

# DOMOTIZACIÓN DE UN EDIFICIO MEDIANTE TECNOLOGÍA SIGFOX

-

# BUILDING AUTOMATION USING SIGFOX TECHNOLOGY



TRABAJO FIN DE MÁSTER  
CURSO 2025-2026

AUTOR  
EMILIANO FRANCO HURTADO

DIRECTOR  
FRANCISCO DANIEL IGUAL PEÑA

MÁSTER EN INTERNET DE LAS COSAS  
FACULTAD DE INFORMÁTICA  
UNIVERSIDAD COMPLUTENSE DE MADRID

# DOMOTIZACIÓN DE UN EDIFICIO MEDIANTE TECNOLOGÍA SIGFOX

-

## BUILDING AUTOMATION USING SIGFOX TECHNOLOGY

TRABAJO DE FIN DE MÁSTER EN INTERNET DE LAS COSAS  
DEPARTAMENTO DE ARQUITECTURA DE COMPUTADORES Y  
AUTOMÁTICA

AUTOR

EMILIANO FRANCO HURTADO

DIRECTOR

FRANCISCO DANIEL IGUAL PEÑA

**CONVOCATORIA: ENERO 2026**

**CALIFICACIÓN: 6.5**

MÁSTER EN INTERNET DE LAS COSAS  
FACULTAD DE INFORMÁTICA  
UNIVERSIDAD COMPLUTENSE DE MADRID

26 ENERO DE 2026



## **AGRADECIMIENTOS**

En primera instancia, a todos los que me acompañaron, apoyaron y ayudaron en mi carrera profesional y académica; especialmente a mis padres sin los cuales este proyecto de completar un máster en otro país, en otro continente lejos del propio no hubiera sido posible.

Asimismo, agradecer a Unabiz por el préstamo de sus plataformas, documentación y sensores para la realización de este proyecto.



## RESUMEN

La sensorización de edificios constituye uno de los pilares fundamentales de la domótica y de las denominadas *Smart Cities*. Dado que las personas pasan la mayor parte de su tiempo en espacios interiores, el control y la monitorización de las condiciones ambientales adquieren una relevancia clave para la salud, el bienestar y la eficiencia energética. Entre los distintos parámetros que permiten evaluar la calidad del aire interior, la concentración de dióxido de carbono (CO<sub>2</sub>) y la temperatura destacan por ser indicadores ampliamente utilizados en entornos docentes, laborales y técnicos.

En este Trabajo Fin de Máster se desarrolla una plataforma IoT orientada a la monitorización de la calidad del aire en distintas salas de un edificio universitario. Para ello, se emplean sensores inalámbricos basados en la tecnología Sigfox, perteneciente a la familia de redes de área amplia y bajo consumo (LPWAN), especialmente adecuada para este tipo de aplicaciones debido a su largo alcance en interiores, bajo consumo energético y capacidad de operar sin saturar infraestructuras de red locales, a diferencia de tecnologías tradicionales como WiFi.

Los datos adquiridos por los sensores se integran en una plataforma IoT construida sobre Node-RED, desde donde se gestionan los flujos de procesamiento, el almacenamiento en una base de datos de series temporales y la visualización mediante paneles de control personalizados.

Adicionalmente, se implementa un modelo de análisis y predicción de datos basado en técnicas de *Machine Learning*, concretamente árboles de decisión, con el objetivo de estimar la evolución futura de las variables ambientales en función de los valores actuales, el día de la semana y el uso específico de cada sala. Este modelo permite anticipar situaciones de riesgo relacionadas con una mala calidad del aire y generar alertas preventivas para los ocupantes del edificio.

Los resultados obtenidos muestran que, aunque el comportamiento de las variables ambientales difiere en función del tipo de sala analizada, el modelo de predicción mejora su precisión a medida que aumenta el volumen de datos disponibles para el entrenamiento. Asimismo, la plataforma desarrollada demuestra ser flexible, escalable y fácilmente replicable, permitiendo la incorporación de nuevos sensores y espacios de manera sencilla gracias a las capacidades de integración de Node-RED.

### Palabras clave

Sigfox, domótica, *smart building*, calidad del aire interior, predicción de CO<sub>2</sub>, plataforma IoT.



## ABSTRACT

Building sensorization is a fundamental pillar of domotics and so-called *Smart Cities*. Since people spend most of their time indoors, monitoring and controlling indoor environmental conditions is essential for ensuring health, comfort, and energy efficiency. Among the different parameters used to assess indoor air quality, carbon dioxide concentration (CO<sub>2</sub>) and temperature stand out as widely adopted indicators in educational, professional, and technical environments.

This Master's Thesis presents the development of an IoT platform aimed at monitoring indoor air quality in different rooms of a university building. For this purpose, wireless sensors based on Sigfox technology, belonging to the family of Low Power Wide Area Networks (LPWAN), are deployed in classrooms, offices, and technical rooms. This technology is particularly suitable for such applications due to its long-range indoor coverage, low energy consumption, and its ability to operate without overloading local network infrastructures, in contrast to traditional technologies such as WiFi.

The data collected by the sensors are integrated into an IoT platform built on Node-RED, where data processing flows, time-series database storage, and visualization through customized dashboards are managed.

Additionally, a data analysis and prediction model based on *Machine Learning* techniques, specifically decision trees, is implemented to estimate the future evolution of environmental variables based on current measurements, the day of the week, and the specific use of each room. This approach enables the anticipation of potential risk situations related to poor air quality and the generation of preventive alerts for building occupants.

The results show that, although the behavior of environmental variables differs depending on the type of room analyzed, the prediction model improves its accuracy as the amount of training data increases. Furthermore, the developed platform proves to be flexible, scalable, and easily replicable, allowing new sensors and monitored

spaces to be incorporated in a simple and intuitive manner thanks to the integration capabilities of Node-RED.

**Keywords**

Sigfox, domotics, smart building, indoor air quality, CO<sub>2</sub> level prediction, IoT platform.

# ÍNDICE

Agradecimientos.....	III
Resumen .....	V
Abstract .....	VII
Índice .....	IX
ÍNDICE DE FIGURAS.....	XIII
Índice de tablas.....	XVII
Capítulo 1 - Introducción .....	19
1.1 Motivación .....	19
1.2 Objetivos.....	20
1.3 Estructura de la memoria.....	21
Capítulo 2 - Estado de la cuestión.....	24
2.1 Factores determinantes en la calidad del aire interior (CAI) .....	24
2.1.1 Dióxido de Carbono (CO <sub>2</sub> ) como Indicador de Ventilación.....	24
2.1.2 Humedad Relativa del Aire .....	25
2.1.3 Temperatura Ambiente.....	25
2.1.4 Interrelación de las Variables .....	26
2.2 Normativas y regulaciones .....	26
2.2.1 NTP 742: Ventilación General de Edificios .....	27
2.2.2 NTP 549: El CO <sub>2</sub> en la Evaluación de la Calidad del Aire Interior.....	27
2.2.3 Implicaciones para el Proyecto .....	28
2.3 Tecnologías de Comunicación LPWAN para Monitorización de Edificios .....	29
2.3.1 Sigfox: Una Red LPWAN Optimizada para Sensores.....	29
2.3.2 Tecnologías Alternativas: LoRaWAN, NB-IoT, LTE-M y WiFi.....	33

2.3.3 Análisis Comparativo y Justificación para Monitorización de Edificios .....	34
2.4 Métodos de tratamiento de datos.....	36
2.4.1 Clustering: Agrupamiento No Supervisado .....	36
2.4.2 Algoritmo K-Means y el Método del Codo.....	37
2.4.3 Métricas de Validación de la Calidad del Clustering .....	37
2.4.4 Aplicación en el Proyecto .....	38
2.5 Revisión del estado del arte .....	39
2.5.1 Comparativa de Tecnologías de Comunicación LPWAN .....	39
2.5.2 Plataformas IoT para Monitorización de Edificios.....	40
2.5.3 Machine Learning para la Predicción de la Calidad del Aire .....	41
2.5.4 Conclusión del Estado del Arte y Posicionamiento de este Trabajo.....	41
Capítulo 3 - Diseño de la solución .....	43
3.1 Descripción de instalaciones físicas .....	43
3.1.1 Aula de Clases B3 (Sensor ID: 1FC823F) .....	43
3.1.2 Despacho de Profesores (Sensor ID: C36596) .....	44
3.1.3 Sala de Servidores (Sensor ID: C4D841) .....	44
3.2 Arquitectura General del Sistema .....	45
3.3 Capa de Percepción: Sensores Sigfox Airwits CO2 Plus .....	47
3.4 Capa de Comunicación: Integración con la Red Sigfox.....	49
3.5 Capa de Plataforma: Procesamiento, Almacenamiento y Lógica de Negocio ..	50
3.5.1 Orquestación con Node-RED .....	50
3.5.2 Recepción y Decodificación de Datos.....	51
3.5.3 Almacenamiento en Base de Datos de Series Temporales: InfluxDB.....	51
3.5.4 Integración de Modelos de Machine Learning.....	52
3.6 Capa de Aplicación: Dashboard de Visualización y Control.....	52

Capítulo 4 - Análisis exploratorio de datos (EDA) .....	55
4.1 Introducción y Metodología del Análisis .....	55
4.2 Análisis del Sensor de Aula B3 - Facultad de Informática (ID: 1FC823F) .....	56
4.2.1 Análisis Exploratorio: Dataset de un Mes (febrero 2025) .....	56
4.2.2 Análisis Exhaustivo: Dataset de Diez Meses (Enero - Octubre 2025) .....	59
4.2.3 Conclusión del Análisis para el Aula B3 .....	61
4.3 Análisis del Sensor de Despacho de Profesores (ID: C36596) .....	62
4.4 Análisis del Sensor de Sala de Servidores (ID: C4D841) .....	65
4.5 Síntesis Comparativa y Conclusiones Generales del Análisis Exploratorio .....	67
Capítulo 5 - Implementación y validación del sistema .....	69
5.1 Diseño y Entrenamiento de Modelos Predictivos .....	69
5.1.1 Justificación de la Selección del Algoritmo .....	69
5.1.2 Ingeniería de Características y Construcción del Dataset .....	70
5.1.3 Entrenamiento y Evaluación Preliminar del Modelo de Calidad del Aire .....	71
5.1.4 Desarrollo del Modelo de Regresión para Predicción Numérica .....	72
5.2 Implementación de la Plataforma IoT con Node-RED .....	73
5.2.1 Arquitectura de Integración y Módulo de Adquisición de Datos .....	73
5.2.2 Módulo de Almacenamiento en Base de Datos de Series Temporales .....	76
5.2.3 Módulo de Visualización: Dashboard de Monitorización y Control .....	77
5.2.4 Módulo de Predicción en Tiempo Real: Integración de los Modelos de ML ...	79
5.2.5 Script de Predicción de CO <sub>2</sub> .....	80
5.2.6 Script de Predicción Multivariable .....	81
5.2.7 Conclusión de la Implementación .....	82
5.3 Validación Experimental y Análisis de Resultados .....	82
5.3.1 Metodología de Validación .....	83

5.3.2 Resultados del Modelo de Predicción de CO <sub>2</sub> para el Aula .....	83
5.3.3 Resultados del Modelo Multivariable para los Tres Sensores .....	85
5.3.4 Discusión y Análisis de Limitaciones.....	86
5.3.5 Conclusión de la Validación .....	87
Capítulo 6 - Conclusiones y trabajo futuro .....	89
6.1 Conclusiones.....	89
6.2 Líneas de Trabajo Futuro .....	91
Capítulo 7 - Introduction .....	93
7.1 Motivation .....	93
7.2 Objectives .....	94
7.3 Document Structure.....	95
Capítulo 8 - Conclusions and future work.....	97
8.1 Conclusions .....	97
8.2 Future Work Lines .....	98
Apéndice A: Script "predict_co2.py".....	101
Bibliografía.....	105

# ÍNDICE DE FIGURAS

Ilustración 2-1. Capas del Modelo OSI. ....	30
Ilustración 2-2. Capas de modelo OSI para mensaje UPLINK en Sigfox. ....	30
Ilustración 2-3. Capas de modelo OSI para mensaje DOWNLINK en Sigfox.....	30
Ilustración 2-4. Especificaciones de Sigfox. ....	31
Ilustración 2-5. Ejemplo de mensaje Uplink.....	32
Ilustración 3-1. Arquitectura en capas del sistema IoT para monitorización y predicción de la calidad del aire. El flujo de datos comienza en los sensores (Percepción), es transmitido vía Sigfox (Comunicación), procesado y almacenado por Node-RED e InfluxDB (Plataforma), y finalmente visualizado y consultado a través del Dashboard (Aplicación).....	47
Ilustración 3-2. Sensor AiWits CO2 Plus.....	48
Ilustración 3-3. Tabla de payload del sensor.....	49
Ilustración 3-4. Diagrama de funcionamiento de Ngrok.....	50
Ilustración 4-1. Distribución de datos de un mes del sensor 1FC823F.....	56
Ilustración 4-2. Cantidad de instancias con valores máximos y mínimos. ....	57
Ilustración 4-3. Matriz de correlación variables del sensor. ....	57
Ilustración 4-4. Gráfica de método del codo para encontrar K.....	57
Ilustración 4-5. Gráfica de cuatro clústeres comparando CO2 vs Temperatura. ....	58
Ilustración 4-6. Gráfica de los clústeres comparando CO2 vs Humedad. ....	58
Ilustración 4-7. Métricas de medición de 4 clústeres. ....	59
Ilustración 4-8. Métricas de medición de tres clústeres. ....	59
Ilustración 4-9. Gráfica de niveles de CO2 de acuerdo con días de la semana.....	59
Ilustración 4-10. Distribución de datos de 10 meses sensor 1FC823F. ....	60
Ilustración 4-11. Gráfica de nivel de CO2 sensor 1FC823F meses ene-nov.....	60

Ilustración 4-12. Gráfica del codo del sensor 1FC823F con 10 meses de datos. ....	61
Ilustración 4-13. Gráfica de niveles de CO2 según el día de la semana. ....	61
Ilustración 4-14. Distribución de datos de 10 meses sensor C36596.....	62
Ilustración 4-15. Matriz de correlación del sensor C36596.....	63
Ilustración 4-16. Gráfica del codo sensor C36596. ....	64
Ilustración 4-17. Clústeres de datos sensor C36596. ....	64
Ilustración 4-18. Distribución de datos de 10 meses sensor C4D841. ....	65
Ilustración 4-19. Matriz de correlación del sensor C4D841. ....	66
Ilustración 4-20. Gráfica del codo sensor C4D841.....	66
Ilustración 4-21. Clústeres de datos sensor C4D841.....	67
Ilustración 5-1. Distribución de los datos de Temperatura, CO <sub>2</sub> y Humedad.....	70
Ilustración 5-2. Tabla de rangos de niveles de CO2 por cada categoría "Action" en sensor 1FC823F.....	71
Ilustración 5-3. Matriz de confusión con nuevas predicciones de las clases. ....	71
Ilustración 5-4. Tabla de precisión de predicciones.....	72
Ilustración 5-5. Matriz de confusión de predicciones futuras.....	72
Ilustración 5-6. Tabla de precisión de predicciones futuras.....	72
Ilustración 5-7. Configuración de payload en backend.....	74
Ilustración 5-8. Backend Sigfox en configuración de callback de la API. ....	74
Ilustración 5-9. Flujo de recepción de backend de Sigfox.....	75
Ilustración 5-10. Función de decodificación de Payload. ....	75
Ilustración 5-11. Decodificación payload.....	76
Ilustración 5-12. Captura de pantalla de la database con datos de sensores.....	76
Ilustración 5-13. Captura del dashboard de Node-Red en interfaz de sensores. ....	77
Ilustración 5-14. Dashboard historico CO2.....	78

Ilustración 5-15. Histórico Temperatura y Humedad. ....	78
Ilustración 5-16. Captura del dashboard de Node-Red en vista de predicción.....	79
Ilustración 5-17. Dashboard predicciones. ....	79
Ilustración 5-18. Formato de JSON construido en Node-RED.....	80
Ilustración 5-19. Sección 1 de función main de predicción.....	81
Ilustración 5-20. Sección 2 de función main de predicción.....	82
Ilustración 5-21. Tabla comprobaciones CO2.....	85
Ilustración 5-22. Tabla comprobaciones Temperatura.....	85
Ilustración 5-23. Tabla comprobaciones Humedad.....	85



## ÍNDICE DE TABLAS

Tabla 1. Categorías de calidad del aire interior según concentración de CO <sub>2</sub> (NTP 742). .....	27
Tabla 2. Tipos de variables configurables para mensaje Uplink.....	32
Tabla 3. Comparativa de tecnologías IoT inalámbricas. ....	35
Tabla 4. Comparación de bases de datos. ....	52
Tabla 5. Cálculo de error en predicciones.....	84



# Capítulo 1 - Introducción

## 1.1 Motivación

El Internet de las Cosas (IoT) se define como un ecosistema de dispositivos y tecnologías interconectadas que permiten la adquisición, transmisión y procesamiento de datos, ya sea a través de servidores locales o infraestructuras en la nube. Esta capacidad de interconexión ha propiciado que objetos y sistemas cotidianos puedan adaptarse de forma inteligente a las necesidades de los usuarios, consolidando al IoT como una tecnología transversal presente en un número creciente de sectores industriales, científicos y sociales.

Dentro de este contexto, la domótica surge como una disciplina resultante de la integración del IoT en el ámbito de la edificación y el urbanismo, permitiendo la monitorización y automatización de edificios con el objetivo de mejorar la eficiencia energética, el confort y la seguridad de los ocupantes. En la actualidad existe una amplia oferta de sensores y dispositivos capaces de medir múltiples variables ambientales en espacios interiores, por lo que el principal reto ya no reside en la disponibilidad del hardware, sino en la selección de tecnologías de comunicación adecuadas y en la integración eficiente de los datos en plataformas que permitan su análisis y explotación.

En este sentido, las tecnologías de comunicación de bajo consumo y largo alcance, conocidas como *Low Power Wide Area Networks* (LPWAN), resultan especialmente relevantes para aplicaciones de sensorización de edificios. Estas tecnologías operan en bandas de frecuencia que presentan una elevada penetración en entornos urbanos y permiten evitar la saturación de redes locales tradicionales, como WiFi o redes celulares. Entre ellas, Sigfox destaca como una alternativa idónea para aplicaciones de monitorización continua, gracias a su bajo consumo energético, su amplia cobertura en interiores y su capacidad para soportar despliegues escalables de sensores con un mantenimiento mínimo.

Por otro lado, la calidad del aire interior constituye un factor determinante para la salud y el bienestar de las personas. Diversos organismos internacionales, como la Agencia de Protección Ambiental de Estados Unidos (EPA) o la Organización Mundial de la Salud (OMS), advierten de que el aire en espacios interiores puede presentar niveles de contaminación significativamente superiores a los del exterior, y que la exposición prolongada a ambientes mal ventilados puede derivar en problemas respiratorios y una reducción del rendimiento cognitivo. Teniendo en cuenta que los seres humanos pasan entre el 80 % y el 90 % de su tiempo en espacios cerrados (EPA, 2025), la monitorización de parámetros como el dióxido de carbono (CO<sub>2</sub>) se convierte en una necesidad crítica, especialmente en entornos educativos y laborales.

Motivado por esta problemática, en este trabajo se plantea el desarrollo de una plataforma de monitorización de la calidad del aire adaptada a un edificio universitario, concretamente a la Universidad Complutense de Madrid. El objetivo es no solo visualizar en tiempo real las condiciones ambientales de distintas salas, sino también analizar el comportamiento histórico de los datos y aplicar modelos de predicción que permitan anticipar situaciones de riesgo durante los periodos de mayor ocupación, como los horarios lectivos.

Finalmente, este proyecto nace con la intención de ir más allá de un ejercicio académico, proponiendo una solución flexible y escalable que pueda ser reproducida y ampliada en otros edificios y contextos. De este modo, se busca sentar las bases de un sistema capaz de contribuir a una gestión más eficiente de los espacios interiores y a la mejora de las condiciones ambientales en recintos con alta afluencia de personas.

## 1.2 Objetivos

El objetivo general de este Trabajo Fin de Máster es **diseñar, desarrollar e implementar una plataforma IoT para la monitorización y predicción de la calidad del aire en un edificio universitario, utilizando sensores inalámbricos basados en tecnología Sigfox y técnicas de análisis de datos, con el fin de anticipar situaciones de riesgo para los ocupantes y facilitar la toma de decisiones en entornos interiores.**

A partir de este objetivo general, se definen los siguientes objetivos específicos:

1. **Evaluar y seleccionar las tecnologías de comunicación y sensorización** idóneas para una red IoT de bajo consumo en edificios, con especial foco en las redes LPWAN (específicamente Sigfox) frente a alternativas como WiFi, justificando su elección en base a los requisitos del caso de uso.
2. **Diseñar e implementar una arquitectura IoT integral y abierta**, basada en Node-RED, que permita la adquisición, procesamiento, almacenamiento en base de datos de series temporales (InfluxDB) y visualización en tiempo real de los datos procedentes del backend de Sigfox, prescindiendo de plataformas propietarias.
3. **Realizar un análisis exploratorio (EDA) de los datos históricos** para caracterizar el comportamiento de las variables ambientales ( $\text{CO}_2$ , temperatura, humedad) en función del tipo de sala (aula, oficina, sala técnica) y de los patrones temporales de ocupación, como base para el diseño de modelos predictivos.
4. **Desarrollar, entrenar e integrar en la plataforma modelos predictivos** basados en técnicas de Machine Learning (Random Forest), capaces de estimar la evolución futura de las variables ambientales, incorporando características de contexto como el día de la semana y la hora.
5. **Validar experimentalmente el rendimiento** de la plataforma y de los modelos predictivos mediante la comparación sistemática entre predicciones y valores reales, analizando su precisión, limitaciones y viabilidad operativa para la gestión preventiva de la calidad del aire interior.

### 1.3 Estructura de la memoria

Esta memoria se estructura en seis capítulos fundamentales, diseñados para guiar al lector de manera lógica desde la introducción y el marco teórico hasta la implementación práctica, la validación y las conclusiones del proyecto:

- **Capítulo 1: Introducción.** Presenta el contexto general del trabajo, partiendo de la motivación que impulsa el desarrollo de plataformas IoT para la gestión de la calidad del aire en edificios inteligentes. Define con precisión el objetivo general y los cinco objetivos específicos que persigue este Trabajo Fin de Máster. Finaliza con esta descripción de la estructura del documento.
- **Capítulo 2: Estado de la cuestión.** Establece los fundamentos teóricos y tecnológicos necesarios para contextualizar el proyecto. Analiza en profundidad los factores determinantes de la calidad del aire interior (CO<sub>2</sub>, temperatura y humedad), revisa el marco normativo de referencia (NTP 742 y NTP 549) y realiza un estudio comparativo de las tecnologías de comunicación LPWAN, justificando la selección de Sigfox. Además, introduce los métodos de tratamiento de datos empleados (clustering) y revisa el estado del arte en plataformas IoT y predicción de la calidad del aire, posicionando la contribución de este trabajo.
- **Capítulo 3: Diseño de la solución.** Describe la arquitectura integral y los componentes del sistema IoT desarrollado, organizado en un modelo de cuatro capas: Percepción (sensores Sigfox Airwits CO2 Plus), Comunicación (red e integración Sigfox), Plataforma (orquestración con Node-RED y almacenamiento en InfluxDB) y Aplicación (dashboard de visualización). Este capítulo detalla la selección y justificación de cada herramienta hardware y software que conforma la columna vertebral del sistema.
- **Capítulo 4: Análisis exploratorio de datos (EDA).** Presenta un examen exhaustivo de los datos históricos recopilados por los sensores en las tres salas monitorizadas (aula, oficina, sala de servidores). Mediante estadística descriptiva, análisis de correlación y técnicas de agrupamiento no supervisado (K-Means), se caracterizan los patrones de comportamiento de las variables ambientales, se identifican los factores críticos en cada entorno y se extraen conclusiones que fundamentan directamente el diseño de los modelos predictivos.
- **Capítulo 5: Implementación y validación del sistema.** Detalla la puesta en marcha operativa de la solución. Explica el proceso de diseño, entrenamiento

e integración en Node-RED de los modelos predictivos de Machine Learning (Random Forest). Describe la implementación de los flujos de adquisición de datos, almacenamiento en base de datos, visualización en tiempo real y el módulo de predicción. El capítulo culmina con la validación experimental del sistema, evaluando cuantitativamente la precisión de las predicciones mediante la comparación sistemática con valores reales.

- **Capítulo 6: Conclusiones y trabajo futuro.** Sintetiza los principales hallazgos, logros y contribuciones del proyecto, evaluando el grado de cumplimiento de los objetivos planteados. Discute las limitaciones identificadas durante el desarrollo y la validación. Finalmente, propone una serie de líneas de trabajo futuro claras y viables para la mejora, escalabilidad e integración avanzada de la plataforma, como la implementación de un pipeline de reentrenamiento automático de modelos (MLOps) o la conexión con sistemas de gestión de edificios (BMS).

# Capítulo 2 - Estado de la cuestión

## 2.1 Factores determinantes en la calidad del aire interior (CAI)

La calidad del aire interior (CAI) es un parámetro multifactorial que afecta directamente a la salud, el confort y el rendimiento cognitivo de los ocupantes de un edificio. Se evalúa a través de la concentración de diversos contaminantes y de condiciones ambientales físicas. En este proyecto, la monitorización se centra en tres indicadores fundamentales y ampliamente reconocidos: la concentración de dióxido de carbono ( $\text{CO}_2$ ), la humedad relativa y la temperatura ambiente. Estos parámetros, de bajo coste de medida, proporcionan una evaluación robusta e indirecta del estado general del ambiente interior (Galobart, 2020).

### 2.1.1 Dióxido de Carbono ( $\text{CO}_2$ ) como Indicador de Ventilación

El dióxido de carbono ( $\text{CO}_2$ ) es un gas incoloro e inodoro resultado de la respiración humana y de procesos de combustión. Aunque no es tóxico en bajas concentraciones, su medición se utiliza universalmente como un **indicador proxy de la tasa de ventilación y de la bioefluencia humana** en espacios cerrados (NTP 549, 2023).

En exteriores, la concentración base ronda las 400 ppm. En interiores, los niveles aumentan proporcionalmente al número de ocupantes y al tiempo de ocupación si la renovación de aire es insuficiente. La normativa técnica española NTP 742 (Calleja, 2006) establece rangos de referencia para la calidad del aire en función del  $\text{CO}_2$ :

- **IDA 1 ( $\geq 400$  ppm):** Calidad Buena o Alta.
- **IDA 2 (400 – 600 ppm):** Calidad Media.
- **IDA 3 (600 – 1000 ppm):** Calidad Moderada.
- **IDA 4 ( $> 1000$  ppm):** Calidad Baja o Pobre.

Concentraciones superiores a 1000 ppm se asocian a una percepción de aire viciado, somnolencia, disminución de la concentración y cefaleas, especialmente en exposiciones prolongadas propias de entornos educativos o laborales (NTP 549, 2023).

Por tanto, el CO<sub>2</sub> es la variable crítica para evaluar la adecuación de la ventilación en espacios ocupados.

### **2.1.2 Humedad Relativa del Aire**

La humedad relativa (HR), expresada en porcentaje (%), mide la cantidad de vapor de agua presente en el aire en relación con la cantidad máxima que podría contener a una temperatura dada. Es un factor clave para el **confort térmico y la salubridad ambiental**.

Un nivel óptimo de HR se sitúa entre el **30% y el 60%** (ASHRAE Standard 55). Valores por debajo de este rango provocan sequedad en las mucosas respiratorias y cutáneas, aumentando la susceptibilidad a irritaciones. Por el contrario, una HR persistentemente superior al 60% crea un ambiente propicio para la proliferación de ácaros del polvo, esporas de moho y otros alérgenos biológicos, degradando significativamente la CAI y pudiendo desencadenar problemas respiratorios como el asma. A diferencia del CO<sub>2</sub>, la humedad no guarda una correlación directa con la ocupación, pero su control es esencial para el bienestar global y la preservación del propio edificio.

### **2.1.3 Temperatura Ambiente**

La temperatura del aire es el parámetro ambiental más perceptible directamente y un componente fundamental del **confort térmico**, definido como "la condición mental que expresa satisfacción con el entorno térmico" (ISO 7730).

Un rango de temperatura considerado de confort para actividades sedentarias en interiores, como el estudio o el trabajo de oficina, oscila generalmente entre **20°C y 24°C**. Sin embargo, este rango puede verse influido por factores como la humedad relativa, la velocidad del aire y la vestimenta.

La temperatura no es solo una cuestión de confort. En contextos específicos, como el de una sala de servidores, se convierte en un **parámetro crítico de seguridad operacional**. Las temperaturas elevadas sostenidas pueden provocar el sobrecalentamiento del hardware, reduciendo su vida útil y aumentando el riesgo de fallos catastróficos. Asimismo, en cualquier espacio, un pico térmico abrupto e

inesperado puede ser el primer indicio de un incendio o de una grave disfunción del sistema de climatización.

#### **2.1.4 Interrelación de las Variables**

Es crucial entender que estos tres factores no actúan de forma aislada. Existe una **interdependencia física** entre ellos:

- La **temperatura** afecta a la capacidad del aire para retener humedad (aire más caliente puede contener más vapor de agua).
- La **humedad relativa** influye en la percepción subjetiva de la temperatura (un ambiente húmedo a 24°C puede percibirse como más cálido que uno seco a la misma temperatura).
- Mientras que el **CO<sub>2</sub>** es principalmente un indicador de ventilación, su dinámica puede verse indirectamente afectada por estrategias de control climático (ej., apertura de ventanas para refrescar, que también ventila).

Por lo tanto, una monitorización integral de la CAI y del confort ambiental requiere la consideración conjunta de estas tres variables, tal como se implementa en la plataforma desarrollada en este trabajo.

## **2.2 Normativas y regulaciones**

El diseño e implementación de un sistema de monitorización de la calidad del aire interior (CAI) debe basarse en criterios técnicos y normativos objetivos que permitan interpretar los datos, establecer umbrales de alerta y evaluar el cumplimiento de condiciones salubres. En el ámbito español, las Notas Técnicas de Prevención (NTP) publicadas por el Instituto Nacional de Seguridad y Salud en el Trabajo (INSST) constituyen la referencia fundamental. Este proyecto se alinea principalmente con dos de ellas, que proporcionan el marco para la evaluación de la ventilación y del dióxido de carbono.

### 2.2.1 NTP 742: Ventilación General de Edificios

La NTP 742 (Calleja, 2006) establece los criterios básicos para la ventilación general en edificios, con el objetivo de **proteger la salud de los ocupantes y garantizar unas condiciones ambientales que no generen discomfort**. Esta norma es de aplicación directa en entornos laborales y, por extensión, en espacios de uso público como los universitarios.

La norma distingue entre el aire exterior (designado como **ODA - Outdoor Air**) y el aire interior (**IDA - Indoor Air**). Para el aire interior, clasifica su calidad en cuatro categorías (IDA 1 a IDA 4) en función de la concentración de dióxido de carbono (CO<sub>2</sub>), como se sintetiza en la Tabla 1. Esta clasificación proporciona una escala cuantitativa y objetiva para transformar una lectura de ppm en un indicador de calidad inmediatamente comprensible, que ha sido adoptada como base para la visualización y las alertas en la plataforma desarrollada.

Categoría	Nivel de CO <sub>2</sub> (ppm)	Calidad Determinada
<b>IDA 1</b>	≥ 400	Buena o Alta
<b>IDA 2</b>	400 – 600	Media
<b>IDA 3</b>	600 – 1000	Moderada
<b>IDA 4</b>	> 1000	Baja o Pobre

Tabla 1. Categorías de calidad del aire interior según concentración de CO<sub>2</sub> (NTP 742).

Fuente: Adaptado de Calleja, A. H. (2006). NTP 742: Ventilación general de edificios. INSST.

### 2.2.2 NTP 549: El CO<sub>2</sub> en la Evaluación de la Calidad del Aire Interior

La NTP 549 (INSST, 2023) profundiza en el estudio del **CO<sub>2</sub> como contaminante indicador**, analizando sus fuentes, dinámica en interiores y efectos sobre la salud y el

confort. Esta norma complementa a la NTP 742 al proporcionar una justificación técnica detallada.

Un aporte clave de esta norma es la distinción entre los **límites de exposición profesional** y los **niveles recomendados para el confort**. Establece un Valor Límite Ambiental (VLA-ED) para exposición diaria (8 horas) de 5.000 ppm, y para exposiciones cortas (15 minutos) de 15.000 ppm. Sin embargo, estos son límites de seguridad laboral muy por encima de lo tolerable para el confort.

Para el contexto de oficinas, aulas y edificios de uso público, la NTP 549 señala que **concentraciones a partir de 600 – 800 ppm** pueden empezar a ser percibidas como aire viciado, y que, para **evitar molestias y pérdidas de concentración, es recomendable no superar las 1.000 ppm**. Este umbral refuerza la categoría IDA 4 de la NTP 742 y justifica técnicamente la activación de alertas preventivas en el sistema desarrollado, incluso muy por debajo de los límites de exposición peligrosa.

### **2.2.3 Implicaciones para el Proyecto**

La adopción de este marco normativo tiene implicaciones directas en el diseño de la solución:

1. **Definición de Umbrales:** Las categorías de la NTP 742 proporcionan los **valores de referencia objetivos** para programar los niveles de alerta (advertencia, crítica) en el dashboard de Node-RED.
2. **Validación del Enfoque:** La NTP 549 valida el uso del **CO<sub>2</sub> como métrica principal** para evaluar la adecuación de la ventilación en espacios ocupados, alineándose con el objetivo del proyecto.
3. **Base para la Comunicación:** Esta normativa ofrece un **lenguaje técnico común y estandarizado** (IDA 1, IDA 2, etc.) para informar a los gestores del edificio o a los ocupantes sobre el estado de la CAI, mejorando la utilidad práctica de la plataforma.

Por lo tanto, estas normas no solo constituyen el estado de la técnica, sino que actúan como el **fundamento regulatorio y técnico sobre el que se construye la lógica de monitorización, evaluación y alerta de la plataforma IoT desarrollada**.

## 2.3 Tecnologías de Comunicación LPWAN para Monitorización de Edificios

La capa de comunicación es un pilar crítico en cualquier despliegue IoT a gran escala, especialmente en entornos de edificios inteligentes donde los sensores pueden estar distribuidos en múltiples plantas y espacios interiores. La elección de la tecnología inalámbrica adecuada determina la viabilidad a largo plazo del sistema en términos de autonomía energética, cobertura, escalabilidad y coste operativo. Para aplicaciones de monitorización ambiental continua con sensores de bajo dato y frecuencia de envío esporádica, las redes de Área Amplia y Bajo Consumo (Low Power Wide Area Network, LPWAN) emergen como la familia tecnológica idónea. Esta sección analiza en detalle la tecnología Sigfox, seleccionada para este proyecto, y la contrasta con las principales alternativas del mercado, justificando la decisión en base a los requisitos específicos del caso de uso.

### 2.3.1 Sigfox: Una Red LPWAN Optimizada para Sensores

Sigfox opera como un proveedor global de red de comunicaciones IoT, diseñado específicamente para dispositivos que requieren una conectividad fiable durante años con un consumo energético mínimo. Es una infraestructura de red propietaria presente en más de 70 países, donde en la península Ibérica el operador es UnaBiz. Su modelo de negocio se basa en una suscripción anual por dispositivo, que mantiene un coste operativo bajo y predecible (aproximadamente 1 USD por dispositivo al año), mientras que los dispositivos en sí son de libre desarrollo al estar basados en patente abierta.

Desde un punto de vista técnico, Sigfox es una red de topología estrella que utiliza bandas de frecuencia sin licencia (868 MHz en Europa, 902 MHz en América). Emplea una señal de banda ultra estrecha (Ultra Narrowband, UNB) que le confiere una alta capacidad de penetración en interiores, un factor clave para aplicaciones de domótica y sensorización de edificios. Su arquitectura opera principalmente en tres capas del modelo OSI (Ilustraciones 2-1, 2-2, 2-3): la capa física (modulación en banda ISM y control de errores CRC), la capa de enlace (definición de trama y autenticación) y la capa de aplicación (contenido del mensaje).

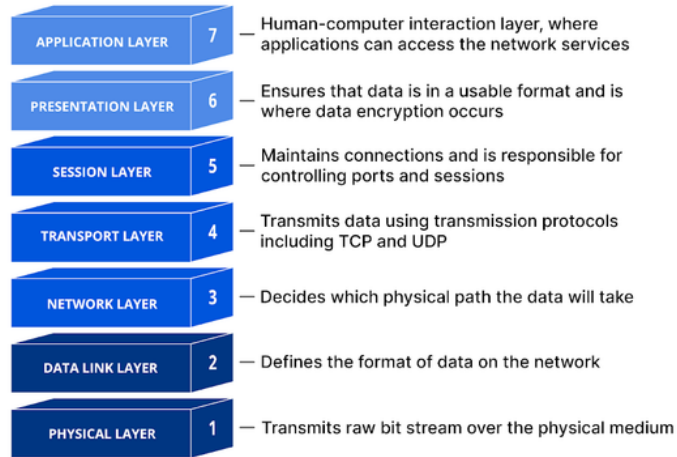


Ilustración 2-1. Capas del Modelo OSI.

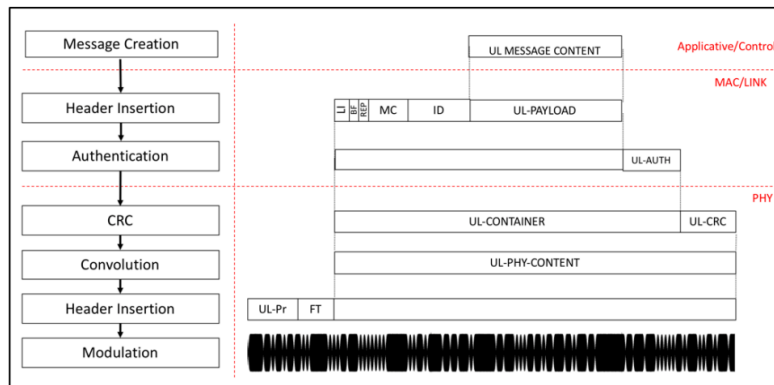


Ilustración 2-2. Capas de modelo OSI para mensaje UPLINK en Sigfox.

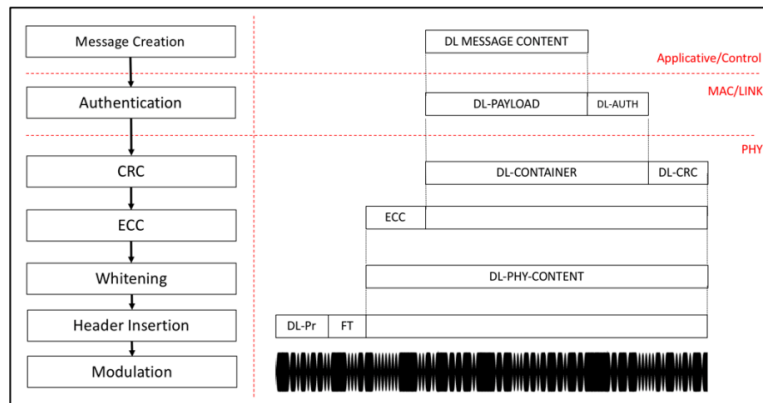


Ilustración 2-3. Capas de modelo OSI para mensaje DOWNLINK en Sigfox.

Specifications	SIGFOX
Frequency Band	868 MHz in Europe and 915 MHz in USA
Bandwidth	200 KHz
distance	30 to 50 Km in rural and 3 to 10 Km in urban areas.
data rate	From 10 to 1000 bps (100 bps is ideal for IoT applications)
message size	For 12 bytes of data to be transported sigfox frame uses 26 bytes in total. This includes overhead.
modulation scheme	BPSK i.e. Binary Phase Shift Keying
applications	M2M and IoT (Internet of Things) based applications to provide wireless connectivity to electric meters, smart watches, washing machines etc. Mostly low energy devices/equipments transmitting/receiving few bytes are covered using Sigfox technology.
Capacity	A Sigfox receiver can handle 30,00000 devices or objects.
power consumption	About 61mA (at the time of transmission)
architecture	star consisting of sigfox objects, sigfox base stations, sigfox cloud, customer IT server

*Ilustración 2-4. Especificaciones de Sigfox.*

Las especificaciones clave de Sigfox se resumen en la Ilustración 2-4. Su diseño prioriza la eficiencia energética mediante limitaciones inteligentes en la comunicación. El modo principal es el mensaje **Uplink** (dispositivo a servidor), con un límite de hasta 140 mensajes por día y un tamaño máximo de payload de 12 bytes, suficiente para encapsular múltiples lecturas de sensores (como temperatura, humedad y CO<sub>2</sub>). El mensaje **Downlink** (servidor a dispositivo) solo está disponible como respuesta a un Uplink y está limitado a 4 mensajes diarios de 8 bytes, lo que la hace menos adecuada para aplicaciones de control en tiempo real pero perfecta para la configuración ocasional o la solicitud de datos. La latencia típica de un mensaje Uplink es de unos segundos, pudiendo extenderse a ~30 segundos si se utiliza un payload personalizado, como es el caso de este proyecto.

La gestión de los dispositivos y mensajes se realiza desde el backend de Sigfox (<https://backend.sigfox.com>), donde es posible configurar callbacks personalizados (webhooks) para redirigir cada mensaje recibido a un servidor propio. Este mecanismo, fundamental para la integración en plataformas IoT personalizadas, permite extraer los datos en formato JSON (Ilustración 2-5) con campos como el identificador del dispositivo (device), la marca temporal (time) y el payload hexadecimal (data), tal como se detalla en la Tabla 2.

<b>Variable</b>	<b>Type</b>	<b>Format</b>	<b>Description</b>
<i>time</i>	integer	Epoch time in seconds	The event timestamp (in seconds since the Unix Epoch)
<i>deviceTypeId</i>	string	in hexadecimal, up to 8 characters	Device Type identifier (in hexadecimal – up to 8 characters <=> 4 bytes)
<i>device</i>	string	in hexadecimal, up to 8 characters	Device identifier (in hexadecimal – up to 8 characters <=> 4 bytes)
<i>rssI</i>	float	two maximum fraction digits	The RSSI (in dBm – Float value with two maximum fraction digits). If there is no data to be returned, then the value is null.
			<b>Condition: for devices with contract option</b>
			<b>NETWORK METADATA</b>
<i>station</i>	string	in hexadecimal, 4 characters	The base station identifier (in hexadecimal – 4 characters <=> 2 bytes)
			<b>Condition: for devices with contract option</b>
			<b>NETWORK METADATA</b>
<i>data</i>	string	hexadecimal	The device data payload
<i>seqNumber</i>	integer		The sequence number of the message if available

Tabla 2. Tipos de variables configurables para mensaje Uplink.

```
{
  "deviceTypeId": "{deviceTypeId}",
  "deviceId": "{device}", "data": "{data}" }
```

Ilustración 2-5. Ejemplo de mensaje Uplink.

En definitiva, Sigfox se presenta como una tecnología LPWAN madura y especializada, cuyo diseño —centrado en la simplicidad, el bajo consumo y la cobertura extensa— la convierte en un candidato de referencia para aplicaciones de monitorización ambiental en edificios.

### 2.3.2 Tecnologías Alternativas: LoRaWAN, NB-IoT, LTE-M y WiFi

Para contextualizar y justificar la selección de Sigfox, es necesario revisar las principales tecnologías alternativas disponibles para la sensorización de edificios, cada una con sus fortalezas y nichos de aplicación.

**LoRaWAN** es otra tecnología LPWAN de código abierto que, a diferencia del modelo de operador único de Sigfox, permite el despliegue de redes privadas. Funciona en la banda de 867-869 MHz y se define como un protocolo de capa de red. Su principal ventaja es una **comunicación bidireccional más flexible y en tiempo real**, lo que la hace ideal para aplicaciones que requieren control de actuadores. Sin embargo, esta flexibilidad conlleva una mayor complejidad en la planificación de la red y un consumo energético ligeramente superior cuando se utiliza la bidireccionalidad de forma activa.

**NB-IoT (Narrow Band IoT)** y **LTE-M** son estándares LPWAN desarrollados por el consorcio 3GPP y operados sobre infraestructuras de redes celulares móviles. NB-IoT destaca por su **excelente penetración en interiores y sótanos**, gracias a su uso de bandas licenciadas de 700, 800 y 900 MHz, y ofrece comunicaciones cifradas. LTE-M, por su parte, proporciona un **mayor ancho de banda y menor latencia**, siendo adecuada para aplicaciones con cierta movilidad. Ambas tecnologías implican un **coste operativo mensual por dispositivo** y un consumo energético generalmente superior al de Sigfox o LoRaWAN, al depender de la infraestructura celular.

**WiFi**, basado en el estándar IEEE 802.11, es ubicuo en entornos domésticos y corporativos. Opera en las bandas de 2.4 GHz y 5.8 GHz, ofreciendo un **ancho de banda muy alto y una latencia mínima**, ideal para la transmisión de grandes volúmenes de datos. Sin embargo, para aplicaciones IoT de sensorización masiva presenta desventajas decisivas: un **consumo energético muy elevado** que impide la operación con baterías de larga duración, una **pobre escalabilidad** debido a la saturación rápida de los puntos de acceso, y una **cobertura en interiores que se degrada significativamente** al atravesar múltiples muros, a diferencia de las tecnologías sub-GHz.

### 2.3.3 Análisis Comparativo y Justificación para Monitorización de Edificios

La Tabla 3 presenta una comparativa sintética de las tecnologías analizadas, evaluándolas en función de los criterios más relevantes para el despliegue de una red de sensores ambientales en un edificio: eficiencia energética, cobertura en interiores, coste operativo, adecuación al caso de uso de monitorización y capacidad para integración sencilla.

	<b>SIGFOX</b>	<b>LORA</b>	<b>NB-IOT</b>	<b>LTE-M</b>	<b>WIFI</b>
<b>LICENCIADO</b>	No	No	Sí	Sí	Sí
<b>FRECUENCIA EUROPEA</b>	868/902 MHz (ISM)	867 - 869 MHz (ISM)	700/800/900 MHz (bandas 3,8	700/800/900/1800 MHz	2.4/5.8 GHz
<b>MODULACIÓN</b>	BPSK / GFSK	SS Chirp	UNB/BPSK/GFSK	OFDMA	DSSS y CKK / OFDM
<b>ANCHO DE BANDA</b>	192 kHz	500 kHz	200kHz	1.4 MHz	20 -140 MHz
<b>RATIO DE TRANSMISIÓN</b>	100bps	50kbps	20 kbps uplink / 60 kbpps downlink	1 Mbps	150 Mbps
<b>ALCANCE URBANO</b>	3-10km	2-5km	1km	1km	15-50m
<b>ALCANCE RURAL</b>	30-50km	15-20km	10km	10km	< 300m
<b>TAMAÑO DE PAQUETE</b>	12 bytes	11-242 bytes (Definido por usuario)	1280 bytes	1280 bytes	2304 bytes
<b>TOPOLOGÍA</b>	Estrella	Estrella	Estrella	Estrella	Estrella
<b>COMUNICACIÓN</b>	Bidireccional Limitada / Half-duplex	Bidireccional / Half-duplex	Bidireccional Half-Duplex	Bidireccional Duplex	Full-Bidireccional
<b>LÍMITE DE MENSAJES</b>	Uplink:140/día, Downlink: 4/día	llimitado	llimitado	llimitado	llimitado
<b>ENCRIPCIÓN</b>	No	No	Sí	Sí	Sí
<b>AUTENTICACIÓN</b>	Sí	No	Sí	Sí	Sí
<b>CONSUMO DE ENERGÍA</b>	Bajo	Muy bajo	Bajo	Medio	Alto
<b>MOVILIDAD / LOCALIZACIÓN</b>	Sí	Sí	No	Sí	No

Tabla 3. Comparativa de tecnologías IoT inalámbricas.

A partir de este análisis, la justificación para la selección de Sigfox en este proyecto se consolida en los siguientes puntos:

1. **Eficiencia Energética y Bajo Mantenimiento:** El consumo "Muy Bajo" de Sigfox, unido a la larga vida útil de la batería de los sensores Airwits CO<sub>2</sub> Plus (10 años), garantiza un despliegue con mantenimiento prácticamente nulo, un requisito fundamental para una red escalable en un edificio universitario.
2. **Cobertura Robusta en Interiores:** Su operación en 868 MHz con modulación UNB le proporciona una "Excelente" capacidad para penetrar muros y llegar a sensores en distintas ubicaciones dentro del edificio, superando claramente a WiFi y equiparándose a NB-IoT.
3. **Modelo de Coste Predecible y Escalable:** El coste operativo fijo y muy bajo por dispositivo anual permite presupuestar y escalar la red de sensores sin sorpresas, a diferencia de los modelos de cuota mensual de las tecnologías celulares (NB-IoT, LTE-M).
4. **Adecuación al Patrón de Comunicación:** Los límites de mensajes (140 Uplink/día) y el pequeño tamaño de payload (12 bytes) de Sigfox están **perfectamente alineados** con las necesidades de los sensores ambientales, que envían lecturas esporádicas de unos pocos bytes. La bidireccionalidad limitada no es un impedimento, sino una caracterización coherente con una aplicación de pura monitorización.
5. **Simplicidad de Integración:** El mecanismo de callbacks del backend de Sigfox permite una integración directa y sencilla con plataformas personalizadas como Node-RED, otorgando control total sobre el flujo de datos sin depender de soluciones propietarias cerradas.

En conclusión, mientras que **LoRaWAN** es una alternativa muy válida para redes privadas o que requieran mayor control bidireccional, y las tecnologías celulares (**NB-IoT, LTE-M**) son ideales para aplicaciones móviles o críticas, **Sigfox ofrece el equilibrio óptimo para el caso de uso específico de este proyecto**. Su combinación de eficiencia energética extrema, cobertura interior fiable, coste operativo mínimo y simplicidad la

convierten en la tecnología LPWAN más idónea para la monitorización ambiental continua y escalable en edificios inteligentes. Esta justificación fundamenta la arquitectura de comunicación adoptada y descrita en el Capítulo 3.

## 2.4 Métodos de tratamiento de datos

La fase de análisis y preprocesamiento de datos es fundamental para extraer conocimiento de los conjuntos de datos crudos procedentes de los sensores IoT. En este proyecto, se ha adoptado un enfoque basado en técnicas de **aprendizaje no supervisado**, concretamente en algoritmos de **clustering** (agrupamiento), con el objetivo principal de descubrir patrones, estructuras ocultas y comportamientos naturales en las series temporales ambientales sin utilizar etiquetas previas. Este análisis exploratorio permite segmentar los datos en grupos homogéneos que facilitan su interpretación, la identificación de estados operativos y la preparación para etapas posteriores de modelado predictivo.

### 2.4.1 Clustering: Agrupamiento No Supervisado

El clustering es una familia de técnicas cuyo objetivo es **particionar un conjunto de datos en grupos (clústeres)**, de tal forma que los elementos dentro de un mismo clúster sean muy similares entre sí (alta cohesión interna) y, a la vez, muy distintos a los elementos de otros clústeres (alta separación externa). Existen dos grandes categorías de algoritmos de clustering:

1. **Clustering Jerárquico:** Construye una estructura de árbol (dendrograma) que representa las relaciones de similitud entre todos los puntos, permitiendo visualizar los agrupamientos a diferentes niveles de granularidad.
2. **Clustering Particional:** Asigna cada punto del dataset a un único clúster, optimizando un criterio global. El algoritmo más representativo de este grupo es **K-Means**, que ha sido el seleccionado para el análisis en este trabajo debido a su eficiencia computacional, escalabilidad y facilidad de interpretación.

## 2.4.2 Algoritmo K-Means y el Método del Codo

**K-Means** es un algoritmo iterativo que busca minimizar la suma de las distancias cuadradas (inercia) entre cada punto y el centroide (media) de su clúster asignado. Su principal parámetro es  $k$ , el número de clústeres a formar, que debe ser definido *a priori*.

Para determinar el valor óptimo de  $k$ , se emplea una técnica heurística ampliamente utilizada: **el Método del Codo (Elbow Method)** (Rodríguez, 2023). Esta metodología consiste en:

1. Ejecutar el algoritmo K-Means para un rango de valores de  $k$  (por ejemplo, de 1 a 10).
2. Para cada  $k$ , calcular la **inercia total** (suma de las distancias al cuadrado intra-clúster).
3. Graficar la inercia en función de  $k$ .

El **valor óptimo de  $k$**  se identifica visualmente como el **punto de inflexión o "codo"** de la curva, donde la ganancia en la reducción de la inercia (compactación de los clústeres) comienza a disminuir notablemente al añadir un clúster más. Este punto representa un buen equilibrio entre la compacidad de los clústeres y la complejidad del modelo.

## 2.4.3 Métricas de Validación de la Calidad del Clustering

Dado que el clustering es no supervisado, es esencial evaluar la calidad de la partición resultante mediante métricas internas que no requieran etiquetas de verdad absoluta. En este proyecto se han utilizado dos de las más representativas:

1. **Índice de la Silueta (Silhouette Score):** Este índice, propuesto por Rousseeuw (1987), mide cuán similar es un objeto a su propio clúster en comparación con otros clústeres. Para cada punto  $i$ , se calcula:
  - a.  $a(i)$ : Distancia promedio de  $i$  a todos los demás puntos en el mismo clúster (cohesión).
  - b.  $b(i)$ : Distancia promedio de  $i$  a todos los puntos en el clúster más cercano al que no pertenece (separación).

El coeficiente de silueta para el punto  $i$  es:  $s(i) = (b(i) - a(i)) / \max(a(i), b(i))$ . El **índice global** es el promedio de  $s(i)$  para todos los puntos. Su valor oscila entre **-1 y 1**:

- c. **Valores cercanos a +1**: Indican que el punto está bien asignado, con alta cohesión y buena separación.
  - d. **Valores cercanos a 0**: Sugieren que el punto se encuentra en la frontera entre dos clústeres.
  - e. **Valores cercanos a -1**: Implican que el punto probablemente estaría mejor asignado a un clúster vecino.
2. **Índice de Davies-Bouldin (DBI)**: Este índice (Davies & Bouldin, 1979) evalúa la calidad del clustering calculando el ratio entre la dispersión intra-clúster y la separación inter-clúster. Se define como:

$$DB = \frac{1}{k} \sum_{i=1}^k \max_{j \neq i} \left( \frac{s_i + s_j}{d(c_i, c_j)} \right)$$

donde  $s_i$  es la dispersión promedio dentro del clúster  $i$ , y  $d(c_i + c_j)$  es la distancia entre los centroides de los clústeres  $i$  y  $j$ . **Un valor de DBI más bajo (idealmente cercano a 0)** indica una mejor partición, con clústeres compactos y bien separados entre sí.

#### 2.4.4 Aplicación en el Proyecto

En el Capítulo 4 de este trabajo, estas técnicas se aplican sistemáticamente a los datasets de cada sensor. El proceso seguido es:

1. **Preprocesamiento**: Limpieza de datos y normalización.
2. **Determinación de k**: Aplicación del Método del Codo para seleccionar el número de clústeres más representativo para cada sala.
3. **Ejecución de K-Means**: Obtención de la partición de datos.
4. **Validación**: Cálculo del **Índice de Silueta** y del **Índice de Davies-Bouldin** para evaluar objetivamente la calidad del agrupamiento obtenido.

5. **Análisis e Interpretación:** Los clústeres resultantes se analizan en función de las variables ambientales (CO<sub>2</sub> vs. temperatura, CO<sub>2</sub> vs. humedad) y del factor tiempo (día de la semana), permitiendo identificar patrones de ocupación, estados de calidad del aire y comportamientos estacionales. Este análisis exploratorio es crucial para comprender la dinámica de cada espacio y sirve como fundamento para el diseño de los modelos predictivos supervisados desarrollados posteriormente.

## 2.5 Revisión del estado del arte

El desarrollo de este Trabajo Fin de Máster se fundamenta y se posiciona dentro de un ámbito de investigación activo que converge en la intersección de las tecnologías IoT para edificios inteligentes, las redes de comunicación LPWAN y la aplicación de técnicas de Machine Learning (ML) para el análisis ambiental. Esta sección revisa el estado del arte, estructurándolo en tres ejes temáticos clave y analizando cómo los trabajos previos informan las decisiones técnicas y metodológicas adoptadas en este proyecto.

### 2.5.1 Comparativa de Tecnologías de Comunicación LPWAN

La elección de la capa de comunicación es crítica en cualquier despliegue IoT a gran escala. El estudio de **Levchenko y Bobrov (2022)**, "Performance Comparison of NB-Fi, Sigfox, and LoRaWAN", proporciona un análisis empírico fundamental. Los autores evalúan estos tres protocolos LPWAN en distintos escenarios, midiendo métricas como la tasa de pérdida de paquetes, el rango y la eficiencia energética bajo diferentes modos de operación (con y sin acknowledgment).

- **Hallazgos Relevantes:** Su trabajo concluye que **Sigfox presenta un rendimiento óptimo en distancias cortas y en modo de envío sin confirmación (unacknowledged)**, siendo extremadamente eficiente en energía. Por el contrario, LoRaWAN es más adecuado para mensajes más largos o cuando se requiere comunicación bidireccional más frecuente.

- **Influencia en este Proyecto:** Este análisis **valida y justifica técnicamente la selección de Sigfox** para nuestro caso de uso: monitorización de sensores ambientales en un edificio con mensajes pequeños (12 bytes), baja frecuencia de envío y donde la prioridad es la máxima vida útil de la batería (años) sobre la comunicación bidireccional en tiempo real. Nuestro proyecto adopta, por tanto, el perfil operativo donde Sigfox demuestra ser más competitivo.

### 2.5.2 Plataformas IoT para Monitorización de Edificios

La arquitectura de la plataforma de integración y visualización es otro pilar del proyecto. El trabajo de **Gómez D'Orazio et al. (2022)**, "Integración de una red de sensores con una plataforma IoT para control inteligente de aulas", aborda un problema similar en un contexto escolar. Su solución se basa en sensores de CO<sub>2</sub> conectados a un dispositivo de borde (Raspberry Pi) que utiliza los protocolos MQTT/CoAP para enviar datos a la plataforma ThingsBoard.

- **Hallazgos Relevantes:** Este trabajo se centra en la **arquitectura de integración, la evaluación de protocolos de aplicación (MQTT vs. CoAP)** y la fiabilidad de la transmisión, analizando la tasa de mensajes recibidos frente a los enviados.
- **Influencia y Diferenciación de este Proyecto:** Si bien compartimos el objetivo de monitorizar la CAI en un entorno educativo, nuestro proyecto introduce diferencias significativas:
  - **Capa de Comunicación:** Nosotros utilizamos **Sigfox (LPWAN)** frente a WiFi/Ethernet, lo que elimina la necesidad de un gateway local y simplifica el despliegue físico.
  - **Arquitectura de Plataforma:** En lugar de una plataforma pre-empaquetada como ThingsBoard, desarrollamos una **plataforma personalizada y ligera basada en Node-RED**, lo que ofrece una flexibilidad total para la integración de modelos de ML personalizados y un control absoluto sobre el flujo de datos, desde el backend de Sigfox hasta la base de datos.

- **Avance:** Nuestro trabajo **avanza más allá de la simple visualización**, integrando directamente **modelos predictivos** en el dashboard operativo.

### **2.5.3 Machine Learning para la Predicción de la Calidad del Aire**

La aplicación de ML para anticipar la evolución de los parámetros ambientales es el tercer pilar innovador. La revisión sistemática de **Iskandaryan et al. (2020)**, "Air Quality Prediction in Smart Cities Using Machine Learning Technologies Based on Sensor Data", ofrece una panorámica invaluable. Los autores categorizan y comparan los métodos más usados (Redes Neuronales, Regresión, Métodos Ensemble, Híbridos), destacando que algoritmos basados en **Random Forest** suelen ofrecer un buen equilibrio entre precisión, velocidad de entrenamiento y robustez frente a outliers.

- **Hallazgos Relevantes:** El estudio identifica a **Random Forest como un método particularmente eficaz** para problemas de regresión con datos de sensores, debido a su capacidad para manejar relaciones no lineales y su menor propensión al sobreajuste en comparación con redes neuronales profundas cuando el volumen de datos es limitado.
- **Influencia en este Proyecto:** Esta revisión **dirigió nuestra selección del algoritmo de modelado**. Inspirados por sus conclusiones, optamos por implementar y evaluar modelos basados en **Árboles de Decisión y Random Forest** para la predicción de CO<sub>2</sub>, temperatura y humedad. Además, el proyecto de **Gracia Cancho (2025)** sobre predicción del Índice de Calidad del Aire utilizando redes neuronales (LSTM) nos alertó sobre la importancia de una **limpieza y preprocesamiento meticuloso de los datos** para evitar que los modelos aprendan patrones espurios, una práctica que hemos adoptado rigurosamente.

### **2.5.4 Conclusión del Estado del Arte y Posicionamiento de este Trabajo**

La revisión del estado del arte revela una clara trayectoria hacia soluciones IoT más integradas e inteligentes para la gestión de edificios. Sin embargo, se

identifica un **espacio de oportunidad** en la **combinación específica de los tres ejes anteriores** en un único sistema cohesivo y de bajo coste operativo.

Este Trabajo Fin de Máster **se posiciona como una contribución integradora** que:

1. Aprovecha las ventajas de **Sigfox (LPWAN)** para un despliegue de sensores escalable y de muy bajo mantenimiento.
2. Propone una **arquitectura de plataforma IoT abierta y flexible basada en Node-RED**, que permite una integración directa y personalizada desde la red de comunicación hasta la visualización.
3. Incorpora **modelos predictivos de Machine Learning (Random Forest)** dentro del flujo operativo de la plataforma, permitiendo no solo monitorizar, sino **anticipar** la calidad del aire en función del contexto (tipo de sala, día, hora).

Por lo tanto, este proyecto no solo aplica tecnologías existentes, sino que **sintetiza y avanza** el estado del arte al construir un puente práctico entre la monitorización LPWAN, las plataformas de integración de código abierto y la analítica predictiva accionable para la domótica de edificios.

## Capítulo 3 - Diseño de la solución

Este capítulo detalla la arquitectura integral y las herramientas tecnológicas específicas que constituyen la columna vertebral del sistema desarrollado. La solución propuesta sigue una **arquitectura en capas** que abarca desde la adquisición de datos en el borde físico (*edge*) hasta su procesamiento, almacenamiento, análisis predictivo y visualización final. Se describirán tanto los componentes hardware (sensores) como software (plataformas, bases de datos, herramientas de desarrollo), explicando su función y justificación dentro del ecosistema IoT implementado.

### 3.1 Descripción de instalaciones físicas

Para contextualizar correctamente el análisis de datos y la implementación de los modelos predictivos que se presentan en este trabajo, es fundamental describir con precisión las características físicas y los patrones de uso de cada sala donde se desplegaron los sensores. La idiosincrasia de cada espacio —determinada por su aforo, actividad principal y condiciones ambientales— define requisitos específicos de monitorización y, en consecuencia, modelos de análisis diferenciados. A continuación, se detallan las tres ubicaciones seleccionadas dentro del campus de la Universidad Complutense de Madrid (UCM).

#### 3.1.1 Aula de Clases B3 (Sensor ID: 1FC823F)

Este sensor se encuentra desplegado en un **laboratorio docente** de la Facultad de Informática, ubicado en la última planta del edificio. La sala está destinada a las clases del Máster en Internet de las Cosas, con una capacidad aproximada para 25 estudiantes y un profesor.

- **Características Físicas:** Se trata de un espacio amplio con iluminación natural y ventilación mecánica. Presenta una **importante variabilidad térmica estacional**, con temperaturas notablemente bajas en invierno.
- **Patrón de Ocupación:** La ocupación es **intensa y periódica**. Las clases se imparten principalmente en horario de tarde, de lunes a jueves, lo que genera picos predecibles de concentración humana.

- **Variable Crítica y Objetivo de Monitorización:** La **concentración de dióxido de carbono (CO<sub>2</sub>)** es el parámetro de interés principal. El objetivo es monitorizar la calidad del aire durante las clases para garantizar condiciones salubres y detectar situaciones de ventilación insuficiente, con especial atención a los incrementos posteriores a las 16:00 horas en días lectivos.

### 3.1.2 Despacho de Profesores (Sensor ID: C36596)

El sensor está ubicado en una oficina compartida por profesores en la Facultad de Físicas de la UCM.

- **Características Físicas y Ocupación:** A diferencia del aula, se trata de un espacio de **menor dimensión y con una ocupación más reducida, pero constante**. El flujo de personas es estable durante la jornada laboral administrativa.
- **Patrón de Emisiones:** Los niveles de CO<sub>2</sub> no experimentan picos tan drásticos como en el aula, pero presentan una **acumulación progresiva y sostenida** a lo largo del día.
- **Objetivo de Monitorización:** En este caso, el interés se extiende al **conjunto de las tres variables ambientales** (CO<sub>2</sub>, temperatura y humedad), con el fin de mantener unas condiciones de confort térmico y calidad del aire óptimas para el desarrollo de actividad intelectual prolongada.

### 3.1.3 Sala de Servidores (Sensor ID: C4D841)

Este sensor monitoriza el ambiente de una sala técnica de servidores de la UCM.

- **Características Físicas:** Es un **entorno crítico y climatizado**, diseñado para mantener condiciones estables que alarguen la vida útil y garanticen el correcto funcionamiento del hardware. La temperatura y la humedad están controladas artificialmente, por lo que sus variaciones naturales son mínimas.
- **Patrón de Ocupación:** La presencia humana es **esporádica o nula**, limitada a tareas de mantenimiento. Por lo tanto, los niveles de CO<sub>2</sub> se mantienen constantes y bajos.

- **Variable Crítica y Objetivo de Monitorización:** La **temperatura** es el parámetro de seguridad primordial. Cualquier pico o anomalía térmica no prevista puede ser el primer indicio de un fallo en los sistemas de refrigeración, un principio de incendio o una avería que ponga en riesgo los equipos. La plataforma debe, por tanto, estar configurada para generar alertas tempranas basadas en esta variable.

Esta heterogeneidad en los casos de uso justifica plenamente el enfoque de **análisis y modelado diferenciado por sala** que se adopta en el presente trabajo. Mientras que para el aula y la oficina el modelo predictivo busca anticipar la degradación de la calidad del aire, para la sala de servidores su función es la de un sistema de alerta temprana para la integridad del equipamiento.

### 3.2 Arquitectura General del Sistema

La arquitectura del sistema se ha diseñado siguiendo un modelo de **cuatro capas principales**: Percepción, Comunicación, Plataforma/Procesamiento y Aplicación. Esta división modular garantiza la escalabilidad, el mantenimiento sencillo y la clara separación de responsabilidades. La Figura 3-1 ilustra este diseño y el flujo de datos entre sus componentes.

- **1. Capa de Percepción (Edge):** Está compuesta por los **sensores Sigfox Airwits CO2 Plus** desplegados en las distintas salas de la UCM. Estos dispositivos son los responsables de la **adquisición física de los datos ambientales** (temperatura, humedad, CO<sub>2</sub>) y de su encapsulación en un mensaje digital.
- **2. Capa de Comunicación (LPWAN):** Los mensajes de los sensores son transmitidos de forma inalámbrica utilizando la **red Sigfox**. Esta red LPWAN actúa como el conducto de largo alcance y bajo consumo que transporta los datos desde los dispositivos hasta la infraestructura en la nube del operador (backend de Sigfox/UNABIZ).
- **3. Capa de Plataforma y Procesamiento:** Es el núcleo de la solución IoT personalizada. Aquí reside el servidor central que:

- **Extrae los datos** del backend de Sigfox mediante un *webhook* (callback) configurado.
- **Procesa y decodifica** los *payloads* hexadecimales utilizando **Node-RED**, que orquesta todos los flujos de trabajo.
- **Almacena** los datos históricos de forma eficiente en una base de datos de series temporales, **InfluxDB**.
- **Ejecuta los modelos predictivos** de Machine Learning (desarrollados en **Jupyter Notebook** y operacionalizados como scripts de Python) cuando se solicita una predicción desde la interfaz de usuario.
- **4. Capa de Aplicación y Visualización:** Proporciona la interfaz de usuario final. Se implementa mediante el módulo **Dashboard de Node-RED**, que ofrece paneles interactivos para:
  - La **monitorización en tiempo real** de las lecturas de todos los sensores.
  - La **visualización de tendencias históricas** mediante gráficos.
  - La **interacción con el sistema predictivo**, permitiendo al usuario solicitar predicciones para horas y variables futuras.
- **Conexión Segura (Ngrok):** Para permitir que el backend de Sigfox envíe datos a nuestro servidor Node-RED (que en desarrollo puede ejecutarse en una máquina local detrás de un router), se utiliza **Ngrok**. Esta herramienta crea un **túnel seguro con una URL pública** que redirige el tráfico a la instancia local de Node-RED, simulando un servidor con IP pública sin necesidad de un despliegue complejo en la nube.

En las siguientes secciones se describirá en detalle cada uno de estos componentes tecnológicos.

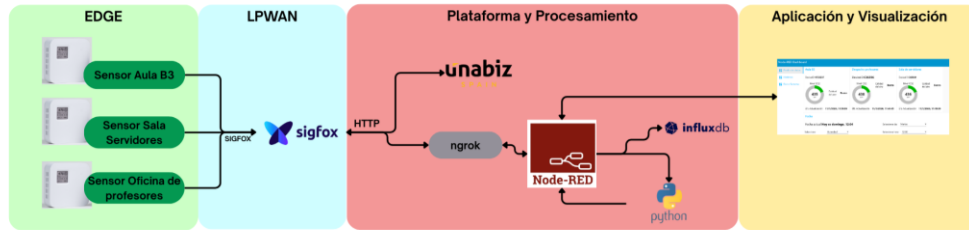


Ilustración 3-1. Arquitectura en capas del sistema IoT para monitorización y predicción de la calidad del aire. El flujo de datos comienza en los sensores (Percepción), es transmitido vía Sigfox (Comunicación), procesado y almacenado por Node-RED e InfluxDB (Plataforma), y finalmente visualizado y consultado a través del Dashboard (Aplicación).

### 3.3 Capa de Percepción: Sensores Sigfox Airwits CO2 Plus

La capa de percepción está constituida por los dispositivos desplegados en las salas de la UCM. Para este proyecto se seleccionaron los sensores **Sigfox Airwits CO2 Plus** de Connected Inventions, diseñados específicamente para monitorización ambiental en interiores. Su idoneidad se fundamenta en varias características clave: integran medición de temperatura, humedad y CO<sub>2</sub> en un único dispositivo; incluyen una pantalla a color que muestra el estado de la calidad del aire localmente; y, lo más importante, cuentan con conectividad nativa Sigfox y una batería con una vida útil estimada de 10 años (16.000 mAh), lo que los hace ideales para despliegues permanentes de muy bajo mantenimiento.

- Las principales especificaciones técnicas del sensor se resumen a continuación:
- Vida útil de la batería: 10 años.
- Rango de medición de CO<sub>2</sub>: 370 – 5000 ppm.
- Rango de operación: Temperatura de 10 a 50°C, Humedad de 0 a 85%.
- Resolución: 1 ppm para CO<sub>2</sub>, 0.1°C para temperatura.
- Frecuencia de envío ajustable: de 10 minutos a 24 horas.
- Conectividad: Sigfox (RCZ1 / RCZ2 / RCZ4 / RCZ7).
- El sensor encapsula cada lectura en un **payload hexadecimal de 12 bytes** que es transmitido como mensaje Uplink. La estructura de este payload, esencial para su

posterior decodificación en la plataforma, se detalla en la Ilustración 3-3. Según el fabricante, los primeros bytes corresponden a un header, seguidos de los valores de temperatura (en complemento a dos), humedad y concentración de CO<sub>2</sub>.



*Ilustración 3-2. Sensor AiWits CO2 Plus.*

De acuerdo con la documentación oficial de Sigfox (Device messages delay), lo común es que el retraso entre el envío y la recepción de los mensajes suele ser de pocos segundos, sin embargo, si se configura en la comunicación un 'payload' personalizado o avanzado como mencionan en la documentación, esto hace que la latencia sea de aproximadamente 30 segundos. Ya que se construye el mensaje JSON con las variables que el usuario seleccione.

Esta capa, por tanto, tiene como salida un flujo de mensajes periódicos con un payload crudo que contiene la información ambiental de cada sala monitorizada.

En esta tabla se indica como obtener la temperatura, humedad y dióxido de carbono del mensaje que envía el sensor, donde los bytes uno y dos son para la temperatura, el tercer byte está reservado para la humedad y los bytes cuatro y cinco para CO<sub>2</sub>. Dentro de la callback del backend de Sigfox se configura esto para enviar el payload y en el cliente, en este caso el servidor de Node-RED, una función para decodificarla (véase Sección 3.5.2) y utilizar las variables legibles para esta aplicación.

Measurement message header byte				
BYTE	BIT	TYPE	NAME	DESC
0	0-2		reserved	not in use
0	3	UInt1	message type	always 0 for measurement message
0	4-7	UInt4	firmware version	0 = special, 1 - 15 = production

Measurement message					
BYTE	BIT	TYPE	NAME	RANGE	DESC
0	0-7	UInt8	<Measurement message header>		
1-2	0-15	Int16	temperature	0x8000, 954D..0x7FFF	0x8000= error, else T (n/10) = -40.0°C..120.0°C
3	0-7	UInt8	humidity	0xFF, 0x00..0x64	0xFF=error, else H(n)= 0%..100%
4-5	0-15	UInt16	CO2	0x00, 0x186..0x1388	0x00=error, else CO2(n) = 390pm...5000ppm
6-7	0-15	UInt16	CO2min (optional)	0x00, 0x186..0x1388	0x00=error, else CO2(n) = 390pm...5000ppm (sent only if configured in downlink)
8-9	0-15	UInt16	CO2max (optional)	0x00, 0x186..0x1388	0x00=error, else CO2(n) = 390pm...5000ppm (sent only if configured in downlink)

Ilustración 3-3. Tabla de payload del sensor.

### 3.4 Capa de Comunicación: Integración con la Red Sigfox

Como se justificó en la Sección 2.3, la red Sigfox fue seleccionada como la tecnología LPWAN óptima para este despliegue. La implementación práctica de esta capa consistió en integrar los sensores en la red del operador (UnaBiz) y configurar un mecanismo para recuperar automáticamente los datos en nuestra plataforma.

Cada sensor fue registrado en el backend de Sigfox (<https://backend.sigfox.com>), asociándolo a una cuenta de usuario. El componente crítico de la integración fue la configuración de un **callback (webhook) personalizado** para el tipo de dispositivo AirwitsCO2 Plus. Este callback se programó para que, por cada mensaje Uplink recibido por la red Sigfox, su backend realice una petición HTTP POST a un endpoint específico de nuestro servidor. La petición incluye un objeto JSON con los metadatos del mensaje, cuyo formato se muestra a continuación:

```
{
  "device": "{deviceId}",
  "time": {timestamp},
  "data": "{payloadHexadecimal}"
}
```

Durante la fase de desarrollo, para que el backend de Sigfox pudiera alcanzar la instancia local de Node-RED ejecutándose tras un router doméstico, se empleó el servicio **Ngrok**. Ngrok proporciona un túnel seguro que asigna una URL pública única (por ejemplo, <https://abc123.ngrok.io>) y redirige todo el tráfico entrante a un puerto

local, simulando de forma efectiva un servidor con IP pública sin necesidad de un despliegue complejo en la nube (Ilustración 3-4).

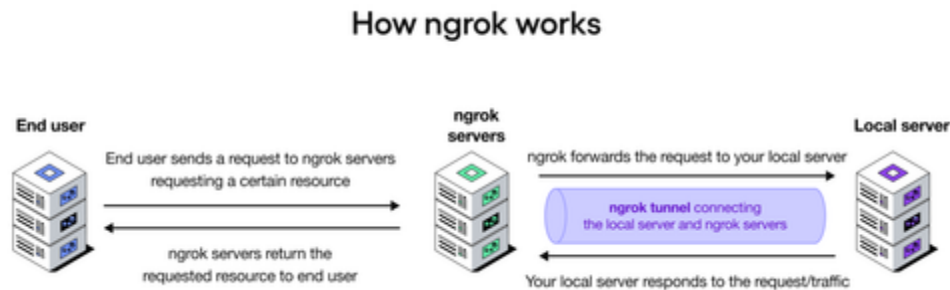


Ilustración 3-4. Diagrama de funcionamiento de Ngrok.

De este modo, la capa de comunicación actúa como un conducto fiable y transparente: recibe los payloads crudos de los sensores y los reenvía, convenientemente empaquetados en JSON con su contexto (ID del dispositivo y marca temporal), al núcleo de procesamiento de la plataforma.

### 3.5 Capa de Plataforma: Procesamiento, Almacenamiento y Lógica de Negocio

Esta capa es el cerebro del sistema, donde se orquestan la recepción, transformación, persistencia y análisis inteligente de los datos. Se construyó utilizando un conjunto de herramientas de código abierto que priorizan la flexibilidad y el control integral del flujo de datos.

#### 3.5.1 Orquestación con Node-RED

Se seleccionó Node-RED, una plataforma de programación visual basada en flujos (flows), como el orquestador central. Su principal ventaja reside en la flexibilidad para crear integraciones personalizadas y ligeras mediante la conexión de nodos, evitando la rigidez y potencial sobrecoste de plataformas IoT pre-empaquetadas. Node-RED

permite gestionar desde la recepción HTTP hasta la ejecución de scripts externos de manera visual e intuitiva.

### 3.5.2 Recepción y Decodificación de Datos

En Node-RED se implementó un flujo iniciado por un nodo http in que escucha en el endpoint configurado para el callback de Sigfox. Cada petición POST entrante es procesada inmediatamente por una **función de decodificación JavaScript personalizada**. Esta función realiza las operaciones críticas para transformar el payload hexadecimal en información estructurada:

1. Parsea el JSON de entrada.
2. Extrae la cadena hexadecimal del campo data.
3. Decodifica los bytes según el formato especificado por el fabricante (Ilustración 3-3): los primeros 2 caracteres para el header, los siguientes 4 para la temperatura, los siguientes 2 para la humedad y los últimos 4 para el CO<sub>2</sub>.
4. Aplica las conversiones y escalados necesarios (por ejemplo, dividir el valor bruto de la temperatura por 10).
5. Devuelve un nuevo objeto JSON estructurado con los campos: device\_id, timestamp, temperature, humidity, co2.

### 3.5.3 Almacenamiento en Base de Datos de Series Temporales: InfluxDB

Para persistir el histórico de mediciones de forma eficiente y alimentar futuros análisis, los datos decodificados se almacenan en InfluxDB. La elección de esta base de datos frente a alternativas relacionales como SQLite se justifica por su diseño optimizado para datos de series temporales, como se muestra en la Tabla 4.

Característica	InfluxDB	SQLite
Tipo de BD	Series temporales	Relacional (SQL)
Modelo de datos	Medidas, etiquetas, tiempo, valores	Tablas, filas, columnas
Escalabilidad	Alta, ideal para producción (Enterprise)	Baja (proyectos pequeños)

<b>Ideal para</b>	Datos de sensores IoT, manejar Smart cities	Datos estructurados tradicionales
-------------------	---	-----------------------------------

Tabla 4. Comparación de bases de datos.

Se configuró una base de datos llamada `calidad_aire` y se utilizó el nodo `node-red-contrib-influxdb` para establecer la conexión. Cada mensaje decodificado se escribe como un nuevo *point* en InfluxDB, utilizando el `device_id` como *tag* para filtrado y los valores ambientales como *fields*.

### 3.5.4 Integración de Modelos de Machine Learning

La capacidad predictiva del sistema se materializa mediante la integración de los modelos de Machine Learning entrenados (detallados en el Capítulo 5). La operacionalización de estos modelos se diseñó como un mecanismo de puente externo. Cuando un usuario solicita una predicción desde el dashboard, un flujo específico en Node-RED:

1. Construye un archivo `input.json` con todos los parámetros necesarios (ID del sensor, variable a predecir, marca de tiempo futura, configuración de InfluxDB).
2. Utiliza un nodo `exec` para invocar un script de Python externo (`predict_co2.py` o `predict_multivar.py`), pasándole el archivo de entrada.
3. Captura la salida JSON del script, la parsea y envía los resultados (valor predicho y categoría de calidad del aire) de vuelta al dashboard para su visualización.

Esta arquitectura desacoplada permite actualizar o reentrenar los modelos sin modificar la lógica central de Node-RED.

### 3.6 Capa de Aplicación: Dashboard de Visualización y Control

La capa de aplicación proporciona la interfaz de usuario final, implementada mediante el módulo `node-red-dashboard`. Se diseñó un panel de control unificado y responsive organizado en varias pestañas o grupos para una experiencia de usuario clara:

- **Monitorización en Tiempo Real:** Esta vista presenta para cada sensor un resumen visual inmediato. Muestra los valores numéricos actuales de temperatura, humedad y CO<sub>2</sub>, una representación gráfica (gage) de cada variable y un **indicador de estado de calidad del aire** basado en los umbrales de la norma NTP 742 (Buena, Media, Moderada, Baja).
- **Gráficas Históricas:** Un módulo dedicado permite visualizar la evolución temporal de las variables. El usuario puede seleccionar el sensor deseado y un rango de tiempo predefinido (últimas 24 horas, última semana, último mes). Las gráficas se generan dinámicamente consultando los datos almacenados en InfluxDB.
- **Módulo de Predicciones:** Es la interfaz central para interactuar con los modelos predictivos. Permite al usuario: (1) Seleccionar el sensor de interés, (2) Elegir la variable ambiental a predecir (CO<sub>2</sub>, Temperatura o Humedad), (3) Especificar un día y hora futuros, y (4) Ejecutar la predicción mediante un botón. Los resultados (valor numérico predicho y, para el CO<sub>2</sub>, la categoría de calidad estimada) se muestran dinámicamente en la misma interfaz.

### 3.6. Conclusión del Capítulo

En este capítulo se ha presentado la arquitectura integral y la implementación práctica de la plataforma IoT desarrollada para la monitorización predictiva de la calidad del aire. El diseño en capas —Percepción (sensores Sigfox), Comunicación (red y callback), Plataforma (Node-RED, InfluxDB, modelos ML) y Aplicación (dashboard interactivo)— ha demostrado ser robusto, escalable y de bajo coste operativo. Cada componente fue seleccionado y configurado para cumplir con los requisitos específicos del caso de uso en un edificio universitario, priorizando la autonomía energética, la fiabilidad en interiores y la flexibilidad de integración.

Con la plataforma operativa y desplegada, se recopilaban meses de datos ambientales continuos. El siguiente capítulo (Capítulo 4) se dedica al **Análisis Exploratorio de Datos (EDA)** de este histórico, un proceso fundamental para comprender los patrones de comportamiento, validar la calidad de los datos y,

sobre todo, sentar las bases informadas para el diseño y entrenamiento de los modelos predictivos que esta misma arquitectura es capaz de ejecutar.

# Capítulo 4 - Análisis exploratorio de datos (EDA)

## 4.1 Introducción y Metodología del Análisis

Este capítulo presenta el análisis exploratorio de los datos históricos recopilados por los sensores Sigfox desplegados en la Universidad Complutense de Madrid (UCM). El objetivo principal es **caracterizar el comportamiento de las variables ambientales (CO<sub>2</sub>, temperatura y humedad) en cada tipo de sala**, identificar patrones temporales, correlaciones y estructuras subyacentes en los datos. Este proceso es fundamental para comprender la dinámica de cada entorno, validar la calidad de los datos y, sobre todo, para **informar y justificar el diseño de los modelos predictivos** que se desarrollarán.

Se analizan los datos de tres sensores, cada uno representativo de un caso de uso distinto, tal como se describió en la Sección 3.1:

1. **Sensor** 1FC823F: Aula de clases (laboratorio) de la Facultad de Informática.
2. **Sensor** C36596: Despacho de profesores.
3. **Sensor** C4D841: Sala de servidores climatizada.

Para cada sensor, se aplica una **metodología de EDA sistemática** que consta de las siguientes fases:

1. **Descripción y Estadística Descriptiva:** Análisis de la distribución, rangos, valores máximos, mínimos y promedios de cada variable.
2. **Análisis de Correlación:** Cálculo de matrices de correlación para evaluar la relación lineal entre pares de variables (CO<sub>2</sub>, temperatura, humedad).
3. **Análisis de Agrupamiento (Clustering):** Aplicación del algoritmo K-Means para descubrir grupos naturales o "estados" en los datos. Esto incluye:
  - a. Determinación del número óptimo de clústeres (k) mediante el **Método del Codo**.
  - b. Validación de la calidad de la partición mediante los **Índices de Silueta y Davies-Bouldin**.

c. Interpretación de los clústeres resultantes en el contexto del uso de la sala.

4. **Análisis Temporal:** Estudio de los patrones en función del **día de la semana** para evidenciar el impacto de la ocupación (especialmente en el aula).

El análisis se realiza sobre dos conjuntos temporales para el sensor del aula (1FC823F): un **subconjunto de un mes** (como prueba de concepto) y el **conjunto completo de ~10 meses** (para un análisis robusto). Para los sensores de oficina y sala de servidores, el análisis se presenta directamente sobre el período largo de ~10 meses.

## 4.2 Análisis del Sensor de Aula B3 - Facultad de Informática (ID: 1FC823F)

El sensor desplegado en el aula B3 monitoriza un entorno con una ocupación intensa y periódica, definida por el horario académico. Para comprender la evolución de los patrones con el volumen de datos, se realizó un análisis en dos etapas: un **análisis exploratorio inicial con un mes de datos** y un **análisis exhaustivo con el dataset completo de ~10 meses**.

### 4.2.1 Análisis Exploratorio: Dataset de un Mes (febrero 2025)

Se seleccionó un mes representativo (febrero) con 1.213 mensajes para un primer estudio ágil de las distribuciones y relaciones.

- **Distribución y Estadísticas Descriptivas:** La Ilustración 4-1 muestra la distribución de las tres variables ambientales durante este periodo. La Ilustración 4-2 detalla la cantidad de instancias que alcanzaron los valores máximos y mínimos registrados, revelando una concentración de las lecturas en rangos medios-bajos.

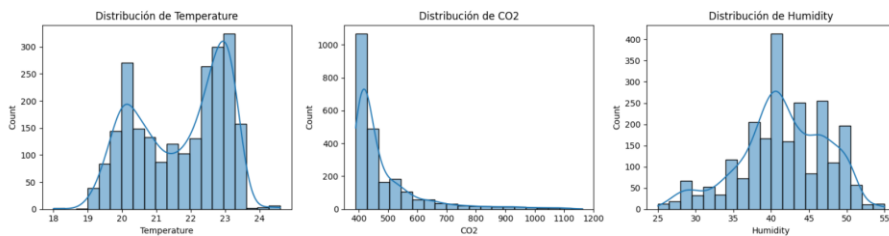


Ilustración 4-1. Distribución de datos de un mes del sensor 1FC823F.

	Action	count	min_CO2	max_CO2	mean_CO2
0	Bad indoor air quality	20	424	463	440.350000
1	Good indoor air quality	175	409	980	598.931429
2	Moderate indoor air quality	61	439	1222	762.245902
3	Poor indoor air quality	957	400	1313	501.482759

Ilustración 4-2. Cantidad de instancias con valores máximos y mínimos.

- Análisis de Correlación Lineal:** La Ilustración 4-3 presenta la matriz de correlación entre temperatura, humedad y CO<sub>2</sub>. Se observa una **correlación muy débil** entre el CO<sub>2</sub> y las otras variables (aprox. 0.15 con humedad y 0.01 con temperatura), confirmando que los factores ambientales puros no son predictores lineales directos del nivel de CO<sub>2</sub> en este corto plazo.

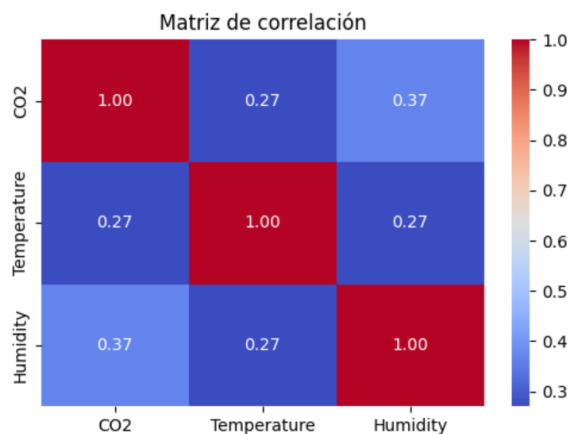


Ilustración 4-3. Matriz de correlación variables del sensor.

- Clustering Basado en Variables Ambientales:** Para descubrir grupos naturales, se aplicó el algoritmo K-Means. La Ilustración 4-4 (Gráfica de método del codo) sugirió k=4 como un punto de inflexión potencial.

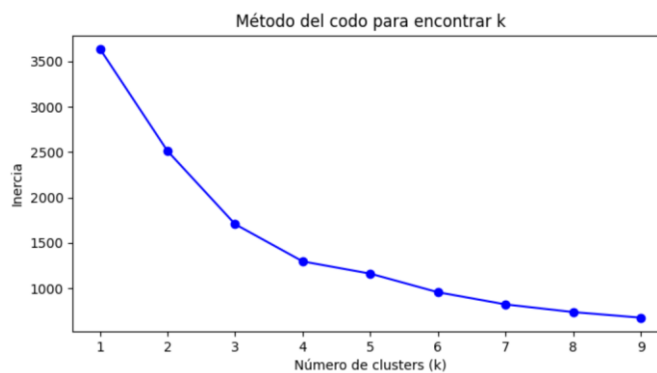


Ilustración 4-4. Gráfica de método del codo para encontrar K.

- o Las Ilustraciones 4-5 y 4-6 muestran los cuatro clústeres resultantes proyectados en los planos CO<sub>2</sub> vs. Temperatura y CO<sub>2</sub> vs. Humedad, respectivamente. Se aprecia una superposición significativa entre grupos.

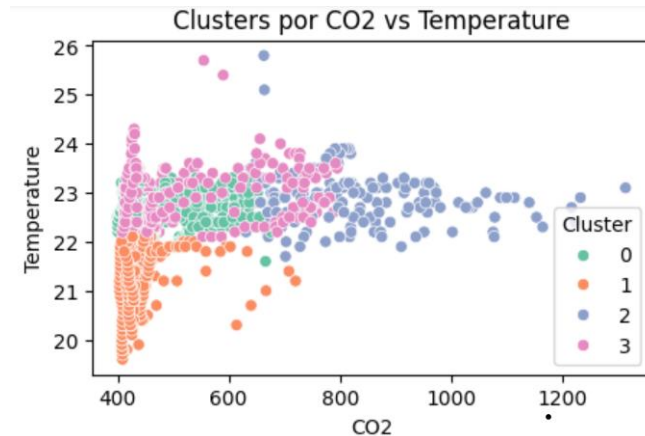


Ilustración 4-5. Gráfica de cuatro clústeres comparando CO<sub>2</sub> vs Temperatura.

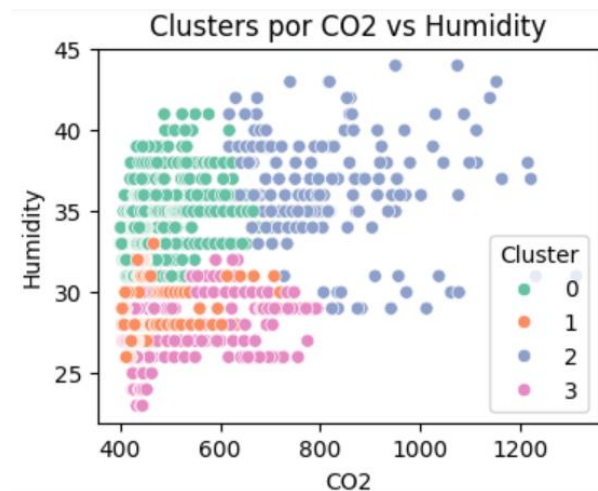


Ilustración 4-6. Gráfica de los clústeres comparando CO<sub>2</sub> vs Humedad.

- o La calidad de esta partición fue evaluada cuantitativamente. La **Ilustración 4-7** reporta métricas pobres: **Índice de Silueta = -0.061** e **Índice de Davies-Bouldin = 2.576**, indicando clústeres mal definidos y solapados. Probar con k=3 (Ilustración 4-8) empeoró aún más las métricas.

Índice de Silueta: -0.061  
Índice de Davies-Bouldin: 2.576

Ilustración 4-7. Métricas de medición de 4 clústeres.

Índice de Silueta: -0.138  
Índice de Davies-Bouldin: 2.767

Ilustración 4-8. Métricas de medición de tres clústeres.

- **Hallazgo Clave: Importancia del Contexto Temporal:** Al incorporar la dimensión del **día de la semana**, el análisis dio un giro. La **Ilustración 4-9** grafica los niveles de CO<sub>2</sub> agrupados por día, mostrando picos marcados de lunes a jueves y valores bajos y estables de viernes a domingo. Al realizar el clustering considerando este patrón, las métricas para k=3 **mejoraron drásticamente: Índice de Silueta = 0.541 e Índice de Davies-Bouldin = 0.627**. Esto demostró de manera concluyente que **el factor determinante en la dinámica del CO<sub>2</sub> es el patrón semanal de ocupación, no las variables ambientales aisladas**.

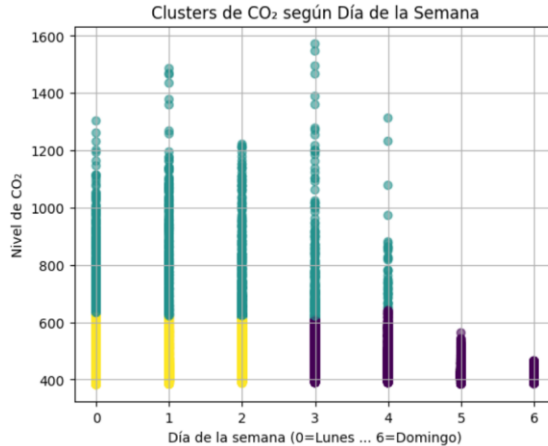


Ilustración 4-9. Gráfica de niveles de CO<sub>2</sub> de acuerdo con días de la semana.

#### 4.2.2 Análisis Exhaustivo: Dataset de Diez Meses (Enero - Octubre 2025)

Para una caracterización robusta, se analizó el dataset completo de 14.729 muestras, abarcando casi un ciclo académico completo.

- **Estadística Descriptiva Ampliada:** La **Ilustración 4-10** muestra la distribución de los datos para el periodo largo. Se observa un **rango mucho más amplio y realista**: CO<sub>2</sub> hasta 1600 ppm, y temperaturas desde menos de 10°C (invierno) hasta 35°C (verano), capturando la variabilidad estacional ausente en el análisis de un mes.

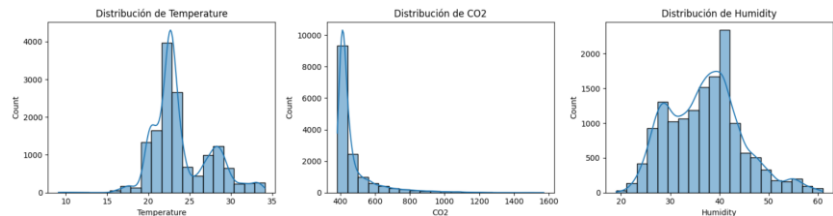


Ilustración 4-10. Distribución de datos de 10 meses sensor IFC823F.

- **Visualización del Patrón Temporal Anual:** La **Ilustración 4-11**, extraída directamente de la plataforma Unabiz, visualiza la serie temporal de CO<sub>2</sub>, donde son evidentes los "bloques" de picos semanales durante el periodo lectivo y la meseta durante los meses de verano (julio, agosto).

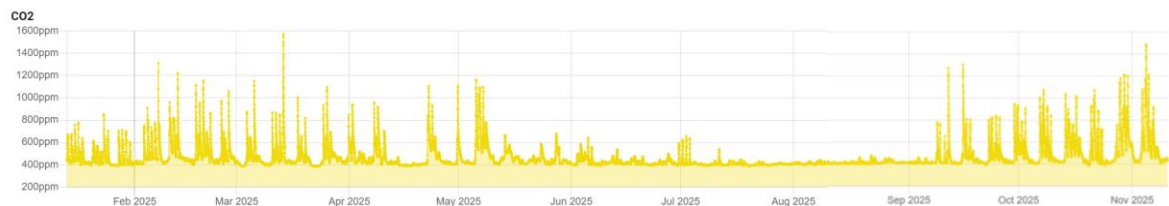


Ilustración 4-11. Gráfica de nivel de CO<sub>2</sub> sensor IFC823F meses ene-nov.

- **Consolidación del Clustering Óptimo:** La **Ilustración 4-12 (Gráfica del codo)** para el dataset extenso confirma k=3 como una elección adecuada. La **Ilustración 4-13** reafirma el patrón semanal de niveles de CO<sub>2</sub>. Las métricas de validación para k=3 se mantuvieron **excelentes y consistentes** con el análisis previo (**Índice de Silueta = 0.541**, **Índice de Davies-Bouldin = 0.627**), probando la robustez y estabilidad de este patrón a lo largo del tiempo.

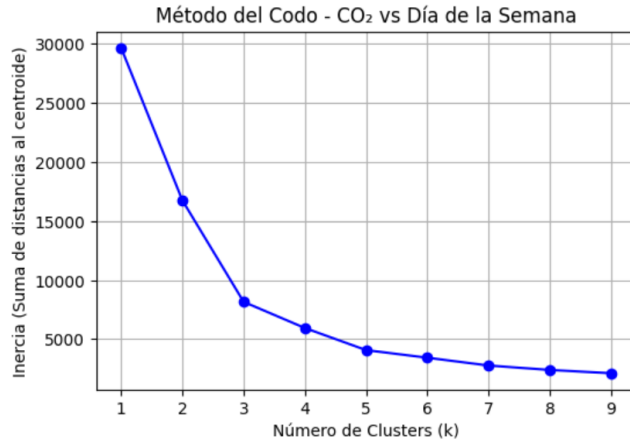


Ilustración 4-12. Gráfica del codo del sensor 1FC823F con 10 meses de datos.

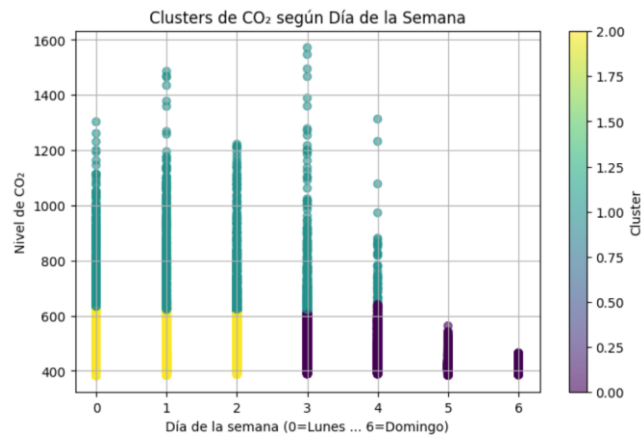


Ilustración 4-13. Gráfica de niveles de CO<sub>2</sub> según el día de la semana.

### 4.2.3 Conclusión del Análisis para el Aula B3

La comparación entre ambos análisis permite establecer conclusiones sólidas para el diseño del sistema:

1. **Un mes de datos es insuficiente** para capturar la verdadera variabilidad (estacional, periodos vacacionales) y puede llevar a una subestimación de los rangos operativos.
2. **La característica de contexto** día de la semana **es crítica** para modelar correctamente el comportamiento del aula. Los modelos predictivos deben incorporarla de forma explícita.

3. **El patrón semanal de ocupación es el driver principal** de la dinámica del CO<sub>2</sub>, lo que genera una señal fuerte y predecible para los algoritmos de Machine Learning.
4. **La alta calidad del clustering con datos temporales** valida la existencia de estados operativos claramente diferenciables (ocupado/no ocupado), lo que fundamenta la viabilidad de un modelo predictivo para este entorno.

Estos hallazgos justifican directamente las decisiones técnicas tomadas en el Capítulo 5: entrenar un **modelo específico para este sensor** utilizando el **dataset extenso** y emplear **características temporales** (weekday, hour, month) como *features* de entrada esenciales junto con las lecturas ambientales en tiempo real.

### 4.3 Análisis del Sensor de Despacho de Profesores (ID: C36596)

Este sensor monitoriza un entorno de oficina en la Facultad de Físicas, caracterizado por una ocupación más constante y menos densa que un aula, pero con una presencia humana sostenida durante la jornada laboral.

- **Distribución y Estadísticas Descriptivas:** La **Ilustración 4-14** muestra la distribución de los datos de ~10 meses (14.554 muestras). En comparación con el aula, se observa un **rango de CO<sub>2</sub> notablemente más estrecho y uniforme**, sin los picos extremos. Las distribuciones de temperatura y humedad son también más concentradas, reflejando un ambiente interior más controlado.

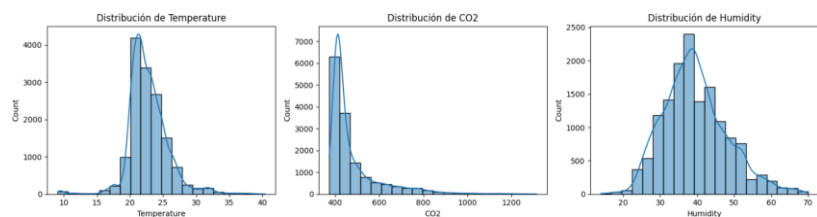


Ilustración 4-14. Distribución de datos de 10 meses sensor C36596.

- **Análisis de Correlación Lineal:** La matriz de correlación presentada en la **Ilustración 4-15** revela un patrón distinto al del aula. La relación entre temperatura y humedad muestra una **correlación negativa más marcada**. La influencia del CO<sub>2</sub> sobre las otras variables sigue siendo muy baja,

confirmando nuevamente su independencia de los factores ambientales directos en este contexto.

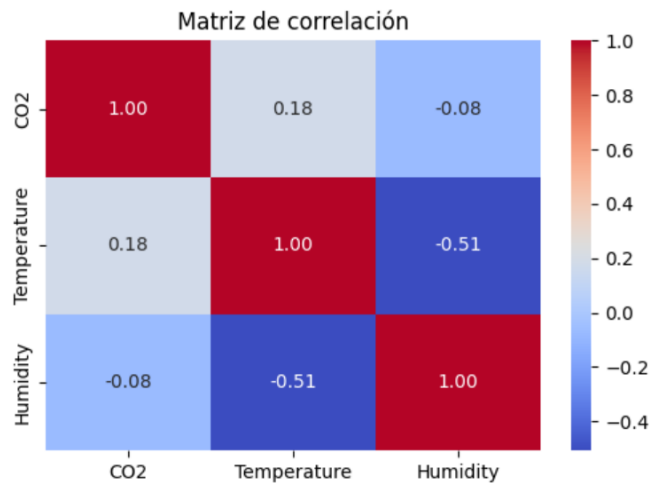


Ilustración 4-15. Matriz de correlación del sensor C36596.

- **Clustering y Patrones:** La aplicación del Método del Codo (**Ilustración 4-16**) sugiere nuevamente  $k=3$  como un número adecuado de clústeres. Los clústeres resultantes, visualizados en la **Ilustración 4-17**, presentan una **dispersión y superposición mayor** que los observados en el aula. Esto es coherente con la naturaleza del espacio: la ocupación es más estable y menos volátil, lo que genera transiciones más suaves entre estados y hace que los grupos basados en datos puramente ambientales y puntuales sean menos distinguibles. Los clústeres parecen diferenciar entre horarios de alta actividad, baja actividad y periodos nocturnos o de fin de semana, pero los límites son difusos.

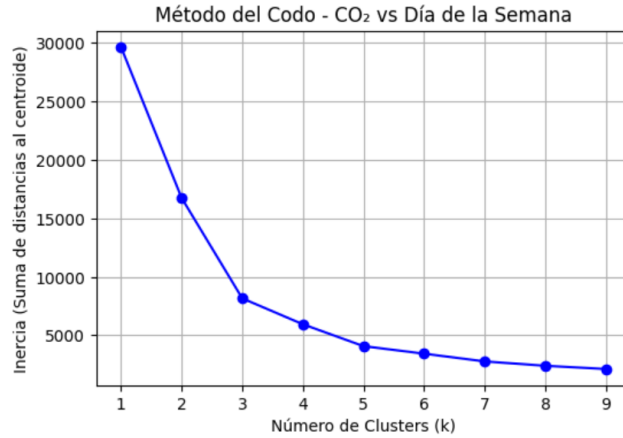


Ilustración 4-16. Gráfica del codo sensor C36596.

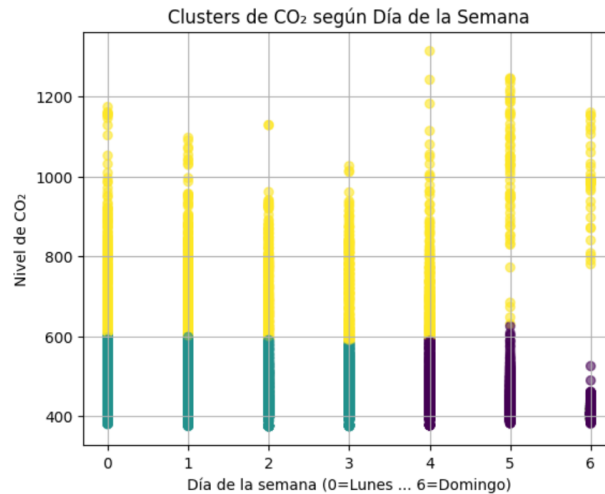


Ilustración 4-17. Clústeres de datos sensor C36596.

**Conclusión para el Despacho:** El comportamiento ambiental de la oficina es **más estable y menos variable** que el del aula. Aunque existe un patrón ligado a la ocupación laboral, este no es tan abrupto ni periódico como el académico. Para la predicción, esto sugiere que un modelo podría beneficiarse de **ventanas de tiempo más largas** para capturar la tendencia, y que las alertas por CO<sub>2</sub> deberían basarse en umbrales sostenidos más que en picos instantáneos.

## 4.4 Análisis del Sensor de Sala de Servidores (ID: C4D841)

Este sensor se ubica en un entorno técnico crítico, climatizado y con una ocupación humana insignificante. Su análisis tiene un objetivo distinto: identificar la normalidad operativa para detectar anomalías.

- **Distribución y Estadísticas Descriptivas:** La **Ilustración 4-18** muestra la distribución de ~10 meses de datos (13.177 muestras). El perfil es radicalmente diferente:
  - **CO<sub>2</sub>:** Concentrado en un rango muy estrecho y bajo (~400-500 ppm), confirmando la ausencia de fuentes de biofluentes.
  - **Temperatura:** Muestra una **distribución extremadamente estable y acotada** (aproximadamente entre 18°C y 25°C), evidencia del control activo del sistema de climatización.
  - **Humedad:** Presenta cierta variabilidad, pero dentro de un rango controlado.

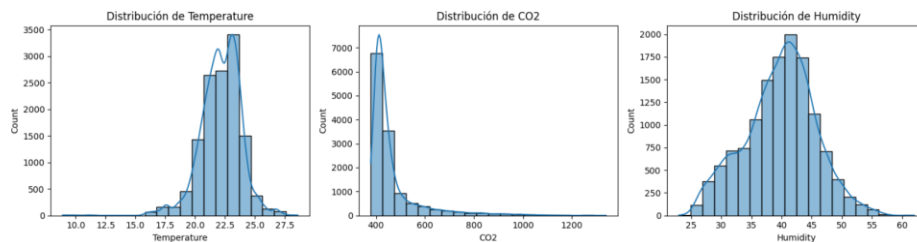


Ilustración 4-18. Distribución de datos de 10 meses sensor C4D841.

- **Análisis de Correlación Lineal:** La matriz de correlación (**Ilustración 4-19**) es la más elocuente. Muestra una **correlación negativa fuerte entre temperatura y humedad**, típica de un sistema de aire acondicionado que deshumidifica al enfriar. El CO<sub>2</sub>, nuevamente, no muestra correlación con las otras variables.

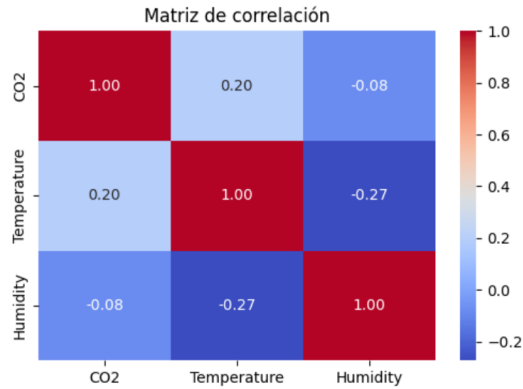


Ilustración 4-19. Matriz de correlación del sensor C4D841.

- Clustering y Patrones:** La gráfica del codo (**Ilustración 4-20**) también sugiere  $k=3$  para este dataset. Los clústeres obtenidos (**Ilustración 4-21**) no se corresponden con patrones de ocupación humana. En su lugar, es probable que diferencien **modos de operación del sistema de climatización** (ej., enfriamiento activo, mantenimiento de consigna, posiblemente un modo de sobrecalentamiento leve). La estrecha agrupación de los datos confirma que el sistema mantiene un **estado basal muy consistente**.

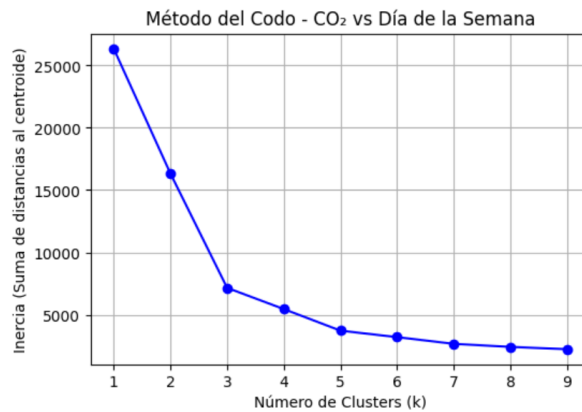


Ilustración 4-20. Gráfica del codo sensor C4D841.

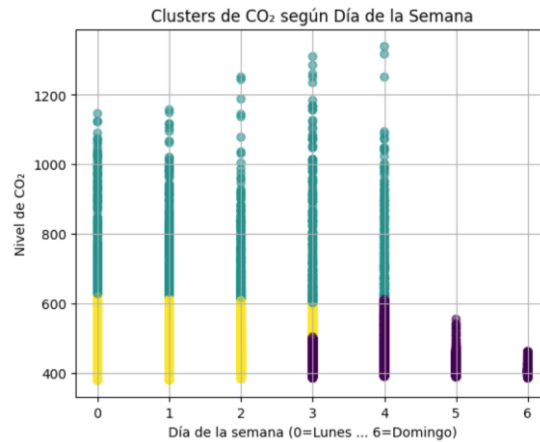


Ilustración 4-21. Clústeres de datos sensor C4D841.

**Conclusión para la Sala de Servidores:** Este entorno opera en un **régimen de control estrecho**. Cualquier dato que se desvíe significativamente de los clústeres identificados (especialmente en temperatura) constituye una **potencial anomalía**. Esto convierte a la temperatura en la **variable crítica para monitorización y alerta temprana**, y justifica un modelo de detección de anomalías o, como se ha implementado, un modelo predictivo que alerte cuando las proyecciones futuras se salgan del rango de normalidad histórica.

#### 4.5 Síntesis Comparativa y Conclusiones Generales del Análisis Exploratorio

El análisis comparativo de los tres entornos permite extraer conclusiones estratégicas para el diseño del sistema global:

1. **El Caso de Uso Define la Variable Crítica y la Estrategia:** Se confirman tres perfiles distintos:
  - a. **Aula (1FC823F):** CO<sub>2</sub> como indicador de **calidad del aire y ventilación**. Patrón **fuertemente periódico (semanal)**. Estrategia: **Predicción basada en contexto temporal**.

- b. **Oficina (C36596):** CO<sub>2</sub> como indicador de **confort y renovación de aire**. Patrón **continuo y estable**. Estrategia: **Monitorización de tendencias y umbrales sostenidos**.
  - c. **Sala de Servidores (C4D841):** **Temperatura** como indicador de **salud operativa y seguridad**. Patrón **de control estrecho**. Estrategia: **Detección de anomalías/desviaciones**.
2. **La Necesidad de Modelos Específicos o Contextualizados:** Las marcadas diferencias en los patrones de datos justifican el desarrollo de **modelos predictivos dedicados para cada tipo de sala**, o, en su defecto, la inclusión obligatoria de una **característica categórica que identifique el tipo de espacio** en un modelo unificado.
  3. **El Contexto Temporal es un Feature Esencial:** El análisis del aula demostró de manera irrefutable que variables como el día de la semana(*weekday*) y la hora son más determinantes para predecir el CO<sub>2</sub> que la temperatura o humedad actuales. Esta lección debe extrapolarse: los modelos deben **ingerir e interpretar el contexto** para ser precisos.
  4. **Calidad y Volumen de Datos:** El dataset de ~10 meses demostró ser suficiente para capturar ciclos estacionales y patrones operativos completos, proporcionando una base sólida para el entrenamiento de modelos supervisados en el siguiente capítulo.

Estas conclusiones no solo resumen el trabajo analítico, sino que **constituyen los requisitos de diseño** para la fase de implementación de la plataforma predictiva que se detalla en el Capítulo 5.

# Capítulo 5 - Implementación y validación del sistema

Este capítulo detalla el proceso de desarrollo integral del sistema, desde la concepción y entrenamiento de los modelos predictivos hasta su implementación operativa en la plataforma IoT y su posterior validación experimental. El objetivo es demostrar cómo los hallazgos del análisis exploratorio (Capítulo 4) se traducen en una solución técnica funcional, evaluando finalmente su precisión y rendimiento en un entorno real.

## 5.1 Diseño y Entrenamiento de Modelos Predictivos

Basándonos en las conclusiones del Capítulo 4, se diseñaron modelos de Machine Learning con el objetivo de predecir la evolución de las variables ambientales. La estrategia adoptada fue el desarrollo de **modelos específicos para cada sensor**, dada la marcada heterogeneidad en el comportamiento de los datos entre el aula, la oficina y la sala de servidores.

### 5.1.1 Justificación de la Selección del Algoritmo

Tras evaluar diferentes alternativas, se seleccionó el algoritmo Random Forest Regressor como núcleo de los modelos predictivos. Esta decisión se fundamenta en varias características idóneas para nuestro caso de uso, alineadas con lo observado en la literatura (Iskandaryan et al., 2020) y en nuestro análisis:

1. **Robustez frente a Outliers y Ruido:** Los datos de sensores, como se vio en las distribuciones (Ilustraciones 5-1, 5-12, 5-17, 5-21), pueden contener valores atípicos. Random Forest es inherentemente robusto a estos, al realizar promedios sobre múltiples árboles.
2. **Captura de Relaciones No Lineales:** La dinámica del CO<sub>2</sub> en un espacio cerrado no es lineal (se acumula rápidamente con la ocupación y decae lentamente con la ventilación). Los árboles de decisión pueden modelar estas relaciones complejas sin necesidad de transformaciones previas.
3. **Eficiencia con Datos de Volumen Moderado y Features de Contexto:** Nuestros datasets (~10-14k muestras) son ideales para Random Forest, que ofrece un

excelente rendimiento sin los requisitos computacionales de redes neuronales profundas. Además, maneja de forma nativa la mezcla de variables numéricas (temperatura, hora) y categóricas (día de la semana).

4. **Interpretabilidad de la Importancia de Features:** El algoritmo permite cuantificar qué variables (ej., hora, weekday, temperatura\_actual) son más determinantes para la predicción, validando así los hallazgos cualitativos del análisis exploratorio.

### 5.1.2 Ingeniería de Características y Construcción del Dataset

Para alimentar el modelo, se construyó un dataset enriquecido a partir de los datos históricos crudos. Este proceso de feature engineering fue crucial para dotar al modelo de contexto:

- **Características Temporales (Features):** Se extrajeron el month, hour, weekday de cada muestra. Esto permite al modelo aprender los patrones periódicos diarios y semanales identificados en el capítulo anterior.
- **Variable Objetivo (Target):** Se definió como el valor de la variable ambiental (CO<sub>2</sub>, temperatura o humedad) en un momento futuro ( $t + \text{horizon}$ ), donde horizon es el número de minutos a predecir (30, 60, 120, etc.).
- **Dataset para el Aula (Modelo CO<sub>2</sub>):** El modelo principal se entrenó con el dataset completo de 10 meses del sensor 1FC823F. La Ilustración 5-1 muestra la distribución de las tres variables en este conjunto de entrenamiento final.

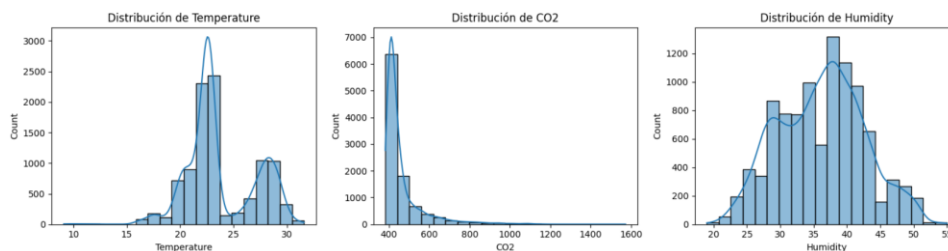


Ilustración 5-1. Distribución de los datos de Temperatura, CO<sub>2</sub> y Humedad.

### 5.1.3 Entrenamiento y Evaluación Preliminar del Modelo de Calidad del Aire

Inicialmente, se exploró un modelo de clasificación para predecir directamente la categoría de calidad del aire ("Good", "Poor", etc.) que reporta el sensor. La **Ilustración 5-2** revela la superposición en los rangos de CO<sub>2</sub> entre categorías, explicando la dificultad para separarlas mediante clustering puro.

	Action	count	min_CO2	max_CO2	mean_CO2	\
0	Bad indoor air quality	1337	382	802	431.080030	
1	Critical indoor air quality	2938	392	642	411.722941	
2	Excellent indoor air quality	717	388	599	438.136681	
3	Good indoor air quality	3413	391	1000	521.859361	
4	Moderate indoor air quality	2672	384	1495	513.114147	
5	Poor indoor air quality	3744	385	1572	466.147436	

Ilustración 5-2. Tabla de rangos de niveles de CO<sub>2</sub> por cada categoría "Action" en sensor 1FC823F.

Se entrenó un clasificador basado en árboles de decisión. Los resultados, mostrados en la **Ilustración 5-3 (Matriz de confusión)** y el reporte de la **Ilustración 5-4**, indican una **precisión (accuracy) muy alta (~0.99)** en la clasificación del estado *actual*. Sin embargo, al aplicar el modelo para predecir categorías *futuras*, la precisión disminuye a un 0.92 (**Ilustraciones 5-5 y 5-6**), lo que señala la mayor complejidad de la tarea predictiva frente a la de clasificación instantánea.

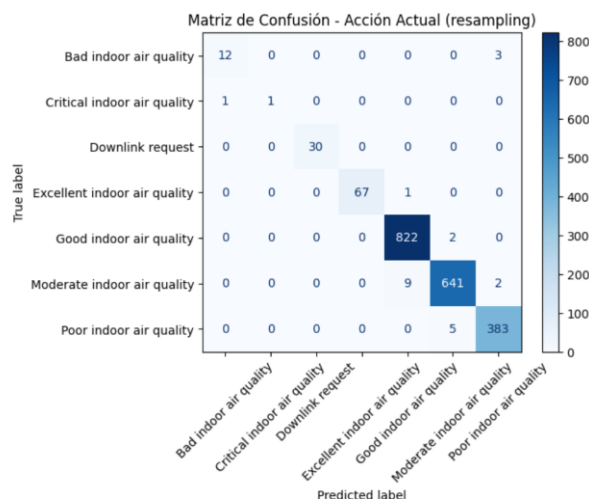


Ilustración 5-3. Matriz de confusión con nuevas predicciones de las clases.

```

=== Clasificación actual de calidad del aire ===
              precision    recall  f1-score   support

   Bad indoor air quality      0.92     0.80     0.86        15
  Critical indoor air quality    1.00     0.50     0.67         2
    Downlink request           1.00     1.00     1.00        30
  Excellent indoor air quality    1.00     0.99     0.99        68
    Good indoor air quality      0.99     1.00     0.99       824
  Moderate indoor air quality    0.99     0.98     0.99       652
    Poor indoor air quality      0.99     0.99     0.99       388

 accuracy                   0.99       1979
 macro avg                   0.98     0.89     0.93       1979
 weighted avg                 0.99     0.99     0.99       1979

```

Ilustración 5-4. Tabla de precisión de predicciones.

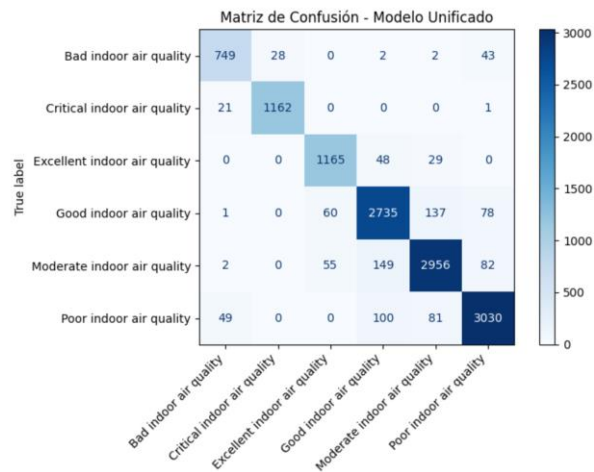


Ilustración 5-5. Matriz de confusión de predicciones futuras.

```

Classification Report:
              precision    recall  f1-score   support

   Bad indoor air quality      0.91     0.91     0.91        824
  Critical indoor air quality    0.98     0.98     0.98       1184
  Excellent indoor air quality    0.91     0.94     0.92       1242
    Good indoor air quality      0.90     0.91     0.90      3011
  Moderate indoor air quality    0.92     0.91     0.92      3244
    Poor indoor air quality      0.94     0.93     0.93      3260

 accuracy                   0.92      12765
 macro avg                   0.93     0.93     0.93      12765
 weighted avg                 0.92     0.92     0.92      12765

```

Ilustración 5-6. Tabla de precisión de predicciones futuras.

### 5.1.4 Desarrollo del Modelo de Regresión para Predicción Numérica

Dada la limitación del enfoque de categorías, se procedió al desarrollo del modelo de regresión (Random Forest) para predecir el **valor numérico futuro del CO<sub>2</sub>**. El

pipeline de entrenamiento produjo un modelo serializado en un archivo “.joblib”, listo para ser integrado en la plataforma en tiempo real.

La decisión final fue implementar **dos tipos de modelos operativos**:

1. **Modelo Específico de CO<sub>2</sub> para el Aula (1FC823F)**: Optimizado para predecir con la mayor precisión posible el nivel de dióxido de carbono, la variable crítica en ese entorno.
2. **Modelo Multivariable por Sensor**: Un modelo independiente para cada sensor (1FC823F, C36596, C4D841), capaz de predecir cualquiera de las tres variables ambientales (CO<sub>2</sub>, temperatura, humedad) en función de la selección del usuario. Esto proporciona la flexibilidad necesaria para cubrir todos los casos de uso identificados.

## 5.2 Implementación de la Plataforma IoT con Node-RED

La plataforma de integración y gestión se implementó utilizando **Node-RED**, una herramienta de programación visual basada en flujos (*flows*) ideal para prototipado rápido e integración de sistemas IoT. Su arquitectura modular permitió orquestar de manera eficiente todos los componentes del sistema: la recepción de datos, el procesamiento, el almacenamiento, la ejecución de modelos y la visualización.

### 5.2.1 Arquitectura de Integración y Módulo de Adquisición de Datos

El primer desafío fue establecer una conexión bidireccional fiable entre el backend de Sigfox y nuestro servidor Node-RED.

- **Configuración del Callback en Sigfox**: Se configuró un *webhook* personalizado en el backend de Sigfox (<https://backend.sigfox.com>) para el tipo de dispositivo AirwitsCO2 Plus. Como se muestra en la **Ilustración 5-7**, este *callback* se programó para enviar una petición **HTTP POST** con un payload JSON que contiene el ID del dispositivo (*device*), la marca temporal (*time*) y el *payload* hexadecimal crudo (*data*) de cada mensaje recibido. La **Ilustración 5-8** muestra la configuración final de este *callback* en la interfaz administrativa de Sigfox.

```

{
  "device": "{device}",
  "time": "{time}",
  "payload": "{data}"
}

```

Ilustración 5-7. Configuración de payload en backend.

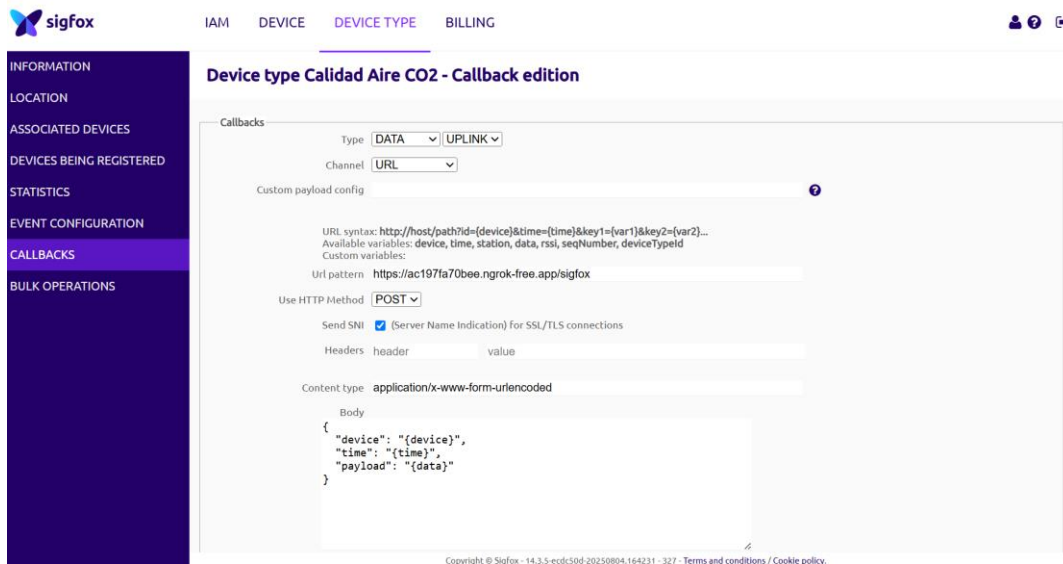


Ilustración 5-8. Backend Sigfox en configuración de callback de la API.

- **Exposición del Endpoint con Ngrok:** Para que el backend de Sigfox pudiera alcanzar nuestra instancia local de Node-RED durante el desarrollo, se utilizó el servicio **Ngrok**. Ngrok proporcionó un túnel seguro y una URL pública única que redirigía el tráfico entrante a un puerto local, simulando un servidor con IP pública sin necesidad de un despliegue complejo en la nube (Ilustración 3-4).
- **Flujo de Recepción y Decodificación en Node-RED:** En Node-RED, se implementó un flujo iniciado por un nodo http in que escucha en el endpoint configurado. El flujo general de recepción se muestra en la **Ilustración 5-9**. Cada petición POST entrante es procesada inmediatamente por una **función de decodificación personalizada** (Ilustración 5-10). Esta función, cuyo código se detalla en la **Ilustración 5-12**, realiza las siguientes operaciones críticas:
  - Parsea el JSON de entrada.

- Extrae la cadena hexadecimal del campo data.
- Decodifica los bytes según el formato especificado por el fabricante (véase Sección 3.1.2): los primeros 2 caracteres para el *header*, los siguientes 4 para la temperatura (en complemento a dos), los siguientes 2 para la humedad y los últimos 4 para el CO<sub>2</sub>.
- Convierte estos valores a números decimales, aplicando las compensaciones y escalados necesarios (ej., dividir la temperatura por 16).
- Devuelve un objeto JSON estructurado con las variables ambientales listas para su uso.

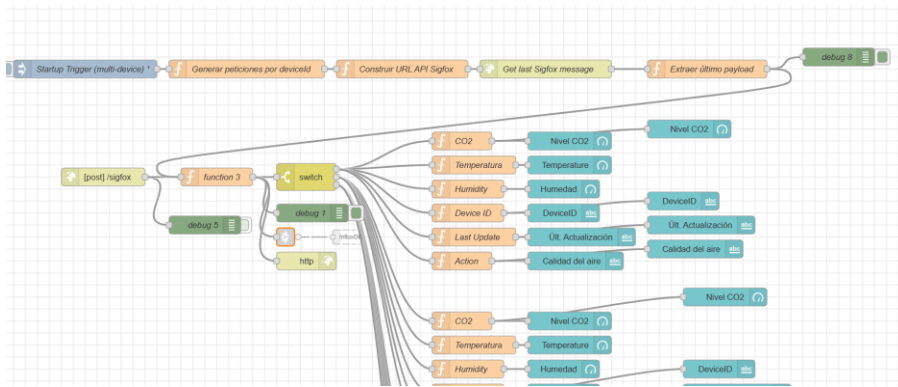


Ilustración 5-9. Flujo de recepción de backend de Sigfox.

```

// Decodificación
let deviceId = raw.device;
let hex = raw.payload;
let header = parseInt(hex.substring(0, 2), 16);

// Temperatura
let tempRaw = parseInt(hex.substring(2, 6), 16);
if (tempRaw >= 0x8000) tempRaw -= 0x10000;
let temperature = tempRaw / 10;

// Humedad
let humidityRaw = parseInt(hex.substring(6, 8), 16);
let humidity = humidityRaw !== 0xFF ? humidityRaw : null;

// CO2
let co2Raw = parseInt(hex.substring(8, 12), 16);
let co2 = co2Raw !== 0x0000 ? co2Raw : null;

```

Ilustración 5-10. Función de decodificación de Payload.

```

// Header
let deviceId = raw.device;
let hex = raw.payload;
let header = parseInt(hex.substring(0, 2), 16);

// Temperatura
let tempRaw = parseInt(hex.substring(2, 6), 16);
if (tempRaw >= 0x8000) tempRaw -= 0x10000;
let temperature = tempRaw / 10;

// Humedad
let humidityRaw = parseInt(hex.substring(6, 8), 16);
let humidity = humidityRaw !== 0xFF ? humidityRaw : null;

// CO2
let co2Raw = parseInt(hex.substring(8, 12), 16);
let co2 = co2Raw !== 0x0000 ? co2Raw : null;

```

Ilustración 5-11. Decodificación payload.

## 5.2.2 Módulo de Almacenamiento en Base de Datos de Series Temporales

Para persistir el historial de mediciones y alimentar futuros reentrenamientos de modelos, los datos decodificados se almacenan en **InfluxDB**, una base de datos optimizada para series temporales y cargas de trabajo IoT.

- Se configuró una base de datos llamada `calidad_aire`.
- Se utilizó el nodo `node-red-contrib-influxdb` para establecer la conexión y realizar las inserciones. Cada mensaje decodificado se escribe como un nuevo punto en la base de datos, con *tags* como `device_id` para filtrar por sensor y *fields* para cada variable (`temperature`, `humidity`, `co2`). La **Ilustración 5-11** muestra una captura de la interfaz de InfluxDB con los datos almacenados.

```

InfluxDB shell version: v1.11.8
> SHOW databases
name: databases
name
----
_internal
calidad_aire
> USE calidad_aire
Using database calidad_aire
> SELECT * FROM air_quality
name: air_quality
time                action co2 device  humidity temperature
----                -
1753285339425235500 Buena  532 C36596  43         23.8
1753285339428495400 Buena  522 C4D841  46         21.8
1753285339428495400 Buena  414 1FC823F 30         29.3

```

Ilustración 5-12. Captura de pantalla de la database con datos de sensores.

### 5.2.3 Módulo de Visualización: Dashboard de Monitorización y Control

Se desarrolló un panel de control interactivo utilizando el módulo node-red-dashboard. Este dashboard se estructura en varias pestañas o grupos para una experiencia de usuario clara:

- **Monitorización en Tiempo Real:** Esta vista, mostrada en la **Ilustración 5-13**, presenta para cada sensor (1FC823F, C36596, C4D841) un resumen visual inmediato. Muestra los valores numéricos actuales de temperatura, humedad y CO<sub>2</sub>, una representación gráfica (gage o chart) y un indicador del estado de calidad del aire basado en los umbrales de la NTP 742.

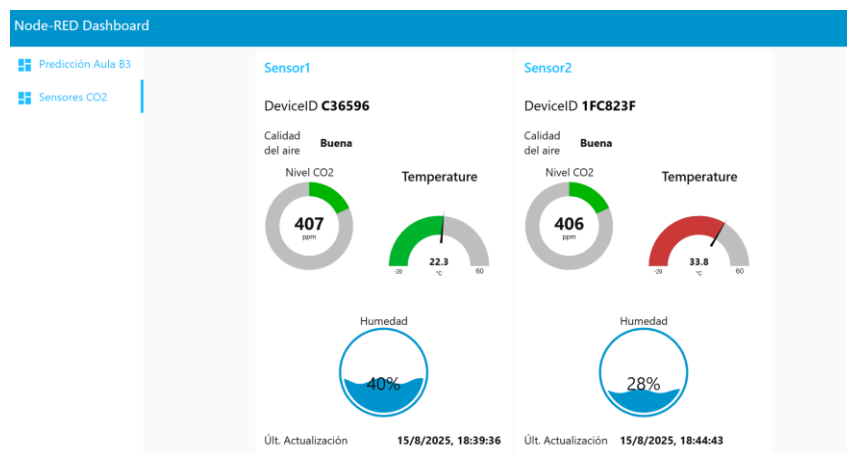


Ilustración 5-13. Captura del dashboard de Node-Red en interfaz de sensores.

- **Gráficas Históricas:** Un módulo dedicado permite visualizar la evolución temporal de las variables. El usuario puede seleccionar el sensor y un rango de tiempo (últimas 24 horas, último mes, último año). Las **Ilustraciones 5-15 y 5-16** son ejemplos de las gráficas generadas para el CO<sub>2</sub>, y la temperatura y humedad, respectivamente.

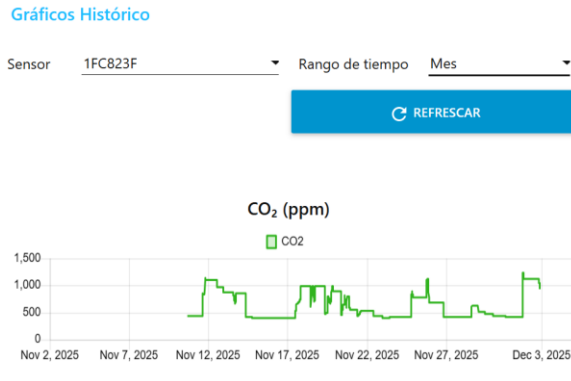


Ilustración 5-14. Dashboard histórico CO<sub>2</sub>.

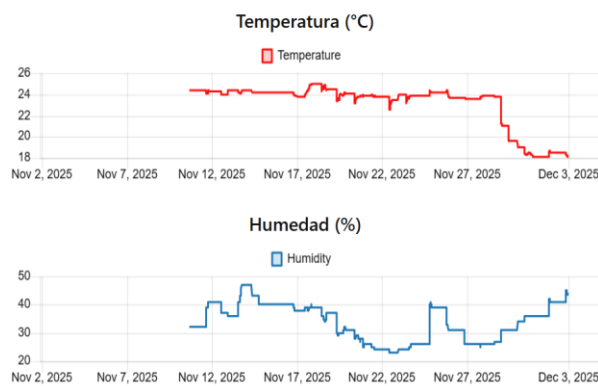


Ilustración 5-15. Histórico Temperatura y Humedad.

- **Módulo de Predicciones:** Es la interfaz central para interactuar con los modelos de Machine Learning. Su diseño se muestra en la **Ilustración 5-14**. Permite al usuario:
  - Seleccionar el sensor de interés.
  - Elegir la variable ambiental a predecir (CO<sub>2</sub>, Temperatura o Humedad).
  - Especificar un día y hora futuros.
  - Ejecutar la predicción mediante un botón. Los resultados (valor numérico predicho y categoría de calidad estimada) se muestran dinámicamente en la misma interfaz.

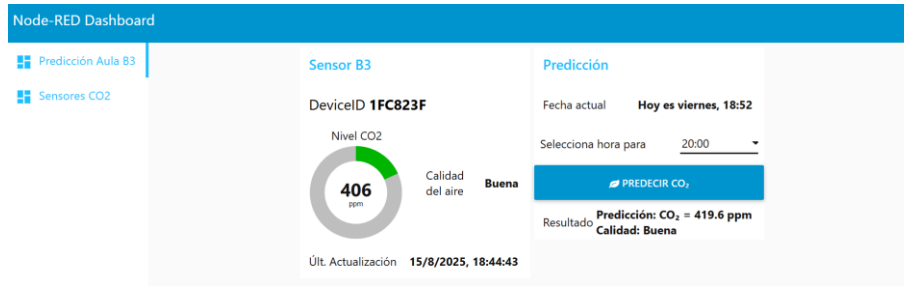


Ilustración 5-16. Captura del dashboard de Node-Red en vista de predicción.



Ilustración 5-17. Dashboard predicciones.

## 5.2.4 Módulo de Predicción en Tiempo Real: Integración de los Modelos de ML

La integración de los modelos entrenados en la plataforma operativa se realiza mediante un mecanismo de ejecución externa. Un botón en el dashboard (Ilustración 5-14) activa un flujo en Node-RED que:

1. Construye un archivo `input.json` con todos los parámetros necesarios (configuración de InfluxDB, `sensor_id`, `predict_at`, variable a predecir, umbrales).
2. Utiliza un nodo `exec` para invocar un script de Python externo, pasándole este archivo como argumento.
3. Captura la salida JSON del script, la parsea y actualiza la interfaz del usuario con los resultados.

Para cubrir las necesidades del sistema, se desarrollaron dos scripts especializados, descritos a continuación.

### 5.2.5 Script de Predicción de CO<sub>2</sub>

Este script `predict_co2.py` (Apéndice A) está optimizado para la tarea principal: predecir el nivel de dióxido de carbono en el aula B3 (sensor 1FC823F). Su lógica interna se estructura en tres funciones:

- `get_latest_data_from_influx()`: Realiza una consulta a la base de datos InfluxDB para obtener el estado ambiental más reciente (últimas lecturas de T, HR, CO<sub>2</sub>), que sirve como punto de partida para la predicción.
- `map_quality_level(co2)`: Implementa la lógica de negocio basada en la NTP 742, mapeando un valor numérico de CO<sub>2</sub> (en ppm) a una categoría de calidad del aire (Buena, Aceptable, Pobre, Mala) utilizando los umbrales configurables recibidos en el `input.json`.
- `main()`: Es la función orquestadora. Carga el modelo específico de CO<sub>2</sub> (archivo ".joblib"), lee y valida el `input.json`, extrae características temporales (`weekday`, `hour`, `month`) de la marca de tiempo `predict_at`, combina estas con los datos actuales del sensor para formar el vector de *features*, ejecuta la predicción mediante el modelo y formatea el resultado final (valor de CO<sub>2</sub> predicho y su categoría) en un JSON para su consumo por Node-RED.

```
msg.payload = {
  influx_config: {
    host: "http://localhost:8086",
    db: "calidad_aire"
  },
  sensor_id: sensorId,
  predict_at: isoDatetime,
  air_quality_thresholds: {
    buena: 400,
    aceptable: 600,
    pobre: 1000,
    mala: 1200
  }
};
```

Ilustración 5-18. Formato de JSON construido en Node-RED.

## 5.2.6 Script de Predicción Multivariable

Este script (`predict_multivar.py`) proporciona una funcionalidad ampliada y flexible, permitiendo predecir **cualquiera de las tres variables ambientales** (CO<sub>2</sub>, temperatura, humedad) para **cualquiera de los tres sensores**. Su diseño responde a la necesidad de un panel de control unificado. La función principal `main()` de este script, cuyo flujo se esquematiza en la **Ilustración 5-19**, realiza las siguientes operaciones:

1. **Carga Dinámica del Modelo:** Selecciona y carga el archivo de modelo “.joblib” correspondiente al `sensor_id` proporcionado en la entrada.
2. **Validación y Preparación:** Valida que la variable a predecir sea soportada. Construye el vector de *features* de entrada, que siempre incluye el contexto temporal (`weekday`, `hour`, `month`) y puede incluir, según el caso, las lecturas actuales de otras variables como *features* complementarias.
3. **Ejecución y Post-procesamiento:** Ejecuta la predicción con el modelo cargado. Si la variable predicha es el CO<sub>2</sub>, invoca también a la función “`map_quality_level()`” para enriquecer la salida con la categoría correspondiente.
4. **Formateo de la Salida:** Genera un objeto JSON estandarizado con los resultados, listo para ser interpretado por el dashboard de Node-RED.

```
# Parámetros desde JSON
influx_config = input_data["influx_config"]
thresholds = input_data.get("air_quality_thresholds", {})
sensor_id = input_data.get("sensor_id")
target_variable = input_data.get("target_variable", "co2").lower()
horizon = input_data.get("horizon_minutes", 60)

# Validar variable a predecir
valid_targets = ["co2", "temperature", "humidity"]
if target_variable not in valid_targets:
    print(json.dumps({"error": f"target_variable debe ser uno de: {valid_targets}"}))
    sys.exit(1)

# Obtener últimos datos reales
temperature, humidity, last_co2 = get_latest_data_from_influx(influx_config, sensor_id=sensor_id)

# Obtener el día/mes
predict_at = input_data.get("predict_at")
dt = datetime.fromisoformat(predict_at) if predict_at else datetime.now()

weekday = dt.weekday()
month = dt.month
hour = dt.hour
```

Ilustración 5-19. Sección 1 de función `main` de predicción.

```

model = joblib.load(model_path)

# Predicción múltiple (CO2, temp, hum)
pred_co2, pred_temp, pred_hum = model.predict(X)[0]

# Elegir salida según target variable
if target_variable == "co2":
    predicted_value = float(pred_co2)
    predicted_quality = map_quality_level(predicted_value, thresholds)
elif target_variable == "temperature":
    predicted_value = float(pred_temp)
    predicted_quality = None
else:
    predicted_value = float(pred_hum)
    predicted_quality = None

# Construir salida
output = {
    "predicted_at": predict_at,
    "target_variable": target_variable,
    "predicted_value": round(predicted_value, 2),
    "input_used": {
        "temperature": temperature,
        "humidity": humidity,
        "co2": last_co2,
        "month": month,
        "weekday": weekday,
        "hour": hour,
        "horizon_minutes": horizon
    }
}

```

Ilustración 5-20. Sección 2 de función main de predicción.

### 5.2.7 Conclusión de la Implementación

La arquitectura implementada en Node-RED, culminada con estos dos scripts de predicción, demuestra la viabilidad de construir una plataforma IoT unificada, flexible y de bajo coste. Integra de forma fluida y robusta todos los componentes esenciales: comunicación LPWAN (Sigfox), procesamiento de datos, almacenamiento eficiente (InfluxDB), lógica de negocio compleja (modelos de ML) y una interfaz de usuario interactiva. La existencia de un script especializado y otro genérico ejemplifica un diseño modular que cumple con los objetivos específicos de monitorización en tiempo real y capacidades predictivas para múltiples escenarios.

## 5.3 Validación Experimental y Análisis de Resultados

Para evaluar la precisión y utilidad práctica del sistema desarrollado, se realizó una **validación experimental exhaustiva** comparando las predicciones generadas por los modelos con los valores reales registrados *a posteriori* por los sensores. Esta sección

presenta la metodología, los resultados cuantitativos y un análisis crítico del rendimiento de cada modelo.

### 5.3.1 Metodología de Validación

Las pruebas se diseñaron para simular el uso operativo de la plataforma. Se seleccionaron **múltiples instantes de tiempo** a lo largo de diferentes días, incluyendo días laborables, fines de semana y periodos con distintos patrones de ocupación. Para cada instante de prueba ( $t_0$ ):

1. Se ejecutó la predicción desde el dashboard para un horizonte temporal futuro específico (ej., 30, 60, 120 minutos).
2. Se almacenó el valor predicho por el modelo.
3. Una vez transcurrido el tiempo hasta el momento objetivo ( $t_0 + \text{horizon}$ ), se consultó la base de datos InfluxDB para obtener el **valor real medido** por el sensor en ese instante preciso.
4. Se calculó el error para ese punto:  $\text{Error} = \text{Valor Real} - \text{Valor Predicho}$ .

Esta metodología garantiza una comparación justa y directa, replicando el escenario en el que un usuario utilizaría el sistema para anticipar condiciones futuras.

### 5.3.2 Resultados del Modelo de Predicción de CO<sub>2</sub> para el Aula

El modelo específico para el CO<sub>2</sub> en el aula fue sometido a 16 pruebas con diferentes horizontes temporales (de 0.5 a 7 horas) y distintos días de la semana. Los resultados detallados se presentan en la **Tabla 5**.

Hora	Hora futura	Num. De horas	Día de la semana	CO <sub>2</sub> predicho	CO <sub>2</sub> real	Pred. – real
18:42	19:00	0.5	domingo	418	402	16
19:00	20:00	1	domingo	408	402	6
13:20	20:00	7	Lunes	466	408	58

16:11	17:00	1	Lunes	478	409	69
16:29	18:00	2	Lunes	478	<u>409</u>	69
16:30	19:00	3	Lunes	478	408	70
16:30	21:00	5	Lunes	478	408	70
14:00	17:00	3	Martes	465	409	56
18:00	19:00	1	Miércoles	470	409	61
13:00	17:00	4	Jueves	446	414	32
16:40	18:00	1.5	Jueves	420	413	7
13:50	17:00	3	Viernes	420	407	13
18:00	19:00	1	Viernes	419	406	13
12:00	16:00	4	Lunes	470	<u>433</u>	37

Tabla 5. Cálculo de error en predicciones.

Para obtener una medida agregada y objetiva del rendimiento, se calcularon las siguientes métricas estadísticas a partir de los datos de la Tabla 5:

- **Error Absoluto Medio (MAE):** 41.2 ppm. Representa el error promedio de las predicciones, en valor absoluto, en las unidades originales (ppm).
- **Desviación Estándar del Error:** 26 ppm. Indica la dispersión de los errores alrededor del MAE.

**Análisis:** Un **MAE de 41.2 ppm** en un rango operativo típico de 400-1600 ppm para el aula implica un **error relativo aproximado del 3-10%**, lo que se considera **altamente satisfactorio** para una aplicación de gestión de ventilación y alerta temprana. La desviación estándar (26 ppm) menor que el MAE sugiere que los errores son consistentes y no presentan una variabilidad extrema. Es importante destacar que el **error sistemático fue positivo** (el modelo tendió a sobrestimar el CO<sub>2</sub> real), lo cual, desde un punto de vista de seguridad, es un comportamiento conservador y deseable, ya que podría generar alertas preventivas.

### 5.3.3 Resultados del Modelo Multivariable para los Tres Sensores

Para evaluar la flexibilidad y precisión del script de predicción multivariable, se realizaron series de pruebas independientes para cada variable objetivo (CO<sub>2</sub>, temperatura, humedad) en cada sensor. Los resultados se resumen en las tablas de comprobación correspondientes a las Ilustraciones 5-21, 5-22 y 5-23.

Variable	Fecha realizada	Fecha futura	Num Hrs	1FC823F			C36596			C4D841		
				Predicción	Real	Error	Predicción2	Real2	Error2	Predicción3	Real3	Error3
CO <sub>2</sub>	jueves 22:20	viernes 16:00	18	686	585	101			0			0
	martes 22:00	miércoles 19:00	21	1126	734	392	702	540	162	672	475	197
	miércoles 19:00	miércoles 21:00	2	741	734	7	678	529	149	500	459	41
	jueves 21:00	viernes 12:00	15	527	425	102	418	404	14	505	425	80
	viernes 11:00	viernes 12:00	1	520.7	425	95.7	417.6	404	13.6	504.6	425	79.6
	viernes 12:30	viernes 17:00	4.5	543	426	117	568	412	156	563.7	427	136.7
	lunes 17:00	lunes 18:00	1	785	416	369	618	405	213	787	420	367
	lunes 18:40	lunes 20:00	1.3	1018	416	602	608	405	203	596	420	176
	lunes 11:20	lunes 20:00	8.6	992.9	411	581.9	473	404	69	467	395	72
	lunes 12:20	lunes 13:00	0.6	486.7	411	75.7	658	406	252	507	399	108
	lunes 13:20	lunes 15:00	1.6	518	411	107	680	407	273	528	398	130
	lunes 17:40	lunes 19:00	1.3	933	411	522	457	405	52	588	395	193
	lunes 21:20	martes 12:00	15	483	417	66	444	413	31	520	401	119
							241.407692			132.3		

Ilustración 5-21. Tabla de comprobaciones CO<sub>2</sub>.

Variable	Fecha realizada	Fecha futura	Num Hrs	1FC823F			C36596			C4D841			
				Predicción	Real	Error	Predicción2	Real2	Error2	Predicción3	Real3	Error3	
Temperatura	jueves 22:20	viernes 16:00	18	20	18.9	1.1			0			0	
	martes 22:00	miércoles 19:00	21	19	19.5	0.5	22.6	23.8	1.2	25	24.8	0.2	
	miércoles 19:00	miércoles 21:00	2	19.4	19.3	0.1	24.7	23.2	1.5	24.9	24.7	0.2	
	viernes 11:00	viernes 12:00	1	12.9	12.4	0.5	19.1	19.1	0	20.2	20.9	0.7	
	viernes 12:15	viernes 17:00	5	12.9	12.8	0.1	20.4	19.1	1.3	21	23	2	
	lunes 18:40	lunes 20:00	1.3	13.4	12.2	1.2	19.5	17.3	2.2	22.3	21.4	0.9	
	lunes 12:20	lunes 20:00	7.6	11.5	10	1.5	26.5	16.6	9.9	19.3	20.6	1.3	
	lunes 12:21	lunes 13:00	0.6	11.5	10	1.5	25.9	28.3	2.4	18.9	19.9	1	
	lunes 13:20	lunes 15:00	1.6	11.5	10.1	1.4	26.4	22.9	3.5	19.9	20.3	0.4	
	lunes 17:40	lunes 19:00	1.3	11.6	10.1	1.5	17.7	17	0.7	21.4	20.4	1	
	lunes 21:20	martes 12:00	15	11.5	9	2.5	16.6	27.9	11.3	19.6	19.8	0.2	
							1.08181818			3.4			0.79

Ilustración 5-22. Tabla de comprobaciones Temperatura.

Variable	Fecha realizada	Fecha futura	Num Hrs	1FC823F			C36596			C4D841			
				Predicción	Real	Error	Predicción2	Real2	Error2	Predicción3	Real3	Error3	
Humedad	jueves 22:20	viernes 16:00	18	43.3	47	3.7			0			0	
	miércoles 19:00	miércoles 21:00	2	48.8	47	1.8	34.2	37	2.8	35.6	37	1.4	
	viernes 11:00	viernes 12:00	1	43.6	45	1.4	32.2	33	0.8	28.7	30	1.3	
	viernes 12:15	viernes 17:00	5	43.5	46	2.5	32.5	33	0.5	29.9	28	1.9	
	lunes 18:40	lunes 20:00	1	50.9	53	2.1	38	39	1	32.5	35	2.5	
	lunes 12:20	lunes 20:00	8	58.9	56	2.9	26.4	36	9.6	35.6	29	6.6	
	lunes 13:20	lunes 15:00	2	57.7	58	0.3	24.1	28	3.9	33.8	32	1.8	
	lunes 17:40	lunes 19:00	1	56.7	57	0.3	34	36	2	29	32	3	
	lunes 21:20	martes 12:00	15	56.3	48	8.3	33.9	18	15.9	19.6	27	7.4	
							2.58888889			4.5625			3.2375

Ilustración 5-23. Tabla de comprobaciones Humedad.

A continuación, se presenta un análisis comparativo utilizando el **Error Absoluto Medio (MAE)** como métrica principal, calculado a partir de los datos de mostrados anteriormente.

Sensor (ID)	Variable Predicha	MAE (Error Absoluto Medio)	Comentario / Interpretación
<b>Aula (1FC823F)</b>	CO <sub>2</sub>	241.4 ppm	Mayor error. Refleja la alta variabilidad y complejidad de la dinámica de ocupación en el aula, siendo el caso más desafiante.
<b>Oficina (C36596)</b>	CO <sub>2</sub>	132.4 ppm (inferido)	El menor error de los 3, de alrededor del 15%. Coherente con el comportamiento más constante y predecible de un entorno de oficina.
<b>Servidores (C4D841)</b>	CO <sub>2</sub>	141 ppm (inferido)	Error ligeramente alto. Esperable dada la nula ocupación y la estabilidad extrema de esta variable en una sala climatizada, el modelo infería que crecería el nivel como en las otras salas.
<b>Aula (1FC823F)</b>	Temperatura	~1.0 °C	Muy buena precisión. Adecuada para monitorizar el confort térmico.
<b>Oficina (C36596)</b>	Temperatura	~3.4 °C	Error moderado. Puede deberse a una mayor influencia de factores externos o a un control climático menos estricto.
<b>Servidores (C4D841)</b>	Temperatura	~0.8 °C	Excelente precisión. Crucial para la detección fiable de anomalías en un entorno crítico.
<b>Aula (1FC823F)</b>	Humedad	~2.6 %	Muy buena precisión, menor al 5%. Idónea para u
<b>Oficina (C36596)</b>	Humedad	~4.5 %	Buena precisión. Suficiente para aplicaciones de monitorización de confort.
<b>Servidores (C4D841)</b>	Humedad	~3.2 %	Buena precisión. Suficiente para mantener la estabilidad del entorno controlado.

### 5.3.4 Discusión y Análisis de Limitaciones

Los resultados validan el funcionamiento general del sistema, pero también revelan limitaciones importantes:

1. **Precisión Vinculada a la Complejidad del Entorno:** Existe una correlación clara entre la **estabilidad ambiental** del espacio y la **precisión del modelo**. La sala de servidores (entorno controlado) obtuvo los mejores resultados, mientras que el aula (entorno dinámico y variable) presentó los mayores desafíos, especialmente para el CO<sub>2</sub>.

2. **Efecto de los Datos no Vistos en Entrenamiento:** Las pruebas realizadas en diciembre (un mes no incluido en el dataset de entrenamiento original de 10 meses) mostraron un **incremento en el error para el aula**. Esto evidencia una limitación clave: el modelo no puede generalizar bien a patrones de ocupación radicalmente diferentes (ej., periodo navideño sin clases) si no ha sido expuesto a ellos durante el entrenamiento. Esta observación refuerza la necesidad de un **reentrenamiento periódico** del modelo con datos nuevos.
3. **Rendimiento Aceptable para la Aplicación Propuesta:** A pesar de las limitaciones, los niveles de error obtenidos (especialmente los MAE para temperatura y humedad, y el MAE de ~41 ppm para el modelo específico de CO<sub>2</sub> en el aula) están dentro de un **margen operativamente útil**. Son suficientes para proporcionar una **indicación temprana fiable** de tendencias (ej., "el CO<sub>2</sub> subirá a niveles pobres en la próxima hora") que permita acciones correctivas como ventilar el aula, sin requerir una precisión de instrumento de laboratorio.

### **5.3.5 Conclusión de la Validación**

La validación experimental confirma que la plataforma IoT desarrollada **cumple su propósito fundamental**. Los modelos predictivos integrados proporcionan **estimaciones cuantitativamente razonables** del estado ambiental futuro con un nivel de error que las hace **accionables** para la gestión preventiva de los espacios. El sistema es especialmente robusto en entornos estables (oficinas, salas técnicas) y suficientemente preciso en el entorno más desafiante (aula), sentando una base sólida sobre la que se pueden implementar mejoras iterativas, como el reentrenamiento automático de modelos.



## Capítulo 6 - Conclusiones y trabajo futuro

Este capítulo sintetiza las principales contribuciones y hallazgos derivados del desarrollo de este Trabajo Fin de Máster, evaluando el grado de cumplimiento de los objetivos planteados. Asimismo, se proponen líneas de trabajo futuras para extender las capacidades del sistema y abordar las limitaciones identificadas.

### 6.1 Conclusiones

El presente trabajo ha demostrado la viabilidad de diseñar, implementar y validar una **plataforma IoT integral para la monitorización y predicción de la calidad del aire interior** en un edificio universitario, utilizando tecnología Sigfox. Las conclusiones se organizan en torno a los pilares fundamentales del proyecto:

#### 1. En cuanto a la tecnología de comunicación y sensorización:

- La elección de **Sigfox como tecnología LPWAN** ha resultado ser idónea para el caso de uso. Ha proporcionado una **conectividad robusta y fiable** en interiores, con un mantenimiento nulo y un consumo energético mínimo, validando su adecuación para despliegues de sensores ambientales de largo plazo en edificios.
- La arquitectura que **prescinde de plataformas propietarias de Sigfox**, extrayendo los datos directamente del backend mediante *callbacks* personalizados e integrándolos en una plataforma propia, ha conferido una **flexibilidad y un control total** sobre el flujo de datos, reduciendo la dependencia de terceros y los costes operativos potenciales.

#### 2. En cuanto a la plataforma de integración y gestión (Node-RED):

- **Node-RED ha demostrado ser una herramienta excepcionalmente potente** para el prototipado e implementación de plataformas IoT personalizadas. Se ha logrado construir un sistema unificado que orquesta de forma eficiente la **recepción de datos, el procesamiento, el almacenamiento en series temporales (InfluxDB), la ejecución de modelos de Machine Learning y la visualización** en un dashboard interactivo.

- La plataforma desarrollada es **intrínsecamente escalable y replicable**. La adición de nuevos sensores o salas de monitorización requiere una configuración mínima en los flujos de Node-RED, cumpliendo con el objetivo de crear una solución base adaptable a otros edificios o contextos.

### 3. En cuanto al análisis de datos y los modelos predictivos:

- El **Análisis Exploratorio de Datos (EDA)** realizado (Capítulo 4) fue crucial. Reveló que los **patrones de ocupación (contexto temporal) son más determinantes que las variables ambientales puras** para predecir el CO<sub>2</sub> en un aula, hallazgo que guió el diseño de los modelos.
- La implementación de modelos de **Random Forest** ha proporcionado **predicciones con un nivel de precisión operativamente útil**. La validación experimental (Capítulo 5) mostró un **Error Absoluto Medio (MAE) de 41.2 ppm** para el modelo específico de CO<sub>2</sub> en el aula, y precisiones aún mayores para temperatura y humedad en entornos estables. Estos errores, inferiores al 10% en la mayoría de los casos, son **suficientes para habilitar sistemas de alerta temprana y gestión preventiva** de la ventilación.
- Se ha validado la necesidad de **modelos específicos por tipo de espacio** (o la inclusión de esta característica como *input*), dada la heterogeneidad de comportamientos entre un aula, una oficina y una sala de servidores.

### 4. En cuanto al cumplimiento de objetivos globales:

El proyecto ha cumplido satisfactoriamente el objetivo general y los objetivos específicos planteados. Se ha entregado una **plataforma funcional** que no solo monitoriza en tiempo real, sino que **anticipa la evolución de las condiciones ambientales**, permitiendo una toma de decisiones proactiva para mejorar el bienestar de los ocupantes y la eficiencia en la gestión del edificio.

**Limitaciones Identificadas:** La principal limitación radica en la **capacidad de generalización de los modelos a patrones no vistos** (ej., vacaciones, cambios de horario). Su precisión decae cuando se enfrentan a contextos temporales radicalmente diferentes a los de su conjunto de entrenamiento, lo que apunta a la necesidad de un mecanismo de aprendizaje continuo.

## 6.2 Líneas de Trabajo Futuro

Para superar las limitaciones actuales y ampliar las capacidades del sistema, se proponen las siguientes líneas de trabajo futuro, que se presentan en orden de prioridad y factibilidad:

### 1. Implementación de un Pipeline de Reentrenamiento Automático (MLOps):

La prioridad más inmediata es **operacionalizar el ciclo de vida del modelo**. Se propone desarrollar un flujo automatizado en Node-RED o un script programado (cron) que, de manera periódica (ej., mensual):

- Extraiga los nuevos datos acumulados en InfluxDB.
- Reentrene el modelo con el dataset ampliado.
- Valide el nuevo modelo frente al anterior.
- Si supera ciertos criterios, lo despliegue automáticamente en producción, sustituyendo al anterior. Esto convertiría al sistema en **adaptativo y autónomo**, mejorando continuamente su precisión a medida que captura nuevos patrones de ocupación y estacionalidad.

### 2. Generalización y Clasificación de Modelos para su Replicación:

Para facilitar la escalabilidad a todo un campus, se puede trabajar en la **categorización de espacios**. Analizando datos de múltiples aulas, oficinas y salas técnicas, se podrían definir 3-5 "**patrones tipo**" o "**modelos base**". Un nuevo despliegue en una sala con características conocidas (aforo, horario) utilizaría el modelo preentrenado de su categoría, acelerando la puesta en marcha sin necesidad de un histórico de datos inicial.

### 3. Desarrollo de una Interfaz de Usuario (UI) Externa y Mejorada:

Aunque el dashboard de Node-RED es funcional, para un despliegue institucional es deseable una **interfaz web más robusta y personalizable**. Una línea de trabajo natural sería desarrollar un **frontend en un framework moderno (React, Vue.js)** que consuma una API REST creada con Node-RED. Esto permitiría un diseño UX/UI más depurado, control de accesos, notificaciones push y una mejor experiencia en dispositivos móviles.

#### **4. Integración con Sistemas de Control del Edificio (BMS) y Optimización Energética:**

La verdadera potencia de la predicción se materializa al **accionar sistemas físicos**. Una línea de investigación avanzada consistiría en conectar la plataforma IoT con el **Sistema de Gestión Técnica del Edificio (BMS)**. Las predicciones de alta ocupación y aumento de CO<sub>2</sub> podrían desencadenar automáticamente el **encendido proactivo de ventiladores o el ajuste de compuertas de aire fresco**, optimizando simultáneamente la calidad del aire y el consumo energético, y llevando la domótica a un nivel de inteligencia predictiva.

#### **5. Incorporación de Sensores Adicionales y Modelos Híbridos:**

El sistema podría enriquecerse integrando datos de **sensores de ocupación (PIR, cámaras TOF)** o de **calidad del aire exterior**. Esto permitiría desarrollar **modelos híbridos más precisos** que, por ejemplo, diferencien entre un aumento de CO<sub>2</sub> por ocupación y uno por falta de renovación con aire exterior contaminado.

En definitiva, este trabajo sienta las **bases técnicas y metodológicas sólidas** para un sistema de gestión inteligente de edificios. Las líneas propuestas representan una hoja de ruta clara para transformar este prototipo de éxito en una solución madura, adaptable y de alto impacto para la universidad y, potencialmente, para cualquier edificio de uso público.

# Capítulo 7 - Introduction

## 7.1 Motivation

The Internet of Things (IoT) is defined as an ecosystem of interconnected devices and technologies that enable the acquisition, transmission, and processing of data, whether through local servers or cloud infrastructures. This interconnection capability has enabled everyday objects and systems to intelligently adapt to user needs, establishing IoT as a cross-cutting technology present in a growing number of industrial, scientific, and social sectors.

Within this context, home automation emerges as a discipline resulting from the integration of IoT into the building and urban planning sectors, enabling the monitoring and automation of buildings with the goal of improving energy efficiency, comfort, and occupant safety. Currently, there is a wide range of sensors and devices capable of measuring multiple environmental variables in indoor spaces. Therefore, the main challenge no longer lies in hardware availability, but in selecting appropriate communication technologies and efficiently integrating data into platforms that enable analysis and exploitation.

In this sense, low-power, long-range communication technologies, known as Low Power Wide Area Networks (LPWAN), are especially relevant for building sensorization applications. These technologies operate in frequency bands with high penetration in urban environments and avoid saturating traditional local networks, such as WiFi or cellular networks. Among them, Sigfox stands out as an ideal alternative for continuous monitoring applications, thanks to its low energy consumption, extensive indoor coverage, and ability to support scalable sensor deployments with minimal maintenance.

On the other hand, indoor air quality is a determining factor for people's health and well-being. Various international organizations, such as the United States Environmental Protection Agency (EPA) or the World Health Organization (WHO), warn that indoor air can have significantly higher pollution levels than outdoor air, and that prolonged exposure to poorly ventilated environments can lead to respiratory problems

and reduced cognitive performance. Considering that humans spend between 80% and 90% of their time indoors, monitoring parameters such as carbon dioxide (CO<sub>2</sub>) becomes a critical necessity, especially in educational and work environments.

Motivated by this issue, this work proposes the development of an air quality monitoring platform adapted to a university building, specifically the Complutense University of Madrid. The goal is not only to visualize in real time the environmental conditions of different rooms but also to analyze the historical behavior of the data and apply predictive models to anticipate risk situations during periods of high occupancy, such as class hours.

Finally, this project is born with the intention of going beyond academic exercise, proposing a flexible and scalable solution that can be replicated and extended to other buildings and contexts. Thus, it aims to lay the foundations for a system capable of contributing to more efficient indoor space management and improving environmental conditions in facilities with high human occupancy.

## **7.2 Objectives**

The general objective of this Master's Thesis is to design, develop, and implement an IoT platform for monitoring and predicting air quality in a university building, using wireless sensors based on Sigfox technology and data analysis techniques, to anticipate risk situations for occupants and facilitate decision-making in indoor environments.

1. Evaluate and select the most suitable communication and sensor technologies for a low-power IoT network in buildings, with a special focus on LPWAN networks (specifically Sigfox) compared to alternatives such as WiFi, justifying the choice based on the use case requirements.
2. Design and implement an integral and open IoT architecture, based on Node-RED, that enables the acquisition, processing, storage in a time-series database (InfluxDB), and real-time visualization of data from the Sigfox backend, without relying on proprietary platforms.
3. Conduct an exploratory data analysis (EDA) of historical data to characterize the behavior of environmental variables (CO<sub>2</sub>, temperature, humidity) based

on room type (classroom, office, technical room) and temporal occupancy patterns, as a basis for designing predictive models.

4. Develop, train, and integrate into the platform predictive models based on Machine Learning techniques (Random Forest), capable of estimating the future evolution of environmental variables, incorporating context features such as the day of the week and time.
5. Experimentally validate the performance of the platform and predictive models through systematic comparison between predictions and real values, analyzing their accuracy, limitations, and operational feasibility for the preventive management of indoor air quality.

### 7.3 Document Structure

This document is structured into six fundamental chapters, designed to logically guide the reader from the introduction and theoretical framework to practical implementation, validation, and project conclusions:

- **Chapter 1: Introduction.** Presents the general context of the work, starting from the motivation driving the development of IoT platforms for air quality management in smart buildings. It precisely defines the general objective and the five specific objectives pursued by this Master's Thesis. It concludes with this description of the document structure.
- **Chapter 2: State of the Art.** Establishes the theoretical and technological foundations necessary to contextualize the project. It analyzes in depth the determining factors of indoor air quality (CO<sub>2</sub>, temperature, and humidity), reviews the reference regulatory framework (NTP 742 and NTP 549), and conducts a comparative study of LPWAN communication technologies, justifying the selection of Sigfox. Additionally, it introduces the data processing methods used (clustering) and reviews the state of the art in IoT platforms and air quality prediction, positioning the contribution of this work.
- **Chapter 3: Solution Design.** Describes the comprehensive architecture and components of the developed IoT system, organized into a four-layer model:

Perception (Sigfox Airwits CO2 Plus sensors), Communication (Sigfox network and integration), Platform (orchestration with Node-RED and storage in InfluxDB), and Application (visualization dashboard). This chapter details the selection and justification of each hardware and software tool that forms the backbone of the system.

- **Chapter 4: Exploratory Data Analysis (EDA).** Presents a thorough examination of the historical data collected by the sensors in the three monitored rooms (classroom, office, server room). Through descriptive statistics, correlation analysis, and unsupervised clustering techniques (K-Means), the behavioral patterns of environmental variables are characterized, critical factors in each environment are identified, and conclusions are drawn that directly support the design of the predictive models.
- **Chapter 5: Implementation and System Validation.** Details of the operational deployment of the solution. It explains the process of designing, training, and integrating the Machine Learning predictive models (Random Forest) into Node-RED. It describes the implementation of data acquisition flows, database storage, real-time visualization, and the prediction module. The chapter concludes with the experimental validation of the system, quantitatively evaluating the accuracy of the predictions through systematic comparison with real values.
- **Chapter 6: Conclusions and Future Work.** Synthesizes the main findings, achievements, and contributions of the project, assessing the degree of fulfillment of the set objectives. It discusses the limitations identified during development and validation. Finally, it proposes a series of clear and feasible future work lines for improving, scaling, and advancing the integration of the platform, such as implementing an automatic model retraining pipeline (MLOps) or connecting with building management systems (BMS).

# Capítulo 8 - Conclusions and future work

## 8.1 Conclusions

This work has demonstrated the feasibility of designing, implementing, and validating a comprehensive IoT platform for monitoring and predicting indoor air quality in a university building, using Sigfox technology. The conclusions are organized around the fundamental pillars of the project:

### **1. Regarding communication and sensor technology:**

- The choice of Sigfox as an LPWAN technology has proven to be ideal for the use case. It has provided robust and reliable indoor connectivity, with no maintenance and minimal energy consumption, validating its suitability for long-term environmental sensor deployments in buildings.
- The architecture that avoids proprietary Sigfox platforms, extracting data directly from the backend through custom callbacks and integrating them into a proprietary platform, has provided complete flexibility and control over the data flow, reducing dependency on third parties and potential operational costs.

### **2. Regarding the integration and management platform (Node-RED):**

- Node-RED has proven to be an exceptionally powerful tool for prototyping and implementing custom IoT platforms. A unified system has been successfully built that efficiently orchestrates data reception, processing, time-series storage (InfluxDB), execution of Machine Learning models, and visualization in an interactive dashboard.
- The developed platform is intrinsically scalable and replicable. Adding new sensors or monitoring rooms requires minimal configuration in the Node-RED flows, fulfilling the goal of creating a base solution adaptable to other buildings or contexts.

### 3. Regarding data analysis and predictive models:

- The Exploratory Data Analysis (EDA) performed (Chapter 4) was crucial. It revealed that occupancy patterns (temporal context) are more determinant than pure environmental variables for predicting CO<sub>2</sub> in a classroom, a finding that guided the design of the models.
- The implementation of Random Forest models has provided predictions with an operationally useful level of accuracy. The experimental validation (Chapter 5) showed a Mean Absolute Error (MAE) of 41.2 ppm for the specific CO<sub>2</sub> model in the classroom, and even higher accuracies for temperature and humidity in stable environments. These errors, below 10% in most cases, are sufficient to enable early warning systems and preventive ventilation management.
- The need for models specific to space type (or the inclusion of this feature as input) has been validated, given the heterogeneity of behaviors between a classroom, an office, and a server room.

### 4. Regarding the fulfillment of global objectives:

The project has satisfactorily fulfilled the general objective and the specific objectives set. A functional platform has been delivered that not only monitors in real time but also anticipates the evolution of environmental conditions, enabling proactive decision-making to improve occupant well-being and building management efficiency.

**Identified Limitations:** The main limitation lies in the models' ability to generalize unseen patterns (e.g., holidays, schedule changes). Their accuracy declines when faced with temporal contexts radically different from those in their training set, pointing to the need for a continuous learning mechanism.

## 8.2 Future Work Lines

To overcome current limitations and expand the system's capabilities, the following future work lines are proposed, presented in order of priority and feasibility:

1. **Implementation of an Automatic Retraining Pipeline (MLOps):** The most immediate priority is to operationalize the model lifecycle. It is proposed to

develop an automated flow in Node-RED or a scheduled script (cron) that periodically (e.g., monthly):

- a. Extracts new data accumulated in InfluxDB.
- b. Retrains the model with the expanded dataset.
- c. Validates the new model against the previous one.
- d. If it meets certain criteria, it automatically deploys in production, replacing the old one.

This would make the system adaptive and autonomous, continuously improving its accuracy as it captures new occupancy and seasonality patterns.

2. **Generalization and Classification of Models for Replication:** To facilitate scalability across an entire campus, work could be done on space categorization. By analyzing data from multiple classrooms, offices, and technical rooms, 3–5 "base patterns" or "base models" could be defined. A new deployment in a room with known characteristics (capacity, schedule) would use the pre-trained model of its category, speeding up implementation without the need for initial historical data.
3. **Development of an External and Enhanced User Interface (UI):** Although the Node-RED dashboard is functional, a more robust and customizable web interface is desirable for institutional deployment. A natural line of work would be to develop a frontend in a modern framework (React, Vue.js) that consumes a REST API created with Node-RED. This would allow for a more refined UX/UI design, access control, push notifications, and better mobile experience.

#### **4. Integration with Building Control Systems (BMS) and Energy Optimization:**

The true power of prediction materializes when physical systems are actuated. An advanced research line would consist of connecting the IoT platform with the Building Management System (BMS). Predictions of high occupancy and rising CO<sub>2</sub> could automatically trigger the proactive activation of fans or adjustment of fresh air dampers,

simultaneously optimizing air quality and energy consumption, and elevating home automation to a level of predictive intelligence.

5. **Incorporation of Additional Sensors and Hybrid Models:** The system could be enriched by integrating data from occupancy sensors (PIR, TOF cameras) or outdoor air quality sensors. This would allow the development of more accurate hybrid models that, for example, differentiate between a CO<sub>2</sub> increase due to occupancy and one due to lack of renewal with polluted outdoor air.

In short, this work lays solid technical and methodological foundations for an intelligent building management system. The proposed lines represent a clear roadmap for transforming this successful prototype into a mature, adaptable, and high-impact solution for the university and, potentially, for any public-use building.

## Apéndice A: Script “predict\_co2.py”

```
import sys
import json
import requests
from datetime import datetime
import joblib
import numpy as np

def get_latest_data_from_influx(influx_config, sensor_id=None):
    host = influx_config["host"]
    db = influx_config["db"]

    if sensor_id:
        query = f"SELECT LAST(temperature), LAST(humidity), LAST(CO2) FROM
air_quality WHERE \"device\" = '{sensor_id}'"
    else:
        query = "SELECT LAST(temperature), LAST(humidity), LAST(CO2) FROM
air_quality"

    params = {
        "db": db,
        "q": query
    }
    try:
        response = requests.get(f"{host}/query", params=params)
        data = response.json()
        if "results" not in data or "series" not in data["results"][0]:
            print(json.dumps({"error": "No se encontraron datos para el sensor
indicado."}))
            sys.exit(1)

        results = data["results"][0]["series"][0]["values"][0]
        temperature = results[1]
        humidity = results[2]
        last_co2 = results[3]
        return temperature, humidity, last_co2
    except Exception as e:
        print(json.dumps({"error": f"Error al obtener datos de InfluxDB:
{str(e)}"}))
        sys.exit(1)

def map_quality_level(co2_value, thresholds):
```

```

if co2_value <= thresholds["buena"]:
    return "Buena"
elif co2_value <= thresholds["aceptable"]:
    return "Aceptable"
elif co2_value <= thresholds["pobre"]:
    return "Moderada"
elif co2_value <= thresholds["mala"]:
    return "Mala"
else:
    return "Muy mala"

def main():
    if len(sys.argv) < 2:
        print(" Debes pasar el archivo de entrada como argumento: python
predict_co2.py input.json")
        sys.exit(1)

    input_file = sys.argv[1]
    try:
        with open(input_file, 'r') as f:
            input_data = json.load(f)
    except Exception as e:
        print(f"Error al leer el archivo JSON: {e}")
        sys.exit(1)

    influx_config = input_data["influx_config"]
    thresholds = input_data["air_quality_thresholds"]
    sensor_id = input_data.get("sensor_id")

    if "forecast_temperature" in input_data:
        temperature = input_data["forecast_temperature"]
        _, humidity = get_latest_data_from_influx(influx_config)
    else:
        temperature, humidity, last_co2 =
get_latest_data_from_influx(influx_config, sensor_id=sensor_id)

    predict_at = input_data.get("predict_at") #(lunes=0, domingo=6)
    if predict_at:
        dt = datetime.fromisoformat(predict_at)
    else:
        dt = datetime.now()
    weekday = dt.weekday()
    month = dt.month

```

```

horizon = input_data.get("horizon_minutes", 60) # Por defecto: 1 hora
X = np.array([[temperature, humidity, month, weekday, horizon, last_co2]])

model =
joblib.load("C:/Users/efh_e/OneDrive/Documentos/MIOT/TFM/IA_Model/model_co2_rf_1F
C823F.joblib")
predicted_co2 = model.predict(X)[0]
predicted_quality = map_quality_level(predicted_co2, thresholds)

output = {
    "predicted_at": predict_at,
    "predicted_co2": round(float(predicted_co2), 2),
    "predicted_quality": predicted_quality,
    "input_used": {
        "temperature": temperature,
        "humidity": humidity,
        "month": month,
        "weekday": weekday
    }
}

print(json.dumps(output))
sys.exit(0)

if __name__ == "__main__":
    main()

```



## BIBLIOGRAFÍA

- ¿Qué es el NB-IoT? (s.f.). Obtenido de Telefonica Tech: <https://conectividaddiot.telefonicatech.com/que-es-el-nb-iot/>
- Calleja, A. H. (2006). *NTP 742: Ventilación general de edificios*. Instituto Nacional de Seguridad y Salud en el Trabajo (INSST).
- Cosio, N. A. (5 de 10 de 2021). *Medium*. Obtenido de Cómo seleccionar el mejor valor de K en K-means: <https://medium.com/@nicolasarrioja/c%C3%B3mo-seleccionar-el-mejor-valor-de-k-en-k-means-21121b604365>
- Device messages delay. (s.f.). Obtenido de Sigfox support: <https://support.sigfox.com/docs/device-messages-delay>
- DFRobot. (16 de 01 de 2026). *LPWAN in 2025: LTE-M vs NB-IoT vs LoRaWAN vs Sigfox*. Obtenido de DFRobot Drive the Future: <https://www.dfrobot.com/blog-17238.html>
- Difference between LoRaWan and Sigfox*. (15 de 07 de 2025). Obtenido de GeekforGeeks: <https://www.geeksforgeeks.org/computer-networks/difference-between-lorawan-and-sigfox/>
- Ditsuhi Iskandaryan, F. R. (1 de Abril de 2020). *Air Quality Prediction in Smart Cities Using Machine Learning Technologies Based on Sensor Data*. Obtenido de MDPI: <https://www.mdpi.com/2076-3417/10/7/2401>
- EPA. (23 de 06 de 2025). *Por qué la calidad del aire de los interiores es importante para las escuelas*. Obtenido de Agencia de Protección Ambiental de Estados Unidos: <https://espanol.epa.gov/cai/por-que-la-calidad-del-aire-de-los-interiores-es-importante-para-las-escuelas>
- España, U. (2024). *unaplatform Manual de Usuario*. Unabiz España.
- Etengoff, A. (30 de 10 de 2024). *Where are LPWAN protocols used?* Obtenido de 5G Technology World: <https://www.5gtechnologyworld.com/where-are-lpwan-protocols-used/>

- Galobart, N. (Septiembre de 2020). *Fontgas*. Obtenido de Calidad del aire interior y supervisión del CO<sup>2</sup>: <https://www.fontgas.com/blog/calidad-del-aire-y-supervision-del-co2/>
- Gast, M. S. (2015). *802.11ac - A Survival Guide*. Sebastopol, CA: O'Reilly Media.
- GeekforGeeks. (Julio de 2025). Obtenido de How to plot confusion matrix with labels in sklearn: <https://www.geeksforgeeks.org/machine-learning/how-to-plot-confusion-matrix-with-labels-in-sklearn/>
- Gomez D'Orazio, L. |. (2022). *Integración de una red de sensores con una plataforma IoT para control inteligente de aulas*. Buenos Aires: Facultad de Informática, Universidad Nacional de La Plata.
- Gracia Cancho, E. (07 de 2025). *Análisis de Datos CabinPaq y Evaluación de Modelos Predictivos de Índice de Calidad del Aire basados en Redes Neuronales*. Obtenido de Universidad Carlos III de Madrid Biblioteca: <https://hdl.handle.net/10016/47748>
- Gracia, L. M. (01 de 05 de 2020). *onesait platform Blog*. Obtenido de Redes IoT: SigFox vs LoRa: <https://blog.onesaitplatform.com/2020/05/01/redes-iot-sigfox-vs-lora/>
- GSMA. (2018). *LTE-M Deployment Guide*. GSMA Association.
- Huang, Y. (28 de 02 de 2024). *IoT de banda estrecha: todo lo que necesita saber sobre NB-IoT*. Obtenido de Moko Smart: <https://www.mokosmart.com/es/what-is-narrowband-iot-nb-iot/>
- Inc, I. (s.f.). *InfluxData*. Obtenido de IoT analytics and predictive maintenance: <https://www.influxdata.com/solutions/iot-analytics-and-predictive-maintenance/>
- INCIBE. (08 de 02 de 2024). *NB-IoT, la conexión ideal y de bajo consumo para IIoT*. Obtenido de INCIBE: <https://www.incibe.es/incibe-cert/blog/nb-iot-la-conexion-ideal-y-de-bajo-consumo-para-iiot>
- Inventions, C. (2024). *Connected Airwits CO2 Plus LoRa Datasheet*.
- Inventions, C. (2024). *Connected Airwits CO2 Plus LoRa User Manual*.

Jesus Sanchez-Gomez, D. G.-C.-P. (7 de 02 de 2020). *Secure Authentication and Credential Establishment in Narrowband IoT and 5G*. Obtenido de MDPI: <https://www.mdpi.com/1424-8220/20/3/882>

*LTE-M Tutorial: Network Architecture, Frequency Bands, and Stack*. (s.f.). Obtenido de RFWirelessWorld: <https://www.rfwireless-world.com/tutorials/lte-m-tutorial-network-architecture-frequency-bands-stack>

Moncho Perol, G. C. (2020). *ThinkBig*. Obtenido de ¿Qué es la tecnología OFDMA?: <https://blogthinkbig.com/tecnologia-ofdma/>

Node-RED. (Abril de 2020). *Node-RED Community*. Obtenido de Node-RED InfluxDB invalid boolean: <https://discourse.nodered.org/t/node-red-influxdb-invalid-boolean/8059>

Node-RED. (2023). *Node-RED Flows*. Obtenido de node-red-contrib-influxdb [Paquete de nodo]: <https://flows.nodered.org/node/node-red-contrib-influxdb>

Node-RED. (2025). *Node-RED Flows*. Obtenido de node-red-dashboard [Paquete de nodo]: <https://flows.nodered.org/node/node-red-dashboard>

*Partners*. (2025). Obtenido de unabiz: <https://www.unabiz.com/about-us/partners/>

Pindoria, A. (09 de 2018). *Understand the OSI model layers affected by the 802.11-2012 standard and amendments*. Obtenido de DOT11AP: <https://dot11ap.wordpress.com/understand-the-osi-model-layers-affected-by-the-802-11-2012-standard-and-amendments/>

Polina Levchenko, D. B. (8 de Diciembre de 2022). *Performance Comparison of NB-Fi, Sigfox, and LoRaWAN*. Obtenido de mdpi: <https://www.mdpi.com/1424-8220/22/24/9633>

RFWirelessWorld. (2025). *LTE-M Frequency Bands: Bands 2, 3, 4, 5, 8, 12, 13, 20, 28*. Obtenido de RFWirelessWorld: <https://www.rfwireless-world.com/terminology/lte-m-frequency-bands>

RFWirelessWorld. (2025). *RFWirelessWorld*. Obtenido de NB-IoT vs. LoRa vs. SigFox: A Detailed Comparison: <https://www.rfwireless-world.com/terminology/nb-iot-lora-sigfox-comparison>

Rodríguez, D. (9 de 06 de 2023). *Analytics Lane*. Obtenido de Método del codo (Elbow method) para seleccionar el número óptimo de clústeres en K-means: <https://www.analyticslane.com/2023/06/09/metodo-del-codo-elbow-method-para-seleccionar-el-numero-optimo-de-clusteres-en-k-means/>

Rodríguez, M. (25 de 01 de 2024). *Comparativa entre sigfox y lorawan*. Obtenido de Bo True Activities: <https://botrueactivities.com/comparativa-entre-sigfox-y-lorawan/>

Ron, T. (30 de 11 de 2016). *Davies-Bouldin Index*. Obtenido de Tom Ron: <https://tomron.net/2016/11/30/davies-bouldin-index/>

Scikit-learn. (s.f.). *scikit-learn*. Obtenido de RandomForestRegressor: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html>

Sigfox. (2023). *Sigfox connected objects: Radio specifications*.

Sigfox. (s.f.). *Sigfox support*. Obtenido de API Documentation: <https://support.sigfox.com/docs/api-documentation>

Sigfox. (s.f.). *Sigfox Support*. Obtenido de Custom callback creation: <https://support.sigfox.com/docs/custom-callback-creation>

*SigFox Tutorial: Network Architecture, Protocol, and MAC Frame*. (2025). Obtenido de RFWirelessWorld: <https://www.rfwireless-world.com/tutorials/sigfox-network-architecture-protocol-mac-frame>

*Techplayon*. (23 de 07 de 2020). Obtenido de NB-IoT Frequency Bands (As per 3GPP Rel. 13, 14 and 15): <https://www.techplayon.com/nb-iot-frequency-bands-as-per-3gpp-rel-13-14-and-15/>









