



Universidad Católica de Santa María

Facultad de Ciencias e Ingenierías Físicas y Formales

Escuela Profesional de Ingeniería Electrónica

**Diseño e implementación de un dispositivo de control de calidad de aire en
interiores usando internet de las cosas e inteligencia artificial**

Tesis presentada por:

Rivera Farfan, Juan Diego

ORCID: 0000-0002-8709-5975

para optar el Título Profesional de Ingeniero Electrónico con Especialidad en Automatización
y Control

Asesor:

Dr. Copa Pineda, Juan Carlos

ORCID: 0000-0002-8884-2039

Arequipa - Perú

2026

UNIVERSIDAD CATÓLICA DE SANTA MARÍA
INGENIERIA ELECTRONICA
CON ESPECIALIDAD EN AUTOMATIZACIÓN Y CONTROL
TITULACIÓN CON TESIS
DICTAMEN APROBACIÓN DE BORRADOR

Arequipa, 29 de Marzo del 2026

Dictamen: 011024-C-EPIE-2026

Visto el borrador del expediente 011024, presentado por:

2011223561 - RIVERA FARFAN JUAN DIEGO

Titulado:

**DISEÑO E IMPLEMENTACIÓN DE UN DISPOSITIVO DE CONTROL DE CALIDAD DE AIRE EN
INTERIORES USANDO INTERNET DE LAS COSAS E INTELIGENCIA ARTIFICIAL**

Nuestro dictamen es:

APROBADO

Titulo Profesional/Titulo de Segunda Especialidad/Grado Académico a optar:

**INGENIERO ELECTRONICO CON ESPECIALIDAD EN
AUTOMATIZACIÓN Y CONTROL**

**29364669 - QUISPE YAUYO JUAN MEDARDO
DICTAMINADOR**



**29329443 - VALDIVIESO HERRERA DIANA ISABEL
DICTAMINADOR**



**29410027 - SULLA TORRES RAUL RICARDO
DICTAMINADOR**



DISEÑO E IMPLEMENTACIÓN DE UN DISPOSITIVO DE CONTROL DE CALIDAD DE AIRE EN INTERIORES USANDO INTERNET DE LAS COSAS E INTELIGENCIA ARTIFICIAL

INFORME DE ORIGINALIDAD

14%

INDICE DE SIMILITUD

12%

FUENTES DE INTERNET

7%

PUBLICACIONES

7%

TRABAJOS DEL
ESTUDIANTE

FUENTES PRIMARIAS

1

zagan.unizar.es

Fuente de Internet

<1 %

2

Submitted to Universidad Católica San Pablo

Trabajo del estudiante

<1 %

3

Submitted to Universidad Católica de Santa
María

Trabajo del estudiante

<1 %

4

ouci.dntb.gov.ua

Fuente de Internet

<1 %

5

Submitted to Universidad Anahuac México
Sur

Trabajo del estudiante

<1 %

6

www.mdpi.com

Fuente de Internet

<1 %

7

juandomingofarnos.wordpress.com

Fuente de Internet

<1 %

8

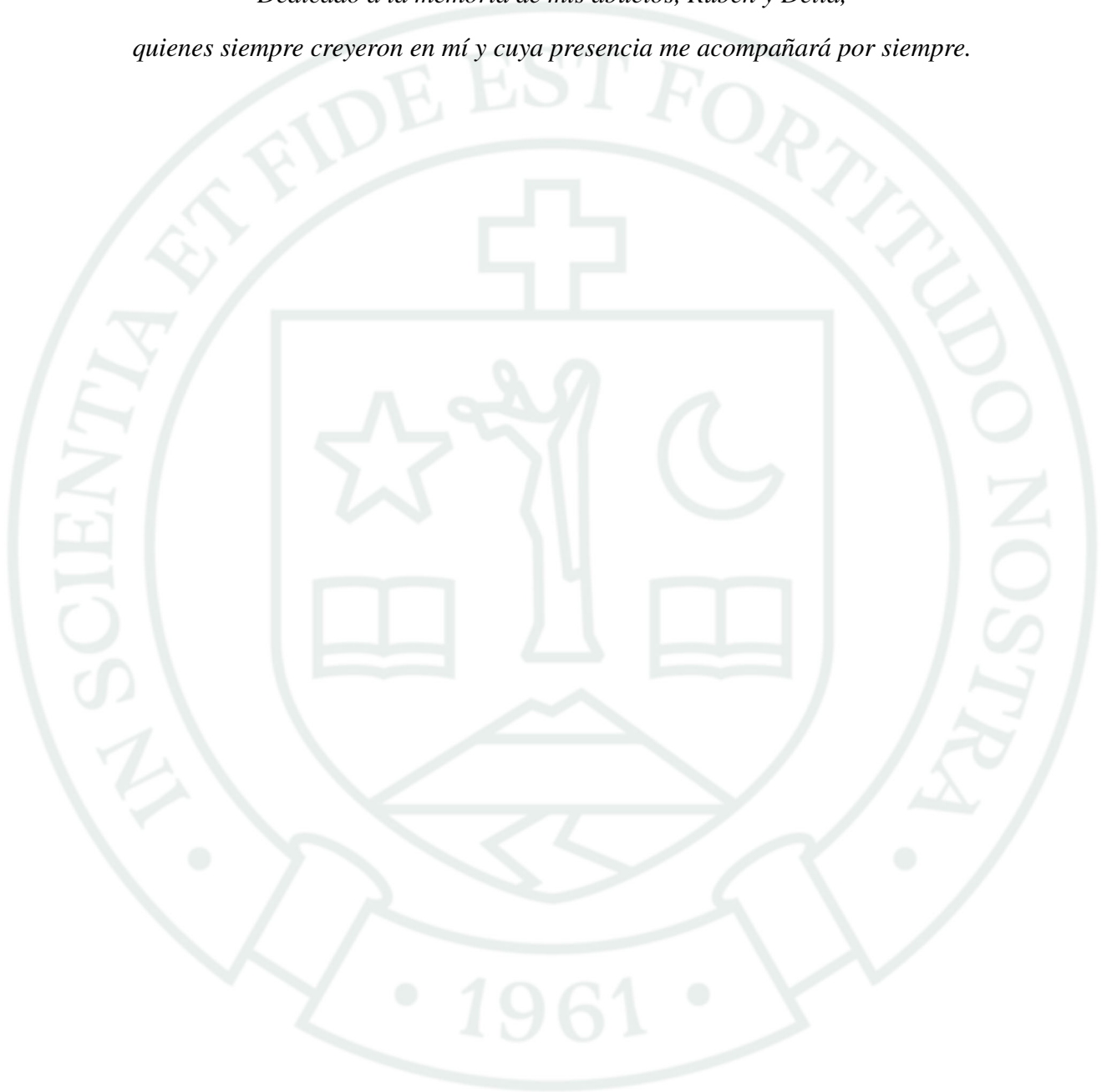
oa.upm.es

Fuente de Internet

<1 %

Dedicatoria

*Dedicado a la memoria de mis abuelos, Rubén y Delia,
quienes siempre creyeron en mí y cuya presencia me acompañará por siempre.*



Agradecimientos

A mis padres, **Julio y Nalda**, por su amor incondicional, su apoyo constante y por inculcarme el valor del esfuerzo. Sin ustedes, nada de esto habría sido posible.

A mi asesor de tesis, **Dr. Juan Carlos Copa Pineda**, por su guía y retroalimentación a lo largo del desarrollo del presente trabajo.

A mis amigos **Manuel, Álvaro, Sergio, Iván y Maryelin**, compañeros de camino durante los años universitarios, por compartir momentos de estudio, aprendizaje y fraternidad.

Y muy especialmente, a **Gustavo**, cuya amistad, alegría y sabiduría permanecen conmigo. Su recuerdo siempre será una inspiración para todos los que tuvimos la suerte de ser su amigo.

A cada uno de ustedes, gracias por ser parte de este logro.

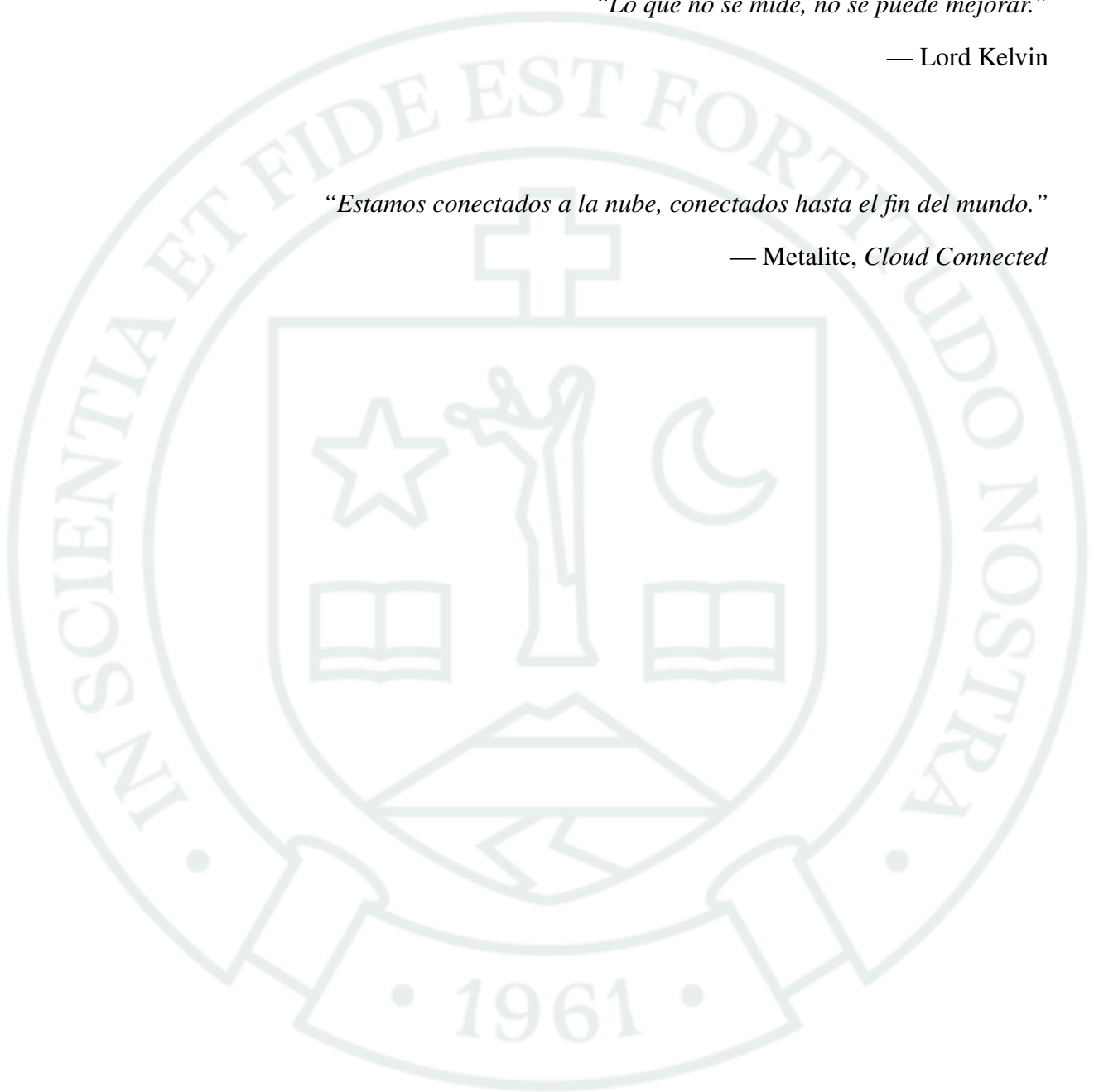
Epígrafe

“Lo que no se mide, no se puede mejorar.”

— Lord Kelvin

“Estamos conectados a la nube, conectados hasta el fin del mundo.”

— Metalite, *Cloud Connected*



Resumen

La calidad del aire interior es uno de los principales factores de salud y el confort humano, sobre todo en ambientes cerrados donde la ventilación natural es limitada. La necesidad de monitoreo y evaluación de las condiciones del aire en tiempo real surge del aumento de contaminantes provenientes de fuentes domésticas, industriales o biológicas. Abordar este problema es interesante debido a su impacto en el bienestar y productividad de las personas, además es un potencial objetivo de aplicación de tecnologías inteligentes dentro del marco del Internet de las Cosas.

En el presente trabajo se presenta el desarrollo e implementación de un prototipo de medición y predicción del Índice de calidad de aire (IAQ), compuesto por sensores de gases contaminantes, un microcontrolador ESP32 para adquisición de mediciones y control de la calidad de aire, y una microcomputadora Raspberry pi que integra un modelo de inteligencia artificial capaz de estimar el IAQ. Además, la Raspberry pi actúa como Gateway, gestionando el almacenamiento local y el envío de información a un servidor remoto(nube). En el servidor se implementó la plataforma web Thingsboard de código abierto, que permite visualizar en tiempo real las mediciones y predicciones del sistema, facultando al usuario, acceder a los datos en cualquier momento y lugar, mediante una conexión a internet. Desde la plataforma web se envían alertas por correo electrónico cuando se alcanzan valores peligrosos de calidad de aire, facilitando la interpretación de los resultados y la rápida toma de decisiones.

Los resultados obtenidos demuestran que el sistema puede predecir con alto nivel de correlación los valores de IAQ, mostrando una concordancia moderadamente fuerte entre los valores reales y los estimados por el modelo. Así, se concluye que el prototipo propuesto permite una evaluación confiable del ambiente interior y sienta las bases para futuras implementaciones o integración con infraestructuras de internet de las cosas e inteligencia artificial orientadas al monitoreo ambiental.

Palabras Clave: IoT, nube, IAQ.

Abstract

Indoor air quality is one of the main factors affecting human health and comfort, especially in enclosed environments where natural ventilation is limited. The need for real-time monitoring and evaluation of air conditions arises from the increase in pollutants from domestic, industrial, or biological sources. Addressing this problem is interesting due to its impact on people's well-being and productivity. It is also a potential target for the application of smart technologies within the framework of the Internet of Things.

This work presents the development and implementation of a prototype for measuring and predicting the Air Quality Index (IAQ). This prototype is composed of pollutant gas sensors, an ESP32 microcontroller for measurement acquisition and air quality control, and a Raspberry Pi microcomputer that integrates an artificial intelligence model capable of estimating IAQ. Furthermore, the Raspberry Pi acts as a gateway, managing local storage and sending information to a remote server (cloud).. The open-source Thingsboard web platform was implemented on the server. It allows real-time visualization of the system's measurements and predictions, allowing users to access the data anytime, anywhere via an internet connection. Email alerts are sent from the web platform when dangerous air quality values are reached, facilitating the interpretation of the results and rapid decision-making.

This results obtained demonstrate that the system can predict IAQ values with a high level of correlation, showing a moderately strong agreement between the actual values and those estimated by the model. Thus, it is concluded that the proposed prototype allows for a reliable assessment of the indoor environment and lays the groundwork for future implementations or integration with IoT and artificial intelligence infrastructures for environmental monitoring.

Keywords: IoT, cloud, IAQ.

Índice general

Dedicatoria

Agradecimientos

Epígrafe

Resumen

Abstract

Introducción 1

1. Planteamiento 3

1.1. Planteamiento del problema 4

1.1.1. Variables de investigación 4

1.2. Objetivos 4

1.2.1. Objetivo general 4

1.2.2. Objetivos específicos 4

1.3. Metodología 5

1.3.1. Línea de investigación 5

1.3.2. Tipo de investigación 5

1.3.3. Métodos para experimentos y pruebas 5

1.3.4. Formas de comprobación 6

1.4. Consideraciones Complementarias 7

1.4.1. Recursos y Materiales 7

2. Marco Teórico 9

2.1. Descripción de ambientes interiores 10

2.1.1.	Cocina Industrial	10
2.1.2.	Laboratorio de mantenimiento industrial	11
2.1.3.	Taller Automotriz	13
2.1.4.	Gimnasio	14
2.2.	Contaminantes de aire interior	15
2.2.1.	Benceno	16
2.2.2.	Monóxido de carbono	16
2.2.3.	Formaldeído	17
2.2.4.	Naftaleno	18
2.2.5.	Dióxido de nitrógeno(NO_2)	18
2.2.6.	Hidrocarburos Aromáticos Policíclicos (HAP)	18
2.2.7.	Radón	19
2.2.8.	Tricloroetileno(C_2HCl_3)	19
2.2.9.	Tetracloroetileno(C_2Cl_4)	20
2.2.10.	Humedad y moho	20
2.2.11.	Temperatura	21
2.2.12.	Partículas de materia (PM)	21
2.2.13.	Compuestos orgánicos volátiles (VOC)	21
2.2.14.	Dióxido de carbono(CO_2)	22
2.3.	Tipos de sensores de calidad de aire	23
2.3.1.	Sensores electroquímicos	23
2.3.2.	Sensores Fotoeléctricos	23
2.4.	Actuadores y accionamientos para calidad de aire	24
2.5.	Sistemas Inteligentes de IoT	25
2.6.	Internet de las Cosas	26
2.6.1.	Niveles de Internet de las Cosas	27
2.7.	Inteligencia Artificial aplicada en IAQ	33
2.8.	Algoritmos de predicción	35

2.8.1.	Regresión Lineal	35
2.8.2.	Árboles de Decisión	36
2.8.3.	Regresión Lasso	37
2.8.4.	Redes Neuronales Artificiales	38
2.9.	Estado del Arte	39
2.9.1.	Recopilación de trabajos sobre el uso de las tecnologías de bajo costo para el monitoreo de calidad de aire en interiores.	40
2.9.2.	Implementación de sistema de largo alcance basado en LoRaWAN para monitoreo de calidad de aire	41
2.9.3.	Modelado de partículas de materia PM2.5 con Árbol del sistema de inferencia difusa, dinámica adaptativa y red de sensores IoT	42
2.9.4.	Monitoreo de calidad de aire interior para mejorar la eficiencia de los sistemas de ventilación usando Redes neuronales convolucionales total- mente conectadas.	43
2.9.5.	Plataforma modular IoT para monitoreo de calidad de aire interior en tiempo real	44
2.9.6.	Detector de IAQ con diferentes interfaces de comunicación basado en IoT	45
2.9.7.	Deep learning para predicción de calidad de aire en interiores	46
2.9.8.	Medidor portátil de IAQ con sensores de bajo costo y plataforma IoT de código abierto	47
2.9.9.	Prevención de COVID-19 con un sistema inteligente basado en IoT para el control de calidad de aire	48
2.9.10.	Recopilación de trabajos sobre el impacto del IAQ en ambientes críticos	49
2.9.11.	Simulación del índice de calidad de aire y ventilación en edificios co- merciales e institucionales	50
2.9.12.	Análisis de métodos de machine learning para predicción de IAQ	51
2.9.13.	Apreciaciones Finales	54

3. Análisis de ambientes representativos y su relación con la calidad de aire interior 58

3.1. Cocina Industrial	59
3.1.1. Aspectos arquitectónicos	59
3.1.2. Aspectos ambientales	60
3.1.3. Monitoreo y estrategias de control del IAQ	60
3.1.4. Impacto del IAQ en el desempeño	61
3.2. Laboratorio de mantenimiento industrial	62
3.2.1. Aspectos arquitectónicos	62
3.2.2. Aspectos ambientales	63
3.2.3. Monitoreo y estrategias de control del IAQ	63
3.2.4. Impacto del IAQ en el desempeño	63
3.3. Taller automotriz	64
3.3.1. Aspectos arquitectónicos	64
3.3.2. Aspectos ambientales	65
3.3.3. Monitoreo y estrategias de control del IAQ	65
3.3.4. Impacto del IAQ en el desempeño	65
3.4. Gimnasio	66
3.4.1. Aspectos arquitectónicos	66
3.4.2. Aspectos ambientales	67
3.4.3. Monitoreo y estrategias de control del IAQ	67
3.4.4. Impacto del IAQ en el desempeño	67
4. Diseño e implementación del prototipo	68
4.1. Selección de sensores	70
4.2. Adquisición, filtrado y almacenamiento de datos	73
4.3. Envío de datos a la nube	74
4.4. Cálculo del índice de calidad de aire interior	75
4.5. Entrenamiento y evaluación de algoritmos de IA	77
4.6. Implementación del hardware	79
4.6.1. Diseño y ensamble del circuito	79

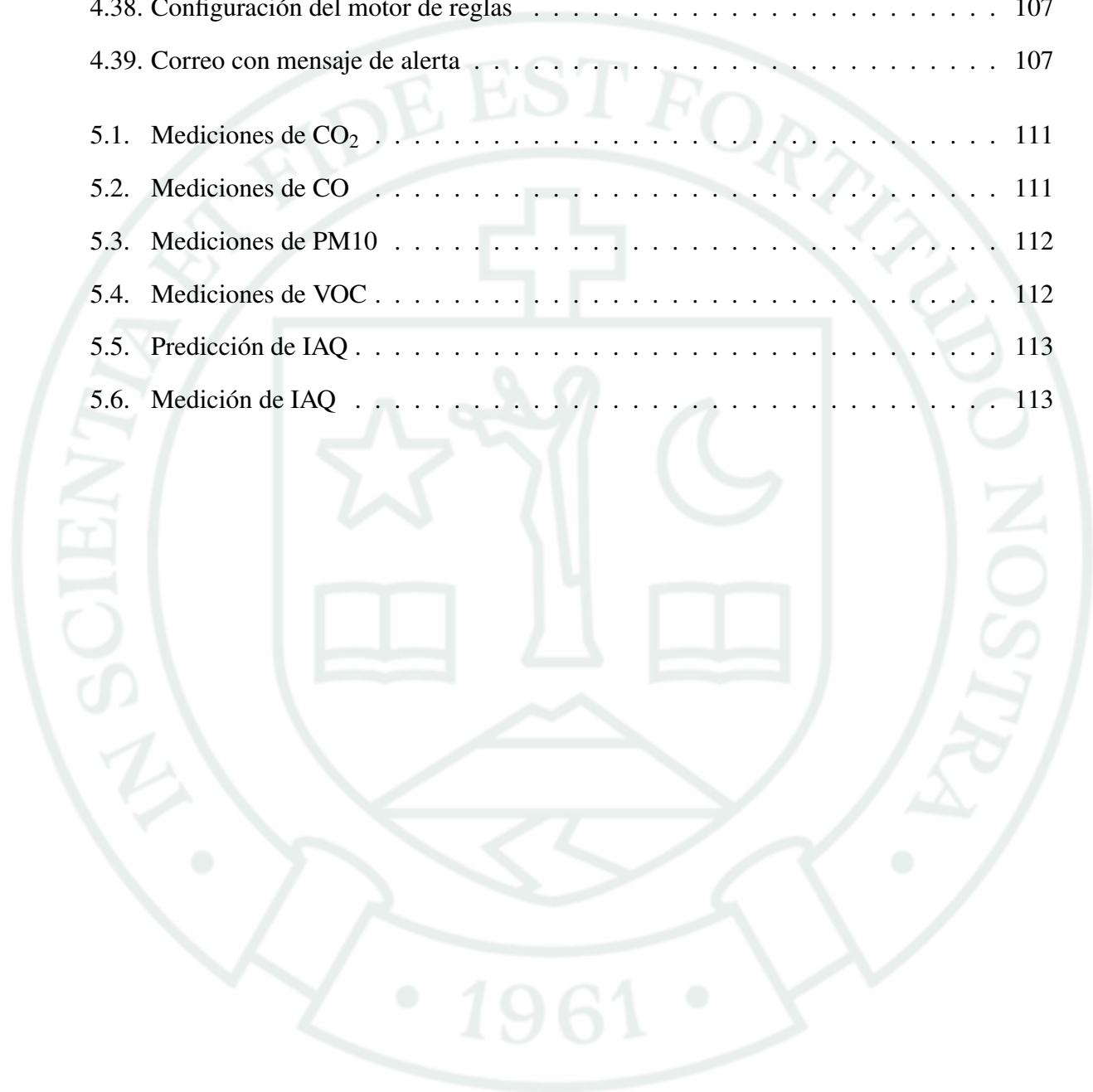
4.6.2. Funcionamiento del sistema	81
4.7. Algoritmos de IA aplicados	85
4.7.1. Selección de algoritmos	85
4.7.2. Entrenamiento y validación	97
4.8. Computación en la nube	105
4.8.1. Arquitectura de la plataforma	105
4.8.2. Comunicación y transmisión de datos	105
4.8.3. Desarrollo de la plataforma web	106
4.9. Costos del prototipo	107
4.10. Especificaciones técnicas del prototipo	108
5. Pruebas experimentales y resultados	109
5.1. Metodología de las pruebas experimentales	110
5.2. Resultados obtenidos	110
5.3. Análisis de congruencia de IAQ predicho vs ENS160	114
5.4. Discusión de resultados	114
5.5. Limitaciones	115
Conclusiones	116
Referencias	119

Índice de figuras

1.1. Diagrama de Bloques del Sistema	6
2.1. Contaminantes que afectan la calidad del aire interior	22
2.2. Sensor óxido metálico	23
2.3. Tipos de sensor fotoeléctrico	24
2.4. Arquitectura simplificada de IoT	26
2.5. Nivel 1	28
2.6. Nivel 2	29
2.7. Nivel 3	30
2.8. Nivel 4	31
2.9. Nivel 5	32
2.10. Nivel 6	33
2.11. Capas de una red neuronal	38
3.1. Plano de la cocina industrial	59
3.2. Plano del laboratorio	62
3.3. Plano del taller	64
3.4. Plano del gimnasio	66
4.1. Adquisición de datos	73
4.2. Procesamiento de datos en la nube	74
4.3. Control automático de la ventilación	78
4.4. Esquema del circuito electrónico	79
4.5. Implementación del circuito	80
4.6. Ensamblaje final del prototipo dentro de un gabinete 3D	81
4.7. Funcionamiento del sistema: etapa de sensado	82
4.8. Funcionamiento del sistema: etapa de recepción de datos en Raspberry Pi	83

4.9. Índice de IAQ para VOC	86
4.10. Índice de IAQ para PM10	87
4.11. Índice de IAQ para NO ₂	87
4.12. Índice de IAQ para CO	88
4.13. Índice de IAQ para CO ₂	88
4.14. Índice de IAQ para VOC	89
4.15. Índice de IAQ para PM10	89
4.16. Índice de IAQ para NO ₂	90
4.17. Índice de IAQ para CO	90
4.18. Índice de IAQ para CO ₂	91
4.19. Índice de IAQ para VOC	91
4.20. Índice de IAQ para PM10	92
4.21. Índice de IAQ para NO ₂	92
4.22. Índice de IAQ para CO	93
4.23. Índice de IAQ para CO ₂	93
4.24. Índice de IAQ para VOC	94
4.25. Índice de IAQ para PM10	94
4.26. Índice de IAQ para NO ₂	95
4.27. Índice de IAQ para CO	95
4.28. Índice de IAQ para CO ₂	96
4.29. Generación de datos sintéticos para entrenamiento	96
4.30. Cálculo de hiperparámetros de la red neuronal	97
4.31. Calibración de hiperparámetros	98
4.32. Estructura del perceptrón para obtención de hiperparámetros	100
4.33. Calibración de número de neuronas y capas	100
4.34. Reducción del error medio absoluto (MAE) para distintas configuraciones de neuronas en la primera capa	101

4.35. Reducción del error medio absoluto (MAE) para distintas configuraciones de neuronas en la segunda capa	102
4.36. Comparativa de funciones de activación	103
4.37. Panel de monitoreo en ThingsBoard	106
4.38. Configuración del motor de reglas	107
4.39. Correo con mensaje de alerta	107
5.1. Mediciones de CO ₂	111
5.2. Mediciones de CO	111
5.3. Mediciones de PM10	112
5.4. Mediciones de VOC	112
5.5. Predicción de IAQ	113
5.6. Medición de IAQ	113

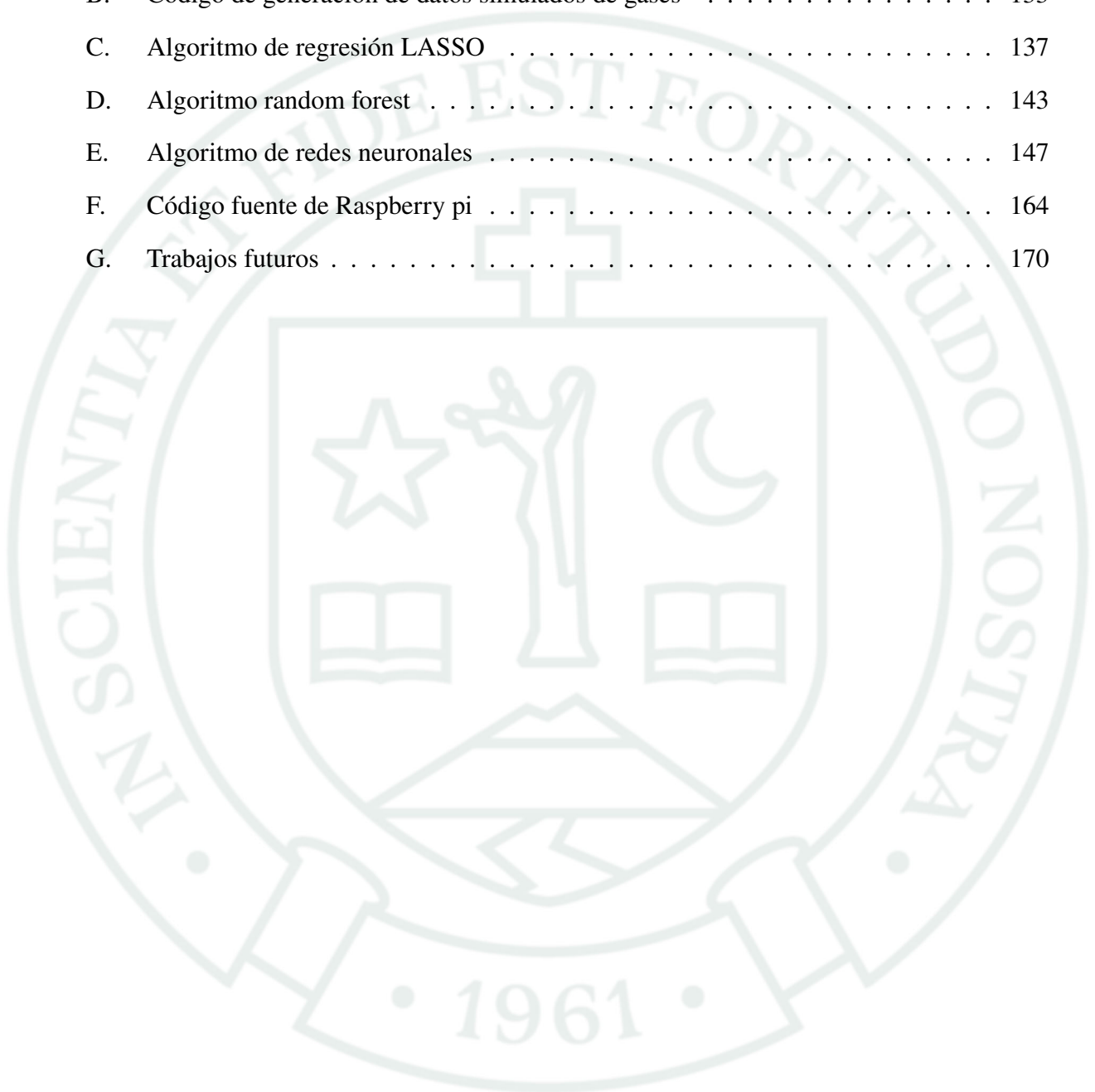


Índice de tablas

4.1. Parámetros del sensor ENS160	71
4.2. Parámetros del sensor de partículas Plantower PMS5003	71
4.3. Parámetros del sensor AHT21	72
4.4. Parámetros del sensor ZE15-CO	72
4.5. Parámetros del sensor MH-Z19B	73
4.6. Clasificación de calidad de aire EPA	75
4.7. Clasificación de calidad de aire interior	77
4.8. Umbrales estándar y con histéresis para transiciones de velocidad del ventilador	84
4.9. Hiperparámetros de la red neuronal. Programas de calibración del número de capas e hiperparámetros. Elaboración propia	103
4.10. Aplicación del modelo de Red neuronal en los ambientes	104
4.11. Coeficientes del regresor LASSO	104
4.12. Coeficientes de los árboles de decisión	105
4.13. Costo de los componentes del prototipo	108
4.14. Especificaciones técnicas del prototipo de medición de IAQ	108
5.1. Tabla de equivalencia de niveles e índice de IAQ	110

Índice de anexos

A.	Código fuente del ESP32	124
B.	Código de generación de datos simulados de gases	133
C.	Algoritmo de regresión LASSO	137
D.	Algoritmo random forest	143
E.	Algoritmo de redes neuronales	147
F.	Código fuente de Raspberry pi	164
G.	Trabajos futuros	170



Índice de algoritmos

1.	Filtro de Media Móvil	74
2.	Algoritmo para generación de valores de contaminantes e IAQ	78



Introducción

En los últimos tiempos, sucesos como el calentamiento global, la superpoblación, el aumento del parque automotor y la más reciente pandemia de coronavirus; han causado que mejorar la calidad de aire que respiramos sea considerada un desafío importante a nivel mundial. Aunque es más común relacionar la contaminación del aire con el exterior, los niveles que se pueden encontrar en hogares, oficinas, hospitales, entre otros; no deben despreciarse debido a que las personas pasan más tiempo en ambientes interiores.

La Organización Mundial de la Salud(WHO) reporta que el aire contaminado causa 7 millones de muertes anualmente (World Health Organization, 2018), de las cuales 3.6 millones son causadas por exposición a contaminación de aire doméstico (World Health Organization, 2022). Por otro lado, la Agencia de Protección Ambiental de Estados Unidos(USEPA) ha informado de los efectos a corto y largo plazo que pueden afectar la salud de una persona en contacto directo con agentes nocivos(U.S. Environmental Protection Agency, 2023), que van desde irritación nasal u ocular hasta enfermedades cardíacas y cáncer.

Debido a las potenciales consecuencias de la constante exposición a aire contaminado, es necesario el monitoreo constante de los posibles agentes nocivos existentes en viviendas, lugares de trabajo o centros de salud.

En la actualidad, la creciente expansión de la tecnología de internet de las cosas (Internet of things,IoT) permite conocer toda clase de datos recolectados de diferentes dispositivos (Zanella et al., 2014). Los sensores de bajo costo han facilitado la tarea de medir parámetros de calidad de aire en interiores (Indoor Air Quality,IAQ)(Sá et al., 2022) y junto al IoT, los usuarios finales pueden conocer más sobre el entorno en que se encuentran, lo que significa que ambas tecnologías son una herramienta potencial para mejorar el bienestar de las personas.

Los grandes volúmenes de datos que se pueden recolectar en un sistema IoT requieren de técnicas de análisis para poder interpretar correctamente su significado, es en este punto donde la inteligencia artificial se convierte en el instrumento fundamental para conseguir este

propósito (Saleem & Chishti, 2021). Los dispositivos de monitoreo de calidad del aire interior no son ajenos a la demanda de manejar vastas cantidades de datos, ya que de manera constante recopilan información de diversos sensores encargados de medir una variedad de elementos contaminantes y condiciones ambientales (Sá et al., 2022).

El propósito de este estudio surge de la importancia de supervisar y examinar la calidad del aire en ambientes interiores, con el fin de brindar a las personas la posibilidad de contar con un aire adecuado dentro de los espacios cerrados en los que se encuentren.





CAPÍTULO I

Planteamiento

1.1 Planteamiento del problema

La continua medición de los elementos contaminantes en el aire y la anticipación de los futuros niveles de calidad del aire interior, basados en las concentraciones actuales, son componentes esenciales en numerosos estudios. Varios de estos trabajos tienen como objetivo demostrar cómo la combinación del Internet de las Cosas (IoT) y la inteligencia artificial aborda este desafío, con el propósito de mantener a las personas constantemente informadas sobre la calidad del aire que respiran. En caso de que los indicadores monitoreados alcancen niveles críticos, las personas pueden tomar las medidas pertinentes, todas las investigaciones revisadas solo llegan al punto de informar al usuario final y dejan a su criterio las acciones requeridas. El enfoque que se propone es tomar los valores medidos y utilizarlos para automatizar las acciones necesarias para la purificación del aire.

1.1.1 Variables de investigación

1. Variables independientes: Temperatura, humedad, CO, CO₂, PM10 y VOC.
2. Variables dependientes: Índice de calidad de aire y control del sistema de regulación.

1.2 Objetivos

1.2.1 Objetivo general

Diseño e implementación de un sistema de monitoreo y control de calidad de aire en interiores basado en internet de las cosas e inteligencia artificial.

1.2.2 Objetivos específicos

- Describir y documentar ambientes representativos(tamaño, ocupación, actividades, ventilación) para la aplicación y prueba del dispositivo.
- Seleccionar e integrar los sensores y dispositivos necesarios, así como diseñar los elementos físicos y electrónicos requeridos para la implementación del prototipo de bajo costo.

- Diseñar y desarrollar un proceso de simulación de datos de calidad del aire, definiendo segmentos temporales, rangos de concentración y niveles de calidad del aire para generar mediciones sintéticas de los sensores.
- Entrenar diferentes algoritmos de inteligencia artificial y evaluar su capacidad predictiva.
- Implementar un sistema de monitoreo en la nube integrado con un mecanismo local de respuesta automática proporcional al índice de calidad del aire, complementado con un sistema de alertas remotas (SMTP) para valores extremos.
- Realizar pruebas experimentales con diferentes fuentes de contaminación y analizar el desempeño del sistema, comparándolo con un sensor de referencia, evaluando la correlación en la predicción del índice de calidad del aire.

1.3 Metodología

1.3.1 Línea de investigación

El presente trabajo se ubica en la línea de investigación en control y automatización de la Escuela Profesional de Ingeniería Electrónica.

1.3.2 Tipo de investigación

La presente investigación se enmarca dentro del desarrollo científico y tecnológico, dado que se pretende el diseño y fabricación de un módulo de medición y control de calidad de aire empleando las tecnologías de internet de las cosas e inteligencia artificial, lo cual conlleva el estudio de diversos artículos científicos para establecer diferencias, ventajas y desventajas de lo propuesto en este proyecto así mismo se debe revisar las hojas de datos o manuales de los dispositivos a utilizar.

1.3.3 Métodos para experimentos y pruebas

La metodología del presente documento se ajusta a los siguientes lineamientos:

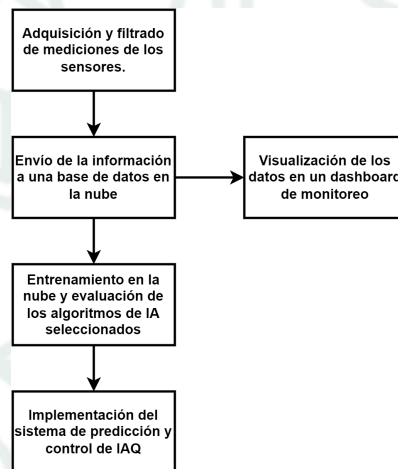
1. Se realizará una búsqueda bibliográfica para determinar los sensores más idóneos para la experimentación.
2. Implementar un sistema IoT nivel 3 para recolección de datos, monitoreo y control.
3. Evaluar diferentes algoritmos de IA a fin de optimizar el control de calidad de aire.

1.3.4 Formas de comprobación

La comprobación se hará a través de la simulación de valores de calidad de aire desde normales a críticos y la activación del sistema de regulación cuando estos últimos son alcanzados, esto permitirá caracterizar el dispositivo y validar su correcto funcionamiento.

Esquema Conceptual. El diagrama de bloques de la figura 1.1 muestra las etapas de implementación del presente trabajo de investigación.

Figura 1.1
Diagrama de Bloques del Sistema



Nota. Elaboración propia.

La descripción de cada bloque es:

- **Adquisición y filtrado de mediciones de los sensores:** Cada sensor mide las señales correspondientes del ambiente y estas se enviarán de manera analógica o digital al Raspberry Pi, pero en ambos casos es adecuado realizar filtrado de posibles cambios abruptos.

tos producto de ruido aleatorio, para esto se aplicará un filtro de medio móvil rápido a las señales recibidas.

- Envío de la información a una base de datos en la nube: Para esto se hará uso de un servidor en la nube, el que recibirá y almacenará la información.
- Dashboard de monitoreo: En el servidor también se implementará un dashboard en el que se observará todos los valores medidos por los sensores.
- Entrenamiento en la nube y evaluación de los algoritmos de IA seleccionados: Las plataformas Google Colab y Databricks serán la herramientas en la que se hará el entrenamiento y evaluación de los algoritmo.
- Implementación del sistema de predicción y control de IAQ: Cuando se ha determinado el mejor algoritmo de predicción, este se aplicará en el control del sistema de ventilación.

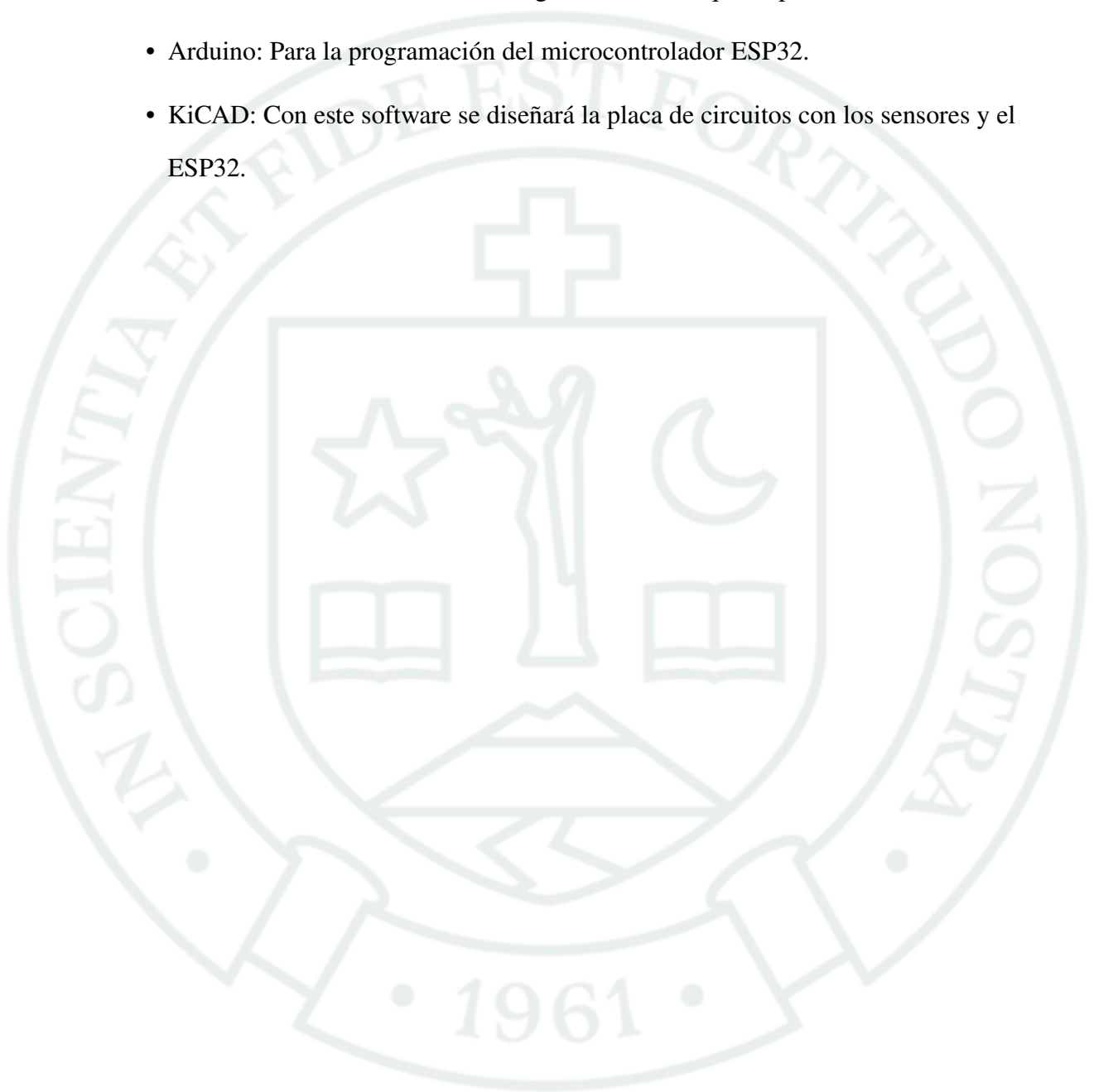
1.4 Consideraciones Complementarias

1.4.1 Recursos y Materiales

- **Hardware:**
 - Computadora: Procesamiento local de los datos y programación.
 - Servidor en la nube: El monitoreo de los valores medidos puede ser observado por el usuario de manera remota.
 - Sensores: Necesarios para la medición de los diferentes gases en el aire.
 - Raspberry Pi: Esta microcomputadora va a desempeñar el rol de *gateway* entre el sistema local y la nube, enviando los datos a la plataforma web. Otra de sus funciones será calcular la predicción del IAQ con el modelo de IA que se haya entrenado.
 - Espressif ESP32: Con el microcontrolador ESP32 se hará la lectura de datos proporcionados por los sensores; también se encargará del control de los actuadores que se seleccionen para la regulación de la calidad de aire.

■ **Software:**

- Python: Todos los algoritmos a desarrollar serán programados con este lenguaje.
- TensorFlow y PySpark: Con la ayuda de estas librerías se desarrollará la implementación de los modelos de inteligencia artificial que se plantearán.
- Arduino: Para la programación del microcontrolador ESP32.
- KiCAD: Con este software se diseñará la placa de circuitos con los sensores y el ESP32.





CAPÍTULO II

Marco Teórico

2.1 Descripción de ambientes interiores

La magnitud del IAQ en interiores varía acorde a los posibles contaminantes que puedan encontrarse en estos lugares, cada tipo de ambiente puede contener los mismos gases, pero en concentraciones diferentes, por lo que la gravedad de la exposición de los ocupantes está sujeta al tiempo, intensidad y tipo de gas.

A continuación, se describe cuatro diferentes recintos en los que se tiene exposición a monóxido de carbono (CO), dióxido de carbono (CO₂), dióxido de nitrógeno (NO₂), compuestos orgánicos volátiles (VOC) y partículas de materia, además de temperatura y humedad como indicadores de confort. Los apartados para tener en cuenta para cada uno serán: arquitectura, cantidad de personas, predominancia de gases y partículas, horarios y fluctuación de los contaminantes.

2.1.1 Cocina Industrial

Arquitectura.

- **Tamaño y estructura:** El área será de 80m², diseñado para preparación de grandes cantidades de alimentos. El recinto está dividido en zonas de cocción y preparación, despensas y limpieza.
- **Equipos y herramientas:** Algunos equipos de uso habitual son hornos, cocinas a gas, freidoras, refrigeradores y campanas extractoras. Los elementos de limpieza comunes son detergentes, quitamanchas, desengrasantes y desinfectantes.
- **Ventilación:** Extractores de gases y vapores sobre las cocinas. Ventilación forzada y ventanas que permitan el paso natural de aire fresco.

Aforo. El lugar se ha pensado para un aforo de 5 a 10 personas entre cocineros, ayudantes de cocina, personal de limpieza y repartidores.

Fuentes de gases y predominancia.

- **Dióxido de carbono:** La principal fuente es la respiración de las personas. Por los me-

dios existentes de flujo de gases y la cantidad de personas, su predominancia es media a baja.

- Monóxido de carbono: Todos los procesos de combustión (hornos, estufas, quemadores) están asociados a la liberación de CO. Su predominancia es alta.
- Dióxido de nitrógeno: También relacionado con los procesos de combustión, su predominio es alto.
- Compuestos Orgánicos Volátiles: Generado por vapores, combustión y productos de limpieza, también se considera una predominancia elevada por su fuerte relación con el NO₂.
- Partículas de materia: Los humos de cocción de parrillas y freidoras, son sus principales fuentes. La influencia sobre el AIQ es media.

Horarios y fluctuación de gases.

- Mañana (5:00 a 10:00): Preparaciones iniciales; todos los gases están en concentraciones bajas, pero aumentando, principalmente CO y NO₂.
- Mediodía (10:00-15:00): Altas concentraciones de CO, NO₂ y VOC, CO₂ y PM en nivel medio a bajo.
- Tarde (15:00-17:00): Reducción de las actividades de cocina y aumento de las de limpieza, VOC y NO₂ alto a medio.
- Resto del día: Concentraciones bajas por la ventilación y el fin de actividades.

2.1.2 Laboratorio de mantenimiento industrial

Arquitectura.

- Tamaño y estructura: Posee un área de 150m², techo alto para almacenamiento y sistemas de ventilación. Sectores para reparación y mantenimiento, almacén de herramientas y pruebas de máquinas.

- Equipos y herramientas: Taladros, esmeriles, amoladoras, llaves, martillos, etc. Motores, compresores, generadores. Estaciones de soldadura, pintura o corte.
- Ventilación: Extractores en las áreas de soldadura. Ventiladores de nivel industrial y ventanales.

Aforo. El laboratorio se ha pensado para albergar de 10 a 15 personas entre técnicos de mantenimiento, soldadura y operadores de maquinaria, personal de limpieza y supervisores. En cada turno hay un máximo de 7 personas.

Fuentes de gases y predominancia.

- Dióxido de carbono: Respiración de las personas y algunos procesos de combustión. Las concentraciones son bajas a moderadas.
- Monóxido de carbono: Las principales fuentes son motores de combustión y estaciones de soldadura. Las concentraciones pueden llegar a ser insalubres en periodos de máxima actividad.
- Dióxido de nitrógeno: También surge de los procesos de soldadura, pintura o motores. La presencia de este compuesto alcanza la escala de insalubre durante las horas de mayor actividad.
- Compuestos Orgánicos Volátiles: Surgen de productos químicos como pinturas, disolventes, combustible, lubricantes y otros compuestos de uso común en ambientes de este tipo. Las concentraciones de VOC pueden llegar a niveles insalubres.
- Partículas de materia: Generados por residuos de lijado, soldadura, corte, polvo de materiales o por maquinarias, su concentración también llega a ser insalubre.

Horarios y fluctuación de gases.

- Mañana (6:00-10:00): Inicio de actividades, CO₂ y VOC en niveles bajos; el resto de los gases comienza a aumentar.

- Mediodía (10:00-14:00): Horas de mucha actividad, se alcanzan los valores más altos de concentración rápidamente.
- Tarde (14:00-16:00): Durante el periodo de descanso, el valor de IAQ es bueno, luego aumenta, pero por el fin de operaciones estos valores son en promedio de intensidad media con algunos picos de insalubridad.
- Resto del día: Todas las concentraciones de gases se reducen al mínimo.

2.1.3 Taller Automotriz

Arquitectura.

- Tamaño y estructura: El área es de 300 m², con capacidad para 5 vehículos. Está dividido en diferentes áreas de trabajo: mecánica general, diagnóstico y limpieza.
- Equipos y herramientas: Taladros, esmeriles, amoladoras, llaves, martillos. Elevadores hidráulicos, compresoras de aire, analizadores de gases.
- Ventilación: Sistemas de extracción en áreas de mecánica general y reparación. Grandes puertas que permiten el ingreso de aire exterior, aunque la ventilación puede ser insuficiente.

Aforo. El taller se ha pensado para albergar de 5 a 10 personas, entre técnicos mecánicos y clientes, personal de limpieza y supervisores.

Fuentes de gases y predominancia.

- Dióxido de carbono: Respiración de las personas y algunos procesos de combustión. Las concentraciones son bajas a moderadas.
- Monóxido de carbono: Las principales fuentes son motores de combustión. Las concentraciones pueden llegar a ser insalubres en periodos de máxima actividad.
- Dióxido de nitrógeno: Surge de los humos de los vehículos, disolventes, combustibles,

etc. La presencia de este compuesto alcanza la escala de insalubre durante las horas pico.

- **Compuestos Orgánicos Volátiles:** Principalmente, se genera por los vehículos en reparación, pero también de productos químicos como disolventes, combustible, lubricantes y otros compuestos de uso común en ambientes de este tipo. Las concentraciones de VOC pueden llegar a niveles insalubres.
- **Partículas de materia:** Generados por residuos de lijado, soldadura, corte, polvo de materiales y humo de los vehículos, su concentración también llega a ser insalubre.

Horarios y fluctuación de gases.

- **Mañana (6:00-10:00):** Inicio de operaciones. Se hacen reparaciones menores, CO, VOC, PM10 en valores moderados.
- **Mediodía (10:00-14:00):** Horas de mucha actividad, se alcanzan los valores más altos de concentración rápidamente.
- **Tarde (14:00-16:00):** Durante el periodo de descanso, el valor de IAQ es bueno, luego aumenta, pero por el fin de operaciones estos valores son en promedio de intensidad media con algunos picos de insalubridad.
- **Resto del día:** Todas las concentraciones de gases se reducen al mínimo y hay un pequeño periodo de aumento de niveles de CO₂ por las actividades de limpieza.

2.1.4 Gimnasio

Arquitectura.

- **Tamaño y estructura:** El área es de 250m². Está dividido en diferentes áreas de ejercicio: pesas, máquinas de cardio y entrenamiento funcional. La altura del techo es promedio, lo que puede dificultar la ventilación.
- **Equipos y herramientas:** Limpia vidrios, desinfectantes, aromatizantes, detergentes.

- Ventilación: Ventanas y ventiladores.

Aforo. Los espacios están diseñados para albergar un máximo de 25 personas.

Fuentes de gases y predominancia.

- Dióxido de carbono: Respiración de las personas. Concentración alta en horas pico.
- Monóxido de carbono: No hay fuentes relevantes de CO.
- VOC y NO₂: Detergentes, desodorantes, ambientadores. Puede alcanzar picos durante los periodos de limpieza.
- Partículas de materia: polvo levantado del suelo, partículas de desgaste de equipos y aerosoles. En general, su presencia es baja.

Horarios y fluctuación de gases.

- Mañana (6:00-10:00): Apertura, niveles de CO₂ de bajos a moderados
- Mediodía (10:00-15:00): Horas de actividad media a baja. VOC y NO₂ alcanzan una intensidad media debido al primer periodo de limpieza general, CO₂ de moderado a bajo
- Tarde (15:00-20:00): Aumento progresivo de usuarios y concentración de CO₂ fluctuando de baja a insalubre.
- Resto del día: Durante el segundo periodo de limpieza general, VOC y NO₂ también se encuentran en cantidades altas.

2.2 Contaminantes de aire interior

En el libro *Modeling indoor air pollution*, Pepper y Carrington (2009) definen como contaminantes de aire interior (Indoor Air Pollutants, IAP) a aquellas sustancias que afectan directamente a la calidad de aire (IAQ) y el confort humano dentro de los edificios. Los autores señalan que el estudio del IAP involucra la observación y análisis de las fuentes de emi-

sión, la acumulación en los recintos y la evaluación de los contaminantes, los que se atribuyen a una mala ventilación e intercambio de aire.

La Organización Mundial de la Salud dispone de las publicaciones *WHO Guidelines for indoor air quality: Selected pollutants* (World Health Organization, 2010) y *WHO Guidelines for indoor air quality: Dampness and mould* (World Health Organization, 2009), en las que se describe la implicación de diferentes gases que afectan la calidad de aire en interiores, así como también la humedad y el moho. De acuerdo con estos documentos, los principales contaminantes del aire en interiores son:

2.2.1 Benceno

Aunque está presente en exteriores e interiores, el benceno puede tener mayores concentraciones en espacios cerrados debido a las infiltraciones externas y fuentes interiores. Esta sustancia representa un potencial riesgo de cáncer en las personas, en especial la leucemia, y es por esto que ningún nivel de exposición es seguro. Las actividades comunes que liberan benceno en el aire son fumar tabaco, utilizar solventes para pasatiempos o limpieza y materiales de construcción que liberen benceno tales como pisos de vinilo alfombras de nailon, adhesivos para piso, etc. Se sabe que el 40 % a 60 % de las concentraciones de benceno en residencias se debe a la presencia de garajes adjuntos y que las concentraciones pueden elevarse a $8\mu\text{g}/\text{m}^3$ cuando la cochera está conectada directamente con la vivienda. Otros combustibles como carbono, madera, gas, queroseno o GLP generan concentraciones de benceno aún más altas, por ejemplo, el uso del queroseno en la cocina potencia su presencia entre 44 y $167\mu\text{g}/\text{m}^3$.

2.2.2 Monóxido de carbono

La exposición al CO merma la capacidad máxima de ejercicio en personas sanas, incluso llega a causar angina, también disminuye el segmento ST de la señal cardíaca en pacientes con enfermedades cardiovasculares, se ha relacionado directamente con el desarrollo de sordera, dolores de cabeza y deficiencia cognitiva y puede generar hipoxia cerebral. En general, la relación entre CO y la carboxihemoglobina en sangre no debe exceder el 2 %, de

acuerdo a la ecuación diferencial de Coburn Forster-Kane. La exposición crónica y aguda al CO es diferente por lo que se establece como referencia que los niveles de concentración deben ser menores a 10mg/m^3 durante 8H, 100mg/m^3 en 15min, 35mg/m^3 en una hora y 7mg/m^3 durante 24 horas. Las fuentes comunes de monóxido de carbono son cocinas de combustión, calefacciones, humo de tabaco o tubos de escape de autos en garajes adjuntos a las viviendas al igual que zonas de aparcamiento subterráneo. Otra fuente relevante de CO son las emisiones de vehículos en el exterior que se filtran fácilmente al interior de las edificaciones.

2.2.3 Formaldehído

El formaldehído se encuentra ubicuamente en el medio ambiente, porque está formado principalmente por numerosas fuentes naturales y actividades antropogénicas. Se libera al medio ambiente mediante la combustión de biomasa. Las fuentes antropogénicas incluyen las directas, como las emisiones industriales in situ y la quema de combustible procedente del tráfico. Otros procesos de combustión (centrales eléctricas, incineración, etc.) también representan fuentes de emisiones de formaldehído a la atmósfera. Sin embargo, el formaldehído también se produce industrialmente en todo el mundo para su uso en la fabricación de desinfectantes, pegamentos y otras resinas. Todos estos productos y usos artificiales son las principales fuentes indirectas de formaldehído. Por último, cabe señalar que la formación secundaria de formaldehído se produce en el aire mediante la oxidación de compuestos orgánicos volátiles (VOC). Para el caso de viviendas, las concentraciones de formaldehído se relacionan a la antigüedad del recinto, temperatura y humedad. Debido a su alta solubilidad en agua, esta sustancia se absorbe fácilmente por el organismo; los efectos no cancerígenos pueden ser irritación sensorial de los ojos y las vías respiratorias superiores, efectos pulmonares (asma o alergia) y eczema. Una exposición prolongada al formaldehído puede causar cáncer de nasofaringe y leucemia mieloide. De acuerdo a la Administración de Seguridad y Salud Ocupacionales (OSHA), la concentración máxima de formaldehído a la que una persona puede estar expuesta es de 0.75PPM durante 8 horas y para un periodo de 30min es de 0.1mg/m^3 .

2.2.4 Naftaleno

Su presencia está asociada a lesiones en el tracto respiratorio, incluidos los tumores en el tracto respiratorio superior demostrados en estudios con animales y la anemia hemolítica en humanos. Las fuentes más comunes son solventes multiuso, lubricantes, herbicidas, encendedores de carbón, calefacciones a base de queroseno y las bolas de naftaleno utilizadas para proteger textiles. Dado que las personas pasan la mayor parte de su tiempo en ambientes interiores, la inhalación del aire en espacios cerrados juega un papel crucial en la exposición humana total al naftaleno. La WHO indica que la cantidad de naftaleno en interiores no debe superar los 0.01mg/m^3 anualmente y en el entorno laboral este valor no debe superar 53mg/m^3 en un periodo de 8 horas.

2.2.5 Dióxido de nitrógeno(NO_2)

Este compuesto químico tiene el potencial de causar lesiones e inflamación en tejidos vivos, esto lo hace altamente peligroso para el sistema respiratorio. Las cocinas de gas son fuentes de NO_2 en hogares, así como también calefacción de combustión. También se ha observado que la relación entre los niveles de NO_2 en el exterior e interior de un edificio, en ausencia de fuentes internas, es casi igual de lo que se deduce que la distancia entre viviendas y carreteras influencia en gran medida su presencia. Oxidantes como el oxígeno, ozono y VOC favorecen a la rápida proliferación del dióxido de nitrógeno. Las pautas para la exposición al NO_2 es de 200ug/m^3 por una hora como máximo y de 40mg/m^3 anualmente.

2.2.6 Hidrocarburos Aromáticos Policíclicos (HAP)

Son potentes carcinógenos que se adhieren fácilmente a partículas suspendidas en el aire. Los HAP se encuentran en interiores donde predomine el uso de combustibles para calefacción o cocina, otras fuentes observadas son alimentos con contenido elevado de grasas que se fríen y el tabaquismo. Las vías de exposición a los HAP son inhalación, ingesta de alimentos o agua y contacto con la piel. Los HAP se relacionan fuertemente con la aparición de cáncer de pulmón, vejiga, mama y la exposición prenatal aumenta la probabilidad de un bajo

peso al nacer. No se puede determinar un umbral y se considera peligroso cualquier nivel de exposición. Las concentraciones a la exposición de por vida a los HAP, que produce riesgos excesivos de cáncer de por vida de 1/10000 1/100000 y 1/1000000 son aproximadamente 1.2, 0.12 y 0.012 ng/m³, respectivamente.

2.2.7 Radón

El gas radón es una fuente importante de radiación ionizante de origen natural y un importante contribuyente a la dosis de radiación ionizante que recibe la población en general. Tiene varios isótopos, es un gas noble radiactivo incoloro e inodoro que se produce naturalmente. El más estable de los isotopos es el radón-222 (222Rn). El radón es un gas con propiedades carcinógenas cuya presencia en interiores se asocia a un mayor riesgo de cáncer de pulmón, especialmente en personas fumadoras, donde sus efectos se potencian notablemente. Se adhiere fácilmente a partículas de materia de 0.1 micras suspendidas en el aire las que al ser inhaladas facilitan la penetración del radón. WHO estima que entre un 3 y 14 % de los casos de cáncer de pulmón se deben a este gas y también se ha encontrado evidencia que lo relaciona con la leucemia y cáncer de las vías respiratorias extratorácicas. El radón se encuentra en el suelo, por lo que puede estar presente en edificios recién construidos, minas, campos de cultivos, entre otros. Las concentraciones de radón con una probabilidad de riesgo a lo largo de la vida de 1/100 y 1/1000 son 6.7 Bq/m³ y 1670 Bq/m³ para los fumadores; para los no fumadores sería 1670 Bq/m³ y 167 Bq/m³, respectivamente.

2.2.8 Tricloroetileno(C₂HCl₃)

Es un disolvente bastante común en industrias, es incoloro y volátil. Es utilizado para desengrasado con vapor, limpieza en frío de piezas metálicas fabricadas y producción de tintas de impresión. Las principales vías de exposición se deben al contacto con tintes para madera, lubricantes, adhesivos, removedores de pintura y ciertos limpiadores. En algunos lugares se ha observado presencia de tricloroetileno en agua potable, sin bien fueron niveles bajos, alrededor de 1ug/L, también se detectó concentraciones más elevadas (hasta 49ug/L). Está presente en algunos alimentos como los derivados de la leche, por ejemplo, se pudo medir 73ug/kg en

mantequilla y margarina y hasta 3ug/kg en el queso. Ya que es una sustancia lipófila, cuando el cuerpo la absorbe esta puede almacenarse en la grasa subcutánea, hígado, riñones, sistema cardiovascular y nervioso. No hay un umbral establecido para un nivel seguro, pero se tiene una estimación de riesgo unitario de exposición de $4.3 \times 10^{-7} \text{ug/m}^3$. Las concentraciones de tricloroetileno en el aire asociadas con la posibilidad de contraer cáncer a lo largo de la vida para 1/10000, 1/100000 y 1/1000000 son respectivamente 230, 23 y 2.3 pg/m^3 .

2.2.9 Tetracloroetileno(C_2Cl_4)

Si bien no hay indicios de que este compuesto sea genotóxico, sí puede ser una causa de deterioro neuroconductual y cambios renales tempranos. La exposición media anual debe ser menor a 25mg/m^3 .

2.2.10 Humedad y moho

La proliferación de diferentes agentes biológicos está fuertemente relacionada a la cantidad de humedad y a una mala ventilación. Una elevada humedad provoca el crecimiento de moho, microbios y bacterias que eventualmente emiten esporas y VOC que pueden ser inhaladas por las personas. La humedad también degrada materiales, lo que también afecta a la calidad de aire en interiores. Por tanto, la humedad es un indicador fuerte y consistente de riesgo de asma y otras enfermedades respiratorias. El moho puede causar enfermedades como neumonitis, alveolitis alérgica, rinosinusitis crónica y sinusitis alérgica por hongos. La presencia de humedad varía ampliamente entre zonas geográficas, por ejemplo, afecta entre 10 % y 50 % de ambientes interiores de Europa, América, Australia, India y Japón; además los niveles de humedad en la costa o valles fluviales son mucho más elevados que los promedios nacionales. Las directrices para la humedad son mantener los niveles entre un 40 % y 60 % además de usar deshumidificadores en temporadas de mayor concentración además de una ventilación adecuada.

2.2.11 Temperatura

Tiene una influencia directa en el confort humano, humedad y reproducción de agentes microbianos. La temperatura ideal para la presencia de personas en interiores ronda entre 21°C y 25°C, dependiendo de la estación del año.

Además de los contaminantes mencionados, la USEPA identifica otros compuestos químicos que tienen el potencial de ser contaminantes del aire:

2.2.12 Partículas de materia (PM)

Son una mezcla de partículas sólidas y líquidas suspendidas en el aire, pueden variar en tamaño, composición y forma. USEPA considera más relevantes las que tienen tamaños inferiores a 10µm de diámetro, en especial las menores a 2.5µm, debido a que son inhalables y cuando esto sucede pueden llegar a afectar a los pulmones, corazón, ojos, vías respiratorias y hasta la sangre. Las principales fuentes de PM son las filtraciones del exterior, actividades que impliquen el uso de combustibles, pinturas, aerosoles y también pueden ser de origen biológico como el moho. Hay diferentes formas de controlar la aglomeración de partículas de materia: instalar extractores que expulsen aire hacia el exterior, evitar el uso de cocinas o calentadores portátiles en interiores sin una adecuada ventilación, en caso de usar leña se debe considerar su aplicación en estufas de tamaño adecuado. USEPA ofrece el programa BurnWise para conocer prácticas seguras de la quema de leña.

2.2.13 Compuestos orgánicos volátiles (VOC)

Se emiten desde sólidos o líquidos, por lo que está compuesto por diversas sustancias químicas que pueden tener efectos en la salud a corto y largo plazo. La concentración de VOC llega a ser diez veces más alta en interiores que exteriores, pero algunas actividades pueden elevar esta cifra a niveles mucho más elevados, por ejemplo, quitar pintura genera un incremento de mil veces en los valores de VOC medidos. Se encuentran en muchos productos de limpieza, cosméticos, pinturas, pesticidas, barnices, entre otros. Los VOC causan irritación de ojos, nariz y garganta; daño del hígado, riñón y sistema nervioso central. Los VOC más co-

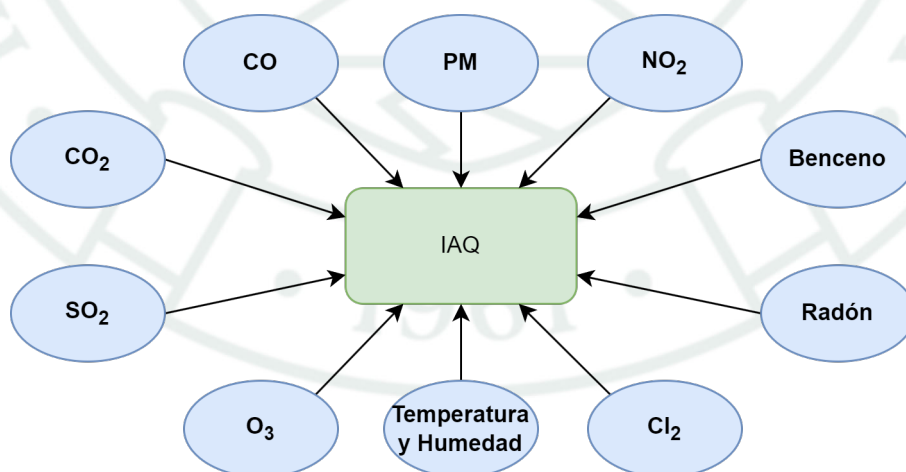
munes son el tetracloroetileno, formaldehído, alcohol isopropílico, acetona, butano, propano y tolueno. Varias organizaciones han establecido los niveles de exposición a los VOC acorde a los del formaldehído.

2.2.14 Dióxido de carbono(CO_2)

Es un gas inodoro e incoloro. En sí mismo, no es tóxico, ya que los humanos convierten el oxígeno en CO_2 durante el proceso de respiración, pero altas concentraciones en espacios interiores pueden generar desde pérdida de concentración, fatiga, aumento de la frecuencia cardíaca hasta visión borrosa, desmayos y asfixia. La aglomeración de CO_2 dentro de una habitación significa una disminución del oxígeno y esto se observa en lugares donde la presencia de personas es elevada. Se considera como normal un valor inferior a 700ppm y se comienza a percibir síntomas leves a partir de 1000ppm.

Además de los gases mencionados, se pueden considerar otros contaminantes del aire en interiores, tales como el plomo, dióxido de azufre, cloro molecular, y ozono y asbestos. En la Figura 2.1 se presenta un diagrama que representa la combinación de todos los contaminantes del aire en espacios interiores que han sido considerados por la OMS, la USEPA y otros estudios, como los mencionados en el estado del arte(Sección 2.9).

Figura 2.1
Contaminantes que afectan la calidad del aire interior



Nota. Elaboración propia basada en World Health Organization (2010).

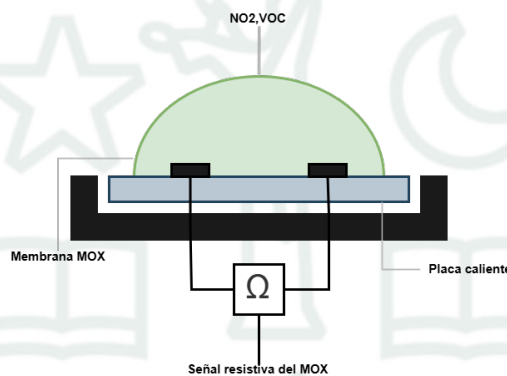
2.3 Tipos de sensores de calidad de aire

2.3.1 Sensores electroquímicos

Son dispositivos que responden a variaciones en el ambiente químico, los cuales permiten obtener información cualitativa y cuantitativa de la composición química del ambiente y están compuestos de una capa químicamente selectiva y un transductor físico (González, 2023).

Un ejemplo de sensor electroquímico son los de óxido metálico (metal oxide, MOX), que tienen una superficie caliente que cambia su resistencia en función del oxígeno en la superficie (Sensirion, 2022). En la figura 2.2 se observan los componentes del sensor.

Figura 2.2
Sensor óxido metálico



Nota. Elaboración propia basada en (Sensirion, 2022).

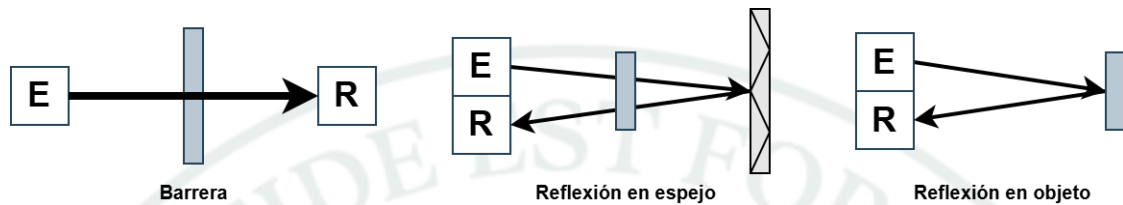
2.3.2 Sensores Fotoeléctricos

Utilizan fotodiodos o fototransistores como componentes principales. El diseño básico de estos sensores consiste en colocar dos fotodiodos actuando como emisor y receptor. La figura 2.3 muestra las configuraciones de los sensores fotoeléctricos de acuerdo a la posición de los fotodiodos, la primera es de barrera, en la que el emisor y receptor están uno frente al otro con el objeto(barrera) a medir en medio, la segunda es de reflexión en espejo en la que también hay una barrera pero la luz del emisor debe traspasar la barrera, reflejarse en un espejo y volver a traspasar la barrera para llegar al receptor; y la tercera es de reflexión en objeto, donde

la luz del emisor se refleja directamente en el objeto que se quiere medir y regresa al receptor (Pérez García, 2014).

Figura 2.3

Tipos de sensor fotoeléctrico



Nota. Elaboración propia basada en (Pérez García, 2014).

2.4 Actuadores y accionamientos para calidad de aire

En el libro *Air pollution control equipment selection guide*, Schiffner (2013) presenta los siguientes tipos de actuadores utilizados en el control del IAQ:

- **Dispositivos de absorción:** Consisten en medios absorbentes, estáticos o móviles, en un recipiente contenedor a través del cual pasan el gas y sus contaminantes. Los contaminantes se absorben sobre y dentro de los poros del medio de absorción. Algunos ejemplos son los absorbedores de carbón y extractores de polvo, entre otros.
- **Biorreactores:** Usan el principio de biorremediación por el cual los gases contaminantes solubles en agua (VOC) que están en el aire son consumidos por bacterias y hongos. Los biorreactores también se pueden usar para algunos sulfuros o NOx. Los dispositivos que pertenecen a esta categoría son los biofiltros, digestores de aire y biodepuradores.
- **Colectores ciclónicos secos:** Eliminan las partículas secas de contaminantes (polvo) mediante fuerza centrífuga, haciendo que estas se peguen a las paredes internas del dispositivo mientras el aire limpio sale por una tubería. Estas máquinas se utilizan con partículas de hasta 5µm porque las de menor tamaño se resisten a la separación inercial.

Otros actuadores:

- Válvulas motorizadas, para movilizar o controlar el flujo de aire en los dispositivos antes mencionados.
- Humidificadores y deshumidificadores, que evitan la proliferación de bacterias y moho.
- Extractores de aire o ventiladores, de uso más común en ambientes no industriales, pueden regular el flujo de aire sin necesidad de sistemas más complejos.

2.5 Sistemas Inteligentes de IoT

El desarrollo de dispositivos de monitoreo de IAQ con tecnología IoT requiere de la selección de microcontroladores y microprocesadores adecuados a las necesidades de estos sistemas. Los más utilizados son Arduino Uno, Raspberry Pi, ESP8266 y ESP32 (Saini et al., 2022)

- Arduino UNO: Esta placa usa un uC ATmega328 que funciona a 16MHz con memoria de programa de 32KB, 2KB de RAM y 1KB de EEPROM. Tiene 14 pines digitales I/O y 6 entradas analógicas.
- Raspberry Pi: Es una microcomputadora con diferentes versiones, todas usan una CPU ARM Cortex de 1 hasta 4 núcleos, RAM desde 512MB hasta 8GB y tienen un módulo de 40 pines o JTAGS de propósito general con entradas y salidas digitales, puertos de comunicación UART, I2C, SPI. Los Raspberry Pi, a diferencia del Arduino UNO, soportan conexión a internet mediante Wi-Fi y Ethernet, lo que les permite servir también como Gateway.
- ESP8266 y ESP32: Ambos pertenecen a la familia de microcontroladores de la marca Espressif Systems y son la alternativa a Arduino UNO más utilizada debido a su potencia, bajo costo y sobre todo al soporte integrado de comunicación WIFI. El módulo ESP8266 cuenta con hasta 16 pines, entradas y salidas digitales, según el modelo; conversor ADC de 10 bits. Otros protocolos de comunicación soportados son I²C, SPI, I²S, UART y PWM. Tiene un procesador RISC de 32 bits basado en el Tensilica Xtensa

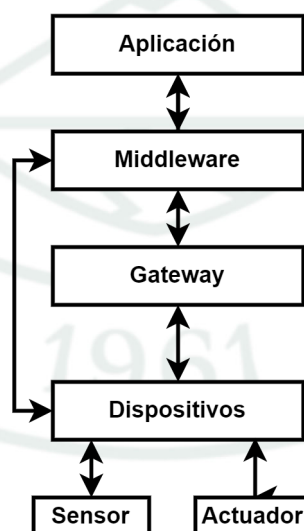
LX106 funcionando a 80 MHz, 50KB de RAM, ROM de 16KB y hasta 4MB de memoria flash. La placa de desarrollo ESP32 tiene un procesador Tensilica Xtensa LX6 de 2 núcleos con frecuencia de reloj de hasta 240MHz, RAM de 512KB, ROM de 448KB y flash integrado de 4MB o más. Posee hasta 36 pines con módulos ADC, DAC, PWM, I²C, SPI, I²S, UART, sensor táctil. También integra comunicación WIFI, además de Bluetooth.

2.6 Internet de las Cosas

En su publicación *Internet of Things A to Z: Technologies and applications*, Hassan (2018) define IoT como la tecnología por la cual los objetos se interconectan, teniendo la capacidad de sensor, actuar, comunicarse entre ellos y con el entorno, mientras proveen la habilidad de compartir información y operar de manera autónoma con eventos del mundo físico, generando procesos y creando servicios con o sin necesidad de la intervención humana.

De esta definición se puede deducir una arquitectura general de un sistema de internet de las cosas, tal como la presentada por Guth et al. (2016) que de manera simplificada la divide de la siguiente manera:

Figura 2.4
Arquitectura simplificada de IoT



Nota. Elaboración propia basada en (Guth et al., 2016).

La descripción de cada bloque es:

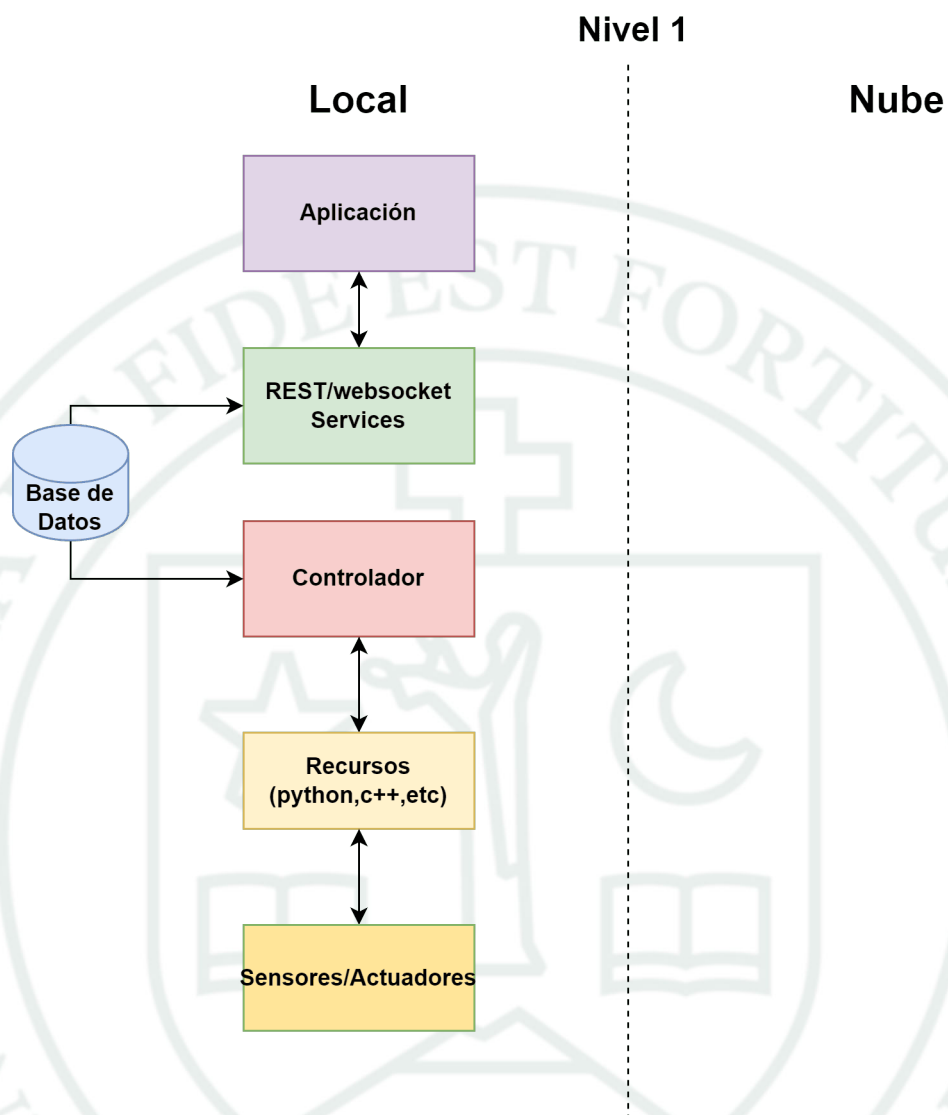
- **Sensores y actuadores:** Son componentes de hardware que se encargan de medir parámetros del ambiente y de actuar, controlar o manipular el entorno físico, respectivamente.
- **Dispositivos:** Es otro elemento de hardware que sirve tanto para leer la información de los sensores como para controlar los actuadores.
- **Gateway:** En caso los dispositivos no sean capaces de comunicarse con otros sistemas, el Gateway sirve como intermediario entre ellos, traduciendo comandos o datos a protocolos y formatos de comunicación compatibles con el sistema que los recibe. Por ejemplo, el dispositivo envía datos al Gateway a través del protocolo MQTT y este lo convierte a XML para ser recibido en la World Wide Web.
- **Middleware:** Es el que recibe y procesa la información enviada por los dispositivos para devolverle comandos de control con el fin de activar los actuadores. Otra de las funciones del middleware es entregar datos a la capa de aplicación.
- **Aplicación:** Es una herramienta de software que sirve para que el usuario final pueda observar los datos enviados desde los sensores o incluso decidir sobre el control de los actuadores. Un ejemplo de aplicación son los dashboards o paneles donde se puede observar la variación de temperatura de un recinto.

2.6.1 Niveles de Internet de las Cosas

De la arquitectura general de IoT descrita en la sección anterior se pueden desglosar diferentes modelos de construcción de un sistema IoT, los que se representan en niveles crecientes de componentes y complejidad. En el libro *Internet of Things: A hands-on approach* de Bahga y Madisetti (2014), se desglosa cada uno de estos niveles como sigue:

Nivel 1. En este modelo, todo el sistema se implementa localmente, es decir, no existe ningún componente externo. Es útil para cuando no se requiere de una nube o un servidor remoto, y los dispositivos interactúan únicamente con el usuario. La figura 2.5 muestra cómo se compone.

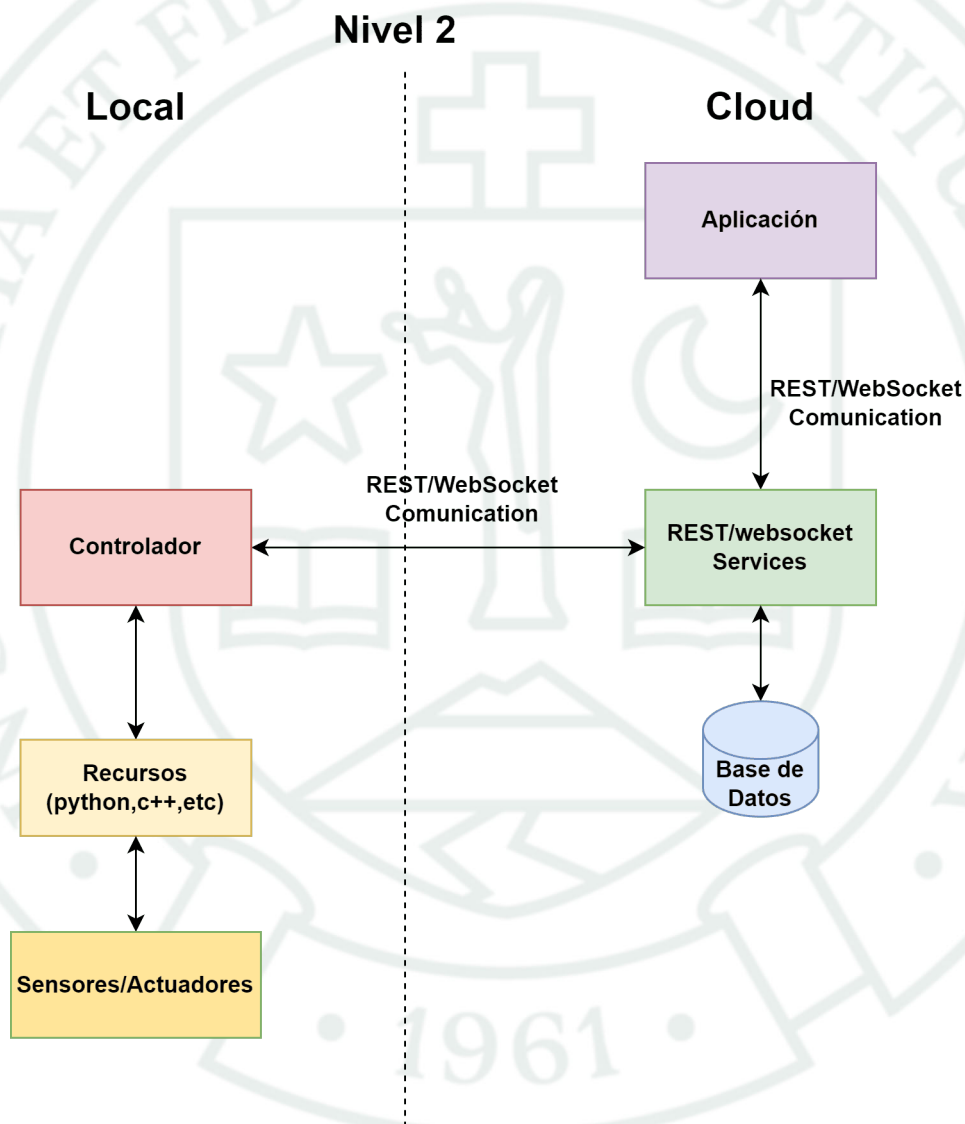
Figura 2.5
Nivel 1



Nota. Elaboración propia basada en Bahga & Madiseti (2014).

Nivel 2. Cuando si se necesita poder acceder a los datos fuera de la red local, se debe implementar una nube en la que se puede almacenar y visualizar las mediciones. Este nivel se desarrolla cuando la cantidad de información es relativamente grande y lo más adecuado es que sea almacenada en un servidor remoto, aunque el procesamiento no es intenso, por lo que puede manejarse localmente.

Figura 2.6
Nivel 2

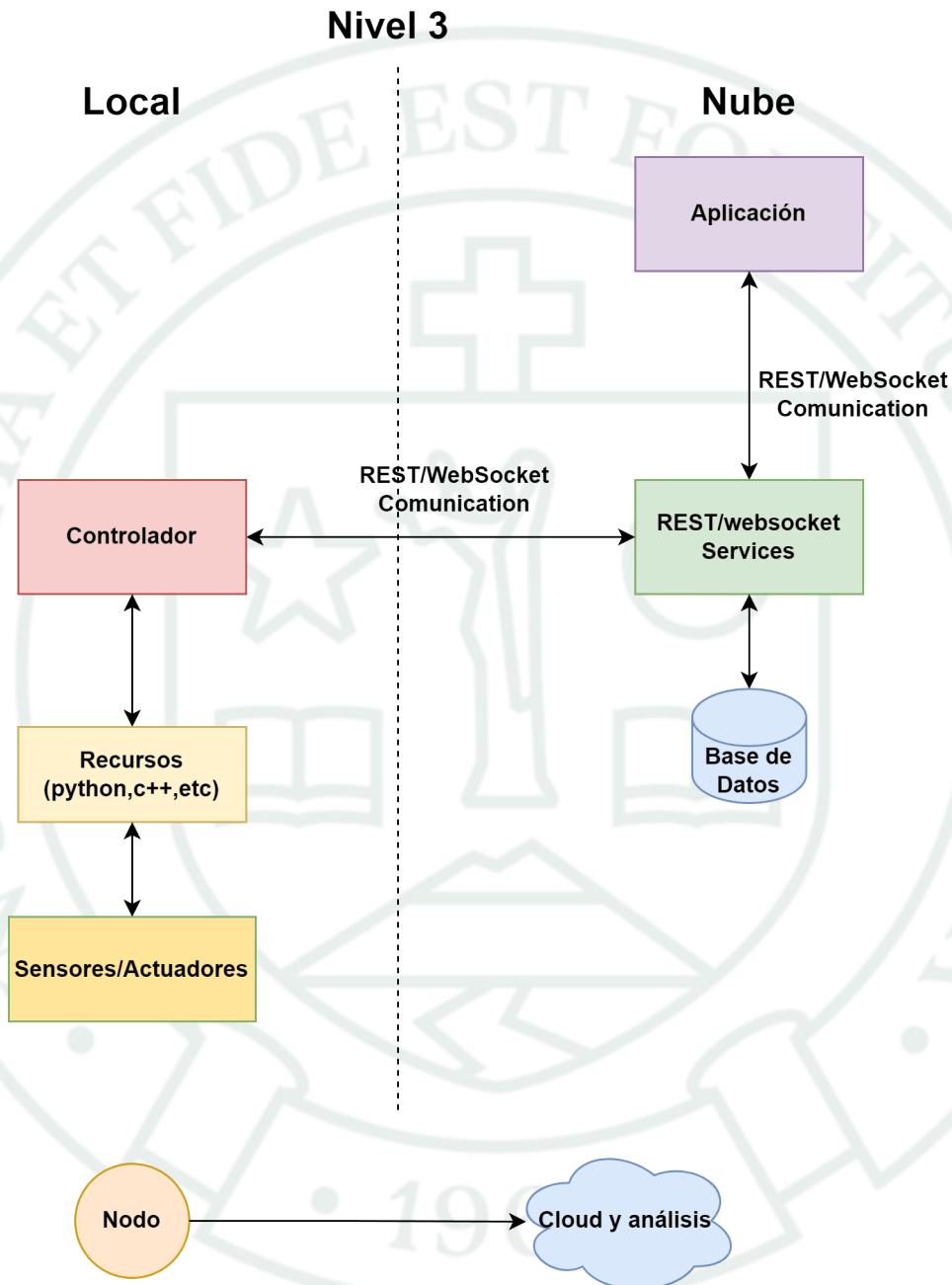


Nota. Elaboración propia basada en Bahga & Madiseti (2014).

Nivel 3. Es muy parecido al nivel 2 pero el procesamiento se realiza ahora en la nube y a nivel local solo se recolecta y envía los datos.

Figura 2.7

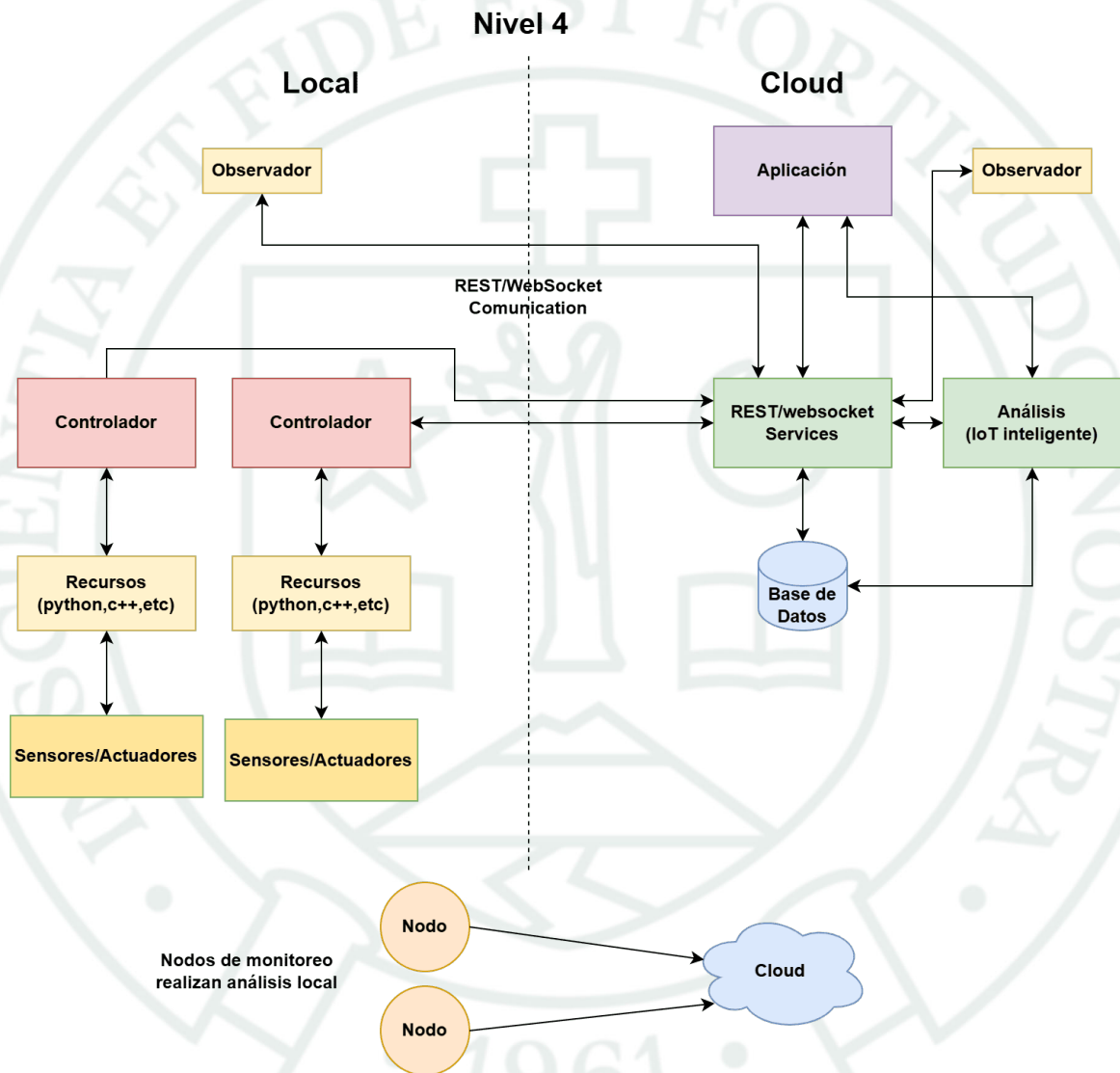
Nivel 3



Nota. Elaboración propia basada en Bahga & Madiseti (2014).

Nivel 4. Es una expansión del nivel tres en el que se tiene múltiples dispositivos o nodos que tienen sensores, actuadores y controladores independientes. Además, en el nivel 4 se integran dos nodos observadores: uno a nivel local y otro en la nube; pueden procesar información y utilizarla para diversas aplicaciones, pero no ejercen ninguna función de control.

Figura 2.8
Nivel 4

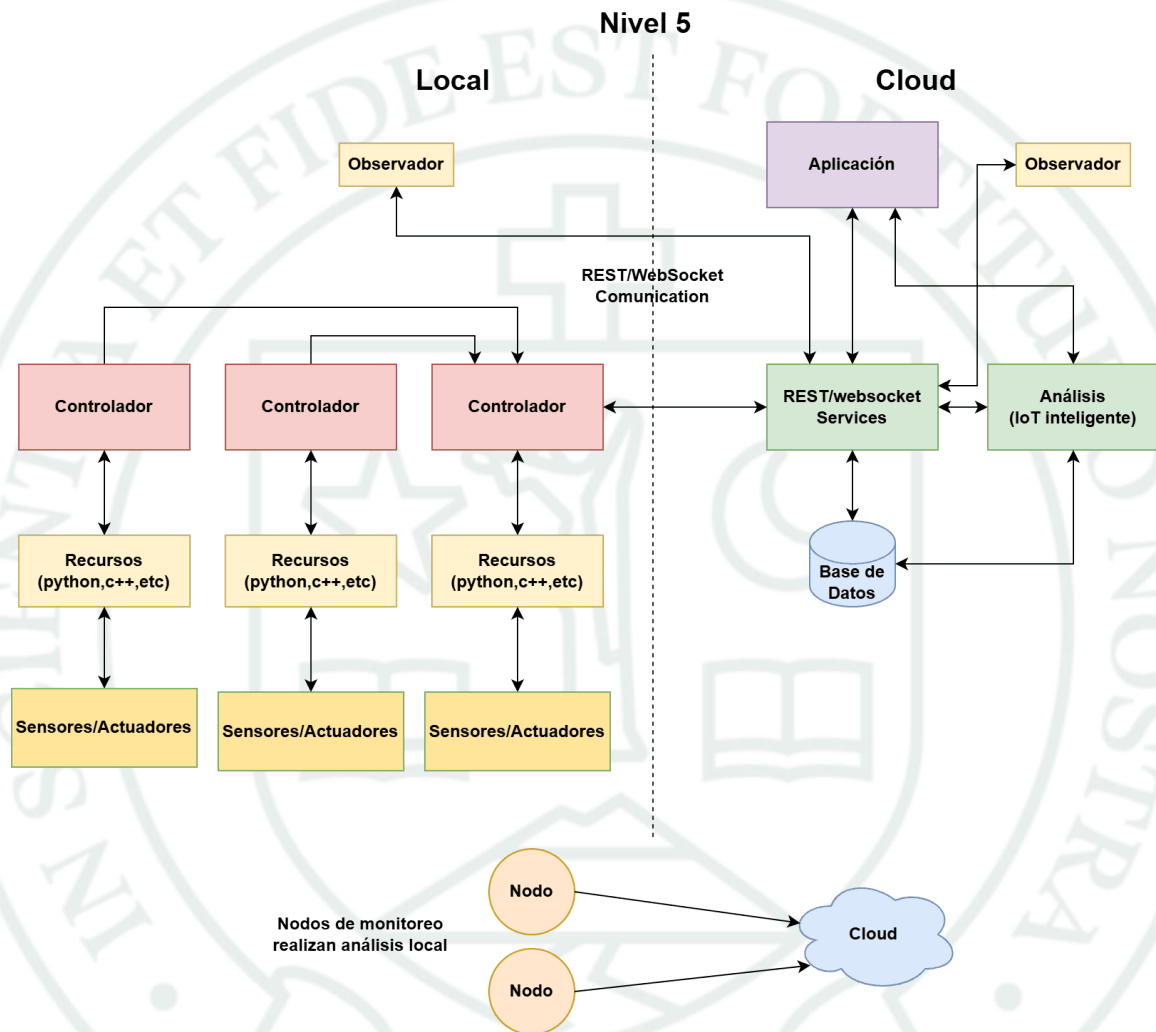


Nota. Elaboración propia basada en Bahga & Madiseti (2014).

Nivel 5. Es una mejora del nivel 4, ahora los nodos se comunican a un nodo coordinador y este se encargará del envío y recepción de los datos a la nube, además, la etapa de procesamiento ahora integra capacidades de predicción.

Figura 2.9

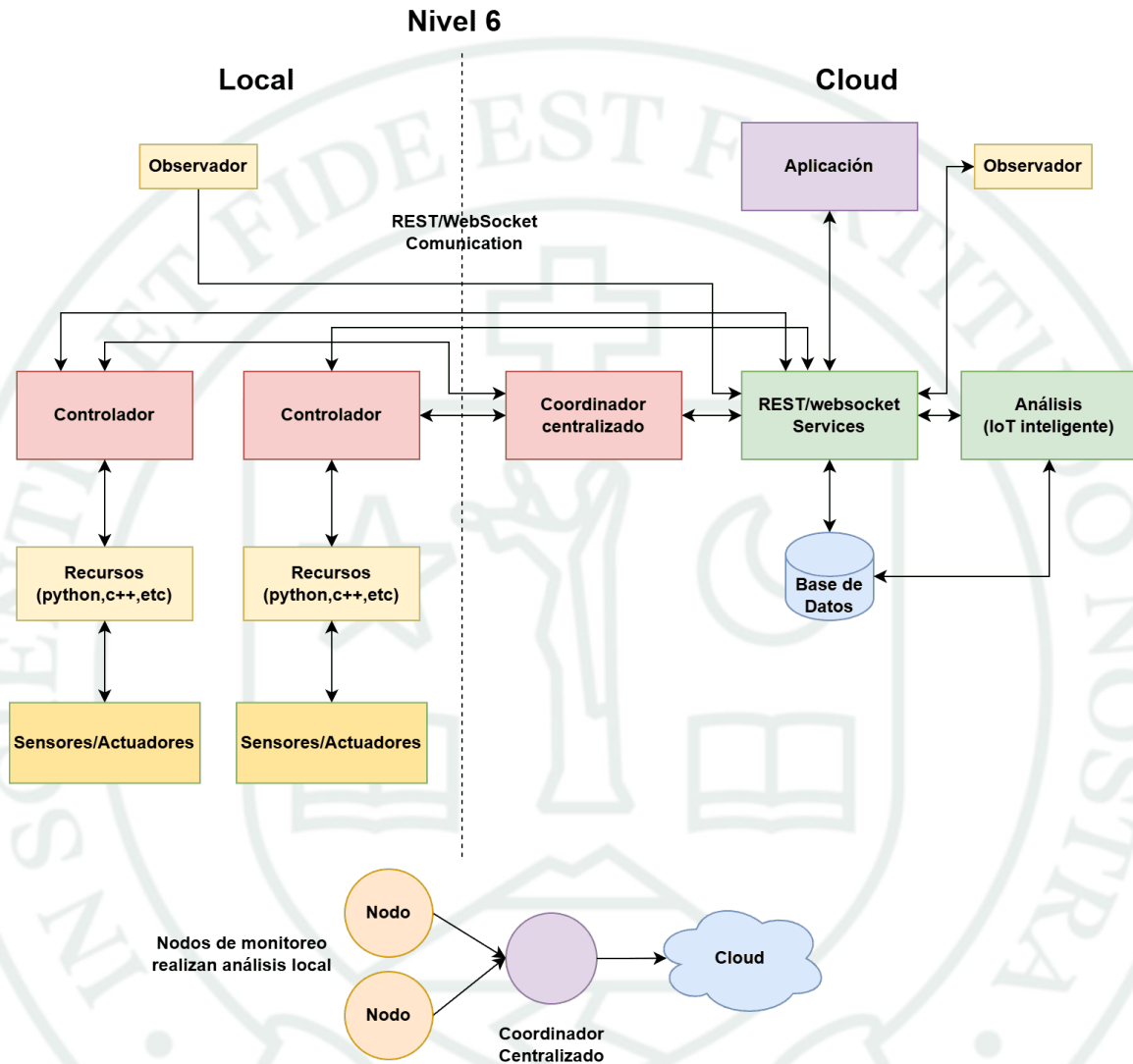
Nivel 5



Nota. Elaboración propia basada en Bahga & Madiseti (2014).

Nivel 6. Su estructura se diferencia de la de nivel 5 en el nodo coordinador, que ahora se encuentra virtualizado en la nube.

Figura 2.10
Nivel 6



Nota. Elaboración propia basada en Bahga & Madiseti (2014).

2.7 Inteligencia Artificial aplicada en IAQ

Esta sección detallará diferentes algoritmos de inteligencia artificial a fin de proporcionar una base adecuada para su aplicación en IAQ. La inteligencia artificial está compuesta por diferentes ramas. En el capítulo “Predicting indoor air quality: Integrating IoT with Artificial

Intelligence”, publicado en el libro *Internet of things for indoor air quality monitoring*, Saini et al. (2021) la subdividen de la siguiente manera:

- **Machine Learning:** Se enfoca en el desarrollo de algoritmos que permitan a las computadoras tomar decisiones directamente a través de datos y experiencias pasadas.
- **Deep Learning:** Básicamente, es una subárea de ML y es la forma más avanzada de IA que las máquinas utilizan para procesar datos de entrada a través de múltiples capas para conseguir una predicción o clasificación más precisa.
- **Redes neuronales:** Su funcionamiento es similar al de una neurona humana y son el algoritmo más popular de IA. Pueden tener múltiples nodos en diferentes capas y conexiones. Sirven para el reconocimiento de patrones (clasificación), predicción y toma de decisiones.
- **Procesamiento de lenguaje natural:** Esta rama de la inteligencia artificial permite a las máquinas leer, comprender e interpretar el lenguaje humano.
- **Visión por ordenador:** También llamada Visión Artificial, es la técnica mediante la cual, las computadoras pueden reconocer y clasificar imágenes.
- **Lógica Difusa:** Inspirada en el razonamiento humano, esta técnica posibilita la toma de decisiones intermedias. En lugar de limitarse a un simple “sí” o “no”, la lógica difusa permite encontrar soluciones intermedias o graduales.

Antes de poder utilizar cualquier algoritmo de inteligencia artificial, es indispensable hacer un preprocesamiento de los datos, que consiste en llenar los espacios vacíos del conjunto de datos, normalización y codificación de variables categóricas.

El conjunto de datos o dataset en inglés es una matriz que contiene en la primera fila los nombres de las variables, mejor llamadas etiquetas, y en las columnas a los valores de estas etiquetas.

Cumplida la etapa de preprocesamiento de datos, la selección de parámetros permite establecer relaciones entre ellos y así determinar cuáles son más idóneos para introducir en el

algoritmo de IA; en el caso de IAQ se debe tener en cuenta este criterio de selección porque son varias variables las que pueden afectar la concentración de contaminantes dentro de un edificio.

Algunos de los métodos más comunes de selección de variables dominantes son:

- Correlación Cruzada
- Análisis de componentes principales
- Teoría de conjuntos aproximados
- Bosques aleatorios

2.8 Algoritmos de predicción

2.8.1 Regresión Lineal

El objetivo de la regresión lineal es establecer una relación lineal entre una variable de entrada X y la de salida Y (Burkov, 2019, Capítulo 3). Entonces para una colección de datos $\{(x_i, y_i)\}_{i=1}^N$, donde N es la cantidad de elementos del conjunto; se requiere obtener un modelo $f_{w,b}(X)$ como una combinación lineal de características de x , matemáticamente sería:

$$f_{w,b}(x) = wx + b \quad (2.1)$$

donde W es un vector de parámetros D -dimensional y b es un número real. La notación f , significa que el modelo f está parametrizado por los valores W y b .

Este modelo sirve como predictor de una Y desconocida para una X dada tal que $y \leftarrow f_{w,b}(x)$. Se debe encontrar los valores de W y b más óptimos para este fin, esto se consigue a través de minimizar la expresión:

$$\frac{1}{N} \sum_{i=1 \dots N} (f_{w,b}(x_i) - y_i)^2 \quad (2.2)$$

Que recibe el nombre de Error medio cuadrático.

2.8.2 Árboles de Decisión

Los árboles de decisión dividen los datos según sus características y sirven para clasificación y predicción (Burkov, 2019, Capítulo 3). Su estructura se inspira en un árbol en la que cada nodo representa una decisión basada en una característica, las ramas representan las reglas de decisión para subdividir un nodo principal en otros más pequeños y las hojas serían los resultados posibles dentro del conjunto de datos. Una extensión de los árboles de decisión son los bosques aleatorios donde se unen múltiples árboles para una predicción aún más precisa.

Uno de los algoritmos de aprendizaje para árboles de decisión es el ID3. El criterio de optimización, en este caso, es la probabilidad logarítmica promedio descrita como:

$$\frac{1}{N} \sum_{i=1}^N y_i \ln f_{ID3}(\mathbf{x}_i) + (1 - y_i) \ln (1 - f_{ID3}(\mathbf{x}_i)) \quad (2.3)$$

Donde f_{ID3} es el árbol de decisión. El algoritmo ID3 funciona así: sea S un conjunto de muestras y al principio existe solo un nodo que tiene todos los datos, es decir,

$$S \stackrel{\text{def}}{=} \{(\mathbf{x}_i, y_i)\}_{i=1}^N. \quad (2.4)$$

El primer nodo es constante y se describe según la ecuación

$$f_{ID3}^S = \frac{1}{|S|} \sum_{(\mathbf{x}, y) \in S} y \quad (2.5)$$

y la predicción dada por f_{ID3}^S sería la misma para cualquier X . Ahora se debe buscar a través de todas las características $i = 0, 1, 2, \dots, D$ y todos los umbrales t , luego se divide el conjunto S en dos: $S_- \stackrel{\text{def}}{=} \{(\mathbf{x}, y) \mid (\mathbf{x}, y) \in S, x^{(j)} < t\}$ y $S_+ = \{(\mathbf{x}, y) \mid (\mathbf{x}, y) \in S, x^{(j)} \geq t\}$. Los dos nuevos subconjuntos irían a dos nuevos nodos hoja, y evaluamos, para todos los pares posibles (j, t) , qué tan buena es la división con las piezas S_- y S_+ a fin de obtener el mejor par (j, t) . Luego se vuelve a dividir S en otros dos subconjuntos S_- y S_+ continuando recursivamente hasta que no se consiga una mejor partición. Para saber que la partición es correcta se

recurre a la entropía que es una medida de incertidumbre sobre una variable aleatoria. La entropía alcanza su máximo valor cuando las incertidumbres de todas las variables aleatorias son equiprobables y valor mínimo cuando la variable aleatoria solo puede tener un valor. Para el conjunto S , la entropía es

$$H(S) = -f_{ID3}^S \ln f_{ID3}^S - (1 - f_{ID3}^S) \ln (1 - f_{ID3}^S) \quad (2.6)$$

Cuando se parte S según las características j y el umbral t , la entropía de S_- y S_+ es la suma ponderada de ambos:

$$H(S_-, S_+) = \frac{|S_-|}{|S|} H(S_-) + \frac{|S_+|}{|S|} H(S_+) \quad (2.7)$$

Entonces, con el algoritmo ID3, en cada nodo hoja se encuentra una división que minimiza la entropía dada por la ecuación 2.6 o se detiene el proceso.

2.8.3 Regresión Lasso

Es una mejora de la regresión lineal que simplifica el modelo mediante un procedimiento de contracción de algunos parámetros hacia cero y selección de características, imponiendo una restricción o una penalización sobre los coeficientes de regresión. Al proceso de reducción de la complejidad del modelo se le conoce como regularización.

Considerando el caso de la regresión lineal y su función objetivo definida por (2.1) se debe añadir el elemento de penalización tal que,

$$\min_{\mathbf{w}, b} C|\mathbf{w}| + \frac{1}{N} \sum_{i=1}^N (f_{\mathbf{w}, b}(\mathbf{x}_i) - y_i)^2 \quad (2.8)$$

Donde $|\mathbf{w}| \stackrel{\text{def}}{=} \sum_{j=1}^D |w^{(j)}|$, $w^{(j)}$ son los componentes de $\mathbf{w}$ en cada dimensión j y C es un hiperparámetro que cuantifica la importancia de la regularización. Con un valor $C = 0$, (2.7) vuelve a tomar la forma de (2.1). Si C asume un valor muy elevado, muchos $w^{(j)}$ tenderán a valores muy pequeños o cero para minimizar la función objetivo. Entonces se debe encontrar un valor óptimo de C para no caer en el subajuste o sobreajuste del modelo.

2.8.4 Redes Neuronales Artificiales

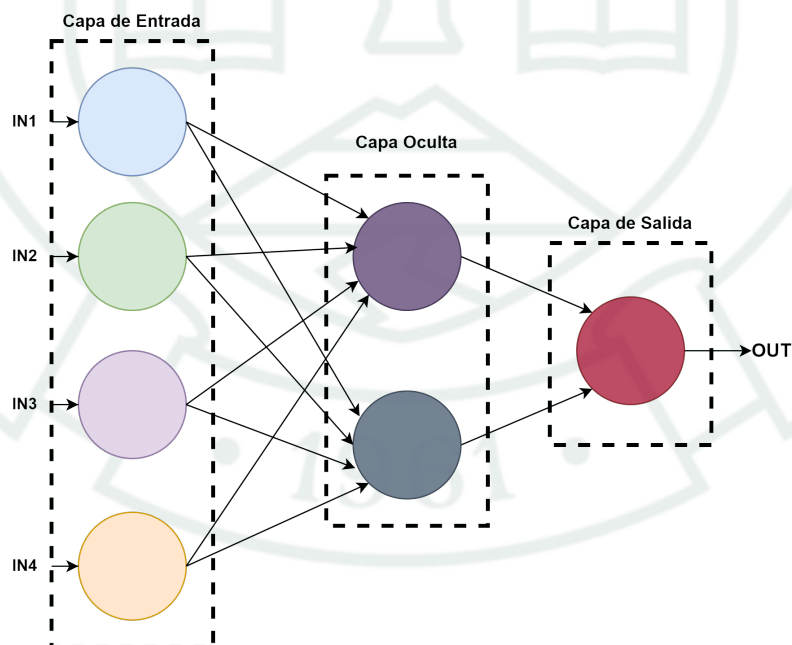
Siguiendo con la definición dada en la introducción de esta sección, una red neuronal es un modelo matemático no lineal que consta de tres capas (Ponce, 2010):

- Capa de entrada: Es la que recibe los datos y los transmite a la capa siguiente a través de conexiones ponderadas, también llamados pesos.
- Capas ocultas: Se encargan de procesar la información asociando pesos a cada conexión a través de reglas de propagación y funciones de activación, que establecen un nuevo nivel de activación basándose en la entrada efectiva y la activación anterior.
- Capa de salida: Aquí se obtiene la respuesta del sistema a la interpretación de las variables de entrada.

En la figura 2.11 se puede observar cómo se representa gráficamente una red neuronal artificial a partir de sus tres capas principales.

Figura 2.11

Capas de una red neuronal



Nota. Elaboración propia basada en Ponce (2010).

Matemáticamente, también para el caso de la figura anterior, se puede escribir como una función anidada $f_{NN}(x)$:

$$y = f_{NN}(x) = f_2(f_1(x)) \quad (2.9)$$

f_2 es una función vectorial de la siguiente forma:

$$f_l(\mathbf{z}) \stackrel{\text{def}}{=} g_l(\mathbf{W}_l \mathbf{z} + \mathbf{b}_l) \quad (2.10)$$

Donde l es el índice de la capa, g_l es la función de activación y es generalmente no lineal, permitiendo a la red ampliar su capacidad de aprendizaje a patrones más complejos. Los parámetros W_l y b_l son una matriz y un vector, respectivamente; para cada capa se aprenden utilizando el familiar descenso de gradiente, optimizando, según la tarea, una función de costo particular. En cuanto a f_1 , esta es una función escalar, pero también puede ser vectorial, dependiendo de los objetivos deseados.

Existen muchos tipos de redes neuronales, de acuerdo a su topología, las más comunes son:

- Redes neuronales Feed-Forward(FNN)
- Redes neuronales recurrentes(RNN)
- Redes neuronales convolucionales(CNN)
- Redes neuronales recurrentes con puertas(GRU)
- Redes Neuronales Recurrentes a Largo Plazo(LSTM)

2.9 Estado del Arte

A continuación, se presenta diferentes estudios que abordan la evaluación de la calidad de aire en interiores, a través del uso de diferentes tecnologías como inteligencia artificial, internet de las cosas (Internet of things, IoT) y sensores de bajo costo.

2.9.1 Recopilación de trabajos sobre el uso de las tecnologías de bajo costo para el monitoreo de calidad de aire en interiores.

En el artículo Application of the low-cost sensing technology for indoor air quality monitoring: A review, Sá et al. (2022) realizan una revisión exhaustiva de diversos estudios sobre detección de contaminantes del aire en interiores aplicando tecnologías de bajo costo. El objetivo de esta recopilación es seleccionar aquellas investigaciones en las que se hace una comparación de sensores de bajo costo y dispositivos de referencia.

Primero señala la importancia de la medición de diferentes parámetros ambientales para evaluar los valores de IAQ, tomando como referencia las indicaciones de la Organización Mundial de la Salud para la protección de la salud pública de los riesgos de la exposición a contaminantes comunes en interiores tales como partículas de materia PM2.5 y PM10, ozono (O₃), dióxido de nitrógeno (NO₂), dióxido de azufre (SO₂), monóxido de carbono (CO) entre otros; además de hacer énfasis en la necesidad del monitoreo en tiempo real de estas sustancias. Los criterios para la selección de los trabajos más relevantes fueron:

1. Que los dispositivos tengan un costo menor a 2000€ en el caso de múltiples sensores y de 500€ para uno solo.
2. Evaluar al menos un tipo de contaminante.
3. Usar sensores disponibles comercialmente.
4. Las mediciones deben ser hechas en ambientes interiores o exteriores, en el caso de haber sido realizadas en un laboratorio, se debe simular un entorno real.

Se excluyeron todos los trabajos en los que no se haya hecho una comparativa con dispositivos de referencia.

Considerando los criterios anteriores, los autores obtuvieron 42 investigaciones, de las que 11 se realizaron en laboratorios y 31 en diferentes ambientes interiores. Un 59 % de los estudios evaluó la concentración de partículas, 18 % midió el CO₂, 8 % el CO y 3 % el NO₂. La

razón de que se evalúe mucho más las partículas PM por parte de los investigadores es debido a que es el principal contaminante en interiores y exteriores.

Acerca de los ambientes donde se realizaron los experimentos, los más recurrentes fueron oficinas y cocinas, pero se hace hincapié en la necesidad de que se apliquen mediciones en centros de salud, vehículos e instalaciones deportivas cerradas.

Finalmente, resaltan la buena concordancia entre los sensores de bajo costo y los dispositivos de referencia, así como también, la importancia de establecer un método de calibración adecuado, ya sea por el método de comparación o incluso a través de un modelo inteligente capaz de calibrar los dispositivos de forma continua a partir de los datos obtenidos.

2.9.2 Implementación de sistema de largo alcance basado en LoRaWAN para monitoreo de calidad de aire

Jabbar et al. (2022) diseñaron un dispositivo de monitoreo de calidad de aire exterior utilizando tecnología IoT que permite recopilación de datos en tiempo real en áreas remotas interconectadas a través de módulos LoRaWAN.

Para la selección de los sensores, los investigadores tomaron como referencia lo estipulado por la United State Environmental Protection Agency (USEPA) sobre la importancia del monitoreo constante de las partículas PM, ozono troposférico (O_3), CO, SO_2 , NO_2 y plomo.

El sistema electrónico consta de un microcontrolador Arduino UNO, un nodo LoRa, un Gateway LoRaWAN, los módulos DHT11, MQ135, MQ136, MiCS 4514, MQ9 y un sensor de partículas de polvo PM2.5. Para la alimentación de energía al dispositivo se empleó un panel fotovoltaico y una batería de 3.7V. Con respecto a la implementación de servicio IoT se ha dispuesto de una conexión a la plataforma The Things Network (TTN) y para la visualización de los datos se ha utilizado ThingSpeak y Virtuino.

Los seis sensores están conectados al Arduino Uno, previa calibración individual; cada nodo LoRa está conectado al Gateway usando una topología de estrella. La configuración de la comunicación comienza con la solicitud de los nodos a los gateways para establecer el enlace, luego este activa la red y los nodos proporcionan la información de red requerida (Dirección

del dispositivo, clave de sesión de red, etc). Finalizada la configuración, comienza la recolección de las mediciones de los sensores, las que son mostradas en dashboards (series de tiempo y gráficos gauge) de TTN, ThingSpeak y Virtuino.

Para la validación de los datos se utilizó el método de comparación con un dispositivo portátil de monitoreo con las mismas capacidades de medición. No se observaron diferencias considerables entre los sensores de bajo costo y el módulo de referencia.

Dos conclusiones en las que los investigadores resaltaron fueron que se debió evitar enviar datos cada 10s por algún método de optimización, ya que esto genera una gran cantidad de datos innecesarios y que también sería de gran utilidad aplicar alguna técnica de predicción de futuros valores de IAQ tomando como base los datos recolectados anteriormente.

2.9.3 Modelado de partículas de materia PM2.5 con Árbol del sistema de inferencia difusa, dinámica adaptativa y red de sensores IoT

El paradigma que presentan Saini et al. (2022) pretende utilizar la técnica de Árbol de sistema de inferencia difusa dinámica (Adaptative Dynamic Fuzzy Inference System Tree, ADFIST) para predecir valores de partículas PM2.5 en interiores, así como monitorear estos valores en tiempo real a través de la tecnología IoT.

Para la experimentación se utilizaron sensores de PM2.5, PM10, CO₂, CO, NO₂, compuestos orgánicos volátiles totales (Total Volatile Organic Compounds,tVOC), temperatura y humedad. La recolección de datos en tiempo real se hizo con un Arduino UNO conectado a un ESP8266 que sirve de gateway para el envío de información al servidor de ThingSpeak.

El pre-procesado de los datos consistió en aplicar el método de rango inter cuartil (Interquartile range,IQR) para eliminar valores atípicos u outliers que se observaron como picos abruptos en los gráficos durante las pruebas. Con los valores faltantes se rellenaron utilizando la media de todos los valores leídos y se aplicó la estandarización de los datos.

Con el acondicionamiento de los datos hecho, el entrenamiento del ADFIST se realiza con la conjunción de tres algoritmos: reconocimiento de patrones, algoritmo genético y el Árbol de sistema de inferencia difusa. Los autores señalan que la principal ventaja de emplear

este modelo es que está influenciado por el razonamiento humano.

Se monitoreó el IAQ en cuatro sitios urbanos y rurales para probar el rendimiento del sistema en diferentes entornos, también con el objetivo de evaluar su capacidad de predicción. Los resultados obtenidos muestran una correcta capacidad de predicción ($\text{NRMSE}=0.3446$) sin embargo, una limitación importante es que el tiempo de entrenamiento (2 horas y 37 minutos) y de predicción (2 minutos y 51 segundos) son elevados a pesar de la optimización del proceso.

2.9.4 Monitoreo de calidad de aire interior para mejorar la eficiencia de los sistemas de ventilación usando Redes neuronales convolucionales totalmente conectadas.

En el artículo Rapid monitoring of indoor air quality for efficient HVAC systems using fully convolutional network deep learning model, Shin et al. (2023) plantean el uso de redes neuronales convolucionales totalmente conectadas (Fully Convolutional Network, FCN) como alternativa a las redes neuronales profundas (Deep Neural Networks, DNN) para predecir rápidamente la edad media del aire (Mean Age of Air, MAA) a través de un modelo de entrenamiento de imagen a imagen basado en datos.

Los autores definen el MAA como un parámetro de IAQ que se refiere a la vida media útil de aire que representa el nivel de frescura del aire, así como también sirve de indicador de áreas mal ventiladas. Debido a que no hay sensores que midan el MAA directamente, la obtención de este valor se hace a través de simulación computacional de fluidos y esto representa un gran costo computacional y de tiempo.

El algoritmo de medición de MAA se aplicó a un contenedor de cadena de frío en el que se dispusieron cámaras a fin de obtener imágenes 2D de la geometría interior. También se compararon diferentes funciones de pérdida durante el proceso de entrenamiento y se evaluó la predicción de MAA con métricas tales como el error medio absoluto (mean absolute error, MAE) y error medio cuadrático (root mean square error, RMSE). Los parámetros de entrada fueron las dimensiones espaciales de la cámara frigorífica y dos cargas que se colocaron en diferentes puntos para probar los resultados del entrenamiento.

Finalmente, se observó que al comparar la performance de las DNN y FCN, los autores concluyen que la red FCN es mejor en varios niveles, por ejemplo, cambios de geometría, variación de la posición de las cargas o incluso al aumentar una carga no es necesario modificar la estructura de la red. Otro punto importante es la importante disminución del MAE y RMSE (-43.14 % y -34.77 %, respectivamente) comparado con los de la DNN.

2.9.5 Plataforma modular IoT para monitoreo de calidad de aire interior en tiempo real

Benammar et al. (2018) implementaron un sistema IAQ modular y escalable que sondea valores de CO₂, CO, SO₂, NO₂, O₃, Cl₂(Cloro), temperatura y humedad; los que son enviados a una plataforma web IoT tomando énfasis en la seguridad de transmisión de cada nodo a fin de evitar pérdida de datos y reenvío de paquetes en caso de que la comunicación sea interrumpida.

A nivel físico, cada nodo consta de un módulo de sensores, un microcontrolador ATmega1281, un transmisor XBee Pro S2 para enviar los datos al Gateway, una memoria SD de 2GB además de una batería recargable de 6600mAh para mantener el funcionamiento en caso el suministro eléctrico sea interrumpido. El gateway está conformado por una computadora raspberry pi 2B, un XBee Pro S2 para la comunicación con los nodos y una memoria microSD de 8GB.

El firmware de los nodos sensores tiene dos modos: Calibración y medición. En el modo de calibración, los sensores, previamente colocados en una cámara especial, leen los valores de una cantidad de gas inyectada en la cámara a fin de realizar un ajuste polinomial entre ambos; entonces para cada sensor se almacena una ecuación en el nodo para la corrección de lectura. Cuando el módulo cambia a modo medición, el microcontrolador reconoce los sensores conectados a través del código serial de cada uno, luego espera 90s para que calienten, toma cinco lecturas de cada uno y guarda el promedio para aplicar la ecuación de calibración a este valor. Entonces los datos son almacenados en la memoria SD y son enviados al gateway. Finalmente, el nodo entra en suspensión durante 13min luego se activa y se repite el ciclo.

Si el envío de datos es interrumpido, los datos almacenados en la SD son leídos du-

rante el intervalo de suspensión del nodo y luego de haber sido introducidos en el servidor principal, son borrados de la memoria.

Para evaluar la pérdida de datos, el sistema ha utilizado unicast y broadcast como forma de transmisión. En broadcast cada nodo envía datos tres veces, el gateway no transmite ningún mensaje de recepción de información. Con unicast el nodo envía un paquete y espera confirmación, si no se da, se repite el proceso por 5s, momento en el que se considera como perdido. Cabe resaltar que el ambiente utilizado es una biblioteca y los nodos están dispersos a unas distancias de entre 30 y 60m del Gateway, con obstáculos como vidrio, estantes o paredes de concreto. En general, el método broadcast mostró mayor pérdida de paquetes que unicast, pero unicast es hasta 50 % más lento.

Acercas de la concentración de gases medidos en diferentes ambientes (biblioteca, laboratorios, oficinas de la Universidad de Qatar) los investigadores observaron una alta correlación entre CO y NO₂. El CO₂ depende de la cantidad de personas haya en los recintos, aunque también se observó un ligero aumento durante los periodos de limpieza. En el caso de SO₂ y O₃, los sensores utilizados no fueron necesarios debido a las bajas concentraciones de estos gases en los diferentes ambientes, a excepción del laboratorio, donde, debido a los experimentos que allí se realizan, los niveles de SO₂ y O₃ fueron mas altos, al igual que con el Cl₂.

El estudio concluye que la selección de parámetros IAQ depende de los ambientes donde se va a realizar el estudio y que la escalabilidad es una propiedad fundamental para la adaptación del sistema a cada situación.

2.9.6 Detector de IAQ con diferentes interfaces de comunicación basado en IoT

La propuesta de Zhao et al. (2019) consiste en medir parámetros IAQ que incluyen temperatura, humedad, CO₂, PM2.5 y formaldehído; usando sensores de bajo costo y diferentes protocolos de comunicación IoT con el propósito de que el dispositivo se adapte a diversos escenarios.

El hardware está compuesto de 4 módulos de comunicación inalámbrica: wifi, LoRa, NB-IoT y GPRS; además de ModBus para interacción por cable. Un uC STM32F103C8T6 se

encarga del procesamiento y control del envío de datos, los que se codifican en formato Mod-Bus RTU y JSON. Debido a que el consumo de GPRS puede llegar a 3 A, se seleccionó un chip MP1584 que proporciona 5 V para los sensores y módulos; el uC opera a 3.3 V, por lo que se adicionó un módulo ASM1117 para su alimentación. Ya que el prototipo consta de 5 modos de comunicación, también se implementó un multiplexor (BL1532) para intercambiar entre los diferentes protocolos. Los sensores seleccionados son SHT30 para temperatura y humedad, PMS5003 para el polvo, S80053 mide CO₂ y WZ-S-K detecta la concentración de formaldehído. Según los autores, estos sensores no necesitan calibración debido a su alta precisión, lo que también motivó su elección.

Para el software se diseñó dos etapas: primero se recolectan y procesan los datos desde el microcontrolador, donde además se codifican para el protocolo ModBu RTU y MQTT. La segunda etapa es en la nube, se dispone de dashboards y gráficas en las que se visualizan las mediciones en tiempo real, también posee otras capacidades como configuración de alarmas.

Se colocaron tres medidores de IAQ en un aula de la Universidad de Tecnología de Dalian durante una semana, cada uno usando uno de los tres protocolos de comunicación que se implementaron y enviando datos cada dos minutos. En cuanto a los parámetros medidos se observó algunos picos en las gráficas debido a la alta sensibilidad de los sensores, el de formaldehído mostró variaciones por la exposición a alcohol o perfumes, mientras que los de CO₂ y PM2.5 no, aunque este último sí tuvo mayor fluctuación cuando se abrió alguna ventana.

Sobre la evaluación de pérdida de paquetes en las conexiones inalámbricas, esta se realizó en diferentes pisos del edificio de innovación de la universidad, incluidos los pisos subterráneos; en general, todos los protocolos funcionaron bien con una pérdida de datos menor al 3 %, aunque cuando se testeó LoRa en el piso más bajo del sótano, la pérdida llegó al 40.9 %, los autores señalan que esto se debe a la complejidad de la construcción bajo tierra.

2.9.7 Deep learning para predicción de calidad de aire en interiores

Ahn et al. (2017) desarrollaron un algoritmo de predicción de calidad de aire basado en memoria a corto plazo (long short-term memory, LSTM) y unidades recurrentes cerradas

(gated recurrent units,GRU) como alternativa a la regresión lineal simple.

Los sensores seleccionados miden CO₂, partículas de polvo, temperatura-humedad, luz y compuestos orgánicos volátiles(volatile organic compounds,VOC). A través de una arquitectura IoT, cada minuto los datos son enviados desde un microcontrolador Arduino UNO y un ESP8266 a un servidor en los que son almacenados en una base de datos SQL. La recolección de información se dio durante siete meses durante tres periodos del día: amanecer (7:30 a.m. a 12:30 p.m.), tarde (12:30 p.m. a 6:30 p.m.) y atardecer (6:30 p.m. a 7:30 a.m.).

Los resultados obtenidos muestran que el modelo GRU es más eficiente que LSTM y la regresión lineal simple, cada uno obtuvo 84.69 %, 70.13 % y 60.96 %, respectivamente. Además, los autores probaron con éxito un algoritmo capaz de predecir los periodos de tiempo de observación más óptimos, lo que permitiría identificar factores externos que afectan la calidad del aire.

2.9.8 Medidor portátil de IAQ con sensores de bajo costo y plataforma IoT de código abierto

En el artículo Real-time monitoring of indoor air quality with Internet of Things-Based E-Nose, Taştan y Gökozan (2019) muestran la implementación y prueba de una “e-nose” capaz de cuantificar la presencia de diferentes gases contaminantes en interiores utilizando sensores de bajo costo y la plataforma de código abierto “Blynk” para mostrar los resultados en tiempo real.

El sistema consta de los sensores GP2Y1010AU (polvo), MH-Z14 (CO₂), MICS-4514 (NO₂ y CO) y DHT22 (temperatura y humedad), una batería de 5000 mAh y el microcontrolador ESP32 que sirve tanto para el procesamiento como para el envío de datos a la plataforma online. Los autores indican que el tiempo de funcionamiento aproximado del prototipo es de 30 h, lo que facilita medir IAQ en diferentes lugares. No se incluyó un método de calibración o comparación con una referencia para asegurar una correcta validación de los datos medidos.

La recolección de datos se hizo en una habitación de 25m² de casa sin ningún sistema de limpieza o ventilación automático durante tres periodos del día durante 4 días. Un análisis

sis de la correlación lineal entre cada parámetro medido mostró diferentes relaciones entre los gases en cada periodo del día. En el periodo 1 (00:00 a 07:30) hubo una correlación positiva entre CO-CO₂, NO₂-humedad y NO₂-temperatura en valores de 0.943, 0.871, 0.755, respectivamente. Temperatura y humedad tuvieron una fuerte correlación negativa con CO-CO₂ y positiva con NO₂. En general, en el período 1 hubo una correlación positiva y negativa elevada entre los gases y los parámetros climáticos. En el periodo 2 (07:30 a 16:30) hubo una correlación positiva entre CO-CO₂, CO-NO₂ y NO₂-CO₂ de 0.966, 0.881, 0.861, respectivamente, y una correlación negativa de 0.915 para temperatura-humedad. Hubo fuertes correlaciones negativas entre el tiempo, CO, CO₂, humedad y NO₂ en -0.844, -0.752, -0.735 y -0.669 durante este período. En cuanto a las correlaciones entre PM10, gases y los parámetros climáticos fueron muy bajas. Finalmente, en el periodo 3 (16:30 a 00:00) la correlación fue positiva entre CO-NO₂ y CO-CO₂ con valores de 0.833 y 0.671, respectivamente. También hubo una correlación positiva entre tiempo-CO, tiempo-NO₂, tiempo-CO₂ y tiempo-PM10 en 0.968, 0.846, 0.622 y 0.407, respectivamente.

A partir del análisis de correlación entre los parámetros ambientales, los autores concluyen que el IAQ depende de las actividades y la cantidad de personas que se encuentran en el edificio.

2.9.9 Prevención de COVID-19 con un sistema inteligente basado en IoT para el control de calidad de aire

En el trabajo IoT-Based smart air quality control system: Prevention to COVID-19, Goyal (2021) propone el diseño de un módulo con sensores de bajo costo para prevenir la acumulación de contaminantes en el aire, ya que esto potencia el riesgo de infección con el virus de la COVID-19.

El modelo propuesto consta de dos partes: los sensores para medir la concentración de CO, CO₂, humo, polvo, temperatura y humedad, estos conectados a un conversor analógico-digital (ADC) de tal manera que los datos puedan ser leídos por una microcomputadora Raspberry Pi y esta los envíe, a través de un módulo LTE, a un servidor en la nube. El servidor re-

moto es el segundo componente del sistema; su función es almacenar y mostrar los datos en tiempo real; también envía una alerta al usuario si la presencia de alguno de los gases ha sobrepasado un umbral máximo permitido.

Sobre la prevención de la expansión del coronavirus, el autor remarca que una alerta temprana de contaminación del aire, en este caso a través de un mensaje SMS, es muy útil para que el usuario final pueda ejecutar las acciones necesarias de ventilación en el ambiente donde se esté monitoreando polución.

2.9.10 Recopilación de trabajos sobre el impacto del IAQ en ambientes críticos

Chamseddine et al. (2025) hicieron una revisión sistemática y exhaustiva de publicaciones científicas Q1 y Q2 entre 2019 y 2024 en las que se evalúa contaminantes, efectos en la salud, factores ambientales y estrategias de mitigación en ambientes críticos como hospitales, escuelas y edificios residenciales con el fin de establecer un guía para la gestión de la calidad de aire interior.

De acuerdo con los trabajos revisados, los autores clasifican los contaminantes según su tipo, origen e impacto.

- Partículas en suspensión (PM10, PM2.5). Proveniente de combustión, polvo y actividades domésticas. Es causante de enfermedades respiratorias y cardíacas.
- Compuestos orgánicos volátiles (VOCs). Es generado por pinturas, disolventes y productos de limpieza. A nivel de salud, los VOCs son carcinógenos y mutágenos.
- Gases químicos, tales como CO, CO₂, NO₂, SO₂, benceno, formaldehído y ozono. Surgen por combustión, cocción, calefacción, materiales de construcción y otros.
- Contaminantes biológicos. Bacterias, virus, hongos o patógenos; que pueden ser segregados por la actividad humana, así como también por la humedad y ventilación deficientes. Causan infecciones respiratorias y alergias

A partir de la clasificación de los contaminantes, se infieren efectos negativos a nivel fisiológico (enfermedades cardiovasculares, cáncer, etc.), cognitivos-psicológicos (fatiga, in-

somnio, ansiedad y depresión) y socioeconómicos (menos rendimiento en actividades cotidianas)

Con respecto a los ambientes críticos, los autores analizan estudios de IAQ en hospitales, escuelas y viviendas para mostrar que el diseño, edad del edificio, clima, sistemas de ventilación, mobiliario y la actividad humana, son los principales factores que influyen en el índice de calidad de aire.

Para finalizar, proponen un marco integral de gestión de IAQ basado en monitoreo continuo, ventilación adaptativa, integración de inteligencia artificial y sistemas de automatización para mejorar la calidad de aire interior.

2.9.11 Simulación del índice de calidad de aire y ventilación en edificios comerciales e institucionales

El objetivo de este estudio fue evaluar, mediante simulación, el comportamiento del IAQ y su relación con la ventilación, en diferentes tipos de edificios y horarios de uso por los ocupantes (Persily et al., 2003). Los contaminantes empleados en las simulaciones fueron CO₂ y VOC.

Las simulaciones se desarrollaron mediante un software desarrollado por el National Institute of Standards and Technology (NIST) que modelaba el flujo de aire, dispersión de contaminantes en edificios y otros elementos como ventilación mecánica, emisión de contaminantes por fuentes internas y sensores.

Entre los recintos simulados incluyen oficinas, aulas universitarias y salas de conferencia. Se consideraron cuatro características en cada uno: Área, altura del techo, ocupantes y horarios de uso. Todos los espacios fueron tratados independientemente con tasas de ventilación controladas según diferentes estrategias. Los datos obtenidos se extrapolaron a ciudades con diferentes climas.

Los resultados muestran una mejora sustancial en el consumo energético de los sistemas de ventilación dependiendo del tipo de espacio, clima y ocupación. Los mayores ahorros se lograron en climas con alta variación térmica. Los incrementos de CO₂ fueron moderados

y menores a 1000 ppm, concluyendo que las estrategias con ventilación variable respondieron bien a la tasa de ocupantes, evitando acumulación prolongada. En cuanto a los VOCs, el estudio señala que se incrementaron con ventilación reducida y que, en general, los contaminantes no dependientes de la ocupación son los que más contribuyen a la degradación del IAQ.

El modelo muestra que un control proporcional de la ventilación reduce la concentración de los contaminantes, favoreciendo la eficiencia energética y mejores valores de calidad de aire en los recintos simulados.

2.9.12 Análisis de métodos de machine learning para predicción de IAQ

Wei et al. (2019) realizaron una investigación exhaustiva acerca de los modelos estadísticos y de machine learning aplicados en la predicción de valores calidad de aire en interiores, además de los parámetros de IAQ que se han estudiado, mostrando así su utilidad en este campo y su potencial como alternativa a los modelos mecanicistas comúnmente aplicados.

Los autores revisaron 37 publicaciones en donde se aplicó ML en el estudio del IAQ. Encontraron que el parámetro más estudiado fue las partículas de materia (PM_{2.5} y PM₁₀), seguido de CO₂ y radón; en cuanto a los modelos predictivos más utilizados se tiene a las redes neuronales artificiales (Artificial Neural Networks, ANN), regresión lineal múltiple, regresión de mínimos cuadrados parciales (Partial Least Squares Regression, PLSR) y árboles de decisión. El criterio de análisis de los estudios revisados fue:

1. Objetivo de estudio.
2. Información detallada de las mediciones, locación y periodo de muestreo.
3. Diseño del modelo estadístico.
4. Métricas de performance de cada modelo aplicado.

Acerca de los datos de entrenamiento, los autores señalan que los datos utilizados en los modelos fueron recolectados en diferentes ventanas de tiempo, la razón de esto se debe a que los modelos predictivos realizan una estimación espacial de un parámetro basado en el conocimiento de otros parámetros y no de datos de series de tiempo.

Otro aspecto relevante observado en la revisión es que la implementación de modelos estadísticos en análisis de calidad de aire ha sido más común en los experimentos hechos en el exterior que en interior. De los 37 estudios observados, los lugares más comunes donde se ha aplicado los modelos estadísticos fueron viviendas (16), oficinas (5), vehículos (9), escuelas (3), y otros tipos de edificios (4).

En cuanto a los datos recolectados de la lectura de los diferentes parámetros, 28 estudios usaron datos transversales y 9 estudios también usaron datos muestreos de anteriores tiempos como entradas. ANN se aplicó en 18 publicaciones, modelos de regresión en 19 y modelos de regresión basados en árboles de decisión en 4.

En los trabajos revisados donde se aplicó ANN las variables de entrada utilizadas eran dependientes el tiempo o categóricas independientes (dimensiones o materiales de construcción del recinto). Las variables de salida estudiadas son PM_{2.5}, PM₁₀, CO, CO₂, NO₂, etc. Se notó que las redes neuronales con retro propagación tenían un R^2 más alto que el perceptrón multicapa, 0.65-0.79 y 0.39-0.62 respectivamente; aunque los autores indican que esto no significa que la red neuronal con retro propagación sea más eficiente que la otra basándose en que no comparten los mismos datos de entrenamiento o que tengan el mismo número de observaciones. Específicamente sobre la comparación entre tipos de redes neuronales se remarca la contribución de Loy-Benitez et al. (2019) quienes analizaron tres; el objetivo era la predicción de PM_{2.5} en una estación de metro en Seúl y los resultados fueron que el modelo de unidad recurrente cerrada tuvo el rendimiento más alto con $R^2=0.65$, mientras que el perceptrón de una capa obtuvo $R^2=0.62$.

En general, para la predicción de partículas de materia, los modelos revisados tuvieron un rendimiento aceptable ($R^2=0.62-0.79$); los parámetros utilizados como entrada fueron diversos: velocidad del viento, concentración de PM en el exterior, temperatura, humedad, etc.; siendo la concentración de PM exterior la principal influencia para la robustez del método estadístico aplicado (Das et al., 2014). Sin embargo, considerar este tipo de entradas es una tarea complicada y requiere de más estudios teniendo en cuenta que los parámetros interiores también son relevantes.

Los estudios revisados muestran que el problema principal de aplicar ANN en IAQ y obtener una predicción eficiente es la selección de variables de entrada. Entonces se debe hacer un preanálisis de los datos aplicando, por ejemplo, regresión o análisis de componentes principales.

Diferentes modelos de regresión han sido aplicadas para la predicción de PM o NO₂ en interiores como escuelas, viviendas o estaciones subterráneas. Nuevamente, los datos de entradas son fundamentales para un correcto funcionamiento del algoritmo, esto fue observado en el trabajo de Elbayoumi et al. (2015), en el que los parámetros de entrada fueron PM2.5 exterior y humedad relativa y el algoritmo fue regresión lineal múltiple (Multiple Linear Regression, MLR), dando como resultado un $R^2=0.59$, mientras que con el mismo algoritmo (MLR) pero usando como entradas PM2.5, PM10, tasa de ventilación, velocidad del viento y temperatura interior; el coeficiente de determinación aumentó a 0.69. Otro estudio evaluó la MLR, regresión LASSO (Least Absolute Shrinkage and Selection Operator) y la regresión por pasos sobre el mismo conjunto de datos; mostrando los tres algoritmos una eficiencia similar.

Ya que la regresión logística multivariable suele ser el modelo más aplicado de los de su tipo, los autores encontraron trabajos que comparan MLR y ANN, los cuales revelaron un valor de error medio cuadrático más alto para las MLR durante el entrenamiento, mientras que en el periodo de validación para ambos métodos, este valor aumentó.

Sobre PLSR, los autores muestran tres experimentos realizados en una estación subterránea de metro en Seúl en los que utilizaron este método estadístico para predicción de PM2.5 y PM10. Los datos de entrada fueron parecidos a los que se aplicaron en ANN's. Durante el entrenamiento, los valores de RMSE fueron más altos con PLSR, en comparación con MLR, pero inferiores durante la etapa de prueba.

El método de árboles de decisión se aplicó en cuatro estudios para predecir radón, virus en el aire y partículas de materia. Los resultados obtenidos fueron $R^2 = 0.74-0.94$ y $R^2 = 0.33-0.49$ para entrenamiento y validación, respectivamente; la brecha entre los valores del coeficiente de determinación en ambas etapas se debe, según los autores, a que los modelos de entrenamiento requieren mejoras. En otras publicaciones se observó un decremento de R^2 al

comparar la performance de árboles de decisión y MLR, lo que implicaría que los árboles de decisión son menos robustos que MLR al emplearse en estos casos de estudio.

De los estudios revisados, los autores identifican 3 aspectos a considerar para la mejora de las predicciones de IAQ:

- Ya que muchos métodos de predicción pueden aplicarse en IAQ y considerando que cada uno tiene ventajas y desventajas, lo más adecuado es probar varios a fin de determinar el mejor para el caso objetivo.
- La correcta selección de los datos de entrada permitirá establecer la varianza de la salida.
- Los datos obtenidos en un tipo de edificio solo deberían aplicarse en lugares similares debido a que estos son influenciados por factores como dimensiones, ubicación o cantidad de ocupantes.

En conclusión, los autores sugieren que promediar las salidas de diferentes modelos estadísticos puede aumentar la robustez de las predicciones. También resaltan el potencial de los sensores de bajo costo para el objetivo de facilitar la obtención de datos.

2.9.13 *Apreciaciones Finales*

El apartado 2.9.1 muestra un panorama general sobre las investigaciones actuales de medición de calidad de aire con sensores de bajo costo. Los criterios que Sá et al. utilizan para seleccionar los estudios pueden tomarse como referencia para definir qué es un dispositivo de medición de bajo costo y cómo deben escogerse de acuerdo al entorno donde se va a experimentar. También proponen métodos de calibración, indispensables para una medición fiable. En general, este trabajo es un excelente punto de partida para el diseño de sistemas de IAQ de bajo costo.

Siguiendo con la dinámica de diseño presentada en el apartado anterior, en la sección 2.9.2 se revisó el trabajo de Jabbar et al., quienes crean un módulo de calidad de aire en exteriores equipado, además de sensores de bajo costo, con el protocolo de comunicación inalám-

brica LoRaWAN e IoT para monitoreo en zonas remotas. Esta investigación muestra cómo se implementa una red estrella entre diferentes nodos interconectados que envían la información a un gateway a fin de mostrarla en tiempo real con un dashboard hecho en la nube.

En 2.9.3 se profundiza más sobre tratamiento de datos; plantea un enfoque de predicción de valores de PM2.5 en entornos cerrados a través de un algoritmo ADFIST. Se detalla aspectos propios del preprocesamiento de datos, análisis de la performance del algoritmo y la presentación de la información al usuario final.

En la sección 2.9.4 se propone un enfoque para predecir la calidad del aire, centrándose en la estimación de la edad media del aire. Este planteamiento implica la comparación de dos tipos de redes neuronales, a saber, FCN y DNN, en términos de su capacidad predictiva. Un detalle interesante de este trabajo es que toma como entradas imágenes bidimensionales de la geometría interior del ambiente donde se hace el experimento y la razón de esto se debe a que el aire es un fluido y como tal tomará la forma del espacio que lo contiene. En el apartado 2.9.7 Ahn et al. también estudia otros algoritmos de predicción aplicados al IAQ: LSTM, GRU y regresión lineal simple; demostrando que LSTM es más eficiente en la predicción. En estos dos trabajos se aprecia cómo es que diferentes métodos de machine learning pueden ser aplicados en el estudio de la calidad de aire, pero que dependen fuertemente de cómo se componen los datos que se emplea para el entrenamiento.

Debido a que un dispositivo IoT debe comunicarse remotamente con un servidor, es posible que haya pérdida de información por diferentes motivos, con esta premisa el dispositivo desarrollado por Benemmar et al. (Sección 2.9.5) propone una solución a este problema potencial. El prototipo diseñado consta de una batería que sirve como respaldo si el servicio eléctrico es interrumpido y una memoria SD para almacenar los datos cuando se pierda el enlace con el servidor donde se almacenan. Otra manera de asegurar la conexión remota es la vista en 2.9.6 donde, gracias a la capacidad de adaptabilidad que puede tener un sistema IoT, se diseña un mecanismo basado en múltiples protocolos de comunicación para asegurar el envío de los valores medidos.

Existen numerosos gases nocivos que pueden ser identificados en el aire, y es posible

que algunos de estos gases presenten relaciones interdependientes. Esta noción es explorada en el artículo sintetizado en la sección 2.9.8. En dicho estudio se procede a evaluar la correlación entre los valores obtenidos a través de los sensores que miden partículas de polvo, dióxido de carbono (CO_2), dióxido de nitrógeno (NO_2), monóxido de carbono (CO) y la temperatura junto con la humedad ambiental. Los resultados de este análisis revelan que las concentraciones de los gases exhiben diversos grados de interrelación dependiendo de la cantidad de personas y las actividades que realicen.

En el artículo de Somya Goyal (Sección 2.9.9), se presenta un sistema de supervisión de la calidad del aire diseñado para mitigar la propagación del coronavirus. La efectividad de este sistema radica en la integración de un módulo de comunicación celular, que se encarga de enviar mensajes de texto al usuario con el fin de alertarlo, permitiéndole así adoptar las medidas adecuadas para mejorar la ventilación del recinto donde se ha instalado el nodo de monitoreo.

La revisión de estudios realizada por Chamseddine et al. (Sección 2.9.10) sobre la calidad de aire en ambientes críticos como hospitales, escuelas y viviendas, evidencia cómo diversos contaminantes inciden en la salud de los ocupantes y cuáles son los factores que favorecen su proliferación. En un esfuerzo de unificar los hallazgos de múltiples investigaciones, los autores plantean un marco integral para la gestión de la calidad de aire, partiendo de los límites de exposición a contaminantes establecidos por la OMS, la EPA o ASHRA. Este enfoque considera el monitoreo continuo de contaminantes, la ventilación adecuada, el mantenimiento y limpieza periódica de los espacios, así como la incorporación de tecnologías inteligentes y sistemas de automatización orientados al control y mejora del ambiente interior.

La simulación de ambientes interiores realizada por el NIST (Sección 2.9.11) constituye una referencia fundamental para el análisis de la calidad del aire interior, al mostrar de manera detallada los procedimientos, parámetros y variables que deben considerarse en la evaluación de un recinto. El estudio permite comprender las fluctuaciones de CO_2 y VOC bajo diferentes condiciones de ocupación y ventilación, y además sirve como guía metodológica para extender el análisis hacia otros contaminantes interiores. Los resultados varían según las

características de cada espacio, tales como dimensiones, número de ocupantes y fuentes de emisión; lo cual confirma la complejidad y variabilidad del comportamiento del aire interior. Finalmente, los autores destacan la importancia de utilizar sensores correctamente calibrados y sistemas de ventilación controlados para garantizar una mayor eficiencia energética sin comprometer la calidad del aire.

La sección 2.9.12 recopila 37 estudios investigados que emplean Aprendizaje Automático en el ámbito de la calidad del aire interior. Los autores evidencian que, aunque diversos algoritmos aplicados demuestran una eficacia aceptable, es esencial seleccionarlos de acuerdo al contexto de implementación. La estrategia más apropiada consiste en compararlos a través de la misma base de datos, permitiendo así observar sus respuestas y determinar cuál es el más idóneo. Adicionalmente, es necesario realizar un análisis preliminar de la información para emplear las variables de entrada más adecuadas.



CAPÍTULO III

Análisis de ambientes representativos y su relación con la calidad de aire interior

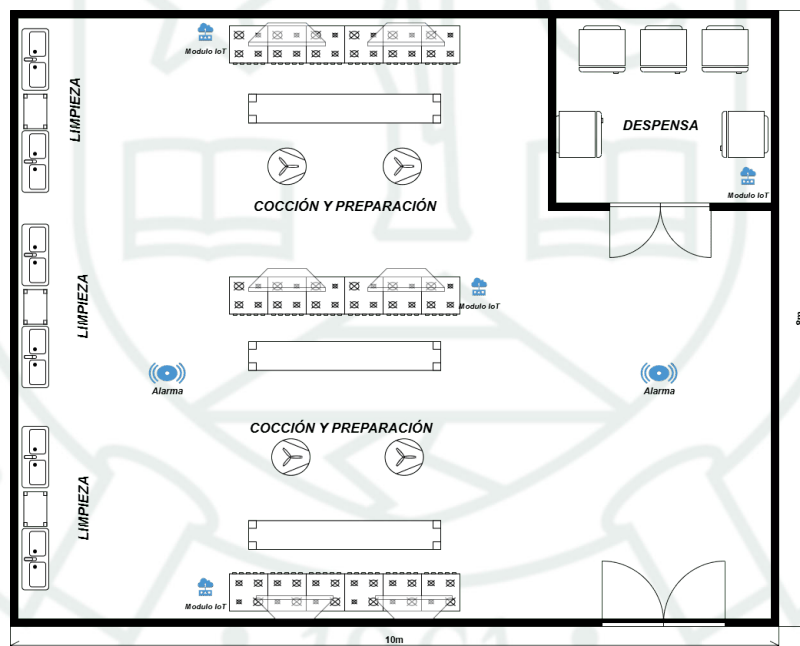
La calidad de aire interior está sujeta al tipo de entorno donde se evalúa (Chamseddine et al., 2025). En la Sección 2.1 se presentó una descripción breve de las características de cuatro ambientes diferentes: una cocina industrial, un taller de mecánica automotriz, un laboratorio de mantenimiento industrial y un gimnasio. En todos ellos se simulará la variación del IAQ considerando las fluctuaciones de cinco gases comunes en cada uno, con comportamientos desiguales. En este capítulo se amplían algunos de los apartados expuestos anteriormente y también se aborda el planteamiento del control del IAQ, así como su impacto en el desempeño de las operaciones y en la salud de los trabajadores.

3.1 Cocina Industrial

3.1.1 Aspectos arquitectónicos

Figura 3.1

Plano de la cocina industrial



Nota. Elaboración propia.

Diseño de la cocina. La figura 3.1 muestra un plano con los sectores propios de este ambiente:

- **Cocción y preparación:** En el centro se tiene una isla donde se prepara los alimentos, hay estufas, campanas extractoras, mesas para cortar y para emplatar. En este sector es donde se encuentra la mayor presencia de gases VOC, CO, CO₂, PM₁₀ y PM_{2.5}.
- **Despensa:** Es un área refrigerada para conservación de los alimentos, requiere de un constante sensado de la temperatura, humedad y VOC por la posibilidad de fugas de refrigerantes como halocarbonos o hidrocarburos.
- **Limpieza:** Aquí se usan detergentes, que liberan VOC y NO₂.

En cuanto a las dimensiones, como se menciona en el Apartado 2.1.1, cuenta con 80m² y una altura de techo de 3m, en la que se instalan las campanas extractoras y los sistemas HVAC.

Ventilación y extracción.

- Campanas extractoras, que se usan para capturar partículas y vapores generados durante la preparación de alimentos; además, cuentan con filtros de grasa.
- Sistemas HVAC, empleados en ventilación forzada a fin de evitar la acumulación de gases nocivos y regular la temperatura ambiental.

3.1.2 Aspectos ambientales

- **Temperatura:** Siendo un ambiente cerrado, la temperatura puede variar de 25 a 35°C dependiendo el sector, tipo de actividad y horas pico. Los hornos, estufas, freidoras y parrillas contribuyen al aumento de la sensación térmica.
- **Humedad:** Puede aumentar por los vapores y los periodos de limpieza. Fluctúa entre 50 y 80 %. Valores altos promueven la formación de hongos y bacterias que afecten la calidad del aire.

3.1.3 Monitoreo y estrategias de control del IAQ

Se instalará un módulo de monitoreo en tiempo real en el área de cocción y preparación, ya que es el principal sector donde ocurre la generación de CO, CO₂, NO₂, VOC y PM.

Las fluctuaciones del IAQ pueden ser observadas por los trabajadores en todo momento en una pantalla instalada en un área de fácil visualización.

La tecnología IoT permite el monitoreo remoto asegurando la integridad del recinto durante el día. Si ocurre algún incendio o fugas de gas fuera del horario laboral, un supervisor puede ser alertado inmediatamente del peligro y tomar las acciones pertinentes.

En caso de valores extremos de algún gas, se cuenta con una alarma que indique la evacuación del personal.

Los sistemas HVAC y las campanas extractoras son controladas de acuerdo con las indicaciones del módulo de monitoreo de IAQ.

3.1.4 Impacto del IAQ en el desempeño

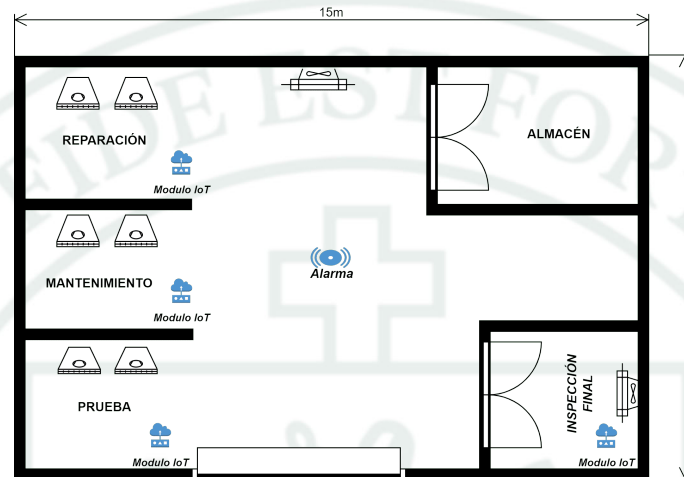
- Cocineros, asistentes y personal de limpieza pueden presentar fatiga, problemas respiratorios, o dolores de cabeza tanto por los gases como las altas temperaturas que soportan, un control eficiente del IAQ mejora sustancialmente la eficiencia de los trabajadores.
- La contaminación cruzada es una constante en este tipo de ambiente por lo que el monitoreo permanente asegura que los alimentos sean óptimos para el consumo humano.

3.2 Laboratorio de mantenimiento industrial

3.2.1 Aspectos arquitectónicos

Figura 3.2

Plano del laboratorio



Nota. Elaboración propia.

Diseño del laboratorio. La figura 3.2 muestra un plano con los sectores propios de este ambiente:

- Zona de trabajo: Subdividida a su vez en reparación, mantenimiento y prueba, es el área donde se realiza la mayoría de las actividades. Se usan herramientas diversas para corte, soldadura, pintura y lubricación; que generan VOC, CO, PM10 y NO₂.
- Zona de inspección final: Donde se da revisión a las maquinarias o dispositivos a los que se hizo mantenimiento para la posterior entrega al cliente. También puede haber presencia de los contaminantes anteriormente mencionados.
- Zona de almacenamiento: Estanterías y gabinetes para el almacenamiento de herramientas, equipos y repuestos.

En cuanto a las dimensiones, como se menciona en el Apartado 2.1.2, cuenta con 150m² y una altura de techo de 5m. El piso es de concreto de alta resistencia y antideslizan-

te, dependiendo de la zona, las paredes y techos tienen paneles metálicos que resisten altas temperaturas y son de fácil limpieza.

Ventilación y extracción. Sistemas HVAC, que renuevan el aire constantemente y filtran polvo, gases y humo.

3.2.2 Aspectos ambientales

- Temperatura: Varía entre 25 y 30°C por el uso de herramientas eléctricas o de soldadura, además de los procesos en los que se utilizan.
- Humedad: Puede aumentar por los vapores y los periodos de limpieza. Fluctúa entre 50 y 80 %. Valores altos promueven la formación de hongos y bacterias que afectan la calidad del aire.

3.2.3 Monitoreo y estrategias de control del IAQ

Se instalará un módulo de monitoreo en tiempo real en las zonas de trabajo y pruebas, ya que son los principales sectores donde se genera CO, CO₂, NO₂, VOC y PM. Las fluctuaciones del IAQ pueden ser observadas por los trabajadores en todo momento en una pantalla instalada en un área de fácil visualización.

Se cuenta con alarmas automáticas que se activan en situaciones donde se tenga exceso de concentración de algún gas.

Los sistemas HVAC son controlados de acuerdo con las indicaciones de los módulos de monitoreo de IAQ.

El cumplimiento de las normativas de seguridad industrial y ambiental es fundamental para un ambiente de trabajo seguro.

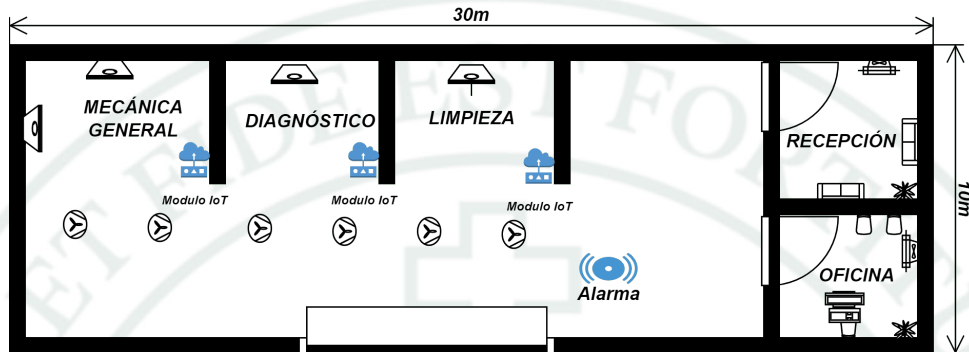
3.2.4 Impacto del IAQ en el desempeño

Ingenieros, técnicos y otros trabajadores pueden tener reducción en la exposición a contaminantes nocivos y enfermedades laborales si existe un control óptimo de la calidad de aire.

3.3 Taller automotriz

3.3.1 Aspectos arquitectónicos

Figura 3.3
Plano del taller



Nota. Elaboración propia.

Diseño del taller. En el plano de la figura 3.3 se observa las tres áreas que lo componen:

- **Mecánica general:** Dedicada a la reparación de motores, sistemas eléctricos y electrónicos, cambios de aceite, revisión de frenos y suspensión. Aquí se generan contaminantes del aire por la quema de combustible, aerosoles y otros químicos.
- **Diagnóstico:** Donde se realizan pruebas de emisiones y medición de gases como CO, NO₂ y VOC.
- **Limpieza:** aquí se realiza el lavado de vehículos y se cuenta con un sistema de tratamiento de aguas residuales. Los productos de limpieza despiden contaminantes como VOC, PM10 y NO₂.

En cuanto a las dimensiones, como se menciona en el Apartado 2.1.3, cuenta con 300m² y una altura de techo de 4 m, portones para acceso de vehículos y áreas cerradas de oficina y espera para los clientes.

Ventilación y extracción.

- Los extractores en el área de diagnóstico son útiles para atrapar los humos residuales durante las pruebas de emisiones.
- Los sistemas HVAC colocados en el área de mecánica general y limpieza facilitan la remoción de partículas PM10 y gases tóxicos.
- Los portones de acceso permiten el flujo del aire exterior y la ventilación natural.

3.3.2 Aspectos ambientales

- La temperatura puede variar entre 20 y 30°C dependiendo de los equipos utilizados o la cantidad de vehículos en reparación.
- En cuanto a la humedad, la media ronda el 70 %, pero es mayor en el área de limpieza, donde alcanza el 90 %.

3.3.3 Monitoreo y estrategias de control del IAQ

Es necesaria la instalación de un módulo de monitoreo de IAQ en las tres zonas del taller y una alarma para alertar concentraciones peligrosas.

El monitoreo puede realizarse desde la oficina mediante una conexión Wi-Fi entre los módulos y el dashboard en un servidor web.

Otras acciones para el control de IAQ pueden ser el uso de cortinas de aire para separar zonas, la inspección regular de los equipos de ventilación y extracción de aire y la eliminación adecuada de residuos.

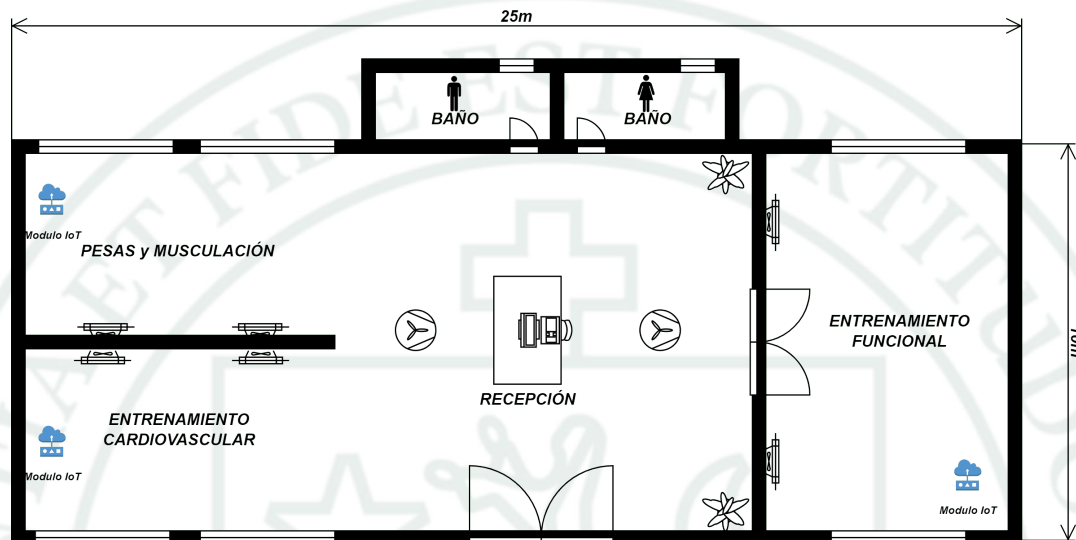
3.3.4 Impacto del IAQ en el desempeño

Conservando la calidad de aire en los mejores niveles posibles, se pueden prevenir síntomas de fatiga, problemas respiratorios, irritación ocular, entre otros. Buena ventilación, monitoreo constante y un correcto mantenimiento de los equipos de trabajo contribuye significativamente a la eficiencia operativa de todo el personal.

3.4 Gimnasio

3.4.1 Aspectos arquitectónicos

Figura 3.4
Plano del gimnasio



Nota. Elaboración propia.

Diseño del gimnasio. El plano adjunto muestra las principales zonas del gimnasio.

- Entrenamiento cardiovascular: Cuenta con caminadoras, bicicletas estacionarias y elípticas.
- Pesas y musculación: Con máquinas de fuerza, pesas y pisos de goma.
- Entrenamiento funcional: Es un espacio amplio para clases de baile, HIIT o tae bo.

Ventilación y extracción.

- Ventilación forzada HVAC en el área de entrenamiento funcional, ya que la aglomeración de grupos de personas haciendo actividad física aumentará bastante los niveles de CO₂.
- Ventilación natural en las zonas de cardio y musculación.

3.4.2 Aspectos ambientales

- Temperatura: Varía entre 20 y 30 °C dependiendo de la época del año y la cantidad de personas.
- Humedad: La alta densidad de personas también aumenta los niveles de humedad; esta fluctúa entre 60 y 80 %.

3.4.3 Monitoreo y estrategias de control del IAQ

El uso de ventiladores de techo y la constante renovación del aire por los sistemas HVAC permite mantener niveles bajos de CO₂ y VOC (durante los periodos de limpieza).

3.4.4 Impacto del IAQ en el desempeño

Altos niveles de CO₂ causan fatiga y cansancio, por lo que mantener un gimnasio bien ventilado y con una cantidad de personas dentro de los límites del aforo permitido contribuirá a mantener condiciones estables y seguras para los usuarios.



CAPÍTULO IV

Diseño e implementación del prototipo

El diseño del sistema de monitoreo de calidad de aire parte de los tres principales apartados del Capítulo 2: contaminantes del aire, internet de las cosas y algoritmos de inteligencia artificial. Los gases que serán medidos son CO, CO₂, NO₂, PM y VOC, además de la temperatura y humedad; la selección de estos compuestos químicos se debe a que son bastante comunes en cualquier tipo de ambiente interior donde exista aglomeración de personas, algún proceso de combustión, polvo, pinturas, aerosoles, moho, alta humedad o valores de temperatura no deseados. También cabe resaltar las relaciones observadas entre ellos cuando se detecta la presencia de cada uno. La tecnología IoT se usará para el proceso de recolección de datos, presentación de la información en un dashboard de monitoreo en la nube y alertar al usuario final en caso de que se haya detectado valores extremos. Finalmente, con respecto a los algoritmos de IA, serán evaluados a fin de determinar el más adecuado al objetivo de controlar la calidad de aire.

En cuanto a la implementación del prototipo de medición de la calidad del aire, se analizará la respuesta de los algoritmos de IA aplicados en la predicción de la IAQ instantánea y se desarrollará una plataforma web que permita acceder a las mediciones en tiempo real. En general, la implementación se divide en tres apartados: Hardware, análisis de algoritmos de IA implementados y Cloud computing.

Considerando los contaminantes que serán medidos, se optó por cuatro ambientes diferentes que tienen en común la presencia de estos gases:

- Cocina industrial: Donde los procesos de combustión y limpieza producen principalmente CO, NO₂ y VOC
- Laboratorio de mantenimiento industrial: Con el uso de maquinaria, procesos de soldadura y sustancias químicas como solventes o pinturas contribuye a la acumulación de CO, PM y VOC en el ambiente, requiriendo estrategias efectivas de ventilación y filtrado.
- Taller automotriz: Los motores de coches, otras maquinarias y el uso de aceites, solventes y productos de limpieza automotriz contribuyen al aumento de CO, NO₂ y PM,

siendo riesgo para los trabajadores si se exponen de manera prolongada.

- Gimnasio: Aunque este ambiente es menos agresivo que los demás, la razón de su inserción en el análisis es la aglomeración de gente causa altos niveles CO_2 , lo que reduce la eficiencia en el ejercicio. Otro compuesto nocivo son los VOC durante los periodos de limpieza.

De acuerdo con el diagrama de flujo mostrado en la figura 1.1 y los objetivos específicos del presente trabajo de investigación, se planteó cómo debería ser implementada cada etapa y ahora se extenderá el detalle de cada elemento:

4.1 Selección de sensores

El principal aspecto tomado en cuenta para la selección de los sensores es que cumplan todo el rango de medición que se requiere para clasificar la calidad de aire desde bueno a peligroso según las normativas EPA y WHO. Al ser un sistema de bajo costo, también es importante que los sensores se mantengan en un rango de precio que permita una fabricación económica pero confiable.

Finalmente, otro criterio que se impuso fue la compatibilidad entre sensores y los sistemas de adquisición.

A continuación se presentan los sensores seleccionados, junto con las características técnicas que justifican su elección.

- Sciosense ENS160: Es un sensor MOX digital diseñado para medir índice de IAQ, VOC y CO_2 (ScioSense B.V., 2023). Sus características son:
 - Control independiente del calentador del sensor para una selectividad máxima a los VOC y una discriminación de fondo excepcional.
 - Alta estabilidad frente a variaciones de temperatura y humedad
 - Bajo consumo energético

La tabla adjunta muestra algunos de los parámetros más importantes del sensor:

Tabla 4.1*Parámetros del sensor ENS160*

Parámetro	Valor
Rango de medición VOC	0-65 000ppb
Rango de medición CO ₂	400 - 65 000 ppb
Protocolo de comunicación	I2C
Voltaje	3.3 V
Consumo de corriente durante medición	150 uA

Nota. Tomado de la hoja de datos del sensor ENS160 (ScioSense B.V., 2023).

- Plantower PMS5003: Mide concentración de partículas de materia (PM) (Plantower, 2025). Sus principales características son las siguientes:
 - Usa tecnología de dispersión láser para medir partículas de 1um, 2.5um y 10um.
 - Bajo ruido y consumo de energía.
 - Amplio rango de medición.

La tabla 4.2 detalla sus parámetros eléctricos.

Tabla 4.2*Parámetros del sensor de partículas Plantower PMS5003*

Parámetro	Valor
Rango de medición PM1.0	0 – 500 µg/m ³
Rango de medición PM2.5	0 – 500 µg/m ³
Rango de medición PM10	0 – 500 µg/m ³
Resolución	1 µg/m ³
Protocolo de comunicación	UART
Voltaje de operación	5 V
Consumo de corriente en medición	< 100 mA

Nota. Tomado de la hoja de datos del sensor (Plantower, 2025).

- ASAIR AHT21: Utilizado en el monitoreo de temperatura y humedad, este sensor digital tiene calibración de fábrica, compensación de temperatura y alta estabilidad (Aosong Electronics Co., Ltd., 2021).

Tabla 4.3*Parámetros del sensor AHT21*

Parámetro	Valor
Rango de medición temperatura	-20 - 80 °C
Rango de medición humedad	0 - 100 ±2 %RH
Protocolo de comunicación	I2C
Voltaje	2.15-5 V
Consumo de corriente durante medición	980 uA

Nota. Tomado de la hoja de datos del sensor (Aosong Electronics Co., Ltd., 2021).

- Winsen ZE15-CO: Es un sensor electroquímico de propósito general que detecta monóxido de carbono (Zhengzhou Winsen Electronics Technology Co., Ltd., 2018). Sus características más relevantes son:
 - Soporta compensación de temperatura.
 - Alta sensibilidad y selectividad.
 - Bajo consumo
 - Respuesta lineal

Tabla 4.4*Parámetros del sensor ZE15-CO*

Parámetro	Valor
Rango de medición CO	0-500 ppm
Protocolo de comunicación	UART y analógico
Resolución	0.1 ppm
Voltaje	5-12 V

Nota. Tomado de la hoja de datos del sensor (Zhengzhou Winsen Electronics Technology Co., Ltd., 2018).

- Winsen MH-Z19B: Es un sensor infrarrojo diseñado para medir CO₂ (Zhengzhou Winsen Electronics Technology Co., Ltd., 2016). Entre las principales características del MH-Z19B resaltan las siguientes:
 - Diseñado para sistemas de calidad de aire interior.

- Bajo consumo.
- Compensación de temperatura y respuesta lineal.
- Alta resistencia a interferencias por vapor de agua.

Tabla 4.5

Parámetros del sensor MH-Z19B

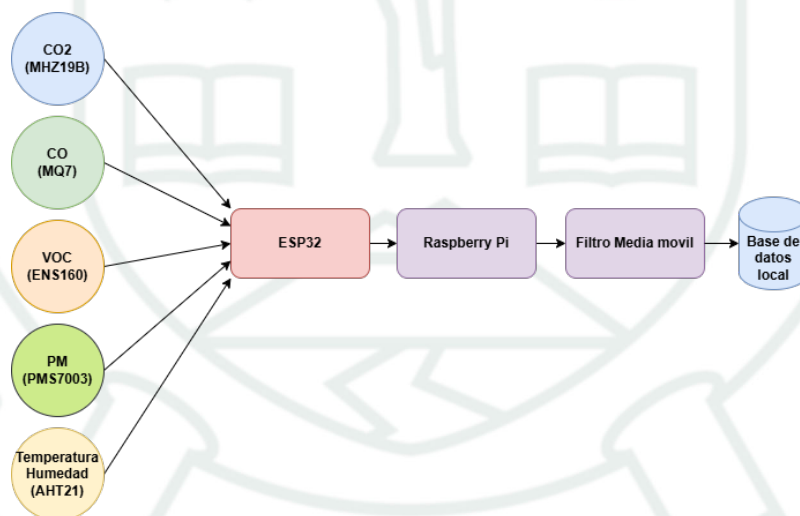
Parámetro	Valor
Rango de medición de CO ₂	0-2000 ppm
Protocolo de comunicación	UART,PWM y analógico
Voltaje de operación	5 V
Consumo de corriente durante medición	<60 mA

Nota. Tomado de la hoja de datos del sensor (Zhengzhou Winsen Electronics Technology Co., Ltd., 2016).

4.2 Adquisición, filtrado y almacenamiento de datos

Figura 4.1

Adquisición de datos



Nota. Elaboración propia.

Ya que se tienen cinco elementos a medir, tal como se observa en la figura 4.1, lo primero que se ha de realizar es el filtrado de las señales recibidas a fin de conseguir estabilizar las mediciones y suavizar los gráficos que se verán en el dashboard. El siguiente algoritmo

describe el funcionamiento del filtro de media móvil que se usará para el suavizado.

Algoritmo 1 Filtro de Media Móvil

Leer datos de cada sensor

for cada dato de entrada **do**

Definir tamaño de ventana (n) y longitud de vector genérico (L)

for cada dato In **do**

$$Out_k = Out_{k-1} + \frac{In(k)}{n} - \frac{In(k-n)}{n}$$

end for

for $i \leftarrow 0$ to $L - 2$ **do**

$InV[L - i - 1] = InV[L - i - 2]$

end for

end for

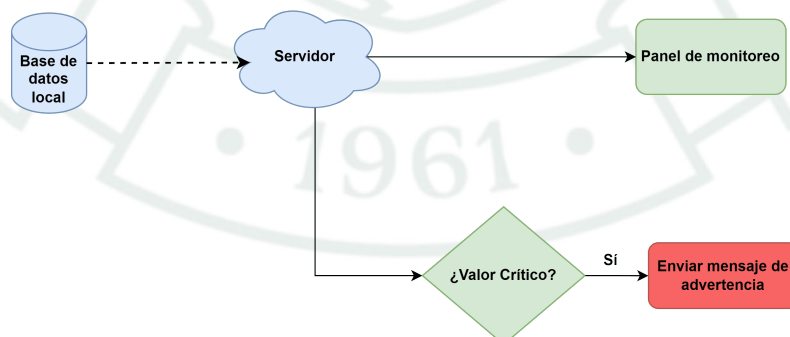
Guardar dato filtrado en base de datos

4.3 Envío de datos a la nube

Aquí se trata lo concerniente a la aplicación de IoT, se implementará un sistema de nivel 3, conforme se detalló en la sección 2.2.1. Siguiendo el esquema presentado en la Figura 2.4 y con todos los elementos de la parte local definidos, procederemos con la configuración de los componentes en la nube. Se instalará la plataforma ThingsBoard en un servidor privado, donde se diseñará el panel de monitoreo y se implementará el sistema de alarma para alertar al usuario final de valores extremos. El diagrama adjunto traza el planteamiento para la etapa en la nube.

Figura 4.2

Procesamiento de datos en la nube



Nota. Elaboración propia.

Con respecto al diseño del panel de monitoreo, contará con medidores gauge en los que se podrá ver valores en tiempo real de cada sensor, también se dispondrá de gráficos históricos para saber los niveles alcanzados anteriormente.

4.4 Cálculo del índice de calidad de aire interior

La EPA ha suministrado una guía integral sobre cómo medir los niveles de calidad del aire interior (IAQ) para varios contaminantes (U.S. Environmental Protection Agency, 2024), de acuerdo con diferentes grados de concentración clasificados según la tabla adjunta.

Tabla 4.6
Clasificación de calidad de aire EPA

These Breakpoints...							Equal this AQI	and this Category
O ₃ (ppm) 8-hour	O ₃ (ppm) 1-hour	PM _{2.5} (ug/m ³) 24-hour	PM ₁₀ (ug/m ³) 24-hour	CO (ppm) 8-hour	SO ₂ (ppb) 1-hour	NO ₂ (ppb) 1-hour	AQI	
0.0	-	0.0–12.0	0–54	0.0–4.4	0–35	0–53	0–50	Good
0.055–0.070	-	12.1–35.4	55–154	4.5–9.4	36–75	54–100	51–100	Moderate
0.071–0.085	0.125–0.164	35.5–55.4	155–254	9.5–12.4	76–185	101–360	101–150	Unhealthy for Sensitive Groups
0.086–0.105	0.165–0.204	55.5–150.4	255–354	12.5–15.4	186–304	361–649	151–200	Unhealthy
0.106–0.200	0.205–0.404	150.5–250.4	355–424	15.5–30.4	305–604	650–1249	201–300	Very unhealthy
-	0.405–0.504	250.5–350.4	425–504	30.5–40.4	605–804	1250–1649	301–400	Hazardous
-	0.505–0.604	350.5–500.4	505–604	40.5–50.4	805–1004	1650–2049	401–500	Hazardous

Nota. Tomado de U.S. Environmental Protection Agency (2024).

El índice de calidad de aire individual (Individual indoor air quality index,IIAQI)para

cada elemento de la tabla adjunto se obtiene con la ecuación

$$I_p = \frac{I_{Hi} - I_{Lo}}{BP_{Hi} - BP_{Lo}} (C_p - BP_{Lo}) + I_{Lo} \quad (4.1)$$

Donde:

I_p = Índice del contaminante p

C_p = Concentración truncada del contaminante p

BP_{Hi} = Punto de corte de concentración que es mayor o igual a C_p

BP_{Lo} = Punto de corte de concentración que es menor o igual a C_p

I_{Hi} = Valor de IAQ correspondiente a BP_{Hi}

I_{Lo} = Valor de IAQ correspondiente a BP_{Lo}

La tabla 4.6 fue diseñada para gases comunes en el exterior, por lo que no se han considerado VOC o CO₂, que son comunes en ambientes cerrados. Para integrar estos contaminantes existen regulaciones impuestas por otros países, como China, Corea del Sur o Malasia, que fueron utilizadas en las investigaciones de Wang et al. (2021), Mad Saad et al. (2017) o Kang y Hwang (2016). En el Perú se dispone de la Resolución Ministerial N.º 181-2016 del Ministerio del Ambiente, que establece el Índice de Calidad de Aire (INCA) (Ministerio del Ambiente, 2016) tomando como base las propuestas de EPA y WHO.

Por lo tanto, se requiere realizar modificaciones en la tabla 4.6 para considerar, además de VOC, CO₂ o NO₂, el confort térmico y posibles cambios rápidos en la concentración. Para abordar esto, se ha adoptado el modelo de Saad et al, con la graduación de la calidad del aire interior (IAQ) comenzando desde 0 como nivel ideal de limpieza del aire. La tabla de datos quedaría así:

Tabla 4.7*Clasificación de calidad de aire interior*

CO ₂ (ppm)	CO (ppm)	NO ₂ (ppm)	VOC (ppm)	PM10 (mg/m ³)	Tempe- ratura (°C)	Hu- medad (%)	IAQ	Clasifi- cación
340– 600	0.0– 1.7	0.0– 0.021	0.0– 0.087	0.0– 0.02	20–26	40–70	0– 25	Bueno
601– 1000	1.8– 8.7	0.022– 0.08	0.088– 0.261	0.021– 0.150	26.1– 29	70.1– 80	26– 50	Moderado
1001– 1500	8.8– 10	0.09– 0.17	0.262– 0.43	0.262– 0.43	29.1– 39	80.1– 90	51– 75	Insalubre
1501– 5000	10–50	0.18–5	0.44– 3.00	0.44– 3.00	39.1– 45	90.1– 100	76– 100	Peligroso

Nota. Tomado de Mad Saad et al. (2017).

4.5 Entrenamiento y evaluación de algoritmos de IA

Los algoritmos de inteligencia artificial requieren de una base de datos que contenga información de diferentes niveles de calidad de aire para poder entrenarse correctamente. Para obtener valores precisos de concentración para cada gas, se realizará una simulación de eventos en los que la cantidad de compuestos varíe a lo largo del tiempo.

El procedimiento para la generación del conjunto de datos consiste en establecer mediciones de cada gas durante 24 horas, con intervalos de 1 segundo, utilizando la Tabla 4.7 y la Fórmula 4.1 de la sección anterior. A cierta cantidad de segundos transcurridos ocurrirá un incremento gradual de los valores para cambiar la clasificación del IAQ. Dado que la magnitud del IAQ tendrá un valor diferente para cada gas en cada instante, se tomará el valor más alto como el indicador representativo de la calidad del aire general del recinto. Todo lo mencionado está detallado en el algoritmo adjunto.

Algoritmo 2 Algoritmo para generación de valores de contaminantes e IAQ

Crear tabla index con los valores de IAQ para cada gas.

Definir 86400 valores aleatorios entre 0 y 1.

Dividir los valores aleatorios en varios segmentos.

Seleccionar un gas de la tabla index.

for cada segmento **do**

 Definir la clasificación de IAQ del segmento según index.

 Multiplicar cada valor del segmento por el valor correspondiente a la clasificación.

end for

Concatenar los segmentos.

for cada valor de cada segmento **do**

 Calcular IAQ

 Concatenar cada resultado en un array

end for

Aplicar filtros Savitzky-Golay y media móvil al array.

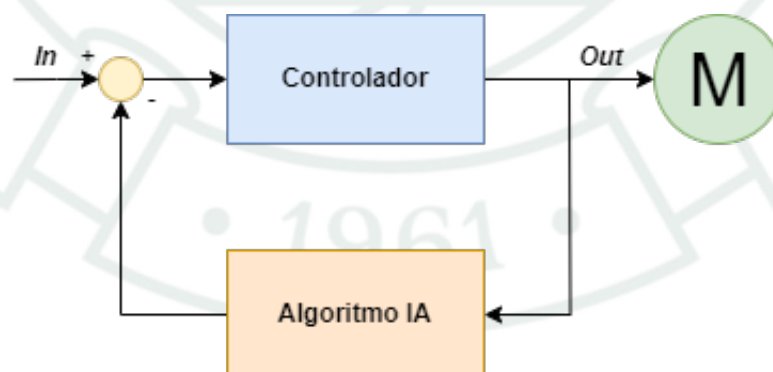
Guardar los valores generados en la base de datos

Con el conjunto de datos generado, el siguiente paso será aplicar los algoritmos de predicción descritos en la sección 2.8. Cada uno de estos algoritmos tiene hiperparámetros que deben calibrarse para obtener la configuración óptima, lo que permitirá realizar una comparación de rendimiento precisa entre ellos.

Para la regulación de la calidad del aire en el recinto, se dispondrá de una salida PWM capaz de controlar, por ejemplo, un ventilador con una velocidad dependiente de los valores de IAQ procedentes de la predicción del algoritmo de IA que se implementará en el sistema.

Figura 4.3

Control automático de la ventilación



Nota. Elaboración propia.

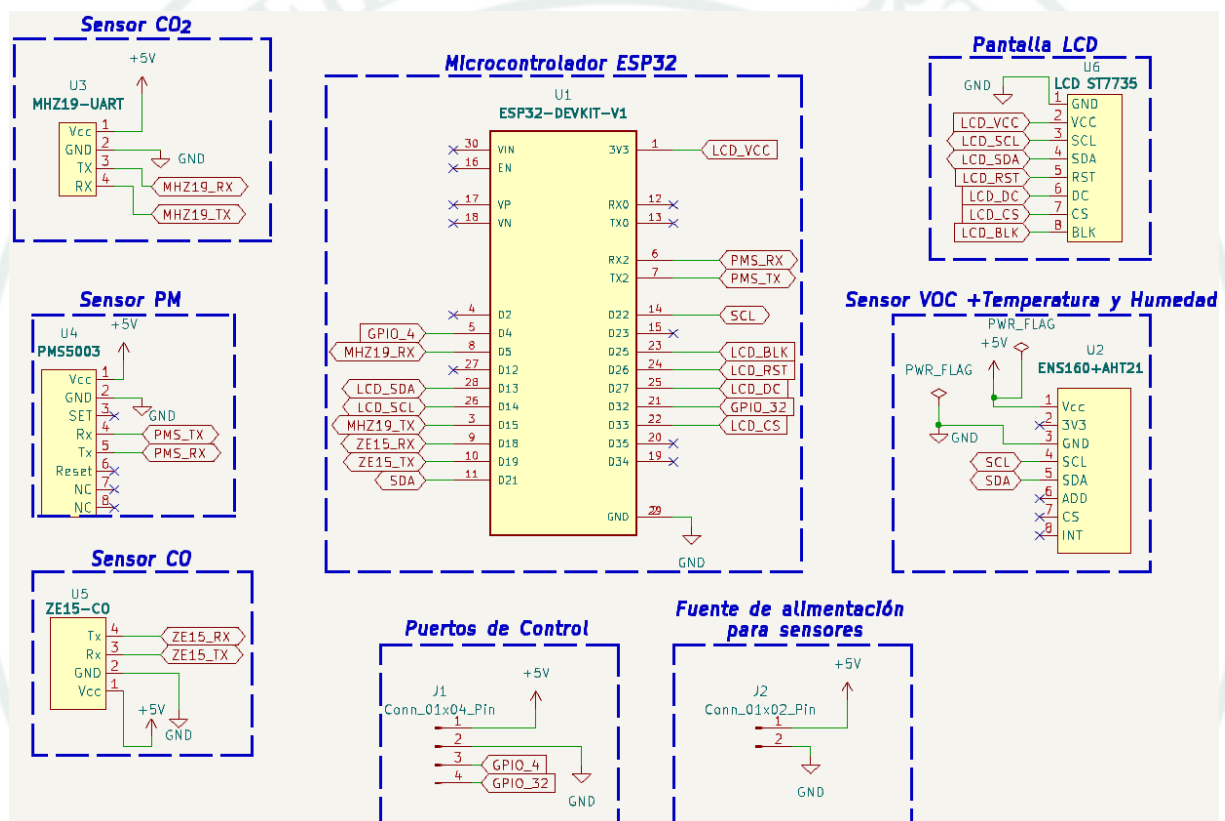
4.6 Implementación del hardware

4.6.1 Diseño y ensamble del circuito

Las figuras 4.4 y 4.5 muestran el esquema del circuito y la implementación, respectivamente:

Figura 4.4

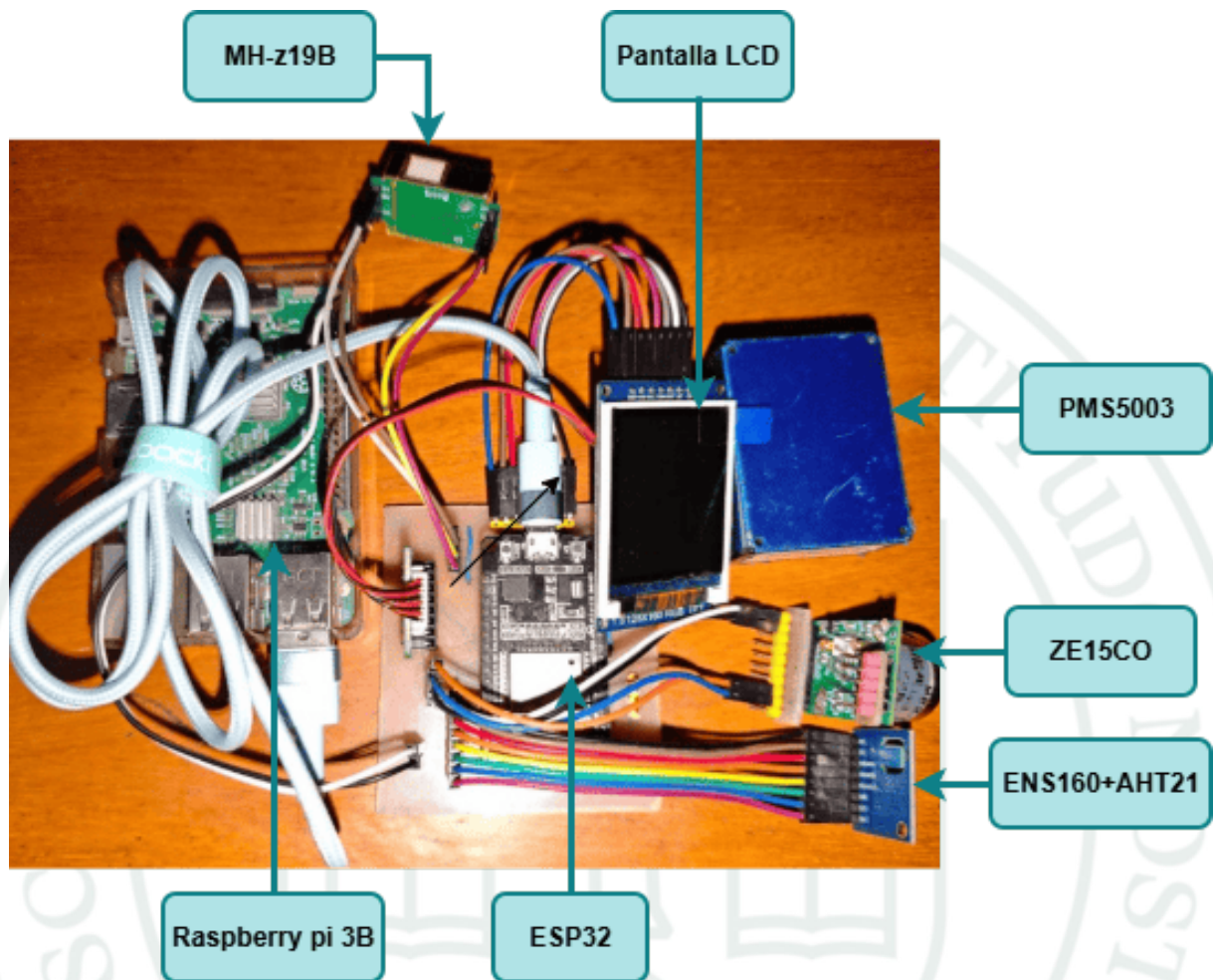
Esquema del circuito electrónico



Nota. Elaboración propia.

Figura 4.5

Implementación del circuito



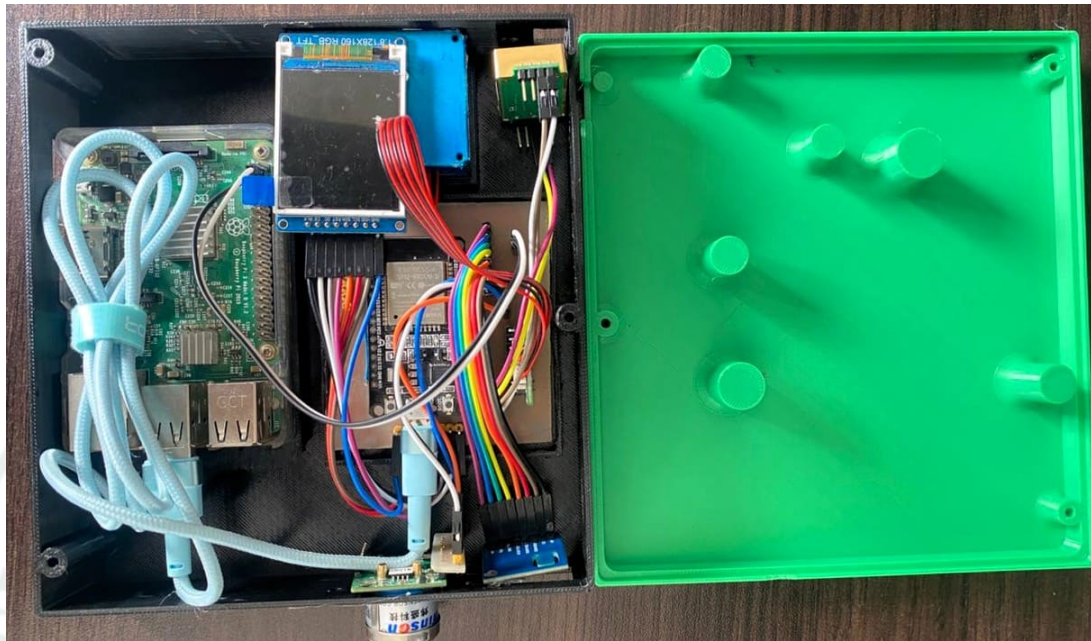
Nota. Elaboración propia.

La información generada por los sensores es leída por el microcontrolador ESP32 y este la envía a través del puerto USB a la computadora Raspberry Pi. Ya que se trata de un sistema que mide la calidad del aire en cada instante, todos los sensores se mantienen encendidos constantemente. De acuerdo con la medición de corriente, el circuito consume **400 mA**.

El circuito implementado se colocó en un gabinete hecho mediante impresión 3D.

Figura 4.6

Ensamblaje final del prototipo dentro de un gabinete 3D



Nota. Elaboración propia.

4.6.2 Funcionamiento del sistema

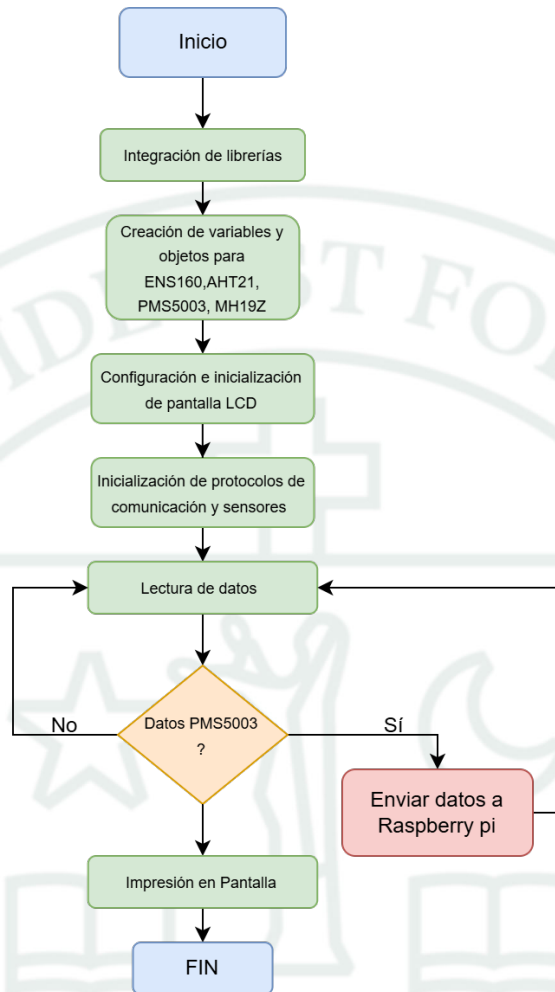
Los sensores usan protocolo de comunicación I²C y UART, este último solo soporta un dispositivo, por lo que se recurre a puertos de comunicación virtuales debido a que ESP32 solo posee un UART físico extra para uso general. Todos los puertos UART operan a una velocidad de 9600 baudios.

En cuanto a los dispositivos I²C, estos se interconectan según el circuito de la Figura 4.4 y son administrados a través de la librería 'Wire.h' del software Arduino.

El algoritmo diseñado para el proceso de medición funciona de acuerdo con el siguiente diagrama de flujo:

Figura 4.7

Funcionamiento del sistema: etapa de sensado



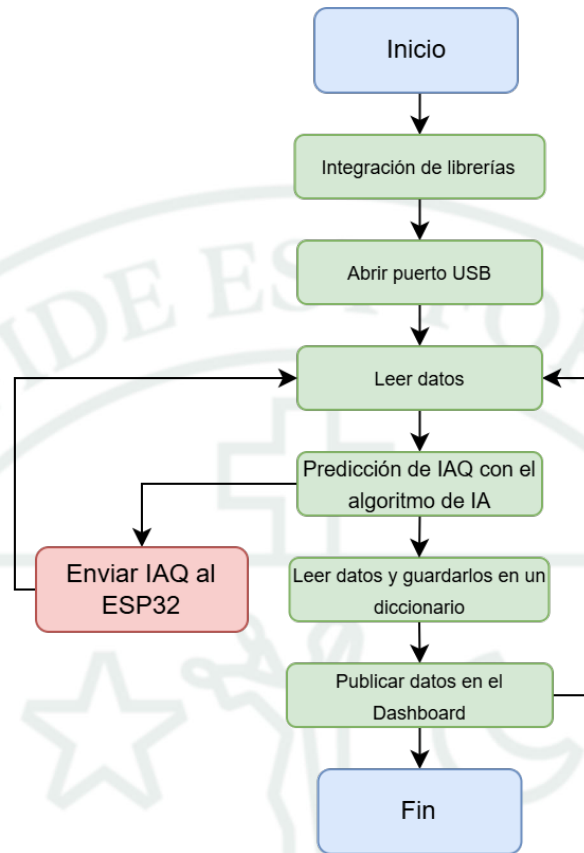
Nota. Elaboración propia.

El sensor PMS5003, según su hoja de datos (Plantower, 2025), requiere un tiempo de medición menor a 10 s para obtener los tres tamaños de partículas. Por lo tanto, cuando se adquieran estos datos, se enviarán todas las mediciones al Raspberry Pi para su procesamiento.

Los datos son recibidos en Raspberry Pi, se publican en el dashboard de la plataforma web IoT ThingsBoard y se muestran en la pantalla integrada, que además posee una alerta visual basada en los colores representativos de cada nivel de IAQ (Cuadro 4.7). Las mediciones de los gases se utilizan para predecir la calidad del aire instantánea con el algoritmo de IA seleccionado. Para ilustrar la descripción anterior se tiene el diagrama de flujo de la Figura 4.8.

Figura 4.8

Funcionamiento del sistema: etapa de recepción de datos en Raspberry Pi



Nota. Elaboración propia.

El valor de IAQ estimado por el sistema predictivo se transmite al microcontrolador, el cual ejecuta la lógica de control del ventilador tal como se plantea en el diagrama de bloques de la figura 4.3. La regulación de la velocidad de giro se efectúa a través de una señal PWM(Pulse Width Modulation) logrando un eficiente suministro de potencia al actuador. Por tanto, el sistema puede responder de manera dinámica a los cambios en la calidad del aire interior, optimizando el consumo energético y manteniendo un ambiente seguro para los ocupantes.

El algoritmo de control de ventilación utilizado es del tipo proporcional con umbrales discretos, logrando una respuesta escalonada dependiendo del rango en que se encuentre el índice de calidad de aire (IAQ):

- **0–25 (Bueno):** El ventilador permanece apagado porque las condiciones ambientales son adecuadas.
- **26–50 (Moderado):** El ventilador se activa con una señal PWM del 25 % de ciclo de trabajo.
- **51–75 (Peligroso):** La señal PWM sube al 50 % si el índice IAQ está entre 51 y 62, aumentando al 75 % si se supera ese intervalo, con el fin de evitar que el IAQ caiga a niveles insalubres.
- **76–100 (Insalubre):** La señal PWM se establece en 95 %, haciendo que el sistema funcione prácticamente a máxima potencia.

En cuanto a la generación de la señal PWM, se ha establecido una configuración de **10 bits de resolución y frecuencia de 1 kHz** para asegurar una respuesta silenciosa y eficiente del ventilador. El actuador puede ser un ventilador tipo *cooler* de PC alimentado por una fuente de 6V, controlado mediante un **driver L298N**, que modula la tensión aplicada al motor según el ciclo de trabajo definido por el nivel de IAQ.

Para evitar cambios frecuentes entre los cuatro niveles de velocidad, se ha aplicado histéresis en cada transición. El siguiente cuadro muestra la parametrización de los rangos:

Tabla 4.8

Umbrales estándar y con histéresis para transiciones de velocidad del ventilador

Transición	Umbral estándar	Umbral con histéresis
Bueno → Moderado	25 → 26	25 → 26
Moderado → Bueno	26 → 25	26 → 23
Moderado → Peligroso	50 → 51	50 → 51
Peligroso → Moderado	51 → 50	51 → 48
Peligroso → Insalubre	75 → 76	75 → 76
Insalubre → Peligroso	76 → 75	76 → 73

Nota. Elaboración propia.

Medidas de robustez del sistema. Para garantizar el funcionamiento confiable del sistema, se han implementado las siguientes medidas:

1. **Validación del dato recibido:** Todo dato recibido por el ESP32 debe estar dentro del rango esperado ($0 \leq \text{IAQ} \leq 100$), evitando errores de transmisión o fallos en los valores de predicción.
2. **Comportamiento en caso de fallos (*fail-safe*):** Ante fallos, el ventilador se establece en una velocidad media (PWM al 50 %).
3. **Pruebas de validación funcional:** Se evaluó el comportamiento en las transiciones de nivel y la reacción del sistema al desconectar la comunicación entre la Raspberry Pi y el ESP32.

4.7 Algoritmos de IA aplicados

4.7.1 Selección de algoritmos

Se tomaron en cuenta tres algoritmos que permiten predecir valores numéricos, como la calidad del aire, a partir de sus características: Redes neuronales, regresión lineal LASSO y árboles de decisión.

La exclusión de otros algoritmos como Kmeans o SVM se debe a que están diseñados para problemas de clasificación o clustering.

Los tres algoritmos poseen ventajas y desventajas inherentes a su estructura:

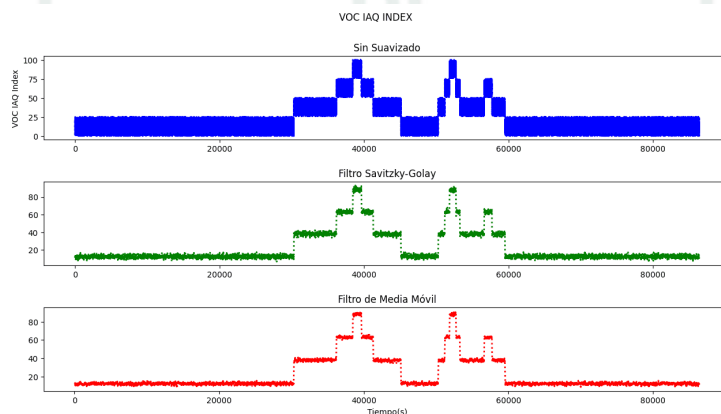
- **Redes neuronales:** tienen una arquitectura flexible, alta capacidad predictiva tanto para valores lineales como no lineales, pero requiere un alto coste computacional y una base de datos relativamente grande para lograr una eficiencia aceptable.
- **Regresión lineal:** Facilita la interpretación de la importancia de cada variable que interviene en el proceso de predicción. El coeficiente L1 de la regresión LASSO ayuda a prevenir sobre ajuste y tiene un coste computación muy bajo en comparación con las ANN o los árboles de decisión. La desventaja más importante de este modelo es que asume relaciones lineales entre las características y etiquetas, reduciendo su eficiencia en casos con tendencias no lineales, además de ser muy sensible a valores atípicos(outliers).

- **Árboles de decisión:** Su estructura es simple, no requiere normalización y es robusto ante outliers. La desventaja de este algoritmo es que tiende al sobre ajuste.

Preparación de los datos. Se han generado datos para 4 entornos diferentes: un gimnasio, una cocina industrial, un taller mecánico y un laboratorio de mantenimiento industrial. Siendo ambientes muy diferentes entre ellos, el comportamiento de los gases también lo será. La variación de la calidad del aire se simuló a lo largo de un día, en una escala de segundos.

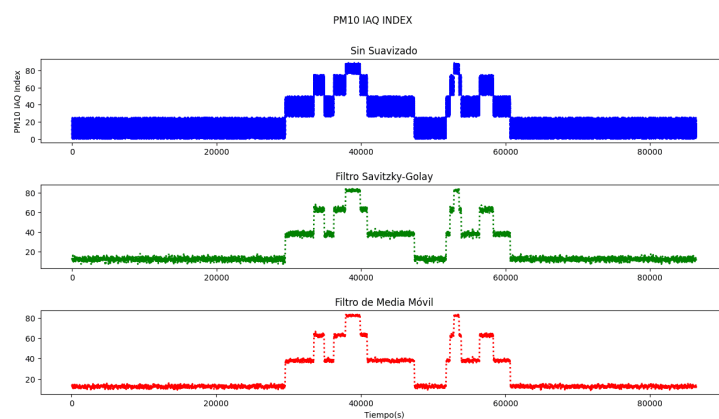
- **Laboratorio de mantenimiento industrial:** Para este entorno se han considerado como gases predominantes el CO, el NO₂, los VOC, el PM10 y, en menor medida, el CO₂. Los dispositivos comunes en un laboratorio de mantenimiento industrial que generan estos gases son motores de combustión y eléctricos, máquinas de soldadura, lijadoras, esmeriles, entre otros. También existen otras fuentes, como lubricantes, solventes o desengrasantes.

Figura 4.9
Índice de IAQ para VOC



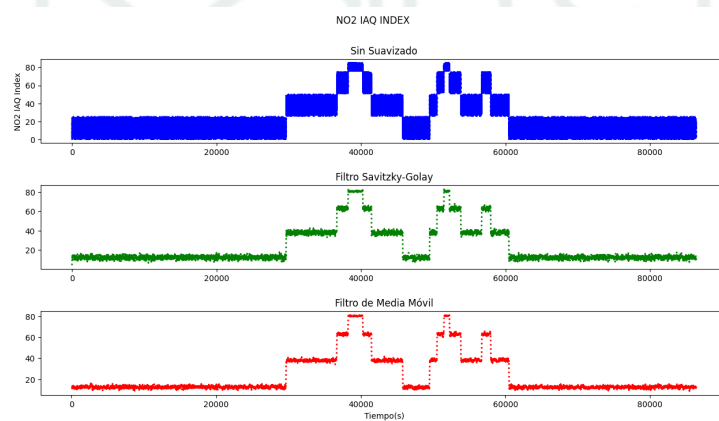
Nota. Elaboración propia basada en el algoritmo de generación de datos sintéticos.

Figura 4.10
Índice de IAQ para PM10



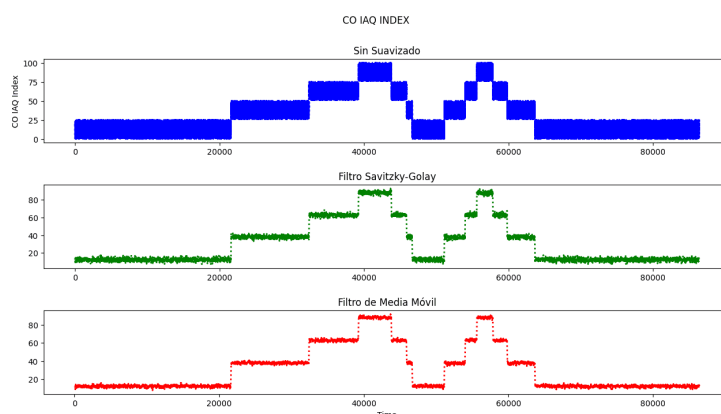
Nota. Elaboración propia basada en el algoritmo de generación de datos sintéticos.

Figura 4.11
Índice de IAQ para NO₂



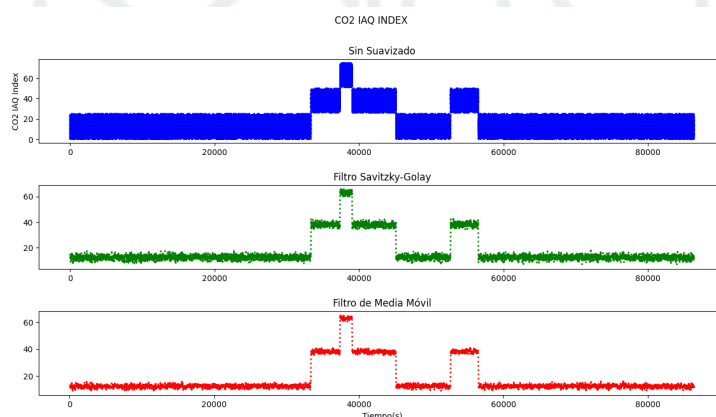
Nota. Elaboración propia basada en el algoritmo de generación de datos sintéticos.

Figura 4.12
Índice de IAQ para CO



Nota. Elaboración propia basada en el algoritmo de generación de datos sintéticos.

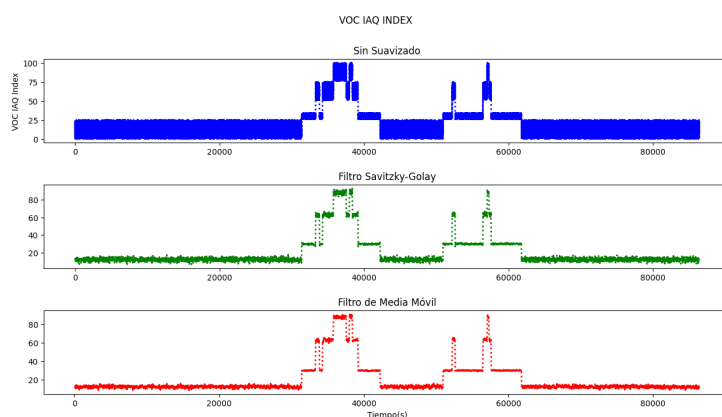
Figura 4.13
Índice de IAQ para CO₂



Nota. Elaboración propia basada en el algoritmo de generación de datos sintéticos.

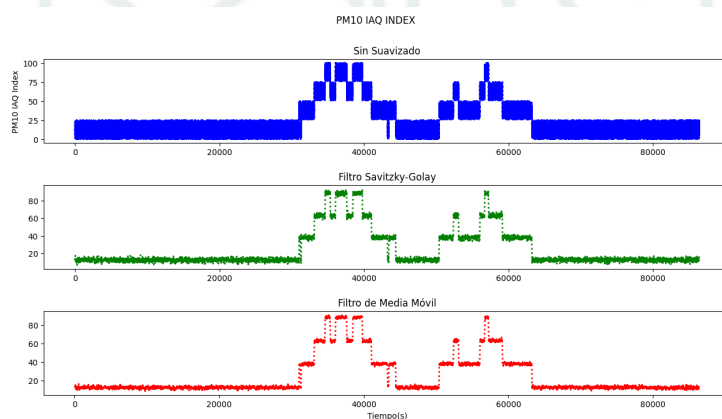
- **Taller mecánico:** En este ambiente se considera un flujo elevado de vehículos que están siendo examinados, piezas mecánicas siendo modificadas o en mantenimiento, soldadura, limpieza de componentes, uso de diferentes compuestos químicos como aditivos, aceites, grasas y combustibles. Se predomina NO₂, VOC, PM10, CO y en menor medida CO₂.

Figura 4.14
Índice de IAQ para VOC



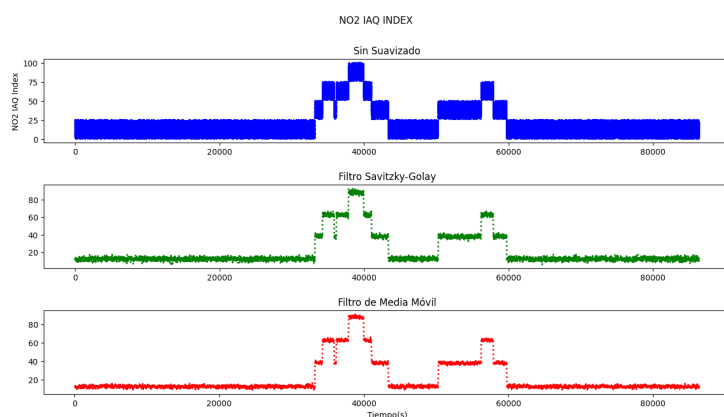
Nota. Elaboración propia basada en el algoritmo de generación de datos sintéticos.

Figura 4.15
Índice de IAQ para PM10



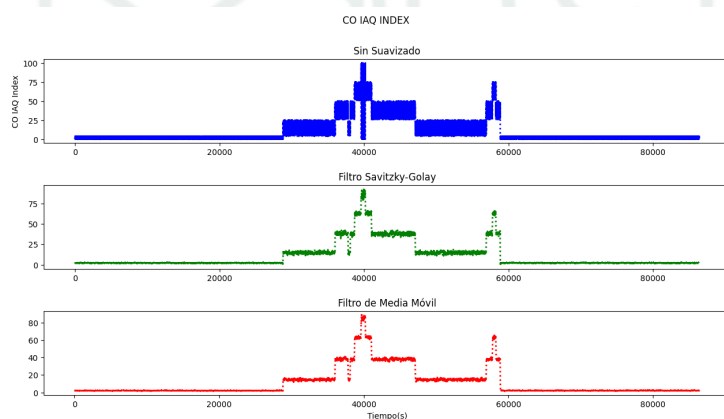
Nota. Elaboración propia basada en el algoritmo de generación de datos sintéticos.

Figura 4.16
Índice de IAQ para NO_2



Nota. Elaboración propia basada en el algoritmo de generación de datos sintéticos.

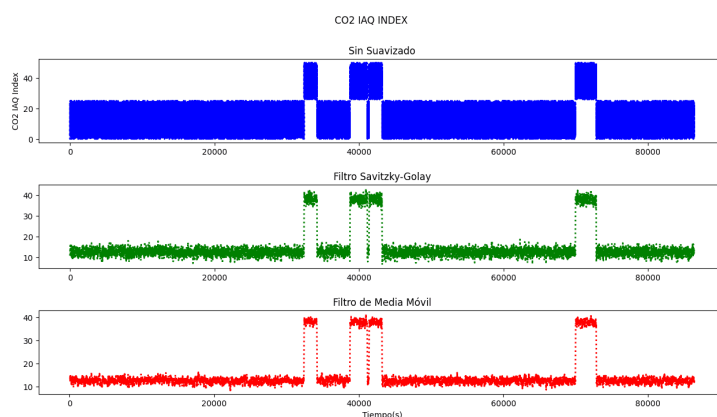
Figura 4.17
Índice de IAQ para CO



Nota. Elaboración propia basada en el algoritmo de generación de datos sintéticos.

Figura 4.18

Índice de IAQ para CO₂

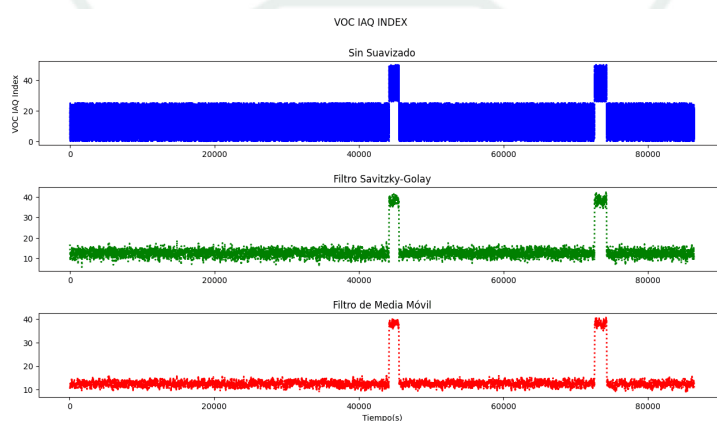


Nota. Elaboración propia basada en el algoritmo de generación de datos sintéticos.

- **Gimnasio:** En un lugar así predomina el CO₂ por la aglomeración de personas, también se ha tomado en cuenta el impacto de NO₂ y VOC durante periodos de limpieza, donde se aplican detergentes, desinfectantes y ambientadores. El gas CO no se ha tomado como relevante, por lo que su influencia es casi nula; de igual manera, sucede con las partículas de materia.

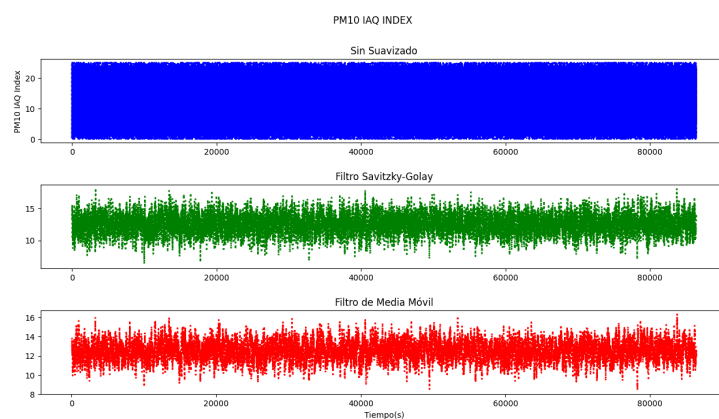
Figura 4.19

Índice de IAQ para VOC



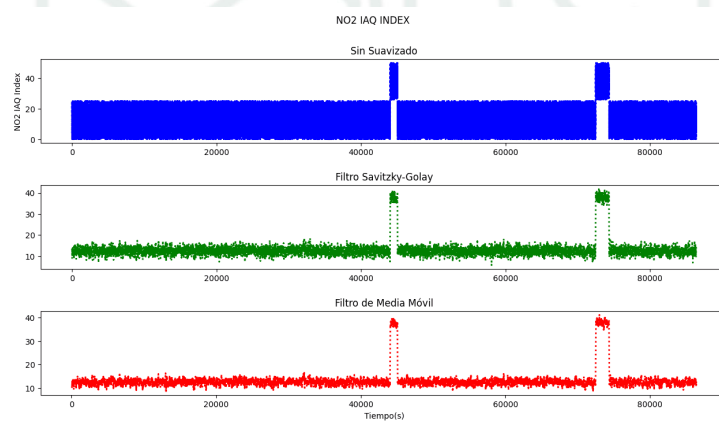
Nota. Elaboración propia basada en el algoritmo de generación de datos sintéticos.

Figura 4.20
Índice de IAQ para PM10



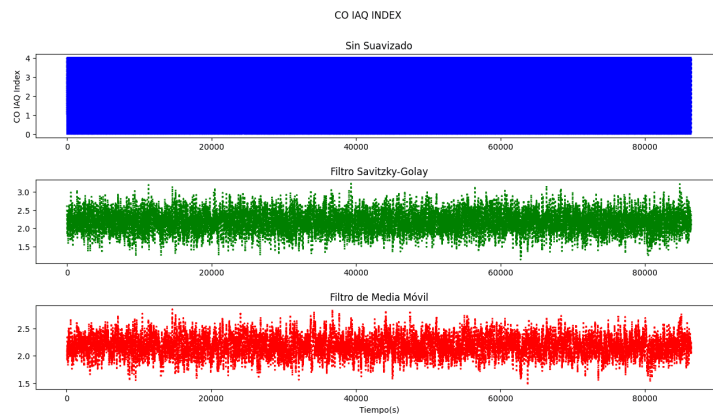
Nota. Elaboración propia basada en el algoritmo de generación de datos sintéticos.

Figura 4.21
Índice de IAQ para NO₂



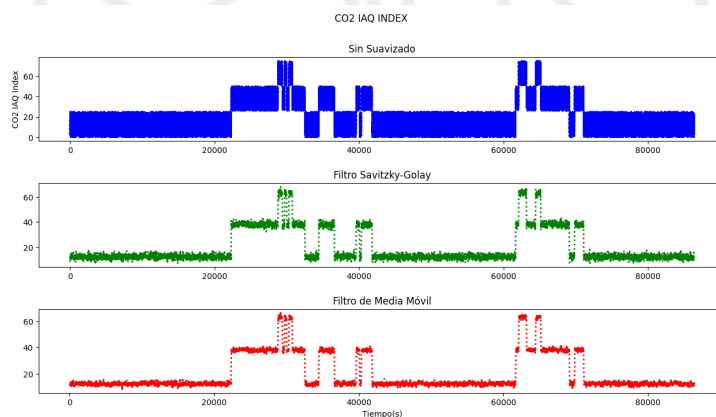
Nota. Elaboración propia basada en el algoritmo de generación de datos sintéticos.

Figura 4.22
Índice de IAQ para CO



Nota. Elaboración propia basada en el algoritmo de generación de datos sintéticos.

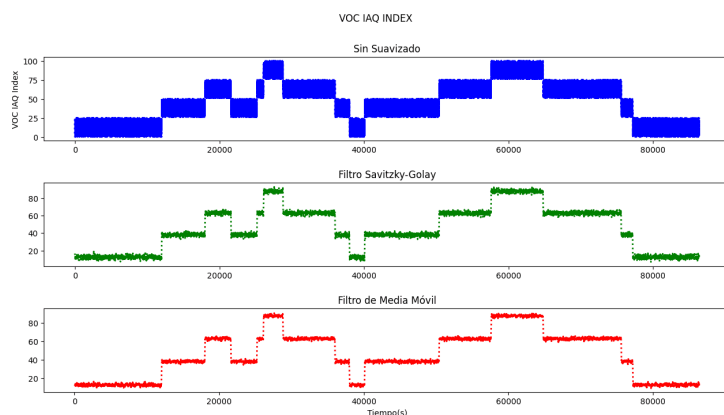
Figura 4.23
Índice de IAQ para CO₂



Nota. Elaboración propia basada en el algoritmo de generación de datos sintéticos.

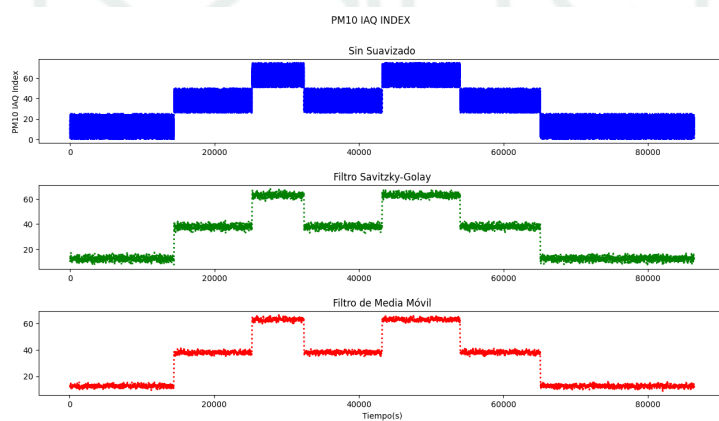
- **Cocina Industrial:** Siendo un espacio relativamente pequeño y cerrado que cuenta con cocinas, parrillas, hornos y otros equipos de combustión, además de los ciclos de limpieza en los que se utilizan quitamanchas o detergentes, los gases CO, NO₂ y VOC tienen un impacto considerable. La presencia de trabajadores que generan CO₂ también tiene influencia en la calidad de aire general en la cocina industrial.

Figura 4.24
Índice de IAQ para VOC



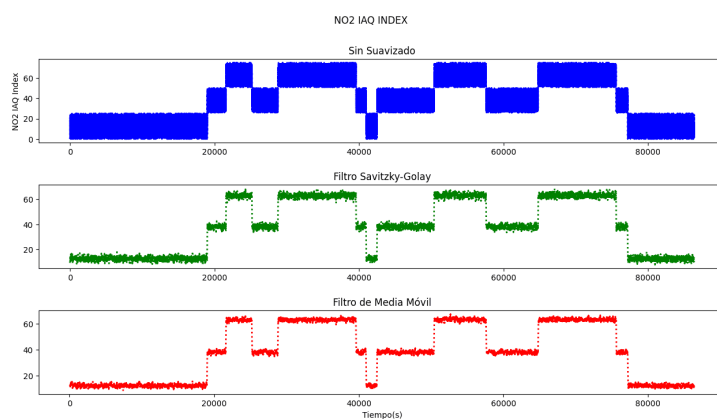
Nota. Elaboración propia basada en el algoritmo de generación de datos sintéticos.

Figura 4.25
Índice de IAQ para PM10



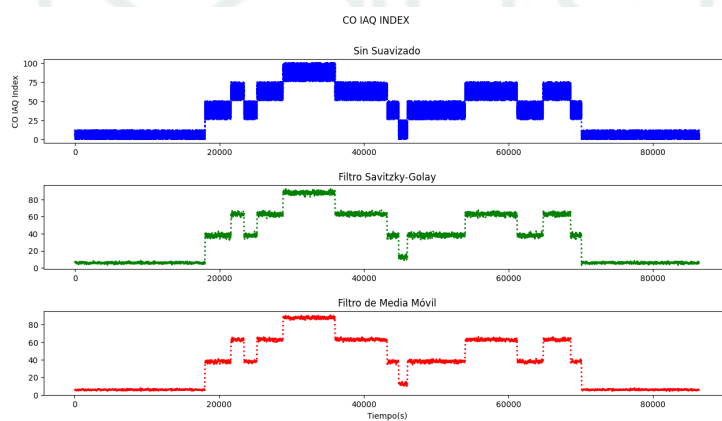
Nota. Elaboración propia basada en el algoritmo de generación de datos sintéticos.

Figura 4.26
Índice de IAQ para NO_2



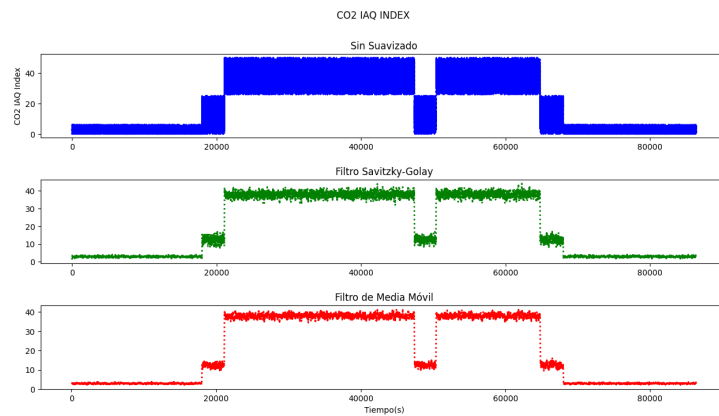
Nota. Elaboración propia basada en el algoritmo de generación de datos sintéticos.

Figura 4.27
Índice de IAQ para CO



Nota. Elaboración propia basada en el algoritmo de generación de datos sintéticos.

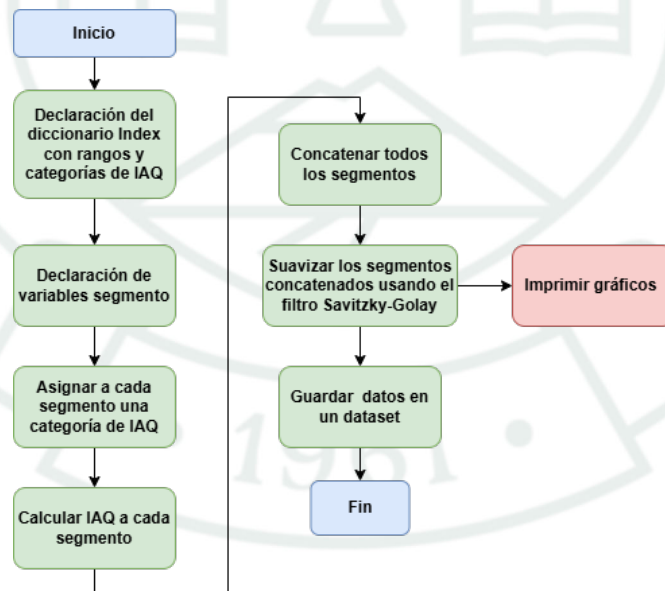
Figura 4.28
Índice de IAQ para CO₂



Nota. Elaboración propia basada en el algoritmo de generación de datos sintéticos.

Para la generación de datos se diseñó un programa en el que se considera la tabla 4.7 de la sección 4.4 para datos instantáneos. El diagrama de flujo en la figura 4.29 muestra cómo funciona el algoritmo.

Figura 4.29
Generación de datos sintéticos para entrenamiento



Nota. Elaboración propia.

Las variables denominadas segmento representan intervalos temporales, expresados en segundos, donde se almacenan valores de IAQ correspondientes a una categoría. La variación de los datos en los segmentos se han diseñado acorde al análisis hecho en el capítulo 3 para cada ambiente.

Debido a la generación artificial de datos, se aplicaron técnicas de suavizado para mejorar la representatividad de las curvas temporales. Los filtros Savitzky-Golay y de media móvil fueron aplicados para suavizar las curvas y se eligió el primero por su capacidad de preservar picos y valles con precisión.

Los datos se generan para cada gas y su respectivo valor de IAQ se almacena en un dataset. Se conserva únicamente el valor máximo, descartando los demás.

4.7.2 Entrenamiento y validación

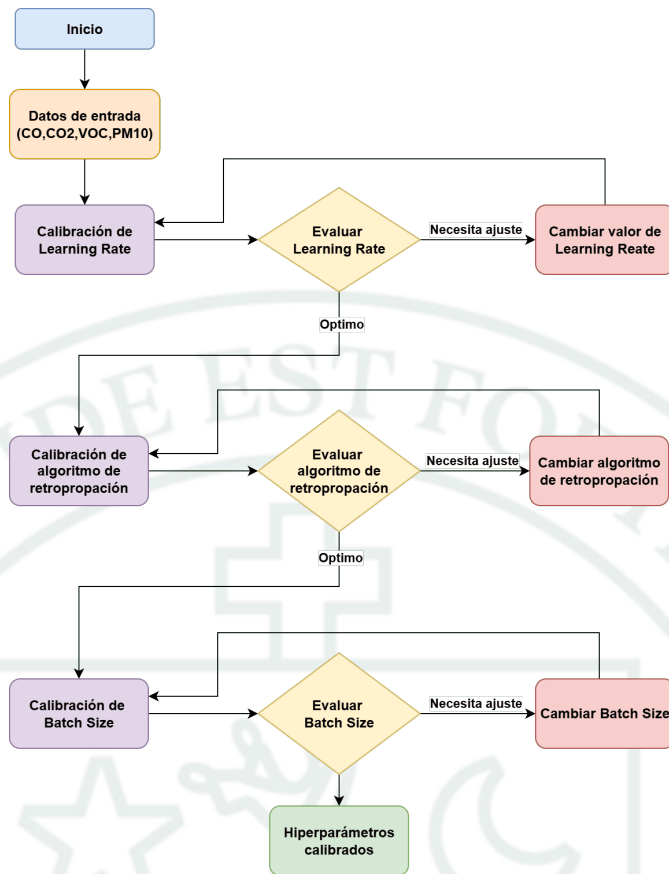
Para el entrenamiento de los modelos se ha realizado el proceso inherente de cada algoritmo, sin embargo, los tres modelos tienen en común la normalización de los datos y la subdivisión del dataset en conjuntos de entrenamiento y validación en una proporción de 80 % y 20 %. Todos los hiperparámetros de los algoritmos han sido calibrados para obtener la mejor predicción posible.

- **Redes neuronales:** En los cuatro ambientes se ha seguido un proceso de calibración iterativo para cada hiperparámetro, es decir, probando diferentes valores y observando el comportamiento de la métrica de evaluación a lo largo de las épocas.

El diagrama de flujo de la figura 4.30 ilustra el proceso de obtención de la tasa de aprendizaje, tamaño de batch y el algoritmo de retropropagación mediante la evaluación de la reducción del error medio absoluto.

Figura 4.30

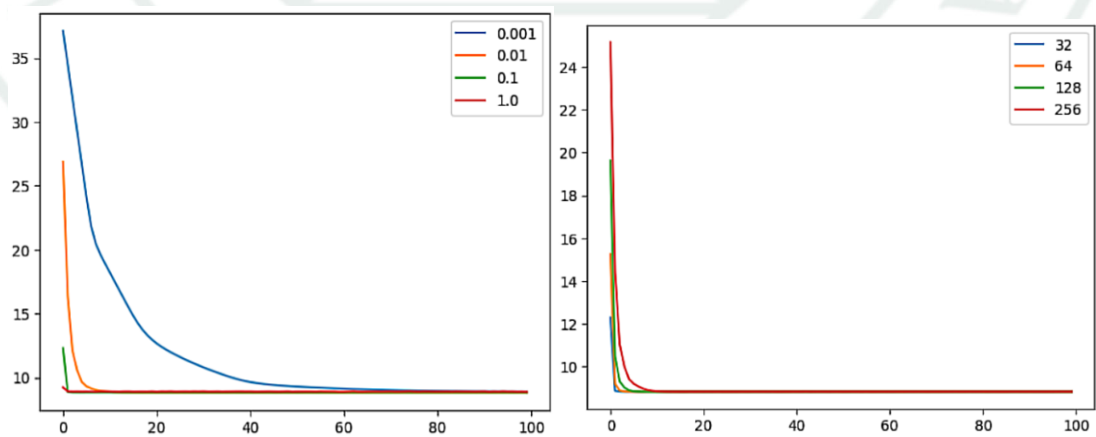
Cálculo de hiperparámetros de la red neuronal



Nota. Elaboración propia.

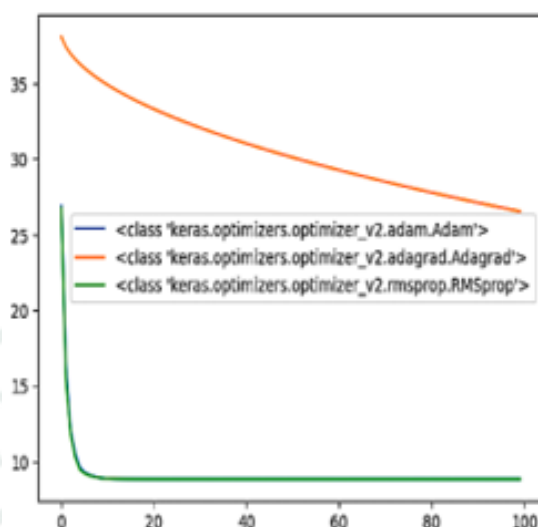
Figura 4.31

Calibración de hiperparámetros



(a) Learning Rate

(b) Tamaño de batch



(c) Algoritmos de retropropagación

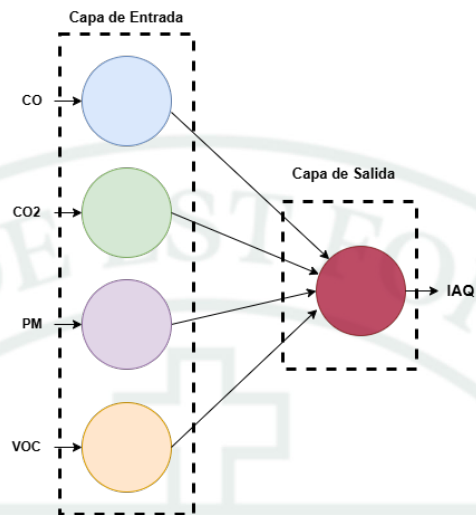
Nota. Elaboración propia basada en el programa de selección de algoritmo de retropropagación.

Los gráficos en 4.31 muestran la reducción del error medio absoluto durante el entrenamiento de un perceptrón con cuatro entradas correspondientes a los valores normalizados de los gases para obtener como salida el valor de IAQ, esta estructura se observa en 4.32. La gráfica 4.31(a) representa los resultados con diferentes tasas de aprendizaje (learning rate), 4.31(b) muestra el impacto del tamaño del batch y 4.31(c) compara diferentes algoritmos de retropropagación (backpropagation).

Con los valores óptimos de learning rate, el tamaño de batch y el algoritmo de backpropagation, se calcula el número de capas y la cantidad de neuronas en cada una. Esto se logra mediante iteración con diferentes configuraciones de neuronas por cada capa, esto se ilustra en el diagrama de flujo de la figura 4.33.

Figura 4.32

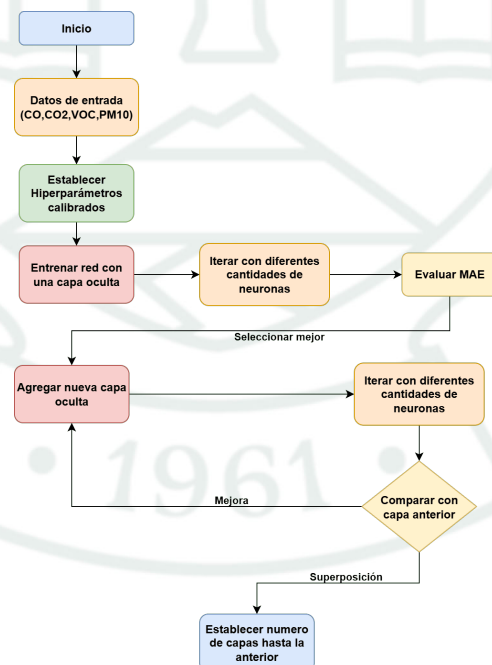
Estructura del perceptrón para obtención de hiperparámetros



Nota. Elaboración propia.

Figura 4.33

Calibración de número de neuronas y capas

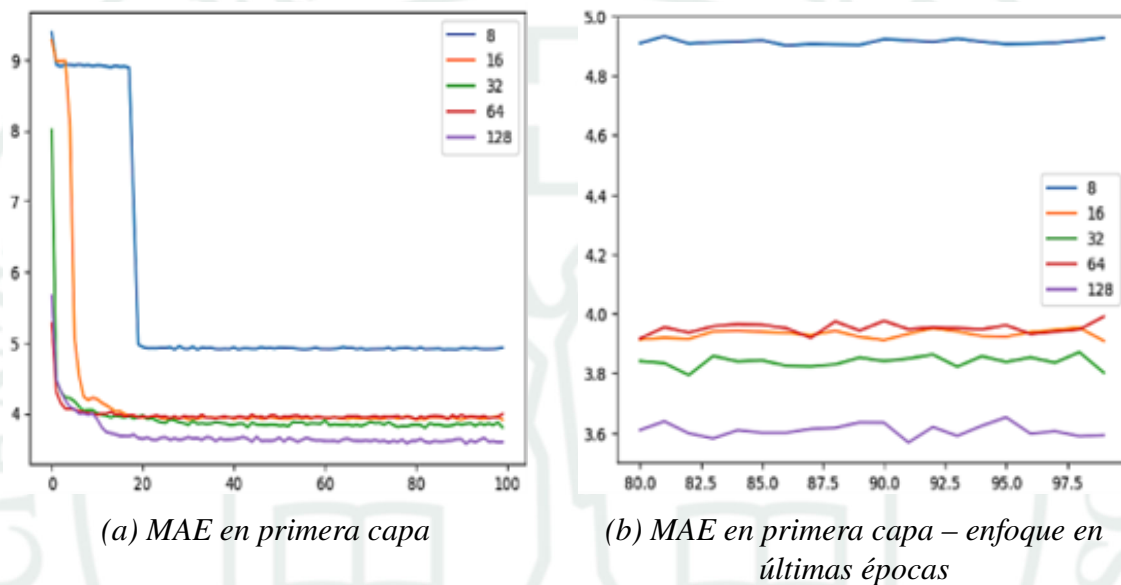


Nota. Elaboración propia.

En la primera capa, se probaron configuraciones de 8 a 128 neuronas. La gráfica en 4.34(a) muestra el valor del MAE a lo largo de 100 épocas para cada una de las configuraciones de neuronas. En 4.34(b) hay una ampliación de la gráfica en la última etapa del entrenamiento donde se aprecia que la configuración que reduce mejor el error es la de 128 neuronas.

Figura 4.34

Reducción del error medio absoluto (MAE) para distintas configuraciones de neuronas en la primera capa

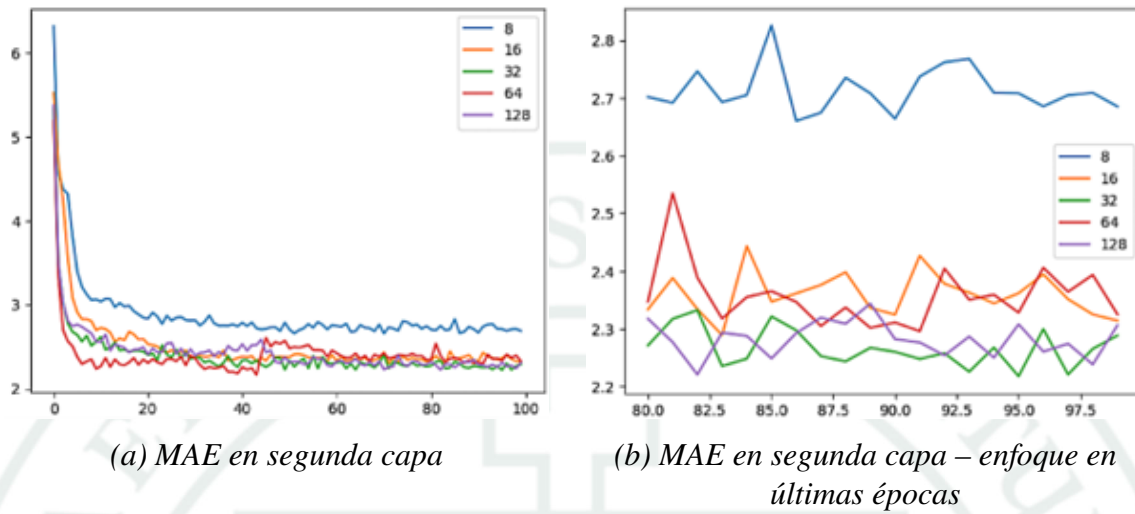


Nota. Elaboración propia basada en el programa de calibración de número de capas.

Se sigue el mismo proceso para a segunda capa y los resultados aparecen en los gráficos a continuación:

Figura 4.35

Reducción del error medio absoluto (MAE) para distintas configuraciones de neuronas en la segunda capa



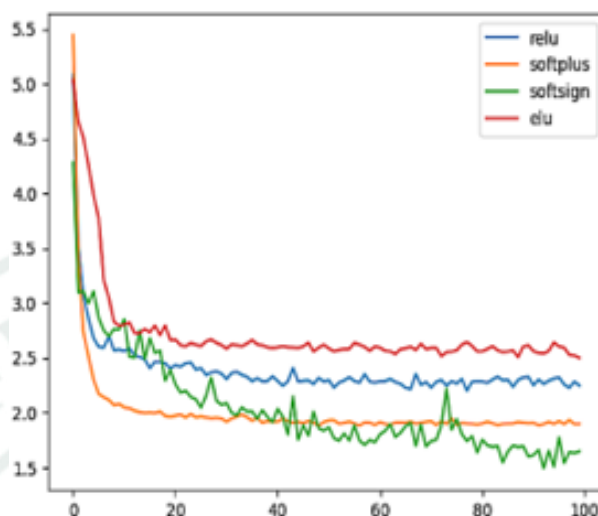
Nota. Elaboración propia basada en el programa de calibración de número de capas.

En 4.35(a) se muestra el comportamiento de la segunda capa en todas las épocas, mientras que 4.35(b) presenta un acercamiento a las últimas etapas del entrenamiento. Aunque se observa una reducción en el error medio absoluto, también existe una superposición entre las distintas configuraciones, lo que indica que no hay una diferencia significativa entre la aplicación de diferentes tamaños de capa, lo que llevó a la conclusión de no usar una segunda capa en el modelo final.

Es importante destacar que el entrenamiento del número de neuronas por capa se hizo con una función de activación relu. Sin embargo, esta podría no ser la más adecuada en el modelo, entonces, con los hiperparámetros y la primera capa ya definidos, se llevó a cabo un nuevo entrenamiento probando distintas funciones de activación.

Figura 4.36

Comparativa de funciones de activación



Nota. Elaboración propia basada en el programa de selección de función de activación.

La figura 4.36 revela que la función de activación que reduce más el error medio absoluto es softsign. En resumen, la estructura optima de la red neuronal es la que aparece en el siguiente cuadro:

Tabla 4.9

Hiperparámetros de la red neuronal. Programas de calibración del número de capas e hiperparámetros. Elaboración propia

Hiperparámetros	Valor
Número de capas ocultas	1
Cantidad de neuronas por capa	128
Learning Rate	0.1
Batch Size	32
Función de Activación	Softsign
Algoritmo de Backpropagation	Adam

Nota. Elaboración propia.

Empleando el modelo definido en el cuadro 4.9 y entrenando la red con los datos de cada ambiente se obtienen los resultados de las métricas de evaluación en la tabla 4.10.

Tabla 4.10*Aplicación del modelo de Red neuronal en los ambientes*

Parámetros/ Ambientes	Laboratorio	Taller	Gimnasio	Cocina Industrial
MAE	1.21	1..7	1.34	2.1
R ²	0.98	0.98	0.97	0.97

Nota. Elaboración propia.

- **Regresión Lasso:** Como se menciona en la sección 2.8.3, este tipo de regresión lineal incluye un hiperparámetro de regularización que debe ajustarse para obtener los coeficientes óptimos en la función de predicción resultante (L1). Su eficiencia puede afinarse más al aplicar un coeficiente de proporcionalidad entre L1 y L2 (regresión Ridge), mediante el uso del parámetro elastic net, el cual permite combinar ambos enfoques de regresión lineal en un solo modelo. Además, se aplicó el parámetro Regparam que indica qué tan importante es aplicar la regularización.

Los valores de elastic net fueron 0.001, 0.005, 0.03, 0.17 y 1.0; mientras que los de Reg-Param fueron 0.001, 0.01, 0.1, 10 y 100. Con estas combinaciones se obtuvo distintas funciones de predicción, evaluadas mediante MAE y R².

El cuadro 4.11 muestra los mejores coeficientes para los gases en cada ambiente además de las métricas de evaluación:

Tabla 4.11*Coeficientes del regresor LASSO*

Parámetros/ Ambientes	Laboratorio	Taller	Gimnasio	Cocina Industrial
Coeficientes (CO ₂ , CO, VOC, PM10)	17.74, 72.40, 34.44, 52.52	13.95, 55.62, 35.10, 88.77	13.95, 55.60, 35.08, 88.81	10.57, 45.85, 45.62, 8.78
MAE	9.97	8.03	8.03	9.0
R ²	0.79	0.80	0.80	0.65

Nota. Elaboración propia basada en el programa de calibración de regresión LASSO.

- **Árboles de decisión:** Para aplicar árboles de decisión se tiene que tomar en consideración 3 hiperparámetros, que son la cantidad de árboles, profundidad y entropía, este último es ignorado porque los aplicaremos como un regresor. Para el número de árboles se evaluaron los valores 5,10 y 20; mientras que para la profundidad se usó 5,10 y 15. La tabla 4.12 muestra los mejores coeficientes para los gases en cada ambiente además de las métricas de evaluación:

Tabla 4.12

Coefficientes de los árboles de decisión

Parámetros/Ambientes	Laboratorio	Taller	Gimnasio	Cocina Industrial
N° de árboles	10	20	20	10
Profundidad	10	10	10	5
MAE	1.22	1.29	1.29	1.6
R ²	0.98	0.98	0.98	0.96

Nota. Elaboración propia basada en el programa de calibración de árboles de decisión.

4.8 Computación en la nube

4.8.1 Arquitectura de la plataforma

De acuerdo con la estructura presentada en la Figura 4.2, luego de que los datos son recibidos por el microcomputador Raspberry Pi, que sirve como gateway y centro de almacenamiento local de información, este envía las mediciones a un servidor en la nube. El servidor cuenta con una CPU Intel® Xeon® compartida de 2 núcleos, 4 GB de RAM y 40 GB de memoria interna, además de una velocidad de comunicación de hasta 10 Gbps. Su elección radica en los requisitos mínimos de la plataforma IoT que se ha escogido. Como sistema operativo, cuenta con Ubuntu 22.04.5 LTS.

El servidor está activo permanentemente, recibiendo la información del dispositivo de monitoreo.

4.8.2 Comunicación y transmisión de datos

En la sección 4.6.2 se mencionaron los protocolos que se han utilizado localmente para la comunicación entre sensores, el ESP32 y la Raspberry Pi. Como la Raspberry Pi actúa tam-

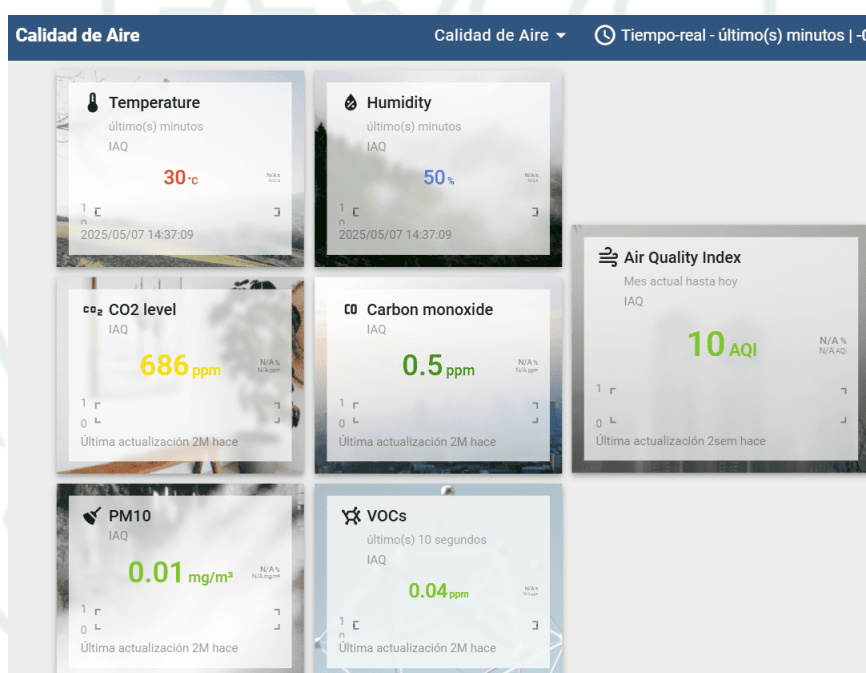
bién como Gateway, enviará toda la información obtenida hacia el servidor y específicamente a la plataforma ThingsBoard usando **broker MQTT** integrado en sus servicios.

4.8.3 Desarrollo de la plataforma web

En el servidor se instaló la versión Community Edition de ThingsBoard, una plataforma IoT de código abierto que cuenta con todos los elementos necesarios para el prototipo implementado.

La interfaz que se ha diseñado en ThingsBoard se muestra en la figura 4.37; en ella se encuentran los diferentes valores de medición de cada gas y el índice de calidad del aire. Las magnitudes de concentración tienen la propiedad de cambiar de color acorde a su rango en la tabla 4.7, facilitando la rápida inspección del usuario.

Figura 4.37
Panel de monitoreo en ThingsBoard

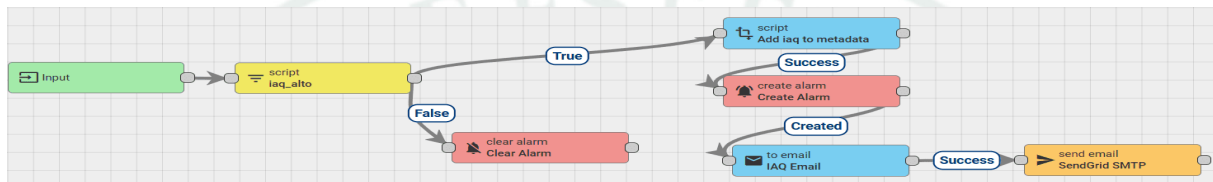


Nota. Elaboración propia basada en la plataforma ThingsBoard.

El motor de reglas (Rule engine) de ThingsBoard es un sistema altamente personalizable para el procesamiento y control de eventos (ThingsBoard, 2025). Se configuró el motor

de reglas para crear un sistema de alarma que, a través del protocolo simple de transferencia de correo (Simple Mail Transfer Protocol,SMTP) envía un correo al usuario advirtiéndolo de que la calidad del aire ha llegado a un nivel peligroso. La figura 4.38 muestra cómo se da este proceso.

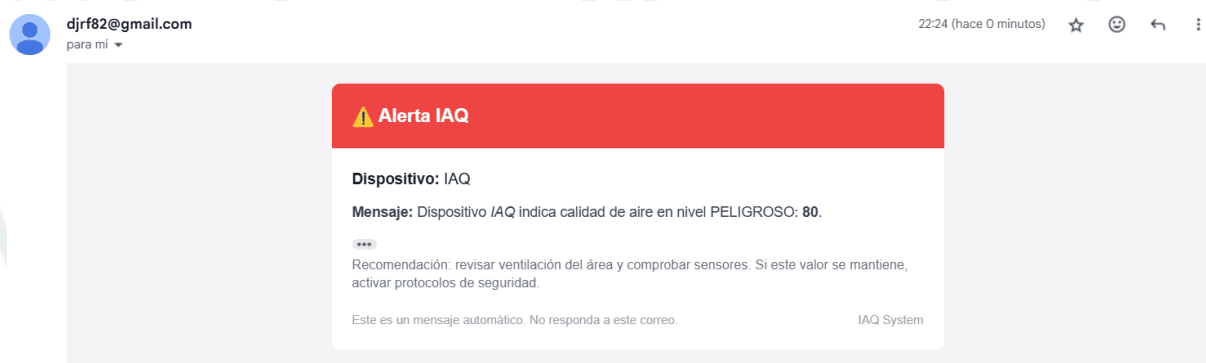
Figura 4.38
Configuración del motor de reglas



Nota. Elaboración propia basada en la plataforma ThingsBoard.

El correo de alerta se muestra así:

Figura 4.39
Correo con mensaje de alerta



Nota. Elaboración propia basada en la plataforma ThingsBoard.

4.9 Costos del prototipo

El prototipo diseñado cumple con la clasificación de dispositivo de bajo costo, ya que su costo total de implementación es de 104 USD, un valor significativamente inferior al umbral de 2000 USD establecido por Sá et al. (2022) para este tipo de dispositivos. La tabla adjunta muestra en detalle el costo de cada uno de sus componentes.

Tabla 4.13*Costo de los componentes del prototipo*

Dispositivo	Precio (USD)
PMS5003	18
ENS160+AHT20	3.63
ESP32	3.65
Placa madre	4.5
Raspberry Pi 3B	44.5
ZE15CO	16
Caja 3D	12
Pantalla LCD	2.39
TOTAL	104.67

Nota. Elaboración propia.

En cuanto a los costos del servicio en la nube, tiene un valor mensual de \$4.5.

4.10 Especificaciones técnicas del prototipo

A partir del diseño e implementación, las especificaciones técnicas del dispositivo se pueden resumir en el siguiente cuadro:

Tabla 4.14*Especificaciones técnicas del prototipo de medición de IAQ*

Especificaciones técnicas	
Rango de medición CO₂	0-2000 ppm
Rango de medición CO	0-500 ppm
Rango de medición VOC	0-65 ppm
Rango de medición PM10	0-500 ug/m3
Rango de medición temperatura	-20 a 80 °C
Rango de medición humedad	0-100 %
Salida PWM	Nivel lógico: 3.3 V Resolución: 10 bits Frecuencia: 1 kHz
Alimentación	5 V
Consumo máximo	400 mA
Conectividad inalámbrica	Wi-Fi (2.4 GHz, IEEE 802.11 b/g/n)

Nota. Elaboración propia.



CAPÍTULO V

Pruebas experimentales y resultados

5.1 Metodología de las pruebas experimentales

La validación del prototipo se realizó mediante su exposición controlada a diversos agentes contaminantes, incluyendo humo, polvo y líquidos de limpieza.

Durante las pruebas, el sistema registró las variables medidas por los sensores (CO_2 , CO, VOC y PM_{10}) y calculó el valor predicho de IAQ (0–100) mediante la red neuronal implementada. Simultáneamente, el sensor ENS160 proporcionó el valor de referencia de IAQ en una escala de 1 a 5, donde 1 es “Excelente”, 2 es “Bueno”, 3 es “Moderado”, 4 es “Insalubre” y 5 es “Peligroso”. Para efectos de comparación, se homologaron los niveles del sistema y el sensor ENS160 de la siguiente manera:

Tabla 5.1

Tabla de equivalencia de niveles e índice de IAQ

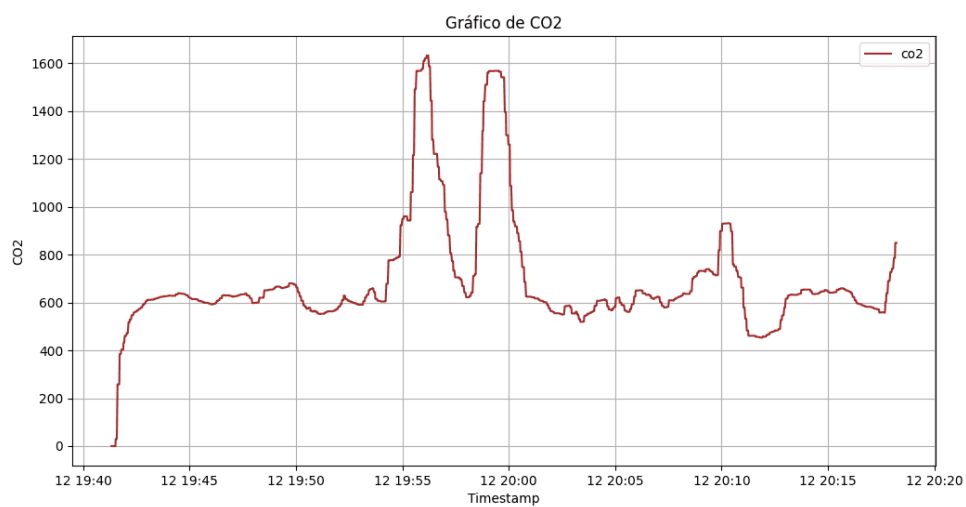
ENS160 (1–5)	Equivalencia en el sistema (0–100)	Clasificación usada
1-2	0–25	Bueno
3	26–50	Moderado
4	51–75	Insalubre
5	76–100	Peligroso

Nota. Clasificación de IAQ del sensor ENS160 y el sistema diseñado.

5.2 Resultados obtenidos

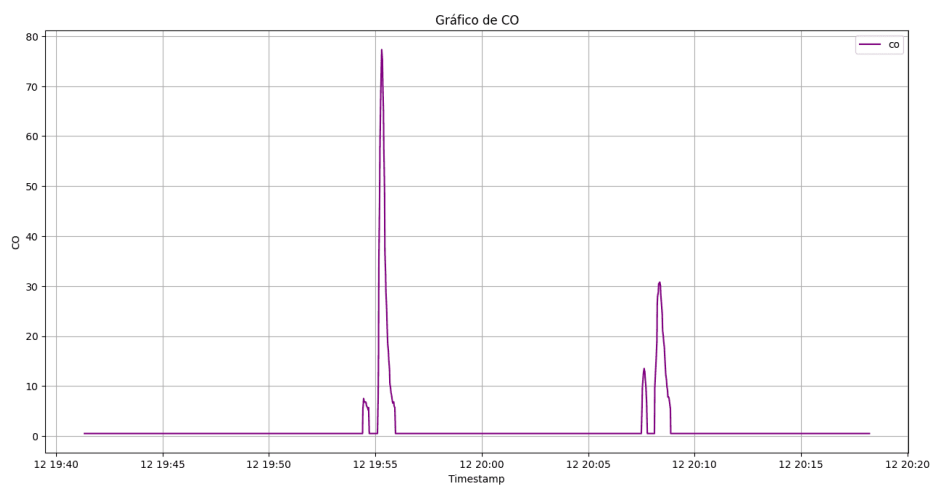
El dispositivo cuenta con una base de datos en formato CSV donde se registraron las mediciones, que posteriormente se procesaron para obtener los gráficos para cada gas:

Figura 5.1
Mediciones de CO₂



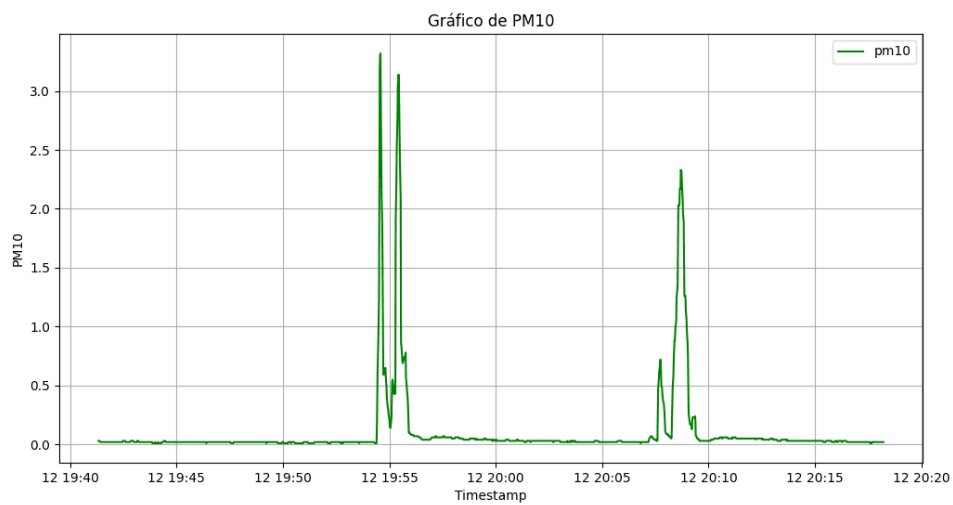
Nota. Elaboración propia.

Figura 5.2
Mediciones de CO



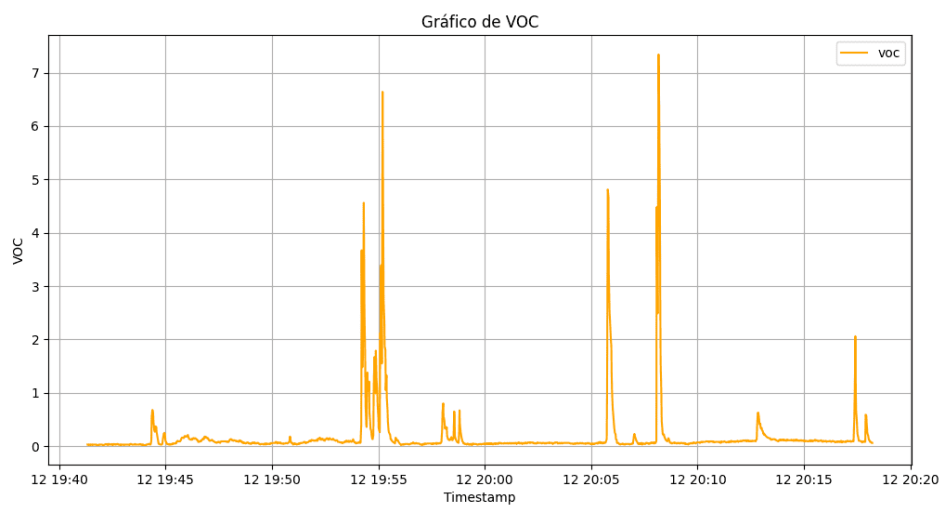
Nota. Elaboración propia.

Figura 5.3
Mediciones de PM10



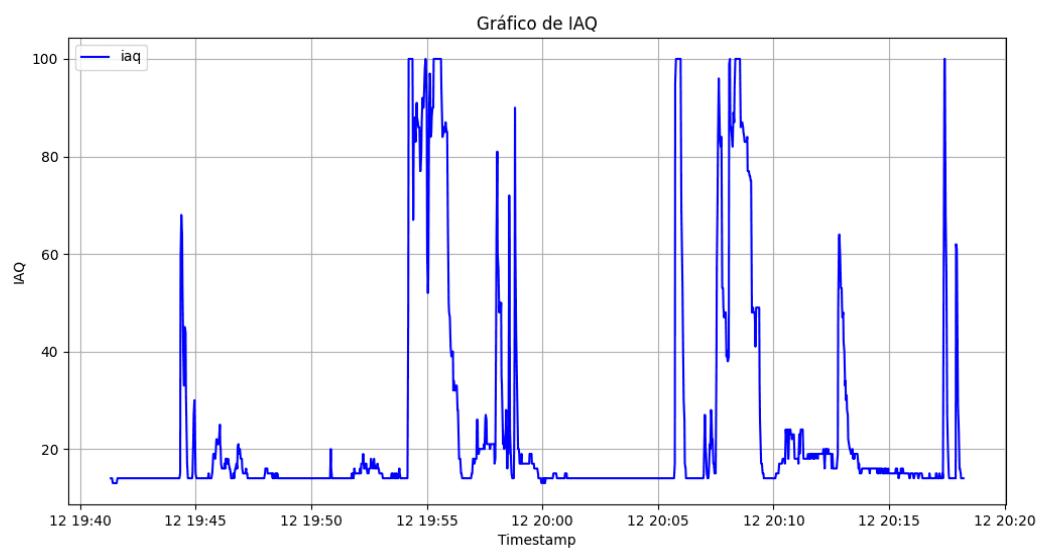
Nota. Elaboración propia.

Figura 5.4
Mediciones de VOC



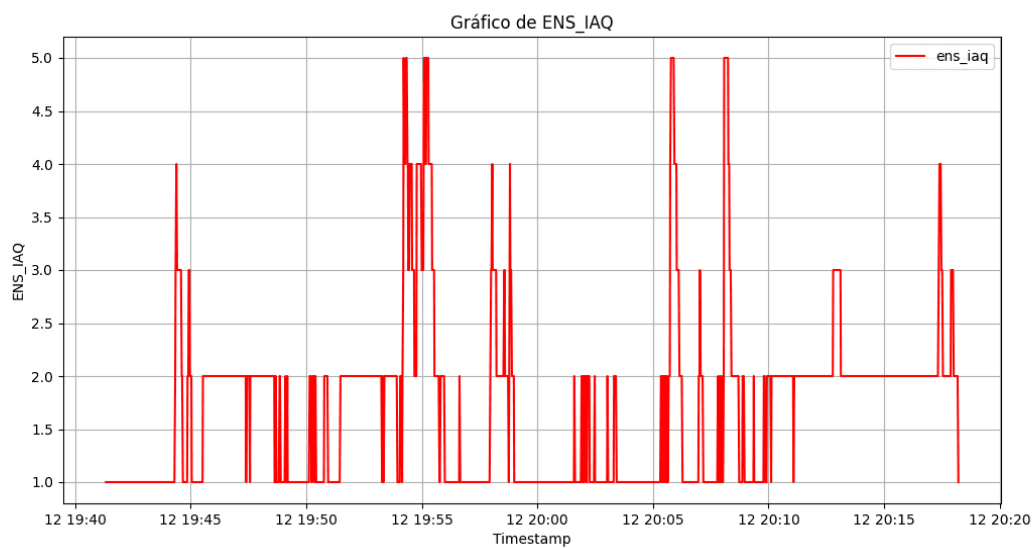
Nota. Elaboración propia.

Figura 5.5
Predicción de IAQ



Nota. Elaboración propia.

Figura 5.6
Medición de IAQ



Nota. Elaboración propia.

5.3 Análisis de congruencia de IAQ predicho vs ENS160

En concordancia con la Tabla 5.1, se compararon los resultados de las Figuras 5.5 y 5.6, correspondientes al IAQ predicho y medido por el ENS160, respectivamente, obteniendo un porcentaje de correlación de **86.81 %**.

El sistema presentó una correlación adecuada con el sensor ENS160. Esto valida la capacidad de la red neuronal para predecir la calidad del aire interior (IAQ) a partir de los contaminantes medidos.

5.4 Discusión de resultados

La correlación de 86.81 % evidencia una muy buena congruencia en la clasificación del nivel de calidad de aire. Este valor es considerado aceptable dentro del ámbito experimental, ya que las mediciones se realizaron en condiciones reales con variaciones ambientales propias del recinto. Además, debe considerarse la diferencia de escalas entre el IAQ predicho (0–100) y los valores discretos del ENS160 (1–5), por lo que la comparación implica una normalización de categorías. Aun así, al observar las Figuras 5.5 y 5.6, la tendencia general entre ambas mediciones se mantuvo coherente. El modelo neuronal implementado puede predecir de manera aceptable las fluctuaciones de calidad de aire a partir de las variables CO₂, CO, VOC y PM10; pero se requieren mediciones mucho más prolongadas y una mayor cantidad de datos distribuidos a lo largo de todo el rango de IAQ para mejorar la capacidad de generalización del modelo y optimizar su desempeño en condiciones variables. En síntesis, los ensayos realizados demuestran que el sistema desarrollado tiene la capacidad de estimar de forma confiable el índice de calidad de aire a partir de los gases medidos, manteniendo la coherencia con el sensor comercial de referencia. Este desempeño confirma la viabilidad del diseño propuesto, evidenciando su funcionalidad y potencial para el monitoreo de ambientes interiores en tiempo real.

5.5 Limitaciones

- Disponibilidad de referencias para validación de gases: Si bien se utilizó el IAQ proporcionado por el sensor ENS160 como referencia para comparar los valores predichos por el sistema y todos los sensores cuentan con capacidad de autocalibración, no se contó con referencias independientes para cada gas medido. Esto restringió una posible validación más exhaustiva de las concentraciones individuales de contaminantes, lo que ayudaría a incrementar la precisión del modelo.
- Necesidad de bases de datos reales en entornos específicos: Aunque los algoritmos de inteligencia artificial fueron evaluados en cuatro ambientes distintos (laboratorio, taller mecánico, cocina industrial y gimnasio), la base de datos principal utilizada para el entrenamiento de los modelos fue generada mediante simulación, a fin de tener suficientes datos de todos los niveles de IAQ. Aunque se realizaron pruebas controladas con contaminantes inducidos, sería ideal llevar a cabo mediciones prolongadas en entornos reales y específicos, para obtener una base de datos más representativa. Con esto se podría validar de manera más precisa el correcto funcionamiento del sistema y aumentar la eficiencia de predicción de la red neuronal.
- Inicialmente, se consideró usar un sensor para NO₂ y VOC (SGP41); sin embargo, este proporcionaba una salida adimensional que requería software del fabricante para su correcta interpretación. Esta limitación aumentaba la complejidad en la implementación y reducía la comparación de resultados, ya que se requieren unidades de medida como ppm o $\mu\text{g}/\text{m}^3$. Por ello, se reemplazó por el ENS160, que ofrece valores de VOC en ppm, a costa de prescindir de una medición directa de NO₂.

Conclusiones

- El sistema diseñado e implementado cumple satisfactoriamente con el objetivo general del proyecto, evidenciando la viabilidad de un sistema de monitoreo y control de la calidad del aire en interiores mediante el uso de internet de las cosas e inteligencia artificial.
- Se implementó un prototipo funcional capaz de medir concentraciones de CO, CO₂, VOC, PM10, temperatura y humedad, cuyo diseño cumple con las características recomendadas para sistemas de monitoreo ambiental de bajo costo, tal como se describe en el trabajo de Sá et al. (2022) (véase la Sección 2.9.1), donde se detallan los criterios de precisión, accesibilidad y capacidad de integración que guiaron el desarrollo del dispositivo.
- En el apartado de inteligencia artificial, se entrenaron y evaluaron redes neuronales, regresión Lasso y árboles de decisión. Siguiendo la recomendación de Wei et al. (2019) de comparar distintos algoritmos según el contexto, el análisis de la Sección 4.7.2 (Tablas 4.10, 4.11 y 4.12) mostró un rendimiento ligeramente superior de los árboles de decisión; sin embargo, se optó por redes neuronales debido a su mayor capacidad de generalización, modelado no lineal y mejor integración con sistemas en tiempo real, como el implementado en este proyecto.
- A partir del diseño del sistema de control presentado en la Sección 4.5, en la Sección 4.6.2 se implementó un esquema de regulación por umbrales capaz de generar una señal PWM proporcional al IAQ predicho. Aunque no se integró un actuador físico en esta fase, la salida PWM queda preparada para gobernar dispositivos de ventilación u otros actuadores. Además, la incorporación de rutinas de fail-safe e histéresis mejoró la estabilidad y seguridad del módulo de control, consolidando su aporte funcional dentro del sistema completo.
- En cuanto al componente de internet de las cosas, se implementó una arquitectura ba-

sada en el modelo IoT de nivel 3, descrita en la Sección 2.6.1, que integra la capa de sensado, el procesamiento local en la Raspberry Pi y la comunicación con la nube, junto con el sistema de alertas para valores elevados de IAQ (Sección 4.8.3). Con base en las pruebas presentadas en el Capítulo 5, se concluye que esta arquitectura permite un procesamiento local eficiente y una integración efectiva con la plataforma en la nube.

Finalmente, el presente proyecto de tesis completó los seis objetivos específicos planteados:

- **Describir y documentar ambientes representativos (tamaño, ocupación, actividades, ventilación) para la aplicación y prueba del dispositivo.** Con base en Chamseddine et al. (2025) y Persily et al. (2003), en el Capítulo 3 se analizaron cuatro ambientes interiores —cocina industrial, taller automotriz, laboratorio y gimnasio— describiendo sus características arquitectónicas, condiciones ambientales y sistemas de ventilación. Además, se identificaron los puntos óptimos para la instalación del dispositivo y se evaluó el impacto del IAQ en la salud y productividad de los ocupantes.
- **Diseñar, seleccionar e integrar los sensores y dispositivos necesarios para la implementación del prototipo de bajo costo.** Se desarrolló un prototipo basado en hardware libre —Raspberry Pi 3B, ESP32 y sensores de bajo costo— cuyo precio total aproximado de \$110 cumple con la clasificación de “dispositivo de bajo costo” propuesta en Sá et al. (2022). La integración de todos los componentes previstos en el diseño confirma la factibilidad técnica del sistema y la correcta interoperabilidad entre sensores heterogéneos en un entorno de procesamiento en tiempo real.
- **Diseñar y desarrollar un proceso de simulación de datos de calidad de aire, definiendo segmentos temporales, rangos de concentración y niveles de calidad del aire para generar mediciones sintéticas de los sensores.** En la Sección 4.7.1 se describe el proceso de generación de datos simulados para los cuatro ambientes definidos, siguiendo los parámetros establecidos en la Sección 2.1, el Capítulo 3 y el algoritmo presentado en el apartado 4.5. Se simularon valores de CO, CO₂, VOC y PM₁₀ durante un período

de 24 horas, incorporando variaciones temporales propias de cada entorno y aplicando filtros de suavizado para mejorar la estabilidad de las curvas. Finalmente, se realizaron tareas de preprocesamiento y normalización para conformar un dataset consistente y balanceado, utilizado como base para el modelado predictivo del IAQ.

- **Entrenar diferentes algoritmos de inteligencia artificial y evaluar su capacidad predictiva.** Se entrenaron y compararon redes neuronales artificiales, regresión Lasso y árboles de decisión. Aunque estos últimos obtuvieron un desempeño ligeramente superior, se seleccionaron las redes neuronales debido a su mayor capacidad de generalización y adecuación al funcionamiento en tiempo real del prototipo. La evaluación de múltiples algoritmos respondió además a la recomendación planteada por Wei et al. (2019), quienes enfatizan la importancia de comparar distintos enfoques para identificar el más adecuado según el caso de estudio.
- **Implementar un sistema de monitoreo en la nube integrado con un mecanismo local de respuesta automática proporcional al índice de calidad del aire, complementado con un sistema de alertas remotas (SMTP) para valores extremos.** Se desarrolló una arquitectura IoT de nivel 3 que integra la plataforma ThingsBoard para el monitoreo remoto en tiempo real de los parámetros de calidad de aire. Además, se habilitó un sistema de alertas SMTP para valores extremos y se implementó un control local proporcional mediante PWM, listo para operar con distintos actuadores y escalar hacia estrategias de control más avanzadas.
- **Realizar pruebas experimentales con diferentes fuentes de contaminación y analizar el desempeño del sistema comparándolo con un sensor de referencia, evaluando la correlación en la predicción del índice de calidad del aire.** Mediante la inducción de diferentes agentes contaminantes, se determinó una correlación de 86.81 % entre los valores de IAQ predichos y los indicados por el sensor de referencia.

Referencias

- Ahn, J., Shin, D., Kim, K., & Yang, J. (2017). Indoor air quality analysis using deep learning with sensor data. *Sensors*, 17(11). <https://doi.org/10.3390/s17112476>
- Aosong Electronics Co., Ltd. (2021, mayo). *Data sheet AHT21 — humidity and temperature sensor* [Version V1.0]. Aosong Electronics Co., Ltd. China. <https://www.aosong.com/userfiles/files/media/Data%20Sheet%20AHT21.pdf>
- Bahga, A., & Madisetti, V. (2014). Internet of Things: A Hands-On Approach [Section 1.5: IoT Levels and Deployment Templates]. Universities Press. <https://doi.org/10.5555/2721615>
- Benammar, M., Abdaoui, A., Ahmad, S. H., Touati, F., & Kadri, A. (2018). A modular IoT platform for real-time indoor air quality monitoring. *Sensors*, 18(2). <https://doi.org/10.3390/s18020581>
- Burkov, A. (2019). *The hundred-page machine learning book*. Andriy Burkov. <https://books.google.com.pe/books?id=0jbxwQEACAAJ>
- Chamseddine, A., Elzein, I., & Hassan, N. (2025). Indoor air quality in critical indoor environments: A review paper. *Water, Air, & Soil Pollution*, 236. <https://doi.org/10.1007/s11270-025-08512-y>
- Das, P., Shrubsole, C., Jones, B., Hamilton, I., Chalabi, Z., Davies, M., Mavrogianni, A., & Taylor, J. (2014). Using probabilistic sampling-based sensitivity analyses for indoor air quality modelling. *Building and Environment*, 78, 171-182. <https://doi.org/10.1016/j.buildenv.2014.04.017>
- Elbayoumi, M., Ramli, N. A., & Yusof, N. F. F. M. (2015). Development and comparison of regression models and feedforward backpropagation neural network models to predict seasonal indoor PM_{2.5-10} and PM_{2.5} concentrations in naturally ventilated schools. *Atmospheric Pollution Research*, 6(6), 1013-1023. <https://doi.org/10.1016/j.apr.2015.09.001>

- González, F. (2023). *Sensores electroquímicos: Teoría y aplicaciones*. Universidad del Valle.
<https://books.google.com.pe/books?id=IffNEAAQBAJ>
- Goyal, S. (2021). IoT-based smart air quality control system: Prevention of COVID-19. https://doi.org/10.1007/978-3-030-73885-3_2
- Guth, J., Breitenbücher, U., Falkenthal, M., Leymann, F., & Reinfurt, L. (2016). Comparison of IoT platform architectures: a field study based on a reference architecture. *2016 Cloudification of the Internet of Things (CIoT)*, 1-6. <https://doi.org/10.1109/CIOT.2016.7872918>
- Hassan, Q. F. (2018, junio). *Internet of things A to Z: Technologies and applications*.
- Jabbar, W. A., Subramaniam, T., Ong, A. E., Shu'ib, M. I., Wu, W., & de Oliveira, M. A. (2022). LoRaWAN-based IoT system implementation for long-range outdoor air quality monitoring. *Internet of Things*, 19, 100540. <https://doi.org/10.1016/j.iot.2022.100540>
- Kang, J., & Hwang, K.-I. (2016). A comprehensive real-time indoor air-quality level indicator. *Sustainability*, 8. <https://doi.org/10.3390/su8090881>
- Loy-Benitez, J., Vilela, P., Li, Q., & Yoo, C. (2019). Sequential prediction of quantitative health risk assessment for the fine particulate matter in an underground facility using deep recurrent neural networks. *Ecotoxicology and Environmental Safety*, 169, 316-324. <https://doi.org/10.1016/j.ecoenv.2018.11.024>
- Mad Saad, S., Shakaff, A. Y., Saad, A., Yusof, A., Andrew, A. M., Zakaria, A., & Adom, A. (2017). Development of indoor environmental index: air quality index and thermal comfort index. *AIP Conference Proceedings*, 1808, 020043. <https://doi.org/10.1063/1.4975276>
- Ministerio del Ambiente. (2016). Resolución Ministerial N° 181-2016-MINAM [Consultado el 2024-06-22]. <https://www.minam.gob.pe/disposiciones/resolucion-ministerial-n-181-2016-minam/>
- Pepper, D., & Carrington, D. (2009, marzo). *Modeling indoor air pollution*. <https://doi.org/10.1142/P612>

- Pérez García, M. A. (2014). *Instrumentación electrónica*. Ediciones Paraninfo, S.A.
- Persily, A., Musser, A., Emmerich, S., & Taylor, M. (2003). *Simulations of Indoor Air Quality and Ventilation Impacts of Demand Controlled Ventilation in Commercial and Institutional Buildings* (NIST Interagency/Internal Report (NISTIR)). National Institute of Standards y Technology (NIST). Gaithersburg, MD. https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=860924
- Plantower. (2025). PMS5003 datasheet [Consultado el 2025-08-30]. https://www.plantower.com/en/products_33/74.html
- Ponce, P. (2010). *Inteligencia artificial con aplicaciones a la ingeniería*. Alfaomega Grupo Editor, S.A. de C.V., México. <https://lelinopontes.wordpress.com/wp-content/uploads/2014/09/inteligencia-artificial-con-aplicaciones-a-la-ingenierc3ada.pdf>
- Sá, J. P., Alvim-Ferraz, M. C. M., Martins, F. G., & Sousa, S. I. (2022). Application of the low-cost sensing technology for indoor air quality monitoring: A review. *Environmental Technology & Innovation*, 28, 102551. <https://doi.org/10.1016/j.eti.2022.102551>
- Saini, J., Dutta, M., & Marques, G. (2021). Predicting indoor air quality: Integrating IoT with artificial intelligence. En *Internet of Things for Indoor Air Quality Monitoring* (pp. 51-67). Springer International Publishing. https://doi.org/10.1007/978-3-030-82216-3_4
- Saini, J., Dutta, M., & Marques, G. (2022). Modeling indoor PM2.5 using adaptive dynamic fuzzy inference system tree (ADFIST) on internet of things-based sensor network data. *Internet of Things*, 20, 100628. <https://doi.org/10.1016/j.iot.2022.100628>
- Saleem, T. J., & Chishti, M. A. (2021). Deep learning for the Internet of Things: Potential benefits and use cases. *Digital Communications and Networks*, 7(4), 526-542. <https://doi.org/10.1016/j.dcan.2020.12.002>
- Schiffner, K. (2013). *Air pollution control equipment selection guide*. CRC Press. <https://books.google.com.pe/books?id=VIPSBQAAQBAJ>

- ScioSense B.V. (2023, marzo). *ENS160 digital metal-oxide multi-gas sensor datasheet* [Version 1.3, Release Date 2023-03-29]. ScioSense B.V. The Netherlands. <https://www.sciosense.com/wp-content/uploads/2023/12/ENS160-Datasheet.pdf>
- Sensirion. (2022). What is a metal oxide (MOX) sensor? [Consultado el 2024-01-15]. https://sensirion.com/media/documents/0083CDF4/6294DFEA/Info_Note_MOX_sensor.pdf
- Shin, S., Baek, K., & So, H. (2023). Rapid monitoring of indoor air quality for efficient HVAC systems using fully convolutional network deep learning models. *Building and Environment*, 234, 110191. <https://doi.org/10.1016/j.buildenv.2023.110191>
- Taştan, M., & Gökozan, H. (2019). Real-time monitoring of indoor air quality with Internet of Things-based e-nose. *Applied Sciences*, 9(16). <https://doi.org/10.3390/app9163435>
- ThingsBoard. (2025). ThingsBoard rule engine [Consultado el 2025-08-16]. <https://thingsboard.io/docs/user-guide/rule-engine-2-0/overview>
- U.S. Environmental Protection Agency. (2023). Introduction to indoor air quality [Consultado el 2023-08-10]. <https://www.epa.gov/indoor-air-quality-iaq/introduction-indoor-air-quality>
- U.S. Environmental Protection Agency. (2024, mayo). *Technical assistance document for the reporting of daily air quality – the air quality index (AQI)* [EPA-454/B-24-002, 32 pages]. U.S. Environmental Protection Agency, Office of Air Quality Planning y Standards. Research Triangle Park, NC, USA. <https://document.airnow.gov/technical-assistance-document-for-the-reporting-of-daily-air-quality.pdf>
- Wang, C., Zhang, F., Wang, J., Doyle, J. K., Hancock, P. A., Mak, C. M., & Liu, S. (2021). How indoor environmental quality affects occupants' cognitive functions: a systematic review. *Building and Environment*, 193, 107647. <https://doi.org/10.1016/j.buildenv.2021.107647>
- Wei, W., Ramalho, O., Malingre, L., Sivanantham, S., Little, J. C., & Mandin, C. (2019). Machine learning and statistical models for predicting indoor air quality. *Indoor Air*, 29(5), 704-726. <https://doi.org/10.1111/ina.12580>

- World Health Organization. (2009). *WHO guidelines for indoor air quality: dampness and mould*.
- World Health Organization. (2010). *WHO guidelines for indoor air quality: selected pollutants*.
- World Health Organization. (2018). 9 out of 10 people worldwide breathe polluted air [Consultado el 2023-08-09]. <https://www.who.int/news-room/detail/02-05-2018-9-out-of-10-people-worldwide-breathe-polluted-air-but-more-countries-are-taking-action>
- World Health Organization. (2022). Household air pollution [Consultado el 2023-08-09]. <https://www.who.int/es/news-room/fact-sheets/detail/household-air-pollution-and-health>
- Zanella, A., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M. (2014). Internet of Things for Smart Cities. *IEEE Internet of Things Journal*, 1(1), 22-32. <https://doi.org/10.1109/JIOT.2014.2306328>
- Zhao, L., Wu, W., & Li, S. (2019). Design and implementation of an IoT-based indoor air quality detector with multiple communication interfaces. *IEEE Internet of Things Journal*, 6(6), 9621-9632. <https://doi.org/10.1109/JIOT.2019.2930191>
- Zhengzhou Winsen Electronics Technology Co., Ltd. (2016, enero). *MH-Z19B NDIR CO2 module user's manual* [Intelligent Infrared CO₂ Module MH-Z19B — User Manual, Valid from 2016-01-21]. Ver. 1.0. Zhengzhou Winsen Electronics Technology Co., Ltd. Zhengzhou, China. https://www.winsen-sensor.com/d/files/infrared-gas-sensor/mh-z19b-co2-ver1_0.pdf
- Zhengzhou Winsen Electronics Technology Co., Ltd. (2018, abril). *ZE15-CO electrochemical carbon monoxide gas module manual* [Electrochemical CO gas module ZE15-CO — User Manual]. Ver. 1.1. Zhengzhou Winsen Electronics Technology Co., Ltd. Zhengzhou, China. https://www.winsen-sensor.com/d/files/ze15-co-module-manual-v1_1.pdf

Anexo A

Código fuente del ESP32

El siguiente código corresponde al programa implementado en el microcontrolador ESP32 para la lectura de sensores, envío de datos al servidor y control del ventilador de extracción.

Código A.1. Código principal del ESP32 para la medición y control del IAQ.

```
1
2 #include <Arduino.h>
3 #include <Adafruit_AHTX0.h>
4 #include <Wire.h>
5 #include "ScioSense_ENS160.h"
6 #include "PMS.h"
7 #include "MHZ19.h"
8 #include <SoftwareSerial.h>
9 #include <TFT_eSPI.h>
10 #include <SPI.h>
11
12 TFT_eSPI tft = TFT_eSPI();
13 TFT_eSprite spr = TFT_eSprite(&tft);
14
15 ////////////////////////////////// CONFIGURACIÓN DE SENSORES //////////////////////////////////
16
17 // --- Sensor ZE15-CO (monóxido de carbono) ---
18 #define RXD1 18
19 #define TXD1 19
20 static const int REQUEST_CNT = 9;
21 static const int RESPONSE_CNT = 9;
22 uint8_t getppm[REQUEST_CNT] = {0xff, 0x01, 0x86, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x79};
```

```

23 #define WAIT_READ_TIMES 100
24 #define WAIT_READ_DELAY 10
25 float measurement; // Valor de CO (ppm)
26
27 // --- Sensor AHT21 (temperatura y humedad) ---
28 Adafruit_AHTX0 aht;
29 int tempC;
30 int humidity;
31
32 // --- Sensor ENS160 (VOC e IAQ) ---
33 ScioSense_ENS160 ens160(ENS160_I2CADDR_1); // Dirección I2C: 0x53
34 float voc;
35 int ens_iaq;
36
37 // --- Sensor MHZ19B (CO2) ---
38 #define RX_PIN 5
39 #define TX_PIN 15
40 #define BAUDRATE 9600
41 MHZ19 myMHZ19;
42 SoftwareSerial mySerial2(RX_PIN, TX_PIN);
43 unsigned long getDataTimer = 0;
44
45 // --- Sensor PMS5003 (material particulado PM10) ---
46 PMS pms(Serial2);
47 PMS::DATA data;
48 #define RXD2 16
49 #define TXD2 17
50
51 ////////////////////////////////////////////////// CONTROL DEL VENTILADOR (PWM) ///////////////////////////////////
52
53 const int PWM_PIN = 2;
54 const int PWM_CHANNEL = 0;
55 const int PWM_FREQ = 1000;
56 const int PWM_RESOLUTION = 10;

```

```

57
58 // Umbral de control con histéresis (según IAQ)
59 const int T1_ON = 26;
60 const int T1_OFF = 23;
61 const int T2_ON = 51;
62 const int T2_OFF = 48;
63 const int T3_ON = 76;
64 const int T3_OFF = 73;
65
66 // Ciclos de trabajo PWM
67 const int DUTY_OFF = 0;
68 const int DUTY_LOW = 256;
69 const int DUTY_MED = 512;
70 const int DUTY_HIGH = 768;
71
72 // Variables de control
73 int currentDuty = DUTY_OFF;
74 int lastLevel = 0;
75 unsigned long lastUpdateTime = 0;
76 const unsigned long TIMEOUT_MS = 60000;
77 int iaq = 0; // IAQ recibido desde Raspberry Pi
78
79 ///////////////////////////////////////////////////
80 /////////////////////////////////////////////////// SETUP ///////////////////////////////////////////////////
81 ///////////////////////////////////////////////////
82
83 void setup() {
84     Serial.begin(9600);
85     Serial2.begin(9600, SERIAL_8N1, RXD2, TXD2); // PMS5003
86     Serial1.begin(9600, SERIAL_8N1, RXD1, TXD1); // ZE15-C0
87     Serial1.flush();
88
89     // Inicialización ENS160
90     Serial.print("Iniciando ENS160...");

```

```

91  ens160.begin();
92  if (ens160.available()) {
93      ens160.setMode(ENS160_OPMODE_STD);
94      Serial.println("OK");
95  }
96
97  // Inicialización AHT21
98  if (!aht.begin()) {
99      Serial.println("Error: no se detecta AHT21. Verificar conexiones.");
100     while (1) delay(10);
101 }
102
103 // Inicialización MHZ19B
104 mySerial2.begin(BAUDRATE);
105 myMHZ19.begin(mySerial2);
106 myMHZ19.autoCalibration();
107
108 // Inicialización del display TFT
109 tft.init();
110 tft.setRotation(1);
111 spr.createSprite(tft.width(), tft.height());
112 spr.setTextColor(TFT_WHITE, TFT_BLACK);
113 spr.setTextSize(1);
114
115 // Configuración del PWM para ventilador
116 ledcAttachChannel(PWM_PIN, PWM_FREQ, PWM_RESOLUTION, PWM_CHANNEL);
117 ledcWrite(PWM_PIN, DUTY_OFF);
118 }
119
120 ///////////////////////////////////////////////////////////////////
121 /////////////////////////////////////////////////////////////////// LOOP ///////////////////////////////////////////////////////////////////
122 ///////////////////////////////////////////////////////////////////
123
124 void loop() {

```

```

125
126 // --- Lectura de temperatura y humedad (AHT21) ---
127 sensors_event_t humidity1, temp;
128 aht.getEvent(&humidity1, &temp);
129 tempC = temp.temperature;
130 humidity = humidity1.relative_humidity;
131
132 // --- Lectura de VOC e IAQ (ENS160) ---
133 if (ens160.available()) {
134     ens160.set_envdata(tempC, humidity);
135     ens160.measure(true);
136     ens160.measureRaw(true);
137     voc = ens160.getTVOC() * 0.001; // Conversión ppb -> ppm
138     ens_iaq = ens160.getAQI();
139 }
140
141 // --- Lectura de CO (ZE15-CO) ---
142 byte buf[RESPONSE_CNT] = {0};
143 writeCommand(getppm, buf);
144 uint8_t checksum_calc = 0;
145 for (int i = 1; i < 8; i++) checksum_calc += buf[i];
146 checksum_calc = 255 - checksum_calc + 1;
147 measurement = (buf[0] == 0xff && buf[1] == 0x86 && buf[8] ==
    checksum_calc)
    ? (buf[2] * 256 + buf[3]) * 0.1
    : -1;
149
150
151 // --- Lectura de CO2 (MHZ19B) ---
152 int CO2 = myMHZ19.getCO2();
153 delay(1000);
154
155 // --- Lectura de material particulado PM10 (PMS5003) ---
156 if (pms.readUntil(data)) {
157     float pm10 = data.PM_AE_UG_10_0 / 1000.0; // Conversión a mg/m3

```

```

158
159 // Envío de datos al puerto serial
160 Serial.println(String(pm10) + "," + String(tempC) + "," +
    String(humidity) + "," +
161         String(voc) + "," + String(measurement) + "," +
    String(CO2) + "," + String(ens_iaq));
162
163 // Lectura del valor IAQ enviado desde la Raspberry Pi
164 if (Serial.available()) {
165     String input = Serial.readStringUntil('\n');
166     input.trim();
167     if (input.length() > 0 && input.toInt() >= 0 && input.toInt() <=
        100) {
168         iaq = input.toInt();
169         lastUpdateTime = millis();
170         actualizarVentilador(iaq);
171     } else {
172         Serial.println("Dato IAQ inválido.");
173     }
174 }
175
176 // Modo seguro si no se recibe IAQ
177 if (millis() - lastUpdateTime > TIMEOUT_MS) {
178     if (currentDuty != DUTY_MED) {
179         Serial.println("Timeout IAQ: activando modo seguro (velocidad
            media).");
180         ledcWrite(PWM_PIN, DUTY_MED);
181         currentDuty = DUTY_MED;
182     }
183 }
184
185 // Actualización de la pantalla
186 drawDisplay(pm10, CO2, tempC, humidity);
187 }

```

```

188 }
189
190 ///////////////////////////////////////////////////
191 /////////////////////////////////// FUNCIONES AUXILIARES ///////////////////////////////////
192 ///////////////////////////////////////////////////
193
194 // Envía comando de lectura al sensor ZE15-CO
195 void writeCommand(uint8_t cmd[], uint8_t *response) {
196     while (Serial1.available()) Serial1.read();
197     Serial1.write(cmd, REQUEST_CNT);
198     Serial1.flush();
199
200     if (response != NULL) {
201         int i = 0;
202         while (Serial1.available() < RESPONSE_CNT) {
203             if (++i > WAIT_READ_TIMES) {
204                 Serial.println("Sin respuesta del ZE15-CO.");
205                 return;
206             }
207             delay(WAIT_READ_DELAY);
208         }
209         Serial1.readBytes(response, RESPONSE_CNT);
210     }
211 }
212
213 // Muestra los datos principales en el display TFT
214 void drawDisplay(float pm10, float co2_ppm, int tC, int H) {
215     uint16_t bgColor;
216     String estadoIAQ;
217
218     if (iaq <= 25) {
219         bgColor = TFT_GREEN; estadoIAQ = "Bueno";
220     } else if (iaq <= 50) {
221         bgColor = TFT_YELLOW; estadoIAQ = "Moderado";

```

```

222 } else if (iaq <= 75) {
223     bgColor = TFT_ORANGE; estadoIAQ = "Malo";
224 } else {
225     bgColor = TFT_RED; estadoIAQ = "Peligroso";
226 }
227
228 spr.fillSprite(bgColor);
229 spr.setTextDatum(TL_DATUM);
230 spr.setTextSize(2);
231 spr.setTextColor(TFT_BLACK, bgColor);
232 spr.drawString("IAQ: " + estadoIAQ, 2, 2);
233 spr.drawLine(0, 18, 159, 18, TFT_WHITE);
234
235 spr.setTextSize(1);
236 spr.drawString("CO: " + String(measurement, 2) + " ppm", 5, 25);
237 spr.drawString("CO2: " + String(co2_ppm, 1) + " ppm", 5, 40);
238 spr.drawString("VOC: " + String(voc, 2) + " ppm", 5, 55);
239 spr.drawString("PM10: " + String(pm10, 3) + " mg/m3", 5, 70);
240 spr.drawString("Temp: " + String(tC) + "C", 5, 85);
241 spr.drawString("Hum: " + String(H) + " %HR", 5, 100);
242 spr.drawString("IAQ: " + String(iaq) + " /100", 5, 115);
243 spr.pushSprite(0, 0);
244 }
245
246 // Control del ventilador con histéresis en función del IAQ
247 void actualizarVentilador(int iaq) {
248     int nuevoDuty = DUTY_OFF;
249
250     if (lastLevel == 0 && iaq >= T1_ON) { nuevoDuty = DUTY_LOW; lastLevel =
        1; }
251     else if (lastLevel == 1 && iaq < T1_OFF) { nuevoDuty = DUTY_OFF;
        lastLevel = 0; }
252     else if (lastLevel == 1 && iaq >= T2_ON) { nuevoDuty = DUTY_MED;
        lastLevel = 2; }

```

```
253 else if (lastLevel == 2 && iaq < T2_OFF) { nuevoDuty = DUTY_LOW;
    lastLevel = 1; }
254 else if (lastLevel == 2 && iaq >= T3_ON) { nuevoDuty = DUTY_HIGH;
    lastLevel = 3; }
255 else if (lastLevel == 3 && iaq < T3_OFF) { nuevoDuty = DUTY_MED;
    lastLevel = 2; }

256
257 if (nuevoDuty != currentDuty) {
258     ledcWrite(PWM_PIN, nuevoDuty);
259     currentDuty = nuevoDuty;
260     Serial.printf("IAQ: %d -> Nivel: %d -> PWM: %d\n", iaq, lastLevel,
        nuevoDuty);
261 }
262 }
```

Anexo B

Código de generación de datos simulados de gases

El código adjunto genera valores simulados para los distintos gases contaminantes, siguiendo variaciones temporales representativas de un ambiente controlado. El mismo programa se usa para cada uno de los ambientes que se simularon, solo se cambian los segmentos de tiempo y el gas para el que se generarán los datos.

Código B.1. Código principal del ESP32 para la medición y control del IAQ.

```
1
2 import numpy as np
3 import pandas as pd
4 import matplotlib.pyplot as plt
5 from scipy.signal import savgol_filter
6
7 # =====
8 # CONFIGURACIÓN DE RANGOS POR CONTAMINANTE Y NIVEL DE CALIDAD
9 # =====
10 # Cada contaminante tiene un rango de valores por nivel:
11 # B = Bueno, M = Medio, I = Insalubre, P = Peligroso
12 # Formato: (CO2, CO, NO2, VOC, PM10)
13 index = {
14     "B": ((340, 600), (0.0, 1.7), (0.0, 0.021), (0.0, 0.087), (0.0, 0.02)),
15     "M": ((601, 1000), (1.8, 8.7), (0.022, 0.08), (0.088, 0.261), (0.021,
16         0.150)),
17     "I": ((1001, 1500), (8.8, 10), (0.09, 0.17), (0.262, 0.43), (0.151,
18         0.180)),
19     "P": ((1501, 5000), (10.1, 50), (0.18, 5), (0.44, 3.00), (0.181,
20         0.600)),
21 }
```

```

20
21 # =====
22 # FUNCIÓN PARA CALCULAR EL ÍNDICE IAQ DE UN GAS ESPECÍFICO
23 # =====
24 def calcular_IAQ(nivel, valor, gas):
25     """
26     Calcula el valor de IAQ (0 a 100) para un contaminante
27     según el nivel de calidad y el valor medido.
28
29     Parámetros:
30         nivel (str): Nivel de calidad ('B', 'M', 'I', 'P')
31         valor (float): Concentración del gas
32         gas (int): Índice del contaminante (0=CO2, 1=CO, 2=NO2, 3=VOC,
33             4=PM10)
34     """
35     niveles = {"B": [0, 25], "M": [26, 50], "I": [51, 75], "P": [76, 100]}
36     rango = index[nivel][gas]
37
38     if rango[0] <= valor <= rango[1]:
39         iaq = ((niveles[nivel][1] - niveles[nivel][0]) /
40             (rango[1] - rango[0])) * (valor - rango[0]) +
41             niveles[nivel][0]
42
43         return iaq
44     return 0
45
46 # =====
47 # GENERACIÓN DE DATOS SIMULADOS PARA 24 HORAS (86400 s)
48 # =====
49 valores_aleatorios = np.random.rand(86400)
50 LB = [] # Lista de valores IAQ generados
51
52 # Segmentos de tiempo (simulan cambios de condiciones)
53 segmentos = [

```

```

52     (0, 30300, "B", (0.0, 0.087)),          # Bueno (00:00 - 08:25)
53     (30300, 36180, "M", (0.088, 0.261)),    # Medio (08:25 - 10:03)
54     (36180, 38460, "I", (0.262, 0.43)),    # Insalubre
55     (38460, 39660, "P", (0.44, 3.00)),    # Peligroso
56     (39660, 41280, "I", (0.262, 0.43)),    # Insalubre
57     (41280, 45120, "M", (0.088, 0.261)),    # Medio
58     (45120, 50280, "B", (0.0, 0.087)),    # Bueno
59     (50280, 51180, "M", (0.088, 0.261)),    # Medio
60     (51180, 51840, "I", (0.262, 0.43)),    # Insalubre
61     (51840, 52740, "P", (0.44, 3.00)),    # Peligroso
62     (52740, 53280, "I", (0.262, 0.43)),    # Insalubre
63     (53280, 56640, "M", (0.088, 0.261)),    # Medio
64     (56640, 57720, "I", (0.262, 0.43)),    # Insalubre
65     (57720, 59520, "M", (0.088, 0.261)),    # Medio
66     (59520, 86400, "B", (0.0, 0.087)),    # Bueno (final)
67 ]
68
69 # =====
70 # CONSTRUCCIÓN DE SEÑALES Y CÁLCULO DE IAQ
71 # =====
72 gas = 3 # 3 corresponde a VOC
73 for inicio, fin, nivel, rango in segmentos:
74     seg = valores_aleatorios[inicio:fin]
75     # Escala de valores dentro del rango del nivel actual
76     valores = rango[1] - seg * (rango[1] - rango[0])
77     # Cálculo del IAQ para el segmento
78     for v in valores:
79         LB.append(round(calcular_IAQ(nivel, v, gas)))
80
81 LB = np.array(LB)
82
83 # =====
84 # FILTRADO Y SUAVIZADO DE LA SEÑAL
85 # =====

```

```

86 LB_savgol = savgol_filter(LB, window_length=51, polyorder=2) # Filtro
    Savitzky-Golay
87 LB_movmean = pd.Series(LB).rolling(50).mean() # Media móvil
88
89 # =====
90 # GRAFICADO DE RESULTADOS
91 # =====
92 fig, axes = plt.subplots(3, 1, figsize=(8, 6), sharex=False)
93 axes[0].plot(LB, color="blue", linestyle=":", linewidth=2)
94 axes[0].set_title("Sin Suavizado")
95 axes[0].set_ylabel("VOC IAQ Index")
96
97 axes[1].plot(LB_savgol, color="green", linestyle=":", linewidth=2)
98 axes[1].set_title("Filtro Savitzky-Golay")
99
100 axes[2].plot(LB_movmean, color="red", linestyle=":", linewidth=2)
101 axes[2].set_title("Filtro de Media Móvil")
102 axes[2].set_xlabel("Tiempo (s)")
103
104 fig.suptitle("Simulación de VOC IAQ Index (24 h)")
105 plt.subplots_adjust(hspace=0.5)
106 plt.show()

```

Anexo C

Algoritmo de regresión LASSO

En este anexo se detalla el procedimiento de implementación del algoritmo de **regresión LASSO** (*Least Absolute Shrinkage and Selection Operator*) aplicado a la predicción del *Índice de Calidad del Aire* (IAQ).

El proceso se desarrolló en dos etapas principales:

1. **Normalización de los datos:** las variables de entrada, correspondientes a las concentraciones de CO_2 , CO , VOC y $PM10$, fueron transformadas al rango $[0,1]$ mediante el método `MinMaxScaler`. Esta etapa garantiza que todas las características contribuyan de manera equitativa durante el entrenamiento del modelo.
2. **Calibración del modelo:** una vez normalizados los datos, se entrenó el modelo de regresión LASSO empleando el conjunto de datos preparado. El objetivo de esta calibración fue ajustar el parámetro de regularización, que controla el grado de penalización sobre los coeficientes, determinando el balance óptimo entre precisión y simplicidad del modelo.

Ambos procesos se desarrollaron en el entorno *PySpark* y la plataforma de cómputo de *databricks.com*, lo cual permitió manejar de forma eficiente grandes volúmenes de datos y realizar el entrenamiento en un contexto de procesamiento distribuido. En las siguientes secciones se presenta el código fuente correspondiente a cada una de las etapas descritas.

Código C.1. Normalización de los datos en PySpark para la regresión LASSO.

```
1 # =====
2 # 1. Importación de librerías
3 # =====
4 from pyspark.sql.types import StructType, StructField, DoubleType
5 from pyspark.ml.feature import VectorAssembler, MinMaxScaler,
   MinMaxScalerModel
```

```

6 import pyspark.pandas as pd
7
8 # Configuración de visualización
9 pd.set_option('display.max_rows', 20)
10
11 # =====
12 # 2. Carga del dataset y definición del esquema
13 # =====
14 dfRaw1 = (
15     spark.read.format("csv")
16     .option("header", "true")
17     .option("delimiter", ",")
18     .option("encoding", "ISO-8859-1")
19     .schema(
20         StructType(
21             [
22                 StructField("CO2(ppm)_1", DoubleType(), True),
23                 StructField("CO(ppm)_1", DoubleType(), True),
24                 StructField("VOC(ppm)_1", DoubleType(), True),
25                 StructField("PM10(mg/m3)_1", DoubleType(), True),
26                 StructField("IAQmax_1", DoubleType(), True),
27             ]
28         )
29     )
30     .load("dbfs:///FileStore/_ciencia/datos_1_iaq_sinNO2_balanced.csv")
31 )
32
33 # =====
34 # 3. Generación del vector de características
35 # =====
36 dfRaw2 = VectorAssembler(
37     inputCols=["CO2(ppm)_1", "CO(ppm)_1", "VOC(ppm)_1", "PM10(mg/m3)_1"],
38     outputCol="features"
39 ).transform(dfRaw1)

```

```

40
41 dfRaw2.show()
42
43 # =====
44 # 4. Normalización de los datos
45 # =====
46 escalador = MinMaxScaler(inputCol="features",
47                           outputCol="features_normalizados").fit(dfRaw2)
48
49 dfRaw3 = escalador.transform(dfRaw2)
50
51 # =====
52 # 5. Almacenamiento del dataset procesado
53 # =====
54 dfRaw3.write.format("parquet").mode("overwrite").option("compression",
55                  "snappy").save(
56                  "dbfs:///FileStore/_ciencia/output/dataset_C1IAQ_sinNO2_balanced/"
57 )

```

Código C.2. Calibración del modelo de regresión LASSO con PySpark.

```
1 # =====
2 # 1. Librerías
3 # =====
4 from pyspark.sql.types import StructType, StructField, StringType,
    IntegerType, DoubleType
5 import pyspark.pandas as pd
6 from pyspark.ml.feature import MinMaxScalerModel
7 from pyspark.ml.regression import LinearRegression
8 from pyspark.ml.evaluation import RegressionEvaluator
9 from pyspark.ml.tuning import ParamGridBuilder, CrossValidator
10
11 # =====
12 # 2. Configuración de visualización
13 # =====
14 pd.set_option('display.max_rows', 20)
15
16 # =====
17 # 3. Lectura del dataset normalizado
18 # =====
19 dfDataset = spark.read.format("parquet").load(
20 "dbfs:///FileStore/_ciencia/output/dataset_IAQ/"
21 )
22 dfDataset.show()
23
24 # =====
25 # 4. Definición del modelo de regresión LASSO
26 # =====
27 algoritmo = LinearRegression(
28     featuresCol="features_normalizados",
29     labelCol="IAQmax"
30 )
31
32 # =====
```

```

33 # 5. Definición del evaluador de desempeño
34 # =====
35 evaluador = RegressionEvaluator(
36     labelCol="IAQmax",
37     predictionCol="prediction"
38 )
39
40 # =====
41 # 6. Configuración de la malla de parámetros
42 # =====
43 mallaDeParametros = ParamGridBuilder()\
44     .addGrid(algoritmo.elasticNetParam, [0.001, 0.005, 0.03, 0.17, 1.0])\
45     .addGrid(algoritmo.regParam, [0.001, 0.01, 0.1, 10, 100])\
46     .build()
47
48 # =====
49 # 7. Definición de la validación cruzada
50 # =====
51 validacionCruzada = CrossValidator(
52     estimator=algoritmo,
53     estimatorParamMaps=mallaDeParametros,
54     evaluator=evaluador,
55     numFolds=5
56 )
57
58 # =====
59 # 8. Entrenamiento y selección del mejor modelo
60 # =====
61 modelosGenerados = validacionCruzada.fit(dfDataset)
62 modelo = modelosGenerados.bestModel
63
64 # =====
65 # 9. Generación de predicciones
66 # =====

```

```

67 dfPrediccion = modelo.transform(dfDataset)
68 # =====
69 # 10. Evaluación del modelo
70 # =====
71 evaluadorMSE = RegressionEvaluator(
72     labelCol="IAQmax",
73     predictionCol="prediction",
74     metricName="mse"
75 )
76 mse = evaluadorMSE.evaluate(dfPrediccion)
77 evaluadorR2 = RegressionEvaluator(
78     labelCol="IAQmax",
79     predictionCol="prediction",
80     metricName="r2"
81 )
82 r2 = evaluadorR2.evaluate(dfPrediccion)
83 print(f"MSE: {mse}")
84 print(f"R2: {r2}")

```

Anexo D

Algoritmo random forest

El contenido de este anexo corresponde a la implementación del modelo *random forest*, que también se desarrolló con el framework de *PySpark* y la plataforma de cómputo *databricks.com*.

Código D.1. Entrenamiento y calibración del modelo random forest con PySpark.

```
1 # =====
2 # 1. Librerías
3 # =====
4 from pyspark.sql.types import StructType, StructField, StringType,
   IntegerType, DoubleType
5 import pyspark.pandas as pd
6 from pyspark.ml.feature import VectorAssembler, StringIndexer,
   StringIndexerModel
7 from pyspark.ml.classification import RandomForestClassifier,
   RandomForestClassificationModel
8 from pyspark.ml.evaluation import MulticlassClassificationEvaluator,
   RegressionEvaluator
9 from pyspark.ml.tuning import ParamGridBuilder, CrossValidator
10 from pyspark.sql.functions import col
11
12 # =====
13 # 2. Configuración de visualización
14 # =====
15 pd.set_option('display.max_rows', 20)
16
17 # =====
18 # 3. Lectura del dataset balanceado
19 # =====
20 dfDataset = spark.read.format("parquet").load(
```

```

21 "dbfs:///FileStore/_ciencia/output/dataset_C1IAQ_sinN02_balanced/"
22 )
23 dfDataset = dfDataset.withColumn("IAQmax_1", col("IAQmax_1").cast("int"))
24 dfDataset.show()
25
26 # =====
27 # 4. Definición del modelo Random Forest
28 # =====
29 algoritmo = RandomForestClassifier(
30     labelCol="IAQmax_1",
31     featuresCol="features"
32 )
33
34 # =====
35 # 5. Definición del evaluador de desempeño
36 # =====
37 evaluador = MulticlassClassificationEvaluator(
38     labelCol="IAQmax_1",
39     predictionCol="prediction",
40     metricName="accuracy"
41 )
42
43 # =====
44 # 6. Configuración de la malla de parámetros
45 # =====
46 mallaDeParametros = ParamGridBuilder()\
47     .addGrid(algoritmo.numTrees, [5, 10, 20])\
48     .addGrid(algoritmo.maxDepth, [5, 10, 15])\
49     .addGrid(algoritmo.impurity, ["entropy"])\
50     .build()
51
52 # =====
53 # 7. Validación cruzada
54 # =====

```

```

55 validacionCruzada = CrossValidator(
56     estimator=algoritmo,
57     estimatorParamMaps=mallaDeParametros,
58     evaluator=evaluador,
59     numFolds=5
60 )
61
62 # =====
63 # 8. Entrenamiento y selección del mejor modelo
64 # =====
65 modelosGenerados = validacionCruzada.fit(dfDataset)
66 modelo = modelosGenerados.bestModel
67
68 # =====
69 # 9. Validación del modelo
70 # =====
71 dfPrediccion = modelo.transform(dfDataset)
72 dfPrediccion.select("IAQmax_1", "prediction").show(40, False)
73
74 # =====
75 # 10. Evaluación del modelo
76 # =====
77 precision = evaluador.evaluate(dfPrediccion)
78 print(f"Precisión (Accuracy): {precision}")
79
80 mae_evaluator = RegressionEvaluator(
81     labelCol="IAQmax_1",
82     predictionCol="prediction",
83     metricName="mae"
84 )
85 mae = mae_evaluator.evaluate(dfPrediccion)
86 print(f"MAE: {mae}")
87
88 r2_evaluator = RegressionEvaluator(

```

```

89     labelCol="IAQmax_1",
90     predictionCol="prediction",
91     metricName="r2"
92 )
93 r2_value = r2_evaluator.evaluate(dfPrediccion)
94 print(f"R2: {r2_value}")
95
96 # =====
97 # 11. Parámetros del modelo
98 # =====
99 print("Número de árboles:", modelo.getNumTrees)
100 print("Profundidad máxima:", modelo.getDefault("maxDepth"))
101
102 # =====
103 # 12. Almacenamiento del modelo entrenado
104 # =====
105 modelo.write().overwrite().save(
106     "dbfs:///FileStore/_ciencia/output/modelo_random_forest_C1/"
107 )
108
109 # =====
110 # 13. Recuperación del modelo desde almacenamiento
111 # =====
112 modelo = RandomForestClassificationModel.load(
113     "dbfs:///FileStore/_ciencia/output/modelo_random_forest_C1/"
114 )
115
116 # =====
117 # 14. Generación final de predicciones
118 # =====
119 dfPrediccion = modelo.transform(dfDataset)

```

Anexo E

Algoritmo de redes neuronales

En este apartado se detallan los programas empleados para la calibración y ajuste del modelo de redes neuronales utilizado en la predicción del Índice de Calidad de Aire (IAQ). Los procedimientos mostrados a continuación corresponden a la implementación descrita en el **capítulo 5**, y abarcan desde la preparación del conjunto de datos hasta la optimización de los principales hiperparámetros del modelo. El mismo proceso se realiza para los datos simulados de los cuatro ambientes.

Las etapas consideradas son las siguientes:

1. Preparación de los datos: generación del dataset normalizado y del escalador.
2. Calibración del *learning rate*.
3. Calibración del algoritmo de *backpropagation*.
4. Calibración del tamaño de lote (*batch size*).
5. Calibración del número de capas y validación cruzada.

A continuación, se presenta el código correspondiente a cada uno de estos procesos para uno de los ambientes.

Código E.1. Preparación del dataset y normalización

```
1 # =====
2 # 1. Importación de librerías
3 # =====
4 import pandas as pd
5 from sklearn.model_selection import train_test_split
6 from sklearn.preprocessing import MinMaxScaler
7 import pickle
8
```

```

9  # =====
10 # 2. Lectura del archivo fuente
11 # =====
12 dfpRawx = pd.read_csv(
13     "/content/datos_4_iaq_sinNO2_balanced.csv",
14     delimiter=";",
15     dtype={
16         "CO2(ppm)_4": float,
17         "CO(ppm)_4": float,
18         "VOC(ppm)_4": float,
19         "PM10(mg/m3)_4": float,
20         "IAQmax_4": float
21     }
22 )
23
24 # =====
25 # 3. Separación de variables de entrada y salida
26 # =====
27 X = dfpRawx.drop(["IAQmax_4"], axis=1)
28 y = dfpRawx[["IAQmax_4"]]
29
30 # División en conjuntos de entrenamiento y prueba
31 X_train, X_test, y_train, y_test = train_test_split(
32     X, y, test_size=0.2, random_state=42
33 )
34
35 dfpRaw = pd.concat([X_train, y_train], axis=1)
36 dfptest = pd.concat([X_test, y_test], axis=1)
37
38 # =====
39 # 4. Normalización de los datos
40 # =====
41 # Definición de columnas a normalizar
42 columnasFeatures = [

```

```

43     "CO2(ppm)_4",
44     "CO(ppm)_4",
45     "VOC(ppm)_4",
46     "PM10(mg/m3)_4"
47 ]
48
49 # Creación de nombres para las columnas normalizadas
50 columnasNormalizadas = [
51     "feature_" + c + "_normalized" for c in columnasFeatures
52 ]
53
54 # Creación y ajuste del escalador
55 escalador = MinMaxScaler().fit(dfpRaw[columnasFeatures])
56
57 # Aplicación de la normalización
58 dfpRaw[columnasNormalizadas] =
59     escalador.transform(dfpRaw[columnasFeatures])
60 dfptest[columnasNormalizadas] =
61     escalador.transform(dfptest[columnasFeatures])
62
63 # =====
64 # 5. Almacenamiento de datasets normalizados
65 # =====
66 dfpRaw.to_hdf("/content/dataset_C4_iaq_train_balanced.h5", key="dataset",
67             mode="w")
68 dfptest.to_hdf("/content/dataset_C4_iaq_test_balanced.h5", key="dataset",
69              mode="w")
70
71 # =====
72 # 6. Almacenamiento del objeto escalador
73 # =====
74 with open("/content/escalador_C4_iaq_sinNO2_balanced.pkl", "wb") as
75     archivo:
76     pickle.dump(escalador, archivo)

```

```

72
73 # =====
74 # 7. Verificación del escalador guardado
75 # =====
76 with open("/content/escalador_C4_iaq_sinNO2_balanced.pkl", "rb") as
    archivo:
77     scaler = pickle.load(archivo)
78
79 print("data_min_", scaler.data_min_)
80 print("data_max_", scaler.data_max_)
81 print("feature_range", scaler.feature_range)
82 print("n_features_in_", scaler.n_features_in_)

```

Código E.2. Calibración del learning rate

```

1 # =====
2 # 1. Verificación de GPU disponible
3 # =====
4 import tensorflow as tf
5
6 # Comprobación de disponibilidad de GPU para entrenamiento
7 print(tf.test.gpu_device_name())
8
9 # =====
10 # 2. Importación de librerías
11 # =====
12 import pandas as pd
13 import pickle
14 from keras.models import Sequential
15 from keras.layers import Dense
16 from tensorflow.keras.optimizers import Adam
17
18 # =====
19 # 3. Lectura del dataset
20 # =====

```

```

21 dfpDataset = pd.read_hdf("/content// dataset_C4_iaq_train_balanced .
    .h5", key="dataset")
22
23 # =====
24 # 4. Definición de matriz de features
25 # =====
26 dfpFeatures = dfpDataset[
27     [
28         "feature_CO2(ppm)_1_normalized",
29         "feature_CO2(ppm)_2_normalized",
30         "feature_CO2(ppm)_3_normalized",
31         "feature_CO2(ppm)_4_normalized",
32         "feature_CO(ppm)_1_normalized",
33         "feature_CO(ppm)_2_normalized",
34         "feature_CO(ppm)_3_normalized",
35         "feature_CO(ppm)_4_normalized",
36         "feature_VOC(ppm)_1_normalized",
37         "feature_VOC(ppm)_2_normalized",
38         "feature_VOC(ppm)_3_normalized",
39         "feature_VOC(ppm)_4_normalized",
40         "feature_PM10(mg/m3)_1_normalized",
41         "feature_PM10(mg/m3)_2_normalized",
42         "feature_PM10(mg/m3)_3_normalized",
43         "feature_PM10(mg/m3)_4_normalized",
44     ]
45 ]
46
47 # =====
48 # 5. Definición de matriz de labels
49 # =====
50 dfpLabels = dfpDataset[
51     [
52         "IAQmax_1",
53         "IAQmax_2",

```

```

54         "IAQmax_3",
55         "IAQmax_4",
56     ]
57 ]
58
59 # =====
60 # 6. Calibracion del learning rate
61 # =====
62 # Se probaran distintos valores de tasa de aprendizaje
63 array_lr = [0.001, 0.01, 0.1, 1.0]
64
65 # Estructura para almacenar las métricas de cada prueba
66 columnas_descripcion = {}
67
68 # =====
69 # 7. Iteración de calibraciones
70 # =====
71 for lr in array_lr:
72     print("lr =", lr)
73
74     # Definición del modelo
75     modelo = Sequential()
76     modelo.add(Dense(4, input_shape=(16,)))
77
78     # Compilación del modelo con la tasa de aprendizaje actual
79     modelo.compile(Adam(learning_rate=lr), "mean_absolute_error")
80
81     # Entrenamiento del modelo
82     descripcion_de_entrenamiento = modelo.fit(
83         dfpFeatures, dfpLabels, epochs=100, verbose=0
84     )
85
86     # Registro de la evolución del error
87     columnas_descripcion[str(lr)] =

```

```

88         descripcion_de_entrenamiento.history["loss"]
89
90 # =====
91 # 8. Análisis de resultados
92 # =====
93
94 dfDescripcion = pd.DataFrame(columnas_descripcion)
95
96 # Graficar la evolución del error para cada lr
97 dfDescripcion.plot()
98
99 # Comparar solo las tasas de aprendizaje mas relevantes
100 dfDescripcion[["0.01", "0.1"]].plot()
101
102 # Selección del learning rate optimo
103 lr_optimo = 0.1

```

Código E.3. Calibración del backpropagation

```

1 # =====
2 # ETAPA 1: Verificacion de GPU
3 # =====
4
5 import tensorflow as tf
6 print(tf.test.gpu_device_name())
7
8 # =====
9 # ETAPA 2: Importación de librerias
10 # =====
11
12 import pandas as pd
13 import pickle
14 from keras.models import Sequential
15 from keras.layers import Dense
16 from tensorflow.keras.optimizers import Adam
17 from google.colab import drive

```

```

18
19 drive.mount('/content/drive')
20
21 # =====
22 # ETAPA 3: Lectura del dataset
23 # =====
24
25 dfpDataset = pd.read_hdf("/content/dataset_C4_iaq_train_balanced.h5",
26                           key="dataset")
27 dfpDataset
28 # =====
29 # ETAPA 4: Definición de la matriz de features
30 # =====
31
32 dfpFeatures = dfpDataset[
33     [
34         'feature_CO2(ppm)_1_normalized',
35         'feature_CO2(ppm)_2_normalized',
36         'feature_CO2(ppm)_3_normalized',
37         'feature_CO2(ppm)_4_normalized',
38         'feature_CO(ppm)_1_normalized',
39         'feature_CO(ppm)_2_normalized',
40         'feature_CO(ppm)_3_normalized',
41         'feature_CO(ppm)_4_normalized',
42         'feature_VOC(ppm)_1_normalized',
43         'feature_VOC(ppm)_2_normalized',
44         'feature_VOC(ppm)_3_normalized',
45         'feature_VOC(ppm)_4_normalized',
46         'feature_PM10(mg/m3)_1_normalized',
47         'feature_PM10(mg/m3)_2_normalized',
48         'feature_PM10(mg/m3)_3_normalized',
49         'feature_PM10(mg/m3)_4_normalized'
50     ]

```

```

51 ]
52 dfpFeatures
53
54 # =====
55 # ETAPA 5: Definición de la matriz de labels
56 # =====
57
58 dfpLabels = dfpDataset[
59     [
60         'IAQmax_1',
61         'IAQmax_2',
62         'IAQmax_3',
63         'IAQmax_4'
64     ]
65 ]
66 dfpLabels
67
68 # =====
69 # ETAPA 6: Definición de la arquitectura de la red neuronal
70 # =====
71
72 def obtenerModelo():
73     modelo = Sequential()
74     modelo.add(Dense(4, input_shape=(16, )))
75     return modelo
76
77 # =====
78 # ETAPA 7: Definición del learning rate optimo
79 # =====
80
81 lr_optimo = 0.01
82
83 # =====
84 # ETAPA 8: Calibración del algoritmo de backpropagation

```

```

85 # =====
86
87 from tensorflow.keras.optimizers import Adam, Adagrad, RMSprop
88
89 array_algoritmo = [Adam, Adagrad, RMSprop]
90 columnas_descripcion = {}
91
92 for algoritmo in array_algoritmo:
93     print("algoritmo = " + str(algoritmo))
94     modelo = obtenerModelo()
95     modelo.compile(algoritmo(learning_rate=lr_optimo),
96                   "mean_absolute_error")
97     descripcion_de_entrenamiento = modelo.fit(dfpFeatures, dfpLabels,
98       epochs=100, verbose=0)
99     columnas_descripcion[str(algoritmo)] =
100       descripcion_de_entrenamiento.history["loss"]
101
102 dfDescripcion = pd.DataFrame(columnas_descripcion)
103
104 # =====
105 # ETAPA 9: Selección del mejor algoritmo
106 # =====
107
108 dfDescripcion.plot()
109 dfDescripcion.iloc[50:100].plot()
110
111 algoritmo_optimo = Adam

```

Código E.4. Calibración del batch size

```

1
2 # =====
3 # ETAPA 1: Verificación de GPU
4 # =====
5
6 import tensorflow as tf

```

```

7 print(tf.test.gpu_device_name())
8
9 # =====
10 # ETAPA 2: Importación de librerías
11 # =====
12
13 import pandas as pd
14 import pickle
15 from keras.models import Sequential
16 from keras.layers import Dense
17 from tensorflow.keras.optimizers import Adam
18
19 # =====
20 # ETAPA 3: Lectura del dataset
21 # =====
22
23 dfpDataset = pd.read_hdf("/content/dataset_C4_iaq_train_balanced.h5",
24                          key="dataset")
25 dfpDataset
26 # =====
27 # ETAPA 4: Definición de la matriz de features
28 # =====
29
30 dfpFeatures = dfpDataset[
31     [
32         'feature_CO2(ppm)_1_normalized',
33         'feature_CO2(ppm)_2_normalized',
34         'feature_CO2(ppm)_3_normalized',
35         'feature_CO2(ppm)_4_normalized',
36         'feature_CO(ppm)_1_normalized',
37         'feature_CO(ppm)_2_normalized',
38         'feature_CO(ppm)_3_normalized',
39         'feature_CO(ppm)_4_normalized',

```

```

40         'feature_VOC(ppm)_1_normalized',
41         'feature_VOC(ppm)_2_normalized',
42         'feature_VOC(ppm)_3_normalized',
43         'feature_VOC(ppm)_4_normalized',
44         'feature_PM10(mg/m3)_1_normalized',
45         'feature_PM10(mg/m3)_2_normalized',
46         'feature_PM10(mg/m3)_3_normalized',
47         'feature_PM10(mg/m3)_4_normalized'
48     ]
49 ]
50 dfpFeatures
51
52 # =====
53 # ETAPA 5: Definición de la matriz de labels
54 # =====
55
56 dfpLabels = dfpDataset[
57     [
58         'IAQmax_1',
59         'IAQmax_2',
60         'IAQmax_3',
61         'IAQmax_4'
62     ]
63 ]
64 dfpLabels
65
66 # =====
67 # ETAPA 6: Definición de la arquitectura de la red neuronal
68 # =====
69
70 def obtenerModelo():
71     modelo = Sequential()
72     modelo.add(Dense(4, input_shape=(16, )))
73     return modelo

```

```

74
75 # =====
76 # ETAPA 7: Definición del learning rate óptimo
77 # =====
78
79 lr_optimo = 0.01
80
81 # =====
82 # ETAPA 8: Definición del algoritmo óptimo
83 # =====
84
85 algoritmo_optimo = Adam
86
87 # =====
88 # ETAPA 9: Calibración del batch size
89 # =====
90
91 array_batch = [32, 64, 128, 256]
92 columnas_descripcion = {}
93
94 for batch in array_batch:
95     print("batch = " + str(batch))
96     modelo = obtenerModelo()
97     modelo.compile(algoritmo_optimo(learning_rate=lr_optimo),
98                   "mean_absolute_error")
99     descripcion_de_entrenamiento = modelo.fit(
100         dfpFeatures,
101         dfpLabels,
102         epochs=100,
103         batch_size=batch,
104         verbose=0
105     )
106     columnas_descripcion[str(batch)] =
107         descripcion_de_entrenamiento.history["loss"]

```

```

106
107 dfDescripcion = pd.DataFrame(columnas_descripcion)
108
109 # =====
110 # ETAPA 10: Selección del mejor batch size
111 # =====
112
113 dfDescripcion.plot()
114 dfDescripcion.iloc[0:20].plot()
115 dfDescripcion[dfDescripcion.columns.difference(["1024",
116         "512"])] .iloc[0:5].plot()
116 batch_optimo = 32

```

Código E.5. Calibración de Número de Capas y Cross Validation

```

1 # =====
2 # 1. Verificación de GPU y configuración inicial
3 # =====
4 import tensorflow
5 print(tensorflow.test.gpu_device_name())
6
7 # =====
8 # 2. Importación de librerías
9 # =====
10 import pandas as pd
11 import numpy as np
12 import pickle
13 from keras.models import Sequential
14 from keras.layers import Dense
15 from tensorflow.keras.optimizers import SGD, Adam, Adagrad, RMSprop
16
17 # =====
18 # 3. Lectura del dataset
19 # =====
20 dfpDataset = pd.read_hdf("/content/dataset_C4_iaq_train_balanced.h5",

```

```

    key="dataset")

21
22 # =====
23 # 4. Definición de matrices de características y etiquetas
24 # =====
25 dfpFeatures = dfpDataset[
26     ['feature_CO2(ppm)_1_normalized', 'feature_CO2(ppm)_2_normalized',
27     'feature_CO2(ppm)_3_normalized', 'feature_CO2(ppm)_4_normalized',
28     'feature_CO(ppm)_1_normalized', 'feature_CO(ppm)_2_normalized',
29     'feature_CO(ppm)_3_normalized', 'feature_CO(ppm)_4_normalized',
30     'feature_VOC(ppm)_1_normalized', 'feature_VOC(ppm)_2_normalized',
31     'feature_VOC(ppm)_3_normalized', 'feature_VOC(ppm)_4_normalized',
32     'feature_PM10(mg/m3)_1_normalized',
33     'feature_PM10(mg/m3)_2_normalized',
34     'feature_PM10(mg/m3)_3_normalized',
35     'feature_PM10(mg/m3)_4_normalized']
36 ]
37 dfpLabels = dfpDataset[['IAQmax_1', 'IAQmax_2', 'IAQmax_3', 'IAQmax_4']]
38
39 # =====
40 # 5. Parámetros iniciales óptimos
41 # =====
42 lr_optimo = 0.1
43 algoritmo_optimo = Adam
44 batch_optimo = 32
45
46 # =====
47 # 6. Calibración de la cantidad de perceptrones por capa
48 # =====
49 array_perceptrones = [8, 16, 32, 64, 128]
50 columnas_descripcion = {}
51
52 for perceptrones in array_perceptrones:
53     print("perceptrones =", perceptrones)

```

```

52     modelo = Sequential()
53     modelo.add(Dense(perceptrones, input_shape=(16,), activation="relu"))
54     modelo.add(Dense(4))
55     modelo.compile(algoritmo_optimo(learning_rate=lr_optimo),
56                     "mean_absolute_error")
57     descripcion_de_entrenamiento = modelo.fit(dfpFeatures, dfpLabels,
58         epochs=100, batch_size=batch_optimo, verbose=0)
59     columnas_descripcion[str(perceptrones)] =
60         descripcion_de_entrenamiento.history["loss"]
61
62     dfDescripcion = pd.DataFrame(columnas_descripcion)
63     dfDescripcion.plot()
64
65     # =====
66     # 7. Validación cruzada
67     # =====
68
69     from sklearn.model_selection import KFold
70     validacionesCruzadas = KFold(n_splits=8)
71     array_modelos = []
72     arrayErrores = []
73
74     x = dfpFeatures.values
75     y = dfpLabels.values
76
77     i = 0
78     for train_idx, test_idx in validacionesCruzadas.split(x, y):
79         print("Validación:", i)
80         x_train, y_train = x[train_idx], y[train_idx]
81         x_test, y_test = x[test_idx], y[test_idx]
82         modelo = Sequential()
83         modelo.add(Dense(64, input_shape=(16,), activation="relu"))
84         modelo.add(Dense(32, activation="relu"))
85         modelo.add(Dense(4))
86         modelo.compile(Adam(learning_rate=0.01), "mean_absolute_error")

```

```

83     modelo.fit(x_train, y_train, epochs=100, batch_size=64, verbose=0)
84     error = modelo.evaluate(x_test, y_test)
85     array_errores.append(error)
86     array_modelos.append(modelo)
87     i += 1
88
89 print(array_errores)
90 print("Índice del mejor modelo:", np.argmin(array_errores))
91
92
93
94 # =====
95 # 8. Evaluación final y guardado del modelo
96 # =====
97 from sklearn.metrics import r2_score
98 modelo = array_modelos[np.argmin(array_errores)]
99 y_pred = modelo.predict(dfpFeatures)
100 print("R2:", r2_score(dfpLabels, y_pred))
101 modelo.save("modelo_balanced.h5")

```

Anexo F

Código fuente de Raspberry pi

El siguiente código fue usado en la etapa de procesamiento de los datos recibidos por la microcomputadora Raspberry Pi, según se describe en las secciones 4.2 y 4.3, para filtrar, almacenar en base de datos y enviar al dashboard en la nube (ThingsBoard) todas las mediciones. Además se emplea el modelo entrenado de redes neuronales para predecir el IAQ.

Código F.1. Código fuente usado en la Raspberry pi

```
1
2 import serial
3 import time
4 import json
5 import numpy as np
6 from collections import deque
7 import csv
8 import os
9 from sklearn.preprocessing import MinMaxScaler
10 import paho.mqtt.client as paho
11
12
13 # =====
14 # 1. Configuración Filtro Media Movil
15 # =====
16
17 VENTANA=5
18
19 buffer_pm10 = deque(maxlen=VENTANA)
20 buffer_co2 = deque(maxlen=VENTANA)
21 buffer_co = deque(maxlen=3)
22 buffer_voc = deque(maxlen=VENTANA)
23 buffer_temp = deque(maxlen=VENTANA)
```

```

24 buffer_hum = deque(maxlen=VENTANA)
25
26
27 # =====
28 # 2. Configuración MQTT / ThingsBoard
29 # =====
30
31 ACCESS_TOKEN = 'ne42VS9RRtVv9ucVZDKe'
32 broker = '162.55.63.249'
33 port = 1883
34
35 # --- Inicialización MQTT ---
36 def on_publish(client, userdata, mid, properties=None, reason_code=None):
37     print("Datos publicados en ThingsBoard")
38
39 client = paho.Client(
40     client_id="Sensores",
41     protocol=paho.MQTTv311,
42     callback_api_version=paho.CallbackAPIVersion.VERSION2
43 )
44 client.on_publish = on_publish
45 client.username_pw_set(ACCESS_TOKEN)
46 client.connect(broker, port, keepalive=60)
47
48 # =====
49 # 3. Inicializar puerto serial
50 # =====
51 try:
52     ser = serial.Serial('/dev/ttyUSB0', 9600)
53     ser.reset_input_buffer()
54     print("Puerto serial abierto correctamente")
55 except Exception as e:
56     print(f"Error al abrir el puerto serial: {e}")
57

```

```

58 # =====
59 # 4. Cargar modelo RNN y escalador
60 # =====
61 def softsign(x):
62     return x / (1 + np.abs(x))
63
64 pesos = np.load("modelo_numpy_softsign_1_capa.npz")
65
66 def modelo_numpy(x):
67     x = np.dot(x, pesos["W0"]) + pesos["b0"]
68     x = softsign(x)
69     x = np.dot(x, pesos["W1"]) + pesos["b1"]
70     return x
71
72 def cargar_escalador_C1_iaq():
73     data_min = [340.005, 0.0, 0.0, 0.0]
74     data_max = [1499.868, 49.995, 2.999, 0.4]
75     scaler = MinMaxScaler()
76     scaler.feature_range = (0, 1)
77     scaler.data_min_ = np.array(data_min)
78     scaler.data_max_ = np.array(data_max)
79     scaler.data_range_ = scaler.data_max_ - scaler.data_min_
80     scaler.scale_ = 1.0 / scaler.data_range_
81     scaler.min_ = -scaler.data_min_ * scaler.scale_
82     scaler.n_features_in_ = 4
83     return scaler
84
85 scaler = cargar_escalador_C1_iaq()
86
87 # =====
88 # 5. Preparar archivo CSV
89 # =====
90 csv_file = "registro_datos.csv"
91 es_nuevo = not os.path.exists(csv_file)

```

```

92
93 with open(csv_file, mode='a', newline='') as archivo_csv:
94     writer = csv.writer(archivo_csv)
95     if es_nuevo:
96         writer.writerow(['timestamp', 'pm10', 'temperature', 'humedad',
97                             'voc', 'co', 'co2', 'iaq', 'ens_iaq'])
98
99 # =====
100 # 6. Bucle principal
101 # =====
102 flag = True
103 while True:
104     if flag:
105         for _ in range(10):
106             ser.readline() # limpiamos el buffer
107             flag = False
108
109     linea = ser.readline().decode('utf-8').strip()
110     datos = linea.split(",")
111
112     if len(datos) == 7:
113         try:
114             # Lectura original
115             pm10_raw = float(datos[0])
116             temperature_raw = int(datos[1])
117             humedad_raw = int(datos[2])
118             voc_raw = float(datos[3])
119             co_raw = float(datos[4])
120             co2_raw = int(datos[5])
121             ens_iaq = int(datos[6])
122
123             # Guardar en buffers
124             buffer_pm10.append(pm10_raw)

```

```

125     buffer_temp.append(temperature_raw)
126     buffer_hum.append(humedad_raw)
127     buffer_voc.append(voc_raw)
128     buffer_co.append(co_raw)
129     buffer_co2.append(co2_raw)
130
131
132     # Media móvil
133     pm10 = round(np.mean(buffer_pm10), 2)
134     temperature = round(np.mean(buffer_temp), 1)
135     humedad = round(np.mean(buffer_hum), 1)
136     voc = round(np.mean(buffer_voc), 2)
137     co = round(np.mean(buffer_co), 2)
138     co2 = round(np.mean(buffer_co2), 1)
139
140
141     print(f"PM10: {pm10}, Temp: {temperature}, Hum: {humedad},
142           VOC: {voc}, CO: {co}, CO2: {co2}, IAQ: {ens_iaq}")
143
144     # Predicción IAQ
145     entrada_escalada = scaler.transform([[co2, co, voc, pm10]])
146     entrada = np.array(entrada_escalada, dtype=np.float32)
147     salida = modelo_numpy(entrada)
148     iaq = float(np.clip(salida[0][0], 0, 100)) # IAQ entre 0 y 100
149     iaq=round(iaq)
150     print("Predicción IAQ:", iaq)
151
152     # Enviar IAQ por serial (opcional)
153     try:
154         ser.write((str(iaq) + "\n").encode())
155     except Exception as e:
156         print("Error al enviar IAQ:", e)
157
158     # Publicar en ThingsBoard

```

```
158     payload = {
159         "pm10": pm10,
160         "temperature": temperature,
161         "humedad": humedad,
162         "voc": voc,
163         "co": co,
164         "co2": co2,
165         "iaq": iaq
166     }
167     client.publish("v1/devices/me/telemetry", json.dumps(payload))
168
169     # Guardar en CSV
170     with open(csv_file, mode='a', newline='') as archivo_csv:
171         writer = csv.writer(archivo_csv)
172         timestamp = time.strftime("%Y-%m-%d %H:%M:%S")
173         writer.writerow([timestamp, pm10, temperature, humedad,
174                         voc, co, co2, iaq, ens_iaq])
175
176     except Exception as e:
177         print("Error procesando datos:", e)
178
179     time.sleep(1)
```

Anexo G

Trabajos futuros

El sistema desarrollado tiene un gran rango de mejora y ampliación, lo que posibilita diversas líneas de investigación y desarrollo futuro:

- **Escalabilidad hacia un nivel superior de IoT:** Actualmente, el sistema está implementado bajo una estructura de nivel 3 de IoT, pero una proyección hacia los niveles superiores es totalmente factible. por ejemplo, se puede avanzar hacia el nivel 4 de IoT, en el que múltiples nodos ubicados en diferentes ambientes pueden transmitir los datos sentidos hacia un mismo Gateway, centralizando la información para su posterior envío al servidor remoto. Una implementación así permitirá monitorear edificios completos.
- **Optimización de la arquitectura de hardware:** Una posible línea de trabajo sería reducir la complejidad del sistema mediante la eliminación del Raspberry pi, de tal manera que el ESP32 asuma todas las funciones de procesamiento, almacenamiento y comunicación en la nube. Aunque este enfoque simplifica el hardware y reduce los costos, también limita la capacidad de procesamiento, velocidad de análisis y almacenamiento de datos, por lo que sería útil hacer un análisis de rendimiento para validar su viabilidad en comparación con el diseño actual.
- **Autocalibración mediante reentrenamiento del algoritmo de predicción:** El algoritmo de predicción podría mejorar aún más a través de un reentrenamiento periódico, usando las nuevas mediciones que se almacenan en la base de datos. Esta funcionalidad permitiría lograr un proceso de autocalibración, aumentando la exactitud de las predicciones y dotar al sistema de la capacidad de adaptarse dinámicamente a las condiciones del entorno.

- **Ampliación de la base de datos con mediciones reales:** Ya que la base de datos principal utilizada en este proyecto de investigación, fue generada mediante simulación, es importante realizar mediciones en entornos reales, por lo que un trabajo futuro debe ser desplegar el sistema durante un tiempo prolongado en un ambiente objetivo. Esta ampliación será útil para afianzar la robustez del sistema y del algoritmo de predicción.



Nomenclatura

IoT Internet of Things

USEPA United States Environmental Protection Agency

WHO World Health Organization

IAQ Indoor Air Quality

TTN The Things Network

IQR Interquartile Range

FCN Fully Convolutional Network

CNN Convolutional Neural Networks

FNN Feedforward Neural Networks

RNN Recurrent Neural Networks

PWM Pulse Width Modulation

DNN Deep Neural Networks

MAA Mean Age of Air

MAE Mean Absolute Error

TVOC Total Volatile Organic Compounds

ADFIST Adaptive Dynamic Fuzzy Inference System Tree

PM Partículas de Materia

NRMSE Normalized Root Mean Square Error

RMSE Root Mean Square Error

UC Microcontroller

LSTM Long Short-Term Memory

GRU Gated Recurrent Units

ANN Artificial Neural Networks

ML Machine Learning

PLSR Partial Least Squares Regression

MLR Multiple Linear Regression

LASSO Least Absolute Shrinkage and Selection Operator

IAP Indoor Air Pollutants

IEEE Institute of Electrical and Electronics Engineers

SMTP Simple Mail Transfer Protocol