span style="color: green;">✔</span> Habilitado	<a href="#">procesa</a>	LAMBDA_Invoke	Apr 22, 2025, 01:52:13 (UTC+00:00)

**Ilustración 45. Configuración de EventBridge para ejecución automática de función Lambda para activación de procesamiento en Sagemaker. Fuente: elaboración propia**

El diagrama de la Ilustración 46, especifica los pasos del código del archivo.py que Sagemaker invoca para rodar el trabajo de procesamiento. Este código se encarga de tomar las carpetas diariamente del día anterior del bucket S3 “buckdiarias” Ilustración 41, de esos datos diarios analiza calidad de aire, confort térmico y saca estadísticas de las 3 variables ambientales. Finalmente debe enviar los resultados y archivarlos diariamente en un nuevo bucket S3 llamado “buckresult” Ilustración 48, donde se encuentra un archivo con el resultado de trabajo de procesamiento Ilustración 47.



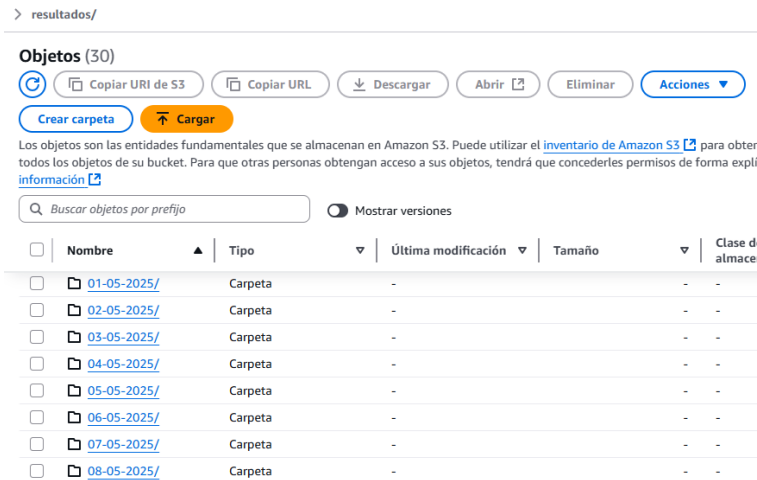
**Ilustración 46. Diagrama de flujo procesamiento mediante Sagemaker. Fuente: elaboración propia**

Trabajos de procesamiento

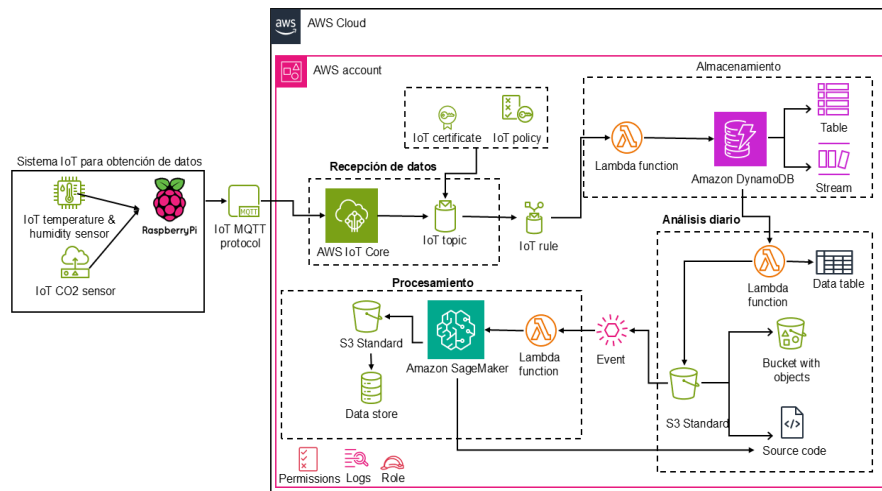
<input type="radio"/>	<a href="#">procesamiento-22-05-2025-00-09-01</a>	arn:aws:sagemaker:us-east-1:012229012674:processing-job/procesamiento-22-05-2025-00-09-01	23/5/2025, 0:09:03	2 minutes	Completed
<input type="radio"/>	<a href="#">procesamiento-21-05-2025-00-09-02</a>	arn:aws:sagemaker:us-east-1:012229012674:processing-job/procesamiento-21-05-2025-00-09-02	22/5/2025, 0:09:03	2 minutes	Completed
<input type="radio"/>	<a href="#">procesamiento-20-05-2025-00-09-02</a>	arn:aws:sagemaker:us-east-1:012229012674:processing-job/procesamiento-20-05-2025-00-09-02	21/5/2025, 0:09:03	2 minutes	Completed
<input type="radio"/>	<a href="#">procesamiento-19-05-2025-00-09-01</a>	arn:aws:sagemaker:us-east-1:012229012674:processing-job/procesamiento-19-05-2025-00-09-01	20/5/2025, 0:09:03	2 minutes	Completed
<input type="radio"/>	<a href="#">procesamiento-18-05-2025-00-09-01</a>	arn:aws:sagemaker:us-east-1:012229012674:processing-job/procesamiento-18-05-2025-00-09-01	19/5/2025, 0:09:03	2 minutes	Completed
<input type="radio"/>	<a href="#">procesamiento-17-05-2025-00-09-01</a>	arn:aws:sagemaker:us-east-1:012229012674:processing-job/procesamiento-17-05-2025-00-09-01	18/5/2025, 0:09:03	3 minutes	Completed
<input type="radio"/>	<a href="#">procesamiento-16-05-2025-00-09-01</a>	arn:aws:sagemaker:us-east-1:012229012674:processing-job/procesamiento-16-05-2025-00-09-01	17/5/2025, 0:09:03	2 minutes	Completed
<input type="radio"/>	<a href="#">procesamiento-15-05-2025-00-09-01</a>	arn:aws:sagemaker:us-east-1:012229012674:processing-job/procesamiento-15-05-2025-00-09-01	16/5/2025, 0:09:03	3 minutes	Completed
<input type="radio"/>	<a href="#">procesamiento-14-05-2025-00-09-01</a>	arn:aws:sagemaker:us-east-1:012229012674:processing-job/procesamiento-14-05-2025-00-09-01	15/5/2025, 0:09:03	3 minutes	Completed

**Ilustración 47. Verificación de procesamiento mediante Sagemaker. Fuente: elaboración propia**

Ahora para la parte de visualización, los resultados del trabajo de procesamiento de los datos diarios son almacenados automáticamente en un bucket S3 llamado “buckresults”, ahí también se almacenan por carpetas diarias Ilustración 48.



**Ilustración 48. Almacenamiento de resultados a partir de Sagemaker. Fuente: elaboración propia**



**Ilustración 49. Servicios de procesamiento de datos del paso 3. Fuente: elaboración propia**

### 3.3.7. Visualización de los datos

La visualización de los datos se llevó a cabo utilizando la herramienta Tableau, la cual facilita la representación ordenada y clara de archivos en formato JSON. Es importante que dichos archivos cuenten con un campo de marca temporal (timestamp) para asegurar su correcta organización y permitir la posterior creación de tablas o diagramas. Para ello, fue necesario extraer los datos procesados y almacenados en un Bucket de Amazon S3. Esta extracción se realizó mediante un script desarrollado en Python, el cual accede directamente al S3 "buckresult" utilizando credenciales generadas al crear un usuario dentro del entorno de AWS. Una vez obtenidos, los datos son almacenados localmente en el escritorio del equipo donde se ejecuta el programa.

*Nota: Cabe destacar que la aplicación de Tableau está disponible únicamente para sistemas operativos Windows y macOS, y no es compatible con Raspberry Pi OS. Esto se debe a que dicho sistema operativo está basado en la arquitectura ARM, mientras que Tableau requiere arquitecturas x86/x64. Además, Raspberry Pi OS no se encuentra entre los sistemas oficialmente soportados y el hardware de la Raspberry Pi no cumple con los requisitos mínimos de rendimiento que demanda Tableau.*

### 3.3.8. Extracción de datos del Bucket S3

Como se ha mencionado previamente, la extracción de los datos se realiza a través de un script desarrollado en Python, el cual permite almacenarlos de forma local en el equipo donde se ejecuta. Para llevar a cabo esta operación, es necesario contar con las credenciales de acceso generadas por AWS clave pública y clave privada al momento de crear un usuario y asignarle los permisos adecuados para la lectura de los datos en el Bucket de S3, y es necesario realizar la instalación de AWS CLI el cual es una herramienta que permite interactuar con los servicios de Amazon Web Services desde la línea de comandos, una de sus aplicaciones es la extracción de datos desde un Bucket de Amazon S3 mediante scripts. En la Ilustración 52 se presenta el funcionamiento general del programa.

```

Procesando carpeta del día: 21-04-2025
Procesando carpeta del día: 22-04-2025
Procesando carpeta del día: 23-04-2025
Procesando carpeta del día: 24-04-2025
Procesando carpeta del día: 25-04-2025
Procesando carpeta del día: 26-04-2025
Procesando carpeta del día: 27-04-2025
Procesando carpeta del día: 28-04-2025
Procesando carpeta del día: 29-04-2025
Procesando carpeta del día: 30-04-2025
Procesando carpeta del día: 01-05-2025
Procesando carpeta del día: 02-05-2025
Procesando carpeta del día: 03-05-2025
Procesando carpeta del día: 04-05-2025

```

**Ilustración 50. Resultados de script de extracción de datos de S3. Fuente: elaboración propia**

01-05-2025	22/05/2025 12:05 p.m.	Carpeta de archivos
02-05-2025	22/05/2025 12:05 p.m.	Carpeta de archivos
03-05-2025	22/05/2025 12:05 p.m.	Carpeta de archivos
04-05-2025	22/05/2025 12:05 p.m.	Carpeta de archivos
05-05-2025	22/05/2025 12:05 p.m.	Carpeta de archivos
06-05-2025	22/05/2025 12:05 p.m.	Carpeta de archivos
07-05-2025	22/05/2025 12:05 p.m.	Carpeta de archivos
08-05-2025	22/05/2025 12:05 p.m.	Carpeta de archivos
09-05-2025	22/05/2025 12:07 p.m.	Carpeta de archivos
12-05-2025	22/05/2025 12:07 p.m.	Carpeta de archivos
13-05-2025	22/05/2025 12:05 p.m.	Carpeta de archivos
14-05-2025	22/05/2025 12:05 p.m.	Carpeta de archivos

**Ilustración 51. Almacenamiento de los resultados en carpetas por día. Fuente: elaboración propia**

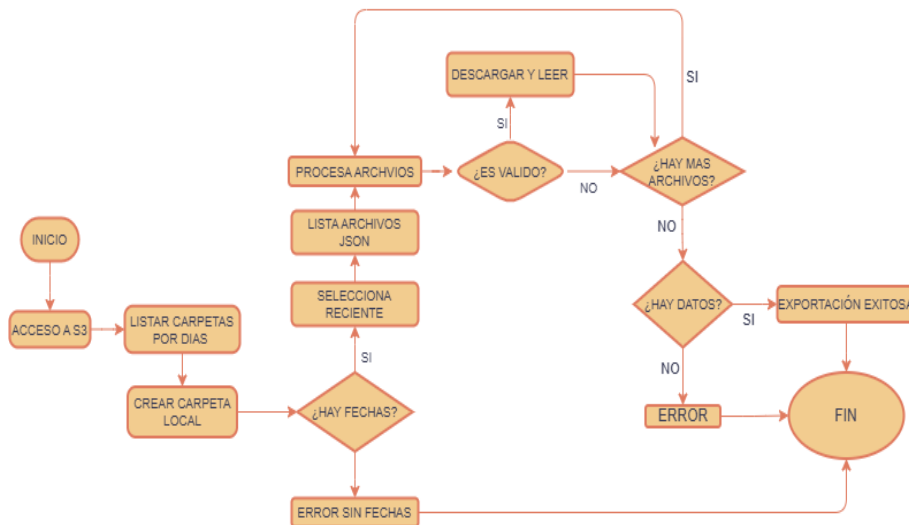


Ilustración 52. Diagrama de flujo script extracción de datos de S3. Fuente: elaboración propia

### 3.4 PRUEBAS DE FUNCIONAMIENTO Y/O DESEMPEÑO

En esta sección se tendrán en cuenta el funcionamiento de los servicios y sistemas planteados, para este caso será el sistema IoT de recolección de datos y el desempeño de la arquitectura.

#### 3.4.1 Obtención de datos de monitoreo IoT

El sistema IoT está conformado por la Raspberry Pi 4 modelo B y los sensores SHT3C (para temperatura y humedad) y MQ-135 (para concentración de CO2). La placa Raspberry Pi es el cliente MQTT en el escenario IoT Ilustración 23, en donde se instala Node – RED y se accede por el buscador a través de la IP asignada a la placa. Con Node – RED se establecieron los nodos de la placa, los sensores y funciones para configurar la forma de leer los datos, la marca temporal (timestamp), y el muestreo de los sensores, el cual se configuró para que el sistema IoT envíe las tres variables cada 5 minutos. En la Ilustración 36, se observa dentro de AWS que los datos llegan completos cada 5 minutos con su respectivo timestamp. Es decir que con el muestreo que tienen los sensores, el sistema IoT debe entregar diariamente 288 datos, siempre y cuando no se presenten fallas de la red eléctrica o de Wifi en la Universidad.

#### 3.4.2 Preparación de los datos para el procesamiento mediante Lambda

En la capa de análisis Ilustración 43, se consiguió un ordenamiento, filtrado y separación de los datos en crudo que llegan de la base de datos DynamoDB Ilustración 38. Estos datos llegan todos a DynamoDB y se almacenan sin tener en cuenta el orden de llegada, lo cual vuelve engorroso una revisión cronológica. Para

eso, se clasifican todos los datos invocados desde DynamoDB por medio del timestamp que especifica día, mes y año (ejemplo: 28-04-2025). Esto hace que se generen carpetas diarias bajo el título de esa fecha que corresponden al día Ilustración 41. Dentro de las carpetas se contiene los datos correspondientes a esa marca temporal y se evita que se sobrescriban al especificar el nombre de cada uno de esta forma: día, mes, año hora, minutos y segundos (ejemplo: 28-04-2025 09:34:20). Con esto la capa de análisis prepara previamente los datos de acuerdo con su marca temporal.

### **3.4.3 Activación del trabajo de procesamiento por medio de Lambda**

Implementar el servicio de Lambda permitió por medio de código Ilustración 44, invocar otros servicios de AWS y llamar tablas de datos de un S3 o DynamoDB. De este modo se estableció la programación y funcionamiento del trabajo de procesamiento en el servicio de SageMaker Studio. Para lograrlo la función Lambda llamada “procesa”, se encargó de invocar los datos del bucket S3 de las carpetas diarias Ilustración 41, establecer el uso de la zona horaria de Bogotá e indicar la ruta que debía seguir SageMaker para invocar el archivo.py que contiene el código de procesamiento y los datos de las carpetas diarias. De esta forma se consiguió un procesamiento de datos automático, en el que Lambda se encarga de orquestar el trabajo de procesamiento a la hora programada por medio del evento en Amazon EventBridge Ilustración 45.

### **3.4.4 Implementación de bucket S3**

El uso de los bucket S3 permitieron traspasar los datos por las etapas de almacenamiento, análisis y procesamiento de la arquitectura, debido a que es un servicio de almacenamiento de objetos que es compatible con todos los servicios de AWS. Además, facilitó el procesamiento de los datos al poder guardar el script del programa de procesamiento en un archivo.py de modo que se evita la configuración compleja dentro de SageMaker que genera costos, ya que para esto se debe crear un dominio, un cuaderno de trabajo e implementar servicios de paga como Amazon Glue y Amazon Athena. Con el manejo de S3 entre las etapas de la arquitectura se incurrió en costos bastante moderados, después de superar el límite de la capa gratuita de AWS, que permite usar máximo 20,000 solicitudes get y 2000 solicitudes put. El manejo de este servicio permitió un control de gastos de facturación de la arquitectura Ilustración 53.

Amazon Web Services, Inc. cargos por servicio [información](#) Expandir todo

Total de servicios activos: **9** Cargos totales por servicios antes de impuestos en USD: **0,05 USD**

Filtrar por nombre de servicio o región < 1 >

Descripción	Cantidad de uso	Importe en USD
SageMaker		0,03 USD
Simple Storage Service		0,02 USD
CloudWatch		0,00 USD
CloudWatch Events		0,00 USD
Data Transfer		0,00 USD
DynamoDB		0,00 USD

Ilustración 53. Administración y control de facturación dentro de AWS. Fuente: elaboración propia

### 3.4.5 Procesamiento de los datos ambientales

Con la arquitectura se consiguió el objetivo del servicio de ventilación planteado para el aula de prueba de concepto. Se obtuvieron análisis de confort térmico con la temperatura y la humedad, y a su vez se analizó la calidad de aire interno con las concentraciones de CO2 medidas. Adicional el trabajo de procesamiento también se encargó de sacar estadísticas a las 3 variables y señalar las horas con los umbrales de medidas más altas. Se puede apreciar en la Ilustración 54 donde se observa un archivo de resultado del trabajo de procesamiento que realizó SageMaker.

```

() resultado (2).json X
C:\Users\pacos > Downloads > () resultado (2).json > ...
 2  "resumen_estadisticas": {
 3    "co2": {
 4      }
 5    }
 6  },
 7  "total_registros": 288,
 8  "horas_criticas": [
 9    {
10      "co2": 527.8592375366569,
11      "humedad": 74.298095703125,
12      "temperatura": 18.465484619140625,
13      "timestamp": "2025-04-28 06:20:14",
14      "calidad_aire": "Buena",
15      "confort_termico": "Inconfortable"
16    },
17    {
18      "co2": 532.7468238694037,
19      "humedad": 71.221923828125,
20      "temperatura": 20.39877319335937,
21      "timestamp": "2025-04-27 23:25:13",
22      "calidad_aire": "Buena",
23      "confort_termico": "Inconfortable"
24    },
25    {
26      "co2": 557.1847507331378,
27      "humedad": 69.818115234375,
28      "temperatura": 21.62176513671875,
29      "timestamp": "2025-04-27 20:50:13",
30      "calidad_aire": "Buena",
31      "confort_termico": "Inconfortable"
32    }
33  ]
34  }
35  }
36  }
37  }
38  }
39  }
40  }
41  }
42  }
43  }
44  }
45  }
46  }
47  }
48  }
49  }
50  }
51  }
52  }
53  }
54  }
55  }
56  }
57  }
58  }
59  }
60  }

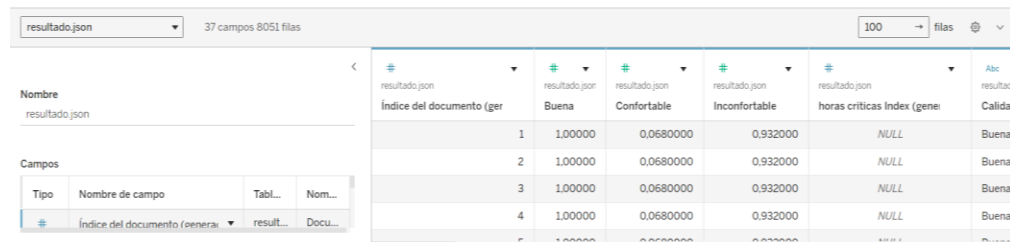
```

Ilustración 54. Resultados de archivos JSON de procesamiento de datos. Fuente: elaboración propia

### 3.4.6 Visualización mediante Tableau.

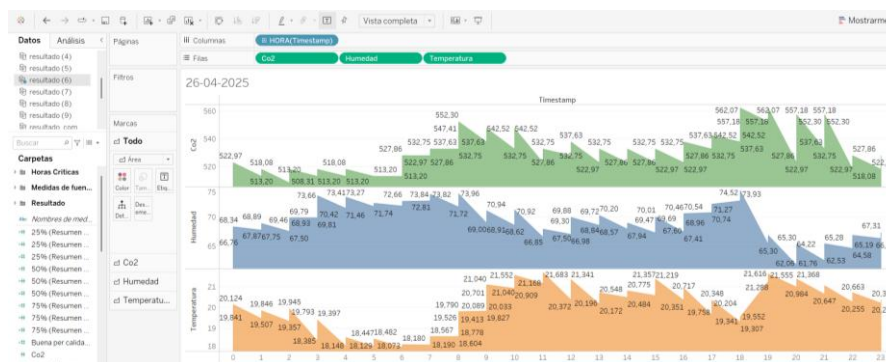
Para la visualización de los resultados generados por el procesamiento en Amazon SageMaker, se utilizó la herramienta Tableau Desktop, la cual permite representar gráficamente archivos en formato JSON, siempre que estos se encuentren

debidamente estructurados. Para ello, se procede a la descarga e instalación de la aplicación de escritorio de Tableau. Una vez instalada, se selecciona como fuente de datos la carpeta que contiene los archivos JSON generados diariamente a partir del monitoreo de las variables del sistema:



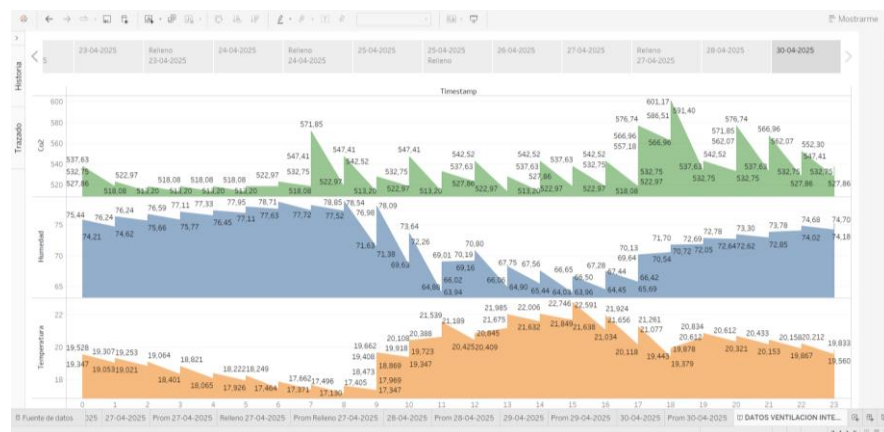
**Ilustración 55. Selección de fuente de datos dentro de la aplicación de Tableau. Fuente: elaboración propia**

Seguido de esto se realiza la creación de una hoja de trabajo, dentro de la cual se establece la gráfica a realizar a partir de los datos seleccionados:



**Ilustración 56. Selección de gráficas y datos relevantes dentro de Tableau. Fuente: elaboración propia**

Por último, se agregan las gráficas creadas a un historial donde se pueden comparar y observar el comportamiento de los datos de manera separa.



**Ilustración 57. Historial para visualización de las gráficas obtenidas. Fuente: elaboración propia**

Para garantizar una visualización adecuada y continua de los datos, se llevó a cabo un proceso de limpieza que incluyó la validación de los datos generados por la arquitectura, mediante su inspección visual Ilustración 60 y la comparación entre las horas registradas por día y las horas esperadas, así como con el total acumulado previsto. Adicionalmente, se implementó un script en Python que permitió completar automáticamente los registros faltantes en el análisis diario. Especialmente, cuando se detectan menos de 24 registros horarios en un día, se procedía a rellenar las horas ausentes utilizando un promedio basado en los datos de los cinco días anteriores. En los casos en los que esta información de registros posteriores no era suficiente para estimar el promedio, se realizaba una consulta al servicio Open-Meteo para obtener los datos climáticos faltantes de forma automática [33]. Esto se realizó con la finalidad asegurar la integridad y continuidad de la visualización de la información.

Se optó por utilizar el promedio en lugar de la mediana o la interpolación lineal debido a que ofrece una solución más simple, estable y menos propensa a errores como los valores NaN (datos faltantes o no válidos) o infinitos, los cuales no son compatibles con el formato JSON y al momento de insertar el archivo en Tableau no permite la visualización de los datos del archivo. Esta decisión se basó en los resultados obtenidos durante las pruebas, como se evidencia en la Ilustración 58. A diferencia de la mediana, que puede no ser representativa o incluso fallar cuando se dispone de pocos datos válidos, o de la interpolación lineal, que tiende a generar valores NaN en intervalos con escasa o mala distribución de datos, el uso del promedio permite suavizar las variaciones sin introducir valores problemáticos. Adicionalmente, funciones como `np.nanmean` permiten calcular promedios ignorando automáticamente los valores nulos, lo que reduce la necesidad de implementar validaciones adicionales y garantiza una ejecución más robusta, especialmente en conjuntos de datos dispersos o incompletos, como en este caso, donde se presentan ausencias de registros en ciertas horas debido a fallas en la red.

```
PS C:\Users\Laura> Files\WindowsApps\PythonSoftwareFoundation.Python.3.11_3.11.2544.0_x64__qbz5n2kfra8p0\Lib\json\encoder.py", line 240, in floatstr
>>     raise ValueError(
>> ValueError: Out of range float values are not JSON compliant: nan
```

**Ilustración 58. Uso de interpolación lineal para relleno de datos faltantes. Fuente: elaboración propia.**

Con los datos recopilados se realizó un análisis de pérdida de información comparando los datos efectivamente recibidos con los esperados por día. Este análisis también se contrastó con los registros generados por SageMaker en los archivos JSON de resultados diarios postprocesados.

presenta los datos esperados y los efectivamente recibidos en AWS, así como el conteo de horas con y sin recepción de información.

```

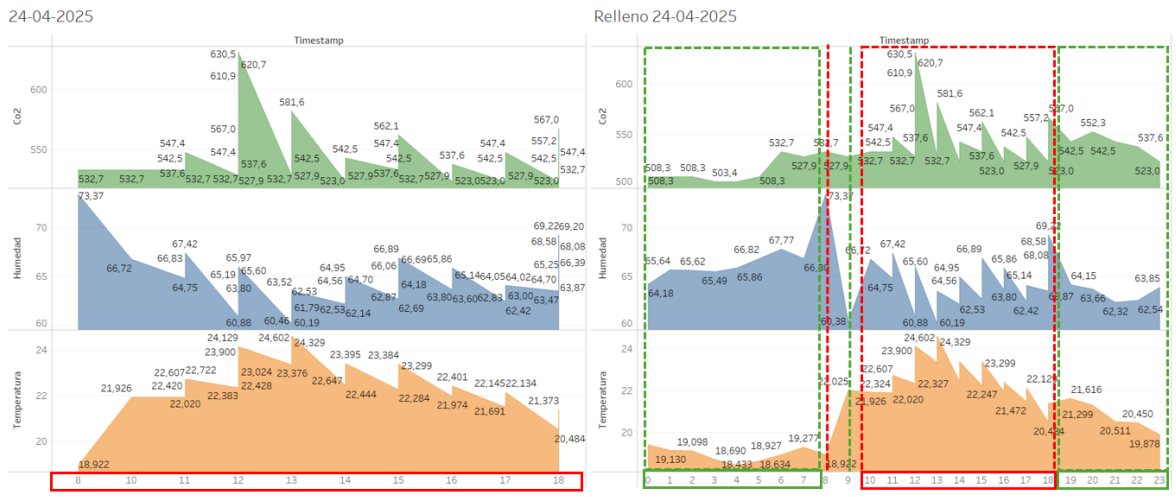
✓ Día 02-05-2025 ya tiene las 24 horas. No se necesita corrección.
✓ Día 03-05-2025 ya tiene las 24 horas. No se necesita corrección.
✓ Día 04-05-2025 ya tiene las 24 horas. No se necesita corrección.
✓ Día 05-05-2025 ya tiene las 24 horas. No se necesita corrección.
✓ Día 06-05-2025 ya tiene las 24 horas. No se necesita corrección.
✓ Día 07-05-2025 ya tiene las 24 horas. No se necesita corrección.
✓ Día 08-05-2025 ya tiene las 24 horas. No se necesita corrección.

📅 Día: 09-05-2025 - Horas faltantes: [11, 12, 13, 14, 15, 16, 17, 18]
✓ Resultado corregido y guardado en: C:\Users\Laura\Desktop\datos_sensor\09-05-2025\resultado_completo_09-05-2025.json

📅 Día: 12-05-2025 - Horas faltantes: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 19, 20, 21, 22, 23]
✓ Resultado corregido y guardado en: C:\Users\Laura\Desktop\datos_sensor\12-05-2025\resultado_completo_12-05-2025.json
✓ Día 13-05-2025 ya tiene las 24 horas. No se necesita corrección.
✓ Día 14-05-2025 ya tiene las 24 horas. No se necesita corrección.
✓ Día 15-05-2025 ya tiene las 24 horas. No se necesita corrección.
✓ Día 16-05-2025 ya tiene las 24 horas. No se necesita corrección.
✓ Día 17-05-2025 ya tiene las 24 horas. No se necesita corrección.
✓ Día 18-05-2025 ya tiene las 24 horas. No se necesita corrección.
✓ Día 19-05-2025 ya tiene las 24 horas. No se necesita corrección.
✓ Día 20-05-2025 ya tiene las 24 horas. No se necesita corrección.

```

**Ilustración 59. Terminal de Python programa de relleno de datos. Fuente: elaboración propia.**



**Ilustración 60. Visualización comparativa de datos "procesados crudos" y datos "procesados rellenados".**

**Nota:** se resaltaron los valores en los cuales se realizó el relleno de datos ■ además de resaltar las horas en las cuales se registraron datos ■ medidos por el sistema.

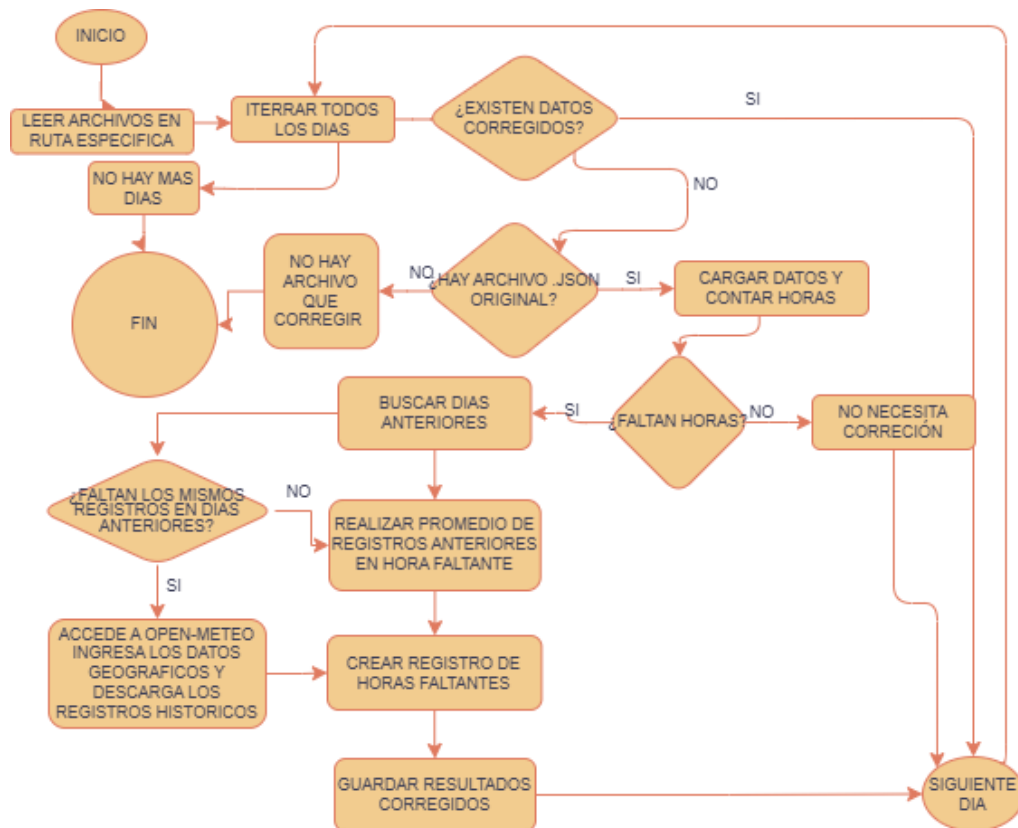


Ilustración 61. Diagrama de flujo script relleno de datos faltantes. Fuente: elaboración propia.

### 3.4.7 Validación de la arquitectura

Con el fin de garantizar que la arquitectura propuesta se desempeña de manera correcta, se realiza una validación basada en tres enfoques: pruebas técnicas funcionales, revisión comparativa de arquitectura en la nube (datos medidos y datos almacenados) y validación la visualización de los datos.

La validación técnica consistió en verificar el funcionamiento de los servicios implementados en Amazon Web Services (AWS), tales como IoT Core, S3, Lambda y SageMaker. Se evaluó el flujo completo de datos desde la adquisición en el sistema IoT (Raspberry Pi 4 Modelo B con sensores SHTC3 y MQ135), su transmisión a través del protocolo MQTT hacia AWS IoT Core, el almacenamiento en S3 y posterior procesamiento automatizado mediante funciones Lambda y SageMaker. Finalmente, los datos procesados se visualizaron en Tableau, permitiendo interpretar condiciones de confort térmico y calidad del aire.

### 3.4.8 Validación de recepción y almacenamiento de los datos

La comparativa entre los datos medidos por el sistema IoT y los almacenados diariamente en los buckets de la arquitectura en nube dentro de AWS, permiten evaluar el desempeño del sistema respecto a la transmisión y llegada de la información, desde la etapa de adquisición hasta el almacenamiento final. Esta evaluación es importante para validar la efectividad del flujo de datos que se inicia en Node-RED, continúa a través de AWS IoT Core y concluye en los servicios de almacenamiento Amazon S3.

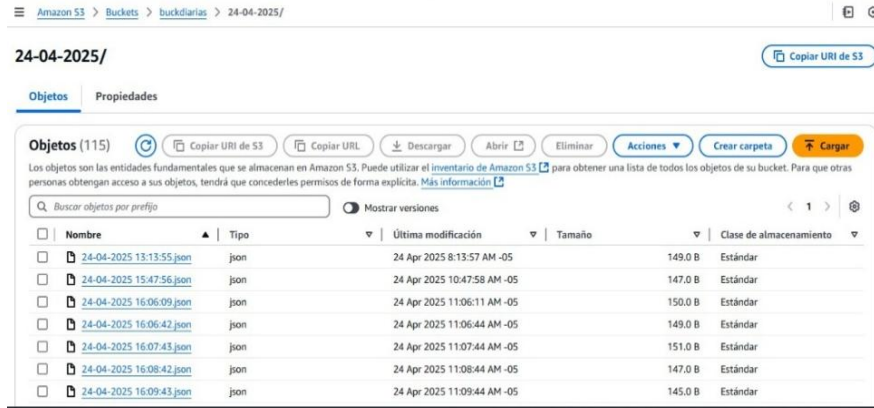
Para esta validación, se recopilaron y analizaron los datos correspondientes a distintos días de operación, contrastando los valores obtenidos por los sensores con los registros almacenados diariamente en el bucket S3 (bucket diarias). Este análisis permite verificar la integridad de los datos a lo largo de la arquitectura, así como identificar posibles pérdidas, inconsistencias o retrasos en la transmisión, la información se puede apreciar en la Tabla 6:

Tabla 6. Análisis de datos recopilados. Fuente: elaboración propia

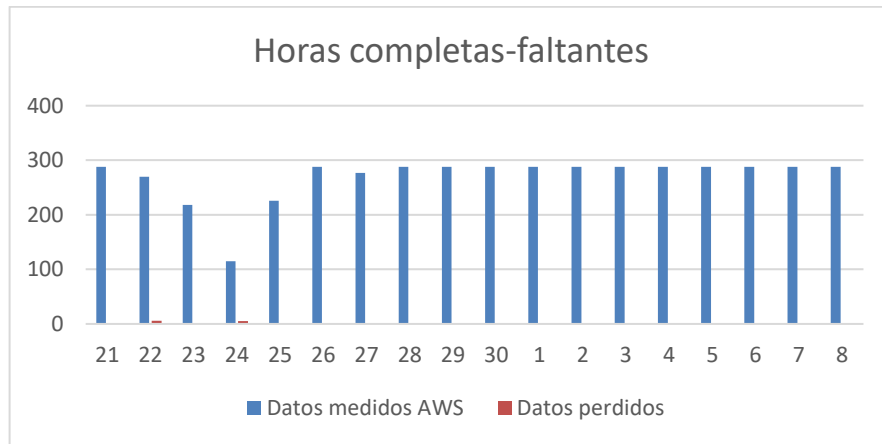
Día	Mes	Horas con datos	Horas sin datos	Horas perdidas (%)	Datos equivalentes (archivos.JSON)	Datos medidos AWS (archivos.JSON)
21	4	24	0	0%	288	288
22	4	22	2	8%	264	270
23	4	18	6	25%	216	218
24	4	10	14	58%	120	115
25	4	18	5	21%	228	218
26	4	24	0	0%	288	288
27	4	23	1	4%	276	288
28	4	24	0	0%	288	288
29	4	24	0	0%	288	288
30	4	24	0	0%	288	288
1	5	24	0	0%	288	288
2	5	24	0	0%	288	288
3	5	24	0	0%	288	288
4	5	24	0	0%	288	288
5	5	24	0	0%	288	288
6	5	24	0	0%	288	288
7	5	24	0	0%	288	288
8	5	24	0	0%	288	288

Se observa que algunos días presentaron una recopilación completa de datos, con 24 horas registradas y sin pérdidas, tanto en términos de tiempo como de volumen de información. Por otro lado, el día 24 de abril, mostró la mayor pérdida, con solo 10 horas de datos disponibles, lo que representa un 58 % de horas sin registro. Se generaron las gráficas correspondientes que ilustran el comportamiento de los datos

recibidos como se observa en la Ilustración 63. En general, se puede inferir que la pérdida de datos en términos de volumen es moderada y, en varios casos, inferior al porcentaje de horas sin información, lo que sugiere una concentración mayor de datos en las horas activas.



**Ilustración 62. Datos almacenados en Bucket (buckdiarias), para análisis de los datos recopilados frente a los estimados. Fuente: elaboración propia**



**Ilustración 63. Grafica de horas completas frente a horas faltantes. Fuente: elaboración propia**

También se realizó una comparación entre los datos registrados en los archivos JSON y los datos estimados a partir de las horas en las que se obtuvo información válida. Para esta estimación, se consideró un promedio de 12 datos por hora. Utilizando este valor junto con el número de horas en las que se registraron datos, se calculó la cantidad aproximada que debieron haberse recibido en cada día esto se ve reflejado en la Tabla 7 y en la Ilustración 64.

**Tabla 7. Datos de AWS frente a datos aproximados por día (horas validas). Fuente: elaboración propia**

Día	Datos equivalentes	Datos medidos AWS	Datos perdidos	% Datos perdidos
-----	--------------------	-------------------	----------------	------------------

21	288	288	0	0,00%
22	264	270	6	2,27%
23	216	218	2	0,92%
24	120	115	5	4,17%
25	228	226	2	0,88%
26	288	288	0	0,00%
27	276	277	1	0,36%
28	288	288	0	0,00%
29	288	288	0	0,00%
30	288	288	0	0,00%
1	288	288	0	0,00%
2	288	288	0	0,00%
3	288	288	0	0,00%
4	288	288	0	0,00%
5	288	288	0	0,00%
6	288	288	0	0,00%
7	288	288	0	0,00%
8	288	288	0	0,00%

Estos datos nos permiten evaluar el desempeño de la arquitectura y el sistema IoT de medición, por una parte, la correcta recolección y poca perdida de datos en el transcurso del día, y por otro lado el correcto almacenamiento de las variables dentro de los buckets de AWS.

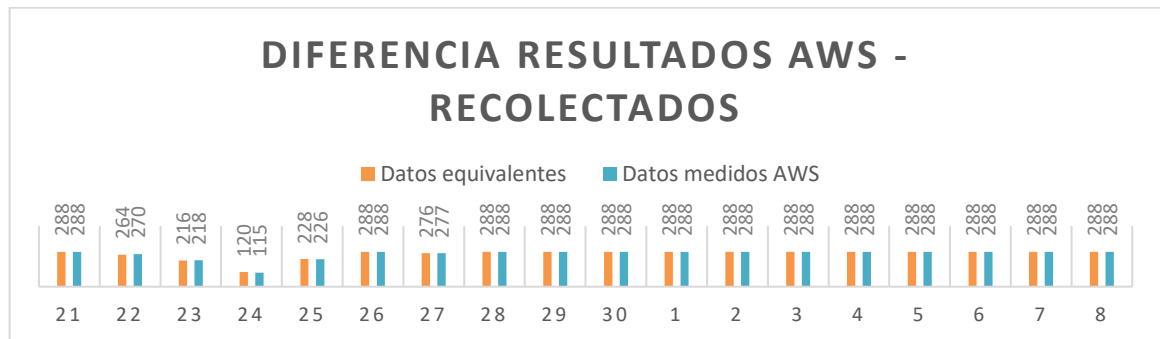


Ilustración 64. Grafica de diferencia de resultados AWS - Recolectados. Fuente: elaboración propia

### 3.4.9 Validación de sistema IoT

La implementación de la simulación computacional del aula mediante ANSYS Fluent permitió determinar la ubicación óptima del sistema IoT dentro del espacio, lo que resultó fundamental para garantizar una recolección eficiente y confiable de las variables ambientales (temperatura, humedad y CO2). Esta decisión basada en el modelado del flujo de aire y la distribución térmica facilitó que los sensores captaran variaciones reales de temperatura, humedad y CO2 de forma estable y continua.

A partir de esto, se realizó una comparación entre las variables de temperatura y humedad obtenidas por el sistema IoT y los valores simulados, complementando el análisis con los registros meteorológicos históricos del municipio de Fusagasugá

Ilustración 18, los cuales indican una temperatura promedio entre 18°C y 22°C y una humedad relativa entre 60% y 75%.

Los resultados evidencian una alta correlación entre los tres conjuntos de datos: simulación, mediciones reales y referencias climáticas. Esta coherencia valida tanto la correcta ubicación del sistema IoT como su funcionamiento en condiciones reales, demostrando su capacidad para registrar fielmente el comportamiento térmico y ambiental de la sala de prueba de concepto, se presentan los resultados del sistema IoT en la Tabla 8.

**Tabla 8. Promedio de temperatura y humedad registrados por el sistema IoT. Fuente: elaboración propia.**

Fecha	Prom. Humedad	Prom. Temperatura
21-04-2025	67.3	20.9
22-04-2025	65.8	20.7
23-04-2025	65.0	19.6
24-04-2025	64.6	22.1
24-04-2025	64.6	22.4
25-04-2025	69.5	19.7
25-04-2025	69.7	19.6
26-04-2025	70.3	19.8
27-04-2025	66.6	20.8
28-04-2025	69.7	20.2
29-04-2025	67.2	20.5
30-04-2025	72.5	19.6

### 3.4.10 Resumen de la validación

Se realiza una tabla con el fin de evidenciar la respectiva validación de la arquitectura en sus diferentes etapas y fases, como se observa en la Tabla 9:

**Tabla 9. Resumen de validación. Fuente: elaboración propia.**

Validaciones	Descripción	Evidencia
<b>Funcionalidad del sistema IoT</b>	Captura y envío de datos desde sensores hacia la nube.	Transmisión en Node-RED, registros en AWS IoT Core.
<b>Recepción y almacenamiento en la nube</b>	Correcto enrutamiento de datos y almacenamiento en formato JSON en S3.	Objetos JSON almacenados diariamente en buckets organizados por fecha.
<b>Procesamiento de datos</b>	Activación automática por medio de Lambda para la ejecución en SageMaker.	Logs de AWS Lambda y scripts en S3 utilizados por SageMaker para análisis.
<b>Visualización de datos</b>	Datos ambientales y métricas mostradas en dashboard.	Visualizaciones en Tableau (gráficas de CO <sub>2</sub> , temperatura, humedad, etc.).
<b>Comparación con simulación CFD</b>	Coincidencia entre zonas de mala ventilación y datos reales de sensores.	Análisis gráfico entre simulaciones de ANSYS Fluent y datos reales recolectados.
<b>Limitaciones de los servicios</b>	Evaluación de desempeño del sistema bajo recursos limitados	Pruebas ejecutadas con límites de recursos gratuitos de AWS,

		sin errores ni costos inesperados.
--	--	------------------------------------

## 4. ANÁLISIS DE RESULTADOS Y DISCUSIÓN

Se desarrolló un segundo script que permite identificar, dentro de las horas críticas procesadas, el valor máximo registrado para cada variable, así como la hora en la que dicho valor fue detectado. Asimismo, se evaluó si estos valores se encontraban dentro de los rangos óptimos de calidad del aire y confort térmico, además de esto se analizaron los datos con el fin de evaluar si los martes se evidencia un incremento de CO2 ya que se realiza una reunión de profesores por lo que el espacio presenta una alta densidad de personas ocupándolo. Los resultados obtenidos fueron almacenados en la Tabla 10 con el fin de facilitar su análisis.

```

📅 Día: 23-04-2025
🌡️ Temperatura máxima: 26.67 °C a las 2025-04-23 13:35:10
Estado: Alto por 6.7%
💧 Humedad máxima: 67.89% a las 2025-04-23 06:59:08
Estado: Alto por 13.2%
👤 CO2 máximo: 581.62 ppm a las 2025-04-22 20:53:06
Estado: Óptimo (0%)

📅 Día: 24-04-2025
🌡️ Temperatura máxima: 24.60 °C a las 2025-04-24 13:15:08
Estado: Óptimo (0%)
💧 Humedad máxima: 73.37% a las 2025-04-24 08:13:54
Estado: Alto por 22.3%
👤 CO2 máximo: 630.50 ppm a las 2025-04-24 12:10:08
Estado: Óptimo (0%)

```

Ilustración 65. Resultados de script de horas críticas. Fuente: elaboración propia

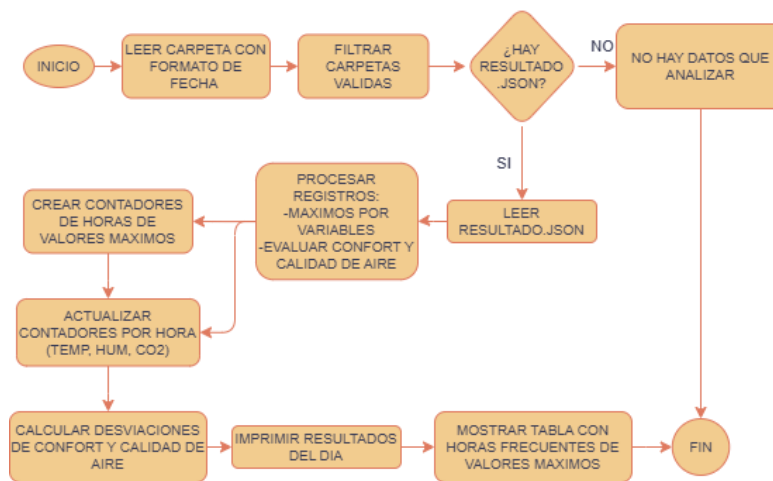


Ilustración 66. Detección de horas críticas. Fuente: elaboración propia

Tabla 10. Horas más frecuentes de valores máximos. Fuente: elaboración propia

Día	Temp. máx (°C)	Hora Temp.	Estado temp.	Hum. máx (%)	Hora Hum.	Estado hum.	CO2 máx (ppm)	Hora CO2	Estado CO2
1/05/2025	25.05	13:05:17	Alto por 0.2%	74.27	6:50:16	Alto por 23.8%	576.74	21:40:15	Óptimo
2/05/2025	23.68	13:15:18	Óptimo	73.14	19:15:17	Alto por 21.9%	581.62	20:55:17	Óptimo
3/05/2025	22.65	12:40:20	Óptimo	77.25	8:10:20	Alto por 28.8%	576.74	20:20:19	Óptimo
4/05/2025	23.21	9:15:22	Óptimo	77.76	18:55:22	Alto por 29.6%	591.40	22:10:21	Óptimo
5/05/2025	20.55	14:20:24	Óptimo	82.17	8:30:23	Alto por 37.0%	679.37	17:25:24	Óptimo
6/05/2025	22.45	12:15:25	Óptimo	78.38	7:15:25	Alto por 30.6%	757.18	15:50:25	Alto dentro de umbral
7/05/2025	21.41	13:55:27	Óptimo	81.44	8:20:27	Alto por 35.7%	620.72	8:35:27	Óptimo
8/05/2025	24.07	12:00:29	Óptimo	77.44	19:00:28	Alto por 29.1%	635.39	17:25:29	Óptimo
21/04/2025	22.43	18:48:02	Óptimo	61.71	18:57:02	Alto por 2.8%	576.74	18:59:02	Óptimo
22/04/2025	24.48	15:41:35	Óptimo	70.33	10:05:05	Alto por 17.2%	789.39	10:34:05	Alto dentro de umbral
23/04/2025	26.67	13:35:10	Alto por 6.7%	67.89	6:59:08	Alto por 13.2%	581.62	20:53:06	Óptimo
24/04/2025	24.60	13:15:08	Óptimo	73.37	8:13:54	Alto por 22.3%	630.50	12:10:08	Óptimo
25/04/2025	24.18	10:30:10	Óptimo	73.48	6:40:09	Alto por 22.5%	610.95	21:10:09	Óptimo
26/04/2025	21.68	11:00:12	Óptimo	74.52	18:50:12	Alto por 24.2%	562.07	19:20:11	Óptimo
27/04/2025	27.71	14:20:13	Alto por 10.8%	77.53	1:50:12	Alto por 29.2%	606.06	20:45:12	Óptimo
28/04/2025	22.68	12:00:14	Óptimo	75.42	7:50:14	Alto por 25.7%	786.90	8:20:14	Alto dentro de umbral
29/04/2025	25.48	14:20:15	Alto por 1.9%	75.78	7:00:15	Alto por 26.3%	890.10	14:20:15	Alto dentro de umbral
30/04/2025	22.75	15:25:15	Óptimo	78.85	8:40:15	Alto por 31.4%	601.17	18:00:15	Óptimo

Nota: se resaltaron las casillas de acuerdo con la condición en la que se encontraba las variables, siendo   un valor óptimo dentro de los umbrales establecidos,   un valor alto por encima de los umbrales establecidos y   para los valores altos pero que se encuentran dentro de los umbrales establecidos.

Durante el periodo analizado, las temperaturas máximas se mantuvieron en gran medida dentro del rango óptimo, excepto los días 1, 21, 23 y 27 de abril, en los

cuales se registraron leves incrementos con variaciones entre el 0.2% y el 10.8%. En cuanto a la humedad relativa, se evidenció un valor elevado por encima del umbral establecido, con incrementos que oscilaron entre el 13.2% y el 31.4%, lo que indica una condición constante de humedad elevada. Por otra parte, los niveles de CO2 se mantuvieron en general dentro de los parámetros aceptables, aunque se detectaron concentraciones elevadas el 22 (789.39 ppm), 28 (786.90 ppm) y 29 de abril (890.10 ppm), así como el 6 de mayo (757.2 ppm). Este último día, al igual que el 22 y 29 de abril, corresponde a un martes, ya que se realiza una reunión de alta ocupación y ventilación deficiente, como se observa en los resultados de recopilados el día 29, evidenciando un pico de CO2 y de temperatura, véase en la Ilustración 67.

Los resultados indican una estabilidad térmica adecuada, pero revelan deficiencias en el control de la humedad y en la calidad del aire en días específicos. La humedad persistentemente elevada puede afectar el confort térmico. Además, los picos de CO2 en jornadas concurridas mayoritariamente los martes durante la reunión de docentes de 2-4 pm reflejan la necesidad de mejorar la ventilación.

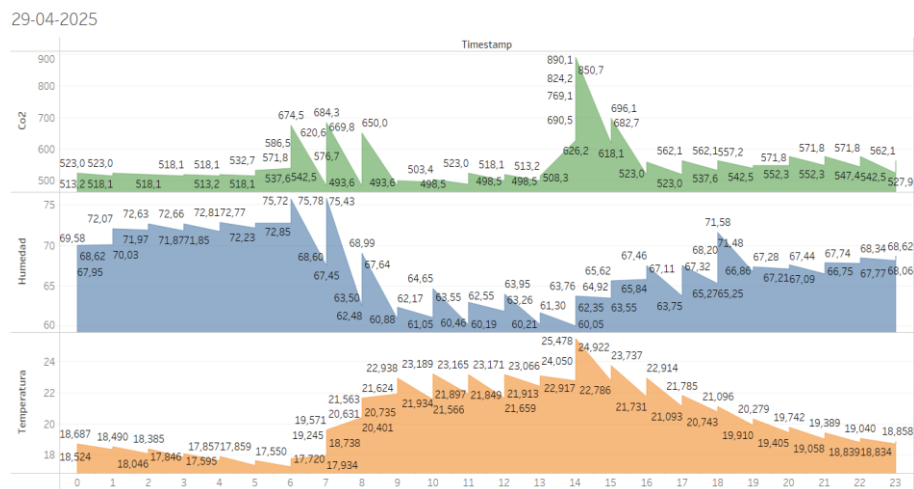


Ilustración 67. Valores altos día 29 de abril de 2025. Fuente: elaboración propia

Los datos obtenidos revelan un desempeño estable del sistema IoT y de la arquitectura en nube en cuanto a recolección y procesamiento de información. Sin embargo, se identifican patrones elevados para algunas variables relacionadas con la calidad del aire y el confort térmico dentro del espacio, en especial durante eventos de alta ocupación. Esto resalta la necesidad de implementar un sistema de ventilación que pueda funcionar de acuerdo con la ocupación del aula y del comportamiento de las variables críticas. Además, la detección de humedad elevada indica que el área ocupada no cuenta con los requerimientos para una correcta circulación del aire, con extractores o suficientes salidas de aire natural.

## **5 . CONCLUSIONES Y TRABAJOS FUTUROS**

---

### **5.1.1. CONCLUSIONES**

- La caracterización de la ventilación de la sala de prueba de concepto con Ansys Fluent, permitió comprobar las observaciones sobre la infraestructura, la cual carece de suficiente circulación de aire. Por medio de los escenarios de ventilación natural cruzada se logró identificar la ubicación para instalar el sistema IoT, de modo que tomara todas las variaciones del ambiente de la sala.
- Con la prueba de concepto se consiguió implementar y validar un sistema IoT fiable, que monitorea valores coherentes. Esto se pudo evidenciar al comparar las mediciones de temperatura y humedad con los históricos meteorológicos en meteoblue; el CO2 estuvo dentro de los valores normales y varió de acuerdo con la ocupación de la sala.
- El sistema IoT permite trabajar un solo sensor de tipo I2C al solo poseer dos pines para este fin (SDA y SCL). Con este sistema se manejó un muestreo de 5 minutos, que está dentro de lo recomendado en sistemas de ventilación.
- Con la aplicación del servicio de ventilación de la arquitectura en nube en la prueba de concepto, se concluye que la sala no ofrece confort térmico. Debido a la región, la humedad es superior al rango seguro (30 - 50%), la ventilación natural no es suficiente frente a horas críticas como las primeras horas en la mañana y el medio día, donde la temperatura se sale del rango seguro (20 – 24 °C) y los niveles de CO2 rozan el umbral máximo seguro con el aumento de la ocupación en la sala. Por ende, se identifica la presencia de síndrome de edificio enfermo.
- Con la validación de la arquitectura completa aplicada a una prueba de concepto se evidenció el monitoreo y procesamiento diario fiable de los problemas de ventilación en la sala en experimentación, que se presentan a nivel de campus y compromete el estado de sus ocupantes.

### **5.1.2. PERSPECTIVAS DE TRABAJO FUTURO**

Este proyecto abre la puerta a un campo de estudio en cuanto a la ventilación inteligente en aulas académicas, con el que se presenta una arquitectura escalable a más dispositivos, sensores y flujos de datos que tenga en cuenta los demás factores que inciden en la mala ventilación como la ocupación, el ruido, compuestos volátiles, entre otros.

La arquitectura presentada es capaz de exportar datos que se pueden visualizar con herramientas más avanzadas como el software de Ansys Fluent para generar

imágenes térmicas y procesamientos de la estimación de la ventilación por medio del modelado computacional. También se pueden obtener estimaciones futuras de acuerdo con los datos del clima, velocidad del viento de la región y tanto AWS como Ansys Fluent pueden intercambiar datos entre sí bajo la licencia de Ansys y una versión paga de AWS.

Con esta arquitectura también se presentó un servicio que tiene mucho para explotar y es Amazon SageMaker. Este servicio implementa aprendizaje automatizado con el cual se pueden crear modelos predictivos, realizar análisis y desempeñar tareas con IA generativa. Con este servicio es posible programar decisiones para controlar un sistema de ventilación inteligente en un futuro.


## 6. BIBLIOGRAFIA

- [1] “Indoor Air Quality | US EPA”. Consultado: el 20 de mayo de 2025. [En línea]. Disponible en: [https://www.epa.gov/report-environment/indoor-air-quality?utm\\_source=chatgpt.com](https://www.epa.gov/report-environment/indoor-air-quality?utm_source=chatgpt.com)
- [2] “Introduction to Indoor Air Quality | US EPA”. Consultado: el 20 de mayo de 2025. [En línea]. Disponible en: [https://www.epa.gov/indoor-air-quality-iaq/introduction-indoor-air-quality?utm\\_source=chatgpt.com](https://www.epa.gov/indoor-air-quality-iaq/introduction-indoor-air-quality?utm_source=chatgpt.com)
- [3] L. Morawska *et al.*, “Mandating indoor air quality for public buildings If some countries lead by example, standards may increasingly become normalized”, *Science (1979)*, vol. 383, núm. 6690, pp. 1418–1420, mar. 2024, doi: 10.1126/SCIENCE.ADL0677.
- [4] J. M. Rey-Hernández, Y. Arroyo-Gómez, J. F. San José-Alonso, y F. J. Rey-Martínez, “Assessment of natural ventilation strategy to decrease the risk of COVID 19 infection at a rural elementary school”, *Heliyon*, vol. 9, núm. 7, jul. 2023, doi: 10.1016/J.HELIYON.2023.E18271.
- [5] J. Allen, J. Spengler, E. Jones, y J. Cedeno-Laurent, “Guía en 5 pasos para medir la tasa de renovación de aire en aulas”, 2020, Consultado: el 2 de abril de 2025. [En línea]. Disponible en: [www.ForHealth.org](http://www.ForHealth.org)
- [6] M. C. Minguillón, X. Querol, J. M. Felisi, y T. Garrido, “Guía para ventilación en aulas”.
- [7] “Lengthy delays in air-conditioning installation creating sweltering conditions in NSW classrooms - ABC News”. Consultado: el 20 de mayo de 2025. [En línea]. Disponible en: [https://www.abc.net.au/news/2025-02-24/nsw-cooler-classrooms-students-learning-heat-temperature/104951660?utm\\_source=chatgpt.com](https://www.abc.net.au/news/2025-02-24/nsw-cooler-classrooms-students-learning-heat-temperature/104951660?utm_source=chatgpt.com)
- [8] H. W. Brink, M. G. L. C. Loomans, M. P. Mobach, y H. S. M. Kort, “Classrooms’ indoor environmental conditions affecting the academic achievement of students and teachers in higher education: A systematic literature review”, *Indoor Air*, vol. 31, núm. 2, p. 405, mar. 2020, doi: 10.1111/INA.12745.
- [9] “La calidad del aire en el salón de clases influye en el aprendizaje”. Consultado: el 2 de abril de 2025. [En línea]. Disponible en: <https://www.benq.com/es-mx/business/resource/trends/por-que-es-importante-la-calidad-del-aire-en-las-aulas.html>
- [10] EDUARDO BOTELLA CERECEDA, “SINDROME DEL EDIFICIO ENFERMO”, TESIS. Consultado: el 20 de mayo de 2025. [En línea]. Disponible en:

[https://dspace.umh.es/bitstream/11000/7123/1/Botella\\_Cereceda\\_Eduardo\\_TFM.pdf?utm\\_source=chatgpt.com](https://dspace.umh.es/bitstream/11000/7123/1/Botella_Cereceda_Eduardo_TFM.pdf?utm_source=chatgpt.com)

- [11] A. I. G. Monroy, L. M. Allende, y E. E. L. González, “Como Afecta El Estrés del Docente, en el Desempeño Académico del Estudiante de UPIBI-IPN”, *Revista Electrónica sobre Tecnología, Educación y Sociedad*, vol. 4, núm. 7, ene. 2017, Consultado: el 4 de mayo de 2025. [En línea]. Disponible en: <https://ctes.org.mx/index.php/ctes/article/view/661>
- [12] A. Kharbouch *et al.*, “Internet-of-Things Based Hardware-in-the-Loop Framework for Model-Predictive-Control of Smart Building Ventilation”, *Sensors*, vol. 22, núm. 20, 2022, doi: 10.3390/s22207978.
- [13] G. Chiesa, S. Cesari, M. Garcia, M. Issa, y S. Li, “Multisensor IoT platform for optimising IAQ levels in buildings through a smart ventilation system”, *Sustainability (Switzerland)*, vol. 11, núm. 20, 2019, doi: 10.3390/su11205777.
- [14] M. Mahbub, M. M. Hossain, y M. S. A. Gazi, “IoT-Cognizant cloud-assisted energy efficient embedded system for indoor intelligent lighting, air quality monitoring, and ventilation”, *Internet of Things (Netherlands)*, vol. 11, 2020, doi: 10.1016/j.iot.2020.100266.
- [15] K. Ashton, “That Internet of Things Thing”, *RFID Journal*, vol. 4986, 2009.
- [16] Q. Zhang, L. Cheng, y R. Boutaba, “Cloud computing: State-of-the-art and research challenges”, *Journal of Internet Services and Applications*, vol. 1, núm. 1, 2010, doi: 10.1007/s13174-010-0007-6.
- [17] Amazon AWS, “Amazon Web Services (AWS)”, Amazon AWS. Consultado: el 2 de abril de 2025. [En línea]. Disponible en: <https://aws.amazon.com/es/what-is-aws/>
- [18] G. Guyot, M. H. Sherman, y I. S. Walker, “Smart ventilation energy and indoor air quality performance in residential buildings: A review”, 2018. doi: 10.1016/j.enbuild.2017.12.051.
- [19] W. Ricardo, R. Ayala, U. De Bogotá, y J. T. Lozano, “UTILIZACIÓN DE DINÁMICA DE FLUIDOS COMPUTACIONAL CFD-ANSYS FLUENT PARA LA SIMULACIÓN DE UN GASIFICADOR”.
- [20] JOYCE ALMEIDA WINFIELD, “VENTILACIÓN NATURAL Y CALIDAD EN EL AMBIENTE INTERIOR EN EDIFICIOS DEL CENTRO DE LA CIUDAD DE MÉXICO”, UNAM , Mexico, 2019. Consultado: el 2 de abril de 2025. [En línea]. Disponible en: [http://132.248.9.195/ptb2010/noviembre/0664260/0664260\\_A1.pdf](http://132.248.9.195/ptb2010/noviembre/0664260/0664260_A1.pdf)

- [21] X. Shen, Q. Sun, G. Mosey, J. Ma, L. Wang, y M. Ge, “Benchmark of plant-based VOCs control effect for indoor air quality: Green wall case in smith campus at Harvard University”, *Science of the Total Environment*, vol. 906, 2024, doi: 10.1016/j.scitotenv.2023.166269.
- [22] R. E. Rojano, Y. I. Mendoza, H. Arregoces, y G. M. Restrepo, “Dispersión de Contaminantes del Aire (PM10, NO2, CO, COV y HAP) emitidos desde una Estación Modular de Compresión, Tratamiento y Medición de Gas Natural Dispersion of Air Pollutants (PM10, NO2, CO, VOC and HAP) emitted from a Modular Station of Compression, Processing and Measurement of Natural Gas”, *Información Tecnológica*, vol. 27, núm. 5, pp. 99–110, 2016, doi: 10.4067/S0718-07642016000500012.
- [23] Cristian Mosquera Escobar, “Desarrollo de un sistema de monitoreo remoto de la calidad del aire en espacios interiores”, Universidad de Antioquia, Medellín, 2023. Consultado: el 2 de abril de 2025. [En línea]. Disponible en: <https://bibliotecadigital.udea.edu.co/server/api/core/bitstreams/1aafdf7a-118c-4131-aea1-f8454abd046d/content>
- [24] M. Luna-López *et al.*, “Sistema inteligente de monitoreo para condiciones ambientales en Industria 4.0”, *Científica*, vol. 25, núm. 2, 2021, doi: 10.46842//ipn.cien.v25n2a07.
- [25] J. A. C. Correa, S. B. S. Mora, B. M. Delgado, C. D. E. Amado, y D. G. Ibarra, “Sistema de monitorización y detección de incendios forestales basado en redes de sensores inalámbricos”, *Scientia et Technica*, vol. 27, núm. 2, pp. 89–96, jun. 2022, doi: 10.22517/23447214.24784.
- [26] “Ladrillo | Densidad, capacidad calorífica, conductividad térmica”. Consultado: el 8 de mayo de 2025. [En línea]. Disponible en: <https://material-properties.org/es/ladrillo-densidad-capacidad-calorifica-conductividad-termica/>
- [27] “FICHA DE PRODUCTO”, Consultado: el 20 de mayo de 2025. [En línea]. Disponible en: [https://www.volcan.cl/system/files/iso\\_9001.pdf](https://www.volcan.cl/system/files/iso_9001.pdf)
- [28] “Vidrio”. Consultado: el 8 de mayo de 2025. [En línea]. Disponible en: [https://es.wikipedia.org/wiki/Vidrio?utm\\_source=chatgpt.com](https://es.wikipedia.org/wiki/Vidrio?utm_source=chatgpt.com)
- [29] “ANSYS FLUENT 12.0 User’s Guide”. Consultado: el 20 de mayo de 2025. [En línea]. Disponible en: [https://www.afs.enea.it/project/neptunius/docs/fluent/html/ug/main\\_pre.htm](https://www.afs.enea.it/project/neptunius/docs/fluent/html/ug/main_pre.htm)
- [30] “Datos climáticos y meteorológicos históricos simulados para Fusagasugá - meteoblue”. Consultado: el 2 de abril de 2025. [En línea]. Disponible en: [https://www.meteoblue.com/es/tiempo/historyclimate/climatemodelled/fusagasug%C3%A1\\_colombia\\_3682274](https://www.meteoblue.com/es/tiempo/historyclimate/climatemodelled/fusagasug%C3%A1_colombia_3682274)

- [31] “Datasheet SHTC3”, Consultado: el 20 de mayo de 2025. [En línea]. Disponible en: [www.sensirion.com](http://www.sensirion.com)
- [32] “Air Quality Gas Sensor”, 2015, Consultado: el 20 de mayo de 2025. [En línea]. Disponible en: [www.winsen-sensor.com](http://www.winsen-sensor.com)
- [33]  Docs | Open-Meteo.com”. Consultado: el 21 de mayo de 2025. [En línea]. Disponible en: [https://open-meteo.com/en/docs?hourly=temperature\\_2m,relative\\_humidity\\_2m&latitude=4.3365&longitude=-74.3638&timezone=auto&forecast\\_days=16&time\\_mode=time\\_interval&start\\_date=2025-05-15&end\\_date=2025-05-29](https://open-meteo.com/en/docs?hourly=temperature_2m,relative_humidity_2m&latitude=4.3365&longitude=-74.3638&timezone=auto&forecast_days=16&time_mode=time_interval&start_date=2025-05-15&end_date=2025-05-29)