



Pontificia Universidad  
**JAVERIANA**  
Cali

**DESARROLLO DE UN SISTEMA DE MONITOREO Y ANÁLISIS PARA LA CALIDAD DEL  
AIRE USANDO INTERNET DE LAS COSAS (IoT) Y TECNOLOGÍAS DE MACHINE  
LEARNING (ML) EN TRES PUNTOS GEOGRÁFICOS DE LA CIUDAD DE BOGOTÁ.**

*Álvaro Ariza, código 8979840*

*Carlos Sierra, código 8976092*

*Proyecto Aplicado para optar al título de  
Magister en Ciencia de Datos*

Director  
Eugenio Tamura

FACULTAD DE INGENIERÍA Y CIENCIAS  
MAestrÍA EN CIENCIA DE DATOS  
SANTIAGO DE CALI, JULIO 14 DE 2025

## TABLA DE CONTENIDO

1.	DEFINICIÓN DEL PROBLEMA .....	3
1.1	PLANTEAMIENTO DEL PROBLEMA .....	3
1.2	FORMULACIÓN DEL PROBLEMA.....	5
2.	OBJETIVOS DEL PROYECTO.....	6
2.1	OBJETIVO GENERAL.....	6
2.2	OBJETIVOS ESPECÍFICOS .....	6
3.	MARCO TEÓRICO Y ANTECEDENTES.....	7
3.1	MARCO TEÓRICO.....	7
3.1.1	Calidad Del Aire Y Contaminantes Principales .....	7
3.1.2	Material Particulado.....	7
3.1.3	INTERNET DE LAS COSAS (IoT).....	8
3.1.4	MACHINE LEARNING .....	9
3.1.5	Modelos LSTM para predicción de series temporales .....	9
3.1.6	Hiperparámetros con Keras Tuner .....	9
3.1.7	Optimizador Adam .....	10
3.1.8	Índice de Calidad del Aire (AQI).....	10
3.2	ANTECEDENTES .....	10
3.2.1	PREDICTION AND ANALYSIS OF AIR QUALITY INDEX USING MACHINE LEARNING ALGORITHMS .....	10
3.2.2	THE PREDICTION OF QUALITY OF THE AIR USING SUPERVISED LEARNING .....	11
3.2.3	A MACHINE LEARNING MODEL FOR AIR QUALITY PREDICTION FOR SMART CITIES.....	11
3.2.4	PROTOTIPO DE UN MODELO DE MACHINE LEARNING PARA LA PREDICCIÓN DE PARTÍCULAS DE CONTAMINACIÓN ATMOSFÉRICA FINAS, EN LA LOCALIDAD DE KENNEDY EN LA CIUDAD DE BOGOTÁ.....	12
3.2.5	PREDICTING AIR QUALITY INDEX USING ATTENTION HYBRID DEEP LEARNING AND QUANTUM-INSPIRED PARTICLE SWARM OPTIMIZATION. ....	12
3.2.6	APPLICATION OF CNN-LSTM ALGORITHM FOR PM2.5 CONCENTRATION FORECASTING IN THE BEIJING-TIANJIN-HEBEI METROPOLITAN AREA.....	12
3.2.7	HYBRID DEEP LEARNING FOR AIR QUALITY PREDICTION: A MULTI-OUTPUT, ATTENTION-BASED APPROACH FOR POLLUTANT AND AQI CLASSIFICATION.....	13
3.2.8	PREDICCIÓN DE LA CALIDAD DEL AIRE DE LA CIUDAD DE MEDELLÍN Y SU ÁREA METROPOLITANA MEDIANTE EL USO DE REDES NEURONALES RECURRENTE. ....	13

3.2.9 TÉCNICAS DE APRENDIZAJE AUTOMÁTICO PARA EL ANÁLISIS DE DATOS DE CALIDAD DEL AIRE.	14
3.2.10 APLICACIÓN DE MODELO XGB PARA PREDICCIÓN DE CALIDAD DE AIRE, BOGOTÁ -COLOMBIA.	14
4. DESARROLLO DEL PROYECTO: ESTRUCTURA, PREPARACIÓN Y DISEÑO DEL TRABAJO	16
4.1 Adquisición de datos	16
4.2 Transmisión y almacenamiento en la nube	16
4.3 Análisis y predicción con Python y modelos LSTM	16
4.4 Cálculo del Índice de Calidad del Aire (AQI)	17
4.5 Visualización y sistema de alertas	18
5. ACTIVIDADES Y RESULTADOS OBTENIDOS OBJETIVO ESPECÍFICO 1	20
5.1 Selección de sensores adecuados	20
5.2 Selección de puntos de monitoreo	24
5.3 Ensamblaje e instalación del hardware	25
5.4 Resultados obtenidos	26
6. ACTIVIDADES Y RESULTADOS OBTENIDOS OBJETIVO ESPECÍFICO 2	27
6.1 Diseño de la infraestructura IoT	27
6.2 Plataforma ThingSpeak como núcleo del sistema	29
6.3 Programación del envío de datos	32
6.4 Visualización y consulta en línea	32
6.5 Interacción con otros servicios:	35
6.6 Resultados obtenidos	37
7. ACTIVIDADES Y RESULTADOS OBTENIDOS OBJETIVO ESPECÍFICO 3	38
7.1 Preparación del entorno de análisis	38
7.2 Estructura de análisis por contaminante y medidor	39
7.3 Exploración de datos (EDA)	50
7.4 Selección del modelo predictivo con LSTM	56
7.4.1 Long Short-Term Memory (LSTM)	56
7.5 Elección de los Hiperparámetros	57
7.6 Interpolación lineal para rellenar datos faltantes	57
7.7 Resultados por medidor y contaminante	58
7.8 Conclusión del objetivo	60
8. ACTIVIDADES Y RESULTADOS OBTENIDOS OBJETIVO ESPECÍFICO 4	61

8.1 Desarrollo de la interfaz de usuario .....	61
8.2 Visualización de datos y conexión con ThingSpeak.....	64
8.3 Integración de resultados del modelo predictivo LSTM.....	64
8.4 Alertas automáticas con Telegram.....	64
8.5 Conclusión del objetivo .....	67
9. ACTIVIDADES Y RESULTADOS OBTENIDOS OBJETIVO ESPECÍFICO 5 .....	72
9.1 Evaluación del funcionamiento integral del sistema .....	72
9.2 Desempeño de los modelos predictivos .....	72
9.3 Aplicabilidad y escalabilidad.....	73
9.4 Comparación con trabajos previos.....	73
10. CONCLUSIONES Y TRABAJOS FUTUROS.....	75
10.1 Conclusiones.....	75
10.2 Trabajos Futuros.....	75
11. REFERENCIAS BIBLIOGRÁFICAS .....	77

## LISTA DE FIGURAS

Figura 1. Topología general de la solución desarrollada.....	19
Figura 2. Sensor de monóxido de carbono. ....	23
Figura 3. Sensor de PM2.5.....	23
Figura 4. Puntos de monitoreo.....	26
Figura 5. Infraestructura IoT.....	27
Figura 6. Código ESP32 – Adquisición de variables y conectividad.....	28
Figura 7. Código ESP32 –Configuración de conectividad WIFI, lectura sensores y envío servidor. .....	28
Figura 8. Montaje del prototipo para captura y envío de datos.....	29
Figura 9. Montaje del prototipo para captura y envío de datos.....	29
Figura 10. Montaje del prototipo para captura y envío de datos.....	29
Figura 11. Operación ThingSpeak.....	30
Figura 12. Configuración de canales ThingSpeak.....	30
Figura 13. Gráficas generadas en ThingSpeak.....	31
Figura 14. Gráficas AQI generadas en ThingSpeak.....	31
Figura 15. Código en ThinkSpeak y Matlab para procesamiento de datos recibidos.....	32
Figura 16. Gráficas generadas en ThingSpeak, medidor 1.....	33
Figura 17. Gráficas generadas en ThingSpeak, medidor 2.....	33
Figura 18. Gráficas generadas en ThingSpeak, medidor 3.....	33
Figura 19. Valores de ThingSpeak publicados.....	34
Figura 20. Configuración del canal en ThingSpeak. ....	34
Figura 21. Configuración de valores AQI CO.....	35
Figura 22. Configuración de valores AQI PM2.5.....	35
Figura 23. Configuración de Google Sites.....	36
Figura 24. Configuración de Google Sites – Diseño de interfaces de usuario.....	36
Figura 25. Uso de Telegram para envío de Notificaciones.....	37
Figura 26. Entorno google colab.....	38
Figura 27. Bibliotecas utilizadas.....	39
Figura 28. Lectura de los datos.....	39
Figura 29. Procesamiento y agrupación por promedio horario.....	40
Figura 30. Ventanas supervisadas.....	41
Figura 31. Entrenamiento y prueba.....	42
Figura 32. Modelo.....	42
Figura 33. Evaluación de métricas.....	43
Figura 34. Fórmula raíz del error cuadrático medio (RMSE).....	43
Figura 35. Fórmula coeficiente de determinación ( $R^2$ ).....	44
Figura 36. Épocas y evaluación de métricas.....	45
Figura 37. Promedio móvil.....	45
Figura 38. Break-points detallados (EPA).....	46
Figura 39. Break-points detallados (EPA).....	46
Figura 40. Barra de color AQI (con recomendación).....	47
Figura 41. Barra de color AQI (con recomendación).....	47

Figura 42. Visualizaciones. ....	47
Figura 43. Gráfico real vs predicción.....	48
Figura 44. Pérdida. ....	49
Figura 45. Dispersión.....	49
Figura 46. Datos reales vs predicción.....	50
Figura 47. Estructura de los datos.....	51
Figura 48. Datos faltantes. ....	51
Figura 49. Histogramas y boxplots. ....	53
Figura 50. Descomposición de la serie.....	54
Figura 51. Curvas de tendencia semanal. ....	55
Figura 52. Promedio mensual. ....	55
Figura 53. Esquema interno de una celda LSTM.....	56
Figura 54. Hiperparámetros. ....	57
Figura 55. Interpolación lineal entre dos puntos. ....	58
Figura 56. ecuación de la interpolación lineal. ....	58
Figura 57. Gráficas en tiempo real provenientes de ThingSpeak, embebidas mediante iframes. 61	
Figura 58. Gráficas en tiempo real. ....	62
Figura 59. Gráficas AQI.....	63
Figura 60. Gráficas AQI colores. ....	63
Figura 61. Alertas automáticas con Telegram.....	65
Figura 62. Generación de Alertas.....	66
Figura 63. Grupo de alertas.....	67

## LISTA DE TABLAS

Tabla 1. Resumen artículos estado del arte .....	15
Tabla 2. Herramientas utilizadas .....	18
Tabla 3. Comparativo de los sensores evaluados para medir monóxido de carbono .....	20
Tabla 4. Comparativo de los sensores para medir material particulado PM2.5.....	21
Tabla 5. Puntos de monitoreo Bogotá .....	25
Tabla 6. Rangos y colores aplicados al AQI (CO) .....	67
Tabla 7. Rangos y colores aplicados al AQI (PM2.5) .....	68
Tabla 8. Herramientas utilizadas y su justificación técnica.....	70

## LISTA DE ANEXOS

Anexo 1: Página web Google Sites .....	81
Anexo 2: Datos Zona UCompensar Teusaquillo (Medidor 1) - ThingSpeak IoT .....	81
Anexo 3: Datos Zona UManuela (Medidor 2) - ThingSpeak IoT.....	81
Anexo 4: Datos Zona centro (Medidor 3) - ThingSpeak IoT .....	81
Anexo 5: Modelo CO medidor 1.....	81
Anexo 6: Hiperparámetros CO medidor 1.....	81
Anexo 7: EDA CO medidor 1.....	81
Anexo 8: Modelo CO medidor 2.....	81
Anexo 9: Hiperparámetros CO medidor 2.....	81
Anexo 10: EDA CO medidor 2.....	81
Anexo 11: Modelo CO medidor 3.....	81
Anexo 12: Hiperparámetros CO medidor 3.....	81
Anexo 13: EDA CO medidor 3.....	82
Anexo 14: Modelo PM2.5 medidor 1 .....	82
Anexo 15: Hiperparámetros PM2.5 medidor 1.....	82
Anexo 16: EDA PM2.5 medidor 1.....	82
Anexo 17: Modelo PM2.5 medidor 2.....	82
Anexo 18: Hiperparámetros PM2.5 medidor 2.....	82
Anexo 19: EDA PM2.5 medidor 2.....	82
Anexo 20: Modelo PM2.5 medidor 3.....	82
Anexo 21: Hiperparámetros PM2.5 medidor 3.....	82
Anexo 22: EDA PM2.5 medidor 3.....	82

## INTRODUCCIÓN

La calidad del aire en entornos urbanos es uno de los desafíos más críticos en materia de salud pública y sostenibilidad ambiental, especialmente en ciudades densamente pobladas como Bogotá. Factores como el tránsito vehicular, las actividades industriales y las condiciones meteorológicas contribuyen a la acumulación de contaminantes atmosféricos que afectan significativamente la salud humana. Entre los contaminantes más perjudiciales se destacan el monóxido de carbono (CO) y el material particulado fino (PM<sub>2.5</sub>), cuyos niveles elevados están estrechamente asociados con enfermedades respiratorias y cardiovasculares.

En respuesta a esta problemática, el presente proyecto propone el desarrollo de un sistema inteligente de monitoreo, análisis y predicción de la calidad del aire, utilizando tecnologías emergentes como el Internet de las Cosas (IoT) y Machine Learning (ML). La solución se basa en la implementación de estaciones de medición en tres puntos estratégicos de Bogotá, donde se integran sensores de bajo costo: el SEN0564 para la detección de CO y el PMS5003 para la medición de PM<sub>2.5</sub>, conectados a microcontroladores ESP32 programados con Arduino IDE.

Los datos capturados en tiempo real cada 5 minutos son transmitidos a la plataforma en la nube ThingSpeak, donde se almacenan y procesan mediante scripts en MATLAB Analysis. Además de visualizarse en gráficos actualizados en tiempo real, estos datos son descargados periódicamente en formato CSV y analizados mediante Python en Google Colab, utilizando herramientas como Pandas, NumPy, Scikit-learn, TensorFlow y Keras. Para cada contaminante y cada punto de monitoreo, se diseñó un modelo específico basado en redes neuronales LSTM (Long Short-Term Memory), seleccionadas por su capacidad para modelar series temporales no lineales y predecir tendencias futuras de contaminación.

El ajuste de hiperparámetros fue realizado mediante Keras Tuner con Random Search, obteniendo configuraciones apropiadas que lograron resultados sobresalientes con coeficientes  $R^2$  superiores al 0.98 y errores cuadráticos medios (RMSE) significativamente bajos. La información predictiva se complementa con un sistema de visualización accesible desde una interfaz web en Google Sites, que permite consultar gráficas, predicciones y condiciones actuales del aire de forma sencilla.

Adicionalmente, se implementó un sistema de alertas automáticas utilizando Telegram, el cual se configura directamente desde ThingSpeak mediante las herramientas TimeControl, React y ThingHTTP. En lugar de utilizar el AQI como criterio, las alertas se activan cuando se superan umbrales preestablecidos (por ejemplo, 4.5 ppm para CO o 9.1  $\mu\text{g}/\text{m}^3$  para PM<sub>2.5</sub>), notificando a los usuarios a través de un grupo de Telegram en tiempo real. Este enfoque mejora la capacidad de respuesta ante eventos críticos y permite a los ciudadanos y autoridades tomar decisiones informadas de forma oportuna.

En conjunto, el sistema desarrollado representa una arquitectura modular, de bajo costo,

escalable y replicable, capaz de monitorear y predecir la calidad del aire, integrando herramientas de programación, analítica de datos, visualización y comunicación. Su implementación no solo fortalece la cobertura del monitoreo ambiental en Bogotá, sino que también sienta las bases para una red inteligente y comunitaria de vigilancia de la calidad del aire.

## **1. DEFINICIÓN DEL PROBLEMA**

### **1.1 PLANTEAMIENTO DEL PROBLEMA**

La ciudad de Bogotá enfrenta una problemática de contaminación del aire, la cual representa un riesgo para la salud pública y el medio ambiente. Los altos niveles de concentración de material particulado PM2.5 y PM10 en la ciudad son motivo de preocupación. Aunque existen estaciones de monitoreo de la calidad del aire, la falta de herramientas efectivas para realizar análisis predictivos de la contaminación limita la capacidad de abordar este problema de manera proactiva. Además, el sistema integrado de transporte público de Bogotá es responsable de la generación de material particulado, lo que representa un desafío significativo para mantener la calidad del aire en la ciudad.

En este contexto, es crucial desarrollar un sistema de monitoreo basado en IoT y ML que permita medir y predecir niveles de CO y material particulado PM2.5 en puntos estratégicos de Bogotá. El objetivo es proporcionar información sobre la contaminación del aire, lo que contribuirá a la toma de decisiones informadas para mejorar la calidad de vida de la población y proteger el entorno ambiental en la ciudad.

La calidad del aire es un factor fundamental para determinar el nivel de vida y el bienestar de los habitantes de una ciudad. En Bogotá, la contaminación del aire se ha convertido en uno de los principales desafíos ambientales, afectando tanto la salud pública como los ecosistemas locales. Es vital implementar medidas efectivas para reducir y controlar esta problemática, promoviendo políticas públicas orientadas hacia la sostenibilidad urbana, la renovación del parque automotor, el uso eficiente de combustibles limpios y la sensibilización ciudadana [1].

Entre las consecuencias más significativas de la contaminación del aire en Bogotá se encuentran, Problemas respiratorios generado por el material particulado (especialmente el PM10 y el PM2.5) y otros contaminantes como el ozono troposférico pueden provocar enfermedades respiratorias graves, incluyendo asma, bronquitis crónica y Enfermedad Pulmonar Obstructiva Crónica (EPOC). Estas partículas penetran profundamente en los pulmones, irritan las vías respiratorias y aumentan la susceptibilidad a infecciones respiratorias [38].

También efectos cardiovasculares, dado por la exposición prolongada a altos niveles de contaminación del aire está asociada con un aumento en la incidencia de enfermedades cardiovasculares, tales como hipertensión arterial, infartos cardíacos y accidentes cerebrovasculares. Las partículas finas, especialmente aquellas menores a 0.1 micras, pueden

ingresar al torrente sanguíneo y causar inflamaciones sistémicas que afectan al corazón y los vasos sanguíneos. A esto adicionalmente se suma la Contaminación del aire y cambio climático por la exposición a emisiones de gases de efecto invernadero, aunque no son el foco principal del monitoreo local, están relacionadas con fuentes comunes como los procesos de combustión. Estas emisiones contribuyen al calentamiento global, alterando patrones climáticos y aumentando la frecuencia de eventos extremos como sequías e inundaciones [1].

De acuerdo con Rojas [1] las principales fuentes de contaminación del aire en Bogotá son:

- Vehículos con motores diésel: responsables del 90% de las emisiones de material particulado. Los buses y camiones antiguos son los mayores responsables.
- Motocicletas de dos tiempos: contribuyen significativamente a las emisiones de compuestos orgánicos volátiles (COV) y material particulado.
- Vehículos a gasolina y gas natural: aunque su aporte al material particulado es menor, son importantes emisores de COV y óxidos de nitrógeno, promotores del ozono troposférico.
- Industria: representa cerca del 60% de las emisiones de material particulado. Sectores como ladrilleras, fundiciones y tintorerías destacan por su impacto.
- Otras fuentes: incluyen rellenos sanitarios, estaciones de servicio y fuentes residenciales, que contribuyen a la contaminación con metano, compuestos orgánicos volátiles (VOCs) y partículas que se vuelven a poner en suspensión.

Algunas cifras generales sobre la contaminación ambiental en Bogotá que son relevantes hasta el momento:

- Contaminación del aire: según el Instituto de Hidrología, Meteorología y Estudios Ambientales de Colombia (IDEAM), Bogotá ha enfrentado problemas de calidad del aire en los últimos años. En 2020, la ciudad superó los límites máximos permitidos para partículas PM2.5 (partículas finas suspendidas en el aire) en varias estaciones de monitoreo [1].
- Emisiones de gases de efecto invernadero: Bogotá es responsable de una cantidad significativa de emisiones de gases de efecto invernadero en Colombia. Según el Sistema de Estimación de Emisiones de Colombia (SEEC), en 2015, las emisiones totales de Dióxido de Carbono (CO<sub>2</sub>) de la ciudad alcanzaron aproximadamente 18,5 millones de toneladas métricas.
- Contaminación del agua: La contaminación del agua en Bogotá es un problema que afecta tanto los cuerpos de agua superficiales como el suministro de agua potable. Se han

registrado problemas de calidad del agua debido a la descarga de aguas residuales sin tratar y la contaminación por productos químicos y desechos industriales.

- Generación de residuos: Bogotá produce una gran cantidad de residuos sólidos. Según la Unidad Administrativa Especial de Servicios Públicos (UAESP), la ciudad generó alrededor de 6.000 toneladas de residuos diarios en 2020.

## **1.2 FORMULACIÓN DEL PROBLEMA**

¿Cómo desarrollar e implementar un sistema de monitoreo de calidad del aire de bajo costo, basado en IoT y modelos de predicción con aprendizaje automático, que permita medir, predecir y visualizar en tiempo real los niveles de CO y PM<sub>2,5</sub> en tres puntos geográficos de la ciudad de Bogotá, facilitando la toma de decisiones ambientales mediante el uso de plataformas accesibles y alertas automatizadas?

## 2. OBJETIVOS DEL PROYECTO

### 2.1 OBJETIVO GENERAL

Desarrollar un sistema de monitoreo y análisis de la calidad del aire basado en Internet de las Cosas (IoT) y tecnologías de Machine Learning (ML) para medir y predecir niveles de monóxido de carbono (CO) y material particulado 2.5 micras (PM2.5) en tres puntos geográficos de Bogotá, con el fin de proporcionar información oportuna sobre la contaminación del aire.

### 2.2 OBJETIVOS ESPECÍFICOS

- Seleccionar y configurar sensores de calidad del aire basados en IoT para medir los niveles de CO y PM2.5 en los puntos de monitoreo, identificando y seleccionando tres puntos geográficos en Bogotá para la instalación de sensores de calidad del aire en el sistema de monitoreo propuesto.
- Desarrollar la infraestructura de IoT necesaria para recopilar, transmitir y almacenar los datos de los sensores en tiempo real, implementando un sistema de recopilación de datos de CO y PM2.5.
- Aplicar técnicas de ML para analizar los datos recopilados y desarrollar un modelo predictivo para los niveles de CO y PM2.5 en los puntos de monitoreo, entrenando y ajustando los modelos de ML utilizando los datos recopilados, con el objetivo de mejorar su precisión en la predicción de los niveles de contaminación del aire.
- Diseñar una interfaz de usuario que muestre los resultados del monitoreo, visualice los niveles de CO y PM2.5 y proporcione información sobre la calidad del aire en los puntos de monitoreo, estableciendo límites aceptables de CO y PM2.5 basados en regulaciones y estándares ambientales, e implementar alertas y notificaciones automáticas cuando los niveles superen dichos límites.
- Realizar una evaluación del sistema de monitoreo implementado, identificando lecciones aprendidas y recomendaciones para mejorar su eficacia y utilidad en la información sobre la contaminación del aire en Bogotá.

### 3. MARCO TEÓRICO Y ANTECEDENTES

#### 3.1 MARCO TEÓRICO

En el contexto del monitoreo ambiental, la combinación de tecnologías emergentes como el Internet de las Cosas (IoT), el aprendizaje automático (ML) y la computación en la nube ha permitido desarrollar sistemas de bajo costo y alta escalabilidad para la predicción de la calidad del aire. En este proyecto, se integraron diferentes conceptos y técnicas que permitieron abordar el problema de forma integral.

##### 3.1.1 Calidad del Aire y Contaminantes Principales

Es la medida de la pureza y composición del aire que se respira en un determinado entorno. Es un indicador importante de la salud ambiental y puede tener impactos significativos en la salud humana, la flora, la fauna y los ecosistemas en general [2].

La calidad del aire se evalúa mediante la medición y el análisis de diversos contaminantes atmosféricos, como gases, partículas sólidas y líquidas suspendidas en el aire, compuestos orgánicos volátiles y otros contaminantes químicos. Estos contaminantes pueden ser de origen natural, como las emisiones volcánicas o el polen, o de origen antropogénico, derivados de actividades humanas, como la quema de combustibles fósiles, la industria, el transporte, la agricultura y otras fuentes de emisiones. Los principales contaminantes atmosféricos que se monitorean para evaluar la calidad del aire incluyen [3], [4]:

- Dióxido de Azufre (SO<sub>2</sub>): Emitido principalmente por la quema de combustibles fósiles, especialmente en plantas de energía y procesos industriales.
- Óxidos de Nitrógeno (NO<sub>x</sub>): Producidos durante la combustión de combustibles fósiles en vehículos, centrales eléctricas y procesos industriales.
- Monóxido de Carbono (CO): Emitido principalmente por la combustión incompleta de combustibles fósiles en vehículos y sistemas de calefacción.
- Partículas Suspendidas (PM): Pequeñas partículas sólidas y líquidas en el aire, que pueden ser de origen natural (polvo, polen) o generado por actividades humanas (emisiones industriales, escape de vehículos).
- Compuestos Orgánicos Volátiles (COV): Compuestos químicos orgánicos que se evaporan fácilmente y se emiten por procesos industriales, combustión de combustibles y disolventes.
- Ozono troposférico (O<sub>3</sub>): Un contaminante secundario formado por la reacción de los óxidos de nitrógeno y los compuestos orgánicos volátiles en presencia de luz solar.

##### 3.1.2 Material Particulado

Se refiere a partículas sólidas o líquidas suspendidas en el aire que pueden tener diferentes

tamaños y composiciones. Estas partículas son una forma común de contaminación del aire y pueden tener un impacto significativo en la calidad del aire y en la salud humana.

El tamaño de las partículas es un factor importante, ya que determina su capacidad para penetrar en el sistema respiratorio humano. La clasificación más comúnmente utilizada para el material particulado se basa en el diámetro aerodinámico de las partículas. Algunos de los tipos más comunes de material particulado son los siguientes [4], [5]:

- PM10: Se refiere a partículas con un diámetro aerodinámico menor o igual a 10 micrómetros ( $\mu\text{m}$ ). Estas partículas son lo suficientemente pequeñas como para ser inhaladas y pueden llegar hasta los bronquios y los pulmones.
- PM2.5: Se refiere a partículas con un diámetro aerodinámico menor o igual a 2.5  $\mu\text{m}$ . Son aún más pequeñas que las PM10 y pueden penetrar aún más profundamente en el sistema respiratorio, alcanzando los alvéolos pulmonares.
- PM1: Se refiere a partículas con un diámetro aerodinámico menor o igual a 1  $\mu\text{m}$ . Son aún más pequeñas y pueden llegar a los tejidos más profundos de los pulmones, así como al torrente sanguíneo.

La exposición crónica a altos niveles de material particulado se ha asociado con problemas respiratorios, cardiovasculares y otros efectos adversos para la salud. Por lo tanto, la monitorización y la reducción de las emisiones de material particulado son aspectos clave en la gestión de la calidad del aire y la protección del medio ambiente y la salud pública [6].

### **3.1.3 INTERNET DE LAS COSAS (IoT)**

Se refiere a la red de objetos físicos interconectados que están equipados con sensores, software y tecnología de red, que les permite recopilar e intercambiar datos. Estos objetos, también conocidos como "dispositivos inteligentes" o "dispositivos conectados", pueden ser cualquier cosa, desde electrodomésticos, vehículos, dispositivos médicos, sensores ambientales, hasta maquinaria industrial.

La idea fundamental detrás del IoT es permitir la comunicación y la interacción entre estos dispositivos, así como con aplicaciones y servicios en la nube. Esto crea un ecosistema en el que los datos recopilados por los dispositivos se pueden analizar, procesar y utilizar para tomar decisiones inteligentes, automatizar tareas, mejorar la eficiencia y generar nuevos servicios.

Varios protocolos de comunicación permiten la interconexión y el intercambio de datos entre dispositivos IoT. Estos protocolos pueden variar según el tipo de aplicación, los requisitos de conectividad y los recursos disponibles en los dispositivos. A continuación, se exponen algunos de los protocolos más comunes utilizados en IoT [7]:

- Message Queuing Telemetry Transport (MQTT): Es un protocolo ligero de mensajería que se basa en el modelo de publicación-suscripción. Es adecuado para aplicaciones con dispositivos de baja potencia y ancho de banda limitado. MQTT es ampliamente utilizado en sistemas de monitorización y control, así como en aplicaciones de hogares inteligentes.

- Hypertext Transfer Protocol (HTTP): Aunque es ampliamente utilizado en aplicaciones web, también se utiliza en IoT para la comunicación entre dispositivos y servicios en la nube. HTTP proporciona una interfaz flexible y familiar para acceder a recursos y servicios a través de la web; utiliza el modelo de solicitud-respuesta.
- Long Range Wide Area Network (LoRaWAN): Es un protocolo de comunicación inalámbrica de largo alcance utilizado en redes de área amplia de bajo consumo (LPWAN). LoRaWAN es adecuado para aplicaciones IoT que requieren una cobertura extendida y una larga duración de la batería, como la monitorización agrícola y la gestión de activos.

### **3.1.4 MACHINE LEARNING**

El Machine Learning (ML), o Aprendizaje Automático en español, es una rama de la inteligencia artificial que se enfoca en desarrollar algoritmos y modelos capaces de aprender y tomar decisiones automáticamente sin ser explícitamente programados. Estos algoritmos permiten a las computadoras analizar grandes cantidades de datos, identificar patrones y realizar predicciones o tomar decisiones basadas en esos patrones.

El ML se basa en la idea de que las máquinas pueden aprender a través de la experiencia y mejorar su rendimiento en tareas específicas a medida que se les proporciona más información. Para lograr esto, se utilizan diferentes enfoques de aprendizaje, como el aprendizaje supervisado, no supervisado y por refuerzo.

El ML tiene una amplia gama de aplicaciones en diversas industrias, como reconocimiento de voz, visión por computadora, análisis de datos, recomendación de productos, detección de fraudes, entre muchas otras. Es una disciplina en constante crecimiento y desarrollo con un gran potencial para mejorar la eficiencia y la toma de decisiones en numerosos campos [8].

### **3.1.5 Modelos LSTM para predicción de series temporales**

Los modelos LSTM (Long Short-Term Memory) son una variante avanzada de las redes neuronales recurrentes (RNN), especialmente diseñadas para modelar secuencias y series temporales. Estos modelos incorporan celdas de memoria que permiten conservar información relevante de pasos anteriores y manejar dependencias de largo plazo. En este proyecto, las LSTM fueron implementadas en Google Colab utilizando Python y bibliotecas como Keras, TensorFlow, Pandas, y Matplotlib [9].

Cada medidor y contaminante cuenta con un modelo independiente de predicción, entrenado con datos reales recolectados por los sensores en campo. Para el CO y el PM2.5 se realizaron análisis exploratorios (EDA), ventanas de supervisión, divisiones de entrenamiento/prueba, evaluación con métricas como RMSE y  $R^2$ , y guardado de modelos finales.

### **3.1.6 Hiperparámetros con Keras Tuner**

La técnica de Random Search mediante Keras Tuner, es una herramienta que permite explorar de forma sistemática combinaciones de parámetros como número de unidades por capa, tasa de dropout, y tamaño de las ventanas de entrada. Este procedimiento automatizado ayudó a

identificar las arquitecturas para cada modelo, maximizando la precisión y minimizando el error de predicción [10].

### **3.1.7 Optimizador Adam**

Para el entrenamiento de los modelos se empleó el optimizador Adam (Adaptive Moment Estimation), debido a su eficiencia en problemas no convexos, su capacidad de adaptarse dinámicamente a cada parámetro y su excelente desempeño en tareas con datos ruidosos y no estacionarios como las series de calidad del aire. Aunque existen otros optimizadores como RMSprop, SGD o Adagrad, Adam combina las ventajas de estos y mostró consistentemente mejor desempeño durante la validación de los modelos [11].

### **3.1.8 Índice de Calidad del Aire (AQI)**

El AQI, definido por la Agencia de Protección Ambiental de Estados Unidos (EPA, por sus siglas en inglés), es un sistema estandarizado que transforma las concentraciones de contaminantes en una escala de 0 a 500, categorizando los valores en seis niveles de riesgo para la salud (de “Bueno” a “Peligroso”) [12].

## **3.2 ANTECEDENTES**

A continuación, se presentan los principales trabajos de investigación revisados que soportan la relevancia científica y metodológica del proyecto. Cada uno incluye su resumen, enfoque y fuente:

### **3.2.1 PREDICTION AND ANALYSIS OF AIR QUALITY INDEX USING MACHINE LEARNING ALGORITHMS**

En [13] proponen un modelo para predecir y analizar el *Índice de Calidad del Aire (AQI)* utilizando algoritmos de aprendizaje automático. Los autores recopilan datos sobre contaminantes y utilizan diversas técnicas de regresión, incluyendo Regresión de Árbol de Decisión, Regresión Lineal, Regresión de Bosque Aleatorio y Regresión de Vector de Soporte, para predecir los valores de AQI.

El artículo discute el uso de algoritmos de aprendizaje automático para predecir el índice de calidad del aire (AQI) basado en el conjunto de datos de contaminación de los Estados Unidos. Se utilizaron cuatro modelos, Regresión de Árbol de Decisión (DTR), Regresión de Bosque Aleatorio (RFR), Regresión Lineal (LR) y Regresión de Vector de Soporte (SVR), para predecir los valores de AQI. Los resultados mostraron que RFR funcionó mejor con los valores más pequeños de MAE, MSE y RMSE. El estudio destaca la importancia de una predicción precisa de la calidad del aire para la salud pública y la respuesta de emergencia.

Por lo tanto, el objetivo principal del documento es proponer un modelo de predicción de AQI utilizando algoritmos de aprendizaje automático y comparar el rendimiento de diferentes técnicas de regresión. Además, se busca resaltar la importancia de la predicción precisa de la calidad del aire para la salud pública y la respuesta de emergencia.

### **3.2.2 THE PREDICTION OF QUALITY OF THE AIR USING SUPERVISED LEARNING**

En [14] discuten el uso de técnicas de aprendizaje automático supervisado para predecir la calidad del aire en áreas urbanas. El objetivo principal es proporcionar una predicción precisa y en tiempo real de la calidad del aire para mitigar los efectos nocivos de la contaminación en la salud humana y el medio ambiente.

El documento propone un marco que involucra análisis exploratorio de datos, manipulación de datos, construcción de un modelo de clasificación y predicción de resultados. Se comparan diferentes algoritmos de aprendizaje automático y se evalúa su rendimiento utilizando métricas como precisión, recuperación (recall) y exactitud.

El modelo de árbol de decisión se utiliza para construir el modelo de predicción, que predice el contaminante presente en el ambiente, así como sus causas y fuentes. El sistema puede ser modificado para su uso potencial en un entorno de inteligencia artificial, y el mecanismo también puede ser automatizado mediante la visualización de los resultados de predicción en un sitio web o aplicación de escritorio.

### **3.2.3 A MACHINE LEARNING MODEL FOR AIR QUALITY PREDICTION FOR SMART CITIES**

En [15] proponen un modelo de aprendizaje automático para predecir el AQI en ciudades inteligentes, utilizando redes neuronales y máquinas de soporte vectorial. Los datos se recopilan del Consejo Central de Control de la Contaminación en India, y el modelo se prueba en el AQI de Delhi. Los resultados muestran una mejora en la precisión de la predicción y sugieren que el modelo también se puede utilizar en otras ciudades inteligentes.

El documento también incluye una revisión de la investigación relacionada con redes neuronales artificiales, monitoreo de AQI y máquinas de soporte vectorial.

El documento concluye que se ha propuesto un modelo de aprendizaje automático para predecir el índice de calidad del aire en ciudades inteligentes utilizando redes neuronales y máquinas de soporte vectorial. El modelo se ha probado en los datos de calidad del aire de Delhi y ha logrado una precisión del 91.62% para redes neuronales y del 97.3% para máquinas de soporte vectorial. Se sugiere que los algoritmos de minería de datos podrían utilizarse en trabajos futuros para mejorar aún más la precisión de la predicción.

En general, el documento destaca la importancia de utilizar modelos de aprendizaje automático para predecir la calidad del aire en ciudades inteligentes y mejorar la salud pública.

### **3.2.4. PROTOTIPO DE UN MODELO DE MACHINE LEARNING PARA LA PREDICCIÓN DE PARTÍCULAS DE CONTAMINACIÓN ATMOSFÉRICA FINAS, EN LA LOCALIDAD DE KENNEDY EN LA CIUDAD DE BOGOTÁ**

En [4] desarrollan un prototipo de modelo de aprendizaje automático para predecir partículas finas de contaminación atmosférica en la localidad de Kennedy en Bogotá. El objetivo del documento es discutir el uso de la inteligencia artificial, específicamente las redes neuronales artificiales, para predecir y mitigar los niveles de contaminación del aire. El documento proporciona antecedentes sobre la contaminación del aire y la importancia de monitorear y mitigar sus efectos en la salud humana.

También se discuten los objetivos y la justificación de la investigación, así como la metodología y los resultados del modelo de prototipo. El documento también discute diferentes tipos de contaminantes atmosféricos y cómo se detectan utilizando sensores electroquímicos. El objetivo principal del documento es presentar un modelo de red neuronal artificial para predecir la calidad del aire en una región específica y destacar la importancia de abordar la contaminación del aire como una prioridad global.

### **3.2.5 PREDICTING AIR QUALITY INDEX USING ATTENTION HYBRID DEEP LEARNING AND QUANTUM-INSPIRED PARTICLE SWARM OPTIMIZATION.**

Por su parte en [16], se confirma la efectividad del modelo CNN-LSTM en la predicción precisa de concentraciones de PM<sub>2.5</sub> en estaciones de monitoreo urbano. El estudio utiliza datos de temperatura, velocidad del viento y presión atmosférica como variables de entrada, demostrando que factores meteorológicos tienen alta correlación con las variaciones de PM<sub>2.5</sub>.

El modelo alcanza un R<sup>2</sup> de 0.91 y un RMSE de 8.216 µg/m<sup>3</sup>, superando con claridad modelos CNN o LSTM aislados. La investigación refuerza la idea de que la integración de redes convolucionales y recurrentes mejora la captura de dependencias espaciotemporales en fenómenos de contaminación atmosférica. Además, destaca la importancia del preprocesamiento de datos para garantizar precisión y confiabilidad en la predicción [16].

### **3.2.6 APPLICATION OF CNN-LSTM ALGORITHM FOR PM2.5 CONCENTRATION FORECASTING IN THE BEIJING-TIANJIN-HEBEI METROPOLITAN AREA.**

Se desarrolla un modelo de predicción de PM<sub>2.5</sub> basado en la combinación de redes neuronales convolucionales (CNN) con redes LSTM, entrenado con datos horarios de contaminación, meteorología y vapor de agua precipitante (PWV) de diez ciudades.

El modelo CNN-LSTM muestra mejoras significativas frente a modelos tradicionales como BPNN y LSTM puro, con reducciones en RMSE de hasta 25%. La ventaja de este enfoque radica en su capacidad para capturar simultáneamente patrones espaciales y temporales, una característica esencial en contextos urbanos donde la dinámica del aire puede cambiar drásticamente.

Este enfoque respalda la necesidad de modelos híbridos en proyectos como el presente, que buscan interpretar múltiples variables ambientales simultáneamente para ofrecer alertas tempranas y apoyar decisiones gubernamentales [17].

### **3.2.7 HYBRID DEEP LEARNING FOR AIR QUALITY PREDICTION: A MULTI-OUTPUT, ATTENTION-BASED APPROACH FOR POLLUTANT AND AQI CLASSIFICATION.**

En "Hybrid Deep Learning for Air Quality Prediction " se propone un modelo híbrido de aprendizaje profundo para predecir el Índice de Calidad del Aire (AQI) y clasificar niveles de severidad de contaminantes individuales como PM<sub>2.5</sub> y CO. El enfoque combina redes neuronales LSTM bidireccionales y mecanismos de atención múltiple (multi-head attention) para capturar relaciones temporales y espaciales entre los contaminantes, logrando una alta precisión predictiva y robustez ante variaciones en los datos.

El artículo resalta cómo el uso de atención permite mejorar la identificación de patrones críticos en los datos de sensores, lo cual es especialmente valioso para sistemas en tiempo real. El estudio valida su modelo con métricas como F1-score, curvas ROC y matrices de confusión, mostrando su potencial para ser utilizado en sistemas de monitoreo ambiental en entornos urbanos con múltiples fuentes de contaminación [18].

### **3.2.8 PREDICCIÓN DE LA CALIDAD DEL AIRE DE LA CIUDAD DE MEDELLÍN Y SU ÁREA METROPOLITANA MEDIANTE EL USO DE REDES NEURONALES RECURRENTE.**

Los autores plantean el uso de una red neuronal recurrente LSTM para predecir el índice de calidad del aire (ICA) en diferentes estaciones de monitoreo de Medellín. El modelo se entrena con datos históricos de ICA, junto con variables climáticas como temperatura, humedad, velocidad del viento y precipitaciones. Se implementa una arquitectura secuencial con múltiples capas LSTM y capas densas, empleando funciones de activación ReLU y softmax para la salida categórica del nivel de calidad del aire. El estudio concluye que las redes LSTM son adecuadas para capturar patrones temporales en series de tiempo ambientales, y que el modelo predice con mayor precisión durante ciertas temporadas del año, lo que destaca la importancia de considerar estacionalidades y variables exógenas. Se valida el desempeño del modelo con métricas como Accuracy y F1-score, obteniendo resultados superiores al 90% en escenarios de predicción binaria [19].

Este estudio local aporta evidencia valiosa de la viabilidad de aplicar modelos LSTM en el contexto colombiano, en particular en ciudades con condiciones ambientales y topográficas complejas como Medellín. Los resultados refuerzan la aplicabilidad de las redes neuronales recurrentes para pronósticos ambientales en entornos urbanos latinoamericanos [19].

### **3.2.9 TÉCNICAS DE APRENDIZAJE AUTOMÁTICO PARA EL ANÁLISIS DE DATOS DE CALIDAD DEL AIRE.**

Se presenta el diseño e implementación de un sistema de predicción del Índice de Calidad del Aire (ICA) utilizando modelos supervisados como regresión lineal, máquinas de soporte vectorial (SVR), bosques aleatorios (Random Forest) y redes neuronales artificiales (ANN). La investigación utiliza datos meteorológicos y de calidad del aire recopilados en estaciones de la Comunidad de Madrid, considerando variables como temperatura, humedad, velocidad del viento, concentración de NO<sub>2</sub>, O<sub>3</sub> y PM10. El análisis comparativo entre modelos se realiza mediante métricas de error (MAE, RMSE, R<sup>2</sup>), identificando que los modelos de bosques aleatorios presentan el mejor rendimiento global, seguidos por las redes neuronales [20].

El sistema se implementa en Python, integrando bibliotecas como Scikit-learn, Pandas y TensorFlow, y se valida su utilidad mediante visualizaciones interactivas y un entorno web para consulta ciudadana. El estudio destaca la importancia de aplicar técnicas de preprocesamiento, normalización de datos, y selección de atributos relevantes para mejorar la precisión de los modelos, además de resaltar los retos asociados al desbalance en los datos y la variabilidad estacional de los contaminantes. El enfoque metodológico y la implementación práctica propuestos en este trabajo aportan elementos replicables y escalables para proyectos orientados al monitoreo ambiental mediante inteligencia artificial. Asimismo, la comparación de múltiples modelos de machine learning permite justificar la elección de arquitecturas avanzadas como Random Forest, ANN o modelos híbridos en desarrollos futuros, especialmente en zonas con condiciones climáticas y emisiones variables [20].

### **3.2.10 APLICACIÓN DE MODELO XGB PARA PREDICCIÓN DE CALIDAD DE AIRE, BOGOTÁ - COLOMBIA.**

Se plantea una solución integral que combina redes neuronales artificiales (ANN) y visualización para la predicción del ICA. El sistema usa los sistemas distribuidos de la Red de monitoreo de calidad del aire de Bogotá, estos recolectan datos ambientales como niveles de PM2.5 y PM10. Estos datos son transmitidos a un servidor central que ejecuta modelos de aprendizaje automático entrenados para clasificar los niveles del ICA según la norma colombiana. El modelo ANN utilizado fue entrenado con datos históricos de la Red de Monitoreo [21].

Esta investigación proporciona un antecedente directo en el contexto urbano de Bogotá, demostrando la viabilidad de combinar aprendizaje automático y visualización en tiempo real para la gestión ambiental. La implementación del sistema inteligente y la arquitectura de red neuronal refuerzan la pertinencia de utilizar modelos basados en datos históricos locales y condiciones atmosféricas específicas [21].

A continuación, se resume el impacto y las técnicas mencionadas en los proyectos que construyeron este estado del arte

Tabla 1. Resumen artículos estado del arte

<b>Artículo - Título abreviado</b>	<b>Técnicas aplicadas</b>	<b>Impacto en el proyecto</b>
Prediction and Analysis of AQI	DTR, LR, RFR, SVR	Resalta la superioridad del modelo RFR para predicción de AQI; base comparativa de técnicas tradicionales.
The Prediction of Quality of the Air Using Supervised Learning	Árbol de decisión	Enfatiza la clasificación de contaminantes y su visualización; promueve desarrollos con interfaz para usuarios.
ML Model for AQI in Smart Cities	ANN, SVR	Demuestra precisión alta con redes neuronales y SVR; justifica el uso de modelos híbridos en ciudades inteligentes.
Prototipo en Kennedy, Bogotá	ANN	Proporciona enfoque local con sensores y ANN para PM; destaca la aplicabilidad en contextos urbanos específicos.
Predicting AQI with Hybrid Deep Learning	CNN-LSTM + PSO	Demuestra robustez en predicción multivariable de PM2.5; sustenta el uso de modelos híbridos con mecanismos de atención.
Application CNN-LSTM (Beijing)	CNN-LSTM	Valida modelos híbridos multiciudad; útil para el diseño predictivo con variables meteorológicas.
Hybrid DL for AQI Classification	Bi-LSTM + Multi-Head Attention	Refuerza precisión en clasificación multisalida; relevante para sistemas urbanos con múltiples contaminantes.
Predicción en Medellín con LSTM	LSTM	Confirma la efectividad del LSTM en ciudades latinoamericanas; respalda la inclusión de estacionalidad y variables climáticas.
ML en Madrid (técnicas múltiples)	LR, SVR, RF, ANN	Comparativa robusta; destaca la utilidad de Random Forest y técnicas de preprocesamiento para precisión y replicabilidad.
Predicción ICA en Bogotá (XGB+ANN)	XGBoost, ANN	Aplicación directa en Bogotá; evidencia la viabilidad de sistemas distribuidos con predicción ICA conforme a norma local.

Fuente: elaboración propia.

#### 4. DESARROLLO DEL PROYECTO: ESTRUCTURA, PREPARACIÓN Y DISEÑO DEL TRABAJO

El diseño e implementación del proyecto requirió el desarrollo de un sistema modular que integra dispositivos de bajo costo, plataformas de procesamiento en la nube, técnicas de inteligencia artificial y visualización web. La estructura general del sistema se organizó en cinco componentes clave: adquisición de datos, transmisión y almacenamiento, análisis y predicción, visualización, y sistema de alertas automáticas.

##### 4.1 Adquisición de datos

Para la medición de contaminantes se emplearon los siguientes sensores:

- [SEN0564](#) (CO): Sensor de detección de gas de monóxido de carbono (CO). Emplea tecnología de sistemas microelectromecánicos (MEMS) de última generación, lo que dota al sensor de dimensiones compactas, bajo consumo de energía, mínima generación de calor, un corto tiempo de precalentamiento y una rápida recuperación de respuesta. [22]
- [PMS5003](#) (PM2.5): Sensor óptico láser capaz de detectar material particulado fino (PM2.5) con alta resolución temporal y exactitud [23].

Cada sensor fue conectado a un microcontrolador ESP32-WROOM-32 [24], que se encarga de la lectura periódica de datos y su transmisión. Este microcontrolador cuenta con Wi-Fi integrado, bajo consumo de energía y capacidad de procesamiento local.

La programación de los microcontroladores se realizó mediante Arduino IDE [25], configurando los intervalos de muestreo en 5 minutos para ambos contaminantes, de acuerdo con las limitaciones de la versión gratuita para estudiante utilizada en la plataforma ThingSpeak [26]. Se implementaron rutinas para ajustes iniciales, verificación de conectividad, y envío de datos a la nube en caso de detección correcta.

##### 4.2 Transmisión y almacenamiento en la nube

La plataforma utilizada para la transmisión y almacenamiento fue ThingSpeak [27], un entorno cloud gratuito para proyectos de IoT, que permite crear canales por dispositivo y por variable. Cada uno de los medidores instalados en Bogotá (Zona UCompensar, UManuela Beltrán y Zona Centro – selección de ubicación explicado en objetivo 2) cuenta con su propio canal y campos definidos para CO y PM2.5.

Los datos recibidos en ThingSpeak se almacenan automáticamente y pueden ser exportados en formato CSV. También es posible consultar los datos mediante API REST para su posterior procesamiento en Python o para ser visualizados mediante widgets o interfaces externas.

##### 4.3 Análisis y predicción con Python y modelos LSTM

Los datos exportados se analizaron y procesaron en Google Colab, empleando Python y

bibliotecas especializadas: pandas, numpy, matplotlib, seaborn, scikit-learn, keras y tensorflow. El flujo de trabajo consistió en:

- Análisis exploratorio de datos (EDA): se generaron gráficos como histogramas, curvas de densidad (KDE), boxplots por hora, mapas de calor día-hora y curvas de tendencia semanal (media móvil de 7 días).
- Creación de ventanas de entrada para series temporales: 12 pasos para CO y 24 para PM2.5.
- División entrenamiento/prueba: 80 % entrenamiento y 20 % prueba.
- Implementación de redes LSTM con arquitectura de dos capas ocultas: cada capa optimizada por medio de Keras Tuner con Random Search, lo cual permitió encontrar combinaciones eficientes de hiperparámetros como:
  - Units\_1, units\_2 (neuronas por capa)
  - Dropout\_1, dropout\_2 (tasa de regularización)
  - Time\_steps (pasos anteriores considerados como entrada)

Cada medidor y contaminante tiene un modelo independiente. Por ejemplo, para el CO del medidor 1, se obtuvieron:

- Hiper-parámetros: time\_steps = 12, units\_1 = 96, units\_2 = 96
- Métricas: RMSE = 0.028 ppm,  $R^2 = 0.988$

Todos los modelos fueron guardados junto a sus escaladores (scaler) para ser reutilizados posteriormente en entornos productivos o desplegados en APIs.

#### 4.4 Cálculo del Índice de Calidad del Aire (AQI)

Con los resultados de las predicciones, se aplicó la documentación oficial de la EPA (Environmental Protection Agency) para convertir los valores de CO y PM2.5 en el Índice de Calidad del Aire (AQI) [28]. Este cálculo requiere:

- Promedio móvil de 8 h (CO) o 24 h (PM2.5)
- Asignación de rangos según "breakpoints"
- Interpretación mediante códigos de color y nivel de riesgo

Esta lógica fue implementada tanto en Python como en MATLAB Analysis dentro de ThingSpeak, donde se generan los scripts que actualizan el AQI automáticamente y lo guardan en campos específicos para su visualización.

#### 4.5 Visualización y sistema de alertas

Para la visualización pública se diseñó una interfaz utilizando Google Sites, donde se incrustan los widgets de gráficos y valores en tiempo real provenientes de ThingSpeak mediante iframe. Esta herramienta fue elegida por su facilidad de integración, gratuidad y compatibilidad con múltiples dispositivos.

El sistema de alertas automáticas se implementó mediante un bot de Telegram. Este bot notifica de forma inmediata cuando los niveles de CO/PM2.5 ingresan a rangos de riesgo para la salud, según los umbrales definidos por la EPA.

Cada componente del sistema está documentado y conectado entre sí, garantizando un flujo continuo desde la adquisición hasta la visualización y alerta, con bajo costo, alta replicabilidad y autonomía técnica.

Tabla 2. Herramientas utilizadas

<b>VERTICAL</b>	<b>HERRAMIENTA/PLATAFORMA</b>	<b>FUNCIÓN PRINCIPAL</b>
Recolección	Arduino IDE + bibliotecas ESP32	Lectura de sensores y envío Wi-Fi
Nube	ThingSpeak	Almacenamiento y visualización de datos
Análisis	MATLAB Analytics	Estadística básica, promedios móviles
ML	Python + TensorFlow/Keras	Entrenamiento y predicción con LSTM
Visualización	Google Sites + Gráficos	Presentación pública
Alertas	Telegram	Notificación ante superación de AQI

Fuente: elaboración propia.

Para consolidar el desarrollo del proyecto se presenta el siguiente diagrama de arquitectura:

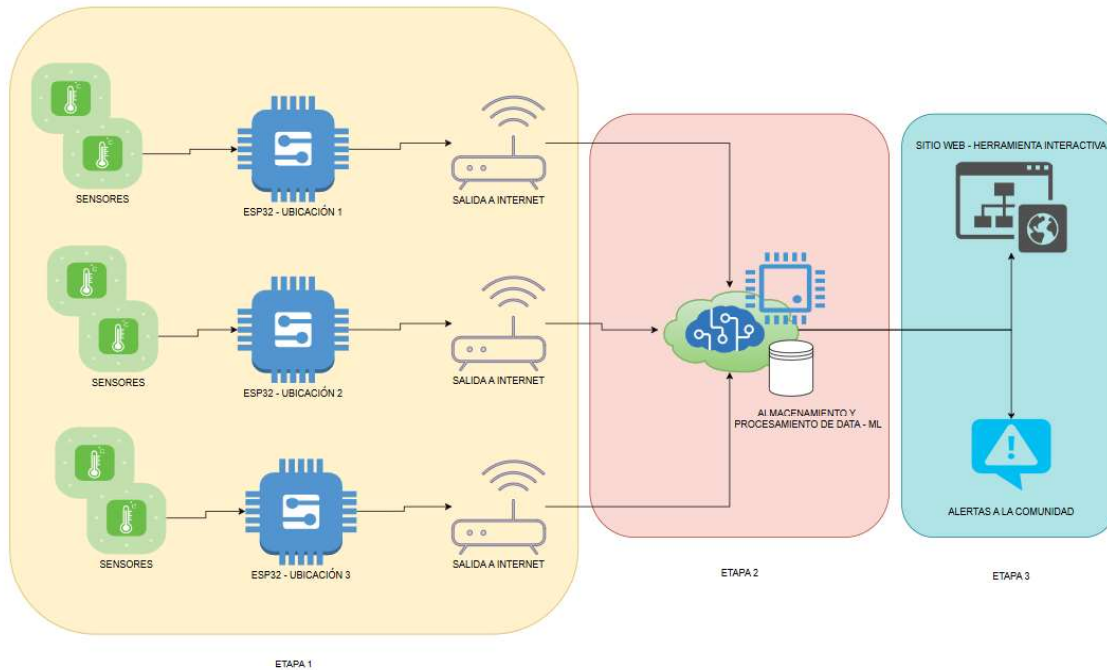


Figura 1. Topología general de la solución desarrollada.  
Fuente: elaboración propia.

El diagrama o topología de la Figura 1 resume el desarrollo del proyecto: se observan las etapas claves del proyecto. En la primera etapa la captura de variables y preparación de estas, posterior la programación para el envío de datos con una ventana de tiempo establecida (cinco minutos) y el aseguramiento de la conectividad WIFI para el envío de datos a través de Internet hasta el servidor de aplicaciones (etapa 2) donde se almacenó y procesó la data para el posterior entrenamiento del modelo de predicción. En la etapa 3 se publican los resultados en un sitio web y se generan las notificaciones correspondientes a la escala definida por el marco de trabajo para la clasificación de datos para la calidad del aire.

## 5. ACTIVIDADES Y RESULTADOS OBTENIDOS OBJETIVO ESPECÍFICO 1

### Objetivo específico 1:

Seleccionar y configurar sensores de calidad del aire basados en IoT para medir los niveles de CO y PM2.5 en los puntos de monitoreo, identificando y seleccionando tres puntos geográficos en Bogotá para la instalación de sensores de calidad del aire en el sistema de monitoreo propuesto.

### 5.1 Selección de sensores adecuados

Para la selección del sensor de CO que brindara buenas condiciones y con un costo en el mercado favorable para la implementación del proyecto se construyó la siguiente tabla comparativa relacionando variables como el rango de medición, precisión, selectividad, entre otros.

Tabla 3. Comparativo de los sensores evaluados para medir monóxido de carbono

CARACTERÍSTICA	MQ-7	MQ-9	KEMET USEQGDAC8L 100	SGX-4CO-LC	CCS TGS862	426-SEN0564
TECNOLOGÍA	Conductividad semiconductor	Conductividad semiconductor	MEMS electroquímica	Infrarrojo (NDIR)	Conductividad semiconductor	MEMS electroquímica
RANGO DE MEDICIÓN	20 ppm - 2000 ppm	50 ppm - 10000 ppm	15 ppm - 300 ppm	10 ppm - 400 ppm	10 ppm - 2000 ppm	5 ppm - 500 ppm
PRECISIÓN	Moderada	Moderada	Alta	Muy alta	Alta	Alta
SELECTIVIDAD	Baja	Baja	Alta	Muy alta	Alta	Alta
TIEMPO DE RESPUESTA	Rápido (menos de 10 segundos)	Rápido (menos de 10 segundos)	Moderado (alrededor de 30 segundos)	Lento (alrededor de 2 minutos)	Rápido (menos de 10 segundos)	Rápido (menos de 10 segundos)
DERIVA A LARGO PLAZO	Alta	Alta	Baja	Muy baja	Alta	Baja
CONSUMO DE ENERGÍA	Moderado	Moderado	Bajo	Muy bajo	Moderado	Bajo
INTERFAZ	Analógica	Analógica	Digital	Digital	Analógica	Digital
VENTAJAS	Bajo costo, fácil de usar	Bajo costo, mayor rango de medición	Alta precisión y selectividad	Alta precisión, selectividad y estabilidad	Bajo costo, alta sensibilidad	Alta precisión, selectividad y bajo consumo de energía
RESISTENCIA A LA INTEMPERIE	Baja	Baja	Media	Alta	Baja	Media
PROTECCIÓN	No	No	Media	Alta	No	Media

CONTRA RAYOS UV						
-----------------	--	--	--	--	--	--

Fuente: elaboración propia.

En relación con lo presentado en la tabla, se selecciona el sensor 426-SEN0564. Esto gracias a los siguientes parámetros evaluados:

- **Precisión:** El 426-SEN0564 ofrece alta precisión en la medición
- **Selectividad:** Este sensor tiene una alta selectividad, lo que significa que puede distinguir de manera efectiva entre diferentes gases, minimizando las interferencias de otros compuestos presentes en el entorno.
- **Tecnología:** Utiliza tecnología MEMS (Sistemas Microelectromecánicos) electroquímica, que proporciona una combinación de alta sensibilidad y estabilidad. Es una tecnología superior a la conductividad semiconductor utilizada en otros sensores como el MQ-7, MQ-9 y CCS TGS862.
- **Consumo de Energía:** Bajo.
- **Rango de Medición:** Con un rango de medición de 5 ppm a 500 ppm, el 426-SEN0564 es adecuado para una amplia variedad de aplicaciones.
- **Interfaz:** Dispone de una interfaz digital que facilita su integración en sistemas modernos de monitoreo y control.
- **Resistencia y Protección:** Este sensor tiene una resistencia media a la intemperie y una protección media contra rayos UV, lo que lo hace adecuado para aplicaciones exteriores y en condiciones ambientales moderadas.

Para la elección de sensores de PM2.5 se consideró parámetros como el rango de medida, la precisión, interfaz de lectura y consumo. A continuación, se presenta una tabla comparativa y las justificaciones relacionadas con la elección del sensor usado en esta tarjeta de adquisición de datos.

Tabla 4. Comparativo de los sensores para medir material particulado PM2.5

Característica	PMS5003	SEN0460	PMSA003I	SEN54
PARÁMETRO MEDIDO	PM2.5, PM10, PM1.0	PM2.5, CO2	PM2.5, PM10, PM1.0	PM2.5, PM10, PM1.0, COV, Humedad, Temperatura
PRECISIÓN	±15% para PM2.5	±3 ppm para CO2, ±10% para PM2.5	±10% para PM2.5	±15% para PM2.5, ±2 ppm para COV, ±3% RH, ±0.2°C
RANGO DE MEDICIÓN	0.3 µg/m <sup>3</sup> a 50 µg/m <sup>3</sup> para PM2.5, 0.3 µg/m <sup>3</sup> a 100 µg/m <sup>3</sup> para PM10, 0.3 µg/m <sup>3</sup> a 30 µg/m <sup>3</sup> para PM1.0	0 ppm a 10,000 ppm para CO2, 0.3 µg/m <sup>3</sup> a 50 µg/m <sup>3</sup> para PM2.5	0.3 µg/m <sup>3</sup> a 50 µg/m <sup>3</sup> para PM2.5, 0.3 µg/m <sup>3</sup> a 100 µg/m <sup>3</sup> para PM10, 0.3 µg/m <sup>3</sup> a 30 µg/m <sup>3</sup> para PM1.0	0 µg/m <sup>3</sup> a 100 µg/m <sup>3</sup> para PM2.5, 0 µg/m <sup>3</sup> a 100 µg/m <sup>3</sup> para PM10, 0 µg/m <sup>3</sup> a 30 µg/m <sup>3</sup> para PM1.0, 0 ppb a 20 ppm para COV, 0% a 100% RH, -40°C a 80°C
INTERFAZ DE COMUNICACIÓN	UART, I2C	UART, I2C	UART, I2C	I2C

Característica	PMS5003	SEN0460	PMSA003I	SEN54
VOLTAJE DE FUNCIONAMIENTO	5V	3.3V o 5V	5V	3.3V o 5V
CONSUMO DE CORRIENTE	30 mA	10 mA	30 mA	10 mA
TEMPERATURA DE FUNCIONAMIENTO	-10°C a 60°C	-20°C a 70°C	-10°C a 60°C	-40°C a 80°C
HUMEDAD DE FUNCIONAMIENTO	20% a 80% RH	0% a 95% RH	20% a 80% RH	0% a 100% RH
VENTAJAS	Bajo costo, alta precisión, amplio rango de medición, interfaz flexible, bajo consumo de corriente, tamaño compacto, fácil de usar, ampliamente disponible	Mide CO <sub>2</sub> , alta precisión para PM <sub>2.5</sub>	Mayor precisión y estabilidad que el PMS5003, mide CO <sub>2</sub>	Mide múltiples parámetros ambientales
DESVENTAJAS	No mide CO <sub>2</sub> o COV	No mide CO <sub>2</sub>	Más caro que el PMS5003	Más caro y sensible a EMI
APLICACIONES	Monitoreo de calidad del aire en interiores y exteriores, sistemas de ventilación inteligentes, purificadores de aire, investigación ambiental	Monitoreo de calidad del aire en interiores, sistemas de ventilación inteligentes, monitoreo de CO <sub>2</sub> en invernaderos, investigación ambiental	Mismo que PMS5003	Monitoreo de calidad del aire en interiores y exteriores, sistemas de ventilación inteligentes, purificadores de aire, investigación ambiental

Fuente: elaboración propia.

De acuerdo con la tabla anterior se destacan las siguientes ventajas del sensor PMS5003 para el desarrollo de este proyecto:

- **Bajo costo:** El PMS5003 es uno de los sensores de PM<sub>2.5</sub> más económicos del mercado, lo que lo convierte en una opción atractiva para **proyectos con limitaciones presupuestarias**.
- **Alta precisión:** El PMS5003 ofrece una precisión de medición de  $\pm 15\%$  para PM<sub>2.5</sub>, lo que lo convierte en un sensor **confiable** para medir niveles de partículas finas en el aire en Bogotá.
- **Amplio rango de medición:** El PMS5003 puede medir concentraciones de PM<sub>2.5</sub> entre 0.3  $\mu\text{g}/\text{m}^3$  y 50  $\mu\text{g}/\text{m}^3$ , lo que lo hace adecuado para monitorear una amplia gama de niveles

de contaminación del aire, que es una característica importante en una ciudad como Bogotá, donde los niveles de PM2.5 pueden variar significativamente dependiendo de la zona y la época del año.

- **Interfaz de comunicación flexible:** El PMS5003 se puede comunicar con microcontroladores y ordenadores a través de interfaces UART e I2C, lo que facilita su integración en diferentes sistemas de adquisición de datos. Esta flexibilidad permite adaptarlo fácilmente a la configuración de la estación prototipo, independientemente de la plataforma de hardware y software utilizada.
- **Bajo consumo de corriente:** El sensor seleccionado consume solo 30 mA, lo que lo convierte en un sensor ideal para aplicaciones que funcionan con bajo consumo de energía.
- **Tamaño compacto:** El PMS5003 tiene un tamaño compacto, lo que lo hace ideal para aplicaciones donde el espacio es limitado. Esto permite integrarlo fácilmente en la estructura de la estación prototipo sin ocupar un volumen considerable.

En comparación con otros sensores:

- El PMSA003I ofrece mayor precisión y estabilidad, pero es más caro.
- El SEN54 mide múltiples parámetros ambientales, pero también es más caro y sensible a EMI.
- El SEN0460 también mide CO<sub>2</sub>, pero no tiene un rango de medición tan amplio como el PMS5003.

En resumen, los sensores seleccionados son los siguientes:

**Sensor SEN0564:** sensor de tipo electroquímico para medir monóxido de carbono (CO). Ofrece una buena sensibilidad, tiempo de respuesta rápido y estabilidad en entornos urbanos. Ideal para aplicaciones donde se requiere precisión con bajo consumo energético.



Figura 2. Sensor de monóxido de carbono.

Fuente: [29]



Figura 3. Sensor de PM2.5.

Fuente: [23]

**Sensor PMS5003:** sensor láser óptico de alta precisión, especializado en la medición de material particulado fino (PM2.5). Funciona mediante dispersión de luz, permitiendo detectar

concentraciones en tiempo real. Posee una alta resolución y es ampliamente validado en investigaciones científicas.

Ambos sensores fueron conectados a una tarjeta ESP32, la cual permite la lectura analógica y digital, además de poseer conectividad Wi-Fi integrada, capacidad de cómputo adecuada y eficiencia energética.

## 5.2 Selección de puntos de monitoreo



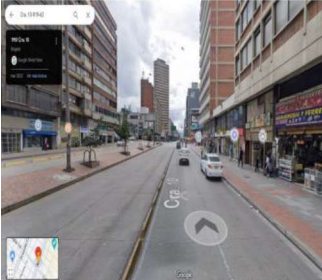
Para la selección estratégica de puntos para el análisis de condiciones ambientales, se revisaron las ubicaciones actuales que tiene la Red de Monitoreo de Calidad del Aire de Bogotá [RMCAB](#). Para estudiar el comportamiento principalmente de las variables de Monóxido de carbono (CO) y Material Particulado 2.5 micras (PM2.5) se seleccionaron 3 ubicaciones estratégicas a las siguientes estaciones de monitoreo. Estas ubicaciones contaron con apoyo de universidades que brindaron el acceso para la instalación de los prototipos. Esto permitió conectar las tarjetas a Internet (en la banda 2.4GHz) y la alimentación constante en diferentes momentos del tiempo de operación de los prototipos.

También se consideraron las concentraciones de material particulado elevado por las situaciones medioambientales que ocurrieron en Bogotá, por ejemplo, la propagación de incendios forestales en los cerros orientales de la ciudad. Estas son las tres estaciones relacionadas en este proyecto ubicadas geográficamente en Bogotá; los datos son tomados del sitio web oficial de RMCAB

- Zona Teusaquillo – Institución Universitaria UCompensar (Medidor 1): Área urbana universitaria, con alta afluencia de vehículos y peatones.
- Zona Chapinero – Universidad Manuela Beltrán (Medidor 2): Contexto mixto residencial-educativo con tráfico intermedio. Ubicación inmersa en los cerros de la ciudad
- Zona Centro – Universidad ECCI (Medidor 3): Corazón de la ciudad, altamente congestionado, zona crítica en términos de emisiones.

Estos puntos fueron seleccionados aprovechando la accesibilidad, seguridad para los equipos, representatividad del entorno urbano y viabilidad técnica (disponibilidad eléctrica y de red Wi-Fi).

Tabla 5. Puntos de monitoreo Bogotá

PUNTO DE MONITOREO	ZONA	CARACTERÍSTICAS GEOGRÁFICAS
Punto 1	Zona Teusaquillo – Institución Universitaria UCompensar - Avenida (Calle) 32 No. 17 – 30 	Zona urbana central, tráfico medio
Punto 2	Zona Chapinero – Universidad Manuela Beltrán - Cra 1ª #No. 60-00 	Sector mixto, industrial-comercial
Punto 3	Zona Centro – Universidad ECCI - Carrera 10 # 19- 62 	Tráfico elevado, alta densidad urbana

Fuente: elaboración propia.

### 5.3 Ensamblaje e instalación del hardware

Cada estación de monitoreo fue ensamblada incluyendo:

- Microcontrolador ESP32
- Sensor SEN0564 para CO y Sensor PMS5003 para PM2.5
- Baquelita para ensamble

- Conversores de voltaje y cables
- Caja de protección resistente al clima
- Fuente de alimentación estable

Las estaciones se instalaron en lugares seguros con acceso a red Wi-Fi, garantizando una transmisión continua hacia la nube mediante [ThingSpeak](#), una plataforma IoT que permite registrar, visualizar y analizar en tiempo real los datos enviados por sensores y dispositivos conectados. Cada canal de datos en ThingSpeak fue configurado con campos independientes para CO, PM2.5 y sus respectivos valores AQI.

#### 5.4 Resultados obtenidos

Al completar esta fase, se logró:

- La instalación funcional de los tres medidores de calidad del aire.
- Lecturas en tiempo real de los contaminantes cada 5 minutos.
- Transmisión automática de los datos vía Wi-Fi a ThingSpeak.
- Visualización en la nube y recolección histórica para su análisis posterior.

Además, se verificó que los sensores entregaban valores coherentes con la variabilidad esperada en zonas urbanas, conforme a las condiciones ambientales observadas y los rangos de referencia ofrecidos por la Red de Monitoreo de Calidad del Aire de Bogotá (RMCAB).

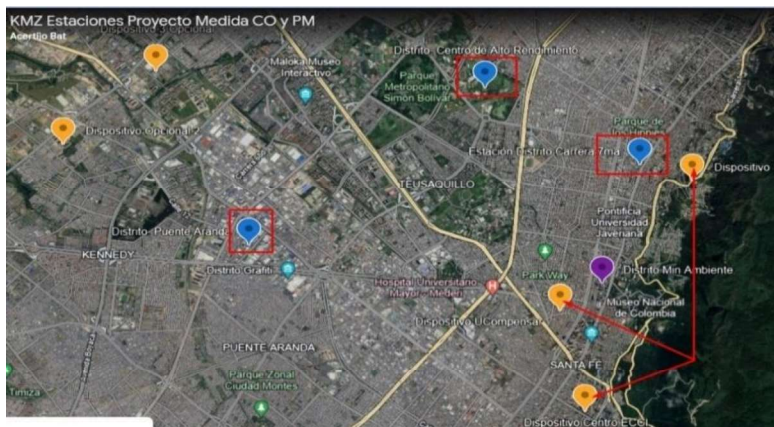


Figura 4. Puntos de monitoreo  
Fuente: elaboración propia.

Mapa con los puntos de medición prototipo comparado con las estaciones de medida oficiales.

## 6. ACTIVIDADES Y RESULTADOS OBTENIDOS OBJETIVO ESPECÍFICO 2

### Objetivo específico 2:

Desarrollar la infraestructura de IoT necesaria para recopilar, transmitir y almacenar los datos de los sensores en tiempo real, implementando un sistema de recopilación de datos de CO y PM2.5.

### 6.1 Diseño de la infraestructura IoT

La arquitectura del sistema fue diseñada bajo el enfoque de IoT (Internet de las Cosas), considerando la integración de sensores, microcontroladores, conectividad inalámbrica, servicios en la nube y entornos de análisis para garantizar un flujo continuo de datos. El flujo de trabajo se definió así:

1. Captura de datos desde los sensores SEN0564 (CO) y PMS5003 (PM2.5).
2. Procesamiento y transmisión local desde el microcontrolador ESP32, programado en Arduino IDE.
3. Conexión Wi-Fi desde el ESP32 hacia ThingSpeak, actuando como servidor en la nube.
4. Almacenamiento estructurado, visualización en tiempo real y descarga en formatos como CSV para análisis posterior.

El desarrollo de la infraestructura IoT siguió el siguiente diseño:

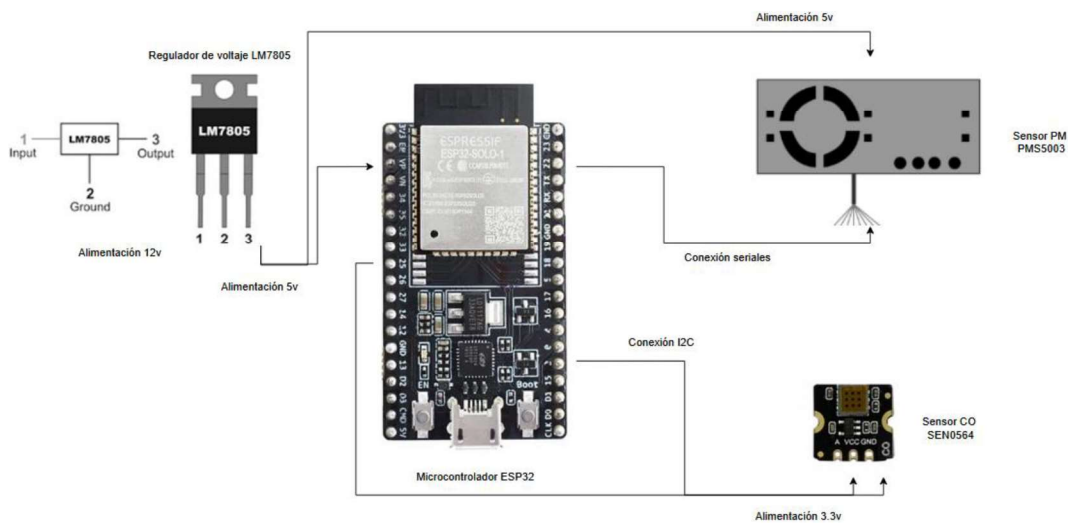


Figura 5. Infraestructura IoT.  
Fuente: elaboración propia.

Para la adquisición de las variables mediante los sensores se construye un código apoyado en las hojas técnicas de los sensores y en las interfaces de lectura que proporcionan. Para el sensor

PMS5003 la lectura se gestiona mediante comunicación RS232 y para el sensor SEN0564 aprovechando la interfaz análoga. Esto permite que el bloque de código tenga una etapa de lectura de sensores, interpretación de valores y envío de variables al servidor de análisis.

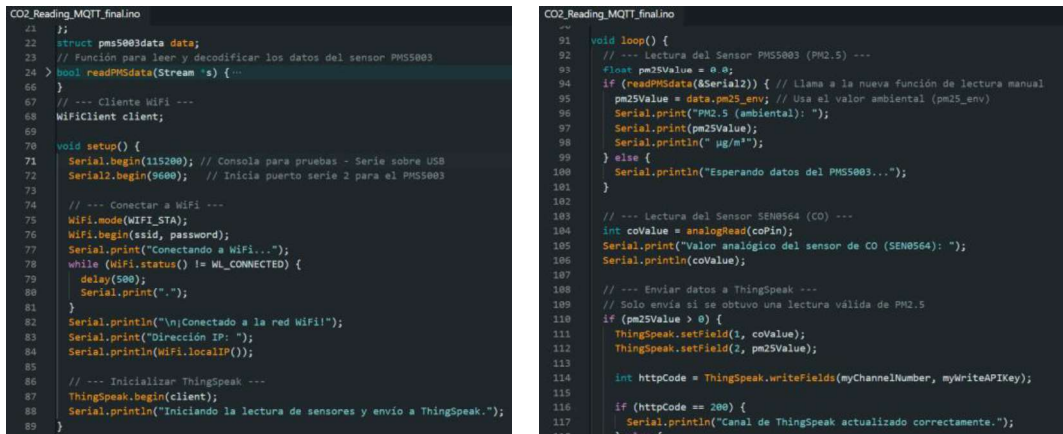
```

1  #include <Arduino.h>
2  #include <WiFi.h>
3  #include "ThingSpeak.h" // Librerías
4
5  // --- Configuración de Red WiFi ---
6  const char* ssid = "Troyn020-2G";
7  const char* password = "Mundo11";
8  // --- Configuración de ThingSpeak ---
9  unsigned long myChannelNumber = 2986083;
10 const char * myWriteAPIKey = "OLMHOGAPMDKYQTB";
11 // --- Configuración del Sensor SEN0564 (CO) ---
12 const int coPin = 36; // Pin ADC para el sensor de CO - ADC1_0
13 // --- Estructura para PMS5003 ---
14 > struct pms5003data { ...
21 };
22 struct pms5003data data;
23 // Función para leer y decodificar los datos del sensor PMS5003
24 > bool readPMSdata(Stream *s) { ...
66 }
67 // --- Cliente WiFi ---
68 WiFiClient client;
69
70 void setup() {
71   Serial.begin(115200); // Consola para pruebas - Serie sobre USB
72   Serial2.begin(9600); // Inicia puerto serie 2 para el PMS5003
73 }

```

Figura 6. Código ESP32 – Adquisición de variables y conectividad.  
Fuente: elaboración propia.

Dentro de este bloque de configuración también se observa la asociación del canal de ThingSpeak, asociación de bibliotecas y configuración de interfaces seriales. Además, se indica las credenciales de la red de WIFI y su configuración, se conecta la tarjeta a la red WIFI para el envío de datos.



```

CO2_Reading_MQTT_finalino
...
21 };
22 struct pms5003data data;
23 // Función para leer y decodificar los datos del sensor PMS5003
24 > bool readPMSdata(Stream *s) { ...
66 }
67 // --- Cliente WiFi ---
68 WiFiClient client;
69
70 void setup() {
71   Serial.begin(115200); // Consola para pruebas - Serie sobre USB
72   Serial2.begin(9600); // Inicia puerto serie 2 para el PMS5003
73
74   // --- Conectar a WiFi ---
75   WiFi.mode(WIFI_STA);
76   WiFi.begin(ssid, password);
77   Serial.print("Conectando a WiFi...");
78   while (WiFi.status() != WL_CONNECTED) {
79     delay(500);
80     Serial.print(".");
81   }
82   Serial.println("\nConectado a la red WiFi!");
83   Serial.print("Dirección IP: ");
84   Serial.println(WiFi.localIP());
85
86   // --- Inicializar ThingSpeak ---
87   ThingSpeak.begin(client);
88   Serial.println("Iniciando la lectura de sensores y envío a ThingSpeak.");
89 }

CO2_Reading_MQTT_finalino
...
91 void loop() {
92   // --- Lectura del Sensor PMS5003 (PM2.5) ---
93   float pm25Value = 0.0;
94   if (readPMSdata(&Serial2)) { // Llama a la nueva función de lectura manual
95     pm25Value = data.pm25_env; // Usa el valor ambiental (pm25_env)
96     Serial.print("PM2.5 (ambiental): ");
97     Serial.print(pm25Value);
98     Serial.println(" µg/m³");
99   } else {
100    Serial.println("Esperando datos del PMS5003...");
101   }
102
103   // --- Lectura del Sensor SEN0564 (CO) ---
104   int coValue = analogRead(coPin);
105   Serial.print("Valor analogico del sensor de CO (SEN0564): ");
106   Serial.println(coValue);
107
108   // --- Enviar datos a ThingSpeak ---
109   // Solo envía si se obtuvo una lectura válida de PM2.5
110   if (pm25Value > 0) {
111     ThingSpeak.setField(1, coValue);
112     ThingSpeak.setField(2, pm25Value);
113
114     int httpCode = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
115
116     if (httpCode == 200) {
117       Serial.println("Canal de ThingSpeak actualizado correctamente.");
118     }
119   }

```

Figura 7. Código ESP32 –Configuración de conectividad WIFI, lectura sensores y envío servidor.  
Fuente: elaboración propia.

Hay una serie de sentencias *Serial.println* para hacer el seguimiento correcto desde la interfaz de desarrollo de Arduino y verificar el flujo correcto del algoritmo.

El montaje físico del prototipo se presenta a continuación:

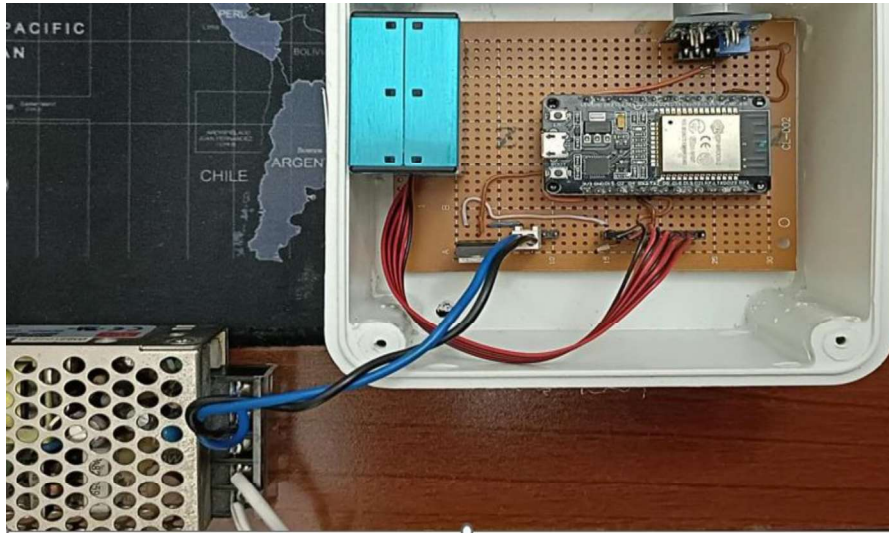


Figura 8. Montaje del prototipo para captura y envío de datos.  
Fuente: elaboración propia.



Figura 9. Montaje del prototipo para captura y envío de datos.  
Fuente: elaboración propia.



Figura 10. Montaje del prototipo para captura y envío de datos.  
Fuente: elaboración propia.

## 6.2 Plataforma ThingSpeak como núcleo del sistema

Se optó por ThingSpeak como plataforma de almacenamiento y visualización, dada su compatibilidad con los microcontroladores ESP32 y con MATLAB, y porque permite crear canales personalizados para cada medidor y contaminante.

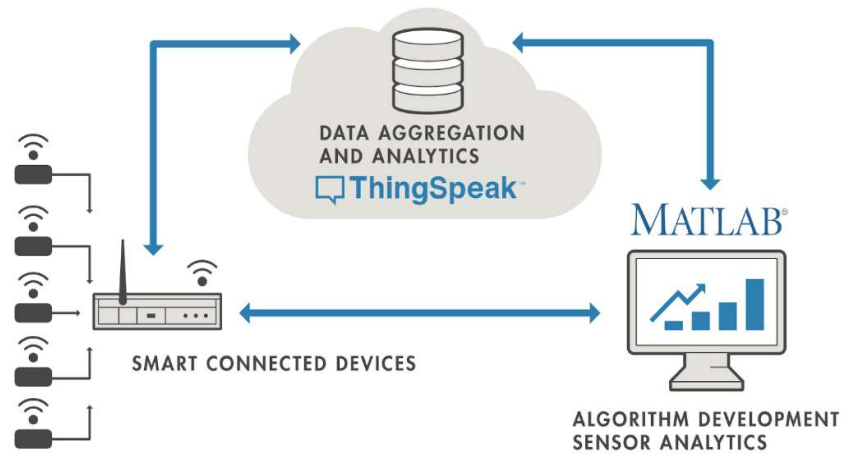


Figura 11. Operación ThingSpeak.  
Fuente: [26]

- Se crearon canales independientes por medidor (uno para cada punto geográfico).

## My Channels

New Channel

Q

Name ↕	Created ↕	Updated ↕
<div style="display: flex; align-items: flex-start;"> <div style="margin-right: 10px;">🔒</div> <div> <p>Datos Zona UCompensar Teusaquillo (Medidor 1)</p> <div style="display: flex; gap: 5px; margin-top: 5px;"> <span>Private</span> <span>Public</span> <span>Settings</span> <span>Sharing</span> <span>API Keys</span> <span>Data Import / Export</span> </div> </div> </div>	2025-06-10	2025-06-20 06:46
<div style="display: flex; align-items: flex-start;"> <div style="margin-right: 10px;">🔒</div> <div> <p>Datos Zona UManuela (Medidor 2)</p> <div style="display: flex; gap: 5px; margin-top: 5px;"> <span>Private</span> <span>Public</span> <span>Settings</span> <span>Sharing</span> <span>API Keys</span> <span>Data Import / Export</span> </div> </div> </div>	2025-06-10	2025-06-20 06:42
<div style="display: flex; align-items: flex-start;"> <div style="margin-right: 10px;">🔒</div> <div> <p>Datos Zona centro (Medidor 3)</p> <div style="display: flex; gap: 5px; margin-top: 5px;"> <span>Private</span> <span>Public</span> <span>Settings</span> <span>Sharing</span> <span>API Keys</span> <span>Data Import / Export</span> </div> </div> </div>	2025-06-10	2025-06-20 06:46

Figura 12. Configuración de canales ThingSpeak.  
Fuente: [26]

- Cada canal tiene campos exclusivos: Field 1 para CO, Field 2 para PM2.5, Field 3 y Field 4 para valores AQI.

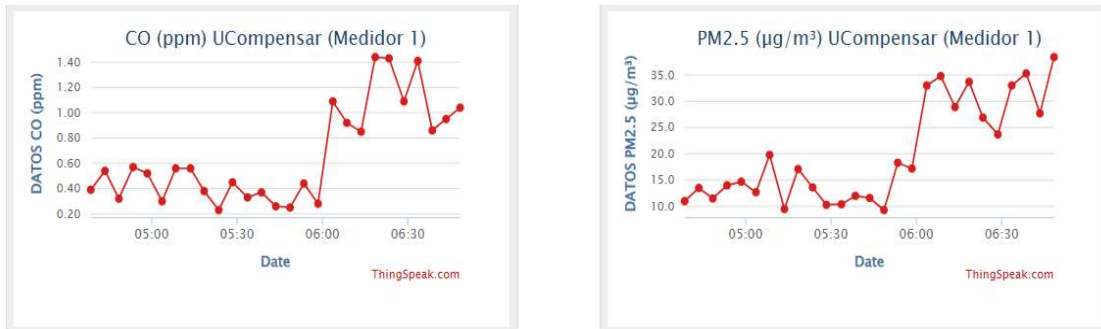


Figura 13. Gráficas generadas en ThingSpeak.  
Fuente: [26]



Figura 14. Gráficas AQI generadas en ThingSpeak.  
Fuente: [26]

- La plataforma permite ver las últimas lecturas, generar gráficos y programar scripts de análisis básico con MATLAB Analysis.

```

ThingSpeak™ Channels ▾ Apps ▾ Devices ▾ Support ▾
AQI PM2.5 MEDIDOR 1

MATLAB Code

1 % =====
2 % AQI PM2.5 → guarda en Field 4
3 % -----
4 % * Lee la concentración de PM2.5 (µg/m³) cada 5 min
5 % * Calcula el promedio móvil de 24 h (EPA + 288 muestras)
6 % * Convierte ese promedio a AQI y lo escribe en Field 4
7 % =====
8
9 % ----- 1. CONFIGURACIÓN DEL CANAL -----
10 readChannelID = 2986083; % <-- TU Channel ID
11 readAPIKey = 'QZP9SQ5FL48TD4Z'; % <-- TU Read Key (si es privada)
12 writeAPIKey = '0LWHDGAPMDKYQT8'; % <-- TU Write Key
13 fieldPM25 = 2; % <-- campo donde guardas PM2.5 en µg/m³
14 fieldAQI = 4; % <-- campo destino para el AQI calculado
15
16 % ----- 2. LEER ÚLTIMO DATO + HISTÓRICO 24 h -----
17 % Trae las últimas 288 muestras (24 h × 12 muestras/h con paso 5 min)
18 history = thingSpeakRead(readChannelID, ...
19 'Fields', fieldPM25, ...
20 'NumPoints', 288, ...
21 'ReadKey', readAPIKey);
22
23 if isempty(history)
24 error('No se pudo leer datos de PM2.5. Verifica el Field y la API key.');
```

Figura 15. Código en ThingSpeak y Matlab para procesamiento de datos recibidos.  
Fuente: [26]

### 6.3 Programación del envío de datos

Mediante el IDE de Arduino, se programaron los ESP32 para:

- Capturar datos cada 5 minutos.
- Verificar la conexión Wi-Fi y reconectarse en caso de fallo.
- Enviar los datos a los campos configurados en ThingSpeak.

El sistema resultó robusto: luego de varios días de funcionamiento continuo, las estaciones no presentaron caídas graves ni pérdida de datos significativa, validando la estabilidad de la infraestructura IoT implementada.

### 6.4 Visualización y consulta en línea

Gracias a ThingSpeak, se lograron los siguientes beneficios:

- Gráficas en tiempo real para cada contaminante:

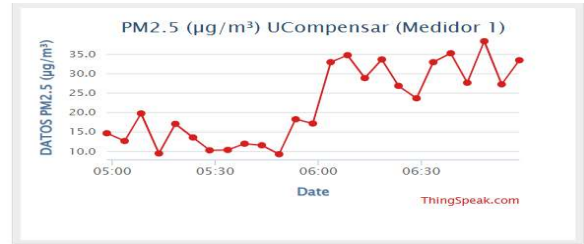
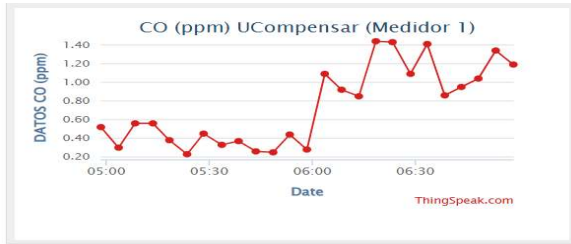


Figura 16. Gráficas generadas en ThingSpeak, medidor 1.

Fuente: [26]

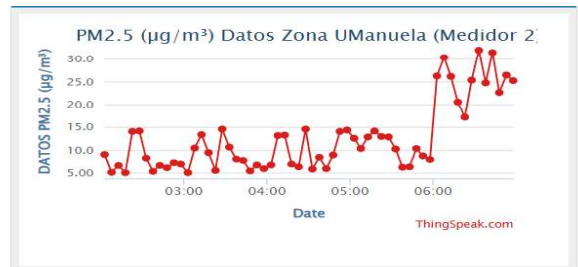
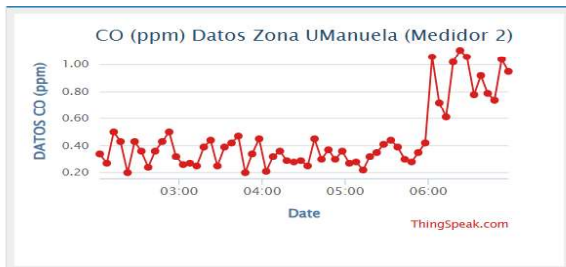


Figura 17. Gráficas generadas en ThingSpeak, medidor 2.

Fuente: [26]

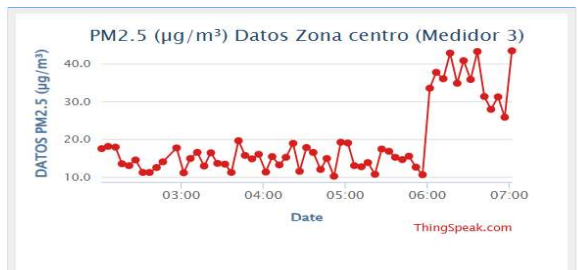
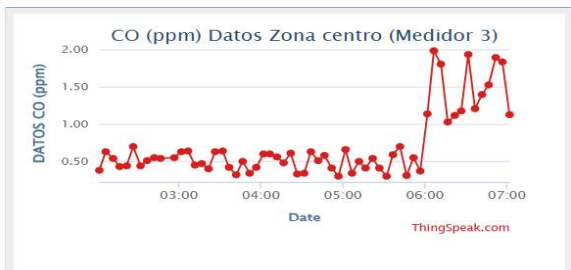


Figura 18. Gráficas generadas en ThingSpeak, medidor 3

Fuente: [26]

- Revisión remota desde cualquier navegador o dispositivo móvil:

<https://thingspeak.mathworks.com/channels/2986083>

### Datos Zona UCompensar Teusaquillo (Medidor 1) - ThingSpeak IoT

Captura de datos en tiempo real Medidor 1 - Datos Zona UCompensar Teusaquillo (Medidor 1) on ThingSpeak - ThingSpeak is the open IoT platform with MATLAB analytics.

thingspeak.mathworks.com

<https://thingspeak.mathworks.com/channels/2986109>

### Datos Zona centro (Medidor 3) - ThingSpeak IoT

Captura de datos en tiempo real de la zona centro en Bogotá Medidor 3 - Datos Zona centro (Medidor 3) on ThingSpeak - ThingSpeak is the open IoT platform with MATLAB analytics.

thingspeak.mathworks.com

<https://thingspeak.mathworks.com/channels/2986097>

### Datos Zona UManuela (Medidor 2) - ThingSpeak IoT

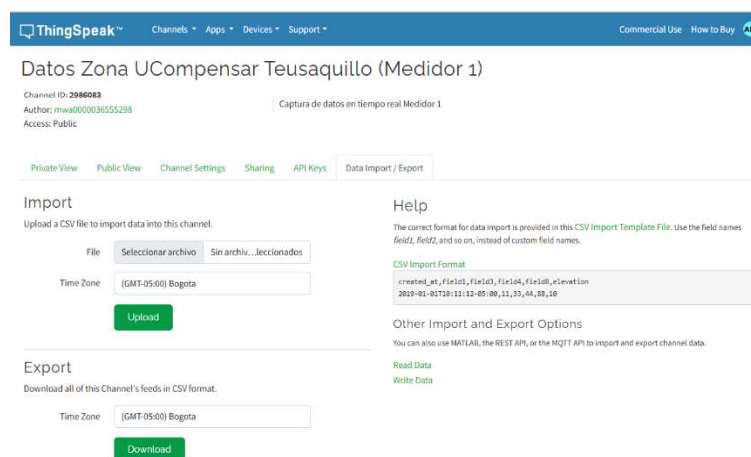
Captura de datos en tiempo real en Bogotá Medidor 2 - Datos Zona UManuela (Medidor 2) on ThingSpeak - ThingSpeak is the open IoT platform with MATLAB analytics.

thingspeak.mathworks.com

<

Figura 19. Valores de ThingSpeak publicados.  
Fuente: [26]

➤ Históricos descargables para cada canal:



ThingSpeak™ Channels Apps Devices Support Commercial Use How to Buy API

### Datos Zona UCompensar Teusaquillo (Medidor 1)

Channel ID: 2986083  
Author: mwa000036555298  
Access: Public

Captura de datos en tiempo real Medidor 1

Private View Public View Channel Settings Sharing API Keys Data Import / Export

#### Import

Upload a CSV file to import data into this channel.

File

Time Zone

#### Export

Download all of this Channel's feeds in CSV format.

Time Zone

#### Help

The correct format for data import is provided in this CSV Import Template File. Use the field names *field1*, *field2*, and so on, instead of custom field names.

CSV Import Format

```
created_at,field1,field2,field3,field4,elevation
2019-01-02T10:11:12-05:00,11,33,64,85,10
```

Other Import and Export Options

You can also use MATLAB, the REST API, or the MQTT API to import and export channel data.

[Read Data](#)  
[Write Data](#)

Figura 20. Configuración del canal en ThingSpeak.  
Fuente: [26]

- Automatización de cálculos intermedios (por ejemplo, promedio móvil de 8 o 24 horas para el cálculo del AQI) mediante scripts en MATLAB integrados en la misma plataforma:



Figura 21. Configuración de valores AQI CO.  
Fuente: [26]

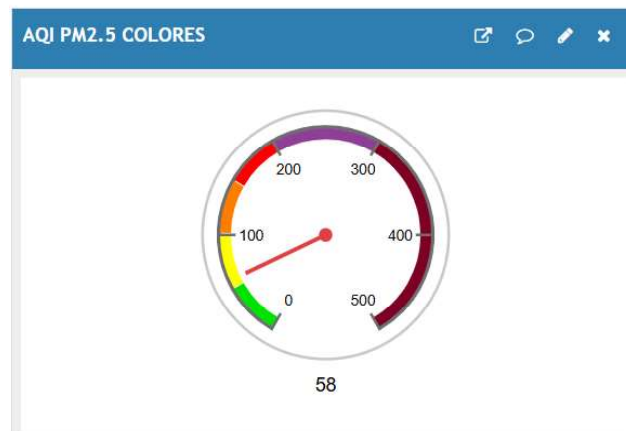


Figura 22. Configuración de valores AQI PM2.5.  
Fuente: [26]

### 6.5 Interacción con otros servicios:

Como parte del diseño abierto, la información de ThingSpeak se integró en una página pública construida en Google Sites, lo que permite que usuarios, investigadores o ciudadanos puedan visualizar los resultados sin conocimientos técnicos:

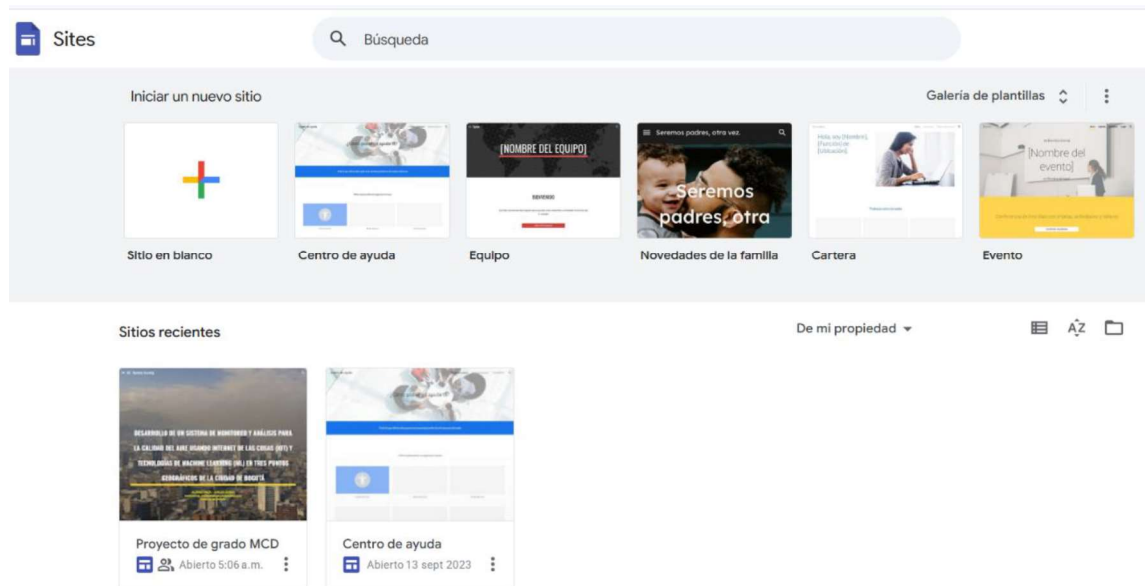


Figura 23. Configuración de Google Sites.  
Fuente: [30]



Figura 24. Configuración de Google Sites – Diseño de interfaces de usuario.  
Fuente: [30]

También se inició la automatización de alertas con Telegram. Los scripts evalúan si los contaminantes superan ciertos umbrales y, en tal caso, se envía una notificación a un canal de Telegram configurado con la API oficial.



Inicia sesión con código QR

- 1 Abre Telegram en tu teléfono
- 2 Ajustes > Dispositivos > Vincular un dispositivo
- 3 Escanea esta imagen para iniciar sesión

[INICIAR SESIÓN CON NÚMERO DE TELÉFONO](#)

Figura 25. Uso de Telegram para envío de Notificaciones.

Fuente: [31]

## 6.6 Resultados obtenidos

- Transmisión estable de datos desde sensores cada 5 minutos.
- Visualización pública en Google Sites, accesible en cualquier momento.
- Recopilación automática de históricos.
- Base consolidada para el análisis posterior en Python (modelos de predicción).
- Sistema flexible, escalable y replicable.

## 7. ACTIVIDADES Y RESULTADOS OBTENIDOS OBJETIVO ESPECÍFICO 3

### Objetivo específico 3:

Aplicar técnicas de ML para analizar los datos recopilados y desarrollar un modelo predictivo para los niveles de CO y PM2.5 en los puntos de monitoreo, entrenando y ajustando los modelos utilizando los datos recopilados, con el objetivo de mejorar su precisión en la predicción de los niveles de contaminación del aire.

### 7.1 Preparación del entorno de análisis

Para este objetivo se utilizó Google Colab como plataforma de trabajo por su compatibilidad con Python, acceso y facilidad para importar datasets desde archivos CSV descargados desde ThingSpeak.



Figura 26. Entorno google colab.

Fuente: [32]

En la fase de preparación del análisis, se estableció un conjunto de bibliotecas y paquetes especializados que proporcionan las herramientas necesarias para el manejo de datos, su transformación y el desarrollo de los modelos predictivos. Se utilizaron bibliotecas como pandas y numpy para la manipulación eficiente de estructuras de datos y operaciones numéricas, mientras que matplotlib facilitó la generación de visualizaciones que permiten interpretar tendencias y patrones en la serie temporal. La biblioteca scikit-learn aportó funciones de escalado y métricas de evaluación, esenciales para estandarizar las variables y medir la precisión del modelo a través de indicadores como el error cuadrático medio y el coeficiente de determinación. Por su parte, TensorFlow y Keras fueron utilizadas para definir y entrenar redes neuronales LSTM, las cuales son idóneas en la predicción de secuencias temporales debido a su capacidad de modelar dependencias a largo plazo.

Este bloque de importación refleja la elección consciente de herramientas ampliamente validadas en la comunidad científica y se justifica en la necesidad de abordar de forma integral el ciclo de

vida de los datos, desde su lectura hasta la generación de predicciones y su posterior interpretación. Se emplearon las siguientes bibliotecas:

- **Pandas y NumPy:** limpieza y manipulación de datos.
- **Matplotlib y Seaborn:** visualización y análisis exploratorio.
- **Scikit-learn:** preprocesamiento, escalamiento, y métricas de evaluación.
- **TensorFlow y Keras:** construcción, entrenamiento y evaluación de modelos LSTM.
- **Keras Tuner:** búsqueda de hiperparámetros óptimos (Random Search).

```
import pandas as pd, numpy as np, math, joblib, matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error, r2_score
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
```

Figura 27. Bibliotecas utilizadas.  
Fuente: elaboración propia.

## 7.2 Estructura de análisis por contaminante y medidor

Se construyeron notebooks individuales para cada contaminante (CO y PM2.5) y para cada punto geográfico (tres medidores), dando como resultado seis notebooks en total. Cada uno contiene:

- Lectura de los datos reales.

	A	B
1	<b>DateTime</b>	<b>CO</b>
2	2025-06-10 11:49:21	0,5
3	2025-06-10 12:06:37	0,76
4	2025-06-10 12:11:25	0,82
5	2025-06-10 12:16:36	0,93
6	2025-06-10 12:21:38	0,89
7	2025-06-10 12:31:48	0,86
8	2025-06-10 12:36:24	0,93
9	2025-06-10 12:46:49	0,76
10	2025-06-10 12:51:36	0,53

Figura 28. Lectura de los datos.  
Fuente: elaboración propia.

- Lectura de datos: En este primer bloque se lee un archivo Excel que contiene lecturas horarias del contaminante, especificando que la columna DateTime debe interpretarse como fechas; a continuación, esa misma columna se convierte en el índice del DataFrame y se ordenan las filas de forma cronológica; posteriormente, se selecciona únicamente la

variable a analizar, se transforman sus valores a tipo flotante y se eliminan posibles datos faltantes mediante dropna; finalmente, los valores resultantes se extraen como un arreglo de NumPy de precisión float32 y se conservan los índices originales en la variable dates para su uso posterior en la generación de gráficos.

```
# ----- 1. Leer archivo y preparar serie
df = (pd.read_excel('/content/CO_Medidor1_Horario.xlsx',
                  parse_dates=['DateTime'])
      .set_index('DateTime')
      .sort_index())

# Filtrar columna CO y eliminar vacíos si existen
series_raw = df['CO'].astype(float).dropna()

# Convertir a arreglo y extraer fechas
series = series_raw.values.astype(np.float32)
dates = series_raw.index
```

Figura 29. Procesamiento y agrupación por promedio horario.  
Fuente: elaboración propia.

- Ventanas supervisadas: Este paso transforma la serie temporal de concentraciones en secuencias de datos que alimentan el modelo. Para cada punto de la serie, se toman las 24 observaciones anteriores como entrada y la siguiente como salida a predecir. Esta técnica permite que el modelo aprenda relaciones y patrones a lo largo del tiempo. Además, los datos se normalizan entre 0 y 1 mediante un escalador para facilitar el entrenamiento y evitar problemas por diferencias de magnitud.

```
# ————— 2. Ventanas supervisadas (24 pasos) —————
TIME_STEPS = 24
def to_supervised(arr, steps=TIME_STEPS):
    X, y = [], []
    for i in range(steps, len(arr)):
        X.append(arr[i-steps:i])
        y.append(arr[i])
    return np.array(X).reshape(-1, steps, 1), np.array(y)

scaler = MinMaxScaler()
scaled = scaler.fit_transform(series.reshape(-1,1)).flatten()
X, y = to_supervised(scaled)
dates_y = dates[TIME_STEPS:]
```

Figura 30. Ventanas supervisadas.  
Fuente: elaboración propia.

- División de datos en entrenamiento (80 %) y prueba (20 %): para el entrenamiento y evaluación del modelo LSTM, se realizó una partición del conjunto de datos en proporción 80/20. Este criterio se fundamenta en las consideraciones de garantizar que el modelo disponga de un volumen adecuado de datos históricos (80%) que permita aprender los patrones estacionales y tendencias de las concentraciones horarias y reservar un 20% de los datos más recientes para la evaluación, asegurando que la predicción se realice sobre un bloque temporal independiente que el modelo nunca ha procesado. Como se ha reportado en [33] la segmentación del conjunto de datos en un 80 % para entrenamiento y un 20 % para prueba se fundamenta en criterios metodológicos ampliamente documentados en la literatura científica sobre aprendizaje profundo con redes LSTM. Esta proporción se ha utilizado en trabajos recientes que abordan problemas de clasificación y predicción de secuencias temporales, como el proyecto de Pacheco Pérez (2025), donde se implementó un modelo LSTM para la clasificación de eventos en subestaciones digitales. En dicho estudio, la partición 80/20 se seleccionó con el objetivo de garantizar una cantidad suficiente de datos para que el modelo aprendiera de manera robusta los patrones temporales de cada clase, mantener un bloque independiente con un volumen adecuado de muestras que permitiera evaluar la capacidad de generalización sin inducir sesgos de representación, asegurar la distribución representativa de las clases y la diversidad de escenarios en ambos subconjuntos. La adopción de esta proporción responde a criterios metodológicos de equilibrio entre precisión estadística, estabilidad en las métricas de desempeño y comparabilidad con investigaciones previas que aplican

enfoques basados en redes LSTM. Además, el uso de una división 80/20 facilita replicar procedimientos y contrastar resultados con estudios afines.

```
# ----- 3. Split 80 % / 20 % -----
split = int(0.8 * len(X))
X_train, X_test = X[:split], X[split:]
y_train, y_test = y[:split], y[split:]
dates_test      = dates_y[split:]
```

Figura 31. Entrenamiento y prueba.  
Fuente: elaboración propia.

- Definición y entrenamiento de un modelo LSTM: En esta etapa se construye una red neuronal de tipo LSTM con dos capas una de 96 y otra de 128 unidades y mecanismos de *dropout* (0.0 y 0.1) que ayudan a prevenir el sobreajuste en este caso para el contaminante CO. El modelo se compila con el optimizador Adam y la función de pérdida de error cuadrático medio, que es adecuada para problemas de regresión. Luego se entrena durante 40 épocas con lotes de 32 muestras y sin mezclar el orden temporal de los datos (*shuffle=False*), reservando el 10% para validación. Este entrenamiento permite que la red aprenda patrones de comportamiento en las series temporales de contaminantes para poder hacer predicciones sobre valores futuros.

```
# ----- 4. Modelo ajustado con hiperparámetros -----
model = Sequential([
    LSTM(96, return_sequences=True, input_shape=(TIME_STEPS,1), activation='tanh'),
    Dropout(0.0),
    LSTM(128, activation='tanh'),
    Dropout(0.1),
    Dense(1)
])
model.compile(optimizer='adam', loss='mse')
history = model.fit(X_train, y_train,
                    epochs=40, batch_size=32,
                    shuffle=False, validation_split=0.1, verbose=1)
```

Figura 32. Modelo.  
Fuente: elaboración propia.

- Evaluación con métricas de regresión: RMSE y R<sup>2</sup>.

```
# ----- 5. Evaluación + Excel de prueba -----
y_pred_scaled = model.predict(X_test).flatten()
y_test_inv = scaler.inverse_transform(y_test.reshape(-1,1))
y_pred_inv = scaler.inverse_transform(y_pred_scaled.reshape(-1,1))

rmse = math.sqrt(mean_squared_error(y_test_inv, y_pred_inv))
r2 = r2_score(y_test_inv, y_pred_inv)
print(f'RMSE: {rmse:.3f} ppm | R2: {r2:.3f}')

pd.DataFrame({
    'DateTime': dates_test,
    'CO_Real': y_test_inv.flatten(),
    'CO_Pred': y_pred_inv.flatten()
}).to_excel('predicciones_co_junio.xlsx', index=False)
```

Figura 33. Evaluación de métricas.  
Fuente: elaboración propia.

Para cuantificar el desempeño predictivo del modelo LSTM, se seleccionaron como métricas principales la raíz del error cuadrático medio (RMSE) y el coeficiente de determinación (R<sup>2</sup>). Esta decisión se fundamenta en su capacidad para reflejar, de manera complementaria, tanto la magnitud promedio del error como la proporción de variabilidad explicada por el modelo. La RMSE es particularmente útil, ya que penaliza con mayor severidad los errores grandes, permitiendo expresar la precisión del modelo en las mismas unidades de la variable objetivo, facilitando su interpretación operativa. La fórmula utilizada es:

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (A_t - F_t)^2}$$

Figura 34. Fórmula raíz del error cuadrático medio (RMSE).  
Fuente: [34].

Donde  $A_t$  es el valor actual o real en el tiempo,  $F_t$  es la predicción o estimación en el tiempo  $t$  y  $n$  es el número total de observaciones.

Por su parte, el coeficiente de determinación R<sup>2</sup> proporciona una medida relativa de la calidad del ajuste, indicando qué porcentaje de la variabilidad de los datos es explicado por el modelo. Su fórmula es:

$$R^2 = 1 - \frac{\sum_{t=1}^n (At - Ft)^2}{\sum_{t=1}^n (At - \bar{A})^2}$$

Figura 35. Fórmula coeficiente de determinación ( $R^2$ ).  
Fuente: [34].

Donde  $At$  es el valor actual o real en el tiempo  $t$ ,  $Ft$  es la predicción o estimación en el tiempo  $t$ ,  $\bar{A}$  es el promedio de los valores reales  $At$  y  $n$  es el número total de observaciones. Estas métricas han sido utilizadas en proyectos similares, como el trabajo de Hernández Hernández y Forero Criollo, donde se aplicaron en un entorno de predicción de series temporales con redes neuronales profundas, destacando su pertinencia para medir la precisión y capacidad explicativa del modelo [34].

- Resultados del entrenamiento y evaluación: En esta parte se muestran los registros del proceso de entrenamiento de la red LSTM durante 40 épocas. Por cada época, se reporta el error cuadrático medio sobre los datos de entrenamiento y el error sobre los datos de validación, que permiten monitorear si el modelo está aprendiendo de forma estable y sin sobreajustar. Al finalizar, se calculan dos métricas clave: el RMSE, que fue de 0.038 ppm, indicando que en promedio el error de predicción es muy bajo respecto a la magnitud de las concentraciones, y un  $R^2$  de 0.981, que refleja que el modelo logra explicar el 98,1% de la variabilidad observada, lo que se interpreta como un desempeño excelente. Además, se calcula el promedio de 8 horas más reciente (3.18 ppm) y su conversión a AQI, que es 35, correspondiente a la categoría Good en color verde. Este bloque es fundamental porque permite valorar la capacidad predictiva del modelo con indicadores objetivos y verificar que los resultados sean consistentes con lo esperado en la dinámica ambiental.

```

Epoch 34/40
49/49 ██████████ 4s 82ms/step - loss: 0.0020 - val_loss: 5.2382e-04
Epoch 35/40
49/49 ██████████ 4s 59ms/step - loss: 0.0016 - val_loss: 5.1206e-04
Epoch 36/40
49/49 ██████████ 5s 54ms/step - loss: 0.0019 - val_loss: 4.9958e-04
Epoch 37/40
49/49 ██████████ 5s 99ms/step - loss: 0.0012 - val_loss: 7.4365e-04
Epoch 38/40
49/49 ██████████ 3s 51ms/step - loss: 0.0022 - val_loss: 6.2726e-04
Epoch 39/40
49/49 ██████████ 5s 51ms/step - loss: 0.0013 - val_loss: 5.3787e-04
Epoch 40/40
49/49 ██████████ 4s 76ms/step - loss: 0.0017 - val_loss: 4.7870e-04
14/14 ██████████ 1s 38ms/step
RMSE: 0.038 ppm | R2: 0.981
Promedio 8 h más reciente: 3.18 ppm

```

AQI CO (8 h): 35  
 Categoría: Bueno (Green)  
 Acción: Es un buen día para realizar actividades al aire libre.  
 Grupos sensibles: Ninguno

Figura 36. Épocas y evaluación de métricas.  
 Fuente: elaboración propia.

- Promedio móvil sobre la serie horaria 8h (CO)/24h (PM2.5): Este bloque de código calcula el promedio móvil de las concentraciones en las últimas 8 horas, un criterio fundamental en normativa ambiental (por ejemplo, en la evaluación de calidad del aire con base en el AQI). Primero, convierte la serie horaria de concentraciones en un objeto Series con fechas como índice. Luego por cada hora, calcula la media de los 8 valores anteriores (equivalente a un promedio ponderado de 8 horas). La función dropna elimina los valores iniciales que quedan vacíos porque no había suficientes datos al principio. Finalmente se selecciona el último promedio disponible, que corresponde al valor más reciente de 8 horas de monitoreo. Este indicador es clave porque es el que se utiliza para clasificar la calidad del aire en tiempo real y emitir alertas si se supera algún umbral regulatorio.

```

# ----- 6. Promedio móvil 8 h sobre la serie horaria -----
co_8h_series = pd.Series(series, index=dates).rolling(8).mean().dropna()
co_8h_avg     = co_8h_series.iloc[-1]           # el promedio 8 h más reciente
print(f'Promedio 8 h más reciente: {co_8h_avg:.2f} ppm')

```

Figura 37. Promedio móvil.  
 Fuente: elaboración propia.

- Break-points detallados (EPA): este bloque detalla los rangos de concentración del contaminante CO definidos por la EPA, asociando cada tramo con su categoría AQI, color,

acción recomendada y grupo sensible afectado. Esta estructura permite generar alertas más precisas, comprensibles y orientadas a la salud pública, reforzando la utilidad del sistema de monitoreo para la toma de decisiones.

```
# ----- 7. Break-points detallados (EPA) -----
breakpoints_co = [
  {"conc_lo": 0.0, "conc_hi": 4.4, "aqi_lo": 0, "aqi_hi": 50,
   "category": "Good", "color": "Green",
   "action": "La calidad del aire es satisfactoria. No se requiere acción.",
   "sensitive": "Ninguno"},
  {"conc_lo": 4.5, "conc_hi": 9.4, "aqi_lo": 51, "aqi_hi": 100,
   "category": "Moderate", "color": "Yellow",
   "action": "La calidad del aire es aceptable; algunas personas sensibles pueden presentar molestias leves.",
   "sensitive": "Niños, adultos mayores y personas con afecciones respiratorias"},
  {"conc_lo": 9.5, "conc_hi": 12.4, "aqi_lo": 101, "aqi_hi": 150,
   "category": "Unhealthy for Sensitive Groups", "color": "Orange",
   "action": "Los grupos sensibles pueden presentar efectos en la salud.",
   "sensitive": "Personas con enfermedades cardíacas o respiratorias"},
  {"conc_lo": 12.5, "conc_hi": 15.4, "aqi_lo": 151, "aqi_hi": 200,
   "category": "Unhealthy", "color": "Red",
   "action": "Toda la población puede experimentar efectos; reducir actividad física al aire libre.",
   "sensitive": "Personas con enfermedades cardiopulmonares, adultos mayores, niños"},
  {"conc_lo": 15.5, "conc_hi": 30.4, "aqi_lo": 201, "aqi_hi": 300,
   "category": "Very Unhealthy", "color": "Purple",
   "action": "Condiciones de emergencia; evitar actividad al aire libre, especialmente grupos sensibles.",
   "sensitive": "Todos, en especial personas con afecciones previas"},
  {"conc_lo": 30.5, "conc_hi": 50.4, "aqi_lo": 301, "aqi_hi": 400,
   "category": "Hazardous", "color": "Maroon",
   "action": "Alerta de salud; tomar medidas inmediatas y evitar la exposición al aire libre.",
   "sensitive": "Toda la población"}
]
```

Figura 38. Break-points detallados (EPA).  
Fuente: elaboración propia.

```
def compute_co_aqi(co_avg):
    c = math.floor(co_avg*10)/10 # trunca a 1 decimal
    for bp in breakpoints_co:
        if bp["conc_lo"] <= c <= bp["conc_hi"]:
            aqi = ((bp["aqi_hi"] - bp["aqi_lo"])
                   / (bp["conc_hi"] - bp["conc_lo"])) * (c - bp["conc_lo"]) + bp["aqi_lo"]
            return round(aqi), bp
    # fuera de rango alto
    bp = breakpoints_co[-1]
    aqi = 500
    return aqi, bp

aqi_val, bp_sel = compute_co_aqi(co_8h_avg)
print(f"\nAQI para CO (promedio 8-hr): {aqi_val}")
print(f"Categoría: {bp_sel['category']}")
print(f"Color representativo: {bp_sel['color']}")
print(f"Acción recomendada: {bp_sel['action']}")
print(f"Grupos sensibles afectados: {bp_sel['sensitive']}")
```

Figura 39. Break-points detallados (EPA).  
Fuente: elaboración propia.

- Barra de color AQI (con recomendación):

```
# ----- 8. Barra de color AQI -----
plt.figure(figsize=(10,1))
plt.barh(0,100, color=bp_sel['color'].lower(), edgecolor='black')
plt.text(50,0, f"{bp_sel['category']} (AQI {aqi_val2})",
        ha='center', va='center', color='white', fontsize=12)
plt.xlim(0,100); plt.yticks([]); plt.xlabel('AQI')
plt.title('Calidad del aire (CO, 8 h)'); plt.show()
```

Figura 40. Barra de color AQI (con recomendación).  
Fuente: elaboración propia.

AQI CO (8 h): 35  
Categoría: Bueno (Green)  
Acción: Es un buen día para realizar actividades al aire libre.  
Grupos sensibles: Ninguno



Figura 41. Barra de color AQI (con recomendación).  
Fuente: elaboración propia.

- Visualizaciones:

```
# ----- 9. Gráficos de desempeño -----
plt.figure(figsize=(12,6))
plt.plot(dates_test, y_test_inv, label='CO Real')
plt.plot(dates_test, y_pred_inv, '--', label='CO Pred')
plt.legend(); plt.xticks(rotation=45); plt.tight_layout(); plt.show()

plt.figure(figsize=(8,4))
plt.plot(history.history['loss'], label='Train')
plt.plot(history.history['val_loss'], label='Val')
plt.title('Pérdida'); plt.legend(); plt.show()

plt.figure(figsize=(6,6))
plt.scatter(y_test_inv, y_pred_inv, alpha=0.5)
m, M = y_test_inv.min(), y_test_inv.max()
plt.plot([m,M],[m,M], 'r--'); plt.xlabel('Real'); plt.ylabel('Pred')
plt.title('Dispersión'); plt.show()
```

Figura 42. Visualizaciones.  
Fuente: elaboración propia.

- Gráfico real vs predicción: esta gráfica compara las concentraciones reales y predichas del contaminante, mostrando visualmente la capacidad del modelo para ajustarse a los valores observados. La cercanía entre ambas líneas indica un buen desempeño predictivo, lo cual refuerza la fiabilidad del modelo para estimaciones futuras en el monitoreo de la calidad del aire.

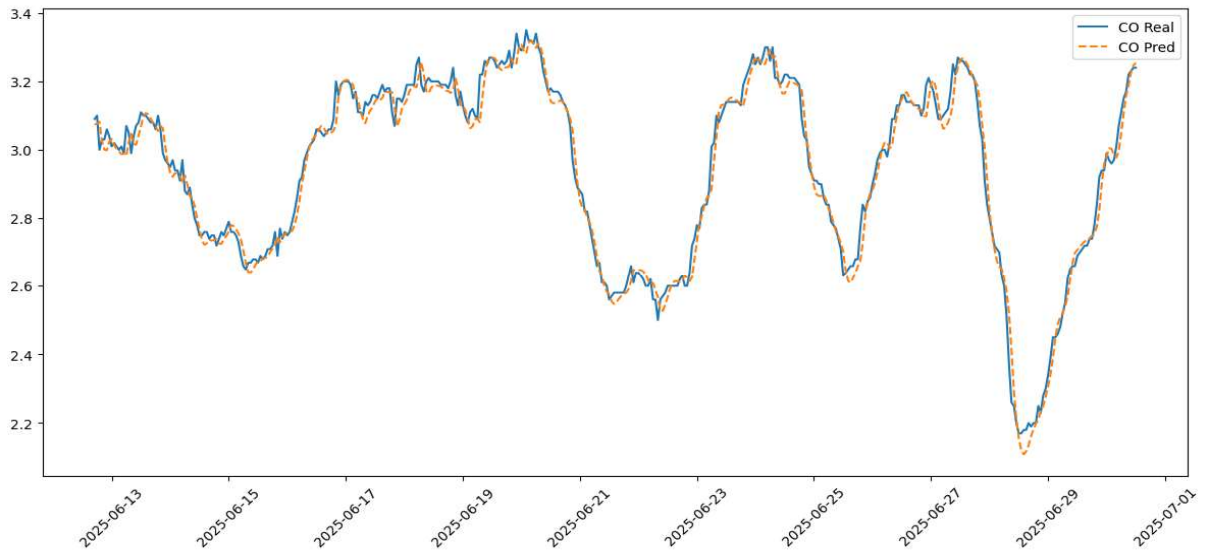


Figura 43. Gráfico real vs predicción.  
Fuente: elaboración propia.

- Pérdida: esta gráfica de pérdida muestra la evolución del error durante el entrenamiento del modelo para los conjuntos de entrenamiento (Train) y validación (Val). Una disminución progresiva y estable en ambas curvas indica que el modelo está aprendiendo adecuadamente sin sobreajustarse. La baja diferencia entre ambas líneas sugiere buena generalización, lo cual valida la eficacia del modelo en datos no vistos.

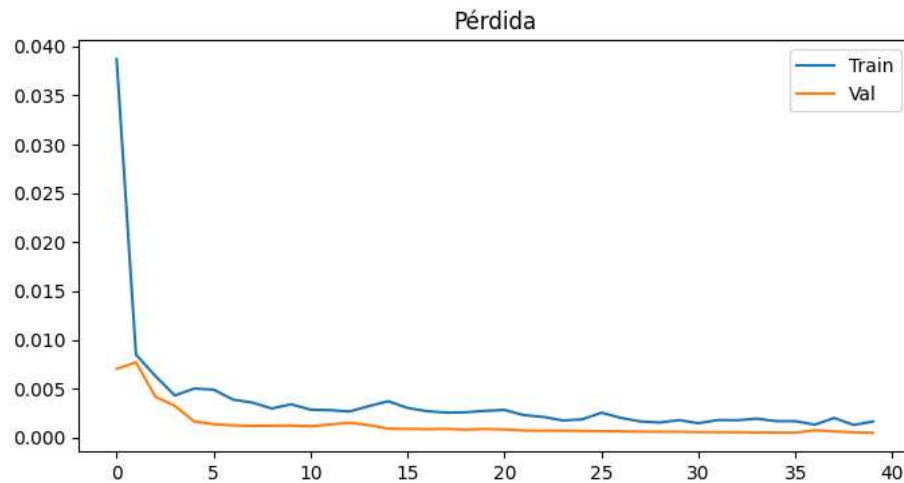


Figura 44. Pérdida.  
Fuente: elaboración propia.

- **Dispersión:** esta gráfica de dispersión compara los valores reales con los valores predichos por el modelo. La proximidad de los puntos a la línea roja de referencia ( $y = x$ ) indica una alta precisión en las predicciones. Su alineación sugiere que el modelo logra capturar correctamente la relación temporal del contaminante, lo cual valida su capacidad de generalización sobre los datos de entrada.



Figura 45. Dispersión.  
Fuente: elaboración propia.

	A	B	C
1	<b>DateTime</b>	<b>CO_Real</b>	<b>CO_Pred</b>
74	2025-06-25 06:25:00	0,300000012	0,316878676
75	2025-06-25 06:30:00	0,308333337	0,316878676
76	2025-06-25 06:35:00	0,316666663	0,320856005
77	2025-06-25 06:40:00	0,324999988	0,329286695
78	2025-06-25 06:45:00	0,333333343	0,340655535
79	2025-06-25 06:50:00	0,341666669	0,352868706
80	2025-06-25 06:55:00	0,349999994	0,364243478
81	2025-06-25 07:00:00	0,358333319	0,373947829
82	2025-06-25 07:05:00	0,366666675	0,381977767
83	2025-06-25 07:10:00	0,375	0,388866961
84	2025-06-25 07:15:00	0,383333325	0,395328671

Figura 46. Datos reales vs predicción.  
Fuente: elaboración propia.

### 7.3 Exploración de datos (EDA)

Previo al desarrollo de los modelos predictivos, se realizó un análisis exploratorio de datos (EDA) con el fin de comprender la distribución, estacionalidad y calidad de los registros de concentración de los contaminantes. Este análisis permitió identificar patrones temporales, posibles anomalías y confirmar que los datos eran representativos del fenómeno que se deseaba modelar. Los criterios empleados y los hallazgos obtenidos se detallan a continuación.

- **Estructura de los datos:** la exploración inicial de los datos recolectados permitió identificar su estructura, volumen y principales características estadísticas, lo que resulta esencial para comprender la naturaleza de la información antes de su procesamiento y modelado. El conjunto de registros está compuesto para el contaminante CO del medidor 1 con 26.208 observaciones con una frecuencia de muestreo de cinco minutos durante tres meses, todas correspondientes a una única variable cuantitativa, almacenada en formato numérico de punto flotante. A través del resumen estadístico se evidenció que las concentraciones presentan un rango que oscila entre valores mínimos de 0,1 y máximos cercanos a 18,75, con una media alrededor de 2,9 y una desviación estándar aproximada de 2,24, indicando la existencia de variabilidad en los niveles observados. Asimismo, se verificó que existen valores faltantes en el periodo analizado, en donde se aplica la técnica de interpolación lineal para rellenar estos datos faltantes, lo que garantiza la integridad y continuidad de la serie temporal. Esta información general constituye un insumo clave para contextualizar las visualizaciones y sustentar las decisiones metodológicas adoptadas en el proceso de modelado predictivo.

```

⇒ Estructura y tipos:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26208 entries, 0 to 26207
Data columns (total 2 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   DateTime    26208 non-null  datetime64[ns]
1   CO          26156 non-null  float64
dtypes: datetime64[ns](1), float64(1)
memory usage: 409.6 KB
None

Primeras 5 filas:

           DateTime      CO
0  2025-04-01 00:00:00  0.94
1  2025-04-01 00:05:00  0.25
2  2025-04-01 00:10:00  0.10
3  2025-04-01 00:15:00  0.61
4  2025-04-01 00:20:00  0.81

Resumen estadístico de columnas numéricas:

                DateTime      CO
count                26208  26156.000000
mean  2025-05-16 11:57:30.000000  2.935118
min    2025-04-01 00:00:00      0.100000

```

Figura 47. Estructura de los datos.  
Fuente: elaboración propia.



Figura 48. Datos faltantes.  
Fuente: elaboración propia.

- **Serie temporal completa:** la primera visualización implementada, la serie temporal, permitió representar la evolución de las mediciones a lo largo del periodo analizado y evidenciar fluctuaciones, picos y periodos de relativa estabilidad. Esta representación es particularmente útil para identificar tendencias generales, detectar días con niveles elevados y observar si existen periodos prolongados de valores bajos que puedan relacionarse con condiciones meteorológicas favorables o disminución temporal de fuentes emisoras.
- **Histogramas con KDE:** la segunda visualización, correspondiente al histograma con estimación de densidad, tuvo como propósito describir la distribución de frecuencias de

los valores registrados y evidenciar si los datos presentan una distribución simétrica, sesgada o multimodal. Esta gráfica contribuye a determinar si los valores extremos constituyen eventos aislados o si forman parte de un patrón recurrente, lo que resulta fundamental para valorar el impacto de dichos eventos en la calibración y evaluación de los modelos predictivos.

- **Boxplots por hora del día:** el boxplot por hora del día aportó una visión detallada de la variabilidad intra-diaria y la dispersión de las concentraciones por cada franja horaria, permitiendo identificar franjas con mayor mediana y rango intercuartílico y, al mismo tiempo, observar la presencia de valores atípicos que pueden corresponder a emisiones puntuales. Esta información respalda la hipótesis de que los contaminantes presentan ciclos diarios asociados con la actividad humana, lo cual es clave para ajustar modelos que contemplen estacionalidades horarias.
- **Análisis de componentes estacionales y tendencia en la serie:** la descomposición aditiva de la serie de CO revela un comportamiento compuesto por una fuerte estacionalidad diaria, una tendencia moderadamente variable a lo largo del tiempo y un componente residual aleatorio sin patrones evidentes. Esto indica que gran parte de la variabilidad puede explicarse por ciclos diarios y cambios suaves en el nivel general de contaminación, lo cual es útil para el diseño de modelos predictivos más precisos.

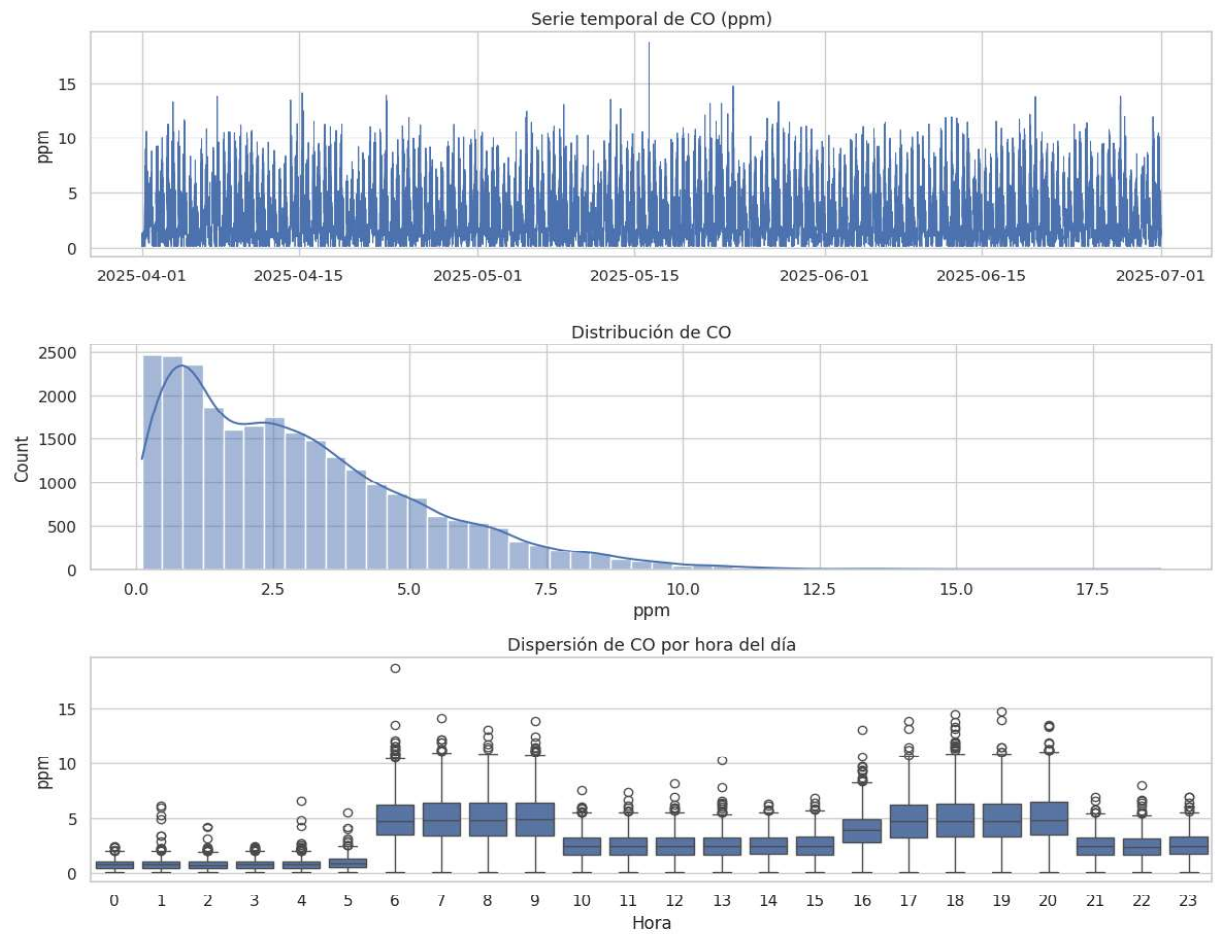


Figura 49. Histogramas y boxplots.  
Fuente: elaboración propia.

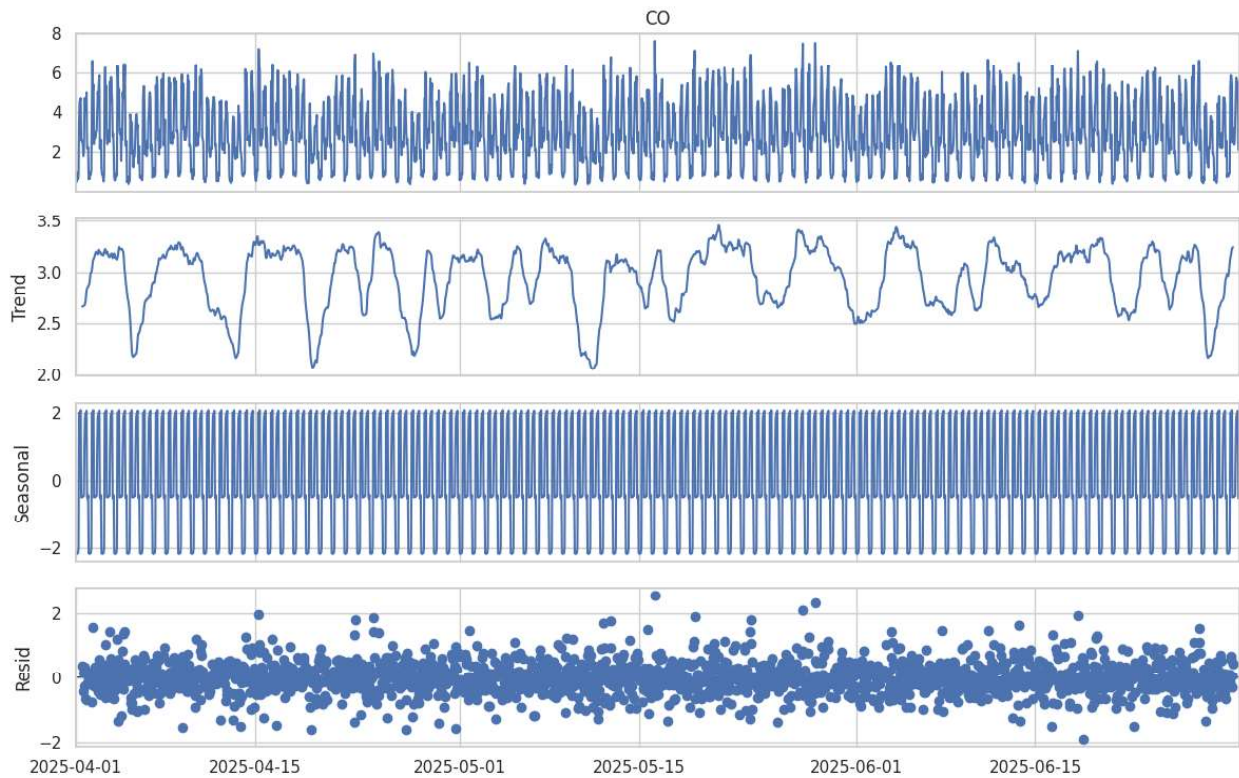


Figura 50. Descomposición de la serie.  
Fuente: elaboración propia.

- **Curvas de tendencia semanal con ventana móvil (rolling de 7 días):** la tendencia suavizada mediante la media móvil semanal aportó un resumen de la evolución de los niveles medios a lo largo del tiempo, eliminando las fluctuaciones de corto plazo y facilitando la identificación de cambios persistentes que pueden deberse a modificaciones en las condiciones meteorológicas, cambios en las fuentes de emisión o procesos de acumulación y dispersión del contaminante.



Figura 51. Curvas de tendencia semanal.  
Fuente: elaboración propia.

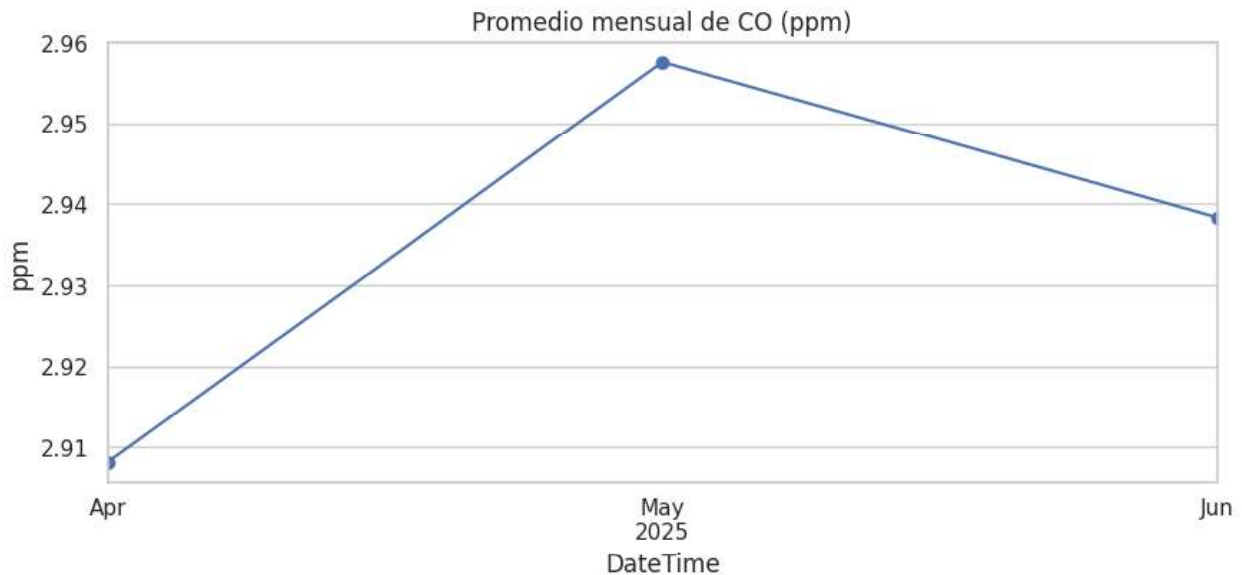


Figura 52. Promedio mensual.  
Fuente: elaboración propia.

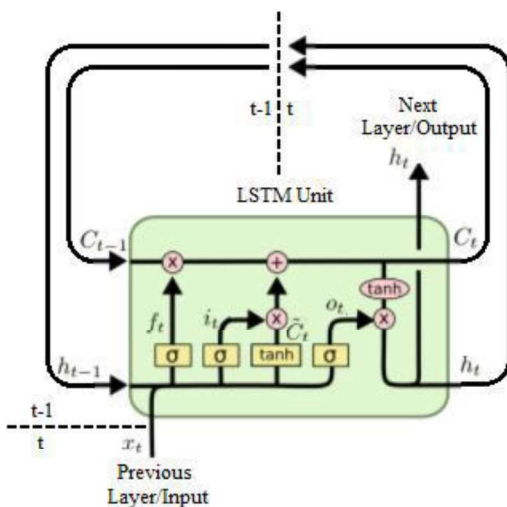
El conjunto de estos análisis visuales y estadísticos permite concluir que los datos presentan una variabilidad esperada en entornos urbanos y que las observaciones extremas se encuentran dentro de rangos plausibles, reforzando la confiabilidad de los registros para su utilización en el entrenamiento de modelos de aprendizaje automático. Asimismo, estos resultados sustentan de manera objetiva los criterios aplicados en las decisiones metodológicas, ya que cada visualización aportó información complementaria que hizo posible comprender la naturaleza estacional, la variabilidad y la distribución de los valores observados, elementos indispensables para cumplir con el objetivo de desarrollar modelos predictivos.

## 7.4 Selección del modelo predictivo con LSTM

Los modelos implementados fueron redes neuronales LSTM (Long Short-Term Memory) ideales para series temporales debido a su capacidad de mantener información contextual en secuencias de datos para la predicción de los contaminantes monóxido de carbono (CO) y material particulado fino (PM2.5). Esta decisión se fundamenta en la necesidad de capturar de forma precisa la dinámica temporal de estos contaminantes, la cual está influenciada por factores como el tráfico urbano, la ubicación geográfica y las condiciones meteorológicas [35].

### 7.4.1 Long Short-Term Memory (LSTM)

Las redes neuronales de tipo LSTM fueron introducidas por Sepp Hochreiter y Jürgen Schmidhuber en 1997 con el objetivo de resolver las dificultades de desvanecimiento y explosión del gradiente en modelos recurrentes [35]. A diferencia de una RNN convencional, una unidad LSTM dispone de un canal de memoria ( $C_t$ ) y de tres compuertas de control: olvido ( $f_t$ ), entrada ( $i_t$ ), salida ( $o_t$ ) que regulan dinámicamente qué información descartar, qué datos incorporar y qué resultados emitir.



$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

Figura 53. Esquema interno de una celda LSTM

Fuente: [19]

La puerta de olvido ( $f_t$ ) evalúa el par  $[h_{t-1}, x_t]$  y decide la fracción de la información previa  $C_{t-1}$  que se conserva. La puerta de entrada ( $i_t$ ) filtra la nueva propuesta de contenido  $\tilde{C}_t$ , generada mediante una función  $\tanh$ , determinando cuánta parte de ese candidato se añadirá al estado. El estado de la celda ( $C_t$ ) se actualiza como suma ponderada de la memoria antigua y de la información entrante:  $C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t$ . Finalmente, la puerta de salida ( $o_t$ ) controla la porción de  $C_t$  que se expone como salida oculta ( $h_t$ ):  $h_t = O_t \times \tanh(C_t)$ . Gracias a este diseño, las LSTM pueden retener o descartar información de forma selectiva, lo que les permite capturar dependencias temporales extensas sin sufrir problemas de desvanecimiento o explosión del gradiente.

## 7.5 Elección de los Hiperparámetros

La técnica Random Search fue seleccionada para la optimización de hiperparámetros debido a su reconocida eficiencia en espacios de búsqueda de alta dimensión, donde explorar de manera exhaustiva todas las combinaciones posibles resulta computacionalmente costoso. A diferencia de la búsqueda en malla (Grid Search), que evalúa sistemáticamente combinaciones fijas, Random Search permite una exploración más diversa y eficaz del espacio de soluciones, aumentando la probabilidad de encontrar configuraciones óptimas con menos evaluaciones [36].

La arquitectura se ajustó automáticamente con Keras Tuner (Random Search), optimizando:

- time\_steps (número de pasos anteriores)
- número de neuronas por capa
- tasa de dropout

```
Mejores hiperparámetros encontrados:  
time_steps: 24  
units_1: 96  
activation: tanh  
dropout_1: 0.0  
units_2: 128  
dropout_2: 0.1  
optimizer: adam
```

Figura 54. Hiperparámetros.  
Fuente: elaboración propia.

Se utilizó el optimizador Adam por su eficiencia en problemas con gradientes ruidosos y su buen rendimiento en comparación con SGD y RMSProp.

## 7.6 Interpolación lineal para rellenar datos faltantes

Para rellenar los datos faltantes se utilizó la técnica de interpolación lineal que es uno de los métodos más simples de interpolación y es usado a menudo como una técnica para llenar datos vacíos; el método consiste en conectar puntos con valores conocidos mediante líneas rectas y estima los valores faltantes en base a estas conexiones [37].

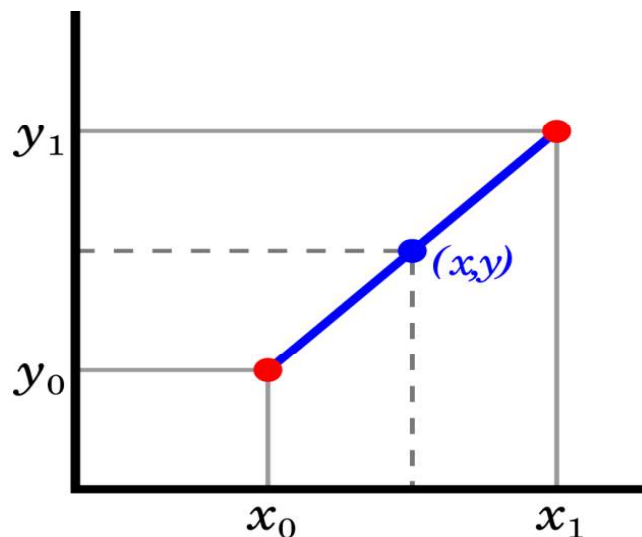


Figura 55. Interpolación lineal entre dos puntos.

Fuente: [37]

Se identifican los lugares donde faltan datos en la variable CO, y se crean nuevos valores en esos huecos, como si trazáramos una línea entre el punto anterior y el siguiente con valor.

Matemáticamente, si los valores conocidos son  $(t_0, y_0)$  y  $(t_1, y_1)$  y falta el valor en un tiempo intermedio  $(t)$ , el valor estimado es:

$$y = y_0 + (x - x_0) \frac{y_1 - y_0}{x_1 - x_0}$$

Figura 56. ecuación de la interpolación lineal.

Fuente: [37].

## 7.7 Resultados por medidor y contaminante

### Medidor 1 – CO:

- Hiperparámetros: time\_steps=24, units\_1=96, units\_2=128, dropout\_1=0.0, dropout\_2=0.1.
- EDA: 26208 datos.
- Desempeño: RMSE: 0.038 ppm | R<sup>2</sup>: 0.981.
- Promedio 8h más reciente: 3.18 ppm.

### Medidor 2 – CO:

- Hiperparámetros: time\_steps=24, units\_1=96, units\_2=128, dropout\_1=0.0, dropout\_2=0.1.
- EDA: 26208 datos.
- RMSE: 0.034 ppm |  $R^2$ : 0.897.
- Promedio 8h: 2.74 ppm.

### Medidor 3 – CO:

- Hiperparámetros: time\_steps=24, units\_1=96, units\_2=128, dropout\_1=0.0, dropout\_2=0.1.
- EDA: 26208 datos.
- RMSE: 0.060 ppm |  $R^2$ : 0.919.
- Promedio 8h: 3.63 ppm.

### Medidor 1 – PM2.5:

- Hiperparámetros: time\_steps=24, units\_1=64, units\_2=96, dropout\_1=0.1, dropout\_2=0.0.
- EDA: 25921 datos.
- RMSE: 0.120  $\mu\text{g}/\text{m}^3$  |  $R^2$ : 0.867.
- Promedio 24h: 15.71  $\mu\text{g}/\text{m}^3$ .

### Medidor 2 – PM2.5:

- Hiperparámetros: time\_steps=24, units\_1=64, units\_2=96, dropout\_1=0.1, dropout\_2=0.0.
- EDA: 25921 datos.
- RMSE: 0.175  $\mu\text{g}/\text{m}^3$  |  $R^2$ : 0.897.
- Promedio 24h: 21.54  $\mu\text{g}/\text{m}^3$ .

### Medidor 3 – PM2.5:

- Hiperparámetros: time\_steps=24, units\_1=64, units\_2=96, dropout\_1=0.1, dropout\_2=0.0.
- EDA: 25921 datos.
- RMSE: 0.263  $\mu\text{g}/\text{m}^3$  |  $R^2$ : 0.917.
- Promedio 24h: 26.48  $\mu\text{g}/\text{m}^3$ .

### **7.8 Conclusión del objetivo**

Los modelos obtenidos presentaron un nivel de precisión adecuado, especialmente considerando el uso de sensores de bajo costo. Los resultados validan la factibilidad de implementar sistemas predictivos de calidad del aire en tiempo real con herramientas accesibles, reforzando la utilidad del enfoque IoT + ML en contextos urbanos como Bogotá.

## 8. ACTIVIDADES Y RESULTADOS OBTENIDOS OBJETIVO ESPECÍFICO 4

### Objetivo específico 4:

Diseñar una interfaz de usuario que muestre los resultados del monitoreo, visualice los niveles de CO y PM2.5 y proporcione información sobre la calidad del aire en los puntos de monitoreo, estableciendo límites aceptables basados en regulaciones ambientales e implementando alertas automáticas cuando los niveles superen dichos límites.

### 8.1 Desarrollo de la interfaz de usuario

Se diseñó una interfaz accesible y visualmente intuitiva mediante la plataforma Google Sites, donde se integraron múltiples elementos del sistema de monitoreo. Esta página pública permite visualizar los niveles de contaminación atmosférica en tiempo real para cada uno de los tres medidores desplegados en Bogotá.

La interfaz fue desarrollada con enfoque responsive para adaptarse a dispositivos móviles y de escritorio, incluyendo:

- Gráficas en tiempo real provenientes de ThingSpeak, embebidas mediante iframes.

### Incorporar desde la Web

```
<iframe width="450" height="260" style="border: 1px solid #cccccc;" src="https://thingspeak.com/channels/2986083/charts/1?bgcolor=%23ffffff&color=%23d62020&dynamic=true&results=60&title=CO+%28ppm%29+UCompensar+%28Medidor+1%29&type=line"></iframe>
```

Pega el código HTML del sitio que deseas incorporar.

Cancelar

Siguiente

Figura 57. Gráficas en tiempo real provenientes de ThingSpeak, embebidas mediante iframes.

Fuente: [30]

## Datos Zona Ucompensar Teusaquillo (Medidor 1)

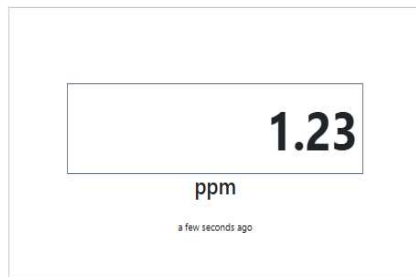
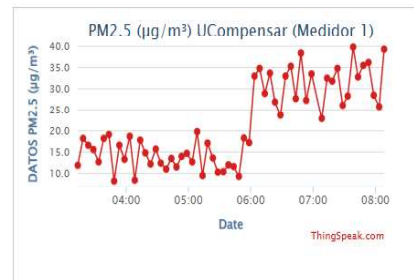
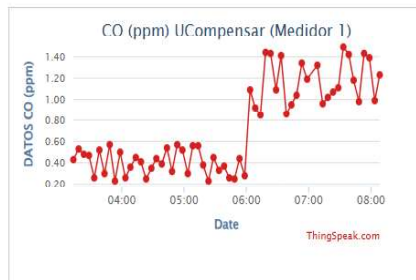
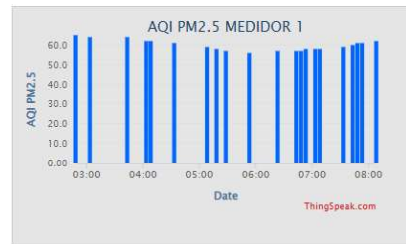
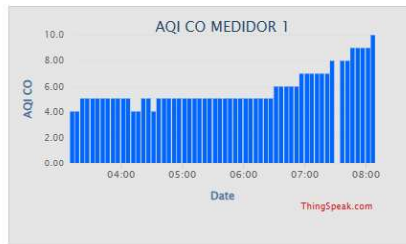


Figura 58. Gráficas en tiempo real.  
Fuente: [30]

- Indicadores de AQI con sistema de colores y números.

### AQI MEDIDOR 1



VALOR AQI CO

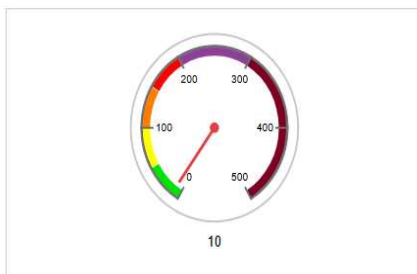


VALOR AQI PM2.5



Figura 59. Gráficas AQI.  
Fuente: [30]

COLOR AQI CO



COLOR AQI PM2.5

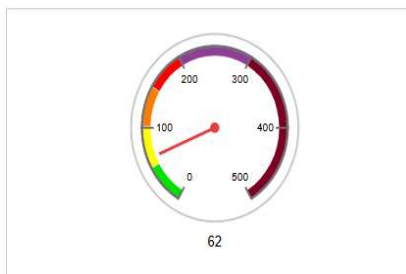


Figura 60. Gráficas AQI colores.  
Fuente: [30]

Esta interfaz está conectada directamente a la base de datos de ThingSpeak, lo que garantiza

actualizaciones automáticas cada cinco minutos sin necesidad de intervención manual.

## **8.2 Visualización de datos y conexión con ThingSpeak**

Cada uno de los tres medidores transmite datos hacia canales individuales en ThingSpeak, una plataforma en la nube especializada en proyectos de IoT. Allí se realiza:

- La visualización gráfica de las series de tiempo para CO y PM2.5.
- El cálculo de promedios móviles, como el de 8 horas para CO y 24 horas para PM2.5.
- El cálculo del Índice de Calidad del Aire (AQI) usando scripts MATLAB integrados en ThingSpeak (dentro de la pestaña "MATLAB Analysis").

Gracias a esta funcionalidad, se pueden mostrar tanto los niveles actuales de contaminación como una clasificación cualitativa del aire (Buena, Moderada, Dañina, etc.), basada en estándares de la EPA.

## **8.3 Integración de resultados del modelo predictivo LSTM**

Los datos recopilados fueron descargados como archivos CSV desde ThingSpeak, para su posterior análisis en Python utilizando notebooks de Google Colab.

Por cada medidor y contaminante, se realizó:

- Un análisis exploratorio de datos (EDA).
- El ajuste de hiperparámetros mediante Keras Tuner con estrategia Random Search.
- La implementación de un modelo LSTM entrenado por canal y contaminante.

Los resultados se reflejan en la interfaz mediante gráficos comparativos entre predicciones y valores reales, acompañados de métricas como RMSE y  $R^2$ .

## **8.4 Alertas automáticas con Telegram**

Se implementó un sistema de alertas tempranas usando la aplicación de mensajería Telegram, aprovechando las herramientas nativas de ThingSpeak sin necesidad de programación externa.



Figura 61. Alertas automáticas con Telegram.  
Fuente: [31]

El flujo fue configurado de la siguiente manera:

1. TimeControl ejecuta revisiones periódicas cada 5 minutos.
2. React evalúa si los datos en Field 1 (CO) o Field 2 (PM2.5) superan un umbral preestablecido, definido con base en los rangos de la Air Quality Index (AQI) de la EPA (2024) [28], los cuales indican niveles de riesgo para la salud:
  - CO: Alerta si supera 4.5 ppm, umbral que corresponde al inicio del rango Moderado (AQI 51–100). Este nivel implica que la calidad del aire ya no es considerada “Buena”

- y puede afectar a personas especialmente sensibles, como aquellas con enfermedades cardíacas.
- PM2.5: Alerta si supera  $9.1 \mu\text{g}/\text{m}^3$  también dentro del rango Moderado, en el cual se recomienda precaución en grupos vulnerables.
- Estos umbrales fueron seleccionados considerando que valores superiores pueden representar riesgos incrementales de exposición, de acuerdo con los Breakpoints for the AQI definidos por la U.S. Environmental Protection Agency [28].
3. Si se activa la condición, se dispara un ThingHTTP configurado con un Webhook de Telegram.
  4. El mensaje generado llega automáticamente a un grupo específico de Telegram:
    - Medidor 1 UCompensar CO alto: Supera los: 4.5 ppm.
    - Medidor 2 UManuela PM2.5 alto: Supera los:  $9.1 \mu\text{g}/\text{m}^3$ .



Figura 62. Generación de Alertas.  
Fuente: [31]

Este grupo puede incluir a cualquier persona interesada, como estudiantes, docentes o autoridades ambientales. Esta estrategia permite alertar de forma rápida, gratuita y automatizada, ayudando a tomar medidas preventivas.

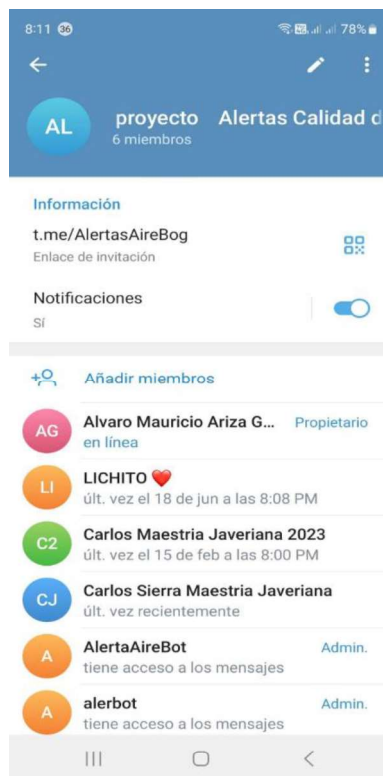


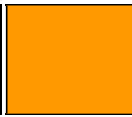
Figura 63. Grupo de alertas.  
Fuente: [31]

## 8.5 Conclusión del objetivo

La interfaz desarrollada permite monitoreo en tiempo real, interpretación automática de los datos, y emisión de alertas inmediatas. La elección de herramientas fue guiada por la necesidad de bajo costo, alta accesibilidad, y facilidad de integración, garantizando así un sistema completo que puede ser utilizado por autoridades locales, comunidad académica y ciudadanía en general.

Tabla 6. Rangos y colores aplicados al AQI (CO)

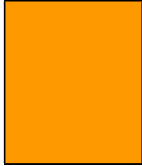


Rango AQI	Categoría	Color	Monóxido de carbono CO (ppm) [8 horas]	Acción recomendada
0–50	Buena	Verde	0 – 4,4	Es un gran día para estar activo al aire libre.
51–100	Moderada	Amarillo	4,5 – 9,4	Es un gran día para estar activo al aire libre.
101–150	Dañina para grupos	Naranja	9,5 – 12,4	Grupo sensible: Limite el esfuerzo intenso al aire libre y evite las fuentes de CO, como el tráfico denso.

Rango AQI	Categoría	Color	Monóxido de carbono CO (ppm) [8 horas]	Acción recomendada
	sensibles			
151–200	Dañina	Rojo	12,5 – 15,4	Grupo sensible: Limite el esfuerzo moderado al aire libre y evite las fuentes de CO, como el tráfico denso.
201–300	Muy dañina	Morado	15,5 – 30,4	Grupo sensible: Evite el esfuerzo al aire libre y las fuentes de CO, como el tráfico pesado.
≥ 301	Peligrosa	Vino	30,5 +	Grupo sensible: Evite el esfuerzo al aire libre y las fuentes de CO, como el tráfico denso. Resto del mundo: Limite el esfuerzo intenso al aire libre.

Fuente: [28].

Tabla 7. Rangos y colores aplicados al AQI (PM2.5)

Rango AQI	Categoría	Color	Material particulado PM2.5 (µg/m³) [24 horas]	Acción recomendada
0–50	Buena	Verde	0.0 – 9.0	Es un gran día para estar activo al aire libre.
51–100	Moderada	Amarillo	9.1 – 35.4	Personas inusualmente sensibles: Consideren acortar y reducir la intensidad de las actividades al aire libre. Refúgiense si presentan síntomas como tos o dificultad para respirar.
101–150	Dañina para grupos sensibles	Naranja	35.5–55.4	Grupos sensibles: Reduzca la duración y la intensidad de las actividades al aire libre. Está bien estar activo al aire libre, pero tome más descansos. Está atento a síntomas como tos o dificultad para respirar.  Personas con asma: Siga su plan de acción para el asma y tenga a mano medicamentos de alivio rápido.  Personas con enfermedades cardíacas: Síntomas

Rango AQI	Categoría	Color	Material particulado PM2.5 (µg/m³) [24 horas]	Acción recomendada
				como palpitaciones, dificultad para respirar o fatiga inusual pueden indicar un problema grave. Si presenta alguno de estos síntomas, comuníquese con su médico.
151–200	Dañina		55.5-125.4	<p>Grupos sensibles: Considere reprogramar o trasladar todas sus actividades al interior. Entre al interior si presenta síntomas.</p> <p>Personas con asma: Siga su plan de acción para el asma y tenga a mano medicamentos de alivio rápido.</p> <p>Personas con enfermedades cardíacas: Síntomas como palpitaciones, dificultad para respirar o fatiga inusual pueden indicar un problema grave. Si presenta alguno de estos síntomas, contacte a su médico.</p> <p>Resto del mundo: Mantenga las actividades al aire libre más cortas y menos intensas. Entre al interior si presenta síntomas.</p>
201–300	Muy dañina		125.5-225.4	<p>Grupos sensibles: Evite toda actividad física al aire libre. Reprograme para un momento en que la calidad del aire sea mejor o realice las actividades en interiores.</p> <p>Personas con asma: Siga su plan de acción para el asma y tenga a mano medicamentos de alivio rápido.</p> <p>Personas con enfermedades cardíacas: Síntomas como palpitaciones, dificultad para respirar o fatiga inusual pueden indicar un problema grave. Si presenta alguno de estos síntomas, comuníquese con su médico.</p>

Rango AQI	Categoría	Color	Material particulado PM2.5 (µg/m³) [24 horas]	Acción recomendada
				Resto del mundo: Limite la actividad física al aire libre. Refúgiense en interiores si presenta síntomas.
≥ 301	Peligrosa	Vino	225.5+	<p>Grupos sensibles: Permanezcan en casa y mantengan un nivel de actividad ligero. Siga los consejos para mantener bajos los niveles de partículas en interiores.</p> <p>Personas con asma: Sigam su plan de acción para el asma y tengan a mano medicamentos de alivio rápido.</p> <p>Personas con enfermedades cardíacas: Síntomas como palpitaciones, dificultad para respirar o fatiga inusual pueden indicar un problema grave. Si presenta alguno de estos síntomas, comuníquese con su médico.</p> <p>Todos: Eviten toda actividad física al aire libre.</p>

Fuente: [28].

Tabla 8. Herramientas utilizadas y su justificación técnica

Herramienta	Justificación técnica
<b>Arduino IDE</b>	Utilizado para programar los microcontroladores ESP32 y establecer la lógica de lectura y transmisión de datos desde los sensores SEN0564 (CO) y PMS5003 (PM2.5). Ofrece una interfaz sencilla, gratuita y ampliamente usada en entornos académicos y de prototipado.
<b>ESP32 DEVKITV1</b>	Microcontrolador central con Wi-Fi integrado, bajo consumo energético y gran capacidad de procesamiento. Permite la transmisión de datos en tiempo real desde los sensores a la nube.
<b>ThingSpeak</b>	Plataforma en la nube especializada en IoT. Recibe datos cada 5 minutos, permite su almacenamiento, visualización, análisis mediante MATLAB, y programación de alertas mediante React, TimeControl y ThingHTTP.
<b>MATLAB Analysis</b>	Ejecuta scripts directamente dentro de ThingSpeak para calcular el AQI, promedios móviles y clasificaciones. No requiere servidores externos ni configuraciones complejas.

Herramienta	Justificación técnica
<b>Python (Google Colab)</b>	Utilizado para el análisis exploratorio de datos (EDA), entrenamiento de modelos LSTM y evaluación de métricas. Google Colab ofrece un entorno gratuito con acceso a GPU y bibliotecas integradas como Pandas, TensorFlow y Keras.
<b>Keras Tuner (Random Search)</b>	Técnica empleada para optimizar hiperparámetros del modelo LSTM, encontrando automáticamente la mejor arquitectura (unidades, dropout, time_steps). Ideal para series temporales.
<b>LSTM (Long Short-Term Memory)</b>	Modelo de red neuronal recurrente especializado en secuencias. Se adapta perfectamente a los datos temporales del proyecto (cada 5 min/hora). Permite capturar patrones temporales y predicciones precisas.
<b>Telegram</b>	Herramienta de mensajería instantánea usada para alertas en tiempo real. Conectado desde ThingSpeak mediante Webhooks usando React + ThingHTTP. Es gratuita, multiplataforma y de fácil integración.
<b>Google Sites</b>	Plataforma para mostrar la interfaz del sistema. Permite integrar gráficos de ThingSpeak mediante iframe, brindando acceso visual a usuarios sin conocimientos técnicos.

Fuente: elaboración propia.

## 9. ACTIVIDADES Y RESULTADOS OBTENIDOS OBJETIVO ESPECÍFICO 5

### Objetivo específico 5

Realizar una evaluación del sistema de monitoreo implementado, identificando lecciones aprendidas y recomendaciones para mejorar su eficacia y utilidad en la información sobre la contaminación del aire en Bogotá.

### Evaluación del sistema y resultados finales

El sistema de monitoreo desarrollado fue sometido a una evaluación integral en condiciones reales de operación para verificar su desempeño técnico, su utilidad práctica y su escalabilidad como solución de bajo costo frente a los sistemas tradicionales como la RMCAB (Red de Monitoreo de Calidad del Aire de Bogotá).

### 9.1 Evaluación del funcionamiento integral del sistema

La arquitectura final incluyó:

- Sensores SEN0564 (CO) y PMS5003 (PM2.5) instalados en tres puntos estratégicos de Bogotá.
- Microcontroladores ESP32 programados en Arduino IDE, encargados de capturar y enviar datos cada 5 minutos.
- Transmisión de datos a través de Wi-Fi hacia la nube en ThingSpeak, donde se almacenan, visualizan y procesan.
- Procesamiento de datos históricos con Python (Google Colab) para construir y entrenar modelos LSTM por cada medidor y contaminante.
- Interfaz de usuario en Google Sites, con visualización de datos en tiempo real, alertas e interpretación AQI.
- Sistema de alertas automáticas por Telegram, activadas desde ThingSpeak sin necesidad de servicios externos.

Este ecosistema demostró ser funcional, replicable y eficiente para propósitos de vigilancia ambiental ciudadana.

### 9.2 Desempeño de los modelos predictivos

Cada modelo LSTM fue ajustado individualmente mediante Random Search para encontrar los mejores hiperparámetros. Se obtuvieron métricas de alto rendimiento, demostrando que el modelo es robusto para predicción de series temporales. Los mejores resultados fueron:

- CO – Medidor 1: RMSE = 0.038 ppm,  $R^2 = 0.981$
- CO – Medidor 2: RMSE = 0.034 ppm,  $R^2 = 0.897$

- CO – Medidor 3: RMSE = 0.060 ppm,  $R^2 = 0.919$
- PM2.5 – Medidor 1: RMSE = 0.120  $\mu\text{g}/\text{m}^3$ ,  $R^2 = 0.867$
- PM2.5 – Medidor 2: RMSE = 0.175  $\mu\text{g}/\text{m}^3$ ,  $R^2 = 0.897$
- PM2.5 – Medidor 3: RMSE = 0.263  $\mu\text{g}/\text{m}^3$ ,  $R^2 = 0.917$

Estas cifras evidencian una buena capacidad de predicción, con errores mínimos y gran correlación entre los valores predichos y reales.

### 9.3 Aplicabilidad y escalabilidad

Este sistema puede ser replicado en diferentes localidades del país o incluso en otras regiones de Latinoamérica con problemáticas similares. Su bajo costo, flexibilidad tecnológica y facilidad de visualización lo hacen ideal para:

- Instituciones educativas (para enseñanza y monitoreo escolar).
- Gobiernos locales con presupuesto limitado.
- Comunidades organizadas que deseen monitorear su entorno.

Además, el uso de herramientas de código abierto y plataformas gratuitas asegura su escalabilidad sin licencias costosas.

### 9.4 Comparación con trabajos previos

En contraste con el trabajo de Pedraza Camelo [4], quien desarrolló un modelo de predicción de partículas finas (PM2.5) en la localidad de Kennedy utilizando técnicas de machine learning clásicas con datos históricos de calidad del aire, el presente proyecto propone un enfoque más integral que combina el uso de sensores físicos en tiempo real con redes neuronales LSTM (Long Short-Term Memory) para la predicción tanto de PM2.5 como de monóxido de carbono (CO) en tres zonas representativas de Bogotá. Mientras el trabajo de Pedraza se centró en una única zona geográfica y no contempló adquisición directa de datos mediante IoT, esta investigación implementa una arquitectura de monitoreo distribuida basada en microcontroladores ESP32 y sensores específicos (PMS5003 y SEN0564), lo cual permite no solo capturar datos en tiempo real, sino también generar alertas automáticas mediante Telegram. Asimismo, la comparación de métricas de desempeño revela que el modelo LSTM alcanzó valores de  $R^2$  superiores a los reportados en el trabajo anterior, evidenciando un mejor ajuste a la variabilidad temporal de los contaminantes atmosféricos monitoreados.

Al comparar los resultados obtenidos en este estudio con los reportados por Ortiz et al. [5], se identifican diferencias significativas en el enfoque metodológico y el contexto de aplicación. Mientras que el presente proyecto emplea sensores de bajo costo (SEN0564 y PMS5003) conectados a microcontroladores ESP32 para la medición directa de CO y PM2.5 en tres puntos

específicos de Bogotá, el estudio de Ortiz et al. combina modelación satelital (GEOS-CF de la NASA) con técnicas de Machine Learning (XGBoost) para ajustar las predicciones a nivel regional. En cuanto al desempeño, ambos enfoques reportan mejoras al incorporar modelos de aprendizaje automático; sin embargo, Ortiz et al. obtuvieron un coeficiente de correlación de  $r = 0.47$ , equivalente a un coeficiente de determinación de  $R^2 \approx 0.22$ , lo cual indica un ajuste moderado. En contraste, los modelos LSTM desarrollados en este trabajo alcanzaron valores de  $R^2$  superiores a 0.90 y RMSE por debajo de  $5 \mu\text{g}/\text{m}^3$ , lo que demuestra una mayor precisión y capacidad de ajuste al contar con datos en tiempo real directamente capturados en los sitios de monitoreo. Esta diferencia puede atribuirse a la granularidad de los datos, el enfoque de predicción local, y la menor dependencia de fuentes satelitales generalizadas.

## 10. CONCLUSIONES Y TRABAJOS FUTUROS

### 10.1 Conclusiones

El presente proyecto logró desarrollar un sistema funcional, escalable y de bajo costo para el monitoreo y análisis de la calidad del aire en Bogotá, aplicando tecnologías de Internet de las Cosas (IoT) y modelos de aprendizaje automático (Machine Learning) sobre contaminantes críticos como el monóxido de carbono (CO) y el material particulado fino (PM2.5).

Entre los principales logros alcanzados se destacan:

- Implementación efectiva de sensores SEN0564 y PMS5003 en tres puntos de la ciudad, conectados a microcontroladores ESP32, programados en Arduino IDE.
- Transmisión y almacenamiento continuo de datos en ThingSpeak cada 5 minutos, lo que permitió un monitoreo detallado en tiempo real.
- Cálculo automático del Índice de Calidad del Aire (AQI) utilizando scripts MATLAB dentro de ThingSpeak, de acuerdo con las normativas de la EPA.
- Desarrollo de modelos LSTM altamente precisos por contaminante y por punto geográfico, alcanzando métricas destacables como  $RMSE < 0.05$  ppm para CO y  $R^2 > 0.98$ .
- Optimización de modelos mediante Keras Tuner con Random Search, logrando identificar arquitecturas robustas adaptadas a las características temporales de los datos.
- Creación de una interfaz web con Google Sites, que integra visualización en tiempo real, gráficos AQI, recomendaciones de salud, predicciones futuras y gráficos comparativos.
- Implementación de un sistema de alertas automáticas vía Telegram, sin requerir servidores adicionales, basado en los umbrales críticos de concentración por contaminante mediante las herramientas integradas de ThingSpeak (React, TimeControl y ThingHTTP).

La experiencia del proyecto evidenció que es posible lograr el acceso al monitoreo ambiental utilizando herramientas accesibles, educativas y reproducibles. Este sistema puede ser replicado fácilmente en otras zonas urbanas o rurales con problemáticas similares.

### 10.2 Trabajos Futuros

A partir de los resultados obtenidos, se plantean las siguientes líneas de mejora y expansión:

- Extensión del sistema a más puntos de monitoreo, buscando una mayor cobertura geográfica de la ciudad.
- Inclusión de nuevos contaminantes como ozono ( $O_3$ ), dióxido de nitrógeno ( $NO_2$ ) o dióxido de azufre ( $SO_2$ ) mediante sensores específicos.
- Despliegue de una aplicación móvil que permita a los ciudadanos consultar la calidad del aire y recibir notificaciones personalizadas.

- Integración con sistemas de decisión urbana, aportando datos a alcaldías locales, universidades o planes de contingencia ambiental.
- Aplicación de otros modelos avanzados como de Deep Learning, redes bidireccionales (Bi-LSTM), redes GRU o arquitecturas híbridas con atención espacial-temporal.
- Automatización del pipeline completo (IoT, Nube, ML, Visualización) con servicios en la nube tipo AWS IoT o Google Cloud Functions.

Este proyecto representa un punto de partida hacia el desarrollo de sistemas inteligentes de vigilancia ambiental que sean sostenibles, accesibles y con impacto social directo.

## 11. REFERENCIAS BIBLIOGRÁFICAS

- [1] Aire y problemas ambientales de Bogotá, Universidad Nacional de Colombia, 2025.
- [2] Secretaría Distrital de Ambiente, «Calidad del aire en Bogotá,» [En línea]. Available: <https://www.ambientebogota.gov.co/calidad-del-aire>.
- [3] A. D. Pulido, J. D. Turriago, R. Jiménez, C. F. Torres, A. Rojas, N. Chaparro, E. Y. Ortiz, S. Granados, J. Rodríguez, V. Berrío, I. C. Figueroa, Á. V. Bohórquez, S. Rojas y J. A. L, «Inventario Nacional y Departamental de Gases Efecto Invernadero,» IDEAM - Instituto de Hidrología, Meteorología y Estudios Ambientales, Bogotá, 2016.
- [4] J. C. Pedraza Camelo, «Prototipo de un modelo de machine learning para la predicción de partículas de contaminación atmosférica finas, en la localidad de Kennedy en la ciudad de Bogotá,» Trabajo de Grado. Especialización en Ingeniería de Software. Universidad Distrital Francisco José de Caldas, Bogotá. Colombia, 2019.
- [5] E. Y. Ortiz, C. A. Keller, B. Cardenas, H. E. Sáenz, K. Wakabayashi, J. Seddon, E. Castillero, A. Retama, L. A. Hernandez y F. Grajales, «Combination of GEOS-CF model with Machine Learning as a tool for forecasting regional pollution in Bogotá,» de *2021 Congreso Colombiano y Conferencia Internacional de Calidad de Aire y Salud Pública (CASAP)*, Bogotá, Colombia, 2021.
- [6] M. A. Guevara Luna, J. M. Vargas Gonzalez, S. Torres Muñoz y L. C. Belalcazar Ceron, «Evaluación espacial y temporal de PM10 y PM2,5 en Colombia utilizando información satelital (CAM5, MODIS-AOD) y mediciones de calidad del aire en superficie,» de *2019 Congreso Colombiano y Conferencia Internacional de Calidad de Aire y Salud Pública (CASAP)*, Barranquilla, Colombia, 2019.
- [7] J. Salazar y S. Silvestre, «Internet de las cosas,» TechPedia - European Virtual Learning Platform for Electrical and Information Engineering.
- [8] «Air Quality Prediction using Machine Learning Algorithm,» Erode, 2023.
- [9] «Load Prediction in Electrical Networks using LSTM,» Morelia, 2021.
- [10] Tuner, «KerasTuner: Hyperparam Tuning,» 2020. [En línea]. Available: [https://keras.io/keras\\_tuner/](https://keras.io/keras_tuner/).
- [11] «Adam: A method for Stochastic Optimization,» 2015.
- [12] «Real-Time AQI Forecasting With High-Throughput Using IoT Devices,» New Delhi, 2024.
- [13] B. D. Parameshchari, G. M. Siddesh, V. Sridhar, M. Latha, K. N. A. Sattar y G. Manjula, «Prediction and Analysis of Air Quality Index using Machine Learning Algorithms,» de *IEEE International Conference on Data Science and Information System, ICDSIS 2022*, Hassan, India, 2022.
- [14] K. Chakradhar Reddy, K. Nagarjuna Reddy, K. Brahmaji Prasad y P. Selvi Rajendran, «The Prediction of Quality of the Air Using Supervised Learning,» de *Proceedings of the 6th International Conference on Communication and Electronics Systems, ICCES 2021*, Coimbatre, India, 2021.

- [15] U. Mahalingam, K. Elangovan, H. Dobhal, C. Valliappa, S. Shrestha y G. Kedam, «A Machine Learning Model for Air Quality Prediction for Smart Cities,» de *2019 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, Chennai, India, 2019.
- [16] «Prediction of PM2.5 concentration based on a CNN-LSTM neural network algorithm,» *PeerJ*, 2024.
- [17] «Application of CNN-LSTM Algorithm for PM2.5 Concentration Forecasting,» *Atmosphere*, 2023.
- [18] «Hybrid Deep Learning for Air Quality Prediction,» *JISEM*, 2024.
- [19] O. J. M. Sanmiguel, «Predicción de la calidad del aire de la ciudad de Medellín y su área metropolitana mediante el uso de redes neuronales recurrentes,» Universitat Oberta de Catalunya, Medellín, 2020.
- [20] C. R. Pastor, «TÉCNICAS DE APRENDIZAJE AUTOMÁTICO PARA EL ANÁLISIS DE DATOS DE CALIDAD DEL AIRE,» Universidad Autónoma de Madrid - Escuela politécnica superior, Madrid, 2017.
- [21] G. R. N. Fernández, «Aplicación de modelo XGB para predicción de calidad de aire, Bogotá - Colombia,» *Revista Apuntes de Ciencia e ingeniería*, Bogotá, 2023.
- [22] «SEN0564-CO: CO gas sensor datasheet,» 2024.
- [23] «PM2.5 Air Quality Sensor,» 2025.
- [24] «ESP32 Series Datasheet Version 4.9,» 2024.
- [25] Arduino programming notebook, Arduino.cc, 2007.
- [26] Thingspeak, «ThingSpeak,» [En línea]. Available: <https://thingspeak.mathworks.com/>. [Último acceso: 15 Febrero 2025].
- [27] B. D.J., «Introducción a IoT con Arduino y ThingSpeak,» 2025. [En línea]. Available: <https://www.bolanosdj.com.ar/MOVIL/ARDUINO2/iotintrothingspeak.pdf>.
- [28] U.S. Environmental Protection Agency, «Technical Assistance Document for the Reporting of Daily Air Quality – the Air Quality Index (AQI),» Office of Air Quality Planning and Standards, Research Triangle Park, NC, 2024.
- [29] Sigma electrónica, «<https://www.sigmaelectronica.net/producto/sen0564/>,» [En línea]. Available: <https://www.sigmaelectronica.net/producto/sen0564/>. [Último acceso: Junio 2023].
- [30] google, «<https://sites.google.com/>,» [En línea]. Available: <https://sites.google.com/>. [Último acceso: mayo 2025].
- [31] Telegram, «<https://web.telegram.org/a/>,» [En línea]. Available: <https://web.telegram.org/a/>. [Último acceso: mayo 2025].
- [32] <https://colab.research.google.com/?pli=1>, «<https://colab.research.google.com/?pli=1>,» [En línea]. Available: <https://colab.research.google.com/?pli=1>. [Último acceso: abril 2024].
- [33] J. P. Pacheco Pérez, «Clasificación de ciberataques, fallas y eventos de operación normal en

- subestaciones digitales utilizando el protocolo GOOSE a partir de redes LSTM,» Universidad de los Andes, Bogotá, 2025.
- [34] C. Hernández Hernández y J. S. Forero Criollo, «MODELOS DE PREDICCIÓN ESTOCÁSTICA PARA BITCOIN: UNA EVALUACIÓN DE MÉTODOS Y DESEMPEÑO,» UNIVERSIDAD EAFIT, BOGOTÁ, 2023.
- [35] S. Hochreiter y J. Schmidhuber, «Long Short-Term Memory,» *Neural Computation*, vol. 9, nº 8, p. 1735–1780, 1997.
- [36] J. Bergstra y Y. Bengio, «Random Search for Hyper-Parameter Optimization,» *Journal of Machine Learning Research*, vol. 13, p. 281–305, 2012.
- [37] O. J. Maestre Sanmiguel, «Predicción de la calidad del aire de la ciudad de Medellín y su área metropolitana mediante el uso de redes neuronales recurrentes,» Universidad Oberta de Catalunya, Medellin, 2020.
- [38] RMCAB, «Red de monitoreo de la calidad del aire en Bogotá,» [En línea]. Available: <http://rmcab.ambientebogota.gov.co/home/map>. [Último acceso: 1 Junio 2025].
- [39] Amazon.com, «<https://www.amazon.com/-/es/PMS5003-detecci%C3%B3n-concentraci%C3%B3n-part%C3%ADculas-acondicionado/dp/B0BHZTCK8J>,» [En línea]. Available: <https://www.amazon.com/-/es/PMS5003-detecci%C3%B3n-concentraci%C3%B3n-part%C3%ADculas-acondicionado/dp/B0BHZTCK8J>. [Último acceso: Junio 2023].
- [42] «Inventario de Emisiones de Bogotá,» Secretaría Distrital de Ambiente – SDA, Bogotá, 2024.
- [45] «The Prediction of Quality of the Air Using Supervised Learning,» Coimbatre, 2021.
- [46] «Inventario Nacional y Departamental de Gases Efecto Invernadero,» IDEAM, Bogotá, 2016.
- [48] «Prediction and Analysis of Air Quality Index using Machine Learning Algorithms,» Hassan, 2022.
- [49] «Combination of GEOS-CF model with Machine Learning as a tool for forecasting regional pollution in Bogotá,» Bogotá, 2021.
- [51] «A Machine Learning Model for Air Quality Prediction for Smart Cities,» Chennai, 2019.
- [52] «Air Quality Prediction using Machine Learning Algorithms – A Review,» Greater Noida, 2020.
- [53] TinyML Cookbook, Packt Publishing, 2023.
- [54] «Instituto de Hidrología, Meteorología y Estudios Ambientales,» IDEAM, Bogotá, 2024.
- [55] «Evaluación espacial y temporal de PM10 y PM2.5 en Colombia utilizando información satelital,» Barranquilla, 2019.
- [56] «PM NAAQS air quality index fact sheet,» US EPA, 2024.
- [57] «Air Quality Criteria for Ozone,» US EPA, 2002.
- [58] «Índice de calidad del aire – SIMA,» DANE, 2012.
- [59] Python para todos, USC, 2025.
- [60] «Exposición personal a Material Particulado 2.5 en sistemas de transporte masivo de Bogotá

y Medellín,» Barranquilla, 2019.

[61] «Distribution of particulate matter in an urban area of Bogotá D.C.,» Barranquilla, 2019.

## ANEXOS

### **Anexo 1: Página web Google Sites**

<https://sites.google.com/view/iot-machine-learning-proyecto/iot-machine-learning>

### **Anexo 2: Datos Zona UCompensar Teusaquillo (Medidor 1) - ThingSpeak IoT**

<https://thingspeak.mathworks.com/channels/2986083>

### **Anexo 3: Datos Zona UManuela (Medidor 2) - ThingSpeak IoT**

<https://thingspeak.mathworks.com/channels/2986097>

### **Anexo 4: Datos Zona centro (Medidor 3) - ThingSpeak IoT**

<https://thingspeak.mathworks.com/channels/2986109>

### **Anexo 5: Modelo CO medidor 1**

<https://colab.research.google.com/drive/1xHOuA5ZdmNHgRouRlLjSwH0UEjTP6DBI?usp=sharing>

### **Anexo 6: Hiperparámetros CO medidor 1**

<https://colab.research.google.com/drive/14RmhbkNYA6qNIFKRMofhjGGCA1filapQ?usp=sharing>

### **Anexo 7: EDA CO medidor 1**

<https://colab.research.google.com/drive/1bHSNzIR8z8BaJUXyolbLvTW3o0M5iYjN?usp=sharing>

### **Anexo 8: Modelo CO medidor 2**

<https://colab.research.google.com/drive/1flfNi-h9AHzrjhMzd6OUU7sF6zU-Jnxs?usp=sharing>

### **Anexo 9: Hiperparámetros CO medidor 2**

[https://colab.research.google.com/drive/1FskZvCBO9KIYmM\\_CKqmhoFw0f9JIYaun?usp=sharing](https://colab.research.google.com/drive/1FskZvCBO9KIYmM_CKqmhoFw0f9JIYaun?usp=sharing)

### **Anexo 10: EDA CO medidor 2**

[https://colab.research.google.com/drive/1NIFoiCgg3CziutO5qR2pp\\_RDIB4slbWt?usp=sharing](https://colab.research.google.com/drive/1NIFoiCgg3CziutO5qR2pp_RDIB4slbWt?usp=sharing)

### **Anexo 11: Modelo CO medidor 3**

[https://colab.research.google.com/drive/17gYbAylciOX\\_vZIVVSL5nlay-DgJY4k?usp=sharing](https://colab.research.google.com/drive/17gYbAylciOX_vZIVVSL5nlay-DgJY4k?usp=sharing)

### **Anexo 12: Hiperparámetros CO medidor 3**

[https://colab.research.google.com/drive/1e7lByx2aP6EWl\\_xTltsEND5lwnQ-84xD?usp=sharing](https://colab.research.google.com/drive/1e7lByx2aP6EWl_xTltsEND5lwnQ-84xD?usp=sharing)

**Anexo 13: EDA CO medidor 3**

<https://colab.research.google.com/drive/1b0q-Gx3EbsN7sjNRDrVJZ1kZSMPeJnwc?usp=sharing>

**Anexo 14: Modelo PM2.5 medidor 1**

<https://colab.research.google.com/drive/1cF5XzjntDUWnhTAjoyKN3JPZCXInilUG?usp=sharing>

**Anexo 15: Hiperparámetros PM2.5 medidor 1**

[https://colab.research.google.com/drive/104\\_z0mz1X2czYDISHc-5UUGjXgijpU3-?usp=sharing](https://colab.research.google.com/drive/104_z0mz1X2czYDISHc-5UUGjXgijpU3-?usp=sharing)

**Anexo 16: EDA PM2.5 medidor 1**

[https://colab.research.google.com/drive/1jKv5TwiVwz4geGC1V0\\_YUpzBTfRkHbWb?usp=sharing](https://colab.research.google.com/drive/1jKv5TwiVwz4geGC1V0_YUpzBTfRkHbWb?usp=sharing)

**Anexo 17: Modelo PM2.5 medidor 2**

[https://colab.research.google.com/drive/1O6yq\\_oq1F3A7tbU9oTd0t62nrQKntV6L?usp=sharing](https://colab.research.google.com/drive/1O6yq_oq1F3A7tbU9oTd0t62nrQKntV6L?usp=sharing)

**Anexo 18: Hiperparámetros PM2.5 medidor 2**

<https://colab.research.google.com/drive/1UZ-9T9jgDsG-xb8z3hiWY2a-MSAuUUEE?usp=sharing>

**Anexo 19: EDA PM2.5 medidor 2**

<https://colab.research.google.com/drive/1P8iTd2ox9oqOqVmxsB-WY1X0eTM2XNaS?usp=sharing>

**Anexo 20: Modelo PM2.5 medidor 3**

[https://colab.research.google.com/drive/1ulU4WISqRdtK60c4fp63z\\_RSIT1loB5d?usp=sharing](https://colab.research.google.com/drive/1ulU4WISqRdtK60c4fp63z_RSIT1loB5d?usp=sharing)

**Anexo 21: Hiperparámetros PM2.5 medidor 3**

<https://colab.research.google.com/drive/1Brij6celptNs43ll6H7ZdZNxy8A2aRPM?usp=sharing>

**Anexo 22: EDA PM2.5 medidor 3**

<https://colab.research.google.com/drive/1tYnA7eR0ai85p9jGmwNyZ-LeXzkNwQr?usp=sharing>